**Oracle Autonomous Data Warehouse: Getting Started Workshop - Lab 4**

**Introduction**

In this lab, you will be querying files on the Oracle Cloud Infrastructure Object Storage (OCI) directly without loading them to your database.

**Objectives**

* Learn how to create external tables on top of files residing on the object store
* Learn how to query external data by the external tables
* Learn how to create data warehouse user

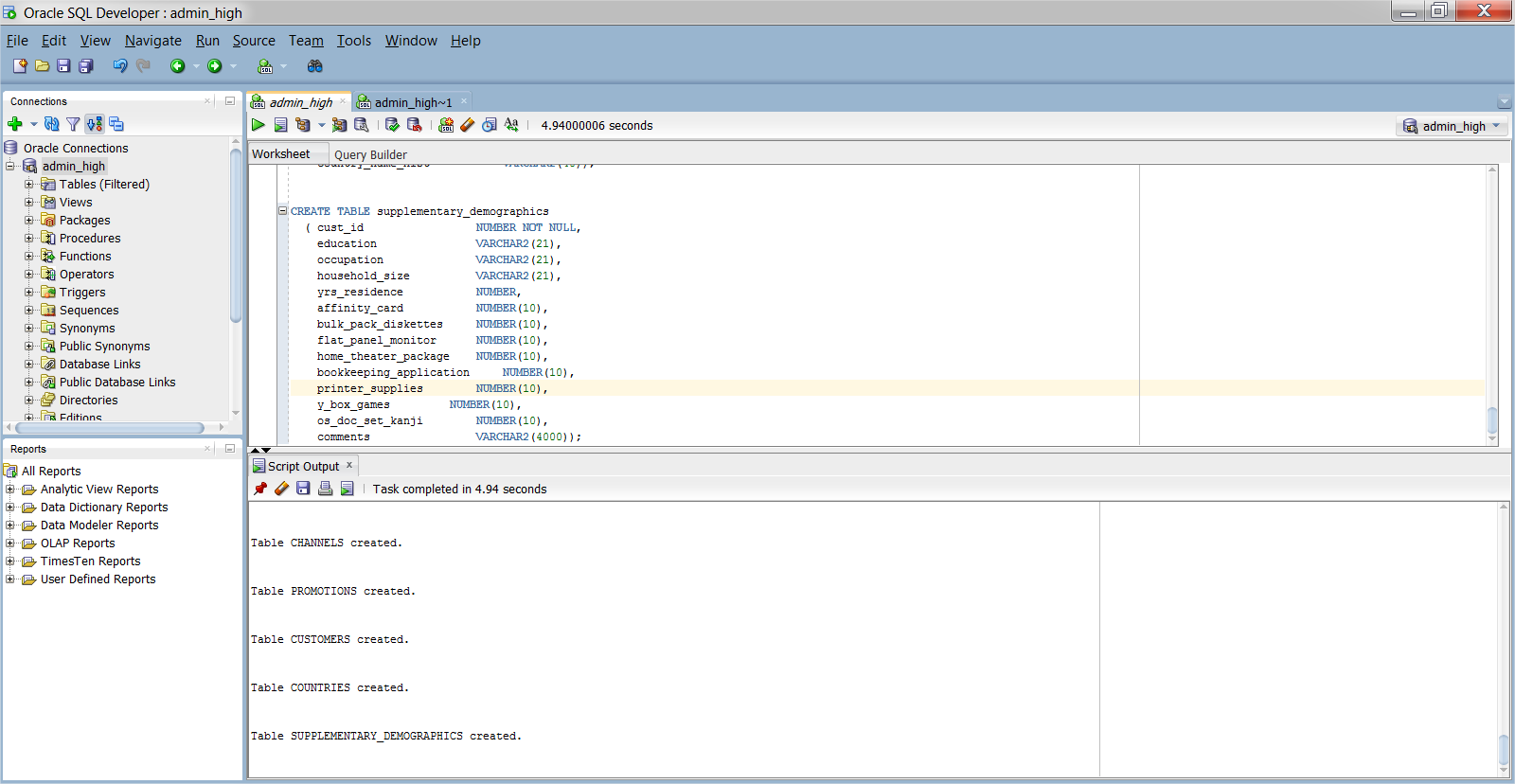
**Required Artifacts**

* The following lab requires an Oracle Cloud account. You may use your own cloud account, a cloud account that you obtained through a trial, or a training account whose details were given to you by an Oracle instructor.
* Oracle SQL Developer (see Lab 1for more specifics on the version of SQL Developer and how to install and configure it).

