ORACLE

Teradata to ADW Migration with SQL Developer

Carlos Álvarez – Carlos.CA.Alvarez@oracle.com

Data Management Modernise Cloud Solution Specialist

July 2023 | Version 1.0

Contents

1. Document Control

1.1 Version Control

Version	Author	Date	Comment
0.1	Carlos Álvarez	2023-06-06	Draft
1.0	Carlos Álvarez	2023-07-04	First version

2. Overview

Legacy Data Warehouses, like Teradata, present a lot of inconveniences nowadays. The monolithic architecture implies high costs both in terms of hosting and management. Also, legacy Data Warehouses are difficult to scale (up/down, out/in) and require very skilled personnel to operate.

Companies are moving these legacy on-prem systems to modern, cloud-native, easy managed Data Warehouses and migrations are becoming more and more frequent. And here is where Oracle Autonomous Data Warehouse (ADW) shines, offering a flexible, scalable, highest performing, highly available, and secure data warehouse solution that reduces administrative costs.

3. Introduction

Migration of a database is a complex process that involves a lot of tasks that must be delivered. From preparation, to design, from data extraction to data loading, from analysis to validation.

SQL Developer can assist this process by using its built-in functionalities and tools. Using SQL Developer in a migration can help to reduce the efforts, accelerate the whole process and reduce human errors.

4. Tools of the trade

The three main parts of the migration explored in this document are the legacy system to migrate (Teradata), the cloud native Data Warehouse solution (Oracle Autonomous Data Warehouse - ADW) and the tool to help and accelerate the process (Oracle SQL Developer).

4.1. About Oracle Autonomous Data Warehouse (ADW)

Oracle Autonomous Data Warehouse (ADW) is a cloud data warehouse service that is designed to mitigate the complexities of operating a data warehouse, securing data, and developing data-driven applications. It is developed to automate provisioning, configuring, tuning, scaling, and backing up of the data warehouse and includes tools for self-service data loading, data transformations, business models, automatic insights, and built-in converged database capabilities that enable streamlined queries across multiple data types and ML analysis. ADW is available in both Oracle Cloud Infrastructure (OCI) and customer's data centers with Oracle Exadata Cloud@Customer, and Oracle Dedicated Region Cloud@Customer, enabling customers to meet data sovereignty requirements.

4.2. About Teradata

Teradata is a Relational Database Management System designed for building large scale data warehousing applications. Teradata is based on MPP (Massively Parallel Processing), Shared Nothing Architecture.

4.3. About SQL Developer

Oracle SQL Developer is a free, integrated development environment that simplifies the development and management of Oracle Database in both traditional and Cloud deployments. SQL

Developer offers complete end-to-end development of PL/SQL applications, a worksheet for running queries and scripts, a DBA console for managing the database, a reports interface, a complete data modeling solution, and a migration platform for moving 3rd party databases to Oracle.

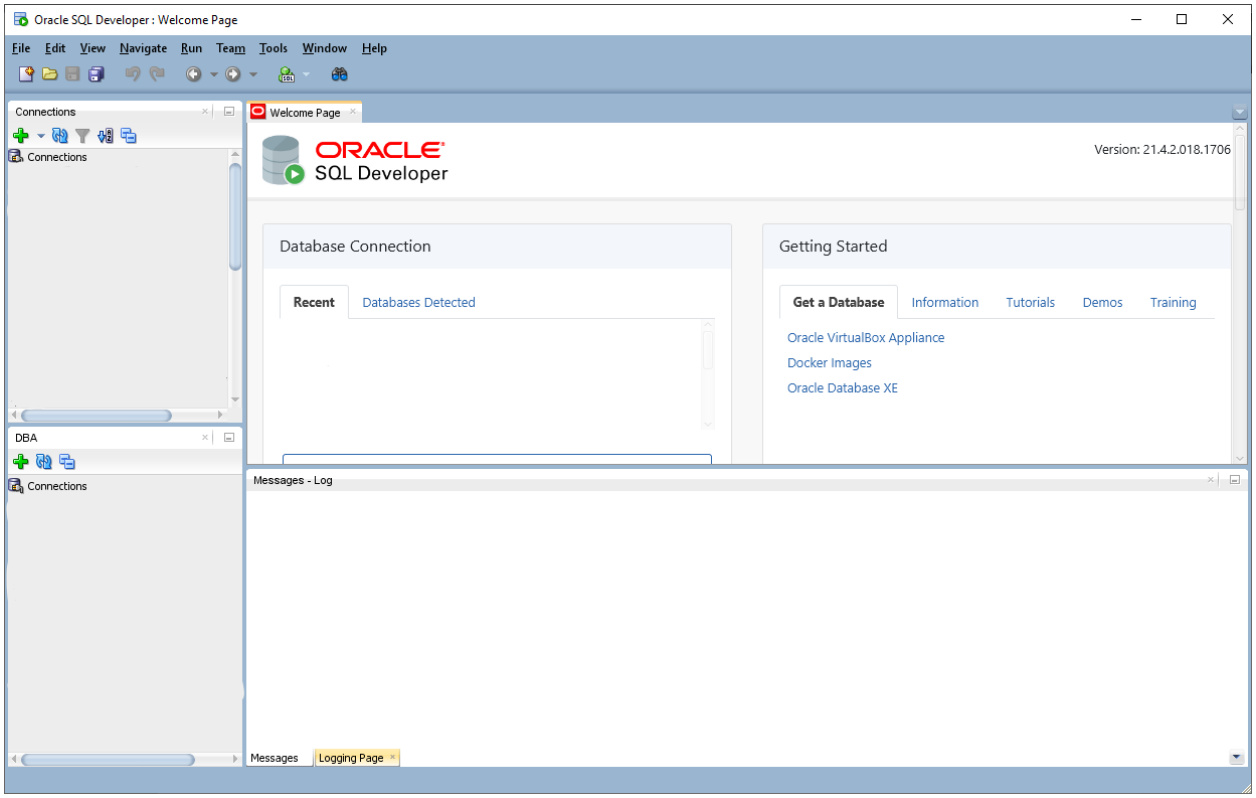
SQL Developer can also connect to third-party databases (i.e.: Teradata) using the appropriate JDBC drivers.

5. Installing SQL Developer

SQL Developer can be installed in Windows, Mac OS X or Linux. The installation consists of downloading the SQL Developer zip file from the Oracle Technology Network Download page for SQL Developer and unzipping it into a directory.

The process is straightforward and out of the scope of this document. The detailed instructions can be found here: https://docs.oracle.com/en/database/oracle/sql-developer/22.2/rptig/installing-sql-developer.html

When the installation process is complete, the SQL Developer can be started just by executing `sqldeveloper.exe` (Windows) or `sqldeveloper.sh` (Linux and Mac OS X) found in the installation directory:



The SQL Developer documentation can be found here: https://docs.oracle.com/en/database/oracle/sql-developer/

6. Connecting to OCI Autonomous Data Warehouse

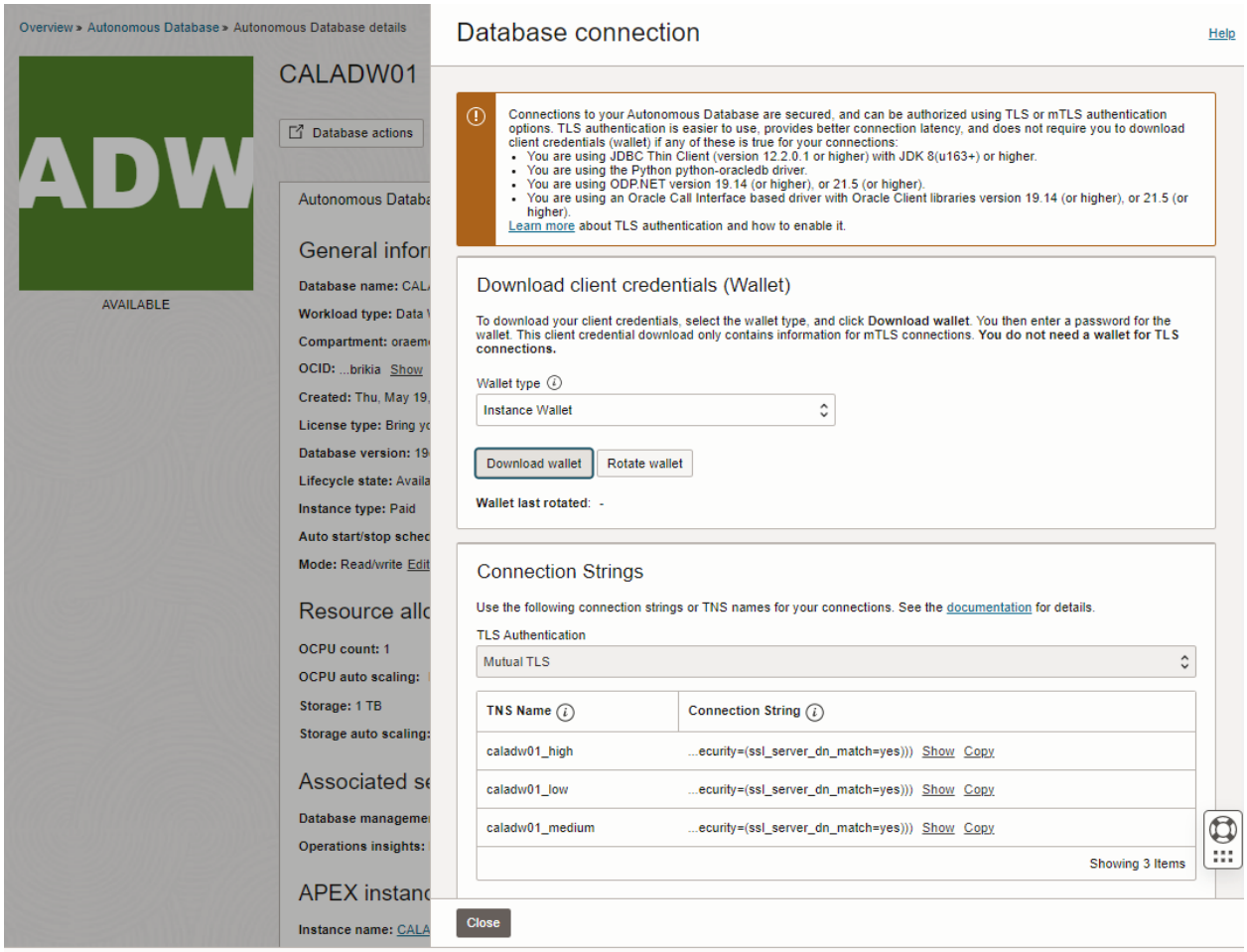
There are different ways to connect SQL Developer to an OCI Autonomous Data Warehouse depending on the options chosen in OCI when the ADW was provisioned (Shared/Dedicated, Public/Private Endpoint/ACLs, etc…).

The OCI ADW is a Shared ADB. For the sake of simplicity, we will be using Mutual Transport Layer Security (mTLS) through a TCPS (Secure TCP) database connection with a Wallet.

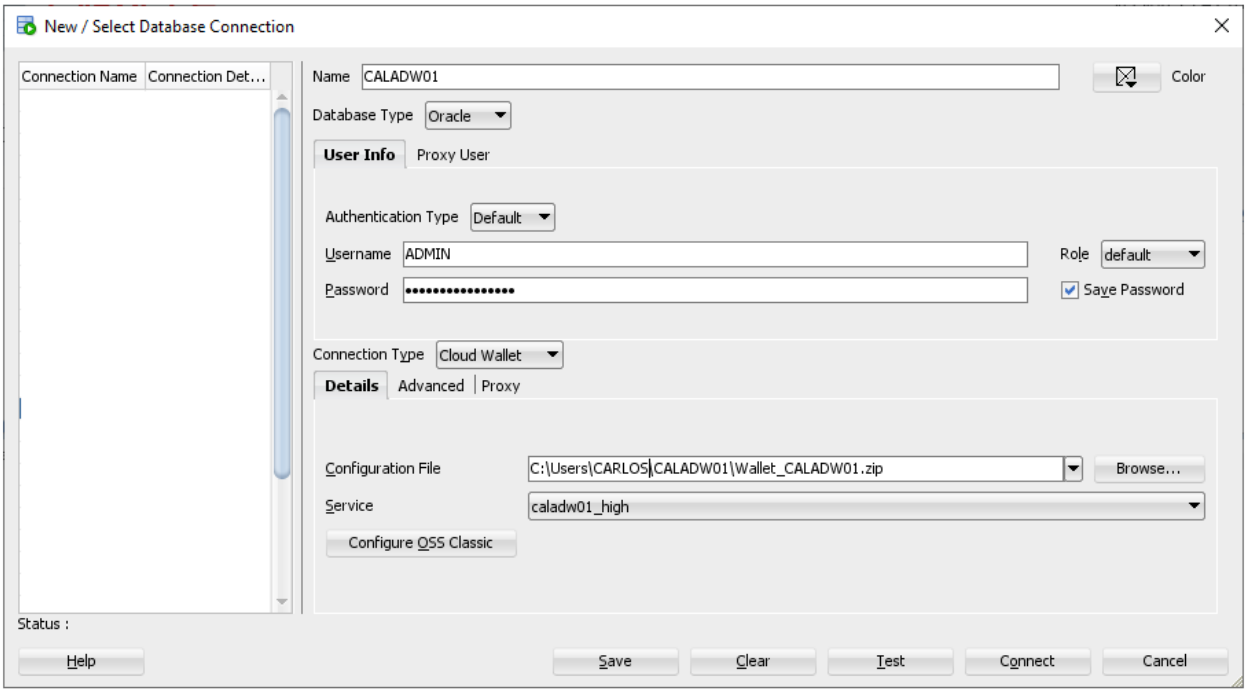
Wallet files, along with the Database user ID and password, provide access to the Autonomous data Warehouse.

The wallet can be downloaded from the OCI Autonomous Database details page.

A password must be provided for the download to complete. The password protects the downloaded Client Credentials wallet.

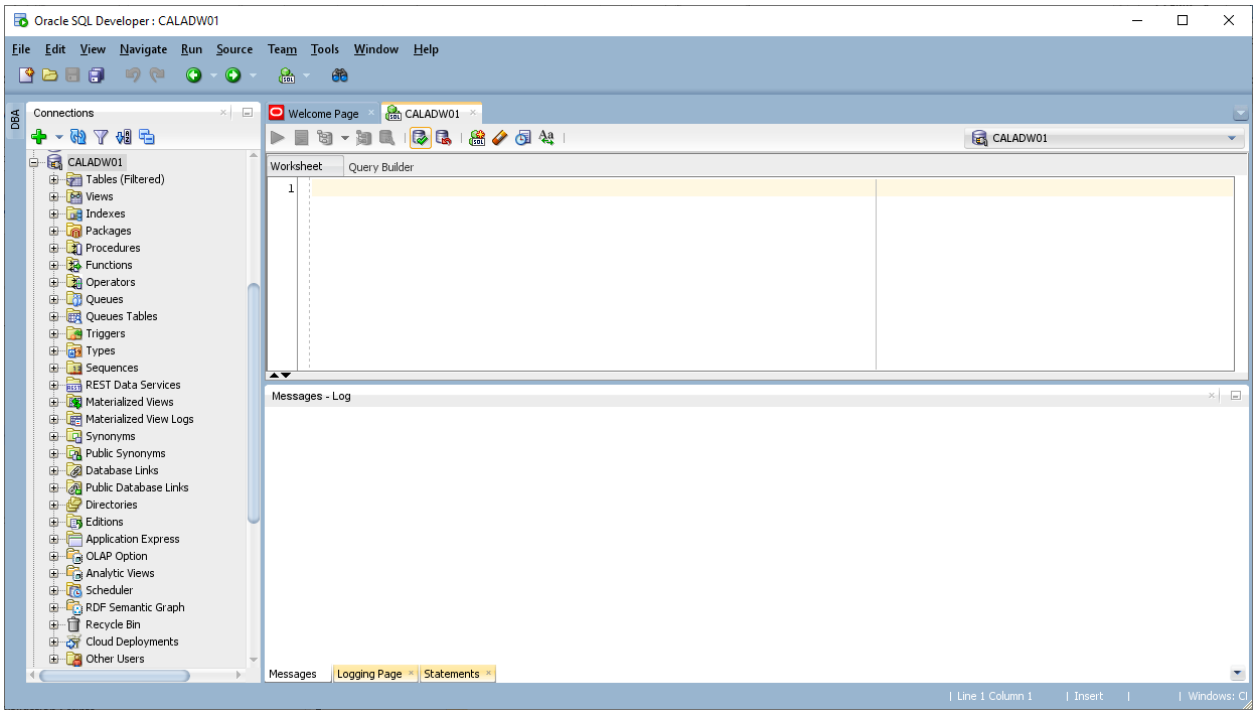


SQL Developer can connect to ADW just referencing the location where the Wallet zip file with the client security credentials is located (“Cloud Wallet” Connection Type).



(Note that SQL Developer can also connect to the ADW using other methods, including Basic, Custom JDBC and TNS).

Once the connection is verified and established with the ADW, SQL Developer shows all the database objects and opens a SQL Worksheet to operate:



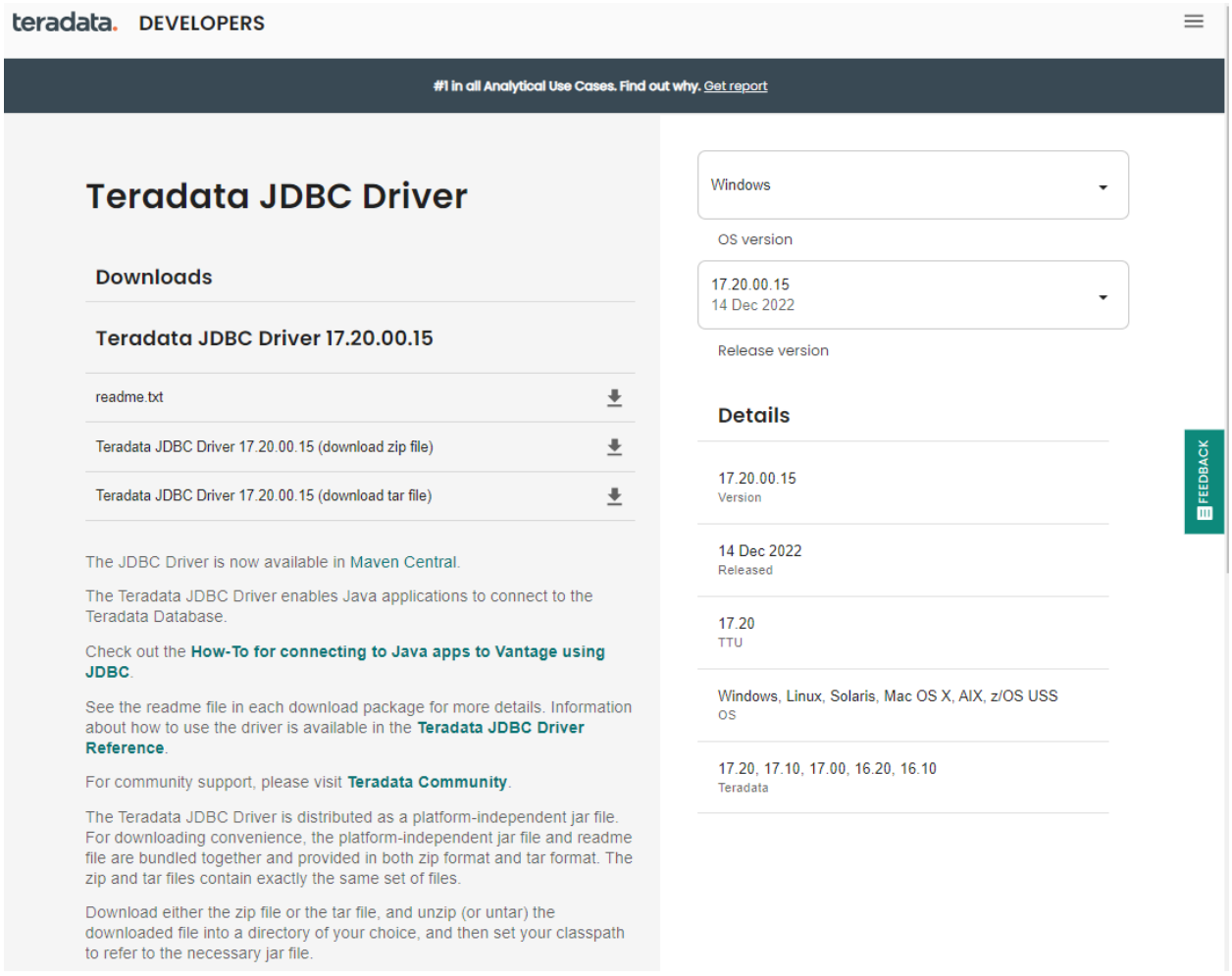
7. Connecting to Teradata

SQL Developer must connect to the Teradata system to extract all the metadata and generate all the required structures for the migration process and optionally transform and load Teradata objects (Tables, Views, Indexes…) into the Autonomous Data Warehouse.

7.1. Download the Teradata JDBC Driver

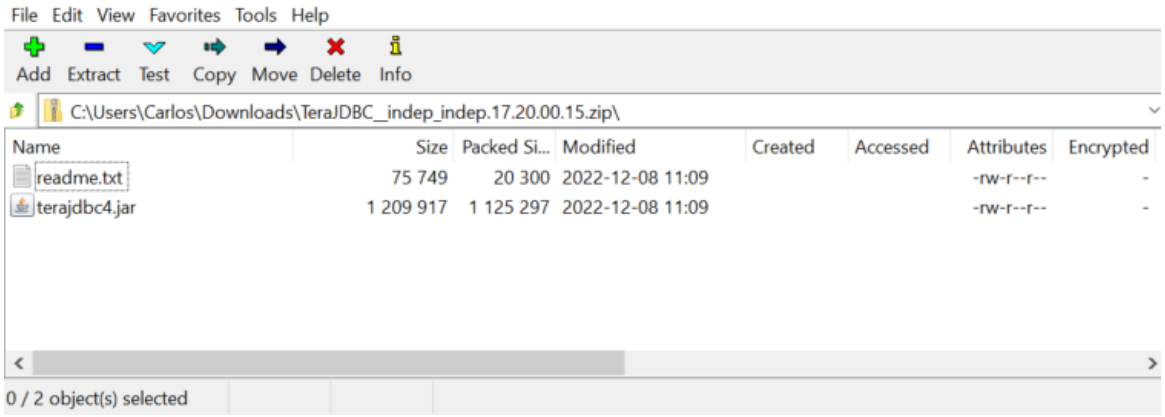
Oracle SQL Developer can connect to third party databases by JDBC connections. To access these databases a third party JDBC driver is required.

In the case of Teradata, the JDBC driver can be downloaded from https://downloads.teradata.com/download/connectivity/jdbc-driver.



(Note that a registered user is needed to download the Teradata JDBC driver from this site, and a License Agreement must be accepted.)

