

Teradata to Redshift Migration

using AWS SCT

Contributor's: Sanket Surendra Wagh

Document creation date: 11/08/2019

Last update date: 11/12/2019

Overview :

This document gives us an overview of setting Teradata(locally), Redshift(on AWS) and utilizing AWS SCT tool for schema and data migration. Also, access Redshift through SQL Workbench and Teradata through Teradata SQL Assistant.

Objectives:

The objective of this document is of setting Teradata, Redshift, and connect them to AWS SCT for schema and data migration.

Prerequisites:

Understanding of Teradata, Redshift, AWS SCT, VPC, and Networking

Requirements/Task(s):

System Requirements:

- Minimum 12gb of local windows system RAM. Recommended is 16gb RAM
- A free IP address in the same CIDR range as of local windows system
- Attach this IP address to the static public IP address of the local windows system

What is Teradata?

Teradata is a massively parallel open processing system for developing large-scale data warehousing applications. Teradata is an open system. It can run on Unix/Linux/Windows server platform. This tool provides support to multiple data warehouse operations at the same time to different clients.

Teradata Corporation is an American IT firm. It is a vendor of analytic data platforms, applications, and other related services. The firm develops a product to consolidate data from various sources and make the data available for analysis.

What is Redshift?

A Redshift Database is a cloud-based, big data warehouse solution offered by Amazon. The platform provides a storage system that lets companies store petabytes of data in easy-to-access “clusters” that can be queried in parallel. Each of these nodes can be accessed independently by users and applications. Redshift is designed to be used with a variety of data sources and data analytics tools and is compatible with several existing SQL-based clients.

The platform’s architecture makes it easy to create a Redshift connection to a variety of business intelligence tools.

What is AWS SCT?

You can use the AWS Schema Conversion Tool (AWS SCT) to convert your existing database schema from one database engine to another. You can convert relational OLTP schema or data warehouse schema. Your converted schema is suitable for an Amazon Relational Database Service (Amazon RDS) MySQL DB instance, an Amazon Aurora DB cluster, an Amazon RDS PostgreSQL DB instance, or an Amazon Redshift cluster. The converted schema can also be used with a database on an Amazon EC2 instance or stored as data on an Amazon S3 bucket.

AWS SCT supports several industry standards, including Federal Information Processing Standards (FIPS), for connections to an Amazon S3 bucket or another AWS resource. AWS SCT is also compliant with the Federal Risk and Authorization Management Program (FedRAMP).

AWS SCT supports the following OLTP conversions.

Source Database	Target Database on Amazon RDS
Microsoft SQL Server (version 2008 and later)	Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, MariaDB 10.2 and 10.3, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (version 5.5 and later)	Aurora PostgreSQL, MySQL, PostgreSQL You can migrate schema and data from MySQL to an Aurora MySQL DB cluster without using AWS SCT. For more information, see Migrating Data to an Amazon Aurora DB Cluster .
Oracle (version 10.2 and later)	Aurora MySQL, Aurora PostgreSQL, MariaDB 10.2 and 10.3, MySQL, Oracle, PostgreSQL
PostgreSQL (version 9.1 and later)	Aurora MySQL, MySQL, PostgreSQL
IBM Db2 LUW (versions 9.1, 9.5, 9.7, 10.5, and 11.1)	Aurora MySQL, MariaDB 10.2 and 10.3, MySQL, PostgreSQL, Aurora PostgreSQL
Apache Cassandra (versions 2.0, 3.0, 3.1.1, and 3.11.2)	Amazon DynamoDB
Sybase (16.0 and 15.7)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

AWS SCT supports the following data warehouse conversions.

Source Database	Target Database on Amazon Redshift
Greenplum Database (version 4.3 and later)	Amazon Redshift
Microsoft SQL Server (version 2008 and later)	Amazon Redshift
Netezza (version 7.0.3 and later)	Amazon Redshift
Oracle (version 10 and later)	Amazon Redshift
Teradata (version 13 and later)	Amazon Redshift
Vertica (version 7.2.2 and later)	Amazon Redshift

Installations and configurations:

