



# Autonomous Data Warehouse Cloud

Updated: May 20, 2018

## ADWC Lab 8: Advanced Data Loading Techniques

You load data into Autonomous Data Warehouse Cloud using Oracle Database tools, and Oracle or other 3rd party data integration tools. In general you load data from files local to your client computer or from files stored in a cloud-based object store.

For the fastest data loading experience Oracle recommends uploading the source files to a cloud-based object store, such as Oracle Cloud Infrastructure (OCI) Object Storage, before loading the data into ADWC.

Oracle provides support for loading files that are located locally in your data center, but you should factor in the transmission speeds across the Internet which may be significantly slower than loading data directly from the OCI object storage.

This lab has the following parts:

- **Part 1:** Loading Data From Local Files Using SQL\*Loader
- **Part 2:** Importing Data Using Oracle Data Pump

### Part 1: Loading Data From Local Files Using Oracle SQL\*Loader

---

You can use **Oracle SQL\*Loader** to load data from local files in your client machine to ADWC.

**Note:** SQL\*Loader may be suitable for loading small amounts of data, as the load performance depends on the network bandwidth between your client and ADWC. For large amounts of data Oracle

recommends loading data from the Cloud Object Storage.

## Objectives

- Generate SQL\*Loader scripts using SQL Developer
- Load a Local CSV file to ADWC using SQL\*Loader scripts
- Validate the load using SQL Developer

## Required Artifacts

- Access to the Lab VM.
  - Locate the Lab VM's **IP Address**, **User Name** and **Password**
- The following artifacts of the Lab VM are utilized:
  - Installed software: **Oracle SQL Developer** and **Oracle Database Client** (includes **Oracle SQL\*Loader**)
  - Sample comma separated file (CSV)
  - Wallet files configured in Oracle Client's home directory

## Lab Steps

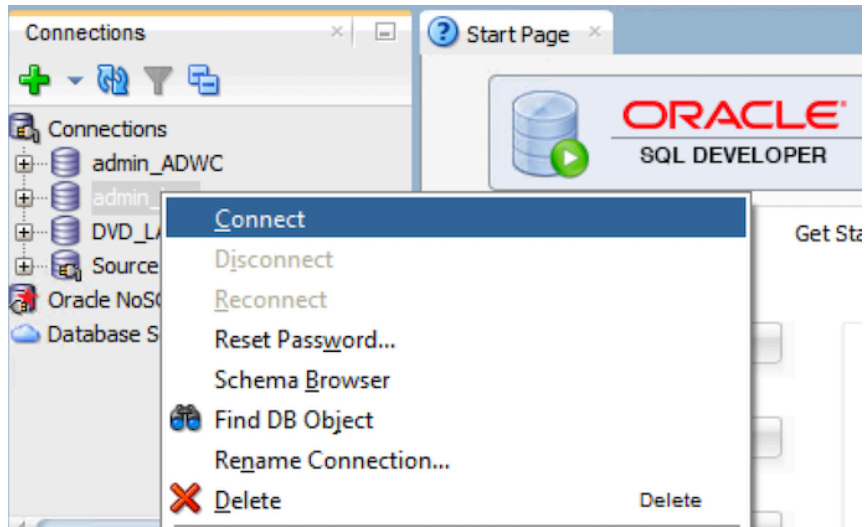
Loading data using SQL\*Loader can be accomplished in 2 steps.

- Generate SQL\*Loader scripts using SQL Developer
- Run the generated SQL\*Loader scripts

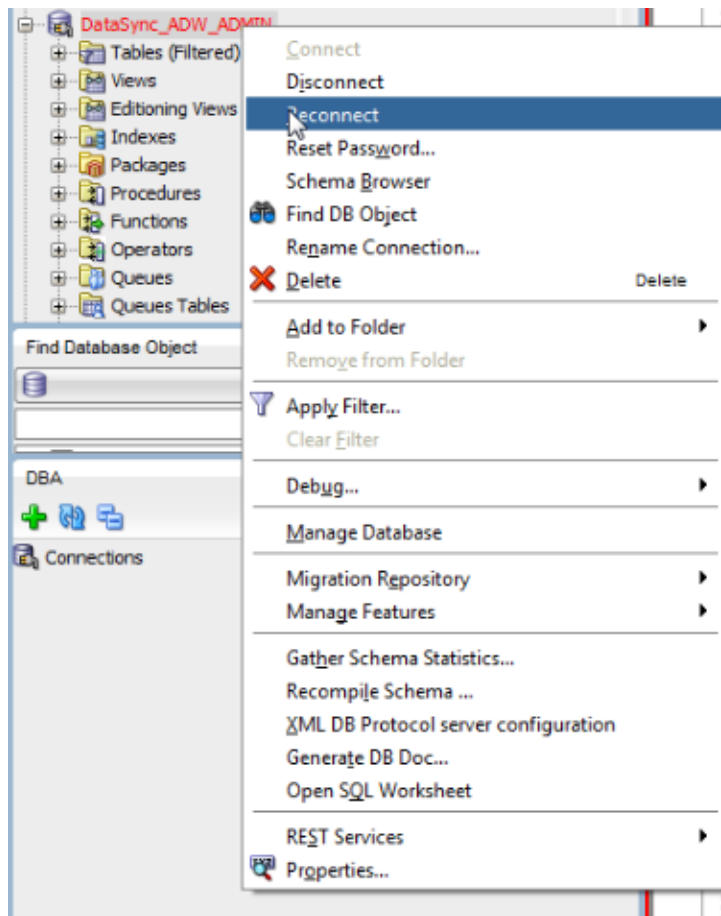
### Step 1: Generate SQL\*Loader Scripts Using SQL Developer

