



Guia do usuário

AWS Schema Conversion Tool



Versão 1.0.672

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

AWS Schema Conversion Tool: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o AWS SCT?	1
Visão geral da conversão de esquema	5
Dando feedback	7
Instalação, verificação e atualização	8
Instalando AWS SCT	8
Verificando o download do AWS SCT arquivo	10
Verificando a soma de verificação do arquivo AWS SCT	10
Verificando os arquivos AWS SCT RPM no Fedora	11
Verificando os arquivos AWS SCT DEB no Ubuntu	12
Verificando o arquivo AWS SCT MSI no Microsoft Windows	12
Baixar os drivers de banco de dados necessários	13
Instalação de drivers JDBC no Linux	16
Armazenamento de caminhos do driver nas configurações globais	17
Atualizando AWS SCT	18
AWS SCT CLI	19
Uso da interface de usuário da AWS SCT	20
A janela do projeto	20
Iniciar o AWS SCT	22
Criando um projeto	22
Usando um novo assistente de projeto	23
Salvando e abrindo um projeto	27
Adicionando um servidor	28
Usando um modo off-line	29
Usando os filtros de árvore	30
.....	31
Importar uma lista de arquivos para o filtro de árvore	33
Ocultando esquemas	34
Gerenciando o relatório de avaliação de migração do banco de dados	35
Como converter seu esquema	39
Aplicando o código convertido	42
Armazenando perfis da AWS	43
Armazenando credenciais da AWS	44
Configurando o perfil padrão de um projeto	46
Permissões para usar o perfil de serviço da AWS	46

Usar o AWS Secrets Manager	47
Armazenando senhas de banco de dados	48
Usando a visualização Union All para projetos com tabelas particionadas	48
Atalhos de teclado	49
Conceitos básicos	51
Origens para a AWS SCT	53
Como criptografar conexões do Amazon RDS	54
Como usar o Apache Cassandra como origem	57
Como se conectar ao Apache Cassandra como origem	57
Como usar o Apache Hadoop como origem	59
Pré-requisitos de uso do Apache Hadoop como origem	60
Permissões para o Hive como origem	61
Permissões para o HDFS como origem	61
Permissões para o HDFS como destino	62
Como se conectar ao Apache Hadoop como origem	62
Como se conectar ao Hive e ao HDFS	64
Como se conectar ao Amazon EMR como destino	67
Como usar o Apache Oozie como origem	70
Pré-requisitos	70
Como se conectar ao Apache Oozie como origem	71
Permissões para AWS Lambda	72
Como se conectar a AWS Step Functions como destino	74
Como usar um banco de dados Azure SQL como origem	76
Privilégios do banco de dados Azure SQL	76
Como se conectar ao banco de dados Azure SQL de origem	76
Como usar o IBM Db2 para z/OS como origem	78
Pré-requisitos para Db2 para z/OS	78
Privilégios do Db2 para z/OS	79
Como se conectar ao Db2 para z/OS como origem	80
Privilégios do MySQL como um destino	82
Privilégios do PostgreSQL como destino	84
Configurações de conversão do Db2 para z/OS para PostgreSQL	85
Usar o IBM Db2 LUW como origem	86
Privilégios do Db2 LUW	87
Como se conectar ao Db2 LUW como origem	89
Db2 LUW para PostgreSQL	92

Db2 LUW para MySQL	94
Usar o MySQL como origem	96
Privilégios do MySQL	96
Conectar-se ao MySQL como origem	96
Privilégios do PostgreSQL como destino	99
Usar um banco de dados Oracle como origem	100
Privilégios do Oracle	101
Conectar-se ao Oracle como origem	101
Oracle para PostgreSQL	106
Oracle para MySQL	112
Oracle para Amazon RDS para Oracle	122
Usar o PostgreSQL como origem	129
Privilégios para o PostgreSQL	129
Conectar-se ao PostgreSQL como origem	130
Privilégios do MySQL como um destino	132
Usando o SAP ASE (Sybase ASE) como origem	134
Privilégios do SAP ASE	134
Conectando-se ao SAP ASE como origem	135
Privilégios do MySQL como um destino	137
Configurações de conversão do SAP ASE para MySQL	139
Privilégios do PostgreSQL como destino	140
Configurações de conversão de SAP ASE para PostgreSQL	140
Usar o SQL Server como origem	141
Privilégios do Microsoft SQL Server	142
Usando a autenticação do Windows com o Microsoft SQL Server	144
Conectando-se ao SQL Server como origem	146
SQL Server para MySQL	149
SQL Server para PostgreSQL	153
SQL Server para Amazon RDS SQL Server	190
Origens de data warehouse para AWS SCT	191
Como usar o Amazon Redshift como origem	192
Como usar o Azure Synapse Analytics como origem	198
Como usar o BigQuery como origem	203
Como usar o banco de dados Greenplum como origem	209
Como usar o Netezza como origem	215
Como usar o data warehouse do Oracle como origem	225

Como usar o Snowflake como origem	233
Como usar o data warehouse do SQL Server como origem	242
Como usar o Teradata como origem	249
Como usar o Vertica como origem	265
Criação de regras de mapeamento	272
Nova regra	273
Gerenciar regras	273
Alvos virtuais	275
Limitações	276
Criar relatórios de conversão	277
Relatórios de avaliação de migração	277
Como criar um relatório de avaliação de migração do banco de dados	278
Como visualizar o relatório de avaliação	279
Como salvar o relatório de avaliação	283
Como configurar o relatório de avaliação	285
Como criar um relatório de avaliação multisservidor	289
Como converter esquemas de bancos de dados	299
Como criar regras de migração	301
Como criar regras de migração	302
Como exportar regras de migração	304
Como converter seu esquema	304
Converter o esquema	305
Como editar o esquema convertido	308
Como limpar um esquema convertido	309
Como lidar com conversões manuais	309
Como modificar seu esquema de origem	309
Como modificar seu esquema de destino	309
Como atualizar e recarregar o esquema convertido	310
Como salvar e aplicar seu esquema	311
Como salvar o esquema convertido	311
Como aplicar seu esquema convertido	312
O esquema do pacote de extensão	313
Como comparar esquemas	313
Objetos relacionados transformados	315
Como converter esquemas de data warehouse em Amazon Redshift	316
Permissões para o Amazon Redshift	317

Como escolher estratégias e regras de otimização	319
Como coletar ou carregar as estatísticas	320
Como criar regras de migração	322
Como criar regras de migração	323
Como exportar regras de migração	325
Como converter seu esquema	325
Converter o esquema	325
Como editar o esquema convertido	328
Como limpar um esquema convertido	329
Como gerenciar e personalizar chaves	329
Tópicos relacionados	330
Como criar e usar o relatório de avaliação	330
Como criar um relatório de avaliação de migração do banco de dados	330
Resumo	331
Itens de ação	333
Como salvar o relatório de avaliação	333
Como lidar com conversões manuais	334
Como modificar seu esquema de origem	335
Como modificar seu esquema de destino	335
Como atualizar e recarregar o esquema convertido	336
Como salvar e aplicar seu esquema convertido	336
Como salvar seu esquema convertido em um arquivo	337
Como aplicar seu esquema convertido	337
O esquema do pacote de extensão	338
Bibliotecas Python	339
Como otimizar o Amazon Redshift	339
Como otimizar seu banco de dados do Amazon Redshift	339
Como converter processos de ETL	342
Como converter processos de ETL para o AWS Glue	343
Pré-requisitos	344
Catálogo de dados do AWS Glue	345
Limitações	345
Etapa 1: criar um novo projeto	347
Etapa 2: criar um tarefa do AWS Glue	348
Como converter processos de ETL usando a API Python para o AWS Glue	349
Etapa 1: criar um banco de dados	350

Etapa 2: criar uma conexão	351
Etapa 3: criar um crawler do AWS Glue	352
Como converter scripts de ETL da Informatica	354
Como converter SSIS em AWS Glue	359
Componentes do SSIS compatíveis	363
Como converter SSIS em AWS Glue Studio	365
Pré-requisitos	366
Como adicionar pacotes SSIS ao seu projeto da AWS SCT	368
Como converter pacotes SSIS	368
Criação de tarefas do AWS Glue Studio	369
Como criar um relatório de avaliação de conversão do SSIS	371
Componentes do SSIS compatíveis	372
Como converter Teradata BTEQ em Amazon Redshift RSQL	373
Adicionando scripts BTEQ ao seu projeto da AWS SCT	375
Como configurar variáveis de substituição em scripts BTEQ	376
Conversão de scripts BTEQ	377
Como gerenciar scripts BTEQ	377
Como criar um relatório de avaliação de conversão de script BTEQ	378
Como editar e salvar seus scripts BTEQ convertidos	379
Como converter scripts de shell em Amazon Redshift RSQL	379
Adicionando scripts de shell ao seu projeto da AWS SCT	380
Como configurar variáveis de substituição em scripts de shell	381
Como converter scripts de shell	382
Como gerenciar scripts de shell	383
Como criar um relatório de avaliação de conversão de script de shell	383
Como editar e salvar seus scripts de shell convertidos	384
Como converter scripts de tarefa do Teradata FastExport em Amazon Redshift RSQL	384
Como adicionar scripts de tarefa do FastExport ao seu projeto da AWS SCT	385
Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport	386
Como converter scripts de tarefa do FastExport	387
Como gerenciar scripts de tarefa do Teradata FastExport	388
Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastExport	389
Como editar e salvar seus scripts de tarefa do Teradata FastExport convertidos	390
Como converter scripts de tarefa do Teradata FastLoad em Amazon Redshift RSQL	390
Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT	391

Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad	392
Como converter scripts de tarefa do FastLoad	394
Como gerenciar scripts de tarefa do FastLoad	395
Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastLoad	395
Como editar e salvar seus scripts de tarefa do Teradata FastLoad convertidos	396
Como converter scripts de tarefa do Teradata MultiLoad em Amazon Redshift RSQL	397
Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT	397
Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad	399
Como converter scripts de tarefa do MultiLoad	400
Como gerenciar scripts de tarefa do MultiLoad	401
Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata MultiLoad	401
Como editar e salvar seus scripts de tarefa do Teradata MultiLoad convertidos	402
Como migrar estruturas de big data	404
Como migrar o Apache Hadoop para o Amazon EMR	404
Visão geral	405
Etapa 1: conectar-se aos clusters do Hadoop	406
Etapa 2: configurar as regras de mapeamento	406
Etapa 3: criar um relatório de avaliação	408
Etapa 4: migrar seu cluster do Apache Hadoop para o Amazon EMR	409
Como executar seu script de CLI	410
Como gerenciar projeto de migração	411
Como converter o Apache Oozie para AWS Step Functions	413
Visão geral	413
Etapa 1: conectar-se aos seus serviços de origem e destino	415
Etapa 2: configurar as regras de mapeamento	415
Etapa 3: configurar parâmetros	416
Etapa 4: criar um relatório de avaliação	418
Etapa 5: converter seus fluxos de trabalho do Apache Oozie para AWS Step Functions	419
Como executar seu script de CLI	422
Nós suportados	422
Uso do AWS SCT com a AWS DMS	424
Como usar um agente de replicação da AWS SCT com o AWS DMS	424
Como usar um agente de extração de dados da AWS SCT com o AWS DMS	424
Como aumentar os níveis de registro ao usar a AWS SCT com o AWS DMS	425

Migração de um data warehouse para o Amazon Redshift	427
Pré-requisitos	430
Configurações do Amazon S3	430
Como assumir perfis do IAM	431
Configurações de segurança	433
Definições de configuração	433
Como instalar agentes do	434
Como configurar atendentes	436
Como instalar e configurar atendentes de cópia dedicados	437
Como iniciar atendentes	439
Como registrar atendentes	440
Ocultar e recuperar informações para um agente AWS SCT	440
Como criar regras de migração de dados	442
Como alterar as configurações de extrator e da cópia para migração de dados	443
Como classificar dados	446
Criando, executando e monitorando uma AWS SCT tarefa	448
Como exportar e importar uma tarefa de extração de dados	452
Extração de dados usando um dispositivo AWS Snowball Edge	453
tep-by-step Procedimentos S para migrar dados usando um AWS SCT Edge AWS Snowball	454
Saída da tarefa de extração de dados	457
Como usar o particionamento virtual	459
Limites ao criar um particionamento virtual	459
Tipo de Partição RANGE	459
Tipo de partição LIST	461
Tipo de partição DATE AUTO SPLIT	462
Como usar o particionamento nativo	463
Trabalhar com LOBs	464
Melhores práticas e solução de problemas	465
Como converter o aplicativo SQL	467
Visão geral da conversão de aplicativos SQL	467
Como converter o código SQL em seus aplicativos	468
Como criar projetos genéricos de conversão de aplicativos	468
Como gerenciar projetos de conversão de aplicativos	473
Como analisar e converter seu código SQL	474
Como criar e usar o relatório de avaliação	475

Como editar e gravar seu código SQL convertido	476
Como converter o código SQL em aplicativos C#	476
Como criar projetos de conversão de aplicativos C#	477
Como converter o código SQL em seu aplicativo C#	478
Como salvar o código do aplicativo convertido	480
Como gerenciar projetos de conversão de aplicativos C#	480
Como criar um relatório de avaliação de conversão de aplicativos C#	481
Como converter o código SQL em aplicativos C++	482
Como criar projetos de conversão de aplicativos C++	483
Como converter o código SQL em seu aplicativo C++	484
Como salvar o código do aplicativo convertido	486
Como gerenciar projetos de conversão de aplicativos C++	487
Como criar um relatório de avaliação de conversão de aplicativos C++	488
Como converter o código SQL em aplicativos Java	489
Como criar projetos de conversão de aplicativos Java	490
Como converter o código SQL em seu aplicativo Java	492
Como salvar o código do aplicativo convertido	494
Como gerenciar projetos de conversão de aplicativos Java	494
Como criar um relatório de avaliação de conversão de aplicativos Java	495
Como converter o código SQL em aplicativos Pro*C	497
Como criar projetos de conversão de aplicativos Pro*C	497
Como converter o código SQL em seu aplicativo Pro*C	498
Como editar e salvar o código do aplicativo convertido	500
Como gerenciar projetos de conversão de aplicativos Pro*C	501
Como criar um relatório de avaliação de conversão de aplicativos Pro*C	502
Como usar pacotes de extensão	504
Permissões para usar o pacote de extensão	505
Como usar o esquema do pacote de extensão	507
Bibliotecas personalizadas para pacotes de extensão	507
Como aplicar o pacote de extensão	508
Usando as funções do Lambda do pacote de extensão da AWS SCT	510
Usando AWS Lambda funções para emular a funcionalidade do banco de dados	510
Como aplicar o pacote de extensão para oferecer suporte às funções do Lambda	511
Configurando as funções do pacote de extensão	512
Práticas recomendadas	514
Como configurar memória adicional	514

Pasta padrão do projeto	514
Como aumentar a velocidade da migração de dados	515
Como aumentar as informações de registro em log	515
Solução de problemas	518
Não é possível carregar objetos em um banco de dados de origem da Oracle	518
Mensagem de aviso	518
Referência de CLI	520
Pré-requisitos	520
Modo interativo	520
Exemplos	522
Obter cenários de CLI	522
Exemplos	527
Editar cenários de CLI	527
Modo de script	528
Exemplos	529
Material de referência	530
Notas de release	531
Notas de lançamento — 676	531
Notas de lançamento — 675	536
Notas de release: 674	539
Notas de release: 673	546
Notas de release: 672	551
Notas de release: 671	559
Notas de release: 670	568
Notas de release: 669	573
Notas de release: 668	578
Notas de release: 667	585
Notas de release: 666	589
Notas de release: 665	594
Notas de release: 664	597
Notas de release: 663	601
Notas de release: 662	604
Notas de release: 661	609
Notas de release: 660	614
Notas de release: 659	618
Notas de release: 658	622

Notas de release: 657	627
Notas de release: 656	632
Notas de release: 655	635
Notas de release: 654	638
Notas de release: 653	641
Notas de release: 652	643
Notas de release: 651	646
Notas de release: 650	648
Notas de release: 649	650
Notas de release: 648	653
Notas de release: 647	654
Notas de release: 646	656
Notas de release: 645	658
Notas de release: 644	659
Notas de release: 642	661
Notas de release: 641	663
Notas de release: 640	664
Alterações do release 1.0.640 do Oracle	664
Alterações da release 1.0.640 do Microsoft SQL Server	670
Alterações do MySQL versão 1.0.640	674
Alterações da release 1.0.640 do PostgreSQL	675
Alterações do release 1.0.640 do Db2 LUW	678
Alterações na release 1.0.640 do Teradata	679
Alterações da release 1.0.640 para outros mecanismos	681
Histórico do documentos	683
Atualizações anteriores	699
.....	dccvii

O que é o AWS Schema Conversion Tool?

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter seu esquema de bancos de dados existentes de um mecanismo de banco de dados para outro. Você pode converter o esquema OLTP relacional ou o esquema de data warehouse. Seu esquema convertido é adequado para um Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, PostgreSQL DB, um cluster Amazon Aurora DB ou um cluster Amazon Redshift. O esquema convertido também pode ser usado com um banco de dados em uma instância do Amazon EC2 ou armazenado em forma de dados em um bucket do Amazon S3.

A AWS SCT oferece suporte a vários padrões do setor, inclusive Federal Information Processing Standards (FIPS) para conexões a um bucket do Amazon S3 ou outro recurso da AWS. A AWS SCT também está em conformidade com o Federal Risk and Authorization Management Program (FedRAMP). Para obter detalhes sobre a AWS e os esforços de conformidade, consulte os serviços [AWS no escopo pelo programa de conformidade](#).

A AWS SCT dá suporte às seguintes conversões de OLTP.

Fonte do banco de dados	Bancos de dados de destino:
IBM Db2 para z/OS (versão 12)	Amazon Aurora Edição compatível com MySQL (Aurora MySQL), Amazon Aurora Edição compatível com PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL Para obter mais informações, consulte Como usar o IBM Db2 para z/OS como origem .
IBM Db2 LUW (versões 9.1, 9.5, 9.7, 10.5, 11.1 e 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Para obter mais informações, consulte Usar o IBM Db2 LUW como origem .
Banco de dados do Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Fonte do banco de dados	Bancos de dados de destino:
	<p>Para obter mais informações, consulte Como usar um banco de dados Azure SQL como origem.</p>
Microsoft SQL Server (versões 2008 R2, 2012, 2014, 2016, 2017, 2019 e 2022).	<p>Aurora MySQL, Aurora PostgreSQL, Babelfish para Aurora PostgreSQL (somente para relatórios de avaliação), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL</p> <p>Para obter mais informações, consulte Usar o SQL Server como origem.</p>
MySQL (versão 5.5 e superior)	<p>Aurora PostgreSQL, MySQL, PostgreSQL</p> <p>Para obter mais informações, consulte Usar o MySQL como origem.</p> <p>Você pode migrar o esquema e os dados do MySQL para um cluster de banco de dados do Aurora MySQL sem usar a AWS SCT. Para obter mais informações, consulte Migração de dados para um cluster de banco de dados do Amazon Aurora.</p>
Oracle (versão 10.1 e superior)	<p>Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL</p> <p>Para obter mais informações, consulte Usar um banco de dados Oracle como origem.</p>
PostgreSQL (versão 9.1 e superior)	<p>Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL</p> <p>Para obter mais informações, consulte Usar o PostgreSQL como origem.</p>

Fonte do banco de dados	Bancos de dados de destino:
SAP ASE (versões 12.5.4, 15.0.2, 15.5, 15.7 e 16.0)	<p>Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL</p> <p>Para obter mais informações, consulte Usando o SAP ASE (Sybase ASE) como origem.</p>

A AWS SCT oferece suporte às conversões de data warehouse a seguir.

Data warehouse de origem	Data warehouse de destino
Amazon Redshift	<p>Amazon Redshift</p> <p>Para obter mais informações, consulte Como usar o Amazon Redshift como origem.</p>
Azure Synapse Analytics	<p>Amazon Redshift</p> <p>Para obter mais informações, consulte Como usar o Azure Synapse Analytics como origem.</p>
BigQuery	<p>Amazon Redshift</p> <p>Para obter mais informações, consulte Como usar o BigQuery como origem.</p>
Banco de dados Greenplum (versões 4.3 e 6.21)	<p>Amazon Redshift</p> <p>Para obter mais informações, consulte Como usar o banco de dados Greenplum como origem.</p>
Microsoft SQL Server (versão 2008 e superior)	<p>Amazon Redshift</p> <p>Para obter mais informações, consulte Como usar o data warehouse do SQL Server como origem.</p>

Data warehouse de origem	Data warehouse de destino
Netezza (versão 7.0.3 e superior)	Amazon Redshift Para obter mais informações, consulte Como usar o Netezza como origem.
Oracle (versão 10.1 e superior)	Amazon Redshift Para obter mais informações, consulte Como usar o data warehouse do Oracle como origem.
Snowflake (versão 3)	Amazon Redshift Para obter mais informações, consulte Como usar o Snowflake como origem.
Teradata (versão 13 e superior)	Amazon Redshift Para obter mais informações, consulte Como usar o Teradata como origem.
Vertica (versão 7.2.2 e superior)	Amazon Redshift Para obter mais informações, consulte Como usar o Vertica como origem.

A AWS SCT dá suporte às seguintes conversões de banco de dados NoSQL.

Fonte do banco de dados	Bancos de dados de destino:
Apache Cassandra (versões 2.1.x, 2.2.16 e 3.11.x)	Amazon DynamoDB Para obter mais informações, consulte Como usar o Apache Cassandra como origem.

A AWS SCT é compatível com conversões dos seguintes processos de extração, transformação e carregamento (ETL). Para obter mais informações, consulte [Como converter processos de ETL.](#)

Origem	Destino
Scripts de ETL da Informatica	Informatica
Pacotes de ETL do Microsoft SQL Server Integration Services (SSIS)	AWS Glue ou AWS Glue Studio
Scripts de shell com comandos incorporados do Teradata Basic Teradata Query (BTEQ)	Amazon Redshift RSQL
Scripts de ETL do Teradata BTEQ	AWS Glue ou Amazon Redshift RSQL
Scripts de tarefa do Teradata FastExport	Amazon Redshift RSQL
Scripts de tarefa do Teradata FastLoad	Amazon Redshift RSQL
Scripts de tarefa do Teradata MultiLoad	Amazon Redshift RSQL

A AWS SCT é compatível com as seguintes migrações da estrutura de big data. Para obter mais informações, consulte [Como migrar estruturas de big data](#).

Origem	Destino
Apache Hive (versão 0.13.0 e superior)	Hive no Amazon EMR
Apache HDFS	Amazon S3 ou HDFS no Amazon EMR
Apache Oozie	AWS Step Functions

Visão geral da conversão de esquema

A AWS SCT oferece uma interface de usuário baseada em projeto que permite converter automaticamente o esquema do banco de dados de origem em um formato que seja compatível com a instância do Amazon RDS de destino. Se o esquema do seu banco de dados de origem não puder ser convertido automaticamente, a AWS SCT fornecerá orientação sobre como criar um esquema equivalente no banco de dados do Amazon RDS de destino.

Para obter informações sobre como instalar a AWS SCT, consulte [Instalando, verificando e atualizando AWS SCT](#).

Para uma introdução à interface de usuário da AWS SCT, consulte [Uso da interface de usuário da AWS SCT](#).

Para obter mais informações sobre o processo de conversão, consulte [Como converter esquemas de bancos de dados usando a AWS SCT](#).

Além de converter o esquema de banco de dados existente de um mecanismo de banco de dados para outro, a AWS SCT tem alguns atributos adicionais que ajudam você a transferir seus dados e aplicativos para a nuvem da AWS:

- Você pode usar agentes de extração de dados para extrair dados de seu data warehouse para preparar a migração para o Amazon Redshift. Para gerenciar os agentes de extração de dados, você pode usar a AWS SCT. Para obter mais informações, consulte [Como migrar dados de um data warehouse on-premises para o Amazon Redshift](#).
- Você pode usar a AWS SCT para criar endpoints e tarefas do AWS DMS. Você pode executar e monitorar essas tarefas na AWS SCT. Para obter mais informações, consulte [Uso do AWS SCT com a AWS DMS](#).
- Em alguns casos, os recursos de banco de dados não podem ser convertidos em atributos do Amazon RDS ou Amazon Redshift equivalentes. O assistente do pacote de extensões da AWS SCT pode ajudá-lo a instalar funções do AWS Lambda e bibliotecas Python para emular os recursos que não podem ser convertidos. Para obter mais informações, consulte [Usando AWS SCT pacotes de extensão](#).
- Você pode usar a AWS SCT para otimizar seu banco de dados do Amazon Redshift existente. A AWS SCT recomenda chaves de classificação e chaves de distribuição para otimizar seu banco de dados. Para obter mais informações, consulte [Como otimizar o Amazon Redshift usando a AWS SCT](#).
- Você pode usar a AWS SCT para copiar o esquema de banco de dados on-premises existente para uma instância de banco de dados do Amazon RDS que esteja executando o mesmo mecanismo. Você pode usar esse recurso para analisar possíveis economias de custo ao mudar para a nuvem e ao alterar o tipo de licença.
- Você pode usar a AWS SCT para converter SQL in C++, C#, Java ou outro código de aplicativo. Você pode visualizar, analisar, editar e salvar o código SQL convertido. Para obter mais informações, consulte [Como converter o aplicativo SQL usando a AWS SCT](#).

- Você pode usar a AWS SCT para migrar processos de extração, transformação e carregamento (ETL). Para obter mais informações, consulte [Conversão de processos de extração, transformação e carregamento \(ETL\) com o AWS Schema Conversion Tool](#).

Fornecendo feedback

Você pode fornecer feedback sobre a AWS SCT. Você pode registrar um relatório de bugs, enviar uma solicitação de atributo ou fornecer informações gerais.

Para fornecer feedback sobre a AWS SCT.

1. Inicie o AWS Schema Conversion Tool.
2. Abra o menu Ajuda e escolha Deixar um comentário. A caixa de diálogo Deixar um comentário é exibida.
3. Em Área, escolha Informações, Bug report ou Solicitação de recursos.
4. Em Banco de dados de origem, escolha o banco de dados de origem. Escolha Qualquer se o seu comentário não for específico para um banco de dados particular.
5. Em Banco de dados de destino, escolha o banco de dados de destino. Escolha Qualquer se o seu comentário não for específico para um banco de dados particular.
6. Em Título, digite um título para o seu comentário.
7. Em Mensagem, digite o seu comentário.
8. Escolha Enviar para enviar seu comentário.

Instalando, verificando e atualizando AWS SCT

O AWS Schema Conversion Tool (AWS SCT) é um aplicativo independente que fornece uma interface de usuário baseada em projetos. AWS SCT está disponível para Microsoft Windows, Fedora Linux e Ubuntu Linux. AWS SCT é suportado somente em sistemas operacionais de 64 bits.

Para garantir que você obtenha a versão correta do arquivo de AWS SCT distribuição, fornecemos etapas de verificação após o download do arquivo compactado. Você pode verificar o arquivo usando as etapas fornecidas.

AWS SCT está disponível como um aplicativo independente e uma ferramenta de linha de comando. Para obter informações sobre a ferramenta de linha de comando, consulte [AWS SCT CLI](#).

Tópicos

- [Instalando AWS SCT](#)
- [Verificando o download do AWS SCT arquivo](#)
- [Baixar os drivers de banco de dados necessários](#)
- [Atualizando AWS SCT](#)
- [AWS SCT CLI](#)

Instalando AWS SCT

Você pode instalar AWS SCT nos seguintes sistemas operacionais:

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

Para instalar AWS SCT

1. Faça o download do arquivo compactado que contém o AWS SCT instalador usando o link do seu sistema operacional. Todos os arquivos compactados têm uma extensão .zip. Ao extrair o arquivo AWS SCT do instalador, ele estará no formato apropriado para o seu sistema operacional.

- [Microsoft Windows](#)

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

2. Extraia o arquivo do AWS SCT instalador do seu sistema operacional, mostrado a seguir.

Sistema operacional	Nome do arquivo
Fedora Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm
Microsoft Windows	AWS Schema Conversion Tool-1.0. <i>build-number</i> .msi
Ubuntu Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .deb

3. Execute o arquivo do AWS SCT instalador extraído na etapa anterior. Use as instruções para o seu sistema operacional, mostradas a seguir.

Sistema operacional	Instruções de instalação
Fedora Linux	<p>Execute o seguinte comando na pasta em que você obteve o arquivo por download:</p> <pre>sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm</pre>
Microsoft Windows	Clique duas vezes no arquivo para executar o instalador.
Ubuntu Linux	<p>Execute o seguinte comando na pasta em que você obteve o arquivo por download:</p> <pre>sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i> .deb</pre>

4. Instale os drivers do Java Database Connectivity (JDBC) para seus mecanismos de banco de dados de origem e de destino. Para obter instruções e links de download, consulte [Baixar os drivers de banco de dados necessários](#).

Agora, você concluiu a configuração do AWS SCT aplicativo. Clique duas vezes no ícone do aplicativo para executar AWS SCT.

Verificando o download do AWS SCT arquivo

Há várias maneiras de verificar o arquivo de distribuição do AWS SCT. A forma mais simples é comparar a soma de verificação do arquivo com a soma de verificação publicada da AWS. Como um nível adicional de segurança, é possível usar os procedimentos abaixo para verificar o arquivo de distribuição com base no sistema operacional no qual o arquivo foi instalado.

Esta seção inclui os seguintes tópicos.

Tópicos

- [Verificando a soma de verificação do arquivo AWS SCT](#)
- [Verificando os arquivos AWS SCT RPM no Fedora](#)
- [Verificando os arquivos AWS SCT DEB no Ubuntu](#)
- [Verificando o arquivo AWS SCT MSI no Microsoft Windows](#)

Verificando a soma de verificação do arquivo AWS SCT

Para detectar quaisquer erros que possam ter sido introduzidos ao baixar ou armazenar o arquivo AWS SCT compactado, você pode comparar a soma de verificação do arquivo com um valor fornecido por AWS. AWS usa o algoritmo SHA256 para a soma de verificação.

Para verificar o arquivo AWS SCT de distribuição usando uma soma de verificação

1. Faça o download do arquivo de AWS SCT distribuição usando os links na seção Instalação. Para ter mais informações, consulte [Instalando AWS SCT](#).
2. Faça download do arquivo de soma de verificação mais recente, chamado [sha256Check.txt](#). Esse arquivo inclui as somas de verificação da versão mais recente AWS SCT. Por exemplo, o arquivo pode ter a seguinte aparência:

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

3. Execute o comando de validação SHA256 para seu sistema operacional no diretório que contém o arquivo de distribuição. Por exemplo, execute o comando a seguir no Linux:

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

4. Compare os resultados do comando com o valor mostrado no arquivo sha256Check.txt. Se as somas de verificação corresponderem, então é seguro executar o arquivo de distribuição. Se as somas de verificação não corresponderem, não execute o arquivo de distribuição e [entre em contato com o AWS Support](#).

Verificando os arquivos AWS SCT RPM no Fedora

AWS fornece outro nível de validação além da soma de verificação do arquivo de distribuição. Todos os arquivos RPM no arquivo de distribuição são assinados por uma chave AWS privada. A chave pública GPG pode ser visualizada em [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).

Para verificar os arquivos AWS SCT RPM no Fedora

1. Faça o download do arquivo de AWS SCT distribuição usando os links na seção Instalação.
2. Verifique a soma de verificação do arquivo de AWS SCT distribuição.
3. Extraia os conteúdos do arquivo de distribuição. Localize o arquivo RPM que você deseja verificar.
4. Faça download da chave pública GPG em [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key)
5. Importe a chave pública para o seu banco de dados de RPM (certifique-se de que você tenha as permissões apropriadas) usando o seguinte comando:

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

6. Verifique se a importação foi bem-sucedida executando o seguinte comando:

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

7. Verifique a assinatura do RPM executando o seguinte comando:

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


Verificando os arquivos AWS SCT DEB no Ubuntu

AWS fornece outro nível de validação além da soma de verificação do arquivo de distribuição. Todos os arquivos DEB no arquivo de distribuição são assinados por uma assinatura separada do GPG.

Para verificar os arquivos AWS SCT DEB no Ubuntu

1. Faça o download do arquivo de AWS SCT distribuição usando os links na seção Instalação.
2. Verificando a soma de verificação do arquivo de AWS SCT distribuição.
3. Extraia os conteúdos do arquivo de distribuição. Localize o arquivo DEB que você deseja verificar.
4. Baixe a assinatura separada de [aws-schema-conversion-tool-1.0.latest.deb.asc](https://aws-schemaconversiontool.s3.amazonaws.com/aws-schema-conversion-tool-1.0.latest.deb.asc).
5. Faça download da chave pública GPG em [amazon.com.public.gpg-key](https://aws-schemaconversiontool.s3.amazonaws.com/aws-schema-conversion-tool-1.0.latest.gpg-key).
6. Importe a chave pública do GPG executando o seguinte comando:

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

7. Verifique a assinatura executando o seguinte comando:

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

Verificando o arquivo AWS SCT MSI no Microsoft Windows

AWS fornece outro nível de validação além da soma de verificação do arquivo de distribuição. O arquivo MSI tem uma assinatura digital que você pode verificar para garantir que ele tenha sido assinado AWS.

Para verificar o arquivo AWS SCT MSI no Windows

1. Faça o download do arquivo de AWS SCT distribuição usando os links na seção Instalação.
2. Verificando a soma de verificação do arquivo de AWS SCT distribuição.
3. Extraia os conteúdos do arquivo de distribuição. Localize o arquivo MSI que você deseja verificar.
4. No Windows Explorer, clique com o botão direito do mouse no arquivo MSI e selecione Propriedades.

5. Escolha a guia Assinaturas digitais.
6. Verifique se a assinatura digital é da Amazon Services LLC.

Baixar os drivers de banco de dados necessários

AWS SCT Para funcionar corretamente, baixe os drivers JDBC para seus mecanismos de banco de dados de origem e destino. Se você usa uma plataforma de banco de dados de destino virtual, não precisa baixar o driver JDBC para seu mecanismo de banco de dados de destino. Para ter mais informações, consulte [Usar alvos virtuais](#).

Depois de fazer download dos drivers, você atribui o local dos arquivos do driver. Para ter mais informações, consulte [Armazenamento de caminhos do driver nas configurações globais](#).

Você pode fazer download dos drivers de banco de dados dos locais a seguir.

Important

Instale a versão mais recente do driver disponível. A tabela a seguir inclui a versão mais baixa do driver de banco de dados suportada pelo AWS SCT.

Mecanismo do banco de dados	Drivers	Local de download
Amazon Aurora Edição Compatível com MySQL	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/connector/
Amazon Aurora Edição Compatível com PostgreSQL	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar

Mecanismo do banco de dados	Drivers	Local de download
Amazon EMR	HiveJDBC42.jar	http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/
Amazon Redshift	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift sem 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
Banco de dados do SQL Azure	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?visão=sql-server-ver-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?visão=sql-server-ver-15#72
Banco de dados 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-jdbc-driver-versions-and-downloads-db2-zos
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

Mecanismo do banco de dados	Drivers	Local de download
Microsoft SQL Server	mssql-jdbc-10.2.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?visualização= 15 sql-server-ver
MySQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/connector/j/
Netezza	nzjdbc.jar Use o software de ferramentas de cliente. Instale a versão do driver 7.2.1, que é compatível retroativamente com a versão de data warehouse 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 Drivers versão 8 e posterior são compatíveis.	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	O driver JConnect JDBC
Snowflake	snowflake-jdbc-3.9.2.jar Para obter mais informações, consulte Baixar/ Integrar o driver JDBC .	https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar

Mecanismo do banco de dados	Drivers	Local de download
Teradata	terajdbc4.jar tdgssconfig.jar Para o driver Teradata JDBC versão 16.20.00.11 e superior, você não precisa do arquivo tdgssconfig.jar .	https://downloads.teradata.com/download/connectivity/jdbc-driver
Vertica	vertica-jdbc-9.1.1-0.jar Drivers versão 7.2.0 e posterior são compatíveis.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Instalação de drivers JDBC no Linux

Você pode usar as etapas a seguir para instalar os drivers JDBC em seu sistema Linux para uso com AWS SCT

Para instalar os drivers JDBC no seu sistema Linux

1. Crie um diretório para armazenar os drivers JDBC.

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

2. Instale o driver JDBC para o seu mecanismo de banco de dados usando os comandos mostrados a seguir.

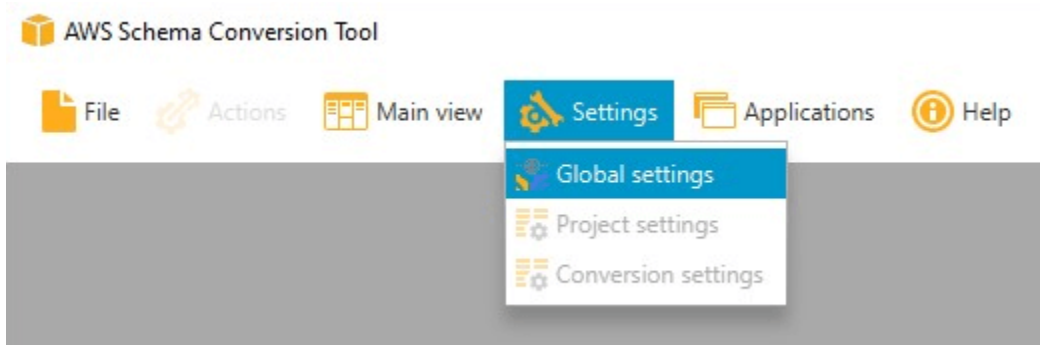
Mecanismo do banco de dados	Comandos de instalação
Amazon Aurora (compatível com MySQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Amazon Aurora (compatível com PostgreSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>
Microsoft SQL Server	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
MySQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Oracle	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo mkdir oracle-jdbc PROMPT> cd oracle-jdbc PROMPT> sudo cp -a /tmp/ojdbc8.jar .</pre>
PostgreSQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

Armazenamento de caminhos do driver nas configurações globais

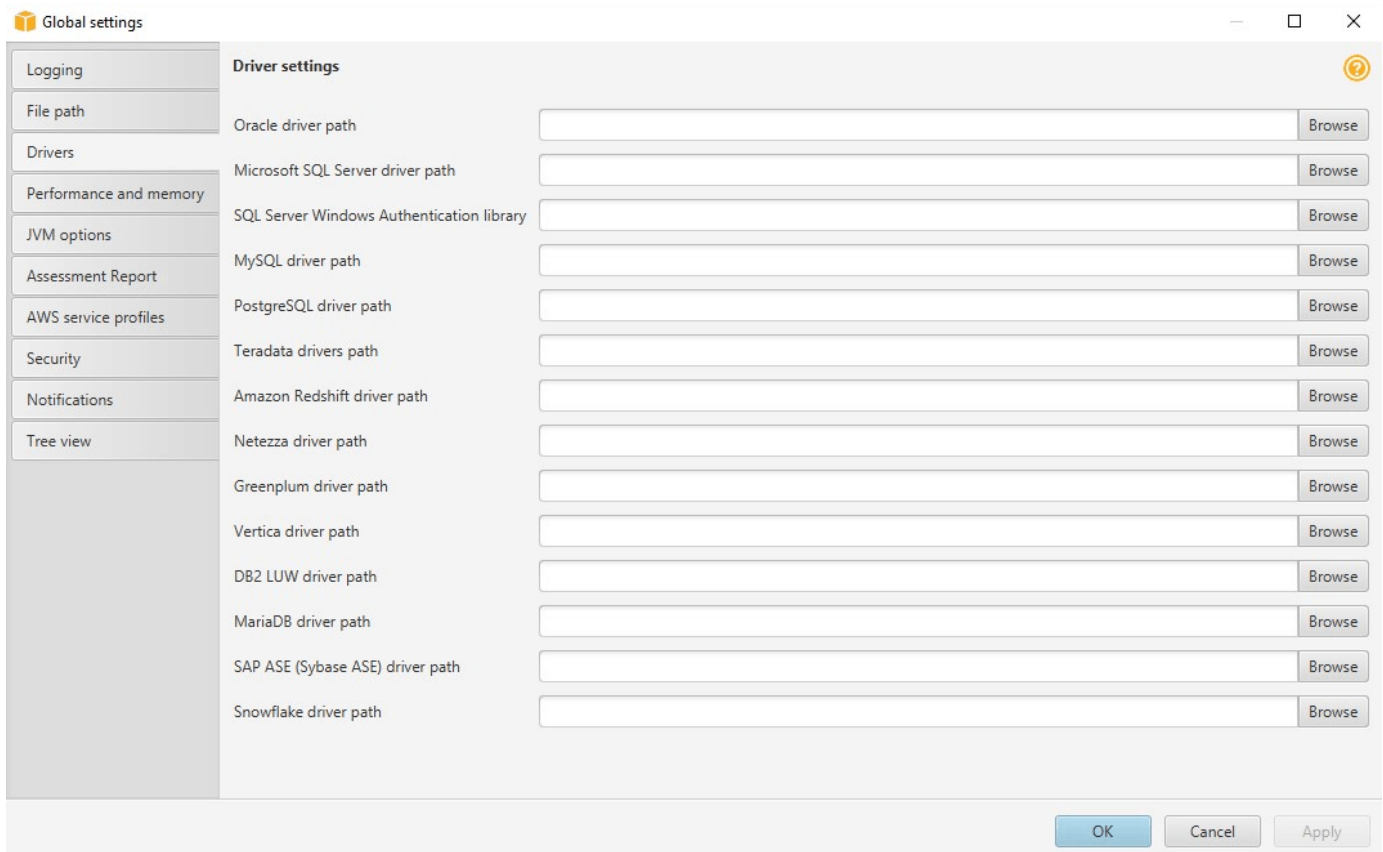
Depois de baixar e instalar os drivers JDBC necessários, você pode definir a localização dos drivers globalmente nas AWS SCT configurações. Se você não definir a localização dos drivers globalmente, o aplicativo solicitará a localização dos drivers quando você se conectar a um banco de dados.

Para atualizar os locais de arquivo do driver

1. Em AWS SCT, escolha Configurações e, em seguida, escolha Configurações globais.



2. Em Configurações globais, escolha Drivers. Adicione o caminho do arquivo para o driver JDBC do mecanismo de banco de dados de origem e do mecanismo de banco de dados da instância de banco de dados Amazon RDS de destino.



3. Ao terminar de adicionar os caminhos do driver, escolha OK.

Atualizando AWS SCT

AWS atualiza periodicamente AWS SCT com novos recursos e funcionalidades. Se você estiver atualizando a partir de uma versão anterior, crie um novo AWS SCT projeto e reconverta todos os objetos de banco de dados que estiver usando.

Você pode verificar se existem atualizações para AWS SCT.

Para verificar se há atualizações para AWS SCT

1. Ao entrar AWS SCT, escolha Ajuda e, em seguida, escolha Verificar atualizações.
2. Na caixa de diálogo Verificar atualizações, selecione Novidades. Se o link não for exibido, você já tem a versão mais recente.

AWS SCT CLI

Você pode baixar a AWS SCT CLI para uso na linha de comando. Para baixar o JAR, use o seguinte link:

[AWSSchemaConversionToolBatch.jar](#)

Uso da interface de usuário da AWS SCT

Use os tópicos a seguir para ajudá-lo a trabalhar com a interface de usuário da AWS SCT. Para mais informações sobre como instalar a AWS SCT, consulte [Instalando, verificando e atualizando AWS SCT](#).

Tópicos

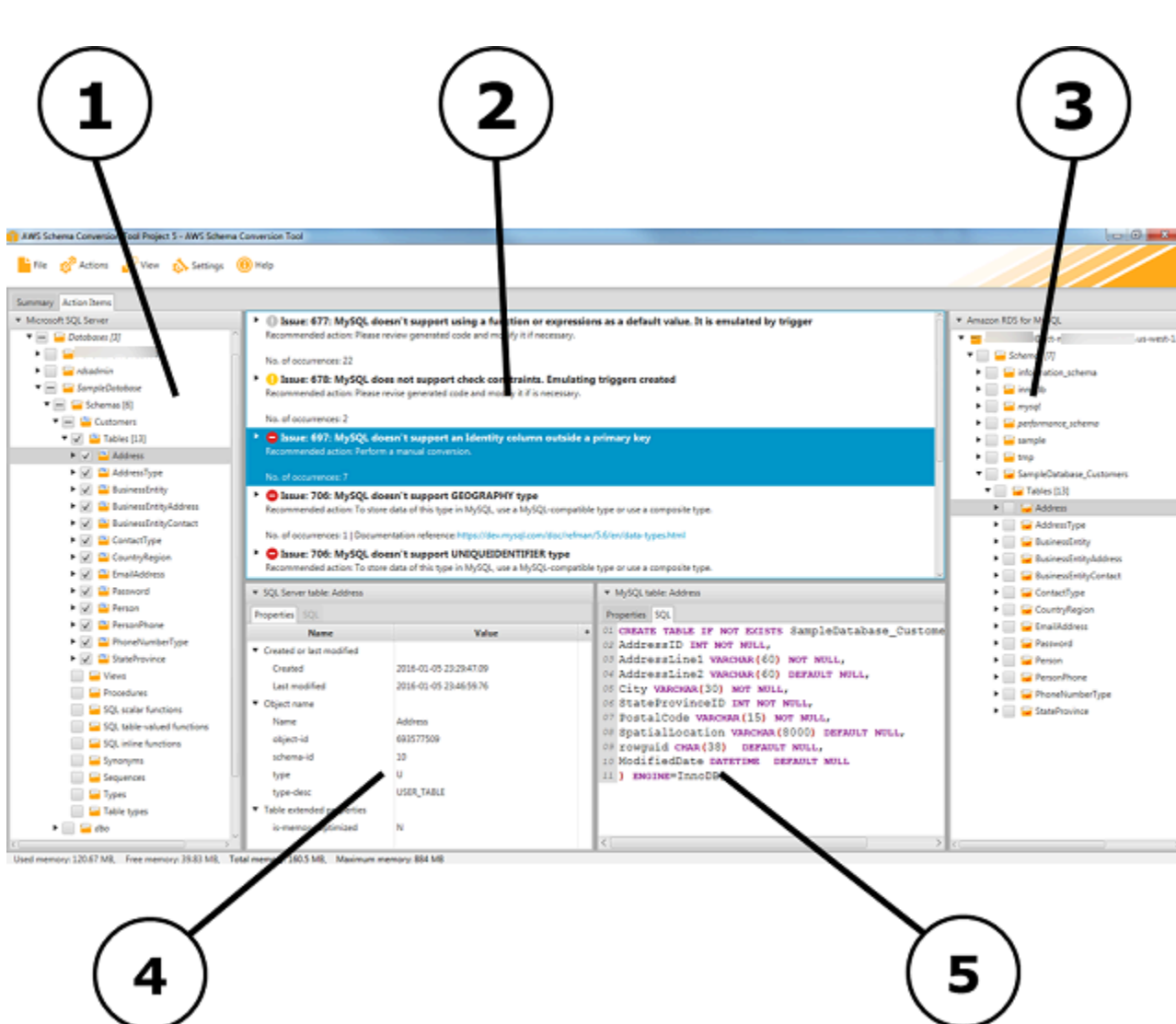
- [A janela do projeto da AWS SCT](#)
- [Iniciar o AWS SCT](#)
- [Criando um projeto da AWS SCT](#)
- [Usando um novo assistente de projeto na AWS SCT](#)
- [Salvando e abrindo um projeto da AWS SCT](#)
- [Adicionando servidores de banco de dados a um projeto da AWS SCT](#)
- [Executando a AWS SCT em um modo off-line](#)
- [Usando os filtros de árvore da AWS SCT](#)
- [Ocultando esquemas na visualização em árvore da AWS SCT](#)
- [Criando e revisando o relatório de avaliação de migração do banco de dados](#)
- [Como converter seu esquema](#)
- [Aplicando o esquema convertido à instância de banco de dados de destino](#)
- [Armazenando perfis de serviço da AWS na AWS SCT](#)
- [Usar o AWS Secrets Manager](#)
- [Armazenando senhas de banco de dados](#)
- [Usando a visualização UNION ALL para projetos com tabelas particionadas](#)
- [Atalhos de teclado da AWS SCT](#)

A janela do projeto da AWS SCT

A ilustração a seguir é o que você vê na AWS SCT ao criar um projeto de migração de esquema e, em seguida, converter um esquema.

1. No painel esquerdo, o esquema do banco de dados de origem é apresentado em uma visualização em árvore. Seu esquema de banco de dados é de "carregamento preguiçoso." Em

- outras palavras, quando você seleciona um item na visualização em árvore, a AWS SCT obtém e exibe o esquema atual do seu banco de dados de origem.
- No painel superior central, os itens de ação aparecem para elementos do esquema do mecanismo de banco de dados de origem que não puderam ser convertidos automaticamente para o mecanismo de banco de dados de destino.
 - No painel direito, o esquema da instância de banco de dados de destino é apresentado em uma visualização em árvore. Seu esquema de banco de dados é de "carregamento preguiçoso." Ou seja, nesse ponto, quando você seleciona um item na visualização em árvore, a AWS SCT obtém e exibe o esquema atual do seu banco de dados de destino.



4. No painel inferior esquerdo, quando você escolhe um elemento do esquema, as propriedades são exibidas. Elas descrevem o elemento do esquema de origem e o comando SQL para criar esse elemento no banco de dados de origem.
5. No painel inferior direito, quando você escolhe um elemento do esquema, as propriedades são exibidas. Elas descrevem o elemento do esquema de destino e o comando SQL para criar esse elemento no banco de dados de destino. Você pode editar esse comando SQL e salvar o comando atualizado com o projeto.

Iniciar o AWS SCT

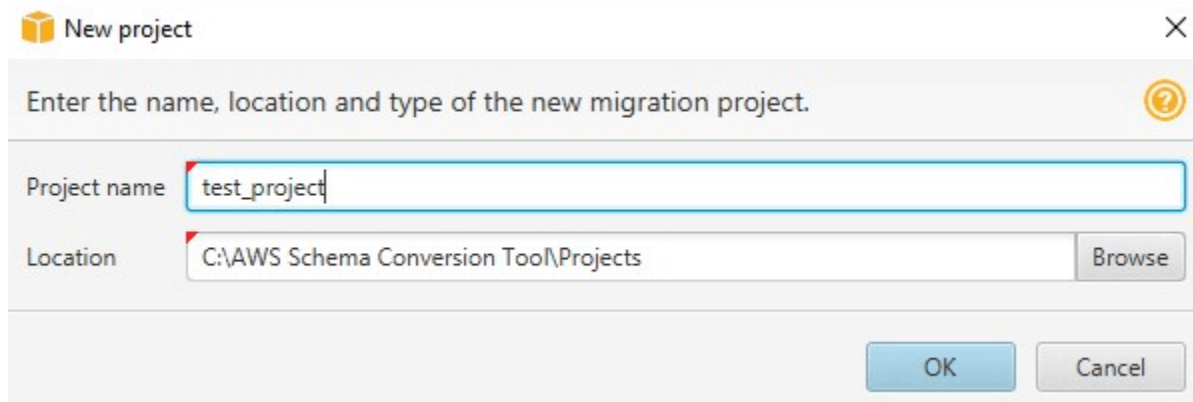
Para iniciar a AWS Schema Conversion Tool, clique duas vezes no ícone do aplicativo.

Criando um projeto da AWS SCT

Use o procedimento a seguir para criar um projeto da AWS Schema Conversion Tool.

Para criar seu projeto

1. Inicie o AWS Schema Conversion Tool.
2. No menu Arquivo, selecione Novo projeto. A caixa de diálogo Novo projeto é exibida.



3. Digite um nome para o projeto, que é armazenado localmente no computador.
4. Digite o local do arquivo do projeto local.
5. Selecione OK para criar o projeto da AWS SCT.
6. Selecione Adicionar origem para adicionar um novo banco de dados de origem ao seu projeto da AWS SCT. Você pode adicionar vários bancos de dados de origem ao seu projeto da AWS SCT.

7. Selecione Adicionar destino para adicionar uma nova plataforma de destino ao seu projeto da AWS SCT. Você pode adicionar várias plataformas de destino ao seu projeto da AWS SCT.
8. Escolha o esquema do banco de dados de origem no painel esquerdo.
9. No painel direito, especifique a plataforma de banco de dados de destino para o esquema de origem selecionado.
10. Selecione Criar mapeamento. Esse botão fica ativo depois que você escolhe o esquema do banco de dados de origem e a plataforma do banco de dados de destino. Para obter mais informações, consulte [Criação de regras de mapeamento](#).

Agora, seu projeto da AWS SCT está configurado. Você pode salvar seu projeto, criar um relatório de avaliação de migração de banco de dados e converter seus esquemas de banco de dados de origem.

Usando um novo assistente de projeto na AWS SCT

Você pode criar um novo projeto de migração de banco de dados usando o novo assistente de projeto. Esse assistente ajuda você a determinar seu destino de migração e a se conectar aos seus bancos de dados. Ele estima a complexidade de uma migração para todos os destinos compatíveis. Depois de executar o assistente, a AWS SCT produz um relatório resumido para a migração do seu banco de dados para diferentes destinos. Você pode usar esse relatório para comparar possíveis destinos e escolher o caminho de migração ideal.

Como executar o novo assistente de projeto

1. Selecione o banco de dados de origem.
 - a. Inicie o AWS Schema Conversion Tool.
 - b. No menu Arquivo, selecione Novo assistente de projeto. A caixa de diálogo Criar um novo projeto de migração de banco de dados é aberta.
 - c. Para inserir as informações de conexão do banco de dados de origem, use as instruções a seguir:

Parâmetro	Ação
Nome do projeto	Digite um nome para o projeto, que é armazenado localmente no computador.

Parâmetro	Ação
Local	Digite o local do arquivo do projeto local.
Source type (Tipo de origem)	<p>Escolha uma das seguintes opções: banco de dados SQL, banco de dados NoSQL ou ETL.</p> <p>Se você quiser ver o relatório resumido que inclui todos os destinos de migração, escolha Banco de dados SQL.</p>
Source engine	Escolha o mecanismo de banco de dados de origem.
Estratégia de migração	<p>Escolha uma das seguintes opções:</p> <ul style="list-style-type: none"> • Quero trocar de mecanismo e otimizar para a nuvem – Essa opção converte seu banco de dados de origem em um novo mecanismo de banco de dados. • Quero manter o mesmo mecanismo, mas otimizar para a nuvem – Essa opção mantém o mecanismo de banco de dados como está e move o banco de dados on-premises para a nuvem. • Quero ver um relatório combinado sobre troca e otimização do mecanismo de banco de dados para a nuvem – Essa opção compara a complexidade da migração de todas as opções de migração disponíveis. <p>Se você quiser ver o relatório de avaliação agregado que inclui todos os destinos de migração, escolha a última opção.</p>

d. Escolha Next (Próximo). A página Conectar-se ao banco de dados de origem é aberta.

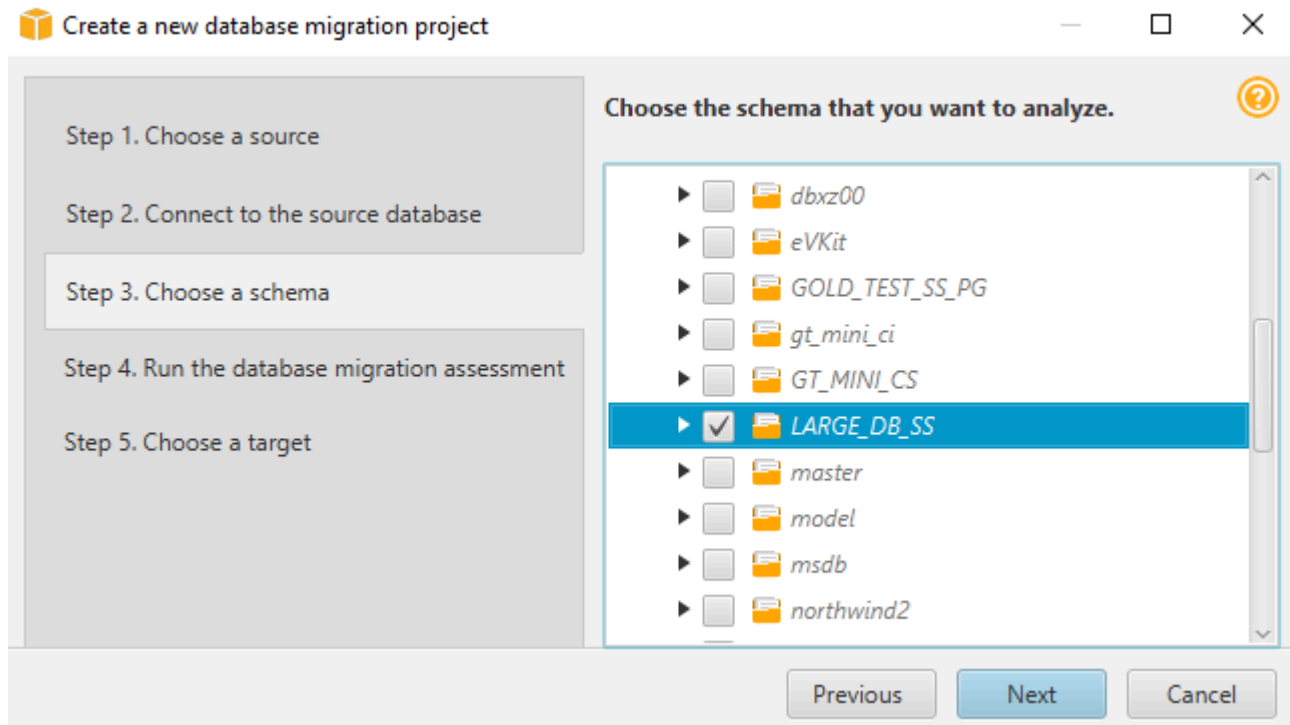
2. Conecte-se ao seu banco de dados de origem.

a. Forneça as informações de conexão do banco de dados de origem. Os parâmetros de conexão dependem do mecanismo do banco de dados de origem. Certifique-se de que o usuário que você usa para a análise do seu banco de dados de origem tenha as permissões aplicáveis. Para obter mais informações, consulte [Origens para a AWS SCT](#).

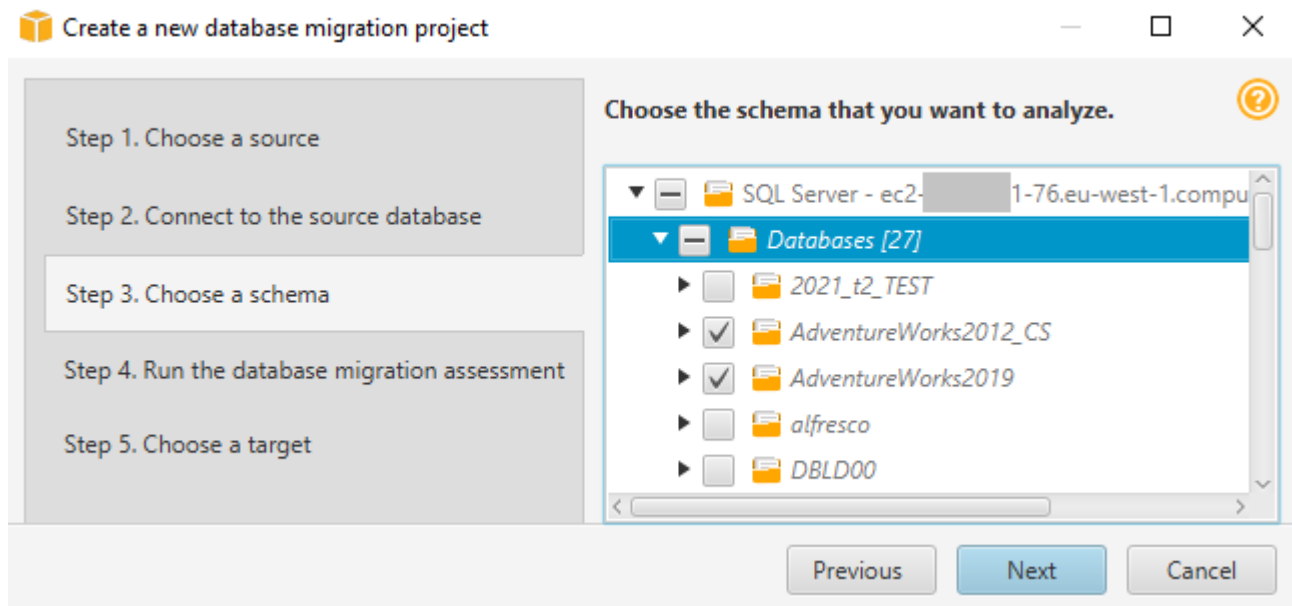
b. Escolha Next (Próximo). A página Escolher um esquema é aberta.

3. Escolha o esquema do banco de dados.

- a. Selecione a caixa de seleção para o nome dos esquemas que você deseja avaliar e, em seguida, escolha o esquema em si. O nome do esquema é destacado em azul quando selecionado e o botão Avançar está disponível.



- b. Se você quiser avaliar vários esquemas de banco de dados, marque as caixas de seleção de todos os esquemas e escolha o nó pai. Para uma avaliação bem-sucedida, você deve escolher o nó pai. Por exemplo, para um banco de dados SQL Server de origem, escolha o nó Bancos de dados. O nome do nó pai é destacado em azul e o botão Avançar está disponível.



- c. Escolha Avançar. A AWS SCT analisa os esquemas do banco de dados de origem e cria um relatório de avaliação da migração do banco de dados. O número de objetos do banco de dados nos esquemas do banco de dados de origem afeta o tempo necessário para executar a avaliação. Quando concluída, a página Executar a avaliação da migração do banco de dados é aberta.
4. Execute a avaliação de migração do banco de dados.
 - a. Você pode revisar e comparar os relatórios de avaliação para diferentes destinos de migração ou salvar uma cópia local dos arquivos do relatório de avaliação para análise posterior.
 - b. Salve uma cópia local do relatório de avaliação da migração do banco de dados. Escolha Salvar e, em seguida, insira o caminho para a pasta para salvar os arquivos e escolha Salvar. A AWS SCT salva os arquivos do relatório de avaliação na pasta especificada.
 - c. Escolha Next (Próximo). A página Escolha um destino é aberta.
 5. Escolha seu banco de dados de destino.
 - a. Para Mecanismo de destino, escolha o mecanismo de banco de dados de destino que você decide usar com base no relatório de avaliação.
 - b. Forneça as suas informações de conexão para o banco de dados de destino. Os parâmetros de conexão que você vê dependem do mecanismo de banco de dados de destino selecionado. Certifique-se de que o usuário especificado para o banco de dados de destino tenha as permissões necessárias. Para obter mais informações sobre as permissões necessárias, consulte as seções que descrevem as permissões para bancos de dados de destino em [Origens para a AWS SCT](#) e [Permissões para o Amazon Redshift como destino](#).

- c. Escolha Concluir. A AWS SCT cria seu projeto e adiciona as regras de mapeamento. Para obter mais informações, consulte [Criação de regras de mapeamento](#).

Agora você pode usar o projeto da AWS SCT para converter seus objetos do banco de dados de origem.

Salvando e abrindo um projeto da AWS SCT

Use o procedimento a seguir para salvar um projeto da AWS Schema Conversion Tool.

Para salvar seu projeto

1. Inicie o AWS Schema Conversion Tool.
2. No menu Arquivo, selecione Salvar projeto.

A AWS SCT salva o projeto na pasta que você especificou ao criar o projeto.

Use o procedimento a seguir para abrir um projeto da AWS Schema Conversion Tool existente.

Para abrir seu projeto

1. No menu Arquivo, escolha Abrir projeto. A caixa de diálogo Abrir é exibida.
2. Escolha a pasta do projeto e, em seguida, escolha o arquivo Windows Script Component (*.sct).
3. A AWS SCT abre o projeto, mas não se conecta automaticamente aos bancos de dados de origem e de destino. Escolha Conectar-se ao servidor na parte superior das árvores do esquema do banco de dados para se conectar aos bancos de dados de origem e de destino.

Se você abrir um projeto salvo na versão 1.0.655 ou anterior da AWS SCT, a AWS SCT criará automaticamente regras de mapeamento para todos os esquemas do banco de dados de origem na plataforma do banco de dados de destino. Para adicionar outras plataformas de banco de dados de destino, exclua as regras de mapeamento existentes e crie novas. Para mais informações sobre as regras de mapeamento, consulte [Criação de regras de mapeamento](#).

Adicionando servidores de banco de dados a um projeto da AWS SCT

Você pode adicionar vários servidores de banco de dados de origem e destino a um projeto da AWS Schema Conversion Tool.

Para adicionar um servidor ao projeto

1. Inicie o AWS Schema Conversion Tool.
2. Crie um novo projeto ou abra um projeto existente.
3. Escolha Adicionar origem no menu para adicionar um novo banco de dados de origem.
4. Escolha uma plataforma de banco de dados e especifique as credenciais de conexão do banco de dados. Para mais informações sobre a conexão com um banco de dados de origem, consulte [Origens para a AWS SCT](#).

Use o procedimento a seguir para conectar-se ao seu banco de dados.

Para se conectar ao seu banco de dados

1. Abra o menu de contexto (clique com o botão direito do mouse) de um servidor de banco de dados e escolha Estabelecer conexão.

Você também pode escolher Conectar-se ao servidor na parte superior da árvore do esquema do banco de dados.

2. Digite a senha para se conectar ao servidor de banco de dados de origem.
3. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
4. Escolha Conectar para se conectar ao banco de dados de origem.

Use o procedimento a seguir para remover um servidor de banco de dados do projeto da AWS SCT.

Para remover um servidor de banco de dados

1. Escolha o servidor de banco de dados a ser removido.
2. Abra o menu de contexto (clique com o botão direito do mouse) e, em seguida, escolha Remover do projeto.

A AWS SCT remove o servidor de banco de dados selecionado, todas as regras de mapeamento, resultados de conversão e outros metadados relacionados a esse servidor.

Executando a AWS SCT em um modo off-line

Você pode executar a AWS Schema Conversion Tool em um modo off-line. A seguir, você pode aprender como trabalhar com um projeto da AWS SCT existente quando desconectado do banco de dados de origem.

A AWS SCT não requer uma conexão com seu banco de dados de origem para executar as seguintes operações:

- Adicionar regras de mapeamento.
- Criar relatórios de avaliação de migração do banco de dados
- Converter esquemas e códigos de banco de dados.
- Editar o código-fonte e o código convertido.
- Salve o código-fonte e o código convertido como scripts SQL em um arquivo de texto.

Antes de usar a AWS SCT no modo off-line, conecte-se ao seu banco de dados de origem, carregue metadados e salve seu projeto. Abra esse projeto ou desconecte-se do servidor de banco de dados de origem para usar a AWS SCT no modo off-line.

Para executar a AWS SCT em um modo off-line

1. Inicie a AWS Schema Conversion Tool e crie um novo projeto. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#).
2. Adicione um servidor de banco de dados de origem e conecte-se ao seu banco de dados de origem. Para obter mais informações, consulte [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).
3. Adicione um servidor de banco de dados de destino ou use uma plataforma virtual de banco de dados de destino. Para obter mais informações, consulte [Usar alvos virtuais](#).
4. Crie uma regra de mapeamento para definir a plataforma de banco de dados de destino para seu banco de dados de origem. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).
5. Escolha Exibir e, em seguida, Visualização principal.

6. No painel esquerdo que exibe os objetos do banco de dados de origem, escolha os esquemas do banco de dados de origem. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e, em seguida, escolha Carregar esquema. Essa operação carrega todos os metadados do esquema de origem em seu projeto da AWS SCT.

As operações Criar relatório e Converter esquema também carregam todos os metadados do esquema de origem em seu projeto da AWS SCT. Se você executou uma dessas operações no menu de contexto, ignore a operação Carregar esquema.

7. No menu Arquivo, escolha Salvar projeto para salvar os metadados do banco de dados de origem em seu projeto.
8. Escolha Desconectar-se do servidor para se desconectar do banco de dados de origem. Agora você pode usar a AWS SCT no modo off-line.

Usando os filtros de árvore da AWS SCT

Para migrar dados de uma origem para um destino, a AWS SCT carrega todos os metadados de bancos de dados de origem e destino em uma estrutura de árvore. Essa estrutura aparece na AWS SCT como a visualização em árvore na janela principal do projeto.

Alguns bancos de dados podem ter um grande número de objetos na estrutura de árvore. Você pode usar filtros de árvore na AWS SCT para pesquisar objetos nas estruturas de árvore de origem e de destino. Ao usar um filtro de árvore, você não altera os objetos que são convertidos quando o banco de dados é convertido. O filtro muda apenas o que você vê na árvore.

Os filtros de árvore trabalham com objetos que a AWS SCT pré-carregou. Em outras palavras, a AWS SCT não carrega objetos do banco de dados durante pesquisas. Essa abordagem significa que a estrutura de árvore geralmente contém um número menor de objetos do que os presentes no banco de dados.

Para os filtros de árvore, lembre-se do seguinte:

- O filtro padrão é QUALQUER, o que significa que o filtro usa uma pesquisa de nome para encontrar objetos.
- Quando você selecionar um ou mais tipos de objeto, verá apenas esses tipos de objetos na árvore.
- Você pode usar a máscara de filtro para mostrar diferentes tipos de símbolos, incluindo Unicode, espaços e caracteres especiais. O "%" é o caractere curinga para qualquer símbolo.
- Depois de aplicar um filtro, a contagem mostra apenas o número de objetos filtrados.

Para criar um filtro de árvore

1. Abra um projeto existente da AWS SCT.
2. Conecte-se ao banco de dados ao qual você deseja aplicar o filtro de árvore.
3. Escolha o ícone de filtro.



O ícone de desfazer filtro é desativado, pois nenhum filtro está aplicado no momento.

4. Insira as informações a seguir na caixa de diálogo Filtro. As opções na caixa de diálogo são diferentes para cada mecanismo de banco de dados.

Opções de filtro da AWS SCT	Ação
Nível	<p>Escolha Categorias para filtrar objetos por categorias.</p> <p>Escolha Status para filtrar objetos por status.</p>
Type	<p>Em Categorias em Nível, escolha as categorias dos objetos filtrados . Escolha Qualquer carregado para exibir objetos de todas as categorias.</p> <p>Em Status em Nível, escolha o status dos objetos filtrados. Você pode escolher uma das seguintes opções:</p> <ul style="list-style-type: none"> • Convertido para exibir todos os objetos convertidos • Tem ações para exibir todos os objetos que têm problemas de conversão • Criptografado para exibir todos os objetos criptografados
Condição	<p>Para Categorias em Nível, escolha a condição de filtragem entre Curtir e Não curtir.</p> <p>Para Status em Nível, a opção de condição de filtragem não está disponível.</p>

Opções de filtro da AWS SCT	Ação
Valor	<p>Para Categorias em Nível, insira o Valor para filtrar a árvore por esse valor.</p> <p>Use a porcentagem (%) como curinga para exibir todos os objetos.</p> <p>Para Status em Nível, escolha o Valor entre Verdadeiro e Falso.</p>
E/Ou	Escolha operadores lógicos AND ou OR para aplicar várias cláusulas de filtro.

Filter

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

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

Add new clause

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

Import Export Download template Reset Apply Close

- Escolha Adicionar nova cláusula para adicionar uma cláusula de filtro adicional. A AWS SCT pode aplicar várias cláusulas de filtro usando operadores lógicos AND ou OR.
- Escolha Apply (Aplicar). Depois de escolher Aplicar, o ícone desfazer filtro (ao lado do ícone de filtro) é habilitado. Use esse ícone se você deseja remover os filtros aplicados.
- Selecione Fechar para fechar a caixa de diálogo.

Ao filtrar o esquema que aparece na árvore, você não altera os objetos que são convertidos quando o esquema é convertido. O filtro muda apenas o que você vê na árvore.

Importar uma lista de arquivos para o filtro de árvore

Você pode importar um arquivo de valores separados por vírgula (CSV) com separadores de ponto e vírgula ou um arquivo JSON que contém nomes ou valores que você deseja que o filtro de árvore use. Abra um projeto da AWS SCT existente, conecte-se ao banco de dados para aplicar o filtro de árvore e, em seguida, selecione o ícone de filtro.

Para baixar um exemplo do arquivo, escolha Baixar modelo. Insira o nome do arquivo e selecione Salvar.

Para baixar suas configurações de filtro existentes, escolha Exportar. Insira o nome do arquivo e selecione Salvar.

Para importar uma lista de arquivos para o filtro de árvore, escolha Importar. Escolha um arquivo para importar e, em seguida, escolha Abrir. Escolha Aplicar e selecione Fechar.

Os arquivos CSV usam ponto e vírgula como separador e têm o seguinte formato:

- `object_type` é o tipo de objeto que você deseja localizar.
- `database_name` é o nome do banco de dados em que esse objeto existe.
- `schema_name` é o nome do esquema em que esse objeto existe.
- `object_name` é o nome do objeto.
- `import_type` especifica para `include` ou `exclude` este item do filtro.

Use arquivos JSON para descrever casos complexos de filtragem, como regras aninhadas. Arquivos JSON têm o seguinte formato:

- `filterGroupType` é o tipo de regra de filtro (operadores lógicos AND ou OR) que se aplica a várias cláusulas de filtro.
- `filterCategory` é o nível do filtro (Categorias ou Status).
- `names` é a lista de nomes de objetos que se aplica ao filtro Categorias.
- `filterCondition` é a condição de filtragem (LIKE ou NOT LIKE) que se aplica ao filtro Categorias.

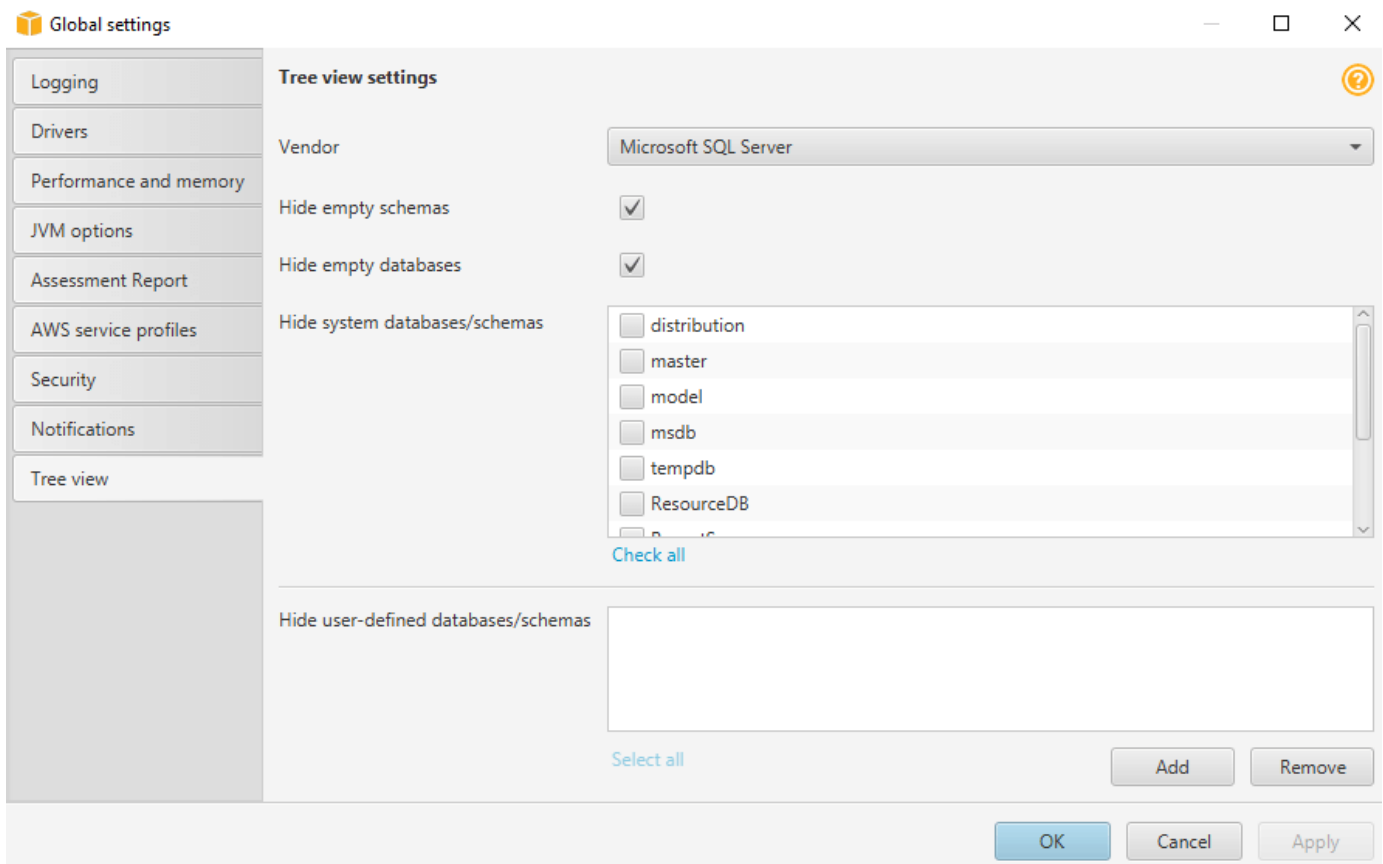
- `transformName` é o nome do status que se aplica ao filtro Status.
- `value` é o valor pelo qual filtrar a árvore.
- `transformValue` é o valor do filtro (TRUE ou FALSE) que se aplica ao filtro Status.

Ocultando esquemas na visualização em árvore da AWS SCT

Ao usar as configurações da visualização em árvore, você especifica quais esquemas e bancos de dados deseja ver na visualização em árvore da AWS SCT. Você pode ocultar esquemas vazios, bancos de dados vazios, bancos de dados de sistemas além de esquemas e bancos de dados definidos pelo usuário.

Para ocultar bancos de dados e esquemas na visualização em árvore

1. Abra um projeto da AWS SCT.
2. Conecte-se ao armazenamento de dados que você deseja mostrar na visualização em árvore.
3. Escolha Configurações, Configurações globais, Visualização em árvore.



4. Na seção Configurações de visualização em árvore, faça o seguinte:

- Para Fornecedor, escolha a plataforma de banco de dados.
- Escolha Ocultar esquemas vazios para ocultar esquemas vazios para a plataforma de banco de dados selecionada.
- Escolha Ocultar bancos de dados vazios para ocultar bancos de dados vazios para a plataforma de banco de dados selecionada.
- Em Ocultar esquemas/bancos de dados do sistema, escolha os esquemas e bancos de dados do sistema por nome para ocultá-los.
- Em Ocultar esquemas/bancos de dados definidos pelo usuário, digite os nomes dos esquemas e bancos de dados definidos pelo usuário que você deseja ocultar e, em seguida, escolha Adicionar. Os nomes não diferenciam maiúsculas de minúsculas.

5. Escolha OK.

Criando e revisando o relatório de avaliação de migração do banco de dados

O relatório de avaliação de migração do banco de dados resume todos os itens de ação dos esquemas que não podem ser convertidos automaticamente para o mecanismo da instância do banco de dados Amazon RDS de destino. O relatório também inclui estimativas do trabalho necessário para gravar o código equivalente de sua instância de banco de dados de destino.

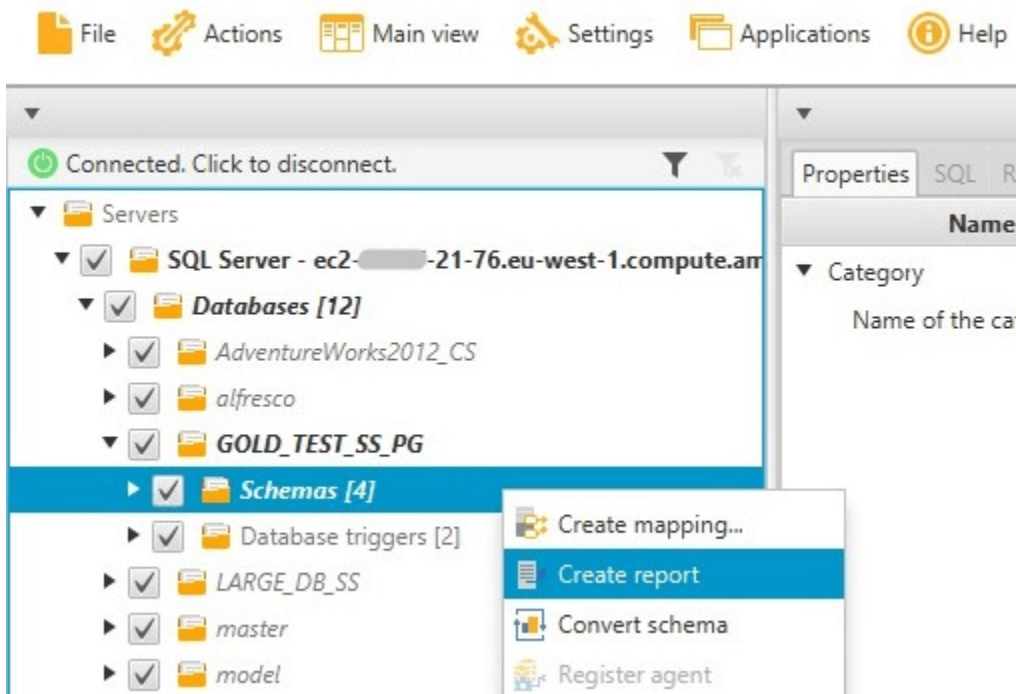
Você pode criar um relatório de avaliação da migração do banco de dados depois de adicionar os bancos de dados de origem e as plataformas de destino ao seu projeto e especificar as regras de mapeamento.

Para criar e visualizar o relatório de avaliação de migração do banco de dados

1. Certifique-se de ter criado uma regra de mapeamento para o esquema do banco de dados de origem para o qual criar um relatório de avaliação. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
2. No menu Visualizar, selecione Visualização principal.
3. No painel esquerdo que exibe o esquema do banco de dados de origem, escolha objetos de esquema para os quais criar um relatório de avaliação.

Certifique-se de ter marcado as caixas de seleção de todos os objetos do esquema para os quais criar um relatório de avaliação.

- Abra o menu de contexto (clique com o botão direito do mouse) do objeto e, em seguida, escolha Criar relatório.



A visualização do relatório de avaliação se abre.

- Selecione a guia Itens de ação.

A guia Itens de ação exibe uma lista de itens que descreve o esquema que não pode ser convertido automaticamente. Escolha um item de ação na lista. A AWS SCT destacará o item do seu esquema ao qual o item de ação se aplica, conforme mostrado a seguir.

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

Summary of Issues:

- Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
- Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes. Number of occurrences: 2
- Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary. Number of occurrences: 1. Parameter: @InputPosNo (Number of occurrences: 1)
- Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable. Number of occurrences: 1
- Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy. Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
- Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually. Number of occurrences: 3
- Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion. Number of occurrences: 1
- Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function. Number of occurrences: 12

Detailed View of Issue 811:

Source: Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK

```

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

```

Target: Amazon RDS for MySQL category: Schemas

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

6. Escolha a guia Resumo.

A guia Resumo exibe as informações de resumo do relatório de avaliação de migração do banco de dados. Ela mostra o número de itens que foram convertidos automaticamente, e o número de itens que não foram convertidos automaticamente. O resumo também inclui uma estimativa do tempo que levará para criar o esquema na instância de banco de dados de destino que é equivalente ao do banco de dados de origem.

A seção Avaliação de licença e suporte à nuvem contém informações sobre como mover seu esquema de banco de dados on-premises existente para uma instância de banco de dados do Amazon RDS que executa o mesmo mecanismo. Por exemplo, se você deseja alterar tipos de licença, esta seção do relatório mostra quais atributos do seu banco de dados atual devem ser removidos.

Um exemplo de um resumo do relatório de avaliação é mostrado a seguir.

Summary | Action items

Save to CSV Save to PDF

Database migration assessment report

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

Executive summary

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

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

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects

Object Type	Count	Automatically converted	With simple actions	With medium-complexity actions	With complex actions
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	2%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	33%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	80%	0%	0%

- Escolha a guia Resumo e depois Salvar como PDF. O relatório de avaliação de migração de banco de dados é salvo como um arquivo PDF. O arquivo PDF contém as informações do resumo e do item de ação.

Você também pode escolher Salvar como CSV para salvar o relatório como um arquivo CSV. Ao escolher essa opção, a AWS SCT cria três arquivos CSV. Esses arquivos contêm as seguintes informações:

- Uma lista de itens de ação de conversão com ações recomendadas.
- Um resumo dos itens da ação de conversão com uma estimativa do esforço necessário para converter uma ocorrência do item de ação.
- Um resumo executivo com vários itens de ação categorizados pelo tempo estimado de conversão.

Database objects with conversion actions for Amazon RDS for PostgreSQL

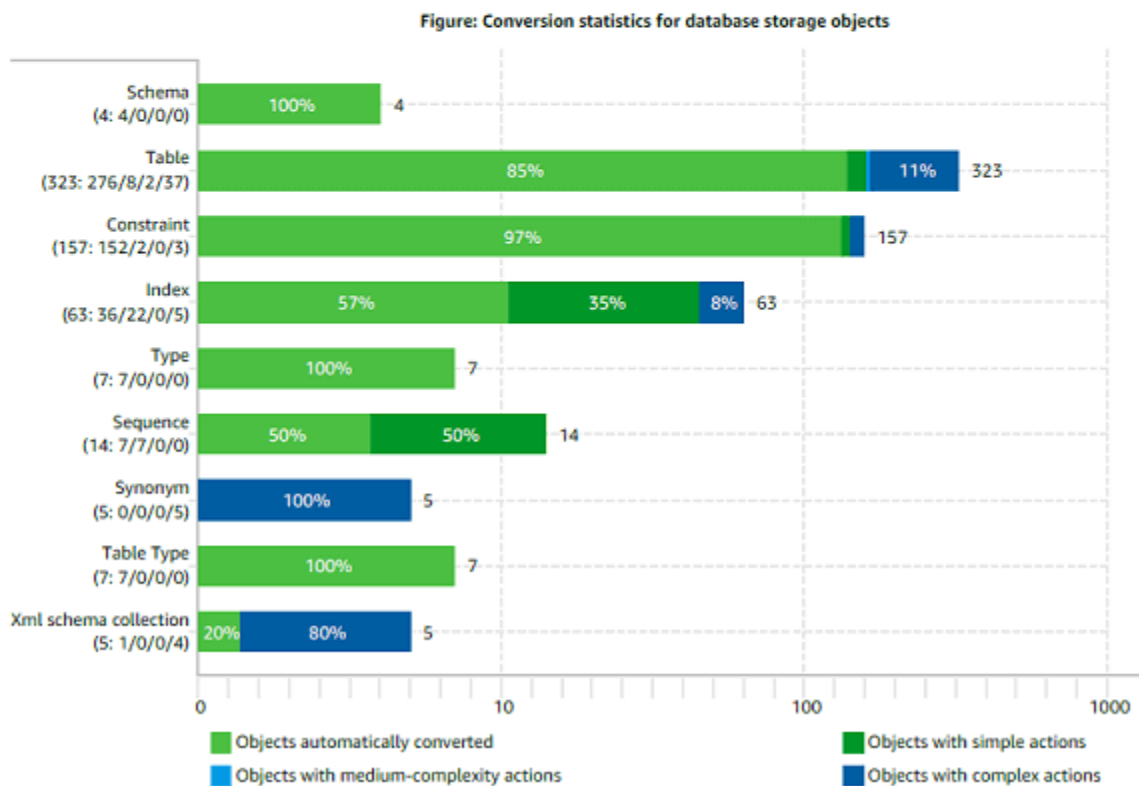
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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

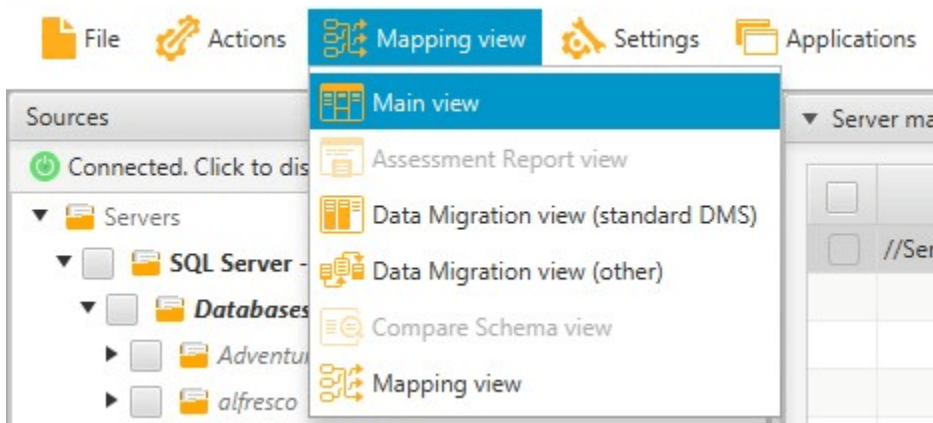


Como converter seu esquema

Depois de adicionar bancos de dados de origem e destino ao seu projeto e criar regras de mapeamento, você pode converter os esquemas do banco de dados de origem. Use o procedimento a seguir para converter o esquema.

Para converter seu esquema

1. Escolha Exibir e, em seguida, Visualização principal.



2. No painel esquerdo que exibe o esquema do banco de dados de origem, selecione a caixa de seleção para o nome do objeto para converter. Em seguida, escolha esse objeto. A AWS SCT destaca o nome do objeto em azul. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Converter esquema.

Para converter vários objetos do banco de dados, marque as caixas de seleção de todos os objetos. Em seguida, escolha o nó pai. Por exemplo, para tabelas, o nó pai é Tabelas. Certifique-se de que a AWS SCT destaca o nome do nó pai em azul. Abra o menu de contexto (clique com o botão direito do mouse) do nó pai e escolha Converter esquema.

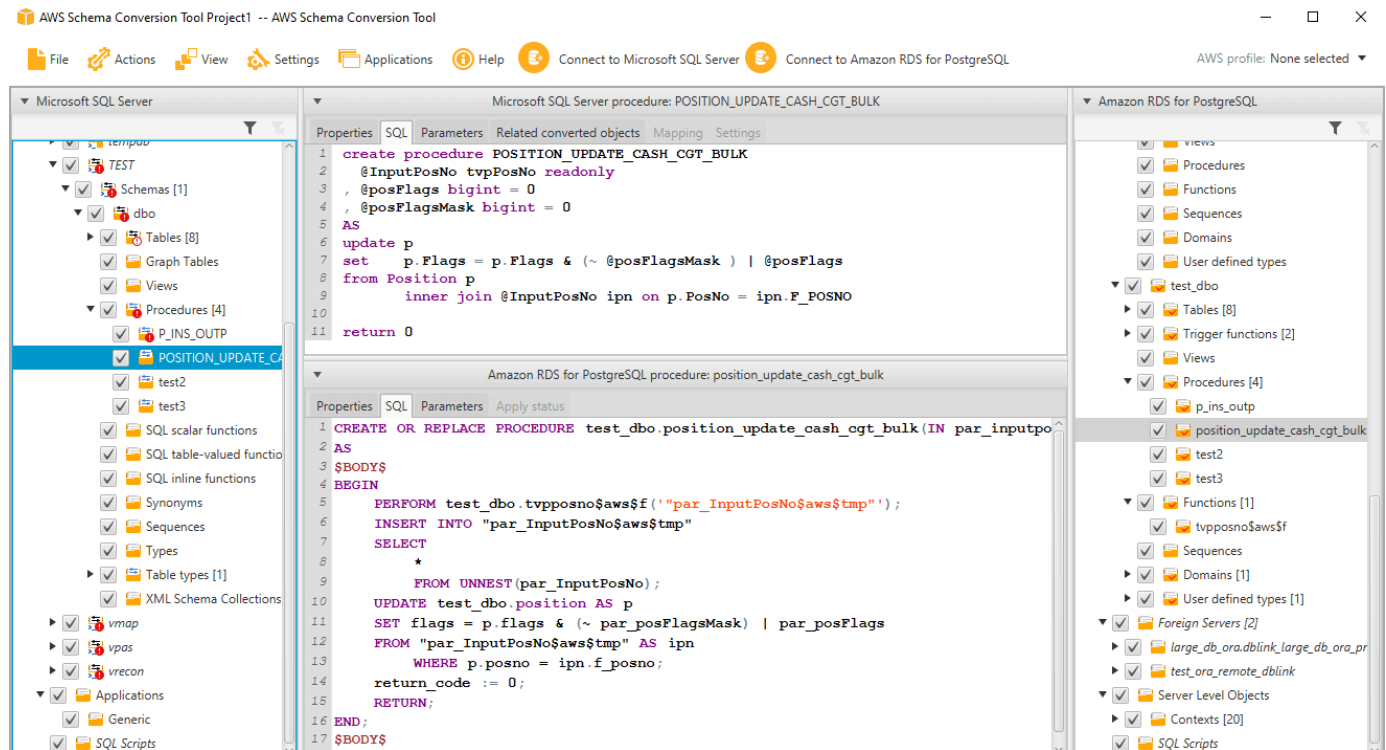
The screenshot displays the AWS Schema Conversion Tool interface. At the top, there is a menu bar with icons for File, Actions, Main view, Settings, Applications, Help, Add source, and Add target. The main workspace is divided into three panels:

- Left Panel (Server Tree):** Shows a tree view of servers. The selected server is "SQL Server - ec2-52-17-21-76.eu-west-1.compute.amazonaws.com". Underneath, there are 12 databases, with "TEST" selected. A context menu is open over "TEST", listing actions such as "Create mapping...", "Create report", "Convert schema" (highlighted), "Register agent", "Compare schema", "Load schema", "Hide schema", "Refresh from database", "Collect statistics", "Upload statistics", "Create DMS task", "Create Local & DMS task", "Create Local task", "Add virtual partitioning", and "Save as SQL".
- Top Right Panel (Properties):** Shows the properties for the selected "TEST" database. The "Name" is "TEST". The "Created or last modified" date is "2021-09-06 09:56:08.26". Other properties include "compatibility-level" (100) and "collation-name" (SQL_Latin1_General_CP1_CI_AS).
- Bottom Right Panel (Properties):** Shows the properties for the selected "TEST" database. The "Name" is "<Aurora_MySQL (virtual)>".

- Quando a AWS SCT terminar a conversão, você poderá visualizar o esquema proposto no painel à direita do seu projeto.

Nesse ponto, nenhum esquema é aplicado à sua instância de banco de dados de destino. O esquema planejado é parte do seu projeto. Se você escolher um item de esquema convertido, será possível visualizar o comando de esquema planejado no painel na parte central inferior da sua instância de banco de dados de destino.

Você pode editar o esquema nesta janela. O esquema editado é armazenado como parte de seu projeto e gravado na instância de banco de dados de destino quando você opta por aplicar seu esquema convertido.



Aplicando o esquema convertido à instância de banco de dados de destino

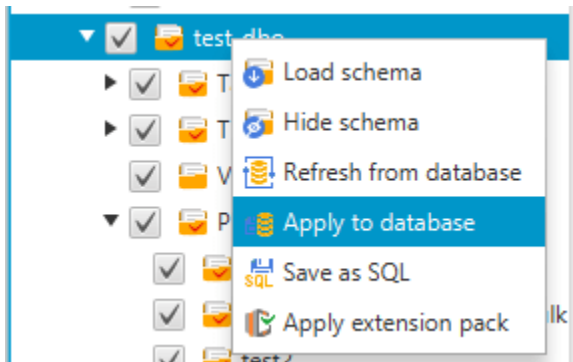
Você pode aplicar o esquema de banco de dados convertido à instância de banco de dados de destino. Depois que o esquema tiver sido aplicado à instância de banco de dados de destino, você poderá atualizar o esquema com base nos itens de ação do relatório de avaliação de migração do banco de dados.

Warning

O procedimento a seguir substitui o esquema de destino existente. Tenha cuidado para não substituir esquemas inadvertidamente. Tenha cuidado para não substituir esquemas em sua instância de banco de dados de destino que já foram modificados, ou você sobrescreverá essas alterações.

Para aplicar o esquema de banco de dados convertido à sua instância de banco de dados de destino

1. Escolha Conectar-se ao servidor na parte superior do painel direito do seu projeto para se conectar ao banco de dados de destino. Se você estiver conectado ao banco de dados de destino, pule esta etapa.
2. Escolha o elemento do esquema no painel direito do seu projeto que exibe o esquema planejado para sua instância de banco de dados de destino.
3. Abra o menu de contexto (clique com o botão direito do mouse) do elemento do esquema e escolha Aplicar ao banco de dados.



O esquema convertido é aplicado à instância do banco de dados de destino.

Armazenando perfis de serviço da AWS na AWS SCT

Você pode armazenar suas credenciais da AWS na AWS SCT. A AWS SCT usa suas credenciais quando você usa atributos que se integram aos serviços da AWS. Por exemplo, a AWS SCT integra-se a Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS) e AWS Database Migration Service (AWS DMS).

A AWS SCT solicita suas credenciais da AWS quando você acessa um atributo que as exige. Você pode armazenar suas credenciais nas configurações globais do aplicativo. Quando a AWS SCT solicitar suas credenciais, você poderá selecionar as credenciais armazenadas.

Você pode armazenar conjuntos diferentes de credenciais da AWS nas configurações globais do aplicativo. Por exemplo, você pode armazenar um conjunto de credenciais que usa em cenários de teste, e um conjunto de credenciais diferente que você usa em cenários de produção. Você também pode armazenar credenciais diferentes para diferentes Região da AWSs.

Armazenando credenciais da AWS

Use o procedimento a seguir para armazenar credenciais da AWS globalmente.

Para armazenar credenciais da AWS

1. Inicie o AWS Schema Conversion Tool.
2. Abra o menu Configurações e, em seguida, escolha Configurações globais. A caixa de diálogo Configurações globais é exibida.
3. Escolha Perfis de serviço da AWS e, em seguida, escolha Adicionar um novo perfil de serviço da AWS.
4. Insira as informações da AWS conforme a seguir.

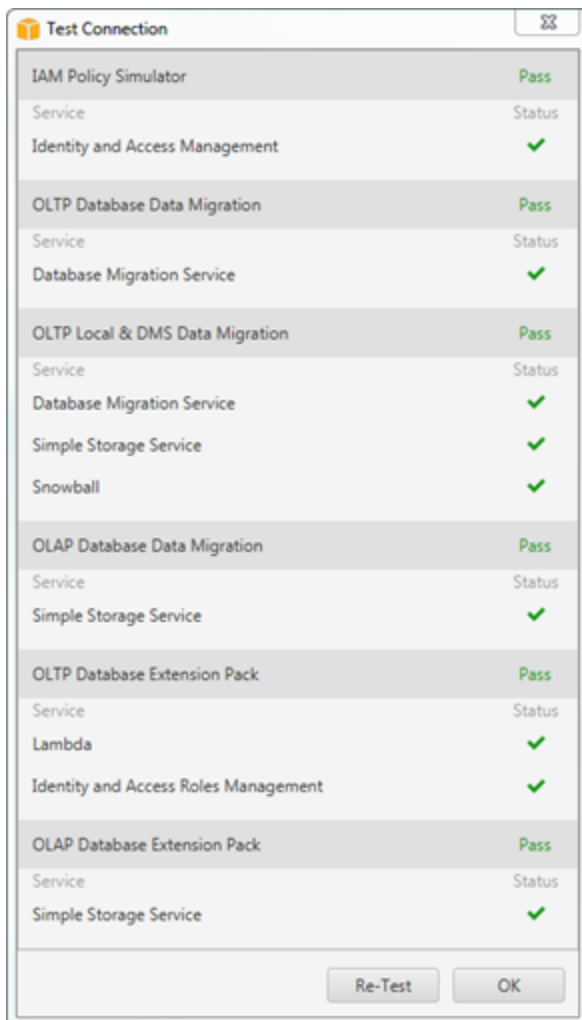
Opção do AWS SCT	Ação
Profile name	Insira um nome para o perfil.
Chave de acesso da AWS	Insira sua chave de acesso da AWS.
Chave secreta da AWS	Insira a chave de acesso secreta da AWS. Para obter mais informações sobre as chaves de acesso da AWS, consulte Como gerenciar chaves de acesso no Guia de usuário do IAM.
Região	Escolha a Região da AWS para o seu perfil.
Pasta de bucket do Amazon S3	Escolha o bucket do Amazon S3 para seu perfil. Você precisará especificar um bucket somente se estiver usando um atributo que se conecte ao Amazon S3. Para obter mais informações sobre os privilégios necessários, consulte a Permissões para usar o perfil de serviço da AWS .

Escolha Usar endpoint do FIPS para S3 se você precisa estar em conformidade com os requisitos de segurança do Federal Information Processing Standard (FIPS). Os endpoints do FIPS estão disponíveis nas seguintes Regiões da AWS:

- Região Leste dos EUA (N. da Virgínia)

- Região Leste dos EUA (Ohio)
 - Região Leste dos EUA (Norte da Califórnia)
 - Região Oeste dos EUA (Oregon)
5. Escolha Testar conexão para verificar se suas credenciais estão corretas e ativas.

A caixa de diálogo Testar conexão é exibida. Você pode ver o status de cada um dos serviços conectados ao seu perfil. Aprovação indica que o perfil pode acessar o serviço com êxito.



6. Depois de configurar seu perfil, selecione Salvar para salvar seu perfil ou Cancelar para cancelar as alterações.
7. Escolha OK para fechar a caixa de diálogo Configurações globais.

Configurando o perfil padrão de um projeto

Você pode definir o perfil padrão de um projeto da AWS SCT. Fazer isso associa as credenciais da AWS armazenadas no perfil com o projeto. Com seu projeto aberto, use o procedimento a seguir para definir o perfil padrão.

Para configurar o perfil padrão de um projeto

1. Inicie a AWS Schema Conversion Tool e crie um novo projeto.
2. No menu Configurações, selecione Configurações do projeto. A caixa de diálogo Configurações do projeto é exibida.
3. Escolha a guia Ambiente do projeto.
4. Escolha Adicionar um novo perfil de serviço da AWS para adicionar um novo perfil. Em Perfil do serviço da AWS, escolha o perfil que você deseja associar ao projeto.
5. Escolha OK para fechar a caixa de diálogo Configurações do projeto. Você também pode escolher Cancelar para cancelar as alterações.

Permissões para usar o perfil de serviço da AWS

As seguintes permissões são necessárias para acessar seu bucket do Amazon S3 a partir do seu perfil de serviço da AWS:

- `s3:PutObject` – para adicionar objetos ao bucket do Amazon S3.
- `s3:DeleteObject` – para remover a versão nula de um objeto e inserir um marcador de exclusão, que se torna a versão atual do objeto.
- `s3:ListBucket` – para retornar até 1.000 objetos do seu bucket do Amazon S3.
- `s3:GetObject` – para recuperar objetos do bucket do Amazon S3.

O exemplo de código a seguir mostra como conceder essas permissões ao usuário.

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

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

Usar o AWS Secrets Manager

A AWS SCT pode usar credenciais de banco de dados que você armazena no AWS Secrets Manager. Você pode preencher todos os valores na caixa de diálogo de conexão do banco de dados no Secrets Manager. Para usar o Secrets Manager, certifique-se de armazenar perfis da AWS no AWS Schema Conversion Tool.

Para obter mais informações sobre o uso do AWS Secrets Manager, consulte [O que é AWS Secrets Manager?](#) no Guia do usuário do AWS Secrets Manager. Para obter mais informações sobre armazenar perfis da AWS, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).

Para recuperar credenciais de banco de dados do Secrets Manager

1. Inicie a AWS Schema Conversion Tool e crie um novo projeto.
2. Escolha Adicionar origem ou Adicionar destino para adicionar um novo banco de dados ao seu projeto.
3. Escolha uma plataforma de banco de dados e, em seguida, escolha Próximo.
4. Para o AWS Secret, escolha o segredo que você quer usar.
5. Escolha Preencher. Em seguida, a AWS SCT preenche todos os valores na caixa de diálogo de conexão do banco de dados.
6. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados.
7. Escolha Conectar para se conectar ao seu banco de dados.

A AWS SCT oferece suporte a segredos que têm a seguinte estrutura.

```
{
```

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

Nessa estrutura, os valores `username` e `password` são obrigatórios, e todos os outros valores são opcionais. Certifique-se de que os valores que você armazena no Secrets Manager incluam todas as credenciais do banco de dados.

Armazenando senhas de banco de dados

Você pode armazenar uma senha de banco de dados ou um certificado SSL no cache da AWS SCT. Para armazenar uma senha, selecione `Store password` (Armazenar senha) ao criar uma conexão.

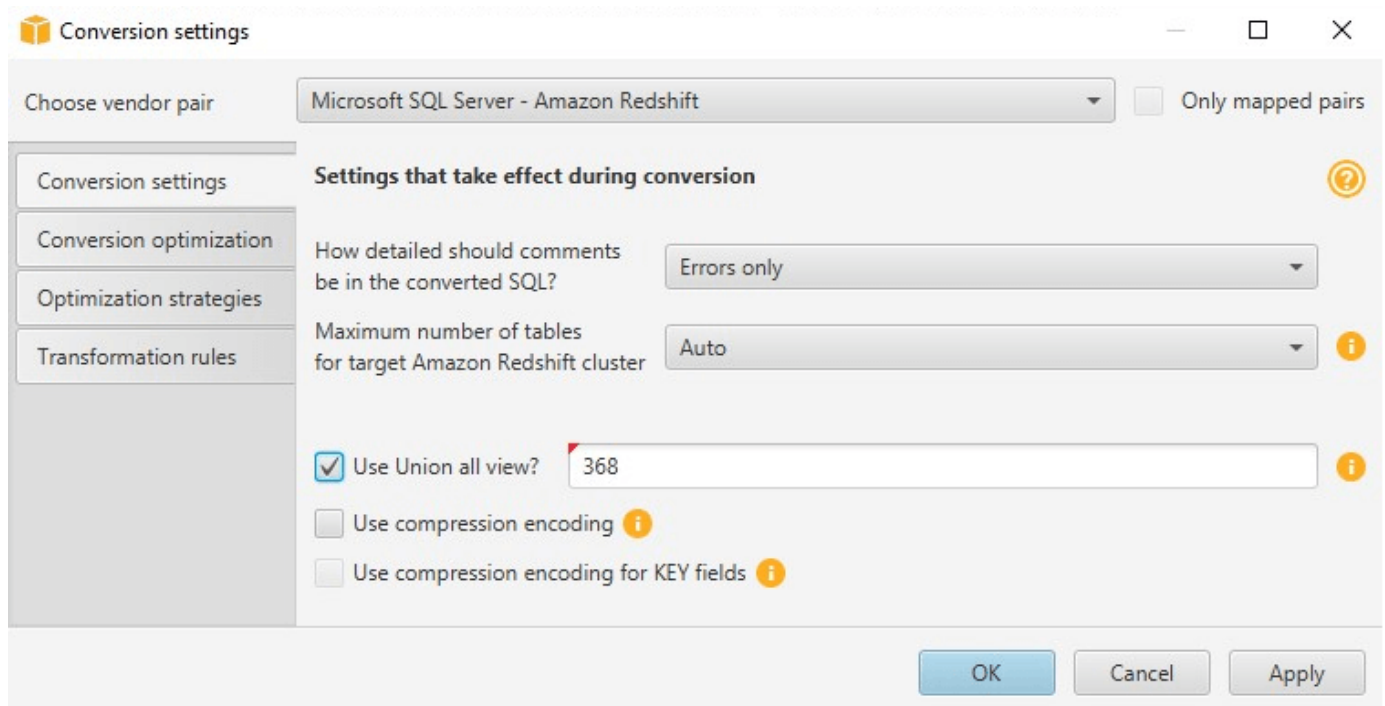
A senha é criptografada usando o token gerado aleatoriamente no arquivo `seed.dat`. Em seguida, ela é armazenada com o nome de usuário no arquivo de cache. Caso você perca o arquivo `seed.dat` ou ele seja corrompido, a senha do banco de dados pode ser descriptografada de maneira incorreta. Nesse caso, ocorre uma falha ao conectar-se.

Usando a visualização UNION ALL para projetos com tabelas particionadas

Se uma tabela de origem for particionada, a AWS SCT cria n tabelas de destino, onde n é o número de partições na tabela de origem. A AWS SCT cria uma visualização UNION ALL na parte superior das tabelas de destino para representar a tabela de origem. Se você usar um extrator de dados da AWS SCT para migrar seus dados, as partições da tabela de origem serão extraídas e carregadas em paralelo por subtarefas separadas.

Para usar a exibição Union All em um projeto

1. Inicie AWS SCT. Crie um novo projeto ou abra um projeto da AWS SCT existente.
2. No menu Configurações, escolha Configurações de conversão.
3. Escolha um par de bancos de dados OLAP na lista na parte superior.
4. Ativar Usar a visualização Union all?



- Escolha OK para salvar as configurações e fechar a caixa de diálogo Configurações de conversão.

Atalhos de teclado da AWS SCT

A seguir estão os atalhos de teclado que você pode usar com a AWS SCT.

Atalho de teclado	Descrição
Ctrl+N	Criar um novo projeto
Ctrl+O	Abrir um projeto existente.
Ctrl+S	Salvar um projeto aberto.
Ctrl+W	Criar um novo projeto usando o assistente.
Ctrl+M	Crie uma nova avaliação de vários servidores.
Ctrl+L	Adicione um novo banco de dados de origem.
Ctrl+R	Adicione um novo banco de dados de destino.

Atalho de teclado	Descrição
Ctrl+F4	Feche um projeto aberto.
F1	Abra o Guia de usuário da AWS SCT.

Conceitos básicos do AWS SCT

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter o esquema para um banco de dados de origem. O provedor de dados de origem pode ser um mecanismo auto-gerenciado em execução on-premises ou em uma instância do Amazon EC2. Você pode converter seu esquema de origem em um esquema para qualquer banco de dados compatível hospedado pelo AWS. O aplicativo do AWS SCT possui uma interface de usuário baseada em projeto.

Praticamente todas as tarefas que você faz com o AWS SCT começam com as etapas a seguir:

1. Instalar o AWS SCT. Para obter mais informações, consulte [Instalando, verificando e atualizando AWS SCT](#).
2. Instale um agente do AWS SCT, se necessário. Os agentes do AWS SCT são necessários apenas para determinados cenários de migração, como entre origens e destinos heterogêneos. Para obter mais informações, consulte [Como migrar dados de um data warehouse on-premises para o Amazon Redshift](#).
3. Familiarize-se com a interface do usuário do AWS SCT. Para obter mais informações, consulte [Uso da interface de usuário da AWS SCT](#).
4. Crie um projeto do AWS SCT. Conecte-se aos bancos de dados de origem e de destino. Para obter mais informações sobre como se conectar ao seu banco de dados de origem, consulte [Origens para a AWS SCT](#).
5. Crie regras de mapeamento. Para obter mais informações sobre regras de mapeamento, consulte [Criação de regras de mapeamento na AWS SCT](#).
6. Execute e analise o relatório de avaliação de migração do banco de dados. Para obter mais informações sobre o relatório de avaliação, consulte [Criando e revisando o relatório de avaliação de migração do banco de dados](#).
7. Converta os esquemas do banco de dados de origem. Há vários aspectos da conversão que você precisa ter em mente, como o que fazer com itens que não são convertidos e como mapear itens que devem ser convertidos de uma maneira específica. Para obter mais informações sobre como converter um esquema de origem, consulte [Como converter esquemas de bancos de dados usando a AWS SCT](#).

Se estiver convertendo um esquema de data warehouse, também há aspectos que você precisa considerar antes de fazer a conversão. Para obter mais informações, consulte [Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT](#).

8. Aplicar a conversão de esquema ao seu destino. Para obter mais informações sobre como aplicar uma conversão de esquema de origem, consulte [Aplicando o código convertido](#).
9. Você também pode usar AWS SCT para converter procedimentos armazenados e outros códigos de aplicativo do SQL. Para obter mais informações, consulte [Como converter o aplicativo SQL usando a AWS SCT](#).

Você também pode usar a AWS SCT para migrar seus dados de um banco de dados de origem para um banco de dados gerenciado pela Amazon. Para ver exemplos, consulte [Como migrar dados de um data warehouse on-premises para o Amazon Redshift](#).