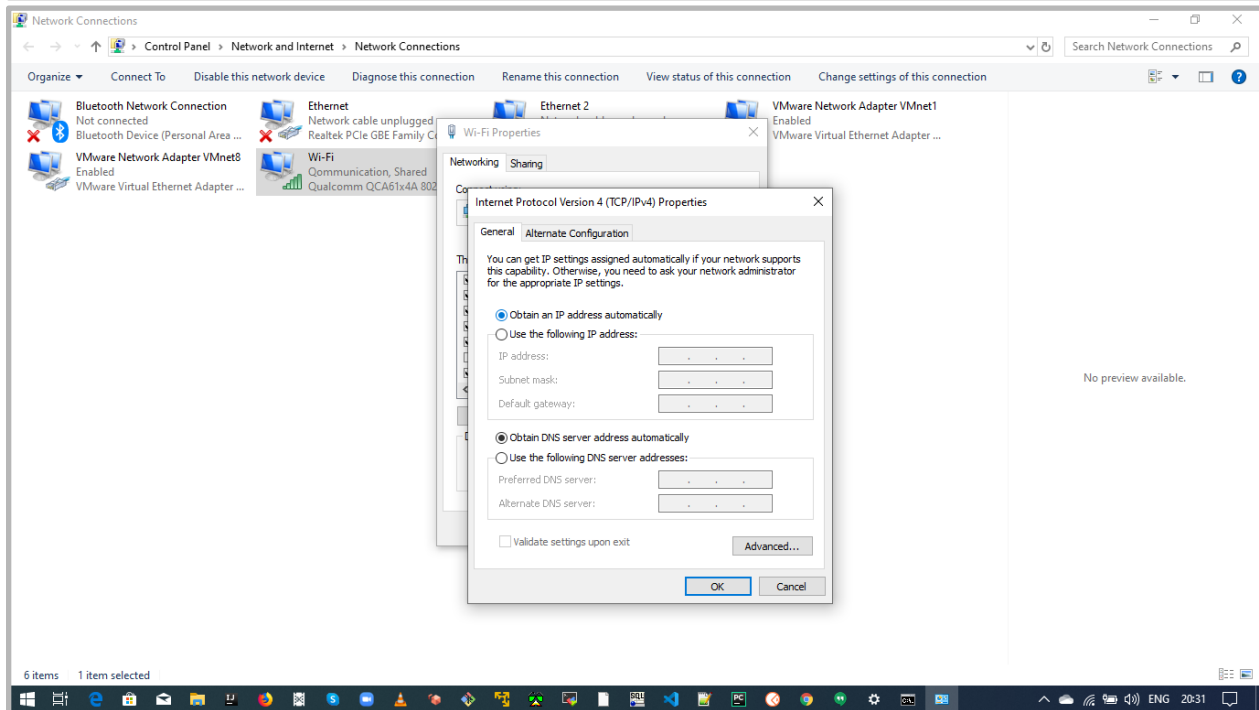
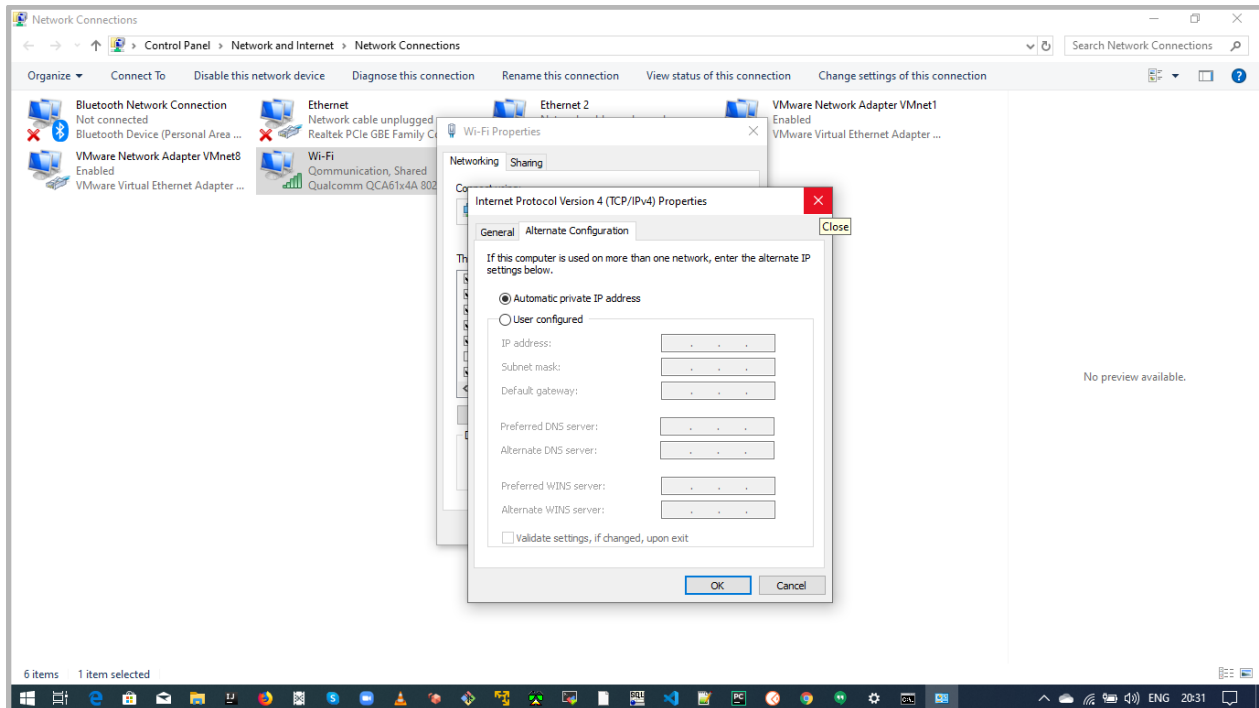
1) Installing Teradata on local windows system:-