You may manually create SQL\*Loader control files or use SQL Developer to generate them for you. We will use the latter method for this lab.

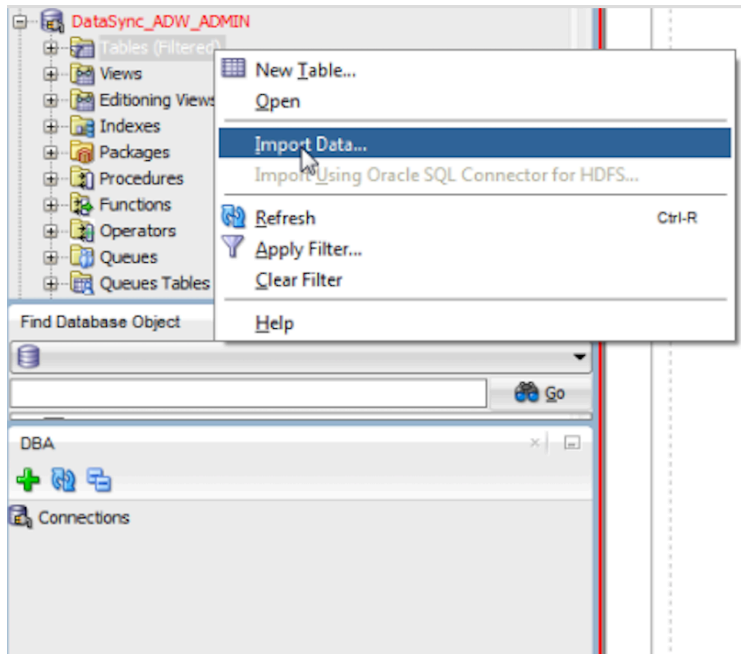
- Sign in to the Lab VM using the credentials provided to you by the instructor.
- Start **SQL Developer** and connect to the ADWC connection you have defined in the previous labs.



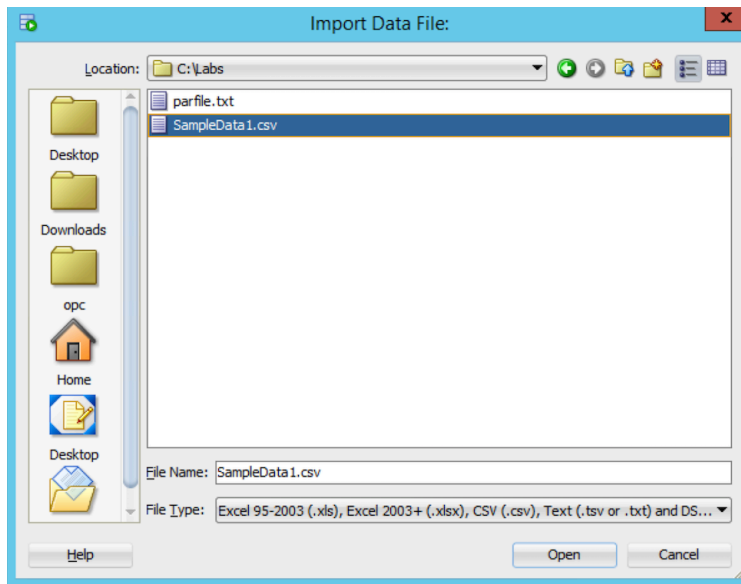
**Note:** If you had an existing connection open, you may need to reconnect as your session may have timed-out due to the database IDLE\_TIME resource limit that is in effect for the session.