# Create the target tables

#### ****STEP 3: Create Target Tables for Data Loading****

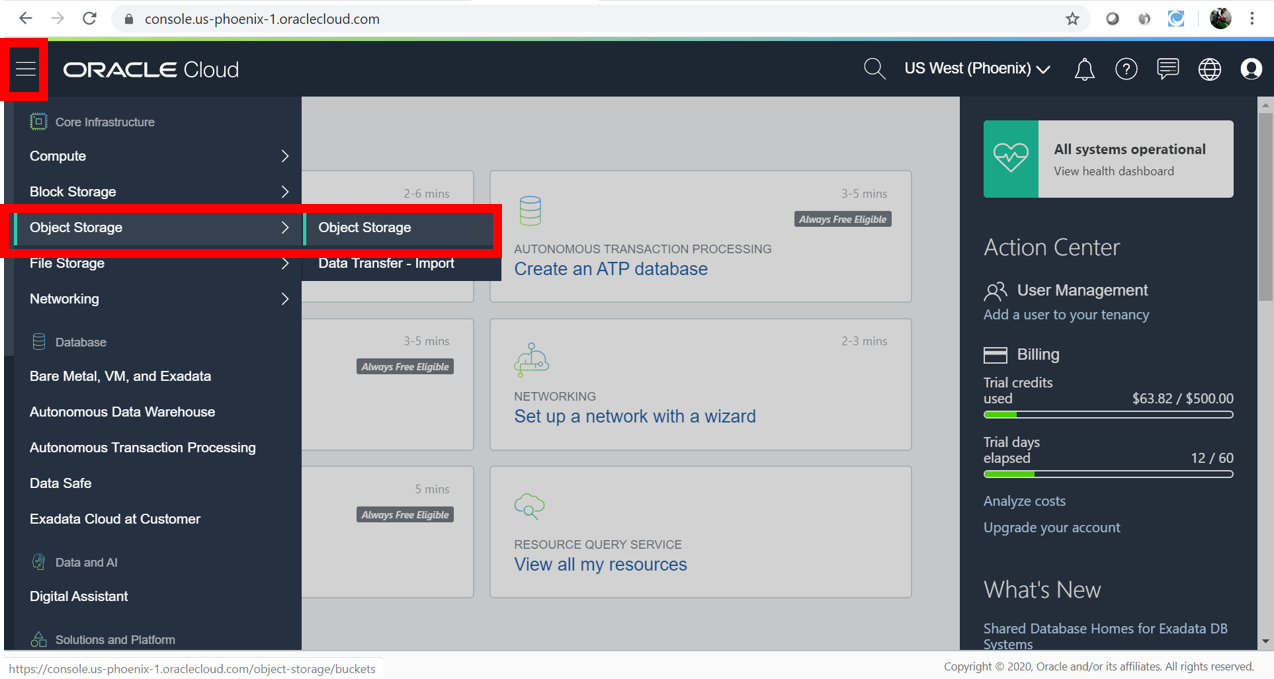
* Connected as your user in SQL Developer, copy and paste [this code snippet](https://raw.githubusercontent.com/oraclehubteam/workshops/master/createTables.txt) to SQL Developer worksheet. Take a moment to examine the script. Then click the **Run Script** button to run it.
* It is expected that you may get ORA-00942 errors during the DROP TABLE commands for the first execution of the script, but you should not see any other errors.



Note that you do not need to specify anything other than the list of columns when creating tables in the SQL scripts. You can use primary keys and foreign keys if you want, but they are not required.

# Setup the OCI Object Store

#### ****STEP 4: Navigate to Object Storage****

* From the oracle cloud console, pull out the left side menu from the top-left corner and select **Object Storage**. 

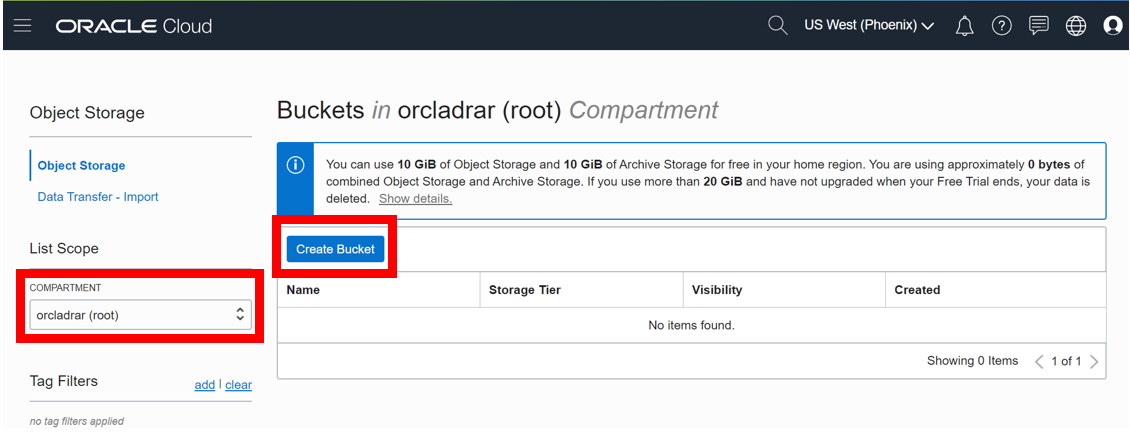
To learn more about the OCI Object Storage, refer to its [documentation](https://docs.us-phoenix-1.oraclecloud.com/Content/GSG/Tasks/addingbuckets.htm) .

* You should now be on the **Object Storage** page. Choose the **root** compartment in the **Compartment** drop-down if it is not already chosen.