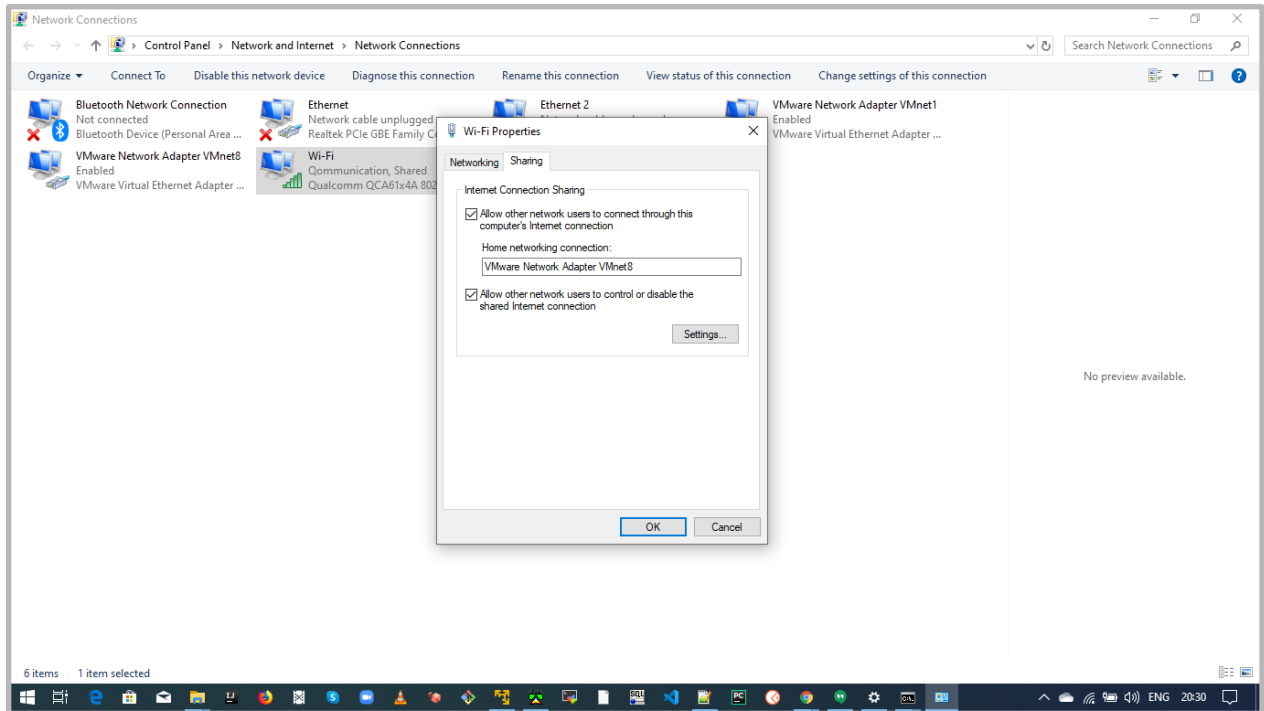
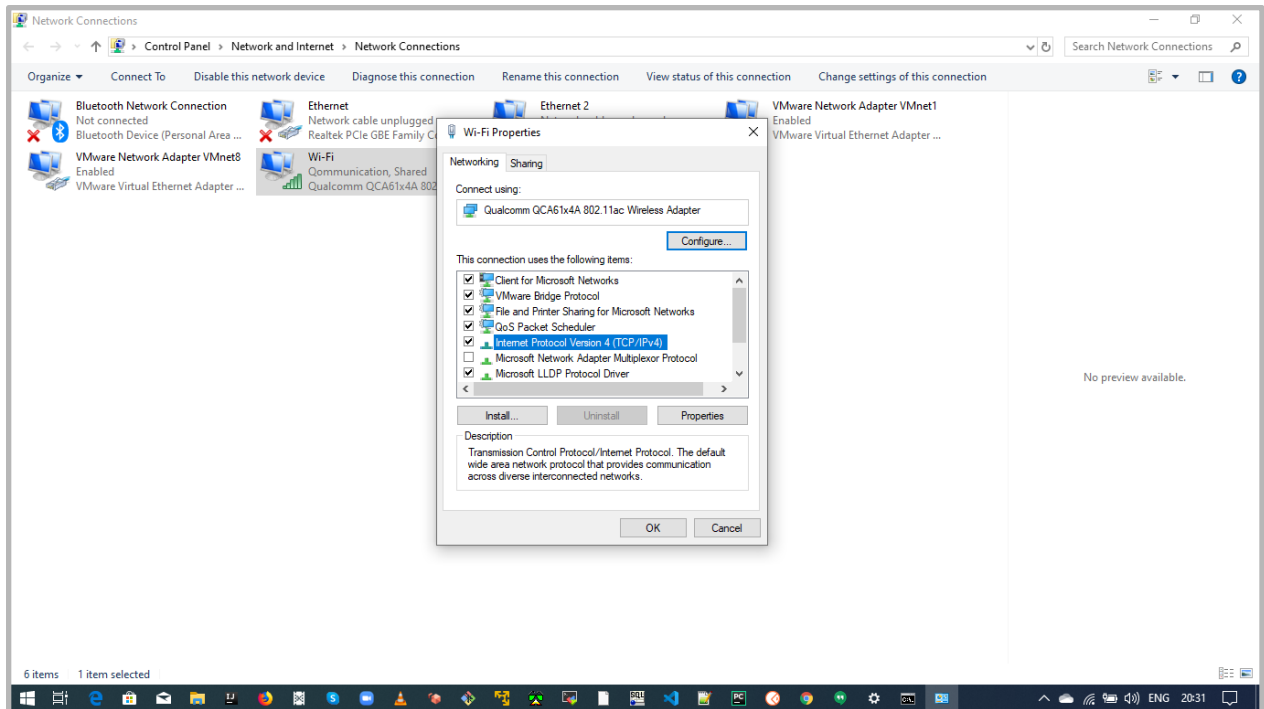
- Use the following [link](#) for setting up Teradata:
- Use the following [link](#) to connect to Teradata Virtual Machine Guest from Windows Host:
- Use the following [link](#) to learn the basics of Teradata

