



使用者指南

AWS Schema Conversion Tool



版本 1.0.672

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

AWS Schema Conversion Tool: 使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任從何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS SCT ?	1
結構描述轉換概	5
提供反饋	5
安裝、驗證和更新	7
安裝中 AWS SCT	7
安裝以前的版本	8
驗證檔 AWS SCT 案下載	9
驗證文件的校驗和 AWS SCT	9
驗證 Fedora 上的 AWS SCT RPM 檔案	10
驗證在 U AWS SCT buntu 上的 DEB 文件	11
驗證 Microsoft 視窗上的 AWS SCT MSI 檔案	11
下載所需資料庫驅動程式	12
在 Linux 上安裝驅動程式	14
在全域設定中儲存驅動程式路徑	16
更新中 AWS SCT	17
AWS SCT CLI	17
使用使用AWS SCT者介面	18
專案視窗	18
啟動 AWS SCT	19
建立專案	20
使用新的專案精靈	21
儲存和開啟專案	24
新增伺服器	24
使用離線模式	25
使用樹狀篩選	26
.....	27
匯入樹狀結構篩選的檔案清單	29
隱藏綱要	29
管理資料庫移轉評估報告	31
轉換您的結構描	34
應用轉換後的代碼	37
儲存AWS設定檔	38
儲存AWS認證	38
設定專案的預設紀要	40

使用服AWS務設定檔的權限	41
使用 AWS Secrets Manager	41
儲存資料庫密碼	42
針對具有分區資料表的專案使用全部聯集檢視	43
鍵盤快速鍵	43
入門	45
AWS SCT 的來源	46
加密亞馬遜 RDS 連接	47
使用阿帕奇卡桑德拉作為源	49
連接到阿帕奇卡桑德拉作為源	50
使用阿帕奇 Hadoop 作為源	51
使用阿帕奇 Hadoop 作為源的先決條件	51
Hive 作為來源的權限	52
HDFS 作為來源的權限	53
HDFS 作為目標的權限	53
連接到阿帕奇 Hadoop 作為源	54
連接到蜂巢和 HDFS	55
連接到亞馬遜 EMR 作為目標	58
使用阿帕奇大西作為來源	61
先決條件	61
連接到阿帕奇奧西作為源	61
AWS Lambda 的許可	63
連接到AWS Step Functions作為目標	65
使用 Azure SQL 資料庫做為來源	66
SQL 資料庫的權限	66
連線至 Azure SQL 資料庫做為來源	67
使用適用於 z/OS 的 IBM Db2 作為來源	68
適用於 z/OS 的 Db2 的先決條件	68
適用於 z/OS 的 Db2 權限	69
連接至 Db2，用於 z/OS 作為來源	70
將 MySQL 作為目標的權限	71
將 PostgreSQL 作為目標的權限	73
適用於 z/OS 到 PostgreSQL 的 Db2 轉換設定	73
使用 IBM Db2 LUW 做為來源	75
Db2 LUW 的權限	75
連接至 Db2 LUW 做為來源	78

Db2 LUW 轉換為波斯後格雷斯	79
將 LUW 轉換為 MySQL	81
使用 MySQL 做為來源	82
適用於 MySQL 的權限	83
以 MySQL 身分驗證作為 MySQL 身分驗證	83
將 PostgreSQL 作為目標的權限	85
使用 Oracle 資料庫作為來源	86
甲骨文的權限	86
連線至 Oracle 作為來源	87
Oracle 轉換至 PostgreSQL	90
Oracle 轉換到 MySQL	95
甲骨文到亞馬遜 RDS 的甲骨文	104
使用 PostgreSQL 做為來源	110
權限	111
連線至 PostgreSQL 作為來源	111
將 MySQL 作為目標的權限	113
使用 SAP 日月光 (系統日月光) 作為來源	114
SAP 日月光的權限	115
連線至 SAP ASE 作為來源	115
將 MySQL 作為目標的權限	117
轉換 MySQL 定	118
將 PostgreSQL 作為目標的權限	119
SAP 日月 PostgreSQL 設定	119
使用 SQL Server 做為來源	120
權限：Microsoft SQL 伺服器	121
使用視窗驗證及 Microsoft SQL 伺服器	122
連線至 SQL 伺服器做為來源	124
SQL Server 轉換到 MySQL	126
SQL Server 轉換到 PostgreSQL	130
SQL Server 轉換到 Amazon RDS SQL Server	164
的資料倉儲來源 AWS SCT	165
使用亞馬遜紅移作為源	166
使用 Azure 突觸分析作為來源	170
使用 BigQuery 做為來源	174
使用綠色數據庫作為來源	178
使用尼泰扎作為來源	183

使用 Oracle 資料倉儲作為來源	191
使用雪花作為來源	196
使用 SQL 伺服器資料倉儲作為來源	204
使用太數據作為源	208
使用 Vertica 作為來源	222
建立對應規則	228
新規則	228
管理規則	229
虛擬目標	230
限制	231
建立轉換報告	232
遷移評定報告	232
建立資料庫遷移評定報告	233
檢視評定報告	234
儲存評定報告	238
設定評定報告	240
建立多伺服器評估報告	243
轉換資料庫結構	252
建立移轉規則	254
建立移轉規則	255
匯出移轉規則	256
轉換您的架構	257
轉換結構描述	257
編輯轉換的結構	259
清除轉換的結構描述	260
處理手動轉換	260
修改來源結構描述	261
修改目標結構描述	261
更新並重新整理轉換後的結構	261
儲存並套用您的結構描述	262
儲存轉換的結構描述	262
應用轉換後的模式	263
擴充套件架構	263
比較綱要	264
相關變形物件	266
將資料倉儲結構描述轉換為亞馬遜紅移	267

亞馬遜紅移的許可	268
選擇最佳化策略和規則	269
收集或上傳統計資料	270
建立移轉規則	272
建立移轉規則	272
匯出移轉規則	274
轉換您的架構	274
轉換結構描述	274
編輯轉換的結構	276
清除轉換的結構描述	277
管理和自訂金鑰	277
相關主題	278
建立和使用評估報告	278
建立資料庫移轉評估報告	279
總結	279
行動項目	281
儲存評估報告	281
處理手動轉換	282
修改來源結構描述	283
修改目標結構描述	283
更新並重新整理轉換後的結構	283
儲存並套用轉換後的結構描述	284
將轉換後的結構定義儲存為檔案	284
套用轉換的結構定義	285
擴充套件架構	285
Python 程式庫	286
優化亞馬遜紅移	286
優化您的亞馬遜紅移數據庫	286
轉換 ETL 程序	288
將 ETL 程序轉換為AWS Glue	289
先決條件	290
AWS Glue 資料目錄	291
限制	291
步驟 1：建立新的 專案	293
步驟 2：建立AWS Glue工作	293
使用 Python API 轉換 ETL 程序AWS Glue	294

步驟 1：建立資料庫	295
步驟 2：建立連線	295
步驟 3：建立AWS Glue爬行者	297
轉換信息 ETL 腳本	299
將 SSIS 轉換為AWS Glue	303
支援的 SSIS 元件	306
將 SSIS 轉換為AWS Glue Studio	308
先決條件	309
將 SSIS 套件新增至您的AWS SCT項目	311
轉換 SSIS 套件	311
創建AWS Glue Studio工作	312
建立 SSIS 轉換評估報告	313
支援的 SSIS 元件	314
將元數據 BTEQ 轉換為亞馬遜紅移 RSQL	315
將 BTEQ 指令碼新增至您的AWS SCT項目	317
在 BTEQ 指令碼中設定替代變數	317
轉換 BTEQ 指令碼	318
管理 BTEQ 指令碼	319
建立 BTEQ 指令碼轉換評估報告	319
編輯並儲存轉換後的 BTEQ 指令碼	320
將殼層指令碼轉換為亞馬遜紅移 RSQL	320
將 shell 腳本添加到您的AWS SCT項目	321
在 shell 腳本中配置替代變量	322
轉換外圍程序檔	322
管理殼層指令碼	323
建立殼層指令碼轉換評估報告	323
編輯和保存轉換後的 shell 腳本	324
轉換太拉數據FastExport到亞馬遜紅移 RSQL	324
新增FastExport工作腳本到您的AWS SCT項目	325
配置替代變量FastExport工作指令碼	326
轉換FastExport工作指令碼	327
管理FastExport工作指令碼	327
創建一個FastExport工作命令檔轉換評估報告	328
編輯並保存轉換後的FastExport工作指令碼	329
轉換太拉數據FastLoad到亞馬遜紅移 RSQL	329
新增FastLoad工作腳本到您的AWS SCT項目	330

配置替代變量FastLoad工作指令碼	330
轉換FastLoad工作指令碼	331
管理FastLoad工作指令碼	332
創建一個FastLoad工作命令檔轉換評估報告	333
編輯並保存轉換後的FastLoad工作指令碼	334
轉換太拉數據MultiLoad到亞馬遜紅移 RSQL	334
新增MultiLoad工作腳本到您的AWS SCT項目	334
配置替代變量MultiLoad工作指令碼	335
轉換MultiLoad工作指令碼	336
管理MultiLoad工作指令碼	337
創建一個MultiLoad工作命令檔轉換評估報告	337
編輯並保存轉換後的MultiLoad工作指令碼	338
遷移大數據架構	339
遷移阿帕奇 Hadoop 到亞馬遜 EMR	339
概要	339
步驟 1：連接到您的 Hadoop 叢集	340
步驟 2：設定對應規則	341
步驟 3：建立評估報告	342
第 4 步：將您的阿帕奇 Hadoop 集群遷移到亞馬遜 EMR	343
執行您的 CLI 指令碼	344
管理您的移轉專案	344
將阿帕奇奧西轉換為 AWS Step Functions	346
概要	347
步驟 1：連線至來源和目標服務	348
步驟 2：設定對應規則	349
步驟 3：設定參數	349
步驟 4：建立評估報告	351
步驟 5：將您的 Apache 工作流程轉換為 AWS Step Functions	352
執行您的 CLI 指令碼	354
支援的節點	354
搭配使用 AWS SCT 與 AWS DMS	356
使用AWS SCT複寫代理程式搭配AWS DMS	356
使用AWS SCT數據提取代理AWS DMS	356
在使用時提高日誌記錄級別AWS SCT取代為AWS DMS	356
從資料倉儲遷移到 Amazon Redshift	358
必要條件	360

Amazon S3 設置	360
假設 IAM 角色	362
安全性設定	363
組態設定	364
安裝 代理程式	364
配置代理	366
安裝和設定專用複製代理程式	367
起始代理	369
註冊代理	369
隱藏和復原 AWS SCT 代理程式的資訊	370
建立資料移轉規則	372
變更資料移轉的擷取程式和複製設定	372
資料排序	375
建立、執行和監視 AWS SCT 工作	376
匯出和匯入資料萃取工作	379
使用 AWS Snowball Edge 裝置擷取資料	380
S 使用 AWS SCT 和 AWS Snowball Edge 移轉資料的tep-by-step 程序	380
資料萃取工作輸出	383
使用虛擬分割	385
建立虛擬分割時的限制	385
範圍分割區類型	385
列出分割區類型	386
日期自動拆分區類型	387
使用原生分割	388
使用 LOB	389
最佳做法和疑難排解	390
轉換應用程式 SQL	392
轉換應用程式 SQL 概觀	392
轉換應用程式中的 SQL 程式碼	393
建立一般應用程式轉換專	393
管理應用程式轉換專	397
分析和轉換您的 SQL 程式碼	397
建立和使用評估報告	398
編輯和保存轉換後的 SQL 代碼	399
在 C# 應用程序中轉換 SQL 代碼	400
創建 C# 應用程序轉換項目	400

轉換您的 C# 應用程式 SQL 程式碼	401
儲存轉換後的應用程式碼	402
管理 C# 應用程式轉換專案	403
建立 C# 應用程式轉換評估報告	404
在 C++ 應用程式中轉換 SQL 程式碼	405
建立 C++ 應用程式轉換專案	405
轉換您的 C++ 應用程式 SQL 程式碼	406
儲存轉換後的應用程式碼	408
管理 C++ 應用程式轉換專案	408
建立 C++ 應用程式轉換評估報告	410
在 Java 應用程式中轉換 SQL 程式碼	411
建立 Java 應用程式轉換專案	411
轉換您的 Java 應用程式 SQL 程式碼	412
儲存轉換後的應用程式碼	414
管理 Java 應用程式轉換專案	414
建立 Java 應用程式轉換評估報告	415
在 Pro*C 應用程式中轉換 SQL 程式碼	416
建立 Pro*C 應用程式轉換專案	417
轉換您的 Pro*C 應用程式 SQL 程式碼	418
編輯並儲存轉換後的應用程式碼	419
管理 Pro*C 應用程式轉換專案	420
建立 Pro*C 應用程式轉換評估報告	421
使用延伸套件	423
使用擴充功能套件的權限	424
使用擴充套件結構描述	425
擴充套件的自訂程式庫	426
套用擴充功能套件	426
使用 AWS SCT 擴充套件中的 Lambda 函數	428
使用 AWS Lambda 函數來模擬數據庫功能	428
套用擴充套件以支援 Lambda 函數	428
設定擴充套件功能	430
最佳實務	432
設定其他記憶體	432
預設專案資料夾	432
提高資料遷移速度	433
增加記錄資訊	433

疑難排解	436
無法從 Oracle 來源資料庫載入物件	436
警告	436
CLI 參考	437
必要條件	437
互動式模式	437
範例	439
取得 CLI 案例	439
範例	442
編輯 CLI 案例	442
指令碼模式	443
範例	444
參考材料	444
版本備註	445
發行版本公告 — 675	445
發行公告 — 674	447
發行公告 — 673	451
發行公告 — 672	455
發行公告 — 671	463
發行版本公告 — 670	472
發行公告 — 669	476
發行公告 — 668	481
發行公告 — 667	487
發行公告 — 666	491
發行公告 — 665	495
發行版本公告 — 664	499
發行版本公告 — 663	502
發行公告 — 662	505
發行公告 — 661	510
發行公告 — 660	514
發行公告 — 659	517
發行公告 — 658	522
發行版本公告 — 657	526
發行公告 — 656	531
發行公告 — 655	534
發行版本公告 — 654	537

發行版本公告 — 653	540
發行版本公告 — 652	542
發行公告 — 651	544
發行公告 — 650	546
發行公告 — 649	548
發行公告 — 648	551
發行公告 — 647	553
發行版本公告 — 646	554
發行版本公告 — 645	556
發行公告 — 644	557
發行版本公告 — 642	559
發行版本公告 — 641	560
發行公告 — 640	561
發行版本 1.0.640 甲骨文變更	562
版本 1.0.640 Microsoft SQL 伺服器變更	567
發行版本 1.0.640 的變更	571
版本 1.0.640 PostgreSQL 更	572
版本 1.0.640 版本變更	574
版本 1.0.640 太级数据变更	575
適用於其他引擎的發行版本 1.0.640 變更	576
文件歷史紀錄	579
舊版更新	589
.....	dxcv

什麼是 AWS Schema Conversion Tool ?

您可以使用 AWS Schema Conversion Tool (AWS SCT) 將現有資料庫結構描述從一個資料庫引擎轉換到另一個資料庫引擎。您可以轉換關聯式 OLTP 結構描述，或資料倉儲結構描述。轉換後的結構描述適用於 Amazon Relational Database Service (Amazon RDS) MySQL、MariaDB、Oracle、SQL Server、PostgreSQL 資料庫、Amazon Aurora 資料庫叢集或 Amazon Redshift 叢集。轉換後的結構描述也可用於 Amazon EC2 執行個體的資料庫，或存放在 Amazon S3 儲存貯體的資料。

AWS SCT 支援多種業界標準，包括聯邦資訊處理標準 (FIPS)，以連接 Amazon S3 儲存貯體或其他 AWS 資源。AWS SCT 同時也符合聯邦風險與授權管理計畫 (FedRAMP) 規範。如需 AWS 和合規性工作的詳細資訊，請參閱[合規計劃的 AWS 服務範圍](#)。

AWS SCT 支援以下 OLTP 轉換。

來源資料庫	目標資料庫
適用於 z/OS 的 IBM Db2 (版本 12)	亞馬遜極光 MySQL 兼容版 (極光 MySQL) ， 亞馬遜極光兼容版 (極光版) 如需詳細資訊，請參閱 使用適用於 z/OS 的 IBM Db2 作為來源 。
IBM Db2 LUW (版本 9.1、9.5、9.5、10.5、11.1 及 11.5 版本)	極光 MySQL, 極光, 瑪麗亞德, MySQL, PostgreSQL 如需詳細資訊，請參閱 使用 IBM Db2 LUW 做為來源 。
微軟 SQL 數據庫	極光 MySQL, 極光, MySQL 如需詳細資訊，請參閱 使用 Azure SQL 資料庫做為來源 。
微軟 SQL 伺服器 (版本二、二零一四年、二零一七年、二零一九年及二〇二〇)	極光 MySQL, 極光 PostgreSQL, 極光的巴貝魚 (僅用於評估報告), 微軟 SQL 服務器, MySQL, PostgreSQL 如需詳細資訊，請參閱 使用 SQL Server 做為來源 。

來源資料庫	目標資料庫
版本 5.5 及更高版本	<p>極光, MySQL,</p> <p>如需詳細資訊, 請參閱使用 MySQL 做為來源。</p> <p>您可以將結構描述和資料從 MySQL 遷移到極光 MySQL 資料庫叢集, 而無需使用 AWS SCT。如需詳細資訊, 請參閱將資料移轉至 Amazon Aurora 資料庫叢集。</p>
甲骨文 (版本 10.1 及更高版本)	<p>極光 MySQL, 極光, 瑪麗亞德, MySQL, 甲骨文, PostgreSQL</p> <p>如需詳細資訊, 請參閱使用 Oracle 資料庫作為來源。</p>
版本 9.1 及更高版本	<p>極光 MySQL, 極光, MySQL</p> <p>如需詳細資訊, 請參閱使用 PostgreSQL 做為來源。</p>
日月光 (版本 12.5.4、15.0.2、15.5、15.7 及 16.0 版本)	<p>極光 MySQL, 極光, 瑪麗亞德, MySQL, PostgreSQL</p> <p>如需詳細資訊, 請參閱使用 SAP 日月光 (系統日月光) 作為來源。</p>

AWS SCT 支援以下資料倉儲轉換。

來源資料倉儲	目標資料倉儲
Amazon Redshift	<p>Amazon Redshift</p> <p>如需詳細資訊, 請參閱使用亞馬遜紅移作為源。</p>
突觸分析	Amazon Redshift

來源資料倉儲	目標資料倉儲
	如需詳細資訊，請參閱 使用 Azure 突觸分析作為來源 。
BigQuery	Amazon Redshift 如需詳細資訊，請參閱 使用 BigQuery 做為來源 。
綠梅數據庫 (版本 4.3 和 6.21)	Amazon Redshift 如需詳細資訊，請參閱 使用綠色數據庫作為來源 。
微軟 SQL 伺服器 (2008 年及更高版本)	Amazon Redshift 如需詳細資訊，請參閱 使用 SQL 伺服器資料倉儲作為來源 。
尼泰扎 (版本 7.0.3 及更高版本)	Amazon Redshift 如需詳細資訊，請參閱 使用尼泰扎作為來源 。
甲骨文 (版本 10.1 及更高版本)	Amazon Redshift 如需詳細資訊，請參閱 使用 Oracle 資料倉儲作為來源 。
雪花 (版本 3)	Amazon Redshift 如需詳細資訊，請參閱 使用雪花作為來源 。
太數據 (版本 13 及更高版本)	Amazon Redshift 如需詳細資訊，請參閱 使用太數據作為源 。
韋爾蒂卡 (版本 7.2.2 及更高版本)	Amazon Redshift 如需詳細資訊，請參閱 使用 Vertica 作為來源 。

AWS SCT支持以下數據 NoSQL 數據庫轉換。

來源資料庫	目標資料庫
阿帕奇卡桑德拉 (2.1.x 版本 , 2.2.16 和 3.11.x 版本)	Amazon DynamoDB 如需詳細資訊，請參閱 使用阿帕奇卡桑德拉作為源 。

AWS SCT支援以下擷取、轉換和載入 (ETL) 程序的轉換。如需詳細資訊，請參閱[轉換 ETL 程序](#)。

來源	Target
ETL 信息腳本	信息
微軟 SQL 服務器集成服務 (SSIS) ETL 軟件包	AWS Glue 或 AWS Glue Studio
包含來自 Teradata 基本資料查詢 (BTEQ) 內嵌指令的殼層指令碼	亞馬遜紅移 RSQL
大數據 BTEQ ETL 腳本	AWS Glue或亞馬遜紅移 RSQL
FastExport工作腳本	亞馬遜紅移 RSQL
FastLoad工作腳本	亞馬遜紅移 RSQL
MultiLoad工作腳本	亞馬遜紅移 RSQL

AWS SCT支持以下大數據框架遷移。如需詳細資訊，請參閱[遷移大數據架構](#)。

來源	Target
阿帕奇蜂巢 (版本 0.13.0 及更高版本)	蜂巢在亞馬遜 EMR
阿帕奇	亞馬遜 S3 或亞馬遜 EMR 上的 HDFS
Apache Oozie	AWS Step Functions

結構描述轉換概

AWS SCT提供以專案為基礎的使用者介面，可自動將來源資料庫的資料庫結構描述轉換為與目標 Amazon RDS 執行個體相容的格式。如果無法自動轉換來源資料庫中的結構描述，請AWS SCT提供有關如何在目標 Amazon RDS 資料庫中建立對等結構描述的指導。

如需如何安裝 AWS SCT 的相關資訊，請參閱[安裝、驗證和更新 AWS SCT](#)。

如需 AWS SCT 使用者界面的簡介，請參閱[使用使用AWS SCT者介面](#)。

如需轉換程序的詳細資訊，請參閱[轉換資料庫結構描述 AWS SCT](#)。

除了將現有的資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎之外，還AWS SCT有一些其他功能可協助您將資料和應用程式移至AWS雲端：

- 您可以使用資料擷取代理程式從資料倉儲擷取資料，以準備將資料遷移到 Amazon Redshift。若要管理資料擷取代理程式，您可以使用 AWS SCT。如需詳細資訊，請參閱[將資料從現場部署資料倉儲遷移到 Amazon Redshift](#)。
- 您可以使用 AWS SCT 來建立 AWS DMS 端點和任務。您可以從 AWS SCT 執行和監控這些任務。如需詳細資訊，請參閱[搭配使用 AWS SCT 與 AWS DMS](#)。
- 在某些情況下，資料庫功能無法轉換為相等的 Amazon RDS 或亞馬遜紅移功能。AWS SCT 延伸套件精靈可以協助您安裝 AWS Lambda 函數和 Python 程式庫，以模擬無法轉換的功能。如需詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。
- 您可以使用AWS SCT來優化您現有的亞馬遜紅移資料庫。AWS SCT建議排序索引鍵和分配金鑰以最佳化您的資料庫。如需詳細資訊，請參閱[通過使用優化亞馬遜紅移 AWS SCT](#)。
- 您可以使用AWS SCT將現有的現場部署資料庫結構描述複製到執行相同引擎的 Amazon RDS 資料庫執行個體。您可以使用此功能分析移到雲端和變更授權類型的潛在成本節省。
- 您可以使用 AWS SCT 來轉換 C++、C#、Java 或其他應用程式碼中的 SQL。您可以檢視、分析、編輯和儲存轉換後的 SQL 程式碼。如需詳細資訊，請參閱[使用轉換應用程式 SQLAWS SCT](#)。
- 您可以使用AWS SCT來移轉擷取、轉換和載入 (ETL) 程序。如需詳細資訊，請參閱[使用以下方式轉換擷取、轉換和載入 \(ETL\) 程序AWS Schema Conversion Tool](#)。

提供意見

您可以提供有關AWS SCT. 您可以提交錯誤報告，提交功能請求或提供一般信息。

提供有關的反饋 AWS SCT

1. 啟動 AWS Schema Conversion Tool。
2. 開啟 Help (說明) 功能表，然後選擇 Leave Feedback (留下意見回饋)。出現 Leave Feedback (留下意見回饋) 對話方塊。
3. 對於 Area (領域)，選擇 Information (資訊)、Bug report (錯誤報告) 或 Feature request (功能請求)。
4. 對於 Source database (來源資料庫)，選擇您的來源資料庫。如果您的意見回饋不是針對特定資料庫，請選擇 Any (任何)。
5. 對於 Target database (目標資料庫)，選擇您的目標資料庫。如果您的意見回饋不是針對特定資料庫，請選擇 Any (任何)。
6. 對於 Title (標題)，輸入您意見回饋的標題。
7. 對於 Message (訊息)，輸入您的意見回饋。
8. 選擇 Send (傳送) 提交您的意見回饋。

安裝、驗證和更新 AWS SCT

AWS Schema Conversion Tool (AWS SCT) 是提供以專案為基礎的使用者介面的獨立應用程式。AWS SCT 可用於 Microsoft 視窗、軟軟呢帽和 Linux 系統。AWS SCT 僅在 64 位元作業系統上受支援。

為確保您取得正確版本的 AWS SCT 散發檔案，我們會在您下載壓縮檔案後提供驗證步驟。您可以使用我們提供的步驟來驗證檔案。

主題

- [安裝中 AWS SCT](#)
- [驗證檔 AWS SCT 案下載](#)
- [下載所需資料庫驅動程式](#)
- [更新中 AWS SCT](#)
- [AWS SCT CLI](#)

安裝中 AWS SCT

您可以 AWS SCT 在下列作業系統上安裝：

- Microsoft Windows 10
- 軟呢帽 36 及更高版本
- 版本 18 及更高版本

若要安裝 AWS SCT

1. 使用您作業系統的連結，下載包含 AWS SCT 安裝程式的壓縮檔案。所有壓縮檔案均使用 .zip 副檔名。當您解壓縮 AWS SCT 安裝程式檔案時，它會採用適合您作業系統的格式。
 - [Microsoft Windows](#)
 - [Ubuntu Linux \(.deb\)](#)
 - [Fedora Linux \(.rpm\)](#)
2. 為您的作業系統解壓縮 AWS SCT 安裝程式檔案，如下所示。

作業系統	檔案名稱
Fedora Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm
Microsoft Windows	AWS Schema Conversion Tool-1.0. <i>build-number</i> .msi
Ubuntu Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .deb

3. 執行在上一個步驟中解壓縮的 AWS SCT 安裝程式檔案。使用適用於您作業系統的指示，如下所示。

作業系統	安裝說明
Fedora Linux	<p>在下載檔案的目標資料夾中，執行下列命令：</p> <pre>sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm</pre>
Microsoft Windows	按兩下檔案來執行安裝程式。
Ubuntu Linux	<p>在下載檔案的目標資料夾中，執行下列命令：</p> <pre>sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i> .deb</pre>

4. 下載來源和目標資料庫引擎的 Java 資料庫連線 (JDBC) 驅動程式。如需說明和下載連結，請參閱[下載所需資料庫驅動程式](#)。

現在，您已經完成了 AWS SCT 應用程序的設置。雙擊應用程序圖標以運行 AWS SCT。

安裝先前版本的 AWS SCT

您可以從 1.0.625 AWS SCT 開始下載並安裝先前版本。若要下載舊版，請使用下列格式提供版本和作業系統資訊。

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

例如，若要下載 AWS SCT 版本 668，請執行下列動作：

- 視窗 [aws-schema-conversion-tool](https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Windows/) — <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Windows/>
- UBUNTU 的 — <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Ubuntu/>. 拉 [aws-schema-conversion-tool](https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Ubuntu/) 鍊
- 軟呢帽 — <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Fedora/> [aws-schema-conversion-tool](https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Fedora/).. 拉鍊

AWS 支持該 AWS SCT 應用程序的三個最新版本。我們建議您使用最新版本的 AWS SCT。

驗證檔 AWS SCT 案下載

有幾種方法可以驗證的分發檔案 AWS SCT。最簡單的方法是將檔案的總和檢查碼與中發佈的總和檢查碼進行比較。AWS 做為額外安全層級，您可以根據您安裝檔案的作業系統，使用下列程序來驗證分發檔案。

本節包含下列主題。

主題

- [驗證文件的校驗和 AWS SCT](#)
- [驗證 Fedora 上的 AWS SCT RPM 檔案](#)
- [驗證在 U AWS SCT buntu 上的 DEB 文件](#)
- [驗證 Microsoft 視窗上的 AWS SCT MSI 檔案](#)

驗證文件的校驗和 AWS SCT

為了偵測在下載或儲存 AWS SCT 壓縮檔案時可能引入的任何錯誤，您可以將檔案總和檢查碼與提供的值進行比較。AWS 使用 SHA256 演算法來執行總和檢查碼。

若要使用總和檢查碼來驗證 AWS SCT 散發檔

1. 使用 [安裝] 區段中的連結下載 AWS SCT 散發檔案。如需詳細資訊，請參閱 [安裝中 AWS SCT](#)。
2. 下載最新的檢查總和檔案，稱為 [sha256Check.txt](#)。此檔案包含最新版 AWS SCT 本的總和檢查碼。例如，檔案可顯示如下：

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

3. 在包含分發檔案的目錄中，執行適用於您作業系統的 SHA256 驗證命令。例如，在 Linux 中執行下列命令。

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

4. 比較命令的結果和 sha256Check.txt 檔案中顯示的值。如果校驗和匹配，那麼運行分發文件是安全的。如果總和檢查碼不相符，請勿執行分發檔案，並[聯絡 AWS Support](#)。

驗證 Fedora 上的 AWS SCT RPM 檔案

AWS 除了發佈檔案總和檢查碼之外，還提供另一個層級的驗證。發佈檔案中的所有 RPM 檔案均由 AWS 私密金鑰簽署。您可以在 [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key) 檢視公開 GPG 金鑰。

驗證 Fedora 上的 AWS SCT RPM 檔案

1. 使用 [安裝] 區段中的連結下載 AWS SCT 散發檔案。
2. 驗證分 AWS SCT 發檔案的總和檢查碼。
3. 擷取分發檔案的內容。找到您要驗證的 RPM 檔案。
4. 從 [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key) 下載 GPG 公開金鑰
5. 使用下列命令，將公開金鑰匯入至您的 RPM 資料庫 (確保您有適當的許可)：

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

6. 執行下列命令，檢查是否成功匯入：

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

7. 執行下列命令，檢查 RPM 簽章：

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

驗證在 U AWS SCT buntu 上的 DEB 文件

AWS 除了發佈檔案總和檢查碼之外，還提供另一個層級的驗證。分發檔案中的所有 DEB 檔案都由 GPG 分離簽章簽署。

若要在 Ubuntu 上驗證 AWS SCT DEB 檔案

1. 使用 [安裝] 區段中的連結下載 AWS SCT 散發檔案。
2. 驗證分 AWS SCT 發文件的校驗和。
3. 擷取分發檔案的內容。找到您要驗證的 DEB 檔案。
4. 從 [aws-schema-conversion-tool-1.0.1](#) 下載分離的簽名最新的 .deb.asc。
5. 從 [amazon.com.public.gpg-key](#) 下載 GPG 公開金鑰。
6. 執行下列命令，匯入 GPG 公開金鑰：

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

7. 執行下列命令，驗證簽章：

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

驗證 Microsoft 視窗上的 AWS SCT MSI 檔案

AWS 除了發佈檔案總和檢查碼之外，還提供另一個層級的驗證。MSI 文件具有數字簽名，您可以檢查以確保它是由簽名 AWS。

在視窗上驗證 AWS SCT MSI 檔案

1. 使用 [安裝] 區段中的連結下載 AWS SCT 散發檔案。
2. 驗證分 AWS SCT 發文件的校驗和。
3. 擷取分發檔案的內容。找到您要驗證的 MSI 檔案。
4. 在 Windows 檔案總管的 MSI 檔案上，按一下滑鼠右鍵，然後選取 內容。
5. 選擇 數位簽章 索引標籤。
6. 確認數位簽章是來自 Amazon Services LLC.。

下載所需資料庫驅動程式

若 AWS SCT 要正確運作，請下載來源和目標資料庫引擎的 JDBC 驅動程式。如果您使用虛擬目標資料庫平台，則不需要下載目標資料庫引擎的 JDBC 驅動程式。如需詳細資訊，請參閱 [使用虛擬目標](#)。

下載驅動程式之後，您提供驅動程式檔案的位置。如需詳細資訊，請參閱 [在全域設定中儲存驅動程式路徑](#)。

您可以從下列位置下載資料庫驅動程式。

Important

下載最新版本的驅動程式。下表包含支援的最低版本的資料庫驅動程式 AWS SCT。

資料庫引擎	驅動程式	下載位置
Amazon Aurora MySQL-Compatible Edition	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/connector/
Amazon Aurora PostgreSQL-Compatible Edition	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
Amazon EMR	HiveJDBC42.jar	http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/
Amazon Redshift	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift Serverless	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip

資料庫引擎	驅動程式	下載位置
Apache Hive	hive-jdbc-2.3.4-standalone.jar	https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar
天青 SQL 數據庫	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/ 驅 release-notes-for-the 動程序? 查看次數 = 1 sql-server-ver #72
突觸分析	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/ 驅 release-notes-for-the 動程序? 查看次數 = 1 sql-server-ver #72
Greenplum 資料庫	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
適用於 z/OS 的 IBM Db2	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and 下載-DB2-佐斯
IBM Db2 LUW	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/node/382667
MariaDB	mariadb-java-client-2.4.1.jar	https://downloads.mariadb.com/Connectors/java/connector-java-2.4.1/ 罐 mariadb-java-client 裝
Microsoft SQL Server	mssql-jdbc-10.2.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/ download-microsoft-jdbc-driver-for-sql-server? 視圖 = 15 sql-server-ver
MySQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/connector/j/
Netezza	nzjdbc.jar 使用用戶端工具軟體。下載驅動程序版本 7.2.1，與數據倉庫版本 7.2.0 向後兼容。	http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html

資料庫引擎	驅動程式	下載位置
Oracle	ojdbc8.jar 支援 8 及更新版本的驅動程式。	https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html
PostgreSQL	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
SAP ASE (Sybase ASE)	jconn4.jar	https://dcx.sap.com/sq1a170/en/html/3bd588c76c5f10149824867aa281fadd.html
Snowflake	snowflake-jdbc-3.9.2.jar 如需詳細資訊，請參閱 下載/整合 JDBC 驅動程式 。	https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar
Teradata	terajdbc4.jar tdgssconfig.jar 對於元數據 JDBC 驅動程序 16.20.00.11 及更高版本，您不需要該文件。tdgssconfig.jar	https://downloads.teradata.com/download/connectivity/jdbc-driver
Vertica	vertica-jdbc-9.1.1-0.jar 支援 7.2.0 及更新版本的驅動程式。	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

在 Linux 上安裝驅動程式

您可以使用下列步驟在 Linux 系統上安裝 JDBC 驅動程式，以便搭配 AWS SCT 使用。

若要在 Linux 系統上安裝 JDBC 驅動程式

1. 建立目錄以存放 JDBC 驅動程式。

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

2. 使用以下所示的命令，安裝您資料庫引擎適用的 JDBC 驅動程式。

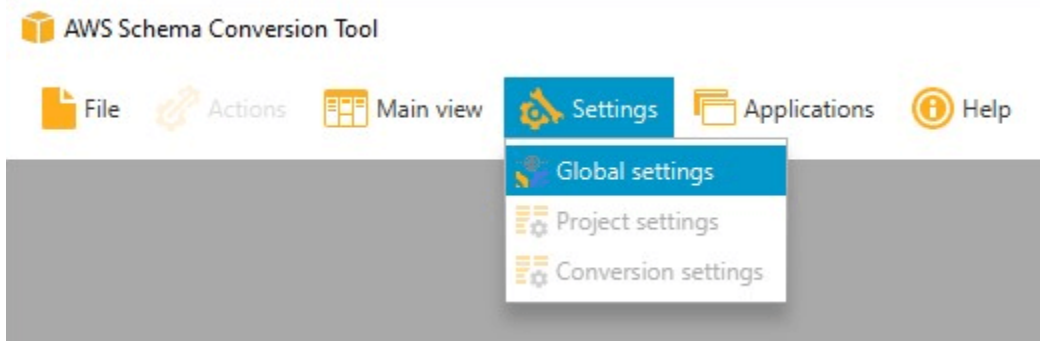
資料庫引擎	安裝指令
Amazon Aurora (相容於 MySQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Amazon Aurora (相容於 PostgreSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>
Microsoft SQL Server	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
MySQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Oracle	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo mkdir oracle-jdbc PROMPT> cd oracle-jdbc PROMPT> sudo cp -a /tmp/ojdbc8.jar .</pre>
PostgreSQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

在全域設定中儲存驅動程式路徑

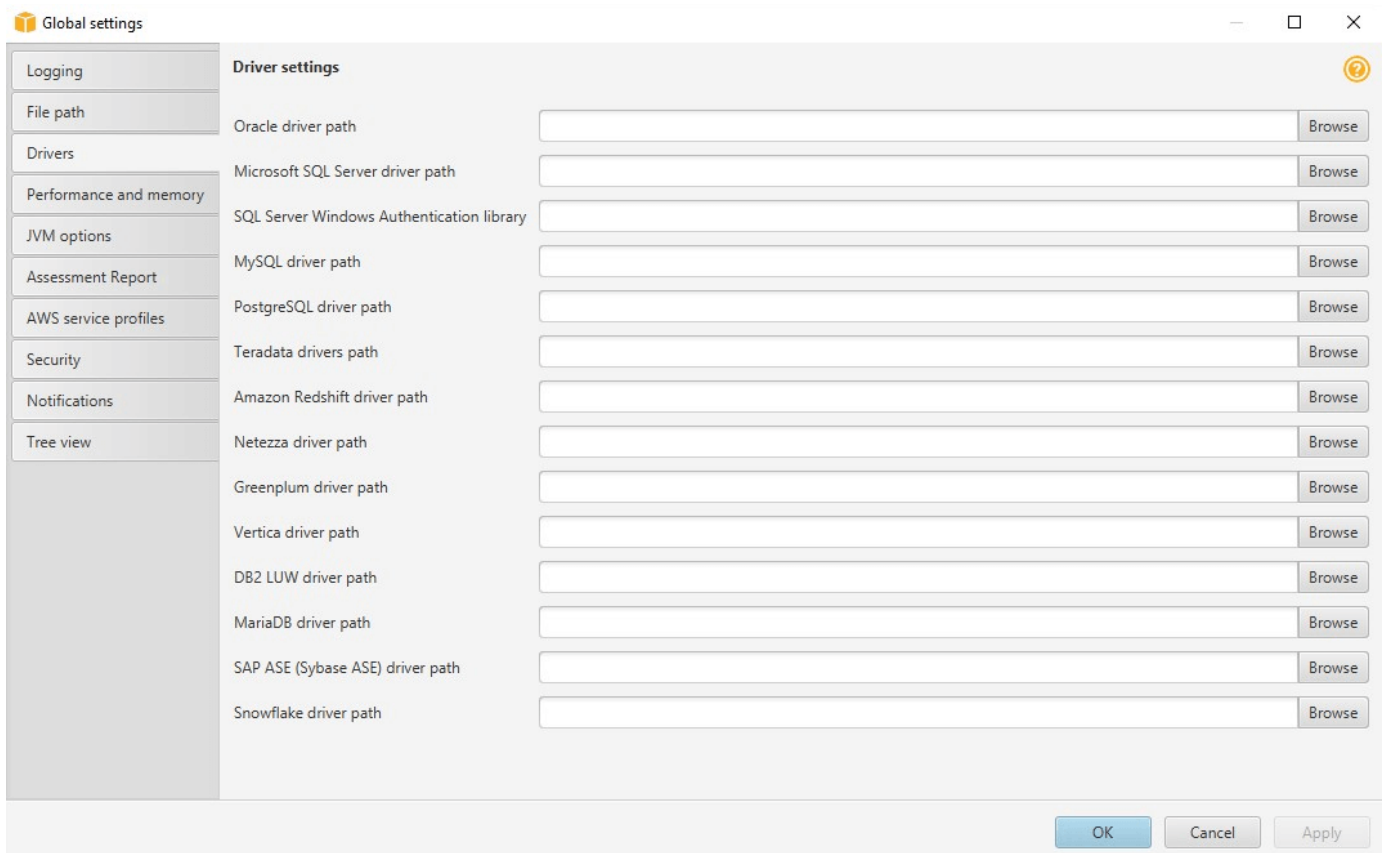
下載並安裝所需的 JDBC 驅動程式後，您可以在 AWS SCT 設定中全域設定驅動程式的位置。如果您不設定驅動程式的全域位置，應用程式會在您連接到資料庫時向您詢問驅動程式的位置。

若要更新驅動程式檔案位置

1. 在中 AWS SCT，選擇 [設定]，然後選擇 [全域設定]。



2. 對於 Global settings (全域設定)，選擇 Drivers (驅動程式)。將檔案路徑新增至來源資料庫引擎和目標 Amazon RDS 資料庫執行個體資料庫引擎的 JDBC 驅動程式。



3. 完成新增驅動程式路徑之後，選擇 OK (確定)。

更新中 AWS SCT

AWS 定期更新新 AWS SCT 的特性和功能。如果您要從舊版更新，請建立新 AWS SCT 專案並重新轉換您正在使用的任何資料庫物件。

您可以檢查是否存在的更新 AWS SCT。

若要檢查是否有更新 AWS SCT

1. 進入時 AWS SCT，請選擇「說明」，然後選擇「檢查更新」。
2. 在 Check for Updates (檢查更新) 對話方塊中，選擇 What's New (最新消息)。如果沒有出現連結，表示您擁有最新的版本。

AWS SCT CLI

您可以下載 AWS SCT CLI 以供命令列使用。若要下載 JAR，請使用下列連結：

[AWSSchemaConversionToolBatch. 罐子](#)

使用使用AWS SCT者介面

請使用下列主題來協助您使用使用AWS SCT者介面。如需安裝的資訊AWS SCT，請參閱[安裝、驗證和更新 AWS SCT](#)。

主題

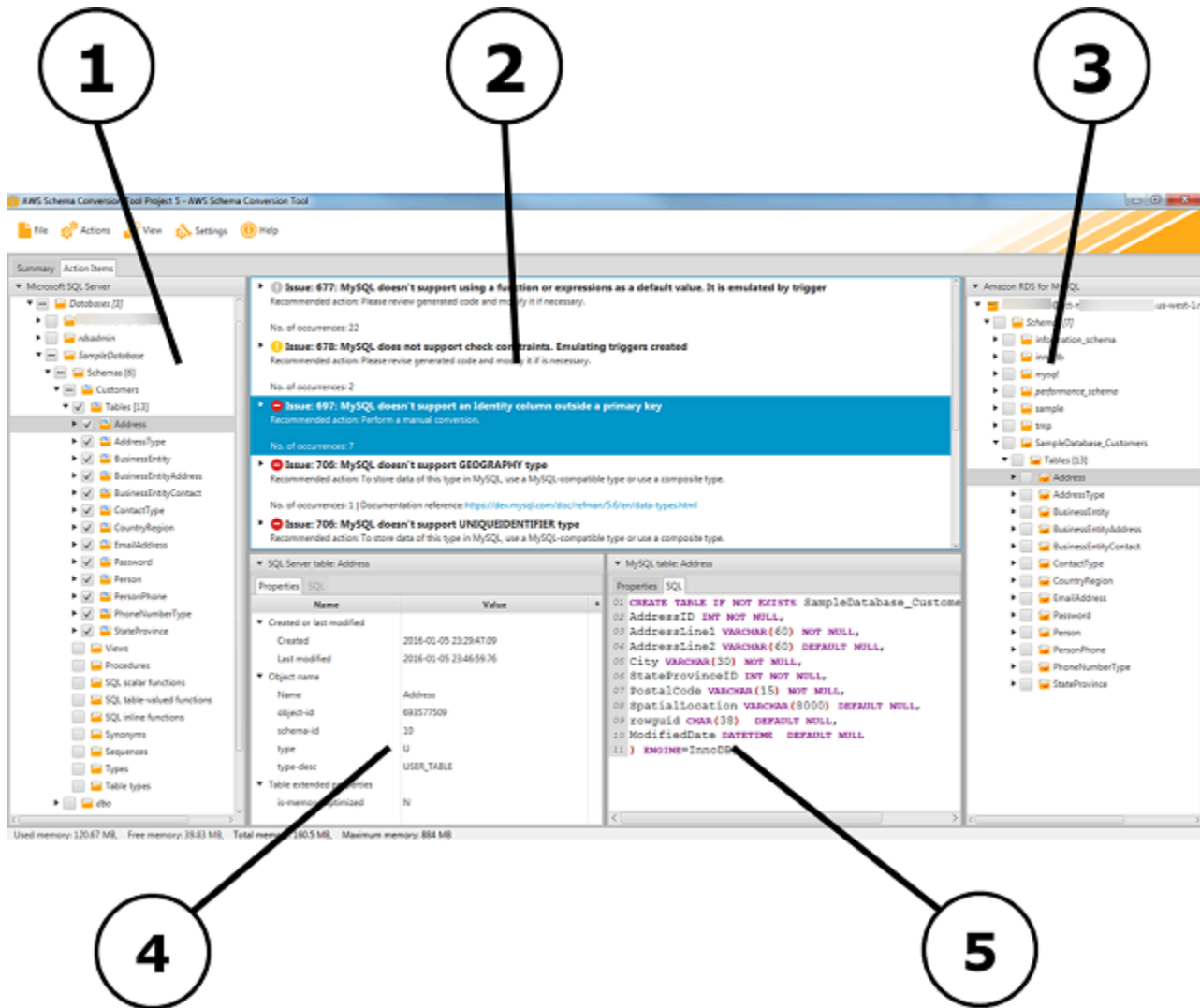
- [AWS SCT專案視窗](#)
- [啟動 AWS SCT](#)
- [建立AWS SCT專案](#)
- [在中使用新專案精靈AWS SCT](#)
- [儲存和開啟專AWS SCT案](#)
- [將資料庫伺服器新增至AWS SCT專案](#)
- [以離線模式執行AWS SCT](#)
- [使用AWS SCT樹狀篩選器](#)
- [在AWS SCT樹狀檢視中隱藏資料架構](#)
- [建立及檢閱資料庫移轉評估報告](#)
- [轉換您的結構描](#)
- [將轉換後的結構描述套用至您的目標資料庫](#)
- [將AWS服務設定檔儲存在AWS SCT](#)
- [使用 AWS Secrets Manager](#)
- [儲存資料庫密碼](#)
- [針對具有分區資料表的專案使用 UNION ALL 檢視](#)
- [鍵盤快速鍵AWS SCT](#)

AWS SCT專案視窗

下圖是您在建立資料架構移轉專案，然後轉換資料架構AWS SCT時所看到的內容。

1. 在左側面板中，來自您來源資料庫的結構描述會呈現在樹狀檢視中。您的資料庫結構描述為「延遲載入」。換言之，當您從樹狀檢視選擇項目，AWS SCT 會從您的來源資料庫取得並顯示目前的結構描述。
2. 在上方中間面板，對於無法自動轉換到目標資料庫引擎的來源資料庫引擎結構描述元素，會顯示動作項目。

- 在右側面板中，來自您目標資料庫執行個體的結構描述會呈現在樹狀檢視中。您的資料庫結構描述為「延遲載入」。也就是說，當您從樹狀檢視選擇項目，AWS SCT 會從您的目標資料庫取得並顯示目前的結構描述。



- 在左下方面板中，當您選擇架構元素時，會顯示屬性。這些描述了源模式元素和 SQL 命令，以便在源數據庫中創建該元素。
- 在右下方面板中，當您選擇架構元素時，會顯示屬性。這些描述了在目標數據庫中創建該元素的目標模式元素和 SQL 命令。您可以編輯此 SQL 命令，並將更新的命令儲存至您的專案。

啟動 AWS SCT

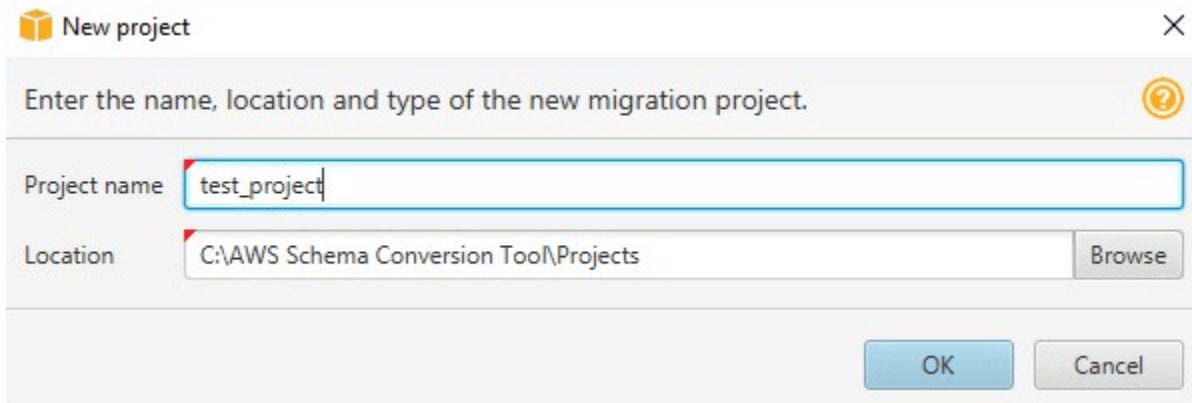
若要啟動 AWS Schema Conversion Tool，請按兩下應用程式圖示。

建立AWS SCT專案

請使用下列程序來建立AWS Schema Conversion Tool專案。

若要建立專案

1. 啟動 AWS Schema Conversion Tool。
2. 在 [檔案] 功能表上，選擇 [新增專案]。[新增專案] 對話方塊隨即出現。



3. 輸入專案的名稱，這會儲存在本機電腦上。
4. 輸入本機專案檔案的位置。
5. 選擇 OK (確定)，建立 AWS SCT 專案。
6. 選擇 [新增來源]，將新的來源資料庫新增至您的AWS SCT專案。您可以將多個源數據庫添加到AWS SCT項目中。
7. 選擇添加目標以在項目中添加新的AWS SCT目標平台。您可以將多個目標平台添加到您的AWS SCT項目中。
8. 在左側面板中選擇源數據庫模式。
9. 在右側面板中，指定所選來源結構描述的目標資料庫平台。
10. 選擇 [建立對應]。在您選擇來源資料庫綱要和目標資料庫平台之後，此按鈕會變成作用中狀態。如需詳細資訊，請參閱[建立對應規則](#)。

現在，您的AWS SCT項目已設置。您可以儲存專案、建立資料庫移轉評估報告，以及轉換來源資料庫結構描述。

在中使用新專案精靈AWS SCT

您可以使用新專案精靈建立新資料庫移轉專案。此精靈會協助您判斷移轉目標並連線至資料庫。它會估計移轉對於所有支援的目標目的地而言可能有多複雜。執行精靈之後，AWS SCT會產生將資料庫移轉至不同目標目的地的摘要報告。您可以使用此報告來比較可能的目標目的地，並選擇最佳移轉路徑。

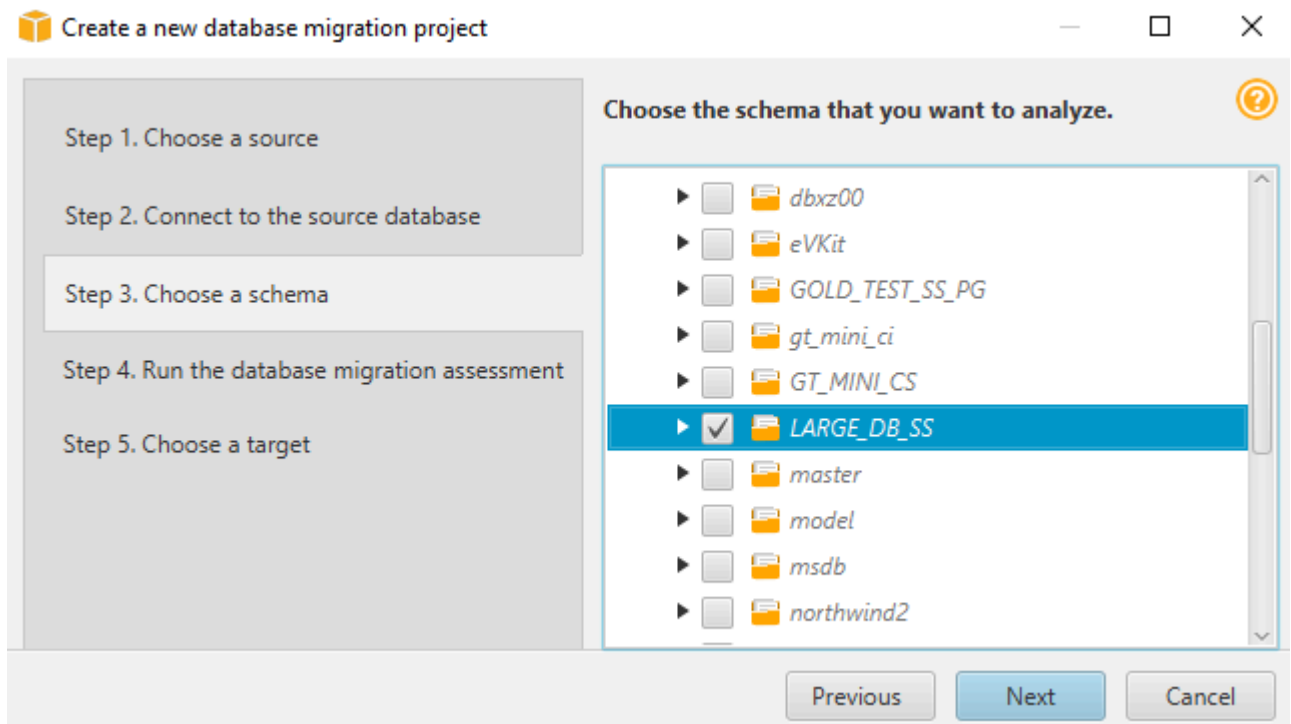
執行新增專案精靈

1. 選擇您的來源資料庫。
 - a. 啟動 AWS Schema Conversion Tool。
 - b. 在 [檔案] 功能表上，選擇 [新增專案精靈]。將開啟 [建立新資料庫移轉專案] 對話方塊。
 - c. 若要輸入來源資料庫連線資訊，請使用下列指示：

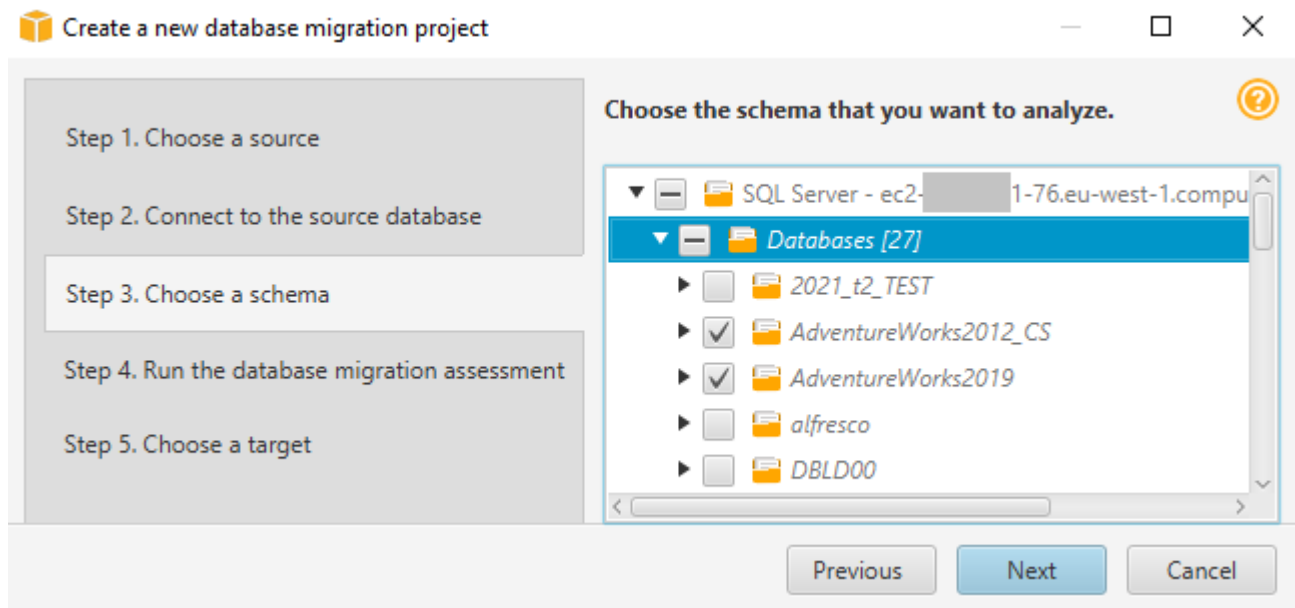
參數	動作
Project name (專案名稱)	輸入專案的名稱，這會儲存在本機電腦上。
位置	輸入本機專案檔案的位置。
Source type (來源類型)	選擇下列其中一個選項：SQL 資料庫、NoSQL 資料庫或 ETL。 如果您想要查看包含所有移轉目的地的摘要報告，請選擇 [SQL 資料庫]。
Source engine (來源引擎)	選擇您的來源資料庫引擎。
遷移策略	請選擇下列其中一個選項： <ul style="list-style-type: none">• 我想切換引擎並針對雲進行優化 — 此選項將您的源數據庫轉換為新的數據庫引擎。• 我想保持相同的引擎，但對雲進行優化 — 此選項使您的數據庫引擎保持原樣，並將數據庫從內部部署移動到雲端。• 我想要查看資料庫引擎切換和雲端最佳化的合併報告 — 此選項會比較所有可用遷移選項的移轉複雜度。

參數	動作
	如果您想要查看包含所有移轉目的地的彙總評估報告，請選擇最後一個選項。

- d. 選擇 下一步。[Connect 至來源資料庫] 頁面隨即開啟。
2. 連接到您的來源資料庫。
 - a. 提供來源資料庫的連線資訊。連線參數取決於您的來源資料庫引擎。請確定您用於分析來源資料庫的使用者具有適用的權限。如需詳細資訊，請參閱[AWS SCT 的來源](#)。
 - b. 選擇 Next (下一步)。隨即開啟 [選擇結構描述] 頁面。
 3. 選擇資料庫結構描述。
 - a. 選取您要評估的綱要名稱的核取方塊，然後選擇 Scheme (本身)。選取時，綱要名稱會以藍色反白顯示，且「下一步」按鈕可用。



- b. 如果您要評估數個資料庫綱要，請選取所有綱要的核取方塊，然後選擇父項節點。若要成功評估，您必須選擇父節點。例如，對於來源 SQL Server 資料庫，請選擇 [資料庫] 節點。父節點的名稱以藍色反白顯示，並且可以使用「下一步」(Next) 按鈕。



- c. 選擇下一步。AWS SCT會分析來源資料庫結構描述，並建立資料庫移轉評估報告。來源資料庫綱要中的資料庫物件數目會影響執行評估所需的時間。完成時，會開啟 [執行資料庫移轉評估] 頁面。
4. 執行資料庫移轉評估。
 - a. 您可以檢閱和比較不同移轉目標的評估報告，或儲存評估報告檔案的本機副本以供進一步分析。
 - b. 儲存資料庫移轉評估報告的本機副本。選擇「儲存」，然後輸入要儲存檔案的資料夾路徑，然後選擇「儲存」。AWS SCT將評估報告檔案儲存到指定的資料夾。
 - c. 選擇 下一步。隨即開啟 [選擇目標] 頁面。
5. 選擇您的目標資料庫。
 - a. 針對 Target 引擎，請根據評估報告選擇您決定使用的目標資料庫引擎。
 - b. 提供目標資料庫的連線資訊。您看到的連線參數取決於您選取的目標資料庫引擎。請確定為目標資料庫指定的使用者具有必要的權限。如需必要權限的詳細資訊，請參閱[AWS SCT 的來源](#)和中描述目標資料庫權限的章節[亞馬遜紅移作為目標的許可](#)。
 - c. 選擇「完成」。AWS SCT會建立您的專案並加入對映規則。如需詳細資訊，請參閱[建立對應規則](#)。

現在您可以使用AWS SCT專案來轉換來源資料庫物件。

儲存和開啟專案AWS SCT專案

請使用下列步驟來儲存AWS Schema Conversion Tool專案。

儲存您的專案

1. 啟動 AWS Schema Conversion Tool。
2. 在 [檔案] 功能表上，選擇 [儲存專案]。

AWS SCT將專案儲存在建立專案時指定的資料夾中。

使用下列步驟開啟現有的AWS Schema Conversion Tool專案。

開啟您的專案

1. 在 [檔案] 功能表上，選擇 [開啟專案]。開啟」對話方塊隨即出現。
2. 選擇專案資料夾，然後選擇 Windows 指令碼元件 (*.sct) 檔案。
3. AWS SCT開啟您的專案，但不會自動連線至您的來源和目標資料庫。選擇資料庫結構描述樹狀目錄頂端的 Connect 線至伺服器，以連線至您的來源和目標資料庫。

如果您開啟以 1.0.655 版或更新AWS SCT版本儲存的專案，AWS SCT會自動為所有來源資料庫結構描述建立對應規則至目標資料庫平台。若要新增其他目標資料庫平台，請刪除現有的對應規則，然後建立新的對應規則。如需建立對應規則的詳細資訊，請參閱[建立對應規則](#)。

將資料庫伺服器新增至AWS SCT專案

您可以將多個來源和目標資料庫伺服器新增至AWS Schema Conversion Tool專案。

若要將伺服器新增至專案

1. 啟動 AWS Schema Conversion Tool。
2. 建立新專案或開啟現有專案。
3. 從功能表中選擇「新增來源」，以新增來源資料庫。
4. 選擇資料庫平台並指定資料庫連線證明資料。如需連接至來源資料庫的詳細資訊，請參閱[AWS SCT 的來源](#)。

請使用下列程序連線至資料庫。

連線至資料庫

1. 開啟資料庫伺服器的內容 (按一下滑鼠右鍵) 功能表，然後選擇 Pro(建立連線)。

您也可以選擇資料庫結構描述樹頂端的 [Connect 到伺服器]。

2. 輸入連線到來源資料庫伺服器的密碼。
3. 選擇 [測試連線] 以確認是否AWS SCT可以連線到來源資料庫。
4. 選擇「Connect」以連線到來源資料庫。

請使用下列程序從AWS SCT專案中移除資料庫伺服器。

若要移除資料庫伺服器

1. 選擇要移除的資料庫伺服器。
2. 開啟內容 (按一下右鍵) 功能表，然後選擇 Chine (從專案中移除)。

AWS SCT移除選取的資料庫伺服器、所有對應規則、轉換結果，以及與此伺服器相關的其他中繼資料。

以離線模式執行AWS SCT

您可以AWS Schema Conversion Tool在離線模式下執行。接下來，您可以了解與來源資料庫中斷連線時，如何使用現有的AWS SCT專案。

AWS SCT不需要連接到源數據庫即可運行以下操作：

- 新增對應規則。
- 建立資料庫移轉評估報告。
- 轉換數據庫模式和代碼。
- 編輯源代碼和轉換後的代碼。
- 將源代碼和轉換後的代碼保存為文本文件中的 SQL 腳本。

AWS SCT在離線模式下使用之前，請先連線到來源資料庫、載入中繼資料並儲存專案。開啟此專案，或中斷與來源資料庫伺服器的連線，以便AWS SCT在離線模式中使用。

AWS SCT在離線模式下執行

1. 啟動AWS Schema Conversion Tool並建立新的專案。如需詳細資訊，請參閱[建立AWS SCT專案](#)。
2. 新增來源資料庫伺服器並連線至來源資料庫。如需詳細資訊，請參閱[將資料庫伺服器新增至AWS SCT專案](#)。
3. 新增目標資料庫伺服器或使用虛擬目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。
4. 建立對應規則以定義來源資料庫的目標資料庫平台。如需詳細資訊，請參閱[在中建立對應規則AWS SCT](#)。
5. 選擇 [檢視]，然後選擇 [主要檢視]。
6. 在顯示來源資料庫物件的左側面板中，選擇您的來源資料庫結構描述。開啟物件的內容 (按一下右鍵) 選單，然後選擇 Load Scheme (載入結構描述)。此作業會將所有來源結構描述資料載入您的AWS SCT專案。

[建立報表] 和 [轉換結構描述] 作業也會將所有來源結構描述中繼資料載入您AWS SCT 如果您從內容功能表執行這些作業之一，請略過「載入結構描述」作業。

7. 在 [檔案] 功能表上，選擇 [儲存專案]，將來源資料庫中繼資料儲存在專案中。
8. 選擇中斷與伺服器的連線，以中斷與來源資料庫的連線。現在，您可以AWS SCT在離線模式下使用。

使用AWS SCT樹狀篩選器

為了將資料從來源遷移到目標，AWS SCT 會將來源和目標資料庫的所有中繼資料都載入到樹狀目錄中。這個結構會顯示在 AWS SCT 主專案視窗的樹狀檢視中。

有些資料庫在樹狀目錄中可能會有大量物件。您可以在 中使用樹狀篩選條件AWS SCT來搜尋來源和目標樹狀目錄中的物件。當您使用樹狀篩選條件，不會變更當您轉換資料庫時所轉換的物件。篩選器只會變更您在樹狀結構中看到的內容。

樹狀篩選條件只適用於 AWS SCT 已預先載入的物件。換言之，AWS SCT 在搜尋期間不會從資料庫載入物件。這個方法表示樹狀目錄包含的物件通常少於資料庫中存在的物件。

對於樹狀篩選條件，請記住下列各項：

- 篩選條件預設為 ANY，這表示篩選條件使用名稱搜尋來尋找物件。
- 當您選擇一或多個物件類型，您只會在樹狀目錄中看到這些類型的物件。

- 您可以使用篩選條件遮罩來顯示不同的符號類型，包括 Unicode、空格和特殊字元。「%」字元是任何符號的萬用字元。
- 套用篩選條件後，計數只會顯示篩選後的物件數量。

若要建立樹狀篩選條件

1. 開啟現有的 AWS SCT 專案。
2. Connect 至您要套用樹狀目錄篩選的資料庫。
3. 選擇篩選條件圖示。



復原篩選條件圖示呈現灰色，因為目前未套用篩選條件。

4. 在「篩選」對話方塊中輸入以下資訊。對話方塊中的選項會因每個資料庫引擎而不同。

AWS SCT篩選器選項	動作
等級	<p>選擇「品類」以依品類篩選物件。</p> <p>選擇「狀態」以按狀態過濾對象。</p>
類型	<p>對於「層級中的品類」，選擇篩選物件的品類。選擇「任何已載入」以顯示所有品類中的物件。</p> <p>對於「層級中的狀態」，請選擇已篩選物件的狀態。您可以選擇以下其中一個選項：</p> <ul style="list-style-type: none"> • 轉換為顯示所有轉換的物件 • 具有顯示有轉換問題的所有物件的動作 • 加密以顯示所有加密物件
Condition	<p>對於「層級中的類別」，選擇「喜歡」和「不喜歡」之間的過濾條件。</p> <p>對於「層級中的狀態」，篩選條件選項無法使用。</p>

AWS SCT篩選器選項	動作
Value (值)	對於「層次中的品類」，請輸入值以依此值篩選樹狀結構。 使用百分比 (%) 作為萬用字元來顯示所有物件。 對於「層次中的狀態」，選擇「真」與「假」之間的值。
及/或	選擇AND或OR邏輯運算子以套用多個篩選子句。

- 選擇 [新增子句] 以新增其他篩選條件子句。AWS SCT可以使用AND或OR邏輯運算符應用多個過濾器子句。
- 選擇 Apply (套用)。選擇 Apply (套用) 後，會啟用還原篩選條件圖示 (篩選條件圖示旁)。如果您想要移除您套用的篩選條件，請使用此圖示。
- 選擇 Close (關閉) 來關閉對話方塊。

當您篩選顯示在樹狀目錄中的結構描述，並不會變更當您轉換結構描述時所轉換的物件。篩選條件只會變更您在樹狀目錄中所看到的項目。

匯入樹狀結構篩選的檔案清單

您可以匯入含有分號分隔符號的逗號分隔值 (CSV) 檔案，或包含您希望樹狀篩選器使用的名稱或值的 JSON 檔案。開啟現有 AWS SCT 專案，連線至要套用樹狀篩選的資料庫，然後選擇篩選器圖示。

若要下載檔案範例，請選擇 [下載範本]。輸入檔案名稱，然後選擇 [儲存]。

若要下載現有的篩選器設定，請選擇「匯出」。輸入檔案名稱，然後選擇 [儲存]。

若要匯入樹狀篩選的檔案清單，請選擇「匯入」。選擇要匯入的檔案，然後選擇 Open (開啟)。選擇 Apply (套用)，然後選擇 Close (關閉)。

CSV 文件使用分號作為分隔符，並具有以下格式：

- `object_type` 是您要尋找的物件類型。
- `database_name` 是此物件存在的資料庫名稱。
- `schema_name` 是此物件存在的結構描述名稱。
- `object_name` 是物件名稱。
- `import_type` 指定為 `include` 或從過濾器中的 `exclude` 此項目。

使用 JSON 檔案描述複雜的篩選案例，例如巢狀規則。JSON 檔案格式如下：

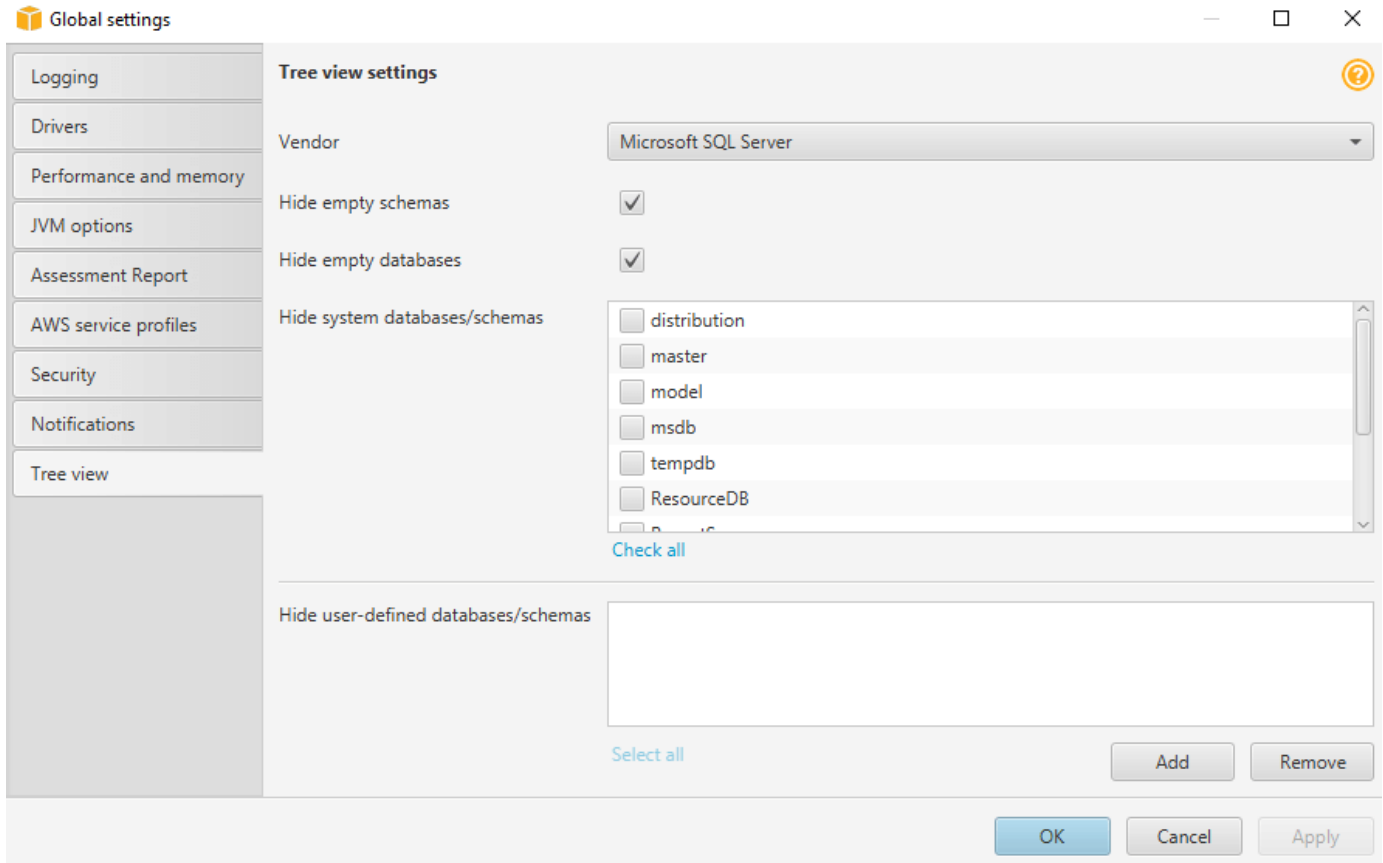
- `filterGroupType` 是套用至多個篩選子句的篩選規則類型 (AND 或 OR 邏輯運算子)。
- `filterCategory` 是篩選條件的層級 (「類別」或「狀態」)。
- `names` 是適用於「品類」篩選器的物件名稱清單。
- `filterCondition` 是套用於「品類」篩選的篩選條件 (LIKE 或 NOT LIKE)。
- `transformName` 是適用於「狀態」篩選器的狀態名稱。
- `value` 是篩選樹狀結構所依據的值。
- `transformValue` 是套用於「狀態」篩選器的篩選器 (TRUE 或 FALSE) 值。

在 AWS SCT 樹狀檢視中隱藏資料架構

透過使用樹狀檢視設定，可以指定您要在 AWS SCT 樹狀檢視中查看哪些結構描述和資料庫。您可以隱藏空的結構描述、空的資料庫、系統資料庫以及使用者定義的資料庫和結構描述。

若要在樹狀檢視中隱藏資料庫和結構描述

1. 開啟 AWS SCT 專案。
2. 連接到您要顯示在樹狀檢視中的資料存放區。
3. 選擇設定、全域設定、樹狀檢視。



4. 在樹狀檢視設定區段中，執行下列動作：
 - 對於供應商，選擇資料庫平台。
 - 選擇隱藏空白綱要，隱藏所選資料庫平台的空白綱要。
 - 選擇隱藏空白資料庫可隱藏所選資料庫平台的空白資料庫。
 - 對於隱藏系統資料庫/綱要，請依名稱選擇系統資料庫和綱要來隱藏它們。
 - 在 [隱藏使用者定義的資料庫/綱要] 中，輸入您要隱藏的使用者定義資料庫和綱要的名稱，然後選擇新增。名稱區分大小寫。
5. 選擇 OK (確定)。

建立及檢閱資料庫移轉評估報告

資料庫遷移評估報告總結了無法自動轉換為目標 Amazon RDS 資料庫執行個體引擎之結構描述的所有行動項目。本報告亦包含撰寫目標資料庫執行個體的相同程式碼所需要進行的作業量估算。

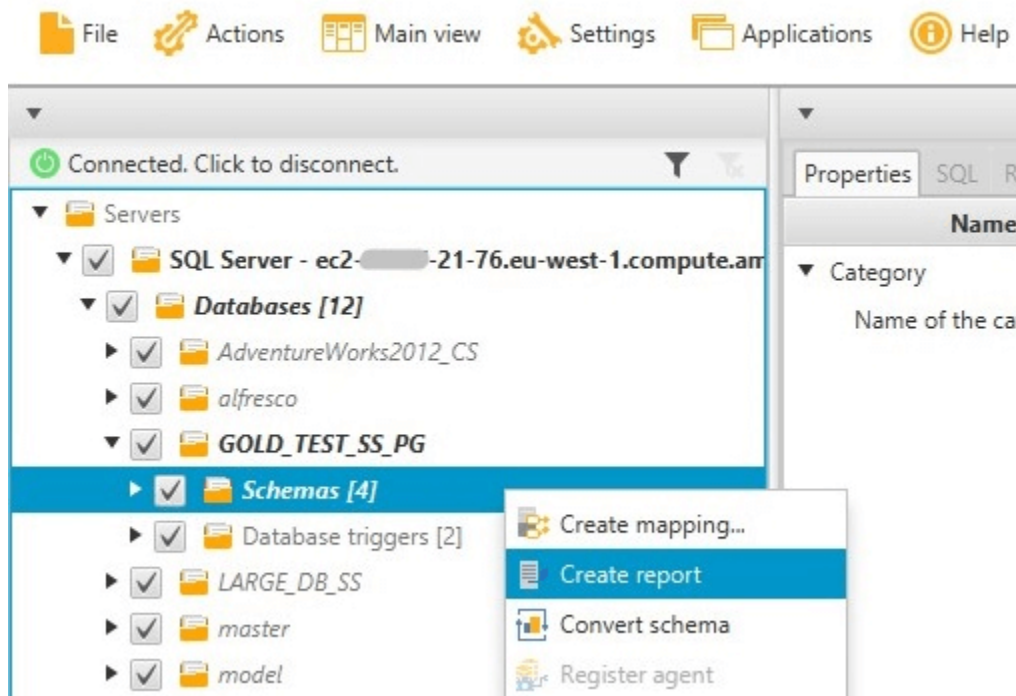
您可以在將來源資料庫和目標平台新增至專案並指定對應規則之後，建立資料庫移轉評估報告。

若要建立並檢視資料庫遷移評估報告

1. 請確定您已為來源資料庫綱要建立對應規則，以建立評估報告。如需詳細資訊，請參閱[新增對應規則](#)。
2. 在「檢視」功能表上，選擇「主視圖」。
3. 在顯示來源資料庫結構描述的左側面板中，選擇要建立評估報告的綱要物件。

請確定您已選取所有綱要物件的核取方塊，以建立評估報告。

4. 開啟物件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Props (建立報告)。



評估報告檢視會開啟。

5. 選擇「事務處理」標籤。

[動作項目] 索引標籤會顯示項目清單，這些項目描述無法自動轉換的結構描述。選擇清單中的其中一個行動項目。AWS SCT反白行動項目套用至的結構描述中的項目，如下所示。

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, schemas, and tables. The main pane shows a list of issues with their recommended actions and occurrence counts. Below the issues, a SQL procedure code snippet is visible, along with a table showing the target Amazon RDS schema details.

Issues Summary:

- Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table.
- Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically.
- Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype.
- Issue 826:** Check the default value for a DateTime variable.
- Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision.
- Issue 9997:** Unable to resolve objects.
- Issue 690:** MySQL doesn't support table types.
- Issue 811:** Unable to convert functions.

SQL Code Snippet:

```

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

```

Target Amazon RDS Schema:

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

6. 選擇 Summary (摘要) 索引標籤。

Summary (摘要) 索引標籤會顯示資料庫遷移評估報告的摘要資訊。它會顯示已自動轉換的項目數，以及未自動轉換的項目數。摘要還包含在您的目標資料庫執行個體中，建立相當於來源資料庫中同等項目之結構描述的預估時間。

授權評估和雲端 Support 一節包含有關將現有現場部署資料庫結構描述移至執行相同引擎之 Amazon RDS 資料庫執行個體的相關資訊。例如，如果您想要變更授權類型，報告的這個區段會告訴您要從目前資料庫中移除哪些功能。

以下顯示評估報告摘要的範例。

Summary | Action items

Save to CSV Save to PDF

Database migration assessment report

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

Executive summary

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

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

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects

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

- 選擇 Summary (摘要) 索引標籤，然後選擇 Save to PDF (儲存至 PDF)。資料庫遷移評估報告會儲存為 PDF 檔案。PDF 檔案同時包含摘要和動作項目資訊。

您也可以選擇 [儲存為 CSV]，將報告儲存為 CSV 檔案。當您選擇此選項時，AWS SCT 會建立三個 CSV 檔案。這些檔案包含下列資訊：

- 包含建議動作的轉換動作項目清單。
- 轉換行動項目摘要，其中包含轉換行動項目出現次數所需的預估工作量。
- 執行摘要，其中包含許多行動項目 (依預估轉換時間分類)。

Database objects with conversion actions for Amazon RDS for PostgreSQL

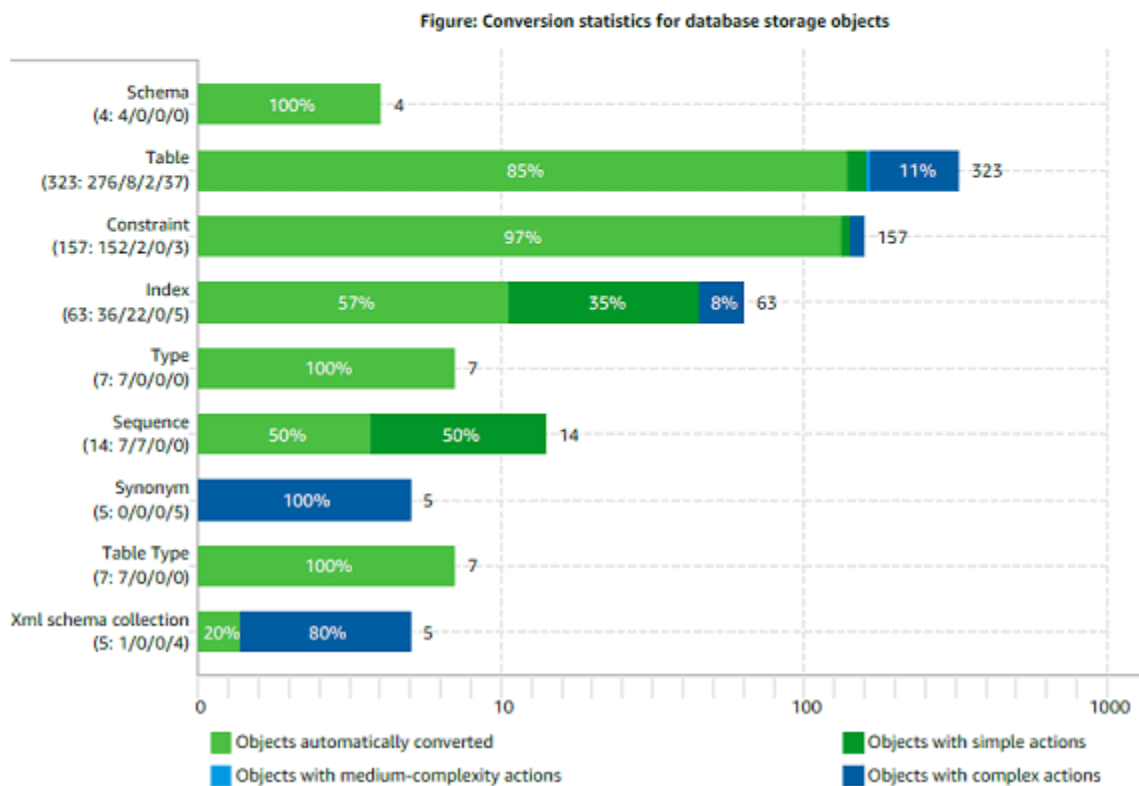
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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

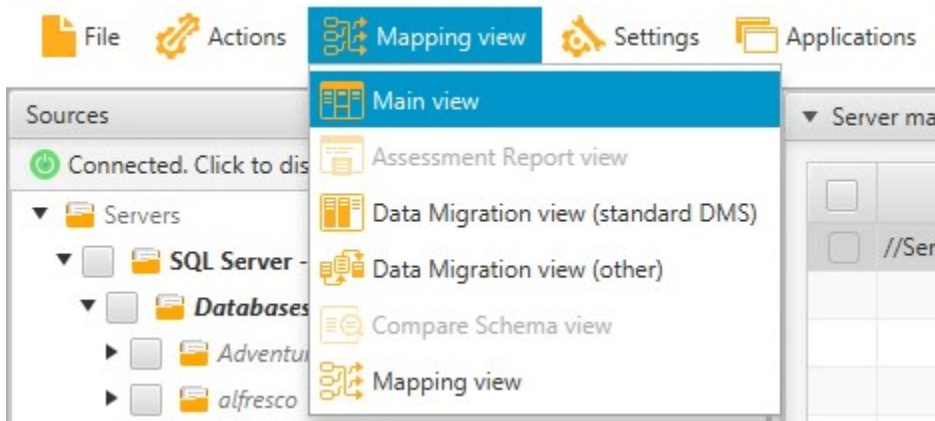


轉換您的結構描

將來源和目標資料庫新增至專案並建立對應規則之後，您可以轉換來源資料庫結構描述。使用下列程序來轉換結構描述。

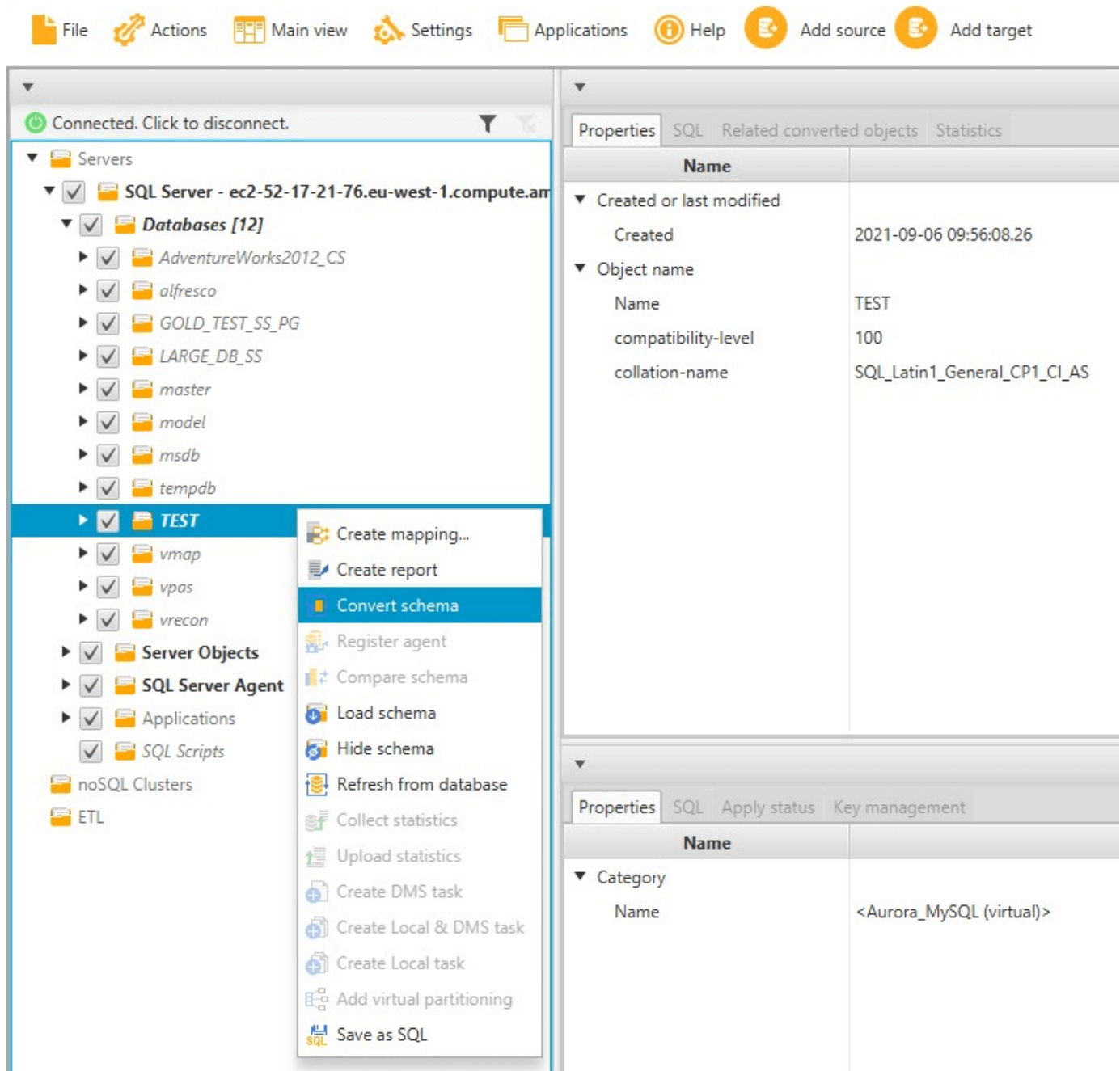
轉換您的結構描述

1. 選擇 [檢視]，然後選擇 [主要檢視]。



2. 在顯示來源資料庫結構描述的左側面板中，選取要轉換之物件名稱的核取方塊。接下來，選擇這個對象。AWS SCT以藍色亮顯物件名稱。開啟物件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Props (轉換結構描述)

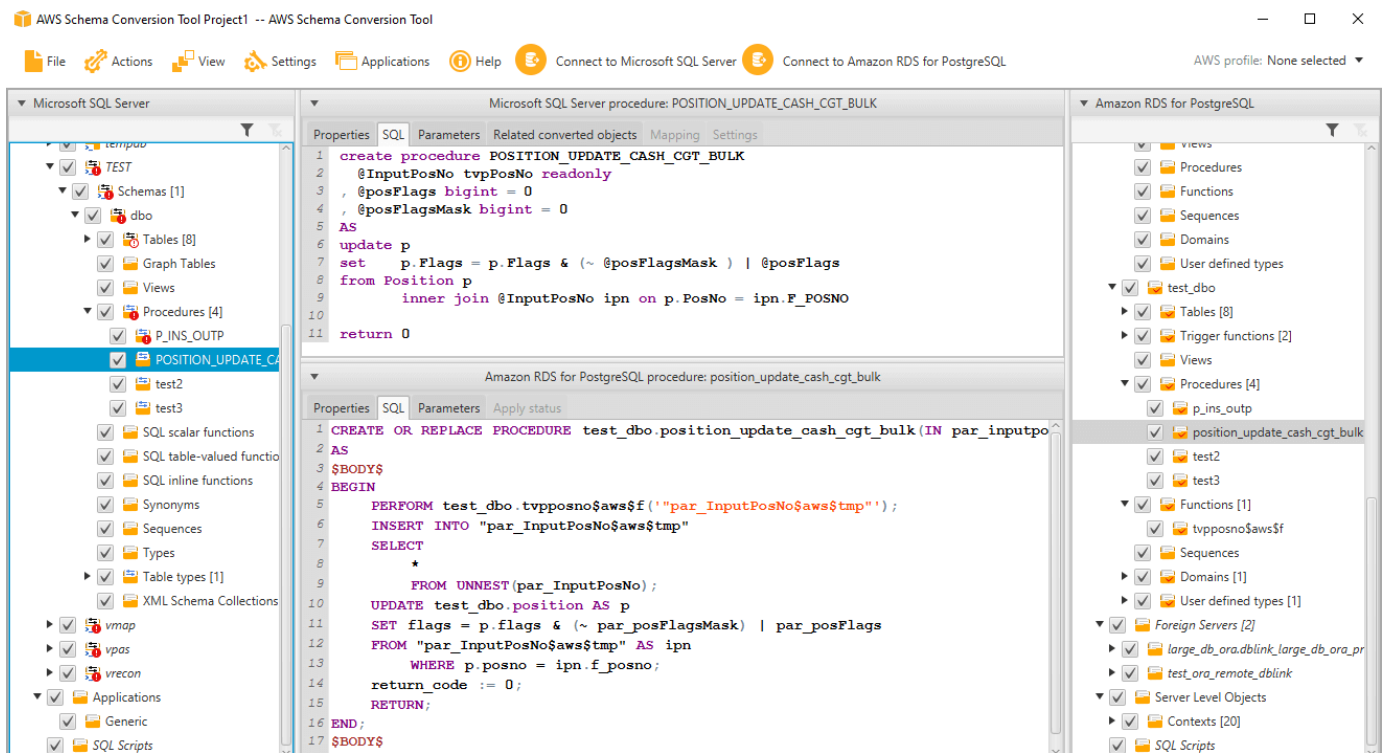
若要轉換數個資料庫物件，請選取所有物件的勾選方塊。接下來，選擇父節點。例如，對於資料表，父節點是 [資料表]。請確定以藍色AWS SCT反白父節點的名稱。開啟父節點的內容 (按一下滑鼠右鍵) 功能表，然後選擇 Props (轉換結構描述)



- 當 AWS SCT 完成轉換結構描述時，您可以在專案右邊的面板中檢視提議的結構描述。

此時，不會將結構描述套用至您的目標資料庫執行個體。計劃的結構描述屬於您專案的一部分。如果您選擇轉換的結構描述項目，您可以在目標資料庫執行個體中下方中央的面板中看到已規劃的結構描述命令。

您可以在這個視窗中編輯結構描述。編輯過的結構描述會儲存為專案的一部分，當您選擇套用轉換後的結構描述時，會寫入目標資料庫執行個體。



將轉換後的結構描述套用至您的目標資料庫

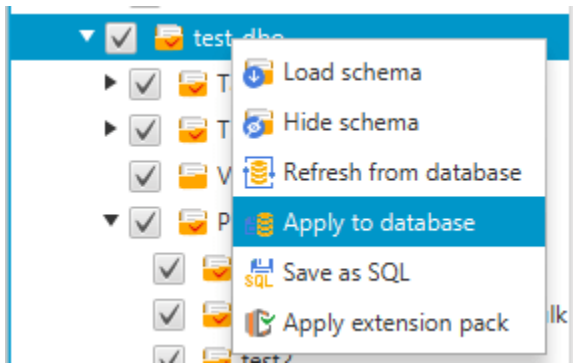
您可將轉換後的結構描述套用至您的目標 資料庫執行個體。將結構描述套用至您的目標資料庫執行個體之後，您可以根據資料庫遷移評估報告中的動作項目來更新結構描述。

⚠ Warning

下列程序會覆寫現有的目標結構描述。請注意不要意外覆寫結構描述。請注意不要覆寫目標資料庫執行個體中已修改的結構描述，或覆寫這些變更。

將轉換後的資料庫綱要套用至目標資料庫執行處理

1. 選擇項目右側面板頂部的「Connect 到服務器」以連接到目標數據庫。如果您已連線至目標資料庫，請跳過此步驟。
2. 在顯示您目標資料庫執行個體之計劃結構描述的專案右側面板中，選擇結構描述元素。
3. 開啟結構描述元素的內容 (按一下右鍵) 選單，然後選擇 Apply to database (套用至資料庫)。



轉換後的結構描述已套用至目標資料庫執行個體。

將AWS服務設定檔儲存在AWS SCT

您可以將AWS憑據存儲在AWS SCT。AWS SCT當您使用與服務整合的功能時，會使用您的認AWS證。例如，AWS SCT與 Amazon S3、AWS Lambda、Amazon Relational Database Service 服務 (Amazon RDS) 和AWS Database Migration Service (AWS DMS) 整合。

AWS SCT當您存取需要AWS認證的功能時，會詢問您提供認證。您可以將登入資料存放在全域應用程式設定中。當 AWS SCT 詢問您的登入資料，您就可以選擇已存放的登入資料。

您可以在全域應用程式設定中儲存不同的AWS認證集。例如，您可以存放一組用於測試案例的登入資料，以及另一組用於生產案例的登入資料。您也可以為不同的AWS 區域 s 存儲不同的憑據。

儲存AWS認證

請使用下列程序來全域儲存AWS認證。

儲存AWS認證

1. 啟動 AWS Schema Conversion Tool。
2. 開啟 [設定] 功能表，然後選擇 [全域設定]。Global settings (全域設定) 對話方塊隨即出現。
3. 選擇AWS服務設定檔，然後選擇 [新增AWS服務設定檔]。
4. 輸入您的AWS信息，如下所示。

AWS SCT 選項	動作
Profile name (設定檔名稱)	為您的個人資料輸入名稱。

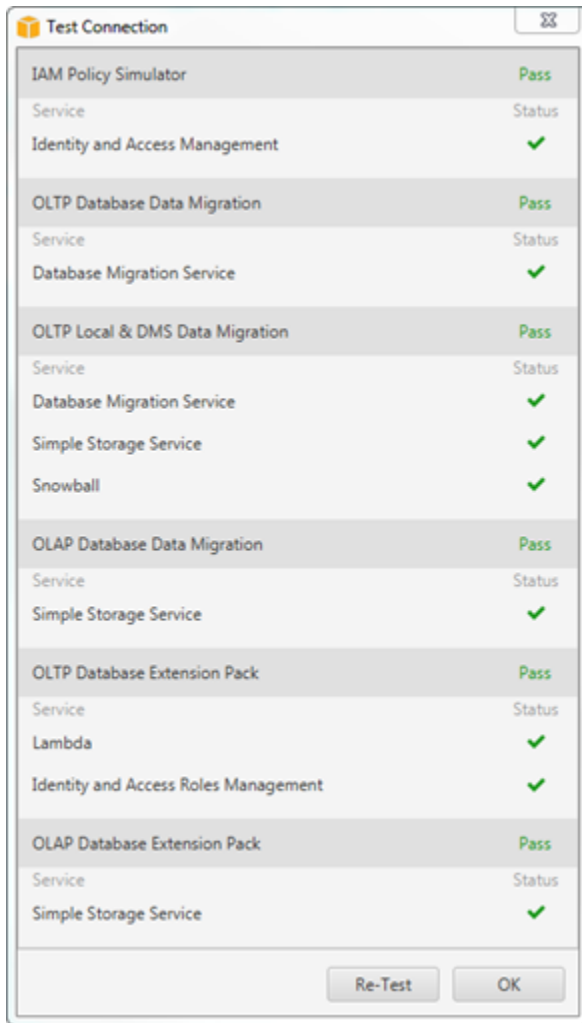
AWS SCT 選項	動作
AWS存取金鑰	輸入您的AWS存取金鑰。
AWS秘密金鑰	輸入您的AWS私密存取金鑰。如需有關AWS存取金鑰的詳細資訊，請參閱《IAM 使用者指南》中的 管理存取金鑰 。
區域	選擇AWS 區域適合您的個人檔案。
Amazon S3 儲存貯體資料夾	為您的設定檔選擇 Amazon S3 儲存貯體。只有在使用連線到 Amazon S3 的功能時，才需要指定儲存貯體。如需必要的權限的詳細資訊，請參閱 使用服AWS務設定檔的權限 。

如果您需要遵守「聯邦資訊處理標準」(FIPS) 的安全要求，請選擇 Use FIPS endpoint for S3 (使用 S3 的 FIPS 端點)。FIPS 端點適用於下列AWS區域：

- 美國東部 (維吉尼亞北部) 區域
- 美國東部 (俄亥俄) 區域
- 美國西部 (加利佛尼亞北部) 區域
- 美國西部 (奧勒岡) 區域

5. 選擇 [測試連線] 以確認您的認證是否正確且作用中。

[測試連線] 對話方塊隨即出現。您可以查看連接到您設定檔的每個服務狀態。Pass (通過) 表示設定檔可以成功存取服務。



- 設定好設定檔之後，選擇 Save (儲存) 以儲存您的設定檔，或選擇 Cancel (取消) 以取消變更。
- 選擇 [確定] 以關閉 [整體設定] 對話方塊。

設定專案的預設紀要

您可以設定 AWS SCT 專案的預設設定檔。這樣做會將儲存在設定檔中的AWS認證與專案產生關聯。開啟專案時，使用下列程序來設定預設設定檔。

若要設定專案的預設設定檔

- 啟動AWS Schema Conversion Tool並建立新的專案。
- 在 [設定] 功能表上，選擇 [專案設定]。將顯示「專案設定」對話方塊。
- 選擇「專案環境」頁籤。

4. 選擇 [新增AWS服務設定檔] 以新增設定檔。然後，對於AWS服務配置文件，選擇要與項目關聯的配置文件。
5. 選擇「確定」以關閉「專案設定」對話方塊。您也可以選擇 Cancel (取消) 以取消變更。

使用服AWS務設定檔的權限

從AWS服務設定檔存取 Amazon S3 儲存貯體需要下列許可：

- s3:PutObject— 在 Amazon S3 儲存貯體中新增物件。
- s3:DeleteObject— 移除物件的 null 版本，然後插入刪除標記，這會變成物件的目前版本。
- s3:ListBucket— 從您的 Amazon S3 存儲桶返回多達 1,000 個對象。
- s3:GetObject— 從 Amazon S3 儲存貯體擷取物件。

下列程式碼範例示範如何將這些權限准許准許給您的使用者。

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

使用 AWS Secrets Manager

AWS SCT可以使用您儲存的資料庫認證AWS Secrets Manager。您可以從 Secrets Manager 填入資料庫連線對話方塊中的所有值。若要使用 Secrets Manager，請確定您已將設定AWS檔儲存在AWS Schema Conversion Tool.