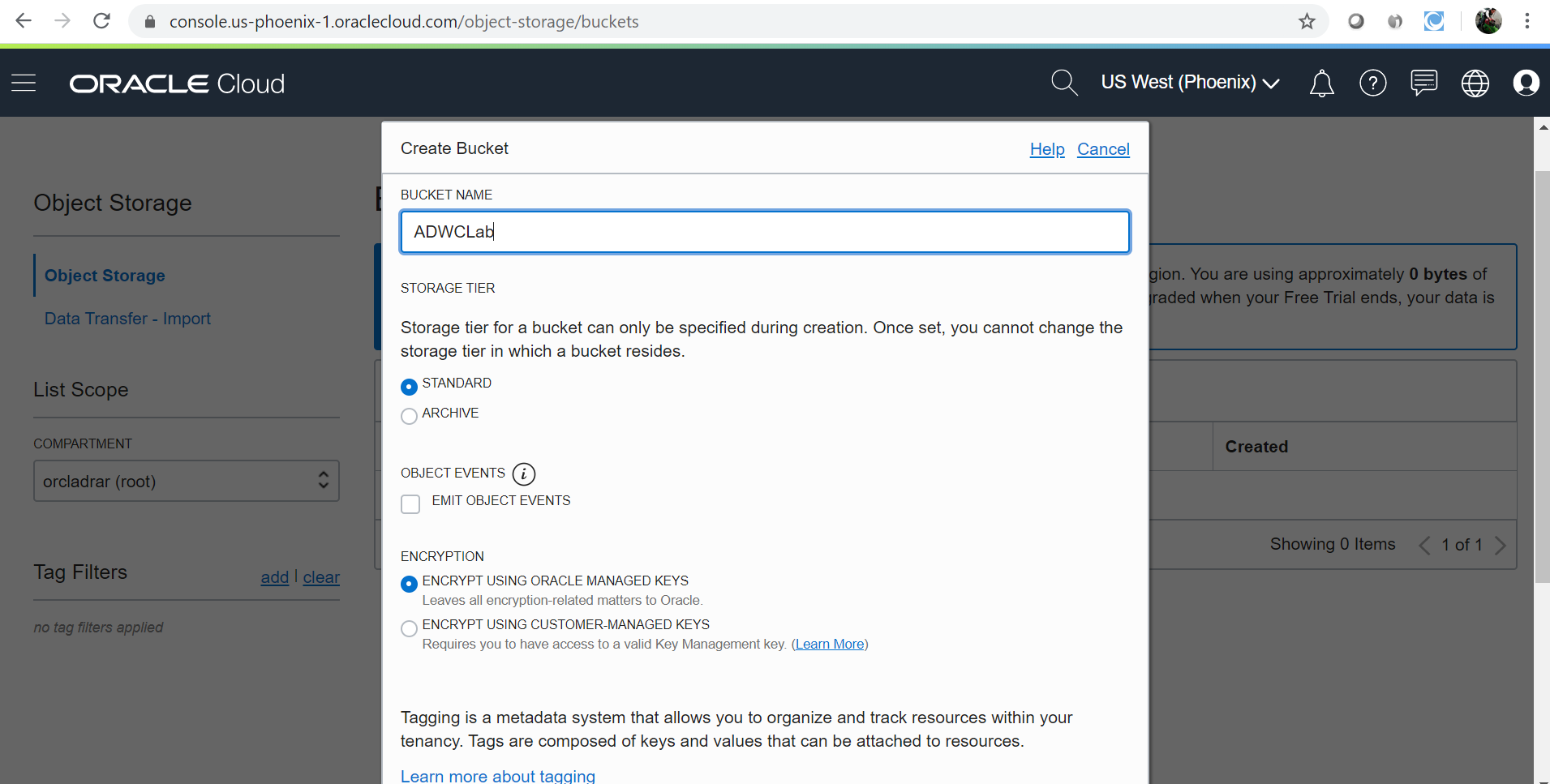
#### ****STEP 5: Create a Bucket for the Object Storage****

In OCI Object Storage, a bucket is the terminology for a container of multiple files.

* Click the **Create Bucket** button:

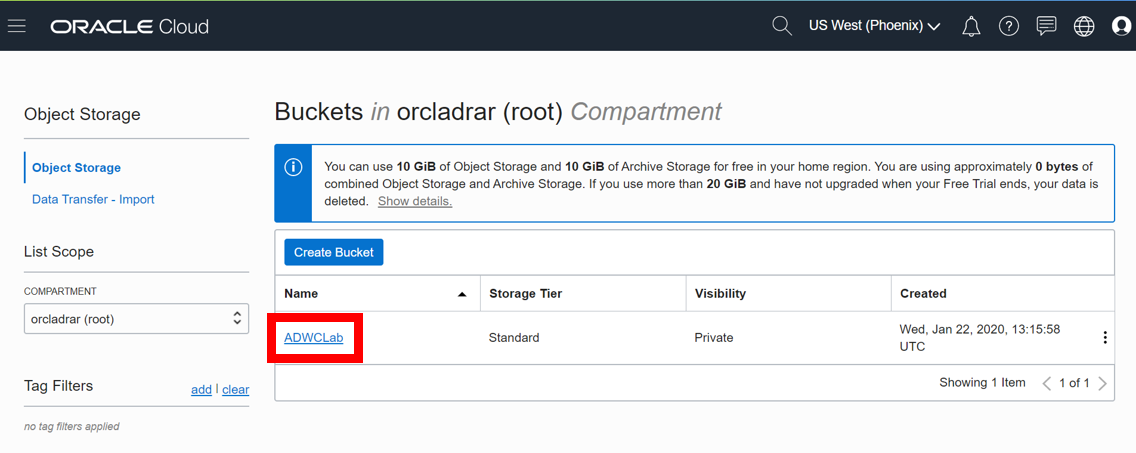


* Name your bucket **ADWCLab** and click the **Create Bucket** button.