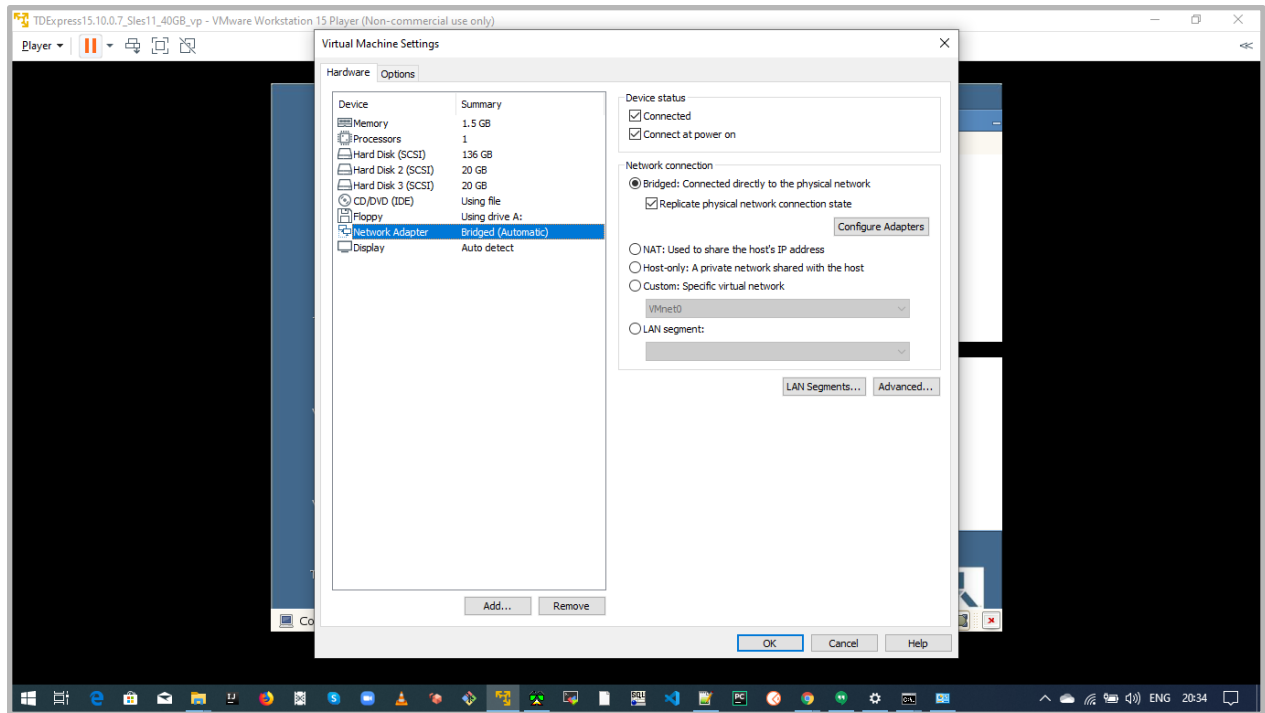
NOTE: This document contains setting up Teradata on VMware workstation player 15(Linux System) and accessing the same on the local windows system. Then using a static public IP address for exposing the local Teradata over the internet.