如需有關使用的詳細資訊AWS Secrets Manager，請參閱[什麼是AWS Secrets Manager？](#) 在《AWS Secrets Manager使用者指南》中。如需儲存AWS描述檔的詳細資訊，請參閱[將AWS服務設定檔儲存在AWS SCT。](#)

若要從 Secrets Manager 擷取資料庫認證

1. 啟動AWS Schema Conversion Tool並建立新的專案。
2. 選擇「添加源」或「添加目標」以將新數據庫添加到項目中。
3. 選擇資料庫平台，然後選擇 [下一步]。
4. 針對 AWS Secret，請選擇您要使用的秘密。
5. 選擇「植入」。然後AWS SCT填寫資料庫連線對話方塊中的所有值。
6. 選擇 [測試連線] 以確認是否AWS SCT可以連線到您的資料庫。
7. 選擇 Connect 以連線至您的資料庫。

AWS SCT支援具有下列結構的秘密。

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

在此結構中，username和password值是必要的，而且所有其他值都是可選的。請確定您儲存在 Secrets Manager 中的值包含所有資料庫認證。

儲存資料庫密碼

您可以將資料庫密碼或 SSL 憑證存放在 AWS SCT 快取中。若要存放密碼，請在建立連線時選擇 Store Password (存放密碼)。

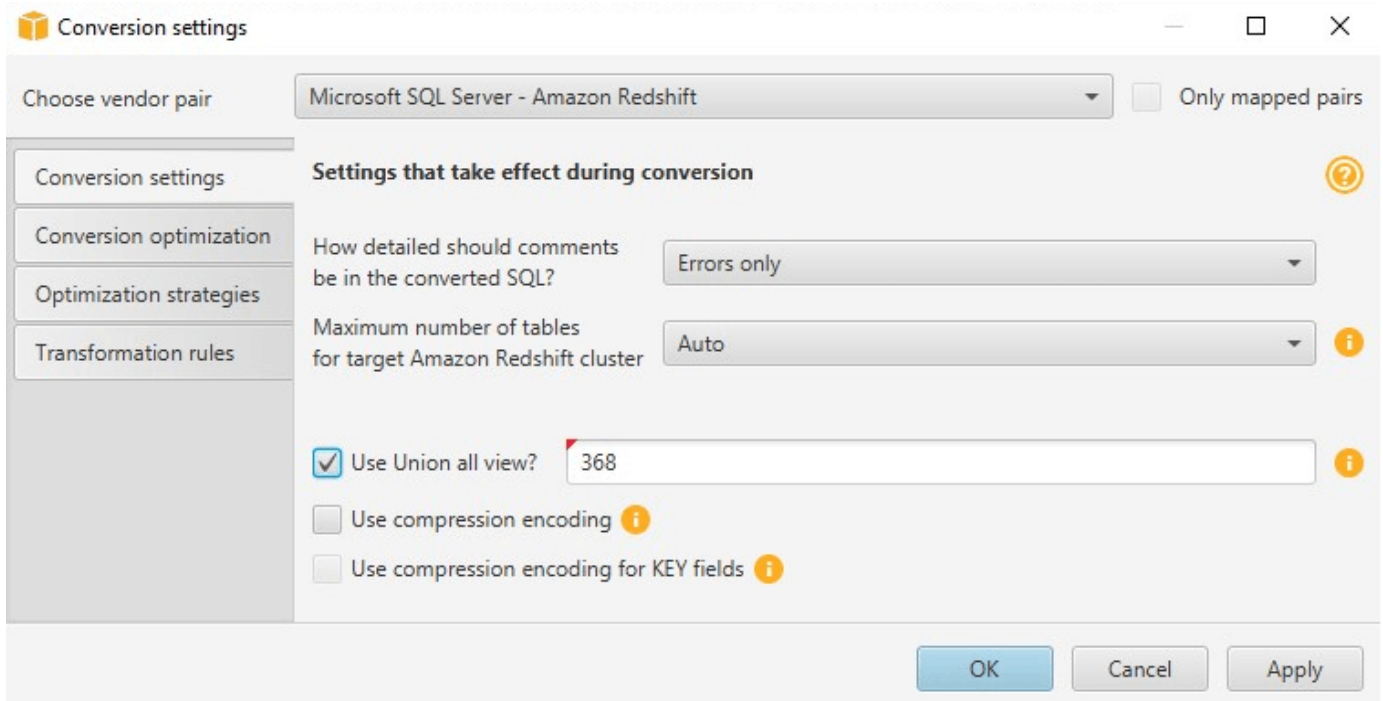
系統會使用 seed.dat 檔案中隨機產生的符記來加密密碼。接著將使用者名稱和密碼一起存放在快取檔案中。如果您遺失 seed.dat 檔案或檔案損毀，資料庫密碼可能無法正確解密。在這種情況下，連線會失敗。

針對具有分區資料表的專案使用 UNION ALL 檢視

如果來源資料表已分割，則AWS SCT會建立 n 個目標資料表，其中 n 是來源資料表上的分割區數目。AWS SCT在目標表的頂部創建一個 UNION ALL 視圖來表示源表。如果您使用AWS SCT資料擷取器來移轉資料，則來源資料表分割區將由個別的子工作 parallel 擷取並載入。

使用專案的 Union All 檢視

1. 啟動 AWS SCT。建立新專案或開啟現有AWS SCT專案。
2. 在 [設定] 功能表上，選擇 [轉換設定]。
3. 從頂端的清單中選擇一對 OLAP 資料庫。
4. 開啟 [使用聯集全部檢視]？



5. 選擇「確定」以儲存設定並關閉「轉換設定」對話方塊。

鍵盤快速鍵AWS SCT

以下是您可以搭配使用的鍵盤快速鍵AWS SCT。

鍵盤快速鍵	描述
Ctrl+N	建立新專案。

鍵盤快速鍵	描述
Ctrl+O	開啟現有的專案。
Ctrl+S	儲存開啟的專案。
Ctrl+W	使用精靈建立新的專案。
按 Ctrl+ M	建立新的多伺服器評估。
Ctrl+L	新增新的來源資料庫。
Ctrl+R	新增新的目標資料庫。
按 Ctrl+ F4	關閉開啟的專案。
F1	開啟使AWS SCT用者指南。

AWS SCT 入門

您可以使用AWS Schema Conversion Tool(AWS SCT)，以轉換來源資料庫的結構描述。來源資料庫可以是以內部部部部部部部部部部部部部部部部部署方式執行的自我管理引擎。您可以將來源結構描述轉換為由下列項目所託管之任何受支援資料庫的結構描述AWS。AWS SCT 應用程式提供以專案為基礎的使用者介面。

您使用 AWS SCT 執行的幾乎所有工作一開始皆要執行下列三個步驟：

1. 安裝 AWS SCT。如需詳細資訊，請參閱 [安裝、驗證和更新 AWS SCT](#)。
2. 安裝AWS SCT代理，如果需要的話。AWS SCT只有特定移轉案例 (例如異質來源與目標之間) 才需要代理程式。如需詳細資訊，請參閱 [將資料從現場部署資料倉儲遷移到 Amazon Redshift](#)。
3. 熟悉的使用者介面AWS SCT。如需詳細資訊，請參閱 [使用使用AWS SCT者介面](#)。
4. 建立 AWS SCT 專案。連接到來源和目標資料庫。如需有關連接到來源資料庫的詳細資訊，請參閱[AWS SCT 的來源](#)。
5. 建立對應規則。如需映射規則的詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。
6. 執行資料庫遷移評估報告，並檢視報告。如需評估報告的詳細資訊，請參閱[建立及檢閱資料庫移轉評估報告](#)。
7. 轉換來源資料庫結構描述。轉換有許多方面需要注意，例如要如何處理不轉換的項目，以及如何對應該以特定方式轉換的項目。如需轉換來源結構描述的詳細資訊，請參閱[轉換資料庫結構描述 AWS SCT](#)。

如果要轉換資料倉儲結構描述，轉換前還有些事項需要考量。如需詳細資訊，請參閱 [使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT](#)。

8. 將結構描述轉換套用至您的目標。如需套用來源結構描述轉換的詳細資訊，請參閱[應用轉換後的代碼](#)。
9. 您也可以使用AWS SCT轉換 SQL 存儲過程和其他應用程序代碼。如需詳細資訊，請參閱《》[使用轉換應用程式 SQLAWS SCT](#)

您也可以使用 AWS SCT 將資料從來源資料庫遷移到 Amazon 受管的資料庫。如需範例，請參閱[將資料從現場部署資料倉儲遷移到 Amazon Redshift](#)。

AWS SCT 的來源

AWS Schema Conversion Tool (AWS SCT) 可以將結構描述從下列來源資料庫和資料倉儲轉換為目標資料庫或資料倉儲。如需權限、連線以及AWS SCT可轉換哪些項目以搭配目標資料庫或資料倉儲使用的相關資訊，請參閱下列主題中的詳細資訊。

加密資訊

[加密亞馬遜 RDS 連接](#)

資料庫來源

- [使用阿帕奇卡桑德拉作為源](#)
- [使用 Azure SQL 資料庫做為來源](#)
- [使用適用於 z/OS 的 IBM Db2 作為來源](#)
- [使用 IBM Db2 LUW 做為來源](#)
- [使用 MySQL 做為來源](#)
- [使用 Oracle 資料庫作為來源](#)
- [使用 PostgreSQL 做為來源](#)
- [使用 SAP 日月光 \(系統日月光\) 作為來源](#)
- [使用 SQL Server 做為來源](#)

資料倉儲來源

- [使用亞馬遜紅移作為源](#)
- [使用 Azure 突觸分析作為來源](#)
- [使用 BigQuery 做為來源](#)
- [使用綠色數據庫作為來源](#)
- [使用尼泰扎作為來源](#)
- [使用 Oracle 資料倉儲作為來源](#)
- [使用雪花作為來源](#)
- [使用 SQL 伺服器資料倉儲作為來源](#)
- [使用太數據作為源](#)

- [使用 Vertica 作為來源](#)

大數據源

- [使用阿帕奇 Hadoop 作為源](#)
- [使用阿帕奇大西作為來源](#)

加密亞馬遜 RDS 和亞馬遜極光連接 AWS SCT

若要從應用程式開啟與 Amazon RDS 或 Amazon Aurora 資料庫的加密連線，您需要將AWS根憑證匯入某種形式的金鑰儲存體。您可以從 AWS Amazon RDS 使用者指南中的[使用 SSL/TLS 加密與資料庫執行個體的連線](#)下載根憑證。

有兩個選項可供選擇：適用於所有AWS區域的根憑證，以及同時包含新舊根憑證的憑證套裝軟體。

根據您要使用的項目，遵循下列兩個程序之一中的步驟。

將一或多個憑證匯入 Windows 系統儲存區

1. 從下列其中一個來源下載一個或多個憑證：

如需下載憑證的相關資訊，請參閱 Amazon RDS 使用者[指南中的使用 SSL/TLS 加密與資料庫執行個體的連線](#)。

2. 在您的視窗搜尋視窗中，輸入**Manage computer certificates**。當系統提示是否讓應用程式對您的電腦進行變更時，請選擇 [是]。
3. 憑證視窗開啟時，如果需要，請展開 [憑證-本機電腦]，以便您可以查看憑證清單。開啟「受信任的根憑證授權單位」的內容 (按一下滑鼠右鍵) 功能表，然後選擇「所有工作」
4. 選擇「下一步」，然後選擇「瀏覽」，然後尋找您在步驟 1 中下載的*.pem檔案。選擇 [開啟] 以選取憑證檔案，選擇 [下一步]，然後選擇 [完成]。

Note

若要尋找檔案，請將瀏覽視窗中的檔案類型變更為「所有檔案 (*.*)」，因為不.pem是標準的憑證副檔名。

5. 在微軟管理主控台中，展開憑證。然後展開「受信任的根憑證授權單位」，選擇「憑證」，然後尋找憑證以確認其存在。憑證的名稱開頭為Amazon RDS。

6. 重新啟動電腦。

若要將一或多個憑證匯入 Java KeyStore

1. 從下列其中一個來源下載一或多個憑證：

如需下載憑證的相關資訊，請參閱 Amazon RDS 使用者 [指南中的使用 SSL/TLS 加密與資料庫執行個體的連線](#)。

2. 如果您已下載憑證服務包，請將其分割為個別的憑證檔案。若要這樣做，請將每個憑證區塊 (以開頭-----BEGIN CERTIFICATE-----和結尾為) 放-----END CERTIFICATE-----入不同的 *.pem 檔案中。為每個憑證建立個別的 *.pem 檔案之後，您可以安全地移除憑證服務包檔案。
3. 在您下載憑證的目錄中開啟命令視窗或終端機工作階段，並針對您在上一個步驟中建立的每個 *.pem 檔案執行下列命令。

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

Example

下列範例假設您已下載 eu-west-1-bundle.pem 檔案。

```
keytool -importcert -file eu-west-1-bundle.pem -alias eu-west-1-bundle.pem -
keystore trust-2019.ks
Picked up JAVA_TOOL_OPTIONS: -Dlog4j2.formatMsgNoLookups=true
Enter keystore password:
Re-enter new password:
Owner: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
ST=Washington, L=Seattle, C=US
Issuer: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
ST=Washington, L=Seattle, C=US
Serial number: c73467369250ae75
Valid from: Thu Aug 22 19:08:50 CEST 2019 until: Thu Aug 22 19:08:50 CEST 2024
Certificate fingerprints:
    SHA1: D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96
    SHA256:
    F2:54:C7:D5:E9:23:B5:B7:51:0C:D7:9E:F7:77:7C:1C:A7:E6:4A:3C:97:22:E4:0D:64:54:78:FC:70:AA:
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:
```

```

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

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

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

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

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

```

- 將金鑰儲存區新增為中的信任存放區AWS SCT。若要這麼做，請從主功能表選擇 [設定]、[全域設定]、[安全性]、[信任存放區]，然後選擇 [選取現有的信任存放區]

新增信任存放區之後，您可以在建立與資料庫的連線時，使用它來設定啟用 SSL 的 AWS SCT 連線。在 [AWS SCT 連線到資料庫] 對話方塊中，選擇 [使用 SSL]，然後選擇先前輸入的信任存放區

使用阿帕奇卡桑德拉作為源 AWS SCT

您可以使用 AWS SCT 將密鑰空間從阿帕奇卡桑德拉轉換為亞馬遜 DynamoDB。

連接到阿帕奇卡桑德拉作為源

請使用下列程序來連接到您的 Apache 卡桑德拉來源資料庫。AWS Schema Conversion Tool

要連接到一個阿帕奇卡桑德拉源數據庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇卡桑德拉，然後選擇下一步。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 要手動輸入 Apache 卡桑德拉源數據庫連接信息，請使用以下說明：

參數	動作
伺服器名稱	輸入您來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
Use SSL (使用 SSL)	如果您要使用安全通訊端層 (SSL) 連線到資料庫，請選擇此選項。在 SSL 索引標籤上提供下列適用的其他資訊： <ul style="list-style-type: none"> • 信任存放區：要使用的信任存放區。

參數	動作
	<ul style="list-style-type: none"> 金鑰存放區：要使用的金鑰存放區。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。

- 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
- 選擇「連線」以連線到來源資料庫。

使用阿帕奇 Hadoop 作為源 AWS SCT

您可以使用命AWS SCT命令行界面 (CLI) 從阿帕奇 Hadoop 遷移到亞馬遜 EMR。AWS SCT使用 Amazon S3 儲存貯體做為移轉期間資料的暫時儲存。

AWS SCT支持作為一個源阿帕奇 Hadoop 的版本 2.2.0 及更高版本。此外，AWS SCT支持阿帕奇蜂巢版本 0.13.0 及更高版本。

AWS SCT支持作為目標亞馬遜 EMR 版本 6.3.0 及更高版本。此外，AWS SCT支持作為一個目標阿帕奇 Hadoop 版本 2.6.0 及更高版本，和阿帕奇蜂巢版本 0.13.0 及更高版本。

主題

- [使用阿帕奇 Hadoop 作為源的先決條件](#)
- [使用 Hive 作為源的權限](#)
- [使用 HDFS 作為來源的權限](#)
- [使用 HDFS 作為目標的權限](#)
- [連接到阿帕奇 Hadoop 作為源](#)
- [連接到您的來源蜂巢和 HDFS 服務](#)
- [連接到亞馬遜 EMR 作為目標](#)

使用阿帕奇 Hadoop 作為源的先決條件

需要以下先決條件才能使用 AWS SCT CLI 連接到阿帕奇的 Hadoop。

- 建立 Amazon S3 儲存貯體以在遷移期間存放資料。然後，您可以將資料複製到亞馬遜 EMR HDFS，或使用 Amazon S3 做為 Hadoop 工作負載的資料儲存庫。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[建立儲存貯體](#)。
- 使用AmazonS3FullAccess政策建立 AWS Identity and Access Management (IAM) 角色。AWS SCT使用此 IAM 角色存取您的 Amazon S3 儲存貯體。
- 記下您的AWS密鑰和AWS秘密訪問密鑰。如需有關AWS存取金鑰的詳細資訊，請參閱 IAM 使用者指南中的[管理存取金鑰](#)。
- 建立和設定目標亞馬遜 EMR 叢集。如需詳細資訊，請參閱[Amazon EMR 管理指南中的開始使用 Amazon EMR](#)。
- 在源阿帕奇 Hadoop 集群上安裝該distcp實用程序。此外，請在目標 Amazon EMR 叢集上安裝s3-dist-cp公用程式。請確定您的資料庫使用者具有執行這些公用程式的權限。
- 將來源 Hadoop 叢集中的core-site.xml檔案設定為使用 s3a 通訊協定。若要這樣做，請將fs.s3a.aws.credentials.provider參數設定為下列其中一個值。
 - 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

您可以將下面的代碼示例添加到core-site.xml文件中。

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

上述範例顯示上述選項清單中的四個選項之一。如果您未在core-site.xml檔案中設定fs.s3a.aws.credentials.provider參數，則會自動AWS SCT選擇提供者。

使用 Hive 作為源的權限

Hive 來源使用者所需的權限如下：

- READ存取來源資料夾和來源 Amazon S3 儲存貯體
- READ+WRITE存取中繼和目標 Amazon S3 儲存貯體

若要提高移轉速度，建議您針對 ACID 交易來源表格執行壓縮。

Amazon EMR Hive 目標使用者所需的許可如下：

- READ訪問目標亞馬遜 S3 存儲桶
- READ+WRITE訪問中間亞馬遜 S3 存儲桶
- READ+WRITE訪問目標 HDFS 文件夾

使用 HDFS 作為來源的權限

HDFS 作為源所需的權限如下：

- EXECUTE (針對NameNode)
- EXECUTE+READ包含在移轉專案中的所有來源資料夾和檔案
- READ+WRITE在遷移NameNode到 Amazon S3 之前執行 Spark 任務和存放檔案的tmp目錄

在 HDFS 中，所有操作都需要遍歷訪問。遍歷存取需要路徑所有現有元件的EXECUTE權限，但最終路徑元件除外。例如，對於存取任何作業/foo/bar/baz，您的使用者必須擁有//foo、和的EXECUTE權限/foo/bar。

下列程式碼範例示範如何授與EXECUTE+READ來源資料夾和檔案的READ+WRITE權限，以及tmp目錄的權限。

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

使用 HDFS 作為目標的權限

亞馬遜 EMR HDFS 作為目標所需的許可如下：

- EXECUTE用於目NameNode標亞馬遜 EMR 集群
- READ+WRITE對於目標 HDFS 文件夾，您將在遷移後存儲數據

連接到阿帕奇 Hadoop 作為源

您可以使用阿帕奇 Hadoop 作為 1.0.670 或更高AWS SCT版本的源。您只能在AWS SCT命令列界面 (CLI) 中將 Hadoop 叢集遷移到亞馬遜 EMR。在開始之前，請先熟悉的指令行介面。AWS SCT如需詳細資訊，請參閱[AWS SCT CLI 參考資料](#)。

要連接到阿帕奇的 Hadoop 在 CLI AWS SCT

1. 建立新的 AWS SCT CLI 指令碼或編輯現有案例範本。例如，您可以下載和編輯HadoopMigrationTemplate.scts範本。如需詳細資訊，請參閱[取得 CLI 案例](#)。
2. 設定AWS SCT應用程式設定，例如驅動程式位置和記錄檔資料夾。

下載所需的 JDBC 驅動程式，並指定儲存檔案的位置。如需詳細資訊，請參閱[下載所需資料庫驅動程式](#)。

下列程式碼範例會示範如何將路徑新增至 Apache Hive 驅動程式。執行此程式碼範例之後，將記錄檔AWS SCT儲存在資c:\sct料夾中。

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

您可以在 Windows 中使用這個範例和下列範例。

3. 創建一個新AWS SCT項目。

下列程式碼範例會在c:\sct資料夾中建立hadoop_emr專案。

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

4. 將您的源 Hadoop 集群添加到項目中。

使用該AddSourceCluster命令連接到源 Hadoop 集群。請確定您提供下列必要參數的值：namehost、port、和user。其他參數是可選的。

下列程式碼範例會新增來源 Hadoop 叢集。此範例會設定HADOOP_SOURCE為來源叢集的名稱。使用此對象名稱添加蜂巢和 HDFS 服務的項目，並創建映射規則。

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

在上述範例中，請將 *hadoop_address* 取代為 Hadoop 叢集的 IP 位址。如果需要，請設定連接埠選項的值。接下來，用您的 *Hadoop ##* 的名稱和此用戶的 *#### hadoop_user* 和密碼。針對 *#\ ##*，請輸入來源 Hadoop 叢集之 PEM 檔案的名稱和路徑。

5. 儲存您的 CLI 指令碼。接下來，添加您的蜂巢和 HDFS 服務的連接信息。

連接到您的來源蜂巢和 HDFS 服務

您可以使用 AWS SCT CLI 連接到源配置單元和 HDFS 服務。要連接到阿帕奇蜂房，請使用配置單元 JDBC 驅動程序 2.3.4 或更高版本。如需詳細資訊，請參閱[下載所需資料庫驅動程式](#)。

AWS SCT使用hadoop叢集使用者連線至 Apache Hive。若要這麼做，請使用AddSourceClusterHive和AddSourceClusterHDFS指令。您可以使用下列其中一種方法。

- 建立新的 SSH 通道。

針對 createTunnel，輸入 **true**。對於host，請輸入來源 Hive 或 HDFS 服務的內部 IP 位址。對於port，輸入您的蜂巢或 HDFS 服務的服務端口。

接下來，輸入您的蜂巢或 HDFS 憑據user和password. 如需 SSH 通道的詳細資訊，請參閱[《Amazon EMR 管理指南》中的使用本機連接埠轉送設定到主節點的 SSH 通道](#)。

- 使用現有的 SSH 通道。

針對 `host`，輸入 `localhost`。對於 `port`，從 SSH 通道參數輸入本機連接埠。

- 直接連接到您的蜂巢和 HDFS 服務。

對於 `host`，請輸入來源 Hive 或 HDFS 服務的 IP 位址或主機名稱。對於 `port`，輸入您的蜂巢或 HDFS 服務的服務端口。接下來，輸入您的蜂巢或 HDFS 憑據 `user` 和 `password`。

若要在 CLI 中連接到配置單元和 HDFS AWS SCT

1. 開啟包含來源 Hadoop 叢集的連線資訊的 CLI 指令碼。請確定您使用您在上一個步驟中定義的 Hadoop 叢集的名稱。
2. 將您的源 Hive 服務添加到項目中。

使用命 `AddSourceClusterHive` 令來連接源 Hive 服務。請確定您提供下列必要參數的值：`userpassword`、`cluster`、`name`、和 `port`。其他參數是可選的。

下列程式碼範例會建立與 Hive 服務搭配使用的通道。AWS SCT 此源 Hive 服務運行在同一台 PC 上 AWS SCT。此範例使用上一個範例中的 `HADOOP_SOURCE` 來源叢集。

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

下列程式碼範例會連線至您的 Hive 服務，而不需要通道。

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

```
-password: 'hive_password'  
/  

```

在前面的例子中，將 *hive_user* # *hive_password* 替換為您的配置單元用戶的名稱和此用戶的密碼。

接下來，用源 *Hadoop ### NameNode IP ##*和*#####*藏地址和隱藏端口。

對於 *hive_remote_address*，您可以使用默認值127.0.0.1或源配置單元服務的 NameNode IP 地址。

3. 將您的源 HDFS 服務添加到項目中。

使用AddSourceClusterHDFS命令連接源 HDFS 服務。請確定您提供下列必要參數的值：userpassword、cluster、name、和port。其他參數是可選的。

請確定您的使用者具有從來源 HDFS 服務遷移資料所需的權限。如需詳細資訊，請參閱[使用 Hive 作為源的權限](#)。

下面的代碼示例創建一個隧道與您的 Apache HDFS 服務的工作。AWS SCT此範例使用您之前建立的HADOOP_SOURCE來源叢集。

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

```

下面的代碼連接到你的 Apache HDFS 服務沒有隧道。

```
AddSourceClusterHDFS  
-cluster: 'HADOOP_SOURCE'  
-name: 'HDFS_SOURCE'  
-host: 'hdfs_address'  
-port: 'hdfs_port'  

```

```
-user: 'hdfs_user'  
-password: 'hdfs_password'  
/
```

在前面的例子中，替換 *hdfs_user* 和 *hdfs_password* 與您的 HDFS 用戶的名稱和該用戶的密碼。

接下來，用您的源 Hadoop 集群的 NameNode IP ##### *hdfs_port* 和 *hdfs_port*。

對於 *hdfs_remote_address*，您可以使用默認值 127.0.0.1 或源配置單元服務的 NameNode IP 地址。

4. 儲存您的 CLI 指令碼。接下來，新增目標 Amazon EMR 叢集的連線資訊和遷移命令。

連接到亞馬遜 EMR 作為目標

您可以使用 AWS SCT CLI 連線到目標亞馬遜 EMR 叢集。若要這麼做，您可以授權輸入流量並使用 SSH。在這種情況下，AWS SCT 具有與 Amazon EMR 叢集搭配使用的所有必要許可。[如需詳細資訊，請參閱《Amazon EMR 管理指南》中的〈使用 SSH 連線和連線到主節點之前〉。](#)

AWS SCT 透過 Hadoop 叢集使用者連線至 Amazon EMR Hive。要連接到亞馬遜 EMR 配置單元，請使用蜂巢 JDBC 驅動程序版本 2.6.2.1002 或更高版本。如需詳細資訊，請參閱[下載所需資料庫驅動程式](#)。

若要在 CLI 中連線到亞馬遜 EMR AWS SCT

1. 開啟包含來源 Hadoop 叢集的連線資訊的 CLI 指令碼。將目標亞馬遜 EMR 登入資料新增到此檔案中。
2. 將您的目標 Amazon EMR 叢集新增至專案。

下列程式碼範例會新增目標 Amazon EMR 叢集。此範例會設定 HADOOP_TARGET 為目標叢集的名稱。使用此物件名稱將您的 Hive 和 HDFS 服務以及 Amazon S3 儲存貯體資料夾新增至專案，並建立映射規則。

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

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

在前面的範例中，輸入您的AWS資源名稱和 Amazon EMR 連線資訊。這包括您的 Amazon EMR 叢集的 IP 位址、AWS存取金鑰、AWS秘密存取金鑰和 Amazon S3 儲存貯體。如果需要，請配置端口變量的值。接下來，將 *emr_user # emr_password* 替換為您的亞馬遜 EMR 用戶的名稱和該用戶的密碼。對於 *##\ ##*，請輸入目標 Amazon EMR 叢集之 PEM 檔案的名稱和路徑。如需詳細資訊，請參閱[下載 EMR 叢集存取的 PEM 檔案](#)。

3. 將您的目標 Amazon S3 儲存貯體新增至專案。

下列程式碼範例會新增目標 Amazon S3 儲存貯體。此範例使用您之前建立的HADOOP_TARGET叢集。

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

在上述範例中，輸入您的AWS存取金鑰、AWS秘密存取金鑰和 Amazon S3 儲存貯體。

4. 將您的目標 Hive 服務添加到項目中。

下列程式碼範例會建立與目標 Hive 服務搭配使用的通道。AWS SCT此範例使用您之前建立的HADOOP_TARGET目標叢集。

```
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'
```

/

在前面的例子中，將 *hive_user* # *hive_password* 替換為您的配置單元用戶的名稱和此用戶的密碼。

接下來，將 *hive_address* 替換為默認值 127.0.0.1 或目標配置單元服務的 NameNode IP 地址。接下來，將 *hive_port* 替換為目標配置單元服務的端口。

5. 將您的目標 HDFS 服務添加到項目中。

下面的代碼示例創建一個隧道與您的 Apache HDFS 服務的工作。AWS SCT 此範例使用您之前建立的 HADOOP_TARGET 目標叢集。

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

/

在前面的例子中，替換 *hdfs_user* 和 *hdfs_password* 與您的 HDFS 用戶的名稱和該用戶的密碼。

接下來，替換 *hdfs_port* ##### HDFS 服務的私有 IP 地址和端口。NameNode

6. 儲存您的 CLI 指令碼。接下來，新增對應規則和移轉指令。如需詳細資訊，請參閱 [遷移阿帕奇 Hadoop 到亞馬遜 EMR](#)。

使用阿帕奇奧西作為來源 AWS SCT

您可以使用命AWS SCT令列介面 (CLI) 將 Apache 工作流程轉換為 AWS Step Functions 將 Apache Hadoop 工作負載移轉至 Amazon EMR 之後，您可以使用中的原生服務AWS 雲端來協調您的任務。如需詳細資訊，請參閱[使用阿帕奇 Hadoop 作為源](#)。

AWS SCT將您的 Oozie 工作流程轉換為AWS Step Functions並用AWS Lambda來模擬AWS Step Functions不支援的功能。此外，AWS SCT將您的 Oozie 工作屬性轉換為AWS Systems Manager。

若要轉換 Apache 工作流程，請確定您使用的是 1.0.671 或更高AWS SCT版本。此外，請熟悉的指令行介面。AWS SCT如需詳細資訊，請參閱[AWS SCT CLI 參考資料](#)。

使用阿帕奇作為源代碼的先決條件

要使用 AWS SCT CLI 連接到阿帕奇奧西需要以下先決條件。

- 建立 Amazon S3 儲存貯體來存放狀態機器的定義。您可以使用這些定義來設定您的狀態機器。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[建立儲存貯體](#)。
- 使用AmazonS3FullAccess政策建立 AWS Identity and Access Management (IAM) 角色。AWS SCT使用此 IAM 角色存取您的 Amazon S3 儲存貯體。
- 記下您的AWS密鑰和AWS秘密訪問密鑰。如需有關AWS存取金鑰的詳細資訊，請參閱 IAM 使用者指南中的[管理存取金鑰](#)。
- 將您的AWS登入資料和 Amazon S3 儲存貯體的相關資訊存放在全域應用程式設定的AWS服務設定檔中。然後，AWS SCT使用此AWS服務設定檔來處理您的AWS資源。如需詳細資訊，請參閱[將AWS服務設定檔儲存在AWS SCT](#)。

若要使用來源 Apache Oozie 工作流程，AWS SCT需要來源檔案的特定結構。您的每個應用程式資料夾都必須包含該job.properties檔案。此檔案包含工作屬性的索引鍵值組。此外，每個應用程式資料夾都必須包含該workflow.xml檔案。此檔案描述工作流程的動作節點和控制流程節點。

連接到阿帕奇奧西作為源

使用以下過程來連接到您的 Apache Oozie 源文件。

若要在 CLI 中連接到阿帕奇 AWS SCT

1. 建立新的 AWS SCT CLI 指令碼或編輯現有案例範本。例如，您可以下載和編輯OozieConversionTemplate.scts範本。如需詳細資訊，請參閱[取得 CLI 案例](#)。

2. 設定AWS SCT應用程式設定。

下列程式碼範例會儲存應用程式設定，並允許在專案中儲存密碼。您可以在其他專案中使用這些儲存的設定。

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

3. 創建一個新AWS SCT項目。

下列程式碼範例會在c:\sct資料夾中建立oozie專案。

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

4. 使用該AddSource命令將包含源 Apache Oozie 文件的文件夾添加到項目中。請確定您使用vendor參數的APACHE_OOZIE值。此外，請提供下列必要參數的值：name和mappingsFolder。

下列程式碼範例會將 Apache Oozie 新增為AWS SCT專案中的原始碼。此範例會以名稱建立來源物件OOZIE。使用此物件名稱可新增對應規則。執行此程式碼範例之後，AWS SCT會使用c:\oozie資料夾在專案中載入來源檔案。

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

您可以在 Windows 中使用這個範例和下列範例。

5. 使用ConnectSource命令連接到您的源阿帕奇 Ozie 文件。使用您在上一個步驟中定義的來源物件名稱。

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

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

6. 儲存您的 CLI 指令碼。接下來，新增AWS Step Functions服務的連線資訊。

在擴充功能套件中使用AWS Lambda函式的權限

對於AWS Step Functions不支持的源函數，AWS SCT創建一個擴展包。此擴展包包含模擬源AWS Lambda功能的功能。

若要使用此擴充套件，請建立具有下列權限的 AWS Identity and Access Management (IAM) 角色。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "lambda",  
      "Effect": "Allow",  
      "Action": [  
        "lambda:InvokeFunction"  
      ],  
      "Resource": [  
        "arn:aws:lambda*:498160209112:function:LoadParameterInitialState:*",  
        "arn:aws:lambda*:498160209112:function:EvaluateJSPELExpressions:*"  
      ]  
    },  
    {  
      "Sid": "emr",  
      "Effect": "Allow",  
      "Action": [  
        "elasticmapreduce:DescribeStep",  
        "elasticmapreduce:AddJobFlowSteps"  
      ],  
      "Resource": [  
        "arn:aws:elasticmapreduce*:498160209112:cluster/*"  
      ]  
    },  
    {  
      "Sid": "s3",  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject"  
      ],  
    }  
  ]  
}
```

```

        "Resource": [
            "arn:aws:s3:::*/*"
        ]
    }
]
}

```

若要套用擴充套件，AWS SCT需要具有下列許可的 IAM 角色。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:CreateRole",
        "iam:TagRole",
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy",
        "iam>DeleteRole",
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::ACCOUNT_NUMBER:role/sct/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies"
      ],
      "Resource": [
        "arn:aws:iam::ACCOUNT_NUMBER:role/lambda_LoadParameterInitialStateRole",
        "arn:aws:iam::ACCOUNT_NUMBER:role/lambda_EvaluateJSPELExpressionsRole",
        "arn:aws:iam::ACCOUNT_NUMBER:role/stepFunctions_MigratedOozieWorkflowRole"
      ]
    }
  ]
}

```

```
    "Effect": "Allow",
    "Action": [
        "lambda:GetFunction",
        "lambda:CreateFunction",
        "lambda:UpdateFunctionCode",
        "lambda:DeleteFunction"
    ],
    "Resource": [
        "arn:aws:lambda:*:ACCOUNT_NUMBER:function:LoadParameterInitialState",
        "arn:aws:lambda:*:ACCOUNT_NUMBER:function:EvaluateJSPELExpressions"
    ]
}
]
```

連接到AWS Step Functions作為目標

使用下列程序來連線AWS Step Functions為目標。

若要在 AWS SCT CLI AWS Step Functions 中連線到

1. 打開您的 CLI 腳本，其中包括您的 Apache Oozie 源文件的連接信息。
2. 使用AddTarget指令在AWS SCT專案中加入有關移轉目標的資訊。請確定您使用vendor參數的STEP_FUNCTIONS值。此外，請提供下列必要參數的值：name和profile。

下列程式碼範例會新增AWS Step Functions為AWS SCT專案中的原始碼。此範例會以名稱建立目標物件AWS_STEP_FUNCTIONS。建立對應規則時，請使用此物件名稱。此外，此範例會使用您在先決條件步驟中建立的AWS SCT服務設定檔。請確定您已將設定####取代為設定檔名稱。

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

如果您不使用AWS服務設定檔，請確定您提供下列必要參數的值：accessKeysecretKey、awsRegion、和s3Path。使用這些參數指定您的AWS秘密存取金鑰AWS 區域、AWS秘密金鑰和 Amazon S3 儲存貯體的路徑。

3. AWS Step Functions使用ConnectTarget指令連接到。使用您在上一個步驟中定義的目標物件的名稱。

下列程式碼範例會使用您的AWS服務設定檔連線至AWS_STEP_FUNCTIONS目標物件。請確定您已將設定####取代為設定檔名稱。

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

4. 儲存您的 CLI 指令碼。接下來，新增對應規則和移轉指令。如需詳細資訊，請參閱[將阿帕奇奧西轉換為 AWS Step Functions](#)。

使用 Azure SQL 資料庫做為下列項目的來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Azure SQL 資料庫轉換為下列目標：

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

主題

- [將 Azure SQL 資料庫作為來源的權限](#)
- [連線至 Azure SQL 資料庫做為來源](#)

將 Azure SQL 資料庫作為來源的權限

Azure SQL 資料庫做為來源所需的權限如下：

- VIEW DEFINITION
- VIEW DATABASE STATE

對您要轉換其結構描述的每個資料庫重複授予。

目標 MySQL 和 PostgreSQL 資料庫所需的權限將在下列各節中說明。

- [將 MySQL 作為目標資料庫的權限](#)
- [將 PostgreSQL 作為目標資料庫的權限](#)

連線至 Azure SQL 資料庫做為來源

使用下列程序，使用連線到 Azure SQL 資料庫來源資料庫AWS Schema Conversion Tool。

若要連線至 Azure SQL 資料庫來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 Azure SQL 資料庫，然後選擇 [下一步]。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：

1. 在「機AWS密」中，選擇密碼的名稱。
2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Azure SQL 資料庫來源資料庫連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入您來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。
Database (資料庫)	輸入要連線的資料庫名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。

AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。

參數	動作
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

使用適用於 z/OS 的 IBM Db2 作為下列項目的來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 IBM Db2 for z/OS 轉換為下列目標。

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

做為來源資料庫的 Db2 先決條件

適用於 z/OS 版本 12 的 IBM Db2 功能層級 100 資料庫版本不支援 IBM Db2 (適用於 z/OS 版本 12) 的大多數新功能。此資料庫版本提供 Db2 版本 11 的後援支援，以及與 Db2 版本 11 共用資料。若要避免轉換 Db2 第 11 版不受支援的功能，建議您使用 IBM Db2 for z/OS 資料庫函數等級 500 或更高版本作為的來源。AWS SCT

您可以使用下列程式碼範例來檢查來源 IBM Db2 for z/OS 資料庫的版本。

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

確保此代碼返回版本DSN12015或更高版本。

您可以使用下列程式碼範例，檢查來源 IBM Db2 for z/OS 資料庫中APPLICATION COMPATIBILITY特殊暫存器的值。

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

確保此代碼返回版本V12R1M500或更高版本。

做為來源資料庫的 Db2 (z/OS) 的權限

連線至 Db2 for z/OS 資料庫以及讀取系統目錄和表格所需的權限如下：

- 在系統位置上選取
- 在系統檢查上選取
- 在系統列上選擇
- 在系統數據庫上選擇
- 在系統資料類型上選取
- 在系統上選擇 1
- 在系統外匯金鑰上選取
- 在系統上選擇
- 在系統上選取系統共用
- 在系統座標上選取
- 在系統關鍵目標上選取
- 在系統對象上選擇
- 在系統套件上選取
- 在系統上選取
- 在系統表上選取
- 在系統例程中選擇
- 在系統序列上選擇
- 在系統序列上選擇
- 在系統同義字上選擇
- 在系統上選擇
- 在系統對象上選擇
- 在系統空間上選擇
- 在系統的系統上選擇
- 在系統變量上選擇
- 在系統視圖上選擇

若要將 Db2 for z/OS 資料表轉換為 PostgreSQL 分區資料表，請使用公用程式收集資料庫中表格空間和表格的統計資料，如下RUNSTATS所示。

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

在上述範例中，將*YOURDB*預留位置取代為來源資料庫的名稱。

連接至 Db2，用於 z/OS 作為來源

使用下列程序來連線到您的 Db2 for z/OS 來源資料庫。AWS SCT

若要連線至適用於 z/OS 的 IBM Db2 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 針對 z/OS 選擇 Db2，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 IBM Db2 for z/OS 來源資料庫連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。

參數	動作
位置	輸入您要存取的 Db2 位置的唯一名稱。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設 AWS SCT 不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>如果您要使用安全通訊端層 (SSL) 連線到資料庫，請選擇此選項。在 SSL 索引標籤上提供下列適用的其他資訊：</p> <ul style="list-style-type: none"> 信任存放區：包含憑證的信任存放區的位置。要使此位置顯示在此處，請確保將其添加到「全局」設置中。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
適用於 z/OS 驅動程式路徑的 Db2	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否 AWS SCT 可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

將 MySQL 作為目標資料庫的權限

MySQL 作為目標所需的權限如下：

- 建立於 *。*
- 在 * 上更改。*
- 下降 *。*

- * 上的索引。 *
- 參考文獻 *。 *
- SELECT ON *.*
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.*
- 觸發器於 *。 *
- 在 * 上創建例程。 *
- 改變例程 *。 *
- 執行日期 *。 *
- SELECT ON mysql.proc
- 插入, 在 AWS_DB2ZO_EXT 上進行更新。 *
- 插入、更新、刪除。 *
- 在檔案資料上建立暫存資料表。 *

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

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

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，用安全#####。

若要使用適用於 MySQL 的亞馬遜 RDS 做為目標，請將 `log_bin_trust_function_creators` 參數設定為 `true`，並將設定 `character_set_server` 為 `latin1`。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

若要使用極光 MySQL 作為目標，請將 `log_bin_trust_function_creators` 參數設定為 `true`，並設定 `character_set_server` 為 `latin1`。此外，將 `lower_case_table_names` 參數設定為 `true`。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

將 PostgreSQL 作為目標資料庫的權限

若要使用 PostgreSQL 做為目標，AWS SCT 需要權限。CREATE ON DATABASE 請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用適用於 PostgreSQL 的亞馬遜 RDS 做為目標，AWS SCT 需要特權。 `rds_superuser`

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為 `"$user"`， `public_synonyms`， `public`。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在上述範例中，將 `user_name` 取代為您的使用者名稱。然後，將 `db_name` 替換為目標數據庫的名稱。最後，用安全的 `#####`。

在 PostgreSQL 中，只有結構描述擁有者或一個 `superuser` 可以刪除結構描述。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的使用者將不同的結構描述轉換並套用至目標資料庫時，您可能會在無法卸除結構描述時 AWS SCT 收到錯誤訊息。若要避免此錯誤訊息，請使用 `superuser` 角色。

適用於 z/OS 到 PostgreSQL 的 Db2 轉換設定

若要將 z/OS 的 Db2 轉換設定編輯為 PostgreSQL，請選擇 [設定]，然後選擇 [轉換設定]。從上方的清單中選擇 Db2 表示 z/OS，然後選擇 Db2 代表 z/OS — PostgreSQL 或 Db2 (表示 z/OS) — 亞馬遜極光 (與 PostgreSQL 相容)。AWS SCT 顯示 IBM Db2 (適用於 z/OS) 到 PostgreSQL 轉換的所有可用設定。

適用於 z/OS 到 PostgreSQL 的 Db2 轉換設定 AWS SCT 包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 為目標數據庫中的約束生成唯一的名稱。

在 PostgreSQL 中，您使用的所有約束名稱都必須是唯一的。AWS SCT可以通過將帶有表名的前綴添加到約束的名稱中，為轉換後的代碼中的約束生成唯一名稱。若要確保為條件約束AWS SCT產生唯一的名稱，請選取「產生限制條件的唯一名稱」。

- 在轉換後的代碼中保留 DML 語句中列名，表達式和子句的格式。

AWS SCT可以將 DML 語句中的列名，表達式和子句的佈局保持在與源代碼相似的位置和順序。若要這麼做，請在 DML 陳述式中保留欄名稱、運算式和子句的格式選取是。

- 從轉換範圍中排除表格分割區。

AWS SCT可以在轉換過程中跳過源表的所有分區。若要這麼做，請選取 [從轉換範圍排除表格分割區]。

- 若要針對依成長分割的資料表使用自動磁碟分割。

對於資料移轉，AWS SCT可以自動分割大於指定大小的所有表格。若要使用此選項，請選取強制分割大於的表格，然後輸入以 GB 為單位的表格大小。接下來，輸入分區的數量。AWS SCT開啟此選項時，會考慮來源資料庫的直接存取儲存裝置 (DASD) 大小。

AWS SCT可以自動確定分區的數量。若要這樣做，請選取按比例增加分割區的數目，然後輸入最大分割區數目。

- 要返回動態結果集作為 refcursor 數據類型的值的數組。

AWS SCT可以將返回動態結果集的源過程轉換為具有開放反射器數組作為附加輸出參數的過程。若要這樣做，請選取「使用反射器陣列」以傳回所有動態結果集。

- 要指定用於將日期和時間值轉換為字符串表示的標準。

AWS SCT可以使用其中一種支援的產業格式，將日期和時間值轉換為字串表示法。若要這樣做，請選取「使用日期值的字串表示」或「使用時間值的字串表示」。接下來，選擇以下標準之一。

- 國際標準組織 (ISO)
- IBM 歐洲標準 (歐元)

- IBM 美國標準 (美國)
- 日本工業標準基督教時代 (JIS)

使用 IBM Db2 LUW 作為下列項目的來源 AWS SCT

您可AWS SCT以使用將結構描述、SQL 語言中的程式碼物件和應用程式程式碼從 IBM Db2 (Linux、Unix 和視窗) 轉換為下列目標。

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

AWS SCT支援做為來源 Db2 LUW 版本 9.1、9.5、9.7、10.1、11.1 和 11.5 版本。

Db2 LUW 作為來源的權限

連線至 Db2 LUW 資料庫、檢查可用權限以及讀取來源的結構描述資料所需的權限如下：

- 建立連線所需要的權限：
 - 在資料庫上連線
- 執行 SQL 陳述式所需要的權限：
 - 在無效的套件上執行
- 取得執行個體層級資訊所需要的權限：
 - 在函數系統上執行. ENV_GET_INST_INFO
 - 在系統資訊上進行選擇
 - 在系統資訊上進行選取
- 檢查透過角色、群組和機構授予權限所需要的權限：
 - 在函數系統上執行。列表權限 _ 授權
 - 在函數上執行系統列表組
 - 在函數上執行系統列表
 - 選擇系統權限

- 系統目錄和表格所需要的權限：
 - 在系統屬性上選擇
 - 在系統分散檢查上選擇
 - 在系統散點屬性上選擇
 - 在系統分佈列上選擇
 - 在系統分散. 數據表達式上選擇
 - 選擇系統分散數據
 - 在系統分散數據上選擇
 - 在系統分散. 數據類型上選擇
 - 在系統層次結構上選取
 - 選擇系統. 索引共用
 - 在系統分散. 索引上選擇
 - 選擇系統分區索引
 - 在系統上選擇密鑰共用
 - 在系統散射. 模塊對象上選擇
 - 在系統上選擇模塊
 - 選擇系統暱稱
 - 在系統分散上選擇
 - 在系統分散上選擇
 - 在系統分析上選擇
 - 在系統程序上選擇
 - 在系統分散. 行字段上選擇
 - 在系統上選擇. 模式
 - 選擇系統散射序列
 - 在系統分散上選擇
 - 在系統分散表上選擇
 - 在系統散點觸發器上選擇
 - 在系統散點上選擇變量
 - 在系統分散變量上選擇
- 在系統視圖上選擇

- 在系統上選擇 1
- 若要執行 SQL 陳述式，使用者帳戶需要能使用資料庫內已啟用的至少一個工作負載的權限。如果沒有指派工作負載給使用者，請確定使用者能存取預設的使用者工作負載：
 - 工作負載系統預設使用者工作負載的使用量

若要執行查詢，您需要建立頁面大小 8K、16K 和 32K 的系統臨時資料表空間 (如果這些不存在)。若要建立臨時資料表空間，請執行下列指令碼。

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

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_8K
  PAGESIZE 8192
  BUFFERPOOL BP8K;

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

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP16K
  PAGESIZE 16384
  BUFFERPOOL BP16K;

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

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

連接至 Db2 LUW 做為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Db2 LUW 來源資料庫。

連接到 Db2 LUW 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇「Db2 LUW」，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 IBM Db2 LUW 來源資料庫連線資訊，請遵循下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 Db2 LUW 資料庫的名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
Use SSL (使用 SSL)	如果您要使用安全通訊端層 (SSL) 連線到資料庫，請選擇此選項。在 SSL 索引標籤上提供下列適用的其他資訊：

參數	動作
	<ul style="list-style-type: none"> 信任存放區：包含憑證的信任存放區的位置。要使此位置顯示在此處，請確保將其添加到「全局」設置中。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
Db2 LUW 驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

將 Db2 LUW 轉換為亞馬遜 RDS，適用於 PostgreSQL 或亞馬遜極光兼容版本

當您將 IBM Db2 LUW 遷移至 PostgreSQL 時，AWS SCT 可以轉換與 Db2 LUW 一起使用的各種觸發陳述式。這些觸發陳述式包含以下項目：

- 觸發事件 — INSERT、DELETE 和 UPDATE 觸發器事件會指定每當事件套用至主旨資料表或主旨檢視時，就會執行觸發的動作。您可以指定 INSERT、刪除和 UPDATE 事件的任何組合，但您只能指定每個事件一次。AWS SCT 支援單一和多個觸發事件。對於事件，PostgreSQL 有幾乎相同的功能。
- 欄的事件 — 您可以從基底資料表指定欄名稱。觸發條件只會被欄名稱清單中所識別欄的更新啟用。PostgreSQL 有相同的功能。
- 陳述式觸發程序 — 這些指定觸發的動作僅套用於整個陳述式一次。您無法為 BEFORE 觸發條件或 INSTEAD OF 觸發條件指定此觸發精細程度類型。若指定，會啟用 UPDATE 或 DELETE 觸發條件，即使沒有任何列受到影響。PostgreSQL 也有此功能，且陳述式觸發條件的觸發宣告與 PostgreSQL 和 Db2 LUW 相同。
- 參考子句 — 指定轉移變數的關聯名稱，以及轉移表的表格名稱。相互關聯名稱識別受觸發 SQL 操作影響的一組列中的特定列。資料表名稱識別一整組受影響的列。受觸發 SQL 操作影響的每一列

可透過具有指定之相互關聯名稱的合格列，用於觸發的動作。PostgreSQL 不支援此功能，只使用 NEW 或 OLD 相互關聯名稱。

- 而不是觸發器-AWS SCT 支持這些。

將資料分割資料表轉換為第 10 版的分區資料表

AWS SCT 可將 Db2 LUW 資料表轉換至 PostgreSQL 10 中的分割資料表。在將 Db2 LUW 分割資料表轉換到 PostgreSQL 時，有多種限制：

- 您可在 Db2 LUW 中以 NULL 欄建立分割資料表，並可以指定一個分割區來存放 NULL 值。但是，PostgreSQL 不支援 RANGE 分割的 NULL 值。
- Db2 LUW 可以使用 INCLUSIVE 或 EXCLUSIVE 子句以設定範圍邊界值。PostgreSQL 僅支援在開始邊界的 INCLUSIVE 和結束邊界的 EXCLUSIVE。轉換後的分割區名稱格式為 `<original_table_name>_<original_partition_name>`。
- 您可在 Db2 LUW 中建立分割資料表的主要或唯一索引鍵。PostgreSQL 要求您為每個分割區直接建立主要或唯一索引鍵。必須從父資料表中移除主要或唯一索引鍵限制。轉換後的金鑰名稱格式為 `<original_key_name>_<original_partition_name>`。
- 您可在 Db2 LUW 中從分割資料表建立外部索引鍵限制，以及對分割資料表建立外部索引鍵限制。但是，PostgreSQL 不支援在分割資料表中的外部索引鍵參考。PostgreSQL 也不支援從一個分割資料表參考另一個資料表的外部索引鍵。
- 您可以在 Db2 LUW 中的分割資料表上建立索引。但是，PostgreSQL 要求您為每個分割區直接建立索引。必須從父資料表移除索引。轉換後的索引名稱格式為 `<original_index_name>_<original_partition_name>`。
- 您必須在個別分割區上定義 row 觸發器，而非在分割資料表上定義。必須從父資料表移除觸發器。轉換後的觸發器名稱格式為 `<original_trigger_name>_<original_partition_name>`。

將 PostgreSQL 作為目標的權限

若要使用 PostgreSQL 做為目標，AWS SCT 需要權限。CREATE ON DATABASE 請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為 "\$user", public_synonyms, public。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，將 *db_name* 替換為目標數據庫的名稱。最後，用安全的#####。

在 PostgreSQL 中，只有結構描述擁有者或一個superuser可以刪除結構描述。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的使用者將不同的結構描述轉換並套用至目標資料庫時，您可能會在無法卸除結構描述時 AWS SCT收到錯誤訊息。若要避免此錯誤訊息，請使用superuser角色。

將 Db2 LUW 轉換為亞馬遜 RDS，適用於 MySQL 或亞馬遜極光 MySQL

當您將 IBM Db2 LUW 資料庫轉換為適用於 MySQL 的 RDS 或亞馬遜極光 MySQL 時，請注意以下事項。

將 MySQL 作為目標的權限

MySQL 作為目標所需的權限如下：

- 建立於 *。 *
- 在 * 上更改。 *
- 下降 *。 *
- * 上的索引。 *
- 參考文獻 *。 *
- SELECT ON *.*
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.*
- 觸發器於 *。 *
- 在 * 上創建例程。 *
- 改變例程 *。 *
- 執行日期 *。 *
- SELECT ON mysql.proc
- 插入，在 AWS_DB2_EXT 上進行更新。 *

- 插入、更新、刪除資料。 *
- 在檔案資料上建立暫存資料表。 *

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

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

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，用安全#####。

若要使用適用於 MySQL 的亞馬遜 RDS 或極光 MySQL 作為目標，請將 `lower_case_table_names` 參數設定為 1。此值意味著 MySQL 服務器處理不區分大小寫的表，索引，觸發器和數據庫等對象名稱的標識符。如果您已在目標執行個體中開啟二進位記錄功能，請將 `log_bin_trust_function_creators` 參數設定為 1。在這種情況下，您不需要使用 DETERMINISTIC、READS SQL DATA 或 NO SQL 特性來建立已儲存的函數。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

使用 MySQL 作為一個源代碼 AWS SCT

您可以使用 AWS SCT 將結構描述、資料庫程式碼物件和應用程式程式碼從 MySQL 轉換為下列目標：

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

如需詳細資訊，請參閱下列章節：

主題

- [MySQL 身分驗證作為 MySQL 身分驗證作為 MySQL](#)
- [以 MySQL 身分驗證作為 MySQL 身分驗證](#)
- [將 PostgreSQL 作為目標資料庫的權限](#)

MySQL 身分驗證作為 MySQL 身分驗證作為 MySQL

MySQL 作為源所需的權限如下：

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

以 MySQL 身分驗證作為 MySQL 身分驗證

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 MySQL 來源資料庫。

若要連線至 MySQL 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 MySQL，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 針對您的資料庫的名稱，輸入您的資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：
 - 若要使用來自 Secrets Manager 的資料庫認證，請遵循下列指示：
 1. 對於 AWS 機密，請選擇密碼的名稱。
 2. 選擇填入來自 Secrets Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自 Secrets Manager 之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 MySQL 來源資料庫連線資訊，請使用下列指示：

參數	動作
伺服器名稱	<p>輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 地址。</p> <p>您可以使用 IPv6 地址通訊協定，MySQL 用 IPv6 地址協定，使用 IPv6 地址協定，若要這麼做，請務必使用方括號來輸入 IP 地址，如下列範例所示。</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用密碼來連線至來源資料庫。為了防止公開來源資料庫的密碼，預設 AWS SCT 不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項，使用 Secure Sockets Layer (SSL) 來使用 SSL (SSL) 來身分驗證，來在 SSL 索引標籤上提供下列其他資訊 (如適用)：</p> <ul style="list-style-type: none"> • 需要 SSL：選擇此選項可僅透過 SSL 連線至伺服器。 <p>如果您選擇 Require SSL (需要 SSL)，這表示如果伺服器不支援 SSL，您就無法連接到伺服器。如果您不選擇 Require SSL (需要 SSL) 且伺服器不支援 SSL，您仍然可以不使用 SSL 而連接到伺服器。如需詳細資訊，請參閱設定 MySQL 以使用安全連線。</p> <ul style="list-style-type: none"> • 驗證伺服器憑證：選取此選項可使用信任存放區驗證伺服器憑證。 • 信任存放區：包含憑證的信任存放區的位置。

參數	動作
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。啟用此選項可讓您存放資料庫密碼，並快速連接到資料庫而無需輸入密碼。
MySQL 驅動路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「Connect」以連線到來源資料庫。

將 PostgreSQL 作為目標資料庫的權限

若要使用 PostgreSQL 做為目標，AWS SCT需要CREATE ON DATABASE權限。請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為"\$user", public_synonyms, public。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在上述範例中，使用您的使用###，使用您的使用者名稱。然後，將 *db_name* 替換為目標數據庫的名稱。最後，用安全#####的密碼。

在 PostgreSQL 中，只有結構描述擁有者或一個superuser可以刪除結構描述。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的使用者將不同的結構描述轉換並套用至目標資料庫時，您可能會在無法卸除結構描述時AWS SCT收到錯誤訊息。若要避免此錯誤訊息，請使用superuser角色。

使用 Oracle 資料庫作為下列項目的來源 AWS SCT

您可以使用AWS SCT將綱要、資料庫程式碼物件和應用程式程式碼從「Oracle 資料庫」轉換為下列目標：

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

當來源是 Oracle 資料庫時，可以在 PostgreSQL 資料庫中將註解轉換為適當的格式。AWS SCT可以轉換表格、檢視表和欄的註解。註解可以包含撇號；轉換 SQL 陳述式時，撇號會AWS SCT加倍，就像字串常值一樣。

如需詳細資訊，請參閱下列內容。

主題

- [Oracle 作為來源的權限](#)
- [連線至 Oracle 作為來源](#)
- [將甲骨文轉換為亞馬遜 RDS 以獲取 PostgreSQL 或亞馬遜極光](#)
- [將甲骨文轉換為亞馬遜 RDS 的 MySQL 或亞馬遜極光 MySQL](#)
- [將 Oracle 轉換到 Amazon RDS for Oracle](#)

Oracle 作為來源的權限

Oracle 作為來源所需的權限如下：

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- 選擇系統參數 \$

連線至 Oracle 作為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Oracle 來源資料庫。

若要連接到 Oracle 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 Oracle，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 對於 AWS 機密，請選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Oracle 來源資料庫連線資訊，請遵循下列指示：

參數	動作
類型	<p>選擇連接到您資料庫的連線類型。根據您的類型，提供以下額外資訊：</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• 伺服器名稱：來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。• 伺服器連接埠：用於連接到來源資料庫伺服器的連接埠。• Oracle SID：Oracle 系統 ID (SID)。若要尋找 Oracle SID，請提交以下查詢至您的 Oracle 資料庫： <pre>SELECT sys_context('userenv','instance_name') AS SID FROM dual;</pre> <ul style="list-style-type: none">• 服務名稱

參數	動作
	<ul style="list-style-type: none">• 伺服器名稱：您來源資料庫伺服器的 DNS 名稱或 IP 地址。 您可以使用 IPv6 位址通訊協定連線到來源 Oracle 資料庫。若要這麼做，請確定您使用方括號來輸入 IP 位址，如下列範例所示。 <div data-bbox="716 506 1507 583" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>• 伺服器連接埠：用於連接到來源資料庫伺服器的連接埠。<ul style="list-style-type: none">◦• 服務名稱：要連線的 Oracle 服務名稱。• TNS 別名<ul style="list-style-type: none">• TNS 檔案路徑：包含透明網路底層 (TNS) 名稱連線資訊之檔案的路徑。 選擇 TNS 檔案之後，AWS SCT 將檔案中的所有 Oracle 資料庫連線新增至 TNS 別名清單。 選擇此選項可連線至「Oracle 實際應用程式叢集 (RAC)」。• TNS 別名：此檔案中用來連線至來源資料庫的 TNS 別名。• TNS 連線識別碼<ul style="list-style-type: none">• TNS 連線識別碼：已註冊 TNS 連線資訊的識別碼。

參數	動作
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>第一次連線到 Oracle 資料庫時，請輸入 Oracle 驅動程式檔案 (ojdbc8.jar) 的路徑。您可從 http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html 下載檔案。請務必在免費的 Oracle 技術網路網站上註冊，以完成下載作業。AWS SCT 將選取的驅動程式用於任何未來的 Oracle 資料庫連線。您可以使用「全域設定」中的「驅動程式」標籤來修改動因路徑。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用密碼來連線至來源資料庫。為了防止公開來源資料庫的密碼，預設 AWS SCT 不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列其他資訊 (如適用)：</p> <ul style="list-style-type: none"> • SSL 驗證：選取此選項可使用憑證的 SSL 驗證在設定、全域設定、安全性中設定您的信任存放區和金鑰存放區。 • 信任存放區：要使用的信任存放區。 • 金鑰存放區：要使用的金鑰存放區。
儲存密碼	<p>AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。選擇此選項可儲存資料庫密碼，並快速連線至資料庫，而不需要輸入密碼。</p>
Oracle 驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您將驅動程式路徑儲存在全域專案設定中，則驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否 AWS SCT 可以連線到來源資料庫。

6. 選擇「連線」以連線到來源資料庫。

將甲骨文轉換為亞馬遜 RDS 以獲取 PostgreSQL 或亞馬遜極光

當您將甲骨文資料庫轉換為 RDS 版或亞馬遜極光 PostgreSQL 時，請注意以下事項。

主題

- [將 PostgreSQL 作為目標資料庫的權限](#)
- [甲骨文到 PostgreSQL 的轉換設置](#)
- [轉換甲骨文序列](#)
- [轉換 Oracle ROWID](#)
- [轉換甲骨文動態 SQL](#)
- [轉換 Oracle 分割區](#)

將 Oracle 系統物件轉換為 PostgreSQL 時，AWS SCT 會執行如下表所示的轉換。

Oracle 系統物件	描述	轉換後的 PostgreSQL 物件
V\$VERSION	顯示 Oracle 資料庫中核心程式庫元件的版本編號	aws_oracle_ext.v\$version
V\$INSTANCE	顯示目前執行個體狀態的檢視。	aws_oracle_ext.v\$instance

您可以使用 AWS SCT 將甲骨文 SQL* Plus 文件轉換為 psql，這是一個基於終端的前端到 PostgreSQL。如需詳細資訊，請參閱 [使用轉換應用程式 SQLAWS SCT](#)。

將 PostgreSQL 作為目標資料庫的權限

若要使用 PostgreSQL 做為目標，AWS SCT 需要權限。CREATE ON DATABASE 請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為 "\$user", public_synonyms, public。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，將 *db_name* 替換為目標數據庫的名稱。最後，用安全的#####。

若要使用適用於 PostgreSQL 的亞馬遜 RDS 做為目標，AWS SCT 需要特權。rds_superuser

在 PostgreSQL 中，只有結構描述擁有者或一個superuser可以刪除結構描述。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的使用者將不同的結構描述轉換並套用至目標資料庫時，您可能會在無法卸除結構描述時 AWS SCT 收到錯誤訊息。若要避免此錯誤訊息，請使用superuser角色。

甲骨文到 PostgreSQL 的轉換設置

若要編輯 Oracle 到 PostgreSQL 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上方的清單中選擇甲骨文，然後選擇甲骨文 — PostgreSQL。AWS SCT 顯示甲骨文到 PostgreSQL 轉換的所有可用設定。

甲骨文到 PostgreSQL 的轉換設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 允許AWS SCT將甲骨文具體化視觀表轉換為 PostgreSQL 上的表格或具體化視觀表。對於具體化視觀表轉換為，請選擇如何轉換來源具體化視觀表。
- 在原始程式碼 Oracle 程式碼包含TO_CHAR TO_DATE、和TO_NUMBER函數時使用 PostgreSQL 不支援的參數。默認情況下，AWS SCT模擬這些參數在轉換後的代碼的使用。

當您的原始程式碼只包含 PostgreSQL 支援的參數時，您可以使用原生 PostgreSQL TO_CHAR 和函數。TO_DATE TO_NUMBER在這種情況下，轉換後的代碼的工作速度更快。若只要包括這些參數，請選取下列值：

- 函數 TO_CHAR () 不使用甲骨文特定的格式化字符串
- 函數 TO_DATE () 不使用甲骨文特定的格式化字符串

- 函數 TO_NUMBER () 不使用甲骨文特定的格式化字符串
- 若要解決來源 Oracle 資料庫僅在資料類型的主索引鍵或外部索引鍵 NUMBER 資料行中儲存整數值時，AWS SCT 可以將這些資料行轉換為 BIGINT 資料類型。這種方法可以提高轉換後的代碼的性能。要採用此方法，請選擇將數字主/外鍵列轉換為 BIGINT 列。請確定您的來源不會在這些資料行中包含浮點值，以避免資料遺失。
- 跳過源代碼中停用的觸發器和約束。若要這麼做，請選擇忽略已停用的觸發程序和條件約
- 若要使用 AWS SCT 來轉換稱為動態 SQL 的字串變數。您的資料庫程式碼可以變更這些字串變數的值。若要確保 AWS SCT 永遠轉換此字串變數的最新值，請選取 [轉換在呼叫的常式中建立的動態 SQL 程式碼]。
- 為了解決 PostgreSQL 版本 10 及更早版本不支援程序。如果您或您的使用者不熟悉 PostgreSQL 中的使用程序，AWS SCT 可以將甲骨文程序轉換為 PostgreSQL 函數。若要這樣做，請選取「將程序轉換為函數」。
- 若要查看有關已發生之行動項目的其他資訊。若要這麼做，您可以針對具有下一個嚴重性層級的移轉問題選取 [例外狀況引發區塊]，將特定功能新增至擴充功能套件。然後選擇嚴重性層級，以引發使用者定義的
- 使用可能包含條件約束與自動產生名稱的來源 Oracle 資料庫。如果您的原始程式碼使用這些名稱，請務必選取「使用來源原始名稱轉換系統產生的限制名稱」。如果您的原始程式碼使用這些限制條件，但不使用其名稱，請清除此選項以提高轉換速度。
- 解決您的資料庫和應用程式是否在不同的時區中執行。默認情況下，AWS SCT 模擬轉換後的代碼中的時區。但是，當您的資料庫和應用程式使用相同的時區時，就不需要此模擬。在此情況下，請選取用戶端上的「時區」與伺服器上的時區相符。
- 解決您的來源和目標資料庫是否在不同的時區中執行。如果這樣做，模擬 SYSDATE 內置 Oracle 函數的函數返回與源函數相比不同的值。若要確保來源和目標函數傳回相同的值，請選擇「設定 SYSDATE 模擬的預設時區」。
- 在轉換後的代碼中使用 oraflce 擴展中的函數。若要這麼做，對於使用 oraflce 實作，請選取要使用的函數。如需有關 oraflce 的更多資訊，請參閱上的章節。GitHub

轉換甲骨文序列

AWS SCT 從甲骨文序列轉換為 PostgreSQL。如果您使用序列來維護完整性條件約束，請確定移轉序列的新值不會與現有值重疊。

使用來源資料庫中的最後一個值填入已轉換序列的步驟

1. 以 Oracle 作為來源開啟您的 AWS SCT 專案。

2. 選擇 [設定]，然後選擇 [轉換設定]。
3. 從上方的清單中選擇甲骨文，然後選擇甲骨文 — PostgreSQL。AWS SCT顯示甲骨文到 PostgreSQL 轉換的所有可用設定。
4. 選擇將在來源端產生的最後一個值植入已轉換序號。
5. 選擇「確定」以儲存設定並關閉「轉換設定」對話方塊。

轉換 Oracle ROWID

在 Oracle 資料庫中，ROWID 虛擬資料欄包含資料表列的地址。ROWID 偽列對甲骨文來說是唯一的，因此AWS SCT將 ROWID 偽列轉換為 PostgreSQL 上的數據列。通過使用此轉換，您可以保留 ROWID 信息。

轉換 ROWID 虛擬資料欄時，AWS SCT可以建立具有資料類型的資料欄。bigint如果沒有主鍵存在，則將 ROWID 列AWS SCT設置為主鍵。如果一個主鍵存在，AWS SCT設置一個唯一的約束 ROWID 列。

如果您的來源資料庫程式碼包含 ROWID 的作業，而您無法使用數值資料類型執行，則AWS SCT可以使用資料類型建立character varying資料行。

為專案的 Oracle ROWID 建立資料欄

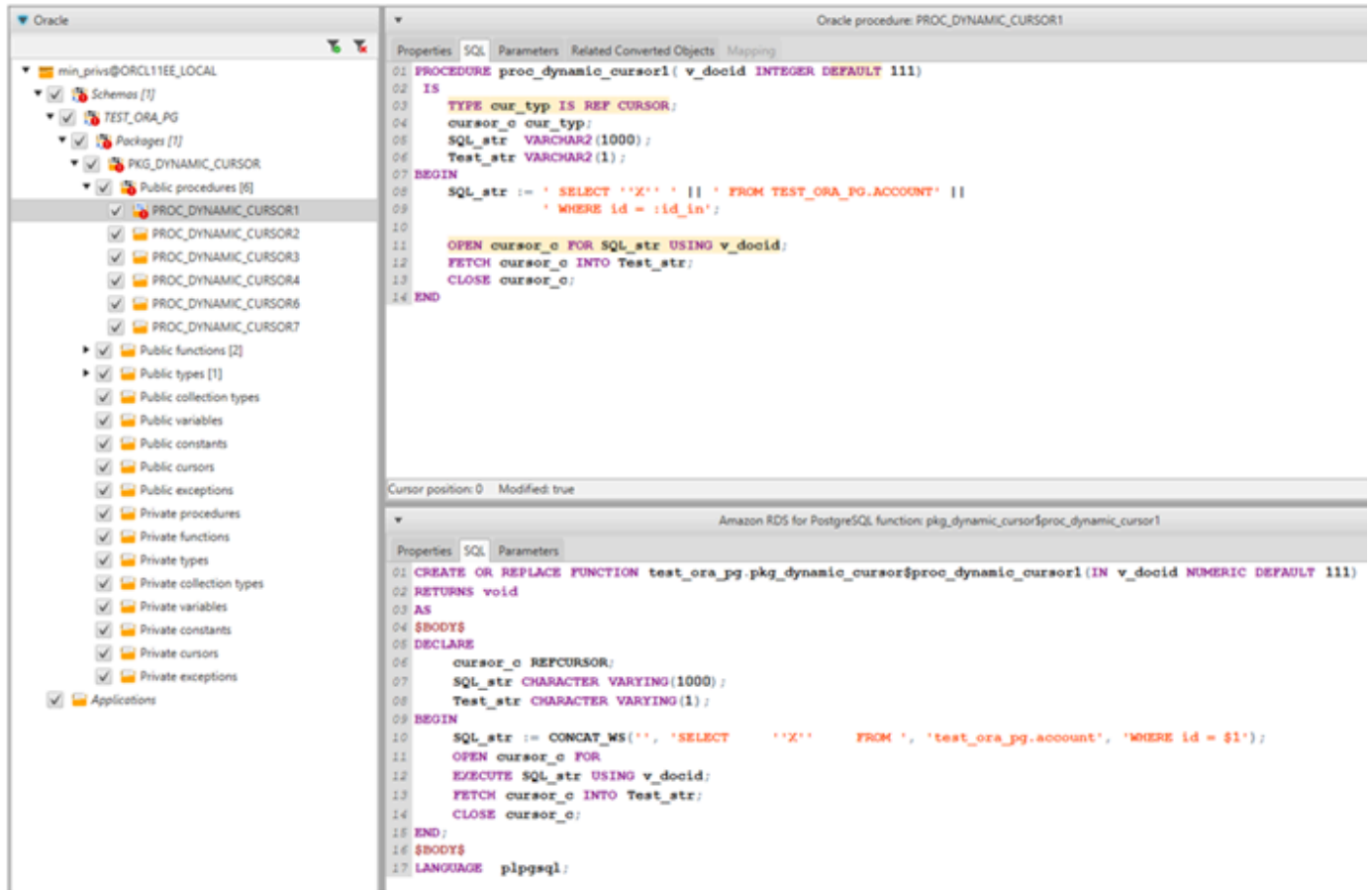
1. 以 Oracle 作為來源開啟您的AWS SCT專案。
2. 選擇 [設定]，然後選擇 [轉換設定]。
3. 從上方的清單中選擇甲骨文，然後選擇甲骨文 — PostgreSQL。AWS SCT顯示甲骨文到 PostgreSQL 轉換的所有可用設定。
4. 針對「產生列 ID」，執行下列其中一項作業：
 - 選擇產生為身分識別以建立數值資料欄。
 - 選擇產生為字元網域類型以建立字元資料欄。
5. 選擇「確定」以儲存設定並關閉「轉換設定」對話方塊。

轉換甲骨文動態 SQL

甲骨文提供了兩種方法來實現動態 SQL：使用立即執行語句或調用 DBMS_SQL 包中的過程。如果您的來源 Oracle 資料庫包含具有動態 SQL 的物件，請使用AWS SCT將 Oracle 動態 SQL 陳述式轉換為 PostgreSQL。

若要將甲骨文動態 SQL 轉換為 PostgreSQL

1. 以 Oracle 作為來源開啟您的 AWS SCT 專案。
2. 在 Oracle 來源樹狀結構檢視中，選擇使用動態 SQL 的資料庫物件。
3. 開啟物件的內容 (按一下滑鼠右鍵) 選單，選擇 [轉換結構描述]，並同意取代物件 (如果物件存在)。
下列螢幕擷取畫面顯示 Oracle 程序下方含有動態 SQL 的轉換程序。



轉換 Oracle 分割區

AWS SCT目前支援下列磁碟分割方法：

- 範圍
- 列出
- 多列範圍
- 雜湊
- 複合 (列表列表，範圍列表，列表範圍，列表散列，範圍散列，哈希散列)

將甲骨文轉換為亞馬遜 RDS 的 MySQL 或亞馬遜極光 MySQL

要在轉換後的 MySQL 代碼中模擬甲骨文數據庫函數，請使用甲骨文到 MySQL 擴展包。AWS SCT 如需擴充功能套件的詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

主題

- [將 MySQL 作為目標資料庫的權限](#)
- [甲骨文到 MySQL 的轉換設置](#)
- [移轉考量](#)
- [轉換與語句甲骨文到 RDS 為 MySQL 或亞馬遜極光 MySQL](#)

將 MySQL 作為目標資料庫的權限

MySQL 作為目標所需的權限如下：

- 建立於 * 。 *
- 更改 * 。 *
- 下降 * 。 *
- 索引 * 。 *
- 參考文獻 * 。 *
- SELECT ON *.*
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.*
- 觸發器於 * 。 *
- 在 * 上創建例程。 *
- 改變例程 * 。 *
- 執行於 * 。 *
- 在 * 上創建臨時表。 *
- AW_LAMBDA 訪問
- 插入，在 AWS_ORACLE_EXT 上進行更新。 *
- 插入、更新、刪除 *

如果您使用 MySQL 資料庫版本 5.7 或更低版本做為目標，請授與叫用 LAMBDA *.* 權限，而非 AWS_LAMBDA 存取權限。對於 MySQL 資料庫 8.0 版及更高版本，請授予 AW_LAMBDA 存取權限。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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';
```

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，用安全的#####。

如果您使用 MySQL 資料庫版本 5.7 或更低版本作為目標，請使用 GRANT INVOKE LAMBDA ON *.* TO 'user_name' 代替 GRANT AWS_LAMBDA_ACCESS TO 'user_name'。

若要使用適用於 MySQL 的亞馬遜 RDS 或極光 MySQL 作為目標，請將 `lower_case_table_names` 參數設定為 1。此值意味著 MySQL 服務器處理不區分大小寫的表，索引，觸發器和數據庫等對象名稱的標識符。如果您已在目標執行個體中開啟二進位記錄功能，請將 `log_bin_trust_function_creators` 參數設定為 1。在這種情況下，您不需要使用 DETERMINISTIC、READS SQL DATA 或 NO SQL 特性來建立已儲存的函數。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

甲骨文到 MySQL 的轉換設置

若要編輯 Oracle 到 MySQL 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇甲骨文，然後選擇甲骨文-MySQL。AWS SCT 顯示甲骨文到 MySQL 的轉換的所有可用設置。

甲骨文到 MySQL 的轉換設置 AWS SCT 包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 為了解決您的源 Oracle 數據庫可以使用ROWID偽列，但 MySQL 不支持類似的功能。AWS SCT可以模擬轉換後的ROWID代碼中的偽列。要這樣做，請選擇生成為生成行 ID 的標識？。

如果您的源 Oracle 代碼不使用ROWID偽列，請選擇不生成為生成行 ID？在這種情況下，轉換後的代碼的工作速度更快。

- 在原始程式碼 Oracle 程式碼包含TO_CHAR、TO_DATE、和TO_NUMBER函數時使用 MySQL 不支援的參數。默認情況下，AWS SCT模擬這些參數在轉換後的代碼的使用。

當您的原始程式碼只包含 PostgreSQL 支援的參數時，您可以使用原生 MySQL TO_CHAR和TO_DATE函數。TO_DATE在這種情況下，轉換後的代碼的工作速度更快。若只要包括這些參數，請選取下列值：

- 函數 TO_CHAR () 不使用甲骨文特定的格式化字符串
- 函數 TO_DATE () 不使用甲骨文特定的格式化字符串
- 函數 TO_NUMBER () 不使用甲骨文特定的格式化字符串
- 為了補充您的數據庫和應用程序是否在不同的時區中運行。默認情況下，AWS SCT模擬轉換後的代碼中的時區。但是，當您的資料庫和應用程式使用相同的時區時，就不需要此模擬。在此情況下，請選取用戶端上的「時區」與伺服器上的時區相符。

移轉考量

當您將 Oracle 轉換為適用於 MySQL 的 RDS 或極光 MySQL 時，若要變更陳述式執行的順序，您可以使用GOTO陳述式和標籤。系統會略過敘述句後面的任何 PL/SQL 敘述句，並在標籤處繼續處理。GOTO您可以在程序、批次或GOTO陳述式區塊中的任何位置使用陳述式和標籤。您也可以接下來GOTO 語句。

MySQL 不使用GOTO語句。當AWS SCT轉換包含GOTO陳述式的程式碼時，會將陳述式轉換為使用BEGIN...END or LOOP...END LOOP 陳述式。

您可以在下表中找到如何AWS SCT轉換GOTO陳述式的範例。

Oracle 陳述式

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

MySQL 陳述式

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

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

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

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

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


Oracle 陳述式	MySQL 陳述式
<pre> statement3; statement4; END </pre>	<pre> statement2; statement3; statement4; END; END </pre>

轉換與語句甲骨文到 RDS 為 MySQL 或亞馬遜極光 MySQL

您可以使用 Oracle 的 WITH 子句 (subquery_factoring) 對子查詢區塊指派名稱 (query_name)。接著，您即可透過指定 query_name 在查詢中的多處位置參考該子查詢區塊。如果子查詢塊不包含鏈接或參數（本地，過程，函數，包），則 AWS SCT 將子句轉換為視圖或臨時表。

將子句轉換為暫時資料表的優點是，對子查詢的重複引用可能會更有效率。效率更高的是因為資料很容易從暫時資料表中檢索，而不是每個引用都需要。您可以通過使用其他視圖或臨時表來模擬這一點。視圖名稱會使用格式 <procedure_name>\${<subselect_alias>。

您可以在下表中找到範例。

Oracle 陳述式	MySQL 陳述式
<pre> CREATE PROCEDURE TEST_ORA_PG.P_WITH_SELECT_V ARIABLE_01 (p_state IN NUMBER) AS l_dept_id NUMBER := 1; BEGIN FOR cur IN (WITH dept_empl(id, name, surname, lastname, state, dept_id) AS (SELECT id, name, surname, </pre>	<pre> CREATE PROCEDURE test_ora_pg.P_WITH _SELECT_VARIABLE_01(IN par_P_STATE DOUBLE) BEGIN DECLARE var_l_dept_id DOUBLE DEFAULT 1; DECLARE var\$id VARCHAR (8000); DECLARE var\$state VARCHAR (8000); DECLARE done INT DEFAULT FALSE; DECLARE cur CURSOR FOR SELECT ID, STATE FROM (SELECT ID, NAME, SURNAME, LASTNAME, STATE, DEPT_ID FROM TEST_ORA_PG.DEPT_E MPLOYEES </pre>

Oracle 陳述式

```
                lastname, state,
dept_id
                FROM test_ora_
pg.dept_employees
                WHERE state =
p_state AND
                dept_id =
l_dept_id)
        SELECT id,state
        FROM dept_empl
        ORDER BY id) LOOP
        NULL;
END LOOP;
```

MySQL 陳述式

```
                WHERE STATE = par_p_sta
te AND DEPT_ID = var_l_dept_id) AS
dept_empl
        ORDER BY ID;
        DECLARE CONTINUE HANDLER FOR NOT
FOUND
        SET done := TRUE;
        OPEN cur;

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

                IF done THEN
                        LEAVE read_label;
                END IF;

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

Oracle 陳述式

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_R
  EGULAR_MULT_01
AS
BEGIN

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

```

MySQL 陳述式

```

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

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

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

```

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

Oracle 陳述式

```

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

```

MySQL 陳述式

```

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$ID DOUBLE;
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT *
    FROM
      (SELECT
        ID
      FROM
        TEST_ORA_
PG.
        `P_WIT_H_S
ELECT_VAR_CROSS_01$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:

```

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

將 Oracle 轉換到 Amazon RDS for Oracle

將 Oracle 結構描述和程式碼遷移到 Amazon RDS for Oracle 的一些考量事項：

- AWS SCT 可以將目錄物件新增至物件樹狀結構。目錄物件是邏輯結構，每個物件都代表伺服器檔案系統上的實體目錄。您可以使用目錄物件搭配套件，例如 DBMS_LOB、UTL_FILE、DBMS_FILE_TRANSFER、DATAPUMP 公用程式等。
- AWS SCT 支援將 Oracle 表空間轉換為適用於 Oracle 資料庫執行個體的亞馬遜 RDS。Oracle 將資料以邏輯方式存放在資料表空間中，並以實體方式存放在與相應資料表空間關聯的資料檔案中。在 Oracle 中，您可以使用資料檔案名稱來建立資料表空間。Amazon RDS 僅針對資料檔案、日誌檔和控制檔案支援 Oracle 受管檔案 (OMF)。AWS SCT 在轉換期間建立所需的資料檔案。
- AWS SCT 可以轉換伺服器層級的角色和權限。Oracle 資料庫引擎是使用以角色為基礎的安全性。角色是一組權限，您可以用來授與給使用者或撤銷。Amazon RDS 的預先定義的角色 (稱為 DBA) 通常允許 Oracle 資料庫引擎的所有管理權限。以下權限不適用於使用 Oracle 引擎之 Amazon RDS 資料庫執行個體的 DBA 角色：
 - 更改資料庫
 - 更改系統
 - 建立任何目錄
 - 授予任何權限
 - 授予任何角色
 - 建立外部工作

您可以對 Amazon RDS for Oracle 使用者角色授與所有其他權限，包括進階篩選及資料欄權限。

- AWS SCT 支援將 Oracle 任務轉換為可在亞馬遜 RDS 上執行的任務。轉換有幾個限制，包括下列項目：
 - 不支援可執行的任務。
 - 不支援使用 ANYDATA 資料類型做為引數的排程任務。
- Oracle Real Application Clusters (RAC) One Node 是 Oracle Database Enterprise Edition 中的一個選項，與 Oracle Database 11g Release 2 一同推出。Amazon RDS for Oracle 不支援 RAC 功能。使用 Amazon RDS 異地同步備份以提供高可用性。

在異地同步備份部署中，Amazon RDS 會自動佈建，並在不同的可用區域中維持同步待命複本。主資料庫執行個體會在待命複本的可用區域間進行同步複製。此功能可提供資料備援，排除 I/O 凍結，並降低系統備份時的延遲遽增。

- Oracle Spatial 提供的 SQL 結構描述和函數，有助於 Oracle 資料庫中空間資料集合的儲存、擷取、更新和查詢。Oracle Locator 提供通常支援下列項目所需的功能：網際網路和無線服務型應用程式和合作夥伴型 GIS 解決方案。Oracle Locator 是功能有限的 Oracle Spatial 子集。

若要使用 Oracle Spatial 和 Oracle Locator 功能，請將 SPATIAL 選項或 LOCATOR 選項 (兩者互斥) 新增至資料庫執行個體的選項群組。

在 Amazon RDS for Oracle 資料庫執行個體上使用 Oracle Spatial 和 Oracle Locator 有一些先決條件：

- 執行個體應該使用甲骨文企業版 12.1.0.2.v6 或更高版本，或者是 11.2.0.4.v10 或更高版本。
- 執行個體應位於 Virtual Private Cloud (VPC) 內。
- 執行個體應為可支援 Oracle 功能的資料庫執行個體類別。例如，db.m1.small、db.t1.micro、db.t2.micro 或 db.t2.small 資料庫執行個體類別不支援 Oracle Spatial。如需詳細資訊，請參閱 [Oracle 的資料庫執行個體類別支援](#)。
- 執行個體必須啟用 Auto Minor Version Upgrade (自動次要版本升級) 選項。如果存在 CVSS 分數 9 分以上的安全漏洞或其他公布的安全漏洞，Amazon RDS 會將您的資料庫執行個體更新為最新的 Oracle PSU。如需詳細資訊，請參閱

[Oracle 資料庫執行個體的設定值。](#)

- 如果您的資料庫執行個體是 11.2.0.4.v10 或更新版本，則必須安裝 XMLDB 選項。如需詳細資訊，請參閱

[甲骨文 XML 數據庫。](#)

- 您應具有 Oracle 提供的 Oracle Spatial 授權。如需詳細資訊，請參閱 Oracle 文件中的 [Oracle Spatial 和 Graph](#)。
- Data Guard 包含在 Oracle Database Enterprise Edition 中。使用 Amazon RDS 異地同步備份功能以提供高可用性。

在異地同步備份部署中，Amazon RDS 會自動佈建，並在不同的可用區域中維持同步待命複本。主資料庫執行個體會在待命複本的可用區域間進行同步複製。此功能可提供資料備援，排除 I/O 凍結，並降低系統備份時的延遲遽增。

- AWS SCT 在遷移到亞馬遜 RDS 適用於甲骨文時，支持轉換甲骨文 DBMS_Scheduler 對象。AWS SCT 評估報告會指出排程物件是否可以轉換。如需有關搭配 Amazon RDS 使用排程物件的詳細資訊，請參閱 [Amazon RDS 文件](#)。
- 對於從 Oracle 到 Amazon RDS for Oracle 的轉換，支援資料庫連結。資料庫連結為某一資料庫內的結構描述物件，讓您能夠存取另一個資料庫上的物件。另一個資料庫不需要是 Oracle 資料庫。不過，若要存取非 Oracle 資料庫，您必須使用 Oracle 異質服務。

一旦建立資料庫連結，您可用 SQL 陳述式中的連結參閱另一資料庫內的表格、檢視和 PL/SQL 物件。若要使用資料庫連結，請附加 @dblink 至表格、檢視或 PL/SQL 物件名稱。您可用 SELECT 陳述式查詢另一資料庫內的表格或檢視。如需有關使用 Oracle 資料庫連結的詳細資訊，請參閱 [Oracle 文件](#)。

如需有關使用資料庫連結與 Amazon RDS 的詳細資訊，請參閱 [Amazon RDS 文件](#)。

- AWS SCT 評估報告會提供轉換的伺服器量度。這類有關 Oracle 執行個體的各项指標包括：
 - 目標資料庫執行個體的運算和記憶體容量。
 - 不支援的 Oracle 功能，例如 Amazon RDS 不支援的實際應用程式叢集。
 - 磁碟讀寫負載。
 - 磁碟輸送量總平均。
 - 伺服器資訊，如：伺服器名稱、作業系統、主機名稱和字元集。

將 RDS 作為甲骨文作為目標的權限

若要遷移到亞馬遜 RDS 版甲骨文，請建立具有特權的資料庫使用者。您可以使用下面的代碼示例。

```
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;
```

在上述範例中，將 *user_name* 取代為您的使用者名稱。然後，用安全的#####。

將甲骨文轉換為亞馬遜 RDS

將 Oracle 結構描述和程式碼遷移到 Amazon RDS for Oracle 時應考量的一些限制：

- Amazon RDS 的預先定義的角色 (稱為 DBA) 通常允許 Oracle 資料庫引擎的所有管理權限。以下權限不適用於使用 Oracle 引擎之 Amazon RDS 資料庫執行個體的 DBA 角色：
 - 更改資料庫
 - 更改系統
 - 建立任何目錄
 - 授予任何權限
 - 授予任何角色
 - 建立外部工作

您可以授與所有其他權限給 Oracle RDS 使用者角色。

- 適用於甲骨文的 Amazon RDS 支援傳統稽核、使用 DBMS_FGA 套件進行精細的稽核，以及甲骨文統一稽核。
- Amazon RDS for Oracle 不支援變更資料擷取 (CDC)。若要在資料庫移轉期間和之後執行 CDC，請使用 AWS Database Migration Service。

使用 PostgreSQL 作為下列項目的來源 AWS SCT

您可以使用 AWS SCT 將結構描述、資料庫程式碼物件和應用程式程式碼從 PostgreSQL 轉換為下列目標：

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

如需詳細資訊，請參閱下列章節：

主題

- [將 PostgreSQL 作為來源資料庫的權限](#)
- [連線至 PostgreSQL 作為來源](#)
- [將 MySQL 作為目標資料庫的權限](#)

將 PostgreSQL 作為來源資料庫的權限

PostgreSQL 作為來源所需的權限如下：

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

連線至 PostgreSQL 作為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 PostgreSQL 來源資料庫。

若要連接到 PostgreSQL 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 PostgreSQL，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 為使用連線，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自 Secrets Manager 的資料庫認證，請遵循下列指示：
 1. 對於 AWS 機密，請選擇密碼的名稱。
 2. 選擇填入來自 Secrets Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自 Secrets Manager 之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 PostgreSQL 來源資料庫連線資訊，請使用下列指示：

參數	動作
伺服器名稱	<p>輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 地址。</p> <p>您可以使用 IPv6 地址協議來連線至來源 PostgreSQL 資料庫。若要這麼做，請務必使用方括號來輸入 IP 地址，如下列範例所示。</p>

參數	動作
	[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 PostgreSQL 資料庫的名稱。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用密碼來連線至來源資料庫。為了防止公開來源資料庫的密碼，預設 AWS SCT 不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項，使用 Secure Sockets Layer (SSL) 來連線至您的資料庫。在 SSL 索引標籤上提供下列其他資訊 (如適用)：</p> <ul style="list-style-type: none"> 驗證伺服器憑證：選取此選項可使用信任存放區驗證伺服器憑證。 信任存放區：包含憑證的信任存放區的位置。若要讓此位置顯示在「全域設定」區段中，請務必新增該位置。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。啟用此選項可讓您存放資料庫密碼，並快速連接到資料庫而無需輸入密碼。
驅動程 PostgreSQL 路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

- 選擇「測試連線」以確認是否 AWS SCT 可以連線到來源資料庫。
- 選擇「Connect」以連線到來源資料庫。

將 MySQL 作為目標資料庫的權限

當您從 PostgreSQL 遷移時，MySQL 作為目標所需的權限如下：

- 建立於 * 。 *
- 變更於 * 。 *
- 下降 * 。 *
- 索引 * 。 *
- 參考 * 。 *
- SELECT ON *.*
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.*
- 觸發 * 。 *
- 在 * 上創建例程。 *
- 改變例程 * 。 *
- 執行日期 * 。 *
- 插入，在下一個更新。 *
- 插入、更新、刪除 *
- 在檔案資料上建立暫存資料表。 *

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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';
```

在上述範例中，使用 `user_name ### user_name`，使用您的使用者名稱。然後，用安全 `#####` 的密碼。

若要使用適用於 MySQL 的 Amazon RDS for MySQL 或 Aurora MySQL 作為目標，請將 `lower_case_table_names` 參數設定為 1。此值意味著 MySQL 服務器處理不區分大小寫的表，索引，觸發器和數據庫等對象名稱的標識符。如果您已在目標執行個體中開啟二進位記錄功能，請將 `log_bin_trust_function_creators` 參數設定為 1。在這種情況下，您不需要使用 DETERMINISTIC、READS SQL DATA 或 NO SQL 特性來建立已儲存的函數。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

使用 SAP 日月光 (系統日月光) 作為下列項目的來源 AWS SCT

您可以使用 AWS SCT 將結構描述、資料庫程式碼物件和應用程式程式碼從 SAP (Sybase) 調適性伺服器企業 (ASE) 轉換為下列目標：

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

如需詳細資訊，請參閱下列章節：

主題

- [SAP ASE 作為來源資料庫的權限](#)
- [連線至 SAP ASE \(系統庫\) 做為來源](#)
- [將 MySQL 作為目標資料庫的權限](#)
- [轉換 MySQL 定](#)
- [將 PostgreSQL 作為目標資料庫的權限](#)
- [SAP 日月 PostgreSQL 設定](#)

SAP ASE 作為來源資料庫的權限

若要使用 SAP ASE 資料庫作為來源，請建立資料庫使用者並授與權限。若要執行此動作，請執行下列步驟。

建立和設定資料庫使用者

1. 連接到來源資料庫。
2. 使用下列命令建立資料庫使用者。為新使用者提供密碼。

```
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. 對於要移轉的每個資料庫，授與下列權限。

```
USE <database_name>
sp_adduser min_privs
grant select on dbo.sysusers to min_privs
grant select on dbo.sysobjects to min_privs
grant select on dbo.sysindexes to min_privs
grant select on dbo.syscolumns to min_privs
grant select on dbo.sysreferences to min_privs
grant select on dbo.syscomments to min_privs
grant select on dbo.syspartitions to min_privs
grant select on dbo.syspartitionkeys to min_privs
grant select on dbo.sysconstraints to min_privs
grant select on dbo.systypes to min_privs
grant select on dbo.sysqueryplans to min_privs
```

連線至 SAP ASE (系統庫) 做為來源

請使用下列程序使用連線至您的 SAP ASE 來源資料庫AWS Schema Conversion Tool。

若要連線至 SAP ASE 來源資料庫


1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 SAP ASE，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 針對 Connection (連線名稱)，輸入您的資料庫名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：
 - 若要使用來自 Secrets Manager 的資料庫認證，請遵循下列指示：
 1. 對於 AWS 機密，請選擇密碼的名稱。
 2. 選擇填入來自 Secrets Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自 Secrets Manager 之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 SAP ASE 來源資料庫連線資訊，請遵循下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 地址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 SAP ASE 資料庫的名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用密碼來連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p> </div>
Use SSL (使用 SSL)	選擇此選項可使用 Secure Sockets Layer (SSL) 來連線至您的資料庫。在 SSL 索引標籤上提供下列其他資訊 (如適用)： <ul style="list-style-type: none"> • 驗證伺服器憑證：選取此選項可使用信任存放區驗證伺服器憑證。 • 信任存放區：包含憑證的信任存放區的位置。

參數	動作
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。啟用此選項可讓您存放資料庫密碼，並快速連接到資料庫而無需輸入密碼。
SAP 日光驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「Connect」以連線到來源資料庫。

將 MySQL 作為目標資料庫的權限

MySQL 作為目標所需的權限如下：

- 建立於 * 。 *
- 變更於 * 。 *
- 下降 * 。 *
- 索引 * 。 *
- 參考資料 * 。 *
- SELECT ON *.*
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.*
- 觸發程序於 * 。 *
- 在 * 上創建例程。 *
- 改變例程 * 。 *
- 執行於 * 。 *
- 插入, 更新. *
- 建立暫存資料表。 *

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

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

在上述範例中，使用您使用者的##### *user_name*。然後，用安全#####的密碼。

若要使用適用於 MySQL 的 Amazon RDS for MySQL 或 Aurora MySQL 作為目標，請將 `lower_case_table_names` 參數設定為 1。此值意味著 MySQL 服務器處理不區分大小寫的表，索引，觸發器和數據庫等對象名稱的標識符。如果您已在目標執行個體中開啟二進位記錄功能，請將 `log_bin_trust_function_creators` 參數設定為 1。在這種情況下，您不需要使用 DETERMINISTIC、READS SQL DATA 或 NO SQL 特性來建立已儲存的函數。若要設定這些參數，請建立新的資料庫參數群組或修改現有的資料庫參數群組。

轉換 MySQL 定

若要編輯 SAP ASE 至 MySQL 的轉換設定，請選擇 [設定]，然後選擇 [轉換設定]。從上方的清單中選擇 SAP 日月光，然後選擇 SAP 日月光 — MySQL 或 SAP ASE — Amazon Aurora (與 MySQL 相容)。AWS SCT 顯示轉換至 PostgreSQL 的所有可用設定。

SAP ASE 到 MySQL 的轉換設定 AWS SCT 包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT 在所選嚴重性及更高之行動項目的轉換程式碼中加入註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 在轉換後的代碼中使用源數據庫對象的確切名稱。

根據預設，會AWS SCT將資料庫物件、變數和參數的名稱轉換為小寫。若要保留這些名稱的原始大小寫，請選取「將來源資料庫物件名稱視為區分大小寫」。如果您在來源 SAP ASE 資料庫伺服器中使用區分大小寫的物件名稱，請選擇此選項。

將 PostgreSQL 作為目標資料庫的權限

若要使用 PostgreSQL 做為目標，AWS SCT需要CREATE ON DATABASE權限。請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為"\$user", public_synonyms, public。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在上述範例中，使用您使用者的##### *user_name*。然後，將 *db_name* 替換為目標數據庫的名稱。最後，用安全#####的密碼。

在 PostgreSQL 中，只有結構描述擁有者或一個superuser可以刪除結構描述。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的使用者將不同的結構描述轉換並套用至目標資料庫時，您可能會在無法卸除結構描述時AWS SCT收到錯誤訊息。若要避免此錯誤訊息，請使用superuser角色。

SAP 日月 PostgreSQL 設定

若要編輯 SAP ASE 至 PostgreSQL 的轉換設定，請選擇 [設定]，然後選擇 [轉換設定]。從上方的清單中選擇 SAP 日月光，然後選擇日月光 — PostgreSQL 或 SAP 日月光 — Amazon Aurora (與 PostgreSQL 相容)。AWS SCT顯示轉換至 PostgreSQL 的所有可用設定。

SAP ASE 轉換至 PostgreSQL 中的轉換設定AWS SCT包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在所選嚴重性及更高之行動項目的轉換程式碼中加入註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 定義用於轉換後的代碼模式名稱的模板。針對 Schema 名稱產生範本，請選擇下列其中一項：
 - <source_db>— 在 PostgreSQL 中使用 SAP ASE 資料庫名稱做為結構描述名稱。
 - <source_schema>— 在 PostgreSQL 中使用 SAP ASE 結構描述名稱作為結構描述名稱。
 - _<source_db><schema>— 使用 SAP ASE 資料庫和結構描述名稱的組合作為 PostgreSQL 中的結構描述名稱。
- 在轉換後的代碼中使用源數據庫對象的確切名稱。

根據預設，會 AWS SCT 將資料庫物件、變數和參數的名稱轉換為小寫。若要保留這些名稱的原始大小寫，請選取「將來源資料庫物件名稱視為區分大小寫」。如果您在來源 SAP ASE 資料庫伺服器中使用區分大小寫的物件名稱，請選擇此選項。

對於區分大小寫的操作，AWS SCT 可以避免將數據庫對象名稱轉換為小寫。若要這樣做，請針對區分大小寫的作業選取「避免轉換為小寫」。

- 為了允許在 SAP ASE 中的不同表中使用具有相同名稱的索引。

在 PostgreSQL 中，您在結構描述中使用的所有索引名稱都必須是唯一的。若要確定為所有索引產生 AWS SCT 唯一的名稱，請選取「產生索引的唯一名稱」。

使用 Microsoft SQL 服務器作為源 AWS SCT

您可以使用 AWS SCT 將結構描述、資料庫程式碼物件和應用程式程式碼從 SQL Server 轉換為下列目標：

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

Note

AWS SCT不支援使用亞馬遜 RDS 版 SQL 伺服器做為來源。

您可以使用AWS SCT以建立將結構描述、資料庫程式碼物件和應用程式程式碼從 SQL Server 移轉至 Aurora PostgreSQL 的 BabelFish 的評估報告，如下所述。

主題

- [權限作為來源的 Microsoft SQL 伺服器](#)
- [使用 Microsoft SQL 服務器作為源時使用視窗身份驗證](#)
- [連線至 SQL 伺服器做為來源](#)
- [將 SQL Server 轉換到 MySQL](#)
- [將一個 SQL Server 轉換到 PostgreSQL](#)
- [將 SQL Server 轉換到 Amazon RDS for SQL Server](#)

權限作為來源的 Microsoft SQL 伺服器

Microsoft SQL 伺服器作為來源所需的權限如下：

- VIEW DEFINITION
- VIEW DATABASE STATE

該VIEW DEFINITION權限可讓具有公開存取權的使用者查看物件定義。AWS SCT使用VIEW DATABASE STATE檢查 SQL 伺服器企業版功能的權限。

對您要轉換其結構描述的每個資料庫重複授予。

此外，授與下列權限master資料庫：

- VIEW SERVER STATE
- 檢視任何定義

AWS SCT使用VIEW SERVER STATE收集伺服器設定和組態的權限。請確定您授與VIEW ANY DEFINITION檢視端點的權限。

若要閱讀 Microsoft 分析服務的相關資訊，請在master數據庫。

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

在前面的範例中，取代<user_name>預留位置，其中包含您之前授與權限的使用者名稱。

若要閱讀 SQL Server 代理程式的相關資訊，請將您的使用者新增至SQLAgentUser角色。執行下列命令msdb數據庫。

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

在前面的範例中，取代<SQLAgentRole>含有 SQL Server 代理程式角色名稱的預留位置。然後更換<user_name>預留位置，其中包含您之前授與權限的使用者名稱。如需詳細資訊，請參閱[將使用者新增至 SQLAgentUser 角色](#)在亞馬遜 RDS 用戶指南。

若要偵測記錄傳送，請授與SELECT on dbo.log_shipping_primary_databases權限msdb數據庫。

若要使用 DDL 複寫的通知方法，請授與RECEIVE ON <schema_name>.<queue_name>源數據庫的權限。在此範例中，取代<schema_name>佔位符與數據庫的模式名稱。然後，更換<queue_name>含有佇列表格名稱的預留位置。

使用 Microsoft SQL 服務器作為源時使用視窗身份驗證

如果您的應用程式執行於以 Windows 為基礎的內部網路，您可以在存取資料庫時使用 Windows 驗證。Windows 驗證使用目前在作業系統執行緒上建立的 Windows 身分存取 SQL Server 資料庫。接著您可將 Windows 身分對應到 SQL Server 資料庫和許可。若要使用 Windows 驗證連接到 SQL Server，您必須指定應用程式使用正在使用的 Windows 身分。您也必須授予 Windows 身分對 SQL Server 資料庫的存取。

SQL Server 有兩種存取模式：Windows 驗證模式和混合模式。Windows 驗證模式可啟用 Windows 驗證並停用 SQL Server 驗證。混合模式可同時啟用 Windows 驗證和 SQL Server 驗證。Windows 驗證一律可用，且無法停用。如需有關 Windows 驗證的詳細資訊，請參閱 Microsoft Windows 文件。

在 TEST_DB 中建立使用者的可能範例如下

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


使用視窗驗證及 JDBC 連線

JDBC 驅動程式用於非 Windows 作業系統時，JDBC 驅動程式不支援 Windows 驗證。當從非 Windows 作業系統連線到 SQL Server 時，系統不會自動指定 Windows 驗證身份證明，例如使用者名稱和密碼。在這種情況下，應用程式必須改用 SQL Server 驗證。

在 JDBC 連線字串中，`integratedSecurity` 參數必須指定使用 Windows 驗證連接。JDBC 驅動程式透過 `integratedSecurity` 連線字串參數支援 Windows 作業系統上的整合式 Windows 驗證。

使用整合式驗證

1. 安裝 JDBC 驅動程式。
2. 將 `sqljdbc_auth.dll` 檔案複製到安裝 JDBC 驅動程式之電腦的 Windows 系統路徑上的目錄。

該 `sqljdbc_auth.dll` 檔案安裝在下列位置：

`<安裝目錄>\sqljdbc_<version>\<language>\auth\`

嘗試使用 Windows 驗證建立到 SQL Server 資料庫的連線時，您可能會收到錯誤：此驅動程式未設定使用整合式驗證。您可透過執行以下動作解決這個問題：

- 宣告兩個變數，指向 JDBC 的安裝路徑：

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu ( 如果你的  
sqljdbc4.jar 存在 );
```

```
variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu\auth  
\x86 ( 如果您正在運行 32 位操作系統 ) 或 D\lib\JDBC4.1\enu\auth\x64 ( 如果您正在運行  
64 位操作系統 )。這是你的 sqljdbc_auth.dll 所在的位置。
```

- 複製 `sqljdbc_auth.dll` 到您的 JDK/JRE 正在運行的文件夾。您可以複製到 `lib` 資料夾、`bin` 資料夾等。例如，您可能會複製到下列資料夾。

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

- 請確定在您的 JDBC 程式庫資料夾中，您只有 `SQLJDBC4.jar` 檔案。從該文件夾中刪除任何其他 `sqljdbc*.jar` 文件（或將它們複製到另一個文件夾）。如果您要將驅動程式新增為程式的一部分，請確定您只新增 `SQLJDBC4.jar` 作為要使用的驅動程式。

- 複製應用程式資料夾內的 `sqljdbc_auth.dll` 檔案。

Note

如果您執行 32 位元 Java 虛擬機器 (JVM)，請使用 x86 資料夾中的 `sqljdbc_auth.dll` 檔案，即使作業系統為 x64 版本。如果您在 x64 處理器上執行 64 位元 JVM，請使用 x64 資料夾中的 `sqljdbc_auth.dll` 檔案。

連接到 SQL Server 資料庫時，您可以選擇 Windows 驗證或 SQL Server 驗證做為驗證選項。

連線至 SQL 伺服器做為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Microsoft SQL Server 來源資料庫。

若要連接到 Microsoft SQL Server 來源資料庫

- 在 AWS Schema Conversion Tool，選擇新增來源。
- 選擇 Microsoft SQL Server，然後選擇下一步。

該新增來源對話方塊。

- 對於「連線名稱」下方，輸入資料庫名稱。AWS SCT 在左側面板的樹狀結構中顯示此名稱。
- 使用資料庫認證 AWS Secrets Manager 或手動輸入：

- 若要使用來自 Secrets Manager 的資料庫認證，請遵循下列指示：
 - 對於「AWS 任務」下方，選擇密碼名稱。
 - 選擇填充從 Secrets Manager 自動填寫數據庫連接對話框中的所有值。

如需在 Secrets Manager 上使用 Database Mail 的相關資訊，請參閱 [使用 AWS Secrets Manager](#)。

- 若要手動輸入 Microsoft SQL Server 來源資料庫連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入您來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。

參數	動作
	<p>您可以在 IPv6 位址通訊協定上使用 IPv6 傳送給您的傳送 SQL Server 資料庫。若要這樣做，請確定您在下列範例中使用方括號輸入 IP 位址。</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</p> </div>
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
執行個體名稱	輸入 SQL Server 資料庫的執行個體名稱。若要尋找執行個體名稱，請在 SQL Server 資料庫上執行查詢 <code>SELECT @@servername; </code>
身分驗證	選擇驗證類型視窗驗證和 SQL 伺服器驗證。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用密碼來連線至來源資料庫。為了防止暴露來源資料庫的密碼，AWS SCT 默認情況下不存儲密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項，您可以在資料庫上使用 Database Sail 層 (SSL)。提供下列額外資訊 (如適用)SSL 標</p> <ul style="list-style-type: none"> • 信任伺服器：選取此選項可信任伺服器憑證。 • 信任商店：包含憑證的信任存放區位置。對於這個位置出現在全域設定部分中，請確保添加它。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。啟用此選項可讓您存放資料庫密碼，並快速連接到資料庫而無需輸入密碼。

參數	動作
Sql Server 驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>
視窗驗證程式庫	<p>輸入路徑sqljdbc_auth.dll 文件。依預設，此檔案會安裝在下列位置：</p> <p><i><installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\</i></p>

5. 選擇測試連接以驗證AWS SCT可以連接到您的源數據庫。
6. 選擇Connect連線至資料庫。

將 SQL Server 轉換到 MySQL

要在轉換後的 MySQL 代碼中模擬 Microsoft SQL 服務器數據庫功能，請使用 SQL 服務器到 MySQL 擴展包AWS SCT。如需延伸功能表的詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

主題

- [將 MySQL 作為目標資料庫的權限](#)
- [SQL 伺服器到 MySQL 的轉換設定](#)
- [移轉考量](#)

將 MySQL 作為目標資料庫的權限

MySQL 作為目標所需的權限如下：

- 建立於 * 。 *
- 改變 * 。 *
- 下降 * 。 *
- 開始新增複本表 。 *

- 參考資料。 *
- SELECT ON *.* *
- 在 * 上建立檢視。 *
- SHOW VIEW ON *.* *
- 開始新任務。 *
- 在 * 上創建例程。 *
- 改變例程 *。 *
- 執行日期 *。 *
- 插入，在 AWS_SQL 伺服器上進行更新。 *
- 插入、更新、刪除。 *
- 在 AWS_SQL 伺服器上建立暫存資料表。 *

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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';
```

在前面的範例中，取代####您的使用者名稱。然後，更換####使用安全密碼。

如果您使用 MySQL 資料庫版本 5.7 或更低版本做為目標，請執行下列命令。對於 MySQL 版本 8.0 及更高版本的資料庫，此命令已被棄用。

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

要使用 Amazon RDS for MySQL 為 MySQL 或 Aurora MySQL 作為目標，設置 `lower_case_table_names` 參數到 1。此值意味著 MySQL 服務器處理不區分大小寫的表，索引，觸發器和數據庫等對象名稱的標識符。如果您已在目標執行個體中開啟二進位記錄功能，請設定 `log_bin_trust_function_creators` 參數到 1。在此情況下，您不需要使用 DETERMINISTIC, READS SQL DATA 或者 NO SQL 特性來創建存儲的函數。若要設定這些參數，請在此建立新的 Database 參數群組，或者修改現有 Database Parameter Mail，

SQL 伺服器到 MySQL 的轉換設定

若要編輯 SQL 伺服器到 MySQL 的轉換設定，請在 AWS SCT 選擇設定，然後選擇轉換設定。從上方的清單中選擇 SQL 伺服器，然後選擇 SQL 伺服器。AWS SCT 顯示 SQL 伺服器到 MySQL 轉換的所有可用設定。

SQL 服務器到 MySQL 的轉換設置 AWS SCT 包括下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

對於在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解」下，選擇行動項目的嚴重性。AWS SCT 在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數量降至最低，請選擇僅限錯誤。若要在轉換的程式碼中包含所有行動項目的註解，請選擇所有訊息。

- 若要允許來源 SQL Server 資料庫儲存的輸出 EXEC 在一個表中。AWS SCT 創建臨時表和一個額外的過程來模擬此功能。若要使用此模擬，請選取建立其他常式來處理開放式資料集。

移轉考量

將 SQL 伺服器結構描述遷移到 MySQL 時，請考慮以下事項：

- MySQL 不支援 MERGE 聲明 然而，AWS SCT 可以模擬 MERGE 轉換期間的陳述式使用 INSERT ON DUPLICATE KEY 子句 UPDATE FROM and DELETE FROM 陳述

要使用正確的仿真 INSERT ON DUPLICATE KEY，請確定目標 MySQL 資料庫上存在唯一條件約束或主索引鍵。

- 您可以使用 GOTO 用來變更陳述式執行順序的陳述式和標籤。接下來的任何交易 SQL 陳述式 GOTO 語句被跳過，並在標籤處繼續處理。您可以使用 GOTO 程序、批次或陳述式區塊中任何位置的陳述式和標籤。您也可以巢 GOTO 陳述

MySQL 不使用GOTO陳述 何時AWS SCT轉換包含GOTO語句中，它將語句轉換為使用 aBEGIN...END或者LOOP...END LOOP聲明 您可以找到如何執行的範例AWS SCT皈依GOTO下表中的語句。

SQL Server 陳述式	MySQL 陳述式
<pre>BEGIN statement1; GOTO label1; statement2; label1: Statement3; END</pre>	<pre>BEGIN label1: BEGIN statement1; LEAVE label1; statement2; END; Statement3; END</pre>
<pre>BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END</pre>

SQL Server 陳述式

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

MySQL 陳述式

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

- MySQL 不支援多重陳述式資料表值函式。AWS SCT透過建立暫存資料表和重寫陳述式以使用這些暫存資料表，在轉換期間模擬資料表值函數。

將一個 SQL Server 轉換到 PostgreSQL

您可以使用 SQL 伺服器來執行 PostgreSQL 擴充套件AWS SCT。這個擴充套件會模擬轉換後的 PostgreSQL 程式碼中的 SQL 伺服器資料庫函式。使用 SQL 伺服器到 PostgreSQL 延伸套件來模擬 SQL 伺服器代理程式和 SQL 伺服器資料庫郵件。如需延伸功能表的詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

主題

- [將 PostgreSQL 作為目標資料庫的權限](#)
- [SQL 伺服 PostgreSQL 換設定](#)
- [將 SQL 伺服器磁碟分割 PostgreSQL 第 10 版磁碟分割](#)
- [移轉考量](#)
- [使用AWS SCT在 PostgreSQL 中模擬 SQL 伺服器代理程式的擴充套件](#)
- [使用AWS SCT在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件的擴充套件](#)

將 PostgreSQL 作為目標資料庫的權限

要使用 PostgreSQL 作為目標，AWS SCT需要CREATE ON DATABASE任務 請確定您已為每個目標 PostgreSQL 資料庫授與此權限。

若要使用轉換後的公用同義字，請將資料庫預設搜尋路徑變更為"\$user", public_synonyms, public。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在前面的範例中，取代####您的使用者名稱。然後，更換#####使用目標資料庫的名稱。最後，更換####使用安全密碼。

在 PostgreSQL 中，只有結構描述擁有者或superuser可以刪除模式。擁有者可以刪除結構描述和此結構描述包含的所有物件，即使結構描述的擁有者不擁有其某些物件也是如此。

當您使用不同的用戶轉換並將不同的模式應用於目標數據庫時，您可能會在以下情況收到錯誤消息 AWS SCT無法刪除綱要。若要避免此錯誤訊息，請使用superuser角色。

SQL 伺服器 PostgreSQL 換設定

若要編輯 SQL 伺服器 PostgreSQL 換設定，請選擇設定，然後選擇轉換設定。從上方的清單中選擇SQL 伺服器，然後選擇SQL 伺服器。AWS SCT顯示 SQL 伺服器到 PostgreSQL 轉換的所有可用設定。

SQL 伺服器到 PostgreSQL 的轉換設定AWS SCT包括下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

對於在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解」下，選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數量降至最低，請選擇僅限錯誤。若要在轉換的程式碼中包含所有行動項目的註解，請選擇所有訊息。

- 為了允許在 SQL Server 中的不同表中使用具有相同名稱的索引。

在 PostgreSQL 中，您在結構描述中使用的所有索引名稱都必須是唯一的。為了確保AWS SCT為您的所有索引生成唯一的名稱，選擇為索引生成唯一名稱。

- 若要將 SQL 伺服器程序 PostgreSQL 為函式。

第 10 版 PostgreSQL 更早版本不支援程序。對於不熟悉 PostgreSQL 中使用程序的客戶，AWS SCT 可以將程序轉換為函數。若要這麼做，請選取將程序轉換為函數。

- 若要模擬輸出 EXEC 在一個表中。

您的來源 SQL 伺服器資料庫可以儲存的輸出 EXEC 在一個表中。AWS SCT 創建臨時表和一個額外的過程來模擬此功能。若要使用此模擬，請選取建立其他常式來處理開放式資料集。

- 要定義用於轉換後的代碼模式名稱的模板。對於架構名稱產生範本，選擇下列其中一個選項：
 - <source_db>— 在 PostgreSQL 中使用 SQL 伺服器資料庫名稱做為結構描述名稱。
 - <source_schema>— 在 PostgreSQL 中使用 SQL 伺服器結構描述名稱做為結構描述名稱。
 - <source_db>_<schema>— 使用 SQL 伺服器資料庫和結構描述名稱的組合做為 PostgreSQL 中的結構描述名稱。
- 保留源對象名稱的字母大小寫。

若要避免將物件名稱轉換為小寫，請選取避免將區分大小寫的操作轉換為小寫。只有當您在目標資料庫中開啟區分大小寫選項時，此選項才適用。

- 保留來源資料庫中的參數名稱。

若要在轉換後的程式碼中的參數名稱中加入雙引號，請選取保留原始參數名稱。

將 SQL 伺服器磁碟分割 PostgreSQL 第 10 版磁碟分割

當您將 Microsoft SQL 伺服器資料庫轉換為 Amazon Aurora 兼容版本 (Aurora PostgreSQL) 或 Amazon Relational Database Service 服務 (適用於 PostgreSQL) 時，請注意以下事項。

在 SQL Server 中，您可以使用分割功能建立分割區。從 SQL Server 分割資料表轉換到 PostgreSQL 版本 10 分割資料表時，請注意幾個潛在問題：

- SQL Server 可讓您使用沒有 NOT NULL 限制的欄來分割資料表。在這種情況下，所有 NULL 值會移至最左邊分割區。PostgreSQL 不支援 RANGE 分割的 NULL 值。
- SQL Server 允許您建立分割資料表的主要和唯一索引鍵。對於 PostgreSQL，您可以為每個分割區直接建立主要或唯一索引鍵。因此，遷移到 PostgreSQL 時，必須從父資料表移除 PRIMARY 或 UNIQUE KEY 限制。產生的金鑰名稱採用格式 <original_key_name>_<partition_number>。

- SQL Server 允許您從分割資料表建立外部索引鍵限制以及對分割資料表建立外部索引鍵限制。PostgreSQL 不支援參考分割資料表的外部索引鍵。此外，PostgreSQL 也不支援從一個分割資料表參考另一個資料表的外部索引鍵。
- SQL Server 允許您為分割資料表建立索引。對於 PostgreSQL，應為每個分割區直接建立索引。因此，遷移到 PostgreSQL 時，必須從父資料表移除索引。產生的索引名稱採用格式 `<original_index_name>_<partition_number>`。
- PostgreSQL 不支援分割索引。

