

Guia do usuário

AWS Schema Conversion Tool



Versão 1.0.672

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Schema Conversion Tool: Guia do usuário

Copyright © 2025 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 AWS SCT	1
Visão geral da conversão de esquema	5
Dando feedback	7
Instalando e configurando AWS SCT	8
Instalando AWS SCT	8
Validar instalação	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
Instalação de drivers JDBC	13
Instalação de drivers JDBC no Linux	16
Armazenamento de caminhos do driver nas configurações globais	17
Atualizando AWS SCT	18
AWS Schema Conversion Tool CLI	19
AWS SCT interface de usuário	20
Janela do projeto	20 22
Como iniciar e gerenciar projetos	
Usar o assistente do	23
Salvar projetos	26
Adicionar servidores de banco de dados	27
Modo off-line	28
Filtros de árvore	29
	30
Importar uma lista de arquivos para o filtro de árvore	32
Ocultando esquemas	33
Relatório de avaliação da	35
Conversão de esquemas	39
Como aplicar esquemas convertidos	42
Como gerenciar perfis	43
Armazenamento de AWS credenciais	44
Configurando o perfil padrão de um projeto	46
Permissões para usar o perfil AWS de serviço	46
Como configurar o Secrets Manager	47

	Armazenando senhas de banco de dados	48
	Visualização UNION ALL	48
	Atalhos de teclado	49
C	onceitos básicos	51
C	omo se conectar ao banco de dados de origem	53
	Como se conectar ao Amazon RDS e ao Aurora criptografados	54
	Como se conectar ao Apache Cassandra	57
	Como se conectar ao Apache Cassandra como origem	. 57
	Como se conectar ao Apache Hadoop	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 se conectar ao Apache Oozie	70
	Pré-requisitos	70
	Como se conectar ao Apache Oozie como origem	71
	Permissões para AWS Lambda	72
	Conectando-se AWS Step Functions como alvo	74
	Conectar ao Azure SQL	76
	Privilégios do banco de dados Azure SQL	76
	Como se conectar ao banco de dados Azure SQL de origem	
	Conectando-se à IBM DB2 para z/OS	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	
	Privilégios do MySQL como um destino	82
	Privilégios do PostgreSQL como destino	
	Configurações de conversão do Db2 para z/OS para PostgreSQL	
	Bancos de dados do IBM Db2 LUW	
	Privilégios do Db2 LUW	
	Como se conectar ao Db2 LUW como origem	
	Db2 LUW para PostgreSQL	92
	Db2 LUW para MvSQL	94

Usar o MySQL como origem	96
Privilégios do MySQL	96
Conectar-se ao MySQL como origem	97
Privilégios do PostgreSQL como destino	100
Bancos de dados Oracle	100
Privilégios do Oracle	101
Conectar-se ao Oracle como origem	102
Oracle para PostgreSQL	106
Oracle para MySQL	112
Oracle para Amazon RDS para Oracle	122
bancos de dados do PostgreSQL	129
Privilégios para o PostgreSQL	130
Conectar-se ao PostgreSQL como origem	130
Privilégios do MySQL como um destino	133
Bancos de dados SAP	134
Privilégios do SAP ASE	135
Conectando-se ao SAP ASE como origem	136
Privilégios do MySQL como um destino	138
Configurações de conversão do SAP ASE para MySQL	139
Privilégios do PostgreSQL como destino	140
Configurações de conversão de SAP ASE para PostgreSQL	141
bancos de dados do SQL Server	142
Privilégios do Microsoft SQL Server	143
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	
SQL Server para Amazon RDS SQL Server	190
Data warehouses	192
Amazon Redshift	192
Azure Synapse Analytics como origem	199
BigQuery como fonte	204
bancos de dados Greenplum	210
bancos de dados do Netezza	
Data warehouse do Oracle	226
Snowflake	234

Data warehouses do SQL Server	243
Bancos de dados do Teradata	250
Bancos de dados do Vertica	266
Mapeamento de tipo de dados	273
Mapeamento de novos tipos de dados	274
Edição de mapeamentos de tipos de dados	274
Mapeamento de alvos virtuais	276
Limitações do mapeamento de tipo de dados	277
Relatórios	278
Relatório de avaliação da	278
Criar relatório de avaliação	279
Visualizar o relatório de avaliação	280
Como salvar o relatório de avaliação	285
Como configurar o relatório de avaliação	287
Relatório de avaliação multisservidor	291
Conversão de esquemas	301
Aplicação de regras de migração	303
Como criar regras de migração	304
Como exportar regras de migração	306
Conversão de esquemas manualmente	306
Conversão de esquemas	307
Como editar o esquema convertido	310
Como limpar um esquema convertido	311
Conversão de esquemas manualmente	311
Como modificar seu esquema de origem	311
Como modificar seu esquema de destino	311
Como atualizar e recarregar esquemas	312
Como salvar e aplicar esquemas convertidos	313
Como salvar o esquema convertido	313
Como aplicar seu esquema convertido	314
O esquema do pacote de extensão	315
Como comparar esquemas	315
Como visualizar objetos transformados relacionados	317
Conversão de esquemas de data warehouse	318
Permissões para o Amazon Redshift	319
Como escolher estratégias e regras de otimização	321

	Como coletar ou carregar as estatísticas	323
	Como criar regras de migração	324
	Como criar regras de migração	325
	Como exportar regras de migração	327
	Como converter seu esquema	327
	Conversão de esquemas	327
	Como editar o esquema convertido	330
	Como limpar um esquema convertido	331
	Como gerenciar e personalizar chaves	331
	Tópicos relacionados	332
	Como criar e usar o relatório de avaliação	332
	Como criar um relatório de avaliação de migração do banco de dados	332
	Resumo	333
	Itens de ação	335
	Como salvar o relatório de avaliação	335
	Como lidar com conversões manuais	336
	Como modificar seu esquema de origem	337
	Como modificar seu esquema de destino	337
	Como atualizar e recarregar o esquema convertido	338
	Como salvar e aplicar seu esquema convertido	338
	Como salvar seu esquema convertido em um arquivo	339
	Como aplicar seu esquema convertido	339
	O esquema do pacote de extensão	340
	Bibliotecas Python	
	Conversão de dados do Amazon Redshift	
	Como otimizar seu banco de dados do Amazon Redshift	341
C	onversão de dados usando ETL	344
	Processos de ETL	345
	Pré-requisitos	346
	AWS Glue Catálogo de dados	347
	Limitações	
	Etapa 1: criar um novo projeto	349
	Etapa 2: criar um AWS Glue trabalho	350
	Processos de ETL usando o Python	351
	Etapa 1: criar um banco de dados	352
	Etapa 2: criar uma conexão	353

Etapa 3: criar um crawler do AWS Glue	354
Scripts de ETL da Informatica	356
Pacotes do SSIS	361
Componentes do SSIS compatíveis	365
SSIS para AWS Glue Studio	367
Pré-requisitos	368
Adicionando pacotes SSIS ao seu projeto AWS SCT	370
Como converter pacotes SSIS	371
Criação de AWS Glue Studio empregos	371
Como criar um relatório de avaliação de conversão do SSIS	373
Componentes do SSIS compatíveis	374
Scripts do Teradata BTEQ	375
Adicionando scripts BTEQ ao seu projeto AWS SCT	377
Como configurar variáveis de substituição em scripts BTEQ	378
Conversão de scripts BTEQ	378
Como gerenciar scripts BTEQ	379
Como criar um relatório de avaliação de conversão de script BTEQ	380
Como editar e salvar seus scripts BTEQ convertidos	381
Scripts de shell	381
Adicionar scripts de shell ao seu AWS SCT projeto	382
Como configurar variáveis de substituição em scripts de shell	383
Como converter scripts de shell	384
Como gerenciar scripts de shell	384
Como criar um relatório de avaliação de conversão de script de shell	385
Como editar e salvar seus scripts de shell convertidos	386
FastExport scripts	386
Adicionar scripts de FastExport trabalho ao seu AWS SCT projeto	387
Configurando variáveis de substituição em FastExport scripts de trabalho	388
Convertendo scripts de FastExport trabalho	389
Gerenciando scripts FastExport de trabalho	390
Criação de um relatório de avaliação de conversão do script de FastExport trabalho	391
Editando e salvando seus scripts de FastExport trabalho convertidos	392
FastLoad scripts	392
Adicionar scripts de FastLoad trabalho ao seu AWS SCT projeto	393
Configurando variáveis de substituição em FastLoad scripts de trabalho	394
Convertendo scripts de FastLoad trabalho	395

Gerenciando scripts FastLoad de trabalho	396
Criação de um relatório de avaliação de conversão do script de FastLoad trabalho	397
Editando e salvando seus scripts de FastLoad trabalho convertidos	398
MultiLoad scripts	398
Adicionar scripts de MultiLoad trabalho ao seu AWS SCT projeto	399
Configurando variáveis de substituição em MultiLoad scripts de trabalho	400
Convertendo scripts de MultiLoad trabalho	401
Gerenciando scripts MultiLoad de trabalho	
Criação de um relatório de avaliação de conversão do script de MultiLoad trabalho	
Editando e salvando seus scripts de MultiLoad trabalho convertidos	
Como migrar estruturas de big data	405
Migração de workloads do Hadoop	405
Visão geral	406
Etapa 1: conectar-se aos clusters do Hadoop	407
Etapa 2: configurar as regras de mapeamento	407
Etapa 3: criar um relatório de avaliação	409
Etapa 4: migrar seu cluster do Apache Hadoop para o Amazon EMR	410
Como executar seu script de CLI	411
Como gerenciar projeto de migração	412
Conversão de fluxos de trabalho do Oozie;	414
Visão geral	414
Etapa 1: conectar-se aos seus serviços de origem e destino	416
Etapa 2: configurar as regras de mapeamento	416
Etapa 3: configurar parâmetros	417
Etapa 4: criar um relatório de avaliação	419
Etapa 5: converter seus fluxos de trabalho do Apache Oozie em AWS Step Functions	420
Como executar seu script de CLI	423
Nós suportados	423
Integrando com AWS DMS	425
Usando um agente de AWS SCT replicação com AWS DMS	425
Usando um agente AWS SCT de extração de dados com AWS DMS	425
Aumentar os níveis de registro ao usar AWS SCT com AWS DMS	425
Migrar de um data warehouse	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	
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 da	dos 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 Edge	453
Step-by-step procedimentos para migrar dados usando um AWS SCT E	dge AWS Snowball
Edge	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
Trabalhando com LOBs	464
Melhores práticas e solução de problemas	465
Como converter o aplicativo SQL	467
Visão geral da conversão de aplicativos SQL	467
Código SQL	468
Como criar projetos genéricos de conversão de aplicativos	468
Como gerenciar projetos de conversão de aplicativos	472
Como analisar e converter seu código SQL	473
Como criar e usar o relatório de avaliação	474
Como editar e gravar seu código SQL convertido	475
Código SQL em aplicativos C#	476
Como criar projetos de conversão de aplicativos C#	476
Como converter o código SQL em seu aplicativo C#	478

Como salvar o código do aplicativo convertido	479
Como gerenciar projetos de conversão de aplicativos C#	480
Como criar um relatório de avaliação de conversão de aplicativos C#	481
Código SQL em C++	482
Como criar projetos de conversão de aplicativos C++	482
Como converter o código SQL em seu aplicativo C++	484
Como salvar o código do aplicativo convertido	485
Como gerenciar projetos de conversão de aplicativos C++	486
Como criar um relatório de avaliação de conversão de aplicativos C++	487
Código SQL em Java	488
Como criar projetos de conversão de aplicativos Java	489
Como converter o código SQL em seu aplicativo Java	491
Como salvar o código do aplicativo convertido	493
Como gerenciar projetos de conversão de aplicativos Java	493
Como criar um relatório de avaliação de conversão de aplicativos Java	494
Código SQL no Pro*C	496
Como criar projetos de conversão de aplicativos Pro*C	496
Como converter o código SQL em seu aplicativo Pro*C	497
Como editar e salvar o código do aplicativo convertido	499
Como gerenciar projetos de conversão de aplicativos Pro*C	500
Como criar um relatório de avaliação de conversão de aplicativos Pro*C	501
Pacotes de extensão	503
Permissões para usar o pacote de extensão	505
Como usar o esquema do pacote de extensão	506
Bibliotecas personalizadas para pacotes de extensão	506
Como aplicar o pacote de extensão	507
Usando as funções do Lambda do pacote de extensão da AWS SCT	509
Usando AWS Lambda funções para emular a funcionalidade do banco de dados	509
Como aplicar o pacote de extensão para oferecer suporte às funções do Lambda	510
Como configurar as funções do pacote de extensão	512
Práticas recomendadas	513
Como configurar memória adicional	513
Pasta padrão do projeto	513
Como aumentar a velocidade da migração de dados	514
Como aumentar as informações de registro em log	514
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	526
Editar cenários de CLI	527
Modo de script	528
Exemplos	529
Material de referência	529
Notas da versão	530
Notas de lançamento — 677	530
Notas da versão: 676	532
Notas da versão: 675	537
Notas de release: 674	540
Notas de release: 673	547
Notas de release: 672	552
Notas de release: 671	560
Notas de release: 670	569
Notas de release: 669	574
Notas de release: 668	579
Notas de release: 667	586
Notas de release: 666	590
Notas de release: 665	595
Notas de release: 664	598
Notas de release: 663	602
Notas de release: 662	605
Notas de release: 661	610
Notas de release: 660	615
Notas de release: 659	619
Notas de release: 658	623
Notas de release: 657	628
Notas de release: 656	633
Notas de release: 655	636

Notas de release: 654	639
Notas de release: 653	642
Notas de release: 652	644
Notas de release: 651	647
Notas de release: 650	649
Notas de release: 649	651
Notas de release: 648	654
Notas de release: 647	655
Notas de release: 646	657
Notas de release: 645	659
Notas de release: 644	660
Notas de release: 642	662
Notas de release: 641	664
Notas de release: 640	665
Alterações do release 1.0.640 do Oracle	665
Alterações da release 1.0.640 do Microsoft SQL Server	671
Alterações do MySQL versão 1.0.640	675
Alterações da release 1.0.640 do PostgreSQL	676
Alterações do release 1.0.640 do Db2 LUW	679
Alterações na release 1.0.640 do Teradata	680
Alterações da release 1.0.640 para outros mecanismos	682
Histórico de documentos	685
Atualizações anteriores	701
	dooy

O que é o AWS Schema Conversion Tool?

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter seu esquema de banco de dados existente de um mecanismo de banco de dados para outro. É possível 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 EC2 instância da Amazon ou armazenado como dados em um bucket do Amazon S3.

AWS SCT suporta vários padrões do setor, incluindo padrões federais de processamento de informações (FIPS), para conexões com um bucket do Amazon S3 ou AWS outro recurso. AWS SCT também está em conformidade com o Programa Federal de Gerenciamento de Riscos e Autorizações (FedRAMP). Para obter detalhes sobre os esforços de conformidade AWS e os esforços de conformidade, consulte AWS serviços no escopo por programa de conformidade.

AWS SCT suporta as seguintes conversões 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 Conectando-se à IBM DB2 para z/OS.
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 Bancos de dados do IBM Db2 LUW.
Banco de dados do Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

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

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)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
	Para obter mais informações, consulte <u>Bancos</u> <u>de dados SAP</u> .

AWS SCT suporta as seguintes conversões de data warehouse.

Data warehouse de origem	Data warehouse de destino
Amazon Redshift	Amazon Redshift
	Para obter mais informações, consulte <u>Amazon</u> <u>Redshift</u> .
Azure Synapse Analytics	Amazon Redshift
	Para obter mais informações, consulte <u>Azure</u> <u>Synapse Analytics como origem</u> .
BigQuery	Amazon Redshift
	Para obter mais informações, consulte BigQuery como fonte.
Banco de dados Greenplum (versões 4.3 e	Amazon Redshift
6.21)	Para obter mais informações, consulte <u>bancos</u> <u>de dados Greenplum</u> .
Microsoft SQL Server (versão 2008 e superior)	Amazon Redshift
	Para obter mais informações, consulte <u>Data</u> warehouses do SQL Server.
Netezza (versão 7.0.3 e superior)	Amazon Redshift

Data warehouse de origem	Data warehouse de destino
	Para obter mais informações, consulte <u>bancos</u> <u>de dados do Netezza</u> .
Oracle (versão 10.1 e superior)	Amazon Redshift
	Para obter mais informações, consulte <u>Data</u> <u>warehouse do Oracle</u> .
Snowflake (versão 3)	Amazon Redshift
	Para obter mais informações, consulte Snowflake.
Teradata (versão 13 e superior)	Amazon Redshift
	Para obter mais informações, consulte <u>Bancos</u> <u>de dados do Teradata</u> .
Vertica (versão 7.2.2 e superior)	Amazon Redshift
	Para obter mais informações, consulte <u>Bancos</u> <u>de dados do Vertica</u> .

AWS SCT suporta as 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 se conectar ao Apache Cassandra.

AWS SCT suporta conversões dos seguintes processos de extração, transformação e carregamento (ETL). Para obter mais informações, consulte <u>Conversão de dados usando ETL</u>.

Origem	Alvo
Scripts de ETL da Informatica	Informatica
Pacotes de ETL do Microsoft SQL Server Integration Services (SSIS)	AWS Glue or 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 trabalho da Teradata FastExport	Amazon Redshift RSQL
Scripts de trabalho da Teradata FastLoad	Amazon Redshift RSQL
Scripts de trabalho da Teradata MultiLoad	Amazon Redshift RSQL

AWS SCT suporta as seguintes migrações da estrutura de big data. Para obter mais informações, consulte Como migrar estruturas de big data.

Origem	Alvo
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

AWS SCT fornece uma interface de usuário baseada em projetos para converter automaticamente o esquema de banco de dados do seu banco de dados de origem em um formato compatível com sua instância de destino do Amazon RDS. Se o esquema do seu banco de dados de origem não puder ser convertido automaticamente, AWS SCT fornece orientação sobre como você pode criar um esquema equivalente no seu banco de dados de destino do Amazon RDS.

Para obter informações sobre como instalar AWS SCT, consulte<u>Instalando e configurando AWS</u> Schema Conversion Tool.

Para obter uma introdução à interface AWS SCT do usuário, consulte Navegando pela interface do usuário do AWS SCT.

Para obter mais informações sobre o processo de conversão, consulte <u>Convertendo esquemas de</u> banco de dados em AWS Schema Conversion Tool.

Além de converter seu esquema de banco de dados existente de um mecanismo de banco de dados para outro, AWS SCT tem alguns recursos adicionais que ajudam você a mover seus dados e aplicativos para a AWS nuvem:

- 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 Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.
- Você pode usar AWS SCT para criar AWS DMS endpoints e tarefas. Você pode executar e
 monitorar essas tarefas a partir de AWS SCT. Para obter mais informações, consulte <u>Integrando</u>
 com AWS Database Migration ServiceAWS Schema Conversion Tool.
- 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 AWS SCT extensões pode ajudá-lo a instalar AWS Lambda funções e bibliotecas Python para emular os recursos que não podem ser convertidos. Para obter mais informações, consulte <u>Usando pacotes de extensão</u> com AWS Schema Conversion Tool.
- 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 Convertendo dados do Amazon Redshift usando
 AWS Schema Conversion Tool.
- 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 Convertendo o SQL do aplicativo usando AWS SCT.

 Você pode usar a AWS SCT para migrar processos de extração, transformação e carregamento (ETL). Para obter mais informações, consulte <u>Convertendo dados usando processos ETL em AWS</u> Schema Conversion Tool.

Fornecendo feedback

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

Para fornecer feedback sobre AWS SCT

- Inicie AWS Schema Conversion Tool o.
- Abra o menu Ajuda e escolha Deixar um comentário. A caixa de diálogo Deixar um comentário é exibida.
- 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.

Dando feedback Versão 1.0.672 7

Instalando e configurando AWS Schema Conversion Tool

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 <u>AWS Schema Conversion</u> Tool CLI.

Tópicos

- Instalando AWS Schema Conversion Tool
- Validando a instalação AWS Schema Conversion Tool
- Instalando drivers JDBC para AWS Schema Conversion Tool
- Atualizando AWS Schema Conversion Tool
- AWS Schema Conversion Tool CLI

Instalando AWS Schema Conversion Tool

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

- Baixe o 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 do AWS SCT instalador, ele estará no formato apropriado para seu sistema operacional.
 - Microsoft Windows

Instalando AWS SCT Versão 1.0.672 8

- 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	<pre>aws-schema-conversion-tool-1.0. ber .x86_64.rpm</pre>	build-num
Microsoft Windows	AWS Schema Conversion Tool-1.0.	build-number .msi
Ubuntu Linux	aws-schema-conversion-tool-1.0.	build-number .deb

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

 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 <u>Instalando</u> drivers JDBC para AWS Schema Conversion Tool.

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

Instalando AWS SCT Versão 1.0.672 9

Validando a instalação AWS Schema Conversion Tool

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 SHA256 algoritmo para a soma de verificação.

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

- Baixe o arquivo de AWS SCT distribuição usando os links na seção Instalação. Para obter mais informações, consulte Instalando AWS Schema Conversion Tool.
- 2. Faça download do arquivo de soma de verificação mais recente, chamado <u>sha256Check.txt</u>. 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

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

Validar instalação Versão 1.0.672 10

```
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.

Para verificar os arquivos AWS SCT RPM no Fedora

- 1. Baixe o 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 <u>amazon.com.public.gpg-key</u>
- 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. Baixe o arquivo de AWS SCT distribuição usando os links na seção Instalação.
- 2. Verificando a soma de verificação do arquivo de AWS SCT distribuição.
- 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.
- 5. Faça download da chave pública GPG em amazon.com.public.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. Baixe o 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.

Instalando drivers JDBC para AWS Schema Conversion Tool

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 obter mais informações, consulte Mapeamento para alvos virtuais no AWS Schema Conversion Tool.

Depois de fazer download dos drivers, você atribui o local dos arquivos do driver. Para obter 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	<pre>mysql-connector-ja va-5.1.6.jar</pre>	https://www.mysql.com/products/conector/
Amazon Aurora Edição Compatíve I com PostgreSQL	postgresql-42.2.19 .jar	https://jdbc.postgresql. org/download/postgresql-42.2.19.jar

Instalação de drivers JDBC Versão 1.0.672 13

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-st andalone.jar	https://repo1.maven.org/maven2/org/apache/ hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standa lone.jar
Banco de dados do SQL Azure	mssql-jdbc-7.2.2.j re11.jar	https://docs.microsoft.com/en-us/sql/connect/ jdbc/release- notes-for-the-jdbc motorista? visão= sql-server-ver 15 #72
Azure Synapse Analytics	mssql-jdbc-7.2.2.j re11.jar	https://docs.microsoft.com/en-us/sql/connect/ jdbc/release- notes-for-the-jdbc motorista? 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/páginas/db2 downloads-db2-zos jdbc-driver-versions-and
IBM Db2 LUW	db2jcc-db2jcc4.jar	https://www.ibm.com/support/páginas/ node/382667
MariaDB	mariadb-java-clien t-2.4.1.jar	https://downloads.mariadb.com/Connectors/ java/connector-java-2.4.1/mariadb-java-clien t-2.4.1.jar

Instalação de drivers JDBC Versão 1.0.672 14

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 Note AWS SCT não suporta a versão mais
		recente do driver JDBC 18.2.1.0 para MSSQL. recomendamos instalar o driver JDBC versão 10.2 mssql-jdb c-10.2.jar).
MySQL	mysql-connector-ja va-8.0.15.jar	https://dev.mysql.com/downloads/conector/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 JDBC jConnect

Instalação de drivers JDBC Versão 1.0.672 15

Mecanismo do banco de dados	Drivers	Local de download
Snowflake	snowflake-jdbc-3.9 .2.jar Para obter mais informaçõ es, consulte Baixar/ Integrar o driver JDBC.	https://repo1.maven. org/maven2/net/sno wflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9 .2.jar
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/conectividade/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
```

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

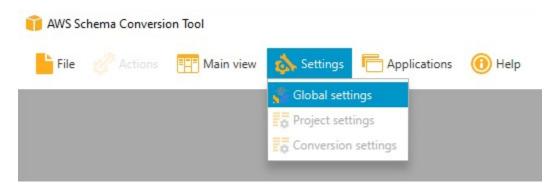
Mecanismo do banco de dados	Comandos de instalação
Amazon Aurora (compatível com MySQL)	PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X .X.X.tar.gz
Amazon Aurora (compatível com PostgreSQL)	PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar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

Armazenamento de caminhos do driver nas configurações globais

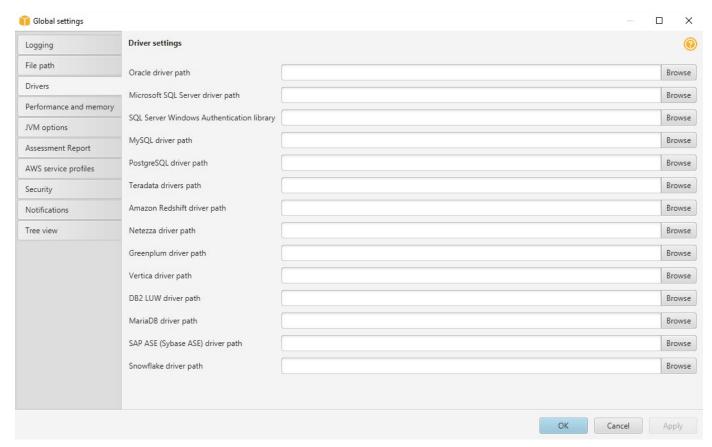
Depois de baixar e instalar os drivers JDBC necessários, você pode definir a localização dos drivers globalmente nas AWS SCT configurações. Se você não definir a localização dos drivers globalmente, o aplicativo solicitará a localização dos drivers quando você se conectar a um banco de dados.

Para atualizar os locais de arquivo do driver

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



 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 Schema Conversion Tool

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.

Atualizando AWS SCT Versão 1.0.672 18

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 Schema Conversion Tool CLI

Você pode baixar a AWS SCT CLI para uso na linha de comando. Use os links a seguir para fazer download do JAR:

AWSSchemaConversionToolBatch.jar

Navegando pela interface do usuário do AWS SCT

Use os tópicos a seguir para ajudá-lo a trabalhar com a interface AWS SCT do usuário. Para obter informações sobre a instalação AWS SCT, consulte<u>Instalando e configurando AWS Schema</u> Conversion Tool.

Tópicos

- Visualização da janela do projeto no AWS SCT
- Iniciando e gerenciando projetos em AWS SCT
- Usar o assistente do AWS SCT
- Salvando projetos em AWS SCT
- Adicionando servidores ao projeto em AWS SCT
- Usando o modo offline em AWS Schema Conversion Tool
- Usando filtros de árvore em AWS Schema Conversion Tool
- Ocultando esquemas em AWS Schema Conversion Tool
- Visualizando o relatório de avaliação em AWS Schema Conversion Tool
- Conversão de esquemas no AWS Schema Conversion Tool
- Aplicando os esquemas convertidos em AWS Schema Conversion Tool
- Gerenciando perfis no AWS Schema Conversion Tool
- Configurando AWS Secrets Manager no AWS Schema Conversion Tool
- Armazenando senhas no AWS Schema Conversion Tool
- Criando a visualização UNION ALL no AWS Schema Conversion Tool
- Usando atalhos de teclado no AWS Schema Conversion Tool

Visualização da janela do projeto no AWS SCT

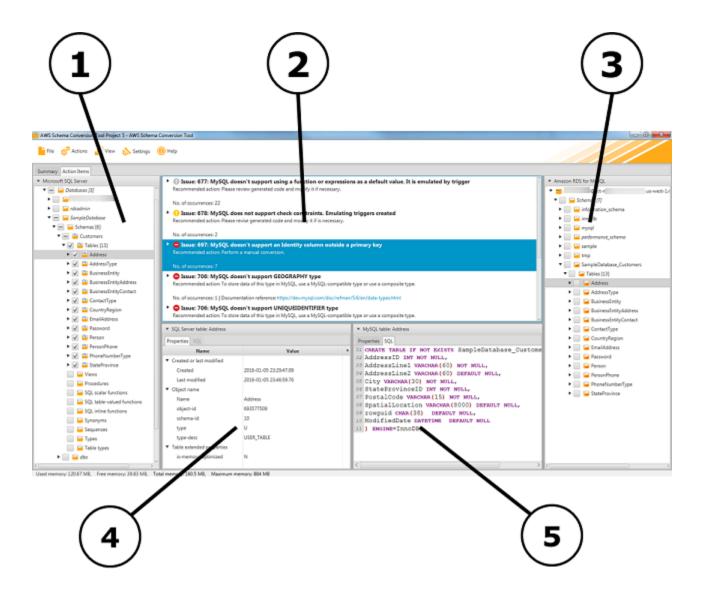
A ilustração a seguir é o que você vê na AWS SCT ao criar um projeto de migração de esquema e, em seguida, converter um esquema.

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, AWS SCT obtém e exibe o esquema atual do seu banco de dados de origem.

Janela do projeto Versão 1.0.672 20

2. 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.

3. 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, no momento em que você seleciona um item na visualização em árvore, 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.

Janela do projeto Versão 1.0.672 21

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.

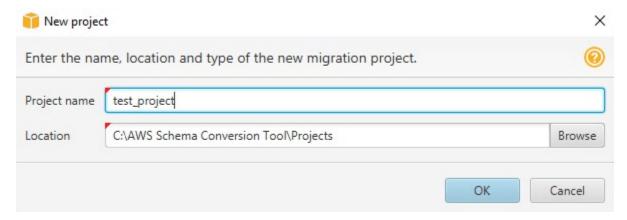
Iniciando e gerenciando projetos em AWS SCT

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

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

Para criar seu projeto

- Inicie AWS Schema Conversion Tool o.
- 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. Escolha OK para criar seu AWS SCT projeto.
- 6. Escolha Adicionar fonte para adicionar um novo banco de dados de origem ao seu AWS SCT projeto. Você pode adicionar vários bancos de dados de origem ao seu AWS SCT projeto.
- 7. Escolha Adicionar destino para adicionar uma nova plataforma de destino ao seu AWS SCT projeto. Você pode adicionar várias plataformas de destino ao seu AWS SCT projeto.
- 8. Escolha o esquema do banco de dados de origem no painel esquerdo.
- 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 Mapeamento de tipo de dados.

Agora, seu AWS SCT projeto 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.

Usar o assistente do 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, AWS SCT produz um relatório resumido para a migração do seu banco de dados para diferentes destinos de destino. 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 AWS Schema Conversion Tool o.
 - 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:

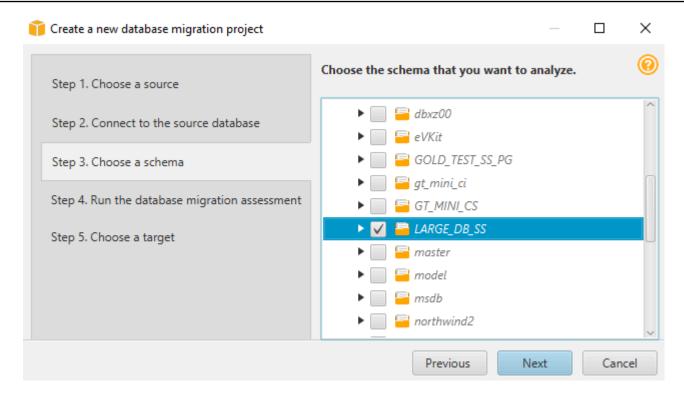
Parameter	Ação
Nome do projeto	Digite um nome para o projeto, que é armazenado localmente no computador.
Local	Digite o local do arquivo do projeto local.
Source type (Tipo de origem)	Escolha uma das seguintes opções: banco de dados SQL, banco de dados NoSQL ou ETL.

Usar o assistente do Versão 1.0.672 23

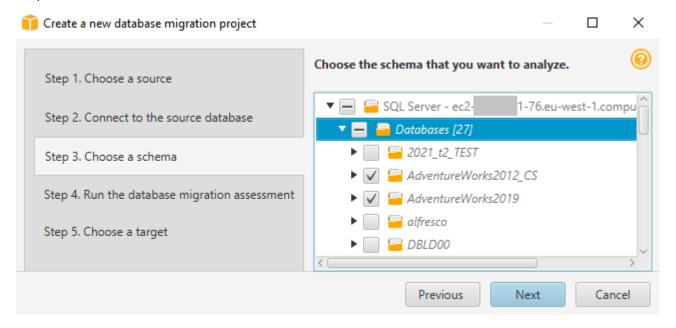
cê quiser ver o relatório resumido que inclui todos os destinos gração, escolha Banco de dados SQL.
na o mecanismo de banco de dados de origem.
na uma das seguintes opções: pero trocar de mecanismo e otimizar para a nuvem: Essa a converte seu banco de dados de origem em um novo canismo de banco de dados. pero manter o mesmo mecanismo, mas otimizar para a nuvem: a opção mantém o mecanismo de banco de dados como está cove o banco de dados on-premises para a nuvem. pero ver um relatório combinado sobre troca e otimização compecanismo de banco de dados para a nuvem: Essa opção compera a complexidade da migração de todas as opções de cração disponíveis. pê quiser ver o relatório de avaliação agregado que inclui cos destinos de migração, escolha a última opção.

- d. Escolha 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 <u>Como se conectar ao banco de dados de</u> origem com o AWS Schema Conversion Tool.
 - b. Escolha 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.

Usar o assistente do Versão 1.0.672 24



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. 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

Usar o assistente do Versão 1.0.672 25

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.
 AWS SCT salva os arquivos do relatório de avaliação na pasta especificada.
 - c. Escolha 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 Como se conectar ao banco de dados de origem com o AWS Schema Conversion Tool e Permissões para o Amazon Redshift como destino.
 - c. Escolha Concluir. AWS SCT cria seu projeto e adiciona as regras de mapeamento. Para obter mais informações, consulte Mapeamento de tipo de dados.

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

Salvando projetos em AWS SCT

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

Para salvar seu projeto

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

Salvar projetos Versão 1.0.672 26

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

Use o procedimento a seguir para abrir um AWS Schema Conversion Tool projeto 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. AWS SCT abre seu projeto, mas não se conecta automaticamente aos bancos de dados de origem e 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 AWS SCT versão 1.0.655 ou anterior, 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 Mapeamento de tipo de dados.

Adicionando servidores ao projeto em AWS SCT

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

Para adicionar um servidor ao projeto

- Inicie AWS Schema Conversion Tool o.
- 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 Como se conectar ao banco de dados de origem.

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

Para se conectar ao seu banco de dados

 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.
- Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 seu AWS SCT projeto.

Para remover um servidor de banco de dados

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

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.

Usando o modo offline em AWS Schema Conversion Tool

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

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.

Modo off-line Versão 1.0.672 28

Antes de usar AWS SCT no modo offline, conecte-se ao 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 usá-lo AWS SCT no modo offline.

Para executar AWS SCT em um modo off-line

- AWS Schema Conversion Tool Inicie o e crie um novo projeto. Para obter mais informações, consulte Iniciando e gerenciando projetos em AWS SCT.
- 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 ao projeto em AWS SCT.
- 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 <u>Mapeamento para alvos virtuais no</u> AWS Schema Conversion Tool.
- 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 <u>Mapeando tipos de</u> dados no AWS Schema Conversion Tool.
- 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. AWS SCT Se você executou uma dessas operações no menu de contexto, ignore a operação Carregar esquema.
- 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 AWS SCT no modo offline.

Usando filtros de árvore em AWS Schema Conversion Tool

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

Filtros de árvore Versão 1.0.672 29

Alguns bancos de dados podem ter um grande número de objetos na estrutura de árvore. Você pode usar filtros de árvore AWS SCT para pesquisar objetos nas estruturas da 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 funcionam com objetos pré-carregados. AWS SCT 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

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



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

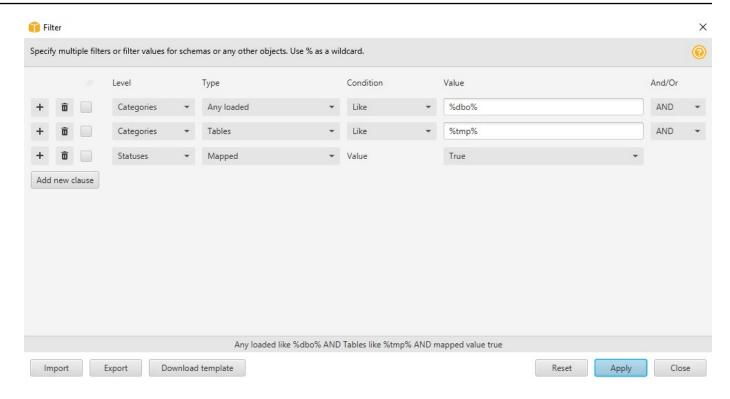
 Insira as informações a seguir na caixa de diálogo Filtro. As opções na caixa de diálogo são diferentes para cada mecanismo de banco de dados.

AWS SCT opção de filtro	Ação
Nível	Escolha Categorias para filtrar objetos por categorias.

Filtros de árvore Versão 1.0.672 30

AWS SCT opção de filtro	Ação
	Escolha Status para filtrar objetos por status.
Tipo	Em Categorias em Nível, escolha as categorias dos objetos filtrados . Escolha Qualquer carregado para exibir objetos de todas as categorias.
	Em Status em Nível, escolha o status dos objetos filtrados. Você pode escolher uma das seguintes opções:
	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	Para Categorias em Nível, escolha a condição de filtragem entre Curtir e Não curtir.
	Para Status em Nível, a opção de condição de filtragem não está disponível.
Valor	Para Categorias em Nível, insira o Valor para filtrar a árvore por esse valor.
	Use a porcentagem (%) como curinga para exibir todos os objetos.
	Para Status em Nível, escolha o Valor entre Verdadeiro e Falso.
E/Ou	Escolha operadores lógicos AND ou OR para aplicar várias cláusulas de filtro.

Filtros de árvore Versão 1.0.672 31



- Escolha Adicionar nova cláusula para adicionar uma cláusula de filtro adicional. AWS SCT pode aplicar várias cláusulas de filtro usando AND nossos operadores OR lógicos.
- Escolha 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 AWS SCT projeto existente, conecte-se ao banco de dados ao qual aplicar o filtro de árvore e escolha o ícone do 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.

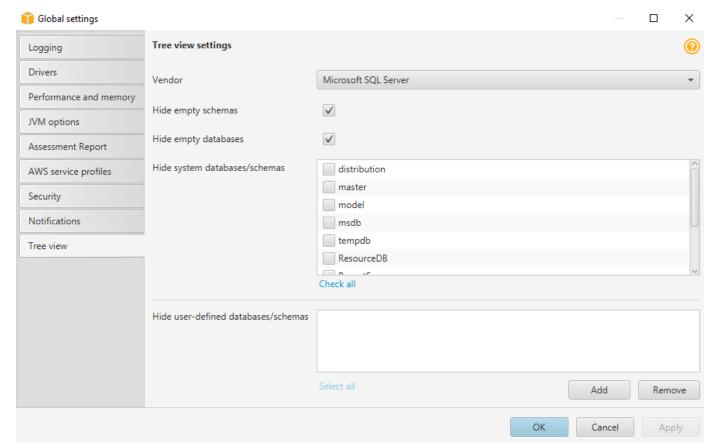
Ocultando esquemas em AWS Schema Conversion Tool

Use as configurações de exibição em árvore para especificar quais esquemas e bancos de dados você deseja ver na visualização em AWS SCT árvore. Você pode ocultar esquemas vazios, bancos de dados vazios, bancos de dados de sistemas além de esquemas e bancos de dados definidos pelo usuário.

Ocultando esquemas Versão 1.0.672 33

Para ocultar bancos de dados e esquemas na visualização em árvore

- 1. Abra um AWS SCT projeto.
- 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.

Ocultando esquemas Versão 1.0.672 34

Escolha OK.

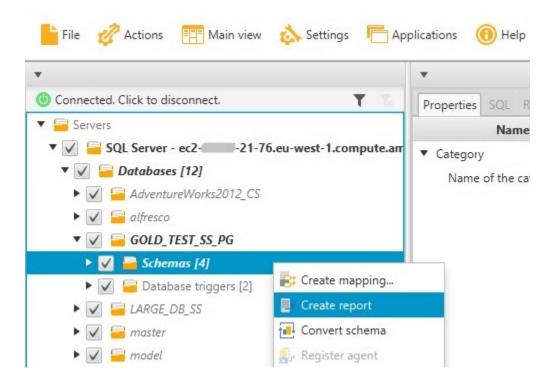
Visualizando o relatório de avaliação em AWS Schema Conversion Tool

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

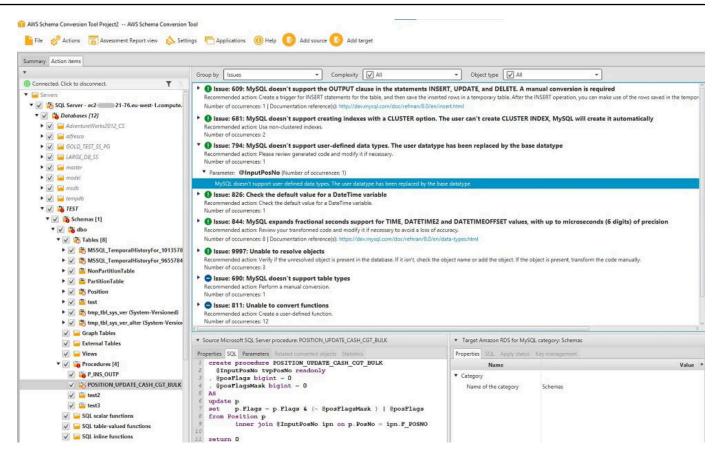
- Certifique-se de ter criado uma regra de mapeamento para o esquema do banco de dados de origem para o qual criar um relatório de avaliação. Para obter mais informações, consulte Mapeando novos tipos de dados no AWS Schema Conversion Tool.
- 2. No menu Visualizar, escolha Visualização principal.
- 3. 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.
- 4. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Criar relatório.



A visualização do relatório de avaliação se abre.

5. 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 dos itens de ação na lista. AWS SCT destaca o item do seu esquema ao qual o item de ação se aplica, conforme mostrado a seguir.

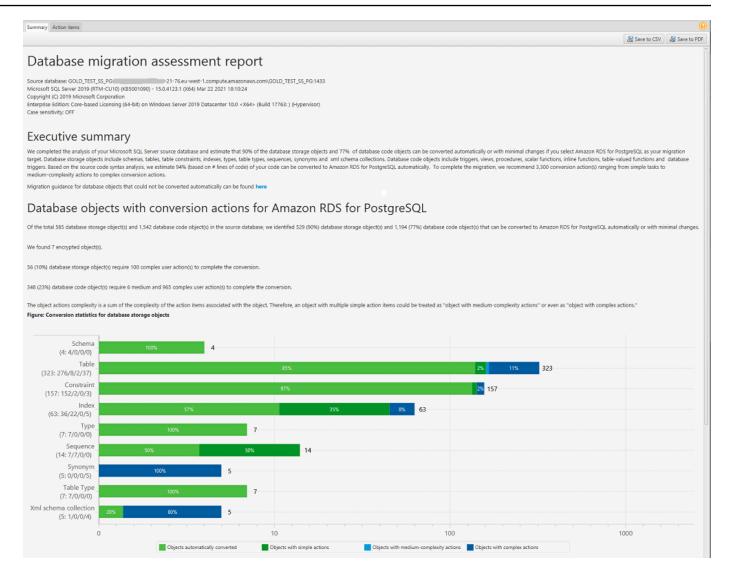


Escolha a guia Resumo.

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.



7. 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 dos itens de ação.

Você também pode escolher Salvar como CSV para salvar o relatório como um arquivo CSV. Quando você escolhe essa opção, 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

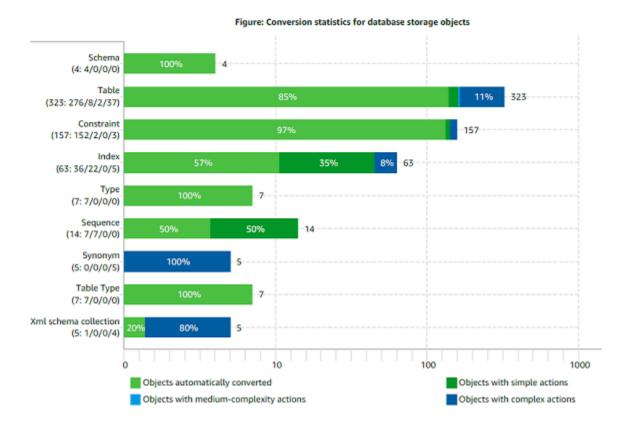
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identifed 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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



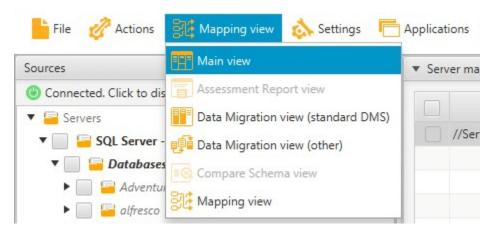
Conversão de esquemas no AWS Schema Conversion Tool

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.

Conversão de esquemas Versão 1.0.672 39

Para converter seu esquema

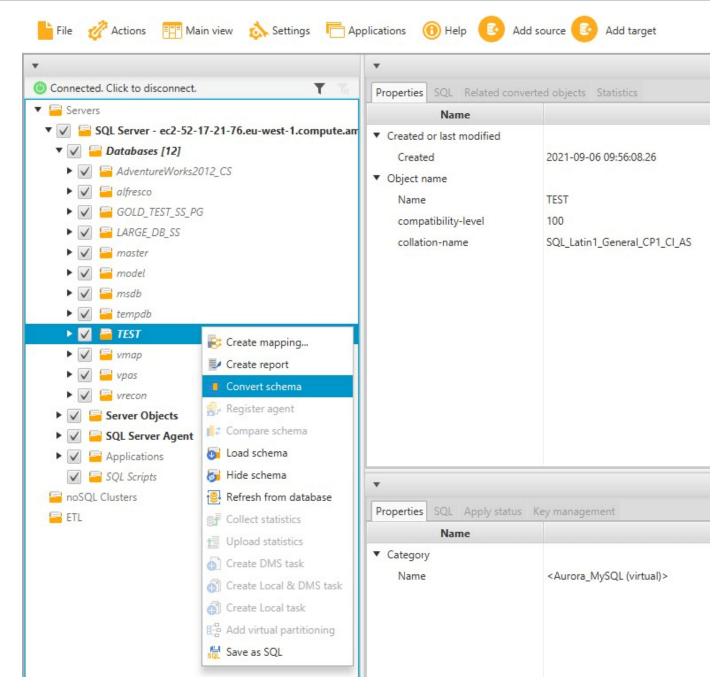
1. Escolha Exibir e, em seguida, Visualização principal.



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 AWS SCT destacar 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.