Origens para a AWS SCT

A AWS Schema Conversion Tool (AWS SCT) pode converter esquemas dos bancos de dados de origem e data warehouses a seguir em um banco de dados de destino ou data warehouse. Para obter informações sobre permissões, conexões e o que a AWS SCT pode converter para uso com o banco de dados de destino ou data warehouse, consulte detalhes nos tópicos listados a seguir.

Informações de criptografia

[Como criptografar conexões do Amazon RDS](#)

Origens dos bancos de dados

- [Como usar o Apache Cassandra como origem](#)
- [Como usar um banco de dados Azure SQL como origem](#)
- [Como usar o IBM Db2 para z/OS como origem](#)
- [Usar o IBM Db2 LUW como origem](#)
- [Usar o MySQL como origem](#)
- [Usar um banco de dados Oracle como origem](#)
- [Usar o PostgreSQL como origem](#)
- [Usando o SAP ASE \(Sybase ASE\) como origem](#)
- [Usar o SQL Server como origem](#)

Origens de data warehouse

- [Como usar o Amazon Redshift como origem](#)
- [Como usar o Azure Synapse Analytics como origem](#)
- [Como usar o BigQuery como origem](#)
- [Como usar o banco de dados Greenplum como origem](#)
- [Como usar o Netezza como origem](#)
- [Como usar o data warehouse do Oracle como origem](#)
- [Como usar o Snowflake como origem](#)
- [Como usar o data warehouse do SQL Server como origem](#)
- [Como usar o Teradata como origem](#)

- [Como usar o Vertica como origem](#)

Origens de big data

- [Como usar o Apache Hadoop como origem](#)
- [Como usar o Apache Oozie como origem](#)

Como criptografar conexões do Amazon RDS e do Amazon Aurora na AWS SCT

Para abrir conexões criptografadas com bancos de dados do Amazon RDS ou do Amazon Aurora a partir de um aplicativo, você precisa importar certificados raiz da AWS em alguma forma de armazenamento de chaves. Você pode baixar os certificados raiz da AWS em [Como usar SSL/TLS para criptografar uma conexão com uma instância de banco de dados](#) no Guia do usuário do Amazon RDS.

Duas opções estão disponíveis: um certificado raiz que funciona para todas as regiões da AWS e um pacote de certificados que contém o certificado raiz antigo e o novo.

Dependendo do que você deseja usar, siga as etapas em um dos dois procedimentos a seguir.

Para importar o certificado ou certificados para o armazenamento do sistema Windows

1. Baixe um certificado ou certificados de uma das seguintes origens:

Para obter informações sobre como baixar certificados, consulte [Como usar SSL/TLS para criptografar uma conexão com uma instância de banco de dados](#) no Guia do usuário do Amazon RDS.

2. Na janela de pesquisa do Windows, digite **Manage computer certificates**. Quando solicitado se você deseja permitir que o aplicativo faça alterações em seu computador, escolha Sim.
3. Quando a janela de certificados abrir, se necessário, expanda Certificados - Computador local para ver a lista de certificados. Abra o menu contextual (clique com o botão direito) para Autoridades de certificação raiz confiáveis, escolha Todas as tarefas, e selecione Importar.
4. Escolha Avançar, depois Procurar e encontre o arquivo *.pem que você baixou na etapa 1. Escolha Abrir para selecionar o arquivo do certificado, escolha Avançar e depois escolha Concluir.

Note

Para encontrar o arquivo, altere o tipo de arquivo na janela de busca para Todos os arquivos (*.*), pois .pem não é uma extensão de certificado padrão.

5. No Microsoft Management Console, expanda Certificados. Em seguida, expanda Autoridades de certificação raiz confiáveis, escolha Certificados e encontre o certificado para confirmar que ele existe. O nome do certificado começa com Amazon RDS.
6. Reinicie o computador.

Para importar o certificado ou certificados para o repositório de chaves Java

1. Baixe o certificado ou certificados de uma das seguintes origens:

Para obter informações sobre como baixar certificados, consulte [Como usar SSL/TLS para criptografar uma conexão com uma instância de banco de dados](#) no Guia do usuário do Amazon RDS.

2. Se você baixou o pacote de certificados, divida-o em arquivos de certificados individuais. Para fazer isso, coloque cada bloco de certificados, começando com -----BEGIN CERTIFICATE----- e terminando com -----END CERTIFICATE----- em arquivos *.pem separados. Depois de criar um arquivo *.pem separado para cada certificado, você pode remover com segurança o arquivo do pacote de certificados.
3. Abra uma janela de comando ou sessão de terminal no diretório em que você baixou o certificado e execute o comando a seguir para cada arquivo *.pem criado na etapa anterior.

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

Example

O exemplo a seguir pressupõe que você baixou o arquivo 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. Adicione o repositório de chaves como um armazenamento confiável em AWS SCT. Para fazer isso, no menu principal, escolha Configurações, Configurações globais, Segurança, Armazenamento confiável e, em seguida, escolha Selecionar armazenamento confiável existente.

Depois de adicionar o armazenamento confiável, você pode usá-lo para configurar uma conexão habilitada para SSL ao criar uma conexão de AWS SCT com o banco de dados. Na caixa de diálogo Conectar ao banco de dados da AWS SCT, escolha Usar SSL e escolha o armazenamento confiável inserido anteriormente.

Como usar o Apache Cassandra como origem para AWS SCT

Você pode usar AWS SCT para converter espaços de chave do Apache Cassandra para o Amazon DynamoDB.

Como se conectar ao Apache Cassandra como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Apache Cassandra com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem do Apache Cassandra

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Selecione Cassandra e Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados de origem do Apache Cassandra, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção se você quiser usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none">• Armazenamento confiável: o armazenamento confiável a ser usado.• Armazenamento de chaves: O armazenamento de chaves a ser usado.

Parâmetro	Ação
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Como usar o Apache Hadoop como origem para AWS SCT

É possível usar a interface de linha de comandos (CLI) da AWS SCT para migrar do Apache Hadoop para o Amazon EMR. A AWS SCT usa seu bucket do Amazon S3 como armazenamento temporário para seus dados durante a migração.

A AWS SCT oferece suporte como Apache Hadoop de origem versão 2.2.0 e superior. Além disso, a AWS SCT oferece suporte com o Apache Hive versão 0.13.0 e superior.

A AWS SCT oferece suporte como Amazon EMR de destino versão 6.3.0 e superior. Além disso, a AWS SCT oferece suporte como Apache Hadoop de destino versão 2.6.0 e superior e Apache Hive versão 0.13.0 e superior.

Tópicos

- [Pré-requisitos de uso do Apache Hadoop como origem](#)
- [Permissões para usar o Hive como origem](#)
- [Permissões para usar o HDFS como origem](#)
- [Permissões para usar o HDFS como destino](#)
- [Como se conectar ao Apache Hadoop como origem](#)
- [Como se conectar aos serviços Hive e HDFS de origem](#)
- [Como se conectar ao Amazon EMR como destino](#)

Pré-requisitos de uso do Apache Hadoop como origem

Os pré-requisitos a seguir são necessários para se conectar ao Apache Hadoop com a CLI da AWS SCT.

- Crie um bucket do Amazon S3 para armazenar dados durante a migração. Em seguida, você pode copiar dados para o Amazon EMR HDFS ou usar o Amazon S3 como um repositório de dados para suas cargas de trabalho do Hadoop. Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.
- Crie um perfil (IAM) AWS Identity and Access Management com a política de AmazonS3FullAccess. A AWS SCT usa esse perfil do IAM para acessar o bucket do Amazon S3.
- Anote sua chave AWS secreta e sua chave de acesso AWS secreta. Para obter mais informações sobre as chaves de acesso da AWS, consulte [Como gerenciar chaves de acesso](#) no Guia de usuário do IAM.
- Criar e configurar um cluster do Amazon EMR de destino. Para obter mais informações, consulte [Conceitos básicos do Amazon EMR](#) no Guia de gerenciamento do Amazon EMR.
- Instale o utilitário `distcp` em seu cluster Apache Hadoop de origem. Além disso, instale o utilitário `s3-dist-cp` em seu cluster do Amazon EMR de destino. Certifique-se de que os usuários do banco de dados tenham permissões para executar esses utilitários.
- Configure o arquivo `core-site.xml` em seu cluster Hadoop de origem para usar o protocolo `s3a`. Para fazer isso, defina o parâmetro `fs.s3a.aws.credentials.provider` com um dos seguintes 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`

É possível adicionar o exemplo de código a seguir ao arquivo `core-site.xml`.

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

O exemplo anterior mostra uma das quatro opções da lista de opções anterior. Se você não definir o parâmetro `fs.s3a.aws.credentials.provider` no arquivo `core-site.xml`, a AWS SCT escolherá o provedor automaticamente.

Permissões para usar o Hive como origem

As permissões necessárias para um usuário do Hive de origem são as seguintes:

- Acesso READ às pastas de dados de origem e ao bucket do Amazon S3 de origem
- Acesso READ+WRITE aos buckets do Amazon S3 intermediários e de destino

Para aumentar a velocidade de migração, recomendamos que você execute a compactação para tabelas de origem transacionais ACID.

As permissões necessárias para um usuário do Amazon EMR Hive de destino são as seguintes:

- Acesso READ ao bucket do Amazon S3 de destino
- Acesso READ+WRITE ao bucket do Amazon S3 intermediário
- Acesso READ+WRITE às pastas HDFS de destino

Permissões para usar o HDFS como origem

As permissões necessárias para o HDFS como origem são as seguintes:

- EXECUTE para o NameNode
- EXECUTE+READ para todas as pastas e arquivos de origem que você inclui no projeto de migração
- READ+WRITE para o diretório `tmp` no NameNode para executar trabalhos do Spark e armazenar arquivos antes da migração para o Amazon S3

No HDFS, todas as operações exigem acesso transversal. O acesso transversal exige a permissão da EXECUTE em todos os componentes existentes do caminho, exceto no componente final do caminho. Por exemplo, para qualquer operação de acesso `/foo/bar/baz`, seu usuário deve ter permissão EXECUTE em `/`, `/foo` e `/foo/bar`.

O exemplo de código a seguir demonstra como conceder permissões EXECUTE+READ para suas pastas e arquivos de origem e permissões READ+WRITE para o diretório `tmp`.

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

Permissões para usar o HDFS como destino

As permissões necessárias para o Amazon EMR HDFS como destino são as seguintes:

- EXECUTE para o NameNode do cluster Amazon EMR de destino
- READ+WRITE para as pastas HDFS de destino nas quais você armazenará dados após a migração

Como se conectar ao Apache Hadoop como origem

Você pode usar o Apache Hadoop como origem na AWS SCT versão 1.0.670 ou superior. É possível migrar clusters do Hadoop para o Amazon EMR somente na interface de linha de comandos (CLI) da AWS SCT. Antes de começar a usar, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte [AWS SCT Referência da CLI](#).

Para se conectar ao Apache Hadoop na CLI da AWS SCT

1. Crie um novo script de CLI da AWS SCT ou edite um modelo de cenário existente. Por exemplo, é possível baixar e editar o modelo `HadoopMigrationTemplate.scts`. Para obter mais informações, consulte [Obter cenários de CLI](#).
2. Defina as configurações do aplicativo AWS SCT, como a localização do driver e a pasta do log.

Faça o download do driver JDBC necessário e especifique o local de armazenamento do arquivo. Para obter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

O exemplo de código a seguir mostra como adicionar o caminho ao driver do Apache Hive. Depois de executar esse exemplo de código, a AWS SCT armazena os arquivos de log na pasta `c:\sct`.

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

```
/
```

Você pode usar esse exemplo e os exemplos a seguir no Windows.

3. Criar um novo projeto da AWS SCT.

O exemplo de código a seguir cria o projeto da `hadoop_emr` na pasta `c:\sct`.

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

4. Adicione seu cluster Hadoop de origem ao projeto.

Use o comando `AddSourceCluster` para se conectar ao cluster Hadoop de origem. Certifique-se de fornecer valores para os seguintes parâmetros obrigatórios: `name`, `host`, `port` e `user`. Outros parâmetros são opcionais.

O exemplo de código a seguir adiciona o cluster Hadoop de origem. Este exemplo define `HADOOP_SOURCE` como um nome do cluster de origem. Use esse nome de objeto para adicionar serviços Hive e HDFS ao projeto e criar regras de mapeamento.

```
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'
/
```

No exemplo anterior, substitua *hadoop_address* pelo endereço IP do seu cluster Hadoop. Se necessário, configure o valor da opção de porta. Em seguida, substitua *hadoop_user* e *hadoop_password* pelo nome do seu usuário do Hadoop e pela senha desse usuário. Em *path\name*, insira o nome e o caminho para o arquivo PEM do seu cluster Hadoop de origem.

5. Salve o script da CLI. Em seguida, adicione as informações de conexão dos seus serviços Hive e HDFS.

Como se conectar aos serviços Hive e HDFS de origem

Você pode se conectar aos serviços Hive e HDFS de origem com a CLI da AWS SCT. Para se conectar ao Apache Hive, use o driver JDBC do Hive versão 2.3.4 ou superior. Para obter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

A AWS SCT se conecta ao Apache Hive com o usuário do cluster hadoop. Para fazer isso, use os comandos `AddSourceClusterHive` e `AddSourceClusterHDFS`. Você pode usar uma das abordagens a seguir.

- Crie um novo túnel SSH.

Em `createTunnel`, digite **true**. Para `host`, insira o endereço IP interno do seu serviço Hive ou HDFS de origem. Para `port`, insira a porta de serviço do seu serviço Hive ou HDFS.

Em seguida, insira suas credenciais do Hive ou do HDFS para `user` e `password`. Para obter mais informações sobre túneis SSH, consulte [Configurar um túnel SSH ao nó primário usando o encaminhamento de portas locais](#) no Guia de gerenciamento do Amazon EMR.

- Use um túnel SSH existente.

Em `host`, digite **localhost**. Para `port`, insira a porta local a partir dos parâmetros do túnel SSH.

- Conecte-se diretamente aos seus serviços Hive e HDFS.

Para `host`, insira o endereço IP ou nome do host do seu serviço Hive ou HDFS de origem. Para `port`, insira a porta de serviço do seu serviço Hive ou HDFS. Em seguida, insira suas credenciais do Hive ou do HDFS para `user` e `password`.

Para se conectar ao Hive e ao HDFS na CLI da AWS SCT

1. Abra seu script de CLI, que inclui as informações de conexão do seu cluster Hadoop de origem. Certifique-se de usar o nome do cluster Hadoop que você definiu na etapa anterior.
2. Adicione seu serviço Hive de origem ao projeto.

Use o comando `AddSourceClusterHive` para conectar o serviço Hive de origem. Certifique-se de fornecer valores para os seguintes parâmetros obrigatórios: `user`, `password`, `cluster`, `name` e `port`. Outros parâmetros são opcionais.

O exemplo de código a seguir cria um túnel para AWS SCT trabalhar com seu serviço Hive. Esse serviço Hive de origem é executado no mesmo PC que AWS SCT. Este exemplo usa o cluster de HADOOP_SOURCE de origem do exemplo 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'
/
```

O exemplo de código a seguir se conecta ao seu serviço Hive sem um túnel.

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

Nos exemplos anteriores, substitua *hive_user* e *hive_password* pelo nome do seu usuário do Hive e pela senha desse usuário.

Em seguida, substitua *hive_address* e *hive_port* pelo endereço IP NameNode e pela porta do seu cluster Hadoop de origem.

Para *hive_remote_address*, você pode usar o valor padrão 127.0.0.1 ou o endereço IP NameNode do seu serviço Hive de origem.

3. Adicione seu serviço HDFS de origem ao projeto.

Use o comando `AddSourceClusterHDFS` para conectar o serviço HDFS de origem. Certifique-se de fornecer valores para os seguintes parâmetros obrigatórios: `user`, `password`, `cluster`, `name` e `port`. Outros parâmetros são opcionais.

Certifique-se de que seu usuário tenha as permissões necessárias para migrar dados do serviço HDFS de origem. Para obter mais informações, consulte [Permissões para usar o Hive como origem](#).

O exemplo de código a seguir cria um túnel para AWS SCT trabalhar com seu serviço Apache HDFS. Este exemplo usa o cluster `HADOOP_SOURCE` de origem que você criou 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'
/
```

O código a seguir se conecta ao seu serviço Apache HDFS sem um túnel.

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

Nos exemplos anteriores, substitua *hdfs_user* e *hdfs_password* pelo nome do seu usuário do HDFS e pela senha desse usuário.

Em seguida, substitua *hdfs_address* e *hdfs_port* pelo endereço IP NameNode e pela porta do seu cluster Hadoop de origem.

Para *hdfs_remote_address*, você pode usar o valor padrão 127.0.0.1 ou o endereço IP NameNode do seu serviço Hive de origem.

4. Salve o script da CLI. Em seguida, adicione as informações de conexão do seu cluster do Amazon EMR de destino e os comandos de migração.

Como se conectar ao Amazon EMR como destino

Você pode se conectar ao seu cluster do Amazon EMR de destino com a CLI da AWS SCT. Para fazer isso, você autoriza o tráfego de entrada e usa o SSH. Nesse caso, AWS SCT tem todas as permissões necessárias para trabalhar com seu cluster do Amazon EMR. Para obter mais informações, consulte [Antes de se conectar](#) e [Conectar-se ao nó primário usando SSH](#) no Guia de gerenciamento do Amazon EMR.

AWS SCT se conecta ao Hive do Amazon EMR com o usuário do cluster hadoop. Para se conectar ao Hive do Amazon EMR, use o driver JDBC do Hive versão 2.6.2.1002 ou superior. Para obter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

Para se conectar ao Amazon EMR na CLI da AWS SCT

1. Abra seu script de CLI, que inclui as informações de conexão do seu cluster Hadoop de origem. Adicione as credenciais do Amazon EMR de destino a esse arquivo.
2. Adicione seu cluster do Amazon EMR de destino ao projeto.

O exemplo de código a seguir adiciona o cluster do Amazon EMR de destino. Este exemplo define HADOOP_TARGET como um nome do cluster de destino. Use esse nome de objeto para adicionar seus serviços Hive e HDFS e uma pasta de bucket do Amazon S3 ao projeto e crie regras de mapeamento.

```
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'  
/
```

No exemplo anterior, insira os nomes dos recursos AWS e as informações de conexão do Amazon EMR. Isso inclui o endereço IP do seu cluster do Amazon EMR, chave de acesso AWS, chave de acesso secreta AWS e bucket do Amazon S3. Se necessário, configure o valor da variável de porta. Em seguida, substitua *emr_user* e *emr_password* pelo nome do seu usuário do Amazon EMR e pela senha desse usuário. Em *path\name*, insira o nome e o caminho para o arquivo PEM do seu cluster do Amazon EMR de destino. Para mais informações, consulte [Baixar arquivo PEM para acesso ao cluster do EMR](#).

3. Adicione o bucket do Amazon S3 de destino ao projeto.

O exemplo de código a seguir adiciona o bucket do Amazon S3 de destino. Este exemplo usa o cluster HADOOP_TARGET que você criou antes.

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

No exemplo anterior, insira sua chave de acesso AWS, chave de acesso secreta AWS e bucket do Amazon S3.

4. Adicione seu serviço Hive de destino ao projeto.

O exemplo de código a seguir cria um túnel para AWS SCT trabalhar com seu serviço Hive de destino. Este exemplo usa o cluster HADOOP_TARGET de destino que você criou 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'  
/  

```

No exemplo anterior, substitua *hive_user* e *hive_password* pelo nome do seu usuário do Hive e pela senha desse usuário.

Em seguida, substitua *hive_address* pelo valor padrão `127.0.0.1` ou pelo endereço IP NameNode do seu serviço Hive de destino. Em seguida, substitua *hive_port* pela porta do serviço Hive de destino.

5. Adicione seu serviço HDFS de destino ao projeto.

O exemplo de código a seguir cria um túnel para AWS SCT trabalhar com seu serviço Apache HDFS. Este exemplo usa o cluster HADOOP_TARGET de destino que você criou 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'  
/  

```

No exemplo anterior, substitua *hdfs_user* e *hdfs_password* pelo nome do usuário do HDFS e pela senha desse usuário.

Em seguida, substitua *hdfs_address* e *hdfs_port* pelo endereço IP privado e pela porta do NameNode do seu serviço HDFS de destino.

6. Salve o script da CLI. Em seguida, adicione regras de mapeamento e comandos de migração. Para obter mais informações, consulte [Como migrar o Apache Hadoop para o Amazon EMR](#).

Como usar o Apache Oozie como origem para AWS SCT

É possível usar a interface de linha de comandos (CLI) da AWS SCT para converter fluxos de trabalho do Apache Oozie para AWS Step Functions. Depois de migrar suas cargas de trabalho do Apache Hadoop para o Amazon EMR, você pode usar um serviço nativo na Nuvem AWS para orquestrar seus trabalhos. Para obter mais informações, consulte [Como usar o Apache Hadoop como origem](#).

A AWS SCT converte seus fluxos de trabalho do Oozie para AWS Step Functions e usa AWS Lambda para emular recursos que AWS Step Functions não oferece suporte. Além disso, a AWS SCT converte suas propriedades de trabalho do Oozie em AWS Systems Manager.

Para converter fluxos de trabalho do Apache Oozie, certifique-se de usar a versão 1.0.671 ou superior da AWS SCT. Além disso, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte [AWS SCT Referência da CLI](#).

Pré-requisitos de uso do Apache Oozie como origem

Os pré-requisitos a seguir são necessários para se conectar ao Apache Oozie com a CLI da AWS SCT.

- Crie um bucket do Amazon S3 para armazenar as definições das máquinas de estado. Você pode usar essas definições para configurar suas máquinas de estado. Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.
- Crie um perfil (IAM) AWS Identity and Access Management com a política de AmazonS3FullAccess. A AWS SCT usa esse perfil do IAM para acessar o bucket do Amazon S3.
- Anote sua chave AWS secreta e sua chave de acesso AWS secreta. Para obter mais informações sobre as chaves de acesso da AWS, consulte [Como gerenciar chaves de acesso](#) no Guia de usuário do IAM.
- Armazene suas credenciais AWS e as informações sobre seu bucket do Amazon S3 no perfil de serviço AWS nas configurações globais do aplicativo. Em seguida, a AWS SCT usa esse perfil de serviço AWS para trabalhar com seus recursos AWS. Para obter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).

Para trabalhar com seus fluxos de trabalho do Apache Oozie de origem, a AWS SCT precisa de estrutura específica dos seus arquivos de origem. Cada uma das pastas do aplicativo deve incluir

o arquivo `job.properties`. Esse arquivo inclui pares de valores-chave das propriedades do seu trabalho. Além disso, cada uma das pastas do aplicativo deve incluir o arquivo `workflow.xml`. Esse arquivo descreve os nós de ação e os nós de fluxo de controle do seu fluxo de trabalho.

Como se conectar ao Apache Oozie como origem

Use o procedimento a seguir para se conectar aos seus arquivos do Apache Oozie de origem.

Para se conectar ao Apache Oozie na CLI da AWS SCT.

1. Crie um novo script de CLI da AWS SCT ou edite um modelo de cenário existente. Por exemplo, é possível baixar e editar o modelo `OozieConversionTemplate.scts`. Para obter mais informações, consulte [Obter cenários de CLI](#).
2. Defina as configurações do aplicativo da AWS SCT.

O exemplo de código a seguir salva as configurações do aplicativo e permite armazenar senhas em seu projeto. Você pode usar essas configurações salvas em outros projetos.

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

3. Criar um novo projeto da AWS SCT.

O exemplo de código a seguir cria o projeto da oozie na pasta `c:\sct`.

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

4. Adicione a pasta com seus arquivos Apache Oozie de origem ao projeto usando o comando `AddSource`. Certifique-se de usar o valor `APACHE_OOZIE` para o parâmetro `vendor`. Forneça também valores para os seguintes parâmetros necessários: `name` e `mappingsFolder`.

O exemplo de código a seguir adiciona o Apache Oozie como origem em seu projeto AWS SCT. Este exemplo cria um objeto de origem com o nome `OOZIE`. Use esse nome de objeto para

adicionar regras de mapeamento. Depois de executar esse exemplo de código, a AWS SCT usa a pasta `c:\oozie` para carregar seus arquivos de origem no projeto.

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

Você pode usar esse exemplo e os exemplos a seguir no Windows.

5. Conecte-se aos arquivos do Apache Oozie de origem usando o comando `ConnectSource`. Use o nome do objeto de origem que você definiu na etapa anterior.

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

6. Salve o script da CLI. Em seguida, adicione as informações de conexão do seu serviço AWS Step Functions.

Permissões para usar funções AWS Lambda no pacote de extensões

Para as funções de origem que a AWS Step Functions não oferece suporte, a AWS SCT cria um pacote de extensão. Esse pacote de extensão inclui funções da AWS Lambda que emulam suas funções de origem.

Para usar esse pacote de extensão, crie um perfil (IAM) da AWS Identity and Access Management com as seguintes permissões.

```
{
  "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 o pacote de extensão, a AWS SCT precisa de um perfil do IAM com as permissões a seguir.

```

{
    "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"
    ]
}
]
}

```

Como se conectar a AWS Step Functions como destino

Use o procedimento a seguir para se conectar à AWS Step Functions como destino.

Para se conectar à AWS Step Functions na CLI da AWS SCT

1. Abra seu script de CLI, que inclui as informações de conexão para seus arquivos do Apache Oozie de origem.
2. Adicione as informações sobre sua meta de migração no projeto AWS SCT usando o comando `AddTarget`. Certifique-se de usar o valor `STEP_FUNCTIONS` para o parâmetro `vendor`. Forneça também valores para os seguintes parâmetros necessários: `name` e `profile`.

O exemplo de código a seguir adiciona AWS Step Functions como origem em seu projeto AWS SCT. Este exemplo cria um objeto de destino com o nome `AWS_STEP_FUNCTIONS`. Use esse nome de objeto ao criar regras de mapeamento. Além disso, esse exemplo usa um perfil de serviço AWS SCT que você criou na etapa de pré-requisitos. Certifique-se de substituir *profile_name* pelo nome do seu perfil.

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

Se você não usar o perfil de serviço AWS, certifique-se de fornecer valores para os seguintes parâmetros obrigatórios: `accessKey`, `secretKey`, `awsRegion` e `s3Path`. Use esses parâmetros para especificar sua chave de acesso secreta AWS, chave secreta AWS, Região da AWS e o caminho para seu bucket do Amazon S3.

3. Conecte-se a AWS Step Functions usando o comando `ConnectTarget`. Use o nome do objeto de destino que você definiu na etapa anterior.

O exemplo de código a seguir se conecta ao objeto de destino `AWS_STEP_FUNCTIONS` usando seu perfil de serviço AWS. Certifique-se de substituir *profile_name* pelo nome do seu perfil.

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

4. Salve o script da CLI. Em seguida, adicione regras de mapeamento e comandos de migração. Para obter mais informações, consulte [Como converter o Apache Oozie para AWS Step Functions](#).

Como usar um banco de dados Azure SQL como origem para AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código e o código de aplicativo do banco de dados Azure SQL para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL

Tópicos

- [Privilégios do banco de dados Azure SQL como origem](#)
- [Como se conectar ao banco de dados Azure SQL de origem](#)

Privilégios do banco de dados Azure SQL como origem

Os privilégios necessários para o banco de dados Azure SQL como origem são os seguintes:

- VIEW DEFINITION
- VIEW DATABASE STATE

Repetir a concessão para cada banco de dados cujo esquema que você está convertendo.

Os privilégios necessários para bancos de dados MySQL e PostgreSQL de destino estão descritos nas seções a seguir.

- [Privilégios do MySQL como um banco de dados de destino](#)
- [Privilégios do PostgreSQL como um banco de dados de destino](#)

Como se conectar ao banco de dados Azure SQL de origem

Use o procedimento a seguir para conectar seu banco de dados Azure SQL de origem com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados Azure SQL de origem

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha banco de dados Azure SQL e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados de origem Azure SQL, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Database	Insira o nome do banco de dados ao qual se conectar.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se</p>

Parâmetro	Ação
	conectar ao banco de dados de origem, conforme necessário.
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Como usar o IBM Db2 para z/OS como origem para AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código e o código de aplicativo do IBM Db2 para z/OS para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL

Pré-requisitos para o Db2 para z/OS como banco de dados de origem

A versão do banco de dados de nível de função 100 do IBM Db2 para z/OS versão 12 não suporta a maioria dos novos recursos do IBM Db2 para z/OS versão 12. Essa versão do banco de dados fornece suporte para fallback para o Db2 versão 11 e compartilhamento de dados com o Db2 versão 11. Para evitar a conversão de recursos não suportados do Db2 versão 11, recomendamos que você use uma função de banco de dados IBM Db2 para z/OS de nível 500 ou superior como origem para AWS SCT.

Você pode usar o exemplo de código a seguir para verificar a versão do seu banco de dados IBM Db2 para z/OS de origem.

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

Certifique-se de que esse código retorne uma versão DSN12015 ou superior.

Você pode usar o exemplo de código a seguir para verificar o valor do registro especial APPLICATION COMPATIBILITY em seu banco de dados IBM Db2 para z/OS de origem.

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

Certifique-se de que esse código retorne uma versão V12R1M500 ou superior.

Privilégios do Db2 para z/OS como um banco de dados de origem

Os privilégios necessários para se conectar a um banco de dados Db2 para z/OS e ler catálogos e tabelas do sistema são os seguintes:

- 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
- SELECT ON 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 converter tabelas do Db2 para z/OS em tabelas particionadas do PostgreSQL, reúna estatísticas sobre espaços de tabela e tabelas em seu banco de dados usando o utilitário RUNSTATS, conforme mostrado a seguir.

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

No exemplo anterior, substitua o espaço reservado *YOURDB* pelo nome do banco de dados de origem.

Como se conectar ao Db2 para z/OS como origem

Use o procedimento a seguir para conectar seu banco de dados Db2 para z/OS de origem com AWS SCT.

Para se conectar a um banco de dados IBM Db2 para z/OS de origem

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Db2 para z/OS e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:

- Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados de origem do IBM Db2 para z/OS, use as seguintes instruções:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Local	Insira o nome exclusivo da localização do Db2 que você deseja acessar.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>

Parâmetro	Ação
Usar SSL	<p>Escolha essa opção se você quiser usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Db2 para z/OS	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Privilégios do MySQL como um banco de dados de destino

Os privilégios obrigatórios para MySQL como destino são listados a seguir:

- 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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```


No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para MySQL como destino, defina o parâmetro `log_bin_trust_function_creators` como verdadeiro e o `character_set_server` como `latin1`. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Para usar o Aurora MySQL como destino, defina o parâmetro `log_bin_trust_function_creators` como verdadeiro e o `character_set_server` como `latin1`. Defina também o parâmetro `lower_case_table_names` como verdadeiro. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio `CREATE ON DATABASE`. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar o Amazon RDS para PostgreSQL como destino, a AWS SCT necessita do privilégio `rds_superuser`.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para `"$user", public_synonyms, public`.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Configurações de conversão do Db2 para z/OS para PostgreSQL

Para editar as configurações de conversão do Db2 para z/OS para PostgreSQL, escolha Configurações e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Db2 para z/OS e, em seguida, escolha Db2 para z/OS — PostgreSQL ou Db2 para z/OS — Amazon Aurora (compatível com PostgreSQL). A AWS SCT exibe todas as configurações disponíveis para a conversão do IBM Db2 para z/OS para PostgreSQL.

As configurações de conversão do Db2 para z/OS para PostgreSQL em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para gerar nomes exclusivos para restrições no banco de dados de destino.

No PostgreSQL, todos os nomes de restrições que você usa devem ser exclusivos. A AWS SCT pode gerar nomes exclusivos para restrições no código convertido adicionando um prefixo com o nome da tabela ao nome da restrição. Para garantir que a AWS SCT gere nomes exclusivos para suas restrições, selecione Gerar nomes exclusivos para restrições.

- Para manter a formatação de nomes de colunas, expressões e cláusulas em instruções DML no código convertido.

A AWS SCT pode manter o layout dos nomes das colunas, expressões e cláusulas nas instruções DML na posição e ordem semelhantes às do código-fonte. Para fazer isso, selecione Sim para Manter a formatação de nomes de colunas, expressões e cláusulas em instruções DML.

- Para excluir partições de tabela do escopo de conversão.

A AWS SCT pode ignorar todas as partições de uma tabela de origem durante a conversão. Para fazer isso, selecione Excluir partições de tabela do escopo de conversão.

- Para usar o particionamento automático para tabelas que são particionadas por crescimento.

Para migração de dados, a AWS SCT pode particionar automaticamente todas as tabelas maiores que o tamanho especificado. Para usar essa opção, selecione Aplicar partição de tabelas maiores que e insira o tamanho das tabelas em gigabytes. Em seguida, insira o número de partições. A AWS SCT considera o tamanho do dispositivo de armazenamento de acesso direto (DASD) do seu banco de dados de origem quando você ativa essa opção.

A AWS SCT pode determinar o número de partições automaticamente. Para fazer isso, selecione Aumentar o número de partições proporcionalmente e insira o número máximo de partições.

- Para retornar conjuntos de resultados dinâmicos como uma matriz de valores do tipo de dados refcursor.

A AWS SCT pode converter procedimentos de origem que retornam conjuntos de resultados dinâmicos em procedimentos que têm uma matriz de refcursors abertos como um parâmetro de saída adicional. Para fazer isso, selecione Usar uma matriz de refcursors para retornar todos os conjuntos de resultados dinâmicos.

- Para especificar o padrão a ser usado para a conversão de valores de data e hora em representações de string.

A AWS SCT pode converter valores de data e hora em representações de string usando um dos formatos do setor suportados. Para fazer isso, selecione Usar representações de string de valores de data ou Usar representações de string de valores de tempo. Em seguida, escolha um dos padrões a seguir.

- Organização Internacional de Normalização (ISO)
- Padrão Europeu IBM (EUR)
- Padrão Norte-Americano IBM (EUA)
- Padrão Industrial Japonês da Era Cristã (JIS)

Usar o IBM Db2 LUW como origem para a AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código na linguagem SQL e código de aplicativo do IBM Db2 para Linux, Unix e Windows (Db2 LUW) para os seguintes destinos.

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon RDS para MariaDB

A AWS SCT oferece suporte como Db2 LUW de origem versões 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 e 11.5.

Privilégios do Db2 LUW como origem

Os privilégios necessários para se conectar a um banco de dados Db2 LUW, para verificar privilégios disponíveis e ler metadados de esquema para uma origem são listados a seguir:

- Privilégio necessário para estabelecer uma conexão:
 - CONNECT ON DATABASE
- Privilégios necessários para executar instruções SQL:
 - EXECUTE ON PACKAGE NULLID.SYSSH200
- Privilégios necessários para obter informações no nível de instância:
 - EXECUTE ON FUNCTION SYSPROC.ENV_GET_INST_INFO
 - SELECT ON SYSIBMADM.ENV_INST_INFO
 - SELECT ON SYSIBMADM.ENV_SYS_INFO
- Privilégios necessários para verificar os privilégios concedidos por meio de funções, grupos e 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
- Privilégios necessários em catálogos e tabelas do 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
- SELECT ON SYSCAT.TABCONST
- SELECT ON SYSCAT.TABLES
- SELECT ON SYSCAT.TRIGGERS
- SELECT ON SYSCAT.VARIABLEDEP
- SELECT ON SYSCAT.VARIABLES
- SELECT ON SYSCAT.VIEWS
- SELECT ON SYSIBM.SYSDUMMY1
- Para executar instruções SQL, a conta do usuário precisa de um privilégio para usar pelo menos uma das cargas de trabalho ativadas no banco de dados. Se nenhuma das cargas de trabalho for atribuída ao usuário, assegure-se de que a carga de trabalho do usuário padrão esteja acessível

- USAGE ON WORKLOAD SYSDEFAULTUSERWORKLOAD

Para executar consultas, é necessário criar um espaço de tabela temporário do sistema com tamanho de página de 8K, 16K e 32K, se não houver. Para criar os espaços de tabela temporários, execute os scripts a seguir.

```
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;
```

Como se conectar ao Db2 LUW como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Db2 LUW com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do Db2 LUW

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Db2 LUW e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados de origem do IBM Db2 LUW, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Digite o nome do banco de dados Db2 LUW.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha</p>

Parâmetro	Ação
	<p>por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção se você quiser usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Db2 LUW	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Como converter Db2 LUW em Amazon RDS para PostgreSQL ou Amazon Aurora Edição Compatível com PostgreSQL

Quando o IBM Db2 LUW é migrado para o PostgreSQL, a AWS SCT pode converter várias instruções de trigger usadas com o Db2 LUW. Essas declarações de trigger incluem:

- Eventos de trigger – Os eventos de trigger INSERT, DELETE e UPDATE especificam que a ação acionada é executada sempre que o evento é aplicado à tabela do assunto ou visualização do assunto. É possível especificar qualquer combinação dos eventos INSERT, DELETE e UPDATE, mas você pode especificar cada evento apenas uma vez. A AWS SCT oferece suporte para eventos de trigger único e múltiplo. O PostgreSQL tem praticamente a mesma funcionalidade para eventos.
- Evento OF COLUMN – Você pode especificar um nome de coluna de uma tabela-base. O trigger é ativado apenas pela atualização de uma coluna identificada na lista de nomes de colunas. O PostgreSQL tem a mesma funcionalidade.
- Triggers de declarações – Especificam que a ação acionada é aplicada somente uma vez para toda a declaração. Não é possível especificar esse tipo de granularidade de trigger para um trigger BEFORE ou um trigger INSTEAD OF. Se especificado, um trigger UPDATE ou DELETE será ativado, mesmo que nenhuma linha seja afetada. O PostgreSQL também tem essa funcionalidade, e a declarações de trigger do PostgreSQL é idêntica à da Db2 LUW.
- Cláusulas de referência – Especificam os nomes de correlações de variáveis de transição e os nomes de tabelas de transição. Os nomes de correlações identificam uma determinada linha no conjunto de linhas que foi afetada pela operação SQL de trigger. Os nomes de tabelas identificam o conjunto completo de linhas afetadas. Cada linha afetada por uma operação SQL de trigger está disponível para a ação acionada por meio da qualificação de colunas com nomes de correlação especificados. O PostgreSQL não é compatível com essa funcionalidade e usa apenas um nome de correlação NEW ou OLD.
- Triggers INSTEAD OF – a AWS SCT oferece suporte a eles.

Como converter tabelas particionadas do Db2 LUW em tabelas particionadas do PostgreSQL versão 10

A AWS SCT pode converter tabelas do Db2 LUW em tabelas particionadas em PostgreSQL 10. Há algumas restrições ao converter uma tabela particionada do Db2 LUW em PostgreSQL:

- Você pode criar uma tabela particionada com uma coluna anulável em Db2 LUW e especificar uma partição para armazenar os valores NULL. No entanto, o PostgreSQL não é compatível com valores NULL para particionamento RANGE.
- O Db2 LUW pode usar uma cláusula INCLUSIVE ou EXCLUSIVE para definir valores limite de intervalo. O PostgreSQL só é compatível com INCLUSIVE para um limite inicial e EXCLUSIVE para um limite final. O nome da partição convertida está no formato <original_table_name>_<original_partition_name>.
- É possível criar chaves primárias ou exclusivas para tabelas particionadas em Db2 LUW. O PostgreSQL exige que você crie chaves primárias ou exclusivas para cada partição diretamente. As restrições da chave primária ou exclusiva devem ser removidas da tabela pai. O nome da chave convertida está no formato <original_key_name>_<original_partition_name>.
- É possível criar uma restrição de chave estrangeira de e para uma tabela particionada em Db2 LUW. No entanto, o PostgreSQL não é compatível com referências de chaves estrangeiras em tabelas particionadas. O PostgreSQL também não é compatível com as referências de chave estrangeira de uma tabela particionada para outra tabela.
- Você pode criar um índice em uma tabela particionada em Db2 LUW. No entanto, o PostgreSQL exige que você crie um índice para cada partição diretamente. Os índices devem ser removidos da tabela pai. O nome do índice convertido está no formato <original_index_name>_<original_partition_name>.
- Você deve definir os acionadores de linha em partições individuais, e não na tabela particionada. Os acionadores devem ser removidos da tabela pai. O nome do acionador convertido está no formato <original_trigger_name>_<original_partition_name>.

Privilégios do PostgreSQL como destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio CREATE ON DATABASE. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para "\$user", public_synonyms, public.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Como converter Db2 LUW para Amazon RDS para MySQL ou Amazon Aurora MySQL.

Ao converter um banco de dados IBM Db2 LUW em RDS para MySQL ou Amazon Aurora MySQL, esteja ciente do seguinte.

Privilégios do MySQL como um destino

Os privilégios obrigatórios para MySQL como destino são listados a seguir:

- 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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como 1. Nesse caso, você não precisa usar as características DETERMINISTIC, READS SQL DATA ou NO SQL para criar funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Uso do MySQL como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código de banco de dados e o código de aplicativo do MySQL para os seguintes destinos:

- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon RDS para MySQL

Para obter mais informações, consulte as seções a seguir:

Tópicos

- [Privilégios para MySQL como um banco de dados de origem](#)
- [Conectar-se ao MySQL como origem](#)
- [Privilégios do PostgreSQL como um banco de dados de destino](#)

Privilégios para MySQL como um banco de dados de origem

Os privilégios obrigatórios do MySQL como origem são listados a seguir:

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

Conectar-se ao MySQL como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do MySQL com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do MySQL

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha MySQL e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.

4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
- Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em AWS Secret, escolha o nome do segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do MySQL, use as instruções a seguir:

Parâmetro	Ação
Nome do servidor	<p>Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.</p> <p>Você pode se conectar ao seu banco de dados do MySQL usando um protocolo de endereço IPv6. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.

Parâmetro	Ação
Nome de usuário e Senha	<p data-bbox="656 226 1474 310">Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p data-bbox="656 352 1507 772">A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p data-bbox="656 814 1497 949">Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul data-bbox="656 970 1503 1054" style="list-style-type: none"><li data-bbox="656 970 1503 1054">• Exigir SSL: escolha essa opção se deseja conectar-se ao servidor somente por meio de SSL. <p data-bbox="688 1096 1481 1423">Se você escolher Exigir SSL, isso significa que se o servidor não oferece suporte a SSL, não será possível se conectar ao servidor. Se não escolher Exigir SSL e o servidor não oferece suporte a SSL, você ainda poderá se conectar ao servidor sem usar SSL. Para obter mais informações, consulte Configurar o MySQL para usar conexões seguras.</p> <ul data-bbox="656 1444 1464 1684" style="list-style-type: none"><li data-bbox="656 1444 1464 1579">• Verificar certificado do servidor: selecione essa opção para verificar o certificado do servidor, usando um armazenamento confiável.<li data-bbox="656 1600 1464 1684">• Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados.

Parâmetro	Ação
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Habilitar essa opção permite que você armazene a senha do banco de dados e se conecte rapidamente a ele sem a necessidade de informar a senha.
Caminho do driver MySQL	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio `CREATE ON DATABASE`. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para `"$user", public_synonyms, public`.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```


No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Usar um banco de dados Oracle como origem do AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código de banco de dados e o código de aplicativo do Oracle Database para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon RDS para Oracle
- Amazon RDS para MariaDB

Quando a fonte é um banco de dados Oracle, os comentários podem ser convertidos para o formato apropriado, por exemplo, em um banco de dados PostgreSQL. AWS SCT pode converter comentários em tabelas, visualizações e colunas. Os comentários podem incluir apóstrofes, a AWS SCT dobra os apóstrofes ao converter instruções SQL, assim como faz para literais de string.

Para obter mais informações, consulte:

Tópicos

- [Privilégios do Oracle como origem](#)
- [Conectar-se ao Oracle como origem](#)
- [Converter Oracle em Amazon RDS para PostgreSQL ou Amazon Aurora \(PostgreSQL\)](#)

- [Converter Oracle em Amazon RDS para MySQL ou em Amazon Aurora MySQL](#)
- [Converter o Oracle para Amazon RDS para Oracle](#)

Privilégios do Oracle como origem

Os privilégios obrigatórios para Oracle como origem são listados a seguir:

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

Conectar-se ao Oracle como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Oracle com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do Oracle

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Oracle e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em AWS Secret, escolha o nome do segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do Oracle, use as instruções a seguir:

Parâmetro	Ação
Type	<p>Escolha o tipo de conexão ao seu banco de dados. Dependendo do tipo, forneça as informações adicionais seguintes:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Nome do servidor: o nome Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.• Porta do servidor: a porta usada para se conectar ao servidor de banco de dados de origem.• Oracle SID: o ID do sistema da Oracle (SID). Para encontrar o Oracle SID, envie a consulta a seguir para seu banco de dados Oracle: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>• Nome do serviço<ul style="list-style-type: none">• Nome do servidor: o nome DNS ou o endereço IP do servidor de banco de dados de origem. Você pode se conectar ao seu banco de dados do Oracle usando um protocolo de endereço IPv6. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir. <div data-bbox="716 1398 1507 1478" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>• Porta do servidor: a porta usada para se conectar ao servidor de banco de dados de origem.• Nome do serviço: o nome do serviço Oracle ao qual se conectar.• Alias do TNS

Parâmetro	Ação
	<ul style="list-style-type: none">• Caminho do arquivo TNS: o caminho para o arquivo que contém as informações de conexão do nome Transparent Network Substrate (TNS). <p>Depois que você escolheu o arquivo TNS, a AWS SCT adiciona todas as conexões do banco de dados Oracle do arquivo à lista de aliases do TNS.</p> <p>Escolha essa opção para se conectar ao Oracle Real Application Clusters (RAC).</p> <ul style="list-style-type: none">• Alias do TNS: o alias do TNS do arquivo a ser usado para a conexão com o banco de dados de origem.• Identificador de conexão TNS• Identificador de conexão TNS: o identificador das informações de conexão TNS registradas.

Parâmetro	Ação
Nome de usuário e Senha	<p data-bbox="656 226 1474 310">Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p data-bbox="656 352 1474 814">Na primeira vez que você se conecta ao banco de dados Oracle, precisa inserir o caminho para o arquivo Oracle Driver (ojdbc8.jar). Você pode fazer download do arquivo em http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. Certifique-se de se registrar no site gratuito da Oracle Technical Network para concluir o download. AWS SCT usa o driver selecionado para qualquer conexão futura do banco de dados Oracle. O caminho do driver pode ser modificado usando a guia Drivers em Configurações globais.</p> <p data-bbox="656 856 1507 1276">A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>

Parâmetro	Ação
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Autenticação SSL: selecione essa opção para usar a autenticação SSL por certificado. Configure seu armazenamento confiável e armazenamento de chaves em Configurações, Configurações globais, Segurança. • Armazenamento confiável: o armazenamento confiável a ser usado. • Armazenamento de chaves: o armazenamento de chaves a ser usado.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Habilitar essa opção permite que você armazene a senha do banco de dados e se conecte rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Oracle	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Converter Oracle em Amazon RDS para PostgreSQL ou Amazon Aurora (PostgreSQL)

Ao converter um banco de dados Oracle para RDS para PostgreSQL ou Amazon Aurora PostgreSQL esteja ciente do seguinte.

Tópicos

- [Privilégios do PostgreSQL como um banco de dados de destino](#)
- [Configurações da conversão do Oracle para o PostgreSQL](#)
- [Converter sequências Oracle](#)
- [Converter Oracle ROWID](#)
- [Converter o SQL dinâmico do Oracle](#)
- [Conversão de partições Oracle](#)

Ao converter objetos do sistema Oracle para PostgreSQL, AWS SCT executa conversões conforme mostrado na tabela a seguir.

Objeto do sistema do Oracle	Descrição	Objeto PostgreSQL convertido
V\$VERSION	Exibe os números de versão dos componentes da biblioteca principal no banco de dados Oracle	aws_oracle_ext.v\$version
V\$INSTANCE	Uma exibição que mostra o estado da instância atual.	aws_oracle_ext.v\$instance

Você pode usar AWS SCT para converter arquivos Oracle SQL*Plus em psql, que é um front-end baseado em terminal para o PostgreSQL. Para obter mais informações, consulte [Como converter o aplicativo SQL usando a AWS SCT](#).

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio `CREATE ON DATABASE`. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para "\$user", public_synonyms, public.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para PostgreSQL como destino, a AWS SCT necessita do privilégio `rds_superuser`.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Configurações da conversão do Oracle para o PostgreSQL

Para editar as configurações de conversão do Oracle para PostgreSQL, escolha Configurações e, em seguida, escolha AWS SCT, depois Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle – PostgreSQL. A AWS SCT exibe todas as configurações disponíveis para conversão do Oracle para o PostgreSQL.

As configurações de conversão do Oracle para PostgreSQL em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para permitir que AWS SCT converta visões materializadas do Oracle em tabelas ou visões materializadas no PostgreSQL. Para Conversão de visão materializada em, escolha como converter as visões materializadas de origem.
- Trabalhar com o código do Oracle de origem quando ele incluir os perfis TO_CHAR, TO_DATE e TO_NUMBER com parâmetros que não são compatíveis com o PostgreSQL. Por padrão, a AWS SCT emula a utilização desses parâmetros no código convertido.

Quando seu código do Oracle de origem inclui somente parâmetros compatíveis com o PostgreSQL, é possível utilizar perfis nativos do PostgreSQL, TO_CHAR, TO_DATE e TO_NUMBER. Nesse caso, o código convertido funciona mais rápido. Para incluir somente esses parâmetros, selecione os seguintes valores:

- A função TO_CHAR() não usa strings de formatação específicas do Oracle
- A função TO_DATE() não usa strings de formatação específicas do Oracle
- A função TO_NUMBER() não usa strings de formatação específicas do Oracle
- Para resolver quando seu banco de dados Oracle de origem armazena somente valores inteiros nas colunas de chave primária ou estrangeira do tipo de dados NUMBER, AWS SCT pode converter essas colunas no tipo de dados BIGINT. Essa abordagem melhorará o desempenho do código convertido. Para adotar essa abordagem, selecione Converter colunas de chave primária/ estrangeira NUMBER em colunas BIGINT. Verifique se a origem não inclui valores de ponto flutuante nessas colunas para evitar perda de dados.
- Ignorar gatilhos e restrições desativados em seu código-fonte. Para fazer isso, escolha Ignorar triggers e restrições desabilitados.
- Para usar AWS SCT para converter variáveis de string chamadas de SQL dinâmico. O código de banco de dados pode alterar os valores dessas variáveis de string. Para garantir que o AWS SCT sempre converta o valor mais recente dessa variável de string, selecione Converter o código SQL dinâmico criado nas rotinas chamadas.
- Para saber se as versões 10 e anteriores do PostgreSQL não são compatíveis com procedimentos. Se você não estiver familiarizado com a utilização de procedimentos no PostgreSQL, o AWS SCT poderá converter procedimentos do Oracle em perfis do PostgreSQL. Para fazer isso, selecione Converter procedimentos em perfis.

- Ver informações adicionais sobre os itens de ação ocorridos. Para fazer isso, você pode adicionar funções específicas ao pacote de extensões selecionando Adicionar um bloco de aumento de exceções para problemas de migração com os próximos níveis de gravidade. Escolha os níveis de gravidade para aumentar as exceções definidas pelo usuário.
- Trabalhar com um banco de dados Oracle de origem que pode incluir restrições com nomes gerados automaticamente. Se o código-fonte utilizar esses nomes, certifique-se de selecionar Converter os nomes das restrições geradas pelo sistema usando os nomes de origem. Se o código-fonte utilizar essas restrições, mas não utilizar seus nomes, desmarque essa opção para aumentar a velocidade da conversão.
- Para saber se o banco de dados e as aplicações são executados em fusos horários diferentes. Por padrão, AWS SCT emula os fusos horários no código convertido. No entanto, essa emulação não é necessária quando o banco de dados e as aplicações utilizam o mesmo fuso horário. Nesse caso, selecione O fuso horário no lado do cliente corresponde ao fuso horário no servidor.
- Para saber se o banco de dados da origem e do destino são executados em fusos horários diferentes. Se eles forem executados em fusos horários diferentes, o perfil que emula o perfil do Oracle integrado SYSDATE retornará valores diferentes em comparação com o perfil de origem. Para garantir que os perfis de origem e de destino retornem os mesmos valores, escolha Definir o fuso horário padrão para a emulação SYSDATE.
- Utilizar os perfis da extensão orafce no código convertido. Para fazer isso, em Usar implementação do orafce, selecione as funções a serem usadas. Para obter mais informações sobre a extensão, consulte o [orafce](#) no GitHub.

Converter sequências Oracle

AWS SCT converte sequências do Oracle para o PostgreSQL. Se você usar sequências para manter as restrições de integridade, certifique-se de que os novos valores de uma sequência migrada não se sobreponham aos valores existentes.

Para preencher sequências convertidas com o último valor do banco de dados de origem

1. Abra seu projeto AWS SCT com a Oracle como origem.
2. Selecione Configurações e, em seguida, escolha Configurações de conversão.
3. Na lista superior, escolha Oracle e, em seguida, escolha Oracle – PostgreSQL. A AWS SCT exibe todas as configurações disponíveis para conversão do Oracle para o PostgreSQL.
4. Escolha Preencher sequências convertidas com o último valor gerado no lado da fonte.

5. Escolha OK para salvar as configurações e fechar a caixa de diálogo Configurações de conversão.

Converter Oracle ROWID

Em um banco de dados da Oracle, a pseudocoluna ROWID contém o endereço da linha da tabela. A pseudocoluna ROWID é exclusiva da Oracle. Portanto, o AWS SCT converte a pseudocoluna ROWID em uma coluna de dados no PostgreSQL. Ao usar essa conversão, você pode manter as informações do ROWID.

Ao converter uma pseudocoluna ROWID, o AWS SCT pode criar uma coluna de dados com o tipo de dados `bigint`. Se não houver nenhuma chave primária, o AWS SCT definirá a coluna ROWID como a chave primária. Se houver uma chave primária, o AWS SCT definirá a coluna ROWID com uma restrição exclusiva.

Se o código-fonte do banco de dados incluir operações com ROWID, que você não pode executar usando um tipo de dados numérico, AWS SCT pode criar uma coluna de dados com o tipo de dados `character varying`.

Para criar uma coluna de dados para a Oracle ROWID em um projeto

1. Abra seu projeto AWS SCT com a Oracle como origem.
2. Selecione Configurações e, em seguida, escolha Configurações de conversão.
3. Na lista superior, escolha Oracle e, em seguida, escolha Oracle – PostgreSQL. A AWS SCT exibe todas as configurações disponíveis para conversão do Oracle para o PostgreSQL.
4. Em Gerar ID de linha, siga um destes procedimentos:
 - Escolha Gerar como identidade para criar uma coluna de dados numéricos.
 - Escolha Gerar como tipo de domínio de caracteres para criar uma coluna de dados de caracteres.
5. Escolha OK para salvar as configurações e fechar a caixa de diálogo Configurações de conversão.

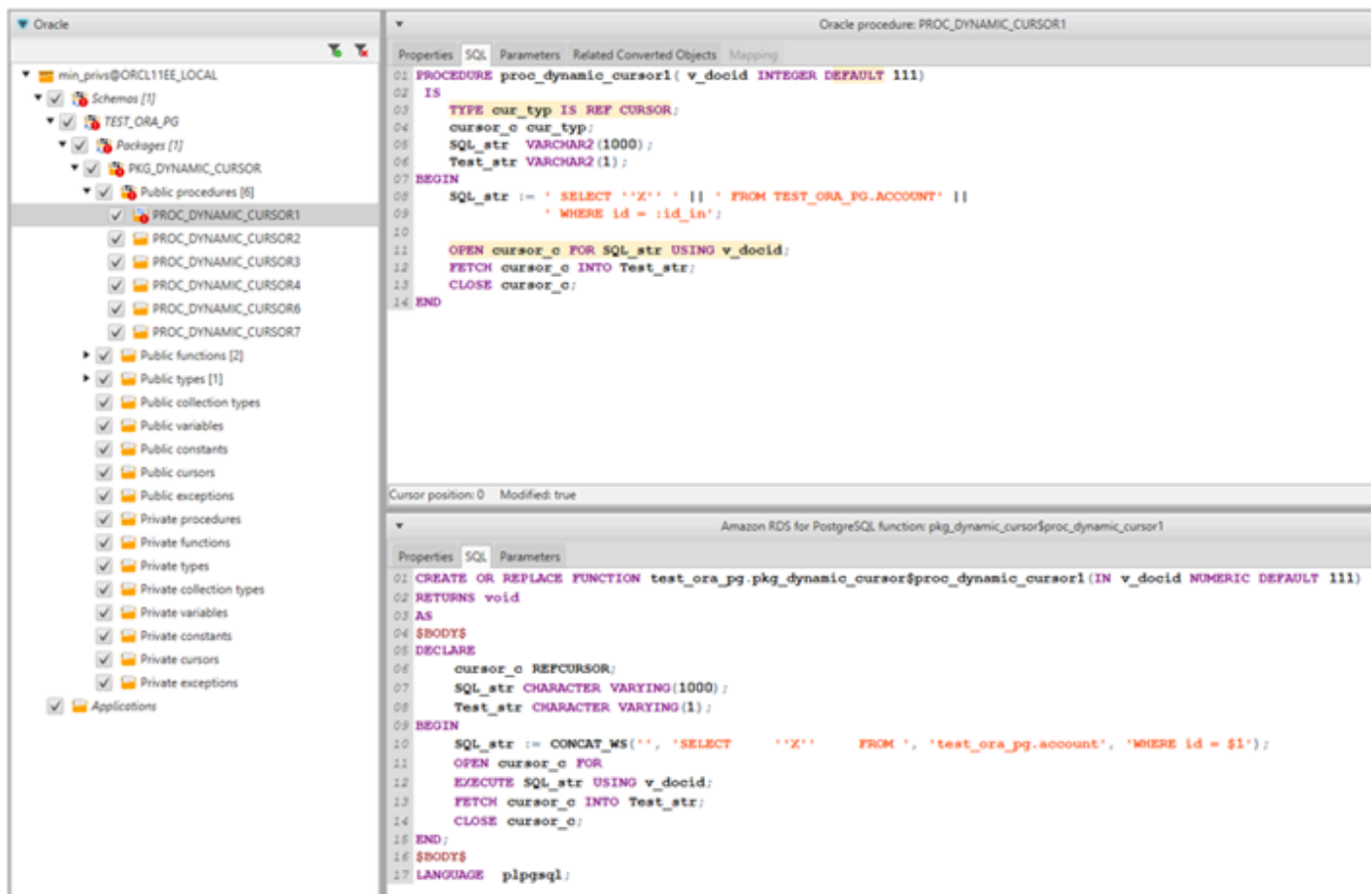
Converter o SQL dinâmico do Oracle

A Oracle fornece duas maneiras de implementar SQL dinâmico: usando uma instrução EXECUTE IMMEDIATE ou chamando procedimentos no pacote DBMS_SQL. Se seu banco de dados Oracle de

origem incluir objetos com SQL dinâmico, use AWS SCT para converter instruções SQL dinâmicas Oracle em PostgreSQL.

Para converter SQL dinâmico Oracle para SQL PostgreSQL

1. Abra seu projeto AWS SCT com a Oracle como origem.
2. Escolha um objeto de banco de dados que use SQL dinâmico na exibição em árvore de origem da Oracle.
3. Abra o menu de contexto (clique com o botão direito do mouse) do objeto, escolha Converter esquema e aceite substituir os objetos se eles existirem. A captura de tela a seguir mostra o procedimento convertido abaixo do procedimento Oracle com SQL dinâmico.



Conversão de partições Oracle

AWS SCT atualmente é compatível com os seguintes métodos de particionamento:

- Intervalo
- Lista

- Intervalo de várias colunas
- Hash
- Composto (lista de lista, lista de intervalos, intervalo de lista, hash de lista, hash de intervalo, hash de hash)

Converter Oracle em Amazon RDS para MySQL ou em Amazon Aurora MySQL

Para emular funções de banco de dados do Oracle em seu código MySQL convertido, use o pacote de extensão Oracle para MySQL em AWS SCT. Para obter mais informações sobre pacotes de extensão, consulte [Usando AWS SCT pacotes de extensão](#).

Tópicos

- [Privilégios do MySQL como um banco de dados de destino](#)
- [Configurações de conversão do Oracle para MySQL](#)
- [Considerações sobre a migração](#)
- [Converter a instrução WITH na Oracle para o RDS para MySQL ou para Amazon Aurora MySQL](#)

Privilégios do MySQL como um banco de dados de destino

Os privilégios obrigatórios para MySQL como destino são listados a seguir:

- 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.*

Se você usar um banco de dados MySQL versão 5.7 ou inferior como destino, conceda a permissão `INVOKE LAMBDA *.*` em vez de `AWS_LAMBDA_ACCESS`. Para bancos de dados MySQL versão 8.0 e superior, conceda a permissão `AWS_LAMBDA_ACCESS`.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua `user_name` pelo nome do seu usuário. Em seguida, substitua `your_password` por uma senha segura.

Se você usar um banco de dados MySQL versão 5.7 ou inferior como destino, use `GRANT INVOKE LAMBDA ON *.* TO 'user_name'` em vez de `GRANT AWS_LAMBDA_ACCESS TO 'user_name'`.

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula

identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como 1. Nesse caso, você não precisa usar as características `DETERMINISTIC`, `READS SQL DATA` ou `NO SQL` para criar funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Configurações de conversão do Oracle para MySQL

Para editar as configurações de conversão do Oracle para MySQL, escolha Configurações e, em seguida, escolha Configurações em AWS SCT, depois Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle – MySQL. A AWS SCT exibe todas as configurações disponíveis para conversão do Oracle para MySQL.

As configurações de conversão do Oracle para MySQL em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para resolver isso, seu banco de dados Oracle de origem pode usar a pseudocoluna `R0WID`, mas o MySQL não oferece suporte a funcionalidades semelhantes. AWS SCT pode emular a pseudocoluna `R0WID` no código convertido. Para fazer isso, escolha Gerar como identidade em Gerar ID de linha?.

Se o código do Oracle de origem não utilizar a pseudocoluna `R0WID`, escolha Não gerar em Gerar ID de linha? Nesse caso, o código convertido funciona mais rápido.

- Trabalhar com o código do Oracle de origem quando ele incluir os perfis `T0_CHAR`, `T0_DATE` e `T0_NUMBER` com parâmetros que não são compatíveis com o MySQL. Por padrão, a AWS SCT emula a utilização desses parâmetros no código convertido.

Quando seu código do Oracle de origem inclui somente parâmetros compatíveis com o PostgreSQL, é possível utilizar os perfis nativos `T0_CHAR`, `T0_DATE` e `T0_NUMBER` do MySQL.

Nesse caso, o código convertido funciona mais rápido. Para incluir somente esses parâmetros, selecione os seguintes valores:

- A função TO_CHAR() não usa strings de formatação específicas do Oracle
- A função TO_DATE() não usa strings de formatação específicas do Oracle
- A função TO_NUMBER() não usa strings de formatação específicas do Oracle
- Para saber se o banco de dados e as aplicações são executados em fusos horários diferentes. Por padrão, AWS SCT emula os fusos horários no código convertido. No entanto, essa emulação não é necessária quando o banco de dados e as aplicações utilizam o mesmo fuso horário. Nesse caso, selecione O fuso horário no lado do cliente corresponde ao fuso horário no servidor.

Considerações sobre a migração

Ao converter Oracle em RDS para MySQL ou para Aurora MySQL, para alterar a ordem em que as instruções são executadas, você pode usar uma instrução GOTO e um rótulo. Qualquer instrução PL/SQL que segue uma instrução GOTO é ignorada, e o processamento continua no rótulo. As instruções GOTO e os rótulos podem ser usados em qualquer lugar em um procedimento, lote ou bloco de instruções. Você também pode usar as instruções GOTO depois.

O MySQL não usa instruções GOTO. Quando a AWS SCT converte o código que contém uma instrução GOTO, ela converte a instrução para usar uma instrução BEGIN...END ou LOOP...END LOOP.

Você pode encontrar exemplos de como a AWS SCT converte instruções GOTO na tabela a seguir.

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

Instrução do Oracle	Instrução do MySQL
<pre>BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END</pre>
<pre>BEGIN statement1; label1: statement2; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END</pre>

Converter a instrução WITH na Oracle para o RDS para MySQL ou para Amazon Aurora MySQL

Use a cláusula WITH (subquery_factoring) na Oracle para atribuir um nome (query_name) a um bloco de subconsulta. Em seguida, você poderá fazer referência ao bloco de subconsulta em vários lugares na consulta especificando query_name. Se um bloco de subconsulta não contiver links ou parâmetros (local, procedimento, função, pacote), a AWS SCT converterá a cláusula em uma visualização ou uma tabela temporária.

A vantagem de converter a cláusula em uma tabela temporária é que referências repetidas à subconsulta podem ser mais eficientes. A maior eficiência ocorre porque os dados são facilmente recuperados da tabela temporária em vez de serem solicitados por cada referência. Isso pode ser emulado com visualizações adicionais ou uma tabela temporária. O nome da visualização usa o formato <procedure_name>\${subselect_alias}.

Você pode encontrar exemplos na tabela a seguir.

Instrução do Oracle	Instrução do MySQL
<pre>CREATE PROCEDURE TEST_ORA_PG.P_WITH_SELECT_V ARIABLE_01 (p_state IN NUMBER) AS l_dept_id NUMBER := 1; BEGIN FOR cur IN (WITH dept_empl(id, name, surname, lastname, state, dept_id) AS (SELECT id, name, surname, lastname, state, dept_id FROM test_ora_ pg.dept_employees WHERE state = p_state AND</pre>	<pre>CREATE PROCEDURE test_ora_pg.P_WITH _SELECT_VARIABLE_01(IN par_P_STATE DOUBLE) BEGIN DECLARE var_l_dept_id DOUBLE DEFAULT 1; DECLARE var\$id VARCHAR (8000); DECLARE var\$state VARCHAR (8000); DECLARE done INT DEFAULT FALSE; DECLARE cur CURSOR FOR SELECT ID, STATE FROM (SELECT ID, NAME, SURNAME, LASTNAME, STATE, DEPT_ID FROM TEST_ORA_PG.DEPT_E MPLOYEES WHERE STATE = par_p_sta te AND DEPT_ID = var_l_dept_id) AS dept_empl ORDER BY ID; DECLARE CONTINUE HANDLER FOR NOT FOUND</pre>

Instrução do Oracle

```
                dept_id =  
l_dept_id)  
        SELECT id,state  
        FROM dept_emp1  
        ORDER BY id) LOOP  
  
        NULL;  
END LOOP;
```

Instrução do MySQL

```
        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;
```

Instrução do Oracle

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_R
  EGULAR_MULT_01
AS
BEGIN

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

```

Instrução do MySQL

```

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

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

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

```

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

Instrução do Oracle

```

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

```

Instrução do MySQL

```

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

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

  read_label:

```

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

Converter o Oracle para Amazon RDS para Oracle

Há alguns aspectos a serem considerados ao migrar o esquema e o código do Oracle para o Amazon RDS para Oracle:

- A AWS SCT pode adicionar objetos de diretório à árvore de objetos. Os objetos do diretório são estruturas lógicas que representam um diretório físico no sistema de arquivos do servidor. Você pode usar objetos de diretório com pacotes, como DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, o utilitário DATAPUMP, etc.
- A AWS SCT é compatível com a conversão de tablespaces do Oracle para uma instância de banco de dados do Amazon RDS para Oracle. O Oracle armazena dados logicamente em tablespaces e fisicamente em arquivos de dados associados ao tablespace correspondente. No Oracle é possível criar tablespace com nomes de arquivo de dados. O Amazon RDS só oferece suporte a arquivos gerenciados Oracle (OMF), arquivos de dados, arquivos de log e arquivos de controle. A AWS SCT cria os arquivos de dados necessários durante a conversão.
- A AWS SCT pode converter funções e privilégios no nível de servidor. O mecanismo de banco de dados Oracle usa a segurança baseada em funções. Uma função é uma coleção de privilégios que podem ser concedidos ou revogados de um usuário. Uma função predefinida no Amazon RDS, chamada de administrador de banco de dados, normalmente permite todos os privilégios administrativos em um mecanismo de banco de dados Oracle. Os seguintes privilégios não estão disponíveis para a função DBA em uma instância de banco de dados do Amazon RDS usando o mecanismo Oracle:

- Alter database
- Alterar o sistema
- Criar qualquer diretório
- Conceder qualquer privilégio
- Conceder qualquer função
- Criar trabalho externo

Você pode conceder todos os outros privilégios a uma função de usuário do Amazon RDS para Oracle, incluindo filtragem avançada e privilégios de coluna.

- A AWS SCT oferece suporte à conversão de trabalhos do Oracle em trabalhos que podem ser executados no Amazon RDS para Oracle. Há algumas limitações à conversão, incluindo as seguintes:
 - Trabalhos executáveis não são suportados.
 - Trabalhos de programação que usam o tipo de dados ANYDATA como argumento não são suportados.
- O Oracle Real Application Clusters (RAC) One Node é uma opção ao Oracle Database Enterprise Edition que foi introduzida com o Oracle Database 11g Release 2. O Amazon RDS para Oracle não é compatível com o atributo RAC. Para obter alta disponibilidade, use o Multi-AZ do Amazon RDS.

Em uma implantação Multi-AZ, o Amazon RDS automaticamente provisiona e mantém uma réplica em espera síncrona em outra Zona de disponibilidade. A instância de banco de dados primária é sincronicamente replicada ao longo das zonas de disponibilidade para uma réplica em espera. Essa funcionalidade fornece redundância de dados, elimina congelamentos de E/S e minimiza picos de latência durante backups do sistema.

- O Oracle Spatial fornece um esquema SQL e funções que facilitam o armazenamento, a recuperação, a atualização e a consulta de coleções de dados espaciais em um banco de dados Oracle. O Oracle Locator oferece recursos normalmente obrigatórios para dar suporte à Internet e a aplicativos baseados em serviço sem fio além de soluções GIS baseadas no parceiro. O Oracle Locator é um subconjunto limitado do Oracle Spatial.

Para usar os recursos do Oracle Spatial e do Oracle Locator, adicione a opção SPATIAL ou LOCATOR (mutuamente exclusivas) ao grupo de opções da instância de banco de dados.

Há alguns pré-requisitos para usar o Oracle Spatial e o Oracle Locator em uma instância de banco de dados do Amazon RDS para Oracle:

- A instância deve usar o Oracle Enterprise Edition versão 12.1.0.2.v6 ou posterior ou 11.2.0.4.v10 ou posterior.
- A instância deve estar em uma nuvem privada virtual (VPC).
- A instância deve ser a classe de instância de banco de dados que pode oferecer suporte ao recurso da Oracle. Por exemplo, o Oracle Spatial não é compatível com as classes de instância de banco de dados db.m1.small, db.t1.micro, db.t2.micro ou db.t2.small. Para obter mais informações, consulte [Suporte a classes de instância de banco de dados para Oracle](#).
- A instância deve ter a opção Atualização de versão do Auto Minor habilitada. O Amazon RDS atualizará a instância de banco de dados para o Oracle PSU mais recente se houver vulnerabilidades de segurança com uma pontuação CVSS de 9 ou mais ou outras vulnerabilidades de segurança anunciadas. Para obter mais informações, consulte

[Configurações para instâncias de banco de dados do Oracle](#).

- Se a instância de banco de dados for a versão 11.2.0.4 .v10 ou posterior, você deverá instalar a opção XMLDB. Para obter mais informações, consulte

[Oracle XML DB](#).

- Você deve ter uma licença do Oracle Spatial da Oracle. Para obter mais informações, consulte [Oracle Spatial and Graph](#) na documentação da Oracle.
- O Data Guard é incluído com o Oracle Database Enterprise Edition. Para obter alta disponibilidade, use o atributo Multi-AZ do Amazon RDS.

Em uma implantação Multi-AZ, o Amazon RDS automaticamente provisiona e mantém uma réplica em espera síncrona em outra Zona de disponibilidade. A instância de banco de dados primária é sincronicamente replicada ao longo das zonas de disponibilidade para uma réplica em espera. Essa funcionalidade fornece redundância de dados, elimina congelamentos de E/S e minimiza picos de latência durante backups do sistema.

- A AWS SCT oferece suporte à conversão de objetos DBMS_SCHEDULER do Oracle na migração para Amazon RDS para Oracle. O relatório de avaliação da AWS SCT indica se um objeto de programação pode ser convertido. Para obter mais informações sobre como usar objetos de programação com o Amazon RDS, consulte a [Documentação do Amazon RDS](#).
- Para conversões o Oracle para Amazon RDS para Oracle, os links de banco de dados são compatíveis. Um link de banco de dados é um objeto de esquema em um banco de dados que permite acessar objetos em outro banco de dados. O outro banco de dados não precisa ser um banco de dados Oracle. No entanto, para acessar bancos de dados que não sejam Oracle, é necessário usar o Oracle Heterogeneous Services.

Assim que criar um link de banco de dados, você pode usar o link em instruções SQL para consultar tabelas, visualizações e objetos PL/SQL em outro banco de dados. Para usar um link de banco de dados, anexe @dblink ao nome da tabela, da visualização ou do objeto PL/SQL. Você pode consultar uma tabela ou visualização em outro banco de dados com a instrução SELECT. Para obter mais informações sobre como usar links do banco de dados Oracle, consulte a [Documentação do Oracle](#).

Para obter mais informações sobre como usar links do banco de dados com o Amazon RDS, consulte a [Documentação do Amazon RDS](#).

- O relatório de avaliação da AWS SCT fornece métricas do servidor para a conversão. Essas métricas sobre sua instância da Oracle incluem:
 - Capacidade de computação e memória da instância de banco de dados de destino.
 - Recursos da Oracle sem suporte, como o Real Application Clusters, aos quais o Amazon RDS não oferece suporte.
 - Carga de leitura/gravação de disco
 - Taxa de throughput total
 - Informações do servidor, como nome do servidor, sistema operacional, nome do host e conjunto de caracteres.

Privilégios para RDS para Oracle como destino

Para migrar para o Amazon RDS para Oracle, crie um usuário de banco de dados privilegiado. Você pode usar o exemplo de código a seguir.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Limitações ao converter Oracle para Amazon RDS para Oracle

Você deve considerar algumas limitações ao migrar o esquema e o código da Oracle para o Amazon RDS para Oracle:

- Uma função predefinida no Amazon RDS, chamada de administrador de banco de dados, normalmente permite todos os privilégios administrativos em um mecanismo de banco de dados Oracle. Os seguintes privilégios não estão disponíveis para a função DBA em uma instância de banco de dados do Amazon RDS usando o mecanismo Oracle:
 - Alter database

- Alterar o sistema
- Criar qualquer diretório
- Conceder qualquer privilégio
- Conceder qualquer função
- Criar trabalho externo

Você pode conceder todos os outros privilégios a uma função de usuário do Oracle RDS.

- O Amazon RDS para Oracle oferece suporte a auditoria tradicional, auditoria de acesso refinado usando o pacote DBMS_FGA e Oracle Unified Auditing.
- O Amazon RDS para Oracle não oferece suporte à captura de dados de alterações (CDC). Para fazer o CDC durante e após a migração do banco de dados, use AWS Database Migration Service.

Usar o PostgreSQL como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código de banco de dados e o código de aplicativo do PostgreSQL para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL

Para obter mais informações, consulte as seções a seguir:

Tópicos

- [Privilégios para PostgreSQL como um banco de dados de origem](#)
- [Conectar-se ao PostgreSQL como origem](#)
- [Privilégios do MySQL como um banco de dados de destino](#)

Privilégios para PostgreSQL como um banco de dados de origem

Os privilégios obrigatórios para PostgreSQL como origem são listados a seguir:

- 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-se ao PostgreSQL como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do PostgreSQL com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem do PostgreSQL

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha PostgreSQL e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em AWS Secret, escolha o nome do segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do PostgreSQL, use as instruções a seguir:

Parâmetro	Ação
Nome do servidor	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem. Você pode se conectar ao seu banco de dados do PostgreSQL de origem usando um protocolo de endereço IPv6. Para fazer isso, certifique-se de usar colchetes para

Parâmetro	Ação
	<p>inserir o endereço IP, conforme mostrado no exemplo a seguir.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Digite o nome do banco de dados PostgreSQL.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none">• Verificar certificado do servidor: selecione essa opção para verificar o certificado do servidor, usando um armazenamento confiável.• Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. Para que esse local apareça na seção Configurações globais, certifique-se de adicioná-lo.

Parâmetro	Ação
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Habilitar essa opção permite que você armazene a senha do banco de dados e se conecte rapidamente a ele sem a necessidade de informar a senha.
Caminho do driver PostgreSQL	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Privilégios do MySQL como um banco de dados de destino

Os privilégios obrigatórios para MySQL como destino quando migrados do PostgreSQL são os seguintes:

- 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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como 1. Nesse caso, você não precisa usar as características DETERMINISTIC, READS SQL DATA ou NO SQL para criar

funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Usando o SAP ASE (Sybase ASE) como origem para a AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código de banco de dados e código de aplicativo do SAP (Sybase) Adaptive Server Enterprise (ASE) para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para MariaDB
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL

Para obter mais informações, consulte as seções a seguir:

Tópicos

- [Privilégios do SAP ASE como um banco de dados de origem](#)
- [Conectando-se ao SAP ASE \(Sybase\) como origem](#)
- [Privilégios do MySQL como um banco de dados de destino](#)
- [Configurações de conversão do SAP ASE para MySQL](#)
- [Privilégios do PostgreSQL como um banco de dados de destino](#)
- [Configurações de conversão de SAP ASE para PostgreSQL.](#)

Privilégios do SAP ASE como um banco de dados de origem

Para usar um banco de dados do SAP ASE como origem, você cria um usuário de banco de dados e concede permissões. Para fazer isso, siga as etapas a seguir.

Criar e configurar um usuário de banco de dados

1. Conectar-se ao banco de dados de origem.
2. Criar um usuário de banco de dados com os seguintes comandos. Forneça uma senha para o novo usuário.

```
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 banco de dados que você vai migrar, conceda os seguintes privilégios.

```
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
```

Conectando-se ao SAP ASE (Sybase) como origem

Use o procedimento a seguir para conectar seu banco de dados do SAP ASE de origem com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados do SAP ASE de origem

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha SAP ASE e, em seguida, escolha Avançar.


A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em AWS Secret, escolha o nome do segredo.

2. Escolha **Preencher** para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do SAP ASE de origem, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Server port	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Insira o nome do banco de dados do SAP ASE.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <div data-bbox="656 1087 1507 1688" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p></div>

Parâmetro	Ação
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Verificar certificado do servidor: selecione essa opção para verificar o certificado do servidor, usando um armazenamento confiável. • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Habilitar essa opção permite que você armazene a senha do banco de dados e se conecte rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver SAP ASE	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Privilégios do MySQL como um banco de dados de destino

Os privilégios obrigatórios para MySQL como destino são listados a seguir:

- 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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como 1. Nesse caso, você não precisa usar as características `DETERMINISTIC`, `READS SQL DATA` ou `NO SQL` para criar funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Configurações de conversão do SAP ASE para MySQL

Para editar as configurações de conversão do SAP ASE para MySQL, escolha Configurações e, em seguida, escolha Configurações de conversão. Na lista superior, escolha SAP ASE e, em seguida, escolha SAP ASE — MySQL ou SAP ASE — Amazon Aurora (compatível com MySQL). A AWS SCT exibe todas as configurações disponíveis para conversão de SAP ASE para PostgreSQL.

As configurações de conversão do SAP ASE para MySQL na AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para usar os nomes exatos dos objetos do banco de dados de origem no código convertido.

Por padrão, a AWS SCT converte os nomes dos objetos, variáveis e parâmetros do banco de dados em letras minúsculas. Para manter as maiúsculas e minúsculas originais desses nomes, selecione Tratar nomes de objetos do banco de dados de origem com distinção entre maiúsculas e minúsculas. Escolha essa opção se você usar nomes de objetos com distinção entre maiúsculas e minúsculas no servidor de banco de dados do SAP ASE de origem.

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio `CREATE ON DATABASE`. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para `"$user", public_synonyms, public`.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Configurações de conversão de SAP ASE para PostgreSQL.

Para editar as configurações de conversão do SAP ASE para PostgreSQL, escolha Configurações e, em seguida, escolha Configurações de conversão. Na lista superior, escolha SAP ASE e, em seguida, escolha SAP ASE – PostgreSQL ou SAP ASE – Amazon Aurora (compatível com PostgreSQL). A AWS SCT exibe todas as configurações disponíveis para conversão de SAP ASE para PostgreSQL.

As configurações de conversão do SAP ASE para PostgreSQL em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o modelo a ser usado para os nomes dos esquemas no código convertido. Para Modelo de geração de nome de esquema, escolha uma das opções a seguir:
 - <source_db> — Usa o nome do banco de dados SAP ASE como nome de esquema no PostgreSQL.
 - <source_schema> — Usa o nome do esquema SAP ASE como nome do esquema no PostgreSQL.
 - <source_db>_<schema> — Usa uma combinação do banco de dados SAP ASE e dos nomes do esquema como um nome de esquema no PostgreSQL.
- Para usar os nomes exatos dos objetos do banco de dados de origem no código convertido.

Por padrão, a AWS SCT converte os nomes dos objetos, variáveis e parâmetros do banco de dados em letras minúsculas. Para manter as maiúsculas e minúsculas originais desses nomes, selecione Tratar nomes de objetos do banco de dados de origem com distinção entre maiúsculas e minúsculas. Escolha essa opção se você usar nomes de objetos com distinção entre maiúsculas e minúsculas no servidor de banco de dados do SAP ASE de origem.

Para operações com distinção entre maiúsculas e minúsculas, a AWS SCT pode evitar a conversão de nomes de objetos do banco de dados em minúsculas. Para fazer isso, selecione Evitar conversão para minúsculas para operações com distinção entre maiúsculas e minúsculas.

- Permitir o uso de índices com o mesmo nome em tabelas diferentes no SAP ASE.

No PostgreSQL, todos os nomes de índice que você usa no esquema devem ser exclusivos. Para garantir que a AWS SCT gere nomes exclusivos para todos os seus índices, selecione Gerar nomes exclusivos para índices.

Usando o Microsoft SQL Server como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código de banco de dados e o código de aplicativo do SQL Server para os seguintes destinos:

- Amazon RDS para MySQL
- Amazon Aurora Edição Compatível com MySQL
- Amazon RDS para PostgreSQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon RDS para SQL Server
- Amazon RDS para MariaDB

Note

A AWS SCT não oferece suporte ao uso do Amazon RDS para SQL server como origem.

Você pode usar a AWS SCT para criar um relatório de avaliação para a migração de esquemas, objetos de código de banco de dados e código de aplicativo do SQL Server para o BabelFish para Aurora PostgreSQL, conforme descrito a seguir.

Tópicos

- [Privilégios do Microsoft SQL Server como origem](#)
- [Usando a autenticação do Windows ao usar o Microsoft SQL Server como origem](#)
- [Conectando-se ao SQL Server como origem](#)
- [Convertendo o SQL Server em MySQL](#)
- [Convertendo um SQL Server em PostgreSQL](#)
- [Converter SQL Server para o Amazon RDS for SQL Server](#)

Privilégios do Microsoft SQL Server como origem

Os privilégios obrigatórios para Microsoft SQL Server como origem são listados a seguir:

- VIEW DEFINITION
- VIEW DATABASE STATE

O privilégio VIEW DEFINITION permite que usuários com acesso público vejam as definições de objetos. A AWS SCT usa o privilégio VIEW DATABASE STATE para verificar os atributos da edição SQL Server Enterprise.

Repetir a concessão para cada banco de dados cujo esquema que você está convertendo.

Além disso, conceda os seguintes privilégios no banco de dados master:

- VIEW SERVER STATE
- VIEW ANY DEFINITION

A AWS SCT usa o privilégio VIEW SERVER STATE para coletar as definições e configurações do servidor. Certifique-se de conceder o privilégio VIEW ANY DEFINITION de visualizar endpoints.

Para ler as informações sobre o Microsoft Analysis Services, execute o comando a seguir no banco de dados master.

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

No exemplo anterior, substitua o espaço reservado <user_name> pelo nome do usuário que você concedeu com os privilégios anteriores.

Para ler informações sobre o SQL Server Agent, adicione seu usuário ao perfil SQLAgentUser. Execute o comando a seguir no banco de dados msdb.

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

No exemplo anterior, substitua o espaço reservado <SQLAgentRole> pelo nome do perfil do SQL Server Agent. Em seguida, substitua o espaço reservado <user_name> pelo nome do usuário que você concedeu com os privilégios anteriores. Para obter mais informações, consulte [Adicionar um usuário ao perfil SQLAgentUser](#) no Guia do usuário do Amazon RDS.

Para detectar o envio de logs, conceda o privilégio SELECT on dbo.log_shipping_primary_databases no banco de dados msdb.

Para usar a abordagem de notificação da replicação DDL, conceda o privilégio RECEIVE ON <schema_name>.<queue_name> em seus bancos de dados de origem. Neste exemplo, substitua o espaço reservado <schema_name> pelo nome do esquema do banco de dados. Substitua o espaço reservado <queue_name> pelo nome de uma tabela de filas.

Usando a autenticação do Windows ao usar o Microsoft SQL Server como origem

Se o seu aplicativo for executado em uma intranet baseada no Windows, você poderá usar a autenticação do Windows para acesso ao banco de dados. A autenticação do Windows usa a identidade atual do Windows estabelecida no thread do sistema operacional para acessar o banco de dados do SQL Server. Em seguida, você pode mapear a identidade do Windows para um banco de dados e para as permissões do SQL Server. Para se conectar ao SQL Server usando a autenticação do Windows, você deve especificar a identidade do Windows que seu aplicativo está usando. Também é necessário conceder o acesso de identidade do Windows ao banco de dados do SQL Server.

O SQL Server possui dois modos de acesso: modo de autenticação do Windows e modo misto. O modo de autenticação do Windows habilita a autenticação do Windows e desabilita a autenticação do SQL Server. O modo misto habilita a autenticação do Windows e a autenticação do SQL Server. A autenticação do Windows está sempre disponível e não pode ser desativada. Para obter mais informações sobre a autenticação do Windows, consulte a documentação do Microsoft Windows.

Um possível exemplo de criação de usuário no TEST_DB é mostrado abaixo.

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

Usando a autenticação do Windows com uma conexão JDBC

O driver JDBC não oferece suporte à autenticação do Windows quando o driver é usado em sistemas operacionais que não sejam Windows. Credenciais de autenticação do Windows, como nome de usuário e senha, não são automaticamente especificados ao se conectar ao SQL Server a partir de sistemas operacionais que não sejam Windows. Nesses casos, os aplicativos devem usar a autenticação do SQL Server.

Na string de conexão JDBC, o parâmetro `integratedSecurity` deve ser especificado para se conectar usando a autenticação do Windows. O driver JDBC oferece suporte à autenticação integrada do Windows em sistemas operacionais Windows por meio do parâmetro de string de conexão `integratedSecurity`.

Para usar a autenticação integrada.

1. Instalar o driver de JDBC.
2. Copie o arquivo `sqljdbc_auth.dll` para um diretório no caminho do sistema Windows no computador em que o driver JDBC está instalado.

Os arquivos `sqljdbc_auth.dll` são instalados no seguinte local:

```
<diretório de instalação>\sqljdbc_<versão>\<linguagem>\auth\
```

Ao tentar estabelecer uma conexão com o banco de dados do SQL Server usando a autenticação do Windows, você pode receber o erro: Este driver não está configurado para autenticação integrada. Esse problema pode ser resolvido executando as seguintes ações:

- Declare duas variáveis que indicam o caminho instalado do seu JDBC:

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu (onde seu sqljdbc4.jar existe);
```

```
variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu\auth\n86 (se estiver executando um sistema operacional de 32 bits) ou D\lib\JDBC4.1\enu\auth\n64 (se estiver executando um sistema operacional de 64 bits). É aqui que o seu sqljdbc_auth.dll está localizado.
```

- Copie `sqljdbc_auth.dll` na pasta onde o JDK/JRE está sendo executado. Você pode copiar na pasta `lib`, `bin`, etc. Por exemplo, você pode copiar na pasta a seguir.

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

- Certifique-se de que na sua pasta da biblioteca JDBC só tenha o arquivo `SQLJDBC4.jar`. Remova qualquer outro arquivo `sqljdbc*.jar` dessa pasta (ou copie em outra pasta). Se você estiver adicionando o driver como parte do programa, certifique-se de adicionar apenas o `SQLJDBC4.jar` como o driver a ser usado.
- Copie o arquivo `sqljdbc_auth.dll` na pasta com o aplicativo.

Note

Se você estiver executando uma Java Virtual Machine (JVM) de 32 bits, use o arquivo `sqljdbc_auth.dll` na pasta `x86`, mesmo se o sistema operacional for a versão `x64`. Se você estiver executando uma JVM de 64 bits em um processador `x64`, use o arquivo `sqljdbc_auth.dll` na pasta `x64`.

Ao se conectar a um banco de dados do SQL Server, você pode escolher a Autenticação do Windows ou a Autenticação do SQL Server para a opção Autenticação.

Conectando-se ao SQL Server como origem

Use o procedimento a seguir para se conectar ao banco de dados de origem do Microsoft SQL Server com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem do Microsoft SQL Server

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Microsoft SQL Server e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em AWS Secret, escolha o nome do segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados de origem do Microsoft SQL Server, use as instruções a seguir:

Parâmetro	Ação
Server name	<p>Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.</p> <p>Você pode se conectar ao seu banco de dados do SQL Server de origem usando um protocolo de endereço IPv6. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Server port	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Nome da instância	Digite o nome da instância do banco de dados SQL Server. Para encontrar o nome da instância, execute a consulta <code>SELECT @@servername;</code> em seu banco de dados SQL Server.
Autenticação	Escolha o tipo de autenticação em Autenticação do Windows e Autenticação do SQL Server.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>

Parâmetro	Ação
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Certificado de servidor confiável: selecione essa opção para confiar no certificado de servidor. • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. Para que esse local apareça na seção Configurações globais, certifique-se de adicioná-lo.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Habilitar essa opção permite que você armazene a senha do banco de dados e se conecte rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Sql Server	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>
Biblioteca de autenticação do Windows	<p>Insira o caminho para o arquivo sqljdbc_auth.dll . Por padrão, esse arquivo é instalado no seguinte local:</p> <pre><installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\</pre>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Convertendo o SQL Server em MySQL

Para emular funções de banco de dados do Microsoft SQL Server em seu código MySQL convertido, use o pacote de extensão SQL Server para MySQL em AWS SCT. Para obter mais informações sobre pacotes de extensão, consulte [Usando AWS SCT pacotes de extensão](#).

Tópicos

- [Privilégios do MySQL como um banco de dados de destino](#)
- [Configurações de conversão do SQL Server para MySQL](#)
- [Considerações sobre a migração](#)

Privilégios do MySQL como um banco de dados de destino

Os privilégios obrigatórios para MySQL como destino são listados a seguir:

- 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.*

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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';
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Em seguida, substitua *your_password* por uma senha segura.

Se você usar um banco de dados MySQL versão 5.7 ou inferior como destino, execute o comando a seguir. Para bancos de dados MySQL versão 8.0 e superior, esse comando está obsoleto.

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

Para usar o Amazon RDS para MySQL ou o Aurora MySQL como destino, defina o parâmetro `lower_case_table_names` como 1. Esse valor significa que o servidor MySQL manipula identificadores de nomes de objetos como tabelas, índices, acionadores e bancos de dados sem distinção entre maiúsculas e minúsculas. Se você ativou o registro binário em sua instância de destino, defina o parâmetro `log_bin_trust_function_creators` como 1. Nesse caso, você não precisa usar as características `DETERMINISTIC`, `READS SQL DATA` ou `NO SQL` para criar funções armazenadas. Para configurar esses parâmetros, crie um novo grupo de parâmetros de banco de dados ou modifique um grupo de parâmetros de banco de dados existente.

Configurações de conversão do SQL Server para MySQL

Para editar as configurações de conversão do SQL Server para MySQL, em AWS SCT escolha Configurações e, em seguida, escolha de conversão. Na lista superior, escolha SQL Server e, em seguida, escolha SQL Server – MySQL. A AWS SCT exibe todas as configurações disponíveis para conversão de SQL Server para MySQL.

As configurações de conversão do SQL Server para MySQL na AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para permitir que seu banco de dados SQL Server de origem armazene a saída de EXEC em uma tabela. A AWS SCT cria tabelas temporárias e um procedimento adicional para emular esse atributo. Para usar essa emulação, selecione Criar rotinas adicionais para lidar com conjuntos de dados abertos.

Considerações sobre a migração

Há alguns aspectos a serem considerados ao migrar um esquema do SQL Server para o MySQL:

- O MySQL não é compatível com a instrução MERGE. No entanto, a AWS SCT pode emular a instrução MERGE durante a conversão usando a cláusula `INSERT ON DUPLICATE KEY` e as instruções `UPDATE FROM` and `DELETE FROM`.

Para emulação correta usando `INSERT ON DUPLICATE KEY`, é necessário que haja uma restrição ou chave primária no banco de dados MySQL de destino.

- Uma instrução GOTO e um rótulo podem ser usados para alterar a ordem em que as instruções são executadas. Todas as instruções Transact-SQL que seguem a instrução GOTO são ignoradas, e o processamento continua no rótulo. As instruções GOTO e os rótulos podem ser usados em qualquer lugar em um procedimento, lote ou bloco de instruções. Você também pode aninhar instruções GOTO.

O MySQL não usa instruções GOTO. Quando a AWS SCT converte o código que contém uma instrução GOTO, ela converte a instrução para usar uma instrução `BEGIN...END` ou `LOOP...END LOOP`. Você pode encontrar exemplos de como a AWS SCT converte instruções GOTO na tabela a seguir.

Instrução do SQL Server

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

Instrução do MySQL

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

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

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

Instrução do SQL Server	Instrução do MySQL
<pre>BEGIN statement1; label1: statement2; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END</pre>

- O MySQL não suporta funções com valor de tabela com múltiplas instruções. A AWS SCT simula as funções com valor de tabela durante uma conversão, criando tabelas temporárias e reescrevendo instruções para usar essas tabelas temporárias.

Convertendo um SQL Server em PostgreSQL

Você pode usar o pacote de extensão SQL Server para PostgreSQL em AWS SCT. Esse pacote de extensão emula as funções do banco de dados do SQL Server no código PostgreSQL convertido. Use o pacote de extensão SQL Server para PostgreSQL para emular o SQL Server Agent e o SQL Server Database Mail. Para obter mais informações sobre pacotes de extensão, consulte [Usando AWS SCT pacotes de extensão](#).

Tópicos

- [Privilégios do PostgreSQL como um banco de dados de destino](#)
- [Configurações de conversão do SQL Server para o PostgreSQL](#)
- [Converter as partições do SQL Server para as partições do PostgreSQL versão 10](#)
- [Considerações sobre a migração](#)
- [Usando um pacote de extensão da AWS SCT para emular o SQL Server Agent no PostgreSQL](#)
- [Usando um pacote de extensão da AWS SCT para emular o SQL Server Database Mail no PostgreSQL](#)

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino, a AWS SCT necessita do privilégio `CREATE ON DATABASE`. Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar os sinônimos públicos convertidos, altere o caminho de pesquisa padrão do banco de dados para `"$user", public_synonyms, public`.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder os privilégios.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

No PostgreSQL, apenas o proprietário do esquema ou um `superuser` pode descartar um esquema. O proprietário pode descartar um esquema e todos os objetos incluídos nesse esquema, mesmo que o proprietário do esquema não possua alguns de seus objetos.

Ao usar usuários diferentes para converter e aplicar esquemas diferentes ao banco de dados de destino, você pode receber uma mensagem de erro quando AWS SCT não consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil `superuser`.

Configurações de conversão do SQL Server para o PostgreSQL

Para editar as configurações de conversão do SQL Server para PostgreSQL, escolha Configurações e, em seguida, escolha Configurações de conversão. Na lista superior, escolha SQL Server e, em seguida, escolha SQL Server – PostgreSQL. A AWS SCT exibe todas as configurações disponíveis para conversão de SQL Server para PostgreSQL.

As configurações de conversão do SQL Server para PostgreSQL em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para permitir o uso de índices com o mesmo nome em tabelas diferentes no SQL Server.

No PostgreSQL, todos os nomes de índice que você usa no esquema devem ser exclusivos. Para garantir que a AWS SCT gere nomes exclusivos para todos os seus índices, selecione Gerar nomes exclusivos para índices.

- Para converter procedimentos do SQL Server em funções do PostgreSQL.

A versão 10 e anteriores do PostgreSQL não oferece suporte a procedimentos. Para clientes que não estão familiarizados com o uso de procedimentos no PostgreSQL, a AWS SCT pode converter procedimentos em perfis. Para fazer isso, selecione Converter procedimentos em perfis.

- Para emular a saída de EXEC em uma tabela.

Seu banco de dados SQL Server de origem pode armazenar a saída de EXEC em uma tabela. A AWS SCT cria tabelas temporárias e um procedimento adicional para emular esse atributo. Para usar essa emulação, selecione Criar rotinas adicionais para lidar com conjuntos de dados abertos.

- Para definir o modelo a ser usado para os nomes dos esquemas no código convertido. Para Modelo de geração de nome de esquema, escolha uma das opções a seguir:
 - <source_db> – Usa o nome do banco de dados SQL Server como o nome de um esquema no PostgreSQL.
 - <source_schema> – Usa o nome do esquema do SQL Server como o nome de um esquema no PostgreSQL.
 - <source_db>_<schema> – Usa uma combinação do banco de dados SQL Server e dos nomes do esquema como um nome de esquema no PostgreSQL.
- Para manter as letras maiúsculas dos nomes dos objetos de origem.

Para evitar a conversão de nomes de objetos em minúsculas, selecione Evitar conversão para minúsculas para operações com distinção entre maiúsculas e minúsculas. Essa opção se aplica somente ao ativar a opção de diferenciação de maiúsculas e minúsculas no banco de dados de destino.

- Para manter os nomes dos parâmetros do seu banco de dados de origem.

Para adicionar aspas duplas aos nomes dos parâmetros no código convertido, selecione Manter nomes de parâmetros originais.

Converter as partições do SQL Server para as partições do PostgreSQL versão 10

Ao converter um banco de dados Microsoft SQL Server em Amazon Aurora Edição Compatível com PostgreSQL (Aurora PostgreSQL) ou o serviço de banco de dados relacional da Amazon para PostgreSQL (Amazon RDS para PostgreSQL), esteja ciente do seguinte.

No SQL Server, você cria partições com funções de partição. Ao fazer a conversão de uma tabela particionada do SQL Server para uma tabela particionada do PostgreSQL versão 10, atente-se aos possíveis problemas:

- O SQL Server permite que você particione uma tabela usando uma coluna sem restrição NOT NULL. Nesse caso, todos os valores NULL passam para a partição mais à esquerda. O PostgreSQL não é compatível com os valores NULL para particionamento RANGE.
- O SQL Server permite que você crie chaves primárias e exclusivas para tabelas particionadas. No PostgreSQL, é possível criar chaves primárias e exclusivas para cada partição diretamente. Assim, a restrição PRIMARY UNIQUE KEY deve ser removida da tabela pai ao migrar para o PostgreSQL. Os nomes de chaves resultantes assumem o formato `<original_key_name>_<partition_number>`.
- O SQL Server permite que você crie a restrição de chave estrangeira para e de tabelas particionadas. O PostgreSQL não é compatível com chaves estrangeiras que referenciam tabelas particionadas. Além disso, o PostgreSQL não é compatível com as referências de chave estrangeira de uma tabela particionada para outra tabela.
- O SQL Server permite que você crie índices para tabelas particionadas. No PostgreSQL, um índice deve ser criado para cada partição diretamente. Assim, os índices devem ser removidos das tabelas pai ao migrar para o PostgreSQL. Os nomes de índices resultantes assumem o formato `<original_index_name>_<partition_number>`.
- O PostgreSQL não é compatível com índices particionados.

Considerações sobre a migração

Há alguns aspectos a serem considerados ao migrar um esquema do SQL Server para o PostgreSQL:

- No PostgreSQL, todos os nomes de objeto em um esquema devem ser exclusivos, incluindo os índices. Os nomes de índice devem ser exclusivos no esquema da tabela-base. No SQL Server, um nome de índice pode ser o mesmo em tabelas diferentes.

Para garantir a exclusividade dos nomes de índice, a AWS SCT oferece a opção de gerar nomes de índice exclusivos se os seus não forem exclusivos. Para isso, clique na opção *Generate unique index names* (Gerar nomes de índice exclusivos) nas propriedades do projeto. Por padrão, essa opção é habilitada. Se essa opção estiver habilitada, os nomes de índice exclusivos são criados usando o formato `IX_table_name_index_name`. Caso contrário, os nomes de índice não são alterados.

- Uma instrução GOTO e um rótulo podem ser usados para alterar a ordem em que as instruções são executadas. Todas as instruções Transact-SQL que seguem a instrução GOTO são ignoradas, e o processamento continua no rótulo. As instruções GOTO e os rótulos podem ser usados em qualquer lugar dentro de um procedimento, lote ou bloco de instruções. Além disso, as instruções GOTO podem ser agrupadas.

O PostgreSQL não usa instruções GOTO. Quando a AWS SCT converte o código que contém uma instrução GOTO, ela converte a instrução para usar uma instrução `BEGIN... END` ou `LOOP... END LOOP`. Você pode encontrar exemplos de como a AWS SCT converte instruções GOTO na tabela a seguir.

As instruções GOTO do SQL Server e as instruções PostgreSQL convertidas

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

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

- O PostgreSQL não suporta uma instrução MERGE. A AWS SCT emula o comportamento de uma instrução MERGE das seguintes formas:
 - Com a construção INSERT ON CONFLICT.
 - Com a instrução UPDATE FROM DML, como MERGE sem uma cláusula WHEN NOT MATCHED.

- Ao usar CURSOR, como com uma cláusula MERGE com DELETE ou usando uma instrução de condição MERGE ON complexa.
- A AWS SCT poderá adicionar triggers de banco de dados à árvore do objeto quando o Amazon RDS for o destino.
- A AWS SCT poderá adicionar triggers no nível do servidor à árvore do objeto quando o Amazon RDS for o destino.
- O SQL Server cria e gerencia tabelas `deleted` e `inserted` automaticamente. Você pode usar essas tabelas temporárias residentes na memória para testar os efeitos de determinadas modificações de dados e definir condições para ações de trigger de DML. A AWS SCT pode converter o uso dessas tabelas em instruções de trigger de DML.
- A AWS SCT poderá adicionar servidores vinculados à árvore do objeto quando o Amazon RDS for o destino.
- Ao migrar do Microsoft SQL Server para PostgreSQL, a função `SUSER_SNAME` incorporada é convertida da seguinte forma:
 - `SUSER_SNAME` – retorna o nome de login associado a um número de identificação de segurança (SID).
 - `SUSER_SNAME(<server_user_sid>)` – Sem suporte.
 - `SUSER_SNAME CURRENT_USER ()` – Retorna o nome do usuário do contexto de execução atual.
 - `SUSER_SNAME (NULL)` – Retorna NULL.
- A conversão de funções com valor de tabela é suportada. As funções com valor de tabela retornam uma tabela e podem substituir uma tabela em uma consulta.
- `PATINDEX` retorna a posição inicial da primeira ocorrência de um padrão em uma expressão especificada em todos os tipos de dados de texto e caracteres válidos. Ele retornará zeros se o padrão não for encontrado. Ao converter do SQL Server para o Amazon RDS para PostgreSQL, a AWS SCT substitui o código de aplicativo que usa `PATINDEX` por `aws_sqlserver_ext.patindex(<pattern character>, <expression character varying>)`.
- No SQL Server, um tipo de tabela definido pelo usuário representa a definição de uma estrutura de tabela. Use um tipo de tabela definido pelo usuário para declarar parâmetros de valor de tabela para procedimentos armazenados ou funções. Também é possível usar um tipo de tabela definido pelo usuário para declarar variáveis de tabela a serem usadas em um lote ou no corpo de um procedimento armazenado ou uma função. A AWS SCT emulou esse tipo no PostgreSQL ao criar uma tabela temporária.

Ao converter do SQL Server para PostgreSQL, a AWS SCT converte objetos do sistema do SQL Server em objetos reconhecíveis no PostgreSQL. A tabela a seguir mostra como os objetos do sistema foram convertidos.

Casos de uso do MS SQL Server	Substituição do 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
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS

Casos de uso do MS SQL Server	Substituição do PostgreSQL
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATABASES
INFORMATION_SCHEMA.SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA.VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA.TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA.COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA.CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA.TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS
INFORMATION_SCHEMA.KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLUMN_USAGE
INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_TABLE_USAGE

Casos de uso do MS SQL Server	Substituição do PostgreSQL
INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_COLUMN_USAGE
INFORMATION_SCHEMA.ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROUTINES
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROCESSES
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

Usando um pacote de extensão da AWS SCT para emular o SQL Server Agent no PostgreSQL

SQL Server Agent é um serviço do Microsoft Windows que executa trabalhos do SQL Server. SQL Server Agent pode executar trabalhos agendados em resposta a um evento específico ou sob demanda. Para obter mais informações sobre o SQL Server Agent, consulte a [documentação técnica da Microsoft](#).

O PostgreSQL não tem um equivalente para SQL Server Agent. Para emular os recursos do SQL Server Agent, a AWS SCT cria um pacote de extensão. Este pacote de extensão usa o AWS Lambda e o Amazon CloudWatch. O AWS Lambda implementa a interface que você usa para gerenciar agendamentos e executar trabalhos. O Amazon CloudWatch mantém as regras de programação.

O AWS Lambda e o Amazon CloudWatch usam um parâmetro JSON para interagir. Esse parâmetro JSON tem a seguinte estrutura.

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

No exemplo anterior, *mode* é o tipo da tarefa e *list of parameters* é um conjunto de parâmetros que dependem do tipo da tarefa. Além disso, *procedure name* é o nome do procedimento executado após a conclusão da tarefa.

A AWS SCT usa uma função do Lambda para controlar e executar trabalhos. A regra do CloudWatch inicia a execução do trabalho e fornece as informações necessárias para iniciar o trabalho. Quando a regra do CloudWatch é acionada, ela inicia a função do Lambda usando os parâmetros da regra.

Para criar um trabalho simples que chame um procedimento, use o formato a seguir.

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

Para criar um trabalho com várias etapas, use o formato a seguir.

```
{
  "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 emular o comportamento do SQL Server Agent no PostgreSQL, o pacote de extensão da AWS SCT também cria as tabelas e os procedimentos a seguir.

Tabelas que emulam o SQL Server Agent no PostgreSQL

Para emular o SQL Server Agent, o pacote de extensão usa as seguintes tabelas:

sysjobs

Armazena as informações sobre os trabalhos.

sysjobsteps

Armazena as informações sobre as etapas de um trabalho.

syschedules

Armazena as informações sobre os cronogramas de trabalho.

sysjobschedules

Armazena as informações de cronograma de trabalhos individuais.

sysjobhistory

Armazena as informações sobre as execuções de trabalhos programados.

Procedimentos que emulam o SQL Server Agent no PostgreSQL

Para emular o SQL Server Agent, o pacote de extensão usa os seguintes procedimentos:

sp_add_job

Adiciona um novo trabalho.

sp_add_jobstep

Adiciona uma etapa a um trabalho.

sp_add_schedule

Cria uma nova regra de cronograma no Amazon CloudWatch. Você pode usar esse cronograma com qualquer número de trabalhos.

sp_attach_schedule

Define um cronograma para o trabalho selecionado.

sp_add_jobschedule

Cria uma regra de cronograma para um trabalho no Amazon CloudWatch e define o destino para essa regra.

sp_update_job

Atualiza os atributos do trabalho criado anteriormente.

sp_update_jobstep

Atualiza os atributos da etapa em um trabalho.

sp_update_schedule

Atualiza os atributos de uma regra de cronograma no Amazon CloudWatch.

sp_update_jobschedule

Atualiza os atributos do cronograma para o trabalho especificado.

sp_delete_job

Exclui um trabalho.

sp_delete_jobstep

Exclui uma etapa do trabalho de um trabalho.

sp_delete_schedule

Exclui um cronograma.

sp_delete_jobschedule

Exclui a regra de cronograma para o trabalho especificado do Amazon CloudWatch.

sp_detach_schedule

Remove uma associação entre um cronograma e um trabalho.

get_jobs, update_job

Procedimentos internos que interagem com 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

Procedimentos internos que verificam as configurações.

Sintaxe para procedimentos que emulam o SQL Server Agent no PostgreSQL

O procedimento `aws_sqlserver_ext.sp_add_job` no pacote de extensão emula o procedimento `msdb.dbo.sp_add_job`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_add_jobstep` no pacote de extensão emula o procedimento `msdb.dbo.sp_add_jobstep`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da Microsoft](#).

```
par_job_id integer = NULL::integer,
par_job_name varchar = NULL::character varying,
par_step_id integer = NULL::integer,
par_step_name varchar = NULL::character varying,
par_subsystem varchar = 'TSQL'::bpchar,
par_command text = NULL::text,
par_additional_parameters text = NULL::text,
par_cmexec_success_code integer = 0,
par_on_success_action smallint = 1,
par_on_success_step_id integer = 0,
par_on_fail_action smallint = 2,
par_on_fail_step_id integer = 0,
par_server varchar = NULL::character varying,
par_database_name varchar = NULL::character varying,
par_database_user_name varchar = NULL::character varying,
par_retry_attempts integer = 0,
par_retry_interval integer = 0,
par_os_run_priority integer = 0,
par_output_file_name varchar = NULL::character varying,
par_flags integer = 0,
par_proxy_id integer = NULL::integer,
par_proxy_name varchar = NULL::character varying,
inout par_step_uid char = NULL::bpchar,
out returncode integer
```

O procedimento `aws_sqlserver_ext.sp_add_schedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_add_schedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_attach_schedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_attach_schedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_add_jobschedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_add_jobschedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_delete_job` no pacote de extensão emula o procedimento `msdb.dbo.sp_delete_job`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_delete_jobstep` no pacote de extensão emula o procedimento `msdb.dbo.sp_delete_jobstep`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da Microsoft](#).

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

O procedimento `aws_sqlserver_ext.sp_delete_jobschedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_delete_jobschedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_delete_schedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_delete_schedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_detach_schedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_detach_schedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_update_job` no pacote de extensão emula o procedimento `msdb.dbo.sp_update_job`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_update_jobschedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_update_jobschedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_update_jobstep` no pacote de extensão emula o procedimento `msdb.dbo.sp_update_jobstep`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sp_update_schedule` no pacote de extensão emula o procedimento `msdb.dbo.sp_update_schedule`. Para obter mais informações sobre o procedimento do SQL Server Agent de origem, consulte a [documentação técnica da 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
```

Exemplos de uso de procedimentos que emulam o SQL Server Agent no PostgreSQL

Para adicionar um novo trabalho, use o procedimento `aws_sqlserver_ext.sp_add_job` conforme mostrado a seguir.

```
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 adicionar uma nova etapa do trabalho, use o procedimento `aws_sqlserver_ext.sp_add_jobstep` conforme mostrado a seguir.

```
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 adicionar um cronograma simples, use o procedimento `aws_sqlserver_ext.sp_add_schedule` conforme mostrado a seguir.

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

Para definir um cronograma para um trabalho, use o procedimento `aws_sqlserver_ext.sp_attach_schedule` conforme mostrado a seguir.

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

Para criar um cronograma para um trabalho, use o procedimento `aws_sqlserver_ext.sp_add_jobschedule` conforme mostrado a seguir.

```
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);
```

Use exemplos de casos para emular o SQL Server Agent no PostgreSQL

Se o código do banco de dados de origem usa o SQL Server Agent para executar trabalhos, você pode usar o pacote de extensão SQL Server para PostgreSQL para AWS SCT para converter esse código em PostgreSQL. O pacote de extensão usa funções AWS Lambda para emular o comportamento do SQL Server Agent.

Você pode criar uma nova função AWS Lambda ou registrar uma função existente.

Para criar uma nova função do AWS Lambda

1. Na AWS SCT, na árvore do banco de dados de destino, abra o menu de contexto (clique com o botão direito do mouse), escolha Aplicar pacote de extensão para e, em seguida, escolha PostgreSQL.

O assistente do pacote de extensões é exibido.