移轉考量

將 SQL 伺服器結構描述遷移到 PostgreSQL 時需要考慮的一些事項：

- 在 PostgreSQL 中，結構描述中所有物件的名稱都必須是唯一的，包含索引。索引名稱在基礎資料表的結構描述中也必須是唯一的。在 SQL Server 中，不同資料表的索引名稱可以相同。

為了確保索引名稱的唯一性，AWS SCT 如果索引名稱不是唯一的，可讓您選擇產生唯一的索引名稱。做法是在專案屬性中選擇選項 `Generate unique index names` (產生唯一的索引名稱)。此選項預設為啟用。如果啟用此選項，會使用格式 `IX_table_name_index_name` 建立唯一的索引名稱。如果停用此選項，索引名稱不變。

- GOTO 陳述式和標籤可用來變更陳述式的執行順序。GOTO 陳述式後面的任何 Transact-SQL 陳述式都會被略過，並繼續處理標籤。GOTO 陳述式與標籤可以用在程序、批次或陳述式區塊中的任何地方。GOTO 陳述式也可以設為巢狀。

PostgreSQL 不使用 GOTO 陳述式。何時 AWS SCT 轉換包含 GOTO 語句的代碼，它將語句轉換為使用 `BEGIN... 結束` 或 `循環... 結束` 循環語句。您可以找到如何執行的範例 AWS SCT 轉換下表中的 GOTO 語句。

SQL Server GOTO 陳述式與轉換後的 PostgreSQL 陳述式

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

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

- PostgreSQL 不支援合併陳述式。AWS SCT您可以在下列方式上模擬 MERGE 陳述式的行為：

- 透過 INSERT ON CONFLICT 建構。
- 透過使用 UPDATE FROM DML 陳述式，例如不含 WHEN NOT MATCHED 子句的 MERGE。
- 透過使用 CURSOR (例如含有 DELETE 子句的 MERGE) 或使用複雜的 MERGE ON 條件陳述式。
- AWS SCT當 Amazon RDS 為目標時，可以將資料庫觸發器新增至物件樹狀結構。
- AWS SCT當 Amazon RDS 為目標時，可以將伺服器層級觸發程序新增至物件樹狀目錄。
- SQL 伺服器會自動建立和管理deleted和inserted表格。您可以使用這些暫存的記憶體駐留資料表來測試特定資料修改的效果，並設定 DML 觸發程序動作的條件。AWS SCT可以轉換 DML 觸發器語句中這些表的使用情況。
- AWS SCT當 Amazon RDS 是目標時，可以將連結的伺服器新增至物件樹狀目錄。
- 當從 Microsoft SQL Server 遷移到 PostgreSQL 時，內建的 SUSER_SNAME 函數將進行轉換，如下所示：
 - SUSER_SNAME – 傳回與安全性識別碼 (SID) 相關聯的登入名稱。
 - SUSER_SNAME(<server_user_sid>) – 不支援。
 - SUSER_SNAME() CURRENT_USER – 傳回目前執行內容的使用者名稱。
 - SUSER_SNAME(NULL) – 傳回 NULL。
- 支援轉換表格值函數。表格值函數會傳回表格，並可在查詢中代替表格。
- PATINDEX 將傳回指定表達式中模式第一次出現的開始位置，或所有有效的文字和字元資料類型。如果沒有找到該模式，它返回零。當從 SQL 伺服器轉換 Amazon RDS for PostgreSQLAWS SCT<pattern character><expression character varying>以 aws_sqlserver 取代使用 PATINDEX 的應用程式程式碼。
- 在 SQL Server 中，使用者定義資料表類型是一種代表資料表結構定義的類型。您可以使用使用者定義的資料表類型來宣告預存程序或函數的資料表值參數。您也可以使用使用者定義的資料表類型，宣告要在批次或預存程序或函數主體中使用的資料表變數。AWS SCT通過創建一個臨時表在 PostgreSQL 中模擬這種類型。

當從 SQL 伺服器轉換為 PostgreSQL 時，AWS SCT在 PostgreSQL 中將 SQL 伺服器系統物件轉換為可識別的物件。下表顯示系統物件轉換方式。

MS SQL 伺服器使用案例	PostgreSQL 代
SYS.SCHEMAS	AWS_SQLSERVER_EXT.SYS_SCHEMAS

MS SQL 伺服器使用案例	PostgreSQL 代
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABLES
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYSFOREIGNKEYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATABASES

MS SQL 伺服器使用案例	PostgreSQL 代
INFORMATION_SCHEMA .SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA .VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA .TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA .COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA .CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CO NSTRAINTS
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_CONSTRAI NT_TABLE_USAGE
INFORMATION_SCHEMA .CONSTRAINT_COLUMN_ USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAI NT_COLUMN_USAGE
INFORMATION_SCHEMA .ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROUTINES
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROCESSES

MS SQL 伺服器使用案例	PostgreSQL 代
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

使用AWS SCT在 PostgreSQL 中模擬 SQL 伺服器代理程式的擴充套件

SQL 伺服器代理程式是執行 SQL 伺服器作業的 Microsoft 視窗服務。SQL Server 代理程式會在排程上執行工作，以回應特定事件。如需 SQL 伺服器代理程式的詳細資訊，請參閱[Microsoft 技術文件](#)。

PostgreSQL 沒有相等資料。若要模擬 SQL 伺服器代理程式功能，AWS SCT建立擴充功能。此擴展包使用AWS Lambda和亞馬遜 CloudWatch。AWS Lambda實作您用來管理排程和執行工作的介面。亞馬遜 CloudWatch 維護排程規則。

AWS Lambda和亞馬遜 CloudWatch 使用 JSON 參數進行交互。此 JSON 參數具有下列結構。

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

在前面的範例中，*mode*是任務的類型和*list of parameters*是一組參數，取決於工作類型。還*procedure name*是工作完成後執行的程序名稱。

AWS SCT使用一個 Lambda 函數來控制和執行工作。該 CloudWatch 規則會啟動工作的執行，並提供啟動工作所需的資訊。當 CloudWatch 規則觸發器時，它會使用規則中的參數啟動 Lambda 函數。

若要建立呼叫程序的簡單工作，請使用下列格式。

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

若要建立具有數個步驟的工作，請使用下列格式。

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

若要在 PostgreSQL 中模擬 SQL 伺服器代理程式行為，AWS SCT 擴充套件也會建立下列表格和程序。

在 PostgreSQL 中模擬 SQL 伺服器代理程式的資料表

若要模擬 SQL Server 代理程式，擴充功能套件會使用下列資料表：

sysjobs

儲存工作資訊。

系統工作步驟

儲存工作步驟的資訊。

系統排程

儲存工作排程資訊。

系統工作排程

儲存個別工作的排程資訊。

sysjobhistory

儲存排程工作執行資訊。

在 PostgreSQL 中模擬 SQL 伺服器代理程式的程序

若要模擬 SQL Server 代理程式，擴充功能套件會使用下列程序：

sp_add_job

新任務

sp_add_jobstep

將步驟新增至工作。

sp_add_schedule

在亞馬遜中創建一個新的計劃規則 CloudWatch。您可以將此排程用於任意數量的工作。

sp_attach_schedule

設定所選工作的排程。

sp_add_jobschedule

在亞馬遜中為任務創建計劃規則 CloudWatch 並設定此規則的目標。

sp_update_job

更新先前建立之工作的屬性。

sp_update_jobstep

更新工作中步驟的屬性。

sp_update_schedule

更新亞馬遜計劃規則的屬性 CloudWatch。

sp_update_jobschedule

更新指定工作的排程屬性。

sp_delete_job

新任務

sp_delete_jobstep

從工作中刪除作業步驟。

sp_delete_schedule

刪除排程。

sp_delete_jobschedule

從 Amazon 刪除指定任務的排程規則 CloudWatch。

sp_detach_schedule

移除排程與工作之間的關聯。

獲取工作，更新工作

與之互動的內部程序AWS Elastic Beanstalk。

日期, 驗證工作時間, 工作時間, 工作, 工作步驟, 驗證工作步驟, 計劃排程, 驗證工作標識符, 標識符

檢查設定的內部程序。

在 PostgreSQL 中模擬 SQL 伺服器代理程式的程序的語法

該aws_sqlserver_ext.sp_add_job擴展包中的過程模擬msdb.dbo.sp_add_job程序 如需來源 SQL Server 代理程式的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該 `aws_sqlserver_ext.sp_add_jobstep` 擴展包中的過程模擬 `msdb.dbo.sp_add_jobstep` 程序
如需來源 SQL Server 代理程序的詳細資訊，請參閱 [Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_add_schedule擴展包中的過程模

擬msdb.dbo.sp_add_schedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_attach_schedule擴展包中的過程模

擬msdb.dbo.sp_attach_schedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_add_jobschedule擴展包中的過程模

擬msdb.dbo.sp_add_jobschedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_delete_job擴展包中的過程模擬msdb.dbo.sp_delete_job程序
如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_delete_jobstep擴展包中的過程模
擬msdb.dbo.sp_delete_jobstep程序 如需來源 SQL Server 代理程序的詳細資訊，請參
閱[Microsoft 技術文件](#)。

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

該aws_sqlserver_ext.sp_delete_jobschedule擴展包中的過程模
擬msdb.dbo.sp_delete_jobschedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參
閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_delete_schedule擴展包中的過程模

擬msdb.dbo.sp_delete_schedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_detach_schedule擴展包中的過程模

擬msdb.dbo.sp_detach_schedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_update_job擴展包中的過程模擬msdb.dbo.sp_update_job程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_update_jobschedule擴展包中的過程模

擬msdb.dbo.sp_update_jobschedule程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sp_update_jobstep擴展包中的過程模

擬msdb.dbo.sp_update_jobstep程序 如需來源 SQL Server 代理程序的詳細資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該 `aws_sqlserver_ext.sp_update_schedule` 擴展包中的過程模擬 `msdb.dbo.sp_update_schedule` 程序。如需來源 SQL Server 代理程序的詳細資訊，請參閱 [Microsoft 技術文件](#)。

```
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
```

在 PostgreSQL 中使用模擬 SQL 伺服器代理程式的程序的範例

若要新增工作，請使用 `aws_sqlserver_ext.sp_add_job` 程序，如下所示。


```
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');
```

若要新增工作步驟，請使用aws_sqlserver_ext.sp_add_jobstep程序，如下所示。

```
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');
```

若要加入簡單的排程，請使用aws_sqlserver_ext.sp_add_schedule程序，如下所示。

```
SELECT * FROM aws_sqlserver_ext.sp_add_schedule(
    par_schedule_name := 'RunOnce',
    par_freq_type := 1,
    par_active_start_time := 233000);
```

若要設定工作的排程，請使用aws_sqlserver_ext.sp_attach_schedule程序，如下所示。

```
SELECT * FROM aws_sqlserver_ext.sp_attach_schedule (
    par_job_name := 'test_job',
    par_schedule_name := 'NightlyJobs');
```

若要建立工作的排程，請使用aws_sqlserver_ext.sp_add_jobschedule程序，如下所示。

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobschedule (
    par_job_name := 'test_job2',
    par_name := 'test_schedule2',
    par_enabled := 1::smallint,
    par_freq_type := 4,
    par_freq_interval := 1,
    par_freq_subday_type := 4,
```

```
par_freq_subday_interval := 1,  
par_freq_relative_interval := 0,  
par_freq_recurrence_factor := 0,  
par_active_start_date := 20100801,  
par_active_end_date := 99991231,  
par_active_start_time := 0,  
par_active_end_time := 0);
```

在 PostgreSQL 中模擬 SQL 伺服器代理程式的使用案例範例

如果您的來源資料庫程式碼使用 SQL Server 代理程式來執行工作，您可以使用 SQL 伺服器到 PostgreSQL 延伸套件AWS SCT將此代碼轉換為 PostgreSQL。擴展包使用AWS Lambda用來模擬 SQL Server 代理程式行為的函式。

您可以建立新的AWS Lambda函數或註冊現有函數。

建立新 AWS Lambda 函數

1. 在AWS SCT，在目標資料庫樹狀結構中，開啟相關資訊環境定義 (按一下滑鼠右鍵)。套用擴充套件，然後選擇PostgreSQL。

會顯示延伸套件精靈。

2. 在「SQL 伺服器代理程式模擬服務」標籤中執行下所示：
 - 選擇新增複本AWS Lambda功能。
 - 對於「資料庫」中，輸入目標資料庫使用者的名稱。
 - 對於「資料庫密碼」下方，輸入您在上一個步驟中輸入的使用者名稱的密碼。
 - 對於「Python 資料夾」下方，輸入您的 Python 程式庫資料夾的路徑。
 - 選擇創建AWS Lambda功能，然後選擇下一步。

若要新增複本AWS Lambda您之前部署的功能

- 在目標資料庫執行下列指令碼。

```
SELECT  
  FROM aws_sqlserver_ext.set_service_setting(  
    p_service := 'JOB',  
    p_setting := 'LAMBDA_ARN',  
    p_value := ARN)
```

在前面的範例中，**ARN**是部署的亞馬遜資源名稱 (ARN)AWS Lambda功能。

下列範例會建立由一個步驟組成的簡單工作。每五分鐘，此任務運行先前創建的job_example功能。此函數將記錄插入job_example_table表

若要建立這個簡單的工作

1. 使用建立工作aws_sqlserver_ext.sp_add_job功能，如下所示。

```
SELECT
  FROM aws_sqlserver_ext.sp_add_job (
    par_job_name := 'test_simple_job');
```

2. 使用建立作業步驟aws_sqlserver_ext.sp_add_jobstep功能，如下所示。

```
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;');
```

作業步驟指定功能的功能。

3. 使用aws_sqlserver_ext.sp_add_jobschedule功能，如下所示。

```
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 */);
```

作業步驟指定功能的功能。

若要刪除此工作，請使用aws_sqlserver_ext.sp_delete_job功能，如下所示。

```
PERFORM aws_sqlserver_ext.sp_delete_job(
  par_job_name := 'PeriodicJob1'::character varying,
```

```
par_delete_history := 1::smallint,  
par_delete_unused_schedule := 1::smallint);
```

使用AWS SCT在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件的擴充套件

您可以在 SQL Server 資料庫引擎上使用 SQL Server Database Mail，從您的傳送電子郵件訊息給使用者。這些電子郵件訊息可以包含查詢結果，或包含來自網路上任何資源的檔案。如需 SQL 伺服器資料庫郵件的詳細資訊，請參閱[Microsoft 技術文件](#)。

PostgreSQL 沒有相同的 SQL 伺服器資料庫郵件。若要模擬 SQL 伺服器資料庫郵件功能，AWS SCT 建立擴充功能。此擴展包使用AWS Lambda及亞馬遜 Simple Email Service (Amazon SES)。AWS Lambda為使用者提供與 Amazon SES 電子郵件傳送服務互動的介面。若要設定此互動，請在 Lambda 函數上加入 Amazon 資源名稱 (ARN)。

對於新的電子郵件帳戶，請使用以下命令。

```
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;
```

若要將 Lambda 函數的 ARN 新增至現有的電子郵件帳戶，請使用下列命令。

```
do  
$$  
begin  
PERFORM sysmail_update_account_sp (  
    par_account_name := 'existind_account_name',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'ARN'  
);  
end;  
$$ language plpgsql;
```

在前面的例子中，**ARN**是你的 Lambda 函數的 ARN。

若要在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件行為，AWS SCT擴充功能套件使用下列資料表、檢視表及程序。

在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件的資料表

若要模擬 SQL Server 資料庫郵件，擴充功能套件會使用下列資料表：

系統郵件帳戶

儲存電子郵件帳戶資訊。

系統設定檔

儲存使用者資料檔的資訊。

系統郵件伺服器

儲存電子郵件伺服器資訊。

系統郵件項目

儲存電子郵件訊息的清單。

系統郵件附件

每個電子郵件附件均包含一資料列。

系統郵件日誌

儲存有關傳送電子郵件訊息的服務資訊。

系統設定檔帳戶

儲存使用者設定檔和電子郵件帳戶的相關資訊。

模擬 SQL 伺服器資料庫郵件的檢視

若要模擬 SQL 伺服器資料庫郵件，AWS SCT會在 PostgreSQL 資料庫中建立下列檢視，以確保相容性。擴展包不使用它們，但轉換後的代碼可以查詢這些視圖。

sysmail_allitems

包括所有電子郵件的列表。

sysmail_faileditems

包括無法發送的電子郵件列表。

sysmail_sentitems

包括已傳送的電子郵件清單。

sysmail_unsentitems

包括尚未發送的電子郵件列表。

sysmail_mailattachments

包括附加檔案的清單。

模擬 SQL 伺服器資料庫郵件的程序

若要模擬 SQL Server 資料庫郵件，擴充功能套件會使用下列程序：

sp_send_dbmail

傳送電子郵件給指定的收件者。

sysmail_add_profile_sp

建立新的使用者設定檔。

sysmail_add_account_sp

建立儲存簡易郵件傳送通訊協定 (SMTP) 認證等資訊的新電子郵件帳戶。

sysmail_add_profileaccount_sp

將電子郵件帳戶新增至指定的使用者設定檔。

sysmail_update_profile_sp

變更使用者設定檔的屬性，例如說明、名稱等。

sysmail_update_account_sp

變更現有電子郵件帳戶中的資訊。

sysmail_update_profileaccount_sp

更新指定使用者設定檔中的電子郵件帳戶資訊。

sysmail_delete_profileaccount_sp

從指定的使用者設定檔中移除電子郵件帳戶。

sysmail_delete_account_sp

刪除電子郵件帳戶。

sysmail_delete_profile_sp

刪除使用者設定檔。

sysmail_delete_mailitems_sp

從內部表中刪除電子郵件。

sysmail_help_profile_sp

顯示使用者設定檔的相關資訊。

sysmail_help_account_sp

顯示電子郵件帳戶的相關資訊。

sysmail_help_profileaccount_sp

顯示與使用者設定檔相關聯的電子郵件帳號。

系統的數據庫郵件

產生 JSON 要求的內部程序AWS Lambda函數。

系統驗證設定檔 SP, 系統郵件驗證帳戶 SP, 系統郵件驗證位址

檢查設定的內部程序。

SP_ 獲取數據庫郵件, SP_SET_ 數據庫郵件, 系統郵件的

已淘汰的內部程序。

在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件的程序的語法

該aws_sqlserver_ext.sp_send_dbmail擴展包中的過程模擬msdb.dbo.sp_send_dbmail程序
如需來源 SQL Server 資料庫郵件程序的相關資訊, 請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_delete_mailitems_sp擴展包中的過程模

擬msdb.dbo.sysmail_delete_mailitems_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_sent_before timestamp = NULL::timestamp without time zone,  
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

該aws_sqlserver_ext.sysmail_add_profile_sp擴展包中的過程模

擬msdb.dbo.sysmail_add_profile_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```


該aws_sqlserver_ext.sysmail_add_account_sp擴展包中的過程模

擬msdb.dbo.sysmail_add_account_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_add_profileaccount_sp擴展包中的過程模

擬msdb.dbo.sysmail_add_profileaccount_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_help_profile_sp擴展包中的過程模

擬msdb.dbo.sysmail_help_profile_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
out returncode integer
```

該aws_sqlserver_ext.sysmail_update_profile_sp擴展包中的過程模

擬msdb.dbo.sysmail_update_profile_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out returncode integer
```

該aws_sqlserver_ext.sysmail_delete_profile_sp擴展包中的過程模

擬msdb.dbo.sysmail_delete_profile_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out returncode integer
```

該aws_sqlserver_ext.sysmail_help_account_sp擴展包中的過程模

擬msdb.dbo.sysmail_help_account_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

該aws_sqlserver_ext.sysmail_update_account_sp擴展包中的過程模

擬msdb.dbo.sysmail_update_account_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_delete_account_sp擴展包中的過程模

擬msdb.dbo.sysmail_delete_account_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

該aws_sqlserver_ext.sysmail_help_profileaccount_sp擴展包中的過程模

擬msdb.dbo.sysmail_help_profileaccount_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_update_profileaccount_sp擴展包中的過程模

擬msdb.dbo.sysmail_update_profileaccount_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

該aws_sqlserver_ext.sysmail_delete_profileaccount_sp擴展包中的過程模

擬msdb.dbo.sysmail_delete_profileaccount_sp程序 如需來源 SQL Server 資料庫郵件程序的相關資訊，請參閱[Microsoft 技術文件](#)。

```
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
```

在 PostgreSQL 中使用模擬 SQL 伺服器資料庫郵件的程序的範例

若要傳送電子郵件，請使用 `aws_sqlserver_ext.sp_send_dbmail` 程序，如下所示。

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',  
    par_body := 'The stored procedure finished'  
);
```

下列範例說明如何傳送電子郵件訊息給您。

```
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'  
);
```

下面的例子演示了如何發送 HTML 代碼的電子郵件。

```
DECLARE var_tableHTML TEXT;  
SET var_tableHTML := CONCAT(  
    '<H1>Work Order Report</H1>',  
    '<table border="1">',  
    '<tr><th>Work Order ID</th><th>Product ID</th>',  
    '<th>Name</th><th>Order Qty</th><th>Due Date</th>',  
    '<th>Expected Revenue</th></tr>',  
    '</table>'  
);  
PERFORM sp_send_dbmail (  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Work Order List',  
    par_body := var_tableHTML,  
    par_body_format := 'HTML'  
);
```

若要刪除電子郵件，請使用 `aws_sqlserver_ext.sysmail_delete_mailitems_sp` 程序，如下所示。

```
DECLARE var_GETDATE datetime;  
SET var_GETDATE = NOW();  
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_before := var_GETDATE  
);
```

下列範例顯示如何刪除最舊的電子郵件。

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_before := '31.12.2015'  
);
```

下列範例說明如何刪除所有無法傳送的電子郵件訊息。

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_status := 'failed'  
);
```

若要建立新使用者資料檔，請使用aws_sqlserver_ext.sysmail_add_profile_sp程序，如下所示。

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

下列範例說明如何在變數中建立新的設定檔，以及如何在變數中儲存唯一的描述檔識別碼。

```
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;
```

若要建立新的電子郵件帳號，請使用aws_sqlserver_ext.sysmail_add_account_sp程序，如下所示。

```
PERFORM sysmail_add_account_sp (
  par_account_name := 'Audit Account',
  par_email_address := 'dba@rusgl.info',
  par_display_name := 'Test Automated Mailer',
  par_description := 'Account for administrative e-mail.',
  par_mailserver_type := 'AWSLAMBDA'
  par_mailserver_name := 'arn:aws:lambda:us-west-2:555555555555:function:pg_v3'
);
```

若要將電子郵件帳戶新增至使用者設定檔，請使用 `aws_sqlserver_ext.sysmail_add_profileaccount_sp` 程序如下圖所示。

```
PERFORM sysmail_add_profileaccount_sp (
  par_account_name := 'Administrator',
  par_account_name := 'Audit Account',
  par_sequence_number := 1
);
```

在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件的使用案例

如果您的來源資料庫程式碼使用 SQL Server 資料庫郵件來傳送電子郵件，您可以使用 AWS SCT 擴展包將此代碼轉換為 PostgreSQL。

若要從您的 PostgreSQL 資料庫傳送電子郵件

1. 建立與設定您的 AWS Lambda 功能。
2. 套用 AWS SCT 擴展包。
3. 建立使用者設定檔 `sysmail_add_profile_sp` 功能，如下所示。
4. 建立電子郵件帳號 `sysmail_add_account_sp` 功能，如下所示。
5. 將此電子郵件帳戶新增至您的使用者設定檔 `sysmail_add_profileaccount_sp` 功能，如下所示。

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.
proc_dbmail_settings_msdb()
RETURNS void
AS
$BODY$
BEGIN
PERFORM aws_sqlserver_ext.sysmail_add_profile_sp(
  par_profile_name := 'Administrator',
```

```

    par_description := 'administrative mail'
);
PERFORM aws_sqlserver_ext.sysmail_add_account_sp(
    par_account_name := 'Audit Account',
    par_description := 'Account for administrative e-mail.',
    par_email_address := 'dba@rusgl.info',
    par_display_name := 'Test Automated Mailer',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'your_ARN'
);
PERFORM aws_sqlserver_ext.sysmail_add_profileaccount_sp(
    par_profile_name := 'Administrator',
    par_account_name := 'Audit Account',
    par_sequence_number := 1
);
END;
$body$
LANGUAGE plpgsql;

```

6. 傳送電子郵件sp_send_dbmail功能，如下所示。

```

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;

```

若要檢視所有使用者設定檔的相關資訊，請使用sysmail_help_profile_sp程序如下圖所示。

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

下列範例會顯示特定使用者設定檔的相關資訊。

```
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');
```

若要檢視所有電子郵件帳號的相關資訊，請使用sysmail_help_account_sp程序如下圖所示。

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

下列範例顯示特定電子郵件帳戶的相關資訊。

```
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');
```

若要檢視與使用者設定檔相關聯之所有電子郵件帳號的相關資訊，請使用sysmail_help_profileaccount_sp程序如下圖所示。

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

下列範例會依識別碼、設定檔名稱或帳戶名稱來篩選記錄。

```
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');
```

若要變更使用者設定檔名稱或描述，請使用sysmail_update_profile_sp程序如下圖所示。

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
  par_profile_id := 2,
  par_profile_name := 'New profile name'
);
```

若要變更電子郵件帳戶設定，請使用ysmail_update_account_sp程序如下圖所示。

```
select from aws_sqlserver_ext.sysmail_update_account_sp (
```



```
par_account_name := 'Audit Account',
par_mailserver_name := 'arn:aws:lambda:region:XXXXXXXXXXXX:function:func_test',
par_mailserver_type := 'AWSLAMBDA'
);
```

將 SQL Server 轉換到 Amazon RDS for SQL Server

將 SQL Server 結構描述和程式碼遷移到 Amazon RDS for SQL Server 的一些考量事項：

- AWS SCT 可以轉換 SQL 伺服器代理程式，以便在適用於 SQL 伺服器資料庫執行個體的 Amazon RDS 上提供排程、警示和任務。轉換後，您可以將 Amazon RDS for SQL Server 資料庫執行個體搭配 SQL Server Reporting Service (SSRS)、SQL Server Analysis Services (SSAS) 和 SQL Server Integration Services (SSIS) 使用。
- Amazon RDS 目前不支援 SQL Server Service Broker，或其他需要您執行 CREATE ENDPOINT 命令的 T-SQL 端點。
- Amazon RDS 對連結伺服器提供有限的支援。當轉換使用連結伺服器的 SQL Server 應用程式程式碼時，AWS SCT 轉換應用程式程式碼。不過，請務必檢閱使用連結伺服器之物件的行為，然後再執行轉換的程式碼。
- 系統會使用 Always on。
- 該 AWS SCT 評估報告提供轉換的伺服器度量。這類有關 SQL Server 執行個體的各项指標包括：
 - 已使用資料鏡像。
 - 已設定 SQL Server 記錄傳送。
 - 已使用容錯移轉叢集。
 - 已設定 Database Mail。
 - 已使用全文檢索搜尋服務。Amazon RDS for SQL Server 具備有限的全文檢索搜尋功能，而且不支援語意搜尋。
 - 已安裝 Data Quality Service (DQS)。Amazon RDS 不支援 DQS，因此建議您在 Amazon EC2 執行個體上安裝 SQL Server。

SQL 伺服器 RDS 作為目標的權限

若要移轉至 RDS for SQL Server，請建立資料庫使用者，然後授與每個資料庫所需的權限。您可以使用下列程式碼範例。

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

```
USE db_name
CREATE USER user_name FOR LOGIN user_name
GRANT VIEW DEFINITION TO user_name
GRANT VIEW DATABASE STATE TO user_name
GRANT CREATE SCHEMA TO user_name;
GRANT CREATE TABLE TO user_name;
GRANT CREATE VIEW TO user_name;
GRANT CREATE TYPE TO user_name;
GRANT CREATE DEFAULT TO user_name;
GRANT CREATE FUNCTION TO user_name;
GRANT CREATE PROCEDURE TO user_name;
GRANT CREATE ASSEMBLY TO user_name;
GRANT CREATE AGGREGATE TO user_name;
GRANT CREATE FULLTEXT CATALOG TO user_name;
GRANT CREATE SYNONYM TO user_name;
GRANT CREATE XML SCHEMA COLLECTION TO user_name;
```

在前面的範例中，取代####您的使用者名稱。然後，更換#####使用目標資料庫的名稱。最後，更換###使用安全密碼。

的資料倉儲來源 AWS Schema Conversion Tool

AWS SCT可以將下列來源資料倉儲的結構描述轉換為支援的目標。如需有關權限、連線以及AWS SCT可轉換哪些項目以搭配目標資料庫或資料倉儲使用的資訊，請參閱下列詳細資訊。

主題

- [使用亞馬遜紅移作為來源 AWS SCT](#)
- [使用 Azure 突觸分析作為來源 AWS SCT](#)
- [使用 BigQuery 做為 AWS SCT 的來源](#)
- [使用綠梅數據庫作為來源 AWS SCT](#)
- [使用尼泰扎作為來源 AWS SCT](#)
- [使用 Oracle 資料倉儲作為下列項目的來源 AWS SCT](#)
- [使用雪花作為來源 AWS SCT](#)
- [使用微軟 SQL 伺服器資料倉儲作為來源 AWS SCT](#)
- [使用太數據作為源 AWS SCT](#)
- [使用 Vertica 作為來源 AWS SCT](#)

使用亞馬遜紅移作為來源 AWS SCT

您可以使用AWS SCT來最佳化您的亞馬遜紅移叢集。AWS SCT為您提供有關 Amazon Redshift 叢集分發和排序金鑰選擇的建議。您可以將 Amazon Redshift 優化專案視為一個AWS SCT專案，其來源和目標指向不同的 Amazon Redshift 叢集。

作為來源資料庫的亞馬遜紅移權限

使用亞馬遜紅移作為來源需要以下權限：

- USAGE ON SCHEMA *<schema_name>*
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON PG_CATALOG.PG_STATISTIC
- SELECT ON SVV_TABLE_INFO
- SELECT ON TABLE STV_BLOCKLIST
- SELECT ON TABLE STV_TBL_PERM
- 在無系統伺服器使用上選取
- 在資料庫資訊上選擇
- 在 PG_統計上選擇

在上述範例中，將<schema_name>預留位置取代為來源結構描述的名稱。

如需將 Amazon 紅移作為目標所需的權限，請參閱[亞馬遜紅移作為目標的許可](#)。

連接到亞馬遜紅移作為源

請使用下列程序連線到您的 Amazon Redshift 來源資料庫。AWS Schema Conversion Tool

連線到亞馬遜紅移來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇亞馬遜紅移，然後選擇下一步。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：

1. 在「機AWS密」中，選擇密碼的名稱。
2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要輸入 Amazon Redshift 來源資料庫的連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入亞馬遜紅移資料庫的名稱。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊：</p> <ul style="list-style-type: none"> • 驗證伺服器憑證：選取此選項可使用信任存放區驗證伺服器憑證。 • 信任存放區：包含憑證的信任存放區的位置。要使此位置顯示在此處，請確保將其添加到「全局」設置中。 <p>如需 Amazon Redshift SSL 支援的詳細資訊，請參閱設定連線的安全選項。</p>
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。

參數	動作
紅移驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

亞馬遜紅移優化設置

若要編輯 Amazon Redshift 最佳化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇亞馬遜紅移，然後選擇亞馬遜紅移-亞馬遜紅移。AWS SCT顯示亞馬遜紅移優化的所有可用設置。

亞馬遜紅移優化設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使表格數目超過 Amazon Redshift 叢集所能存放的數量也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程

式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 若要選擇移轉策略。

AWS建議使用不同的叢集做為最佳化專案的來源和目標。在亞馬遜紅移優化程序開始之前，您需要建立一份來源亞馬遜紅移叢集的副本。您可以將來源資料包含在此副本中，或建立空叢集。

對於移轉策略，請選擇移轉至副本，將來源叢集中的資料包含在目標叢集中。

對於移轉策略，請選擇 [移轉至乾淨] 以檢閱最佳化建議。接受這些建議後，請將來源資料移轉至目標叢集。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有在選取 [使用壓縮編碼] 選項時，才能選取此選項。

- 要使用自動表優化工作。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要在自動表格最佳化上使用，請選擇左窗格中的 [最佳化策略]。然後選取 [使用 Amazon Redshift 自動表格調整]，並針對 [初始金鑰選取策略] 選取選取選取選取

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇您要分析其查詢統計資料的資料庫使用者。

使用 Azure 突觸分析作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Azure 突觸分析轉換為亞馬遜紅移。

Azure 突觸分析作為來源資料庫的權限

使用 Azure 突觸分析資料倉儲作為來源需要下列權限：

- VIEW DEFINITION
- VIEW DATABASE STATE

針對要轉換其結構描述的每個資料庫套用權限。

連接到 Azure 突觸分析作為源

使用下列程序，使用連線到 Azure 突觸分析資料倉儲。AWS Schema Conversion Tool

連線至 Azure 突觸分析資料倉儲做為來源

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 Azure 突觸分析，然後選擇下一步。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Azure 突觸分析資料倉儲的連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入您來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。
SQL 集區	輸入 Azure SQL 集區的名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
Use SSL (使用 SSL)	選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊： <ul style="list-style-type: none"> • 信任伺服器憑證：選擇此選項可信任伺服器憑證。 • 信任存放區：您在全域設定中設定的信任存放區。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。通過打開此選項，您可以存儲數據庫密碼並快速連接到數據庫，而無需輸入密碼。

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

Azure 突觸分析到亞馬遜紅移轉換設置

若要將 Azure 突觸分析編輯為 Amazon Redshift 轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇 Azure 突觸，然後選擇 Azure 突觸-亞馬遜紅移。AWS SCT顯示 Azure 突觸分析到亞馬遜紅移轉換的所有可用設置。

Azure 突觸分析到亞馬遜紅移轉換設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

Azure 突觸分析到亞馬遜紅移轉換優化設置

要將 Azure 突觸分析編輯為 Amazon Redshift 轉換優化設置，請選擇中的設置AWS SCT，然後選擇轉換設置。從上面的列表中選擇 Azure 突觸，然後選擇 Azure 突觸-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示 Azure 突觸分析到亞馬遜紅移轉換的轉換優化設置。

Azure 突觸分析到亞馬遜紅移轉換優化設置AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用 BigQuery 做為 AWS SCT 的來源

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從BigQuery轉換為 Amazon Redshift。

BigQuery作為來源的權限

若要在中使用BigQuery資料倉儲作為來源AWS SCT，請建立服務帳戶。在 GCP 中，應用程式會使用服務帳戶進行授權的 API 呼叫。服務帳戶與使用者帳戶不同。如需詳細資訊，請參閱 Google Cloud 身分識別與存取管理說明文件中的[服務帳戶](#)。

請確定您已將下列角色授與您的服務帳戶：

- BigQuery Admin
- Storage Admin

此BigQuery Admin角色提供管理專案內所有資源的權限。AWS SCT使用此角色在移轉專案中載入中BigQuery繼資料。

此Storage Admin角色會授與資料物件和值區的完整控制權。您可以在下找到此角色Cloud Storage。AWS SCT使用此角色從中擷取您的資料，BigQuery然後將其載入 Amazon Redshift。

若要建立服務帳戶金鑰檔案

1. 請登入谷歌雲端管理主控台，網址為 <https://console.cloud.google.com/>。

2. 在 [\[BigQueryAPI\]](#) 頁面上，選擇 [\[啟用\]](#)。如果您看到「已啟用 API」，請略過此步驟。
3. 在 [\[服務帳戶\]](#) 頁面上，選擇您的專案，然後選擇 [\[建立服務帳戶\]](#)。
4. 在 [\[服務帳戶詳細資料\]](#) 頁面上，輸入 [\[服務帳戶名稱\]](#) 的描述性值。選擇建立並繼續。[\[將此服務帳戶存取權授予專案\]](#) 頁面隨即開啟。
5. 針對 [\[選取角色\]](#)，選擇 BigQuery，然後選擇 [\[BigQuery管理員\]](#)。
6. 選擇 [\[新增其他角色\]](#)。針對 [\[選取角色\]](#)，選擇 [\[雲端儲存\]](#)，然後選擇 [\[儲存空間管理員\]](#)。
7. 選擇「繼續」，然後選擇「完成」。
8. 在 [\[服務帳戶\]](#) 頁面上，選擇您建立的服務帳戶。
9. 選擇 [\[金鑰\]](#)，然後選擇 [\[建立新金鑰\]](#) 做為 [\[新增金鑰\]](#)。
10. 選擇 [\[JSON\]](#)，然後選擇 [\[建立\]](#)。選擇要儲存私密金鑰的資料夾，或在瀏覽器中選擇要下載的預設資料夾。

若要從資料倉儲擷取BigQuery資料，請AWS SCT使用 GCP 儲存貯體資料夾。開始資料移轉之前，請先建立此儲存貯體。在 [\[建立本機工作\]](#) 對話方塊中，輸入 GCP 儲存貯體資料夾的路徑。如需詳細資訊，請參閱[建立、執行和監視 AWS SCT 工作](#)。

BigQuery作為來源連線到

請使用下列程序連接到您的來源BigQuery專案AWS Schema Conversion Tool。

若要連線至來BigQuery源資料倉儲

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
 2. 選擇 BigQuery，然後選擇「下一步」。
- 這時系統顯示「添加源」對話框。
3. 在「連線名稱」中，輸入BigQuery專案的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
 4. 針對金鑰路徑，輸入服務帳戶金鑰檔案的路徑。若要取得有關建立此檔案的更多資訊，請參閱[BigQuery作為來源的權限](#)。
 5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源BigQuery專案。
 6. 選擇「連接」以連接到源BigQuery項目。

BigQuery作為來源使用的限制 AWS SCT

當BigQuery作為來源使用時，會有下列限制AWS SCT：

- AWS SCT不支持分析函數中子查詢的轉換。
- 您不能使用AWS SCT轉換BigQuerySELECT AS STRUCT和SELECT AS VALUE語句。
- AWS SCT不支援下列函數類型的轉換：
 - 近似彙總
 - 位元
 - 除錯
 - 同盟查詢
 - 地理
 - 雜湊
 - 數學
 - 淨值
 - 統計彙總
 - UUID
- AWS SCT對字串函數的轉換提供有限的支援。
- AWS SCT不支持UNNEST運算符的轉換。
- 您無法在AWS SCT中轉換相關聯結作業。
- AWS SCT不支援QUALIFY、WINDOW、和OFFSET子句LIMIT的轉換。
- 您無法使用AWS SCT來轉換遞迴通用資料表運算式。
- AWS SCT不支持使用子INSERT句內的子查詢轉換語VALUES句。
- AWS SCT不支援巢狀欄位和重複記錄的UPDATE陳述式轉換。
- 您不能使用AWS SCT轉換STRUCT和ARRAY數據類型。

BigQuery到亞馬遜紅移轉換設置

若BigQuery要編輯 Amazon Redshift 轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇谷歌 BigQuery，然後選擇谷歌 BigQuery-亞馬遜紅移。AWS SCT顯示亞馬遜紅移轉換的BigQuery所有可用設定。

BigQuery到亞馬遜紅移轉換設置，AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

BigQuery到亞馬遜紅移轉換優化設置

若BigQuery要編輯 Amazon Redshift 轉換優化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇谷歌 BigQuery，然後選擇谷歌 BigQuery-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示轉換為亞馬遜紅移轉BigQuery換的轉換優化設定。

BigQuery到亞馬遜紅移轉換優化設置，AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用綠梅數據庫作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Greenplum 資料庫轉換為 Amazon Redshift。

綠梅資料庫作為來源的權限

使用 Greenplum 數據庫作為來源需要以下權限：

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<schema_name>*
- 選取「開啟」 *<schema_name>*# *<table_name>*
- 選取「依序」 *<schema_name>*# *<sequence_name>*

在前面的範例中，取代預留位置如下：

- 將#####取代為來源資料庫的名稱。
- 將#####取代為來源結構描述的名稱。
- 將####替換為源表的名稱。
- 將####替換為序列名稱的名稱。

連接到綠色數據庫作為源

使用下列程序連線到 Greenplum 來源資料庫。AWS SCT

若要連接到 Greenplum 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 SAP ASE，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Greenplum 來源資料庫認證，請使用下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 Greenplum 資料庫的名稱。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT 只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設 AWS SCT 不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊：</p> <ul style="list-style-type: none"> 驗證伺服器憑證：選取此選項可使用信任存放區驗證伺服器憑證。 信任存放區：包含憑證的信任存放區的位置。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
綠色數據庫驅動程序路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

- 選擇「測試連線」以確認是否 AWS SCT 可以連線到來源資料庫。
- 選擇「連線」以連線到來源資料庫。

綠色到亞馬遜紅移轉換設置

若要編輯綠色到 Amazon Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇綠色，然後選擇綠色-亞馬遜紅移。AWS SCT顯示綠色到亞馬遜紅移轉換的所有可用設定。

綠色到亞馬遜紅移轉換設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

綠色到亞馬遜紅移轉換優化設置

若要編輯 Greenplum 到 Amazon Redshift 的轉換優化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇綠色，然後選擇綠色-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示綠色到亞馬遜紅移轉換的轉換優化設置。

綠色到亞馬遜紅移轉換優化設置AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱[Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用尼泰扎作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Netezza 轉換為亞馬遜紅移。

尼泰扎作為來源的特權

使用 Netezza 作為來源需要以下權限：

- 在系統定義架構. 系統視圖上選擇
- 在系統定義架構表上選擇
- 在系統定義管理表上選擇
- 上的清單 <database_name>
- 上的清單 <schema_name>
- 所有<database_name>. 表格上的清單
- .all. 外<database_name>部表格上的清單
- 列表上的所有<database_name>.
- <database_name>.all. 具體化視觀表上的清單
- 列表上的所有<database_name>. 程序
- 列表上的所有<database_name>. 序列
- 列表上的所有功<database_name>能
- 列表上的所有<database_name>. 聚合

在前面的範例中，取代預留位置如下：

- 將#####取代為來源資料庫的名稱。
- 將#####取代為來源結構描述的名稱。

AWS SCT需要存取下列系統表格和檢視表。您可以授與這些物件的存取權，而不是授與上述清單system.definition_schema.system tables中的system.definition_schema.system view和存取權。

- 在系統定義架構上選擇。_彙總
- 在系統定義架構上選擇。_T類
- 在系統定義架構上選擇。_限制
- 在系統定義架構上選擇。_t常量_雷拉特
- 在系統定義架構上選擇。_t_T資料庫
- 在系統定義架構上選擇。_t_grpobj_私人
- 在系統定義架構上選擇。_格普斯
- 在系統定義架構上選擇。_吉斯特_配置
- 在系統定義架構上選擇。_對象
- 在系統定義架構上選擇。_對象類
- 在系統定義架構上選擇。_程序
- 在系統定義架構上選擇。_T型
- 在系統定義架構上選擇。_T用戶
- 在系統定義架構上選擇。_t_u羅比_普里夫
- 在系統定義架構上選擇。_vt_序列
- 在系統定義架構上選擇。_彙總
- 在系統定義架構上選擇。_v_約束_從屬
- 在系統定義架構上選擇。_v_資料庫
- 在系統定義架構上選擇。_v_數據類型
- 在系統定義架構上選擇。_v_德斯利斯
- 在系統定義架構上選擇。_v_函數
- 在系統定義架構上選擇。_群組

- 在系統定義架構上選擇。 `_` 對象關係
- 在系統定義架構上選擇。 `_v_` 對象關係 XDB
- 在系統定義架構上選擇。 `_V` 程序
- 在系統定義架構上選擇。 `_V` 關係欄
- 在系統定義架構上選擇。 `_V` 關係密鑰數據
- 在系統定義架構上選擇。 `_v_` 雷洛比類
- 在系統定義架構上選擇。 `_` 模式 `_xdb`
- 在系統定義架構上選擇。 `_V` 序列
- 在系統定義架構上選擇。 `_v_` 同義詞
- 在系統定義架構上選擇。 `_V` 系統資訊
- 在系統定義架構上選擇。 `_v_sy_` 約束
- 在系統定義架構上選擇。 `_` 系統對象資訊
- 在系統定義架構上選擇。 `_v_sy_user`
- 在系統定義架構上選擇。 `_V` 表
- 在系統定義架構上選擇。 `_` 表格約束
- 在系統定義架構上選擇。 `_` 表格距離映射
- 在系統定義架構上選擇。 `_V` 表格組織欄
- 在系統定義架構上選擇。 `_V` 表格儲存空間
- 在系統定義架構上選擇。 `_V` 使用者
- 在系統定義架構上選擇。 `_` 視圖
- 在系統信息模式上選擇。 `_V` 關係欄
- 在系統信息模式上選擇。 `_V` 表
- 在 \$ 歷史列訪問 * 上選擇

連接到尼泰扎作為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Netezza 來源資料庫。

若要連接到 Netezza 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇下一步，然後選擇下一步。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：

1. 在「機AWS密」中，選擇密碼的名稱。
2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Netezza 來源資料庫連線資訊，請遵循下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
內泰扎驅動程序路徑	輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱 下載所需資料庫驅動程式 。 如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱 在全域設定中儲存驅動程式路徑 。

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。

6. 選擇「連線」以連線到來源資料庫。

設定持續的資料複製

轉換 Netezza 資料庫結構描述並將其套用至 Amazon Redshift 資料庫之後，您就可以使用資料擷取代理程式遷移資料。代理程式會擷取您的資料並將其上傳到您的 Amazon S3 儲存貯體。然後，您可以使 AWS SCT 用將資料從亞馬遜 S3 複製到亞馬遜紅移。

如果來源資料庫中的資料在移轉過程中發生變更，您可以使用 AWS SCT 資料擷取代理程式擷取持續的變更。然後，您可以在完成初始資料移轉之後，在目標資料庫中複寫這些持續的變更。此程序稱為持續資料複寫或變更資料擷取 (CDC)。

設定從 Netezza 遷移到亞馬遜紅移的持續資料複寫

1. 在源數據庫中，創建一個歷史數據庫。您可以在 Netezza 命令列介面 (CLI) 中使用下列程式碼範例。

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

在前面的例子中，用您的 `#####_database_name`。接下來，將 `load_user` 取代為您定義將歷史記錄資料載入資料庫的使用者名稱。然後，將 `histdb_owner` 替換為您定義為歷史數據庫所有者的用戶的名稱。請確定您已建立此使用者並授與 CREATE DATABASE 權限。最後，用安全的 `#####`。

2. 設定歷程記錄。若要這麼做，請使用下列程式碼範例。

```
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;
```

在前面的例子中，將歷史 `#####`。接下來，將 `load_user` 取代為您定義將歷史記錄資料載入資料庫的使用者名稱。然後，用安全 `#####`。

3. 授予歷史記錄數據庫中所有表的讀取權限。您可以使用下列程式碼範例來授與 SELECT 權限。

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```


在前面的例子中，用您的##### `_database_name`。接下來，將 `your_user` 替換為用戶的名稱，以使用最小的權限來使用您的 Netezza 數據庫。您可以在中使用此資料庫使用者的認證AWS SCT。

4. 收集來源結構描述中每個表格的統計資料，以取得資料欄基數的相關資訊。您可以使用以下命令在歷史數據庫中生成統計信息。

```
GENERATE STATISTICS on "schema_name".table_name";
```

在上述範例中，請將 `schema_name #####` 取代為資料庫結構描述和資料表的名稱。

5. 請確定您已執行下列查詢完成先決條件：

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

在前面的例子中，用您的#####和##### `_schema_name`。接下來，將 `N` 替換為歷史記錄數據庫的版本號。如需有關記錄資料庫版本的詳細資訊，請參閱 [IBM Netezza 說明文件](#)。

6. 安裝資料擷取代理程式。如需詳細資訊，請參閱[安裝擷取代理](#)。

請確定所有擷取程式執行個體的 `settings.properties` 檔案中的 `{working.folder}` 參數都指向相同的資料夾。在這種情況下，您的擷取器可以協調 CDC 工作階段，並針對所有子工作使用單一交易點。

7. 註冊您的資料擷取代理程式。如需詳細資訊，請參閱[將擷取代理程式註冊到 AWS Schema Conversion Tool](#)。
8. 創建您的 CDC 任務。如需詳細資訊，請參閱[建立、執行和監視 AWS SCT 工作](#)。
 - a. 在中開啟您的專案AWS SCT。在左窗格中，選擇您的來源表格。開啟關聯式 (按一下右鍵) 功能表，然後選擇「建立本機工作」。
 - b. 在「工作名稱」中，輸入資料移轉工作的描述性名稱。
 - c. 對於移轉模式，請選擇 [擷取]、[上傳] 和 [複製]。
 - d. 選取「啟用 CDC」。
 - e. 選擇 CDC 設定索引標籤，並定義 CDC 工作階段的範圍和排程。
 - f. 選擇 [測試任務] 以確認您可以連線到工作資料夾、Amazon S3 儲存貯體和 Amazon Redshift 資料倉儲。

- g. 選擇 [建立] 以建立您的工作。
 - h. 選擇 [工作] 索引標籤，從清單中選擇您的工作，然後選擇 [開始]。
9. AWS SCT工作會維護目標資料庫上的交易一致性。資料擷取代理程式會以交易 ID 順序複寫來自來源的交易。

如果您停止任何移轉工作階段或失敗，CDC 處理也會停止。

內泰扎到亞馬遜紅移轉換設置

若要編輯 Netezza 到亞馬遜 Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇尼泰扎，然後選擇尼泰扎 — 亞馬遜紅移。AWS SCT顯示所有可用的設定，以便將 Netezza 轉換為亞馬遜紅移。

亞馬遜到亞馬遜紅移轉換設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

亞馬遜紅移轉換優化設置

若要編輯 Netezza 到 Amazon Redshift 的轉換優化設定，請選擇中的設定 AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇尼泰扎，然後選擇尼泰扎 — 亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT 顯示奈泰扎到亞馬遜紅移轉換的轉換優化設置。

Netezza 到亞馬遜紅移轉換優化設置 AWS SCT 包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選擇選擇選擇策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT 定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以 AWS 移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT 將 ALL 分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。

- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用 Oracle 資料倉儲作為下列項目的來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Oracle 資料倉儲轉換為 Amazon Redshift 或 Amazon Redshift，並結合使AWS Glue用。

Oracle 資料倉儲作為來源的權限

使用「Oracle 資料倉儲」作為來源時，需要下列權限：

- 連線
- select_catalog_role
- 選取任何字典

連線至 Oracle 資料倉儲作為來源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Oracle 資料倉儲來源資料庫。

連線至 Oracle 資料倉儲來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 Oracle，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：

1. 在「機AWS密」中，選擇密碼的名稱。
2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Oracle 來源資料倉儲連線資訊，請遵循下列指示：

參數	動作
類型	<p>選擇連接到您資料庫的連線類型。根據您的類型，提供以下額外資訊：</p> <ul style="list-style-type: none"> • SID <ul style="list-style-type: none"> • 伺服器名稱：來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。 • 伺服器連接埠：用於連接到來源資料庫伺服器的連接埠。 • Oracle SID：Oracle 系統 ID (SID)。若要尋找 Oracle SID，請提交以下查詢至您的 Oracle 資料庫： <pre>SELECT sys_context('userenv','instance_name') AS SID FROM dual;</pre> • 服務名稱 <ul style="list-style-type: none"> • 伺服器名稱：您來源資料庫伺服器的 DNS 名稱或 IP 地址。 • 伺服器連接埠：用於連接到來源資料庫伺服器的連接埠。 • 服務名稱：要連接的 Oracle 服務名稱。 • TNS 別名 <ul style="list-style-type: none"> • TNS 檔案路徑：包含透明網路底層 (TNS) 名稱連線資訊之檔案的路徑。 • TNS 檔案路徑：來自此檔案，用於連接至來源資料庫的 TNS 別名。 • TNS 連線識別碼 <ul style="list-style-type: none"> • TNS 連線識別碼：已註冊 TNS 連線資訊的識別碼。

參數	動作
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
Use SSL (使用 SSL)	<p>選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊：</p> <ul style="list-style-type: none"> • SSL 驗證：選取此選項可對連線使用 SSL 驗證。 • 信任存放區：包含憑證的信任存放區的位置。 • 金鑰存放區：包含私密金鑰和憑證的金鑰存放區位置。如果選取 SSL 驗證，則需要此值，否則為選用值。
儲存密碼	<p>AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。</p>
Oracle 驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

甲骨文資料倉儲到亞馬遜紅移轉換設定

若要編輯 Oracle 資料倉儲到 Amazon Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇甲骨文，然後選擇甲骨文-亞馬遜紅移。AWS SCT顯示 Oracle 資料倉儲到亞馬遜紅移轉換的所有可用設定。

Oracle 資料倉儲到亞馬遜 Redshift 的轉換設定AWS SCT包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 若要轉換資料類型格式化函數TO_CHAR，例如TO_DATE、和 TO_NUMBER Amazon Redshift 不支援的日期時間格式元素。根據預設，AWS SCT會使用延伸模組函式來模擬轉換後的程式碼中這些不受支援的格式元素的使用方式。

與亞馬遜 Redshift 中的日期時間格式字符串相比，Oracle 中的日期時間格式模型包含更多元素。如果原始程式碼只包含 Amazon Redshift 支援的日期時間格式元素，則轉換後的程式碼中不需要擴充套件函數。若要避免在轉換後的程式碼中使用擴充套件函數，請選取您在 Oracle 程式碼中使用的日

期類型格式元素類似於 Amazon Redshift 中的日期時間格式字串。在這種情況下，轉換後的代碼的工作速度更快。

與亞馬遜 Redshift 中的數字格式字符串相比，Oracle 中的數字格式模型包含更多元素。如果原始程式碼只包含 Amazon Redshift 支援的數值格式元素，則轉換後的程式碼中不需要擴充套件函數。若要避免在轉換後的程式碼中使用擴充套件函數，請選取您在 Oracle 程式碼中使用的數值格式元素類似於 Amazon Redshift 中的數值格式字串。在這種情況下，轉換後的代碼的工作速度更快。

- 轉換甲骨文 LEAD 和 LAG 分析函數。依預設，AWS SCT 會引發每個 LEAD 和 LAG 函數的行動項目。

當您的源代碼不使用這些函數中的 offset 的默認值時，AWS SCT 可以模擬這些函數與該 NVL 函數的使用。若要這麼做，請選取 [使用 NVL 函數來模擬甲骨文領導程式和 LAG 函數的行為]。

- 若要模擬 Amazon Redshift 叢集中主金鑰和唯一金鑰的行為，請選取「模擬主金鑰和唯一金鑰的行為」。

Amazon Redshift 不會強制執行唯一金鑰和主索引鍵，而且只會將它們用於資訊目的。如果您在程式碼中使用這些條件約束，請確定在轉換後的程式碼中 AWS SCT 模擬其行為。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT 使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的 [壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

甲骨文資料倉儲到亞馬遜紅移轉換優化設定

若要編輯 Oracle 資料倉儲到 Amazon Redshift 的轉換優化設定，請在中選擇 [設定] AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇甲骨文，然後選擇甲骨文-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT 顯示 Oracle 資料倉儲到亞馬遜紅移轉換的轉換優化設定。

Oracle 資料倉儲到 Amazon Redshift 的轉換優化設定 AWS SCT 包含下列選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用雪花作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從雪花轉換為 Amazon Redshift。

「雪花」作為來源資料庫的權限

您可以使用角色和SECURITYADMIN工作階段前後關聯來建立具有權限的角色，並將使用者的名稱授與此SECURITYADMIN角色。

下列範例會建立最低權限，並將其授與使min_privs用者。

```
create role role_name;  
grant role role_name to role sysadmin;  
grant usage on database db_name to role role_name;  
grant usage on schema db_name.schema_name to role role_name;  
grant usage on warehouse datawarehouse_name to role role_name;  
grant monitor on database db_name to role role_name;  
grant monitor on warehouse datawarehouse_name to role role_name;  
grant select on all tables in schema db_name.schema_name to role role_name;  
grant select on future tables in schema db_name.schema_name to role role_name;  
grant select on all views in schema db_name.schema_name to role role_name;  
grant select on future views in schema db_name.schema_name to role role_name;  
grant select on all external tables in schema db_name.schema_name to role role_name;  
grant select on future external tables in schema db_name.schema_name to role role_name;  
grant usage on all sequences in schema db_name.schema_name to role role_name;  
grant usage on future sequences in schema db_name.schema_name to role role_name;  
grant usage on all functions in schema db_name.schema_name to role role_name;  
grant usage on future functions in schema db_name.schema_name to role role_name;  
grant usage on all procedures in schema db_name.schema_name to role role_name;  
grant usage on future procedures in schema db_name.schema_name to role role_name;  
create user min_privs password='real_user_password'  
DEFAULT_ROLE = role_name DEFAULT_WAREHOUSE = 'datawarehouse_name';  
grant role role_name to user min_privs;
```

在前面的範例中，取代預留位置如下：

- *role_name*以唯讀權限的角色名稱取代。
- *db_name*以來源資料庫的名稱取代。
- *schema_name*以來源結構描述的名稱取代。
- 取代*datawarehouse_name*為所需資料倉儲的名稱。
- *min_privs*以具有最低權限的使用者名稱取代。

DEFAULT_ROLE和DEFAULT_WAREHOUSE參數是按鍵敏感的。

設定對亞馬遜 S3 的安全存取

Amazon S3 儲存貯體的安全和存取管理政策允許 Snowflake 存取、讀取資料，以及將資料寫入 S3 儲存貯體。您可以使用雪花STORAGE INTEGRATION物件類型，設定對私有 Amazon S3 儲存貯體的安全存取。雪花儲存體整合物件會將驗證責任委派給 Snowflake 身分識別與存取管理實體。

如需詳細資訊，請參閱[雪花文件中的設定雪花儲存整合以存取 Amazon S3](#)。

連接到雪花作為源

使用下列程序來使用連線到來源資料庫AWS Schema Conversion Tool。

連線至雪花來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇雪花，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Snowflake 來源資料倉儲連線資訊，請遵循下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入雪花資料庫的名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有在您明確要求時，才會以加密格式儲存您的密碼。
Use SSL (使用 SSL)	如果您要使用安全通訊端層 (SSL) 連線到資料庫，請選擇此選項。在 SSL 索引標籤上提供下列適用的其他資訊： <ul style="list-style-type: none"> • 私密金鑰路徑：私密金鑰的位置。

參數	動作
	<ul style="list-style-type: none"> • 複雜密碼：私密金鑰的複雜密碼。 <p>如需 Snowflake SSL 支援的詳細資訊，請參閱設定連線的安全性選項。</p>
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。如果您設定這個選項，您可以儲存資料庫密碼。這樣做意味著您可以快速連接到數據庫，而無需輸入密碼。
雪花驅動路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

雪花作為來源的限制

以下是使用 Snowflake 作為來源時的限制AWS SCT：

- 物件識別元在物件類型和父物件的前後關聯內必須是唯一的：

資料庫

結構描述識別碼在資料庫中必須是唯一的。

結構描述

資料表和檢視表等物件識別碼在結構描述中必須是唯一的。

表/檢視

資料行識別碼在資料表中必須是唯一的。

- 大型和 xlarge 叢集節點類型的資料表數量上限為 9,900。對於 8xlarge 叢集節點類型，資料表的數目上限為 100,000 個。此限制包括使用者定義和 Amazon Redshift 在查詢處理或系統維護期間建立的臨時資料表。如需詳細資訊，請參閱[亞馬遜紅移叢集管理指南](#)中的配額。

- 對於存儲過程，輸入和輸出參數的最大數量是 32。

雪花的來源資料類型

接下來，您可以找到使用時支援的雪花來源資料類型，以AWS SCT及 Amazon Redshift 目標的預設對應。

雪花數據類型	亞馬遜紅移數據類型
NUMBER	數字的 (38)
NUMBER(p)	如果 p 是 = <4，則小麥林 如果 p 是 => 5 和 = < 9，則整數 如果 p 是 => 10 和 = < 18，那麼大 如果 p 是 => 19 則數字 (p)
數字 (p, 0)	如果 p 是 = <4，則小麥林 如果 p 是 => 5 和 = < 9，則整數 如果 p 是 => 10 和 = < 18，那麼大 如果 p 是 => 19，那麼：數字 (p,0)
數目 (p, s)	如果 p 是 => 1 和 = < 38，如果 s 是 => 1 和 = < 37，那麼 NUMERIC(p,s)
FLOAT	FLOAT
TEXT 萬國碼字元最多可達 16,777,216 個位元組；每個字元最多 4 個位元組。	水平 (最大)
文字 (p)	如果 p 是 = <65,535 那麼，瓦爾查爾 (p)

雪花數據類型	亞馬遜紅移數據類型
萬國碼字元最多可達 65,535 個位元組；每個字元最多 4 個位元組。	
文字 (p) 萬國碼字元最多可達 16,777,216 個位元組；每個字元最多 4 個位元組。	如果 p 是 => 65,535 和 = <16,777,216 然後，瓦爾查爾 (最大值)
BINARY 單一位元組字元最多可達 8,388,608 個位元組；每個字元 1 個位元組。	水平 (最大)
二進制 (p) 單位元組字元最多可達 65,535 個位元組；每個字元 1 個位元組。	瓦爾查爾 (p)
二進制 (p) 單一位元組字元最多可達 8,388,608 個位元組；每個字元 1 個位元組。	水平 (最大)
BOOLEAN	BOOLEAN
DATE	DATE
TIME 時間值介於下午 12 時至晚上 9 時之間	瓦爾查爾 (18)
時間 (f) 時間值介於凌晨 12 時至 9 時 59 分之間。	瓦爾查爾 (N) — 9 + D-在 -1
TIMESTAMP_NTZ	TIMESTAMP
時間戳	TIMESTAMPTZ

雪花到亞馬遜紅移轉換設置

若要編輯雪花到 Amazon Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇雪花，然後選擇雪花-亞馬遜紅移。AWS SCT顯示雪花到亞馬遜紅移轉換的所有可用設定。

雪花到亞馬遜 Redshift 轉換設定AWS SCT包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

雪花到亞馬遜紅移轉換優化設置

若要編輯雪花到 Amazon Redshift 的轉換優化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇雪花，然後選擇雪花-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示雪花到亞馬遜紅移轉換的轉換優化設置。

雪花到亞馬遜 Redshift 的轉換優化設置AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用微軟 SQL 伺服器資料倉儲作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從微軟 SQL Server DW 轉換為亞馬遜紅移或亞馬遜紅移，並組合使AWS Glue用。

作為來源的微軟 SQL 伺服器資料倉儲的權限

使用微軟 SQL Server 資料倉儲作為來源需要下列權限：

- VIEW DEFINITION
- VIEW DATABASE STATE
- SELECT ON SCHEMA :: *<schema_name>*

在上述範例中，將<source_schema>預留位置取代為來源 source_schema 的名稱。

對您要轉換其結構描述的每個資料庫重複授予。

此外，授予以下項目，並在主資料庫上執行授予：

- VIEW SERVER STATE

SQL 伺服器資料倉儲作為來源的限制

目前不支援使用微軟 SQL 伺服器平行資料倉儲 (PDW) 作為來源。

連線至 SQL 伺服器資料倉儲做為來源

使用下列程序來連線到您的 SQL Server 資料倉儲來源資料庫AWS Schema Conversion Tool。

若要連線到 SQL 伺服器資料倉儲來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇微軟 SQL 伺服器，然後選擇下一步。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：

1. 在「機AWS密」中，選擇密碼的名稱。
2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入微軟 SQL Server 來源資料倉儲連線資訊，請使用下列指示：

參數	動作
伺服器名稱	輸入您來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
執行個體名稱	輸入 SQL 伺服器資料倉儲的執行個體名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
Use SSL (使用 SSL)	選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊： <ul style="list-style-type: none"> • 信任伺服器憑證：選取此選項可信任伺服器憑證。 • 信任存放區：您在全域設定中設定的信任存放區。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。

參數	動作
SQL Server 驅動程式路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

SQL 伺服器資料倉儲到亞馬遜紅移轉換設定

若要編輯 SQL Server 資料倉儲到 Amazon Redshift 轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇微軟 SQL 服務器，然後選擇微軟 SQL 服務器-亞馬遜紅移。AWS SCT顯示 SQL 伺服器資料倉儲到亞馬遜紅移轉換的所有可用設定。

SQL 伺服器資料倉儲到亞馬遜紅移轉換設定AWS SCT包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程

式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

SQL 伺服器資料倉儲到亞馬遜紅移轉換優化設定

若要編輯 SQL Server 資料倉儲到 Amazon Redshift 轉換最佳化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇微軟 SQL 服務器，然後選擇微軟 SQL 服務器-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示 SQL 伺服器資料倉儲到亞馬遜紅移轉換的轉換優化設定。

SQL 伺服器資料倉儲到亞馬遜紅移轉換優化設定AWS SCT包含下列選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱[Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選擇選擇選擇策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT 定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以 AWS 移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT 將 ALL 分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT 從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT 將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT 將 ALL 分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用太數據作為源 AWS SCT

您可以使用 AWS SCT 將結構描述、程式碼物件和應用程式程式碼從 Teradata 轉換為亞馬遜紅移或亞馬遜紅移，並組合使用 AWS Glue。

Teradata 作為來源的權限

使用 Teradata 作為來源需要下列權限：

- SELECT ON DBC
- 在系統上選擇
- 在系統上選取
- 選擇開啟 <source_database>
- 開啟建立程序 <source_database>

在上述範例中，將<source_database>預留位置取代為來源資料庫的名稱。

AWS SCT需要「建立程序」權限，才能針對來源資料庫中的所有程序執行「說明程序」。AWS SCT不會使用此權限在來源 Teradata 資料庫中建立任何新物件。

連接到太數據作為源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Teradata 來源資料庫。

若要連接到 Teradata 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇 Teradata，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Teradata 來源資料庫連線資訊，請遵循下列指示：

參數	動作
連線名稱 (Connection name)	輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 Teradata 資料庫的名稱。
使用者名稱和密碼	<p>輸入要連線到來源資料庫伺服器的資料庫認證。</p> <p>AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。</p>
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
加密資料	選擇此選項可加密您與資料庫交換的資料。如果您選擇此選項，則連接埠號碼 443 將用於在AWS SCT與 Teradata 資料庫之間傳輸加密資料。
元數據驅動程序路徑	<p>輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱下載所需資料庫驅動程式。</p> <p>如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱在全域設定中儲存驅動程式路徑。</p>

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。
6. 選擇「連線」以連線到來源資料庫。

使用 LDAP 驗證與太資料來源

若要為在 Windows 中執行 Microsoft Active Directory 的 Teradata 使用者設定輕量型目錄存取通訊協定 (LDAP) 身分驗證，請使用下列步驟。

在下列程序中，作用中的目錄網域是test.local.com。Windows 服務器是DC，並且它使用默認設置進行配置。下列指令碼會建立使用test_ldap中目錄帳戶，而且此帳戶會使用test_ldap密碼。

要設置 LDAP 身份驗證誰運行微軟活動目錄的 Teradata 用戶視窗

1. 在 /opt/teradata/tdat/tdgss/site 目錄中，編輯檔案 TdgssUserConfigFile.xml。將 LDAP 區段變更為下列內容。

```
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. 透過執行組態來套用變更，如下所示。

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. 透過執行以下命令來測試組態。

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

輸出格式應類似以下內容。

```
LdapGroupBaseFQDN: dc=Test, dc=local, dc=com
LdapUserBaseFQDN: dc=Test, dc=local, dc=com
LdapSystemFQDN: dc= test, dc= local, dc=com
LdapServerName: DC.test.local.com
LdapServerPort: 389
LdapServerRealm: test.local.com
LdapClientUseTls: no
LdapClientTlsReqCert: never
LdapClientMechanism: SASL/DIGEST-MD5
LdapServiceBindRequired: no
LdapClientTlsCRLCheck: none
LdapAllowUnsafeServerConnect: yes
UseLdapConfig: no
AuthorizationSupported: no
FQDN: CN=test, CN=Users, DC=Anthem, DC=local, DC=com
```



```
AuthUser: ldap://DC.test.local.com:389/CN=test1,CN=Users,DC=test,DC=local,DC=com
DatabaseName: test
Service: tdsbind
```

4. 使用以下命令重新啟動 TPA。

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. 在 Teradata 資料庫中建立與 Active Directory 相同的使用者，如下所示。

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

如果您變更 LDAP 使用者的使用中目錄中的使用者密碼，請在 LDAP 模式下與 Teradata 連線時指定此新密碼。在預設模式下，您可以使用 LDAP 使用者名稱和任何密碼連線到 Teradata。

在來源 Teradata 資料倉儲中設定統計資料收集

若要轉換來源 Teradata 資料倉儲，請 AWS SCT 使用統計資料來優化轉換後的 Amazon Redshift 資料倉儲。您可以在中收集統計資料 AWS SCT 或上傳統計資料檔案。如需詳細資訊，請參閱 [收集或上傳統計資料](#)。

若要確定 AWS SCT 可以從資料倉儲收集統計資料，請完成下列先決條件工作。

若要從 Teradata 資料倉儲收集統計資料

1. 執行下列查詢，以記住資料倉儲中所有資料表的統計資料。

```
collect summary statistics on table_name;
```

在上述範例中，將 *table_name* 取代為來源資料表的名稱。針對您轉換的每個資料表重複查詢。

2. 執行下列查詢以決定使用者的帳戶字串，該字串用於轉換資料倉儲。

```
select * from dbc.accountinfo where username = 'user_name'
```

3. 使用上一個範例中的帳戶字串，為特定使用者開啟查詢記錄。

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('M$BUSI$$D$H');
```

或者，開啟所有資料庫使用者的查詢記錄。

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

完成收集資料倉儲統計資料之後，請關閉查詢記錄。若要這麼做，您可以使用下列程式碼範例。

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

從來源 Teradata 資料倉儲以離線模式收集統計資料

在 Teradata 資料倉儲中設定統計資料收集之後，您就可以在AWS SCT專案中收集統計資料。或者，您也可以使用基本 Teradata 查詢 (BTEQ) 指令碼，以離線模式收集統計資料。然後，您可以將包含收集到統計信息的文件上傳到您的AWS SCT項目中。如需詳細資訊，請參閱[收集或上傳統計資料](#)。

在離線模式下從 Teradata 資料倉儲收集統計資料

1. 使用下列內容建立off-line_stats.bteq指令碼。

```
.OS IF EXIST column-stats-tera.csv del /F column-stats-tera.csv
.OS IF EXIST table-stats-tera.csv del /F table-stats-tera.csv
.OS IF EXIST column-skew-script-tera.csv del /F column-skew-script-tera.csv
.OS IF EXIST column-skew-stats-tera.csv del /F column-skew-stats-tera.csv
.OS IF EXIST query-stats-tera.csv del /F query-stats-tera.csv
.LOGON your_teradata_server/your_login, your_password
.EXPORT REPORT FILE = table-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
  ''' || OREPLACE(COALESCE(c.DatabaseName, ''), '', '""') || ';' ||
  ''' || OREPLACE(COALESCE(c.TableName, ''), '', '""') || ';' ||
  ''' || TRIM(COALESCE(s.reference_count, '0')) || ';' ||
  ''' || TRIM(COALESCE(CAST(p.RowCount AS BIGINT), '0')) || ';' ||
  ''' || CAST(CAST(w.size_in_mb AS DECIMAL (38,1) FORMAT 'Z9.9') AS VARCHAR(38))
  || ';' ||
  ''' || TRIM(COALESCE(r.stat_fk_dep_count, '0')) || ';' ||
  ''' || CAST(CAST(current_timestamp(0) as timestamp(0) format 'YYYY-MM-
DDBHH:MI:SS') as VARCHAR(19)) || '''
(TITLE
  "database_name";"table_name";"reference_count";"row_count";"size_in_mb";"stat_fk_dep_coun
FROM (select databasename, tablename
      from DBC.tablesv
```

```

        where tablekind IN ('T','O')
        and databasename = 'your_database_name'
        ) c
left join
    (select DatabaseName, TableName, max(RowCount) RowCount
     from dbc.tableStatsv
     group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
    (SELECT r.ChildDB as DatabaseName,
     r.ChildTable as TableName,
     COUNT(DISTINCT r.ParentTable) reference_count
     FROM DBC.All_RI_ChildrenV r
     GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
    (SELECT r.ParentDB as DatabaseName,
     r.ParentTable as TableName,
     COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
     FROM DBC.All_RI_ParentsV r
     GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
    (select databasename, tablename,
     sum(currentperm)/1024/1024 as size_in_mb
     from dbc.TableSizeV
     group by 1,2) w
on w.databasename = c.databasename
and w.tablename = c.tablename
WHERE COALESCE(r.stat_fk_dep_count,0) + COALESCE(CAST(p.RowCount AS BIGINT),0) +
    COALESCE(s.reference_count,0) > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000
    ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) || ';' ||

SELECT
    ''' || OREPLACE(COALESCE(trim(tv.DatabaseName), ''), '', '""') || ';' ||

```

```

        ''' || OREPLACE(COALESCE(trim(tv.TableName), ''), '', '''') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.columnname), ''), '', '''') || ';' ||
        ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) ||

    ';' ||

    ''' || CAST(current_timestamp AS VARCHAR(19)) || ''' (TITLE
    '''database_name";"table_name";"column_name";"cardinality";"current_ts"')
FROM dbc.columnsv tv
LEFT JOIN
(
    SELECT
        c.DatabaseName AS DATABASE_NAME,
        c.TABLENAME AS TABLE_NAME,
        c.ColumnName AS COLUMN_NAME,
        c.UniqueValueCount AS CARD
    FROM dbc.tablestatsv c
    WHERE c.DatabaseName = 'your_database_name'
    AND c.RowCount <> 0
) t2
ON tv.DATABASENAME = t2.DATABASE_NAME
AND tv.TABLENAME = t2.TABLE_NAME
AND tv.COLUMNNAME = t2.COLUMN_NAME
WHERE t2.card > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-script-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
'SELECT CAST(''''' || TRIM(c.DatabaseName) || ''';'''' || TRIM(c.TABLENAME) || ''';''''
|| TRIM(c.COLUMNNAME) || ''';'''' ||
TRIM(CAST(COALESCE(MAX(cnt) * 1.0 / SUM(cnt), 0) AS NUMBER FORMAT '9.9999')) ||
''';'''' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || '''''' AS VARCHAR(512))
AS ''''DATABASE_NAME'''';''''TABLE_NAME'''';''''COLUMN_NAME'''';''''SKEWED'''';''''CURRENT_TS''''
FROM(
SELECT COUNT(*) AS cnt
FROM ''' || c.DATABASENAME || '''.''' || c.TABLENAME ||
''' GROUP BY ''' || c.COLUMNNAME || ''') t' ||
CASE WHEN ROW_NUMBER() OVER(PARTITION BY c.DATABASENAME
ORDER BY c.TABLENAME DESC, c.COLUMNNAME DESC) <> 1
THEN ' UNION ALL '

```

```

ELSE ';' END (TITLE '--SKEWED--')
FROM dbc.columnsv c
INNER JOIN
(SELECT databasename, TABLENAME
FROM dbc.tablesv WHERE tablekind = 'T'
AND databasename = 'your_database_name') t
ON t.databasename = c.databasename
AND t.TABLENAME = c.TABLENAME
INNER JOIN
(SELECT databasename, TABLENAME, columnname FROM dbc.indices GROUP BY 1,2,3
WHERE TRANSLATE_CHK (databasename USING LATIN_TO_UNICODE) + TRANSLATE_CHK
(TABLENAME USING LATIN_TO_UNICODE) + TRANSLATE_CHK (columnname USING
LATIN_TO_UNICODE) = 0
) i
ON i.databasename = c.databasename
AND i.TABLENAME = c.TABLENAME
AND i.columnname = c.columnname
WHERE c.ColumnType NOT IN ('CO','JN','N','++','VA','UT','AN','XM','A1','B0')
ORDER BY c.TABLENAME, c.COLUMNNAME;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

.RUN FILE = column-skew-script-tera.csv

.EXPORT RESET

.EXPORT REPORT FILE = query-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 32000

SELECT
'' || RTRIM(CAST(SqlTextInfo AS VARCHAR(31900)), ';') || ';' ||
TRIM(QueryCount) || ';' ||
TRIM(QueryId) || ';' ||
TRIM(SqlRowNo) || ';' ||
TRIM(QueryParts) || ';' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || ''
(TITLE
'"query_text";"query_count";"query_id";"sql_row_no";"query_parts";"current_ts"')
FROM

```

```

(
  SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
  SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
  ROWS UNBOUNDED PRECEDING) AS topN
  FROM
  (SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
  CASE WHEN
  ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
  SqlRowNo) = 1 AND SqlRowNo = 1
  THEN 1 ELSE 0 END AS QueryFirstRow
  FROM (
  SELECT q.QueryId, q.SqlTextInfo, q.SqlRowNo,
  MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
  COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
  FROM DBC.dbqsqltbl q
  INNER JOIN
  (
  SELECT QueryId
  FROM DBC.DBQLogTbl t
  WHERE TRIM(t.StatementType) IN ('SELECT')
  AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
  AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
  TRIM(t.USERNAME) END) IN ('All users') --user_name list
  AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
  GROUP BY 1
  ) t
  ON q.QueryId = t.QueryId
  INNER JOIN
  (
  SELECT QueryId
  FROM DBC.QryLogObjectsV
  WHERE ObjectDatabaseName = 'your_database_name'
  AND ObjectType = 'Tab'
  AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
  GROUP BY 1
  ) r
  ON r.QueryId = t.QueryId
  WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
  ) t
  ) t
  WHERE SqlTextInfo NOT LIKE '%"%"
  ) q
  WHERE
  QueryParts >=1

```

```
AND topN <= 50
ORDER BY QueryCount DESC, QueryId, SqlRowNo
QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

.EXPORT RESET

.LOGOFF

.QUIT
```

2. 建立執行您在上一個步驟中建立的 BTEQ 指令碼的 `td_run_bteq.bat` 檔案。對此檔案使用下列內容。

```
@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. 建立執行您在上一個步驟中建立的批次檔案的檔案。runme.bat對此檔案使用下列內容。

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

在runme.bat檔案中 *ServerNameUserName*，以適用*DatabaseName*的值取代、和。

然後，運行該runme.bat文件。對您轉換為 Amazon Redshift 的每個資料倉儲重複此步驟。

執行此指令碼之後，您會收到三個檔案，其中包含每個資料庫的統計資料。您可以將這些文件上傳到您的AWS SCT項目中。為此，請從項目的左側面板中選擇數據倉庫，然後打開上下文（右鍵單擊）菜單。選擇上傳統計資料。

太數據到亞馬遜紅移轉換設置

若要編輯 Teradata 到 Amazon Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇太數據，然後選擇太數據-亞馬遜紅移。AWS SCT顯示 Teradata 到亞馬遜紅移轉換的所有可用設定。

Teradata 到亞馬遜紅移轉換設定AWS SCT包含下列選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

- 若要在陳述式的轉換程式碼中使用明確的資料行清SELECT *單，請選取 [使用明確資料行宣告]。
- 若要模擬 Amazon Redshift 叢集中主金鑰和唯一金鑰的行為，請選取「模擬主金鑰和唯一金鑰的行為」。

Amazon Redshift 不會強制執行唯一金鑰和主索引鍵，而且只會將它們用於資訊目的。如果您在程式碼中使用這些條件約束，請確定在轉換後的程式碼中AWS SCT模擬其行為。

- 為了確保目標亞馬遜紅移表中的數據唯一性。若要這麼做，請選取「模擬 SET 表格的行為」。

Teradata 會使用SET語法元素作為預設選項來建立表格。您無法在表格中新增重複的SET列。如果您的源代碼不使用此唯一性約束，請關閉此選項。在這種情況下，轉換後的代碼的工作速度更快。

如果您的原始程式碼使用資料表中的SET選項做為唯一性限制條件，請開啟此選項。在這種情況下，AWS SCT重寫轉換後的代碼中的INSERT...SELECT語句以模擬源數據庫的行為。

太數據到亞馬遜紅移轉換優化設置

若要編輯 Teradata 到 Amazon Redshift 的轉換優化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中選擇太數據，然後選擇太數據-亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示 Teradata 到亞馬遜紅移轉換的轉換優化設定。

Teradata 到亞馬遜紅移轉換優化設置AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

使用 Vertica 作為來源 AWS SCT

您可以使用AWS SCT將結構描述、程式碼物件和應用程式程式碼從 Vertica 轉換為 Amazon Redshift。

Vertica 作為來源的權限

使用 Vertica 作為來源需要下列權限：

- USAGE ON SCHEMA *<schema_name>*
- USAGE ON SCHEMA PUBLIC
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON ALL SEQUENCES IN SCHEMA *<schema_name>*
- EXECUTE ON ALL FUNCTIONS IN SCHEMA *<schema_name>*
- EXECUTE ON PROCEDURE *<schema_name.procedure_name(procedure_signature)>*

在前面的範例中，取代預留位置如下：

- 將#####取代為來源結構描述的名稱。
- 將####取代為來源程序的名稱。針對要轉換的每個程序重複授權。
- 以逗號分隔#####*re_signature*。

連接到 Vertica 作為源

使用以下程序，透過 AWS Schema Conversion Tool 連接到您的 Vertica 來源資料庫。

若要連接到 Vertica 來源資料庫

1. 在中AWS Schema Conversion Tool，選擇「新增來源」。
2. 選擇「垂直」，然後選擇「下一步」。

這時系統顯示「添加源」對話框。

3. 在「連線名稱」中，輸入資料庫的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 使用資料庫認證來源AWS Secrets Manager或手動輸入：

- 若要使用來自秘密管理員的資料庫認證，請遵循下列指示：
 1. 在「機AWS密」中，選擇密碼的名稱。
 2. 選擇填入來自 Secret Manager 的資料庫連線對話方塊中自動填入所有值。

如需使用來自秘密管理員之資料庫認證的資訊，請參閱[使用 AWS Secrets Manager](#)。

- 若要手動輸入 Vertica 來源資料庫連線資訊，請遵循下列指示：

參數	動作
伺服器名稱	輸入來源資料庫伺服器的網域名稱系統 (DNS) 名稱或 IP 位址。
伺服器連接埠	輸入用於連接到來源資料庫伺服器的連接埠。
Database (資料庫)	輸入 Vertica 資料庫的名稱。
使用者名稱和密碼	輸入要連線到來源資料庫伺服器的資料庫認證。 AWS SCT只有當您選擇連線至專案中的資料庫時，才會使用此密碼連線至來源資料庫。為了防止公開來源資料庫的密碼，預設AWS SCT不會儲存密碼。如果您關閉 AWS SCT 專

參數	動作
	案再重新開啟，系統會在需要時提示您輸入密碼以便連接到您的來源資料庫。
Use SSL (使用 SSL)	選擇此選項可使用安全通訊端層 (SSL) 連線到資料庫。在 SSL 索引標籤上提供下列適用的其他資訊： <ul style="list-style-type: none"> 驗證伺服器憑證：選擇此選項可使用信任存放區驗證伺服器憑證。 信任存放區：您在全域設定中設定的信任存放區。 金鑰存放區：您在全域設定中設定的金鑰存放區。
儲存密碼	AWS SCT 會建立安全的保存庫來存放 SSL 憑證和資料庫密碼。開啟此選項後，您可以儲存資料庫密碼並快速連線到資料庫，而無需輸入密碼。
垂直的驅動程序路徑	輸入用於連接至來源資料庫的驅動程式路徑。如需詳細資訊，請參閱 下載所需資料庫驅動程式 。 如果您在全域專案設定中存放驅動程式路徑，驅動程式路徑不會顯示在連線對話方塊中。如需詳細資訊，請參閱 在全域設定中儲存驅動程式路徑 。

5. 選擇「測試連線」以確認是否AWS SCT可以連線到來源資料庫。

6. 選擇「連線」以連線到來源資料庫。

Vertica 到亞馬遜紅移轉換設置

若要編輯 Vertica 到 Amazon Redshift 的轉換設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇 Vertica，然後選擇 Vertica — 亞馬遜紅移。AWS SCT顯示 Vertica 到亞馬遜紅移轉換的所有可用設定。

Vertica 到亞馬遜紅移轉換設置AWS SCT包括以下選項：

- 用於限制轉換後代碼中行動項目的註釋數量。

針對所選嚴重性及更高之行動項目的已轉換程式碼中新增註解，請選擇行動項目的嚴重性。AWS SCT在已轉換的程式碼中加入所選嚴重性及更高嚴重性之行動項目的註解。

例如，若要將轉換後的程式碼中的註解數目降到最低，請選擇「僅限錯誤」。若要在轉換的程式碼中包含所有行動項目的註解，請選擇 [所有郵件]。

- 設定AWS SCT可套用至目標 Amazon Redshift 叢集的表格數目上限。

對於目標 Amazon Redshift 叢集的表格數目上限，請選擇AWS SCT可套用至 Amazon Redshift 叢集的表格數目。

Amazon Redshift 的配額會限制不同叢集節點類型的使用表格。如果選擇自動，請根據節點類型AWS SCT決定要套用至目標 Amazon Redshift 叢集的表格數目。(選擇性) 手動選擇值。如需詳細資訊，請參閱[亞馬遜紅移管理指南中的配額和限制](#)。

AWS SCT轉換所有來源表格，即使這個資料表超過 Amazon Redshift 叢集所能存放的空間也一樣。AWS SCT將轉換後的代碼存儲在項目中，並且不將其應用於目標數據庫。如果您在套用轉換後的程式碼時達到表格的 Amazon Redshift 叢集配額，則AWS SCT會顯示警告訊息。此外，請將表格AWS SCT套用至目標 Amazon Redshift 叢集，直到表格數量達到上限為止。

- 將來源表格的分割區遷移到亞馬遜 Redshift 中的單獨表格。若要這麼做，請選取 [使用 UNION ALL] 檢視，然後輸入AWS SCT可為單一來源表格建立的目標表格數目上限。

亞馬遜紅移不支持表分區。若要模擬此行為並使查詢執行更快，AWS SCT可以將來源資料表的每個分區遷移到 Amazon Redshift 中的個別資料表。然後，AWS SCT創建一個包含所有這些表中數據的視圖。

AWS SCT會自動決定來源資料表中的分割區數目。視來源表格分割的類型而定，此數字可能會超過您可套用至 Amazon Redshift 叢集之表格的配額。若要避免達到此配額，請輸入AWS SCT可為單一來源表格分割區建立的目標表格數目上限。預設選項為 368 個資料表，代表一年 366 天的分割區，以及兩個和分UNKNOWN割區的NO RANGE資料表。

- 將壓縮應用於亞馬遜紅移表列。若要這樣做，請選取 [使用壓縮編碼]。

AWS SCT使用預設的 Amazon Redshift 演算法，自動將壓縮編碼指派給資料行。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的[壓縮編碼](#)。

根據預設，Amazon Redshift 不會將壓縮套用至定義為排序金鑰和分配金鑰的資料行。您可以變更此行為，並將壓縮套用至這些資料行。若要這麼做，請選取 [使用 KEY 欄的壓縮編碼]。只有當您選取 [使用壓縮編碼] 選項時，才能選取此選項。

Vertica 到亞馬遜紅移轉換優化設置

若要編輯 Vertica 到 Amazon Redshift 的轉換優化設定，請在中選擇 [設定]AWS SCT，然後選擇 [轉換設定]。從上面的列表中，選擇 Vertica，然後選擇 Vertica — 亞馬遜紅移。在左窗格中，選擇 [最佳化策略]。AWS SCT顯示 Vertica 到亞馬遜紅移轉換的轉換優化設置。

Vertica 到亞馬遜紅移轉換優化設置AWS SCT包括以下選項：

- 要使用自動表優化工作。若要這麼做，請選取 [使用亞馬遜紅移自動表格調整]。

自動表格最佳化是 Amazon Redshift 中的一項自我調整程序，可自動優化資料表的設計。如需詳細資訊，請參閱 [Amazon Redshift 資料庫開發人員指南中的使用自動表格最佳化](#)。

若只要依賴自動表格最佳化，請針對初始索引鍵選取策略選取選擇選擇選取策略

- 若要使用您的策略選擇排序和分配鍵。

您可以使用 Amazon Redshift 中繼資料、統計資訊或這兩個選項來選擇排序和分發金鑰。對於「最佳化策略」標籤上的「初始金鑰選取策略」，請選擇下列其中一個選項：

- 使用中繼資料，忽略統計資訊
- 忽略中繼資料，使用統計資訊
- 使用中繼資料和統計資訊

根據您選擇的選項，您可以選取最佳化策略。然後，針對每個策略輸入值 (0—100)。這些值定義每個策略的權重。使用這些加權值，AWS SCT定義每個規則如何影響分佈和排序索引鍵的選擇。預設值以AWS移轉最佳作法為基礎。

您可以為尋找小型資料表策略定義小型資料表的大小。對於「最小表格列計數」和「表格列數上限」，請輸入表格中的最小和最大列數，將其定義為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

- 若要設定策略詳細資訊。

除了定義每個最佳化策略的權重之外，您還可以設定最佳化設定。若要這樣做，請選擇 [轉換優化]。

- 在「排序索引鍵欄限制」中，輸入排序索引鍵中的欄數上限。
- 在「傾斜臨界值」中，為欄輸入傾斜值的百分比 (0—100)。AWS SCT從分配索引鍵的候選清單中排除偏斜值大於臨界值的欄。AWS SCT將欄的傾斜值定義為最常見值出現次數與記錄總數的百分比。
- 對於查詢歷史記錄表格中的前 N 個查詢，請輸入要分析的最常用查詢數目 (1—100)。
- 在選取統計資料使用者中，選擇要分析查詢統計資料的資料庫使用者。

此外，在「最佳化策略」標籤上，您可以定義「尋找小表格」策略的小型資料表大小。對於「最小表格資料列計數」和「表格列數上限」，請輸入表格中的最小與最大列數，將其視為小型表格。AWS SCT將ALL分佈型式套用至小型表格。在這種情況下，整個表的副本會分發到每個節點。

在中建立對應規則 AWS SCT

您可以在單一AWS SCT專案中新增多個來源和目標資料庫。當您將多個資料庫遷移到不同的目標平台時，這樣可簡化專案的管理。

建立新專案並新增來源和目標資料庫之後，請建立對應規則。AWS SCT至少需要一個對應規則，才能建立移轉評估報表並轉換資料庫結構描述。

對應規則說明包含來源資料庫綱要或來源資料庫以及目標資料庫平台的來源-目標目標配對。您可以在單一AWS SCT專案中建立多個對映規則。使用對應規則將每個來源資料庫結構描述轉換為正確的目標資料庫平台。

若要在轉換後的程式碼中變更結構描述的名稱，請設定移轉規則。例如，透過移轉規則，您可以重新命名結構描述、新增物件名稱前置詞、變更資料行定序或變更資料類型。若要將這些變更套用至轉換後的程式碼，請確定在轉換來源結構描述之前先建立遷移規則。如需詳細資訊，請參閱 [建立移轉規則](#)。

您只能為支援的資料庫轉換配對建立對映規則。如需支援的轉換配對清單，請參閱 [AWS SCT 的來源](#)。

如果您開啟以 1.0.655 版或更新AWS SCT版本儲存的專案，AWS SCT會自動為所有來源資料庫結構描述建立對應規則至目標資料庫平台。若要新增其他目標資料庫平台，請刪除現有的對應規則，然後建立新的對應規則。

主題

- [新增對應規則](#)
- [管理對應規則](#)
- [使用虛擬目標](#)
- [在單一AWS SCT專案中使用多部伺服器的限制](#)

新增對應規則

您可以在單一專案中建立多個對映規則。AWS SCT將對映規則儲存為專案的一部分。在專案開啟的情況下，使用下列程序來新增對應規則。

若要建立對應規則

1. 在 [檢視] 功能表上，選擇 [對應檢視]。

2. 在左側面板中，選擇要新增至對應規則的綱要或資料庫。
3. 在右側面板中，為選取的來源結構描述或資料庫選擇目標資料庫平台。

您可以選擇虛擬資料庫平台作為目標。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 選擇 [建立對應]。

AWS SCT將此新對應規則新增至伺服器對應清單。

新增所有轉換配對的對映規則。若要建立評估報告或轉換資料庫架構，請選擇 [檢視] 功能表上的 [主要檢視]。

AWS SCT以粗體反白屬於對應規則一部分的所有綱要物件。

管理對應規則

您可以篩選或刪除現有的對應規則，並在 AWS Schema Conversion Tool (AWS SCT) 專案中新增對應規則。

當您為整個來源資料庫建立對應規則時，AWS SCT會為每個來源資料庫綱要建立一個對應規則。對於涉及數十個模式甚至數據庫的項目，可能很難理解，哪個目標用於某個模式。若要快速尋找結構描述的對應規則，請在中使用下列一或多個篩選選項AWS SCT。

篩選對映規則

1. 在 [檢視] 功能表上，選擇 [對應檢視]。
2. 對於來源伺服器，請選擇來源資料庫。

篩選預設值為「全部」，這表示AWS SCT會顯示所有來源資料庫的對應規則。

3. 在來源綱要中，輸入來源結構描述名稱。使用百分比 (%) 作為萬用字元來取代結構描述名稱中任意數量的符號。

篩選器預設值為% 萬用字元，這表示AWS SCT會顯示所有來源資料庫結構描述名稱的對應規則。

4. 對於具有移轉規則，選擇是以顯示為其建立資料移轉規則的對映規則。選擇否以顯示沒有資料遷移規則的對應規則。如需詳細資訊，請參閱[在中建立資料移轉規則 AWS SCT](#)。

篩選器預設值為「全部」，這表示AWS SCT會顯示所有對應規則。

5. 對於「目標伺服器」，請選擇目標資料庫。

篩選預設值為「全部」，這表示AWS SCT會顯示所有目標資料庫的對應規則。

在專案開啟的情況下，使用下列程序刪除對應規則。如需新增對應規則的詳細資訊，請參閱[新增對應規則](#)。

若要刪除對應規則

1. 在 [檢視] 功能表上，選擇 [對應檢視]。
2. 對於伺服器對映，請選擇要刪除的對應規則。
3. 選擇刪除選取的對映。

AWS SCT刪除選取的對映規則。

使用虛擬目標

您可以看到如何AWS SCT將源數據庫模式轉換為任何支持的目標數據庫平台。若要這麼做，您不需要連線至現有的目標資料庫。相反地，您可以在建立對應規則時，在右側面板中選擇虛擬目標資料庫平台。如需詳細資訊，請參閱[新增對應規則](#)。請確定您展開右側面板中的 [伺服器]、[NoSQL 叢集] 和 [ETL] 節點，以查看虛擬目標資料庫平台的清單。

AWS SCT支援下列虛擬目標資料庫平台：

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon DynamoDB
- Amazon Redshift
- 亞馬遜紅移和 AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish for Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle

- PostgreSQL

如果您使用適用於 Aurora PostgreSQL 的 Babel 魚做為目標資料庫平台，則只能建立資料庫移轉評估報告。如需詳細資訊，請參閱[the section called “遷移評定報告”](#)。

如果您使用虛擬目標資料庫平台，則可以將轉換後的程式碼儲存到檔案中。如需詳細資訊，請參閱[the section called “儲存轉換的結構描述”](#)。

在單一AWS SCT專案中使用多部伺服器的限制

在單一AWS SCT專案中使用多部伺服器轉換結構描述時，會套用下列限制：

- 您只能將相同的伺服器新增至專案一次。
- 您無法將伺服器結構描述對應至特定的目標結構描述，只能對應至目標伺服器。AWS SCT在轉換期間建立目標綱要。
- 您無法將較低層級的來源物件對應至目標伺服器。
- 您只能將一個來源結構描述對應至專案中的一個目標伺服器。
- 請務必將來源對映至目標伺服器，以建立評估報告、轉換結構描述或擷取資料。

建立轉換報告

當您在規劃資料庫轉換時，建立一些報告來協助您了解有關的內容很有用。您可以使用 AWS Schema Conversion Tool 建立報告。

您可以使用 AWS SCT 來建立資料庫遷移評估報告。您將在此報告中取得結構描述轉換作業的摘要，以及無法自動轉換至目標資料庫之項目的詳細資訊。您可以將此報告用於評估可以完成多少專案，方法是使用 AWS SCT，以及完成轉換所需的其他內容。若要建立評估報告，請使用建立報告從數據庫的內容選單 (按一下滑鼠右鍵) AWS SCT。

主題

- [建立移轉評估報告 AWS SCT](#)

建立移轉評估報告 AWS SCT

其中一個重要的部分 AWS Schema Conversion Tool 是它產生的評估報告，以估計結構描述轉換的複雜性。此資料庫移轉評估報告會彙總所有結構描述轉換工作，並詳細說明無法轉換為目標資料庫執行個體之資料庫引擎之結構描述的行動項目。您可以在應用程式中檢視報告，或將報告匯出為逗號分隔值 (CSV) 或 PDF 檔案。

如果您在單一專案中新增多個來源和目標資料庫，則會將所有轉換配對的報表 AWS SCT 彙總為一個資料庫移轉評估報告。

您可以使用虛擬目標資料庫平台來產生評估報告，並瞭解移轉至所選資料庫平台的複雜性。在此情況下，您不需要連線至目標資料庫平台。例如，您可以使用適用於 Aurora PostgreSQL 的 Babel 魚做為虛擬目標資料庫平台，以建立資料庫移轉評估報告。如需有關虛擬目標資料庫平台的詳細資訊，請參閱 [the section called “虛擬目標”](#)。

遷移評估報告包括下列項目：

- 執行摘要
- 授權評估
- 雲端支援，表示來源資料庫中的任何功能在目標上無法使用。
- 建議，包括伺服器物件的轉換、備份建議和連結的伺服器變更

本報告亦包含將相同的程式碼寫入無法自動轉換的目標資料庫執行個體時，所需要進行的作業量估算。

如果您使用AWS SCT將現有結構描述遷移到 Amazon RDS 資料庫執行個體，則可以使用報告來協助您分析移至AWS雲端的需求並變更授權類型。

主題

- [建立資料庫遷移評定報告](#)
- [檢視評定報告](#)
- [儲存評定報告](#)
- [設定評定報告](#)
- [建立資料庫移轉的多伺服器評估報告](#)

建立資料庫遷移評定報告

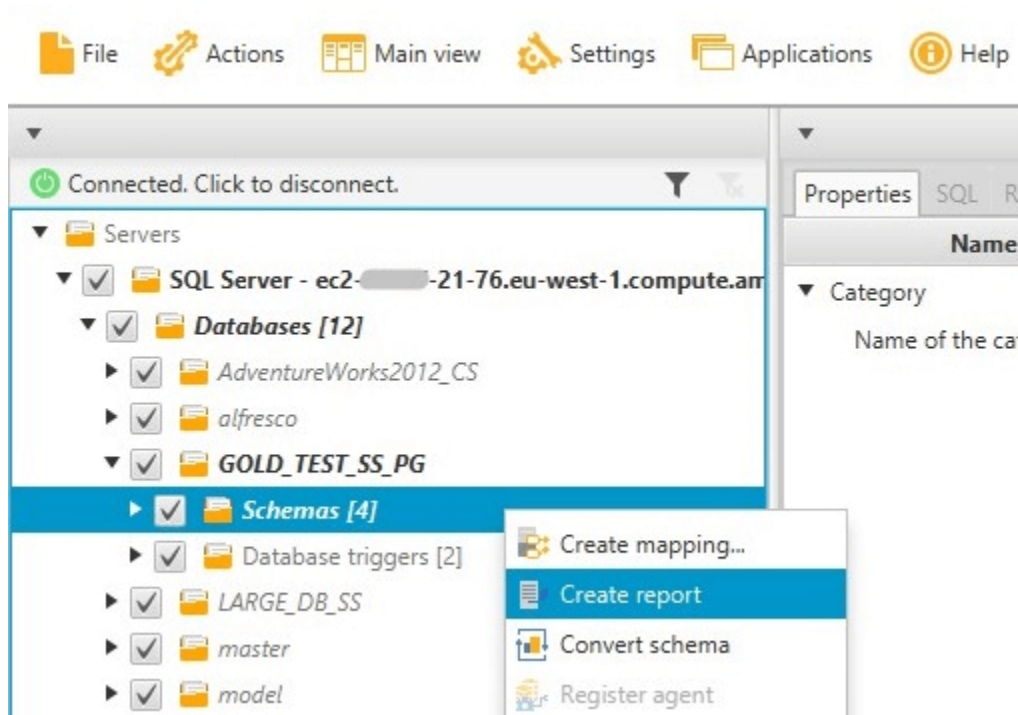
請使用下列程序，建立資料庫遷移評估報告。

若要建立資料庫遷移評估報告

1. 請確定您已為來源資料庫綱要建立對應規則，以建立評估報告。如需詳細資訊，請參閱[新增對應規則](#)。
2. 在「檢視」功能表上，選擇「主視圖」。
3. 在顯示來源資料庫結構描述的左側面板中，選擇要建立評估報告的綱要物件。若要在報表中包含多個資料庫結構描述，請選擇父節點，例如 Schema。

請確定您已選取所有綱要物件的核取方塊，以建立評估報告。

4. 開啟物件的內容 (按一下滑鼠右鍵) 選單，然後選擇建立報告。



檢視評定報告

當您建立評估報告後，評估報告檢視會開啟，顯示以下索引標籤：

- 摘要
- Action Items (動作項目)

Summary (摘要) 索引標籤會顯示自動轉換或未轉換的項目。

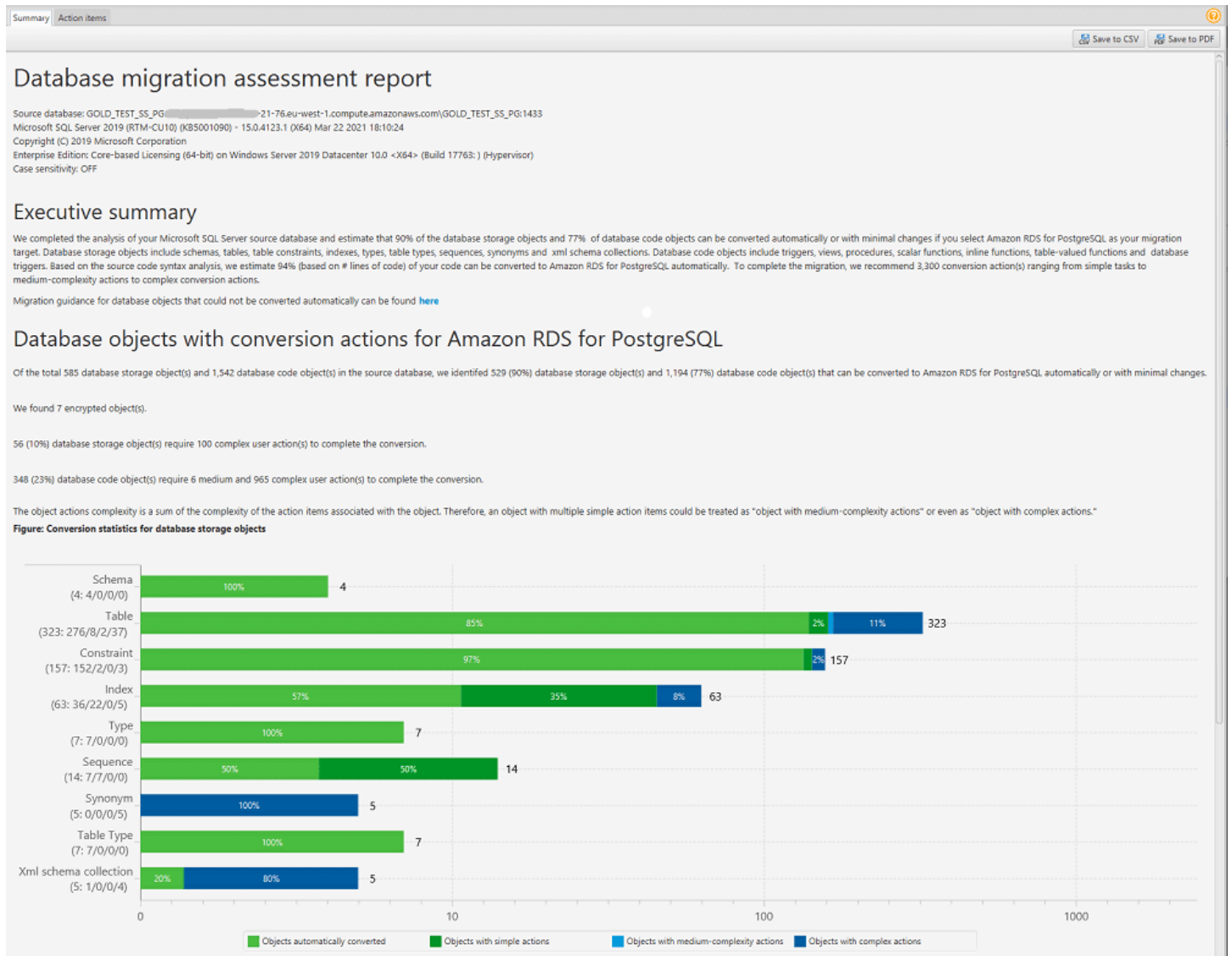
Action Items (動作項目) 索引標籤會顯示無法自動轉換的項目，以及相關建議動作。

主題

- [評估報告摘要](#)
- [評估報告行動項目](#)
- [評估報告警告訊息](#)

評估報告摘要

Summary (摘要) 索引標籤會顯示資料庫遷移評估報告的摘要資訊。它會顯示已自動轉換的項目，以及未自動轉換的項目。



針對無法自動轉換至目標資料庫引擎的結構描述項目，此摘要包括在您的目標資料庫執行個體中建立與來源相同的結構描述項目時，所需要進行的作業量估算。

此報告將轉換上述結構描述項目的估算時間分類如下：

- 簡單 — 可在兩小時內完成的動作。
- 中 — 較複雜且可在兩到六小時內完成的動作。
- 重要 — 非常複雜且需要六個小時以上才能完成的動作。

授權評估和雲端 Support 一節包含有關將現有現場部署資料庫結構描述移至執行相同引擎之 Amazon RDS 資料庫執行個體的相關資訊。例如，如果您要變更授權類型，此報告區段會告訴您應從目前的資料庫中移除哪些功能。

License evaluation

Our analysis shows that current schema uses the following Enterprise Edition features unavailable in Standard Edition.

Feature	Description
Database In-Memory	Oracle Database In-Memory optimizes both analytics and mixed workload OLTP, delivering outstanding performance for transactions while simultaneously supporting real-time analytics, business intelligence, and reports.
Materialized View Query Rewrite	Oracle Database employs an extremely powerful process called query rewrite to quickly answer the query using materialized views.
Partitioning	Partitioning is powerful functionality that allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity.
Oracle Advanced Security/TDE	Oracle Advanced Security provides two important preventive controls to protect sensitive data at the source: encryption and redaction. Together, these two controls form the foundation of Oracle's defense-in-depth, multi-layered database security solution.

If you choose Standard Edition as your migration target, remove dependencies on these features.

Cloud support

Our analysis shows that your current schema uses the following features that require configuration steps in Amazon RDS for Oracle.

Feature	Description
Locator	Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. Please read prerequisites and configuration steps in the next article: Oracle Locator .
Spatial	Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial data in an Oracle database. Please read prerequisites and configuration steps in the next article: Oracle Spatial .
Oracle XML DB	Oracle XML DB provides full support for all of the key XML standards, including XML Namespaces, DOM, XQuery, SQL/XML and XSLT. Amazon RDS for Oracle supports XML DB feature without the XML DB Protocol Server. Please read prerequisites and configuration steps in the next article: Oracle XML DB option .

If choose Amazon RDS for Oracle as your migration target, please follow the abovementioned steps to continue to use these features on the target database after migration completes.

評估報告行動項目

評估報告檢視也包含 Action Items (動作項目) 索引標籤。此索引標籤包含無法自動轉換為目標 Amazon RDS 資料庫執行個體資料庫引擎的項目清單。如果您在清單中選取動作項目，AWS SCT 會在結構描述中反白該動作項目套用的項目。

此報告也包含如何手動轉換結構描述項目的建議。例如，在評估執行之後，資料庫/綱要的詳細報告會顯示設計和實作轉換行動項目建議所需的工作量。如需決定如何處理手動轉換的詳細資訊，請參閱[處理手動轉換 AWS SCT](#)。

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, tables, and schemas. The main pane shows a list of issues with details for each, such as recommended actions and occurrence counts. Below the issues, a SQL procedure definition is shown for the target Amazon RDS MySQL category.

Issues:

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

SQL Procedure Definition:

```

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

```

Target Amazon RDS for MySQL category: Schemas

Name	Value
Category	Schemas
Name of the category	Schemas

評估報告警告訊息

若要評估轉換為其他資料庫引擎的複雜性，AWS SCT需要存取來源資料庫中的物件。如果 SCT 因掃描期間遇到問題而無法執行評估，則會發出警告訊息，指出整體轉換百分比已降低。

Warning!

We found that your source database may be configured not in correct way or you have not enough privileges for reading all necessary metadata. Please check your configuration and run report again. For more details please review [help documentation](#).

List of Action Items to review:

- Issue 9997** Unable to resolve objects (number of occurrences: 3)
Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.