Conversão de esquemas Versão 1.0.672 40

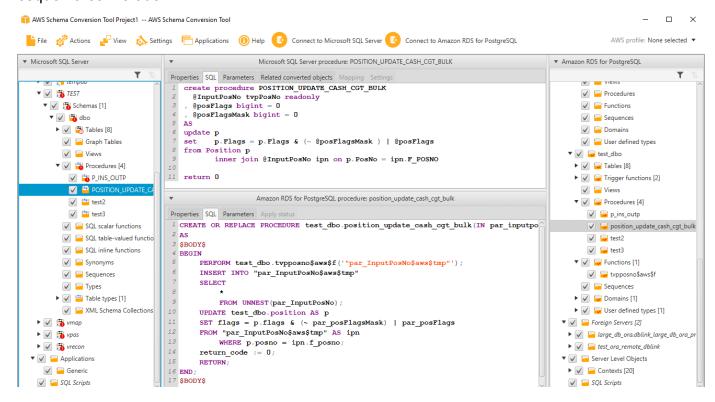


 Ao AWS SCT concluir a conversão do esquema, você poderá visualizar o esquema proposto no painel à direita do seu projeto.

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.

Conversão de esquemas Versão 1.0.672 41

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 os esquemas convertidos em AWS Schema Conversion Tool

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.

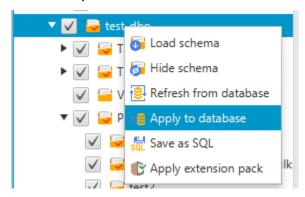


Marning

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

- 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.
- 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.

Gerenciando perfis no AWS Schema Conversion Tool

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

AWS SCT solicita suas AWS credenciais quando você acessa um recurso que as exige. Você pode armazenar suas credenciais nas configurações globais do aplicativo. Ao AWS SCT solicitar suas credenciais, você pode selecionar as credenciais armazenadas.

Você pode armazenar diferentes conjuntos de AWS credenciais 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 AWS s.

Como gerenciar perfis Versão 1.0.672 43

Armazenamento de AWS credenciais

Use o procedimento a seguir para armazenar AWS credenciais globalmente.

Para armazenar AWS credenciais

- 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. Escolha perfis AWS de serviço e, em seguida, escolha Adicionar um novo perfil AWS de serviço.
- 4. Insira suas AWS informações da seguinte forma.

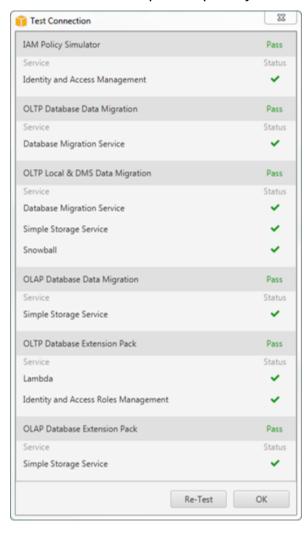
AWS SCT opção	Ação
Profile name	Insira um nome para o perfil.
AWS chave de acesso	Insira sua chave de AWS acesso.
AWS chave secreta	Insira sua chave de acesso AWS secreta. Para obter mais informaçõ es sobre chaves de AWS acesso, consulte Gerenciamento de chaves de acesso no Guia do usuário do IAM.
Região	Escolha o 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 AWS de serviço .

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 FIPS estão disponíveis nas seguintes regiões: AWS

- Região Leste dos EUA (N. da Virgínia)
- Região Leste dos EUA (Ohio)

- Região Oeste dos EUA (N. 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 para um AWS SCT projeto. Isso associa as AWS credenciais armazenadas no perfil ao projeto. Com seu projeto aberto, use o procedimento a seguir para definir o perfil padrão.

Para configurar o perfil padrão de um projeto

- AWS Schema Conversion Tool Inicie o 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 AWS de serviço para adicionar um novo perfil. Em Perfil do serviço da AWS, escolha o perfil que você deseja associar ao projeto.
- 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 AWS de serviço

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

- s3:Put0bject: 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.

Configurando AWS Secrets Manager no AWS Schema Conversion Tool

AWS SCT pode usar as credenciais do banco de dados que você armazena. 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 AWS perfis no AWS Schema Conversion Tool.

Para obter mais informações sobre o uso AWS Secrets Manager, consulte <u>O que é AWS Secrets</u> <u>Manager?</u> no Guia do AWS Secrets Manager usuário. Para obter mais informações sobre o armazenamento AWS de perfis, consulteGerenciando perfis no AWS Schema Conversion Tool.

Para recuperar credenciais de banco de dados do Secrets Manager

- 1. AWS Schema Conversion Tool Inicie o 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, AWS SCT preenche todos os valores na caixa de diálogo de conexão do banco de dados.
- 6. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu banco de dados.
- 7. Escolha Conectar para se conectar ao seu banco de dados.

AWS SCT suporta 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 no AWS Schema Conversion Tool

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

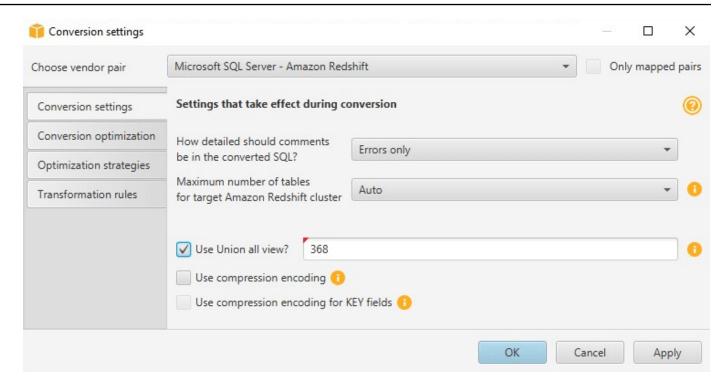
A senha é criptografada usando o token gerado aleatoriamente no arquivo seed.dat. Em seguida, ela é armazenada com o nome de usuário no arquivo de cache. Caso você perca o arquivo seed.dat ou ele seja corrompido, a senha do banco de dados pode ser descriptografada de maneira incorreta. Nesse caso, ocorre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

Criando a visualização UNION ALL no AWS Schema Conversion Tool

Se uma tabela de origem for particionada, AWS SCT cria n tabelas de destino, onde n é o número de partições na tabela de origem. 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 AWS SCT dados para migrar seus dados, as partições da tabela de origem serão extraídas e carregadas paralelamente por subtarefas separadas.

Para usar a exibição Union All em um projeto

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



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

Usando atalhos de teclado no AWS Schema Conversion Tool

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

Atalho de teclado	Descrição
Ctrl+N	Criar um novo projeto da .
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.

Atalhos de teclado Versão 1.0.672 49

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

Atalhos de teclado Versão 1.0.672 50

Começando com AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter o esquema para um banco de dados de origem. O banco de dados de origem pode ser um mecanismo autogerenciado executado localmente ou em uma instância da Amazon EC2. Você pode converter seu esquema de origem em um esquema para qualquer banco de dados compatível hospedado pelo AWS. O AWS SCT aplicativo fornece uma interface de usuário baseada em projetos.

Quase todo o trabalho que você faz AWS SCT começa com as seguintes etapas:

- Instalar AWS SCT. Para obter mais informações, consulte <u>Instalando e configurando AWS</u> Schema Conversion Tool.
- Instale um AWS SCT agente, se necessário. AWS SCT os agentes são necessários somente para determinados cenários de migração, como entre fontes e destinos heterogêneos. Para obter mais informações, consulte Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.
- 3. Familiarize-se com a interface do usuário do AWS SCT. Para obter mais informações, consulte Navegando pela interface do usuário do AWS SCT.
- 4. Crie um AWS SCT projeto. 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 <u>Como se</u> conectar ao banco de dados de origem com o AWS Schema Conversion Tool.
- 5. Crie regras de mapeamento. Para obter mais informações sobre regras de mapeamento, consulte Mapeando tipos de dados no AWS Schema Conversion Tool.
- Execute e analise o relatório de avaliação de migração do banco de dados. Para obter mais informações sobre o relatório de avaliação, consulte <u>Visualizando o relatório de avaliação em</u> AWS Schema Conversion Tool.
- 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 Convertendo esquemas de banco de dados em AWS Schema Conversion Tool.

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 <u>Convertendo</u> esquemas de data warehouse para o Amazon RDS usando 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 Como aplicar esquemas convertidos.

 Você também pode usar AWS SCT para converter procedimentos SQL armazenados e outros códigos de aplicativos. Para ter mais informações, consulte <u>Convertendo o SQL do aplicativo</u> <u>usando AWS SCT</u>

Você também pode usar AWS SCT para migrar seus dados de um banco de dados de origem para um banco de dados gerenciado pela Amazon. Para obter exemplos, consulte Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.

Como se conectar ao banco de dados de origem com o AWS Schema Conversion Tool

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

Informações de criptografia

Como se conectar ao Amazon RDS e ao Aurora criptografados

Origens dos bancos de dados

- Como se conectar ao Apache Cassandra
- Conectar ao Azure SQL
- Conectando-se à IBM DB2 para z/OS
- · Bancos de dados do IBM Db2 LUW
- Usar o MySQL como origem
- Bancos de dados Oracle
- · bancos de dados do PostgreSQL
- Bancos de dados SAP
- bancos de dados do SQL Server

Origens de data warehouse

- Amazon Redshift
- Azure Synapse Analytics como origem
- BigQuery como fonte
- bancos de dados Greenplum
- bancos de dados do Netezza
- Data warehouse do Oracle
- Snowflake

- · Data warehouses do SQL Server
- · Bancos de dados do Teradata
- Bancos de dados do Vertica

Origens de big data

- Como se conectar ao Apache Hadoop
- · Como se conectar ao Apache Oozie

Conectando-se aos bancos de dados criptografados do Amazon Relational Database Service e do Amazon Aurora com o AWS Schema Conversion Tool

Para abrir conexões criptografadas com bancos de dados Amazon RDS ou Amazon Aurora a partir de um aplicativo, você precisa AWS importar certificados raiz em alguma forma de armazenamento de chaves. Você pode baixar os certificados raiz AWS em <u>Usando SSL/TLS para criptografar uma conexão com uma instância de banco de dados no Guia do usuário do Amazon RDS.</u>

Duas opções estão disponíveis: um certificado raiz que funciona para todas as AWS regiões e um pacote de certificados que contém os certificados raiz antigos e novos.

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 <u>Como usar SSL/TLS para criptografar uma conexão com uma instância de banco de dados</u> 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.

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.

- 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.
- Reinicie o computador.

Para importar o certificado ou certificados para o Java KeyStore

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

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
                                                          ` . . .
1
]
#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen: 2147483647
1
#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 AWS SCT conexão com o banco de dados. Na caixa de diálogo AWS SCT Conectar ao banco de dados, escolha Usar SSL e escolha o armazenamento confiável inserido anteriormente.

Conectando-se aos bancos de dados do Apache Cassandra com o AWS Schema Conversion Tool

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

- No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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 você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.
Usar SSL	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:
	 Armazenamento confiável: o armazenamento confiável a ser usado.
	Armazenamento de chaves: o armazenamento de chaves a ser usado.

Parameter	Ação
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 AWS SCT pode se conectar ao seu banco de dados de origem.
- Escolha Conectar para se conectar ao banco de dados de origem.

Conectando-se aos bancos de dados do Apache Hadoop com o AWS Schema Conversion Tool

Você pode usar a interface de linha de AWS SCT comando (CLI) para migrar do Apache Hadoop para o Amazon EMR. AWS SCT usa seu bucket Amazon S3 como armazenamento temporário para seus dados durante a migração.

AWS SCT suporta como fonte o Apache Hadoop versão 2.2.0 e superior. Além disso, é AWS SCT compatível com o Apache Hive versão 0.13.0 e superior.

AWS SCT suporta como destino a versão 6.3.0 e superior do Amazon EMR. Além disso, AWS SCT suporta como destino o Apache Hadoop versão 2.6.0 e superior e o 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 seus workloads do Hadoop. Para obter mais informações, consulte <u>Como criar um bucket</u> no Guia do usuário do Amazon S3.
- Crie uma função AWS Identity and Access Management (IAM) com a AmazonS3FullAccess política. AWS SCT usa essa função do IAM para acessar seu bucket do Amazon S3.
- Anote sua chave AWS secreta e sua chave de acesso AWS secreta. Para obter mais informações sobre chaves de AWS acesso, consulte <u>Gerenciamento de chaves de acesso</u> no Guia do 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 distop em seu cluster Apache Hadoop de origem. Além disso, instale o utilitário s3-dist-op 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.

O exemplo anterior mostra uma das quatro opções da lista de opções anterior. Se você não definir o fs.s3a.aws.credentials.provider parâmetro no core-site.xml arquivo, 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:

- EXECUTEpara o NameNode
- EXECUTE+READ para todas as pastas e arquivos de origem que você inclui no projeto de migração
- READ+WRITEpara o tmp diretório 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:

- EXECUTEpara o NameNode cluster de destino do Amazon EMR
- 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 fonte na AWS SCT versão 1.0.670 ou superior. Você pode migrar clusters do Hadoop para o Amazon EMR somente na interface de AWS SCT linha de comando (CLI). Antes de começar a usar, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte Referência de CLI para AWS Schema Conversion Tool.

Para se conectar ao Apache Hadoop na CLI AWS SCT

- Crie um novo script de AWS SCT CLI 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 AWS SCT aplicativo, como a localização do driver e a pasta de registro.

Faça o download do driver JDBC necessário e especifique o local de armazenamento do arquivo. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema</u> Conversion Tool.

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

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

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

3. Crie um novo AWS SCT projeto.

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. Certifiquese 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: 'HAD00P_SOURCE'
-vendor: 'HAD00P'
-host: 'hadoop_address'
-port: '22'
-user: 'hadoop_user'
-password: 'hadoop_password'
-useSSL: 'true'
-privateKeyPath: 'c:\path\name.pem'
-passPhrase: 'hadoop_passphrase'

/
```

No exemplo anterior, *hadoop_address* substitua 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 a senha desse usuário. Parapath \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 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 Instalando drivers JDBC para AWS Schema Conversion Tool.

AWS SCT se conecta ao Apache Hive com o usuário do hadoop cluster. 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 <u>Configurar um túnel SSH ao nó primário usando o encaminhamento de portas locais no Guia de gerenciamento do Amazon EMR.</u>

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 AWS SCT

- 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. Certifiquese 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 AWS SCT para 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: 'HAD00P_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 usuário do Hive e a senha desse usuário.

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

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

Adicione seu serviço HDFS de origem ao projeto.

Use o comando AddSourceClusterHDFS para conectar o serviço HDFS de origem. Certifiquese 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 <u>Permissões para usar o Hive como origem</u>.

O exemplo de código a seguir cria um túnel AWS SCT para trabalhar com seu serviço Apache HDFS. Este exemplo usa o cluster HAD00P_S0URCE de origem que você criou anteriormente.

```
AddSourceClusterHDFS
-cluster: 'HAD00P_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 usuário do HDFS e a senha desse usuário.

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

Para isso*hdfs_remote_address*, você pode usar o valor padrão 127.0.0.1 ou o endereço NameNode IP 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 de destino do Amazon EMR com a CLI 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 Amazon EMR Hive com o usuário do hadoop cluster. 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 Instalando drivers JDBC para AWS Schema Conversion Tool.

Para se conectar ao Amazon EMR na CLI 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 HAD00P_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'
```

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

4. Adicione seu serviço Hive de destino ao projeto.

O exemplo de código a seguir cria um túnel AWS SCT para 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 a senha desse usuário.

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

Adicione seu serviço HDFS de destino ao projeto.

O exemplo de código a seguir cria um túnel AWS SCT para trabalhar com seu serviço Apache HDFS. Este exemplo usa o cluster HAD00P_TARGET de destino que você criou anteriormente.

```
AddTargetClusterHDFS
-cluster: 'HAD00P_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 a senha desse usuário.

Em seguida, *hdfs_port* substitua *hdfs_address* e pelo endereço IP privado e pela porta NameNode do 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 Migração de workloads do Hadoop.

Conectando-se aos fluxos de trabalho do Apache Oozie com o AWS Schema Conversion Tool

Você pode usar a interface de linha de AWS SCT comando (CLI) para converter fluxos de trabalho do Apache Oozie em. AWS Step Functions Depois de migrar suas cargas de trabalho do Apache Hadoop para o Amazon EMR, você pode usar um serviço nativo no para orquestrar seus trabalhos. Nuvem AWS Para obter mais informações, consulte Como se conectar ao Apache Hadoop.

AWS SCT converte seus fluxos de trabalho do Oozie AWS Step Functions e os usa AWS Lambda para emular recursos que AWS Step Functions não são compatíveis. Além disso, AWS SCT converte suas propriedades de trabalho da Oozie em. AWS Systems Manager

Para converter fluxos de trabalho do Apache Oozie, certifique-se de usar a AWS SCT versão 1.0.671 ou superior. Além disso, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte Referência de CLI para AWS Schema Conversion Tool.

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 uma função AWS Identity and Access Management (IAM) com a AmazonS3FullAccess política. AWS SCT usa essa função do IAM para acessar seu bucket do Amazon S3.
- Anote sua chave AWS secreta e sua chave de acesso AWS secreta. Para obter mais informações sobre chaves de AWS acesso, consulte <u>Gerenciamento de chaves de acesso</u> no Guia do usuário do IAM.
- Armazene suas AWS credenciais e as informações sobre seu bucket do Amazon S3 no perfil
 de serviço AWS nas configurações globais do aplicativo. Em seguida, AWS SCT usa esse perfil
 de AWS serviço para trabalhar com seus AWS recursos. Para obter mais informações, consulte
 Gerenciando perfis no AWS Schema Conversion Tool.

Para trabalhar com seus fluxos de trabalho de origem do Apache Oozie, é AWS SCT necessária a 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 AWS SCT

- Crie um novo script de AWS SCT CLI 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 AWS SCT aplicativo.

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. Crie um novo AWS SCT projeto.

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_00ZIE 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 fonte em seu AWS SCT projeto. Este exemplo cria um objeto de origem com o nome 00ZIE. Use esse nome de objeto para adicionar regras de mapeamento. Depois de executar esse exemplo de código, AWS SCT usa a c:\oozie pasta para carregar seus arquivos de origem no projeto.

```
AddSource
-name: '00ZIE'
-vendor: 'APACHE_00ZIE'
-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 AWS Step Functions serviço.

Permissões para usar AWS Lambda funções no pacote de extensões

Para as funções de origem que AWS Step Functions não são compatíveis, AWS SCT cria um pacote de extensão. Esse pacote de extensão inclui AWS Lambda funções que emulam suas funções de origem.

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

```
"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 extensões, é AWS SCT necessária uma função do IAM com as seguintes permissões.

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

Conectando-se AWS Step Functions como alvo

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

Para se conectar AWS Step Functions na AWS SCT CLI

 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 AWS SCT projeto usando o AddTarget comando. 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 é adicionado AWS Step Functions como fonte em seu AWS SCT projeto. 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 AWS SCT de serviço que você criou na etapa de pré-requisitos. Certifique-se de *profile_name* substituir pelo nome do seu perfil.

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

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

 Conecte-se AWS Step Functions usando o ConnectTarget comando. 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 *profile name* substituir 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 Conversão de fluxos de trabalho do Oozie;.

Conectando-se aos bancos de dados SQL do Microsoft Azure com o AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo do Banco de Dados SQL do Azure 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.

Conectar ao Azure SQL Versão 1.0.672 76

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

- 1. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
Banco de dados	Insira o nome do banco de dados ao qual se conectar.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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

Parameter	Ação
	conectar ao banco de dados de origem, conforme necessári o.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 AWS SCT pode se conectar ao seu banco de dados de origem.
- 6. Escolha Conectar para se conectar ao banco de dados de origem.

Conectando-se aos bancos de dados IBM DB2 for z/OS com o AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo do IBM Db2 for 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
- SELECIONE NO SISTEMA. 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

No AWS Schema Conversion Tool, escolha Adicionar fonte.

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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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 evitar a exposição da senha do seu banco de dados de origem, AWS SCT não armazena a senha por padrão. Se você fechar o AWS SCT projeto e reabri-lo, será solicitada a

Parameter	Ação
	senha para se conectar ao banco de dados de origem, conforme necessário.
Usar SSL	 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: Armazenamento de confiança: a localização de um armazenamento de confiança que contém certifica dos. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 Db2 para z/OS	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> . 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 <u>Armazenamento de caminhos do driver nas configurações globais</u> .

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 *.*
- SELECT ON mysql.proc
- INSIRA E ATUALIZE EM AWS_DB 2ZOS_EXT. *
- INSIRA, ATUALIZE E EXCLUA NO AWS DB 2ZOS EXT DATA. *
- CRIE TABELAS TEMPORÁRIAS EM AWS_DB 2ZOS_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 SELECT 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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 AWS SCT, é necessário o privilégio. CREATE ON DATABASE Certifique-se de conceder esse privilégio para cada banco de dados PostgreSQL de destino.

Para usar o Amazon RDS for PostgreSQL como destino, é necessário o privilégio. AWS SCT 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua 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 não AWS SCT 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 for z/OS para PostgreSQL incluem opções para o seguinte: AWS SCT

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. 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.

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.

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, 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. 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.

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.

AWS SCT pode converter procedimentos de origem que retornam conjuntos de resultados dinâmicos em procedimentos que têm uma matriz de refcursores 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.

AWS SCT pode converter valores de data e hora em representações de seqüências de caracteres 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)

ConnConnecting para a IBM DB2 para bancos de dados Linux, UNIX e Windows com o AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código na linguagem SQL e código de aplicativo do IBM Db2 for 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

AWS SCT suporta como fonte as versões 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 e 11.5 do Db2 LUW.

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:
 - EXECUTAR NO PACOTE 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

Privilégios do Db2 LUW Versão 1.0.672 88

SELECIONE NO SISTEMA, 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 ao usuário:

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
```

Privilégios do Db2 LUW Versão 1.0.672 89

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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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.
Banco de dados	Digite o nome do banco de dados Db2 LUW.

Parameter	Ação
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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 evitar a exposição da senha do seu banco de dados de origem, AWS SCT não armazena a senha por padrão. Se você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.
Usar SSL	 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: Armazenamento de confiança: a localização de um armazenamento de confiança que contém certifica dos. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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.

Parameter	Ação
Caminho do driver Db2 LUW	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

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

Migração da IBM DB2 para Linux, UNIX e Windows para o Amazon Relational Database Service for PostgreSQL ou Amazon Aurora PostgreSQL Compatible Edition

Quando você migra o IBM Db2 LUW para o PostgreSQL, AWS SCT pode converter várias instruções de gatilho 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. Você pode especificar qualquer combinação dos eventos INSERT, DELETE e UPDATE, mas você pode especificar cada evento somente uma vez. AWS SCT suporta eventos de gatilho únicos e múltiplos. 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

Db2 LUW para PostgreSQL Versão 1.0.672 92

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

AWS SCT pode converter tabelas LUW do Db2 em tabelas particionadas no 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

Db2 LUW para PostgreSQL Versão 1.0.672 93

devem ser removidos da tabela pai. O nome do índice convertido está no formato original_index_name, original_index_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 AWS SCT, é necessário o 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 <u>user_name</u> LOGIN PASSWORD '<u>your_password</u>';
GRANT CREATE ON DATABASE <u>db_name</u> TO <u>user_name</u>;
ALTER DATABASE <u>db_name</u> SET SEARCH_PATH = "$user", public_synonyms, public;
```

No exemplo anterior, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua 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 não AWS SCT consegue descartar um esquema. Para evitar essa mensagem de erro, use o perfil superuser.

Migração da IBM DB2 para Linux, UNIX e Windows para o Amazon RDS for 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.

Db2 LUW para MySQL Versão 1.0.672 94

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
- INSIRA E ATUALIZE EM AWS_DB 2_EXT. *
- INSIRA, ATUALIZE E EXCLUA EM AWS_DB 2_EXT_DATA. *
- CRIE TABELAS TEMPORÁRIAS EM AWS_DB 2_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';
```

Db2 LUW para MySQL Versão 1.0.672 95

```
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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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 MySQL como fonte para AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código de banco de dados e 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:

Usar o MySQL como origem Versão 1.0.672 96

- 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha MySQL e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	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 MySQL de origem usando um protocolo de IPv6 endereço. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.

Parameter	Ação
	[2001:db8:ffff:ffff:ffff:ffff:fffe]
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.

Parameter	Ação
Usar SSL	 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: Exigir SSL: escolha essa opção se deseja conectar-se ao servidor somente por meio de SSL. 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. 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	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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.
MySql caminho do motorista	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais.

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

6. 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 AWS SCT, é necessário o 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 <u>user_name</u> LOGIN PASSWORD '<u>your_password</u>';

GRANT CREATE ON DATABASE <u>db_name</u> TO <u>user_name</u>;

ALTER DATABASE <u>db_name</u> SET SEARCH_PATH = "$user", public_synonyms, public;
```

No exemplo anterior, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua 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 não AWS SCT consegue descartar um esquema. Para evitar essa mensagem de erro, utilize o perfil superuser.

Conectando-se aos bancos de dados Oracle com o AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código de banco de dados e 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 apóstrofos; AWS SCT duplica os apóstrofos ao converter instruções SQL, assim como acontece com literais de string.

Para obter mais informações, consulte.

Tópicos

- Privilégios do Oracle como origem
- Conectar-se ao Oracle como origem
- Migração da Oracle para o Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL com AWS Schema Conversion Tool
- Migração da Oracle para o Amazon RDS for MySQL ou Amazon Aurora MySQL com o AWS Schema Conversion Tool
- Migração do banco de dados Oracle para o Amazon RDS for Oracle com AWS Schema Conversion Tool

Privilégios do Oracle como origem

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

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

Privilégios do Oracle Versão 1.0.672 101

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

- No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha Oracle e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

- 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 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.
 - 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Tipo	 Escolha o tipo de conexão ao seu banco de dados. Dependendo do tipo, forneça as informações adicionais seguintes: SID 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.

Parameter	Ação
	 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','inst ance_name') AS SID FROM dual;</pre>
	Nome do serviço
	 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 Oracle de origem usando um protocolo IPv6 de endereço. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.
	[2001:db8:ffff:ffff:ffff:ffff:fffe]
	 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
	 Caminho do arquivo TNS: o caminho para o arquivo que contém as informações de conexão do nome Transparent Network Substrate (TNS).
	Depois de escolher o arquivo TNS, AWS SCT adiciona todas as conexões do banco de dados Oracle do arquivo à lista de aliases do TNS.
	Escolha essa opção para se conectar ao Oracle Real Application Clusters (RAC).
	 Alias do TNS: o alias do TNS do arquivo a ser usado para a conexão com o banco de dados de origem.

Parameter	 Ação Identificador de conexão TNS Identificador de conexão TNS: o identificador das informações de conexão TNS registradas.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 baixar o 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. 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 você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.

Parameter	Ação
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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 Oracle	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

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

Migração da Oracle para o Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL com AWS Schema Conversion Tool

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 em PostgreSQL AWS SCT, realiza 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\$v ersion
V\$INSTANCE	Uma exibição que mostra o estado da instância atual.	aws_oracle_ext.v\$i nstance

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 Convertendo o SQL do aplicativo usando AWS SCT.

Privilégios do PostgreSQL como um banco de dados de destino

Para usar o PostgreSQL como destino AWS SCT, é necessário o 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 <u>user_name</u> LOGIN PASSWORD '<u>your_password</u>';

GRANT CREATE ON DATABASE <u>db_name</u> TO <u>user_name</u>;

ALTER DATABASE <u>db_name</u> SET SEARCH_PATH = "$user", public_synonyms, public;
```

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

Para usar o Amazon RDS for PostgreSQL como destino, é necessário o privilégio. AWS SCT 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 não AWS SCT 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 o PostgreSQL, escolha Configurações em e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle — PostgreSQL. AWS SCT exibe todas as configurações disponíveis para conversão de Oracle para PostgreSQL.

As configurações AWS SCT de conversão de Oracle para PostgreSQL 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. 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 AWS SCT a conversão de visualizações materializadas da Oracle em tabelas ou visualizações 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 T0_CHAR, T0_DATE e T0_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, T0_CHAR, T0_DATE e T0_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 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 ou seus usuários não estiverem familiarizados com o uso de procedimentos no PostgreSQL AWS SCT, podem converter procedimentos Oracle em funções 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 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 orafce, consulte orafce on. 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 AWS SCT projeto com a Oracle como fonte.
- 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, então 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 a pseudocoluna ROWID, AWS SCT pode criar uma coluna de dados com o tipo de dados. bigint Se não existir uma chave primária, AWS SCT define a coluna ROWID como a chave primária. Se existir uma chave primária, AWS SCT define 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 character varying tipo de dados.

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

- 1. Abra seu AWS SCT projeto com a Oracle como fonte.
- 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.
- Escolha OK para salvar as configurações e fechar a caixa de diálogo Configurações de conversão.

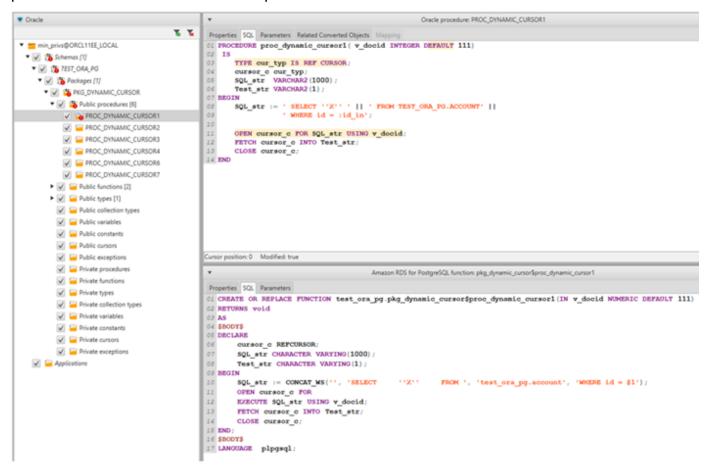
Converter o SQL dinâmico do Oracle

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 AWS SCT projeto com a Oracle como fonte.
- Escolha um objeto de banco de dados que use SQL dinâmico na exibição em árvore de origem da Oracle.
- 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 suporta 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)

Migração da Oracle para o Amazon RDS for MySQL ou Amazon Aurora MySQL com o AWS Schema Conversion Tool

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 pacotes de extensão com AWS Schema Conversion Tool.

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_ACESSO
- INSIRA, ATUALIZE EM AWS_ORACLE _EXT. *
- INSIRA, ATUALIZE E EXCLUA EM 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 _ACCESS. AWS_LAMBDA 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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 de Oracle para MySQL, escolha Configurações em e AWS SCT, em seguida, escolha Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle — MySQL. AWS SCT exibe todas as configurações disponíveis para conversão de Oracle para MySQL.

As configurações de conversão de Oracle para MySQL 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. 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 ROWID pseudocoluna, mas o MySQL não oferece suporte a funcionalidades semelhantes. AWS SCT pode emular a ROWID pseudocoluna 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 TO CHAR, TO DATE e TO 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 G0T0 e um rótulo. Qualquer instrução PL/SQL que segue uma instrução G0T0 é ignorada, e o processamento continua no rótulo. As instruções G0T0 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 G0T0. Quando AWS SCT converte o código que contém uma G0T0 instrução, ele converte a instrução para usar uma instrução BEGIN...END ouL00P...END L00P.

Você pode encontrar exemplos de como AWS SCT converte G0T0 declarações na tabela a seguir.

```
Instrução do Oracle
                                                  Instrução do MySQL
 BEGIN
                                                   BEGIN
                                                    label1:
                                                    BEGIN
    statement1;
    GOTO label1;
                                                      statement1;
    statement2;
                                                      LEAVE label1;
    label1:
                                                      statement2;
    Statement3;
                                                       . . . .
                                                    END;
 END
                                                      Statement3;
                                                   END
```

Instrução do Oracle

Instrução do MySQL

```
BEGIN

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

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

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

```
BEGIN

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

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), AWS SCT converterá a cláusula em uma exibição ou 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

```
CREATE PROCEDURE
                                              CREATE PROCEDURE test_ora_pg.P_WITH
TEST_ORA_PG.P_WITH_SELECT_V
                                              _SELECT_VARIABLE_01(IN par_P_STATE
                                               DOUBLE)
ARIABLE 01
     (p_state IN NUMBER)
                                              BEGIN
AS
                                                  DECLARE var_l_dept_id DOUBLE
  l_dept_id NUMBER := 1;
                                               DEFAULT 1;
BEGIN
                                                  DECLARE var$id VARCHAR (8000);
FOR cur IN
                                                  DECLARE var$state VARCHAR (8000);
           (WITH dept_empl(id, name,
                                                  DECLARE done INT DEFAULT FALSE;
                                                  DECLARE cur CURSOR FOR SELECT
 surname,
              lastname, state, dept_id)
                                                      ID, STATE
              AS
                                                      FROM (SELECT
                                                          ID, NAME, SURNAME,
                                               LASTNAME, STATE, DEPT_ID
                    SELECT id, name,
                                                           FROM TEST_ORA_PG.DEPT_E
 surname,
                     lastname, state,
                                              MPLOYEES
dept_id
                                                          WHERE STATE = par_p_sta
                                              te AND DEPT_ID = var_l_dept_id) AS
                      FROM test_ora_
                                               dept_empl
pg.dept_employees
                     WHERE state =
                                                      ORDER BY ID;
                                                  DECLARE CONTINUE HANDLER FOR NOT
 p_state AND
                                               FOUND
```

Instrução do Oracle

```
dept_id =
l_dept_id)

SELECT id,state
    FROM dept_empl
    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_empl AS
                    (
                         SELECT id,
name, surname,
                          lastname,
state, dept_id
                           FROM
test_ora_pg.dept_employees
                          WHERE state =
1),
                     dept AS
                    (SELECT id deptid,
parent_id,
                       name deptname
                       FROM test_ora_
pg.department
                    )
                SELECT dept_empl
.*,dept.*
                 FROM dept_empl, dept
                 WHERE dept_empl
.dept_id = dept.deptid
              ) L00P
              NULL;
            END LOOP;
```

Instrução do MySQL

```
CREATE VIEW TEST_ORA_PG. `P_WIT
H_SELECT_REGULAR_MULT_01$dept_empl
 `(id, name, surname, lastname, state,
 dept_id)
AS
(SELECT id, name, surname, lastname,
 state, dept_id
   FROM test_ora_pg.dept_employees
  WHERE state = 1);
CREATE VIEW TEST_ORA_PG. `P_WIT
H_SELECT_REGULAR_MULT_01$dept
 `(deptid, parent_id,deptname)
AS
(SELECT id deptid, parent_id, name
 deptname
   FROM test_ora_pq.department);
CREATE PROCEDURE test_ora_pg.P_WITH
_SELECT_REGULAR_MULT_01()
BEGIN
    DECLARE var$ID DOUBLE;
    DECLARE var$NAME VARCHAR (30);
    DECLARE var$SURNAME VARCHAR (30);
    DECLARE var$LASTNAME VARCHAR (30);
    DECLARE var$STATE DOUBLE;
    DECLARE var$DEPT_ID DOUBLE;
    DECLARE var$deptid DOUBLE;
    DECLARE var$PARENT_ID DOUBLE;
    DECLARE var$deptname VARCHAR
 (200);
    DECLARE done INT DEFAULT FALSE;
    DECLARE cur CURSOR FOR SELECT
        dept_empl.*, dept.*
        FROM TEST_ORA_PG. `P_WIT
H_SELECT_REGULAR_MULT_01$dept_empl
          ` AS dept_empl,
             TEST_ORA_PG. `P_WIT
H_SELECT_REGULAR_MULT_01$dept
            AS dept
```

```
Instrução do Oracle
                                              Instrução do MySQL
                                                        WHERE dept_empl.DEPT_ID =
                                                 dept.DEPTID;
                                                    DECLARE CONTINUE HANDLER FOR NOT
                                                 FOUND
                                                        SET done := TRUE;
                                                    OPEN cur;
                                                    read_label:
                                                    L00P
                                                    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()
```

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
              ) L00P
         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
	LOOP FETCH cur INTO var\$ID;
	<pre>IF done THEN LEAVE read_label; END IF;</pre>
	BEGIN END; END LOOP; CLOSE cur; END;

Migração do banco de dados Oracle para o Amazon RDS for Oracle com AWS Schema Conversion Tool

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

- 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.
- AWS SCT suporta a conversão de espaços de tabela Oracle em uma instância de banco de dados Amazon RDS for 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 oferece suporte ao Oracle Managed Files (OMF) somente para arquivos de dados, arquivos de log e arquivos de controle. AWS SCT cria os arquivos de dados necessários durante a conversão.
- AWS SCT pode converter funções e privilégios em 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.

- AWS SCT suporta a conversão de trabalhos Oracle em trabalhos que podem ser executados no Amazon RDS for 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 <u>Oracle Spatial and Graph</u> 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.
- AWS SCT suporta a conversão de objetos Oracle DBMS_SCHEDULER ao migrar para o Amazon RDS for Oracle. O relatório AWS SCT de avaliação indica se um objeto de cronograma 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 AWS SCT avaliação 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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.

Conectando-se aos bancos de dados PostgreSQL com o AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código de banco de dados e 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:

- CONECTE-SE AO BANCO DE DADOS < database_name >
- USO NO ESQUEMA < database_name >
- SELECIONE EM TODAS AS TABELAS NO ESQUEMA <database_name>
- SELECIONE EM TODAS AS SEQUÊNCIAS NO ESQUEMA < 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha PostgreSQL e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	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 PostgreSQ L de origem usando IPv6 um protocolo de endereço. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.
	[2001:db8:ffff:ffff:ffff:ffff:fffe]
Porta do servidor	Digite a porta usada para se conectar ao servidor de banco de dados de origem.
Banco de dados	Digite o nome do banco de dados PostgreSQL.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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 você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.

Parameter	Ação
Usar SSL	 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: 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.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 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 *.*
- INSIRA, ATUALIZE EM AWS_POSTGRESQL_EXT. *
- INSIRA, ATUALIZE E EXCLUA EM AWS_POSTGRESQL_EXT_DATA. *
- CRIE TABELAS TEMPORÁRIAS EM 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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.

Conectando-se aos bancos de dados SAP com o AWS Schema Conversion Tool

Você pode usar 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

Bancos de dados SAP Versão 1.0.672 134

- 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.
- 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.systypes to min_privs
grant select on dbo.systypes to min_privs
```

Privilégios do SAP ASE Versão 1.0.672 135

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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha SAP ASE e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

- 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 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.
 - 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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.
Banco de dados	Insira o nome do banco de dados do SAP ASE.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.

Parameter	Ação
	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 você fechar o AWS SCT projeto e reabri-lo, será solicitad a a senha para se conectar ao banco de dados de origem, conforme necessário.
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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.

Parameter	Ação
Caminho do driver SAP ASE	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 *.*
- INSIRA, ATUALIZE EM AWS_SAPASE _EXT. *

CRIE TABELAS TEMPORÁRIAS EM 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 TRIGGER 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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 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. 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, AWS SCT converte os nomes dos objetos, variáveis e parâmetros do banco de dados em 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 AWS SCT, é necessário o 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua 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 não AWS SCT 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 AWS SCT de conversão do SAP ASE para PostgreSQL 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. 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, AWS SCT converte os nomes dos objetos, variáveis e parâmetros do banco de dados em 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, 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 isso AWS SCT gere nomes exclusivos para todos os seus índices, selecione Gerar nomes exclusivos para índices.

Conecte Microsoft SQL Servers com AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código de banco de dados e 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



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

Você pode usar 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 o 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
- Migrando do SQL Server para o PostgreSQL com AWS Schema Conversion Tool
- Migração do SQL Server para o Amazon RDS for SQL Server com AWS Schema Conversion Tool

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 VIEW DEFINITION privilégio permite que usuários com acesso público vejam as definições de objetos. AWS SCT usa o VIEW DATABASE STATE privilégio para verificar os recursos 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

AWS SCT usa o VIEW SERVER STATE privilégio 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 *sex_name* pelo nome do usuário que você concedeu com os privilégios anteriores. Para obter mais informações, consulte <u>Adicionar um usuário à função de SQLAgent usuário</u> 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 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.

- 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\x86 (se estiver executando um sistema operacional de 32 bits) ou D\lib\JDBC4.1\enu\auth\x64 (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 pasta da biblioteca JDBC, você tenha somente o SQLJDBC4 arquivo.jar.
 Remova qualquer outro arquivo sqljdbc*.jar dessa pasta (ou copie em outra pasta). Se você estiver adicionando o driver como parte do seu programa, certifique-se de adicionar somente SQLJDBC4 o.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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.
	Você pode se conectar ao seu banco de dados SQL Server de origem usando um protocolo IPv6 de endereço. Para fazer isso, certifique-se de usar colchetes para inserir o endereço IP, conforme mostrado no exemplo a seguir.
	[2001:db8:ffff:ffff:ffff:fffe]
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 banco de dados SQL Server. Para encontrar o nome da instância, execute a consulta

Parameter	Ação
	SELECT @@servername; 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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 evitar a exposição da senha do seu banco de dados de origem, AWS SCT não armazena a senha por padrão. Se você fechar o AWS SCT projeto e reabri-lo, será solicitada a senha para se conectar ao banco de dados de origem, conforme necessário.
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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.

Parameter	Ação
Caminho do driver Sql Server	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> . 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .
Biblioteca de autenticação do Windows	<pre>Insira o caminho para o arquivo sqljdbc_auth.dll . Por padrão, esse arquivo é instalado no seguinte local: <installation directory="" driver="" jdbc="" of="" the="">sqljdbc_<version> \<language> \auth\</language></version></installation></pre>

- Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu banco de dados de origem.
- 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 <u>Usando pacotes de extensão com AWS Schema Conversion</u> Tool.

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 *.*
- INSIRA, ATUALIZE EM AWS_SQLSERVER _EXT. *
- INSIRA, ATUALIZE E EXCLUA EM AWS_SQLSERVER _EXT_DATA. *
- CRIE TABELAS TEMPORÁRIAS EM 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *your_password* substitua 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, AWS SCT escolha Configurações e, em seguida, escolha Configurações 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 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. 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. AWS SCT cria tabelas temporárias e um procedimento adicional para emular esse recurso.
 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, AWS SCT pode emular a MERGE declaração durante a conversão usando a INSERT ON DUPLICATE KEY cláusula e as UPDATE FROM and DELETE FROM declarações.

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 G0T0 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 G0T0 são ignoradas, e o processamento continua no rótulo. As instruções G0T0 e os rótulos podem ser usados em qualquer lugar em um procedimento, lote ou bloco de instruções. Você também pode aninhar instruções G0T0.

O MySQL não usa instruções G0T0. Quando AWS SCT converte o código que contém uma G0T0 instrução, ele converte a instrução para usar uma instrução BEGIN...END ouL00P...END L00P. Você pode encontrar exemplos de como AWS SCT converte G0T0 declarações na tabela a seguir.

```
Instrução do SQL Server
                                                Instrução do MySQL
 BEGIN
                                                  BEGIN
                                                   label1:
    statement1;
                                                   BEGIN
    GOTO label1;
                                                     statement1;
    statement2;
                                                     LEAVE label1;
    label1:
                                                     statement2;
    Statement3;
                                                     . . . .
                                                   END;
 END
                                                     Statement3;
```

Instrução do SQL Server Instrução do MySQL **END BEGIN BEGIN** statement1; statement1; label1: label1: L00P statement2; statement2; GOTO label1; statement3; ITERATE label1; LEAVE label1; END LOOP; statement4; statement3; **END** statement4; **END BEGIN BEGIN** statement1; statement1; label1: label1: statement2; **BEGIN** statement2; statement3; statement3; statement4; statement4; **END** END; **END**

 O MySQL não oferece suporte a funções com valor de tabela de várias instruções. AWS SCT simula funções com valor de tabela durante uma conversão criando tabelas temporárias e reescrevendo instruções para usar essas tabelas temporárias.

Migrando do SQL Server para o PostgreSQL com AWS Schema Conversion Tool

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 <u>Usando</u> pacotes de extensão com AWS Schema Conversion Tool.

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 AWS SCT de extensão para emular o SQL Server Agent no PostgreSQL
- Usando um pacote AWS SCT de extensão 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 AWS SCT, é necessário o 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua 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 não AWS SCT 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 AWS SCT de conversão do SQL Server para PostgreSQL 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. 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 isso 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 AWS SCT, podem converter procedimentos em funções. 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, você tem AWS SCT a opção de gerar nomes de índice exclusivos se seus nomes de índice 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. Ao AWS SCT converter o código que contém uma instrução GOTO, ele converte a instrução para usar uma instrução BEGIN... END ou LOOP... END LOOP. Você pode encontrar exemplos de como 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
 BEGIN
                                                    BEGIN
                                                     label1:
    statement1;
                                                     BEGIN
     . . . .
    GOTO label1;
                                                       statement1;
    statement2;
                                                       . . . .
                                                       EXIT label1;
    label1:
                                                       statement2;
    Statement3;
                                                       . . . .
                                                     END;
 END
                                                       Statement3;
                                                    END
 BEGIN
                                                    BEGIN
    statement1;
                                                       statement1;
    label1:
                                                       label1:
    statement2;
                                                       L<sub>00</sub>P
                                                        statement2;
    GOTO label1;
    statement3;
                                                        CONTINUE label1;
                                                        EXIT label1;
     . . . .
    statement4;
                                                       END LOOP;
                                                        statement3;
 END
                                                        statement4;
                                                         . . . .
                                                    END
```

```
Instrução do SQL Server
                                                 Instrução do PostgreSQL
 BEGIN
                                                  BEGIN
    statement1;
                                                      statement1;
    label1:
                                                      label1:
    statement2;
                                                      BEGIN
                                                       statement2;
    statement3;
                                                       statement3;
    statement4;
                                                       statement4;
 END
                                                       . . . .
                                                      END;
                                                  END
```

- O PostgreSQL não suporta uma instrução MERGE. 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.
- AWS SCT pode adicionar acionadores de banco de dados à árvore de objetos quando o Amazon RDS é o destino.
- AWS SCT pode adicionar gatilhos em nível de servidor à árvore de objetos quando o Amazon RDS é 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 acionamento de DML. AWS SCT pode converter o uso dessas tabelas em declarações de gatilho DML.
- AWS SCT pode adicionar servidores vinculados à árvore de objetos quando o Amazon RDS é 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. <pattern character><expression character varying>Ao converter do SQL Server para o Amazon RDS for AWS SCT PostgreSQL, substitui o código do aplicativo que usa PATINDEX por aws_sqlserver_ext.patindex (,).
- 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. Você também pode usar um tipo de tabela definido pelo usuário para declarar variáveis de tabela que você deseja usar em um lote ou no corpo de um procedimento ou função armazenado. AWS SCT emulou esse tipo no PostgreSQL criando uma tabela temporária.

