



## Autonomous Data Warehouse Cloud Service Workshop

April 2018



---

# TABLE OF CONTENTS

<b>Introduction .....</b>	<b>4</b>
<b>Lab prerequisites.....</b>	<b>5</b>
<b>Section 1. Provisioning an ADWC Service .....</b>	<b>6</b>
<b>Section 2. Connecting to ADWC.....</b>	<b>11</b>
<b>Section 3. Database Services and Sample Data Sets .....</b>	<b>15</b>
<b>Section 4. Loading a Local File into ADW .....</b>	<b>18</b>
<b>Section 5. Using Data Visualization Desktop on ADW .....</b>	<b>23</b>
<b>Section 6. Generating the Swift Password for Object Storage .....</b>	<b>38</b>
<b>Section 7. Data Loading into Cloud Object Storage.....</b>	<b>41</b>
<b>Section 8. Data Loading from Cloud Object Storage using SQL Developer Import Wizard.....</b>	<b>46</b>
<b>Section 9. Data Loading from Cloud Object Storage using PL/SQL .....</b>	<b>53</b>
<b>Section 10. Querying External Data .....</b>	<b>57</b>
<b>Section 11. Scaling an ADWC Service .....</b>	<b>59</b>
<b>Section 12. Monitoring ADWC.....</b>	<b>60</b>
<b>Apendix – SQL Scripts .....</b>	<b>62</b>

---

# INTRODUCTION

Oracle Autonomous Data Warehouse Cloud provides an easy-to-use, fully autonomous database that scales elastically, delivers fast query performance and requires no database administration.

It is a fully-managed cloud service that makes it very simple to provision a data warehouse, quickly and easily load data and query that data using built-in web-based tools such as notebooks.

Oracle's unique autonomous database framework ensures high availability and automatic security—without requiring any additional tasks.

Delivers high performance data warehousing straight out-of-the-box with unparalleled scalability and reliability. Built on key Oracle Database capabilities: parallelism, columnar processing and compression. All aspects of performance tuning are automatically managed so the service requires no database tuning.

Scale as needed—create and expand your data warehouse's compute and storage capacity on demand and independently of each other with no downtime. Pay only for the resources you consume.

Integrates directly with the full spectrum of business analytics, data integration and IoT services within Oracle's comprehensive range of integrated cloud solutions.

This lab walks you through all the steps to get started using the Oracle Autonomous Data Warehouse Cloud. You will provision a new ADWC database, create DW users, load data from the object store and troubleshoot data loads, query external data residing on the object store, and scale up your ADWC database.

This is an instructor led lab, please follow the instructor's guidance before doing the exercises. Stop and wait for indications before moving on to the next section.

---

# LAB PREREQUISITES

You will need SQL Developer installed on your computer to do the exercises in this lab guide. The minimum SQL Developer version that is required to connect to an Oracle Autonomous Data Warehouse Cloud instance is SQL Developer 17.4.

To exercise all labs, you will need the latest SQL Developer version 18.1.

Follow these steps to install SQL Developer depending on your platform.

## **Windows 64-bit**

Install SQL Developer 18.1 using “Windows 64-bit with JDK 8 included” from:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

## **Other platforms**

1. Install JDK 8u161 from:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

2. Install SQL Developer 18.1 for your platform:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

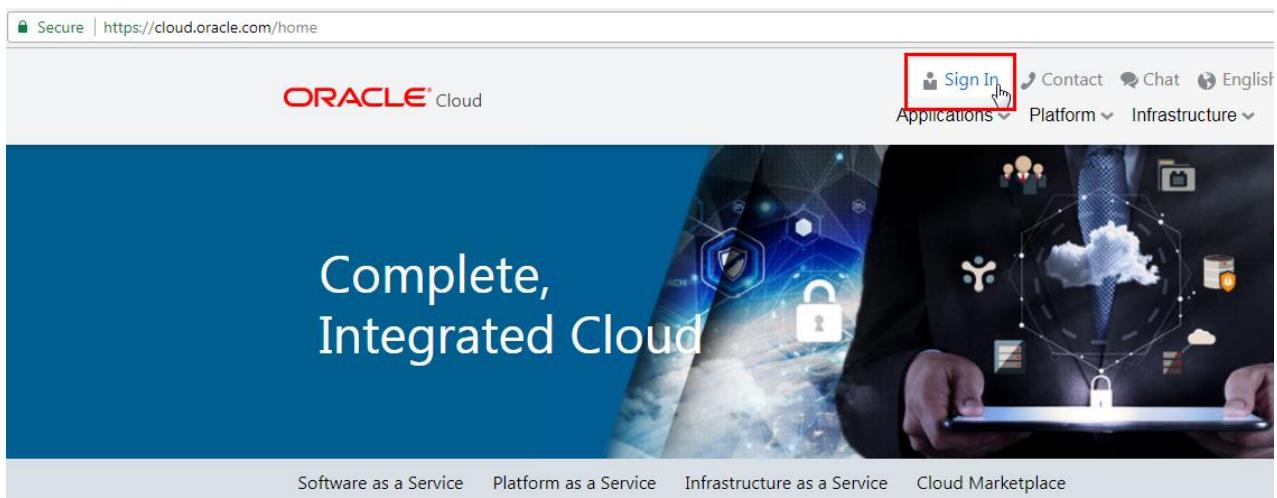
## Section 1. Provisioning an ADWC Service

In this section you will be provisioning an ADWC instance using the UI capabilities of the service.

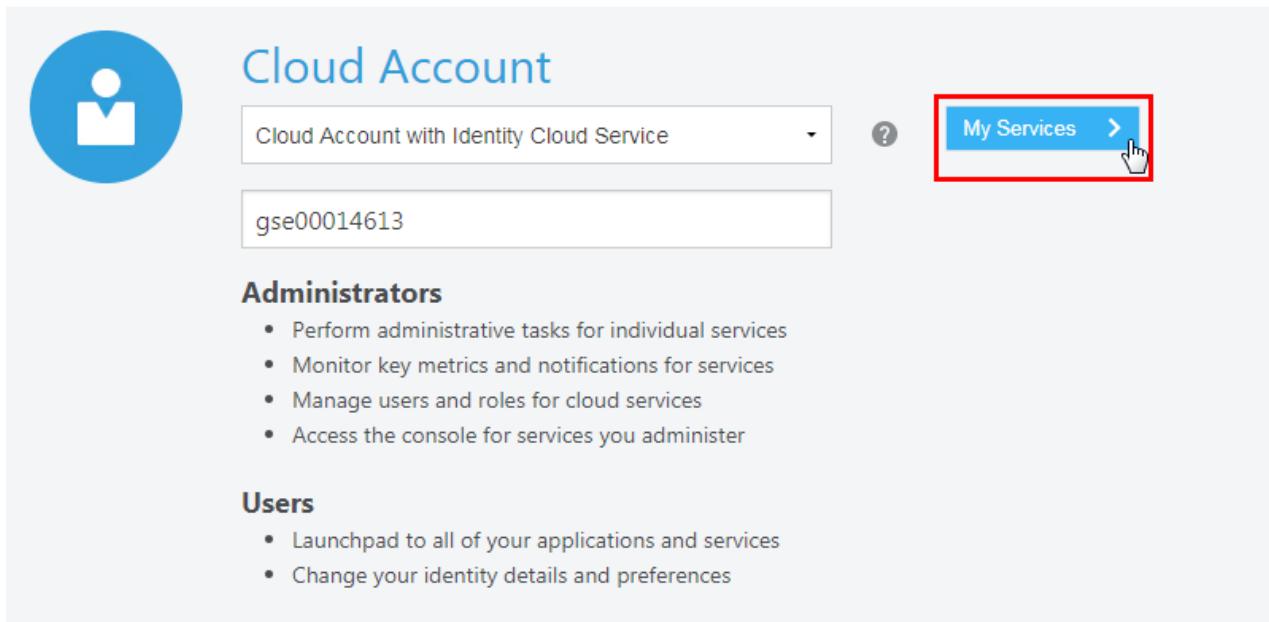
**Note:** The cloud environment will be provided separately to this lab guide, by the host. It will include:

- The Cloud Account Name
- The Username
- The Password

1. Go to [cloud.oracle.com](https://cloud.oracle.com) and click **Sign In** to login to your Oracle Cloud account.



2. Enter the **Cloud Account Name** provided to you by the host. In this example, the Cloud Account Name is `gse00014613`. Click **My Services**.



The screenshot shows the Oracle Cloud Account interface. At the top left is a blue circular icon with a white person icon. To its right is the title "Cloud Account". Below the title is a dropdown menu set to "Cloud Account with Identity Cloud Service". To the right of the dropdown is a question mark icon. Further right is a button labeled "My Services" with a right-pointing arrow, which is highlighted with a red box and a cursor icon pointing at it. Below the dropdown is a text input field containing "gse00014613". Underneath this is a section titled "Administrators" with a bulleted list: "Perform administrative tasks for individual services", "Monitor key metrics and notifications for services", "Manage users and roles for cloud services", and "Access the console for services you administer". Below that is a section titled "Users" with a bulleted list: "Launchpad to all of your applications and services" and "Change your identity details and preferences".

3. On the login page, fill in the Username and Password for your cloud account. Click **Sign In**.



cloud.admin

.....

**Sign In**

Can't sign in?

4. On the Home page, browse through the list of services and click the **Autonomous Data Warehouse Cloud** service.



5. On the service home page, click **Create Instance** to create a new database instance.

The screenshot shows the Oracle Autonomous Data Warehouse Cloud Instances page. At the top right, it says "Welcome!" and has a menu icon. Below that, it shows the date and time: "As of Apr 20, 2018 7:38:52 AM UTC". The main area is titled "Instances" and shows a message: "You don't have any instances. After meeting the [prerequisites](#), use this button to create an instance." Below this, there's a section for help: "Need help creating the instance? - Watch a video - Step through a tutorial". A red box highlights the "Create Instance" button, and a large gray arrow points to it from the bottom right.

6. Enter the following information in the **Create New Instance** screen:

**Database Name:** ADW<no.> – Choose a unique number if multiple attendees are allocated to the same cloud account.

**Note:** Do not use any special characters as the provisioning Wizard will not allow you to continue.

**CPU Count:** 2

**Storage Capacity (TB):** 1

**Administrator Password:** Welcome12345

Click **Next** to go to the confirmation screen.

The screenshot shows the "Create New Instance" wizard. At the top, it says "Create New Instance". Below that, it shows the progress: "Instance" (blue dot) — "Confirm" (gray dot). The "Instance" step is active. It asks for basic service instance information. The "Details" section contains fields for Database Name (ADW1), Description, Notification Email (<enter your email address>), and Tags. The "Configuration" section contains fields for CPU Count (2), Storage Capacity [TB] (1), Administrator Password, and Confirm Administrator Password. A checkbox for "Bring Your Own License" is also present. A red box highlights the "Next >" button at the top right of the step.

7. Review your information and click **Create** to provision the database. This will take you to the *Instances* screen.

## Create New Instance

[Previous](#) [Cancel](#)

Instance  Confirm

**Create ➔**

**Confirmation**  
Confirm your responses and create this Autonomous Data Warehouse.

Service	Configuration
Database Name: ADW1	CPU Count: 2
Bring Your Own License: false	Storage Capacity [TB]: 1
Software Release: 18.1.4	
Metering Frequency: HOURLY	

8. Click the **refresh** arrow button to see the status of your database. Once the service is provisioned click on its name to open the *Service Overview* page.

## Autonomous Data Warehouse Cloud

Welcome! 

[Instances](#) [Activity](#)

As of Apr 20, 2018 7:43:40 AM UTC 

**Summary** 1 Databases

**Instances**

Search by instance name or tags  

**Create Instance**

 Oracle Cloud Service instance creation request was accepted

 <b>ADW1</b>	Status: Creating service ...	Submitted On: Apr 20, 2018 7:43:39 AM UTC
---	------------------------------	---

  **Instance Create and Delete History**

9. Check the **Service Overview** page for more details on your ADW instance.

The screenshot shows the Oracle Database Service Overview page for a service named 'ADW1'. The top navigation bar includes a back arrow, a cloud icon, the service name 'ADW1', and a menu with three dots. Below the navigation is a toolbar with icons for play, pause, stop, refresh, and health.

The main content area has tabs for 'Service Overview' (which is selected) and 'Backup'. It displays the following instance overview information:

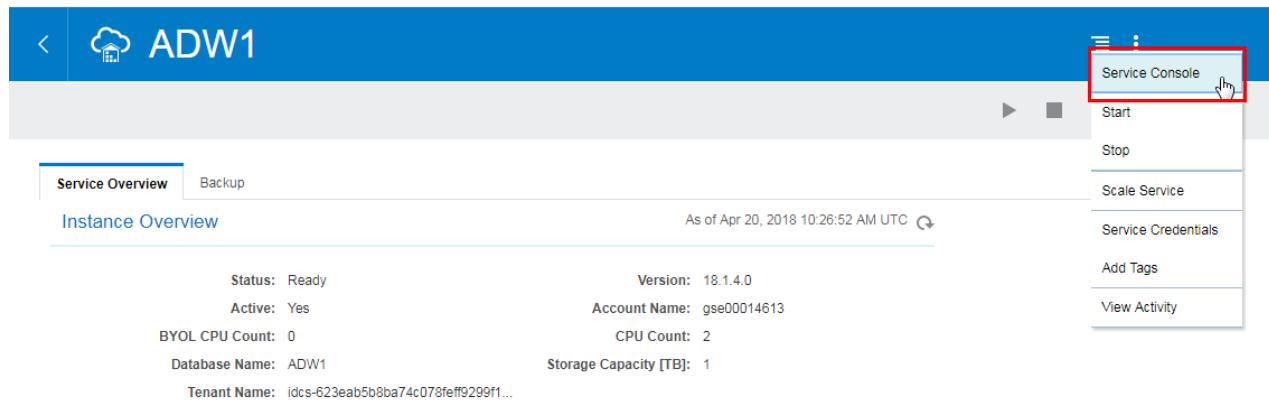
Status:	Ready	Version:	18.1.4.0
Active:	Yes	Account Name:	gse00014613
BYOL CPU Count:	0	CPU Count:	2
Database Name:	ADW1	Storage Capacity [TB]:	1
Tenant Name:	idcs-623eab5b8ba74c078feff9299f1...		

At the bottom right, it says 'As of Apr 20, 2018 7:45:30 AM UTC' with a refresh icon.

## Section 2. Connecting to ADWC

In this section you will go through the steps of accessing the credentials for connecting to your ADW instance with external clients, like SQL Developer.

1. The first step in accessing the credentials is to log in to the *Service Console* associated to your ADW instance. To do that, click on the menu in the upper right corner and click on **Service Console**.



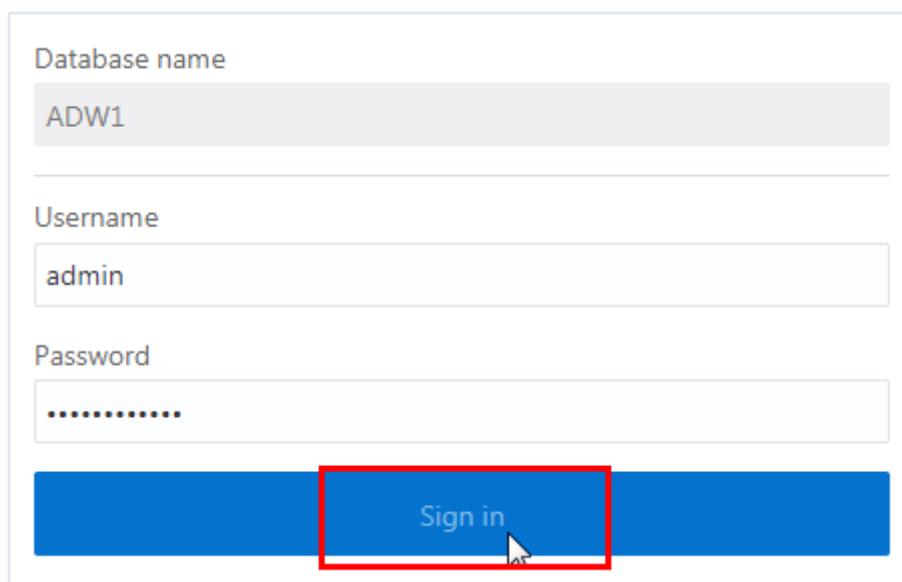
The screenshot shows the Oracle Service Console for an instance named ADW1. The top navigation bar has a blue header with the instance name 'ADW1'. On the right side of the header, there is a vertical menu with several options: 'Service Console' (which is highlighted with a red box), 'Start', 'Stop', 'Scale Service', 'Service Credentials', 'Add Tags', and 'View Activity'. Below the header, there are two tabs: 'Service Overview' (which is selected) and 'Backup'. Under the 'Service Overview' tab, there is a section titled 'Instance Overview' with various status metrics. At the bottom of the page, it says 'As of Apr 20, 2018 10:26:52 AM UTC'.

2. Login to the *Service Console* with the following credentials:

**Username:** admin

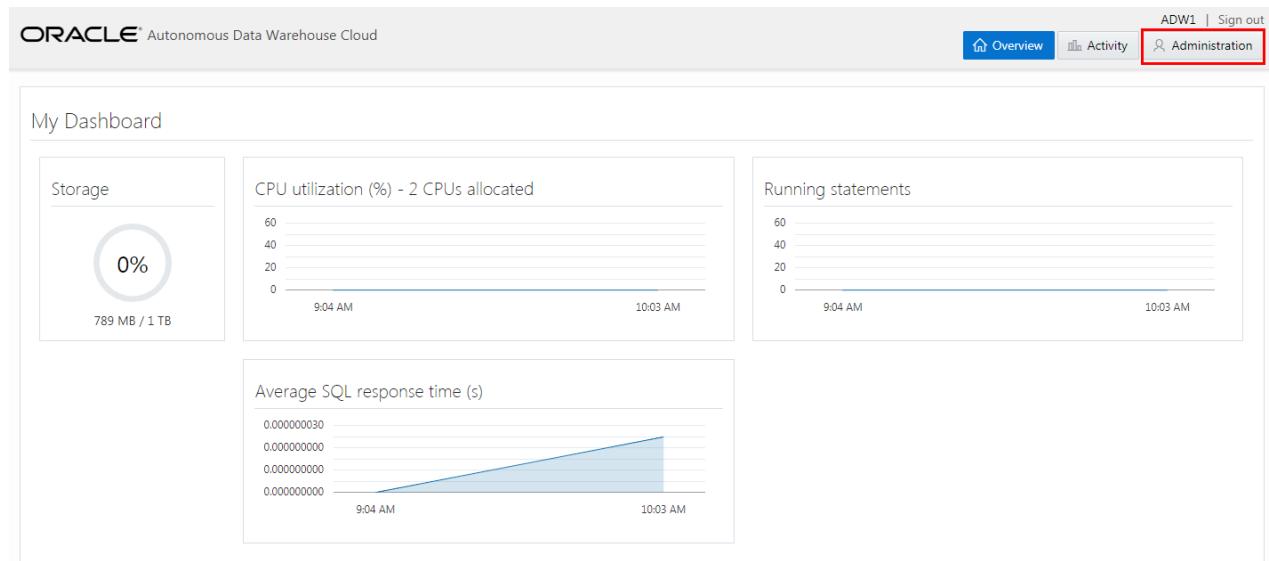
**Password:** Welcome12345

**Note:** this is the password you specified during the provisioning of the service.



The screenshot shows the login page of the Oracle Service Console. It has three input fields: 'Database name' (containing 'ADW1'), 'Username' (containing 'admin'), and 'Password' (containing a masked password). Below these fields is a large blue 'Sign in' button, which is also highlighted with a red box.

3. On the home page, click the **Administration** tab.



#### 4. Click Download Client Credentials.

<a href="#">Download Client Credentials</a>	Set Resource Management Rules Set rules for long running statements.
<a href="#">Set Administrator Password</a> Set or reset your database's privileged user (ADMIN) account password.	<a href="#">Manage Oracle ML Users</a> Create and manage Oracle ML users.
<a href="#">Download Oracle Instant Client</a> Download the latest version of Oracle Instant Client for Linux x86-64.	<a href="#">Send Feedback to Oracle</a> Provide feedback about the service to Oracle.

5. Enter a password before downloading the wallet zip file containing the credentials. This password will protect the sensitive data residing in the file. You can use **Welcome12345** as the password and then re-type it for confirmation.

Click **Download** and save the file on your local computer.

Client Credentials

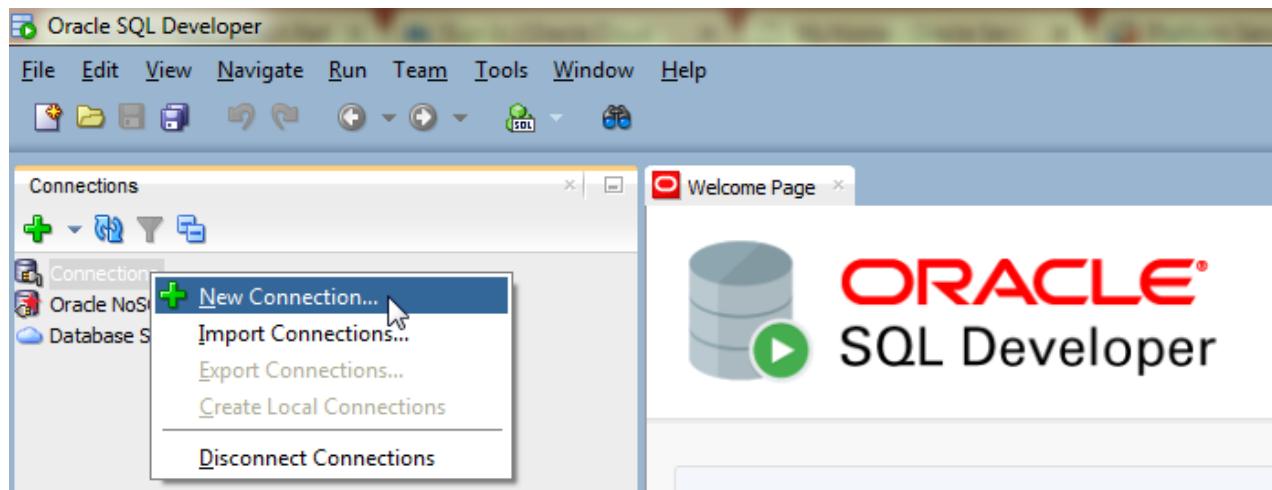
Download a zip file containing client security credentials and network configuration settings required to access your cloud database. You must protect this file to prevent unauthorized database access.

Password:  .....

Confirm password:  .....