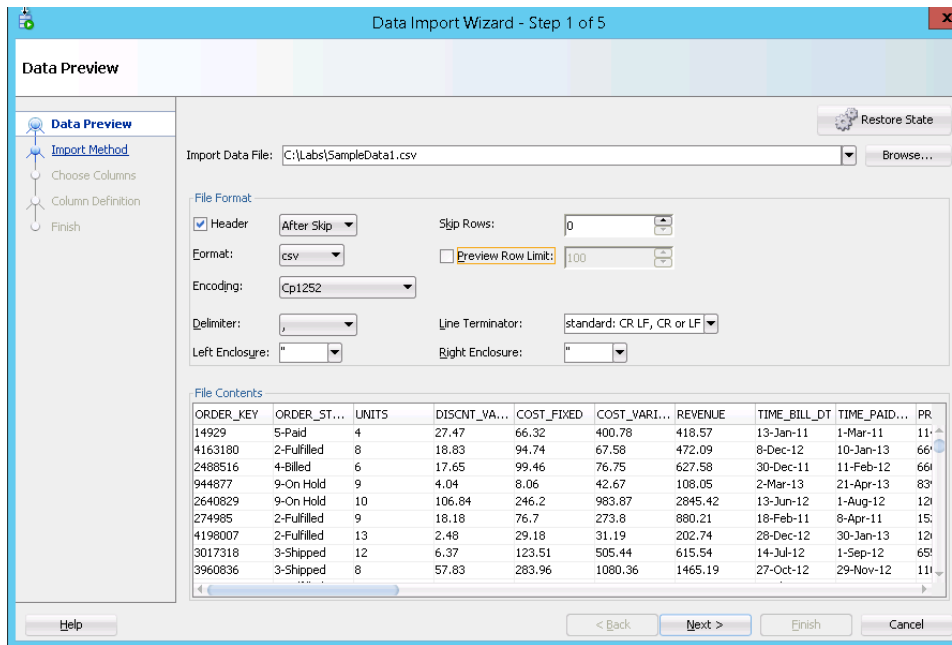
- In the left pane, click **Tables** and **Right-Click** and select **Import Data**.



- The **Data Import wizard** will pop-up. Click **Browse** and locate the CSV file which is saved as **C:\Labs\SampleData1.csv**. Click **Open**.



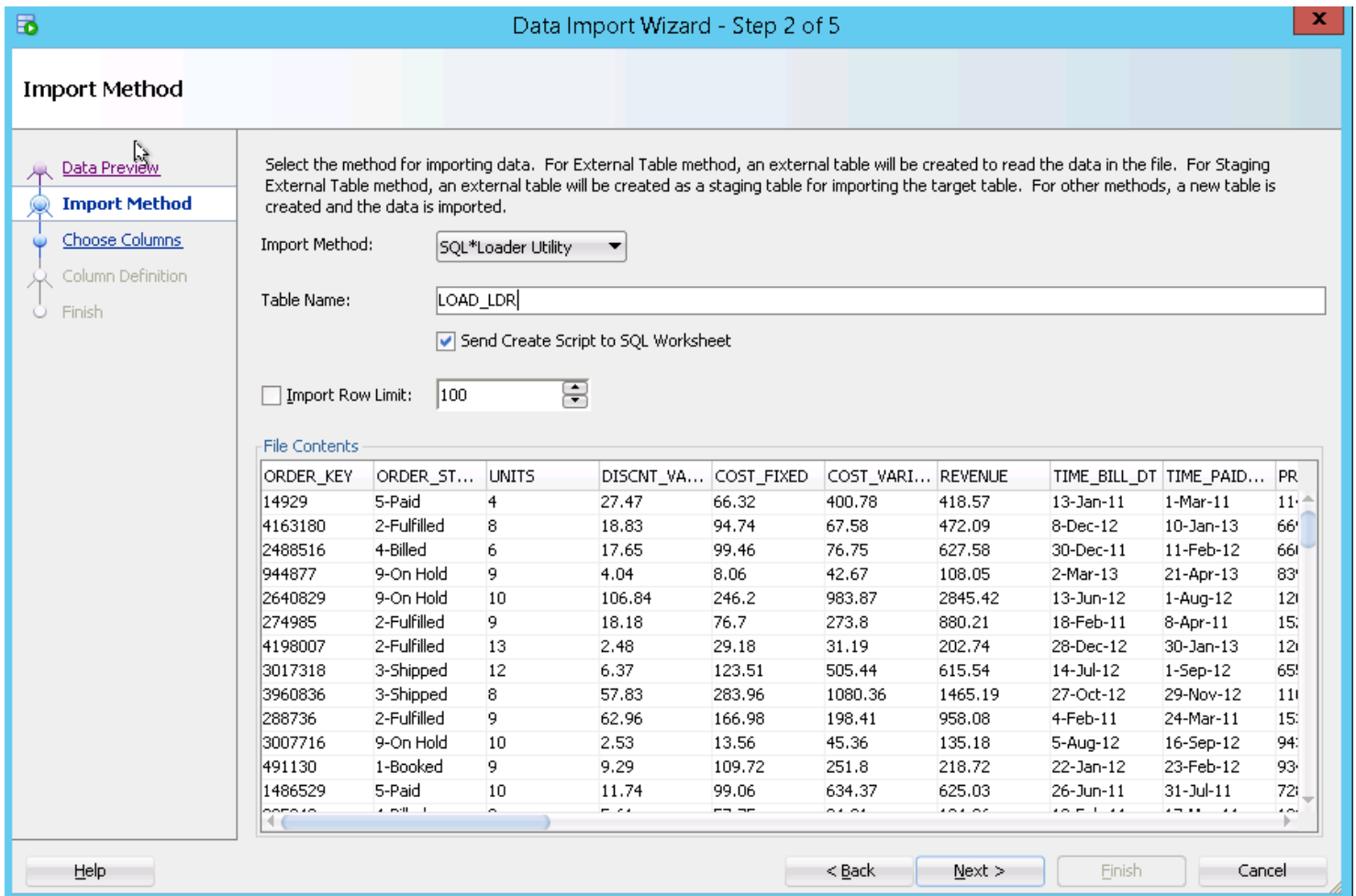
- Uncheck the **Preview Row Limit** and ensure the delimiter is **,**. Click **Next**.