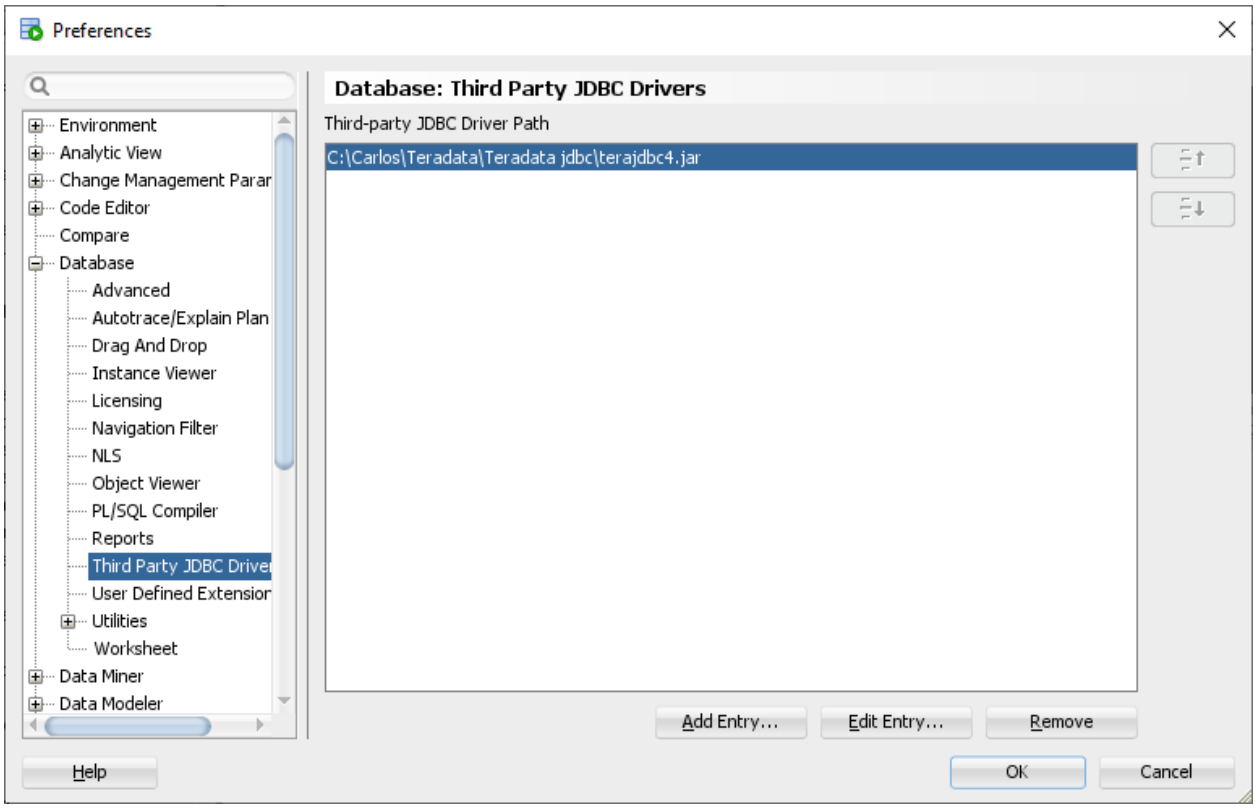
The zip file (in this version `TeraJDBC__indep_indep.17.20.00.15.zip`) contains two files: a `readme.txt` with information, and the JDBC driver file `terajdbc4.jar`.



7.2. Create a Connection to the Teradata Database

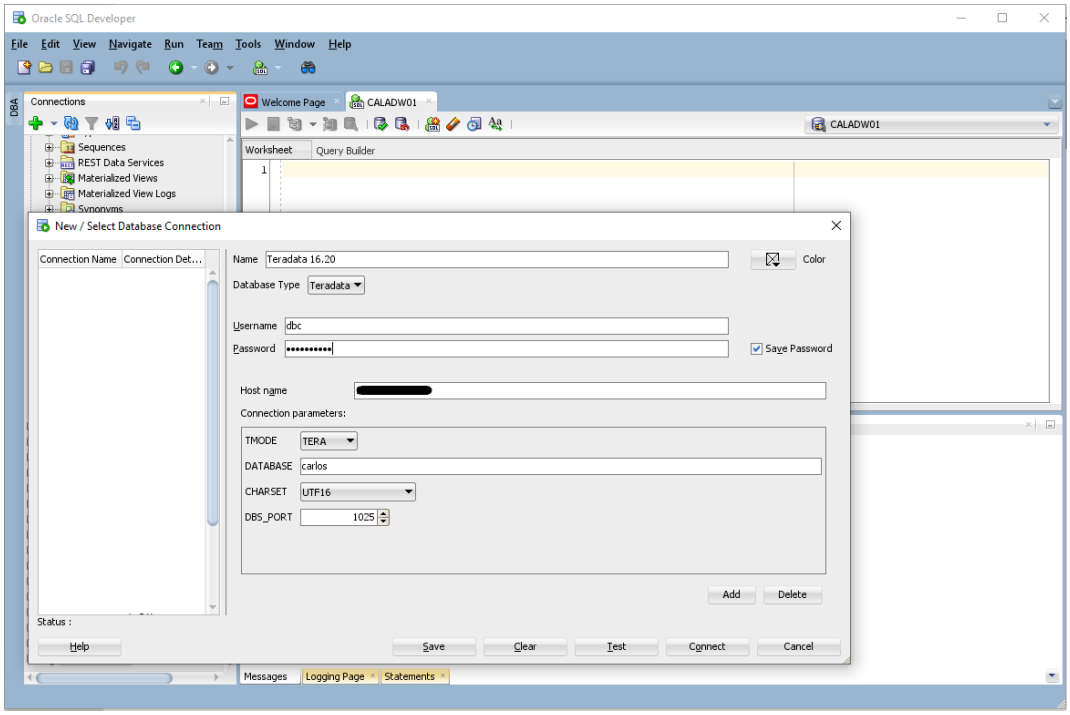
To create a connection to the Teradata database the Teradata JDBC driver must be added to the SQL Developer Third Party JDBC Drivers.

The `terajdbc4.jar` file must be extracted to a directory and the path included in the "Preferences" menu option under Database -> Third Party JDBC Drivers section:

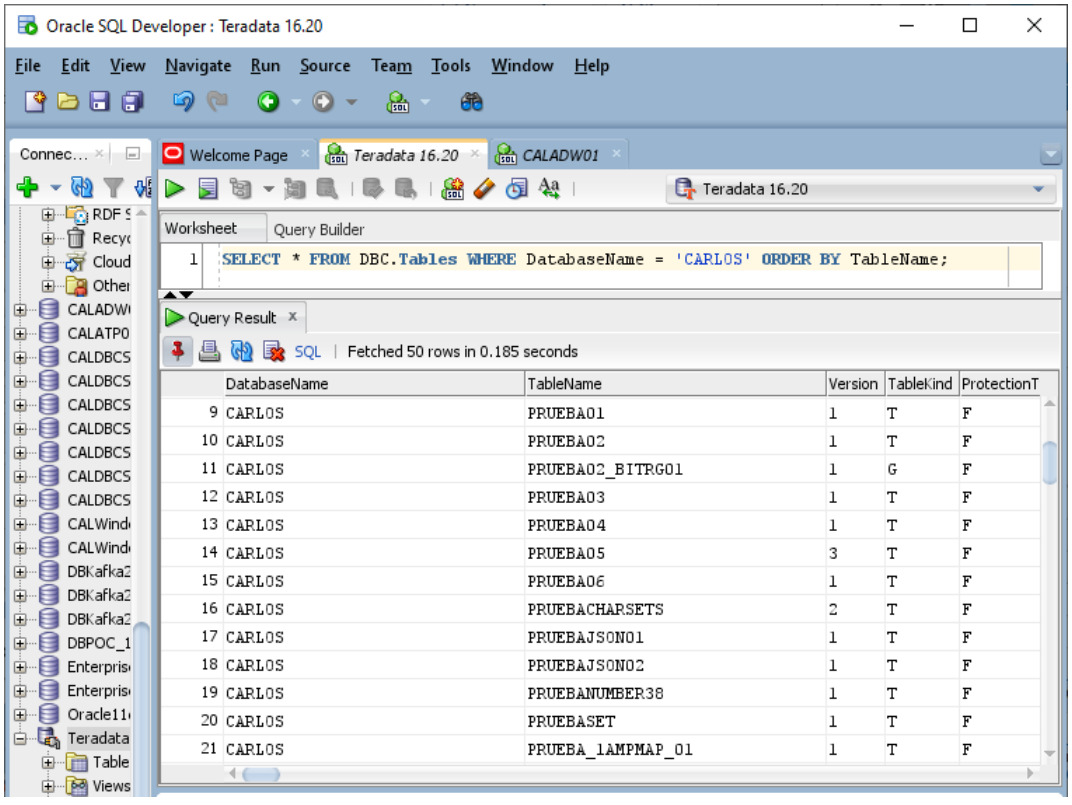


Once the JDBC driver is added, the new connection with Database Type "Teradata" can be created.

By filling all the options for the connection as username/password, Host name or IP address, PORT (typically 1025), mode, database (same as Oracle schema) the connection can be tested and saved:



And the new connection allows SQL Developer to operate with the Teradata database:



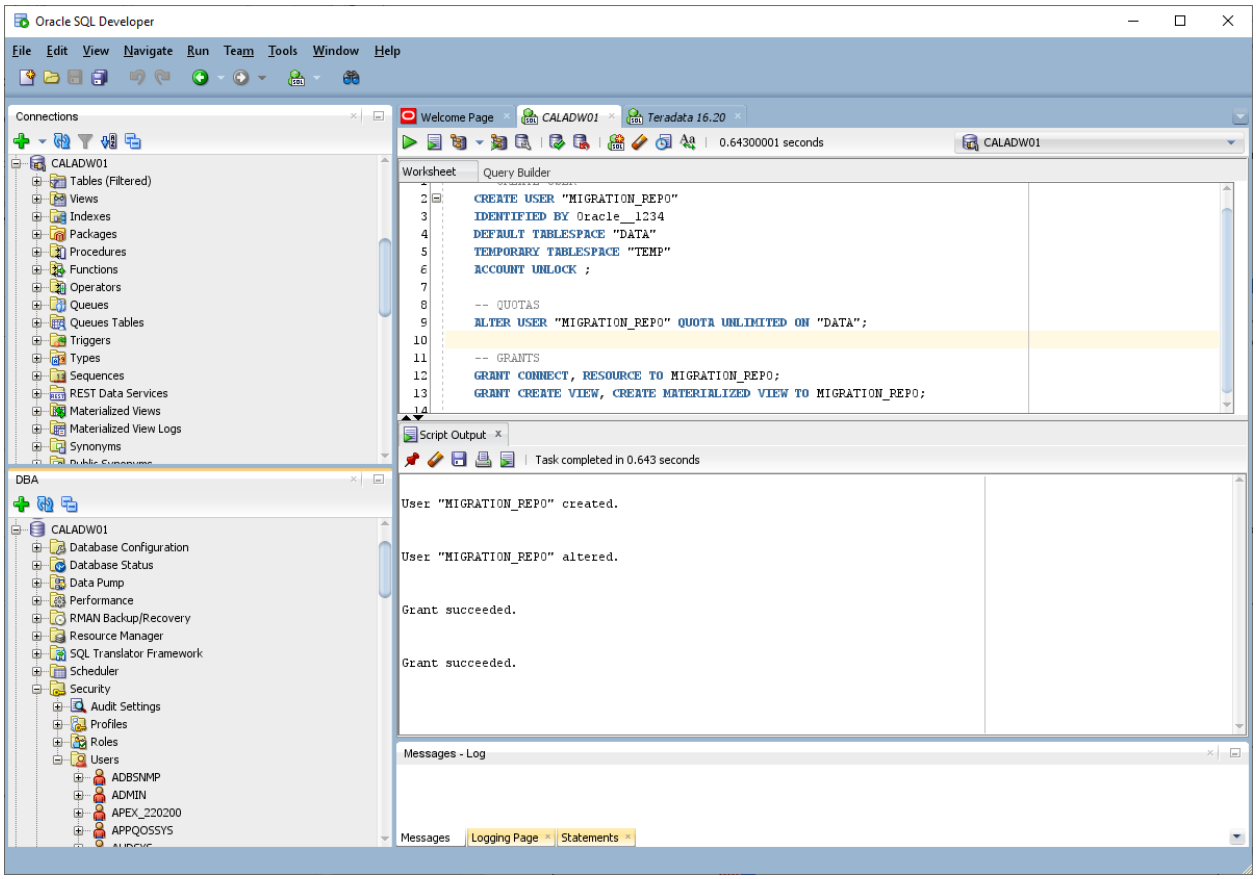
8. Preparing the Migration

8.1. Create a Schema for the migration repository

The migration will use a repository to store the necessary metadata. This metadata will be stored in an Oracle Schema. The tables and objects will be created there when the repository is created.

An Oracle user for the repository must be created in the Autonomous Database. This user must have at least the RESOURCE role and the CREATE SESSION, CREATE VIEW, and CREATE MATERIALIZED VIEW privileges.

This user can be created by the ADMIN user:

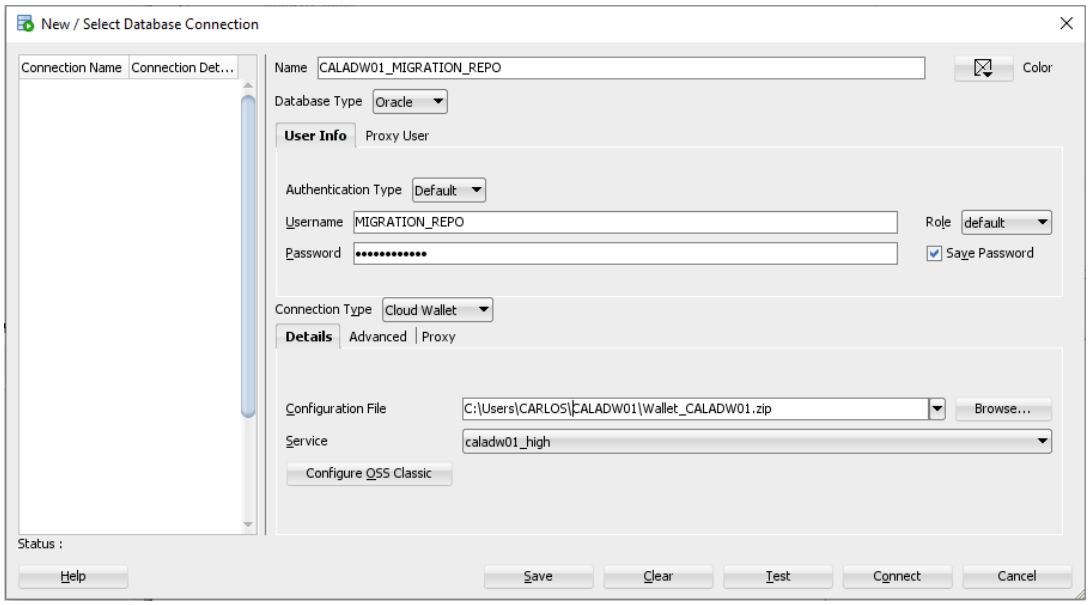


Note: For multischema migrations, the repository user must be granted the “RESOURCE” role with the ADMIN option; and the CREATE ROLE, CREATE USER, and ALTER ANY TRIGGER privileges with the ADMIN option too.

8.2. Create a connection for the migration repository user.

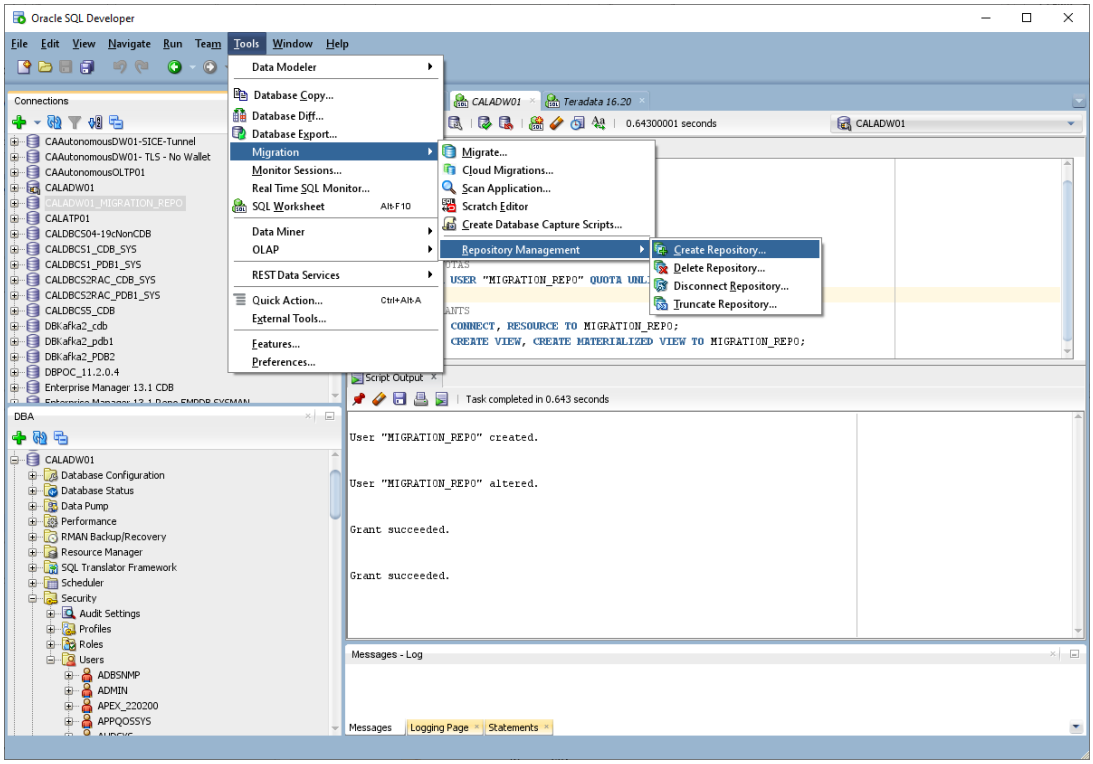
SQL Developer will use a connection to this repository user, so it must be created.

The same Wallet used before for the connection as ADMIN to the ADW can be used for this connection.

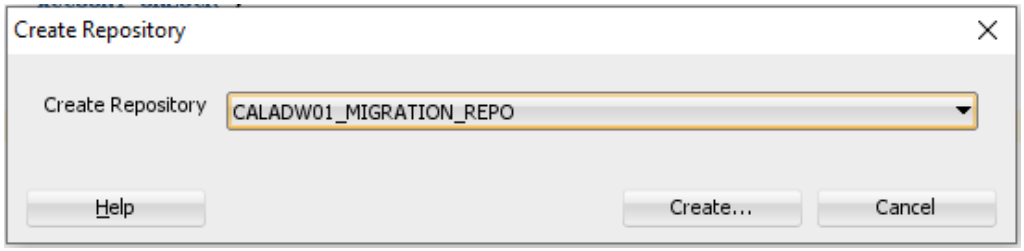


8.3. Create the Repository for the migration.

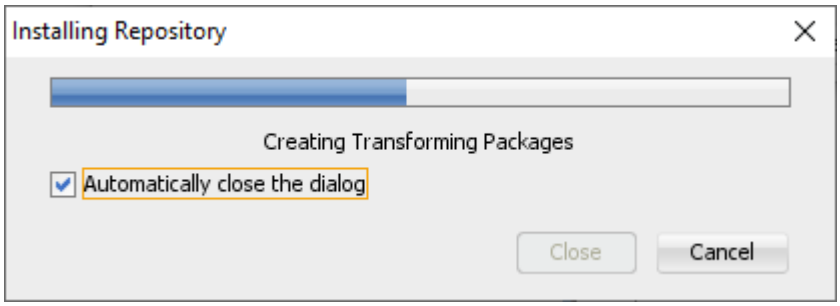
The Tools -> Migration -> Repository Management -> Create Repository menu option is used to create a new repository.



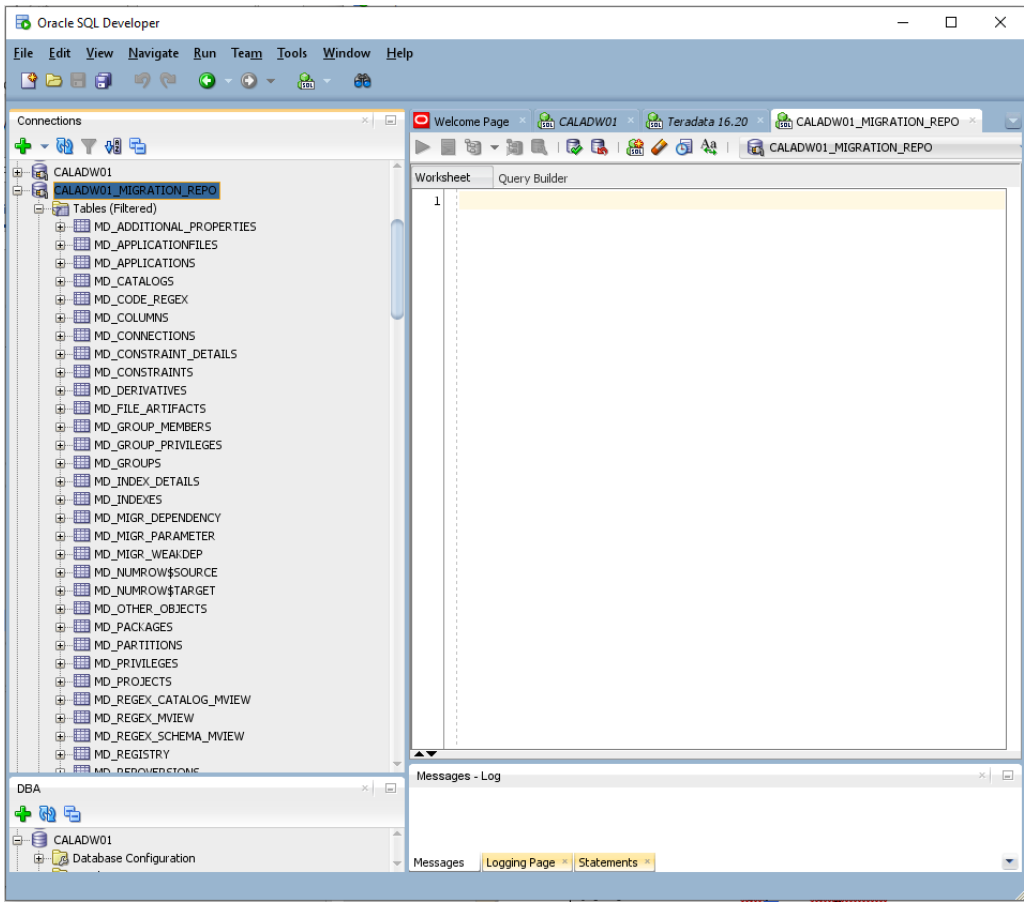
The connection for the repository user created in previous step is selected:



Upon clicking “Create…” the new repository is created in the repository user schema:

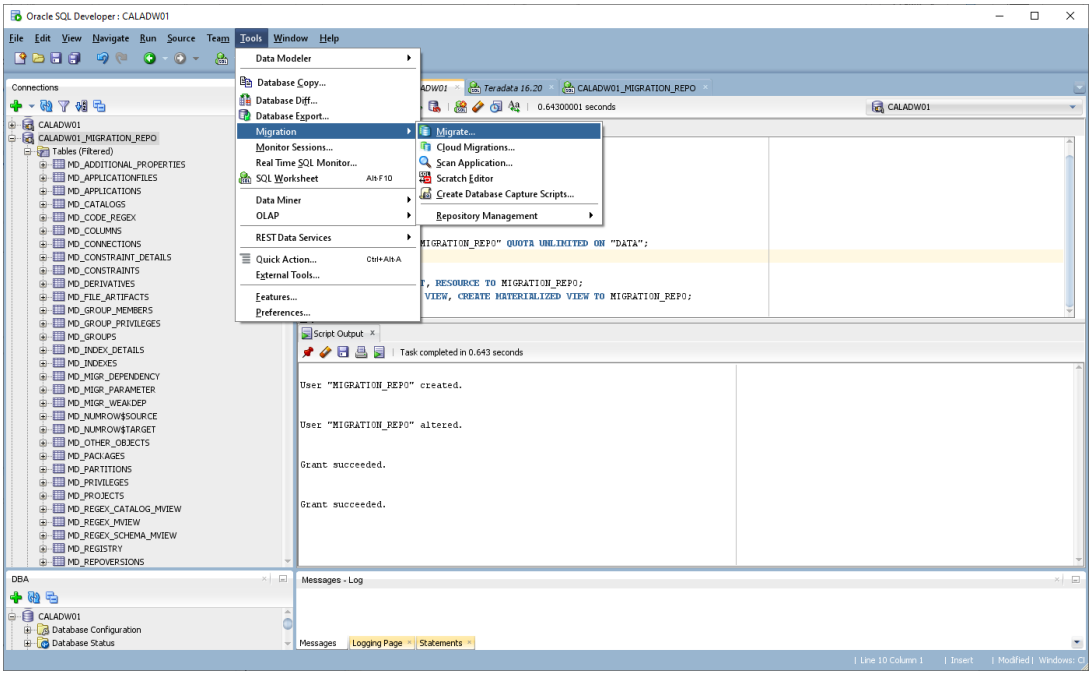


The repository user schema has now all the tables and other objects for the migration projects:

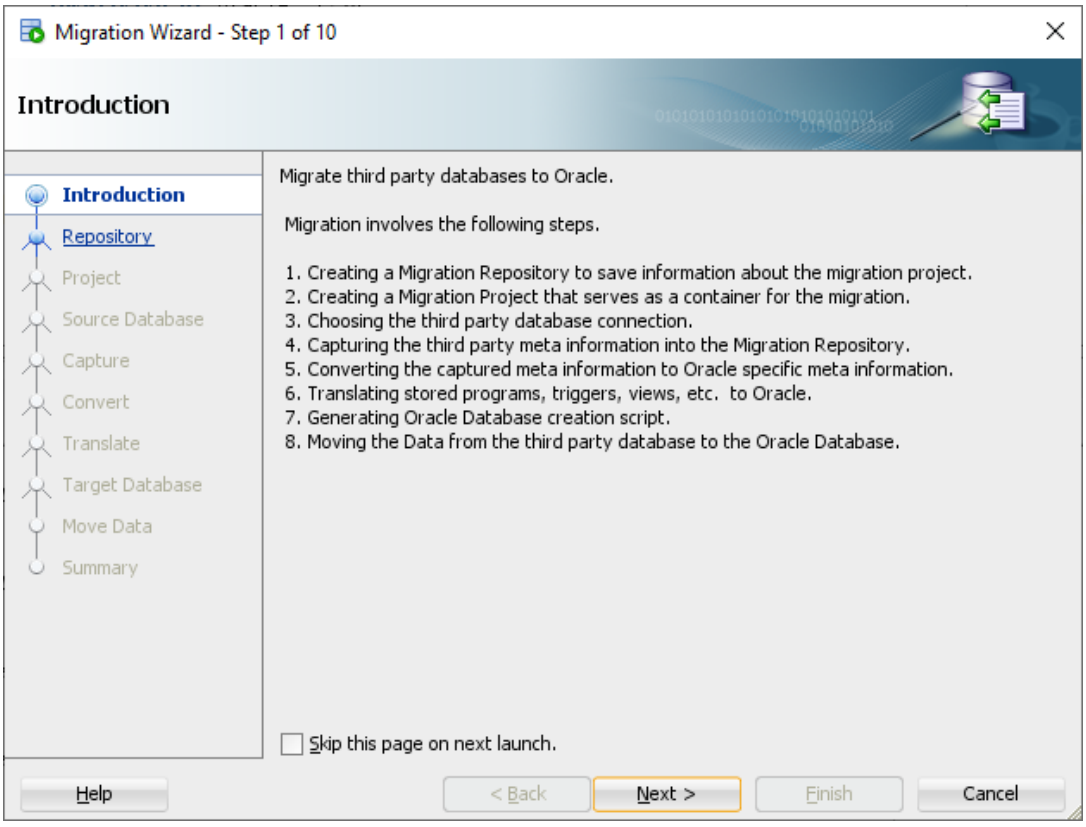


9. Creating the Migration

With the repository created, the migration process can be started by selecting the Menu option “Tools” -> Migrations -> Migrate…



A new window opens with the migration steps to guide the user through the process:

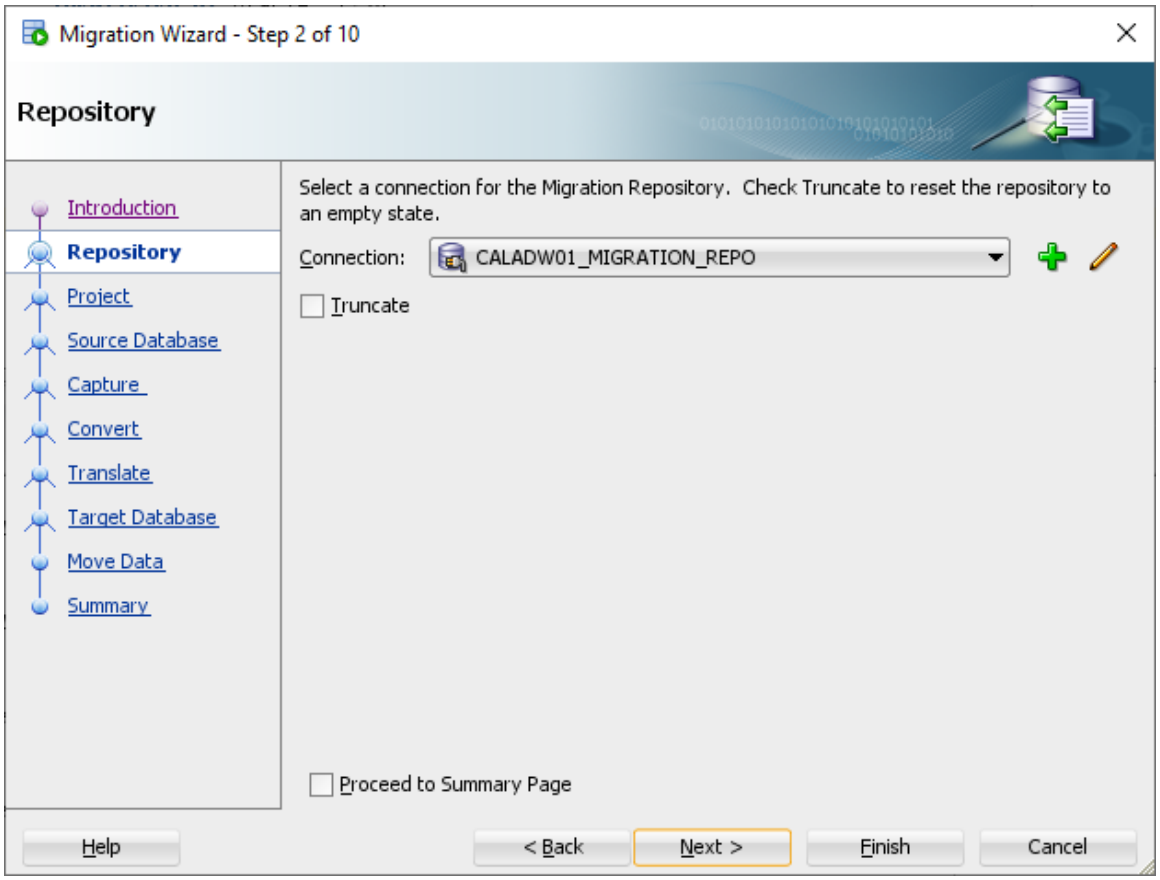


9.1. Choose the Repository Connection

The first step is to choose a connection to a migration repository.

This step also allows to create a new connection or edit an existing one.

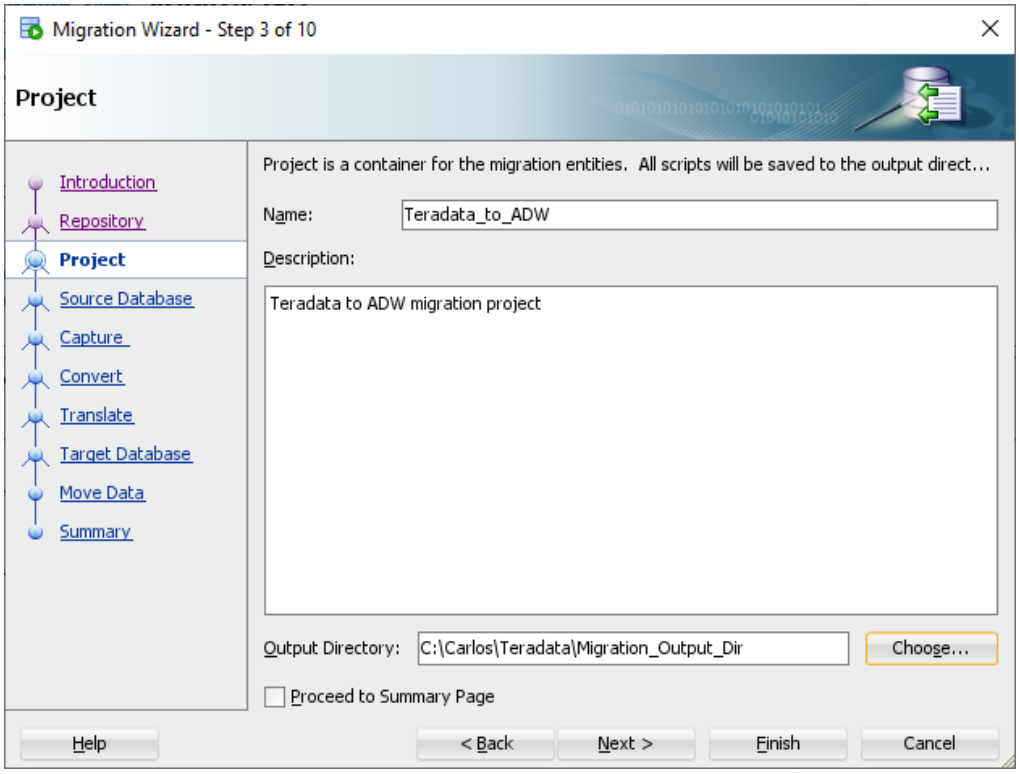
There is an option to truncate all the tables in the repository user schema to clean the data from previous migrations.



9.2. Create the Migration Project

The Project page is where the info for the migration project must be provided:

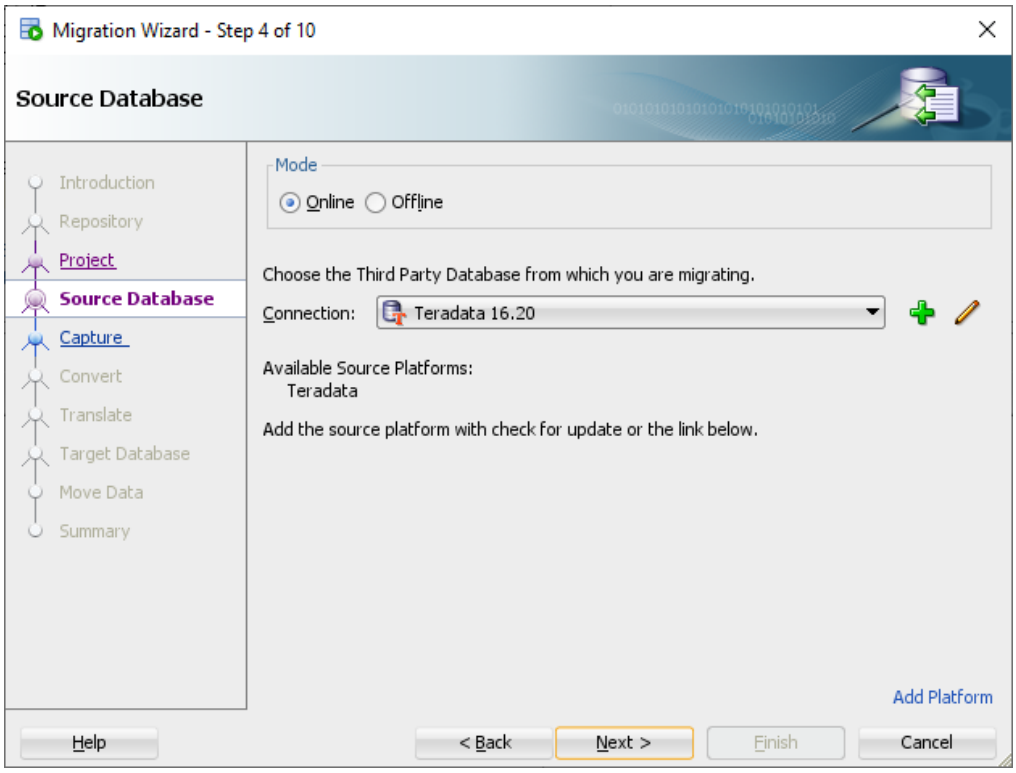
- A name for the migration project.
- A brief description of the migration project (optional).
- The directory in which all the scripts generated by the migration wizard will be placed.



9.3. Select the source database (Teradata).

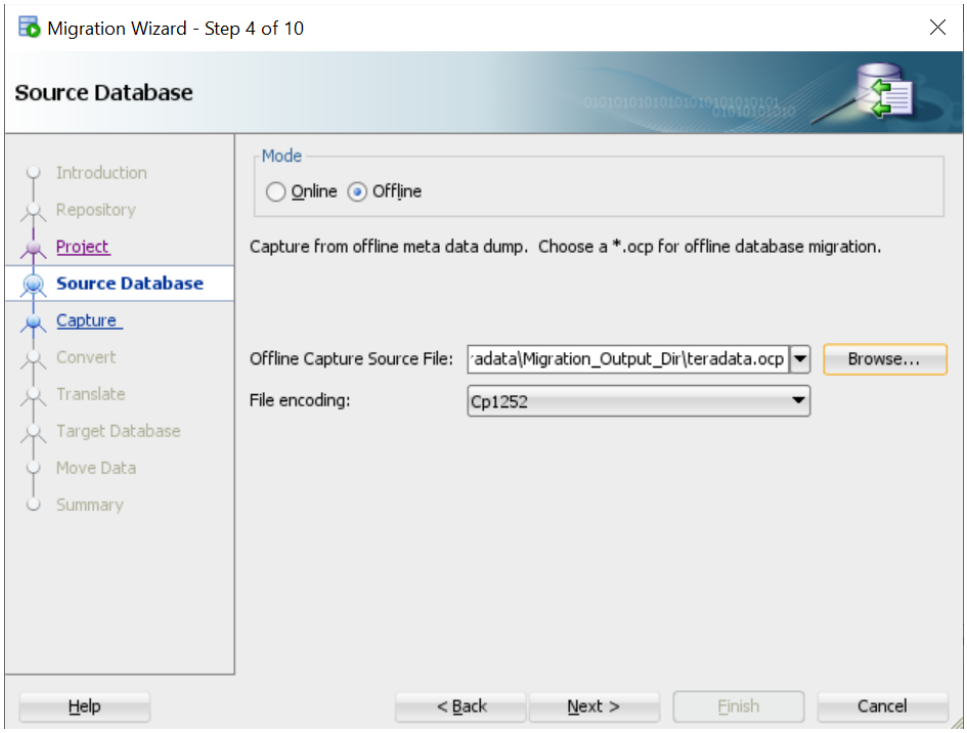
This page allows to choose between Online and Offline modes.

Online Mode extracts the Teradata Database Metadata using the JDBC connection.

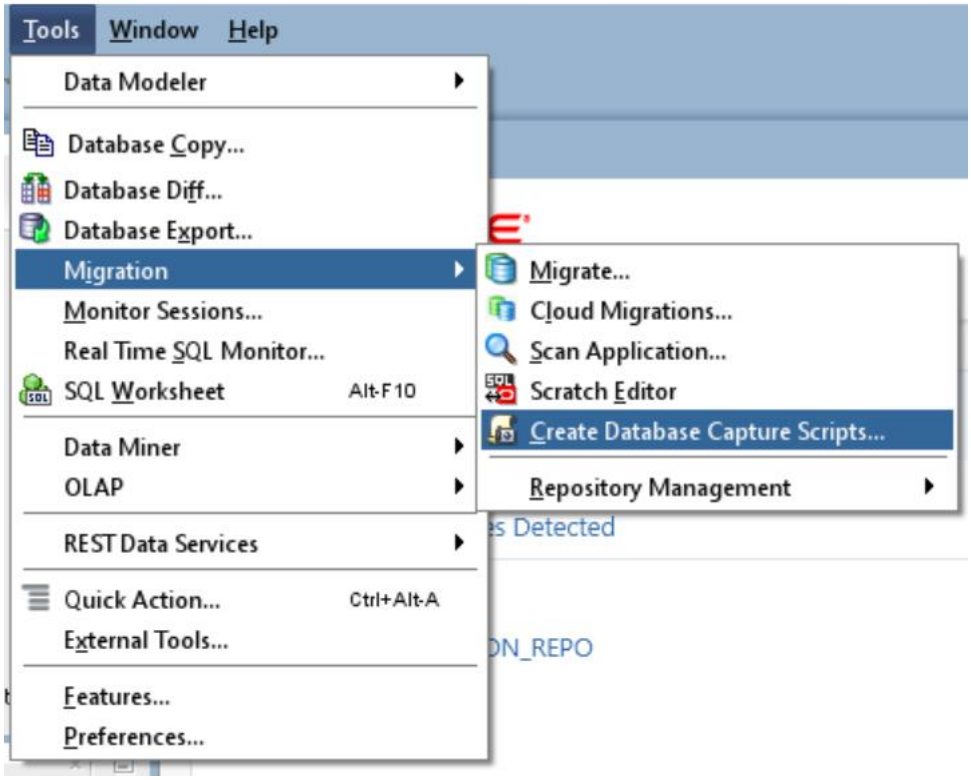


(The Teradata Connection created in the point 7.2 must be selected in this page).

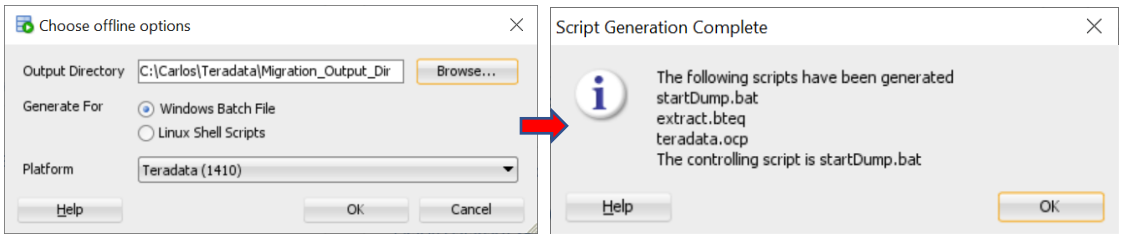
Offline Mode performs the migration using a specified file (*.ocp - Offline Capture Source File).



The *.ocp file can be created in “Tools” -> “Migration” -> “Create Database Capture Scripts...”:

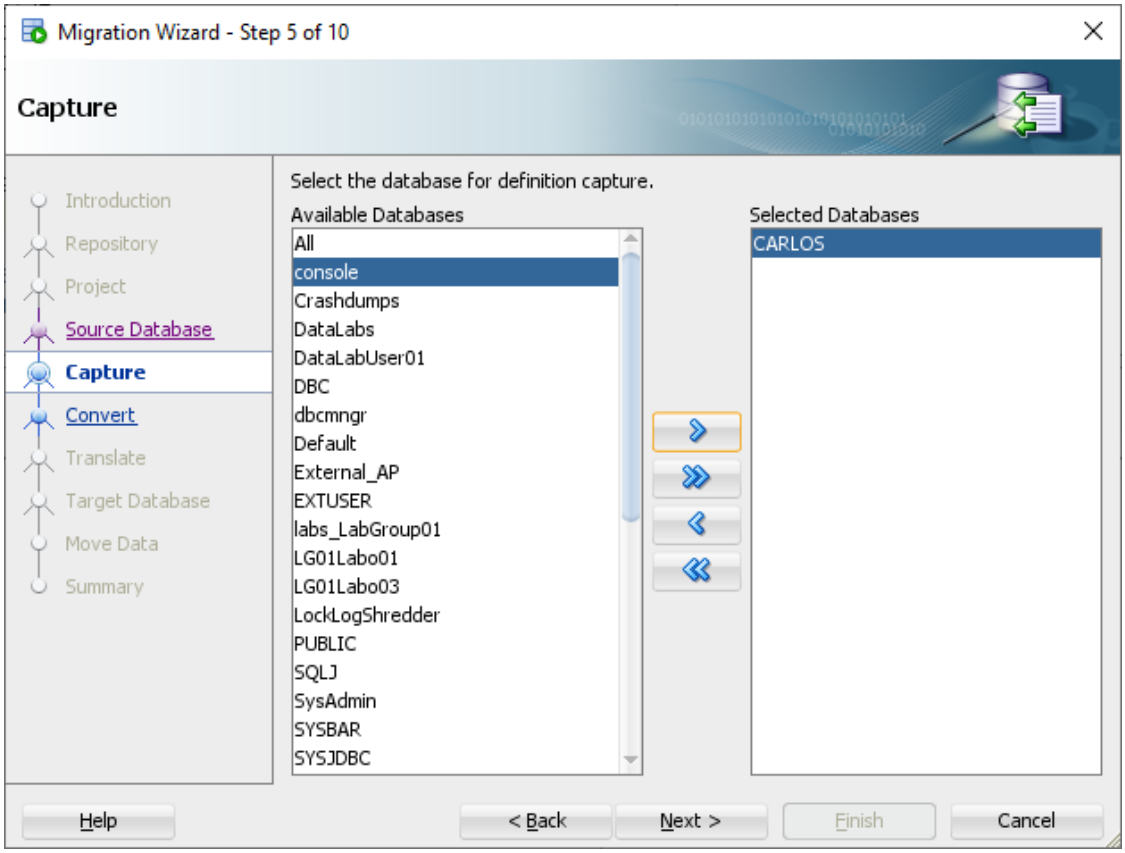


After choosing the output directory and the Source Database Platform, the required scripts are generated:



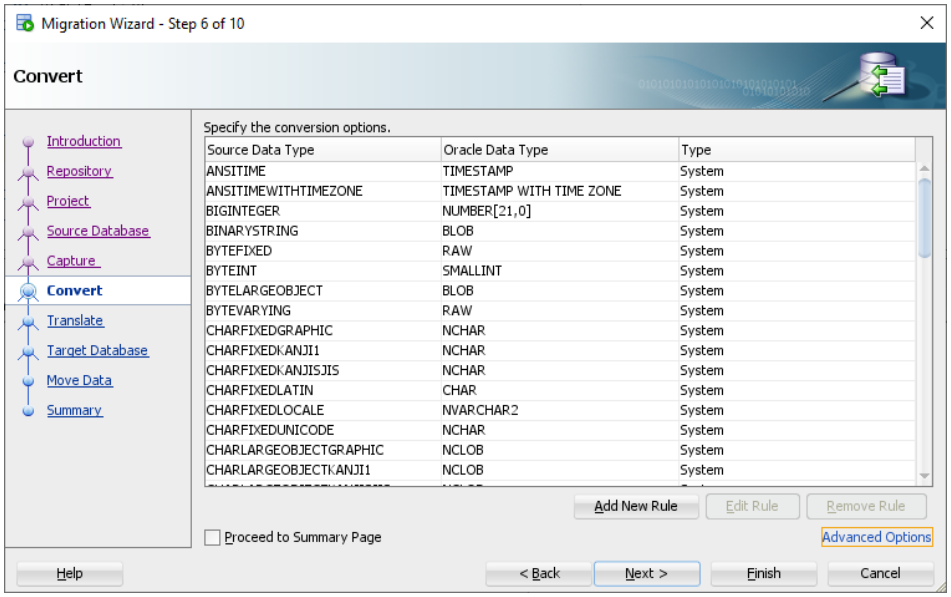
9.4. Capture Databases (Schemas).

The Capture page allows to select the databases ("schemas" in Oracle terminology) to be migrated.



9.5. Convert Datatypes.

The Convert page establishes the mapping (correlations and conversion rules) between the third party database and the Oracle database data types.



With the “Edit Rule” button the default mappings between data types can be modified.

With the “Add New Rule” button new mappings between data types can be defined.

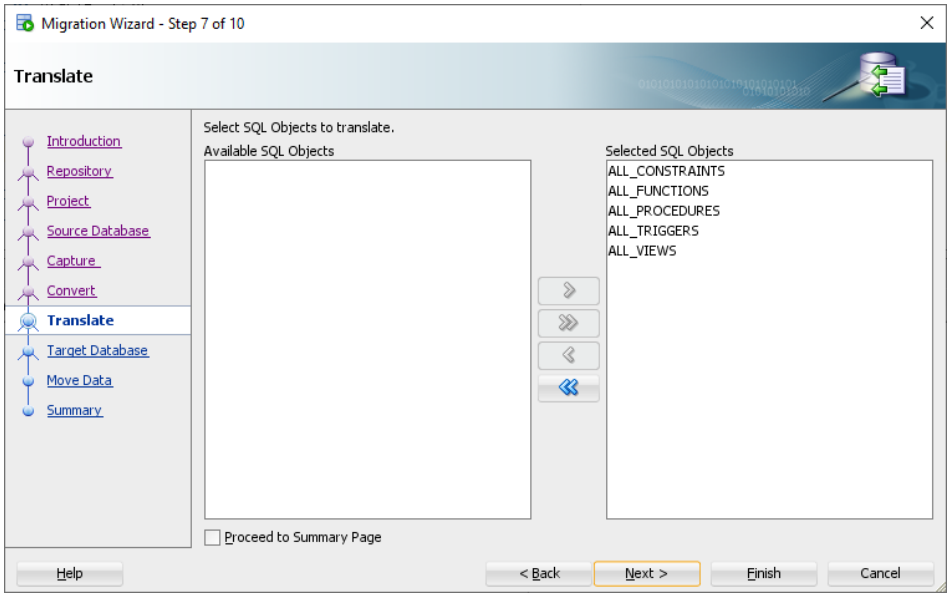
With the “Remove Rule” button mappings between data types can be removed.

The default rules are configured to work for the majority of migrations and should only be modified if there are requirements that justify it.

Changing the datatype mapping should be done by personnel with skills both in Teradata and Oracle.

9.6. Translate Objects

The Translate page specifies other objects (constraints, functions, stored procedures, triggers, views) to be translated into Oracle SQL.



Note: only views will be translated from Teradata DDL to Oracle DDL.