Ao converter do SQL Server para o PostgreSQL AWS SCT, converte objetos do sistema 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_ESQUEMAS
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABELAS
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS

Casos de uso do MS SQL Server	Substituição do PostgreSQL
SYS.TYPES	AWS_SQLSERVERTIPOS_EXT.SYS
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUNAS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_CHAVES ESTRANGEIRAS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYS CHAVES ESTRANGEIRAS
SYS.FOREIGN_KEY_CO	AWS_SQLSERVER_EXT.SYS_COLUNAS_CHAVE_ESTRANGEI RAS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVERRESTRIÇÕES _EXT.SYS_KEY_
SYS.IDENTITY_COLUM NS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVERPROCEDIMENTOS _EXT.SYS
SYS.INDEXES	AWS_SQLSERVERÍNDICES _EXT.SYS
SYS.SYSINDEXES	AWS_SQLSERVERÍNDICES _EXT.SYS_SYS
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_TODOS_OBJETOS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_BANCOS DE DADOS
INFORMATION_SCHEMA .SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA

Casos de uso do MS SQL Server	Substituição do PostgreSQL
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_CONST RAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENT IAL_CONSTRAINTS
INFORMATION_SCHEMA .TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CO NSTRAINTS
INFORMATION_SCHEMA .KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLU MN_USAGE
INFORMATION_SCHEMA .CONSTRAINT_TABLE_ USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_TABLE_USAGE
INFORMATION_SCHEMA .CONSTRAINT_COLUMN _USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_COLUMN_USAGE
INFORMATION_SCHEMA .ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROTINAS
SYS.SYSPROCESSES	AWS_SQLSERVERPROCESSOS _EXT.SYS_SYS
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

Usando um pacote AWS SCT de extensão 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, AWS SCT cria um pacote de extensão. Este pacote de extensão usa AWS Lambda e Amazon CloudWatch. AWS Lambda implementa a interface que você usa para gerenciar agendamentos e executar trabalhos. A Amazon CloudWatch mantém as regras de programação.

AWS Lambda e a 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.

AWS SCT usa uma função Lambda para controlar e executar trabalhos. A CloudWatch regra inicia a execução do trabalho e fornece as informações necessárias para iniciar o trabalho. Quando a CloudWatch regra é acionada, ela inicia a função 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, AWS SCT o pacote de extensões 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.

sysschedules

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 agendamento na 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 agendamento para um trabalho na Amazon CloudWatch e define a meta 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 agendamento na 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 agendamento para o trabalho especificado da 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_cmdexec_success_code integer = 0,
par_on_success_action smallint = 1,
par_on_success_step_id integer = 0,
par_on_fail_action smallint = 2,
par_on_fail_step_id integer = 0,
par_server varchar = NULL::character varying,
par_database_name varchar = NULL::character varying,
par_database_user_name varchar = NULL::character varying,
par_retry_attempts integer = 0,
par_retry_interval integer = 0,
par_os_run_priority integer = 0,
par_output_file_name varchar = NULL::character varying,
```

```
par_flags integer = 0,
par_proxy_id integer = NULL::integer,
par_proxy_name varchar = NULL::character varying,
inout par_step_uid char = NULL::bpchar,
out returncode integer
```

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 <u>documentação técnica da Microsoft</u>.

```
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_end_time := 0);
```

Use exemplos de casos para emular o SQL Server Agent no PostgreSQL

Se o código-fonte do banco de dados usa o SQL Server Agent para executar trabalhos, você pode usar o pacote de extensão SQL Server para PostgreSQL AWS SCT para converter esse código em PostgreSQL. O pacote de extensão usa AWS Lambda funções para emular o comportamento do SQL Server Agent.

Você pode criar uma nova AWS Lambda função ou registrar uma função existente.

Para criar uma nova AWS Lambda função

 Em 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 AWS Lambda função.
 - 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 AWS Lambda função e, em seguida, escolha Avançar.

Para registrar uma AWS Lambda função 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.

 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 AWS SCT de extensão 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. Esse pacote de extensão usa AWS Lambda o Amazon Simple Email Service (Amazon SES). 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, 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 ison
```

Um procedimento interno que gera solicitações JSON para AWS Lambda funções.

```
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>',
   '',
   'Work Order IDProduct ID',
   '',
   'NameOrder QtyDue Date',
   'Expected Revenue<',
   '</th>
// ''
);
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.

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-fonte do banco de dados usa o SQL Server Database Mail para enviar e-mails, você pode usar o pacote de AWS SCT extensão para converter esse código em PostgreSQL.

Para enviar um e-mail do seu banco de dados PostgreSQL

- 1. Crie e configure sua AWS Lambda função.
- 2. Aplique o pacote AWS SCT de extensão.
- 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.
- 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 plpqsql;
```

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:XXXXXXXXXXXXXXXXIfunction:func_test',
   par_mailserver_type := 'AWSLAMBDA'
);
```

Migração do SQL Server para o Amazon RDS for SQL Server com AWS Schema Conversion Tool

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

 AWS SCT pode converter o SQL Server Agent para fornecer agendas, alertas e trabalhos em uma instância de banco de dados Amazon RDS for 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. Ao converter o código do aplicativo SQL Server que usa servidores vinculados, 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 AWS SCT avaliação 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, então recomendamos que você instale o SQL Server em uma instância da 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 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino. Por fim, *your_password* substitua por uma senha segura.

Origens de data warehouse para AWS Schema Conversion Tool

AWS SCT pode converter esquemas dos seguintes data warehouses de origem em um destino compatível. Para obter informações sobre permissões, conexões e o que AWS SCT pode ser convertido para uso com o banco de dados de destino ou o data warehouse, consulte os detalhes a seguir.

Tópicos

- Conectando o Amazon Redshift com o AWS Schema Conversion Tool
- Conectando o Azure Synapse Analytics com AWS Schema Conversion Tool
- Conectando-se ao Google BigQuery com AWS Schema Conversion Tool
- Conectando o banco de dados Greenplum com AWS Schema Conversion Tool
- · Conectando-se ao Netezza com AWS Schema Conversion Tool
- · Conectando o Oracle Data Warehouse com AWS SCT
- Conectando-se a um data warehouse do Snowflake com AWS Schema Conversion Tool
- Conectando-se a um SQL Server Data Warehouse com o AWS Schema Conversion Tool
- Como se conectar a um data warehouse do Teradata com o AWS Schema Conversion Tool.
- Conectando os bancos AWS Schema Conversion Tool de dados Vertica

Conectando o Amazon Redshift com o AWS Schema Conversion Tool

Você pode usar AWS SCT para otimizar seu cluster do Amazon Redshift. AWS SCT fornece recomendações sobre a seleção de chaves de distribuição e classificação para seu cluster do

Data warehouses Versão 1.0.672 192

Amazon Redshift. Você pode considerar o projeto de otimização do Amazon Redshift como um AWS SCT projeto 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:

- USO NO ESQUEMA <schema_name>
- SELECIONE EM TODAS AS TABELAS NO ESQUEMA <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 <schema_name> espaço reservado pelo nome do esquema de origem.

Para obter os privilégios necessários para o Amazon Redshift como destino, consulte a <u>Permissões</u> 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

 Para inserir as informações de conexão para o banco de dados de origem do Amazon Redshift, use as instruções a seguir:

Parameter	Ação
Nome do servidor	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.
Banco de dados	Insira o nome do banco de dados do Amazon Redshift.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 necessári o.

Parameter	Ação
Usar SSL	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:
	 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 certifica dos. Para que esse local apareça aqui, certifique-se de adicioná-lo nas configurações globais.
	Para obter mais informações sobre o suporte a SSL para o Amazon Redshift, consulte Configurar opções de segurança para conexões.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 Redshift	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais.

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 em e AWS SCT, 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 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

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. 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, AWS SCT exibirá uma mensagem de aviso. Além disso, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift até que o número de tabelas atinja o limite.

Para escolher a estratégia de migração.

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.

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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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.

Conectando o Azure Synapse Analytics com AWS Schema Conversion Tool

Você pode usar 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

- No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

 Para inserir as informações de conexão do data warehouse do Azure Synapse Analytics manualmente, use as instruções a seguir:

Parameter	Ação
Nome do servidor	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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 necessári o.
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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 e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Azure Synapse e, em seguida, escolha Azure Synapse — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão do Azure Synapse Analytics para o Amazon Redshift.

As configurações AWS SCT de conversão do Azure Synapse Analytics para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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.

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 e, AWS SCT 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. AWS SCT exibe as

configurações de otimização de conversão para a conversão do Azure Synapse Analytics para o Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Azure Synapse Analytics para o Amazon Redshift 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 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando-se ao Google BigQuery com AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo BigQuery para o Amazon Redshift.

Privilégios BigQuery como fonte

Para usar um BigQuery data warehouse como fonte 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 BigQuery Admin função fornece permissões para gerenciar todos os recursos do projeto. AWS SCT usa essa função para carregar seus BigQuery metadados no projeto de migração.

A função Storage Admin concede controle total sobre objetos e buckets de dados. Você pode encontrar essa função emCloud Storage. AWS SCT usa essa função para extrair seus dados 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 BigQuery API, 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 e BigQuery, em seguida, escolha BigQuery Administrador.
- 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.
- 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 BigQuery data warehouse, AWS SCT use a pasta 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.

Conectando-se BigQuery como fonte

Use o procedimento a seguir para se conectar ao seu BigQuery projeto de origem com AWS Schema Conversion Tool o.

Para se conectar a um data warehouse de BigQuery origem

1. No AWS Schema Conversion Tool, escolha Adicionar fonte.

- 2. Escolha BigQuerye, em seguida, escolha Avançar.
 - A caixa de diálogo Adicionar origem é exibida.
- 3. Em Nome da conexão, insira um nome para seu BigQuery projeto. 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 BigQuery como fonte.
- Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu BigQuery projeto de origem.
- 6. Escolha Connect para se conectar ao seu BigQuery projeto de origem.

Limitações no uso BigQuery como fonte para AWS SCT

As seguintes limitações se aplicam ao uso BigQuery como fonte para AWS SCT:

- AWS SCT não suporta a conversão de subconsultas em funções analíticas.
- Você não pode usar AWS SCT para converter BigQuery SELECT AS STRUCT SELECT AS VALUE declarações.
- AWS SCT não suporta a conversão dos seguintes tipos de funções:
 - Agregado aproximado
 - Bit
 - Depuração
 - Consulta federada
 - Geografia
 - Hash
 - Matemáticas
 - Net
 - Agregado estatístico
 - UUID
- AWS SCT fornece suporte limitado para a conversão de funções de string.
- AWS SCT não suporta a conversão de UNNEST operadores.
- Você não pode converter operações de junção correlacionadas em AWS SCT.

- AWS SCT não suporta a conversão de OFFSET cláusulas QUALIFY WINDOWLIMIT,, e.
- Você não pode usar AWS SCT para converter expressões de tabela comuns recursivas.
- AWS SCT não suporta a conversão de INSERT declarações com subconsultas dentro VALUES de cláusulas.
- AWS SCT não suporta a conversão de UPDATE declarações para campos aninhados e registros repetidos.
- Você não pode usar AWS SCT para converter STRUCT nem tipos de ARRAY dados.

BigQuery para as configurações de conversão do Amazon Redshift

BigQuery Para editar as configurações de conversão do Amazon Redshift, escolha Configurações em e, em seguida AWS SCT, escolha Configurações de conversão. Na lista superior, escolha Google e BigQuery, em seguida, escolha Google BigQuery — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão BigQuery para o Amazon Redshift.

BigQuery para as configurações de conversão do Amazon Redshift em AWS SCT Inclua 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.
 - Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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.

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.

BigQuery para as configurações de otimização de conversão do Amazon Redshift

BigQuery Para editar as configurações de otimização de conversão do Amazon Redshift, escolha Configurações em e AWS SCT, em seguida, escolha Configurações de conversão. Na lista superior, escolha Google e BigQuery, em seguida, escolha Google BigQuery — Amazon Redshift. No painel esquerdo, escolha Estratégias de otimização. AWS SCT exibe as configurações de otimização de conversão para BigQuery a conversão para o Amazon Redshift.

BigQuery para as configurações de otimização de conversão do Amazon Redshift, AWS SCT inclua 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando o banco de dados Greenplum com AWS Schema Conversion Tool

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo do Greenplum Database 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:

- CONECTE-SE AO BANCO DE DADOS database_name
- USO NO ESQUEMA <schema_name>
- SELECIONE ATIVADO <schema_name>. <table_name>
- SELECIONE NA SEQUÊNCIA <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.
- table_nameSubstitua pelo nome da tabela de origem.
- sequence_nameSubstitua pelo nome do nome da sequência.

Como se conectar ao banco de dados Greenplum como origem

Use o procedimento a seguir para se conectar ao seu banco de dados de origem do Greenplum com. AWS SCT

bancos de dados Greenplum Versão 1.0.672 210

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

- 1. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

 Para inserir as credenciais do banco de dados do Greenplum de origem manualmente, use as instruções a seguir:

Parameter	Ação
Nome do servidor	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.
Banco de dados	Digite o nome do banco de dados do Greenplum.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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

bancos de dados Greenplum Versão 1.0.672 211

Parameter	Ação
	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 necessári o.
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 do banco de dados Greenplum	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais.

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Greenplum e, em seguida, escolha Greenplum — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para a conversão do Greenplum para o Amazon Redshift.

As configurações AWS SCT de conversão do Greenplum para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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.

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 e, AWS SCT 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. AWS SCT exibe as configurações de otimização de conversão da Greenplum para a conversão do Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Greenplum para o Amazon Redshift incluem opções para o seguinte:

bancos de dados Greenplum Versão 1.0.672 214

 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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.

bancos de dados Greenplum Versão 1.0.672 215

 Em Valor limite distorcido, insira a porcentagem (0—100) de um valor distorcido para uma coluna. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando-se ao Netezza com AWS Schema Conversion Tool

Você pode usar 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
- lista em <database name>
- lista em <schema name>
- lista em <database_name> .all.table
- lista na <database_name> tabela.all.external
- lista em <database name> .all.view
- lista na visualização < database_name > .all.materialized
- lista em <database_name> .all.procedure

- lista em <database name> .all.sequence
- lista em <database name> .all.function
- lista em <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.

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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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

Parameter	Ação
	novamente, você será solicitado a informar a senha para se conectar ao banco de dados de origem, conforme necessári o.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais.

- Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu banco de dados de origem.
- 6. 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. AWS SCT O agente extrai seus dados e os carrega no bucket do Amazon S3. Em seguida, você pode usar AWS SCT para copiar os 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 AWS SCT dados. 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, <code>history_database_name</code> substitua pelo nome do seu banco de dados de histórico. Em seguida, <code>load_user</code> substitua pelo nome do usuário que você definiu para carregar os dados do histórico no banco de dados. Em seguida, <code>histdb_owner</code> substitua 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, <code>your_password</code> substitua por uma senha segura.

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. Em seguida, *load_user* substitua pelo nome do usuário que você definiu para carregar os dados do histórico no banco de dados. Em seguida, *your_password* substitua por uma senha segura.

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, *history_database_name* substitua pelo nome do seu banco de dados de histórico. Em seguida, *your_user* substitua 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, *N* substitua 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 <u>Como</u> 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.
- Crie sua tarefa da CDC. Para obter mais informações, consulte <u>Criando, executando e</u> 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 AWS SCT tarefa 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 e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Netezza e, em seguida, escolha Netezza — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão de Netezza para Amazon Redshift.

As configurações AWS SCT de conversão do Netezza para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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.

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 e, AWS SCT 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. AWS SCT exibe as configurações de otimização de conversão para a conversão do Netezza para o Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Netezza para o Amazon Redshift 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando o Oracle Data Warehouse com AWS SCT

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo do Oracle Data Warehouse para o Amazon Redshift ou o Amazon Redshift e usá-los em combinação AWS Glue.

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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

 Para inserir manualmente as informações de conexão do data warehouse do Oracle de origem, use as instruções a seguir:

Parameter	Ação
Tipo	Escolha o tipo de conexão ao seu banco de dados. Dependendo do tipo, forneça as informações adicionais seguintes: • SID
	 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','inst ance_name') AS SID FROM dual;</pre>
	Nome do serviço
	 Nome do servidor: o nome DNS ou o endereço IP do servidor de banco de dados de origem.

Parameter	Ação
	 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 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 Identificador de conexão TNS: o identificador das informações de conexão TNS registradas.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 necessári o.

Parameter	Ação
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 Oracle	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais.

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 Oracle Data Warehouse para o Amazon Redshift, escolha Configurações em e AWS SCT, em seguida, escolha Configurações de conversão. Na lista superior, escolha Oracle e, em seguida, escolha Oracle — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão do Oracle Data Warehouse para o Amazon Redshift.

As configurações de conversão do Oracle Data Warehouse para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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 T0_CHAR, T0_DATE e
T0_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 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.

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 Oracle Data Warehouse para o Amazon Redshift, escolha Configurações em e AWS SCT, 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. AWS SCT exibe as configurações de otimização de conversão para conversão do Oracle Data Warehouse para o Amazon Redshift.

As configurações de otimização de conversão do Oracle Data Warehouse para o Amazon Redshift 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando-se a um data warehouse do Snowflake com AWS Schema Conversion Tool

Você pode usar 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 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 select on future external tables in schema db_name.schema_name to role role_name;
```

```
grant usage on all sequences in schema <code>db_name.schema_name</code> to role <code>role_name;</code> grant usage on future sequences in schema <code>db_name.schema_name</code> to role <code>role_name;</code> grant usage on all functions in schema <code>db_name.schema_name</code> to role <code>role_name;</code> grant usage on future functions in schema <code>db_name.schema_name</code> to role <code>role_name;</code> grant usage on all procedures in schema <code>db_name.schema_name</code> to role <code>role_name;</code> grant usage on future procedures in schema <code>db_name.schema_name</code> to role <code>role_name;</code> create user <code>min_privs</code> <code>password='real_user_password'</code>

<code>DEFAULT_ROLE = role_name</code> <code>DEFAULT_WAREHOUSE = 'datawarehouse_name';</code> grant role <code>role_name</code> to user <code>min_privs;</code>
```

No exemplo anterior, substitua os espaços reservados conforme a seguir:

- Substitua role_name pelo nome de uma função com privilégios somente leitura.
- Substitua <u>db_name</u> pelo nome do banco de dados de origem.
- Substitua schema_name pelo nome do esquema de origem.
- Substitua datawarehousename 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha Snowflake e, em seguida, escolha Avançar.

A caixa de diálogo Adicionar origem é exibida.

 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

• Para inserir manualmente as informações de conexão do data warehouse do Snowflake de origem, use as instruções a seguir:

Parameter	Ação
Nome do servidor	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.
Banco de dados	Insira o nome do banco de dados do Snowflake.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	AWS SCT armazena sua senha em um formato criptogra fado somente se você a solicitar explicitamente.
Usar SSL	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:
	 Caminho da chave privada: a localização de uma chave privada.

Parameter	Ação
	 Senha: A frase secreta da chave privada. Para obter mais informações sobre o suporte a SSL para Snowflake, consulte <u>Configurar opções de segurança para conexões</u>.
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certifica dos SSL e senhas de 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	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> . 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

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

Limitações do Snowflake como origem

A seguir estão as limitações ao usar o Snowflake como fonte para: 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 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	VARCHAR(MAX)
Caracteres Unicode de até 16.777.216 bytes; até 4 bytes por caractere.	
TEXT(p)	Se p for =< 65.535 então, VARCHAR(p)
Caracteres Unicode de até 65.535 bytes; até 4 bytes por caractere.	
TEXT(p)	Se p for => 65.535 e =< 16.777.216 então,
Caracteres Unicode de até 16.777.216 bytes; até 4 bytes por caractere.	VARCHAR(MAX)
BINARY	VARCHAR(MAX)
Caracteres de byte único de até 8.388.608 bytes; 1 byte por caractere.	
BYNARY(p)	VARCHAR(p)
Caracteres de byte único de até 65.535 bytes; 1 byte por caractere.	
BYNARY(p)	VARCHAR(MAX)
Caracteres de byte único de até 8.388.608 bytes; 1 byte por caractere.	
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 23:59:59.9999999999.	
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 e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Snowflake e, em seguida, escolha Snowflake — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão do Snowflake para o Amazon Redshift.

As configurações AWS SCT de conversão do Snowflake para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Snowflake Versão 1.0.672 240

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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.

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 e, AWS SCT 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. AWS SCT exibe as configurações de otimização de conversão para a conversão do Snowflake para o Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Snowflake para o Amazon Redshift incluem opções para o seguinte:

Snowflake Versão 1.0.672 241

 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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.

Snowflake Versão 1.0.672 242

 Em Valor limite distorcido, insira a porcentagem (0—100) de um valor distorcido para uma coluna. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando-se a um SQL Server Data Warehouse com o AWS Schema Conversion Tool

Você pode usar 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 usá-los em combinação. AWS Glue

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
- SELECIONE NO ESQUEMA: <schema_name>

No exemplo anterior, substitua o < source_schema > espaço reservado 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 2. Escolha Microsoft SQL Server e, em seguida, escolha Avançar.
 - A caixa de diálogo Adicionar origem é exibida.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

 Para inserir manualmente as informações de conexão do data warehouse de origem do Microsoft SQL Server, use as instruções a seguir:

Parameter	Ação
Nome do servidor	Digite o Serviço de Nome de Domínio (DNS) ou o endereço IP do servidor de banco de dados de origem.

Parameter	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	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem.
	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 necessári o.
Usar SSL	 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: 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	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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.

Parameter	Ação
Caminho do driver SQL Server	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 informaçõ es, consulte <u>Armazenamento de caminhos do driver nas configurações globais</u> .

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 SQL Server Data Warehouse para o Amazon Redshift, escolha Configurações em e AWS SCT, em seguida, escolha Configurações de conversão. Na lista superior, escolha Microsoft SQL Server e, em seguida, escolha Microsoft SQL Server — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão do SQL Server Data Warehouse para o Amazon Redshift.

As configurações de conversão do SQL Server Data Warehouse para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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.

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 SQL Server Data Warehouse para o Amazon Redshift, escolha Configurações em e AWS SCT, 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. AWS SCT exibe as configurações de otimização de conversão para conversão do SQL Server Data Warehouse para o Amazon Redshift.

As configurações de otimização de conversão do SQL Server Data Warehouse para o Amazon Redshift 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Como se conectar a um data warehouse do Teradata com o AWS Schema Conversion Tool.

Você pode usar AWS SCT para converter esquemas, objetos de código e código de aplicativo da Teradata para o Amazon Redshift ou Amazon Redshift e usá-los em combinação. AWS Glue

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
- SELECIONE ATIVADO <source_database>
- CRIAR PROCEDIMENTO EM <source_database>

No exemplo anterior, substitua o espaço reservado < source_database > pelo nome do banco de dados de origem.

AWS SCT requer o privilégio CREATE PROCEDURE para executar HELP PROCEDURE em todos os procedimentos no banco de dados de origem. AWS SCT não usa esse privilégio para criar novos objetos no banco de dados 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.
- 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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Connection name (Nome da conexão)	Insira um nome para seu banco de dados. AWS SCT exibe esse nome na árvore no painel esquerdo.
Nome do servidor	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.
Banco de dados	Digite o nome do banco de dados Teradata.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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 necessári o.

Parameter	Ação
Armazenar senha	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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 trocar com o banco de dados. Se você escolher essa opção, o número da porta 443 será usado para transferir dados criptografados entre AWS SCT e seu banco de dados Teradata.
Caminho do driver Teradata	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool.</u> 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 <u>Armazenamento de caminhos do driver nas configurações globais</u> .

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

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
```

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
LdapClientTlsRegCert: 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 GDO"
```

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 armazém de dados Teradata de origem, AWS SCT usa estatísticas para otimizar seu armazém de dados 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 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, *table_name* substitua pelo nome da 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$S$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$S$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 Teradata, você pode coletar estatísticas em seu AWS SCT projeto. 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.

```
.0S IF EXIST column-stats-tera.csv del /F column-stats-tera.csv
.0S IF EXIST table-stats-tera.csv del /F table-stats-tera.csv
.0S IF EXIST column-skew-script-tera.csv del /F column-skew-script-tera.csv
.0S IF EXIST column-skew-stats-tera.csv del /F column-skew-stats-tera.csv
.0S 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))
 11 '";' 11
    '"' || 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_cour
FROM (select databasename, tablename
        from DBC.tablesv
        where tablekind IN ('T','0')
        and databasename = 'your_database_name'
left join
        (select DatabaseName, TableName, max(RowCount) RowCount
        from dbc.tableStatsv
        group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
        (SELECT r.ChildDB as DatabaseName,
        r.ChildTable as TableName,
        COUNT(DISTINCT r.ParentTable) reference_count
        FROM DBC.All_RI_ChildrenV r
        GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
        (SELECT r.ParentDB as DatabaseName,
        r.ParentTable as TableName,
        COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
        FROM DBC.All_RI_ParentsV r
        GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
        (select databasename, tablename,
        sum(currentperm)/1024/1024 as size_in_mb
        from dbc.TableSizeV
```

```
group by 1,2) w
on w.databasename = c.databasename
and w.tablename = c.tablename
WHERE COALESCE(r.stat_fk_dep_count,0) + COALESCE(CAST(p.RowCount AS BIGINT),0) +
COALESCE(s.reference_count,0) > 0;
.EXPORT RESET
.EXPORT REPORT FILE = column-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000
    '"' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) || '";' ||
SELECT
 '"' || OREPLACE(COALESCE(trim(tv.DatabaseName), ''), '"', '""') || '";' ||
     '"' || OREPLACE(COALESCE(trim(tv.TableName), ''), '"', '""') || '";' ||
 '"' || OREPLACE(COALESCE(trim(tv.columnname), ''), '"', '""') || '";' ||
                         '"' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) ||
 '";' ||
 '"' || CAST(current_timestamp AS VARCHAR(19)) || '"' (TITLE
'"database_name";"table_name";"column_name";"cardinality";"current_ts"')
FROM dbc.columnsv tv
LEFT JOIN
SELECT
  c.DatabaseName AS DATABASE_NAME,
  c.TABLENAME AS TABLE_NAME,
  c.ColumnName AS COLUMN_NAME,
 c.UniqueValueCount AS CARD
FROM dbc.tablestatsv c
WHERE c.DatabaseName = 'your_database_name'
AND c.RowCount <> 0
) t2
ON tv.DATABASENAME = t2.DATABASE_NAME
AND tv.TABLENAME = t2.TABLE_NAME
AND tv.COLUMNNAME = t2.COLUMN_NAME
WHERE t2.card > 0;
.EXPORT RESET
.EXPORT REPORT FILE = column-skew-script-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000
```

```
SELECT
'SELECT CAST(''"' || TRIM(c.DatabaseName) || '";"' || TRIM(c.TABLENAME) || '";"'
|| TRIM(c.COLUMNNAME) || '";"'' ||
TRIM(CAST(COALESCE(MAX(cnt) * 1.0 / SUM(cnt), 0) AS NUMBER FORMAT ''9.9999'')) ||
''";"'' II
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"')
  (
   SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
   SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
    ROWS UNBOUNDED PRECEDING) AS topN
   FROM
    (SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
     CASE WHEN
     ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
 SqlRowNo) = 1 AND SqlRowNo = 1
   THEN 1 ELSE 0 END AS QueryFirstRow
   FROM (
     SELECT q.QueryId, q.SqlTextInfo, q.SqlRowNo,
     MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
     COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
     FROM DBC.dbqlsqltbl q
     INNER JOIN
       SELECT QueryId
       FROM DBC.DBQLogTbl t
       WHERE TRIM(t.StatementType) IN ('SELECT')
       AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
       AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
TRIM(t.USERNAME) END) IN ('All users') --user_name list
       AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
       GROUP BY 1
      ) t
     ON q.QueryId = t.QueryId
      INNER JOIN
```

```
SELECT QueryId
       FROM DBC.QryLogObjectsV
       WHERE ObjectDatabaseName = 'your_database_name'
       AND ObjectType = 'Tab'
       AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
       GROUP BY 1
     ) r
     ON r.QueryId = t.QueryId
     WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
   ) t
 ) t
 WHERE SqlTextInfo NOT LIKE '%";"%'
 ) q
 WHERE
 QueryParts >=1
 AND topN <= 50
 ORDER BY QueryCount DESC, QueryId, SqlRowNo
 QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;
.EXPORT RESET
.LOGOFF
.QUIT
```

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 runme. bat arquivo, substitua *ServerNameUserName*, 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 AWS SCT projeto. 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 e, AWS SCT em seguida, escolha Configurações de conversão. Na lista superior, escolha Teradata e, em seguida, escolha Teradata — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão de Teradata para Amazon Redshift.

As configurações AWS SCT de conversão de Teradata para Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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.

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 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, AWS SCT reescreve INSERT..SELECT as instruções 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 e, AWS SCT 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. AWS SCT exibe as configurações de otimização de conversão para conversão de Teradata para Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Teradata para o Amazon Redshift 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Conectando os bancos AWS Schema Conversion Tool de dados Vertica

Você pode usar 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:

- USO NO ESQUEMA <schema_name>
- USAGE ON SCHEMA PUBLIC
- SELECIONE EM TODAS AS TABELAS NO ESQUEMA <schema_name>
- SELECIONE EM TODAS AS SEQUÊNCIAS NO ESQUEMA <schema_name>
- EXECUTAR EM TODAS AS FUNÇÕES NO ESQUEMA < schema_name >
- EXECUTAR O PROCEDIMENTO

```
<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.
- *procedure_name*Substitua pelo nome de um procedimento de origem. Repita a concessão para cada procedimento que você está convertendo.
- procedure_signatureSubstitua pela lista delimitada por vírgula dos tipos de argumentos do 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. No AWS Schema Conversion Tool, escolha Adicionar fonte.

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 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 Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

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

Parameter	Ação
Nome do servidor	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.
Banco de dados	Digite o nome do banco de dados Vertica.
Nome de usuário e Senha	Insira as credenciais do banco de dados para se conectar ao seu servidor de banco de dados de origem. 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

Parameter	Ação
	conectar ao banco de dados de origem, conforme necessári o.
Usar SSL	 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: Verificar certificado do servidor: Escolha essa opção para verificar o certificado do servidor, usando um armazenam ento 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	AWS SCT cria um cofre seguro para armazenar certificados SSL e senhas de 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	Digite o caminho para o driver a ser usado para a conexão com o banco de dados de origem. Para obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .
	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 informaçõ es, consulte Armazenamento de caminhos do driver nas configurações globais .

- 5. Escolha Testar conexão para verificar se AWS SCT pode se conectar ao seu 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 AWS SCT em e, em seguida, escolha Configurações de conversão. Na lista superior, escolha Vertica e, em seguida, escolha Vertica — Amazon Redshift. AWS SCT exibe todas as configurações disponíveis para conversão de Vertica para Amazon Redshift.

As configurações AWS SCT de conversão do Vertica para o Amazon Redshift 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. 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 AWS SCT podem ser aplicadas ao seu cluster de destino do Amazon Redshift.

Em O número máximo de tabelas para o cluster de destino do Amazon Redshift, escolha o número de tabelas que AWS SCT podem ser aplicadas 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, AWS SCT determinará o número de tabelas a serem aplicadas ao seu cluster de destino do Amazon Redshift, 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.

AWS SCT converte todas as suas tabelas de origem, mesmo que isso seja mais do que o seu cluster do Amazon Redshift pode armazenar. 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, AWS SCT aplica tabelas ao seu cluster de destino do Amazon Redshift 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, AWS SCT você pode migrar cada partição da sua tabela de origem para uma tabela separada no Amazon Redshift. Em seguida, AWS SCT cria uma exibição que inclui dados de todas essas tabelas.

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 AWS SCT podem ser criadas 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.

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 AWS SCT em 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. AWS SCT exibe as configurações de otimização de conversão para a conversão de Vertica para Amazon Redshift.

As configurações AWS SCT de otimização de conversão do Vertica para o Amazon Redshift 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 AWS migração.

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. AWS SCT aplica o estilo ALL de distribuição 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. AWS SCT exclui colunas com o valor de inclinação maior que o limite da lista de candidatos para a chave de distribuição. 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. AWS SCT aplica o estilo ALL de distribuição a tabelas pequenas. Nesse caso, uma cópia de toda a tabela é distribuída para cada nó.

Mapeando tipos de dados no AWS Schema Conversion Tool

Você pode adicionar vários bancos de dados de origem e destino em um único AWS SCT projeto. 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. O 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. Você pode criar várias regras de mapeamento em um único AWS SCT projeto. 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 Aplicação de 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 Como se conectar ao banco de dados de origem com o AWS Schema Conversion Tool.

Se você abrir um projeto salvo na AWS SCT versão 1.0.655 ou anterior, 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

- Mapeando novos tipos de dados no AWS Schema Conversion Tool
- Editando mapeamentos de tipos de dados no AWS Schema Conversion Tool
- Mapeamento para alvos virtuais no AWS Schema Conversion Tool
- Limitações do mapeamento de tipos de dados no AWS Schema Conversion Tool

Mapeando novos tipos de dados no AWS Schema Conversion Tool

Você pode criar várias regras de mapeamento em um único projeto. AWS SCT salva as regras de mapeamento como parte do seu projeto. Com seu projeto aberto, use o procedimento a seguir para adicionar novas regras de mapeamento.

Para criar regras de mapeamento

- 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.
- 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 Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- 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 Visualizar.

AWS SCT destaca em negrito todos os objetos do esquema que fazem parte de uma regra de mapeamento.

Editando mapeamentos de tipos de dados no AWS Schema Conversion Tool

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 <u>Criação</u> 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 <u>Mapeando novos tipos de dados</u> no AWS Schema Conversion Tool.

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.

Mapeamento para alvos virtuais no AWS Schema Conversion Tool

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 Mapeando novos tipos de dados no AWS Schema Conversion Tool. 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 de banco de dados de destino virtual:

- 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 "Relatório de avaliação da".

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 do mapeamento de tipos de dados no AWS Schema Conversion Tool

As limitações a seguir se aplicam ao converter esquemas usando vários servidores em um único AWS SCT projeto:

- · 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.

Trabalhando com relatórios no AWS Schema Conversion Tool

Quando você está planejando uma conversão de banco de dados, é útil criar alguns relatórios para ajudá-lo a entender o que está envolvido. Você pode criar relatórios usando AWS Schema Conversion Tool o.

Você pode usar AWS SCT para criar um relatório de avaliação da migração do 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 AWS SCT e o que mais você precisa 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

Usando o relatório de avaliação no AWS Schema Conversion Tool

Usando o relatório de avaliação no AWS Schema Conversion Tool

Uma parte importante do AWS Schema Conversion Tool é o relatório de avaliação que ele 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 exportálo 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, 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

Relatório de avaliação da Versão 1.0.672 278

de dados. Para obter mais informações sobre plataformas de banco de dados de destino virtual, consulte the section called "Mapeamento de alvos virtuais".

O relatório de avaliação de migração inclui:

- 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 de servidor, sugestões de backup e alterações de servidor vinculado

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 costuma AWS SCT migrar seu esquema existente para uma instância de banco de dados Amazon RDS, então você pode usar o relatório para ajudá-lo a analisar os requisitos para migrar para a AWS nuvem e alterar seu tipo de licença.

Tópicos

- Criando um relatório de avaliação no AWS Schema Conversion Tool
- Visualizando um relatório de avaliação no AWS Schema Conversion Tool
- Salvando o relatório de avaliação em AWS Schema Conversion Tool
- Configurando um relatório de avaliação no AWS Schema Conversion Tool
- Criando um relatório de avaliação de vários servidores no AWS Schema Conversion Tool

Criando um relatório de avaliação no AWS Schema Conversion Tool

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 de banco de dados

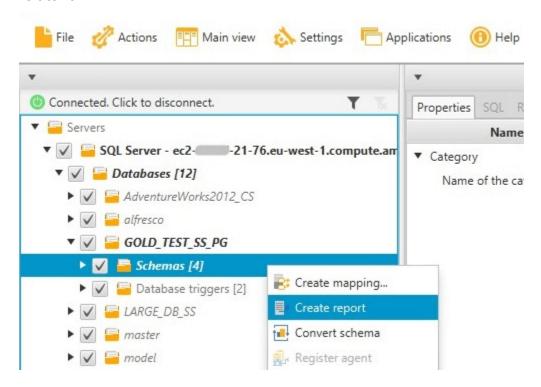
- Certifique-se de ter criado uma regra de mapeamento para o esquema do banco de dados de origem para o qual criar um relatório de avaliação. Para obter mais informações, consulte Mapeando novos tipos de dados no AWS Schema Conversion Tool.
- 2. No menu Visualizar, escolha Visualização principal.

Criar relatório de avaliação Versão 1.0.672 279

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.

4. Abra o menu de contexto (clique com o botão direito do mouse) do objeto e escolha Criar relatório.



Visualizando um relatório de avaliação no AWS Schema Conversion Tool

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.

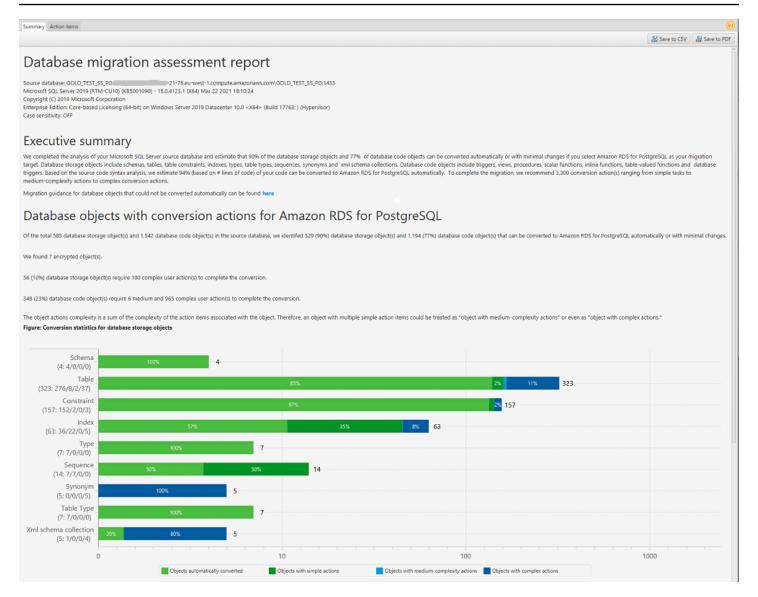
A guia Action Items (Itens de ação) mostra os itens que não puderam ser convertidos automaticamente e as recomendações sobre o que fazer com eles.

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.



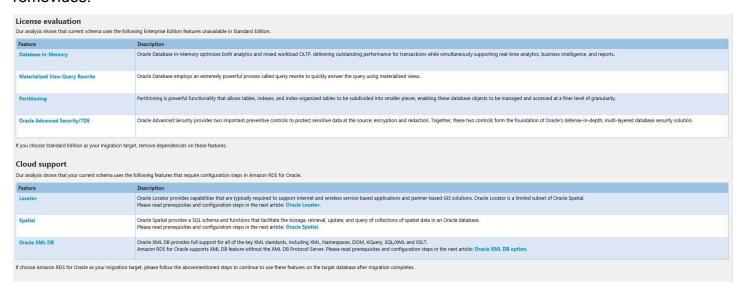
Para os itens de esquema que não podem ser convertidos automaticamente para o mecanismo de banco de dados de destino, o resumo inclui uma estimativa do trabalho necessário para criar itens de esquema na instância de banco de dados de destino que equivalem aos de origem.

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

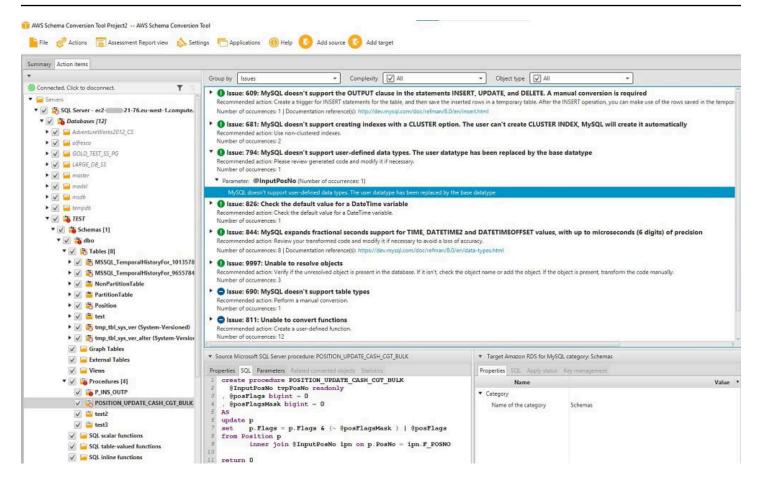
Amazon RDS que executa o mesmo mecanismo. Por exemplo, se você deseja alterar tipos de licença, esta seção do relatório mostra quais recursos do seu banco de dados atual devem ser removidos.



Itens de ação do relatório de avaliação

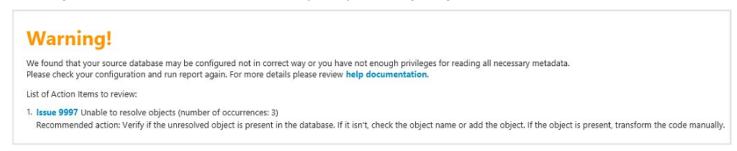
A visualização do relatório de avaliação também inclui uma guia Itens de ação. Esta guia contém uma lista de itens que não podem ser convertidos automaticamente para o mecanismo de banco de dados de sua instância de banco de dados do Amazon RDS de destino. Se você selecionar um item de ação na lista, 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 Conversão de esquemas usando o AWS SCT.



Mensagem de aviso do relatório de avaliação

Para avaliar a complexidade da conversão para outro mecanismo de banco de dados, é AWS SCT necessário 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.



A seguir estão os motivos pelos quais AWS SCT podem 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 n\u00e3o 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 Como se conectar ao banco de dados de origem com o AWS Schema Conversion Tool na seção apropriada do banco de dados de origem neste guia.

Salvando o relatório de avaliação em AWS Schema Conversion Tool

Depois de <u>criar um relatório de avaliação de migração de banco de dados</u>, 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

- No menu superior, selecione Exibir e, em seguida, selecione Visualização do relatório de avaliação.
- 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 identifed 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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

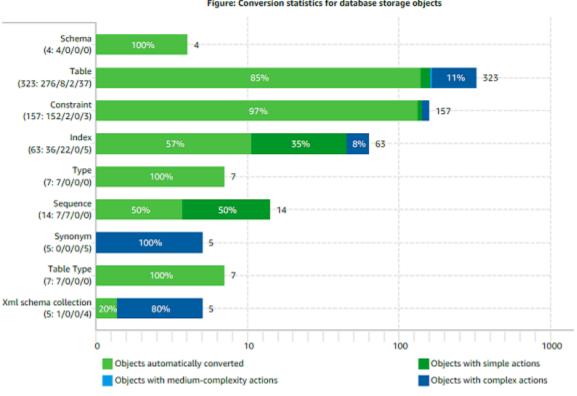


Figure: Conversion statistics for database storage objects

Quando você escolhe a opção Salvar em CSV, 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
- Assunto
- 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

Configurando um relatório de avaliação no AWS Schema Conversion Tool

Você pode configurar a quantidade de detalhes AWS SCT incluídos nos relatórios de avaliação.

Para configurar um relatório de avaliação de migração do banco de dados

 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 <u>criar um relatório de avaliação de migração de banco de dados</u>, 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 identifed 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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

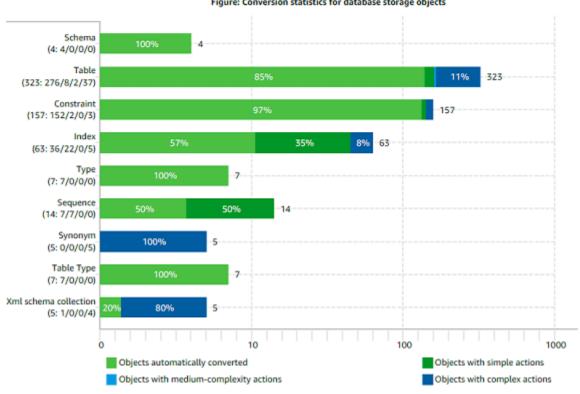


Figure: Conversion statistics for database storage objects

Quando você escolhe a opção Salvar em CSV, 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
- Assunto
- 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

•

Criando um relatório de avaliação de vários servidores no AWS Schema Conversion Tool

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, 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 AWS SCT para criar um relatório de avaliação de vários servidores 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 PostgreSQ			

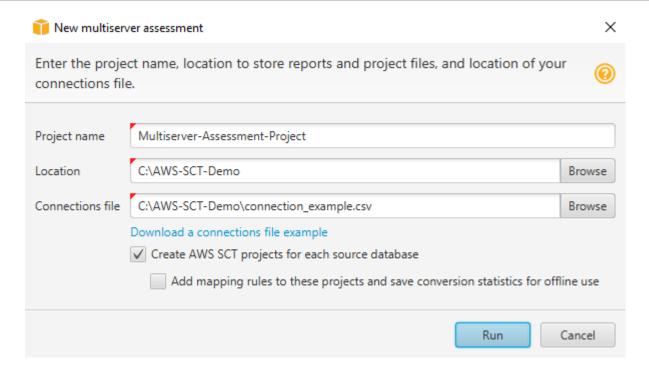
Fonte do banco de dados	Bancos de dados de destino			
	L, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL			
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL			
Netezza	Amazon Redshift			
Oracle	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQ L			
PostgreSQL	Aurora MySQL, 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 de vários servidores com AWS SCT. Você não precisa criar um novo projeto AWS SCT para realizar uma avaliação de vários servidores. 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 Instalando drivers JDBC para AWS Schema Conversion Tool.

Para realizar uma avaliação multisservidor e criar um relatório resumido agregado

1. Em AWS SCT, escolha Arquivo, Nova avaliação de vários servidores. A caixa de diálogo Nova avaliação multisservidor é aberta.



- 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. Escolha Criar AWS SCT projetos para cada banco de dados de origem para criar automaticamente projetos de migração após gerar o relatório de avaliação.
- 5. Com a opção Criar AWS SCT projetos 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 offline. Nesse caso, 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 Usando o modo offline em AWS Schema Conversion Tool.
- 6. Escolha Executar.
 - 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, 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 AWS SCT projetos para cada banco de dados de origem ativada, AWS SCT cria um projeto vazio para cada banco de dados de origem. 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 estatísticas de conversão para uso off-line ativada, 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 Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- Uma regra de mapeamento para esse par de origem e destino. Para obter mais informações, consulte Mapeamento de tipo de dados.
- 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 AWS SCT projeto em um modo off-line. Para obter mais informações, consulte <u>Usando o modo offline em AWS Schema</u> Conversion Tool.

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_POSTGRESQL; AURORA_POSTGRESQL; AURORA_POSTGRESQL; Aurora_Postgresquestandere, MSSQL, customers.dbo,,user,password,,,,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 Customers banco de dados e a Autenticação do Windows para se conectar ao HR banco de dados.

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. Começar 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 AWS perfis em AWS SCT. Para obter mais informações, consulte Configurando AWS Secrets Manager no AWS Schema Conversion Tool.

▲ Important

AWS SCT ignorará o parâmetro Secret Manager Key se você incluir os parâmetros IP, porta, login e senha do servidor 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).
- BigQuery path o caminho para o arquivo de chave da conta de serviço do seu BigQuery banco de dados de origem. Para obter mais informações sobre a criação desse arquivo, consulte a Privilégios BigQuery como fonte.
- 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 Synapse Analytics.
 - GOOGLE_BIGQUERY para um banco de dados. BigQuery
 - DB2ZOS para um banco de dados IBM Db2 for 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 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 *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.

						Code object	Storage object	Syntax Elements	Conversion
						conversion % for	conversion % for	conversion % for	Complexity for
						"Amazon RDS for	"Amazon RDS for	"Amazon RDS for	"Amazon RDS for
Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	PostgreSQL"	PostgreSQL"	PostgreSQL"	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 multiservidor 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, 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 AWS SCT podem ser convertidos 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.

Convertendo esquemas de banco de dados em AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter seus esquemas de banco de dados existentes de um mecanismo de banco de dados para outro. Converter um banco de dados usando a interface AWS SCT do usuário pode ser bem simples, mas há várias coisas a serem consideradas antes de fazer a conversão.

Por exemplo, você pode usar AWS SCT para fazer o seguinte:

- Você pode usar AWS SCT para copiar um esquema de banco de dados local existente para uma instância de banco de dados Amazon RDS 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 autogerenciar um banco de dados na plataforma Amazon Elastic Compute Cloud EC2 (Amazon), poderá emular esses recursos AWS substituindoos por serviços.
- AWS SCT automatiza grande parte do processo de conversão do esquema de banco de dados de processamento de transações on-line (OLTP) em uma instância de banco de dados MySQL do Amazon Relational Database Service (Amazon RDS), em um cluster de banco de dados Amazon Aurora ou em uma instância de banco de dados PostgreSQL. Os mecanismos de banco de dados de origem e destino contêm muitos recursos e capacidades diferentes e AWS SCT tentam criar um esquema equivalente em sua instância de banco de dados Amazon RDS sempre que possível. Se nenhuma conversão direta for possível, AWS SCT fornece uma lista de ações possíveis para você realizar.

Tópicos

- Aplicando regras de migração em AWS Schema Conversion Tool
- Conversão de esquemas usando o AWS SCT
- Conversão manual de esquemas em AWS SCT
- Atualizando e atualizando esquemas convertidos no AWS SCT
- Salvando e aplicando esquemas convertidos em AWS SCT
- Comparando esquemas em AWS Schema Conversion Tool

· Visualizando objetos transformados relacionados em AWS Schema Conversion Tool

AWS SCT suporta as seguintes conversões de processamento de transações on-line (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 Aurora MySQL sem usar. 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 Convertendo esquemas de data warehouse para o Amazon RDS usando AWS SCT.

Para converter o esquema de banco de dados para o Amazon RDS, siga as etapas de alto nível a seguir:

- <u>Criação de regras de migração em AWS SCT</u>— Antes de converter seu esquema com AWS SCT, você pode configurar regras que alteram o tipo de dados das colunas, movem objetos de um esquema para outro e alteram os nomes dos objetos.
- ???: 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.
- Usando o relatório de avaliação no AWS Schema Conversion Tool: 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.
- Conversão de esquemas usando o 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.
- <u>Atualizando e atualizando esquemas convertidos no AWS SCT</u>— Você pode atualizar seu AWS SCT projeto com o esquema mais recente do seu banco de dados de origem.
- <u>Salvando e aplicando esquemas convertidos em AWS SCT</u>— Quando estiver pronto, AWS SCT aplique o esquema convertido em seu projeto local à sua instância de banco de dados Amazon RDS de destino.

Aplicando regras de migração em AWS Schema Conversion Tool

Antes de converter seu esquema com AWS SCT, você pode configurar regras de migração. As regras de migração AWS SCT podem fazer transformações 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
- Schema
- 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:

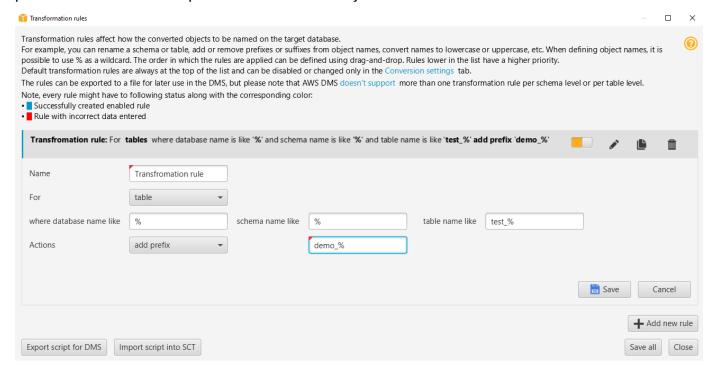
- a. Em Name (Nome), insira um nome para a regra.
- b. Em Para, escolha o tipo de objeto ao qual a regra se aplica.
- c. Em onde, digite um filtro para aplicar aos objetos antes de aplicar a regra de migração. A cláusula where é avaliada usando uma cláusula like. É possível inserir um nome exato para selecionar um objeto ou 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.
- 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).

 Após configurar sua regra de migração, selecione Salvar para salvar sua regra. Você também pode escolher Cancelar para cancelar as alterações.



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 costuma AWS DMS migrar seus dados do banco de dados de origem para o banco de dados de destino, você pode fornecer informações sobre suas 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. No AWS Schema Conversion Tool, escolha Visualização de mapeamento no menu Exibir.
- 2. Em Regras de migração, escolha uma regra de migração e, em seguida, selecione Modificar regra de migração.
- Escolha Exportar script para 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.

Conversão de esquemas usando o AWS SCT

Depois de conectar seu projeto ao banco de dados de origem e à sua instância de banco de dados Amazon RDS de destino, seu AWS Schema Conversion Tool projeto exibe o esquema do seu 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, AWS SCT solicita as informações do esquema do seu banco de dados de origem naquele momento.

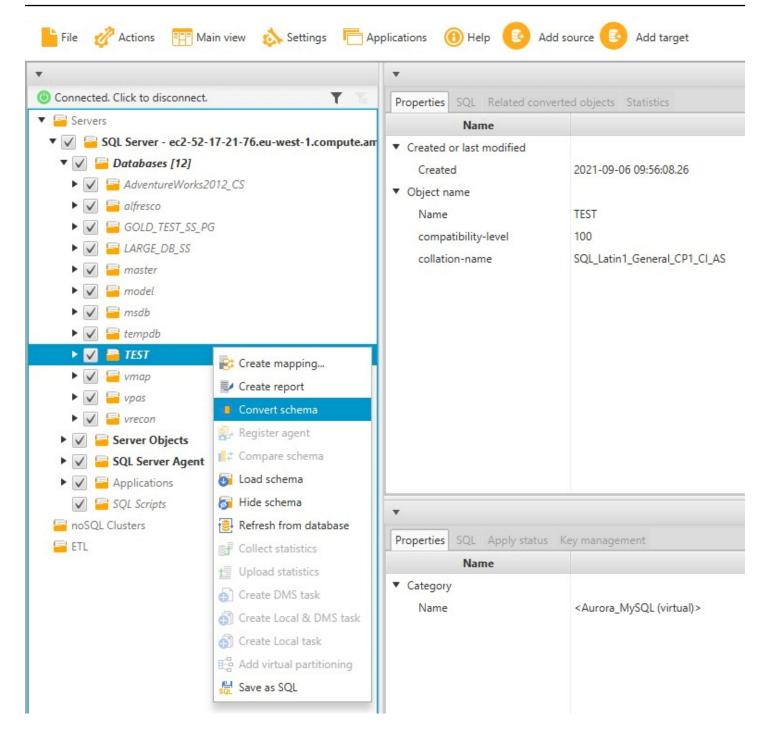
Você pode escolher itens do esquema de seu banco de dados de origem e converter o esquema para o esquema equivalente do mecanismo de banco de dados de sua instância de banco de

dados de destino. É possível escolher qualquer item do esquema do banco de dados de origem para converter. Se o item do esquema que você escolher depender de um item pai, ele AWS SCT também gerará o esquema para o item pai. Por exemplo, suponha que você escolha uma tabela para converter. Em caso afirmativo, AWS SCT gera o esquema para a tabela e o banco de dados em que a tabela está.

Conversão de esquemas

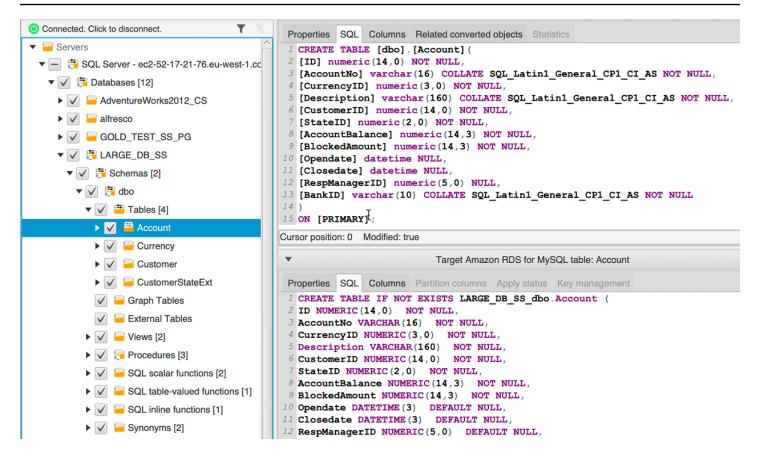
Para converter um esquema do seu banco de dados de origem, marque a caixa de seleção do nome do esquema a ser convertido. Em seguida, selecione esse esquema no painel esquerdo do seu projeto. A AWS SCT destaca o nome do esquema em azul. Abra o menu de contexto (clique com o botão direito do mouse) do esquema e selecione Converter esquema, conforme mostrado a seguir.

Conversão de esquemas Versão 1.0.672 307



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.

Conversão de esquemas Versão 1.0.672 308



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 Atualizando esquemas convertidos no 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 Usando o relatório de avaliação no AWS Schema Conversion Tool.

Quando AWS SCT gera um esquema convertido, ele 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.

Conversão de esquemas Versão 1.0.672 309

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

- 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.