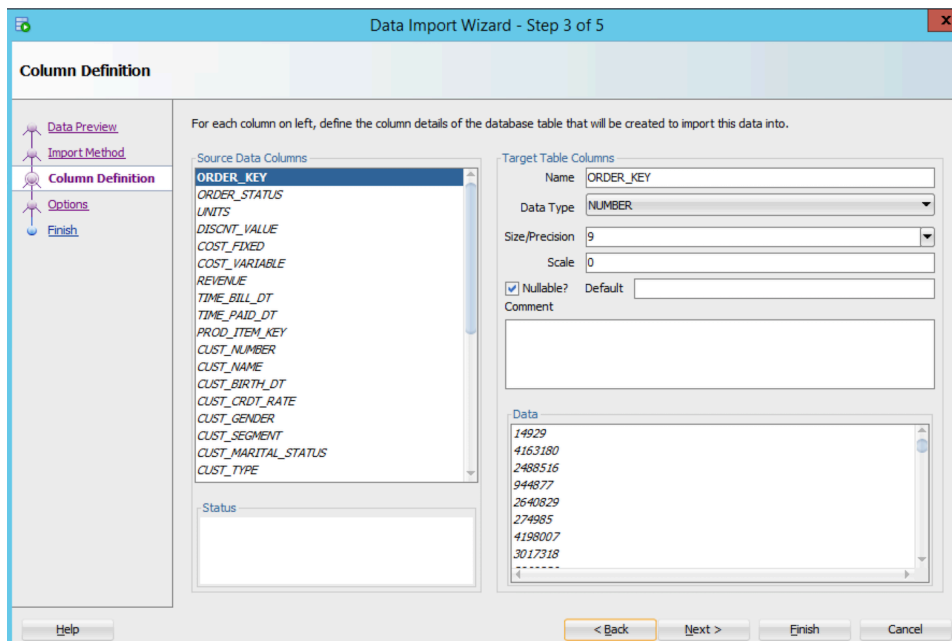
- In the **Import Method** choose **SQL\*Loder Utility**. Enter a table name where you like the data to be loaded, e.g. **LOAD\_LDR**.

**Note:** The Database Table need not be precreated in ADWC. The Create Table DDL will be generated as a part of this process and you can run it to create the table.

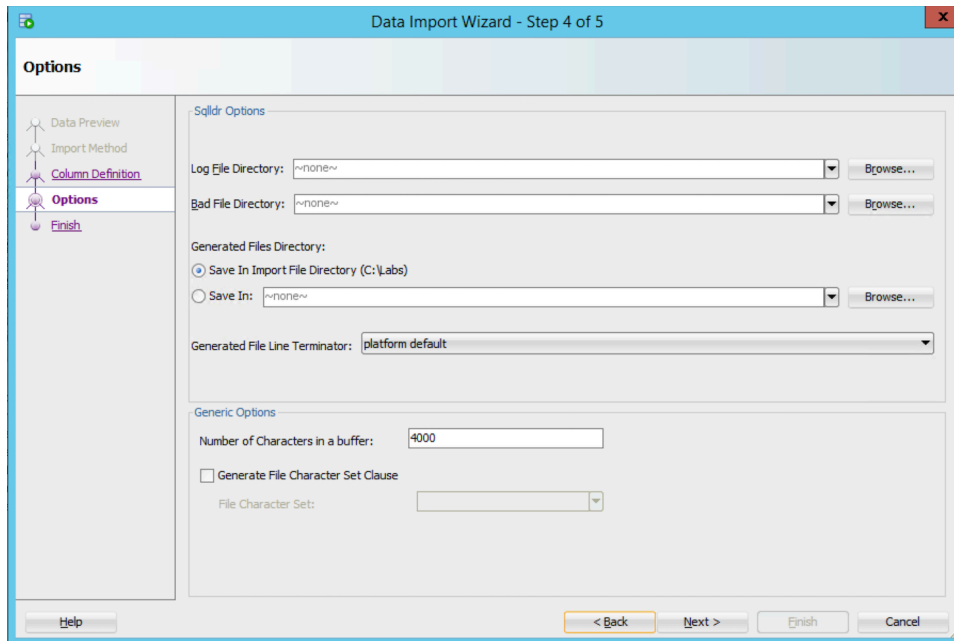
- Ensure that **Send Create Script to SQL Worksheet** is Checked. Click **Next**.