9.7. Target Database

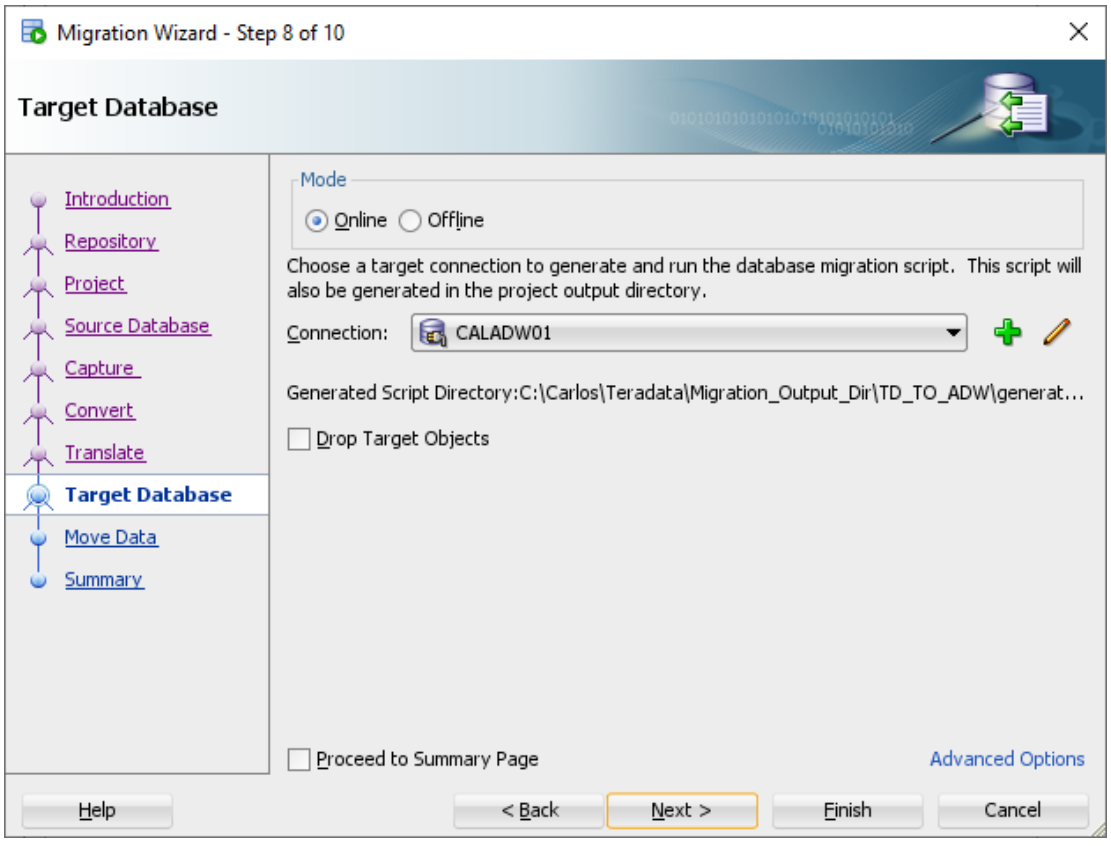
The Target Database page allows to choose the Oracle database to which the third-party databases will be migrated.

There are two options: **Online** and **Offline**.

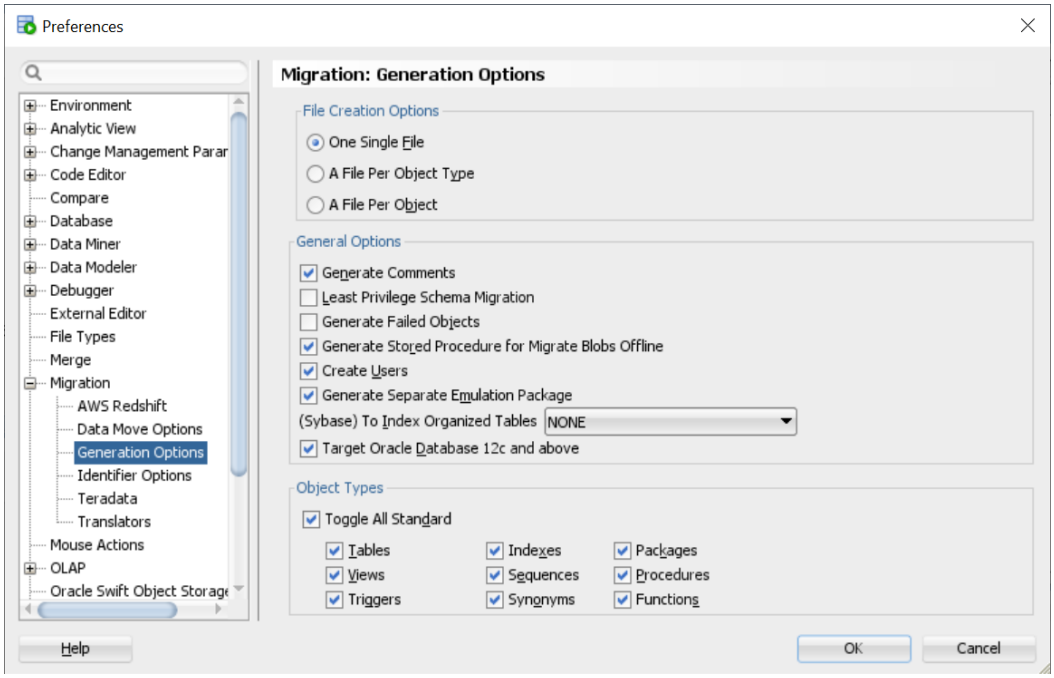
If the **Online** mode is selected, SQL Developer will execute the creation of the Oracle SCHEMAS corresponding to the Teradata Databases selected in the previous step. It will also create a special schema "EMULATION" where the main Teradata built-in functions will be created as Oracle User Defined Functions. It will also create some auxiliar ROLE for the migration and the Tables and Views in the selected schemas.

A connection to the target ADW must be provided to run the processes (SQL Developer will create scripts in the designated directory and then run them).

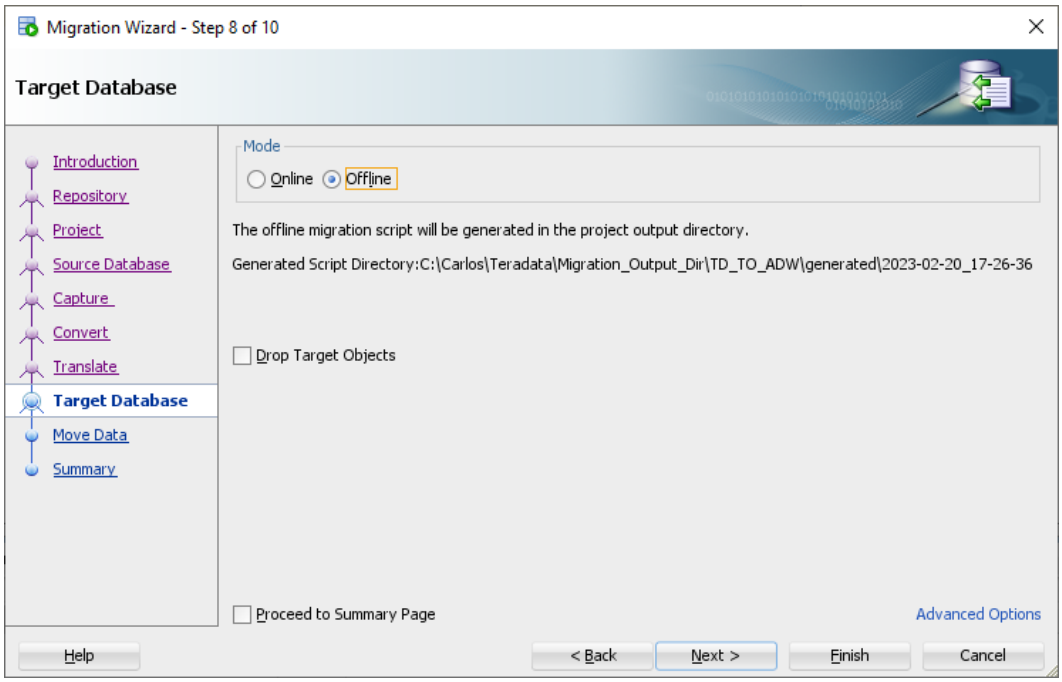
There is the option "Drop Target Objects" to drop any existing database objects in the target schemas.



The "Advanced Options" link opens the Generation Options preferences page to choose different alternatives.



If the **Offline** mode is selected, SQL Developer will only generate the scripts, but they must be manually executed later to perform the migration.

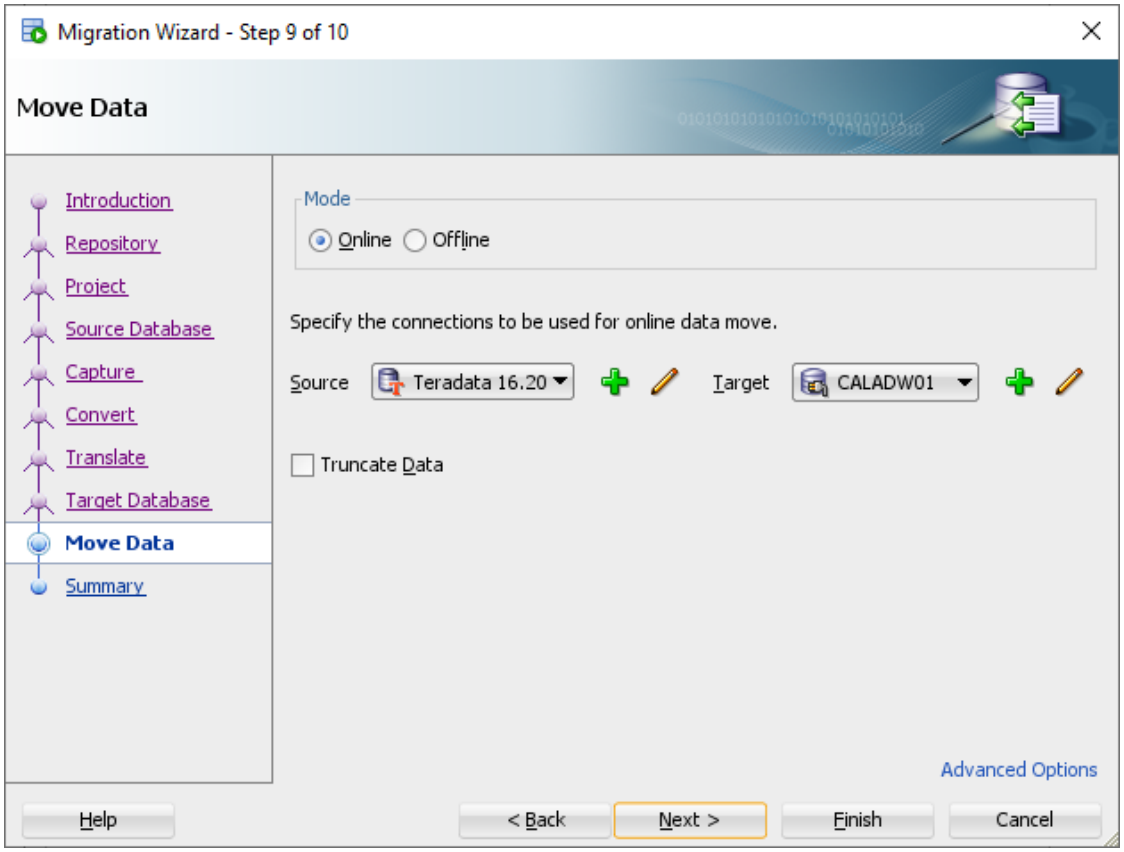


9.8. Move Data

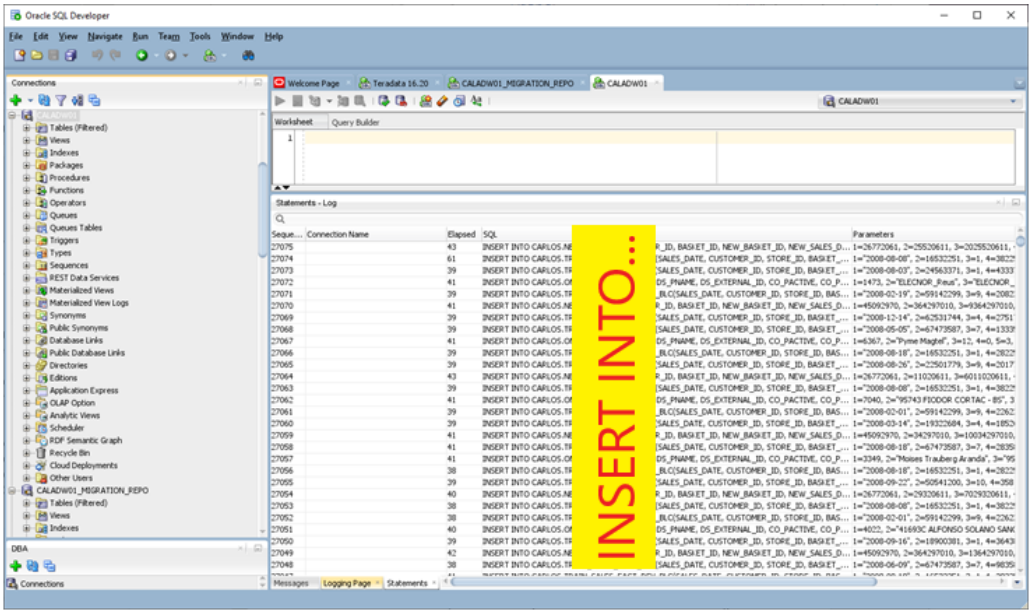
The Move Data page specifies options for the movement of the data from the Teradata tables to the ADW tables.

Online mode will cause the table data to be moved by SQL Developer. This mode needs Source and Target connections for the Teradata and ADW.

The “Truncate Data” option checked will delete all existing data in a target Oracle table that has the same name as the source Teradata table. If unchecked, the data from a source Teradata table with the same name as the corresponding target Oracle table will be appended to the target table.



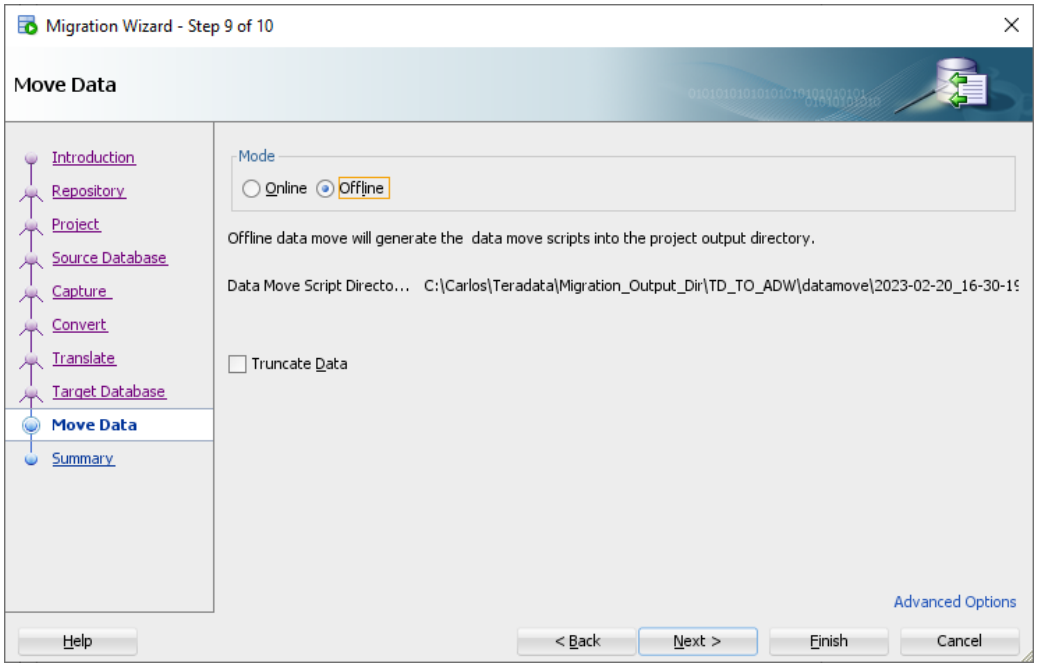
Note: **Online** mode should only be used with small migrations (few rows and few tables), since it uses SQL INSERTs to move the data from Teradata to ADW and this method is far from being optimal for large volumes of data (the typical situation in Data Warehouse environments):



Note: **Online** mode will only move the data online if the **Online** mode was selected in the previous step “Target Database”.

In **Offline** mode SQL Developer will generate scripts for all the data movement process covering different options for the export phase (extract the data from Teradata) and for the import phase (load the data into the ADW tables). The scripts will be run manually using different tools (bteq, fast export, sql loader, sql*plus…) depending on the best alternatives and the decisions that the migration will require.

The nature and details of the different scripts will be covered later in this document.

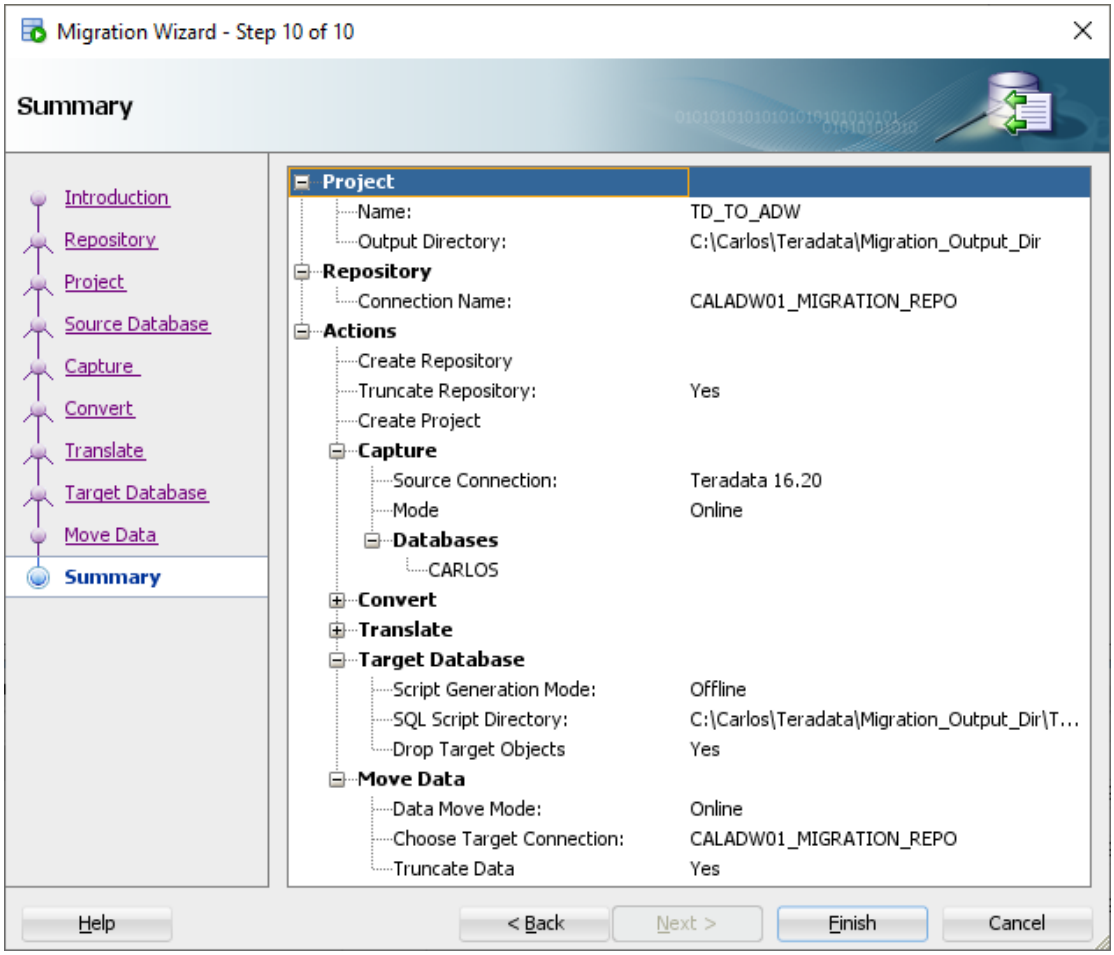


The **Offline** mode should be the typical method and the starting point to develop a real world Data Warehouse migration.

9.9. Summary Page

The Summary page shows a summary of the options selected throughout the process.

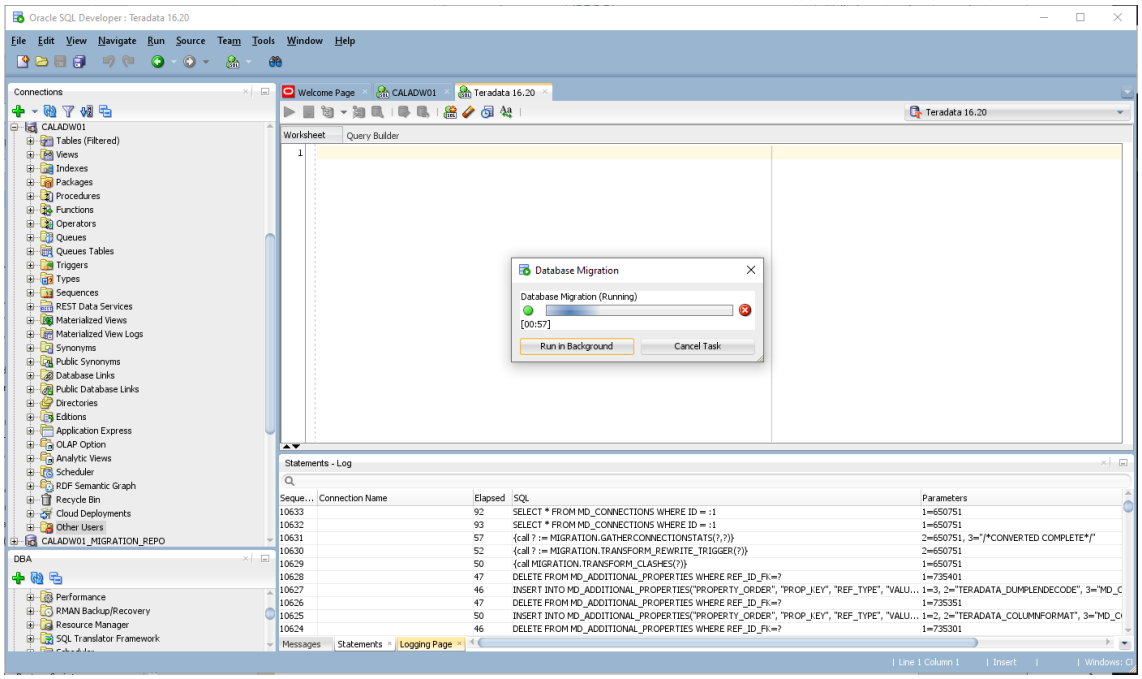
It also allows to navigate back to modify options and make last minute changes.



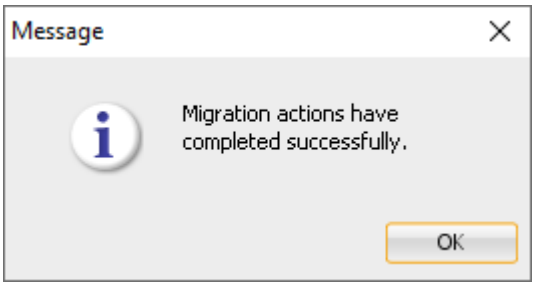
10. Running the Migration

Upon clicking on the “Finish” button the migration is run.

Depending on the different options chosen, the process can just create the scripts to run manually for the migration (offline) or can execute the migration Online.



When the process is complete, a message window informs about the success of the operations:



10.1. Online Migration

As explained before, in online migration SQL Developer will move the data from Teradata tables to Oracle tables. This mode needs Source and Target active connections (jdbc) for the Teradata database and ADW, since it uses SQL INSERTs from source tables to target tables. This **mode should only be used with small sets of data**.

10.2. Offline Migration

Instead of actually moving the data, offline migration will generate all the scripts needed to move the data from Teradata to Oracle ADW. The set of files will be placed into the project

output directory. The execution of these scripts will migrate the data from the Teradata host machine to the Oracle Autonomous Database.