掃描期間AWS SCT可能會遇到問題的原因如下：

- 連接到數據庫的用戶帳戶無法訪問所有需要的對象。
- 結構描述中引用的對象不再存在於數據庫中。
- SCT 正在嘗試評估已加密的物件。

如需 SCT 所需資料庫之安全性權限和權限的詳細資訊，請參閱[AWS SCT 的來源](#)本指南中的適當來源資料庫一節。

儲存評定報告

[建立資料庫移轉評估報告](#)後，您可以將資料庫移轉評估報告的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案。

若要將資料庫移轉評估報告另存為 PDF 檔案

1. 在頂端功能表中，選擇 [檢視]，然後選擇 [評估報告檢視]。
2. 選擇 Summary (摘要) 索引標籤。
3. 選擇右上角的「儲存為 PDF」。

將資料庫移轉評估報告儲存為 CSV 檔案

1. 在頂端功能表中，選擇 [檢視]，然後選擇 [評估報告檢視]。
2. 選擇 Summary (摘要) 索引標籤。
3. 選擇右上角的「儲存為 CSV」。

PDF 檔案同時包含摘要和動作項目資訊，如以下範例所示。

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

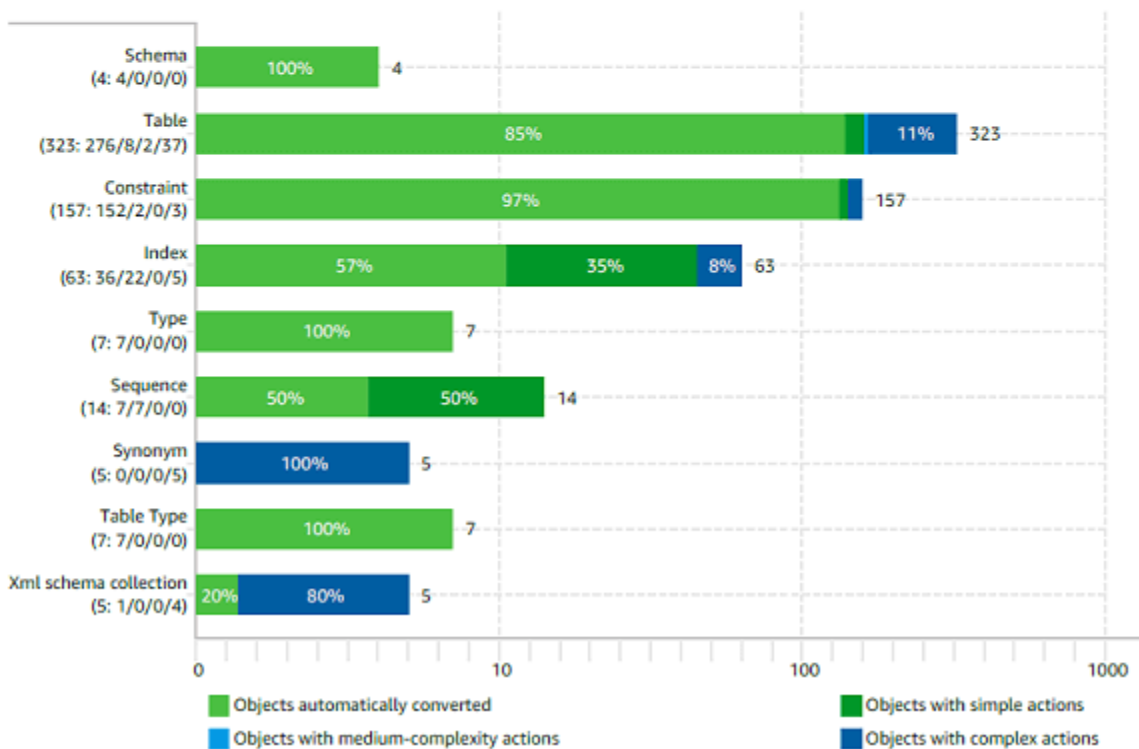
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



當您選擇「儲存為 CSV」選項時，AWS SCT會建立三個 CSV 檔案。

第一個 CSV 檔案包含關於行動項目的下列資訊：

- 類別
- 出現位置 — 項目的檔案名稱、行號和位置
- 動作項目編號
- 主旨
- 群組

- 描述
- 文件參考
- 建議的動作
- 預估複雜性

第二個 CSV 檔案在其名稱中包含 Action_Items_Summary 字尾，並包含有關所有行動項目出現次數的資訊。

在下列範例中，「學習曲線工時」欄中的值指出設計轉換每個行動項目的方法所需的工作量。「轉換行動項目出現位置的努力」欄中的值表示按照設計方法轉換每個行動項目所需的工作量。用來表示所需工作量的值是以加權比例為基礎，範圍從低 (最小) 到高 (最多)。

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

第三個 CSV 檔案包含 Summary 在其名稱中，並包含下列摘要：

- 類別
- 物件數
- 自動轉換
- 具有簡單動作的物件
- 具有中複雜度動作的物件
- 具有複雜動作的物件
- 程式碼總行

設定評定報告

您可以設定 AWS SCT 包含在評估報告中的詳細資料量。

設定資料庫移轉評估報告

1. 在 [設定] 功能表上，選擇 [全域設定]，然後選擇 [評估報告]。

2. 對於行動項目發生次數，請選擇「僅前五個問題」，以限制評估報告中單一類型的行動項目數。選擇 [所有問題]，將每種類型的所有行動項目納入評估報告中。
3. 對於 SQL 命令檔分析的檔案，請選擇列出不超過 *X* 檔案，將評估報告中 SQL 命令檔檔案的數目限制為 *X*。輸入檔案數目。選擇「列出所有分析的檔案」，將所有 SQL 命令檔檔案納入評估報告中。
4. 選取儲存後開啟報表，以在您儲存資料庫移轉評估報告的本機複本後自動開啟檔案。如需詳細資訊，請參閱

建立資料庫移轉評估報告後，您可以將資料庫移轉評估報告的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案。

若要將資料庫移轉評估報告另存為 PDF 檔案

-
1. 在頂端功能表中，選擇 [檢視]，然後選擇 [評估報告檢視]。
 2. 選擇 Summary (摘要) 索引標籤。
 3. 選擇右上角的「儲存為 PDF」。

將資料庫移轉評估報告儲存為 CSV 檔案

-
1. 在頂端功能表中，選擇 [檢視]，然後選擇 [評估報告檢視]。
 2. 選擇 Summary (摘要) 索引標籤。
 3. 選擇右上角的「儲存為 CSV」。

PDF 檔案同時包含摘要和動作項目資訊，如以下範例所示。

Database objects with conversion actions for Amazon RDS for PostgreSQL

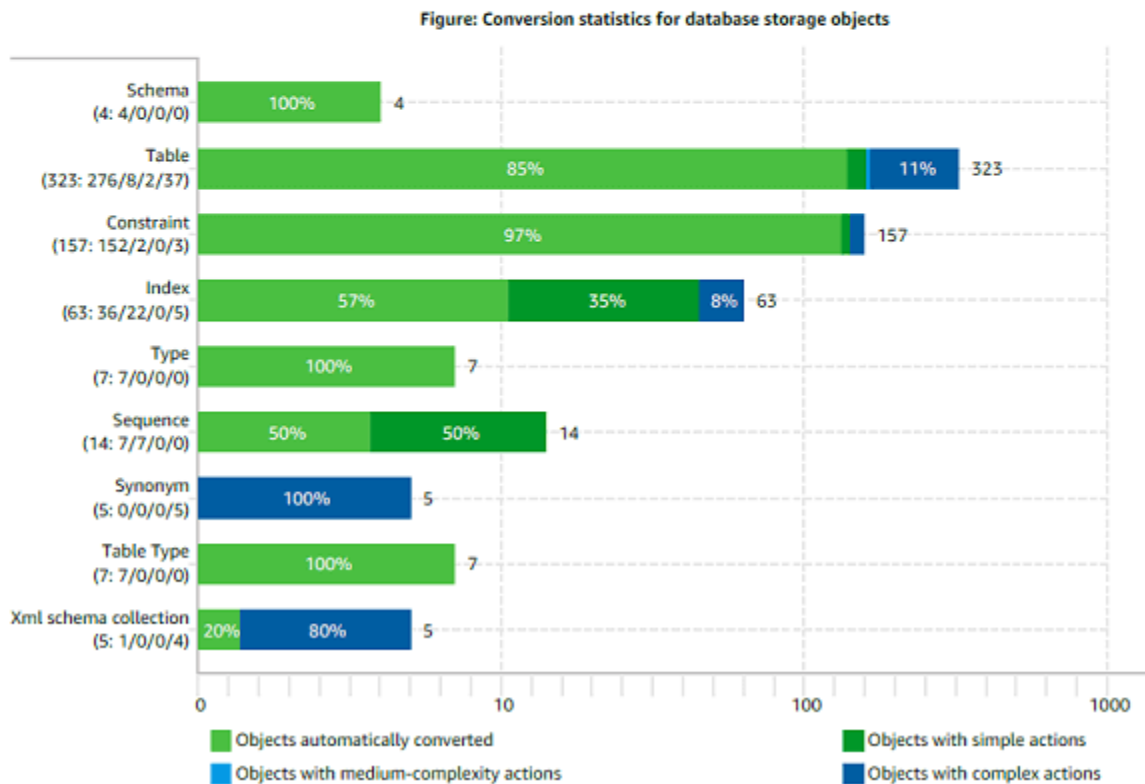
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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



當您選擇「儲存為 CSV」選項時，AWS SCT會建立三個 CSV 檔案。

第一個 CSV 檔案包含關於行動項目的下列資訊：

- 類別
- 出現位置 — 項目的檔案名稱、行號和位置
- 動作項目編號
- 主旨
- 群組

- 描述

-
- 文件參考

-
- 建議的動作

-
- 預估複雜性

第二個 CSV 檔案在其名稱中包含Action_Items_Summary字尾，並包含有關所有行動項目出現次數的資訊。

在下列範例中，「學習曲線工時」欄中的值指出設計轉換每個行動項目的方法所需的工作量。

「轉換行動項目出現位置的努力」欄中的值表示按照設計方法轉換每個行動項目所需的工作量。用來表示所需工作量的值是以加權比例為基礎，範圍從低 (最小) 到高 (最多)。

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

第三個 CSV 檔案包含Summary在其名稱中，並包含下列摘要：

- 類別

-
- 物件數

-
- 自動轉換

-
- 具有簡單動作的物件

-
- 具有中複雜度動作的物件

-
- 具有複雜動作的物件

-
- 程式碼總行
-

。

建立資料庫移轉的多伺服器評估報告

若要決定整體環境的最佳目標方向，請建立多伺服器評估報告。

多伺服器評估報告會根據您為每個要評估的結構描述定義提供的輸入，評估多部伺服器。您的結構定義包含資料庫伺服器連線參數和每個結構描述的全名。評估每個結構描述後，AWS SCT會針對跨多部伺服器的資料庫移轉產生摘要、彙總的評估報告。此報告顯示每個可能移轉目標的預估複雜度。

您可以使用AWS SCT為下列來源和目標資料庫建立多伺服器評估報告。

來源資料庫	目標資料庫
Amazon Redshift	Amazon Redshift
天青 SQL 數據庫	Aurora PostgreSQL、Aurora PostgreSQL
Azure 突觸 Analytics	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
適用於 z/OS 的 IBM Db2	Amazon Aurora MySQL 相容版本、Aurora MySQL、Aurora PostgreSQL、Amazon Aurora PostgreSQL、Aurora PostgreSQL
IBM Db2 LUW	Aurora PostgreSQL、MariaDB、MariaDB、MySQL、PostgreSQL
Microsoft SQL Server	Aurora PostgreSQL、Amazon Redshift、Aurora PostgreSQL、Aurora、Aurora、Aurora、MariaDB ora、PostgreSQL、MySQL、PostgreSQL、PostgreSQL
MySQL	Aurora PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora PostgreSQL、Amazon Redshift、MariaDB、MariaDB、MySQL、Oracle、PostgreSQL
PostgreSQL	Aurora PostgreSQL、Aurora PostgreSQL

來源資料庫	目標資料庫
SAP ASE	Aurora PostgreSQL、MariaDB、MariaDB、MySQL、PostgreSQL
Snowflake	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

執行多伺服器評估

使用下列程序來使用執行多伺服器評估AWS SCT。您不必在中建立新專案，即可執AWS SCT行多伺服器評定。開始之前，請確定已準備好含有資料庫連線參數的逗號分隔值 (CSV) 檔案。另外，請確保您已安裝所有必需的數據庫驅動程序，並在設置中設置驅動程序的位AWS SCT置。如需詳細資訊，請參閱[下載所需資料庫驅動程式](#)。

執行多伺服器評估並建立彙總摘要報告

1. 在中AWS SCT，選擇檔案，新增多伺服器評估。[新增多伺服器評估] 對話方塊隨即開啟。

New multiserver assessment [Close]

Enter the project name, location to store reports and project files, and location of your connections file. [Help]

Project name:

Location: [Browse]

Connections file: [Browse]

[Download a connections file example](#)

Create AWS SCT projects for each source database

Add mapping rules to these projects and save conversion statistics for offline use

[Run] [Cancel]

2. 選擇 [下載連線檔案範例] 以下載含有資料庫連線參數的 CSV 檔案的空白範本。

3. 輸入「專案名稱」、「位置」(用於儲存報告)和「連接檔案」(CSV 檔案)的值。
4. 選擇 [為每個來源資料庫建立AWS SCT專案]，以在產生評估報告後自動建立移轉專案。
5. 開啟 [為每個來源資料庫建立AWS SCT專案] 後，您可以選擇 [將對映規則新增至這些專案]，並儲存轉換統計資料以供離線使用。在此情況下，AWS SCT會將對應規則新增至每個專案，並將來源資料庫中繼資料儲存在專案中。如需詳細資訊，請參閱[以離線模式執行AWS SCT](#)。
6. 選擇 Run (執行)。

會出現一個進度列，指出資料庫評估的速度。目標引擎的數目可能會影響評估執行階段。

7. 如果顯示下列訊息，請選擇是：完整分析所有資料庫伺服器可能需要一些時間。您要繼續嗎？

當多伺服器評估報告完成時，會出現一個畫面，指出這樣做。

8. 選擇「開啟報表」以檢視彙總的摘要評估報告。

依預設，AWS SCT會為所有來源資料庫產生彙總報表，並針對來源資料庫中的每個結構描述名稱產生詳細的評估報告。如需詳細資訊，請參閱[尋找和檢視報表](#)。

打開「為每個來源資料庫建AWS SCT立AWS SCT專案」選項後，為每個來源資料庫建立一個空專案。AWS SCT也會建立先前所述的評估報告。分析這些評估報告並為每個來源資料庫選擇移轉目的地之後，請將目標資料庫新增至這些空白專案。

開啟「將對映規則新增至這些專案並儲存轉換統計資料以供離線使用」選項後，為每個來源資料庫AWS SCT建立專案。這些專案包含下列資訊：

- 您的來源資料庫和虛擬目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。
- 此來源-目標配對的對應規則。如需詳細資訊，請參閱[建立對應規則](#)。
- 此來源-目標配對的資料庫移轉評估報表。
- 來源結構描述資料，可讓您以離線模式使用此AWS SCT專案。如需詳細資訊，請參閱[以離線模式執行AWS SCT](#)。

準備一個輸入的 CSV 文件

若要提供連線參數做為多伺服器評定報告的輸入，請使用 CSV 檔案，如以下範例所示。

```
Name,Description,Secret Manager Key,Server IP,Port,Service Name,Database name,BigQuery path,Source Engine,Schema Names,Use Windows Authentication,Login,Password,Use SSL,Trust store,Key store,SSL authentication,Target Engines
Sales,,,192.0.2.0,1521,pdb,,,ORACLE,Q4_2021;FY_2021,,user,password,,,,,POSTGRESQL;AURORA_POSTGR
```

```
Marketing,,,ec2-a-b-c-d.eu-  
west-1.compute.amazonaws.com,1433,,target_audience,,MSSQL,customers.dbo,,user,password,,,,,AURORA  
HR,,,192.0.2.0,1433,,employees,,MSSQL,employees.%,true,,,,,AURORA_POSTGRESQL  
Customers,,secret-name,,,,,MYSQL,customers,,,,,AURORA_POSTGRESQL  
Analytics,,,198.51.100.0,8195,,STATISTICS,,DB2LUW,BI_REPORTS,,user,password,,,,,POSTGRESQL  
Products,,,203.0.113.0,8194,,,,,TERADATA,new_products,,user,password,,,,,REDSHIFT
```

上述範例會使用分號來分隔Sales資料庫的兩個結構描述名稱。它也會使用分號來分隔資料庫的兩個目標資料Sales庫移轉平台。

此外，上述範例會使用連線AWS Secrets Manager至Customers資料庫，並使用 Windows 驗證連線至資料HR料庫。

您可以從中建立新的 CSV 檔案或下載 CSV 檔案的範本，AWS SCT然後填寫必要的資訊。請確定 CSV 檔案的第一列包含與上述範例所示相同的欄名稱。

下載輸入 CSV 檔案的範本

1. 啟動 AWS SCT。
2. 選擇檔案，然後選擇新增多伺服器評估。
3. 選擇 [下載連線檔案範例]。

請確定您的 CSV 檔案包含範本所提供的下列值：

- 名稱 — 有助於識別資料庫的文字標籤。AWS SCT在評估報告中顯示此文字標籤。
- 說明 — 選擇性值，您可以在其中提供有關資料庫的其他資訊。
- 秘密管理員金鑰 — 將資料庫認證儲存在中的秘密名稱AWS Secrets Manager。若要使用 Secrets Manager，請確定您已將設定AWS檔儲存在中AWS SCT。如需詳細資訊，請參閱[使用 AWS Secrets Manager](#)。

Important

AWS SCT如果您在輸入檔案中包含伺服器 IP、連接埠、登入和密碼參數，則會忽略秘密管理員金鑰參數。

- Server IP — 來源資料庫伺服器的網域名稱服務 (DNS) 名稱或 IP 地址。
- 連接埠 — 用來連線到來源資料庫伺服器的連接埠。
- 服務名稱 — 如果您使用服務名稱連線至 Oracle 資料庫，則為要連線的 Oracle 服務名稱。

- 資料庫名稱 — 資料庫名稱。若為 Oracle 資料庫，請使用 Oracle 系統 ID (SID)。
- BigQuery path — 來源 BigQuery 資料庫服務帳戶金鑰檔案的路徑。如需建立此檔案的詳細資訊，請參閱[BigQuery作為來源的權限](#)。
- 來源引擎 — 來源資料庫的類型。使用以下其中一個值：
 - 適用於 Azure SQL 資料庫的資料庫。
 - Azure 突觸分析數據庫的 AZURE 突觸。
 - 谷歌查詢的 BigQuery 數據庫。
 - DB2ZOS 適用於 z/OS 資料庫的 IBM Db2。
 - DB2LUW 適用於 IBM Db2 LUW 資料庫。
 - 綠梅花的綠梅數據庫。
 - MSSQL Server 的 Microsoft SQL Server 資料庫。
 - 一個 MySQL 數據庫的 MYSQL。
 - 內泰扎的數據庫。
 - 甲骨文對於甲骨文數據庫。
 - 適用於 PostgreSQL 資料庫。
 - Redshift 的 Amazon Redshift Datashift。
 - 雪花雪花數據庫。
 - 適用於 SAP 日月光資料庫的系統資料庫。
 - 用於太元數據庫的太數據。
 - VERTICA 的一個 VERTICA 數據庫。
- 綱要名稱 — 要包含在評估報告中的資料庫綱要名稱。

對於 Azure SQL 資料庫，天藍突觸分析，BigQuery，網站，SAP ASE，雪花，和 SQL 伺服器，使用下列格式的結構描述名稱：

db_name.schema_name

db_name 以來源資料庫的名稱取代。

schema_name 以來源結構描述的名稱取代。

以雙引號括住包含點的資料庫或結構描述名稱，如下所示："database.name"."schema.name"

資料庫和結構描述名稱區分大小寫。

使用百分比 (%) 作為萬用字元，取代資料庫或結構描述名稱中任意數量的任何符號。上述範例使用百分比 (%) 作為萬用字元，將employees資料庫中的所有結構描述納入評估報告中。

- 使用視窗驗證 — 如果您使用視窗驗證連線到您的微軟 SQL 伺服器資料庫，請輸入 true。如需詳細資訊，請參閱[使用 Microsoft SQL 服務器作為源時使用視窗身份驗證](#)。
- 登入 — 連線到來源資料庫伺服器的使用者名稱。
- 密碼 — 連線到來源資料庫伺服器的密碼。
- 使用 SSL — 如果您使用安全通訊端層 (SSL) 連線到來源資料庫，請輸入 true。
- 信任存放區 — 用於 SSL 連線的信任存放區。
- 金鑰存放區 — 用於 SSL 連線的金鑰存放區。
- SSL 驗證 — 如果您使用憑證的 SSL 驗證，請輸入 true。
- 目標引擎 — 目標資料庫平台。使用下列值來指定評估報告中的一或多個目標：
 - 適用於 Aurora 與 MySQL 相容的資料庫。
 - 適用於 Aurora 與 PostgreSQL 相容的資料庫。
 - Babelfish 為 Aurora PostgreSQL 的資料庫。
 - 瑪麗亞資料庫的 MariaDB 資料庫。
 - MSSQL Server 的 Microsoft SQL Server 資料庫。
 - 一個 MySQL 數據庫的 MYSQL。
 - 甲骨文對於甲骨文數據庫。
 - 適用於 PostgreSQL 資料庫。
 - Redshift 的 Amazon Redshift Datashift。

使用像這樣的分號來分隔多個目標：MYSQL; MARIA_DB。目標數目會影響執行評估所需的時間。

尋找和檢視報表

多伺服器評估會產生兩種類型的報告：

- 所有來源資料庫的彙總報表。
- 來源資料庫中每個綱要名稱之目標資料庫的詳細評估報告。

報告儲存在您在 [新增多伺服器評估] 對話方塊中為 [位置] 選擇的目錄中。

若要存取詳細報表，您可以瀏覽子目錄，這些子目錄是依來源資料庫、結構描述名稱和目標資料庫引擎進行組織。

彙總報表會在四個資料欄中顯示有關目標資料庫轉換複雜度的資訊。這些資料行包括程式碼物件轉換、儲存物件、語法元素和轉換複雜度的相關資訊。

下列範例顯示將兩個 Oracle 資料庫結構描述轉 Amazon RDS for PostgreSQL

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

相同的四個欄會附加至每個指定的其他目標資料庫引擎的報表。

如需詳細資訊，請參閱以下內容。

彙總評估報告的輸出

中的彙總多伺服器資料庫移轉評估報告AWS Schema Conversion Tool是一個 CSV 檔案，其中包含下列欄位：

- 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*

若要收集資訊，請AWS SCT執行完整評估報告，然後依結構描述彙總報表。

在報告中，下列三個欄位會根據評估顯示可能自動轉換的百分比：

程式碼物件轉換%

結構描述中AWS SCT可以自動轉換或變更極少的程式碼物件百分比。Code 物件包含程序、函數、檢視和類似。

儲存物件轉換%

SCT 可以自動轉換或變更極少的儲存物件百分比。儲存物件包括資料表、索引、條件約束及類似項目。

語法元素轉換%

SCT 可以自動轉換的語法元素百分比。語法元素包括SELECTFROM、DELETE、和JOIN子句和類似項目。

轉換複雜度計算以行動項目的概念為基礎。行動項目反映在原始程式碼中發現的一種問題類型，您需要在移轉至特定目標期間手動修正這些問題。一個行動項目可以有多個出現位置。

加權量表可識別執行移轉的複雜程度。數字 1 代表複雜度的最低層級，數字 10 代表最高的複雜度。

轉換資料庫結構描述 AWS SCT

您可以使用 AWS Schema Conversion Tool (AWS SCT) 將現有資料庫結構描述從一個資料庫引擎轉換到另一個資料庫引擎。使用用AWS SCT戶界面轉換數據庫可能相當簡單，但是在進行轉換之前有幾件事需要考慮。

例如，您可以使用AWS SCT來執行下列動作：

- 您可以使用AWS SCT將現有的現場部署資料庫結構描述複製到執行相同引擎的 Amazon RDS 資料庫執行個體。您可以使用此功能分析移到雲端和變更授權類型的潛在成本節省。
- 在某些情況下，資料庫功能無法轉換為相等的 Amazon RDS 功能。如果您在 Amazon 彈性運算雲端 (Amazon EC2) 平台上託管和自行管理資料庫，則可以透過替換AWS服務來模擬這些功能。
- AWS SCT將您的線上交易處理 (OLTP) 資料庫架構轉換為 Amazon 關聯式資料庫服務 (Amazon RDS) MySQL 資料庫執行個體、Amazon Aurora 資料庫叢集或 PostgreSQL 資料庫執行個體的大部分程序自動化。來源和目標資料庫引擎包含許多不同的功能和功能，並AWS SCT嘗試盡可能在 Amazon RDS 資料庫執行個體中建立相等的結構描述。如果不可能進行直接轉換，AWS SCT 會提供一個可讓您採取動作的清單。

主題

- [在中建立移轉規則 AWS SCT](#)
- [使用轉換結構描述 AWS SCT](#)
- [處理手動轉換 AWS SCT](#)
- [更新並重新整理轉換後的結構描述 AWS SCT](#)
- [儲存並套用轉換後的結構描述 AWS SCT](#)
- [比較資料庫結構](#)
- [尋找相關的變形物件](#)

AWS SCT支援以下線上交易處理 (OLTP) 轉換。

來源資料庫	目標資料庫
適用於 z/OS 的 IBM Db2 (版本 12)	亞馬遜極光與 MySQL 相容的版本, 亞馬遜極光兼容版, MySQL

來源資料庫	目標資料庫
IBM Db2 LUW (版本 9.1、9.5、9.5、10.5、11.1 及 11.5 版本)	極光 MySQL, 極光, 瑪麗亞德, MySQL, PostgreSQL
微軟 SQL 數據庫	極光 MySQL, 極光, MySQL,
微軟 SQL 伺服器 (版本 2008 R2 及更高版本)	極光 MySQL, 極光, 巴貝爾魚為極光, 瑪麗亞德, 微軟 SQL 服務器, MySQL, PostgreSQL
版本 5.5 及更高版本	極光, MySQL, 您可以將結構描述和資料從 MySQL 遷移到極光 MySQL 資料庫叢集, 而無需使用 AWS SCT。如需詳細資訊, 請參閱 將資料移轉至 Amazon Aurora 資料庫叢集 。
甲骨文 (版本 10.2 及更高版本)	極光 MySQL, 極光, 瑪麗亞德, MySQL, 甲骨文, PostgreSQL
版本 9.1 及更高版本	極光 MySQL, 極光, MySQL,
日月光 (12.5、15.0、15.5、15.7 和 16.0)	極光 MySQL, 極光, 瑪麗亞德, MySQL, PostgreSQL

如需有關轉換資料倉儲結構描述的資訊, 請參閱 [使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT](#)。

若要將資料庫結構描述轉換為 Amazon RDS, 請執行下列高階步驟:

- [在中建立移轉規則 AWS SCT](#)— 在使用轉換結構描述之前 AWS SCT, 您可以設定規則來變更資料行的資料類型、將物件從一個結構描述移至另一個結構描述, 以及變更物件名稱。
- [使用轉換結構描述 AWS SCT](#)— AWS SCT 建立轉換後結構描述的本機版本供您檢閱, 但在您準備好之前, 不會將其套用至目標資料庫執行個體。
- [建立移轉評估報告 AWS SCT](#)— AWS SCT 建立資料庫遷移評估報告, 其中詳細說明無法自動轉換的結構描述元素。您可以使用此報告來識別需要在與來源資料庫相容的 Amazon RDS 資料庫執行個體中建立結構描述的位置。

- [處理手動轉換 AWS SCT](#)— 如果您有無法自動轉換的結構描述元素，則有兩種選擇：更新來源結構描述然後再次轉換，或在目標 Amazon RDS 資料庫執行個體中建立等效的結構描述元素。
- [更新並重新整理轉換後的結構描述 AWS SCT](#)— 您可以使用AWS SCT源數據庫中的最新模式更新項目。
- [儲存並套用轉換後的結構描述 AWS SCT](#)— 準備就緒後，請將本機專案中轉換的結構描述AWS SCT套用到目標 Amazon RDS 資料庫執行個體。

在中建立移轉規則 AWS SCT

在使用轉換結構描述之前AWS SCT，您可以設定移轉規則。中的移轉規則AWS SCT可以執行此類轉換，例如變更資料行的資料類型、將物件從一個結構描述移至另一個結構描述，以及變更物件名稱。例如，假設您的來源結構描述中有一組名為的資料表test_TABLE_NAME。您可以設定test_將首碼變更為目標結構描述demo_中的前置字元的規則。

Note

您只能為不同的來源和目標資料庫引擎建立移轉規則。

您可以建立執行下列工作的移轉規則：

- 新增、移除或取代字首
- 新增、移除或取代尾碼
- 變更資料行排序規則
- 變更資料類型
- 變更char、varchar、nvarchar和string資料類型的長度
- 移動物件
- 重新命名物件

您可以為下列物件建立移轉規則：

- 資料庫
- 結構描述
- 資料表

- 資料行

建立移轉規則

您可以建立移轉規則，並將規則儲存為專案的一部分。開啟專案後，請使用下列程序來建立移轉規則。

若要建立移轉規則

1. 在 [檢視] 功能表上，選擇 [對應檢視]。
2. 在伺服器對映中，選擇一對來源伺服器和目標伺服器。
3. 選擇 [新增移轉規則]。[轉換規則] 對話方塊隨即出現。
4. 選擇 Add new rule (新增新的規則)。新的資料列會新增到規則清單。
5. 設定您的規則：
 - a. 在 Name (名稱) 中，輸入規則名稱。
 - b. 在 For (對象) 中，選擇規則套用的物件類型。
 - c. 針對其中，輸入要套用至物件的篩選器，然後再套用移轉規則。將使用類似子句來評估 where 子句。您可以輸入確切名稱以選擇一個物件，也可以輸入模式以選擇多個物件。

where 子句可用的欄位，根據物件類型而不同。例如，如果物件類型是結構描述，則只有一個欄位可用於結構描述名稱。
 - d. 在動作中，選擇您要建立的移轉規則類型。
 - e. 根據規則類型，輸入一或兩個其他值。例如，若要更名物件，請輸入物件的新名稱。若要取代字首，請輸入舊字首和新字首。

對於 char、varchar、nvarchar 和字串資料類型，您可以使用乘法運算子變更資料類型長度。例如，該 %*4 值將 varchar(10) 數據類型轉換為 varchar(40)。

6. 設定遷移規則後，請選擇 [儲存] 以儲存規則。您也可以選擇 Cancel (取消) 以取消變更。

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS **doesn't support** more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' **add prefix 'demo_%'**

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, Add new rule, Save all, Close, Export script for DMS, Import script into SCT

7. 完成新增、編輯和刪除規則之後，請選擇 **Save All (全部儲存)** 儲存所有變更。

8. 選擇 **[關閉]** 以關閉 **[轉換規則]** 對話方塊。

您可以使用切換圖示來關閉移轉規則，而不刪除它。您可以使用複製圖示來複製現有的移轉規則。您可以使用鉛筆圖示來編輯現有的移轉規則。您可以使用刪除圖示來刪除現有的移轉規則。若要儲存您對遷移規則所做的任何變更，請選擇 **[全部儲存]**。

匯出移轉規則

如果您使用AWS DMS將資料從來源資料庫移轉至目標資料庫，則可以提供移轉規則的相關資訊AWS DMS。如需有關工作的詳細資訊，請參閱[使用AWS Database Migration Service複製工作](#)。

匯出移轉規則

1. 在中AWS Schema Conversion Tool，選擇 **[檢視]** 功能表上的 **[對應檢視]**。
2. 在 **[移轉規則]** 中，選擇移轉規則，然後選擇 **[修改移轉規則]**。
3. 選擇「匯出」的指令碼AWS DMS。
4. 瀏覽至您要儲存指令碼的位置，然後選擇 **Save (儲存)**。您的遷移規則會儲存為可供使用的 JSON 指令碼AWS DMS。

使用轉換結構描述 AWS SCT

將專案連線到來源資料庫和目標 Amazon RDS 資料庫執行個體後，AWS Schema Conversion Tool 專案會在左側面板中顯示來源資料庫的結構描述。結構描述會呈現為樹狀檢視格式，並且延遲載入樹狀目錄的每個節點。當您在樹狀檢視中選擇節點，AWS SCT 會在此時向來源資料庫請求結構描述資訊。

您可以從來源資料庫選擇結構描述項目，然後將結構描述轉換到適用於目標資料庫執行個體之資料庫引擎的對等結構描述。您可以從來源資料庫選擇任何結構描述項目加以轉換。如果您選擇的結構描述項目相依於父項目，則 AWS SCT 也會產生父項目的結構描述。例如，假設您選擇要轉換的表格。如果是這樣，AWS SCT 會產生資料表的結構定義，以及資料表所在的資料庫。

轉換結構描述

若要從來源資料庫轉換結構描述，請選取要轉換的結構描述名稱核取方塊。接下來，從項目的左側面板中選擇此模式。AWS SCT 以藍色反白結構描述名稱。開啟綱要的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [轉換結構描述]，如下所示。

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
Create report
Convert schema
Register agent
Compare schema
Load schema
Hide schema
Refresh from database
Collect statistics
Upload statistics
Create DMS task
Create Local & DMS task
Create Local task
Add virtual partitioning
Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

當您轉換來源資料庫的結構描述後，您可以從專案左側面板選擇結構描述項目，並在專案中央面板查看轉換後的結構描述。中央下方面板顯示用來建立轉換後結構描述的屬性和 SQL 命令，如下所示。

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > AdventureWorks2012_CS > Tables [4] > Account. The main pane shows the SQL definition for the [Account] table in the [dbo] schema, including columns like [ID], [AccountNo], [CurrencyID], [Description], [CustomerID], [StateID], [AccountBalance], [BlockedAmount], [Opendate], [Closedate], [RespManagerID], and [BankID]. Below this, the target Amazon RDS for MySQL table structure is shown, with the table name [Account] in the [LARGE_DB_SS_dbo] schema. The target table definition uses MySQL-compatible data types and includes a primary key constraint on the [ID] column.

轉換結構描述之後，您可以儲存您的專案。來自來源資料庫的結構描述資訊會與您的專案一起儲存。此功能表示您可以在不連線至來源資料庫的情況下離線工作。AWS SCT如果您為來源資料庫選擇「從資料庫重新整理」，則連線到來源資料庫以更新專案中的結構描述。如需詳細資訊，請參閱[更新並重新整理轉換後的結構描述 AWS SCT](#)。

您可以為無法自動轉換的項目建立資料庫遷移評估報告。評估報告可以用來識別和解決無法自動轉換的結構描述項目。如需詳細資訊，請參閱[建立移轉評估報告AWS SCT](#)。

當 AWS SCT 產生轉換後的結構描述時，並不會立即將它套用到目標資料庫執行個體。反之，轉換後的結構描述會儲存於本機，直到您準備將其套用至目標資料庫執行個體。如需詳細資訊，請參閱[應用轉換後的模式](#)。

編輯轉換的結構

您可以編輯轉換後的結構描述，並將變更儲存在專案中。

若要編輯轉換後的結構描述

1. 在顯示來源資料庫結構描述的左側窗格中，選擇您要編輯其轉換後結構描述的結構描述項目。
2. 在顯示所選項目之轉換後結構描述的中央下方面板，選擇 SQL 索引標籤。

3. 在針對 SQL 索引標籤顯示的文字中，視需要變更結構描述。當您更新結構描述後，結構描述會自動與您的專案一起儲存。

The screenshot shows a window titled "Target Amazon RDS for MySQL table: Account". Below the title bar are several tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". The "SQL" tab is selected, displaying the following SQL code:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );

```

在您進行更新後，您對轉換後結構描述進行的變更會與專案一起儲存。如果您新近從來源資料庫轉換結構描述項目，並曾對先前已轉換結構描述的該項目進行了更新，則會根據您的來源資料庫，以新轉換的結構描述項目取代這些現有更新。

清除轉換的結構描述

在您將結構描述套用到目標資料庫執行個體之前，AWS SCT 只會將轉換後的結構描述存放在本機專案中。您可以選擇資料庫執行個體的樹狀檢視節點，然後選擇「從資料庫重新整理」，從專案中清除已規劃的結構描述。由於尚未將結構描述寫入目標資料庫執行個體，因此從資料庫重新整理會移除 AWS SCT 專案中已規劃的結構描述元素，以符合來源資料庫執行個體中的資料。

處理手動轉換 AWS SCT

評估報告包含無法自動轉換為目標 Amazon RDS 資料庫執行個體資料庫引擎的項目清單。對於每個無法轉換的項目，Action Items (動作項目) 索引標籤上都會有一個動作項目。

您可以下列方式回應評估報告中的動作項目：

- 修改您的來源資料庫結構描述。
- 修改您的目標資料庫結構描述。

修改來源結構描述

對於某些項目，將來源資料庫中的資料庫結構描述修改為可以自動轉換的結構描述可能會比較容易。首先，驗證新的變更與您的應用程式架構相容，然後更新來源資料庫中的結構描述。最後，使用更新後的結構描述資訊來重新整理您的專案。您可以轉換已更新的結構描述，並產生新的資料庫遷移評估報告。已在來源結構描述中變更的項目，不會再顯示動作項目。

此程序的優點是當您從來源資料庫重新整理時，已更新的結構描述永遠可用。

修改目標結構描述

對於某些項目，將轉換後的結構描述套用到您的目標資料庫，然後針對無法自動轉換的項目，手動新增同等結構描述項目到目標資料庫，可能會比較輕鬆。您可以透過套用結構描述，寫入所有可自動轉換至目標資料庫執行個體的結構描述。如需詳細資訊，請參閱[儲存並套用轉換後的結構描述 AWS SCT](#)。

寫入至您的目標資料庫執行個體的結構描述，不包含無法自動轉換的項目。將結構描述套用到目標資料庫執行個體後，接著您可以在目標資料庫執行個體中手動建立等同於來源資料庫中項目的結構描述。資料庫遷移評估報告中的動作項目包含如何建立相同結構描述的建議。

Warning

如果您在目標資料庫執行個體中手動建立結構描述，請儲存一份手動作業的複本。如果您再次從專案套用轉換後的結構描述到目標資料庫執行個體，它會覆寫您已完成的手動作業。

在部分案例中，您無法在目標資料庫執行個體中建立相同的結構描述。您可能必須重新架構部分的應用程式與資料庫，以使用資料庫引擎為您的目標資料庫執行個體提供的功能。在其他案例中，您可以直接忽略無法自動轉換的結構描述。

更新並重新整理轉換後的結構描述 AWS SCT

您可以同時更新 AWS Schema Conversion Tool 專案中的來源結構描述和目標結構描述。

- 來源 — 如果您更新來源資料庫的結構描述，請使用來源資料庫中的最新結構描述AWS SCT取代專案中的結構描述。使用此功能，如果已對來源資料庫的結構描述進行了變更，您可以更新您的專案。

- 目標 — 如果您更新目標 Amazon RDS 資料庫執行個體的結構描述，請使用目標資料庫執行個體中的最新結構描述AWS SCT取代專案中的結構描述。如果您尚未套用任何結構描述到您的目標資料庫執行個體，AWS SCT 會從您的專案清除轉換後的結構描述。然後，您可以從來源資料庫轉換結構描述以用於全新的目標資料庫執行個體。

您可以選擇「從資料庫重新整理」來更新AWS SCT專案中的結構描述。

Note

重新整理結構描述時，只會在需要時AWS SCT載入中繼資料。若要完全載入資料庫的所有結構描述，請開啟綱要的內容 (按一下滑鼠右鍵) 功能表，然後選擇「載入結構描述」。例如，您可以使用此選項一次載入資料庫的中繼資料，然後離線工作。

儲存並套用轉換後的結構描述 AWS SCT

當 AWS Schema Conversion Tool 產生轉換後的結構描述 (如 [使用轉換結構描述 AWS SCT](#) 所示) 時，它並不會立即將轉換後的結構描述套用到目標資料庫執行個體。反之，轉換後的結構描述會儲存於本機專案中，直到您準備將其套用至目標資料庫執行個體。利用此功能，您可以處理無法自動轉換到目標資料庫引擎的結構描述項目。如需無法自動轉換的項目的詳細資訊，請參閱[建立移轉評估報告AWS SCT](#)。

您可以選擇讓工具將轉換後的結構描述儲存為 SQL 指令碼檔案，之後再套用結構描述到目標資料庫執行個體。您也可以讓工具直接將轉換後的結構描述套用到目標資料庫執行個體。

將轉換後的結構定義儲存為檔案

您可以將轉換後的結構描述儲存為文字檔案中的 SQL 指令碼。使用此方法，您可以修改從 AWS SCT 產生的 SQL 指令碼，來解決工具無法自動轉換的項目。然後，您可以在目標資料庫執行個體上執行更新後的指令碼，以將轉換後的結構描述套用到目標資料庫。

若要將轉換的結構描述儲存為 SQL 命令檔

1. 選擇您的架構並開啟內容 (按一下滑鼠右鍵) 功能表。
2. 選擇「另存為 SQL」。
3. 輸入檔案名稱，然後選擇 [儲存]。
4. 使用下列其中一個選項儲存轉換後的結構描述：

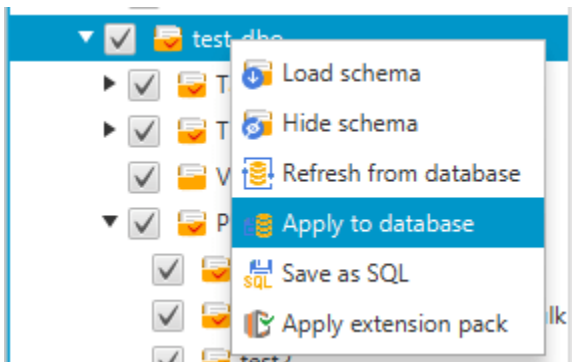
- 單一檔案
- 每個階段單一檔案
- 每個陳述式單一檔案

若要選擇 SQL 命令檔的格式

1. 在 [設定] 功能表上，選擇 [專案設定]。
2. 選擇儲存指令碼。
3. 對於供應商，選擇數據庫平台。
4. 在「儲存 SQL 命令檔至」中，選擇儲存資料庫結構描述指令碼的方式。
5. 選擇「確定」以儲存設定。

應用轉換後的模式

當您準備好將轉換後的結構描述套用到目標 Amazon RDS 資料庫執行個體時，請從專案的右側面板中選擇結構描述元素。開啟結構描述元素的內容 (按一下右鍵) 選單，然後選擇 Apply to database (套用至資料庫)，如下所示。



擴充套件架構

第一次套用轉換後的結構描述到目標資料庫執行個體時，AWS SCT 會新增額外的結構描述到您的目標資料庫執行個體。此結構描述會實作來源資料庫的系統功能，當您將轉換後結構描述寫入至目標資料庫執行個體時需要這些功能。此結構描述稱為延伸套件結構描述。

請勿修改延伸套件結構描述，否則寫入至目標資料庫執行個體的轉換後結構描述會發生未預期的結果。當您的結構描述已完全遷移至目標資料庫執行個體，而且您不再需要 AWS SCT 時，即可刪除延伸套件結構描述。

延伸套件結構描述的命名是根據您的來源資料庫，如下所示：

- 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 日月光 : `aws_sapase_ext`

如需詳細資訊，請參閱[使用 AWS SCT 擴展包中的 AWS Lambda 功能](#)。

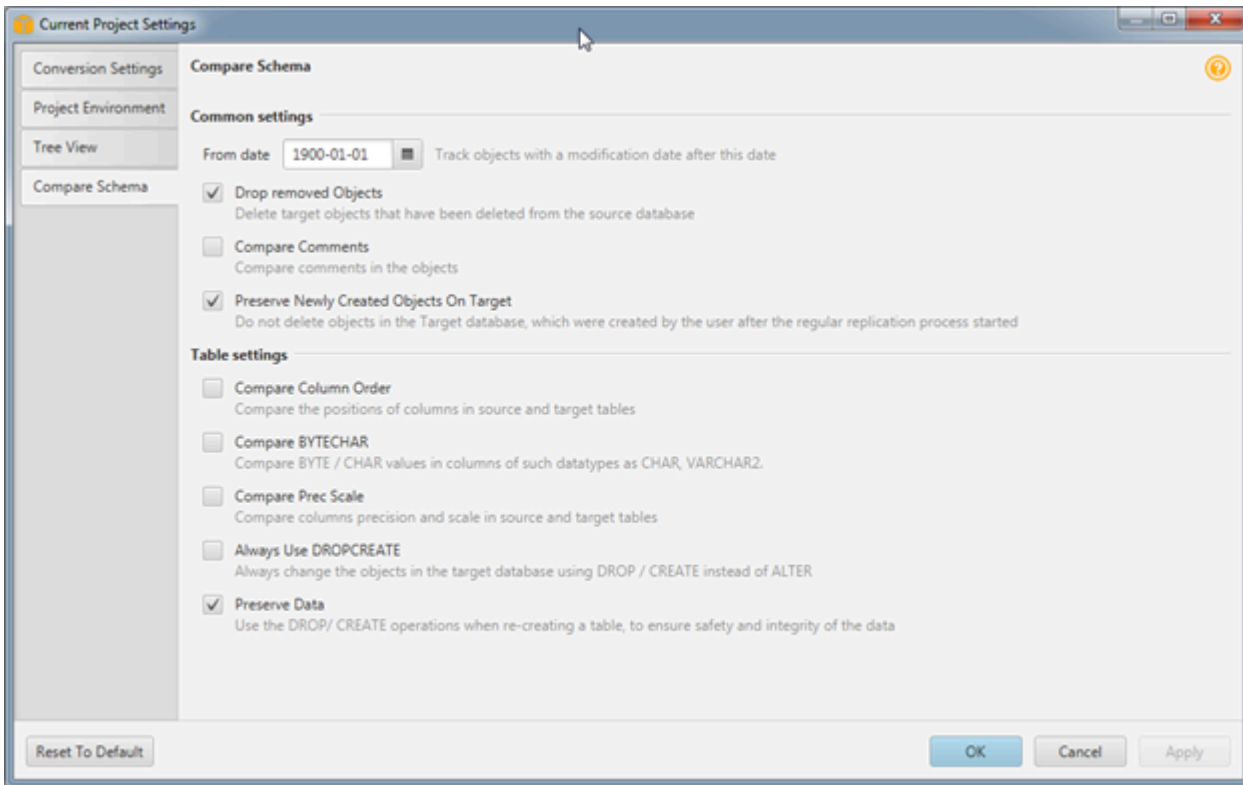
比較資料庫結構

如果您在遷移後對來源或目標結構描述進行變更，您可以使用 AWS SCT 比較兩個資料庫結構描述。您可以比較與來源結構描述相同或更早版本的結構描述。

支援以下結構描述比較：

- Oracle 對 Oracle ，版本 12.1.0.2.0、11.1.0.7.0、11.2.0.1.0、10
- SQL Server 對 SQL Server ，版本 2016、2014、2012、2008 RD2、2008
- 與 PostgreSQL 和極光兼容版本，版本 9.6，9.5.9，9.5.4 版
- MySQL 對 MySQL ，版本 5.6.36、5.7.17、5.5

您可以在 Project Settings (專案設定) 頁面的 Compare Schema (比較結構描述) 索引標籤上，指定結構描述比較的設定。












若要比較結構描述，請選擇結構描述，AWS SCT 會指出兩個結構描述之間不同的物件以及相同的物件。

若要比較兩個結構描述

1. 開啟現有的 AWS SCT 專案，或是建立專案並連接到來源與目標端點。
2. 選擇您想要比較的結構描述。
3. 開啟右鍵功能表 (按一下滑鼠右鍵)，然後選擇比較綱要

AWS SCT 會在物件圖示上加上黑色圓圈，指出兩個結構描述之間不同的物件。

- ▶  testddl_tbl_view_02
- ▼  testddl_trg_tbl_01
 -  Constraints
 -  Indexes
 - ▶  Triggers [1]
 - ▶  testddl_trg_tbl_02
 - ▶  testddl_trg_tbl_03
 -  Foreign tables
 - ▼  Views [3]

您可以將結構描述比較的結果套用到單一物件、單一物件類別，或整個結構描述。選擇您要套用結果的類別、物件或是結構描述旁的方塊。

尋找相關的變形物件

結構描述轉換後，在某些情況下，AWS SCT 可能會為來源資料庫的一個結構描述物件建立數個物件。例如，執行 Oracle 到 PostgreSQL 的轉換時，AWS SCT 會採用每個 Oracle 觸發並將其轉換到 PostgreSQL 目標上的觸發和觸發函數。此外，當 AWS SCT 將 Oracle 套件函數或程序轉換到 PostgreSQL，它會建立對等函數和 INIT 函數，這些函數必須執行為 init 區塊，程序或函數才能執行。

下列程序可讓您查看在結構描述轉換後建立的所有相關物件。

若要檢視在結構描述轉換期間建立的相關物件

1. 結構描述轉換後，在目標樹狀檢視中選擇轉換物件。
2. 選擇 Related Converted Objects (相關轉換物件) 索引標籤。
3. 檢視相關目標物件的清單。

使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT

AWS Schema Conversion Tool(AWS SCT) 會將您的資料倉儲結構描述轉換為 Amazon Redshift 資料庫結構描述的大部分程序自動化。來源和目標資料庫引擎有許多不同的特性和功能，而 AWS SCT 會盡可能嘗試在您的目標資料庫中建立對等結構描述。如果不可能進行直接轉換，AWS SCT 會提供評估報告，報告內有一份您可採取之動作的清單。透過使用 AWS SCT，您可以管理金鑰、對應資料類型和物件，並建立手動轉換。

AWS SCT可以將下列資料倉儲結構描述轉換為亞馬遜紅移。

- Amazon Redshift
- 天藍突觸分析 (版本 10)
- BigQuery
- 綠梅資料庫 (版本 4.3)
- 微軟 SQL 伺服器 (2008 年及更高版本)
- 尼泰扎 (版本 7.0.3 及更高版本)
- 甲骨文 (版本 10.2 及更高版本)
- 雪花 (版本 3)
- 太數據 (版本 13 及更高版本)
- Vertica (版本 7.2 及更高版本)

如需有關轉換線上交易處理 (OLTP) 資料庫結構描述的資訊，請參閱 [〈〉](#)。[轉換資料庫結構描述 AWS SCT](#)

若要轉換資料倉儲結構描述，請執行下列步驟：

1. 指定最佳化策略和規則，並指定您要AWS SCT使用的移轉規則。您可以設定變更欄的資料類型、將物件從某個結構描述移至另一個結構描述，以及變更物件名稱的規則。

您可以在 [設定] 中指定最佳化和移轉規則。如需最佳化策略的詳細資訊，請參閱[選擇與搭配使用的最佳化策略和規則 AWS SCT](#)。如需移轉規則的詳細資訊，請參閱。[在中建立移轉規則 AWS SCT](#)

2. 提供關於來源資料倉儲的統計資料，讓 AWS SCT 可將您的資料倉儲轉換最佳化。您可以直接從資料庫收集統計資料，也可以上傳現有的統計資料檔案。如需有關提供資料倉儲統計資料的詳細資訊，請參閱[收集或上傳統計資料 AWS SCT](#)。

3. 建立資料庫遷移評估報告，詳細說明無法自動轉換的結構描述元素。您可以使用此報告來識別哪些地方需要您在目標資料庫中手動建立與來源資料庫相容的結構描述。如需評估報告的詳細資訊，請參閱[建立移轉評估報告AWS SCT](#)。
4. 轉換結構描述。AWS SCT 會建立轉換後結構描述的本機版本供您檢閱，但在您準備好之前，不會套用到您的目標資料庫。如需轉換的詳細資訊，請參閱[使用轉換結構描述 AWS SCT](#)。
5. 轉換結構描述後，您可以管理和編輯您的索引鍵。索引鍵管理是資料倉儲轉換的核心。如需對應金鑰的詳細資訊，請參閱[管理和自訂金鑰 AWS SCT](#)。
6. 如果有無法自動轉換的結構描述元素，您有兩個選擇：更新來源結構描述然後再次轉換，或是在目標資料庫中建立同等結構描述元素。如需手動轉換結構描述元素的詳細資訊，請參閱[處理手動轉換 AWS SCT](#)。如需更新來源結構描述的詳細資訊，請參閱[更新並重新整理轉換後的結構描述 AWS SCT](#)。
7. 準備好後，您可將轉換的結構描述套用到目標資料庫。如需儲存及套用轉換的結構描述的詳細資訊，請參閱[儲存並套用轉換後的結構描述 AWS SCT](#)。

亞馬遜紅移作為目標的許可

亞馬遜紅移作為目標所需的許可如下所示：

- 在數據庫上創建-允許在數據庫中創建新的模式。
- 在模式上創建-允許在數據庫模式中創建對象。
- 授予使用語言 — 允許在數據庫中創建新的函數和過程。
- 授予架構 pg_目錄中的所有表格選取選取 — 為使用者提供有關 Amazon Redshift 叢集的系統資訊。
- 在 pg_class_info 上授予選擇-為用戶提供有關表分佈樣式的信息。

您可以使用下列程式碼範例來建立資料庫使用者並授與權限。

```
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;
```

在前面的範例中，用您的使用者#### *user_name*。然後，將 *db_name* 替換為目標亞馬遜紅移數據庫的名稱。接下來，將##### 紅移模式的名稱。針對要套用轉換後的程式碼或移轉資料的每個目標結構描述重複此 GRANT CREATE ON SCHEMA 作業。最後，用安全的#####。

您可以在目標亞馬遜紅移資料庫上套用擴充套件。擴充套件是一種附加模組，可模擬將物件轉換為 Amazon Redshift 時所需的來源資料庫功能。如需詳細資訊，請參閱 [使用 AWS SCT 擴充套件](#)。

對於此操作，AWS SCT 需要代表您存取 Amazon S3 儲存貯體的權限。若要提供此權限，請使用下列政策建立 AWS Identity and Access Management (IAM) 使用者。

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

選擇與搭配使用的最佳化策略和規則 AWS SCT

為了最佳化 AWS Schema Conversion Tool 轉換資料倉儲結構描述的方式，您可以選擇希望工具使用的策略和規則。在轉換結構描述並檢閱建議的索引鍵後，您可以調整規則或變更策略，以取得您想要的結果。

若要選擇最佳化策略和規則

1. 選擇 Settings (設定)，然後選擇 Project Settings (專案設定)。Current project settings (目前專案設定) 對話方塊會出現。
2. 在左側窗格中選擇 Optimization Strategies (最佳化策略)。最佳化策略會顯示在右側窗格中，並已選取預設值。
3. 對於 Strategy Sector (策略區段)，選擇您要使用的最佳化策略。您可以選擇下列項目：
 - 使用中繼資料，忽略統計資訊 — 在此策略中，只有中繼資料中的資訊用於最佳化決策。例如，如果來源資料表上有多個索引，則會使用來源資料庫排序順序，且第一個索引會變成散發索引鍵。
 - 忽略中繼資料，使用統計資訊 — 在此策略中，最佳化決策僅衍生自統計資訊。這個策略僅套用到有提供統計資料的資料表和欄。如需詳細資訊，請參閱[收集或上傳統計資料 AWS SCT](#)。
 - 使用中繼資料並使用統計資訊 — 在此策略中，中繼資料和統計資料均用於最佳化決策。
4. 選擇最佳化策略之後，您可以選擇您要使用的規則。您可以選擇下列項目：
 - Choose Distribution Key and Sort Keys using metadata (使用中繼資料來選擇散發索引鍵和排序索引鍵)
 - Choose fact table and appropriate dimension for collation (選擇事實資料表和定序的適當維度)
 - Analyze cardinality of indexes' columns (分析索引欄的基數)
 - 從查詢記錄資料表中尋找最常用的資料表和資料欄

對於每個規則，您可以輸入排序索引鍵的權重和散發索引鍵的權重。AWS SCT 會在轉換您的結構描述時使用您選擇的權重。當您稍後檢閱建議的索引鍵，如果不滿意結果，可以返回此處並變更設定。如需詳細資訊，請參閱[管理和自訂金鑰 AWS SCT](#)。

收集或上傳統計資料 AWS SCT

為了最佳化 AWS Schema Conversion Tool 轉換資料倉儲結構描述的方式，您可以從來源資料庫提供工具可以使用的統計資料。您可以直接從資料庫收集統計資料，也可以上傳現有的統計資料檔案。

在中建立移轉規則 AWS SCT

在使用轉換結構描述之前AWS SCT，您可以設定移轉規則。遷移規則可以執行以下操作，例如更改列的數據類型，將對象從一個模式移動到另一個模式，以及更改對象的名稱。例如，假設您的來源結構描述中有一組名為的資料表test_TABLE_NAME。您可以設定test_將首碼變更為目標結構描述demo_中前置詞的規則。

Note

您只能為不同的來源和目標資料庫引擎建立移轉規則。

您可以建立執行下列工作的移轉規則：

- 新增、移除或取代字首
- 新增、移除或取代尾碼
- 變更資料行排序規則
- 變更資料類型
- 變更char、varchar、nvarchar和string資料類型的長度
- 移動物件
- 重新命名物件

您可以為下列物件建立移轉規則：

- 資料庫
- 結構描述
- 資料表
- 資料行

建立移轉規則

您可以建立移轉規則，並將規則儲存為專案的一部分。開啟專案後，請使用下列程序來建立移轉規則。

若要建立移轉規則

1. 在 [檢視] 功能表上，選擇 [對應檢視]。

2. 在伺服器對映中，選擇一對來源伺服器 and 目標伺服器。
3. 選擇 [新增移轉規則]。[轉換規則] 對話方塊隨即出現。
4. 選擇 Add new rule (新增新的規則)。新的資料列會新增到規則清單。
5. 設定您的規則：
 - a. 在 Name (名稱) 中，輸入規則名稱。
 - b. 在 For (對象) 中，選擇規則套用的物件類型。
 - c. 針對其中，輸入要套用至物件的篩選器，然後再套用移轉規則。將使用類似子句來評估 where 子句。您可以輸入確切名稱以選擇一個物件，也可以輸入模式以選擇多個物件。

where 子句可用的欄位，根據物件類型而不同。例如，如果物件類型是結構描述，則只有一個欄位可用於結構描述名稱。
 - d. 在動作中，選擇您要建立的移轉規則類型。
 - e. 根據規則類型，輸入一或兩個其他值。例如，若要更名物件，請輸入物件的新名稱。若要取代字首，請輸入舊字首和新字首。
6. 設定遷移規則後，請選擇 [儲存] 以儲存規則。您也可以選擇 Cancel (取消) 以取消變更。

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS [doesn't support](#) more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For tables where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, + Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. 完成新增、編輯和刪除規則之後，請選擇 Save All (全部儲存) 儲存所有變更。
8. 選擇 [關閉] 以關閉 [轉換規則] 對話方塊。

您可以使用切換圖示來關閉移轉規則，而不刪除它。您可以使用複製圖示來複製現有的移轉規則。您可以使用鉛筆圖示來編輯現有的移轉規則。您可以使用刪除圖示來刪除現有的移轉規則。若要儲存您對遷移規則所做的任何變更，請選擇 [全部儲存]。

匯出移轉規則

如果您使用 AWS Database Migration Service (AWS DMS) 將資料從來源資料庫遷移到目標資料庫，則可以提供移轉規則的相關資訊 AWS DMS。如需有關工作的詳細資訊，請參閱 [使用 AWS Database Migration Service 複製工作](#)。

匯出移轉規則

1. 在中 AWS Schema Conversion Tool，選擇 [檢視] 功能表上的 [對應檢視]。
2. 在 [移轉規則] 中，選擇移轉規則，然後選擇 [修改移轉規則]。
3. 選擇「匯出」的指令碼 AWS DMS。
4. 瀏覽至您要儲存指令碼的位置，然後選擇 Save (儲存)。您的遷移規則會儲存為可供使用的 JSON 指令碼 AWS DMS。

使用轉換結構描述 AWS SCT

在將您的專案同時連接至來源資料庫與目標資料庫後，您的 AWS Schema Conversion Tool 專案會在左側面板中顯示來自來源資料庫的結構描述。結構描述會呈現為樹狀檢視格式，並且延遲載入樹狀目錄的每個節點。當您在樹狀檢視中選擇節點，AWS SCT 會在此時向來源資料庫請求結構描述資訊。

您可以從來源資料庫選擇結構描述項目，然後將結構描述轉換到適用於目標資料庫之資料庫引擎的對等結構描述。您可以從來源資料庫選擇任何結構描述項目加以轉換。如果您選擇的結構描述項目相依於父項目，則 AWS SCT 也會產生父項目的結構描述。例如，如果您選擇表格中的欄加以轉換，則 AWS SCT 會為該欄、欄所在的表格，以及表格所在的資料庫都產生結構描述。

轉換結構描述

若要從來源資料庫轉換結構描述，請選取要轉換的結構描述名稱核取方塊。接下來，從項目的左側面板中選擇此模式。AWS SCT 以藍色反白結構描述名稱。開啟綱要的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [轉換結構描述]，如下所示。

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
Create report
Convert schema
Register agent
Compare schema
Load schema
Hide schema
Refresh from database
Collect statistics
Upload statistics
Create DMS task
Create Local & DMS task
Create Local task
Add virtual partitioning
Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

當您轉換來源資料庫的結構描述後，您可以從專案左側面板選擇結構描述項目，並在專案中央面板查看轉換後的結構描述。中央下方面板顯示用來建立轉換後結構描述的屬性和 SQL 命令，如下所示。

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source SQL Server instance 'SQL Server - ec2-52-17-21-76.eu-west-1.cc' with its databases and schemas. The 'dbo' schema and 'Account' table are selected. The main pane shows the SQL definition for the 'Account' table, including columns like ID, AccountNo, CurrencyID, Description, CustomerID, StateID, AccountBalance, BlockedAmount, Operdate, Closedate, RespManagerID, and BankID. Below this, the 'Target Amazon RDS for MySQL table: Account' pane shows the converted MySQL table structure, using 'CREATE TABLE IF NOT EXISTS' and 'NUMERIC' data types.

轉換結構描述之後，您可以儲存您的專案。來自來源資料庫的結構描述資訊會與您的專案一起儲存。此功能表示您可以在不連線至來源資料庫的情況下離線工作。AWS SCT如果您為來源資料庫選擇「從資料庫重新整理」，則連線到來源資料庫以更新專案中的結構描述。如需詳細資訊，請參閱[更新並重新整理轉換後的結構描述 AWS SCT](#)。

您可以為無法自動轉換的項目建立資料庫遷移評估報告。評估報告可以用來識別和解決無法自動轉換的結構描述項目。如需詳細資訊，請參閱[建立移轉評估報告AWS SCT](#)。

當 AWS SCT 產生轉換後的結構描述時，並不會將它立即套用到目標資料庫。反之，轉換後的結構描述會儲存於本機，直到您準備將其套用至目標資料庫。如需詳細資訊，請參閱[套用轉換的結構定義](#)。

編輯轉換的結構

您可以編輯轉換後的結構描述，並將變更儲存在專案中。

若要編輯轉換後的結構描述

1. 在顯示來源資料庫結構描述的左側窗格中，選擇您要編輯其轉換後結構描述的結構描述項目。
2. 在顯示所選項目之轉換後結構描述的中央下方面板，選擇 SQL 索引標籤。

3. 在針對 SQL 索引標籤顯示的文字中，視需要變更結構描述。當您更新結構描述後，結構描述會自動與您的專案一起儲存。

```

▼ Target Amazon RDS for MySQL table: Account
Properties SQL Columns Partition columns Apply status Key management
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );

```

在您進行更新後，您對轉換後結構描述進行的變更會與專案一起儲存。如果您新近從來源資料庫轉換結構描述項目，並曾對先前已轉換結構描述的該項目進行了更新，則會根據您的來源資料庫，以新轉換的結構描述項目取代這些現有更新。

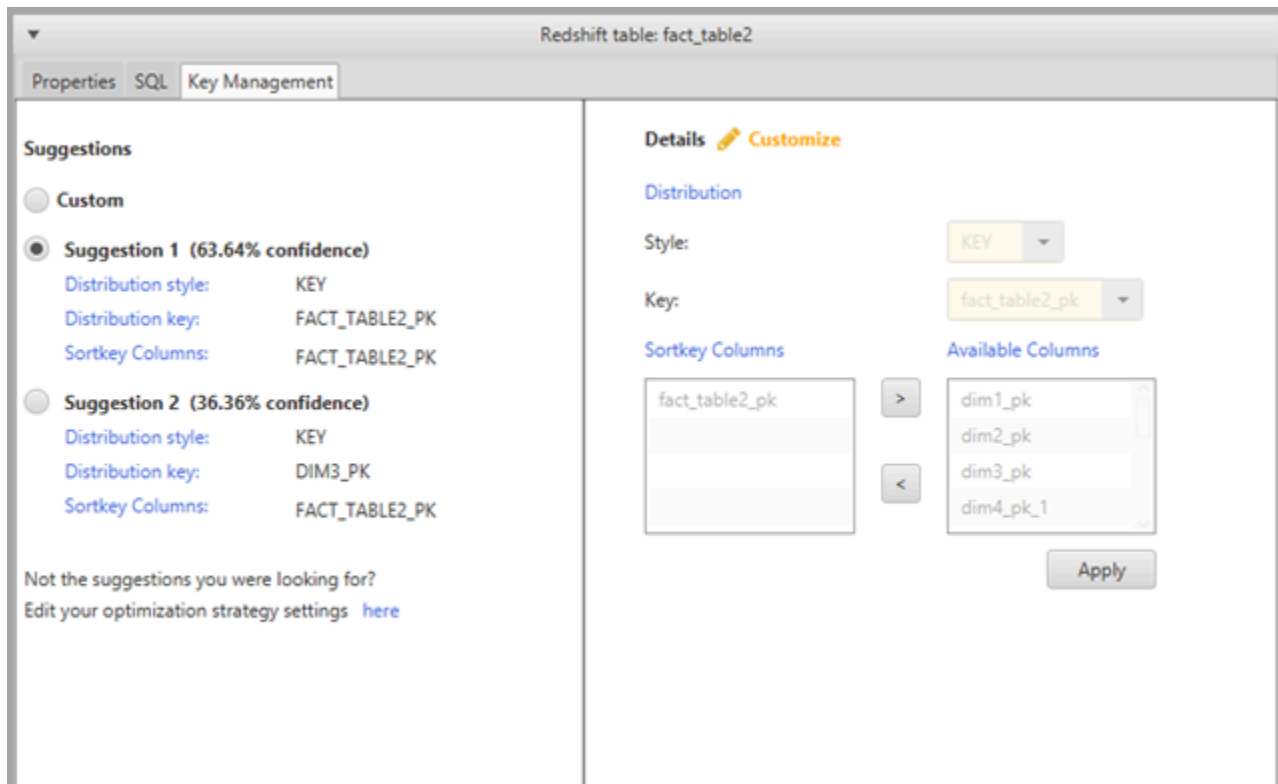
清除轉換的結構描述

在您將結構描述套用到目標資料庫之前，AWS SCT 只會將轉換後的結構描述儲存在本機專案中。您可以透過選擇目標資料庫的樹狀檢視節點，然後選擇 Refresh from Database (從資料庫重新整理)，從專案中清除計劃的結構描述。由於尚未將任何結構描述寫入目標資料庫，所以從資料庫重新整理會移除 AWS SCT 專案中計劃的結構描述元素，以符合目標資料庫中存在的元素。

管理和自訂金鑰 AWS SCT

在使用 AWS Schema Conversion Tool 轉換您的結構描述之後，您可以管理和編輯您的索引鍵。索引鍵管理是資料倉儲轉換的核心。

若要管理索引鍵，請在目標資料庫中選擇一個表格，然後選擇 Key Management (索引鍵管理) 索引標籤，如下所示。



左側窗格包含索引鍵建議，也包含每個建議的信心度。您可以選擇其中一個建議，也可以在右側窗格中編輯以自訂索引鍵。

如果索引鍵的選擇看起來不如您的預期，您可以編輯您的最佳化策略，然後重試轉換。如需詳細資訊，請參閱[選擇與搭配使用的最佳化策略和規則 AWS SCT](#)。

相關主題

- [選擇最佳的排序鍵](#)
- [選擇最佳的分佈風格](#)

在中建立和使用評估報告 AWS SCT

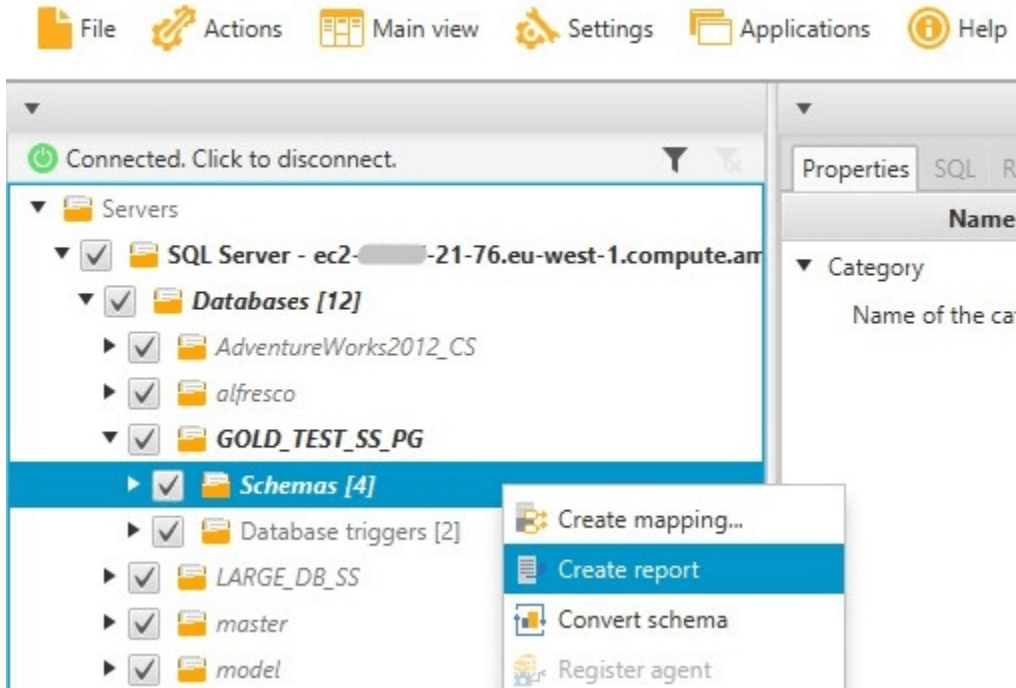
AWS Schema Conversion Tool 會建立資料庫遷移評定報告來協助您轉換結構描述。資料庫移轉評估報表提供有關將結構描述從來源資料庫轉換為目標資料庫的重要資訊。此報告會彙總所有結構描述轉換工作，並詳細說明無法轉換為目標資料庫資料庫引擎之結構描述的行動項目。本報告亦包含將相同的程式碼寫入無法自動轉換的資料庫時，所需要進行的作業量估算。

建立資料庫移轉評估報告

請使用下列程序，建立資料庫遷移評估報告。

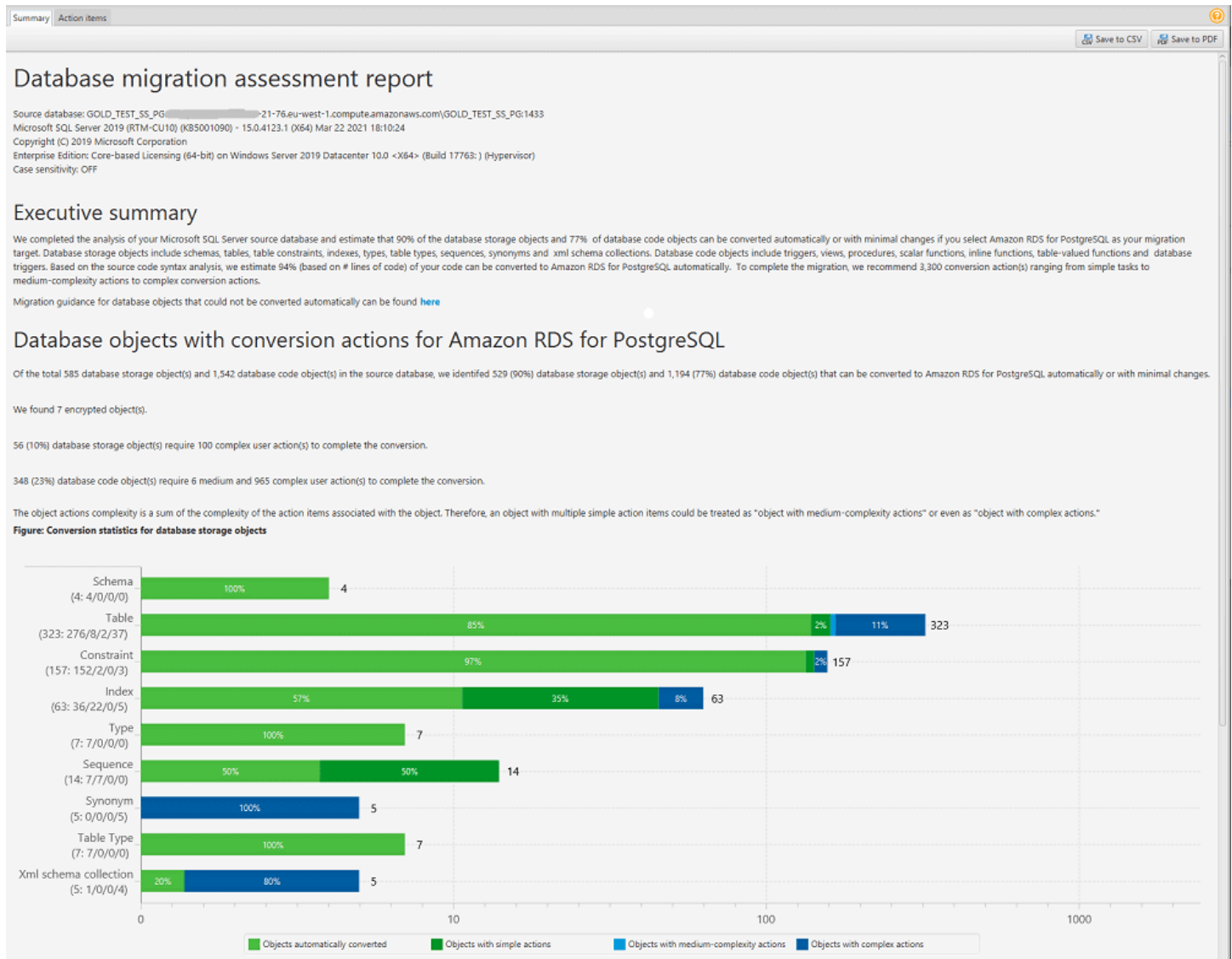
若要建立資料庫遷移評估報告

1. 在顯示來源資料庫結構描述的左側窗格中，選擇您要建立評估報告的結構描述物件。
2. 開啟該物件的內容 (按一下右鍵) 選單，然後選擇 **Create Report (建立報告)**。



評估報告摘要

當您建立評估報告後，評估報告檢視會開啟，顯示 Summary (摘要) 索引標籤。Summary (摘要) 索引標籤會顯示資料庫遷移評估報告的摘要資訊。它會顯示已自動轉換的項目，以及未自動轉換的項目。



針對無法自動轉換至目標資料庫引擎的結構描述項目，此摘要包括在您的目標資料庫執行個體中建立與來源相同的結構描述項目時，所需要進行的作業量估算。

此報告將轉換上述結構描述項目的估算時間分類如下：

- 簡單 — 可在一小時內完成的動作。
- 中 — 較複雜且可在一到四小時內完成的動作。
- 重要 — 非常複雜且需要四小時以上才能完成的動作。

評估報告行動項目

評估報告檢視也包含 Action Items (動作項目) 索引標籤。此索引標籤包含無法自動轉換為目標資料庫的資料庫引擎的項目清單。如果您在清單中選取動作項目，AWS SCT 會在結構描述中反白該動作項目套用的項目。

此報告也包含如何手動轉換結構描述項目的建議。如需決定如何處理手動轉換的詳細資訊，請參閱 [處理手動轉換 AWS SCT](#)。

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

- Summary / Action items:** Shows a tree view of the source database structure, including servers, databases, tables, and views.
- Issues List:** A list of conversion issues with their recommended actions and occurrence counts.
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table.
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes.
 - Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary.
 - Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable.
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy.
 - Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.
 - Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion.
 - Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function.
- SQL Code:** A preview of the generated SQL code for a stored procedure:


```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint unsigned
3   , @posFlags bigint - 0
4   , @posFlagsMask bigint - 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask) | @posFlags
8 from Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0
      
```
- Target Amazon RDS for MySQL category: Schemas:** A table showing the target schema details.

Name	Value
Category	Schemas
Name of the category	Schemas

儲存評估報告

您可以將資料庫遷移評估報告儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案形式的本機複本。CSV 檔案只包含動作項目資訊。PDF 檔案同時包含摘要和動作項目資訊，如以下範例所示。

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

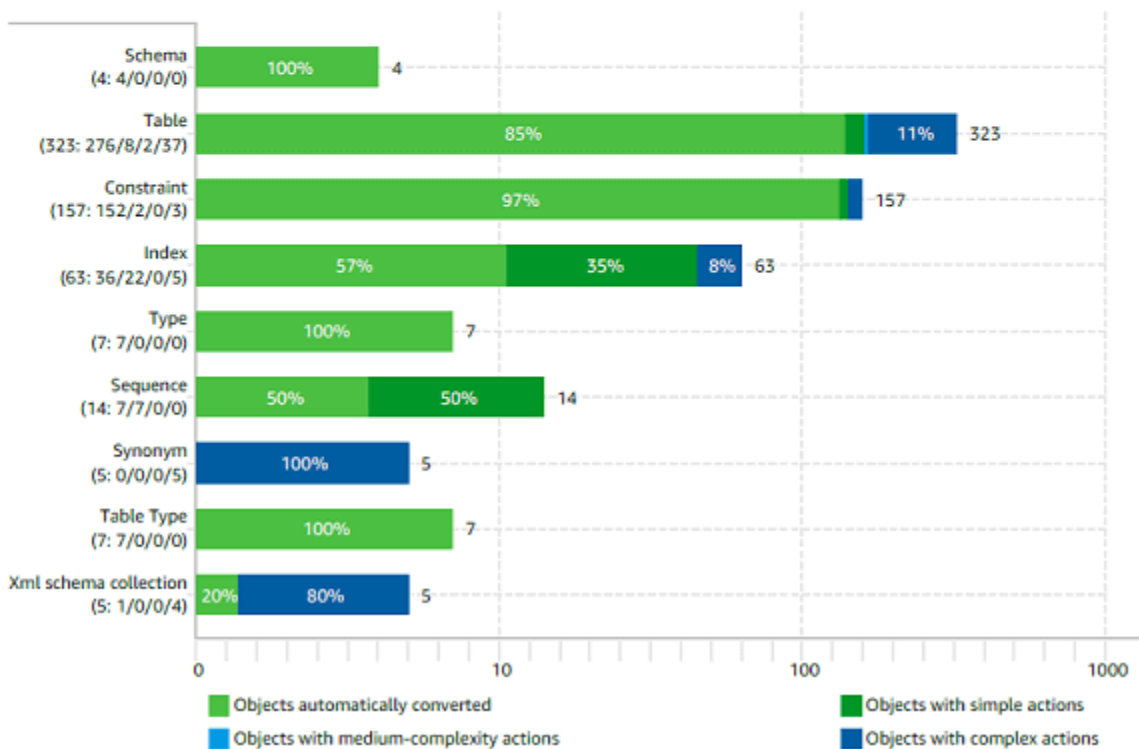
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



處理手動轉換 AWS SCT

評估報告包含無法自動轉換到目標資料庫之資料庫引擎的項目清單。對於每個無法轉換的項目，Action Items (動作項目) 索引標籤上都有一個動作項目。

您可以用下列方式回應評估報告中的動作項目：

- 修改您的來源資料庫結構描述。
- 修改您的目標資料庫結構描述。

修改來源結構描述

對於某些項目，將來源資料庫中的資料庫結構描述修改為可以自動轉換的結構描述，可能會比較輕鬆。首先，驗證新的變更與您的應用程式架構相容，然後更新來源資料庫中的結構描述。最後，使用更新後的結構描述資訊來重新整理您的專案。您可以轉換已更新的結構描述，並產生新的資料庫遷移評估報告。已在來源結構描述中變更的項目，不會再顯示動作項目。

此程序的優點是當您從來源資料庫重新整理時，已更新的結構描述永遠可用。

修改目標結構描述

對於某些項目，將轉換後的結構描述套用到您的目標資料庫，然後針對無法自動轉換的項目，手動新增同等結構描述項目到目標資料庫，可能會比較輕鬆。您可以透過套用結構描述，寫入所有可自動轉換至目標資料庫的結構描述。如需詳細資訊，請參閱[儲存並套用轉換後的結構描述 AWS SCT](#)。

寫入至目標資料庫的結構描述不包含無法自動轉換的項目。將結構描述套用到目標資料庫後，接著您可以在目標資料庫中手動建立等同於來源資料庫中項目的結構描述。資料庫遷移評估報告中的動作項目包含如何建立相同結構描述的建議。

Warning

如果您在目標資料庫中手動建立結構描述，請儲存一份手動作業的複本。如果您再次從專案套用轉換後的結構描述到目標資料庫，它會覆寫您已完成的手動作業。

在部分案例中，您無法在目標資料庫中建立相同的結構描述。您可能必須重新架構部分的應用程式與資料庫，以使用引擎為您的目標資料庫提供的功能。在其他案例中，您可以直接忽略無法自動轉換的結構描述。

更新並重新整理轉換後的結構描述 AWS SCT

您可以同時更新 AWS Schema Conversion Tool 專案中的來源結構描述和目標結構描述。

- 來源 — 如果您更新來源資料庫的結構描述，請使用來源資料庫中的最新結構描述AWS SCT取代專案中的結構描述。使用此功能，如果已對來源資料庫的結構描述進行了變更，您可以更新您的專案。
- 目標 — 如果您更新目標資料庫的結構描述，請以目標資料庫中的最新結構描述AWS SCT取代專案中的結構描述。如果您尚未套用任何結構描述到您的目標資料庫，AWS SCT 會從您的專案清除轉換後的結構描述。然後，您可以從來源資料庫轉換結構描述以用於全新的目標資料庫。

您可以選擇「從資料庫重新整理」來更新AWS SCT專案中的結構描述。

儲存並套用轉換後的結構描述 AWS SCT

當 AWS Schema Conversion Tool 產生轉換後的結構描述 (如 [使用轉換結構描述 AWS SCT](#) 所示) 時，它並不會立即將轉換後的結構描述套用到目標資料庫。反之，轉換後的結構描述會儲存於本機專案中，直到您準備將其套用至目標資料庫。利用此功能，您可以處理無法自動轉換到目標資料庫引擎的結構描述項目。如需無法自動轉換的項目的詳細資訊，請參閱[建立移轉評估報告AWS SCT](#)。

您可以選擇讓工具將轉換後的結構描述儲存為 SQL 指令碼檔案，之後再套用結構描述到目標資料庫。您也可以讓工具直接將轉換後的結構描述套用到目標資料庫。

將轉換後的結構定義儲存為檔案

您可以將轉換後的結構描述儲存為文字檔案中的 SQL 指令碼。使用此方法，您可以修改從 AWS SCT 產生的 SQL 指令碼，來解決工具無法自動轉換的項目。然後，您可以在目標資料庫執行個體上執行更新後的指令碼，以將轉換後的結構描述套用到目標資料庫。

若要將轉換的結構描述儲存為 SQL 命令檔

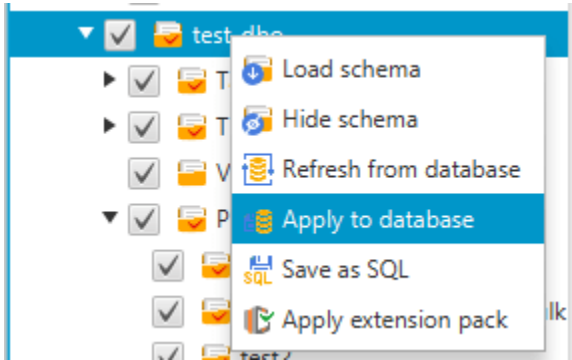
1. 選擇您的架構並開啟內容 (按一下滑鼠右鍵) 功能表。
2. 選擇「另存為 SQL」。
3. 輸入檔案名稱，然後選擇 [儲存]。
4. 使用下列其中一個選項儲存轉換後的結構描述：
 - 單一檔案
 - 每個階段單一檔案
 - 每個陳述式單一檔案

若要選擇 SQL 命令檔的格式

1. 在 [設定] 功能表上，選擇 [專案設定]。
2. 選擇儲存指令碼。
3. 對於供應商，選擇數據庫平台。
4. 在「儲存 SQL 命令檔至」中，選擇儲存資料庫結構描述指令碼的方式。
5. 選擇「確定」以儲存設定。

套用轉換的結構定義

當您準備好將已轉換的結構描述套用至目標資料庫時，請從您專案的右側窗格選擇結構描述元素。開啟結構描述元素的內容 (按一下右鍵) 選單，然後選擇 **Apply to database** (套用至資料庫)，如下所示。



擴充套件架構

第一次套用轉換後的結構描述到目標資料庫執行個體時，AWS SCT 會新增額外的結構描述到您的目標資料庫執行個體。此結構描述會實作來源資料庫的系統功能，當您將轉換後結構描述寫入至目標資料庫執行個體時需要這些功能。此結構描述稱為延伸套件結構描述。

請勿修改延伸套件結構描述，否則寫入至目標資料庫執行個體的轉換後結構描述會發生未預期的結果。當您的結構描述已完全遷移至目標資料庫執行個體，而且您不再需要 AWS SCT 時，即可刪除延伸套件結構描述。

延伸套件結構描述的命名是根據您的來源資料庫，如下所示：

- Greenplum : aws_greenplum_ext
- Microsoft SQL Server: aws_sqlserver_ext
- Netezza : aws_netezza_ext
- Oracle: aws_oracle_ext
- 雪花 : aws_snowflake_ext
- Teradata : aws_teradata_ext
- Vertica : aws_vertica_ext

如需詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

Python 程式庫

要在亞馬遜紅移中創建自定義函數，您可以使用 Python 語言。使用AWS SCT擴充套件為您的亞馬遜紅移資料庫安裝 Python 程式庫。如需詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

通過使用優化亞馬遜紅移 AWS SCT

您可以使用AWS Schema Conversion Tool來優化您的亞馬遜紅移資料庫。使用您的 Amazon Redshift 資料庫做為來源，並將 Amazon Redshift 資料庫測試為目標，AWS SCT建議對金鑰和分發金鑰進行排序以最佳化資料庫。

優化您的亞馬遜紅移數據庫

請使用下列程序來最佳化您的 Amazon 紅移資料庫。

優化您的亞馬遜紅移數據庫

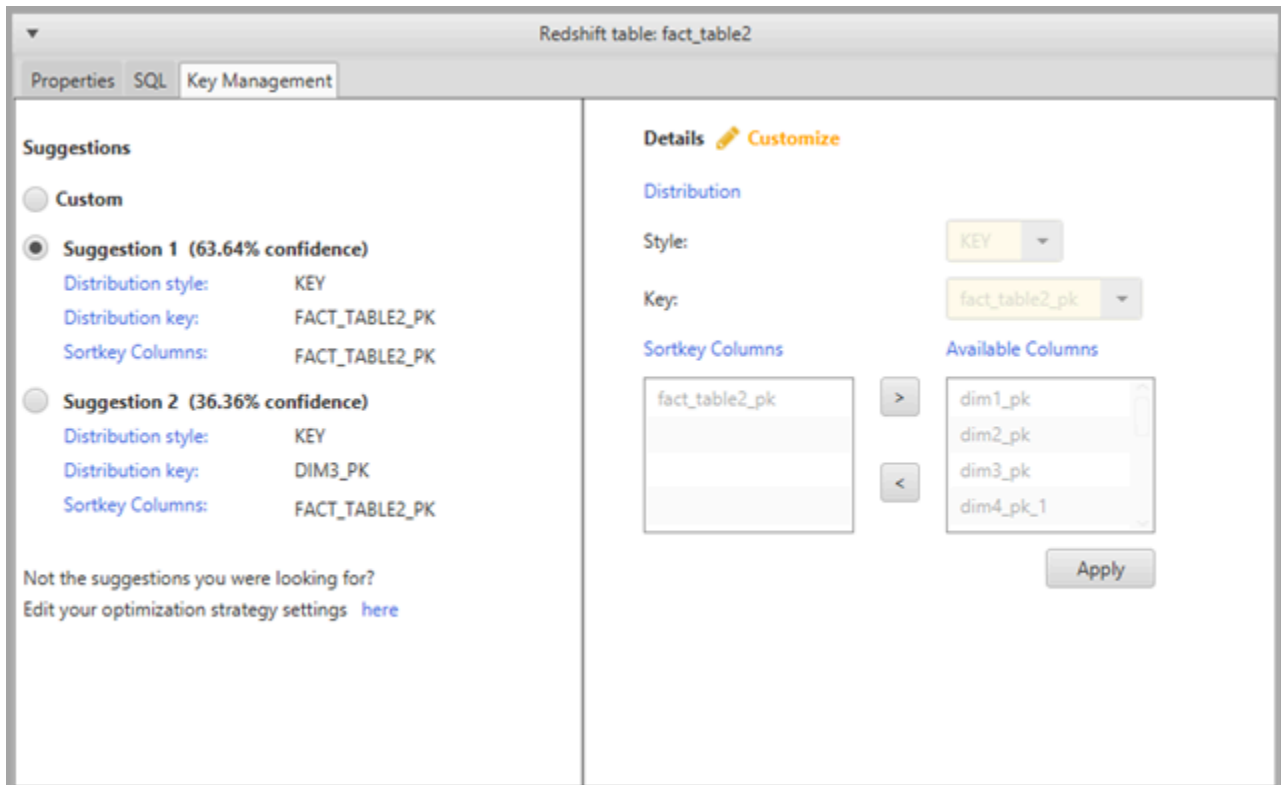
1. 拍攝您的 Amazon 紅移叢集的手動快照做為備份。您可以在完成 Amazon Redshift 叢集最佳化並測試您所做的任何變更之後刪除快照。如需詳細資訊，請參閱[Amazon 紅移快照](#)。
2. 從專案左側面板選擇要轉換的結構描述物件。開啟該物件的內容 (按一下右鍵) 選單，然後選擇 Collect Statistics (收集統計資料)。

AWS SCT 會使用統計資料來提供排序索引鍵和散發索引鍵的建議。

3. 從專案左側面板選擇要最佳化的結構描述物件。開啟該物件的內容 (按一下右鍵) 選單，然後選擇 Run Optimization (執行最佳化)。

AWS SCT 會提供排序索引鍵和散發索引鍵的建議。

4. 若要檢閱建議，請在專案左側面板中展開結構描述下方的表格節點，然後選擇表格。選擇 Key Management (索引鍵管理) 索引標籤，如下所示。



左側窗格包含索引鍵建議，也包含每個建議的信心度。您可以選擇其中一個建議，也可以在右側窗格中編輯以自訂索引鍵。

5. 您可以建立包含最佳化建議的報告。若要建立報告，請執行下列步驟：
 - a. 從專案左側面板選擇您已最佳化的結構描述物件。開啟該物件的內容 (按一下右鍵) 選單，然後選擇 Create Report (建立報告)。

報告會在主視窗中開啟，並顯示 Summary (摘要) 索引標籤。報告中會顯示具有最佳化建議的物件數量。
 - b. 選擇 Action Items (動作項目) 索引標籤，在報告格式中查看索引鍵建議。
 - c. 您可以將最佳化報告儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案形式的本機複本。CSV 檔案只包含動作項目資訊。PDF 檔案同時包含摘要和動作項目資訊。
6. 若要將建議的最佳化套用到您的資料庫，請在專案的右側窗格中選擇物件。開啟該物件的內容 (按一下右鍵) 選單，然後選擇 Apply to database (套用至資料庫)。

使用以下方式轉換擷取、轉換和載入 (ETL) 程序AWS Schema Conversion Tool

您可以使用AWS Schema Conversion Tool(AWS SCT) 以移轉擷取、轉換和載入 (ETL) 程序。這種類型的遷移包括 ETL 相關業務邏輯的轉換。此邏輯可以位於來源資料倉儲內，也可以位於您個別執行的外部指令碼中。

目前，AWS SCT支援將 ETL 指令碼轉換為物件AWS Glue和亞馬遜紅移 RSQL，如下表所示。

來源	Target
ETL 信息腳本	信息
微軟 SQL 服務器集成服務 (SSIS) ETL 軟件包	AWS Glue 或 AWS Glue Studio
包含來自 Teradata 基本資料查詢 (BTEQ) 內嵌指令的殼層指令碼	亞馬遜紅移 RSQL
太數據 BTEQ ETL 腳本	AWS Glue或亞馬遜紅移 RSQL
太元數據FastExport工作指令碼	亞馬遜紅移 RSQL
太元數據FastLoad工作指令碼	亞馬遜紅移 RSQL
太元數據MultiLoad工作指令碼	亞馬遜紅移 RSQL

主題

- [將 ETL 程序轉換為AWS Glue與AWS SCT](#)
- [使用 Python API 轉換 ETL 程序AWS Glue與AWS SCT](#)
- [轉換信息 ETL 腳本AWS SCT](#)
- [將 SSIS 轉換為AWS Glue與AWS SCT](#)
- [將 SSIS 轉換為AWS Glue Studio與AWS SCT](#)
- [使用以下方式將 Teradata 指令碼轉換為亞馬遜紅移 RSQLAWS SCT](#)
- [將具有嵌入式太拉數據 BTEQ 命令的殼層腳本轉換為亞馬遜紅移 RSQLAWS SCT](#)
- [轉換太拉數據FastExport亞馬遜紅移 RSQL 的任務腳本AWS SCT](#)

- [轉換太拉數據FastLoad亞馬遜紅移 RSQL 的任務腳本AWS SCT](#)
- [轉換太拉數據MultiLoad亞馬遜紅移 RSQL 的任務腳本AWS SCT](#)

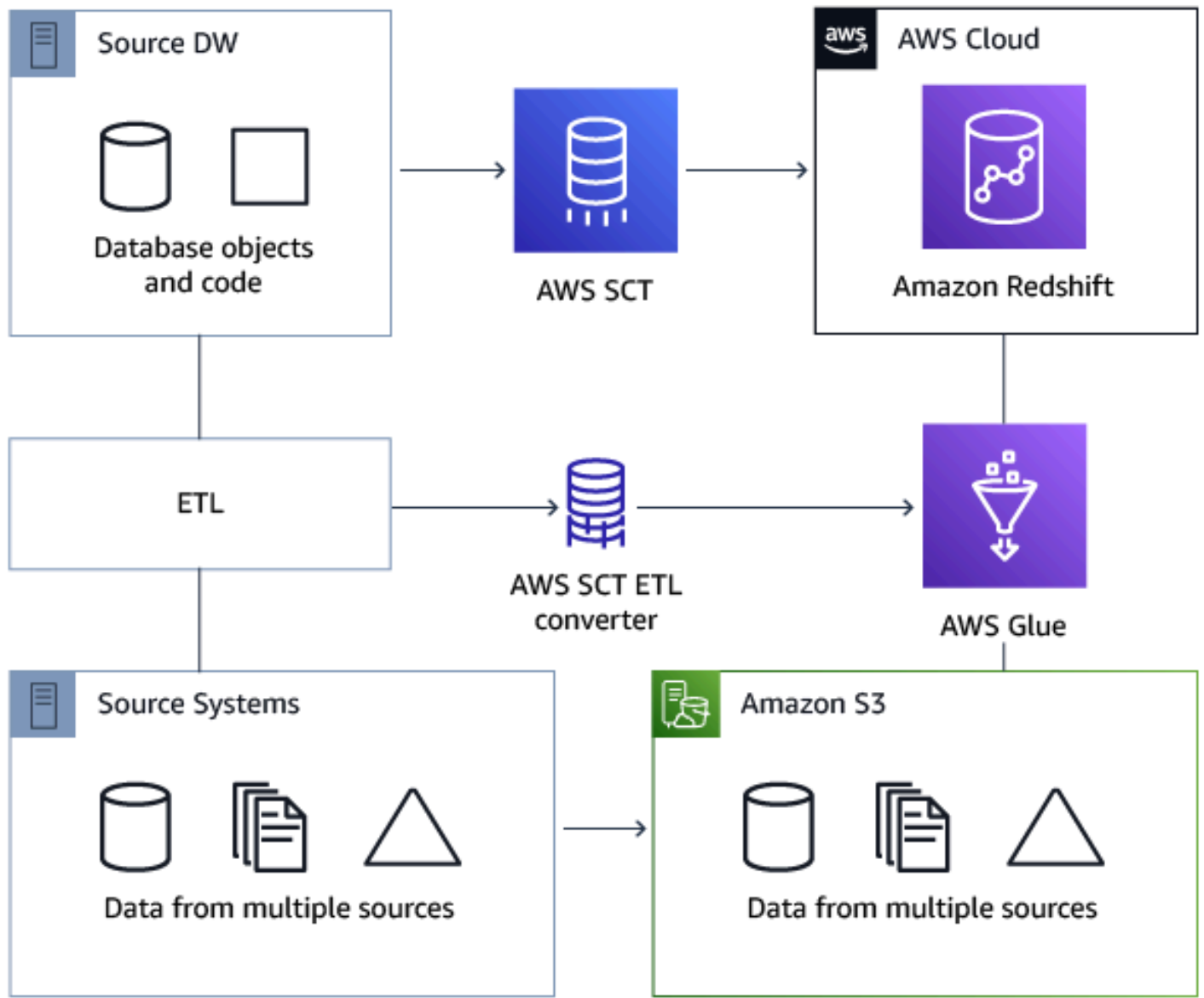
將 ETL 程序轉換為AWS Glue與AWS SCT

接下來，您可以找到將 ETL 腳本轉換為過程的概述AWS Glue與AWS SCT。在此範例中，我們將 Oracle 資料庫轉換為 Amazon Redshift，以及與來源資料庫和資料倉儲搭配使用的 ETL 程序。

主題

- [先決條件](#)
- [了解 AWS Glue 資料目錄](#)
- [使用轉換的限制AWS SCT與AWS Glue](#)
- [步驟 1：建立新的 專案](#)
- [步驟 2：建立AWS Glue工作](#)

下列架構圖顯示範例資料庫移轉專案，其中包含 ETL 指令碼轉換為AWS Glue。



先決條件

開始之前，請執行以下動作：

- 移轉您要移轉至的任何來源資料庫AWS。
- 將目標資料倉儲移轉至AWS。
- 收集 ETL 程序中所有代碼的清單。
- 收集每個資料庫所有必要連線資訊的清單。

還有AWS Glue需要權限才能訪問其他AWS代表您的資源。您可用 AWS Identity and Access Management (IAM) 來提供這些許可。請確定您已建立的 IAM 政策AWS Glue。如需詳細資訊，請參閱[為以下項目建立 IAM 政策AWS糖膠服務](#)在AWS Glue開發者指南。

了解 AWS Glue 資料目錄

在轉換過程中，AWS Glue 會載入來源資料庫和目標資料庫的相關資訊。它將此信息組織成類別，在稱為 a 的結構中樹。結構包含以下項目：

- 連接— 連接參數
- 爬行-爬蟲列表，每個模式一個爬蟲
- 資料庫— 保存表格的容器
- 表格— 表示表格中資料的中繼資料定義
- ETL 工作— 執行 ETL 工作的業務邏輯
- 觸發— 控制 ETL 工作何時執行的邏輯AWS Glue(無論是按需求、按排程還是由工作事件觸發)

AWS Glue 資料目錄是資料的位置、結構描述及執行時間指標的索引。當您使用 AWS Glue 和 AWS SCT，AWS Glue 資料目錄包含對在 AWS Glue 中作為 ETL 任務之來源和目標的資料參考。若要建立資料倉儲，請編目此資料。

您可以使用資料目錄中的資訊來建立和監控 ETL 任務。一般而言，您會執行爬蟲程式來清點資料存放區中的庫存，但仍有其他方式可將中繼資料資料表新增到資料目錄。

當您在資料目錄中定義資料表時，就會將資料表加入資料庫。資料庫是用來彙整 AWS Glue 中的資料表。

使用轉換的限制AWS SCT與AWS Glue

以下限制適用於使用 AWS SCT 與 AWS Glue 的轉換。

Resource	預設限制
每個帳戶的資料庫數量	10,000
每個資料庫的資料表數量	100,000
每個資料表的分割區數量	1,000,000

每個資料表的資料表版本數量	100,000
每個帳戶的資料表數量	1,000,000
每個帳戶的分割區數量	10,000,000
每個帳戶的資料表版本數量	1,000,000
每個帳戶的連線數量	1,000
每個帳戶的爬蟲程式數量	25
每個帳戶的任務數量	25
每個帳戶的觸發數量	25
每個帳戶的並行執行任務數量	30
每個任務的並行執行任務數量	3
每個觸發的任務數量	10
每個帳戶的開發端點數量	5
開發端點一次使用的最大資料處理單元 (DPU)	5
角色一次所能使用的 DPU 數量上限	100
資料庫名稱長度	<p>無限制</p> <p>為了相容於其他中繼資料存放 (例如 Apache Hive), 名稱會變為使用小寫字元。</p> <p>如果您打算從 Amazon Athena 存取資料庫, 請提供僅包含英數字元和底線字元的名稱。</p>
連線名稱長度	無限制
爬蟲程式名稱長度	無限制

步驟 1：建立新的 專案

若要建立新專案，請執行下列高階步驟：

1. 在 AWS SCT 中建立新專案。如需詳細資訊，請參閱[建立AWS SCT專案](#)。
2. 將您的來源和目標資料庫新增至專案。如需詳細資訊，請參閱[將資料庫伺服器新增至AWS SCT專案](#)。

確保您已選擇使用AWS Glue在目標資料庫連線設定中。若要這麼做，請選擇AWS Glue標籤。對於複製來源AWS個人資料」下方，選擇您要使用的描述檔。設定檔應自動填入AWS存取金鑰、秘密金鑰和 Amazon S3 儲存貯體資料夾。若未自動填入，請手動輸入此資訊。選擇 OK (確定) 後，AWS Glue 會分析物件，並將中繼資料載入到 AWS Glue 資料目錄。

根據您的安全設定，您可能看到警告訊息，指出您的帳戶對於伺服器上的某些結構描述沒有足夠權限。若您可以存取正在使用的結構描述，則可以放心地忽略此訊息。

3. 若要完成匯入 ETL 的準備，請連線至來源和目標資料庫。若要這麼做，請在來源或目標中繼資料樹狀結構中選擇您的資料庫，然後選擇連接到服務器。

AWS Glue在來源資料庫伺服器上建立資料庫，並在目標資料庫伺服器上建立一個資料庫，以協助進行 ETL 轉換。目標伺服器上的資料庫包含 AWS Glue 資料目錄。若要尋找特定物件，請使用來源面板或目標面板上的搜尋。

若要查看特定物件的轉換方式，請尋找您要轉換的項目，然後選擇轉換結構從其上下文 (右鍵單擊) 菜單。AWS SCT將此選取的物件轉換為腳本。

您可以查看轉換後的腳本腳本右側面板中的文件夾。目前，該腳本是一個虛擬對象，只能作為您的一部分使用AWS SCT項目。

若要建立AWS Glue使用轉換後的指令碼進行任務，將您的指令碼上傳到 Amazon S3。若要將指令碼上傳到 Amazon S3，請選擇指令碼，然後選擇儲存到 S3從其上下文 (右鍵單擊) 菜單。

步驟 2：建立AWS Glue工作

將指令碼儲存到 Amazon S3 之後，您可以選擇它，然後選擇配置AWS Glue工作以開啟精靈以設定AWS Glue工作。該嚮導可以更輕鬆地進行設置：

1. 在精靈的第一個索引標籤上，設計資料流，您可以選擇執行策略以及要包含在此工作中的指令碼清單。您可以為每個指令碼選擇參數。您也可以重新安排指令碼，讓它們以正確的順序執行。

2. 在第二個選項卡上，您可以命名您的工作，並直接配置設置AWS Glue。在此畫面上，您可以進行下列設定：
 - AWS Identity and Access Management(IAM) 角色
 - 指令碼檔案名稱和檔案路徑
 - 使用 Amazon S3 受管金鑰使用伺服器端加密指令碼加密 (SSE-S3)
 - 暫時目錄
 - 產生的 Python 程式庫路徑
 - 使用者 Python 程式庫路徑
 - 相依 .jar 檔案的路徑
 - 參考檔案路徑
 - 每個任務執行的並行 DPU 數
 - 最大並行數量
 - 任務逾時 (以分鐘為單位)
 - 延遲通知閾值 (以分鐘為單位)
 - 重試次數
 - 安全組態
 - 伺服器端加密
3. 在第三個步驟或標籤中，您可以選擇已設定的目標端點連線。

完成任務的設定後，任務會顯示在 AWS Glue 資料目錄的 ETL 任務下。如果您選擇任務，系統會顯示設定，以便您檢閱或編輯任務。若要在中建立新工作AWS Glue，選擇創建AWS Glue工作從工作的前後關聯 (按一下滑鼠右鍵) 功能表。這麼做適用於結構描述定義。若要重新整理顯示，請選擇內容 (按一下滑鼠右鍵) 功能表中的 Refresh from database (從資料庫重新整理)。

此時，您可以在 AWS Glue 主控台中查看您的任務。若要這麼做，請登入AWS Management Console 並打開AWS Glue控制台<https://console.aws.amazon.com/glue/>。

您可以測試新任務是否能正確運作。若要這麼做，請先檢查來源資料表中的資料，並驗證目標資料表是否為空。執行任務，然後再檢查一次。您可以從 AWS Glue 主控台檢視錯誤日誌。

使用 Python API 轉換 ETL 程序AWS Glue與AWS SCT

在以下區段，您就可以找到轉換描述，該轉換是使用 Python 來呼叫 AWS Glue API。如需詳細資訊，請參閱[程式AWS Glue蟒蛇中的 ETL 腳本](#)在 AWS Glue開發人員指南。

主題

- [步驟 1：建立資料庫](#)
- [步驟 2：建立連線](#)
- [步驟 3：建立AWS Glue爬行者](#)

步驟 1：建立資料庫

第一步是在一個新的數據庫中創建一個新的數據庫AWS Glue使用「資料目錄」[AWSSDK API](#)。當您在資料目錄中定義資料表時，就會將資料表加入資料庫。資料庫是用來彙整 AWS Glue 中的資料表。

以下範例示範 AWS Glue 適用之 Python Glue API 的 `create_database` 方法。

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

如果您使用的是亞馬遜紅移，則數據庫名稱形成如下。

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

本範例中，亞馬遜紅移叢集的完整名稱如下。

```
rsddb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

以下顯示正確格式的資料庫名稱範例。在這種情況下，`rsddb03` 是名稱，這是叢集端點完整名稱的第一部分。資料庫名為 `dev` 且結構描述為 `ora_glue`。

```
rsddb03_dev_ora_glue
```

步驟 2：建立連線

使用在資料目錄中建立新連線[AWSSDK API](#)。

下面的例子演示了如何使用 `create_connection` 對於 Python API 的方法 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'  
            ]  
        }  
    }  
)
```

`create_connection` 中使用的參數如下：

- Name(UTF-8 字串) — 必須填寫。對於亞馬遜紅移，連接名稱形成如下：`Redshift_<Endpoint-name>_<redshift-database-name>`，例如：`Redshift_abcde03_dev`
- Description(UTF-8 字串) — 您對連線的描述。
- ConnectionType(UTF-8 字串) — 必要項目。連接的類型。目前，只支援 JDBC，不支援 SFTP。
- ConnectionProperties (字典) -必需的。用作此連線參數的索引鍵值配對清單，包括 JDBC 連線 URL、使用者名稱和密碼。
- PhysicalConnectionRequirements (dict) -物理連接要求，其中包括以下內容：
 - SubnetId(UTF-8 字串) — 連線所使用之子網路的識別碼。
 - SecurityGroupIdList(清單) — 連線所使用的安全性群組 ID 清單。
 - AvailabilityZone(UTF-8 字串) — 必要項目。包含端點的可用區域。此參數已棄用。

步驟 3：建立AWS Glue爬行者

接著，您建立 AWS Glue 爬蟲程式來填入 AWS Glue 目錄。如需詳細資訊，請參閱[使用爬行者程式編目表格](#)在AWS Glue開發人員指南。

新增爬行者程式的第一個步驟是在「資料目錄」中使用[AWSSDK API](#)。在開始之前，請務必先刪除任何先前版本，方法是使用delete_crawler操作。

建立爬蟲程式時，請納入幾個考量：

- 對於爬蟲程式名稱，請使用格式 `<redshift_node_name>_<redshift_database_name>_<redshift_shema_name>`，例如：abcde03_dev_ora_glue
- 使用已存在的 IAM 角色。如需建立 IAM 角色的詳細資訊，請參閱[建立 IAM 角色](#)在IAM 使用者指南。
- 使用您在上一個步驟建立的資料庫名稱。
- 使用 ConnectionName 參數 (此為必要)。
- 對於 path 參數，請使用 JDBC 目標的路徑，例如：dev/ora_glue/%

以下範例會刪除現有的爬蟲程式，然後使用 AWS Glue 適用的 Python API 建立新的爬蟲程式。

```
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'
)

```

建立連接到一或多個資料存放區的爬蟲程式並加以執行，判斷資料結構，然後將資料表寫入資料目錄。您可以依排程執行您的爬蟲程式，如下所示。