2. Na guia Serviço de emulação do SQL Server Agent, faça o seguinte:
 - Escolha Criar uma função AWS Lambda.
 - Em Login do banco de dados, insira o nome do usuário do banco de dados de destino.
 - Em Senha do banco de dados, insira a senha do nome de usuário que você inseriu na etapa anterior.
 - Para a pasta da biblioteca Python, insira o caminho para a pasta da biblioteca Python.
 - Escolha Criar uma função AWS Lambda e, em seguida, escolha Avançar.

Para registrar uma função AWS Lambda que você implantou anteriormente

- Execute o script a seguir no banco de dados de destino.

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

No exemplo anterior, *ARN* é o nome do recurso da Amazon (ARN) da função AWS Lambda implantada.

O exemplo a seguir cria uma tarefa simples que consiste em uma etapa. A cada cinco minutos, essa tarefa executa a função `job_example` criada anteriormente. Essa função insere registros na tabela `job_example_table`.

Para criar essa tarefa simples

1. Crie um trabalho usando a função `aws_sqlserver_ext.sp_add_job`, conforme mostrado a seguir.

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

2. Crie uma etapa de trabalho usando a função `aws_sqlserver_ext.sp_add_jobstep`, conforme mostrado a seguir.

```
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;');
```

A etapa do trabalho especifica o que a função faz.

3. Crie um programador para o trabalho usando a função `aws_sqlserver_ext.sp_add_jobschedule`, conforme mostrado a seguir.

```
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 */);
```

A etapa do trabalho especifica o que a função faz.

Para excluir esse trabalho, use a função `aws_sqlserver_ext.sp_delete_job` conforme mostrado a seguir.

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

Usando um pacote de extensão da AWS SCT para emular o SQL Server Database Mail no PostgreSQL

Você pode usar o SQL Server Database Mail para enviar mensagens de e-mail aos usuários a partir do mecanismo de banco de dados do SQL Server ou da instância gerenciada do Azure SQL. Essas mensagens de e-mail podem conter resultados de consultas ou incluir arquivos de qualquer recurso em sua rede. Para obter mais informações sobre o SQL Server Database Mail, consulte a [documentação técnica da Microsoft](#).

O PostgreSQL não tem um equivalente para o SQL Server Database Mail. Para emular os atributos do SQL Server Database Mail, a AWS SCT cria um pacote de extensão. Este pacote de extensão usa a AWS Lambda o Amazon Simple Email Service (Amazon SES). A AWS Lambda fornece aos usuários uma interface para interagir com o serviço de envio de e-mail do Amazon SES. Para configurar essa interação, adicione o nome do recurso da Amazon (ARN) da sua função do Lambda.

Para uma nova conta de e-mail, use o comando a seguir.

```
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 adicionar o ARN da sua função do Lambda à conta de e-mail existente, use o comando a seguir.

```
do  
$$
```

```
begin
PERFORM sysmail_update_account_sp (
    par_account_name := 'existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

Nos exemplos anteriores, *ARN* é o ARN da sua função do Lambda.

Para emular o comportamento do SQL Server Database Mail no PostgreSQL, o pacote de extensão da AWS SCT usa as seguintes tabelas, visualizações e procedimentos.

Tabelas que emulam o SQL Server Database Mail no PostgreSQL

Para emular o SQL Server Database Mail, o pacote de extensão usa as seguintes tabelas:

`sysmail_account`

Armazena as informações sobre as contas de e-mail.

`sysmail_profile`

Armazena as informações sobre os perfis de usuário.

`sysmail_server`

Armazena as informações sobre os servidores de e-mail.

`sysmail_mailitems`

Armazena a lista das mensagens de e-mail.

`sysmail_attachments`

Contém uma linha para cada anexo de e-mail.

`sysmail_log`

Armazena as informações do serviço sobre o envio de mensagens de e-mail.

`sysmail_profileaccount`

Armazena as informações sobre os perfis de usuário e contas de e-mail.

Visualizações que emulam o SQL Server Database Mail no PostgreSQL

Para emular o SQL Server Database Mail, a AWS SCT cria as seguintes visualizações no banco de dados PostgreSQL para garantir a compatibilidade. O pacote de extensão não as usa, mas seu código convertido pode consultar essas visualizações.

sysmail_allitems

Inclui uma lista de todos os e-mails.

sysmail_faileditems

Inclui uma lista de e-mails que não puderam ser enviados.

sysmail_sentitems

Inclui uma lista de e-mails enviados.

sysmail_unsentitems

Inclui uma lista de e-mails que ainda não foram enviados.

sysmail_mailattachments

Inclui uma lista de arquivos anexados.

Procedimentos que emulam o SQL Server Database Mail no PostgreSQL

Para emular o SQL Server Database Mail, o pacote de extensão usa os seguintes procedimentos:

sp_send_dbmail

Envia um e-mail para os destinatários especificados.

sysmail_add_profile_sp

Cria um novo perfil de usuário.

sysmail_add_account_sp

Cria uma nova conta de e-mail que armazena informações como credenciais do Simple Mail Transfer Protocol (SMTP) e assim por diante.

sysmail_add_profileaccount_sp

Adiciona uma conta de e-mail ao perfil de usuário especificado.

sysmail_update_profile_sp

Altera os atributos do perfil do usuário, como descrição, nome e assim por diante.

sysmail_update_account_sp

Altera as informações na conta de e-mail existente.

sysmail_update_profileaccount_sp

Atualiza as informações da conta de e-mail no perfil de usuário especificado.

sysmail_delete_profileaccount_sp

Remove uma conta de e-mail do perfil de usuário especificado.

sysmail_delete_account_sp

Exclui a conta de e-mail.

sysmail_delete_profile_sp

Exclui o perfil de usuário.

sysmail_delete_mailitems_sp

Exclui e-mails de tabelas internas.

sysmail_help_profile_sp

Exibe informações sobre o perfil do usuário.

sysmail_help_account_sp

Exibe informações sobre a conta de e-mail.

sysmail_help_profileaccount_sp

Exibe informações sobre contas de e-mail associadas ao perfil de usuário.

sysmail_dbmail_json

Um procedimento interno que gera solicitações JSON para funções AWS Lambda.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Procedimentos internos que verificam as configurações.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Procedimentos internos obsoletos.

Sintaxe para procedimentos que emulam o SQL Server Database Mail no PostgreSQL

O procedimento `aws_sqlserver_ext.sp_send_dbmail` no pacote de extensão emula o procedimento `msdb.dbo.sp_send_dbmail`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_delete_mailitems_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_delete_mailitems_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,
```

```
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_add_profile_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_add_profile_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_add_account_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_add_account_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_add_profileaccount_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_add_profileaccount_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_help_profile_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_help_profile_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
out_returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_update_profile_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_update_profile_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out_returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_delete_profile_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_delete_profile_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out_returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_help_account_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_help_account_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_account_id integer = NULL::integer,
```

```
par_account_name varchar = NULL::character varying,  
out returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_update_account_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_update_account_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_delete_account_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_delete_account_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

O procedimento `aws_sqlserver_ext.sysmail_help_profileaccount_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_help_profileaccount_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_update_profileaccount_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_update_profileaccount_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

O procedimento `aws_sqlserver_ext.sysmail_delete_profileaccount_sp` no pacote de extensão emula o procedimento `msdb.dbo.sysmail_delete_profileaccount_sp`. Para obter mais informações sobre o procedimento do SQL Server Database Mail de origem, consulte a [documentação técnica da 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
```

Exemplos de uso de procedimentos que emulam o SQL Server Database Mail no PostgreSQL

Para enviar um e-mail, use o procedimento `aws_sqlserver_ext.sp_send_dbmail` conforme mostrado a seguir.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',  
    par_body := 'The stored procedure finished'  
);
```

O exemplo a seguir mostra como enviar um e-mail com resultados da 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'  
);
```

O exemplo a seguir mostra como enviar e-mails brutos com 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 excluir e-mails, use o procedimento

`aws_sqlserver_ext.sysmail_delete_mailitems_sp` conforme mostrado a seguir.

```
DECLARE var_GETDATE datetime;  
SET var_GETDATE = NOW();  
PERFORM sysmail_delete_mailitems_sp (  
  par_sent_before := var_GETDATE  
);
```

O exemplo a seguir mostra como excluir os e-mails mais antigos.

```
PERFORM sysmail_delete_mailitems_sp (  
  par_sent_before := '31.12.2015'  
);
```

O exemplo a seguir mostra como excluir todos os e-mails que não podem ser enviados.

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_status := 'failed'  
);
```

Para criar um novo perfil de usuário, use o procedimento `aws_sqlserver_ext.sysmail_add_profile_sp` conforme mostrado a seguir.

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

O exemplo a seguir mostra como criar um novo perfil e salvar o identificador de perfil exclusivo em uma variável.

```
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 criar uma nova conta de e-mail, use o procedimento `aws_sqlserver_ext.sysmail_add_account_sp` conforme mostrado a seguir.

```
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 adicionar uma conta de e-mail ao perfil do usuário, use o procedimento `aws_sqlserver_ext.sysmail_add_profileaccount_sp` conforme mostrado a seguir.

```
PERFORM sysmail_add_profileaccount_sp (  

```

```
par_account_name := 'Administrator',  
par_account_name := 'Audit Account',  
par_sequence_number := 1  
);
```

Use exemplos de casos para emular o SQL Server Database Mail no PostgreSQL

Se o código do banco de dados de origem usa o SQL Server Database Mail para enviar e-mails, você pode usar o pacote de extensão da AWS SCT para converter esse código em PostgreSQL.

Para enviar um e-mail do seu banco de dados PostgreSQL

1. Crie e configure sua função AWS Lambda.
2. Aplique o pacote de extensão da AWS SCT
3. Crie um perfil de usuário usando a função `sysmail_add_profile_sp` conforme mostrado a seguir.
4. Crie uma conta de e-mail usando a função `sysmail_add_account_sp` conforme mostrado a seguir.
5. Adicione essa conta de e-mail ao seu perfil de usuário usando a função `sysmail_add_profileaccount_sp` mostrada a seguir.

```
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. Envie um e-mail usando a função `sp_send_dbmail`, conforme mostrado a seguir.

```

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 as informações sobre todos os perfis de usuário, use o procedimento `sysmail_help_profile_sp` conforme mostrado a seguir.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

O exemplo a seguir exibe as informações sobre o perfil de usuário 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 visualizar as informações sobre todas as contas de e-mail, use o procedimento `sysmail_help_account_sp` conforme mostrado a seguir.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

O exemplo a seguir exibe as informações sobre a conta de e-mail 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 visualizar as informações sobre todas as contas de e-mail que estão associadas com os perfis de usuário, use o procedimento `sysmail_help_profileaccount_sp` conforme mostrado a seguir.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

O exemplo a seguir filtra os registros por identificador, nome do perfil ou nome da conta.

```
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 alterar o nome ou a descrição do perfil do usuário, use o procedimento `sysmail_update_profile_sp` conforme mostrado a seguir.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
par_profile_id := 2,
par_profile_name := 'New profile name'
);
```

Para alterar as configurações da conta de e-mail, use o procedimento `ysmail_update_account_sp` conforme mostrado a seguir.

```
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'
);
```

Converter SQL Server para o Amazon RDS for SQL Server

Há alguns aspectos a serem considerados ao migrar o esquema e o código do SQL Server para o Amazon RDS for SQL Server:

- A AWS SCT pode converter o SQL Server Agent para fornecer cronogramas, alertas e trabalhos em uma instância de banco de dados do Amazon RDS para SQL Server. Após a conversão, é possível usar uma instância de banco de dados do Amazon RDS for SQL Server com o SQL Server Reporting Service (SSRS), o SQL Server Analysis Services (SSAS) e o SQL Server Integration Services (SSIS).
- Atualmente, o Amazon RDS não é compatível com o SQL Server Service Broker nem outros endpoints do T-SQL que exijam a execução do comando CREATE ENDPOINT.
- O Amazon RDS tem suporte limitado para servidores vinculados. Quando o código de aplicativo do SQL Server que usa servidores vinculados é convertido, a AWS SCT converte o código do aplicativo. No entanto, certifique-se de revisar o comportamento de objetos que usam servidores vinculados antes de executar o código convertido.
- Sempre ativo é usado.
- O relatório de avaliação da AWS SCT fornece métricas do servidor para a conversão. Essas métricas sobre sua instância do SQL Server incluem:
 - O espelhamento de dados é usado.
 - O envio de logs do SQL Server é configurado.
 - O cluster de failover é usado.
 - O e-mail do banco de dados é configurado.
 - O serviço Full Text Search é usado. O Amazon RDS for SQL Server tem um Full Text Search limitado. Além disso, ele não oferece suporte à pesquisa semântica.
 - O Data Quality Service (DQS) está instalado. O Amazon RDS não oferece suporte ao DQS. Portanto, recomendamos que você instale o SQL Server em uma instância do Amazon EC2.

Privilégios do RDS para SQL Server como destino

Para migrar para o RDS para SQL Server, crie um usuário de banco de dados e, em seguida, conceda os privilégios necessários para cada banco de dados. Você pode usar o exemplo de código a seguir.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino. Por fim, substitua *your_password* por uma senha segura.

Origens de data warehouse para AWS Schema Conversion Tool

A AWS SCT pode converter esquemas dos seguintes data warehouses de origem em um destino compatível. Para obter informações sobre permissões, conexões e o que a AWS SCT pode converter para uso com o banco de dados de destino ou data warehouse, consulte detalhes a seguir.

Tópicos

- [Como usar o Amazon Redshift como origem para AWS SCT](#)
- [Como usar o Azure Synapse Analytics como origem para AWS SCT](#)
- [Como usar o BigQuery como origem para AWS SCT](#)
- [Como usar o banco de dados Greenplum como origem para AWS SCT](#)
- [Como usar o Netezza como origem para AWS SCT](#)
- [Como usar o data warehouse do Oracle como origem para a AWS SCT](#)
- [Como usar o Snowflake como origem para AWS SCT](#)
- [Como usar o data warehouse do Microsoft SQL Server como origem para AWS SCT](#)

- [Como usar o Teradata como origem para AWS SCT](#)
- [Como usar o Vertica como origem para AWS SCT](#)

Como usar o Amazon Redshift como origem para AWS SCT

É possível usar a AWS SCT para otimizar seu cluster do Amazon Redshift. A AWS SCT fornece recomendações sobre a seleção de chaves de distribuição e classificação para seu cluster do Amazon Redshift. Você pode considerar o projeto de otimização do Amazon Redshift como um projeto AWS SCT com a origem e o destino apontando para os diferentes clusters do Amazon Redshift.

Privilégios do Amazon Redshift como um banco de dados de origem

Os privilégios obrigatórios para usar o Amazon Redshift como uma origem estão listados a seguir:

- 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

Nos exemplos anteriores, substitua o espaço reservado *<schema_name>* pelo nome do esquema de origem.

Para obter os privilégios necessários para o Amazon Redshift como destino, consulte a [Permissões para o Amazon Redshift como destino](#).

Como se conectar ao Amazon Redshift como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Amazon Redshift com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados do Amazon Redshift de origem

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Amazon Redshift e, em seguida, escolha Avançar

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir as informações de conexão para o banco de dados de origem do Amazon Redshift, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Insira o nome do banco de dados do Amazon Redshift.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha</p>

Parâmetro	Ação
	<p>por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none">• Verificar certificado do servidor: selecione essa opção para verificar o certificado do servidor, usando um armazenamento confiável.• Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais. <p>Para obter mais informações sobre o suporte a SSL para o Amazon Redshift, consulte Configurar opções de segurança para conexões.</p>
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Redshift	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de otimização do Amazon Redshift

Para editar as configurações de otimização do Amazon Redshift, escolha Configurações na AWS SCT, e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Amazon Redshift e, em seguida, escolha Amazon Redshift — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para a otimização do Amazon Redshift.

As configurações de otimização do Amazon Redshift na AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que o número de tabelas seja maior do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá

uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para escolher a estratégia de migração.

A AWS recomenda o uso de clusters diferentes como origem e destino para seu projeto de otimização. Antes do início do processo de otimização do Amazon Redshift, você cria uma cópia do seu cluster do Amazon Redshift de origem. Você pode incluir seus dados de origem nessa cópia ou criar um cluster vazio.

Em Estratégia de migração, escolha Migração para uma cópia para incluir dados do seu cluster de origem no cluster de destino.

Em Estratégia de migração, escolha Migração para uma lista limpa para analisar as sugestões de otimização. Depois de aceitar essas sugestões, migre seus dados de origem para o cluster de destino.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente quando você selecionou a opção Usar codificação de compactação.

- Para trabalhar com otimização automática de tabelas.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para usar somente na otimização automática de tabelas, escolha Estratégias de otimização no painel esquerdo. Em seguida, selecione Usar ajuste automático de tabela do Amazon Redshift e escolha Nenhum para a Estratégia de seleção inicial de chaves.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com um valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados cujas estatísticas de consulta você deseja analisar.

Como usar o Azure Synapse Analytics como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Azure Synapse Analytics para o Amazon Redshift.

Privilégios do Azure Synapse Analytics como banco de dados de origem

Os seguintes privilégios são necessários para usar um data warehouse do Azure Synapse Analytics como origem:

- VIEW DEFINITION
- VIEW DATABASE STATE

Aplique os privilégios para cada banco de dados cujo esquema você está convertendo.

Como se conectar ao Azure Synapse Analytics como origem

Use o procedimento a seguir para se conectar ao data warehouse do Azure Synapse Analytics com a AWS Schema Conversion Tool.

Para se conectar a um data warehouse do Azure Synapse Analytics como origem

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Azure Synapse Analytics e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir as informações de conexão do data warehouse do Azure Synapse Analytics manualmente, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
SQL pool	Insira o nome do grupo do Azure SQL.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Certificado de servidor confiável: Escolha essa opção para confiar no certificado de servidor. • Armazenamento de confiança: um armazenamento de confiança que você configura nas Configurações globais.
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem informar a senha.

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.

6. Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de conversão do Azure Synapse Analytics para o Amazon Redshift

Para editar as configurações de conversão do Azure Synapse Analytics para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Azure Synapse e, em seguida, escolha Azure Synapse — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do Azure Synapse Analytics para o Amazon Redshift.

As configurações de conversão do Azure Synapse Analytics para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de

aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compactação.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do Azure Synapse Analytics para o Amazon Redshift

Para editar as configurações de otimização de conversão do Azure Synapse Analytics para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Azure Synapse e, em seguida, escolha Azure Synapse — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as

configurações de otimização de conversão para a conversão do Azure Synapse Analytics para o Amazon Redshift.

As configurações de otimização de conversão do Azure Synapse Analytics para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o BigQuery como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do BigQuery para o Amazon Redshift.

Privilégios do BigQuery como origem

Para usar um data warehouse do BigQuery como origem em AWS SCT, crie uma conta de serviço. No Google Cloud, os aplicativos usam contas de serviço para fazer chamadas de API autorizadas. As contas de serviço são diferentes das contas de usuário. Para obter mais informações, consulte [Contas de serviço](#) na documentação do Gerenciamento de acesso e identidade do Google Cloud.

Certifique-se de conceder as seguintes funções à sua conta de serviço:

- BigQuery Admin
- Storage Admin

A função **BigQuery Admin** fornece permissões para gerenciar todos os recursos do projeto. A AWS SCT usa essa função para carregar seus metadados do BigQuery no projeto de migração.

A função **Storage Admin** concede controle total sobre objetos e buckets de dados. Você pode encontrar essa função em **Cloud Storage**. A AWS SCT usa essa função para extrair seus dados do BigQuery e depois carregá-los no Amazon Redshift.

Para criar um arquivo de chave da conta de serviço

1. Faça login no console de gerenciamento do Google Cloud em <https://console.cloud.google.com/>.
2. Na página da [API do BigQuery](#), escolha Ativar. Ignore esta etapa se você vir a API ativada.
3. Na página [Contas de serviço](#), escolha seu projeto e, em seguida, escolha Criar conta de serviço.
4. Na página Detalhes da conta de serviço, insira um valor descritivo para o Nome da conta de serviço. Escolha Criar e continuar. A página Conceder à conta de serviço acesso ao projeto é aberta.
5. Em Selecionar uma função, escolha BigQuery e, em seguida, escolha BigQuery Admin.
6. Escolha Adicionar outra função. Em Selecionar uma função, escolha Armazenamento na nuvem e, em seguida, escolha Storage Admin.
7. Escolha Continuar e, em seguida, escolha Salvar.
8. Na página [Contas de serviço](#), escolha a conta de serviço que você criou.
9. Escolha Chaves e, em seguida, escolha Criar nova chave em Adicionar chave.
10. Selecione JSON e, em seguida, escolha Criar. Escolha a pasta para salvar sua chave privada ou selecione a pasta padrão para downloads em seu navegador.

Para extrair dados de um data warehouse do BigQuery, a AWS SCT usa a pasta de bucket do Google Cloud Storage. Crie esse bucket antes de iniciar a migração de dados. Insira o caminho para sua pasta de bucket do Google Cloud Storage na caixa de diálogo Criar tarefa local. Para obter mais informações, consulte [Criando, executando e monitorando uma AWS SCT tarefa](#).

Como se conectar ao BigQuery como origem

Use o procedimento a seguir para conectar seu projeto de origem do BigQuery com a AWS Schema Conversion Tool.

Para se conectar a um data warehouse de origem do BigQuery

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.

2. Escolha BigQuery e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o projeto do BigQuery. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Em Caminho da chave, insira o caminho para o arquivo de chave da conta de serviço. Para obter mais informações sobre a criação desse arquivo, consulte a [Privilégios do BigQuery como origem](#).
5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao seu projeto de origem do BigQuery.
6. Escolha Conectar para se conectar ao seu projeto de origem do BigQuery.

Limitações de uso do BigQuery como origem da AWS SCT

As limitações a seguir se aplicam ao uso do BigQuery como origem da AWS SCT:

- A AWS SCT não suporta a conversão de subconsultas em funções analíticas.
- Você não pode usar a AWS SCT para converter as declarações `SELECT AS STRUCT` e `SELECT AS VALUE` do BigQuery.
- A AWS SCT não suporta a conversão dos seguintes tipos de funções:
 - Agregado aproximado
 - Bit
 - Depuração
 - Consulta federada
 - Geografia
 - Hash
 - Matemáticas
 - Net
 - Agregado estatístico
 - UUID
- A AWS SCT fornece suporte limitado para a conversão de funções de string.
- A AWS SCT não suporta a conversão de operadores `UNNEST`.
- Você não pode converter operações de junção correlacionadas em AWS SCT.

- A AWS SCT não suporta a conversão de cláusulas QUALIFY, WINDOW, LIMIT e OFFSET.
- Você não pode usar a AWS SCT para converter expressões de tabela comuns recursivas.
- A AWS SCT não suporta a conversão de declarações INSERT com subconsultas dentro de cláusulas VALUES.
- A AWS SCT não suporta a conversão de declarações UPDATE para campos aninhados e registros repetidos.
- Você não pode usar a AWS SCT para converter tipos de dados STRUCT e ARRAY.

Configurações de conversão do BigQuery para o Amazon Redshift

Para editar as configurações de conversão do BigQuery para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Google BigQuery e, em seguida, escolha Google BigQuery — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do BigQuery para o Amazon Redshift.

As configurações de conversão do BigQuery para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do BigQuery para o Amazon Redshift

Para editar as configurações de otimização de conversão do BigQuery para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Google BigQuery e, em seguida, escolha Google BigQuery — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão da conversão do BigQuery para o Amazon Redshift.

As configurações de otimização de conversão do BigQuery para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.

- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o banco de dados Greenplum como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do banco de dados Greenplum para o Amazon Redshift.

Privilégios do banco de dados Greenplum como origem

Os privilégios obrigatórios para o banco de dados Greenplum como origem são listados a seguir:

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<schema_name>*
- SELECT ON *<schema_name>.<table_name>*
- SELECT ON SEQUENCE *<schema_name>.<sequence_name>*

No exemplo anterior, substitua os espaços reservados conforme a seguir:

- Substitua *database_name* pelo nome do banco de dados de origem.
- Substitua *schema_name* pelo nome do esquema de origem.
- Substitua *table_name* pelo nome da tabela de origem.
- Substitua *sequence_name* pelo nome do nome da sequência.

Como se conectar ao banco de dados Greenplum como origem

Use o procedimento a seguir para conectar seu banco de dados do Greenplum de origem com a AWS SCT.

Para conectar-se a um banco de dados de origem do Greenplum

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha SAP ASE e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir as credenciais do banco de dados do Greenplum de origem manualmente, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Digite o nome do banco de dados do Greenplum.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha</p>

Parâmetro	Ação
	<p>por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Verificar certificado do servidor: selecione essa opção para verificar o certificado do servidor, usando um armazenamento confiável. • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver do banco de dados Greenplum	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de conversão do Greenplum para o Amazon Redshift

Para editar as configurações de conversão do Greenplum para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Greenplum e, em seguida, escolha Greenplum — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para a conversão do Greenplum para o Amazon Redshift.

As configurações de conversão do Greenplum para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do Greenplum para o Amazon Redshift

Para editar as configurações de otimização de conversão do Greenplum para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Greenplum e, em seguida, escolha Greenplum — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para a conversão do Greenplum para o Amazon Redshift.

As configurações de otimização de conversão do Greenplum para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.

- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o Netezza como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Netezza para o Amazon Redshift.

Privilégios do Netezza como origem

Os privilégios obrigatórios para Netezza como origem são listados a seguir:

- 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`

No exemplo anterior, substitua os espaços reservados conforme a seguir:

- Substitua `database_name` pelo nome do banco de dados de origem.
- Substitua `schema_name` pelo nome do esquema de origem.

A AWS SCT requer acesso às seguintes tabelas e visualizações do sistema. Você pode conceder acesso a esses objetos em vez de conceder acesso à `system.definition_schema.system view` e `system.definition_schema.system tables` na 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_table_constraint
- select on system.definition_schema._v_table_dist_map
- select on system.definition_schema._v_table_organize_column
- select on system.definition_schema._v_table_storage_stat
- select on system.definition_schema._v_user
- select on system.definition_schema._v_view
- select on system.information_schema._v_relation_column
- select on system.information_schema._v_table
- select on \$hist_column_access_*

Conectando-se ao Netezza como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Netezza com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do Netezza

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Netezza e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do Netezza de origem, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo</p>

Parâmetro	Ação
	novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.
Caminho do driver Netezza	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Como configurar a replicação contínua de dados

Depois de converter seus esquemas de banco de dados Netezza e aplicá-los ao seu banco de dados do Amazon Redshift, você pode migrar dados com agentes de extração de dados da AWS SCT. O agente extrai seus dados e os carrega no bucket do Amazon S3. É possível usar a AWS SCT para copiar dados do Amazon S3 para o Amazon Redshift.

Se os dados em seu banco de dados de origem mudarem durante o processo de migração, você poderá capturar as alterações contínuas com seus agentes de extração de dados da AWS SCT. Em seguida, você pode replicar essas alterações contínuas em seu banco de dados de destino depois

de concluir a migração inicial de dados. Esse processo é chamado de replicação contínua de dados ou captura de dados de alteração (CDC).

Para configurar a replicação contínua de dados para migrações do Netezza para o Amazon Redshift

1. Em seu banco de dados de origem, crie um banco de dados de histórico. É possível usar o exemplo de código a seguir na interface de linha de comandos (CLI) do Netezza.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

No exemplo anterior, substitua *history_database_name* pelo nome do seu banco de dados de histórico. Depois, substitua *load_user* pelo nome do usuário que você definiu para carregar dados do histórico no banco de dados. Em seguida, substitua *histdb_owner* pelo nome do usuário que você definiu como proprietário do banco de dados de histórico. Verifique se você já criou esse usuário e concedeu a permissão CREATE DATABASE. Por fim, substitua *your_password* por uma senha segura.

2. Configure o registro do histórico. Para fazer isso, use o exemplo de código a seguir:

```
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;
```

No exemplo anterior, substitua *history_configuration_name* e *history_database_name* pelos nomes da configuração do histórico e do banco de dados do histórico. Depois, substitua *load_user* pelo nome do usuário que você definiu para carregar dados do histórico no banco de dados. Em seguida, substitua *your_password* por uma senha segura.

3. Conceda permissões de leitura para todas as tabelas no banco de dados do histórico. Você pode usar o exemplo de código a seguir para conceder a permissão SELECT.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

No exemplo anterior, substitua *history_database_name* pelo nome do seu banco de dados de histórico. Em seguida, substitua *your_user* pelo nome do usuário com permissões mínimas

para trabalhar com seu banco de dados Netezza. Você usa as credenciais desse usuário do banco de dados em AWS SCT.

4. Colete estatísticas para cada tabela em seu esquema de origem para obter as informações sobre a cardinalidade das colunas. É possível usar o seguinte comando para gerar estatísticas em seu banco de dados de histórico.

```
GENERATE STATISTICS on "schema_name".table_name";
```

No exemplo anterior, substitua *schema_name* e *table_name* pelo nome do esquema e da tabela do banco de dados.

5. Atenda aos pré-requisitos executando a seguinte consulta:

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

No exemplo anterior, substitua *history_database_name* e *history_schema_name* pelo nome do seu banco de dados de histórico e esquema. Em seguida, substitua *N* pelo número da versão do seu banco de dados de histórico. Para obter mais informações sobre as versões do banco de dados de histórico, consulte a [Documentação do IBM Netezza](#).

6. Instale seus agentes de extração de dados. Para obter mais informações, consulte [Como instalar atendentes de extração](#).

Certifique-se de que o parâmetro {working.folder} no arquivo settings.properties para todas as instâncias do extrator aponte para a mesma pasta. Nesse caso, seus extratores podem coordenar a sessão da CDC e usar um único ponto de transação para todas as subtarefas.

7. Registre seu agente de extração de dados. Para obter mais informações, consulte [Registrando agentes de extração com o AWS Schema Conversion Tool](#).
8. Crie sua tarefa da CDC. Para obter mais informações, consulte [Criando, executando e monitorando uma AWS SCT tarefa](#).

- a. Abra seu projeto em AWS SCT. No painel esquerdo, escolha sua tabela de origem. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Criar tarefa local.
- b. Em Nome da tarefa, insira um nome descritivo para a tarefa de migração de dados.
- c. Para o Modo de migração, escolha Extrair, carregar e copiar.
- d. Selecione Habilitar CDC.

- e. Escolha a guia Configurações da CDC e defina o escopo e a programação das sessões da CDC.
 - f. Selecione Testar tarefa para verificar se você pode se conectar à sua pasta de trabalho, bucket do Amazon S3 e data warehouse do Amazon Redshift.
 - g. Escolha Criar para criar sua tarefa.
 - h. Escolha a guia Tarefas, escolha sua tarefa na lista e escolha Iniciar.
9. A tarefa AWS SCT mantém a consistência transacional no banco de dados de destino. O agente de extração de dados replica as transações da origem na ordem do ID da transação.

Se você interromper qualquer uma das sessões de migração ou se ela falhar, o processamento da CDC também será interrompido.

Configurações de conversão do Netezza para o Amazon Redshift

Para editar as configurações de conversão do Netezza para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Netezza e, em seguida, escolha Netezza — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do Netezza para o Amazon Redshift.

As configurações de conversão do Netezza para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do Netezza para o Amazon Redshift

Para editar as configurações de otimização de conversão do Netezza para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Netezza e, em seguida, escolha Netezza — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para a conversão do Netezza para o Amazon Redshift.

As configurações de otimização de conversão do Netezza para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a

proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.

- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o data warehouse do Oracle como origem para a AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo do data warehouse do Oracle para o Amazon Redshift ou o Amazon Redshift e AWS Glue usados em combinação.

Privilégios do data warehouse do Oracle como origem

Os privilégios obrigatórios para data warehouse do Oracle como origem são listados a seguir:

- connect
- select_catalog_role
- select any dictionary

Conectando-se a um data warehouse Oracle de origem

Use o procedimento a seguir para se conectar ao banco de dados de origem do data warehouse do Oracle com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do data warehouse do Oracle

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Oracle e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do data warehouse do Oracle de origem, use as instruções a seguir:

Parâmetro	Ação
Type	<p>Escolha o tipo de conexão ao seu banco de dados. Dependendo do tipo, forneça as informações adicionais seguintes:</p> <ul style="list-style-type: none"> • SID <ul style="list-style-type: none"> • Nome do servidor: o nome Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem. • Porta do servidor: a porta usada para se conectar ao servidor de banco de dados de origem. • Oracle SID: o ID do sistema da Oracle (SID). Para encontrar o Oracle SID, envie a consulta a seguir para seu banco de dados Oracle: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> • Nome do serviço <ul style="list-style-type: none"> • Nome do servidor: o nome DNS ou o endereço IP do servidor de banco de dados de origem.

Parâmetro	Ação
	<ul style="list-style-type: none">• Porta do servidor: a porta usada para se conectar ao servidor de banco de dados de origem.• Nome do serviço: o nome do serviço Oracle ao qual se conectar.• Alias TNS<ul style="list-style-type: none">• Caminho do arquivo TNS: o caminho para o arquivo que contém as informações de conexão do nome Transparent Network Substrate (TNS).• Caminho do arquivo TNS: o alias TNS do arquivo a ser usado para a conexão com o banco de dados de origem.• Identificador de conexão TNS<ul style="list-style-type: none">• Identificador de conexão TNS: o identificador das informações de conexão TNS registradas.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>

Parâmetro	Ação
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Autenticação SSL: selecione esta opção para usar a autenticação SSL para conexão. • Armazenamento de confiança: a localização de um armazenamento de confiança que contém certificados. • Armazenamento de chaves: a localização de um armazenamento de chaves que contém uma chave privada e certificados. Este valor será obrigatório se a autenticação SSL estiver selecionada. Caso contrário, é opcional.
Armazenar senha	<p>A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.</p>
Caminho do driver Oracle	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de conversão do data warehouse Oracle para o Amazon Redshift

Para editar as configurações de conversão do data warehouse do Oracle para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do data warehouse do Oracle para o Amazon Redshift.

As configurações de conversão do data warehouse do Oracle para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para converter as funções de formatação de tipo de dados como TO_CHAR, TO_DATE e TO_NUMBER com elementos de formato de data e hora que o Amazon Redshift não suporta. Por padrão, a AWS SCT usa as funções do pacote de extensão para emular o uso desses elementos de formato não suportados no código convertido.

O modelo de formato de data e hora no Oracle inclui mais elementos em comparação com strings de formato de data e hora no Amazon Redshift. Quando seu código-fonte inclui somente elementos de formato de data e hora compatíveis com o Amazon Redshift, você não precisa das funções do pacote de extensões no código convertido. Para evitar o uso das funções do pacote de extensões no código convertido, selecione Elementos do formato Datetype que você usa no código Oracle são semelhantes às strings de formato de data e hora no Amazon Redshift. Nesse caso, o código convertido funciona mais rápido.

O modelo de formato numérico no Oracle inclui mais elementos em comparação com strings de formato numérico no Amazon Redshift. Quando seu código-fonte inclui somente elementos de formato numérico compatíveis com o Amazon Redshift, você não precisa das funções do pacote de extensões no código convertido. Para evitar o uso das funções do pacote de extensões no código convertido, selecione Elementos de formato numérico que você usa no código Oracle são semelhantes às strings de formato numérico no Amazon Redshift. Nesse caso, o código convertido funciona mais rápido.

- Para converter funções analíticas LEAD e LAG do Oracle. Por padrão, a AWS SCT gera um item de ação para cada função LEAD e LAG.

Quando seu código-fonte não usa os valores padrão para offset nessas funções, a AWS SCT pode emular o uso dessas funções com a função NVL. Para fazer isso, selecione Usar a função NVL para emular o comportamento das funções LEAD e LAG do Oracle.

- Para emular o comportamento das chaves primárias e exclusivas em seu cluster do Amazon Redshift, selecione Emular o comportamento das chaves primárias e exclusivas.

O Amazon Redshift não impõe chaves exclusivas e primárias e as usa apenas para fins informativos. Se você usar essas restrições em seu código, certifique-se de que a AWS SCT emula o comportamento delas no código convertido.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do data warehouse Oracle para o Amazon Redshift

Para editar as configurações de otimização de conversão do data warehouse do Oracle para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para conversão do data warehouse do Oracle para o Amazon Redshift.

As configurações de otimização de conversão do data warehouse do Oracle para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a

proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.

- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o Snowflake como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Snowflake para o Amazon Redshift.

Privilégios do Snowflake como um banco de dados de origem

Você pode criar uma função com privilégios e conceder a essa função o nome de um usuário usando a função SECURITYADMIN e o contexto da sessão SECURITYADMIN.

O exemplo a seguir cria privilégios mínimos e os concede ao usuário `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;
```

No exemplo anterior, substitua os espaços reservados conforme a seguir:

- Substitua *role_name* pelo nome de uma função com privilégios somente leitura.
- Substitua *db_name* pelo nome do banco de dados de origem.
- Substitua *schema_name* pelo nome do esquema de origem.
- Substitua *datawarehouse_name* pelo nome de um data warehouse necessário.
- Substitua *min_privs* pelo nome de um usuário que tenha privilégios mínimos.

Os parâmetros DEFAULT_ROLE e DEFAULT_WAREHOUSE são sensíveis a maiúsculas e minúsculas.

Como configurar o acesso seguro ao Amazon S3

As políticas de gerenciamento de segurança e acesso para um bucket do Amazon S3 permitem que o Snowflake acesse, leia e grave dados no bucket do S3. Você pode configurar o acesso seguro a um bucket privado do Amazon S3 usando o tipo de objeto STORAGE INTEGRATION do Snowflake. Um objeto de integração de armazenamento do Snowflake delega a responsabilidade de autenticação a uma entidade de gerenciamento de identidade e acesso do Snowflake.

Para obter mais informações, consulte [Como configurar uma integração de armazenamento do Snowflake para acessar o Amazon S3](#) na documentação do Snowflake.

Como se conectar ao Snowflake como origem

Use o procedimento a seguir para conectar seu banco de dados de origem com a AWS Schema Conversion Tool.

Para se conectar a um banco de dados de origem do Snowflake

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Snowflake e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do data warehouse do Snowflake de origem, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Insira o nome do banco de dados do Snowflake.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT armazena a senha em um formato criptografado somente se você solicitar explicitamente.</p>
Usar SSL	<p>Escolha essa opção se você quiser usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Caminho da chave privada: a localização de uma chave privada.

Parâmetro	Ação
	<ul style="list-style-type: none"> Senha: A frase secreta da chave privada. <p>Para obter mais informações sobre o suporte a SSL para Snowflake, consulte Configurar opções de segurança para conexões.</p>
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Se você definir essa opção, poderá armazenar a senha do banco de dados. Isso significa que você pode se conectar rapidamente ao banco de dados sem precisar digitar a senha.
Caminho do driver Snowflake	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Limitações do Snowflake como origem

A seguir estão relacionadas as limitações de uso do Snowflake como origem para a AWS SCT:

- Os identificadores de objeto devem ser exclusivos dentro do contexto do tipo de objeto e do objeto pai:

Banco de dados

Os identificadores do esquema devem ser exclusivos dentro de um banco de dados.

Esquemas

Os identificadores de objetos, como tabelas e visualizações, devem ser exclusivos em um esquema.

Tabelas/Visualizações

Os identificadores de colunas devem ser exclusivos em uma tabela.

- O número máximo de tabelas para tipos de nó de cluster large e xlarge é 9.900. O número máximo de tabelas para tipos de nó de cluster 8xlarge é 100.000. O limite inclui tabelas temporárias, tanto definidas pelo usuário ou criadas pelo Amazon Redshift durante o processamento de consultas ou a manutenção do sistema. Para obter mais informações, consulte [Cotas do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.
- Para procedimentos armazenados, o número máximo de argumentos de entrada e saída é 32.

Tipos de dados de origem do Snowflake

A seguir, você pode encontrar os tipos de dados de origem do Snowflake que são compatíveis com o uso da AWS SCT e o mapeamento padrão para um destino do Amazon Redshift.

Tipos de dados do Snowflake	Tipos de dados do Amazon Redshift
NUMBER	NUMERIC(38)
NUMBER(p)	Se p for =< 4, então SMALLINT Se p for => 5 e =< 9, então INTEGER Se p for => 10 e =< 18, então BIGINT Se p for => 19, então NUMERIC(p)
NUMBER(p, 0)	Se p for =< 4, então SMALLINT Se p for => 5 e =< 9, então INTEGER Se p for => 10 e =< 18, então BIGINT Se p for => 19, então: NUMERIC(p,0)

Tipos de dados do Snowflake	Tipos de dados do Amazon Redshift
NUMBER(p, s)	Se p for => 1 e =< 38, e se s for => 1 e =< 37, então NUMERIC (p,s)
FLOAT	FLOAT
TEXT Caracteres Unicode de até 16.777.216 bytes; até 4 bytes por caractere.	VARCHAR(MAX)
TEXT(p) Caracteres Unicode de até 65.535 bytes; até 4 bytes por caractere.	Se p for =< 65.535 então, VARCHAR(p)
TEXT(p) Caracteres Unicode de até 16.777.216 bytes; até 4 bytes por caractere.	Se p for => 65.535 e =< 16.777.216 então, VARCHAR(MAX)
BINARY Caracteres de byte único de até 8.388.608 bytes; 1 byte por caractere.	VARCHAR(MAX)
BYNARY(p) Caracteres de byte único de até 65.535 bytes; 1 byte por caractere.	VARCHAR(p)
BYNARY(p) Caracteres de byte único de até 8.388.608 bytes; 1 byte por caractere.	VARCHAR(MAX)
BOOLEAN	BOOLEAN
DATE	DATE

Tipos de dados do Snowflake	Tipos de dados do Amazon Redshift
TIME	VARCHAR(18)
Valores de tempo entre 00:00:00 e 23:59:59.999999999.	
TIME(f)	VARCHAR (n) – 9 + dt-attr-1
Valores de tempo entre 00:00:00 e 23:59:59.9(f).	
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

Configurações de conversão do Snowflake para o Amazon Redshift

Para editar as configurações de conversão do Snowflake para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Snowflake e, em seguida, escolha Snowflake — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para a conversão do Snowflake para o Amazon Redshift.

As configurações de conversão do Snowflake para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do Snowflake para o Amazon Redshift

Para editar as configurações de otimização de conversão do Snowflake para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, selecione Configurações de conversão. Na lista superior, escolha Snowflake e, em seguida, escolha Snowflake — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para a conversão do Snowflake para o Amazon Redshift.

As configurações de otimização de conversão do Snowflake para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.

- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o data warehouse do Microsoft SQL Server como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Microsoft SQL Server DW para o Amazon Redshift ou o Amazon Redshift e AWS Glue usados em combinação.

Privilégios do data warehouse do Microsoft SQL Server como origem

Os privilégios obrigatórios para o data warehouse do Microsoft SQL Server como origem são listados a seguir:

- VIEW DEFINITION
- VIEW DATABASE STATE
- SELECT ON SCHEMA :: *<schema_name>*

No exemplo anterior, substitua o espaço reservado *<source_schema>* pelo nome do source_schema de origem.

Repetir a concessão para cada banco de dados cujo esquema que você está convertendo.

Além disso, conceda o seguinte e execute a concessão no banco de dados mestre:

- VIEW SERVER STATE

Limitações de uso do data warehouse do SQL Server como origem

Atualmente, não há suporte para o uso do data warehouse do Microsoft SQL Server Parallel (PDW) como origem.

Como se conectar ao data warehouse do SQL Server como origem

Use o procedimento a seguir para se conectar ao banco de dados de origem do data warehouse do SQL Server com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem do data warehouse do SQL Server

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Microsoft SQL Server e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do data warehouse de origem do Microsoft SQL Server, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.

Parâmetro	Ação
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Nome da instância	Digite o nome da instância do data warehouse do SQL Server.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none">• Certificado de servidor confiável: selecione essa opção para confiar no certificado de servidor.• Armazenamento de confiança: um armazenamento de confiança que você configura nas Configurações globais.
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.

Parâmetro	Ação
Caminho do driver SQL Server	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de conversão do data warehouse do SQL Server para o Amazon Redshift

Para editar as configurações de conversão do data warehouse do SQL Server para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Microsoft SQL Server e, em seguida, escolha Microsoft SQL Server — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do data warehouse do SQL Server para o Amazon Redshift.

As configurações de conversão do data warehouse do SQL Server para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do data warehouse do SQL Server para o Amazon Redshift

Para editar as configurações de otimização de conversão do data warehouse do SQL Server para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Microsoft SQL Server e, em seguida, escolha Microsoft SQL Server – Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para conversão do data warehouse do SQL Server para o Amazon Redshift.

As configurações de otimização de conversão do data warehouse do SQL Server para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o Teradata como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Teradata para o Amazon Redshift ou Amazon Redshift e AWS Glue usados em combinação.

Privilégios do Teradata como origem

Os privilégios obrigatórios para Teradata como origem são listados a seguir:

- SELECT ON DBC
- SELECT ON SYSUDTLIB
- SELECT ON SYSLIB
- SELECT ON *<source_database>*
- CRIAR PROCEDIMENTO EM *<source_database>*

No exemplo anterior, substitua o espaço reservado *<source_database>* pelo nome do banco de dados de origem.

A AWS SCT requer o privilégio CREATE PROCEDURE para executar HELP PROCEDURE em todos os procedimentos no banco de dados de origem. A AWS SCT não usa esse privilégio para criar novos objetos no banco de dados do Teradata de origem.

Como se conectar ao Teradata como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Teradata com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem Teradata

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.
2. Escolha Teradata e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:

1. Em Segredo da AWS, escolha o nome do seu segredo.
2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do Teradata de origem, use as instruções a seguir:

Parâmetro	Ação
Connection name (Nome da conexão)	Insira um nome para seu banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Digite o nome do banco de dados Teradata.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessário.</p>

Parâmetro	Ação
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.
Criptografar dados	Escolha esta opção para criptografar os dados que trocam com o banco de dados. Se você escolher essa opção, o número da porta 443 será usado para transferir dados criptografados entre a AWS SCT e seu banco de dados do Teradata.
Caminho do driver Teradata	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

- Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Usar autenticação LDAP com um Teradata de origem

Para configurar a autenticação Lightweight Directory Access Protocol (LDAP) para usuários Teradata que executam o Microsoft Active Directory no Windows, use o procedimento a seguir.

No procedimento a seguir, o domínio do Active Directory é `test.local.com`. O servidor do Windows é DC e está configurado com as configurações padrão. O script a seguir cria a conta no Active Directory `test_ldap` e essa conta usa a senha `test_ldap`.

Para configurar a autenticação LDAP para usuários do Teradata que executam o Microsoft Active Directory no Windows

1. No diretório `/opt/teradata/tdat/tdgss/site`, edite o arquivo `TdgssUserConfigFile.xml`. Altere a seção LDAP para o seguinte:

```
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 as alterações executando a configuração a seguir.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Teste a configuração executando o comando a seguir.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

A saída deve ser semelhante a esta.

```
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 o TPA usando o comando a seguir.

```
#tpareset -f "use updated TDGSSCONFIG GD0"
```

5. Crie o mesmo usuário no banco de dados Teradata como no Active Directory, conforme mostrado a seguir.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;  
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Se você alterar a senha de usuário no Active Directory do seu usuário LDAP, especifique essa nova senha durante a conexão ao Teradata em modo LDAP. No modo PADRÃO, você se conecta ao Teradata usando o nome de usuário LDAP e qualquer senha.

Como configurar a coleta de estatísticas em seu data warehouse do Teradata de origem

Para converter seu data warehouse do Teradata de origem, a AWS SCT usa estatísticas para otimizar seu data warehouse convertido do Amazon Redshift. Você pode coletar estatísticas em AWS SCT ou fazer o upload do arquivo de estatísticas. Para obter mais informações, consulte [Como coletar ou carregar as estatísticas](#).

Para garantir que a AWS SCT possa coletar estatísticas do seu data warehouse, conclua as seguintes tarefas de pré-requisito.

Para coletar estatísticas do seu data warehouse do Teradata

1. Execute a consulta a seguir para recuperar as estatísticas de todas as tabelas em seu data warehouse.

```
collect summary statistics on table_name;
```

No exemplo anterior, substitua *table_name* pelo nome da sua tabela de origem. Repita a consulta para cada tabela que você converter.

2. Execute a consulta a seguir para determinar a string da conta do usuário, que você usa para converter seu data warehouse.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Ative o registro de consultas para um usuário específico usando a string da conta do exemplo anterior.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=(' $M$BUSI$$D$H');
```

Como alternativa, ative o registro de consultas para todos os usuários do banco de dados.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Depois de concluir a coleta das estatísticas do data warehouse, desative o registro de consultas. Para fazer isso, você pode usar o exemplo de código a seguir.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

Como coletar estatísticas em modo off-line do seu data warehouse do Teradata de origem

Depois de configurar a coleta de estatísticas em seu data warehouse do Teradata, você pode coletar estatísticas em seu projeto da AWS SCT. Como alternativa, você pode usar scripts Basic Teradata Query (BTEQ) para coletar estatísticas em um modo off-line. Em seguida, você pode fazer o upload dos arquivos com as estatísticas coletadas para o seu projeto da AWS SCT. Para obter mais informações, consulte [Como coletar ou carregar as estatísticas](#).

Para coletar estatísticas do seu data warehouse do Teradata em um modo off-line

1. Crie o script `off-line_stats.bteq` com o conteúdo a seguir.

```
.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_count"
FROM (select databasename, tablename
      from DBC.tablesv
      where tablekind IN ('T','0')
      and databasename = 'your_database_name'
      ) c
left join
    (select DatabaseName, TableName, max(RowCount) RowCount
     from dbc.tableStatsv
     group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
    (SELECT r.ChildDB as DatabaseName,
     r.ChildTable as TableName,
     COUNT(DISTINCT r.ParentTable) reference_count
     FROM DBC.All_RI_ChildrenV r
     GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
    (SELECT r.ParentDB as DatabaseName,
     r.ParentTable as TableName,
     COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
     FROM DBC.All_RI_ParentsV r
     GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
    (select databasename, tablename,
     sum(currentperm)/1024/1024 as size_in_mb
     from dbc.TableSizeV

```

```

        group by 1,2) w
on w.databasesname = c.databasesname
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.dbqsqltbl q
    INNER JOIN
    (
    SELECT QueryId
    FROM DBC.DBQLogTbl t
    WHERE TRIM(t.StatementType) IN ('SELECT')
    AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
    AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
    TRIM(t.USERNAME) END) IN ('All users') --user_name list
    AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
    GROUP BY 1
    ) t
    ON q.QueryId = t.QueryId
    INNER JOIN
    (

```

```

SELECT QueryId
FROM DBC.QryLogObjectsV
WHERE ObjectDatabaseName = 'your_database_name'
AND ObjectType = 'Tab'
AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
GROUP BY 1
) r
ON r.QueryId = t.QueryId
WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
) t
) t
WHERE SqlTextInfo NOT LIKE '%"%"'
) q
WHERE
QueryParts >=1
AND topN <= 50
ORDER BY QueryCount DESC, QueryId, SqlRowNo
QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

.EXPORT RESET

.LOGOFF

.QUIT

```

2. Crie o arquivo `td_run_bteq.bat` que executa o script BTEQ que você criou na etapa anterior. Use o conteúdo a seguir para esse arquivo.

```

@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. Crie o arquivo `runme.bat` que executa o arquivo de lote que você criou na etapa anterior. Use o conteúdo a seguir para esse arquivo.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

No arquivo `runme.bat`, substitua *ServerName*, *UserName* e *DatabaseName* pelos valores aplicáveis.

Em seguida, execute o arquivo `runme.bat`. Repita essa etapa para cada data warehouse que você converter para o Amazon Redshift.

Depois de executar esse script, você recebe três arquivos com estatísticas para cada banco de dados. Você pode fazer o upload desses arquivos para o seu projeto da AWS SCT. Para fazer isso, escolha seu data warehouse no painel esquerdo do projeto e abra o menu contextual (clique com o botão direito do mouse). Escolha Fazer upload estatísticas.

Configurações de conversão do Teradata para o Amazon Redshift

Para editar as configurações de conversão do Teradata para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Teradata e, em seguida, escolha Teradata — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão do Teradata para o Amazon Redshift.

As configurações de conversão do Teradata para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compactação.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

- Para usar uma lista explícita de colunas no código convertido para as instruções SELECT *, selecione Usar instrução de coluna explícita.
- Para emular o comportamento das chaves primárias e exclusivas em seu cluster do Amazon Redshift, selecione Emular o comportamento das chaves primárias e exclusivas.

O Amazon Redshift não impõe chaves exclusivas e primárias e as usa apenas para fins informativos. Se você usar essas restrições em seu código, certifique-se de que a AWS SCT emula o comportamento delas no código convertido.

- Para garantir a exclusividade dos dados nas tabelas do Amazon Redshift de destino. Para fazer isso, selecione Emular o comportamento das tabelas SET.

O Teradata cria tabelas usando o elemento de sintaxe SET como opção padrão. Você não pode adicionar linhas duplicadas em uma tabela SET. Se o código-fonte não usar essa restrição de exclusividade, desative essa opção. Nesse caso, o código convertido funciona mais rápido.

Se o código-fonte usa a opção SET em tabelas como uma restrição de exclusividade, ative essa opção. Nesse caso, a AWS SCT reescreve as instruções INSERT . . SELECT no código convertido para emular o comportamento do seu banco de dados de origem.

Configurações de otimização de conversão do Teradata para o Amazon Redshift

Para editar as configurações de otimização de conversão do Teradata para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Teradata e, em seguida, escolha Teradata — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para conversão do Teradata para o Amazon Redshift.

As configurações de otimização de conversão do Teradata para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.
- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e

Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como usar o Vertica como origem para AWS SCT

Você pode usar a AWS SCT para converter esquemas, objetos de código e código de aplicativo do Vertica para o Amazon Redshift.

Privilégios do Vertica como origem

Os privilégios obrigatórios para o Vertica como origem são listados a seguir:

- 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)>*

No exemplo anterior, substitua os espaços reservados conforme a seguir:

- Substitua *schema_name* pelo nome do esquema de origem.
- Substitua *procedure_name* pelo nome de um procedimento de origem. Repita a concessão para cada procedimento que você está convertendo.
- Substitua *procedure_signature* pela lista delimitada por vírgula dos tipos de argumentos de procedimento.

Como se conectar ao Vertica como origem

Use o procedimento a seguir para conectar seu banco de dados de origem do Vertica com a AWS Schema Conversion Tool.

Para conectar-se a um banco de dados de origem Vertica

1. Na AWS Schema Conversion Tool, escolha Adicionar origem.

2. Escolha Vertica e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

3. Em Nome da conexão, insira um nome para o banco de dados. A AWS SCT exibe esse nome na árvore no painel esquerdo.
4. Use as credenciais do banco de dados da AWS Secrets Manager ou insira-as manualmente:
 - Para usar as credenciais do banco de dados do Secrets Manager, use as instruções a seguir:
 1. Em Segredo da AWS, escolha o nome do seu segredo.
 2. Escolha Preencher para preencher automaticamente todos os valores na caixa de diálogo de conexão do banco de dados do Secrets Manager.

Para obter informações sobre o uso de credenciais de banco de dados do Secrets Manager, consulte a [Usar o AWS Secrets Manager](#).

- Para inserir manualmente as informações de conexão do banco de dados do Vertica de origem, use as instruções a seguir:

Parâmetro	Ação
Server name	Digite o Sistema de Nomes de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Database	Digite o nome do banco de dados Vertica.
Nome de usuário e Senha	<p>Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.</p> <p>A AWS SCT usa a senha para se conectar ao seu banco de dados de origem somente quando você escolhe se conectar ao seu banco de dados em um projeto. Para proteger-se contra a exposição da senha do banco de dados de origem, a AWS SCT não armazena a senha por padrão. Se fechar o projeto da AWS SCT e abri-lo novamente, você será solicitado a informar a senha para se</p>

Parâmetro	Ação
	conectar ao banco de dados de origem, conforme necessário.
Usar SSL	<p>Escolha essa opção para usar Secure Sockets Layer (SSL) para se conectar ao banco de dados. Forneça as seguintes informações adicionais, conforme aplicável, na guia SSL:</p> <ul style="list-style-type: none"> • Verificar certificado do servidor: Escolha essa opção para verificar o certificado do servidor, usando um armazenamento de segurança. • Armazenamento de confiança: um armazenamento de confiança que você configura nas Configurações globais. • Armazenamento de chaves: um armazenamento de chaves que você configura nas Configurações globais.
Armazenar senha	A AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas do banco de dados. Ao habilitar essa opção, você pode armazenar a senha do banco de dados e se conectar rapidamente a ele sem a necessidade de informar a senha.
Caminho do driver Vertica	<p>Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários.</p> <p>Se você armazenar o caminho do driver nas configurações globais do projeto, o caminho do driver não aparecerá na caixa de diálogo de conexão. Para obter mais informações, consulte Armazenamento de caminhos do driver nas configurações globais.</p>

5. Escolha Testar conexão para verificar se a AWS SCT pode se conectar ao banco de dados de origem.
6. Escolha Conectar para se conectar ao banco de dados de origem.

Configurações de conversão do Vertica para o Amazon Redshift

Para editar as configurações de conversão do Vertica para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Vertica e, em seguida, escolha Vertica — Amazon Redshift. A AWS SCT exibe todas as configurações disponíveis para conversão de Vertica para Amazon Redshift.

As configurações de conversão do Vertica para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para limitar o número de comentários com itens de ação no código convertido.

Em Adicionar comentários no código convertido para os itens de ação de severidade selecionada e superior, escolha a severidade dos itens de ação. A AWS SCT adiciona comentários no código convertido para itens de ação da severidade selecionada e superior.

Por exemplo, para minimizar o número de comentários em seu código convertido, escolha Somente erros. Para incluir comentários para todos os itens de ação em seu código convertido, escolha Todas as mensagens.

- Para definir o número máximo de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift de destino.

Em O número máximo de tabelas para o cluster do Amazon Redshift de destino, escolha o número de tabelas que a AWS SCT pode aplicar ao seu cluster do Amazon Redshift.

O Amazon Redshift tem cotas que limitam as tabelas de uso para diferentes tipos de nós de cluster. Se você escolher Auto, a AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster do Amazon Redshift de destino, dependendo do tipo de nó. Opcionalmente, escolha o valor manualmente. Para obter mais informações, consulte [“Cotas e limites no Amazon Redshift”](#) no Guia de gerenciamento de clusters do Amazon Redshift.

A AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. A AWS SCT armazena o código convertido em seu projeto e não o aplica ao banco de dados de destino. Se você atingir a cota de cluster do Amazon Redshift para as tabelas ao aplicar o código convertido, a AWS SCT exibirá uma mensagem de aviso. Além disso, a AWS SCT aplica tabelas ao seu cluster do Amazon Redshift de destino até que o número de tabelas atinja o limite.

- Para migrar partições da tabela de origem para tabelas separadas no Amazon Redshift. Para fazer isso, selecione Usar a exibição UNION ALL e insira o número máximo de tabelas de destino que a AWS SCT pode criar para uma única tabela de origem.

O Amazon Redshift não oferece suporte ao particionamento de tabelas. Para emular esse comportamento e acelerar a execução das consultas, a AWS SCT pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, a AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

A AWS SCT determina automaticamente o número de partições na tabela de origem. Dependendo do tipo de particionamento da tabela de origem, esse número pode exceder a cota das tabelas que você pode aplicar ao seu cluster do Amazon Redshift. Para evitar atingir essa cota, insira o número máximo de tabelas de destino que a AWS SCT pode criar para partições de uma única tabela de origem. A opção padrão é 368 tabelas, que representam uma partição para 366 dias do ano e duas tabelas para partições NO RANGE e UNKNOWN.

- Para aplicar compactação às colunas da tabela do Amazon Redshift. Para fazer isso, selecione Usar codificação de compressão.

A AWS SCT atribui a codificação de compactação às colunas automaticamente usando o algoritmo padrão do Amazon Redshift. Para obter mais informações, consulte [Codificações de compactação](#) no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Por padrão, o Amazon Redshift não aplica compactação a colunas definidas como chaves de classificação e distribuição. É possível alterar esse comportamento e aplicar compactação a essas colunas. Para fazer isso, selecione Usar codificação de compactação para colunas KEY. Você pode selecionar essa opção somente ao selecionar a opção Usar codificação de compactação.

Configurações de otimização de conversão do Vertica para o Amazon Redshift

Para editar as configurações de otimização de conversão do Vertica para o Amazon Redshift, escolha Configurações em AWS SCT e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Vertica e, em seguida, escolha Vertica — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. A AWS SCT exibe as configurações de otimização de conversão para a conversão do Vertica para o Amazon Redshift.

As configurações de otimização de conversão do Vertica para o Amazon Redshift em AWS SCT incluem opções para o seguinte:

- Para trabalhar com otimização automática de tabelas. Para fazer isso, selecione Usar o ajuste automático da tabela do Amazon Redshift.

A otimização automática de tabelas é um processo de autoajuste no Amazon Redshift que otimiza automaticamente o design de tabelas. Para obter mais informações, consulte [Como trabalhar com a otimização automática de tabelas](#) no Guia do desenvolvedor do banco de dados Amazon Redshift.

Para confiar somente na otimização automática da tabela, escolha Nenhuma para a Estratégia inicial de seleção de chave.

- Para escolher chaves de classificação e distribuição usando sua estratégia.

Você pode escolher chaves de classificação e distribuição usando metadados, informações estatísticas ou ambas as opções do Amazon Redshift. Para Estratégia inicial de seleção de chave na guia Estratégias de otimização, escolha uma das seguintes opções:

- Use metadados, ignore informações estatísticas
- Ignore metadados, use informações estatísticas
- Use metadados e informações estatísticas

Dependendo da opção escolhida, você pode selecionar estratégias de otimização. Em seguida, para cada estratégia, insira o valor (0-100). Esses valores definem o peso de cada estratégia. Usando esses valores de peso, a AWS SCT define como cada regra influencia na escolha das chaves de distribuição e classificação. Os valores padrão são baseados nas melhores práticas de migração da AWS.

Você pode definir o tamanho de tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para defini-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

- Para configurar os detalhes da estratégia.

Além de definir o peso de cada estratégia de otimização, você pode definir as configurações de otimização. Para fazer isso, escolha Otimização de conversões.

- Em Limite de colunas da chave de classificação, insira o número máximo de colunas na chave de classificação.

- Em Valor limite distorcido, insira a porcentagem (0-100) de um valor distorcido para uma coluna. A AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. A AWS SCT define o valor distorcido de uma coluna como a proporção percentual entre o número de ocorrências do valor mais comum e o número total de registros.
- Para as N principais consultas da tabela do histórico de consultas, insira o número (1-100) das consultas usadas com mais frequência para análise.
- Em Selecionar usuário de estatísticas, escolha o usuário do banco de dados para o qual você deseja analisar as estatísticas de consulta.

Além disso, na guia Estratégias de otimização, você pode definir o tamanho das tabelas pequenas para a estratégia Localizar tabelas pequenas. Em Contagem mínima de linhas da tabela e Contagem máxima de linhas da tabela, insira o número mínimo e máximo de linhas em uma tabela para considerá-la como uma tabela pequena. A AWS SCT aplica o estilo de distribuição ALL a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Criação de regras de mapeamento na AWS SCT

Você pode adicionar vários bancos de dados de origem e de destino em um único projeto AWS SCT. Isso simplifica o gerenciamento de projetos quando você migra vários bancos de dados para diferentes plataformas de destino.

Depois de criar um novo projeto e adicionar bancos de dados de origem e destino, crie regras de mapeamento. AWS SCT requer pelo menos uma regra de mapeamento para criar um relatório de avaliação de migração e converter esquemas de banco de dados.

Uma regra de mapeamento descreve um par de origem e destino que inclui um esquema de banco de dados de origem ou um banco de dados de origem e uma plataforma de banco de dados de destino. É possível adicionar várias regras de mapeamento a um único projeto da AWS SCT. Use regras de mapeamento para converter cada esquema de banco de dados de origem na plataforma de banco de dados de destino correta.

Para alterar o nome do seu esquema no código convertido, configure uma regra de migração. Por exemplo, com as regras de migração, você pode renomear seu esquema, adicionar um prefixo aos nomes dos objetos, alterar o agrupamento de colunas ou alterar os tipos de dados. Para aplicar essas alterações ao seu código convertido, certifique-se de criar regras de migração antes de converter seu esquema de origem. Para obter mais informações, consulte [Como criar regras de migração](#).

Você pode criar regras de mapeamento somente para pares de conversão de banco de dados compatíveis. Para ver a lista de pares de conversão compatíveis, consulte [Origens para a AWS SCT](#).

Se você abrir um projeto salvo na versão 1.0.655 ou anterior da AWS SCT, a AWS SCT criará automaticamente regras de mapeamento para todos os esquemas do banco de dados de origem na plataforma do banco de dados de destino. Para adicionar outras plataformas de banco de dados de destino, exclua as regras de mapeamento existentes e crie novas.

Tópicos

- [Adicionar uma nova regra de mapeamento](#)
- [Gerenciar regras de mapeamento](#)
- [Usar alvos virtuais](#)
- [Limitações ao uso de vários servidores em um único projeto AWS SCT](#)

Adicionar uma nova regra de mapeamento

Você pode criar várias regras de mapeamento em um único projeto. AWS SCT salva regras de mapeamento como parte do projeto. Com seu projeto aberto, use o procedimento a seguir para adicionar novas regras de mapeamento.

Para criar regras de mapeamento

1. No menu Exibir, selecione Exibição de mapeamento.
2. No painel esquerdo, escolha um esquema ou um banco de dados para adicionar à regra de mapeamento.
3. No painel direito, escolha uma plataforma de banco de dados de destino para o esquema de origem ou para o banco de dados selecionado.

Você pode escolher uma plataforma de banco de dados virtual como destino. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Selecione Criar mapeamento.

AWS SCT adiciona essa nova regra de mapeamento à lista de Mapeamentos do servidor.

Adicione regras de mapeamento para todos os pares de conversão. Para criar um relatório de avaliação ou converter esquemas de banco de dados, escolha Visualização principal no menu Exibir.

AWS SCT destaca em negrito todos os objetos do esquema que fazem parte de uma regra de mapeamento.

Gerenciar regras de mapeamento

Você pode filtrar ou excluir as regras de mapeamento existentes e adicionar uma nova regra de mapeamento ao seu projeto AWS Schema Conversion Tool (AWS SCT).

Quando você cria uma regra de mapeamento para todo o banco de dados de origem, AWS SCT cria uma regra de mapeamento para cada esquema do banco de dados de origem. Em projetos que envolvem dezenas de esquemas ou mesmo bancos de dados, pode ser difícil entender qual destino é usado para um determinado esquema. Para encontrar rapidamente uma regra de mapeamento para seu esquema, use uma ou várias das seguintes opções de filtro em AWS SCT.

Para filtrar as regras de mapeamento

1. No menu Exibir, selecione Exibição de mapeamento.
2. Em Servidores de origem, escolha o banco de dados de origem.

O filtro padrão é Tudo, o que significa que AWS SCT exibe regras de mapeamento para todos os bancos de dados de origem.

3. Em Esquema de origem, insira o nome do esquema de origem. Utilize a porcentagem (%) como curinga para substituir qualquer número de quaisquer símbolos no nome do esquema.

O filtro padrão é o curinga %, o que significa que AWS SCT exibe regras de mapeamento para todos os nomes de esquema dos bancos de dados de origem.

4. Em Tem regras de migração, escolha Sim para exibir as regras de mapeamento para as quais as regras de migração de dados foram criadas. Escolha Não para exibir regras de mapeamento que não tenham regras de migração de dados. Para obter mais informações, consulte [Criação de regras de migração de dados no AWS SCT](#).

O filtro padrão é Tudo, o que significa que AWS SCT exibe todas as regras de mapeamento.

5. Em Servidores de destino, selecione o banco de dados de destino.

O filtro padrão é Tudo, o que significa que AWS SCT exibe regras de mapeamento para todos os bancos de dados de destino.

Com seu projeto aberto, use o procedimento a seguir para criar regras de mapeamento. Para obter mais informações sobre adicionar regras de mapeamento, consulte [Adicionar uma nova regra de mapeamento](#).

Para excluir regras de mapeamento

1. No menu Exibir, selecione Exibição de mapeamento.
2. Para Mapeamentos do servidor, escolha as regras de mapeamento a serem excluídas.
3. Escolha Excluir mapeamentos selecionados.

AWS SCT exclui as regras de mapeamento selecionadas.

Usar alvos virtuais

Você pode ver como AWS SCT converte seu esquema de banco de dados de origem em qualquer plataforma de banco de dados de destino compatível. Para fazer isso, você não precisa se conectar a um banco de dados de destino existente. Em vez disso, você pode escolher uma plataforma de banco de dados de destino virtual no painel direito ao criar uma regra de mapeamento. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#). Certifique-se de expandir os Servidores, Clusters NoSQL e nós ETL no painel direito para ver a lista de plataformas virtuais de banco de dados de destino.

AWS SCT suporta as seguintes plataformas virtuais de banco de dados de destino:

- Amazon Aurora Edição Compatível com MySQL
- Amazon Aurora Edição Compatível com PostgreSQL
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift e AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish for Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Se você usa o Babelfish para Aurora PostgreSQL como plataforma de banco de dados de destino, só poderá criar um relatório de avaliação da migração do banco de dados. Para obter mais informações, consulte [the section called “Relatórios de avaliação de migração”](#).

Se você usar uma plataforma virtual de banco de dados de destino, poderá salvar o código convertido em um arquivo. Para obter mais informações, consulte [the section called “Como salvar o esquema convertido”](#).

Limitações ao uso de vários servidores em um único projeto AWS SCT

As limitações a seguir se aplicam ao converter esquemas usando vários servidores em um único projeto AWS SCT:

- Você pode adicionar o mesmo servidor a um projeto somente uma vez.
- Você não pode mapear esquemas de servidor para um esquema de destino específico, somente para um servidor de destino. AWS SCT cria o esquema de destino durante a conversão.
- Você não pode mapear objetos de origem de nível inferior para o servidor de destino.
- Você pode mapear um esquema de origem para somente um servidor de destino em um projeto.
- Certifique-se de mapear uma fonte para um servidor de destino para criar um relatório de avaliação, converter esquemas ou extrair dados.

Criar relatórios de conversão

Quando você está planejando uma conversão de banco de dados, é útil criar alguns relatórios para ajudá-lo a entender o que está envolvido. É possível criar relatórios usando a AWS Schema Conversion Tool.

Você pode usar o AWS SCT para criar um relatório de avaliação de migração de banco de dados. Com esse relatório, você obtém um resumo de suas tarefas de conversão de esquemas e os detalhes dos itens que não podem ser convertidos automaticamente para o banco de dados de destino. Você pode usar esse relatório para avaliar quanto do projeto pode ser concluído usando o AWS SCT e o que mais é necessário para concluir a conversão. Para criar um relatório de avaliação, use Create Report (Criar relatório) no menu de contexto (clique com o botão direito do mouse) do banco de dados no AWS SCT.

Tópicos

- [Como criar relatórios de avaliação de migração com a AWS SCT](#)

Como criar relatórios de avaliação de migração com a AWS SCT

Uma parte importante da AWS Schema Conversion Tool é o relatório de avaliação que ela gera para estimar a complexidade da conversão do esquema. Este relatório de avaliação de migração do banco de dados resume todas as tarefas de conversão de esquema e detalha os itens de ação do esquema que não podem ser convertidos automaticamente no mecanismo de banco de dados de sua instância de banco de dados de destino. É possível visualizar o relatório no aplicativo ou exportá-lo como um valor separado por vírgula (CSV) ou arquivo PDF.

Se você adicionar vários bancos de dados de origem e destino em um único projeto, a AWS SCT agregará os relatórios de todos os pares de conversão em um relatório de avaliação da migração do banco de dados.

Você pode usar plataformas de banco de dados de destino virtual para gerar um relatório de avaliação e entender a complexidade da migração para uma plataforma de banco de dados selecionada. Nesse caso, você não precisa se conectar à sua plataforma de banco de dados de destino. Por exemplo, você pode usar o Babelfish para Aurora PostgreSQL como uma plataforma de banco de dados de destino virtual para criar um relatório de avaliação de migração de banco de dados. Para obter mais informações sobre plataformas de banco de dados de destino virtual, consulte [the section called “Alvos virtuais”](#).

O relatório de avaliação de migração inclui o seguinte:

- Resumo executivo
- Licença de avaliação
- Suporte para nuvem, indicando todos os recursos no banco de dados de origem não disponíveis no destino.
- Recomendações, incluindo a conversão de objetos do servidor, sugestões de backup e alterações de servidor vinculadas

O relatório também inclui estimativas do trabalho necessário para gravar o código equivalente na instância de banco de dados de destino que não pode ser convertido automaticamente.

Se você usar a AWS SCT para migrar o esquema existente para uma instância de banco de dados do Amazon RDS, então você pode usar o relatório para ajudá-lo a analisar os requisitos para migrar para a nuvem da AWS e alterar seu tipo de licença.

Tópicos

- [Como criar um relatório de avaliação de migração do banco de dados](#)
- [Como visualizar o relatório de avaliação](#)
- [Como salvar o relatório de avaliação](#)
- [Como configurar o relatório de avaliação](#)
- [Como criar um relatório de avaliação multisservidor para migração de banco de dados](#)

Como criar um relatório de avaliação de migração do banco de dados

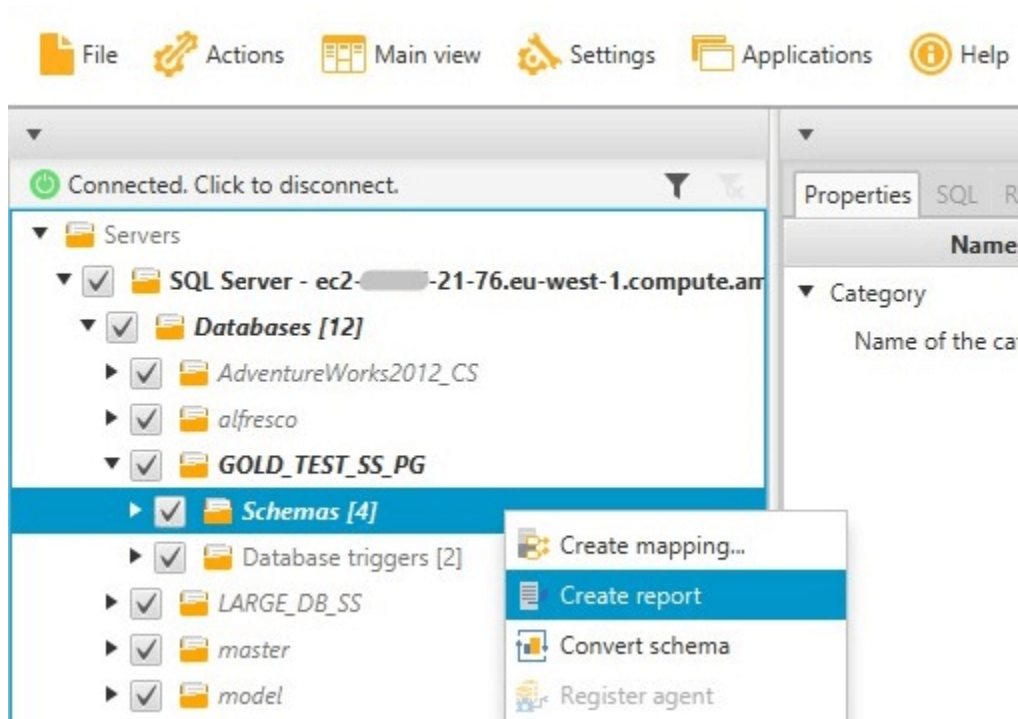
Use o procedimento a seguir para criar um relatório de avaliação de migração do banco de dados.

Para criar um relatório de avaliação de migração do banco de dados

1. Certifique-se de ter criado uma regra de mapeamento para o esquema do banco de dados de origem para o qual criar um relatório de avaliação. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
2. No menu Exibir, selecione Visualização principal.
3. No painel esquerdo que exibe o esquema do banco de dados de origem, escolha um objeto de esquema para o qual criar um relatório de avaliação. Para incluir vários esquemas de banco de dados no relatório, escolha o nó pai, por exemplo, Esquemas.

Certifique-se de ter marcado as caixas de seleção de todos os objetos do esquema para os quais criar um relatório de avaliação.

- Abra o menu de contexto (clique com o botão direito do mouse) do objeto e, em seguida, escolha Criar relatório.



Como visualizar o relatório de avaliação

Depois que você criar um relatório de avaliação, a visualização do relatório de avaliação será aberta, exibindo as seguintes abas:

- Resumo
- Action Items (Itens de ação)

A guia Summary (Resumo) mostra os itens que foram convertidos automaticamente ou não convertidos.

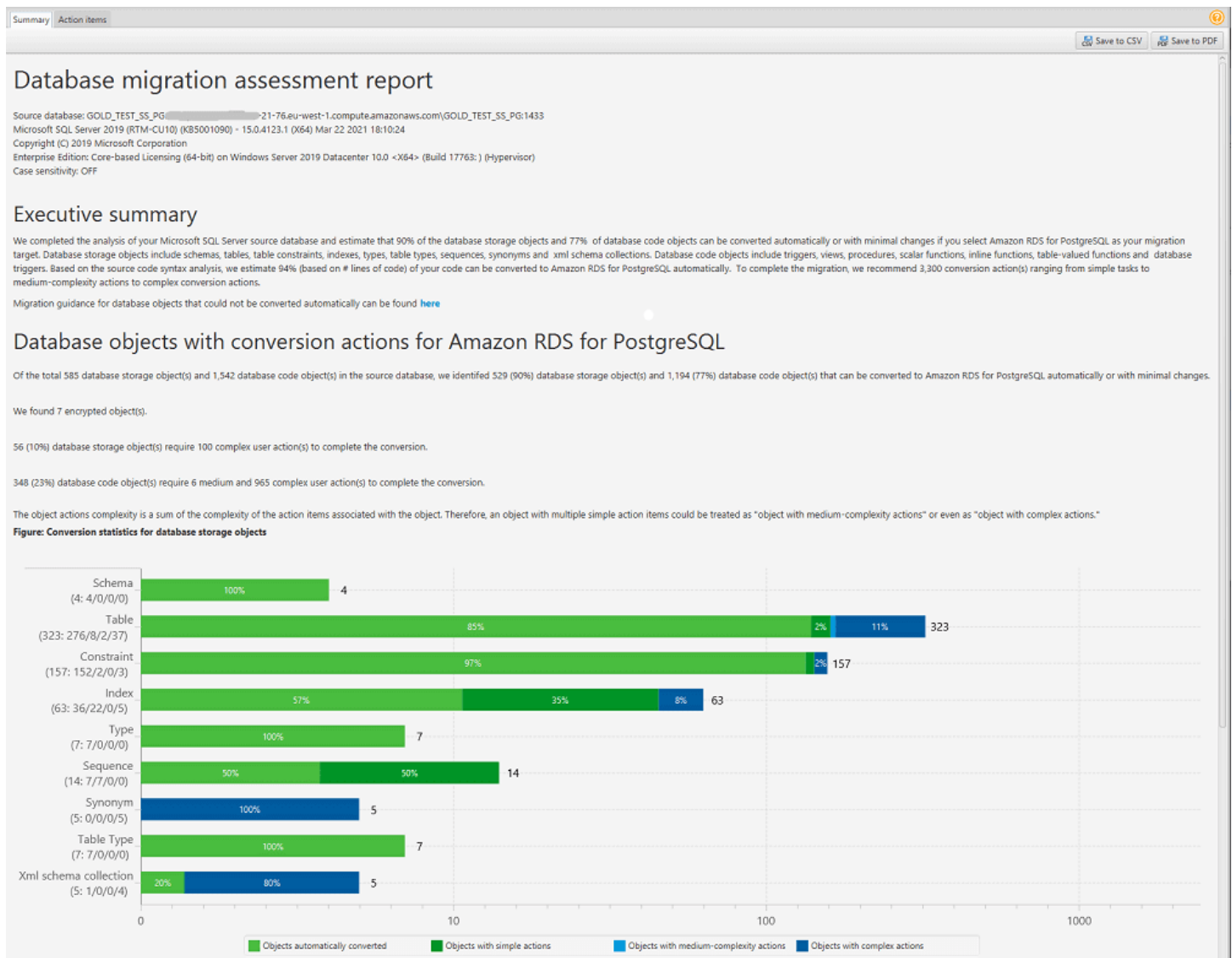
A guia Action Items (Itens de ação) mostra os itens que não puderam ser convertidos automaticamente e as recomendações sobre o que fazer com eles.

Tópicos

- [Resumo do relatório de avaliação](#)
- [Itens de ação do relatório de avaliação](#)
- [Mensagem de aviso do relatório de avaliação](#)

Resumo do relatório de avaliação

A guia Resumo exibe as informações de resumo do relatório de avaliação de migração do banco de dados. Ela mostra os itens que foram convertidos automaticamente e os itens que não foram convertidos automaticamente.



Para os itens de esquema que não podem ser convertidos automaticamente para o mecanismo de banco de dados de destino, o resumo inclui uma estimativa do trabalho necessário para criar itens de esquema na instância de banco de dados de destino que equivalem aos de origem.

O relatório classifica o tempo estimado de conversão desses itens de esquema da seguinte maneira:

- **Simples** – Ações que podem ser concluídas em menos de duas horas.
- **Médio** – Ações mais complexas que podem ser concluídas de duas a seis horas.
- **Significativo** – Ações que são muito complexas e levarão mais de seis horas para serem concluídas.

A seção Avaliação de licença e suporte à nuvem contém informações sobre como mover seu esquema de banco de dados on-premises existente para uma instância de banco de dados do Amazon RDS que executa o mesmo mecanismo. Por exemplo, se você deseja alterar tipos de licença, esta seção do relatório mostra quais recursos do seu banco de dados atual devem ser removidos.

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.

Itens de ação do relatório de avaliação

A visualização do relatório de avaliação também inclui uma guia Itens de ação. Esta guia contém uma lista de itens que não podem ser convertidos automaticamente para o mecanismo de banco de dados de sua instância de banco de dados do Amazon RDS de destino. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu esquema ao qual o item de ação se aplica.

O relatório também contém recomendações sobre como converter manualmente o item do esquema. Por exemplo, após a execução da avaliação, relatórios detalhados do banco de dados/esquema mostram o esforço necessário para projetar e implementar as recomendações para converter itens de ação. Para obter mais informações sobre decidir como lidar com conversões manuais, consulte [Como lidar com conversões manuais na AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, tables, and schemas. The main pane shows a list of action items (issues) with details for each, such as the issue number, description, recommended action, and number of occurrences. A detailed view of a specific issue is shown below the list, including the source SQL code and the target Amazon RDS schema details.

Issue 609: MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required
 Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table.
 Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>

Issue 681: MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically
 Recommended action: Use non-clustered indexes.
 Number of occurrences: 2

Issue 794: MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype
 Recommended action: Please review generated code and modify it if necessary.
 Number of occurrences: 1
 Parameter: @InputPosNo (Number of occurrences: 1)
 MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype

Issue 826: Check the default value for a DateTime variable
 Recommended action: Check the default value for a DateTime variable.
 Number of occurrences: 1

Issue 844: MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision
 Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy.
 Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Issue 9997: Unable to resolve objects
 Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.
 Number of occurrences: 3

Issue 690: MySQL doesn't support table types
 Recommended action: Perform a manual conversion.
 Number of occurrences: 1

Issue 811: Unable to convert functions
 Recommended action: Create a user-defined function.
 Number of occurrences: 12

Source Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK

```

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

```

Target Amazon RDS for MySQL category: Schemas

Name	Value
Category	Schemas
Name of the category	Schemas

Mensagem de aviso do relatório de avaliação

Para avaliar a complexidade da conversão para outro mecanismo de banco de dados, a AWS SCT requer acesso aos objetos em seu banco de dados de origem. Quando o SCT não consegue realizar uma avaliação porque foram encontrados problemas durante o escaneamento, uma mensagem de aviso é emitida indicando que a porcentagem geral de conversão foi reduzida.

Warning!

We found that your source database may be configured not in correct way or you have not enough privileges for reading all necessary metadata. Please check your configuration and run report again. For more details please review [help documentation](#).

List of Action Items to review:

- Issue 9997** Unable to resolve objects (number of occurrences: 3)
 Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.

A seguir estão os motivos pelos quais a AWS SCT pode encontrar problemas durante a digitalização:

- A conta de usuário conectada ao banco de dados não tem acesso a todos os objetos necessários.
- Um objeto citado no esquema não existe mais no banco de dados.
- O SCT está tentando avaliar um objeto criptografado.

Para obter mais informações sobre as permissões e privilégios de segurança necessários pela SCT para o seu banco de dados, consulte [Origens para a AWS SCT](#) na seção apropriada do banco de dados de origem neste guia.

Como salvar o relatório de avaliação

Depois de [criar um relatório de avaliação de migração de banco de dados](#), é possível salvar uma cópia local do relatório de avaliação de migração de banco de dados como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV).

Para salvar o relatório de avaliação de migração de banco de dados como um arquivo PDF

1. No menu superior, selecione Exibir e, em seguida, selecione Visualização do relatório de avaliação.
2. Escolha a guia Resumo.
3. Selecione Salvar em PDF no canto superior direito.

Para salvar o relatório de avaliação de migração de banco de dados como arquivos CSV

1. No menu superior, selecione Exibir e, em seguida, selecione Visualização do relatório de avaliação.
2. Escolha a guia Resumo.
3. Selecione Salvar em CSV no canto superior direito.

O arquivo PDF contém as informações de item de ação e de resumo, conforme mostrado no exemplo a seguir.

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

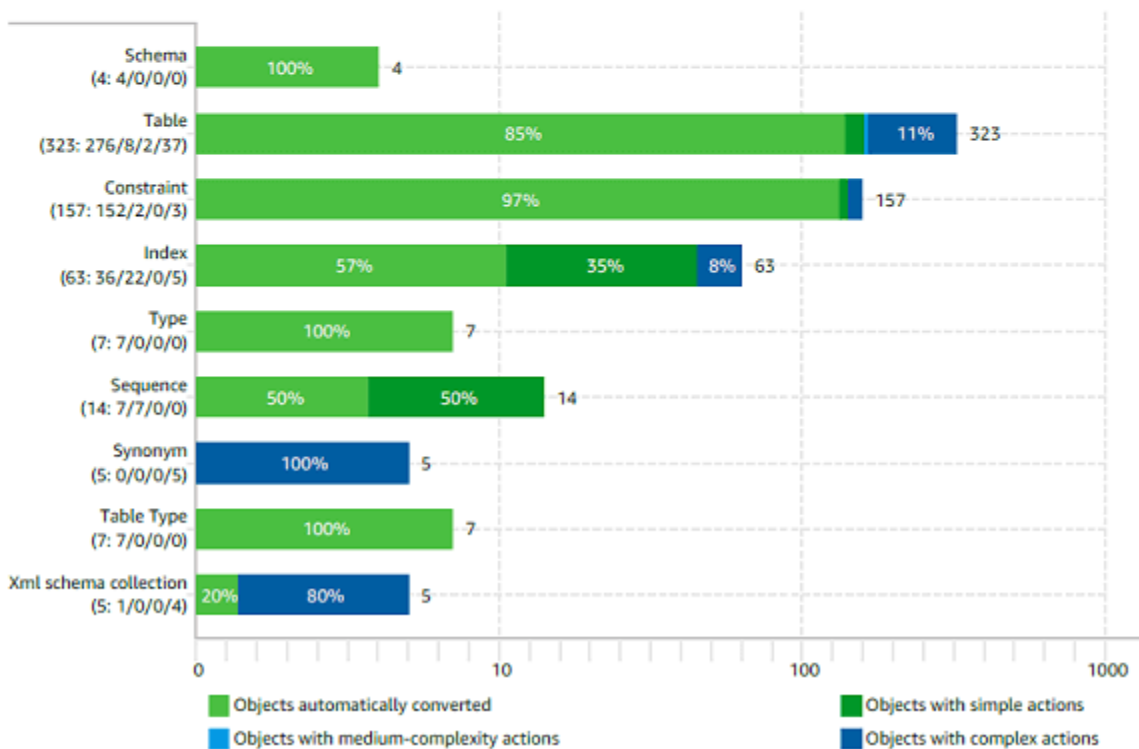
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



Ao escolher a opção Salvar em CSV, a AWS SCT cria três arquivos CSV.

O primeiro arquivo CSV contém as seguintes informações sobre os itens de ação:

- Categoria
- Ocorrência – O nome do arquivo, o número da linha e a posição do item
- Número do item de ação
- Subject
- Grupo

- Descrição
- Referências da documentação
- Ação recomendada
- Complexidade estimada

O segundo arquivo CSV inclui o sufixo `Action_Items_Summary` em seu nome e contém as informações sobre o número de ocorrências de todos os itens de ação.

No exemplo a seguir, os valores na coluna `Esforço da curva de aprendizado` indicam a quantidade de esforço necessária para criar uma abordagem para converter cada item de ação. Os valores na coluna `Esforço para converter uma ocorrência do item de ação` indicam o esforço necessário para converter cada item de ação, seguindo a abordagem projetada. Os valores usados para indicar o nível dos esforços necessários são baseados em uma escala ponderada, que varia de baixo (mínimo) a 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

O terceiro arquivo CSV inclui o `Summary` em seu nome e contém o seguinte resumo:

- Categoria
- Número de objetos
- Objetos convertidos automaticamente
- Objetos com ações simples
- Objetos com ações de complexidade média
- Objetos com ações complexas
- Total de linhas de código

Como configurar o relatório de avaliação

Você pode configurar a quantidade de detalhes que a AWS SCT inclui nos relatórios de avaliação.

Para configurar um relatório de avaliação de migração do banco de dados

1. No menu Configurações, selecione Configurações globais e, em seguida, selecione Relatório de avaliação.
2. Em Ocorrências de itens de ação, selecione Primeiros cinco problemas somente para limitar o número de itens de ação de um único tipo no relatório de avaliação. Selecione Todas os problemas para incluir todos os itens de ação de cada tipo no relatório de avaliação.
3. Em Arquivos analisados por script SQL, selecione Listar não mais do que **X** arquivos para limitar o número de arquivos de script SQL no relatório de avaliação para **X**. Insira o número de arquivos. Selecione Listar todos os arquivos analisados para incluir todos os arquivos de script SQL no relatório de avaliação.
4. Selecione Abrir relatórios após salvar para abrir automaticamente o arquivo depois de salvar uma cópia local do relatório de avaliação da migração do banco de dados. Para obter mais informações, consulte

[Depois de criar um relatório de avaliação de migração de banco de dados, é possível salvar uma cópia local do relatório de avaliação de migração de banco de dados como um arquivo PDF ou um arquivo de valores separados por vírgula \(CSV\).](#)

[Para salvar o relatório de avaliação de migração de banco de dados como um arquivo PDF](#)

-
1. No menu superior, selecione Exibir e, em seguida, selecione Visualização do relatório de avaliação.
 2. Escolha a guia Resumo.
 3. Selecione Salvar em PDF no canto superior direito.
-

[Para salvar o relatório de avaliação de migração de banco de dados como arquivos CSV](#)

-
1. No menu superior, selecione Exibir e, em seguida, selecione Visualização do relatório de avaliação.
 2. Escolha a guia Resumo.
 3. Selecione Salvar em CSV no canto superior direito.
-

[O arquivo PDF contém as informações de item de ação e de resumo, conforme mostrado no exemplo a seguir.](#)

Database objects with conversion actions for Amazon RDS for PostgreSQL

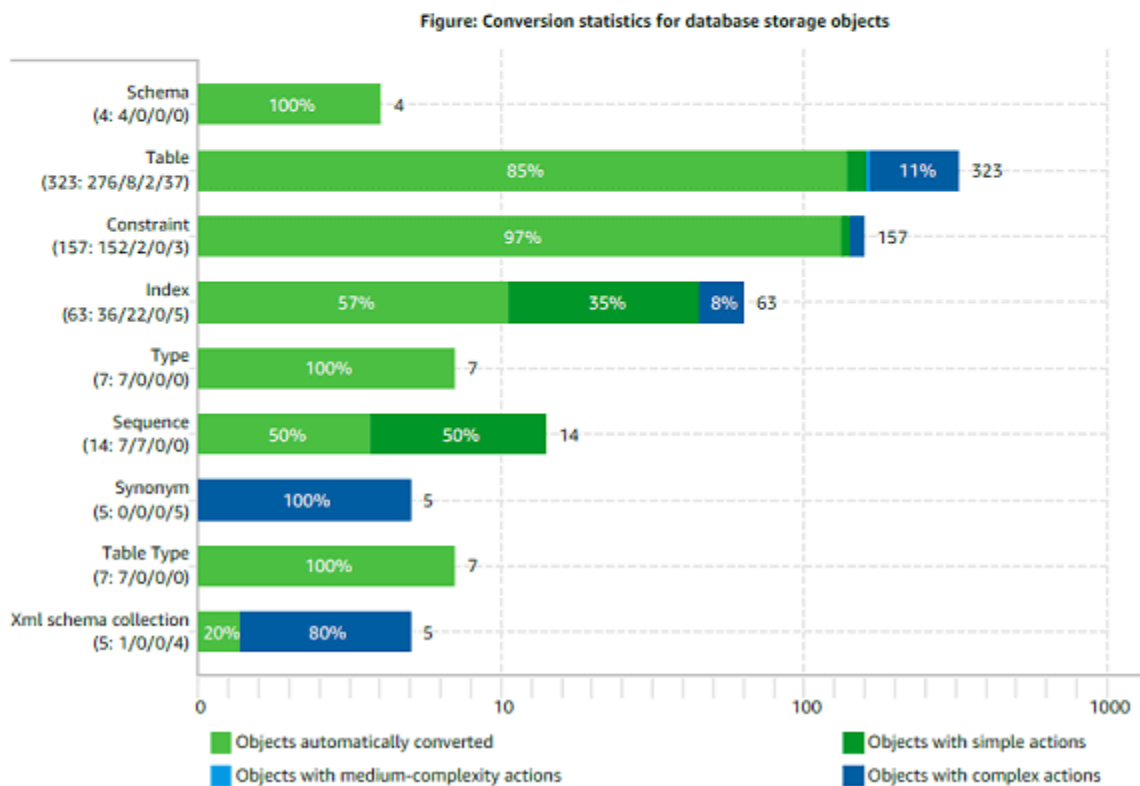
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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



Ao escolher a opção Salvar em CSV, a AWS SCT cria três arquivos CSV.

O primeiro arquivo CSV contém as seguintes informações sobre os itens de ação:

- Categoria
- Ocorrência – O nome do arquivo, o número da linha e a posição do item
- Número do item de ação
- Subject
- Grupo

- Descrição

- Referências da documentação

- Ação recomendada

- Complexidade estimada

O segundo arquivo CSV inclui o sufixo `Action_Items_Summary` em seu nome e contém as informações sobre o número de ocorrências de todos os itens de ação.

No exemplo a seguir, os valores na coluna `Esforço` da curva de aprendizado indicam a quantidade de esforço necessária para criar uma abordagem para converter cada item de ação. Os valores na coluna `Esforço para converter uma ocorrência do item de ação` indicam o esforço necessário para converter cada item de ação, seguindo a abordagem projetada. Os valores usados para indicar o nível dos esforços necessários são baseados em uma escala ponderada, que varia de baixo (mínimo) a 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

O terceiro arquivo CSV inclui o `Summary` em seu nome e contém o seguinte resumo:

- Categoria

- Número de objetos

- Objetos convertidos automaticamente

- Objetos com ações simples

- Objetos com ações de complexidade média

- Objetos com ações complexas

- Total de linhas de código

Como criar um relatório de avaliação multisservidor para migração de banco de dados

Para determinar a melhor direção de destino para seu ambiente geral, crie um relatório de avaliação multisservidor.

Um relatório de avaliação multisservidor avalia vários servidores com base na entrada que você fornece para cada definição de esquema que você deseja avaliar. Sua definição de esquema contém parâmetros de conexão do servidor de banco de dados e o nome completo de cada esquema. Depois de avaliar cada esquema, a AWS SCT produz um relatório de avaliação resumido e agregado para a migração do banco de dados em seus vários servidores. Esse relatório mostra a complexidade estimada para cada possível destino de migração.

Você pode usar a AWS SCT para criar um relatório de avaliação multisservidor para os seguintes bancos de dados de origem e destino.

Fonte do banco de dados	Bancos de dados de destino:
Amazon Redshift	Amazon Redshift
Banco de dados do 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 Edição compatível com MySQL (Aurora MySQL), Amazon Aurora Edição compatível com PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL
IBM Db2 LUW	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, Babelfish para Aurora PostgreSQL

Fonte do banco de dados	Bancos de dados de destino:
	L, 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

Como realizar uma avaliação multisservidor

Use o procedimento a seguir para realizar uma avaliação multisservidor com AWS SCT. Não é necessário criar um projeto na AWS SCT para realizar uma avaliação multisservidor. Antes de começar, certifique-se de ter preparado um arquivo de valores separados por vírgula (CSV) com parâmetros de conexão do banco de dados. Além disso, verifique se você instalou todos os drivers de banco de dados necessários e defina a localização dos drivers nas configurações da AWS SCT. Para obter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

Para realizar uma avaliação multisservidor e criar um relatório resumido agregado

1. Na AWS SCT, selecione Arquivo, Nova avaliação multisservidor. A caixa de diálogo Nova avaliação multisservidor é aberta.

New multiserver assessment ✕

Enter the project name, location to store reports and project files, and location of your connections file. ?

Project name

Location

Connections file

[Download a connections file example](#)

Create AWS SCT projects for each source database

Add mapping rules to these projects and save conversion statistics for offline use

2. Selecione Baixar um exemplo de arquivo de conexões para baixar um modelo vazio de um arquivo CSV com parâmetros de conexão do banco de dados.
3. Insira valores para Nome do projeto, Localização (para armazenar relatórios) e Arquivo de conexões (um arquivo CSV).
4. Selecione Criar projetos AWS SCT para cada banco de dados de origem para criar automaticamente projetos de migração após gerar o relatório de avaliação.
5. Com a opção Criar projetos AWS SCT para cada banco de dados de origem ativada, você pode escolher Adicionar regras de mapeamento a esses projetos e salvar as estatísticas de conversão para uso off-line. Nesse caso, a AWS SCT adicionará regras de mapeamento a cada projeto e salvará os metadados do banco de dados de origem no projeto. Para obter mais informações, consulte [Executando a AWS SCT em um modo off-line](#).

6. Escolha Run.

Uma barra de progresso aparece indicando o ritmo da avaliação do banco de dados. O número de mecanismos de destino pode afetar o runtime da avaliação.

7. Selecione Sim se a seguinte mensagem for exibida: A análise completa de todos os servidores de banco de dados pode levar algum tempo. Você quer continuar?

Quando o relatório de avaliação multisservidor é concluído, uma tela é exibida indicando isso.

8. Selecione Abrir relatório para ver o relatório de avaliação resumida agregado.

Por padrão, a AWS SCT gera um relatório agregado para todos os bancos de dados de origem e um relatório de avaliação detalhado para cada nome de esquema em um banco de dados de origem. Para obter mais informações, consulte [Como localizar e visualizar relatórios](#).

Com a opção Criar projetos da AWS SCT para cada banco de dados de origem ativada, a AWS SCT cria um projeto vazio para cada banco de dados de origem. A AWS SCT também cria relatórios de avaliação conforme descrito anteriormente. Depois de analisar esses relatórios de avaliação e escolher o destino da migração para cada banco de dados de origem, adicione bancos de dados de destino a esses projetos vazios.

Com a opção Adicionar regras de mapeamento a esses projetos e salvar as estatísticas de conversão para uso off-line ativada, a AWS SCT cria um projeto para cada banco de dados de origem. Esses projetos incluem as seguintes informações:

- Seu banco de dados de origem e uma plataforma de banco de dados de destino virtual. Para obter mais informações, consulte [Usar alvos virtuais](#).
- Uma regra de mapeamento para esse par de origem e destino. Para obter mais informações, consulte [Criação de regras de mapeamento](#).
- Um relatório de avaliação da migração do banco de dados para esse par de origem e destino.
- Metadados do esquema de origem, que permitem que você use esse projeto da AWS SCT em um modo off-line. Para obter mais informações, consulte [Executando a AWS SCT em um modo off-line](#).

Como preparar um arquivo CSV de entrada

Para fornecer parâmetros de conexão como entrada para o relatório de avaliação multisservidor, use um arquivo CSV conforme exibido no exemplo a seguir.