The directory tree should look like this:

```
C:\Carlos\Teradata\Migration_Output_Dir>tree TD_TO_ADW /F
Folder PATH listing for volume System
Volume serial number is 00000000 B45F:A21F
C:\CARLOS\TERADATA\MIGRATION_OUTPUT_DIR\TD_TO_ADW
├───datamove
│   └───2023-01-31_14-07-42
│           oracle_bteqloader.bat
│           oracle_bteqloader.sh
│           oracle_externalbteq.bat
│           oracle_externalbteq.sh
│           oracle_externalfexp.bat
│           oracle_externalfexp.sh
│           oracle_fexpsqlldr.bat
│           oracle_fexpsqlldr.sh
│           Teradata_bteq.bat
│           Teradata_bteq.sh
│           Teradata_fexp.bat
│           Teradata_fexp.sh
│
│           └───Teradata
│                   oracle_bteqloader.bat
│                   oracle_bteqloader.sh
│                   oracle_externalbteq.bat
│                   oracle_externalbteq.sh
│                   oracle_externalfexp.bat
│                   oracle_externalfexp.sh
│                   oracle_fexpsqlldr.bat
│                   oracle_fexpsqlldr.sh
│                   Teradata_bteq.bat
│                   Teradata_bteq.sh
│                   Teradata_fexp.bat
│                   Teradata_fexp.sh
│
│                   └───CARLOS
│                           oracle_bteqloader.bat
│                           oracle_bteqloader.sh
│                           oracle_externalbteq.bat
│                           oracle_externalbteq.sh
│                           oracle_externalfexp.bat
│                           oracle_externalfexp.sh
│                           oracle_fexpsqlldr.bat
│                           oracle_fexpsqlldr.sh
│                           Teradata_bteq.bat
│                           Teradata_bteq.sh
│                           Teradata_fexp.bat
│                           Teradata_fexp.sh
│                           unload_script.bteq
│
│                           ├───bteq
│                           │       post_load.sql
│                           │       pre_load.sql
│                           │
│                           ├───control
│                           │       CARLOS.CLIENTES_ORACLE.bteq.ctl
│                           │       CARLOS.CLIENTES_ORACLE.fexp.ctl
│                           │       CARLOS.CLIENTES_ACME.bteq.ctl
│                           │       CARLOS.CLIENTES_ACME.fexp.ctl
```

```
|   |   |   |       CARLOS.DM_CUENTA_ACME.bteq.ctl
|   |   |   |       CARLOS.DM_CUENTA_ACME.fexp.ctl
|   |   |   |       CARLOS.DM_MUNICIPIOS.bteq.ctl
|   |   |   |       CARLOS.DM_MUNICIPIOS.fexp.ctl
|   |   |   |       CARLOS.M21019_38688_00252_ET.bteq.ctl
|   |   |   |       CARLOS.M21019_38688_00252_ET.fexp.ctl
|   |   |   |       CARLOS.meetings.bteq.ctl
|   |   |   |       CARLOS.meetings.fexp.ctl
|   |   |   |       CARLOS.NEW_BASKETS.bteq.ctl
|   |   |   |       CARLOS.NEW_BASKETS.fexp.ctl
|   |   |   |       CARLOS.TELCO_PROVIDER.bteq.ctl
|   |   |   |       CARLOS.TELCO_PROVIDER.fexp.ctl
|   |   |   |       CARLOS.plays.bteq.ctl
|   |   |   |       CARLOS.plays.fexp.ctl
|   |   |   |       CARLOS.PRUEBA01.bteq.ctl
|   |   |   |       CARLOS.PRUEBA01.fexp.ctl
|   |   |   |       CARLOS.PRUEBA02.bteq.ctl
|   |   |   |       CARLOS.PRUEBA02.fexp.ctl
|   |   |   |       CARLOS.PRUEBA03.bteq.ctl
|   |   |   |       CARLOS.PRUEBA03.fexp.ctl
|   |   |   |       CARLOS.PRUEBA04.bteq.ctl
|   |   |   |       CARLOS.PRUEBA04.fexp.ctl
|   |   |   |       CARLOS.PRUEBA05.bteq.ctl
|   |   |   |       CARLOS.PRUEBA05.fexp.ctl
|   |   |   |       CARLOS.PRUEBA06.bteq.ctl
|   |   |   |       CARLOS.PRUEBA06.fexp.ctl
|   |   |   |       CARLOS.PRUEBACHARSETS.bteq.ctl
|   |   |   |       CARLOS.PRUEBACHARSETS.fexp.ctl
|   |   |   |       CARLOS.PRUEBAJSON01.bteq.ctl
|   |   |   |       CARLOS.PRUEBAJSON01.fexp.ctl
|   |   |   |       CARLOS.PRUEBAJSON02.bteq.ctl
|   |   |   |       CARLOS.PRUEBAJSON02.fexp.ctl
|   |   |   |       CARLOS.PRUEBANUMBER38.bteq.ctl
|   |   |   |       CARLOS.PRUEBANUMBER38.fexp.ctl
|   |   |   |       CARLOS.PRUEBASET.bteq.ctl
|   |   |   |       CARLOS.PRUEBASET.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_1AMPMAP_01.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_1AMPMAP_01.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_MLPPI.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_MLPPI.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_PERIOD_01.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_PERIOD_01.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_RENAME02.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_RENAME02.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_TEMPORAL_04.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_TEMPORAL_04.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_TEMPORAL_05.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_TEMPORAL_05.fexp.ctl
|   |   |   |       CARLOS.PRUEBA_TRIGGER01.bteq.ctl
|   |   |   |       CARLOS.PRUEBA_TRIGGER01.fexp.ctl
|   |   |   |       CARLOS.reservations.bteq.ctl
|   |   |   |       CARLOS.reservations.fexp.ctl
|   |   |   |       CARLOS.table_newDT.bteq.ctl
|   |   |   |       CARLOS.table_newDT.fexp.ctl
|   |   |   |       CARLOS.table_newTS.bteq.ctl
|   |   |   |       CARLOS.table_newTS.fexp.ctl
|   |   |   |       CARLOS.table_oldTS.bteq.ctl
|   |   |   |       CARLOS.table_oldTS.fexp.ctl
|   |   |   |       CARLOS.CATALOGO_TARIFAS_ACME.bteq.ctl
|   |   |   |       CARLOS.CATALOGO_TARIFAS_ACME.fexp.ctl
|   |   |   |       CARLOS.TRAIN_SALES_FACT_10Y.bteq.ctl
|   |   |   |       CARLOS.TRAIN_SALES_FACT_10Y.fexp.ctl
|   |   |   |       CARLOS.TRAIN_SALES_FACT_10Y_DAY.bteq.ctl
```

```
        │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_2019.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_2019.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_2019_DAY.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_2019_DAY.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_CUSTOMER.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_CUSTOMER.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_EXP.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_EXP.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_FIN.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_FIN.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_HIVE.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_HIVE.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_IMP.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_IMP.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_REV.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_REV.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_REV_BLC.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_REV_BLC.fexp.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_TS.bteq.ctl
        │   │   │       CARLOS.TRAIN_SALES_FACT_TS.fexp.ctl
        │   │   │       CARLOS.T_XMLDOCS.bteq.ctl
        │   │   │       CARLOS.T_XMLDOCS.fexp.ctl
        │   │   │       CARLOS.T_XMLDOCS2.bteq.ctl
        │   │   │       CARLOS.T_XMLDOCS2.fexp.ctl
        │   │   │       post_load.sql
        │   │   │       pre_load.sql
        │   │   │
        │   │   ├──data
        │   │   │       CARLOS.CLIENTES_ORACLE.csv
        │   │   │       CARLOS.CLIENTES_ACME.csv
        │   │   │       CARLOS.DM_CUENTA_ACME.csv
        │   │   │       CARLOS.DM_MUNICIPIOS.csv
        │   │   │       CARLOS.M21019_38688_00252_ET.csv
        │   │   │       CARLOS.meetings.csv
        │   │   │       CARLOS.NEW_BASKETS.csv
        │   │   │       CARLOS.TELCO_PROVIDER.csv
        │   │   │       CARLOS.plays.csv
        │   │   │       CARLOS.PRUEBA01.csv
        │   │   │       CARLOS.PRUEBA02.csv
        │   │   │       CARLOS.PRUEBA03.csv
        │   │   │       CARLOS.PRUEBA04.csv
        │   │   │       CARLOS.PRUEBA05.csv
        │   │   │       CARLOS.PRUEBA06.csv
        │   │   │       CARLOS.PRUEBACHARSETS.csv
        │   │   │       CARLOS.PRUEBAJSON01.csv
        │   │   │       CARLOS.PRUEBAJSON02.csv
        │   │   │       CARLOS.PRUEBANUMBER38.csv
        │   │   │       CARLOS.PRUEBASET.csv
        │   │   │       CARLOS.PRUEBA_1AMPMAP_01.csv
        │   │   │       CARLOS.PRUEBA_MLPPI.csv
        │   │   │       CARLOS.PRUEBA_PERIOD_01.csv
        │   │   │       CARLOS.PRUEBA_RENAME02.csv
        │   │   │       CARLOS.PRUEBA_TEMPORAL_04.csv
        │   │   │       CARLOS.PRUEBA_TEMPORAL_05.csv
        │   │   │       CARLOS.PRUEBA_TRIGGER01.csv
        │   │   │       CARLOS.reservations.csv
        │   │   │       CARLOS.table_newDT.csv
        │   │   │       CARLOS.table_newTS.csv
        │   │   │       CARLOS.table_oldTS.csv
        │   │   │       CARLOS.CATALOGO_TARIFAS_ACME.csv
```

```
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019_DAY.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_CUSTOMER.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_EXP.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_FIN.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_HIVE.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_IMP.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV_BLC.csv
│   │   │   │       CARLOS.TRAIN_SALES_FACT_TS.csv
│   │   │   │       CARLOS.T_XMLDOCS.csv
│   │   │   │       CARLOS.T_XMLDOCS2.csv
│   │   │   │
│   │   │   ├──external
│   │   │   │       CARLOS.CLIENTES_ORACLE.bteq.sql
│   │   │   │       CARLOS.CLIENTES_ACME.bteq.sql
│   │   │   │       CARLOS.DM_CUENTA_ACME.bteq.sql
│   │   │   │       CARLOS.DM_MUNICIPIOS.bteq.sql
│   │   │   │       CARLOS.M21019_38688_00252_ET.bteq.sql
│   │   │   │       CARLOS.meetings.bteq.sql
│   │   │   │       CARLOS.NEW_BASKETS.bteq.sql
│   │   │   │       CARLOS.TELCO_PROVIDER.bteq.sql
│   │   │   │       CARLOS.plays.bteq.sql
│   │   │   │       CARLOS.PRUEBA01.bteq.sql
│   │   │   │       CARLOS.PRUEBA02.bteq.sql
│   │   │   │       CARLOS.PRUEBA03.bteq.sql
│   │   │   │       CARLOS.PRUEBA04.bteq.sql
│   │   │   │       CARLOS.PRUEBA05.bteq.sql
│   │   │   │       CARLOS.PRUEBA06.bteq.sql
│   │   │   │       CARLOS.PRUEBACHARSETS.bteq.sql
│   │   │   │       CARLOS.PRUEBAJSON01.bteq.sql
│   │   │   │       CARLOS.PRUEBAJSON02.bteq.sql
│   │   │   │       CARLOS.PRUEBANUMBER38.bteq.sql
│   │   │   │       CARLOS.PRUEBASET.bteq.sql
│   │   │   │       CARLOS.PRUEBA_1AMPMAP_01.bteq.sql
│   │   │   │       CARLOS.PRUEBA_MLPPI.bteq.sql
│   │   │   │       CARLOS.PRUEBA_PERIOD_01.bteq.sql
│   │   │   │       CARLOS.PRUEBA_RENAME02.bteq.sql
│   │   │   │       CARLOS.PRUEBA_TEMPORAL_04.bteq.sql
│   │   │   │       CARLOS.PRUEBA_TEMPORAL_05.bteq.sql
│   │   │   │       CARLOS.PRUEBA_TRIGGER01.bteq.sql
│   │   │   │       CARLOS.reservations.bteq.sql
│   │   │   │       CARLOS.table_newDT.bteq.sql
│   │   │   │       CARLOS.table_newTS.bteq.sql
│   │   │   │       CARLOS.table_oldTS.bteq.sql
│   │   │   │       CARLOS.CATALOGO_TARIFAS_ACME.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019_DAY.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_CUSTOMER.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_EXP.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_FIN.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_HIVE.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_IMP.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV_BLC.bteq.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_TS.bteq.sql
│   │   │   │       CARLOS.T_XMLDOCS.bteq.sql
```

```
│   │   │   │       CARLOS.T_XMLDOCS2.bteq.sql
│   │   │   │       post_load.sql
│   │   │   │       pre_load.sql
│   │   │   │
│   │   │   ├──external.fexp
│   │   │   │       CARLOS.CLIENTES_ORACLE.fexp.sql
│   │   │   │       CARLOS.CLIENTES_ACME.fexp.sql
│   │   │   │       CARLOS.DM_CUENTA_ACME.fexp.sql
│   │   │   │       CARLOS.DM_MUNICIPIOS.fexp.sql
│   │   │   │       CARLOS.M21019_38688_00252_ET.fexp.sql
│   │   │   │       CARLOS.meetings.fexp.sql
│   │   │   │       CARLOS.NEW_BASKETS.fexp.sql
│   │   │   │       CARLOS.TELCO_PROVIDER.fexp.sql
│   │   │   │       CARLOS.plays.fexp.sql
│   │   │   │       CARLOS.PRUEBA01.fexp.sql
│   │   │   │       CARLOS.PRUEBA02.fexp.sql
│   │   │   │       CARLOS.PRUEBA03.fexp.sql
│   │   │   │       CARLOS.PRUEBA04.fexp.sql
│   │   │   │       CARLOS.PRUEBA05.fexp.sql
│   │   │   │       CARLOS.PRUEBA06.fexp.sql
│   │   │   │       CARLOS.PRUEBACHARSETS.fexp.sql
│   │   │   │       CARLOS.PRUEBAJSON01.fexp.sql
│   │   │   │       CARLOS.PRUEBAJSON02.fexp.sql
│   │   │   │       CARLOS.PRUEBANUMBER38.fexp.sql
│   │   │   │       CARLOS.PRUEBASET.fexp.sql
│   │   │   │       CARLOS.PRUEBA_1AMPMAP_01.fexp.sql
│   │   │   │       CARLOS.PRUEBA_MLPPI.fexp.sql
│   │   │   │       CARLOS.PRUEBA_PERIOD_01.fexp.sql
│   │   │   │       CARLOS.PRUEBA_RENAME02.fexp.sql
│   │   │   │       CARLOS.PRUEBA_TEMPORAL_04.fexp.sql
│   │   │   │       CARLOS.PRUEBA_TEMPORAL_05.fexp.sql
│   │   │   │       CARLOS.PRUEBA_TRIGGER01.fexp.sql
│   │   │   │       CARLOS.reservations.fexp.sql
│   │   │   │       CARLOS.table_newDT.fexp.sql
│   │   │   │       CARLOS.table_newTS.fexp.sql
│   │   │   │       CARLOS.table_oldTS.fexp.sql
│   │   │   │       CARLOS.CATALOGO_TARIFAS_ACME.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_2019_DAY.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_CUSTOMER.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_EXP.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_FIN.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_HIVE.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_IMP.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_REV_BLC.fexp.sql
│   │   │   │       CARLOS.TRAIN_SALES_FACT_TS.fexp.sql
│   │   │   │       CARLOS.T_XMLDOCS.fexp.sql
│   │   │   │       CARLOS.T_XMLDOCS2.fexp.sql
│   │   │   │       post_load.sql
│   │   │   │       pre_load.sql
│   │   │   │
│   │   │   └──fexp
│   │   │           CARLOS.CLIENTES_ORACLE.fe
│   │   │           CARLOS.CLIENTES_ACME.fe
│   │   │           CARLOS.DM_CUENTA_ACME.fe
│   │   │           CARLOS.DM_MUNICIPIOS.fe
│   │   │           CARLOS.M21019_38688_00252_ET.fe
│   │   │           CARLOS.meetings.fe
│   │   │           CARLOS.NEW_BASKETS.fe
```

```
│                       CARLOS.TELCO_PROVIDER.fe
│                       CARLOS.plays.fe
│                       CARLOS.PRUEBA01.fe
│                       CARLOS.PRUEBA02.fe
│                       CARLOS.PRUEBA03.fe
│                       CARLOS.PRUEBA04.fe
│                       CARLOS.PRUEBA05.fe
│                       CARLOS.PRUEBA06.fe
│                       CARLOS.PRUEBACHARSETS.fe
│                       CARLOS.PRUEBAJSON01.fe
│                       CARLOS.PRUEBAJSON02.fe
│                       CARLOS.PRUEBANUMBER38.fe
│                       CARLOS.PRUEBASET.fe
│                       CARLOS.PRUEBA_1AMPMAP_01.fe
│                       CARLOS.PRUEBA_MLPPI.fe
│                       CARLOS.PRUEBA_PERIOD_01.fe
│                       CARLOS.PRUEBA_RENAME02.fe
│                       CARLOS.PRUEBA_TEMPORAL_04.fe
│                       CARLOS.PRUEBA_TEMPORAL_05.fe
│                       CARLOS.PRUEBA_TRIGGER01.fe
│                       CARLOS.reservations.fe
│                       CARLOS.table_newDT.fe
│                       CARLOS.table_newTS.fe
│                       CARLOS.table_oldTS.fe
│                       CARLOS.CATALOGO_TARIFAS_ACME.fe
│                       CARLOS.TRAIN_SALES_FACT_10Y.fe
│                       CARLOS.TRAIN_SALES_FACT_10Y_DAY.fe
│                       CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.fe
│                       CARLOS.TRAIN_SALES_FACT_2019.fe
│                       CARLOS.TRAIN_SALES_FACT_2019_DAY.fe
│                       CARLOS.TRAIN_SALES_FACT_CUSTOMER.fe
│                       CARLOS.TRAIN_SALES_FACT_EXP.fe
│                       CARLOS.TRAIN_SALES_FACT_FIN.fe
│                       CARLOS.TRAIN_SALES_FACT_HIVE.fe
│                       CARLOS.TRAIN_SALES_FACT_IMP.fe
│                       CARLOS.TRAIN_SALES_FACT_REV.fe
│                       CARLOS.TRAIN_SALES_FACT_REV_BLC.fe
│                       CARLOS.TRAIN_SALES_FACT_TS.fe
│                       CARLOS.T_XMLDOCS.fe
│                       CARLOS.T_XMLDOCS2.fe
│                       pre_post_load.sql
│
└───generated
    └───2023-01-31_14-07-25
            dropallschemas.sql
            master.sql
            passworddefinition.sql
            reportallstatus.sql


C:\Carlos\Teradata\Migration_Output_Dir>
```

The structure of the directories can be summarized as:

```
C:\Carlos\Teradata\Migration_Output_Dir>tree TD_TO_ADW
Folder PATH listing for volume System
Volume serial number is 00000000 B45F:A21F
C:\CARLOS\TERADATA\MIGRATION_OUTPUT_DIR\TD_TO_ADW
├───datamove
│   └───2023-01-31_14-07-42
```

```
│       └───Teradata
│           └───CARLOS
│               ├───bteq
│               ├───control
│               ├───data
│               ├───external
│               ├───external.fexp
│               └───fexp
└───generated
    └───2023-01-31_14-07-25

C:\Carlos\Teradata\Migration_Output_Dir>
```

Where:

`generated` will contain the scripts to create the necessary objects (users/schemas, roles, packages, functions, etc…) **in the Oracle ADW** for the migration to be executed.
There will be one subdirectory (noted by the timestamp) for each generation that is completed in SQL Developer.

`datamove` will contain the scripts to run with different Teradata tools (fast export `fexp`, `bteq`) to export the data from the Teradata tables and also the scripts to import these data into the Oracle tables using different Oracle tools (`sqlplus`, `sqlldr`).
There will be one subdirectory (noted by the timestamp) for each generation that is completed in SQL Developer.

The structure of the subdirectories & files in **`datamove`** is:

- **<Generation Timestamp>** (f. ex.: "2023-01-31_14-07-42"): These scripts only go down one level (to "Teradata"), execute the file with the same name, and go back up to the current directory. There are two versions each: one for Windows (.bat files) and one for Linux shell (.sh files):

  ```
  oracle_bteqloader.bat
  oracle_bteqloader.sh
  oracle_externalbteq.bat
  oracle_externalbteq.sh
  oracle_externalfexp.bat
  oracle_externalfexp.sh
  oracle_fexpsqlldr.bat
  oracle_fexpsqlldr.sh
  Teradata_bteq.bat
  Teradata_bteq.sh
  Teradata_fexp.bat
  Teradata_fexp.sh
  ```

- **`Teradata`**: These scripts do almost the same as the previous ones, but they do it once per Teradata Database/Oracle Schema that is going to be migrated (in our example there is only

one called “CARLOS”). They go down one level to the Database/Schema directory (“CARLOS”), execute the file with the same name, and go back up to current directory. There are also versions for Windows (.bat files) and for Linux shell (.sh files) for each one of them:

```
oracle_bteqloader.bat
oracle_bteqloader.sh
oracle_externalbteq.bat
oracle_externalbteq.sh
oracle_externalfexp.bat
oracle_externalfexp.sh
oracle_fexpsqlldr.bat
oracle_fexpsqlldr.sh
Teradata_bteq.bat
Teradata_bteq.sh
Teradata_fexp.bat
Teradata_fexp.sh
```

- **<Schema Name>** (f. ex.: “CARLOS”): In this directory are the scripts that actually execute the processes that export and import the data from Teradata to Oracle ADW:

  - **`oracle_bteqloader.bat`** Load data exported from Teradata with `bteq` into Oracle ADW with `sqlldr` using control files in the `.\control` directory `(*.bteq.ctl)`. It also uses `.\bteq\pre_load.sql` and `.\bteq\post_load.sql`. For Windows environments.
  - **`oracle_bteqloader.sh`** Same as the above for Linux environments.
  - **`oracle_externalbteq.bat`** Load data exported from Teradata with `bteq` into Oracle ADW using auxiliary External Tables and INSERT ... SELECT. These scripts use the `sqlplus` script files in the `.\external` directory `(*.bteq.sql)`. It also uses `.\external\pre_load.sql` and `.\external\post_load.sql`. For Windows environments.
  - **`oracle_externalbteq.sh`** Same as the above for Linux environments.
  - **`oracle_externalfexp.bat`** Load data exported from Teradata with `fexp` into Oracle ADW using auxiliary External Tables and INSERT ... SELECT. These scripts use the `sqlplus` script files in the `.\external.fexp` directory `(*.fexp.sql)`. It also uses `.\external.fexp\pre_load.sql` and `.\external.fexp\post_load.sql`. For Windows environments.
  - **`oracle_externalfexp.sh`** Same as the above for Linux environments.
  - **`oracle_fexpsqlldr.bat`** Load data exported from Teradata with `fexp` into Oracle ADW with `sqlldr` using control files in the `.\control` directory `(*.fexp.ctl)`. It also uses `.\control\pre_load.sql` and `.\control\post_load.sql`. For Windows environments.
  - **`oracle_fexpsqlldr.sh`** Same as the above for Linux environments.
  - **`Teradata_bteq.bat`** Export Teradata data with `bteq` by calling the script `unload_script.bteq`. For Windows environments.

- **Teradata_bteq.sh.** Same as the above for Linux environments.
- **Teradata_fexp.bat** Export Teradata data with `fexp` using the `.\fexp\*.fe` files. It also creates the `.\data` and `.\log` directories. Finally, it runs the script file `.\fexp\pre_post_load.sql` . to drop all the Fastexport auxiliary tables. For Windows environments.
- **Teradata_fexp.sh** Same as the above for Linux environments.
- **unload_script.bteq** Export Teradata data with `bteq (EXPORT DATA FILE)` into text files in `./data/*.dat` for each of the Teradata tables.

  The schema of the relationships between the utilities and the scripts can be summarized in the table:

		Oracle Import Utility	
		sqlldr	**sqlplus** (EXTERNAL TABLE)
Teradata Export Utility	**bteq**	**oracle_bteqloader.***	**oracle_externalbteq.***
	fexp	**oracle_fexpsqlldr.***	**oracle_externalfexp.***

On the next level in the directory tree there are the following directories:

- **bteq**: In this directory are two scripts used by `oracle_bteqloader.bat` (and `oracle_bteqloader.sh`).
  - **pre_load.sql** Disables the constraints for the Oracle ADW tables before importing the data with `sqlldr`.
  - **post_load.sql** Re-enables the constraints for the Oracle ADW tables after exporting the data with `sqlldr`.