```

response = client.start_crawler(
    Name='string'
)

```

此範例使用亞馬遜紅移作為目標。亞馬遜紅移數據類型映射到AWS Glue搜索器運行後，以下面的方式數據類型。

亞馬遜紅移數據類型	AWS Glue 資料類型
smallint	smallint
integer	int
bigint	bigint
decimal	decimal(18,0)

decimal(p,s)	decimal(p,s)
real	double
double precision	double
布林值	布林值
char	string
varchar	string
varchar(n)	string
日期	日期
timestamp	timestamp
timestampz	timestamp

轉換信息 ETL 腳本AWS SCT

您可以使用AWS SCT命令行界面 (CLI) 轉換您的信息 ETL 腳本，以便您可以將這些腳本與新的目標數據庫一起使用。此轉換包括三個關鍵步驟。首先，AWS SCT轉換嵌入在您的信息對象的 SQL 代碼。下一步AWS SCT根據您在專案中指定的移轉規則變更資料庫物件的名稱。最後，AWS SCT將您的信息 ETL 腳本的連接重定向到新的目標數據庫。

您可以轉換信息 ETL 腳本作為的一部分AWS SCT資料庫轉換專案。當您轉換 Information ETL 指令碼時，請務必將來源和目標資料庫新增至專案。

要轉換信息 ETL 腳本，請確保您使用AWS SCT版本 1.0.667 或更高版本。另外，熟悉的命令行界面AWS SCT。如需詳細資訊，請參閱[AWS SCT CLI 參考資料](#)。

若要使用轉換資訊 ETL 指令碼AWS SCT

1. 創建一個新的AWS SCTCLI 指令碼或編輯現有案例範本。例如，您可以下載並編輯InformaticConversionTemplate.scts範本。如需詳細資訊，請參閱[取得 CLI 案例](#)。
2. 下載來源和目標資料庫所需的 JDBC 驅動程式。使用指定這些驅動程式的位置SetGlobalSettings指令。另外，指定文件夾AWS SCT可以保存日誌文件。

下列程式碼範例示範如何將路徑新增至甲骨文和 PostgreSQL 驅動程式至AWS SCT設定。執行此程式碼範例之後，AWS SCT將記錄檔儲存在C:\sct_log資料夾。還有AWS SCT將控制台日誌文件存儲在C:\Temp\oracle_postgresql資料夾。

```
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. 創建一個新的AWS SCT項目。輸入專案的名稱和位置。

下列程式碼範例會建立oracle_postgresql專案中的C:\Temp資料夾。

```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. 新增有關來源和目標資料庫的連線資訊。

下列程式碼範例會將甲骨文和 PostgreSQL 資料庫新增為您的資料庫的來源和目標AWS SCT項目。

```
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'
/
```

在前面的例子中，替換#####和#####使用您的數據庫用戶的名稱。接下來，更換#####和#####使用您的密碼。對於##### (_C)和#####，輸入來源和目標資料庫伺服器的 IP 位址。

若要連線到 Oracle 資料庫版本 19 及更新版本，請使用AddSource指令。若要這麼做，請新增-connectionType參數並將其值設置為'basic_service_name'。然後，添加-servicename參數並將其值設置為您的 Oracle 服務名稱。有關的更多信息AddSource指令，請參閱[AWS Schema Conversion Tool CLI 參考資料](#)。

5. 創建一個新的AWS SCT對映規則，定義每個來源資料庫綱要的目標資料庫引擎。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

下列程式碼範例會建立包含所有來源 Oracle 資料庫結構描述的對應規則，並將 PostgreSQL 定義為移轉目標。

```
AddServerMapping
-sourceTreePath: 'Servers.ORACLE'
-targetTreePath: 'Servers.POSTGRESQL'
/
```

6. 添加有關信息源和目標 XML 文件的連接信息。

下列程式碼範例會從C:\Informatica_source和C:\Informatica_target資料夾。

```
AddSource
-name: 'INFA_SOURCE'
-vendor: 'INFORMATICA'
-mappingsFolder: 'C:\Informatica_source'
/
AddTarget
-name: 'INFA_TARGET'
-vendor: 'INFORMATICA'
-mappingsFolder: 'C:\Informatica_target'
/
```

7. 建立另一個對應規則，以定義來源資訊 XML 檔案的目標資訊 XML 檔案。

下列程式碼範例會建立對應規則，其中包含前述範例中使用的來源和目標 Informatica XML 檔案。

```
AddServerMapping
-sourceTreePath: 'ETL.INFA_SOURCE'
-targetTreePath: 'ETL.INFA_TARGET'
/
```

8. 指定對應至 Information 連線名稱參照的資料庫伺服器連線。

下列程式碼範例會設定 Informatica ETL 指令碼從您的來源重新導向至新的目標資料庫。此範例也會設定連接變數。

```
ConfigureInformaticaConnectionsRedirect
-treePath: 'ETL.INFA_SOURCE.Files'
-connections: '{
"ConnectionNames": [
{
"name": "Oracle_src",
"newName": "postgres",
"treePath": "Servers.ORACLE"
}
]
"ConnectionVariables": [
{
"name": "$Source",
"treePath": "Servers.ORACLE"
}
]
}'
/
```

9. 轉換來源資料庫結構描述和資訊 ETL 指令碼。

下列程式碼範例會轉換所有來源 Oracle 資料庫結構描述和資訊 XML 檔案。

```
Convert
-treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
-treePath: 'ETL.INFA_SOURCE.Files'
```

```
/
```

10. (選擇性) 儲存轉換專案和評估報告。此報告包含轉換動作項目，以及如何處理每個動作項目的建議。

下列程式碼範例會儲存您的專案，並將評估報告的副本儲存為 PDF 檔案C:\Temp資料夾。

```
SaveProject
/  
SaveReportPDF
-treePath: 'ETL.INFA_SOURCE.Files'  
-file: 'C:\Temp\Informatica.pdf'  
/
```

11. 保存轉換後的 XML 信息文件。

下列程式碼範例會將轉換的 XML 檔案儲存在C:\Temp資料夾。您在上一個步驟中使用AddTarget指令。

```
SaveTargetInformaticaXML
-treePath: 'ETL.INFA_TARGET.Files'  
/
```

12. 將您的腳本另存為.scts文件並使用RunSCTBatch中的指令AWS SCTCLI。如需詳細資訊，請參閱[AWS SCT CLI 指令碼模式](#)。

下列範例會執行Informatica.scts中的指令碼C:\Temp資料夾。您可以在視窗中使用這個範例。

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

如果您編輯源信息 ETL 腳本，然後運行AWS SCT再次使用 CLI 指令碼。

將 SSIS 轉換為AWS Glue與AWS SCT

接下來，你可以找到如何轉換微軟 SQL 服務器集成服務 (SSIS) 包AWS Glue運用AWS SCT。

若要將微軟 SSIS 套件轉換為AWS Glue，請確定您使用AWS SCT版本 1.0.642 或更高版本。您還需要有一個包含 ETL 軟件包的 SSIS 項目-.dtsx,.conmgr，以及.params本機資料夾中的檔案。

您不需要已安裝的 SSIS 伺服器。轉換程序會經過本機 SSIS 檔案。

若要將 SSIS 套件轉換為AWS Glue運用AWS SCT

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從菜單中添加一個新的源 SSIS 包到您的項目。
3. 選擇SQL 伺服器整合服務並完成以下操作：
 - 連線名稱— 輸入連線名稱。AWS SCT在中繼資料樹狀結構中顯示此名稱。
 - SSIS 套件資料夾— 選擇包含 SSIS 項目文件夾的路徑。

AWS SCT讀取專案檔案 (副檔名為的檔案).dtsx,.conmgr或者.params) 從本地文件夾中解析它們。然後它將它們組織成一個AWS SCT樹的類別。

4. 選擇新增目標從菜單中添加新的目標平台以轉換源 SSIS 包。
5. 選擇AWS Glue並完成以下操作：
 - 連線名稱— 輸入連線名稱。AWS SCT在中繼資料樹狀結構中顯示此名稱。
 - 複製來源AWS個人資料— 選擇要使用的設定檔。
 - AWS存取金鑰— 輸入您的AWS存取金鑰。
 - AWS秘密金鑰— 輸入您的AWS秘密金鑰。
 - 地區— 選擇AWS 區域您要從清單中使用的。
 - 亞馬遜 S3 桶文件夾— 輸入您計劃使用的 Amazon S3 儲存貯體的資料夾路徑。

您可以使用虛擬AWS Glue目標。在此情況下，您不需要指定連線認證。如需詳細資訊，請參閱[the section called “虛擬目標”](#)。

6. 建立新的對應規則，其中包含您的來源 SSIS 套件和AWS Glue目標。如需詳細資訊，請參閱[the section called “新規則”](#)。
7. 在「」檢視功能表中，選擇主視圖。
8. 在 SSIS 樹狀檢視中，開啟內容 (按一下滑鼠右鍵) 功能表連線管理員，然後選擇設定連線。
9. 設定專案連接管理員。

若要設定 SSIS 連接管理員的連線對應，請指定AWS Glue對應 SSIS 連接管理員的連接。確保您的AWS Glue已建立連線。

- a. 下連接，選擇專案連接。

- b. 對於膠水目錄連接，選擇合適的AWS Glue連接。
10. 設定套件連線管理員：
 - a. 下連接，選擇您的包裹。
 - b. 對於膠水目錄連接，選擇合適的AWS Glue連接。
 - c. 針對封裝可用的所有連線重複這些動作。
 11. 選擇 Apply (套用)。
 12. 轉換您的包裹。在原始碼樹狀檢視中，尋找套餐。開啟套件的內容 (按一下滑鼠右鍵) 選單，然後選擇轉換套件。
 13. 將轉換後的腳本保存到亞馬遜 S3。在目標樹狀檢視中，尋找套件指令碼。開啟已轉換指令碼的內容 (按一下滑鼠右鍵) 選單，然後選擇儲存到 S3。
 14. 配置您的AWS Glue工作。在目標樹狀檢視中，尋找套件指令碼。開啟已轉換指令碼的內容 (按一下滑鼠右鍵) 選單，然後選擇配置AWS Glue工作。
 15. 完成三個設定區段：
 - a. 完成設計資料流區段：
 - 執行策略— 選擇工作執行 ETL 指令碼的方式。選擇連續的，以按照精靈中指定的順序執行指令碼。選擇平行以平行執行指令碼，忽略精靈中指定的順序。
 - 腳本— 選擇轉換後的腳本的名稱。
 - 選擇 下一步。
 - b. 完成工作屬性區段：
 - 姓名— 輸入您的姓名AWS Glue工作。
 - IAM 角色— 選擇 IAM 角色，該角色用於授權用於執行任務和存取資料存放區的資源。
 - 腳本檔案名稱— 輸入轉換後的腳本的名稱。
 - 指令碼檔案 S3 路徑— 輸入轉換後指令碼的 Amazon S3 路徑。
 - 使用 SSE-S3 加密腳本— 選擇此選項可使用 Amazon S3 受管加密金鑰 (SSE-S3) 使用伺服器端加密來保護資料。
 - 暫存目錄— 輸入 Amazon S3 到暫存目錄的路徑，以取得中繼結果。AWS Glue和AWS Glue內建轉換工具會使用此目錄來讀取或寫入亞馬遜 Redshift。
 - AWS SCT會自動產生 Python 程式庫的路徑。您可以在以下位置檢閱此路徑生成的蟒蛇庫路徑。您無法編輯此自動產生的路徑。若要使用其他 Python 程式庫，請在用戶蟒蛇庫路徑。

- 用戶蟒蛇庫路徑— 輸入其他使用者 Python 程式庫的路徑。請使用逗號分隔 Amazon S3 路徑。
 - 依賴罐子路徑— 輸入相依 jar 檔案的路徑。請使用逗號分隔 Amazon S3 路徑。
 - 參考檔案路徑— 輸入指令碼所需的其他檔案 (例如組態檔) 的路徑。請使用逗號分隔 Amazon S3 路徑。
 - 最大容量— 輸入的最大數量AWS Glue執行此工作時可配置的資料處理單元 (DPU)。您可以輸入 2 到 100 之間的整數。預設為 2。
 - 最大並發— 輸入此工作允許的並行執行數目上限。預設值為 1。AWS Glue達到此閾值時返回錯誤。
 - 工作逾時 (分鐘)— 輸入 ETL 工作的逾時值，以防止失控工作。批次任務的預設值為 2880 分鐘 (48 小時)。如果工作超過此限制，工作執行狀態會變更為TIMEOUT。
 - 延遲通知閾值 (分鐘) — 在幾分鐘之前輸入閾值AWS SCT傳送延遲通知。
 - 重試次數— 輸入的次數 (0—10)AWS Glue如果工作失敗，應該自動重新啟動作業。達到逾時限制的工作不會重新啟動。預設值為 0。
 - 選擇 下一步。
- c. 設定所需的連線：
- i. 從所有連接，選擇所需的AWS Glue連接並將其添加到列表選取的連線。
 - ii. 選擇 Finish (完成)。
16. 建立已設定的AWS Glue工作。在目標樹狀檢視中，尋找並展開ETL 工作。開啟您設定之 ETL 工作的內容 (按一下滑鼠右鍵) 功能表，然後選擇創建AWS Glue工作。
17. 運行AWS Glue工作:
- a. 打開AWS Glue控制台<https://console.aws.amazon.com/glue/>。
 - b. 在導覽窗格中，選擇 Jobs (任務)。
 - c. 選擇新增工作，然後選擇您要執行的工作。
 - d. 在「」動作」頁籤上，選擇執行工作。

SSIS 組件AWS SCT可以轉換為AWS Glue

您可以使用AWS SCT轉換資料流程並控制流程元件，以及容器、參數和變數。

支援的資料流程元件包括下列項目：

- ADO 淨目的地

- ADO 淨源
- Aggregate
- 快取轉換
- 字符映射轉換
- 條件式分割轉換
- 複製資料欄轉換
- 資料轉換轉換
- 衍生資料欄轉換
- 卓越目的地
- 卓越來源
- 匯出資料欄轉換
- 平面檔案目的地
- 平面檔案來源
- 模糊查找轉換
- 匯入資料欄轉換
- 查閱轉換
- 合併聯結轉換
- 合併轉換
- 多點傳送轉換
- 目的地
- 來源
- OLE DB 指令轉換
- OLE 資料庫目標
- OLE 資料庫來源
- 百分比取樣轉換
- 樞紐轉換
- 原始檔案目的地
- 原始檔案來源
- RecordSet目的地

- 資料列計數轉換
- 資料列取樣轉換
- 排序轉換
- SQL 伺服器目的地
- 聯集所有轉換
- 取消樞紐轉換
- XML 原始碼

支援的控制流程元件包括下列項目：

- 大量插入任務
- 執行套件任務
- 執行 SQL 任務
- 執行 T-SQL 陳述式任務
- 運算式工作
- 檔案系統工作
- 通知運算子任務
- 傳送郵件工作

支援的 SSIS 容器包括下列項目：

- 對於循環容器
- 福拉赫循環容器
- 序列容器

將 SSIS 轉換為AWS Glue Studio與AWS SCT

您可以使用AWS SCT將微軟 SQL 伺服器整合服務 (SSIS) 套件轉換為AWS Glue Studio。

一個SSIS 套件包括執行特定擷取、轉換和載入 (ETL) 工作的必要元件，例如連接管理員、工作、控制流程、資料流程、參數、事件處理常式和變數。AWS SCT將 SSIS 套件轉換為相容的格式AWS Glue Studio。將來源資料庫移轉至AWS 雲端，你可以運行這些轉換AWS Glue Studio工作執行 ETL 任務。

若要將微軟 SSIS 套件轉換為AWS Glue Studio，請確定您使用AWS SCT版本 1.0.661 或更高版本。

主題

- [先決條件](#)
- [將 SSIS 套件新增至您的AWS SCT項目](#)
- [將 SSIS 套件轉換為AWS Glue Studio與AWS SCT](#)
- [創建AWS Glue Studio使用轉換後的代碼的工作](#)
- [建立 SSIS 套件的評估報告AWS SCT](#)
- [SSIS 組件AWS SCT可以轉換為AWS Glue Studio](#)

先決條件

在本節中，瞭解 SSIS 封裝轉換為的先決條件工作AWS Glue。這些工作包括建立必要的AWS您帳戶中的資源。

您可以使用AWS Identity and Access Management(IAM) 定義存取資源所需的政策和角色AWS Glue Studio用途。如需詳細資訊，請參閱的 [IAM 許可AWS Glue Studio用戶](#)。

之後AWS SCT將您的來源指令碼轉換為AWS Glue Studio，將轉換後的指令碼上傳到 Amazon S3 儲存貯體。請確定您已建立此 Amazon S3 儲存貯體，並在AWS服務設定檔設定。如需建立 S3 儲存貯體的詳細資訊，請參閱[建立第一個 S3 儲存貯體](#)在亞馬遜簡單存儲服務用戶指南。

為了確保AWS Glue Studio可以連接到您的資料倉庫、建立自訂連接器和連接。另外，將數據庫憑據存儲在AWS Secrets Manager。

建立自訂連接器的步驟

1. 下載資料倉庫的 JDBC 驅動程式。有關 JDBC 驅動程序的更多信息AWS SCT使用，請參閱[下載所需資料庫驅動程式](#)。
2. 將此驅動程式檔案上傳到您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱[將物件上傳至值區](#)在亞馬遜簡單存儲服務用戶指南。
3. 登入 AWS Management Console，然後開啟位於 <https://console.aws.amazon.com/gluestudio/> 的 AWS Glue Studio 主控台。
4. 選擇連接器，然後選擇建立自訂連接器。
5. 對於連接器，選擇瀏覽 S3，然後選擇您上傳到 Amazon S3 儲存貯體的 JDBC 驅動程式檔案。
6. 輸入描述性姓名為您的連接器。例如，輸入 **SQLServer**。

7. 對於連接器類型，選擇JDBC。
8. 對於類別名稱」下方，輸入 JDBC 驅動程式的主類別名稱。對於 SQL 伺服器，請輸入 `com.microsoft.sqlserver.jdbc.SQLServerDriver`。
9. 對於網址基礎」下方，輸入 JDBC 基本網址。JDBC 基本 URL 的語法取決於您的來源資料庫引擎。對於 SQL 伺服器，請使用下列格式：`jdbc:sqlserver://$<host>:$<port>;databaseName=$<dbname>;user=$<username>;password=$<password>`。

確保您更換 `<host>`、`<port>`、`<dbname>`、`<username>`，以及 `<password>` 與你的價值觀。
10. 對於URL 參數分隔符號，輸入分號 (;)。
11. 選擇 Create connector (建立連接器)。

若要將資料庫認證儲存在AWS Secrets Manager

1. 登入AWS Management Console並打開AWS Secrets Manager控制台<https://console.aws.amazon.com/secretsmanager/>。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
 - a. 對於秘密類型，選擇其他類型的秘密。
 - b. 對於鍵/值對」中，輸入下列按鍵：`host,port,dbname,username`，以及`password`。

接下來，輸入這些鍵的值。

4. 在「」配置密碼頁面上，輸入描述性秘密名稱。例如，輸入 `SQL_Server_secret`。
5. 選擇 下一步。然後，在設定旋轉頁面上，選擇下一步再次。
6. 在 Review (檢閱) 頁面上，檢閱機密詳細資訊，然後選擇 Store (存放)。

建立連接器的連線

1. 登入 AWS Management Console，然後開啟位於 <https://console.aws.amazon.com/gluestudio/> 的 AWS Glue Studio 主控台。
2. 選擇您要建立連線的連接器，然後選擇建立連線。
3. 在「」建立連線頁面上，輸入描述性姓名用於您的連接。例如，輸入 `SQL-Server-connection`。
4. 對於AWS秘密」下方，選擇您建立的密碼AWS Secrets Manager。
5. 配置網路選項，然後選擇建立連線。

現在，您可以創建AWS Glue Studio使用自訂連接器的工作。如需詳細資訊，請參閱[創建AWS Glue Studio工作](#)。

將 SSIS 套件新增至您的AWS SCT項目

您可以將多個 SSIS 軟件包添加到單個AWS SCT項目。

若要將 SSIS 套件新增至您的AWS SCT項目

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇SQL 伺服器整合服務。
3. 對於連線名稱」中，輸入 SSIS 套件的名稱。AWS SCT在左側面板的樹狀結構中顯示此名稱。
4. 對於SSIS 套件資料夾」中，輸入含有來源 SSIS 套件之資料夾的路徑。
5. 選擇新增目標從功能表中，然後選擇AWS Glue Studio。

若要連線到AWS Glue Studio,AWS SCT使用您的AWS設定檔。如需詳細資訊，請參閱[將AWS服務設定檔儲存在AWS SCT](#)。

6. 建立對應規則，其中包括來源 SSIS 套件和AWS Glue Studio目標。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。
7. 創建AWS Glue Studio中的連線AWS Glue Studio控制台。如需詳細資訊，請參閱[建立連接器的連接](#)。
8. 選擇連線管理員在左側樹中，打開上下文（右鍵單擊）菜單，然後選擇設定連線。

AWS SCT顯示設定連線窗口。

9. 針對每個來源 SSIS 連線，選擇AWS Glue Studio連接。

將 SSIS 套件轉換為AWS Glue Studio與AWS SCT

接下來，找到如何將 SSIS 包轉換為AWS Glue Studio運用AWS SCT。

若要將 SSIS 套件轉換為AWS Glue Studio

1. 將您的 SSIS 套件新增至您的AWS SCT項目。如需詳細資訊，請參閱[將 SSIS 套件新增至您的AWS SCT項目](#)。
2. 在左側面板中，展開ETL和SSIS節點。

3. 選擇套餐，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換套件。

AWS SCT將您選取的 SSIS 套件轉換為 JSON 檔案。這些 JSON 物件代表有向無環圖 (DAG) 中的節點。找到您的轉換文件封裝 DAG右樹中的節點。

4. 選擇封裝 DAG，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇保存到亞馬遜 S3。

現在，您可以使用這些腳本在AWS Glue Studio。

創建AWS Glue Studio使用轉換後的代碼的工作

轉換來源 SSIS 套件之後，您可以使用轉換後的 JSON 檔案來建立AWS Glue Studio工作。

若要建立AWS Glue Studio工作

1. 選擇封裝 DAG在右側樹中，打開上下文 (右鍵單擊) 菜單，然後選擇配置AWS Glue Studio工作。
2. (選擇性) 套用模擬 SSIS 函數的延伸套件AWS Glue Studio。
3. 該配置AWS Glue Studio工作窗口打開。

完成基本工作屬性區段：

- 姓名— 輸入您的名稱AWS Glue Studio工作。
- 腳本檔案名稱— 輸入工作指令碼的名稱。
- 工作參數— 新增參數並輸入其值。

選擇 下一步。

4. 完成進階工作屬性區段：

- IAM 角色— 選擇用於授權的 IAM 角色AWS Glue Studio並存取資料存放區。
- 指令碼檔案 S3 路徑— 輸入轉換後指令碼的 Amazon S3 路徑。
- 暫存目錄— 輸入 Amazon S3 到暫存目錄的路徑，以取得中繼結果。AWS Glue Studio使用此目錄讀取或寫入亞馬遜紅移。
- AWS SCT會自動產生 Python 程式庫的路徑。您可以在以下位置檢閱此路徑生成的蟒蛇庫路徑。您無法編輯此自動產生的路徑。若要使用其他 Python 程式庫，請在用戶蟒蛇庫路徑。
- 用戶蟒蛇庫路徑— 輸入其他使用者 Python 程式庫的路徑。請使用逗號分隔 Amazon S3 路徑。
- 依賴罐子路徑— 輸入相依的路徑*.jar文件。請使用逗號分隔 Amazon S3 路徑。

- 參考檔案路徑— 輸入指令碼所需的其他檔案 (例如組態檔) 的路徑。請使用逗號分隔 Amazon S3 路徑。
- 工人類型— 選擇G.1X或者G.2X。

當您選擇G.1X每個工作者會對應至 1 個 DPU (4 個 vCPU、16 GB 記憶體和 64 GB 磁碟)。

當您選擇G.2X每個工作者會對應至 2 個 DPU (8 個 vCPU、32 GB 記憶體和 128 GB 磁碟)。

- 要求的工作人數— 輸入工作執行時配置的 Worker 數目。
- 最大並發— 輸入此工作允許的並行執行數目上限。預設值為 1。AWS Glue達到此閾值時返回錯誤。
- 工作逾時 (分鐘)— 輸入 ETL 工作的逾時值，以防止失控工作。批次工作的預設值為 2,880 分鐘 (48 小時)。如果工作超過此限制，工作執行狀態會變更為TIMEOUT。
- 延遲通知閾值 (分鐘) — 在幾分鐘之前輸入閾值AWS SCT傳送延遲通知。
- 重試次數— 輸入的次數 (0—10)AWS Glue如果工作失敗，應該自動重新啟動作業。達到逾時限制的工作不會重新啟動。預設值為 0。

選擇 Finish (完成)。

AWS SCT設定您選取的AWS Glue Studio工作。

5. 在下方尋找您設定的工作ETL 工作在正確的樹中。選擇您設定的工作，開啟環境定義 (按一下滑鼠右鍵) 功能表，然後選擇創建AWS Glue Studio工作。
6. 選擇套用狀態並確保狀態你的工作價值是成功。
7. 打開AWS Glue Studio控制台，選擇刷新，然後選擇您的工作。然後選擇執行。

建立 SSIS 套件的評估報告AWS SCT

該ETL 移轉評估報告提供有關將 SSIS 套件轉換為相容格式的資訊AWS Glue Studio。評估報告包含 SSIS 套件元件的行動項目。這些行動項目會顯示哪些元件AWS SCT無法自動轉換。

若要建立 ETL 移轉評估報告

1. 展開SSIS節點下ETL在左側面板中。
2. 選擇套餐，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇建立報告。
3. 檢視摘要標籤。在這裡，AWS SCT顯示 ETL 移轉評估報告中的執行摘要資訊。它包括 SSIS 套件所有元件的轉換結果。

4. (選擇性) 將 ETL 移轉評估報告的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：

- 若要將 ETL 移轉評估報告儲存為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。

PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。

- 若要將 ETL 移轉評估報告儲存為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。

AWS SCT 會建立三個 CSV 檔案。這些檔案包含行動項目、建議的動作，以及轉換指令碼所需手動工作的估計複雜性。

5. 選擇合適的行動項目標籤。此標籤包含需要手動轉換為的項目清單 AWS Glue Studio。當您從清單中選擇行動項目時，AWS SCT 反白行動項目套用至的來源 SSIS 套件中的項目。

SSIS 組件 AWS SCT 可以轉換為 AWS Glue Studio

您可以使用 AWS SCT 將 SSIS 資料流程元件和參數轉換為 AWS Glue Studio。

支援的資料流程元件包括下列項目：

- ADO 淨目的地
- ADO 淨源
- Aggregate
- 字符映射
- 條件式分割
- 複製欄
- 資料轉換
- 衍生資料欄
- 查找
- Merge
- 合併聯結
- 多點傳送
- 國際旅遊目的地
- 來源
- 奧萊德布目的地
- 奧萊德布源

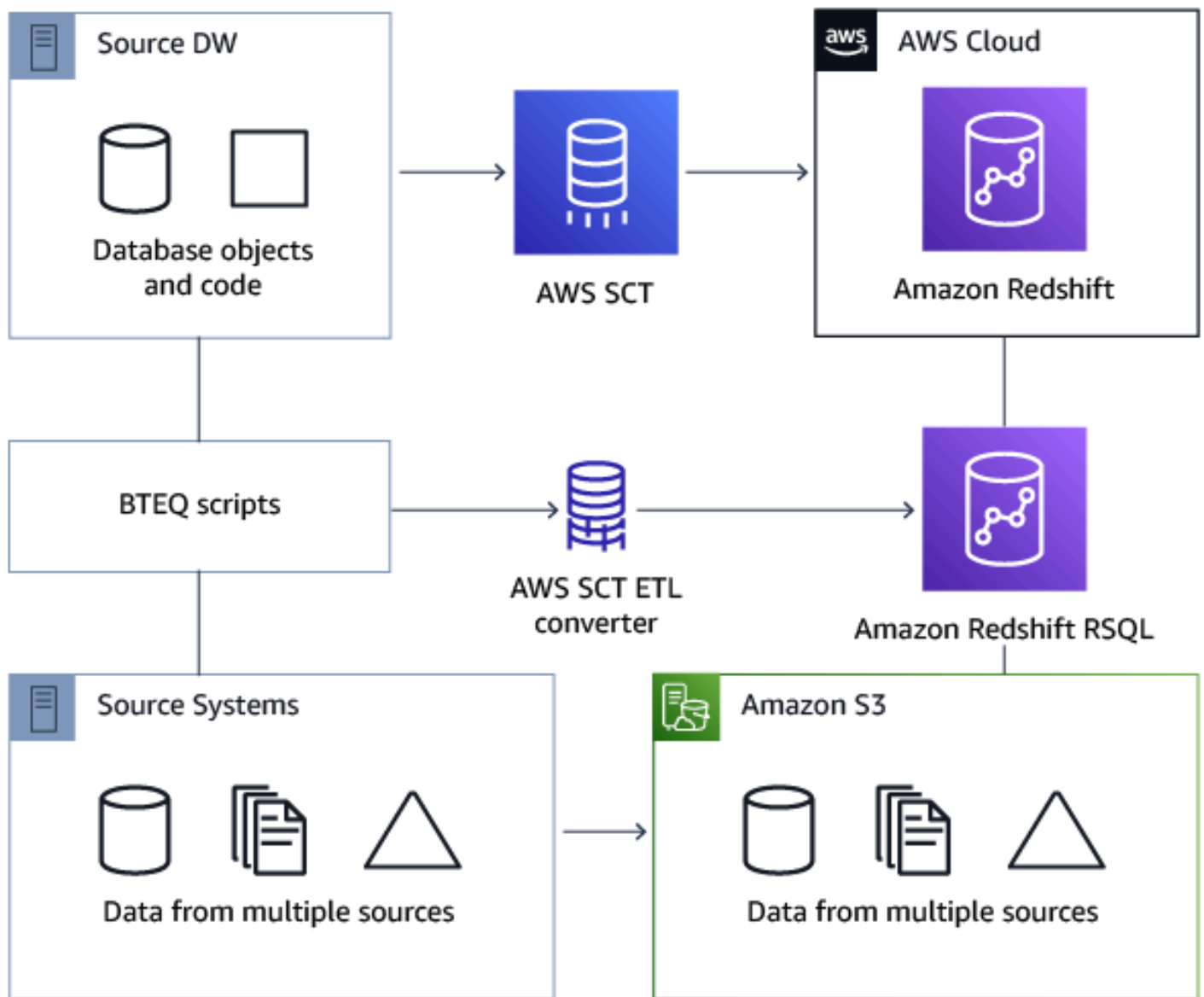
- 列計數
- Sort
- SQL 伺服器目的地
- 全部聯集

AWS SCT可以將更多 SSIS 組件轉換為AWS Glue。如需詳細資訊，請參閱[SSIS 組件AWS SCT可以轉換為AWS Glue](#)。

使用以下方式將 Teradata 指令碼轉換為亞馬遜紅移 RSQLAWS SCT

您可以使用AWS Schema Conversion Tool(AWS SCT) 將基本太資料查詢 (BTEQ) 指令碼轉換為亞馬遜紅移 RSQL。

下列架構圖顯示資料庫遷移專案，其中包括將擷取、轉換和載入 (ETL) 指令碼轉換為 Amazon Redshift RSQL。



主題

- [將 BTEQ 指令碼新增至您的 AWS SCT 項目](#)
- [在 BTEQ 指令碼中設定替代變數 AWS SCT](#)
- [使用以下方式將 Teradata 指令碼轉換為亞馬遜紅移 RSQL AWS SCT](#)
- [使用管理 BTEQ 指令碼 AWS SCT](#)
- [使用建立 BTEQ 指令碼轉換評估報告 AWS SCT](#)
- [使用編輯和保存轉換後的 BTEQ 腳本 AWS SCT](#)

將 BTEQ 指令碼新增至您的AWS SCT項目

您可以將多個腳本添加到一個AWS SCT項目。

若要將 BTEQ 指令碼新增至您的AWS SCT項目

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇太元數據將您的源數據庫添加到項目中。如需詳細資訊，請參閱[使用太數據作為源](#)。
3. 選擇新增目標從菜單中將目標亞馬遜紅移數據庫添加到您的AWS SCT項目。

您可以使用虛擬的亞馬遜紅移目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 建立包含來源 Teradata 資料庫和 Amazon Redshift 目標的新對應規則。如需詳細資訊，請參閱[新增對應規則](#)。
5. 在「」檢視功能表中，選擇主視圖。
6. 在左側面板中，展開腳本節點。
7. 選擇BTEQ 指令碼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇載入指令碼。
8. 輸入您 Teradata BTEQ 指令碼的原始程式碼位置，然後選擇選擇資料夾。

AWS SCT顯示載入指令碼窗口。

9. 執行下列任意一項：
 - a. 如果您的 Teradata BTEQ 指令碼不包含替代變數，請選擇無替代變數，然後選擇好將腳本添加到您的AWS SCT項目。
 - b. 如果您的 Teradata BTEQ 指令碼包含替代變數，請設定替代變數。如需詳細資訊，請參閱[在 BTEQ 指令碼中設定替代變數](#)。

在 BTEQ 指令碼中設定替代變數AWS SCT

您的 Teradata 指令碼可以包含替代變數。例如，您可以使用一個 BTEQ 指令碼搭配替代變數，在多個資料庫環境中執行相同的命令集。您可以使用AWS SCT以在 BTEQ 指令碼中設定替代變數。

在使用替代變數執行 BTEQ 指令碼之前，請務必為所有變數指派值。為此，您可以使用其他工具或應用程式，例如 Bash 腳本，UC4 (自動) 等。AWS SCT只有在指定替代變數值之後，才能解析和轉換替代變數。

若要在 BTEQ 指令碼中設定替代變數

1. 將您的 BTEQ 指令碼新增至您的 AWS SCT 項目。如需詳細資訊，請參閱[將 BTEQ 指令碼新增至您的 AWS SCT 項目](#)。

當您新增指令碼時，請選擇使用替代變數。

2. 對於定義變數格式」中，輸入符合指令碼中所有替代變數的規則運算式。

例如，如果替換變數的名稱開頭為 \${ 並以下列結束 }，使用 `\$\{w+\}` 規則運算式。若要比對以美元符號或百分比符號開頭的替代變數，請使用 `\$w+|\%w+` 規則運算式。

AWS SCT 中的常規表達式皆符合 Java 常規表達式語法。如需詳細資訊，請參閱[正則表達式類模式](#)在 Java 文檔中。

3. 選擇好將腳本加載到您的 AWS SCT 項目，然後選擇好以關閉載入指令碼窗口。
4. 選擇變數查看所有發現的替代變數及其值。
5. 對於價值，輸入替代變數的值。

使用以下方式將 Teradata 指令碼轉換為亞馬遜紅移 RSQLAWS SCT

接下來，找到如何使用 BTEQ ETL 腳本轉換為亞馬遜紅移 RSQLAWS SCT。

若要将元數據 BTEQ 指令碼轉換為亞馬遜紅移 RSQL

1. 將您的 BTEQ 指令碼新增至您的 AWS SCT 項目。如需詳細資訊，請參閱[將 BTEQ 指令碼新增至您的 AWS SCT 項目](#)。
2. 設定替代變數。如需詳細資訊，請參閱[在 BTEQ 指令碼中設定替代變數](#)。
3. 在左側面板中，展開腳本節點。
4. 執行下列任意一項：
 - 若要轉換單一 BTEQ 指令碼，請展開 BTEQ 指令碼節點上，選擇要轉換的指令碼，然後選擇轉換為 RSQL 從上下文 (右鍵單擊) 菜單。
 - 若要隱藏多個指令碼，請務必選取要轉換的所有指令碼。然後選擇 BTEQ 指令碼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換為 RSQL 下轉換腳本。

AWS SCT 將您選取的所有 Teradata 指令碼轉換為與亞馬遜紅移 RSQL 相容的格式。找到您轉換的腳本腳本目標資料庫面板中的節點。

5. 編輯轉換後的亞馬遜紅移 RSQL 腳本，或保存它們。如需詳細資訊，請參閱[編輯並儲存轉換後的 BTEQ 指令碼](#)。

使用管理 BTEQ 指令碼AWS SCT

您可以新增多個 BTEQ 指令碼，或從您的AWS SCT項目。

若要將其他 BTEQ 指令碼新增至您的AWS SCT項目

1. 展開腳本左側面板中的節點。
2. 選擇合適的BTEQ 指令碼節點，然後打開上下文 (右鍵單擊) 菜單。
3. 選擇載入指令碼。
4. 輸入新增 BTEQ 指令集和設定替代變數所需的資訊。如需詳細資訊，請參閱 [將 BTEQ 指令碼新增至您的AWS SCT項目](#) 及 [在 BTEQ 指令碼中設定替代變數](#)。

若要從您的電腦中移除 BTEQ 指令碼AWS SCT項目

1. 展開BTEQ 指令碼節點下腳本在左側面板中。
2. 選擇要移除的指令碼，然後開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇刪除腳本。

使用建立 BTEQ 指令碼轉換評估報告AWS SCT

一個BTEQ 指令碼轉換評估報告提供有關將 BTEQ 命令和 SQL 陳述式從 BTEQ 指令碼轉換為與 Amazon Redshift RSQL 相容格式的資訊。評估報告包括 BTEQ 命令和 SQL 語句的行動項目，AWS SCT無法轉換。

若要建立 BTEQ 指令碼轉換評估報告

1. 展開BTEQ 指令碼節點下腳本在左側面板中。
2. 選擇要轉換的腳本，然後打開上下文 (右鍵單擊) 菜單。
3. 選擇轉換為 RSQL下建立報告。
4. 查看摘要標籤。該摘要標籤顯示 BTEQ 指令碼評估報告中的執行摘要資訊。它包含 BTEQ 指令碼中所有 BTEQ 命令和 SQL 陳述式的轉換結果。
5. (選擇性) 將 BTEQ 指令碼轉換評估報告的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：

- 若要將 BTEQ 指令碼轉換評估報告另存為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。
PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。
 - 若要將 BTEQ 指令碼轉換評估報告儲存為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。
CSV 檔案包含行動項目、建議的動作，以及轉換指令碼所需手動工作的估計複雜性。
6. 選擇合適的行動項目標籤。此索引標籤包含需要手動轉換為亞馬遜紅移 RSQL 的項目清單。當您從清單中選擇行動項目時，AWS SCT 反白行動項目套用的來源 BTEQ 指令碼中的項目。

使用編輯和保存轉換後的 BTEQ 腳本 AWS SCT

您可以在下面的面板中編輯轉換後的腳本 AWS SCT 項目。AWS SCT 將編輯過的指令碼儲存為專案的一部分。

儲存轉換後的指令碼

1. 展開 RSQL 指令碼節點下腳本在目標資料庫面板中。
2. 選擇轉換後的腳本，打開上下文（右鍵單擊）菜單，然後選擇儲存腳本。
3. 輸入文件夾的路徑以保存轉換後的腳本，然後選擇儲存。

AWS SCT 將轉換後的腳本儲存到檔案中，並開啟此檔案。

將具有嵌入式太拉數據 BTEQ 命令的殼層腳本轉換為亞馬遜紅移 RSQL AWS SCT

您可以使用 AWS Schema Conversion Tool (AWS SCT)，將具有內嵌式太級資料基本查詢 (BTEQ) 命令的殼層指令碼轉換為具有內嵌式 Amazon Redshift RSQL 命令的殼層指令碼。

AWS SCT 從殼層指令碼擷取 Teradata BTEQ 命令，並將其轉換為與 Amazon Redshift 相容的格式。將 Teradata 資料庫遷移到 Amazon Redshift 之後，您可以使用這些轉換後的指令碼來管理新的 Amazon Redshift 資料庫。

您也可以使用 AWS SCT 將帶有太數據 BTEQ ETL 腳本的文件轉換為亞馬遜紅移 RSQL。如需詳細資訊，請參閱 [使用以下方式將 Teradata 指令碼轉換為亞馬遜紅移 RSQL AWS SCT](#)。

主題

- [將含有嵌入式 Teradata BTEQ 指令的殼層指令碼新增至您的AWS SCT項目](#)
- [使用嵌入式 Teradata BTEQ 命令在殼層指令碼中設定替代變數AWS SCT](#)
- [使用嵌入式 Teradata BTEQ 命令轉換殼層指令碼AWS SCT](#)
- [使用嵌入式 Teradata BTEQ 命令來管理命令介面指令碼AWS SCT](#)
- [建立殼層指令碼轉換的評估報告AWS SCT](#)
- [編輯和保存轉換後的 shell 腳本AWS SCT](#)

將含有嵌入式 Teradata BTEQ 指令的殼層指令碼新增至您的AWS SCT項目

您可以將多個腳本添加到一個AWS SCT項目。

若要將命令介面指令碼新增至您的AWS SCT項目

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇太元數據將您的源數據庫添加到項目中。如需詳細資訊，請參閱[使用太數據作為源](#)。
3. 選擇新增目標從菜單和目標亞馬遜紅移數據庫添加到您的AWS SCT項目。

您可以使用虛擬的亞馬遜紅移目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 建立包含來源 Teradata 資料庫和 Amazon Redshift 目標的新對應規則。如需詳細資訊，請參閱[新增對應規則](#)。
5. 在「」檢視功能表中，選擇主視圖。
6. 在左側面板中，展開腳本節點。
7. 選擇外殼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇載入指令碼。
8. 使用內嵌 Teradata BTEQ 命令輸入來源殼層指令碼的位置，然後選擇選擇資料夾。

AWS SCT顯示載入指令碼窗口。

9. 執行下列任意一項：
 - 如果您的 shell 腳本不包含替代變量，請選擇無替代變數，然後選擇好將腳本添加到您的AWS SCT項目。
 - 如果您的 shell 腳本包含替代變量，請配置替代變量。如需詳細資訊，請參閱[在 shell 腳本中配置替代變量](#)。

使用嵌入式 Teradata BTEQ 命令在殼層指令碼中設定替代變數AWS SCT

您的 shell 腳本可以包含替代變量。例如，您可以使用具有替代變數的單一指令碼來管理不同環境中的資料庫。您可以使用AWS SCT在 shell 腳本中配置替代變量。

使用 shell 腳本中的替代變量運行 BTEQ 命令之前，請確保為此 shell 腳本中的所有變量分配值。AWS SCT只有在指定替代變數值之後，才能解析和轉換替代變數。

若要在 shell 指令碼中設定替代變數

1. 將您的源 shell 腳本添加到您的AWS SCT項目。如需詳細資訊，請參閱[將 shell 腳本添加到您的 AWS SCT項目](#)。

當您新增指令碼時，請選擇使用替代變量。

2. 對於定義變數格式」中，輸入符合指令碼中所有替代變數的規則運算式。

例如，如果替換變量的名稱開頭為\${並以下列結束}，使用`\$\{\w+\}`規則運算式。若要比對以美元符號或百分比符號開頭的替代變數，請使用`\$\w+|\%\w+`規則運算式。

AWS SCT 中的常規表達式皆符合 Java 常規表達式語法。如需詳細資訊，請參閱[正則表達式類模式](#)在 Java 文檔中。

3. 選擇好將腳本加載到您的AWS SCT項目，然後選擇好以關閉載入指令碼窗口。
4. 選擇變數查看所有發現的替代變量及其值。
5. 對於價值，輸入替代變數的值。

使用嵌入式 Teradata BTEQ 命令轉換殼層指令碼AWS SCT

接下來，瞭解如何將具有嵌入式 Teradata BTEQ 命令的殼層指令碼轉換為使用嵌入式亞馬遜 Redshift RSQL 命令的指令碼進行殼層處理AWS SCT。

若要轉換殼層指令碼

1. 將您的 shell 腳本添加到您的AWS SCT項目。如需詳細資訊，請參閱[將 shell 腳本添加到您的 AWS SCT項目](#)。
2. 設定替代變數。如需詳細資訊，請參閱[在 shell 腳本中配置替代變量](#)。
3. 在左側面板中，展開腳本節點。
4. 執行下列任意一項：

- 若要從單一殼層指令碼轉換 BTEQ 命令，請展開外殼節點上，選擇要轉換的指令碼，然後選擇轉換腳本從上下文 (右鍵單擊) 菜單。
 - 若要隱藏多個指令碼，請務必選取要轉換的所有指令碼。然後選擇外殼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換腳本。
5. 選擇 OK (確定)。

AWS SCT將所選命令介面指令碼中的 BTEQ 命令轉換為與 Amazon Redshift RSQL 相容的格式。找到您轉換的腳本腳本目標資料庫面板中的節點。

6. 編輯轉換後的亞馬遜紅移 RSQL 腳本或保存它們。如需詳細資訊，請參閱[編輯和保存轉換後的 shell 腳本](#)。

使用嵌入式 Teradata BTEQ 命令來管理命令介面指令碼AWS SCT

您可以添加多個 shell 腳本，也可以從AWS SCT項目。

若要將新的 shell 腳本添加到您的AWS SCT項目

1. 展開腳本左側面板中的節點。
2. 選擇合適的外殼節點，然後打開上下文 (右鍵單擊) 菜單。
3. 選擇載入指令碼。
4. 輸入新增 Shell 指令碼和設定替代變數所需的資訊。如需詳細資訊，請參閱 [將 shell 腳本添加到您的AWS SCT項目](#) 及 [在 shell 腳本中配置替代變量](#)。

若要從您的AWS SCT項目

1. 展開外殼節點下腳本在左側面板中。
2. 選擇要移除的指令碼，然後開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇刪除腳本。

建立殼層指令碼轉換的評估報告AWS SCT

該shell 腳本轉換評估報告提供有關轉換 BTEQ 命令和 SQL 陳述式的資訊。將您的來源指令碼轉換為與 Amazon 紅移 RSQL 相容的格式。評估報告包括 BTEQ 命令和 SQL 語句的行動項目，AWS SCT無法轉換。

建立殼層指令碼轉換評估報告

1. 展開外殼節點下腳本在左側面板中。
2. 選擇要轉換的指令碼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇建立報告。
3. 查看摘要標籤。該摘要標籤會顯示 shell 指令碼評估報告中的執行摘要資訊。它包括來源指令碼中所有 BTEQ 命令和 SQL 陳述式的轉換結果。
4. (選擇性) 將 shell 指令碼轉換評估報告的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：
 - 若要將殼層指令碼轉換評估報告另存為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。
PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。
 - 若要將殼層指令碼轉換評估報告儲存為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。
CSV 檔案包含行動項目、建議的動作，以及轉換指令碼所需手動工作的估計複雜性。
5. 選擇合適的行動項目標籤。此索引標籤包含需要手動轉換為亞馬遜紅移 RSQL 的項目清單。當您從清單中選取行動項目時，AWS SCT反白行動項目套用的來源 shell 指令碼中的項目。

編輯和保存轉換後的 shell 腳本AWS SCT

您可以在下面的面板中編輯轉換後的腳本AWS SCT項目。AWS SCT將編輯過的指令碼儲存為專案的一部分。

儲存轉換後的指令碼

1. 展開RSQL 指令碼節點下腳本在目標資料庫面板中。
2. 選擇轉換後的腳本，打開上下文 (右鍵單擊) 菜單，然後選擇儲存腳本。
3. 輸入文件夾的路徑以保存轉換後的腳本，然後選擇儲存。

AWS SCT將轉換後的腳本儲存到檔案中，並開啟此檔案。

轉換太拉數據FastExport亞馬遜紅移 RSQL 的任務腳本AWS SCT

您可以使用AWS Schema Conversion Tool(AWS SCT) 以轉換太元FastExport任務腳本到亞馬遜紅移 RSQL。

一個FastExport工作命令檔是一套FastExport從 Teradata 資料庫中選取和匯出資料的命令和 SQL 陳述式。AWS SCT 皈依FastExport命令和 SQL 語句的格式與亞馬遜紅移 RSQL 兼容。將 Teradata 資料庫遷移到亞馬遜 Redshift 之後，您可以使用這些轉換後的指令碼從亞馬遜 Redshift 資料庫匯出資料。

主題

- [新增FastExport工作腳本到您的AWS SCT項目](#)
- [在 Teradata 中配置替代變量FastExport工作腳本AWS SCT](#)
- [轉換太拉數據FastExport工作腳本AWS SCT](#)
- [管理泰瑞資料FastExport工作腳本AWS SCT](#)
- [為 Teradata 建立評估報告FastExport工作命令檔轉換AWS SCT](#)
- [編輯和保存轉換後的 TeradataFastExport工作腳本AWS SCT](#)

新增FastExport工作腳本到您的AWS SCT項目

您可以將多個腳本添加到一個AWS SCT項目。

若要新增FastExport工作腳本到您的AWS SCT項目

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇太元數據將您的源數據庫添加到項目中。如需詳細資訊，請參閱[使用太數據作為源](#)。
3. 選擇新增目標從菜單和目標亞馬遜紅移數據庫添加到您的AWS SCT項目。

您可以使用虛擬的亞馬遜紅移目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 建立包含來源 Teradata 資料庫和 Amazon Redshift 目標的新對應規則。如需詳細資訊，請參閱[新增對應規則](#)。
5. 在「」檢視功能表中，選擇主視圖。
6. 在左側面板中，展開腳本節點。
7. 選擇FastExport，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇載入指令碼。
8. 輸入原始程式碼的位置FastExport工作腳本並選擇選擇資料夾。

AWS SCT顯示載入指令碼窗口。

9. 執行下列任意一項：

- 如果您的太元數據FastExport工作指令碼不包含替代變數，請選擇無替代變數然後選擇好將腳本添加到您的AWS SCT項目。
- 如果您的太元數據FastExport工作腳本包括替代變量，配置替代變量。如需詳細資訊，請參閱[配置替代變量FastExport工作指令碼](#)。

在 Teradata 中配置替代變量FastExport工作腳本AWS SCT

您的太元數據FastExport工作指令碼可以包含替代變數。例如，您可以使用具有替代變數的單一指令碼，從多個資料庫匯出資料。您可以使用AWS SCT在您的 Teradata 指令碼中設定替代變數。

在運行之前FastExport帶有替代變量的工作腳本，請確保為所有變量分配值。為此，您可以使用其他工具或應用程序，例如 Bash 腳本，UC4 (自動) 等。AWS SCT只有在指定替代變數值之後，才能解析和轉換替代變數。

若要設定替代變數FastExport工作命令檔

1. 新增您的來源FastExport工作腳本到您的AWS SCT項目。如需詳細資訊，請參閱[將 BTEQ 指令碼新增至您的AWS SCT項目](#)。

當您新增指令碼時，請選擇使用替代變量。

2. 對於定義變數格式」中，輸入符合指令碼中所有替代變數的規則運算式。

例如，如果替換變量的名稱開頭為\${並以下列結束}，使用`\${\w+}`規則運算式。若要比對以美元符號或百分比符號開頭的替代變數，請使用`\$\w+|\%\w+`規則運算式。

AWS SCT 中的常規表達式皆符合 Java 常規表達式語法。如需詳細資訊，請參閱[正則表達式類模式](#)在 Java 文檔中。

3. 選擇好將腳本加載到您的AWS SCT項目，然後選擇好以關閉載入指令碼窗口。
4. 在左側面板中，展開腳本節點。選擇FastExport，然後使用腳本選擇您的文件夾。開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇匯出變數下替代變量。
5. 匯出一個指令碼的替代變數。使用指令碼展開資料夾、選擇指令碼、開啟內容 (按一下滑鼠右鍵) 選單，然後選擇匯出變數下替代變量。
6. 輸入逗號分隔值 (CSV) 檔案的名稱以儲存替代變數，然後選擇儲存。
7. 開啟此 CSV 檔案並填入替代變數的值。

視作業系統而定AWS SCT CSV 檔案使用不同的格式。文件中的值可以用引號括起來，也可以不用引號括起來。請確定替代變數的值使用與檔案中其他值相同的格式。AWS SCT無法匯入含有不同格式值的 CSV 檔案。

8. 儲存 CSV 檔案。
9. 在左側面板中，展開腳本節點。選擇FastExport，然後選擇您的指令碼。開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇匯入變數下替代變量。
10. 選擇您的 CSV 檔案，然後選擇打開。
11. 選擇變數查看所有發現的替代變量及其值。

轉換太拉數據FastExport工作腳本AWS SCT

接下來，找到如何轉換太數據FastExport使用亞馬遜紅移 RSQL 的工作AWS SCT。

若要轉換太元資料FastExport任務腳本到亞馬遜紅移 RSQL

1. 添加您的FastExport工作腳本到您的AWS SCT項目。如需詳細資訊，請參閱[新增FastExport工作腳本到您的AWS SCT項目](#)。
2. 設定替代變數。如需詳細資訊，請參閱[配置替代變量FastExport工作指令碼](#)。
3. 在左側面板中，展開腳本節點。
4. 執行下列任意一項：
 - 若要轉換單一FastExport工作命令檔，展開FastExport節點上，選擇要轉換的指令碼，然後選擇轉換腳本從上下文 (右鍵單擊) 菜單。
 - 若要隱藏多個指令碼，請務必選取要轉換的所有指令碼。然後選擇FastExport，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換腳本。

AWS SCT轉換您所有選擇的太數據FastExport任務指令碼為與亞馬遜紅移 RSQL 相容的格式。找到您轉換的腳本腳本目標資料庫面板中的節點。

5. 編輯轉換後的亞馬遜紅移 RSQL 腳本或保存它們。如需詳細資訊，請參閱[編輯並保存轉換後的FastExport工作指令碼](#)。

管理泰瑞資料FastExport工作腳本AWS SCT

您可以添加多個太數據FastExport工作指令碼或移除FastExport來自您的工作腳本AWS SCT項目。

若要新增FastExport工作腳本到您的AWS SCT項目

1. 展開腳本左側面板中的節點。
2. 選擇合適的FastExport節點，然後打開上下文（右鍵單擊）菜單。
3. 選擇載入指令碼。
4. 輸入新增所需的資訊FastExport工作命令檔和設定替代變數。如需詳細資訊，請參閱 [新增FastExport工作腳本到您的AWS SCT項目](#) 及 [配置替代變量FastExport工作指令碼](#)。

若要移除FastExport來自您的工作腳本AWS SCT項目

1. 展開FastExport節點下腳本在左側面板中。
2. 選擇要移除的指令碼，然後開啟內容（按一下滑鼠右鍵）功能表。
3. 選擇刪除腳本。

為 Teradata 建立評估報告FastExport工作命令檔轉換AWS SCT

該FastExport工作命令檔轉換評估報告提供有關轉換FastExport來自您的命令和 SQL 語句FastExport指令碼為與亞馬遜紅移 RSQL 相容的格式。評估報告包含下列項目的行動項目FastExport命令和 SQL 語句AWS SCT無法轉換。

若要建立 Teradata 的指令碼轉換評估報告FastExport工作

1. 展開FastExport節點下腳本在左側面板中。
2. 選擇要轉換的指令碼，開啟內容（按一下滑鼠右鍵）功能表，然後選擇建立報告。
3. 查看摘要標籤。該摘要標籤顯示執行摘要資訊FastExport工作命令檔評估報告。它包括所有人的轉換結果FastExport來自來源指令碼的命令和 SQL 陳述式。
4. 您可以儲存FastExport工作命令檔轉換評估報告為 PDF 檔案或逗號分隔值 (CSV) 檔案。

- a. 若要儲存FastExport工作指令碼轉換評估報告為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。

PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。

- b. 若要儲存FastExport工作指令碼轉換評估報告為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。

CSV 檔案包含行動項目、建議的動作，以及轉換指令碼所需手動工作的估計複雜性。

5. 選擇合適的行動項目標籤。此索引標籤包含需要手動轉換為亞馬遜紅移 RSQL 的項目清單。當您從清單中選取行動項目時，AWS SCT突出顯示來源中的項目FastExport行動項目適用的工作指令碼。

編輯和保存轉換後的 TeradataFastExport工作腳本AWS SCT

您可以在下面的面板中編輯轉換後的腳本AWS SCT項目。AWS SCT將編輯過的指令碼儲存為專案的一部分。

儲存轉換後的指令碼

1. 展開RSQL 指令碼節點下腳本在目標資料庫面板中。
2. 選擇轉換後的腳本，打開上下文（右鍵單擊）菜單，然後選擇儲存腳本。
3. 輸入文件夾的路徑以保存轉換後的腳本，然後選擇儲存。

AWS SCT將轉換後的腳本儲存到檔案中，並開啟此檔案。

轉換太拉數據FastLoad亞馬遜紅移 RSQL 的任務腳本AWS SCT

您可以使用AWS Schema Conversion Tool(AWS SCT) 以轉換太元FastLoad任務腳本到亞馬遜紅移 RSQL。

一個太元數據FastLoad腳本是一組使用多個工作階段在 Teradata 資料庫的空白資料表中載入資料的命令。太元數據FastLoad處理一系列太數據FastLoad命令和 SQL 語句。太元數據FastLoad命令提供會話控制和數據傳輸的數據處理。SQL 敘述句會建立、維護及刪除資料表。

AWS SCT轉換太拉數據FastLoad命令和 SQL 語句的格式與亞馬遜紅移 RSQL 兼容。將 Teradata 資料庫遷移到 Amazon Redshift 之後，您可以使用這些轉換後的指令碼將資料載入您的 Amazon Redshift 資料庫。

主題

- [新增FastLoad工作腳本到您的AWS SCT項目](#)
- [在 Teradata 中配置替代變量FastLoad工作腳本AWS SCT](#)
- [轉換太拉數據FastLoad工作腳本AWS SCT](#)
- [管理泰瑞資料FastLoad工作腳本AWS SCT](#)
- [為 Teradata 建立評估報告FastLoad工作命令檔轉換AWS SCT](#)
- [編輯和保存轉換後的 TeradataFastLoad工作腳本AWS SCT](#)

新增FastLoad工作腳本到您的AWS SCT項目

您可以將多個腳本添加到一個AWS SCT項目。

若要新增FastLoad工作腳本到您的AWS SCT項目

1. 在中建立新專案AWS SCT，或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇太元數據將您的源數據庫添加到項目中。如需詳細資訊，請參閱[使用太數據作為源](#)。
3. 選擇新增目標從菜單中添加一個目標亞馬遜紅移數據庫到您的AWS SCT項目。

您可以使用虛擬的亞馬遜紅移目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 建立包含來源 Teradata 資料庫和 Amazon Redshift 目標的新對應規則。如需詳細資訊，請參閱[新增對應規則](#)。
5. 在「」檢視功能表中，選擇主視圖。
6. 在左側面板中，展開腳本節點。
7. 選擇FastLoad，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇載入指令碼。
8. 輸入來源的位置FastLoad工作腳本並選擇選擇資料夾。

AWS SCT顯示載入指令碼窗口。

9. 執行下列任意一項：
 - 如果您的太元數據FastLoad工作指令碼不包含替代變數，請選擇無替代變數，然後選擇好將腳本添加到您的AWS SCT項目。
 - 如果您的太元數據FastLoad工作腳本包括替代變量，配置替代變量。如需詳細資訊，請參閱[配置替代變量FastLoad工作指令碼](#)。

在 Teradata 中配置替代變量FastLoad工作腳本AWS SCT

您的太元數據FastLoad工作指令碼可能包含替代變數。例如，您可以使用具有替代變數的單一指令碼，將資料載入不同的資料庫。

在運行之前FastLoad帶有替代變量的工作腳本，請確保為所有變量分配值。為此，您可以使用其他工具或應用程序，例如 Bash 腳本，UC4 (自動) 等。

AWS SCT只有在指定替代變數值之後，才能解析和轉換替代變數。在您開始轉換您的源 Teradata 之前FastLoad工作指令碼，請確定您為所有替代變數指派值。您可以使用AWS SCT在您的 Teradata 指令碼中設定替代變數。

若要設定替代變數FastLoad工作命令檔

1. 當您新增您的來源FastLoad工作腳本到您的AWS SCT專案，選擇使用替代變量。如需新增這些指令碼的詳細資訊，請參閱[新增FastLoad工作腳本到您的AWS SCT項目](#)。
2. 對於定義變數格式」中，輸入符合指令碼中所有替代變數的規則運算式。

例如，如果替換變量的名稱開頭為\${並以下列結束}，使用\\$\{\w+\}規則運算式。若要比對以美元符號或百分比符號開頭的替代變數，請使用\\$\w+|\%\w+規則運算式。

AWS SCT 中的常規表達式皆符合 Java 常規表達式語法。如需詳細資訊，請參閱[正則表達式類模式](#)在 Java 文檔中。

3. 選擇好將腳本加載到您的AWS SCT項目，然後選擇好以關閉載入指令碼窗口。
4. 在左側面板中，展開腳本節點。選擇FastLoad，然後使用腳本選擇您的文件夾。開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇匯出變數下替代變量。

此外，您也可以匯出一個指令碼的替代變數。使用指令碼展開資料夾、選擇指令碼、開啟內容 (按一下滑鼠右鍵) 選單，然後選擇匯出變數下替代變量。

5. 輸入逗號分隔值 (CSV) 檔案的名稱以儲存替代變數，然後選擇儲存。
6. 開啟此 CSV 檔案並填入替代變數的值。

視作業系統而定AWS SCTCSV 檔案使用不同的格式。文件中的值可以用引號括起來，也可以不用引號括起來。請確定替代變數的值使用與檔案中其他值相同的格式。AWS SCT無法匯入含有不同格式值的 CSV 檔案。

7. 儲存 CSV 檔案。
8. 在左側面板中，展開腳本節點。選擇FastLoad，然後選擇您的指令碼。開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇匯入變數下替代變量。
9. 選擇您的 CSV 檔案，然後選擇打開。
10. 選擇變數查看所有發現的替代變量及其值。

轉換太拉數據FastLoad工作腳本AWS SCT

接下來，找到如何轉換太數據FastLoad使用亞馬遜紅移 RSQL 的工作AWS SCT。

若要轉換太元資料FastLoad任務腳本到亞馬遜紅移 RSQL

1. 添加您的FastLoad工作腳本到您的AWS SCT項目。如需詳細資訊，請參閱[新增FastLoad工作腳本到您的AWS SCT項目](#)。
2. 設定替代變數。如需詳細資訊，請參閱[配置替代變量FastLoad工作指令碼](#)。
3. 在左側面板中，展開腳本節點。
4. 執行下列任意一項：
 - 若要轉換單一FastLoad工作命令檔，展開FastLoad節點上，選擇要轉換的指令碼，然後選擇轉換腳本從上下文 (右鍵單擊) 菜單。
 - 若要隱藏多個指令碼，請務必選取要轉換的所有指令碼。選擇FastLoad，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換腳本。然後執行下列其中一項：
 - 如果您將來源資料檔案存放在 Amazon S3 上，請選擇S3 物件路徑為來源資料檔案位置。
輸入值亞馬遜 S3 桶文件夾和亞馬遜 S3 存儲桶清單文件為您的源數據文件。
 - 如果您不在 Amazon S3 上存放來源資料檔案，請選擇主機位址為來源資料檔案位置。
輸入值主機的網址或 IP 位址,主機使用者登入名稱，以及亞馬遜 S3 存儲桶用於清單文件為您的源數據文件。
5. 選擇 OK (確定)。

AWS SCT轉換您所有選擇的太數據FastLoad任務指令碼為與亞馬遜紅移 RSQL 相容的格式。找到您轉換的腳本腳本目標資料庫面板中的節點。
6. 編輯轉換後的亞馬遜紅移 RSQL 腳本或保存它們。如需詳細資訊，請參閱[編輯並保存轉換後的FastLoad工作指令碼](#)。

管理泰瑞資料FastLoad工作腳本AWS SCT

您可以添加多個太數據FastLoad工作指令碼或移除FastLoad來自您的工作腳本AWS SCT項目。

若要新增FastLoad工作腳本到您的AWS SCT項目

1. 展開腳本左側面板中的節點。
2. 選擇合適的FastLoad節點並打開上下文 (右鍵單擊) 菜單。
3. 選擇載入指令碼。

4. 輸入新增所需的資訊FastLoad工作命令檔和設定替代變數。如需詳細資訊，請參閱 [新增FastLoad工作腳本到您的AWS SCT項目](#) 及 [配置替代變量FastLoad工作指令碼](#)。

若要移除FastLoad來自您的工作腳本AWS SCT項目

1. 展開FastLoad節點下腳本在左側面板中。
2. 選擇要移除的指令碼，然後開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇刪除腳本。

為 Teradata 建立評估報告FastLoad工作命令檔轉換AWS SCT

該FastLoad工作命令檔轉換評估報告提供有關轉換FastLoad命令和 SQL 語句。將您的來源指令碼轉換為與 Amazon 紅移 RSQL 相容的格式。評估報告包含下列項目的行動項目FastLoad命令和 SQL 語句AWS SCT無法轉換。

若要建立 Teradata 的指令碼轉換評估報告FastLoad工作

1. 展開FastLoad節點下腳本在左側面板中。
2. 選擇要轉換的指令碼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇建立報告。
3. 查看摘要標籤。

該摘要標籤顯示執行摘要資訊FastLoad工作命令檔評估報告。它包括所有人的轉換結果FastLoad來自來源指令碼的命令和 SQL 陳述式。

4. (選擇性) 儲存FastLoad工作命令檔轉換評估報告 (PDF 檔案或逗號分隔值 (CSV) 檔案)：
 - 若要儲存FastLoad工作指令碼轉換評估報告為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。
PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。
 - 若要儲存FastLoad工作指令碼轉換評估報告為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。
CSV 檔案包含行動項目、建議的動作，以及轉換指令碼所需手動工作的估計複雜性。
5. 選擇合適的行動項目標籤。此索引標籤包含需要手動轉換為亞馬遜紅移 RSQL 的項目清單。當您從清單中選取行動項目時，AWS SCT突出顯示來源中的項目FastLoad行動項目適用的工作指令碼。

編輯和保存轉換後的 TeradataFastLoad工作腳本AWS SCT

您可以在下面的面板中編輯轉換後的腳本AWS SCT項目。AWS SCT將編輯過的指令碼儲存為專案的一部分。

儲存轉換後的指令碼

1. 展開RSQL 指令碼節點下腳本在目標資料庫面板中。
2. 選擇轉換後的腳本，打開上下文（右鍵單擊）菜單，然後選擇儲存腳本。
3. 輸入文件夾的路徑以保存轉換後的腳本，然後選擇儲存。

AWS SCT將轉換後的腳本儲存到檔案中，並開啟此檔案。

轉換太拉數據MultiLoad亞馬遜紅移 RSQL 的任務腳本AWS SCT

您可以使用AWS SCT轉換太拉數據MultiLoad任務腳本到亞馬遜紅移 RSQL。

一個太元數據MultiLoad工作命令檔是一組用於批量維護 Teradata 數據庫的命令。太元數據MultiLoad匯入工作會在最多五個不同的表格和檢視上執行許多不同的插入、更新和刪除作業。太元數據MultiLoad刪除任務可以從單個表中刪除大量行。

AWS SCT轉換太拉數據MultiLoad命令和 SQL 語句的格式與亞馬遜紅移 RSQL 兼容。將 Teradata 資料庫遷移到 Amazon Redshift 之後，請使用這些轉換後的指令碼來管理 Amazon Redshift 資料庫中的資料。

主題

- [新增MultiLoad工作腳本到您的AWS SCT項目](#)
- [在 Teradata 中配置替代變量MultiLoad工作腳本AWS SCT](#)
- [轉換太拉數據MultiLoad工作腳本AWS SCT](#)
- [管理泰瑞資料MultiLoad工作腳本AWS SCT](#)
- [為 Teradata 建立評估報告MultiLoad工作命令檔轉換AWS SCT](#)
- [編輯和保存轉換後的 TeradataMultiLoad工作腳本AWS SCT](#)

新增MultiLoad工作腳本到您的AWS SCT項目

您可以將多個腳本添加到一個AWS SCT項目。

若要新增MultiLoad工作腳本到您的AWS SCT項目

1. 在中建立新專案AWS SCT或開啟現有的專案。如需詳細資訊，請參閱[the section called “建立專案”](#)。
2. 選擇新增來源從功能表中，然後選擇太元數據將您的源數據庫添加到項目中。如需詳細資訊，請參閱[使用太數據作為源](#)。
3. 選擇新增目標從菜單和目標亞馬遜紅移數據庫添加到您的AWS SCT項目。

您可以使用虛擬的亞馬遜紅移目標資料庫平台。如需詳細資訊，請參閱[使用虛擬目標](#)。

4. 建立包含來源 Teradata 資料庫和 Amazon Redshift 目標的新對應規則。如需詳細資訊，請參閱[新增對應規則](#)。
5. 在「」檢視功能表中，選擇主視圖。
6. 在左側面板中，展開腳本節點。
7. 選擇MultiLoad，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇載入指令碼。
8. 輸入來源的位置MultiLoad工作腳本並選擇選擇資料夾。

AWS SCT顯示載入指令碼窗口。

9. 執行下列任意一項：
 - 如果您的太元數據MultiLoad工作指令碼不包含替代變數，請選擇無替代變數，然後選擇好將腳本添加到您的AWS SCT項目。
 - 如果您的太元數據MultiLoad工作腳本包括替代變量，配置替代變量。如需詳細資訊，請參閱[配置替代變量MultiLoad工作指令碼](#)。

在 Teradata 中配置替代變量MultiLoad工作腳本AWS SCT

您的太元數據MultiLoad工作指令碼可能包含替代變數。例如，您可以使用具有替代變數的單一指令碼，將資料載入不同的資料庫。

在運行之前MultiLoad帶有替代變量的工作腳本，請確保為所有變量分配值。為此，您可以使用其他工具或應用程式，例如 Bash 腳本，UC4 (自動) 等。

AWS SCT只有在指定替代變數值之後，才能解析和轉換替代變數。在您開始轉換您的源 Teradata 之前MultiLoad工作指令碼，請確定您已為所有替代變數指派值。您可以使用AWS SCT在您的 Teradata 指令碼中設定替代變數。

若要設定替代變數MultiLoad工作命令檔

1. 當您新增您的來源MultiLoad工作腳本到您的AWS SCT專案，選擇使用替代變量。如需新增這些指令碼的詳細資訊，請參閱[新增MultiLoad工作腳本到您的AWS SCT項目](#)。
2. 對於定義變數格式」中，輸入符合指令碼中所有替代變數的規則運算式。

例如，如果替換變量的名稱開頭為\${並以下列結束}，使用`\$\{\w+\}`規則運算式。若要比對以美元符號或百分比符號開頭的替代變數，請使用`\$\w+|\%\w+`規則運算式。

AWS SCT 中的常規表達式皆符合 Java 常規表達式語法。如需詳細資訊，請參閱[正則表達式類模式](#)在 Java 文檔中。

3. 選擇好將腳本加載到您的AWS SCT項目，然後選擇好以關閉載入指令碼窗口。
4. 選擇變數查看所有發現的替代變量及其值。
5. 對於價值，輸入替代變數的值。

轉換太拉數據MultiLoad工作腳本AWS SCT

接下來，找到如何轉換太數據MultiLoad使用亞馬遜紅移 RSQL 的工作AWS SCT。

若要轉換太元資料MultiLoad任務腳本到亞馬遜紅移 RSQL

1. 添加您的MultiLoad工作腳本到您的AWS SCT項目。如需詳細資訊，請參閱[新增MultiLoad工作腳本到您的AWS SCT項目](#)。
2. 設定替代變數並輸入其值。如需詳細資訊，請參閱[配置替代變量MultiLoad工作指令碼](#)。
3. 在左側面板中，展開腳本節點。
4. 執行下列任意一項：
 - 若要轉換單一MultiLoad工作命令檔，展開MultiLoad節點上，選擇要轉換的指令碼，然後選擇轉換腳本從上下文（右鍵單擊）菜單。
 - 若要隱藏多個指令碼，請務必選取要轉換的所有指令碼。選擇MultiLoad，開啟內容（按一下滑鼠右鍵）功能表，然後選擇轉換腳本。
5. 執行下列任意一項：
 - 如果您將來源資料檔案存放在 Amazon S3 上，請選擇S3 物件路徑為了來源資料檔案位置。
輸入亞馬遜 S3 桶文件夾和亞馬遜 S3 存儲桶清單文件為您的源數據文件。
 - 如果您不在 Amazon S3 上存放來源資料檔案，請選擇主機位址為了來源資料檔案位置。

輸入主機的網址或 IP 位址,主機使用者登入名稱, 以及亞馬遜 S3 存儲桶清單文件為您的源數據文件。

6. 選擇 OK (確定)。

AWS SCT轉換您所有選擇的太數據MultiLoad任務指令碼為與亞馬遜紅移 RSQL 相容的格式。在腳本目標資料庫面板中的節點。

7. 編輯轉換後的亞馬遜紅移 RSQL 腳本或保存它們。如需詳細資訊, 請參閱[編輯並保存轉換後的 MultiLoad工作指令碼](#)。

管理泰瑞資料MultiLoad工作腳本AWS SCT

您可以添加多個太數據MultiLoad工作指令碼或移除MultiLoad來自您的工作腳本AWS SCT項目。

若要新增MultiLoad工作腳本到您的AWS SCT項目

1. 展開腳本左側面板中的節點。
2. 選擇合適的MultiLoad節點並打開上下文 (右鍵單擊) 菜單。
3. 選擇載入指令碼。
4. 輸入添加新信息所需的信息MultiLoad工作命令檔和設定替代變數。如需詳細資訊, 請參閱 [新增 MultiLoad工作腳本到您的AWS SCT項目](#) 及 [配置替代變量MultiLoad工作指令碼](#)。

若要移除MultiLoad來自您的工作腳本AWS SCT項目

1. 展開MultiLoad節點下腳本在左側面板中。
2. 選擇要移除的指令碼, 然後開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇刪除腳本。

為 Teradata 建立評估報告MultiLoad工作命令檔轉換AWS SCT

該MultiLoad工作命令檔轉換評估報告提供有關轉換MultiLoad命令和 SQL 語句。這是從您的來源指令碼轉換成亞馬遜紅移 RSQL 命令和適用於亞馬遜紅移的 SQL 陳述式。評估報告包含下列項目的行動項目MultiLoad命令和 SQL 語句AWS SCT無法轉換。

若要建立 Teradata 的指令碼轉換評估報告MultiLoad工作

1. 展開MultiLoad節點下腳本在左側面板中。

2. 選擇要建立評量報告的指令碼，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇建立報告。
3. 查看摘要標籤。該摘要標籤顯示執行摘要資訊MultiLoad工作命令檔評估報告。它包括所有人的轉換結果MultiLoad來自來源指令碼的命令和 SQL 陳述式。
4. (選擇性) 儲存MultiLoad作為 PDF 檔案或逗號分隔值 (CSV) 檔案的工作命令檔轉換評估報告：
 - 若要儲存MultiLoad工作指令碼轉換評估報告為 PDF 檔案，請選擇儲存為 PDF 格式在右上角。
PDF 檔案包含執行摘要、行動項目和指令碼轉換的建議。
 - 若要儲存MultiLoad工作指令碼轉換評估報告為 CSV 檔案，請選擇儲存為 CSV 格式在右上角。
AWS SCT會建立兩個 CSV 檔案。這些檔案包含執行摘要、行動項目、建議的動作，以及轉換指令碼所需的預估手動工作複雜度。
5. 選擇合適的行動項目標籤。此索引標籤包含需要手動轉換為亞馬遜紅移 RSQL 的項目清單。當您從清單中選取行動項目時，AWS SCT突出顯示來源中的項目MultiLoad行動項目適用的工作指令碼。

編輯和保存轉換後的 TeradataMultiLoad工作腳本AWS SCT

您可以在下面的面板中編輯轉換後的腳本AWS SCT項目。AWS SCT將編輯過的指令碼儲存為專案的一部分。

儲存轉換後的指令碼

1. 展開RSQL 指令碼節點下腳本在目標資料庫面板中。
2. 選擇轉換後的腳本，打開上下文 (右鍵單擊) 菜單，然後選擇儲存腳本。
3. 輸入文件夾的路徑以保存轉換後的腳本，然後選擇儲存。

AWS SCT將轉換後的腳本儲存到檔案中，並開啟此檔案。

遷移大數據框架 AWS Schema Conversion Tool

您可以使用 AWS Schema Conversion Tool (AWS SCT) 將大數據框架遷移到AWS 雲端。

目前，AWS SCT支持 Hadoop 集群遷移到亞馬遜 EMR 和亞馬遜 S3。此遷移過程包括配置單元和 HDFS 服務。

此外，您可以使用自動AWS SCT將 Apache Oozie 協調流程工作流程轉換為 AWS Step Functions

主題

- [遷移阿帕奇 Hadoop 到亞馬遜 EMR 與 AWS Schema Conversion Tool](#)
- [將阿帕奇奧西轉換AWS Step Functions為 AWS Schema Conversion Tool](#)

遷移阿帕奇 Hadoop 到亞馬遜 EMR 與 AWS Schema Conversion Tool

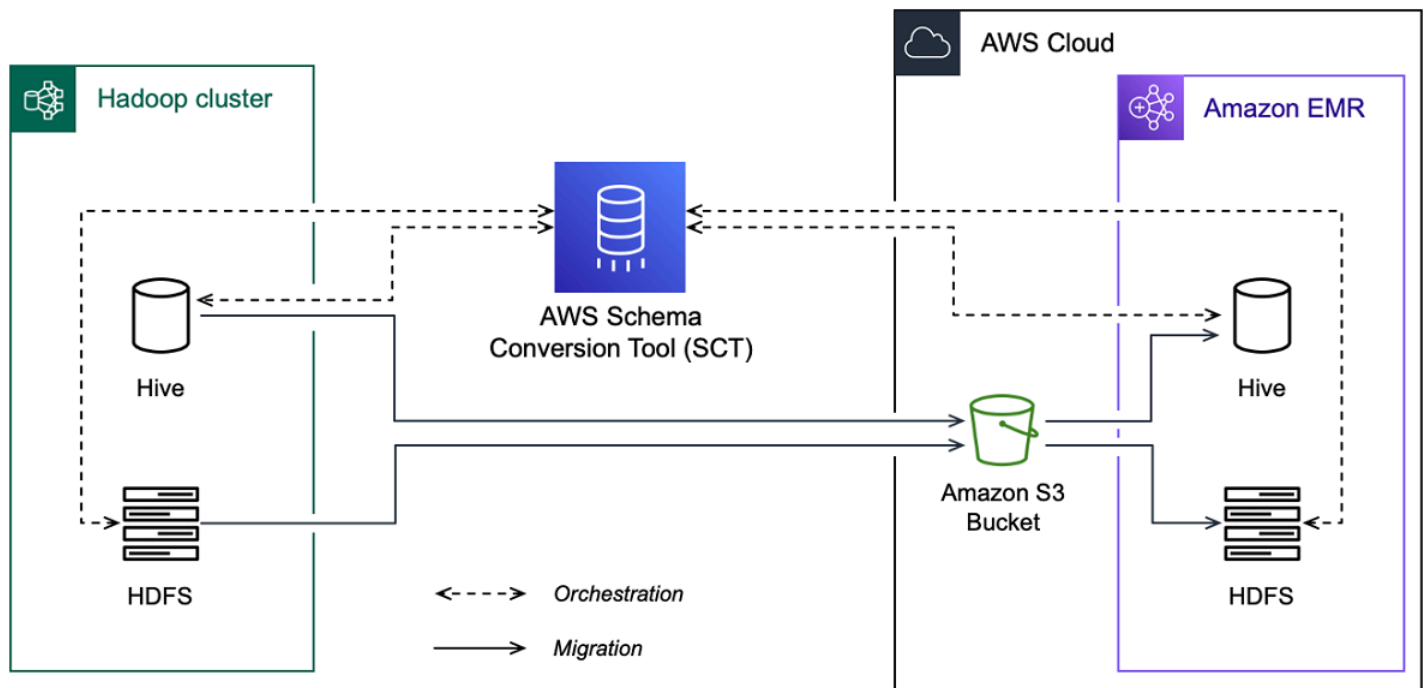
若要移轉阿帕奇 Hadoop 叢集，請確定您使用的是 1.0.670 或更高AWS SCT版本。此外，請熟悉的命令行介面 (CLI)。AWS SCT如需詳細資訊，請參閱[AWS SCT CLI 參考資料](#)。

主題

- [移轉概觀](#)
- [步驟 1：連接到您的 Hadoop 叢集](#)
- [步驟 2：設定對應規則](#)
- [步驟 3：建立評估報告](#)
- [第 4 步：將您的阿帕奇 Hadoop 集群遷移到亞馬遜 EMR AWS SCT](#)
- [執行您的 CLI 指令碼](#)
- [管理您的大數據遷移專案](#)

移轉概觀

下圖顯示了從阿帕奇 Hadoop 遷移到亞馬遜 EMR 的體系結構圖。



AWS SCT將資料和中繼資料從來源 Hadoop 叢集遷移到 Amazon S3 儲存貯體。接下來，AWS SCT 使用來源 Hive 中繼資料，在目標 Amazon EMR Hive 服務中建立資料庫物件。或者，您可以配置 Hive 使用AWS Glue Data Catalog作為其中繼存儲。在此情況下，請將來源 Hive 中繼資料AWS SCT移轉至. AWS Glue Data Catalog

然後，您可以使用AWS SCT將資料從 Amazon S3 儲存貯體遷移到目標亞馬遜 EMR HDFS 服務。或者，您可以將資料保留在 Amazon S3 儲存貯體中，並將其用作 Hadoop 工作負載的資料儲存庫。

若要啟動 Hadoop 移轉，請建立並執行 AWS SCT CLI 指令碼。此指令碼包含用於執行移轉的完整命令集。您可以下載並編輯 Hadoop 遷移指令碼的範本。如需詳細資訊，請參閱[取得 CLI 案例](#)。

請確定您的指令碼包含下列步驟，以便您可以執行從 Apache Hadoop 到亞馬遜 S3 和亞馬遜 EMR 的遷移。

步驟 1：連接到您的 Hadoop 叢集

若要開始移轉您的 Apache Hadoop 叢集，請建立新AWS SCT專案。接下來，連接到您的來源和目標叢集。在開始移轉之前，請確定您已建立並佈建目標AWS資源。

在此步驟中，您可以使用下列 AWS SCT CLI 命令。

- `CreateProject`— 創建一個新的AWS SCT項目。
- `AddSourceCluster`-連接到源 Hadoop 集群在您的AWS SCT項目。

- AddSourceClusterHive—連接到源 Hive 服務在您的項目。
- AddSourceClusterHDFS—連接到項目中的源 HDFS 服務。
- AddTargetCluster—連接到您的項目中的目標亞馬遜 EMR 集群。
- AddTargetClusterS3—將亞馬遜 S3 存儲桶添加到您的項目中。
- AddTargetClusterHive—連接到項目中的目標蜂巢服務
- AddTargetClusterHDFS—連接到項目中的目標 HDFS 服務

如需使用這些 AWS SCT CLI 指令的範例，請參閱[使用阿帕奇 Hadoop 作為源](#)。

當您執行連線至來源或目標叢集的命令時，AWS SCT 會嘗試建立與此叢集的連線。如果連線嘗試失敗，則會 AWS SCT 停止從 CLI 指令碼執行命令，並顯示錯誤訊息。

步驟 2：設定對應規則

連線到來源和目標叢集之後，請設定對應規則。對應規則定義來源叢集的移轉目標。請確定您已為 AWS SCT 專案中新增的所有來源叢集設定對應規則。如需對應規則的詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

在此步驟中，您可以使用 AddServerMapping 指令。此命令使用兩個參數，用於定義來源叢集和目標叢集。您可以使用 AddServerMapping 指令搭配資料庫物件的明確路徑或物件名稱。對於第一個選項，您可以包括物件的類型及其名稱。對於第二個選項，您只包括物件名稱。

- sourceTreePath—來源資料庫物件的明確路徑。
 - targetTreePath—目標資料庫物件的明確路徑。
- sourceNamePath—僅包含來源物件名稱的路徑。
 - targetNamePath—僅包含目標物件名稱的路徑。

下列程式碼範例會使用來源 testdb Hive 資料庫和目標 EMR 叢集的明確路徑來建立對應規則。

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

您可以在 Windows 中使用這個範例和下列範例。若要在 Linux 中執行 CLI 指令，請確定您已適當地更新作業系統的檔案路徑。

下列程式碼範例會使用僅包含物件名稱的路徑建立對應規則。

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

您可以選擇亞馬遜 EMR 或亞馬遜 S3 作為來源物件的目標。對於每個來源物件，您只能在單一 AWS SCT 專案中選擇一個目標。若要變更來源物件的移轉目標，請刪除現有的對應規則，然後建立新的對應規則。若要刪除對應規則，請使用 `DeleteServerMapping` 指令。此命令使用以下兩個參數之一。

- `sourceTreePath`— 來源資料庫物件的明確路徑。
- `sourceNamePath`— 僅包含來源物件名稱的路徑。

如需有關 `AddServerMapping` 和 `DeleteServerMapping` 命令的詳細資訊，請參閱 [AWS Schema Conversion Tool CLI 參考資料](#)。

步驟 3：建立評估報告

在開始移轉之前，我們建議您先建立評估報告。此報告概述了所有移轉工作，並詳細說明移轉期間將出現的行動項目。若要確定您的移轉不會失敗，請在移轉前檢視此報告並處理行動項目。如需詳細資訊，請參閱 [遷移評定報告](#)。

在此步驟中，您可以使用 `CreateMigrationReport` 指令。此命令使用兩個參數。該 `treePath` 參數是必要的，並且 `forceMigrate` 參數是可選的。

- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- `forceMigrate`— 當設置為 `true`，AWS SCT 繼續遷移，即使您的項目包括 HDFS 文件夾和 Hive 表參考相同的對象。預設值為 `false`。

然後，您可以將評估報告的副本儲存為 PDF 或逗號分隔值 (CSV) 檔案。若要這麼做，請使用 `SaveReportPDF` 或 `SaveReportCSV` 指令。

此指 `SaveReportPDF` 令會將評估報告的副本儲存為 PDF 檔案。此命令使用四個參數。該 `file` 參數是強制性的，其他參數是可選的。

- `file`— PDF 檔案的路徑及其名稱。
- `filter`— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。

- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- `namePath`— 僅包含您儲存評估報告副本之目標物件名稱的路徑。

此指 `SaveReportCSV` 命令會將您的評估報告儲存為三個 CSV 檔案。此命令使用四個參數。該 `directory` 參數是強制性的，其他參數是可選的。

- `directory`— AWS SCT 儲存 CSV 檔案的資料夾路徑。
- `filter`— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。
- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- `namePath`— 僅包含您儲存評估報告副本之目標物件名稱的路徑。

下列程式碼範例會將評估報告的副本儲存在 `c:\sct\ar.pdf` 檔案中。

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

下列程式碼範例會將評估報告的副本儲存為 CSV 檔案在 `c:\sct` 資料夾中。

```
SaveReportCSV
-file:'c:\sct'
/
```

如需有關 `SaveReportPDF` 和 `SaveReportCSV` 命令的詳細資訊，請參閱 [AWS Schema Conversion Tool CLI 參考資料](#)。

第 4 步：將您的阿帕奇 Hadoop 集群遷移到亞馬遜 EMR AWS SCT

設定 AWS SCT 專案之後，請開始將內部部署的 Apache Hadoop 叢集移轉至 AWS 雲端

在此步驟中，您將使用 `MigrateMigrationStatus`、和 `ResumeMigration` 指令。

此指 `Migrate` 命令會將您的來源物件移轉至目標叢集。此命令使用四個參數。請確定您已指定 `filter` 或 `treePath` 參數。其他參數是可選的。

- `filter`— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。
- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。

- `forceLoad`— 設為時`true`，AWS SCT會在移轉期間自動載入資料庫中繼資料樹狀結構。預設值為`false`。
- `forceMigrate`— 當設置為`true`，AWS SCT繼續遷移，即使您的項目包括HDFS文件夾和Hive表參考相同的對象。預設值為`false`。

此命`MigrationStatus`令會傳回有關移轉進度的資訊。若要執行此指令，請輸入`name`參數的移轉專案名稱。您在`CreateProject`命令中指定了此名稱。

此命`ResumeMigration`令會繼續您使用指`Migrate`令啟動的中斷移轉。該`ResumeMigration`命令不使用參數。若要繼續移轉，您必須連線至來源和目標叢集。如需詳細資訊，請參閱[管理您的移轉專案](#)。

下列程式碼範例會將資料從來源HDFS服務遷移到Amazon EMR。

```
Migrate
-treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
-forceMigrate: 'true'
/
```

執行您的 CLI 指令碼

完成AWS SCT CLI指令碼的編輯後，請將其儲存為`.scts`副檔名為的檔案。現在，您可以從AWS SCT安裝路徑的`app`文件夾運行腳本。為此，請使用下列命令。

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

在上述範例中，使用CLI指令碼將`script_path`取代為檔案的路徑。如需有關在中執行CLI指令碼的詳細資訊AWS SCT，請參閱[指令碼模式](#)。

管理您的大數據遷移專案

完成移轉後，您可以儲存和編輯AWS SCT專案以供將來使用。

若要儲存AWS SCT專案，請使用`SaveProject`指令。此命令不使用參數。

下列程式碼範例會儲存您的AWS SCT專案。

```
SaveProject
/
```

若要開啟AWS SCT專案，請使用OpenProject指令。此命令使用一個強制參數。對於file參數，請輸入AWS SCT專案檔案的路徑及其名稱。您在指CreateProject令中指定了專案名稱。確保將.scts擴展名添加到項目文件的名稱中以運行該OpenProject命令。

下列程式碼範例會從c:\sct資料夾開啟hadoop_emr專案。

```
OpenProject
-file: 'c:\sct\hadoop_emr.scts'
/
```

開啟AWS SCT專案之後，您不需要新增來源和目標叢集，因為您已將它們新增至專案。若要開始使用來源和目標叢集，您必須連線至它們。若要這麼做，您可以使用ConnectSourceCluster和ConnectTargetCluster指令。這些指令使用與指AddTargetCluster令相同AddSourceCluster的參數。您可以編輯 CLI 指令碼，並取代這些命令的名稱，不需要變更即可離開參數清單。

下面的代碼示例連接到源 Hadoop 集群。

```
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'
/
```

下列程式碼範例會連線至目標 Amazon EMR 叢集。

```
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'
```

/

在上述範例中，請將 *hadoop_address* 取代為 Hadoop 叢集的 IP 位址。如果需要，請配置端口變量的值。接下來，用您的 *Hadoop ##* 的名稱和此用戶的 *#### hadoop_ user* 和密碼。針對 *## \ ##*，請輸入來源 Hadoop 叢集之 PEM 檔案的名稱和路徑。如需新增來源和目標叢集的詳細資訊，請參閱[使用阿帕奇 Hadoop 作為源 AWS SCT](#)。

連線到來源和目標 Hadoop 叢集之後，您必須連線到您的 Hive 和 HDFS 服務，以及您的 Amazon S3 儲存貯體。若要這麼做，您可以使用 `ConnectSourceClusterHiveConnectSourceClusterHdfsConnectTargetClusterHive`、`ConnectSourceClusterHdfs` 和 `ConnectTargetClusterS3` 指令。這些命令使用的參數與您用來將 Hive 和 HDFS 服務以及 Amazon S3 儲存貯體新增至專案的命令相同的參數。編輯 CLI 指令碼，以取代命令名稱 `Connect` 中的 `Add` 前置詞。

將阿帕奇奧西轉換為 AWS Step Functions 為 AWS Schema Conversion Tool

若要轉換 Apache 工作流程，請確定您使用的是 1.0.671 或更高 AWS SCT 版本。此外，請熟悉的指令行介面 (CLI)。AWS SCT 如需詳細資訊，請參閱[AWS SCT CLI 參考資料](#)。

主題

- [轉換概述](#)
- [步驟 1：連線至來源和目標服務](#)
- [步驟 2：設定對應規則](#)
- [步驟 3：設定參數](#)
- [步驟 4：建立評估報告](#)
- [步驟 5：將您的 Apache 工作流程轉換為 AWS Step Functions AWS SCT](#)
- [執行您的 CLI 指令碼](#)
- [阿帕奇奧西節點，可 AWS SCT 以轉換為 AWS Step Functions](#)

轉換概述

您的 Apache Oozie 原始程式碼包含動作節點、控制流程節點和工作屬性。動作節點定義工作，您在 Apache Oozie 工作流程中執行。當您使用阿帕奇奧西來協調您的 Apache Hadoop 集群，然後一個動作節點包括一個 Hadoop 作業。控制流程節點提供控制工作流程路徑的機制。控制流程節點包括、start、enddecisionfork、和等節點join。

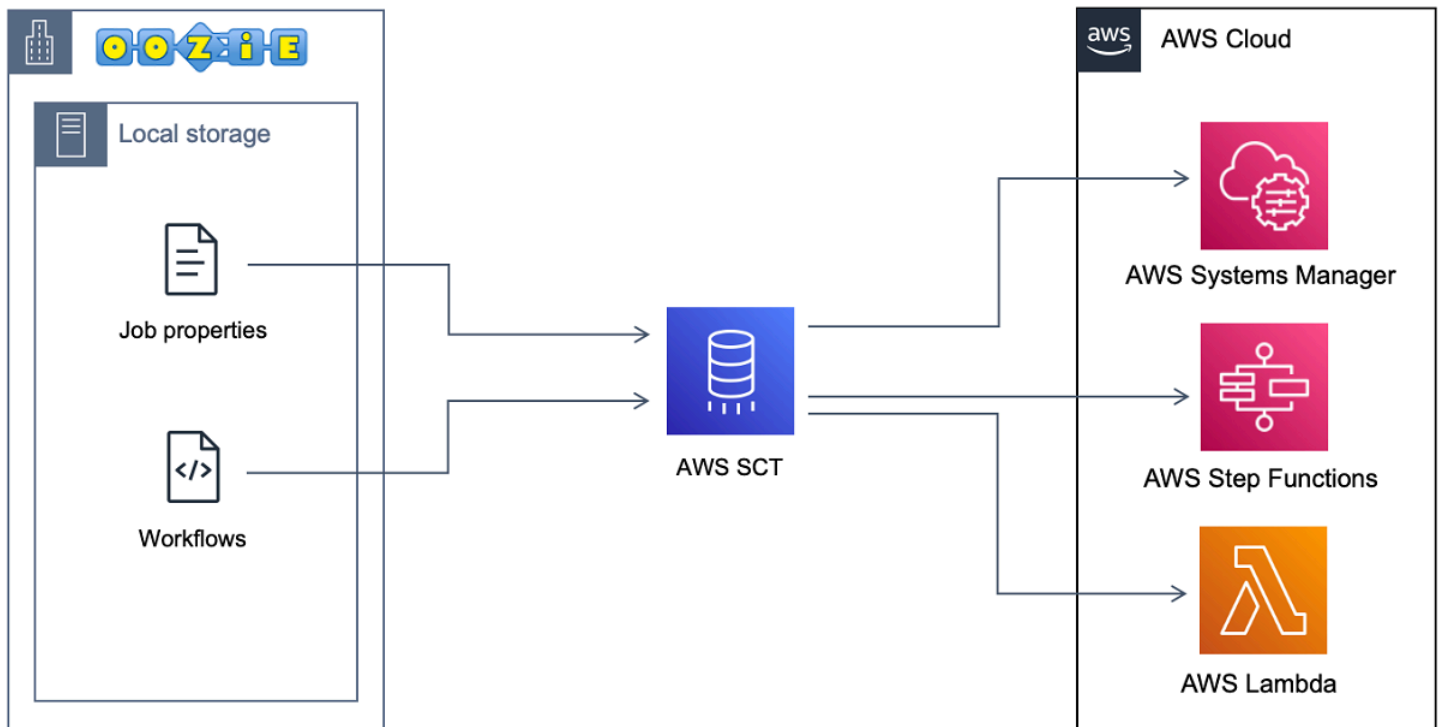
AWS SCT將您的源動作節點和控制流節點轉換為AWS Step Functions。在中AWS Step Functions，您可以使用亞馬遜州語言 (ASL) 定義工作流程。AWS SCT使用 ASL 來定義您的狀態機器，這是狀態的集合，可以執行工作，確定要轉換到下一個狀態，停止並出現錯誤，等等。接下來，AWS SCT上傳帶有狀態機器定義的 JSON 文件。然後，AWS SCT可以使用您的AWS Identity and Access Management (IAM) 角色在中配置狀態機器AWS Step Functions。如需詳細資訊，請參閱《AWS Step Functions 開發人員指南》中的[什麼是 AWS Step Functions ?](#) 章節。

此外，AWS SCT創建一個擴展包AWS Lambda，其中包含模擬AWS Step Functions不支持的源函數的函數。如需詳細資訊，請參閱[使用 AWS SCT 擴充套件](#)。

AWS SCT將來源工作屬性移轉至AWS Systems Manager。若要儲存參數名稱和值，請AWS SCT使用的功能參數存放區AWS Systems Manager。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的[什麼是 AWS Systems Manager ?](#)。

您可以使用AWS SCT來自動更新參數的值和名稱。由於 Apache Oozie 和之間的架構差異AWS Step Functions，您可能需要配置參數。AWS SCT可以在源文件中找到指定的參數名稱或值，並用新值替換它們。如需詳細資訊，請參閱[步驟 3：設定參數](#)。

下圖顯示了阿帕奇 Oozie 轉換為AWS Step Functions的體系結構圖。



若要開始轉換，請建立並執行 AWS SCT CLI 指令碼。此指令碼包含執行轉換的完整命令集。您可以下載和編輯 Apache Oozie 轉換腳本的模板。如需詳細資訊，請參閱[取得 CLI 案例](#)。

請確定您的指令碼包含下列步驟。

步驟 1：連線至來源和目標服務

若要開始轉換您的 Apache Oozie 叢集，請建立新 AWS SCT 專案。接下來，連接到您的源和目標服務。在開始移轉之前，請確定您已建立並佈建目標 AWS 資源。如需詳細資訊，請參閱[使用阿帕奇作為源代碼的先決條件](#)。

在此步驟中，您可以使用下列 AWS SCT CLI 命令。

- `CreateProject`— 創建一個新的 AWS SCT 項目。
- `AddSource`— 在您的 AWS SCT 項目中添加您的源阿帕奇 Ozie 文件。
- `ConnectSource`— 連接到阿帕奇奧西作為源。
- `AddTarget`-添加 AWS Step Functions 為項目中的遷移目標。
- `ConnectTarget`— 要連接到 AWS Step Functions。

如需使用這些 AWS SCT CLI 指令的範例，請參閱[使用阿帕奇大西作為來源](#)。

當您執行ConnectSource或命ConnectTarget令時，AWS SCT會嘗試建立與服務的連線。如果連線嘗試失敗，則會AWS SCT停止從 CLI 指令碼執行命令，並顯示錯誤訊息。

步驟 2：設定對應規則

連線到來源和目標服務之後，請設定對應規則。對應規則會定義來源 Apache Oozie 工作流程和參數的移轉目標。如需對應規則的詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

若要定義要轉換的來源物件和目標物件，請使用AddServerMapping指令。此命令使用兩個參數：sourceTreePath和targetTreePath。這些參數的值包括來源和目標物件的明確路徑。對於 Apache Oozie 進行轉AWS Step Functions換，這些參數必須以開頭ETL。

下列程式碼範例會建立OOZIE和AWS_STEP_FUNCTIONS物件的對應規則。您已在上一個步驟中使用AddSource和AddTarget指令將這些物件加入到AWS SCT專案中。

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

如需有關AddServerMapping命令的詳細資訊，請參閱 [AWS Schema Conversion Tool CLI 參考資料](#)。

步驟 3：設定參數

如果您的來源 Apache Oozie 工作流程使用參數，您可能需要在轉換為AWS Step Functions. 此外，您可能需要添加新參數以與AWS Step Functions.

在此步驟中，您將使用AddParameterMapping和AddTargetParameter指令。

若要取代來源檔案中的參數值，請使用AddParameterMapping指令。AWS SCT掃描來源檔案、依名稱或值尋找參數，並變更其值。您可以運行單個命令來掃描所有源文件。您可以使用下列清單中的前三個參數之一來定義要掃描的檔案範圍。此指令最多可使用六個參數。

- filterName— 來源物件的篩選器名稱。您可以使用CreateFilter指令建立篩選。
- treePath— 來源物件的明確路徑。
- namePath— 特定來源物件的明確路徑。
- sourceParameterName— 來源參數的名稱。

- `sourceValue`— 來源參數的值。
- `targetValue`— 目標參數的值。

下列程式碼範例會取代值等於值的 `c:\oozie\hive.py` 所有參數。 `s3://bucket-oozie/hive.py`

```
AddParameterMapping
  -treePath: 'ETL.OOZIE.Applications'
  -sourceValue: 'c:\oozie\hive.py'
  -targetValue: 's3://bucket-oozie/hive.py'
/
```

下列程式碼範例會取代名稱等於 `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` 值的 `nameNode` 所有參數。

```
AddParameterMapping
  -treePath: 'ETL.OOZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

下列程式碼範例會取代名稱等於 `nameNode` 且值等於參數值 `hdfs://ip-55.eu-west-1.compute.internal:8020` 的所有 `targetValue` 參數。

```
AddParameterMapping
  -treePath: 'ETL.OOZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

除了來源檔案中的現有參數外，若要在目標檔案中新增參數，請使用 `AddTargetParameter` 指令。此指令使用與指 `AddParameterMapping` 令相同的一組參數。

下列程式碼範例會新增 `clusterId` 目標參數，而不是 `nameNode` 參數。

```
AddTargetParameter
  -treePath: 'ETL.OOZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetParameter: 'clusterId'
```

```
-targetValue: '1234567890abcdef0'
```

```
/
```

如需有

關AddServerMapping、AddParameterMapping、AddTargetParameter和CreateFilter命令的詳細資訊，請參閱 [AWS Schema Conversion Tool CLI 參考資料](#)。

步驟 4：建立評估報告

在開始轉換之前，我們建議您先建立評估報告。此報告概述了所有移轉工作，並詳細說明移轉期間將出現的行動項目。若要確定您的移轉不會失敗，請在移轉前檢視此報告並處理行動項目。如需詳細資訊，請參閱 [遷移評定報告](#)。

在此步驟中，您可以使用CreateReport指令。此命令使用兩個參數。第一個參數說明AWS SCT建立評估報告的來源物件。若要這麼做，請使用下列其中一個參數：filterNametreePath、或namePath。此參數是強制性的。此外，您還可以添加一個可選的布林參數forceLoad。如果將此參數設定為true，則AWS SCT會自動載入您在指CreateReport令中指定之來源物件的所有子物件。

下列程式碼範例會針對來源 Oozie 檔案的Applications節點建立評估報告。

```
CreateReport
  -treePath: 'ETL.APACHE_00ZIE.Applications'
```

然後，您可以將評估報告的副本儲存為 PDF 或逗號分隔值 (CSV) 檔案。若要這麼做，請使用SaveReportPDF或SaveReportCSV指令。

此指SaveReportPDF令會將評估報告的副本儲存為 PDF 檔案。此命令使用四個參數。該file參數是強制性的，其他參數是可選的。

- file— PDF 檔案的路徑及其名稱。
- filter— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。
- treePath— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- namePath— 僅包含您儲存評估報告副本之目標物件名稱的路徑。

此指SaveReportCSV令會將您的評估報告儲存為 CSV 檔案。此命令使用四個參數。該directory參數是強制性的，其他參數是可選的。

- `directory`— AWS SCT 儲存 CSV 檔案的資料夾路徑。
- `filter`— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。
- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- `namePath`— 僅包含您儲存評估報告副本之目標物件名稱的路徑。

下列程式碼範例會將評估報告的副本儲存在 `c:\sct\ar.pdf` 檔案中。

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

下列程式碼範例會將評估報告的副本儲存為 CSV 檔案在 `c:\sct` 資料夾中。

```
SaveReportCSV
-file:'c:\sct'
/
```

如需有關 `CreateReport`、`SaveReportPDF` 和 `SaveReportCSV` 命令的詳細資訊，請參閱 [AWS Schema Conversion Tool CLI 參考](#)。

步驟 5：將您的 Apache 工作流程轉換為 AWS Step Functions AWS SCT

設定 AWS SCT 專案之後，請轉換原始程式碼並將其套用至 AWS 雲端。

在此步驟中，您將使用 `ConvertSaveOnS3`、`ConfigureStateMachine`、和 `ApplyToTarget` 指令。

此指 `Migrate` 指令會將您的來源物件移轉至目標叢集。此指令使用四個參數。請確定您已指定 `filter` 或 `treePath` 參數。其他參數是可選的。

- `filter`— 您之前建立的篩選器名稱，以定義要移轉之來源物件的範圍。
- `namePath`— 特定來源物件的明確路徑。
- `treePath`— 您儲存評估報告副本之來源資料庫物件的明確路徑。
- `forceLoad`— 設為 `true`，AWS SCT 會在移轉期間自動載入資料庫中繼資料樹狀結構。預設值為 `false`。

下列程式碼範例會從來源 Oozie 檔案中的 `Applications` 資料夾轉換檔案。