```
Target Amazon RDS for MySQL table: Account
               Columns Partition columns Apply status Key management
Properties
          SQL
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 NUMBRIC(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)
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, armazene AWS SCT somente 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 remove os elementos do esquema planejado em seu AWS SCT projeto para corresponder ao que existe na sua instância de banco de dados de origem.

Conversão manual de esquemas em 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 Salvando e aplicando esquemas convertidos em 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 contêm sugestões de como criar o esquema equivalente.

Marning

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 rearquitetar 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.

Atualizando e atualizando esquemas convertidos no AWS SCT

Você pode atualizar o esquema de origem e o esquema de destino em seu AWS Schema Conversion Tool projeto.

- 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 nenhum esquema à sua instância de banco de dados de destino, AWS SCT limpe o esquema convertido do 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.

Você atualiza o esquema em seu AWS SCT projeto escolhendo Atualizar do banco de dados.



Note

Quando você atualiza seu esquema, 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.

Salvando e aplicando esquemas convertidos em AWS SCT

Quando AWS Schema Conversion Tool gera o esquema convertido (conforme mostrado em???), ele não aplica imediatamente o esquema convertido à instância de banco de dados de destino. 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 Usando o relatório de avaliação no AWS Schema Conversion Tool.

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

É possível salvar o esquema convertido como scripts SQL em um arquivo de texto. Usando essa abordagem, você pode modificar os scripts SQL gerados AWS SCT para abordar 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

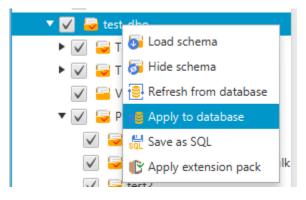
- 1. Selecione seu esquema e abra o menu de contexto (clique com o botão direito do mouse).
- 2. Selecione Salvar como SQL.
- 3. Insira o nome do arquivo e selecione Salvar.
- 4. Salve seu esquema convertido usando uma das seguintes opções:
 - Arquivo único
 - Arquivo único por estágio
 - Arquivo único 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 for totalmente migrado para sua instância de banco de dados de destino e você não precisar mais AWS SCT, você poderá 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 <u>Usando as AWS Lambda funções do pacote AWS SCT de</u> extensão .

Comparando esquemas em AWS Schema Conversion Tool

Se você fez alterações no esquema de origem ou de destino após a migração, você pode comparar os dois esquemas de banco de dados usando. AWS SCT Você pode comparar esquemas para versões iguais ou anteriores do esquema de origem.

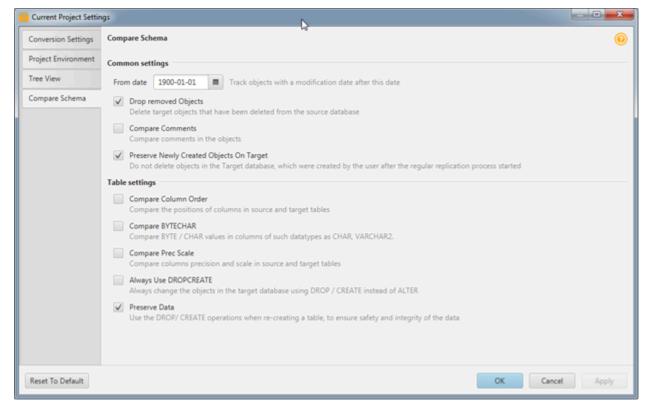
As seguintes comparações de esquemas são compatíveis:

- 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).



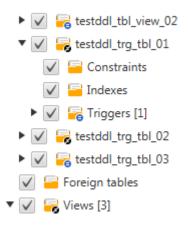
Para comparar esquemas, você seleciona os esquemas e AWS SCT indica os objetos que diferem entre os dois esquemas e os objetos que não diferem.

Para comparar dois esquemas

- Abra um projeto da existente da AWS SCT ou crie um projeto e conecte-o aos endpoints de origem e de destino.
- Selecione o esquema que você deseja comparar.
- 3. Abra o menu de contexto (clique com botão direito do mouse) e selecione Comparar esquema.

AWS SCT indica objetos que são diferentes entre os dois esquemas adicionando um círculo preto ao ícone do objeto.

Como comparar esquemas Versão 1.0.672 316



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.

Visualizando objetos transformados relacionados em AWS Schema Conversion Tool

Depois de uma conversão de esquema, em alguns casos, AWS SCT pode ter criado vários objetos para um objeto de esquema no banco de dados de origem. Por exemplo, ao realizar uma conversão de Oracle para PostgreSQL AWS SCT, pega cada gatilho Oracle e o transforma em uma função de gatilho e gatilho no destino do PostgreSQL. Além disso, ao AWS SCT converter uma função ou procedimento do pacote Oracle para o PostgreSQL, ele cria uma função equivalente e uma função INIT que devem ser executadas como bloco de inicialização antes que o procedimento ou a 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

- 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.

Convertendo esquemas de data warehouse para o Amazon RDS usando AWS SCT

O AWS Schema Conversion Tool (AWS SCT) automatiza grande parte do processo de conversão do seu esquema de armazém de dados em um esquema de banco de dados Amazon RDS ou Aurora. Como os mecanismos de banco de dados de origem e de destino podem ter muitos recursos e capacidades diferentes, AWS SCT tente criar um esquema equivalente no banco de dados de destino sempre que possível. Se nenhuma conversão direta for possível, AWS SCT fornece um relatório de avaliação com uma lista de possíveis ações a serem tomadas. Usando AWS SCT, você pode gerenciar chaves, mapear tipos de dados e objetos e criar conversões manuais.

AWS SCT pode converter os seguintes esquemas de data warehouse.

- 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 <u>Convertendo esquemas de banco de dados</u> em AWS Schema Conversion Tool.

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 Escolhendo estratégias e regras de otimização para uso com AWS SCT. Para obter mais informações sobre as regras de migração, consulte Criação de regras de migração em 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 Coletando ou fazendo upload de estatísticas para 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 <u>Usando o relatório de avaliação no AWS Schema Conversion Tool</u>.
- 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 Convertendo seu esquema usando 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 Gerenciando e personalizando chaves em 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 <u>Como lidar com conversões</u> <u>manuais em AWS SCT</u>. Para obter mais informações sobre a atualização do esquema de origem, consulte Atualizando e atualizando seu esquema convertido no 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 Salvando e aplicando seu esquema convertido no 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, *user_name* substitua pelo nome do seu usuário. Em seguida, *db_name* substitua pelo nome do seu banco de dados de destino do Amazon Redshift. Em seguida, *schema_name* substitua 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, *your_password* substitua 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 pacotes de extensão com AWS Schema Conversion Tool.

Para essa operação, AWS SCT 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":[
      {
         "Effect": "Allow",
         "Action":[
             "s3:DeleteObject",
             "s3:GetObject",
             "s3:ListBucket",
             "s3:PutObject"
         ],
         "Resource":[
             "arn:aws:s3:::aws-sct-*"
         ]
      },
         "Effect": "Allow",
         "Action":[
             "s3:ListAllMyBuckets"
         ],
         "Resource":""
      }
   ]
}
```

Escolhendo estratégias e regras de otimização para uso com AWS SCT

Para otimizar a forma como AWS Schema Conversion Tool converte seu esquema de data warehouse, você pode escolher as estratégias e regras que 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

- 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 Coletando ou fazendo upload de estatísticas para 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

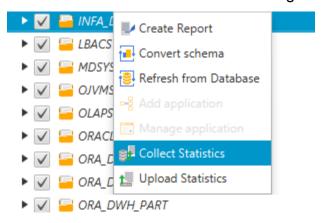
Para cada regra, você pode especificar um peso para a chave de classificação e um peso para a chave de distribuição. A AWS SCT usa os pesos que você escolher para converter seu esquema. Posteriormente, ao analisar as sugestões de chaves, se você não estiver satisfeito com os resultados, é possível retornar aqui e alterar suas configurações. Para obter mais informações, consulte Gerenciando e personalizando chaves em AWS SCT.

Coletando ou fazendo upload de estatísticas para AWS SCT

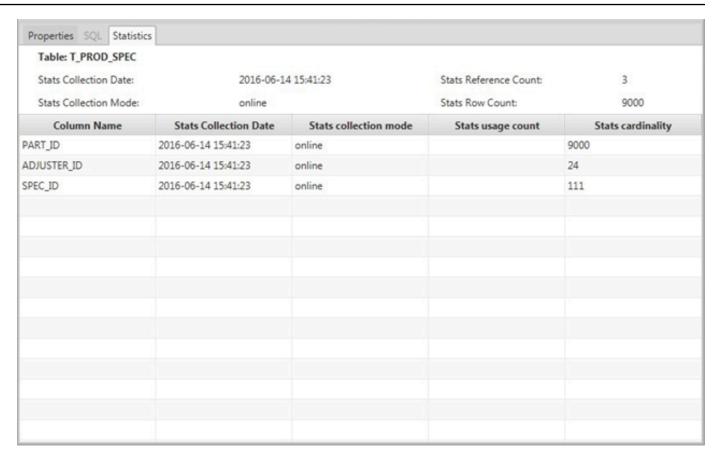
Para otimizar a forma como AWS Schema Conversion Tool converte seu esquema de armazém de dados, você pode fornecer estatísticas do seu banco de dados de origem que a ferramenta possa usar. Você pode coletar estatísticas diretamente do banco de dados ou fazer upload de um arquivo de estatísticas existente.

Para fornecer e analisar estatísticas

- 1. Abra seu projeto e conecte-se ao banco de dados de origem.
- 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.



Escolha um objeto de esquema no painel à esquerda do projeto e escolha a guia Estatísticas.
 Você pode analisar as estatísticas do objeto.



Posteriormente, ao analisar as sugestões de chaves, se você não estiver satisfeito com os resultados, é possível coletar estatísticas adicionais e repetir o procedimento. Para obter mais informações, consulte Gerenciando e personalizando chaves em AWS SCT.

Criação de regras de migração em AWS SCT

Antes de converter seu esquema com AWS SCT, você pode configurar regras de migração. As regras de migração 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 pode criar regras de migração somente 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
- Schema
- Tabela
- Coluna

Como criar regras de migração

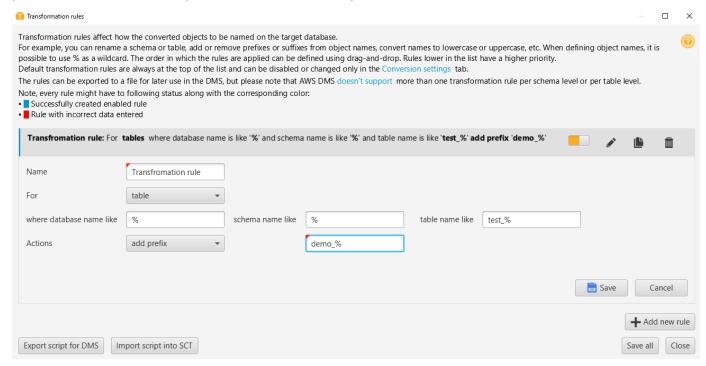
Você pode criar regras de migração salvar as regras como parte do seu projeto. Com seu projeto aberto, use o procedimento a seguir para criar regras de migração.

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.
- 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. É possível inserir um nome exato para selecionar um objeto ou 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.
- Após configurar sua regra de migração, selecione Salvar para salvar sua regra. Você também pode escolher Cancelar para cancelar as alterações.



- Ao finalizar a adição, edição e exclusão das regras, escolha Salvar tudo para salvar todas as alterações.
- 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 AWS Database Migration Service (AWS DMS) para migrar seus dados do banco de dados de origem para o banco de dados de destino, poderá fornecer informações sobre suas regras de migração para o. AWS DMS Para obter mais informações sobre tarefas, consulte <u>Trabalhando</u> com tarefas de AWS Database Migration Service replicação.

Para exportar regras de migração

- 1. No AWS Schema Conversion Tool, escolha Visualização de mapeamento no menu Exibir.
- 2. Em Regras de migração, escolha uma regra de migração e, em seguida, selecione Modificar regra de migração.
- Escolha Exportar script para 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.

Convertendo seu esquema usando AWS SCT

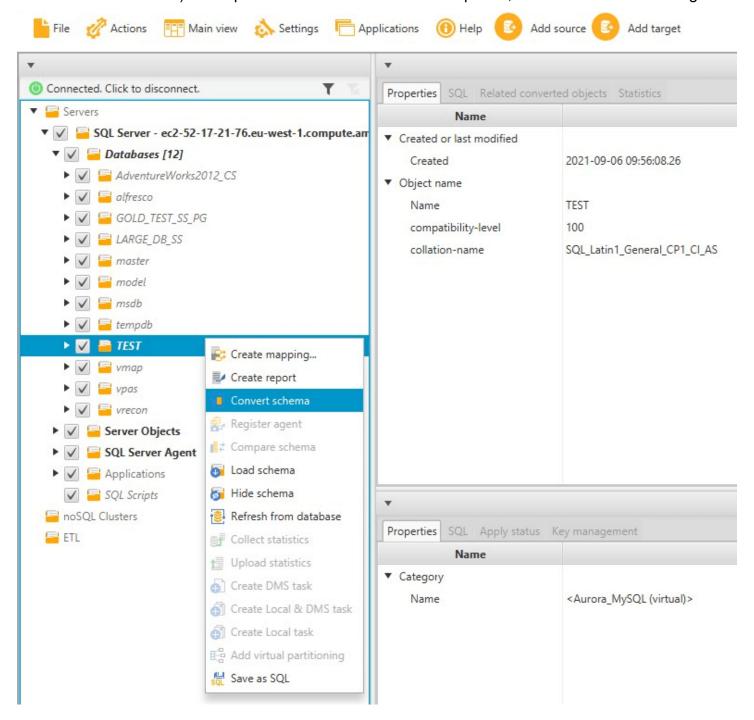
Depois de conectar seu projeto ao banco de dados de origem e ao banco de dados de destino, seu AWS Schema Conversion Tool projeto exibe o esquema do banco de dados de origem no painel esquerdo. O esquema é apresentado em um formato de visualização em árvore, e cada nó da árvore é de carregamento preguiçoso. 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. É possível escolher qualquer item do esquema do banco de dados de origem para converter. Se o item do esquema que você escolher depender de um item pai, ele AWS SCT também gerará o esquema para o item pai. Por exemplo, se você escolher uma coluna de uma tabela para converter, AWS SCT gera o esquema para a coluna, a tabela em que a coluna está e o banco de dados em que a tabela está.

Conversão de esquemas

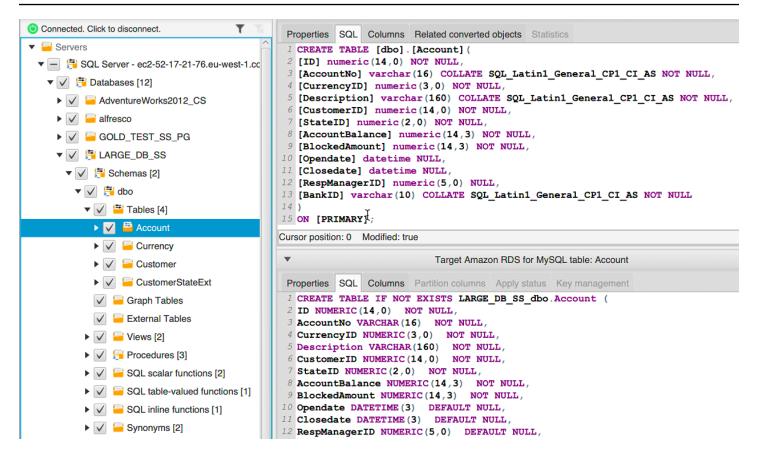
Para converter um esquema do seu banco de dados de origem, marque a caixa de seleção do nome do esquema a ser convertido. Em seguida, selecione esse esquema no painel esquerdo do seu

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.



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.

Conversão de esquemas Versão 1.0.672 328



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 <u>Atualizando e atualizando seu esquema convertido no AWS SCT</u>.

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 Usando o relatório de avaliação no AWS Schema Conversion Tool.

Quando AWS SCT gera um esquema convertido, ele 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.

Conversão de esquemas Versão 1.0.672 329

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

- 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.

```
Target Amazon RDS for MySQL table: Account
               Columns Partition columns Apply status Key management
Properties
          SQL
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 NUMBRIC(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)
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.

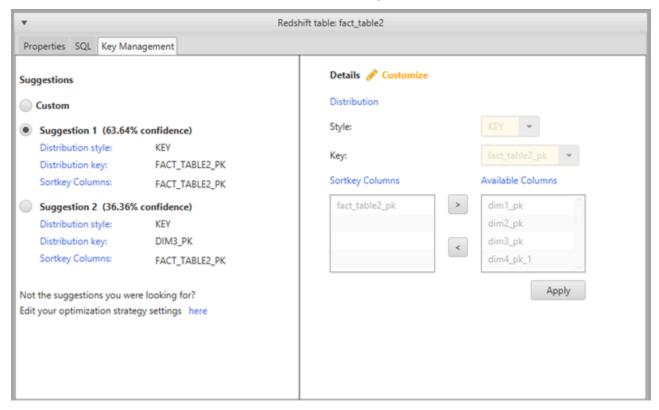
Como limpar um esquema convertido

Até você aplicar o esquema ao seu banco de dados de destino, armazena AWS SCT somente 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 em seu banco de dados de destino, a atualização do banco de dados remove os elementos do esquema planejado em seu AWS SCT projeto para corresponder ao que existe em seu banco de dados de destino.

Gerenciando e personalizando chaves em AWS SCT

Depois de converter seu esquema com o AWS Schema Conversion Tool, você pode 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 Escolhendo estratégias e regras de otimização para uso com AWS SCT.

Tópicos relacionados

- Selecione a melhor chave de classificação
- Selecione o melhor estilo de distribuição

Criando e usando o relatório de avaliação no AWS SCT

O AWS Schema Conversion Tool cria um relatório de avaliação da migração do banco de dados para ajudá-lo a converter seu 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.

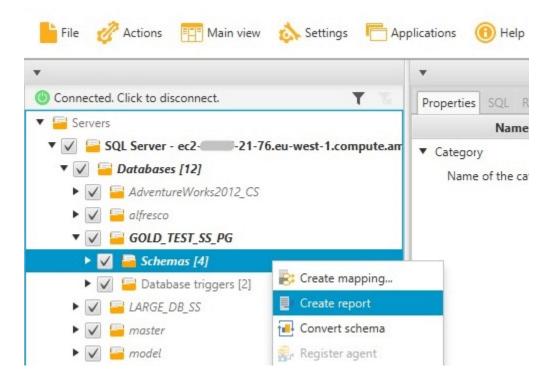
Como criar um relatório de avaliação de migração do banco de dados

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 de banco de dados

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

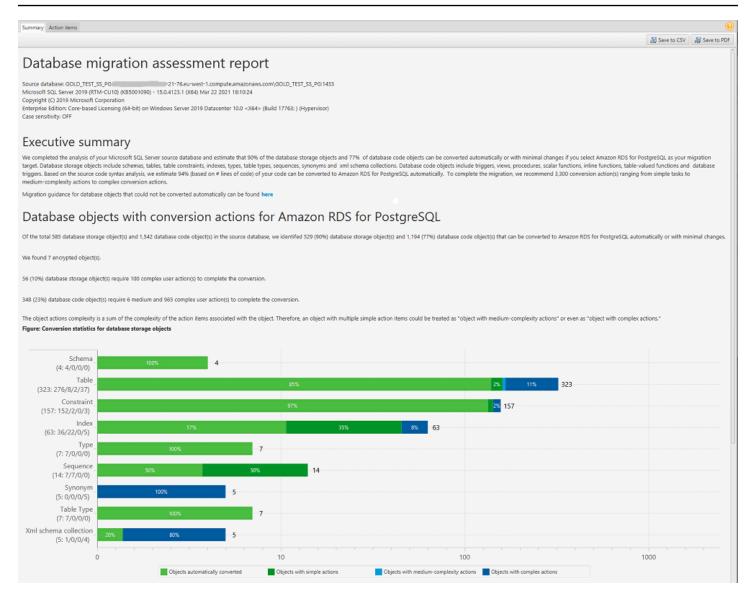
Tópicos relacionados Versão 1.0.672 332



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.

Resumo Versão 1.0.672 333



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:

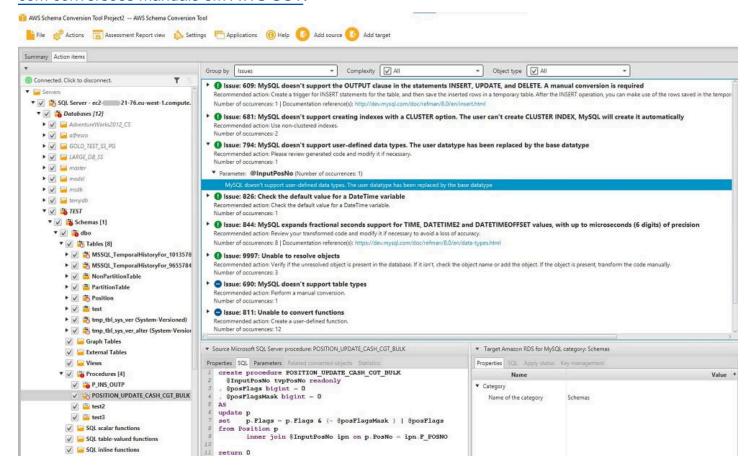
- Simples: Ações que podem ser concluídas em menos de uma hora.
- Média: Ações mais complexas que podem ser concluídas de uma a quatro horas.
- Significativa: Ações muito complexas que levarão mais do que quatro horas para serem concluídas.

Resumo Versão 1.0.672 334

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 do banco 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 em AWS SCT.



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.

Itens de ação Versão 1.0.672 335

Database objects with conversion actions for Amazon RDS for PostgreSQL

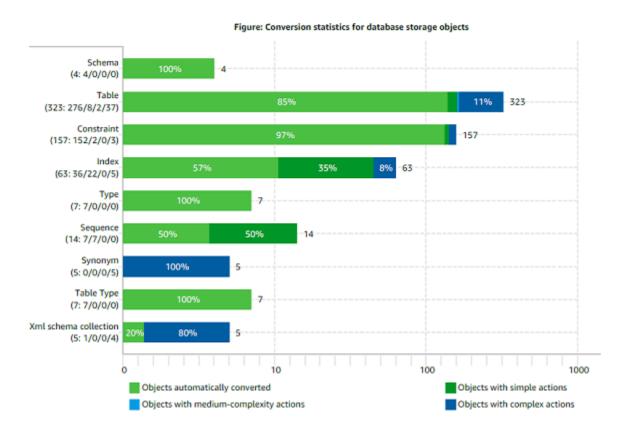
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identifed 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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



Como lidar com conversões manuais em 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 Salvando e aplicando seu esquema convertido no 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.



Marning

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 rearquitetar 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.

Atualizando e atualizando seu esquema convertido no AWS SCT

Você pode atualizar o esquema de origem e o esquema de destino em seu AWS Schema Conversion Tool projeto.

- 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 nenhum esquema ao banco de dados de destino, AWS SCT limpe 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.

Você atualiza o esquema em seu AWS SCT projeto escolhendo Atualizar do banco de dados.

Salvando e aplicando seu esquema convertido no AWS SCT

Quando AWS Schema Conversion Tool gera o esquema convertido (conforme mostrado em<u>Convertendo seu esquema usando AWS SCT</u>), ele não aplica 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 <u>Usando o</u> relatório de avaliação no AWS Schema Conversion Tool.

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

É possível salvar o esquema convertido como scripts SQL em um arquivo de texto. Usando essa abordagem, você pode modificar os scripts SQL gerados AWS SCT para abordar 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

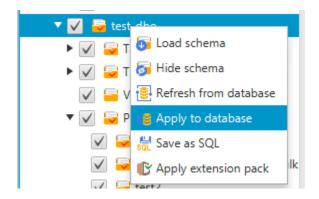
- 1. Selecione seu esquema e abra o menu de contexto (clique com o botão direito do mouse).
- 2. Selecione Salvar como SQL.
- 3. Insira o nome do arquivo e selecione Salvar.
- 4. Salve seu esquema convertido usando uma das seguintes opções:
 - · Arquivo único
 - · Arquivo único por estágio
 - Arquivo 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.
- Em Salvar scripts de SQL em, selecione como você deseja salvar o script do esquema de banco de dados.
- 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 for totalmente migrado para sua instância de banco de dados de destino e você não precisar mais AWS SCT, você poderá 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 <u>Usando pacotes de extensão com AWS Schema Conversion Tool</u>.

Bibliotecas Python

Para criar funções personalizadas no Amazon Redshift, você deve usar a linguagem Python. Use o pacote AWS SCT de extensão para instalar bibliotecas python no seu banco de dados do Amazon Redshift. Para obter mais informações, consulte <u>Usando pacotes de extensão com AWS Schema</u> Conversion Tool.

Convertendo dados do Amazon Redshift usando AWS Schema Conversion Tool

Você pode usar o 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.

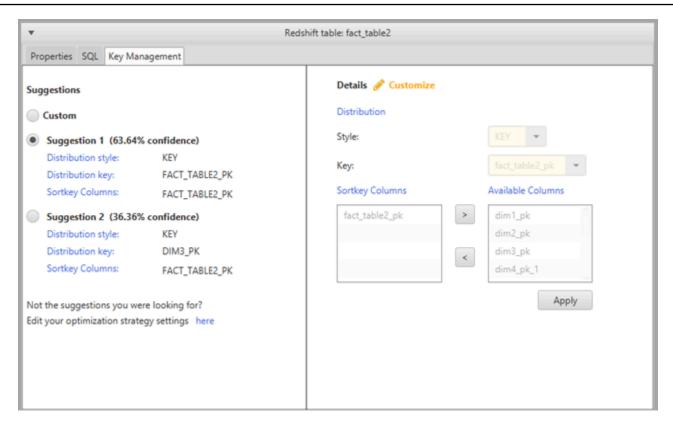
Como otimizar seu banco de dados do Amazon Redshift

Use o procedimento a seguir para otimizar seu banco de dados do Amazon Redshift.

Para otimizar seu banco de dados do Amazon Redshift

- 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 <u>Snapshots do Amazon</u> 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.
 - AWS SCT usa as estatísticas para fazer sugestões de 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.
 - AWS SCT faz sugestões para chaves de classificação e chaves de distribuição.
- 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.

Bibliotecas Python Versão 1.0.672 341



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.

- 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 dos itens 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.

Convertendo dados usando processos ETL em 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, AWS SCT suporta a conversão de scripts de ETL em objetos para o AWS Glue Amazon Redshift RSQL, conforme mostrado na tabela a seguir.

Origem	Alvo
Scripts de ETL da Informatica	Informatica
Pacotes de ETL do Microsoft SQL Server Integration Services (SSIS)	AWS Glue or 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 trabalho da Teradata FastExport	Amazon Redshift RSQL
Scripts de trabalho da Teradata FastLoad	Amazon Redshift RSQL
Scripts de trabalho da Teradata MultiLoad	Amazon Redshift RSQL

Tópicos

- Convertendo processos ETL em in AWS GlueAWS Schema Conversion Tool
- Convertendo processos ETL em in AWS GlueAWS Schema Conversion Tool
- Convertendo scripts ETL da Informatica com AWS Schema Conversion Tool
- · Convertendo SSIS em com AWS GlueAWS SCT
- Convertendo pacotes SSIS em com AWS Glue StudioAWS Schema Conversion Tool

- Convertendo scripts Teradata BTEQ em Amazon Redshift RSQL com AWS SCT
- Conversão de scripts de shell com comandos BTEQ incorporados para o Amazon Redshift RSQL com AWS Schema Conversion Tool
- Conversão de FastExport scripts para Amazon Redshift RSQL com AWS Schema Conversion Tool
- Convertendo scripts de FastLoad trabalho em Amazon Redshift RSQL com AWS Schema Conversion Tool
- Conversão de MultiLoad scripts para Amazon Redshift RSQL com AWS Schema Conversion Tool

Convertendo processos ETL em in AWS GlueAWS Schema Conversion Tool

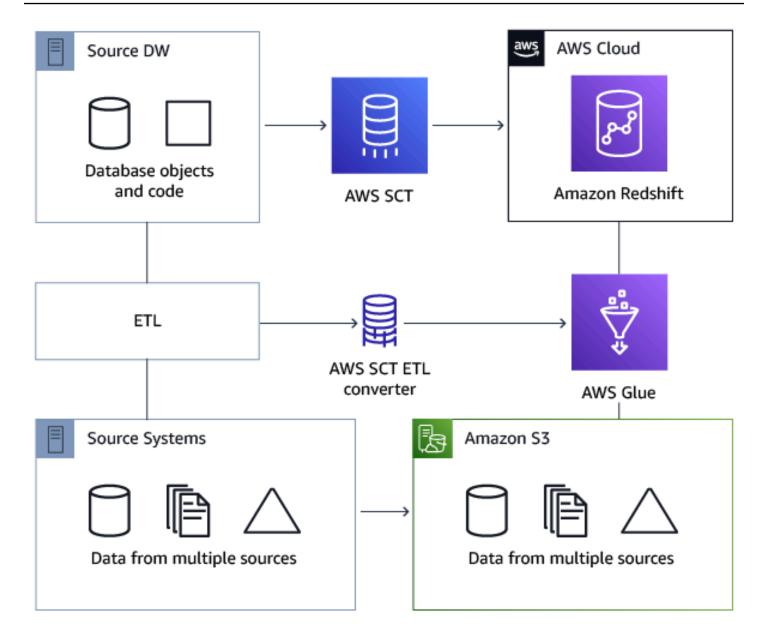
A seguir, você pode encontrar um resumo do processo para converter scripts ETL em with. AWS Glue AWS SCT Neste exemplo, convertemos um banco de dados Oracle para Amazon Redshift com os processos de ETL usados nos data warehouses e bancos de dados de origem.

Tópicos

- Pré-requisitos
- Entendendo o catálogo AWS Glue de dados
- Limitações para conversão usando com AWS SCTAWS Glue
- Etapa 1: criar um novo projeto
- Etapa 2: criar um AWS Glue trabalho

O diagrama de arquitetura a seguir mostra um exemplo de projeto de migração de banco de dados que inclui a conversão de scripts ETL em AWS Glue.

Processos de ETL Versão 1.0.672 345



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 o. 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.

Pré-requisitos Versão 1.0.672 346

Além disso, AWS Glue precisa de permissões para acessar outros AWS recursos em seu nome. Você fornece essas permissões usando AWS Identity and Access Management (IAM). Certifique-se de ter criado uma política do IAM para AWS Glue. Para obter mais informações, consulte <u>Criar uma política do IAM para o AWS Glueservice no Guia do AWS Glue desenvolvedor.</u>

Entendendo o catálogo AWS Glue de dados

Como parte do processo de conversão, 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
- Acionadores Lógica que controla quando uma tarefa de ETL é executada AWS Glue (seja sob demanda, por agendamento ou acionada por eventos de trabalho)

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 AWS Glue e AWS SCT, o Catálogo de AWS Glue Dados contém referências a dados que são usados como fontes e destinos de suas tarefas de ETL em 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 em AWS Glue.

Limitações para conversão usando com AWS SCTAWS Glue

As limitações a seguir se aplicam ao converter usando AWS SCT com AWS Glue.

Recurso Limite padrão

AWS Glue Catálogo de dados Versão 1.0.672 347

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 ao mesmo tempo	5
Máximo DPUs usado por uma função ao mesmo tempo	100

Limitações Versão 1.0.672 348

Tamanho do nome do banco de dados	Ilimitado
	Para compatibilidade com outros armazenam entos 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 em AWS SCT. Para obter mais informações, consulte <u>Iniciando e</u> gerenciando projetos em AWS SCT.
- 2. Adicione seus bancos de dados de origem e destino ao projeto. Para obter mais informações, consulte Adicionando servidores ao projeto em 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 AWS perfil, escolha o perfil que você deseja usar. O perfil deve preencher automaticamente a chave de AWS acesso, a chave secreta e a pasta de bucket do Amazon S3. Se isso não ocorrer, insira essas informações por conta própria. Depois de escolher OK, AWS Glue analisa os objetos e carrega os metadados no Catálogo de AWS Glue Dados.

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.

AWS Glue cria um banco de dados no servidor de banco de dados de origem e outro no servidor de banco de dados de destino para ajudar na conversão de ETL. O banco de dados no servidor de destino contém o Catálogo AWS Glue de Dados. 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 AWS SCT projeto.

Para criar um AWS Glue trabalho com seu script convertido, faça o upload do seu script 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 AWS Glue trabalho

Depois de salvar o script no Amazon S3, você pode escolhê-lo e, em seguida, escolher Configure AWS Glue Job para abrir o assistente para configurar o AWS Glue trabalho. 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:
 - AWS Identity and Access Management Função (IAM)
 - · 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
- Concorrente DPUs para cada execução de trabalho
- 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.

Depois de concluir a configuração da tarefa, ela é exibida sob as tarefas ETL no Catálogo de AWS Glue Dados. Se você escolher a tarefa, as configurações serão exibidas para que possa rever e editá-las. Para criar um novo trabalho em AWS Glue, escolha Create AWS Glue Job no menu de contexto (clique com o botão direito do mouse) do job. 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).

Nesse ponto, você pode ver seu trabalho no AWS Glue console. Para fazer isso, faça login no AWS Management Console e abra o AWS Glue console 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 ver os registros de erros no AWS Glue console.

Convertendo processos ETL em in AWS GlueAWS Schema Conversion Tool

Nas seções a seguir, você encontrará uma descrição de uma conversão que chama operações de AWS Glue API em Python. Para obter mais informações, consulte <u>Programar scripts de ETL do AWS</u> 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 AWS Glue dados usando a <u>API do AWS SDK</u>. Ao definir uma tabela no catálogo de dados, adicione-a a um banco de dados. Um banco de dados é usado para organizar as tabelas em AWS Glue.

O exemplo a seguir demonstra o create_database método da API Python para. AWS Glue

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 create_connectionmétodo da API Python para. AWS Glue

```
response = client.create_connection(
    ConnectionInput={
        'Name': 'Redshift_abcde03.aabbcc112233.us-west-2.redshift.amazonaws.com_dev',
        'Description': 'Created from SCT',
        'ConnectionType': 'JDBC',
        'ConnectionProperties': {
             'JDBC_CONNECTION_URL': 'jdbc:redshift://aabbcc03.aabbcc112233.us-
west-2.redshift.amazonaws.com:5439/dev',
            'USERNAME': 'user_name',
            'PASSWORD': 'password'
        },
        'PhysicalConnectionRequirements': {
            'AvailabilityZone': 'us-west-2c',
            'SubnetId': 'subnet-a1b23c45',
            'SecurityGroupIdList': [
                'sg-000a2b3c', 'sg-1a230b4c', 'sg-aba12c3d', 'sg-1abb2345'
            ]
        }
    }
)
```

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_<<u>Endpoint-name</u>>_<<u>redshift-database-name</u>>, 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.

Etapa 2: criar uma conexão Versão 1.0.672 353

 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 AWS Glue rastreador para preencher o catálogo. AWS Glue Para obter mais informações, consulte <u>Catalogar tabelas com um crawler</u> 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 <u>API do SDK da AWS</u>. 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_shema_name>, por exemplo: abcde03_dev_ora_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/ora_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. Os tipos de dados do Amazon Redshift são AWS Glue mapeados para tipos de dados da seguinte forma após a execução do crawler.

Tipo de dados do Amazon Redshift	AWS Glue tipo de dados
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

Convertendo scripts ETL da Informatica com AWS Schema Conversion Tool

Você pode usar a interface de linha de AWS SCT comando (CLI) para converter seus scripts ETL da Informatica para que você possa usar os scripts com seu novo banco de dados de destino. Essa conversão inclui três etapas principais. Primeiro, AWS SCT converte o código SQL que está

incorporado em seus objetos da Informatica. Em seguida, AWS SCT altere os nomes dos objetos do banco de dados de acordo com as regras de migração que você especificou no seu projeto. Por fim, AWS SCT redireciona as conexões dos seus scripts ETL da Informatica para o novo banco de dados de destino.

Você pode converter scripts ETL da Informatica como parte do projeto de conversão de AWS SCT banco de dados. Certifique-se de adicionar seus bancos de dados de origem e destino ao projeto quando converter scripts de ETL da Informatica.

Para converter scripts ETL da Informatica, certifique-se de usar a AWS SCT versão 1.0.667 ou superior. Além disso, familiarize-se com a interface de linha de comandos da AWS SCT. Para obter mais informações, consulte Referência de CLI para AWS Schema Conversion Tool.