- **control**: In this directory are the `sqlldr` control files (one for each table) for scripts used by `oracle_bteqloader.bat` (and `oracle_bteqloader.sh`) in the format `SCHEMA.TABLENAME.bteq.ctl` and by `oracle_fexpsqlldr.bat` (and `oracle_fexpsqlldr.sh`) in the format `SCHEMA.TABLENAME.fexp.ctl`.
  It also contains the files used by `oracle_fexpsqlldr.bat` (and `oracle_fexpsqlldr.sh`):
  - **pre_load.sql** Disables the constraints for the Oracle ADW tables before importing the data with `sqlldr`.
  - **post_load.sql** Re-enables the constraints for the Oracle ADW tables after importing the data with `sqlldr`.

- **data:** This directory contains .csv files (one for each table in the form of `SCHEMA.TABLENAME.csv`) with the table definitions in csv format.

- **`external`:** In this directory are the `sqlplus` files (one for each table) in the format `SCHEMA.TABLENAME.bteq.sql` used by `oracle_externalbteq.bat` (and `oracle_externalbteq.sh`) that creates the external tables for the Teradata tables that were exported with `bteq` and inserts the data in the final Oracle ADW Tables with `INSERT…SELECT`.
  It also contains the files used by `oracle_externalbteq.bat` (and `oracle_externalbteq.sh`):
  - **`pre_load.sql`** Disables the constraints for the Oracle ADW tables before importing the data with `sqlplus`. It also creates auxiliary directories and GRANTs privileges.
  - **`post_load.sql`** Re-enables the constraints for the Oracle ADW tables after importing the data with `sqlplus`. It also DROPs the auxiliary directories.

- **`external.fexp`:** In this directory are the `sqlplus` files (one for each table) in the format `SCHEMA.TABLENAME.fexp.sql` used by `oracle_externalfexp.bat` (and `oracle_externalfexp.sh`) that creates the external tables for the Teradata tables that were exported with `fexp` and inserts the data in the final Oracle ADW Tables with `INSERT…SELECT`.
  It also contains the files used by `oracle_externalfexp.bat` (and `oracle_externalfexp.sh`):
  - **`pre_load.sql`** Disables the constraints for the Oracle ADW tables before importing the data with `sqlplus`. It also creates auxiliary directories and GRANTs privileges.
  - **`post_load.sql`** Re-enables the constraints for the Oracle ADW tables after importing the data with `sqlplus.` It also DROPs the auxiliary directories.

- **`fexp`**: In this directory are the `fexp` files (one for each table) for scripts used by `Teradata_fexp.bat` (and `Teradata_fexp.sh`) in the format `SCHEMA.TABLENAME.fe.`
  It also contains the file:
  - **`pre_post_load.sql`** DROPs all the auxiliary tables used by `fexp` in the export process.

11. About the Teradata Tools (bteq, fexp).

In the previous sections we have been referring to the two Teradata tools used to export the data from the Teradata tables into files. These two tools are `bteq` (Basic TEradata Query) and `fexp` (FastEXPort).

The next sections describe briefly what the tools are, how they work, the differences between them and which one should be used depending on the scenarios.

11.1. bteq

Basic Teradata Query (`bteq`) is a general-purpose, command-line based utility to communicate and interact with Teradata Database systems.

It can be used to run DDL statements, DML statements, submit SQL queries, create Teradata native objects like Macros and Stored Procedures and, in general, to operate, administer and manage the Teradata databases.

Teradata `bteq` can be used in both batch and interactive mode.

`bteq` can be used to export data from Teradata tables into flat files or reports using the command EXPORT ("`.EXPORT DATA FILE='<file name>'`").

As a tool, and with the inherent differences, it can be compared to Oracle `sqlplus` and it is usually one of the main tools used in Teradata environments.

11.2. fexp (Fastexport)

Fastexport (`fexp`) is a command-driven utility that uses multiple sessions to quickly transfer large amounts of data from tables and views of Teradata Database to a client-based application.

Data can be exported from any table or view where the SELECT privilege has been granted.

The utility executes the Fastexport commands and Teradata SQL statements in the Fastexport job script. The Fastexport commands provide the session control and data handling specifications for the data transfer operations.

11.3. Differences between bteq and fexp

The main difference between Fastexport and bteq EXPORT is that Fastexport can export the data using multiple session connections simultaneously leveraging the total connectivity available between the client platform and the database engine and taking advantage of the native unconditional parallelism that exists in the Teradata Database systems.

On the other hand, Fastexport spends more resources on executing the query in order to prepare the blocks in such a way that when they are exported over multiple sessions they can

easily be reassembled in the right order by the client without additional sorting or processing of the rows.

Another difference is that the data files generated by Fastexport have a special binary format used by the tool, whereas the bteq generated files are regular text flat, human-readable files.

11.4. When to use each

As explained in the previous sections, Fastexport (`fexp`) should be used for large Teradata tables, whereas `bteq` can be used to export data from Teradata when the tables are smaller.

12. Executing the migration

Once we have all the pieces, it's time to execute the (offline) migration by executing the generated scripts.

As stated previously, the **`generated`** directory contains the scripts to create the necessary objects (users/schemas, roles, packages, functions, etc…) **in the Oracle ADW**.

The **`datamove`** directory contains the scripts to run with the Teradata tools to export the data from the Teradata tables and the scripts to import these data into the Oracle tables using the Oracle tools.

12.1. Generating the infrastructure in ADW

We create the infrastructure in ADW by executing the `sqlplus` script `master.sql` in the **`datamove`**`\<TIMESTAMP>` directory.
Note: Before executing `master.sql`, the Oracle ADW must be up & running

This script will ask for the passwords needed, will create the SCHEMAS, auxiliary USER "Emulation", ROLES, and will GRANT the appropriate privileges.

The script will also create a PACKAGE called TERADATA_UTILITIES with User Defined Functions (UDFs) that will emulate the native Teradata functions.

Finally, the script will create the Oracle Tables and CONSTRAINTS derived from the Teradata Tables definitions.

`master.sql` can be executed from `sqlplus`, `sqlcl` or SQL Developer.

In our example we will use `sqlplus`:

```
C:\...\TD_TO_ADW\generated\2023-01-31_14-07-25>sqlplus ADMIN@caladw01_high

SQL*Plus: Release 19.0.0.0.0 - Production on Wed May 31 16:38:04 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

ADMIN@caladw01_high>@master.sql
```

The execution of `master.sql` will start:

```
ADMIN@caladw01_high> set serveroutput on
```

```
ADMIN@caladw01_high>
ADMIN@caladw01_high> COLUMN date_time NEW_VAL filename noprint;
ADMIN@caladw01_high> SELECT to_char(systimestamp,'yyyy-mm-dd_hh24-mi-ssxff') date_time
FROM DUAL;




1 row selected.

ADMIN@caladw01_high> spool "TD_TO_ADW_&filename..log"
ADMIN@caladw01_high>
ADMIN@caladw01_high> -- Password file execution
ADMIN@caladw01_high> @passworddefinition.sql
ADMIN@caladw01_high> /* This file is used to prompt schema password */
ADMIN@caladw01_high> -- Password for 'CARLOS' user  ...
ADMIN@caladw01_high> ACCEPT CARLOS_password PROMPT "Provide the password for CARLOS:
"HIDE
Provide the password for CARLOS:
ADMIN@caladw01_high>
ADMIN@caladw01_high> -- Password for 'Emulation' user  ...
ADMIN@caladw01_high> ACCEPT Emulation_password PROMPT "Provide the password for
Emulation: "HIDE
Provide the password for Emulation:
ADMIN@caladw01_high>
ADMIN@caladw01_high>
ADMIN@caladw01_high> PROMPT Dropping Role ROLE_TD_TO_ADW ...
Dropping Role ROLE_TD_TO_ADW ...



... <snipped>



ADMIN@caladw01_high>
ADMIN@caladw01_high> spool off;
ADMIN@caladw01_high>
ADMIN@caladw01_high> COMMIT;

Commit complete.

ADMIN@caladw01_high>
```

At this point, the process has converted the Teradata "CARLOS" database to an Oracle Database SCHEMA on the Oracle ADW.

The conversion process has also created the corresponding tables in the new SCHEMA.

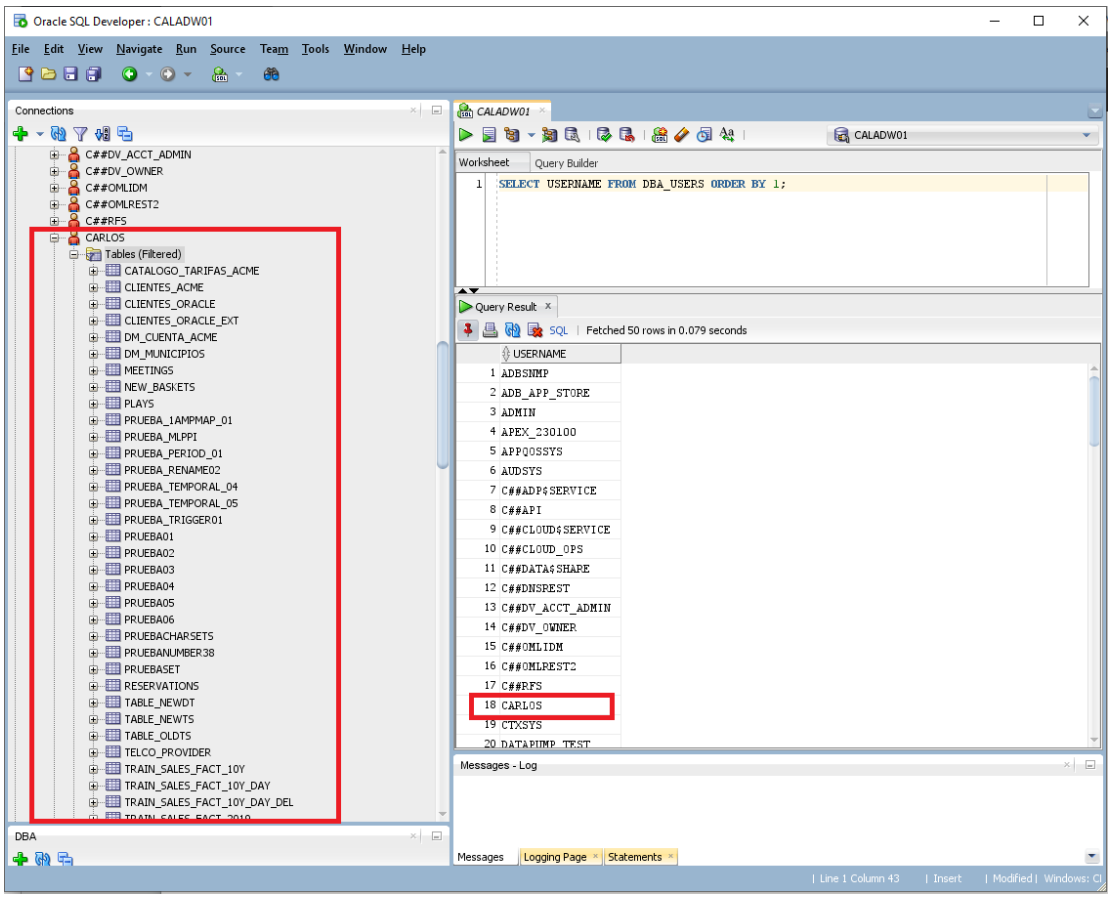
A log file with the spool of the execution of the script `master.sql` is generated in the directory in the format of **<Migration>_<Timestamp>.log**.

In our example the file is: **`TD_TO_ADW_2023-05-31_14-39-27.585192.log:`**

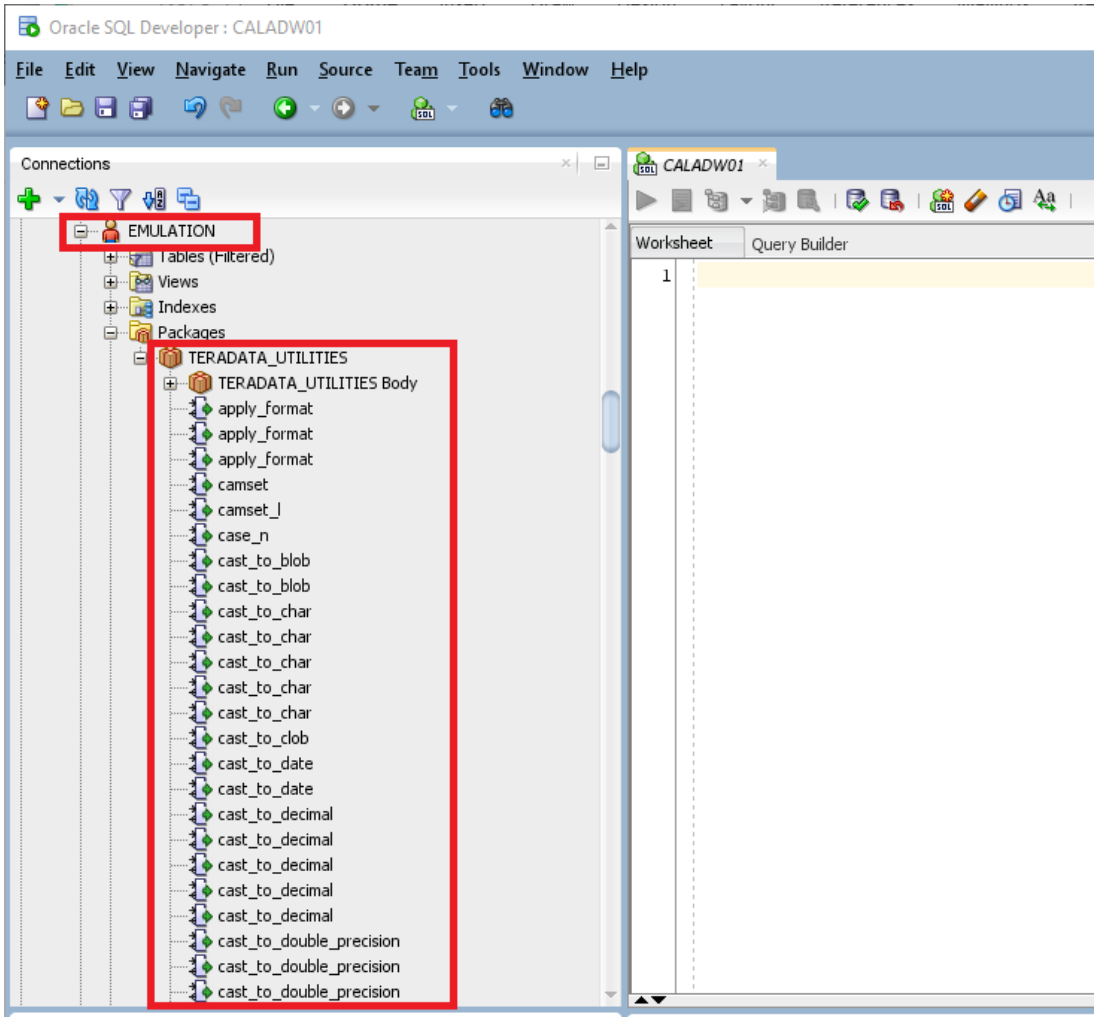
```
C:\...\TD_TO_ADW\generated\2023-01-31_14-07-25>dir
 Volume in drive C is System
 Volume Serial Number is B45F-A21F

 Directory of C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\generated\2023-01-
31_14-07-25
```

```
2023-05-31  16:39    <DIR>          .
2023-05-31  16:39    <DIR>          ..
2023-01-31  15:08               118 dropallschemas.sql
2023-01-31  15:09           170.213 master.sql
2023-01-31  15:08               274 passworddefinition.sql
2023-01-31  15:08               946 reportallstatus.sql
2023-05-31  16:40           214.361 TD_TO_ADW_2023-05-31_14-39-27.585192.log
               5 File(s)        385.912 bytes
               2 Dir(s)  44.804.014.080 bytes free

C:\...\TD_TO_ADW\generated\2023-01-31_14-07-25>
```

Upon successfully execution, the new migrated SCHEMA with all the tables should appear in the Oracle ADW:



And the generated SCHEMA “EMULATION” with the Teradata Functions as UDFs in the TERADATA_UTILITIES package:



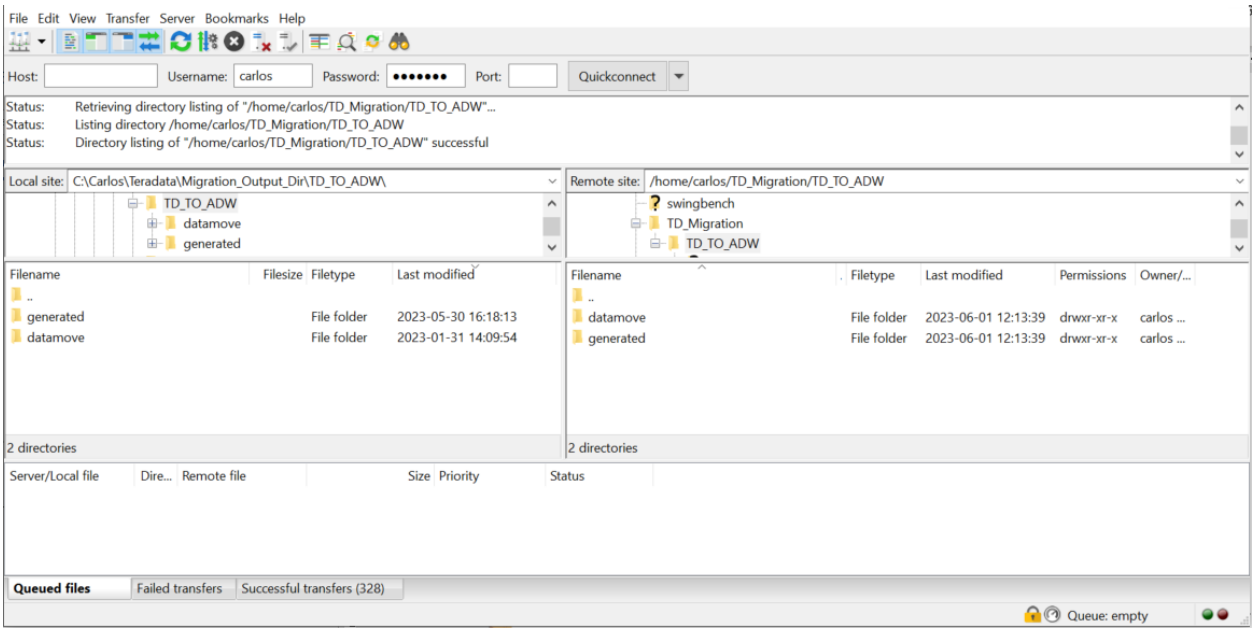
12.2. Exporting the data from Teradata

To export the data from Teradata we need the Teradata tools (`bteq`, `fexp`) installed in the machine where the migration is to be executed.

One option is to download and install the Teradata Tools and utilities package from Teradata.

The other option is to copy the **`TD_TO_ADW`** directory to the Teradata Machine or to any other machine in the source system that has the tools installed.

In our example we will use the second option. We copy the TD_TO_ADW directory tree to a Linux machine in the Teradata system that has the Teradata Tools (`bteq`, `fexp`) installed already.



Then, we connect to the Linux machine through ssh to execute the export tasks using the linux versions of the scripts (`.sh`):

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42> ll
total 48
-rw-r--r-- 1 carlos users 223 Jun  1 12:13 oracle_bteqloader.bat
-rw-r--r-- 1 carlos users 169 Jun  1 12:13 oracle_bteqloader.sh
-rw-r--r-- 1 carlos users 227 Jun  1 12:13 oracle_externalbteq.bat
-rw-r--r-- 1 carlos users 173 Jun  1 12:13 oracle_externalbteq.sh
-rw-r--r-- 1 carlos users 227 Jun  1 12:13 oracle_externalfexp.bat
-rw-r--r-- 1 carlos users 173 Jun  1 12:13 oracle_externalfexp.sh
-rw-r--r-- 1 carlos users 223 Jun  1 12:13 oracle_fexpsqlldr.bat
-rw-r--r-- 1 carlos users 169 Jun  1 12:13 oracle_fexpsqlldr.sh
drwxr-xr-x 1 carlos users 488 Jun  1 12:13 Teradata
-rw-r--r-- 1 carlos users 215 Jun  1 12:13 Teradata_bteq.bat
-rw-r--r-- 1 carlos users 161 Jun  1 12:13 Teradata_bteq.sh
-rw-r--r-- 1 carlos users 215 Jun  1 12:13 Teradata_fexp.bat
-rw-r--r-- 1 carlos users 161 Jun  1 12:13 Teradata_fexp.sh
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42>
```

12.2.1.Exporting the data using Fastexport

As explained before, Fastexport is better for large tables/databases exports and, in general, this is the tool to use as the first option.

To execute the export with Fastexport tool we run the shell script `Teradata_fexp.sh`:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42> sh
Teradata_fexp.sh x.x.x.x XXXXXX XXXXXXXX
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42>
```

The script exports Teradata data with `fexp` using the `./fexp/*.fe` files. It will create the directories `data` for the exported data and `log` for the logs for each one of the export processes. Finally, it will run the `./fexp/pre_post_load.sql` sql script.