**Download** **Cancel**

6. Open **SQL Developer** and click to create a new connection.



7. Fill in the details to connect to the database as follows:

**Connection Name:** admin\_low

**Username:** admin

**Password:** Welcome12345

**Connection Type:** Cloud PDB

**Role:** default

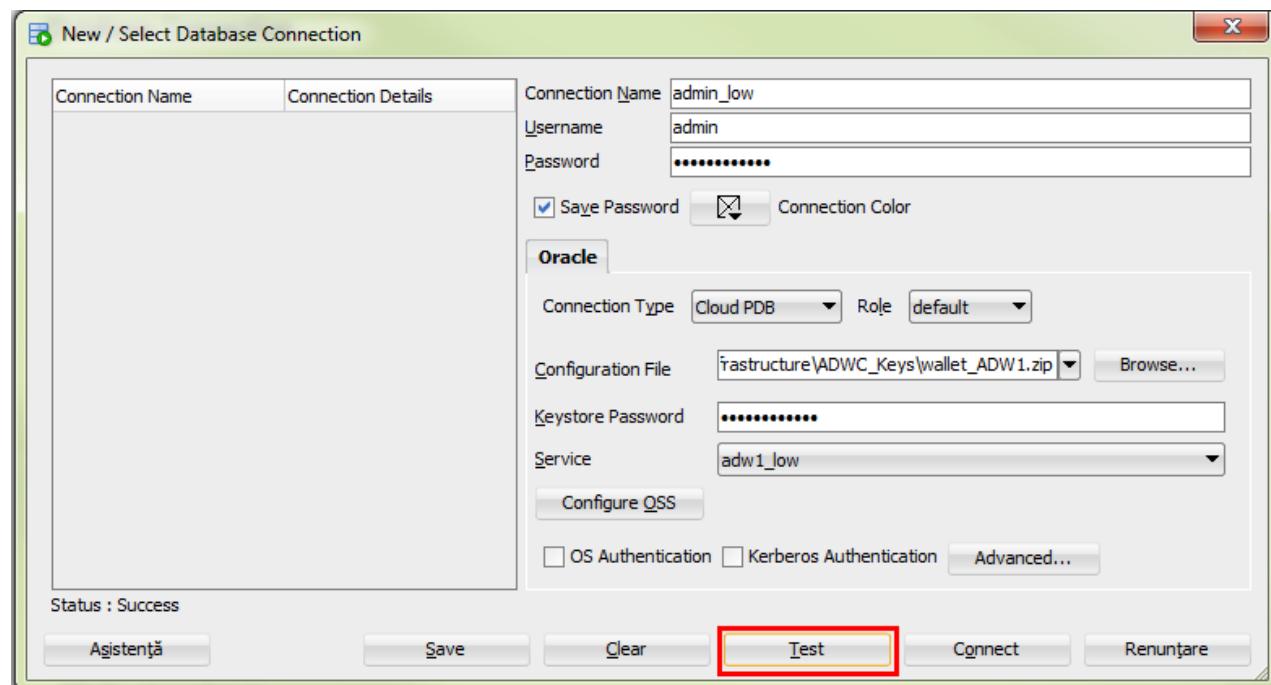
**Configuration File:** Browse to the location of the zipped wallet and select it

**Keystore Password:** Welcome12345

**Service:** adw1\_low

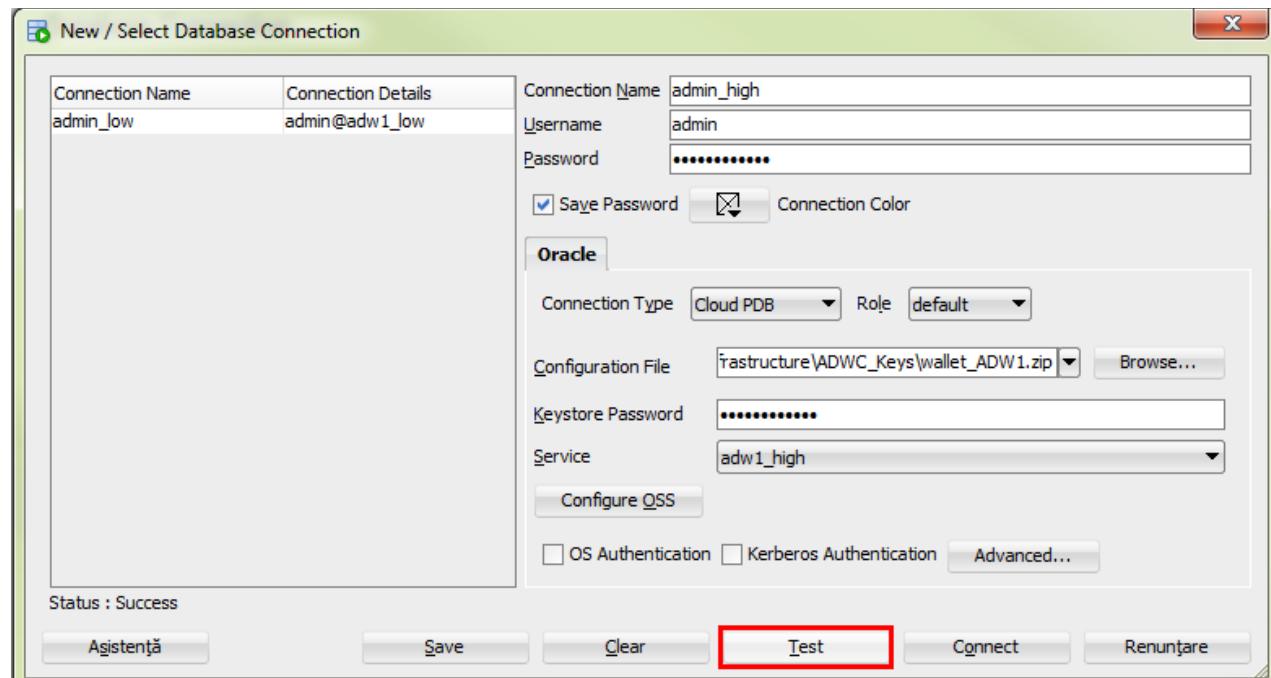
**Note:** The Keystore Password is the one you selected for your wallet before downloading it. The Service you need to choose is the name you gave to your service when provisioning it.

Click **Test** to check the connection.



8. If the test is successful **Save** the connection and click **Connect** to access the database.

9. Create a similar connection *admin\_high*.



---

## Section 3. Database Services and Sample Data Sets

**Autonomous Data Warehouse Cloud** provides three database services that you can choose when connecting to the database. These are named as HIGH, MEDIUM, and LOW services and provide different levels of performance and concurrency.

The **HIGH** database service provides the maximum amount of resources for a query, this also means the number of concurrent queries you can run with this service will not be as much as with the other services.

The **MEDIUM** database service provides multiple compute and IO resources for a query. This service also provides more concurrency compared to the HIGH database service.

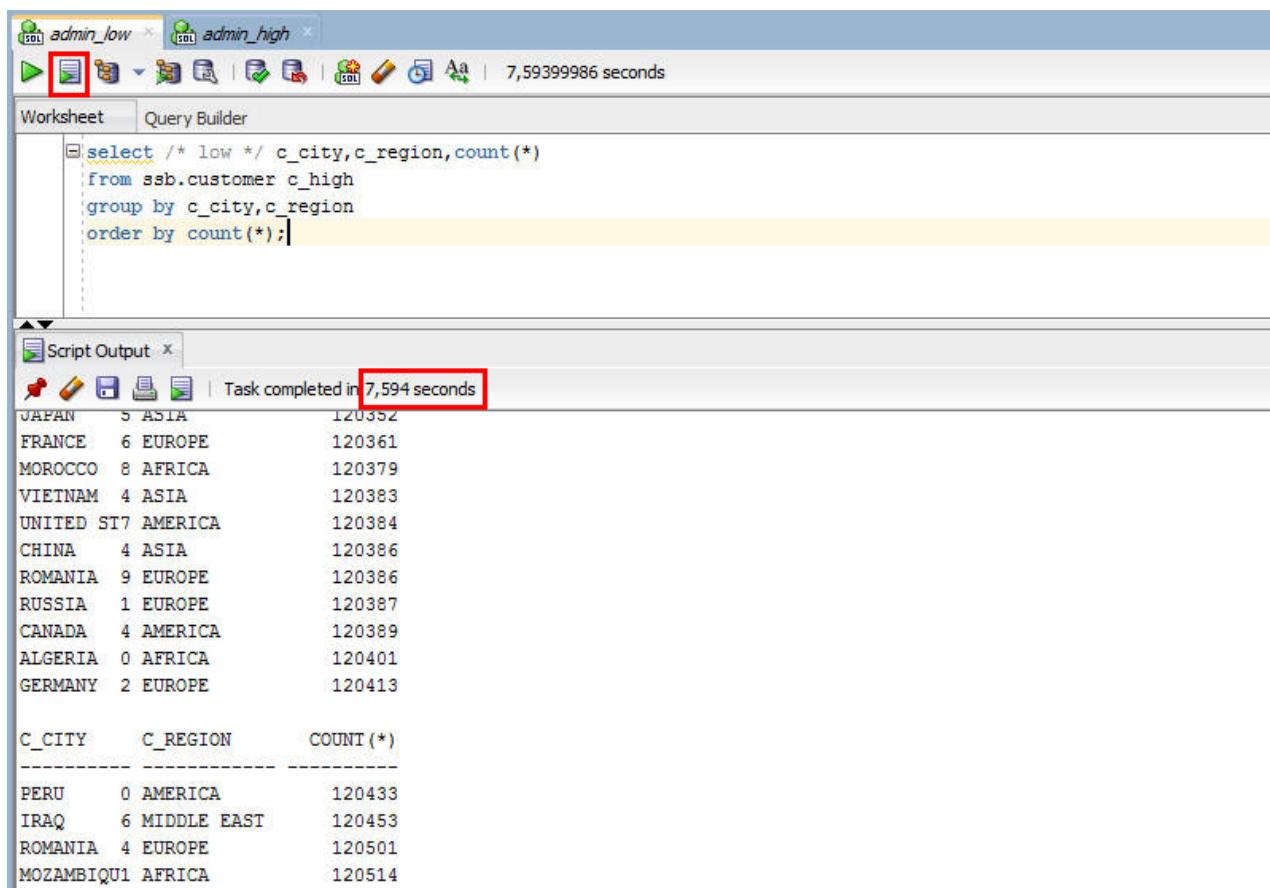
The **LOW** database service provides the least amount of resources for a query, this also means the number of concurrent queries you can run in this service will be higher than the other services.

In this section you will use the LOW and HIGH database services to understand the performance differences between them. For this exercise you will run queries on sample data sets provided out of the box with ADWC. ADWC provides the Oracle Sales History sample schema (<https://github.com/oracle/db-sample-schemas>) and the Star Schema Benchmark (SSB) data set, these data sets are in the SH and SSB schemas respectively.

You will run queries on the SSB data set which is a 1TB data set with one facttable with around 6 billion rows, and several dimension tables.

1. In SQL Developer, open the *admin\_low* connection and run the query below. Make sure you click the *Run Script* button, so that all rows are displayed on the screen.

```
select /* low */ c_city,c_region,count(*)  
from ssb.customer c_low  
group by c_city,c_region  
order by count(*);
```

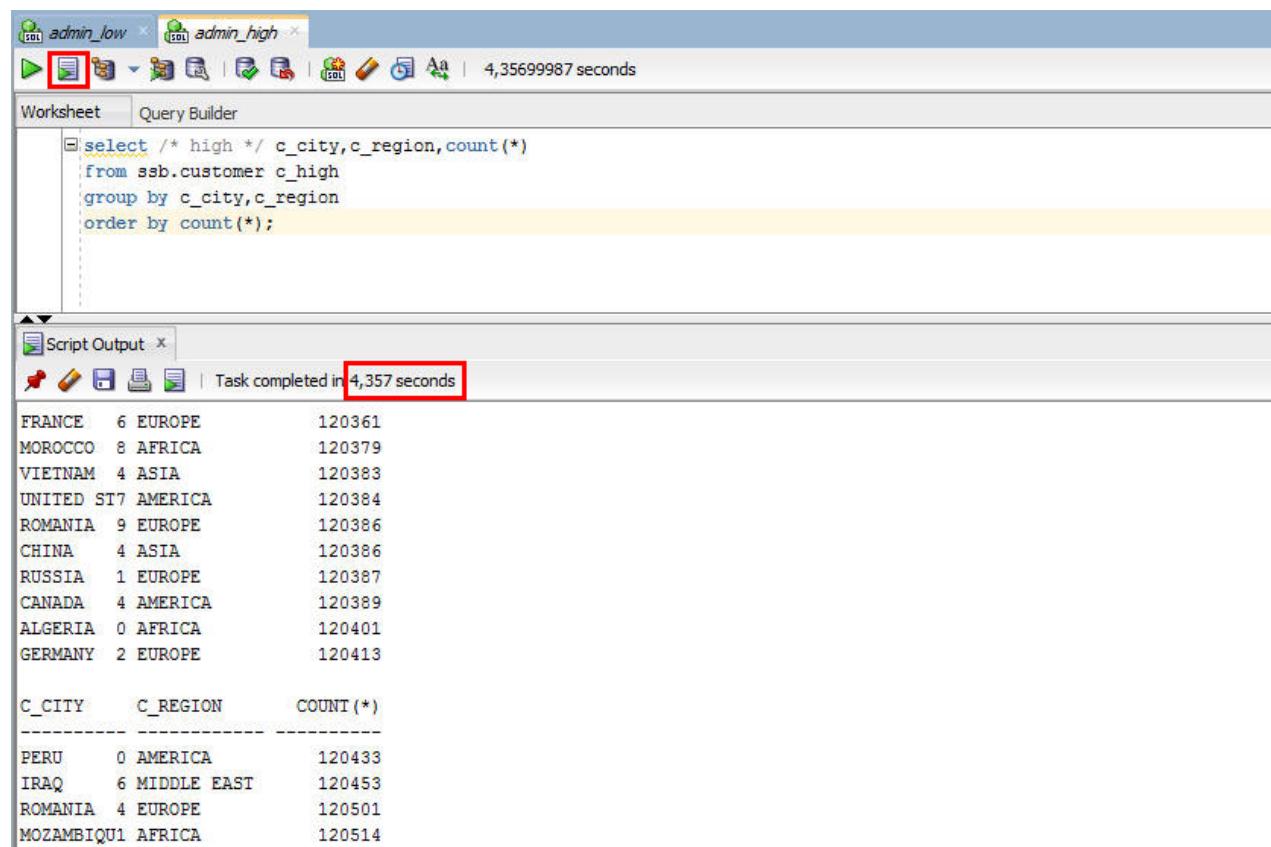


```
select /* low */ c_city,c_region,count(*)  
from ssb.customer c_high  
group by c_city,c_region  
order by count(*);
```

CITY	REGION	COUNT(*)
JAPAN	ASIA	120352
FRANCE	EUROPE	120361
MOROCCO	AFRICA	120379
VIETNAM	ASIA	120383
UNITED STATES	AMERICA	120384
CHINA	ASIA	120386
ROMANIA	EUROPE	120386
RUSSIA	EUROPE	120387
CANADA	AMERICA	120389
ALGERIA	AFRICA	120401
GERMANY	EUROPE	120413
PERU	AMERICA	120433
IRAQ	MIDDLE EAST	120453
ROMANIA	EUROPE	120501
MOZAMBIQUE	AFRICA	120514

2. Similarly, open the *admin\_high* connection and run the query below:

```
select /* high */ c_city,c_region,count(*)  
from ssb.customer c_high  
group by c_city,c_region  
order by count(*);
```



The screenshot shows the Oracle SQL Developer interface. At the top, there are two tabs: 'admin\_low' and 'admin\_high'. The 'admin\_high' tab is active, indicated by a red box around its icon in the toolbar. The status bar at the bottom shows '4,35699987 seconds'. The main area has two tabs: 'Worksheet' and 'Query Builder', with 'Worksheet' selected. A SQL query is written in the worksheet:select /\* high \*/ c\_city,c\_region,count(\*) from ssb.customer c\_high group by c\_city,c\_region order by count(\*);

```
Below the query, the 'Script Output' tab is selected, showing the results of the query. The results are displayed in two parts: a header row and a data table. The header row contains columns: C_CITY, C_REGION, and COUNT(*). The data table follows, showing the count of cities per region.
```

C_CITY	C_REGION	COUNT(*)
PERU	0 AMERICA	120433
IRAQ	6 MIDDLE EAST	120453
ROMANIA	4 EUROPE	120501
MOZAMBIQU1	AFRICA	120514

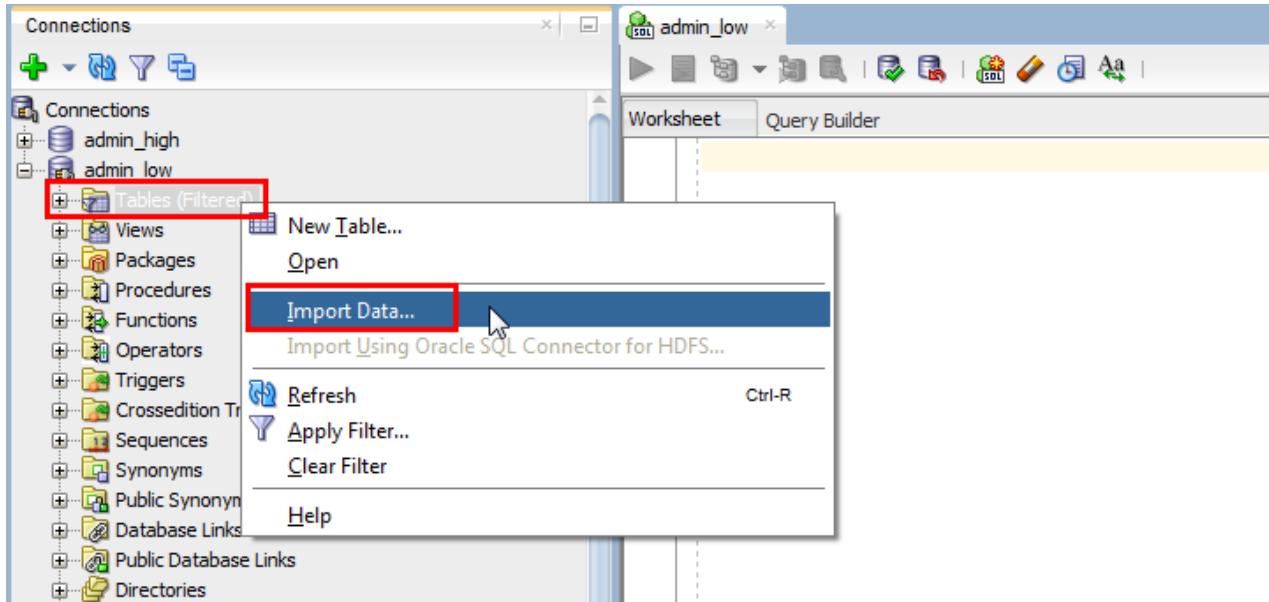
**Note:** As you can see from the example above, a query running in the HIGH database service can use more resources and run faster compared to a query running in the LOW database service.

As you scale up the compute capacity of your ADW service you will realize that the queries will get faster with the HIGH database service.

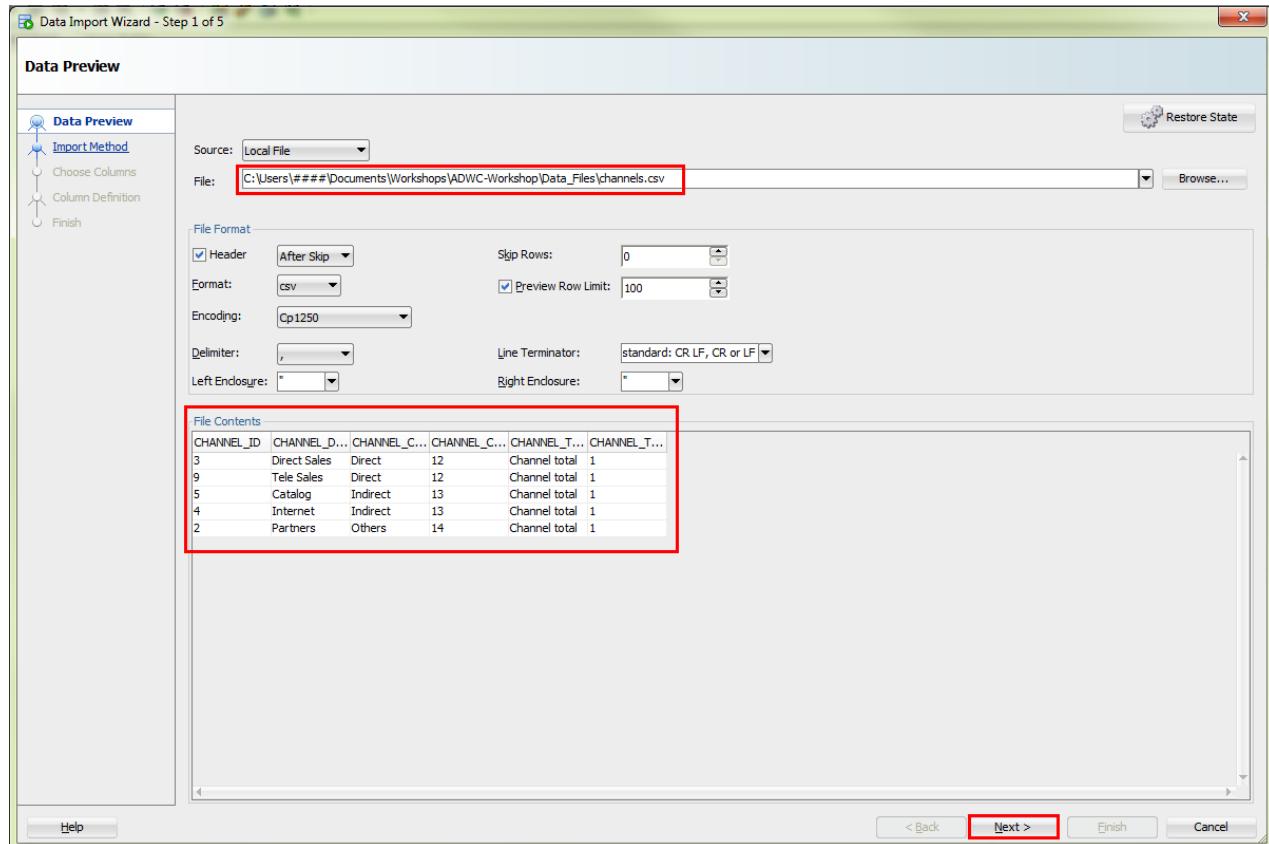
## Section 4. Loading a Local File into ADW

In this section you will load a local .csv file into your ADW instance using SQL\*Loader. The file, *channels.csv*, will be provided with the rest of the materials for this workshop.

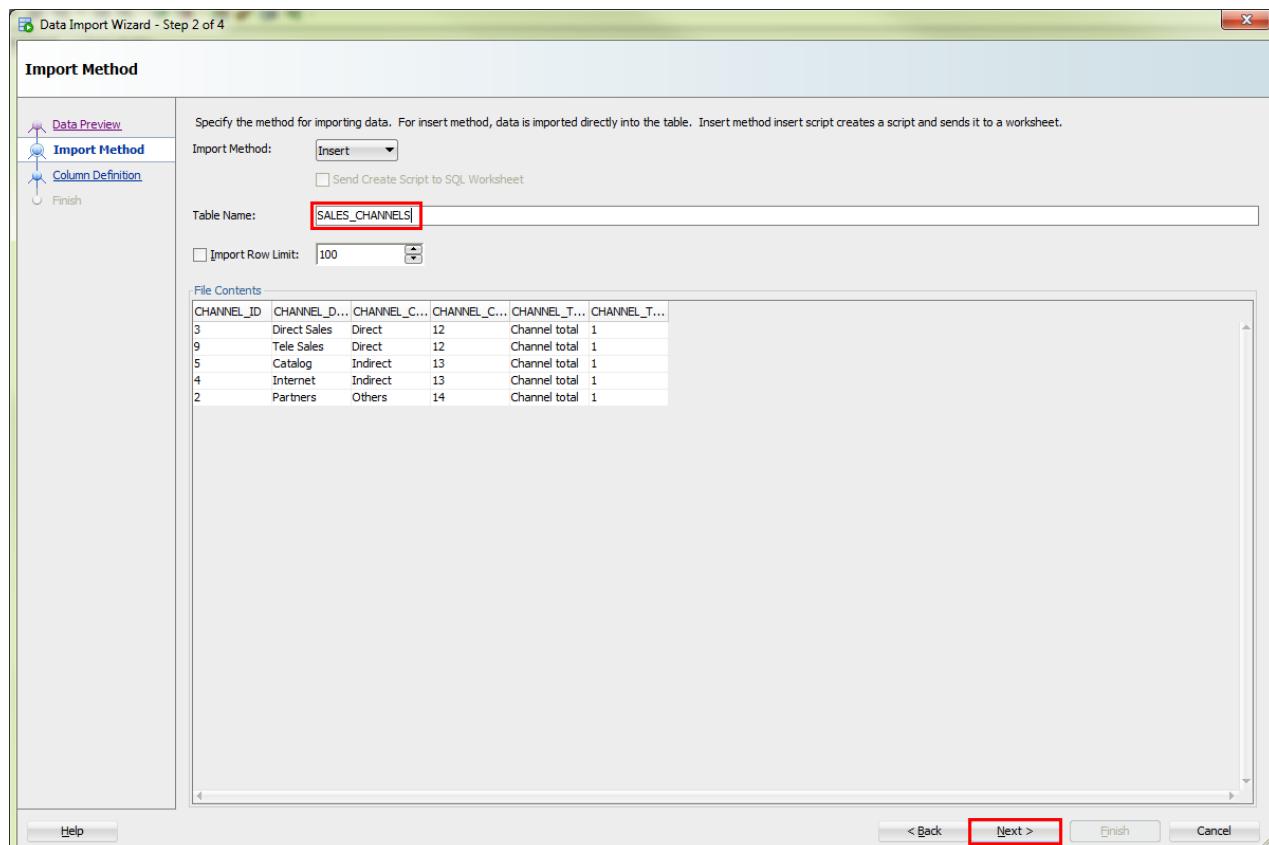
1. In SQL Developer, expand the *admin\_low* connection, right click **Tables** and click **Import Data**.