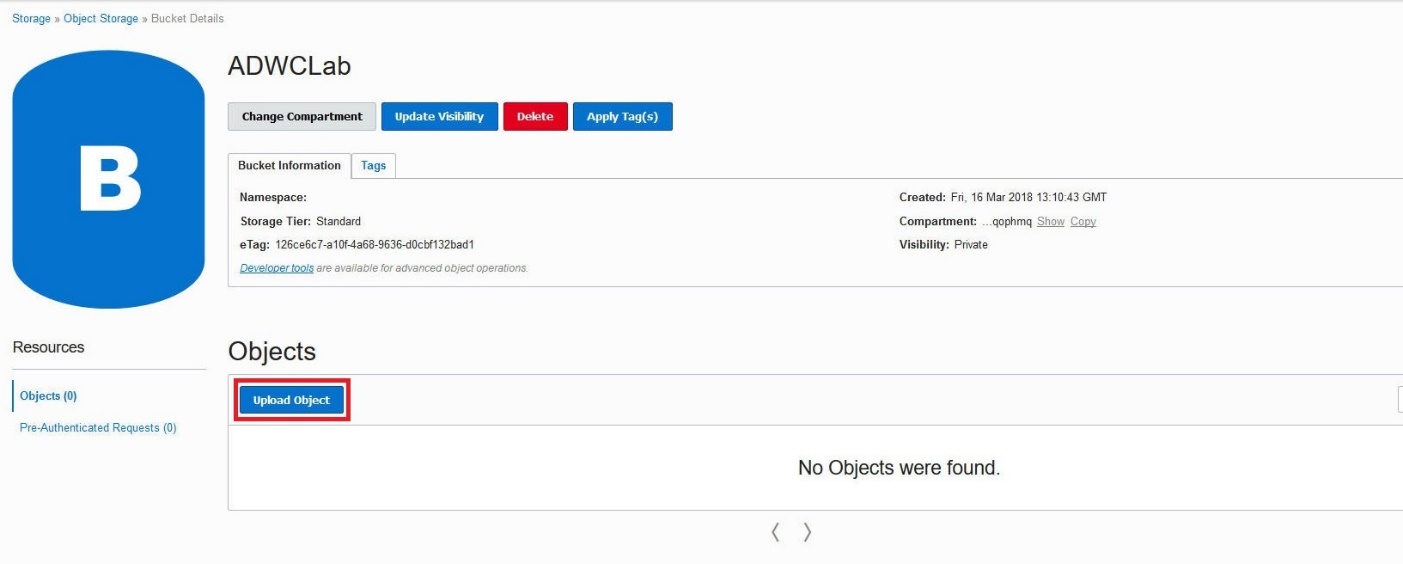


#### ****STEP 6: Upload Files to Your OCI Object Store Bucket****

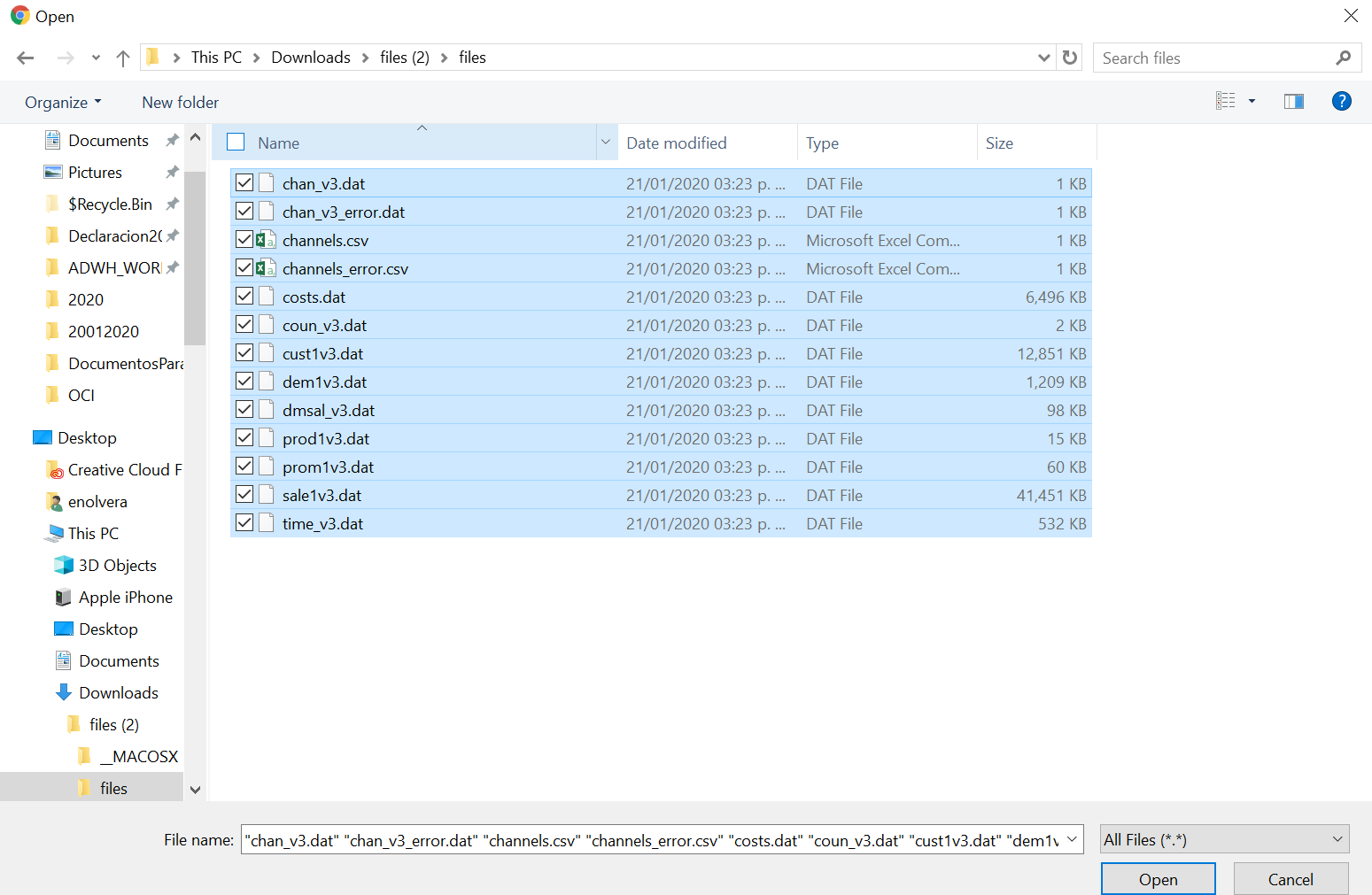
* Click on your **bucket name** to open it:



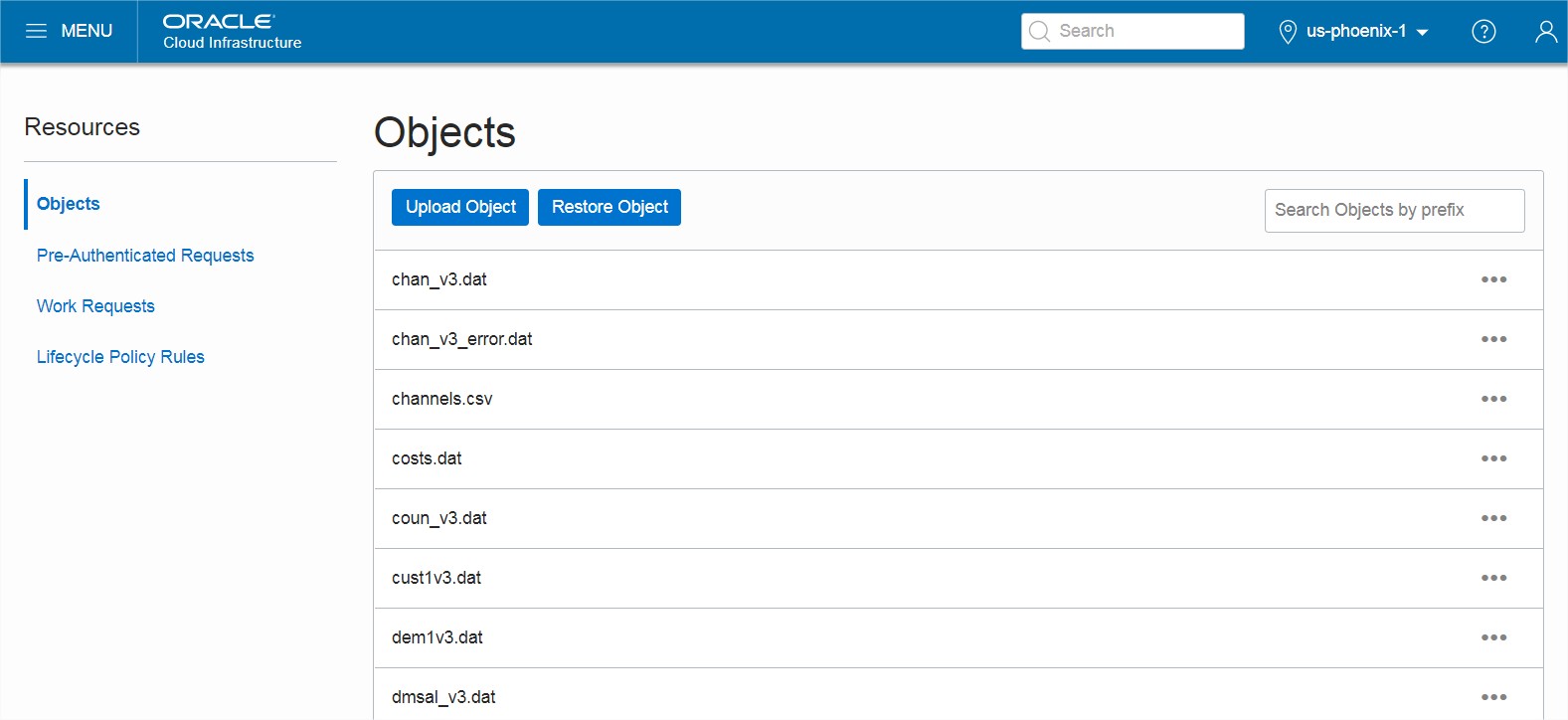
* Click on the **Upload Object** button:



* Using the browse button or select all the files downloaded in the earlier step, click Upload and wait for the upload to complete:

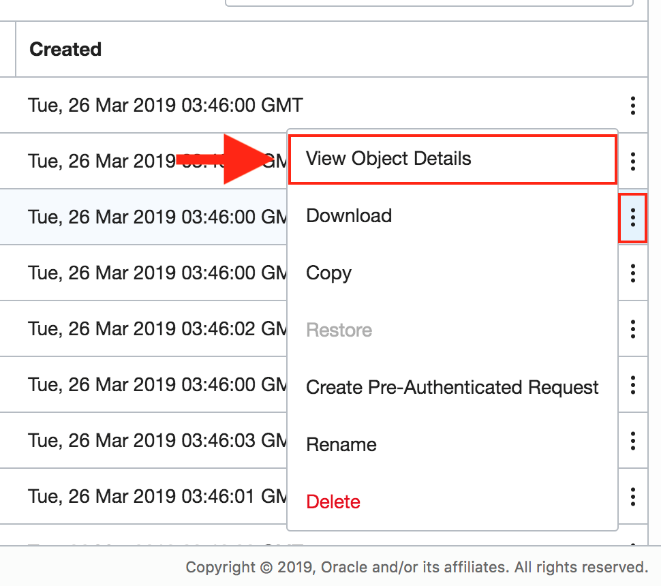


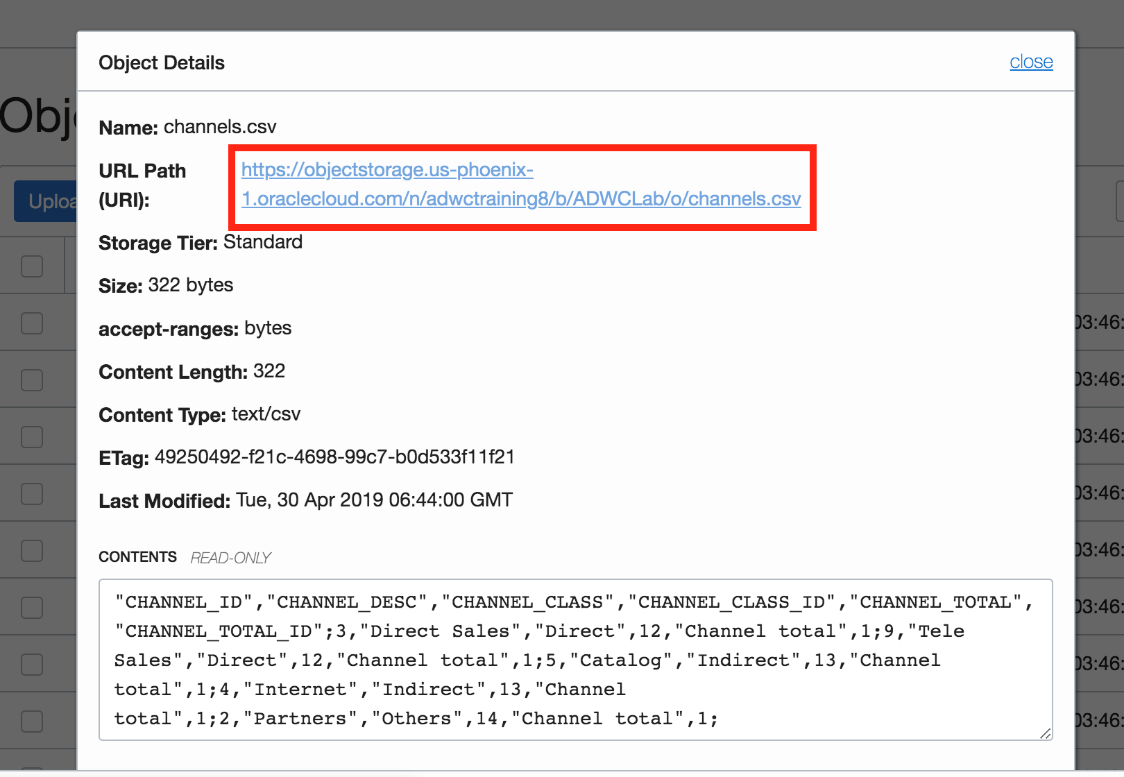
* The end result should look like this with all files listed under Objects:



#### ****STEP 7: Copy the URL for the Files in Your OCI Object Storage****

* Copy following base URL that points to the location of your files staged in the OCI Object Storage. The simplest way to get this URL is from the "Object Details" in the right hand side ellipsis menu in the Object Store.





* + Take a look at the URL you copied. In this example above, the **region name** is us-phoenix-1, the **name space** is axk1ywem6xup, and the **bucket name** is ADWCLab.

Note: The URL can also be constructed as below:

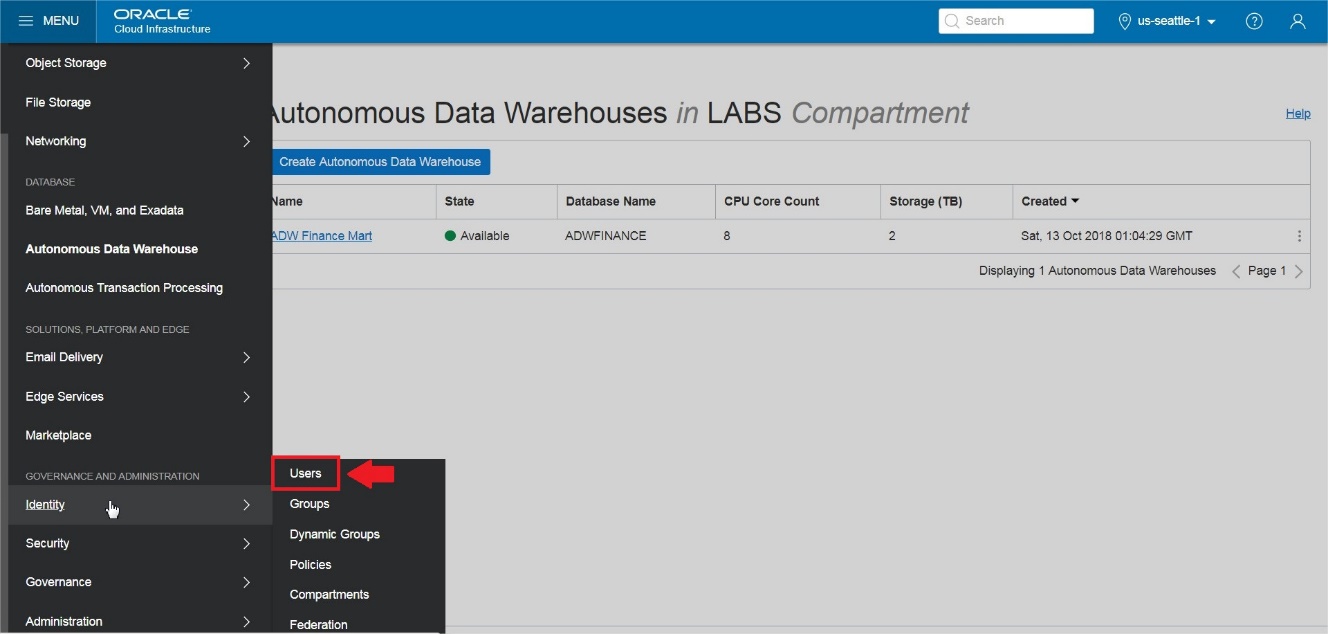
https://objectstorage.<**region\_name**>.oraclecloud.com/n/<**tenant\_name**>/b/<**bucket\_name**>/o

* **Save** the base URL you copied in a note. We will use the base URL in following steps.