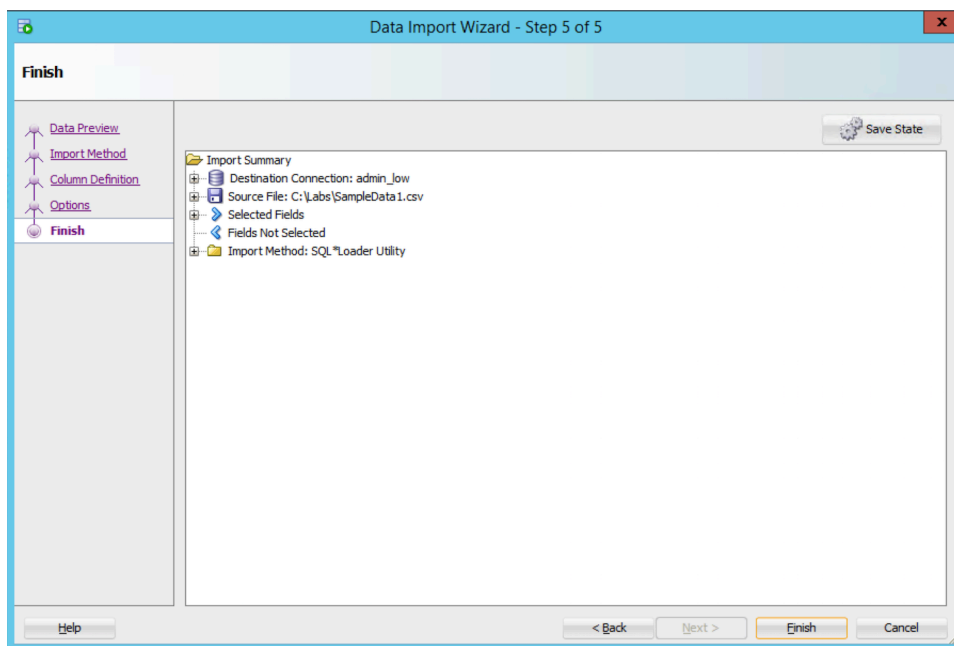
- The next screen allows you to select the columns you like to include as part of the load and also as part of the table column list for the DDL. For this exercise, keep all settings as default the effect of which is to include all columns. Click **Next**.



- This screen allows you to set SQL\*Loader options. For this exercise just note the location where the scripts will be generated. Leave the rest of the options as default. Click **Next**.



- Click **Finish**.



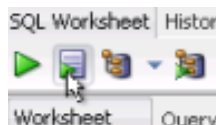
- A new SQL Worksheet is created with the create table DDL command.

```

SET DEFINE OFF
CREATE TABLE LOAD_LDR ( ORDER_KEY NUMBER(9),
ORDER_STATUS VARCHAR2(11),
UNITS NUMBER(2),
DISCNT_VALUE NUMBER(5, 2),
COST_FIXED NUMBER(7, 2),
COST_VARIABLE NUMBER(7, 2),
REVENUE NUMBER(6, 2),
TIME_BILL_DT DATE,
TIME_PAID_DT DATE,
PROD_ITEM_KEY NUMBER(4),
CUST_NUMBER NUMBER(4),
CUST_NAME VARCHAR2(20),
CUST_BIRTH_DT DATE,
CUST_CRDT_RATE NUMBER(3),
CUST_GENDER VARCHAR2(1),
CUST_SEGMENT VARCHAR2(14),
CUST_MARITAL_STATUS VARCHAR2(8),
CUST_TYPE VARCHAR2(6),
ADDR_KEY NUMBER(5),
ADDRESS1 NUMBER(5),
ADDRESS2 VARCHAR2(30),
POSTAL_CODE VARCHAR2(8),
CITY VARCHAR2(21),
STATE_PROV VARCHAR2(14),
REGION VARCHAR2(8),
COUNTRY_CODE VARCHAR2(3),
COUNTRY_NAME VARCHAR2(35),
AREA VARCHAR2(13),
CHANNEL_NAME VARCHAR2(8),
PROD_BRAND VARCHAR2(8),
PROD_LOB VARCHAR2(13),
PROD_TYPE VARCHAR2(12),
PRODUCT VARCHAR2(26),

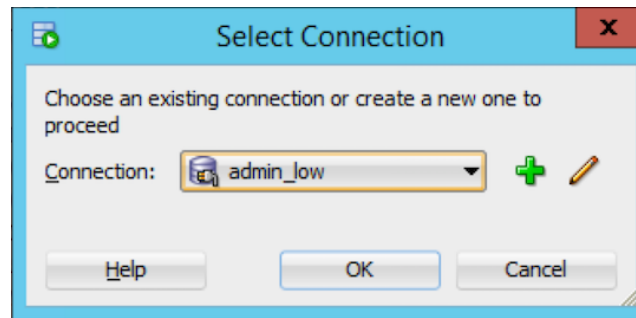
```

- Execute the Create Table script. Click **F5** or the **Run Script** button.