```
Convert
  -treePath: 'ETL.APACHE_00ZIE.Applications'
/
```

將狀態機器定義SaveOnS3上傳到您的 Amazon S3 儲存貯體。此命令使用treePath參數。若要執行此命令，請使用具有狀態機器定義的目標資料夾做為此參數的值。

以下內容將AWS_STEP_FUNCTIONS目標物件的State machine definitions資料夾上傳到 Amazon S3 儲存貯體。AWS SCT使用您在[先決條件](#)步驟中存放在AWS服務設定檔中的 Amazon S3 儲存貯體。

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

該ConfigureStateMachine命令配置狀態機器。此指令最多可使用六個參數。請確定您使用下列清單中的前三個參數之一來定義目標範圍。

- filterName— 目標物件的篩選器名稱。您可以使用CreateFilter指令建立篩選。
- treePath— 目標物件的明確路徑。
- namePath— 指向特定目標物件的明確路徑。
- iamRole— IAM 角色的亞馬遜資源名稱 (ARN)，可讓您存取步驟機器。此為必要參數。

下列程式碼範例會設定AWS_STEP_FUNCTIONS使用 *role_name* IAM 角色中定義的狀態機器。

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

該ApplyToTarget命令將轉換後的代碼應用於目標服務器。若要執行此命令，請使用下列其中一個參數：filterName、treePath、或namePath來定義要套用的目標物件。

下列程式碼範例會將狀態機器套用於AWS Step Functions。

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

若要確保轉換後的程式碼產生與原始程式碼相同的結果，您可以使用AWS SCT延伸套件。這是一組模擬不支持的 Apache Oozie 函數的AWS Step Functions函數。AWS Lambda要安裝此擴展包，您可以使用CreateLambdaExtPack命令。

此指令最多可使用五個參數。請確定您使**Oozie2SF**用extPackId. 在這種情況下，AWS SCT為源Apache Oozie 函數創建一個擴展包。

- extPackId— 一組 Lambda 函數的唯一識別碼。此為必要參數。
- tempDirectory— AWS SCT 可儲存暫存檔案的路徑。此為必要參數。
- awsProfile— 您的AWS個人資料的名稱。
- lambdaExecRoles— 要用於 Lambda 函數之執行角色的亞馬遜資源名稱 (ARN) 清單。
- createInvokeRoleFlag— 布林旗標，指出是否要建立的執行角色AWS Step Functions。

若要安裝並使用擴充功能套件，請確定您提供必要的權限。如需詳細資訊，請參閱[在擴充功能套件中使用AWS Lambda函式的權限](#)。

如需有關Convert、SaveOnS3、ConfigureStateMachine和CreateLambdaExtPack命令的詳細資訊，請參閱 [AWS Schema Conversion ToolCLI 參考資料](#)。ApplyToTarget

執行您的 CLI 指令碼

完成 AWS SCT CLI 指令碼的編輯後，請將其儲存為.scts副檔名為的檔案。現在，您可以從AWS SCT安裝路徑的app文件夾運行腳本。為此，請使用下列命令。

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

在上述範例中，使用 CLI 指令碼將 *script_path* 取代為檔案的路徑。如需有關在中執行 CLI 指令碼的詳細資訊AWS SCT，請參閱[指令碼模式](#)。

阿帕奇奧齊節點，可AWS SCT以轉換為 AWS Step Functions

您可以使用AWS SCT將 Apache Oozie 動作節點和控制流節點轉換為AWS Step Functions.

支援的動作節點包括下列項目：

- 蜂巢行動
- 「隱藏 2」動作
- 火花行動

- MapReduce串流動作
- 爪哇動作
- DistCp 動作
- 豬, 行動
- 方向動作
- FS 動作
- 殼牌動作

支援的控制流程節點包括下列項目：

- 開始動作
- 結束動作
- 殺死動作
- 決策行動
- 叉動作
- 加入動作

搭配使用 AWS SCT 與 AWS DMS

使用AWS SCT複寫代理程式搭配AWS DMS

對於非常大的數據庫遷移，您可以使用AWS SCT複製代理 (aws-schema-conversion-tool-dms-agent) 將資料從現場部署資料庫複製到 Amazon S3 或AWS SnowballEdge 裝置。複寫代理程式可與 AWS DMS 搭配運作，而且複寫代理程式可在 AWS SCT 關閉時的背景中運作。

使用時AWS SnowballEdge,AWS SCT代理將數據複製到AWS Snowball裝置。然後將設備發送到AWS 並將資料載入到 Amazon S3 儲存儲體。在這段期間，AWS SCT 代理程式會繼續執行。代理程式接著會採用 Amazon S3 上的資料並將資料複製到目標端點。

如需詳細資訊，請參閱 [將資料從現場部署資料倉儲遷移到 Amazon Redshift](#)。

使用AWS SCT數據提取代理AWS DMS

InAWS SCT，您可以找到一個數據提取代理 (aws-schema-conversion-tool-extractor)，使從 Apache Cassandra 遷移到 Amazon DynamoDB 的作業更加簡單。卡桑德拉和 DynamoDB 是 NoSQL 數據庫，但它們在系統體繫結構和數據表示方面有所不同。您可以使用基於嚮導的工作流程 AWS SCT來自動化卡桑德拉到 DynamoDB 的遷移過程。AWS SCT與 整合AWS Database Migration Service(AWS DMS) 來執行實際遷移。

如需詳細資訊，請參閱 [將資料從現場部署資料倉儲遷移到 Amazon Redshift](#)。

在使用時提高日誌記錄級別AWS SCT取代為AWS DMS

您可以在使用AWS SCT取代為AWS DMS，例如，如果您需要使用AWSSupport。

安裝後AWS SCT和所需的驅動程序，請通過選擇AWS SCT圖示。如果看到更新通知，則可以選擇在項目完成之前或之後進行更新。如果自動項目窗口打開，請關閉窗口並手動創建項目。

在使用時提高日誌記錄級別AWS SCT取代為AWS DMS

1. 在設定菜單中，選擇全域設定。
2. 在 中全域設定窗口中，選擇日誌。
3. 適用於調試模式，選擇True。
4. 從消息級別部分中，您可以修改以下類型的日誌：

- 一般
- 載入器
- 解析器
- 打印機
- 解析程式
- 遙測
- 轉換器

默認情況下，所有消息級別都設置為Info。

5. 為要更改的任何消息級別類型選擇日誌記錄級別：

- 跟蹤 (最詳細的日誌記錄)
- 偵錯
- Info
- 警告
- 錯誤 (最詳細的日誌記錄)
- 嚴重
- 強制性

6. 選擇套用修改專案的設定。

7. 選擇確定關閉全域設定窗口。

將資料從現場部署資料倉儲遷移到 Amazon Redshift

您可以使用 AWS SCT 代理程式從現場部署資料倉儲擷取資料，並將其遷移到 Amazon Redshift。代理程式會擷取您的資料，並將資料上傳到 Amazon S3，或將大規模移轉上傳至 AWS Snowball Edge 裝置。然後，您可以使用 AWS SCT 代理程式將資料複製到 Amazon Redshift。

或者，您也可以使用 AWS Database Migration Service (AWS DMS) 將資料遷移到 Amazon Redshift。其優點 AWS DMS 是支援持續複寫 (變更資料擷取)。不過，若要提高資料移轉的速度，請 parallel 時使用多個 AWS SCT 代理程式。根據我們的測試，AWS SCT 代理程式遷移資料的速度比 AWS DMS 15—35% 的速度快。速度的差異在於資料壓縮、支援 parallel 移轉表格分割區，以及不同的組態設定。如需詳細資訊，請參閱[使用 Amazon Redshift 資料庫做為 AWS Database Migration Service 的目標](#)。

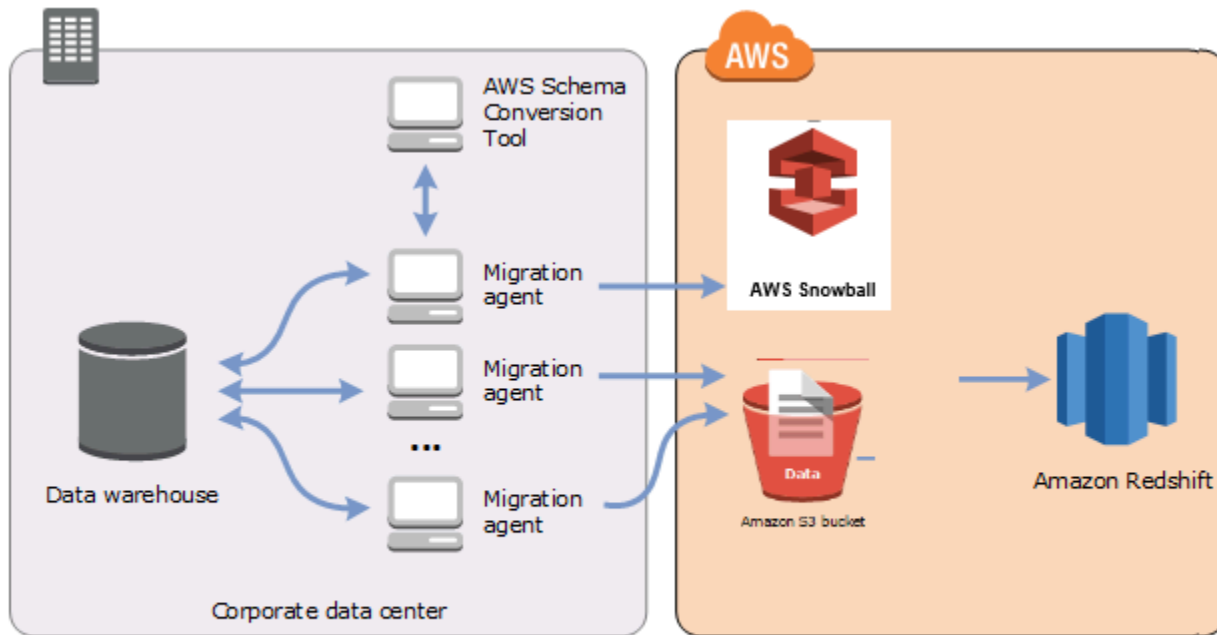
Amazon S3 是一種儲存和擷取服務。若要將物件存放在 Amazon S3 中，請上傳您要存放至 Amazon S3 儲存貯體的檔案。當您上傳檔案時，您可以設定物件及任何中繼資料的許可。

大規模移轉

大規模的資料移轉可以包含許多 TB 的資訊，而且可能會因為網路效能和必須移動的大量資料而減慢速度。AWS Snowball Edge 是一項 AWS 服務，您可以使用 AWS 自有的設備 faster-than-network 快速將資料傳輸到雲端。AWS Snowball 邊緣裝置最多可容納 100 TB 的資料。它使用 256 位加密和業界標準的可信賴平台模塊 (TPM) 來確保數據的安全性和完整 chain-of-custody 性。AWS SCT 適用於 AWS Snowball 邊緣設備。

使用 AWS SCT 和 AWS Snowball Edge 裝置時，您會分兩個階段移轉資料。首先，您可 AWS SCT 以使用本機處理資料，然後將該資料移至 AWS Snowball Edge 裝置。然後，您可以 AWS 使用 AWS Snowball Edge 程序將裝置傳送到，然後 AWS 自動將資料載入 Amazon S3 儲存貯體。接下來，當資料在 Amazon S3 上可用時，您可 AWS SCT 以使用將資料遷移到 Amazon Redshift。資料擷取代理程式可以在關閉時 AWS SCT 在背景運作。

下圖顯示所支援的案例。



目前支援資料擷取代理程式的來源資料倉儲如下：

- 突觸分析
- BigQuery
- 綠梅資料庫 (版本 4.3)
- Microsoft SQL 伺服器 (2008 年及更高版本)
- 尼泰扎 (版本 7.0.3 及更高版本)
- 甲骨文 (版本 10 及更高版本)
- 雪花 (版本 3)
- 太數據 (版本 13 及更高版本)
- 韋爾蒂卡 (版本 7.2.2 及更高版本)

如果您需要遵守聯邦資訊處理標準 (FIPS) 安全要求，可以連線到適用於 Amazon Redshift 的 FIPS 端點。FIPS 端點可在下列 AWS 區域中使用：

- 美國東部 (維吉尼亞北部) 區域 (維吉尼亞北部)
- 美國東部 (俄亥俄州) 區域 (中國東部)
- 美國西部 (加利福尼亞北部) 區域 (紅移-五月-西-1.amazonaws.com)

- [美國西部 \(奧勒岡\) 區域 \(西部-亞馬遜\)](#)

使用下列主題中的資訊，以了解如何使用資料擷取代理程式。

主題

- [使用資料擷取代理程式的前提](#)
- [安裝擷取代理](#)
- [設定擷取代理](#)
- [將擷取代理程式註冊到 AWS Schema Conversion Tool](#)
- [隱藏和復原 AWS SCT 代理程式的資訊](#)
- [在中建立資料移轉規則 AWS SCT](#)
- [更改提取器並從項目設置複製設置](#)
- [使用移轉前排序資料 AWS SCT](#)
- [建立、執行和監視 AWS SCT 資料萃取工作](#)
- [匯出和匯入資 AWS SCT 料萃取工作](#)
- [使用 AWS Snowball Edge 裝置擷取資料](#)
- [資料萃取工作輸出](#)
- [使用虛擬磁碟分割 AWS Schema Conversion Tool](#)
- [使用原生分割](#)
- [將 LOB 遷移到 Amazon Redshift](#)
- [資料擷取代理程式的最佳做法和疑難排](#)

使用資料擷取代理程式的前提

在您使用資料擷取代理程式之前，請將 Amazon Redshift 所需的許可作為目標新增至您的 Amazon Redshift 使用者。如需詳細資訊，請參閱 [亞馬遜紅移作為目標的許可](#)。

然後，存放您的 Amazon S3 儲存貯體資訊，並設定安全通訊端層 (SSL) 信任和金鑰存放區。

Amazon S3 設置

代理程式擷取資料後，他們會將資料上傳到您的 Amazon S3 儲存貯體。在繼續之前，您必須提供登入資料以連接到您的 AWS 帳戶和 Amazon S3 儲存貯體。您可以將認證和值區資訊儲存在全域應用程式

設定中的設定檔中，然後將設定檔與 AWS SCT 專案建立關聯。如有必要，請選擇「全域設定」來建立新的設定檔。如需詳細資訊，請參閱 [將AWS服務設定檔儲存在AWS SCT](#)。

若要將資料遷移到目標 Amazon Redshift 資料庫，AWS SCT 資料擷取代理程式需要您存取 Amazon S3 儲存貯體的權限。若要提供此權限，請使用下列政策建立 AWS Identity and Access Management (IAM) 使用者。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/*",
        "arn:aws:s3:::bucket_name"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Action": [
        "iam:GetUser"
      ],

```

```
        "Resource": [
            "arn:aws:iam::<111122223333:user/DataExtractionAgentName"
        ],
        "Effect": "Allow"
    }
}
```

在上述範例中，請`bucket_name`以 Amazon S3 儲存貯體的名稱取代。然後，`111122223333:user/DataExtractionAgentName`以 IAM 使用者的名稱取代。

假設 IAM 角色

為了獲得額外的安全性，您可以使用 AWS Identity and Access Management (IAM) 角色存取 Amazon S3 儲存貯體。若要這麼做，請在沒有任何許可的情況下為資料擷取代理程式建立 IAM 使用者。然後，建立啟用 Amazon S3 存取的 IAM 角色，並指定可擔任此角色的服務和使用者清單。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 角色](#)。

設定 IAM 角色以存取您的 Amazon S3 儲存貯體

1. 建立新的 IAM 使用者。對於使用者認證，請選擇程式設計存取類型。
2. 設定主機環境，以便您的資料擷取代理程式可以擔任 AWS SCT 提供的角色。請確定您在上一個步驟中設定的使用者可讓資料擷取代理程式使用認證提供者鏈結。如需詳細資訊，請參閱 AWS SDK for Java 開發人員指南中的 [使用認證](#)。
3. 建立可存取 Amazon S3 儲存貯體的新 IAM 角色。
4. 修改此角色的「信任」區段，以信任您之前建立的使用者，以擔任該角色。在下列範例中，請以您的使用者名稱取`111122223333:user/DataExtractionAgentName`代。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<111122223333:user/DataExtractionAgentName"
  },
  "Action": "sts:AssumeRole"
}
```

5. 修改此角色的信任區段以信任`redshift.amazonaws.com`以擔任該角色。

```
{
  "Effect": "Allow",
```

```
"Principal": {
  "Service": [
    "redshift.amazonaws.com"
  ]
},
"Action": "sts:AssumeRole"
}
```

6. 將此角色附加到您的 Amazon Redshift 叢集。

現在，您可以在中執行資料擷取代理程式 AWS SCT。

假設使用 IAM 角色時，數據遷移按以下方式工作。資料擷取代理程式會啟動並使用認證提供者鏈結取得使用者認證。接下來，您可以在中建立資料移轉任務 AWS SCT，然後指定要承擔的資料擷取代理程式的 IAM 角色，然後啟動工作。AWS Security Token Service (AWS STS) 產生臨時登入資料以存取 Amazon S3。資料擷取代理程式使用這些登入資料將資料上傳到 Amazon S3。

然後，AWS SCT 為 Amazon Redshift 提供 IAM 角色。反過來，Amazon Redshift 獲得新的臨時登入資料 AWS STS 以存取 Amazon S3。Amazon Redshift 使用這些登入資料將資料從 Amazon S3 複製到您的 Amazon Redshift 表。

安全性設定

AWS Schema Conversion Tool 和擷取代理程式可透過安全通訊端層 (SSL) 進行通訊。若要啟用 SSL，請設定信任存放區和金鑰存放區。

設定與擷取代理程式的安全通訊

1. 啟動 AWS Schema Conversion Tool.
2. 開啟 [設定] 功能表，然後選擇 [全域設定]。Global settings (全域設定) 對話方塊隨即出現。
3. 選擇 Security (安全性)。
4. 選擇 [產生信任和金鑰存放區]，或選擇 [選取現有信任存放區]

如果您選擇「產生信任與金鑰存放區」，請指定信任和金鑰存放區的名稱和密碼，以及產生檔案的位置路徑。您可以在稍後的步驟中使用這些檔案。

如果您選擇 [選取現有的信任存放區]，請指定信任和金鑰存放區的密碼和檔案名稱。您可以在稍後的步驟中使用這些檔案。

5. 指定信任存放區和金鑰存放區之後，請選擇 [確定] 關閉 [全域設定] 對話方塊。

設定資料擷取代理程式的環境

您可以在單一主機上安裝多個資料擷取代理程式。但是，我們建議您在一台主機上執行一個資料擷取代理程式。

若要執行資料擷取代理程式，請確定您使用的主機至少具有四個 vCPUs 和 32 GB 記憶體。此外，請將可用的最小記憶體設定 AWS SCT 為至少四 GB。如需詳細資訊，請參閱 [設定其他記憶體](#)。

最佳組態和代理程式主機數量取決於每個客戶的特定情況。請務必考慮移轉的資料量、網路頻寬、擷取資料的時間等因素。您可以先執行概念驗證 (PoC)，然後根據此 PoC 的結果設定資料擷取代理程式和主機。

安裝擷取代理

建議您在個別的電腦上 (與執行 AWS Schema Conversion Tool 的電腦分開)，安裝多個擷取代理程式。

目前支援擷取代理程式的作業系統如下：

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- 版本 14.04 及更高版本

請使用下列程序來安裝擷取代理程式。針對要安裝擷取代理程式的每部電腦，重複這個程序。

安裝擷取代理程式

1. 如果您尚未下載 AWS SCT 安裝程式檔案，請按照中的指示 [安裝、驗證和更新 AWS SCT](#) 進行下載。包含 AWS SCT 安裝程式檔案的 .zip 檔案也包含解壓縮代理程式安裝程式檔案。
2. 下載並安裝最新版本的 Amazon Corretto 11。如需詳細資訊，請參閱 [Amazon Corretto 11 使用者指南中的下載](#)。
3. 在名為 agents (代理程式) 的子資料夾中，找出擷取代理程式的安裝程式檔案。針對各種電腦作業系統，用來安裝擷取代理程式的正確檔案如下所示。

作業系統	檔案名稱
Microsoft Windows	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .msi

作業系統	檔案名稱
RHEL	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm
Ubuntu Linux	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb

- 透過將安裝程式檔案複製到新電腦，在另一台電腦上安裝解壓縮代理程式。
- 執行安裝程式檔案。使用適用於您作業系統的指示，如下所示。

作業系統	安裝說明
Microsoft Windows	按兩下檔案來執行安裝程式。
RHEL	<p>在您下載或移動檔案的目標資料夾中執行下列命令。</p> <pre>sudo rpm -ivh aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm sudo ./sct-extractor-setup.sh --config</pre>
Ubuntu Linux	<p>在您下載或移動檔案的目標資料夾中執行下列命令。</p> <pre>sudo dpkg -i aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

- 選擇 [下一步]，接受授權合約，然後選擇 [下一步]。
- 輸入安裝 AWS SCT 資料擷取代理程式的路徑，然後選擇 [下一步]。
- 選擇 [安裝] 以安裝資料擷取代理程式。

AWS SCT 安裝您的資料擷取代理程式。若要完成安裝，請設定您的資料擷取代理程式。AWS SCT 會自動啟動組態設定程式。如需詳細資訊，請參閱 [設定擷取代理](#)。

- 選擇「完成」以在規劃資料萃取代理程式之後關閉安裝精靈。

設定擷取代理

請使用下列程序來設定擷取代理程式。在已安裝擷取代理程式的每部電腦上，重複這個程序。

設定擷取代理程式

1. 啟動組態設定程式：

- 在 Windows 中，在安裝資料擷取代理程式期間，會自動 AWS SCT 啟動組態設定程式。

根據需要，您可以手動啟動安裝程序。若要這麼做，請在視窗中執行該 `ConfigAgent.bat` 檔案。您可以在安裝代理程式的資料夾中找到此檔案。

- 在 RHEL 和 Ubuntu 中，從安裝代理程式的位置執行 `sct-extractor-setup.sh` 檔案。

安裝程式會提示您輸入資訊。針對每個提示，都會顯示預設值。

2. 在每個提示下接受預設值，或輸入新值。

指定下列資訊：

- 針對監聽通訊埠，輸入代理程式監聽的連接埠號碼。
- 針對 [新增來源供應商]，輸入 `yes`，然後輸入您的來源資料倉儲平台。
- 對於 JDBC 驅動程式，請輸入安裝 JDBC 驅動程式的位置。
- 對於「工作資料夾」，輸入 AWS SCT 資料萃取代理程式將儲存萃取資料的路徑。工作資料夾可以在與代理程式不同的電腦上，而且單一工作資料夾可由位在不同電腦上的多個代理程式共用。
- 針對「啟用 SSL 通訊」，輸入 `yes`。
- 在金鑰存放區中，輸入金鑰存放區檔案的位置。
- 對於金鑰存放區密碼，請輸入金鑰存放區的密碼。
- 在 [啟用用戶端 SSL 驗證] 中，輸入 `yes`。
- 在「信任存放區」中，輸入信任存放區檔案的位置。
- 在「信任存放區密碼」中，輸入信任存放區的密碼。

安裝程式會更新擷取代理程式的設定檔。設定檔名為 `settings.properties`，位在擷取代理程式的安裝位置。

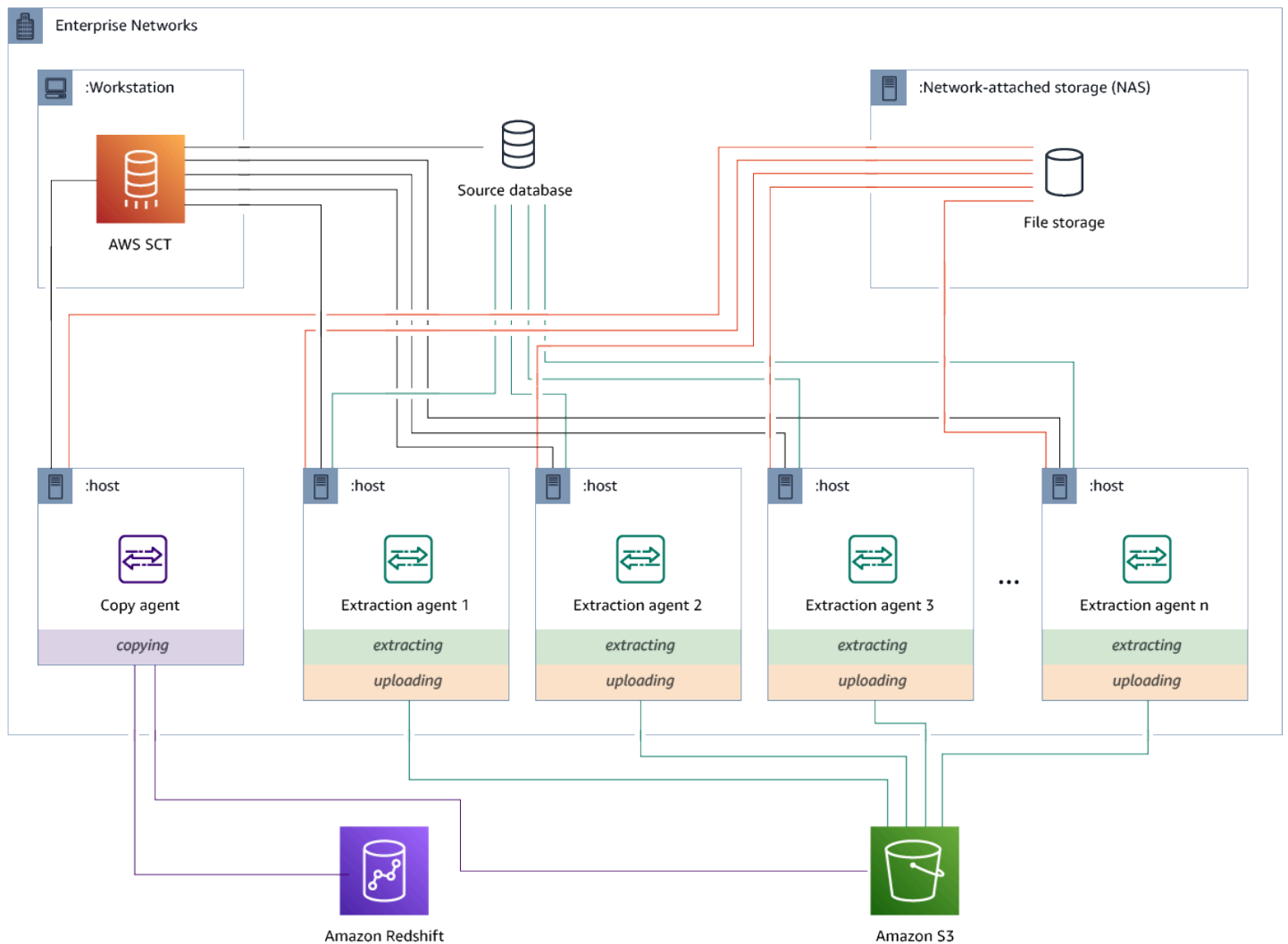
以下是設定檔範例。

```
$ 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
```

若要變更組態設定，您可以使用文字編輯器編輯settings.properties檔案，或再次執行代理程式組態。

使用專用複製代理程式安裝和設定擷取代理

您可以在具有共用儲存區和專用複製代理程式的組態中安裝擷取代理程式。下圖說明此案例。



當來源資料庫伺服器最多支援 120 個連線，且您的網路已連接充足的儲存空間時，該組態可能很有用。使用下列程序來設定具有專用複製代理程式的擷取代理程式。

安裝和設定擷取代理程式和專用複製代理程式

1. 請確定所有解壓縮代理程式的工作目錄都使用共用儲存裝置上的相同資料夾。
2. 依照中 [安裝擷取代理](#) 的步驟安裝擷取程式代理程式。
3. 依照中的步驟設定擷取代理程式 [設定擷取代理](#)，但僅指定來源 JDBC 驅動程式。
4. 按照中的步驟設定專用複製代理程式 [設定擷取代理](#)，但僅指定 Amazon Redshift JDBC 驅動程式。

啟動萃取劑

請使用下列程序來啟動擷取代理程式。在已安裝擷取代理程式的每部電腦上，重複這個程序。

擷取代理程式是做為接聽程式。當您使用此程序來啟動代理程式時，代理程式即開始接聽指示。在稍後的章節中，您會傳送指示給代理程式，以從資料倉儲擷取資料。

啟動擷取代理程式

- 在已安裝擷取代理程式的電腦中，針對您的作業系統執行下列命令。

作業系統	啟動命令
Microsoft Windows	按兩下 StartAgent.bat 批次檔。
RHEL	在安裝代理程式的資料夾路徑中，執行下列命令： <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	在安裝代理程式的資料夾路徑中，執行下列命令。使用 Ubuntu 版本適用的命令。 Ubuntu 14.04 : <code>sudo initctl <i>start</i> sct-extractor</code> 版本 15.04 及更高版本 : <code>sudo systemctl <i>start</i> sct-extractor</code>

若要查看代理程式的狀態，請執行相同的命令，但將 `start` 取代為 `status`。

若要停止代理程式，請執行相同的命令，但將 `start` 取代為 `stop`。

將擷取代理程式註冊到 AWS Schema Conversion Tool

您可以使用來管理擷取代理程式 AWS SCT。擷取代理程式是做為接聽程式。當他們收到指令時 AWS SCT，他們會從您的數據倉庫中提取數據。

使用下列程序在 AWS SCT 專案中註冊擷取代理程式。

註冊擷取代理程式

1. 啟動 AWS Schema Conversion Tool，然後開啟專案。
2. 開啟 [檢視] 功能表，然後選擇 [資料移轉檢視 (其他)]。Agents (代理程式) 索引標籤隨即出現。如果您先前已註冊代理程式，則 AWS SCT 會在索引標籤頂端的網格中顯示代理程式。
3. 選擇註冊。

在專案中註冊代理程式之後，您無法在不同 AWS SCT 專案中註冊相同的代理程式。如果您不再在專案中使用代理 AWS SCT 程式，您可以取消註冊該代理程式。然後可以再將其註冊於不同的專案。

4. 選擇 Redshift 資料代理程式，然後選擇 [確定]。
5. 在對話方塊的「連線」頁籤上輸入您的資訊：
 - a. 在說明中，輸入代理程式的說明。
 - b. 在「主機名稱」中，輸入代理程式電腦的主機名稱或 IP 位址。
 - c. 在連接埠中，輸入代理程式偵聽的通訊埠號碼。
 - d. 選擇「註冊」以在您的 AWS SCT 專案中註冊代理程式。
6. 重複上述步驟，以向 AWS SCT 專案註冊多個代理程式。

隱藏和復原 AWS SCT 代理程式的資訊

AWS SCT 代理程式會加密大量資訊，例如使用者金鑰信任存放區的密碼、資料庫帳 AWS 戶、帳戶資訊和類似項目。它是使用一個名為 `seed.dat` 的特殊檔案來執行此作業。依預設，代理程式會在第一個設定代理程式之使用者的工作資料夾中建立此檔案。

由於不同的使用者都可以設定及執行代理程式，因此 `seed.dat` 的路徑會存放在 `settings.properties` 檔案的 `{extractor.private.folder}` 參數中。代理程式啟動時，可以使用此路徑來尋找 `seed.dat` 檔案，以存取其作用之資料庫的金鑰信任存放區資訊。

您可能會需要復原代理程式在下列情況下存放的密碼：

- 如果使用者遺失 `seed.dat` 檔案，而 AWS SCT 代理程式的位置和通訊埠並未變更。
- 如果使用者遺失 `seed.dat` 檔案，且 AWS SCT 代理程式的位置和通訊埠已變更。在此情況下，通常會發生變更，因為代理程式已遷移到另一個主機或連接埠，而 `seed.dat` 檔案中的資訊不再有效。

在這些情況下，如果啟動代理程式時未使用 SSL，則代理程式會啟動，然後存取先前建立的代理程式儲存體。接著它會變成 Waiting for recovery (等待復原) 狀態。

不過，在這些情況下，如果啟動代理程式時有使用 SSL，您就無法將其重新啟動。這是因為代理程式無法將存放在 `settings.properties` 檔案中之憑證的密碼解密。在這種類型的啟動作業中，代理程式無法啟動。類似下列的錯誤會寫入日誌中：「代理程式無法在啟用 SSL 模式的情況下啟動。請重新設定代理程式。原因：金鑰存放區的密碼不正確。」

若要修正此問題，請建立新的代理程式，並將該代理程式設定為使用現有的密碼來存取 SSL 憑證。若要完成此操作，請遵循下列程序：

執行此程序之後，代理程式應該會執行並移至 [等待復原] 狀態。AWS SCT 會自動將必要的密碼傳送至處於等待復原狀態的代理程式。當代理程式取得密碼後，就會重新啟動任何任務。側面不需要進一步的用戶操 AWS SCT 作。

重新設定代理程式並還原密碼來存取 SSL 憑證

1. 安裝新的 AWS SCT 代理程式並執行設定。
2. 將 `instance.properties` 檔案的 `agent.name` 屬性，變更為已為其建立儲存體之代理程式的名稱，讓新的代理程式使用現有的代理程式儲存體。

`instance.properties` 檔案存放在代理程式的私有資料夾中，其使用下列命名慣例：`{output.folder}\dmt\{hostName}_{portNumber}\`。

3. 將 `{output.folder}` 的名稱變更為先前代理程式的輸出資料夾名稱。

此時，仍 AWS SCT 在嘗試訪問舊主機和端口上的舊提取器。因此，無法存取的擷取器會發生 FAILED (失敗) 狀態。然後您就可以變更主機和連接埠。

4. 使用 `Modify` 命令，將請求流量重新導向至新的代理程式，藉以修改舊代理程式的主機及/或連接埠。

何時 AWS SCT 可以偵測新代理程式，AWS SCT 會收到「等待從代理程式復原」狀態。AWS SCT 然後自動復原代理程式的密碼。

使用代理程式儲存體的每個代理程式都會更新一個名為 `storage.lck` 的特殊檔案 (位於 `{output.folder}\{agentName}\storage\`)。這個檔案包含代理程式的網路 ID，以及直到儲存體鎖定的時間。當代理程式使用代理程式儲存體時，會更新 `storage.lck` 檔案，並將儲存體的每 5 分鐘租賃期延長 10 分鐘。在租賃到期之前，其他執行個體都不能使用此代理程式儲存體。

在中建立資料移轉規則 AWS SCT

使用擷取資料之前 AWS Schema Conversion Tool，您可以設定篩選器，以減少擷取的資料量。您可以使用 WHERE 子句來減少擷取的資料，以建立資料移轉規則。例如，您可以編寫從單一資料表選取資料的 WHERE 子句。

您可以建立資料移轉規則，並將篩選儲存為專案的一部分。開啟專案後，請使用下列程序建立資料移轉規則。

若要建立資料移轉規則

1. 開啟 [檢視] 功能表，然後選擇 [資料移轉檢視 (其他)]。
2. 選擇 [資料遷移規則]，然後選擇 [新增規則]。
3. 設定您的資料移轉規則：
 - a. 在名稱中，輸入資料移轉規則的名稱。
 - b. 在綱要名稱類似的位置中，輸入要套用至結構描述的篩選。此篩選條件是使用 LIKE 子句來評估 WHERE 子句。若要選擇一個綱要，請輸入完全相同的綱要名稱。若要選擇多個綱要，請使用「%」字元作為萬用字元，以符合綱要名稱中任意數目的字元。
 - c. 對於類似的表格名稱，請輸入要套用至表格的篩選條件。此篩選條件是使用 LIKE 子句來評估 WHERE 子句。若要選擇一個表格，請輸入確切的名稱。若要選擇多個資料表，請使用「%」字元作為萬用字元，以符合資料表名稱中任意數目的字元。
 - d. 在 Where 子句中，輸入要篩選資料的 WHERE 子句。
4. 設定篩選條件之後，選擇 Save (儲存)，以儲存您的篩選條件，或選擇 Cancel (取消)，以取消變更。
5. 新增、編輯和刪除篩選器完成後，請選擇 [全部儲存] 以儲存所有變更。

若要關閉篩選條件，但不將其刪除，請使用切換圖示。若要複製現有的篩選條件，請使用複製圖示。若要刪除現有的篩選條件，請使用刪除圖示。若要儲存您對篩選器所做的任何變更，請選擇 [全部儲存]。

更改提取器並從項目設置複製設置

在的專案設定視窗中 AWS SCT，您可以選擇資料擷取代理程式和 Amazon Redshift COPY 命令的設定。

若要選擇這些設定，請選擇 [設定]、[專案設定]，然後選擇 [資料移轉]。您可以在此編輯擷取設定、Amazon S3 設定和複製設定。

使用下表中的說明提供有關萃取設定的資訊。

對於此參數	執行此作業
壓縮格式	指定輸入檔案的壓縮格式。選擇下列其中一個選項：GZIP、BZIP2、ZSTD 或無壓縮。
分隔符號	指定用於分隔輸入檔案中欄位的 ASCII 字元。不支援非列印字元。
空值作為一個字符串	如果您的資料包含空結束字元，請開啟此選項。如果關閉此選項，Amazon Redshift COPY 命令會將空值視為記錄的結尾，並結束載入程序。
排序策略	使用排序從失敗點重新啟動解壓縮。選擇下列其中一個排序策略：在第一次失敗後使用排序 (建議選項)、如果可能，請使用排序或永不使用排序。如需詳細資訊，請參閱 the section called “資料排序” 。
源臨時模式	在來源資料庫中輸入綱要的名稱，擷取代理程式可在此建立暫存物件。
輸出檔案大小 (MB)	輸入上傳到 Amazon S3 的檔案大小 (以 MB 為單位)。
Snowball 出檔案大小 (MB)	輸入上傳目標檔案的大小 (以 MB 為單位) AWS Snowball。檔案的大小可以有 1 到 1,000 MB。
使用自動分割。如果是綠色和 Netezza，請輸入支援資料表的最小大小 (以 MB 為單位)	開啟此選項以使用資料表分割，然後輸入 Greenplum 和 Netezza 來源資料庫要分割的表格大小。對於 Oracle 到 Amazon Redshift 的遷移，您可以將此欄位保留空白，因為 AWS SCT 會為所有分區資料表建立子任務。
提取 LOB	開啟此選項可從來源資料庫萃取大型物件 (LOB)。LOB 包括 BLOB、CLOB、NCLOBs、XML 檔案等。對於每個 LOB，AWS SCT 擷取代理程式都會建立一個資料檔。
Amazon S3 存儲桶 LOB 文件夾	輸入 AWS SCT 擷取代理程式儲存 LOB 的位置。

對於此參數	執行此作業
將 RTRIM 套用至字串資料行	開啟此選項可從擷取的字串結尾修剪一組指定的字元。
上傳到 Amazon S3 後將文件保留在本地	開啟此選項可在資料擷取代理程式上傳檔案到 Amazon S3 後，將檔案保留在本機電腦上。

使用下表中的說明提供 Amazon S3 設定的相關資訊。

對於此參數	執行此作業
使用代理	開啟此選項以使用代理伺服器將資料上傳到 Amazon S3。然後選擇數據傳輸協議，輸入主機名，端口，用戶名和密碼。
端點類型	選擇 FIPS 以使用聯邦資訊處理標準 (FIPS) 端點。選擇 VPCE 以使用虛擬私有雲端 (VPC) 端點。然後，針對 VPC 端點，輸入 VPC 端點的網域名稱系統 (DNS)。
將文件複製到亞馬遜紅移後保留在 Amazon S3 上	開啟此選項可在將這些檔案複製到 Amazon Redshift 之後，將擷取的檔案保留在 Amazon S3 上。

使用下表中的指示來提供「複製」設定的資訊。

對於此參數	執行此作業
最大錯誤計數	輸入載入錯誤的數目。作業達到此限制之後，AWS SCT 資料擷取代理程式就會結束資料載入程序。預設值為 0，這表示無論失敗為何，AWS SCT 資料擷取代理程式都會繼續載入資料。
取代無效的 UTF-8 字元	開啟此選項，以指定的字元取代無效的 UTF-8 字元，並繼續資料載入作業。
使用空白作為空值	開啟此選項可將包含空格字元的空白欄位載入為 null。
使用空值作為空值	開啟此選項可將空白 CHAR 和 VARCHAR 欄位載入為 null。
截斷資料欄	開啟此選項可截斷欄中的資料以符合資料類型規格。

對於此參數	執行此作業
自動壓縮	開啟此選項可在複製作業期間套用壓縮編碼。
自動統計數據刷	開啟此選項可在複製作業結束時重新整理統計資料。
載入前檢查檔案	開啟此選項可在將資料檔案載入 Amazon Redshift 之前先驗證這些檔案。

使用移轉前排序資料 AWS SCT

在移轉之前對資料進行排序可 AWS SCT 提供一些好處。如果您先排序資料，則 AWS SCT 可以在失敗後的最後一個儲存點重新啟動擷取代理程式。此外，如果您要將資料遷移到 Amazon Redshift，並且先對資料進行排序，AWS SCT 可以更快地將資料插入 Amazon Redshift。

這些好處與如何 AWS SCT 建立資料萃取查詢有關。在某些情況下，AWS SCT 會在這些查詢中使用 DENSE_RANK 分析函數。但是，DENSE_RANK 可以使用大量時間和服務器資源對從提取產生的數據集進行排序，因此，如果沒有它，則 AWS SCT 可以。

若要在移轉前對資料進行排序 AWS SCT

1. 開啟專 AWS SCT 案。
2. 開啟物件的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「建立本機工作」。
3. 選擇「進階」標籤，然後針對「排序策略」選擇一個選項：
 - 永不使用排序 — 擷取代理程式不使用 DENSE_RANK 分析函式，如果發生失敗，則會從頭開始重新啟動。
 - 如果可能，請使用排序 — 如果資料表具有主索引鍵或唯一條件約束，擷取代理程式會使用 DENSE_RANK。
 - 第一次失敗後使用排序 (建議) — 擷取代理程式會先嘗試取得資料，而不使用 DENSE_RANK。如果第一次嘗試失敗，擷取代理程式就會使用 DENSE_RANK 來重建查詢，並且在發生失敗時，保留其位置。

Create Local task

General | **Advanced** | Source server | AWS S3 settings | Source SSL settings

Extraction settings

Delimiter character: |

Compression format: GZIP

NULL value as a string

Sorting strategy: Use sorting after first fail (recommen...)

Source temp schema:

Out file size (in MB): 10

Apply RTRIM to string columns

Keep files locally after upload to AWS S3

Use subtasks auto-balancing between agents

Freezing interval: 10

Copy settings

Maximum error count: 0

Replace invalid UTF-8 character: ?

Use blank as null value
BLANKSASNULL: This option loads blank fields, which consist of only white space characters, as NULL. The default behavior, without this option, is to load the space characters as is.

Use empty as null value
EMPTYASNULL: This option indicates that Amazon Redshift should load empty CHAR and VARCHAR fields as NULL.

Truncate columns
TRUNCATECOLUMNS: This option truncates data in columns to the appropriate number of characters so that it fits the column specification. This option applies only to columns with a VARCHAR or CHAR data type, and rows 4 MB or less in size.

Automatic compression
COMPUPDATE: This option controls whether compression encodings are automatically

Test Task | Cancel | Create

4. 依下列說明設定額外的參數，然後選擇 Create (建立) 來建立您的資料擷取任務。

建立、執行和監視 AWS SCT 資料萃取工作

使用下列程序來建立、執行及監控資料擷取任務。

將任務指派給代理程式並遷移資料

1. 在中 AWS Schema Conversion Tool，轉換結構定義後，從專案的左側面板中選擇一或多個表格。

您可以選擇所有資料表，但基於效能考量，建議您不要這麼做。建議您根據資料倉儲中的資料表大小，為多個資料表建立多個任務。

2. 開啟每個表格的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「建立工作」。建立本機工作對話方塊隨即開啟。
3. 在工作名稱中，輸入工作的名稱。
4. 針對移轉模式，選擇下列其中一項：
 - 僅擷取 — 萃取資料，並將資料儲存至本端工作資料夾。
 - 擷取和上傳 — 擷取資料，然後將資料上傳到 Amazon S3。
 - 擷取、上傳和複製 — 擷取資料、將資料上傳到 Amazon S3，然後將資料複製到 Amazon Redshift 資料倉儲。
5. 針對「加密」類型，選擇下列其中一項：
 - 無 — 關閉整個資料移轉程序的資料加密。
 - CSE_SK — 使用用戶端加密搭配對稱金鑰來移轉資料。AWS SCT 自動產生加密金鑰，並使用安全通訊端層 (SSL) 將其傳輸至資料擷取代理程式。AWS SCT 在資料移轉期間不會加密大型物件 (LOB)。
6. 選擇 Extract LOBs (擷取 LOB)，以擷取大型物件。如果您不需要擷取大型物件，可以清除此核取方塊。這麼做可以減少您擷取的資料量。
7. 若要查看工作的詳細資訊，請選擇 [啟用工作記錄]。您可以使用任務日誌來進行問題偵錯。

如果您啟用任務記錄功能，請選擇您要查看的詳細層級。各層級如下所示，每個層級都包含上一個層級中的所有訊息：

- ERROR— 細節的最小量。
 - WARNING
 - INFO
 - DEBUG
 - TRACE— 最大量的細節。
8. 若要從中匯出資料 BigQuery，請 AWS SCT 使用 GCP 儲存貯體資料夾。在此資料夾中，資料擷取代理程式會儲存您的來源資料。

- 若要輸入 GCP 儲存貯體資料夾的路徑，請選擇 [進階]。對於 Google CS 值區資料夾，請輸入值區名稱和資料夾名稱。
- 若要擔任資料擷取代理程式使用者的角色，請選擇 Amazon S3 設定。對於 IAM 角色，請輸入要使用的角色名稱。針對「區域」，選 AWS 區域 擇此角色的。
 - 選擇 [測試任務] 以確認您可以連線到工作資料夾、Amazon S3 儲存貯體和 Amazon Redshift 資料倉儲。此驗證取決於您選擇的遷移模式。
 - 選擇 Create (建立)，以建立任務。
 - 針對您要遷移的所有資料，重複上述步驟來建立任務。

執行及監控任務

- 對於「檢視」，選擇「資料移轉視圖」 Agents (代理程式) 索引標籤隨即出現。
- 選擇 Tasks (任務) 索引標籤。您的任務會出現在網格頂端，如下所示。您可以在頂端網格中查看任務的狀態，以及在底部網格中查看其子任務的狀態。

Name	Extract	Upload	Copy
CUSTOMER	0%		
LINEORDER_100K	0%		
LINEORDER_150K	0%		
LINEORDER_1M	0%		
LocalTask 2	100%	100%	
CUSTOMER	100%	100%	
LINEORDER_100K	100%	100%	
LINEORDER_150K	100%	100%	
LocalTask 3	100%	100%	0%
LINEORDER_100K	100%	100%	0%

Download log All migration Tasks Resume Stop Restart Reset Delete Replace Refresh all Refresh selected

Properties Processing details

Extracting

Data

100%

- 在頂端網格中選擇任務，並將其展開。依據您選擇的遷移模式，您會看到任務區分成 Extract (擷取)、Upload (上傳) 和 Copy (複製)。
- 針對某項任務選擇 Start (啟動)，以啟動該任務。您可以在任務進行時，監控其狀態。子任務會平行執行。擷取、上傳和複製也會平行執行。
- 如果在設定任務時，已啟用記錄功能，則可檢視日誌：

- a. 選擇 [下載記錄]。隨即出現訊息顯示包含該日誌檔的資料夾名稱。請關閉訊息。
- b. Task details (任務詳細資訊) 索引標籤中會顯示一個連結。請選擇該連結，以開啟包含日誌檔的資料夾。

您可以關閉 AWS SCT，代理程式和工作會繼續執行。您可以 AWS SCT 稍後重新開啟以檢查工作狀態並檢視工作記錄。

您可以將資料萃取工作儲存到本端磁碟，並使用匯出和匯入將其還原到相同或其他專案。若要匯出工作，請確保在專案中至少建立了一個萃取工作。您可以匯入單一萃取工作或在專案中建立的所有工作。

匯出萃取工作時，AWS SCT 會為該工作建立單獨的 .xml 檔案。該 .xml 文件存儲該任務的元數據信息，例如任務屬性，描述和子任務。 .xml 檔案不包含有關處理解壓縮工作的資訊。匯入工作時，會重新建立類似下列的資訊：

- 任務進度
- 子工作和階段狀態
- 按子任務和階段分配提取代理
- 任務和子任務 ID
- Task name (任務名稱)

匯出和匯入資料萃取工作

您可以快速保存一個項目中的現有任務，並使用導出和導入將其還原到另一個項目（或同一項目）中。使用下列程序匯出和匯入資料萃取工作。

匯出和匯入資料萃取工作的步驟

1. 對於「檢視」，選擇「資料移轉視圖」Agents (代理程式) 索引標籤隨即出現。
2. 選擇 Tasks (任務) 索引標籤。您的工作會列在出現的網格中。
3. 選擇位於任務列表下方右下角的三個垂直對齊點（省略號圖標）。
4. 從彈出式選單中選擇「匯出工作」。
5. 選擇您要 AWS SCT 放置工作匯出 .xml 檔案的資料夾。

AWS SCT 建立檔案名稱格式為的工作匯出檔案 *TASK-DESCRIPTION_TASK-ID.xml*。

6. 選擇任務列表下右下角的三個垂直對齊點（省略號圖標）。

7. 從彈出式選單中選擇「匯入工作」。

您可以將萃取任務匯入至連接至來源資料庫的專案，並且專案至少具有一個作用中的已註冊萃取代理程式。

8. 為匯出的萃取工作選取 .xml 檔案。

AWS SCT 從檔案中取得擷取工作的參數、建立工作，並將工作新增至擷取代理程式。

9. 重複這些步驟以匯出和匯入其他資料萃取工作。

在此過程結束時，您的匯出和匯入已完成，並且您的資料萃取工作已準備就緒可供使用。

使用 AWS Snowball Edge 裝置擷取資料

使用 AWS SCT 和 AWS Snowball Edge 的過程有幾個步驟。遷移涉及本機任務，其中 AWS SCT 使用資料擷取代理程式將資料移至 AWS Snowball Edge 裝置，然後執行中繼動作，將資料從 AWS Snowball Edge 裝置 AWS 複製到 Amazon S3 儲存貯體。此程序完成將資料從 Amazon S3 儲存貯體 AWS SCT 載入到 Amazon Redshift。

本概觀後面的各節提供了這些工作的 step-by-step 指南。此程序假設您已 AWS SCT 安裝，並且已在專用機器上設定並註冊資料擷取代理程式。

執行以下步驟，以使用 AWS Snowball Edge 將資料從本端資料倉庫移轉至 AWS 資料倉庫。

1. 使用 AWS Snowball 主控台建立 AWS Snowball Edge 工作。
2. 使用本機專用 Linux 機器解除鎖定 AWS Snowball 邊緣裝置。
3. 在中建立新專案 AWS SCT。
4. 安裝和設定資料擷取代理程式。
5. 建立並設定要使用的 Amazon S3 儲存貯體的權限。
6. 將 AWS Snowball 工作匯入至您的 AWS SCT 專案。
7. 在中註冊您的資料擷取代理程式 AWS SCT。
8. 在中建立本機工作 AWS SCT。
9. 執行並監視中的資料移轉工作 AWS SCT。

S 使用 AWS SCT 和 AWS Snowball Edge 移轉資料的tep-by-step 程序

以下小節提供遷移步驟的詳細資訊。

步驟 1：建立 AWS Snowball 邊緣工作

依照 Edge AWS Snowball 開發人員指南中建立 [AWS Snowball Edge Job](#) 一節中概述的 AWS Snowball 步驟建立工作。

步驟 2：解鎖邊 AWS Snowball 緣設備

從安裝 AWS DMS 代理程式的機器執行解除鎖定並提供認證給 Snowball Edge 裝置的命令。透過執行這些命令，您可以確定 AWS DMS 代理程式呼叫已連線至 AWS Snowball Edge 裝置。如需解除鎖定 AWS Snowball Edge 裝置的詳細資訊，請參閱 [解除鎖定 Snowball 邊緣](#)。

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

步驟 3：建立新 AWS SCT 專案

接下來，創建一個新 AWS SCT 項目。

若要在中建立新專案 AWS SCT

1. 啟動 AWS Schema Conversion Tool. 在 [檔案] 功能表上，選擇 [新增專案]。[新增專案] 對話方塊隨即出現。
2. 輸入專案的名稱，這會儲存在本機電腦上。
3. 輸入本機專案檔案的位置。
4. 選擇「確定」以建立 AWS SCT 專案。
5. 選擇 [新增來源]，將新的來源資料庫新增至您的 AWS SCT 專案。
6. 選擇添加目標以在項目中添加新的 AWS SCT 目標平台。
7. 在左側面板中選擇源數據庫模式。
8. 在右側面板中，指定所選來源結構描述的目標資料庫平台。
9. 選擇建立對應。選擇來源資料庫綱要和目標資料庫平台之後，此按鈕會變成作用中狀態。

步驟 4：安裝並設定資料擷取代理程式

AWS SCT 使用資料擷取代理程式將資料遷移到 Amazon Redshift。您下載要安裝的 .zip 檔案包含解壓縮代理程式安裝 AWS SCT 程式檔案。您可以在視窗、RHEL 或 Ubuntu 中安裝資料擷取代理程式。如需詳細資訊，請參閱 [安裝擷取代理](#)。

若要設定資料擷取代理程式，請輸入來源和目標資料庫引擎。此外，請確定您已在執行資料擷取代理程式的電腦上下載來源和目標資料庫的 JDBC 驅動程式。資料擷取代理程式會使用這些驅動程式連線至來源和目標資料庫。如需詳細資訊，請參閱 [下載所需資料庫驅動程式](#)。

在 Windows 中，資料擷取代理程式安裝程式會在命令提示字元視窗中啟動設定精靈。在 Linux 中，從安裝代理程式的位置執行 `sct-extractor-setup.sh` 檔案。

步驟 5：設定 AWS SCT 以存取 Amazon S3 儲存貯體

如需設定 Amazon S3 儲存貯體的相關資訊，請參閱 Amazon 簡單儲存服務使用者指南中的 [儲存貯體概觀](#)。

步驟 6：將 AWS Snowball 工作匯入 AWS SCT 專案

若要將 AWS SCT 專案與 AWS Snowball Edge 裝置連線，請匯入 AWS Snowball 工作。

若要匯入您的 AWS Snowball 工作

1. 開啟 [設定] 功能表，然後選擇 [全域設定]。Global settings (全域設定) 對話方塊隨即出現。
2. 選擇 AWS 服務設定檔，然後選擇 [匯入工作]。
3. 選擇你的 AWS Snowball 工作。
4. 輸入您的 AWS Snowball IP 位址。如需詳細資訊，請參閱《[使用指南](#)》中的 [AWS Snowball 〈變更 IP 位址〉](#)。
5. 輸入您的 AWS Snowball 港口。如需詳細資訊，請參閱 [Edge 開發人員指南](#) 中的 [AWS Snowball 邊緣裝置使用 AWS 服務所需的連接埠](#)。AWS Snowball
6. 輸入您的 AWS Snowball 訪問密鑰和 AWS Snowball 密鑰。如需詳細資訊，請參閱《[AWS Snowball 使用指南](#)》[AWS Snowball 中的〈授權和存取控制〉](#)。
7. 選擇 Apply (套用)，然後選擇 OK (確定)。

步驟 7：在中註冊資料擷取代理程式 AWS SCT

在本節中，您將在中註冊資料萃取代理程式 AWS SCT。

若要註冊資料萃取代理程式

1. 在 [檢視] 功能表上，選擇 [資料移轉檢視 (其他)]，然後選擇 [註冊]。
2. 在「說明」中，輸入資料擷取代理程式的名稱。

3. 在「主機名稱」中，輸入執行資料擷取代理程式之電腦的 IP 位址。
4. 在連接埠中，輸入您所設定的監聽通訊埠。
5. 選擇註冊。

步驟 8：建立本機工作

接下來，您會建立移轉任務。任務包含兩個子任務，一個子工作會將資料從來源資料庫移轉至 AWS Snowball Edge 應用裝置。另一子任務會擷取裝置載入到 Amazon S3 儲存貯體的資料，並將資料遷移到目標資料庫。

若要建立移轉任務

1. 在 [檢視] 功能表上，然後選擇 [資料移轉檢視 (其他)]。
2. 左側窗格會顯示您的來源資料庫的結構描述，請選擇要遷移的結構描述物件。開啟物件的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「建立本機工作」。
3. 在「工作名稱」中，輸入資料移轉工作的描述性名稱。
4. 對於移轉模式，請選擇 [擷取]、[上傳] 和 [複製]。
5. 選擇 Amazon S3 設置。
6. 選取 [使用 Snowball]。
7. 在資料擷取代理程式可以存放資料的 Amazon S3 儲存貯體中輸入資料夾和子資料夾。
8. 選擇 Create (建立)，以建立任務。

步驟 9：執行和監視中的資料移轉任務 AWS SCT

若要啟動資料移轉工作，請選擇 [開始]。請確定已建立與來源資料庫、Amazon S3 儲存貯體、AWS Snowball 裝置的連線，以及與目標資料庫的連線 AWS。

您可以在 [工作] 索引標籤中監視和管理資料移轉工作及其子工作。您可以查看資料移轉進度，以及暫停或重新啟動資料移轉工作。

資料萃取工作輸出

在遷移任務完成後，您的資料即已準備好。利用下列資訊，根據您選擇的遷移模式和資料的位置來決定如何繼續進行。

移轉模式	資料位置
擷取、上傳和複製	資料已經存在於您的 Amazon Redshift 資料倉儲中。您可以確認資料在那裡，並開始使用。如需詳細資訊，請參閱 從用戶端工具和程式碼連線到叢集 。
擷取並上傳	<p>擷取代理程式會將您的資料以檔案形式儲存在 Amazon S3 儲存貯體中。您可以使用 Amazon Redshift 複製命令將數據加載到 Amazon Redshift。如需詳細資訊，請參閱亞馬遜紅移文件中的從 Amazon S3 載入資料。</p> <p>Amazon S3 儲存貯體中有多個資料夾，與您設定的擷取任務相對應。將資料載入 Amazon Redshift 時，請指定每個任務建立的資訊清單檔案名稱。資訊清單檔案會顯示在 Amazon S3 儲存貯體的任務資料夾中，如下所示。</p>  <p>The screenshot shows the Amazon S3 console interface for a bucket named '5204158f4823'. The breadcrumb path is 'Amazon S3 > extractortest > 5204158f4823'. There are tabs for 'Objects', 'Properties', 'Permissions', and 'Management'. A search bar is present with the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'More', 'All', and 'Deleted objects'. A list of objects is shown with checkboxes and file names: 'Name ↑', 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' file is circled in red.</p>
僅擷取	擷取代理程式會將您的資料儲存成檔案，放在工作資料夾中。手動將資料複製到 Amazon S3 儲存貯體，然後繼續執行擷取和上傳的指示。

使用虛擬磁碟分割 AWS Schema Conversion Tool

通常管理大型未分割資料表的最佳方式，就是建立子任務，以使用篩選規則來建立資料表資料的虛擬分割區。在中 AWS SCT，您可以為移轉的資料建立虛擬分割區。有三種分割區類型，適用於特定的資料類型：

- RANGE 分割區類型適用於數值以及日期和時間資料類型。
- LIST 分割區類型適用於數值、字元，以及日期和時間資料類型。
- DATE AUTO 分割區類型適用於數值、日期和時間資料類型。

AWS SCT 驗證您為建立分割區所提供的值。例如，如果您嘗試分割具有資料類型 NUMERIC 的資料行，但您提供不同資料類型的值，就 AWS SCT 會擲回錯誤。

此外，如果您使 AWS SCT 用將資料遷移到 Amazon Redshift，則可以使用原生分割來管理遷移或大型表格。如需詳細資訊，請參閱 [使用原生分割](#)。

建立虛擬分割時的限制

以下是建立虛擬分割區的限制：

- 您只能將虛擬分割用於未分割的資料表。
- 您只能在資料遷移檢視中使用虛擬分割。
- 您不能將 UNION ALL VIEW 選項與虛擬分割搭配使用。

範圍分割區類型

RANGE 分割區類型會根據數值以及日期和時間資料類型的資料欄值範圍來分割資料。這個分割區類型會建立 WHERE 子句，而您要針對每個分割區提供值的範圍。若要指定資料分割資料行的值清單，請使用「值」方塊。您可以使用 .csv 檔案來載入值資訊。

RANGE 分割區類型會在分割區值的兩端建立預設分割區。這些預設分割區會 catch 取小於或大於指定分割區值的任何資料。

例如，您可以根據您提供的值範圍建立多個分割區。下列範例會指定 LO_TAX 的分割值來建立多個分割區。

```
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
```

建立 RANGE 虛擬分割區

1. 打開 AWS SCT。
2. 選擇「資料移轉檢視 (其他)」模式。
3. 選擇您要在其中設定虛擬分割的資料表。開啟資料表的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [新增虛擬分割]。
4. 在 [新增虛擬磁碟分割] 對話方塊中，輸入下列資訊。

選項	動作
分割區類型	選擇 RANGE。對話方塊 UI 會依據您選擇的類型而變更。
資料欄名稱	選擇您想要分割的資料欄。
資料欄類型	選擇資料欄值的資料類型。
Values (數值)	在 New Value (新值) 方塊中輸入每個值，以新增值，然後選擇加號，以加入該值。
從檔案載入	(選用) 輸入包含分割區值的 .csv 檔案名稱。

5. 選擇確定。

列出分割區類型

LIST 分割區類型會根據數值、字元以及日期和時間資料類型的資料欄值來分割資料。這個分割區類型會建立 WHERE 子句，而您要針對每個分割區提供值。若要指定資料分割資料行的值清單，請使用「值」方塊。您可以使用 .csv 檔案來載入值資訊。

例如，您可以根據您提供的值來建立多個分割區。下列範例會指定 LO_ORDERKEY 的分割值來建立多個分割區。

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```


您也可以針對未包含在指定分割區中的值，建立預設分割區。

如果您想要從移轉中排除特定值，可以使用 LIST 分割區類型來篩選來源資料。例如，假設您想要省略列 `LO_ORDERKEY = 4`。在此情況下，請勿在分割區值清單中包含該值，並確定未選取 [包含其他值]。

建立 LIST 虛擬分割區

1. 打開 AWS SCT。
2. 選擇「資料移轉檢視 (其他)」模式。
3. 選擇您要在其中設定虛擬分割的資料表。開啟資料表的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [新增虛擬分割]。
4. 在 [新增虛擬磁碟分割] 對話方塊中，輸入下列資訊。

選項	動作
分割區類型	選擇 LIST。對話方塊 UI 會依據您選擇的類型而變更。
資料欄名稱	選擇您想要分割的資料欄。
新價值	在這裡輸入一個值，以將其新增至一組分割值。
包括其他值	選擇此選項來建立的預設分割區中，會存放不符合分割條件的所有值。
從檔案載入	(選用) 輸入包含分割區值的 .csv 檔案名稱。

5. 選擇確定。

日期自動拆分分區類型

DATE 自動分割分割區類型是產生 RANGE 分割區的自動化方式。使用 DATA AUTO SPLIT，您可以告訴 AWS SCT 分區屬性，從哪裡開始和結束，以及值之間的範圍大小。然後自動 AWS SCT 計算分區值。

DATA AUTO 拆分可以自動執行與創建範圍分區相關的許多工作。使用此技術和範圍分區之間的權衡是您需要對分區邊界進行多少控制。自動分割程序永遠會建立相等大小 (統一) 範圍。範圍分割可讓您根據特定資料分佈的需要，變更每個範圍的大小。例如，您可以每天、每週、每兩週、每月等使用。


```

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'

```

建立 DATE AUTO SPLIT 虛擬分割區

1. 打開 AWS SCT。
2. 選擇「資料移轉檢視 (其他)」模式。
3. 選擇您要在其中設定虛擬分割的資料表。開啟資料表的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [新增虛擬分割]。
4. 在 [新增虛擬磁碟分割] 對話方塊中，輸入下列資訊。

選項	動作
分割區類型	選擇 DATE AUTO SPLIT。對話方塊 UI 會依據您選擇的類型而變更。
資料欄名稱	選擇您想要分割的資料欄。
開始日期	輸入開始日期。
結束日期	輸入結束日期。
Interval (間隔)	輸入間隔單位，然後選擇該單位的值。

5. 選擇確定。

使用原生分割

為了加速資料移轉，您的資料擷取代理程式可以使用來源資料倉儲伺服器上的表格原生分割區。AWS SCT 支援從綠梅林、Netezza 和甲骨文遷移到 Amazon Redshift 的原生分區。

例如，建立專案之後，您可能會收集結構描述的統計資料，並分析選取要移轉的表格大小。對於超過指定大小的資料表，AWS SCT 會觸發原生資料分割機制。

若要使用原生分割

1. 開啟 AWS SCT，然後為 [檔案] 選擇 [新增專案]。[新增專案] 對話方塊隨即出現。

2. 建立新專案、新增來源和目標伺服器，以及建立對應規則。如需詳細資訊，請參閱 [建立AWS SCT 專案](#)。
3. 選擇 [檢視]，然後選擇 [主要檢視]。
4. 對於「專案設定」，請選擇「資料移轉」頁籤。選擇「使用自動分割」。對於 Greenplum 和 Netezza 來源資料庫，請輸入受支援資料表的大小下限 (以 MB 為單位) (例如 100)。AWS SCT 會自動為每個非空白的原生磁碟分割建立個別的遷移子工作。對於甲骨文到 Amazon Redshift 遷，請為所有分區資料表 AWS SCT 建立子任務。
5. 在顯示來源資料庫結構描述的左側面板中，選擇結構描述。開啟物件的內容 (按一下滑鼠右鍵) 功能表，然後選擇「收集統計資料」。對於從甲骨文遷移到 Amazon Redshift 的數據，您可以跳過此步驟。
6. 選擇要移轉的所有表格。
7. 註冊所需的代理程式數目。如需詳細資訊，請參閱 [將擷取代理程式註冊到 AWS Schema Conversion Tool](#)。
8. 為選取的表格建立資料萃取工作。如需詳細資訊，請參閱 [建立、執行和監視 AWS SCT 資料萃取工作](#)。

檢查大型資料表是否分割為子工作，以及每個子工作是否符合顯示位於來源資料倉儲中某個資料片段上之資料表一部分的資料集。

9. 啟動並監控移轉程序，直到 AWS SCT 資料擷取代理程式完成來源表格中的資料移轉為止。

將 LOB 遷移到 Amazon Redshift

Amazon Redshift 不支持存儲大型二進制對象 (LOB)。但是，如果您需要將一個或多個 LOB 遷移到 Amazon Redshift，則 AWS SCT 可以執行遷移。為此，請 AWS SCT 使用 Amazon S3 儲存貯體來存放 LOB，並將 Amazon S3 儲存貯體的 URL 寫入存放在 Amazon Redshift 中的已遷移資料。

若要將 LOB 移轉至 Amazon Redshift

1. 開啟專 AWS SCT 案。
2. 連接到來源和目標資料庫。從目標資料庫重新整理中繼資料，並確定已轉換的表格存在於該資料庫中。
3. 對於動作，選擇建立本機工作。
4. 針對移轉模式，選擇下列其中一項：
 - 擷取並上傳以擷取資料，然後將資料上傳到 Amazon S3。

- 擷取、上傳和複製以擷取資料、將資料上傳到 Amazon S3，然後將其複製到 Amazon Redshift 資料倉儲。
5. 選擇 Amazon S3 設置。
 6. 對於 Amazon S3 儲存貯體 LOB 資料夾，請在您要存放 LOB 的 Amazon S3 儲存貯體中輸入資料夾的名稱。
- 如果您使用 AWS 服務設定檔，則此欄位為選擇性欄位。AWS SCT 可以使用您設定檔中的預設設定。若要使用其他 Amazon S3 儲存貯體，請在此處輸入路徑。
7. 開啟使用代理選項以使用代理伺服器將資料上傳到 Amazon S3。然後選擇數據傳輸協議，輸入主機名，端口，用戶名和密碼。
 8. 針對端點類型，選擇 FIPS 以使用聯邦資訊處理標準 (FIPS) 端點。選擇 VPCE 以使用虛擬私有雲端 (VPC) 端點。然後，針對 VPC 端點，輸入 VPC 端點的網域名稱系統 (DNS)。
 9. 開啟將檔案複製到亞馬遜紅移後保留在 Amazon S3 上的選項，以便在將這些檔案複製到亞馬遜 Redshift 之後，將擷取的檔案保留在 Amazon S3 上。
 10. 選擇 Create (建立)，以建立任務。

資料擷取代理程式的最佳做法和疑難排

以下是針對使用擷取代理程式的一些最佳實務和故障診斷建議。

問題	故障診斷建議
效能緩慢	<p>為了改善效能，我們的建議如下：</p> <ul style="list-style-type: none"> • 安裝多個代理程式。 • 在靠近資料倉儲的電腦上安裝代理程式。 • 不要在單一代理程式任務中執行所有資料表。
爭用延遲	避免有太多代理程式同時存取您的資料倉儲。
代理程式暫時故障	如果代理程式故障，其每個任務的狀態會在 AWS SCT 中顯示為失敗。如果您稍候一下，在某些情況下，代理程式可以復原。在此情況下，其任務的狀態會在 AWS SCT 中更新。
代理程式永久故障	如果執行代理程式的電腦發生永久故障，且該代理程式正在執行任務，您可以更換新的代理程式來繼續執行任務。原始代理程式

問題	故障診斷建議
	<p>的工作資料夾，必須位在與原始代理程式不同的電腦上，才可以更換新的代理程式。若要更換新的代理程式，請執行下列動作：</p> <ul style="list-style-type: none">• 在新的電腦上安裝代理程式。• 使用原始代理程式的相同設定 (包括連接埠號碼和工作資料夾) 來設定新的代理程式。• 啟動代理程式。代理程式啟動之後，任務會探索新的可用代理程式，並繼續在新的代理程式上執行。

使用轉換應用程式 SQLAWS SCT

當您將資料庫結構描述從一個引擎轉換到另一個引擎，您也需更新應用程式中的 SQL 程式碼，以便與新的資料庫引擎互動，取代舊引擎。您可以檢視、分析、編輯和儲存轉換後的 SQL 程式碼。

您可以使用 AWS Schema Conversion Tool (AWS SCT) 來轉換 C++、C#、Java 或其他應用程式程式碼中的 SQL 程式碼。對於甲骨文到 PostgreSQL 的轉換，您可以使用 AWS SCT 將 SQL* 加代碼轉換為 PSQL。另外，對於甲骨文到 PostgreSQL 的轉換，您可以使用 AWS SCT 轉換嵌入到 C#、C++、Java 和 Pro*C 應用程序中的 SQL 代碼。

主題

- [轉換應用程式 SQL 概觀](#)
- [在您的應用程序中轉換 SQL 代碼AWS SCT](#)
- [在 C# 應用程序中轉換 SQL 代碼AWS SCT](#)
- [在 C++ 應用程序中轉換 SQL 代碼AWS SCT](#)
- [在 Java 應用程序中轉換 SQL 代碼AWS SCT](#)
- [在 Pro*C 應用程序中轉換 SQL 代碼AWS SCT](#)

轉換應用程式 SQL 概觀

若要在應用程式中轉換 SQL 程式碼，請執行下列高階步驟：

- 建立應用程式轉換專案— 應用程式轉換專案是資料庫結構描述轉換專案的子系。每個資料庫結構描述轉換專案可以有一或多個子項應用程式轉換專案。如需詳細資訊，請參閱[建立一般應用程式轉換專案AWS SCT](#)。
- 分析並轉換您的 SQL 程式碼— AWS SCT 會分析您的應用程式、擷取 SQL 程式碼，並建立轉換後的 SQL 的本機版本，供您檢閱和編輯。除非您準備好，否則此工具不會變更應用程式中的程式碼。如需詳細資訊，請參閱[分析和轉換您的 SQL 代碼AWS SCT](#)。
- 建立應用程式評估報告— 應用程式評估報表提供有關將應用程式 SQL 程式碼從來源資料庫結構描述轉換為目標資料庫結構描述的重要資訊。如需詳細資訊，請參閱[建立與使用AWS SCT評估報告AWS SCT](#)。
- 編輯、套用變更，並儲存轉換後的 SQL 程式碼— 評估報告包含無法自動轉換的 SQL 程式碼項目清單。對於這些項目，您可以手動編輯 SQL 程式碼以執行轉換。如需詳細資訊，請參閱[編輯和保存轉換後的 SQL 代碼AWS SCT](#)。

在您的應用程式中轉換 SQL 代碼AWS SCT

您可以使用AWS SCT轉換嵌入到您的應用程式中的 SQL 代碼。泛型AWS SCT應用程式轉換器將您的應用程式代碼視為純文本 它掃描您的應用程式代碼，並使用正則表達式提取 SQL 代碼。該轉換器支持不同類型的源代碼文件，並與以任何編程語言編寫的應用程式代碼一起使用。

一般應用程式轉換器有下列限制。它不會深入研究應用程式的編程語言特定的應用程式邏輯。此外，泛型轉換器不支援來自不同應用程式物件的 SQL 陳述式，例如函數、參數、區域變數等。

若要改善應用程式 SQL 程式碼轉換，請使用特定語言的應用程式 SQL 程式碼轉換器。如需詳細資訊，請參閱 [在 C# 應用程式中轉換 SQL 代碼](#)、[在 Java 應用程式中轉換 SQL 程式碼](#) 及 [在 Pro*C 應用程式中轉換 SQL 程式碼](#)。

建立一般應用程式轉換專案AWS SCT

在 AWS Schema Conversion Tool 中，應用程式轉換專案是資料庫結構描述轉換專案的子項。每個資料庫結構描述轉換專案可以有一或多個子項應用程式轉換專案。

Note

AWS SCT不支援下列來源與目標之間的轉換：

- 甲骨文目標
- PostgreSQL 轉換為 Aurora
- MySQL 目標資料庫
- SQL 伺服器目標資料庫
- Amazon Redshift 到 Amazon Redshift
- SQL 服務器到巴貝魚
- SQL 伺服器整合服務AWS Glue
- 阿帕奇卡桑德拉到 Amazon DynamoDB

使用下列程序建立一般應用程式轉換專案。

若要建立應用程式轉換專案

1. 在AWS Schema Conversion Tool，選擇新一般應用程式在應用功能表

會出現 New application conversion project (新增應用程式轉換專案) 對話方塊。

Creating a generic application conversion project

Enter the name, location and type of the new application conversion project.

Name: Application conversion project 1

Location: C:\AWS-SCT-Demo Browse

Language: Java Target parameter style: Same as in source

Settings

Don't cast bind variables to SQL types i

Keep object names i

Choose the source database schema that your application uses which is mapped with the target tree object:

- ▼ Schemas [58]
 - ANONYMOUS
 - APPQOSSYS
 - AUDSYS
 - CHINOOK
 - CTXSYS
 - DVSYs

OK Cancel

2. 新增下列專案資訊。

對於此參數	執行此作業
名稱	輸入應用程式轉換專案的名稱。每個資料庫結構描述轉換專案可以有一或多個子項應用程式轉換專案，因此如果您稍後會新增更多專案，請選擇讓人好理解的名稱。
位置	輸入應用程式原始程式碼的位置。
語言	選擇下列其中一項： <ul style="list-style-type: none"> • Java • C++

對於此參數	執行此作業
目標參數型式	<p>選擇要用於已轉換程式碼中繫結變數的語法。不同的數據庫平台對綁定變量使用不同的語法。請選擇下列其中一個選項：</p> <ul style="list-style-type: none"> • 與來源相同 • Positional (?) • Indexed (:1) • Indexed (\$1) • Named (@name) • Named (:name) • 具名 (&N) • 命名 (\$ 名稱) • 已命名的 (#name) • 已命名 (! 名字 !)
選擇源資料庫模式	在來源樹狀結構中，選擇應用程式使用的結構描述。請確定此綱要是對應規則的一部分。

3. 選擇不要將綁定變量轉換為 SQL 類型以避免將變量類型綁定到 SQL 類型轉換。此選項僅適用於甲骨文到 PostgreSQL 的轉換。

例如，您的來源應用程式程式碼包含下列 Oracle 查詢：

```
SELECT * FROM ACCOUNT WHERE id = ?
```

當您選擇不要將綁定變量轉換為 SQL 類型,AWS SCT轉換此查詢，如下圖所示。

```
SELECT * FROM account WHERE id = ?
```

當你清除不要將綁定變量轉換為 SQL 類型,AWS SCT將繫結變數類型變更為NUMERIC資料類型 轉換結果如下所示。


```
SELECT * FROM account WHERE id = (?):NUMERIC
```

4. 選擇保留物件名稱以避免將模式名稱添加到轉換對象的名稱中。此選項僅適用於甲骨文到 PostgreSQL 的轉換。

例如，假設您的原始應用程式程式碼包含下列 Oracle 查詢。

```
SELECT * FROM ACCOUNT
```

當您選擇保留物件名稱,AWS SCT轉換此查詢，如下圖所示。

```
SELECT * FROM account
```

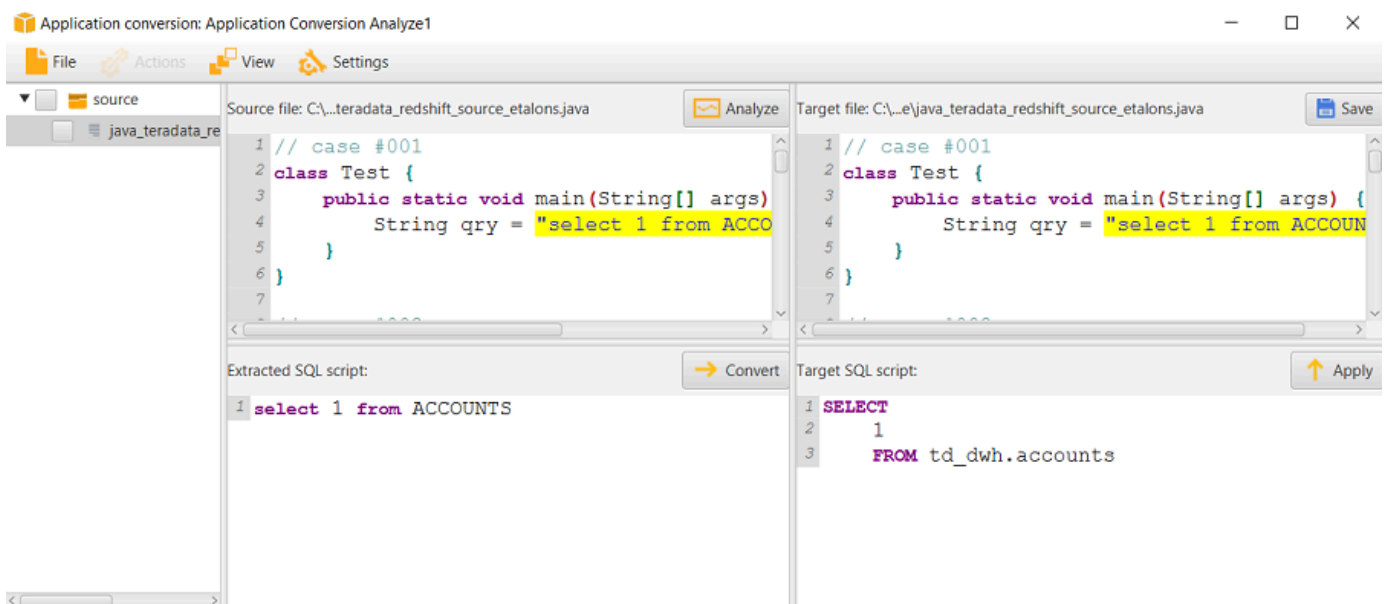
當你清除保留物件名稱,AWS SCT會將結構定義名稱加入表的名稱。轉換結果如下所示。

```
SELECT * FROM schema_name.account
```

如果您的源代碼在對象的名稱中包含父對象的名稱，AWS SCT在轉換後的代碼中使用此格式。在這種情況下，忽略保留物件名稱選項，因為AWS SCT在轉換後的代碼中添加父對象的名稱。

5. 選擇 OK (確定) 以建立應用程式轉換專案。

專案視窗會開啟。



管理應用程式轉換專案AWS SCT

您可以開啟現有的應用程式轉換專案，並新增多個應用程式轉換專案。

建立應用程式轉換專案後，專案視窗會自動開啟。您可以關閉應用程式轉換專案視窗，稍後再回到它。

若要開啟現有的應用程式轉換專案

1. 在左側面板中，選擇應用程式轉換專案節點，然後開啟內容 (按一下滑鼠右鍵) 選單。
2. 選擇管理申請。

若要新增其他應用程式轉換專案

1. 在左側面板中，選擇應用程式轉換專案節點，然後開啟內容 (按一下滑鼠右鍵) 選單。
2. 選擇 New application (新增應用程式)。
3. 輸入建立新應用程式轉換專案所需的資訊。如需詳細資訊，請參閱[建立一般應用程式轉換專案](#)。

分析和轉換您的 SQL 代碼AWS SCT

使用下列程序來分析並轉換您的 SQL 程式碼AWS Schema Conversion Tool。

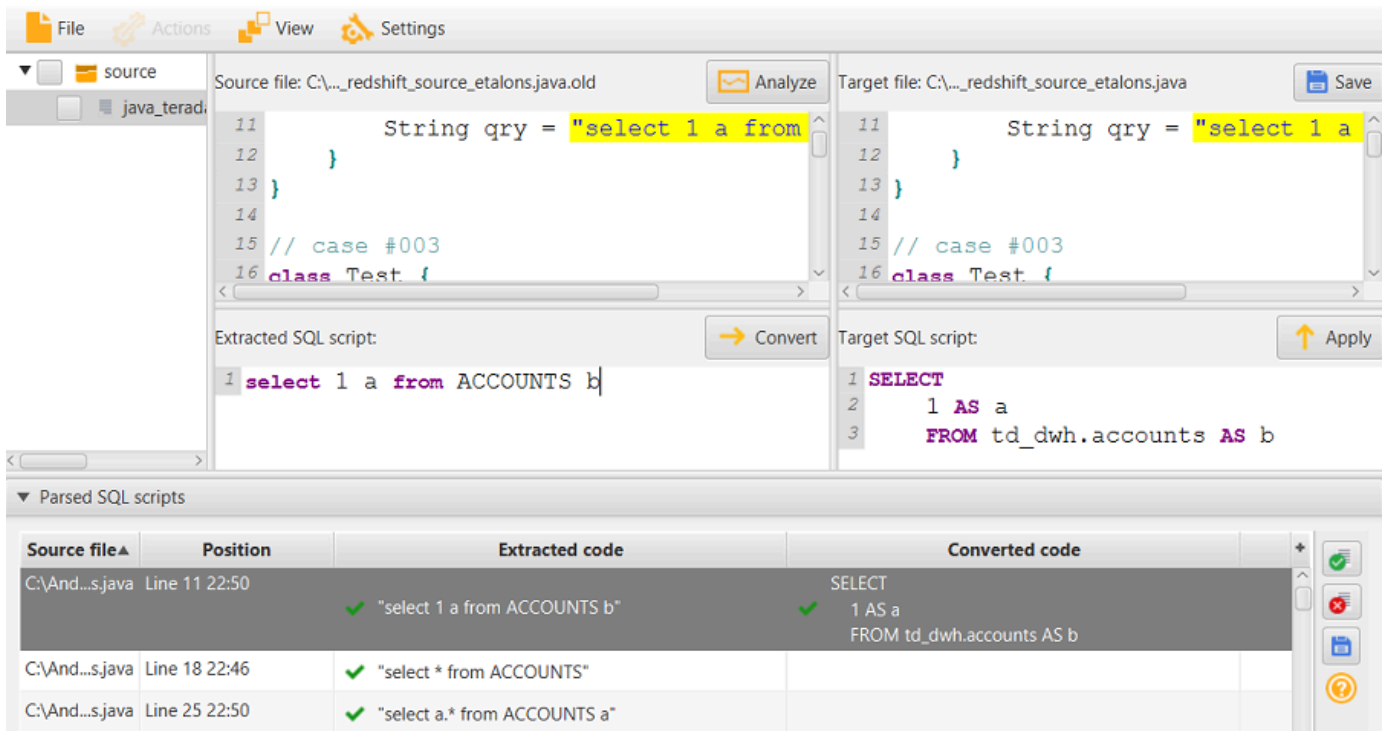
若要分析和轉換您的 SQL 程式碼

1. 開啟現有的應用程式轉換專案，然後選擇分析。

AWS SCT分析您的應用程式程式碼並擷取 SQL 程式碼。AWS SCT顯示擷取的 SQL 程式碼已剖析的 SQL 指令碼列表。

2. 對於已剖析的 SQL 指令碼」下方，選擇要檢閱其擷取的 SQL 程式碼的項目。AWS SCT顯示所選項目的代碼擷取的 SQL 腳本檔窗格
3. 選擇轉換將 SQL 代碼轉換為擷取的 SQL 腳本檔窗格AWS SCT將代碼轉換為與目標數據庫兼容的格式。

您可以編輯轉換後的 SQL 代碼。如需詳細資訊，請參閱[編輯和保存轉換後的 SQL 代碼](#)。



4. 建立應用程式轉換評估報告時，AWS SCT轉換所有提取的 SQL 代碼項目。如需詳細資訊，請參閱[建立和使用評估報告](#)。

建立與使用AWS SCT評估報告AWS SCT

該應用轉換評估報告提供有關將應用程式 SQL 程式碼轉換為與目標資料庫相容格式的資訊。該報告詳細介紹了所有提取的 SQL 代碼，所有轉換的 SQL 代碼以及 SQL 代碼的行動項目AWS SCT無法轉換。

建立應用程式轉換評估報告

使用下列程序建立應用程式轉換評估報告。

建立應用程式轉換評估報告

1. 在應用程式轉換專案視窗中，選擇建立報告在動作功能表

AWS SCT會建立應用程式轉換評估報告，並在應用程式轉換專案視窗中開啟。

2. 檢閱 Summary (摘要) 索引標籤。

該摘要標籤 (如下所示) 會顯示應用程式評估報告的摘要資訊。它會顯示已自動轉換的 SQL 程式碼項目，以及無法自動轉換的項目。



3. 選擇SQL 擷取動作。

檢閱 SQL 程式碼項目的清單AWS SCT無法從您的源代碼中提取。

4. 選擇SQL 轉換動作。

檢閱 SQL 程式碼項目的清單AWS SCT無法自動轉換。使用建議的動作手動轉換 SQL 程式碼。如需如何編輯轉換後的 SQL 程式碼的資訊，請參閱[編輯和保存轉換後的 SQL 代碼AWS SCT](#)。

5. (選擇性) 將報表的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：

- 選擇儲存資料庫位於右上角，將報告另存為 PDF 檔案。

PDF 檔案包含執行摘要、行動項目和應用程式轉換的建議。

- 選擇儲存資料庫。位於右上角，以將報告儲存為 CSV 檔案。

CSV 檔案包含行動項目、建議的動作，以及轉換 SQL 程式碼所需手動工作的估計複雜度。

編輯和保存轉換後的 SQL 代碼AWS SCT

評估報告包含的 SQL 程式碼項目清單AWS SCT無法轉換。對於每個項目，AWS SCT建立行動項目SQL 轉換動作設定檔 對於這些項目，您可以手動編輯 SQL 程式碼以執行轉換。

使用下列步驟來編輯您的轉換後 SQL 程式碼、套用變更，然後儲存。

若要編輯、套用變更，並儲存轉換後的 SQL 程式碼

- 直接在 Target SQL script (目標 SQL 指令碼) 窗格中編輯轉換後 SQL 程式碼。如果未顯示轉換後的程式碼，您可以在窗格中按一下，並開始輸入。
- 當您完成編輯轉換後的 SQL 程式碼，請選擇 Apply (套用)。此時，變更就會儲存在記憶體中，但尚未寫入您的檔案。
- 選擇 Save (儲存) 將變更儲存至您的檔案。

當您選擇儲存，您會覆寫原始檔案。在儲存前先複製您的原始檔案，即可擁有一份原始應用程式程式碼的記錄。

在 C# 應用程序中轉換 SQL 代碼AWS SCT

對於甲骨文到 PostgreSQL 的轉換，您可以使用AWS Schema Conversion Tool(AWS SCT) 轉換嵌入到 C# 應用程序中的 SQL 代碼。這個特定的 C# 應用程序轉換器了解應用程序邏輯 它會收集位於不同應用程式物件中的陳述式，例如函數、參數、區域變數等。

由於這項深入分析，C# 應用程式 SQL 程式碼轉換器會提供比一般轉換器更好的轉換結果。

在中創建 C# 應用程序轉換項目AWS SCT

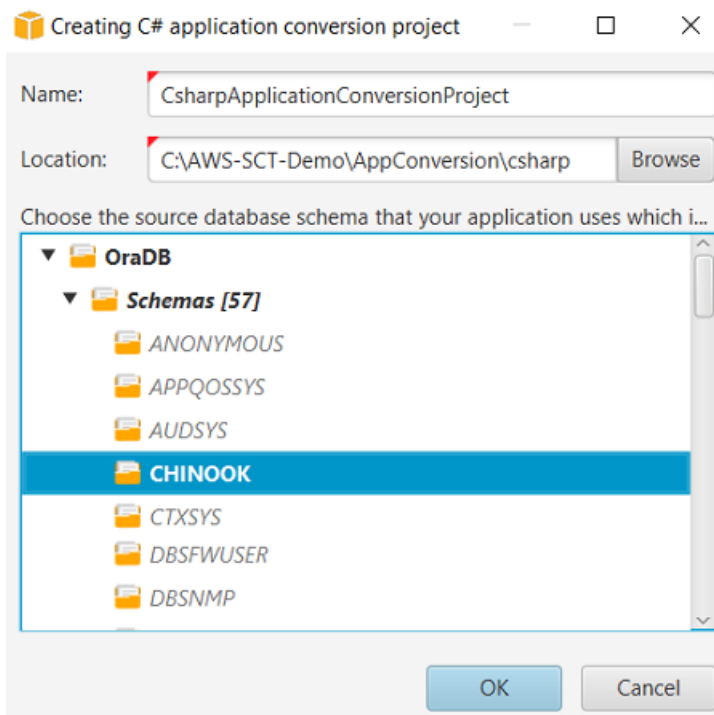
您只能建立 C# 應用程式轉換專案，將 Oracle 資料庫結構描述轉換為 PostgreSQL 資料庫結構描述。請務必在專案中新增對應規則，其中包含來源 Oracle 結構描述和目標 PostgreSQL 資料庫。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

您可以在一個單一新增多個應用程式轉換專案AWS SCT項目 使用下列程序建立 C# 應用程式轉換專案。

若要建立 C# 應用程式轉換專案

1. 建立資料庫轉換專案，並加入來源 Oracle 資料庫。如需詳細資訊，請參閱 [建立AWS SCT專案](#) 及 [將資料庫伺服器新增至AWS SCT專案](#)。
2. 新增包含來源 Oracle 資料庫和目標 PostgreSQL 資料庫的對應規則。您可以新增目標 PostgreSQL 資料庫，或在對應規則中使用虛擬 PostgreSQL 目標資料庫平台。如需詳細資訊，請參閱 [在中建立對應規則 AWS SCT](#) 及 [使用虛擬目標](#)。
3. 在「」檢視功能表中，選擇主視圖。
4. 在「」应用功能表中，選擇新的 C# 應用程式。

該建立 C# 應用程式轉換專案出現對話方塊。



5. 對於姓名，輸入 C# 應用程式轉換專案的名稱。由於每個資料庫結構描述轉換專案都可以有一或多個子應用程式轉換專案，因此請選擇加入多個專案時合理的名稱。
6. 對於位置下，輸入應用程式的原始碼位置。
7. 在來源樹狀結構中，選擇應用程式使用的結構描述。請確定此綱要是對應規則的一部分。AWS SCT以粗體亮顯屬於對映規則一部分的資料架構。
8. 選擇好創建您的 C# 應用程序轉換項目。
9. 在中尋找您的 C# 應用程式轉換專案应用左側面板中的節點。

將您的 C # 應用程序 SQL 代碼轉換為AWS SCT

將 C# 應用程式新增至AWS SCT項目，將 SQL 代碼從此應用程式轉換為與目標數據庫平台兼容的格式。使用下列程序來分析和轉換內嵌在 C# 應用程式中的 SQL 程式碼AWS Schema Conversion Tool。

若要轉換 SQL 程式碼

1. 展開C # 節點下应用在左側面板中。
2. 選擇要轉換的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇轉換。AWS SCT會分析您的原始程式碼檔案、決定應用程式邏輯，並將程式碼中繼資料載入專案。此程式碼中繼資料包括 C# 類別、物件、方法、全域變數、介面等。

目標資料庫目標資料庫。AWS SCT會建立與來源應用程式專案類似的資料夾結構。在這裡您可以查看轉換後的應用程式代碼。

```

Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18     {
19         string str1 = "SELECT *\n" +
20             "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
21             "WHERE COL_CHAR = \'CHAR\''";
22
23         command.CommandText = str1;
24         command.ExecuteNonQuery();
25     }
26     connection.Close();

Cursor position: 1005

Target Amazon RDS for PostgreSQL file: SpecialEscapeSequences.cs
Properties Text Apply status Key management
19     {
20         string str1 = "SELECT *\n" +
21             "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
22             "WHERE COL_CHAR = \'CHAR\''";
23
24         command.CommandText = str1;
25         command.ExecuteNonQuery();
26     }
27     connection.Close();

```

4. 保存轉換後的應用程式代碼。如需詳細資訊，請參閱[儲存轉換後的應用程式碼](#)。

您的 C# 應用程式可能包含與不同來源資料庫互動的 SQL 程式碼。您可以移轉至 PostgreSQL 的數個這些來源資料庫。在這種情況下，請確定您不會轉換與從移轉範圍排除的資料庫互動的 SQL 程式碼。您可以從轉換範圍中排除 C# 應用程式的來源檔案。若要這麼做，請清除您要從轉換範圍中排除的檔案名稱的核取方塊。

變更轉換範圍之後，AWS SCT 仍然分析 SQL 代碼您的 C# 應用程式的所有源文件。然後 AWS SCT 將您從轉換範圍中排除的所有來源檔案複製到目標資料夾。此作業可讓您在儲存轉換後的應用程式檔案之後建置應用程式。

保存轉換後的應用程式代碼 AWS SCT

使用下列程序儲存轉換的應用程式程式碼。

儲存轉換後的應用程式代碼

1. 展開 C# 節點下应用目標資料庫目標資料庫。
2. 選擇轉換後的應用程式，然後選擇儲存。

3. 輸入要儲存轉換後的應用程式程式碼的資料夾路徑，然後選擇選取資料庫。

管理 C# 應用程式轉換項目AWS SCT

您可以新增多個 C# 應用程式轉換專案，並在AWS SCT項目，或從您的 C# 轉換項目中刪除AWS SCT項目

若要新增其他 C# 應用程式轉換專案

1. 展開应用左側面板中的節點。
2. 選擇C # node (我的電腦)，開啟內容功能表 (按一下右鍵)。
3. 選擇 New application (新增應用程式)。
4. 輸入建立新 C# 應用程式轉換專案所需的資訊。如需詳細資訊，請參閱[創建 C# 應用程式轉換項目](#)。

在原始應用程式程式碼中進行變更之後，請將其上傳至AWS SCT項目

上傳更新的應用程式代碼

1. 展開C # 節點下应用在左側面板中。
2. 選擇要更新的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刷新然後選擇是。

AWS SCT從來源檔案上傳您的應用程式程式碼，並移除轉換結果。若要保留您在中所做的程式碼變更AWS SCT和轉換結果，創建一個新的 C# 轉換項目。

若要移除 C# 應用程式轉換專案

1. 展開C # 節點下应用在左側面板中。
2. 選擇要移除的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刪除然後選擇好。

在中建立 C# 應用程式轉換評估報告AWS SCT

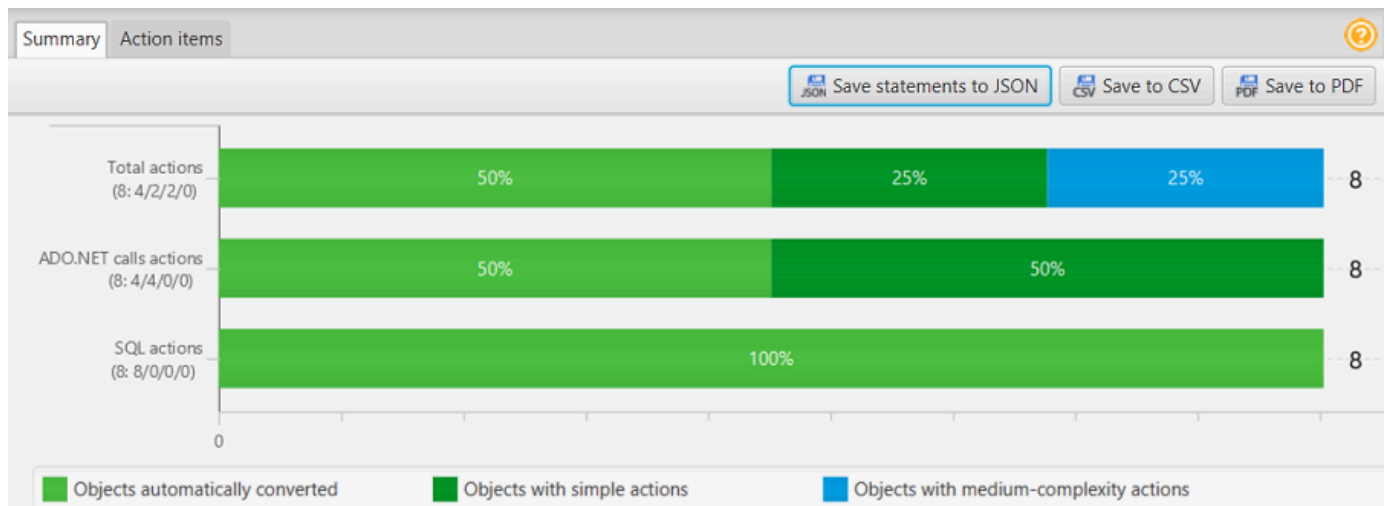
該C# 應用程式轉換評估報告提供有關將 C # 應用程式中嵌入的 SQL 代碼轉換為與目標數據庫兼容的格式的信息。評估報告提供所有 SQL 執行點和所有原始程式碼檔案的轉換詳細資訊。評估報告也包含 SQL 程式碼的行動項目，AWS SCT無法轉換。

使用下列程序建立 C# 應用程式轉換評估報告。

若要建立 C# 應用程式轉換評估報告

1. 展開C # 節點下应用在左側面板中。
2. 選擇要轉換的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇轉換。
4. 在「」檢視功能表中，選擇評估報告檢視。
5. 檢視摘要設定檔

該摘要選項卡，如下所示，顯示從 C # 應用程式評估報告的執行摘要信息。它顯示了所有 SQL 執行點和所有源代碼文件的轉換結果。



6. 選擇將陳述式儲存為 JSON將 C # 應用程式中提取的 SQL 代碼保存為 JSON 文件。
7. (選擇性) 將報表的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：

- 選擇儲存筆記本位於右上角，將報告另存為 PDF 檔案。

PDF 檔案包含執行摘要、行動項目和應用程式轉換的建議。

- 選擇儲存筆記本位於右上角，以將報告儲存為 CSV 檔案。

CSV 檔案包含行動項目、建議的動作，以及轉換 SQL 程式碼所需手動工作的估計複雜度。

在 C++ 應用程式中轉換 SQL 代碼AWS SCT

對於甲骨文到 PostgreSQL 的轉換，您可以使用AWS SCT轉換嵌入到 C++ 應用程式中的 SQL 代碼。這個特定的 C++ 應用程式轉換器理解應用程式邏輯，它會收集位於不同應用程式物件中的陳述式，例如函數、參數、區域變數等。

由於這種深入的分析，C++ 應用程式 SQL 程式碼轉換器提供了比一般轉換器更好的轉換結果。

在中建立 C++ 應用程式轉換專案AWS SCT

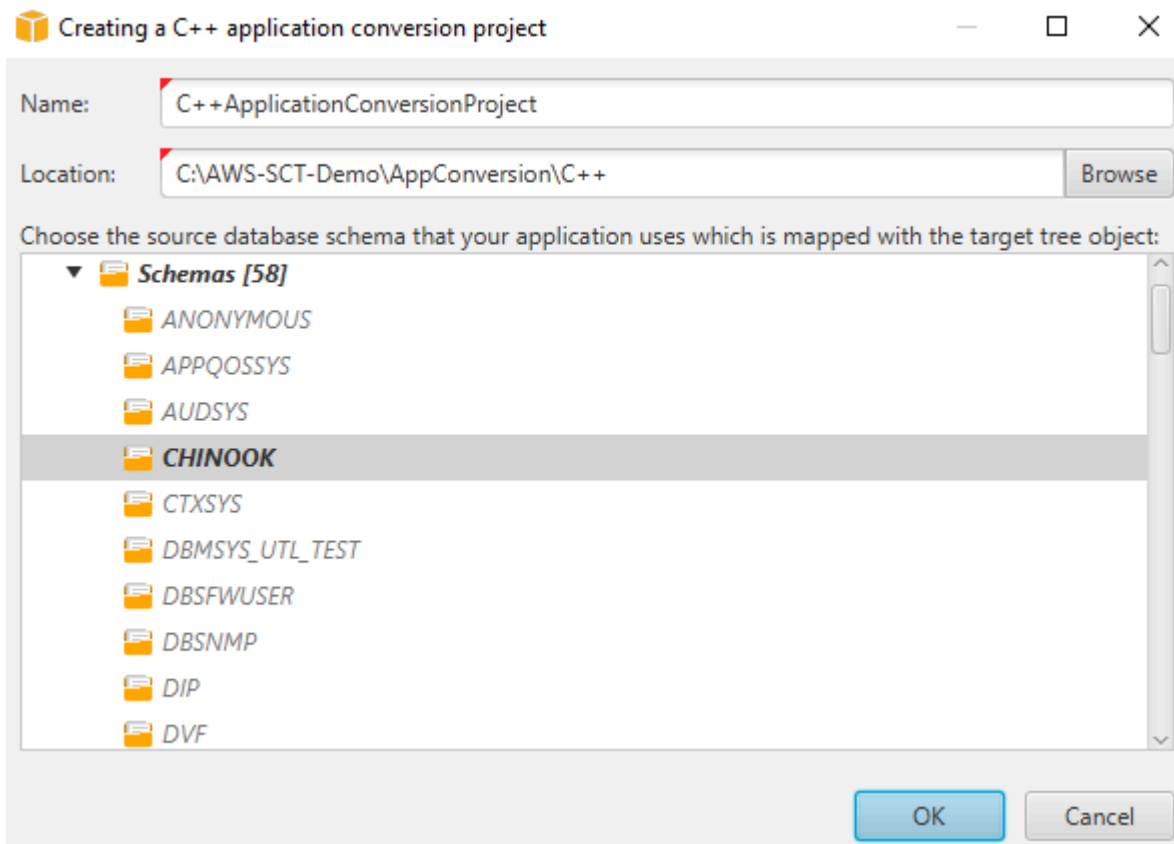
您只能建立 C++ 應用程式轉換專案，將 Oracle 資料庫結構描述轉換為 PostgreSQL 資料庫結構描述。請務必在專案中新增對應規則，其中包含來源 Oracle 結構描述和目標 PostgreSQL 資料庫。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

您可以在一個單一新增多個應用程式轉換專案AWS SCT項目

若要建立 C++ 應用程式轉換專案

1. 建立資料庫轉換專案，並加入來源 Oracle 資料庫。如需詳細資訊，請參閱 [建立AWS SCT專案](#) 及 [將資料庫伺服器新增至AWS SCT專案](#)。
2. 新增包含來源 Oracle 資料庫和目標 PostgreSQL 資料庫的對應規則。您可以新增目標 PostgreSQL 資料庫，或在對應規則中使用虛擬 PostgreSQL 目標資料庫平台。如需詳細資訊，請參閱 [在中建立對應規則 AWS SCT](#) 及 [使用虛擬目標](#)。
3. 在「」檢視功能表中，選擇主視圖。
4. 在「」应用功能表中，選擇新的 C++ 應用程式。

該建立 C++ 應用程式轉換專案出現對話方塊。



5. 對於姓名下，輸入 C++ 應用程式轉換專案的名稱。由於每個資料庫結構描述轉換專案都可以有一或多個子應用程式轉換專案，因此請選擇加入多個專案時合理的名稱。
6. 對於位置下，輸入應用程式的原始碼位置。
7. 在來源樹狀結構中，選擇應用程式使用的結構描述。請確定此綱要是對應規則的一部分。AWS SCT以粗體亮顯屬於對映規則一部分的資料架構。
8. 選擇好創建您的 C++ 應用程式轉換項目。
9. 在中找到您的 C++ 應用程式轉換專案应用左側面板中的節點。

轉換您的 C++ 應用程式 SQL 代碼AWS SCT

在您將 C++ 應用程式新增至AWS SCT項目，將 SQL 代碼從此應用程式轉換為與目標數據庫平台兼容的格式。使用下列程序來分析和轉換內嵌在 C++ 應用程式中的 SQL 程式碼AWS SCT。

若要轉換 SQL 程式碼

1. 展開C++節點下应用在左側面板中，然後選擇要轉換的應用程式。

2. 在來源 Oracle 應用模組專案，選擇設定。檢閱並編輯所選 C++ 應用程式的轉換設定。您也可以為新增至您的所有 C++ 應用程式指定轉換設定AWS SCT項目 如需詳細資訊，請參閱[管理 C++ 應用程式轉換專案](#)。
3. 對於編譯器類型」下方，選擇您用於 C++ 應用程式原始程式碼的編譯器。AWS SCT支持以下 C++ 編譯器：Microsoft 視覺 C++、海灣合作委員會，GNU 編譯器集合，以及鏜。預設選項為Microsoft 視覺 C++。
4. 對於使用者定義」中，輸入包含 C++ 專案中使用者定義巨集的檔案路徑。確定此檔案具有下列結構：`#define name value`。在前面的範例中，value是選用參數。此選擇性參數的預設值為1。

若要建立這個檔案，請在 Microsoft 視覺工作室中開啟您的專案，然後選擇項目、性質、C/C++，以及預處理器。對於預處理器定義，選擇編輯並將名稱和值複製到新的文本文件中。然後，對於文件中的每個字符串，添加以下前綴：`#define`。

5. 對於外部包含目錄」中，輸入包含您在 C++ 專案中使用之外部程式庫的資料夾路徑。
6. 在左窗格中，選擇要轉換的應用程式，然後開啟內容 (按一下滑鼠右鍵) 選單。
7. 選擇轉換。AWS SCT會分析您的原始程式碼檔案、決定應用程式邏輯，並將程式碼中繼資料載入專案。此程式碼中繼資料包括 C++ 類別、物件、方法、全域變數、介面等。

目標資料庫目標資料庫。AWS SCT會建立與來源應用程式專案類似的資料夾結構。在這裡，您可以查看轉換後的應用程式代碼，如下圖所示。

Source Oracle file: StringInitialization.cpp

```

44     if ((dRet == SQLDriverConnect(hDBC, NULL, lpConnectionStr, connectionStr.size(), OutConnStr, 0xFF
45     {
46         SQLHANDLE hSelectStm = NULL;
47
48         if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDBC, &hSelectStm)) == SQL_SUCCESS)
49         {
50
51             char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
52             strncpy_s(&buff[0], 0xFF, "SELECT JAVADB.GET_INT() FROM DUAL", 18);
53
54             if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
55             {

```

Cursor position: 0

Target Amazon RDS for PostgreSQL file: StringInitialization.cpp