2) Configure the network settings from the Network and Sharing Center:-

- Go to Change adapter settings and make the following changes:







Then you will require a static public IP address that needs to be attached to your internet IP for exposing it to the Internet

Step 1:

Login into the SLES11 VM with username “root” and password “root”. Wait for sometimes, it will initialize the Teradata.

Step 2

Once it is ready, open Gnome terminal and you can check the status of Teradata database server using below command-

```
TDExpress15.00.01_Sles11:~# pdestate -a
PDE state is RUN/STARTED.
DBS state is 4: Logons are enabled - Users are logged on
TDExpress15.00.01_Sles11:~#
```

How to start Teradata database?

```
/etc/init.d/tpa start
bteq
```

Check your system IP using ifconfig to login into the database

```
.logon localip/dbc
```

```
password: dbc
```

```
sel * from dbc.dbcinfo;
```

Use the following command to create database

```
CREATE DATABASE ADW_TABLES FROM dbc AS PERMANENT = 5000000 BYTES,  
FALLBACK, BEFORE JOURNAL, DUAL AFTER JOURNAL, DEFAULT JOURNAL  
TABLE = tbl_jrn;
```

```
Show databases;
```

```
SELECT * FROM DBC.Databases;
```

To execute the btq script file using windows bteq cmd

```
bteq < filename.btq log.log
```

```
.logon server.mycompany.com/username, password;  
DATABASE databasename;  
SELECT DATE;  
.QUIT ERRORCODE;
```

[Reference](#)

3) Insert data into Teradata using Teradata SQL Assistant

```
INSERT INTO TABLE_NAME VALUES (?, ?, ?, ?....);
```

[Reference](#)

Default port number for teradata is 1025

While the Hyper-V role should not be installed by default on any EC2 Windows instances, it is also not possible to run additional layers of virtualization within your EC2 instances. VMWare needs to be installed on bare-metal hardware with certain virtualization features exposed in the CPU which are unavailable in EC2. You can learn more about these requirements in [this detailed](#) article.

Nested virtualization is not supported on AWS instances unless you are using AWS bare metal instances:

[Reference1](#)

[Reference2](#)

4) Setup Redshift cluster on AWS Cloud:-

Creating a Redshift Cluster inside a VPC involves creating a separate Security Group, a Cluster Subnet Group and finally Node Configuration to launch a cluster.

Step 1:

Create a security group for Redshift access

1. Go to VPC dashboard and then on the Security Group tab,
2. Click on Create Security Group and add VPC ID.
3. Then go to Inbound Rules and choose Type as Redshift. Add PORT and Source accordingly.

Step 2:

In order to create a Redshift Cluster under VPC, we need to create Cluster Subnet Group first.

1. Open Redshift dashboard and click on create Cluster
2. In the Security tab, go to Security Group, then click on Create Cluster Subnet Group.
3. Add Name, Description and the choose the VPC ID of the VPC under which we need to create Cluster.
4. Choose Availability Zone and Subnet ID, then add it to the configuration. For best practice, ADD 3 subnet ID.
5. Click on Create, Cluster Subnet Group with the fields mentioned that will be created.