Para converter scripts ETL da Informatica usando AWS SCT

- Crie um novo script de AWS SCT CLI 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 AWS SCT você 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. AWS SCT Depois de executar esse exemplo de código, AWS SCT armazena os arquivos de log na C:\sct_log pasta. Além disso, AWS SCT armazena os arquivos de log do console na C:\Temp\oracle_postgresql pasta.

```
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. Crie um novo AWS SCT projeto. 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. 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* pelos nomes dos usuários do seu 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 <u>Referência da CLI do AWS Schema</u> Conversion Tool.

5. Crie uma nova regra de AWS SCT mapeamento, que define os mecanismos de banco de dados de destino para cada esquema de banco de dados de origem. Para obter mais informações, consulte Mapeando tipos de dados no AWS Schema Conversion Tool.

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'
```

 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"
    }
    ]
}'
```

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'
/
```

 (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 .scts arquivo e execute-o usando o RunSCTBatch comando na AWS SCT CLI. 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 ETL de origem da Informatica, execute o script AWS SCT CLI novamente.

Convertendo SSIS em com AWS GlueAWS SCT

A seguir, você pode descobrir como converter pacotes do Microsoft SQL Server Integration Services (SSIS) em AWS Glue uso AWS SCT.

Para converter pacotes Microsoft SSIS em AWS Glue, certifique-se de usar a AWS SCT versão 1.0.642 ou superior. 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 uso AWS GlueAWS SCT

- 1. Crie um novo projeto AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 (arquivos com .dtsx as extensões .conmgr ou.params) da pasta local e os analisa. Em seguida, ele os organiza em uma AWS SCT árvore de categorias.

- 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 AWS perfil Escolha o perfil a ser usado.
 - AWS chave de acesso Insira sua chave de AWS acesso.
 - AWS chave secreta Insira sua chave AWS secreta.
 - Região Escolha a Região da AWS que você deseja usar na lista.
 - 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 AWS Glue alvo virtual. Nesse caso, você não precisa especificar as credenciais de conexão. Para obter mais informações, consulte the section called "Mapeamento de alvos virtuais".

6. Crie uma nova regra de mapeamento que inclua seu pacote SSIS de origem e seu AWS Glue destino. Para obter mais informações, consulte the section called "Mapeamento de novos tipos de dados".

- 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 AWS Glue conexão para o gerenciador de conexões SSIS correspondente. Certifique-se de que suas AWS Glue conexões já tenham sido criadas.

- a. Em Conexões, escolha Conexões do projeto.
- b. Para conexão com o catálogo Glue, escolha a AWS Glue conexão apropriada.
- 10. Configure o gerenciador de conexões de pacotes:
 - a. Em Conexões, escolha seu pacote.
 - b. Para conexão com o catálogo Glue, escolha a AWS Glue conexão apropriada.
 - 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 seu AWS Glue trabalho. 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 AWS Glue tarefa.
- 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 do seu AWS Glue trabalho.
 - 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).
 - Diretório temporário Insira o caminho do Amazon S3 para um diretório temporário para obter resultados intermediários. AWS Glue e as transformações AWS Glue integradas usam esse diretório para ler ou gravar no Amazon Redshift.
 - 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 caminho 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 AWS Glue dados (DPUs) que podem ser alocadas quando esse trabalho é executado. 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. AWS Glue retorna um erro quando esse 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 de AWS SCT enviar uma notificação de atraso.
- Número de novas tentativas insira o número de vezes (0 a 10) que AWS Glue devem reiniciar automaticamente a tarefa se ela falhar. As tarefas que atingem o limite de tempo não são reiniciadas. O padrão é 0.
- Escolha Próximo.
- c. Configure as conexões necessárias:
 - Em Todas as conexões, escolha as AWS Glue conexões necessárias e adicione-as à lista de conexões selecionadas.
 - ii. Escolha Terminar.
- 16. Crie um AWS Glue trabalho configurado. 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 ETL que você configurou e escolha Create Job AWS Glue.
- 17. Execute o AWS Glue trabalho:
 - a. Abra o AWS Glue console 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 AWS SCT podem ser convertidos em AWS Glue

Você pode usar AWS SCT para converter componentes de fluxo de dados e fluxo de controle, bem como contêineres, parâmetros e variáveis.

Os componentes de fluxo de dados compatíveis incluem os seguintes:

- Destino ADO NET
- Origem ADO NET
- Agregar
- 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 do ODBC
- Origem do ODBC
- Transformação de comando OLE DB
- Destino do OLEDB
- Origem do OLEDB
- Transformação de amostragem percentual
- Transformação de articulação
- Destino do arquivo bruto
- · Origem do arquivo bruto
- RecordSet Destino
- 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 contêiners SSIS suportados incluem os seguintes:

- Contêiner For Loop
- Contêiner Foreach Loop
- · Contêiner de sequência

Convertendo pacotes SSIS em com AWS Glue StudioAWS Schema Conversion Tool

Você pode usar 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

SSIS para AWS Glue Studio Versão 1.0.672 367

origem para o Nuvem AWS, você pode executar esses AWS Glue Studio trabalhos convertidos para realizar tarefas de ETL.

Para converter pacotes Microsoft SSIS em AWS Glue Studio, certifique-se de usar a AWS SCT versão 1.0.661 ou superior.

Tópicos

- Pré-requisitos
- Adicionando pacotes SSIS ao seu projeto AWS SCT
- Convertendo pacotes SSIS em com AWS Glue StudioAWS SCT
- Criação AWS Glue Studio de trabalhos usando o código convertido
- Criando um relatório de avaliação para um pacote SSIS com AWS SCT
- Componentes do SSIS que AWS SCT podem ser convertidos 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 AWS dos recursos necessários em sua conta.

Você pode usar AWS Identity and Access Management (IAM) para definir políticas e funções necessárias para acessar os recursos que AWS Glue Studio usa. Para obter mais informações, consulte Permissões do IAM para o AWS Glue Studio usuário.

Depois de AWS SCT converter seus scripts de origem em AWS Glue Studio, faça o upload dos scripts convertidos em 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 ele AWS Glue Studio possa se conectar ao seu armazenamento de dados, crie um conector personalizado e uma conexão. Além disso, armazene as credenciais do banco de dados em AWS Secrets Manager.

Para criar um conector personalizado

 Faça download do driver JDBC para seu armazenamento de dados. Para obter mais informações sobre os drivers JDBC que AWS SCT usa, consulte. <u>Instalando drivers JDBC para</u> AWS Schema Conversion Tool

Pré-requisitos Versão 1.0.672 368

 Faça o upload desse arquivo com drivers no seu bucket do Amazon S3. Para obter mais informações, consulte <u>Adicionar um objeto a um bucket</u> no Guia do Usuário do Amazon Simple Storage Service.

- 3. Faça login no AWS Management Console e abra o AWS Glue Studio console 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=\$\$<username>;password>.

Certifique-se de substituir<host>,,<port>, <dbname><username>, e <password> por 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 em AWS Secrets Manager

- Faça login no AWS Management Console e abra o AWS Secrets Manager console 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.

Pré-requisitos Versão 1.0.672 369

4. Na página Configurar segredo, insira um Nome de segredo descritivo. Por exemplo, digite **SQL_Server_secret**.

- 5. Escolha 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

- Faça login no AWS Management Console e abra o AWS Glue Studio console em https://console.aws.amazon.com/gluestudio/.
- 2. Escolha o conector que receberá a conexão e selecione Criar conexão.
- Na página Criar conexão, insira um nome descritivo para ela. Por exemplo, digite SQL-Serverconnection.
- 4. Em AWS Secret, escolha o segredo que você criou em AWS Secrets Manager.
- 5. Configure Opções de rede e escolha Criar conexão.

Agora, você pode criar uma AWS Glue Studio tarefa com um conector personalizado. Para obter mais informações, consulte Criação de AWS Glue Studio empregos.

Adicionando pacotes SSIS ao seu projeto AWS SCT

Você pode adicionar vários pacotes SSIS a um único AWS SCT projeto.

Para adicionar um pacote SSIS ao seu projeto AWS SCT

- Crie um novo projeto com AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 AWS Glue Studio, AWS SCT use seu AWS perfil. Para obter mais informações, consulte Gerenciando perfis no AWS Schema Conversion Tool.

 Crie uma regra de mapeamento, que inclua seu pacote SSIS de origem e seu AWS Glue Studio destino. Para obter mais informações, consulte <u>Mapeando tipos de dados no AWS Schema</u> Conversion Tool.

- 7. Crie AWS Glue Studio conexões no AWS Glue Studio console. 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.
 - AWS SCT exibe a janela Configurar conexões.
- 9. Para cada conexão SSIS de origem, escolha uma AWS Glue Studio conexão.

Convertendo pacotes SSIS em com AWS Glue StudioAWS SCT

A seguir, descubra como converter pacotes SSIS em AWS Glue Studio uso AWS SCT.

Para converter um pacote SSIS em AWS Glue Studio

- Adicione seu pacote SSIS ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionando pacotes SSIS ao seu projeto AWS SCT.
- 2. No painel esquerdo, expanda os nós ETL e SSIS.
- Escolha Pacotes, abra o menu de contexto (clique com o botão direito do mouse) do pacote e escolha Converter pacote.
 - 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 DAGs nó Package na árvore certa.
- Escolha Package DAGs, abra o menu de contexto (clique com o botão direito do mouse) e escolha Save to Amazon S3.

Agora você pode usar esses scripts para criar tarefas no AWS Glue Studio.

Criação AWS Glue Studio de trabalhos usando o código convertido

Depois de converter seus pacotes SSIS de origem, você pode usar os arquivos JSON convertidos para criar AWS Glue Studio trabalhos.

Para criar um AWS Glue Studio trabalho

 Escolha Package DAGs na árvore à direita, abra o menu de contexto (clique com o botão direito do mouse) e escolha Configure AWS Glue Studio job.

- 2. (Opcional) Aplique o pacote de extensão que emula as funções do SSIS no AWS Glue Studio.
- 3. A janela Configurar AWS Glue Studio tarefa 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.

Escolha Próximo.

- 4. Preencha a seção Propriedades avançadas da tarefa:
 - Função do IAM Escolha a função do IAM que é usada para autorização AWS Glue Studio e acesso aos armazenamentos de dados.
 - Caminho do arquivo de script do S3: Insira o caminho do Amazon S3 para seu script convertido.
 - Diretório temporário Insira o caminho do Amazon S3 para um diretório temporário para obter resultados intermediários. AWS Glue Studio usa esse diretório para ler ou gravar no Amazon Redshift.
 - 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 caminho 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. AWS Glue retorna um erro quando esse 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 de AWS SCT enviar uma notificação de atraso.
- Número de novas tentativas insira o número de vezes (0 a 10) que AWS Glue devem reiniciar automaticamente a tarefa se ela falhar. As tarefas que atingem o limite de tempo não são reiniciadas. O padrão é 0.

Escolha Terminar.

AWS SCT configura seus AWS Glue Studio trabalhos selecionados.

- Encontre suas tarefas configuradas em Tarefas de ETL na árvore à direita. Escolha seu trabalho configurado, abra o menu de contexto (clique com o botão direito do mouse) e escolha Criar AWS Glue Studio trabalho.
- 6. Escolha Aplicar status e certifique-se de que o valor do Status da tarefa seja Êxito.
- Abra o AWS Glue Studio console, escolha Atualizar e escolha seu trabalho. Em seguida, escolha Executar.

Criando um relatório de avaliação para um pacote SSIS com 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 não AWS SCT 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.
- Visualize a guia Resumo. Aqui, são AWS SCT exibidas 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.
- (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.
 - 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. Quando você escolhe um item de ação da lista, AWS SCT destaca o item do pacote SSIS de origem ao qual o item de ação se aplica.

Componentes do SSIS que AWS SCT podem ser convertidos em AWS Glue Studio

Você pode usar 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
- Agregar

- Mapa de caracteres
- Divisão condicional
- Coluna Copiar
- Conversão de dados
- Coluna derivada
- Busca
- Mesclar
- Junção de mesclagem
- Multicast
- ODBCDestination
- ODBCSource
- OLEDBDestination
- OLEDBSource
- Número de linhas
- Classificar
- Destino do SQL Server
- Union All

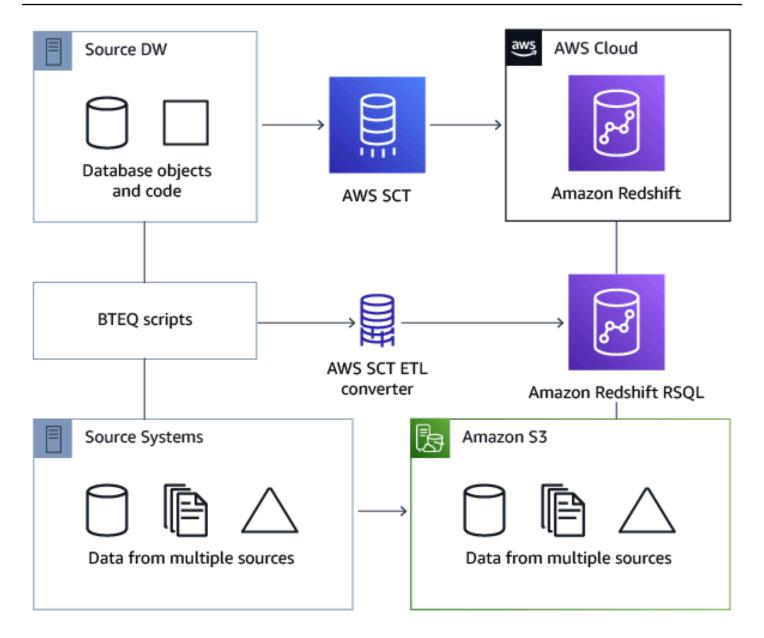
AWS SCT pode converter mais componentes do SSIS em AWS Glue. Para obter mais informações, consulte Componentes do SSIS que AWS SCT podem ser convertidos em AWS Glue.

Convertendo scripts Teradata BTEQ em Amazon Redshift RSQL com AWS SCT

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter scripts 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.

Scripts do Teradata BTEQ Versão 1.0.672 375



Tópicos

- Adicionando scripts BTEQ ao seu projeto AWS SCT
- Configurando variáveis de substituição em scripts BTEQ com AWS SCT
- Convertendo scripts Teradata BTEQ em Amazon Redshift RSQL com AWS SCT
- Gerenciando scripts BTEQ com AWS SCT
- Criando um relatório de avaliação de conversão de script BTEQ com AWS SCT
- Editando e salvando seus scripts BTEQ convertidos com AWS SCT

Scripts do Teradata BTEQ Versão 1.0.672 376

Adicionando scripts BTEQ ao seu projeto AWS SCT

Você pode adicionar vários scripts a um único AWS SCT projeto.

Para adicionar um script BTEQ ao seu projeto AWS SCT

- 1. Crie um novo projeto AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 <u>Bancos de dados do</u> Teradata.
- Escolha Adicionar destino no menu para adicionar um banco de dados de destino do Amazon Redshift ao seu AWS SCT projeto.
 - Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- 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 <u>Mapeando novos tipos de</u> dados no AWS Schema Conversion Tool.
- 5. No menu Visualizar, escolha Visualização principal.
- 6. No painel à esquerda, expanda o nó Scripts.
- Selecione scripts BTEQ, abra o menu de contexto (clique com o botão direito do mouse) e selecione Carregar scripts.
- Insira a localização do código-fonte para seus scripts Teradata BTEQ e escolha Selecionar pasta.
 - AWS SCT exibe a janela Carregar scripts.
- Execute um destes 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.
 - Se seus scripts Teradata BTEQ incluírem as variáveis de substituição, configure-as. Para obter mais informações, consulte <u>Como configurar variáveis de substituição em scripts</u> BTEQ.

Configurando variáveis de substituição em scripts BTEQ com 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 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. AWS SCT só podem resolver e converter variáveis de substituição depois de atribuir seus valores.

Para configurar variáveis de substituição em seus scripts BTEQ

 Adicione seus scripts BTEQ ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionando scripts BTEQ ao seu projeto 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+.

Expressões regulares em AWS SCT conformidade com a sintaxe da expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Java.

- 3. Escolha OK para carregar scripts em seu AWS SCT projeto 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.

Convertendo scripts Teradata BTEQ em Amazon Redshift RSQL com AWS SCT

A seguir, descubra como converter scripts BTEQ ETL em RSQL do Amazon Redshift usando. AWS SCT

Como converter um script Teradata BTEQ em Amazon Redshift RSQL

 Adicione seus scripts BTEQ ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionando scripts BTEQ ao seu projeto AWS SCT.

- Configure as variáveis de substituição. Para obter mais informações, consulte <u>Como configurar</u> variáveis de substituição em scripts BTEQ.
- 3. No painel à esquerda, expanda o nó Scripts.
- 4. Execute um destes 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.

AWS SCT converte todos os scripts selecionados do Teradata BTEQ 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.

Gerenciando scripts BTEQ com AWS SCT

Você pode adicionar vários scripts BTEQ ou remover um script BTEQ do seu projeto. AWS SCT

Para adicionar um script BTEQ adicional ao seu projeto 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).
- Escolha Carregar scripts.
- 4. Insira as informações necessárias para adicionar um novo script BTEQ e configurar as variáveis de substituição. Para ter mais informações, consulte <u>Adicionando scripts BTEQ ao seu projeto</u> <u>AWS SCT</u> e <u>Como configurar variáveis de substituição em scripts BTEQ</u>.

Para remover um script BTEQ do seu projeto 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.

Criando um relatório de avaliação de conversão de script BTEQ com 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 não AWS SCT podem ser convertidos.

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 de avaliação de conversão de script BTEQ 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 de avaliação de conversão de script BTEQ 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. Quando você escolhe um item de ação da lista, AWS SCT destaca o item do script BTEQ de origem ao qual o item de ação se aplica.

Editando e salvando seus scripts BTEQ convertidos com AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu AWS SCT projeto. AWS SCT armazena o script editado como parte do seu projeto.

Para salvar seus scripts convertidos

- 1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
- 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.

AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Conversão de scripts de shell com comandos BTEQ incorporados para o Amazon Redshift RSQL com AWS Schema Conversion Tool

Você pode usar o 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 RSQL incorporados do Amazon Redshift.

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 AWS SCT para converter arquivos com scripts Teradata BTEQ ETL para Amazon Redshift RSQL. Para obter mais informações, consulte Convertendo scripts Teradata BTEQ em Amazon Redshift RSQL com AWS SCT.

Tópicos

Adicionar scripts de shell com comandos Teradata BTEQ incorporados ao seu projeto AWS SCT

Configurando variáveis de substituição em scripts de shell com comandos Teradata BTEQ incorporados com AWS SCT

- Convertendo scripts de shell com comandos Teradata BTEQ incorporados com AWS SCT
- Gerenciando scripts de shell com comandos Teradata BTEQ incorporados com AWS SCT
- Criando um relatório de avaliação para uma conversão de script de shell com AWS SCT
- Editando e salvando seus scripts de shell convertidos com AWS SCT

Adicionar scripts de shell com comandos Teradata BTEQ incorporados ao seu projeto AWS SCT

Você pode adicionar vários scripts a um único AWS SCT projeto.

Para adicionar um script de shell ao seu AWS SCT projeto

- 1. Crie um novo projeto AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 <u>Bancos de dados do</u> Teradata.
- 3. Escolha Adicionar destino no menu e adicionar um banco de dados de destino do Amazon Redshift ao seu AWS SCT projeto.
 - Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte <u>Mapeamento para alvos virtuais no AWS Schema Conversion</u> Tool.
- 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 <u>Mapeando novos tipos de</u> dados no AWS Schema Conversion Tool.
- 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.
- Insira a localização dos scripts de shell com comandos Teradata BTEQ incorporados e escolha Selecionar pasta.

AWS SCT exibe a janela Carregar scripts.

- 9. Execute um destes 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.

Configurando variáveis de substituição em scripts de shell com comandos Teradata BTEQ incorporados com 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 AWS SCT para configurar variáveis de substituição em seus scripts de shell.

Antes de executar comandos BTEQ com variáveis de substituição de um script de shell, certifiquese de atribuir os valores para todas as variáveis dentro desse script de shell. AWS SCT só podem resolver e converter variáveis de substituição depois de atribuir seus valores.

Para configurar variáveis de substituição no seu script de shell

- Adicione seus scripts de shell de origem ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionar scripts de shell ao seu AWS SCT projeto.
 - 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+.
 - Expressões regulares em AWS SCT conformidade com a sintaxe da expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Java.
- 3. Escolha OK para carregar scripts em seu AWS SCT projeto 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.

Convertendo scripts de shell com comandos Teradata BTEQ incorporados com AWS SCT

A seguir, descubra como converter scripts de shell com comandos incorporados do Teradata BTEQ em scripts de shell com comandos RSQL incorporados do Amazon Redshift usando. AWS SCT

Para converter um script de shell

- Adicione seus scripts de shell ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionar scripts de shell ao seu AWS SCT projeto.
- 2. Configure as variáveis de substituição. Para obter mais informações, consulte Como configurar variáveis de substituição em scripts de shell.
- No painel à esquerda, expanda o nó Scripts.
- 4. Execute um destes 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.
 - AWS SCT converte comandos BTEQ em seus scripts de shell selecionados em um formato compatível com Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.
- Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte <u>Como editar e salvar seus scripts de shell convertidos</u>.

Gerenciando scripts de shell com comandos Teradata BTEQ incorporados com AWS SCT

Você pode adicionar ou remover vários scripts de shell do seu projeto da AWS SCT.

Para adicionar um novo script de shell ao seu AWS SCT projeto

- 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 ter mais informações, consulte <u>Adicionar scripts de shell ao seu</u> AWS SCT projeto e Como configurar variáveis de substituição em scripts de shell.

Para remover um script de shell do seu AWS SCT projeto

- 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.

Criando um relatório de avaliação para uma conversão de script de shell com 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 não AWS SCT podem ser convertidos.

Como criar um relatório de avaliação de conversão de script de shell

- 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.
- 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.
- (Opcional) 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. Quando você seleciona um item de ação na lista, AWS SCT destaca o item do script de shell de origem ao qual o item de ação se aplica.

Editando e salvando seus scripts de shell convertidos com AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu AWS SCT projeto. AWS SCT armazena o script editado como parte do seu projeto.

Para salvar seus scripts convertidos

- 1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
- 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.

AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Conversão de FastExport scripts para Amazon Redshift RSQL com AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter scripts de FastExport trabalho do Teradata em RSQL do Amazon Redshift.

Um script de FastExport trabalho é um conjunto de FastExport comandos e instruções SQL que selecionam e exportam dados do banco de dados Teradata. AWS SCT converte FastExport comandos 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

- Adicionar scripts de FastExport trabalho ao seu AWS SCT projeto
- Configurando variáveis de substituição em scripts de trabalho do Teradata FastExport com AWS
 SCT
- Convertendo scripts de trabalho do Teradata FastExport com AWS SCT
- Gerenciando scripts de FastExport trabalho da Teradata com AWS SCT
- Criando um relatório de avaliação para uma conversão de script de FastExport trabalho da Teradata com AWS SCT
- Editando e salvando seus scripts de FastExport trabalho convertidos do Teradata com AWS SCT

Adicionar scripts de FastExport trabalho ao seu AWS SCT projeto

Você pode adicionar vários scripts a um único AWS SCT projeto.

Para adicionar um script de FastExport trabalho ao seu AWS SCT projeto

- 1. Crie um novo projeto AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 <u>Bancos de dados do</u> Teradata.
- 3. Escolha Adicionar destino no menu e adicionar um banco de dados de destino do Amazon Redshift ao seu AWS SCT projeto.
 - Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- 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 <u>Mapeando novos tipos de</u> dados no AWS Schema Conversion Tool.
- 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 de FastExport trabalho do Teradata e escolha Selecionar pasta.

- AWS SCT exibe a janela Carregar scripts.
- 9. Execute um destes procedimentos:
 - Se seus scripts de FastExport trabalho do Teradata não incluírem as variáveis de substituição, escolha Sem variáveis de substituição e escolha OK para adicionar scripts ao seu projeto. AWS SCT
 - Se seus scripts de FastExport trabalho do Teradata incluírem as variáveis de substituição, configure as variáveis de substituição. Para obter mais informações, consulte <u>Configurando</u> variáveis de substituição em FastExport scripts de trabalho.

Configurando variáveis de substituição em scripts de trabalho do Teradata FastExport com AWS SCT

Seus scripts de FastExport trabalho do Teradata 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 AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Antes de executar um script de FastExport trabalho 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. AWS SCT só podem resolver e converter variáveis de substituição depois de atribuir seus valores.

Para configurar variáveis de substituição em seu script de FastExport trabalho

- Adicione seus scripts de FastExport trabalho Teradata de origem ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionando scripts BTEQ ao seu projeto 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+.

Expressões regulares em AWS SCT conformidade com a sintaxe da expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Java.

- 3. Escolha OK para carregar scripts em seu AWS SCT projeto e, em seguida, escolha OK para fechar a janela Carregar scripts.
- 4. No painel à esquerda, expanda o nó Scripts. Escolha e FastExport, 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, 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. AWS SCT não é possível importar o arquivo CSV com valores em formatos diferentes.
- 8. Salve o arquivo CSV.
- No painel à esquerda, expanda o nó Scripts. Escolha e FastExport, 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.
- 10. Escolha seu arquivo CSV e clique em Abrir.
- 11. Escolha Variáveis para ver todas as variáveis de substituição descobertas e seus valores.

Convertendo scripts de trabalho do Teradata FastExport com AWS SCT

A seguir, descubra como converter uma FastExport tarefa Teradata em RSQL do Amazon Redshift usando. AWS SCT

Para converter um script de FastExport trabalho do Teradata em Amazon Redshift RSQL

 Adicione seus scripts de FastExport trabalho ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionar scripts de FastExport trabalho ao seu AWS SCT projeto.

2. Configure as variáveis de substituição. Para obter mais informações, consulte <u>Configurando</u> variáveis de substituição em FastExport scripts de trabalho.

- 3. No painel à esquerda, expanda o nó Scripts.
- 4. Execute um destes procedimentos:
 - Para converter um único script de FastExport trabalho, expanda o FastExportnó, 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. Em seguida FastExport, escolha, abra o menu de contexto (clique com o botão direito do mouse) e escolha Converter script.
 - AWS SCT converte todos os scripts de FastExport trabalho selecionados do Teradata em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.
- Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte Editando e salvando seus scripts de FastExport trabalho convertidos.

Gerenciando scripts de FastExport trabalho da Teradata com AWS SCT

Você pode adicionar vários scripts de FastExport trabalho do Teradata ou remover um script de FastExport trabalho do seu AWS SCT projeto.

Para adicionar um novo script de FastExport trabalho ao seu AWS SCT projeto

- 1. No painel à esquerda, expanda o nó Scripts.
- 2. Escolha o FastExportnó e abra o menu de contexto (clique com o botão direito do mouse).
- 3. Escolha Carregar scripts.
- 4. Insira as informações necessárias para adicionar um novo script de FastExport trabalho e configurar as variáveis de substituição. Para ter mais informações, consulte <u>Adicionar scripts</u> de <u>FastExport trabalho ao seu AWS SCT projeto</u> e <u>Configurando variáveis de substituição em FastExport scripts</u> de trabalho.

Para remover um script de FastExport trabalho do seu AWS SCT projeto

Expanda o FastExportnó 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.

Criando um relatório de avaliação para uma conversão de script de FastExport trabalho da Teradata com AWS SCT

O relatório de avaliação de conversão do script de FastExport trabalho fornece informações sobre a conversão dos FastExport comandos e instruções SQL dos seus FastExport scripts em um formato compatível com o Amazon Redshift RSQL. O relatório de avaliação inclui itens de ação para FastExport comandos e instruções SQL que não AWS SCT podem ser convertidos.

Para criar um relatório de avaliação de conversão de script para um trabalho da FastExport Teradata

- 1. Expanda o FastExportnó 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 FastExport trabalho. Ele inclui resultados de conversão para todos os FastExport comandos e instruções SQL de seus scripts de origem.
- 4. Você pode salvar uma cópia local do relatório de avaliação de conversão do script de FastExport trabalho como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV).
 - a. Para salvar o relatório de avaliação de conversão do script de FastExport trabalho como um arquivo PDF, escolha 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.
 - b. Para salvar o relatório de avaliação de conversão do script de FastExport trabalho como um arquivo CSV, escolha Salvar em CSV no canto superior direito.
 - 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. Quando você seleciona um item de ação na lista, AWS SCT destaca o item do script de FastExport trabalho de origem ao qual o item de ação se aplica.

Editando e salvando seus scripts de FastExport trabalho convertidos do Teradata com AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu AWS SCT projeto. AWS SCT armazena o script editado como parte do seu projeto.

Para salvar seus scripts convertidos

- 1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
- 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.

AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Convertendo scripts de FastLoad trabalho em Amazon Redshift RSQL com AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para converter scripts de FastLoad trabalho do Teradata em RSQL do Amazon Redshift.

Um FastLoad script Teradata 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 FastLoad comandos Teradata e instruções SQL. Os FastLoad comandos Teradata fornecem controle de sessão e tratamento de dados das transferências de dados. As instruções SQL criam, mantêm e eliminam tabelas.

AWS SCT converte FastLoad comandos Teradata 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

- Adicionar scripts de FastLoad trabalho ao seu AWS SCT projeto
- Configurando variáveis de substituição em scripts de trabalho do Teradata FastLoad com AWS SCT

- Convertendo scripts de trabalho do Teradata FastLoad com AWS SCT
- Gerenciando scripts de FastLoad trabalho da Teradata com AWS SCT
- Criando um relatório de avaliação para uma conversão de script de FastLoad trabalho da Teradata com AWS SCT

Editando e salvando seus scripts de FastLoad trabalho convertidos do Teradata com AWS SCT

Adicionar scripts de FastLoad trabalho ao seu AWS SCT projeto

Você pode adicionar vários scripts a um único AWS SCT projeto.

Para adicionar um script de FastLoad trabalho ao seu AWS SCT projeto

- 1. Crie um novo projeto ou abra um projeto existente. AWS SCT Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 <u>Bancos de dados do</u> Teradata.
- Escolha Adicionar destino no menu e adicione um banco de dados de destino do Amazon Redshift ao seu AWS SCT projeto.
 - Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- 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 <u>Mapeando novos tipos de</u> dados no AWS Schema Conversion Tool.
- 5. No menu Visualizar, escolha Visualização principal.
- 6. No painel à esquerda, expanda o nó Scripts.
- Escolha FastLoad, abra o menu de contexto (clique com o botão direito do mouse) e escolha Carregar scripts.
- Insira a localização dos scripts de FastLoad trabalho do Teradata de origem e escolha Selecionar pasta.
 - AWS SCT exibe a janela Carregar scripts.
- Execute um destes procedimentos:

 Se seus scripts de FastLoad trabalho do Teradata 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 FastLoad trabalho do Teradata incluírem as variáveis de substituição, configure as variáveis de substituição. Para obter mais informações, consulte <u>Configurando</u> variáveis de substituição em FastLoad scripts de trabalho.

Configurando variáveis de substituição em scripts de trabalho do Teradata FastLoad com AWS SCT

Seus scripts de FastLoad trabalho do Teradata 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 FastLoad trabalho 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.

AWS SCT só podem resolver e converter variáveis de substituição depois de atribuir seus valores. Antes de iniciar a conversão dos scripts de FastLoad trabalho de origem do Teradata, certifique-se de atribuir valores para todas as variáveis de substituição. Você pode usar AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Para configurar variáveis de substituição em seu script de FastLoad trabalho

- Ao adicionar seus scripts de FastLoad trabalho de origem do Teradata ao seu AWS SCT projeto, escolha Variáveis de substituição são usadas. Para obter mais informações sobre como adicionar esses scripts, consulte <u>Adicionar scripts de FastLoad trabalho ao seu AWS SCT</u> projeto.
- 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+.

Expressões regulares em AWS SCT conformidade com a sintaxe da expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Java.

- Escolha OK para carregar scripts em seu AWS SCT projeto e, em seguida, escolha OK para fechar a janela Carregar scripts.
- 4. No painel à esquerda, expanda o nó Scripts. Escolha e FastLoad, 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. 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.
- 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, AWS SCT usa formatos diferentes para o arquivo 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. AWS SCT não é possível importar o arquivo CSV com valores em formatos diferentes.
- 7. Salve o arquivo CSV.
- 8. No painel à esquerda, expanda o nó Scripts. Escolha e FastLoad, 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.

Convertendo scripts de trabalho do Teradata FastLoad com AWS SCT

A seguir, descubra como converter uma FastLoad tarefa Teradata em RSQL do Amazon Redshift usando. AWS SCT

Para converter um script de FastLoad trabalho do Teradata em Amazon Redshift RSQL

1. Adicione seus scripts de FastLoad trabalho ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionar scripts de FastLoad trabalho ao seu AWS SCT projeto.

2. Configure as variáveis de substituição. Para obter mais informações, consulte <u>Configurando</u> variáveis de substituição em FastLoad scripts de trabalho.

- 3. No painel à esquerda, expanda o nó Scripts.
- 4. Execute um destes procedimentos:
 - Para converter um único script de FastLoad trabalho, expanda o FastLoadnó, 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 FastLoad, abra o menu de contexto (clique com o botão direito do mouse) e escolha 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.
- Escolha OK.
 - AWS SCT converte todos os scripts de FastLoad trabalho selecionados do Teradata 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 Editando e salvando seus scripts de FastLoad trabalho convertidos.

Gerenciando scripts de FastLoad trabalho da Teradata com AWS SCT

Você pode adicionar vários scripts de FastLoad trabalho do Teradata ou remover um script de FastLoad trabalho do seu AWS SCT projeto.

Para adicionar um novo script de FastLoad trabalho ao seu AWS SCT projeto

No painel à esquerda, expanda o nó Scripts.

2. Escolha o FastLoadnó e abra o menu de contexto (clique com o botão direito do mouse).

- 3. Escolha Carregar scripts.
- 4. Insira as informações necessárias para adicionar um novo script de FastLoad trabalho e configurar as variáveis de substituição. Para ter mais informações, consulte <u>Adicionar scripts</u> de <u>FastLoad trabalho ao seu AWS SCT projeto</u> e <u>Configurando variáveis de substituição em FastLoad scripts de trabalho.</u>

Para remover um script de FastLoad trabalho do seu AWS SCT projeto

- 1. Expanda o FastLoadnó 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.

Criando um relatório de avaliação para uma conversão de script de FastLoad trabalho da Teradata com AWS SCT

O relatório de avaliação da conversão do script de FastLoad trabalho fornece informações sobre a conversão dos FastLoad comandos e 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 FastLoad comandos e instruções SQL que não AWS SCT podem ser convertidos.

Para criar um relatório de avaliação de conversão de script para um trabalho da FastLoad Teradata

- 1. Expanda o FastLoadnó 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.
- Visualize a guia Resumo.
 - A guia Resumo exibe as informações do resumo executivo do relatório de avaliação do script de FastLoad trabalho. Ele inclui resultados de conversão para todos os FastLoad comandos e instruções SQL de seus scripts de origem.
- 4. (Opcional) Salve uma cópia local do relatório de avaliação de conversão do script de FastLoad trabalho como um arquivo PDF ou um arquivo de valores separados por vírgula (CSV):
 - Para salvar o relatório de avaliação de conversão do script de FastLoad trabalho como um arquivo PDF, escolha 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.

- Para salvar o relatório de avaliação de conversão do script de FastLoad trabalho como um arquivo CSV, escolha Salvar em CSV no canto superior direito.
 - 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. Quando você seleciona um item de ação na lista, AWS SCT destaca o item do script de FastLoad trabalho de origem ao qual o item de ação se aplica.

Editando e salvando seus scripts de FastLoad trabalho convertidos do Teradata com AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu AWS SCT projeto. AWS SCT armazena o script editado como parte do seu projeto.

Para salvar seus scripts convertidos

- 1. Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
- 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.
 - AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Conversão de MultiLoad scripts para Amazon Redshift RSQL com AWS Schema Conversion Tool

Você pode usar AWS SCT para converter scripts de MultiLoad trabalho do Teradata em RSQL do Amazon Redshift.

Um script de MultiLoad trabalho Teradata é um conjunto de comandos para manutenção em lote do seu banco de dados Teradata. Uma tarefa de MultiLoad importação do Teradata executa várias operações diferentes de inserção, atualização e exclusão em até cinco tabelas e visualizações

diferentes. As tarefas de MultiLoad exclusão do Teradata podem remover um grande número de linhas de uma única tabela.

AWS SCT converte MultiLoad comandos Teradata 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

- Adicionar scripts de MultiLoad trabalho ao seu AWS SCT projeto
- Configurando variáveis de substituição em scripts de trabalho do Teradata MultiLoad com AWS
 SCT
- Convertendo scripts de trabalho do Teradata MultiLoad com AWS SCT
- · Gerenciando scripts de MultiLoad trabalho da Teradata com AWS SCT
- Criando um relatório de avaliação para uma conversão de script de MultiLoad trabalho da Teradata com AWS SCT
- Editando e salvando seus scripts de MultiLoad trabalho convertidos do Teradata com AWS SCT

Adicionar scripts de MultiLoad trabalho ao seu AWS SCT projeto

Você pode adicionar vários scripts a um único AWS SCT projeto.

Para adicionar um script de MultiLoad trabalho ao seu AWS SCT projeto

- 1. Crie um novo projeto AWS SCT ou abra um projeto existente. Para obter mais informações, consulte the section called "Como iniciar e gerenciar projetos".
- 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 <u>Bancos de dados do</u> Teradata.
- 3. Escolha Adicionar destino no menu e adicionar um banco de dados de destino do Amazon Redshift ao seu AWS SCT projeto.

Você pode usar uma plataforma virtual de banco de dados de destino do Amazon Redshift. Para obter mais informações, consulte Mapeamento para alvos virtuais no AWS Schema Conversion Tool.

 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 <u>Mapeando novos tipos de</u> dados no AWS Schema Conversion Tool.

- 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 scripts de MultiLoad trabalho do Teradata de origem e escolha Selecionar pasta.
 - AWS SCT exibe a janela Carregar scripts.
- 9. Execute um destes procedimentos:
 - Se seus scripts de MultiLoad trabalho do Teradata 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 MultiLoad trabalho do Teradata incluírem as variáveis de substituição, configure as variáveis de substituição. Para obter mais informações, consulte <u>Configurando</u> variáveis de substituição em MultiLoad scripts de trabalho.

Configurando variáveis de substituição em scripts de trabalho do Teradata MultiLoad com AWS SCT

Seus scripts de MultiLoad trabalho do Teradata 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 MultiLoad trabalho 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.

AWS SCT só podem resolver e converter variáveis de substituição depois de atribuir seus valores. Antes de iniciar a conversão dos scripts de MultiLoad trabalho de origem do Teradata, certifiquese de ter atribuído valores para todas as variáveis de substituição. Você pode usar AWS SCT para configurar variáveis de substituição em seus scripts Teradata.

Para configurar variáveis de substituição em seu script de MultiLoad trabalho

 Ao adicionar seus scripts de MultiLoad trabalho de origem do Teradata ao seu AWS SCT projeto, escolha Variáveis de substituição são usadas. Para obter mais informações sobre como adicionar esses scripts, consulte <u>Adicionar scripts de MultiLoad trabalho ao seu AWS SCT</u> <u>projeto</u>.

- 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+.
 - Expressões regulares em AWS SCT conformidade com a sintaxe da expressão regular Java. Para obter mais informações, consulte java.util.regex Class Pattern na documentação do Java.
- 3. Escolha OK para carregar scripts em seu AWS SCT projeto 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.

Convertendo scripts de trabalho do Teradata MultiLoad com AWS SCT

A seguir, descubra como converter uma MultiLoad tarefa Teradata em RSQL do Amazon Redshift usando. AWS SCT

Para converter um script de MultiLoad trabalho do Teradata em Amazon Redshift RSQL

- Adicione seus scripts de MultiLoad trabalho ao seu AWS SCT projeto. Para obter mais informações, consulte Adicionar scripts de MultiLoad trabalho ao seu AWS SCT projeto.
- Configure as variáveis de substituição e insira seus valores. Para obter mais informações, consulte Configurando variáveis de substituição em MultiLoad scripts de trabalho.
- 3. No painel à esquerda, expanda o nó Scripts.
- Execute um destes procedimentos:
 - Para converter um único script de MultiLoad trabalho, expanda o MultiLoadnó, 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. Execute 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 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.
- Escolha OK.
 - AWS SCT converte todos os scripts de MultiLoad trabalho selecionados do Teradata em um formato compatível com o Amazon Redshift RSQL. Encontre seus scripts convertidos no nó Scripts, no painel do banco de dados de destino.
- Edite seus scripts Amazon Redshift RSQL convertidos ou salve-os. Para obter mais informações, consulte Editando e salvando seus scripts de MultiLoad trabalho convertidos.

Gerenciando scripts de MultiLoad trabalho da Teradata com AWS SCT

Você pode adicionar vários scripts de MultiLoad trabalho do Teradata ou remover um script de MultiLoad trabalho do seu AWS SCT projeto.

Para adicionar um novo script de MultiLoad trabalho ao seu AWS SCT projeto

- No painel à esquerda, expanda o nó Scripts.
- 2. Escolha o MultiLoadnó e abra o menu de contexto (clique com o botão direito do mouse).
- 3. Escolha Carregar scripts.
- 4. Insira as informações necessárias para adicionar um novo script de MultiLoad trabalho e configurar as variáveis de substituição. Para ter mais informações, consulte <u>Adicionar scripts</u> de <u>MultiLoad trabalho ao seu AWS SCT projeto</u> e <u>Configurando variáveis de substituição em MultiLoad scripts de trabalho.</u>

Para remover um script de MultiLoad trabalho do seu AWS SCT projeto

- 1. Expanda o MultiLoadnó 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.

Criando um relatório de avaliação para uma conversão de script de MultiLoad trabalho da Teradata com AWS SCT

O relatório de avaliação da conversão do script de MultiLoad trabalho fornece informações sobre a conversão dos MultiLoad comandos e instruções SQL. A conversão é feita dos seus scripts de origem para comandos do Amazon Redshift RSQL e para instruções SQL do Amazon Redshift. O relatório de avaliação inclui itens de ação para MultiLoad comandos e instruções SQL que não AWS SCT podem ser convertidos.

Para criar um relatório de avaliação de conversão de script para um trabalho da MultiLoad Teradata

- 1. Expanda o MultiLoadnó 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 MultiLoad trabalho. Ele inclui resultados de conversão para todos os MultiLoad comandos e instruções SQL de seus scripts de origem.
- (Opcional) Salve uma cópia local do relatório de avaliação de conversão do script de MultiLoad trabalho como um arquivo PDF ou em arquivos de valores separados por vírgula (CSV):
 - Para salvar o relatório de avaliação de conversão do script de MultiLoad trabalho como um arquivo PDF, escolha 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.
 - Para salvar o relatório de avaliação de conversão do script de MultiLoad trabalho como arquivos CSV, escolha Salvar em CSV no canto superior direito.
 - 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. Quando você seleciona um item de ação na lista, AWS SCT destaca o item do script de MultiLoad trabalho de origem ao qual o item de ação se aplica.

Editando e salvando seus scripts de MultiLoad trabalho convertidos do Teradata com AWS SCT

Você pode editar seus scripts convertidos no painel inferior do seu AWS SCT projeto. AWS SCT armazena o script editado como parte do seu projeto.

Para salvar seus scripts convertidos

- Expanda o nó de scripts RSQL em Scripts no painel do banco de dados de destino.
- 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.
 - AWS SCT salva o script convertido em um arquivo e abre esse arquivo.

Migrando estruturas de big data com AWS Schema Conversion Tool

Você pode usar o AWS Schema Conversion Tool (AWS SCT) para migrar estruturas de big data para o. Nuvem AWS

Atualmente, 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 AWS SCT para automatizar a conversão de seus fluxos de trabalho de orquestração do Apache Oozie em. AWS Step Functions

Tópicos

- Migração de cargas de trabalho do Hadoop para o Amazon EMR com AWS Schema Conversion
 Tool
- Convertendo fluxos de trabalho do Oozie em com AWS Step FunctionsAWS Schema Conversion
 Tool

Migração de cargas de trabalho do Hadoop para o Amazon EMR com AWS Schema Conversion Tool

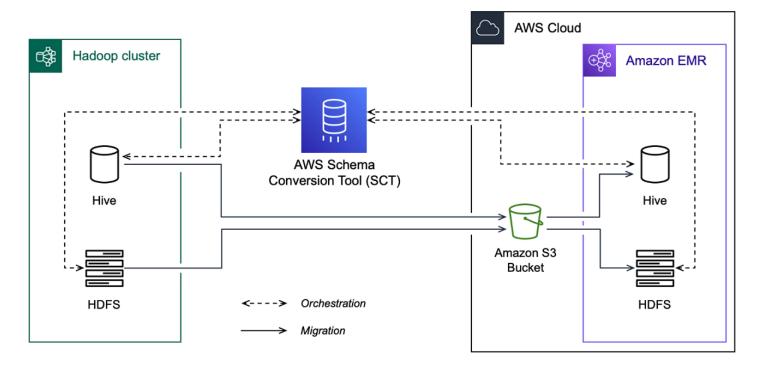
Para migrar clusters do Apache Hadoop, certifique-se de usar a AWS SCT versão 1.0.670 ou superior. Além disso, familiarize-se com a interface de linha de comandos (CLI) da AWS SCT. Para obter mais informações, consulte Referência de CLI para AWS Schema Conversion Tool.

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: Migre seu cluster Apache Hadoop para o Amazon EMR com 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.



AWS SCT migra dados e metadados do seu cluster 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, AWS SCT migra seus metadados de origem do Hive para o. AWS Glue Data Catalog

Em seguida, você pode usar 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 Hapood, você cria e executa seu script CLI 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.

Visão geral Versão 1.0.672 406

Etapa 1: conectar-se aos clusters do Hadoop

Para iniciar a migração do seu cluster Apache Hadoop, crie um novo projeto. AWS SCT Em seguida, conecte-se aos clusters de origem e de destino. Certifique-se de criar e provisionar seus AWS recursos de destino antes de iniciar a migração.

Nesta etapa, você usa os seguintes comandos da AWS SCT CLI.

- CreateProject— para criar um novo AWS SCT projeto.
- 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 da AWS SCT CLI, consulte. <u>Como se conectar ao</u> Apache Hadoop

Quando você executa o comando que se conecta a um cluster de origem ou de destino, AWS SCT tenta estabelecer a conexão com esse cluster. Se a tentativa de conexão falhar, AWS SCT interrompe a execução dos comandos do script da CLI e exibirá 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 AWS SCT projeto. Para obter mais informações sobre regras de mapeamento, consulte Mapeando tipos de dados no AWS Schema Conversion Tool.