```

45     if ((dRet == SQLDriverConnect(hDBC, NULL, lpConnectionStr, connectionStr.size(), OutConnStr,
46     {
47         SQLHANDLE hSelectStm = NULL;
48
49         if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDBC, &hSelectStm)) == SQL_SUCCESS)
50         {
51
52             char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
53             strncpy_s(&buff[0], 0xFF, "SELECT javadb.get_int()", 18);
54
55             if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
56             {

```

8. 保存轉換後的應用程序代碼。如需詳細資訊，請參閱[儲存轉換後的應用程式碼](#)。

保存轉換後的應用程序代碼AWS SCT

使用下列程序儲存轉換的應用程式碼。

儲存轉換後的應用程式代碼

1. 展開C++節點下应用目標資料庫目標資料庫。
2. 選擇轉換後的應用程序，然後選擇儲存。
3. 輸入要儲存轉換後的應用程式碼的資料夾路徑，然後選擇選取資料庫。

管理 C++ 應用程序轉換項目AWS SCT

您可以新增多個 C++ 應用程式轉換專案、編輯轉換設定、更新 C++ 應用程式碼，或從您的AWS SCT項目

若要新增其他 C++ 應用程式轉換專案

1. 展開应用左側面板中的節點。
2. 選擇C++node (我的電腦)，開啟內容功能表 (按一下右鍵)。
3. 選擇 New application (新增應用程式)。
4. 輸入建立新 C++ 應用程式轉換專案所需的資訊。如需詳細資訊，請參閱[建立 C++ 應用程式轉換專案](#)。

您可以為您的所有 C++ 應用程式轉換專案指定轉換設定AWS SCT項目

編輯所有 C++ 應用程式的轉換設定

1. 在「」設定功能表中，選擇專案設定，然後選擇申請轉換。
2. 對於編譯器類型」下方，選擇您用於 C++ 應用程式原始程式碼的編譯器。AWS SCT支持以下 C ++ 編譯器：Microsoft 視覺 C++，海灣合作委員會，GNU 編譯器集合，以及鐳。預設選項為Microsoft 視覺 C++。
3. 對於使用者定義」中，輸入包含 C++ 專案中使用者定義巨集的檔案路徑。確定此檔案具有下列結構：`#define name value`。在前面的範例中，value是選用參數。此選擇性參數的預設值為1。

若要建立這個檔案，請在 Microsoft 視覺工作室中開啟您的專案，然後選擇項目,性質,C/C ++，以及預處理器。對於預處理器定義，選擇編輯並將名稱和值複製到新的文本文件中。然後，對於文件中的每個字符串，添加以下前綴：`#define`。

4. 對於外部包含目錄」中，輸入包含您在 C++ 專案中使用之外部程式庫的資料夾路徑。
5. 選擇好以儲存專案設定並關閉視窗。

或者，您可以為每個 C++ 應用程式轉換專案指定轉換設定。如需詳細資訊，請參閱[轉換您的 C++ 應用程式 SQL 程式碼](#)。

在原始應用程式程式碼中進行變更之後，請將其上傳至AWS SCT項目

上傳更新的應用程式代碼

1. 展開C++節點下应用在左側面板中。
2. 選擇要更新的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刷新然後選擇是。

AWS SCT從來源檔案上傳您的應用程式程式碼，並移除轉換結果。若要保留您在中所做的程式碼變更AWS SCT和轉換結果，創建一個新的 C++ 轉換項目。

還有AWS SCT移除您為所選應用程式指定的應用程式轉換設定。上傳更新的應用程式代碼後，AWS SCT套用專案設定中的預設值。

若要移除 C++ 應用程式轉換專案

1. 展開C++節點下应用在左側面板中。
2. 選擇要移除的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刪除然後選擇好。

在中建立 C++ 應用程式轉換評估報告AWS SCT

該C++ 應用程式轉換評估報告提供有關將 C++ 應用程式中嵌入的 SQL 代碼轉換為與目標數據庫兼容的格式的信息。評估報告提供所有 SQL 執行點和所有原始程式碼檔案的轉換詳細資訊。評估報告也包含 SQL 程式碼的行動項目，AWS SCT無法轉換。

建立 C++ 應用程式轉換評估報告

1. 展開C++節點下应用在左側面板中。
2. 選擇要轉換的應用程式，開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇轉換。
4. 在「」檢視功能表中，選擇評估報告檢視。
5. 檢視摘要設定檔

該摘要標籤顯示 C++ 應用程式評估報告中的執行摘要信息。它顯示了所有 SQL 執行點和所有源代碼文件的轉換結果。

6. 選擇將陳述式儲存為 JSON將 Java 應用程式中提取的 SQL 代碼保存為 JSON 文件。
7. (選擇性) 將報表的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：
 - 選擇儲存筆記本位於右上角，將報告另存為 PDF 檔案。

PDF 檔案包含執行摘要、行動項目和應用程式轉換的建議。

- 選擇儲存筆記本位於右上角，以將報告儲存為 CSV 檔案。

CSV 檔案包含行動項目、建議的動作，以及轉換 SQL 程式碼所需手動工作的估計複雜度。

在 Java 應用程式中轉換 SQL 代碼AWS SCT

對於甲骨文到 PostgreSQL 的轉換，您可以使用AWS Schema Conversion Tool轉換嵌入到 Java 應用程式中的 SQL 代碼。此特定 Java 應用程式轉換器了解應用程式邏輯。它會收集位於不同應用程式物件中的陳述式，例如函數、參數、區域變數等。

由於這種深入的分析，Java 應用程式 SQL 代碼轉換器提供了比通用轉換器更好的轉換結果。

如果您的 Java 應用程式使用 MyBatis 框架與數據庫交互，那麼您可以使用AWS SCT轉換嵌入到的 SQL 語句 MyBatis XML 文件和註釋。要了解這些 SQL 語句的邏輯，AWS SCT使用 MyBatis組態檔。AWS SCT可以在應用程式資料夾中自動探索此檔案，或者您也可以手動輸入此檔案的路徑。

在中建立 Java 應用程式轉換專案AWS SCT

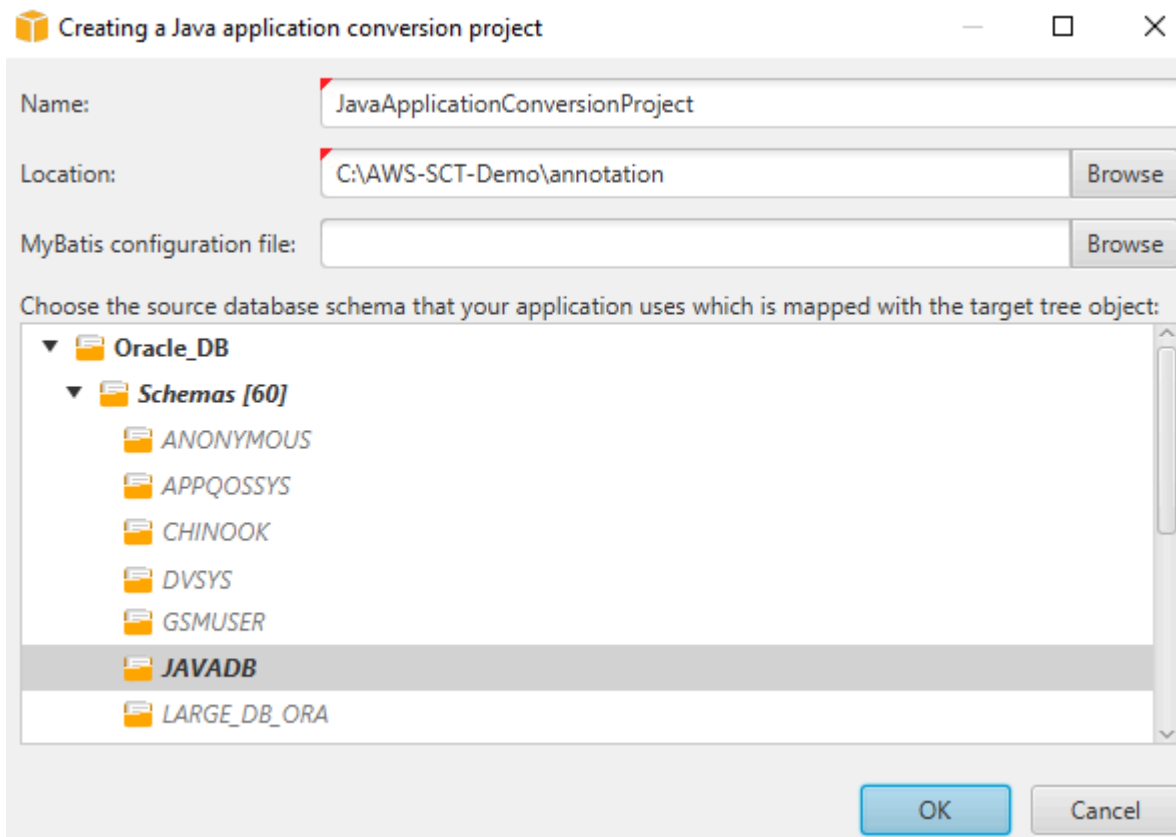
您只能為將 Oracle 資料庫結構描述轉換為 PostgreSQL 資料庫結構描述而建立 Java 應用程式轉換專案。請務必在專案中新增對應規則，其中包含來源 Oracle 結構描述和目標 PostgreSQL 資料庫。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

您可以在一個單一新增多個應用程式轉換專案AWS SCT項目 使用下列程序建立 Java 應用程式轉換專案。

若要建立 Java 應用程式轉換專案

1. 建立資料庫轉換專案，並加入來源 Oracle 資料庫。如需詳細資訊，請參閱 [建立AWS SCT專案](#) 及 [將資料庫伺服器新增至AWS SCT專案](#)。
2. 新增包含來源 Oracle 資料庫和目標 PostgreSQL 資料庫的對應規則。您可以新增目標 PostgreSQL 資料庫，或在對應規則中使用虛擬 PostgreSQL 目標資料庫平台。如需詳細資訊，請參閱 [在中建立對應規則 AWS SCT](#) 及 [使用虛擬目標](#)。
3. 在「」檢視功能表中，選擇主視圖。
4. 在「」应用功能表中，選擇新的 Java 應用程式。

該建立 Java 應用程式轉換專案出現對話方塊。



5. 對於姓名」，輸入 Java 應用程式轉換專案的名稱。由於每個資料庫結構描述轉換專案都可以有一或多個子應用程式轉換專案，因此請選擇加入多個專案時合理的名稱。
6. 對於位置下，輸入應用程式的原始碼位置。
7. (選擇性) 對於MyBatis 組態檔，輸入路徑 MyBatis 組態檔。AWS SCT掃描您的應用程序文件夾以自動發現此文件。如果此檔案不在您的應用程式資料夾中，或者您使用多個設定檔，請手動輸入路徑。
8. 在來源樹狀結構中，選擇應用程式使用的結構描述。請確定此綱要是對應規則的一部分。AWS SCT以粗體亮顯屬於對映規則一部分的資料架構。
9. 選擇好以建立您的 Java 應用程式轉換專案。
10. 在「」中尋找您的 Java 應用程式轉換專案应用左側面板中的節點。

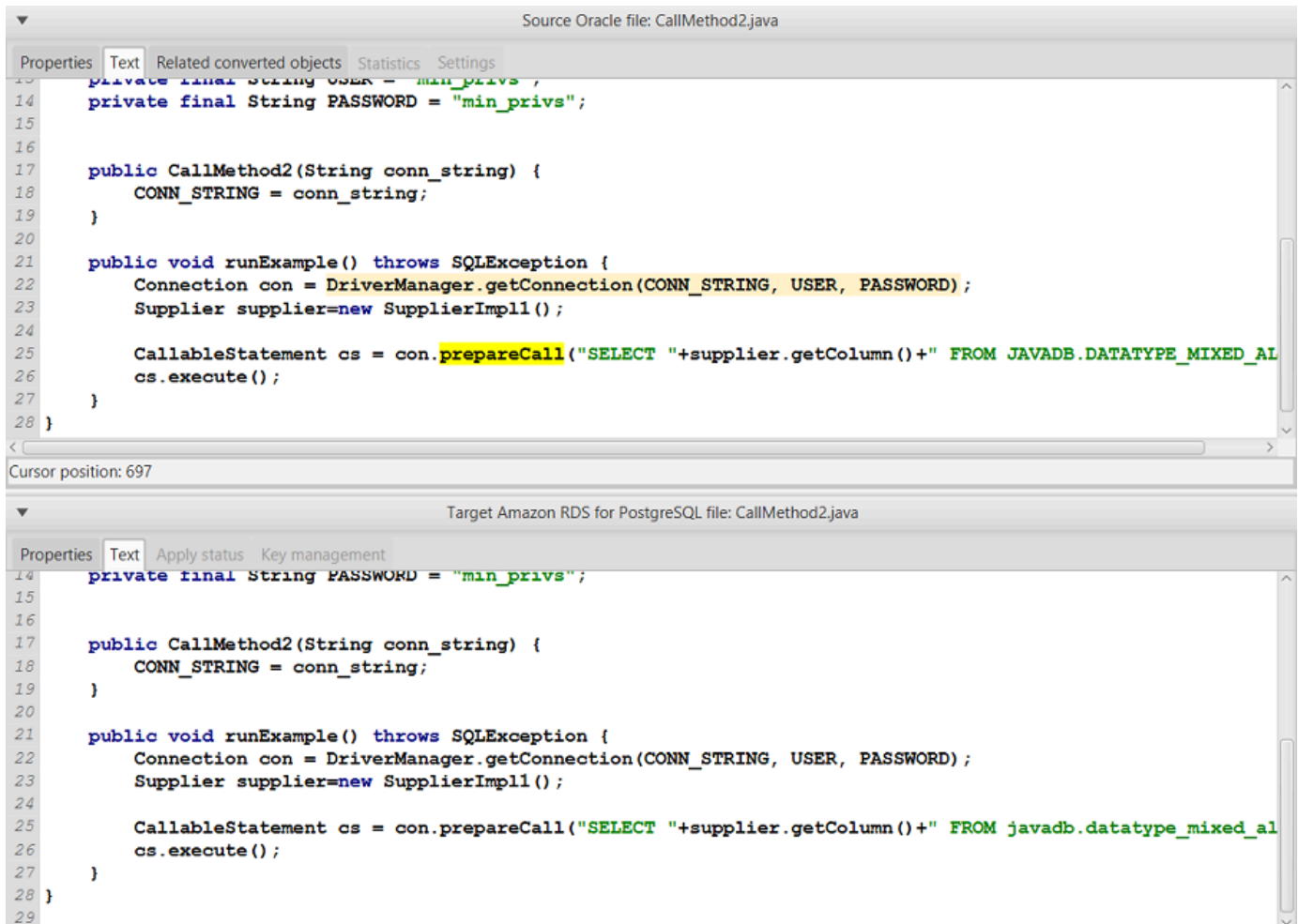
將您的 Java 應用程式 SQL 程式碼轉換為AWS SCT

在您將 Java 應用程式新增至AWS SCT項目，將 SQL 代碼從此應用程序轉換為與目標數據庫平台兼容的格式。使用下列程序來分析及轉換內嵌在 Java 應用程式中的 SQL 程式碼AWS Schema Conversion Tool。

若要轉換 SQL 程式碼

1. 展開爪哇節點下应用在左側面板中。
2. 選擇要轉換的應用程式，開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇轉換。AWS SCT會分析您的原始程式碼檔案、決定應用程式邏輯，並將程式碼中繼資料載入專案。此程式碼中繼資料包括 Java 類別、物件、方法、全域變數、介面等。

目標資料庫目標資料庫。AWS SCT會建立與來源應用程式專案類似的資料夾結構。在這裡您可以查看轉換後的應用程序代碼。



```
Source Oracle file: CallMethod2.java
13 private final String USER = "min_privs";
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new 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
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl1();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26     cs.execute();
27 }
28 }
29 }
```

4. 保存轉換後的應用程序代碼。如需詳細資訊，請參閱[儲存轉換後的應用程式碼](#)。

您的 Java 應用程式可能包含與不同來源資料庫互動的 SQL 程式碼。您可以移轉至 PostgreSQL 的數個這些來源資料庫。在這種情況下，請確定您不會轉換與從移轉範圍排除的資料庫互動的 SQL 程式碼。您可以從轉換範圍中排除 Java 應用程式的來源檔案。若要這麼做，請清除您要從轉換範圍中排除的檔案名稱的核取方塊。

變更轉換範圍之後，AWS SCT 仍然會分析 Java 應用程式的所有源文件的 SQL 代碼。然後 AWS SCT 將您從轉換範圍中排除的所有來源檔案複製到目標資料夾。此作業可讓您在儲存轉換後的應用程式檔案之後建置應用程式。

保存轉換後的應用程式代碼 AWS SCT

使用下列程序儲存轉換的應用程式程式碼。

儲存轉換後的應用程式代碼

1. 展開爪哇節點下应用目標資料庫目標資料庫。
2. 選擇轉換後的應用程式，然後選擇儲存。
3. 輸入要儲存轉換後的應用程式程式碼的資料夾路徑，然後選擇選取資料庫。

如果您的來源 Java 應用程式使用 MyBatis 框架，請確保您更新配置文件以使用新數據庫。

管理 Java 應用程式轉換專案 AWS SCT

您可以添加多個 Java 應用程式轉換項目，更新應用程式代碼 AWS SCT 專案，或從您的專案中移除 Java 轉換專案 AWS SCT 項目

若要新增其他 Java 應用程式轉換專案

1. 展開应用左側面板中的節點。
2. 選擇爪哇 node (我的電腦)，開啟內容功能表 (按一下右鍵)。
3. 選擇 New application (新增應用程式)。
4. 輸入建立新 Java 應用程式轉換專案所需的資訊。如需詳細資訊，請參閱 [建立 Java 應用程式轉換專案](#)。

在原始應用程式程式碼中進行變更之後，請將其上傳至 AWS SCT 項目

上傳更新的應用程式代碼

1. 展開爪哇節點下应用在左側面板中。
2. 選擇要更新的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刷新然後選擇是。

AWS SCT從來源檔案上傳您的應用程式程式碼，並移除轉換結果。若要保留您在中所做的程式碼變更AWS SCT和轉換結果，創建一個新的 Java 轉換項目。

如果您的來源 Java 應用程式使用 MyBatis 框架AWS SCT使用 MyBatis 配置文件來解析您的 SQL 代碼。變更此檔案之後，請將其上傳至AWS SCT項目

編輯路徑 MyBatis 組態檔

1. 展開爪哇節點下应用在左側面板中。
2. 選擇您的應用程式，然後選擇設定。
3. 選擇瀏覽，然後選擇 MyBatis組態檔。
4. 選擇 Apply (套用)。
5. 在左側面板中，選擇應用程式，開啟內容 (按一下滑鼠右鍵) 選單，然後選擇刷新。

若要移除 Java 應用程式轉換專案

1. 展開爪哇節點下应用在左側面板中。
2. 選擇要移除的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刪除然後選擇好。

在中建立 Java 應用程式轉換評估報告AWS SCT

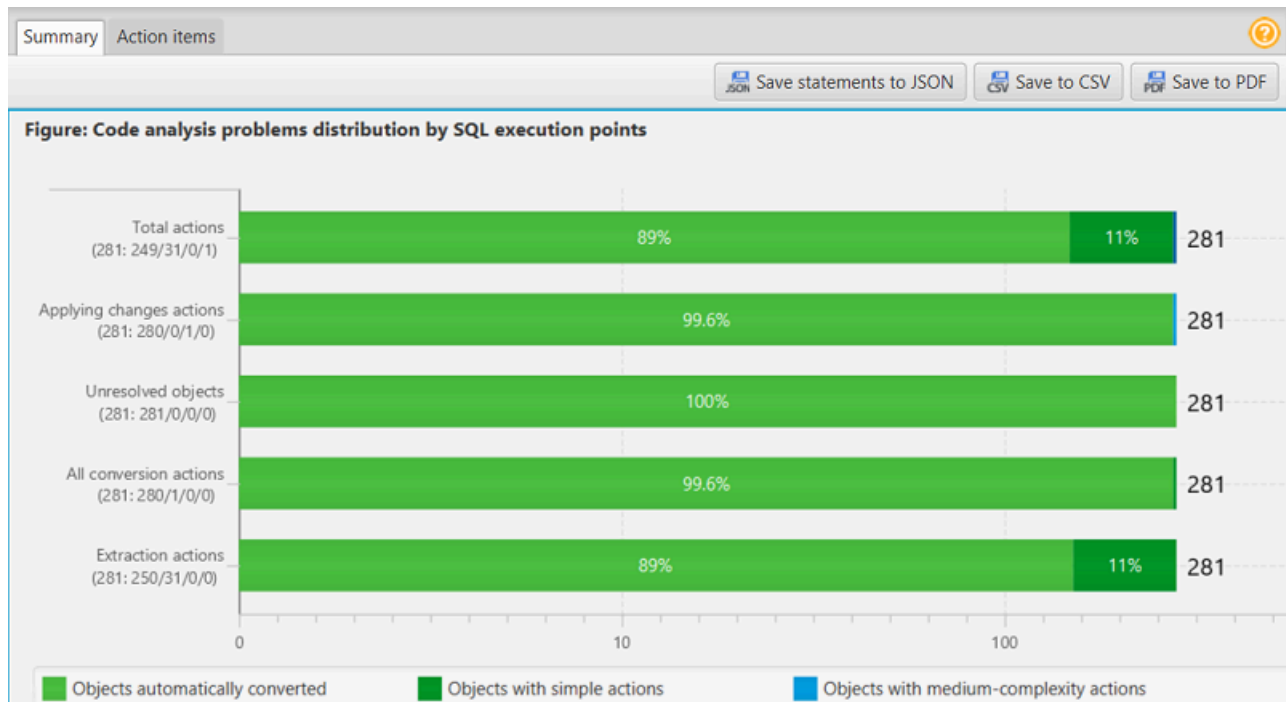
該Java 應用程式轉換評估報告提供有關將 Java 應用程式中嵌入的 SQL 程式碼轉換為與目標資料庫相容之格式的資訊。評估報告提供所有 SQL 執行點和所有原始程式碼檔案的轉換詳細資訊。評估報告也包含 SQL 程式碼的行動項目，AWS SCT無法轉換。

使用下列程序建立 Java 應用程式轉換評估報表。

建立 Java 應用程式轉換評估報表

1. 展開爪哇節點下应用在左側面板中。
2. 選擇要轉換的應用程式，開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇轉換。
4. 在「」檢視功能表中，選擇評估報告檢視。
5. 檢閱 Summary (摘要) 索引標籤。

該摘要標籤 (如下所示) 顯示來自 Java 應用程式評估報告的執行摘要資訊。它顯示了所有 SQL 執行點和所有源代碼文件的轉換結果。



6. 選擇將陳述式儲存為 JSON 將 Java 應用程式中提取的 SQL 代碼保存為 JSON 文件。

7. (選擇性) 將報表的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：

- 選擇儲存筆記本位於右上角，將報告另存為 PDF 檔案。

PDF 檔案包含執行摘要、行動項目和應用程式轉換的建議。

- 選擇儲存資料庫。位於右上角，以將報告儲存為 CSV 檔案。

CSV 檔案包含行動項目、建議的動作，以及轉換 SQL 程式碼所需手動工作的估計複雜度。

在 Pro*C 應用程式中轉換 SQL 代碼 AWS SCT

對於甲骨文到 PostgreSQL 的轉換，您可以使用 AWS Schema Conversion Tool (AWS SCT) 以轉換嵌入到 Pro*C 應用程式中的 SQL 程式碼。這個特定的 Pro*C 應用程式轉換器了解應用程式邏輯。它會收集位於不同應用程式物件中的陳述式，例如函數、參數、區域變數等。

由於這種深入的分析，Pro*C 應用程式 SQL 程式碼轉換器提供了比通用轉換器更好的轉換結果。

在中建立 Pro*C 應用程式轉換專案AWS SCT

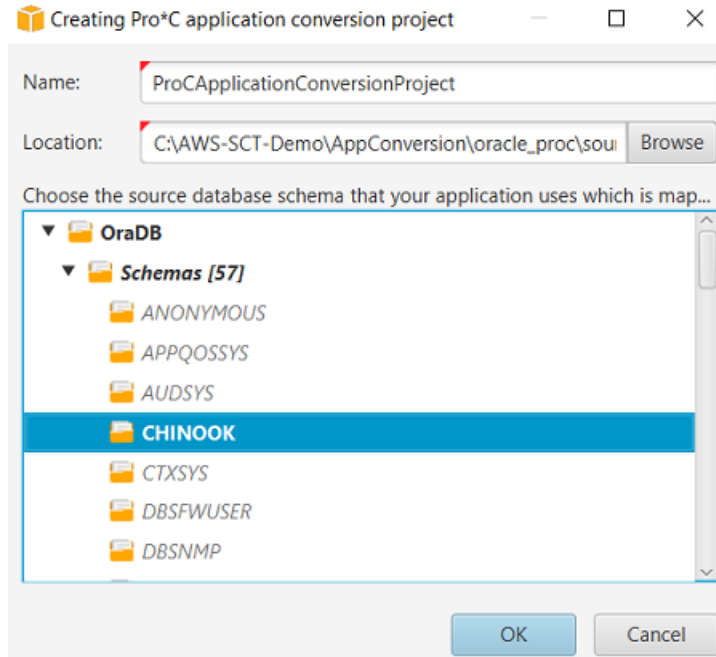
您只能建立 Pro*C 應用程式轉換專案，以將 Oracle 資料庫結構描述轉換為 PostgreSQL 資料庫結構描述。請務必在專案中新增對應規則，其中包含來源 Oracle 結構描述和目標 PostgreSQL 資料庫。如需詳細資訊，請參閱[在中建立對應規則 AWS SCT](#)。

您可以在一個單一新增多個應用程式轉換專案AWS SCT項目 使用下列程序建立 Pro*C 應用程式轉換專案。

若要建立 Pro*C 應用程式轉換專案

1. 建立資料庫轉換專案，並加入來源 Oracle 資料庫。如需詳細資訊，請參閱 [建立AWS SCT專案](#) 及 [將資料庫伺服器新增至AWS SCT專案](#)。
2. 新增包含來源 Oracle 資料庫和目標 PostgreSQL 資料庫的對應規則。您可以新增目標 PostgreSQL 資料庫，或在對應規則中使用虛擬 PostgreSQL 目標資料庫平台。如需詳細資訊，請參閱 [在中建立對應規則 AWS SCT](#) 及 [使用虛擬目標](#)。
3. 在「」檢視功能表中，選擇主視圖。
4. 在「」應用功能表中，選擇新的 Pro*C 應用程序。

該建立 Pro*C 應用程式轉換專案出現對話方塊。



5. 對於姓名中，輸入 Pro*C 應用程式轉換專案的名稱。由於每個資料庫結構描述轉換專案都可以有一或多個子應用程式轉換專案，因此請選擇加入多個專案時合理的名稱。
6. 對於位置下，輸入應用程式的原始碼位置。

7. 在來源樹狀結構中，選擇應用程式使用的結構描述。請確定此綱要是對應規則的一部分。AWS SCT以粗體亮顯屬於對映規則一部分的資料架構。
8. 選擇好創建您的 Pro*C 應用程序轉換項目。
9. 尋找您的 Pro*C 應用程式轉換專案應用左側面板中的節點。

將您的 Pro*C 應用程式 SQL 程式碼轉換為AWS SCT

將 Pro*C 應用程式新增至AWS SCT項目，將 SQL 代碼從此應用程序轉換為與目標數據庫平台兼容的格式。使用下列程序來分析及轉換內嵌在 Pro*C 應用程式中的 SQL 程式碼AWS Schema Conversion Tool。

若要轉換 SQL 程式碼

1. 展開設定檔節點下應用在左側面板中。
2. 選擇要轉換的應用程序，然後選擇設定。
 - a. 對於全域標頭檔案路徑」下方，輸入應用程式專案所使用之標頭檔案的路徑。
 - b. 選擇將所有未解析的主機變量解釋為以查看轉換後的代碼中所有未解析的變量。
 - c. 選擇使用擴展包中的固定寬度字符串轉換功能在轉換後的 SQL 代碼中使用擴展包函數。AWS SCT在您的應用程式專案中包含擴充套件檔案。
 - d. 選擇將名PL/SQL 塊轉換為獨立的 SQL 調用或存儲的函數在目標資料庫中為所有匿名 PL/SQL 區塊建立預存程序。AWS SCT然後在轉換後的應用程式程式碼中包含這些預存程序的執行。
 - e. 選擇使用自訂游標流以改善 Oracle 資料庫游標的轉換。
3. 在左側面板中，選擇要轉換的應用程序，然後開啟內容 (按一下滑鼠右鍵) 選單。
4. 選擇轉換。AWS SCT會分析您的原始程式碼檔案、決定應用程式邏輯，並將程式碼中繼資料載入專案。此程式碼中繼資料包括 Pro*C 類別、物件、方法、全域變數、介面等。

目標資料庫目標資料庫。AWS SCT會建立與來源應用程式專案類似的資料夾結構。在這裡您可以查看轉換後的應用程序代碼。

The screenshot displays two panels of code. The top panel, titled 'Source Oracle app function: main() int', shows Oracle Pro*C code with lines 11 through 20. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted PostgreSQL code with lines 8 through 18. The converted code uses 'EXEC SQL' to execute Pro*C statements within a PostgreSQL context.

```
Source Oracle app function: main() int
11  int i = 5;
12
13  /* Connect string */
14
15  EXEC SQL INSERT INTO embeddedc.t_insert(i) VALUES (:i);
16
17  EXEC SQL COMMIT;
18
19  return 0;
20 }
```

Cursor position: 118 Click position: 197

```
Target Amazon RDS for PostgreSQL app function: main() int
8
9  EXEC SQL int i = 5;
10
11  /* Connect string */
12
13  EXEC SQL INSERT INTO embeddedc.t_insert (i)
14  VALUES (:i);
15
16  EXEC SQL COMMIT;
17
18  return 0;
```

5. 儲存轉換後的應用程式代碼。如需詳細資訊，請參閱[編輯並儲存轉換後的應用程式碼](#)。

編輯和保存轉換後的應用程序代碼AWS SCT

您可以編輯轉換後的 SQL 語句並使用AWS SCT將此編輯的代碼嵌入到轉換後的 Pro*C 應用程序代碼中。使用下列程序編輯轉換的 SQL 程式碼。

若要編輯已轉換的 SQL 程式碼

1. 展開設定檔節點下應用在左側面板中。
2. 選擇要轉換的應用程式，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇轉換。
3. 在「」檢視功能表中，選擇評估報告檢視。
4. 選擇儲存陳述式資料庫。將 Pro*C 應用程式中擷取的 SQL 程式碼儲存為 CSV 檔案。
5. 輸入 CSV 文件的名稱以保存提取的 SQL 代碼，然後選擇儲存。
6. 編輯擷取的 SQL 程式碼。
7. 在「」檢視功能表中，選擇主視圖。
8. 展開設定檔節點下應用目標資料庫目標資料庫。
9. 選擇轉換後的應用程式，開啟內容 (按一下滑鼠右鍵) 選單，然後選擇匯入資料庫資料庫。。
10. 選擇是，然後選擇包含您編輯過的 SQL 代碼的文件，然後選擇打開。

AWS SCT將轉換後的 SQL 語句分成幾部分，並將其放入源應用程序代碼的適當對象中。使用下列程序儲存轉換的應用程式程式碼。

儲存轉換後的應用程式代碼

1. 展開設定檔節點下應用在目標資料庫面板中。
2. 選擇轉換後的應用程序，然後選擇儲存。
3. 輸入要儲存轉換後的應用程式程式碼的資料夾路徑，然後選擇選取資料庫。

管理 Pro*C 應用程式轉換專案AWS SCT

您可以添加多個 Pro*C 應用程序轉換項目，更新應用程序代碼AWS SCT項目，或從您的AWS SCT項目

若要新增其他 Pro*C 應用程式轉換專案

1. 展開應用左側面板中的節點。
2. 選擇設定檔node (我的電腦)，開啟內容功能表 (按一下右鍵)。
3. 選擇 New application (新增應用程式)。
4. 輸入建立新 Pro*C 應用程式轉換專案所需的資訊。如需詳細資訊，請參閱[建立 Pro*C 應用程式轉換專案](#)。

在原始應用程式程式碼中進行變更之後，請將其上傳至AWS SCT項目

上傳更新的應用程式代碼

1. 展開設定檔節點下應用在左側面板中。
2. 選擇要更新的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。
3. 選擇刷新然後選擇是。

AWS SCT從來源檔案上傳您的應用程式程式碼，並移除轉換結果。若要保留您在中所做的程式碼變更AWS SCT和轉換結果，創建一個新的 Pro*C 轉換項目。

若要移除 Pro*C 應用程式轉換專案

1. 展開設定檔節點下應用在左側面板中。
2. 選擇要移除的應用程式，然後開啟內容選單 (按一下滑鼠右鍵)。

3. 選擇刪除然後選擇好。

在中建立 Pro*C 應用程式轉換評估報告AWS SCT

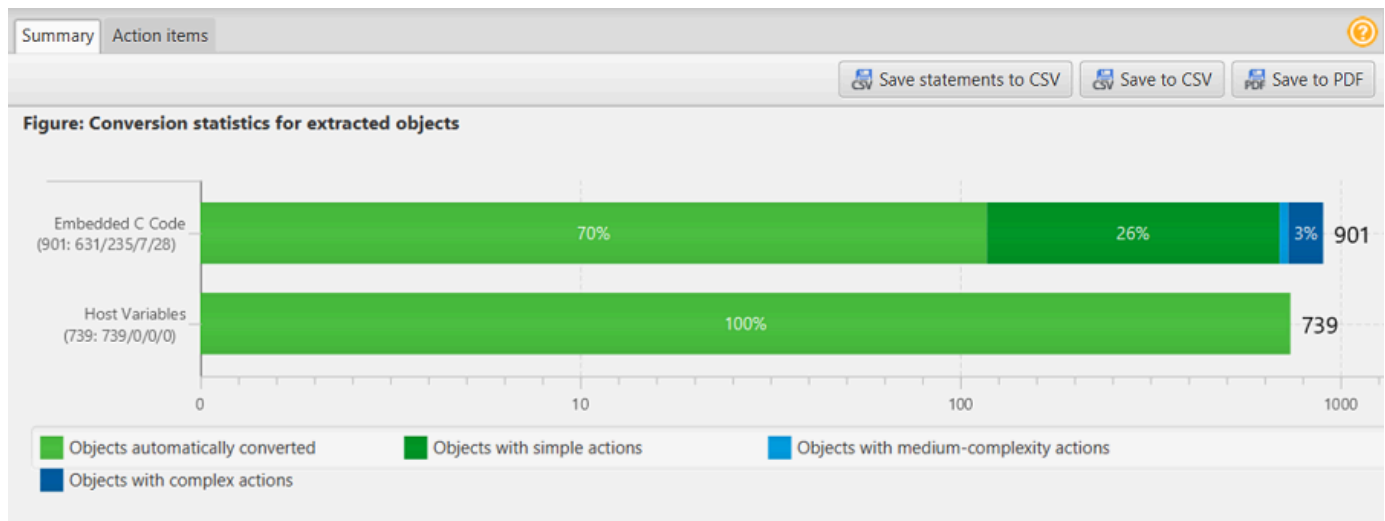
該Pro*C 應用程式轉換評估報告提供有關將 Pro*C 應用程式中嵌入的 SQL 程式碼轉換為與目標資料庫相容的格式的資訊。評估報告提供所有 SQL 執行點和所有原始程式碼檔案的轉換詳細資訊。評估報告也包含 SQL 程式碼的行動項目，AWS SCT無法轉換。

使用下列程序建立 Pro*C 應用程式轉換評估報告。

若要建立 Pro*C 應用程式轉換評估報告

1. 展開設定檔節點下應用在左側面板中。
2. 選擇要轉換的應用程式，開啟內容 (按一下滑鼠右鍵) 功能表。
3. 選擇轉換。
4. 在「」檢視功能表中，選擇評估報告檢視。
5. 檢閱 Summary (摘要) 索引標籤。

該摘要標籤 (如下所示) 顯示 Pro*C 應用程式評估報告中的執行摘要資訊。它顯示了所有 SQL 執行點和所有源代碼文件的轉換結果。



6. 選擇儲存陳述式資料庫。 ，將 Pro*C 應用程式中擷取的 SQL 程式碼儲存為逗號分隔值 (CSV) 檔案。
7. (選擇性) 將報表的本機副本儲存為 PDF 檔案或逗號分隔值 (CSV) 檔案：
 - 選擇儲存資料庫位於右上角，將報告另存為 PDF 檔案。

PDF 檔案包含執行摘要、行動項目和應用程式轉換的建議。

- 選擇儲存資料庫。位於右上角，以將報告儲存為 CSV 檔案。

CSV 檔案包含行動項目、建議的動作，以及轉換 SQL 程式碼所需手動工作的估計複雜度。

使用 AWS SCT 擴充套件

AWS SCT 擴充套件是一種附加元件模組，可模擬將物件轉換為目標資料庫時所需的來源資料庫中存在的函數。在您可以安裝 AWS SCT 擴充套件之前，請先轉換資料庫結構描述。

每個 AWS SCT 擴展包包括以下組件：

- 資料庫結構描述 — 包含用於模擬特定線上交易處理 (OLTP) 和線上分析處理 (OLAP) 資料庫物件 (例如序列) 的 SQL 函數、程序和表格。此外，模擬來源資料庫 built-in-functions 中不受支援的功能。此結構描述的名稱格式如下：`aws_<database_engine_name>_ext`
- AWS Lambda 函數 (適用於特定 OLTP 資料庫) — 包含模擬複雜資料庫功能的 AWS Lambda 函數，例如作業排程和傳送電子郵件。
- OLAP 資料庫的自訂程式庫 — 包括一組 Java 和 Python 程式庫，您可以使用這些程式庫將擷取、轉換和載入 (ETL) 指令碼移轉至或。AWS Glue AWS Glue Studio

Java 庫包括以下模塊：

- `spark-excel_2.11-0.13.1.jar`— 模擬 Excel 來源元件和目標元件的功能。
- `spark-xml_2.11-0.9.0.jar`、`poi-ooxml-schemas-4.1.2.jar`、和 `xmlbeans-3.1.0.jar` — 模擬 XML 來源元件的功能。

Python 庫包括以下模塊：

- `sct_utils.py`— 模擬來源資料類型並準備 Spark SQL 查詢的參數。
- `ssis_datetime.py`— 模擬日期和時間內置功能。
- `ssis_null.py`— 模擬 ISNULL 和內 REPLACENULL 置功能。
- `ssis_string.py`-模擬字符串內置函數。

如需這些程式庫的詳細資訊，請參閱 [使用 AWS SCT 擴充套件的自訂程式庫](#)。

您可以使用兩種方式套用 AWS SCT 擴充功能套件：

- AWS SCT 當您從內容功能表中選擇套用至資料庫，套用目標資料庫指令集時，可以自動套用擴充套件。AWS SCT 在套用所有其他綱要物件之前，先套用擴充功能套件。
- 若要手動套用擴充功能套件，請選擇目標資料庫，然後從內容功能表 (按一下滑鼠右鍵) 選擇 [套用擴充功能套件]。在大多數情況下，自動應用就足夠了。不過，如果意外刪除套件，您可能會想要手動套用套件。

每次將 AWS SCT 擴充功能套件套用至目標資料倉庫時，元件都會遭到覆寫，並 AWS SCT 顯示有關此項目的通知。若要關閉這些通知，請選擇 [設定]、[全域設定]、[通知]，然後選取 [隱藏延伸套件更換警示]。

對於從 Microsoft SQL 服務器轉換為 PostgreSQL，您可以使用 SQL 服務器到 PostgreSQL 擴展包。AWS SCT 此延伸套件會模擬 SQL 伺服器代理程式和 SQL 伺服器資料庫郵件。如需詳細資訊，請參閱 [使用擴充套件在 PostgreSQL 中模擬 SQL 伺服器代理程式](#) 及 [使用擴充套件在 PostgreSQL 中模擬 SQL 伺服器資料庫郵件](#)。

接下來，您可以找到有關使用 AWS SCT 擴充功能套件的詳細資訊。

主題

- [使用 AWS SCT 擴充功能套件的權限](#)
- [使用擴充套件結構描述](#)
- [使用 AWS SCT 擴充套件的自訂程式庫](#)
- [使用 AWS SCT 擴展包中的 AWS Lambda 功能](#)
- [設定 AWS SCT 擴充套件的功能](#)

使用 AWS SCT 擴充功能套件的權限

Amazon Aurora 的 AWS SCT 擴充套件可使用功能模擬郵件傳送、任務排程、佇列和其他操作 AWS Lambda。將 AWS SCT 擴充套件套用至目標 Aurora 資料庫時，AWS SCT 會建立新的 AWS Identity and Access Management (IAM) 角色和內嵌 IAM 政策。接下來，AWS SCT 建立新的 Lambda 函數，並設定 Aurora 資料庫叢集以進行輸出連線。AWS Lambda 若要執行這些作業，請務必將下列必要許可授與 IAM 使用者：

- `iam:CreateRole`— 為您的 AWS 帳戶建立新的 IAM 角色。
- `iam:CreatePolicy`— 為您的 AWS 帳戶建立新的 IAM 政策。
- `iam:AttachRolePolicy`— 將指定的政策附加到您的 IAM 角色。
- `iam:PutRolePolicy`— 更新內嵌在 IAM 角色中的內嵌政策文件。
- `iam:PassRole`— 將指定的 IAM 角色傳遞給規則引擎。
- `iam:TagRole`— 將標籤新增至 IAM 角色。
- `iam:TagPolicy`— 將標籤新增至 IAM 政策。
- `lambda:ListFunctions`— 查看您的 Lambda 函數列表。

- `lambda:ListTags`-查看 Lambda 函數的標籤列表。
- `lambda:CreateFunction`-創建一個新的 Lambda 函數。
- `rds:AddRoleToDBCluster`— 將 IAM 角色與您的 Aurora 資料庫叢集建立關聯。

Amazon Redshift 的 AWS SCT 擴充套件會模擬將轉換後的物件套用至 Amazon Redshift 時所需的來源資料倉儲基礎功能。在將轉換後的程式碼套用至 Amazon Redshift 之前，您必須先套用適用於 Amazon Redshift 的擴充套件。若要這麼做，請在 IAM 政策中包含該 `iam:SimulatePrincipalPolicy` 動作。

AWS SCT 使用 IAM 政策模擬器來檢查安裝 Amazon Redshift 擴充套件所需的許可。即使您已正確設定 IAM 使用者，IAM 政策模擬器仍會顯示錯誤訊息。這是 IAM 政策模擬器的已知問題。此外，當您的 IAM 政策中沒有 `iam:SimulatePrincipalPolicy` 動作時，IAM 政策模擬器會顯示錯誤訊息。在這些情況下，您可以忽略錯誤訊息，並使用擴充功能套件精靈套用擴充功能套件。如需詳細資訊，請參閱 [套用擴充功能套件](#)。

使用擴充套件結構描述

當您轉換您的資料庫或資料倉儲結構描述，AWS SCT 會新增額外的結構描述到您的目標資料庫。此結構描述會實作來源資料庫的 SQL 系統功能，當您將已轉換的結構描述寫入至目標資料庫時需要這些功能。這個額外的結構描述稱為延伸套件結構描述。

OLTP 資料庫的延伸套件結構描述的命名是根據來源資料庫，如下所示：

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL : `AWS_POSTGRESQL_EXT`

OLAP 資料倉儲應用程式的延伸套件結構描述的命名是根據來源資料存放區，如下所示：

- 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

使用 AWS SCT 擴充套件的自訂程式庫

在某些情況下，AWS SCT 無法將來源資料庫功能轉換為目標資料庫中的對等功能。相關的 AWS SCT 擴充套件包含可模擬目標資料庫中某些來源資料庫功能的自訂程式庫。

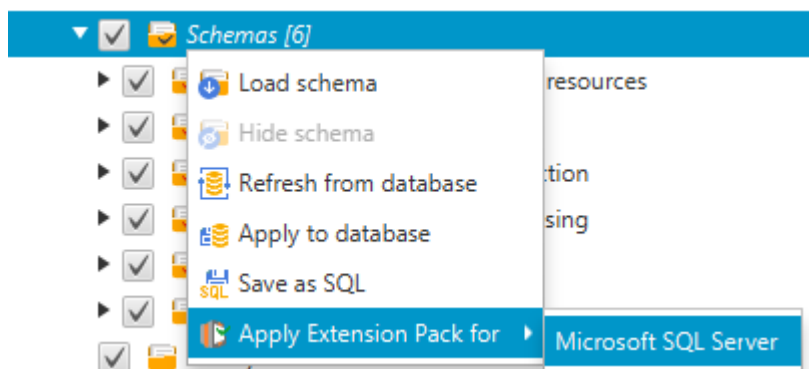
如果要轉換交易式資料庫，請參閱[使用 AWS SCT 擴展包中的 AWS Lambda 功能](#)。

套用擴充功能套件

您可以使用 AWS SCT 擴充套件精靈或將轉換後的程式碼套用至目標資料庫時，套用擴充功能套件。

若要使用擴充套件精靈套用擴充功能套件

1. 在目標資料庫樹狀結構中 AWS Schema Conversion Tool，開啟相關資訊環境 (按一下滑鼠右鍵) 功能表，選擇 [套用擴充功能套件]，然後選擇您的來源資料庫平台。



會顯示延伸套件精靈。

2. 閱讀 Welcome (歡迎) 頁面，然後選擇 Next (下一步)。
3. 在設定AWS 檔設定頁面上，執行下列動作：
 - 如果您只是要重新安裝延伸套件結構描述，請選擇 Skip this step for now (暫時略過此步驟)，然後選擇 Next (下一步)。「立即略過此步驟」選項僅適用於線上交易處理 (OLTP) 資料庫。
 - 如果您要上傳新的程式庫，請提供認證以連線到您的 AWS 帳戶。只有當您轉換 OLAP 資料庫或 ETL 指令碼時，才使用此步驟。如果您已 AWS CLI 安裝，則可以使用您的 AWS Command Line Interface (AWS CLI) 憑據。您也可以使用先前存放在全域應用程式設定的設定檔中，並與專案建立關聯的登入資料。如有必要，請選擇「導覽至全域設定」，以規劃將不同的設定檔與 AWS SCT 專案相關聯。如需詳細資訊，請參閱[將AWS服務設定檔儲存在AWS SCT](#)。

4. 如果您要上傳新的資料庫，請選擇「圖庫上傳」頁面上的「我需要上傳圖庫」。只有當您轉換 OLAP 資料庫或 ETL 指令碼時，才使用此步驟。接下來，提供 Amazon S3 路徑，然後選擇「上傳程式庫到 S3」。

如果您已經上傳程式庫，請選擇 [我已上傳程式庫]，在 [程式庫上傳] 頁面上使用我現有的 S3 儲存貯體。接下來，提供 Amazon S3 路徑。

完成時請選擇 Next (下一步)。

5. 在 [函數模擬] 頁面上，選擇 [建立擴充功能套件]。會出現訊息，包含延伸套件操作的狀態。

完成時請選擇 Finish (完成)。

套用轉換後的程式碼時套用擴充功能套件

1. 在您的 AWS 服務設定檔中指定 Amazon S3 儲存貯體。只有當您轉換 OLAP 資料庫或 ETL 指令碼時，才使用此步驟。如需詳細資訊，請參閱 [將AWS服務設定檔儲存在AWS SCT](#)。

請確定您的 Amazon S3 儲存貯體政策包含下列許可：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["iam:SimulatePrincipalPolicy"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["iam:GetUser"],
      "Resource": ["arn:aws:iam::111122223333:user/DataExtractionAgentName"]
    }
  ]
}
```



```
    }  
  ]  
}
```

在前面的範例中，將 `11112222333####/#####` `DataExtractionAgentName` 的名稱。

2. 轉換來源資料倉儲結構描述。如需詳細資訊，請參閱 [將資料倉儲結構描述轉換為亞馬遜紅移](#)。
3. 在右窗格中，選擇轉換後的結構描述。
4. 開啟結構描述元素的內容 (按一下右鍵) 選單，然後選擇 Apply to database (套用至資料庫)。
5. AWS SCT 產生具有必要元件的擴充套件，並在目標樹狀 `aws_database_engine_name_ext` 結構中新增結構描述。接下來，AWS SCT 將轉換後的代碼和擴展包結構描述應用到目標數據倉庫。

當您使用 Amazon Redshift 和 AWS Glue 做為目標資料庫平台的組合時，會在擴充功能套件中 AWS SCT 新增額外的結構描述。

使用 AWS SCT 擴展包中的 AWS Lambda 功能

AWS SCT 提供延伸套件，其中包含用於 Amazon EC2 上託管之資料庫的電子郵件、任務排程和其他功能的 Lambda 函數。

使用 AWS Lambda 函數來模擬數據庫功能

在某些情況下，資料庫功能無法轉換為相等的 Amazon RDS 功能。例如，Oracle 傳送使用 UTL_SMTP 的電子郵件呼叫，而 Microsoft SQL Server 可以使用任務排程工具。如果您在 Amazon EC2 上託管和自行管理資料庫，則可以透過替換 AWS 服務來模擬這些功能。

AWS SCT 擴充套件精靈可協助您安裝、建立和設定 Lambda 函數，以模擬電子郵件、工作排程和其他功能。

套用擴充套件以支援 Lambda 函數

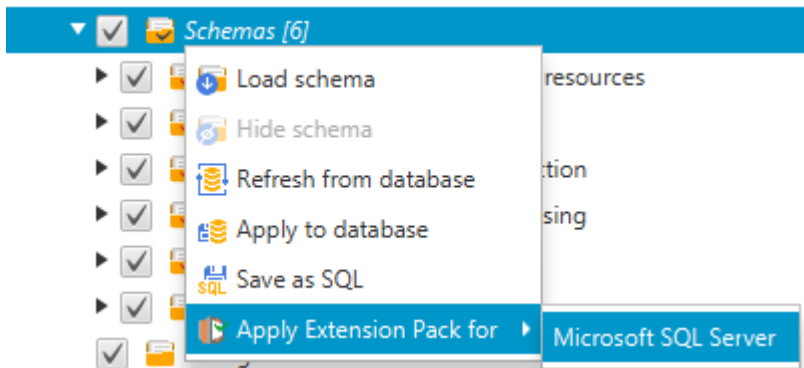
您可以使用擴充套件精靈或將轉換後的程式碼套用至目標資料庫時，套用擴充套件來支援 Lambda 函數。

⚠ Important

只有在 Amazon EC2 上安裝和自行管理的資料庫才支援 AWS 服務模擬功能。如果您的目標資料庫位於 Amazon RDS 資料庫執行個體上，請勿安裝服務模擬功能。

若要使用擴充套件精靈套用擴充功能套件

1. 在目標資料庫樹狀結構中 AWS Schema Conversion Tool，開啟相關資訊環境 (按一下滑鼠右鍵) 功能表，選擇 [套用擴充功能套件]，然後選擇您的來源資料庫平台。



會顯示延伸套件精靈。

2. 閱讀 Welcome (歡迎) 頁面，然後選擇 Next (下一步)。
3. 在設定AWS 檔設定頁面上，執行下列動作：
 - 如果您只是要重新安裝延伸套件結構描述，請選擇 Skip this step for now (暫時略過此步驟)，然後選擇 Next (下一步)。
 - 如果您要安裝 AWS 服務，請提供認證以連線到您的 AWS 帳戶。如果您已 AWS CLI 安裝，則可以使用您的 AWS CLI 憑據。您也可以使用先前存放在全域應用程式設定的設定檔中，並與專案建立關聯的登入資料。如有需要，選擇 Navigate to Project Settings (前往專案設定) 將不同的設定檔關聯至專案。如有需要，可以選擇 Global Settings (全域設定) 來建立新的設定檔。如需詳細資訊，請參閱 [將AWS服務設定檔儲存在AWS SCT](#)。
4. 在 Email Sending Service (電子郵件傳送服務) 頁面上，執行下列步驟：
 - 如果您只是要重新安裝延伸套件結構描述，請選擇 Skip this step for now (暫時略過此步驟)，然後選擇 Next (下一步)。
 - 如果您正在安裝 AWS 服務，並且擁有現有的 Lambda 函數，則可以提供它。否則，精靈會為您建立。完成時請選擇 Next (下一步)。
5. 在 Job Emulation Service (任務模擬服務) 頁面上，執行下列步驟：

- 如果您只是要重新安裝延伸套件結構描述，請選擇 Skip this step for now (暫時略過此步驟)，然後選擇 Next (下一步)。
 - 如果您正在安裝 AWS 服務，並且擁有現有的 Lambda 函數，則可以提供它。否則，精靈會為您建立。完成時請選擇 Next (下一步)。
6. 在 [函數模擬] 頁面上，選擇 [建立擴充功能套件]。會出現訊息，包含延伸套件操作的狀態。
- 完成時請選擇 Finish (完成)。

Note

若要更新擴充功能套件並覆寫舊的延伸功能套件元件，請確定您使用的是最新版本的 AWS SCT。如需詳細資訊，請參閱 [安裝、驗證和更新 AWS SCT](#)。

設定 AWS SCT 擴充套件的功能

擴展包包含使用前必須配置的功能。常數會 CONVERSION_LANG 定義服務套件使用的語言。這些功能可用於英語和德語。

若要將語言設定為英文或德文，請在函數程式碼中進行下列變更。查找以下常量聲明：

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

若要設定 CONVERSION_LANG 為英文，請將該行變更為以下內容：

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

若要設定 CONVERSION_LANG 為德文，請將該行變更為以下內容：

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

為下列功能設定此設定：

- 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`

AWS SCT 的最佳實務

尋找有關使用 AWS Schema Conversion Tool (AWS SCT) 的最佳實務和選項資訊。

設定其他記憶體

若要轉換大型資料庫結構描述 (例如具有 3,500 個預存程序的資料庫)，您可以設定AWS Schema Conversion Tool。

修改AWS SCT消耗的記憶體數量

1. 在 [設定] 功能表上，選擇 [全域設定]，然後選擇 [JVM 選項]。
2. 選擇 [編輯組態檔]，然後選擇文字編輯器以開啟組態檔案。
3. 編輯 JavaOptions 區段，設定可用的記憶體下限和上限。下列範例會將最小值設定為四 GB，最大值設定為 40 GB。

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

我們建議您將最小記憶體設定為至少 4 GB。

4. 儲存組態檔案，選擇「確定」，然後重新啟動AWS SCT以套用變更。

設定預設專案資料夾

AWS SCT使用專案資料夾來儲存專案檔案、儲存評估報告，以及儲存轉換後的程式碼。依預設，會將所有檔案AWS SCT儲存在應用程式資料夾中。您可以指定另一個資料夾做為預設專案資料夾。

變更預設專案資料夾的步驟

1. 在 [設定] 功能表上，選擇 [全域設定]，然後選擇 [檔案路徑]。
2. 對於「預設專案檔案路徑」，輸入預設專案資料夾的路徑。
3. 選擇 Apply (套用)，然後選擇 OK (確定)。

提高資料遷移速度

若要移轉大型資料集 (例如資料超過 1 TB 的一組表格)，您可能需要提高移轉速度。使用資料擷取代理程式時，資料移轉的速度取決於各種因素。這些因素包括目標 Amazon Redshift 叢集中的磁碟片段數量、遷移任務中的區塊檔案大小、執行資料擷取代理程式的電腦上可用 RAM 等等。

若要提高資料移轉速度，我們建議您執行數個測試移轉工作階段，其中包含較小的生產資料集。此外，我們建議您在具有至少 500 GB 大小的 SSD 的 PC 上執行資料擷取代理程式。在這些測試工作階段期間，變更不同的移轉參數會監控您的磁碟使用率，以找出可確保最高資料移轉速度的組態。然後，使用此設定移轉整個資料集。

增加記錄資訊

您可以增加轉換資料庫、指令碼和應用程式 SQLAWS SCT 時所產生的記錄資訊。雖然增加記錄資訊可能會降低轉換速度，但這些變更可協助您在發生錯誤時向 SupAWS port 人員提供可靠的資訊

AWS SCT將記錄儲存在您的本機環境中。您可以檢視這些記錄檔，並與 SupAWS portAWS SCT 人員或開發人員共用以進行疑難排解。

若要變更記錄設定

1. 在 [設定] 功能表上，選擇 [全域設定]，然後選擇 [記錄]。
2. 在記錄資料夾路徑中，輸入要從使用者介面儲存記錄的資料夾。
3. 對於主控台記錄資料夾路徑，請輸入要儲存AWS SCT命令列介面 (CLI) 記錄的資料夾。
4. 在記錄檔大小上限 (MB) 中，輸入單一記錄檔的大小 (以 MB 為單位)。檔案達到此限制後，AWS SCT會建立新的記錄檔。
5. 針對記錄檔數目上限，輸入要儲存的記錄檔數目。資料夾中的記錄檔數目達到此限制後，AWS SCT會刪除最舊的記錄檔。
6. 對於擷取器記錄檔下載路徑，請輸入要儲存資料擷取代理程式記錄檔的資料夾。
7. 對於卡桑德拉提取日誌路徑，輸入存儲 Apache 卡桑德拉數據提取代理日誌的文件夾。
8. 選取「載入前詢問路徑」，以確定每次使用資料擷取代理程式時，都AWS SCT會詢問記錄檔儲存位置。
9. 針對偵錯模式，選擇 True。當標準AWS SCT記錄檔未包含任何問題時，請使用此選項來記錄其他資訊。
10. 選擇關鍵的應用程式模組來增加記錄資訊。您可以增加下列應用程式模組的記錄資訊：
 - 一般

- 載入器
- 解析器
- 印表機
- 解析器
- 遙測
- 轉換器
- 類型對映
- 使用者界面
- 控制器
- 比較綱要
- 複製資料中心
- 應用分析儀

在命令模組請選擇下列其中一個：

- 追蹤 — 最詳細的資訊。
- 偵錯 — 有關通過系統之流程的詳細資訊。
- 資訊 — 執行階段事件，例如啟動或關閉。
- 警告 — 使用已過時的 API，API 的使用不良，其他不希望或意外的運行時情況。
- Error — 執行階段錯誤或未預期的情況。
- 嚴重 — 導致應用程式關閉的錯誤。
- 強制性 — 可能出現的最高錯誤層級。

根據預設，在您開啟偵錯模式之後，會設 AWS SCT 定所有應用程式模組的資訊記錄層級。

例如，若要在轉換期間協助處理關鍵問題區域，請將剖析器、類型對映和使用者介面設定為 Trace。

如果資訊對於記錄檔進行串流處理的檔案系統而言變得過於冗長，請變更至具有足夠空間來擷取記錄檔的位置。

若要將記錄檔傳送至 SupAWS port 部門，請移至儲存記錄檔的目錄，並將所有檔案壓縮為可管理的單一 .zip 檔案。然後上傳包含支援案例的 .zip 檔案。當初始分析完成並繼續進行中的開發時，請將調試模式返回 false 以消除詳細的日誌記錄。然後提高轉換速度。

 Tip

若要管理記錄檔大小並簡化報告問題，請在成功轉換後移除記錄檔或將記錄移至其他位置。執行此工作可確保只有相關的錯誤和資訊會傳輸至 SupAWS port 部門，並防止記錄檔系統填滿。

故障診斷問題AWS SCT

您可在以下找到使用 AWS Schema Conversion Tool (AWS SCT) 來疑難排解問題的相關資訊。

無法從 Oracle 來源資料庫載入物件

當您嘗試從 Oracle 資料庫載入結構描述，您可能會遇到以下其中一個錯誤。

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

發生這些錯誤是因為您用來連接到 Oracle 資料庫的使用者 ID 沒有 AWS SCT 所需結構描述的足夠讀取許可。

您可以透過授予使用者 `select_catalog_role` 許可以及資料庫中任何字典的許可，來解決此問題。這些許可提供 AWS SCT 所需檢視和系統資料表的唯讀存取權。以下範例會建立名為 `min_privs` 的使用者 ID，並授與擁有此 ID 的使用者從 Oracle 來源資料庫轉換結構描述所需的最低許可。

```
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;
```

評估報告警訊

若要評估轉換為其他資料庫引擎的複雜性，AWS SCT 需要存取來源資料庫中的物件。如果在掃描期間 AWS SCT 遇到問題且無法執行評估，則會發出警告訊息。此訊息表示整體轉換百分比已減少。掃描期間 AWS SCT 可能會遇到問題的原因如下：

- 您的數據庫用戶無法存取所有需要的物件。如 AWS SCT 需有關資料庫所需安全性權限和權限的詳細資訊，請參閱 [AWS SCT 的來源](#) 本指南中的適當來源資料庫一節。
- 結構描述中引用的對象不再存在於數據庫中。若要協助解決問題，您可以使用 `SYSDBA` 權限連線，並檢查物件是否存在於資料庫中。
- SCT 正嘗試評估已加密的物件。

AWS SCT CLI 參考資料

本節說明如何開始使用命 AWS SCT 令列介面 (CLI)。此外，本節還提供有關按鍵指令和使用模式的資訊。如需 AWS SCT CLI 指令的完整參考資料，請參閱[參考材料](#)。

主題

- [使用 AWS SCT 指令行介面的先決條件](#)
- [AWS SCT CLI 互動式模式](#)
- [取得 AWS SCT CLI 案例](#)
- [編輯 AWS SCT CLI 案例](#)
- [AWS SCT CLI 指令碼模式](#)
- [AWS SCT CLI 參考資料](#)

使用 AWS SCT 指令行介面的先決條件

下載並安裝最新版本的 Amazon Corretto 11. 如需詳細資訊，請參閱[Amazon Corretto 11 使用者指南中的下載](#)。

下載並安裝最新版本的 AWS SCT. 如需詳細資訊，請參閱[安裝中 AWS SCT](#)。

AWS SCT CLI 互動式模式

您可以在交互模式下使用 AWS SCT 命令行界面。在此模式下，您可以一個接一個地將命令輸入控制台。您可以使用此互動模式進一步了解 CLI 命令或下載最常用的 CLI 案例。

若要在中轉換來源資料庫結構描述 AWS SCT，請執行序列作業：建立新專案、連線至來源和目標資料庫、建立對應規則，以及轉換資料庫物件。由於此工作流程可能很複雜，因此建議您在 AWS SCT CLI 模式下使用指令碼。如需詳細資訊，請參閱[指令碼模式](#)。

您可以從 AWS SCT 安裝路徑的app資料夾執行 AWS SCT CLI 指令。在視窗中，預設安裝路徑為C:\Program Files\AWS Schema Conversion Tool\。請確定此資料夾包含AWSSchemaConversionToolBatch.jar檔案。

若要進入 AWS SCT CLI 互動模式，請在完成先決條件後使用下列命令。

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

現在您可以執行 AWS SCT CLI 命令。確保您以新行結束命令。/此外，請確定在指令參數值之前和之後使用直單引號 (')。

Note

如果前面的命令返回 Unexpected error，請嘗試以下操作：

```
java -Djdk.jar.maxSignatureFileSize=20000000 -jar  
AWSSchemaConversionToolBatch.jar
```

若要查看 AWS SCT CLI 互動模式下可用命令的清單，請執行下列命令。

```
help  
/
```

若要檢視 AWS SCT CLI 命令的相關資訊，請使用下列命令。

```
help -command: 'command_name'  
/
```

在前面的例子中，用#####名稱。

若要檢視 AWS SCT CLI 命令參數的相關資訊，請使用下列命令。

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```

在前面的例子中，用#####名稱。然後，以逗號分隔的參數名稱清單取代##s_list。

若要在 AWS SCT CLI 互動模式下從檔案執行指令碼，請使用下列命令。

```
ExecuteFile -file: 'file_path'  
/
```

在上述範例中，以指令碼的檔案路徑取代 *file_path*。請確定您的檔案具有 .scts 副檔名。

若要結束 AWS SCT CLI 互動模式，請執行quit指令。

範例

下列範例會顯示有關Convert指令的資訊。

```
help -command: 'Convert'  
/
```

下列範例會顯示有關Convert命令兩個參數的資訊。

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

取得 AWS SCT CLI 案例

要獲得最常用的 AWS SCT 場景，您可以使用GetCliScenario命令。您可以在交互模式下運行此命令，然後編輯下載的模板。在腳本模式下使用編輯過的檔案。

指GetCliScenario令會將選取的樣板或所有可用的樣板儲存到指定的目錄中。該模板包含用於運行腳本的完整命令集。請務必編輯這些範本中的檔案路徑、資料庫認證、物件名稱和其他資料。此外，請確定您移除不使用的命令，並在需要時將新命令新增至指令碼。

若要執行命GetCliScenario令，請完成先決條件並進入 AWS SCT CLI 互動模式。如需詳細資訊，請參閱 [互動式模式](#)。

接下來，使用下列語法來執行GetCliScenario命令並取得 AWS SCT 案例。

```
GetCliScenario -type: 'template_type' -directory: 'file_path'  
/
```

在上述範例中，使用下表中的其中一個範本類### *template_type*。接下來，將 *file_path* 替換為要下載腳本的文件夾的路徑。確保 AWS SCT 可以在不要求管理員權限的情況下訪問此文件夾。此外，請確定在指令參數值之前和之後使用直單引號 (')。

若要下載所有 AWS SCT CLI 範本，請不使用-type選項執行上述命令。

下表包含您可以下載的 AWS SCT CLI 範本類型。對於每個範本，表格包含檔案名稱和您可以使用指令碼執行的作業說明。

模板類型	檔案名稱	描述
BTEQ ScriptConversion	BTEQScriptConversionTemplate.scts	將基本太資料查詢 (BTEQ)、FastExport和 MultiLoad 指令碼轉換為 Amazon Redshift RSQL。FastLoad如需詳細資訊，請參閱 轉換 ETL 程序 。
ConversionApply	ConversionTemplate.scts	轉換來源資料庫結構描述，並將轉換的程式碼套用至目標資料庫。您也可以選擇將轉換後的程式碼儲存為 SQL 指令集，並儲存評估報告。如需詳細資訊，請參閱 轉換資料庫結構 。
GenericAppConversion	GenericApplicationConversionTemplate.scts	使用一般應用程式轉換器，轉換內嵌至應用 AWS SCT 程式的 SQL 程式碼。如需詳細資訊，請參閱 轉換應用程式中的 SQL 程式碼 。
HadoopMigration	HadoopMigrationTemplate.scts	將您的現場部署 Hadoop 叢集遷移至 Amazon EMR。如需詳細資訊，請參閱 使用阿帕奇 Hadoop 作為源 AWS SCT 。
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	繼續將現場部署 Hadoop 叢集中斷移轉至 Amazon EMR。如需詳細資訊，請參閱 使用阿帕奇 Hadoop 作為源 AWS SCT 。
信息	InformaticaConversionTemplate.scts	轉換內嵌到資訊擷取、轉換和載入 (ETL) 指令碼中的 SQL 程式碼。在 ETL 指令碼中設定與來源和目標資料庫的連線，並在轉換後儲存轉換的指令碼。

模板類型	檔案名稱	描述
		如需詳細資訊，請參閱 轉換信息 ETL 腳本 。
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	使用應用程式轉換器，轉換內嵌在 C#、C++、Java 和 Pro*C 應用程式中的 SQL AWS SCT 程式碼。如需詳細資訊，請參閱 轉換應用程式 SQL 。
OozieConversion	OozieConversionTemplate.scts	將您的工作流程轉換為 AWS Step Functions。如需詳細資訊，請參閱 使用阿帕奇奧西作為來源 AWS SCT 。
RedshiftAgent	DWHDataMigrationTemplate.scts	轉換來源資料倉儲結構描述，並將轉換後的程式碼套用至目標 Amazon Redshift 資料庫。然後註冊資料擷取代理程式，建立並啟動資料移轉任務。如需詳細資訊，請參閱 從資料倉儲遷移到 Amazon Redshift 。
ReportCreation	ReportCreationTemplate.scts	為數個來源資料庫結構描述建立資料庫移轉報表。然後將此報告儲存為 PDF 檔案的 CSV 檔案。如需詳細資訊，請參閱 遷移評定報告 。
SQL ScriptConversion	SQLScriptConversionTemplate.scts	將 SQL*Plus 或 TSQL 命令檔轉換為 PL/SQL，並儲存轉換後的命令檔。此外，還可以儲存評估報告。

下載 AWS SCT CLI 範本後，請使用文字編輯器設定要在來源和目標資料庫上執行的指令碼。接下來，使用 AWS SCT CLI 指令碼模式執行指令碼。如需詳細資訊，請參閱 [AWS SCT CLI 指令碼模式](#)。

範例

下列範例會將所有範本下載到資C:\SCT\Templates料夾中。

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

下列範例會將ConversionApply作業的範本下載到C:\SCT\Templates資料夾中。

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

編輯 AWS SCT CLI 案例

下載案例範本之後，請將它們設定為取得可在資料庫上執行的工作指令碼。

對於所有範本，請確定您已為來源和目標資料庫提供驅動程式的路徑。如需詳細資訊，請參閱 [下載所需資料庫驅動程式](#)。

請確定包含來源和目標資料庫的資料庫認證。此外，請務必設定對映規則來描述轉換專案的來源-目標配對。如需詳細資訊，請參閱 [建立對應規則](#)。

接下來，設定要執行的作業範圍。您可以移除不使用的命令，或將新命令新增至指令碼。

例如，假設您計劃將來源 Oracle 資料庫中的所有結構描述轉換為 PostgreSQL。然後，您計劃將資料庫移轉評估報告儲存為 PDF，並將轉換後的程式碼套用至目標資料庫。在此情況下，您可以使用範本進行ConversionApply作業。請使用下列程序來編輯 AWS SCT CLI 範本。

若要編輯**ConversionApply**作業的 AWS SCT CLI 範本

1. 打開ConversionTemplate.scts您下載的。如需詳細資訊，請參閱 [範例](#)。
2. 從腳本中刪除 CreateFilter，轉換ApplyToTarget 過濾器，篩選，SaveTargetSQLlbyStatement，SaveTargetSQ 和 SaveReportCSV 操作。
3. 對於SetGlobalSettings作業中的 Oracle_driver 檔案，請輸入 Oracle 驅動程式的路徑。然後，對於 Postgreql_file，請輸入您的 PostgreSQL 驅動程序的路徑。

如果您使用其他資料庫引擎，請為設定使用適當的名稱。如需可在作業中設定之全域設定的完整清單，請參閱中的全域設定矩陣[參考材料](#)。

4. (可選) 在中 `CreateProject`，輸入專案名稱和本端專案檔的位置。如果您選擇繼續使用預設值，請確定 AWS SCT 可以在不要求管理員權限的情況下在 `C:\temp` 資料夾中建立檔案。
5. 在中 `AddSource`，輸入來源資料庫伺服器的 IP 位址。此外，請輸入使用者名稱、密碼和連接埠，以連接到來源資料庫伺服器。
6. 在中 `AddTarget`，輸入目標資料庫伺服器的 IP 位址。此外，請輸入要連線到目標資料庫伺服器的使用者名稱、密碼和連接埠。
7. (選擇性) 在中 `AddServerMapping`，輸入要新增至對應規則的來源和目標資料庫物件。您可以使用 `sourceTreePath` 和 `targetTreePath` 參數來指定資料庫物件的路徑。或者，您可以使用 `sourceNamePath` 和 `targetNamePath` 來指定資料庫物件的名稱。如需詳細資訊，請參閱中的伺服器對應命令 [參考材料](#)。

`AddServerMapping` 作業的預設值會將所有來源結構描述與您的目標資料庫對映。

8. 保存文件，然後使用腳本模式運行它。如需詳細資訊，請參閱 [指令碼模式](#)。

AWS SCT CLI 指令碼模式

建立 AWS SCT CLI 指令碼或編輯範本之後，您可以使用 `RunSCTBatch` 命令來執行它。請確定您以 CLI 指令碼儲存您的檔案做為 `.scts` 副檔名。

您可以從 AWS SCT 安裝路徑的 `app` 資料夾執行 AWS SCT CLI 指令碼。在視窗中，預設安裝路徑為 `C:\Program Files\AWS Schema Conversion Tool\`。請確定此資料夾包含 `RunSCTBatch.cmd` 或 `RunSCTBatch.sh` 檔案。此外，此文件夾應包含該 `AWSSchemaConversionToolBatch.jar` 文件。

或者，您可以在作業系統的 `PATH` 環境變數中新增 `RunSCTBatch` 檔案的路徑。更新 `PATH` 環境變數之後，您可以從任何資料夾執行 AWS SCT CLI 指令碼。

若要執行 AWS SCT CLI 指令碼，請在視窗中使用下列命令。

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

在上述範例中，以指令碼的檔案路徑取代 `file_path`。

若要執行 AWS SCT CLI 指令碼，請在 Linux 中使用下列命令。

```
RunSCTBatch.sh --pathtoscts "file_path"
```

在上述範例中，以指令碼的檔案路徑取代 `file_path`。

您可以在此命令中提供選擇性參數，例如資料庫認證、主控台輸出中的詳細資料層級等。如需詳細資訊，請在下載 AWS SCT 指令行介面參考[參考材料](#)。

範例

下列範例會在C:\SCT\Templates資料夾中執行ConversionTemplate.scts指令碼。您可以在視窗中使用這個範例。

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

下列範例會在/home/user/SCT/Templates目錄中執行ConversionTemplate.scts指令碼。您可以在 Linux 中使用這個範例。

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT CLI 參考資料

您可以在以下指南中找到有關 AWS Schema Conversion Tool 命令行界面 (CLI) 的參考資料：[AWS Schema Conversion Tool CLI 參考](#)。

的版本說明 AWS SCT

本節包含的發行說明 AWS SCT，從版本 1.0.640 開始。

AWS SCT 組建 675 的發行公告

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
Cassandra	DynamoDB	修復了卡桑德拉在目標數據中心安裝失敗的錯誤。	否
DB2 LUW	PostgreSQL	動態 SQL：準備語句：解析和轉換沒有動態 SQL。	否
DB2 LUW	PostgreSQL	增加了對特殊寄存器的支持。	否
DB2 LUW	PostgreSQL	擴充套件更新	否
Hadoop	Amazon EMR	增加了對通過 rsa-sha2 協議連接到 Hadoop 集群的支持。	否
Microsoft SQL Server	Amazon Redshift	修正強制 TLS 的 JDBC 驅動程式，儘管未設定。	否
Netezza	Amazon Redshift	新增對具體化視觀表轉換的支援。	否
Oracle	Amazon Redshift	增加了對 Amazon Redshift 遞歸查詢的支持。	是
Oracle	Aurora	修正了 NUMBER 數據類型轉換不正確的問題。	是
Oracle	Amazon Redshift	資料移轉。甲骨文自動分區。已新增資料表片段值的到期時間。到期時間為 72 小時。發生到期時，會在建立資料移轉任務時重建資料缺陷。	否

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
Oracle	Amazon Redshift	SCT 數據提取器：更改了將數據上傳到 Amazon Redshift 的方法。默認情況下，提取器不會創建分段表。相反地，在所有資料檔案都位於 Amazon S3 儲存貯體之後，擷取程式會使用單一 COPY 命令將它們複製到目標資料表。	否
Oracle	Amazon Redshift	已新增原始資料類型移轉至 VARBYTE 資料行。	否
Oracle	Aurora	多版本轉換	否
Oracle	PostgreSQL	已新增對合併陳述式 PostgreSQL 支援。	是
Oracle	PostgreSQL	增加了對 PostgreSQL 15.x 中新的正則表達式函數的支持。	是
Oracle	Aurora	在衝突時，DO UPDATE 陳述式會轉換而不排除別名。	是
Teradata	Amazon Redshift	增加了對前導/LAG 功能的轉換支持。	否
Teradata	Amazon Redshift	以明確指示資料格式的增強型資料類型轉換。	否
Teradata	Amazon Redshift	改善時間/時間戳記運算式中 AT「TIME ZONE」子句的轉換。	否
Teradata	Amazon Redshift	AI-9996 在使用合併陳述式的轉換程序中。	否

AWS SCT 組建 674 的發行公告

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
全部	全部	各種錯誤修復和性能改進	部分 (僅適用於支援的來源和目標配對)
Azure SQL/ Microsoft SQL 伺服器	Amazon Redshift	移除在結構描述評估/轉換期間誤導使用者的「AI 18066：無法轉換結構描述名稱」訊息	否
Azure SQL/ Microsoft SQL 伺服器	Amazon RDS for MySQL 對於 MySQL 的/ Amazon Aurora MySQL	程序轉換不正確，但未指派傳回碼	部分 (結構描述轉換目前不支援 Azure SQL 做為來源)
Azure SQL/ Microsoft SQL 伺服器	亞 Amazon Aurora RDS	修正了 AI9997 對於 XML 路徑子句轉換的某些情況	部分 (結構描述轉換目前不支援 Azure SQL 做為來源)
Azure SQL/ Microsoft SQL 伺服器	亞 Amazon Aurora RDS	在程序/函數體中將值四捨五入到原始比例	部分 (結構描述轉換目前不支援 Azure SQL 做為來源)

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
Azure SQL/ Microsoft SQL 伺服器	亞 Amazon Aurora RDS	對 EXECUTE 語句轉換的各種改進	部分 (結構描述轉換目前不支援 Azure SQL 做為來源)
Azure SQL/ Microsoft SQL 伺服器/天藍突觸	Amazon Redshift	改進了以下語句和模式的轉換： <ul style="list-style-type: none"> • 例外區塊 • 自動提交 • NONATOMIC • 群組集 • CUBE • ROLLUP 	否
DB2 LUW	亞 Amazon Aurora RDS	元數據加載 SQL 查詢中的各種修復	否
DB2 LUW	亞 Amazon Aurora RDS	觸發器預期不會出現 AI 9996	否
DB2 z/ OS	亞 Amazon Aurora RDS	行號分析函數	否
DB2 z/ OS	亞 Amazon Aurora RDS	十六進制字符串常量	否

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
DB2 z/OS	亞 Amazon Aurora RDS	元數據加載 SQL 查詢中的各種修復。	否
DB2 z/OS	亞 Amazon Aurora RDS	序列參考支援的下一個值	否
DB2 z/OS	亞 Amazon Aurora RDS	取得診斷陳述式 DB2_數字_列選項支援	否
DB2 z/OS	亞 Amazon Aurora RDS	GET 診斷多個語句	否
DB2 z/OS	亞 Amazon Aurora RDS	修正了 FOR 語句轉換中的錯誤。	否
Oracle	Amazon RDS for MySQL 對於 MySQL 的/ Amazon Aurora MySQL	修正了當包函數的參數節點沒有定義的 bug。	是

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
Oracle	亞 Amazon Aurora RDS	修正了擴展包的功能 AW_ORACLE_EXT. 下一天	是
Oracle	亞 Amazon Aurora RDS	修復了甲骨文外部連接中轉換「(+)」的各種錯誤	是
Oracle		Support 驗證	否
SAP ASE	亞 Amazon Aurora RDS	在 UPDATE 語句中轉換 FROM 子句中的多個標識符時修復了錯誤	否
SAP ASE	亞 Amazon Aurora RDS	修正了多行註釋和語句轉換	否
SAP ASE		增加了對連接時加密參數的支持	否
Teradata	Amazon Redshift	改進了具有指定模式名稱的 VOLATION 表的轉換	否
Teradata	Amazon Redshift	複雜 CTE 中的轉換 WHERE 子句不正確	否
Teradata	Amazon Redshift	已新增使用 SCT 資料擷取代理程式移轉資料時對 INTERVAL 資料類型的支援。	否

來源	目標	最新消息、增強功能或修正錯誤	AWS DMS 架構轉換中的可用性
泰瑞數據腳本	Amazon Redshift RSQL 腳本	BTEQ 執行的程序中的參數轉換不正確	否

AWS SCT 組建 673 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	一般錯誤修復和性能改進
Azure SQL/ Microsoft SQL 伺服器	Aurora 郵政/Amazon RDS	修正函數調用轉換不正確
Azure SQL/ Microsoft SQL 伺服器	Aurora 郵政/Amazon RDS	實現了 FOR XML 子句的轉換
Azure SQL/ Microsoft SQL 伺服器	Aurora 郵政/Amazon RDS	FORXML 子句的轉換與錯誤的別名。
Azure SQL/ Microsoft	Aurora 郵政/Amazon RDS	修復了 AWS SCT 不轉換使用過程參數運行字符串的 EXECUTE 語句時的錯誤。

來源	目標	最新消息、增強功能或修正錯誤
SQL 伺服器		
Azure SQL/ Microsoft SQL 伺服器	Aurora 郵政/Amazon RDS	改進了具有內部聯接的 UPDATE 語句的轉換。
Azure Synapse	Amazon Redshift	修正了OBJECT_ID 內置函數的不正確轉換。
適用於 z/ OS 的 IBM DB2	Aurora 郵政/Amazon RDS	實作下列陳述式和物件的轉換： <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • PK和分區資料表的UNIQUE條件約束 • TIMESTAMPDIFF 函數 • TO_DATE 函數 • EBCDIC_STR 函數 • VARCHAR_FORMAT 函數
適用於 z/ OS 的 IBM DB2	Aurora 郵政/Amazon RDS	修正了當基於函數的索引轉換後跳過功能的 bug。
適用於 z/ OS 的 IBM DB2	Aurora 郵政/Amazon RDS	修正了轉換後使用 AI 9996 關閉REPEAT語句的錯誤
適用於 z/ OS 的 IBM DB2	Aurora 郵政/Amazon RDS	修正了FINAL TABLE條款以 9996 關閉的錯誤。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/OS 的 IBM DB2	Aurora 郵政/Amazon RDS	LOADER 參考條件約束中的分割索引鍵。AWS SCT 現在可以將分區資料表中的主索引鍵和唯一條件約束轉換為次要索引。
適用於 z/OS 的 IBM DB2	Aurora 郵政/Amazon RDS	支持變量格式函數
適用於 z/OS 的 IBM DB2	Aurora 郵政/Amazon RDS	在CreateTransformationRule 和 ModifyTransformationRule SCT CLI 命令中實作定序變更。
Greenplum	Amazon Redshift	修復了轉換後存儲過程調用不正確的錯誤
Hadoop	Amazon EMR	已新增使用 rsa-sha2 通訊協定連線至 Hadoop 叢集的支援。
Hadoop	Amazon EMR	增加了對 Amazon EMR 的支持與非 Glue 蜂巢元存儲，
Oracle	Amazon Redshift	修正了遞歸查詢不正確轉換的錯誤，其中 PRIOR 列不在 SELECT 列表中。
Oracle	Aurora 郵政/Amazon RDS	實現返回關聯數組的元素
Oracle	Aurora 郵政/Amazon RDS	修復了UNPIVOT帶括號中的意外 AI 9996
Oracle	Aurora 郵政/Amazon RDS	修正了人工智能 9996 中的UNPIVOT意外問題 UNION ALL

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora 郵政/Amazon RDS	改進了Number 數據類型轉換
Oracle	Amazon Redshift 數據提取器	Support Oracle 資料表的自動分割。建立移轉任務的最佳化。
Teradata	Amazon Redshift	實現EXCEPTION BLOCK語句的轉換
Teradata	Amazon Redshift	Support 將ALLANY、和SOME述詞轉換為 Amazon Redshift。
Teradata	Amazon Redshift	已新增對QUALIFY謂詞的原生支援。
Teradata	Amazon Redshift	改進了以下內容的轉換： <ul style="list-style-type: none"> 遞迴查詢 GROUPING SET CUBE ROLLUP UPDATE帶有隱式連接的語句
OLAP 來源	Amazon Redshift 數據提取器	針對 Amazon Redshift 資料擷取器任務實作停止/繼續的 CLI 命令。
OLAP 來源	Amazon Redshift 數據提取器	新增了選取在設定移轉任務期間需要移轉的表格欄的功能。

組建 672 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	Amazon RDS for PostgreSQL	已實作 PostgreSQL 主要版本 15 做為移轉目標的支援。
全部	Amazon Redshift	在PrintTaskStatus 命 AWS SCT 命令行界面 (CLI) 中添加了一個新命令，以顯示數據遷移任務的狀態。
全部	Amazon Redshift	改進了數據提取代理程序的配置流程。
全部	Amazon Redshift	已解決資料擷取代理程式未顯示子工作相關資訊的錯誤。
Apache Oozie	AWS Step Functions	添加了一個選項，將狀態機定義保存為轉換後的代碼中的腳本。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	實作COALESCE、DATEADD、GETDATE和SUM函數的轉換。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改進了帶有JOIN和OUTPUT子UPDATE句的語句的轉換。
天青 SQL 數據庫	Aurora PostgreSQL	已解決轉換SELECT TOP 1 WITH TIES陳述式期間發生的錯誤。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	PostgreSQL	
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	已解決在內建函數中轉換FOR XML子句期間發生的多個問題。
Greenplum	Amazon Redshift	透過使用原生 Amazon Redshift EXCEPTION 區塊來實作GET DIAGNOSTICS 和RAISE陳述式的轉換。
Greenplum	Amazon Redshift	透過在轉換的程式碼中新增EXCEPTION 區塊的支援，改善預存程序的轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	修正了帶有時間格式範本的TO_CHAR函數轉換不正確的錯誤。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	實作巢狀資料表運算式的轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	實作GOTO、MERGE和SIGNAL陳述式的轉換。REPEAT

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	使用BEFORE和AFTER方向關鍵字實現了FETCH語句的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作FINAL TABLE和OLD TABLE表格參照的轉換。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實現了以下功能的轉換。 <ul style="list-style-type: none"> • ADD_MONTHS • DAY與字符數據類型的參數 • DAYOFWEEK • DAYS • DECODE • HOUR • LAST_DAY • LOCATE_IN_STRING • MICROSECOND • MINUTE • MONTH • ROUND • TIME • TIMESTAMP • TIMESTAMP_FORMAT • TRANSLATE • UNICODE_STR • XMLCAST • XMLELEMENT • XMLQUERY • XMLSERIALIZE • YEAR
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進子JOIN句中子查詢別名的轉換。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了COALESCE功能的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了EXPLICIT索引的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改善複合運算式中欄名稱的轉換，以解決轉換期間行動項目 9997 意外出現的問題。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了主鍵和唯一約束的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改善XMLTABLE陳述式中INSERT陳述式的轉換，以解決轉換期間行動項目 9996 意外出現的問題。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	已解決行動項目 9996 在轉換具有引數的函SUBSTR數期間意外出現的問題。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	已解決行動項目 9996 在轉換CURRENT_TIMESTAMP 特殊暫存器期間意外出現的問題。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	解決行動項目 9996 在轉換陳述式、不支援的MERGE陳述式和不支援的內建函數期間意外出現的問題。
Microsoft SQL Server	全部	增加了對 2022 年 Microsoft SQL 服務器版本的支持作為源。
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	改善使用字串連接運算子之SELECT陳述式的轉換。AWS SCT 在轉換後的代碼中使用該STRING_AGG 函數。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL L	實現了對 Babelfish 功能配置文件的新版本 3.1.0 的支持。此檔案定義了特定 Babelfish 版本所支援且不支援的 SQL 功能。

來源	目標	最新消息、增強功能或修正錯誤
Netezza	Amazon Redshift	已解決資料擷取代理程式未從指定的 CDC 點啟動資料移轉的問題。
Oracle	全部	更新 Oracle 資料庫 19 版的評估報告作為來源。
Oracle	Aurora PostgreSQL L PostgreSQL L	通過向 AWS SCT 擴展包添加新功能來實現 DBMS_OUTPUT 軟件包的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	實作使用關聯陣列做為引數或參數之函數和程序的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	改善 SELECT 陳述式中 DISTINCT 子句的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	已改善主索引鍵條件約束與資料表名稱相同之表格的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL PostgreSQL	使用第三個參數改進了RAISE_APPLICATION_ERROR 程序的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	解決遷移規則未自動將NUMERIC資料類型變更為適用INTEGER的問題。
甲骨文 DW	Amazon Redshift	在轉換後的程式碼中實作原生 Amazon Redshift CONNECT BY 子句的支援。
甲骨文 DW	Amazon Redshift	透過為移轉範圍中的每個資料表或磁碟分割自動新增子工作，改善了資料移轉。這種方法可以防止分區後插入的數據丟失。
Teradata	Amazon Redshift	實現了遞歸視圖的轉換。
Teradata	Amazon Redshift	透過新增原生 Amazon Redshift AUTOCOMMIT 交易模式的支援，改善使用BTET和ANSI交易模式之預存程序的轉換。
Teradata	Amazon Redshift	透過在轉換的程式碼中新增NONATOMIC 關鍵字，改善使用TERADATA交易語意的預存程序轉換。
Teradata	Amazon Redshift RSQL	解決轉換後的程式碼包含 AWS 存取金鑰 ID 和秘密存取金鑰的問題。

AWS SCT 組建 671 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	修復了 AWS SCT 沒有權限在 Windows 中保存項目文件的錯誤。
全部	全部	<p>更新了以下 AWS SCT 命令行界面 (CLI) 模板。</p> <ul style="list-style-type: none"> • BTEQ ScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigration • 信息 <p>如需 AWS SCT CLI 範本的詳細資訊，請參閱取得 CLI 案例。</p>
全部	Amazon Redshift	修復了在命令行界面 (CLI) 中 AWS SCT 未創建擴展包的錯誤。
全部	Amazon Redshift	解決資 AWS SCT 料擷取代理程式未在命令列介面 (CLI) 中使用 AWS Snowball 組態的問題。
Apache Oozie	AWS Step Functions	AWS Step Functions 在命令行界面 (CLI) 模式下實現了對從 Apache Oozie 遷移到的支持。將您的 Hadoop 工作負載遷移到 Amazon EMR 之後，您現在可以將工作流程排程系統遷移到 AWS 雲端如需詳細資訊，請參閱 將阿帕奇奧西轉換為 AWS Step Functions 。
天青 SQL 數據庫	Aurora PostgreSQL	已修正資料表和別名發生的解析程式錯誤。
Microsoft SQL Server	PostgreSQL	
天青 SQL 數據庫	Aurora PostgreSQL	實作子INDEX ON句的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	PostgreSQL	
天青 SQL 數據庫	Aurora PostgreSQL	已改善下列物件的轉換，以避免非預期的動作項目。
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Batch 陳述式 • 表示式清單 • 表格別名 • 暫時資料表 • 觸發 • 使用者變數
天青 SQL 數據庫	Aurora PostgreSQL	已解決程序發生的剖析錯誤。
Microsoft SQL Server	PostgreSQL	
天青 SQL 數據庫	Aurora PostgreSQL	修正了在轉換的OBJECT_ID 函數代碼中 AWS SCT 使用不正確的臨時表名稱的錯誤。
Microsoft SQL Server	PostgreSQL	

來源	目標	最新消息、增強功能或修正錯誤
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	已解決行動項目 9996 在轉換下列程式碼元素期間意外出現的問題。 <ul style="list-style-type: none"> • CONVERT 函式 • DATEADD 函式 • DELETE內聯函數中的語句 • IF 陳述式 • INSERT或欄上的UPDATE動作 • RETURN 陳述式 • UPDATE具有複雜查詢或函數的陳述式
BigQuery	Amazon Redshift	已新增對 BigQuery 做為多伺服器評估程序來源的支援。如需詳細資訊，請參閱 建立多伺服器評估報告 。
Hadoop	Amazon EMR	已更新您用來連線至來源資料庫的支援 Apache Hive JDBC 驅動程式版本。如需詳細資訊，請參閱 下載所需資料庫驅動程式 。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	增強來源中繼資料載入器，以確保 AWS SCT 載入來源資料庫物件，例如主索引鍵、隱含索引等。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	修正隱含游標中資料行發生的解析程式錯誤。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作了在轉換的程式碼中保留 DML 陳述式中資料行名稱、運算式和子句的格式化功能。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實現了跨模式外鍵的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作了LENGTH和VARCHAR函數的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作LABEL ON和DECLARE CONDITION 陳述式的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	使用OPTIMIZE FOR子句實作SELECT陳述式的轉換。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	為所有支援的資料類型新增預設值，改善CREATE TABLE陳述式的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了INCREMENT BY屬性的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	新增從轉換範圍中排除資料表分割的功能，改善分區資料表的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了帶有INCLUDE列的主鍵定義的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了SUBSTRING 功能的轉換。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了 SET 和 DECLARE HANDLER FOR 語句的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了變量數據類型的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了 XMLTABLE 功能的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	透過實作下列順序，將轉換的物件套用至目標資料庫，改善了移轉流程：資料表、分割區、索引、條件約束、外部索引鍵和觸發程序。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	已解決行動項目 9996 在原始程式碼中轉換註解期間意外出現的問題。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	解決行動項目 9997 在 FROM 子句中轉換別名期間意外出現的問題。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	解決在轉換游標別名期間，行動項目 9997 意外出現的問題。
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	修正轉換後的程式碼會針對含有 ORDER BY 子句的 SELECT 陳述式傳回不同結果的錯誤。由於 SQL Server 和 PostgreSQL 會以不同的方式處理 NULL 值，因此轉換後的程式碼現在會包含 NULLS FIRST 或 NULLS LAST 子句，以確保轉換後的程式碼以與原始程式碼相同的順序傳回結果。
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	解決表格函數中的資料類型轉換不正確的問題。
MySQL	Amazon RDS for MySQL	解決轉換後的程式碼中資料庫物件名稱周圍意外出現單引號 (' ') 的問題。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL PostgreSQL	已新增檢視表至擴充套件，以模擬 Oracle 系統檢視表，以顯示分割區和子分割區的相關資訊。
Oracle	Aurora PostgreSQL PostgreSQL	已更新擴充套件中的兩個函式，以在轉換的程式碼中新增結構描述名稱做為引數。
Oracle	Aurora PostgreSQL PostgreSQL	修復了在用戶界面中刷新應用程序代碼後，AWS SCT 未使用正確參數進行 C++ 應用程序轉換的錯誤。
Oracle	Aurora PostgreSQL PostgreSQL	改進了CREATE TYPE語句的轉換以避免意外的異常。
Oracle	Aurora PostgreSQL PostgreSQL	改進了嵌套表格的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL L PostgreSQL L	已解決封裝物件發生的剖析錯誤。
Oracle	Aurora PostgreSQL L PostgreSQL L	解決當名稱長度超過 60 個字元時，在轉換的程式碼中 AWS SCT 意外裁剪物件名稱的問題。
Oracle	Aurora PostgreSQL L PostgreSQL L	已解決不正確轉換分區資料表的資料列層級觸發程序的問題。
甲骨文 DW	Amazon Redshift	實作資料移轉的自動資料表分割的支援。為了加速資料移轉，AWS SCT 可以根據ROWID虛擬資料欄中的值自動分割大型資料表或分割區。如需詳細資訊，請參閱 使用原生分割 。
Teradata	Amazon Redshift	在轉換後的 Amazon Redshift 程式碼中實作原生MERGE命令的支援。如需有關 Amazon Redshift MERGE 命令的詳細資訊，請參閱亞 Amazon Redshift 資料庫開發人員指南中的 合併 。
Teradata	Amazon Redshift	改進了不使用明確表格名稱的DELETE和UPDATE語句的轉換。
Teradata	Amazon Redshift	解決了IN和NOT IN語句轉換不正確的問題。

AWS SCT 組建 670 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	<p>已解決行動項目 9996 在轉換下列程式碼元素期間意外出現的問題。</p> <ul style="list-style-type: none"> • CREATE INDEX 陳述式內的 INCLUDE 陳述 • DECLARE 陳述式 • DECLARE ... TABLE 陳述式 • DECLARE 在 LOOP 語句中使用默認值 • DELETE 陳述式 • DROP CONSTRAINT 陳述式內的 ALTER TABLE 陳述 • EXECUTE AS CALLER 和 REVERT • IIF 陳述式 • 表示式清單 • MONTH() 函式 • UPDATE 陳述式 • YEAR() 函式
突觸分析	Amazon Redshift	<p>已新增 Azure Synapse 分析支援，做為多伺服器評估程序的來源。如需詳細資訊，請參閱 建立多伺服器評估報告。</p>
Hadoop	Amazon EMR	<p>在命令列界面 (CLI) 模式中實作了 Hadoop 叢集遷移至 Amazon EMR 的支援。如需詳細資訊，請參閱 遷移大數據架構。</p>
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	<p>已修正來源資料表和資料行發生的解析程式錯誤。</p>
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL	<p>實現了 CASE 表達式的轉換。</p>

來源	目標	最新消息、增強功能或修正錯誤
	PostgreSQL L	
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實現了CURRENT_DATE 引用到特殊寄存器的轉換。Db2 for z/OS 中特殊暫存器的參考是目前伺服器所提供的值的參考。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作了DATE和POSSTR函數的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了日期時間常量的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改善下列資料類型欄的預設值轉換：DATETIME、TIMESTAMP、和TIMESTAMP WITH TIME ZONE。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	解決行動項目 9996 在轉換SELECT INTO陳述式期間意外出現的問題。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改進了DATEDIFF功能的轉換。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	修正了ISNULL函數轉換為的錯誤NULLIF。因此，與原始程式碼相比，轉換後的程式碼產生不同的結果。現在，AWS SCT 將ISNULL函數轉換為COALESCE。
Netezza	Amazon Redshift	已改善資料擷取代理程式，以解決為成功完成的工作設定失敗狀態的問題。
Netezza	Amazon Redshift	新增使用資料擷取代理程式開始資料移轉後變更子工作中端點的功能。
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	新增使用 IPv6 位址通訊協定連線至資料庫的功能。
Oracle	Amazon RDS for Oracle	實作DBMS_JOB封裝的轉換，可排程和管理工作佇列中的工作。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL	在擴充功能套件中新增功能，以改善全域巢狀表格的轉換。這些新函式會在您的原始程式碼中模擬DELETEEXTEND、和TRIM函式。
	PostgreSQL	
Oracle	Aurora PostgreSQL	新增了為 Java 應用程式中嵌入的 SQL 程式碼指定轉換範圍的功能。您現在可以從轉換範圍中排除來源應用程式專案的子集。如需詳細資訊，請參閱 將您的 Java 應用程式 SQL 程式碼轉換為AWS SCT 。
	PostgreSQL	
Oracle	Aurora PostgreSQL	改善函數索引內連接運算子 () 的轉換。
	PostgreSQL	
Oracle	Aurora PostgreSQL	改善了原始程式碼不包含單一運算式括號的IN條件轉換。
	PostgreSQL	
Oracle	Aurora PostgreSQL	改進了MERGE語句 PostgreSQL。INSERT ON CONFLICT
	PostgreSQL	

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL	已解決程序套件所發生的剖析錯誤。
	PostgreSQL	
Oracle	Aurora PostgreSQL	解決行動項目 5072 在套件轉換期間意外出現的問題。
	PostgreSQL	
甲骨文 DW	Amazon Redshift	修復了將轉換後的代碼應用到目標數據庫時 AWS SCT 未應用擴展包的錯誤。
甲骨文 DW	Amazon Redshift	修復了使用擴展包嚮導時 AWS SCT 未應用某些擴展包文件的錯誤。
甲骨文 DW	Amazon Redshift	解決 AWS SCT 無法處理並行執 parallel 超過 500 個工作 AWS Snowball 的資料移轉至的問題。
甲骨文 DW	Amazon Redshift	解決了使用者定義型別的使用者定義函數轉換不正確的問題。

AWS SCT 組建 669 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	改善多伺服器評估程序，有助於判斷來源資料庫的最佳目標資料庫平台。現在，如果您在輸入逗號分隔值 (CSV) 檔案中提供資料庫認證，請 AWS SCT 忽略該機 AWS Secrets Manager 碼。如需詳細資訊，請參閱 建立多伺服器評估報告 。

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	解決多伺服器評估報告會在使用來源密碼連線 AWS Secrets Manager 至資料庫時，包含來源資料庫的 IP 位址的問題。
全部	Amazon Redshift	根據操作系統和可用 RAM 實現 Java 虛擬機 (JVM) 設置的自動配置。 AWS SCT 使用此 JVM 運行數據提取代理工作。
全部	Amazon Redshift	解決資料擷取代理程式無法在 Ubuntu 中啟動的問題。
全部	Amazon Redshift	解決在 Windows 中執行 StartAgent.bat 檔案後，資料擷取工作無法啟動的問題。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決在開啟「產生索引的唯一名稱」選項時，欄名稱轉換不正確的問題。
Greenplum	Amazon Redshift	實現了返回 VOID 程序的函數轉換。
Greenplum	Amazon Redshift	解決當來源資料庫在數值欄中未包含數值 (NaN) 值時，資料移轉失敗的問題。 AWS SCT 資料擷取代理程式現在會以 NULL 取代 NaN 值。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	添加了一個新的轉換設置來指定 CHAR 內置函數轉換過程中的 DATE FORMAT 和 TIME FORMAT 選項。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	已新增行動項目 8534，用於轉換使用子句宣告的預先定義游標。 WITHOUT RETURN 如果游標未傳回結果集，請在轉換後的程式碼中為游標名稱 AWS SCT 指派 NULL 值，並引發動作項目。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	已編輯連線至來源資料庫 AWS SCT 時識別的CURRENT CLIENT_AP PLNAME 屬性。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作新的轉換設定，以在轉換CHAR內建函數期間指定DATE FORMAT和TIME FORMAT選項。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作BEGIN...END 區塊LEAVE陳述式中陳述式的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	實作XMLPARSE、XMLTABLE和XMLNAMESPACES 函數的轉換。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	改進了CHAR內置功能的轉換。

來源	目標	最新消息、增強功能或修正錯誤
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	改進了游標的轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	解決行動項目 9996 在轉換 FOR 迴圈陳述式期間意外出現的問題。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	已改善 SELECT 陳述式中表格類型用法的轉換。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	實現了對巴貝爾魚功能配置文件的新版本 2.2.0 的支持。此檔案定義了特定 Babelfish 版本所支援且不支援的 SQL 功能。
Netezza	Amazon Redshift	已改善資料擷取代理程式，以解決在進行中的資料複寫期間未從目標資料表中刪除某個資料列的問題。
Oracle	Amazon RDS for Oracle	改進了 Oracle 數據庫企業版功能的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL L PostgreSQL L	實現了GROUPING_ID 函數的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	透過在命令列介面 (CLI) 模式中新增對自訂資料類型對應的支援，改善 C# 應用程式中的 SQL 程式碼轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	已改善巢狀資料表的轉換，以避免出現非預期的動作項目 9996。
Oracle	Aurora PostgreSQL L PostgreSQL L	解決了對象構造函數的調用轉換不正確的問題。
甲骨文 DW	Amazon Redshift	實現了對現有表分區的數據遷移的支持。若要加快資料移轉速度，請為來源資料表的每個分割區 AWS SCT 建立子工作，但不是空的。如需詳細資訊，請參閱 使用原生分割 。
Teradata	Amazon Redshift	使用TIME WITH TIME ZONE AS TIMESTAMP、TIME WITH TIME ZONE AS CHAR和引數改進了CAST函TIMESTAMP AS TIME WITH TIME ZONE數的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	使用該FORMAT選項改進了CAST功能的轉換。
Teradata	Amazon Redshift	解決了CEIL函數未轉換的問題。
Teradata	Amazon Redshift	解決了帶有DELETE子MERGE句的語句轉換不正確的問題。
Teradata	Amazon Redshift	解決了帶有日期和格式引數的TO_CHAR函數轉換錯誤的問題。

AWS SCT 組建 668 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	Amazon Redshift	解決遷移規則中的乘法運算子無法正常運作的問題。這些運算子可以變更char、varcharnvarchar、和string資料類型的長度。如需詳細資訊，請參閱 建立移轉規則 。
突觸分析	Amazon Redshift	使用引數實現了對CONVERT函VARCHAR數的支持。
突觸分析	Amazon Redshift	改進了帶有子SELECT句的NOLOCK語句轉換。
突觸分析	Amazon Redshift	改進了具有別名或 and FROM 子UPDATE句的語句的轉換。SET
Greenplum	Amazon Redshift	實作資料移轉的自動虛擬磁碟分割。AWS SCT 使用系GP_SEGMENT_ID 統欄來建立分割區。
Greenplum	Amazon Redshift	實施的支持RETURN QUERY和RETURN SETOF條款。

來源	目標	最新消息、增強功能或修正錯誤
Greenplum	Amazon Redshift	使用三個參數實現了對SUBSTRING 函數的支持。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	改進了具有LOCATE參數的SUBSTR函數轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	新增使用REFCURSOR 變數陣列傳回動態結果集的選項。當您在轉換設定中選取此選項時，會在轉換的程式碼中 AWS SCT 加入其他OUT參數。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	實現了對FOR循環語句的支持。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	實現了對XMLPARSE功能的支持。為函數中XMLPARSE的空白區域分割新增了一個動作項目 8541。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	改進了單個BEGIN ... END塊中多個異常處理程序的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL	改進了轉換INSERT和DELETE觸發器。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	已改善巢狀程序呼叫的轉換。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	改進了表格類型的轉換。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	解決錯誤地轉換為整數值的位元邏輯NOT運算的問題。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	解決了在 PostgreSQL 版本 8.0.2 及更低版本中未初始化本機陣列的問題。
	PostgreSQL	

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決了帶有WHEN NOT MATCHED BY SOURCE子MERGE句的語句轉換不正確的問題。
MySQL	Aurora MySQL	解決錯 AWS SCT 誤判斷 rds_superuser_role 角色授與使用者權限的問題。
Netezza	Amazon Redshift	增強來源中繼資料載入器，以 AWS SCT 確保以小寫名稱正確載入資料庫物件。
Oracle	Aurora PostgreSQL PostgreSQL	在擴充功能套件中新增功能，以改善本機巢狀表格的轉換。這些新函數會在原始程式碼中模擬PRIORFIRSTLASTEXISTSEXTENDTRIM、DELETE、、、、和SET函數。NEXT LIMIT如需詳細資訊，請參閱 使用延伸套件 。
Oracle	Aurora PostgreSQL PostgreSQL	新增指定 C# 應用程式轉換範圍的功能。使用者現在可以從轉換範圍中排除來源應用程式專案的子集。
Oracle	Aurora PostgreSQL PostgreSQL	實現了集合中COUNT方法的支持。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL L PostgreSQL L	在嵌套表中實現了對變量和構造函數的支持。
Oracle	Aurora PostgreSQL L PostgreSQL L	實作的RATIO_TO_REPORT 和STANDARD_HASH 功能的支援。
Oracle	Aurora PostgreSQL L PostgreSQL L	改進了作為 AWS SCT 擴展包一部分的大型對象 (LOB) 的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	改進了本地集合的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	改進了JOIN語句的USING語句轉換，其中列名稱不包含表格名稱。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL L PostgreSQL L	實現了EMPTY_BLOB 和EMPTY_CLOB 函數的轉換。
Oracle	Aurora PostgreSQL L PostgreSQL L	在 C # 應用程序中實現位置綁定變量的轉換。
SAP ASE	Aurora PostgreSQL L PostgreSQL L	實現了多事件觸發器的轉換。
SAP ASE	Aurora PostgreSQL L PostgreSQL L	實作遞迴觸發程序的轉換。
SAP ASE	Aurora PostgreSQL L PostgreSQL L	改進了使用@@rowcount 全局變量觸發器的轉換。

來源	目標	最新消息、增強功能或修正錯誤
SAP ASE	Aurora PostgreSQL PostgreSQL	解決陳述式SET子UPDATE句中的彙總函式轉換不正確的問題。
SAP ASE	Aurora PostgreSQL PostgreSQL	已解決行動項目 42702 在轉換陳述式期間意外出現的UPDATE問題。
SAP ASE	Aurora PostgreSQL PostgreSQL	解決了帶有CHAR引數的CONVERT函數轉換不正確的問題。
Snowflake	Amazon Redshift	新增對 Snowflake 作為資料擷取代理程式進行資料移轉來源的支援。如需詳細資訊，請參閱 將資料從現場部署資料倉儲遷移到 Amazon Redshift 。
Teradata	Amazon Redshift	改進了帶TIMESTAMP AS TIME WITH TIMEZONE參數的CAST函數轉換。

AWS SCT 組建 667 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	在命令列介面 (CLI) 模式中實作資訊擷取、轉換和載入 (ETL) 指令碼的支援。AWS SCT 自動將您的信息 ETL 腳本重定向到新的目標數據

來源	目標	最新消息、增強功能或修正錯誤
		庫。此外，AWS SCT 轉換物件名稱和內嵌在您的資訊物件中的 SQL 程式碼。如需詳細資訊，請參閱 轉換信息 ETL 腳本 。
全部	Amazon Redshift	將 Amazon Redshift 支持的最低驅動程序版本增加到 2.1.0.9。如需詳細資訊，請參閱 下載所需資料庫驅動程式 。
突觸分析	Amazon Redshift	在擴展包中添加了一個新功能，以改善具有三個日期和時間參數的 CONVERT 函數轉換。
突觸分析	Amazon Redshift	改進了 DATEDIFF 功能的轉換。
突觸分析 Microsoft SQL 服務器 DW	Amazon Redshift	更新了擴展包版本。確保您在現有 AWS SCT 項目中應用了最新版本的擴展包。如需詳細資訊，請參閱 使用延伸套件 。
BigQuery	Amazon Redshift	解決未在命令列介面 (CLI) 模式中轉換篩選物件的問題。
Greenplum	Amazon Redshift	修正 AWS SCT 未轉換預存程序中宣告之暫存資料表的錯誤。
Greenplum	Amazon Redshift	修正轉換後的程式碼中缺少欄編碼屬性的錯誤。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	針對具有多個 INNER JOIN 子 UPDATE 句的自我參照資料表實作陳述式轉換。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	實作 SQL Server 用於 DML 觸發程序的deleted暫存資料表的支援inserted和暫存資料表。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改善在不同資料庫結構描述中建立之預存程序中使用使用者定義型別的轉換。解決找 AWS SCT 不到資料類型並顯示行動項目 9996 的問題。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決轉換後的程式碼中資料庫物件名稱周圍意外出現方括號 ([]) 的問題。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決了@@ROWCOUNT 函數轉換不正確的問題。
Microsoft SQL 服務器 DW	Amazon Redshift	實作的geometry和資geography 料類型的支援。
Microsoft SQL 服務器 DW	Amazon Redshift	在轉換後的代碼中實現了對數據類型聲明中MAX關鍵字的支持。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL 服務器 DW	Amazon Redshift	改進了DATEADD功能的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	透過增加對 MyBatis架構的支援，改善 Java 應用程式中的 SQL 程式碼轉換。如需詳細資訊，請參閱 在 Java 應用程式中轉換 SQL 程式碼 。
Oracle	Aurora PostgreSQL PostgreSQL	改進了使用該 MyBatis 框架的 Java 應用程序中的 SQL 代碼轉換。針對具有不支援語法的 SQL 程式碼新增行動項目 30411。
Oracle	Aurora PostgreSQL PostgreSQL	新增宣告支援，改善 Pro*C 應用程式中的 SQL 程式碼轉換。typedef struct
Oracle	Aurora PostgreSQL PostgreSQL	實作的支援CROSS JOIN和LEFT JOIN陳述式。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL	已改善MERGE陳述式的轉換。解決轉換後的程式碼中缺少要插入的值的問題。
	PostgreSQL	
Teradata	Amazon Redshift	已變更轉換後程式碼中 AWS SCT 使用的預設欄壓縮編碼設定，以符合預設 Amazon Redshift 設定。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的 壓縮編碼 。
Teradata	Amazon Redshift	解決使用TIME資料類型的數學運算轉換不正確的問題。
Teradata	Amazon Redshift RSQL	實現了 shell 腳本中 FastExport 代碼的轉換。
泰瑞数据	Amazon Redshift RSQL	修正了 AWS SCT 沒有轉換COALESCE和%data語句的錯誤。
Vertica	Amazon Redshift	改善使用者選取一種最佳化策略時的轉換最佳化建議。

AWS SCT 組建 666 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
天青 SQL 數據庫	Aurora PostgreSQL	已解決JOIN陳述式內部ON子句所發生的剖析錯誤。(
Microsoft SQL Server	PostgreSQL	
	L	

來源	目標	最新消息、增強功能或修正錯誤
突觸分析	Amazon Redshift	在擴展包中添加了三個新功能，以改善具有日期和時間參數CONVERT功能的轉換。
突觸分析	Amazon Redshift	增強來源中繼資料載入器，以確保 AWS SCT 載入系統資料庫結構描述。
突觸分析	Amazon Redshift	修正暫存資料表資料行發生的解析程式錯誤。
突觸分析	Amazon Redshift	實作BINARY和資VARBINARY 料類型轉換為資VARBYTE料類型。
突觸分析	Amazon Redshift	在轉換後的代碼中實現了對TIME數據類型的支持。
突觸分析	Amazon Redshift	已改善COLLATE子句的轉換。解決在使用預設資料庫定序轉換資料行期間，行動項目 31141 意外出現的問題。
BigQuery	Amazon Redshift	實作變更輸入參數之程序的轉換。
Greenplum	Amazon Redshift	解決了 AWS SCT 使用與 Greenplum 6.x 資料庫不兼容的查詢的問題。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL	藉由將例外狀況處理常式從 Db2 (適用於 z/OS) 傳輸至 PostgreSQL，改善EXCEPTION 區段的轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL	已改善OPEN CURSOR陳述式的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	通過使用CASE表達式實現了IIF函數的轉換。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決當陳述式未包含BEGIN...END 區塊時，含有資料表值參數的程序無CREATE PROCEDURE 法正確轉換的問題。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決了SCOPE_IDENTITY 函數轉換不正確的問題。
Oracle	Amazon RDS for Oracle	修正使用 Oracle 10g 作為來源時，SELECT_CATALOG_ROLE 角色發生的載入器錯誤。
Oracle	Amazon RDS for Oracle	改進了加載程序以支持 Oracle 計劃程序作業。
Oracle	Aurora PostgreSQL PostgreSQL	使用USING子句實作JOIN陳述式的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL	改善轉換程式碼的效能，其中原始程式碼在WHERE子句中包含全域變數。
	PostgreSQL	
Oracle	Aurora PostgreSQL	透過增加對 MyBatis架構的支援，改善 Java 應用程式中的 SQL 程式碼轉換。如需詳細資訊，請參閱 在 Java 應用程式中轉換 SQL 程式碼 。
	PostgreSQL	
甲骨文 DW	Amazon Redshift	實作PIVOT和UNPIVOT關係運算子的轉換。
Teradata	Amazon Redshift	修正使用 JSON 物件的原始程式碼未轉換的錯誤。
Teradata	Amazon Redshift	修正拖放使用者建立的表格未正確載入的錯誤。
Teradata	Amazon Redshift	實作了將INSTR函數轉換為原生 Amazon Redshift STRPOS 函數。
Teradata	Amazon Redshift	實作NVP和TRANSLATE 函數的轉換。
Teradata	Amazon Redshift	改進了COALESCE表達式的轉換。
Teradata	Amazon Redshift	已改善DECLARE CONDITION 陳述式的轉換。
Teradata	Amazon Redshift	改進了與SECOND語法元素的EXTRACT函數轉換。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	已改善LOOP陳述式內部SQLSTATE和SQLCODE變數的轉換。
Teradata	Amazon Redshift	改進了唯一索引的轉換。
Teradata	Amazon Redshift	解決行動項目 9996 在轉換CURRENT_TIMESTAMP 陳述式時意外出現，且分數精確度設定為 3 的問題。
Teradata	Amazon Redshift	已解決反斜線在字串常值中不正確轉換的問題。
Teradata	Amazon Redshift	已解決轉換的陳述式在EXEC陳述式中包含不正確欄位名稱的ADD CONSTRAINT 問題。
Teradata	Amazon Redshift	解決轉換後的子查QUALIFY詢包含不正確的子查詢名稱的問題。
Teradata	Amazon Redshift	解決轉換後的檢視未套用的問題。在轉換的代碼中為 NULL 值添加了明確轉換為特定數據類型。
Teradata	Amazon Redshift	解決日期和時間函數轉換錯誤的問題。
Teradata	Amazon Redshift	解決未轉換十六進位字串常值的問題。

AWS SCT 組建 665 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
突觸分析	Amazon Redshift	實現了帶VARCHAR參數的CONCAT函數轉換。