The `data` directory with the data (`.fexp.dat`) files:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/data> ll
*.dat
-rw-r--r-- 1 carlos users       990 Jun  1 13:10 CARLOS.CLIENTES_ORACLE.fexp.dat
-rw-r--r-- 1 carlos users       560 Jun  1 13:11 CARLOS.CLIENTES_ACME.fexp.dat
-rw-r--r-- 1 carlos users  29505398 Jun  1 13:11 CARLOS.DM_CUENTA_ACME.fexp.dat
-rw-r--r-- 1 carlos users      2528 Jun  1 13:10 CARLOS.DM_MUNICIPIOS.fexp.dat
-rw-r--r-- 1 carlos users       495 Jun  1 13:10 CARLOS.meetings.fexp.dat
-rw-r--r-- 1 carlos users 127425540 Jun  1 13:13 CARLOS.NEW_BASKETS.fexp.dat
-rw-r--r-- 1 carlos users   2445312 Jun  1 13:13 CARLOS.TELCO_PROVIDER.fexp.dat
-rw-r--r-- 1 carlos users       301 Jun  1 13:13 CARLOS.plays.fexp.dat
-rw-r--r-- 1 carlos users      2118 Jun  1 13:10 CARLOS.PRUEBA01.fexp.dat
-rw-r--r-- 1 carlos users       186 Jun  1 13:13 CARLOS.PRUEBA02.fexp.dat
-rw-r--r-- 1 carlos users       363 Jun  1 13:12 CARLOS.PRUEBA03.fexp.dat
-rw-r--r-- 1 carlos users       384 Jun  1 13:13 CARLOS.PRUEBA04.fexp.dat
-rw-r--r-- 1 carlos users        90 Jun  1 13:11 CARLOS.PRUEBA05.fexp.dat
-rw-r--r-- 1 carlos users       129 Jun  1 13:13 CARLOS.PRUEBA06.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:10 CARLOS.PRUEBA_1AMPMAP_01.fexp.dat
-rw-r--r-- 1 carlos users       925 Jun  1 13:13 CARLOS.PRUEBACHARSETS.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.PRUEBAJSON01.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.PRUEBAJSON02.fexp.dat
-rw-r--r-- 1 carlos users       408 Jun  1 13:11 CARLOS.PRUEBA_MLPPI.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.PRUEBANUMBER38.fexp.dat
-rw-r--r-- 1 carlos users        80 Jun  1 13:13 CARLOS.PRUEBA_PERIOD_01.fexp.dat
-rw-r--r-- 1 carlos users  67683465 Jun  1 13:11 CARLOS.PRUEBA_RENAME02.fexp.dat
-rw-r--r-- 1 carlos users       194 Jun  1 13:11 CARLOS.PRUEBASET.fexp.dat
-rw-r--r-- 1 carlos users       495 Jun  1 13:13 CARLOS.PRUEBA_TEMPORAL_04.fexp.dat
-rw-r--r-- 1 carlos users       228 Jun  1 13:10 CARLOS.PRUEBA_TEMPORAL_05.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.PRUEBA_TRIGGER01.fexp.dat
-rw-r--r-- 1 carlos users       567 Jun  1 13:13 CARLOS.reservations.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.table_newDT.fexp.dat
-rw-r--r-- 1 carlos users        66 Jun  1 13:11 CARLOS.table_newTS.fexp.dat
-rw-r--r-- 1 carlos users        43 Jun  1 13:11 CARLOS.table_oldTS.fexp.dat
-rw-r--r-- 1 carlos users     31199 Jun  1 13:12 CARLOS.CATALOGO_TARIFAS_ACME.fexp.dat
-rw-r--r-- 1 carlos users 744518115 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.fexp.dat
-rw-r--r-- 1 carlos users 744518115 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_10Y_DAY.fexp.dat
-rw-r--r-- 1 carlos users 744518115 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_10Y.fexp.dat
-rw-r--r-- 1 carlos users  67683465 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_2019_DAY.fexp.dat
-rw-r--r-- 1 carlos users  67683465 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_2019.fexp.dat
-rw-r--r-- 1 carlos users     34720 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_CUSTOMER.fexp.dat
-rw-r--r-- 1 carlos users  11400000 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_EXP.fexp.dat
-rw-r--r-- 1 carlos users  10980000 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_FIN.fexp.dat
-rw-r--r-- 1 carlos users  67683465 Jun  1 13:10 CARLOS.TRAIN_SALES_FACT_HIVE.fexp.dat
-rw-r--r-- 1 carlos users   9500000 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_IMP.fexp.dat
-rw-r--r-- 1 carlos users  68793030 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_REV_BLC.fexp.dat
-rw-r--r-- 1 carlos users  67683465 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_REV.fexp.dat
-rw-r--r-- 1 carlos users  73601145 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_TS.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:10 CARLOS.T_XMLDOCS2.fexp.dat
-rw-r--r-- 1 carlos users         0 Jun  1 13:11 CARLOS.T_XMLDOCS.fexp.dat
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/data>
```

And the `log` directory with the log (`.fexp.log`) files:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/log> ll
*.fexp.log
total 524
-rw-r--r-- 1 carlos users  8751 Jun  1 13:10 CARLOS.CLIENTES_ORACLE.fexp.log
-rw-r--r-- 1 carlos users  8643 Jun  1 13:11 CARLOS.CLIENTES_ACME.fexp.log
-rw-r--r-- 1 carlos users 38159 Jun  1 13:11 CARLOS.DM_CUENTA_ACME.fexp.log
-rw-r--r-- 1 carlos users  8204 Jun  1 13:10 CARLOS.DM_MUNICIPIOS.fexp.log
-rw-r--r-- 1 carlos users  8612 Jun  1 13:10 CARLOS.meetings.fexp.log
```

```
-rw-r--r-- 1 carlos users  9263 Jun  1 13:13 CARLOS.NEW_BASKETS.fexp.log
-rw-r--r-- 1 carlos users 14192 Jun  1 13:13 CARLOS.TELCO_PROVIDER.fexp.log
-rw-r--r-- 1 carlos users  8004 Jun  1 13:13 CARLOS.plays.fexp.log
-rw-r--r-- 1 carlos users  7981 Jun  1 13:10 CARLOS.PRUEBA01.fexp.log
-rw-r--r-- 1 carlos users  7979 Jun  1 13:13 CARLOS.PRUEBA02.fexp.log
-rw-r--r-- 1 carlos users  7980 Jun  1 13:12 CARLOS.PRUEBA03.fexp.log
-rw-r--r-- 1 carlos users  7981 Jun  1 13:13 CARLOS.PRUEBA04.fexp.log
-rw-r--r-- 1 carlos users  7798 Jun  1 13:11 CARLOS.PRUEBA05.fexp.log
-rw-r--r-- 1 carlos users  8501 Jun  1 13:13 CARLOS.PRUEBA06.fexp.log
-rw-r--r-- 1 carlos users  8088 Jun  1 13:10 CARLOS.PRUEBA_1AMPMAP_01.fexp.log
-rw-r--r-- 1 carlos users  8419 Jun  1 13:13 CARLOS.PRUEBACHARSETS.fexp.log
-rw-r--r-- 1 carlos users  7091 Jun  1 13:11 CARLOS.PRUEBAJSON01.fexp.log
-rw-r--r-- 1 carlos users  7091 Jun  1 13:11 CARLOS.PRUEBAJSON02.fexp.log
-rw-r--r-- 1 carlos users  9085 Jun  1 13:11 CARLOS.PRUEBA_MLPPI.fexp.log
-rw-r--r-- 1 carlos users  7185 Jun  1 13:11 CARLOS.PRUEBANUMBER38.fexp.log
-rw-r--r-- 1 carlos users  8219 Jun  1 13:13 CARLOS.PRUEBA_PERIOD_01.fexp.log
-rw-r--r-- 1 carlos users 11068 Jun  1 13:11 CARLOS.PRUEBA_RENAME02.fexp.log
-rw-r--r-- 1 carlos users  8423 Jun  1 13:11 CARLOS.PRUEBASET.fexp.log
-rw-r--r-- 1 carlos users  8251 Jun  1 13:13 CARLOS.PRUEBA_TEMPORAL_04.fexp.log
-rw-r--r-- 1 carlos users  8252 Jun  1 13:10 CARLOS.PRUEBA_TEMPORAL_05.fexp.log
-rw-r--r-- 1 carlos users  8076 Jun  1 13:11 CARLOS.PRUEBA_TRIGGER01.fexp.log
-rw-r--r-- 1 carlos users  8300 Jun  1 13:13 CARLOS.reservations.fexp.log
-rw-r--r-- 1 carlos users  8226 Jun  1 13:11 CARLOS.table_newDT.fexp.log
-rw-r--r-- 1 carlos users  8323 Jun  1 13:11 CARLOS.table_newTS.fexp.log
-rw-r--r-- 1 carlos users  8145 Jun  1 13:11 CARLOS.table_oldTS.fexp.log
-rw-r--r-- 1 carlos users 11131 Jun  1 13:12 CARLOS.CATALOGO_TARIFAS_ACME.fexp.log
-rw-r--r-- 1 carlos users 11255 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.fexp.log
-rw-r--r-- 1 carlos users 11202 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_10Y_DAY.fexp.log
-rw-r--r-- 1 carlos users 11144 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_10Y.fexp.log
-rw-r--r-- 1 carlos users 11213 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_2019_DAY.fexp.log
-rw-r--r-- 1 carlos users 11156 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_2019.fexp.log
-rw-r--r-- 1 carlos users  7826 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_CUSTOMER.fexp.log
-rw-r--r-- 1 carlos users 11138 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_EXP.fexp.log
-rw-r--r-- 1 carlos users 11138 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_FIN.fexp.log
-rw-r--r-- 1 carlos users 11156 Jun  1 13:10 CARLOS.TRAIN_SALES_FACT_HIVE.fexp.log
-rw-r--r-- 1 carlos users 11142 Jun  1 13:11 CARLOS.TRAIN_SALES_FACT_IMP.fexp.log
-rw-r--r-- 1 carlos users 11579 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_REV_BLC.fexp.log
-rw-r--r-- 1 carlos users 11144 Jun  1 13:13 CARLOS.TRAIN_SALES_FACT_REV.fexp.log
-rw-r--r-- 1 carlos users 11167 Jun  1 13:12 CARLOS.TRAIN_SALES_FACT_TS.fexp.log
-rw-r--r-- 1 carlos users  6949 Jun  1 13:10 CARLOS.T_XMLDOCS2.fexp.log
-rw-r--r-- 1 carlos users  6938 Jun  1 13:11 CARLOS.T_XMLDOCS.fexp.log
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/log>
```

At this point we have the Teradata tables exported in text files.

12.2.2.Exporting the data using bteq

The other option to export data from the Teradata tables is to use `bteq` as the exporting tool. This option offers less performance, since it does not use parallelism, but it is perfectly suitable in certain circumstances and especially when the tables to migrate are not too big.

To execute the export with the `bteq` tool we run the shell script `Teradata_bteq.sh`:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42> sh
Teradata_bteq.sh x.x.x.x XXXXXX XXXXXXXX
```

The export process with `bteq` will take longer, since it will execute many SELECT sentences with a lot of formatting options to generate the text flat files.

This time the execution is showed verbosely on the screen:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42> sh
Teradata_bteq.sh x.x.x.x XXXXXX XXXXXXXX
BTEQ 16.20.00.08 (64-bit) Thu Jun  1 14:09:22 2023 PID: 18307

+---------+---------+---------+---------+---------+---------+---------+----
.logon 192.168.227.16/dbc,

 *** Logon successfully completed.
 *** Teradata Database Release is 16.20.23.01
 *** Teradata Database Version is 16.20.23.01
 *** Transaction Semantics are BTET.
 *** Session Character Set Name is 'UTF8'.

 *** Total elapsed time was 1 second.

+---------+---------+---------+---------+---------+---------+---------+----
.DEFAULTS
+---------+---------+---------+---------+---------+---------+---------+----
.FORMCHAR OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FORMAT OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FOLDLINE OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FULLYEAR ON
+---------+---------+---------+---------+---------+---------+---------+----
.HEADING ''
+---------+---------+---------+---------+---------+---------+---------+----
.PAGEBREAK OFF
+---------+---------+---------+---------+---------+---------+---------+----
.TITLEDASHES OFF
+---------+---------+---------+---------+---------+---------+---------+----
.WIDTH 65531
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.EXPORT DATA FILE='data/CARLOS.DM_MUNICIPIOS.dat'
 *** To reset export, type .EXPORT RESET
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.LARGEDATAMODE
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
SELECT
    '<EOFD>' ,
    (CAST((CAST((CASE WHEN ("CODIGO_INE_MUNICIPIO") IS NULL THEN 0 ELSE
CHARACTER_LENGTH(("CODIGO_INE_MUNICIPIO")) END) AS FORMAT '999999')) AS VARCHAR(6)) ||
("CODIGO_INE_MUNICIPIO")) ,
    (CAST((CAST((CASE WHEN ("MUNICIPIO") IS NULL THEN 0 ELSE
CHARACTER_LENGTH(("MUNICIPIO")) END) AS FORMAT '999999')) AS VARCHAR(6)) ||
("MUNICIPIO"))
    , '<EORD>'
FROM "CARLOS"."DM_MUNICIPIOS" ;

 *** Success, Stmt# 1 ActivityCount = 66
 *** Query completed. 66 rows found. 4 columns returned.
 *** Total elapsed time was 1 second.


+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.EXPORT RESET
```

```
 *** Output returned to console.
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-

... <snipped>

.DEFAULTS
+---------+---------+---------+---------+---------+---------+---------+----
.FORMCHAR OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FORMAT OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FOLDLINE OFF
+---------+---------+---------+---------+---------+---------+---------+----
.FULLYEAR ON
+---------+---------+---------+---------+---------+---------+---------+----
.HEADING ''
+---------+---------+---------+---------+---------+---------+---------+----
.PAGEBREAK OFF
+---------+---------+---------+---------+---------+---------+---------+----
.TITLEDASHES OFF
+---------+---------+---------+---------+---------+---------+---------+----
.WIDTH 65531
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.EXPORT DATA FILE='data/CARLOS.PRUEBACHARSETS.dat'
 *** To reset export, type .EXPORT RESET
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.LARGEDATAMODE
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
SELECT
    '<EOFD>' ,
    ('000012' || CAST(((CAST("ID_N" AS FORMAT '+9(11)'))) AS VARCHAR(12))) ,
    case when "C_TXT" is null then 'Y' else 'N' end,
    (CAST((CAST((CASE WHEN ("C_TXT") IS NULL THEN 0 ELSE CHARACTER_LENGTH(("C_TXT"))
END) AS FORMAT '999999')) AS VARCHAR(6)) || case when "C_TXT" is null then '' else
("C_TXT") end) ,
    case when "C_TXT_UNICODE" is null then 'Y' else 'N' end,
    (CAST((CAST((CASE WHEN ("C_TXT_UNICODE") IS NULL THEN 0 ELSE
CHARACTER_LENGTH(("C_TXT_UNICODE")) END) AS FORMAT '999999')) AS VARCHAR(6)) || case
when "C_TXT_UNICODE" is null then '' else ("C_TXT_UNICODE") end)
    , '<EORD>'
FROM "CARLOS"."PRUEBACHARSETS" ;

 *** Success, Stmt# 1 ActivityCount = 11
 *** Query completed. 11 rows found. 7 columns returned.
 *** Total elapsed time was 1 second.


+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
.EXPORT RESET
 *** Output returned to console.
+---------+---------+---------+---------+---------+---------+---------+---------+-----
----+---------+---------+---------+---------+-
 *** BTEQ exiting due to EOF on stdin.
 *** Exiting BTEQ...
 *** RC (return code) = 0
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42>
```

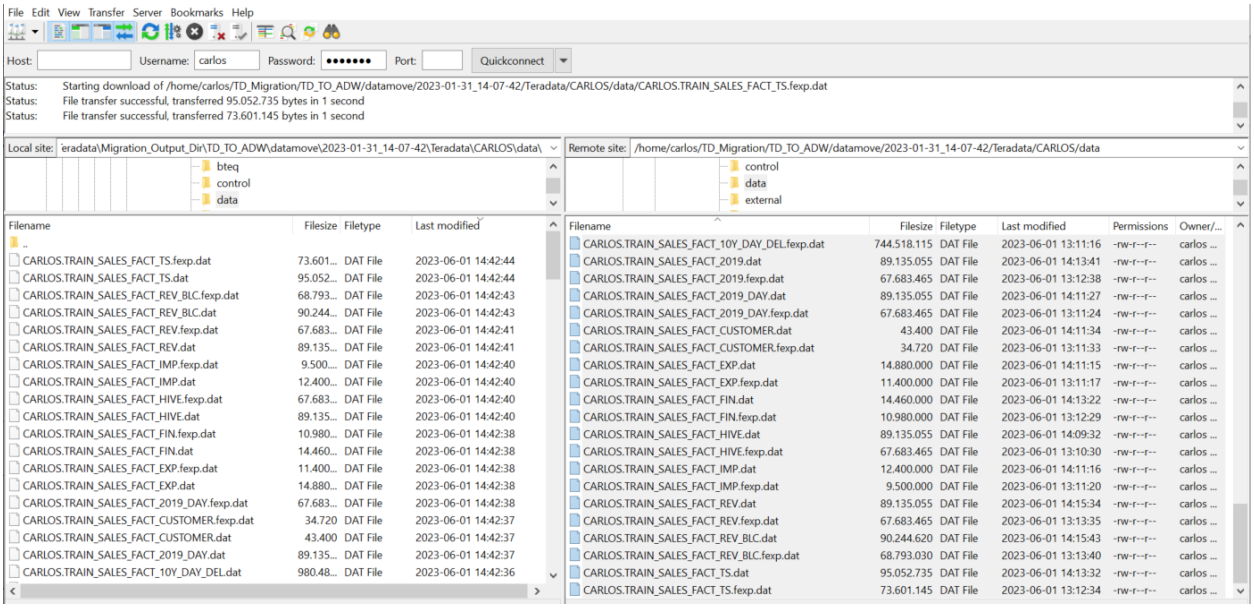
The `data` directory with the data (`.dat`) files:

```
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/data> ll
-rw-r--r-- 1 carlos users      1320 Jun  1 14:09 CARLOS.CLIENTES_ORACLE.dat
-rw-r--r-- 1 carlos users       780 Jun  1 14:11 CARLOS.CLIENTES_ACME.dat
-rw-r--r-- 1 carlos users  51902339 Jun  1 14:11 CARLOS.DM_CUENTA_ACME.dat
-rw-r--r-- 1 carlos users      3452 Jun  1 14:09 CARLOS.DM_MUNICIPIOS.dat
-rw-r--r-- 1 carlos users       675 Jun  1 14:09 CARLOS.meetings.dat
-rw-r--r-- 1 carlos users 161155830 Jun  1 14:16 CARLOS.NEW_BASKETS.dat
-rw-r--r-- 1 carlos users   3704882 Jun  1 14:16 CARLOS.TELCO_PROVIDER.dat
-rw-r--r-- 1 carlos users       401 Jun  1 14:16 CARLOS.plays.dat
-rw-r--r-- 1 carlos users      2822 Jun  1 14:09 CARLOS.PRUEBA01.dat
-rw-r--r-- 1 carlos users       228 Jun  1 14:15 CARLOS.PRUEBA02.dat
-rw-r--r-- 1 carlos users       489 Jun  1 14:13 CARLOS.PRUEBA03.dat
-rw-r--r-- 1 carlos users       524 Jun  1 14:16 CARLOS.PRUEBA04.dat
-rw-r--r-- 1 carlos users       132 Jun  1 14:11 CARLOS.PRUEBA05.dat
-rw-r--r-- 1 carlos users       194 Jun  1 14:16 CARLOS.PRUEBA06.dat
-rw-r--r-- 1 carlos users         0 Jun  1 14:09 CARLOS.PRUEBA_1AMPMAP_01.dat
-rw-r--r-- 1 carlos users      1145 Jun  1 14:16 CARLOS.PRUEBACHARSETS.dat
-rw-r--r-- 1 carlos users       878 Jun  1 14:11 CARLOS.PRUEBAJSON01.dat
-rw-r--r-- 1 carlos users       746 Jun  1 14:11 CARLOS.PRUEBAJSON02.dat
-rw-r--r-- 1 carlos users       570 Jun  1 14:11 CARLOS.PRUEBA_MLPPI.dat
-rw-r--r-- 1 carlos users         0 Jun  1 14:11 CARLOS.PRUEBANUMBER38.dat
-rw-r--r-- 1 carlos users       100 Jun  1 14:16 CARLOS.PRUEBA_PERIOD_01.dat
-rw-r--r-- 1 carlos users  89135055 Jun  1 14:11 CARLOS.PRUEBA_RENAME02.dat
-rw-r--r-- 1 carlos users       274 Jun  1 14:11 CARLOS.PRUEBASET.dat
-rw-r--r-- 1 carlos users       635 Jun  1 14:15 CARLOS.PRUEBA_TEMPORAL_04.dat
-rw-r--r-- 1 carlos users       268 Jun  1 14:09 CARLOS.PRUEBA_TEMPORAL_05.dat
-rw-r--r-- 1 carlos users         0 Jun  1 14:11 CARLOS.PRUEBA_TRIGGER01.dat
-rw-r--r-- 1 carlos users       749 Jun  1 14:16 CARLOS.reservations.dat
-rw-r--r-- 1 carlos users         0 Jun  1 14:11 CARLOS.table_newDT.dat
-rw-r--r-- 1 carlos users        80 Jun  1 14:11 CARLOS.table_newTS.dat
-rw-r--r-- 1 carlos users        57 Jun  1 14:11 CARLOS.table_oldTS.dat
-rw-r--r-- 1 carlos users     49228 Jun  1 14:13 CARLOS.CATALOGO_TARIFAS_ACME.dat
-rw-r--r-- 1 carlos users 980485605 Jun  1 14:13 CARLOS.TRAIN_SALES_FACT_10Y.dat
-rw-r--r-- 1 carlos users 980485605 Jun  1 14:15 CARLOS.TRAIN_SALES_FACT_10Y_DAY.dat
-rw-r--r-- 1 carlos users 980485605 Jun  1 14:11 CARLOS.TRAIN_SALES_FACT_10Y_DAY_DEL.dat
-rw-r--r-- 1 carlos users  89135055 Jun  1 14:13 CARLOS.TRAIN_SALES_FACT_2019.dat
-rw-r--r-- 1 carlos users  89135055 Jun  1 14:11 CARLOS.TRAIN_SALES_FACT_2019_DAY.dat
-rw-r--r-- 1 carlos users     43400 Jun  1 14:11 CARLOS.TRAIN_SALES_FACT_CUSTOMER.dat
-rw-r--r-- 1 carlos users  14880000 Jun  1 14:11 CARLOS.TRAIN_SALES_FACT_EXP.dat
-rw-r--r-- 1 carlos users  14460000 Jun  1 14:13 CARLOS.TRAIN_SALES_FACT_FIN.dat
-rw-r--r-- 1 carlos users  89135055 Jun  1 14:09 CARLOS.TRAIN_SALES_FACT_HIVE.dat
-rw-r--r-- 1 carlos users  12400000 Jun  1 14:11 CARLOS.TRAIN_SALES_FACT_IMP.dat
-rw-r--r-- 1 carlos users  90244620 Jun  1 14:15 CARLOS.TRAIN_SALES_FACT_REV_BLC.dat
-rw-r--r-- 1 carlos users  89135055 Jun  1 14:15 CARLOS.TRAIN_SALES_FACT_REV.dat
-rw-r--r-- 1 carlos users  95052735 Jun  1 14:13 CARLOS.TRAIN_SALES_FACT_TS.dat
-rw-r--r-- 1 carlos users       948 Jun  1 14:09 CARLOS.T_XMLDOCS2.dat
-rw-r--r-- 1 carlos users       948 Jun  1 14:11 CARLOS.T_XMLDOCS.dat
carlos@linux-wt5k:~/TD_Migration/TD_TO_ADW/datamove/2023-01-31_14-07-42/Teradata/CARLOS/data>
```

At this point we have the Teradata tables exported in text files.

12.3. Importing the data into the Oracle ADW

To import the data from the generated `.dat` files we have to copy them from the Linux machine to the local Windows one into the …`\Teradata\CARLOS\data\` directory:



In the previous section “Offline Migration” we discussed the relationships between the export and import utilities and the scripts:

		Oracle Import Utility	
		sqlldr	**sqlplus** (EXTERNAL TABLE)
Teradata Export Utility	**bteq**	**oracle_bteqloader.***	**oracle_externalbteq.***
	fexp	**oracle_fexpsqlldr.***	**oracle_externalfexp.***

SQL Developer generates two hypothetical ways to load the exported Teradata files into an Oracle Database: using Oracle SQL LOADER (`sqlldr`) and using auxiliary EXTERNAL TABLES to INSERT…SELECT the data into the Oracle ADW Tables.

But for the migration to Oracle ADW the EXTERNAL TABLES methods (`externalbteq.*`, `externalfexp.*`) are not really an option, since the external data files must be stored outside the Oracle database, either on the database server or on a remote file server accessible to the database server, and in Oracle Autonomous Database the user has no access to the underlying server filesystem.

It’s true that the scripts could be modified to use Object Storage or even NFS mounted volumes with the Autonomous Database to enable the use of the EXTERNAL TABLES (and directories database objects), but this is a manual task that requires skills and a lot of code changing of the SQL Developer generated scripts.

This is a consequence of the fact that the SQL Developer migration process is intended for ANY Oracle Database, and the migration to Autonomous Database (ADW) is somewhat a special case of migration. So, we'll leave the EXTERNAL TABLES option out of the loading methods explained in this section.