Reference ([YouTube video](#))

Step 3:

1. Go to Redshift Dashboard and click on Quick Launch Cluster
2. Go to Advanced Setting, add (1) Cluster Details and (2) Node Configuration.
3. In (3) Additional Configuration, choose the VPC and Security Group created in Step 1 and 2.

NOTE: The Redshift cluster is publically available for AWS SCT to connect.

5) Setup AWS SCT:-

- Use the following [link](#) for setting up AWS SCT on windows system:
- Use the following [link](#) for setting up SCT agent on RHEL and data migration:

Appendix

Database migration assessment report (sample):-

Source database: EODS_TABLES.dbc@59.152.53.170

Teradata 15.10.00.07

AWS Schema Conversion Tool Version 1.0.631

10/17/2019 3:44PM

Executive summary

We completed the analysis of your Teradata source database and estimate that 7% of the database storage objects and none of database code objects can be converted automatically or with minimal changes if you select

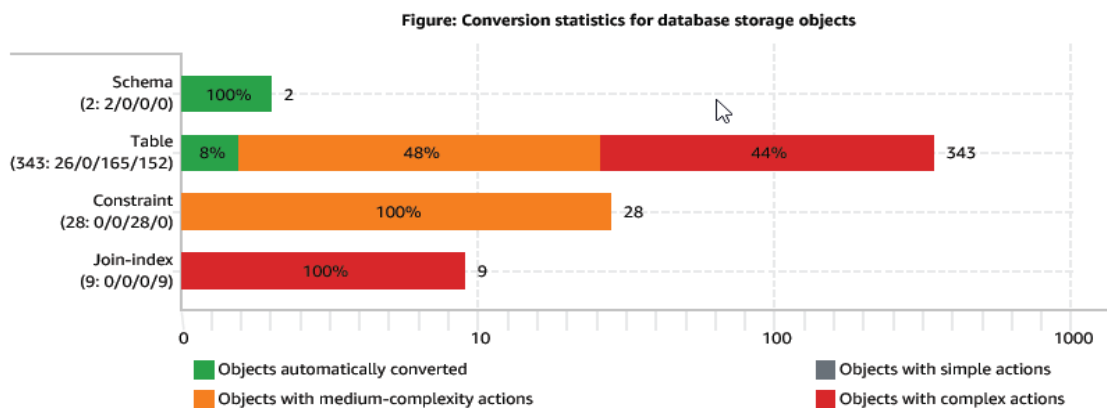
Amazon Redshift as your migration target. Database storage objects include databases, tables, constraints, indexes, join-indexes and user defined types. Database code objects include functions, procedures, triggers,

views, attributes, variables, constants, cursors, exceptions and other objects. To complete the migration, we

recommend 540 conversion actions ranging from simple tasks to medium-complexity actions to complex conversion actions.

Database objects with conversion actions for Amazon Redshift

Of the total 382 database storage object(s) in the source database, we identified 28 (7%) database storage object(s) that can be converted to Amazon Redshift automatically or with minimal changes.



Detailed recommendations for Amazon Redshift migrations

If you migrate your Teradata database to Amazon Redshift, we recommend the following actions.

Storage object actions

Join-index Changes

Not all join-indexes can be converted automatically. You'll need to address these issues manually.

Issue 13110: Amazon Redshift does not support Join Indexes

Recommended action: Join Index columns are candidates for a sortkey.

Issue code: 13110 | Number of occurrences: 9 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/c_unsupported-postgresql-features

Schemas.EODS_TABLES.Join indexes.Claim_TJI004

Schemas.EODS_TABLES.Join indexes.Disbursement_TJI004

Schemas.EODS_TABLES.Join indexes.Disbursement_TJI008

Schemas.EODS_TABLES.Join indexes.GroupAccount_TJI004

Page 1 of 4

AWS Schema Conversion Tool Version 1.0.631

10/17/2019 3:44PM

Database migration assessment report

Source database: EODS_TABLES.dbc@59.152.53.170

Teradata 15.10.00.07

Schemas.EODS_TABLES.Join indexes.GroupInsuranceAgreement_TJI004

+4 more

Constraint Changes

Not all constraints can be converted automatically. You'll need to address these issues manually.

Issue 13102: Unique constraints are informational and are not enforced by the system

Recommended action: Please check your application code.