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 testab de origem e o cluster EMR de destino.

```
AddServerMapping
-sourceTreePath: 'Clusters.HAD00P_SOURCE.HIVE_SOURCE.Databases.testdb'
-targetTreePath: 'Clusters.HAD00P_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 AWS SCT projeto. 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 Relatório de avaliação da.

Para esta etapa, use o comando CreateMigrationReport. Esse comando usa dois parâmetros. O parâmetrotreePath 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 comotrue, 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 os arquivos CSV são AWS SCT salvos.
- 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: Migre seu cluster Apache Hadoop para o Amazon EMR com AWS SCT

Depois de configurar seu AWS SCT projeto, inicie a migração do seu cluster Apache Hadoop local para o. 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 comotrue, carrega AWS SCT automaticamente as árvores de metadados do banco de dados durante a migração. O valor padrão é false.
- forceMigrate— quando definido comotrue, 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.HAD00P_SOURCE.HDFS_SOURCE'
  -forceMigrate: 'true'
/
```

Como executar seu script de CLI

Depois de terminar de editar seu script de AWS SCT CLI, salve-o como um arquivo com a .scts extensão. Agora, você pode executar seu script a partir da app pasta do caminho de AWS SCT instalação. Para fazer isso, use o comando a seguir.

```
{\tt RunSCTBatch.cmd --pathtoscts "C: $$ script\_path$$ \hadoop.scts"}
```

No exemplo anterior, *script_path* substitua pelo caminho do seu arquivo pelo script CLI. Para obter mais informações sobre a execução de scripts de CLI em 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 AWS SCT projeto para uso futuro.

Para salvar seu AWS SCT projeto, use o SaveProject comando. Este comando não usa parâmetros.

O exemplo de código a seguir salva seu AWS SCT projeto.

```
SaveProject /
```

Para abrir seu AWS SCT projeto, use o OpenProject comando. Este comando usa um parâmetro obrigatório. Para o file parâmetro, insira o caminho para o arquivo do AWS SCT projeto 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 AWS SCT projeto, 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, <code>hadoop_address</code> substitua pelo endereço IP do seu cluster Hadoop. Se necessário, configure o valor da variável de porta. Em seguida, substitua <code>hadoop_user</code> e <code>hadoop_password</code> pelo nome do seu usuário do Hadoop e a senha desse usuário. Para<code>path</code> <code>\name</code>, 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 <code>Conectando-se aos bancos</code> de dados do Apache Hadoop com o AWS Schema Conversion Tool.

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.

Convertendo fluxos de trabalho do Oozie em com AWS Step FunctionsAWS Schema Conversion Tool

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

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: Converta seus fluxos de trabalho do Apache Oozie em com AWS Step FunctionsAWS
 SCT
- Como executar seu script de CLI
- Nodos Apache Oozie que AWS SCT podem ser convertidos em 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.

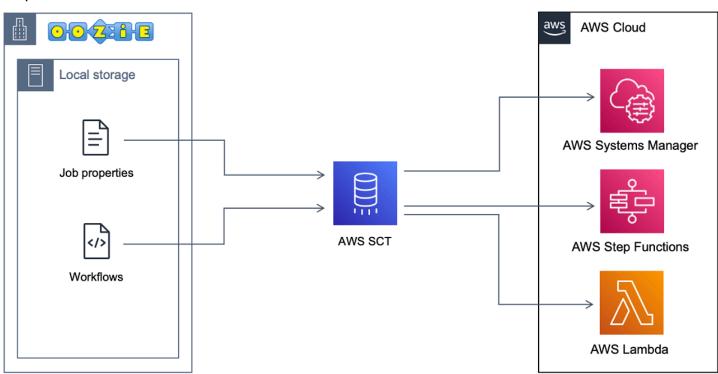
AWS SCT converte seus nós de ação de origem e nós de fluxo de controle em AWS Step Functions. Em AWS Step Functions, você define seus fluxos de trabalho na Amazon States Language (ASL). AWS SCT usa ASL para definir sua máquina de estados, que é uma coleção de estados que pode funcionar, determinar para quais estados fazer a transição a seguir, parar com um erro e assim por diante. Em seguida, AWS SCT carrega os arquivos JSON com as definições das máquinas de estado. Em seguida, AWS SCT pode usar sua função AWS Identity and Access Management (IAM) para configurar suas máquinas de estado em AWS Step Functions. Para obter mais informações, consulte O que é AWS Step Functions? no Guia do AWS Step Functions desenvolvedor.

Além disso, AWS SCT cria um pacote de extensão com AWS Lambda funções que emulam as funções de origem que AWS Step Functions não são suportadas. Para obter mais informações, consulte Usando pacotes de extensão com AWS Schema Conversion Tool.

AWS SCT migra suas propriedades de trabalho de origem para o. AWS Systems Manager Para armazenar nomes e valores de parâmetros, AWS SCT usa Parameter Store, um recurso de AWS Systems Manager. Para obter mais informações, consulte O que é AWS Systems Manager? no Guia do AWS Systems Manager usuário.

Você pode usar 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. 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. AWS Step Functions



Para iniciar a conversão, crie e execute seu script de AWS SCT CLI. 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.

Visão geral Versão 1.0.672 415

Etapa 1: conectar-se aos seus serviços de origem e destino

Para iniciar a conversão do seu cluster Apache Oozie, crie um novo AWS SCT projeto. Em seguida, conecte-se aos seus serviços de origem e destino. Certifique-se de criar e provisionar seus AWS recursos 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 da AWS SCT CLI.

- CreateProject— para criar um novo AWS SCT projeto.
- 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 da AWS SCT CLI, consulte. Como se conectar ao Apache Oozie

Quando você executa ConnectTarget os comandos ConnectSource ou, AWS SCT tenta estabelecer a conexão com seus serviços. Se a tentativa de conexão falhar, AWS SCT interrompe a execução dos comandos do script da CLI e exibirá 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 Mapeando tipos de dados no AWS Schema Conversion Tool.

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 00ZIE e AWS_STEP_FUNCTIONS. Você adicionou esses objetos ao seu AWS SCT projeto usando AddSource AddTarget comandos na etapa anterior.

```
AddServerMapping
-sourceTreePath: 'ETL.APACHE_00ZIE'
-targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Para obter mais informações sobre o comando AddServerMapping, consulte <u>Referência da CLI do</u> 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, talvez seja necessário adicionar novos parâmetros para usar com 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 <u>Referência da CLI da AWS Schema</u> 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 Relatório de avaliação da.

Para esta etapa, use o comando CreateReport. Esse comando usa dois parâmetros. O primeiro parâmetro descreve os objetos de origem para os quais 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 comotrue, carregará AWS SCT automaticamente todos os objetos secundários do objeto de origem que você especificar no CreateReport comando.

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 os arquivos CSV são AWS SCT salvos.
- 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: Converta seus fluxos de trabalho do Apache Oozie em com AWS Step FunctionsAWS SCT

Depois de configurar seu AWS SCT projeto, converta seu código-fonte e aplique-o ao 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 comotrue, carrega AWS SCT 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_00ZIE.Applications'
/
```

O Save0nS3 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 State machine definitions pasta do seu objeto de AWS_STEP_FUNCTIONS destino no bucket do Amazon S3. AWS SCT usa o bucket do Amazon S3 que você armazenou no perfil AWS de serviço na Pré-requisitos etapa.

```
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 no AWS_STEP_FUNCTIONS uso da função do *role_name* IAM.

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 app_wp estado AWS Step Functions a.

```
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 AWS Lambda funções que emulam suas funções do Apache Oozie que AWS Step Functions não suportam. Para instalar esse pacote de extensão, você pode usar o comando CreateLambdaExtPack.

Esse comando usa até cinco parâmetros. Certifique-se de usar o **0ozie2SF** para extPackId. Nesse caso, 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 AWS SCT pode armazenar arquivos temporários. Esse parâmetro é obrigatório.
- awsProfile— o nome do seu AWS perfil.
- lambdaExecRoles— a lista de nomes de recursos da Amazon (ARNs) das funções de execução a serem usadas para funções 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 <u>Permissões para usar AWS Lambda funções no pacote de extensões</u>.

Para obter mais informações sobre os comandos Convert, SaveOnS3, ConfigureStateMachine, ApplyToTarget e CreateLambdaExtPack, consulte a <u>Referência</u> da CLI da AWS Schema Conversion Tool.

Como executar seu script de CLI

Depois de terminar de editar seu script de AWS SCT CLI, salve-o como um arquivo com a .scts extensão. Agora, você pode executar seu script a partir da app pasta do caminho de AWS SCT instalação. Para fazer isso, use o comando a seguir.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

No exemplo anterior, *script_path* substitua pelo caminho do seu arquivo pelo script CLI. Para obter mais informações sobre a execução de scripts de CLI em AWS SCT, consulte. <u>Modo de script</u>

Nodos Apache Oozie que AWS SCT podem ser convertidos em AWS Step Functions

Você pode usar AWS SCT para converter nós de ação e nós de fluxo de controle do Apache Oozie em. AWS Step Functions

Os nós de ação suportados são os seguintes:

- Ação Hive
- Ação Hive2
- Ação Spark
- MapReduce Ação de streaming
- Ação Java
- DistCp ação
- 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

Nós suportados Versão 1.0.672 424

Integrando com AWS Database Migration ServiceAWS Schema Conversion Tool

Usando um agente de AWS SCT replicação com AWS DMS

Para migrações de bancos de dados muito grandes, você pode usar um agente de AWS SCT replicação (aws-schema-conversion-tool-dms-agent) para copiar dados do seu banco de dados local para o Amazon S3 ou um dispositivo Edge. AWS Snowball Edge O agente de replicação trabalha em conjunto AWS DMS e pode funcionar em segundo plano enquanto AWS SCT está fechado.

Ao trabalhar com o AWS Snowball Edge Edge, o AWS SCT agente replica os dados para o AWS Snowball Edge dispositivo. O dispositivo é então enviado AWS e os dados são carregados em um bucket do Amazon S3. Durante esse período, o AWS SCT agente continua funcionando. O agente recolhe os dados no Amazon S3 e copia-os em um endpoint de destino.

Para obter mais informações, consulte <u>Migração de dados do armazém de dados local para o</u> Amazon Redshift com AWS Schema Conversion Tool.

Usando um agente AWS SCT de extração de dados com AWS DMS

Em 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 AWS SCT para automatizar o processo de migração. Cassandra-to-DynamoDB AWS SCT integra-se com AWS Database Migration Service (AWS DMS) para realizar a migração real.

Para obter mais informações, consulte <u>Migração de dados do armazém de dados local para o</u> Amazon Redshift com AWS Schema Conversion Tool.

Aumentar os níveis de registro ao usar AWS SCT com AWS DMS

Você pode aumentar os níveis de registro ao usar AWS SCT com AWS DMS, por exemplo, se precisar trabalhar com o AWS Support.

Após a instalação AWS SCT e os drivers necessários, abra o aplicativo escolhendo o AWS SCT ícone. 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 AWS SCT com 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)
 - Crítico
 - Obrigatório
- 6. Escolha Aplicar para modificar as configurações do seu projeto.
- 7. Escolha OK para fechar a janela Configurações globais.

Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool

Você pode usar um AWS SCT agente para extrair dados do seu armazém de dados local e migrálos 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 Edge 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 <u>Usar um banco de dados do Amazon Redshift como destino</u> 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.

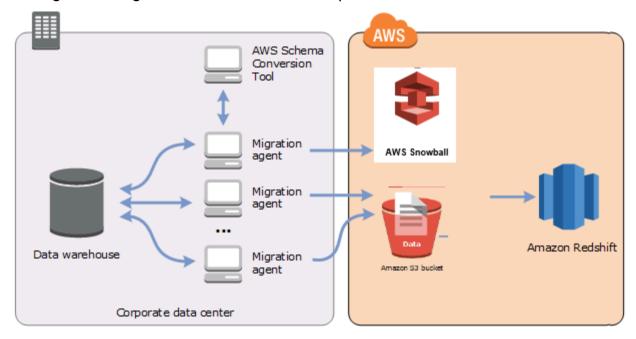
Migrações em larga escala

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 Edge 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 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 Edge.

Ao usar AWS SCT um dispositivo AWS Snowball Edge 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 Edge. Em seguida, você envia o dispositivo para AWS usar o processo AWS Snowball Edge 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 Edge
- Saída da tarefa de extração de dados
- Usando o particionamento virtual com AWS Schema Conversion Tool
- Como usar o particionamento nativo
- Migração 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 obter 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, escolha Configurações globais para criar um novo perfil. Para obter mais informações, consulte Gerenciando perfis no AWS Schema Conversion Tool.

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"
```

Pré-requisitos Versão 1.0.672 430

```
{
             "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.

Como assumir perfis do IAM Versão 1.0.672 431

2. 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 para Java.

- 3. Crie um novo perfil do IAM que tenha acesso ao seu bucket do Amazon S3.
- 4. 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"
}
```

5. Modifique a seção confiável desse perfil para confiar em redshift.amazonaws.com para assumir o perfil.

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.

Como assumir perfis do IAM Versão 1.0.672 432

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.

Configurações de segurança Versão 1.0.672 433

Para executar seu agente de extração de dados, certifique-se de usar um host com pelo menos quatro V CPUs e 32 GB de memória. Além disso, defina a memória mínima disponível AWS SCT para pelo menos quatro GB. Para obter 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

- Se você ainda não baixou o arquivo AWS SCT do instalador, siga as instruções em <u>Instalando e</u> <u>configurando AWS Schema Conversion Tool</u> para baixá-lo. O arquivo.zip que contém o arquivo do AWS SCT instalador também contém o arquivo instalador do agente de extração.
- Faça download e instale a versão mais recente do Amazon Corretto 11. Para obter mais informações, consulte <u>Downloads do Amazon Corretto 11</u> no Guia do usuário do Amazon Corretto 11.
- 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.

Como instalar agentes do Versão 1.0.672 434

Sistema operacional	Nome do arquivo
Microsoft Windows	<pre>aws-schema-conversion-tool-extractor -2.0.1. build-number .msi</pre>
RHEL	<pre>aws-schema-conversion-tool-extractor -2.0.1. build-number .x86_64.rpm</pre>
Ubuntu Linux	<pre>aws-schema-conversion-tool-extractor -2.0.1. build-number .deb</pre>

- 4. 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	Execute os seguintes comandos na pasta para a qual você baixou ou moveu o arquivo.
	<pre>sudo rpm -ivh aws-schema-conversion-tool-extractor -2.0.1. build-number .x86_64.rpm sudo ./sct-extractor-setup.shconfig</pre>
Ubuntu Linux	Execute os seguintes comandos na pasta para a qual você baixou ou moveu o arquivo.
	<pre>sudo dpkg -i aws-schema-conversion-tool-extractor -2.0.1. build-number .deb sudo ./sct-extractor-setup.shconfig</pre>

- 6. Selecione Avançar, aceite o contrato de licença e depois Avançar.
- 7. Insira o caminho para instalar o agente AWS SCT de extração de dados e escolha Avançar.
- 8. Selecione Instalar para instalar seu atendente de extração de dados.

Como instalar agentes do Versão 1.0.672 435

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 obter 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

- 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

Como configurar atendentes Versão 1.0.672 436

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.

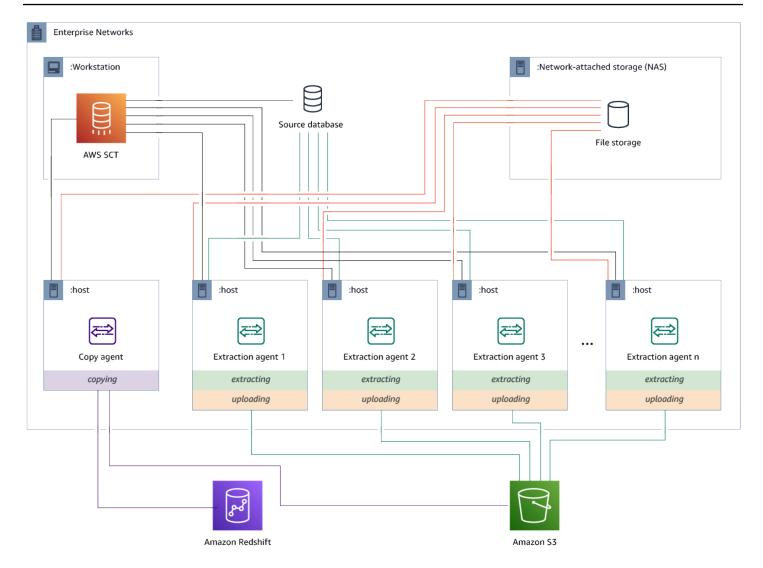
Veja a seguir um exemplo de arquivo de configurações.

```
$ 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 emComo 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 StartAgent.bat .
RHEL	Execute o seguinte comando no caminho para a pasta em que você instalou o agente:
	sudo initctl <i>start</i> sct-extractor
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: sudo initctl <i>start</i> sct-extractor
	Ubuntu 15.04 e superior: sudo systemctl <i>start</i> sct-extra ctor

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.

Como iniciar atendentes Versão 1.0.672 439

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.
- 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 Registrar.

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.
- 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 registrar atendentes Versão 1.0.672 440

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 ao 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.
- 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 RedshiftC0PY.

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 classific ação: Use a classificação após a primeira falha (recomendado), Use a classificação, se possível, ou Nunca use a classificação. Para obter 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 Edge. 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 suportada s (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. 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 bucket do Amazon S3 LOBs	Insira o local para armazenar os agentes de AWS SCT extração 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 localment 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

Como classificar dados Versão 1.0.672 446

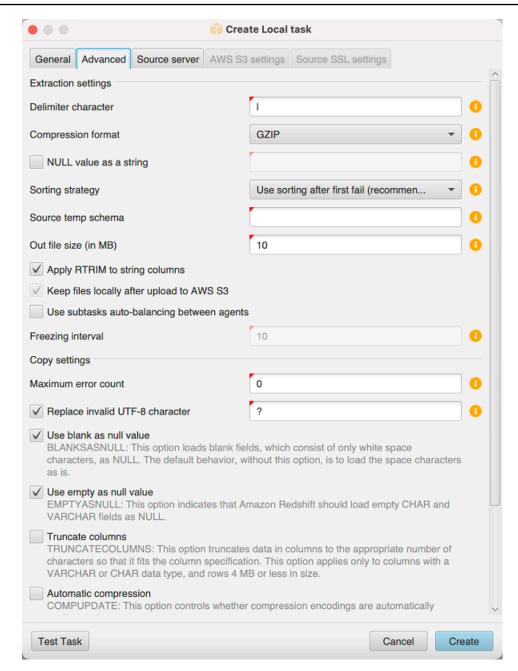
falha. Além disso, se você estiver migrando dados para o Amazon Redshift e classificar os dados primeiro AWS SCT, poderá inseri-los no Amazon Redshift com mais rapidez.

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 os 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.

Como classificar dados Versão 1.0.672 447



 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.
 AWS SCT gera chaves de criptografia automaticamente e as transmite para agentes de extração de dados usando Secure Sockets Layer (SSL). AWS SCT não criptografa objetos grandes (LOBs) durante a migração de dados.
- 6. Escolha LOBsExtrair 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.

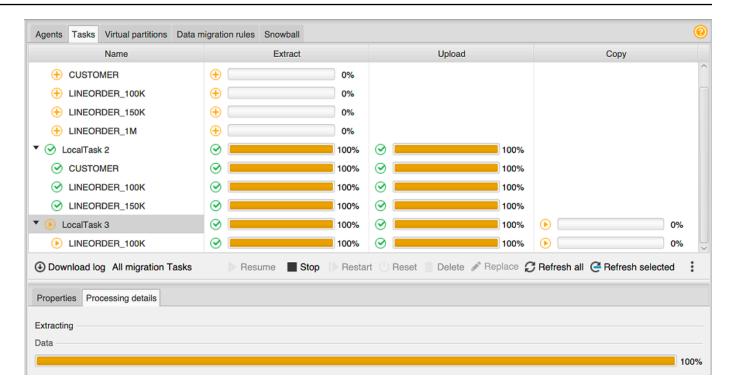
Se você habilitar o log de tarefas, escolha o nível de detalhes que você deseja ver. Os níveis são os seguintes, com cada nível incluindo todas as mensagens do nível anterior:

- 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 o 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.



- 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
- Tarefa e subtarefa IDs
- 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.
- 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.
- Selecione os três pontos alinhados verticalmente (ícone de reticências) no canto inferior direito abaixo da lista de tarefas.
- 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 Edge

O processo de uso AWS SCT do AWS Snowball Edge 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 Edge e, em seguida, uma ação intermediária em que AWS copia os dados do dispositivo AWS Snowball Edge 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 Edge.

- 1. Crie uma tarefa AWS Snowball Edge do Edge usando o AWS Snowball Edge console.
- 2. Desbloqueie o dispositivo AWS Snowball Edge 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.
- Crie e defina permissões para o bucket do Amazon S3 usar.
- 6. Importe um AWS Snowball Edge trabalho para o seu AWS SCT projeto.
- 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.

Step-by-step procedimentos para migrar dados usando um AWS SCT Edge AWS Snowball Edge

As seções a seguir apresentam informações detalhadas sobre as etapas de migração.

Etapa 1: criar uma tarefa AWS Snowball Edge Edge

Crie um AWS Snowball Edge trabalho seguindo as etapas descritas na seção <u>Criando um trabalho</u> <u>de AWS Snowball Edge borda</u> no Guia do desenvolvedor do AWS Snowball Edge Edge.

Etapa 2: desbloquear o dispositivo AWS Snowball Edge Edge

Execute os comandos que desbloqueiam e fornecem credenciais para o dispositivo Snowball Edge 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 Edge. Para obter mais informações sobre como desbloquear o dispositivo AWS Snowball Edge Edge, consulte Desbloquear o Snowball Edge 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

- 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.
- 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 obter mais informações, consulte <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u>.

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 <u>Visão geral dos</u> buckets no Guia do usuário do Amazon Simple Storage Service.

Etapa 6: importar um AWS Snowball Edge trabalho para seu AWS SCT projeto

Para conectar seu AWS SCT projeto ao seu dispositivo AWS Snowball Edge Edge, importe seu AWS Snowball Edge trabalho.

Para importar seu AWS Snowball Edge trabalho

 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 Edge emprego.
- 4. Insira seu IP do AWS Snowball Edge . Para obter mais informações, consulte <u>Como alterar seu</u> <u>Endereço IP</u> no Guia do usuário do AWS Snowball Edge .
- Entre no seu AWS Snowball Edge porto. Para obter mais informações, consulte <u>Portas</u> <u>necessárias para usar AWS serviços em um dispositivo AWS Snowball Edge Edge</u> no Guia do desenvolvedor do AWS Snowball Edge Edge.
- 6. Insira sua chave de acesso do AWS Snowball Edge e a chave secreta do AWS Snowball Edge .

 Para obter mais informações, consulte <u>Autorização e controle de acesso no AWS Snowball Edge</u> no Guia do usuário do AWS Snowball Edge .
- 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

- 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 Registrar.

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 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 Edge.
- 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 estabelecer conexões com o banco de dados de origem, o bucket do Amazon S3, o AWS Snowball Edge 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 Local dos dados migração carregar dados no Amazon Redshift. Para obter mais informações, consulte Como carregar dados do Amazon S3 na documentação do Amazon Redshift. 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. Amazon S3 > extractortest > > 5204158f4823 **Properties Permissions** Management Objects Q Type a prefix and press Enter to search. Press ESC to clear. Upload Create folder More v **Deleted objects** Name dms_sample_person_chunk_10.csv.lzo dms_sample_person_chunk_9.csv.lzo unit.manifest Extrair Os agentes de extração salvaram os dados como arquivos em sua pasta de somente trabalho. Copie manualmente os dados para o seu bucket do Amazon S3 e, em seguida, prossiga com as instruções para Extrair e carregar.

Usando o 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 obter 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 especificas 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

- 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.

Escolha OK.

Tipo de Partição RANGE Versão 1.0.672 460

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 especificas 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

- 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.

Tipo de partição LIST Versão 1.0.672 461

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 particion amento são armazenados.
Carregar do arquivo	(Opcional) Digite o nome de um arquivo .csv que contém os valores de partição.

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

- 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.

 Crie um novo projeto, adicione seus servidores de origem e de destino e crie regras de mapeamento. Para obter mais informações, consulte <u>Iniciando e gerenciando projetos em AWS</u> 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 obter 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 obter 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.

Migração LOBs para o Amazon Redshift

O Amazon Redshift não oferece suporte ao armazenamento de objetos binários grandes ()LOBs. No entanto, se você precisar migrar um ou mais LOBs para o Amazon Redshift AWS SCT, poderá realizar a migração. Para fazer isso, AWS SCT usa um bucket do Amazon S3 para armazenar LOBs e grava a URL do bucket do Amazon S3 nos dados migrados armazenados no Amazon Redshift.

Trabalhando com LOBs Versão 1.0.672 464

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.
- Para a LOBs pasta de bucket do Amazon S3, insira o nome da pasta em um bucket do Amazon S3 onde você deseja armazená-la. LOBs
 - Se você usa o perfil de AWS 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:
	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
temporanamente	alguns casos, o agente pode se recuperar. Neste caso, o status de suas tarefas é atualizado na AWS SCT.
Um agente é desativado permanentemente	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 computado r que o agente original. Para substituir um novo agente, faça o seguinte:
	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.

Convertendo o SQL do aplicativo usando 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 o AWS Schema Conversion Tool (AWS SCT) para converter o código SQL em seu C++, C#, Java ou outro código de aplicativo. Para uma conversão de Oracle para PostgreSQL, você pode AWS SCT usar 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
- Convertendo código SQL em aplicativos C# com AWS Schema Conversion Tool
- · Convertendo código SQL em aplicativos C++ com AWS Schema Conversion Tool
- Convertendo código SQL em aplicativos Java com AWS Schema Conversion Tool
- Convertendo código SQL em aplicativos Pro*C com AWS Schema Conversion Tool

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 Criação de projetos genéricos de conversão
 de aplicativos no 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 <u>Analisando e</u> <u>convertendo seu código SQL em AWS SCT</u>.
- 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 Criando e usando o relatório AWS SCT de avaliação no 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 Editando e salvando seu código SQL convertido com AWS SCT.

Como converter o código SQL em seus aplicativos com a AWS SCT

Você pode usar AWS SCT para converter 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 ter mais informações, consulte Código SQL em aplicativos C#, Código SQL em Java e Código SQL no Pro*C.

Criação de projetos genéricos de conversão de aplicativos no AWS SCT

No AWS Schema Conversion Tool, o projeto de conversão do aplicativo é filho do projeto de conversão do 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.



Note

AWS SCT não oferece suporte à conversão entre as seguintes fontes e destinos:

- Oracle para Oracle
- PostgreSQL para PostgreSQL ou Aurora PostgreSQL

Código SQL Versão 1.0.672 468

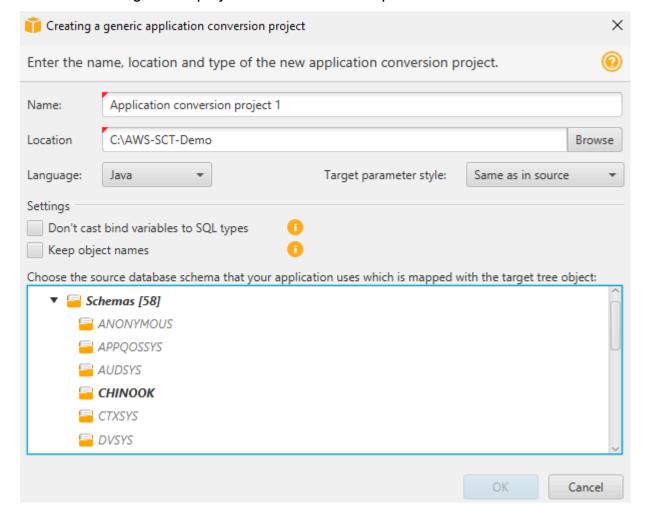
- MySQL para MySQL
- · SQL Server para SQL Server
- · Amazon Redshift para Amazon Redshift
- SQL Server para Babelfish
- Serviços de integração do SQL Server para AWS Glue
- Apache Cassandra para o Amazon DynamoDB

Use o procedimento a seguir para criar um projeto genérico de conversão de aplicativos.

Para criar um projeto de conversão de aplicativos

No AWS Schema Conversion Tool, escolha Novo aplicativo genérico no menu Aplicativos.

A caixa de diálogo Novo projeto de conversão de aplicativo será exibida.



2. Adicione as seguintes informações de projeto.

Para este parâmetro	Faça o seguinte
Nome	Digite um nome para o seu projeto de conversão de aplicativ os. 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.
Idioma	Escolha uma das seguintes opções: • Java • C++ • C# • Quaisquer
Estilo do parâmetro de destino	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: O mesmo da origem Posicional (?) Indexado (:1) Indexado (\$1) Nomeadas (@ nome) Nomeadas (: nome) Nomeadas (*nome) Nomeadas (\$nome) Nomeadas (#nome) Nomeadas (#nome) Nomeadas (!nome!)

Para este parâmetro	Faça o seguinte
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.

 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
```

 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
```

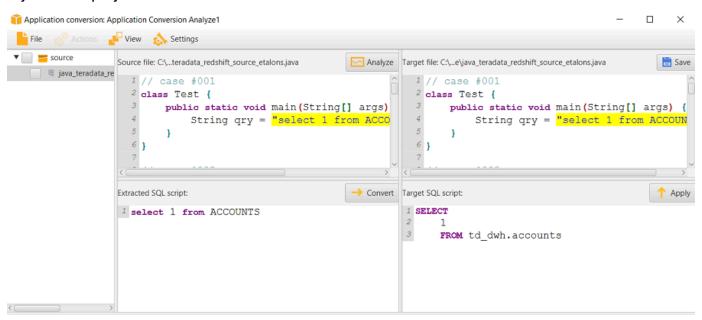
Quando você desmarca Manter nomes de objetos, a AWS SCT adiciona o nome do esquema ao nome da tabela. O resultado da conversão é mostrado a seguir.

```
SELECT * FROM schema_name.account
```

Se o código-fonte incluir os nomes dos objetos principais nos nomes dos objetos, AWS SCT use 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.



Gerenciando projetos de conversão de aplicativos no 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

 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

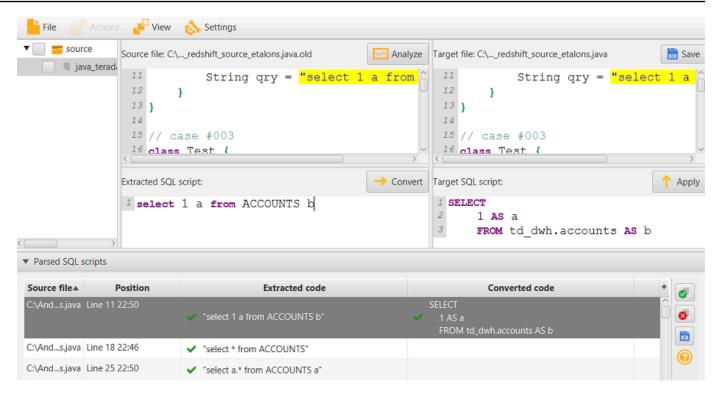
- No painel esquerdo, escolha o nó do projeto de conversão do aplicativo e abra o menu de contexto (clique com o botão direito do mouse).
- 2. Escolha New application (Nova aplicação).
- 3. Insira as informações necessárias para criar um novo projeto de conversão de aplicativos. Para obter mais informações, consulte Como criar projetos genéricos de conversão de aplicativos.

Analisando e convertendo seu código SQL em 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.
 - AWS SCT analisa o código do aplicativo e extrai o código SQL. AWS SCT exibe o código SQL extraído na lista de scripts SQL analisados.
- Para scripts SQL analisados, escolha um item para revisar o código SQL extraído. 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. 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.



 Quando você cria um relatório de avaliação de conversão de aplicativos, AWS SCT converte todos os itens de código SQL extraídos. Para obter mais informações, consulte <u>Como criar e</u> usar o relatório de avaliação.

Criando e usando o relatório AWS SCT de avaliação no 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 os itens de ação do código SQL que não AWS SCT podem ser convertidos.

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.
 - 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.



Selecione Ações de extração de SQL.

Examine a lista de itens de código SQL que não AWS SCT podem ser extraídos do seu códigofonte.

Selecione Ações de conversão de SQL.

Examine a lista de itens de código SQL que não AWS SCT podem ser convertidos automaticamente. Use as ações recomendadas para converter manualmente o código SQL. Para obter informações sobre como converter o código SQL, consulte Editando e salvando seu código SQL convertido com 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.

Editando e salvando seu código SQL convertido com AWS SCT

O relatório de avaliação inclui uma lista de itens de código SQL que não AWS SCT podem ser convertidos. Para cada item, 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

- 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.

Convertendo código SQL em aplicativos C# com AWS Schema Conversion Tool

Para uma conversão de Oracle para PostgreSQL, você pode AWS Schema Conversion Tool usar AWS SCT() para converter código SQL incorporado em seus 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.

Criação de projetos de conversão de aplicativos C# em 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 Mapeando tipos de dados no AWS Schema Conversion Tool.

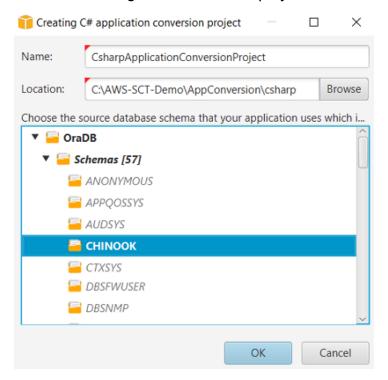
Você pode adicionar vários projetos de conversão de aplicativos em um único AWS SCT projeto. Use o procedimento a seguir para criar um projeto de conversão de aplicativos C#.

Para criar um projeto de conversão de aplicativos C#

 Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para ter mais informações, consulte <u>Iniciando e gerenciando projetos em AWS SCT</u> e Adicionando servidores ao projeto em 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 ter mais informações, consulte Mapeando tipos de dados no AWS Schema Conversion Tool e Mapeamento para alvos virtuais no AWS Schema Conversion Tool.
- No menu Visualizar, escolha 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. 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.

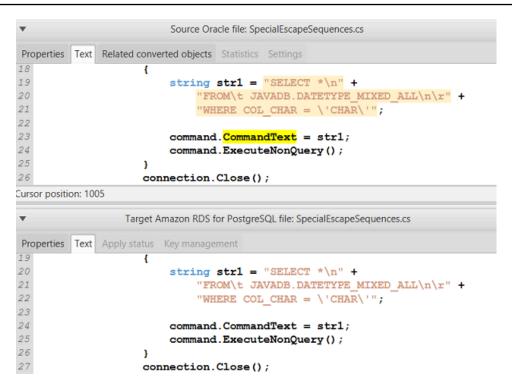
Convertendo o código SQL do seu aplicativo C# em AWS SCT

Depois de adicionar seu aplicativo C# ao AWS SCT projeto, 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. Escolha Converter. 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, AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido.



4. Salve o código do seu aplicativo convertido. Para obter mais informações, consulte <u>Como salvar</u> 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, AWS SCT ainda analisa o código SQL de todos os arquivos de origem de seus aplicativos C#. Em seguida, 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.

Salvando o código do aplicativo convertido com AWS SCT

Use o procedimento a seguir para salvar o código do seu aplicativo convertido.

Para salvar o código do seu aplicativo convertido

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.

Gerenciando projetos de conversão de aplicativos C# em AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos em C#, atualizar o código do aplicativo no AWS SCT projeto ou remover um projeto de conversão em C# do seu AWS SCT projeto.

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).
- 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 AWS SCT projeto.

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.

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 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).
- Selecione Excluir e OK.

Criação de um relatório de avaliação de conversão de aplicativos C# no AWS SCT

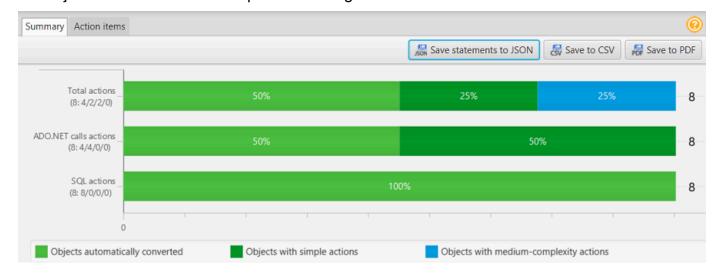
O Relatório de avaliação da conversão do aplicativo C# fornece informações sobre como converter o código SQL incorporado em seu aplicativo C# para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui itens de ação para código SQL que não AWS SCT podem ser convertidos.

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#

- 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).
- 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 arguivos de código fonte.



- 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.

Convertendo código SQL em aplicativos C++ com AWS Schema Conversion Tool

Para uma conversão de Oracle para PostgreSQL, você pode AWS SCT usar para converter código SQL incorporado em seus 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.

Criação de projetos de conversão de aplicativos C++ em 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 Mapeando tipos de dados no AWS Schema Conversion Tool.

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++

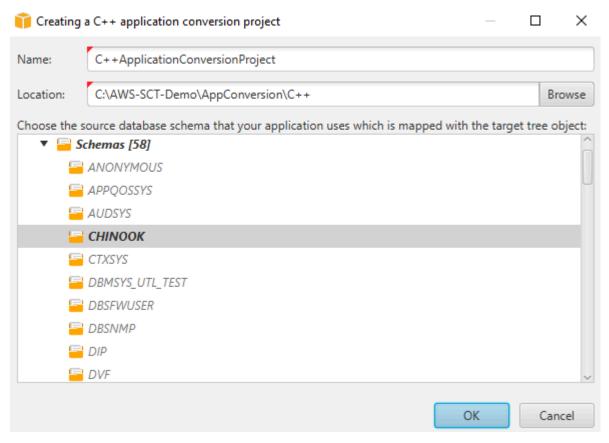
 Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para ter mais informações, consulte <u>Iniciando e gerenciando projetos em AWS SCT</u> e Adicionando servidores ao projeto em AWS SCT.

Código SQL em C++ Versão 1.0.672 482

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 ter mais informações, consulte Mapeando tipos de dados no AWS Schema Conversion Tool e Mapeamento para alvos virtuais no AWS Schema Conversion Tool.

- 3. No menu Visualizar, escolha 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. 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.

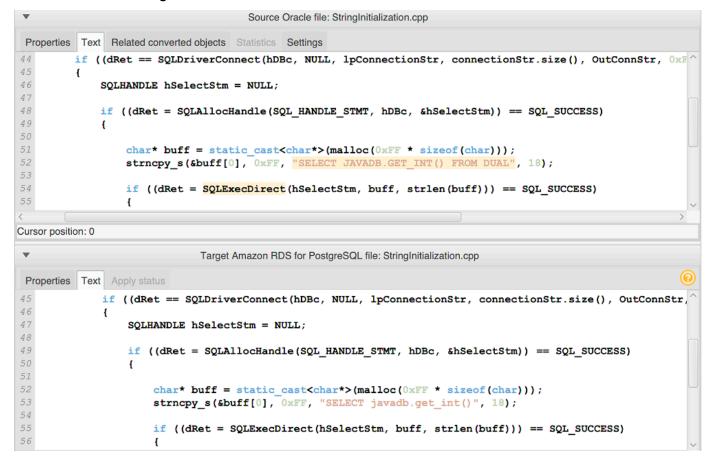
Como converter o código SQL em seu aplicativo C++ na AWS SCT

Depois de adicionar seu aplicativo C++ ao AWS SCT projeto, 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 AWS SCT projeto. Para obter mais informações, consulte Como gerenciar projetos de conversão de aplicativos C++.
- Em Tipo de compilador, escolha o compilador que você usa para o código-fonte do seu aplicativo C++. AWS SCT suporta os seguintes compiladores C++: Microsoft Visual C++, GCC, GNU Compiler Collection e 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++.
- No painel esquerdo, selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
- 7. Escolha Converter. 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, 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.



 Salve o código do seu aplicativo convertido. Para obter mais informações, consulte <u>Como salvar</u> o código do aplicativo convertido.

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.
- Insira o caminho para a pasta para salvar o código do aplicativo convertido e escolha Selecionar pasta.

Gerenciando projetos de conversão de aplicativos C++ no AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos em C++, editar configurações de conversão, atualizar o código do aplicativo em C++ ou remover um projeto de conversão em C++ do seu projeto. 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 AWS SCT projeto.

Para editar as configurações de conversão para todos os aplicativos C++

- No menu Configurações, selecione Configurações do projeto e, em seguida, selecione Conversão de aplicativos.
- Em Tipo de compilador, escolha o compilador que você usa para o código-fonte do seu aplicativo C++. AWS SCT suporta os seguintes compiladores C++: Microsoft Visual C++, GCC, GNU Compiler Collection e 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 AWS SCT projeto.

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.

AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações de código feitas AWS SCT e os resultados da conversão, crie um novo projeto de conversão em C++.

Além disso, AWS SCT remove as configurações de conversão do aplicativo que você especificou para o aplicativo selecionado. Depois de carregar o código atualizado do aplicativo, 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.

Criando um relatório de avaliação de conversão de aplicativos em C++ no 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 itens de ação para código SQL que não AWS SCT podem ser convertidos.

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.

- 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.

Convertendo código SQL em aplicativos Java com AWS Schema Conversion Tool

Para uma conversão de Oracle para PostgreSQL, você pode AWS Schema Conversion Tool usar para converter código SQL incorporado em seus 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 MyBatis estrutura para interagir com bancos de dados, você pode usála AWS SCT para converter instruções SQL incorporadas em arquivos e anotações MyBatis XML.

Código SQL em Java Versão 1.0.672 488

Para entender a lógica dessas instruções SQL, AWS SCT use o arquivo MyBatis de configuração. AWS SCT pode descobrir automaticamente esse arquivo na pasta do aplicativo ou você pode inserir o caminho para esse arquivo manualmente.

Criação de projetos de conversão de aplicativos Java no AWS SCT

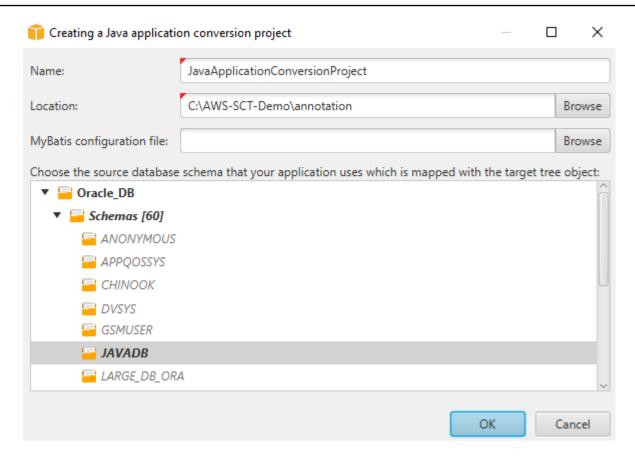
Você pode criar um projeto de conversão de aplicativos Java somente para converter esquemas de banco de dados Oracle em esquemas de banco de dados PostgreSQL. Certifique-se de adicionar uma regra de mapeamento em seu projeto que inclua um esquema Oracle de origem e um banco de dados PostgreSQL de destino. Para obter mais informações, consulte Mapeando tipos de dados no AWS Schema Conversion Tool.

Você pode adicionar vários projetos de conversão de aplicativos em um único AWS SCT projeto. Use o procedimento a seguir para criar um projeto de conversão de aplicativos Java.

Para criar um projeto de conversão de aplicativos Java

- Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para ter mais informações, consulte <u>Iniciando e gerenciando projetos em AWS SCT</u> e Adicionando servidores ao projeto em 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 ter mais informações, consulte Mapeando tipos de dados no AWS Schema Conversion Tool.
- 3. No menu Visualizar, escolha 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.
- Em Local, digite o local do código fonte referente ao aplicativo.
- 7. (Opcional) Para arquivo MyBatis de configuração, insira o caminho para o arquivo MyBatis de configuração. 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. 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.