12.3.1.Loading the data with `oracle_fexpsqlldr.bat`

`oracle_fexpsqlldr.bat` (`oracle_fexpsqlldr.sh` in Linux) must be used to load the data files that were exported from Teradata using the Fastexport (`fexp`) utility. These files are in the **`data`** directory with the name `<SCHEMA>.<TABLE NAME>.fexp.dat`

We only have to execute the `oracle_fexpsqlldr.bat` file from the command line, passing the database tnsname, user and password as arguments:

```
C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-
42\Teradata\CARLOS>oracle_fexpsqlldr.bat caladw01_high ADMIN XXXXXXXXXXXXXXXX

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 2 13:19:17 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Last Successful login time: Thu Jun 01 2023 18:28:19 +02:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.


ADMIN@caladw01_high> ADMIN@caladw01_high> Disconnected from Oracle Database 19c Enterprise
Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0
```

```
SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:30 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 43
Commit point reached - logical record count 66

Table CARLOS.DM_MUNICIPIOS:
  66 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.DM_MUNICIPIOS.bteq.log
for more information about the load.


. . . <snipped>



SQL*Loader: Release 19.0.0.0.0 - Production on Fri Jun 2 13:19:24 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 15
Commit point reached - logical record count 30
Commit point reached - logical record count 45
Commit point reached - logical record count 60
Commit point reached - logical record count 75
Commit point reached - logical record count 90
Commit point reached - logical record count 105
Commit point reached - logical record count 120
Commit point reached - logical record count 135


. . . <snipped>


Commit point reached - logical record count 49962
Commit point reached - logical record count 49977
Commit point reached - logical record count 49992
Commit point reached - logical record count 50000

Table CARLOS.TRAIN_SALES_FACT_IMP:
  50000 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.TRAIN_SALES_FACT_IMP.sqlldr.fexp.log
for more information about the load.


. . . <snipped>


SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 18:28:14 2023
Version 19.3.0.0.0

Table CARLOS.PRUEBACHARSETS:
  11 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.PRUEBACHARSETS.bteq.log
for more information about the load.


SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 2 13:23:39 2023
Version 19.3.0.0.0
```

```
Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Fri Jun 02 2023 13:23:37 +02:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high> ADMIN@caladw01_high> Disconnected from Oracle Database 19c Enterprise
Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42\Teradata\CARLOS>
```

12.3.2.Loading the data with `oracle_bteqloader.bat`

`oracle_bteqloader.bat` (`oracle_bteqloader.sh` in Linux) must be used to load the data files that were exported from Teradata using the `bteq` utility. These files are in the **`data`** directory with the name `<SCHEMA>.<TABLE NAME>.dat`

We only have to execute the `oracle_bteqloader.bat` file from the command line, passing the database tnsname, user and password as arguments:

```
C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-
42>oracle_bteqloader.bat caladw01_high ADMIN XXXXXXXXXXXXXXX

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42>IF "caladw01_high"
== "" goto INPUT_ERROR

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42>IF "ADMIN" == ""
goto INPUT_ERROR

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42>pushd Teradata   &
call oracle_bteqloader.bat caladw01_high ADMIN ADMINPassword123456 & popd
```

```
C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42\Teradata>IF
"caladw01_high" == "" goto INPUT_ERROR

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42\Teradata>IF
"ADMIN" == "" goto INPUT_ERROR

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42\Teradata>pushd
CARLOS   & call oracle_bteqloader.bat caladw01_high ADMIN ADMINPassword123456 & popd

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:27 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Last Successful login time: Thu Jun 01 2023 16:45:51 +02:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high> ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>

ADMIN@caladw01_high> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0
- Production
Version 19.19.0.1.0

SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:30 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 43
Commit point reached - logical record count 66

Table CARLOS.DM_MUNICIPIOS:
  66 Rows successfully loaded.
```

```
Check the log file:
  ..\log\CARLOS.DM_MUNICIPIOS.bteq.log
for more information about the load.

SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:37 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 2

Table CARLOS.PRUEBA_TEMPORAL_05:
  2 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.PRUEBA_TEMPORAL_05.bteq.log
for more information about the load.

SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:40 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 7
Commit point reached - logical record count 14
Commit point reached - logical record count 21
Commit point reached - logical record count 28
Commit point reached - logical record count 35
Commit point reached - logical record count 42
Commit point reached - logical record count 49
Commit point reached - logical record count 56
Commit point reached - logical record count 63
Commit point reached - logical record count 70
Commit point reached - logical record count 77
Commit point reached - logical record count 84
Commit point reached - logical record count 91
Commit point reached - logical record count 98
Commit point reached - logical record count 105


. . . <snipped>


Commit point reached - logical record count 50000

Table CARLOS.DM_CUENTA_ACME:
  50000 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.DM_CUENTA_ACME.bteq.log
for more information about the load.

SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 17:06:37 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

. . . <snip>

SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 18:24:21 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

ORA-28002: the password will expire within 29 days
Path used:      Conventional
Commit point reached - logical record count 4
Commit point reached - logical record count 8
```

```
Commit point reached - logical record count 12
Commit point reached - logical record count 16
Commit point reached - logical record count 20
Commit point reached - logical record count 24
Commit point reached - logical record count 28
Commit point reached - logical record count 32
Commit point reached - logical record count 36
Commit point reached - logical record count 40
Commit point reached - logical record count 44
Commit point reached - logical record count 48
Commit point reached - logical record count 52

. . . <snipped>

Commit point reached - logical record count 11537
Commit point reached - logical record count 11538

Table CARLOS.TELCO_PROVIDER:
  11538 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.TELCO_PROVIDER.bteq.log
for more information about the load.


. . . <snipped>


SQL*Loader: Release 19.0.0.0.0 - Production on Thu Jun 1 18:28:14 2023
Version 19.3.0.0.0

Table CARLOS.PRUEBACHARSETS:
  11 Rows successfully loaded.

Check the log file:
  ..\log\CARLOS.PRUEBACHARSETS.bteq.log
for more information about the load.

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 1 18:28:18 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Thu Jun 01 2023 18:28:16 +02:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

ADMIN@caladw01_high>ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high> ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
```

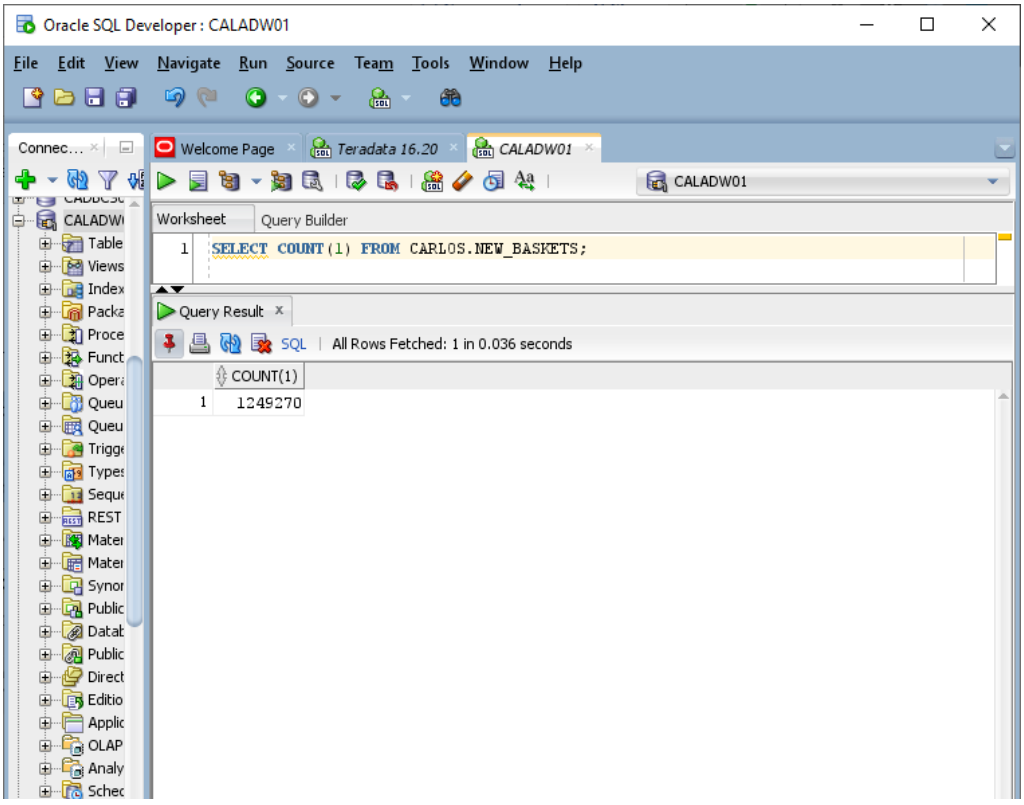
```
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high>
Table altered.

ADMIN@caladw01_high> ADMIN@caladw01_high> Disconnected from Oracle Database 19c Enterprise
Edition Release 19.0.0.0.0 - Production
Version 19.19.0.1.0

C:\Carlos\Teradata\Migration_Output_Dir\TD_TO_ADW\datamove\2023-01-31_14-07-42\Teradata\CARLOS>
```

Checking the data integrity after the migration is essential to ensure accuracy and completeness of the migrated data. Once the process is finished, we can check the tables in Oracle ADB and compare them to the Teradata ones to look for discrepancies. For example, executing row counts in Source and Target DBs, selecting sample data sets or performing data comparisons using SQL queries.

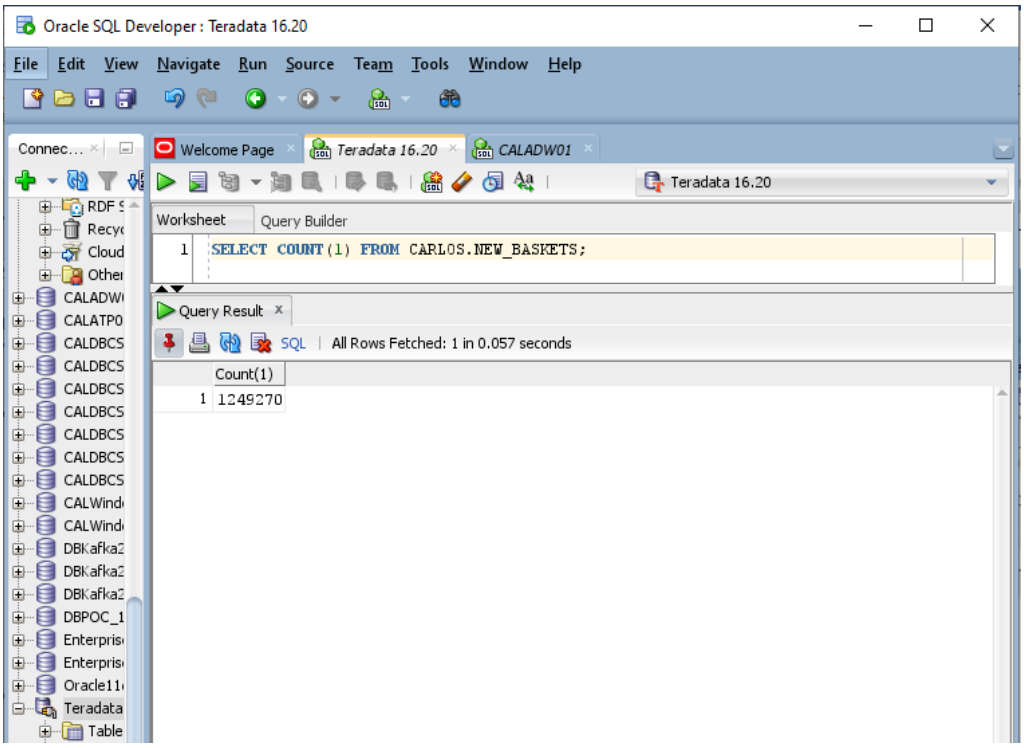
The complexity of the data validation may vary depending on the nature of the migration.

Some simple checks may be as following:

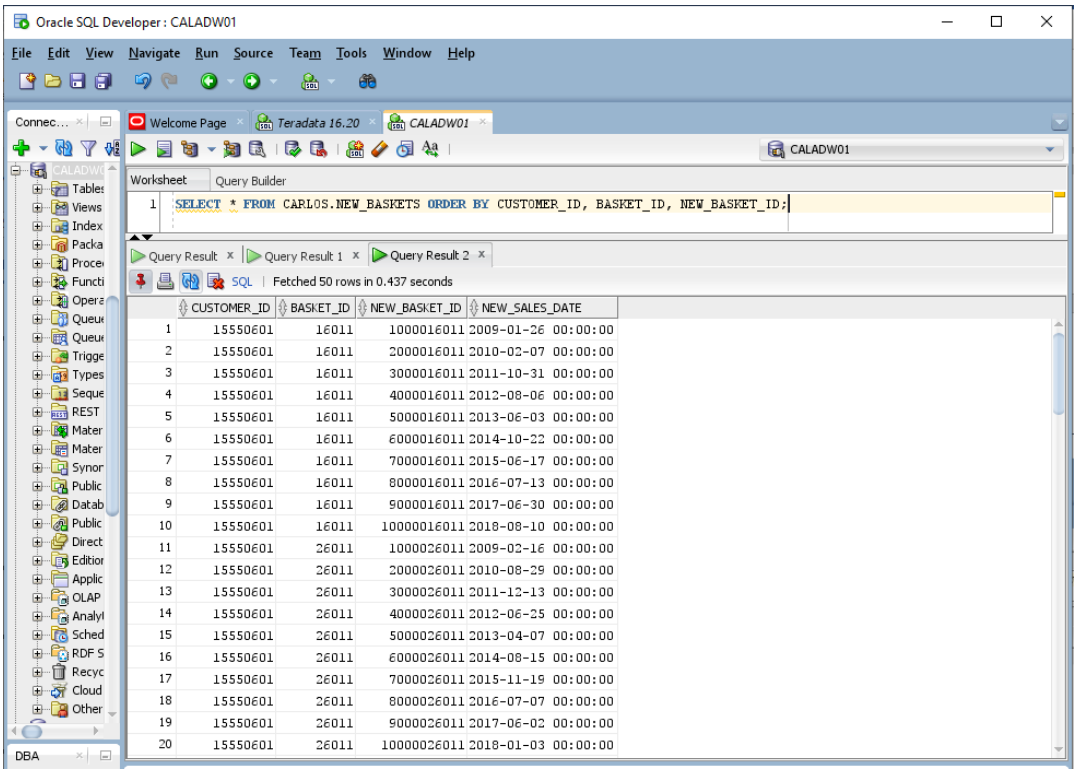
Check the number of rows in ADW:



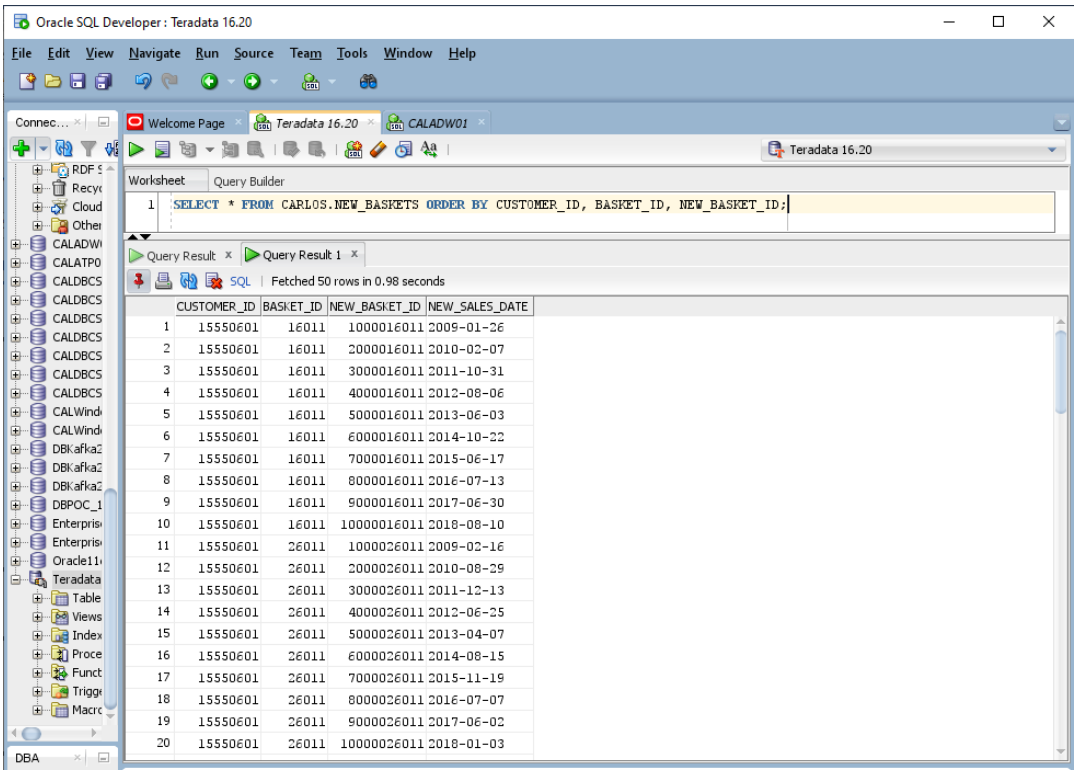
Check the number of rows in Teradata:



Check the actual data in the tables in ADB:



Check the actual data in the tables in Teradata:



13. Some final thoughts

In this document we have gone through the process and different options of a migration from Teradata to Oracle Autonomous Datawarehouse (ADW) using the features provided by SQL Developer to assist.

Now we will briefly comment on some of the aspects regarding the objects that are migrated (and how) and the aspects regarding the performance of the total process.

13.1. Regarding the objects that are migrated

The SQL Developer migration will basically migrate Tables and Views (and their data). Also, some Constraints and Indexes will be inferred from the Teradata tables definitions (more specifically, the primary indexes definitions). However, there are many objects and features that may not be directly migrated or may require manual intervention, since Teradata supports some SQL features and functions that may not have direct equivalents in Oracle. Also, functions or procedures written using Teradata-specific syntax or functions need to be rewritten in Oracle SQL or PL/SQL. This applies for the SQL code for the views as well: some Teradata SQL extensions that might be used in the views are not available in Oracle (for example: the QUALIFY clause is profusely used in Teradata SQL).

It's essential to carefully analyze the Teradata Database and the specific requirements for the migration to identify any objects or features that may require additional attention or manual handling during the migration process (e.g: secondary indexes, primary keys/foreign keys). A good method is to examine the Teradata Data Model (physical/logical) and re-create those deemed appropriate.

It's important to note that migrating complex databases may require additional steps, like the ones already mentioned: data transformations, procedural language differences, and specific Teradata or Oracle features that don't have direct equivalents, so it is highly advisable to involve database administrators and experts to ensure a smooth and successful migration process.

There is an appendix in this document commenting on the equivalences and differences between Teradata and Oracle objects.

13.2. Regarding the performance in the migration process

As we've seen, the typical migration for a real-life Teradata database would be an Offline one. From the performance point of view, the ideal method would be using the specialized Teradata export tool Fastexport (`fexp`), this is: using **`Teradata_fexp.*`** (and the `./fexp/*.fe` files) for exporting the data from Teradata and **`fexpsqlldr.*`** (and the `*.fexp.ctl` files) for the import phase in ADW (this may vary based on the circumstances, though).

Another important aspect regarding the performance is that the offline migration process is script-based. This means that, although it may be a laborious task, the generated scripts can be modified to achieve better performance.

For example, modifying the `sqlldr` parameters to allow direct path operations (direct=true), increase parallelism (`PARALLEL=TRUE`), or reduce the number of "commit" operations (`ROWS=n`) could speed the import phase into ADW.

The same can be done on the Teradata side. For example, the Fastexport (`fexp`) scripts can be modified to increase the number of sessions (`SESSIONS` *`n`*) to improve the performance of the export tasks.

Nevertheless, if decided, these modifications should be done upon thorough analysis and by skilled personnel.

Finally, both the export and import processes could be optimized if they were run as close to the respective systems as possible: execute the Teradata tasks from a host locally connected to the Database and execute the Oracle tasks within OCI infrastructure (i.e.: from a OCI compute instance with the Oracle tools installed) could speed the whole process up dramatically, since the network factor (bandwidth, throughput, latency) would be minimized by using the internal hi-speed OCI networking infrastructure.

14. References

Some useful references regarding the migration from Teradata to Oracle Autonomous Datawarehouse can be found here:

Architecture Center / Reference Architectures: Migrate a Teradata database to an Autonomous Data Warehouse: https://docs.oracle.com/en/solutions/migrate-teradata/index.html

SQL Developer User's Guide: Migrating Third-Party Databases: https://docs.oracle.com/en/database/oracle/sql-developer/23.1/rptug/migrating-third-party-databases.html

SQL Developer User's Guide: Before Migrating From Teradata: https://docs.oracle.com/en/database/oracle/sql-developer/23.1/rptug/migrating-third-party-databases.html#GUID-AFDBDC24-4983-439A-9BF5-CC23D298E7BA

15. Appendix: Equivalence and differences between Oracle and Teradata Objects

When comparing the references and equivalences between Oracle database objects and Teradata database objects, it's important to note that Oracle and Teradata are two distinct relational database management systems (RDBMS) with their own unique features and implementations. While there may be similarities in certain concepts, the specific references and relationships can vary between the two.

Tables:

Tables are the fundamental database objects used to store structured data in both Oracle and Teradata.

Tables can be referenced by other database objects such as views, secondary indexes, join indexes, stored procedures, user-defined functions, materialized views, macros, volatile tables, global temporary tables, etc…

Tables ("*heap tables*") are equivalent in both systems, representing the primary storage units for data.

Nonetheless, both Teradata and Oracle have special kind of tables to serve specific purposes. For example, in Teradata there are NoPI Tables (tables without primary index) and in Oracle there are Index Organized tables and Clustered tables.

Partitioned Tables:

Both Oracle and Teradata systems provide partitioned tables functionality.

In Teradata the partitioning is based on the distribution and ordering of the data (through a primary index definition). The table data share a common storage space.

In Oracle the partitioning is based on the storage of the data. The partitioned table is based on partitions that have an independent storage unit for each (segment).

External Tables:

External tables are a special type of tables that allows the access and query of data stored outside the database in files with several formats (csv, json, parquet, avro…).

Both Oracle and Teradata systems provide external tables functionality.

Views:

Views in both Oracle and Teradata are virtual tables derived from underlying tables or other views.

Views provide a customized or filtered view of the data and can reference tables, join indexes, stored procedures, user-defined functions, materialized views, macros, volatile tables, and global temporary tables in both systems.

Views are equivalent in both systems, providing customized or filtered views of the data.

Primary Indexes (Teradata) / Primary Keys (Oracle):

Primary indexes in Teradata and primary keys in Oracle can serve similar purposes.

They ensure the uniqueness of data within a table and provide a quick access mechanism.

Primary indexes/keys are used as references for maintaining data integrity and enforcing relationships between tables.

Nonetheless, there are important differences between them:

In Oracle, a primary key is a constraint that can be defined on one or more columns of a table. In Teradata, a primary index is defined on a table as a mechanism for physically organizing and distributing the data.

In Oracle, primary keys are defined explicitly as constraints during table creation or through ALTER TABLE statements. Teradata primary indexes are defined explicitly during table creation with the PRIMARY INDEX clause and cannot be modified.

In Oracle, primary keys do not enforce any specific ordering of the data. In Teradata, the primary index determines the physical order of the rows in the table based on the (hash) index values. Teradata primary indexes need to be explicitly defined during table creation.

In Teradata, the primary index plays a crucial role in data distribution across AMPs (Access Module Processors) for parallel processing. Oracle primary keys do not have a direct impact on data distribution.

In Oracle, primary keys can improve query performance by using the index for data retrieval. In Teradata, the primary index affects both data retrieval and data distribution, impacting the overall performance of queries and join operations.

Secondary Indexes:

Secondary indexes in both Oracle and Teradata are additional indexes created on tables to improve query performance.

They provide alternative access paths to the data and can be used as references for faster data retrieval.

In Oracle, indexes are b-tree structures, whereas in Teradata they have a structure similar to tables (often called "*subtables*").

In Teradata, although they operate in almost the same way, unique and non-unique secondary indexes are implemented differently.

Join Indexes (Teradata):

Join indexes in Teradata are specifically designed to optimize join operations between large tables.

Join indexes are materialized views that store pre-joined data, improving query performance by reducing the need for on-the-fly joins.

Join indexes can reference one or more base tables in Teradata.

Join indexes are similar to materialized views in Oracle, providing pre-joined data for improved query performance.

Hash Indexes (Teradata):

Hash indexes in Teradata are specific types of indexes used to support equality-based queries.

Hash indexes use a hashing algorithm to map the index value to the specific row and provide an alternative distribution of the data for a table.

Oracle does not have an equivalent hash index concept, but it could somehow be related to Materialized View.

Stored Procedures:

Stored procedures are precompiled sets of SQL statements stored in the database.

Stored procedures can reference tables, views, join indexes, materialized views, macros, volatile tables, global temporary tables, or other database objects in both Oracle and Teradata.

They are used for encapsulating business logic and performing complex operations on the data.

Stored procedures are equivalent in both systems (using each own syntax), though there is a huge difference in features and functionalities on the Oracle side.

User-defined Functions:

User-defined functions (UDFs) in both Oracle and Teradata allow users to create custom functions.

UDFs can reference tables, views, join indexes, stored procedures, materialized views, macros, volatile tables, global temporary tables, or other database objects.

They are used to perform specific computations or transformations on the data and can be called within SQL statements.

Whereas in Oracle the UDFs are written in PL/SQL code, in Teradata they are typically written in C/C++.

Materialized Views (Oracle):

Materialized views in Oracle are precomputed and stored summaries of data from one or more tables.

They are used to improve query performance by allowing faster access to aggregated or summarized data.

Teradata does not have an exact equivalent to materialized views but join indexes (and even hash indexes) can serve a similar purpose.

Macros (Teradata):

Macros in Teradata are pre-defined sets of SQL statements that can be executed as a single unit.

Macros can reference tables, views, join indexes, or other database objects in Teradata.

They are used for code reusability and simplifying complex queries or tasks.

In the latest versions, Oracle provides SQL Macros functionality.

Volatile Tables (Teradata):

Volatile tables in Teradata are temporary tables that are created and exist only for the duration of a session or transaction.

They are used for temporary storage or intermediate results during query processing.

In the latest versions, Oracle provides Private Temporary Tables, which give the same functionality.

Global Temporary Tables:

Global temporary tables are temporary tables that are shared across sessions.

The definition of Global Temporary table is stored in the data dictionary and the table can be used by many users/sessions. But the data loaded into the global temporary table is local to the session and is retained only during the session.

Synonyms (Oracle):

A synonym is an alias or friendly name for the database objects (such as tables, views, stored procedures, functions, and packages). Synonyms are used to resolve references to an object if the object is not fully qualified (prefixed by a schema name).

Teradata does not have an exact equivalent, but these kind of references for tables can be resolved using view definitions.

While there are similarities and equivalences between certain database objects in Oracle and Teradata, it's essential to consider the specific functionalities, syntax, and guidelines provided by each database system for accurate usage and implementation.