```
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
```

O exemplo anterior usa ponto e vírgula para separar os dois nomes de esquema do banco de dados Sales. Ele também usa um ponto e vírgula para separar as duas plataformas de migração do banco de dados de destino para o banco de dados Sales.

Além disso, o exemplo anterior usa AWS Secrets Manager para se conectar ao banco de dados Customers e a Autenticação do Windows para se conectar ao banco de dados HR.


Você pode criar um novo arquivo CSV ou baixar um modelo para um arquivo CSV da AWS SCT e preencher as informações necessárias. Certifique-se de que a primeira linha do seu arquivo CSV inclua os mesmos nomes de coluna mostrados no exemplo anterior.

Para baixar um modelo do arquivo CSV de entrada

1. Inicie AWS SCT.
2. Selecione Arquivo, depois selecione Nova avaliação multisservidor.
3. Selecione Baixar um exemplo de arquivo de conexões.

Certifique-se de que seu arquivo CSV inclua os seguintes valores, fornecidos pelo modelo:

- Nome – O rótulo de texto que ajuda a identificar seu banco de dados. A AWS SCT exibe esse rótulo de texto no relatório de avaliação.
- Descrição – Um valor opcional, no qual você pode fornecer informações adicionais sobre o banco de dados.
- Chave do Secrets Manager – O nome do segredo que armazena suas credenciais de banco de dados na AWS Secrets Manager. Para usar o Secrets Manager, certifique-se de armazenar perfis da AWS na AWS SCT. Para obter mais informações, consulte [Usar o AWS Secrets Manager](#).

 Important

A AWS SCT ignorará o parâmetro Chave do Secrets Manager se você incluir os parâmetros Servidor IP, Porta, Login e Senha no arquivo de entrada.

- Servidor IP – Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
- Porta – A porta usada para se conectar ao servidor de banco de dados de origem.
- Nome do serviço – Se você usar um nome de serviço para se conectar ao seu banco de dados Oracle, o nome do serviço Oracle ao qual se conectar.

- Nome do banco de dados – O nome do banco de dados. Para bancos de dados Oracle, use o Oracle System ID (SID).
- Caminho do BigQuery – O caminho para o arquivo de chave da conta de serviço do seu banco de dados de origem do BigQuery. Para obter mais informações sobre a criação desse arquivo, consulte a [Privilégios do BigQuery como origem](#).
- Mecanismo de origem – O tipo do seu banco de dados de origem. Use um dos seguintes valores:
 - AZURE_MSSQL para um banco de dados SQL do Azure.
 - AZURE_SYNAPSE para um banco de dados do Azure SynapseAnalytics.
 - GOOGLE_BIGQUERY para um banco de dados do BigQuery.
 - DB2ZOS para um banco de dados IBM Db2 para z/OS.
 - DB2LUW para um banco de dados IBM Db2 LUW.
 - GREENPLUM para um banco de dados Greenplum.
 - MSSQL para um banco de dados Microsoft SQL Server.
 - MYSQL para um banco de dados MySQL.
 - NETEZZA para um banco de dados Netezza.
 - ORACLE para um banco de dados Oracle.
 - POSTGRESQL para um banco de dados PostgreSQL.
 - REDSHIFT para um banco de dados do Amazon Redshift.
 - SNOWFLAKE para um banco de dados Snowflake.
 - SYBASE_ASE para um banco de dados SAP ASE.
 - TERADATA para um banco de dados Teradata.
 - VERTICA para um banco de dados Vertica.
- Nomes do esquema – Os nomes dos esquemas do banco de dados a serem incluídos no relatório de avaliação.

Para o Banco de Dados SQL do Azure, o Azure Synapse Analytics, o BigQuery, o Netezza, o SAP ASE, o Snowflake e o SQL Server, use o seguinte formato do nome do esquema:

db_name.schema_name

Substitua *db_name* pelo nome do banco de dados de origem.

Substitua *schema_name* pelo nome do esquema de origem.

Inclua nomes de bancos de dados ou esquemas que incluam um ponto entre aspas duplas, conforme mostrado a seguir: "database.name"."schema.name".

Separe vários nomes de esquemas usando ponto e vírgula, conforme mostrado a seguir: Schema1;Schema2.

Os nomes do banco de dados e do esquema diferenciam maiúsculas de minúsculas.

Utilize a porcentagem (%) como curinga para substituir qualquer número de quaisquer símbolos no nome do banco de dados ou esquema. O exemplo anterior usa a porcentagem (%) como curinga para incluir todos os esquemas do banco de dados employees no relatório de avaliação.

- Use a Autenticação do Windows – Se você usa a Autenticação do Windows para se conectar ao banco de dados do Microsoft SQL Server, digite verdadeiro. Para obter mais informações, consulte [Usando a autenticação do Windows ao usar o Microsoft SQL Server como origem](#).
- Login – O nome de usuário para se conectar ao servidor do banco de dados de origem.
- Senha – A senha para se conectar ao servidor de banco de dados de origem.
- Use SSL – Se você usar Secure Sockets Layer (SSL) para se conectar ao seu banco de dados de origem, digite verdadeiro.
- Armazenamento confiável — O armazenamento confiável a ser usado para sua conexão SSL.
- Armazenamento de chaves — O armazenamento de chaves a ser usado para sua conexão SSL.
- Autenticação SSL – Se você usar a autenticação SSL por certificado, digite verdadeiro.
- Mecanismos de destino – As plataformas de banco de dados de destino. Use os valores a seguir para especificar um ou mais destinos no relatório de avaliação:
 - AURORA_MYSQL para um banco de dados compatível com o Aurora MySQL.
 - AURORA_POSTGRESQL para um banco de dados compatível com o Aurora PostgreSQL.
 - BABELFISH para um banco de dados Babelfish para Aurora PostgreSQL.
 - MARIA_DB para um banco de dados MariaDB.
 - MSSQL para um banco de dados Microsoft SQL Server.
 - MYSQL para um banco de dados MySQL.
 - ORACLE para um banco de dados Oracle.
 - POSTGRESQL para um banco de dados PostgreSQL.
 - REDSHIFT para um banco de dados do Amazon Redshift.

Separe vários destinos usando ponto e vírgula como este: MYSQL ; MARIA_DB. O número de destinos afeta o tempo necessário para executar a avaliação.

Como localizar e visualizar relatórios

A avaliação multisservidor gera dois tipos de relatórios:

- Um relatório agregado de todos os bancos de dados de origem.
- Um relatório de avaliação detalhado dos bancos de dados de destino para cada nome de esquema em um banco de dados de origem.

Os relatórios são armazenados no diretório que você escolheu para Localização na caixa de diálogo Nova avaliação multisservidor.

Para acessar os relatórios detalhados, você pode navegar pelos subdiretórios, que são organizados por banco de dados de origem, nome do esquema e mecanismo de banco de dados de destino.

Os relatórios agregados mostram informações em quatro colunas sobre a complexidade da conversão de um banco de dados de destino. As colunas incluem informações sobre conversão de objetos de código, objetos de armazenamento, elementos de sintaxe e complexidade de conversão.

O exemplo a seguir mostra informações para conversão de dois esquemas de banco de dados Oracle para 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

As mesmas quatro colunas são anexadas aos relatórios para cada mecanismo de banco de dados de destino adicional especificado.

Para obter detalhes sobre como ler essas informações, consulte a seguir.

Saída para um relatório de avaliação agregado

O relatório agregado de avaliação da migração de banco de dados multisservidor em AWS Schema Conversion Tool é um arquivo CSV com as seguintes colunas:

- 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 coletar informações, a AWS SCT executa relatórios de avaliação completos e, em seguida, agrega relatórios por esquemas.

No relatório, os três campos a seguir mostram a porcentagem de conversão automática possível com base na avaliação:

% de conversão de objeto de código

A porcentagem de objetos de código no esquema que a AWS SCT pode converter automaticamente ou com alterações mínimas. Objetos de código incluem procedimentos, perfis, visualizações e similares.

% de conversão de objeto de armazenamento

A porcentagem de objetos de armazenamento que a SCT pode converter automaticamente ou com alterações mínimas. Os objetos de armazenamento incluem tabelas, índices, restrições e similares.

% de conversão de elementos de sintaxe

A porcentagem de elementos de sintaxe que a SCT pode converter automaticamente. Os elementos de sintaxe incluem as cláusulas SELECT, FROM, DELETE e JOIN e similares.

O cálculo da complexidade de conversão é baseado na noção de itens de ação. Um item de ação reflete um tipo de problema encontrado no código fonte que você precisa corrigir manualmente durante a migração para um destino específico. Um item de ação pode ter várias ocorrências.

Uma escala ponderada identifica o nível de complexidade para realizar uma migração. O número 1 representa o nível mais baixo de complexidade e o número 10 representa o nível mais alto.

Como converter esquemas de bancos de dados usando a AWS SCT

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter esquemas de bancos de dados existentes de um mecanismo de banco de dados para outro. Converter um banco de dados usando a interface do usuário da AWS SCT pode ser bastante simples, mas há vários aspectos a considerar antes de fazer a conversão.

Por exemplo, você pode usar a AWS SCT para fazer o seguinte:

- Você pode usar a AWS SCT para copiar um esquema de banco de dados on-premises existente para uma instância de banco de dados do Amazon RDS que esteja executando o mesmo mecanismo. Você pode usar esse recurso para analisar possíveis economias de custo ao mudar para a nuvem e ao alterar o tipo de licença.
- Em alguns casos, os recursos do banco de dados não podem ser convertidos em recursos do Amazon RDS equivalentes. Se você hospedar e gerenciar um banco de dados na plataforma Amazon Elastic Compute Cloud (Amazon EC2), será possível emular esses recursos, substituindo os serviços da AWS por eles.
- A AWS SCT automatiza grande parte do processo de conversão do esquema de banco de dados do processamento de transações online (OLTP) para uma instância de banco de dados MySQL do Amazon Relational Database Service (Amazon RDS), para um cluster de banco de dados do Amazon Aurora ou para uma instância de banco de dados PostgreSQL. Os mecanismos de banco de dados de origem e de destino contêm diversos recursos e capacidades, e a AWS SCT tentará criar um esquema equivalente na instância de banco de dados do Amazon RDS sempre que possível. Se a conversão direta não for possível, a AWS SCT fornecerá uma lista de ações possíveis para você executar.

Tópicos

- [Como criar regras de migração na AWS SCT](#)
- [Como converter o esquema com a AWS SCT](#)
- [Como lidar com conversões manuais na AWS SCT](#)
- [Como atualizar e recarregar o esquema convertido na AWS SCT](#)
- [Como salvar e aplicar seu esquema convertido na AWS SCT](#)
- [Como comparar esquemas de bancos de dados](#)

- [Como localizar objetos transformados relacionados](#)

A AWS SCT suporta as seguintes conversões de processamento de transações online (OLTP).

Fonte do banco de dados	Bancos de dados de destino:
IBM Db2 para z/OS (versão 12)	Amazon Aurora Edição Compatível com MySQL, Amazon Aurora Edição Compatível com PostgreSQL, MySQL, PostgreSQL
IBM Db2 LUW (versões 9.1, 9.5, 9.7, 10.5, 11.1 e 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Banco de dados do Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (versão 2008 R2 e superior)	Aurora MySQL, Aurora PostgreSQL, Babelfish para Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (versão 5.5 e superior)	Aurora PostgreSQL, MySQL, PostgreSQL Você pode migrar o esquema e os dados do MySQL para um cluster de banco de dados do Aurora MySQL sem usar a AWS SCT. Para obter mais informações, consulte Migração de dados para um cluster de banco de dados do Amazon Aurora .
Oracle (versão 10.2 e superior)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (versão 9.1 e superior)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12.5, 15.0, 15.5, 15.7 e 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Para obter mais informações sobre como converter um esquema de data warehouse, consulte [Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT](#).

Para converter o esquema de banco de dados para o Amazon RDS, siga as etapas de alto nível a seguir:

- [Como criar regras de migração na AWS SCT](#): antes de converter seu esquema com a AWS SCT, você pode configurar regras que alteram o tipo de dados de colunas, movem objetos de um esquema para outro e alteram os nomes de objetos.
- [Como converter o esquema com a AWS SCT](#): a AWS SCT cria uma versão local do esquema convertido para você analisar, mas não a aplica à instância de banco de dados de destino até que você esteja pronto.
- [Como criar relatórios de avaliação de migração com a AWS SCT](#): a AWS SCT cria um relatório de avaliação de migração de banco de dados que detalha os elementos do esquema que não podem ser convertidos automaticamente. É possível usar esse relatório para identificar onde você precisa criar um esquema na sua instância de banco de dados do Amazon RDS que seja compatível com o seu banco de dados de origem.
- [Como lidar com conversões manuais na AWS SCT](#): se você tem elementos do esquema que não podem ser convertidos automaticamente, há duas opções: atualizar o esquema de origem e fazer a conversão novamente ou criar elementos do esquema equivalentes na instância de banco de dados de destino do Amazon RDS.
- [Como atualizar e recarregar o esquema convertido na AWS SCT](#): você pode atualizar seu projeto da AWS SCT com o esquema mais recente do banco de dados de origem.
- [Como salvar e aplicar seu esquema convertido na AWS SCT](#): quando estiver pronto, faça com que a AWS SCT aplique o esquema convertido em seu projeto local à instância de banco de dados do Amazon RDS de destino.

Como criar regras de migração na AWS SCT

Antes de converter seu esquema com a AWS SCT, você pode configurar regras de migração. As regras de migração na AWS SCT podem fazer coisas como alterar o tipo de dados das colunas, mover objetos de um esquema para outro e alterar os nomes dos objetos. Por exemplo, suponha que você tenha um conjunto de tabelas no esquema de origem chamado `test_TABLE_NAME`. É possível configurar uma regra que altera o prefixo `test_` para o prefixo `demo_` no esquema de destino.

Note

Você só pode criar regras de migração para diferentes mecanismos de banco de dados de origem e destino.

Você pode criar regras de migração que executam as seguintes tarefas:

- Adicionar, remover ou substituir um prefixo
- Adicionar, remover ou substituir um sufixo
- Alterar agrupamento de colunas
- Alterar tipo de dados
- Alterar o comprimento dos tipos de dados `char`, `varchar`, `nvarchar` e `string`
- Mover objetos
- Renomear objetos

Você pode criar regras de migração para os seguintes objetos:

- Banco de dados
- Esquema
- Tabela
- Coluna

Como criar regras de migração

Você pode criar regras de migração salvar as regras como parte do seu projeto. Com seu projeto aberto, use o procedimento a seguir para criar regras de migração.

Para criar regras de migração

1. No menu Exibir, selecione Exibição de mapeamento.
2. Em Mapeamentos de servidores, escolha um par de servidores de origem e de destino.
3. Selecione Nova regra de migração. A caixa de diálogo Regras de transformação é exibida.
4. Escolha Adicionar nova regra. Uma nova linha é adicionada à lista de regras.
5. Configure sua regra:

- a. Em Name (Nome), insira um nome para a regra.
- b. Em Para, escolha o tipo de objeto ao qual a regra se aplica.
- c. Em onde, digite um filtro para aplicar aos objetos antes de aplicar a regra de migração. A cláusula where é avaliada usando uma cláusula like. Você pode inserir um nome exato para selecionar um objeto, ou pode inserir um padrão para selecionar vários objetos.

Os campos disponíveis para a cláusula where são diferentes, dependendo do tipo do objeto. Por exemplo, se o tipo de objeto for esquema, haverá apenas um campo disponível para o nome do esquema.

- d. Em Ações, escolha o tipo de regra de migração que você deseja criar.
- e. Dependendo do tipo de regra, insira um ou dois valores adicionais. Por exemplo, para renomear um objeto, insira o novo nome do objeto. Para substituir um prefixo, insira o prefixo antigo e o novo prefixo.

Para os tipos de dados char, varchar, nvarchar e string, você pode alterar o comprimento do tipo de dados usando o operador de multiplicação. Por exemplo, o valor `4` transforma o tipo de dados `varchar(10)` em `varchar(40)`.

6. Após configurar sua regra de migração, selecione Salvar para salvar sua regra. Você também pode escolher Cancelar para cancelar as alterações.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS [doesn't support](#) more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Ao finalizar a adição, edição e exclusão das regras, escolha Salvar tudo para salvar todas as alterações.
8. Selecione Fechar para fechar a caixa de diálogo Regras de transformação.

Você pode usar o ícone de alternância para desativar uma regra de migração sem excluí-la. Você pode usar o ícone de cópia para duplicar uma regra de migração existente. Use o ícone de lápis para editar uma regra de migração existente. Você pode usar o ícone de exclusão para excluir uma regra de migração existente. Para salvar as alterações feitas em suas regras de migração, selecione Salvar tudo.

Como exportar regras de migração

Se você usar o AWS DMS para migrar os dados do seu banco de dados de origem para o banco de dados de destino, será possível fornecer informações sobre as regras de migração para o AWS DMS. Para obter mais informações sobre tarefas, consulte [Como trabalhar com tarefas de replicação do AWS Database Migration Service](#).

Para exportar regras de migração

1. Na AWS Schema Conversion Tool, selecione Exibição de mapeamento no menu Exibir.
2. Em Regras de migração, escolha uma regra de migração e, em seguida, selecione Modificar regra de migração.
3. Selecione Exportar script para o AWS DMS.
4. Navegue até o local onde você deseja salvar seu script e escolha Salvar. Suas regras de migração são salvas como um script JSON que pode ser consumido pelo AWS DMS.

Como converter o esquema com a AWS SCT

Depois que você tiver conectado o projeto ao banco de dados de origem e de destino da instância de banco de dados do Amazon RDS, seu projeto da AWS Schema Conversion Tool exibirá o esquema do banco de dados de origem no painel esquerdo. O esquema é apresentado em um formato de visualização em árvore, e cada nó da árvore é de carregamento preguiçoso. Quando você escolhe um nó na visualização em árvore, a AWS SCT solicita as informações do esquema do seu banco de dados de origem nesse momento.

Você pode escolher itens do esquema de seu banco de dados de origem e converter o esquema para o esquema equivalente do mecanismo de banco de dados de sua instância de banco de dados

de destino. Você pode escolher qualquer item de esquema de seu banco de dados de origem para converter. Se o item de esquema que você escolher depende de um item pai, a AWS SCT também gerará o esquema para o item pai. Por exemplo, suponha que você escolha uma tabela para converter. Em caso afirmativo, a AWS SCT gera o esquema para a tabela e o banco de dados em que a tabela está.

Converter o esquema

Para converter um esquema do seu banco de dados de origem, marque a caixa de seleção do nome do esquema a ser convertido. Em seguida, selecione esse esquema no painel esquerdo do seu projeto. A AWS SCT destaca o nome do esquema em azul. Abra o menu de contexto (clique com o botão direito do mouse) do esquema e selecione Converter esquema, conforme mostrado a seguir.

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
 Create report
Convert schema
 Register agent
 Compare schema
 Load schema
 Hide schema
 Refresh from database
 Collect statistics
 Upload statistics
 Create DMS task
 Create Local & DMS task
 Create Local task
 Add virtual partitioning
 Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

Depois de converter o esquema do seu banco de dados de origem, você pode escolher os itens de esquema no painel esquerdo do seu projeto e visualizar o esquema convertido nos painéis centrais do seu projeto. O painel central inferior exibe as propriedades e o comando SQL para criar o esquema convertido, como mostrado a seguir.

The screenshot shows the AWS Schema Conversion Tool interface. On the left, a tree view displays the source database structure: 'SQL Server - ec2-52-17-21-76.eu-west-1.cc' contains 'Databases [12]', including 'LARGE_DB_SS'. Under 'LARGE_DB_SS', there are 'Schemas [2]', including 'dbo', which contains 'Tables [4]', including 'Account'. The 'Account' table is selected.

The main pane shows the SQL for the source table:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the SQL, it indicates 'Cursor position: 0 Modified: true'.

The bottom pane shows the target Amazon RDS schema for the 'Account' table:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Depois de converter seu esquema, você poderá salvar seu projeto. As informações do esquema de seu banco de dados de origem são salvas com o seu projeto. Essa funcionalidade significa que você pode trabalhar off-line sem estar conectado ao seu banco de dados de origem. A AWS SCT vai se conectar ao seu banco de dados de origem para atualizar o esquema em seu projeto, se você selecionar Atualizar do banco de dados. Para obter mais informações, consulte [Como atualizar e recarregar o esquema convertido na AWS SCT](#).

Você pode criar um relatório de avaliação de migração do banco de dados dos itens que não podem ser convertidos automaticamente. O relatório de avaliação é útil para identificar e resolver itens de esquema que não podem ser convertidos automaticamente. Para obter mais informações, consulte [Como criar relatórios de avaliação de migração com a AWS SCT](#).

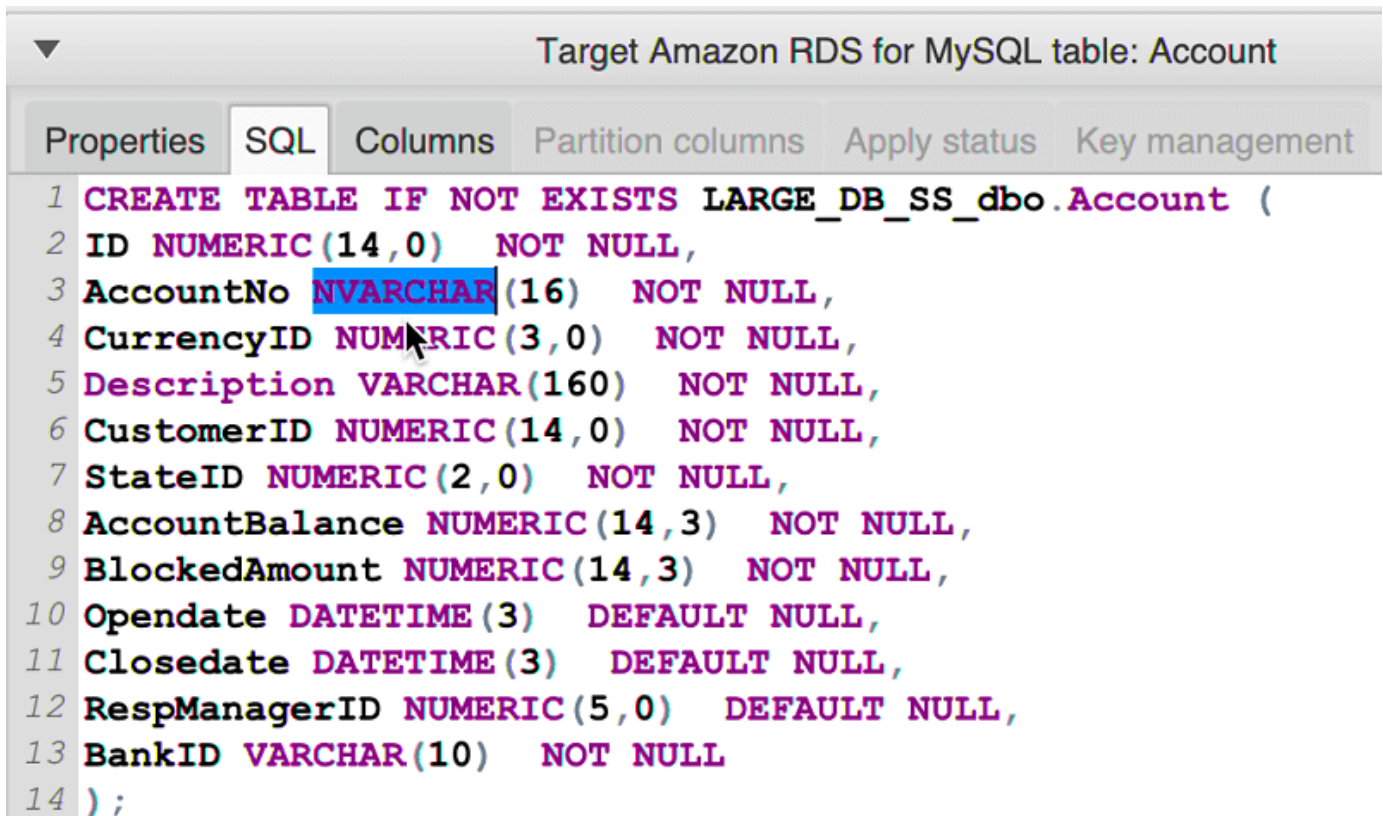
Quando a AWS SCT gera um esquema convertido, ela não o aplica imediatamente à instância de banco de dados de destino. Em vez disso, o esquema convertido é armazenado localmente até que você esteja pronto para aplicá-lo à instância de banco de dados de destino. Para obter mais informações, consulte [Como aplicar seu esquema convertido](#).

Como editar o esquema convertido

Você pode editar o esquema convertido e salvar as alterações como parte do seu projeto.

Para editar o esquema convertido

1. No painel esquerdo que exibe o esquema de seu banco de dados de origem, escolha o item do esquema para o qual você deseja editar o esquema convertido.
2. No painel central inferior que exibe o esquema convertido referente ao item selecionado, escolha a guia SQL.
3. No texto exibido referente à guia SQL, altere o esquema conforme necessário. O esquema é salvo automaticamente com o seu projeto, à medida que você o atualiza.



The screenshot shows the 'Target Amazon RDS for MySQL table: Account' interface. The 'SQL' tab is selected, displaying the following SQL code:

```
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (  
2 ID NUMERIC(14,0) NOT NULL,  
3 AccountNo NVARCHAR(16) NOT NULL,  
4 CurrencyID NUMERIC(3,0) NOT NULL,  
5 Description VARCHAR(160) NOT NULL,  
6 CustomerID NUMERIC(14,0) NOT NULL,  
7 StateID NUMERIC(2,0) NOT NULL,  
8 AccountBalance NUMERIC(14,3) NOT NULL,  
9 BlockedAmount NUMERIC(14,3) NOT NULL,  
10 Opendate DATETIME(3) DEFAULT NULL,  
11 Closedate DATETIME(3) DEFAULT NULL,  
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,  
13 BankID VARCHAR(10) NOT NULL  
14 );
```

As alterações que você fizer no esquema convertido serão armazenadas com seu projeto à medida que você o atualiza. Se você converteu recentemente um item do esquema de seu banco de dados de origem, e fez atualizações no esquema convertido anteriormente para esse item, essas atualizações existentes serão substituídas pelo item do esquema recém-convertido com base em seu banco de dados de origem.

Como limpar um esquema convertido

Até que você aplique o esquema à sua instância de banco de dados de destino, a AWS SCT armazenará apenas o esquema convertido localmente em seu projeto. Você pode limpar o esquema planejado do seu projeto, escolhendo o nó de visualização em árvore para a sua instância de banco de dados e, em seguida, escolhendo Atualizar do banco de dados. Como nenhum esquema foi gravado em sua instância de banco de dados de destino, a atualização do banco de dados removerá os elementos do esquema planejado em seu projeto da AWS SCT para coincidir com o que existe em sua instância de banco de dados de origem.

Como lidar com conversões manuais na AWS SCT

O relatório de avaliação inclui uma lista de itens que não podem ser convertidos automaticamente para o mecanismo de banco de dados de sua instância de banco de dados de destino do Amazon RDS. Para cada item que não pode ser convertido, existe um item de ação na guia Itens de ação.

Você pode responder aos itens de ação do relatório de avaliação das seguintes formas:

- Modifique o esquema do banco de dados de origem.
- Modifique o esquema do banco de dados de destino.

Como modificar seu esquema de origem

Para alguns itens, pode ser mais fácil modificar o esquema de banco de dados no seu banco de dados de origem para um esquema que pode ser convertido automaticamente. Primeiro, verifique se as novas alterações são compatíveis com sua arquitetura de aplicativo e, em seguida, atualizar o esquema no seu banco de dados de origem. Finalmente, atualize seu projeto com as informações do esquema atualizado. É possível converter o esquema atualizado, e gerar um novo relatório de avaliação de migração de banco de dados. Os itens de ação não são mais exibidos para os itens que são alterados no esquema de origem.

A vantagem desse processo é que o esquema atualizado está sempre disponível quando você atualiza a partir de seu banco de dados de origem.

Como modificar seu esquema de destino

Para alguns itens, pode ser mais fácil aplicar o esquema convertido ao seu banco de dados de destino e, em seguida, adicionar itens de esquema equivalente manualmente ao seu banco de dados

de destino para os itens que não puderam ser convertidos automaticamente. Você pode gravar tudo do esquema que pode ser convertido automaticamente para sua instância de banco de dados destino, aplicando o esquema. Para obter mais informações, consulte [Como salvar e aplicar seu esquema convertido na AWS SCT](#).

O esquema que é gravado na instância de banco de dados de destino não contém os itens que não podem ser convertidos automaticamente. Depois de aplicar o esquema à instância de banco de dados de destino, você pode criar manualmente o esquema na instância de banco de dados de destino que é equivalente ao do banco de dados de origem. Os itens de ação no relatório de avaliação de migração de banco de dados contém sugestões de como criar o esquema equivalente.

Warning

Se você criar manualmente o esquema em sua instância de banco de dados de destino, salve uma cópia de qualquer trabalho manual. Se você aplicar o esquema convertido do seu projeto à instância de banco de dados de destino novamente, ela substituirá o trabalho manual que você tenha feito.

Em alguns casos, não é possível criar um esquema equivalente na instância de banco de dados de destino. Convém rearquitar uma parte do aplicativo e do banco de dados para usar a funcionalidade que está disponível no mecanismo de banco de dados para a instância de banco de dados de destino. Em outros casos, você pode simplesmente ignorar o esquema que não pode ser convertido automaticamente.

Como atualizar e recarregar o esquema convertido na AWS SCT

Você pode atualizar o esquema de origem e o esquema de destino em seu projeto da AWS Schema Conversion Tool.

- **Origem:** Se você atualizar o esquema de seu banco de dados de origem, a AWS SCT substituirá o esquema em seu projeto pelo esquema mais recente do seu banco de dados de origem. Usando essa funcionalidade, você pode atualizar seu projeto, se as alterações tiverem sido feitas no esquema de seu banco de dados de origem.
- **Destino:** se você atualizar o esquema de sua instância de banco de dados do Amazon RDS de destino, a AWS SCT substituirá o esquema em seu projeto pelo esquema mais recente de sua instância de banco de dados de destino. Se você não aplicou qualquer esquema à sua instância de banco de dados de destino, a AWS SCT apagará o esquema convertido de seu projeto. Em

seguida, você pode converter o esquema do seu banco de dados de origem para uma instância de banco de dados de destino limpo.

Atualize o esquema em seu projeto da AWS SCT, selecionando Atualizar do banco de dados.

Note

Ao atualizar seu esquema, a AWS SCT carrega metadados somente quando necessário. Para carregar completamente todo o esquema do banco de dados, abra o menu de contexto (clique com o botão direito do mouse) do esquema e selecione Carregar esquema. Por exemplo, você pode usar essa opção para carregar metadados para seu banco de dados de uma só vez e depois trabalhar offline.

Como salvar e aplicar seu esquema convertido na AWS SCT

Quando a AWS Schema Conversion Tool gerar o esquema convertido (conforme mostrado em [Como converter o esquema com a AWS SCT](#)), ela não aplicará imediatamente o esquema convertido à instância de banco de dados. Em vez disso, o esquema convertido é armazenado localmente em seu projeto até que você esteja pronto para aplicá-lo à instância de banco de dados de destino. Usando essa funcionalidade, você pode trabalhar com itens do esquema que não podem ser convertidos automaticamente para o mecanismo de banco de dados de destino. Para obter mais informações sobre itens que não podem ser convertidos automaticamente, consulte [Como criar relatórios de avaliação de migração com a AWS SCT](#).

Opcionalmente, você pode fazer a ferramenta salvar o esquema convertido em um arquivo como um script SQL antes de aplicar o esquema à instância de banco de dados de destino. Você também pode fazer a ferramenta aplicar o esquema convertido diretamente à sua instância de banco de dados de destino.

Como salvar seu esquema convertido em um arquivo

Você pode salvar seu esquema convertido como scripts SQL em um arquivo de texto. Ao usar essa abordagem, você pode modificar os scripts SQL gerados da AWS SCT para lidar com itens que a ferramenta não pode converter automaticamente. Em seguida, você pode executar seus scripts atualizados em sua instância de banco de dados de destino para aplicar o esquema convertido ao seu banco de dados de destino.

Como salvar o esquema convertido como scripts SQL

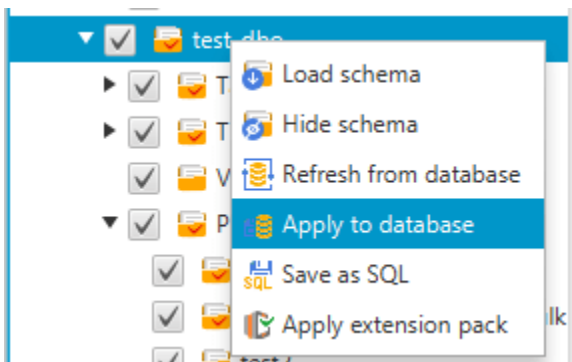
1. Selecione seu esquema e abra o menu de contexto (clique com o botão direito do mouse).
2. Selecione Salvar como SQL.
3. Insira o nome do arquivo e selecione Salvar.
4. Salve seu esquema convertido usando uma das seguintes opções:
 - Arquivo único
 - Arquivo único por estágio
 - Arquivo único por instrução

Para escolher o formato do script SQL

1. No menu Configurações, selecione Configurações do projeto.
2. Selecione Salvar scripts.
3. Em Fornecedor, selecione a plataforma de banco de dados.
4. Em Salvar scripts de SQL em, selecione como você deseja salvar o script do esquema de banco de dados.
5. Escolha OK para salvar as configurações.

Como aplicar seu esquema convertido

Quando você estiver pronto para aplicar o esquema convertido à instância de banco de dados de destino do Amazon RDS, escolha o elemento de esquema no painel direito do projeto. Abra o menu de contexto (clique com o botão direito do mouse) do elemento do esquema e escolha Aplicar ao banco de dados, como mostrado a seguir.



O esquema do pacote de extensão

Na primeira vez em que você aplicar o esquema convertido à sua instância de banco de dados de destino, a AWS SCT adicionará um esquema adicional à sua instância de banco de dados de destino. Esse esquema implementa as funções de sistema do banco de dados de origem necessárias para gravar o esquema convertido na instância de banco de dados de destino. O esquema é chamado de esquema de pacote de extensões.

Não modifique o esquema do pacote de extensões, ou você poderá obter resultados inesperados no esquema convertido que é gravado na instância de banco de dados de destino. Quando seu esquema estiver totalmente migrado para sua instância de banco de dados de destino, e você não precisar mais da AWS SCT, será possível excluir o esquema do pacote de extensões.

O esquema do pacote de extensões é nomeado de acordo com seu banco de dados de origem da seguinte forma:

- 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 obter mais informações, consulte [Usando as AWS Lambda funções do pacote AWS SCT de extensão](#).

Como comparar esquemas de bancos de dados

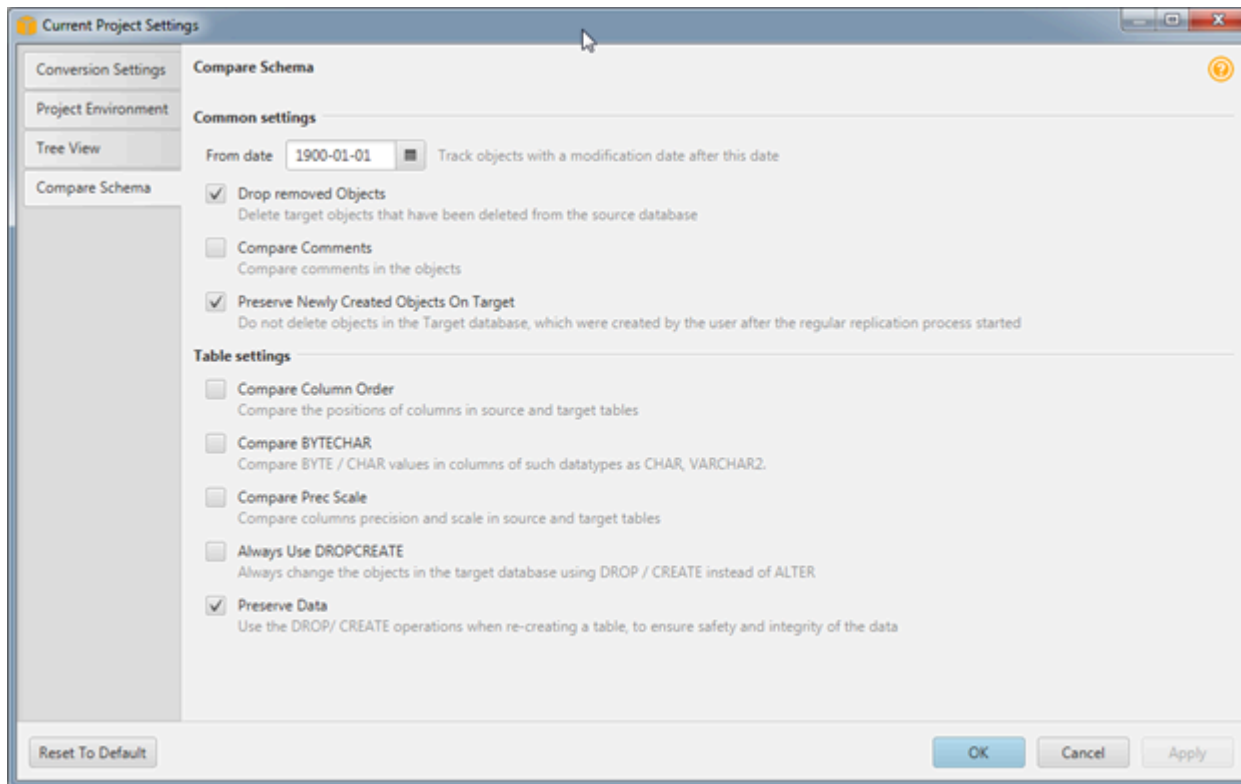
Se você fez alterações no esquema de origem ou de destino após a migração, pode comparar os dois esquemas de bancos de dados usando a AWS SCT. Você pode comparar esquemas para versões iguais ou anteriores do esquema de origem.

As seguintes comparações de esquemas são compatíveis:

- Oracle para Oracle, versões 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0, 10
- SQL Server para SQL Server, versões 2016, 2014, 2012, 2008 RD2, 2008

- PostgreSQL para PostgreSQL e Aurora Edição Compatível com PostgreSQL, versões 9.6, 9.5.9, 9.5.4
- MySQL para MySQL versões 5.6.36, 5.7.17, 5.5

Você especifica as configurações para a comparação de esquemas na guia Compare Schema (Comparar esquema) na página Project Settings (Configurações do projeto).

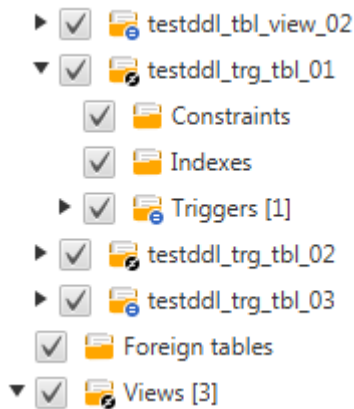


Para comparar esquemas, selecione-os e a AWS SCT indica os objetos que são diferentes entre os dois esquemas e os objetos que são iguais.

Para comparar dois esquemas

1. Abra um projeto da existente da AWS SCT ou crie um projeto e conecte-o aos endpoints de origem e de destino.
2. Selecione o esquema que você deseja comparar.
3. Abra o menu de contexto (clique com botão direito do mouse) e selecione Comparar esquema.

A AWS SCT indica os objetos que são diferentes entre os dois esquemas, adicionando um círculo preto em volta do ícone do objeto.



Você pode aplicar os resultados da comparação de esquemas a um único objeto, a uma única categoria de objetos ou a todo o esquema. Marque a caixa de seleção próxima à categoria, ao objeto ou ao esquema em que você deseja aplicar os resultados.

Como localizar objetos transformados relacionados

Após uma conversão de esquema, em alguns casos, a AWS SCT pode ter criado vários objetos para um objeto de esquema no banco de dados de origem. Por exemplo, ao executar uma conversão de Oracle para PostgreSQL, a AWS SCT pega cada gatilho do Oracle e transforma-o em um gatilho e uma função de gatilho no PostgreSQL de destino. Além disso, quando a AWS SCT converte uma função ou procedimento do pacote Oracle para PostgreSQL, ela cria uma função equivalente e uma função INIT que deve ser executada como bloqueio init antes que o procedimento ou função possam ser executados.

O procedimento a seguir permite que você visualize todos os objetos relacionados que foram criados após uma conversão de esquema.

Para visualizar os objetos relacionados que foram criados durante uma conversão de esquema

1. Após a conversão de esquema, escolha o objeto convertido na visualização em árvore de destino.
2. Selecione a guia Related Converted Objects (Objetos relacionados convertidos).
3. Visualize a lista de objetos de destino relacionados.

Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT

A AWS Schema Conversion Tool (AWS SCT) automatiza grande parte do processo de conversão do esquema de data warehouse em um esquema de banco de dados do Amazon Redshift. Como os mecanismos de banco de dados de origem e de destino podem conter diversos recursos e funcionalidades, a AWS SCT tenta criar um esquema equivalente no banco de dados de destino sempre que possível. Se a conversão direta não for possível, a AWS SCT fornecerá um relatório de avaliação com uma lista de possíveis ações a serem executadas. Usando a AWS SCT, você pode gerenciar chaves, mapear tipos de dados e objetos e criar conversões manuais.

A AWS SCT pode converter os seguintes esquemas de data warehouse para o Amazon Redshift.

- Amazon Redshift
- Azure Synapse Analytics (versão 10)
- BigQuery
- Banco de dados Greenplum (versão 4.3)
- Microsoft SQL Server (versão 2008 e superior)
- Netezza (versão 7.0.3 e superior)
- Oracle (versão 10.2 e superior)
- Snowflake (versão 3)
- Teradata (versão 13 e superior)
- Vertica (versão 7.2 e superior)

Para obter mais informações sobre como converter um esquema de banco de dados do processamento de transações on-line (OLTP), consulte [Como converter esquemas de bancos de dados usando a AWS SCT](#).

Para converter um esquema de data warehouse, siga as etapas a seguir:

1. Especifique a estratégia e as regras de otimização, e especifique as regras de migração que você deseja que a AWS SCT use. Você pode configurar regras que alteram o tipo de dados de colunas, movem objetos de um esquema para outro e alteram os nomes de objetos.

- Você pode especificar a otimização e as regras de migração em Configurações. Para obter mais informações sobre as estratégias de otimização, consulte [Como escolher estratégias e regras de otimização para uso com a AWS SCT](#). Para obter mais informações sobre as regras de migração, consulte [Como criar regras de migração na AWS SCT](#)
2. Forneça estatísticas em seu data warehouse de origem para que a AWS SCT possa otimizar como seu data warehouse é convertido. Você pode coletar estatísticas diretamente do banco de dados ou fazer upload de um arquivo de estatísticas existente. Para obter mais informações sobre o fornecimento de estatísticas de data warehouse, consulte [Como coletar ou carregar as estatísticas para a AWS SCT](#).
 3. Cria um relatório de avaliação de migração de banco de dados que detalha os elementos do esquema que não podem ser convertidos automaticamente. É possível usar esse relatório para identificar onde você precisa criar manualmente um esquema no banco de dados de destino que seja compatível com o seu banco de dados de origem. Para obter mais informações sobre o relatório de avaliação, consulte [Como criar relatórios de avaliação de migração com a AWS SCT](#).
 4. Converta o esquema. A AWS SCT cria uma versão local do esquema convertido para você analisar, mas não a aplica ao banco de dados de destino até que você esteja pronto. Para obter mais informações sobre conversão, consulte [Como converter seu esquema usando a AWS SCT](#).
 5. Depois de converter o esquema, você pode gerenciar e editar suas chaves. O gerenciamento de chaves é a essência da conversão de data warehouse. Para obter mais informações sobre gerenciamento de chaves, consulte [Como gerenciar e personalizar chaves na AWS SCT](#).
 6. Se você tem elementos do esquema que não podem ser convertidos automaticamente, há duas opções: atualizar o esquema de origem e, em seguida, fazer a conversão novamente ou criar elementos do esquema equivalentes no banco de dados de destino. Para obter mais informações sobre conversão manual de elementos de esquema, consulte [Como lidar com conversões manuais na AWS SCT](#). Para obter mais informações sobre a atualização do esquema de origem, consulte [Como atualizar e recarregar o esquema convertido na AWS SCT](#).
 7. Quando estiver pronto, você pode aplicar o esquema convertido ao banco de dados de destino. Para obter mais informações sobre como salvar e aplicar o esquema convertido, consulte [Como salvar e aplicar seu esquema convertido na AWS SCT](#).

Permissões para o Amazon Redshift como destino

As permissões necessárias para o Amazon Redshift como destino são as seguintes:

- CREATE ON DATABASE – permite criar novos esquemas no banco de dados.

- CREATE ON SCHEMA – permite criar objetos no esquema do banco de dados.
- GRANT USAGE ON LANGUAGE – permite criar novas funções e procedimentos no banco de dados.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog – fornece ao usuário informações do sistema sobre o cluster Amazon Redshift.
- GRANT SELECT ON pg_class_info – fornece ao usuário informações sobre o estilo de distribuição das tabelas.

É possível utilizar o exemplo de código a seguir para criar um usuário do banco de dados e conceder as permissões.

```
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;
```

No exemplo anterior, substitua *user_name* pelo nome do seu usuário. Substitua, então, *db_name* pelo nome do banco de dados de destino do Amazon Redshift. Em seguida, substitua *schema_name* pelo nome do seu esquema do Amazon Redshift. Repita a operação GRANT CREATE ON SCHEMA para cada esquema de destino em que você aplicará o código convertido ou migrará os dados. Por fim, substitua *your_password* por uma senha segura.

Você pode aplicar um pacote de extensão em seu banco de dados de destino do Amazon Redshift. Um pacote de extensões é um módulo complementar que emula perfis de banco de dados de origem necessários ao converter objetos para o Amazon Redshift. Para obter mais informações, consulte [Usando AWS SCT pacotes de extensão](#).

Para essa operação, AWS SCT precisa ter permissão para acessar o bucket do Amazon S3 em seu nome. Para fornecer essa permissão, crie um usuário (IAM) do AWS Identity and Access Management com a política a seguir.

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

Como escolher estratégias e regras de otimização para uso com a AWS SCT

Para otimizar a forma como a AWS Schema Conversion Tool converte seu esquema de data warehouse, é possível selecionar as estratégias e regras que você deseja que a ferramenta use. Depois de converter seu esquema e analisar as chaves sugeridas, você pode ajustar suas regras ou alterar sua estratégia para obter os resultados desejados.

Para escolher suas estratégias e regras de otimização

1. Selecione Configurações e Configurações do projeto. A caixa de diálogo Configurações atuais do projeto é exibida.
2. No painel de navegação à esquerda, selecione Estratégias de otimização. As estratégias de otimização são exibidas no painel direito com os valores padrão selecionados.
3. Em Setor de estratégia, selecione a estratégia de otimização que deseja usar. Você pode escolher entre as seguintes opções:

- Use metadados, ignore informações estatísticas – Nessa estratégia, as decisões de otimização baseiam-se somente nas informações dos metadados. Por exemplo, se há mais de um índice em uma tabela de origem, a ordem de classificação do banco de dados de origem é usada e o primeiro índice torna-se uma chave de distribuição.
 - Ignore os metadados, use informações estatísticas – Nessa estratégia, as decisões de otimização são derivadas somente das informações estatísticas. Essa estratégia se aplica apenas a tabelas e colunas para as quais as estatísticas são fornecidas. Para obter mais informações, consulte [Como coletar ou carregar as estatísticas para a AWS SCT](#).
 - Use metadados e as informações estatísticas – Nessa estratégia, tanto os metadados quanto as estatísticas são usados nas decisões de otimização.
4. Depois que você escolher sua estratégia de otimização, poderá escolher as regras que deseja usar. Você pode escolher entre as seguintes opções:
- Selecione a chave de distribuição e as chaves de classificação que usam metadados
 - Selecione a tabela de fatos e a dimensão apropriada para os agrupamentos
 - Analise a cardinalidade das colunas dos índices
 - Localize as tabelas e colunas mais usadas na tabela de log de consulta

Para cada regra, você pode especificar um peso para a chave de classificação e um peso para a chave de distribuição. A AWS SCT usa os pesos que você escolher para converter seu esquema. Posteriormente, ao analisar as sugestões de chaves, se você não estiver satisfeito com os resultados, é possível retornar aqui e alterar suas configurações. Para obter mais informações, consulte [Como gerenciar e personalizar chaves na AWS SCT](#).

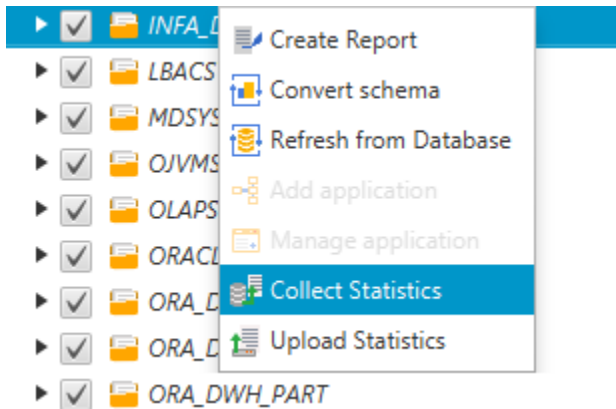
Como coletar ou carregar as estatísticas para a AWS SCT

Para otimizar como a AWS Schema Conversion Tool converte seu esquema de data warehouse, é possível fornecer estatísticas de seu banco de dados de origem que a ferramenta possa usar.

Você pode coletar estatísticas diretamente do banco de dados ou fazer upload de um arquivo de estatísticas existente.

Para fornecer e analisar estatísticas

1. Abra seu projeto e conecte-se ao banco de dados de origem.
2. Escolha um objeto de esquema no painel esquerdo do seu projeto e abra o menu de contexto (clique com o botão direito do mouse) do objeto. Selecione Coletar estatísticas ou Fazer upload de estatísticas conforme mostrado a seguir.



3. Escolha um objeto de esquema no painel à esquerda do projeto e escolha a guia Estatísticas. Você pode analisar as estatísticas do objeto.

Você pode criar regras de migração que executam as seguintes tarefas:

- Adicionar, remover ou substituir um prefixo
- Adicionar, remover ou substituir um sufixo
- Alterar agrupamento de colunas
- Alterar tipo de dados
- Alterar o comprimento dos tipos de dados `char`, `varchar`, `nvarchar` e `string`
- Mover objetos
- Renomear objetos

Você pode criar regras de migração para os seguintes objetos:

- Banco de dados
- Esquema
- Tabela
- Coluna

Como criar regras de migração

Você pode criar regras de migração salvar as regras como parte do seu projeto. Com seu projeto aberto, use o procedimento a seguir para criar regras de migração.

Para criar regras de migração

1. No menu Exibir, selecione Exibição de mapeamento.
2. Em Mapeamentos de servidores, escolha um par de servidores de origem e de destino.
3. Selecione Nova regra de migração. A caixa de diálogo Regras de transformação é exibida.
4. Escolha Adicionar nova regra. Uma nova linha é adicionada à lista de regras.
5. Configure sua regra:
 - a. Em Name (Nome), insira um nome para a regra.
 - b. Em Para, escolha o tipo de objeto ao qual a regra se aplica.
 - c. Em onde, digite um filtro para aplicar aos objetos antes de aplicar a regra de migração. A cláusula `where` é avaliada usando uma cláusula `like`. Você pode inserir um nome exato para selecionar um objeto, ou pode inserir um padrão para selecionar vários objetos.

Os campos disponíveis para a cláusula where são diferentes, dependendo do tipo do objeto. Por exemplo, se o tipo de objeto for esquema, haverá apenas um campo disponível para o nome do esquema.

- d. Em Ações, escolha o tipo de regra de migração que você deseja criar.
 - e. Dependendo do tipo de regra, insira um ou dois valores adicionais. Por exemplo, para renomear um objeto, insira o novo nome do objeto. Para substituir um prefixo, insira o prefixo antigo e o novo prefixo.
6. Após configurar sua regra de migração, selecione Salvar para salvar sua regra. Você também pode escolher Cancelar para cancelar as alterações.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS *doesn't support* more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, + Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Ao finalizar a adição, edição e exclusão das regras, escolha Salvar tudo para salvar todas as alterações.
8. Selecione Fechar para fechar a caixa de diálogo Regras de transformação.

Você pode usar o ícone de alternância para desativar uma regra de migração sem excluí-la. Você pode usar o ícone de cópia para duplicar uma regra de migração existente. Use o ícone de lápis para editar uma regra de migração existente. Você pode usar o ícone de exclusão para excluir uma regra de migração existente. Para salvar as alterações feitas em suas regras de migração, selecione Salvar tudo.

Como exportar regras de migração

Se você usar o AWS Database Migration Service (AWS DMS) para migrar os dados do seu banco de dados de origem para o banco de dados de destino, será possível fornecer informações sobre as regras de migração para o AWS DMS. Para obter mais informações sobre tarefas, consulte [Como trabalhar com tarefas de replicação do AWS Database Migration Service](#).

Para exportar regras de migração

1. Na AWS Schema Conversion Tool, selecione Exibição de mapeamento no menu Exibir.
2. Em Regras de migração, escolha uma regra de migração e, em seguida, selecione Modificar regra de migração.
3. Selecione Exportar script para o AWS DMS.
4. Navegue até o local onde você deseja salvar seu script e escolha Salvar. Suas regras de migração são salvas como um script JSON que pode ser consumido pelo AWS DMS.

Como converter seu esquema usando a AWS SCT

Depois que você tiver conectado o projeto ao banco de dados de origem e de destino, seu projeto da AWS Schema Conversion Tool exibirá o esquema do banco de dados de origem no painel esquerdo. O esquema é apresentado em um formato de visualização em árvore, e cada nó da árvore é de carregamento preguiçoso. Quando você escolhe um nó na visualização em árvore, a AWS SCT solicita as informações do esquema do seu banco de dados de origem nesse momento.

É possível selecionar itens do esquema de seu banco de dados de origem e converter o esquema para o equivalente do mecanismo de banco de dados do seu banco de dados de destino. Você pode escolher qualquer item de esquema de seu banco de dados de origem para converter. Se o item de esquema que você escolher depende de um item pai, a AWS SCT também gerará o esquema para o item pai. Por exemplo, se você escolher uma coluna de uma tabela para converter, a AWS SCT gerará o esquema para a coluna, a tabela em que a coluna está e o banco de dados em que a tabela está.

Converter o esquema

Para converter um esquema do seu banco de dados de origem, marque a caixa de seleção do nome do esquema a ser convertido. Em seguida, selecione esse esquema no painel esquerdo do seu

projeto. A AWS SCT destaca o nome do esquema em azul. Abra o menu de contexto (clique com o botão direito do mouse) do esquema e selecione Converter esquema, conforme mostrado a seguir.

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

- Left Pane (Servers):** Shows a tree view of servers and databases. The 'TEST' database is highlighted in blue. A context menu is open over it, with 'Convert schema' selected.
- Top Right Pane (Properties):** Shows the properties of the selected 'TEST' database. The 'Name' is 'TEST'. Other properties include 'Created or last modified' (2021-09-06 09:56:08.26), 'compatibility-level' (100), and 'collation-name' (SQL_Latin1_General_CP1_CI_AS).
- Bottom Right Pane (Properties):** Shows the properties of the converted schema. The 'Name' is '<Aurora_MySQL (virtual)>'. Other properties include 'Category' and 'Apply status'.

Depois de converter o esquema do seu banco de dados de origem, você pode escolher os itens de esquema no painel esquerdo do seu projeto e visualizar o esquema convertido nos painéis centrais do seu projeto. O painel central inferior exibe as propriedades e o comando SQL para criar o esquema convertido, como mostrado a seguir.

The screenshot shows the AWS Schema Conversion Tool interface. On the left, a tree view displays the source database structure: Servers > SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > AdventureWorks2012_CS, alfresco, GOLD_TEST_SS_PG, LARGE_DB_SS > Schemas [2] > dbo > Tables [4] > Account. The right pane shows the SQL code for the 'Account' table, comparing the source schema with the target Amazon RDS schema.

Source Schema SQL:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Target Amazon RDS for MySQL table: Account SQL:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Depois de converter seu esquema, você poderá salvar seu projeto. As informações do esquema de seu banco de dados de origem são salvas com o seu projeto. Essa funcionalidade significa que você pode trabalhar off-line sem estar conectado ao seu banco de dados de origem. A AWS SCT vai se conectar ao seu banco de dados de origem para atualizar o esquema em seu projeto, se você selecionar Atualizar do banco de dados. Para obter mais informações, consulte [Como atualizar e recarregar o esquema convertido na AWS SCT](#).

Você pode criar um relatório de avaliação de migração do banco de dados dos itens que não podem ser convertidos automaticamente. O relatório de avaliação é útil para identificar e resolver itens de esquema que não podem ser convertidos automaticamente. Para obter mais informações, consulte [Como criar relatórios de avaliação de migração com a AWS SCT](#).

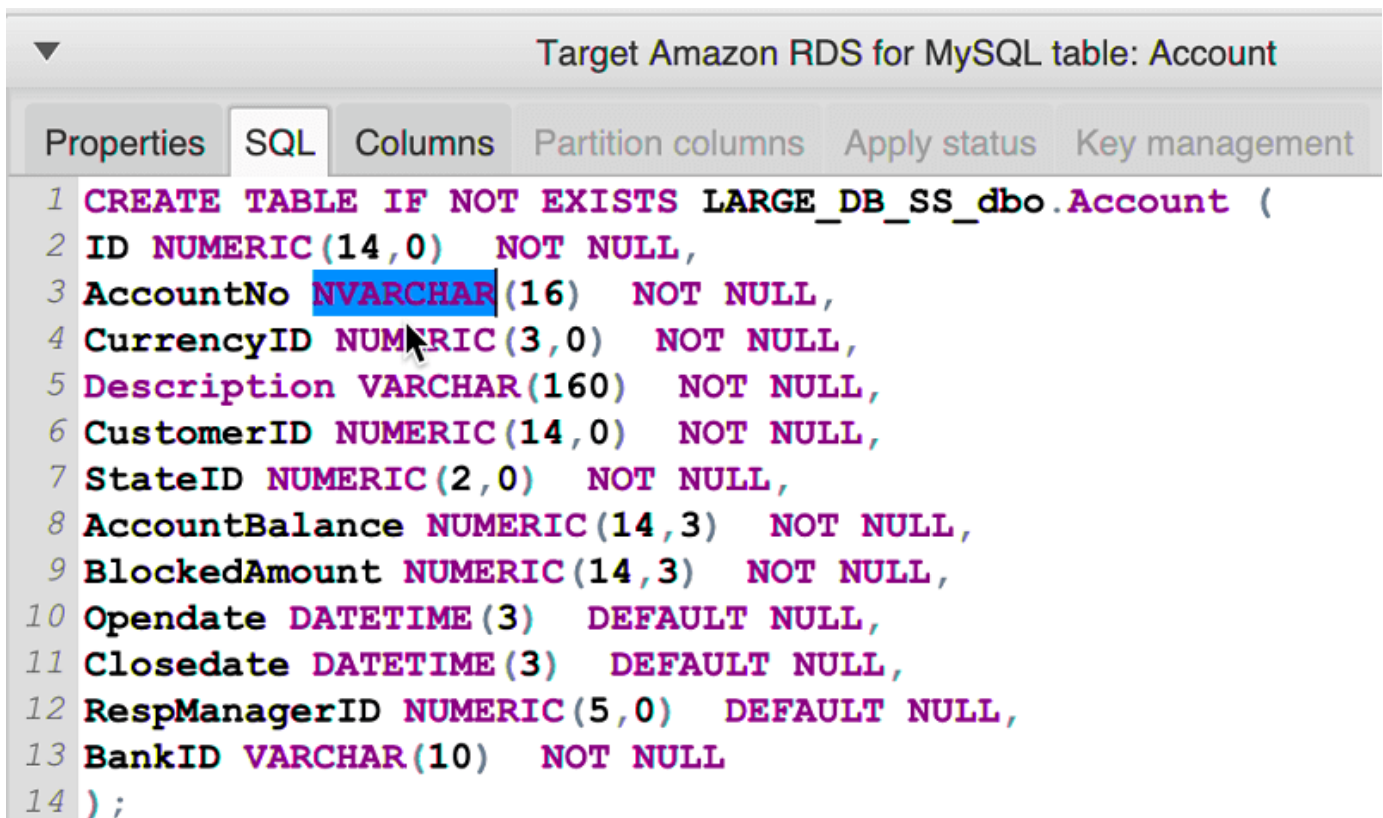
Quando a AWS SCT gera um esquema convertido, ela não o aplica imediatamente ao banco de dados de destino. Em vez disso, o esquema convertido é armazenado localmente até que você esteja pronto para aplicá-lo ao banco de dados de destino. Para obter mais informações, consulte [Como aplicar seu esquema convertido](#).

Como editar o esquema convertido

Você pode editar o esquema convertido e salvar as alterações como parte do seu projeto.

Para editar o esquema convertido

1. No painel esquerdo que exibe o esquema de seu banco de dados de origem, escolha o item do esquema para o qual você deseja editar o esquema convertido.
2. No painel central inferior que exibe o esquema convertido referente ao item selecionado, escolha a guia SQL.
3. No texto exibido referente à guia SQL, altere o esquema conforme necessário. O esquema é salvo automaticamente com o seu projeto, à medida que você o atualiza.



The screenshot shows the AWS Schema Conversion Tool interface. At the top, it says "Target Amazon RDS for MySQL table: Account". Below this, there are several tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". The "SQL" tab is selected. The SQL code displayed is:

```
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (  
2 ID NUMERIC(14,0) NOT NULL,  
3 AccountNo NVARCHAR(16) NOT NULL,  
4 CurrencyID NUMERIC(3,0) NOT NULL,  
5 Description VARCHAR(160) NOT NULL,  
6 CustomerID NUMERIC(14,0) NOT NULL,  
7 StateID NUMERIC(2,0) NOT NULL,  
8 AccountBalance NUMERIC(14,3) NOT NULL,  
9 BlockedAmount NUMERIC(14,3) NOT NULL,  
10 Opendate DATETIME(3) DEFAULT NULL,  
11 Closedate DATETIME(3) DEFAULT NULL,  
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,  
13 BankID VARCHAR(10) NOT NULL  
14 );
```

As alterações que você fizer no esquema convertido serão armazenadas com seu projeto à medida que você o atualiza. Se você converteu recentemente um item do esquema de seu banco de dados de origem, e fez atualizações no esquema convertido anteriormente para esse item, essas atualizações existentes serão substituídas pelo item do esquema recém-convertido com base em seu banco de dados de origem.

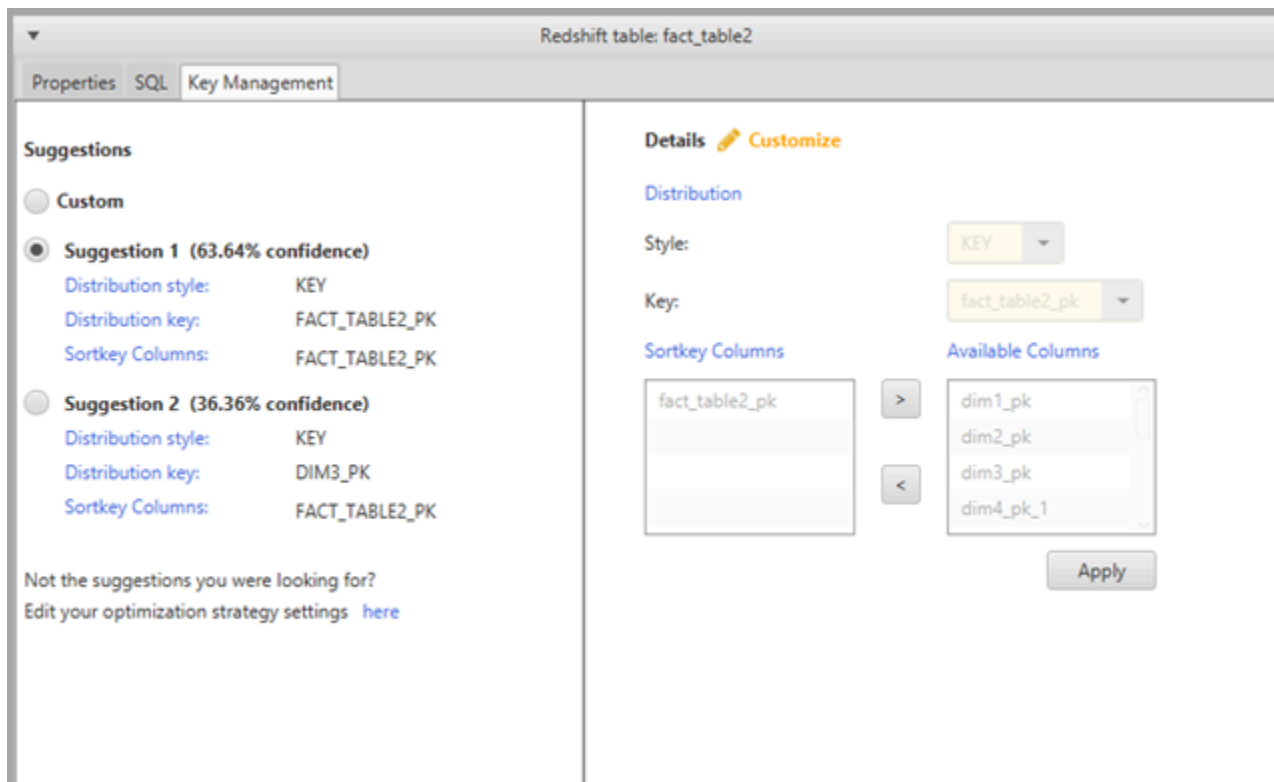
Como limpar um esquema convertido

Até que você aplique o esquema ao banco de dados de destino, a AWS SCT armazenará apenas o esquema convertido localmente em seu projeto. Basta limpar o esquema planejado do projeto, escolhendo o nó de visualização em árvore referente ao banco de dados de destino e, em seguida, escolher a opção Atualizar do banco de dados. Como nenhum esquema foi gravado no banco de dados de destino, atualizar o banco de dados removerá os elementos do esquema planejado do projeto da AWS SCT a fim de coincidir com o existente no banco de dados de destino.

Como gerenciar e personalizar chaves na AWS SCT

Depois de converter o esquema com a AWS Schema Conversion Tool, você poderá gerenciar e editar suas chaves. O gerenciamento de chaves é a essência da conversão de data warehouse.

Para gerenciar chaves, selecione uma tabela em seu banco de dados de destino e escolha a guia Gerenciamento de chaves conforme mostrado a seguir.



O painel esquerdo contém as principais sugestões, e inclui a classificação de confiança de cada sugestão. Você pode escolher uma das sugestões, ou pode personalizar a chave, editando-a no painel direito.

Se as opções para a chave não parecerem com o que você espera, será possível editar suas estratégias de otimização e, em seguida, tentar executar a conversão novamente. Para obter mais informações, consulte [Como escolher estratégias e regras de otimização para uso com a AWS SCT](#).

Tópicos relacionados

- [Selecione a melhor chave de classificação](#)
- [Selecione o melhor estilo de distribuição](#)

Como criar e usar o relatório de avaliação na AWS SCT

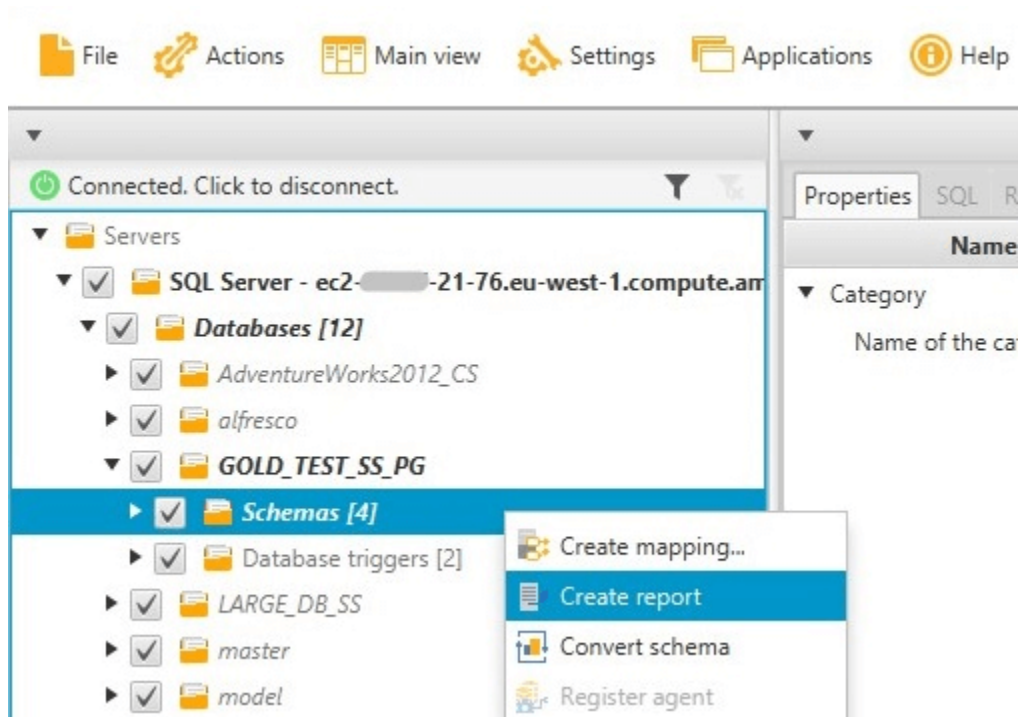
A AWS Schema Conversion Tool cria um relatório de avaliação de migração do banco de dados para ajudar você a converter o esquema. O relatório de avaliação da migração de banco de dados fornece informações importantes sobre a conversão do esquema do banco de dados de origem para o de destino. O relatório resume todas as tarefas de conversão e detalha os itens de ação do esquema que não podem ser convertidos para o mecanismo de banco de dados do banco de dados de destino. O relatório também inclui estimativas do trabalho necessário para gravar o código equivalente no banco de dados de destino que não pode ser convertido automaticamente.

Como criar um relatório de avaliação de migração do banco de dados

Use o procedimento a seguir para criar um relatório de avaliação de migração do banco de dados.

Para criar um relatório de avaliação de migração do banco de dados

1. No painel esquerdo que exibe o esquema do banco de dados de origem, escolha um objeto de esquema para o qual criar um relatório de avaliação.
2. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Criar relatório.



Resumo do relatório de avaliação

Depois que você criar um relatório de avaliação, a visualização do relatório de avaliação será aberta, mostrando a guia Resumo. A guia Resumo exibe as informações de resumo do relatório de avaliação de migração do banco de dados. Ela mostra os itens que foram convertidos automaticamente e os itens que não foram convertidos automaticamente.

Summary
Action items

Save to CSV
Save to PDF

Database migration assessment report

Source database: GOLD_TEST_SS_PG: >21-76.eu-west-1.compute.amazonaws.com/GOLD_TEST_SS_PG:1433
 Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (x64) Mar 22 2021 18:10:24
 Copyright (C) 2019 Microsoft Corporation
 Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763:) (Hypervisor)
 Case sensitivity: OFF

Executive summary

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

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

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects

Object Type	Count	Automatically Converted (%)	Simple Actions (%)	Medium-Complexity Actions (%)	Complex Actions (%)
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	2%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	35%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	0%	80%	0%

Para os itens de esquema que não podem ser convertidos automaticamente para o mecanismo de banco de dados de destino, o resumo inclui uma estimativa do trabalho necessário para criar itens de esquema na instância de banco de dados de destino que equivalem aos de origem.

O relatório classifica o tempo estimado de conversão desses itens de esquema da seguinte maneira:

- **Simple** – Ações que podem ser concluídas em menos de uma hora.
- **Média** – Ações mais complexas que podem ser concluídas de uma a quatro horas.
- **Significativa** – Ações muito complexas que levarão mais do que quatro horas para serem concluídas.

Itens de ação do relatório de avaliação

A visualização do relatório de avaliação também inclui uma guia Itens de ação. Esta guia contém uma lista de itens que não podem ser convertidos automaticamente para o mecanismo de banco de dados de dados de destino. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu esquema ao qual o item de ação se aplica.

O relatório também contém recomendações sobre como converter manualmente o item do esquema. Para obter mais informações sobre decidir como lidar com conversões manuais, consulte [Como lidar com conversões manuais na AWS SCT](#).

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

- Summary / Action items:** A tabbed interface with 'Action items' selected.
- Left Panel:** A tree view showing the database structure. It includes 'Servers' (SQL Server - ec2-21-76.eu-west-1.compute), 'Databases [12]' (AdventureWorks2012_CS, alfresco, GOLD_TEST_SS_PG, LARGE_DB_SS, master, model, msdb, tempdb), 'Schemas [1]' (dbo), 'Tables [8]' (MSSQL_TemporalHistoryFor_1013578, MSSQL_TemporalHistoryFor_9655784, NonPartitionTable, PartitionTable, Position, test, tmp_tbl_sys_ver (System-Versioned), tmp_tbl_sys_ver_alter (System-Version)), 'Graph Tables', 'External Tables', 'Views', 'Procedures [4]' (P_INS_OUTP, POSITION_UPDATE_CASH_CGT_BULK, test2, test3), 'SQL scalar functions', 'SQL table-valued functions', and 'SQL inline functions'.
- Right Panel (Issues):** A list of issues with details:
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes. Number of occurrences: 2
 - Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary. Number of occurrences: 1
 - Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable. Number of occurrences: 1
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy. Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
 - Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually. Number of occurrences: 3
 - Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion. Number of occurrences: 1
 - Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function. Number of occurrences: 12
- Bottom Panel:** A detailed view of the selected issue (Issue 690). It shows the source Microsoft SQL Server procedure code:


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

 The target is Amazon RDS for MySQL category: Schemas.

Como salvar o relatório de avaliação

Você pode salvar uma cópia local do relatório de avaliação de migração do banco de dados como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV). O arquivo CSV contém apenas as informações do item de ação. O arquivo PDF contém as informações de item de ação e de resumo, conforme mostrado no exemplo a seguir.

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

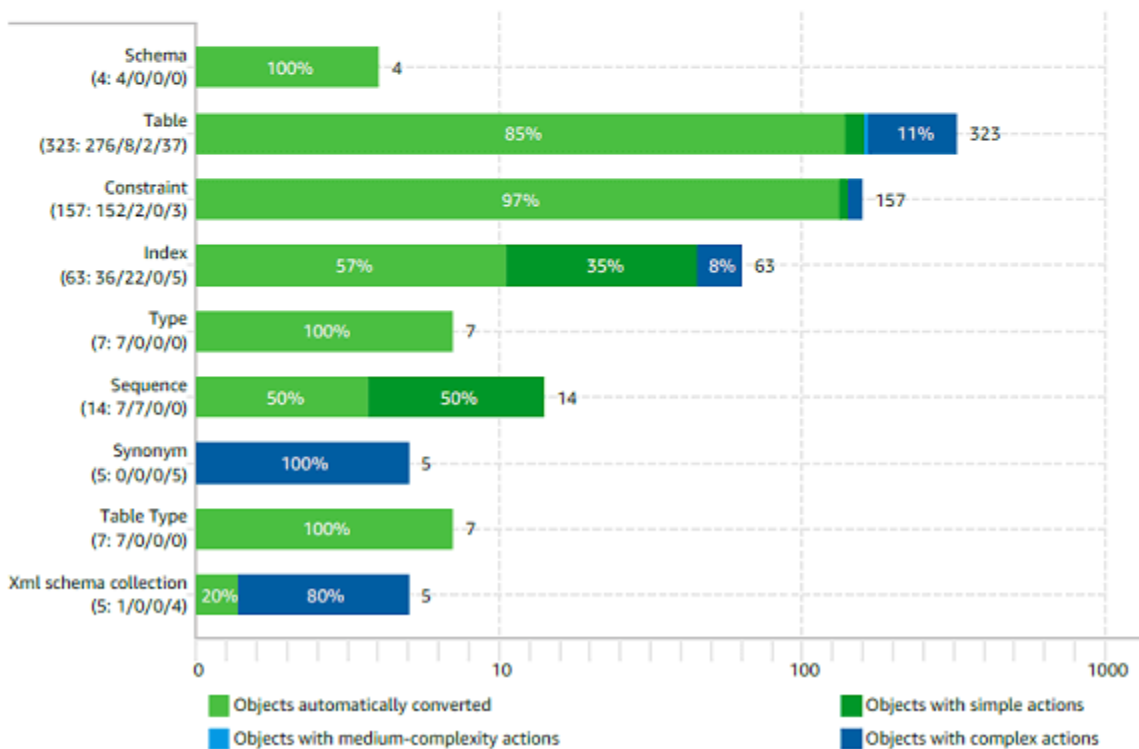
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



Como lidar com conversões manuais na AWS SCT

O relatório de avaliação inclui uma lista de itens que não podem ser convertidos automaticamente para o mecanismo de banco de dados do banco de dados de destino. Para cada item que não pode ser convertido, existe um item de ação na guia Itens de ação.

Você pode responder aos itens de ação do relatório de avaliação das seguintes formas:

- Modifique o esquema do banco de dados de origem.

- Modifique o esquema do banco de dados de destino.

Como modificar seu esquema de origem

Para alguns itens, pode ser mais fácil modificar o esquema de banco de dados no seu banco de dados de origem para o esquema que pode ser convertido automaticamente. Primeiro, verifique se as novas alterações são compatíveis com sua arquitetura de aplicativo e, em seguida, atualizar o esquema no seu banco de dados de origem. Finalmente, atualize seu projeto com as informações do esquema atualizado. É possível converter o esquema atualizado, e gerar um novo relatório de avaliação de migração de banco de dados. Os itens de ação não são mais exibidos para os itens que são alterados no esquema de origem.

A vantagem desse processo é que o esquema atualizado está sempre disponível quando você atualiza a partir de seu banco de dados de origem.

Como modificar seu esquema de destino

Para alguns itens, pode ser mais fácil aplicar o esquema convertido ao seu banco de dados de destino e, em seguida, adicionar itens de esquema equivalente manualmente ao seu banco de dados de destino para os itens que não puderam ser convertidos automaticamente. Você pode gravar tudo do esquema que pode ser convertido automaticamente para o banco de dados de destino, aplicando o esquema. Para obter mais informações, consulte [Como salvar e aplicar seu esquema convertido na AWS SCT](#).

O esquema gravado no banco de dados de destino não contém os itens que não podem ser convertidos automaticamente. Depois de aplicar o esquema ao banco de dados de destino, você pode criar manualmente o esquema no banco de dados de destino equivalente ao do banco de dados de origem. Os itens de ação no relatório de avaliação de migração de banco de dados contêm sugestões de como criar o esquema equivalente.

Warning

Se você criar manualmente o esquema no banco de dados de destino, salve uma cópia de qualquer trabalho manual realizado. Se você aplicar o esquema convertido do seu projeto ao banco de dados de destino novamente, ele substituirá o trabalho manual realizado.

Em alguns casos, não é possível criar um esquema equivalente no banco de dados de destino. Você pode precisar rearquitar uma parte do aplicativo e do banco de dados para usar a funcionalidade disponível no mecanismo de banco de dados para o banco de dados de destino. Em outros casos, você pode simplesmente ignorar o esquema que não pode ser convertido automaticamente.

Como atualizar e recarregar o esquema convertido na AWS SCT

Você pode atualizar o esquema de origem e o esquema de destino em seu projeto da AWS Schema Conversion Tool.

- **Origem:** Se você atualizar o esquema de seu banco de dados de origem, a AWS SCT substituirá o esquema em seu projeto pelo esquema mais recente do seu banco de dados de origem. Usando essa funcionalidade, você pode atualizar seu projeto, se as alterações tiverem sido feitas no esquema de seu banco de dados de origem.
- **Destino** – Se você atualizar o esquema do seu banco de dados de destino, a AWS SCT substituirá o esquema de seu projeto pelo esquema mais recente do banco de dados de destino. Se você não aplicou qualquer esquema ao banco de dados de destino, a AWS SCT apagará o esquema convertido do seu projeto. Em seguida, você poderá converter o esquema do seu banco de dados de origem para um banco de dados de destino limpo.

Atualize o esquema em seu projeto da AWS SCT, selecionando Atualizar do banco de dados.

Como salvar e aplicar seu esquema convertido na AWS SCT

Quando a AWS Schema Conversion Tool gerar o esquema convertido (conforme mostrado em [Como converter seu esquema usando a AWS SCT](#)), ela não aplicará imediatamente o esquema convertido ao banco de dados de destino. Em vez disso, o esquema convertido será armazenado localmente no projeto, até que você esteja pronto para aplicá-lo ao banco de dados de destino. Usando essa funcionalidade, você pode trabalhar com os itens do esquema que não podem ser convertidos automaticamente para o mecanismo de banco de dados de destino. Para obter mais informações sobre itens que não podem ser convertidos automaticamente, consulte [Como criar relatórios de avaliação de migração com a AWS SCT](#).

Opcionalmente, você pode fazer a ferramenta salvar o esquema convertido em um arquivo como um script SQL antes de aplicar o esquema ao banco de dados de destino. Você também pode fazer com que a ferramenta aplique o esquema convertido diretamente ao banco de dados de destino.

Como salvar seu esquema convertido em um arquivo

Você pode salvar seu esquema convertido como scripts SQL em um arquivo de texto. Ao usar essa abordagem, você pode modificar os scripts SQL gerados da AWS SCT para lidar com itens que a ferramenta não pode converter automaticamente. Em seguida, você pode executar seus scripts atualizados em sua instância de banco de dados de destino para aplicar o esquema convertido ao seu banco de dados de destino.

Como salvar o esquema convertido como scripts SQL

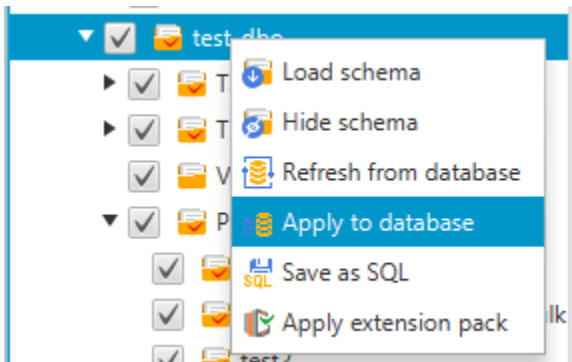
1. Selecione seu esquema e abra o menu de contexto (clique com o botão direito do mouse).
2. Selecione Salvar como SQL.
3. Insira o nome do arquivo e selecione Salvar.
4. Salve seu esquema convertido usando uma das seguintes opções:
 - Arquivo único
 - Arquivo único por estágio
 - Arquivo único por instrução

Para escolher o formato do script SQL

1. No menu Configurações, selecione Configurações do projeto.
2. Selecione Salvar scripts.
3. Em Fornecedor, selecione a plataforma de banco de dados.
4. Em Salvar scripts de SQL em, selecione como você deseja salvar o script do esquema de banco de dados.
5. Escolha OK para salvar as configurações.

Como aplicar seu esquema convertido

Quando estiver pronto para aplicar o esquema convertido ao banco de dados de destino, escolha o elemento de esquema no painel à direita do projeto. Abra o menu de contexto (clique com o botão direito do mouse) do elemento do esquema e escolha Aplicar ao banco de dados, como mostrado a seguir.



O esquema do pacote de extensão

Na primeira vez em que você aplicar o esquema convertido à sua instância de banco de dados de destino, a AWS SCT adicionará um esquema adicional à sua instância de banco de dados de destino. Esse esquema implementa as funções de sistema do banco de dados de origem necessárias para gravar o esquema convertido na instância de banco de dados de destino. O esquema é chamado de esquema de pacote de extensões.

Não modifique o esquema do pacote de extensões, ou você poderá obter resultados inesperados no esquema convertido que é gravado na instância de banco de dados de destino. Quando seu esquema estiver totalmente migrado para sua instância de banco de dados de destino, e você não precisar mais da AWS SCT, será possível excluir o esquema do pacote de extensões.

O esquema do pacote de extensões é nomeado de acordo com seu banco de dados de origem da seguinte forma:

- 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 obter mais informações, consulte [Usando AWS SCT pacotes de extensão](#).

Bibliotecas Python

Para criar funções personalizadas no Amazon Redshift, você deve usar a linguagem Python. Use o pacote de extensão da AWS SCT para instalar as bibliotecas python no banco de dados do Amazon Redshift. Para obter mais informações, consulte [Usando AWS SCT pacotes de extensão](#).

Como otimizar o Amazon Redshift usando a AWS SCT

Você pode usar a AWS Schema Conversion Tool para otimizar seu banco de dados do Amazon Redshift. Usando seu banco de dados do Amazon Redshift como origem, e um banco de dados de teste do Amazon Redshift como destino, a AWS SCT recomenda chaves de classificação e chaves de distribuição para otimizar seu banco de dados.

Como otimizar seu banco de dados do Amazon Redshift

Use o procedimento a seguir para otimizar seu banco de dados do Amazon Redshift.

Para otimizar seu banco de dados do Amazon Redshift

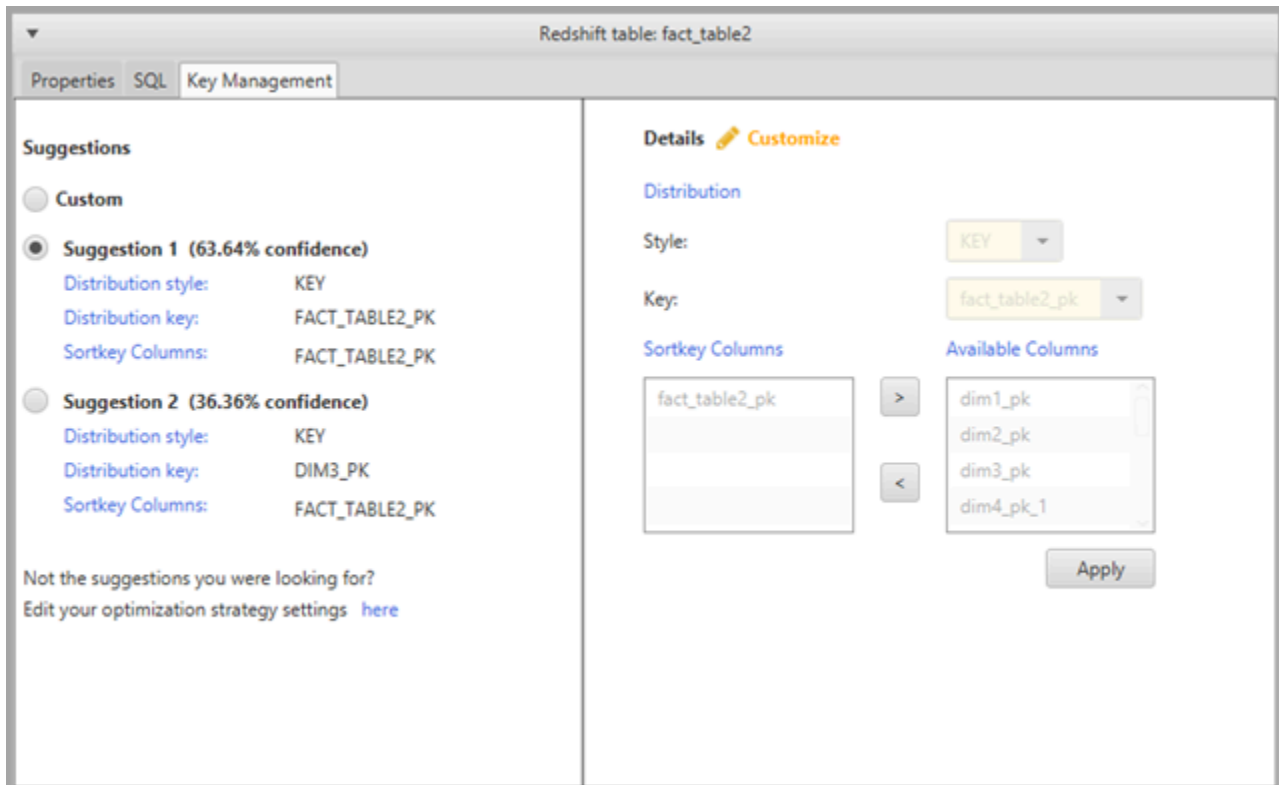
1. Faça um snapshot manual de seu cluster do Amazon Redshift como um backup. Você pode excluir o snapshot após concluir a otimização do seu cluster do Amazon Redshift e testar quaisquer alterações feitas. Para obter mais informações, consulte [Snapshots do Amazon Redshift](#).
2. Escolha um objeto de esquema a ser convertido no painel esquerdo do seu projeto. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Coletar estatísticas.

A AWS SCT usa as estatísticas para fazer sugestões para chaves de classificação e chaves de distribuição.

3. Escolha um objeto de esquema a ser otimizado no painel esquerdo do seu projeto. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Executar otimização.

A AWS SCT faz sugestões para chaves de classificação e chaves de distribuição.

4. Para analisar as sugestões, expanda o nó de tabelas, sob seu esquema no painel esquerdo de seu projeto, e escolha uma tabela. Escolha a guia Gerenciamento de chaves, como mostrado a seguir.



O painel esquerdo contém as principais sugestões, e inclui a classificação de confiança de cada sugestão. Você pode escolher uma das sugestões, ou pode personalizar a chave, editando-a no painel direito.

5. Você pode criar um relatório que contém as sugestões de otimização. Para criar o relatório, faça o seguinte:

a. Escolha um objeto de esquema que você otimizou no painel esquerdo do seu projeto. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Criar relatório.

O relatório é aberto na janela principal, e a guia Resumo é exibida. O número de objetos com sugestões de otimização é exibido no relatório.

b. Escolha a guia Itens de ação para ver as principais sugestões em um formato de relatório.

c. Você pode salvar uma cópia local do relatório de otimização como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV). O arquivo CSV contém apenas as informações do item de ação. O arquivo PDF contém as informações do resumo e do item de ação.

6. Para aplicar as otimizações sugeridas ao seu banco de dados, escolha um objeto no painel direito do seu projeto. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Aplicar ao banco de dados.

Conversão de processos de extração, transformação e carregamento (ETL) com o AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para migrar processos de extração, transformação e carregamento (ETL). Esse tipo de migração inclui a conversão da lógica de negócios relacionada a ETL. Essa lógica pode residir dentro dos data warehouses de origem ou em scripts externos que você executa separadamente.

Atualmente, o AWS SCT suporta a conversão de scripts de ETL em objetos para o AWS Glue e o Amazon Redshift RSQL, conforme mostrado na tabela a seguir.

Origem	Destino
Scripts de ETL da Informatica	Informatica
Pacotes de ETL do Microsoft SQL Server Integration Services (SSIS)	AWS Glue ou AWS Glue Studio
Scripts de shell com comandos incorporados do Teradata Basic Teradata Query (BTEQ)	Amazon Redshift RSQL
Scripts de ETL do Teradata BTEQ	AWS Glue ou Amazon Redshift RSQL
Scripts de tarefa do Teradata FastExport	Amazon Redshift RSQL
Scripts de tarefa do Teradata FastLoad	Amazon Redshift RSQL
Scripts de tarefa do Teradata MultiLoad	Amazon Redshift RSQL

Tópicos

- [Como converter processos de ETL para o AWS Glue com o AWS SCT](#)
- [Como converter processos de ETL usando a API Python para o AWS Glue com o AWS SCT](#)
- [Como converter scripts de ETL da Informatica com o AWS SCT](#)
- [Como converter SSIS em AWS Glue com a AWS SCT](#)
- [Como converter SSIS em AWS Glue Studio com a AWS SCT](#)

- [Como converter scripts Teradata BTEQ em Amazon Redshift RSQL com a AWS SCT](#)
- [Como converter scripts de shell com comandos Teradata BTEQ incorporados em Amazon Redshift RSQL com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata FastExport em Amazon Redshift RSQL com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata FastLoad em Amazon Redshift RSQL com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata MultiLoad em Amazon Redshift RSQL com a AWS SCT](#)

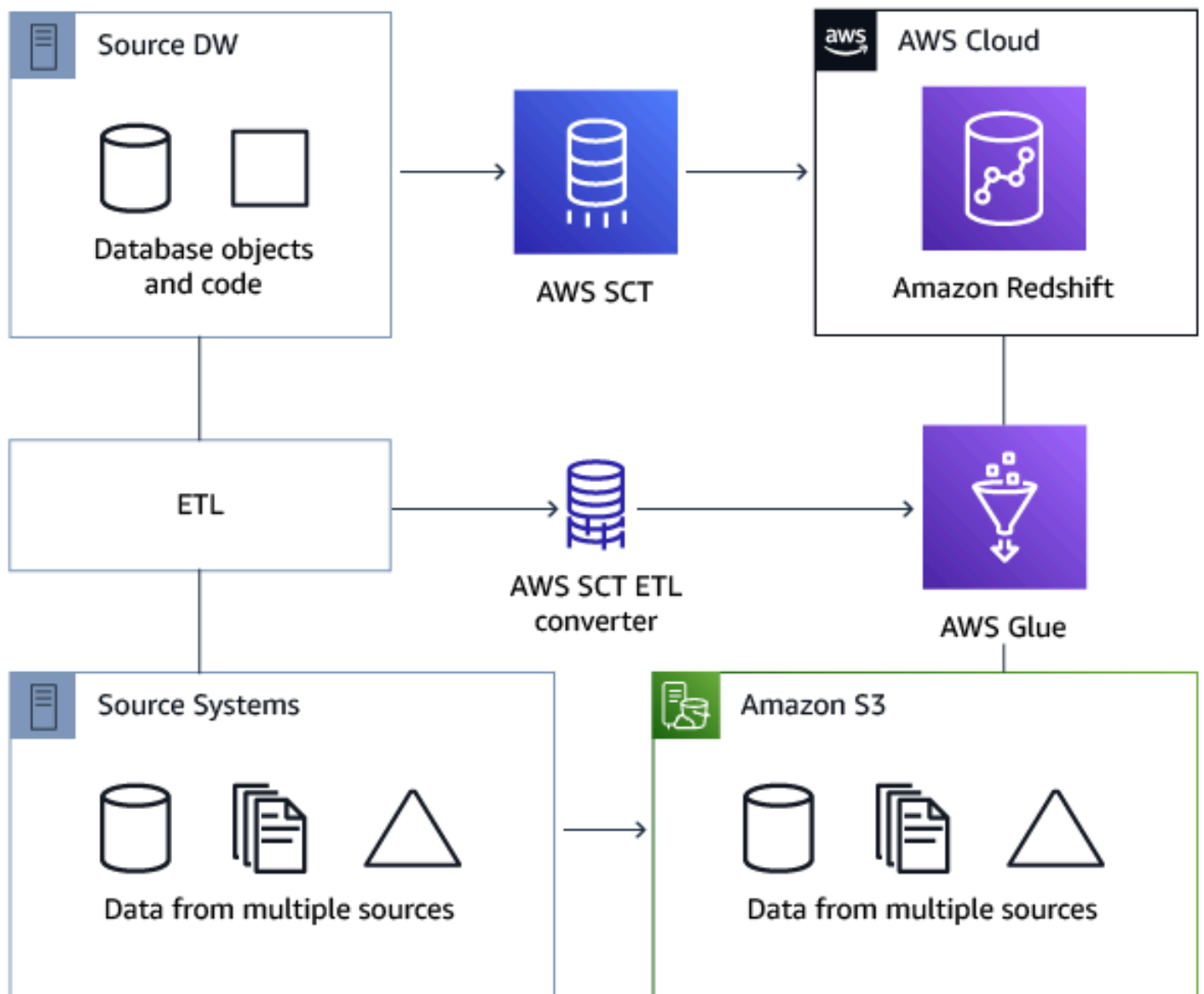
Como converter processos de ETL para o AWS Glue com o AWS SCT

A seguir, você pode encontrar uma descrição do processo que converte scripts de ETL para o AWS Glue usando o AWS SCT. Neste exemplo, converteremos um banco de dados Oracle para Amazon Redshift com os processos de ETL usados nos data warehouses e bancos de dados de origem.

Tópicos

- [Pré-requisitos](#)
- [Noções básicas sobre o catálogo de dados do AWS Glue](#)
- [Limitações ao converter usando o AWS SCT com o AWS Glue](#)
- [Etapa 1: criar um novo projeto](#)
- [Etapa 2: criar um tarefa do AWS Glue](#)

O diagrama de arquitetura a seguir mostra um projeto de migração de banco de dados que inclui a conversão de scripts ETL em AWS Glue.



Pré-requisitos

Antes de começar, faça o seguinte:

- Migre todos os bancos de dados de origem que pretende migrar para a AWS.
- Migre os data warehouses de destino para a AWS.
- Colete uma lista de todos os códigos envolvidos em seu processo de ETL.
- Colete uma lista de todas as informações de conexão necessárias para cada banco de dados.

O AWS Glue precisa de permissões para acessar outros recursos da AWS em seu nome. Você concede essas permissões usando o AWS Identity and Access Management (IAM). Certifique-se de ter criado uma política do IAM para o AWS Glue. Para obter mais informações, consulte [Criar uma política do IAM para o AWS Glueservice](#) no Guia do desenvolvedor do AWS Glue.

Noções básicas sobre o catálogo de dados do AWS Glue

Como parte do processo de conversão, o AWS Glue carrega informações sobre os bancos de dados de origem e destino. Ele organiza essas informações em categorias, em uma estrutura chamada de árvore. A estrutura inclui o seguinte:

- Conexões - parâmetros de conexão
- Crawlers - uma lista de crawlers, um crawler para cada esquema
- Bancos de dados - contêineres que contêm tabelas
- Tabelas - definições de metadados que representam os dados nas tabelas
- Tarefas de ETL - lógica de negócios que executa a tarefa de ETL
- Gatilhos - lógica que controla quando uma tarefa de ETL é executada no AWS Glue (se sob demanda, por programação, ou acionado por eventos de tarefa)

O Catálogo de dados do AWS Glue é um índice para as métricas de localização, esquema e tempo de execução de seus dados. Quando você trabalha com o AWS Glue e o AWS SCT, o catálogo de dados do AWS Glue contém referências aos dados que são usados como origens e destinos das suas tarefas de ETL no AWS Glue. Para criar seu data warehouse, catalogue esses dados.

Use as informações no catálogo de dados para criar e monitorar suas tarefas de ETL. Normalmente, você executa um crawler para fazer o inventário dos dados nos armazenamentos de dados, mas há outras maneiras de adicionar tabelas de metadados ao seu Catálogo de dados.

Ao definir uma tabela no catálogo de dados, você a adiciona a um banco de dados. Um banco de dados é usado para organizar tabelas no AWS Glue.

Limitações ao converter usando o AWS SCT com o AWS Glue

As seguintes limitações se aplicam ao fazer conversões usando o AWS SCT com o AWS Glue

Recurso	Limite padrão
---------	---------------

Número de bancos de dados para cada conta	10.000
Número de tabelas para cada banco de dados	100.000
Número de partições para cada tabela	1.000.000
Número de versões de tabela para cada tabela	100.000
Número de tabelas para cada conta	1.000.000
Número de partições para cada conta	10,000,000
Número de versões de tabela para cada conta	1.000.000
Número de conexões para cada conta	1.000
Número de crawlers para cada conta	25
Número de tarefas para cada conta	25
Número de gatilhos para cada conta	25
Número de execuções de tarefas simultâneas para cada conta	30
Número de execuções de tarefas simultâneas para cada tarefa	3
Número de tarefas para cada gatilho	10
Número de endpoints de desenvolvimento para cada conta	5
Máximo de unidades de processamento de dados (DPUs) usadas por um endpoint de desenvolvimento por vez	5
Máximo de DPUs usadas por uma função por vez	100

Tamanho do nome do banco de dados	Ilimitado Para compatibilidade com outros armazenamentos de metadados, como o Apache Hive, o nome é alterado para usar caracteres minúsculos. Se você planeja acessar o banco de dados a partir do Amazon Athena, forneça um nome somente com caracteres alfanuméricos e sublinhados.
Tamanho do nome da conexão	Ilimitado
Tamanho do nome do crawler	Ilimitado

Etapa 1: criar um novo projeto

Para criar um novo projeto, siga estas etapas de alto nível:

1. Crie um novo projeto na AWS SCT. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#).
2. Adicione seus bancos de dados de origem e destino ao projeto. Para obter mais informações, consulte [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).

Certifique-se de ter escolhido Usar AWS Glue nas configurações de conexão do banco de dados de destino. Para fazer isso, escolha a guia AWS Glue. Em Copiar do perfil da AWS, escolha o perfil que você deseja usar. O perfil deve preencher automaticamente a chave de acesso da AWS, a chave secreta e a pasta do bucket do Amazon S3. Se isso não ocorrer, insira essas informações por conta própria. Depois que você selecionar OK, o AWS Glue analisará os objetos e carregará os metadados no catálogo de dados do AWS Glue.

Dependendo de suas configurações de segurança, você poderá receber uma mensagem de aviso informando que sua conta não tem privilégios suficientes para alguns dos esquemas no servidor. Se você tiver acesso aos esquemas que está usando, poderá ignorar essa mensagem com segurança.

3. Para concluir a preparação para importar o ETL, conecte-se com os bancos de dados de origem e destino. Para fazer isso, escolha seu banco de dados na árvore de metadados de origem ou de destino e, em seguida, selecione Conectar ao servidor.

O AWS Glue cria um banco de dados no servidor do banco de dados de origem e de destino para ajudar com a conversão de ETL. O banco de dados no servidor de destino contém o catálogo de dados do AWS Glue. Para localizar objetos específicos, use a pesquisa nos painéis de origem ou destino.

Para ver como um objeto específico é convertido, localize um item que você deseja converter e selecione Converter esquema no menu de contexto (clique com o botão direito do mouse). O AWS SCT transforma o objeto selecionado em um script.

Você pode revisar o script convertido na pasta Scripts, no painel direito. Atualmente, o script é um objeto virtual, disponível somente como parte do seu projeto do AWS SCT.

Para criar um tarefa do AWS Glue com seu script convertido, faça o upload dele para o Amazon S3. Para isso, escolha o script e selecione Salvar no S3 no menu de contexto (clique com o botão direito do mouse).

Etapa 2: criar um tarefa do AWS Glue

Depois de salvar o script no Amazon S3, selecione-o e escolha Configurar tarefa do AWS Glue para abrir o assistente e configurar a tarefa do AWS Glue. O assistente facilita essa configuração:

1. A primeira guia, Fluxo de dados de design, permite escolher uma estratégia de execução e a lista de scripts que você deseja incluir na tarefa. Você pode escolher parâmetros para cada script. Também é possível reorganizar os scripts para que eles sejam executados na ordem correta.
2. Na segunda guia, dê um nome à tarefa e configure diretamente as definições para o AWS Glue. Nessa tela, configure as seguintes definições:
 - Perfil do IAM AWS Identity and Access Management
 - Nomes e caminhos de arquivo do script
 - Criptografe o script usando criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3)
 - Diretório temporário
 - Caminho da biblioteca Python gerada
 - Caminho da biblioteca Python do usuário

- Caminho para os arquivos .jar dependentes
- Caminho de arquivos referenciados
- DPUs simultâneas para cada execução de tarefa
- Máximo de simultaneidade
- Tempo limite da tarefa (em minutos)
- Limite de notificação de atraso (em minutos)
- Número de novas tentativas
- Configuração de segurança
- Criptografia do lado do servidor

3. Na terceira etapa, ou guia, escolha a conexão configurada com o endpoint de destino.

Ao terminar de configurar a tarefa, ela será exibida nas tarefas de ETL no catálogo de dados do AWS Glue. Se você escolher a tarefa, as configurações serão exibidas para que possa rever e editá-las. Para criar uma nova tarefa no AWS Glue, escolha Criar tarefa do AWS Glue no menu de contexto da tarefa (clique com o botão direito do mouse). Isso aplica a definição de esquema. Para atualizar a exibição, selecione Atualizar a partir do banco de dados no menu de contexto (clique com o botão direito do mouse).

Neste momento, você pode visualizar a tarefa no console do AWS Glue. Para isso, faça login no AWS Management Console e abra o console do AWS Glue em <https://console.aws.amazon.com/glue/>.

Você pode testar a nova tarefa para garantir que ela está funcionando corretamente. Para fazer isso, verifique os dados na tabela de origem e se a tabela de destino está vazia. Execute a tarefa e verifique novamente. Você pode visualizar os logs de erros no console do AWS Glue.

Como converter processos de ETL usando a API Python para o AWS Glue com o AWS SCT

Nas seções a seguir, você encontra a descrição de uma conversão que chama operações de API do AWS Glue em Python. Para obter mais informações, consulte [Programar scripts de ETL do AWS Glue em Python](#) no Guia do desenvolvedor do AWS Glue.

Tópicos

- [Etapa 1: criar um banco de dados](#)

- [Etapa 2: criar uma conexão](#)
- [Etapa 3: criar um crawler do AWS Glue](#)

Etapa 1: criar um banco de dados

A primeira etapa é criar um novo banco de dados em um catálogo de dados do AWS Glue usando a [API do SDK da AWS](#). Ao definir uma tabela no catálogo de dados, adicione-a a um banco de dados. Um banco de dados é usado para organizar tabelas no AWS Glue.

O exemplo a seguir demonstra o método `create_database` da API do AWS Glue em Python.

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Se você estiver usando o Amazon Redshift, o nome do banco de dados é formado da maneira a seguir.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

O nome completo do cluster do Amazon Redshift para este exemplo é o seguinte.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Abaixo, você vê um exemplo de um nome do banco de dados bem estruturado. Nesse caso, `rsdbb03` é o nome, que é a primeira parte do nome completo do endpoint do cluster. O banco de dados é chamado de `dev` e o esquema, de `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Etapa 2: criar uma conexão

Crie uma conexão em um catálogo de dados usando a [API do SDK da AWS](#).

O exemplo a seguir demonstra o uso do método [create_connection](#) da API do AWS Glue em Python.

```
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'  
            ]  
        }  
    }  
)
```

Os parâmetros usados em `create_connection` são os seguintes:

- **Name** (string UTF-8) - obrigatório. Para o Amazon Redshift, o nome da conexão é formado da seguinte forma: `Redshift_<Endpoint-name>_<redshift-database-name>`. Por exemplo, `Redshift_abcde03_dev`
- **Description** (string UTF-8) - sua descrição da conexão.
- **ConnectionType** (string UTF-8) - obrigatório. O tipo de conexão. Atualmente, o JDBC é compatível, mas o SFTP não.
- **ConnectionProperties** (dict) - obrigatório. Uma lista de pares de chave-valor usados como parâmetros para essa conexão, incluindo o URL da conexão JDBC, o nome de usuário e a senha.

- `PhysicalConnectionRequirements` (dict) - requisitos de conexão física, que incluem o seguinte:
 - `SubnetId` (string UTF-8) - o ID da sub-rede usada pela conexão.
 - `SecurityGroupIdList` (lista) - a lista de IDs de grupos de segurança usada pela conexão.
 - `AvailabilityZone` (string UTF-8) - obrigatório. A zona de disponibilidade que contém o endpoint. Esse parâmetro está suspenso.

Etapa 3: criar um crawler do AWS Glue

Em seguida, você cria um crawler do AWS Glue para preencher o catálogo do AWS Glue. Para obter mais informações, consulte [Catalogar tabelas com um crawler](#) no Guia do desenvolvedor do AWS Glue.

A primeira etapa para adicionar um crawler é criar um novo banco de dados em um catálogo de dados usando a [API do SDK da AWS](#). Antes de começar, você deve excluir qualquer versão anterior dele usando a operação `delete_crawler`.

Quando você cria o crawler, algumas considerações se aplicam:

- Para o nome do crawler, use o formato `<redshift_node_name>_<redshift_database_name>_<redshift_schema_name>`, por exemplo: `abcde03_dev_oracle_glue`
- Use um perfil do IAM que já existe. Para obter mais informações sobre como criar um perfil do IAM, consulte [Criar perfis do IAM](#) no Guia do usuário do IAM.
- Use o nome do banco de dados que você criou nas etapas anteriores.
- Use o parâmetro `ConnectionName`, que é obrigatório.
- Para o parâmetro `path`, use o caminho para o destino de JDBC, por exemplo: `dev/oracle_glue/%`

O exemplo a seguir exclui um crawler existente e cria outro usando a API do AWS Glue em Python.

```
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'
)
```

Crie e execute um crawler que se conecta a um ou mais armazenamentos de dados, determina as estruturas de dados e grava tabelas no catálogo de dados. Você pode executar o crawler em uma programação, como mostrado a seguir.

```
response = client.start_crawler(
    Name='string'
)
```

Este exemplo usa o Amazon Redshift como destino. Após a execução do crawler, os tipos de dados do Amazon Redshift são mapeados aos tipos de dados do AWS Glue da seguinte maneira.

Tipo de dados do Amazon Redshift	Tipo de dados do 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
data	data
timestamp	timestamp
timestamptz	timestamp

Como converter scripts de ETL da Informatica com o AWS SCT

Você pode usar a interface de linha de comandos (CLI) do AWS SCT para converter seus scripts de ETL da Informatica. Assim, você pode usá-los com seu novo banco de dados de destino. Essa conversão inclui três etapas principais. Primeiro, o AWS SCT converte o código SQL incorporado em seus objetos da Informatica. Depois, o AWS SCT altera os nomes dos objetos do banco de dados de

acordo com as regras de migração que você especificou no projeto. Por fim, o AWS SCT redireciona as conexões dos seus scripts de ETL da Informatica para o novo banco de dados de destino.

Você pode converter scripts de ETL da Informatica como parte do projeto de conversão de banco de dados do AWS SCT. Certifique-se de adicionar seus bancos de dados de origem e destino ao projeto quando converter scripts de ETL da Informatica.

Para converter scripts de ETL da Informatica, certifique-se de usar a versão 1.0.667 ou superior do AWS SCT. Além disso, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte [AWS SCT Referência da CLI](#).

Para converter scripts de ETL da Informatica usando o AWS SCT

1. Crie um novo script de CLI da AWS SCT ou edite um modelo de cenário existente. Por exemplo, é possível baixar e editar o modelo `InformaticConversionTemplate.scts`. Para obter mais informações, consulte [Obter cenários de CLI](#).
2. Baixe os drivers JDBC necessários para os bancos de dados de origem e destino. Especifique a localização desses drivers usando o comando `SetGlobalSettings`. Além disso, especifique as pastas nas quais o AWS SCT pode salvar os arquivos de log.

O exemplo de código a seguir mostra como adicionar o caminho para os drivers Oracle e PostgreSQL às configurações do AWS SCT. Depois de executar esse exemplo de código, a AWS SCT armazena os arquivos de log na pasta `C:\sct_log`. Além disso, AWS SCT armazena os arquivos de log do console na pasta `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. Criar um novo projeto da AWS SCT. Insira o nome e a localização do projeto.

O exemplo de código a seguir cria o projeto da `oracle_postgresql` na pasta `C:\Temp`.


```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. Adicione informações de conexão sobre seus bancos de dados de origem e de destino.

O exemplo de código a seguir adiciona bancos de dados Oracle e PostgreSQL como fonte e destino para seu projeto do 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'
/
```

No exemplo anterior, substitua *source_user* e *target_user* pelo nomes dos usuários do banco de dados. Em seguida, substitua *source_password* e *target_password* por suas senhas. Para *source_address* e *target_address*, insira os endereços IP dos seus servidores de banco de dados de origem e destino.

Para se conectar a um banco de dados Oracle versão 19 e superior, use o nome do serviço Oracle no comando AddSource. Para fazer isso, adicione o parâmetro `-connectionType` e defina seu valor como `'basic_service_name'`. Em seguida, adicione o parâmetro `-servicename` e coloque como valor o nome do seu serviço Oracle. Para obter mais informações sobre o comando AddSource, consulte [Referência da CLI do AWS Schema Conversion Tool](#).

5. Crie uma nova regra de mapeamento do AWS SCT, que define os mecanismos de banco de dados de destino para cada esquema de banco de dados de origem. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).

O exemplo de código a seguir cria uma regra de mapeamento que inclui todos os esquemas de banco de dados Oracle de origem e define o PostgreSQL como um destino de migração.

```
AddServerMapping
  -sourceTreePath: 'Servers.ORACLE'
  -targetTreePath: 'Servers.POSTGRESQL'
/
```

6. Adicione informações de conexão sobre seus arquivos XML da Informatica de origem e destino.

O exemplo de código a seguir adiciona os arquivos XML da Informatica das pastas C:\Informatica_source e C:\Informatica_target.

```
AddSource
  -name: 'INFA_SOURCE'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
  -name: 'INFA_TARGET'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_target'
/
```

7. Crie outra regra de mapeamento para definir o arquivo XML da Informatica de destino para o arquivo de origem.

O exemplo de código a seguir cria uma regra de mapeamento que inclui arquivos XML da Informatica de origem e destino usados no exemplo anterior.

```
AddServerMapping
  -sourceTreePath: 'ETL.INFA_SOURCE'
  -targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Especifique a conexão do servidor de banco de dados que corresponde à referência do nome da conexão da Informatica.

O exemplo de código a seguir configura o redirecionamento dos scripts de ETL da Informatica da origem para o novo banco de dados de destino. Este exemplo também configura variáveis de conexão.

```
ConfigureInformaticaConnectionsRedirect
-treePath: 'ETL.INFA_SOURCE.Files'
-connections: '{
"ConnectionNames": [
{
"name": "Oracle_src",
"newName": "postgres",
"treePath": "Servers.ORACLE"
}
]
"ConnectionVariables": [
{
"name": "$Source",
"treePath": "Servers.ORACLE"
}
]
}'
/
```

9. Converta seus esquemas de banco de dados de origem e scripts de ETL da Informatica.

O exemplo de código a seguir converte todos os seus esquemas de banco de dados Oracle de origem e seu arquivo XML da Informatica.

```
Convert
-treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
-treePath: 'ETL.INFA_SOURCE.Files'
/
```

10. (Opcional) Salve seu projeto de conversão e o relatório de avaliação. Esse relatório inclui os itens da ação de conversão e recomendações sobre como lidar com cada um deles.

O exemplo de código a seguir salva seu projeto e uma cópia do relatório de avaliação em formato PDF na pasta C:\Temp.

```
SaveProject
/  
SaveReportPDF  
-treePath: 'ETL.INFA_SOURCE.Files'  
-file: 'C:\Temp\Informatica.pdf'  
/
```

11. Salve o arquivo XML da Informatica convertido.

O exemplo de código a seguir salva o arquivo XML convertido na pasta C:\Temp. Você especificou essa pasta na etapa anterior usando o comando AddTarget.

```
SaveTargetInformaticaXML  
-treePath: 'ETL.INFA_TARGET.Files'  
/
```

12. Salve seu script como um arquivo .scts e execute-o usando o comando RunSCTBatch CLI da AWS SCT. Para obter mais informações, consulte [AWS SCT Modo de script CLI](#).

O exemplo a seguir executa o script Informatica.scts na pasta C:\Temp. Você pode usar esse exemplo no Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Se você editar seus scripts de ETL da Informatica de origem, execute o script AWS SCT CLI novamente.

Como converter SSIS em AWS Glue com a AWS SCT

A seguir, você encontra como converter pacotes do Microsoft SQL Server Integration Services (SSIS) em AWS Glue usando o AWS SCT.

Para fazer isso, certifique-se de usar a versão 1.0.642 ou superior da AWS SCT. Você também precisa ter um projeto SSIS com pacotes ETL — arquivos .dtsx, .conmgr e .params na pasta local.

Não é necessário ter um servidor SSIS instalado. O processo de conversão passa pelos arquivos SSIS locais.

Para converter um pacote SSIS em AWS Glue usando o AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar origem no menu para adicionar um novo pacote SSIS de origem ao seu projeto.
3. Escolha o SQL Server Integration Services e conclua o seguinte:
 - Nome da conexão — Insira o nome da conexão. AWS SCT exibe esse nome na árvore de metadados.
 - Pasta de pacotes SSIS — Escolha o caminho para a pasta do projeto SSIS com pacotes.

AWS SCT lê os arquivos do projeto (com as extensões `.dtsx`, `.conmgr` ou `.params`) da pasta local e os analisa. Em seguida, ele os organiza em uma árvore de categorias do AWS SCT.

4. Escolha Adicionar destino no menu para adicionar a nova plataforma de destino que irá converter seus pacotes SSIS de origem.
5. Escolha AWS Glue e preencha o seguinte:
 - Nome da conexão — Insira o nome da conexão. AWS SCT exibe esse nome na árvore de metadados.
 - Copiar do perfil da AWS — Escolha o perfil a ser usado.
 - Chave de acesso da AWS — Insira sua chave de acesso da AWS.
 - Chave secreta da AWS – Insira sua chave secreta da AWS.
 - Região – Escolha a região da Região da AWS que você quer usar.
 - Pasta de bucket do Amazon S3 — Insira o caminho da pasta para o bucket do Amazon S3 que você planeja usar.

Você pode usar um destino virtual do AWS Glue. Nesse caso, você não precisa especificar as credenciais de conexão. Para obter mais informações, consulte [the section called “Alvos virtuais”](#).

6. Crie uma nova regra de mapeamento que inclua seu pacote SSIS de origem e seu destino do AWS Glue. Para obter mais informações, consulte [the section called “Nova regra”](#).
7. No menu Visualizar, escolha Visualização principal.

8. Na visualização em árvore do SSIS, abra o menu de contexto (clique com o botão direito do mouse) de Gerenciadores de conexão e escolha Configurar conexões.
9. Configure o gerenciador de conexões do projeto.

Para configurar um mapeamento de conexão para gerenciadores de conexão SSIS, especifique a conexão do AWS Glue para o gerenciador de conexões SSIS correspondente. As suas conexões do AWS Glue já devem ter sido criadas.

- a. Em Conexões, escolha Conexões do projeto.
 - b. Para Conexão com o catálogo do Glue, escolha a conexão apropriada AWS Glue.
10. Configure o gerenciador de conexões de pacotes:
 - a. Em Conexões, escolha seu pacote.
 - b. Para Conexão com o catálogo do Glue, escolha a conexão apropriada AWS Glue.
 - c. Repita essas ações em todas as conexões disponíveis para seu pacote.
 11. Escolha Aplicar.
 12. Converta seu pacote. Na visualização da árvore de origem, encontre Pacotes. Abra o menu de contexto (clique com o botão direito do mouse) do pacote e escolha Converter pacote.
 13. Salve o script convertido no Amazon S3. Na visualização da árvore de destino, encontre os Scripts de pacote. Abra o menu de contexto (clique com o botão direito do mouse) do script convertido e escolha Salvar no S3.
 14. Configure sua tarefa do AWS Glue. Na visualização da árvore de destino, encontre os Scripts de pacote. Abra o menu de contexto (clique com o botão direito do mouse) do script convertido e escolha Configurar tarefa do AWS Glue.
 15. Preencha as três seções de configuração a seguir.
 - a. Seção Fluxo de dados do projeto:
 - Estratégia de execução — Escolha como sua tarefa executará scripts de ETL. Escolha SEQUENCIAL para executar os scripts na ordem especificada no assistente. Escolha PARALELO para executar os scripts em paralelo, desconsiderando a ordem especificada no assistente.
 - Scripts — Escolha o nome do script convertido.
 - Selecione Próximo.
 - b. Seção Propriedades da tarefa:

- Nome — Insira o nome da sua tarefa do AWS Glue.
- Perfil do IAM — Especifique o perfil do IAM usado para a autorização de recursos necessários para a execução da tarefa e acesso aos armazenamentos de dados.
- Nome do arquivo de script — Insira o nome do script convertido.
- Caminho do arquivo de script do S3 — Insira o caminho do Amazon S3 para seu script convertido.
- Criptografar script usando SSE-S3 — Escolha essa opção para proteger dados usando criptografia no lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3).
- Diretórios temporários — Insira o caminho do Amazon S3 para um diretório temporário para obter resultados intermediários. O AWS Glue e as transformações integradas do AWS Glue usam esse diretório para ler ou gravar no Amazon Redshift.
- A AWS SCT gera automaticamente o caminho para as bibliotecas Python. Você pode revisar esse caminho em Caminho gerado da biblioteca Python. Não é possível editar esse caminho gerado automaticamente. Para usar bibliotecas Python adicionais, insira o caminho em Caminho da biblioteca Python do usuário.
- Caminho da biblioteca Python – Insira os caminhos de outras bibliotecas Python do usuário. Separe os caminhos do Amazon S3 com vírgulas.
- Caminho de .jars dependentes – Insira os caminhos dos arquivos .jar dependentes. Separe os caminhos do Amazon S3 com vírgulas.
- Caminho dos arquivos referenciados — Insira os caminhos para arquivos adicionais que são exigidos pelo seu script (por exemplo, arquivos de configuração). Separe os caminhos do Amazon S3 com vírgulas.
- Capacidade máxima – Insira o número máximo de unidades de processamento de dados (DPUs) do AWS Glue que poderão ser alocadas quando essa tarefa for executada. Você pode inserir qualquer número inteiro de 2 a 100. O padrão é 2.
- Simultaneidade máxima – Insira o número máximo de execuções simultâneas permitidas para a tarefa. O padrão é 1. O AWS Glue retorna um erro quando este limite é atingido.
- Limite de tempo da tarefa (minutos) — Insira o tempo limite da sua tarefa de ETL, como uma proteção contra tarefas descontroladas. O padrão é 2.880 minutos (48 horas) para tarefas em lotes. Se a tarefa exceder esse limite, seu estado da execução é alterado para TIMEOUT.
- Limite de notificação de atraso (minutos) — Insira o limite em minutos antes da AWS SCT enviar uma notificação de atraso.

- Número de novas tentativas — Insira o número de vezes (0 a 10) que o AWS Glue deve reiniciar automaticamente a tarefa se ela falhar. As tarefas que atingem o limite de tempo não são reiniciadas. O padrão é 0.
 - Selecione Próximo.
- c. Configure as conexões necessárias:
- i. Em Todas as conexões, escolha as conexões do AWS Glue necessárias e adicione-as à lista de conexões selecionadas.
 - ii. Escolha Terminar.
16. Crie uma tarefa configurada do AWS Glue. Na visualização da árvore de destino, encontre e expanda Tarefas de ETL. Abra o menu de contexto (clique com o botão direito do mouse) da tarefa de ETL que você configurou e escolha Criar tarefa configurada do AWS Glue.
17. Execute a tarefa do AWS Glue:
- a. Abra o console do AWS Glue em <https://console.aws.amazon.com/glue/>.
 - b. No painel de navegação, escolha Tarefas.
 - c. Escolha Adicionar tarefa e, em seguida, escolha a tarefa que você deseja executar.
 - d. Na guia Ações, selecione Executar.

Componentes do SSIS que a AWS SCT pode converter em AWS Glue

Você pode usar a AWS SCT para converter componentes de fluxo de dados e de controle, bem como contêineres, parâmetros e variáveis.

Os componentes de fluxo de dados compatíveis incluem os seguintes:

- Destino ADO NET
- Origem ADO NET
- Aggregate
- Transformação de cache
- Transformação do mapa de caracteres
- Transformação de divisão condicional
- Transformação da coluna de cópia
- Transformação da conversão de dados

- Transformação de coluna derivada
- Destino do Excel
- Origem do Excel
- Transformação da exportação de coluna
- Destino do arquivo simples
- Origem do arquivo simples
- Transformação de Pesquisa Difusa
- Transformação da coluna de importação
- Transformação de Pesquisa
- Transformação de junção de mesclagem
- Transformação de mesclagem
- Transformação multicast
- Destino ODBC
- Origem ODBC
- Transformação de comando OLE DB
- Destino do OLE DB
- Origem do OLE DB
- Transformação de amostragem percentual
- Transformação de articulação
- Destino do arquivo bruto
- Origem do arquivo bruto
- Destino do RecordSet
- Transformação da contagem de linhas
- Transformação de amostragem de linhas
- Transformação de classificação
- Destino do SQL Server
- Transformação de Union All
- Transformação de desarticulação
- Origem de XML

Os componentes de controle de fluxo suportados incluem os seguintes:

- Tarefa de inserção em massa
- Tarefa de execução do pacote
- Tarefa de execução do SQL
- Tarefa de execução de instrução T-SQL
- Tarefa de expressão
- Tarefa de sistema de arquivos
- Tarefa de notificação do operador
- Tarefa de enviar e-mail

Os contêineres SSIS suportados incluem os seguintes:

- Contêiner For Loop
- Contêiner Foreach Loop
- Contêiner de sequência

Como converter SSIS em AWS Glue Studio com a AWS SCT

Você pode usar a AWS SCT para converter pacotes do Microsoft SQL Server Integration Services (SSIS) em AWS Glue Studio.