Convertendo o código SQL do seu aplicativo Java em AWS SCT

Depois de adicionar seu aplicativo Java ao AWS SCT projeto, 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. Escolha Converter. 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, 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.iava
Properties Text Related converted objects Statistics Settings
       private final String PASSWORD = "min_privs";
14
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 SupplierImpl1();
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
 Properties Text Apply status Key management
       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 SupplierImpl1();
24
25
            CallableStatement os = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26
            cs.execute();
27
       }
28 }
```

 Salve o código do seu aplicativo convertido. Para obter mais informações, consulte <u>Como salvar</u> 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, AWS SCT ainda analisa o código SQL de todos os arquivos de origem de seus aplicativos Java. Em seguida, 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.

Salvando o código do aplicativo convertido com 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.
- 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 MyBatis estrutura, certifique-se de atualizar seu arquivo de configuração para funcionar com seu novo banco de dados.

Gerenciando projetos de conversão de aplicativos Java no AWS SCT

Você pode adicionar vários projetos de conversão de aplicativos Java, atualizar o código do aplicativo no AWS SCT projeto ou remover um projeto de conversão Java do seu AWS SCT projeto.

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).
- 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).
- Selecione Atualizar e, em seguida, selecione Sim.

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 AWS SCT e os resultados da conversão, crie um novo projeto de conversão Java.

Se seu aplicativo Java de origem usa a MyBatis estrutura, AWS SCT usa o arquivo de MyBatis configuração para analisar seu código SQL. Depois de alterar esse arquivo, carregue-o no AWS SCT projeto.

Para editar o caminho para o arquivo MyBatis de configuração

- 1. Expanda o nó Java em Aplicativos no painel esquerdo.
- 2. Escolha seu aplicativo e, em seguida, selecione Configurações.
- 3. Escolha Procurar e, em seguida, escolha o arquivo MyBatis de configuração.
- 4. Escolha 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

- 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.

Criando um relatório de avaliação de conversão de aplicativos Java no 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 itens de ação para código SQL que não AWS SCT podem ser convertidos.

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.

Convertendo código SQL em aplicativos Pro*C com AWS Schema Conversion Tool

Para uma conversão de Oracle para PostgreSQL, você pode usar AWS Schema Conversion Tool o AWS SCT() para converter o código SQL incorporado em seus 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.

Criação de projetos de conversão de aplicativos Pro*C em 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 Mapeando tipos de dados no AWS Schema Conversion Tool.

Você pode adicionar vários projetos de conversão de aplicativos em um único AWS SCT projeto. 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

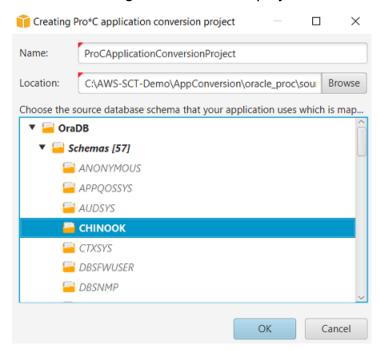
- Crie um projeto de conversão de banco de dados e adicione um banco de dados Oracle de origem. Para ter mais informações, consulte <u>Iniciando e gerenciando projetos em AWS SCT</u> e Adicionando servidores ao projeto em AWS SCT.
- Adicione uma regra de mapeamento que inclua seu banco de dados Oracle 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

Código SQL no Pro*C Versão 1.0.672 496

regra de mapeamento. Para ter mais informações, consulte <u>Mapeando tipos de dados no AWS</u> Schema Conversion Tool e Mapeamento para alvos virtuais no AWS Schema Conversion Tool.

- 3. No menu Visualizar, escolha 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. 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.

Convertendo o código SQL do seu aplicativo Pro*C em AWS SCT

Depois de adicionar seu aplicativo Pro*C ao AWS SCT projeto, 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.
 - 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. Escolha Transformar blocos PL/SQL anônimos em chamadas SQL autônomas ou funções armazenadas para criar procedimentos armazenados em seu banco de dados de destino para todos os blocos PL/SQL anônimos. AWS SCT em seguida, 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.
- No painel esquerdo, selecione o aplicativo a ser convertido e abra o menu de contexto (clique com o botão direito).
- 4. Escolha Converter. 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, AWS SCT cria a estrutura de pastas semelhante ao seu projeto de aplicativo de origem. Aqui você pode revisar o código do aplicativo convertido.



5. Salve o código do seu aplicativo convertido. Para obter mais informações, consulte Como editar e salvar o código do aplicativo convertido.

Editando e salvando o código do aplicativo convertido com AWS SCT

Você pode editar as instruções SQL convertidas e usá-las 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

- Expanda o nó Pro*C em Aplicativos no painel esquerdo.
- 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.
- 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 Visualizar, escolha Visualização principal.

- 8. Expanda o nó Pro*C em Aplicativos no painel do banco de dados de destino.
- 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.

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.
- Insira o caminho para a pasta para salvar o código do aplicativo convertido e escolha Selecionar pasta.

Gerenciando projetos de conversão de aplicativos Pro*C em AWS SCT

Você pode adicionar vários projetos de conversão do aplicativo Pro*C, atualizar o código do aplicativo no AWS SCT projeto ou remover um projeto de conversão do Pro*C do seu projeto. AWS SCT

Para adicionar um projeto adicional de conversão de aplicativos Pro*C

- Expanda o nó Aplicativos no painel esquerdo.
- 2. Escolha o nó Pro*C e abra o menu de contexto (clique com o botão direito).
- 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 AWS SCT projeto.

Para carregar o código do aplicativo atualizado

- 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.

AWS SCT carrega o código do aplicativo dos arquivos de origem e remove os resultados da conversão. Para manter as alterações de código que você fez AWS SCT e os resultados da conversão, crie um novo projeto de conversão 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).
- Selecione Excluir e OK.

Criando um relatório de avaliação de conversão de aplicativos Pro*C no AWS SCT

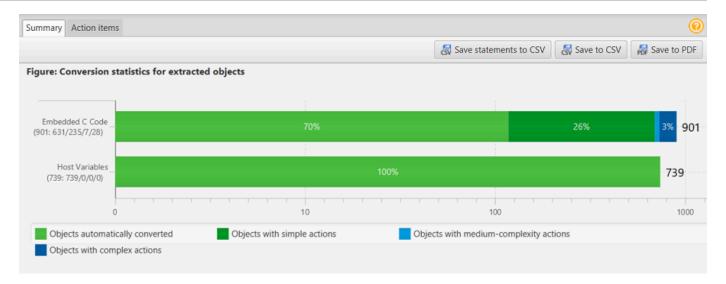
O Relatório de avaliação da conversão do aplicativo Pro*C fornece informações sobre como converter o código SQL incorporado em seu aplicativo Pro*C para um formato compatível com seu banco de dados de destino. O relatório de avaliação fornece detalhes de conversão para todos os pontos de execução do SQL e todos os arquivos de código fonte. O relatório de avaliação também inclui itens de ação para código SQL que não AWS SCT podem ser convertidos.

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

- 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.



- 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 pacotes de extensão com AWS Schema Conversion Tool

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 builtin-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 Database Mail 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 seu perfil do IAM.
- iam: PassRole: para passar o perfil do IAM especificado para o mecanismo de regras.
- iam: TagRole: para adicionar tags a um perfil 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 perfil 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 obter 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 AWS SCT possível converter recursos do banco de dados de origem em recursos equivalentes no banco de dados de destino. O pacote de AWS SCT extensão relevante

contém bibliotecas personalizadas que emulam algumas funcionalidades do banco de dados de origem em seu banco de dados de destino.

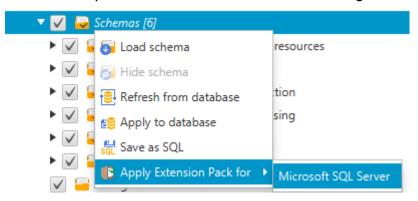
Se você estiver convertendo um banco de dados transacional, consulte <u>Usando as AWS Lambda</u> <u>funções do pacote AWS SCT de extensão</u> .

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

 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 obter mais informações, consulte <u>Gerenciando perfis no AWS</u> Schema Conversion Tool.

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.

 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

 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 obter mais informações, consulte Gerenciando perfis no AWS Schema Conversion Tool.

Certifique-se de que sua política do bucket do Amazon S3 inclua as seguintes permissões:

No exemplo anterior, 111122223333:user/DataExtractionAgentName substitua pelo nome do seu usuário do IAM.

- 2. Converter seus esquemas de data warehouse de origem. Para obter mais informações, consulte Conversão de esquemas de data warehouse .
- 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 trabalhos e outros recursos para bancos de dados hospedados na 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 hospeda e gerencia automaticamente um banco de dados na Amazon EC2, pode 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.

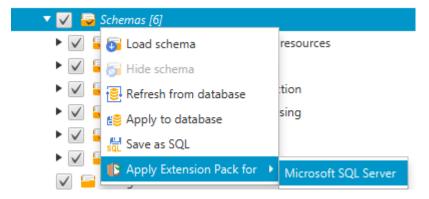


♠ Important

Os recursos AWS de emulação de serviços são compatíveis somente com bancos de dados instalados e autogerenciados na 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 obter mais informações, consulte Gerenciando perfis no AWS Schema Conversion Tool.
- 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.



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 obter mais informações, consulte Instalando e configurando AWS Schema Conversion Tool.

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 pacote de serviço 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 := '';
```

Para definir CONVERSION_LANG em inglês, altere a linha para o seguinte:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Para definir CONVERSION_LANG em 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

Práticas recomendadas de uso AWS Schema Conversion Tool

Encontre informações sobre as melhores práticas e opções para usar o 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 que AWS SCT consome

- 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.
- 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, escolha OK e reinicie AWS SCT para aplicar as alterações.

Como configurar a pasta padrão do projeto

AWS SCT usa a pasta do projeto para armazenar os arquivos do projeto, salvar relatórios de avaliação e armazenar o código convertido. Por padrão, 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

 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 produzidas AWS SCT ao converter seus bancos de dados, scripts e SQL do aplicativo. Embora o aumento das informações de registro possa retardar a conversão, as alterações podem ajudar você a fornecer informações robustas ao AWS Support em caso de erros.

AWS SCT armazena registros em seu ambiente local. Você pode visualizar esses arquivos de log e compartilhá-los com o AWS Support ou com AWS SCT os desenvolvedores para solucionar problemas.

Para alterar as configurações de registro em log

 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 registros da interface de linha de AWS SCT comando (CLI).
- 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, 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 excluirá 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.
- 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 AWS SCT ele pergunte onde armazenar os registros sempre que você usar agentes de extração de dados.
- 9. Em Modo de depuração, selecione Verdadeiro. Use essa opção para registrar informações adicionais quando AWS SCT os registros padrão 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 aplicações obsoletas APIs, mau uso da API e outras situações de tempo de execução que sejam 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, AWS SCT define o nível de registro 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 AWS Support, acesse o diretório em que os registros 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.



(i) 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 AWS Support e evita que o sistema de arquivos de log seja preenchido.

Solução de problemas com AWS Schema Conversion Tool

A seguir, você encontrará informações sobre como solucionar problemas com o 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 você usou para se conectar ao banco de dados Oracle não tem permissões suficientes para ler o esquema, conforme exigido pelo 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, é AWS SCT necessário acesso aos objetos em seu banco de dados de origem. Quando 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 AWS SCT necessários para seu banco de dados, consulte <u>Como se conectar ao banco de dados de origem com o AWS Schema</u> Conversion Tool a seção apropriada do banco de dados de origem neste guia.

- Um objeto citado no esquema n\u00e3o existe mais no banco de dados. Para ajudar a resolver o
 problema, voc\u00e0 pode se conectar com as permiss\u00f3es do SYSDBA e verificar se o objeto est\u00e1
 presente no banco de dados.
- O SCT está tentando avaliar um objeto criptografado.

Mensagem de aviso Versão 1.0.672 519

Referência de CLI para AWS Schema Conversion Tool

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 obter mais informações, consulte Instalando AWS Schema Conversion Tool.

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 obter mais informações, consulte Modo de script.

Pré-requisitos Versão 1.0.672 520

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 prérequisitos.

```
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.



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, *command_name* substitua 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'
```

Modo interativo Versão 1.0.672 521

No exemplo anterior, *command_name* substitua pelo nome de um comando. Em seguida, *parameters_list* substitua por uma lista de nomes de parâmetros separados por uma 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, *file_path* substitua 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.

Exemplos Versão 1.0.672 522

Para entrar no modo interativo da CLI do GetCliScenario, conclua os pré-requisitos e entre no modo interativo da CLI da AWS SCT . Para obter mais informações, consulte Modo interativo.

Em seguida, use a sintaxe a seguir para executar o comando GetCliScenario e obter os cenários do AWS SCT.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

No exemplo anterior, <code>template_type</code> substitua por um dos tipos de modelo da tabela a seguir. Em seguida, <code>file_path</code> substitua pelo caminho a 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
BTEQScriptConversão	BTEQScriptConversi onTemplate.scts	Converte Teradata Basic Teradata Query (BTEQ),,, FastExport e FastLoad scripts em RSQL do Amazon Redshift. MultiLoad Para obter mais informações, consulte Conversão de dados usando ETL.
ConversionApply	ConversionTemplate .scts	Converte esquemas do banco de dados de origem e aplica o código convertido ao banco de dados de destino. Opcionalm ente, salva o código convertid o como um script SQL e

Obter cenários de CLI Versão 1.0.672 523

Tipo de modelo	Nome do arquivo	Descrição
		salva o relatório de avaliação . Para obter mais informaçõ es, consulte <u>Conversão de</u> esquemas.
GenericAppConversion	GenericApplication ConversionTemplate .scts	Converte o código SQL incorporado em seus aplicativ os com o conversor genérico de AWS SCT aplicativos. Para obter mais informações, consulte Código SQL.
HadoopMigration	HadoopMigrationTem plate.scts	Migra seu cluster Hadoop on-premises para o Amazon EMR. Para obter mais informações, consulte Conectando-se aos bancos de dados do Apache Hadoop com o AWS Schema Conversion Tool.
HadoopResumeMigration	HadoopResumeMigrat ionTemplate.scts	Retoma uma migração interrompida do seu cluster Hadoop on-premises para o Amazon EMR. Para obter mais informações, consulte Conectando-se aos bancos de dados do Apache Hadoop com o AWS Schema Conversion Tool.

Obter cenários de CLI Versão 1.0.672 524

Tipo de modelo	Nome do arquivo	Descrição
Informatica	<pre>InformaticaConvers ionTemplate.scts</pre>	Converte o código SQL incorporado em seus scripts de extração, transform ação 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 obter mais informações, consulte Scripts de ETL da Informatica.
LanguageSpecificAppConversi on	LanguageSpecificAp pConversionTemplat e.scts	Converte o código SQL incorporado em seus aplicativ os C#, C++, Java e Pro*C com o conversor genérico de aplicativos do AWS SCT . Para obter mais informações, consulte Como converter o aplicativo SQL.
OozieConversion	OozieConversionTem plate.scts	Converte seus fluxos de trabalho do Apache Oozie em. AWS Step Functions Para obter mais informações, consulte Conectando-se aos fluxos de trabalho do Apache Oozie com o AWS Schema Conversion Tool.

Obter cenários de CLI Versão 1.0.672 525

Tipo de modelo	Nome do arquivo	Descrição
RedshiftAgent	DWHDataMigrationTe mplate.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 obter mais informações, consulte Migrar de um data warehouse.
ReportCreation	ReportCreationTemp late.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 obter mais informaçõ es, consulte Relatório de avaliação da.
SQLScriptConversão	SQLScriptConversio nTemplate.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 obter mais informações, consulte AWS SCT Modo de script CLI.

Exemplos

O exemplo a seguir baixa todos os modelos para a pasta C:\SCT\Templates.