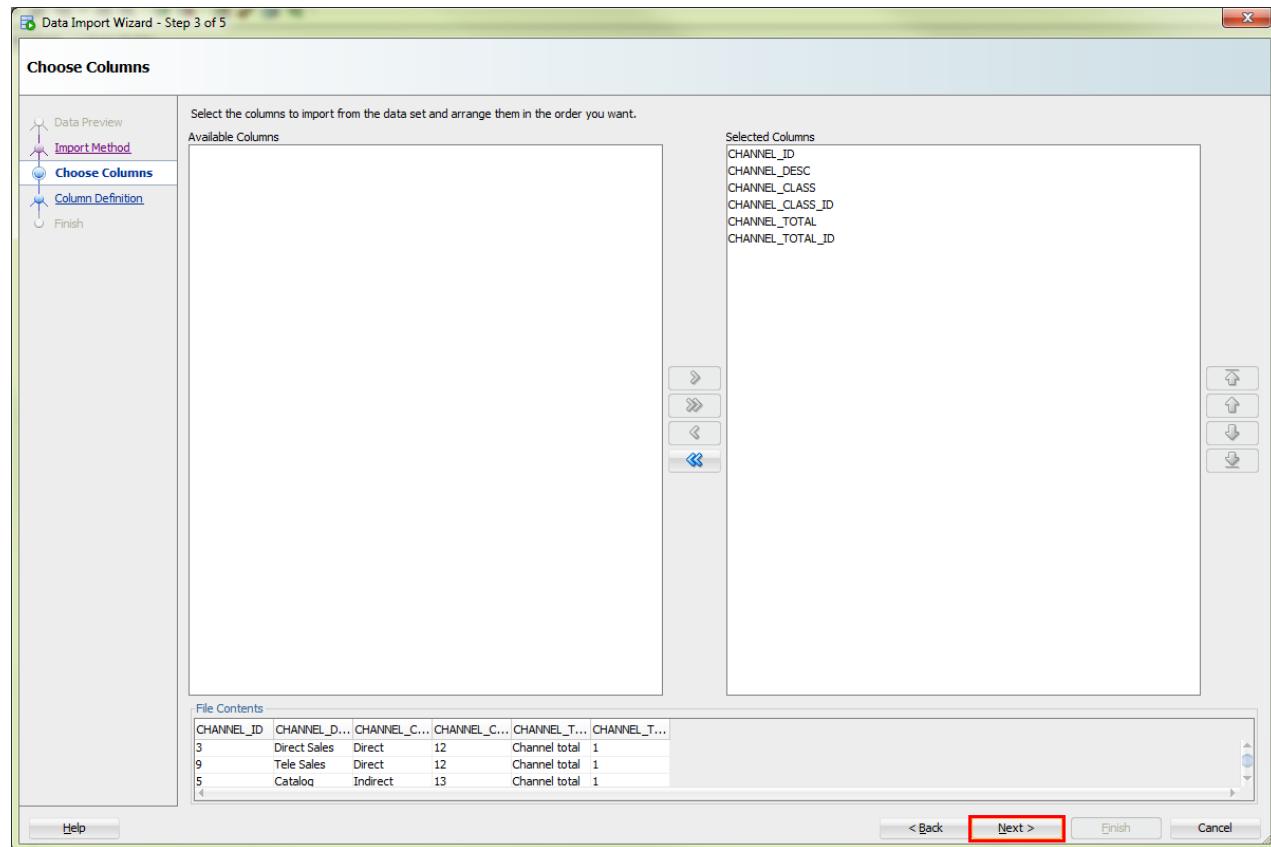
2. This will open the data import wizard. Click **Browse** and locate the *channels.csv* file on your local File System. Once you select the file you will see the file contents in the import wizard, as shown below. Click **Next**.



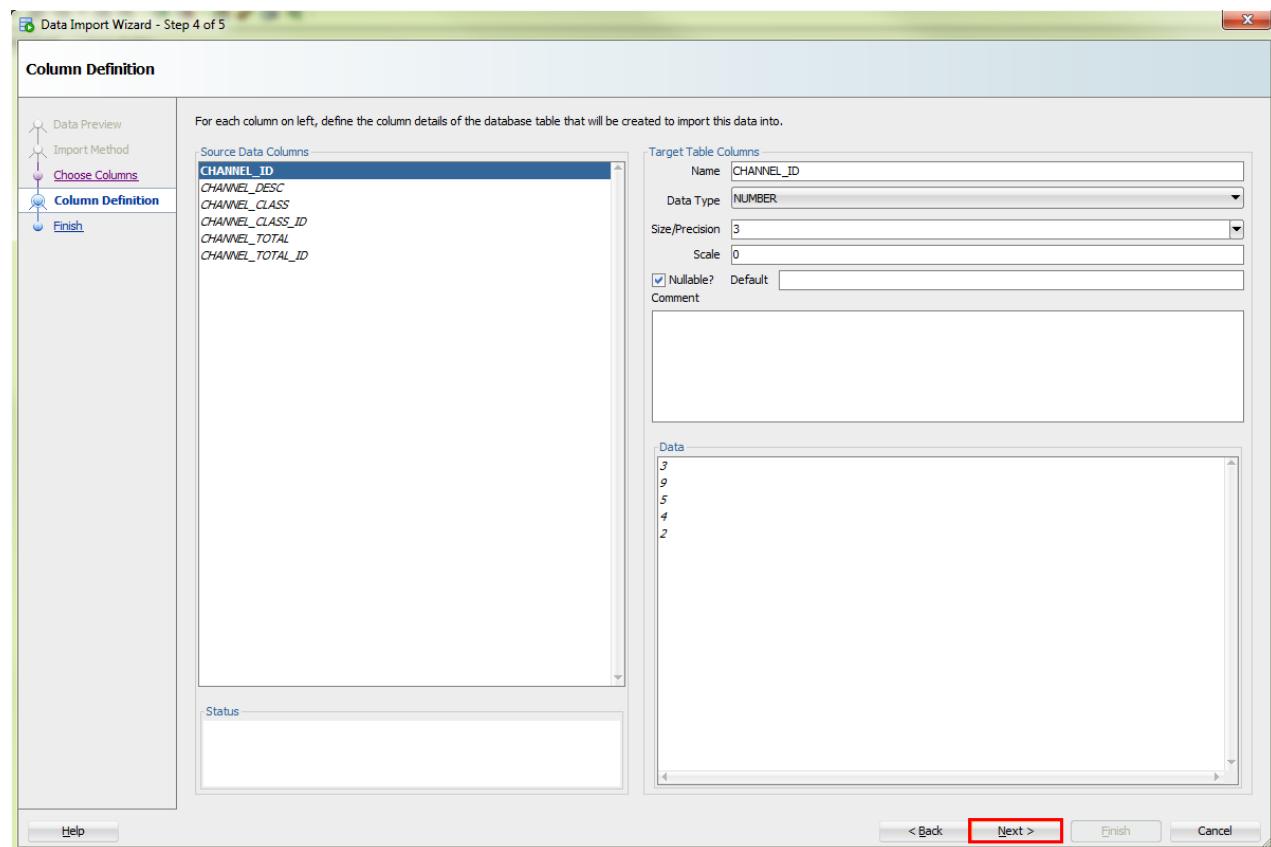
3. For the Table Name enter SALES\_CHANNELS and click **Next**.



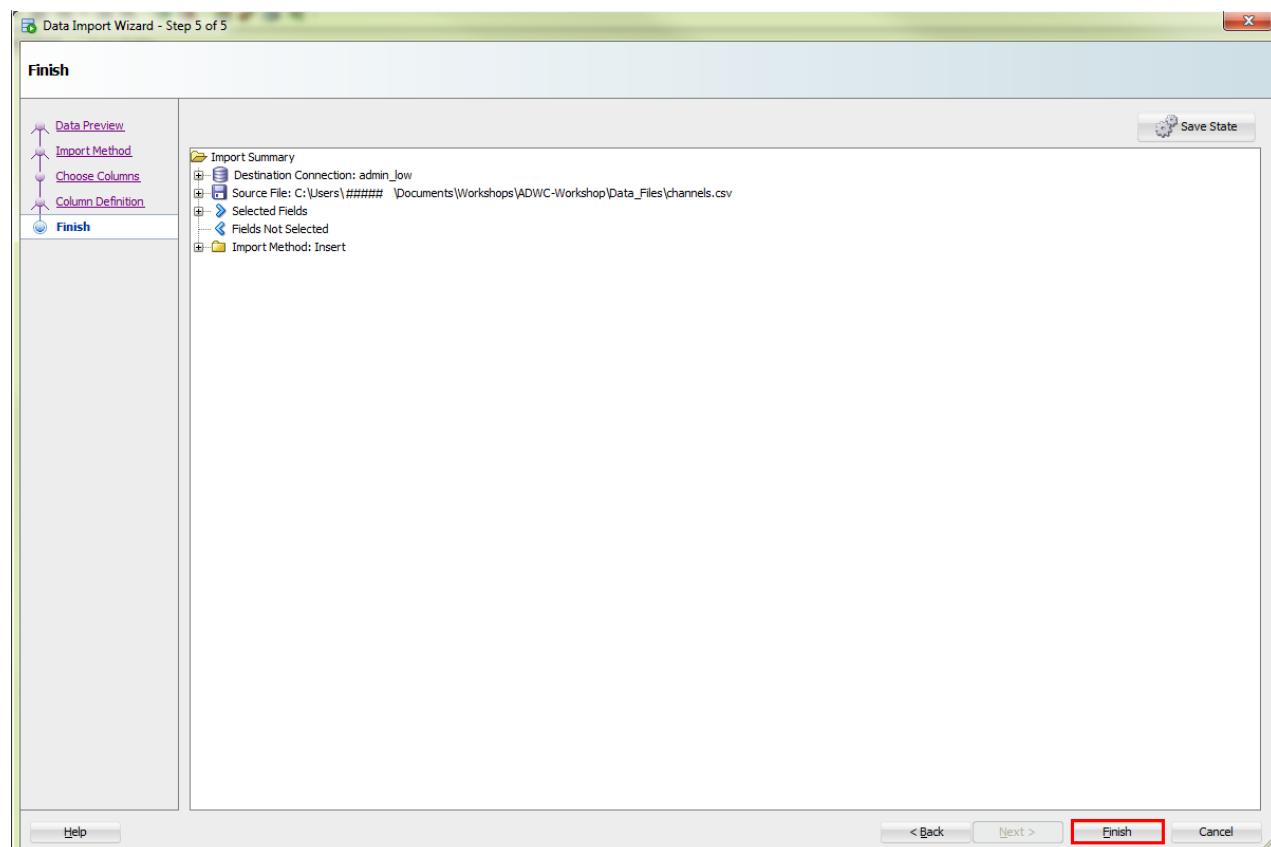
4. Select all the columns available in the data file and click **Next**.



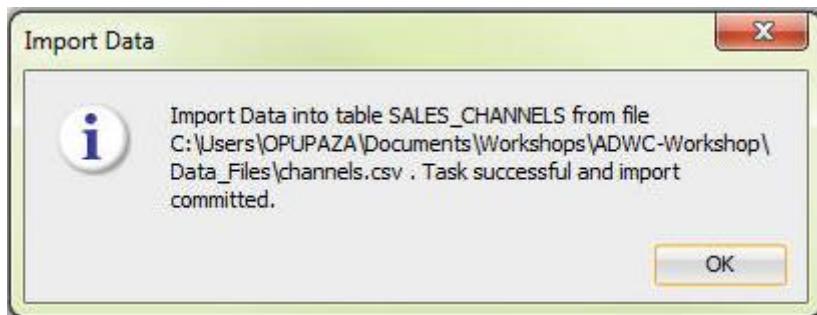
5. On the next screen, you can change the data types for each column. For this exercise, keep the defaults and click **Next**.



6. On the *Import Summary* page, review the details of the import operation and click **Finish**.



7. Once the data is loaded, a confirmation message will be displayed. Click **OK**.



8. We can now query the newly created table.

```
select * from sales_channels;
```

A screenshot of the Oracle SQL Developer interface. The top bar shows the connection name "admin\_low" and the status "0.7749998 seconds". Below it, the "Worksheet" tab is active, showing the query "select \* from sales\_channels;". To the right, the "Script Output" tab shows the results:

CHANNEL_ID	CHANNEL_DESC	CHANNEL_CLASS	CHANNEL_CLASS_ID	CHANNEL_TOTAL	CHANNEL_TOTAL_ID
3	Direct Sales	Direct		12 Channel total	1
9	Tele Sales	Direct		12 Channel total	1
5	Catalog	Indirect		13 Channel total	1
4	Internet	Indirect		13 Channel total	1
2	Partners	Others		14 Channel total	1

The status bar at the bottom indicates "Task completed in 0.775 seconds".

## Section 5. Using Data Visualization Desktop on ADW

The purpose of this section is to showcase an analytics tool on top of the powerful Autonomous Data Warehouse.

For this exercise, we chose the Oracle Data Visualization Desktop, but you will be able to connect a variety of different tools to the Autonomous Data Warehouse

The lab guide will walk you through the following steps:

- Connect a desktop analytics tool to the Autonomous Data Warehouse
- Secure a desktop client connection to Autonomous Data Warehouse
- Create a simple data visualization project with Oracle Data Visualization Desktop
- Access and gain insights from data in the Autonomous Data Warehouse

### Step 1. Installing Data Visualization Desktop

**Note:** The Autonomous Data Warehouse includes a license of Oracle Data Visualization Desktop.

1. Download the latest version of the Oracle Data Visualization Desktop from the URL below. You can select either the Windows or Mac version, based on your host operating system.  
If you already have DV Desktop installed, please consider upgrading it as there is now a new connection type for Autonomous Data Warehouse.

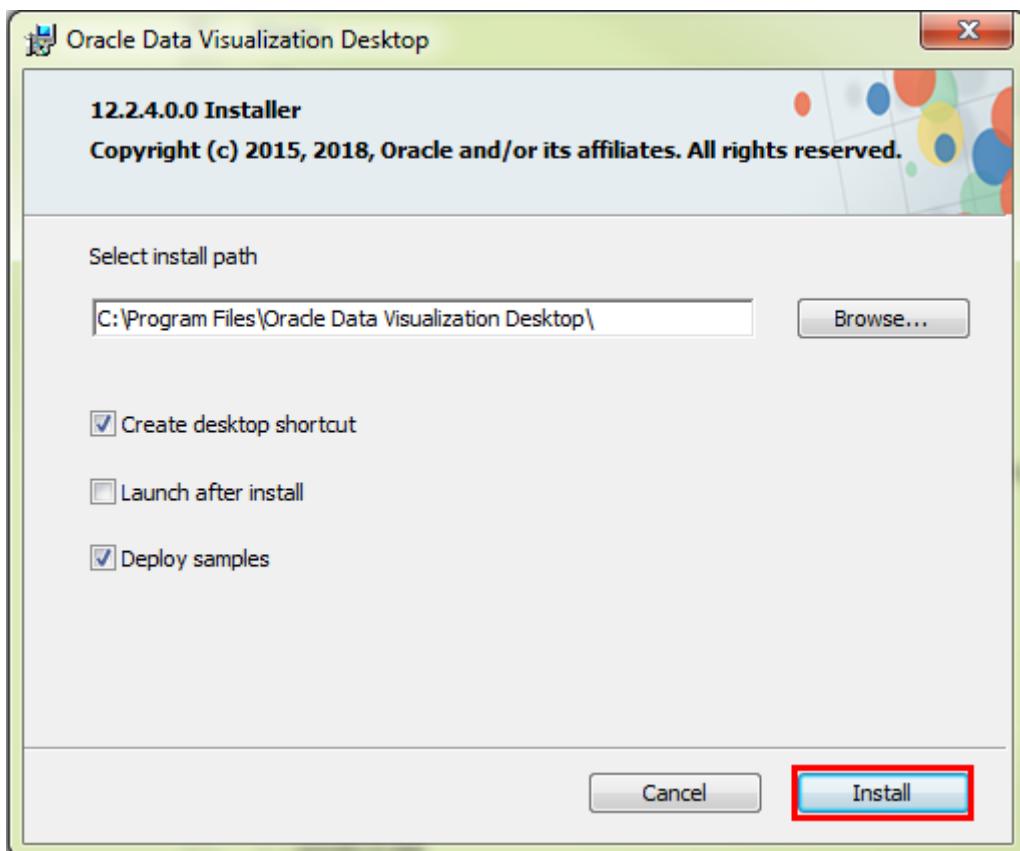
<http://www.oracle.com/technetwork/middleware/oracle-data-visualization/downloads/oracle-data-visualization-desktop-2938957.html>

The screenshot shows the Oracle Technology Network (OTN) website. At the top, there's a red header bar with the 'ORACLE' logo. To the right of the logo are navigation links for 'Menu', a search bar, and account information ('Account' and 'Country/Region'). Below the header, a grey navigation bar contains the text 'Oracle Technology Network / Middleware / Oracle Data Visualization / Downloads'. The main content area has a left sidebar with a vertical list of Oracle products: Fusion Middleware Home, Business Intelligence, Coherence, Developer Tools, GlassFish Server, Identity Management, JRockit, SOA Suite, TopLink, Tuxedo, WebCenter, WebCenter Content, WebCenter Sites, WebCenter Portal, Social Network, WebLogic Server, Web Tier, Traffic Director, Event Processing, Business Intelligence Foundation, and Foundation. The main content area is titled 'Oracle Data Visualization Desktop Downloads' and contains instructions for downloading the software, including accepting the license agreement. It also notes that limited usage rights are granted through the Oracle Technology Network License Agreement. Below this, there are two download links for 'Oracle Data Visualization Desktop 12c': one for Windows (502,892,377 bytes) and one for MAC (550,585,977 bytes). A note at the bottom states that upgrading to version 12.2.4.1.1 will mean that new project exports will be incompatible with earlier versions of Oracle Data Visualization both on-premises and in the cloud.

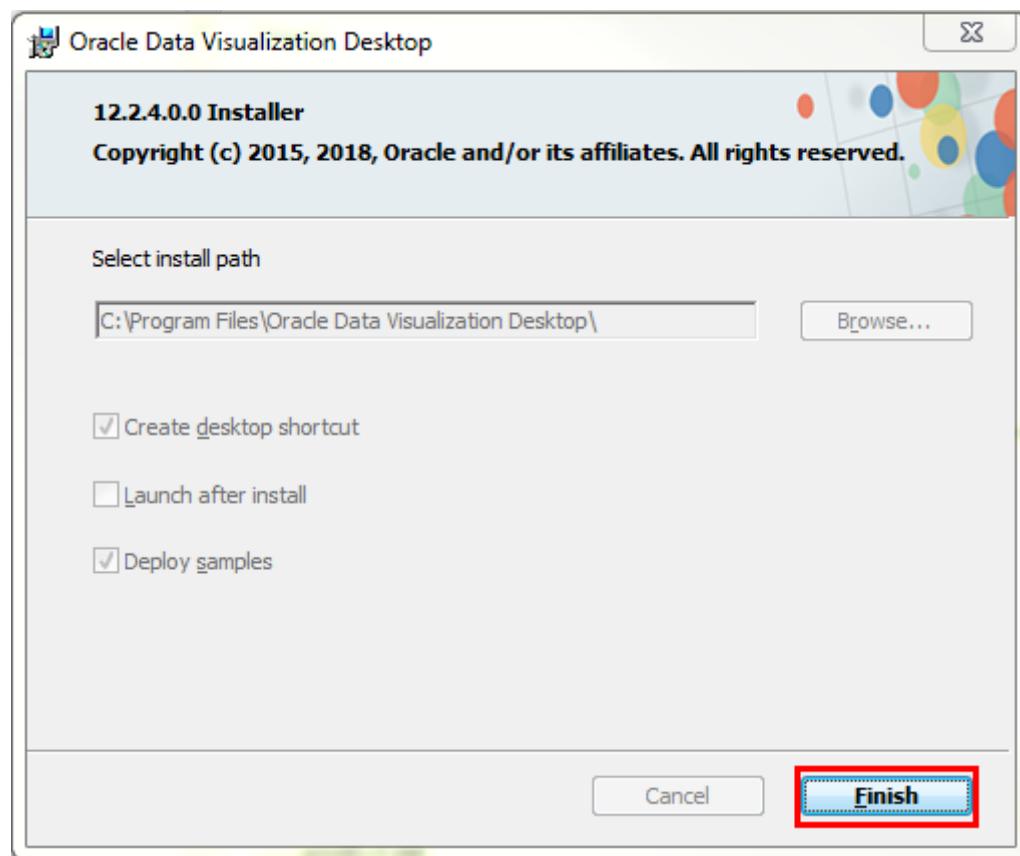
2. Save the **.zip** file on your local file system, decompress the archive and double click the **.exe** file to start the install wizard.



3. Select the install path and click **Install**.



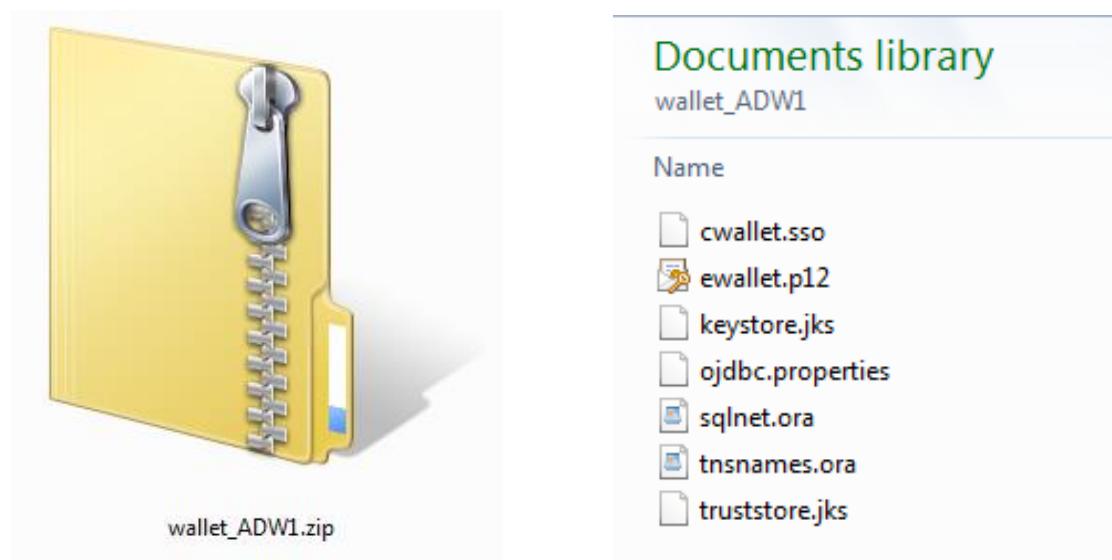
4. Wait for the install process to complete and click **Finish**.