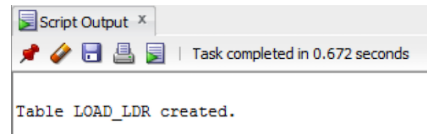


- You may be presented with **Select a Connection** dialog. Ensure that it points to the right ADWC connection and click **OK**.





- Your table should be created.



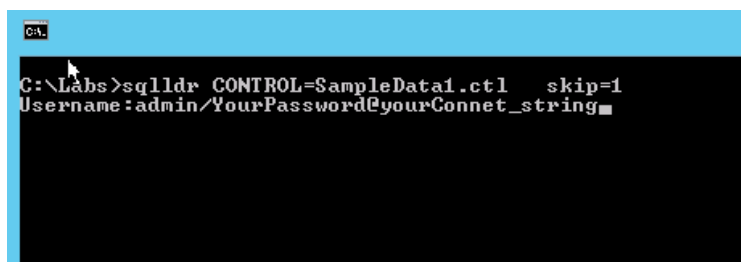
## Step 2: Run the Generated SQL\*Loader Scripts

Next, let's use the generated scripts from the previous step to load data using SQL\*Loader.

- Using **Windows Explorer**, browse to **C:\Labs** directory and click to run **SampleData1.bat**.
- A command screen will pop-up prompting for a **Username**. Enter the following string :

```
ADMIN/<YourPassword>@<YourConnect_String>
```

- Where:
  - **<YourPassword>** is the password for the ADMIN user (entered when creating the service).
  - **<YourConnect\_String>** is the TNS Alias to one of the ADWC Services (e.g. LOW, MEDIUM or HIGH).



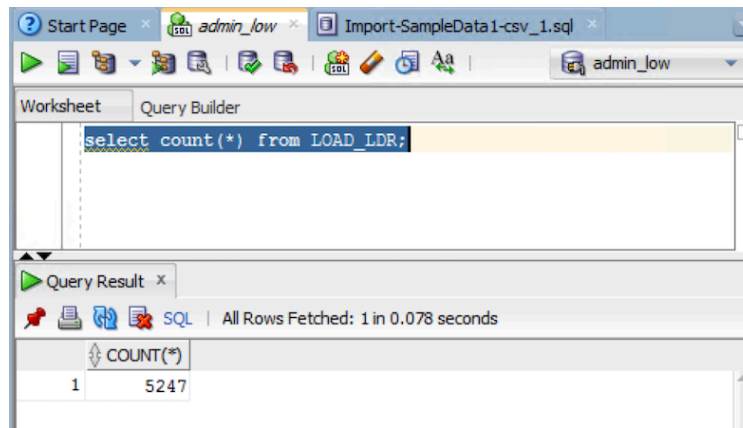
- Your source file is now being loaded into ADWC.

```
Commit point reached - logical record count 2490
Commit point reached - logical record count 2493
Commit point reached - logical record count 2496
Commit point reached - logical record count 2499
Commit point reached - logical record count 2502
Commit point reached - logical record count 2505
Commit point reached - logical record count 2508
Commit point reached - logical record count 2511
Commit point reached - logical record count 2514
Commit point reached - logical record count 2517
Commit point reached - logical record count 2520
Commit point reached - logical record count 2523
Commit point reached - logical record count 2526
Commit point reached - logical record count 2529
Commit point reached - logical record count 2532
Commit point reached - logical record count 2535
Commit point reached - logical record count 2538
Commit point reached - logical record count 2541
Commit point reached - logical record count 2544
Commit point reached - logical record count 2547
Commit point reached - logical record count 2550
Commit point reached - logical record count 2553
Commit point reached - logical record count 2556
Commit point reached - logical record count 2559
Commit point reached - logical record count 2562
Commit point reached - logical record count 2565
Commit point reached - logical record count 2568
Commit point reached - logical record count 2571
Commit point reached - logical record count 2574
Commit point reached - logical record count 2577
Commit point reached - logical record count 2580
Commit point reached - logical record count 2583
Commit point reached - logical record count 2586
Commit point reached - logical record count 2589
Commit point reached - logical record count 2592
Commit point reached - logical record count 2595
Commit point reached - logical record count 2598
Commit point reached - logical record count 2601
Commit point reached - logical record count 2604
Commit point reached - logical record count 2607
Commit point reached - logical record count 2610
Commit point reached - logical record count 2613
Commit point reached - logical record count 2616
Commit point reached - logical record count 2619
Commit point reached - logical record count 2622
Commit point reached - logical record count 2625
Commit point reached - logical record count 2628
Commit point reached - logical record count 2631
Commit point reached - logical record count 2634
Commit point reached - logical record count 2637
Commit point reached - logical record count 2640
Commit point reached - logical record count 2643
Commit point reached - logical record count 2646
```