Exemplos Versão 1.0.672 526

```
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 obter mais informações, consulte <u>Instalando drivers JDBC para AWS</u> Schema Conversion Tool.

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 obter mais informações, consulte <u>Mapeamento</u> de tipo de dados.

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 obter mais informações, consulte Exemplos.
- Remova CreateFilter, converta as operações -filter, ApplyToTarget -filter, SaveTargetSQL, SaveTargetSQLbyStatement e SaveReportCSV do seu script.

Editar cenários de CLI Versão 1.0.672 527

3. Para oracle_driver_file na SetGlobalSettingsoperação, insira o caminho para seu driver Oracle. Em seguida, para postgresql_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 SetGlobalSettingsoperação, 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.
- 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 AddServerMappingoperação 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 obter 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

Modo de script Versão 1.0.672 528

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, *file_path* substitua 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, *file_path* substitua 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 emMaterial 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.

Exemplos Versão 1.0.672 529

Notas de lançamento para AWS Schema Conversion Tool

Esta seção contém notas de lançamento para AWS SCT, começando com a versão 1.0.640.

Notas de lançamento do AWS SCT Build 677

Notas de lançamento para AWS Schema Conversion Tool.

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Todos	Todos	Atualização do Java da versão 11 para a versão 17.	Sim	Sim
Todos	Todos	Todas as bibliotecas e dependências desatualizadas são atualizadas.	Sim	Sim
BigQuery	Redshift	Cotação correta para identific adores de objetos na extração de dados.	Sim	Não
Db2 z/OS	Todos	Os nomes dos esquemas com espaços à direita agora	Sim	Não

Notas de lançamento — 677 Versão 1.0.672 530

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
		são tratados corretamente.		
SAP ASE	Todos	Impediu o estouro aritmétic o nos cálculos.	Sim	Não
Tudo	Todos	. jaros tamanhos dos arquivos agora estão otimizado s.	Sim	Não
Db2 z/OS	PostgreSQ L/Aurora PostgreSQL	Foi adicionad o suporte para subconsultas na cláusula WHEN do gatilho.	Sim	Não
MSSQL	Todos	Escopo estendido para resolução de identificadores.	Sim	Sim

Notas de lançamento — 677 Versão 1.0.672 531

Notas de lançamento do AWS SCT Build 676

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Oracle	Postgre L/ Aurora Postgre L	Nova emulação de função integrada para as seguintes funções: • SYS.UTL_RAW.BIT_AND(RAW,RAW) • XDB.DBMS_XSLPROCESSOR.CLOB2 FILE(CLOB) • XDB.DBMS_XSLPROCESSOR.READ2 CLOB(VARCHAR2) • SYS.UTL_RAW.BIT_OR(RAW,RAW) • SYS.UTL_RAW.BIT_COMPLEMENT(RAW)	Não	Sim
MS SQL Server		Mensagem Database Mail not supported removida do relatório em PDF	Sim	Sim
Oracle	Postgre L/ Aurora Postgre L	Implementou a conversão de restrições para tabelas particionadas.	Sim	Sim
Oracle	MySQL	Revisão da aplicabilidade do Al-602 na conversão de tabelas	Sim	Sim

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
MS SQL Server	Postgre L/ Aurora Postgre L	agora é compatível com a declaração MERGE no PostgreSQL 15.x	Sim	Sim
Todos	Todos	Conexões JDBC implementadas: propriedades avançadas	Sim	Não
Tudo	Todos	CLI: falha de comando PrintOLAP TaskStatus corrigida	Sim	Não
Teradat		Implementou a conversão de tipo de dados no estilo Teradata.	Sim	Não
Teradat		Conversão MERGE incorreta corrigida em SQL/BTEQ.	Sim	Não
Teradat		Implementou a conversão de tipo de dados no estilo Teradata.	Sim	Não
Teradat		Conversões de funções LEAD/LAG implementadas.	Sim	Não
Teradat		Erro AI-9996 Transformer error occurred in statement corrigido.	Sim	Não
Teradat		Erro AI-9996 Transformer error in selectItem corrigido.	Sim	Não
Teradat		Conversão implementada para procedime nto de armazenamento parcial: XbiDQM.SpCmprsnDly	Sim	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Teradat	Amazor Redshif	Declaração UNPIVOT implementada com alias.	Sim	Não
Teradat		Declaração Delete implementada com várias tabelas de origem.	Sim	Não
Teradat		Corrigir para AI-9996 Transformer error occurred in functionC allExpression .	Sim	Não
Teradat		Conversão de cláusula NORMALIZE implementada.	Sim	Não
Teradat		Foi corrigida a conversão incorreta em declarações DELETE com subconsultas.	Sim	Não
Teradat		Erro AI-9996 Transformer error occurred in tableOperatorSourc e corrigido.	Sim	Não
Teradat		Erro AI-9996 Transformer error occurred in additiveExpression corrigido.	Sim	Não
Teradat		Implementou a conversão de objetos do sistema DBC.	Sim	Não
Teradat		Solução alternativa implementada de atualização com predicados de junção implícitos.	Sim	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Netezza		Resolvido o erro de conversão de declarações CREATE MATERIALIZED VIEW.	Sim	Não
Db2luw	Postgre L/ Aurora Postgre L	Conexão de opções estendidas do JDBC: opções adicionais de conexão adicionad as.	Sim	Não
Db2luw	Postgre L/ Aurora Postgre L	Foi adicionado suporte para declaração MERGE no PostgreSQL 15.x	Sim	Não
Db2luw	Postgre L/ Aurora Postgre L	Conversão GLOBAL TEMPORARY TABLE implementada.	Sim	Não
Db2luw	Postgre L/ Aurora Postgre L	Conversão USER DEFINED TYPES implementada.	Sim	Não
Db2luw	MySQL	Conversão GLOBAL TEMPORARY TABLE implementada.	Sim	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Db2luw	MySQL	Conversão USER DEFINED TYPES implementada.	Sim	Não
Db2luw	MySQL	Conversão USER DEFINED FUNCTIONS implementada.	Sim	Não
Db2luw	MariaDI	Conversão GLOBAL TEMPORARY TABLE implementada.	Sim	Não
Db2luw	MariaDI	Conversão USER DEFINED TYPES implementada.	Sim	Não
Sybase	Todos	Foi adicionado suporte para autenticação do Kerberos.	Sim	Não
Db2luw	Postgre L/ Aurora Postgre L	Foi adicionado suporte para a conversão de várias versões para os alvos	Sim	Não
Azure SQL/ Micro soft SQL Server	Postgre L/ Aurora Postgre L	Foi adicionado suporte para a conversão de várias versões para os alvos	Sim	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibi lidade na AWS DMS Schema Conversion Tool (SCT)	Disponibilidade na AWS DMS conversão de esquema
Db2luw	Postgre L/ Aurora Postgre L	Foi adicionado suporte para declaração MERGE no PostgreSQL 15.x.	Sim	Não
Teradat	_	Conversão de alteração de função não suportada corrigida.	Sim	Não
Tudo		Extratores de dados: particionamento implementado por uma coluna indexada.	Sim	Não

Notas de lançamento do AWS SCT Build 675

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Cassandra	DynamoD	Corrigido um bug em que a instalação do Cassandra falhava no data center de destino.	Não
DB2 LUW	PostgreS(L	SQL DINÂMICO: instrução PREPARE: resolução e conversão sem SQL dinâmico.	Não
DB2 LUW	PostgreS(L	Foi adicionado suporte para SPECIAL REGISTER.	Não
DB2 LUW	PostgreS(L	Atualização do pacote de extensão	Não

Origem	Alvo	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 visões materializadas.	Não
Oracle	Amazon Redshift	Foi adicionado suporte para consultas recursivas no Amazon Redshift.	Sim
Oracle	PostgreS(L, Aurora PostgreS(L	Correção para conversão incorreta do tipo de dados NUMBER.	Sim
Oracle	Amazon Redshift	Migração de dados. Particionamento automático 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	Alvo	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	PostgreS(L, Aurora PostgreS(L	Conversão de várias versões	Não
Oracle	PostgreS(L	Foi adicionado suporte para declaração MERGE no PostgreSQL 15.x.	Sim
Oracle	PostgreS(L	Foi adicionado suporte para novas funções de expressão regular no PostgreSQL 15.x.	Sim
Oracle	PostgreSC L, Aurora PostgreSC L	A instrução ON CONFLICT DO UPDATE é convertid a sem alias excluído.	Sim
Teradata	Amazon Redshift	Foi adicionado suporte de conversão para LEAD/ LAG funções.	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. time/timestamp	Não
Teradata	Amazon Redshift	AI-9996 durante procedimentos de conversão com instruções MERGE.	Não

Notas de lançamento do AWS SCT Build 674

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Todos	Todos	Várias correções de erros e melhorias na performan ce	Parcial (somente para pares de origem e destino compatíveis)
Azure SQL/ Micro soft 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/ Micro soft 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/ Micro soft SQL Server	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
Azure SQL/ Micro soft SQL Server	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Parcial (a conversão de esquema atualmente não oferece suporte ao Azure SQL como origem)
Azure SQL/ Micro soft SQL Server	Amazon RDS para PostgreS0 L/ Amazon Aurora PostgreS0 L		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: • EXCEPTION BLOCK • CONFIRMAÇÃO AUTOMÁTICA • NONATOMIC • CONJUNTO DE AGRUPAMENTO • CUBE • ROLLUP	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 LUW	Amazon RDS para PostgreS0 L/ Amazon Aurora PostgreS0 L		Não
DB2 LUW	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Não
DB2 z/ OS	Amazon RDS para PostgreS0 L/ Amazon Aurora PostgreS0 L		Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 z/ OS	Amazon RDS para PostgreS(L/ Amazon Aurora PostgreS(L		Não
DB2 z/ OS	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Não
DB2 z/ OS	Amazon RDS para PostgreS(L/ Amazon Aurora PostgreS(L		Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
DB2 z/ OS	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Não
DB2 z/ OS	Amazon RDS para PostgreS(L/ Amazon Aurora PostgreS(L		Não
DB2 z/ OS	Amazon RDS para PostgreS0 L/ Amazon Aurora PostgreS0 L		Não

Origem	Alvo	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 PostgreSC L/ Amazon Aurora PostgreSC L		Sim
Oracle	Amazon RDS para PostgreS(L/ Amazon Aurora PostgreS(L		Sim
Oracle		Suporte à autenticação Kerberos	Não

Origem	Alvo	Novidades, aprimoramentos ou correções	Disponibilidade na AWS DMS conversão de esquema
SAP ASE	Amazon RDS para PostgreSC L/ Amazon Aurora PostgreSC L		Não
SAP ASE	Amazon RDS para PostgreS0 L/ Amazon Aurora PostgreS0 L		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	Alvo	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 procedime nto executado pelo BTEQ	Não

Notas de lançamento do AWS SCT Build 673

Origem	Alvo	Novidades, aprimoramentos ou correções
Todos	Todos	Melhorias no desempenho geral e correções de erros
Azure SQL/Micro soft SQL Server	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Conversão de chamada de função incorreta corrigida
Azure SQL/Micro soft SQL Server	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Conversão implementada da cláusula F0R XML
Azure SQL/Micro soft SQL Server	Aurora PostgreSQ L/Amazon RDS	Conversão da cláusula F0R XML com alias errado.

Origem	Alvo	Novidades, aprimoramentos ou correções
	PostgreSQ L	
Azure SQL/Micro soft SQL Server	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o erro ao AWS SCT não converter EXECUTE declaraçõ es que executam uma cadeia de caracteres com parâmetros de procedimento.
Azure SQL/Micro soft SQL Server	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Conversão aprimorada de instruções UPDATE com junções internas.
Azure Synapse	Amazon Redshift	Foi corrigida a conversão incorreta da função OBJECT_ID incorpora da.
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	 Implementada a conversão das seguintes instruções e objetos: 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	Alvo	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o erro quando o índice baseado em funções ignora as funções após a conversão.
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o erro em que a instrução REPEAT fechava com AI 9996 após a conversão
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o erro em que a FINAL TABLE cláusula fechava com 9996.
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ 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 PostgreSQ L/Amazon RDS PostgreSQ L	Suporte à função PostgreSQL.VARCHAR_FORMAT

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Implementada a alteração de agrupamento nos comandos SCT CLI CreateTransformationRule e ModifyTransformationRule .
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 rsa-sha2.
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 PRIOR não estava na lista SELECT.
Oracle	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Implementado retorno a um elemento de uma matriz associativa
Oracle	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o inesperado Al 9996 em UNPIVOT com colchetes

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L/Amazon RDS PostgreSQ L	Corrigido o inesperado Al 9996 em UNPIVOT com UNION ALL
Oracle	Aurora PostgreSQ L/Amazon RDS PostgreSQ 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 S0ME para o Amazon Redshift.
Teradata	Amazon Redshift	Foi adicionado suporte nativo para predicado QUALIFY.
Teradata	Amazon Redshift	Conversão aprimorada do seguinte: Consultas recursivas GROUPING SET CUBE ROLLUP Instrução UPDATE com junção implícita

Origem	Alvo	Novidades, aprimoramentos ou correções
Fontes OLAP	Extrator de dados do Amazon Redshift	Comandos CLI implementados Stop/Resume para tarefas do Amazon Redshift Data Extractor.
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	Alvo	Novidades, aprimoramentos ou correções
Todos	Amazon RDS para PostgreSQ L	Implementou o suporte à versão principal 15 do PostgreSQL como destino de migração.
Todos	Amazon Redshift	Foi adicionado um novo PrintTaskStatus 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 PostgreSQ L	Implementada a conversão de funções COALESCE, DATEADD, GETDATE e SUM.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQ L	
Banco de dados do SQL Azure	Aurora PostgreSQ L	Melhorada a conversão de instruções UPDATE com cláusulas JOIN e OUTPUT.
Microsoft SQL Server	PostgreSQ L	
Banco de dados do SQL Azure Microsoft SQL	Aurora PostgreSQ L PostgreSQ L	Resolvido um erro que ocorria durante a conversão da instrução SELECT TOP 1 WITH TIES.
Server	L	
Banco de dados do SQL Azure	Aurora PostgreSQ L	Foram resolvidos vários problemas que ocorreram durante a conversão das cláusulas FOR XML em funções integradas.
Microsoft SQL Server	PostgreSQ L	
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	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Corrigido um erro em que a função T0_CHAR com modelos de formato de hora era convertida incorretamente.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementou a conversão de expressões de tabela aninhadas.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão das instruções GOTO, MERGE, REPEAT e SIGNAL.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de instruções FETCH com palavras-chave de orientação BEFORE e AFTER.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão das referências da tabela FINAL TABLE e OLD TABLE.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementou a conversão das seguintes funções. 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 XMLCAST XMLCAST XMLSERIALIZE YEAR
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de um alias de uma subconsulta em cláusulas J0IN.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de funções COALESCE.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de índices EXPLICIT.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ 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.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de chaves primárias e restrições exclusivas.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ 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.
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	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	Foi adicionado suporte ao Microsoft SQL Server versão 2022 como uma origem.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L	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	Alvo	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 PostgreSQ L PostgreSQ L	Implementou a conversão do DBMS_OUTPUT pacote adicionando novas funções ao pacote AWS SCT de extensão.
Oracle	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de funções e procedimentos que usam matrizes associativas como argumentos ou parâmetros.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de cláusulas DISTINCT em instruções SELECT.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de tabelas em que a restrição de chave primária tem o mesmo nome da tabela.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão do procedimento RAISE_APPLICATION_ ERROR com o terceiro parâmetro.
Oracle	Aurora PostgreSQ L PostgreSQ 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	Alvo	Novidades, aprimoramentos ou correções
Oligoni	71100	Trovidades, 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	Os seguintes modelos de interface de linha de AWS SCT comando (CLI) foram atualizados. • BTEQScriptConversão • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatica Para obter mais informações sobre os modelos de AWS SCT CLI, consulte. Obter cenários de CLI
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 Edge 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 obter mais informações, consulte Conversão de fluxos de trabalho do Oozie;.
Banco de dados do SQL Azure	Aurora PostgreSQ L	Corrigido um erro de resolução que ocorria para tabelas e aliases.

Origem Microsoft SQL Server	Alvo PostgreSQ L	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de cláusulas INDEX ON.
Banco de dados do SQL Azure Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Foi aprimorada a conversão dos seguintes objetos para evitar itens de ação inesperados. • Declarações em lote • Listas de expressões • Alias de tabelas • Tabelas temporárias • Acionadores • Variáveis do usuário
Banco de dados do SQL Azure Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um erro de análise que ocorreu para procedimentos.

Origem	Alvo	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQ L	Corrigido um erro em que AWS SCT eram usados nomes incorreto s de tabelas temporárias no código convertido para OBJECT_ID funções.
Microsoft SQL Server	PostgreSQ L	
Banco de dados do SQL Azure	Aurora PostgreSQ L	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão dos seguintes elementos de código.
Microsoft SQL Server	PostgreSQ L	 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 obter mais informações, consulte 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 <u>Instalando drivers JDBC para AWS Schema Conversion Tool</u> .

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Corrigido um erro de resolução que ocorria para colunas em cursores implícitos.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Implementou a conversão de chaves estrangeiras de esquema cruzado.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de funções LENGTH e VARCHAR.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L	Implementada a conversão de instruções LABEL ON e DECLARE CONDITION .
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Implementada a conversão de instruções SELECT com cláusulas OPTIMIZE FOR.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de instruções CREATE TABLE adicionando valores padrão para todos os tipos de dados compatíveis.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de atributos INCREMENT BY.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de tabelas particionadas adicionando a capacidade de excluir partições de tabela do escopo de conversão.
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão das definições de chave primária com colunas INCLUDE.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão da função SUBSTRING .
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções SET e DECLARE HANDLER FOR.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de tipos de dados variáveis.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de funções XMLTABLE.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão de comentários no código-fo nte.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Resolvido um problema em que o item de ação 9997 aparecia inesperadamente durante a conversão de aliases do cursor.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ 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	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	Foram atualizadas duas funções no pacote de extensões para adicionar nomes de esquemas como argumentos no código convertid o.
Oracle	Aurora PostgreSQ L PostgreSQ L	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	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L	Melhorada a conversão de instruções CREATE TYPE para evitar exceções inesperadas.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Melhorada a conversão de tabelas aninhadas.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Resolveu um erro de análise que ocorria nos objetos do pacote.
	PostgreSQ L	
Oracle	Aurora PostgreSQ 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.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Resolveu um problema em que acionadores em nível de linha para tabelas particionadas eram convertidos incorretamente.
	PostgreSQ L	

Origem	Alvo	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 obter 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 <u>MERGE</u> 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	Alvo	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQ L	Resolvidos problemas em que o item de ação 9996 apareciam inesperadamente durante a conversão dos seguintes elementos de código.
Microsoft SQL Server	PostgreSQ L	 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	Alvo	Novidades, aprimoramentos ou correções
		 Instruções IIF Listas de expressões Funções do MONTH() Instruções UPDATE Funções do YEAR()
Azure Synapse Analytics	Amazon Redshift	Foi adicionado suporte ao Azure Synapse Analytics como origem para o processo de avaliação de vários servidores. Para obter mais informações, consulte 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 PostgreSQ L PostgreSQ L	Corrigido um erro de resolução que ocorria para tabelas e colunas.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de expressões CASE.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de referências CURRENT_DATE em registros especiais. Uma referência a um registro especial no Db2 for z/OS é uma referência a um valor fornecido pelo servidor atual.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQ L	Implementada a conversão de funções DATE e POSSTR.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de constantes de data e hora.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de valores padrão para colunas dos seguintes tipos de dados: DATE, TIME, TIMESTAMP e TIMESTAMP WITH TIME ZONE.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções SELECT INTO.
	PostgreSQ L	
Microsoft SQL Server	Aurora PostgreSQ L	Melhorada a conversão de funções DATEDIFF.
	PostgreSQ L	

Outions	Alexa	No. idealar annima
Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um erro em que as funções ISNULL eram convertidas em NULLIF. Como resultado, o código convertido produzia resultado s 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 concluída s 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 PostgreSQ L	Aurora MySQL Aurora PostgreSQ L MySQL PostgreSQ L	Foi adicionada a capacidade de se conectar a bancos de dados usando um protocolo IPv6 de endereço.
Oracle	Amazon RDS para Oracle	Implementada a conversão do pacote DBMS_J0B que agenda e gerencia trabalhos na fila de trabalhos.
Oracle	Aurora PostgreSQ L PostgreSQ L	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 códigofonte Oracle.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ 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 Convertendo o código SQL do seu aplicativo Java em AWS SCT.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de operadores de concatenação () dentro de índices funcionais.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de condições IN em que seu código-fonte não inclui parênteses para uma única expressão.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções MERGE para INSERT ON CONFLICT no PostgreSQL.
Oracle	Aurora PostgreSQ L PostgreSQ L	Resolvido um erro de análise que ocorreu para pacotes de procedimentos.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ 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 Edge 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	Alvo	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 obter mais informações, consulte Relatório de avaliação multisservidor.

Origem	Alvo	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 Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que os nomes das colunas eram convertidos incorretamente com a opção Gerar nomes exclusivos para índices ativada.
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	Alvo	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQ L	Foi adicionada uma nova configuração de conversão para especificar as opções DATE FORMAT e TIME FORMAT durante a conversão das funções CHAR integradas.
	PostgreSQ L	
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	Implementada a conversão de instruções LEAVE em instruções do bloco BEGINEND .

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de funções XMLPARSE, XMLTABLE e XMLNAMESPACES .
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de funções CHAR integradas.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de cursores.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que o item de ação 9996 aparecia inesperadamente durante a conversão de instruções loop F0R.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão do uso de tipos de tabela em instruções SELECT.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Babelfish for Aurora PostgreSQ L	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 PostgreSQ L PostgreSQ L	Implementada a conversão de funções GROUPING_ID .
Oracle	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de código SQL em aplicativos C# adicionan do suporte ao mapeamento personalizado de tipos de dados no modo de interface de linha de comandos (CLI).
Oracle	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de tabelas aninhadas para evitar um item de ação 9996 inesperado.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L	Resolvido um problema em que a chamada de um construtor de objetos era convertida incorretamente.
	PostgreSQ L	
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 obter 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 convertid as.
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 T0_CHAR com argumentos de data e formato eram convertidas incorretamente.

Notas de lançamento do AWS SCT Build 668

Origem	Alvo	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	Alvo	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 parâmetro s.
IBM DB2 para z/OS	Aurora PostgreSQ L	Melhorada a conversão de funções SUBSTR com parâmetros LOCATE.
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 convertid o.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Suporte implementado para instruções de loop F0R.
IBM Db2 para z/OS	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Melhorada a conversão de vários manipuladores de exceções em um único bloco BEGIN END.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Conversão de acionadores INSERT e DELETE aprimorados.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Foi aprimorada a conversão de chamadas de procedimentos aninhados.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de tabelas.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que operações lógicas NOT bit a bit eram convertidas incorretamente em valores inteiros.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que matrizes locais não eram inicializ adas na versão 8.0.2 e inferior do PostgreSQL.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que instruções MERGE com cláusulas WHEN NOT MATCHED BY SOURCE eram convertidas incorretamente.

Origem	Alvo	Novidades, aprimoramentos ou correções
MySQL	Aurora MySQL	Resolveu um problema em que determinava AWS SCT incorreta mente as permissões de usuário concedidas pela rds_super user_role 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 PostgreSQ L PostgreSQ 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 Pacotes de extensão .
Oracle	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ L	Suporte implementado de métodos COUNT em coleções.
Oracle	Aurora PostgreSQ L PostgreSQ L	Suporte implementado de variáveis e construtores em tabelas aninhadas.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L	Suporte implementado de funções RATIO_TO_REPORT e STANDARD_HASH .
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Conversão aprimorada de objetos grandes (LOBs) como parte do pacote de AWS SCT extensão.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Conversão aprimorada de coleções locais.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Melhorada a conversão de instruções JOIN com cláusulas USING em que os nomes das colunas não incluem o nome da tabela.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Implementada a conversão de funções EMPTY_BLOB e EMPTY_CLO B .
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de variáveis de ligação posicional em aplicativos C#.
SAP ASE	Aurora PostgreSQ L PostgreSQ L	Conversão implementada de acionadores de vários eventos.
SAP ASE	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de acionadores recursivos.
SAP ASE	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de acionadores com a variável global @erowcount .
SAP ASE	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que funções agregadas na cláusula SET de instruções UPDATE eram convertidas incorretamente.

Origem	Alvo	Novidades, aprimoramentos ou correções
SAP ASE	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que o item de ação 42702 aparecia inesperadamente durante a conversão de instruções UPDATE.
SAP ASE	Aurora PostgreSQ L PostgreSQ 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 obter mais informações, consulte Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.
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	Alvo	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 incorpora dos em seus objetos da Informatica. Para obter mais informações, consulte Scripts de ETL da Informatica.

Alvo	Novidades, aprimoramentos ou correções
Amazon Redshift	Aumentada a versão mínima do driver compatível com o Amazon Redshift para 2.1.0.9. Para obter mais informações, consulte Instalando drivers JDBC para AWS Schema Conversion Tool.
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.
Amazon Redshift	Melhorada a conversão da função DATEDIFF.
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 obter mais informações, consulte Pacotes de extensão .
Amazon Redshift	Resolvido um problema em que os objetos filtrados não eram convertidos no modo de interface de linha de comandos (CLI).
Amazon Redshift	Corrigido um erro em que AWS SCT não convertia tabelas temporári as declaradas em um procedimento armazenado.
Amazon Redshift	Corrigido um erro em que os atributos de codificação da coluna estavam ausentes no código convertido.
Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de instruções UPDATE para tabelas de autorreferência que têm mais de uma cláusula INNER JOIN.
	Amazon Redshift AppostgreSQ L PostgreSQ L

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Implementado o suporte a inserted e às tabelas deleted temporárias que o SQL Server usa para acionadores DML.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	Resolvido um problema em que colchetes ([]) apareciam inesperadamente nos nomes dos objetos do banco de dados no código convertido.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que funções @@R0WC0UNT eram convertidas incorretamente.
Microsoft SQL Server DW	Amazon Redshift	Suporte implementado a tipos de dados geometry e geography .
Microsoft SQL Server DW	Amazon Redshift	Implementado o suporte à palavra-chave MAX nas instruções de tipo de dados no código convertido.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server DW	Amazon Redshift	Melhorada a conversão de funções do DATEADD.
Oracle	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de código SQL em aplicativos Java adicionan do suporte à MyBatis estrutura. Para obter mais informações, consulte Código SQL em Java.
Oracle	Aurora PostgreSQ L PostgreSQ L	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 PostgreSQ L PostgreSQ L	Melhorada a conversão de código SQL em aplicativos Pro*C adicionando suporte para instruções typedef struct.
Oracle	Aurora PostgreSQ L PostgreSQ L	Suporte implementado a instruções CROSS JOIN e LEFT JOIN.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções MERGE. Resolvido um problema em que os valores a serem inseridos estavam ausentes no código convertido.
Teradata	Amazon Redshift	Alterou as configurações padrão de codificação de compactação de colunas que AWS SCT usa no código convertido para correspon der 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	Alvo	Novidades, aprimoramentos ou correções
Banco de dados do SQL Azure	Aurora PostgreSQ L	Resolvido um erro de análise que ocorria em cláusulas 0N que estão dentro das instruções J0IN.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQ L	
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 convertid o.
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	Alvo	Novidades, aprimoramentos ou correções
IBM DB2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de EXCEPTION seções com a transferência de manipuladores de exceções do Db2 para z/OS o PostgreSQL.
IBM DB2 para z/OS	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções OPEN CURSOR.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de funções IIF usando expressões CASE.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que procedimentos com parâmetros com valores de tabela eram convertidos incorretamente quando a instrução CREATE PROCEDURE não incluía um bloco BEGINE
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que a função SCOPE_IDENTITY era convertida incorretamente.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Amazon RDS para Oracle	Corrigido um erro de carregador que ocorria com a função SELECT_CATALOG_ROLE ao usar o Oracle 10g como origem.
Oracle	Amazon RDS para Oracle	O carregador foi aprimorado para suportar tarefas do Oracle Scheduler.
Oracle	Aurora PostgreSQ L PostgreSQ L	Implementada a conversão de instruções JOIN com cláusulas USING.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhor desempenho do código convertido em que o código-fonte inclui variáveis globais nas cláusulas WHERE.
Oracle	Aurora PostgreSQ L PostgreSQ L	Conversão aprimorada de código SQL em aplicativos Java adicionan do suporte à MyBatis estrutura. Para obter mais informações, consulte Código SQL em Java.
Oracle DW	Amazon Redshift	Conversão implementada dos operadores relacionais PIV0T e UNPIV0T.
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	Alvo	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_T IMESTAMP 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 CONSTRAIN T .
Teradata	Amazon Redshift	Resolvido um problema em que as subconsultas QUALIFY convertid as incluíam um nome de subconsulta incorreto.

Origem	Alvo	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 convertid as 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	Alvo	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 BEGINEND .
Azure Synapse Analytics	Amazon Redshift	Resolveu um erro em que AWS SCT não era possível converter procedimentos armazenados com comentários em bloco.

Origem	Alvo	Novidades, aprimoramentos ou correções
BigQuery	Amazon Redshift	Implementou a conversão de BigQuery data warehouses para o Amazon Redshift. Para obter mais informações, consulte Conectand o-se ao Google BigQuery com AWS Schema Conversion Tool.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de acionadores que manipulam vários eventos e funcionam com tabelas inserted e deleted do sistema no SQL Server.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um erro de resolução que ocorria para tabelas inserted e deleted do sistema no SQL Server.
Microsoft SQL Server	Babelfish for Aurora PostgreSQ L	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 varchar2 era convertido incorretamente.

Origem	Alvo	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 PostgreSQ L	VARCHAR2NVARCHAR2RAW
	MariaDB	
	MySQL	AWS SCT aumentou o tamanho máximo da coluna suportada de 8.000 para 32.767 bytes para esses tipos de dados.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Resolvido um erro de análise que ocorria para pacotes do Oracle Event Processing
	PostgreSQ 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 localizad as 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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon	Melhorada a conversão de cláusulas GROUP BY.
	Redshift	
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-c have 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	Alvo	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	Alvo	Novidades, aprimoramentos ou correções
		criadas. Para obter mais informações, consulte <u>Mapeamento de tipo</u> <u>de dados</u> .
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 Edge .
Banco de dados do SQL Azure IBM Db2 para z/OS IBM Db2 LUW Microsoft SQL Server MySQL Oracle PostgreSQ L SAP ASE	Aurora MySQL Aurora PostgreSQ L MariaDB MySQL PostgreSQ L	Implementado o suporte do SUSE Linux 15.3 como uma plataforma para executar agentes de extração de dados.
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de funções do DATEADD.

Origem	Alvo	Novidades, aprimoramentos ou correções
IBM Db2 para z/OS	Aurora PostgreSQ L	Foi adicionada a capacidade de alterar o agrupamento de colunas em regras de migração.
	PostgreSQ L	
Microsoft SSIS	AWS Glue	Resolvido um erro inesperado que ocorria quando os usuários selecionavam um script de origem.
0010	AWS Glue Studio	Selecionavam um script de origem.
Oracle	Aurora 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.
	MariaDB	
	MySQL	
Oracle	Aurora PostgreSQ L	Implementou a conversão de funções do UTL_MATCH pacote como parte do pacote de AWS SCT extensão.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Implementada conversão da função REGEXP_LIKE com o parâmetro NULL.
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L	Melhorada a conversão da função SYS_EXTRACT_UTC .
	PostgreSQ L	
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de código SQL em aplicativos C++ implementando o suporte de funções Wcscats, Wcscpys e Wcsncats. Para obter mais informações, consulte Convertendo código SQL em aplicativos C++ com AWS Schema Conversion Tool.
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 case sensitive e case insensitive nas regras de migração. Para obter mais informações, consulte <u>Aplicação de regras de migração</u> .
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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Implementada a conversão da função STR0KE 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 MULTISET 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	Alvo	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 informaçõ es, consulte Aplicação de 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 obter mais informações, consulte Criando um relatório de avaliação de vários servidores no AWS Schema Conversion Tool.

Origem	Alvo	Novidades, aprimoramentos ou correções
Azure Synapse Analytics	Amazon Redshift	Melhorada a conversão de instruções 0BJECT_ID .
Microsoft SQL Server	Babelfish for Aurora PostgreSQ L	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 Funcional idade 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 PostgreSQ L PostgreSQ L	Corrigido um erro de carregador que ocorria ao usar o Oracle 10g como origem.
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de cláusulas 0FFSET e FETCH.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L	Resolvido um problema em que procedimentos com parâmetros 0UT com valores padrão eram convertidos incorretamente.
	PostgreSQ L	
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 F0RMAT usando a função T0_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	Alvo	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	Alvo	Novidades, aprimoramentos ou correções
Todos	Todos	Foi adicionada a capacidade de criar AWS SCT projetos automatic amente 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 obter mais informações, consulte Criando um relatório de avaliação de vários servidores no AWS Schema Conversion Tool.
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 PostgreSQ L	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 PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções DELETE com cláusulas NON EXISTS.

Origem	Alvo	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 PostgreSQ L PostgreSQ L	Resolvido um erro em que o código convertido de um acionador incluía duas menções ao alias do objeto.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de objetos com nomes em letras maiúscula s 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 PIV0T e UNPIV0T.
Netezza	Amazon Redshift	Implementada a conversão dos tipos de dados TIME.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora MySQL	Implementada a conversão constante do pacote UTL_TCP.CRLF .
	Aurora PostgreSQ L	
	MySQL	
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	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.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Implementou a conversão de código SQL em aplicativos C++. Para obter mais informações, consulte Convertendo código SQL em aplicativos C++ com AWS Schema Conversion Tool.
	PostgreSQ L	
Oracle	Aurora PostgreSQ L	Suporte implementado de nomenclatura com distinção entre maiúsculas e minúsculas para a conversão de variáveis globais e matrizes associativas.
	PostgreSQ L	

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão das funções T0_CHAR, T0_DATE e T0_NUMBER no pacote de extensão.
Oracle	Aurora PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ 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 PostgreSQ L PostgreSQ 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	Alvo	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	Resolveu um problema em que expressões de tabela comuns não recursivas (CTEs) com uma WITH cláusula eram convertidas em recursivas. CTEs
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	Alvo	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	Alvo	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 obter mais informações, consulte Editando mapeamentos de tipos de dados no AWS Schema Conversion Tool.
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	Alvo	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_MES SAGE , ERROR_NUMBER , ERROR_PROCEDURE , ERROR_SEV ERITY e ERROR_STATE do sistema para tratamento de erros.
IBM Db2 para z/OS	Aurora MySQL Aurora PostgreSQ L MySQL PostgreSQ L	Foi adicionado suporte ao IBM Db2 para a z/OS versão 12 como fonte para projetos de migração de banco de dados em AWS SCT. Para obter mais informações, consulte Conectando-se à IBM DB2 para z/OS.
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	Alvo	Novidades, aprimoramentos ou correções
Banco de dados do Microsoft Azure SQL Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um erro do transformador para procedimentos com a instrução do conjunto SET NOCOUNT ON.
Banco de dados do Microsoft Azure SQL Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão da função CONCAT quando um valor de entrada é uma variável do tipo definido pelo usuário.
Banco de dados do Microsoft Azure SQL Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que a função DATEPART era convertida incorretamente.
Microsoft SQL Server	Babelfish for Aurora PostgreSQ L	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 convertid as incorretamente.

Origem	Alvo	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 obter mais informações, consulte SSIS para 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	PostgreSQ L	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	PostgreSQ L	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	PostgreSQ L	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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Foi adicionada uma opção para emular tabelas SET no código convertido. Para essa emulação, AWS SCT suportes 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	Alvo	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 obter mais informações, consulte <u>Criando um relatório de avaliação de vários servidores no AWS Schema Conversion Tool</u> .
Todos	Todos	Coleta de estatísticas aprimorada para objetos convertidos.
Todos	PostgreSQ 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 Microsoft SQL Server	Aurora PostgreSQ L	Melhorada a conversão de tipos de dados XML.
Banco de dados do Microsoft Azure SQL Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que cláusulas NOT LIKE eram convertid as incorretamente.
Banco de dados do	Aurora PostgreSQ L	Corrigido um erro de transformador para procedimentos com instruções INSERT, DELETE e UPDATE que incluem a cláusula OUTPUT.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft Azure SQL	PostgreSQ L	
Microsoft SQL Server		
Banco de dados do Microsoft Azure SQL Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um erro do transformador para procedimentos com a instrução RETURN @@ROWCOUNT .
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 PostgreSQ L PostgreSQ L	Corrigido um erro de transformador para construtores de valores de tabela.
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	Alvo	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	PostgreSQ L Aurora PostgreSQ L	Foi adicionado suporte para conversão de índices para visões materializadas.
Oracle	PostgreSQ L Aurora PostgreSQ L	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	Alvo	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 convertid o.
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	Alvo	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-fo nte 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 PostgreSQ L	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 maiúscula s 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	PostgreSQ L	Melhorada a conversão de procedimentos que incluem instruções UPDATE com condições de junção.

Origem	Alvo	Novidades, aprimoramentos ou correções
	Aurora PostgreSQ L	
Microsoft SQL Server	PostgreSQ L Aurora PostgreSQ L	Foi aprimorada a conversão de acionadores, procedimentos armazenados e funções que incluem o valor após o sinal de igual.
Microsoft SQL Server	PostgreSQ L Aurora PostgreSQ L	Corrigido um erro do transformador para procedimentos com a instrução DELETE e o operador OR.
Microsoft SQL Server	PostgreSQ L Aurora PostgreSQ L	Melhorada a conversão da cláusula 0UTPUT.
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	Alvo	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	PostgreSQ L Aurora PostgreSQ L	Implementada a conversão de código SQL em aplicativos C#. Para obter mais informações, consulte Código SQL em aplicativos C#.
Oracle	PostgreSQ L Aurora PostgreSQ L	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	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L Aurora PostgreSQ L	Melhorada a conversão de partições de hash sem a restrição NOT NULL.
Overale		Tai adicionada aurante nono comunessa de machricsas de Orante
Oracle	Aurora PostgreSQ L	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	PostgreSQ L Aurora PostgreSQ L	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	Alvo	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	Alvo	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	Alvo	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 compatibi lidade com o MySQL 8.0.
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQ L MySQL PostgreSQ L	Corrigido um problema em que o item de ação 810 não aparecia quando necessário.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de procedimentos com instruções UPDATE DELETE e INSERT.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um problema em que o item de ação 7810 não aparecia quando necessário.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de uma instrução EXEC que está aninhada dentro de uma instrução IFELSE .
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Foi aprimorada a conversão de visualizações indexadas.
Netezza	Amazon Redshift	Agentes de migração de dados aprimorados por meio do rastreame nto de transações em tempo real durante a carga total da operação de captura de dados de alteração (CDC). Agora você pode interromp er 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 interromp er uma tarefa com o CDC.

Origem	Alvo	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 PostgreSQ L PostgreSQ L	Melhorada a conversão da função SYSDATE e adicionou a capacidad e de alterar o fuso horário nas Configurações de conversão.
Oracle	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que instruções dinâmicas não eram convertidas.
Oracle	Aurora PostgreSQ L PostgreSQ L	Corrigido um problema em que o código convertido não incluía nomes gerados pelo sistema.
Oracle DW	Aurora PostgreSQ L PostgreSQ L	Melhorada a conversão de instruções SELECT que estão aninhadas em acionadores.
Oracle DW	Amazon Redshift	Melhorada a conversão das funções T0_DATE, T0_TIMESTAMP e T0_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	Alvo	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 MULTISET 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	Alvo	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	Alvo	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	Alvo	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 PostgreSQ L PostgreSQ L	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.
Banco de dados do SQL Azure Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Foi aprimorada a conversão de funções com valor de tabela.
Banco de dados do SQL Azure Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Resolvido um problema em que o argumento 0UT em um procedime nto armazenado com o valor de retorno padrão não era convertido no argumento INOUT.
Greenplum	Amazon Redshift	Estratégias de otimização aprimoradas ao encontrar as tabelas e colunas mais usadas na tabela QueryLog.

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	 Problemas corrigidos com a conversão do seguinte: Operador de atribuição de concatenação de strings (+=) Função do SCOPE_IDENTITY Tipo de dados do varchar(max)
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Foi aprimorada a conversão de visualizações com funções não suportadas.
Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Corrigido um problema em que funções não suportadas como argumento para outra função eram convertidas incorretamente.
Microsoft SQL Server	Babelfish for Aurora PostgreSQ 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.

Alvo	Novidades, aprimoramentos ou correções
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.
AWS Glue	Resolvido um problema em que os scripts dos objetos de origem não eram exibidos na interface do usuário.
Amazon Redshift	Estratégias de otimização aprimoradas ao escolher a tabela de fatos e as dimensões apropriadas para colocação.
Aurora PostgreSQ L PostgreSQ L	Resolvido um problema para converter corretamente os acionadores Oracle, que usam números de sequência.
Aurora PostgreSQ L PostgreSQ	Foi aprimorada a conversão de visualizações com links de bancos de dados públicos.
	Amazon Redshift Aws Glue Amazon Redshift Aurora PostgreSQ L Aurora PostgreSQ L

Origem	Alvo	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 convertid as 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 LOADER ERROR.
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 CAST.
Teradata	Amazon Redshift	Corrigida uma conversão quebrada para ST_Line_Interpolat e_Point .
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 DATABASE.
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	Alvo	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	Foi 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: 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 cassandra.yaml ou da pasta conf.
Cassandra	Amazon DynamoDB	Foi adicionado 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	Alvo	Novidades, aprimoramentos ou correções
Cassandra	Amazon DynamoDB	Melhorada a conversão alterando uma propriedade no arquivo cassandra-env.yaml 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: An I/O error occurred while sending to the backend .
Greenplum Microsoft SQL Server DW	Amazon Redshift	Resolvido um problema em que a opção Apply RTRIM to string columns não era exibida na interface do usuário.
Microsoft SQL Server	Babelfish for Aurora PostgreSQ L	Foi adicionado 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 PostgreSQ L PostgreSQ L	Implementada a capacidade de gerar nomes exclusivos para índices.

Origem	Alvo	Novidades, aprimoramentos ou correções
SAP ASE	Aurora PostgreSQ L PostgreSQ 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/esq uemas 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 Al 13177 agora fosse comentado no script convertido.
Teradata	Amazon Redshift	Corrigiu uma conversão interrompida de tabelas temporais.

Origem	Alvo	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 Al 17008.

Problemas resolvidos:

· Melhorias gerais.

Notas de lançamento do AWS SCT Build 655

Origem	Alvo	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.

Notas de release: 655 Versão 1.0.672 636

Origem	Alvo	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: Failed to compile udf .
Oracle DW	Amazon Redshift	Melhorada a conversão usando instruções de tipo apropriadas, como pls-type para parâmetros %ROWTYPE.
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 FastExport -> EXPORT -> 'null' era exibido em vez de 'CAST' após a conversão.
Teradata	Amazon Redshift	Resolvido um problema em que algumas funções de um pacote de extensão falhavam quando aplicadas com Cause:[JDBC Driver]String index out of range: 0 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.

Notas de release: 655 Versão 1.0.672 637

Origem	Alvo	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	PostgreSQ L Aurora PostgreSQ L	Resolvido um problema em que o script de destino de objetos com a INSTRUÇÃO SAVE EXCEPTIONS era alterado após a reconversão.
Oracle	PostgreSQ L Aurora PostgreSQ L	Resolvido um problema em que um campo errado era especificado na cláusula ORDER BY após a conversão proc_cursor_with_c alc_columns .
Oracle	PostgreSQ L Aurora PostgreSQ L	Resolvido: uma instrução de aws_oracle_ext\$array_id\$tem porary variável extra é necessária em uma conversão ASSOCIATI VE COLLECTION.
Oracle	PostgreSQ L Aurora PostgreSQ L	Resolvido: a conversão errada de uma CHAVE PRIMÁRIA com o mesmo nome de um ÍNDICE pertencente à mesma tabela.

Notas de release: 655 Versão 1.0.672 638

Problemas resolvidos:

• Melhorias gerais.

Notas de lançamento do AWS SCT Build 654

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L Aurora PostgreSQ L	Resolvido um problema com pseudocolunas de consulta hierárquica, erro de análise de colunas PRIOR.
Oracle	PostgreSQ L Aurora PostgreSQ L	Resolvido um problema para converter corretamente um comentário de várias linhas contendo uma barra e um asterisco (/*).
Oracle	PostgreSQ L Aurora PostgreSQ L	Foi adicionada a emulação USER_COL_COMMENTS da visualiza ção do sistema ao pacote de extensões.
Oracle	PostgreSQ L Aurora PostgreSQ L	Foi aprimorada a conversão de literais citados.
DB2 LUW	PostgreSQ L	Foi aprimorada a conversão de instruções LABEL que adicionam ou substituem rótulos nas descrições de tabelas, visualizações, aliases ou colunas.

Notas de release: 654 Versão 1.0.672 639

Origem	Alvo	Novidades, aprimoramentos ou correções
	Aurora PostgreSQ L	
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 incorreta mente.
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_NORMAL IZE_OVERLAP estava eliminando linhas.
Teradata	Amazon Redshift	Agora oferece suporte à verificação estrita de data para a função TO_DATE aprimorada.

Notas de release: 654 Versão 1.0.672 640

Origem	Alvo	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 Al 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—R edshift 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.

Notas de release: 654 Versão 1.0.672 641

Origem	Alvo	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	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L	Implementada a capacidade de converter SQL dinâmico criado em funções ou procedimentos chamados.
	Aurora PostgreSQ L	
Oracle	PostgreSQ L	Conversão SQL dinâmica aprimorada: parâmetros IN como variáveis de ligação.

Notas de release: 653 Versão 1.0.672 642

Origem	Alvo	Novidades, aprimoramentos ou correções
	Aurora PostgreSQ L	
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\$: • 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	Conversão aprimorada de elementos de Date/Timestamp formato:TO_DATE,TO_TIMESTAMP , e TO_TIMESTAMP_TZ

Notas de release: 653 Versão 1.0.672 643

Origem	Alvo	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_NORMAL IZE_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	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	PostgreSQ L	Adicionado bloqueio de aplicativos para funções sp_getapplock e sp_releaseapplock .
Nenhum	Amazon Redshift	Melhoria da Interface de Linha de Comandos (CLI): implementado o modo de Comando de Script.
Oracle	PostgreSQ L	Implementada a amostragem de parâmetros de rotina dentro do SQL dinâmico.

Notas de release: 652 Versão 1.0.672 644

Origem	Alvo	Novidades, aprimoramentos ou correções
	Aurora PostgreSQ L	
Oracle	PostgreSQ L Aurora PostgreSQ L	Melhorias na conversão de SQL dinâmico criado em funções ou procedimentos chamados.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQ 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	PostgreSQ L	Problema resolvido que causava a mensagem de erro "9996 — Gravidade crítica — Ocorreu um erro no transformador" ao usar DB2 LUW como fonte.
Teradata	Amazon Redshift	Suporte para expressões de tabela recursivas no próximo lançament o 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).

Notas de release: 652 Versão 1.0.672 645

Origem	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Suporte a identificadores com distinção entre maiúsculas e minúsculas por meio da configuração do usuário, enable_ca se_sensitive_identifier . 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 corretame nte.
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 release: 652 Versão 1.0.672 646

Notas de lançamento do AWS SCT Build 651

Origem	Alvo	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	PostgreSQ L	Foi adicionado suporte para conversão de funções STR().
MS SQL Server	PostgreSQ L	Foi adicionado suporte para converter o operador EXOR bit a bit (^ no Microsoft SQL Server) em PostgreSQL como operador #.
Oracle	PostgreSQ L	Resolveu um problema em que a aws_oracle_ext.UNI STR(null) 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_IN TVAL_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.

Notas de release: 651 Versão 1.0.672 647

Origem	Alvo	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 visualiza ção.
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 release: 651 Versão 1.0.672 648

Notas de lançamento do AWS SCT Build 650

Origem	Alvo	Novidades, aprimoramentos ou correções
Todos	Todos	 Uso atualizado e aprimorado de agentes extratores, incluindo: 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. Para obter mais informações, consulte Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.
Todos	Amazon RDS PostgreSQ L 13	AWS SCT agora oferece suporte ao Amazon RDS PostgreSQL 13 como destino.
Microsoft SQL Server	Aurora PostgreSQ L	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 implement adas.
Oracle DW	Amazon Redshift	Melhorias implementadas na conversão de instruções SQL dinâmicas.

Notas de release: 650 Versão 1.0.672 649

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle DW	Amazon Redshift	Melhorias implementadas na conversão de UDF de SQL.
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 Microsoft SQL Server Azure Azure Synapse	Todos	Atualizada a versão mínima compatível do driver JDBC para Microsoft SQL Server, Azure e 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 release: 650 Versão 1.0.672 650

Notas de lançamento do AWS SCT Build 649

Origem	Alvo	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 TO_BINARY_DOUBLE TO_BINARY_FLOAT TO_NUMBER TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ TO_DSINTERVAL VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	Aprimoramentos de função implementados para Processamento Aproximado de Consultas, como: Funções agregadas • ANY_VALUE • APPROX_COUNT_DISTINCT

Notas de release: 649 Versão 1.0.672 651

Origem	Alvo	Novidades, aprimoramentos ou correções
		· APPROX_COUNT_DISTINCT_DETAIL · APPROX_COUNT_DISTINCT_AGG · LISTAGG · TO_APPROX_COUNT_DISTINCT
Teradata	Amazon Redshift	Aprimoramentos de conversão implementados para seleção automática de chaves de classificação e distribuição do Teradata. O mecanismo de banco de dados seleciona chaves de classific ação 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.
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	PostgreSQ L Aurora	Suporte implementado para conversão de DYNAMIC RESULTS SETS, incluindo:
	PostgreSQ L	Cláusula do cursor COM RETURN/WITH RETORNO AO CLIENTE conversão de cláusulas de rotina DYNAMIC RESULT SETS

Notas de release: 649 Versão 1.0.672 652

Origem	Alvo	Novidades, aprimoramentos ou correções
Microsoft SQL Server	Aurora PostgreSQ L	Suporte implementado para o PostgreSQL atual do Aurora RDS como destino.
Oracle		
DB2 LUW		
SAP ASE		
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 PostgreSQ L	Foi adicionado suporte à função XMLFOREST para conversão do Oracle para o Aurora PostgreSQL.

Problemas resolvidos:

• Melhorias gerais.

Notas de release: 649 Versão 1.0.672 653

Notas de lançamento do AWS SCT Build 648

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L Amazon Aurora Edição Compatíve I com PostgreSQ L	Modo de aplicação personalizado do pacote de extensões Aurora PostgreSQL implementado: operadores e tipos de texto. numeric/d ate
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQ L	Configuração do Aurora PostgreSQL Lambda Invoke implement ada: criação da extensão aws_lambda; atribuição do perfil do IAM ao cluster Aurora PostgreSQL. Oracle — E-mails, trabalhos, filas, arquivos WebAgent DB2— E-mails, tarefas, arquivos Microsoft SQL Server: e-mails, agente
Oracle	PostgreSQ L	A refatoração de conversão de instruções FORALL foi implementada: Instrução FORALL FORALL SAVE EXCEPTIONS RETURNING INTO com BULK COLLECT Coleção do sistema SQL%BULK_EXCEPTIONS

Notas de release: 648 Versão 1.0.672 654

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle DW 18, 19	Amazon Redshift	Melhorias de conversão de Oracle para Amazon Redshift implement adas: conversão aprimorada integrada. LISTAGG agregado; LISTAGG analítico.
Oracle DW 18, 19	Amazon Redshift	Melhorias na conversão de Oracle para Amazon Redshift implement adas: consulte novos atributos.
Vertica	Amazon Redshift	Melhorias na conversão de Vertica para Amazon Redshift implement adas: 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	Alvo	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

Notas de release: 647 Versão 1.0.672 655

Origem	Alvo	Novidades, aprimoramentos ou correções
		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. Corte nomes com mais de 64 caracteres da seguinte forma:
		<pre>[first N chars]() + "" + [checksum]()</pre>
		<pre>[first N chars] = 64 - 1 - [length of checksum string]</pre>
		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
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 em Table/ View colunas.
Teradata	Amazon Redshift	Foi adicionado suporte para conversão de tipo de dados TIME.
Teradata	Amazon Redshift	Melhorias na conversão: TD_NORMALIZE_OVERLAP implement ado.
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.

Notas de release: 647 Versão 1.0.672 656

Origem	Alvo	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	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L	Implementação aprimorada do modelo no formato TM.
Oracle	PostgreSQ L	A implementação da máscara no formato SP fornece suporte básico para o sufixo SP, somente para o idioma inglês.
Oracle	PostgreSQ L	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	Codificação Amazon Redshift AZ64 com AWS SCT — Codificação de compressão adicionada para alguns tipos de AZ64 dados
Teradata	Amazon Redshift	Foi adicionado suporte para conversão de transações implícitas.

Notas de release: 646 Versão 1.0.672 657

Origem	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Foi adicionado suporte para funções geoespaciais internas do Teradata: Métodos ST_LineString
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	Tamanho da conversão de caracteres — PL/pgSQL Conversão 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].

Notas de release: 646 Versão 1.0.672 658

 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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Forneça uma solução para resolver visualizações não totalment e 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 comoCHAR(N), eles são convertidos no Amazon VARCHAR(3*N) Redshift.
Teradata	Amazon Redshift	Fornece a conversão de CAST no Teradata entre datas e números. SELECT Cast('2020-07-17' AS BIGINT) SELECT Cast(20200630 - 190000000 AS DATE)
Teradata	Amazon Redshift	Suporte à conversão de tipos de dados Teradata PERIOD em duas colunas TIMESTAMP do Amazon Redshift: PERIOD(TIMESTAMP) PERIOD(TIMESTAMP WITH TIMEZONE)
Teradata	Amazon Redshift	Suporte à conversão da função Teradata RANK com cláusula RESET WHEN.

Notas de release: 645 Versão 1.0.672 659

Origem	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Suporte aprimorado do CAST em conversões explícitas de tipos de dados e implícitas CASTs em expressões.
Teradata	Amazon Redshift	Relatório de padrões de subconsulta correlacionados não suportado s. Para obter mais informações, consulte <u>Subconsultas correlacionadas</u> no Guia do desenvolvedor de banco de dados do Amazon Redshift.
none	Amazon Redshift	Tabelas aprimoradas limitam o suporte para tipos de RA3 nós.
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	PostgreSQ L	A opção convert_procedures_to_function 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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Várias melhorias na conversão. Conversões aprimoradas de QUALIFY com o alias da tabela.
		Conversões aprimoradas com o operador IN.

Notas de release: 644 Versão 1.0.672 660

Origem	Alvo	Novidades, aprimoramentos ou correções
		 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 J0IN().
		Foi aprimorada a conversão de macros em procedimentos armazenados.
		Foram adicionados comandos AWS SCT CLI especiais que podem analisar os sql/bteq scripts fornecidos e gerar um relatório sobre o número de estruturas de sintaxe encontradas no código-fonte.
		Contagem de comandos BTEQ
		· Contagem de HANDLERS
		Contagem de casos CAST
		Contagem de DML/DDL casos
		Conte com DMLs visualizações atualizáveis
		Foi adicionado um item de ação do relatório de avaliação: colunas Teradata com formatos de data personalizados não são suportadas no Amazon Redshift.

Notas de release: 644 Versão 1.0.672 661

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L/Aurora PostgreSQ L	Funcionalidade adicionada para salvar scripts de instalação do pacote de extensão. Nível de severidade alterado para Al 5334. Melhor desempenho do uso de um registro como variável de pacote IMPLEMENTATION . Adicionado suporte à função de agregação XMLAGG
IBM Db2	PostgreSQ L/Aurora PostgreSQ L	Adicionado carregamento e conversão de comentários na implement ação de objetos de armazenamento.
MS SQL DW	Amazon Redshift	Melhoria na conversão: problema resolvido com PATINDEX. Melhorias da UI: 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.

Notas de release: 642 Versão 1.0.672 662



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	Alvo	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 obter mais informações, consulte Convertendo SSIS em com AWS GlueAWS SCT.	r
Oracle	MariaDB/S QL MODE=C LE/MySQL/ AmazonAur ora MySQL	Implementou a seção de PL/SQL declaração na cláusula WITH.	
Oracle	PostgreSQ L/Aurora PostgreSQ L	Foi adicionado suporte para DBMS_SESSION.RESET_PACKAGE DBMS_SESSION.MODIFY_PACKAGE .	е
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 release: 642 Versão 1.0.672 663

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	Alvo	Novidades, aprimoramentos ou correções
Oracle/ MS SQL/ MySQL/ PostgreS QL/Db2 LUW	Todos	Produza cálculos do Time Report no arquivo.csv.
Teradata	Amazon Redshift	Foi adicionado 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 release: 641 Versão 1.0.672 664

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



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	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L	Implementada a conversão de código SQL em aplicativos Java e Pro*C.

Notas de release: 640 Versão 1.0.672 665

Origem	Alvo	Novidades, aprimoramentos ou correções
	Aurora PostgreSQ L	
Oracle	PostgreSQ L Aurora PostgreSQ L	Desempenho aprimorado das funções a seguir quando usadas em uma cláusula WHERE: • aws_oracle_ext.to_date • aws_oracle_ext.to_char • aws_oracle_ext.to_number • aws_oracle_ext.sysdate • aws_oracle_ext.sys_context
Oracle	RDS MariaDB 10.4	Adição de suporte ao RDS MariaDB 10.4 para todos os fornecedores de processamento transacional online (OLTP).
Oracle	PostgreSQ L/Aurora PostgreSQ L	Adição de suporte para DBMS_UTILITY.GET_TIME. Adição das seguintes emulações: DBMS_UTILITY.GET_TIME DBMS_UTILITY.FORMAT_CALL_STACK DBMS_UTILITY.CURRENT_INSTANCE

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	MariaDB/M ySQL/Auro ra MySQL/ Microsoft SQL Server Mode=Orac le/Postgr eSQL/ Aurora PostgreSQ L/RDSOrác ulo	Adição de suporte à cláusula de compartilhamento para TABLE(DAT A,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) e SEQUENCE(DATA)
Oracle	PostgreSQ L/Aurora PostgreSQ L/OracleR DS	A definição DEFAULT de uma coluna pode ser estendida para que o DEFAULT seja aplicado para inserção NULL explícita. 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.
Oracle	MariaDB/ MariaDB (SQL MODE=ORA LE)	Adição de suporte para "Colunas de Identidade", que são increment adas 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	Alvo	Novidades, aprimoramentos ou correções
Oracle	MariaDB 10.2/ MariaDB 10.3/Post greSQL MySQL/ Aurora MySQL/ Pos tgreSQL/A urora	A cláusula DEFAULT tem uma nova cláusula 0N 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/ Pos tgreSQL/ Aurora PostgreSQ L	Adição de suporte para colunas IDENTITY.
Oracle	MySQL 8.x	Adição de suporte para restrição CHECK.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L/Aurora PostgreSQ L	Implementou a verificação de ANYDATA IS NULL/IS NOT NULL usando a rotina do pacote de extensão. Implementou a emulação da função VALUE usada em uma consulta baseada na função TABLE de XMLSequence. Adição de suporte a DBMS_LOB para as seguintes rotinas internas: DBMS_LOB.CREATETEMPORARY DBMS_LOB.FREETEMPORARY DBMS_LOB.APPEND
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.
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	Implementação da conversão de atributos de cursor em blocos aninhados. 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. Implementado o estilo de distribuição do Amazon Redshift = AUTO.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	PostgreSQ L/Aurora PostgreSQ L	Se uma palavra não reservada no Oracle estiver reservada no PostgreSQL, o seguinte será verdadeiro: * 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. Implementação da capacidade de usar funções como entrada para funções LTRIM, RTRIM e TRIM. Expressões SELECT DISTINCT, ORDER BY devem aparecer na lista de seleção. Para parâmetros do cursor que seguem um parâmetro com um valor DEFAULT, AWS SCT adiciona a cláusula DEFAULT IS NULL Os parâmetros do cursor Source OUT são convertidos em parâmetro s do cursor IN. 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". Implementação de suporte pragma AUTONOMOUS_TRANSACTION com dblink e pg_background.
Oracle	Todos	Implementação da visualização SYS_%_TAB_COMMENTS.
Oracle	PostgreSQ L	Entradas variáveis para filtros não são compatíveis com o PostgreSQ L. Na conversão do Oracle em PostgreSQL, se um filtro de variável for encontrado, uma exceção será relatada agora.

Origem	Alvo	Novidades, aprimoramentos ou correções
Oracle	Amazon Redshift	Implementação de melhorias de conversão do FORLOOP Cursor do código armazenado.
		Implementou a invocação de código armazenado function/procedure s com parâmetros padrão.
		Implementação de capacidade de código armazenado para UPDATE com alias sem a cláusula WHERE.
		As funções de código armazenado implementadas pré-formam casos adicionais com SELECT From dual.
		Implementação de parâmetros Table%ROWTYPE e variáveis de pacote do código armazenado.
		O código armazenado implementado usava JAVA e procedimentos externos.
		Implementação do pacote Oracle standard em código armazenado.

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	Alvo	Novidades, aprimoramentos ou correções
Microsoft Azure/Mic rosoft SQL Server	PostgreSQ L/Aurora PostgreSQ L/MySQL/ Aurora MySQL	Adição de suporte para índices COLUMN STORE.

Oniona	A.L	No. idealar and in the control of th
Origem	Alvo	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/M ySQL/Auro ra MySQL/ Pos tgreSQL/ Aurora PostgreSQ L	Adição de suporte para o atributo de índice OPTIMIZE_FOR_SEQUE NTIAL_KEY.
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQ L	Adição de suporte para tipos de tabela Nó de Bancos de Dados e Edge.
Azure/SQL Server	MariaDB/M ySQL/Auro ra MySQL/ Pos tgreSQL/ Aurora PostgreSQ L	Adição de suporte para TEMPORAL TABLES.

Origem	Alvo	Novidades, aprimoramentos ou correções
Todos	Todos	Atualize o Amazon Corretto JDK 8 para o JDK 11. Para mais informações, incluindo links para download, consulte <u>O que é o Amazon Corretto 11?</u> 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/ Pos tgreSQL/ Aurora PostgreSQ L/MariaDB	Adição de suporte para processamento DML para SQL Server Graph Architecture.
SQL Server	Aurora PostgreSQ L	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	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.
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	Alvo	Novidades, aprimoramentos ou correções
SQL Server	AWS Glue (Concha Python)	 Melhorias de conversão, incluindo: 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	PostgreSQ L/Aurora PostgreSQ L	Implementação que torna os procedimentos \$TMP opcionais.
SQL Server	MySQL/ Aurora MySQL	Operações aritméticas estendidas com datas. Emulação de construção 'TOP (expressão) WITH TIES. Depois de chamar procedimentos com o refcursor gerado fora, o refcursor agora é fechado. A definição de um nível de isolamento GLOBAL não é compatíve I 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.

Origem	Alvo	Novidades, aprimoramentos ou correções
SQL Server	AWS Glue (Concha Python)	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. Implementou o encerramento de uma série de instruções Transact-SQL para código armazenado para que um grupo de instruçõe s Transact-SQL possa ser executado mesmo que o Python não ofereça suporte a BEGIN e END do SQL Server como. control-of-flow 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.
SQL Server	Amazon Redshift	Implementação do processamento condicional de instruções Transact-SQL para código armazenado implementando o IF Controle ELSE. 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. Implementação de SET e SELECT em código armazenado. Implementação de CREATE INDEX no Amazon Redshift (que não oferece suporte para índices) criando uma chave de classificação especificada pelo usuário nas tabelas.

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	Alvo	Novidades, aprimoramentos ou correções
MySQL	PostgreSQ 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 <u>O que é o Amazon Corretto 11?</u> 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	PostgreSQ L/Aurora PostgreSQ L 11.	 Suporte adicionado para o seguinte: Incorporação de transações em procedimentos armazenados SQL. A capacidade de chamar procedimentos armazenados SQL. A capacidade de criar procedimentos armazenados SQL.
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.
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	Alvo	Novidades, aprimoramentos ou correções
PostgreSQ L	MySQL 8.x	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. O MySQL agora é compatível com CTE e CTE recursivo.
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.
PostgreSQ L 11.x	PostgreSQ L/Aurora PostgreSQ L 11.	 Suporte adicionado para o seguinte: Incorporação de transações em procedimentos armazenados SQL. A capacidade de chamar procedimentos armazenados SQL. A capacidade de criar procedimentos armazenados SQL.
PostgreSQ L	MySQL 8.x	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. Adição de suporte do MySQL ao uso de expressões como valores padrão em especificações de tipo de dados, incluindo expressõe s como valores padrão para os tipos de dados BLOB, TEXT,
		GEOMETRY e JSON. Agora várias funções agregadas podem ser usadas como funções de janela: • AVG()

Origem	Alvo	Novidades, aprimoramentos ou correções
		• 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()
		O MySQL oferece suporte a funções de janela que, para cada linha de uma consulta, realizam um cálculo usando linhas relacionadas àquela linha.
		• CUME_DIST()
		• DENSE_RANK()
		• FIRST_VALUE()
		• LAG()
		• LAST_VALUE()
		• LEAD()
		• NTH_VALUE()
		• NTILE()

Alvo	Novidades, aprimoramentos ou correções
	PERCENT_RANK()RANK()ROW_NUMBER()
MySQL 8.x	Adição de suporte para restrição CHECK.
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.
Todos	Implementação da capacidade de substituir partições físicas por partições virtuais. Extratores de data warehouse extraem dados de acordo com as partições virtuais criadas.
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.
	MySQL 8.x SQL Server

Alterações do release 1.0.640 do Db2 LUW

A tabela a seguir lista as alterações da compilação 1.0.640 nas quais o DB2 LUW é o mecanismo de origem.

Origem	Alvo	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).
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	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.
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	Alvo	Novidades, aprimoramentos ou correções
Teradata	Amazon Redshift	Adição de suporte para as instruções MERGE e QUALIFY.

Origem	Alvo	Novidades, aprimoramentos ou correções
		Remoção da cláusula LOCKING ROWS FOR ACCESS de instruções Teradata.
		Adição de suporte para a função CAST.
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_SU BSTR().
Todos	Todos	Adição de informações ao relatório de avaliação sobre possíveis inconsistências no banco de dados do usuário.
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.
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	Implementou a capacidade de salvar o SQL da árvore de origem em um único arquivo ou em vários arquivos por estágio usando as configurações em Configurações do projeto, Salvar como SQL e Aplicar, Lista suspensa: arquivos únicos file/Multiple.
		Implementação de melhorias implementadas em visualizações e conversões de procedimentos.

Origem	Alvo	Novidades, aprimoramentos ou correções
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	Alvo	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: • MariaDB 10.4 • instrução EXECUTE IMMEDIATE • Definições DEFAULT • Suporte à restrição CHECK
SAP ASE	PostgreSQ 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.

Origem	Alvo	Novidades, aprimoramentos ou correções
		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.
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 substituí das por NULL como um espaço reservado.
Sybase	PostgreSQ L	Adição de suporte para funções nativas.
SAP ASE	MySQL/ Aurora MySQL	O nível de isolamento padrão para o Aurora MySQL é REPEATABL E 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 projetado s para serem executados com READ COMMITTED podem precisar ser modificados. Ou você pode alterar explicitamente o padrão para READ COMMITTED.
SAP ASE	PostgreSQ L	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.
SAP ASE	L MySQL/ Aurora MySQL PostgreSQ L	O nível de isolamento padrão para o Aurora MySQL é RE READ. A definição de um nível de isolamento GLOBA compatível com o Aurora MySQL. Somente o escopo da pode ser alterado. O comportamento padrão das transac REPEATABLE READ e leituras consistentes. Aplicativos s para serem executados com READ COMMITTED pode ser modificados. Ou você pode alterar explicitamente o READ COMMITTED. Adição de suporte para a função CONVERT(optimistic) de extensão. Adição de emulação de sysindexes de visualização do se Se houver uma instrução SELECT em um procedimento especificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de suporte de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva de sepecificar INTO, o parâmetro INOUT p_refcur do tipo respectiva d

Origem	Alvo	Novidades, aprimoramentos ou correções
Greenplum	Amazon Redshift	Implementação de CREATE TEMPORARY TABLE da seguinte forma:

Novos recursos no AWS Schema Conversion Tool

A tabela a seguir descreve as mudanças importantes no guia do usuário AWS Schema Conversion Tool (AWS SCT) após janeiro de 2018.

É possível se inscrever em um feed RSS para ser notificado sobre atualizações nessa documentação.

Alteração	Descrição	Data
AWS SCT compilação #1 .0.672	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
AWS SCT compilação #1 .0.671	A compilação 1.0.671 fornece suporte às migrações do Apache Oozie para o. AWS Step Functions Ele também adiciona suporte BigQuery como fonte 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.

AWS SCT compilação #1 .0.670

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

AWS SCT compilação #1 .0.669

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

AWS SCT compilação #1 .0.668

A compilação 1.0.668 implementa o particion amento 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

AWS SCT compilação #1 .0.667

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 compatíve I com o Amazon Redshift e resolve vários problemas de conversão.

13 de outubro de 2022

AWS SCT compilação #1 .0.666

O Build 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

AWS SCT compilação #1 .0.665

A compilação 1.0.665 fornece suporte BigQuery como fonte de migração. Ela também implementa o suporte da nova versão do arquivo de configura ção de atributos do Babelfish . Além disso, ela melhora a conversão de data warehouse s para o Amazon Redshift e resolve vários problemas de conversão.

29 de agosto de 2022

AWS SCT compilação #1 .0.664

A compilação 1.0.664 fornece suporte ao Amazon Redshift sem servidor como origem ou destino de migração. Ele também implementa o balanceamento automático de memória nas tarefas de extração de dados e corrige um erro em que AWS SCT não foi possível se conectar aos dispositivos. AWS Snowball Edge 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

AWS SCT compilação #1 .0.663

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

AWS SCT compilação #1 .0.662

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

AWS SCT compilação #1 .0.661

A compilação 1.0.661 fornece suporte ao IBM Db2 para z/
OS como origem de migração.
Ele também adiciona suporte à conversão de scripts de extração, transformação e carregamento (ETL) AWS
Glue Studio e resolve vários problemas de conversão.

21 de abril de 2022

AWS SCT compilação #1 .0.660

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 visões materializadas e resolve vários problemas de conversão.

21 de março de 2022

AWS SCT compilação #1 .0.659

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 implement a 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

AWS SCT compilação #1 .0.658

O Build 1.0.658 fornece integração AWS Secrets Manager e adiciona suporte à plataforma de banco de dados de destino virtual Amazon Redshift. Ela também adiciona várias melhorias de conversão e correções de bugs.

20 de janeiro de 2022

AWS SCT compilação #1 .0.657

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

AWS SCT compilação #1 .0.656

A compilação 1.0.656 fornece suporte a vários bancos de dados de origem e destino em um projeto. Ela também adiciona conversão, estratégi a de otimização, melhorias gerais e várias correções de bugs.

22 de novembro de 2021

AWS SCT compilação #1 .0.655

O Build 1.0.655 implement a a conversão de 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

AWS SCT compilação #1 .0.654

A compilação 1.0.654 implementa a conversão de scripts Shell, Teradata e Teradata FastLoad Basic Teradata Query (BTEQ) para Amazon Redshift RSQL. Ela também resolve vários problemas de conversão e adiciona várias melhorias e correções de bugs.

16 de setembro de 2021

AWS SCT compilação #1 .0.653

A compilação 1.0.653 implementa a conversão de SQL dinâmico criado em funções ou procedime ntos chamados. Ela também melhora a conversão de rotinas criptografadas e adiciona várias melhorias e correções de bugs.

10 de agosto de 2021

AWS SCT compilação #1 .0.652

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

AWS SCT compilação #1 .0.651

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

AWS SCT compilação #1 .0.650

A compilação 1.0.650 implementa o suporte do Amazon RDS para PostgreSQ L 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

AWS SCT compilação #1 .0.649

A compilação 1.0.649 implementa o suporte ao MariaDB 10.5 como banco de dados de destino e implement a 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

AWS SCT compilação #1 .0.648

A compilação 1.0.648 adiciona várias melhorias de conversão

e correções de bugs.

22 de fevereiro de 2021

AWS SCT compilação #1 .0.647

A compilação 1.0.647 adiciona suporte ao atributo Database Mail no Amazon RDS, implementa o carregamento e a conversão de comentári os em objetos de armazenam ento. Ele também adiciona o AWS SCT Data Migration Service Assessor and AWS SCT Wizard e implement a 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.

15 de janeiro de 2021

AWS SCT compilação #1 .0.646

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.

28 de dezembro de 2020

AWS SCT compilação #1 .0.645

A compilação 1.0.645 adiciona suporte para ETL SSIS para AWS Glue conversão e várias melhorias e correções de erros.

16 de novembro de 2020

AWS SCT compilação #1 .0.643-1.0.644

A compilação 1.0.644 adiciona várias melhorias de conversão , desempenho e interface de usuário e correções de bugs.

14 de outubro de 2020

AWS SCT compilação #1 .0.642	A compilação 1.0.642 implementa a conversão de pacotes ETL do Microsoft SQL Server Integration Services AWS Glue para e adiciona uma série de melhorias e correções de erros.	28 de agosto de 2020
AWS SCT compilação #1 .0.641	Foi adicionado suporte SSL para extratores de dados. A compilação também inclui várias melhorias e correções.	17 de julho de 2020
AWS SCT constrói #1 .0.633-1.0.640	Atualização do JDK 8 para o Amazon Corretto JDK 11. Adição de tabelas que identific am outras atualizações, alterações e correções.	22 de junho de 2020
Disponibilidade do AWS WQF	AWS SCT não está mais fornecendo a ferramenta AWS Workload Qualification Framework (AWS WQF) para download.	19 de junho de 2020
AWS SCT constrói #1 .0.632	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

AWS SCT constrói #1 .0.631 e #1 .0.630 (combinadas)

Melhor suporte ROWIDs no
Oracle e para objetos do
sistema no Microsoft SQL
Server e no SAP ASE. Melhor
tratamento para especific
adores 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

AWS SCT compilação #1 .0.629 Suporte a procedimentos armazenados para conversõe s 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

AWS SCT compilação #1 .0.628

Support para emulação de serviços para conversões do DB2 SQL Server e do Oracle. Melhorias em conversões para o Amazon Redshift, incluindo mais suporte para cursores e procedimentos armazenados.

22 de junho de 2019

AWS SCT compilação #1 .0.627

Suporte para conversões do SQL Server para procedime ntos armazenados no Amazon Redshift. Melhorias da conversão para o PostgreSQL 11 e MySQL 8.0.

31 de maio de 2019

AWS SCT compilação #1 .0.626

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

AWS SCT compilação #1 .0.625

As atualizações incluem a capacidade de converter Teradata BTEQ em AWS Glue, suporte para conversõe s para MariaDB 10.3 com suporte ao modo de compatibi lidade Oracle, suporte para SAP ASE 15.7 e substituições de serviços para emular a funcionalidade ausente.

25 de março de 2019

AWS SCT compilação #1 .0.624

As atualizações incluem a capacidade de converter o Oracle ETL e o AWS Glue suporte para conversões do Microsoft SQL Server, Oracle e IBM Db2 LUW para Amazon RDS for 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

AWS SCT compilação #1 .0.623

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

AWS SCT compilação #1 .0.622

As atualizações incluem o Workload Qualification Framework, que analisa o workload para uma migração completa, incluindo modificaç ões de aplicativo e banco de dados.

20 de dezembro de 2018

AWS SCT compilação #1 .0.621	As atualizações incluem suporte ao Aurora PostgreSQ L 10 como um destino e a capacidade de migrar do Netezza usando as opções de tabela externa.	21 de novembro de 2018
AWS SCT compilação #1 .0.620	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
AWS SCT compilação #1 .0.619	As atualizações incluem suporte para a migração do Apache Cassandra ao DynamoDB e suporte para Vertica 9 como origem.	20 de setembro de 2018
AWS SCT compilação #1 .0.618	As atualizações incluem relatórios de avaliação expandidos, suporte para conversão do Oracle ROWIDs e suporte para tabelas definidas pelo usuário do SQL Server.	24 de agosto de 2018
AWS SCT compilação #1 .0.617	As atualizações incluem relatórios de avaliação expandidos, suporte para conversão do Oracle ROWIDs e suporte para tabelas definidas pelo usuário do SQL Server.	24 de julho de 2018

AWS SCT compilação #1 .0.616

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

AWS SCT compilação #1 .0.615

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

AWS SCT compilação #1 .0.614

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

AWS SCT compilação #1 .0.613

As atualizações incluem suporte para Db2 LUW, conversão de arquivos SQL*Plus e SQL Server Windows Authentication. 28 de março de 2018

AWS SCT compilação #1 .0.612	As atualizações incluem suporte para o mapeament o de tipo de dados personali zado, comparação de esquemas para Oracle 10 e conversão do Oracle em PostgreSQL de variáveis globais.	22 de fevereiro de 2018
AWS SCT compilação #1 .0.611	As atualizações incluem suporte para instruções dinâmicas do Oracle em PostgreSQL, abertura do arquivo de log seleciona ndo 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 as mudanças importantes no guia do usuário 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 requisito s de segurança do Federal Information Processing Standard (FIPS). Para obter mais informações, consulte Armazenamento de AWS credenciais.	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 requisito s de segurança do Federal Information Processing Standard (FIPS). Para obter mais informações, consulte Armazenamento de AWS credenciais.	30 de outubro de 2017
1.0.607	As tarefas de extração de dados podem ser ignoradas LOBs	Ao criar tarefas de extração de dados, agora você pode optar por ignorar objetos grandes (LOBs) para reduzir a quantidade de dados que você extrai. Para obter mais informaçõ es, consulte Criação, execução e monitoram ento 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 registro de tarefas do agente de extração de dados a partir de um link conveniente na interface AWS Schema Conversion Tool do usuário. 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	Aprimoram entos do Converter	O AWS Schema Conversion Tool mecanismo foi aprimorado para oferecer conversões aprimoradas 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	AWS SCT suporta versões adicionai s de data warehouse	Agora você pode usar o AWS Schema Conversion Tool para converter seus esquemas do Teradata 13 e do Oracle Data Warehouse 10 em esquemas equivalen tes do Amazon Redshift. Para obter mais informações, consulte Converten do esquemas de data warehouse para o Amazon RDS usando 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 Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.	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 diretamen te 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

Versão	Alteração	Descrição	Alterado em
1.0.601	AWS SCT suporta armazéns de dados 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 Convertendo esquemas de data warehouse para o Amazon RDS usando AWS SCT.	18 de abril de 2017
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 Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.	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 automatic amente 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

Versão	Alteração	Descrição	Alterado em
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 o AWS Schema Conversion Tool para registrar os agentes e criar tarefas de extração de dados para eles. Para obter mais informações, consulte Migração de dados do armazém de dados local para o Amazon Redshift com AWS Schema Conversion Tool.	16 de fevereiro de 2017
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 o AWS Schema Conversion Tool para criar AWS DMS endpoints e tarefas. Você pode executar e monitorar as tarefas do AWS SCT. Para obter mais informações, consulte Integrando com AWS Database Migration ServiceAWS Schema Conversion Tool.	20 de dezembro de 2016

Versão	Alteração	Descrição	Alterado em
1.0.502	Compatibi lidade do Amazon Aurora com PostgreSQ L 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 Convertendo esquemas de banco de dados em AWS Schema Conversion Tool.	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 <u>Gerenciando</u> perfis no AWS Schema Conversion Tool.	20 de dezembro de 2016
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 Convertendo esquemas de data warehouse para o Amazon RDS usando 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 Convertendo dados do Amazon Redshift usando AWS Schema Conversion Tool.	17 de novembro de 2016

Versão	Alteração	Descrição	Alterado em
1.0.500	Regras de mapeamento	Antes de converter seu esquema com o AWS Schema Conversion Tool, agora você pode configurar regras que alteram o tipo de dados das colunas, movem objetos de um esquema para outro e alteram os nomes dos objetos. Para obter mais informações, consulte Criação de regras de migração em 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 <u>Usando o relatório</u> de avaliação no AWS Schema Conversion Tool.	4 de outubro de 2016
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 Converten do esquemas de data warehouse para o Amazon RDS usando AWS SCT.	13 de julho de 2016

Versão	Alteração	Descrição	Alterado em
1.0.400	Conversõe s de SQL do aplicativo	Agora, você pode usar a AWS Schema Conversion Tool para converter SQL em C ++, C#, Java ou outro código de aplicativ o. Para obter mais informações, consulte Convertendo o SQL do aplicativo usando AWS SCT.	13 de julho de 2016
1.0.400	Novo recurso	O AWS Schema Conversion Tool agora contém um pacote de extensões e um assistente para ajudá-lo a instalar, criar e configurar AWS Lambda funções e bibliotec as Python para fornecer e-mail, agendamen to de trabalhos e outros recursos. Para ter mais informações, consulte <u>Usando as AWS Lambda funções do pacote AWS SCT de extensão e Como usar bibliotecas personali zadas para pacotes de extensão da AWS SCT.</u>	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 recurso	Adiciona suporte para MySQL e PostgreSQ L 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 salválos 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 para a versão AWS Schema Conversion Tool prévia.	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.