Um pacote SSIS inclui os componentes necessários, como gerenciador de conexões, tarefas, fluxo de controle, fluxo de dados, parâmetros, manipuladores de eventos e variáveis, para executar uma tarefa específica de extração, transformação e carregamento (ETL). A AWS SCT converte pacotes SSIS em um formato compatível com o AWS Glue Studio. Depois de migrar seu banco de dados de origem para a Nuvem Nuvem AWS, é possível executar essas tarefas AWS Glue Studio convertidas para realizar tarefas de ETL.

Para fazer isso, certifique-se de usar a versão 1.0.661 ou superior da AWS SCT.

Tópicos

- [Pré-requisitos](#)
- [Como adicionar pacotes SSIS ao seu projeto da AWS SCT](#)
- [Como converter pacotes SSIS em AWS Glue Studio com a AWS SCT](#)
- [Criação de tarefas do AWS Glue Studio usando o código convertido](#)

- [Como criar um relatório de avaliação para um pacote SSIS com a AWS SCT](#)
- [Componentes do SSIS que a AWS SCT pode converter em AWS Glue Studio](#)

Pré-requisitos

Nesta seção, veja quais são as tarefas de pré-requisito para a conversão de pacotes SSIS em AWS Glue. Essas tarefas incluem a criação dos recursos necessários da AWS na sua conta.

Você pode usar o AWS Identity and Access Management (IAM) para definir as políticas e os perfis necessários para acessar os recursos usados pelo AWS Glue Studio. Para obter mais informações, consulte [Permissões do IAM para o usuário do AWS Glue Studio](#).

Depois da AWS SCT converter seus scripts de origem em AWS Glue Studio, faça o upload deles para um bucket do Amazon S3. Certifique-se de criar esse bucket do Amazon S3 e selecioná-lo nas configurações do perfil de serviço da AWS. Para obter mais informações sobre como criar um bucket do Amazon S3, consulte [Crie seu primeiro bucket do S3](#) no Guia do usuário do Amazon Simple Storage Service.

Para garantir que o AWS Glue Studio possa se conectar ao seu armazenamento de dados, crie um conector personalizado e uma conexão. Também é necessário armazenar as credenciais do banco de dados no AWS Secrets Manager.

Para criar um conector personalizado

1. Faça download do driver JDBC para seu armazenamento de dados. Para obter mais informações sobre os drivers JDBC que a AWS SCT usa, consulte [Baixar os drivers de banco de dados necessários](#).
2. Faça o upload desse arquivo com drivers no seu bucket do Amazon S3. Para obter mais informações, consulte [Adicionar um objeto a um bucket](#) no Guia do Usuário do Amazon Simple Storage Service.
3. Faça login no AWS Management Console e abra o console do AWS Glue Studio em <https://console.aws.amazon.com/gluestudio/>.
4. Escolha Conectores e, em seguida, escolha Criar conector personalizado.
5. Em URL do Connector S3, selecione Explorar S3 e escolha o arquivo do driver JDBC que você carregou no seu bucket do Amazon S3.
6. Insira um nome que descreva seu conector. Por exemplo, digite **SQLServer**.
7. Em Tipo de conector, escolha JDBC.

8. Em Nome da classe, insira o nome da classe principal do seu driver JDBC. Para SQL Server, insira **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
 9. Para URL base do JDBC, insira o URL base do JDBC. A sintaxe desse URL depende do mecanismo de banco de dados de origem. Para servidor SQL, use o formato a seguir:
jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>.
- Substitua *<host>*, *<port>*, *<dbname>*, *<username>* e *<password>* pelos seus valores.
10. Para delimitador de parâmetros de URL, insira o ponto e vírgula (;).
 11. Escolha Criar conector.

Para armazenar as credenciais do banco de dados no AWS Secrets Manager

1. Faça login no AWS Management Console e abra o console do AWS Secrets Manager em <https://console.aws.amazon.com/secretsmanager/>.
2. Selecione Armazenar um novo segredo.
3. Na página Selecionar tipo de segredo, faça o seguinte:
 - a. Em Tipo de segredo, escolha Outro tipo de segredo.
 - b. Para Pares de chave/valor, insira as seguintes chaves: **host**, **port**, **dbname**, **username** e **password**.

Em seguida, insira seus valores para essas chaves.

4. Na página Configurar segredo, insira um Nome de segredo descritivo. Por exemplo, digite **SQL_Server_secret**.
5. Selecione Próximo. Depois, na página Configurar alternância escolha Próximo.
6. Na página Revisar, revise os detalhes do segredo e escolha Armazenar.

Como criar uma conexão para um conector

1. Faça login no AWS Management Console e abra o console do AWS Glue Studio em <https://console.aws.amazon.com/gluestudio/>.
2. Escolha o conector que receberá a conexão e selecione Criar conexão.
3. Na página Criar conexão, insira um nome descritivo para ela. Por exemplo, digite **SQL-Server-connection**.

4. Em Segredo da AWS, escolha o segredo que você criou em AWS Secrets Manager.
5. Configure Opções de rede e escolha Criar conexão.

Agora, é possível criar uma tarefa do AWS Glue Studio com um conector personalizado. Para obter mais informações, consulte [Criação de tarefas do AWS Glue Studio](#).

Como adicionar pacotes SSIS ao seu projeto da AWS SCT

Você pode adicionar vários pacotes SSIS a um único projeto AWS SCT.

Para adicionar pacotes SSIS ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar origem no menu e, em seguida, escolha SQL Server Integration Services.
3. Em Nome da conexão, insira um nome para os pacotes SSIS. A AWS SCT exibe esse nome na árvore do painel esquerdo.
4. Para a Pasta de pacotes SSIS, insira o caminho para a pasta com os pacotes SSIS de origem.
5. Escolha Adicionar destino no menu e, em seguida, escolha AWS Glue Studio.

Para se conectar ao AWS Glue Studio, a AWS SCT usa o seu perfil da AWS. Para obter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).

6. Crie uma regra de mapeamento que inclua seu pacote SSIS de origem e seu destino do AWS Glue Studio. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).
7. Crie conexões do AWS Glue Studio no console do AWS Glue Studio. Para obter mais informações, consulte [Como criar conexões para conectores](#).
8. Escolha Gerenciadores de conexão na árvore à esquerda, abra o menu de contexto (clique com o botão direito do mouse) e depois selecione Configurar conexões.

A AWS SCT exibe a janela Configurar conexões.

9. Para cada conexão SSIS de origem, escolha uma conexão do AWS Glue Studio.

Como converter pacotes SSIS em AWS Glue Studio com a AWS SCT

A seguir, descubra como converter pacotes SSIS em AWS Glue Studio usando a AWS SCT.

Para converter um pacote SSIS em AWS Glue Studio

1. Adicione seu pacote SSIS ao projeto da AWS SCT. Para obter mais informações, consulte [Como adicionar pacotes SSIS ao seu projeto da AWS SCT](#).
2. No painel esquerdo, expanda os nós ETL e SSIS.
3. Escolha Pacotes, abra o menu de contexto (clique com o botão direito do mouse) do pacote e escolha Converter pacote.

A AWS SCT converte seus pacotes SSIS selecionados em arquivos JSON. Eles representam um nó em um gráfico acíclico dirigido (DAG). Encontre seus arquivos convertidos no nó Pacotes DAGs na árvore à direita.

4. Escolha Pacotes DAGs, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar no Amazon S3.

Agora você pode usar esses scripts para criar tarefas no AWS Glue Studio.

Criação de tarefas do AWS Glue Studio usando o código convertido

Depois de converter seus pacotes SSIS de origem, você pode usar os arquivos JSON convertidos para criar tarefas do AWS Glue Studio.

Para criar uma tarefa no AWS Glue Studio

1. Escolha Pacotes DAGs na árvore à direita, abra o menu de contexto (clique com o botão direito do mouse) e escolha Configurar tarefa do AWS Glue Studio.
2. (Opcional) Aplique o pacote de extensão que emula as funções do SSIS no AWS Glue Studio.
3. A janela Configurar tarefa do AWS Glue Studio é aberta.

Preencha a seção Propriedades básicas da tarefa:

- Nome — Insira o nome da tarefa do AWS Glue Studio.
- Nome do arquivo de script — Insira o nome do script da tarefa.
- Parâmetros da tarefa - Adicione parâmetros e insira seus valores.

Selecione Próximo.

4. Preencha a seção Propriedades avançadas da tarefa:

- Perfil do IAM — Especifique o perfil do IAM usado para a autorização do AWS Glue Studio e para acessar armazenamentos de dados.
- Caminho do arquivo de script do S3 — Insira o caminho do Amazon S3 para seu script convertido.
- Diretórios temporários — Insira o caminho do Amazon S3 para um diretório temporário para obter resultados intermediários. O AWS Glue Studio usa esse diretório para ler ou gravar no Amazon Redshift.
- A AWS SCT gera automaticamente o caminho para as bibliotecas Python. Você pode revisar esse caminho em Caminho gerado da biblioteca Python. Não é possível editar esse caminho gerado automaticamente. Para usar bibliotecas Python adicionais, insira o caminho em Caminho da biblioteca Python do usuário.
- Caminho da biblioteca Python – Insira os caminhos de outras bibliotecas Python do usuário. Separe os caminhos do Amazon S3 com vírgulas.
- Caminho de .jars dependentes – Insira os caminhos dos arquivos *.jar dependentes. Separe os caminhos do Amazon S3 com vírgulas.
- Caminho dos arquivos referenciados — Insira os caminhos para arquivos adicionais que são exigidos pelo seu script (por exemplo, arquivos de configuração). Separe os caminhos do Amazon S3 com vírgulas.
- Tipo de operador — Escolha G.1X ou G.2X.

Quando você escolhe G.1X, cada operador é mapeado para 1 DPU (4 vCPU, 16 GB de memória e disco de 64 GB).

Quando você escolhe G.2X, cada operador é mapeado para 2 DPU (8 vCPU, 32 GB de memória e disco de 128 GB).

- Número de operadores solicitado — Insira o número de operadores alocados quando a tarefa é executada.
- Simultaneidade máxima – Insira o número máximo de execuções simultâneas permitidas para a tarefa. O padrão é 1. O AWS Glue retorna um erro quando este limite é atingido.
- Limite de tempo da tarefa (minutos) — Insira o tempo limite da sua tarefa de ETL, como uma proteção contra tarefas descontroladas. O padrão é 2.880 minutos (48 horas) para tarefas em lotes. Se a tarefa exceder esse limite, seu estado da execução é alterado para TIMEOUT.
- Limite de notificação de atraso (minutos) — Insira o limite em minutos antes da AWS SCT enviar uma notificação de atraso.

- Número de novas tentativas — Insira o número de vezes (0 a 10) que o AWS Glue deve reiniciar automaticamente a tarefa se ela falhar. As tarefas que atingem o limite de tempo não são reiniciadas. O padrão é 0.

Escolha Terminar.

A AWS SCT configura as tarefas do AWS Glue Studio selecionadas.

5. Encontre suas tarefas configuradas em Tarefas de ETL na árvore à direita. Escolha a tarefa, configurada, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar tarefa do AWS Glue Studio.
6. Escolha Aplicar status e certifique-se de que o valor do Status da tarefa seja Êxito.
7. Abra o console do AWS Glue Studio, escolha Atualizar e selecione a tarefa. Em seguida, escolha Executar.

Como criar um relatório de avaliação para um pacote SSIS com a AWS SCT

O Relatório de avaliação da migração de ETL fornece informações sobre como converter seus pacotes SSIS em um formato compatível com o AWS Glue Studio. O relatório de avaliação inclui itens de ação para os componentes de seus pacotes SSIS. Esses itens de ação mostram quais componentes da AWS SCT não podem ser convertidos automaticamente.

Para criar um relatório de avaliação de migração de ETL

1. Expanda o nó SSIS embaixo de ETL, no painel esquerdo.
2. Escolha Pacotes, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar relatório.
3. Visualize a guia Resumo. Aqui, a AWS SCT exibe as informações do resumo executivo do relatório de avaliação da migração de ETL. Ele inclui resultados de conversão para todos os componentes de seus pacotes SSIS.
4. (Opcional) Salve uma cópia local do relatório de avaliação de migração ETL como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo CSV.

A AWS SCT cria três arquivos CSV. Eles contêm itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter os scripts.

5. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para AWS Glue Studio. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu pacote SSIS de origem ao qual o item de ação se aplica.

Componentes do SSIS que a AWS SCT pode converter em AWS Glue Studio

Você pode usar a AWS SCT para converter componentes e parâmetros do fluxo de dados SSIS em AWS Glue Studio.

Os componentes de fluxo de dados compatíveis incluem os seguintes:

- Destino ADO NET
- Origem ADO NET
- Aggregate
- Mapa de caracteres
- Divisão condicional
- Coluna Copiar
- Conversão de dados
- Coluna derivada
- Busca
- Mesclar
- Junção de mesclagem
- Multicast
- Destino ODBC
- Fonte ODBC

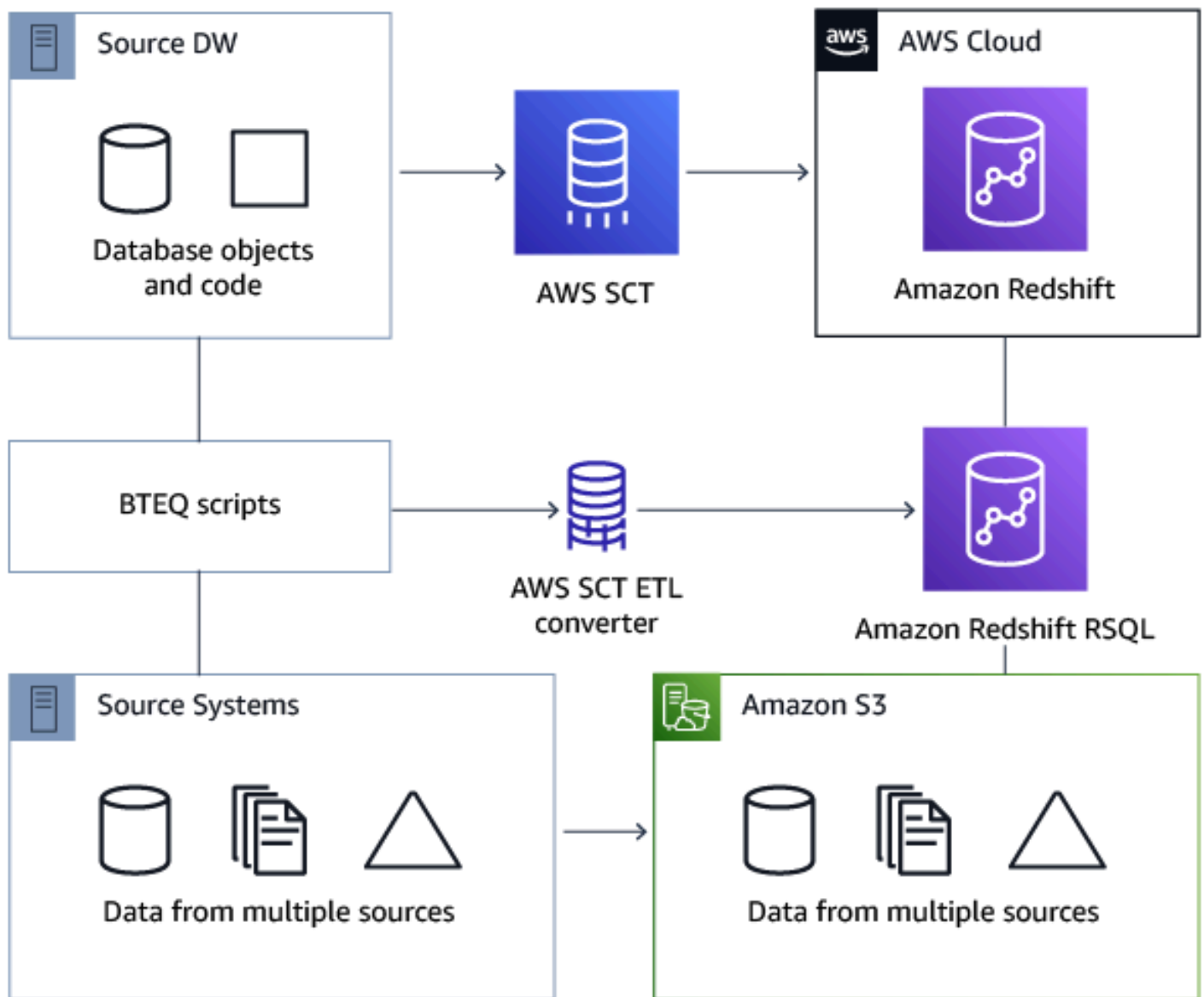
- Destino OLEDB
- Fonte OLEDB
- Número de linhas
- Classificar
- Destino do SQL Server
- Union All

A AWS SCT pode converter mais componentes SSIS em AWS Glue. Para obter mais informações, consulte [Componentes do SSIS que a AWS SCT pode converter em AWS Glue](#).

Como converter scripts Teradata BTEQ em Amazon Redshift RSQL com a AWS SCT

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter scripts de ETL do Teradata Basic Teradata Query (BTEQ) em Amazon Redshift RSQL.

O diagrama de arquitetura a seguir mostra um projeto de migração de banco de dados que inclui a conversão de scripts de extração, transformação e carregamento (ETL) em Amazon Redshift RSQL.



Tópicos

- [Adicionando scripts BTEQ ao seu projeto da AWS SCT](#)
- [Como configurar variáveis de substituição em scripts BTEQ com a AWS SCT](#)
- [Como converter scripts Teradata BTEQ em Amazon Redshift RSQL com a AWS SCT](#)
- [Como gerenciar scripts BTEQ com a AWS SCT](#)
- [Como criar um relatório de avaliação de conversão de script BTEQ com a AWS SCT](#)
- [Como editar e salvar seus scripts BTEQ convertidos com a AWS SCT](#)

Adicionando scripts BTEQ ao seu projeto da AWS SCT

É possível adicionar vários scripts a um único projeto da AWS SCT.

Para adicionar um script BTEQ ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar fonte no menu e, em seguida, escolha Teradata para adicionar seu banco de dados de origem ao projeto. Para obter mais informações, consulte [Como usar o Teradata como origem](#).
3. Escolha Adicionar destino no menu para adicionar um banco de dados de destino do Amazon Redshift ao seu projeto da AWS SCT.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Crie uma nova regra de mapeamento que inclua seu banco de dados Teradata de origem e o de destino do Amazon Redshift. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
5. No menu Visualizar, escolha Visualização principal.
6. No painel à esquerda, expanda o nó Scripts.
7. Selecione scripts BTEQ, abra o menu de contexto (clique com o botão direito do mouse) e selecione Carregar scripts.
8. Insira a localização do código-fonte para seus scripts Teradata BTEQ e escolha Selecionar pasta.

A AWS SCT exibe a janela Carregar scripts.

9. Faça um dos seguintes procedimentos:
 - a. Se os scripts do Teradata BTEQ não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e, em seguida, escolha OK para adicionar scripts ao seu projeto da AWS SCT.
 - b. Se seus scripts Teradata BTEQ incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts BTEQ](#).

Como configurar variáveis de substituição em scripts BTEQ com a AWS SCT

Seus scripts Teradata BTEQ podem incluir variáveis de substituição. Por exemplo, você pode usar um script BTEQ com variáveis de substituição para executar o mesmo conjunto de comandos em vários ambientes de banco de dados. Você pode usar a AWS SCT para configurar variáveis de substituição em seus scripts BTEQ.

Antes de executar um script BTEQ com variáveis de substituição, certifique-se de atribuir valores para todas as variáveis. Para fazer isso, você pode usar outras ferramentas ou aplicativos, como um script Bash, UC4 (Automic) e assim por diante. A AWS SCT só pode resolver e converter variáveis de substituição depois que esses valores forem atribuídos.

Para configurar variáveis de substituição em seus scripts BTEQ

1. Adicione os scripts BTEQ ao seu projeto da AWS SCT. Para obter mais informações, consulte [Adicionando scripts BTEQ ao seu projeto da AWS SCT](#).

Ao adicionar seus scripts, escolha Variáveis de substituição são usadas.

2. Em Definir formato da variável, insira uma expressão regular que corresponda a todas as variáveis de substituição no script.

Por exemplo, se os nomes das variáveis de substituição começarem com `{` e terminarem com `}`, use a expressão regular `\${\w+}`. Para combinar variáveis de substituição que começam com um cifrão ou um sinal de porcentagem, use a expressão regular `\$\w+|\%\w+`.

As expressões regulares na AWS SCT estão em conformidade com a sintaxe de expressão regular Java. Para obter mais informações, consulte [java.util.regex Class Pattern](#) na documentação do Apache Hive.

3. Escolha OK para carregar scripts no seu projeto da AWS SCT e, em seguida, escolha OK para fechar a janela Carregar scripts.
4. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.
5. Em Valor, insira o valor da variável de substituição.

Como converter scripts Teradata BTEQ em Amazon Redshift RSQL com a AWS SCT

A seguir, descubra como converter scripts de ETL BTEQ em Amazon Redshift RSQL usando a AWS SCT.

Como converter um script Teradata BTEQ em Amazon Redshift RSQL

1. Adicione os scripts BTEQ ao seu projeto da AWS SCT. Para obter mais informações, consulte [Adicionando scripts BTEQ ao seu projeto da AWS SCT](#).
2. Configure as variáveis de substituição. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts BTEQ](#).
3. No painel à esquerda, expanda o nó Scripts.
4. Faça um dos seguintes procedimentos:
 - Para converter um único script BTEQ, expanda o nó Scripts BTEQ, escolha o script a ser convertido e selecione Converter em RSQL no menu de contexto (clique com o botão direito do mouse).
 - Para converter vários scripts, certifique-se de selecionar todos eles. Escolha scripts BTEQ, abra o menu de contexto (clique com o botão direito do mouse) e depois selecione Converter em RSQL em Converter script.

A AWS SCT converte todos os scripts Teradata BTEQ selecionados em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.

5. Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte [Como editar e salvar seus scripts BTEQ convertidos](#).

Como gerenciar scripts BTEQ com a AWS SCT

Você pode adicionar ou remover vários scripts BTEQ do seu projeto da AWS SCT.

Para adicionar um script BTEQ adicional ao seu projeto da AWS SCT

1. No painel à esquerda, expanda o nó Scripts.
2. Escolha o nó Scripts BTEQ e abra o menu de contexto (clique com o botão direito).

3. Escolha Carregar scripts.
4. Insira as informações necessárias para adicionar um novo script BTEQ e configurar as variáveis de substituição. Para obter mais informações, consulte [Adicionando scripts BTEQ ao seu projeto da AWS SCT](#) e [Como configurar variáveis de substituição em scripts BTEQ](#).

Para remover um script BTEQ do seu projeto da AWS SCT

1. Expanda o nó scripts BTEQ em Scripts, no painel esquerdo.
2. Selecione o script a ser removido e abra o menu de contexto (clique com o botão direito).
3. Escolha Excluir script.

Como criar um relatório de avaliação de conversão de script BTEQ com a AWS SCT

Um relatório de avaliação de conversão de scripts BTEQ fornece informações sobre a conversão dos comandos BTEQ e instruções SQL de seus scripts BTEQ em um formato compatível com o Amazon Redshift RSQL. O relatório de avaliação inclui itens de ação para comandos BTEQ e instruções SQL que a AWS SCT não pode converter.

Como criar um relatório de avaliação de conversão de script BTEQ

1. Expanda o nó scripts BTEQ em Scripts, no painel esquerdo.
2. Selecione o script a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Escolha Conversão para RSQL em Criar relatório.
4. Visualize a guia Resumo. A guia Resumo exibe as informações do resumo executivo do relatório de avaliação dos scripts BTEQ. Ele inclui resultados de conversão para todos os comandos BTEQ e instruções SQL de seus scripts BTEQ.
5. (Opcional) Salve uma cópia local do relatório de avaliação de conversão do script BTEQ como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter os scripts.

6. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para o Amazon Redshift RSQL. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu script BTEQ de origem ao qual o item de ação se aplica.

Como editar e salvar seus scripts BTEQ convertidos com a AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu projeto da AWS SCT. A AWS SCT armazena o script editado como parte do projeto.

Para salvar seus scripts convertidos

1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
2. Escolha o script convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar script.
3. Insira o caminho para a pasta para salvar o script convertido e escolha Salvar.

A AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Como converter scripts de shell com comandos Teradata BTEQ incorporados em Amazon Redshift RSQL com a AWS SCT

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter scripts de shell com comandos Teradata Basic Teradata Query (BTEQ) incorporados em scripts de shell com comandos Amazon Redshift RSQL incorporados.

A AWS SCT extrai comandos Teradata BTEQ dos seus scripts de shell e os converte em um formato compatível com o Amazon Redshift. Depois de migrar o banco de dados Teradata para o Amazon Redshift, você pode usar esses scripts convertidos para gerenciar seu novo banco de dados do Amazon Redshift.

Você também pode usar a AWS SCT para converter arquivos com scripts de ETL Teradata BTEQ para Amazon Redshift RSQL. Para obter mais informações, consulte [Como converter scripts Teradata BTEQ em Amazon Redshift RSQL com a AWS SCT](#).

Tópicos

- [Adicionar scripts de shell com comandos Teradata BTEQ incorporados ao seu projeto da AWS SCT](#)
- [Como configurar variáveis de substituição em scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT](#)
- [Como converter scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT](#)
- [Como gerenciar scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT](#)
- [Como criar um relatório de avaliação de conversão de script de shell com a AWS SCT](#)
- [Como editar e salvar seus scripts de shell convertidos com a AWS SCT](#)

Adicionar scripts de shell com comandos Teradata BTEQ incorporados ao seu projeto da AWS SCT

É possível adicionar vários scripts a um único projeto da AWS SCT.

Para adicionar um script de shell ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar fonte no menu e, em seguida, escolha Teradata para adicionar seu banco de dados de origem ao projeto. Para obter mais informações, consulte [Como usar o Teradata como origem](#).
3. Escolha Adicionar destino no menu para adicionar um banco de dados de destino do Amazon Redshift ao seu projeto da AWS SCT.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Crie uma nova regra de mapeamento que inclua seu banco de dados Teradata de origem e o de destino do Amazon Redshift. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
5. No menu Visualizar, escolha Visualização principal.

6. No painel à esquerda, expanda o nó Scripts.
7. Selecione Shell, abra o menu de contexto (clique com o botão direito do mouse) e selecione Carregar scripts.
8. Insira a localização dos scripts de shell com comandos Teradata BTEQ incorporados e escolha Selecionar pasta.

A AWS SCT exibe a janela Carregar scripts.

9. Faça um dos seguintes procedimentos:
 - Se os scripts de shell não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e, em seguida, escolha OK para adicionar scripts ao seu projeto da AWS SCT.
 - Se seus scripts de shell incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de shell](#).

Como configurar variáveis de substituição em scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT

Seus scripts de shell podem incluir variáveis de substituição. Por exemplo, você pode usar um único script com variáveis de substituição para gerenciar bancos de dados em ambientes diferentes. Você pode usar a AWS SCT para configurar variáveis de substituição em seus scripts de shell.

Antes de executar comandos BTEQ com variáveis de substituição a partir de um script de shell, certifique-se de atribuir valores para todas as variáveis dentro desse script de shell. A AWS SCT só pode resolver e converter variáveis de substituição com valores atribuídos.

Para configurar variáveis de substituição no seu script de shell

1. Adicione os scripts de shell de origem ao seu projeto da AWS SCT. Para obter mais informações, consulte [Adicionando scripts de shell ao seu projeto da AWS SCT](#).

Ao adicionar seus scripts, escolha Variáveis de substituição são usadas.

2. Em Definir formato da variável, insira uma expressão regular que corresponda a todas as variáveis de substituição no script.

Por exemplo, se os nomes das variáveis de substituição começarem com `{` e terminarem com `}`, use a expressão regular `\${\w+\}`. Para combinar variáveis de substituição que começam com um cifrão ou um sinal de porcentagem, use a expressão regular `\$\w+|\%\w+`.

As expressões regulares na AWS SCT estão em conformidade com a sintaxe de expressão regular Java. Para obter mais informações, consulte [java.util.regex Class Pattern](#) na documentação do Apache Hive.

3. Escolha OK para carregar scripts no seu projeto da AWS SCT e, em seguida, escolha OK para fechar a janela Carregar scripts.
4. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.
5. Em Valor, insira o valor da variável de substituição.

Como converter scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT

A seguir, saiba como converter scripts de shell com comandos Teradata BTEQ incorporados em scrips de shell com comandos Amazon Redshift RSQL incorporados usando a AWS SCT.

Para converter um script de shell

1. Adicione os scripts de shell ao seu projeto da AWS SCT. Para obter mais informações, consulte [Adicionando scripts de shell ao seu projeto da AWS SCT](#).
2. Configure as variáveis de substituição. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de shell](#).
3. No painel à esquerda, expanda o nó Scripts.
4. Faça um dos seguintes procedimentos:
 - Para converter comandos BTEQ de um único script de shell, expanda o nó Shell, escolha o script a ser convertido e selecione Converter script no menu de contexto (clique com o botão direito do mouse).
 - Para converter vários scripts, certifique-se de selecionar todos eles. Selecione Shell, abra o menu de contexto (clique com o botão direito do mouse) e selecione Converter scripts.
5. Escolha OK.

A AWS SCT converte comandos BTEQ no seus scrips de shell selecionados em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.

6. Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte [Como editar e salvar seus scripts de shell convertidos](#).

Como gerenciar scripts de shell com comandos Teradata BTEQ incorporados usando a AWS SCT

Você pode adicionar ou remover vários scrips de shell do seu projeto da AWS SCT.

Para adicionar um novo script de shell ao seu projeto da AWS SCT

1. No painel à esquerda, expanda o nó Scripts.
2. Escolha o nó Shell e abra o menu de contexto (clique com o botão direito).
3. Escolha Carregar scripts.
4. Insira as informações necessárias para adicionar um novo script de shell e configurar as variáveis de substituição. Para obter mais informações, consulte [Adicionando scripts de shell ao seu projeto da AWS SCT](#) e [Como configurar variáveis de substituição em scripts de shell](#).

Para remover um script de shell do seu projeto da AWS SCT

1. Expanda o nó Shell, em Scripts, no painel esquerdo.
2. Selecione o script a ser removido e abra o menu de contexto (clique com o botão direito).
3. Escolha Excluir script.

Como criar um relatório de avaliação de conversão de script de shell com a AWS SCT

O relatório de avaliação de conversão do script de shell fornece informações sobre a conversão dos comandos BTEQ e das instruções SQL. A conversão é dos seus scripts de origem para um formato compatível com o Amazon Redshift RSQL. O relatório de avaliação inclui itens de ação para comandos BTEQ e instruções SQL que a AWS SCT não pode converter.

Como criar um relatório de avaliação de conversão de script de shell

1. Expanda o nó Shell, em Scripts, no painel esquerdo.
2. Selecione o script a ser convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar relatório.
3. Visualize a guia Resumo. A guia Resumo exibe as informações do resumo executivo do relatório de avaliação dos scripts de shell. Ele inclui resultados de conversão para todos os comandos BTEQ e instruções SQL de seus scripts de origem.

4. Salve uma cópia local do relatório de avaliação de conversão de script de shell como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Para salvar o relatório de avaliação de conversão de script de shell como um arquivo PDF, selecione Salvar em PDF no canto superior direito.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.
 - Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter os scripts.
5. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para o Amazon Redshift RSQL. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu script de shell de origem ao qual o item de ação se aplica.

Como editar e salvar seus scripts de shell convertidos com a AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu projeto da AWS SCT. A AWS SCT armazena o script editado como parte do projeto.

Para salvar seus scripts convertidos

1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
2. Escolha o script convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar script.
3. Insira o caminho para a pasta para salvar o script convertido e escolha Salvar.

A AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Como converter scripts de tarefa do Teradata FastExport em Amazon Redshift RSQL com a AWS SCT

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter scripts de tarefa do Teradata FastExport em Amazon Redshift RSQL.

Um script de tarefa do FastExport é um conjunto de comandos do FastExport e de instruções SQL que selecionam e exportam dados do banco de dados Teradata. A AWS SCT converte comandos do FastExport e instruções SQL em um formato compatível com o Amazon Redshift RSQL. Depois de migrar o banco de dados Teradata para o Amazon Redshift, você pode usar esses scripts convertidos para exportar dados do banco de dados do Amazon Redshift.

Tópicos

- [Como adicionar scripts de tarefa do FastExport ao seu projeto da AWS SCT](#)
- [Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata FastExport com a AWS SCT](#)
- [Como gerenciar scripts de tarefa do Teradata FastExport com a AWS SCT](#)
- [Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastExport com a AWS SCT](#)
- [Como editar e salvar seus scripts de tarefa do Teradata FastExport convertidos com a AWS SCT](#)

Como adicionar scripts de tarefa do FastExport ao seu projeto da AWS SCT

É possível adicionar vários scripts a um único projeto da AWS SCT.

Para adicionar scripts de tarefa do FastExport ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar fonte no menu e, em seguida, escolha Teradata para adicionar seu banco de dados de origem ao projeto. Para obter mais informações, consulte [Como usar o Teradata como origem](#).
3. Escolha Adicionar destino no menu para adicionar um banco de dados de destino do Amazon Redshift ao seu projeto da AWS SCT.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Crie uma nova regra de mapeamento que inclua seu banco de dados Teradata de origem e o de destino do Amazon Redshift. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
5. No menu Visualizar, escolha Visualização principal.

6. No painel à esquerda, expanda o nó Scripts.
7. Escolha FastExport, abra o menu de contexto (clique com o botão direito do mouse) e escolha Carregar scripts.
8. Insira a localização do código-fonte para seus scripts Teradata FastExport e escolha Seleccionar pasta.

A AWS SCT exibe a janela Carregar scripts.

9. Faça um dos seguintes procedimentos:
 - Se os scripts de tarefa do Teradata FastExport não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e, em seguida, escolha OK para adicionar scripts ao seu projeto da AWS SCT.
 - Se seus scripts de tarefa Teradata FastExport incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport](#).

Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport com a AWS SCT

Seus scripts de tarefa Teradata FastExport podem incluir variáveis de substituição. Por exemplo, você pode usar um único script com variáveis de substituição para exportar dados de vários bancos de dados. Você pode usar a AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Antes de executar um script de tarefa FastExport com variáveis de substituição, certifique-se de atribuir valores para todas as variáveis. Para fazer isso, você pode usar outras ferramentas ou aplicativos, como um script Bash, UC4 (Automic) e assim por diante. A AWS SCT só pode resolver e converter variáveis de substituição depois que esses valores forem atribuídos.

Para configurar variáveis de substituição em seus scripts de tarefa FastExport

1. Adicione seus scripts de tarefa do Teradata FastExport de origem ao seu projeto da AWS SCT. Para obter mais informações, consulte [Adicionando scripts BTEQ ao seu projeto da AWS SCT](#).

Ao adicionar seus scripts, escolha Variáveis de substituição são usadas.
2. Em Definir formato da variável, insira uma expressão regular que corresponda a todas as variáveis de substituição no script.

Por exemplo, se os nomes das variáveis de substituição começarem com `{` e terminarem com `}`, use a expressão regular `\${\w+}`. Para combinar variáveis de substituição que começam com um cifrão ou um sinal de porcentagem, use a expressão regular `\$\w+|\%\w+`.

As expressões regulares na AWS SCT estão em conformidade com a sintaxe de expressão regular Java. Para obter mais informações, consulte [java.util.regex Class Pattern](http://java.util.regex) na documentação do Apache Hive.

3. Escolha OK para carregar scripts no seu projeto da AWS SCT e, em seguida, escolha OK para fechar a janela Carregar scripts.
4. No painel à esquerda, expanda o nó Scripts. Escolha FastExport e, em seguida, escolha sua pasta com scripts. Abra o menu de contexto (clique com o botão direito do mouse) e escolha Exportar variáveis em Variáveis de substituição.
5. Exporte variáveis de substituição para um script. Expanda sua pasta com scripts, escolha seu script, abra o menu de contexto (clique com o botão direito do mouse) e escolha Exportar variáveis em Variáveis de substituição.
6. Insira o nome do arquivo com valores separados por vírgula (CSV) para salvar as variáveis de substituição e depois escolha Salvar.
7. Abra esse arquivo CSV e preencha os valores das variáveis de substituição.

Dependendo do sistema operacional, a AWS SCT usa formatos diferentes para arquivos CSV. Os valores no arquivo podem estar entre aspas ou não. Certifique-se de usar o mesmo formato para os valores das variáveis de substituição que os outros valores no arquivo. A AWS SCT não consegue importar um arquivo CSV com valores em formatos diferentes.

8. Salve o arquivo CSV.
9. No painel à esquerda, expanda o nó Scripts. Escolha FastExport e, em seguida, escolha o script. Abra o menu de contexto (clique com o botão direito do mouse) e escolha Importar variáveis em Variáveis de substituição.
10. Escolha seu arquivo CSV e clique em Abrir.
11. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.

Como converter scripts de tarefa do Teradata FastExport com a AWS SCT

A seguir, saiba como converter scripts de tarefa do Teradata FastExport em Amazon Redshift RSQL usando a AWS SCT.

Para converter scripts de tarefa Teradata FastExport em Amazon Redshift RSQL

1. Adicione scripts de tarefa do FastExport ao seu projeto da AWS SCT. Para obter mais informações, consulte [Como adicionar scripts de tarefa do FastExport ao seu projeto da AWS SCT](#).
2. Configure as variáveis de substituição. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport](#).
3. No painel à esquerda, expanda o nó Scripts.
4. Faça um dos seguintes procedimentos:
 - Para converter um único script de tarefa do FastExport, expanda o nó FastExport, escolha o script a ser convertido e selecione Converter script no menu de contexto (clique com o botão direito do mouse).
 - Para converter vários scripts, certifique-se de selecionar todos eles. Selecione FastExport, abra o menu de contexto (clique com o botão direito do mouse) e selecione Converter script.

A AWS SCT converte todos os scripts de tarefa Teradata FastExport selecionados em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.

5. Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte [Como editar e salvar seus scripts de tarefa do Teradata FastExport convertidos](#).

Como gerenciar scripts de tarefa do Teradata FastExport com a AWS SCT

Você pode adicionar ou remover vários scripts de tarefa do Teradata FastExport do seu projeto da AWS SCT.

Para adicionar um novo script de tarefa do FastExport ao seu projeto da AWS SCT

1. No painel à esquerda, expanda o nó Scripts.
2. Escolha o nó FastExport e abra o menu de contexto (clique com o botão direito).
3. Escolha Carregar scripts.
4. Insira as informações necessárias para adicionar um novo script de tarefa do FastExport e configurar as variáveis de substituição. Para obter mais informações, consulte [Como adicionar](#)

[scripts de tarefa do FastExport ao seu projeto da AWS SCT](#) e [Como configurar variáveis de substituição em scripts de tarefa Teradata FastExport](#).

Para remover um script de tarefa do FastExport do seu projeto da AWS SCT

1. Expanda o nó FastExport, em Scripts, no painel esquerdo.
2. Selecione o script a ser removido e abra o menu de contexto (clique com o botão direito).
3. Escolha Excluir script.

Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastExport com a AWS SCT

Um relatório de avaliação de conversão de script de tarefa do FastExport fornece informações sobre a conversão dos comandos FastExport e instruções SQL de seus scripts FastExport em um formato compatível com o Amazon Redshift RSQL. O relatório de avaliação inclui itens de ação para comandos FastExport e instruções SQL que a AWS SCT não pode converter.

Para criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastExport

1. Expanda o nó FastExport, em Scripts, no painel esquerdo.
2. Selecione o script a ser convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar relatório.
3. Visualize a guia Resumo. A guia Resumo exibe as informações do resumo executivo do relatório de avaliação do script de tarefa do FastExport. Ele inclui resultados de conversão para todos os comandos FastExport e instruções SQL de seus scripts de origem.
4. Você pode salvar uma cópia local do relatório de avaliação de conversão de scripts de tarefa do FastExport como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV).
 - a. Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.

- b. Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter os scripts.

5. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para o Amazon Redshift RSQL. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu scripts de tarefa do FastExport de origem ao qual o item de ação se aplica.

Como editar e salvar seus scripts de tarefa do Teradata FastExport convertidos com a AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu projeto da AWS SCT. A AWS SCT armazena o script editado como parte do projeto.

Para salvar seus scripts convertidos

1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
2. Escolha o script convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar script.
3. Insira o caminho para a pasta para salvar o script convertido e escolha Salvar.

A AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Como converter scripts de tarefa do Teradata FastLoad em Amazon Redshift RSQL com a AWS SCT

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter scripts de tarefa do Teradata FastLoad em Amazon Redshift RSQL.

Um script do Teradata FastLoad é um conjunto de comandos que usa várias sessões para carregar dados em uma tabela vazia em um banco de dados Teradata. O Teradata FastLoad processa uma série de comandos do Teradata FastLoad e de instruções do SQL. Os comandos do Teradata FastLoad fornecem controle de sessão e tratamento de dados das transferências de dados. As instruções SQL criam, mantêm e eliminam tabelas.

A AWS SCT converte comandos do Teradata FastLoad e instruções SQL em um formato compatível com o Amazon Redshift RSQL. Depois de migrar o banco de dados Teradata para o Amazon

Redshift, você pode usar esses scripts convertidos para carregar dados ao seu banco de dados do Amazon Redshift.

Tópicos

- [Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT](#)
- [Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata FastLoad com a AWS SCT](#)
- [Como gerenciar scripts de tarefa do Teradata FastLoad com a AWS SCT](#)
- [Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastLoad com a AWS SCT](#)
- [Como editar e salvar seus scripts de tarefa do Teradata FastLoad convertidos com a AWS SCT](#)

Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT

É possível adicionar vários scripts a um único projeto da AWS SCT.

Para adicionar um script de tarefa do FastLoad ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar fonte no menu e, em seguida, escolha Teradata para adicionar seu banco de dados de origem ao projeto. Para obter mais informações, consulte [Como usar o Teradata como origem](#).
3. Escolha Adicionar destino no menu e adicione um banco de dados de destino do Amazon Redshift ao seu projeto da AWS SCT.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Crie uma nova regra de mapeamento que inclua seu banco de dados Teradata de origem e o de destino do Amazon Redshift. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
5. No menu Visualizar, escolha Visualização principal.
6. No painel à esquerda, expanda o nó Scripts.

7. Escolha FastLoad, abra o menu de contexto (clique com o botão direito do mouse) e escolha Carregar scripts.
8. Insira a localização dos seus scripts de tarefa do Teradata FastLoad de origem e escolha Selecionar pasta.

A AWS SCT exibe a janela Carregar scripts.

9. Faça um dos seguintes procedimentos:
 - Se os scripts de tarefa do Teradata FastLoad não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e, em seguida, escolha OK para adicionar scripts ao seu projeto da AWS SCT.
 - Se seus scripts de tarefa Teradata FastLoad incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad](#).

Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad com a AWS SCT

Seus scripts de tarefa Teradata FastLoad podem incluir variáveis de substituição. Por exemplo, você pode usar um único script com variáveis de substituição para carregar dados para diferentes bancos de dados.

Antes de executar um script de tarefa FastLoad com variáveis de substituição, certifique-se de atribuir valores para todas as variáveis. Para fazer isso, você pode usar outras ferramentas ou aplicativos, como um script Bash, UC4 (Automic) e assim por diante.

A AWS SCT só pode resolver e converter variáveis de substituição depois que os valores forem atribuídos. Antes de iniciar a conversão dos scripts de tarefa do Teradata FastLoad de origem, certifique-se de atribuir valores para todas as variáveis de substituição. Você pode usar a AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Para configurar variáveis de substituição em seus scripts de tarefa FastLoad

1. Ao adicionar seus scripts de tarefa do Teradata FastLoad de origem ao seu projeto da AWS SCT, escolha Variáveis de substituição são usadas. Para obter mais informações sobre como adicionar esses scripts, consulte [Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT](#).

2. Em Definir formato da variável, insira uma expressão regular que corresponda a todas as variáveis de substituição no script.

Por exemplo, se os nomes das variáveis de substituição começarem com `{` e terminarem com `}`, use a expressão regular `\${\w+}`. Para combinar variáveis de substituição que começam com um cifrão ou um sinal de porcentagem, use a expressão regular `\$\w+|\%\w+`.

As expressões regulares na AWS SCT estão em conformidade com a sintaxe de expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Apache Hive.

3. Escolha OK para carregar scripts no seu projeto da AWS SCT e, em seguida, escolha OK para fechar a janela Carregar scripts.
4. No painel à esquerda, expanda o nó Scripts. Escolha FastLoad e, em seguida, escolha sua pasta com scripts. Abra o menu de contexto (clique com o botão direito do mouse) e escolha Exportar variáveis em Variáveis de substituição.

Além disso, você pode exportar variáveis de substituição para um script. Expand a sua pasta com scripts, escolha seu script, abra o menu de contexto (clique com o botão direito do mouse) e escolha Exportar variáveis em Variáveis de substituição.

5. Insira o nome do arquivo de valores separados por vírgula (CSV) para salvar as variáveis de substituição e escolha Salvar.
6. Abra esse arquivo CSV e preencha os valores das variáveis de substituição.

Dependendo do sistema operacional, a AWS SCT usa formatos diferentes para arquivos CSV. Os valores no arquivo podem estar entre aspas ou não. Certifique-se de usar o mesmo formato para os valores das variáveis de substituição que os outros valores no arquivo. A AWS SCT não consegue importar um arquivo CSV com valores em formatos diferentes.

7. Salve o arquivo CSV.
8. No painel à esquerda, expanda o nó Scripts. Escolha FastLoad e, em seguida, escolha seu script. Abra o menu de contexto (clique com o botão direito do mouse) e escolha Importar variáveis em Variáveis de substituição.
9. Escolha seu arquivo CSV e clique em Abrir.
10. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.

Como converter scripts de tarefa do Teradata FastLoad com a AWS SCT

A seguir, saiba como converter uma tarefa do Teradata FastLoad em Amazon Redshift RSQL usando a AWS SCT.

Para converter scripts de tarefa Teradata FastLoad em Amazon Redshift RSQL

1. Adicione scripts de tarefa do FastLoad ao seu projeto da AWS SCT. Para obter mais informações, consulte [Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT](#).
2. Configure as variáveis de substituição. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad](#).
3. No painel à esquerda, expanda o nó Scripts.
4. Faça um dos seguintes procedimentos:
 - Para converter um único script de tarefa do FastLoad, expanda o nó FastLoad, escolha o script a ser convertido e selecione Converter script no menu de contexto (clique com o botão direito do mouse).
 - Para converter vários scripts, certifique-se de selecionar todos eles. Selecione FastLoad, abra o menu de contexto (clique com o botão direito do mouse) e selecione Converter script. Depois, siga um destes procedimentos:
 - Se você armazenar seu arquivo de dados de origem no Amazon S3, escolha o caminho do objeto S3 para Localização do arquivo de dados de origem.

Insira valores para a Pasta de bucket do Amazon S3 e Bucket do Amazon S3 como arquivo de manifesto para seu arquivo de dados de origem.
 - Se você armazenar seu arquivo de dados de origem no Amazon S3, escolha o Endereço do host para Localização do arquivo de dados de origem.

Insira valores para URL ou endereço IP do host, Login do usuário do host e Bucket do Amazon S3 como arquivo de manifesto do seu arquivo de dados de origem.
5. Escolha OK.

A AWS SCT converte todos os scripts de tarefa Teradata FastLoad selecionados em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.

6. Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte [Como editar e salvar seus scripts de tarefa do Teradata FastLoad convertidos](#).

Como gerenciar scripts de tarefa do Teradata FastLoad com a AWS SCT

Você pode adicionar ou remover vários scripts de tarefa do Teradata FastLoad do seu projeto da AWS SCT.

Para adicionar um novo script de tarefa do FastLoad ao seu projeto da AWS SCT

1. No painel à esquerda, expanda o nó Scripts.
2. Escolha o nó FastLoad e abra o menu de contexto (clique com o botão direito).
3. Escolha Carregar scripts.
4. Insira as informações necessárias para adicionar um novo script de tarefa do FastLoad e configurar as variáveis de substituição. Para obter mais informações, consulte [Como adicionar scripts de tarefa do FastLoad ao seu projeto da AWS SCT](#) e [Como configurar variáveis de substituição em scripts de tarefa Teradata FastLoad](#).

Para remover um script de tarefa do FastLoad do seu projeto da AWS SCT

1. Expanda o nó FastLoad, em Scripts, no painel esquerdo.
2. Selecione o script a ser removido e abra o menu de contexto (clique com o botão direito).
3. Escolha Excluir script.

Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastLoad com a AWS SCT

O relatório de avaliação de conversão do script de tarefa do FastLoad fornece informações sobre a conversão dos comandos BTEQ e das instruções SQL. A conversão é dos seus scripts de origem para um formato compatível com o Amazon Redshift RSQL. O relatório de avaliação inclui itens de ação para comandos FastLoad e instruções SQL que a AWS SCT não pode converter.

Para criar um relatório de avaliação de conversão para um script de tarefa do Teradata FastLoad

1. Expanda o nó FastLoad, em Scripts, no painel esquerdo.

2. Selecione o script a ser convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar relatório.
3. Visualize a guia Resumo.

A guia Resumo exibe as informações do resumo executivo do relatório de avaliação do script de tarefa do FastLoad. Ele inclui resultados de conversão para todos os comandos do FastLoad e instruções SQL de seus scripts de origem.

4. (Opcional) Salve uma cópia local do relatório de avaliação de conversão do script de tarefa FastLoad como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter os scripts.

5. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para o Amazon Redshift RSQL. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu script de tarefa do FastLoad de origem ao qual o item de ação se aplica.

Como editar e salvar seus scripts de tarefa do Teradata FastLoad convertidos com a AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu projeto da AWS SCT. A AWS SCT armazena o script editado como parte do projeto.

Para salvar seus scripts convertidos

1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
2. Escolha o script convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar script.
3. Insira o caminho para a pasta para salvar o script convertido e escolha Salvar.

A AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Como converter scripts de tarefa do Teradata MultiLoad em Amazon Redshift RSQL com a AWS SCT

Você pode usar a AWS SCT para converter scripts de tarefa do Teradata MultiLoad em Amazon Redshift RSQL.

Um script de tarefa do Teradata MultiLoad é um conjunto de comandos para manutenção em lote do seu banco de dados Teradata. Uma tarefa de importação do Teradata MultiLoad executa várias operações de inserção, atualização e exclusão em até cinco tabelas e visualizações diferentes. As tarefas de exclusão do Teradata MultiLoad podem remover um grande número de linhas de uma única tabela.

A AWS SCT converte comandos do Teradata MultiLoad e instruções SQL em um formato compatível com o Amazon Redshift RSQL. Depois de migrar o banco de dados Teradata para o Amazon Redshift, use esses scripts convertidos para gerenciar dados no seu banco de dados do Amazon Redshift.

Tópicos

- [Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT](#)
- [Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad com a AWS SCT](#)
- [Como converter scripts de tarefa do Teradata MultiLoad com a AWS SCT](#)
- [Como gerenciar scripts de tarefa do Teradata MultiLoad com a AWS SCT](#)
- [Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata MultiLoad com a AWS SCT](#)
- [Como editar e salvar seus scripts de tarefa do Teradata MultiLoad convertidos com a AWS SCT](#)

Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT

É possível adicionar vários scripts a um único projeto da AWS SCT.

Para adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT

1. Crie um novo projeto na AWS SCT ou abra um projeto existente. Para obter mais informações, consulte [the section called “Criando um projeto”](#).
2. Escolha Adicionar fonte no menu e, em seguida, escolha Teradata para adicionar seu banco de dados de origem ao projeto. Para obter mais informações, consulte [Como usar o Teradata como origem](#).
3. Escolha Adicionar destino no menu para adicionar um banco de dados de destino do Amazon Redshift ao seu projeto da AWS SCT.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte [Usar alvos virtuais](#).

4. Crie uma nova regra de mapeamento que inclua seu banco de dados Teradata de origem e o de destino do Amazon Redshift. Para obter mais informações, consulte [Adicionar uma nova regra de mapeamento](#).
5. No menu Visualizar, escolha Visualização principal.
6. No painel à esquerda, expanda o nó Scripts.
7. Escolha MultiLoad, abra o menu de contexto (clique com o botão direito do mouse) e escolha Carregar scripts.
8. Insira a localização dos seus scripts de tarefa do Teradata MultiLoad de origem e escolha Selecionar pasta.

A AWS SCT exibe a janela Carregar scripts.

9. Faça um dos seguintes procedimentos:
 - Se seus scripts de tarefa do Teradata MultiLoad não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e, em seguida, escolha OK para adicionar scripts ao seu projeto. AWS SCT
 - Se seus scripts de tarefa Teradata MultiLoad incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad](#).

Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad com a AWS SCT

Seus scripts de tarefa Teradata MultiLoad podem incluir variáveis de substituição. Por exemplo, você pode usar um único script com variáveis de substituição para carregar dados para diferentes bancos de dados.

Antes de executar um script de tarefa do MultiLoad com variáveis de substituição, certifique-se de atribuir os valores para todas as variáveis. Para fazer isso, você pode usar outras ferramentas ou aplicativos, como um script Bash, UC4 (Automic) e assim por diante.

A AWS SCT só pode resolver e converter variáveis de substituição depois que os valores forem atribuídos. Antes de iniciar a conversão dos scripts de tarefa do Teradata MultiLoad de origem, certifique-se de ter atribuído valores para todas as variáveis de substituição. Você pode usar a AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Para configurar variáveis de substituição em seus scripts de tarefa do MultiLoad

1. Ao adicionar seus scripts de tarefa do Teradata MultiLoad de origem ao seu projeto da AWS SCT, escolha Variáveis de substituição são usadas. Para obter mais informações sobre como adicionar esses scripts, consulte [Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT](#).
2. Em Definir formato da variável, insira uma expressão regular que corresponda a todas as variáveis de substituição no script.

Por exemplo, se os nomes das variáveis de substituição começarem com `{` e terminarem com `}`, use a expressão regular `\${\w+\}`. Para combinar variáveis de substituição que começam com um cifrão ou um sinal de porcentagem, use a expressão regular `\$\w+|\%\w+`.

As expressões regulares na AWS SCT estão em conformidade com a sintaxe de expressão regular Java. Para obter mais informações, consulte [java.util.regex Class Pattern](#) na documentação do Apache Hive.

3. Escolha OK para carregar scripts no seu projeto da AWS SCT e, em seguida, escolha OK para fechar a janela Carregar scripts.
4. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.
5. Em Valor, insira o valor da variável de substituição.

Como converter scripts de tarefa do Teradata MultiLoad com a AWS SCT

A seguir, saiba como converter uma tarefa do Teradata MultiLoad em Amazon Redshift RSQL usando a AWS SCT.

Para converter scripts de tarefa do Teradata MultiLoad em Amazon Redshift RSQL

1. Adicione scripts de tarefa do MultiLoad ao seu projeto da AWS SCT. Para obter mais informações, consulte [Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT](#).
2. Configure as variáveis de substituição e insira seus valores. Para obter mais informações, consulte [Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad](#).
3. No painel à esquerda, expanda o nó Scripts.
4. Faça um dos seguintes procedimentos:
 - Para converter um único script de tarefa MultiLoad, expanda o nó MultiLoad, escolha o script a ser convertido e escolha Converter script no menu de contexto (clique com o botão direito do mouse).
 - Para converter vários scripts, certifique-se de selecionar todos eles. Escolha MultiLoad, abra o menu de contexto (clique com o botão direito do mouse) e escolha Converter script.
5. Faça um dos seguintes procedimentos:
 - Se você armazenar seu arquivo de dados de origem no Amazon S3, escolha o caminho do objeto S3 para Localização do arquivo de dados de origem.

Insira a pasta de bucket do Amazon S3 e o Bucket do Amazon S3 como arquivo de manifesto para seu arquivo de dados de origem.
 - Se você armazenar seu arquivo de dados de origem no Amazon S3, escolha o Endereço do host para Localização do arquivo de dados de origem.

Insira o URL ou endereço IP do host, Login do usuário do host e Bucket do Amazon S3 como arquivo de manifesto do seu arquivo de dados de origem.
6. Escolha OK.

A AWS SCT converte todos os scripts de tarefa Teradata MultiLoad selecionados em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.

7. Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte [Como editar e salvar seus scripts de tarefa do Teradata MultiLoad convertidos](#).

Como gerenciar scripts de tarefa do Teradata MultiLoad com a AWS SCT

Você pode adicionar ou remover vários scripts de tarefa do Teradata MultiLoad do seu projeto da AWS SCT.

Para adicionar um novo script de tarefa do MultiLoad ao seu projeto da AWS SCT

1. No painel à esquerda, expanda o nó Scripts.
2. Escolha o nó MultiLoad e abra o menu de contexto (clique com o botão direito).
3. Escolha Carregar scripts.
4. Insira as informações necessárias para adicionar um novo script de tarefa do MultiLoad e configurar as variáveis de substituição. Para obter mais informações, consulte [Como adicionar scripts de tarefa do MultiLoad ao seu projeto da AWS SCT](#) e [Como configurar variáveis de substituição em scripts de tarefa Teradata MultiLoad](#).

Para remover um script de tarefa do MultiLoad do seu projeto da AWS SCT

1. Expanda o nó MultiLoad, em Scripts, no painel esquerdo.
2. Selecione o script a ser removido e abra o menu de contexto (clique com o botão direito).
3. Escolha Excluir script.

Como criar um relatório de avaliação de conversão para um script de tarefa do Teradata MultiLoad com a AWS SCT

O relatório de avaliação de conversão do script de tarefa do MultiLoad fornece informações sobre a conversão dos comandos MultiLoad e das instruções SQL. A conversão é feita dos seus scrips de origem para comandos do Amazon Redshift RSQL e para instruções SQL do Amazon Redshift. O relatório de avaliação inclui itens de ação para comandos MultiLoad e instruções SQL que a AWS SCT não pode converter.

Para criar um relatório de avaliação de conversão para um script de tarefa do Teradata MultiLoad

1. Expanda o nó MultiLoad, em Scripts, no painel esquerdo.
2. Escolha os scripts para os quais criar o relatório de avaliação, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar relatório.
3. Visualize a guia Resumo. A guia Resumo exibe as informações do resumo executivo do relatório de avaliação do script de tarefa do MultiLoad. Ele inclui resultados de conversão para todos os comandos MultiLoad e instruções SQL de seus scripts de origem.
4. (Opcional) Salve uma cópia local do relatório de avaliação de conversão do script de tarefa MultiLoad como um arquivo PDF ou arquivos de valores separados por vírgula (CSV):

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão de scripts.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

A AWS SCT cria dois arquivos CSV. Eles contêm o resumo executivo, itens de ação e uma complexidade estimada do esforço manual necessário para converter os scripts.

5. Selecione a guia Itens de ação. Essa guia contém uma lista de itens que exigem conversão manual para o Amazon Redshift RSQL. Se você selecionar um item de ação na lista, a AWS SCT destacará o item do seu script de tarefa do MultiLoad de origem ao qual o item de ação se aplica.

Como editar e salvar seus scripts de tarefa do Teradata MultiLoad convertidos com a AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu projeto da AWS SCT. A AWS SCT armazena o script editado como parte do projeto.

Para salvar seus scripts convertidos

1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
2. Escolha o script convertido, abra o menu de contexto (clique com o botão direito do mouse) e escolha Salvar script.

3. Insira o caminho para a pasta para salvar o script convertido e escolha Salvar.

A AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Como migrar estruturas de big data com AWS Schema Conversion Tool

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para migrar estruturas de big data para a Nuvem AWS.

Atualmente, a AWS SCT oferece suporte à migração de clusters do Hadoop para o Amazon EMR e o Amazon S3. Esse processo de migração inclui os serviços Hive e HDFS.

Além disso, você pode usar a AWS SCT para automatizar a conversão de seus fluxos de trabalho de orquestração do Apache Oozie em AWS Step Functions.

Tópicos

- [Como migrar o Apache Hadoop para o Amazon EMR com AWS Schema Conversion Tool](#)
- [Como converter o Apache Oozie para AWS Step Functions com a AWS Schema Conversion Tool](#)

Como migrar o Apache Hadoop para o Amazon EMR com AWS Schema Conversion Tool

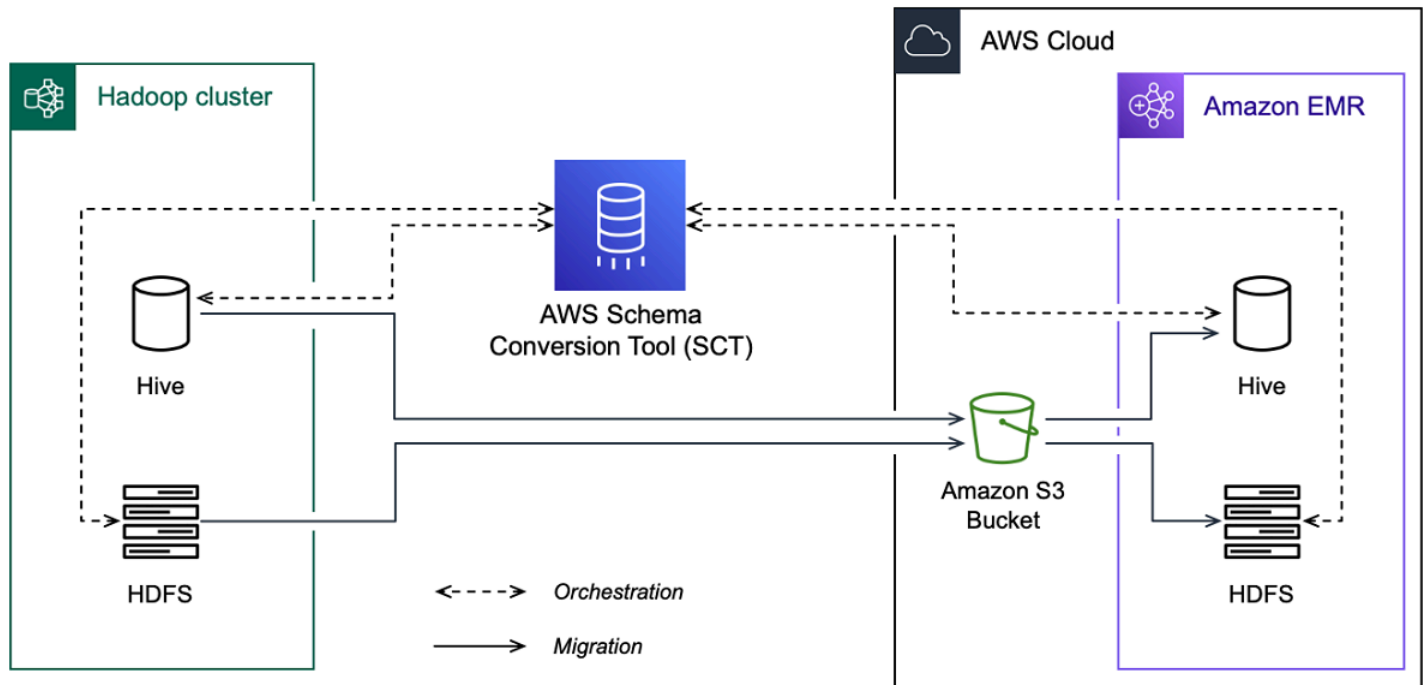
Para migrar clusters do Apache Hadoop, certifique-se de usar a versão 1.0.670 ou superior da AWS SCT. Além disso, familiarize-se com a interface de linha de comandos (CLI) da AWS SCT. Para obter mais informações, consulte [AWS SCT Referência da CLI](#).

Tópicos

- [Visão geral da migração](#)
- [Etapa 1: conectar-se aos clusters do Hadoop](#)
- [Etapa 2: configurar as regras de mapeamento](#)
- [Etapa 3: criar um relatório de avaliação](#)
- [Etapa 4: migrar seu cluster do Apache Hadoop para o Amazon EMR com a AWS SCT](#)
- [Como executar seu script de CLI](#)
- [Como gerenciar projeto de migração de big data](#)

Visão geral da migração

A imagem a seguir mostra o diagrama de arquitetura da migração do Apache Hadoop para o Amazon EMR.



A AWS SCT migra dados e metadados do cluster do Hadoop de origem para um bucket do Amazon S3. Em seguida, a AWS SCT usa seus metadados de origem do Hive para criar objetos de banco de dados no serviço Hive do Amazon EMR de destino. Opcionalmente, você pode configurar o Hive para usar o AWS Glue Data Catalog como seu metastore. Nesse caso, a AWS SCT migra seus metadados de origem do Hive para o AWS Glue Data Catalog.

Em seguida, você pode usar a AWS SCT para migrar os dados de um bucket do Amazon S3 para seu serviço de destino do Amazon EMR HDFS. Como alternativa, você pode deixar os dados em seu bucket do Amazon S3 e usá-los como um repositório de dados para suas workloads do Hadoop.

Para iniciar a migração do Hadoop, você cria e executa seu script CLI da AWS SCT. Esse script inclui o conjunto completo de comandos para executar a migração. Você pode baixar e editar um modelo do script de migração do Hadoop. Para obter mais informações, consulte [Obter cenários de CLI](#).

Certifique-se de que seu script inclua as etapas a seguir para que você possa executar sua migração do Apache Hadoop para o Amazon S3 e o Amazon EMR.

Etapa 1: conectar-se aos clusters do Hadoop

Para iniciar a migração do seu cluster Apache Hadoop, crie um novo projeto da AWS SCT. Em seguida, conecte-se aos clusters de origem e de destino. Certifique-se de criar e provisionar seus recursos da AWS de destino antes de iniciar a migração.

Nesta etapa, você usa os seguintes comandos CLI da AWS SCT.

- `CreateProject` – para criar um novo projeto da AWS SCT.
- `AddSourceCluster` – para conectar-se ao cluster Hadoop de origem em seu projeto da AWS SCT.
- `AddSourceClusterHive` – para conectar-se ao serviço Hive de origem em seu projeto.
- `AddSourceClusterHDFS` – para conectar-se ao serviço HDFS de origem em seu projeto.
- `AddTargetCluster` – para conectar-se cluster do Amazon EMR de destino em seu projeto.
- `AddTargetClusterS3` – para adicionar o bucket do Amazon S3 ao seu projeto.
- `AddTargetClusterHive` – para conectar-se ao serviço Hive de destino em seu projeto
- `AddTargetClusterHDFS` – para conectar-se ao serviço HDFS de destino em seu projeto

Para obter exemplos de uso desses comandos CLI da AWS SCT, consulte [Como usar o Apache Hadoop como origem](#).

Quando você executa o comando que se conecta a um cluster de origem ou de destino, a AWS SCT tenta estabelecer a conexão com esse cluster. Se a tentativa de conexão falhar, a AWS SCT interrompe a execução dos comandos do script da CLI e exibe uma mensagem de erro.

Etapa 2: configurar as regras de mapeamento

Depois de se conectar aos clusters de origem e de destino, configure as regras de mapeamento. Uma regra de mapeamento define a meta de migração para um cluster de origem. Certifique-se de configurar regras de mapeamento para todos os clusters de origem que você adicionou ao seu projeto da AWS SCT. Para obter mais informações sobre regras de mapeamento, consulte [Criação de regras de mapeamento na AWS SCT](#).

Para esta etapa, use o comando `AddServerMapping`. Esse comando usa dois parâmetros, que definem os clusters de origem e de destino. Você pode usar o comando `AddServerMapping` com o caminho explícito para seus objetos de banco de dados ou com nomes de objetos. Para a primeira

opção, você inclui o tipo do objeto e seu nome. Para a segunda opção, você inclui somente os nomes dos objetos.

- `sourceTreePath` – o caminho explícito para os objetos do banco de dados de origem.
`targetTreePath` – o caminho explícito para os objetos do banco de dados de destino.
- `sourceNamePath` – o caminho que inclui somente os nomes dos seus objetos de origem.
`targetNamePath` – o caminho que inclui somente os nomes dos seus objetos de destino.

O exemplo de código a seguir cria uma regra de mapeamento usando caminhos explícitos para o banco de dados Hive `testdb` de origem e o cluster EMR de destino.

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Você pode usar esse exemplo e os exemplos a seguir no Windows. Para executar os comandos da CLI no Linux, verifique se você atualizou os caminhos de arquivo adequadamente para o seu sistema operacional.

O exemplo de código a seguir cria uma regra de mapeamento usando os caminhos que incluem somente os nomes dos objetos.

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Você pode escolher o Amazon EMR ou o Amazon S3 como destino para seu objeto de origem. Para cada objeto de origem, você pode escolher somente um destino em um único projeto da AWS SCT. Para alterar o destino de migração de um objeto de origem, exclua a regra de mapeamento existente e crie uma nova regra de mapeamento. Para excluir uma regra de mapeamento, use o comando `DeleteServerMapping`. Esse comando usa um dos dois parâmetros a seguir.

- `sourceTreePath` – o caminho explícito para os objetos do banco de dados de origem.
- `sourceNamePath` – o caminho que inclui somente os nomes dos seus objetos de origem.

Para obter mais informações sobre os comandos `AddServerMapping` e `DeleteServerMapping`, consulte a [Referência da CLI da AWS Schema Conversion Tool](#).

Etapa 3: criar um relatório de avaliação

Antes de iniciar a migração, recomendamos criar um relatório de avaliação. Esse relatório resume todas as tarefas de migração e detalha os itens de ação que surgirão durante a migração. Para garantir que sua migração não falhe, visualize esse relatório e aborde os itens de ação antes da migração. Para obter mais informações, consulte [Relatórios de avaliação de migração](#).

Para esta etapa, use o comando `CreateMigrationReport`. Esse comando usa dois parâmetros. O parâmetro `treePath` é obrigatório e o parâmetro `forceMigrate` é opcional.

- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `forceMigrate` – quando definido como `true`, a AWS SCT continua a migração mesmo que seu projeto inclua uma pasta HDFS e uma tabela do Hive que façam referência ao mesmo objeto. O valor padrão é `false`.

Você pode então salvar uma cópia do relatório de avaliação como um arquivo PDF ou arquivos de valores separados por vírgula (CSV). Para isso, use o comando `SaveReportPDF` ou `SaveReportCSV`.

O comando `SaveReportPDF` salva uma cópia do seu relatório de avaliação como um arquivo PDF. Esse comando usa quatro parâmetros. O parâmetro `file` é obrigatório, outros parâmetros são opcionais.

- `file` – o caminho para o arquivo PDF e seu nome.
- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `namePath` – o caminho que inclui somente os nomes dos objetos de destino para os quais você salva uma cópia do relatório de avaliação.

O comando `SaveReportCSV` salva seu relatório de avaliação em três arquivos CSV. Esse comando usa quatro parâmetros. O parâmetro `directory` é obrigatório, outros parâmetros são opcionais.

- `directory` – o caminho para a pasta em que a AWS SCT salva os arquivos CSV.
- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `namePath` – o caminho que inclui somente os nomes dos objetos de destino para os quais você salva uma cópia do relatório de avaliação.

O exemplo de código a seguir salva uma cópia do seu relatório de avaliação no arquivo `c:\sct\ar.pdf`.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

O exemplo de código a seguir salva uma cópia do seu relatório de avaliação como arquivos CSV na pasta `c:\sct`.

```
SaveReportCSV
-file:'c:\sct'
/
```

Para obter mais informações sobre os comandos `SaveReportPDF` e `SaveReportCSV`, consulte a [Referência da CLI da AWS Schema Conversion Tool](#).

Etapa 4: migrar seu cluster do Apache Hadoop para o Amazon EMR com a AWS SCT

Depois de configurar seu projeto AWS SCT, inicie a migração do seu cluster do Apache Hadoop on-premises para a Nuvem AWS.

Para esta etapa, use os comandos `Migrate`, `MigrationStatus` e `ResumeMigration`.

O comando `Migrate` migra seus objetos de origem para o cluster de destino. Esse comando usa quatro parâmetros. Certifique-se de especificar o parâmetro `filter` ou `treePath`. Outros parâmetros são opcionais.

- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `forceLoad` – quando definido como `true`, a AWS SCT carrega automaticamente as árvores de metadados do banco de dados durante a migração. O valor padrão é `false`.
- `forceMigrate` – quando definido como `true`, a AWS SCT continua a migração mesmo que seu projeto inclua uma pasta HDFS e uma tabela do Hive que façam referência ao mesmo objeto. O valor padrão é `false`.

O comando `MigrationStatus` retorna informações sobre o progresso da migração. Para executar esse comando, insira o nome do seu projeto de migração para o parâmetro `name`. Você especificou esse nome no comando `CreateProject`.

O comando `ResumeMigration` retoma a migração interrompida que você iniciou usando o comando `Migrate`. O comando `ResumeMigration` não usa parâmetros. Para continuar a migração, você deve se conectar aos clusters de origem e de destino. Para obter mais informações, consulte [Como gerenciar projeto de migração](#).

O exemplo de código a seguir migra dados do seu serviço HDFS de origem para o Amazon EMR.

```
Migrate
-treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
-forceMigrate: 'true'
/
```

Como executar seu script de CLI

Depois de terminar de editar seu script de CLI da AWS SCT, salve-o como um arquivo com a extensão `.scts`. Agora, você pode executar seu script a partir da pasta `app` do caminho de instalação da AWS SCT. Para fazer isso, use o comando a seguir.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

No exemplo anterior, substitua `script_path` pelo caminho do seu arquivo por um script de CLI. Para obter mais informações sobre a execução de scripts de CLI na AWS SCT, consulte [Modo de script](#).

Como gerenciar projeto de migração de big data

Depois de concluir a migração, você pode salvar e editar seu projeto da AWS SCT para uso futuro.

Para salvar seu projeto da AWS SCT, use o comando `SaveProject`. Este comando não usa parâmetros.

O exemplo de código a seguir salva seu projeto da AWS SCT.

```
SaveProject  
/
```

Para abrir seu projeto da AWS SCT, use o comando `OpenProject`. Este comando usa um parâmetro obrigatório. Para o parâmetro `file`, insira o caminho para o arquivo do projeto AWS SCT e seu nome. Você especificou o nome do projeto no comando `CreateProject`. Certifique-se de adicionar a extensão `.scts` ao nome do arquivo do projeto para executar o comando `OpenProject`.

O exemplo de código a seguir cria o projeto do `hadoop_emr` a partir da pasta `c:\sct`.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Depois de abrir seu projeto da AWS SCT, você não precisa adicionar os clusters de origem e de destino porque você já os adicionou ao seu projeto. Para começar a trabalhar com seus clusters de origem e destino, você deve se conectar a eles. Para fazer isso, use os comandos `ConnectSourceCluster` e `ConnectTargetCluster`. Esses comandos usam os mesmos parâmetros dos comandos `AddSourceCluster` e `AddTargetCluster`. Você pode editar seu script de CLI e substituir o nome desses comandos, deixando a lista de parâmetros sem alterações.

O exemplo de código a seguir conecta o cluster do Hadoop de origem.

```
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'  
/  

```

O exemplo de código a seguir conecta o cluster do 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'  
/  

```

No exemplo anterior, substitua *hadoop_address* pelo endereço IP do seu cluster Hadoop. Se necessário, configure o valor da variável de porta. Em seguida, substitua *hadoop_user* e *hadoop_password* pelo nome do seu usuário do Hadoop e pela senha desse usuário. Em *path \name*, insira o nome e o caminho para o arquivo PEM do seu cluster Hadoop de origem. Para obter mais informações sobre a inclusão dos clusters de origem e destino, consulte [Como usar o Apache Hadoop como origem para AWS SCT](#).

Depois de se conectar aos clusters do Hadoop de origem e de destino, você deve se conectar aos serviços Hive e HDFS, bem como ao bucket do Amazon S3. Para fazer isso, use os comandos `ConnectSourceClusterHive`, `ConnectSourceClusterHdfs`, `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs` e `ConnectTargetClusterS3`. Esses comandos usam os mesmos parâmetros que você usou para adicionar serviços Hive e HDFS e o bucket Amazon S3 ao seu projeto. Edite o script de CLI para substituir o prefixo `Add` pelos nomes dos comandos `Connect`.

Como converter o Apache Oozie para AWS Step Functions com a AWS Schema Conversion Tool

Para converter fluxos de trabalho do Apache Oozie, certifique-se de usar a versão 1.0.671 ou superior da AWS SCT. Além disso, familiarize-se com a interface de linha de comandos (CLI) da AWS SCT. Para obter mais informações, consulte [AWS SCT Referência da CLI](#).

Tópicos

- [Visão geral da conversão](#)
- [Etapa 1: conectar-se aos seus serviços de origem e destino](#)
- [Etapa 2: configurar as regras de mapeamento](#)
- [Etapa 3: configurar parâmetros](#)
- [Etapa 4: criar um relatório de avaliação](#)
- [Etapa 5: converter seus fluxos de trabalho do Apache Oozie para AWS Step Functions com AWS SCT](#)
- [Como executar seu script de CLI](#)
- [Nós do Apache Oozie que a AWS SCT pode converter para AWS Step Functions](#)

Visão geral da conversão

Seu código fonte do Apache Oozie inclui nós de ação, nós de fluxo de controle e propriedades do trabalho. Os nós de ação definem as tarefas que você executa em seu fluxo de trabalho do Apache Oozie. Quando você usa o Apache Oozie para orquestrar seu cluster do Apache Hadoop, um nó de ação inclui uma tarefa do Hadoop. Os nós de controle de fluxo fornecem um mecanismo para controlar o caminho do fluxo de trabalho. Os nós do fluxo de controle incluem nós como `start`, `end`, `decision`, `fork` e `join`.

A AWS SCT converte seus nós de ação de origem e nós de fluxo de controle em AWS Step Functions. Em AWS Step Functions, você define seus fluxos de trabalho na Amazon States Language (ASL). A AWS SCT usa a ASL para definir sua máquina de estado, que é uma coleção de estados que pode realizar trabalho, determinar quais estados mudar depois, interromper com um erro e assim por diante. Em seguida, a AWS SCT carrega os arquivos JSON com as definições das máquinas de estado. Depois, a AWS SCT pode usar seu perfil (IAM) AWS Identity and Access Management para configurar suas máquinas de estado em AWS Step Functions. Para obter mais

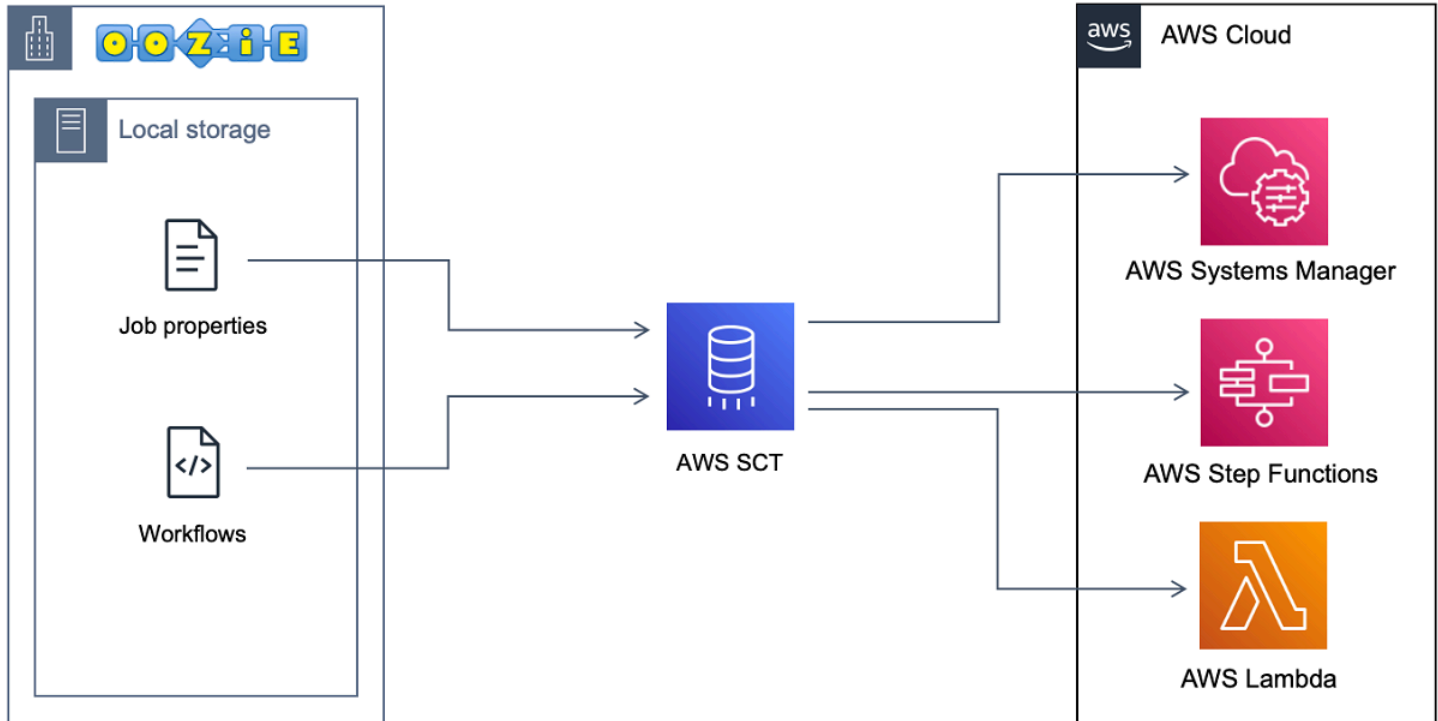
informações, consulte [O que é o AWS Step Functions?](#) no Guia do desenvolvedor do AWS Step Functions.

Além disso, a AWS SCT cria um pacote de extensão com perfis AWS Lambda que emulam os perfis de origem que o AWS Step Functions não suporta. Para obter mais informações, consulte [Usando AWS SCT pacotes de extensão](#).

A AWS SCT migra suas propriedades de trabalho de origem para o AWS Systems Manager. Para armazenar nomes e valores de parâmetros, a AWS SCT usa o repositório de parâmetros, um recurso do AWS Systems Manager. Para obter mais informações, consulte [O que é o AWS Systems Manager?](#) no Guia do usuário do AWS Systems Manager.

Você pode usar a AWS SCT para atualizar automaticamente os valores e os nomes dos seus parâmetros. Devido às diferenças de arquitetura entre o Apache Oozie e AWS Step Functions, talvez seja necessário configurar seus parâmetros. A AWS SCT pode encontrar um nome ou valor de parâmetro especificado em seus arquivos de origem e substituí-los por novos valores. Para obter mais informações, consulte [Etapa 3: configurar parâmetros](#).

A imagem a seguir mostra o diagrama de arquitetura da conversão do Apache Oozie para o AWS Step Functions.



Para iniciar a conversão, crie e execute seu script de CLI da AWS SCT. Esse script inclui o conjunto completo de comandos para executar a conversão. Você pode baixar e editar um modelo do script de conversão do Apache Oozie. Para obter mais informações, consulte [Obter cenários de CLI](#).

Verifique se o script inclui as seguintes etapas.

Etapa 1: conectar-se aos seus serviços de origem e destino

Para iniciar a conversão do seu cluster Apache Oozie, crie um novo projeto da AWS SCT. Em seguida, conecte-se aos seus serviços de origem e destino. Certifique-se de criar e provisionar seus recursos da AWS de destino antes de iniciar a migração. Para obter mais informações, consulte [Pré-requisitos de uso do Apache Oozie como origem](#).

Nesta etapa, você usa os seguintes comandos CLI da AWS SCT.

- `CreateProject` – para criar um novo projeto da AWS SCT.
- `AddSource` – para adicionar seus arquivos Apache Oozie de origem em seu projeto da AWS SCT.
- `ConnectSource` – para conectar-se ao Apache Oozie como origem.
- `AddTarget` – para adicionar AWS Step Functions como alvo de migração em seu projeto.
- `ConnectTarget` – Para conectar-se ao AWS Step Functions.

Para obter exemplos de uso desses comandos CLI da AWS SCT, consulte [Como usar o Apache Oozie como origem](#).

Quando você executa os comandos `ConnectSource` ou `ConnectTarget`, a AWS SCT tenta estabelecer a conexão com seus serviços. Se a tentativa de conexão falhar, a AWS SCT interrompe a execução dos comandos do script da CLI e exibe uma mensagem de erro.

Etapa 2: configurar as regras de mapeamento

Depois de se conectar aos serviços de origem e de destino, configure as regras de mapeamento. Uma regra de mapeamento define a meta de migração para seus fluxos de trabalho e parâmetros de origem do Apache Oozie. Para obter mais informações sobre regras de mapeamento, consulte [Criação de regras de mapeamento na AWS SCT](#).

Para definir objetos de origem e destino para conversão, use o comando `AddServerMapping`. Esse comando usa dois parâmetros: `sourceTreePath` e `targetTreePath`. Os valores desses parâmetros incluem um caminho explícito para seus objetos de origem e destino. Para que o Apache Oozie faça a conversão do AWS Step Functions, esses parâmetros devem começar com ETL.

O exemplo de código a seguir cria uma regra de mapeamento para os objetos OOOZIE e AWS_STEP_FUNCTIONS. Você adicionou esses objetos ao seu projeto da AWS SCT usando os comandos AddSource e AddTarget na etapa anterior.

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Para obter mais informações sobre o comando AddServerMapping, consulte [Referência da CLI do AWS Schema Conversion Tool](#).

Etapa 3: configurar parâmetros

Se seus fluxos de trabalho de origem do Apache Oozie usarem parâmetros, talvez seja necessário alterar seus valores após a conversão para o AWS Step Functions. Além disso, pode ser necessário adicionar novos parâmetros para usar com o seu AWS Step Functions.

Para esta etapa, use os comandos AddParameterMapping e AddTargetParameter.

Para substituir os valores dos parâmetros em seus arquivos de origem, use o comando AddParameterMapping. A AWS SCT verifica seus arquivos de origem, encontra os parâmetros por nome ou valor e altera seus valores. Você pode executar um único comando para verificar todos os seus arquivos de origem. Você define o escopo dos arquivos a serem examinados usando um dos três primeiros parâmetros da lista a seguir. Esse comando usa até seis parâmetros.

- `filterName` – o nome do filtro para seus objetos de origem. Você pode criar um filtro usando o comando `CreateFilter`.
- `treePath` – o caminho explícito para os objetos de origem.
- `namePath` – o caminho explícito para um objeto específico de origem.
- `sourceParameterName` – o nome do seu parâmetro de origem.
- `sourceValue` – o valor do seu parâmetro de origem.
- `targetValue` – o nome do seu parâmetro de destino.

O exemplo de código a seguir substitui todos os parâmetros em que o valor é igual a `c:\oozie\hive.py` com o valor `s3://bucket-oozie/hive.py`.

```
AddParameterMapping
-treePath: 'ETL.00ZIE.Applications'
-sourceValue: 'c:\oozie\hive.py'
-targetValue: 's3://bucket-oozie/hive.py'
/
```

O exemplo de código a seguir substitui todos os parâmetros em que o nome é igual a `nameNode` com o valor `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020`.

```
AddParameterMapping
-treePath: 'ETL.00ZIE_SOURCE.Applications'
-sourceParameter: 'nameNode'
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

O exemplo de código a seguir substitui todos os parâmetros em que o nome é igual a `nameNode` e o valor é igual a `hdfs://ip-55.eu-west-1.compute.internal:8020` com o valor do parâmetro `targetValue`.

```
AddParameterMapping
-treePath: 'ETL.00ZIE_SOURCE.Applications'
-sourceParameter: 'nameNode'
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

Para adicionar um novo parâmetro nos arquivos de destino, além de um parâmetro existente nos arquivos de origem, use o comando `AddTargetParameter`. Esse comando usa o mesmo conjunto de parâmetros do comando `AddParameterMapping`.

O exemplo de código a seguir adiciona o parâmetro de destino `clusterId` em vez do parâmetro `nameNode`.

```
AddTargetParameter
-treePath: 'ETL.00ZIE_SOURCE.Applications'
-sourceParameter: 'nameNode'
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
-targetParameter: 'clusterId'
-targetValue: '1234567890abcdef0'
/
```

Para obter mais informações sobre os comandos `AddServerMapping`, `AddParameterMapping`, `AddTargetParameter` e `CreateFilter`, consulte a [Referência da CLI da AWS Schema Conversion Tool](#).

Etapa 4: criar um relatório de avaliação

Antes de iniciar a conversão, recomendamos criar um relatório de avaliação. Esse relatório resume todas as tarefas de migração e detalha os itens de ação que surgirão durante a migração. Para garantir que sua migração não falhe, visualize esse relatório e aborde os itens de ação antes da migração. Para obter mais informações, consulte [Relatórios de avaliação de migração](#).

Para esta etapa, use o comando `CreateReport`. Esse comando usa dois parâmetros. O primeiro parâmetro descreve os objetos de origem para os quais a AWS SCT cria um relatório de avaliação. Para fazer isso, utilize um dos seguintes parâmetros: `filterName`, `treePath` ou `namePath`. Esse parâmetro é obrigatório. Além disso, você pode adicionar um parâmetro booleano opcional `forceLoad`. Se você definir esse parâmetro como `true`, AWS SCT carregará automaticamente todos os objetos secundários do objeto de origem que você especificar no comando `CreateReport`.

O exemplo de código a seguir cria um relatório de avaliação para o nó `Applications` dos seus arquivos `Oozie` de origem.

```
CreateReport
  -treePath: 'ETL.APACHE_00ZIE.Applications'
/
```

Você pode então salvar uma cópia do relatório de avaliação como um arquivo PDF ou arquivos de valores separados por vírgula (CSV). Para isso, use o comando `SaveReportPDF` ou `SaveReportCSV`.

O comando `SaveReportPDF` salva uma cópia do seu relatório de avaliação como um arquivo PDF. Esse comando usa quatro parâmetros. O parâmetro `file` é obrigatório, outros parâmetros são opcionais.

- `file` – o caminho para o arquivo PDF e seu nome.
- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.

- `namePath` – o caminho que inclui somente os nomes dos objetos de destino para os quais você salva uma cópia do relatório de avaliação.

O comando `SaveReportCSV` salva seu relatório de avaliação em arquivos CSV. Esse comando usa quatro parâmetros. O parâmetro `directory` é obrigatório, outros parâmetros são opcionais.

- `directory` – o caminho para a pasta em que a AWS SCT salva os arquivos CSV.
- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `namePath` – o caminho que inclui somente os nomes dos objetos de destino para os quais você salva uma cópia do relatório de avaliação.

O exemplo de código a seguir salva uma cópia do seu relatório de avaliação no arquivo `c:\sct\ar.pdf`.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

O exemplo de código a seguir salva uma cópia do seu relatório de avaliação como arquivos CSV na pasta `c:\sct`.

```
SaveReportCSV
-file:'c:\sct'
/
```

Para obter mais informações sobre os comandos `CreateReport`, `SaveReportPDF` e `SaveReportCSV`, consulte a [Referência da CLI da AWS Schema Conversion Tool](#).

Etapa 5: converter seus fluxos de trabalho do Apache Oozie para AWS Step Functions com AWS SCT

Depois de configurar seu projeto da AWS SCT, converta seu código fonte e aplique-o à Nuvem AWS.

Para esta etapa, use os comandos `Convert`, `SaveOnS3`, `ConfigureStateMachine` e `ApplyToTarget`.

O comando `Migrate` migra seus objetos de origem para o cluster de destino. Esse comando usa quatro parâmetros. Certifique-se de especificar o parâmetro `filter` ou `treePath`. Outros parâmetros são opcionais.

- `filter` – o nome do filtro que você criou antes para definir o escopo dos objetos de origem a serem migrados.
- `namePath` – o caminho explícito para um objeto específico de origem.
- `treePath` – o caminho explícito para os objetos do banco de dados de origem para os quais você salva uma cópia do relatório de avaliação.
- `forceLoad` – quando definido como `true`, a AWS SCT carrega automaticamente as árvores de metadados do banco de dados durante a migração. O valor padrão é `false`.

O exemplo de código a seguir converte arquivos da pasta `Applications` em seus arquivos Oozie de origem.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

O `SaveOnS3` carrega as definições de estado das máquinas para o seu bucket do Amazon S3. Esse comando usa o parâmetro `treePath`. Para executar esse comando, use a pasta de destino com definições de máquinas de estado como o valor desse parâmetro.

O seguinte carrega a pasta `State machine definitions` do seu objeto de destino `AWS_STEP_FUNCTIONS` no bucket do Amazon S3. A AWS SCT usa o bucket do Amazon S3 que você armazenou no perfil de serviço da AWS na etapa [Pré-requisitos](#).

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

O comando `ConfigureStateMachine` configura as máquinas de estado. Esse comando usa até seis parâmetros. Certifique-se de definir o escopo de destino usando um dos três primeiros parâmetros da lista a seguir.

- `filterName` – o nome do filtro para seus objetos de destino. Você pode criar um filtro usando o comando `CreateFilter`.
- `treePath` – o caminho explícito para os objetos de destino.
- `namePath` – o caminho explícito para um objeto específico de destino.
- `iamRole` – o nome do recurso da Amazon (ARN) do perfil do IAM que fornece acesso às suas máquinas de etapa. Esse parâmetro é obrigatório.

O exemplo de código a seguir configura as máquinas de estado definidas em `AWS_STEP_FUNCTIONS` usando o perfil do IAM `role_name`.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

O comando `ApplyToTarget` aplica seu código convertido ao servidor de destino. Para executar esse comando, use um dos seguintes parâmetros: `filterName`, `treePath`, ou `namePath` para definir os objetos de destino a serem aplicados.

O exemplo de código a seguir aplica a máquina de estado `app_wp` ao AWS Step Functions.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Para garantir que seu código convertido produza os mesmos resultados que seu código fonte, você pode usar o pacote de extensão AWS SCT. Este é um conjunto de perfis AWS Lambda que emulam seus perfis do Apache Oozie que o AWS Step Functions não suporta. Para instalar esse pacote de extensão, você pode usar o comando `CreateLambdaExtPack`.

Esse comando usa até cinco parâmetros. Certifique-se de usar o **Oozie2SF** para `extPackId`. Nesse caso, a AWS SCT cria um pacote de extensão para as funções de origem do Apache Oozie.

- `extPackId` – o identificador exclusivo para um conjunto de funções do Lambda. Esse parâmetro é obrigatório.
- `tempDirectory` – o caminho onde a AWS SCT pode armazenar arquivos temporários. Esse parâmetro é obrigatório.

- `awsProfile` – o nome do seu perfil da AWS.
- `lambdaExecRoles` – a lista de nomes do recurso da Amazon (ARN) dos perfis de execução a serem usados para funções do Lambda.
- `createInvokeRoleFlag` – o sinalizador booleano que indica se uma função de execução deve ser criada para AWS Step Functions.

Para instalar e usar o pacote de extensão, verifique se você fornece as permissões necessárias. Para obter mais informações, consulte [Permissões para usar funções AWS Lambda no pacote de extensões](#).

Para obter mais informações sobre os comandos `Convert`, `SaveOnS3`, `ConfigureStateMachine`, `ApplyToTarget` e `CreateLambdaExtPack`, consulte a [Referência da CLI da AWS Schema Conversion Tool](#).

Como executar seu script de CLI

Depois de terminar de editar seu script de CLI da AWS SCT, salve-o como um arquivo com a extensão `.scts`. Agora, você pode executar seu script a partir da pasta `app` do caminho de instalação da AWS SCT. Para fazer isso, use o comando a seguir.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

No exemplo anterior, substitua `script_path` pelo caminho do seu arquivo por um script de CLI. Para obter mais informações sobre a execução de scripts de CLI na AWS SCT, consulte [Modo de script](#).

Nós do Apache Oozie que a AWS SCT pode converter para AWS Step Functions

Você pode usar AWS SCT para converter os nós de ação do Apache Oozie e controlar os nós de fluxo para AWS Step Functions.

Os nós de ação suportados são os seguintes:

- Ação Hive
- Ação Hive2
- Ação Spark

- Ação de streaming do MapReduce
- Ação Java
- Ação DistCP
- Ação Pig
- Ação Sqoop
- Ação FS
- Ação Shell

Os nós de fluxo de controle suportados incluem o seguinte:

- Ação de iniciar
- Ação de encerrar
- Ação de eliminar
- Ação de decisão
- Ação de garfo
- Ação de junção

Uso do AWS SCT com a AWS DMS

Como usar um agente de replicação da AWS SCT com o AWS DMS

Para migrações de banco de dados muito grandes, é possível usar um agente de replicação da AWS SCT (`aws-schema-conversion-tool-dms-agent`) para copiar dados do banco de dados on-premises para o Amazon S3 ou um dispositivo AWS Snowball Edge. O agente de replicação funciona junto com o AWS DMS e pode funcionar em segundo plano enquanto a AWS SCT está fechada.

Ao funcionar com o AWS Snowball Edge, o agente da AWS SCT extrai os dados para o dispositivo do AWS Snowball. Em seguida, o dispositivo é enviado à AWS, e os dados são carregados em um bucket do Amazon S3. Durante esse tempo, o agente da AWS SCT continua em execução. O agente recolhe os dados no Amazon S3 e copia-os em um endpoint de destino.

Para obter mais informações, consulte [Como migrar dados de um data warehouse on-premises para o Amazon Redshift](#).

Como usar um agente de extração de dados da AWS SCT com o AWS DMS

Na AWS SCT, você encontra um agente de extração de dados (`aws-schema-conversion-tool-extractor`) que ajuda a facilitar as migrações do Apache Cassandra para o Amazon DynamoDB. O Cassandra e o DynamoDB são bancos de dados NoSQL, mas diferem na arquitetura do sistema e na representação dos dados. Você pode usar fluxos de trabalho baseados em assistentes na AWS SCT para automatizar o processo de migração do Cassandra para o DynamoDB. A AWS SCT integra-se ao AWS Database Migration Service (AWS DMS) para realizar efetivamente a migração.

Para obter mais informações, consulte [Como migrar dados de um data warehouse on-premises para o Amazon Redshift](#).

Como aumentar os níveis de registro ao usar a AWS SCT com o AWS DMS

Você pode aumentar os níveis de registro ao usar a AWS SCT com o AWS DMS, por exemplo, se precisar trabalhar com o Suporte da AWS.

Após a instalação da AWS SCT e dos drivers necessários, abra o aplicativo escolhendo o ícone da AWS SCT. Se você receber uma notificação de atualização, poderá optar por atualizar antes ou depois da conclusão do projeto. Se uma janela de projeto automático for aberta, feche a janela e crie um projeto manualmente.

Para aumentar os níveis de registro ao usar a AWS SCT com o AWS DMS

1. No menu Configurações, escolha Configurações globais.
2. Na janela Configurações globais, escolha Registro em log.
3. Em Modo de depuração, selecione Verdadeiro.
4. Na seção Nível da mensagem, você pode modificar os seguintes tipos de logs:
 - Geral
 - Carregador
 - Analisador
 - Impressora
 - Resolvedor
 - Telemetria
 - Conversor

Por padrão, todos os níveis de mensagem são definidos como Informações.

5. Escolha um nível de registro para qualquer tipo de nível de mensagem que você queira alterar:
 - Rastreamento (registro mais detalhado)
 - Depure
 - Informações
 - Aviso
 - Erro (registro menos detalhado)

- Critical
 - obrigatório
6. Escolha Aplicar para modificar as configurações do seu projeto.
 7. Escolha OK para fechar a janela Configurações globais.

Como migrar dados de um data warehouse on-premises para o Amazon Redshift

Você pode usar um AWS SCT agente para extrair dados do seu armazém de dados local e migrá-los para o Amazon Redshift. O agente extrai seus dados e os carrega para o Amazon S3 ou, para migrações em grande escala, para um dispositivo Edge. AWS Snowball Em seguida, você pode usar um AWS SCT agente para copiar os dados para o Amazon Redshift.

Como alternativa, você pode usar AWS Database Migration Service (AWS DMS) para migrar dados para o Amazon Redshift. A vantagem do AWS DMS é o suporte à replicação contínua (captura de dados de alteração). No entanto, para aumentar a velocidade da migração de dados, use vários AWS SCT agentes em paralelo. De acordo com nossos testes, os AWS SCT agentes migram dados mais rápido do que 15 a AWS DMS 35 por cento. A diferença na velocidade se deve à compactação de dados, ao suporte à migração de partições de tabela em paralelo e às diferentes configurações. Para obter mais informações, consulte [Usar um banco de dados do Amazon Redshift como destino do AWS Database Migration Service](#).

O Amazon S3 é um serviço de armazenamento e recuperação. Para armazenar um objeto no Amazon S3, você carrega o arquivo que deseja armazenar em um bucket do Amazon S3. Quando você carrega um arquivo, pode definir permissões sobre o objeto e também em todos os metadados.

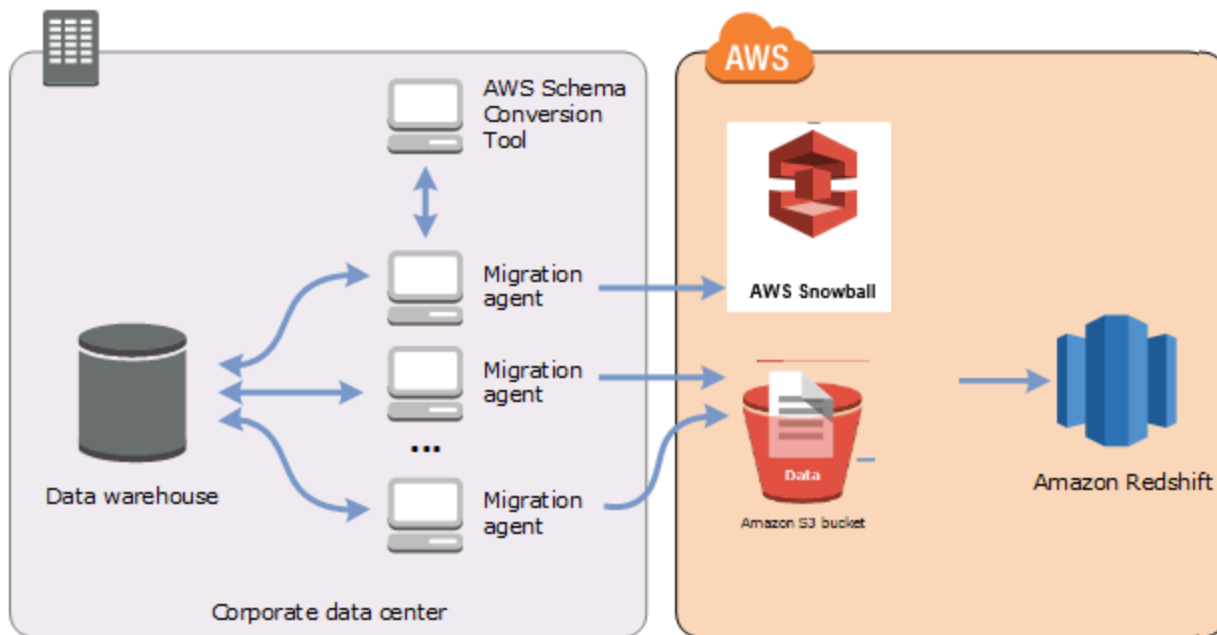
Migrações em larga escala

As migrações de dados em grande escala podem incluir muitos terabytes de informações e podem ser retardadas pelo desempenho da rede e pela grande quantidade de dados que precisam ser movidos. AWS Snowball O Edge é um AWS serviço que você pode usar para transferir dados para a nuvem em faster-than-network alta velocidade usando um dispositivo AWS próprio. Um dispositivo AWS Snowball Edge pode armazenar até 100 TB de dados. Ele usa criptografia de 256 bits e um Trusted Platform Module (TPM) padrão do setor para garantir a segurança e a integridade de seus dados. chain-of-custody AWS SCT funciona com dispositivos AWS Snowball Edge.

Ao usar AWS SCT um dispositivo AWS Snowball Edge, você migra seus dados em dois estágios. Primeiro, você usa AWS SCT para processar os dados localmente e depois mover esses dados para o dispositivo AWS Snowball Edge. Em seguida, você envia o dispositivo para AWS usar o processo AWS Snowball Edge e, em seguida, carrega AWS automaticamente os dados em um bucket do Amazon S3. Em seguida, quando os dados estiverem disponíveis no Amazon S3, você os usará

AWS SCT para migrar os dados para o Amazon Redshift. Os agentes de extração de dados podem trabalhar em segundo plano enquanto AWS SCT estão fechados.

O diagrama a seguir mostra o cenário com suporte.



No momento, os agentes de extração de dados recebem suporte dos seguintes data warehouses de origem:

- Azure Synapse Analytics
- BigQuery
- Banco de dados Greenplum (versão 4.3)
- Microsoft SQL Server (versão 2008 e superior)
- Netezza (versão 7.0.3 e superior)
- Oracle (versão 10 e superior)
- Snowflake (versão 3)
- Teradata (versão 13 e superior)
- Vertica (versão 7.2.2 e superior)

Você pode conectar os endpoints do FIPS para o Amazon Redshift se precisar estar em conformidade com os requisitos de segurança do Padrão Federal de Processamento de Informações (FIPS). Os endpoints FIPS estão disponíveis nas seguintes regiões: AWS

- Região Leste dos EUA (Norte da Virgínia) (redshift-fips.us-east-1.amazonaws.com)
- Região Leste dos EUA (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Região Oeste dos EUA (Norte da Califórnia) (redshift-fips.us-west-1.amazonaws.com)
- Região Oeste dos EUA (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Use as informações nos tópicos a seguir para saber como trabalhar com os agentes de extração de dados.

Tópicos

- [Pré-requisitos para usar os atendentes de extração de dados](#)
- [Como instalar atendentes de extração](#)
- [Como configurar atendentes de extração](#)
- [Registrando agentes de extração com o AWS Schema Conversion Tool](#)
- [Ocultar e recuperar informações para um agente AWS SCT](#)
- [Criação de regras de migração de dados no AWS SCT](#)
- [Como alterar as configurações do extrator e da cópia das configurações do projeto](#)
- [Classificando dados antes de migrar usando AWS SCT](#)
- [Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados](#)
- [Exportação e importação de uma tarefa de extração de AWS SCT dados](#)
- [Extração de dados usando um dispositivo AWS Snowball Edge](#)
- [Saída da tarefa de extração de dados](#)
- [Usando particionamento virtual com AWS Schema Conversion Tool](#)
- [Como usar o particionamento nativo](#)
- [Como migrar LOBs para o Amazon Redshift](#)
- [Melhores práticas e solução de problemas para atendentes de extração de dados](#)

Pré-requisitos para usar os atendentes de extração de dados

Antes de trabalhar com atendentes de extração de dados, adicione as permissões necessárias para o Amazon Redshift como destino para seu usuário do Amazon Redshift. Para ter mais informações, consulte [Permissões para o Amazon Redshift como destino](#).

Em seguida, armazene as informações do bucket do Amazon S3 e configure a confiança e o armazenamento de chaves do Secure Sockets Layer (SSL).

Configurações do Amazon S3

Os atendentes extraem os dados e, depois, os carregam para o bucket do Amazon S3. Antes de continuar, você deve fornecer as credenciais para se conectar à sua AWS conta e ao seu bucket do Amazon S3. Você armazena suas credenciais e informações do bucket em um perfil nas configurações globais do aplicativo e, em seguida, associa o perfil ao seu AWS SCT projeto. Se necessário, selecione Configurações globais para criar um novo perfil. Para ter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).

Para migrar dados para seu banco de dados de destino do Amazon Redshift, AWS SCT o agente de extração de dados precisa de permissão para acessar o bucket do Amazon S3 em seu nome. Para fornecer essa permissão, crie um usuário AWS Identity and Access Management (IAM) com a política a seguir.

```
{
  "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"
}
]
```

No exemplo anterior, substitua *bucket_name* pelo nome do seu bucket do Amazon S3. Depois, substitua *111122223333:user/DataExtractionAgentName* pelo nome do seu usuário do IAM.

Como assumir perfis do IAM

Para segurança adicional, você pode usar funções AWS Identity and Access Management (IAM) para acessar seu bucket do Amazon S3. Para fazer isso, crie um usuário do IAM para seus atendentes de extração de dados sem nenhuma permissão. Em seguida, crie um perfil do IAM que permita o acesso ao Amazon S3 e especifique a lista de serviços e usuários que podem assumir esse perfil. Para obter mais informações, consulte [Funções do IAM](#) no Guia do usuário do IAM.

Para configurar um perfil do IAM para acessar o bucket do Amazon S3

1. Crie um novo usuário do IAM. Para credenciais do usuário, selecione o tipo de Acesso programático.

- Configure o ambiente do host para que seu agente de extração de dados possa assumir a função que AWS SCT fornece. Certifique-se de que o usuário que você configurou na etapa anterior permita que os atendentes de extração de dados usem a cadeia de fornecedores de credenciais. Para obter mais informações, consulte [Como usar credenciais](#) no Guia do desenvolvedor do AWS SDK for Java .
- Crie um novo perfil do IAM que tenha acesso ao seu bucket do Amazon S3.
- Modifique a seção confiável desse perfil para confiar no usuário que você criou antes para assumir o perfil. Nos exemplos a seguir, substitua `111122223333:user/DataExtractionAgentName` pelo nome do seu usuário.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
  },
  "Action": "sts:AssumeRole"
}
```

- Modifique a seção confiável desse perfil para confiar em `redshift.amazonaws.com` para assumir o perfil.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

- Anexe este perfil ao seu cluster do Amazon Redshift.

Agora, você pode executar seu atendente de extração de dados na AWS SCT.

Quando você usa a suposição do perfil do IAM, a migração de dados funciona da seguinte maneira. O atendente de extração de dados inicia e obtém as credenciais do usuário usando a cadeia de fornecedores de credenciais. Em seguida, você cria uma tarefa de migração de dados em AWS SCT, especifica a função do IAM a ser assumida pelos agentes de extração de dados e inicia a tarefa.

AWS Security Token Service (AWS STS) gera credenciais temporárias para acessar o Amazon S3. O atendente de extração de dados usa essas credenciais para carregar dados para o Amazon S3.

Em seguida, AWS SCT fornece ao Amazon Redshift a função IAM. Por sua vez, o Amazon Redshift obtém novas credenciais temporárias AWS STS para acessar o Amazon S3. O Amazon Redshift usa essas credenciais para copiar dados do Amazon S3 para a tabela do Amazon Redshift.

Configurações de segurança

Os agentes de extração AWS Schema Conversion Tool e os agentes de extração podem se comunicar por meio do Secure Sockets Layer (SSL). Para habilitar a SSL, configure um armazenamento de confiança e um armazenamento de chaves.

Para configurar a comunicação segura com o agente de extração

1. Inicie AWS Schema Conversion Tool o.
2. Abra o menu Configurações e selecione Configurações globais. A caixa de diálogo Configurações globais é exibida.
3. Selecione Segurança.
4. Selecione Gerar armazenamento confiável e de chaves ou Selecionar armazenamento confiável e de chaves existente.

Caso escolha Gerar armazenamento de confiança e de chaves, você poderá especificar o nome e a senha para os armazenamentos de confiança e de chaves, e o caminho para o local dos arquivos gerados. Você usa esses arquivos em etapas posteriores.

Caso você escolha Selecionar armazenamento confiável e de chaves existente, você poderá especificar a senha e o nome de arquivo para os armazenamentos confiável e de chaves. Você usa esses arquivos em etapas posteriores.

5. Depois de especificar o armazenamento de confiança e o armazenamento de chaves, escolha OK para fechar a caixa de diálogo Configurações globais.

Como configurar o ambiente para atendentes de extração de dados

Você pode instalar vários atendentes de extração de dados em um único host. No entanto, recomendamos executar um atendente de extração de dados em um host.