- Once the load completes, run a COUNT(\*) query and check the rowcount of table data that just got loaded.
- Open a Worksheet in **SQL Developer** connected to the ADWC Service. Run the query below:

```
select count(*) from <TableName>;
```

- Where **<TableName>** is the name of the table you entered in the wizard while loading (e.g. LOAD\_LDR).



- This completes the SQL Loader lab.

## Part 2: Importing Data Using Oracle Data Pump

---

Oracle Data Pump offers very fast bulk data and metadata movement between Oracle databases and Autonomous Data Warehouse Cloud.

Data Pump Import lets you import data from Data Pump files residing on the Oracle Cloud Infrastructure Object Storage, Oracle Cloud Infrastructure Object Storage Classic, and AWS S3. You can save your data to your Cloud Object Store and use Oracle Data Pump to load data to Autonomous Data Warehouse Cloud.

In this lab you will import a dump file from the Oracle Object Storage into **ADWC** using Oracle Data Pump.

### Objectives

- Import a schema into ADWC from Oracle Object Storage
- Understand the role of Credentials

### Required Artifacts

- Access to the Lab VM.
  - Locate the Lab VM's **IP Address**, **User Name** and **Password**
- The following artifacts of the Lab VM are used:
  - Installed software: **Oracle SQL Developer** and **Oracle Database Client** (which includes **Oracle Data Pump**)
  - Wallet files configured in Oracle Client's home directory

- Data Pump Export file previously uploaded to Oracle Object Storage
  - Object Storage Swift credentials are provided in the lab

## Lab Steps

### Step 1: Creating Cloud Object Storage Credentials

- To load data from the Oracle Cloud Infrastructure Object Storage you will need a Cloud user with the appropriate privileges to read data (or upload) data from Object Store. The communication between the database and the object store relies on the Swift protocol and username/password authentication.

**Note:** For the purposes of this lab, we are providing the Swift username/password to connect to the Object Store instead of the credentials to the OCI Object Storage.

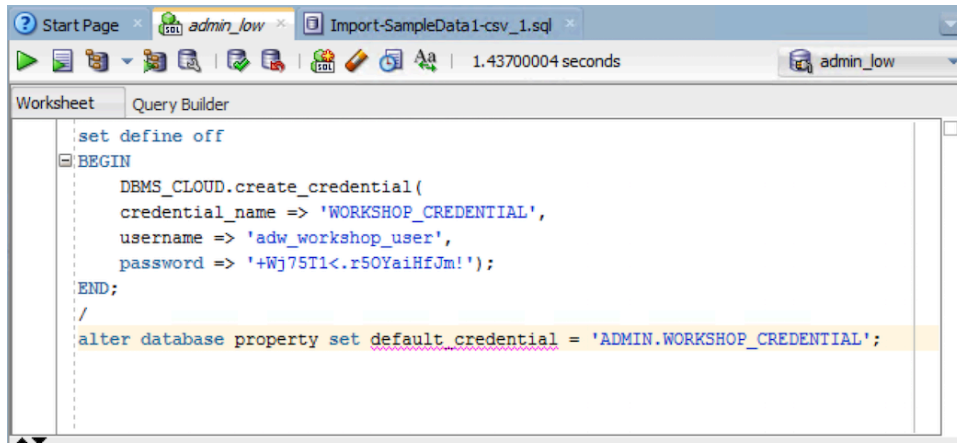
- In order to access data in the Object Store you need to enable the database user to authenticate with the Object Store using your object store account and Swift password.
- This is done by creating a private CREDENTIAL object that stores this information encrypted in ADWC.
- Using a SQL worksheet of SQL Developer and connected to admin\_low connection, execute the following code to create the object store credential.

```
set define off
BEGIN
  DBMS_CLOUD.create_credential(
    credential_name => 'WORKSHOP_CREDENTIAL',
    username => 'adw_workshop_user',
    password => '+Wj75T1<.r50YaiHfJm!');
END;
/
```

- Set the credential as the default credential for your ADWC, as the ADMIN user.

```
alter database property set default_credential = 'ADMIN.WORKSHOP_CREDENTIAL';
```

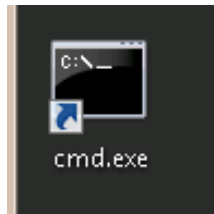
**Note:** The above step is required only when using Oracle Data Pump version 12.2.0.1. and earlier.



```
set define off
BEGIN
  DBMS_CLOUD.create_credential(
    credential_name => 'WORKSHOP_CREDENTIAL',
    username => 'adw_workshop_user',
    password => '+Wj75T1<.r50YaiHfJm!');
END;
/
alter database property set default_credential = 'ADMIN.WORKSHOP_CREDENTIAL';
```

## Step 2: Run Data Pump Import

- Click on the cmd.exe to start a command terminal.