Issue code: 13102 | Number of occurrences: 11 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/t_Defining_constraints.html

Schemas.EODS_TABLES.Tables.AccountMember.Constraints.ak_AccountMember

Schemas.EODS_TABLES.Tables.Commission.Constraints.ak_1Commission

Schemas.EODS_TABLES.Tables.CommissionLine.Constraints.SUI_CommissionLine

Schemas.EODS_TABLES.Tables.CommissionLine_13062019.Constraints.SUI_CommissionLine

Schemas.EODS_TABLES.Tables.CoveragePlan.Constraints.ak_CoveragePlan

+6 more

Issue 13103: Primary key constraints are informational and are not enforced by the system

Recommended action: Please check your application code.

Issue code: 13103 | Number of occurrences: 17 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/t_Defining_constraints.html

Schemas.EODS_TABLES.Tables.AccountLine_Sprint44.Constraints.pk_AccountLine

Schemas.EODS_TABLES.Tables.BillStatus.Constraints.pk_BillStatus

Schemas.EODS_TABLES.Tables.ClaimGracePeriodUnit.Constraints.pk_ClaimGracePeriodUnit

Schemas.EODS_TABLES.Tables.ClaimMedicalProcedureCode.Constraints.pk_ClaimMedicalProcedureCode

Schemas.EODS_TABLES.Tables.ContractLevel.Constraints.pk_ContractLevel

+12 more

Table Changes

Not all tables can be converted automatically. You'll need to address these issues manually.

Issue 13002: Unable to convert datatypes

Recommended action: Performed autoconversion to the CHARACTER VARYING type.

Issue code: 13002 | Number of occurrences: 12 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/c_unsupported-postgresql-datatype

Schemas.ADW_TABLES.Tables.WorkflowDatacapBatchDocumentFact_Sprint49.Columns.DatacapIndexingInHouse...

Schemas.ADW_TABLES.Tables.WorkflowDatacapBatchDocumentFact_Sprint49.Columns.DatacapIndexingInHouse...

Schemas.ADW_TABLES.Tables.WorkflowDatacapBatchDocumentFact_Sprint49.Columns.DatacapMetricAflacRecei...

Schemas.ADW_TABLES.Tables.WorkflowDatacapBatchDocumentFact_Sprint49.Columns.DatacapMetricAflacRece...

Schemas.EODS_TABLES.Tables.Bill.Columns.EffectivePeriod

+7 more

Issue 13101: Amazon Redshift does not support set tables. Duplicate rows are permitted

Recommended action: Please check your application code.

Page 2 of 4

AWS Schema Conversion Tool Version 1.0.631

10/17/2019 3:44PM

Database migration assessment report

Source database: EODS_TABLES.dbc@59.152.53.170

Teradata 15.10.00.07

Issue code: 13101 | Number of occurrences: 18 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/c_unsupported-postgresql-features

Schemas.ADW_TABLES.Tables.IndividualClaimBalancingFact
Schemas.ADW_TABLES.Tables.IndividualClaimBalancingG0G1Fact
Schemas.ADW_TABLES.Tables.IndividualClaimBatchSummaryDim
Schemas.ADW_TABLES.Tables.IndividualClaimGen0Dim
Schemas.ADW_TABLES.Tables.IndividualClaimGen1Dim
+13 more

Issue 13102: Unique constraints are informational and are not enforced by the system

Recommended action: Please check your application code.

Issue code: 13102 | Number of occurrences: 325 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/t_Defining_constraints.html

Schemas.ADW_TABLES.Tables.AccountDim
Schemas.ADW_TABLES.Tables.AccountLineDim
Schemas.ADW_TABLES.Tables.AccountLineTransactionFact
Schemas.ADW_TABLES.Tables.AccountLineTransactionFact_Sprint50Sep13
Schemas.ADW_TABLES.Tables.AccountManagerCaseListing
+320 more

Issue 13106: Secondary indexes are not supported in Amazon Redshift

Recommended action: Index columns are a good option for Sortkey columns.

Issue code: 13106 | Number of occurrences: 107 | Estimated complexity: Simple

Documentation references: https://docs.aws.amazon.com/redshift/latest/dg/c_unsupported-postgresql-features

Schemas.ADW_TABLES.Tables.AccountDim
Schemas.ADW_TABLES.Tables.AccountLineDim
Schemas.ADW_TABLES.Tables.AccountLineTransactionFact
Schemas.ADW_TABLES.Tables.AccountLineTransactionFact_Sprint50Sep13
Schemas.ADW_TABLES.Tables.AccountManagerCaseListing
+102 more

Issue 13109: Only the first column from Primary Index has been taken as Distribution Key

Recommended action: Please check selectiveness for all primary index columns and choose the best candidate for Distribution Key.