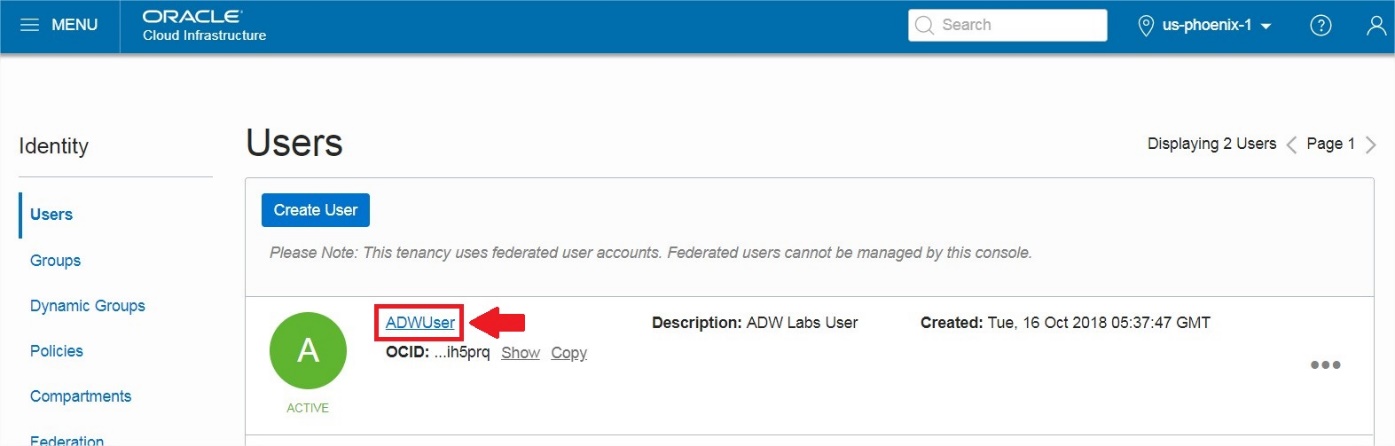
#### ****STEP 8: Creating an Object Store Auth Token****

To load data from the Oracle Cloud Infrastructure(OCI) Object Storage you will need an OCI user with the appropriate privileges to read data (or upload) data to the Object Store. The communication between the database and the object store relies on the Swift protocol and the OCI user Auth Token.

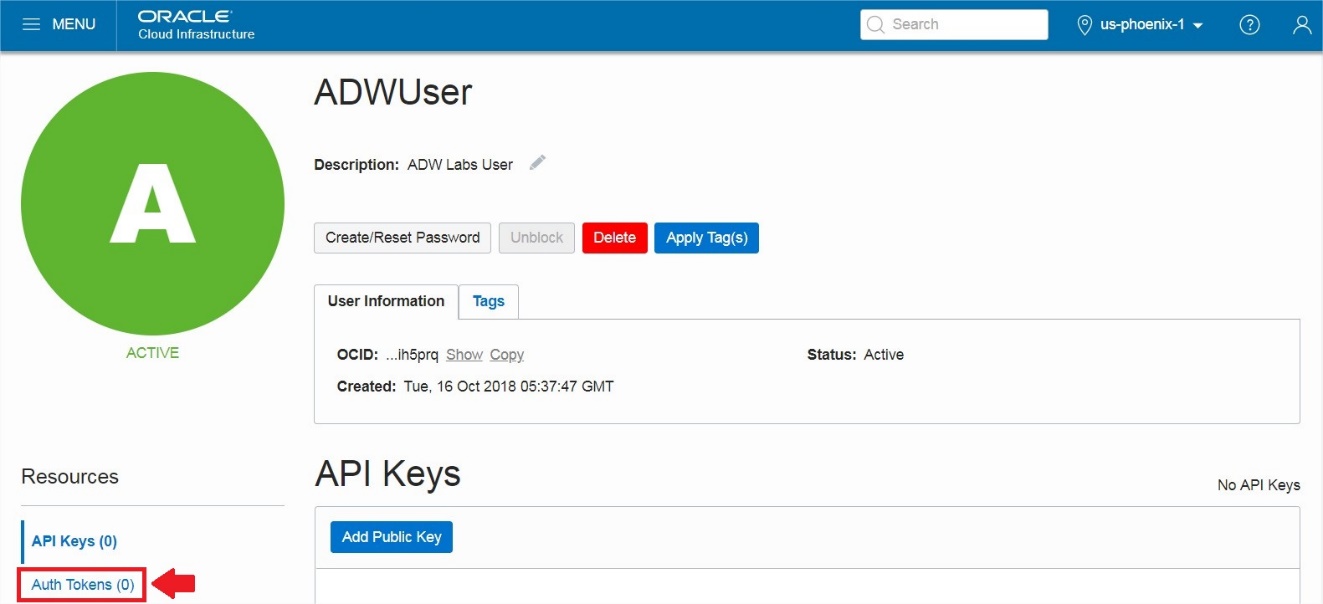
* Go back to the **Autonomous Data Warehouse Console** in your browser. From the pull-out menu on the top left, under **Identity**, click **Users**.