- Copy the below impdp command and paste it in a text editor or SQL Developer Worksheet in the Lab VM.

```
impdp userid=admin/<YourPassword>@<YourConnect_string> ^
remap_schema=workshop_schema:admin ^
schemas=workshop_schema remap_tablespace=USERS:DATA ^
directory=data_pump_dir ^
dumpfile=default_credential:https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v
1/oraclepartnersas/workshop_object_storage_bucket/workshop_schema.dmp ^
logfile=data_pump_dir:workshop_schema.dmp.log ^
```

- Edit the pasted script and enter **<YourPassword>** and **<YourConnect\_string>** (keep the windows shell line continuation character **^ (caret)** at the end of each line intact when copying/pasting).
  - **<YourPassword>** is the password entered while creating the service
  - **<YourConnect\_string>** is the TNS Alias you have used earlier to connect to ADWC (preferably use the HIGH service for improved performance)
- Run the impdp command by copying and pasting the edited command that has your password and connect string for user **ADMIN**.

**Note:** The dumpfile has the syntax "default\_credential:https://swiftobjectstorage.

<CLOUD\_REGION>.oraclecloud.com/v1/<OCI\_TENANT\_NAME>/<BUCKET\_NAME>/<OBJECT\_NAME>"

```
C:\user\opc>impdp userid=admin/password@workshop_low
remap_schema=workshop_schema:admin schemas=workshop_schema remap_tablespace=USERS:DAT
A directory=data_pump_dir
dumpfile=default_credential:https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v
1/oraclepartnersas/workshop_object_storage_bucket/workshop_schema.dmp logfile=data_p
ump_dir:workshop_schema.dmp.log
```

```
Import: Release 12.1.0.2.0 - Production on Thu Apr 26 19:59:29 2018
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.
Connected to: Oracle Database 18c Enterprise Edition Release 12.2.0.1.0 - 64bit Produ
ction
Master table "ADMIN"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
```

```
Starting "ADMIN"."SYS_IMPORT_SCHEMA_01":  userid=admin/*****@workshop_low remap_sc
hema=workshop_schema:admin schemas=workshop_schema remap_tablespace=USERS:DATA direct
ory=data_pump_dir dumpfile=default_credential:https://swiftobjectstorage.us-phoenix-1
.oraclecloud.com/v1/oraclepartnersas/workshop_object_storage_bucket/workshop_schema.d
mp logfile=data_pump_dir:workshop_schema.dmp.log
```

```
Processing object type SCHEMA_EXPORT/USER
ORA-31684: Object type USER:"ADMIN" already exists
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
ORA-39151: Table "ADMIN"."COPY$3_LOG" exists. All dependent metadata and data will be
skipped due to table_exists_action of skip
```

```
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "ADMIN"."SALES"          29.63 MB   918843 rows
. . imported "ADMIN"."CUSTOMERS"     10.27 MB    55500 rows
. . imported "ADMIN"."COSTS"         2.420 MB    82112 rows
. . imported "ADMIN"."SUPPLEMENTARY_DEMOGRAPHICS" 697.6 KB    4500 rows
. . imported "ADMIN"."TIMES"         381.7 KB    1826 rows
. . imported "ADMIN"."PROMOTIONS"    59.18 KB     503 rows
. . imported "ADMIN"."PRODUCTS"     26.73 KB     72 rows
. . imported "ADMIN"."COUNTRIES"    10.47 KB     23 rows
. . imported "ADMIN"."CHANNELS"      7.562 KB     5 rows
. . imported "ADMIN"."ABC"           0 KB         0 rows
. . imported "ADMIN"."TEST_SQLLDR"  1.376 MB    5249 rows
. . imported "ADMIN"."TEST_SQLLDR1" 4.088 MB   15741 rows
```

```
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
```

```
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCOBJ
Job "ADMIN"."SYS_IMPORT_SCHEMA_01" completed with 2 error(s) at Fri Apr 27 01:03:56 2
018 elapsed 0 00:04:26
```

**Note:** The impdp logfile is created in the default DATA\_PUMP\_DIR directory on the ADWC Service instance. If you need to view the impdp log, you need to use DBMS\_CLOUD.PUT\_OBJECT procedure to copy the logfile into object storage and then download the logfile from object storage.

- This completes the Data Pump lab.