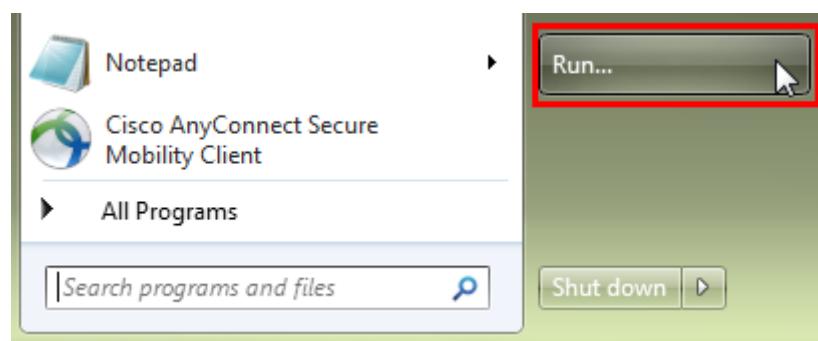
## Step 2. Securing the connection to ADW

**Note:** We want to secure the data transferred between the client application(DVD) and the ADW instance where the data is stored. Password credentials for connecting to the ADW instance can now be stored in a client-side Oracle wallet, a secure software container used to store authentication and signing credentials. This wallet usage can simplify large-scale deployments that rely on password credentials for connecting to databases. When this feature is configured, application code, batch jobs, and scripts no longer need embedded user names and passwords. Risk is reduced because such passwords are no longer exposed in the clear, and password management policies are more easily enforced without changing application code whenever user names or passwords change.

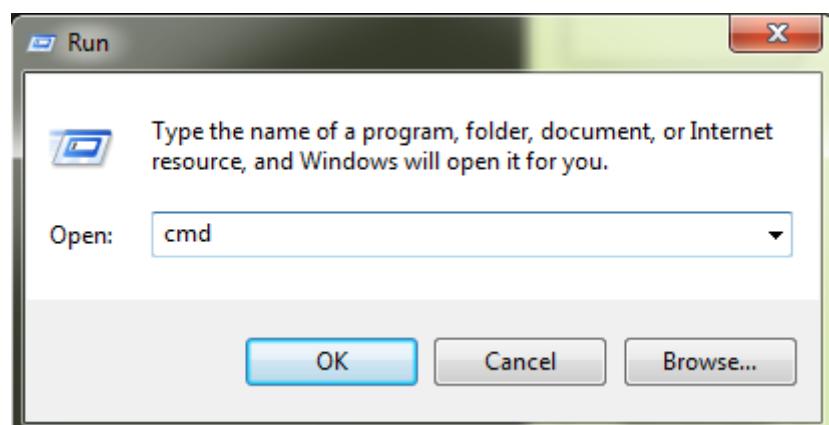
5. Go to the directory where you saved the ADW credentials for your instance. Decompress the .zip file and browse through the extracted files. In the next steps, we'll use `cwallet.sso` and `tnsnames.ora`.



6. To secure the communication between Oracle Data Visualization Desktop(DVD) and the Autonomous Data Warehouse(ADW) instance, we'll copy the wallet (cwallet.sso) to a predefined location expected by DVD.  
To do that, in Windows, click the **Start** menu and select **Run**.



7. Open the command prompt by typing **cmd**. Click **OK**.



8. DVD is expecting to find the **DWCS** directory (under %HOMEPATH%\AppData\Local\DVDesktop\components\OBIS) and have the wallet file (cwallet.sso) inside.  
 Considering the DWCS directory is not created automatically during the DVD installation, we need to create it manually.

```
cd %HOMEPATH%\AppData\Local\DVDesktop\components\OBIS
mkdir DWCS
DIR
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\OPUPAZA>cd %HOMEPATH%\AppData\Local\DVDesktop\components\OBIS
C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS>mkdir DWCS
C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS>DIR
 Volume in drive C is System
 Volume Serial Number is 4AE7-2D00

 Directory of C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS

25/04/2018  16:35    <DIR>      .
25/04/2018  16:35    <DIR>      ..
25/04/2018  16:35    <DIR>      DWCS
25/04/2018  10:47    <DIR>      instance_mapping
          0 File(s)           0 bytes
          4 Dir(s)  28,470,812,672 bytes free
```

9. Open a new terminal and switch to your ADW credentials. Copy the wallet file, **cwallet.sso**, to the DVD expected directory, **DWCS**.

```
COPY cwallet.sso C:\%HOMEPATH%\AppData\Local\DVDesktop\components\OBIS\DWCS
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\OPUPAZA>cd C:\Users\OPUPAZA\Documents\Infrastructure\ADWC_Keys\wallet_A
DW1
C:\Users\OPUPAZA\Documents\Infrastructure\ADWC_Keys\wallet_A\DW1>COPY cwallet.sso
C:\%HOMEPATH%\AppData\Local\DVDesktop\components\OBIS\DWCS
 1 file(s) copied.
```

10. To check if the file was copied correctly, switch to the first terminal and list the content of the DWCS directory.

```
cd DWCS
DIR
```

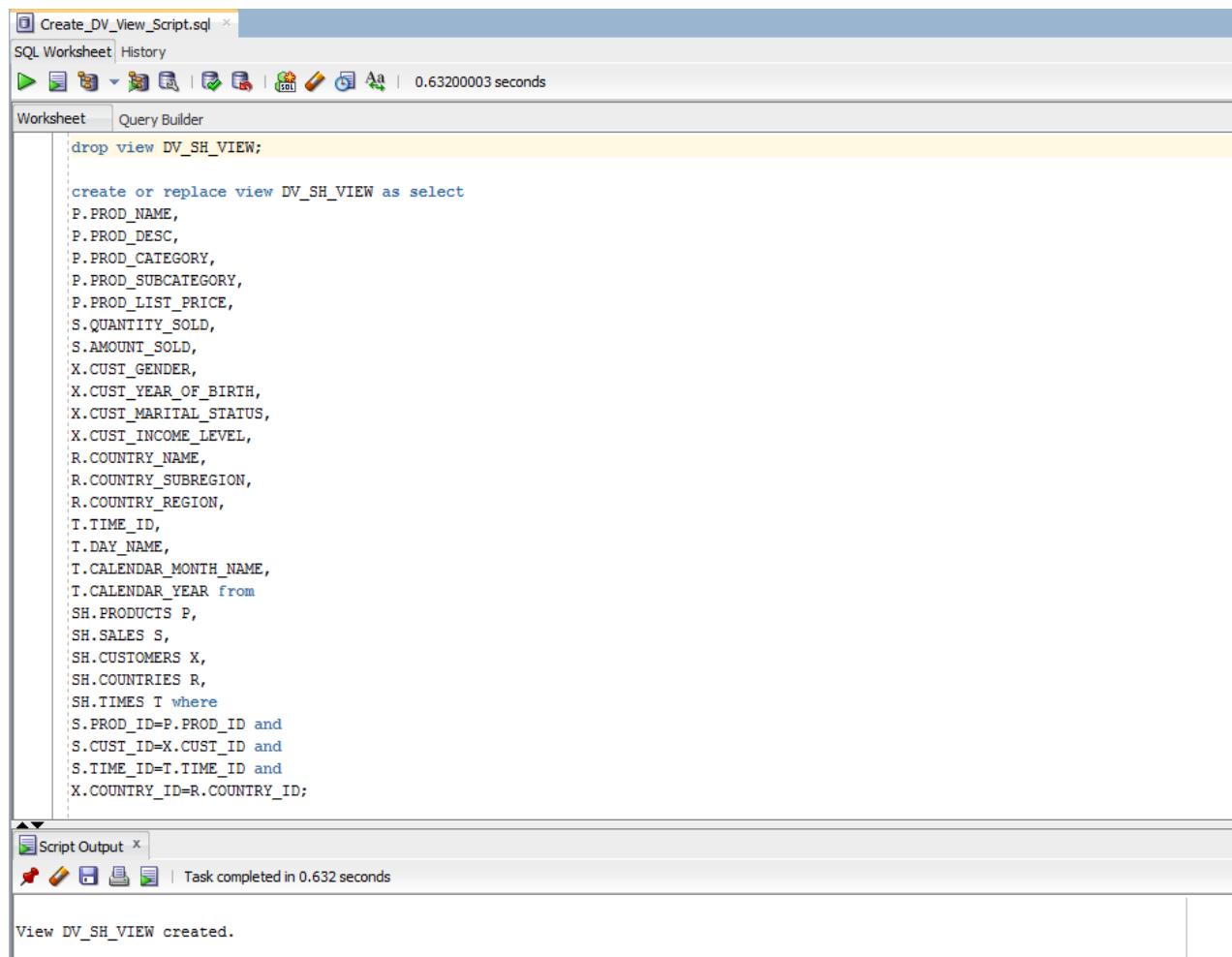
```
C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS>cd DWCS
C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS\DWCS>DIR
 Volume in drive C is System
 Volume Serial Number is 4AE7-2D00

 Directory of C:\Users\OPUPAZA\AppData\Local\DVDesktop\components\OBIS\DWCS

25/04/2018  16:50    <DIR>      .
25/04/2018  16:50    <DIR>      ..
20/04/2018  10:29           6,661 cwallet.sso
                           1 File(s)   6,661 bytes
                           2 Dir(s)  28,470,255,616 bytes free
```

### Step 3. Create the data object in SQL Developer

11. In SQL Developer, create a database view based on the tables available in the SH schema. Connect to the *admin\_low* service and open the *Create\_DV\_View\_Script.sql* file. Run the script and inspect the DV\_SH\_VIEW object.



```
drop view DV_SH_VIEW;

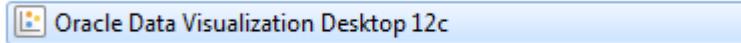
create or replace view DV_SH_VIEW as select
P.PROD_NAME,
P.PROD_DESC,
P.PROD_CATEGORY,
P.PROD_SUBCATEGORY,
P.PROD_LIST_PRICE,
S.QUANTITY SOLD,
S.AMOUNT SOLD,
X.CUST_GENDER,
X.CUST_YEAR_OF_BIRTH,
X.CUST_MARITAL_STATUS,
X.CUST_INCOME_LEVEL,
R.COUNTRY_NAME,
R.COUNTRY_SUBREGION,
R.COUNTRY_REGION,
T.TIME_ID,
T.DAY_NAME,
T.CALENDAR_MONTH_NAME,
T.CALENDAR_YEAR from
SH.PRODUCTS P,
SH.SALES S,
SH.CUSTOMERS X,
SH.COUNTRIES R,
SH.TIMES T where
S.PROD_ID=P.PROD_ID and
S.CUST_ID=X.CUST_ID and
S.TIME_ID=T.TIME_ID and
X.COUNTRY_ID=R.COUNTRY_ID;
```

Script Output | Task completed in 0.632 seconds

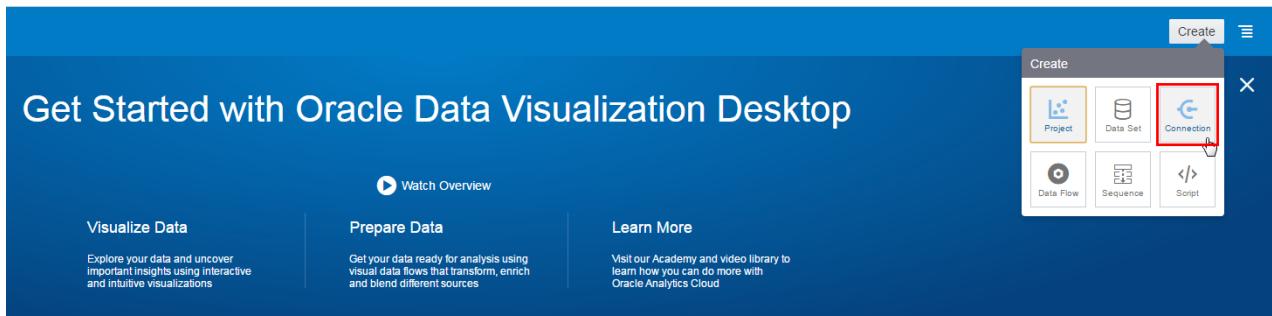
View DV\_SH\_VIEW created.

**Step 4. Connect DVD to ADW**

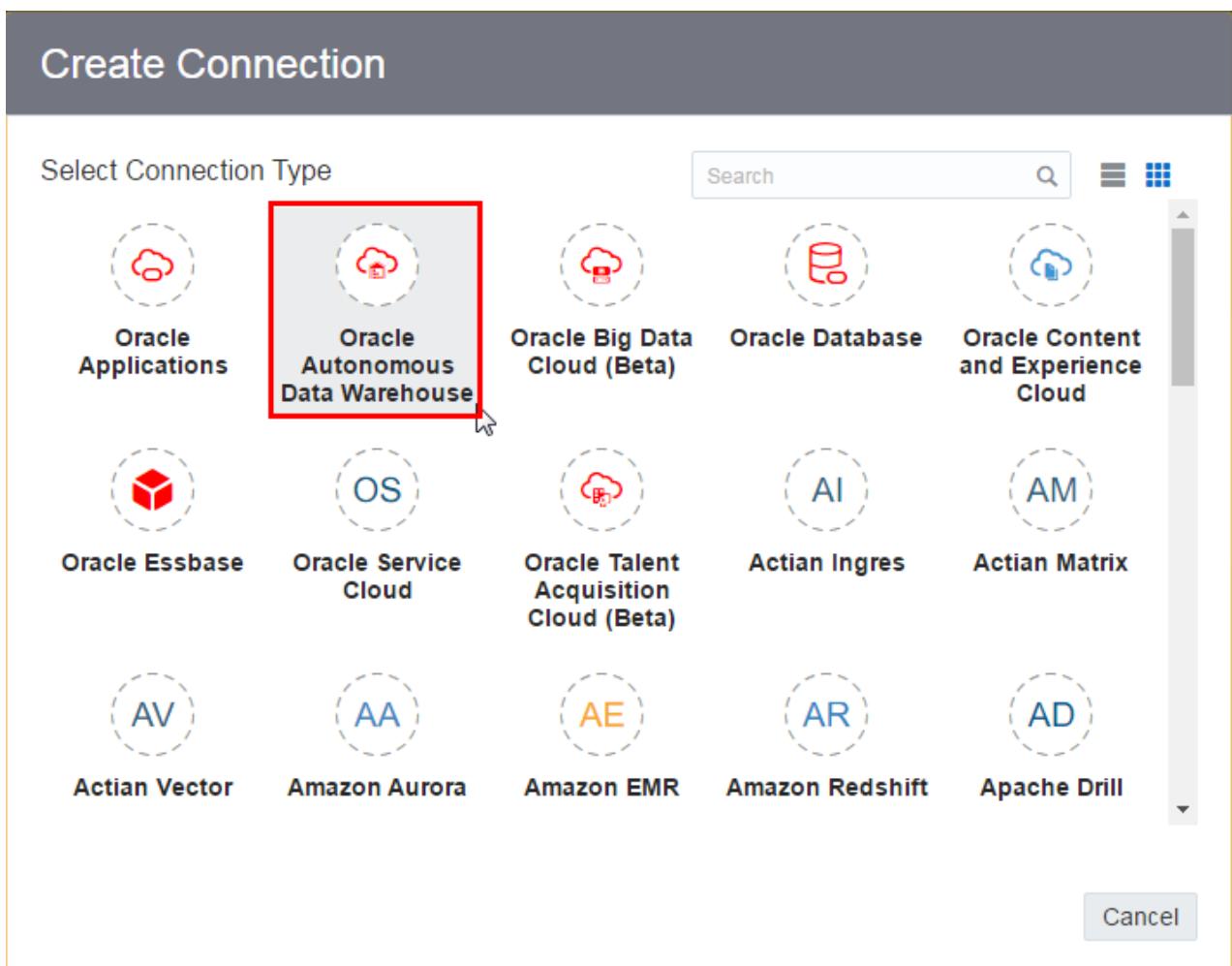
12. Start Oracle Data Visualization Desktop from the Windows Start Menu.



13. On the Home page, click **Create** and select **Connection**.



14. From the list of Connectors, select **Oracle Autonomous Data Warehouse**.



15. To fill in the database details, open the *tnsnames.ora* file. It will contain three entries for each of the three services: HIGH, MEDIUM, LOW. Pick one of them and note down the **host**, **port** and **service\_name**.

To establish the connection to the ADW instance you need to complete the following fields:

**New Connection Name: SALES\_HISTORY**

**Host:** <available in tnsnames.ora>

**Port:** <available in tnsnames.ora>

**Username:** admin

**Password:** <password provided during ADW creation>

**Service Name:** <available in tnsnames.ora>

Click **Save**.

**Create Connection**

Oracle Autonomous Data Warehouse Cloud

Requires Wallet ⓘ

\*New Connection Name

\*Host

\*Port

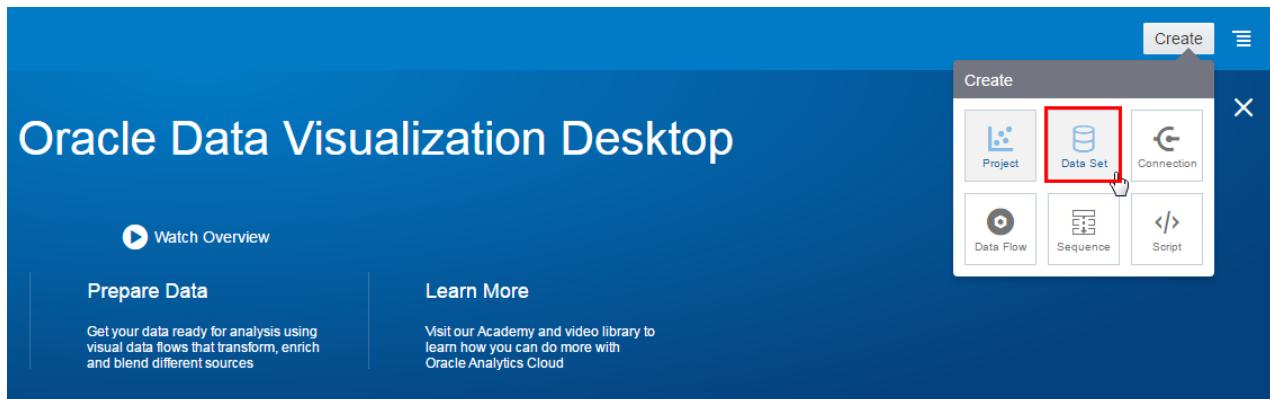
\*Username

\*Password

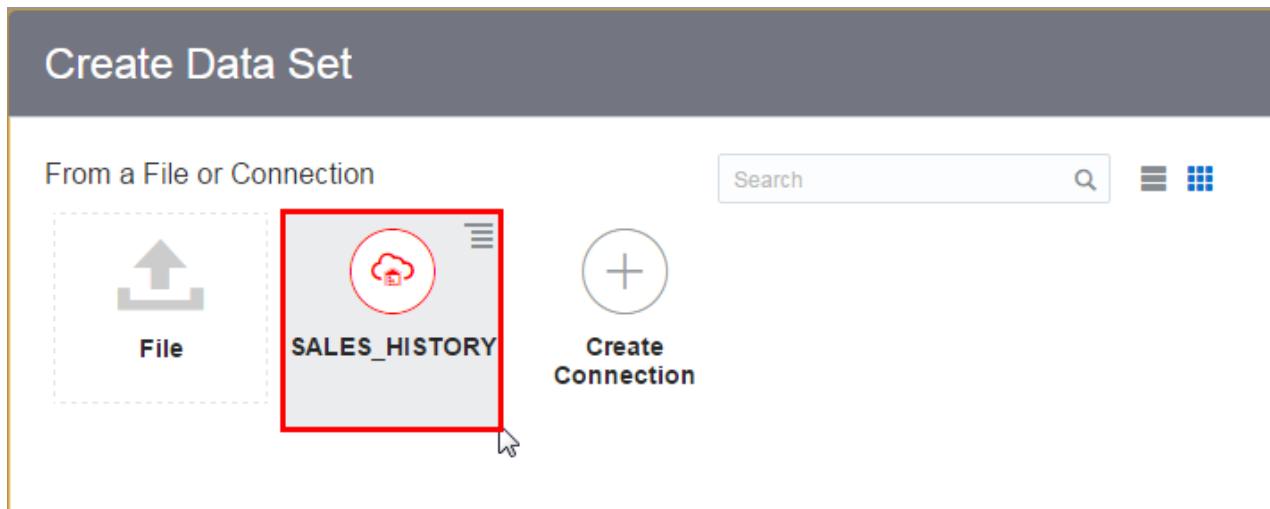
\*Service Name

**Save** **Cancel**

16. Once the connection is created, we can proceed with creating the data set. Click **Create** and select **Data Set**.



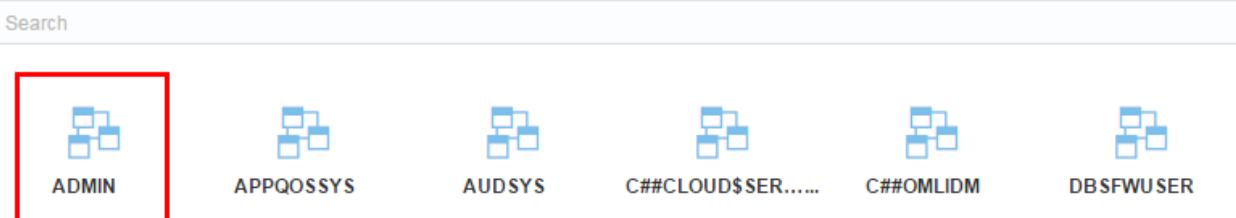
17. Select the **SALES\_HISTORY** connection that was created in a previous step.



18. Click the **Admin** database user.

The screenshot shows the Oracle Data Visualization Desktop interface. At the top, it says "ORACLE Data Visualization Desktop". Below that is a blue header bar with the title "Untitled - Data Set". The main area contains a flow diagram: a blue rounded rectangle with a cloud icon labeled "SALE..." is connected by a line to an oval labeled "Filter", which is then connected to another rounded rectangle labeled "Untitled".

SALES\_HISTORY



19. The list of available objects under the Admin schema is displayed. Click **DB\_SH\_VIEW**.

The screenshot shows the same interface as before, but now under the "SALES\_HISTORY > ADMIN" context. A search bar is present. Below it, a list of objects includes "DV\_SH\_VIEW" (highlighted with a red box) and "SALES\_CHAN". A cursor is hovering over "DV\_SH\_VIEW".

20. First click the **Add All** button on the left panel to populate the right panel. Fill in the Name as **SALES\_HISTORY** and click **Add**.

The screenshot shows the Oracle Data Visualization Desktop interface with the "SALES\_HISTORY > ADMIN > DV\_SH\_VIEW" context. On the left, there's a list of columns: PROD\_NAME, AMOUNT\_SOLD, CALENDAR\_MONTH\_NAME, CALENDAR\_YEAR, COUNTRY\_NAME, COUNTRY\_REGION, COUNTRY\_SUBRE, CUST\_GENDER, CUST\_INCOME\_LEVEL, CUST\_MARITAL\_ST, CUST\_YEAR\_OF\_B, DAY\_NAME, PROD\_CATEGORY, and PROD\_DESC. A "Search" field and a "Q" button are also present. On the right, there's a "Selections (18/18)" panel showing the selected objects: ADMIN, DV\_SH\_VIEW, PROD\_NAME, AMOUNT\_SOLD, CALENDAR\_MONTH\_NAME, and CALENDAR\_YEAR. A "Name" field is set to "SALES\_HISTORY". Buttons for "Select Columns", "Enter SQL", "Remove All", and "Remove Selected" are at the top of the right panel. A message at the bottom says "No data to display."