來源	目標	最新消息、增強功能或修正錯誤
突觸分析	Amazon Redshift	改善建立暫存資料表且不包含結構定義名稱之CREATE TABLE陳述式的轉換。AWS SCT 會建立結dbo構描述，將這些暫存資料表儲存在目標資料庫中。
突觸分析	Amazon Redshift	改善您在暫存資料表上執行之DROP TABLE陳述式的轉換。
突觸分析	Amazon Redshift	改進了與BEGIN...END 塊OBJECT_ID 語句的轉換。
突觸分析	Amazon Redshift	解決了 AWS SCT 無法轉換帶有塊註釋的存儲過程的錯誤。
BigQuery	Amazon Redshift	實作將資 BigQuery 料倉儲轉換為 Amazon Redshift。如需詳細資訊，請參閱 使用 BigQuery 做為 AWS SCT 的來源 。
Microsoft SQL Server	Aurora PostgreSQL	改善處理多個事件以及在 SQL Server 中使用inserted和deleted系統資料表之觸發程序的轉換。
Microsoft SQL Server	Aurora PostgreSQL	已修正 SQL Server 中針對inserted和deleted系統資料表所發生的解析程式錯誤。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	實現了對 Babelfish 功能配置文件的新版本 2.1.0 的支持。此檔案定義了特定 Babelfish 版本所支援且不支援的 SQL 功能。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora MySQL MariaDB MySQL	解決 <code>varchar2</code> 資料類型轉換不正確的問題。
Oracle	Aurora MySQL Aurora PostgreSQL MariaDB MySQL PostgreSQL	對於版本 12c 及更高版本的 Oracle 資料庫，AWS SCT 支援下列延伸資料類型： <ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW AWS SCT 將這些資料類型支援的最大資料行長度從 8,000 個增加到 32,767 個位元組。
Oracle	Aurora PostgreSQL PostgreSQL	已解決 Oracle 事件處理封裝所發生的剖析錯誤。
Teradata	Amazon Redshift	在單一陳述式中針對多個 <code>RESET WHEN</code> 子 <code>SELECT</code> 句新增行動項目 13214。
Teradata	Amazon Redshift	已針對位於例外處理區塊之外的 <code>SQLSTATE</code> 變數新增行動項目。
Teradata	Amazon Redshift	實現了 <code>ACTIVITY_COUNT</code> 變量轉換為 <code>ROW_COUNT</code> 。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	實現了內置幾何ST_TRANSFORM 函數的轉換。
Teradata	Amazon Redshift	改進了在沒有WHERE子句的視圖中刪除語句的轉換。
Teradata	Amazon Redshift	改善運算式中CAST運算子的轉換。
Teradata	Amazon Redshift	已改善GROUP BY子句的轉換。
Teradata	Amazon Redshift	改進了轉換INSTR和REGEXP_INSTR 內置功能。
Teradata	Amazon Redshift	解決橫向欄別名參照轉換不正確的問題。
Teradata	Amazon Redshift	已解決子QUALIFY查詢中資料行名稱轉換錯誤的問題。
Teradata	Amazon Redshift	使用 ERRORCODE status 值關鍵字實現了.QUIT命令的轉換。
泰瑞数据	Amazon Redshift RSQL	解決行動項目 9996 在轉換CREATE陳述式期間意外出現的問題。
泰瑞数据	Amazon Redshift RSQL	解決行動項目 9998 在轉換END陳述式期間意外出現的問題。

AWS SCT 組建 664 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	已新增對 Amazon Redshift 無伺服器支援，做為中資料庫遷移專案的來源和目標。AWS SCT 要連接到 Amazon Redshift 無服務器，請確保您使用 Amazon Redshift JDBC 驅動程序版本 2.1.0.9 或更高版本。
全部	全部	改進了轉換設置窗口的用戶界面。AWS SCT 現在僅顯示具有建立對映規則的資料庫轉換配對的設定。如需詳細資訊，請參閱 建立對應規則 。
全部	全部	更新評估報告，移除有關行動項目行和位置的重複資訊。
全部	Amazon Redshift	在資料擷取工作中實作自動記憶體平衡。
全部	Amazon Redshift	解決資料擷取代理程式無法連線至 AWS Snowball 裝置的錯誤。
天青 SQL 數據庫	Aurora MySQL	實作 SUSE Linux 15.3 的支援，做為執行資料擷取代理程式的平台。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL	
IBM Db2 LUW	MariaDB	
Microsoft SQL Server	MySQL	
MySQL	PostgreSQL	
Oracle	PostgreSQL	
	MySQL	

來源	目標	最新消息、增強功能或修正錯誤
PostgreSQL		
SAP ASE		
突觸分析	Amazon Redshift	改進了DATEADD功能的轉換。
適用於 z/OS 的 IBM Db2	Aurora PostgreSQL PostgreSQL	新增在移轉規則中變更資料行定序的功能。
Microsoft 社會服務	AWS Glue AWS Glue Studio	已解決使用者選取來源指令碼時發生的未預期錯誤。
Oracle	Aurora MySQL MariaDB MySQL	將儲存函數的使用轉換實作產生的資料行運算式。AWS SCT 創建觸發器來模擬這種行為，因為 MySQL 不支持將存儲的函數作為生成的列表達式的使用。
Oracle	Aurora PostgreSQL PostgreSQL	實現了從包中的函數轉換作為 AWS SCT 擴展包的一部分。UTL_MATCH

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL PostgreSQL	使用參數實現了REGEXP_LIKE 函NULL數的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	改進了SYS_EXTRACT_UTC 功能的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	透過實作Wcscats、Wcscpys和Wcsncats函數的支援，改善 C++ 應用程式中的 SQL 程式碼轉換。如需詳細資訊，請參閱 在 C++ 應用程序中轉換 SQL 代碼AWS SCT 。
甲骨文 DW Snowflake	Amazon Redshift	已解決轉換的陳述式不包含將值明確轉換為資料行資料類型的問題。這個問題發生在使用其他資料表的查詢結果的陳述式中。
Teradata	Amazon Redshift	新增case insensitive 在移轉規則之間case sensitive 和之間變更資料行定序的功能。如需詳細資訊，請參閱 建立移轉規則 。
Teradata	Amazon Redshift	修正陳述式發生的CREATE TABLE AS解析程式錯誤。
Teradata	Amazon Redshift	修復了帶有COALESCE表達式的內置P_INTERSECT 函數未轉換的錯誤。
Teradata	Amazon Redshift	實作名為OID的資料行轉換，_OID以避免在 Amazon Redshift 中使用保留關鍵字。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	實作函數、程序、檢視和巨集的RENAME陳述式轉換。
Teradata	Amazon Redshift	實現了在 Amazon Redshift STROKE 函SPLIT_PART 數的功能轉換。
Teradata	Amazon Redshift	改進了INSTR和REGEXP_INSTR 系統功能的轉換。
Teradata	Amazon Redshift	改進了TIME數據類型的轉換。
Teradata	Amazon Redshift	透過實作主要SET和次要唯一索引的轉換來改善和MULTISET表格的模擬。
Teradata	Amazon Redshift	已解決CHARACTER 函數發生的剖析錯誤。
泰瑞数据	Amazon Redshift RSQL	已解決使用者從專案中移除 Teradata 基本 Teradata 查詢 (BTEQ) 指令碼時發生的錯誤。 AWS SCT

AWS SCT 組建 663 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	新增了移轉規則中使用乘法運算子變更 charvarcharnvarchar、和string資料類型長度的功能。如需詳細資訊，請參閱 建立移轉規則 。
全部	全部	在多伺服器評估報告中實作了三個新欄的支援，並更新了輸入檔案的格式。請確定您使用的是最新版本的輸入檔案的更新範本 AWS SCT。如需詳細資訊，請參閱 建立資料庫移轉的多伺服器評估報告 。

來源	目標	最新消息、增強功能或修正錯誤
突觸分析	Amazon Redshift	已改善OBJECT_ID 陳述式的轉換。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	已新增對 Aurora PostgreSQL 1.2.0 的巴貝爾魚作為資料庫移轉評估報告的目標平台的支援。如需詳細資訊，請參閱 Amazon Aurora 使用者指南 中依版本分類的 Babelfish 支援的功能 。
Microsoft SQL 服務器 DW	Amazon Redshift	增加了對AT TIME ZONE條款的支持。
Microsoft SQL 服務器 DW	Amazon Redshift	解決BEGIN/END 區塊外的陳述式轉換不正確的問題。
Netezza	Amazon Redshift	改進了TIME數據類型的轉換，並實現了相關內置函數，表達式和常值的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	修正使用 Oracle 10g 作為來源時發生的載入器錯誤。
Oracle	Aurora PostgreSQL PostgreSQL	改進了OFFSET和子FETCH句的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL L PostgreSQL L	已解決具有預設值之OUT參數的程序轉換錯誤的問題。
甲骨文 DW	Amazon Redshift	改進了 Oracle 函數轉換為 Amazon Redshift 用戶定義函數的功能。
Snowflake	Amazon Redshift	已改善WITH子句的轉換。
Teradata	Amazon Redshift	已針對CHAR資料類型不支援的多位元組字元新增行動項目 13209。
Teradata	Amazon Redshift	修正了一個加載器錯誤，表沒有完全加載。
Teradata	Amazon Redshift	修復了變壓器錯誤，其中內置P_INTERSECT 函數在JOIN條件下沒有轉換。
Teradata	Amazon Redshift	修正在名稱中含有特殊字元的資料表上執行SELECT陳述式時，檢視名稱會在錯誤大小寫下轉換的問題。
Teradata	Amazon Redshift	改進了具有PERIOD(DATE) 數據類型中UNTIL_CHANGED 值的INSERT語句的轉換。
Teradata	Amazon Redshift	使用 Amazon Redshift 中的FORMAT功能改進了內置TO_CHAR函數的轉換。
Teradata	Amazon Redshift	改進了內置RANK函數的轉換，以確保轉換後的代碼以與源代碼相同的順序返回 NULL 值。
Teradata	Amazon Redshift	改善唯一條件約束 (例如主要或次要唯一索引) 的轉換。

AWS SCT 組建 662 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	新增在建立多伺服器評估報告時，為每個來源資料庫自動建立 AWS SCT 專案的功能。開啟此選項後，AWS SCT 可以將對應規則新增至這些專案，並儲存轉換統計資料以供離線使用。如需詳細資訊，請參閱 建立資料庫移轉的多伺服器評估報告 。
全部	全部	在建立多伺服器評估報告時，在資料庫和結構描述名稱中實作百分比 (%) 作為萬用字元的支援。
全部	Aurora MySQL Aurora PostgreSQL	將所有 AWS Lambda 函數的運行時更新為 Python 3.9 版本。
全部	Amazon Redshift	升級所有要使用的資料擷取代理程式 AWS SDK for Java 2.x。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	改進了帶有子DELETE句的NON EXISTS語句轉換。
突觸分析	Amazon Redshift	解決與來源資料庫的連線失敗的錯誤。
適用於 z/ OS 的 IBM Db2	Aurora PostgreSQL L PostgreSQL L	已解決觸發程序的轉換程式碼包含兩個物件別名提及的錯誤。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改善開啟「將資料庫物件名稱視為區分大小寫」選項時，具有混合大小寫名稱之物件的轉換功能。
Microsoft SQL 服務器 DW Teradata	Amazon Redshift	實作PIVOT和UNPIVOT關係運算子的轉換。
Netezza	Amazon Redshift	實現了TIME數據類型的轉換。
Oracle	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	實現了UTL_TCP.CRLF 包常量轉換。
Oracle	Aurora PostgreSQL PostgreSQL	修正延伸套件問題：轉換期間無法維持可變長度欄的資料類型長度。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Aurora PostgreSQL PostgreSQL	在 C ++ 應用程序中實現了 SQL 代碼轉換。如需詳細資訊，請參閱 在 C ++ 應用程序中轉換 SQL 代碼AWS SCT 。
Oracle	Aurora PostgreSQL PostgreSQL	對全局變量和關聯數組的轉換實現了區分大小寫命名的支持。
Oracle	Aurora PostgreSQL PostgreSQL	改善延伸套件中TO_CHAR TO_DATE、和TO_NUMBER 功能的轉換。
Oracle	Aurora PostgreSQL PostgreSQL	改進了TABLE()運算符的轉換。
甲骨文 DW	Amazon Redshift	增加了對轉換主鍵和其他約束的支持。
甲骨文 DW	Amazon Redshift	修正行動項目 12054 在轉換條件陳述式期間不會出現的問題。

來源	目標	最新消息、增強功能或修正錯誤
SAP ASE	Aurora PostgreSQL PostgreSQL	已解決在轉換具有使用者定義類型之資料行的表格期間，在目標樹狀結構中建立空名稱的物件時的錯誤。
SAP ASE	Aurora PostgreSQL PostgreSQL	修正儲存物件 (例如指令碼、常式等) 的載入程式錯誤。
Snowflake	Amazon Redshift	修正行動項目 22152 在需要時不會顯示，並將轉換結果 AWS SCT 顯示為註解的問題。
Snowflake	Amazon Redshift	改進了日期和時間函數的轉換；實現了對時區的支持。
Snowflake	Amazon Redshift	已解決具有WITH子句的非遞迴通用資料表運算式 (CTE) 轉換為遞迴 CTE 的問題。
Teradata	Amazon Redshift	改善具有條件表格連結之UPDATE陳述式的轉換。
Teradata	Amazon Redshift	已改善RENAME TABLE陳述式的轉換。
Teradata	Amazon Redshift	解決了空白欄與評估報告一起出現在逗號分隔值 (CSV) 檔案中的問題。
Teradata	Amazon Redshift RSQL	修正轉換的基本 Teradata 查詢 (BTEQ) 巨集結尾缺少分號的錯誤。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift RSQL	已改善CASE陳述式中多個資料類型值的轉換。
Teradata	Amazon Redshift RSQL	改進了具有ESCAPE字符的LIKE ANY子句的轉換。
Teradata	Amazon Redshift RSQL	已改善INSERT陳述式中CAST函數的轉換。
Teradata	Amazon Redshift RSQL	改進了時區的轉換，實現了時區區域映射。
Teradata	Amazon Redshift RSQL	已解決在使用 BTEQ 指令碼轉換殼層指令碼期間，行動項目 9998 意外出現的問題。
Teradata	Amazon Redshift RSQL AWS Glue	對替代變量的值實現了 500 個字符的限制。
Vertica	Amazon Redshift	實作BINARY、VARBINARY LONG BINARYBYTEA、和RAW資料類型轉換為VARBYTE料類型。
Vertica	Amazon Redshift	改進了內置函數和文字的轉換。

AWS SCT 組建 661 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	已新增篩選器，以在對應檢視中搜尋對應規則。套用篩選時，AWS SCT 會顯示符合 [伺服器對應] 清單中篩選條件的規則。如需詳細資訊，請參閱 管理對應規則 。
全部	全部	升級阿帕奇 Log4j 的 2.17.1 版本。
全部	Amazon Redshift	使用 COPY 命令中的 ENCRYPTED 子句添加了對數據遷移到 Amazon Redshift 的支持。
全部	Amazon Redshift	增強了資料擷取代理程式的 REST API。更新後的 REST API 增加了對新屬性的支援，例如加密金鑰、加密類型等。
全部	Amazon Redshift	在資料擷取代理程式中實作的角色。此更新可改善子工作的散佈，並允許將工作指派 AWS SCT 給指定角色的免費代理程式。
全部	Amazon Redshift	在將擴充套件套用至 Amazon Redshift 之前，實作是否已安裝所有必要元件的檢查。
突觸分析 Microsoft SQL 服務器 DW	Amazon Redshift	改進了用於錯誤處理的 ERROR_LINE ERROR_MESSAGE ERROR_NUMBER ERROR_PROCEDURE 、 、 ERROR_SEVERITY 、 、 和 ERROR_STATE 系統功能的轉換。
適用於 z/OS 的 IBM Db2	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	已新增對 z/OS 12 版 IBM Db2 的支援，做為中資料庫移轉專案的來源。AWS SCT 如需詳細資訊，請參閱 使用適用於 z/OS 的 IBM Db2 作為來源 。

來源	目標	最新消息、增強功能或修正錯誤
IBM Db2 LUW	全部	增強了源元數據加載器，以確保 AWS SCT 加載重複列名稱的常規參數。
Microsoft Azure SQL Database	Aurora PostgreSQL	修正了 SET NOCOUNT ON set 語句程序的變壓器錯誤。
Microsoft SQL Server	PostgreSQL	
Microsoft Azure SQL Database	Aurora PostgreSQL	當輸入值是使用者定義型別的變數時，已改善 CONCAT 函數的轉換。
Microsoft SQL Server	PostgreSQL	
Microsoft Azure SQL Database	Aurora PostgreSQL	解決了 DATEPART 函數轉換不正確的問題。
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	實現了對新版本的 Babelfish 功能配置文件的支持。此檔案定義了特定 Babelfish 版本所支援且不支援的 SQL 功能。
Microsoft SQL 服務器 DW	Amazon Redshift	解決了帶有 EXECUTE 語句的程序轉換不正確的問題。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft 社會服務	AWS Glue	改進了作業配置嚮導的用戶界面。AWS SCT 現在只會在連線組態區段中顯示可用的連線。
Microsoft 社會服務	AWS Glue	解決轉換規則未套用至封裝工作和變數規則的問題。
Microsoft 社會服務	AWS Glue AWS Glue Studio	為不支援的元件新增了新的行動項目 25042。
Microsoft 社會服務	AWS Glue Studio	實現 Microsoft SQL 服務器集成服務 (SSIS) 提取，轉換和加載 (ETL) 包的轉換。AWS Glue Studio 如需詳細資訊，請參閱 將 SSIS 轉換為 AWS Glue Studio 。
Oracle	MariaDB	修復了 MINUS 運算符轉換的問題。
Oracle	MariaDB	當 MariaDB 中 ROWNUM 的 sql_mode 系統變數為甲骨文時，改進了 TO_CHAR、和 ADD_MONTHS 函數的轉換。SYS_GUID
Oracle	PostgreSQL	新增了一個選項，以避免在一般應用程式轉換專案中將變數類型轉換為 SQL 類型。
Oracle	PostgreSQL	新增了一個選項，以避免在一般應用程式轉換專案中將結構描述名稱新增至轉換物件的名稱。
Oracle	PostgreSQL	增加了對應用程式 SQL 代碼轉換的 ?x 綁定變量格式的支持。
甲骨文 DW	Amazon Redshift	實現了 RAW 數據類型到數 VARBYTE 數據類型的轉換。
Teradata	Amazon Redshift	添加了一個選項來模擬轉換後的代碼中的 SET 表。對於此模擬，AWS SCT 支持 MIN 和 MAX 條件。
Teradata	Amazon Redshift	改進了具有不同數據類型參數的聯接操作的轉換。此更新可讓您 AWS SCT 在轉換此類作業期間套用轉換規則。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	解決了GROUP BY子句轉換不正確的問題。
Teradata	Amazon Redshift	解決了QUALIFY子句轉換不正確的問題。
Teradata	Amazon Redshift	已解決 FastExport 指令碼匯入期間發生未預期的錯誤。
Teradata	Amazon Redshift RSQL	實作了在 Teradata BTEQ 和殼層指令碼中編輯變數值的功能。
Teradata	Amazon Redshift RSQL	解決轉換的 Teradata FastLoad 工作階段遺失資訊清單指令碼的問題。
Teradata	Amazon Redshift RSQL	解決轉換FastLoad 指令碼的統一資源定位器 (URL) 中遺失資訊清單檔案副檔名的問題。
泰瑞数据	Amazon Redshift RSQL	修正了帶有替代變量腳本的加載器錯誤。
泰瑞数据	Amazon Redshift RSQL	修正行動項目 27022 在需要時不會出現的問題。

AWS SCT 組建 660 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	在多伺服器評估報告中新增對安全通訊端層 (SSL) 的 AWS Secrets Manager 支援。如需詳細資訊，請參閱 建立資料庫移轉的多伺服器評估報告 。
全部	全部	改善已轉換物件的統計資料收集。
全部	PostgreSQL	作為移轉目標實作對 PostgreSQL 主要版本 14 和 MariaDB 10.6 的支援。
突觸分析	Amazon Redshift	改善轉換物件名稱的轉換邏輯。
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	改進了 XML 數據類型的轉換。
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決了 NOT LIKE 子句轉換不正確的問題。
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	已修正具有 INSERT、DELETE 和包含 OUTPUT 子 UPDATE 句之程序的轉換器錯誤。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft Azure SQL Database	Aurora PostgreSQL	修正使用RETURN @@ROWCOUNT 陳述式程序的變壓器錯誤。
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	全部	改善使用連結伺服器之程序的轉換。
Microsoft SQL Server	全部	在多伺服器評估報告中新增對 Microsoft 視窗驗證的支援。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	修正表格值建構函式的變壓器錯誤。
Microsoft SQL 服務器 DW	Amazon Redshift 和 AWS Glue	改善擷取、轉換和載入 (ETL) 指令碼的轉換，以包含轉換指令碼的正確路徑。
Microsoft SQL 服務器 DW	Amazon Redshift	解決了針對虛擬和實際目標資料庫平台產生不同轉換指令碼的問題。
Oracle	PostgreSQL Aurora PostgreSQL	已新增對具體化視觀表索引轉換的支援。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Aurora PostgreSQL	修正轉換PRIMARY KEY和使用選項UNIQUE限制時，行動NOVALIDATE 項目 5982 不會出現的問題。
甲骨文 DW	Amazon Redshift	已解決轉換的結構描述中顯示其他類別的問題。
Teradata	Amazon Redshift	修正將未解析的欄轉換為函數引數時，行動項目 13185 不會出現的CAST問題。
Teradata	Amazon Redshift	改進了DELETE和DELETE ALL 語句的轉換以在轉換後的代碼中使用該TRUNCATE命令。
Teradata	Amazon Redshift	改進了SET表格的轉換。
Teradata	Amazon Redshift	改進了NORMALIZE 條件的轉換。
Teradata	Amazon Redshift	已更新評估報告，從資料庫儲存物件清單中移除資料庫結構描述轉換統計資料。
Teradata	Amazon Redshift	改進了沒有子句的UPDATE語FROM句的轉換。
Teradata	Amazon Redshift	在轉換後的代碼中實現了對VARBYTE數據類型的支援。
泰瑞数据	AWS Glue	解決了在內容功能表中停用「轉換為」AWS Glue選項的問題。
泰瑞数据	Amazon Redshift RSQL	解決轉換後的程式碼中遺失資料類型的問題。

來源	目標	最新消息、增強功能或修正錯誤
泰瑞数据	Amazon Redshift RSQL	解決了替代變量在轉換後的代碼中引用錯誤的問題。
泰瑞数据	Amazon Redshift RSQL	修正指FastLoad 令碼中使用值轉換替代變數的問題。
Vertica	Amazon Redshift	在轉換後的代碼中實現了對TIME數據類型的支持。
Vertica	Amazon Redshift	改進了SELECT DISTINCT和ORDER BY表達式的轉換。
Vertica	Amazon Redshift	增加了對約束轉換的支持。
Vertica	Amazon Redshift	解決評估報告未儲存為逗號分隔值 (CSV) 檔案的錯誤。

AWS SCT 組建 659 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	改進了新建項目嚮導，該嚮導可為多個源數據庫生成合併的評估報告。
全部	全部	修正在包含多個來源和目標資料庫的專案中未建立擴充功能套件的問題。
全部	全部	改善內嵌於應用程式原始程式碼中之 SQL 程式碼的轉換。
全部	全部	添加了在 AWS SCT 命令行界面中從不同文件夾運行腳本的功能。
全部	Amazon Redshift	改善使用者在使用 Amazon Redshift 虛擬目標資料庫平台的移轉專案中選擇執行最佳化時所提供的警告訊息。

來源	目標	最新消息、增強功能或修正錯誤
全部	Aurora PostgreSQL	在 Aurora 與 PostgreSQL 相容版本上實作 PostgreSQL 主要版本 13 的支援作為移轉目標。
全部	Amazon RDS for MySQL	默認情況下實現了不區分大小寫的代碼轉換。
突觸分析	Amazon Redshift	已解決命令列介面中與來源資料庫的連線失敗的錯誤。
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	改進了包含具有聯結條件之UPDATE陳述式的程序轉換。
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	改善觸發程序、預存程序和包含等號後面值的函數的轉換。
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	修正使用DELETE陳述式和OR運算子程序的變壓器錯誤。(

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	已改善OUTPUT子句的轉換。
Microsoft SQL 服務器 DW	Amazon Redshift 和 AWS Glue	改進了NUMERIC數據類型的轉換。
Microsoft SQL 服務器 DW	Amazon Redshift	改善具有與原始資料表名稱相同之資料表別名之檢視的轉換。
Microsoft 社會服務	AWS Glue	修正「設定 AWS Glue 連線」視窗中未顯示連線認證的問題。
Netezza	Amazon Redshift	新增每天重複執行變更資料擷取 (CDC) 資料移轉工作的功能。
Netezza	Amazon Redshift	修正取消註冊資料擷取代理程式後，「工作」標籤會變成非作用中的問題。
Netezza	Amazon Redshift	修正使用者介面中未顯示資料遷移代理程式註冊確認的問題。
Netezza	Amazon Redshift	修正與來源資料庫的連線失敗並出現 Loader 錯誤的問題。
Netezza	Amazon Redshift	解決開啟儲存的專案後，資料移轉代理程式無法執行的錯誤。
Oracle	Amazon RDS for Oracle	已實作 Oracle 統一稽核的支援。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Aurora PostgreSQL	在 C# 應用程序中實現了 SQL 代碼轉換。如需詳細資訊，請參閱 在 C# 應用程序中轉換 SQL 代碼 。
Oracle	PostgreSQL Aurora PostgreSQL	針對區分大小寫的物件名稱實作新的轉換邏輯，以改善程式碼轉換變更的可見度。AWS SCT 將大寫的物件名稱轉換為小寫。相反的情況也是如此；AWS SCT 將小寫的物件名稱轉換為大寫。其他物件名稱和保留字會在不變更的情況下轉換。
Oracle	PostgreSQL Aurora PostgreSQL	改進了哈希分區的轉換，沒有 NOT NULL 約束。
Oracle	Aurora PostgreSQL	增加了對 Oracle CHECK FOREIGN KEY、和 ENABLE NOVALIDATE 條 NOT NULL 約束的轉換支援。
甲骨文 DW	Amazon Redshift	修正移轉浮點數字不正確值的問題。
甲骨文 DW	Amazon Redshift 和 AWS Glue	已解決資料庫移轉評估報告中逗號分隔值 (CSV) 檔案中空白欄的問題。

來源	目標	最新消息、增強功能或修正錯誤
SAP ASE	PostgreSQL Aurora PostgreSQL	修正意外轉換中斷的問題。
Snowflake	Amazon Redshift	改進了VARIANT數據類型的轉換。
Teradata	Amazon Redshift	改進了COLLECT STATISTICS 語句的轉換。
Teradata	Amazon Redshift	修正在轉換含PERIOD欄的巢狀檢視時，行動項目 9998 不會出現的問題。
Teradata	Amazon Redshift 和 AWS Glue	修正開啟儲存的專案後，虛擬 AWS Glue 目標平台未顯示在 UI 中的問題。
泰瑞数据	AWS Glue	修正開啟儲存的專案後，不支援轉換為虛擬 AWS Glue 目標平台的問題。
泰瑞数据	Amazon Redshift RSQL	改進了轉換後代碼的語法突出顯示。
泰瑞数据	Amazon Redshift RSQL	實作上傳後檢查參數值。不支援的值會在「變數」標籤上反白顯示。
Vertica	Amazon Redshift	實作彙總函式的轉換。
Vertica	Amazon Redshift	實作投影到具體化視圖的轉換，並改善了顯示投影原始程式碼的 UI。

AWS SCT 組建 658 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	提供與 AWS Secrets Manager. 您現在可以使用儲存在 Secrets Manager 中的資料庫連線認證。
全部	全部	在 AWS SCT 命令列介面中新增對 YAML 格式的指令碼支援。
全部	Amazon Redshift	在資料擷取代理程式中實作 Amazon S3 介面端點 (VPCE) 的支援。
全部	Amazon Redshift	除了已經支持的 Amazon Redshift 和 AWS Glue 組合外，還添加了對 Amazon Redshift 虛擬目標數據庫平台的支持。
Greenplum	Amazon Redshift	修正「另存為 SQL」選項未將轉換後的 SQL 程式碼儲存至檔案的問題。
IBM Db2 LUW	Aurora MySQL	改進了轉換以支持具有 MySQL 8.0 兼容性的 Amazon Aurora MySQL 兼容版本的新功能。
Microsoft Azure SQL Database		
Microsoft SQL Server		
Oracle		
SAP ASE		
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL	修正行動項目 810 在需要時不會出現的問題。

來源	目標	最新消息、增強功能或修正錯誤
	MySQL	
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	改進了使用UPDATE、DELETE和INSERT語句的程序轉換。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	修正行動項目 7810 在需要時不會出現的問題。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	已改善巢狀EXEC陳述式內部陳IF...ELSE 述式的轉換。
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	已改善索引檢視的轉換。
	PostgreSQL	
Netezza	Amazon Redshift	透過在變更資料擷取 (CDC) 作業中全負載期間追蹤即時交易，藉此改善資料移轉代理程式。如果 CDC 工作階段已排定在特定時間啟動，您現在可以停止資料移轉工作。此外，您可以在使用 CDC 停止工作之後，在主控台中看到錯誤記錄層級。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	全部	增強了表格載入器，以確保使用共用選項 AWS SCT 載入物件。
Oracle	Aurora PostgreSQL L PostgreSQL L	改進了SYSDATE功能的轉換，並在轉換設置中添加了更改時區的功能。
Oracle	Aurora PostgreSQL L PostgreSQL L	解決動態陳述式未轉換的問題。
Oracle	Aurora PostgreSQL L PostgreSQL L	修正轉換後的程式碼不包含系統產生名稱的問題。
Oracle 甲骨文 DW	Aurora PostgreSQL L PostgreSQL L	改進了嵌套在觸發器中的SELECT語句的轉換。
甲骨文 DW	Amazon Redshift	改善延伸套件中TO_DATE_TO_TIMESTAMP、和TO_TIMESTAMP_TZ 功能的轉換。
Snowflake	Amazon Redshift	添加了一個選項，以將轉換後的 SQL 代碼保存在每個對象或每個語句的不同文件中。
Teradata	Amazon Redshift	改進了CONCAT功能的轉換。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	已改善巢狀在WHERE子句中之SELECT陳述式的轉換。
Teradata	Amazon Redshift	已解決使用者卸除SET並重新建立MULTISET資料表之後不正確轉換和資料表的問題。
Teradata	Amazon Redshift	改進了包含WITH子句的程序的轉換。
Teradata	Amazon Redshift	改進了DATE數據類型的轉換。
Teradata	Amazon Redshift RSQL	解決在FastExport 指令碼轉換期間發生非預期的轉換器錯誤的問題。
泰瑞数据	Amazon Redshift RSQL	已新增將聯結索引轉換為具體化視觀表的支援。
泰瑞数据	Amazon Redshift RSQL	已新增對包含多行之TITLE定義的轉換支援。
泰瑞数据	Amazon Redshift RSQL	解決地理空間資料類型大小未轉換的問題。
泰瑞数据	Amazon Redshift RSQL	修正參數名稱轉換為小寫字元的問題。
泰瑞数据	Amazon Redshift RSQL	修正巢狀MACRO陳述式中的預存程序未轉換的問題。(

來源	目標	最新消息、增強功能或修正錯誤
Vertica	Amazon Redshift	改進了ALL運算符的轉換。
Vertica	Amazon Redshift	解決未套用「轉換」設定中的Use Union all view?選項的問題。
Vertica	Amazon Redshift	改進了TIME和TIME WITH TIMEZONE數據類型的轉換。
Vertica	Amazon Redshift	解決了加載彈性表格的問題。

已解決的問題：

- 一般改善。

組建 657 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	將 Apache Log4j 升級至 2.17 版以解決安全漏洞問題。
全部	Amazon Redshift	改進的結構描述優化項目，其中密鑰管理統計信息未保存在項 AWS SCT 目中。
Amazon Redshift	Amazon Redshift	修正伺服器資訊更新的問題。
阿帕奇·卡桑德拉	Amazon DynamoDB	修正使用 AWS SCT 指令列介面時對應規則的問題。
阿帕奇·卡桑德拉	Amazon DynamoDB	已解決因為憑證中的標題更新而未建立遷移任務的問題。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	修正 Microsoft SQL Server 程序與動態 SQL 轉換期間，行動項目 7672 不會出現的問題。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改善資料表值函數的轉換。
天青 SQL 數據庫 Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	解決了具有默認返回值的存儲過程中的參數未轉換為INOUT引數的問題。OUT
Greenplum	Amazon Redshift	通過從表中找到最常用的表和列改進的QueryLog優化策略。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	修正下列項目轉換的問題： <ul style="list-style-type: none"> • 字符串連接賦值運算符 () += • SCOPE_IDENTITY 函數 • varchar(max) 資料類型

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	改進了具有不支持功能的視圖轉換。
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	修正了不受支援的函數作為另一個函數的引數轉換錯誤的問題。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	改善轉換表參照的轉換。
Microsoft SQL 服務器 DW	Amazon Redshift	已將彙總函式類別新增至來源資料庫中繼資料樹狀結構。
Microsoft SQL 服務器 DW	Amazon Redshift	改進了TIME數據類型的轉換。

來源	目標	最新消息、增強功能或修正錯誤
突觸分析 Greenplum Netezza Microsoft SQL 服務器 DW Snowflake Teradata	Amazon Redshift	修正使用虛擬目標資料庫平台時未儲存DROP和CREATE指令碼的問題。
Microsoft SQL 服務器集成服務	AWS Glue	已解決來源物件指令碼未顯示在 UI 中的問題。
Netezza	Amazon Redshift	通過選擇事實表和適當的維度進行搭配，改進了優化策略。
Oracle	Aurora PostgreSQL PostgreSQL	已解決正確轉換使用序號之 Oracle 觸發程式的問題。
Oracle	Aurora PostgreSQL PostgreSQL	改進了具有公共數據庫鏈接的視圖轉換。
甲骨文 DW	Amazon Redshift	藉由分析索引資料行的基數，改善最佳化策略。

來源	目標	最新消息、增強功能或修正錯誤
甲骨文 DW	Amazon Redshift	修正具有字串串連的自訂使用者定義純量函數轉換錯誤的問題。
Snowflake	Amazon Redshift	修正「另存為 SQL」選項未顯示在 UI 中的問題。
Teradata	Amazon Redshift	修正統計資料收集失敗，但發生LOADER ERROR例外狀況的問題。
Teradata	Amazon Redshift	修正使用者介面中未顯示 [建立報告] 選項的問題。
Teradata	Amazon Redshift	改進了CAST功能的轉換。
Teradata	Amazon Redshift	修正了中斷的轉換ST_Line_Interpolate_Point 。
Teradata	Amazon Redshift	從 Python 庫路徑中刪除了一個意外的值。
Teradata	Amazon Redshift RSQL	修復了在轉換多 FastLoad個腳本過程中出現的解析器錯誤。
泰瑞数据	Amazon Redshift RSQL	改進了DATABASE命令和幾何數據類型的轉換。
泰瑞数据	AWS Glue	修正 UI 中來源指令碼和目標指令碼同步處理不正確的問題。

已解決的問題：

- 一般改善。

AWS SCT 組建 656 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	在一個項目中添加了對多個源和目標數據庫的支持。使用者現在可以建立對應規則，以符合相同專案中的不同資料庫結構描述和目標平台。
全部	全部	增加了對虛擬目標數據庫平台的支持。使用者現在不需要連線到目標資料庫，即可查看來源資料庫結構描述的 AWS SCT 轉換方式。
全部	全部	<p>用戶界面改進：</p> <ul style="list-style-type: none"> • 添加了「Connect 到服務器」和「斷開與服務器斷開」選項到源和目標元數據樹中。 • 已新增從專案中移除資料庫伺服器的選 AWS SCT 項。
Cassandra	Amazon DynamoDB	已解決CASSANDRA_HOME 變數之後cassandra.yaml 或conf資料夾未包含斜線 (/) 的搜尋問題。
Cassandra	Amazon DynamoDB	增加了對 Amazon Linux 2 的 Amazon 機器映像 (AMI) 的支持。
Cassandra	Amazon DynamoDB	改進了卡桑德拉給出了不正確的密鑰時提供的錯誤消息。
Cassandra	Amazon DynamoDB	通過根據目標數據庫的版本更改cassandra-env.yaml 文件中的屬性來改進轉換。
Cassandra	Amazon DynamoDB	增加了目標卡桑德拉數據中心的 Java 版本到 1.8.0。
Greenplum	Amazon Redshift	改進了項目設置中的優化策略。

來源	目標	最新消息、增強功能或修正錯誤
Greenplum	Amazon Redshift	已解決資料移轉問題，即物件未套用至資料庫，並發生下列錯誤：An I/O error occurred while sending to the backend。
Greenplum Microsoft SQL 服務器 DW	Amazon Redshift	解決選Apply RTRIM to string columns項未顯示在 UI 中的問題。
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	新增對 Aurora PostgreSQL 作為目標平台的巴貝爾魚支援。使用者現在可以建立評估報告，以估計從 SQL 伺服器遷移到 Aurora PostgreSQL 的巴貝爾魚。
Netezza	Amazon Redshift	改進了項目設置中的優化策略。
SAP ASE	Aurora PostgreSQL PostgreSQL	實現了為索引生成唯一名稱的功能。
SAP ASE	Aurora PostgreSQL PostgreSQL	修正目標指令碼中重複索引欄的問題。
Snowflake	Amazon Redshift	解決「隱藏空白結構描述」、「隱藏空白資料庫」和「隱藏系統資料庫/結構描述」選項未顯示在 UI 中的問題。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift RSQL	增加了對將太數據任務 MultiLoad 腳本轉換為 Amazon Redshift RSQL 腳本的支持。
Teradata	Amazon Redshift RSQL	修正了在 FastLoad 和 FastExport 腳本中轉換替代變量的問題。
Teradata	Amazon Redshift RSQL	修正從 [摘要] 索引標籤切換之後，行動項目未顯示在 [行動項目] 索引標籤中的問題。
Teradata	Amazon Redshift RSQL	已解決在 FastExport 指令碼轉換期間產生報表後發生錯誤的問題。
Teradata	Amazon Redshift RSQL	解決殼層指令碼轉換後的格式化問題。
Teradata	Amazon Redshift RSQL	修正了 AI 13177 現在在轉換後的指令碼中加上註解的問題。
Teradata	Amazon Redshift	修正時間表格中斷的轉換問題。
Teradata	Amazon Redshift	已改善 SET QUERY_BAND 陳述式的轉換。
Teradata	Amazon Redshift	修正了 NORMALIZE 操作中斷的轉換。
Vertica	Amazon Redshift	改進了人工智能 17008 的描述。

已解決的問題：

- 一般改善。

AWS SCT 組建 655 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift RSQL	修正了確保所有評估問題在使用 FastLoad 或使用時都出現在報告中 MultiLoad 的問題。
Teradata	Amazon Redshift RSQL	增加了對將太數據任務 FastExport 腳本轉換為 Amazon Redshift RSQL 腳本的支持。
Teradata	Amazon Redshift RSQL	修正確保在使用時在離線模式下啟用「將資訊清單儲存至 S3」動作的問題 FastLoad。
Teradata	Amazon Redshift RSQL	已修正確保對應規則套用於指令碼的問題，例如 FastLoad。
Greenplum	Amazon Redshift	將綠色的最低支持驅動程序版本提高到 42.2.5。
Greenplum	Amazon Redshift	增加了通過驅動程序版本 42.2.5 或更高版本的 SSL 連接到綠色。
甲骨文 DW	Amazon Redshift	改善在另一個 UDF 中執行自訂使用者定義純量函數 (UDF) 的支援。
甲骨文 DW	Amazon Redshift	修正函數未套用至資料庫的問題，而發生此錯誤：Failed to compile udf。
甲骨文 DW	Amazon Redshift	使用適當的型別宣告 (例如，pls-type 針對 %ROWTYPE 參數) 來改善轉換。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift RSQL	解決報告中未顯示資訊類型評估問題的問題。
Teradata	Amazon Redshift RSQL	已解決轉換部分指令碼後的變壓器錯誤。
Teradata	Amazon Redshift RSQL	修正了一個問題，以便現在在轉換的腳本中註釋了一個問題。
Teradata	Amazon Redshift	解決了轉換後- FastExport > 導出-> 「空」而不是顯示「CAST」的問題。
Teradata	Amazon Redshift	解決了Cause:[JDBC Driver]String index out of range: 0如果使用驅動程序版本 1.2.43，則應用擴展包的某些功能失敗的問題
Teradata	Amazon Redshift	SET 資料表轉換 — 設定為插入選取陳述式新增的表格模擬。
Teradata	Amazon Redshift	鑄造-支援其他資料類型轉換。
Teradata	Amazon Redshift	修正「其他目前 _ 時間 _01」的中斷轉換
Teradata	Amazon Redshift	太數據 FastExport -Amazon Redshift RSQL：改進了太拉數據命令字段的轉換 FastExport
Teradata	Amazon Redshift	太數據 FastExport — Amazon Redshift RSQL：改進了 Teradata 命令 — 佈局的轉換 FastExport

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Aurora PostgreSQL	解決了具有「保存例外」語句的對象的目標腳本在重新轉換後發生更改的問題。
Oracle	PostgreSQL Aurora PostgreSQL	解決proc_cursor_with_calc_columns 轉換後在 ORDER BY 子句中指定錯誤欄位的問題。
Oracle	PostgreSQL Aurora PostgreSQL	解決：關聯集合轉換中需要額外的aws_oracle_ext\$array_id \$temporary 變量聲明。
Oracle	PostgreSQL Aurora PostgreSQL	解決：主鍵的錯誤轉換與同一個表擁有的索引相同的名稱。

已解決的問題：

- 一般改善。

組建 654 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL L Aurora PostgreSQL L	已解決階層查詢虛擬資料欄，PRIOR 資料行剖析錯誤的問題。
Oracle	PostgreSQL L Aurora PostgreSQL L	已解決正確轉換包含斜線和星號 (/*) 的多行註解的問題。
Oracle	PostgreSQL L Aurora PostgreSQL L	增加了系統視圖用戶 _COL_ 評論模擬到擴展包。
Oracle	PostgreSQL L Aurora PostgreSQL L	改進了引用文字的轉換。
DB2 LUW	PostgreSQL L Aurora PostgreSQL L	改善 LABEL 陳述式的轉換，這些陳述式會在表格、檢視表、別名或欄的描述中新增或取代標籤。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	無	使用 DBA_USERS 視圖和改進的查詢取代的 SYS.USER \$ 系統表。
甲骨文 DW	Amazon Redshift	更新了甲骨文 DW 元數據查詢。
Teradata	Amazon Redshift RSQL	添加了對外殼，太數據和太數據 FastLoad 基本查詢 (BTEQ) 腳本轉換為 Amazon Redshift RSQL 腳本的支持。
泰瑞数据	Amazon Redshift RSQL	已解決「merge_01」轉換不正確的問題。
泰瑞数据	Amazon Redshift RSQL	已解決「結束」或「識別」(EOI) 出現在新行指令碼結尾的問題。
Azure Synapse	Amazon Redshift	改進了為 Azure 突觸提供了不正確的密碼時提供的錯誤消息。
Teradata	Amazon Redshift	改善 UPDATE 陳述式轉換功能，可結轉每個 Teradata 標準的正確別名名稱。
Teradata	Amazon Redshift	已解決未收到動作的游標轉換錯誤。
Teradata	Amazon Redshift	已解決 TD_ 正常化 _ 重疊轉換會丟棄資料列的問題。
Teradata	Amazon Redshift	現在支援增強型 TO_DATE 函數的嚴格日期檢查。
Teradata	Amazon Redshift	改進了內置函數轉換為 _NUMBER (n) 。
Teradata	Amazon Redshift	解決了元數據樹中缺少「架構」類別的問題。

來源	目標	最新消息、增強功能或修正錯誤
Greenplum	Amazon Redshift	在為綠色資料表建立虛擬磁碟分割時，將 GP_SEGMENT_ID 選項新增到清單中。
Greenplum	Amazon Redshift	已解決目標上未套用函式的問題。
微軟 SQL 服務器數據庫	Amazon Redshift	解決在沒有 AI 9996 的情況下轉換後發生轉換錯誤的問題。
微軟 SQL 服務器數據庫	Amazon Redshift	解決開啟擴充功能套件精靈時記錄錯誤的問題。
微軟 SQL 服務器數據庫	Amazon Redshift	已解決當 Redshift Python 函式使用不正確的註解樣式時的問題。
Netezza	Amazon Redshift	解決了具有設定檔的 Netezza — Redshift 擴充套件無法建立的問題。 AWS
Teradata	Amazon Redshift RSQL	改進了 FastLoad 會話命令的轉換。
Teradata	Amazon Redshift RSQL	改進了 FastLoad 腳本評估報告。
Teradata	Amazon Redshift RSQL	實 FastLoad 作寫入器儲存至 S3 動作。
Teradata	Amazon Redshift RSQL	解決了 FastLoad 保存腳本\ 將清單保存到 s3 按鈕未激活的問題。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift RSQL	解決 FastLoad multifile_script 在轉換後僅建立一個資訊清單檔案，而非預期的三個檔案的問題。
Teradata	Amazon Redshift RSQL	解決 S3 路徑中顯示額外資料夾的問題。FastLoad
Teradata	Amazon Redshift RSQL	解決 S3 路徑中資訊清單檔案名稱不正確的問題。FastLoad

已解決的問題：

- 一般改善。

組建 653 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL L Aurora PostgreSQL L	實現了轉換在調用函數或過程中創建的動態 SQL 的功能。
Oracle	PostgreSQL L Aurora PostgreSQL L	改進了動態 SQL 轉換：作為綁定變量的參數。

來源	目標	最新消息、增強功能或修正錯誤
甲骨文	Amazon Redshift	實現了甲骨文到紅移轉換的改進：增強的轉換內置。聚合列表；分析列表。
甲骨文	Amazon Redshift	已實施甲骨文到紅移轉換的改進：查詢新功能。
Vertica	Amazon Redshift	實現了 Vertica 到 Redshift 轉換的改進：SSL 到 JDBC 的連接與 SSL = 真。
微軟 SQL 服務器數據庫	Amazon Redshift	MS SQL 服務器到 Redshift 轉換的改進：外部表。
Teradata	Amazon Redshift	太數據到 Redshift 轉換的改進：間隔數據類型算術運算。
Teradata	Amazon Redshift	Teradata 到 Redshift 轉換的改進：Support 橫向資料行別名。
Oracle	無	下列載入程式查詢現在使用，DBA_USERS 而不是SYS.USER\$： <ul style="list-style-type: none"> • get-tree-path-listby-name-path。-SQL • estimate-table-or-viewconstraints-by-schema。-SQL • estimate-table-or-viewconstraints-by-selected-schemas。-SQL
Teradata	Amazon Redshift	改善 SCT 將 Teradata 巨集轉換為 Redshift 預存程序時註解的對齊方式。
甲骨文 DW	Amazon Redshift	改進了日期/時間戳記格式元素的轉換：TO_DATE TO_TIMESTAMP、和 TO_TIMESTAMP_TZ
Teradata	Amazon Redshift	解決了 Teradata 光標轉換錯誤。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	已解決在轉換期間造成的屬性遭TD_NORMALIZE_OVERLAP 到捨棄的問題。
Teradata	Amazon Redshift	解決 SCT 轉換查詢時忽略MAX函數的問題。
Teradata	Amazon Redshift	SCT 現在轉換大數據字符函數 Redshift 長度函數。
Teradata	Amazon Redshift	SCT 現在支持將格式轉換為 TO_CHAR，用於最常用的格式。
全部	全部	改進了加密常式的轉換。

已解決的問題：

- 一般改善。

組建 652 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	PostgreSQL	添加了應用程式鎖定sp_getapplock 和sp_releaseapplock 功能。
無	Amazon Redshift	命令行界面 (CLI) 改進：實現腳本命令模式。
Oracle	PostgreSQL Aurora PostgreSQL	在動態 SQL 中實現了例行參數採樣。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Aurora PostgreSQL	轉換改進到在調用函數或過程中創建的動態 SQL。
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL	每個 lambda 函數只會透過政策部署和設定一次，而常見的 lambda 函數則會重複用於所有可能的來源。
DB2 LUW	PostgreSQL	已解決使用 DB2 LUW 作為來源時，造成錯誤訊息「9996 — 嚴重性嚴重性 — 發生變壓器錯誤」的問題。
Teradata	Amazon Redshift	在即將推出的 Amazon Redshift 中 Support 遞迴資料表運算式。
Azure Synapse	Amazon Redshift	實作結構描述優化規則。
Teradata	Amazon Redshift	Support 從 Teradata 宏到 Redshift 存儲過程的時區轉換。
Teradata	Amazon Redshift	Support 週期值的算術。
Teradata	Amazon Redshift	Support 轉換 Teradata 遞迴通用資料表運算式 (遞迴 CTE)。
Teradata	Amazon Redshift	透過使用者設定 Support 區分大小寫的識別碼 <code>enable_case_sensitive_identifier</code> 。因此，「列名」和「列名稱」成為不同的列名。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	解決了十進制數據類型問題，以便十進制字段以相同的精度進行轉換。
Teradata	Amazon Redshift	已解決間隔算術轉換的問題，以便間隔算術減法正確轉換。
Teradata	Amazon Redshift	改進太數據數到日期類型轉換。
Teradata	Amazon Redshift	改進太拉數據日期到數字類型轉換
泰瑞数据	Amazon Redshift	改善期間資料類型轉換。
Teradata	Amazon Redshift	已解決載入含有 GEOMETRY 欄之表格中繼資料的問題，現在可以正確地從 Teradata 載入。
Teradata	Amazon Redshift	將 Teradata 宏轉換為 Redshift 存儲過程時，Support 合併語句的轉換。
Teradata	Amazon Redshift	改進了從 Teradata 遷移到紅移時簡單宏的轉換。
Teradata	Amazon Redshift	確保 Teradata UPDATE 陳述式的轉換會傳遞每個 Teradata 標準的正確別名。

已解決的問題：

- 一般改善。

組建 651 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	增強型 AWS SCT 報告可更新列出之建議轉換動作項目的連結。

來源	目標	最新消息、增強功能或修正錯誤
SQL 伺服器	PostgreSQL	增加了對STR()功能轉換的支持。
SQL 伺服器	PostgreSQL	新增將位元 EXOR 運算子 (^在 Microsoft SQL 伺服器中) 轉換為 PostgreSQL 作為運算子的支援。#
Oracle	PostgreSQL	解決了 AWS SCT 擴展包aws_oracle_ext.UNISTR(null) 功能在 PostgreSQL 目標NULL上掛起的問題。AWS SCT 現在處理NULL。
泰瑞数据	Amazon Redshift RSQL	改進了轉換，以解決 Amazon Redshift RSQL 合併轉換發生轉換錯誤的問題。
甲骨文 DW	Amazon Redshift	實現了增強的內置插件。
甲骨文 DW	Amazon Redshift	新增中繼資料功能驅動的增強功能，包括自動清單分割 (TBL_零件_清單_自動)、多欄清單 (TBL_零件_多重_清單) 和間隔參考 (TBL_零件_範圍_英特瓦_參考)。
無	Amazon Redshift	增加了用於UNION ALL轉換的物理分區的分區表限制。
Teradata	Amazon Redshift	改善評估報告範圍的轉換。
Teradata	Amazon Redshift	對複雜的 Teradata 宏轉換進行了轉換改進。
Teradata	Amazon Redshift	改善將 Teradata 巨集轉換為 Amazon Redshift 儲存程序，同時註解不受支援的 SQL。
Teradata	Amazon Redshift	解決將 Teradata 巨集轉換為 Amazon Redshift 存放程序導致別名參照錯誤的問題。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	改善 Teradata QUALIFY 陳述式的轉換。
Teradata	Amazon Redshift	改進了轉換以將註釋轉發到 Amazon Redshift，並保留在檢視上執行的變更歷史記錄。
Teradata	Amazon Redshift	解決 RESET WHEN 子句無法導致正確轉換的問題。
泰瑞数据	Amazon Redshift	改進了包含 MERGE 語句的 BTEQ 腳本的轉換。
Teradata	Amazon Redshift	新增內建函數，以改善「期間」資料類型欄位的轉換。
Microsoft SQL Server	Amazon Redshift	增強了 TIME 資料類型的轉換資料類型對應。
全部	全部	新增了對初始發行 PDF 格式的 AWS Schema Conversion Tool CLI 參考手冊的存取權。請參閱 AWS Schema Conversion Tool CLI 參考資料 。

已解決的問題：

- 一般改善。

AWS SCT 組建 650 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	更新和增強了提取器代理的使用，包括： <ul style="list-style-type: none"> • 與共用儲存體和專用複製代理程式搭配使用的組態。 •

來源	目標	最新消息、增強功能或修正錯誤
		<p>將資料萃取工作從一個專案匯出和匯入到另一個專案。</p> <ul style="list-style-type: none"> • Support Azure SQL 資料倉儲 (Azure 突觸) 做為來源。 • 使用本地奈泰扎分區。 <p>如需詳細資訊，請參閱 將資料從現場部署資料倉儲遷移到 Amazon Redshift。</p>
全部	Amazon	AWS SCT 現在支援 Amazon PostgreSQL 做為目標。
Microsoft SQL Server	Aurora PostgreSQL	改善從 Microsoft SQL 伺服器程序到 Aurora PostgreSQL 目標的結果集的轉換。
甲骨文 DW	Amazon Redshift	實施甲骨文到 Amazon Redshift 轉換改進。
甲骨文 DW	Amazon Redshift	已實作轉換動態 SQL 陳述式的改良功能。
甲骨文 DW	Amazon Redshift	已實作 SQL UDF 轉換的改良功能。
甲骨文 DW	Amazon Redshift	AWS SCT 不支援外部資料表轉換的澄清訊息。
甲骨文 DW	Amazon Redshift	增強的內置轉換功能。
泰瑞数据	Amazon Redshift RSQL	改進了在使 AWS SCT 用 GUI 時處理 BTEQ 腳本中的替代參數。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL 服務器 DW	全部	升級了 Microsoft SQL 服務器，Azure，Azure 突觸的最低支持 JDBC 驅動程序版本。
Microsoft SQL Server		
Azure		
Azure Synapse		

已解決的問題：

- Teradata：宏轉換的其他改進 [已解決]
- 在目標中轉義特殊字符導致 SQL 錯誤和重新工作需要將它們放回 [已解決]
- 一般改善

組建 649 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL 服務器 DW	Amazon Redshift	MSSQL 到 Amazon Redshift 轉換改進，以支持臨時表。
甲骨文 DW	Amazon Redshift	實作內建功能增強功能，例如： 轉換函數 <ul style="list-style-type: none"> • TO_BINARY_DOUBLE • TO_BINARY_FLOAT

來源	目標	最新消息、增強功能或修正錯誤
		<ul style="list-style-type: none"> • TO_NUMBER • TO_DATE • TO_TIMESTAMP • TO_TIMESTAMP_TZ • TO_DSINTERVAL • TO_YMINTERVAL • 驗證轉換 (_C)
甲骨文 DW	Amazon Redshift	<p>已實作近似查詢處理的函數增強功能，例如：</p> <p>彙總函數</p> <ul style="list-style-type: none"> • ANY_VALUE • APPROX_COUNT_DISTINCT • 大約 _ 計數 _ 不同詳細資訊 • 大約值 _ 不同的計數 • LISTAGG • 到大約不同計數 (_L)
Teradata	Amazon Redshift	<p>實作 Teradata auto 排序和分發金鑰選取的轉換增強功能。數據庫引擎會自動選擇分發和排序鍵。引入了標示為「將 Amazon Redshift 自動表格調整用於目前專案設定 > 最佳化策略 > 初始金鑰選擇策略對話方塊的選項按鈕。</p>

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	增強的 AWS SCT 表格載入器，以確保從 Teradata AWS SCT 載入所有表格。
Teradata	Amazon Redshift	實作轉換增強功能，讓 Amazon Redshift 支援包含簡單「不存在」子句的相關子查詢模式。
Teradata	Amazon Redshift	增加了對在宏中使用 ECHO 命令的支持。
DB2 LUW	PostgreSQL Aurora PostgreSQL	已實作動態結果集轉換的支援，包括： <ul style="list-style-type: none"> 帶返回/返回客戶端的光標子句 動態結果集例程子句轉換
Microsoft SQL Server Oracle DB2 LUW SAP ASE	Aurora PostgreSQL	實作對目前的 Aurora RDS PostgreSQL 作為目標的支援。
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	實現了對 MariaDB 10.5 作為目標的支持。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	MariaDB	實現了對返回插入行的結果集的支持。
Oracle	Aurora PostgreSQL	增加了對從甲骨文轉換為 Aurora PostgreSQL 的 XML 森林功能的支持。

已解決的問題：

- 一般改善。

組建 648 AWS SCT 版的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Amazon Aurora PostgreSQL-Compatible Edition	Aurora PostgreSQL 擴展包自定義應用模式實現：用於數字/日期和文本類型的運營商。
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL	<p>已實作 Aurora Lambda 叫用組態：aws_lambda 擴充功能建立；IAM 角色指派給 Aurora PostgreSQL 叢集。</p> <ul style="list-style-type: none"> • Oracle — 電子郵件、工作、佇列、WebAgent檔案 • DB2-電子郵件，任務，文件 • Microsoft SQL 服務器-電子郵件，代理

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL	實現了 FORALL 語句轉換重構： <ul style="list-style-type: none"> • 福勒聲明 • 為所有... 儲存例外 • 使用批量收集返回 • SQL% 批量異常系統集合
甲骨文	Amazon Redshift	實施甲骨文到 Amazon Redshift 轉換改進：增強的轉換內置。聚合列表; 分析列表。
甲骨文	Amazon Redshift	實施甲骨文到 Amazon Redshift 轉換改進：查詢新功能。
Vertica	Amazon Redshift	實施了 Vertica 到 Amazon Redshift 轉換的改進：SSL 到 JDBC 的連接與 SSL = 真。
Microsoft SQL 服務器 DW	Amazon Redshift	Microsoft SQL 服務器到 Redshift 轉換的改進：外部表。
Teradata	Amazon Redshift	太數據到 Redshift 轉換的改進：間隔數據類型算術運算。
Teradata	Amazon Redshift	Teradata 到 Redshift 轉換的改進：Support 橫向資料行別名。

已解決的問題：

- 一般改善

AWS SCT 組建 647 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL Server	Microsoft SQL Server	RDS 現在支援資料庫郵件功能。
Microsoft SQL Server	MySQL	<p>實作每種識別碼類型的最大名稱 — SQL Server 中物件名稱 (例如資料表、條件約束、資料行) 的最大長度為 128 個字元。在 MySQL 中的對象名稱的最大長度為 64 個字符。要將轉換後的對象寫入 MySQL 數據庫，您需要縮短它們的名稱。為了防止剪切後重複名稱，您需要將原始對象名稱的「校驗和」添加到新名稱中。</p> <p>剪下名稱長度超過 64 個字元，如下所示：</p> <pre>[first N chars]() + "" + [checksum]()</pre> <p>[first N chars] = 64 - 1 - [length of checksum string]</p> <p>例如：</p> <pre>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</pre>
Oracle	MySQL/ Aurora MySQL	實作儲存物件註解的載入與轉換。例如，處理表格上的註解，以及處理表格/檢視欄上的註解。
Teradata	Amazon Redshift	增加了對 TIME 數據類型轉換的支持。
Teradata	Amazon Redshift	轉換改進 — 實施了 TD_ 正常化 _ 重疊。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft SQL 服務器 DW	Amazon Redshift	轉換改進-使用 WITH 子句選擇; 選擇不從
全部	全部	AWS SCT 資料移轉服務評估員 (DMSA) — 這項新功能可讓您評估多部伺服器，並接收摘要報告，以顯示環境的最佳目標方向。
全部	全部	AWS SCT 精靈 — 目標比較現在會在單一表格檢視中顯示目標之間的差異。
全部	全部	樹狀篩選器 UI — 重新設計的中繼資料篩選可處理更複雜的篩選
全部	全部	評估報告 — 重新設計的「警告」區段可提供更好的說明並更清楚地瞭解問題。

已解決的問題：

- 一般改善
- 資料擷取器 — 子工作失敗， ConcurrentModificationException [已解決]。
- Microsoft SQL 服務器到 MySQL-最大標識符長度 [已解決]。

AWS SCT 組建 646 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL	改善 TM 格式模型實作。
Oracle	PostgreSQL	SP 格式掩碼實現僅針對英文語言提供 SP 後綴的基本支持。
Oracle	PostgreSQL	Oracle 長對象名稱處理 — AWS SCT 現在根據目標標識符最大標識符長度屬性處理 Oracle 長對象名稱。

來源	目標	最新消息、增強功能或修正錯誤
	Amazon Redshift	使用 Amazon Redshift 編碼 AZ64 AWS SCT — 為某些數據類型添加了壓縮編碼 AZ64
Teradata	Amazon Redshift	增加了對隱式交易轉換的支持。
Teradata	Amazon Redshift	增加了對 Teradata 地理空間內置函數的支持：ST_LineString 方法
Greenplum	Amazon Redshift	Greenplum 序列轉換-將下一個項目添加到屬性選項卡中：最小值，最大值，增量，循環。
Greenplum	Amazon Redshift	解析器-增加了「char」數據類型解析。
Greenplum	Amazon Redshift	字元轉換長度 — 已更新字元類型的 PL/pgSQL 轉換。
Greenplum	Amazon Redshift	解決 Greenplum 發佈金鑰選擇的問題，其中資料表具有分發金鑰，但 AWS SCT 無法將資料表識別為隨機分配並擷取資料表。
Teradata	Amazon Redshift	Teradata 游標支援 — 新增了對游標轉換的支援。
Teradata	Amazon Redshift	[識別] 欄 — 已新增對 [識別] 欄轉換的支援。
Teradata	Amazon Redshift	間隔資料類型 — 已新增對間隔資料類型轉換的支援。

已解決的問題：

- 一般改善
- Greenplum：由於日誌中的錯誤 [已解決] 而無法運行轉換。
- MSSQL — PostgreSQL：轉換 LAG 函數時發生變壓器錯誤 [已解決]。
- MSSQL-後：範圍身份 [已解決]。

- AWS SCT 掛在 DW 項目中 [已解決]。
- 需要映射規則來刪除 AWS SCT [已解決] 中列名稱上的其他空格。

AWS SCT 組建 645 的發行公告

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	提供解決 Teradata 非完全限定視圖 (視圖名稱或視圖中的非完全限定對象) 的解決方案。
Teradata	Amazon Redshift	增加了對 ASCII 函數的支持來計算節點。
Teradata	Amazon Redshift	當在 CHAR 定義為的 Teradata 中發現多 AWS SCT 位元組資料時 CHAR(N) , 該資料會 VARCHAR(3*N) 在 Amazon Redshift 中轉換為。
Teradata	Amazon Redshift	提供日期和數字之間的 Teradata CAST 轉換。 <ul style="list-style-type: none"> • <code>SELECT Cast('2020-07-17' AS BIGINT)</code> • <code>SELECT Cast(20200630 - 19000000 AS DATE)</code>
Teradata	Amazon Redshift	Support 將 Teradata 資 PERIOD 料類型轉換為兩個 Amazon Redshift TIMESTAMP 資料行 : <ul style="list-style-type: none"> • <code>PERIOD(TIMESTAMP)</code> • <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code>
Teradata	Amazon Redshift	Support 帶 RESET WHEN 子句的 Teradata RANK 函數的轉換。
Teradata	Amazon Redshift	改善顯式資料類型轉換中 CAST 的支援 , 以及運算式上隱含的 CAST。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	報告不支持的相關子查詢模式。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的 相關子查詢 。
無	Amazon Redshift	改善表格限制對 RA3 節點類型的支援。
Teradata	Amazon Redshift	加入了對 Teradata 空間資料萃取的支援。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的在 Amazon Redshift 中查詢空間資料 。
Microsoft SQL Server	PostgreSQL	添加了選項， <code>convert_procedures_to_function</code> 。

已解決的問題：

- 一般改善

AWS SCT 組建 644 版的發行公告

1.0.643 AWS SCT 版本的變更會合併到 1.0.644 版本中 AWS SCT。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	<p>多種轉換改進。</p> <ul style="list-style-type: none"> • 使QUALIFY用資料表別名改善轉換。 • 改善IN運算子的轉換。 • 改進了LIKE運算符的轉換。 • 改進了轉換後的代碼中突出顯示問題的轉換 • 改進了 SQL 中QUALIFY條款的不尋常順序的WHERE轉換。

來源	目標	最新消息、增強功能或修正錯誤
		<p>最新消息、增強功能或修正錯誤</p> <ul style="list-style-type: none"> • 修正程序JOIN()UPD_FT_SVC_TRANS_BH_CBH_IND 建構轉換期間發生的變壓器錯誤。 • 已改善巨集到預存程序的轉換。 <p>已新增特殊 AWS SCT CLI 命令，可剖析提供的 sql/bteq 指令碼，並產生原始程式碼中所遇到之語法結構數目的報告。</p> <ul style="list-style-type: none"> • BTEQ 命令計數 • 處理程式計數 • 演員案件數 • DML/DDI 案例計數 • 可更新檢視上的 DML 計數 <p>已新增評估報告動作項目：Amazon Redshift 不支援具有自訂日期格式的 Teradata 資料欄。</p>
Oracle	PostgreSQL/Aurora PostgreSQL	<p>添加了保存擴展包安裝腳本的功能。</p> <p>已變更 AI 5334 的嚴重性等級。</p> <p>改善使用記錄做為封裝變數的效能IMPLEMENTATION 。</p> <p>新增XMLAGG聚合函數支援</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	<p>新增儲存物件實作上註解的載入與轉換。</p>

來源	目標	最新消息、增強功能或修正錯誤
女士 SQL DW	Amazon Redshift	轉換改善：已解決與PATINDEX。 用戶界面改進： <ul style="list-style-type: none"> • 另存為 SQL 以實現源代碼樹。 • 為多個文件的腳本生成添加了額外的邏輯。
Vertica	Amazon Redshift	UI 改進：另存為 SQL 以實現源代碼樹。

已解決的問題：

- Teradata 和 Amazon Redshift 之間轉換的一般改進
- 一般錯誤修復和 UI 改進

組建 642 AWS SCT 版的發行公告

1.0.642 AWS Schema Conversion Tool 版本的變更內容。

Note

AWS Schema Conversion Tool (AWS SCT) 建立 1.0.642 變更適用於視窗、Ubuntu 和軟呢帽。沒有適用於 macOS 的 1.0.642 版本。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft 社會服務	AWS Glue	實現 Microsoft SQL 服務器集成服務 (SSIS) ETL 包轉換為 AWS Glue如需詳細資訊，請參閱 將 SSIS 轉換為AWS Glue與AWS SCT 。
Oracle	瑪麗亞德/SQL 模式 = Oracle /	在 WITH 子句中實作 PL/SQL 宣告區段。

來源	目標	最新消息、增強功能或修正錯誤
	神秘/ Amazon Aurora MySQL	
Oracle	PostgreSQL/Aurora PostgreSQL	新增對 DBMS_SESSION.RESET_PACKAGE 和 DBMS_SESSION.MODIFY_PACKAGE 的支援。
Vertica	Amazon Redshift	允許將 SQL 腳本從 Vertica 數據庫導出到 Amazon Redshift。

已解決的問題：

- 評估報告增強功能。
- 評估報告 UI 增強功能。
- 新增從使用者介面變更 JVM 設定的功能。
- 一般改善。

版本 641 的 AWS SCT 發行公告

1.0.641 AWS Schema Conversion Tool 版本的變更內容。

Note

AWS Schema Conversion Tool (AWS SCT) 建立 1.0.641 變更適用於視窗、Ubuntu 和軟呢帽。沒有適用於 macOS 的 1.0.641 版本。

來源	目標	最新消息、增強功能或修正錯誤
Oracle / 小姐SQL/My	全部	在 .csv 檔案中產生時間報告計算。

來源	目標	最新消息、增強功能或修正錯誤
SQL/ 後期/ DB2 LUW		
Teradata	Amazon Redshift	增加了對 CSUM 功能的支持。 已新增對 Teradata 地理空間資料類型的支援。
Teradata	全部	增加了對轉換身份列的支持。
Greenplum	Amazon Redshift	在 Greenplum 表轉換過程中添加了對分佈樣式 AUTO 的支持。
SAP ASE	全部	在 .csv 檔案中產生時間報告計算。

已解決：

- 各種錯誤修正。
- 各種性能改進。

AWS SCT 組建 640 的發行公告

AWS SCT 版本 1.0.633、1.0.634、1.0.635、1.0.636、1.0.636、1.0.637、1.0.638、和 1.0.640 版的變更會合併到 1.0.640 版本中。AWS SCT

Note

AWS SCT 版本 1.0.640 更改適用於視窗，Ubuntu 和軟呢帽。它們不適用於 macOS。您無法在蘋果 macOS 上安裝 1.0.640 或更高 AWS SCT 版本。AWS SCT 版本 1.0.632 是最後一個版本，以支持蘋果 macOS 上的安裝。

在下表中，您可以找到已合併到版本 1.0.640 的 AWS Schema Conversion Tool 版本的功能和錯誤修正的清單。這些資料表會將來源引擎的功能和錯誤修正分組在一起。

主題

- [發行版本 1.0.640 甲骨文變更](#)

- [版本 1.0.640 Microsoft SQL 伺服器變更](#)
- [發行版本 1.0.640 的變更](#)
- [版本 1.0.640 PostgreSQL 更](#)
- [版本 1.0.640 版本變更](#)
- [版本 1.0.640 太级数据变更](#)
- [適用於其他引擎的發行版本 1.0.640 變更](#)

發行版本 1.0.640 甲骨文變更

下表列出建置 1.0.640 的變更，其中 Oracle 是來源引擎。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL Aurora PostgreSQL	在 Java 和 Pro*C 應用程式中實現了 SQL 代碼轉換。
Oracle	PostgreSQL Aurora PostgreSQL	改善在 WHERE 子句中使用下列函數的效能： <ul style="list-style-type: none"> • 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	新增 RDS MariaDB 10.4 支援所有線上交易處理 (OLTP) 廠商。
Oracle	PostgreSQL/Aurora	已新增對 DBMS_UTILITY.GET_TIME 的支援。 新增了以下模擬：

來源	目標	最新消息、增強功能或修正錯誤
	PostgreSQL	<ul style="list-style-type: none"> DBMS_UTILITY.GET_TIME DBMS_UTILITY.FORMAT_CALL_STACK DBMS_UTILITY.CURRENT_INSTANCE
Oracle	瑪麗亞德/MySQL/Aurora 的 MySQL/Microsoft SQL 服務器模式 = Oracle 後/Aurora 後/RDS 甲骨文	新增對 TABLE(DATA,EXTENDED DATA)、VIEW(DATA,EXTENDED DATA) 和 SEQUENCE(DATA) 的支援
Oracle	PostgreSQL/Aurora PostgreSQL/Oracle RDS	<p>資料欄的 DEFAULT 定義可以延伸至將 DEFAULT 套用至明確 NULL 插入。</p> <p>DEFAULT 子句有一個新的 ON NULL 子句。當 INSERT 陳述式嘗試指派評定為 NULL 時，這個新子句會指示資料庫指派特定預設的資料欄值。</p>
Oracle	MariaDB/MariaDB (SQL MODE=ORACLE)	新增對「身分資料欄」的支援，它會在插入時自動增加。
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11? 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	MariaDB 10.2/MariaDB 10.3/ MySQL/ Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	DEFAULT 子句有一個新的 ON NULL 子句，它指示當 INSERT 陳述式嘗試指定一個值時，資料庫指定一個指定的預設資料欄值 NULL。
Oracle	Oracle RDS/ MySQL /Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	新增對 IDENTITY 資料欄的支援。
Oracle	MySQL 8.x	新增對 CHECK 限制條件的支援。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL/Aurora PostgreSQL	<p>使用擴展包例程實現了檢查 ANYDATA 為空/不為空。</p> <p>根據 XMLSequence 的 TABLE 函數實作查詢中使用的 VALUE 函數的模擬。</p> <p>新增 DBMS_LOB 對下列內建常式的支援：</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
全部	SQL Server	<p>SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。</p> <p>SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。</p> <p>SQL Server 2016：新增 TEMPORAL TABLES 的支援。</p>
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。
Oracle	Amazon Redshift	<p>在巢狀區塊中實作游標屬性的轉換。</p> <p>Amazon Redshift 不支持集合。相關變量會轉換為 VARCHAR。除了將一個變數指定給另一個變數以外的所有收集作業都會遭到拒絕，包括初始化與收集要素存取權。</p> <p>實施 Amazon Redshift 分佈風格 = 自動。</p>

來源	目標	最新消息、增強功能或修正錯誤
Oracle	PostgreSQL/Aurora PostgreSQL	<p>如果 Oracle 中的非預留字詞在 PostgreSQL 預留，則會有以下情況：</p> <ul style="list-style-type: none"> • 如果該字詞有引號，系統會保留其大小寫，並保留引號。 • 如果字詞沒有引號的，系統會將其轉換為大寫並附上引號。 <p>實作使用函數以將其作為輸入 LTRIM、RTRIM 和 TRIM 函數的能力。</p> <p>SELECT DISTINCT、ORDER BY 表達式必須出現在選擇清單中。</p> <p>對於在具有 DEFAULT 值之後的參數之後的游標參數，會 AWS SCT 增加 DEFAULT 為 NULL 子句</p> <p>Source OUT 游標參數會轉換為 IN 游標參數。</p> <p>透過在「轉換設定」下新增「套件變數邏輯實作」選項來重新實現套件變數。可用的設定為：「工作階段變數」和「plv8 全域物件」。預設值是「工作階段變數」。</p> <p>實作具有 dblink 和 pg_background 的 AUTONOMOUS_TRANSACTION 支援。</p>
Oracle	全部	實作檢視 SYS_%_TAB_COMMENTS。
Oracle	PostgreSQL	篩選器的可變輸入不 PostgreSQL。從 Oracle 轉換為 PostgreSQL 時，如果遇到變數篩選條件，現在會報告例外狀況。

來源	目標	最新消息、增強功能或修正錯誤
Oracle	Amazon Redshift	<p>實作儲存的程式碼 FOR..LOOP 游標轉換的改進。</p> <p>使用預設參數實現函數/程序的儲存程式碼叫用。</p> <p>實現儲存的程式碼能力，使用別名更新沒有 WHERE 子句。</p> <p>實作儲存的程式碼函數使用 SELECT FROM 雙重函數預先執行的其他案例。</p> <p>實作儲存的程式碼 Table%ROWTYPE 參數和套件變數。</p> <p>實作使用 JAVA 和外部程序的儲存程式碼。</p> <p>在儲存的程式碼中實作標準的 Oracle 套件。</p>

版本 1.0.640 Microsoft SQL 伺服器變更

下表列出建置 1.0.640 的變更，其中 Microsoft SQL Server 是來源引擎。

來源	目標	最新消息、增強功能或修正錯誤
Microsoft Azure/ Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL/MySQL/ Aurora MySQL	增加了對 COLUMN STORE 索引的支援。
Microsoft SQL Server	RDS MariaDB 10.4	新增 RDS MariaDB 10.4 支援所有線上交易處理 (OLTP) 廠商。
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/	新增對 OPTIMIZE_FOR_SEQUENTIAL_KEY 屬性的支援。

來源	目標	最新消息、增強功能或修正錯誤
	Pos tgreSQL/ Aurora PostgreSQL	
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQL	新增對資料庫節點和 Edge 資料表類型的支援。
Azure/SQL Server	MariaDB/M ySQL/Auro ra MySQL/ Pos tgreSQL/ Aurora PostgreSQL	新增對 TEMPORAL TABLES 的支援。
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11? 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。

來源	目標	最新消息、增強功能或修正錯誤
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQL/ MariaDB	新增對 SQL Server 圖形架構的 DML 處理支援。
SQL Server	Aurora PostgreSQL	新增轉換參數沒有 par_ 字首的選項。
Azure/SQL Server	MySQL 8.x	新增對 CHECK 限制條件的支援。
全部	SQL Server	<p>SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。</p> <p>SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。</p> <p>SQL Server 2016：新增 TEMPORAL TABLES 的支援。</p>
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。
SQL Server	AWS Glue (Python 殼)	<p>轉換改進功能，包括：</p> <ul style="list-style-type: none"> 實作內建函數轉換為 Python.String。 在儲存程式碼中實作 EXECUTE 和 EXEC。 使用資料表類型實作。

來源	目標	最新消息、增強功能或修正錯誤
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	實作使 \$TMP 程序變為選用。
SQL Server	MySQL/Aurora MySQL	<p>具有日期的延伸算術運算。</p> <p>建構模擬 'TOP (expression) WITH TIES。</p> <p>現在叫出產生的參考游標後，參考游標便會關閉。</p> <p>在 Aurora MySQL 中不支援設定 GLOBAL 隔離等級。您只能變更工作階段範圍。交易的預設行為是使用 REPEATABLE READ 和一致性讀取。專為與 READ COMMITTED 搭配使用而設計的應用程式可能需要修改。或者，他們可以明確地將預設值變更為 READ COMMITTED。</p>
SQL Server	AWS Glue (Python 殼)	<p>SQL Server 陳述式會產生完整的結果集，但有時候最好一次處理一個資料列的結果。在結果集上開啟游標可讓您一次處理一列結果集。您可以將游標指派給具有游標資料類型的變數或參數。</p> <p>實作為預存程式碼封閉一系列的 Transact-SQL 陳述式，如此一來，即使 Python 不支援 SQL 伺服器的開始和結束，仍可以執行一組 Transact-SQL 陳述式。control-of-flow</p> <p>SQL 伺服器標籤和 GOTO 陳述式不受支援 AWS Glue。如果在程式碼中 AWS SCT 遇到標籤，則會略過該標籤。如果 AWS SCT 遇到一個 GOTO 陳述式，系統會加上註解。</p>

來源	目標	最新消息、增強功能或修正錯誤
SQL Server	Amazon Redshift	<p>透過實作 IF ... ELSE 控制實作儲存程式碼的 Transact-SQL 語句的條件處理。</p> <p>針對預存程式碼實作封閉一系列 Transact-SQL 陳述式，如此一組 Transact-SQL 陳述式就可以當做區塊執行。支援巢狀 BEGIN... END 區塊。</p> <p>在儲存程式碼中實作 SET 和 SELECT。</p> <p>透過在資料表上建立使用者指定的排序索引鍵，在 Amazon Redshift 中實作建立索引鍵 (不支援索引)。</p>

發行版本 1.0.640 的變更

下表列出建置 1.0.640 的變更，其中 MySQL 是來源引擎。

來源	目標	最新消息、增強功能或修正錯誤
MySQL	PostgreSQL 12.x	新增對已產生資料欄的支援。
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11? 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。
MySQL	PostgreSQL/Aurora PostgreSQL 11.	<p>已新增下列項目的支援：</p> <ul style="list-style-type: none"> • SQL 儲存程序中的內嵌交易。 • CALL SQL 儲存程序的能力。 • 建立 SQL 儲存程序的能力。
全部	SQL Server	

來源	目標	最新消息、增強功能或修正錯誤
		<p>SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。</p> <p>SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。</p> <p>SQL Server 2016：新增 TEMPORAL TABLES 的支援。</p>
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。

版本 1.0.640 PostgreSQL 更

下表列出了建置 1.0.640 的變更，其中 PostgreSQL 是來源引擎。

來源	目標	最新消息、增強功能或修正錯誤
PostgreSQL	MySQL 8.x	<p>MySQL 現在支援建立索引運算式而非欄位值的功能索引鍵部分。函數金鑰部分可索引無法以其他方式索引的值，例如 JSON 值。</p> <p>MySQL 現在支援 CTE 和 Recursive CTE。</p>
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11? 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>已新增下列項目的支援：</p> <ul style="list-style-type: none"> • SQL 儲存程序中的內嵌交易。 • CALL SQL 儲存程序的能力。 • 建立 SQL 儲存程序的能力。
PostgreSQL	MySQL 8.x	新增 MySQL 對降序索引的支援。不再忽略索引定義中的 DESC，但會導致儲存金鑰值按降序排列。

來源	目標	最新消息、增強功能或修正錯誤
		<p>已新增 MySQL 支援使用表達式作為資料類型規格中的預設值，包括將表達式作為 BLOB、TEXT、幾何和 JSON 資料類型的預設值。</p> <p>現在現有彙總函數中的有些函數可做為視窗函數。</p> <ul style="list-style-type: none"> • AVG() • BIT_AND() • BIT_OR() • BIT_XOR() • COUNT() • JSON_ARRAYAGG() • JSON_OBJECTAGG() • MAX() • MIN() • STDDEV_POP() • STDDEV() • STD() • STDDEV_SAMP() • SUM() • VAR_POP() • VARIANCE() • VAR_SAMP() <p>MySQL 支援視窗函數，也就是在處理每個資料列時，都使用與該資料列相關的資料列進行查詢，以及執行計算。</p> <ul style="list-style-type: none"> • CUME_DIST() • DENSE_RANK() • FIRST_VALUE() • LAG() • LAST_VALUE()

來源	目標	最新消息、增強功能或修正錯誤
		<ul style="list-style-type: none"> • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK() • ROW_NUMBER()
PostgreSQL	MySQL 8.x	新增對 CHECK 限制條件的支援。
全部	SQL Server	<p>SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。</p> <p>SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。</p> <p>SQL Server 2016：新增 TEMPORAL TABLES 的支援。</p>
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。
PostgreSQL/Aurora PostgreSQL	全部	<p>新增系統檢視系統索引模擬。</p> <p>如果程序中沒有指定 INTO 的 SELECT 陳述式，則會針對目標上的程序建立類型參考游標的 INOUT p_refcur 參數。</p>

版本 1.0.640 版本變更

下表列出建置 1.0.640 的變更，其中 DB2 LUW 是來源引擎。

來源	目標	最新消息、增強功能或修正錯誤
DB2 LUW	RDS MariaDB 10.4	新增 RDS MariaDB 10.4 支援所有線上交易處理 (OLTP) 廠商。
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11? 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。
DB2 LUW	MySQL 8.0.17	新增 CHECK 限制條件的支援。
全部	SQL Server	SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。 SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。 SQL Server 2016：新增 TEMPORAL TABLES 的支援。
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。

版本 1.0.640 太级数据变更

下表列出了生成 1.0.640 更改太瑞數據源引擎。

來源	目標	最新消息、增強功能或修正錯誤
Teradata	Amazon Redshift	新增 MERGE 和 QUALIFY 陳述式的支援。 從 Terdata 陳述式移除 LOCKING ROWS FOR ACCESS 子句。 新增對 CAST 函數的支援。

來源	目標	最新消息、增強功能或修正錯誤
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11？ 在 Amazon Corretto 11 用戶指南。
Teradata	Teradata	實作 REGEXP_INSTR() 和 REGEXP_SUBSTR() 中的改進。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。
全部	SQL Server	SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。 SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。 SQL Server 2016：新增 TEMPORAL TABLES 的支援。
Teradata	全部	新增了 REGEXP_INSTR() 和 REGEXP_SUBSTR() 的支援。
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。
Teradata	Amazon Redshift	使用「專案設定」、「另存為 SQL」和「套用」、「下拉式清單：單一檔案/多個檔案」中的設定，實作將來源樹狀結構的 SQL 儲存為單一檔案或多個檔案的函數。 實作檢視和程序轉換的改進。
Teradata	全部	新增對 Terdata 版本 16.20 的支援

適用於其他引擎的發行版本 1.0.640 變更

下表列出建置 1.0.640 對其他來源引擎的變更。

來源	目標	最新消息、增強功能或修正錯誤
Sybase	RDS MariaDB 10.4	新增 RDS MariaDB 10.4 支援所有線上交易處理 (OLTP) 廠商。
SAP ASE	MariaDB	實作下列項目： <ul style="list-style-type: none"> • MariaDB 10.4 • EXECUTE IMMEDIATE 陳述式 • DEFAULT 定義 • CHECK 限制條件的支援
SAP ASE	PostgreSQL 12.x	新增對已產生資料欄的支援。
全部	全部	從 JDK 8 升級到 Amazon Corretto JDK 11。如需詳細資訊，包括下載連結，請參閱 什麼是 Amazon Corretto 11？ 在 Amazon Corretto 11 用戶指南。
全部	全部	在評定報告中新增關於使用者資料庫中可能不一致的資訊。
SAP ASE	MySQL 8.0.17	新增 CHECK 限制條件的支援。
全部	SQL Server	SQL Server 2019：新增對新索引屬性 OPTIMIZE_FOR_SEQUENTIAL_KEY 的支援。 SQL Server 2017：新增支援圖形資料庫節點和 Edge 資料表類型。 SQL Server 2016：新增 TEMPORAL TABLES 的支援。
Vertica	Amazon Redshift	新增了對分佈樣式 = AUTO 的支援。
全部	全部	實作了使用虛擬分割區覆寫實體分割區的功能。資料倉儲提取器根據建立的虛擬分割區提取資料。

來源	目標	最新消息、增強功能或修正錯誤
Amazon Redshift	Amazon Redshift	DML 陳述式中不支援的內建函數會以 NULL 取代為預留位置。
Sybase	PostgreSQL	新增對原生函數的支援。
SAP ASE	MySQL/Aurora MySQL	Aurora MySQL 的預設隔離等級為 REPEATABLE READ。在 Aurora MySQL 中不支援設定 GLOBAL 隔離等級。您只可變更任務階段範圍。交易的預設行為是使用 REPEATABLE READ 和一致性讀取。設計為使用 READ COMMITTED 執行的應用程式可能需要修改。或者，您可以明確地將預設值更改為 READ COMMITTED。
SAP ASE	PostgreSQL	添加了對沒有擴展包的 CONVERT 功能 (樂觀) 的支持。
SAP ASE	全部	<p>新增系統檢視系統索引模擬。</p> <p>如果程序中沒有指定 INTO 的 SELECT 陳述式，則會針對目標上的程序建立類型參考游標的 INOUT p_refcur 參數。</p>
Greenplum	Amazon Redshift	實作 CREATE TEMPORARY TABLE，如下所示：

文件歷史紀錄

下表說明 2018 年 1 月之後對 AWS Schema Conversion Tool (AWS SCT) 使用者指南進行的重要變更。

您可以訂閱 RSS 摘要，以取得此文件更新的通知。

變更	描述	日期
AWS SCT 建立 #1	構建 1.0.672 提供支持亞馬遜 RDS 作為目標和微軟 SQL 服務器 2022 版作為源。它也會在轉換後的程式碼中新增對 Amazon Redshift 功能的支援、針對 IBM Db2 for z/OS 來源實作多項轉換改進功能，並解決許多轉換問題。	2023 年 5 月 8 日
AWS SCT 建立 #1	構建 1.0.671 提供了從阿帕奇奧西遷移到 AWS Step Functions 它還增加了 BigQuery 作為多伺服器評估程序來源的支援。此外，它還新增 IBM Db2 for z/OS 的新轉換設定做為來源，並解決了許多轉換問題。	2023 年 3 月 8 日
AWS SCT 建立 #1	建置 1.0.670 提供從 Hadoop 遷移到亞馬遜 EMR 的支援。它還添加了對 Azure Synapse 分析的支持，作為多伺服器評估過程的來源。此外，它還改善了內嵌在 Java 應用程式中的 SQL 程式碼的轉換，並解決了許多轉換問題。	2023 年 1 月 23 日

AWS SCT建立 #1	建置 1.0.669 實作原生磁碟分割的支援，以便從 Oracle 資料倉儲進行資料移轉。它也改善了多伺服器評估程序、在資料擷取代理程式中新增功能，並解決了許多轉換問題。	2022 年 12 月 19 日
AWS SCT建立 #1	Build 1.0.668 為從 Greenplum 資料庫進行資料遷移實作自動虛擬磁碟分割，並新增將資料從雪花資料庫遷移到亞馬遜 Redshift 的支援。它也改善了內嵌在 C# 應用程式中的 SQL 程式碼的轉換，並解決了許多轉換問題。	2022 年 11 月 16 日
AWS SCT建立 #1	建置 1.0.667 提供資訊擷取、轉換和載入 (ETL) 引擎做為移轉來源的支援。它也會更新擴充套件版本、增加 Amazon Redshift 支援的最低驅動程式版本，並解決許多轉換問題。	2022 年 10 月 13 日
AWS SCT建立 #1	構建 1.0.666 通過添加對框架的支持來改善 Java 應用程序的轉換。MyBatis 它還將新功能添加到擴展包中，增強了源元數據加載器，並解決了許多轉換問題。	2022 年 9 月 20 日
AWS SCT建立 #1	建置 1.0.665 提供作為移轉來源 BigQuery 的支援。它還實現了對 Babelfish 功能配置文件的新版本的支持。此外，它還改善了將資料倉儲轉換為 Amazon Redshift 的功能，並解決了許多轉換問題。	2022 年 8 月 29 日

[AWS SCT建立 #1](#)

建置 1.0.664 提供 Amazon Redshift 無伺服器作為遷移來源或目標的支援。它還在數據提取任務中實現自動內存平衡，並修復了AWS SCT無法連接到AWS Snowball設備的錯誤。此外，它還新增了移轉規則中變更資料行定序的功能、改善使用者介面，以及解決許多轉換問題。

2022 年 7 月 14 日

[AWS SCT建立 #1](#)

建置 1.0.663 新增對極光 PostgreSQL 1.2.0 的巴貝爾魚的支援，並改善了多伺服器評估報告功能。它還在遷移規則中添加了新功能，修復了兩個加載器錯誤，並解決了許多轉換問題。

2022 年 6 月 20 日

[AWS SCT建立 #1](#)

建置 1.0.662 會在 C# 應用程式中實作 SQL 程式碼轉換，並改善多伺服器評估報告工作流程。它還增加了多個轉換改進功能，並解決了許多轉換問題。

2022 年 5 月 19 日

[AWS SCT建立 #1](#)

建置 1.0.661 可提供 IBM Db2 (適用於 z/OS) 的支援，做為移轉來源。它還增加了對轉換擷取、轉換和載入 (ETL) 指令碼的支援，AWS Glue Studio並解決了許多轉換問題。

2022 年 4 月 21 日

AWS SCT建立 #1	建置 1.0.660 提供了對 PostgreSQL 主要版本 14 和 MariaDB 10.6 作為遷移目標的支援。它還增加了對具體化視觀表轉換 Oracle 索引的支援，並解決了許多轉換問題。	2022 年 3 月 21 日
AWS SCT建立 #1	建置 1.0.659 可在極光與 PostgreSQL 相容版本上提供 PostgreSQL 主要版本 13 的支援，做為移轉目標。它在 C# 應用程式中實作 SQL 程式碼轉換、新增 Oracle 統一稽核的支援，以及解決許多轉換問題。	2022 年 2 月 21 日
AWS SCT建立 #1	建置 1.0.658 可整合 AWS Secrets Manager 並增加對亞馬遜 Redshift 虛擬目標資料庫平台的支援。它還添加了許多轉換改進和錯誤修復。	2022 年 1 月 20 日
AWS SCT建立 #1	建置 1.0.657 改善了從微軟 SQL 伺服器轉換為極光兼容版本、亞馬遜 RDS 對於 PostgreSQL 和其他遷移目的地的轉換。它還增加了許多用戶界面改進和錯誤修復。	2021 年 12 月 20 日
AWS SCT建立 #1 .0.656	構建 1.0.656 在一個項目中提供了多個源和目標數據庫的支援。它還添加了轉換，優化策略和一般改進以及一些錯誤修復。	2021 年 11 月 22 日

AWS SCT建立 #1	建置 1.0.655 實作 Teradata FastExport 工作指令碼轉換為亞馬遜紅移 RSQL，並將綠色梅林的最低支援驅動程式版本增加到 42.2.5。它還增加了許多改進和錯誤修復。	2021年10月18日
AWS SCT建立 #1	建置 1.0.654 實作命令介面、太資料和太資料基本太資料 FastLoad查詢 (BTEQ) 指令碼轉換為亞馬遜紅移 RSQL。它還解決了許多轉換問題，並添加了許多改進和錯誤修復。	2021 年 9 月 16 日
AWS SCT建立 #1	構建 1.0.653 實現在被調用的函數或過程中創建的動態 SQL 的轉換。它還改善了加密常式的轉換，並增加了許多改進和錯誤修復。	2021 年 8 月 10 日
AWS SCT建立 #1	Build 1.0.652 在命令行界面中實現腳本命令模式，並實現模式優化規則。它還添加了許多轉換和性能改進以及錯誤修復。	2021 年 6 月 30 日
AWS SCT建立 #1	版本 1.0.651 增加了許多改進和錯誤修復。它還提供對 AWS Schema Conversion ToolCLI 參考的初始副本的訪問。	2021 年 6 月 4 日

AWS SCT建立 #1	建置 1.0.650 將 Amazon RDS 支援實作為目標資料庫，並更新擷取程式代理程式。它還升級微軟 SQL 服務器，Azure 和 Azure 突觸的最低支持 JDBC 驅動程序版本。此外，它還添加了許多轉換改進和錯誤修復。	2021 年 4 月 30 日
AWS SCT建立 #1	構建 1.0.649 將 MariaDB 10.5 的支持實現為目標數據庫，並實現了轉換甲骨文內置函數的功能增強功能。它還添加了許多轉換和性能改進以及錯誤修復。	2021 年 3 月 29 日
AWS SCT建立 #1	版本 1.0.648 增加了許多轉換改進和錯誤修復。	2021 年 2 月 22 日
AWS SCT建立 #1	建置 1.0.647 新增 Amazon RDS 上資料庫郵件功能的支援，在儲存物件上實作註解的載入和轉換。它也會新增AWS SCT資料移轉服務評估員和 AWS SCT精靈，並實作樹狀結構篩選器使用者介面。此外，它還在評估報告中添加了一個重新設計的部分，以及一些改進和錯誤修復。	2021 年 1 月 15 日
AWS SCT建立 #1	Build 1.0.646 新增對間隔資料類型、識別資料欄和游標轉換的支援，並新增了許多改進和錯誤修正。	2020 年 12 月 28 日

AWS SCT建立 #1	版本 1.0.645 增加了對 ETL SSIS AWS Glue 轉換的支持，以及一些改進和錯誤修復。	2020 年 11 月 16 日
AWS SCT建置 #1 .643-1.0.644	Build 1.0.644 新增了許多轉換、效能和使用者介面改進和錯誤修正。	2020年10月14日
AWS SCT建立 #1	構建 1.0.642 實現 ETL 軟件包從微軟 SQL 服務器集成服務轉換，AWS Glue並添加了一些改進和錯誤修復。	2020 年 8 月 28 日
AWS SCT建立 #1	添加了對數據提取器的 SSL 支持。組建還包括一些改進和修復。	2020 年 7 月 17 日
AWS SCT組建 #1	從 JDK 8 升級到 Amazon Corretto JDK 11。新增辨別其他升級、變更和修正的資料表。	2020 年 6 月 22 日
AWS WQF 可用性	AWS SCT 不再提供 AWS Workload Qualification Framework (AWS WQF) 工具的下載。	2020 年 6 月 19 日
AWS SCT組建 #1 .0.632	SCT UI - 新增標籤來顯示在套用指令碼時所發生的錯誤。您現在可以在從 SAP ASE 轉換時將來源樹狀結構另存為 SQL。轉換至 PostgreSQL、Aurora PostgreSQL 或 Redshift 的效能提升。	2019 年 11 月 19 日

[AWS SCT建構 #1 .0.631 和 \(合併\) #1](#)

在甲骨文和微軟 SQL 服務器和 SAP ASE 系統對象更好地支持 ROWID。以更好的方式處理 SQL Server 結構描述遺失的指標。對從 Greenplum 到 Redshift 的轉換提供更好的支援。改進了在移動到亞馬遜紅移、瑪麗亞德、MySQL 和 PostgreSQL 時對存儲代碼轉換的支援。

2019 年 9 月 30 日

[AWS SCT建立 #1](#)

支援從 Netezza 轉換的預存程序。改進了對亞馬遜紅移、動態 B、MySQL 和 PostgreSQL 的轉換支援。新增將 SAP ASE 12.5 做為來源的支援

2019 年 8 月 20 日

[AWS SCT建立 #1](#)

支援從 DB2、SQL Server 和 Oracle 轉換的服務模擬。Amazon Redshift 轉換的增強功能，包括對游標和預存程序的更多支援。

2019 年 6 月 22 日

[AWS SCT建立 #1](#)

支援從 SQL 伺服器轉換為亞馬遜紅移中的預存程序。轉換為 PostgreSQL 11 和 MySQL 8.0 的增強功能。

2019 年 5 月 31 日

[AWS SCT建立 #1](#)

現在支援以目標形式支援 PostgreSQL 11 和 MySQL 8.0。現在支援 SAP ASE 15.5 做為來源。

2019 年 4 月 26 日

AWS SCT建立 #1	更新包括將 Teradata BTEQ 轉換為的能力AWS Glue、支援使用甲骨文相容模式支援轉換為 MariaDB 10.3、對 SAP ASE 15.7 的支援，以及模擬缺少功能的服務替換。	2019 年 3 月 25 日
AWS SCT建立 #1	更新包括將甲骨文 ETL 轉換為的能力AWS Glue，以及從微軟 SQL 服務器，甲骨文和 IBM Db2 LUW 轉換為亞馬遜 RDS 的 MariaDB 的支持。我們也新增使用 MySQL 相容性從 SAP ASE 轉換到 RDS for MySQL 和 Amazon Aurora 的支援。此外，我們新增在 Oracle 轉換到 PostgreSQL 時對 Oracle 延伸的支援。	2019 年 2 月 22 日
AWS SCT建立 #1	更新包含以下能力：轉換 SAP ASE 資料庫、讓客戶能夠將 T-SQL 指令碼、DML 和 DDL 轉換為對等程式碼或元件。我們也新增 Oracle 和 Microsoft SQL Server 模擬功能，以改善轉換。	2019 年 1 月 25 日
AWS SCT建立 #1	更新包含工作負載資格架構，可分析整個遷移的工作負載 (包括資料庫和應用程式修改)。	2018 年 12 月 20 日
AWS SCT建置 #1	更新包括支援 Aurora PostgreSQL 10 做為目標，以即能夠使用外部表格選項從 Netezza 遷移。	二〇一八年十一月二

AWS SCT建立 #1	更新包括能夠儲存 SQL 指令碼，以及在遷移至 MySQL 時支援 Oracle 全域游標。	2018 年 10 月 22 日
AWS SCT建立 #1	更新包括支援從 Apache Cassandra 遷移至 DynamoDB，以及支援將 Vertica 9 做為來源。	2018 年 9 月 20 日
AWS SCT建立 #1	更新包括擴展的評估報告、Oracle ROWID 轉換支援，以及 SQL Server 使用者定義資料表支援。	2018 年 8 月 24 日
AWS SCT建立 #1	更新包括擴展的評估報告、Oracle ROWID 轉換支援，以及 SQL Server 使用者定義資料表支援。	2018 年 7 月 24 日
AWS SCT建立 #1 .0.616	更新包括支援 RDS 從 Oracle 轉換到 Amazon RDS for Oracle、轉換 Oracle 排程物件，以及支援 Oracle 任務、分割和 Db2 LUW 10.1 版。	2018 年 6 月 26 日
AWS SCT建立 #1	更新包括 SQL Server 至 PostgreSQL GOTO 陳述式支援、PostgreSQL 10 分割，和 Db2 LUW 10.1 版。	2018 年 5 月 24 日
AWS SCT建立 #1	更新包括 Oracle 到 Oracle 資料庫連結支援、SQL Server 至 PostgreSQL 內嵌函數，和 Oracle 系統物件模擬。	2018 年 4 月 25 日

AWS SCT建立 #1	更新包括 Db2 LUW 支援、SQL*Plus 檔案轉換，和 SQL Server Windows 身分驗證。	2018 年 3 月 28 日
AWS SCT建立 #1	更新包括自訂資料類型映射支援、Oracle 10 結構描述比較，和全域變數的 Oracle 到 PostgreSQL 轉換。	2018 年 2 月 22 日
AWS SCT建立 #1	更新包括 Oracle 到 PostgreSQL 動態陳述式支援、透過選擇錯誤訊息開啟日誌檔，以及在樹狀檢視隱藏結構描述的能力。	2018 年 1 月 23 日

舊版更新

下表說明 2018 年 1 月之前的 AWS Schema Conversion Tool (AWS SCT) 使用者指南的重要變更。

版本	變更	描述	變更日期
1.0.608	適用於亞馬遜 S3 的 FIPS 端點支援	您現在可以使用 FIPS 端點 AWS SCT 要求連接到 Amazon S3 和 Amazon Redshift，以符合聯邦資訊處理標準安全要求。如需詳細資訊，請參閱 儲存 AWS 認證 。	2017 年 11 月 17 日
1.0.607	適用於亞馬遜 S3 的 FIPS 端點支援	您現在可以使用 FIPS 端點 AWS SCT 要求連接到 Amazon S3 和 Amazon Redshift，以符合聯邦資訊處理標準安全要求。如需詳細資訊，請參閱 儲存 AWS 認證 。	2017 年 10 月 30 日
1.0.607	資料擷取任務可以忽略 LOB	當您建立資料擷取任務，現在可以選擇忽略大型物件 (LOB) 以降低您擷取的資料量。如需詳細資訊，請參閱 建立、執行和監視 AWS SCT 資料萃取工作 。	2017 年 10 月 30 日

版本	變更	描述	變更日期
1.0.605	資料擷取代理程式任務日誌存取	您現在可以從 AWS Schema Conversion Tool 使用者界面的方便連結，存取資料擷取代理程式任務日誌。如需詳細資訊，請參閱 建立、執行和監視 AWS SCT 資料萃取工作 。	2017 年 8 月 28 日
1.0.604	轉換器增強功能	AWS Schema Conversion Tool 引擎已增強，可為異質遷移提供改進的轉換。	2017 年 6 月 24 日
1.0.603	資料擷取代理程式支援篩選條件	您現在可以篩選擷取代理程式從您的資料倉儲擷取的資料。如需詳細資訊，請參閱 在中建立資料移轉規則 AWS SCT 。	2017 年 6 月 16 日
1.0.603	AWS SCT 支援額外的資料倉儲版本	您現在可以使用將 Teradata 13 和甲骨文資料倉儲 10 架構轉換為等效的亞馬遜紅移架構。AWS Schema Conversion Tool 如需詳細資訊，請參閱 使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT 。	2017 年 6 月 16 日
1.0.602	資料擷取代理程式支援額外的資料倉儲	您現在可以使用資料擷取代理程式，從 Microsoft SQL Server 資料倉儲擷取資料。如需詳細資訊，請參閱 將資料從現場部署資料倉儲遷移到 Amazon Redshift 。	2017 年 5 月 11 日
1.0.602	資料擷取代理程式可以將資料複製到亞馬遜紅移	資料擷取代理程式現在有三個上傳模式。您現在可以指定是要擷取資料、擷取資料然後將資料上傳到 Amazon S3，還是直接擷取、上傳和複製資料到 Amazon Redshift。如需詳細資訊，請參閱 建立、執行和監視 AWS SCT 資料萃取工作 。	2017 年 5 月 11 日
1.0.601	AWS SCT 支援額外的資料倉儲	您現在可以使用將您的 AWS Schema Conversion Tool Vertica 和微軟 SQL 伺服器結構描述轉換為等效的亞馬遜紅移架構。如需詳細資訊，請參閱 使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT 。	2017 年 4 月 18 日

版本	變更	描述	變更日期
1.0.601	資料擷取代理程式支援額外的資料倉儲	您現在可以使用資料擷取代理程式，從 Greenplum、Netezza 和 Vertica 資料倉儲擷取資料。如需詳細資訊，請參閱 將資料從現場部署資料倉儲遷移到 Amazon Redshift 。	2017 年 4 月 18 日
1.0.601	資料擷取代理程式支援額外的作業系統	您現在可以在執行 macOS 和 Microsoft Windows 作業系統的電腦上，安裝資料擷取代理程式。如需詳細資訊，請參閱 安裝擷取代理 。	2017 年 4 月 18 日
1.0.601	資料擷取代理程式會自動上傳到 Amazon S3	資料擷取代理程式現在會自動將擷取的資料上傳到 Amazon S3。如需詳細資訊，請參閱 資料萃取工作輸出 。	2017 年 4 月 18 日
1.0.600	資料擷取代理程式	您現在可以安裝從資料倉儲擷取資料的資料擷取代理程式，並準備好與 Amazon Redshift 搭配使用。您可以使用 AWS Schema Conversion Tool 來註冊代理程式並建立資料擷取任務。如需詳細資訊，請參閱 將資料從現場部署資料倉儲遷移到 Amazon Redshift 。	2017 年 2 月 16 日
1.0.600	客戶意見回饋	您現在可以提供有關 AWS Schema Conversion Tool 的意見回饋。您可以提交錯誤報告、可以提交功能請求，也可以提供一般資訊。如需詳細資訊，請參閱 提供意見 。	2017 年 2 月 16 日
1.0.502	與 AWS DMS 的整合	您現在可以使用 AWS Schema Conversion Tool 來建立 AWS DMS 端點和任務。您可以從 AWS SCT 執行和監控任務。如需詳細資訊，請參閱 搭配使用 AWS SCT 與 AWS DMS 。	2016 年 12 月 20 日

版本	變更	描述	變更日期
1.0.502	與 PostgreSQL 相容的 Amazon Aurora 可做為目標資料庫	AWS Schema Conversion Tool 現在支援與 PostgreSQL 相容的 Amazon Aurora 做為目標資料庫。如需詳細資訊，請參閱 轉換資料庫結構描述 AWS SCT 。	2016 年 12 月 20 日
1.0.502	支援設定檔	您現在可以在 AWS Schema Conversion Tool 存放不同的設定檔，並在這些設定檔間輕鬆切換。如需詳細資訊，請參閱 將 AWS 服務設定檔儲存在 AWS SCT 。	2016 年 12 月 20 日
1.0.501	支援 Greenplum 資料庫和 Netezza	您現在可以使用將資料倉儲結構描述從綠色資料庫和 Netezza 轉換為亞馬遜紅移。AWS Schema Conversion Tool 如需詳細資訊，請參閱 使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT 。	2016 年 11 月 17 日
1.0.501	Redshift 最佳化	您現在可以使用 AWS Schema Conversion Tool 來最佳化您的 Amazon 紅移資料庫。如需詳細資訊，請參閱 通過使用優化亞馬遜紅移 AWS SCT 。	2016 年 11 月 17 日
1.0.500	對應規則	使用 AWS Schema Conversion Tool 轉換結構描述之前，您現在可以先設定變更欄的資料類型、將物件從某個結構描述移至另一個結構描述，以及變更物件名稱的規則。如需詳細資訊，請參閱 在中建立移轉規則 AWS SCT 。	2016 年 10 月 4 日
1.0.500	移至雲端	您現在可以使用將現有的現場部署資料庫結構描述複製到執行相同引擎的 Amazon RDS 資料庫執行個體。AWS Schema Conversion Tool 您可以使用此功能分析移到雲端和變更授權類型的潛在成本節省。如需詳細資訊，請參閱 建立移轉評估報告 AWS SCT 。	2016 年 10 月 4 日

版本	變更	描述	變更日期
1.0.400	資料倉儲結構描述轉換	您現在可以使用將您的AWS Schema Conversion Tool資料倉儲結構描述從甲骨文和 Teradata 轉換為亞馬遜紅移。如需詳細資訊，請參閱 使用將資料倉儲結構描述轉換為亞馬遜紅移 AWS SCT 。	2016 年 7 月 13 日
1.0.400	應用程式 SQL 轉換	您現在可以使用 AWS Schema Conversion Tool 來轉換 C++、C#、Java 或其他應用程式程式碼中的 SQL。如需詳細資訊，請參閱 使用轉換應用程式 SQLAWS SCT 。	2016 年 7 月 13 日
1.0.400	新功能	AWS Schema Conversion Tool現在包含擴充套件和精靈，可協助您安裝、建立和設定 AWS Lambda函式和 Python 程式庫，以提供電子郵件、工作排程和其他功能。如需詳細資訊，請參閱 使用 AWS SCT 擴展包中的 AWS Lambda 功能 及 使用 AWS SCT 擴充套件的自訂程式庫 。	2016 年 7 月 13 日
1.0.301	SSL 支援	現在當您使用 AWS Schema Conversion Tool 時，可以使用 Secure Sockets Layer (SSL) 來連接到您的來源資料庫。	2016 年 5 月 19 日
1.0.203	新功能	新增支援將 MySQL 和 PostgreSQL 做為轉換的來源資料庫。	2016 年 4 月 11 日
1.0.202	維護版本	新增支援編輯為目標資料庫引擎產生的轉換後 SQL。新增改進來源資料庫與目標資料庫執行個體樹狀檢視中的選擇功能。新增支援使用透明網路底層 (TNS) 名稱連接至 Oracle 來源資料庫。	2016 年 3 月 2 日
1.0.200	維護版本	新增支援 PostgreSQL 做為目標資料庫引擎。新增在套用結構描述到目標資料庫執行個體之前，能夠產生轉換結構描述的指令碼並將指令碼儲存到檔案。	2016 年 1 月 14 日

版本	變更	描述	變更日期
1.0.103	維護版本	新增離線專案功能、檢查新版本的能力，以及記憶體和效能管理。	2015 年 12 月 2 日
1.0.101	維護版本	新增建立新的資料庫遷移專案精靈。新增可將資料庫遷移評估報告儲存為 PDF 檔案的功能。	2015 年 10 月 19 日
1.0.100	預覽版	提供 AWS Schema Conversion Tool 預覽版的使用者指南。	2015 年 10 月 7 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。