Para executar seu atendente de extração de dados, certifique-se de usar um host com pelo menos quatro vCPUs e 32 GB de memória. Além disso, defina a memória mínima disponível AWS SCT para pelo menos quatro GB. Para ter mais informações, consulte [Como configurar memória adicional](#).

A configuração ideal e o número de hosts de atendentes dependem da situação específica de cada cliente. Certifique-se de considerar fatores como quantidade de dados a serem migrados, largura de banda da rede, tempo para extrair dados e assim por diante. Você pode realizar uma prova de conceito (PoC) primeiro e depois configurar seus atendentes e hosts de extração de dados de acordo com os resultados dessa PoC.

Como instalar atendentes de extração

Recomendamos que você instale vários agentes de extração em computadores individuais, separados do computador que está executando a AWS Schema Conversion Tool.

No momento, os agentes de extração têm suporte nos seguintes sistemas operacionais:

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (versão 14.04 e posterior)

Use o seguinte procedimento para instalar agentes de extração. Repita esse procedimento para cada computador em que você deseja instalar um agente de extração.

Para instalar um agente de extração

1. Se você ainda não baixou o arquivo AWS SCT do instalador, siga as instruções em [Instalando, verificando e atualizando AWS SCT](#) para baixá-lo. O arquivo.zip que contém o arquivo do AWS SCT instalador também contém o arquivo instalador do agente de extração.
2. Faça download e instale a versão mais recente do Amazon Corretto 11. Para obter mais informações, consulte [Downloads do Amazon Corretto 11](#) no Guia do usuário do Amazon Corretto 11.
3. Localize o arquivo do instalador de seu agente de extração em uma subpasta chamada agentes. Para cada sistema operacional do computador, o arquivo correto para instalar o agente de extração é mostrado a seguir.

Sistema operacional	Nome do arquivo
Microsoft Windows	aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .msi
RHEL	aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .x86_64.rpm
Ubuntu Linux	aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .deb

- Instale o atendente de extração em um computador separado, copiando o arquivo do instalador para o novo computador.
- Execute o arquivo do instalador. Use as instruções para o seu sistema operacional, mostradas a seguir.

Sistema operacional	Instruções de instalação
Microsoft Windows	Clique duas vezes no arquivo para executar o instalador.
RHEL	<p>Execute os seguintes comandos na pasta para a qual você baixou ou moveu o arquivo.</p> <pre>sudo rpm -ivh aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .x86_64.rpm sudo ./sct-extractor-setup.sh --config</pre>
Ubuntu Linux	<p>Execute os seguintes comandos na pasta para a qual você baixou ou moveu o arquivo.</p> <pre>sudo dpkg -i aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

- Selecione Avançar, aceite o contrato de licença e depois Avançar.
- Insira o caminho para instalar o agente AWS SCT de extração de dados e escolha Avançar.
- Selecione Instalar para instalar seu atendente de extração de dados.

AWS SCT instala seu agente de extração de dados. Para concluir a instalação, configure seu agente de extração de dados. AWS SCT inicia automaticamente o programa de configuração. Para ter mais informações, consulte [Como configurar atendentes de extração](#).

9. Selecione Concluir para fechar o assistente de instalação depois de configurar seu atendente de extração de dados.

Como configurar atendentes de extração

Use o seguinte procedimento para configurar agentes de extração. Repita esse procedimento em cada computador em que você tenha um agente de extração instalado.

Para configurar seu agente de extração

1. Inicie o programa de configuração:

- No Windows, AWS SCT inicia o programa de configuração automaticamente durante a instalação de um agente de extração de dados.

Se necessário, você pode iniciar o programa de configuração manualmente. Para fazer isso, execute o arquivo `ConfigAgent.bat` no Windows. Você pode encontrar esse arquivo na pasta onde você instalou o atendente.

- No RHEL e no Ubuntu, execute o arquivo `sct-extractor-setup.sh` do local onde você instalou o atendente.

O programa de configuração solicitará informações. Para cada solicitação, um valor padrão será exibido.

2. Aceite o valor padrão em cada solicitação ou insira um novo valor.

Especifique as seguintes informações:

- Em Porta de escuta, digite o número da porta em que o atendente está escutando.
- Em Adicionar um fornecedor de origem, digite sim e, em seguida, insira sua plataforma de data warehouse de origem.
- Em driver JDBC, digite o local onde você instalou os drivers JDBC.
- Em Pasta de trabalho, insira o caminho em que o agente de extração de AWS SCT dados armazenará os dados extraídos. A pasta de trabalho pode estar em um computador diferente

do agente, e uma única pasta de trabalho pode ser compartilhada por vários agentes em diferentes computadores.

- Em Ativar comunicação SSL, digite sim.
- Em Armazenamento de chaves, insira a localização do arquivo de armazenamento de chaves.
- Em Senha do armazenamento de chaves, digite a senha do armazenamento de chaves.
- Em Habilitar a autenticação SSL do cliente, digite sim.
- Em Armazenamento confiável, insira a localização do arquivo de armazenamento confiável.
- Em Senha do armazenamento confiável, digite a senha do armazenamento confiável.

O programa de configuração atualiza o arquivo de configurações do agente de extração. O arquivo de configurações é chamado de `settings.properties`, e está localizado onde você instalou o agente de extração.

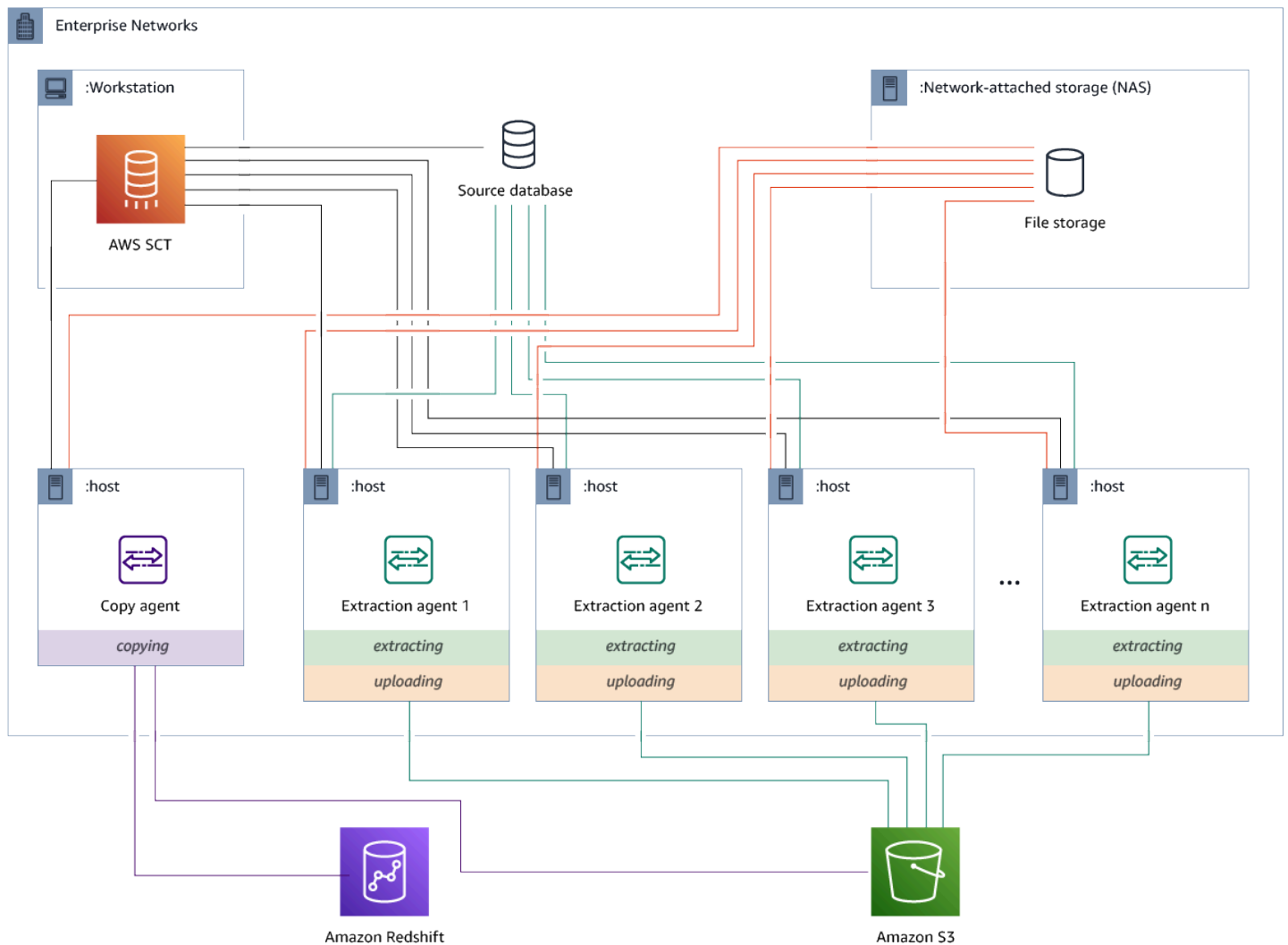
Veja a seguir um exemplo de arquivo de configurações.

```
$ 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 alterar as configurações, você pode editar o arquivo `settings.properties` usando um editor de texto ou executar a configuração do atendente novamente.

Como instalar e configurar atendentes de extração com atendentes de cópia dedicados

Você pode instalar atendentes de extração em uma configuração que tenha armazenamento compartilhado e um atendente de cópia dedicado. O diagrama a seguir ilustra esse cenário.



Essa configuração pode ser útil quando um servidor de banco de dados de origem suporta até 120 conexões e sua rede tem amplo armazenamento conectado. Use o procedimento a seguir para configurar atendentes de extração que possuem um atendente de cópia dedicado.

Para instalar e configurar atendentes de extração e um atendente de cópia dedicado

1. Certifique-se de que o diretório de trabalho de todos os atendentes de extração use a mesma pasta no armazenamento compartilhado.
2. Instale atendentes extratores seguindo as etapas em [Como instalar atendentes de extração](#).
3. Configure os atendentes de extração seguindo as etapas em [Como configurar atendentes de extração](#), mas especifique somente o driver JDBC de origem.
4. Configure um atendente de cópia dedicado seguindo as etapas em [Como configurar atendentes de extração](#), mas especifique somente um driver JDBC do Amazon Redshift.

Como iniciar atendentes de extração

Use o seguinte procedimento para iniciar agentes de extração. Repita esse procedimento em cada computador em que você tenha um agente de extração instalado.

Os agentes de extração atuam como ouvintes. Quando você inicia um agente com esse procedimento, o agente começa ouvindo para obter instruções. Você envia aos agentes instruções para extrair dados do seu data warehouse em uma seção posterior.

Para iniciar seu agente de extração

- No computador que tem o agente de extração instalado, execute o comando a seguir para seu sistema operacional.

Sistema operacional	Comando para iniciar
Microsoft Windows	Clique duas vezes no arquivo de lote <code>StartAgent.bat</code> .
RHEL	Execute o seguinte comando no caminho para a pasta em que você instalou o agente: <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	Execute o seguinte comando no caminho para a pasta em que você instalou o agente. Use o comando apropriado para a sua versão do Ubuntu. Ubuntu 14.04: <code>sudo initctl <i>start</i> sct-extractor</code> Ubuntu 15.04 e superior: <code>sudo systemctl <i>start</i> sct-extractor</code>

Para verificar o status do agente, execute o mesmo comando, mas substitua `start` por `status`.

Para interromper um agente, execute o mesmo comando, mas substitua `start` por `stop`.

Registrando agentes de extração com o AWS Schema Conversion Tool

Você gerencia seus agentes de extração usando AWS SCT. Os agentes de extração atuam como ouvintes. Quando recebem instruções de AWS SCT, eles extraem dados do seu data warehouse.

Use o procedimento a seguir para registrar agentes de extração em seu AWS SCT projeto.

Para registrar um agente de extração

1. Inicie AWS Schema Conversion Tool o e abra um projeto.
2. Abra o menu Exibir e selecione Visualização de migração de dados (outros). A guia Agentes é exibida. Se você já registrou atendentes, a AWS SCT os exibirá em uma grade na parte superior da guia.
3. Escolha Register.

Depois de registrar um agente em um AWS SCT projeto, você não pode registrar o mesmo agente em um projeto diferente. Se você não estiver mais usando um agente em um AWS SCT projeto, você pode cancelar o registro. Em seguida, você pode registrá-lo com um projeto diferente.

4. Selecione Atendente de dados do Redshift e depois OK.
5. Insira suas informações na guia Conexão da caixa de diálogo:
 - a. Em Descrição, insira uma descrição do atendente.
 - b. Em Nome do host, digite o nome do host ou o endereço IP do computador do atendente.
 - c. Em Porta, digite o número da porta em que o atendente está escutando.
 - d. Escolha Registrar para registrar o agente em seu AWS SCT projeto.
6. Repita as etapas anteriores para registrar vários agentes com seu projeto da AWS SCT .

Ocultar e recuperar informações para um agente AWS SCT

Um AWS SCT agente criptografa uma quantidade significativa de informações, por exemplo, senhas para repositórios confiáveis de chaves do usuário, contas de banco de dados, AWS informações de contas e itens similares. Ele faz isso usando um arquivo especial chamado `seed.dat`. Por padrão, o agente cria esse arquivo na pasta de trabalho do usuário que configura o agente primeiro.

Como usuários diferentes podem configurar e executar o agente, o caminho para `seed.dat` é armazenado no parâmetro `{extractor.private.folder}` do arquivo `settings.properties`. Quando o agente é iniciado, ele pode usar esse caminho para localizar o arquivo `seed.dat` para acessar as informações do armazenamento de confiança de chaves para o banco de dados em que ele funciona.

Você pode precisar recuperar as senhas que um agente armazenou nos seguintes casos:

- Se o usuário perder o `seed.dat` arquivo e a localização e a porta do AWS SCT agente não mudarem.
- Se o usuário perder o `seed.dat` arquivo e a localização e a porta do AWS SCT agente mudarem. Nesse caso, a alteração geralmente ocorre porque o agente foi migrado para outro host ou outra porta, e as informações no arquivo `seed.dat` não são mais válidas.

Nesses casos, se um agente for iniciado sem SSL, ele inicia e, em seguida, acessa o armazenamento do agente criado anteriormente. Em seguida, ele entra no estado `Waiting for recovery` (Aguardando recuperação).

No entanto, nesses casos, se um agente for iniciado com SSL, não será possível reiniciá-lo. Isso ocorre porque o agente não pode descriptografar as senhas para os certificados armazenados no arquivo `settings.properties`. Nesse tipo de inicialização, ocorre uma falha ao iniciar o agente. Um erro semelhante ao seguinte é gravado no log: "O agente não pôde ser iniciado com o modo SSL ativado. Reconfigure o agente. Motivo: a senha para o armazenamento de chaves está incorreta."

Para corrigir isso, crie um novo agente e configure-o para usar as senhas existentes para acessar os certificados SSL. Para fazer isso, use o procedimento a seguir.

Depois de executar esse procedimento, o agente deve ser executado e ir para o estado `Aguardando recuperação`. AWS SCT envia automaticamente as senhas necessárias para um agente no estado `Aguardando recuperação`. Quando o agente tem as senhas, ele reinicia as tarefas. Nenhuma outra ação do usuário é necessária por parte da AWS SCT .

Para reconfigurar o agente e restaurar senhas para acessar certificados SSL

1. Instale um novo AWS SCT agente e execute a configuração.
2. Altere a propriedade `agent.name` no arquivo `instance.properties` para o nome do agente para o qual o armazenamento foi criado, para que o novo agente funcione com o armazenamento de agente existente.

O arquivo `instance.properties` é armazenado na pasta privada do agente, que é nomeada usando a seguinte convenção: `{output.folder}\dmt\{hostName}_{portNumber}\`.

3. Altere o nome de `{output.folder}` para o da pasta de saída do agente anterior.

Neste momento, ainda AWS SCT está tentando acessar o extrator antigo no host e na porta antigos. Como resultado, o extrator inacessível obtém o status FAILED. Em seguida, você pode alterar o host e a porta.

4. Modifique o host, a porta ou ambos para o antigo agente usando o comando Modify para redirecionar o fluxo de solicitações para o novo agente.

Quando AWS SCT consegue fazer ping no novo agente, AWS SCT recebe o status Aguardando recuperação do agente. AWS SCT em seguida, recupera automaticamente as senhas do agente.

Cada agente que funciona com o armazenamento de agente atualiza um arquivo especial chamado `storage.lck` localizado em `{output.folder}\{agentName}\storage\`. Esse arquivo contém o ID de rede do agente e o tempo até que o armazenamento seja bloqueado. Quando o agente funciona com o armazenamento de agente, ele atualiza o arquivo `storage.lck` e amplia a locação do armazenamento em 10 minutos a cada 5 minutos. Nenhuma outra instância pode utilizar esse armazenamento de agente até que a locação expire.

Criação de regras de migração de dados no AWS SCT

Antes de extrair seus dados com o AWS Schema Conversion Tool, você pode configurar filtros que reduzam a quantidade de dados que você extrai. Você pode criar filtros de migração de dados, usando cláusulas WHERE para reduzir os dados a serem extraídos. Por exemplo, você pode gravar uma cláusula WHERE que seleciona dados de uma única tabela.

Você pode criar filtros de migração de dados e salvá-los como parte do seu projeto. Com seu projeto aberto, use o procedimento a seguir para criar regras de migração de dados.

Para criar regras de migração de dados

1. Abra o menu Exibir e selecione Visualização de migração de dados (outros).
2. Selecione Regras de migração de dados e, em seguida, selecione Adicionar nova regra.
3. Configure sua regra de migração de dados:
 - a. Em Nome, insira um nome para sua regra de migração.

- b. Em Onde o nome do esquema é como, digite um filtro para aplicar a esquemas. Neste filtro, uma cláusula WHERE é avaliada usando uma cláusula LIKE. Para escolher um esquema, insira o nome exato do esquema. Para escolher vários esquemas, use o caractere “%” como curinga para corresponder a qualquer número de caracteres no nome do esquema.
 - c. Em nome da tabela como, digite um filtro para aplicar às tabelas. Neste filtro, uma cláusula WHERE é avaliada usando uma cláusula LIKE. Para escolher uma tabela, insira um nome exato. Para escolher várias tabelas, use o caractere “%” como curinga para corresponder a qualquer número de caracteres no nome da tabela.
 - d. Em Onde cláusula, digite uma cláusula WHERE para filtrar os dados.
4. Depois de configurar seu filtro, selecione Salvar para salvar seu filtro ou Cancelar para cancelar as alterações.
 5. Ao finalizar a adição, edição e exclusão dos filtros, selecione Salvar tudo para salvar todas as alterações.

Para desativar um filtro sem excluí-lo, use o ícone de alternância. Para duplicar um filtro existente, use o ícone de cópia. Para excluir um filtro existente, use o ícone de exclusão. Para salvar as alterações feitas em seus filtros, selecione Salvar tudo.

Como alterar as configurações do extrator e da cópia das configurações do projeto

Na janela Configurações do projeto AWS SCT, você pode escolher as configurações dos agentes de extração de dados e do comando Amazon RedshiftCOPY.

Para escolher essas configurações, selecione Configurações, Configurações do projeto e, em seguida, Migração de dados. Aqui, você pode editar as Configurações de extração, as Configurações do Amazon S3 e as Configurações de cópia.

Use as instruções na tabela a seguir para fornecer as informações sobre as Configurações de extração.

Para este parâmetro	Faça o seguinte
Compression format (Formato de compactação)	Especifique o formato de compactação dos arquivos de entrada. Escolha uma das seguintes opções: GZIP, BZIP2, ZSTD ou Sem compressão.

Para este parâmetro	Faça o seguinte
Caractere delimitador	Especifique o caractere ASCII que separa os campos nos arquivos de entrada. Não há suporte para caracteres não imprimíveis.
Valor NULO como uma string	Ative essa opção se seus dados incluírem um terminador nulo. Se essa opção estiver desativada, o comando COPY do Amazon Redshift tratará nulo como o fim do registro e encerrará o processo de carregamento.
Estratégia de classificação	Use a classificação para reiniciar a extração a partir do ponto de falha. Escolha uma das seguintes estratégias de classificação: Use a classificação após a primeira falha (recomendado), Use a classificação, se possível, ou Nunca use a classificação. Para ter mais informações, consulte the section called “Como classificar dados” .
Esquema temporário de origem	Insira o nome do esquema no banco de dados de origem, onde o atendente de extração pode criar os objetos temporários.
Tamanho do arquivo de saída (em MB)	Insira o tamanho, em MB, dos arquivos carregados no Amazon S3.
Tamanho do arquivo Snowball out (em MB)	Insira o tamanho, em MB, dos arquivos enviados para AWS Snowball. Os arquivos podem ter de 1 a 1.000 MB de tamanho.
Usar o particionamento automático. Para Greenplum e Netezza, insira o tamanho mínimo das tabelas suportadas (em megabytes)	Ative essa opção para usar o particionamento de tabelas e, em seguida, insira o tamanho das tabelas a serem particionadas para os bancos de dados de origem Greenplum e Netezza. Para migrações do Oracle para o Amazon Redshift, você pode manter esse campo vazio porque AWS SCT cria subtarefas para todas as tabelas particionadas.

Para este parâmetro	Faça o seguinte
Extrair LOBs	Ative essa opção para extrair objetos grandes (LOBs) do seu banco de dados de origem. Os LOBs incluem BLOBs, CLOBs, NCLobs, arquivos XML e assim por diante. Para cada LOB, os atendentes de extração da AWS SCT criam um arquivo de dados.
Pasta de LOBs do bucket do Amazon S3	Insira a localização dos agentes AWS SCT de extração para armazenar LOBs.
Aplicar RTRIM às colunas de string	Ative essa opção para cortar um conjunto específico de caracteres do final das sequências extraídas.
Manter os arquivos localmente e após carregá-los para o Amazon S3	Ative essa opção para manter os arquivos em sua máquina local depois que os atendentes de extração de dados os enviarem para o Amazon S3.

Use as instruções na tabela a seguir para fornecer as informações sobre as Configurações do Amazon S3.

Para este parâmetro	Faça o seguinte
Usar proxy	Ative essa opção para usar um servidor proxy para fazer upload de dados para o Amazon S3. Em seguida, escolha o protocolo de transferência de dados, insira o nome do host, a porta, o nome do usuário e a senha.
Endpoint type	Selecione FIPS para usar o endpoint do Padrão Federal de Processamento de Informações (FIPS). Selecione VPCE para usar o endpoint da nuvem privada virtual (VPC). Em seguida, em endpoint da VPC, insira o Sistema de Nomes de Domínio (DNS) de seu endpoint da VPC.
Manter os arquivos no Amazon S3 depois de copiar para o Amazon Redshift	Ative essa opção para manter os arquivos extraídos no Amazon S3 depois de copiar esses arquivos para o Amazon Redshift.

Use as instruções na tabela a seguir para fornecer as informações para Copiar configurações.

Para este parâmetro	Faça o seguinte
Contagem máxima de erros	Insira o número de erros de carregamento. Depois que a operação atinge esse limite, os agentes de extração de AWS SCT dados encerram o processo de carregamento de dados. O valor padrão é 0, o que significa que os agentes de extração de AWS SCT dados continuam o carregamento de dados, independentemente das falhas.
Substituir caracteres UTF-8 não válidos	Ative essa opção para substituir caracteres UTF-8 não válidos pelo caractere especificado e continuar a operação de carregamento de dados.
Usar espaço em branco como valor nulo	Ative essa opção para carregar campos em branco que consistem em caracteres de espaço em branco como nulos.
Usar vazio como valor nulo	Ative essa opção para carregar campos vazios CHAR e VARCHAR como nulos.
Truncar colunas	Ative essa opção para truncar dados em colunas de acordo com a especificação do tipo de dados.
Compactação automática	Ative essa opção para aplicar a codificação de compactação durante uma operação de cópia.
Atualização automática de estatísticas	Ative essa opção para atualizar as estatísticas ao final de uma operação de cópia.
Verificar o arquivo antes de carregar	Ative essa opção para validar arquivos de dados antes de carregá-los no Amazon Redshift.

Classificando dados antes de migrar usando AWS SCT

Classificar seus dados antes da migração AWS SCT oferece alguns benefícios. Se você classificar os dados primeiro, AWS SCT poderá reiniciar o agente de extração no último ponto salvo após uma

falha. Além disso, se você estiver migrando dados para o Amazon Redshift e classificar os dados primeiro AWS SCT, poderá inseri-los no Amazon Redshift com mais rapidez.

Esses benefícios têm a ver com a forma como AWS SCT cria consultas de extração de dados. Em alguns casos, AWS SCT usa a função analítica `DENSE_RANK` nessas consultas. No entanto, o `DENSE_RANK` pode usar muito tempo e recursos do servidor para classificar o conjunto de dados resultante da extração. Portanto, se AWS SCT puder funcionar sem ele, funcionará.

Para classificar dados antes de migrar usando AWS SCT

1. Abra um AWS SCT projeto.
2. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e, em seguida, escolha Criar tarefa local.
3. Selecione a guia Avançado e em Estratégia de classificação, escolha uma opção:
 - Nunca usar classificação – O atendente de extração não usará a função analítica `DENSE_RANK` e reiniciará do começo se ocorrer uma falha.
 - Usar a classificação, se possível – O atendente de extração usará a função `DENSE_RANK` se a tabela tiver uma chave primária ou uma restrição exclusiva.
 - Usar a classificação após a primeira falha (recomendado) – Primeiro o atendente de extração tenta obter os dados sem usar a função `DENSE_RANK`. Se a primeira tentativa falhar, o agente de extração recriará a consulta usando a função `DENSE_RANK` e preservará sua localização em caso de falha.

The screenshot shows the 'Create Local task' dialog box with the 'Advanced' tab active. The 'Extraction settings' section includes:

- Delimiter character: |
- Compression format: GZIP
- NULL value as a string:
- Sorting strategy: Use sorting after first fail (recommen...)
- Source temp schema:
- Out file size (in MB): 10
- Apply RTRIM to string columns:
- Keep files locally after upload to AWS S3:
- Use subtasks auto-balancing between agents:
- Freezing interval: 10

The 'Copy settings' section includes:

- Maximum error count: 0
- Replace invalid UTF-8 character: ?
- Use blank as null value: BLANKSASNULL: This option loads blank fields, which consist of only white space characters, as NULL. The default behavior, without this option, is to load the space characters as is.
- Use empty as null value: EMPTYASNULL: This option indicates that Amazon Redshift should load empty CHAR and VARCHAR fields as NULL.
- Truncate columns: TRUNCATECOLUMNS: This option truncates data in columns to the appropriate number of characters so that it fits the column specification. This option applies only to columns with a VARCHAR or CHAR data type, and rows 4 MB or less in size.
- Automatic compression: COMPUPDATE: This option controls whether compression encodings are automatically

Buttons at the bottom: Test Task, Cancel, Create.

4. Defina parâmetros adicionais como descrito abaixo e, em seguida, escolha Criar para criar a tarefa de extração de dados.

Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados

Use os procedimentos a seguir para criar, executar e monitorar tarefas de extração de dados.

Para atribuir tarefas a agentes e migrar dados

1. No AWS Schema Conversion Tool, depois de converter seu esquema, escolha uma ou mais tabelas no painel esquerdo do seu projeto.

Você pode escolher todas as tabelas, mas não recomendamos isso por motivos de desempenho. Recomendamos que você crie várias tarefas para várias tabelas com base no tamanho das tabelas do seu data warehouse.

2. Abra o menu de contexto (clique com o botão direito do mouse) de cada tabela e selecione Criar tarefa. A caixa de diálogo Criar tarefa local é aberta.
3. Em Nome da tarefa, digite um nome para a tarefa.
4. Em Modo de migração, escolha uma das seguintes opções:
 - Extrair somente – Extraia os dados e salve-os em suas pastas de trabalho locais.
 - Extrair e carregar – Extraia seus dados e carregue-os para o Amazon S3.
 - Extrair, carregar e copiar – Extraia seus dados, carregue-os para o Amazon S3 e copie-os para seu data warehouse do Amazon Redshift.
5. Em Tipo de criptografia, selecione uma das seguintes opções:
 - NENHUMA – Desative a criptografia de dados em todo o processo de migração de dados.
 - CSE_SK – Use criptografia do lado do cliente com uma chave simétrica para migrar dados. A AWS SCT gera automaticamente as chaves de criptografia e as transmite aos atendentes de extração de dados usando Secure Sockets Layer (SSL). A AWS SCT não criptografa objetos grandes (LOBs) durante a migração de dados.
6. Escolha Extrair LOBs para extrair objetos grandes. Se você não precisar extrair objetos grandes, poderá desmarcar a caixa de seleção. Isso reduz a quantidade de dados extraídos.
7. Para ver informações detalhadas sobre uma tarefa, selecione Habilitar log de tarefas. Você pode usar o log de tarefas para depurar problemas.

Se você habilitar o log de tarefas, escolha o nível de detalhes que você deseja ver. Os níveis são os seguintes, com cada nível incluindo todas as mensagens do nível anterior:

- ERROR – A menor quantidade de detalhes.
- WARNING
- INFO
- DEBUG

- TRACE – A maior quantidade de detalhes.
8. Para exportar dados de BigQuery, AWS SCT use a pasta bucket do Google Cloud Storage. Nessa pasta, os atendentes de extração de dados armazenam seus dados de origem.

Para inserir o caminho para sua pasta de bucket do Google Cloud Storage, selecione Avançado. Para a pasta de bucket do Google CS, insira o nome do bucket e o nome da pasta.
 9. Para assumir uma função para o usuário do seu atendente de extração de dados, escolha as configurações do Amazon S3. Quanto ao perfil do IAM, insira o nome do perfil a ser usado. Em Região, escolha a Região da AWS para essa função.
 10. Selecione Testar tarefa para verificar se você pode se conectar à sua pasta de trabalho, bucket do Amazon S3 e data warehouse do Amazon Redshift. A verificação depende do modo de migração que você escolheu.
 11. Escolha Criar para criar a tarefa.
 12. Repita as etapas anteriores para criar tarefas para todos os dados que você deseja migrar.

Para executar e monitorar tarefas

1. Em Visualização, selecione Visualização de migração de dados. A guia Agentes é exibida.
2. Escolha a guia Tasks (Tarefas). Suas tarefas são exibidas na grade na parte superior, como mostrado a seguir. Você pode ver o status de uma tarefa na grade superior, e o status de suas subtarefas na grade inferior.

Name	Extract	Upload	Copy
+ CUSTOMER	0%		
+ LINEORDER_100K	0%		
+ LINEORDER_150K	0%		
+ LINEORDER_1M	0%		
LocalTask 2	100%	100%	
+ CUSTOMER	100%	100%	
+ LINEORDER_100K	100%	100%	
+ LINEORDER_150K	100%	100%	
LocalTask 3	100%	100%	0%
+ LINEORDER_100K	100%	100%	0%

Download log All migration Tasks Resume Stop Restart Reset Delete Replace Refresh all Refresh selected

Properties Processing details

Extracting _____

Data _____

100%

3. Escolha uma tarefa na grade superior e expanda-a. Dependendo do modo de migração que foi escolhido, você vê a tarefa dividida em Extrair, Fazer upload e Copiar.
4. Escolha Iniciar para iniciar uma tarefa. Você pode monitorar o status de suas tarefas ao mesmo tempo em que elas estão ativas. As subtarefas são executadas em paralelo. Extrair, fazer upload e copiar também funcionam em paralelo.
5. Se você habilitou o registro em log quando configurou a tarefa, será possível visualizar o log:
 - a. Selecione Fazer download do log. É exibida uma mensagem com o nome da pasta que contém o arquivo de log. Ignore a mensagem.
 - b. É exibido um link na guia Detalhes da tarefa. Escolha o link para abrir a pasta que contém o arquivo de log.

Você pode fechar AWS SCT e seus agentes e tarefas continuarão funcionando. Você pode reabrir AWS SCT mais tarde para verificar o status de suas tarefas e visualizar os registros de tarefas.

Você pode salvar tarefas de extração de dados em seu disco local e restaurá-las no mesmo projeto ou em outro projeto usando exportação e importação. Para exportar uma tarefa, verifique se você tem pelo menos uma tarefa de extração criada em um projeto. Você pode importar uma única tarefa de extração ou todas as tarefas criadas no projeto.

Quando você exporta uma tarefa de extração, AWS SCT cria um `.xml` arquivo separado para essa tarefa. O arquivo `.xml` armazena as informações de metadados dessa tarefa, como propriedades, descrição e subtarefas da tarefa. O arquivo `.xml` não contém informações sobre o processamento de uma tarefa de extração. Informações como as seguintes são recriadas quando a tarefa é importada:

- Progresso da tarefa
- Estados da subtarefa e do estágio
- Distribuição dos atendentes extratores por subtarefas e estágios
- IDs de tarefas e subtarefas
- Task name

Exportação e importação de uma tarefa de extração de AWS SCT dados

Você pode salvar rapidamente uma tarefa existente de um projeto e restaurá-la em outro projeto (ou no mesmo projeto) usando AWS SCT exportação e importação. Use o procedimento a seguir para exportar e importar tarefas de extração de dados.

Para exportar e importar uma tarefa de extração de dados

1. Em Visualização, selecione Visualização de migração de dados. A guia Agentes é exibida.
2. Escolha a guia Tasks (Tarefas). Suas tarefas estão listadas na grade que é exibida.
3. Selecione os três pontos alinhados verticalmente (ícone de reticências) localizados no canto inferior direito abaixo da lista de tarefas.
4. Selecione Exportar tarefa no menu pop-up.
5. Escolha a pasta na qual você deseja AWS SCT colocar o `.xml` arquivo de exportação da tarefa.

AWS SCT cria o arquivo de exportação da tarefa com um formato de nome de arquivo de **TASK-DESCRIPTION_TASK-ID.xml**.

6. Selecione os três pontos alinhados verticalmente (ícone de reticências) no canto inferior direito abaixo da lista de tarefas.
7. Selecione Importar tarefa no menu pop-up.

Você pode importar uma tarefa de extração para um projeto conectado ao banco de dados de origem, e o projeto tem pelo menos um atendente de extração registrado ativo.

8. Selecione o arquivo `.xml` para a tarefa de extração que você exportou.

AWS SCT obtém os parâmetros da tarefa de extração do arquivo, cria a tarefa e adiciona a tarefa aos agentes de extração.

9. Repita essas etapas para exportar e importar tarefas adicionais de extração de dados.

Ao final desse processo, sua exportação e importação estão concluídas e suas tarefas de extração de dados estão prontas para uso.

Extração de dados usando um dispositivo AWS Snowball Edge

O processo de uso AWS SCT do AWS Snowball Edge tem várias etapas. A migração envolve uma tarefa local, na qual AWS SCT usa um agente de extração de dados para mover os dados para o dispositivo AWS Snowball Edge e, em seguida, uma ação intermediária em que AWS copia os dados do dispositivo AWS Snowball Edge para um bucket do Amazon S3. O processo termina de AWS SCT carregar os dados do bucket do Amazon S3 para o Amazon Redshift.

As seções a seguir a essa visão geral fornecem um step-by-step guia para cada uma dessas tarefas. O procedimento pressupõe que você tenha AWS SCT instalado, configurado e registrado um agente de extração de dados em uma máquina dedicada.

Execute as etapas a seguir para migrar dados de um armazenamento de dados local para um armazenamento de AWS dados usando o AWS Snowball Edge.

1. Crie uma tarefa AWS Snowball do Edge usando o AWS Snowball console.
2. Desbloqueie o dispositivo AWS Snowball Edge usando a máquina Linux dedicada local.
3. Crie um novo projeto em AWS SCT.
4. Instale, configure e execute seus atendentes de extração de dados.
5. Crie e defina permissões para o bucket do Amazon S3 usar.
6. Importe um AWS Snowball trabalho para o seu AWS SCT projeto.
7. Registre seu atendente de extração de dados na AWS SCT.
8. Crie uma tarefa local em AWS SCT.

9. Execute e monitore a tarefa de migração de dados na AWS SCT.

tep-by-step Procedimentos S para migrar dados usando um AWS SCT Edge AWS Snowball

As seções a seguir apresentam informações detalhadas sobre as etapas de migração.

Etapa 1: criar uma tarefa AWS Snowball Edge

Crie um AWS Snowball trabalho seguindo as etapas descritas na seção [Criando um trabalho de AWS Snowball borda](#) no Guia do desenvolvedor do AWS Snowball Edge.

Etapa 2: desbloquear o dispositivo AWS Snowball Edge

Execute os comandos que desbloqueiam e fornecem credenciais para o dispositivo Snowball Edge a partir da máquina em que você instalou AWS DMS o agente. Ao executar esses comandos, você pode ter certeza de que a chamada do AWS DMS agente se conecta ao dispositivo AWS Snowball Edge. Para obter mais informações sobre como desbloquear o dispositivo AWS Snowball Edge, consulte [Desbloquear](#) o Snowball Edge.

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

Etapa 3: criar um novo AWS SCT projeto

Em seguida, crie um novo AWS SCT projeto.

Para criar um novo projeto no AWS SCT

1. Inicie AWS Schema Conversion Tool o. No menu Arquivo, selecione Novo projeto. A caixa de diálogo Novo projeto é exibida.
2. Digite um nome para o projeto, que é armazenado localmente no computador.
3. Digite o local do arquivo do projeto local.
4. Escolha OK para criar seu AWS SCT projeto.
5. Escolha Adicionar fonte para adicionar um novo banco de dados de origem ao seu AWS SCT projeto.
6. Escolha Adicionar destino para adicionar uma nova plataforma de destino ao seu AWS SCT projeto.

7. Escolha o esquema do banco de dados de origem no painel esquerdo.
8. No painel direito, especifique a plataforma de banco de dados de destino para o esquema de origem selecionado.
9. Selecione Criar mapeamento. Esse botão fica ativo depois que você escolhe o esquema do banco de dados de origem e a plataforma do banco de dados de destino.

Etapa 4: instalar, configurar e executar seu atendente de extração de dados

AWS SCT usa um agente de extração de dados para migrar dados para o Amazon Redshift. O arquivo.zip que você baixou para instalar AWS SCT inclui o arquivo instalador do agente de extração. Você pode instalar o atendente de extração de dados no Windows, Red Hat Enterprise Linux ou Ubuntu. Para obter mais informações, consulte [Como instalar atendentes de extração](#).

Para configurar seu atendente de extração de dados, insira seus mecanismos de banco de dados de origem e destino. Além disso, certifique-se de ter baixado os drivers JDBC para seus bancos de dados de origem e destino no computador em que você executa o atendente de extração de dados. Os atendentes de extração de dados usam esses drivers para se conectar aos bancos de dados de origem e de destino. Para ter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

No Windows, o instalador do atendente de extração de dados inicia o assistente de configuração na janela do prompt de comando. No Linux, execute o arquivo `sct-extractor-setup.sh` do local onde você instalou o atendente.

Etapa 5: Configurar AWS SCT para acessar o bucket do Amazon S3

Para obter informações sobre como configurar um bucket do Amazon S3, consulte [Visão geral dos buckets](#) no Guia do usuário do Amazon Simple Storage Service.

Etapa 6: importar um AWS Snowball trabalho para seu AWS SCT projeto

Para conectar seu AWS SCT projeto ao seu dispositivo AWS Snowball Edge, importe seu AWS Snowball trabalho.

Para importar seu AWS Snowball trabalho

1. Abra o menu Configurações e selecione Configurações globais. A caixa de diálogo Configurações globais é exibida.

2. Selecione perfis de serviço da AWS e, em seguida, selecione Importar trabalho.
3. Escolha seu AWS Snowball emprego.
4. Insira seu IP do AWS Snowball . Para obter mais informações, consulte [Como alterar seu Endereço IP](#) no Guia do usuário do AWS Snowball .
5. Entre no seu AWS Snowball porto. Para obter mais informações, consulte [Portas necessárias para usar AWS serviços em um dispositivo AWS Snowball Edge](#) no Guia do desenvolvedor do AWS Snowball Edge.
6. Insira sua chave de acesso do AWS Snowball e a chave secreta do AWS Snowball . Para obter mais informações, consulte [Autorização e controle de acesso no AWS Snowball](#) no Guia do usuário do AWS Snowball .
7. Escolha Apply e, em seguida, escolha OK.

Etapa 7: registrar um agente de extração de dados no AWS SCT

Nesta seção, você registra o atendente de extração de dados na AWS SCT.

Para registrar um atendente de extração de dados

1. Abra o menu Exibir, selecione Visualização de migração de dados (outros) e depois selecione Registrar.
2. Em Descrição, insira um nome para seu atendente de extração de dados.
3. Em Nome do host, insira o endereço IP do computador em que você executa o atendente de extração de dados.
4. Em Porta, insira a porta de escuta que você configurou.
5. Escolha Register.

Etapa 8: como criar uma tarefa local

Em seguida, você cria a tarefa de migração. A tarefa inclui duas subtarefas. Uma subtarefa migra os dados do banco de dados de origem para o equipamento AWS Snowball Edge. A outra subtarefa leva os dados que o dispositivo carrega em um bucket do Amazon S3 e os migra para o banco de dados de destino.

Para criar uma tarefa de migração

1. No menu Exibir e selecione Visualização de migração de dados (outros).

2. No painel esquerdo que exibe o schema do banco de dados de origem, escolha um objeto de schema para migrar. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e, em seguida, escolha Criar tarefa local.
3. Em Nome da tarefa, insira um nome descritivo para a tarefa de migração de dados.
4. Para o Modo de migração, escolha Extrair, carregar e copiar.
5. Escolha Configurações do Amazon S3.
6. Selecione Usar Snowball.
7. Insira pastas e subpastas em seu bucket do Amazon S3 onde o atendente de extração de dados pode armazenar dados.
8. Escolha Criar para criar a tarefa.

Etapa 9: Executar e monitorar a tarefa de migração de dados no AWS SCT

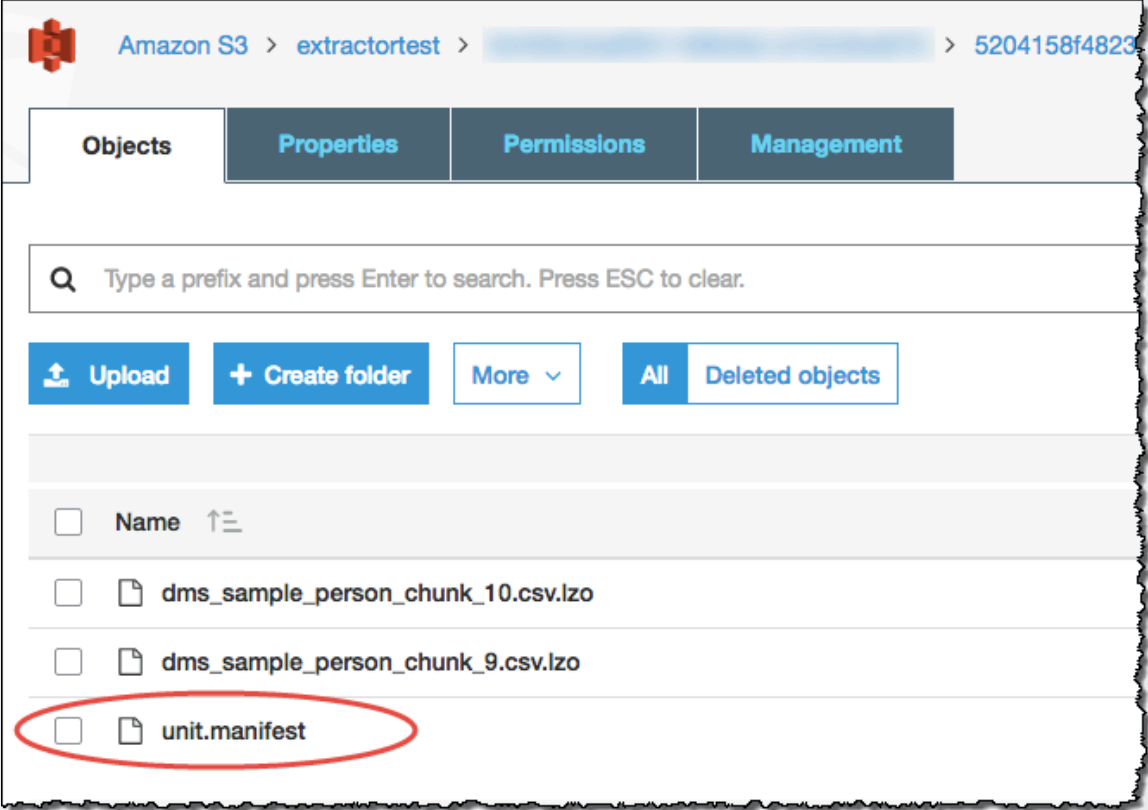
Para iniciar sua tarefa de migração de dados, selecione Iniciar. Certifique-se de ter estabelecido conexões com o banco de dados de origem, o bucket do Amazon S3, o AWS Snowball dispositivo, bem como a conexão com o banco de dados de destino em AWS.

Você pode monitorar e gerenciar as tarefas de migração de dados e suas subtarefas na guia Tarefas. Você pode ver o progresso da migração de dados, bem como pausar ou reiniciar suas tarefas de migração de dados.

Saída da tarefa de extração de dados

Após a conclusão de suas tarefas de migração, seus dados estão prontos. Use as informações seguintes para determinar como proceder com base no modo de migração escolhido e a localização de seus dados.

Modo de migração	Local dos dados
Extrair, fazer upload e copiar	Os dados já estão em seu data warehouse do Amazon Redshift. Você pode verificar se os dados estão lá e começar a usá-los. Para obter mais informações, consulte Conexão com clusters a partir de ferramentas do cliente e de código .
Extrair e fazer upload	Os atendentes de extração salvaram os dados como arquivos em seu bucket do Amazon S3. Você pode usar o comando COPIAR do Amazon Redshift para

Modo de migração	Local dos dados
	<p>carregar dados no Amazon Redshift. Para obter mais informações, consulte Como carregar dados do Amazon S3 na documentação do Amazon Redshift.</p> <p>Há várias pastas em seu bucket do Amazon S3 correspondentes às tarefas de extração que você configurou. Quando você carregar os dados para o Amazon Redshift, especifique o nome do arquivo de manifesto criado por cada tarefa. O arquivo de manifesto é exibido na pasta da tarefa no seu bucket do Amazon S3, conforme mostrado a seguir.</p>  <p>The screenshot shows the Amazon S3 console interface. At the top, the breadcrumb navigation reads 'Amazon S3 > extractortest > [redacted] > 5204158f4823'. Below this are four tabs: 'Objects', 'Properties', 'Permissions', and 'Management'. A search bar contains the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', 'Create folder', 'More', 'All', and 'Deleted objects'. A table of objects is displayed with columns for selection, name, and sorting. The objects listed are 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' entry is circled in red.</p>
Extrair somente	<p>Os agentes de extração salvam os dados como arquivos em sua pasta de trabalho. Copie manualmente os dados para o seu bucket do Amazon S3 e, em seguida, prossiga com as instruções para Extrair e carregar.</p>

Usando particionamento virtual com AWS Schema Conversion Tool

Muitas vezes, você pode gerenciar melhor tabelas grandes não particionadas criando subtarefas que criam partições virtuais dos dados de tabela usando regras de filtragem. Em AWS SCT, você pode criar partições virtuais para seus dados migrados. Há três tipos de partição que funcionam com tipos de dados específicos:

- O tipo de partição RANGE funciona com tipos de dados numéricos e de data e hora.
- O tipo de partição LIST funciona com tipos de dados numéricos, de caracteres e de data e hora.
- O tipo de partição DATE AUTO SPLIT funciona com tipos de dados numéricos, de data e hora.

AWS SCT valida os valores que você fornece para criar uma partição. Por exemplo, se você tentar particionar uma coluna com o tipo de dados NUMERIC, mas fornecer valores de um tipo de dados diferente, AWS SCT gerará um erro.

Além disso, se você estiver usando AWS SCT para migrar dados para o Amazon Redshift, você pode usar o particionamento nativo para gerenciar a migração de tabelas grandes. Para ter mais informações, consulte [Como usar o particionamento nativo](#).

Limites ao criar um particionamento virtual

Estas são as limitações para a criar uma partição virtual:

- Você pode usar o particionamento virtual apenas para tabelas não particionadas.
- Você pode usar o particionamento virtual apenas na visualização de migração de dados.
- Você não pode usar a opção UNION ALL VIEW com particionamento virtual.

Tipo de Partição RANGE

O tipo de partição RANGE particiona dados com base em um intervalo de valores de coluna para tipos de dados numéricos e de data e hora. Esse tipo de partição cria uma cláusula WHERE, e você fornece o intervalo de valores para cada partição. Para especificar uma lista de valores para a coluna particionada, use a caixa Valores. Você pode carregar informações de valor usando um arquivo .csv.

O tipo de partição RANGE cria partições padrão nas duas extremidades dos valores da partição. Essas partições padrão capturam quaisquer dados menores ou maiores que os valores de partição especificados.

Por exemplo, você pode criar várias partições com base em um intervalo de valores fornecido. No exemplo a seguir, os valores de particionamento para LO_TAX são especificados para criar várias partições.

```
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 criar uma partição virtual RANGE

1. Aberto AWS SCT.
2. Selecione o modo Visualização de migração de dados (outros).
3. Escolha a tabela na qual você deseja configurar o particionamento virtual. Abra o menu de contexto (clique com o botão direito do mouse) da tabela e selecione Adicionar particionamento virtual.
4. Na caixa de diálogo Adicionar particionamento virtual, insira as informações conforme a seguir.

Opção	Ação
Tipo de partição	Escolha RANGE. A caixa de diálogo da interface do usuário muda de acordo com tipo de partição escolhida.
Nome da coluna	Escolha a coluna que você deseja particionar.
Tipo de coluna	Escolha o tipo de dados para os valores na coluna.
Valores	Adicione novos valores digitando cada valor na caixa Novo valor e, em seguida, escolhendo o sinal de adição para acrescentar o valor.
Carregar do arquivo	(Opcional) Digite o nome de um arquivo .csv que contém os valores de partição.

5. Escolha OK.

Tipo de partição LIST

O tipo de partição LIST particiona dados com base nos valores de coluna para tipos de dados numéricos, de caractere e de data e hora. Esse tipo de partição cria uma cláusula WHERE, e você fornece os valores para cada partição. Para especificar uma lista de valores para a coluna particionada, use a caixa Valores. Você pode carregar informações de valor usando um arquivo .csv.

Por exemplo, você pode criar várias partições com base em um valor fornecido. No exemplo a seguir, os valores de particionamento para LO_ORDERKEY são especificados para criar várias partições.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Você também pode criar uma partição padrão para valores não incluídos nas partições especificadas.

Você pode usar o tipo de partição LIST para filtrar os dados de origem se quiser excluir valores específicos da migração. Por exemplo, suponha que você queira omitir as linhas com LO_ORDERKEY = 4. Nesse caso, não inclua o valor 4 na lista de valores de partição e certifique-se de que a opção Incluir outros valores não esteja selecionada.

Para criar uma partição virtual LIST

1. Aberto AWS SCT.
2. Selecione o modo Visualização de migração de dados (outros).
3. Escolha a tabela na qual você deseja configurar o particionamento virtual. Abra o menu de contexto (clique com o botão direito do mouse) da tabela e selecione Adicionar particionamento virtual.
4. Na caixa de diálogo Adicionar particionamento virtual, insira as informações conforme a seguir.

Opção	Ação
Tipo de partição	Selecione LIST. A caixa de diálogo da interface do usuário muda de acordo com tipo de partição escolhida.

Opção	Ação
Nome da coluna	Escolha a coluna que você deseja particionar.
Novo valor	Digite aqui um valor para adicioná-lo ao conjunto de valores de particionamento.
Incluir outros valores	Escolha essa opção para criar uma partição padrão em que todos os valores que não estiverem de acordo com critérios de particionamento são armazenados.
Carregar do arquivo	(Opcional) Digite o nome de um arquivo .csv que contém os valores de partição.

5. Escolha OK.

Tipo de partição DATE AUTO SPLIT

O tipo de partição DATE AUTO SPLIT é uma forma automatizada de gerar partições RANGE. Com DATA AUTO SPLIT, você informa AWS SCT o atributo de particionamento, onde começar e terminar e o tamanho do intervalo entre os valores. Em seguida, a AWS SCT calcula os valores da partição automaticamente.

O DATA AUTO SPLIT automatiza grande parte do trabalho envolvido na criação de partições de intervalo. A compensação entre o uso dessa técnica e o particionamento de intervalos é quanto controle você precisa sobre os limites da partição. O processo de divisão automática sempre cria intervalos de tamanhos iguais (uniformes). O particionamento de intervalo permite que você varie o tamanho de cada intervalo conforme necessário para sua distribuição de dados específica. Por exemplo, você pode usar diariamente, semanalmente, quinzenalmente, mensalmente e assim por diante.

```
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 criar uma partição virtual DATE AUTO SPLIT

1. Aberto AWS SCT.
2. Selecione o modo Visualização de migração de dados (outros).
3. Escolha a tabela na qual você deseja configurar o particionamento virtual. Abra o menu de contexto (clique com o botão direito do mouse) da tabela e selecione Adicionar particionamento virtual.
4. Na caixa de diálogo Adicionar particionamento virtual, insira as informações conforme a seguir.

Opção	Ação
Tipo de partição	Escolha DATE AUTO SPLIT. A caixa de diálogo da interface do usuário muda de acordo com tipo de partição escolhida.
Nome da coluna	Escolha a coluna que você deseja particionar.
Data de início	Digite uma data de início.
Data de término	Digite uma data de término.
Interval	Digite a unidade de intervalo e escolha o valor para essa unidade.

5. Escolha OK.

Como usar o particionamento nativo

Para acelerar a migração de dados, seus agentes de extração de dados podem usar partições nativas de tabelas no servidor de data warehouse de origem. AWS SCT oferece suporte ao particionamento nativo para migrações do Greenplum, Netezza e Oracle para o Amazon Redshift.

Por exemplo, depois de criar um projeto, você pode coletar estatísticas sobre um esquema e analisar o tamanho das tabelas selecionadas para migração. Para tabelas que excedem o tamanho especificado, AWS SCT aciona o mecanismo de particionamento nativo.

Para usar o particionamento nativo

1. Abra AWS SCT e escolha Novo projeto para Arquivo. A caixa de diálogo Novo projeto é exibida.

2. Crie um novo projeto, adicione seus servidores de origem e de destino e crie regras de mapeamento. Para ter mais informações, consulte [Criando um projeto da AWS SCT](#).
3. Escolha Exibir e, em seguida, Visualização principal.
4. Em Configurações do projeto, selecione a guia Migração de dados. Selecione Usar o particionamento automático. Para bancos de dados de origem Greenplum e Netezza, insira o tamanho mínimo das tabelas suportadas em megabytes (por exemplo, 100). A AWS SCT cria automaticamente subtarefas de migração separadas para cada partição nativa que não está vazia. Para migrações do Oracle para o Amazon Redshift, AWS SCT cria subtarefas para todas as tabelas particionadas.
5. No painel esquerdo que exibe o esquema do banco de dados de origem, escolha um esquema. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Coletar estatísticas. Quanto à migração de dados da Oracle para o Amazon Redshift, você pode ignorar esta etapa.
6. Selecione todas as tabelas para migrar.
7. Registre o número necessário de atendentes. Para ter mais informações, consulte [Registrando agentes de extração com o AWS Schema Conversion Tool](#).
8. Crie uma tarefa de extração de dados para as tabelas selecionadas. Para ter mais informações, consulte [Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados](#).

Verifique se tabelas grandes estão divididas em subtarefas e se cada subtarefa corresponde ao conjunto de dados que apresenta uma parte da tabela localizada em uma fatia em seu data warehouse de origem.

9. Inicie e monitore o processo de migração até que AWS SCT os agentes de extração de dados concluam a migração dos dados de suas tabelas de origem.

Como migrar LOBs para o Amazon Redshift

O Amazon Redshift não oferece suporte ao armazenamento de objetos binários grandes (LOBs). No entanto, se você precisar migrar uma ou mais LOBs para o Amazon Redshift AWS SCT, poderá realizar a migração. Para fazer isso, a AWS SCT usa um bucket do Amazon S3 para armazenar os LOBs e grava o URL para o bucket do Amazon S3 para os dados migrados armazenados no Amazon Redshift.

Para migrar LOBs para o Amazon Redshift

1. Abra um AWS SCT projeto.

2. Conecte-se aos bancos de dados de origem e de destino. Atualize os metadados do banco de dados de destino e certifique-se de que as tabelas convertidas existam lá.
3. Em **Ações**, selecione **Criar tarefa local**.
4. Em **Modo de migração**, escolha uma das seguintes opções:
 - **Extrair e carregar para extrair seus dados e carregá-los para o Amazon S3.**
 - **Extrair, carregar e copiar para extrair seus dados, carregá-los para o Amazon S3 e copiá-los para seu data warehouse do Amazon Redshift.**
5. Escolha **Configurações do Amazon S3**.
6. Em **Pasta de LOBs no bucket do Amazon S3**, digite o nome da pasta em um bucket do Amazon S3 onde você deseja que os LOBs sejam armazenados.

Se você usa perfil AWS de serviço, esse campo é opcional. AWS SCT pode usar as configurações padrão do seu perfil. Para usar outro bucket do Amazon S3, insira o caminho aqui.

7. Ative essa opção **Usar proxy** para usar um servidor proxy para carregar os dados para o Amazon S3. Em seguida, escolha o protocolo de transferência de dados, insira o nome do host, a porta, o nome do usuário e a senha.
8. Para o **Tipo de endpoint**, selecione **FIPS** para usar o endpoint do Padrão Federal de Processamento de Informações (FIPS). Selecione **VPCE** para usar o endpoint da nuvem privada virtual (VPC). Em seguida, em **endpoint da VPC**, insira o Sistema de Nomes de Domínio (DNS) de seu endpoint da VPC.
9. Ative a opção **Manter arquivos no Amazon S3 após copiar para o Amazon Redshift** para manter os arquivos extraídos no Amazon S3 depois de copiar esses arquivos para o Amazon Redshift.
10. Escolha **Criar** para criar a tarefa.

Melhores práticas e solução de problemas para atendentes de extração de dados

Veja a seguir algumas sugestões de práticas recomendadas e de solução de problemas para o uso de agentes de extração.

Problema	Sugestões de solução de problemas
O desempenho está lento	Para melhorar o desempenho, recomendamos o seguinte:

Problema	Sugestões de solução de problemas
	<ul style="list-style-type: none">• Instale vários agentes.• Instale agentes em computadores próximos ao seu data warehouse.• Não execute todas as tabelas em uma única tarefa de agente.
Atrasos de contenção	Evite ter muitos agentes acessando seu data warehouse ao mesmo tempo.
Um agente é desativado temporariamente	Se um agente estiver desativado, o status de cada uma de suas tarefas aparece como falha na AWS SCT. Se você esperar, em alguns casos, o agente pode se recuperar. Neste caso, o status de suas tarefas é atualizado na AWS SCT.
Um agente é desativado permanentemente	<p>Se o computador que executa um agente for desativado permanentemente, e se o agente estiver executando uma tarefa, você poderá substituir um novo agente para continuar a tarefa. Você pode substituir um novo agente somente se a pasta de trabalho do agente original não estava no mesmo computador que o agente original. Para substituir um novo agente, faça o seguinte:</p> <ul style="list-style-type: none">• Instale um agente em um novo computador.• Configure o novo agente com as mesmas configurações, incluindo o número da porta e a pasta de trabalho, que o agente original.• Inicie o agente. Depois que o agente é iniciado, a tarefa descobre o novo agente disponível e continua sendo executada no novo agente.

Como converter o aplicativo SQL usando a AWS SCT

Ao converter o esquema do seu banco de dados de um mecanismo para outro, é preciso também atualizar o código SQL nos seus aplicativos, a fim de interagir com o novo mecanismo de banco de dados, em vez do antigo. Você pode visualizar, analisar, editar e salvar o código SQL convertido.

Você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter o código SQL em C++, C#, Java, ou outro código de aplicativo. Para uma conversão do Oracle para o PostgreSQL, você pode usar a AWS SCT para converter o código SQL*Plus em PSQL. Além disso, para uma conversão de Oracle para PostgreSQL, você pode usar a AWS SCT para converter o código SQL incorporado em aplicativos C#, C++, Java e Pro*C.

Tópicos

- [Visão geral da conversão de aplicativos SQL](#)
- [Como converter o código SQL em seus aplicativos com a AWS SCT](#)
- [Como converter o código SQL em aplicativos C# com a AWS SCT](#)
- [Como converter o código SQL em aplicativos C++ com a AWS SCT](#)
- [Como converter o código SQL em aplicativos Java com a AWS SCT](#)
- [Como converter o código SQL em aplicativos Pro*C com a AWS SCT](#)

Visão geral da conversão de aplicativos SQL

Para converter o código SQL em seu aplicativo, siga as etapas de alto nível a seguir:

- Crie um projeto de conversão de aplicativos – O projeto de conversão de aplicativos é filho do projeto de conversão de esquema do banco de dados. Cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de aplicativos de conversão do esquema de banco de dados filho. Para obter mais informações, consulte [Como criar projetos genéricos de conversão de aplicativos em na AWS SCT](#).
- Analise e converta o código SQL – A AWS SCT analisa seu aplicativo, extrai o código SQL e cria uma versão local do SQL convertido para você analisar e editar. A ferramenta não altera o código em seu aplicativo até que você esteja pronto. Para obter mais informações, consulte [Como analisar e converter seu código SQL na AWS SCT](#).
- Crie um relatório de avaliação do aplicativo – O relatório de avaliação do aplicativo fornece informações importantes sobre a conversão do código do aplicativo SQL do seu esquema de

banco de dados de origem para o esquema de banco de dados de destino. Para obter mais informações, consulte [Como criar e usar o relatório de avaliação da AWS SCT na AWS SCT](#).

- Edite, aplique alterações e salve o código SQL convertido – O relatório de avaliação inclui uma lista de itens de códigos SQL que não podem ser convertidos automaticamente. Para esses itens, é possível editar o código SQL manualmente para executar a conversão. Para obter mais informações, consulte [Como editar e gravar seu código SQL convertido com a AWS SCT](#).

Como converter o código SQL em seus aplicativos com a AWS SCT

Você pode usar AWS SCT para converter o código SQL incorporado em seus aplicativos. O conversor genérico do aplicativo da AWS SCT trata o código do aplicativo como texto simples. Ele escaneia o código do seu aplicativo e extrai o código SQL com expressões regulares. Esse conversor suporta diferentes tipos de arquivos de código fonte e funciona com código de aplicativo escrito em qualquer linguagem de programação.

O conversor de aplicativo genérico tem as limitações a seguir. Ele não se aprofunda na lógica do aplicativo que é específica para a linguagem de programação do seu aplicativo. Além disso, o conversor genérico não oferece suporte a instruções SQL de diferentes objetos do aplicativo, como funções, parâmetros, variáveis locais e assim por diante.

Para melhorar a conversão de código SQL do seu aplicativo, use conversores de código SQL de aplicativo específicos da linguagem. Para obter mais informações, consulte [Como converter o código SQL em aplicativos C#](#), [Como converter o código SQL em aplicativos Java](#) e [Como converter o código SQL em aplicativos Pro*C](#).

Como criar projetos genéricos de conversão de aplicativos em na AWS SCT

Na AWS Schema Conversion Tool, o projeto de conversão do aplicativo é filho do projeto de conversão do esquema de banco de dados. Cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de aplicativos de conversão do esquema de banco de dados filho.

Note

A AWS SCT não oferece suporte à conversão entre as seguintes fontes e destinos:

- Oracle para Oracle
- PostgreSQL para PostgreSQL ou Aurora PostgreSQL
- MySQL para MySQL
- SQL Server para SQL Server
- Amazon Redshift para (Amazon Redshift)
- SQL Server para Babelfish
- SQL Server Integration Services para AWS Glue
- Apache Cassandra para o Amazon DynamoDB

Use o procedimento a seguir para criar um projeto genérico de conversão de aplicativos.

Para criar um projeto de conversão de aplicativos

1. Na AWS Schema Conversion Tool, selecione Novo aplicativo genérico no menu Aplicativos.

A caixa de diálogo Novo projeto de conversão de aplicativo será exibida.

Creating a generic application conversion project

Enter the name, location and type of the new application conversion project.

Name: Application conversion project 1

Location: C:\AWS-SCT-Demo Browse

Language: Java Target parameter style: Same as in source

Settings

Don't cast bind variables to SQL types i

Keep object names i

Choose the source database schema that your application uses which is mapped with the target tree object:

- ▼ Schemas [58]
 - ANONYMOUS
 - APPQOSSYS
 - AUDSYS
 - CHINOOK**
 - CTXSYS
 - DVSYs

OK Cancel

2. Adicione as seguintes informações de projeto.

Para este parâmetro	Faça o seguinte
Nome	Digite um nome para o seu projeto de conversão de aplicativos. Cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de conversão de aplicativos filho, portanto, escolha um nome que faça sentido se você adicionar mais projetos mais tarde.
Local	Digite o local do código-fonte referente ao aplicativo.
Linguagem	Escolha uma das seguintes opções: <ul style="list-style-type: none"> • Java • C++

Para este parâmetro	Faça o seguinte
	<ul style="list-style-type: none"> • C# • Quaisquer
Estilo do parâmetro de destino	<p>Escolha a sintaxe a ser usada para vincular variáveis no código convertido. Plataformas de banco de dados diferentes usam sintaxe diferente para variáveis de vinculação. Escolha uma das seguintes opções:</p> <ul style="list-style-type: none"> • O mesmo da origem • Posicional (?) • Indexado (:1) • Indexado (\$1) • Nomeadas (@ nome) • Nomeadas (: nome) • Nomeadas (&nome) • Nomeadas (\$nome) • Nomeadas (#nome) • Nomeadas (!nome!)
Escolha o esquema do banco de dados de origem	Na árvore de origem, escolha o esquema que seu aplicativo usa. Certifique-se de que esse esquema faça parte de uma regra de mapeamento.

3. Selecione Não converter variáveis de ligação em tipos SQL para evitar a conversão de tipos de variáveis de ligação em tipos SQL. Essa opção só está disponível para uma conversão de Oracle para PostgreSQL.

Por exemplo, o código fonte do aplicativo inclui a seguinte consulta Oracle:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Quando você seleciona Não converter variáveis de ligação em tipos SQL, a AWS SCT converte essa consulta conforme mostrado a seguir.

```
SELECT * FROM account WHERE id = ?
```

Quando você desmarca Não converter variáveis de ligação em tipos SQL, a AWS SCT altera o tipo de variável de associação para o tipo de dados NUMERIC. O resultado da conversão é mostrado a seguir.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

4. Selecione Manter nomes de objetos para evitar adicionar o nome do esquema ao nome do objeto convertido. Essa opção só está disponível para uma conversão de Oracle para PostgreSQL.

Por exemplo, suponha que o código do seu aplicativo fonte inclua a seguinte consulta Oracle.

```
SELECT * FROM ACCOUNT
```

Quando você seleciona Manter nomes de objetos, a AWS SCT converte essa consulta conforme mostrado a seguir.

```
SELECT * FROM account
```

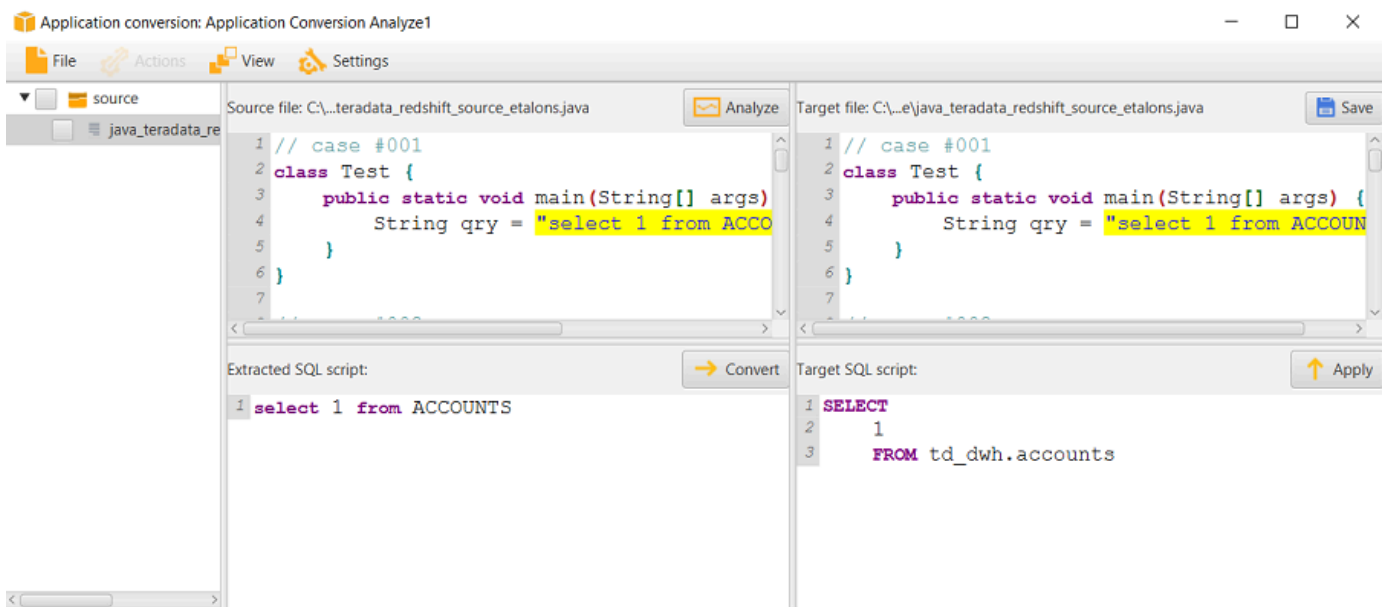
Quando você desmarca Manter nomes de objetos, a AWS SCT adiciona o nome do esquema ao nome da tabela. O resultado da conversão é mostrado a seguir.

```
SELECT * FROM schema_name.account
```

Se o código fonte incluir os nomes dos objetos principais nos nomes dos objetos, a AWS SCT usa esse formato no código convertido. Nesse caso, ignore a opção Manter nomes de objetos porque a AWS SCT adiciona os nomes dos objetos principais no código convertido.

5. Selecione OK para criar um projeto de conversão de aplicativos.

A janela do projeto será aberta.



Como gerenciar projetos de conversão de aplicativos na AWS SCT

Você pode abrir um projeto de conversão de aplicativos existente e adicionar vários projetos de conversão de aplicativos.

Depois que você criar um projeto de conversão de aplicativo, a janela de projeto do projeto de conversão se abrirá automaticamente. Você pode fechar a janela do projeto de conversão do aplicativo e voltar a ela mais tarde.

Para abrir um projeto de conversão de aplicativos existente

1. No painel esquerdo, escolha o nó do projeto de conversão do aplicativo e abra o menu de contexto (clique com o botão direito do mouse).
2. Escolha Gerenciar aplicativo.

Para adicionar um projeto adicional de conversão de aplicativos

1. No painel esquerdo, escolha o nó do projeto de conversão do aplicativo e abra o menu de contexto (clique com o botão direito do mouse).
2. Escolha New application (Nova aplicação).
3. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos. Para obter mais informações, consulte [Como criar projetos genéricos de conversão de aplicativos](#).

Como analisar e converter seu código SQL na AWS SCT

Use o procedimento a seguir para analisar e converter seu código SQL na AWS Schema Conversion Tool.

Para analisar e converter seu código SQL

1. Abra um projeto de conversão de aplicativo existente e selecione Analisar.

A AWS SCT analisa o código do aplicativo e extrai o código SQL. A AWS SCT exibe o código SQL extraído na lista de Scripts SQL analisados.

2. Para Scripts SQL analisados, escolha um item para revisar o código SQL extraído. A AWS SCT exibe o código do item selecionado no painel Script SQL extraído.
3. Escolha Converter para converter o código SQL no painel Script SQL extraído. A AWS SCT converte o código em um formato compatível com seu banco de dados de destino.

Você pode editar o código SQL convertido. Para obter mais informações, consulte [Como editar e gravar seu código SQL convertido](#).

Source file	Position	Extracted code	Converted code
C:\And...s.java	Line 11 22:50	✓ "select 1 a from ACCOUNTS b"	✓ SELECT 1 AS a FROM td_dwh.accounts AS b
C:\And...s.java	Line 18 22:46	✓ "select * from ACCOUNTS"	
C:\And...s.java	Line 25 22:50	✓ "select a.* from ACCOUNTS a"	

4. Quando você cria um relatório de avaliação de conversão de aplicativos, a AWS SCT converte todos os itens de código SQL extraídos. Para obter mais informações, consulte [Como criar e usar o relatório de avaliação](#).

Como criar e usar o relatório de avaliação da AWS SCT na AWS SCT

O relatório de avaliação da conversão do aplicativo fornece informações sobre a conversão do código SQL do aplicativo em um formato compatível com seu banco de dados de destino. O relatório detalha todo o código SQL extraído, todo o código SQL convertido e itens de ação para o código SQL que a AWS SCT não pode converter.

Como criar um relatório de avaliação de conversão de aplicativos

Use o procedimento a seguir para criar um relatório de avaliação de conversão de aplicativos.

Para criar um relatório de avaliação de conversão de aplicativos

1. Na janela do projeto de conversão do aplicativo, selecione Criar relatório no menu Ações.

A AWS SCT cria o relatório de avaliação da conversão do aplicativo e o abre na janela do projeto de conversão do aplicativo.

2. Analise a guia Resumo.

A guia Resumo mostrada a seguir exibe as informações de resumo do relatório de avaliação de aplicativos. Ela mostra os códigos SQL que foram convertidos automaticamente e os itens que não foram convertidos automaticamente.



3. Selecione Ações de extração de SQL.

Examine a lista de itens de código SQL que a AWS SCT não pode extrair do seu código fonte.

4. Selecione Ações de conversão de SQL.

Examine a lista de itens de código SQL que a AWS SCT não pode converter automaticamente. Use as ações recomendadas para converter manualmente o código SQL. Para obter informações sobre como converter o código SQL, consulte [Como editar e gravar seu código SQL convertido com a AWS SCT](#).

5. (Opcional) Salve uma cópia local do relatório como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão do aplicativo.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter o código SQL.

Como editar e gravar seu código SQL convertido com a AWS SCT

O relatório de avaliação inclui uma lista de itens do código SQL que a AWS SCT não pode converter. Para cada item, a AWS SCT cria um item de ação na guia de Ações de conversão de SQL. Para esses itens, é possível editar o código SQL manualmente para executar a conversão.

Use o procedimento a seguir para editar seu código de SQL convertido, aplicar as alterações e, em seguida, salvá-los.

Para editar, aplicar alterações e salvar o código SQL convertido

1. Edite seu código de SQL convertido diretamente no painel Script SQL de destino. Se não houver códigos convertidos mostrados, você pode clicar no painel e começar a digitar.
2. Após concluir as edições do SQL convertido do seu código, escolha Aplicar. Neste momento, as alterações serão salvas na memória, mas ainda não serão gravados em seu arquivo.
3. Selecione Salvar para salvar as alterações no arquivo.

Ao selecionar Salvar, o arquivo original será substituído. Faça uma cópia do arquivo original antes de salvar, para que você tenha um registro de seu código de aplicativo original.

Como converter o código SQL em aplicativos C# com a AWS SCT

Para uma conversão de Oracle para PostgreSQL, você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter o código SQL incorporado em aplicativos C#. Esse conversor de aplicativos C# específico compreende a lógica do aplicativo. Ele coleta instruções que estão

localizadas em diferentes objetos do aplicativo, como perfis, parâmetros, variáveis locais e assim por diante.

Devido a essa análise profunda, o conversor de código SQL do aplicativo C# fornece melhores resultados de conversão do que o conversor genérico.

Como criar projetos de conversão de aplicativos C# na AWS SCT

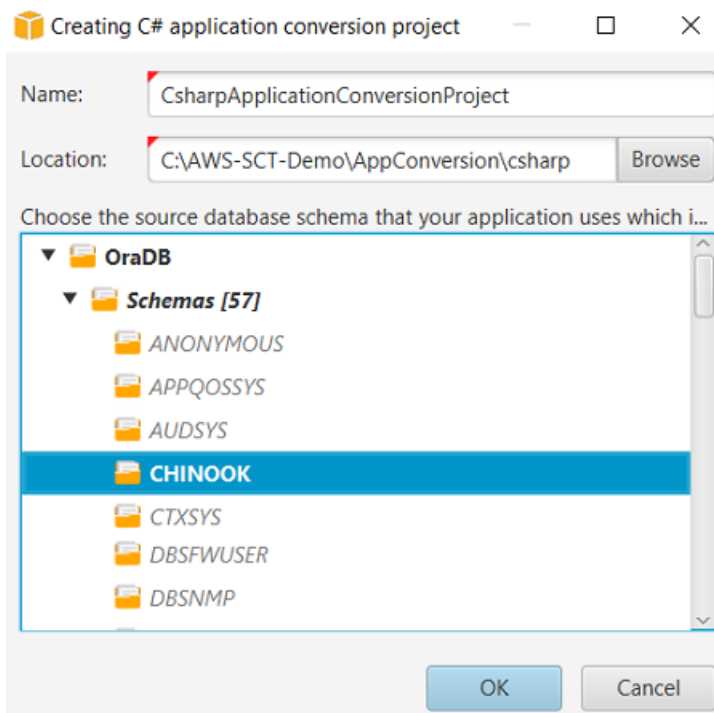
Você pode criar um projeto de conversão de aplicativos C# somente para converter esquemas de banco de dados Oracle em esquemas de banco de dados PostgreSQL. Certifique-se de adicionar uma regra de mapeamento em seu projeto que inclua um esquema Oracle de origem e um banco de dados PostgreSQL de destino. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).

Você pode adicionar vários projetos de conversão de aplicativos em um único projeto de AWS SCT. Use o procedimento a seguir para criar um projeto de conversão de aplicativos C#.

Para criar um projeto de conversão de aplicativos C#

1. Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#) e [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).
2. Adicione uma regra de mapeamento que inclua seu banco de dados Oracle de origem e um banco de dados PostgreSQL de destino. Você pode adicionar um banco de dados PostgreSQL de destino ou usar uma plataforma de banco de dados de destino PostgreSQL virtual em uma regra de mapeamento. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#) e [Usar alvos virtuais](#).
3. No menu Exibir, selecione Visualização principal.
4. No menu Aplicativos, selecione Novo aplicativo C#.

A caixa de diálogo Como criar um projeto de conversão de aplicativos C# será exibida.



5. Em Nome, digite um nome para o seu projeto de conversão de aplicativos C#. Como cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de conversão de aplicativos filho, escolha um nome que faça sentido se você adicionar vários projetos.
6. Em Local, digite o local do código fonte referente ao aplicativo.
7. Na árvore de origem, escolha o esquema que seu aplicativo usa. Certifique-se de que esse esquema faça parte de uma regra de mapeamento. A AWS SCT destaca os esquemas que fazem parte de uma regra de mapeamento em negrito.
8. Selecione OK para criar um projeto de conversão de aplicativos C#.
9. Encontre seu projeto de conversão de aplicativos C# no nó Aplicativos no painel esquerdo.

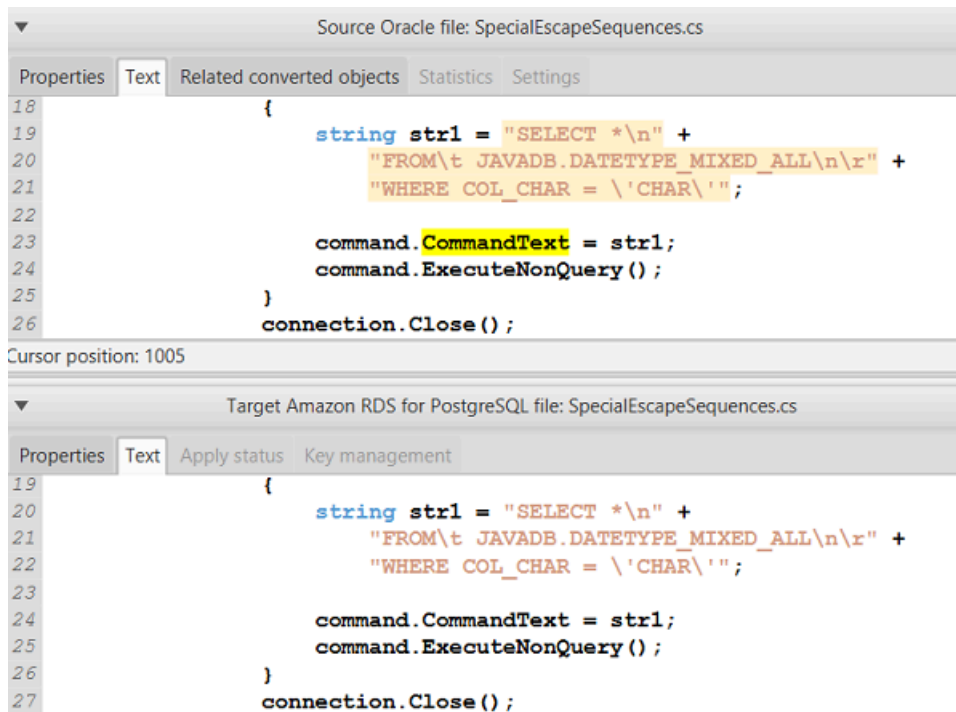
Como converter o código SQL em seu aplicativo C# na AWS SCT

Depois de adicionar seu aplicativo C# ao projeto AWS SCT, converta o código SQL desse aplicativo em um formato compatível com sua plataforma de banco de dados de destino. Use o procedimento a seguir para analisar e converter o código SQL incorporado em seu aplicativo C# na AWS Schema Conversion Tool.

Para converter seu código SQL

1. Expanda o nó C# em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter. A AWS SCT analisa seus arquivos de código fonte, determina a lógica do aplicativo e carrega os metadados do código no projeto. Esses metadados de código incluem classes, objetos, métodos, variáveis globais, interfaces em C# e assim por diante.

No painel do banco de dados de destino, a AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido.



```
Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18      {
19          string str1 = "SELECT *\n" +
20                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
21                      "WHERE COL_CHAR = \'CHAR\''";
22
23          command.CommandText = str1;
24          command.ExecuteNonQuery();
25      }
26      connection.Close();

Cursor position: 1005

Target Amazon RDS for PostgreSQL file: SpecialEscapeSequences.cs
Properties Text Apply status Key management
19      {
20          string str1 = "SELECT *\n" +
21                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
22                      "WHERE COL_CHAR = \'CHAR\''";
23
24          command.CommandText = str1;
25          command.ExecuteNonQuery();
26      }
27      connection.Close();
```

4. Salve o código do seu aplicativo convertido. Para obter mais informações, consulte [Como salvar o código do aplicativo convertido](#).

Seus aplicativos C# podem incluir código SQL que interage com diferentes bancos de dados de origem. Você pode migrar para o PostgreSQL vários desses bancos de dados de origem. Nesse caso, certifique-se de não converter o código SQL que interage com bancos de dados que você excluiu do escopo da migração. Você pode excluir os arquivos de origem do seu aplicativo C# do escopo da conversão. Para fazer isso, desmarque as caixas de seleção dos nomes dos arquivos que você deseja excluir do escopo da conversão.

Depois de alterar o escopo da conversão, a AWS SCT ainda analisa o código SQL de todos os arquivos de origem de seus aplicativos C#. Em seguida, a AWS SCT copia para a pasta de destino todos os arquivos de origem que você excluiu do escopo da conversão. Essa operação possibilita a criação do aplicativo depois de salvar os arquivos convertidos do aplicativo.

Como salvar o código do aplicativo convertido com a AWS SCT

Use o procedimento a seguir para salvar o código do seu aplicativo convertido.

Para salvar o código do seu aplicativo convertido

1. Expanda o nó C# em Aplicativos no painel do banco de dados de destino.
2. Escolha seu aplicativo convertido e selecione Salvar.
3. Insira o caminho para a pasta para salvar o código do aplicativo convertido e escolha Selecionar pasta.

Como gerenciar projetos de conversão de aplicativos C# na AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos C#, atualizar o código do aplicativo no projeto da AWS SCT ou remover um projeto de conversão em C# do seu projeto da AWS SCT.

Para adicionar um projeto adicional de conversão de aplicativos C#

1. Expanda o nó Aplicativos no painel esquerdo.
2. Escolha o nó C# e abra o menu de contexto (clique com o botão direito).
3. Escolha New application (Nova aplicação).
4. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos C#. Para obter mais informações, consulte [Como criar projetos de conversão de aplicativos C#](#).

Depois de fazer alterações no código fonte do aplicativo, carregue-o no projeto da AWS SCT.

Para carregar o código do aplicativo atualizado

1. Expanda o nó C# em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser atualizado e abra o menu de contexto (clique com o botão direito).
3. Selecione Atualizar e, em seguida, selecione Sim.

A AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações no código que você fez na AWS SCT e os resultados da conversão, crie um novo projeto de conversão em C#.

Para remover um projeto de conversão de aplicativos C#

1. Expanda o nó C# em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser removido e abra o menu de contexto (clique com o botão direito).
3. Selecione Excluir e OK.

Como criar um relatório de avaliação de conversão de aplicativos C# na AWS SCT

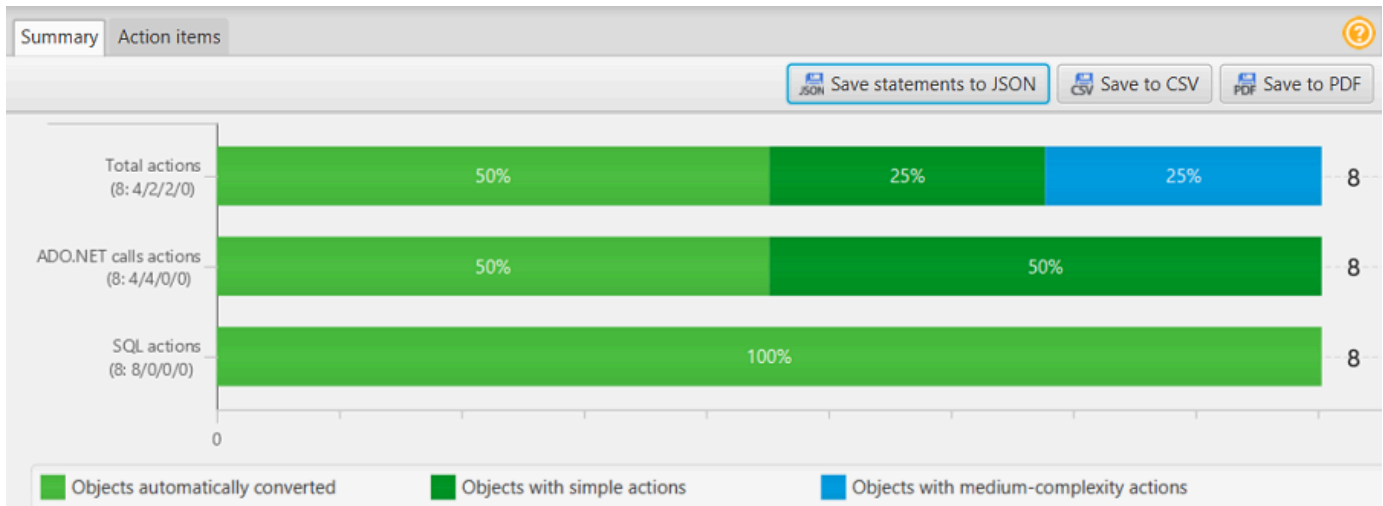
O Relatório de avaliação da conversão do aplicativo C# fornece informações sobre como converter o código SQL incorporado em seu aplicativo C# para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui também inclui itens de ação para código SQL que a AWS SCT não pode converter.

Use o procedimento a seguir para criar um relatório de avaliação de conversão de aplicativos C#.

Para criar um relatório de avaliação de conversão de aplicativos C#

1. Expanda o nó C# em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter.
4. No menu Exibir, selecione Visualização do relatório de avaliação.
5. Visualize a guia Resumo.

A guia Resumo mostrada a seguir exibe as informações do resumo executivo do relatório de avaliação de aplicativos C#. Ele mostra os resultados da conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte.



6. Selecione Salvar instruções em JSON para salvar o código SQL extraído do seu aplicativo C# como um arquivo JSON.
7. (Opcional) Salve uma cópia local do relatório como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão do aplicativo.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter o código SQL.

Como converter o código SQL em aplicativos C++ com a AWS SCT

Para uma conversão de Oracle para PostgreSQL, você pode usar a AWS SCT para converter o código SQL incorporado em aplicativos C++. Esse conversor de aplicativos C++ específico compreende a lógica do aplicativo. Ele coleta instruções que estão localizadas em diferentes objetos do aplicativo, como perfis, parâmetros, variáveis locais e assim por diante.

Devido a essa análise profunda, o conversor de código SQL do aplicativo C++ fornece melhores resultados de conversão do que o conversor genérico.

Como criar projetos de conversão de aplicativos C++ na AWS SCT

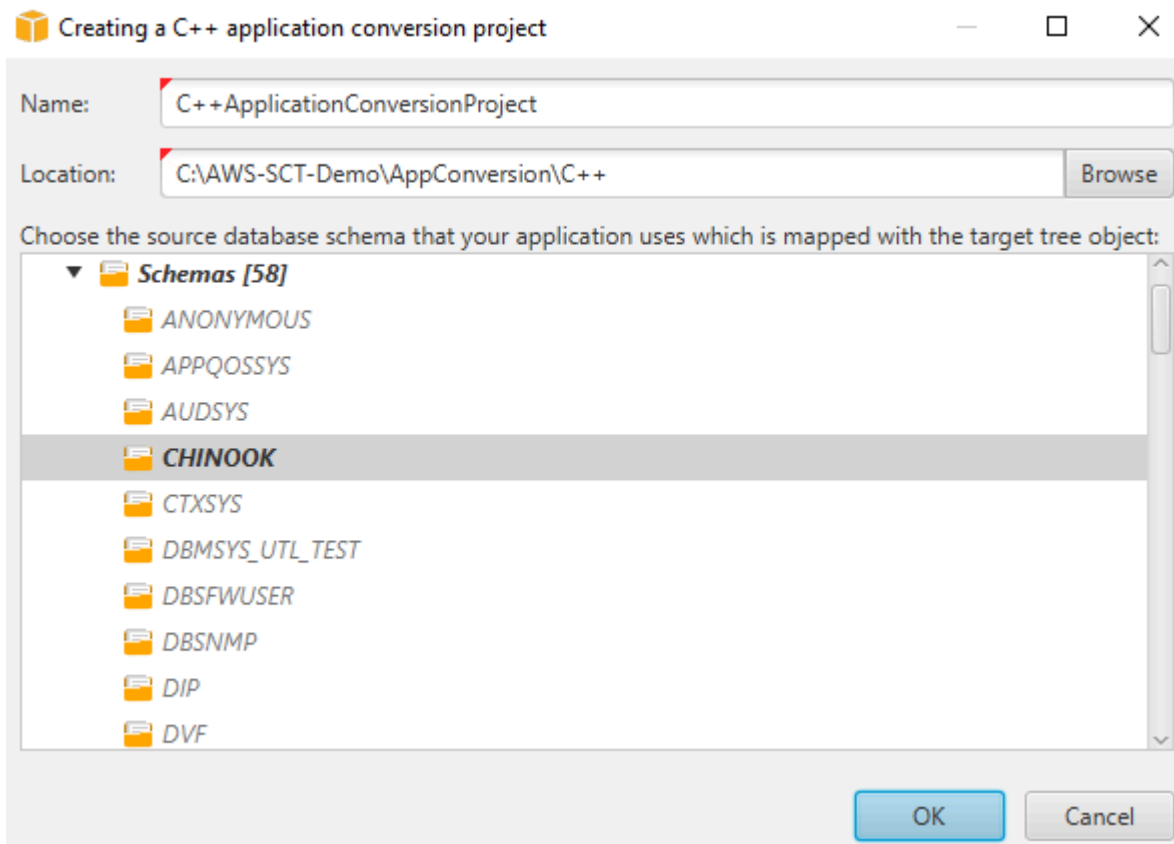
Você pode criar um projeto de conversão de aplicativos C++ somente para converter esquemas de banco de dados Oracle em esquemas de banco de dados PostgreSQL. Certifique-se de adicionar uma regra de mapeamento em seu projeto que inclua um esquema Oracle de origem e um banco de dados PostgreSQL de destino. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).

Você pode adicionar vários projetos de conversão de aplicativos em um único projeto de AWS SCT.

Para criar um projeto de conversão de aplicativos C++

1. Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#) e [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).
2. Adicione uma regra de mapeamento que inclua seu banco de dados Oracle de origem e um banco de dados PostgreSQL de destino. Você pode adicionar um banco de dados PostgreSQL de destino ou usar uma plataforma de banco de dados de destino PostgreSQL virtual em uma regra de mapeamento. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#) e [Usar alvos virtuais](#).
3. No menu Exibir, selecione Visualização principal.
4. No menu Aplicativos, selecione Novo aplicativo C++.

A caixa de diálogo Como criar um projeto de conversão de aplicativos C++ será exibida.



5. Em Nome, digite um nome para o seu projeto de conversão de aplicativos C++. Como cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de conversão de aplicativos filho, escolha um nome que faça sentido se você adicionar vários projetos.
6. Em Local, digite o local do código fonte referente ao aplicativo.
7. Na árvore de origem, escolha o esquema que seu aplicativo usa. Certifique-se de que esse esquema faça parte de uma regra de mapeamento. A AWS SCT destaca os esquemas que fazem parte de uma regra de mapeamento em negrito.
8. Selecione OK para criar um projeto de conversão de aplicativos C++.
9. Encontre seu projeto de conversão de aplicativos C++ no nó Aplicativos no painel esquerdo.

Como converter o código SQL em seu aplicativo C++ na AWS SCT

Depois de adicionar seu aplicativo C++ ao projeto AWS SCT, converta o código SQL desse aplicativo em um formato compatível com sua plataforma de banco de dados de destino. Use o procedimento a seguir para analisar e converter o código SQL incorporado em seu aplicativo C++ na AWS SCT.

Para converter seu código SQL

1. Expanda o nó C++ em Aplicativos no painel esquerdo e escolha o aplicativo a ser convertido.
2. No Projeto do aplicativo Oracle de origem, selecione Configurações. Revise e edite as configurações de conversão do aplicativo C++ selecionado. Você também pode especificar as configurações de conversão para todos os aplicativos C++ que você adicionou ao seu projeto da AWS SCT. Para obter mais informações, consulte [Como gerenciar projetos de conversão de aplicativos C++](#).
3. Em Tipo de compilador, selecione o compilador que você usa para o código fonte do seu aplicativo C++. A AWS SCT suporta os seguintes compiladores C++: Microsoft Visual C++, GCC, o GNU Compiler Collection e o Clang. A opção padrão é o Microsoft Visual C++.
4. Em Macros definidas pelo usuário, insira o caminho para o arquivo que inclui macros definidas pelo usuário do seu projeto em C++. Certifique-se de que esse arquivo tenha a seguinte estrutura: `#define name value`. No exemplo anterior, `value` é um parâmetro opcional. O valor padrão para este parâmetro opcional é 1.

Para criar esse arquivo, abra seu projeto no Microsoft Visual Studio e selecione Projeto, Propriedades, C/C++ e Pré-processador. Em Definições de pré-processador, selecione Editar e copie nomes e valores para um novo arquivo de texto. Em seguida, para cada string no arquivo, adicione o seguinte prefixo: `#define` .

5. Em Diretórios de inclusão externos, insira os caminhos para as pastas que incluem bibliotecas externas que você usa em seu projeto C++.
6. No painel esquerdo, selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
7. Selecione Converter. A AWS SCT analisa seus arquivos de código fonte, determina a lógica do aplicativo e carrega os metadados do código no projeto. Esses metadados de código incluem classes, objetos, métodos, variáveis globais, interfaces em C++ e assim por diante.

No painel do banco de dados de destino, a AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido, como mostrado a seguir.



```
Source Oracle file: StringInitialization.cpp
44     if ((dRet == SQLDriverConnect(hDBC, NULL, lpConnectionStr, connectionStr.size(), OutConnStr, 0xFF
45     {
46         SQLHANDLE hSelectStm = NULL;
47
48         if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDBC, &hSelectStm)) == SQL_SUCCESS)
49         {
50
51             char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
52             strncpy_s(&buff[0], 0xFF, "SELECT JAVADB.GET_INT() FROM DUAL", 18);
53
54             if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
55             {
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
26
```

Como gerenciar projetos de conversão de aplicativos C++ na AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos C++, editar configurações de conversão, atualizar o código do aplicativo C++ ou remover um projeto de conversão C++ do seu projeto da AWS SCT.

Para adicionar um projeto adicional de conversão de aplicativos C++

1. Expanda o nó Aplicativos no painel esquerdo.
2. Escolha o nó C++ e abra o menu de contexto (clique com o botão direito).
3. Escolha New application (Nova aplicação).
4. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos C++. Para obter mais informações, consulte [Como criar projetos de conversão de aplicativos C++](#).

Você pode especificar as configurações de conversão para todos os projetos de conversão de aplicativos C++ em seu projeto da AWS SCT.

Para editar as configurações de conversão para todos os aplicativos C++

1. No menu Configurações, selecione Configurações do projeto e, em seguida, selecione Conversão de aplicativos.
2. Em Tipo de compilador, selecione o compilador que você usa para o código fonte do seu aplicativo C++. A AWS SCT suporta os seguintes compiladores C++: Microsoft Visual C++, GCC, o GNU Compiler Collection e o Clang. A opção padrão é o Microsoft Visual C++.
3. Em Macros definidas pelo usuário, insira o caminho para o arquivo que inclui macros definidas pelo usuário do seu projeto em C++. Certifique-se de que esse arquivo tenha a seguinte estrutura: `#define name value`. No exemplo anterior, `value` é um parâmetro opcional. O valor padrão para este parâmetro opcional é 1.

Para criar esse arquivo, abra seu projeto no Microsoft Visual Studio e selecione Projeto, Propriedades, C/C++ e Pré-processador. Em Definições de pré-processador, selecione Editar e copie nomes e valores para um novo arquivo de texto. Em seguida, para cada string no arquivo, adicione o seguinte prefixo: `#define` .

4. Em Diretórios de inclusão externos, insira os caminhos para as pastas que incluem bibliotecas externas que você usa em seu projeto C++.
5. Selecione OK para salvar as configurações do projeto e fechar a janela.

Ou você pode especificar configurações de conversão para cada projeto de conversão de aplicativos C++. Para obter mais informações, consulte [Como converter o código SQL em seu aplicativo C++](#).

Depois de fazer alterações no código fonte do aplicativo, carregue-o no projeto da AWS SCT.

Para carregar o código do aplicativo atualizado

1. Expanda o nó C++ em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser atualizado e abra o menu de contexto (clique com o botão direito).
3. Selecione Atualizar e, em seguida, selecione Sim.

A AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações no código que você fez na AWS SCT e os resultados da conversão, crie um novo projeto de conversão em C++.

Além disso, a AWS SCT remove as configurações de conversão do aplicativo que você especificou para o aplicativo selecionado. Depois de carregar o código atualizado do aplicativo, a AWS SCT aplica os valores padrão das configurações do projeto.

Para remover um projeto de conversão de aplicativos C++

1. Expanda o nó C++ em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser removido e abra o menu de contexto (clique com o botão direito).
3. Selecione Excluir e OK.

Como criar um relatório de avaliação de conversão de aplicativos C++ na AWS SCT

O Relatório de avaliação da conversão do aplicativo C++ fornece informações sobre como converter o código SQL incorporado em seu aplicativo C++ para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui também inclui itens de ação para código SQL que a AWS SCT não pode converter.

Para criar um relatório de avaliação de conversão de aplicativos C++

1. Expanda o nó C++ em Aplicativos no painel esquerdo.

2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter.
4. No menu Exibir, selecione Visualização do relatório de avaliação.
5. Visualize a guia Resumo.

A guia Resumo exibe as informações do resumo executivo do relatório de avaliação de aplicativos C++. Ele mostra os resultados da conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte.

6. Selecione Salvar instruções em JSON para salvar o código SQL extraído do seu aplicativo Java como um arquivo JSON.
7. (Opcional) Salve uma cópia local do relatório como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão do aplicativo.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter o código SQL.

Como converter o código SQL em aplicativos Java com a AWS SCT

Para uma conversão de Oracle para PostgreSQL, você pode usar a AWS Schema Conversion Tool para converter o código SQL incorporado em aplicativos Java. Esse conversor de aplicativos Java específico compreende a lógica do aplicativo. Ele coleta instruções que estão localizadas em diferentes objetos do aplicativo, como perfis, parâmetros, variáveis locais e assim por diante.

Devido a essa análise profunda, o conversor de código SQL do aplicativo Java fornece melhores resultados de conversão do que o conversor genérico.

Se seu aplicativo Java usa a estrutura MyBatis para interagir com bancos de dados, você pode usar a AWS SCT para converter instruções SQL incorporadas aos arquivos e anotações XML do MyBatis.

Para entender a lógica dessas instruções SQL, a AWS SCT usa o arquivo de configuração MyBatis. A AWS SCT pode descobrir automaticamente esse arquivo na pasta do aplicativo, ou você pode inserir o caminho para esse arquivo manualmente.

Como criar projetos de conversão de aplicativos Java na AWS SCT

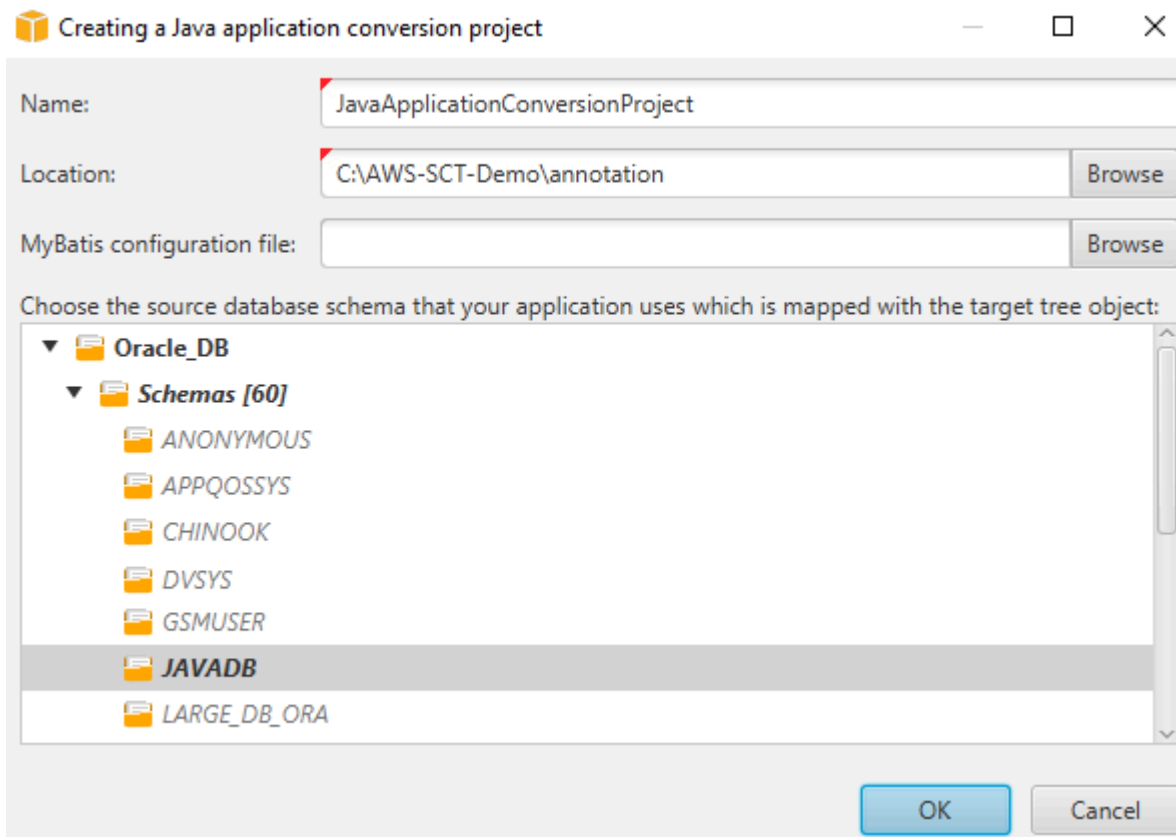
Você pode criar um projeto de conversão de aplicativos Java somente para converter esquemas de banco de dados Oracle em esquemas de banco de dados PostgreSQL. Certifique-se de adicionar uma regra de mapeamento em seu projeto que inclua um esquema Oracle de origem e um banco de dados PostgreSQL de destino. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).

Você pode adicionar vários projetos de conversão de aplicativos em um único projeto de AWS SCT. Use o procedimento a seguir para criar um projeto de conversão de aplicativos Java.

Para criar um projeto de conversão de aplicativos Java

1. Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#) e [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).
2. Adicione uma regra de mapeamento que inclua seu banco de dados Oracle de origem e um banco de dados PostgreSQL de destino. Você pode adicionar um banco de dados PostgreSQL de destino ou usar uma plataforma de banco de dados de destino PostgreSQL virtual em uma regra de mapeamento. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#) e [Usar alvos virtuais](#).
3. No menu Exibir, selecione Visualização principal.
4. No menu Aplicativos, selecione Novo aplicativo Java.

A caixa de diálogo Como criar um projeto de conversão de aplicativos Java será exibida.



5. Em Nome, digite um nome para o seu projeto de conversão de aplicativos Java. Como cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de conversão de aplicativos filho, escolha um nome que faça sentido se você adicionar vários projetos.
6. Em Local, digite o local do código fonte referente ao aplicativo.
7. (Opcional) Para o Arquivo de configuração MyBatis, insira o caminho para o arquivo de configuração MyBatis. A AWS SCT escaneia a pasta do aplicativo para descobrir esse arquivo automaticamente. Se esse arquivo não estiver localizado na pasta do aplicativo ou se você usar vários arquivos de configuração, insira o caminho manualmente.
8. Na árvore de origem, escolha o esquema que seu aplicativo usa. Certifique-se de que esse esquema faça parte de uma regra de mapeamento. A AWS SCT destaca os esquemas que fazem parte de uma regra de mapeamento em negrito.
9. Selecione OK para criar um projeto de conversão de aplicativos Java.
10. Encontre seu projeto de conversão de aplicativos Java no nó Aplicativos no painel esquerdo.

Como converter o código SQL em seu aplicativo Java na AWS SCT

Depois de adicionar seu aplicativo Java ao projeto AWS SCT, converta o código SQL desse aplicativo em um formato compatível com sua plataforma de banco de dados de destino. Use o procedimento a seguir para analisar e converter o código SQL incorporado em seu aplicativo Java na AWS Schema Conversion Tool.

Para converter seu código SQL

1. Expanda o nó Java em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter. A AWS SCT analisa seus arquivos de código fonte, determina a lógica do aplicativo e carrega os metadados do código no projeto. Esses metadados de código incluem classes, objetos, métodos, variáveis globais, interfaces em Java e assim por diante.

No painel do banco de dados de destino, a AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido.