21. After running for a couple of minutes, the data set will be populated with data.

PROD_NAME	AMOUNT_SOLD	CALENDAR_MONTH_NAME	CALENDAR_YEAR	COUNTRY_NAME	COUNTRY_REGION	COUNTRY_SUBREGION	CUST_GENDER	CUST_INCOME_LEVEL	CUST_M
5MP Telephoto Digital Camera	1,205.99	December	1998	Australia	Oceania	Australia	F	E: 90,000 - 109,999	
5MP Telephoto Digital Camera	1,205.99	December	1998	Singapore	Asia	Asia	F	F: 110,000 - 129,999	
5MP Telephoto Digital Camera	1,205.99	December	1998	United States of America	Americas	Northern America	F	I: 170,000 - 189,999	single
5MP Telephoto Digital Camera	1,205.99	December	1998	United States of America	Americas	Northern America	M	E: 90,000 - 109,999	
5MP Telephoto Digital Camera	1,205.99	December	1998	United States of America	Americas	Northern America	M	F: 110,000 - 129,999	
5MP Telephoto Digital Camera	1,205.99	January	1998	Canada	Americas	Northern America	F	F: 110,000 - 129,999	single
5MP Telephoto Digital Camera	1,205.99	October	1998	Canada	Americas	Northern America	F		single
5MP Telephoto Digital Camera	1,205.99	October	1998	Germany	Europe	Western Europe	M	B: 30,000 - 49,999	married
5MP Telephoto Digital Camera	1,205.99	October	1998	Japan	Asia	Asia	F	E: 90,000 - 109,999	married
5MP Telephoto Digital Camera	1,205.99	October	1998	Japan	Asia	Asia	F	I: 170,000 - 189,999	single
5MP Telephoto Digital Camera	1,205.99	October	1998	United Kingdom	Europe	Western Europe	M	C: 50,000 - 69,999	married
5MP Telephoto Digital Camera	1,205.99	October	1998	United States of America	Americas	Northern America	F	C: 50,000 - 69,999	
5MP Telephoto Digital Camera	1,205.99	October	1998	United States of America	Americas	Northern America	F	C: 50,000 - 69,999	

22. Next, let's see how can we modify the data set and change the data types for some of the columns.  
Click the main menu and select the **Data** option.

PROD_NAME	AMOUNT_SOLD	CALENDAR_MONTH_NAME
5MP Telephoto Digital Camera	1,205.99	December
5MP Telephoto Digital Camera	1,205.99	December
5MP Telephoto Digital Camera	1,205.99	December
5MP Telephoto Digital Camera	1,205.99	December
5MP Telephoto Digital Camera	1,205.99	December
5MP Telephoto Digital Camera	1,205.99	January
5MP Telephoto Digital Camera	1,205.99	October
5MP Telephoto Digital Camera	1,205.99	October

23. Right-click the SALES\_HISTORY data set and click **Inspect**.

Type	Name	Description
External Data Set	SALES_HISTORY	
Uploaded from Sample Order Lines.xlsx.	Sample Order Lines	
Uploaded from Sample States.xlsx.	Sample States	

24. Next, we are going to override the data types for two columns recognized as numeric and correctly set them as attributes: CALENDAR\_YEAR and CUST\_YEAR\_OF\_BIRTH. Change the *Treat As* field as **Attribute** for the two columns and click **OK**.

### Data Set - SALES\_HISTORY

General

Name	SALES_HISTORY	Data				
Last Updated	25/04/2018, 11:17:40	Name	Data Type	Treat As	Aggregation	Hidden
Description	External Data Set	PROD_NA...	Text	Attribute		
Type	SQL Query	AMOUNT_...	Number	Measure	Sum	
Connection	SALES_HISTORY	CALENDAR...	Text	Attribute		
Database Type	Oracle Autonomous Data Warehouse Cloud	CALENDAR...	Number	Measure ▾	Sum ▾	
Query Mode	Auto	COUNTRY...	Text	Attribute		
		COUNTRY...	Text	Attribute		
		COUNTRY...	Text	Attribute		

**OK**      **Cancel**

25. When prompted to confirm the changes to the Data Set, click **Yes**.

General

Name	SALES_HISTORY	Data				
Last Updated	25/04/2018, 11:17:40	Name	Data Type	Treat As	Aggregation	Hidden
Description	External Data Set	COUNTRY...	Text	Attribute		
Type	SQL Query	CUST_GEO...	Text	Attribute		
Connection	SALES_HISTORY	CUST_INC...	Text	Attribute		
Database Type	Oracle Autonomous Data Warehouse Cloud	CUST_MA...	Text	Attribute		
Query Mode	Auto	CUST_YEA...	Number	Attribute		
		DAY_NAME	Text	Attribute		

**Yes**      **No**

**⚠ Are you sure you want to update SALES\_HISTORY?**

26. Once the *Data Set* has been updated successfully, we are ready to create our first project. Click on **Create Project**.

27. Select the *SALES\_HISTORY* Data Set and click the **Add to Project** button.

#### **Step 5. Create a new project in Data Visualization**

**Note:** No matter what your role is in the organization, access to timely data can provide greater insights to improve the performance of your business. Whether you're creating a data warehouse or data marts for yourself or others, Autonomous Data Warehouse is making it far simpler than ever before. Easy, fast and elastic. This small project demonstrates this. This is how business users would interact with the Autonomous Data Warehouse.

28. We will start by browsing the data available in our Data Set. To do that, click on the **Prepare** button and then select the **Data Tiles** option.

The screenshot shows the Oracle Data Visualization Desktop interface. The top navigation bar includes 'ORACLE Data Visualization Desktop', 'Untitled - Project', 'Prepare' (highlighted in red), 'Visualize', 'Narrate', 'Save', and a help icon. On the left, the 'Data Elements' pane lists columns from the 'SALES\_HISTORY' dataset: PROD\_NAME, AMOUNT SOLD, CALENDAR\_MONTH\_NAME, CALENDAR\_YEAR, COUNTRY\_NAME, and COUNTRY\_REGION. The main area displays a table with five rows of data. A context menu is open over the last row, with 'Data Table' and 'Data Tiles' options visible. The 'Data Tiles' option is highlighted with a red box.

29. Explore the various charts generated for your data and then click the **Visualize** button to come back to the blank canvas.

The screenshot shows the Oracle Data Visualization Desktop interface after selecting 'Data Tiles'. It displays six charts generated from the data set:

- COUNTRY\_REGION**: Bar chart showing sales volume by region: America (~500K), Europe (~250K), Asia (~100K), Oceania (~20K), Middle East (~10K).
- CUST\_MARITAL\_STATUS**: Bar chart showing marital status distribution: single (~450K), married (~10K), divorced (~10K), widow (~10K), NeverM (~10K), Divor (~10K), Widowed (~10K), Separ (~10K), MarAF (~10K), NULL (~10K).
- PROD\_CATEGORY**: Bar chart showing product category sales: Software/Other (~400K), Peripherals and Accessories (~250K), Electronics (~100K), Photo (~50K), Hardware (~10K).
- COUNTRY\_SUBREGION**: Bar chart showing sales by subregion: Northern America (~500K), Western Europe (~250K), Asia (~100K), Australia (~10K), Southern America (~10K), Eastern Europe (~10K), Middle East (~10K).
- COUNTRY\_NAME**: Bar chart showing sales by country: United States of America (~500K), Germany (~10K), Japan (~10K), United Kingdom (~10K), Italy (~10K), Australia (~10K), France (~10K), Singapore (~10K), Canada (~10K), Spain (~10K), OTHER (~10K).
- CUST\_YEAR\_OF\_BIRTH**: Histogram showing birth year distribution from 1915 to 1990. The distribution peaks around 1945-1955.

The top navigation bar includes 'ORACLE Data Visualization Desktop', 'Untitled - Project', 'Prepare' (highlighted in red), 'Visualize' (highlighted in red), 'Narrate', 'Save', and a help icon. The 'Data Elements' pane on the left is identical to the previous screenshot.

30. In the end, we will create a very simple visualization. Multi-select (ctrl+click) the five Data Elements within the SALES\_HISTORY data set:  
PROD\_NAME, AMOUNT\_SOLD, CALENDAR\_YEAR, PROD\_CATEGORY, and QUANTITY\_SOLD.  
Drag the five selected columns to the middle of the screen.  
Based on the selected data, Oracle Data Visualization Desktop will choose a default visualization.  
In this case it's a *Scattered Chart*.

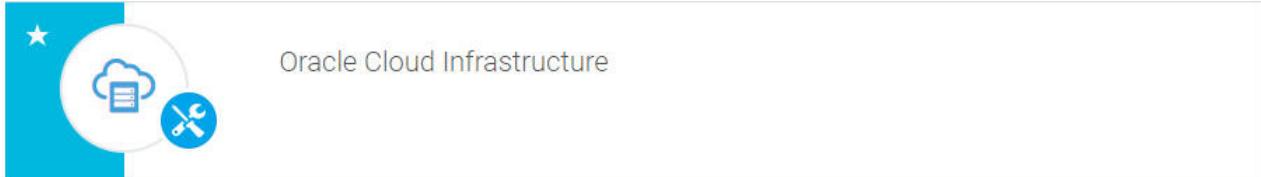


**Note:** In just a few steps, you can have a powerful tool that can further bring your data to life and allow you to make data-driven decisions. Combined with the agility of the Autonomous Data Warehouse, you will be able to quickly scale the solution to meet your growing data and end users demand.

## Section 6. Generating the Swift Password for Object Storage

Swift is the OpenStack object store service. The Swift Password is a special password that Oracle provides and is associated with your Console user login. These credentials allow you to access data stored in Object Storage.

1. From your Cloud Dashboard, click **Oracle Cloud Infrastructure**. This will open the OCI Console.



2. From the top right menu, click **Identity->Users**. This will display the list of users available on the cloud account.

A screenshot of the Oracle Cloud Infrastructure Identity &amp; Access Management (IAM) service. The top navigation bar shows "oracleidentitycloudservice/cloud.admin" and tabs for Home, Identity, Compute, Database, Networking, Storage, Audit, and Email. The "Identity" tab is highlighted with a red box. A dropdown menu for "Identity" is open, showing options: Home, Groups, Dynamic Groups, Policies, Service Limits, Compartments, Federation, and Tag Namespaces. The "Users" option is highlighted with a red box and has a cursor icon over it. To the left, there's a "View Health Dashboard" section with a clipboard icon and a "Service Health Status" button. To the right, there's a "Getting Started Guide" section with a "Getting Started Guide" button. At the bottom, there's a "Visit Oracle Blogs" section with a speech bubble icon and a "Learn more about Oracle Cloud from our product experts." link.

3. From the list of users, click the one you want to use. In this example, the user is *api.user*.

4. On the user details page, on the left menu, click **Swift Passwords**.

5. To create a new Swift password, click **Generate Password**.

6. First you need to fill in a short description and then click **Generate Password**.

Generate Password [help](#) [cancel](#)

DESCRIPTION

ADW Swift PW

**Generate Password**

7. The resulting password is a string of characters which together with the `api.user` will allow you to access the Object Storage. Click **Copy** and make sure you save it, as this password will no longer be accessible from the UI.

Generate Password [help](#) [close](#)

GENERATED PASSWORD

yg/...

Copy this password for your records. It will not be shown again.

**Copy**

**Close**

## Section 7. Data Loading into Cloud Object Storage

In this section, you will prepare the Object Storage as an intermediate step to loading the data into the ADW instance. To do that, you will create a new storage bucket and load the data files into it.

1. Navigate to **Storage → Object Storage** to open the Object Storage Console.

The screenshot shows the Oracle Cloud Infrastructure console interface. At the top, there is a navigation bar with links for Home, Identity, Compute, Database, Networking, Storage (which is highlighted with a red box), Support, Documentation, Audit, and Email. Below the navigation bar, there are two main sections: 'View Health Dashboard' and 'Getting Started'. The 'View Health Dashboard' section contains a button labeled 'Service Health Status'. The 'Getting Started' section contains a button labeled 'Getting Started Guide'. To the right of these sections is a vertical sidebar with links for Block Volumes, Boot Volumes, Backups, Object Storage (which is highlighted with a red box and has a hand cursor icon over it), File Systems, and Data Transfer.

2. Create a new storage bucket to store your data files. Click **Create Bucket**.

The screenshot shows the 'Buckets in Demo Compartment' page. On the left, there is a sidebar with a 'Storage' heading and links for Block Volumes, Boot Volumes, Backups, Object Storage (which is highlighted with a blue box), File Systems, and Data Transfers. The main area displays a table with one row for 'Rex-Bucket1'. This row includes a blue circular icon with a white letter 'B', the bucket name 'Rex-Bucket1', and the creation timestamp 'Created: Mon, 19 Mar 2018 06:56:48 GMT'. At the top of the main area, there is a red box around the 'Create Bucket' button. Navigation arrows for paginating through more buckets are visible at the bottom right.

3. Fill in *DEMO\_DATA* as the bucket name and click **Create Bucket**.

Create Bucket [help](#) [cancel](#)

Specify the storage tier for this bucket. Storage tier for a bucket can only be specified during creation.

BUCKET NAME

STORAGE TIER  STANDARD  ARCHIVE

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE	TAG KEY	VALUE
None (apply a free-form tag)		

[Create Bucket](#)

4. Click the newly created bucket, **DEMO\_DATA**.

Storage **Buckets in Demo Compartment**

Block Volumes	Created: Mon, 16 Apr 2018 12:35:53 GMT
Boot Volumes:	
Backups	
<b>Object Storage</b>	
File Systems	
Data Transfers	

[Create Bucket](#)

<b>B</b>	<b>DEMO_DATA</b>	Created: Mon, 16 Apr 2018 12:35:53 GMT
<b>B</b>	<a href="#">Rex-Bucket1</a>	Created: Mon, 19 Mar 2018 06:56:48 GMT

5. Under the *Objects* section, click **Upload Object**.

Resources **Objects**

Objects (0)	Upload Object
Pre-Authenticated Requests (0)	No Objects were found.

6. Click **Browse** and navigate to the location where your data files were saved.  
(The data files will be provided by the host.)

Upload Object [help](#) [cancel](#)

Choose a file from your computer:

**Browse**    

OR

Drop a file here...

[Upload Object](#)

7. Start with the first data file, *prod1v3.dat*, and continue with the rest of the files in the folder.

Documents library

Data\_Files

Name
<input checked="" type="checkbox"/> prod1v3.dat
<input type="checkbox"/> coun_v3.dat
<input type="checkbox"/> sale1v3.dat
<input type="checkbox"/> time_v3.dat
<input type="checkbox"/> prom1v3.dat
<input type="checkbox"/> dmsal_v3.dat
<input type="checkbox"/> dem1v3.dat
<input type="checkbox"/> cust1v3.dat
<input type="checkbox"/> costs.dat
<input type="checkbox"/> chan_v3_error.dat
<input type="checkbox"/> chan_v3.dat
<input checked="" type="checkbox"/> channels.csv

8. Click **Upload Object**.

Upload Object [help](#) [cancel](#)

Choose a file from your computer: [Browse](#)

OR

Drop a file here...

OBJECT NAME

14.7 KB

**Upload Object** 

9. Repeat the upload steps for the rest of the data files.

# Objects

[Upload Object](#)

chan\_v3.dat

chan\_v3\_error.dat

channels.csv

costs.dat

coun\_v3.dat

cust1v3.dat

dem1v3.dat

dmsal\_v3.dat

prod1v3.dat

prom1v3.dat

sale1v3.dat

time\_v3.dat

## Section 8. Data Loading from Cloud Object Storage using SQL Developer Import Wizard

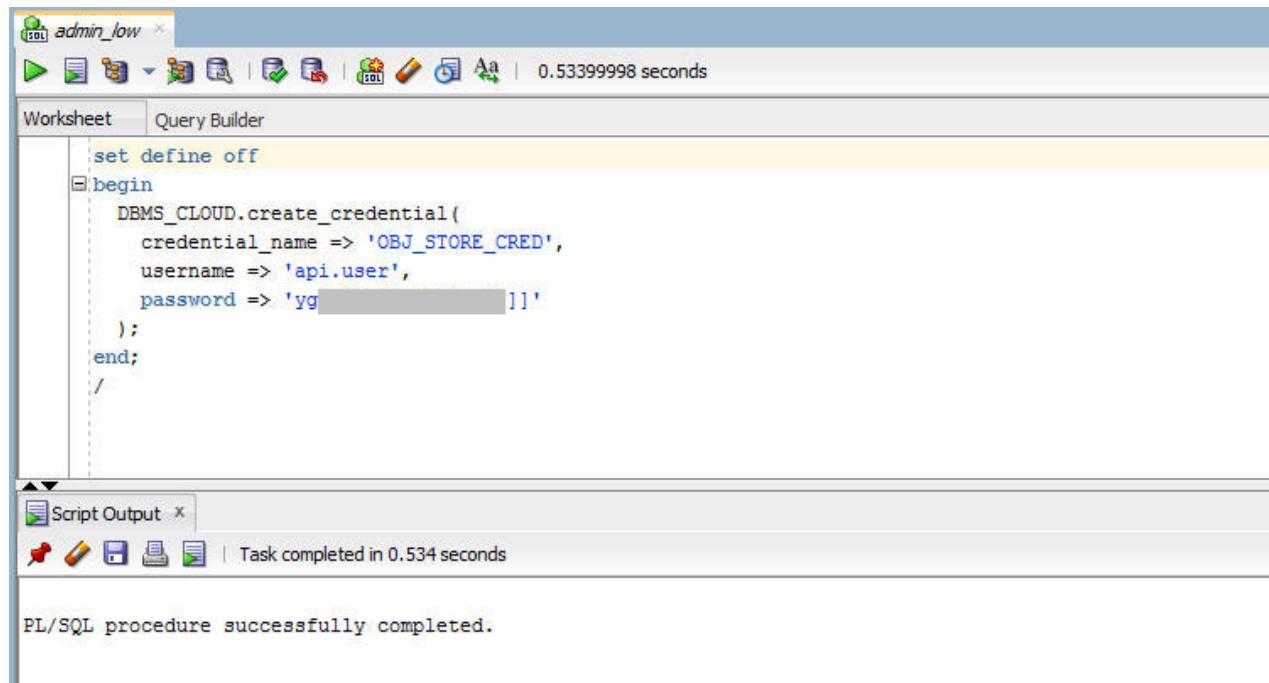
In this section you will see a demo of data loading from the Cloud Object Storage using SQL Developer data import wizard.

Beginning with SQL Developer 18.1 the data import wizard supports loading from files in the Object Store straight into your Autonomous Data Warehouse.

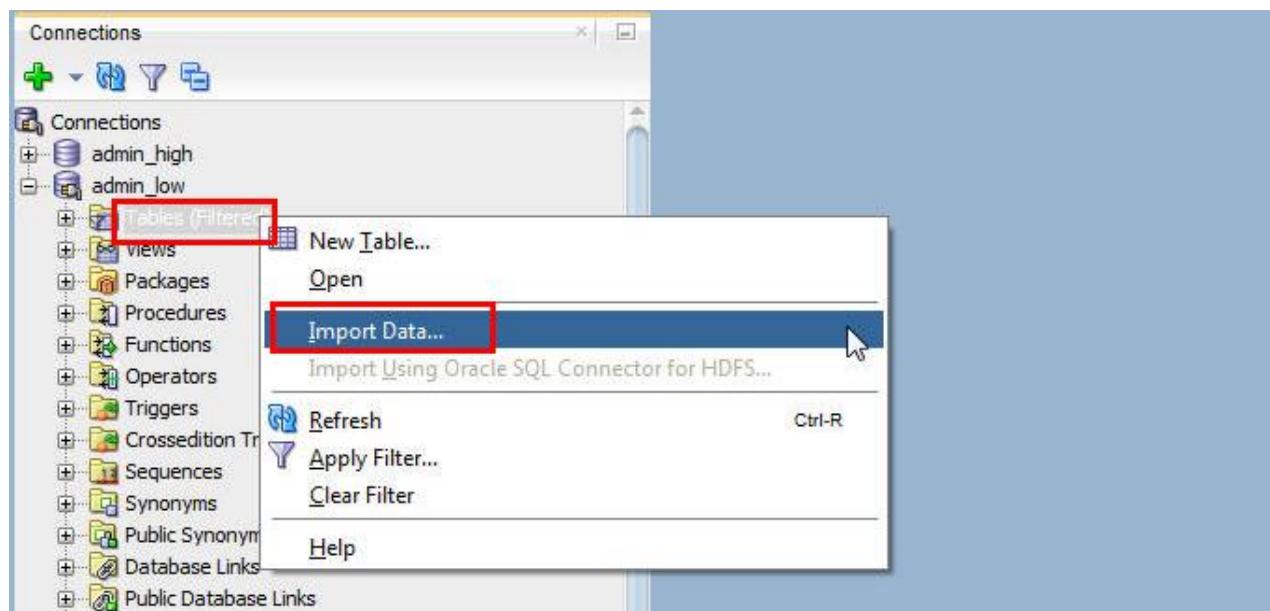
1. In SQL Developer, open the `admin_low` connection and create the credentials to the Object Storage.

```
set define off
begin
    DBMS_CLOUD.create_credential(
        credential_name => 'OBJ_STORE_CRED',
        username => 'api.user',
        password => '<Swift Password>'
    );
end;
/

```



2. To start the import wizard, right-click on **Tables** and navigate to the **Import Data** option.



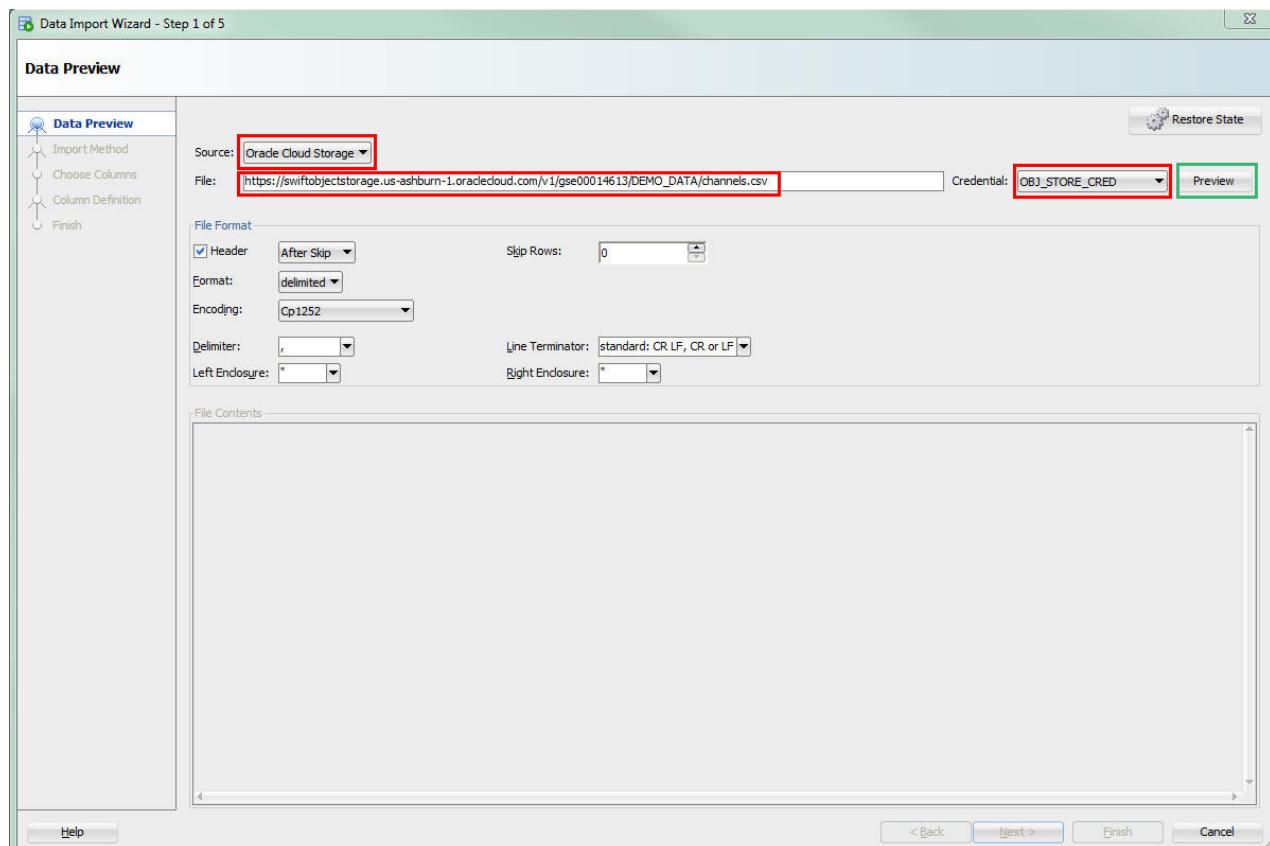
3. In the first step of the wizard, fill in the following details:

**Source: Oracle Cloud Storage**

**File:** [https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/<identity\\_domain>/DEMO\\_DATA/channels.csv](https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/<identity_domain>/DEMO_DATA/channels.csv)

**Credentials:** OBJ\_STORE\_CRED

Click Preview to see the data.