Issue code: 13109 | Number of occurrences: 34 | Estimated complexity: Simple

Documentation references:

https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_TABLE_NEW.html

Schemas.ADW_TABLES.Tables.EnrollmentSetBatchFact

Schemas.ADW_TABLES.Tables.WorkflowCaseApplicationDateTimeMetricFact

Schemas.ADW_TABLES.Tables.WorkflowCaseDim

Schemas.ADW_TABLES.Tables.WorkflowDataCapDocumentDim

Schemas.ADW_TABLES.Tables.WorkflowDatacapBatchDim

+29 more

Issue 13140: Amazon Redshift TIMESTAMP WITH TIME ZONE columns store values with up to a maximum of 6 digits of precision for fractional seconds. TIMESTAMP WITH TIME

Page 3 of 4

AWS Schema Conversion Tool Version 1.0.631

10/17/2019 3:44PM

Database migration assessment report

Source database: EODS_TABLES.dbc@59.152.53.170

Teradata 15.10.00.07

ZONE will be truncated to the supported number of digits

Recommended action: TIMESTAMP WITH TIME ZONE truncated for supported digits.

Issue code: 13140 | Number of occurrences: 7 | Estimated complexity: Simple

Schemas.EODS_TABLES.Tables.EnrollmentCase.Columns.EnrollmentCaseEffectiveEndDate

Schemas.EODS_TABLES.Tables.EnrollmentCase.Columns.EnrollmentCaseEffectiveStartDate

Schemas.EODS_TABLES.Tables.EnrollmentCase.Columns.OpenEnrollmentBeginDate

Schemas.EODS_TABLES.Tables.EnrollmentCase.Columns.OpenEnrollmentEndDate

Schemas.EODS_TABLES.Tables.EnrollmentCaseTransmittal.Columns.TransmittalDate

+2 more

Page 4 of 4

References:

Description	Links
Teradata Installation Guide	https://www.teradatapoint.com/teradata/install-teradata-15-in-your-pc.htm
Teradata Installation	https://www.tutorialspoint.com/teradata/teradata_installation.htm
Teradata on AWS Documentation	https://docs.teradata.com/reader/FhzPBW7kXjoiJ1KIWkg~uw/UX1WmoAgR3azw9hVnIeP0A
Teradata tools and utilities (Windows)	https://downloads.teradata.com/download/tools/teradata-tools-and-utilities-windows-installation-package
Teradata Express for VMPlayer	https://downloads.teradata.com/download/database/teradata-express-for-vmware-player
Teradata tools and utilities (Linux)	https://downloads.teradata.com/download/tools/teradata-tools-and-utilities-linux-installation-package-0
Teradata CLIV2 for Linux	https://downloads.teradata.com/download/connectivity/teradata-cliv2-for-linux
AWS SCT Installation	https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Installing.html#CHAP_Installing.JDBCDrivers
How to use SSL with AWS SCT Data Extractors	https://aws.amazon.com/blogs/database/how-to-use-ssl-with-the-aws-schema-conversion-tool-data-extractors/
Migrate Data from On-Premises Data Warehouse to Redshift	https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/agents.dw.html#agents.Tasks
Amazon EC2 Pricing	https://aws.amazon.com/ec2/pricing/on-demand/
AWS Forum thread	https://forums.aws.amazon.com/thread.jspa?threadID=180236

AWS Forum thread	https://forums.aws.amazon.com/thread.jspa?threadID=293113
Amazon EC2 Bare Metal Instances	https://aws.amazon.com/blogs/aws/new-amazon-ec2-bare-metal-instances-with-direct-access-to-hardware/
General Availability of Bare Metal EC2 Instances	https://aws.amazon.com/about-aws/whats-new/2018/05/announcing-general-availability-of-amazon-ec2-bare-metal-instances/
AWS MarketPlace	https://aws.amazon.com/marketplace/pp/Amazon-Web-Services-Microsoft-Windows-Server-2019-/B07RJTTGML?stl=true#pdp-pricing
AWS MarketPlace	https://aws.amazon.com/marketplace/pp/B06Y4Z54R5?qid=1570089580090&sr=0-1&ref=srh_res_product_title#pdp-pricing
Connect to Teradata Data in AWS Glue Jobs using JDBC	https://www.cdata.com/kb/tech/teradata-jdbc-aws-glue.rst