```
Source Oracle file: CallMethod2.java
13 private final String USER = "min_privs";
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM JAVADB.DATATYPE_MIXED_AL
26     cs.execute();
27 }
28 }
Cursor position: 697

Target Amazon RDS for PostgreSQL file: CallMethod2.java
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26     cs.execute();
27 }
28 }
29 }
```

4. Salve o código do seu aplicativo convertido. Para obter mais informações, consulte [Como salvar o código do aplicativo convertido](#).

Seus aplicativos Java podem incluir código SQL que interage com diferentes bancos de dados de origem. Você pode migrar para o PostgreSQL vários desses bancos de dados de origem. Nesse caso, certifique-se de não converter o código SQL que interage com bancos de dados que você excluiu do escopo da migração. Você pode excluir os arquivos de origem do seu aplicativo Java do escopo da conversão. Para fazer isso, desmarque as caixas de seleção dos nomes dos arquivos que você deseja excluir do escopo da conversão.

Depois de alterar o escopo da conversão, a AWS SCT ainda analisa o código SQL de todos os arquivos de origem de seus aplicativos Java. Em seguida, a AWS SCT copia para a pasta de destino todos os arquivos de origem que você excluiu do escopo da conversão. Essa operação possibilita a criação do aplicativo depois de salvar os arquivos convertidos do aplicativo.

Como salvar o código do aplicativo convertido com a AWS SCT

Use o procedimento a seguir para salvar o código do seu aplicativo convertido.

Para salvar o código do seu aplicativo convertido

1. Expanda o nó Java em Aplicativos no painel do banco de dados de destino.
2. Escolha seu aplicativo convertido e selecione Salvar.
3. Insira o caminho para a pasta para salvar o código do aplicativo convertido e escolha Selecionar pasta.

Se seu aplicativo Java de origem usa a estrutura MyBatis, certifique-se de atualizar seu arquivo de configuração para funcionar com seu novo banco de dados.

Como gerenciar projetos de conversão de aplicativos Java na AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos Java, atualizar o código do aplicativo no projeto da AWS SCT ou remover um projeto de conversão em Java do seu projeto da AWS SCT.

Para adicionar um projeto adicional de conversão de aplicativos Java

1. Expanda o nó Aplicativos no painel esquerdo.
2. Escolha o nó Java e abra o menu de contexto (clique com o botão direito).
3. Escolha New application (Nova aplicação).
4. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos Java. Para obter mais informações, consulte [Como criar projetos de conversão de aplicativos Java](#).

Depois de fazer alterações no código fonte do aplicativo, carregue-o no projeto da AWS SCT.

Para carregar o código do aplicativo atualizado

1. Expanda o nó Java em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser atualizado e abra o menu de contexto (clique com o botão direito).
3. Selecione Atualizar e, em seguida, selecione Sim.

A AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações no código que você fez na AWS SCT e os resultados da conversão, crie um novo projeto de conversão em Java.

Se seu aplicativo Java de origem usa a estrutura MyBatis, a AWS SCT usa o arquivo de configuração MyBatis para analisar seu código SQL. Depois de alterar esse arquivo, carregue-o no projeto da AWS SCT.

Para editar o caminho para o arquivo de configuração do MyBatis

1. Expanda o nó Java em Aplicativos no painel esquerdo.
2. Escolha seu aplicativo e, em seguida, selecione Configurações.
3. Selecione Procurar e, em seguida, selecione o arquivo de configuração MyBatis.
4. Escolha Apply (Aplicar).
5. No painel esquerdo, escolha seu aplicativo, abra o menu de contexto (clique com o botão direito do mouse) e selecione Atualizar.

Para remover um projeto de conversão de aplicativos Java

1. Expanda o nó Java em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser removido e abra o menu de contexto (clique com o botão direito).
3. Selecione Excluir e OK.

Como criar um relatório de avaliação de conversão de aplicativos Java na AWS SCT

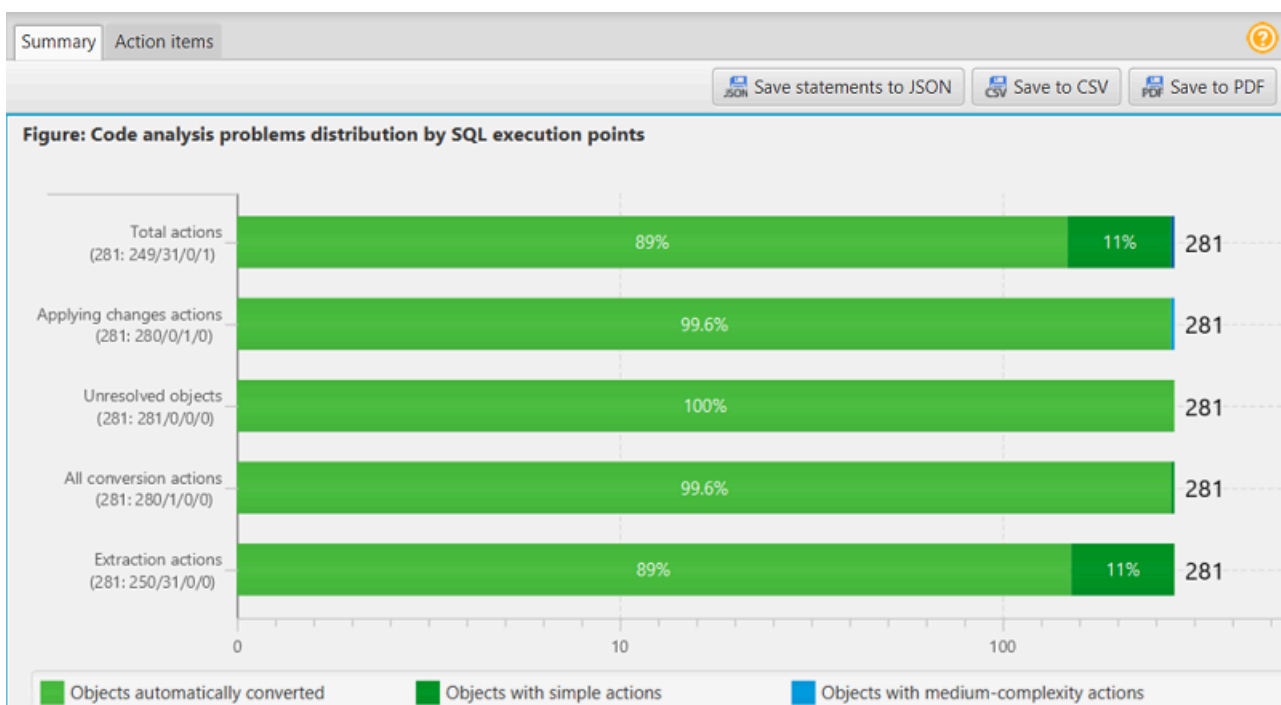
O Relatório de avaliação da conversão do aplicativo Java fornece informações sobre como converter o código SQL incorporado em seu aplicativo Java para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui também inclui itens de ação para código SQL que a AWS SCT não pode converter.

Use o procedimento a seguir para criar um relatório de avaliação de conversão de aplicativos Java.

Para criar um relatório de avaliação de conversão de aplicativos Java

1. Expanda o nó Java em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter.
4. No menu Exibir, selecione Visualização do relatório de avaliação.
5. Analise a guia Resumo.

A guia Resumo mostrada a seguir exibe as informações do resumo executivo do relatório de avaliação de aplicativos Java. Ele mostra os resultados da conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte.



6. Selecione Salvar instruções em JSON para salvar o código SQL extraído do seu aplicativo Java como um arquivo JSON.
7. (Opcional) Salve uma cópia local do relatório como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão do aplicativo.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter o código SQL.

Como converter o código SQL em aplicativos Pro*C com a AWS SCT

Para uma conversão de Oracle para PostgreSQL, você pode usar a AWS Schema Conversion Tool (AWS SCT) para converter o código SQL incorporado em aplicativos Pro*C. Esse conversor de aplicativos Pro*C específico compreende a lógica do aplicativo. Ele coleta instruções que estão localizadas em diferentes objetos do aplicativo, como perfis, parâmetros, variáveis locais e assim por diante.

Devido a essa análise profunda, o conversor de código SQL do aplicativo Pro*C fornece melhores resultados de conversão do que o conversor genérico.

Como criar projetos de conversão de aplicativos Pro*C na AWS SCT

Você pode criar um projeto de conversão de aplicativos Pro*C somente para converter esquemas de banco de dados Oracle em esquemas de banco de dados PostgreSQL. Certifique-se de adicionar uma regra de mapeamento em seu projeto que inclua um esquema Oracle de origem e um banco de dados PostgreSQL de destino. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#).

Você pode adicionar vários projetos de conversão de aplicativos em um único projeto de AWS SCT. Use o procedimento a seguir para criar um projeto de conversão de aplicativos Pro*C.

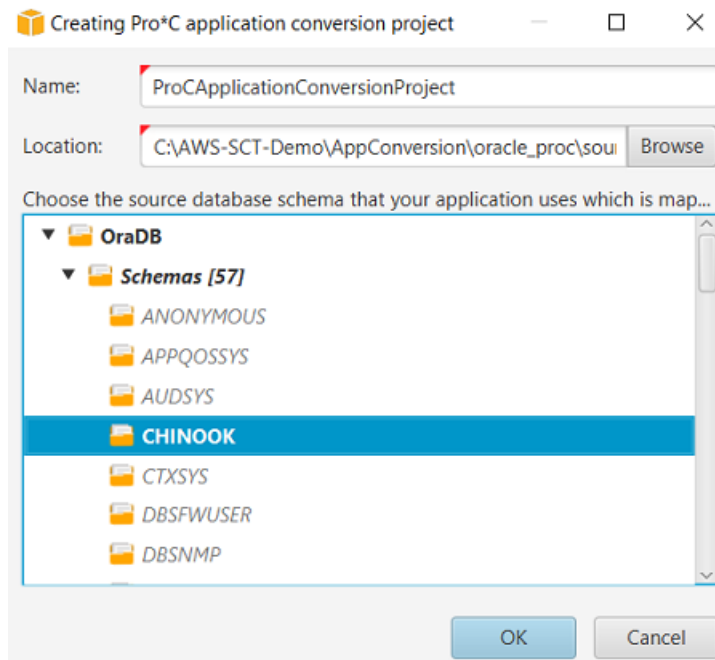
Para criar um projeto de conversão de aplicativos Pro*C

1. Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para obter mais informações, consulte [Criando um projeto da AWS SCT](#) e [Adicionando servidores de banco de dados a um projeto da AWS SCT](#).
2. Adicione uma regra de mapeamento que inclua seu banco de dados Oracle de origem e um banco de dados PostgreSQL de destino. Você pode adicionar um banco de dados PostgreSQL de destino ou usar uma plataforma de banco de dados de destino PostgreSQL virtual em

uma regra de mapeamento. Para obter mais informações, consulte [Criação de regras de mapeamento na AWS SCT](#) e [Usar alvos virtuais](#).

3. No menu Exibir, selecione Visualização principal.
4. No menu Aplicativos, selecione Novo aplicativo Pro*C.

A caixa de diálogo Como criar um projeto de conversão de aplicativos Pro*C será exibida.



5. Em Nome, digite um nome para o seu projeto de conversão de aplicativos Pro*C. Como cada projeto de conversão de esquema de banco de dados pode ter um ou mais projetos de conversão de aplicativos filho, escolha um nome que faça sentido se você adicionar vários projetos.
6. Em Local, digite o local do código fonte referente ao aplicativo.
7. Na árvore de origem, escolha o esquema que seu aplicativo usa. Certifique-se de que esse esquema faça parte de uma regra de mapeamento. A AWS SCT destaca os esquemas que fazem parte de uma regra de mapeamento em negrito.
8. Selecione OK para criar um projeto de conversão de aplicativos Pro*C.
9. Encontre seu projeto de conversão de aplicativos Pro*C no nó Aplicativos no painel esquerdo.

Como converter o código SQL em seu aplicativo Pro*C na AWS SCT

Depois de adicionar seu aplicativo Pro*C ao projeto AWS SCT, converta o código SQL desse aplicativo em um formato compatível com sua plataforma de banco de dados de destino. Use o

procedimento a seguir para analisar e converter o código SQL incorporado em seu aplicativo Pro*C na AWS Schema Conversion Tool.

Para converter seu código SQL

1. Expanda o nó Pro*C em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e selecione Configurações.
 - a. Em Caminho do arquivo de cabeçalho global, insira o caminho para os arquivos de cabeçalho que seu projeto de aplicativo usa.
 - b. Selecione Interpretar todas as variáveis não resolvidas do host para ver todas as variáveis não resolvidas no código convertido.
 - c. Selecione Usar função de conversão de cadeia de caracteres de largura fixa do pacote de extensão para usar as funções do pacote de extensão no código SQL convertido. A AWS SCT inclui os arquivos do pacote de extensão no projeto do seu aplicativo.
 - d. Selecione Transformar blocos PL/SQL anônimos em chamadas SQL autônomas ou perfis armazenados para criar procedimentos armazenados em seu banco de dados de destino para todos os blocos PL/SQL anônimos. A AWS SCT então inclui as execuções desses procedimentos armazenados no código do aplicativo convertido.
 - e. Selecione Usar fluxo de cursor personalizado para melhorar a conversão dos cursores do banco de dados Oracle.
3. No painel esquerdo, selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
4. Selecione Converter. A AWS SCT analisa seus arquivos de código fonte, determina a lógica do aplicativo e carrega os metadados do código no projeto. Esses metadados de código incluem classes, objetos, métodos, variáveis globais, interfaces em Pro*C e assim por diante.

No painel do banco de dados de destino, a AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido.

The screenshot displays two panels of code. The top panel, titled 'Source Oracle app function: main() int', shows the original Pro*C code with lines 11 through 20. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted SQL code with lines 8 through 18. The converted code uses 'EXEC SQL' statements to execute the same logic as the original Pro*C code.

```
Source Oracle app function: main() int
Properties Text Related converted objects Statistics Settings
11 int i = 5;
12
13 /* Connect string */
14
15 EXEC SQL INSERT INTO embeddedc.t_insert(i) VALUES (:i);
16
17 EXEC SQL COMMIT;
18
19 return 0;
20 }
Cursor position: 118 Click position: 197

Target Amazon RDS for PostgreSQL app function: main() int
Properties Text Apply status Key management
8
9 EXEC SQL int i = 5;
10
11 /* Connect string */
12
13 EXEC SQL INSERT INTO embeddedc.t_insert (i)
14 VALUES (:i);
15
16 EXEC SQL COMMIT;
17
18 return 0;
```

5. Salve o código do seu aplicativo convertido. Para obter mais informações, consulte [Como editar e salvar o código do aplicativo convertido](#).

Como editar e salvar o código do aplicativo convertido com a AWS SCT

Você pode editar as instruções SQL convertidas e usar a AWS SCT para incorporar esse código editado no código do aplicativo Pro*C convertido. Use o procedimento a seguir para editar seu código SQL convertido.

Para editar seu código SQL convertido

1. Expanda o nó Pro*C em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido, abra o menu de contexto (clique com o botão direito) e selecione Converter.
3. No menu Exibir, selecione Visualização do relatório de avaliação.
4. Selecione Salvar instruções em CSV para salvar o código SQL extraído do seu aplicativo Pro*C como um arquivo CSV.
5. Insira o nome do arquivo CSV para salvar o código SQL extraído e escolha selecione Salvar.
6. Edite o código SQL extraído.
7. No menu Exibir, selecione Visualização principal.

8. Expanda o nó Pro*C em Aplicativos no painel do banco de dados de destino.
9. Escolha seu aplicativo convertido, abra o menu de contexto (clique com o botão direito) e selecione Importar instruções do CSV.
10. Selecione Sim, depois escolha o arquivo com seu código SQL editado e selecione Abrir.

A AWS SCT divide as instruções SQL convertidas em partes e as coloca nos objetos apropriados do código fonte do aplicativo. Use o procedimento a seguir para salvar o código do seu aplicativo convertido.

Para salvar o código do seu aplicativo convertido

1. Expanda o nó Pro*C em Aplicativos no painel do banco de dados de destino.
2. Escolha seu aplicativo convertido e selecione Salvar.
3. Insira o caminho para a pasta para salvar o código do aplicativo convertido e escolha Selecionar pasta.

Como gerenciar projetos de conversão de aplicativos Pro*C na AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos Pro*C, atualizar o código do aplicativo no projeto da AWS SCT ou remover um projeto de conversão em Pro*C do seu projeto da AWS SCT.

Para adicionar um projeto adicional de conversão de aplicativos Pro*C

1. Expanda o nó Aplicativos no painel esquerdo.
2. Escolha o nó Pro*C e abra o menu de contexto (clique com o botão direito).
3. Escolha New application (Nova aplicação).
4. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos Pro*C. Para obter mais informações, consulte [Como criar projetos de conversão de aplicativos Pro*C](#).

Depois de fazer alterações no código fonte do aplicativo, carregue-o no projeto da AWS SCT.

Para carregar o código do aplicativo atualizado

1. Expanda o nó Pro*C em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser atualizado e abra o menu de contexto (clique com o botão direito).

3. Selecione Atualizar e, em seguida, selecione Sim.

A AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações no código que você fez na AWS SCT e os resultados da conversão, crie um novo projeto de conversão em Pro*C.

Para remover um projeto de conversão de aplicativos Pro*C

1. Expanda o nó Pro*C em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser removido e abra o menu de contexto (clique com o botão direito).
3. Selecione Excluir e OK.

Como criar um relatório de avaliação de conversão de aplicativos Pro*C na AWS SCT

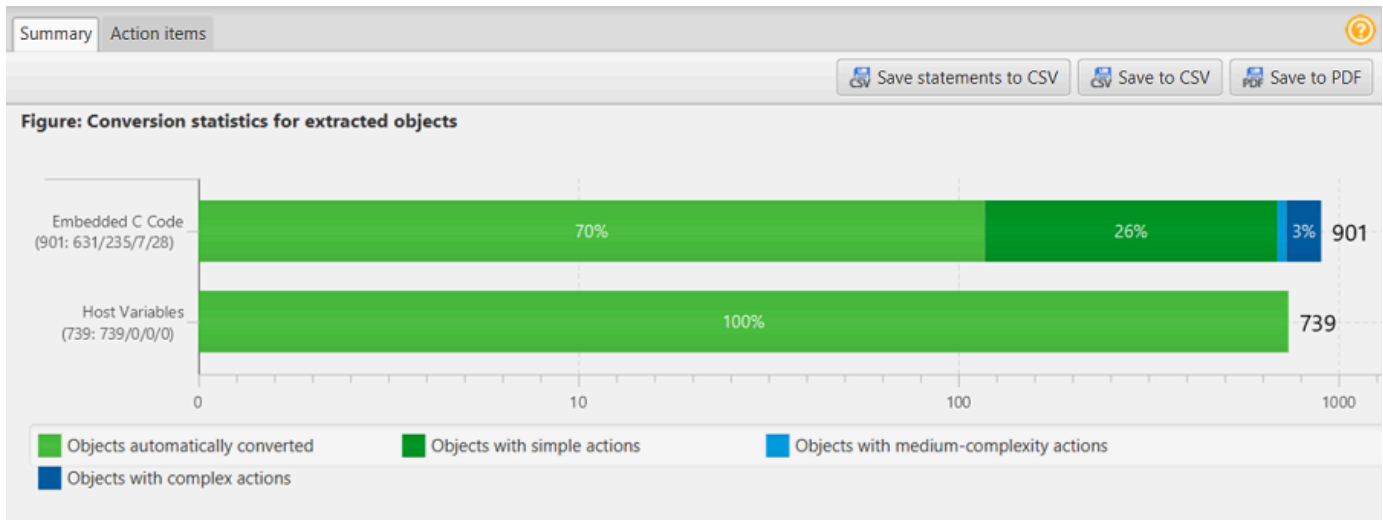
O Relatório de avaliação da conversão do aplicativo Pro*C fornece informações sobre como converter o código SQL incorporado em seu aplicativo Pro*C para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui também inclui itens de ação para código SQL que a AWS SCT não pode converter.

Use o procedimento a seguir para criar um relatório de avaliação de aplicativos Pro*C.

Para criar um relatório de avaliação de conversão de aplicativos Pro*C

1. Expanda o nó Pro*C em Aplicativos no painel esquerdo.
2. Selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
3. Selecione Converter.
4. No menu Exibir, selecione Visualização do relatório de avaliação.
5. Analise a guia Resumo.

A guia Resumo mostrada a seguir exibe as informações do resumo executivo do relatório de avaliação de aplicativos Pro*C. Ele mostra os resultados da conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte.



6. Selecione Salvar instruções em CSV para salvar o código SQL extraído do seu aplicativo Pro*C como um arquivo de valores separados por vírgulas (CSV).
7. (Opcional) Salve uma cópia local do relatório como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):

- Selecione Salvar em PDF no canto superior direito para salvar o relatório como um arquivo PDF.

O arquivo PDF contém o resumo executivo, os itens de ação e as recomendações para a conversão do aplicativo.

- Selecione Salvar em CSV no canto superior direito para salvar o relatório como um arquivo CSV.

O arquivo CSV contém itens de ação, ações recomendadas e uma complexidade estimada do esforço manual necessário para converter o código SQL.

Usando AWS SCT pacotes de extensão

O pacote de extensão da AWS SCT é um módulo complementar que emula funções presentes em um banco de dados de origem e necessárias ao converter objetos para o banco de dados de destino. Antes de instalar um pacote de AWS SCT extensão, você converte o esquema do banco de dados.

Cada pacote AWS SCT de extensão inclui os seguintes componentes:

- Esquema de banco de dados: inclui funções, procedimentos e tabelas SQL para emular determinados objetos de banco de dados de processamento de transações on-line (OLTP) e processamento analítico on-line (OLAP), como sequências. Além disso, emula sem suporte built-in-functions do banco de dados de origem. O nome desse esquema tem o seguinte formato: `aws_database_engine_name_ext`.
- AWS Lambda funções (para determinados bancos de dados OLTP) — Inclui AWS Lambda funções que emulam funcionalidades complexas do banco de dados, como agendamento de tarefas e envio de e-mails.
- Bibliotecas personalizadas para bancos de dados OLAP — Inclui um conjunto de bibliotecas Java e Python que você pode usar para migrar scripts de extração, transformação e carregamento (ETL) do Microsoft SQL Server Integration Services (SSIS) para ou. AWS Glue AWS Glue Studio

As bibliotecas Java incluem os seguintes módulos:

- `spark-excel_2.11-0.13.1.jar`: para emular a funcionalidade dos componentes de origem e destino do Excel.
- `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar` e `xmlbeans-3.1.0.jar`: para emular a funcionalidade do componente de origem XML.

As bibliotecas Python incluem os seguintes módulos:

- `sct_utils.py`: para emular os tipos de dados de origem e preparar parâmetros para a consulta do Spark SQL.
- `ssis_datetime.py`: para emular funções integradas de data e hora.
- `ssis_null.py`: para emular as funções integradas ISNULL e REPLACENULL.
- `ssis_string.py`: para emular funções integradas de string.

Para mais informações sobre essas bibliotecas, consulte [Como usar bibliotecas personalizadas para pacotes de extensão da AWS SCT](#).

Você pode aplicar pacotes AWS SCT de extensão de duas maneiras:

- AWS SCT pode aplicar automaticamente um pacote de extensão ao aplicar um script de banco de dados de destino escolhendo Aplicar ao banco de dados no menu de contexto. AWS SCT aplica o pacote de extensão antes de aplicar todos os outros objetos do esquema.
- Para aplicar manualmente o pacote de extensão escolha o banco de dados de destino e em seguida escolha Aplicar pacote de extensão para no menu de contexto (clique com o botão direito do mouse). Na maioria das situações, a aplicação automática é suficiente. No entanto, talvez você queira aplicar o pacote manualmente caso ele seja excluído acidentalmente.

Cada vez que você aplica um pacote de AWS SCT extensão a um armazenamento de dados de destino, os componentes são sobrescritos e AWS SCT exibem uma notificação sobre isso. Para desativar essas notificações, escolha Configurações, Configurações globais, Notificações e selecione Ocultar o alerta de substituição do pacote de extensão.

Para uma conversão do Microsoft SQL Server para o PostgreSQL, você pode usar o pacote de extensão SQL Server para PostgreSQL na AWS SCT. Esse pacote de extensão emula o SQL Server Agent e o SQL Server Database Mail. Para ter mais informações, consulte [Emulando o SQL Server Agent no PostgreSQL com um pacote de extensão](#) e [Emulando o SQL Server Database Mail no PostgreSQL com um pacote de extensão](#).

A seguir, você pode encontrar mais informações sobre como trabalhar com pacotes AWS SCT de extensão.

Tópicos

- [Permissões para usar o pacote AWS SCT de extensão](#)
- [Como usar o esquema do pacote de extensão](#)
- [Como usar bibliotecas personalizadas para pacotes de extensão da AWS SCT](#)
- [Usando as AWS Lambda funções do pacote AWS SCT de extensão](#)
- [Configurando funções para o pacote de AWS SCT extensão](#)

Permissões para usar o pacote AWS SCT de extensão

O pacote AWS SCT de extensão para o Amazon Aurora emula envio de e-mails, agendamento de trabalhos, filas e outras operações usando funções. AWS Lambda Quando você aplica o pacote de AWS SCT extensão ao seu banco de dados Aurora de destino, AWS SCT cria uma nova função

AWS Identity and Access Management (IAM) e uma política de IAM embutida. Em seguida, AWS SCT cria uma nova função Lambda e configura seu cluster de banco de dados Aurora para conexões de saída com o. AWS Lambda Para executar essas operações, certifique-se de conceder as seguintes permissões necessárias ao seu usuário do IAM:

- `iam:CreateRole`— para criar uma nova função do IAM para sua AWS conta.
- `iam:CreatePolicy`— para criar uma nova política do IAM para sua AWS conta.
- `iam:AttachRolePolicy`: para vincular a política especificada ao seu perfil do IAM.
- `iam:PutRolePolicy`: para atualizar um documento de política em linha incorporado em sua função do IAM.
- `iam:PassRole`: para passar a função do IAM especificada para o mecanismo de regras.
- `iam:TagRole`: para adicionar tags a uma função do IAM.
- `iam:TagPolicy`: para adicionar tags a uma política do IAM.
- `lambda:ListFunctions`: para ver a lista de suas funções do Lambda.
- `lambda:ListTags`: para ver a lista de tags de suas funções do Lambda.
- `lambda:CreateFunction`: para criar uma nova função do Lambda.
- `rds:AddRoleToDBCluster`: para associar essa função do IAM ao seu cluster de banco de dados do Aurora.

O pacote AWS SCT de extensão para o Amazon Redshift emula as funções básicas do armazém de dados de origem que são necessárias ao aplicar objetos convertidos ao Amazon Redshift. Antes de aplicar seu código convertido ao Amazon Redshift, você deve aplicar o pacote de extensão para o Amazon Redshift. Para fazer isso, inclua a ação `iam:SimulatePrincipalPolicy` na sua política do IAM.

AWS SCT usa o IAM Policy Simulator para verificar as permissões necessárias para instalar o pacote de extensões do Amazon Redshift. O simulador de políticas do IAM pode exibir uma mensagem de erro mesmo que você tenha configurado corretamente seu usuário do IAM. Este é um problema conhecido do simulador de políticas do IAM. Além disso, o simulador de políticas do IAM exibe uma mensagem de erro quando você não tem a ação `iam:SimulatePrincipalPolicy` na sua política do IAM. Nesses casos, você pode ignorar a mensagem de erro e aplicar o pacote de extensão usando o assistente de pacote de extensão. Para ter mais informações, consulte [Como aplicar o pacote de extensão](#).

Como usar o esquema do pacote de extensão

Quando você converte seu banco de dados ou esquema de data warehouse, a AWS SCT adiciona mais um esquema ao seu banco de dados de destino. Esse esquema implementa as funções de sistema SQL do banco de dados de origem necessárias para gravar o esquema convertido no banco de dados de destino. Esse esquema adicional é chamado de esquema do pacote de extensões.

O esquema do pacote de extensões para bancos de dados OLTP é nomeado de acordo com o banco de dados de origem da seguinte forma:

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

O esquema do pacote de extensões para aplicativos de data warehouse OLAP é nomeado de acordo com o armazenamento de dados de origem da seguinte forma:

- 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`

Como usar bibliotecas personalizadas para pacotes de extensão da AWS SCT

Em alguns casos, não é possível converter recursos do banco de dados de origem em recursos equivalentes no banco de dados de destino. O pacote de extensão relevante contém bibliotecas personalizadas que emulam algumas funcionalidades do banco de dados de origem em seu banco de dados de destino.

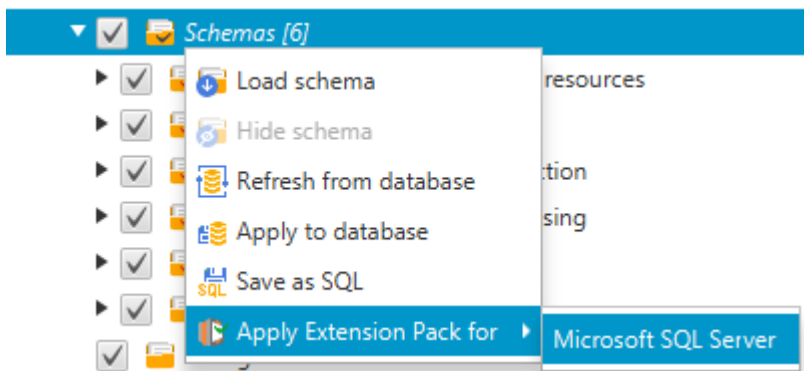
Se você estiver convertendo um banco de dados transacional, consulte [Usando as AWS Lambda funções do pacote AWS SCT de extensão](#).

Como aplicar o pacote de extensão

Você pode aplicar o pacote de AWS SCT extensão usando o assistente do pacote de extensão ou ao aplicar o código convertido ao seu banco de dados de destino.

Para aplicar o pacote de extensão usando o assistente de pacote de extensão

1. Na árvore do AWS Schema Conversion Tool banco de dados de destino, abra o menu de contexto (clique com o botão direito do mouse), escolha Aplicar pacote de extensão para e escolha sua plataforma de banco de dados de origem.



O assistente do pacote de extensões é exibido.

2. Leia a página Bem-vindo e escolha Próximo.
3. Na página Configurações do perfil do AWS , faça o seguinte:
 - Se você estiver reinstalando apenas o esquema do pacote de extensões, escolha Ignorar esta etapa por enquanto e, em seguida, Próximo. A opção Ignorar etapa por enquanto só está disponível para bancos de dados de processamento de transações online (OLTP).
 - Se você estiver fazendo upload da biblioteca nova, forneça as credenciais para se conectar à sua Conta da AWS. Use essa etapa somente ao converter bancos de dados OLAP ou scripts de ETL. Você pode usar suas credenciais AWS Command Line Interface (AWS CLI) se tiver as AWS CLI instaladas. Também é possível usar as credenciais armazenadas anteriormente em um perfil nas configurações globais do aplicativo associadas ao projeto. Se necessário, escolha Navegar até as configurações globais para configurar ou associar um perfil diferente ao seu AWS SCT projeto. Para ter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).
4. Se você estiver fazendo o upload de uma nova biblioteca escolha Preciso carregar uma biblioteca na página de Upload da biblioteca. Use essa etapa somente ao converter bancos de

dados OLAP ou scripts de ETL. Em seguida, forneça o caminho do Amazon S3 e escolha Fazer upload da biblioteca para o S3.

Se você já fez o upload da biblioteca escolha Já tenho bibliotecas carregadas, use meu bucket S3 existente na página Upload da biblioteca. Em seguida, forneça o caminho do Amazon S3.

Quando concluir, escolha Next.

5. Na página Emulação de funções, escolha Criar pacote de extensão. As mensagens são exibidas com o status das operações do pacote de extensões.

Quando terminar, escolha Concluir.

Para aplicar o pacote de extensão ao aplicar o código convertido

1. Especifique o bucket do Amazon S3 em seu perfil de AWS serviço. Use essa etapa somente ao converter bancos de dados OLAP ou scripts de ETL. Para ter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).

Certifique-se de que sua política do bucket do Amazon S3 inclua as seguintes permissões:

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

No exemplo anterior, substitua *DataExtractionAgentName111122223333:user/* pelo nome do seu usuário do IAM.

2. Converter seus esquemas de data warehouse de origem. Para ter mais informações, consulte [Como converter esquemas de data warehouse em Amazon Redshift](#).
3. No painel do direito escolha o esquema convertido.
4. Abra o menu de contexto (clique com o botão direito do mouse) do elemento do esquema e escolha Aplicar ao banco de dados.
5. AWS SCT gera pacotes de extensão com os componentes necessários e adiciona o `aws_database_engine_name_ext` esquema na árvore de destino. Em seguida, AWS SCT aplica o código convertido e o esquema do pacote de extensões ao seu data warehouse de destino.

Quando você usa uma combinação do Amazon Redshift e AWS Glue como sua plataforma de banco de dados de destino, AWS SCT adiciona um esquema adicional ao pacote de extensões.

Usando as AWS Lambda funções do pacote AWS SCT de extensão

AWS SCT fornece um pacote de extensões que contém funções Lambda para e-mail, agendamento de tarefas e outros recursos para bancos de dados hospedados no Amazon EC2.

Usando AWS Lambda funções para emular a funcionalidade do banco de dados

Em alguns casos, os recursos do banco de dados não podem ser convertidos em recursos do Amazon RDS equivalentes. Por exemplo, a Oracle envia chamadas de e-mail que usam UTL_SMTP, e o Microsoft SQL Server pode usar um programador de trabalho. Se você hospedar e autogerenciar um banco de dados no Amazon EC2, poderá emular esses recursos AWS substituindo-os por serviços.

O assistente do pacote de AWS SCT extensões ajuda você a instalar, criar e configurar funções do Lambda para emular e-mail, agendamento de trabalhos e outros recursos.

Como aplicar o pacote de extensão para oferecer suporte às funções do Lambda

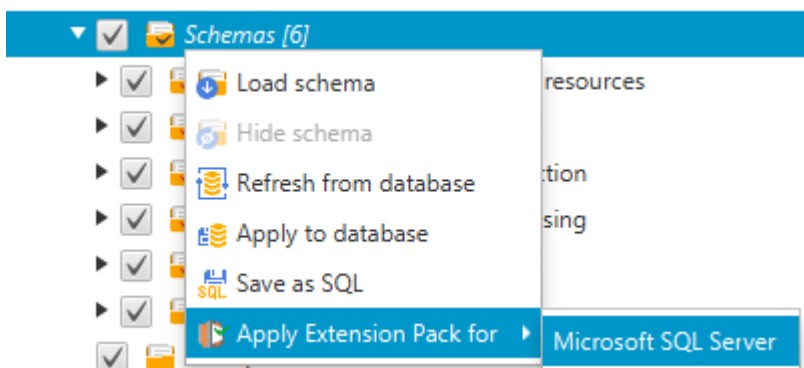
Você pode aplicar o pacote de extensão para oferecer suporte às funções do Lambda usando o assistente de pacote de extensão ou aplicando o código convertido ao seu banco de dados de destino.

⚠ Important

Os recursos AWS de emulação de serviços são compatíveis somente com bancos de dados instalados e autogerenciados no Amazon EC2. Não instale os recursos de emulação de serviço se o seu banco de dados de destino estiver em uma instância de banco de dados do Amazon RDS.

Para aplicar o pacote de extensão usando o assistente de pacote de extensão

1. Na árvore do AWS Schema Conversion Tool banco de dados de destino, abra o menu de contexto (clique com o botão direito do mouse), escolha Aplicar pacote de extensão para e escolha sua plataforma de banco de dados de origem.



O assistente do pacote de extensões é exibido.

2. Leia a página Bem-vindo e escolha Próximo.
3. Na página Configurações do perfil do AWS , faça o seguinte:
 - Se você estiver reinstalando apenas o esquema do pacote de extensões, escolha Ignorar esta etapa por enquanto e, em seguida, Próximo.

- Se você estiver instalando AWS serviços, forneça as credenciais para se conectar ao seu Conta da AWS. Você pode usar suas AWS CLI credenciais se tiver AWS CLI instalado. Também é possível usar as credenciais armazenadas anteriormente em um perfil nas configurações globais do aplicativo associadas ao projeto. Se necessário, escolha Navegar até configurações do projeto para associar um perfil diferente ao projeto. Se necessário, escolha Configurações globais para criar um novo perfil. Para ter mais informações, consulte [Armazenando perfis de serviço da AWS na AWS SCT](#).
4. Na página Serviço de envio de e-mail, faça o seguinte:
 - Se você estiver reinstalando apenas o esquema do pacote de extensões, escolha Ignorar esta etapa por enquanto e, em seguida, Próximo.
 - Se você estiver instalando AWS serviços e tiver uma função Lambda existente, poderá fornecê-la. Caso contrário, o assistente a cria para você. Quando concluir, escolha Next.
 5. Na página Serviço de emulação de tarefas, faça o seguinte:
 - Se você estiver reinstalando apenas o esquema do pacote de extensões, escolha Ignorar esta etapa por enquanto e, em seguida, Próximo.
 - Se você estiver instalando AWS serviços e tiver uma função Lambda existente, poderá fornecê-la. Caso contrário, o assistente a cria para você. Quando concluir, escolha Next.
 6. Na página Emulação de funções, escolha Criar pacote de extensão. As mensagens são exibidas com o status das operações do pacote de extensões.

Quando terminar, escolha Concluir.

Note

Para atualizar um pacote de extensão e substituir os componentes antigos do pacote de extensão, certifique-se de usar a versão mais recente do AWS SCT. Para ter mais informações, consulte [Instalando, verificando e atualizando AWS SCT](#).

Configurando funções para o pacote de AWS SCT extensão

O pacote de extensão contém funções que você deve configurar antes de usar. A constante `CONVERSION_LANG` define o idioma que o service pack usa. As funções estão disponíveis para inglês e alemão.

Para definir o idioma como inglês ou alemão, faça a seguinte alteração no código da função. Encontre a seguinte declaração constante:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

`CONVERSION_LANG` Para definir como inglês, altere a linha para o seguinte:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

`CONVERSION_LANG` Para definir como inglês, altere a linha para o seguinte:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Defina essa configuração para as seguintes funções:

- `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`

Melhores recomendadas para o AWS SCT

Encontre informações sobre as melhores práticas e opções para usar a AWS Schema Conversion Tool (AWS SCT).

Como configurar memória adicional

Para converter grandes esquemas de banco de dados, como um banco de dados com 3.500 procedimentos armazenados, você pode configurar a quantidade de memória disponível para a AWS Schema Conversion Tool.

Para modificar a quantidade de memória consumida pela AWS SCT

1. No menu Configurações, selecione Configurações globais e, em seguida, selecione Opções de JVM.
2. Selecione Editar arquivo de configuração e escolha o editor de texto para abrir o arquivo de configuração.
3. Edite a seção `JavaOptions` para definir a quantidade mínima e máxima de memória disponível. O exemplo a seguir define o mínimo como quatro GB e o máximo como 40 GB.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Recomendamos que você defina a memória mínima disponível para pelo menos quatro GB.

4. Salve o arquivo de configuração, selecione OK e reinicie a AWS SCT para aplicar as alterações.

Como configurar a pasta padrão do projeto

A AWS SCT usa a pasta do projeto para armazenar os arquivos do projeto, salvar relatórios de avaliação e armazenar o código convertido. Por padrão, a AWS SCT armazena todos os arquivos na pasta do aplicativo. Você pode especificar outra pasta como a pasta padrão do projeto.

Para alterar a pasta padrão do projeto

1. No menu Configurações, selecione Configurações globais e, em seguida, selecione Caminho do arquivo.

2. Em Caminho padrão do arquivo de projeto, insira o caminho para a pasta padrão do projeto.
3. Escolha Apply e, em seguida, escolha OK.

Como aumentar a velocidade da migração de dados

Para migrar grandes conjuntos de dados, como um conjunto de tabelas com mais de 1 TB de dados, talvez você queira aumentar a velocidade da migração. Quando você usa atendentes de extração de dados, a velocidade das migrações de dados depende de vários fatores. Esses fatores incluem o número de fatias em seu cluster de destino do Amazon Redshift, o tamanho de um arquivo fragmentado em sua tarefa de migração, a RAM disponível no PC em que você executa seus atendentes de extração de dados e assim por diante.

Para aumentar a velocidade de migração de dados, recomendamos realizar várias sessões de migração de teste com pequenos conjuntos de dados de seus dados de produção. Além disso, recomendamos que você execute seus atendentes de extração de dados em um PC com um SSD com pelo menos 500 GB de tamanho. Durante essas sessões de teste, altere os diferentes parâmetros de migração e monitore a utilização do disco para descobrir a configuração que garante a velocidade máxima de migração de dados. Em seguida, use essa configuração para migrar todo o conjunto de dados.

Como aumentar as informações de registro em log

Você pode aumentar as informações de registro em log produzidas pela AWS SCT ao converter seus bancos de dados, scripts e SQL do aplicativo. Embora o aumento das informações de registro em log possa retardar a conversão, as alterações podem ajudar você a fornecer informações robustas ao Suporte da AWS em caso de erros.

A AWS SCT armazena logs em seu ambiente local. Você pode visualizar esses arquivos de log e compartilhá-los com o Suporte da AWS ou com os desenvolvedores da AWS SCT para solucionar problemas.

Para alterar as configurações de registro em log

1. No menu Configurações, selecione Configurações globais e, em seguida, selecione Registro em log.
2. Em Caminho da pasta de log, insira a pasta para armazenar os logs da interface do usuário.

3. Em Caminho da pasta de log do console, insira a pasta para armazenar os logs da interface de linha de comandos (CLI) da AWS SCT.
4. Em Tamanho máximo do arquivo de log (MB), insira o tamanho, em MB, de um único arquivo de log. Depois que seu arquivo atingir esse limite, a AWS SCT cria um novo arquivo de log.
5. Em Número máximo de arquivos de log, insira o número de arquivos de log a serem armazenados. Depois que o número de arquivos de log na pasta atingir esse limite, a AWS SCT exclui o arquivo de log mais antigo.
6. Em Caminho de download do log dos extratores, insira a pasta para armazenar os logs dos atendentes de extração de dados.
7. Em Caminho do log do extrator Cassandra, insira a pasta para armazenar os logs dos atendentes de extração de dados.
8. Selecione Solicitar um caminho antes de carregar para garantir que a AWS SCT ele pergunte onde armazenar os registros sempre que você usar atendentes de extração de dados.
9. Em Modo de depuração, selecione Verdadeiro. Use essa opção para registrar informações adicionais quando os logs padrão da AWS SCT não incluírem nenhum problema.
10. Escolha os principais módulos do aplicativo para aumentar as informações de registro em log. Você pode aumentar as informações de registro em log para os seguintes módulos de aplicativos:
 - Geral
 - Carregador
 - Analisador
 - Impressora
 - Resolvedor
 - Telemetria
 - Conversor
 - Mapeamento de tipos
 - User interface (Interface do usuário)
 - Controlador
 - Comparar esquema
 - Clone de datacenter
 - Analisador de aplicativos

Para cada um dos módulos de aplicativo anteriores, escolha um dos seguintes níveis de registro em log:

- Rastreamento – Informações mais detalhadas.
- Depuração – Informações detalhadas sobre o fluxo pelo sistema.
- Informações – Eventos de runtime, como startup ou encerramento.
- Aviso – Uso de APIs obsoletas, mau uso da API e outras situações de runtime indesejáveis ou inesperadas.
- Erro – Erros de runtime ou condições inesperadas.
- Crítico – Erros que levam ao encerramento do aplicativo.
- Obrigatório – O nível mais alto possível de erros.

Por padrão, depois de ativar o Modo de depuração, a AWS SCT define o nível de registro em log de Informações para todos os módulos do aplicativo.

Por exemplo, para ajudar nas principais áreas problemáticas durante a conversão, defina Analisador, Mapeamento de tipos e Interface do usuário como Rastreamento.

Se as informações se tornarem muito detalhadas para o sistema de arquivos em que os logs estão sendo transmitidos, mude para um local com espaço suficiente para capturar os logs.

Para transmitir registros para o Suporte da AWS, acesse o diretório em que os logs estão armazenados e compacte todos os arquivos em um único arquivo .zip gerenciável. Em seguida, carregue o arquivo .zip com o caso de suporte. Quando a análise inicial for concluída e o desenvolvimento contínuo for retomado, retorne o Modo de depuração para falso para eliminar o registro de log detalhado. Em seguida, aumente a velocidade de conversão.

Tip

Para gerenciar o tamanho do log e simplificar os problemas de emissão de relatórios, remova os logs ou mova-os para outro local após uma conversão bem-sucedida. Fazer essa tarefa garante que somente os erros e as informações relevantes sejam transmitidos ao Suporte da AWS e evita que o sistema de arquivos de log seja preenchido.

Solução de problemas com o AWS SCT

A seguir, é possível encontrar informações sobre a solução de problemas com a AWS Schema Conversion Tool (AWS SCT).

Não é possível carregar objetos em um banco de dados de origem da Oracle

Ao tentar carregar o esquema a partir de um banco de dados Oracle, você pode encontrar um dos seguintes erros.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Esses erros ocorrem porque o usuário cujo ID é usado para se conectar ao banco de dados Oracle não tem permissões suficientes para ler o esquema, conforme exigido por AWS SCT.

Você pode resolver o problema ao conceder `select_catalog_role` a permissão ao usuário e a todos os dicionários do banco de dados. Essas permissões fornecem o acesso somente leitura para visualizações e tabelas do sistema que são necessárias na AWS SCT. O exemplo a seguir cria um ID de usuário chamado `min_privs` e concede ao usuário com esse ID as permissões mínimas necessárias para converter o esquema de um banco de dados de origem da Oracle.

```
create user min_privs identified by min_privs;  
grant connect to min_privs;  
grant select_catalog_role to min_privs;  
grant select any dictionary to min_privs;
```

Mensagem de aviso do relatório de avaliação

Para avaliar a complexidade da conversão para outro mecanismo de banco de dados, a AWS SCT requer acesso aos objetos em seu banco de dados de origem. Quando a AWS SCT encontra problemas durante a digitalização e não consegue realizar uma avaliação, uma mensagem de aviso é emitida. Essa mensagem indica que a porcentagem geral de conversão foi reduzida. A seguir estão os motivos pelos quais a AWS SCT pode encontrar problemas durante a digitalização:

- O usuário do seu banco de dados não tem acesso a todos os objetos necessários. Para obter mais informações sobre as permissões e privilégios de segurança necessários pela AWS SCT para o seu banco de dados, consulte o [Origens para a AWS SCT](#) na seção apropriada do banco de dados de origem neste guia.
- Um objeto citado no esquema não existe mais no banco de dados. Para ajudar a resolver o problema, você pode se conectar com as permissões do SYSDBA e verificar se o objeto está presente no banco de dados.
- O SCT está tentando avaliar um objeto criptografado.

AWS SCT Referência da CLI

Esta seção descreve como começar a usar a interface de linha de AWS SCT comando (CLI). Além disso, esta seção fornece informações sobre os principais comandos e modos de uso. Para obter uma referência completa dos comandos da AWS SCT CLI, consulte [Material de referência](#)

Tópicos

- [Pré-requisitos para usar a interface de linha de comando do AWS SCT](#)
- [AWS SCT Modo interativo CLI](#)
- [Obtendo AWS SCT cenários de CLI](#)
- [Editando AWS SCT cenários de CLI](#)
- [AWS SCT Modo de script CLI](#)
- [AWS SCT Material de referência CLI](#)

Pré-requisitos para usar a interface de linha de comando do AWS SCT

Faça download e instale a versão mais recente do Amazon Corretto 11. Para obter mais informações, consulte [Downloads do Amazon Corretto 11](#) no Guia do usuário do Amazon Corretto 11.

Baixe e instale a versão mais recente do AWS SCT. Para ter mais informações, consulte [Instalando AWS SCT](#).

AWS SCT Modo interativo CLI

Você pode usar a interface da linha de AWS SCT comando no modo interativo. Nesse modo, você insere comandos no console um por um. Você pode usar esse modo interativo para saber mais sobre os comandos da CLI ou baixar os cenários da CLI mais usados.

Para converter o esquema do banco de dados de origem em AWS SCT, execute uma operação de sequência: crie um novo projeto, conecte-se aos bancos de dados de origem e destino, crie regras de mapeamento e converta objetos do banco de dados. Como esse fluxo de trabalho pode ser complexo, recomendamos o uso de scripts no modo AWS SCT CLI. Para ter mais informações, consulte [Modo de script](#).

Você pode executar os comandos da AWS SCT CLI na app pasta do seu caminho de AWS SCT instalação. No Windows, o caminho de instalação padrão é `C:\Program Files\AWS Schema Conversion Tool\`. Certifique-se de que essa pasta inclua o arquivo `AWSSchemaConversionToolBatch.jar`.

Para entrar no modo interativo da AWS SCT CLI, use o comando a seguir depois de concluir os pré-requisitos.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Agora você pode executar comandos da AWS SCT CLI. Certifique-se de finalizar seus comandos com `/` em uma nova linha. Além disso, certifique-se de usar aspas simples retas (`'`) antes e depois dos valores dos parâmetros do comando.

Note

Se o comando anterior retornar `Unexpected error`, tente o seguinte:

```
java -Djdk.jar.maxSignatureFileSize=20000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Para ver a lista de comandos disponíveis no modo interativo AWS SCT CLI, execute o comando a seguir.

```
help  
/
```

Para ver informações sobre um comando da AWS SCT CLI, use o comando a seguir.

```
help -command: 'command_name'  
/
```

No exemplo anterior, substitua *command_name* pelo nome de um comando.

Para ver informações sobre os parâmetros de um comando da AWS SCT CLI, use o comando a seguir.

```
help -command: 'command_name' -parameters: 'parameters_list'
```

```
/
```

No exemplo anterior, substitua *command_name* pelo nome de um comando. Em seguida, substitua *parameters_list* por uma lista de nomes de parâmetros separados por vírgula.

Para executar um script a partir de um arquivo no modo interativo AWS SCT CLI, use o comando a seguir.

```
ExecuteFile -file: 'file_path'  
/
```

No exemplo anterior, substitua *file_path* pelo caminho do seu arquivo por um script. Certifique-se de que seu arquivo tenha uma extensão `.scts`.

Para sair do modo interativo AWS SCT CLI, execute o `quit` comando.

Exemplos

O exemplo a seguir mostra informações sobre o comando `Convert`.

```
help -command: 'Convert'  
/
```

O exemplo a seguir mostra informações sobre dois parâmetros do comando `Convert`.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

Obtendo AWS SCT cenários de CLI

Para obter os AWS SCT cenários mais usados, você pode usar o `GetCliScenario` comando. Você pode executar esse comando no modo interativo e, em seguida, editar os modelos baixados. Use os arquivos editados no modo script.

O comando `GetCliScenario` salva o modelo selecionado ou todos os modelos disponíveis no diretório especificado. O modelo contém o conjunto completo de comandos para executar um script. Certifique-se de editar os caminhos do arquivo, as credenciais do banco de dados, os nomes dos objetos e outros dados nesses modelos. Além disso, certifique-se de remover os comandos que não usa e adicionar novos comandos ao script quando necessário.

Para entrar no modo interativo da CLI do GetCliScenario, conclua os pré-requisitos e entre no modo interativo da CLI da AWS SCT . Para ter mais informações, consulte [Modo interativo](#).

Em seguida, use a sintaxe a seguir para executar o comando GetCliScenario e obter os cenários AWS SCT .

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

No exemplo anterior, substitua *template_type* por um dos tipos de modelo da tabela a seguir. Em seguida, substitua *file_path* pelo caminho da pasta na qual você deseja baixar os scripts. Certifique-se de que AWS SCT possa acessar essa pasta sem solicitar direitos de administrador. Além disso, certifique-se de usar aspas simples retas (') antes e depois dos valores dos parâmetros do comando.

Para baixar todos os modelos de AWS SCT CLI, execute o comando anterior sem a opção. -type

A tabela a seguir inclui os tipos de modelos de AWS SCT CLI que você pode baixar. Para cada modelo, a tabela inclui o nome do arquivo e a descrição das operações que você pode executar usando o script.

Tipo de modelo	Nome do arquivo	Descrição
BTEQ ScriptConversion	BTEQScriptConversionTemplate.scts	Converte Teradata Basic Teradata Query (BTEQ),, FastExport e FastLoad scripts em RSQL do Amazon Redshift. MultiLoad Para ter mais informações, consulte Como converter processos de ETL .
ConversionApply	ConversionTemplate.scts	Converte esquemas do banco de dados de origem e aplica o código convertido ao banco de dados de destino. Opcionalmente, salva o código convertido como um script SQL e salva

Tipo de modelo	Nome do arquivo	Descrição
		o relatório de avaliação. Para ter mais informações, consulte Como converter esquemas de bancos de dados .
GenericAppConversion	GenericApplicationConversionTemplate.scts	Converte o código SQL incorporado em seus aplicativos com o conversor genérico de AWS SCT aplicativos. Para ter mais informações, consulte Como converter o código SQL em seus aplicativos .
HadoopMigration	HadoopMigrationTemplate.scts	Migra seu cluster Hadoop on-premises para o Amazon EMR. Para ter mais informações, consulte Como usar o Apache Hadoop como origem para AWS SCT .
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Retoma uma migração interrompida do seu cluster Hadoop on-premises para o Amazon EMR. Para ter mais informações, consulte Como usar o Apache Hadoop como origem para AWS SCT .

Tipo de modelo	Nome do arquivo	Descrição
Informatica	InformaticaConversionTemplate.scts	Converte o código SQL incorporado em seus scripts de extração, transformação e carregamento (ETL) da Informatica. Configura conexões com seus bancos de dados de origem e de destino em seus scripts ETL e salva os scripts convertidos após a conversão. Para ter mais informações, consulte Como converter scripts de ETL da Informatica .
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Converte o código SQL incorporado em seus aplicativos C#, C++, Java e Pro*C com o conversor genérico de aplicativos do AWS SCT. Para ter mais informações, consulte Como converter o aplicativo SQL .
OozieConversion	OozieConversionTemplate.scts	Converte seus fluxos de trabalho do Apache Oozie em AWS Step Functions Para ter mais informações, consulte Como usar o Apache Oozie como origem para AWS SCT .

Tipo de modelo	Nome do arquivo	Descrição
RedshiftAgent	DWHDataMigrationTemplate.scts	Converte esquemas do data warehouse de origem e aplica o código convertido ao banco de dados Amazon Redshift de destino. Em seguida, registra um agente de extração de dados, cria e inicia uma tarefa de migração de dados. Para ter mais informações, consulte Migração de um data warehouse para o Amazon Redshift .
ReportCreation	ReportCreationTemplate.scts	Cria um relatório de migração de banco de dados para vários esquemas de banco de dados de origem. Em seguida, salva esse relatório como um arquivo CSV ou PDF. Para ter mais informações, consulte Relatórios de avaliação de migração .
SQL ScriptConversion	SQLScriptConversionTemplate.scts	Converte scripts SQL*Plus ou TSQL em PL/SQL e salva os scripts convertidos. Além disso, salva um relatório de avaliação.

Depois de baixar o modelo de AWS SCT CLI, use o editor de texto para configurar o script para ser executado nos bancos de dados de origem e destino. Em seguida, use o modo de script AWS SCT CLI para executar seu script. Para ter mais informações, consulte [AWS SCT Modo de script CLI](#).

Exemplos

O exemplo a seguir baixa todos os modelos para a pasta C:\SCT\Templates.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

O exemplo a seguir baixa todos os modelos para a operação `ConversionApply` para a pasta C:\SCT\Templates.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

Editando AWS SCT cenários de CLI

Depois de baixar os modelos de cenário, configure-os para obter scripts funcionais que possam ser executados em seus bancos de dados.

Para todos os modelos, certifique-se de fornecer o caminho para os drivers dos bancos de dados de origem e de destino. Para ter mais informações, consulte [Baixar os drivers de banco de dados necessários](#).

Certifique-se de incluir as credenciais do banco de dados para os bancos de dados de origem e de destino. Além disso, certifique-se de configurar regras de mapeamento para descrever um par de origem e destino para seu projeto de conversão. Para ter mais informações, consulte [Criação de regras de mapeamento](#).

Em seguida, configure o escopo das operações a serem executadas. Você pode remover os comandos que não usa ou adicionar novos comandos ao script.

Por exemplo, suponha que você planeje converter todos os esquemas em seu banco de dados Oracle de origem para PostgreSQL. Em seguida, você planeja salvar o relatório de avaliação da migração do banco de dados em PDF e aplicar o código convertido ao banco de dados de destino. Nesse caso, você pode usar o modelo para a operação `ConversionApply`. Use o procedimento a seguir para editar seu modelo de AWS SCT CLI.

Para editar o modelo de AWS SCT CLI para a operação **ConversionApply**

1. Abra o `ConversionTemplate.scts` que você baixou. Para ter mais informações, consulte [Exemplos](#).

2. Remova `CreateFilter`, converta as operações `ApplyToTarget -filter`, `-filter`, `SaveTargetSQL`, `SaveTarget SQL` e `SaveReportCSV` do seu script. `LbyStatement`
3. Para `oracle_driver_file` na `SetGlobalSettings` operação, insira o caminho para seu driver Oracle. Em seguida, para `postgres_driver_file`, insira o caminho para seu driver PostgreSQL.

Se você usa outros mecanismos de banco de dados, use nomes apropriados para as configurações. Para obter uma lista completa das configurações globais que você pode definir na `SetGlobalSettings` operação, consulte Matriz de configurações globais no [Material de referência](#).

4. (Opcional) Para `CreateProject`, insira o nome do seu projeto e a localização do arquivo do projeto local. Se optar por continuar com os valores padrão, certifique-se de que AWS SCT pode criar arquivos na pasta `C:\temp` sem solicitar direitos de administrador.
5. Para `AddSource`, insira o endereço IP do seu servidor de banco de dados de origem. Digite o nome de usuário, a senha e a porta para se conectar ao servidor de banco de dados de origem.
6. Para `AddTarget`, insira o endereço IP do seu servidor de banco de dados de destino. Além disso, digite o nome de usuário, a senha e a porta para se conectar ao servidor de banco de dados de destino.
7. (Opcional) Para `AddServerMapping`, insira os objetos do banco de dados de origem e destino que você deseja adicionar a uma regra de mapeamento. Você pode usar os parâmetros `sourceTreePath` e `targetTreePath` para especificar o caminho para os objetos do banco de dados. Opcionalmente, você pode usar `sourceNamePath` e `targetNamePath` para especificar os nomes dos objetos do banco de dados. Para obter mais informações, consulte Comandos de mapeamento de servidor em [Material de referência](#).

Os valores padrão da `AddServerMapping` operação mapeiam todos os esquemas de origem com seu banco de dados de destino.

8. Salve o arquivo e use o modo de script para executá-lo. Para ter mais informações, consulte [Modo de script](#).

AWS SCT Modo de script CLI

Depois de criar um script de AWS SCT CLI ou editar um modelo, você pode executá-lo com o `RunSCTBatch` comando. Certifique-se de salvar seu arquivo com o script CLI como uma extensão `.scts`.

Você pode executar scripts de AWS SCT CLI a partir da app pasta do seu caminho de AWS SCT instalação. No Windows, o caminho de instalação padrão é C:\Program Files\AWS Schema Conversion Tool\. Certifique-se de que essa pasta inclua o arquivo RunSCTBatch.cmd ou o RunSCTBatch.sh. Além disso, essa pasta deve incluir o arquivo AWSSchemaConversionToolBatch.jar.

Como alternativa, você pode adicionar o caminho para o arquivo RunSCTBatch na variável de ambiente PATH do seu sistema operacional. Depois de atualizar a variável de PATH ambiente, você pode executar scripts de AWS SCT CLI de qualquer pasta.

Para executar um script de AWS SCT CLI, use o comando a seguir no Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

No exemplo anterior, substitua *file_path* pelo caminho do seu arquivo por um script.

Para executar um script de AWS SCT CLI, use o comando a seguir no Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

No exemplo anterior, substitua *file_path* pelo caminho do seu arquivo por um script.

Você pode fornecer parâmetros opcionais nesse comando, como credenciais do banco de dados, o nível de detalhes na saída do console e outros. Para obter mais informações, baixe a referência da interface de linha de AWS SCT comando em [Material de referência](#).

Exemplos

O exemplo a seguir executa o script ConversionTemplate.scts na pasta C:\SCT\Templates. Você pode usar esse exemplo no Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

O exemplo a seguir executa o script ConversionTemplate.scts no diretório /home/user/SCT/Templates. Você pode usar esse exemplo no Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT Material de referência CLI

[Você pode encontrar material de referência sobre a interface de linha de AWS Schema Conversion Tool comando \(CLI\) no guia a seguir: Referência de CLI AWS Schema Conversion Tool .](#)

Notas de lançamento para AWS SCT

Esta seção contém notas de lançamento para AWS SCT, começando com a versão 1.0.640.

Notas de lançamento do AWS SCT Build 676

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Oracle	PostgreSQL/ Aurora PostgreSQL	<p>Nova emulação de função integrada para as seguintes funções:</p> <ul style="list-style-type: none"> • <code>SYS.UTL_RAW.BIT_AND(RAW, RAW)</code> • <code>XDB.DBMS_XSLPROCESSOR.CLOB2FILE(CLOB)</code> • <code>XDB.DBMS_XSLPROCESSOR.READ2CLOB(VARCHAR2)</code> • <code>SYS.UTL_RAW.BIT_OR(RAW, RAW)</code> • <code>SYS.UTL_RAW.BIT_COMPLEMENT(RAW)</code> 	Não	Sim
MS SQL Server	Servidor SQL do Amazon RDS	<p>Database Mail not supported</p> <p>Mensagem removida do relatório em PDF</p>	Sim	Sim
Oracle	PostgreSQL/ Aurora	Implementou a conversão de restrições para tabelas particionadas.	Sim	Sim

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
	PostgreSQL			
Oracle	MySQL	Revisão da aplicabilidade do AI-602 na conversão de tabelas	Sim	Sim
MS SQL Server	PostgreSQL/Aurora PostgreSQL	agora suporta a MERGE Declaração no PostgreSQL 15.x	Sim	Sim
Todos	Todos	Conexões JDBC implementadas: propriedades avançadas	Sim	Não
Tudo	Todos	CLI: falha de comando corrigida <code>PrintOLAPTaskStatus</code>	Sim	Não
Teradat	Amazon Redshift	Implementou a conversão de tipo de dados no estilo Teradata.	Sim	Não
Teradat	Amazon Redshift	MERGEConversão incorreta corrigida em SQL/BTEQ.	Sim	Não
Teradat	Amazon Redshift	Implementou a conversão de tipo de dados no estilo Teradata.	Sim	Não
Teradat	Amazon Redshift	Conversões de LEAD/LAG funções implementadas.	Sim	Não
Teradat	Amazon Redshift	Erro corrigidoAI-9996 <code>Transformer error occurred in statement .</code>	Sim	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Teradat	Amazon Redshift	Erro corrigidoAI-9996 Transformer error in selectItem .	Sim	Não
Teradat	Amazon Redshift	Conversão implementada para procedimento de armazenamento parcial: XbiDQM.SpCmprsnDly	Sim	Não
Teradat	Amazon Redshift	UNPIVOTDeclaração implementada com alias.	Sim	Não
Teradat	Amazon Redshift	DeleteDeclaração implementada com várias tabelas de origem.	Sim	Não
Teradat	Amazon Redshift	Corrigir paraAI-9996 Transformer error occurred in functionCallExpression .	Sim	Não
Teradat	Amazon Redshift	Conversão de NORMALIZE cláusula implementada.	Sim	Não
Teradat	Amazon Redshift	Foi corrigida a conversão incorreta em DELETE declarações com subconsultas.	Sim	Não
Teradat	Amazon Redshift	Erro corrigidoAI-9996 Transformer error occurred in tableOperatorSource .	Sim	Não
Teradat	Amazon Redshift	Erro corrigidoAI-9996 Transformer error occurred in additiveExpression .	Sim	Não
Teradat	Amazon Redshift	Implementou a conversão de objetos do sistema DBC.	Sim	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Teradat	Amazon Redshift	Solução alternativa implementada de atualização com predicados de junção implícitos.	Sim	Não
Netezza	Amazon Redshift	Erro de conversão de CREATE MATERIALIZED VIEW declarações corrigido.	Sim	Não
DB2Luv	PostgreSQL/Aurora PostgreSQL	Conexão de opções estendidas do JDBC: opções adicionais de conexão adicionadas.	Sim	Não
DB2Luv	PostgreSQL/Aurora PostgreSQL	Adicionado suporte para MERGE Statement no PostgreSQL 15.x	Sim	Não
DB2Luv	PostgreSQL/Aurora PostgreSQL	GLOBAL TEMPORARY TABLE Conversão implementada.	Sim	Não
DB2Luv	PostgreSQL/Aurora PostgreSQL	USER DEFINED TYPES Conversão implementada.	Sim	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
DB2Luv	MySQL	GLOBAL TEMPORARY TABLE Conversão implementada.	Sim	Não
DB2Luv	MySQL	USER DEFINED TYPES Conversão implementada.	Sim	Não
DB2Luv	MySQL	USER DEFINED FUNCTIONS Conversão implementada.	Sim	Não
DB2Luv	MariaDB	GLOBAL TEMPORARY TABLE Conversão implementada.	Sim	Não
DB2Luv	MariaDB	USER DEFINED TYPES Conversão implementada.	Sim	Não
Sybase	Todos	Suporte adicional para autenticação Kerberos	Sim	Não
DB2Luv	PostgreSQL/ Aurora PostgreSQL	Suporte adicionado para conversão de várias versões para os alvos	Sim	Não
Azure SQL/ Microsoft SQL Server	PostgreSQL/ Aurora PostgreSQL	Suporte adicionado para conversão de várias versões para os alvos	Sim	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
DB2Luv	PostgreSQL/Aurora PostgreSQL	Foi adicionado suporte para MERGE declaração no PostgreSQL 15.x.	Sim	Não
Teradata	Amazon Redshift	Conversão de alteração de função não suportada corrigida.	Sim	Não
Tudo	Amazon Redshift	Extratores de dados: particionamento implementado por uma coluna indexada.	Sim	Não

Notas de lançamento do AWS SCT Build 675

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Cassandra	DynamoDB	Corrigido um bug em que a instalação do Cassandra falhava no data center de destino.	Não
DB2 LUW	PostgreSQL	SQL DINÂMICO: Instrução PREPARE: resolução e conversão sem SQL dinâmico.	Não
DB2 LUW	PostgreSQL	Foi adicionado suporte para SPECIAL REGISTER.	Não
DB2 LUW	PostgreSQL	Atualização do pacote de extensão	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Hadoop	Amazon EMR	Foi adicionado suporte para conexão com um cluster Hadoop por meio do protocolo rsa-sha2.	Não
Microsoft SQL Server	Amazon Redshift	Correção para o driver JDBC forçando o TLS apesar de não estar configurado.	Não
Netezza	Amazon Redshift	Foi adicionado suporte para conversão de visualizações materializadas.	Não
Oracle	Amazon Redshift	Foi adicionado suporte para consultas recursivas no Amazon Redshift.	Sim
Oracle	PostgreSQL, Aurora PostgreSQL	Correção para conversão incorreta do tipo de dados NUMBER.	Sim
Oracle	Amazon Redshift	Migração de dados. Particionamento automático da Oracle. Tempo de expiração adicionado para o valor dos fragmentos da tabela. O tempo de expiração é de 72h. Quando ocorre a expiração, os fragmentos de dados são reconstruídos quando uma tarefa de migração de dados é criada.	Não
Oracle	Amazon Redshift	SCT Data Extractor: mudou a abordagem de upload de dados para o Amazon Redshift. Por padrão, o extrator não cria tabelas preparadas. Em vez disso, depois que todos os arquivos de dados estiverem no bucket do Amazon S3, o extrator os copia para a tabela de destino usando um único comando COPY.	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Oracle	Amazon Redshift	Foi adicionada a migração do tipo de dados RAW para a coluna VARBYTE.	Não
Oracle	PostgreSQL, Aurora PostgreSQL	Conversão de várias versões	Não
Oracle	PostgreSQL	Foi adicionado suporte para a declaração MERGE no PostgreSQL 15.x.	Sim
Oracle	PostgreSQL	Foi adicionado suporte para novas funções de expressão regular no PostgreSQL 15.x.	Sim
Oracle	PostgreSQL, Aurora PostgreSQL	A instrução ON CONFLICT DO UPDATE é convertida sem alias excluído.	Sim
Teradata	Amazon Redshift	Foi adicionado suporte de conversão para funções LEAD/LAG.	Não
Teradata	Amazon Redshift	Conversão aprimorada do tipo de dados com indicação explícita do formato dos dados.	Não
Teradata	Amazon Redshift	Conversão aprimorada da cláusula AT 'TIME ZONE' em expressões de timestamp e timestamp.	Não
Teradata	Amazon Redshift	AI-9996 durante procedimentos de conversão com instruções MERGE.	Não

Notas de lançamento do AWS SCT Build 674

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Todos	Todos	Várias correções de erros e melhorias na performance	Parcial (somente para pares de origem e destino compatíveis)
Azure SQL/Microsoft SQL Server	Amazon Redshift	Foi removida a mensagem “AI 18066: Não é possível converter o nome do esquema” que enganava os usuários durante a avaliação/conversão do esquema	Não
Azure SQL/Microsoft SQL Server	Amazon RDS para MySQL/Amazon Aurora MySQL	Conversão incorreta do procedimento sem atribuir um código de retorno	Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)
Azure SQL/Microsoft SQL Server	Amazon RDS para PostgreSQL/Amazon Aurora PostgreSQL	AI9997 corrigido para alguns casos de conversão da cláusula FOR XML PATH	Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Azure SQL/Micro soft SQL Server	Amazon RDS para PostgreSQL/Amazon Aurora PostgreSQL	O valor é arredondado para a escala original no corpo do procedimento/função	Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)
Azure SQL/Micro soft SQL Server	Amazon RDS para PostgreSQL/Amazon Aurora PostgreSQL	Várias melhorias na conversão de instruções EXECUTE	Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)
Azure SQL/Micro soft SQL Server/Azure Synapse	Amazon Redshift	Conversão aprimorada das seguintes instruções e modos: <ul style="list-style-type: none"> • EXCEPTION BLOCK • CONFIRMAÇÃO AUTOMÁTICA • NONATOMIC • CONJUNTO DE AGRUPAMENTO • CUBE • ROLLUP 	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 LUW	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Várias correções nas consultas SQL de carregamento de metadados	Não
DB2 LUW	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	O AI 9996 não é esperado em acionadores	Não
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Função analítica ROWNUMBER	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Suporte constante de string hexadecimal	Não
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Várias correções nas consultas SQL de carregamento de metadados.	Não
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Suporte à sequência de referência NEXT VALUE FOR	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Suporte à opção DB2_NUMBER_ROWS para a instrução GET DIAGNOSTICS	Não
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Várias instruções GET DIAGNOSTICS	Não
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Erros corrigidos na conversão da instrução FOR.	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Oracle	Amazon RDS para MySQL/ Amazon Aurora MySQL	Corrigido o erro quando o nó de parâmetros da função de pacote não estava definido.	Sim
Oracle	Amazon RDS para PostgreSQL/ Amazon Aurora PostgreSQL	Erros corrigidos na função AWS_ORACLE_EXT.NEXT_DAY do pacote de extensão	Sim
Oracle	Amazon RDS para PostgreSQL/ Amazon Aurora PostgreSQL	Foram corrigidos vários bugs com a conversão “(+)” nas junções externas da Oracle	Sim
Oracle		Suporte à autenticação Kerberos	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
SAP ASE	Amazon RDS para PostgreSQL/ Amazon Aurora PostgreSQL	Corrigido o erro ao converter mais de um identificador na cláusula FROM da instrução UPDATE	Não
SAP ASE	Amazon RDS para PostgreSQL/ Amazon Aurora PostgreSQL	Corrigido o erro com a conversão de comentários e instruções em várias linhas	Não
SAP ASE		Foi adicionado suporte para o parâmetro ENCRYPT_PASSWORD durante a conexão	Não
Teradata	Amazon Redshift	Conversão aprimorada da tabela VOLATILE com nome de esquema especificado	Não
Teradata	Amazon Redshift	Conversão incorreta WHERE CLAUSE em CTE complexo	Não
Teradata	Amazon Redshift	Foi adicionado suporte para o tipo de dados INTERVAL ao migrar dados usando agentes de extração de dados SCT.	Não

Origem	Destino	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Scripts do Teradata BTEQ	Scripts do Amazon Redshift RSQL	Parâmetros de conversão incorretos no procedimento executado pelo BTEQ	Não

Notas de lançamento do AWS SCT Build 673

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Melhorias no desempenho geral e correções de erros
Azure SQL/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Conversão de chamada de função incorreta corrigida
Azure SQL/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Conversão implementada da cláusula FOR XML
Azure SQL/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS	Conversão da cláusula FOR XML com alias errado.

Origem	Destino	Novidades, aprimoramentos ou correções
	PostgreSQL	
Azure SQL/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Corrigido o erro ao AWS SCT não converter EXECUTE declarações que executam uma cadeia de caracteres com parâmetros de procedimento.
Azure SQL/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Conversão aprimorada de instruções UPDATE com junções internas.
Azure Synapse	Amazon Redshift	Foi corrigida a conversão incorreta da função OBJECT_ID incorporada.
IBM Db2 para z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementada a conversão das seguintes instruções e objetos: <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • Restrições PK e UNIQUE em tabelas particionadas • Função do TIMESTAMPDIFF • Função do TO_DATE • Função do EBCDIC_STR • Função do VARCHAR_FORMAT

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Corrigido o erro quando o índice baseado em funções ignora as funções após a conversão.
IBM Db2 para z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Corrigido o erro em que a instrução REPEAT fechava com AI 9996 após a conversão
IBM Db2 para z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Corrigido o erro em que a FINAL TABLE cláusula fechava com 9996.
IBM Db2 para z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	LOADER Chave de particionamento na restrição de referências. AWS SCT agora consegue converter chaves primárias e restrições exclusivas em tabelas particionadas como índices secundários.
IBM Db2 para z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Suporte à função PostgreSQL.VARCHAR_FORMAT

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementada a alteração de agrupamento nos comandos SCT CLI <code>CreateTransformationRule</code> e <code>ModifyTransformationRule</code> .
Greenplum	Amazon Redshift	Corrigido o erro com a chamada incorreta do procedimento armazenado após a conversão
Hadoop	Amazon EMR	Foi adicionado suporte para conexão com um cluster Hadoop usando o protocolo <code>rsa-sha2</code> .
Hadoop	Amazon EMR	Foi adicionado suporte para o Amazon EMR com metastore que não é do GLUE Hive,
Oracle	Amazon Redshift	Corrigido o erro com a conversão incorreta da consulta recursiva em que a coluna <code>PRIOR</code> não estava na lista <code>SELECT</code> .
Oracle	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementado retorno a um elemento de uma matriz associativa
Oracle	Aurora PostgreSQL/Amazon RDS PostgreSQL	Corrigido o inesperado <code>AI 9996</code> em <code>UNPIVOT</code> com colchetes

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Corrigido o inesperado AI 9996 em UNPIVOT com UNION ALL
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	melhorias para conversões de tipos de dados Number
Oracle	Extrator de dados do Amazon Redshift	Suporte para particionamento automático para tabelas Oracle. Otimização para criar tarefas de migração.
Teradata	Amazon Redshift	Implementar a conversão da instrução EXCEPTION BLOCK
Teradata	Amazon Redshift	Suporte para conversão de predicados ALL, ANY, e SOME para o Amazon Redshift.
Teradata	Amazon Redshift	Foi adicionado suporte nativo para predicado QUALIFY.
Teradata	Amazon Redshift	<p>Conversão aprimorada do seguinte:</p> <ul style="list-style-type: none"> • Consultas recursivas • GROUPING SET • CUBE • ROLLUP • Instrução UPDATE com junção implícita

Origem	Destino	Novidades, aprimoramentos ou correções
Fontes OLAP	Extrator de dados do Amazon Redshift	Comandos CLI implementados para Stop/Resume para tarefas do Extrator de dados do Amazon Redshift.
Fontes OLAP	Extrator de dados do Amazon Redshift	Foi adicionada a capacidade de selecionar as colunas da tabela que precisam ser migradas durante a configuração da tarefa de migração.

Notas de lançamento do AWS SCT Build 672

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Amazon RDS para PostgreSQL	Implementou o suporte à versão principal 15 do PostgreSQL como destino de migração.
Todos	Amazon Redshift	Foi adicionado um novo <code>PrintTaskStatus</code> comando na interface de linha de AWS SCT comando (CLI) para exibir o status da tarefa de migração de dados.
Todos	Amazon Redshift	Melhorou o fluxo de configuração dos agentes de extração de dados.
Todos	Amazon Redshift	Resolveu um erro em que os agentes de extração de dados não exibiam as informações sobre as subtarefas.
Apache Oozie	AWS Step Functions	Foi adicionada uma opção para salvar as definições da máquina de estado como um script no código convertido.
Banco de dados do SQL Azure	Aurora PostgreSQL	Implementada a conversão de funções <code>COALESCE</code> , <code>DATEADD</code> , <code>GETDATE</code> e <code>SUM</code> .

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Melhorada a conversão de instruções UPDATE com cláusulas JOIN e OUTPUT.
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvido um erro que ocorria durante a conversão da instrução SELECT TOP 1 WITH TIES.
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Foram resolvidos vários problemas que ocorreram durante a conversão das cláusulas FOR XML em funções integradas.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Implementada a conversão de instruções GET DIAGNOSTICS e RAISE usando um bloco EXCEPTION nativo do Amazon Redshift.
Greenplum	Amazon Redshift	Melhorada a conversão de procedimentos armazenados adicionando suporte a um bloco EXCEPTION no código convertido.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Corrigido um erro em que a função TO_CHAR com modelos de formato de hora era convertida incorretamente.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementou a conversão de expressões de tabela aninhadas.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão das instruções GOTO, MERGE, REPEAT e SIGNAL.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de instruções FETCH com palavras-chave de orientação BEFORE e AFTER.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão das referências da tabela FINAL TABLE e OLD TABLE.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementou a conversão das seguintes funções. <ul style="list-style-type: none"> • ADD_MONTHS • DAY com parâmetros do tipo de dados do caractere • 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 L PostgreSQL L	Melhorada a conversão de um alias de uma subconsulta em cláusulas JOIN.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de funções COALESCE.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de índices EXPLICIT.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Foi aprimorada a conversão de nomes de colunas em expressões compostas para resolver um problema em que o item de ação 9997 aparece inesperadamente durante a conversão.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de chaves primárias e restrições exclusivas.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções XMLTABLE em instruções INSERT para resolver um problema em que o item de ação 9996 aparece inesperadamente durante a conversão.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de funções com o argumento SUBSTR.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão do registro especial CURRENT_TIMESTAMP .
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções MERGE, instruções sem suporte e funções internas sem suporte.
Microsoft SQL Server	Todos	Adicionado suporte ao Microsoft SQL Server versão 2022 como uma origem.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Melhorou a conversão de SELECT declarações que usam operadores de concatenação de strings. AWS SCT usa a STRING_AGG função no código convertido.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Implementado o suporte da nova versão 3.1.0 do arquivo de configuração de atributos do Babelfish. Esse arquivo define atributos SQL que são suportados e não suportados por versões específicas do Babelfish.

Origem	Destino	Novidades, aprimoramentos ou correções
Netezza	Amazon Redshift	Resolveu um problema em que os agentes de extração de dados não iniciavam a migração de dados do ponto CDC especificado.
Oracle	Todos	Atualizou o relatório de avaliação dos bancos de dados Oracle versão 19 como origem.
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementou a conversão do DBMS_OUTPUT pacote adicionando novas funções ao pacote AWS SCT de extensão.
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de funções e procedimentos que usam matrizes associativas como argumentos ou parâmetros.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de cláusulas DISTINCT em instruções SELECT.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de tabelas em que a restrição de chave primária tem o mesmo nome da tabela.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão do procedimento RAISE_APPLICATION_ERROR com o terceiro parâmetro.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que a regra de migração não alterava automaticamente o tipo de dados NUMERIC para INTEGER onde aplicável.
Oracle DW	Amazon Redshift	Implementado o suporte às cláusulas CONNECT BY nativas do Amazon Redshift no código convertido.
Oracle DW	Amazon Redshift	Melhorou a migração de dados adicionando automaticamente uma subtarefa para cada tabela ou partição no escopo da migração. Essa abordagem evita a perda de dados dos dados inseridos após o particionamento.
Teradata	Amazon Redshift	Implementou a conversão de visualizações recursivas.
Teradata	Amazon Redshift	Melhorada a conversão de procedimentos armazenados que usam modos de transação BTET e ANSI adicionando suporte ao modo de transação AUTOCOMMIT nativo do Amazon Redshift.
Teradata	Amazon Redshift	Melhorada a conversão de procedimentos armazenados que usam a semântica da transação TERADATA adicionando a palavra-chave NONATOMIC no código convertido.
Teradata	Amazon Redshift RSQL	Resolveu um problema em que o código convertido incluía o ID da chave de AWS acesso e a chave de acesso secreta.

Notas de lançamento do AWS SCT Build 671

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Corrigido um erro em que AWS SCT não tinha permissões para salvar um arquivo de projeto no Windows.
Todos	Todos	<p>Os seguintes modelos de interface de linha de AWS SCT comando (CLI) foram atualizados.</p> <ul style="list-style-type: none"> • BTEQ ScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigração • Informatica <p>Para obter mais informações sobre os modelos de AWS SCT CLI, consulte. Obter cenários de CLI</p>
Todos	Amazon Redshift	Corrigido um erro em AWS SCT que não criava um pacote de extensão na interface de linha de comando (CLI).
Todos	Amazon Redshift	Resolveu um problema em que AWS SCT os agentes de extração de dados não usavam a AWS Snowball configuração na interface de linha de comando (CLI).
Apache Oozie	AWS Step Functions	Suporte implementado para a migração do Apache Oozie para o AWS Step Functions modo de interface de linha de comando (CLI). Depois de migrar suas workload do Hadoop para o Amazon EMR, agora você pode migrar o sistema de agendamento de fluxo de trabalho para o Nuvem AWS. Para ter mais informações, consulte Como converter o Apache Oozie para AWS Step Functions .
Banco de dados do SQL Azure	Aurora PostgreSQL	Corrigido um erro de resolução que ocorria para tabelas e aliases.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Implementada a conversão de cláusulas INDEX ON.
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Foi aprimorada a conversão dos seguintes objetos para evitar itens de ação inesperados.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Declarações em lote • Listas de expressões • Alias de tabelas • Tabelas temporárias • Acionadores • Variáveis do usuário
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvido um erro de análise que ocorreu para procedimentos.
Microsoft SQL Server	PostgreSQL	

Origem	Destino	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQL	Corrigido um erro em que AWS SCT eram usados nomes incorretos de tabelas temporárias no código convertido para OBJECT_ID funções.
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão dos seguintes elementos de código.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Funções do CONVERT • Funções do DATEADD • Instruções DELETE dentro de funções embutidas • Instruções IF • Ações INSERT ou UPDATE em uma coluna • Instruções RETURN • Instruções UPDATE com consultas ou funções complexas
BigQuery	Amazon Redshift	Foi adicionado suporte BigQuery como fonte para o processo de avaliação de vários servidores. Para ter mais informações, consulte Como criar um relatório de avaliação multisservidor .
Hadoop	Amazon EMR	Atualizou a versão do driver JDBC Apache Hive compatível que você usa para se conectar aos seus bancos de dados de origem. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários .

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	O carregador de metadados de origem foi aprimorado para garantir que ele AWS SCT carregue objetos do banco de dados de origem, como chaves primárias, índices implícitos e assim por diante.
IBM DB2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Corrigido um erro de resolução que ocorria para colunas em cursores implícitos.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a capacidade de manter a formatação de nomes de colunas, expressões e cláusulas em instruções DML no código convertido.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementou a conversão de chaves estrangeiras de esquema cruzado.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de funções LENGTH e VARCHAR.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de instruções LABEL ON e DECLARE CONDITION .
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de instruções SELECT com cláusulas OPTIMIZE FOR.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções CREATE TABLE adicionando valores padrão para todos os tipos de dados compatíveis.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de atributos INCREMENT BY.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de tabelas particionadas adicionando a capacidade de excluir partições de tabela do escopo de conversão.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão das definições de chave primária com colunas INCLUDE.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão da função SUBSTRING .
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções SET e DECLARE HANDLER FOR.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de tipos de dados variáveis.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de funções XMLTABLE.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorado o fluxo de migração implementando a seguinte ordem de aplicação de objetos convertidos ao banco de dados de destino: tabelas, partições, índices, restrições, chaves estrangeiras e acionadores.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão de comentários no código-fonte.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 9997 aparecia inesperadamente durante a conversão de aliases em cláusulas FROM.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 9997 aparecia inesperadamente durante a conversão de aliases do cursor.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Corrigido um erro em que o código convertido retornava resultados diferentes para instruções SELECT com cláusulas ORDER BY. Como o SQL Server e o PostgreSQL tratam os valores NULL de forma diferente, o código convertido agora inclui cláusulas NULLS FIRST ou NULLS LAST que garantem que o código convertido retorne os resultados na mesma ordem do código-fonte.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Resolveu um problema em que tipos de dados em funções de tabelas eram convertidos incorretamente.
MySQL	Amazon RDS para MySQL	Resolvido um problema em que aspas simples (' ') apareciam inesperadamente nos nomes dos objetos do banco de dados no código convertido.
Oracle	Aurora PostgreSQL PostgreSQL	Foram adicionadas novas visualizações ao pacote de extensões para emular as visualizações do sistema Oracle que você usa para exibir informações sobre partições e subpartições.
Oracle	Aurora PostgreSQL PostgreSQL	Foram atualizadas duas funções no pacote de extensões para adicionar nomes de esquemas como argumentos no código convertido.
Oracle	Aurora PostgreSQL PostgreSQL	Corrigido um erro em AWS SCT que não usava os parâmetros corretos para a conversão de aplicativos C++ após atualizar o código do aplicativo na interface do usuário.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções CREATE TYPE para evitar exceções inesperadas.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de tabelas aninhadas.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolveu um erro de análise que ocorria nos objetos do pacote.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolveu um problema em que os nomes de objetos eram cortados AWS SCT inesperadamente no código convertido quando o tamanho do nome excedia 60 caracteres.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolveu um problema em que acionadores em nível de linha para tabelas particionadas eram convertidos incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle DW	Amazon Redshift	Suporte implementado de particionamento automático de tabelas para migração de dados. Para acelerar a migração de dados, AWS SCT pode particionar automaticamente grandes tabelas ou partições com base nos valores na ROWID pseudocoluna. Para ter mais informações, consulte Como usar o particionamento nativo .
Teradata	Amazon Redshift	Implementado o suporte de comandos MERGE nativos no código convertido do Amazon Redshift. Para obter mais informações sobre o comando MERGE no Amazon Redshift, consulte MERGE no Guia do desenvolvedor de banco de dados do Amazon Redshift.
Teradata	Amazon Redshift	Melhorada a conversão de instruções DELETE e UPDATE que não usam nomes de tabela explícitos.
Teradata	Amazon Redshift	Resolvido um problema em que as instruções IN e NOT IN eram convertidas incorretamente.

Notas de lançamento do AWS SCT Build 670

Origem	Destino	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão dos seguintes elementos de código.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> Instruções CREATE INDEX dentro de instruções INCLUDE Instruções DECLARE Instruções DECLARE . . . TABLE DECLARE com valores padrão dentro das instruções LOOP Instruções DELETE Instruções DROP CONSTRAINT dentro de instruções ALTER TABLE EXECUTE AS CALLER e REVERT

Origem	Destino	Novidades, aprimoramentos ou correções
		<ul style="list-style-type: none"> • Instruções IIF • Listas de expressões • Funções do MONTH() • Instruções UPDATE • Funções do YEAR()
Azure Synapse Analytics	Amazon Redshift	Adicionado suporte ao Azure Synapse Analytics como origem para o processo de avaliação de vários servidores. Para obter mais informações, consulte Como criar um relatório de avaliação multisservidor .
Hadoop	Amazon EMR	Suporte implementado para a migração de clusters Hadoop para o Amazon EMR no modo de interface de linha de comandos (CLI). Para obter mais informações, consulte Como migrar estruturas de big data .
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Corrigido um erro de resolução que ocorria para tabelas e colunas.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de expressões CASE.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de referências CURRENT_DATE em registros especiais. Uma referência a um registro especial no Db2 para z/OS é uma referência a um valor fornecido pelo servidor atual.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de funções DATE e POSSTR.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de constantes de data e hora.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de valores padrão para colunas dos seguintes tipos de dados: DATE, TIME, TIMESTAMP e TIMESTAMP WITH TIME ZONE.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções SELECT INTO.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de funções DATEDIFF.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Corrigido um erro em que as funções ISNULL eram convertidas em NULLIF. Como resultado, o código convertido produzia resultados diferentes em comparação com o código-fonte. Agora, AWS SCT converte ISNULL funções em. COALESCE
Netezza	Amazon Redshift	Agentes de extração de dados aprimorados para resolver um problema em que o status de falha foi definido para tarefas concluídas com êxito.
Netezza	Amazon Redshift	Foi adicionada a capacidade de alterar endpoints em subtarefas após iniciar uma migração de dados com agentes de extração de dados.
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Foi adicionada a capacidade de se conectar a bancos de dados usando um protocolo de endereço IPv6.
Oracle	Amazon RDS para Oracle	Implementada a conversão do pacote DBMS_JOB que agenda e gerencia trabalhos na fila de trabalhos.
Oracle	Aurora PostgreSQL PostgreSQL	Foram adicionadas novas funções ao pacote de extensões para melhorar a conversão das tabelas globais aninhadas. Essas novas funções emulam funções DELETE, EXTEND, e TRIM em seu código-fonte Oracle.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Foi adicionada a capacidade de especificar o escopo de conversão para o código SQL incorporado em aplicativos Java. Agora você pode excluir os subconjuntos do projeto do aplicativo de origem do escopo da conversão. Para obter mais informações, consulte Como converter o código SQL em seu aplicativo Java na AWS SCT .
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de operadores de concatenação () dentro de índices funcionais.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de condições IN em que seu código-fonte não inclui parênteses para uma única expressão.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções MERGE para INSERT ON CONFLICT no PostgreSQL.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um erro de análise que ocorreu para pacotes de procedimentos.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 5072 aparecia inesperadamente durante a conversão de instruções.
Oracle DW	Amazon Redshift	Corrigido um erro em AWS SCT que não aplicava o pacote de extensão ao aplicar o código convertido ao banco de dados de destino.
Oracle DW	Amazon Redshift	Corrigido um erro em AWS SCT que alguns dos arquivos do pacote de extensões não eram aplicados ao usar o assistente do pacote de extensões.
Oracle DW	Amazon Redshift	Resolveu um problema em que AWS SCT não era possível processar a migração de dados AWS Snowball com mais de 500 tarefas em execução paralelamente.
Oracle DW	Amazon Redshift	Resolveu um problema em que funções definidas pelo usuário com tipos definidos pelo usuário eram convertidas incorretamente.

Notas de lançamento do AWS SCT Build 669

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Melhorado o processo de avaliação de vários servidores, o que ajuda a determinar a plataforma de banco de dados de destino ideal para seus bancos de dados de origem. Agora, AWS SCT ignora a AWS Secrets Manager chave se você fornecer as credenciais do banco de dados no arquivo de valores separados por vírgula (CSV) de entrada. Para ter mais informações, consulte Como criar um relatório de avaliação multisservidor .

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Resolveu um problema em que o relatório de avaliação de vários servidores incluía o endereço IP do seu banco de dados de origem ao usar um código secreto AWS Secrets Manager para se conectar ao banco de dados.
Todos	Amazon Redshift	Implementou a configuração automática das configurações da máquina virtual Java (JVM), dependendo do sistema operacional e da RAM disponível. AWS SCT usa essa JVM para executar o trabalho dos agentes de extração de dados.
Todos	Amazon Redshift	Resolvido um problema em que os agentes de extração de dados não iniciavam no Ubuntu.
Todos	Amazon Redshift	Resolvido um problema em que as tarefas de extração de dados não iniciavam após a execução do arquivo StartAgent.bat no Windows.
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvido um problema em que os nomes das colunas eram convertidos incorretamente com a opção Gerar nomes exclusivos para índices ativada.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Conversão implementada de funções que retornam VOID aos procedimentos.
Greenplum	Amazon Redshift	Resolveu um problema em que a migração de dados falhava quando o banco de dados de origem não incluía valores numéricos (NaN) em colunas numéricas. AWS SCT os agentes de extração de dados agora substituem os valores NaN por NULL.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Foi adicionada uma nova configuração de conversão para especificar as opções DATE FORMAT e TIME FORMAT durante a conversão das funções CHAR integradas.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Foi adicionado um item de ação 8534 para a conversão de cursores predefinidos que foram declarados com a cláusula WITHOUT RETURN. Se o cursor não retornar conjuntos de resultados, AWS SCT atribui um NULL valor ao nome do cursor no código convertido e gera um item de ação.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Editou a propriedade CURRENT CLIENT_APPLNAME que identifica AWS SCT durante a conexão com o banco de dados de origem.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Implementada uma nova configuração de conversão para especificar as opções DATE FORMAT e TIME FORMAT durante a conversão das funções CHAR integradas.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Implementada a conversão de instruções LEAVE em instruções do bloco BEGIN . . . END .

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de funções XMLPARSE, XMLTABLE e XMLNAMESPACES .
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de funções CHAR integradas.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Conversão aprimorada de cursores.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções loop FOR.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão do uso de tipos de tabela em instruções SELECT.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Implementado o suporte da nova versão 2.2.0 do arquivo de configuração de atributos do Babelfish. Esse arquivo define atributos SQL que são suportados e não suportados por versões específicas do Babelfish.
Netezza	Amazon Redshift	Agentes de extração de dados aprimorados para resolver um problema em que uma linha não foi excluída da tabela de destino durante a replicação contínua de dados.
Oracle	Amazon RDS para Oracle	Conversão aprimorada dos recursos do Oracle Database Enterprise Edition.
Oracle	Aurora PostgreSQL PostgreSQL	Implementada a conversão de funções GROUPING_ID .
Oracle	Aurora PostgreSQL PostgreSQL	Conversão aprimorada de código SQL em aplicativos C# adicionando o suporte ao mapeamento personalizado de tipos de dados no modo de interface de linha de comandos (CLI).
Oracle	Aurora PostgreSQL PostgreSQL	Conversão aprimorada de tabelas aninhadas para evitar um item de ação 9996 inesperado.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que a chamada de um construtor de objetos era convertida incorretamente.
Oracle DW	Amazon Redshift	Suporte implementado de particionamento de tabelas existentes para migração de dados. Para acelerar a migração de dados, AWS SCT cria subtarefas para cada partição da tabela de origem que não esteja vazia. Para ter mais informações, consulte Como usar o particionamento nativo .
Teradata	Amazon Redshift	Melhorada a conversão de funções CAST com argumentos TIME WITH TIME ZONE AS TIMESTAMP , TIME WITH TIME ZONE AS CHAR e TIMESTAMP AS TIME WITH TIME ZONE.
Teradata	Amazon Redshift	Melhorada a conversão de funções CAST com a opção FORMAT.
Teradata	Amazon Redshift	Resolvido um problema em que as funções CEIL não eram convertidas.
Teradata	Amazon Redshift	Resolvido um problema em que instruções MERGE com cláusulas DELETE eram convertidas incorretamente.
Teradata	Amazon Redshift	Resolvido um problema em que funções TO_CHAR com argumentos de data e formato eram convertidas incorretamente.

Notas de lançamento do AWS SCT Build 668

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Amazon Redshift	Resolvido um problema em que os operadores de multiplicação nas regras de migração não funcionavam corretamente. Esses operadores possibilitam alterar o comprimento de tipos de dados

Origem	Destino	Novidades, aprimoramentos ou correções
		char, varchar, nvarchar e string. Para obter mais informações, consulte Como criar regras de migração .
Azure Synapse Analytics	Amazon Redshift	Suporte implementado de funções CONVERT com argumentos VARCHAR.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções SELECT com cláusulas NOLOCK.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções UPDATE com aliases ou com cláusulas SET e FROM.
Greenplum	Amazon Redshift	Implementado o particionamento virtual automático para migração de dados. AWS SCT usa a coluna GP_SEGMENT_ID do sistema para criar partições.
Greenplum	Amazon Redshift	Suporte implementado de cláusulas RETURN QUERY e RETURN SETOF.
Greenplum	Amazon Redshift	Suporte implementado de funções SUBSTRING com três parâmetros.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de funções SUBSTR com parâmetros LOCATE.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Foi adicionada uma opção para usar uma matriz de variáveis REFCURSOR para retornar conjuntos de resultados dinâmicos. Quando você seleciona essa opção nas configurações de conversão, AWS SCT adiciona um parâmetro adicional OUT no código convertido.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Suporte implementado para instruções de loop FOR.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Suporte implementado de funções XMLPARSE. Foi adicionado um item de ação 8541 para a distribuição de espaço em branco nas funções XMLPARSE.
IBM Db2 para z/OS	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de vários manipuladores de exceções em um único bloco BEGIN . . . END.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversão de acionadores INSERT e DELETE aprimorados.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Foi aprimorada a conversão de chamadas de procedimentos aninhados.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de tabelas.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que operações lógicas NOT bit a bit eram convertidas incorretamente em valores inteiros.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que matrizes locais não eram inicializadas na versão 8.0.2 e inferior do PostgreSQL.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que instruções MERGE com cláusulas WHEN NOT MATCHED BY SOURCE eram convertidas incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
MySQL	Aurora MySQL	Resolveu um problema em que determinava AWS SCT incorretamente as permissões de usuário concedidas pela <code>rds_superuser_role</code> função.
Netezza	Amazon Redshift	O carregador de metadados de origem foi aprimorado para garantir que os objetos do banco de dados sejam carregados AWS SCT corretamente com nomes em minúsculas.
Oracle	Aurora PostgreSQL L PostgreSQL L	Foram adicionadas novas funções ao pacote de extensões para melhorar a conversão das tabelas locais aninhadas. Essas novas funções emulam funções PRIOR, NEXT, LIMIT, FIRST, LAST, EXISTS, EXTEND, TRIM, DELETE e SET em seu código-fonte Oracle. Para obter mais informações, consulte Como usar pacotes de extensão .
Oracle	Aurora PostgreSQL L PostgreSQL L	Foi adicionada a capacidade de especificar o escopo de conversão para aplicativos C#. Agora usuários podem excluir os subconjuntos do projeto do aplicativo de origem do escopo da conversão.
Oracle	Aurora PostgreSQL L PostgreSQL L	Suporte implementado de métodos COUNT em coleções.
Oracle	Aurora PostgreSQL L PostgreSQL L	Suporte implementado de variáveis e construtores em tabelas aninhadas.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Suporte implementado de funções <code>RATIO_TO_REPORT</code> e <code>STANDARD_HASH</code> .
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de objetos grandes (LOBs) como parte do pacote de extensão AWS SCT .
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversão aprimorada de coleções locais.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções <code>JOIN</code> com cláusulas <code>USING</code> em que os nomes das colunas não incluem o nome da tabela.
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de funções <code>EMPTY_BLOB</code> e <code>EMPTY_CLOB</code> .

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de variáveis de ligação posicional em aplicativos C#.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Conversão implementada de acionadores de vários eventos.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Implementada a conversão de acionadores recursivos.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de acionadores com a variável global @@rowcount .
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que funções agregadas na cláusula SET de instruções UPDATE eram convertidas incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que o item de ação 42702 aparecia inesperadamente durante a conversão de instruções UPDATE.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que funções CONVERT com argumentos CHAR eram convertidas incorretamente.
Snowflake	Amazon Redshift	Foi adicionado suporte ao Snowflake como fonte para migração de dados com agentes de extração AWS SCT de dados. Para ter mais informações, consulte Como migrar dados de um data warehouse on-premises para o Amazon Redshift .
Teradata	Amazon Redshift	Melhorada a conversão de funções CAST com argumentos TIMESTAMP AS TIME WITH TIMEZONE.

Notas de lançamento do AWS SCT Build 667

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Suporte implementado para scripts de extração, transformação e carregamento (ETL) da Informatica no modo de interface de linha de comando (CLI). AWS SCT redireciona automaticamente seus scripts ETL da Informatica para o novo banco de dados de destino. Além disso, AWS SCT converte nomes de objetos e código SQL incorporados em seus objetos da Informatica. Para ter mais informações, consulte Como converter scripts de ETL da Informatica .

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Amazon Redshift	Aumentada a versão mínima do driver compatível com o Amazon Redshift para 2.1.0.9. Para obter mais informações, consulte Baixar os drivers de banco de dados necessários .
Azure Synapse Analytics	Amazon Redshift	Foi adicionada uma nova função ao pacote de extensão para melhorar a conversão da função CONVERT com três argumentos de data e hora.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão da função DATEDIFF.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Atualizada a versão do pacote de extensão. Certifique-se de aplicar a versão mais recente do pacote de extensões em seus AWS SCT projetos existentes. Para ter mais informações, consulte Como usar pacotes de extensão .
BigQuery	Amazon Redshift	Resolvido um problema em que os objetos filtrados não eram convertidos no modo de interface de linha de comandos (CLI).
Greenplum	Amazon Redshift	Corrigido um erro em que AWS SCT não convertia tabelas temporárias declaradas em um procedimento armazenado.
Greenplum	Amazon Redshift	Corrigido um erro em que os atributos de codificação da coluna estavam ausentes no código convertido.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Implementada a conversão de instruções UPDATE para tabelas de autorreferência que têm mais de uma cláusula INNER JOIN.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Implementado o suporte a <code>inserted</code> e às tabelas <code>deleted</code> temporárias que o SQL Server usa para acionadores DML.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Foi aprimorada a conversão de tipos definidos pelo usuário em procedimentos armazenados que foram criados em diferentes esquemas de banco de dados. Resolveu um problema em que AWS SCT não foi possível encontrar o tipo de dados e exibiu um item de ação 9996.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que colchetes ([]) apareciam inesperadamente nos nomes dos objetos do banco de dados no código convertido.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que funções <code>@@ROWCOUNT</code> eram convertidas incorretamente.
Microsoft SQL Server DW	Amazon Redshift	Suporte implementado a tipos de dados <code>geometry</code> e <code>geography</code> .
Microsoft SQL Server DW	Amazon Redshift	Implementado o suporte à palavra-chave <code>MAX</code> nas instruções de tipo de dados no código convertido.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server DW	Amazon Redshift	Melhorada a conversão de funções do DATEADD.
Oracle	Aurora PostgreSQL PostgreSQL	Conversão aprimorada de código SQL em aplicativos Java adicionando suporte à MyBatis estrutura. Para ter mais informações, consulte Como converter o código SQL em aplicativos Java .
Oracle	Aurora PostgreSQL PostgreSQL	Conversão aprimorada de código SQL em aplicativos Java que usam a MyBatis estrutura. Foi adicionado um item de ação 30411 para código SQL com sintaxe não suportada.
Oracle	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de código SQL em aplicativos Pro*C adicionando suporte para instruções typedef struct.
Oracle	Aurora PostgreSQL PostgreSQL	Suporte implementado a instruções CROSS JOIN e LEFT JOIN.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL	Melhorada a conversão de instruções MERGE. Resolvido um problema em que os valores a serem inseridos estavam ausentes no código convertido.
	PostgreSQL	
Teradata	Amazon Redshift	Alterou as configurações padrão de codificação de compactação de colunas que AWS SCT usa no código convertido para corresponder às configurações padrão do Amazon Redshift. Para obter mais informações, consulte Codificações de compactação no Guia do desenvolvedor de banco de dados do Amazon Redshift.
Teradata	Amazon Redshift	Resolvido um problema em que as operações matemáticas que usam o tipo de dados TIME eram convertidas incorretamente.
Teradata	Amazon Redshift RSQL	Implementou a conversão do FastExport código que está dentro dos scripts de shell.
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um erro em AWS SCT que não convertia COALESCE nenhuma %data declaração.
Vertica	Amazon Redshift	Sugestões aprimoradas de otimização de conversão quando um usuário seleciona uma estratégia de otimização.

Notas de lançamento do AWS SCT Build 666

Origem	Destino	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvido um erro de análise que ocorria em cláusulas ON que estão dentro das instruções JOIN.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQL	
Azure Synapse Analytics	Amazon Redshift	Foram adicionadas três novas funções ao pacote de extensão para melhorar a conversão da função CONVERT com argumentos de data e hora.
Azure Synapse Analytics	Amazon Redshift	O carregador de metadados de origem foi aprimorado para garantir que ele AWS SCT carregue os esquemas do banco de dados do sistema.
Azure Synapse Analytics	Amazon Redshift	Corrigido um erro de resolução que ocorria em colunas de tabelas temporárias.
Azure Synapse Analytics	Amazon Redshift	Implementada a conversão dos tipos de dados BINARY e VARBINARY para o tipo de dados VARBYTE.
Azure Synapse Analytics	Amazon Redshift	Implementado o suporte ao tipo de dados TIME no código convertido.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de cláusulas COLLATE. Resolvido um problema em que o item de ação 31141 aparecia inesperadamente durante a conversão de colunas com o agrupamento de banco de dados padrão.
BigQuery	Amazon Redshift	Implementada a conversão de procedimentos que alteram os parâmetros de entrada.
Greenplum	Amazon Redshift	Resolveu um problema em que AWS SCT usava uma consulta que não é compatível com os bancos de dados Greenplum 6.x.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de seções EXCEPTION com a transferência de manipuladores de exceções do Db2 para z/OS para o PostgreSQL.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de instruções OPEN CURSOR.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Implementada a conversão de funções IIF usando expressões CASE.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que procedimentos com parâmetros com valores de tabela eram convertidos incorretamente quando a instrução CREATE PROCEDURE não incluía um bloco BEGIN . . . END .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Resolvido um problema em que a função SCOPE_IDENTITY era convertida incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Amazon RDS para Oracle	Corrigido um erro de carregador que ocorria com a função <code>SELECT_CATALOG_ROLE</code> ao usar o Oracle 10g como origem.
Oracle	Amazon RDS para Oracle	O carregador foi aprimorado para suportar tarefas do Oracle Scheduler.
Oracle	Aurora PostgreSQL PostgreSQL	Implementada a conversão de instruções JOIN com cláusulas USING.
Oracle	Aurora PostgreSQL PostgreSQL	Melhor desempenho do código convertido em que o código-fonte inclui variáveis globais nas cláusulas WHERE.
Oracle	Aurora PostgreSQL PostgreSQL	Conversão aprimorada de código SQL em aplicativos Java adicionando o suporte à MyBatis estrutura. Para ter mais informações, consulte Como converter o código SQL em aplicativos Java .
Oracle DW	Amazon Redshift	Conversão implementada dos operadores relacionais PIVOT e UNPIVOT.
Teradata	Amazon Redshift	Corrigido um erro em que o código-fonte que usa objetos JSON não era convertido.
Teradata	Amazon Redshift	Corrigido um erro em que as tabelas criadas por um usuário descartado não eram carregadas corretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Implementada a conversão de funções INSTR em funções nativas do Amazon Redshift STRPOS.
Teradata	Amazon Redshift	Implementada a conversão de funções NVP e TRANSLATE .
Teradata	Amazon Redshift	Melhorada a conversão de expressões COALESCE.
Teradata	Amazon Redshift	Melhorada a conversão de instruções DECLARE CONDITION .
Teradata	Amazon Redshift	Melhorada a conversão de funções EXTRACT com o elemento de sintaxe SECOND.
Teradata	Amazon Redshift	Melhorada a conversão das variáveis SQLSTATE e SQLCODE dentro das instruções LOOP.
Teradata	Amazon Redshift	Foi aprimorada a conversão de índices exclusivos.
Teradata	Amazon Redshift	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções CURRENT_TIMESTAMP com precisão fracionária definida como 3.
Teradata	Amazon Redshift	Resolveu um problema em que barras invertidas eram convertidas incorretamente em literais de string.
Teradata	Amazon Redshift	Resolvido um problema em que as instruções EXEC convertidas incluíam um nome de campo incorreto na instrução ADD CONSTRAINT .
Teradata	Amazon Redshift	Resolvido um problema em que as subconsultas QUALIFY convertidas incluíam um nome de subconsulta incorreto.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Resolvido um problema em que as visualizações convertidas não eram aplicadas. Foi adicionada uma conversão explícita a um tipo de dados específico para valores NULL no código convertido.
Teradata	Amazon Redshift	Resolveu um problema em que funções data e hora eram convertidas incorretamente.
Teradata	Amazon Redshift	Resolvido um problema em que literais de string hexadecimais não eram convertidos.

Notas de lançamento do AWS SCT Build 665

Origem	Destino	Novidades, aprimoramentos ou correções
Azure Synapse Analytics	Amazon Redshift	Conversão implementada de funções CONCAT com argumentos VARCHAR.
Azure Synapse Analytics	Amazon Redshift	Conversão aprimorada de CREATE TABLE instruções que criam tabelas temporárias e não incluem o nome do esquema. AWS SCT cria o dbo esquema para armazenar essas tabelas temporárias no banco de dados de destino.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções DROP TABLE que você executa em tabelas temporárias.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções OBJECT_ID com os blocos BEGIN...END .
Azure Synapse Analytics	Amazon Redshift	Resolveu um erro em que AWS SCT não era possível converter procedimentos armazenados com comentários em bloco.

Origem	Destino	Novidades, aprimoramentos ou correções
BigQuery	Amazon Redshift	Implementou a conversão de BigQuery data warehouses para o Amazon Redshift. Para ter mais informações, consulte Como usar o BigQuery como origem para AWS SCT .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de acionadores que manipulam vários eventos e funcionam com tabelas <code>inserted</code> e <code>deleted</code> do sistema no SQL Server.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Corrigido um erro de resolução que ocorria para tabelas <code>inserted</code> e <code>deleted</code> do sistema no SQL Server.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Implementado o suporte da nova versão 2.1.0 do arquivo de configuração de atributos do Babelfish. Esse arquivo define atributos SQL que são suportados e não suportados por versões específicas do Babelfish.
Oracle	Aurora MySQL MariaDB MySQL	Resolvido um problema em que o tipo de dados <code>varchar2</code> era convertido incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora MySQL	Para bancos de dados Oracle versão 12c e superior, AWS SCT suporta os seguintes tipos de dados estendidos:
	Aurora PostgreSQL	<ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW
	MariaDB	
	MySQL	AWS SCT aumentou o tamanho máximo da coluna suportada de 8.000 para 32.767 bytes para esses tipos de dados.
	PostgreSQL	
	L	
Oracle	Aurora PostgreSQL	Resolvido um erro de análise que ocorria para pacotes do Oracle Event Processing
	L	
	PostgreSQL	
	L	
Teradata	Amazon Redshift	Foi adicionado um item de ação 13214 para cláusulas RESET WHEN múltiplas em uma única instrução SELECT.
Teradata	Amazon Redshift	Foi adicionado um item de ação para variáveis SQLSTATE localizadas fora de um bloco de tratamento de exceções.
Teradata	Amazon Redshift	Implementada a conversão de variáveis ACTIVITY_COUNT para ROW_COUNT .
Teradata	Amazon Redshift	Conversão implementada da função ST_TRANSFORM de geometria integrada.
Teradata	Amazon Redshift	Melhorada a conversão de instruções de exclusão em visualizações sem a cláusula WHERE.
Teradata	Amazon Redshift	Melhorada a conversão de operadores CAST em expressões.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Melhorada a conversão de cláusulas GROUP BY.
Teradata	Amazon Redshift	Melhorada a conversão de funções INSTR e REGEXP_INSTR integradas.
Teradata	Amazon Redshift	Resolveu um problema em que as referências de alias da coluna lateral eram convertidas incorretamente.
Teradata	Amazon Redshift	Resolvido um problema em que os nomes das colunas eram convertidos incorretamente na subconsulta QUALIFY.
Teradata	Amazon Redshift	Implementada a conversão de comandos .QUIT com a palavra-chave de valor com status ERRORCODE .
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções CREATE.
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que o item de ação 9998 aparecia inesperadamente durante a conversão de instruções END.

Notas de lançamento do AWS SCT Build 664

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionado suporte ao Amazon Redshift sem servidor como origem e destino para projetos de migração de banco de dados em AWS SCT. Para se conectar ao Amazon Redshift sem servidor, certifique-se de usar o driver JDBC do Amazon Redshift versão 2.1.0.9 ou superior.
Todos	Todos	Melhorou a interface do usuário da janela de configurações de conversão. AWS SCT agora exibe configurações somente para pares de conversão de banco de dados com regras de mapeamento

Origem	Destino	Novidades, aprimoramentos ou correções	
		o criadas. Para ter mais informações, consulte Criação de regras de mapeamento .	
Todos	Todos	Atualizado o relatório de avaliação para remover informações duplicadas sobre a linha e a posição do item de ação.	
Todos	Amazon Redshift	Implementado o balanceamento automático de memória nas tarefas de extração de dados.	
Todos	Amazon Redshift	Resolvido um erro em que os agentes de extração de dados não conseguiam se conectar aos dispositivos AWS Snowball .	
Banco de dados do SQL Azure	Aurora MySQL	Implementado o suporte do SUSE Linux 15.3 como uma plataforma para executar agentes de extração de dados.	
IBM Db2 para z/OS	Aurora PostgreSQL		
IBM Db2 LUW	MySQL		
Microsoft SQL Server	PostgreSQL		
MySQL	L		
Oracle	MySQL		
PostgreSQL	PostgreSQL		
SAP ASE	L		
Azure Synapse Analytics	Amazon Redshift		Melhorada a conversão de funções do DATEADD.

Origem	Destino	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Foi adicionada a capacidade de alterar o agrupamento de colunas em regras de migração.
Microsoft SSIS	AWS Glue AWS Glue Studio	Resolvido um erro inesperado que ocorria quando os usuários selecionavam um script de origem.
Oracle	Aurora MySQL MariaDB MySQL	Conversão implementada do uso de funções armazenadas como expressões de coluna geradas. AWS SCT cria acionadores para emular esse comportamento porque o MySQL não suporta o uso de funções armazenadas como expressões de coluna geradas.
Oracle	Aurora PostgreSQL PostgreSQL	Implementou a conversão de funções do UTL_MATCH pacote como parte do pacote de AWS SCT extensão.
Oracle	Aurora PostgreSQL PostgreSQL	Implementada conversão da função REGEXP_LIKE com o parâmetro NULL.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão da função SYS_EXTRACT_UTC .
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de código SQL em aplicativos C++ implementando o suporte de funções Wcscats, Wcscpys e Wcsncats. Para obter mais informações, consulte Como converter o código SQL em aplicativos C++ com a AWS SCT .
Oracle DW Snowflake	Amazon Redshift	Resolvido um problema em que as instruções convertidas não incluíam a conversão explícita de valores para o tipo de dados da coluna. Esse problema ocorreu em instruções que usam resultados de consultas de outras tabelas.
Teradata	Amazon Redshift	Foi adicionada a capacidade de alterar o agrupamento de colunas entre <code>case sensitive</code> e <code>case insensitive</code> nas regras de migração. Para obter mais informações, consulte Como criar regras de migração .
Teradata	Amazon Redshift	Corrigido um erro de resolução que ocorria nas instruções CREATE TABLE AS.
Teradata	Amazon Redshift	Corrigido um erro em que a função integrada P_INTERSECT com uma expressão COALESCE não era convertida.
Teradata	Amazon Redshift	Implementada a conversão de colunas nomeadas de 0ID para _0ID evitando o uso de uma palavra-chave reservada no Amazon Redshift.
Teradata	Amazon Redshift	Implementada a conversão de instruções RENAME para funções, procedimentos, visualizações e macros.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Implementada a conversão da função STROKE em função SPLIT_PART no Amazon Redshift.
Teradata	Amazon Redshift	Melhorada a conversão de funções INSTR e REGEXP_INSTR do sistema.
Teradata	Amazon Redshift	Melhorada a conversão de tipos de dados TIME.
Teradata	Amazon Redshift	Emulação aprimorada das tabelas SET e MULTISSET implementando a conversão de índices exclusivos primários e secundários.
Teradata	Amazon Redshift	Resolvido um erro de análise que ocorria com a função CHARACTER .
Teradata BTEQ	Amazon Redshift RSQL	Resolveu um erro que ocorria quando os usuários removiam os scripts Teradata Basic Teradata Query (BTEQ) do projeto. AWS SCT

Notas de lançamento do AWS SCT Build 663

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionada a capacidade de alterar o comprimento dos tipos de dados char, varchar, nvarchar e string usando o operador de multiplicação em uma regra de migração. Para obter mais informações, consulte Como criar regras de migração .
Todos	Todos	Implementado o suporte de três novas colunas no relatório de avaliação de vários servidores e atualizou o formato do arquivo de entrada. Certifique-se de usar o modelo atualizado do arquivo de entrada com a versão mais recente do AWS SCT. Para ter mais informações, consulte Como criar um relatório de avaliação multisservidor para migração de banco de dados .