4. The data is parsed and displayed correctly. Click **Next**.

Data Import Wizard - Step 1 of 5

**Data Preview**

Source: Oracle Cloud Storage

File: [https://swiftobjectstorage.us-ashburn-1.oracledata.com/v1/gse00014613/DEMO\\_DATA/channels.csv](https://swiftobjectstorage.us-ashburn-1.oracledata.com/v1/gse00014613/DEMO_DATA/channels.csv)

Credential: OBJ\_STORE\_CRED

Preview

**File Format**

Header      Skip Rows: 0

Format: csv   Encoding: Cp1252

Delimiter: ,   Line Terminator: standard: CR LF, CR or LF

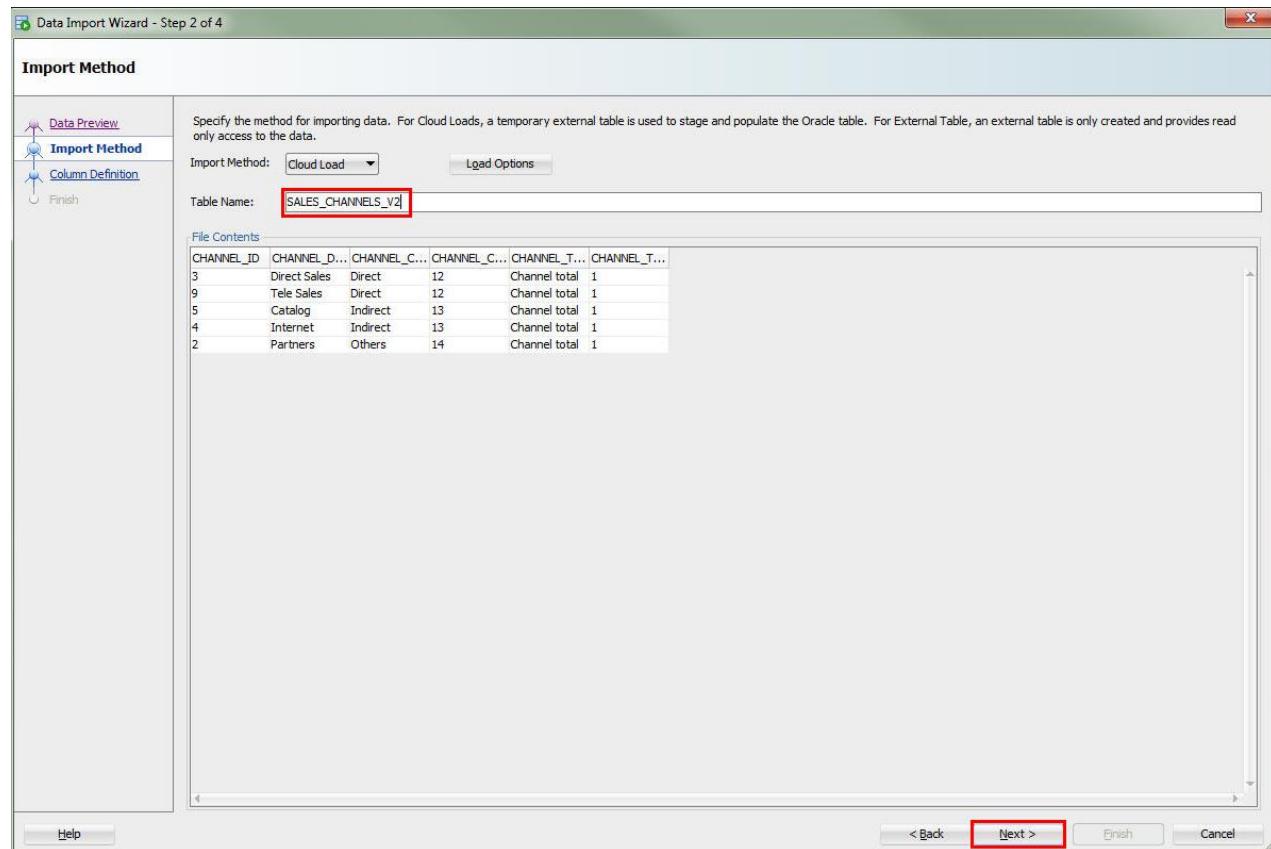
Left Enclosure: "   Right Enclosure: "

**File Contents**

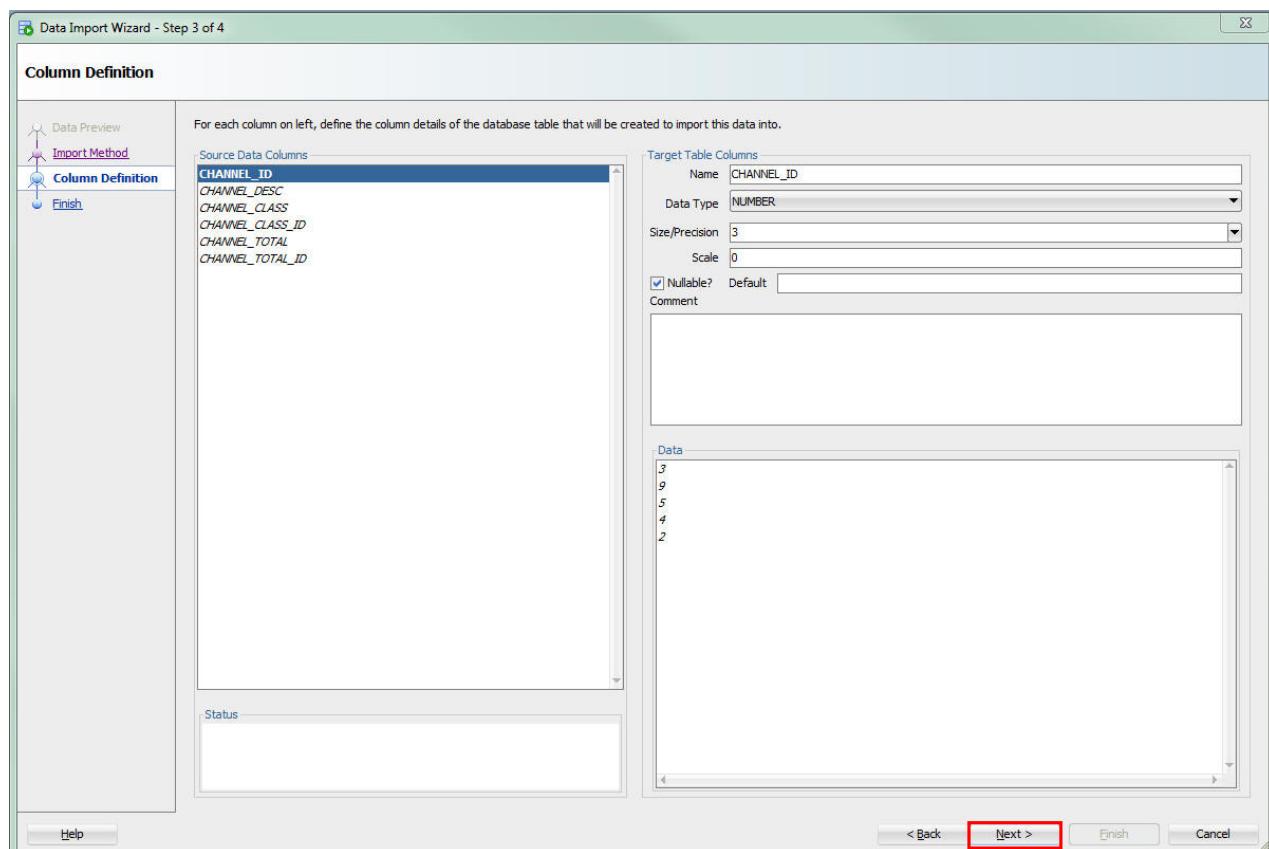
CHANNEL_ID	CHANNEL_DESC	CHANNEL_CARRIER	CHANNEL_CARRIER_ID	CHANNEL_TYPE	CHANNEL_TOTAL
3	Direct Sales	Direct	12	Channel total	1
9	Tele Sales	Direct	12	Channel total	1
5	Catalog	Indirect	13	Channel total	1
4	Internet	Indirect	13	Channel total	1
2	Partners	Others	14	Channel total	1

< Back   Next >   Finish   Cancel

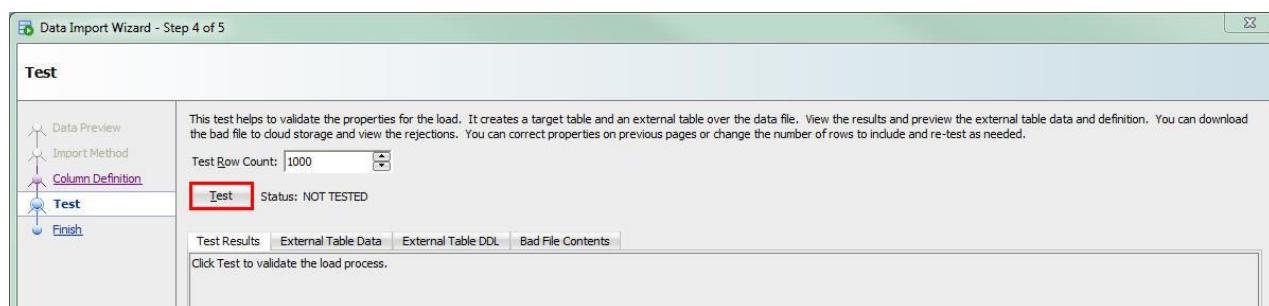
5. Fill in the Table Name as **SALES\_CHANNELS\_V2** and click **Next**.



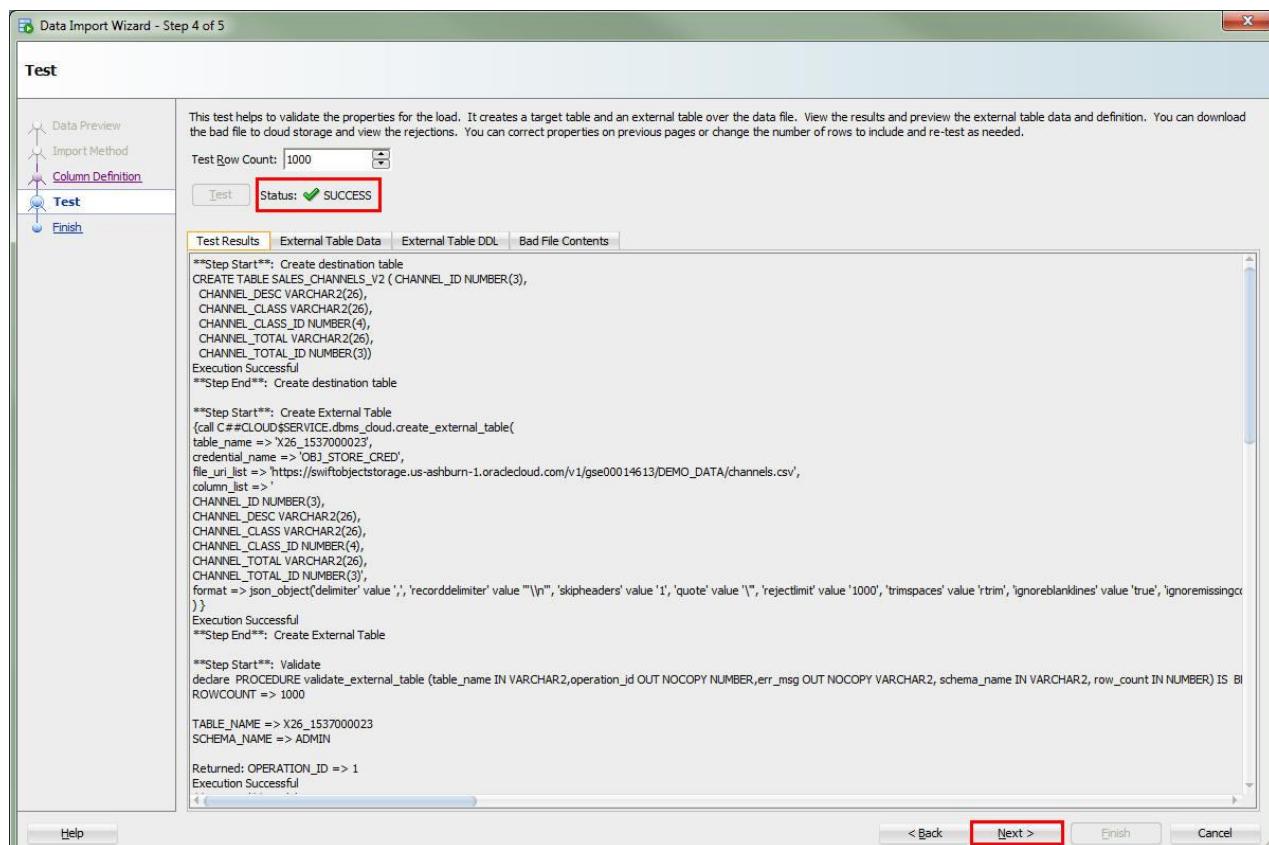
6. The column definition screen shows you the columns and the corresponding data types and some sample data for each column. You can change the data types here if you need to.  
Click **Next**.



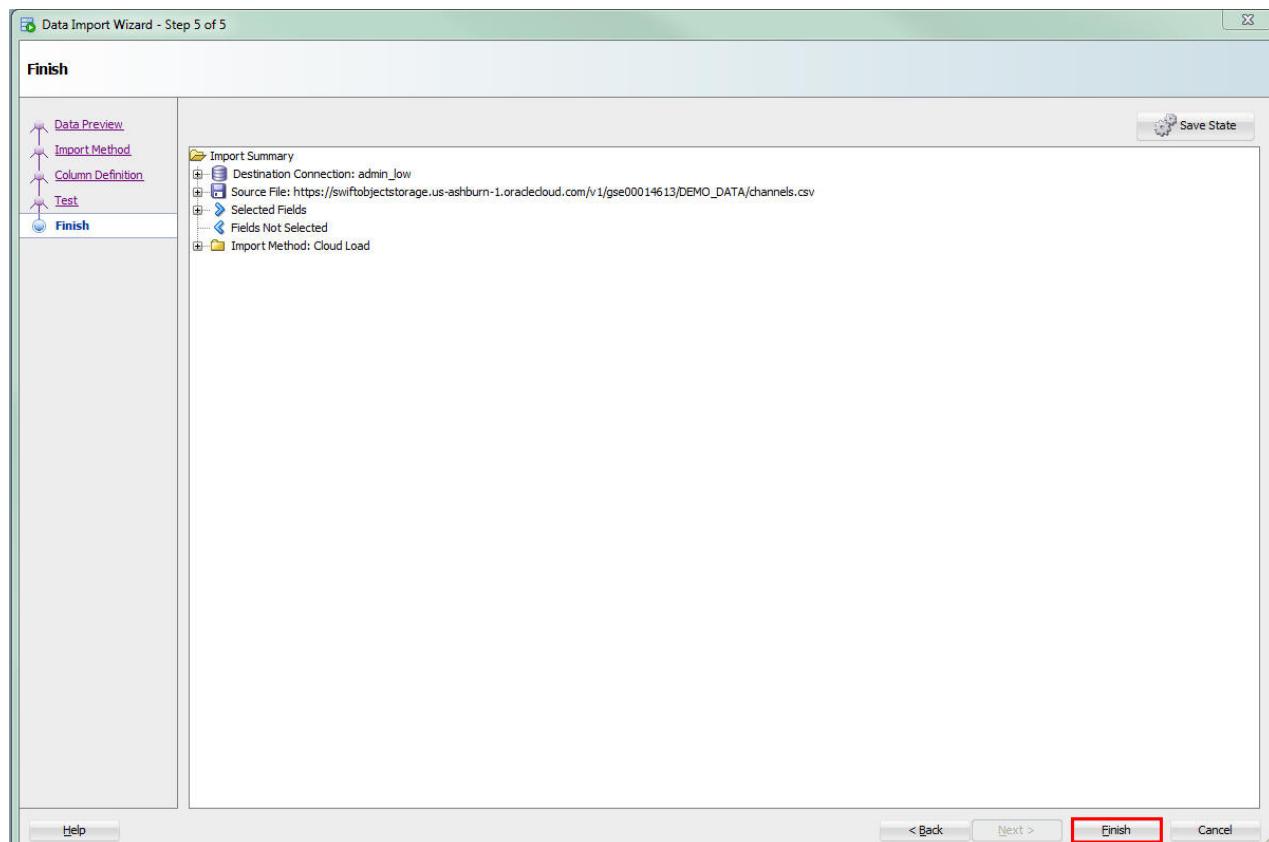
7. Click **Test** to validate the import operation.



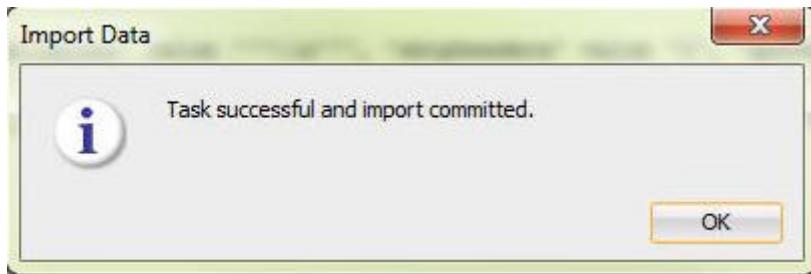
8. The test was successful and the steps performed by the Import Data wizard are displayed. Click **Next**.



9. On the *Import Summary* page, click **Finish**.

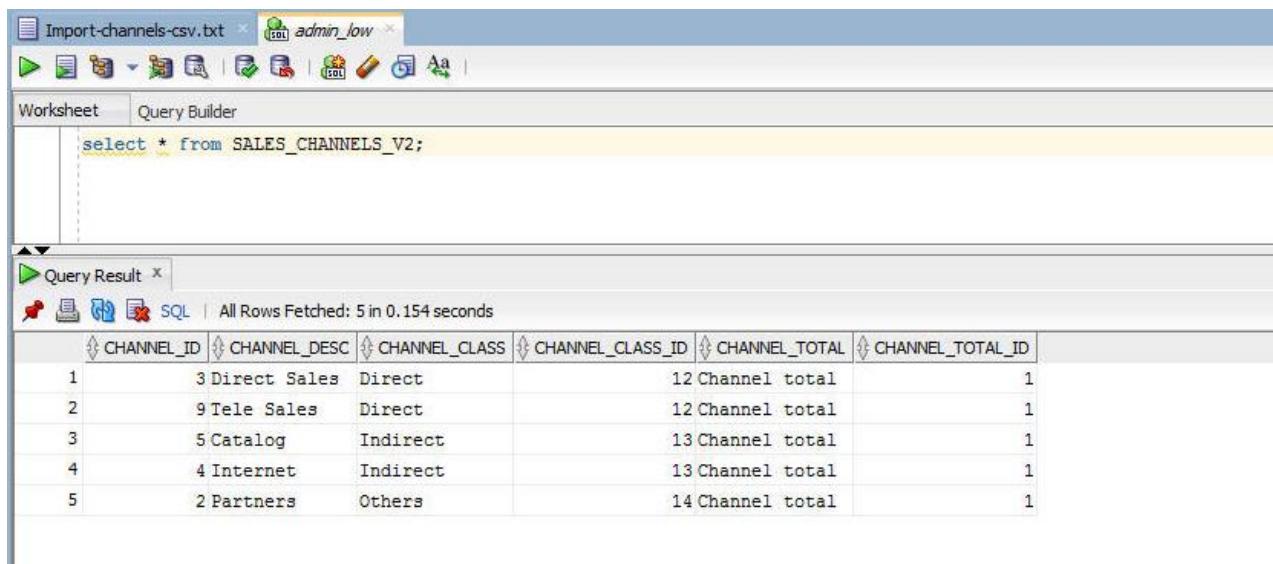


10. Click **OK** on the confirmation message.



11. To validate the import operation, query the SALES\_CHANNELS\_V2 table.

```
select * from SALES_CHANNELS_V2;
```



A screenshot of the Oracle SQL Worksheet interface. The top tab bar shows "Import-channels-csv.txt" and "admin\_low". The main area has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. A query is entered in the worksheet pane:

```
select * from SALES_CHANNELS_V2;
```

The "Query Result" pane below shows the output of the query:

CHANNEL_ID	CHANNEL_DESC	CHANNEL_CLASS	CHANNEL_CLASS_ID	CHANNEL_TOTAL	CHANNEL_TOTAL_ID
1	3 Direct Sales	Direct		12 Channel total	1
2	9 Tele Sales	Direct		12 Channel total	1
3	5 Catalog	Indirect		13 Channel total	1
4	4 Internet	Indirect		13 Channel total	1
5	2 Partners	Others		14 Channel total	1

Below the table, it says "All Rows Fetched: 5 in 0.154 seconds".

## Section 9. Data Loading from Cloud Object Storage using PL/SQL

In this section, you will perform a similar load operation as in the previous lab, but this time you'll use the DBMS\_CLOUD package.

1. Open an *admin\_low* connection and create the database tables by running the *Create\_Tables\_DDL\_Script.sql* file.

```

CREATE TABLE sales (
    prod_id      NUMBER      NOT NULL,
    cust_id      NUMBER      NOT NULL,
    time_id      DATE        NOT NULL,
    channel_id   NUMBER      NOT NULL,
    promo_id     NUMBER      NOT NULL,
    quantity_sold NUMBER(10,2) NOT NULL,
    amount_sold  NUMBER(10,2) NOT NULL);

CREATE TABLE costs (
    prod_id      NUMBER      NOT NULL,
    time_id      DATE        NOT NULL,
    channel_id   NUMBER      NOT NULL,
    unit_cost    NUMBER(10,2) NOT NULL,
    unit_price   NUMBER(10,2) NOT NULL);

CREATE TABLE times (
    time_id      DATE        NOT NULL,
    day_name     VARCHAR2(9)  NOT NULL,
    day_number_in_week NUMBER(1) NOT NULL,
    day_number_in_month NUMBER(2) NOT NULL,
    calendar_week_number NUMBER(2) NOT NULL);

```

Table COSTS altered.  
Table CHANNELS altered.  
Table SALES altered.

2. Once the data files are available on the object storage (as described in a previous lab), we can start running the loading script which will use the DBMS\_CLOUD package.

Open the *Load\_Data\_Script.sql* file and run it.

Make sure you use the connection details provided to you by the host.

```

begin
  dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true')
  );
end;
/

begin
  dbms_cloud.copy_data(
    table_name =>'COUNTRIES',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/coun_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true')
  );
end;
/

begin
  dbms_cloud.copy_data(
    table_name =>'CUSTOMERS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/cust_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true')
  );
end;
/

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

3. After the data has been loaded, we can run a sample query.

Open the Query\_Data\_Script.sql file and run the SELECT statement.

```

SELECT c.cust_id, t.calendar_quarter_desc, TO_CHAR(SUM(amount_sold),
  '9,999,999,999.99') AS Q_SALES, TO_CHAR(SUM(SUM(amount_sold))
OVER (PARTITION BY c.cust_id ORDER BY c.cust_id, t.calendar_quarter_desc
ROWS UNBOUNDED
PRECEDING), '9,999,999,999.99') AS CUM_SALES
FROM sales s, times t, customers c
WHERE s.time_id=t.time_id AND s.cust_id=c.cust_id AND t.calendar_year=2000
  AND c.cust_id IN (2595, 9646, 11111)
GROUP BY c.cust_id, t.calendar_quarter_desc
ORDER BY c.cust_id, t.calendar_quarter_desc;

```

CUST_ID	CALENDAR_QUARTER_DESC	Q_SALES	CUM_SALES
1	2595 2000-01	659.92	659.92
2	2595 2000-02	224.79	884.71
3	2595 2000-03	313.90	1,198.61
4	2595 2000-04	6,015.08	7,213.69
5	9646 2000-01	1,337.09	1,337.09
6	9646 2000-02	185.67	1,522.76
7	9646 2000-03	203.86	1,726.62
8	9646 2000-04	458.29	2,184.91
9	11111 2000-01	43.18	43.18
10	11111 2000-02	33.33	76.51
11	11111 2000-03	579.73	656.24
12	11111 2000-04	307.58	963.82

4. To see the status of the loading jobs, you can query `user_load_operations`. Here you will see all load operations along with status and elapsed time.

```
select * from user_load_operations;
```

ID	TYPE	SID	SER...	START_TIME	UPDATE_TIME	STATUS	TABLE_NAME	OWNER_NAME	FILE_URI_LIST
1	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	CHANNELS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3.dat</a>
2	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	COUNTRIES	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/coun_v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/coun_v3.dat</a>
3	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	CUSTOMERS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/custiv3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/custiv3.dat</a>
4	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	SUPPLEMENTARY_DEMOGRAPHICS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/demiv3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/demiv3.dat</a>
5	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	SALES	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/dmsal_v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/dmsal_v3.dat</a>
6	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	PRODUCTS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/prod1v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/prod1v3.dat</a>
7	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	PROMOTIONS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/prom1v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/prom1v3.dat</a>
8	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	SALES	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/sale1v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/sale1v3.dat</a>
9	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	TIMES	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/time_v3.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/time_v3.dat</a>
10	COPY	13781	41093 30-APR-18 ...	30-APR-18 ...		COMPLETED	COSTS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/costs.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/costs.dat</a>

5. The next step will showcase a failed job. Open `Load_Data_Error_Script.sql`.

The script will throw an error due to a mismatch in the mapping of the table to the data file.

```

begin
  dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO\_DATA/chan\_v3\_error.dat,
    format => json_object('ignorereadingcolumns' value 'true', 'removequotes' value 'true')
  );
end;
/

```

Error starting at line : 1 in command -

```

begin
  dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO\_DATA/chan\_v3\_error.dat,
    format => json_object('ignorereadingcolumns' value 'true', 'removequotes' value 'true')
  );
end;
Error report -
ORA-29913: error in executing ODCIEXTTABLEFETCH callout
ORA-06512: at "C##CLOUD$SERVICE.DBMS_CLOUD", line 801
ORA-06512: at "C##CLOUD$SERVICE.DBMS_CLOUD", line 782
ORA-30653: reject limit reached
ORA-06512: at "C##CLOUD$SERVICE.DBMS_CLOUD", line 757
ORA-06512: at "C##CLOUD$SERVICE.DBMS_CLOUD", line 819
ORA-06512: at line 2
29913. 00000 -  "error in executing #s callout"
*Cause:  The execution of the specified callout caused an error.
*Action: Examine the error messages take appropriate action.

```

6. Query again the `user_load_operations` view for the failed jobs.

```
select * from user_load_operations
where status='FAILED';
```



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a tab labeled "admin\_low~4". The main window has two tabs: "Worksheet" and "Query Builder". The "Worksheet" tab is active, displaying the following SQL query:

```
select * from user_load_operations  
where status='FAILED';
```

Below the query, the "Query Result" tab is selected, showing the output of the query. The results are presented in a table with the following columns:

ID	TYPE	SID	SERIAL#	START TIME	UPDATE TIME	STATUS	TABLE NAME	OWNER NAME	FILE URI LIST
1	COPY	13781	65476	30-APR-18 ...	30-APR-18 ...	FAILED	CHANNELS	ADMIN	<a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3_error.dat">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3_error.dat</a>

The table shows one row of data where the status is "FAILED". The "FILE URI LIST" column contains a single URL: [https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO\\_DATA/chan\\_v3\\_error.dat](https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3_error.dat).

## Section 10. Querying External Data

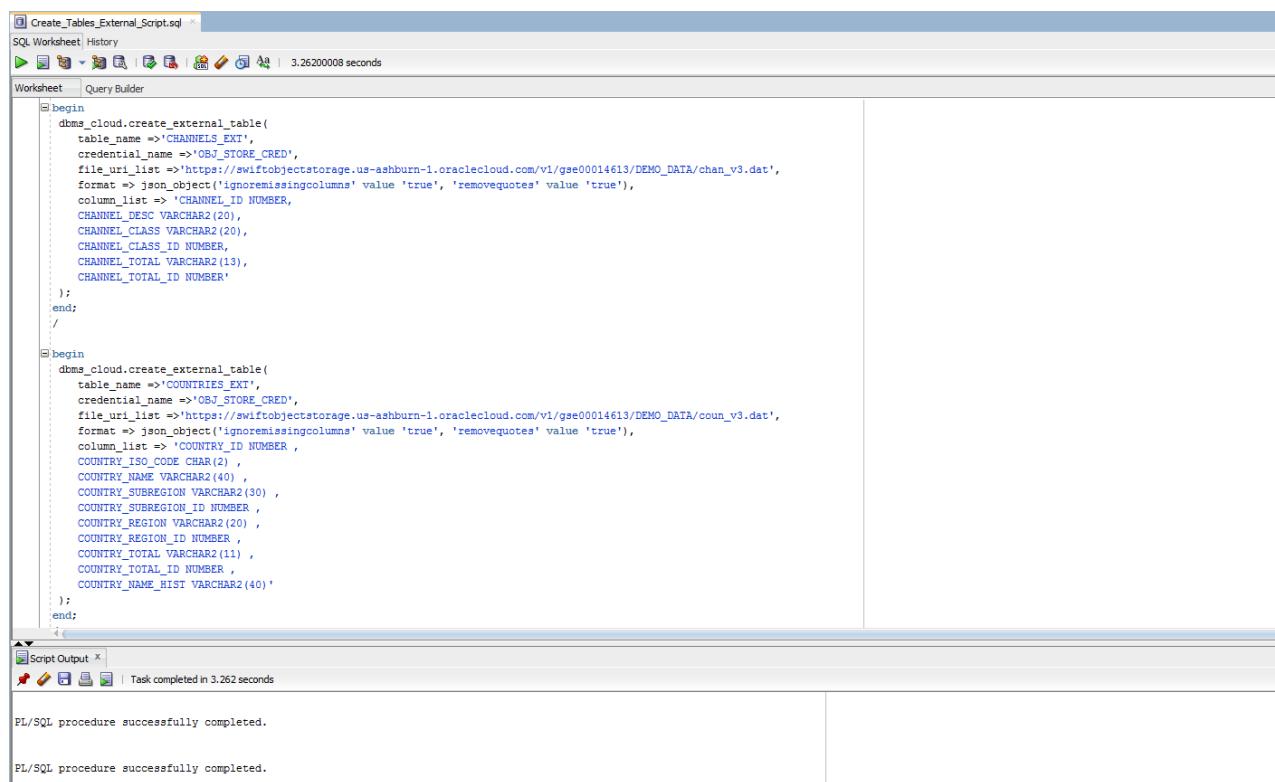
In this section you will be querying files on the OCI Object Storage directly without loading them to your database.

Connected as the Admin user, copy and paste the below script to SQL Developer and run it to create external tables on top of files residing on the OCI Object Storage. Note that we are still using the same credential we created when loading data in the previous section.

1. Use the DBMS\_CLOUD package to create external tables for each of the files in the Object Storage.

Open *Create\_Tables\_External\_Script.sql* and run the code.

Make sure you use the connection details provided to you by the host.



The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Create\_Tables\_External\_Script.sql
- Toolbar:** Standard SQL Developer toolbar.
- Worksheet Tab:** Active tab.
- Script Content:**

```

@begin
  dbms_cloud.create_external_table(
    table_name =>'CHANNELS_EXT',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true'),
    column_list => 'CHANNEL_ID NUMBER,
    CHANNEL_DESC VARCHAR2(20),
    CHANNEL_CLASS VARCHAR2(20),
    CHANNEL_CLASS_ID NUMBER,
    CHANNEL_TOTAL VARCHAR2(13),
    CHANNEL_TOTAL_ID NUMBER'
  );
end;
/

@begin
  dbms_cloud.create_external_table(
    table_name =>'COUNTRIES_EXT',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/gse00014613/DEMO_DATA/coun_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value 'true'),
    column_list => 'COUNTRY_ID NUMBER,
    COUNTRY_ISO_CODE CHAR(2) ,
    COUNTRY_NAME VARCHAR2(40) ,
    COUNTRY_SUBREGION VARCHAR2(30) ,
    COUNTRY_SUBREGION_ID NUMBER ,
    COUNTRY_REGION VARCHAR2(20) ,
    COUNTRY_REGION_ID NUMBER ,
    COUNTRY_TOTAL VARCHAR2(11) ,
    COUNTRY_TOTAL_ID NUMBER ,
    COUNTRY_NAME_HIST VARCHAR2(40) '
  );
end;
/

```
- Script Output:** Shows the message "PL/SQL procedure successfully completed." twice.

2. Run the *Query\_Data\_External\_Script.sql* on the newly created external tables.

The screenshot shows the Oracle SQL Worksheet interface. The top window is titled "Query\_Data\_External\_Script.sql". The code in the worksheet pane is:

```
SELECT c.cust_id, t.calendar_quarter_desc, TO_CHAR (SUM(amount_sold),
  '9,999,999,999.99') AS Q_SALES, TO_CHAR(SUM(SUM(amount_sold))
OVER (PARTITION BY c.cust_id ORDER BY c.cust_id, t.calendar_quarter_desc
ROWS UNBOUNDED
PRECEDING), '9,999,999,999.99') AS CUM_SALES
FROM sales_ext s, times_ext t, customers_ext c
WHERE s.time_id=t.time_id AND s.cust_id=c.cust_id AND t.calendar_year=2000
  AND c.cust_id IN (2595, 9646, 11111)
GROUP BY c.cust_id, t.calendar_quarter_desc
ORDER BY c.cust_id, t.calendar_quarter_desc;
```

The bottom window is titled "Query Result" and displays the results of the query:

CUST_ID	CALENDAR_QUARTER_DESC	Q_SALES	CUM_SALES
1	2595 2000-01	659.92	659.92
2	2595 2000-02	224.79	884.71
3	2595 2000-03	313.90	1,198.61
4	2595 2000-04	6,015.08	7,213.69
5	9646 2000-01	1,337.09	1,337.09
6	9646 2000-02	185.67	1,522.76
7	9646 2000-03	203.86	1,726.62
8	9646 2000-04	458.29	2,184.91
9	11111 2000-01	43.18	43.18
10	11111 2000-02	33.33	76.51
11	11111 2000-03	579.73	656.24
12	11111 2000-04	307.58	963.82

## Section 11. Scaling an ADWC Service

In this section we will scale up the ADWC service by adding additional CPUs and Storage.

1. Identify your database in the instances tab and click **Scale Service** in the actions menu.

The screenshot shows the Oracle Autonomous Data Warehouse Cloud Instances page. At the top, there's a summary bar with counts for Databases (1), CPUs (2), and TB (1). Below this is a table for the instance 'ADW1', which was created on April 20, 2018, at 7:43:39 AM UTC. The table includes columns for Service Console, Start, Stop, Scale Service (which is highlighted with a red box), and Delete. A search bar and a 'Create Instance' button are also visible.

2. In the popup window choose the scale operation and fill in the additional number of CPUs / TB of Storage and click the **Scale Service** button.

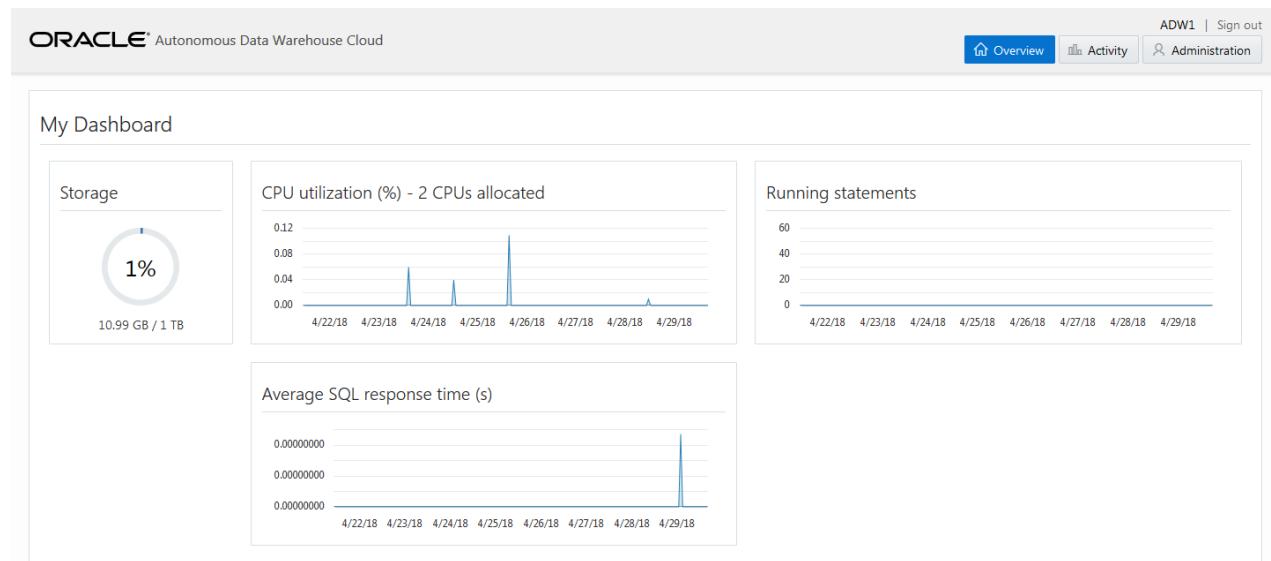
The screenshot shows the 'Scale Service' dialog box. It has fields for 'Type of change, increase or decrease' (set to 'Scale Up'), 'CPU Core Count Change' (set to '4'), and 'Storage Capacity [TB] Change' (set to '0'). At the bottom are 'Scale Service' and 'Cancel' buttons, with 'Scale Service' also highlighted with a red box.

3. The Scale operation will restart your service. Once the operation is done you can check the new values.

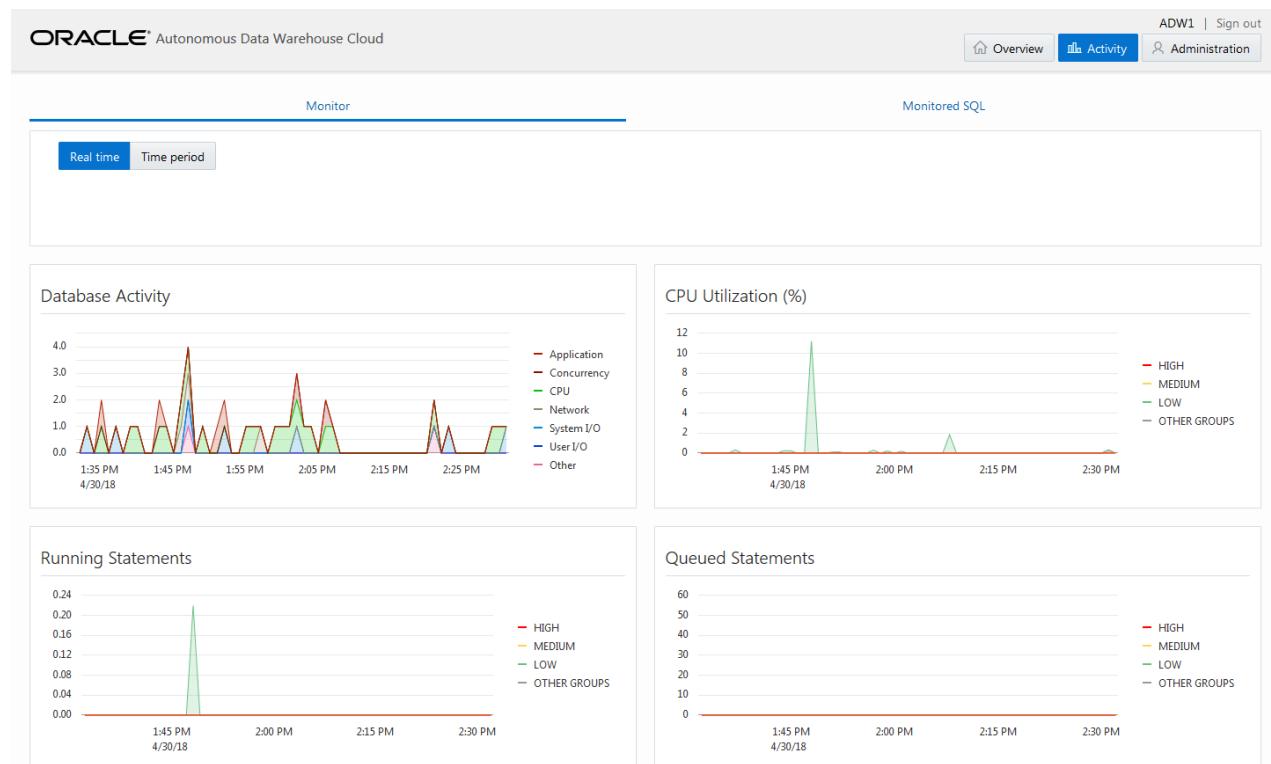
## Section 12. Monitoring ADWC

In this section you will use the ADWC service console to monitor the database by going through the Overview and Activity tabs in the service console to monitor the performance of your database.

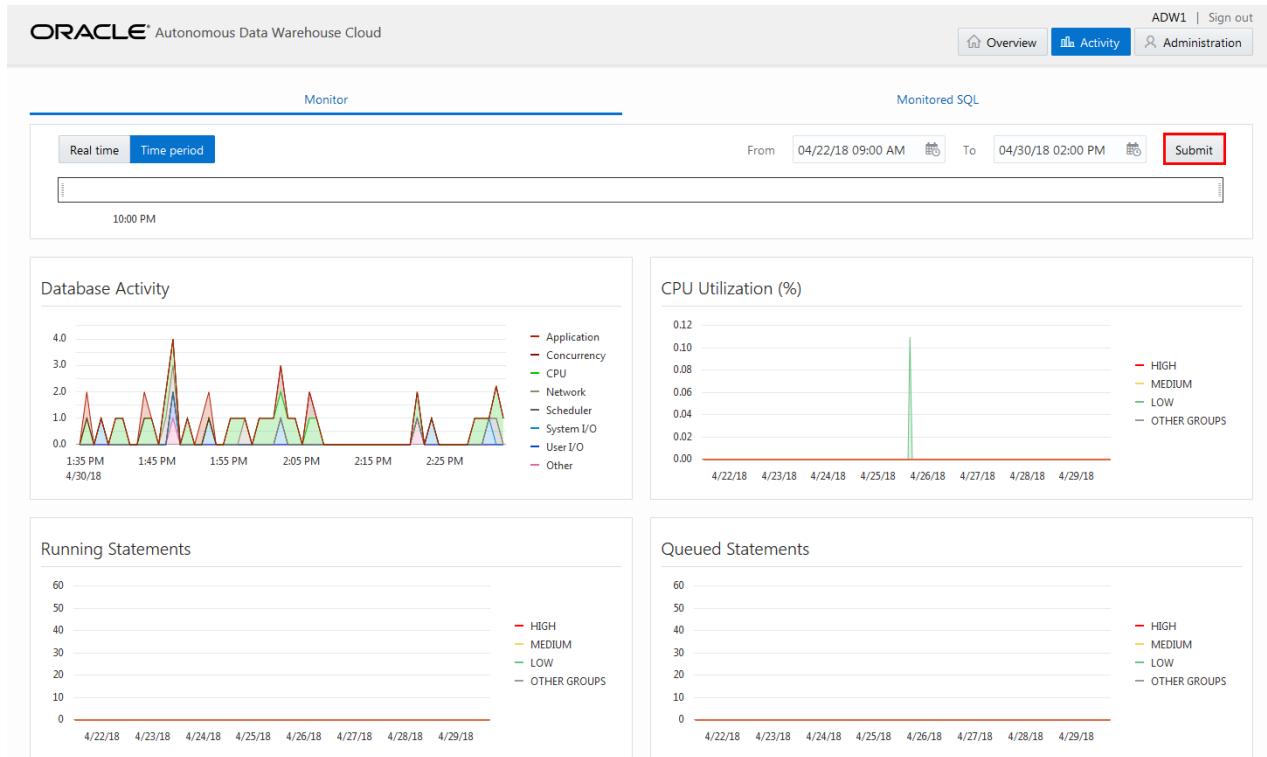
1. Log in to the Service Console and on the **Overview** page you will see a summary of your instance' activity and metrics.



2. From the upper right corner switch to the Activity tab and by default you will see the **Real Time** graphics on your database' activity and resource consumption.



3. For a historical view, click on the **Time period** tab, select the time period you wish to analyse and hit **Submit**. Similar graphics will be displayed.



4. Navigate to the **Monitor SQL** tab where you will see all the queries and loading jobs that were made against the database.



---

## Apendix – SQL Scripts

### 1. Create\_Tables\_DDL\_Script.sql

```
CREATE TABLE sales (
    prod_id          NUMBER      NOT NULL,
    cust_id          NUMBER      NOT NULL,
    time_id          DATE        NOT NULL,
    channel_id       NUMBER      NOT NULL,
    promo_id         NUMBER      NOT NULL,
    quantity_sold    NUMBER(10,2) NOT NULL,
    amount_sold      NUMBER(10,2) NOT NULL);

CREATE TABLE costs (
    prod_id          NUMBER      NOT NULL,
    time_id          DATE        NOT NULL,
    promo_id         NUMBER      NOT NULL,
    channel_id       NUMBER      NOT NULL,
    unit_cost        NUMBER(10,2) NOT NULL,
    unit_price       NUMBER(10,2) NOT NULL);

CREATE TABLE times (
    time_id          DATE        NOT NULL,
    day_name         VARCHAR2(9) NOT NULL,
    day_number_in_week NUMBER(1)  NOT NULL,
    day_number_in_month NUMBER(2)  NOT NULL,
    calendar_week_number NUMBER(2)  NOT NULL,
    fiscal_week_number NUMBER(2)  NOT NULL,
    week_ending_day  DATE        NOT NULL,
    week_ending_day_id NUMBER     NOT NULL,
    calendar_month_number NUMBER(2)  NOT NULL,
    fiscal_month_number NUMBER(2)  NOT NULL,
    calendar_month_desc VARCHAR2(8) NOT NULL,
    calendar_month_id NUMBER     NOT NULL,
    fiscal_month_desc VARCHAR2(8) NOT NULL,
```