Origem	Destino	Novidades, aprimoramentos ou correções
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções OBJECT_ID .
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Foi adicionado suporte ao Babelfish para Aurora PostgreSQL 1.2.0 como plataforma de destino para relatórios de avaliação de migração de banco de dados. Para obter mais informações, consulte Funcionalidade suportada no Babelfish por versão no Guia do usuário do Amazon Aurora.
Microsoft SQL Server DW	Amazon Redshift	Suporte adicionado para cláusulas AT TIME ZONE.
Microsoft SQL Server DW	Amazon Redshift	Resolvido um problema em que uma instrução fora do bloco BEGIN/END era convertida incorretamente.
Netezza	Amazon Redshift	Melhorada a conversão do tipo de dados TIME e conversão implementada de funções, expressões e literais incorporadas relacionadas.
Oracle	Aurora PostgreSQL PostgreSQL	Corrigido um erro de carregador que ocorria ao usar o Oracle 10g como origem.
Oracle	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de cláusulas OFFSET e FETCH.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que procedimentos com parâmetros OUT com valores padrão eram convertidos incorretamente.
Oracle DW	Amazon Redshift	Foi aprimorada a conversão de funções Oracle para funções definidas pelo usuário do Amazon Redshift.
Snowflake	Amazon Redshift	Melhorada a conversão de cláusulas WITH.
Teradata	Amazon Redshift	Foi adicionado um novo item de ação 13209 para caracteres multibyte não suportados para o tipo de dados CHAR.
Teradata	Amazon Redshift	Corrigido um erro no carregador em que as tabelas não estavam totalmente carregadas.
Teradata	Amazon Redshift	Corrigido um erro do transformador em que a função incorporada P_INTERSECT em uma condição JOIN não era convertida.
Teradata	Amazon Redshift	Corrigido um problema em que o nome de uma visualização era convertido em letras maiúsculas e minúsculas quando a instrução SELECT era executada em uma tabela com caracteres especiais em seu nome.
Teradata	Amazon Redshift	Melhorada a conversão de instruções INSERT com o valor UNTIL_CHANGED no tipo de dados PERIOD(DATE) .
Teradata	Amazon Redshift	Melhorada a conversão da função integrada FORMAT usando a função TO_CHAR no Amazon Redshift.
Teradata	Amazon Redshift	Melhorada a conversão da função integrada RANK para garantir que o código convertido retorne valores NULL na mesma ordem do código-fonte.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Foi aprimorada a conversão de restrições exclusivas, como índices exclusivos primários ou secundários.

Notas de lançamento do AWS SCT Build 662

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionada a capacidade de criar AWS SCT projetos automaticamente para cada banco de dados de origem ao criar o relatório de avaliação de vários servidores. Com essa opção ativada, é AWS SCT possível adicionar regras de mapeamento a esses projetos e salvar estatísticas de conversão para uso off-line. Para ter mais informações, consulte Como criar um relatório de avaliação multisservidor para migração de banco de dados .
Todos	Todos	Implementado o suporte ao percentual (%) como curinga nos nomes do banco de dados e do esquema ao criar o relatório de avaliação de vários servidores.
Todos	Aurora MySQL Aurora PostgreSQL	Atualizou o tempo de execução de todas as AWS Lambda funções para a versão 3.9 do Python.
Todos	Amazon Redshift	Atualizou todos os agentes de extração de dados para uso AWS SDK for Java 2.x.
Banco de dados do SQL Azure	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de instruções DELETE com cláusulas NON EXISTS.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server		
Azure Synapse Analytics	Amazon Redshift	Resolvido um erro em que a conexão com um banco de dados de origem falhava.
IBM Db2 para z/OS	Aurora PostgreSQL PostgreSQL	Resolvido um erro em que o código convertido de um acionador incluía duas menções ao alias do objeto.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de objetos com nomes em letras maiúsculas e minúsculas quando a opção Tratar nome do objeto do banco de dados com distinção entre maiúsculas e minúsculas está ativada.
Microsoft SQL Server DW Teradata	Amazon Redshift	Conversão implementada dos operadores relacionais PIVOT e UNPIVOT.
Netezza	Amazon Redshift	Implementada a conversão dos tipos de dados TIME.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Implementada a conversão constante do pacote <code>UTL_TCP.CRLF</code> .
Oracle	Aurora PostgreSQL PostgreSQL	Corrigido um problema de pacote de extensão em que o tamanho dos tipos de dados para colunas de tamanho variável não era mantido durante a conversão.
Oracle	Aurora PostgreSQL PostgreSQL	Implementou a conversão de código SQL em aplicativos C++. Para obter mais informações, consulte Como converter o código SQL em aplicativos C++ com a AWS SCT .
Oracle	Aurora PostgreSQL PostgreSQL	Suporte implementado de nomenclatura com distinção entre maiúsculas e minúsculas para a conversão de variáveis globais e matrizes associativas.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão das funções TO_CHAR, TO_DATE e TO_NUMBER no pacote de extensão.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão do operador TABLE().
Oracle DW	Amazon Redshift	Foi adicionado suporte para conversão de chaves primárias e outras restrições.
Oracle DW	Amazon Redshift	Corrigido um problema em que o item de ação 12054 não aparecia durante a conversão de instruções condicionais.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Resolvido um erro quando um objeto com um nome vazio foi criado na árvore de destino durante a conversão de tabelas com colunas do tipo definido pelo usuário.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Corrigido um erro do carregador para objetos armazenados, como scripts, rotinas e assim por diante.
Snowflake	Amazon Redshift	Corrigido um problema em que o item de ação 22152 não aparecia quando necessário e AWS SCT exibia o resultado da conversão como um comentário.

Origem	Destino	Novidades, aprimoramentos ou correções
Snowflake	Amazon Redshift	Foi aprimorada a conversão das funções de data e hora; suporte implementado de fusos horários.
Snowflake	Amazon Redshift	Resolvido um problema em que expressões de tabela comuns (CTEs) não recursivas com uma cláusula WITH eram convertidas em CTEs recursivas.
Teradata	Amazon Redshift	Melhorada a conversão de instruções UPDATE com links de tabela em condições.
Teradata	Amazon Redshift	Melhorada a conversão de instruções RENAME TABLE.
Teradata	Amazon Redshift	Foi resolvido um problema em que colunas vazias apareciam no arquivo de valores separados por vírgula (CSV) com um relatório de avaliação.
Teradata	Amazon Redshift RSQL	Corrigido um erro em que faltava um ponto e vírgula no final da macro convertida do BTEQ (Basic Teradata Query).
Teradata	Amazon Redshift RSQL	Melhorada a conversão de vários valores de tipos de dados em instruções CASE.
Teradata	Amazon Redshift RSQL	Melhorada a conversão da cláusula LIKE ANY com um caractere ESCAPE.
Teradata	Amazon Redshift RSQL	Melhorada a conversão da função CAST em instruções INSERT.
Teradata	Amazon Redshift RSQL	Foi aprimorada a conversão dos fusos horários, implementou o mapeamento da região do fuso horário.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift RSQL	Resolvido um problema em que o item de ação 9998 aparecia inesperadamente durante a conversão de scripts de shell com scripts BTEQ.
Teradata	Amazon Redshift RSQL AWS Glue	Implementado o limite de 500 caracteres para os valores das variáveis de substituição.
Vertica	Amazon Redshift	Implementada a conversão dos tipos de dados BINARY, VARBINARY, LONG BINARY, BYTEA e RAW para o tipo de dados VARBYTE.
Vertica	Amazon Redshift	Melhorada a conversão de funções e literais integradas.

Notas de lançamento do AWS SCT Build 661

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Filtros adicionados para pesquisar regras de mapeamento na exibição de mapeamento. Quando você aplica um filtro, AWS SCT exibe regras que correspondem às condições de filtragem na lista Mapeamentos do servidor. Para ter mais informações, consulte Gerenciar regras de mapeamento .
Todos	Todos	Apache Log4j atualizado para a versão 2.17.1.
Todos	Amazon Redshift	Foi adicionado suporte à migração de dados para o Amazon Redshift usando a cláusula ENCRYPTED no comando COPY.
Todos	Amazon Redshift	Aprimorou a API REST dos agentes de extração de dados. A API REST atualizada adiciona suporte a novas propriedades, como chave de criptografia, tipo de criptografia e assim por diante.

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Amazon Redshift	Implementada a assunção de papéis nos agentes de extração de dados. Essa atualização melhora a distribuição de subtarefas e permite que AWS SCT atribua tarefas a agentes livres da função especificada.
Todos	Amazon Redshift	Implementou uma verificação de que todos os componentes necessários estão instalados antes que o pacote de extensão seja aplicado ao Amazon Redshift.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Melhorada a conversão das funções ERROR_LINE , ERROR_MESSAGE , ERROR_NUMBER , ERROR_PROCEDURE , ERROR_SEVERITY e ERROR_STATE do sistema para tratamento de erros.
IBM Db2 para z/OS	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Foi adicionado suporte ao IBM Db2 para z/OS versão 12 como origem para projetos de migração de banco de dados em AWS SCT. Para ter mais informações, consulte Como usar o IBM Db2 para z/OS como origem .
IBM Db2 LUW	Todos	Aprimorou o carregador de metadados de origem para garantir o AWS SCT carregamento de parâmetros de rotina que duplicam os nomes das colunas.

Origem	Destino	Novidades, aprimoramentos ou correções
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL	Corrigido um erro do transformador para procedimentos com a instrução do conjunto SET NOCOUNT ON.
Microsoft SQL Server	PostgreSQL	
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL	Melhorada a conversão da função CONCAT quando um valor de entrada é uma variável do tipo definido pelo usuário.
Microsoft SQL Server	PostgreSQL	
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL	Resolvido um problema em que a função DATEPART era convertida incorretamente.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Implementado o suporte da nova versão do arquivo de configuração de atributos do Babelfish. Esse arquivo define atributos SQL que são suportados e não suportados por versões específicas do Babelfish.
Microsoft SQL Server DW	Amazon Redshift	Resolvido um problema em que instruções EXECUTE eram convertidas incorretamente.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SSIS	AWS Glue	Melhorou a interface do usuário do assistente de configuração de tarefas. AWS SCT agora exibe somente as conexões disponíveis na seção de configuração da conexão.
Microsoft SSIS	AWS Glue	Resolvido um problema em que as regras de transformação não eram aplicadas às tarefas do pacote e às regras variáveis.
Microsoft SSIS	AWS Glue AWS Glue Studio	Foi adicionado um novo item de ação 25042 para componentes não suportados.
Microsoft SSIS	AWS Glue Studio	Implementou a conversão de pacotes de extração, transformação e carregamento (ETL) do Microsoft SQL Server Integration Services (SSIS) em AWS Glue Studio Para ter mais informações, consulte Como converter SSIS em AWS Glue Studio .
Oracle	MariaDB	Corrigido um problema com a conversão do operador MINUS.
Oracle	MariaDB	Melhorada a conversão das funções ROWNUM, SYS_GUID, TO_CHAR e ADD_MONTHS quando a variável do sistema sql_mode no MariaDB é para Oracle.
Oracle	PostgreSQL	Foi adicionada uma opção para evitar a conversão de tipos de variáveis de vinculação em tipos SQL em projetos genéricos de conversão de aplicativos.
Oracle	PostgreSQL	Foi adicionada uma opção para evitar a adição do nome do esquema ao nome do objeto convertido em projetos genéricos de conversão de aplicativos.
Oracle	PostgreSQL	Foi adicionado suporte ao formato de variável de ligação ?x para conversão de código SQL do aplicativo.
Oracle DW	Amazon Redshift	Implementada a conversão do tipo de dados RAW para o tipo de dados VARBYTE.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Foi adicionada uma opção para emular tabelas SET no código convertido. Para essa emulação, AWS SCT suporta MIN e MAX condições.
Teradata	Amazon Redshift	Foi aprimorada a conversão de operações de junção que têm parâmetros de diferentes tipos de dados. Essa atualização permite AWS SCT aplicar regras de transformação durante a conversão de tais operações.
Teradata	Amazon Redshift	Resolvido um problema em que a cláusula GROUP BY era convertida incorretamente.
Teradata	Amazon Redshift	Resolvido um problema em que a cláusula QUALIFY era convertida incorretamente.
Teradata	Amazon Redshift	Resolveu um erro inesperado ocorrido durante a importação de FastExport scripts.
Teradata	Amazon Redshift RSQL	Implementada a capacidade de editar os valores das variáveis nos scripts Teradata BTEQ e shell.
Teradata	Amazon Redshift RSQL	Resolveu um problema em que o script de manifesto estava ausente nas sessões convertidas do Teradata FastLoad .
Teradata	Amazon Redshift RSQL	Resolveu um problema em que a extensão do arquivo de manifesto estava ausente no localizador uniforme de recursos (URL) dos FastLoad scripts convertidos.
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um erro do carregador para scripts com variáveis de substituição.
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um problema em que o item de ação 27022 não aparecia quando necessário.

Notas de lançamento do AWS SCT Build 660

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionado suporte AWS Secrets Manager e Secure Sockets Layer (SSL) no relatório de avaliação de vários servidores. Para ter mais informações, consulte Como criar um relatório de avaliação multisservidor para migração de banco de dados .
Todos	Todos	Coleta de estatísticas aprimorada para objetos convertidos.
Todos	PostgreSQL L	Implementado o suporte da versão principal 14 do PostgreSQL e do MariaDB 10.6 como destinos de migração.
Azure Synapse Analytics	Amazon Redshift	Lógica de transformação aprimorada para os nomes dos objetos convertidos.
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL L	Melhorada a conversão de tipos de dados XML.
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que cláusulas NOT LIKE eram convertidas incorretamente.
Banco de dados do Microsoft Azure SQL Server	Aurora PostgreSQL L	Corrigido um erro de transformador para procedimentos com instruções INSERT, DELETE e UPDATE que incluem a cláusula OUTPUT.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft Azure SQL	PostgreSQL	
Microsoft SQL Server	L	
Banco de dados do Microsoft Azure SQL	Aurora PostgreSQL	Corrigido um erro do transformador para procedimentos com a instrução RETURN @@ROWCOUNT .
Microsoft SQL Server	L PostgreSQL	
Microsoft SQL Server	Todos	Foi aprimorada a conversão de procedimentos que usam servidores vinculados.
Microsoft SQL Server	Todos	Foi adicionado suporte à Autenticação do Microsoft Windows no relatório de avaliação de vários servidores.
Microsoft SQL Server	Aurora PostgreSQL	Corrigido um erro de transformador para construtores de valores de tabela.
	L PostgreSQL	
Microsoft SQL Server DW	Amazon Redshift e AWS Glue	Foi aprimorada a conversão de scripts de extração, transformação e carregamento (ETL) para incluir o caminho correto para os scripts convertidos.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server DW	Amazon Redshift	Resolvido um problema em que diferentes scripts convertidos eram gerados para plataformas de banco de dados de destino virtuais e reais.
Oracle	PostgreSQL Aurora PostgreSQL	Foi adicionado suporte para conversão de índices para visões materializadas.
Oracle	PostgreSQL Aurora PostgreSQL	Corrigido um problema em que o item de ação 5982 não aparecia durante a conversão de PRIMARY KEY e as restrições UNIQUE com a opção NOVALIDATE .
Oracle DW	Amazon Redshift	Resolvido um problema em que categorias adicionais eram exibidas no esquema convertido.
Teradata	Amazon Redshift	Corrigido um problema em que o item de ação 13185 não aparecia ao converter uma coluna não resolvida como argumento da função CAST.
Teradata	Amazon Redshift	Melhorada a conversão de instruções DELETE e DELETE ALL para usar o comando TRUNCATE no código convertido.
Teradata	Amazon Redshift	Melhorada a conversão de tabelas SET.
Teradata	Amazon Redshift	Melhorada a conversão da condição NORMALIZE .
Teradata	Amazon Redshift	O relatório de avaliação foi atualizado para remover as estatísticas de conversão do esquema do banco de dados da lista de objetos de armazenamento do banco de dados.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Melhorada a conversão da instrução UPDATE sem a cláusula FROM.
Teradata	Amazon Redshift	Implementado o suporte ao tipo de dados VARBYTE no código convertido.
Teradata BTEQ	AWS Glue	Resolvido um problema em que a opção Converter para AWS Glue estava desativada no menu de contexto.
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que os tipos de dados estavam ausentes no código convertido.
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que as variáveis de substituição eram citadas incorretamente no código convertido.
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um problema com a conversão de variáveis de substituição com valores em FastLoad scripts.
Vertica	Amazon Redshift	Implementado o suporte ao tipo de dados TIME no código convertido.
Vertica	Amazon Redshift	Melhorada a conversão de expressões SELECT DISTINCT e ORDER BY.
Vertica	Amazon Redshift	Foi adicionado suporte para conversão de restrições.
Vertica	Amazon Redshift	Resolvido um erro em que um relatório de avaliação não era salvo como um arquivo de valores separados por vírgula (CSV).

Notas de lançamento do AWS SCT Build 659

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi aprimorado o Assistente para novos projetos que gera um relatório de avaliação combinado para vários bancos de dados de origem.
Todos	Todos	Corrigido um problema em que o pacote de extensão não era criado em projetos que incluem vários bancos de dados de origem e destino.
Todos	Todos	Foi aprimorada a conversão do código SQL incorporado ao código-fonte do aplicativo.
Todos	Todos	Foi adicionada a capacidade de executar scripts de diferentes pastas na interface de AWS SCT linha de comando.
Todos	Amazon Redshift	Melhorada a mensagem de aviso fornecida quando os usuários escolhem Executar otimização em projetos de migração com a plataforma de banco de dados de destino virtual Amazon Redshift.
Todos	Aurora PostgreSQL	Implementado o suporte da versão principal 13 do PostgreSQL na edição compatível com o Aurora PostgreSQL como destino de migração.
Todos	Amazon RDS para MySQL	Implementada a conversão de código sem distinção entre maiúsculas e minúsculas por padrão.
Azure Synapse Analytics	Amazon Redshift	Resolvido um erro em que a conexão com um banco de dados de origem falhava na interface da linha de comando.
Microsoft SQL Server	PostgreSQL	Melhorada a conversão de procedimentos que incluem instruções UPDATE com condições de junção.

Origem	Destino	Novidades, aprimoramentos ou correções
	Aurora PostgreSQL	
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Foi aprimorada a conversão de acionadores, procedimentos armazenados e funções que incluem o valor após o sinal de igual.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Corrigido um erro do transformador para procedimentos com a instrução DELETE e o operador OR.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Melhorada a conversão da cláusula OUTPUT.
Microsoft SQL Server DW	Amazon Redshift e AWS Glue	Melhorada a conversão de tipos de dados NUMERIC.
Microsoft SQL Server DW	Amazon Redshift	Foi aprimorada a conversão de visualizações que têm um alias de tabela com o mesmo nome da tabela original.
Microsoft SSIS	AWS Glue	Corrigido um problema em que as credenciais de AWS Glue conexão não eram exibidas na janela Configurar conexões.

Origem	Destino	Novidades, aprimoramentos ou correções
Netezza	Amazon Redshift	Foi adicionada a capacidade de repetir a execução das tarefas de migração de dados de captura de dados de alteração (CDC) todos os dias.
Netezza	Amazon Redshift	Corrigido um problema em que a guia Tarefas ficava inativa após cancelar o registro de um agente de extração de dados.
Netezza	Amazon Redshift	Corrigido um problema em que a confirmação do registro do agente de migração de dados não era exibida na interface do usuário.
Netezza	Amazon Redshift	Corrigido um problema em que uma conexão com um banco de dados de origem falhava com um Erro do carregador.
Netezza	Amazon Redshift	Resolvido um erro em que os agentes de migração de dados não eram executados após a abertura de um projeto salvo.
Oracle	Amazon RDS para Oracle	Suporte implementado do Oracle Unified Auditing.
Oracle	PostgreSQL Aurora PostgreSQL	Implementada a conversão de código SQL em aplicativos C#. Para obter mais informações, consulte Como converter o código SQL em aplicativos C# .
Oracle	PostgreSQL Aurora PostgreSQL	Implementou uma nova lógica de transformação para nomes de objetos com distinção entre maiúsculas e minúsculas para melhorar a visibilidade das alterações na conversão de código. AWS SCT converte nomes de objetos em maiúsculas em minúsculas. O oposto também é verdadeiro; AWS SCT converte nomes de objetos de minúsculas para maiúsculas. Outros nomes de objetos e palavras reservadas são convertidos sem alterações.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL Aurora PostgreSQL	Melhorada a conversão de partições de hash sem a restrição NOT NULL.
Oracle	Aurora PostgreSQL	Foi adicionado suporte para conversão de restrições do Oracle CHECK, FOREIGN KEY e NOT NULL com a cláusula ENABLE NOVALIDATE .
Oracle DW	Amazon Redshift	Corrigido um problema em que os valores incorretos para números de ponto flutuante eram migrados.
Oracle DW	Amazon Redshift e AWS Glue	Foi resolvido um problema com colunas vazias no relatório de avaliação de migração de banco de dados em um arquivo de valores separados por vírgula (CSV).
SAP ASE	PostgreSQL Aurora PostgreSQL	Corrigido um problema com uma interrupção inesperada da conversão.
Snowflake	Amazon Redshift	Melhorada a conversão de tipos de dados VARIANT.
Teradata	Amazon Redshift	Melhorada a conversão da instrução COLLECT STATISTICS .
Teradata	Amazon Redshift	Corrigido um problema em que o item de ação 9998 não aparecia durante a conversão de exibições aninhadas com colunas PERIOD.
Teradata	Amazon Redshift e AWS Glue	Corrigido um problema em que uma plataforma de AWS Glue destino virtual não era exibida na interface após a abertura de um projeto salvo.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata BTEQ	AWS Glue	Corrigido um problema em que a conversão para uma plataforma de AWS Glue destino virtual não era suportada após a abertura de um projeto salvo.
Teradata BTEQ	Amazon Redshift RSQL	Destaque de sintaxe do código convertido aprimorado.
Teradata BTEQ	Amazon Redshift RSQL	Implementada a verificação dos valores dos parâmetros após o upload. Os valores sem suporte são destacados na guia Variáveis.
Vertica	Amazon Redshift	Implementada a conversão de funções agregadas.
Vertica	Amazon Redshift	Implementada a conversão de projeções em visões materializadas e melhoria da interface do usuário que exibe o código-fonte das projeções.

Notas de lançamento do AWS SCT Build 658

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Integração fornecida com AWS Secrets Manager. Agora você pode usar as credenciais de conexão do banco de dados armazenadas no Secrets Manager.
Todos	Todos	Foi adicionado suporte para scripts no formato YAML na interface de AWS SCT linha de comando.
Todos	Amazon Redshift	Implementado o suporte de endpoints de interface Amazon S3 (VPCE) em agentes de extração de dados.

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Amazon Redshift	Foi adicionado suporte para a plataforma de banco de dados de destino virtual Amazon Redshift, além do Amazon AWS Glue Redshift e da combinação já suportados.
Greenplum	Amazon Redshift	Corrigido um problema em que a opção Salvar como SQL não salvava o código SQL convertido em um arquivo.
IBM Db2 LUW Banco de dados do Microsoft Azure SQL Microsoft SQL Server Oracle SAP ASE	Aurora MySQL	Foi aprimorada a conversão para oferecer suporte a novos recursos da edição compatível com Amazon Aurora MySQL com compatibilidade com o MySQL 8.0.
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Corrigido um problema em que o item de ação 810 não aparecia quando necessário.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de procedimentos com instruções UPDATE DELETE e INSERT.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Corrigido um problema em que o item de ação 7810 não aparecia quando necessário.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Melhorada a conversão de uma instrução EXEC que está aninhada dentro de uma instrução IF . . . ELSE .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Foi aprimorada a conversão de visualizações indexadas.
Netezza	Amazon Redshift	Agentes de migração de dados aprimorados por meio do rastreamento de transações em tempo real durante a carga total da operação de captura de dados de alteração (CDC). Agora você pode interromper as tarefas de migração de dados se a sessão do CDC estiver programada para começar em um determinado horário. Além disso, você pode ver o nível de log de erros no console depois de interromper uma tarefa com o CDC.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Todos	O carregador de tabelas foi aprimorado para garantir que os objetos sejam AWS SCT carregados com opções de compartilhamento.
Oracle	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão da função SYSDATE e adicionou a capacidade de alterar o fuso horário nas Configurações de conversão.
Oracle	Aurora PostgreSQL L PostgreSQL L	Resolvido um problema em que instruções dinâmicas não eram convertidas.
Oracle	Aurora PostgreSQL L PostgreSQL L	Corrigido um problema em que o código convertido não incluía nomes gerados pelo sistema.
Oracle Oracle DW	Aurora PostgreSQL L PostgreSQL L	Melhorada a conversão de instruções SELECT que estão aninhadas em acionadores.
Oracle DW	Amazon Redshift	Melhorada a conversão das funções TO_DATE, TO_TIMESTAMP e TO_TIMESTAMP_TZ no pacote de extensão.
Snowflake	Amazon Redshift	Foi adicionada uma opção para salvar o código SQL convertido em arquivos diferentes para cada objeto ou para cada instrução.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Melhorada a conversão da função CONCAT.
Teradata	Amazon Redshift	Melhorada a conversão de uma instrução SELECT que está aninhada em uma cláusula WHERE.
Teradata	Amazon Redshift	Resolvido um problema em que tabelas SET e MULTISSET eram convertidas incorretamente após os usuários eliminarem e recriarem uma tabela.
Teradata	Amazon Redshift	Melhorada a conversão dos procedimentos que incluem uma cláusula WITH.
Teradata	Amazon Redshift	Melhorada a conversão de tipos de dados DATE.
Teradata	Amazon Redshift RSQL	Resolveu um problema em que ocorria um erro inesperado do transformador durante a conversão de FastExport scripts.
Teradata BTEQ	Amazon Redshift RSQL	Foi adicionado suporte para conversão de um índice de junção em uma visão materializada.
Teradata BTEQ	Amazon Redshift RSQL	Foi adicionado suporte para conversão de uma definição TITLE que inclui várias linhas.
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que o tamanho de um tipo de dados geoespaciais não era convertido.
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um problema em que os nomes dos parâmetros eram convertidos em caracteres em letra minúscula.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata BTEQ	Amazon Redshift RSQL	Corrigido um problema em que um procedimento armazenado aninhado em uma instrução MACRO não era convertido.
Vertica	Amazon Redshift	Melhorada a conversão do operador ALL.
Vertica	Amazon Redshift	Resolvido um problema em que a opção Use Union all view? nas Configurações de conversão não era aplicada.
Vertica	Amazon Redshift	Melhorada a conversão dos tipos de dados TIME e TIME WITH TIMEZONE.
Vertica	Amazon Redshift	Resolvido um problema com o carregamento de tabelas flexíveis.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 657

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Atualizou o Apache Log4j para a versão 2.17 para resolver problemas de vulnerabilidade de segurança.
Todos	Amazon Redshift	Projetos aprimorados de otimização de esquema, nos quais as principais estatísticas de gerenciamento não foram salvas no projeto AWS SCT .
Amazon Redshift	Amazon Redshift	Corrigido um problema com a atualização das informações do servidor.

Origem	Destino	Novidades, aprimoramentos ou correções
Apache Cassandra	Amazon DynamoDB	Corrigido um problema com as regras de mapeamento ao usar a interface de AWS SCT linha de comando.
Apache Cassandra	Amazon DynamoDB	Resolvido um problema em que a tarefa de migração não foi criada devido a um título atualizado no certificado.
Microsoft SQL Server	Aurora PostgreSQL	Foi corrigido um problema para que o item de ação 7672 não aparecesse durante a conversão dos procedimentos do Microsoft SQL Server com SQL dinâmico.
	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Foi aprimorada a conversão de funções com valor de tabela.
Microsoft SQL Server	PostgreSQL	
Banco de dados do SQL Azure	Aurora PostgreSQL	Resolvido um problema em que o argumento OUT em um procedimento armazenado com o valor de retorno padrão não era convertido no argumento INOUT.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Estratégias de otimização aprimoradas ao encontrar as tabelas e colunas mais usadas na tabela QueryLog.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Problemas corrigidos com a conversão do seguinte: <ul style="list-style-type: none"> • Operador de atribuição de concatenação de strings (+=) • Função do SCOPE_IDENTITY • Tipo de dados do varchar(max)
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Foi aprimorada a conversão de visualizações com funções não suportadas.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Corrigido um problema em que funções não suportadas como argumento para outra função eram convertidas incorretamente.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL L	Foi aprimorada a conversão das referências da tabela de transição.
Microsoft SQL Server DW	Amazon Redshift	A categoria de funções agregadas foi adicionada à árvore de metadados do banco de dados de origem.
Microsoft SQL Server DW	Amazon Redshift	Melhorada a conversão de tipos de dados TIME.

Origem	Destino	Novidades, aprimoramentos ou correções
Azure Synapse Analytics Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	Corrigido um problema em que os scripts DROP e CREATE não eram salvos ao usar uma plataforma de banco de dados de destino virtual.
Microsoft SQL Server Integration Services	AWS Glue	Resolvido um problema em que os scripts dos objetos de origem não eram exibidos na interface do usuário.
Netezza	Amazon Redshift	Estratégias de otimização aprimoradas ao escolher a tabela de fatos e as dimensões apropriadas para colocação.
Oracle	Aurora PostgreSQL PostgreSQL	Resolvido um problema para converter corretamente os acionadores Oracle, que usam números de sequência.
Oracle	Aurora PostgreSQL PostgreSQL	Foi aprimorada a conversão de visualizações com links de bancos de dados públicos.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle DW	Amazon Redshift	Estratégias de otimização aprimoradas por meio da análise da cardinalidade das colunas de índice.
Oracle DW	Amazon Redshift	Corrigido um problema em que funções escalares personalizadas definidas pelo usuário com concatenação de strings eram convertidas incorretamente.
Snowflake	Amazon Redshift	Corrigido um problema em que a opção Salvar como SQL não era exibida na interface do usuário.
Teradata	Amazon Redshift	Corrigido um problema em que a coleta de estatísticas falhava com a exceção <code>LOADER ERROR</code> .
Teradata	Amazon Redshift	Corrigido um problema em que a opção Criar relatório não era exibida na UI.
Teradata	Amazon Redshift	Melhorada a conversão da função <code>CAST</code> .
Teradata	Amazon Redshift	Corrigida uma conversão quebrada para <code>ST_Line_Interpolate_Point</code> .
Teradata	Amazon Redshift	Foi removido um valor inesperado do caminho da biblioteca Python.
Teradata	Amazon Redshift RSQL	Corrigido um erro de resolução que aparecia durante a conversão de vários FastLoad scripts.
Teradata BTEQ	Amazon Redshift RSQL	Melhorada a conversão dos tipos de dados de comando e geometria <code>DATABASE</code> .
Teradata BTEQ	AWS Glue	Corrigido um problema com a sincronização incorreta dos scripts de origem e destino na interface do usuário.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 656

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionado suporte a vários bancos de dados de origem e destino em um projeto. Agora, os usuários podem criar regras de mapeamento para combinar diferentes esquemas de banco de dados e plataformas de destino no mesmo projeto.
Todos	Todos	Adicionado suporte para plataformas de banco de dados de destino virtual. Agora, os usuários não precisam se conectar a um banco de dados de destino para ver como AWS SCT converte o esquema do banco de dados de origem.
Todos	Todos	Melhorias da UI: <ul style="list-style-type: none"> • As opções Conectar ao servidor e Desconectar do servidor foram adicionadas às árvores de metadados de origem e de destino. • Foi adicionada uma opção para remover um servidor de banco de dados do AWS SCT projeto.
Cassandra	Amazon DynamoDB	Resolvido um problema de pesquisa em que a variável CASSANDRA_HOME não incluía uma barra (/) depois de <code>cassandra.yaml</code> ou da pasta <code>conf</code> .
Cassandra	Amazon DynamoDB	Adicionado o suporte da imagem de máquina da Amazon (AMI) para o Amazon Linux 2.
Cassandra	Amazon DynamoDB	Melhora da mensagem de erro fornecida quando uma chave incorreta é fornecida para Cassandra.

Origem	Destino	Novidades, aprimoramentos ou correções
Cassandra	Amazon DynamoDB	Melhorada a conversão alterando uma propriedade no arquivo <code>cassandra-env.yaml</code> dependendo da versão do banco de dados de destino.
Cassandra	Amazon DynamoDB	Aumentada a versão Java no Cassandra Datacenter de destino para 1.8.0.
Greenplum	Amazon Redshift	Estratégias de otimização aprimoradas nas configurações do projeto.
Greenplum	Amazon Redshift	Resolvido um problema de migração de dados em que os objetos não eram aplicados ao banco de dados com este erro: <code>An I/O error occurred while sending to the backend</code> .
Greenplum Microsoft SQL Server DW	Amazon Redshift	Resolvido um problema em que a opção <code>Apply RTRIM to string columns</code> não era exibida na interface do usuário.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Adicionado o suporte para o Babelfish para Aurora PostgreSQL como plataforma de destino. Agora, os usuários podem criar um relatório de avaliação para estimar a migração do SQL Server para o Babelfish para Aurora PostgreSQL.
Netezza	Amazon Redshift	Estratégias de otimização aprimoradas nas configurações do projeto.
SAP ASE	Aurora PostgreSQL PostgreSQL	Implementada a capacidade de gerar nomes exclusivos para índices.

Origem	Destino	Novidades, aprimoramentos ou correções
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Corrigido um problema com uma coluna de índice duplicada no script de destino.
Snowflake	Amazon Redshift	Resolvido um problema em que as opções Ocultar esquemas vazios, Ocultar bancos de dados vazios e Ocultar bancos de dados/esquemas do sistema não eram exibidas na interface do usuário.
Teradata	Amazon Redshift RSQL	Foi adicionado suporte para conversão de scripts de MultiLoad trabalho do Teradata em scripts RSQL do Amazon Redshift.
Teradata	Amazon Redshift RSQL	Corrigido um problema com a conversão de variáveis de substituição em FastLoad FastExport scripts.
Teradata	Amazon Redshift RSQL	Corrigido um problema em que os itens de ação não eram exibidos na guia Itens de ação depois de sair da guia Resumo.
Teradata	Amazon Redshift RSQL	Resolveu um problema em que ocorria um erro após a geração do relatório durante a conversão de FastExport scripts.
Teradata	Amazon Redshift RSQL	Problemas de formatação resolvidos após a conversão de scripts de shell.
Teradata	Amazon Redshift RSQL	Foi corrigido um problema para que o AI 13177 agora fosse comentado no script convertido.
Teradata	Amazon Redshift	Corrigiu uma conversão interrompida de tabelas temporais.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Melhorada a conversão da instrução SET QUERY_BAND .
Teradata	Amazon Redshift	Corrigida uma conversão interrompida da operação NORMALIZE .
Vertica	Amazon Redshift	Melhorou a descrição do AI 17008.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 655

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift RSQL	Foi corrigido um problema para garantir que todos os problemas de avaliação apareçam nos relatórios quando FastLoad ou MultiLoad forem usados.
Teradata	Amazon Redshift RSQL	Foi adicionado suporte para conversão de scripts de FastExport trabalho do Teradata em scripts RSQL do Amazon Redshift.
Teradata	Amazon Redshift RSQL	Foi corrigido um problema para garantir que a ação Salvar manifesto no S3 fosse ativada no modo offline durante o usoFastLoad.
Teradata	Amazon Redshift RSQL	Foi corrigido um problema para garantir que as regras de mapeamento fossem aplicadas a scripts como FastLoad o.
Greenplum	Amazon Redshift	Aumentada a versão mínima do driver compatível com o Greenplum para 42.2.5.

Origem	Destino	Novidades, aprimoramentos ou correções
Greenplum	Amazon Redshift	Foi adicionada uma conexão com o Greenplum via SSL com a versão 42.2.5 ou superior do driver.
Oracle DW	Amazon Redshift	Suporte aprimorado para executar funções escalares personalizadas definidas pelo usuário (UDF) em outra UDF.
Oracle DW	Amazon Redshift	Corrigido um problema em que as funções não eram aplicadas ao banco de dados com esse erro: <code>Failed to compile udf</code> .
Oracle DW	Amazon Redshift	Melhorada a conversão usando instruções de tipo apropriadas, como <code>pls-type</code> para parâmetros <code>%ROWTYPE</code> .
Teradata	Amazon Redshift RSQL	Resolvido um problema em que os problemas de avaliação do tipo de informação não apareciam no relatório.
Teradata	Amazon Redshift RSQL	Resolvido um erro do transformador após a conversão de alguns scripts.
Teradata	Amazon Redshift RSQL	Foi corrigido um problema para que um erro agora seja comentado no script convertido.
Teradata	Amazon Redshift	Resolveu um problema em que <code>FastExport ->EXPORT -> 'null'</code> era exibido em vez de <code>'CAST'</code> após a conversão.
Teradata	Amazon Redshift	Resolvido um problema em que algumas funções de um pacote de extensão falhavam quando aplicadas com <code>Cause: [JDBC Driver]String index out of range: 0</code> ao usar a versão 1.2.43 do driver
Teradata	Amazon Redshift	Conversão de tabela SET: Emulação de tabela SET adicionada para instruções de inserção e seleção.
Teradata	Amazon Redshift	CAST: suporte de conversão adicional de tipos de dados.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Corrigida uma conversão quebrada para “other_current_time_01”
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversão aprimorada dos comandos do Teradata — campo FastExport
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversão aprimorada de comandos do Teradata — layout FastExport
Oracle	PostgreSQL Aurora PostgreSQL	Resolvido um problema em que o script de destino de objetos com a INSTRUÇÃO SAVE EXCEPTIONS era alterado após a reconversão.
Oracle	PostgreSQL Aurora PostgreSQL	Resolvido um problema em que um campo errado era especificado na cláusula ORDER BY após a conversão <code>proc_cursor_with_calc_columns</code> .
Oracle	PostgreSQL Aurora PostgreSQL	Resolvido: uma instrução de <code>aws_oracle_ext\$array_id\$temporary</code> variável extra é necessária em uma conversão ASSOCIATIVE COLLECTION.
Oracle	PostgreSQL Aurora PostgreSQL	Resolvido: a conversão errada de uma CHAVE PRIMÁRIA com o mesmo nome de um ÍNDICE pertencente à mesma tabela.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 654

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL L Aurora PostgreSQL L	Resolvido um problema com pseudocolunas de consulta hierárquica, erro de análise de colunas PRIOR.
Oracle	PostgreSQL L Aurora PostgreSQL L	Resolvido um problema para converter corretamente um comentário de várias linhas contendo uma barra e um asterisco (/ *).
Oracle	PostgreSQL L Aurora PostgreSQL L	Foi adicionada a emulação USER_COL_COMMENTS da visualização do sistema ao pacote de extensões.
Oracle	PostgreSQL L Aurora PostgreSQL L	Foi aprimorada a conversão de literais citados.
DB2 LUW	PostgreSQL L	Foi aprimorada a conversão de instruções LABEL que adicionam ou substituem rótulos nas descrições de tabelas, visualizações, aliases ou colunas.

Origem	Destino	Novidades, aprimoramentos ou correções
	Aurora PostgreSQL	
Oracle	Nenhum	Tabela do sistema SYS.USER\$ substituída pela visualização DBA_USERS e consultas aprimoradas.
Oracle DW	Amazon Redshift	Consultas de metadados do Oracle DW foram atualizadas.
Teradata	Amazon Redshift RSQL	Foi adicionado suporte para conversão de scripts shell, Teradata e Teradata FastLoad Basic Teradata Query (BTEQ) em scripts RSQL do Amazon Redshift.
Teradata BTEQ	Amazon Redshift RSQL	Resolvido um problema em que “merge_01” era convertido incorretamente.
Teradata BTEQ	Amazon Redshift RSQL	Problema resolvido para que End or Identify (EOI) apareça no final de um script em uma nova linha.
Azure Synapse	Amazon Redshift	Melhoria da mensagem de erro fornecida quando uma senha incorreta é fornecida para o Azure Synapse.
Teradata	Amazon Redshift	Foi aprimorada a conversão da instrução UPDATE para transmitir o nome de alias correto de acordo com o padrão Teradata.
Teradata	Amazon Redshift	Resolvido um erro de conversão do cursor em que as ações não foram recebidas.
Teradata	Amazon Redshift	Resolvido um problema em que uma conversão TD_NORMALIZE_OVERLAP estava eliminando linhas.
Teradata	Amazon Redshift	Agora oferece suporte à verificação estrita de data para a função TO_DATE aprimorada.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Foi aprimorada a conversão da função integrada TO_NUMBER (n).
Teradata	Amazon Redshift	Resolvido um problema em que a categoria Esquemas estava ausente da árvore de metadados.
Greenplum	Amazon Redshift	Foi adicionada a seleção GP_SEGMENT_ID à lista ao criar uma partição virtual para uma tabela Greenplum.
Greenplum	Amazon Redshift	Resolvido um problema em que as funções não eram aplicadas no destino.
MS SQL Server DW	Amazon Redshift	Resolvido um problema em que ocorria um erro de transformação após a conversão sem o AI 9996.
MS SQL Server DW	Amazon Redshift	Resolvido um problema em que um erro foi registrado ao abrir o assistente do pacote de extensões.
MS SQL Server DW	Amazon Redshift	Resolvido um problema em que um estilo incorreto de comentários era usado para funções do Redshift Python.
Netezza	Amazon Redshift	Resolveu um problema em que um pacote de extensão Netezza—Redshift com um AWS perfil falhava na criação.
Teradata	Amazon Redshift RSQL	Conversão aprimorada do comando FastLoad SESSIONS.
Teradata	Amazon Redshift RSQL	Relatórios de avaliação de FastLoad scripts aprimorados.
Teradata	Amazon Redshift RSQL	Implementou a ação FastLoad WRITER Save to S3.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift RSQL	Resolveu um problema em que FastLoad os botões Salvar script\ Salvar manifesto no s3 não estavam ativos.
Teradata	Amazon Redshift RSQL	Resolveu um problema em que o FastLoad multifile_script criava apenas um arquivo de manifesto após a conversão, em vez dos três arquivos esperados.
Teradata	Amazon Redshift RSQL	Resolveu um problema em FastLoad que pastas extras eram exibidas em um caminho do S3.
Teradata	Amazon Redshift RSQL	Resolveu um problema em FastLoad que havia o nome incorreto do arquivo de manifesto em um caminho do S3.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 653

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL L Aurora PostgreSQL L	Implementada a capacidade de converter SQL dinâmico criado em funções ou procedimentos chamados.
Oracle	PostgreSQL L	Conversão SQL dinâmica aprimorada: parâmetros IN como variáveis de ligação.

Origem	Destino	Novidades, aprimoramentos ou correções
	Aurora PostgreSQL	
Oracle DW 18, 19	Amazon Redshift	Melhorias de conversão de Oracle para Redshift implementadas: conversão aprimorada integrada. LISTAGG agregado; LISTAGG analítico.
Oracle DW 18, 19	Amazon Redshift	Melhorias na conversão de Oracle para Redshift implementadas: consulte novos atributos.
Vertica	Amazon Redshift	Melhorias na conversão de Vertica para Redshift implementadas: conexão SSL para JDBC com SSL=true.
MS SQL Server DW	Amazon Redshift	Melhorias na conversão do MS SQL Server para Redshift: Tabelas externas.
Teradata	Amazon Redshift	Melhorias na conversão de Teradata para Redshift: operações aritméticas de tipos de dados INTERVAL.
Teradata	Amazon Redshift	Melhorias na conversão de Teradata para Redshift: Suporte para aliases de colunas laterais.
Oracle	Nenhum	As seguintes consultas do Loader agora usam DBA_USERS em vez de SYS.USER\$: <ul style="list-style-type: none"> get-tree-path-list-by-name-path.sql estimate-table-or-view-constraints-by-schema.sql estimate-table-or-view-constraints-by-selected-schemas.sql
Teradata	Amazon Redshift	Alinhamento aprimorado dos comentários quando o SCT converte macros Teradata em procedimentos armazenados do Redshift.
Oracle DW	Amazon Redshift	Melhorada a conversão dos elementos do formato de data/hora: TO_DATE, TO_TIMESTAMP e TO_TIMESTAMP_TZ

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Resolvido o erro de conversão do cursor Teradata.
Teradata	Amazon Redshift	Problema resolvido que fazia com que os atributos de TD_NORMALIZE_OVERLAP fossem eliminados durante a conversão.
Teradata	Amazon Redshift	Resolvido um problema em que a função MAX era ignorada quando o SCT convertia uma consulta.
Teradata	Amazon Redshift	O SCT agora converte a função Teradata CHARACTERS na função Redshift LENGTH.
Teradata	Amazon Redshift	O SCT agora suporta a conversão de FORMAT em TO_CHAR para os formatos mais usados.
Todos	Todos	Foi aprimorada a conversão de rotinas criptografadas.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 652

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQL	Adicionado bloqueio de aplicativos para funções <code>sp_getapplock</code> e <code>sp_releaseapplock</code> .
Nenhum	Amazon Redshift	Melhoria da Interface de Linha de Comandos (CLI): implementado o modo de Comando de Script.
Oracle	PostgreSQL	Implementada a amostragem de parâmetros de rotina dentro do SQL dinâmico.

Origem	Destino	Novidades, aprimoramentos ou correções
	Aurora PostgreSQL L	
Oracle	PostgreSQL L Aurora PostgreSQL L	Melhorias na conversão de SQL dinâmico criado em funções ou procedimentos chamados.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL L	Cada função do lambda é implantada e configurada por meio de política apenas uma vez, e as funções do lambda comuns são reutilizadas para todas as origens possíveis.
DB2 LUW	PostgreSQL L	Problema resolvido que causava a mensagem de erro “9996: Gravidade crítica: Ocorreu um erro no transformador” ao usar o DB2 LUW como origem.
Teradata	Amazon Redshift	Suporte para expressões de tabela recursivas no próximo lançamento do Amazon Redshift.
Azure Synapse	Amazon Redshift	Implementou regras de otimização de esquema.
Teradata	Amazon Redshift	Suporte à conversão de fuso horário de macros Teradata para procedimentos armazenados do Redshift.
Teradata	Amazon Redshift	Suporte aritmética em valores de PERIOD.
Teradata	Amazon Redshift	Suporte à conversão de expressões de tabela comuns recursivas do Teradata (CTE RECURSIVA).

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Suporte a identificadores com distinção entre maiúsculas e minúsculas por meio da configuração do usuário, <code>enable_case_sensitive_identifier</code> . Então, "COLUMN_NAME" e "Column_Name" se tornam nomes de colunas diferentes.
Teradata	Amazon Redshift	Foi resolvido o problema do tipo de dados decimal para que os campos decimais fossem convertidos com a mesma precisão.
Teradata	Amazon Redshift	Problema resolvido com a conversão aritmética de intervalos para que a subtração aritmética de intervalos seja convertida corretamente.
Teradata	Amazon Redshift	Transmissão aprimorada do tipo Teradata NUMBER to DATE.
Teradata	Amazon Redshift	Transmissão aprimorada do tipo Teradata DATE to NUMBER.
Teradata BTEQ	Amazon Redshift	Foi aprimorada a conversão do tipo de dados PERIOD.
Teradata	Amazon Redshift	Problema resolvido com o carregamento de metadados de uma tabela com colunas GEOMETRY para que agora ela seja carregada corretamente do Teradata.
Teradata	Amazon Redshift	Suporte à conversão de instruções de mesclagem ao converter macros Teradata em procedimentos armazenados do Redshift.
Teradata	Amazon Redshift	Foi aprimorada a conversão de macros simples ao migrar do Teradata para o Redshift.
Teradata	Amazon Redshift	Foi garantida a conversão da instrução UPDATE do Teradata para transmitir o nome de alias correto de acordo com o padrão Teradata.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 651

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	AWS SCT Relatórios aprimorados para atualizar os links para os itens de ação de conversão recomendados listados.
MS SQL Server	PostgreSQL	Foi adicionado suporte para conversão de funções STR().
MS SQL Server	PostgreSQL	Foi adicionado suporte para converter o operador EXOR bit a bit (^ no Microsoft SQL Server) em PostgreSQL como operador #.
Oracle	PostgreSQL	Resolveu um problema em que a <code>aws_oracle_ext.UNI STR(null)</code> função do pacote de AWS SCT extensões ficava suspensa NULL em um destino do PostgreSQL. AWS SCT agora lida com NULL o.
Teradata BTEQ	Amazon Redshift RSQL	Melhorias de conversão feitas para resolver um problema em que a conversão do Amazon Redshift RSQL MERGE gerava um erro de transformação.
Oracle DW	Amazon Redshift	Implementou integrações aprimoradas.
Oracle DW	Amazon Redshift	Foram adicionados aprimoramentos baseados em atributos de metadados, incluindo particionamento de lista automática (TBL_PART_LIST_AUTO), lista de várias colunas (TBL_PART_MULTI_LIST) e referência de intervalo (TBL_PART_RANGE_INTERVAL_REF).
Nenhum	Amazon Redshift	Aumento dos limites da tabela de partições físicas usadas para conversões UNION ALL.
Teradata	Amazon Redshift	Melhorias na conversão feitas no escopo dos relatórios de avaliação.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Melhorias na conversão feitas em conversões complexas de MACRO do Teradata.
Teradata	Amazon Redshift	Foi aprimorada a conversão de macros Teradata para procedimentos armazenados do Amazon Redshift ao comentar SQL não suportado.
Teradata	Amazon Redshift	Resolvido um problema em que a conversão de macros do Teradata em procedimentos armazenados do Amazon Redshift resultava em referências erradas de nome de alias.
Teradata	Amazon Redshift	Melhorada a conversão da instrução Teradata QUALIFY.
Teradata	Amazon Redshift	Conversão aprimorada para transferir comentários para o Amazon Redshift e reter um histórico das alterações realizadas na visualização.
Teradata	Amazon Redshift	Resolvido um problema em que a cláusula RESET WHEN não resultava na conversão correta.
Teradata BTEQ	Amazon Redshift	Foi aprimorada a conversão de scripts BTEQ que contêm instruções MERGE.
Teradata	Amazon Redshift	Foram adicionadas funções integradas para melhorar a conversão dos campos do tipo de dados PERIOD.
Microsoft SQL Server	Amazon Redshift	Mapeamento aprimorado do tipo de dados de transformação para o tipo de dados TIME.
Todos	Todos	Foi adicionado acesso à publicação inicial do manual de referência da CLI AWS Schema Conversion Tool em formato PDF. Consulte AWS Schema Conversion Tool Referência da CLI .

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 650

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	<p>Uso atualizado e aprimorado de agentes extratores, incluindo:</p> <ul style="list-style-type: none"> • Uma configuração para uso com armazenamento compartilhado e um agente de cópia dedicado. • Exportação e importação de tarefas de extração de dados de um projeto para outro. • Suporte ao Azure SQL Data Warehouse (Azure Synapse) como origem. • Usando o particionamento nativo do Netezza. <p>Para obter mais informações, consulte Como migrar dados de um data warehouse on-premises para o Amazon Redshift.</p>
Todos	Amazon RDS PostgreSQL 13	AWS SCT agora oferece suporte ao Amazon RDS PostgreSQL 13 como destino.
Microsoft SQL Server	Aurora PostgreSQL	Foi aprimorada a conversão de um conjunto de resultados de um procedimento do Microsoft SQL Server para um destino do Aurora PostgreSQL.
Oracle DW	Amazon Redshift	Melhorias na conversão de Oracle para Amazon Redshift implementadas.
Oracle DW	Amazon Redshift	Melhorias implementadas na conversão de instruções SQL dinâmicas.
Oracle DW	Amazon Redshift	Melhorias implementadas na conversão de UDF de SQL.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle DW	Amazon Redshift	Mensagem esclarecida que AWS SCT não suporta a conversão de TABELAS EXTERNAS.
Oracle DW	Amazon Redshift	Funções de conversão integradas aprimoradas.
Teradata BTEQ	Amazon Redshift RSQL	Melhoria no tratamento dos parâmetros de substituição nos scripts BTEQ ao usar a GUI. AWS SCT
Microsoft SQL Server DW	Todos	Atualizada a versão mínima compatível do driver JDBC para Microsoft SQL Server, Azure e Azure Synapse.
Microsoft SQL Server		
Azure		
Azure Synapse		

Problemas resolvidos:

- Teradata: melhorias adicionais na conversão de macro [RESOLVIDO]
- Caracteres especiais escapavam do destino causando erros de SQL e foi necessário retrabalhar para colocá-los de volta [RESOLVIDO]
- Melhorias gerais

Notas de lançamento do AWS SCT Build 649

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server DW	Amazon Redshift	Melhorias na conversão de MSSQL para Amazon Redshift para oferecer suporte a tabelas temporais.
Oracle DW	Amazon Redshift	Implementou aprimoramentos de funções integradas, como: Funções de conversão <ul style="list-style-type: none">• TO_BINARY_DOUBLE• TO_BINARY_FLOAT• TO_NUMBER• TO_DATE• TO_TIMESTAMP• TO_TIMESTAMP_TZ• TO_DSINTERVAL• TO_YMINTERVAL• VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	Aprimoramentos de função implementados para Processamento Aproximado de Consultas, como: Funções agregadas <ul style="list-style-type: none">• ANY_VALUE• APPROX_COUNT_DISTINCT

Origem	Destino	Novidades, aprimoramentos ou correções
		<ul style="list-style-type: none"> • APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • TO_APPROX_COUNT_DISTINCT
Teradata	Amazon Redshift	<p>Aprimoramentos de conversão implementados para seleção automática de chaves de classificação e distribuição do Teradata. O mecanismo de banco de dados seleciona chaves de classificação e distribuição automaticamente. Introduziu um botão de rádio chamado Use o ajuste automático da tabela do Amazon Redshift nas configurações de projetos atuais > Estratégias de otimização > Caixa de diálogo Estratégia de seleção inicial de chaves.</p>
Teradata	Amazon Redshift	Carregador AWS SCT de tabelas aprimorado para garantir o AWS SCT carregamento de todas as tabelas da Teradata.
Teradata	Amazon Redshift	Implementou aprimoramentos de conversão para que o Amazon Redshift ofereça suporte a padrões de subconsulta correlacionados que incluem uma cláusula simples WHERE NOT EXISTS.
Teradata	Amazon Redshift	Foi adicionado suporte para o uso de comandos ECHO em macros.
DB2 LUW	PostgreSQL	<p>Suporte implementado para conversão de DYNAMIC RESULTS SETS, incluindo:</p> <ul style="list-style-type: none"> • Cláusula do cursor WITH RETURN/WITH RETURN TO CLIENT • conversão de cláusulas de rotina DYNAMIC RESULT SETS
	Aurora PostgreSQL	

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server Oracle DB2 LUW SAP ASE	Aurora PostgreSQL	Suporte implementado para o PostgreSQL atual do Aurora RDS como destino.
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	Implementado o suporte para o MariaDB 10.5 como destino.
Microsoft SQL Server	MariaDB	Suporte implementado de INSERT-RETURNING, que retorna um conjunto de resultados das linhas inseridas.
Oracle	Aurora PostgreSQL	Foi adicionado suporte à função XMLFOREST para conversão do Oracle para o Aurora PostgreSQL.

Problemas resolvidos:

- Melhorias gerais.

Notas de lançamento do AWS SCT Build 648

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL L Amazon Aurora Edição Compatível com PostgreSQL L	Modo de aplicação personalizado do pacote de extensões Aurora PostgreSQL implementado: operadores para tipos numéricos/de data e texto.
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL L	Configuração do Aurora PostgreSQL Lambda Invoke implementada: criação da extensão <code>aws_lambda</code> ; atribuição do perfil do IAM ao cluster Aurora PostgreSQL. <ul style="list-style-type: none"> • Oracle — E-mails, trabalhos, filas, arquivos WebAgent • DB2: e-mails, tarefas, arquivos • Microsoft SQL Server: e-mails, agente
Oracle	PostgreSQL L	A refatoração de conversão de instruções FORALL foi implementada: <ul style="list-style-type: none"> • Instrução FORALL • FORALL ... SAVE EXCEPTIONS • RETURNING INTO com BULK COLLECT • Coleção do sistema SQL%BULK_EXCEPTIONS

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle DW 18, 19	Amazon Redshift	Melhorias de conversão de Oracle para Amazon Redshift implementadas: conversão aprimorada integrada. LISTAGG agregado; LISTAGG analítico.
Oracle DW 18, 19	Amazon Redshift	Melhorias na conversão de Oracle para Amazon Redshift implementadas: consulte novos atributos.
Vertica	Amazon Redshift	Melhorias na conversão de Vertica para Amazon Redshift implementadas: conexão SSL para JDBC com SSL=true.
Microsoft SQL Server DW	Amazon Redshift	Melhorias na conversão do Microsoft SQL Server para Redshift: Tabelas externas.
Teradata	Amazon Redshift	Melhorias na conversão de Teradata para Redshift: operações aritméticas de tipos de dados INTERVAL.
Teradata	Amazon Redshift	Melhorias na conversão de Teradata para Redshift: Suporte para aliases de colunas laterais.

Problemas resolvidos:

- Melhorias gerais

Notas de lançamento do AWS SCT Build 647

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Microsoft SQL Server	O RDS agora oferece suporte ao atributo Database Mail.
Microsoft SQL Server	MySQL	Implementação do nome máximo de cada tipo de identificador: O tamanho máximo dos nomes de objetos (por exemplo, tabelas, restrições, colunas) no SQL Server é de 128 caracteres. O tamanho

Origem	Destino	Novidades, aprimoramentos ou correções
		<p>máximo dos nomes de objetos no MySQL é de 64 caracteres. Para gravar objetos convertidos no banco de dados MySQL, você precisa encurtar seus nomes. Para evitar nomes duplicados após o corte, você precisa adicionar a “soma de verificação” do nome do objeto original aos novos nomes.</p> <p>Corte nomes com mais de 64 caracteres da seguinte forma:</p> <pre>[first N chars]() + "" + [checksum]()</pre> <pre>[first N chars] = 64 - 1 - [length of checksum string]</pre> <p>Por exemplo: . example_of_a_test_schema_with_a_name_length_g reater_than_64_characters ?? example_of_a_test _schema_with_a_name_length_greater_than_64_97 03</p>
Oracle	MySQL/ Aurora MySQL	Implementado o carregamento e a conversão de comentários em objetos de armazenamento. Por exemplo, processamento de comentários em tabelas e processamento de comentários nas colunas Tabela/Visualização.
Teradata	Amazon Redshift	Foi adicionado suporte para conversão de tipo de dados TIME.
Teradata	Amazon Redshift	Melhorias na conversão: TD_NORMALIZE_OVERLAP implementado.
Microsoft SQL Server DW	Amazon Redshift	Melhorias na conversão: SELECT com a cláusula WITH; SELECT sem FROM
Todos	Todos	AWS SCT Data Migration Service Assessor (DMSA) — Esse novo recurso permite avaliar vários servidores e receber um relatório resumido que mostra a melhor direção de destino para seu ambiente.

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	AWS SCT Assistente — A comparação de alvos agora mostra as diferenças entre os alvos em uma única exibição de tabela.
Todos	Todos	Tree Filter UI: O filtro de metadados redesenhado lida com padrões de filtragem mais complexos.
Todos	Todos	Relatório de avaliação: A seção de Aviso redesenhada fornece uma descrição melhor e uma compreensão mais clara de um problema.

Problemas resolvidos:

- Melhorias gerais
- Extratores de dados — A subtarefa falhou com `ConcurrentModificationException` [RESOLVIDO].
- Microsoft SQL Server para MySQL: comprimentos máximos do identificador [RESOLVIDO].

Notas de lançamento do AWS SCT Build 646

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL	Implementação aprimorada do modelo no formato TM.
Oracle	PostgreSQL	A implementação da máscara no formato SP fornece suporte básico para o sufixo SP, somente para o idioma inglês.
Oracle	PostgreSQL	Tratamento de nomes de objetos longos da Oracle — AWS SCT agora manipula nomes de objetos longos da Oracle de acordo com o atributo de comprimento máximo do identificador alvo.
	Amazon Redshift	Amazon Redshift codificando AZ64 com AWS SCT — Codificação de compressão AZ64 adicionada para alguns tipos de dados
Teradata	Amazon Redshift	Foi adicionado suporte para conversão de transações implícitas.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Adicionado o suporte para funções geoespaciais internas do Teradata: Métodos <code>ST_LineString</code>
Greenplum	Amazon Redshift	Conversão de sequência Greenplum: Foram adicionados os próximos itens às guias Propriedades: valor mínimo, valor máximo, incremento, ciclo.
Greenplum	Amazon Redshift	Resolvedor: Foi adicionada a resolução do tipo de dados "char".
Greenplum	Amazon Redshift	Comprimento da conversão de caracteres: Conversão PL/pgSQL atualizada para o tipo de caractere.
Greenplum	Amazon Redshift	Resolveu um problema com a seleção da chave de distribuição do Greenplum em que uma tabela tinha CHAVE DE DISTRIBUIÇÃO, mas não AWS SCT conseguia reconhecer e buscar a tabela como DISTRIBUÍDA ALEATORIAMENTE.
Teradata	Amazon Redshift	Suporte ao cursor Teradata: Foi adicionado suporte para conversão de cursores.
Teradata	Amazon Redshift	Colunas de identidade: Foi adicionado suporte para conversão de colunas de identidade.
Teradata	Amazon Redshift	Tipos de dados INTERVAL: Foi adicionado suporte para conversão de tipos de dados INTERVAL.

Problemas resolvidos:

- Melhorias gerais
- Greenplum: Não foi possível executar a conversão devido ao erro no log [RESOLVIDO].
- MSSQL: PostgreSQL: erro do transformador ao converter a função LAG [RESOLVIDO].
- MSSQL: PostgreSQL: SCOPE_IDENTITY [RESOLVIDO].
- AWS SCT pendurado em projetos de DW [RESOLVIDO].

- É necessária uma regra de mapeamento para remover espaço adicional no nome da coluna em AWS SCT [RESOLVIDO].

Notas de lançamento do AWS SCT Build 645

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Forneça uma solução para resolver visualizações não totalmente qualificadas do Teradata (nomes de exibição ou objetos não totalmente qualificados na exibição).
Teradata	Amazon Redshift	Foi adicionado suporte à função ASCII para computar nós.
Teradata	Amazon Redshift	Quando AWS SCT detecta dados de vários bytes em um Teradata CHAR definido como CHAR(N), eles são convertidos no Amazon VARCHAR(3*N) Redshift.
Teradata	Amazon Redshift	Fornece a conversão de CAST no Teradata entre datas e números. <ul style="list-style-type: none"> • <code>SELECT Cast('2020-07-17' AS BIGINT)</code> • <code>SELECT Cast(20200630 - 19000000 AS DATE)</code>
Teradata	Amazon Redshift	Suporte à conversão de tipos de dados Teradata PERIOD em duas colunas TIMESTAMP do Amazon Redshift: <ul style="list-style-type: none"> • <code>PERIOD(TIMESTAMP)</code> • <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code>
Teradata	Amazon Redshift	Suporte à conversão da função Teradata RANK com cláusula RESET WHEN.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Suporte aprimorado de CAST em conversões explícitas de tipos de dados e CASTS implícitos em expressões.
Teradata	Amazon Redshift	Relatório de padrões de subconsulta correlacionados não suportados. Para obter mais informações, consulte Subconsultas correlacionadas no Guia do desenvolvedor de banco de dados do Amazon Redshift.
none	Amazon Redshift	Tabelas aprimoradas limitam o suporte para tipos de nós RA3.
Teradata	Amazon Redshift	Foi adicionado suporte para extração de dados geoespaciais do Teradata. Para obter mais informações, consulte Consultar dados espaciais no Amazon Redshift no Guia do desenvolvedor de banco de dados do Amazon Redshift.
Microsoft SQL Server	PostgreSQL	A opção <code>convert_procedures_to_function</code> foi adicionada.

Problemas resolvidos:

- Melhorias gerais

Notas de lançamento do AWS SCT Build 644

As alterações nas AWS SCT versões 1.0.643 são mescladas na versão 1.0.644. AWS SCT

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Várias melhorias na conversão. <ul style="list-style-type: none"> • Conversões aprimoradas de QUALIFY com o alias da tabela. • Conversões aprimoradas com o operador IN.

Origem	Destino	Novidades, aprimoramentos ou correções
		<ul style="list-style-type: none">• Melhorada a conversão com o operador LIKE.• Conversões aprimoradas com problemas de destaque no código convertido.• Conversões aprimoradas com uma ordem incomum de cláusulas WHERE, QUALIFY em SQL.• Erros fixos do transformador ocorreram durante a conversão do procedimento UPD_FT_SVC_TRANS_BH_CBH_IND nas construções JOIN().• Foi aprimorada a conversão de macros em procedimentos armazenados. <p>Foram adicionados comandos AWS SCT CLI especiais que podem analisar os scripts sql/bteq fornecidos e gerar um relatório sobre o número de estruturas de sintaxe encontradas no código-fonte.</p> <ul style="list-style-type: none">• Contagem de comandos BTEQ• Contagem de HANDLERS• Contagem de casos CAST• Contagem de casos de DML/DDI• Contagem de DMLs em visualizações atualizáveis <p>Foi adicionado um item de ação do relatório de avaliação: colunas Teradata com formatos de data personalizados não são suportadas no Amazon Redshift.</p>

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Funcionalidade adicionada para salvar scripts de instalação do pacote de extensão.</p> <p>Nível de severidade alterado para AI 5334.</p> <p>Melhor desempenho do uso de um registro como variável de pacote IMPLEMENTATION .</p> <p>Adicionado suporte à função de agregação XMLAGG</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	Adicionado carregamento e conversão de comentários na implementação de objetos de armazenamento.
MS SQL DW	Amazon Redshift	<p>Melhoria na conversão: problema resolvido com PATINDEX.</p> <p>Melhorias da UI:</p> <ul style="list-style-type: none"> • Salvar como SQL para implementação da árvore de origem. • Adição de lógica extra para geração de scripts em vários arquivos.
Vertica	Amazon Redshift	Melhoria de UI: Salvar como SQL para implementação da árvore de origem.

Problemas resolvidos:

- Melhorias gerais nas conversões entre Teradata e o Amazon Redshift
- Correção de bugs e melhorias da UI em geral

Notas de lançamento do AWS SCT Build 642

Alterações na AWS Schema Conversion Tool versão 1.0.642.

Note

AWS Schema Conversion Tool (AWS SCT) as alterações da compilação 1.0.642 são aplicáveis ao Windows, Ubuntu e Fedora. Não há versão 1.0.642 para macOS.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SSIS	AWS Glue	Implementou a conversão de pacotes ETL do Microsoft SQL Server Integration Services (SSIS) para o AWS Glue. Para ter mais informações, consulte Como converter SSIS em AWS Glue com a AWS SCT .
Oracle	MariaDB/SQL MODE=C LE/MySQL/ Amazon Aurora MySQL	Implementada a seção de instrução PL/SQL na cláusula WITH.
Oracle	PostgreSQL/Aurora PostgreSQL	Adicionado suporte para <code>DBMS_SESSION.RESET_PACKAGE</code> e <code>DBMS_SESSION.MODIFY_PACKAGE</code> .
Vertica	Amazon Redshift	Habilitação da exportação de scripts SQL de um banco de dados Vertica para o Amazon Redshift.

Problemas resolvidos:

- Aprimoramento do relatório de avaliação.
- Aprimoramento da UI do relatório de avaliação.
- Adicionada a capacidade de alterar as configurações da JVM a partir da interface do usuário.
- Melhorias gerais.

Notas de lançamento para a AWS SCT compilação 641

Alterações na AWS Schema Conversion Tool versão 1.0.641.

Note

AWS Schema Conversion Tool (AWS SCT) as alterações da compilação 1.0.641 são aplicáveis ao Windows, Ubuntu e Fedora. Não há versão 1.0.641 para macOS.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle/ MS SQL/ MySQL/ PostgreS QL/Db2 LUW	Todos	Produza cálculos do Time Report no arquivo.csv.
Teradata	Amazon Redshift	Adicionado o suporte para a função CSUM. Foi adicionado suporte para extração tipos de dados geoespaciais do Teradata.
Teradata	Todos	Foi adicionado suporte para converter colunas IDENTITY.
Greenplum	Amazon Redshift	Foi adicionado suporte para o estilo de distribuição AUTO durante a conversão da tabela Greenplum.
SAP ASE	Todos	Produza cálculos do Time Report no arquivo.csv.

Resolvido:

- Várias correções de bugs.
- Várias melhorias de desempenho.

Notas de lançamento do AWS SCT Build 640

As alterações nas AWS SCT versões 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 e 1.0.640 são mescladas na versão 1.0.640. AWS SCT

Note

AWS SCT as alterações da compilação 1.0.640 são aplicáveis ao Windows, Ubuntu e Fedora. Elas não se aplicam ao MacOS.

Você não pode instalar a AWS SCT versão 1.0.640 ou superior no Apple macOS. AWS SCT a versão 1.0.632 foi a última versão a suportar a instalação no macOS da Apple.

Nas tabelas a seguir, você pode encontrar listas de recursos e correções de bugs para as versões da AWS Schema Conversion Tool que foram combinadas na versão 1.0.640. Essas tabelas agrupam recursos e correções de bugs pelo mecanismo de origem.

Tópicos

- [Alterações do release 1.0.640 do Oracle](#)
- [Alterações da release 1.0.640 do Microsoft SQL Server](#)
- [Alterações do MySQL versão 1.0.640](#)
- [Alterações da release 1.0.640 do PostgreSQL](#)
- [Alterações do release 1.0.640 do Db2 LUW](#)
- [Alterações na release 1.0.640 do Teradata](#)
- [Alterações da release 1.0.640 para outros mecanismos](#)

Alterações do release 1.0.640 do Oracle

A tabela a seguir lista as alterações do build 1.0.640 nas quais o Oracle é o mecanismo de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL	Implementada a conversão de código SQL em aplicativos Java e Pro*C.

Origem	Destino	Novidades, aprimoramentos ou correções
	Aurora PostgreSQL L	
Oracle	PostgreSQL L Aurora PostgreSQL L	<p>Desempenho aprimorado das funções a seguir quando usadas em uma cláusula WHERE:</p> <ul style="list-style-type: none"> • <code>aws_oracle_ext.to_date</code> • <code>aws_oracle_ext.to_char</code> • <code>aws_oracle_ext.to_number</code> • <code>aws_oracle_ext.sysdate</code> • <code>aws_oracle_ext.sys_context</code>
Oracle	RDS MariaDB 10.4	Adição de suporte ao RDS MariaDB 10.4 para todos os fornecedores de processamento transacional online (OLTP).
Oracle	PostgreSQL L/Aurora PostgreSQL L	<p>Adição de suporte para <code>DBMS_UTILITY.GET_TIME</code>.</p> <p>Adição das seguintes emulações:</p> <ul style="list-style-type: none"> • <code>DBMS_UTILITY.GET_TIME</code> • <code>DBMS_UTILITY.FORMAT_CALL_STACK</code> • <code>DBMS_UTILITY.CURRENT_INSTANCE</code>

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	MariaDB/MySQL/Aurora MySQL/Microsoft SQL Server Mode=Oracle/PostgreSQL/Aurora PostgreSQL/RDS Oracle	Adição de suporte à cláusula de compartilhamento para TABLE(DATA,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) e SEQUENCE(DATA)
Oracle	PostgreSQL/Aurora PostgreSQL/Oracle RDS	<p>A definição DEFAULT de uma coluna pode ser estendida para que o DEFAULT seja aplicado para inserção NULL explícita.</p> <p>A cláusula DEFAULT tem uma nova cláusula ON NULL. Essa nova cláusula instrui o banco de dados a atribuir um valor de coluna padrão especificado quando uma instrução INSERT tenta atribuir um valor que é avaliado como NULL.</p>
Oracle	MariaDB/MariaDB (SQL MODE=ORACLE)	Adição de suporte para "Colunas de Identidade", que são incrementadas automaticamente no momento da inserção.
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	MariaDB 10.2/MariaDB 10.3/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	A cláusula DEFAULT tem uma nova cláusula ON NULL, que instrui o banco de dados a atribuir um valor de coluna padrão especificado quando uma instrução INSERT tenta atribuir um valor avaliado como NULL.
Oracle	Oracle RDS/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Adição de suporte para colunas IDENTITY.
Oracle	MySQL 8.x	Adição de suporte para restrição CHECK.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Implementação da verificação ANYDATA IS NULL/IS NOT NULL usando rotina do pacote de extensão.</p> <p>Implementação da emulação da função VALUE usada em uma consulta com base na função TABLE de XMLSequence.</p> <p>Adição de suporte a DBMS_LOB para as seguintes rotinas internas:</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.
Oracle	Amazon Redshift	<p>Implementação da conversão de atributos de cursor em blocos aninhados.</p> <p>O Amazon Redshift não oferece suporte a coleções. Variáveis relacionadas são convertidas como VARCHAR. Todas as operações de coleta, exceto a atribuição de uma variável a outra, são rejeitadas, incluindo o acesso a elementos de iniciação e coleta.</p> <p>Implementado o estilo de distribuição do Amazon Redshift = AUTO.</p>

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Se uma palavra não reservada no Oracle estiver reservada no PostgreSQL, o seguinte será verdadeiro:</p> <ul style="list-style-type: none"> • Se a palavra for citada, ela manterá o tamanho da letra e permanecerá citada. • Se a palavra não for citada, ela será convertida em maiúsculas e citada. <p>Implementação da capacidade de usar funções como entrada para funções LTRIM, RTRIM e TRIM.</p> <p>Expressões SELECT DISTINCT, ORDER BY devem aparecer na lista de seleção.</p> <p>Para parâmetros do cursor que seguem um parâmetro com um valor DEFAULT, AWS SCT adiciona a cláusula DEFAULT IS NULL</p> <p>Os parâmetros do cursor Source OUT são convertidos em parâmetros do cursor IN.</p> <p>Nova implementação da variável de pacote adicionando a opção "Implementação lógica de variáveis de pacote" em "Configurações de conversão". As configurações disponíveis são: "variáveis de sessão" e "objetos globais plv8". O padrão é "variáveis de sessão".</p> <p>Implementação de suporte pragma AUTONOMOUS_TRANSACTION com dblink e pg_background.</p>
Oracle	Todos	Implementação da visualização SYS_%_TAB_COMMENTS.
Oracle	PostgreSQL	Entradas variáveis para filtros não são compatíveis com o PostgreSQL. Na conversão do Oracle em PostgreSQL, se um filtro de variável for encontrado, uma exceção será relatada agora.

Origem	Destino	Novidades, aprimoramentos ou correções
Oracle	Amazon Redshift	<p>Implementação de melhorias de conversão do FOR..LOOP Cursor do código armazenado.</p> <p>Implementação de invocação de código armazenado de função/pr ocedimentos com parâmetros padrão.</p> <p>Implementação de capacidade de código armazenado para UPDATE com alias sem a cláusula WHERE.</p> <p>As funções de código armazenado implementadas pré-formam casos adicionais com SELECT From dual.</p> <p>Implementação de parâmetros Table%ROWTYPE e variáveis de pacote do código armazenado.</p> <p>O código armazenado implementado usava JAVA e procedimentos externos.</p> <p>Implementação do pacote Oracle standard em código armazenado.</p>

Alterações da release 1.0.640 do Microsoft SQL Server

A tabela a seguir lista alterações da build 1.0.640 em que o Microsoft SQL Server é o mecanismo de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft Azure/Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL/MySQL/Aurora MySQL	Adição de suporte para índices COLUMN STORE.

Origem	Destino	Novidades, aprimoramentos ou correções
Microsoft SQL Server	RDS MariaDB 10.4	Adição de suporte ao RDS MariaDB 10.4 para todos os fornecedores de processamento transacional online (OLTP).
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Adição de suporte para o atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Adição de suporte para tipos de tabela Nó de Bancos de Dados e Edge.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Adição de suporte para TEMPORAL TABLES.

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL/MariaDB	Adição de suporte para processamento DML para SQL Server Graph Architecture.
SQL Server	Aurora PostgreSQL	Adição da opção para converter parâmetros sem o prefixo par_.
Azure/SQL Server	MySQL 8.x	Adição de suporte para restrição CHECK.
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.

Origem	Destino	Novidades, aprimoramentos ou correções
SQL Server	AWS Glue (Concha Python)	<p>Melhorias de conversão, incluindo:</p> <ul style="list-style-type: none"> • Implementação da conversão de funções incorporadas em Python.String. • Implementação de EXECUTE e EXEC em código armazenado. • Implementação usando tipos de tabela.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Implementação que torna os procedimentos \$TMP opcionais.
SQL Server	MySQL/Aurora MySQL	<p>Operações aritméticas estendidas com datas.</p> <p>Emulação de construção 'TOP (expressão) WITH TIES.</p> <p>Depois de chamar procedimentos com o refcursor gerado fora, o refcursor agora é fechado.</p> <p>A definição de um nível de isolamento GLOBAL não é compatível com o Aurora MySQL. Somente o escopo da sessão pode ser alterado. O comportamento padrão das transações é usar REPEATABLE READ e leituras consistentes. Aplicativos projetados para uso com READ COMMITTED podem precisar ser modificados. Como alternativa, eles podem alterar explicitamente o padrão para READ COMMITTED.</p>

Origem	Destino	Novidades, aprimoramentos ou correções
SQL Server	AWS Glue (Concha Python)	<p>Instruções do SQL Server produzem um conjunto de resultados completo, mas há momentos em que os resultados são melhor processados uma linha de cada vez. Abrir um cursor em um conjunto de resultados permite processar o conjunto de resultados uma linha de cada vez. É possível atribuir um cursor a uma variável ou um parâmetro com um tipo de dados de cursor.</p> <p>Implementação envolvendo uma série de instruções Transact-SQL para código armazenado para que um grupo de instruções Transact-SQL possa ser executado mesmo que o Python não ofereça suporte a BEGIN e END do SQL Server como controle de fluxo.</p> <p>As instruções SQL Server LABEL e GOTO não são suportadas pelo AWS Glue. Se a AWS SCT encontrar um rótulo no código, ele será ignorado. Se a AWS SCT encontrar uma instrução GOTO, ela será comentada.</p>
SQL Server	Amazon Redshift	<p>Implementação do processamento condicional de instruções Transact-SQL para código armazenado implementando o IF... Controle ELSE.</p> <p>Implementação que encerra uma série de instruções Transact-SQL para código armazenado para que um grupo de instruções Transact-SQL possa ser executado como um bloco. Oferece suporte para BEGIN aninhado... Blocos END.</p> <p>Implementação de SET e SELECT em código armazenado.</p> <p>Implementação de CREATE INDEX no Amazon Redshift (que não oferece suporte para índices) criando uma chave de classificação especificada pelo usuário nas tabelas.</p>

Alterações do MySQL versão 1.0.640

A tabela a seguir lista alterações na build 1.0.640 em que o MySQL é o mecanismo de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
MySQL	PostgreSQL L 12.x	Adição de suporte para colunas geradas.
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
MySQL	PostgreSQL L/Aurora PostgreSQL L 11.	<p>Suporte adicionado para o seguinte:</p> <ul style="list-style-type: none"> • Incorporação de transações em procedimentos armazenados SQL. • A capacidade de chamar procedimentos armazenados SQL. • A capacidade de criar procedimentos armazenados SQL.
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.

Alterações da release 1.0.640 do PostgreSQL

A tabela a seguir lista as alterações na build 1.0.640 em que o PostgreSQL é o mecanismo de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
PostgreSQL	MySQL 8.x	<p>Agora o MySQL oferece suporte à criação de partes de chave de índice funcionais que indexam valores de expressão em vez de valores de coluna. Partes de chave funcionais permitem indexar valores que não poderiam ser indexados de outra forma, como valores JSON.</p> <p>O MySQL agora é compatível com CTE e CTE recursivo.</p>
Todos	Todos	<p>Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.</p>
Todos	Todos	<p>Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.</p>
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>Suporte adicionado para o seguinte:</p> <ul style="list-style-type: none"> • Incorporação de transações em procedimentos armazenados SQL. • A capacidade de chamar procedimentos armazenados SQL. • A capacidade de criar procedimentos armazenados SQL.
PostgreSQL	MySQL 8.x	<p>Adição de suporte ao MySQL para índices descendentes. DESC em uma definição de índice não é mais ignorado, mas faz com que o armazenamento de valores de chave fique em ordem decrescente.</p> <p>Adição de suporte do MySQL ao uso de expressões como valores padrão em especificações de tipo de dados, incluindo expressões como valores padrão para os tipos de dados BLOB, TEXT, GEOMETRY e JSON.</p> <p>Agora várias funções agregadas podem ser usadas como funções de janela:</p> <ul style="list-style-type: none"> • AVG()

Origem	Destino	Novidades, aprimoramentos ou correções
		<ul style="list-style-type: none">• BIT_AND()• BIT_OR()• BIT_XOR()• COUNT()• JSON_ARRAYAGG()• JSON_OBJECTAGG()• MAX()• MIN()• STDDEV_POP()• STDDEV()• STD()• STDDEV_SAMP()• SUM()• VAR_POP()• VARIANCE()• VAR_SAMP() <p>O MySQL oferece suporte a funções de janela que, para cada linha de uma consulta, realizam um cálculo usando linhas relacionadas àquela linha.</p> <ul style="list-style-type: none">• CUME_DIST()• DENSE_RANK()• FIRST_VALUE()• LAG()• LAST_VALUE()• LEAD()• NTH_VALUE()• NTILE()

Origem	Destino	Novidades, aprimoramentos ou correções
		<ul style="list-style-type: none"> • PERCENT_RANK() • RANK() • ROW_NUMBER()
PostgreSQL	MySQL 8.x	Adição de suporte para restrição CHECK.
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.
PostgreSQL/Aurora PostgreSQL	Todos	<p>Adição de emulação de sysindexes de visualização do sistema.</p> <p>Se houver uma instrução SELECT em um procedimento sem especificar INTO, o parâmetro INOUT p_refcur do tipo refcursor será criado para um procedimento no destino.</p>

Alterações do release 1.0.640 do Db2 LUW

A tabela a seguir lista as alterações na build 1.0.640 em que o DB2 LUW é o mecanismo de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
DB2 LUW	RDS MariaDB 10.4	Adição de suporte ao RDS MariaDB 10.4 para todos os fornecedores de processamento transacional online (OLTP).

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
DB2 LUW	MySQL 8.0.17	Adição de suporte à restrição CHECK.
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.

Alterações na release 1.0.640 do Teradata

A tabela a seguir lista alterações na build 1.0.640 para outros mecanismos de origem do Teradata.

Origem	Destino	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	<p>Adição de suporte para as instruções MERGE e QUALIFY.</p> <p>Remoção da cláusula LOCKING ROWS FOR ACCESS de instruções Teradata.</p> <p>Adição de suporte para a função CAST.</p>

Origem	Destino	Novidades, aprimoramentos ou correções
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Teradata	Teradata	Implementação de melhorias em REGEXP_INSTR() e REGEXP_SUBSTR().
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
Todos	SQL Server	<p>SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph.</p> <p>SQL Server 2016: adição de suporte para TEMPORAL TABLES.</p>
Teradata	Todos	Adição de suporte para REGEXP_INSTR() e REGEXP_SUBSTR().
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.
Teradata	Amazon Redshift	<p>Implementação da capacidade de salvar SQL da árvore de origem em um único ou em vários arquivos por etapa usando as configurações em Configurações do projeto, Salvar como SQL e Aplicar, lista suspensa: Arquivo único/Vários arquivos.</p> <p>Implementação de melhorias implementadas em visualizações e conversões de procedimentos.</p>
Teradata	Todos	Adição de suporte para Teradata versão 16.20

Alterações da release 1.0.640 para outros mecanismos

A tabela a seguir lista as alterações na build 1.0.640 para outros mecanismos de origem.

Origem	Destino	Novidades, aprimoramentos ou correções
Sybase	RDS MariaDB 10.4	Adição de suporte ao RDS MariaDB 10.4 para todos os fornecedores de processamento transacional online (OLTP).
SAP ASE	MariaDB	Implementação do seguinte: <ul style="list-style-type: none"> • MariaDB 10.4 • instrução EXECUTE IMMEDIATE • Definições DEFAULT • Suporte à restrição CHECK
SAP ASE	PostgreSQL L 12.x	Adição de suporte para colunas geradas.
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte O que é o Amazon Corretto 11? no Guia do usuário do Amazon Corretto 11.
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
SAP ASE	MySQL 8.0.17	Adição de suporte à restrição CHECK.
Todos	SQL Server	SQL Server 2019: adição de suporte para novo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: adição de suporte para tipos de tabela Edge e nó de bancos de dados Graph. SQL Server 2016: adição de suporte para TEMPORAL TABLES.

Origem	Destino	Novidades, aprimoramentos ou correções
Vertica	Amazon Redshift	Adição de suporte para o estilo de distribuição = AUTO.
Todos	Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.
Amazon Redshift	Amazon Redshift	Funções internas não compatíveis em instruções DML são substituídas por NULL como um espaço reservado.
Sybase	PostgreSQL	Adição de suporte para funções nativas.
SAP ASE	MySQL/Aurora MySQL	O nível de isolamento padrão para o Aurora MySQL é REPEATABLE READ. A definição de um nível de isolamento GLOBAL não é compatível com o Aurora MySQL. Somente o escopo da sessão pode ser alterado. O comportamento padrão das transações é usar REPEATABLE READ e leituras consistentes. Aplicativos projetados para serem executados com READ COMMITTED podem precisar ser modificados. Ou você pode alterar explicitamente o padrão para READ COMMITTED.
SAP ASE	PostgreSQL	Adição de suporte para a função CONVERT(optimistic) sem o pacote de extensão.
SAP ASE	Todos	Adição de emulação de sysindexes de visualização do sistema. Se houver uma instrução SELECT em um procedimento sem especificar INTO, o parâmetro INOUT p_refcur do tipo refcursor será criado para um procedimento no destino.
Greenplum	Amazon Redshift	Implementação de CREATE TEMPORARY TABLE da seguinte forma:

Histórico do documentos

A tabela a seguir descreve alterações importantes no guia do usuário da AWS Schema Conversion Tool (AWS SCT) após janeiro de 2018.

Você pode se inscrever em um feed RSS para ser notificado sobre atualizações nessa documentação.

Alteração	Descrição	Data
Compilação #1.0.672 da AWS SCT	A compilação 1.0.672 fornece suporte ao Amazon RDS para PostgreSQL 15 como destino e ao Microsoft SQL Server versão 2022 como origem. Ela também adiciona suporte para novos recursos do Amazon Redshift no código convertido, implementa várias melhorias de conversão para o IBM Db2 para z/OS como origem e resolve vários problemas de conversão.	8 de maio de 2023
Compilação #1.0.671 da AWS SCT	A compilação 1.0.671 fornece suporte às migrações do Apache Oozie para a AWS Step Functions. Ela também adiciona suporte ao BigQuery como origem para o processo de avaliação de vários servidores. Além disso, ela adiciona novas configurações de conversão para o IBM Db2 para z/OS como origem e	8 de março de 2023

resolve vários problemas de conversão.

[Compilação #1.0.670 da AWS SCT](#)

A compilação 1.0.670 fornece suporte às migrações do Hadoop para o Amazon EMR. Ela também adiciona suporte ao Azure Synapse Analytics como origem para o processo de avaliação de vários servidores. Além disso, ela melhora a conversão do código SQL incorpora do em aplicativos Java e resolve vários problemas de conversão.

23 de janeiro de 2023

[Compilação #1.0.669 da AWS SCT](#)

A compilação 1.0.669 implementa o suporte de particionamento nativo para migração de dados de data warehouses Oracle. Ela também melhora o processo de avaliação de vários servidores, adiciona novos atributos nos agentes de extração de dados e resolve vários problemas de conversão.

19 de dezembro de 2022

[Compilação #1.0.668 da AWS SCT](#)

A compilação 1.0.668 implementa o particionamento virtual automático para migração de dados dos bancos de dados Greenplum e adiciona suporte à migração de dados dos bancos de dados Snowflake para o Amazon Redshift. Ela também melhora a conversão do código SQL incorporado em aplicativos C# e resolve vários problemas de conversão.

16 de novembro de 2022

[Compilação #1.0.667 da AWS SCT](#)

A compilação 1.0.667 fornece suporte para o mecanismo de extração, transformação e carregamento (ETL) da Informatica como origem de migração. Ela também atualiza a versão do pacote de extensões, aumenta a versão mínima do driver compatível com o Amazon Redshift e resolve vários problemas de conversão.

13 de outubro de 2022

[Compilação #1.0.666 da AWS SCT](#)

A compilação 1.0.666 melhora a conversão de aplicativos Java adicionando suporte à estrutura MyBatis. Ela também adiciona novas funções aos pacotes de extensão, aprimora o carregador de metadados de origem e resolve vários problemas de conversão.

20 de setembro de 2022

[Compilação #1.0.665 da AWS SCT](#)

A compilação 1.0.665 oferece suporte ao BigQuery como origem de migração. Ela também implementa o suporte da nova versão do arquivo de configuração de atributos do Babelfish. Além disso, ela melhora a conversão de data warehouses para o Amazon Redshift e resolve vários problemas de conversão.

29 de agosto de 2022

[Compilação #1.0.664 da AWS SCT](#)

A compilação 1.0.664 fornece suporte ao Amazon Redshift sem servidor como origem ou destino de migração. Ela também implementa o balanceamento automático de memória nas tarefas de extração de dados e corrige um erro em que a AWS SCT não conseguiu se conectar aos dispositivos AWS Snowball. Além disso, ela adiciona a capacidade de alterar o agrupamento de colunas nas regras de migração, melhora a interface do usuário e resolve vários problemas de conversão.

14 de julho de 2022

[Compilação #1.0.663 da AWS SCT](#)

A compilação 1.0.663 adiciona suporte ao Babelfish para Aurora PostgreSQL 1.2.0 e melhora os recursos do relatório de avaliação de vários servidores. Ela também adiciona novos atributos às regras de migração, corrige dois erros do carregador e resolve vários problemas de conversão.

20 de junho de 2022

[Compilação #1.0.662 da AWS SCT](#)

A compilação 1.0.662 implementa a conversão de código SQL em aplicativos C# e melhora o fluxo de trabalho do relatório de avaliação de vários servidores. Ela também adiciona várias melhorias de conversão e resolve vários problemas de conversão.

19 de maio de 2022

[Compilação #1.0.661 da AWS SCT](#)

A compilação 1.0.661 fornece suporte ao IBM Db2 para z/OS como origem de migração. Ela também adiciona suporte para conversão de scripts de extração, transformação e carregamento (ETL) para AWS Glue Studio e resolve vários problemas de conversão.

21 de abril de 2022

[Compilação #1.0.660 da AWS SCT](#)

A compilação 1.0.660 fornece suporte à versão principal do PostgreSQL 14 e ao MariaDB 10.6 como destinos de migração. Ela também adiciona suporte à conversão de índices Oracle para visualizações materializadas e resolve vários problemas de conversão.

21 de março de 2022

[Compilação #1.0.659 da AWS SCT](#)

A compilação 1.0.659 fornece suporte à versão principal do PostgreSQL 13 na edição compatível com o Aurora PostgreSQL como destino de migração. Ela implementa a conversão de código SQL em aplicativos C#, adiciona suporte ao Oracle Unified Auditing e resolve vários problemas de conversão.

21 de fevereiro de 2022

[Compilação #1.0.658 da AWS SCT](#)

A compilação 1.0.658 fornece integração com a AWS Secrets Manager e adiciona suporte à plataforma de banco de dados de destino virtual do Amazon Redshift. Ela também adiciona várias melhorias de conversão e correções de bugs.

20 de janeiro de 2022

Compilação #1.0.657 da AWS SCT	A compilação 1.0.657 melhora a conversão do Microsoft SQL Server para Aurora Edição compatível com PostgreSQL, Amazon RDS para PostgreSQL e outros destinos de migração. Ela também adiciona várias melhorias na interface do usuário e correções de bugs.	20 de dezembro de 2021
Compilação #1.0.656 da AWS SCT	A compilação 1.0.656 fornece suporte a vários bancos de dados de origem e destino em um projeto. Ela também adiciona conversão, estratégia de otimização, melhorias gerais e várias correções de bugs.	22 de novembro de 2021
Compilação #1.0.655 da AWS SCT	A compilação 1.0.655 implementa a conversão dos scripts de trabalho do Teradata FastExport para o Amazon Redshift RSQL e aumenta a versão mínima do driver compatível com o Greenplum para 42.2.5. Ela também adiciona uma série de melhorias e correções de bugs.	18 de outubro de 2021

[Compilação #1.0.654 da AWS SCT](#)

A compilação 1.0.654 implementa a conversão dos scripts Shell, Teradata FastLoad e Teradata Basic Teradata Query (BTEQ) para o Amazon Redshift RSQL. Ela também resolve vários problemas de conversão e adiciona várias melhorias e correções de bugs.

16 de setembro de 2021

[Compilação #1.0.653 da AWS SCT](#)

A compilação 1.0.653 implementa a conversão de SQL dinâmico criado em funções ou procedimentos chamados. Ela também melhora a conversão de rotinas criptografadas e adiciona várias melhorias e correções de bugs.

10 de agosto de 2021

[Compilação #1.0.652 da AWS SCT](#)

A compilação 1.0.652 implementa o modo de comando de script na interface de linha de comando e implementa regras de otimização de esquema. Ela também adiciona várias melhorias de conversão e desempenho e correções de bugs.

30 de junho de 2021

Compilação #1.0.651 da AWS SCT	A compilação 1.0.651 adiciona várias melhorias e correções de bugs. Ela também fornece acesso à cópia inicial da referência da CLI da AWS Schema Conversion Tool.	04 de junho de 2021
Compilação #1.0.650 da AWS SCT	A compilação 1.0.650 implementa o suporte do Amazon RDS para PostgreSQL 13 como um banco de dados de destino e atualiza os agentes extratores. Ela também atualiza a versão mínima compatível do driver JDBC para Microsoft SQL Server, Azure e Azure Synapse. Além disso, ela adiciona várias melhorias de conversão e correções de bugs.	30 de abril de 2021
Compilação #1.0.649 da AWS SCT	A compilação 1.0.649 implementa o suporte ao MariaDB 10.5 como banco de dados de destino e implementa aprimoramentos de funções para conversão de funções integradas do Oracle. Ela também adiciona várias melhorias de conversão e desempenho e correções de bugs.	29 de março de 2021
Compilação #1.0.648 da AWS SCT	A compilação 1.0.648 adiciona várias melhorias de conversão e correções de bugs.	22 de fevereiro de 2021

[Compilação #1.0.647 da AWS SCT](#)

A compilação 1.0.647 adiciona suporte ao atributo Database Mail no Amazon RDS, implementa o carregamento e a conversão de comentários em objetos de armazenamento. Ela também adiciona o Data Migration Service Assessor da AWS SCT e o Wizard da AWS SCT e implementa a interface de usuário do filtro de árvore. Além disso, ela adiciona uma seção redesenhada no Relatório de Avaliação e várias melhorias e correções de bugs.

[Compilação #1.0.646 da AWS SCT](#)

A compilação 1.0.646 adiciona suporte para tipos de dados INTERVAL, colunas de identidade e conversão de cursores, além de adicionar várias melhorias e correções de bugs.

[Compilação #1.0.645 da AWS SCT](#)

A compilação 1.0.645 adiciona suporte para ETL SSIS para conversão AWS Glue e várias melhorias e correções de bugs.

[Compilações #1.0.643-1.0.644 da AWS SCT](#)

A compilação 1.0.644 adiciona várias melhorias de conversão, desempenho e interface de usuário e correções de bugs.

Compilação #1.0.642 da AWS SCT	A compilação 1.0.642 implementa a conversão de pacotes ETL do Microsoft SQL Server Integration Services para AWS Glue e adiciona várias melhorias e correções de bugs.	28 de agosto de 2020
Compilação #1.0.641 da AWS SCT	Foi adicionado suporte SSL para extratores de dados. A compilação também inclui várias melhorias e correções.	17 de julho de 2020
Compilações #1.0.633-1.0.640 da AWS SCT	Atualização do JDK 8 para o Amazon Corretto JDK 11. Adição de tabelas que identificam outras atualizações, alterações e correções.	22 de junho de 2020
Disponibilidade do AWS WQF	A AWS SCT não está mais fornecendo a ferramenta AWS Workload Qualification Framework (AWS WQF) para download.	19 de junho de 2020
Compilação #1.0.632 da AWS SCT	UI da SCT - adicionada nova guia para mostrar os erros que ocorrem ao aplicar scripts. Agora você pode salvar a árvore de origem como SQL ao converter do SAP ASE. Melhorias para conversões para PostgreSQL ou Aurora PostgreSQL ou Redshift.	19 de novembro de 2019

[Compilações #1.0.631 e #1.0.630 da AWS SCT \(combinadas\)](#)

Melhor suporte a ROWIDs no Oracle e para objetos do sistema no Microsoft SQL Server e no SAP ASE. Melhor tratamento para especificadores ausentes de esquemas do SQL Server. Melhor suporte para conversões de Greenplum para Redshift. Suporte aprimorado para a conversão de código armazenado ao migrar para Amazon Redshift, MariaDB, MySQL e PostgreSQL.

30 de setembro de 2019

[Compilação #1.0.629 da AWS SCT](#)

Suporte a procedimentos armazenados para conversões do Netezza. Suporte aprimorado a conversões para Amazon Redshift, DynamoDB, MySQL e PostgreSQL. Adicionado suporte para SAP ASE 12.5 como origem

20 de agosto de 2019

[Compilação #1.0.628 da AWS SCT](#)

Suporte à emulação de serviço para conversões de DB2, SQL Server e Oracle. Melhorias em conversões para o Amazon Redshift, incluindo mais suporte para cursores e procedimentos armazenados.

22 de junho de 2019

Compilação #1.0.627 da AWS SCT	Suporte para conversões do SQL Server para procedimentos armazenados no Amazon Redshift. Melhorias da conversão para o PostgreSQL 11 e MySQL 8.0.	31 de maio de 2019
Compilação #1.0.626 da AWS SCT	O PostgreSQL 11 e o MySQL 8.0 agora são compatíveis como destino. O SAP ASE 15.5 agora é compatível como origem.	26 de abril de 2019
Compilação #1.0.625 da AWS SCT	As atualizações incluem a capacidade de converter Teradata BTEQ para AWS Glue, suporte para conversões para MariaDB 10.3 com suporte ao modo de compatibilidade com Oracle, suporte para SAP ASE 15.7 e substituições de serviço para emular a funcionalidade ausente.	25 de março de 2019

[Compilação #1.0.624 da AWS SCT](#)

As atualizações incluem a capacidade de converter o Oracle ETL para AWS Glue e suporte para conversões do Microsoft SQL Server, Oracle e IBM Db2 LUW para o Amazon RDS para MariaDB. Também adicionamos suporte para conversões do SAP ASE para RDS para MySQL e Amazon Aurora com compatibilidade com MySQL. Além disso, adicionamos suporte para a extensão Orafce durante a conversão de Oracle para PostgreSQL.

22 de fevereiro de 2019

[Compilação #1.0.623 da AWS SCT](#)

As atualizações incluem a capacidade de converter bancos de dados do SAP ASE e a capacidade de converter scripts T-SQL, DML e DDL para componentes ou código equivalente. Também adicionamos emulações do Oracle e Microsoft SQL Server para melhorar as conversões.

25 de janeiro de 2019

[Compilação #1.0.622 da AWS SCT](#)

As atualizações incluem o Workload Qualification Framework, que analisa a carga de trabalho para uma migração completa, incluindo modificações de aplicativo e banco de dados.

20 de dezembro de 2018

Compilação #1.0.621 da AWS SCT	As atualizações incluem suporte ao Aurora PostgreSQL 10 como um destino e a capacidade de migrar do Netezza usando as opções de tabela externa.	21 de novembro de 2018
Compilação #1.0.620 da AWS SCT	As atualizações incluem a capacidade de salvar scripts SQL e suporte para cursores globais do Oracle ao migrar para o MySQL.	22 de outubro de 2018
Compilação #1.0.619 da AWS SCT	As atualizações incluem suporte para a migração do Apache Cassandra ao DynamoDB e suporte para Vertica 9 como origem.	20 de setembro de 2018
Compilação #1.0.618 da AWS SCT	As atualizações incluem relatórios de avaliação expandidos, suporte para converter o Oracle ROWIDs e suporte para tabelas do SQL Server definidas pelo usuário.	24 de agosto de 2018
Compilação #1.0.617 da AWS SCT	As atualizações incluem relatórios de avaliação expandidos, suporte para converter o Oracle ROWIDs e suporte para tabelas do SQL Server definidas pelo usuário.	24 de julho de 2018

[Compilação #1.0.616 da AWS SCT](#)

As atualizações incluem suporte para RDS ao fazer a conversão de Oracle em Amazon RDS for Oracle, a conversão de objetos de programação Oracle e suporte para trabalhos do Oracle, particionamento e Db2 LUW versão 10.1.

26 de junho de 2018

[Compilação #1.0.615 da AWS SCT](#)

As atualizações incluem suporte para SQL Server para instruções do PostgreSQL GOTO, particionamento do PostgreSQL 10 e Db2 LUW versão 10.1.

24 de maio de 2018

[Compilação #1.0.614 da AWS SCT](#)

As atualizações incluem suporte para Oracle para links de banco de dados Oracle, SQL Server para funções em linha do PostgreSQL e emulação de objetos do sistema Oracle.

25 de abril de 2018

[Compilação #1.0.613 da AWS SCT](#)

As atualizações incluem suporte para Db2 LUW, conversão de arquivos SQL*Plus e SQL Server Windows Authentication.

28 de março de 2018

[Compilação #1.0.612 da AWS SCT](#)

As atualizações incluem suporte para o mapeamento de tipo de dados personalizado, comparação de esquemas para Oracle 10 e conversão do Oracle em PostgreSQL de variáveis globais.

22 de fevereiro de 2018

[Compilação #1.0.611 da AWS SCT](#)

As atualizações incluem suporte para instruções dinâmicas do Oracle em PostgreSQL, abertura do arquivo de log selecionando uma mensagem de erro e a capacidade de ocultar esquemas na visualização em árvore.

23 de janeiro de 2018

Atualizações anteriores

A tabela a seguir descreve alterações importantes no guia do usuário da AWS Schema Conversion Tool (AWS SCT) antes de janeiro de 2018.

Versão	Alteração	Descrição	Alterado em
1.0.608	Suporte a endpoint do FIPS para Amazon S3	Agora você pode solicitar que a AWS SCT se conecte ao Amazon S3 e ao Amazon Redshift usando os endpoints do FIPS para ficar em conformidade com os requisitos de segurança do Federal Information Processing Standard (FIPS). Para obter mais informações, consulte Armazenando credenciais da AWS .	17 de novembro de 2017

Versão	Alteração	Descrição	Alterado em
1.0.607	Suporte a endpoint do FIPS para Amazon S3	Agora você pode solicitar que a AWS SCT se conecte ao Amazon S3 e ao Amazon Redshift usando os endpoints do FIPS para ficar em conformidade com os requisitos de segurança do Federal Information Processing Standard (FIPS). Para obter mais informações, consulte Armazenando credenciais da AWS .	30 de outubro de 2017
1.0.607	As tarefas de extração de dados podem ignorar os LOBs	Quando cria tarefas de extração de dados, agora você pode optar por ignorar objetos grandes (LOBs) para reduzir a quantidade e de dados extraídos. Para obter mais informações, consulte Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados .	30 de outubro de 2017
1.0.605	Acesso ao log de tarefas do agente de extração de dados	Agora, você pode acessar o log de tarefas do agente de extração de dados de um link conveniente na interface do usuário da AWS Schema Conversion Tool. Para obter mais informações, consulte Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados .	28 de agosto de 2017
1.0.604	Aprimoramentos do Converter	O mecanismo da AWS Schema Conversion Tool foi aprimorado para oferecer melhores conversões para migrações heterogêneas.	24 de junho de 2017
1.0.603	Filtros de suporte de agentes de extração de dados	Agora, você pode filtrar os dados que os agentes de extração extraem de seu data warehouse. Para obter mais informações, consulte Criação de regras de migração de dados no AWS SCT .	16 de junho de 2017

Versão	Alteração	Descrição	Alterado em
1.0.603	A AWS SCT oferece suporte a versões de data warehouse adicionais	Agora, você pode usar a AWS Schema Conversion Tool para converter os esquemas do Teradata 13 e do data warehouse do Oracle 10 para esquemas equivalentes do Amazon Redshift. Para obter mais informações, consulte Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT .	16 de junho de 2017
1.0.602	Os agentes de extração de dados dão suporte a data warehouses adicionais	Agora, você pode usar agentes de extração de dados para extrair dados de seus data warehouses do Microsoft SQL Server. Para obter mais informações, consulte Como migrar dados de um data warehouse on-premises para o Amazon Redshift .	11 de maio de 2017
1.0.602	Agentes de extração de dados podem copiar dados para o Amazon Redshift	Os agentes de extração de dados agora têm três modos de upload. Agora, você pode especificar se deseja apenas extrair os dados, extrair os dados e apenas fazer upload deles para o Amazon S3 ou extrair, fazer upload e copiar os dados diretamente para o Amazon Redshift. Para obter mais informações, consulte Criação, execução e monitoramento de uma tarefa AWS SCT de extração de dados .	11 de maio de 2017
1.0.601	A AWS SCT oferece suporte a data warehouses adicionais	Agora, você pode usar a AWS Schema Conversion Tool para converter esquemas do Vertica e do Microsoft SQL Server em esquemas equivalentes do Amazon Redshift. Para obter mais informações, consulte Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT .	18 de abril de 2017

Versão	Alteração	Descrição	Alterado em
1.0.601	Os agentes de extração de dados dão suporte a data warehouses adicionais	Agora, você pode usar agentes de extração de dados para extrair dados de seus data warehouses do Greenplum, Netezza e Vertica. Para obter mais informações, consulte Como migrar dados de um data warehouse on-premises para o Amazon Redshift .	18 de abril de 2017
1.0.601	Os agentes de extração de dados dão suporte a sistemas operacionais adicionais	Agora, você pode instalar agentes de extração de dados em computadores que executam os sistemas operacionais Microsoft Windows e macOS. Para obter mais informações, consulte Como instalar atendentes de extração .	18 de abril de 2017
1.0.601	Os agentes de extração de dados fazem upload automaticamente para o Amazon S3	Os agentes de extração de dados agora fazem upload de seus dados extraídos automaticamente para o Amazon S3. Para obter mais informações, consulte Saída da tarefa de extração de dados .	18 de abril de 2017
1.0.600	Agentes de extração de dados	Agora, você pode instalar agentes de extração de dados que extraem os dados do seu data warehouse e os prepara para uso com o Amazon Redshift. Você pode usar a AWS Schema Conversion Tool para registrar os agentes e criar tarefas de extração de dados para eles. Para obter mais informações, consulte Como migrar dados de um data warehouse on-premises para o Amazon Redshift .	16 de fevereiro de 2017

Versão	Alteração	Descrição	Alterado em
1.0.600	Comentários do cliente	Agora, você pode enviar comentários sobre a AWS Schema Conversion Tool. Você pode registrar um relatório de bugs, enviar uma solicitação de recurso ou fornecer informações gerais. Para obter mais informações, consulte Fornecendo feedback .	16 de fevereiro de 2017
1.0.502	Integração com AWS DMS	Agora, você pode usar a AWS Schema Conversion Tool para criar endpoints e tarefas do AWS DMS. Você pode executar e monitorar as tarefas na AWS SCT. Para obter mais informações, consulte Uso do AWS SCT com a AWS DMS .	20 de dezembro de 2016
1.0.502	Compatibilidade do Amazon Aurora com PostgreSQL como um banco de dados de destino	Agora, a AWS Schema Conversion Tool oferece suporte ao Amazon Aurora com compatibilidade do PostgreSQL como um banco de dados de destino. Para obter mais informações, consulte Como converter esquemas de bancos de dados usando a AWS SCT .	20 de dezembro de 2016
1.0.502	Suporte para perfis	Agora, você pode armazenar diferentes perfis na AWS Schema Conversion Tool e facilmente alternar entre eles. Para obter mais informações, consulte Armazenando perfis de serviço da AWS na AWS SCT .	20 de dezembro de 2016

Versão	Alteração	Descrição	Alterado em
1.0.501	Suporte para Greenplum Database e Netezza	Agora, você pode usar a AWS Schema Conversion Tool para converter seus esquemas de data warehouse do banco de dados Greenplum e Netezza para o Amazon Redshift. Para obter mais informações, consulte Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT .	17 de novembro de 2016
1.0.501	Otimização do Redshift	Agora, você pode usar a AWS Schema Conversion Tool para otimizar seus bancos de dados do Amazon Redshift. Para obter mais informações, consulte Como otimizar o Amazon Redshift usando a AWS SCT .	17 de novembro de 2016
1.0.500	Regras de mapeamento	Antes de converter seu esquema com a AWS Schema Conversion Tool, agora você pode configurar regras que alteram o tipo de dados de colunas, movem objetos de um esquema para outro e alteram os nomes de objetos. Para obter mais informações, consulte Como criar regras de migração na AWS SCT .	4 de outubro de 2016
1.0.500	Mover para a nuvem	Você agora pode usar a AWS Schema Conversion Tool para copiar o esquema de banco de dados on-premises existente para uma instância de banco de dados do Amazon RDS que esteja executando o mesmo mecanismo. Você pode usar esse recurso para analisar possíveis economias de custo ao mudar para a nuvem e ao alterar o tipo de licença. Para obter mais informações, consulte Como criar relatórios de avaliação de migração com a AWS SCT .	4 de outubro de 2016

Versão	Alteração	Descrição	Alterado em
1.0.400	Conversões de esquema de data warehouse	Agora, você pode usar a AWS Schema Conversion Tool para converter seus esquemas de data warehouse de Oracle e Teradata para Amazon Redshift. Para obter mais informações, consulte Como converter esquemas de data warehouse em Amazon Redshift usando a AWS SCT .	13 de julho de 2016
1.0.400	Conversões de SQL do aplicativo	Agora, você pode usar a AWS Schema Conversion Tool para converter SQL em C++, C#, Java ou outro código de aplicativo. Para obter mais informações, consulte Como converter o aplicativo SQL usando a AWS SCT .	13 de julho de 2016
1.0.400	Novo atributo	Agora, a AWS Schema Conversion Tool contém um pacote de extensões e um assistente para ajudá-lo a instalar, criar e configurar funções do AWS Lambda e bibliotecas Python para oferecer e-mail, programação de trabalhos e outros recursos. Para obter mais informações, consulte Usando as AWS Lambda funções do pacote AWS SCT de extensão e Como usar bibliotecas personalizadas para pacotes de extensão da AWS SCT .	13 de julho de 2016
1.0.301	SSL Support	Agora, você pode usar o Secure Sockets Layer (SSL) para se conectar ao banco de dados de origem ao usar a AWS Schema Conversion Tool.	19 de maio de 2016
1.0.203	Novo atributo	Adiciona suporte para MySQL e PostgreSQL como bancos de dados de origem para conversões.	11 de abril de 2016

Versão	Alteração	Descrição	Alterado em
1.0.202	Versão de manutenção	Adiciona suporte para a edição de SQL convertido que seja gerado para o mecanismo de banco de dados de destino. Adiciona recursos de seleção aprimorada no banco de dados de origem e visualizações em árvore da instância de banco de dados de destino. Adiciona suporte para a conexão com um banco de dados de origem Oracle por meio de nomes Transparent Network Substrate (TNS).	2 de março de 2016
1.0.200	Versão de manutenção	Adiciona suporte para PostgreSQL como um mecanismo de banco de dados de destino. Adiciona a capacidade de gerar esquema convertido como scripts e salvá-los em arquivos antes de aplicar o esquema à instância de banco de dados de destino.	14 de janeiro de 2016
1.0.103	Versão de manutenção	Adiciona recursos de projeto off-line, a capacidade de verificação de novas versões e o gerenciamento da memória e do desempenho.	2 de dezembro de 2015
1.0.101	Versão de manutenção	Adiciona o assistente Criar novo projeto de migração de banco de dados. Adiciona a capacidade de salvar o relatório de avaliação de migração de banco de dados como um arquivo PDF.	19 de outubro de 2015
1.0.100	Versão de visualização	Fornece o guia do usuário da versão de pré-visualização da AWS Schema Conversion Tool.	7 de outubro de 2015

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.