```

fiscal_month_id          NUMBER      NOT NULL,
days_in_cal_month       NUMBER      NOT NULL,
days_in_fis_month        NUMBER      NOT NULL,
end_of_cal_month         DATE        NOT NULL,
end_of_fis_month         DATE        NOT NULL,
calendar_month_name     VARCHAR2(9) NOT NULL,
fiscal_month_name        VARCHAR2(9) NOT NULL,
calendar_quarter_desc   CHAR(7)    NOT NULL,
calendar_quarter_id     NUMBER      NOT NULL,
fiscal_quarter_desc     CHAR(7)    NOT NULL,
fiscal_quarter_id        NUMBER      NOT NULL,
days_in_cal_quarter     NUMBER      NOT NULL,
days_in_fis_quarter     NUMBER      NOT NULL,
end_of_cal_quarter      DATE        NOT NULL,
end_of_fis_quarter      DATE        NOT NULL,
calendar_quarter_number NUMBER(1)  NOT NULL,
fiscal_quarter_number   NUMBER(1)  NOT NULL,
calendar_year            NUMBER(4)   NOT NULL,
calendar_year_id          NUMBER      NOT NULL,
fiscal_year              NUMBER(4)   NOT NULL,
fiscal_year_id            NUMBER      NOT NULL,
days_in_cal_year          NUMBER      NOT NULL,
days_in_fis_year          NUMBER      NOT NULL,
end_of_cal_year           DATE        NOT NULL,
end_of_fis_year           DATE        NOT NULL );

```

```

CREATE TABLE products (
prod_id          NUMBER(6)      NOT NULL,
prod_name        VARCHAR2(50)   NOT NULL,
prod_desc         VARCHAR2(4000)  NOT NULL,
prod_subcategory VARCHAR2(50)   NOT NULL,
prod_subcategory_id NUMBER        NOT NULL,
prod_subcategory_desc VARCHAR2(2000) NOT NULL,
prod_category    VARCHAR2(50)   NOT NULL,
prod_category_id NUMBER        NOT NULL,
prod_category_desc VARCHAR2(2000) NOT NULL,
prod_weight_class NUMBER(3)    NOT NULL,

```

```

prod_unit_of_measure      VARCHAR2(20)   ,
prod_pack_size            VARCHAR2(30)    NOT NULL,
supplier_id                NUMBER(6)      NOT NULL,
prod_status                VARCHAR2(20)   NOT NULL,
prod_list_price            NUMBER(8,2)    NOT NULL,
prod_min_price              NUMBER(8,2)    NOT NULL,
prod_total                  VARCHAR2(13)   NOT NULL,
prod_total_id                NUMBER        NOT NULL,
prod_src_id                  NUMBER        ,
prod_eff_from                DATE          ,
prod_eff_to                  DATE          ,
prod_valid                  VARCHAR2(1)  );

```

```

CREATE TABLE channels (
  channel_id                NUMBER        NOT NULL,
  channel_desc                VARCHAR2(20)   NOT NULL,
  channel_class                VARCHAR2(20)   NOT NULL,
  channel_class_id              NUMBER        NOT NULL,
  channel_total                  VARCHAR2(13)   NOT NULL,
  channel_total_id                NUMBER        NOT NULL);

```

```

CREATE TABLE promotions (
  promo_id                  NUMBER(6)     NOT NULL,
  promo_name                  VARCHAR2(30)   NOT NULL,
  promo_subcategory            VARCHAR2(30)   NOT NULL,
  promo_subcategory_id          NUMBER        NOT NULL,
  promo_category                VARCHAR2(30)   NOT NULL,
  promo_category_id              NUMBER        NOT NULL,
  promo_cost                  NUMBER(10,2)   NOT NULL,
  promo_begin_date              DATE          NOT NULL,
  promo_end_date                DATE          NOT NULL,
  promo_total                  VARCHAR2(15)   NOT NULL,
  promo_total_id                NUMBER        NOT NULL);

```

```

CREATE TABLE customers (
  cust_id                  NUMBER        NOT NULL,
  cust_first_name            VARCHAR2(20)   NOT NULL,

```

```

    cust_last_name          VARCHAR2(40)      NOT NULL,
    cust_gender              CHAR(1)           NOT NULL,
    cust_year_of_birth       NUMBER(4)         NOT NULL,
    cust_marital_status     VARCHAR2(20)        ,
    cust_street_address     VARCHAR2(40)      NOT NULL,
    cust_postal_code         VARCHAR2(10)        NOT NULL,
    cust_city                VARCHAR2(30)        NOT NULL,
    cust_city_id              NUMBER           NOT NULL,
    cust_state_province      VARCHAR2(40)      NOT NULL,
    cust_state_province_id    NUMBER           NOT NULL,
    country_id                NUMBER           NOT NULL,
    cust_main_phone_number   VARCHAR2(25)      NOT NULL,
    cust_income_level         VARCHAR2(30)        ,
    cust_credit_limit         NUMBER           ,
    cust_email                VARCHAR2(50)        ,
    cust_total                VARCHAR2(14)      NOT NULL,
    cust_total_id              NUMBER           NOT NULL,
    cust_src_id                NUMBER           ,
    cust_eff_from              DATE             ,
    cust_eff_to                DATE             ,
    cust_valid                VARCHAR2(1)       );

```

```

CREATE TABLE countries (
    country_id                NUMBER           NOT NULL,
    country_iso_code            CHAR(2)           NOT NULL,
    country_name               VARCHAR2(40)      NOT NULL,
    country_subregion           VARCHAR2(30)      NOT NULL,
    country_subregion_id         NUMBER           NOT NULL,
    country_region              VARCHAR2(20)      NOT NULL,
    country_region_id            NUMBER           NOT NULL,
    country_total                VARCHAR2(11)      NOT NULL,
    country_total_id              NUMBER           NOT NULL,
    country_name_hist            VARCHAR2(40)) ;

```

```
CREATE TABLE supplementary_demographics
```

```
( CUST_ID          NUMBER not null,
```

```
EDUCATION      VARCHAR2(21),
OCCUPATION     VARCHAR2(21),
HOUSEHOLD_SIZE VARCHAR2(21),
YRS_RESIDENCE  NUMBER,
AFFINITY_CARD   NUMBER(10),
bulk_pack_diskettes NUMBER(10),
flat_panel_monitor  NUMBER(10),
home_theater_package NUMBER(10),
bookkeeping_application NUMBER(10),
printer_supplies NUMBER(10),
y_box_games NUMBER(10),
os_doc_set_kanji NUMBER(10),
COMMENTS        VARCHAR2(4000));

ALTER TABLE promotions
ADD CONSTRAINT promo_pk
PRIMARY KEY (promo_id)
RELY DISABLE NOVALIDATE;

ALTER TABLE sales
ADD CONSTRAINT sales_promo_fk
FOREIGN KEY (promo_id) REFERENCES promotions (promo_id)
RELY DISABLE NOVALIDATE;

ALTER TABLE costs
ADD CONSTRAINT costs_promo_fk
FOREIGN KEY (promo_id) REFERENCES promotions (promo_id)
RELY DISABLE NOVALIDATE;

ALTER TABLE customers
ADD CONSTRAINT customers_pk
PRIMARY KEY (cust_id)
RELY DISABLE NOVALIDATE;

ALTER TABLE sales
ADD CONSTRAINT sales_customer_fk
FOREIGN KEY (cust_id) REFERENCES customers (cust_id)
```

```
RELY DISABLE NOVALIDATE;

ALTER TABLE products
  ADD CONSTRAINT products_pk
    PRIMARY KEY (prod_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE sales
  ADD CONSTRAINT sales_product_fk
    FOREIGN KEY (prod_id) REFERENCES products (prod_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE costs
  ADD CONSTRAINT costs_product_fk
    FOREIGN KEY (prod_id) REFERENCES products (prod_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE times
  ADD CONSTRAINT times_pk
    PRIMARY KEY (time_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE sales
  ADD CONSTRAINT sales_time_fk
    FOREIGN KEY (time_id) REFERENCES times (time_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE costs
  ADD CONSTRAINT costs_time_fk
    FOREIGN KEY (time_id) REFERENCES times (time_id)
  RELY DISABLE NOVALIDATE;

ALTER TABLE channels
  ADD CONSTRAINT channels_pk
    PRIMARY KEY (channel_id)
  RELY DISABLE NOVALIDATE;
```

```

ALTER TABLE sales
  ADD CONSTRAINT sales_channel_fk
    FOREIGN KEY (channel_id) REFERENCES channels (channel_id)
    RELY DISABLE NOVALIDATE;

ALTER TABLE costs
  ADD CONSTRAINT costs_channel_fk
    FOREIGN KEY (channel_id) REFERENCES channels (channel_id)
    RELY DISABLE NOVALIDATE;

ALTER TABLE countries
  ADD CONSTRAINT countries_pk
    PRIMARY KEY (country_id)
    RELY DISABLE NOVALIDATE;

ALTER TABLE customers
  ADD CONSTRAINT customers_country_fk
    FOREIGN KEY (country_id) REFERENCES countries (country_id)
    RELY DISABLE NOVALIDATE;

ALTER TABLE supplementary_demographics
  ADD CONSTRAINT supp_demo_pk
    PRIMARY KEY (cust_id)
    RELY DISABLE NOVALIDATE;

```

## **2. Load\_Data\_Script.sql**

```

set define off;

begin
  DBMS_CLOUD.create_credential(
    credential_name => 'OBJ_STORE_CRED',
    username => '<STORAGE_USERNAME>',
    password => '<SWIFT_PASSWORD>'
  );
end;
/

```

```
begin
    dbms_cloud.copy_data(
        table_name =>'CHANNELS',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/chan_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true')
    );
end;
/

begin
    dbms_cloud.copy_data(
        table_name =>'COUNTRIES',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>//DEMO
_DATA/coun_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true')
    );
end;
/

begin
    dbms_cloud.copy_data(
        table_name =>'CUSTOMERS',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/cust1v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS')
    );
end;
/

begin
```

```
dbms_cloud.copy_data(
    table_name =>'SUPPLEMENTARY_DEMOGRAPHICS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/dem1v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true')
);
end;
/

begin
dbms_cloud.copy_data(
    table_name =>'SALES',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/dmsal_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD')
);
end;
/

begin
dbms_cloud.copy_data(
    table_name =>'PRODUCTS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/prod1v3.dat',
    format => json_object('delimiter' value '|', 'quote' value '^',
'ignoremissingcolumns' value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS',
'blankasnull' value 'true')
);
end;
/

begin
dbms_cloud.copy_data(
    table_name =>'PROMOTIONS',
```

```
credential_name =>'OBJ_STORE_CRED',
file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/prom1v3.dat',
format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS', 'blankasnull' value
'true')
);
end;
/

begin
dbms_cloud.copy_data(
table_name =>'SALES',
credential_name =>'OBJ_STORE_CRED',
file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/sale1v3.dat',
format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD', 'blankasnull' value 'true')
);
end;
/

begin
dbms_cloud.copy_data(
table_name =>'TIMES',
credential_name =>'OBJ_STORE_CRED',
file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/time_v3.dat',
format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS', 'blankasnull' value
'true')
);
end;
/

begin
dbms_cloud.copy_data(
table_name =>'COSTS',
```

```

        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/costs.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'dateformat' value
'YYYY-MM-DD', 'blankasnull' value 'true')
    );
end;
/

```

### **3. Query\_Data\_Script.sql**

```

SELECT c.cust_id, t.calendar_quarter_desc, TO_CHAR (SUM(amount_sold),
      '9,999,999,999.99') AS Q_SALES, TO_CHAR(SUM(SUM(amount_sold))
OVER (PARTITION BY c.cust_id ORDER BY c.cust_id, t.calendar_quarter_desc
ROWS UNBOUNDED
PRECEDING), '9,999,999,999.99') AS CUM_SALES
FROM sales s, times t, customers c
WHERE s.time_id=t.time_id AND s.cust_id=c.cust_id AND t.calendar_year=2000
      AND c.cust_id IN (2595, 9646, 11111)
GROUP BY c.cust_id, t.calendar_quarter_desc
ORDER BY c.cust_id, t.calendar_quarter_desc;

```

### **4. Load\_Data\_Error\_Script.sql**

```

begin
  dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/chan_v3_error.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true')
  );
end;
/

```

## 5. Create\_Tables\_External\_Script.sql

```

begin
    dbms_cloud.create_external_table(
        table_name =>'CHANNELS_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/chan_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true'),
        column_list => 'CHANNEL_ID NUMBER,
                        CHANNEL_DESC VARCHAR2(20),
                        CHANNEL_CLASS VARCHAR2(20),
                        CHANNEL_CLASS_ID NUMBER,
                        CHANNEL_TOTAL VARCHAR2(13),
                        CHANNEL_TOTAL_ID NUMBER'
    );
end;
/
begin
    dbms_cloud.create_external_table(
        table_name =>'COUNTRIES_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/coun_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true'),
        column_list => 'COUNTRY_ID NUMBER ,
                        COUNTRY_ISO_CODE CHAR(2) ,
                        COUNTRY_NAME VARCHAR2(40) ,
                        COUNTRY_SUBREGION VARCHAR2(30) ,
                        COUNTRY_SUBREGION_ID NUMBER ,
                        COUNTRY_REGION VARCHAR2(20) ,
                        COUNTRY_REGION_ID NUMBER ,
                        COUNTRY_TOTAL VARCHAR2(11) ,
                        COUNTRY_TOTAL_ID NUMBER ,
                        COUNTRY_NAME_HIST VARCHAR2(40) '
    );
end;
/

```

```
);

end;
/

begin

dbms_cloud.create_external_table(
    table_name =>'CUSTOMERS_EXT',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/cust1v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS'),
    column_list => 'CUST_ID NUMBER ,
    CUST_FIRST_NAME VARCHAR2(20) ,
    CUST_LAST_NAME VARCHAR2(40) ,
    CUST_GENDER CHAR(1) ,
    CUST_YEAR_OF_BIRTH NUMBER(4,0) ,
    CUST_MARITAL_STATUS VARCHAR2(20),
    CUST_STREET_ADDRESS VARCHAR2(40) ,
    CUST_POSTAL_CODE VARCHAR2(10) ,
    CUST_CITY VARCHAR2(30) ,
    CUST_CITY_ID NUMBER ,
    CUST_STATE_PROVINCE VARCHAR2(40) ,
    CUST_STATE_PROVINCE_ID NUMBER ,
    COUNTRY_ID NUMBER ,
    CUST_MAIN_PHONE_NUMBER VARCHAR2(25) ,
    CUST_INCOME_LEVEL VARCHAR2(30),
    CUST_CREDIT_LIMIT NUMBER,
    CUST_EMAIL VARCHAR2(50),
    CUST_TOTAL VARCHAR2(14) ,
    CUST_TOTAL_ID NUMBER ,
    CUST_SRC_ID NUMBER,
    CUST_EFF_FROM DATE,
    CUST_EFF_TO DATE,
    CUST_VALID VARCHAR2(1) '

);
end;
/
```

```

begin

    dbms_cloud.create_external_table(
        table_name =>'SUPPLEMENTARY_DEMOGRAPHICS_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/dem1v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true'),
        column_list => 'CUST_ID NUMBER ,
EDUCATION VARCHAR2(21),
OCCUPATION VARCHAR2(21),
HOUSEHOLD_SIZE VARCHAR2(21),
YRS_RESIDENCE NUMBER,
AFFINITY_CARD NUMBER(10,0),
BULK_PACK_DISKETTES NUMBER(10,0),
FLAT_PANEL_MONITOR NUMBER(10,0),
HOME_THEATER_PACKAGE NUMBER(10,0),
BOOKKEEPING_APPLICATION NUMBER(10,0),
PRINTER_SUPPLIES NUMBER(10,0),
Y_BOX_GAMES NUMBER(10,0),
OS_DOC_SET_KANJI NUMBER(10,0),
COMMENTS VARCHAR2(4000)'

);

end;
/
begin

    dbms_cloud.create_external_table(
        table_name =>'PRODUCTS_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/prod1v3.dat',
        format => json_object('delimiter' value '|', 'quote' value '^',
'ignoremissingcolumns' value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS',
'blankasnull' value 'true'),
        column_list => 'PROD_ID NUMBER(6,0) ,
PROD_NAME VARCHAR2(50) ,
```

```

        PROD_DESC VARCHAR2(4000) ,
        PROD_SUBCATEGORY VARCHAR2(50) ,
        PROD_SUBCATEGORY_ID NUMBER ,
        PROD_SUBCATEGORY_DESC VARCHAR2(2000) ,
        PROD_CATEGORY VARCHAR2(50) ,
        PROD_CATEGORY_ID NUMBER ,
        PROD_CATEGORY_DESC VARCHAR2(2000) ,
        PROD_WEIGHT_CLASS NUMBER(3,0) ,
        PROD_UNIT_OF_MEASURE VARCHAR2(20),
        PROD_PACK_SIZE VARCHAR2(30) ,
        SUPPLIER_ID NUMBER(6,0) ,
        PROD_STATUS VARCHAR2(20) ,
        PROD_LIST_PRICE NUMBER(8,2) ,
        PROD_MIN_PRICE NUMBER(8,2) ,
        PROD_TOTAL VARCHAR2(13) ,
        PROD_TOTAL_ID NUMBER ,
        PROD_SRC_ID NUMBER,
        PROD_EFF_FROM DATE,
        PROD_EFF_TO DATE,
        PROD_VALID VARCHAR2(1) '
    );
end;
/
begin
    dbms_cloud.create_external_table(
        table_name =>'PROMOTIONS_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/prom1v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS', 'blankasnull' value
'true'),
        column_list => 'PROMO_ID NUMBER(6,0) ,
        PROMO_NAME VARCHAR2(30) ,
        PROMO_SUBCATEGORY VARCHAR2(30) ,
        PROMO_SUBCATEGORY_ID NUMBER ,
        PROMO_CATEGORY VARCHAR2(30) ,
```

```

        PROMO_CATEGORY_ID NUMBER ,
        PROMO_COST NUMBER(10,2) ,
        PROMO_BEGIN_DATE DATE ,
        PROMO_END_DATE DATE ,
        PROMO_TOTAL VARCHAR2(15) ,
        PROMO_TOTAL_ID NUMBER '
    );
end;
/

begin
    dbms_cloud.create_external_table(
        table_name =>'SALES_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/sale1v3.dat,https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_
_domain>/DEMO_DATA/dmsal_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD', 'blankasnull' value 'true'),
        column_list => 'PROD_ID NUMBER ,
        CUST_ID NUMBER ,
        TIME_ID DATE ,
        CHANNEL_ID NUMBER ,
        PROMO_ID NUMBER ,
        QUANTITY SOLD NUMBER(10,2) ,
        AMOUNT SOLD NUMBER(10,2)'
    );
end;
/

```

```

begin
    dbms_cloud.create_external_table(
        table_name =>'TIMES_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/time_v3.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'removequotes'
value 'true', 'dateformat' value 'YYYY-MM-DD-HH24-MI-SS', 'blankasnull' value
'true'),

```

```
column_list => 'TIME_ID DATE ,
DAY_NAME VARCHAR2(9) ,
DAY_NUMBER_IN_WEEK NUMBER(1,0) ,
DAY_NUMBER_IN_MONTH NUMBER(2,0) ,
CALENDAR_WEEK_NUMBER NUMBER(2,0) ,
FISCAL_WEEK_NUMBER NUMBER(2,0) ,
WEEK_ENDING_DAY DATE ,
WEEK_ENDING_DAY_ID NUMBER ,
CALENDAR_MONTH_NUMBER NUMBER(2,0) ,
FISCAL_MONTH_NUMBER NUMBER(2,0) ,
CALENDAR_MONTH_DESC VARCHAR2(8) ,
CALENDAR_MONTH_ID NUMBER ,
FISCAL_MONTH_DESC VARCHAR2(8) ,
FISCAL_MONTH_ID NUMBER ,
DAYS_IN_CAL_MONTH NUMBER ,
DAYS_IN_FIS_MONTH NUMBER ,
END_OF_CAL_MONTH DATE ,
END_OF_FIS_MONTH DATE ,
CALENDAR_MONTH_NAME VARCHAR2(9) ,
FISCAL_MONTH_NAME VARCHAR2(9) ,
CALENDAR_QUARTER_DESC CHAR(7) ,
CALENDAR_QUARTER_ID NUMBER ,
FISCAL_QUARTER_DESC CHAR(7) ,
FISCAL_QUARTER_ID NUMBER ,
DAYS_IN_CAL_QUARTER NUMBER ,
DAYS_IN_FIS_QUARTER NUMBER ,
END_OF_CAL_QUARTER DATE ,
END_OF_FIS_QUARTER DATE ,
CALENDAR_QUARTER_NUMBER NUMBER(1,0) ,
FISCAL_QUARTER_NUMBER NUMBER(1,0) ,
CALENDAR_YEAR NUMBER(4,0) ,
CALENDAR_YEAR_ID NUMBER ,
FISCAL_YEAR NUMBER(4,0) ,
FISCAL_YEAR_ID NUMBER ,
DAYS_IN_CAL_YEAR NUMBER ,
DAYS_IN_FIS_YEAR NUMBER ,
END_OF_CAL_YEAR DATE ,
```

```

        END_OF_FIS_YEAR DATE '
    );
end;
/
begin
    dbms_cloud.create_external_table(
        table_name =>'COSTS_EXT',
        credential_name =>'OBJ_STORE_CRED',
        file_uri_list
=>'https://swiftobjectstorage.<region>.oraclecloud.com/v1/<identity_domain>/DEMO_
DATA/costs.dat',
        format => json_object('ignoremissingcolumns' value 'true', 'dateformat' value
'YYYY-MM-DD', 'blankasnull' value 'true'),
        column_list => 'PROD_ID NUMBER ,
        TIME_ID DATE ,
        PROMO_ID NUMBER ,
        CHANNEL_ID NUMBER ,
        UNIT_COST NUMBER(10,2) ,
        UNIT_PRICE NUMBER(10,2) '
    );
end;
/

```

## 6. Query\_Data\_External\_Script.sql

```

SELECT c.cust_id, t.calendar_quarter_desc, TO_CHAR (SUM(amount_sold),
    '9,999,999,999.99') AS Q_SALES, TO_CHAR(SUM(SUM(amount_sold))
OVER (PARTITION BY c.cust_id ORDER BY c.cust_id, t.calendar_quarter_desc
ROWS UNBOUNDED
PRECEDING), '9,999,999,999.99') AS CUM_SALES
FROM sales_ext s, times_ext t, customers_ext c
WHERE s.time_id=t.time_id AND s.cust_id=c.cust_id AND t.calendar_year=2000
AND c.cust_id IN (2595, 9646, 11111)
GROUP BY c.cust_id, t.calendar_quarter_desc
ORDER BY c.cust_id, t.calendar_quarter_desc;

```

## 7. Create\_DV\_View\_Script.sql

```
drop view DV_SH_VIEW;

create or replace view DV_SH_VIEW as select
P.PROD_NAME,
P.PROD_DESC,
P.PROD_CATEGORY,
P.PROD_SUBCATEGORY,
P.PROD_LIST_PRICE,
S.QUANTITY SOLD,
S.AMOUNT SOLD,
X.CUST_GENDER,
X.CUST_YEAR_OF_BIRTH,
X.CUST_MARITAL_STATUS,
X.CUST_INCOME_LEVEL,
R.COUNTRY_NAME,
R.COUNTRY_SUBREGION,
R.COUNTRY_REGION,
T.TIME_ID,
T.DAY_NAME,
T CALENDAR_MONTH_NAME,
T.CALENDAR_YEAR from
SH.PRODUCTS P,
SH.SALES S,
SH.CUSTOMERS X,
SH.COUNTRIES R,
SH.TIMES T where
S.PROD_ID=P.PROD_ID and
S.CUST_ID=X.CUST_ID and
S.TIME_ID=T.TIME_ID and
X.COUNTRY_ID=R.COUNTRY_ID;
```