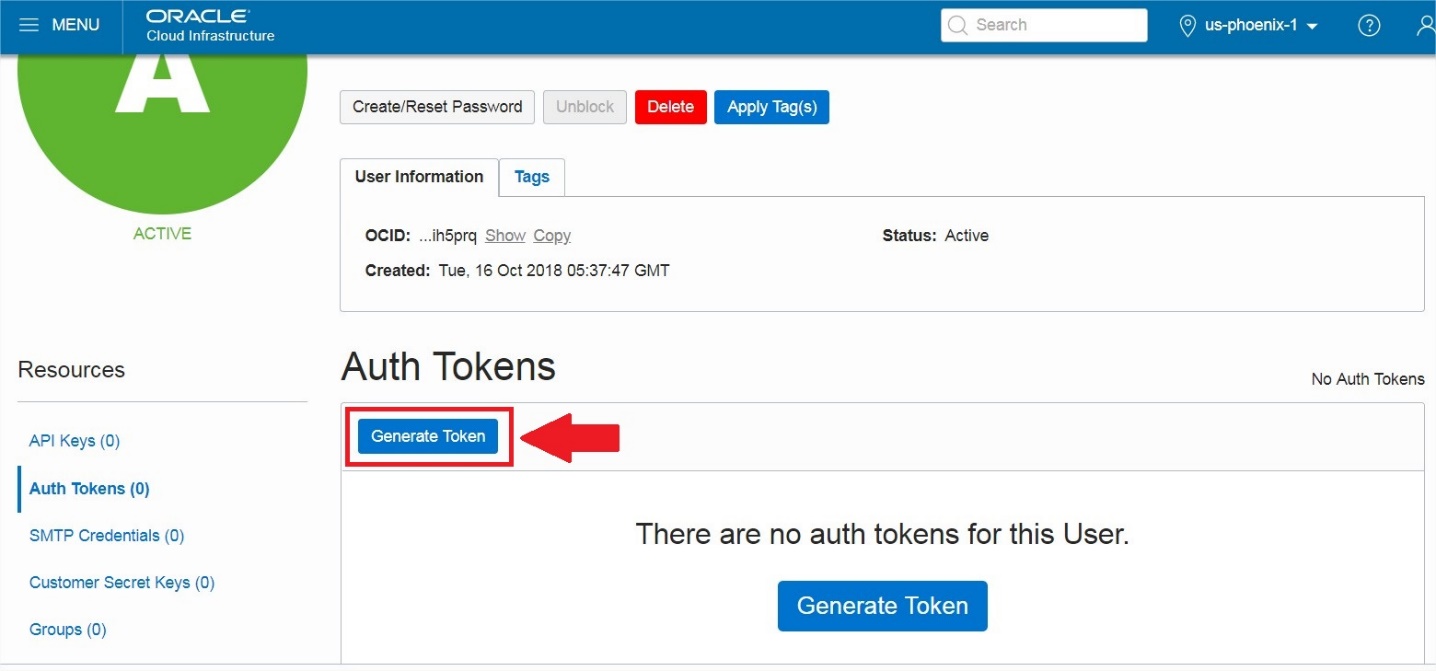
* Click the **user's name** to view the details. Also, remember the username as you will need that in the next step. This username could also be an email address.



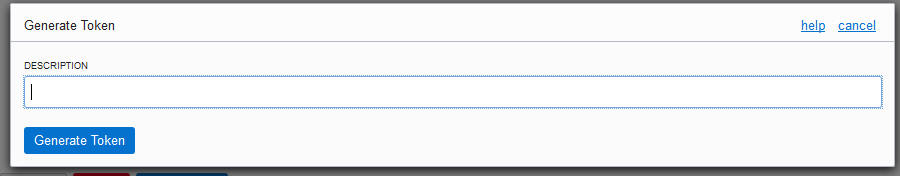
* On the left side of the page, click **Auth Tokens**.



* Click **Generate Token**.



* Enter a friendly **description** for the token and click **Generate Token**.



* The new Auth Token is displayed. Click **Copy** to copy the Auth Token to the clipboard. You probably want to save this in a temporary notepad document for the next few minutes (you'll use it in the next step).  
  Note: You can't retrieve the Auth Token again after closing the dialog box.

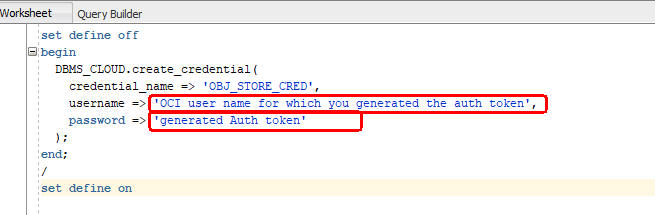


#### ****STEP 9: Create a Database Credential for Your User****

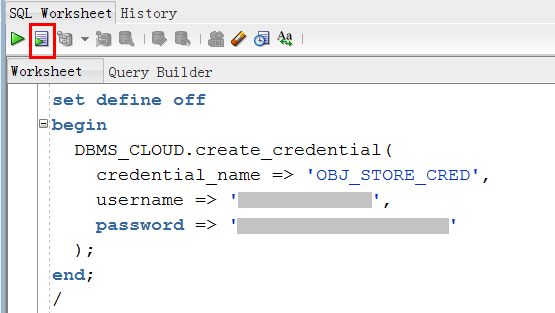
In order to access data in the Object Store you have to enable your database user to authenticate itself with the Object Store using your OCI object store account and Auth token. You do this by creating a private CREDENTIAL object for your user that stores this information encrypted in your Autonomous Data Warehouse. This information is only usable for your user schema.

* Connected as your user in SQL Developer, copy and paste [this code snippet](https://raw.githubusercontent.com/oraclehubteam/workshops/master/create_credential.txt) to SQL Developer worksheet.

Specify the credentials for your Oracle Cloud Infrastructure Object Storage service: The username will be your **OCI username** (usually your email address, not your database username) and the password is the OCI Object Store **Auth Token** you generated in the previous step. In this example, the credential object named **OBJ\_STORE\_CRED** is created. You reference this credential name in the following steps.



* Run the script.

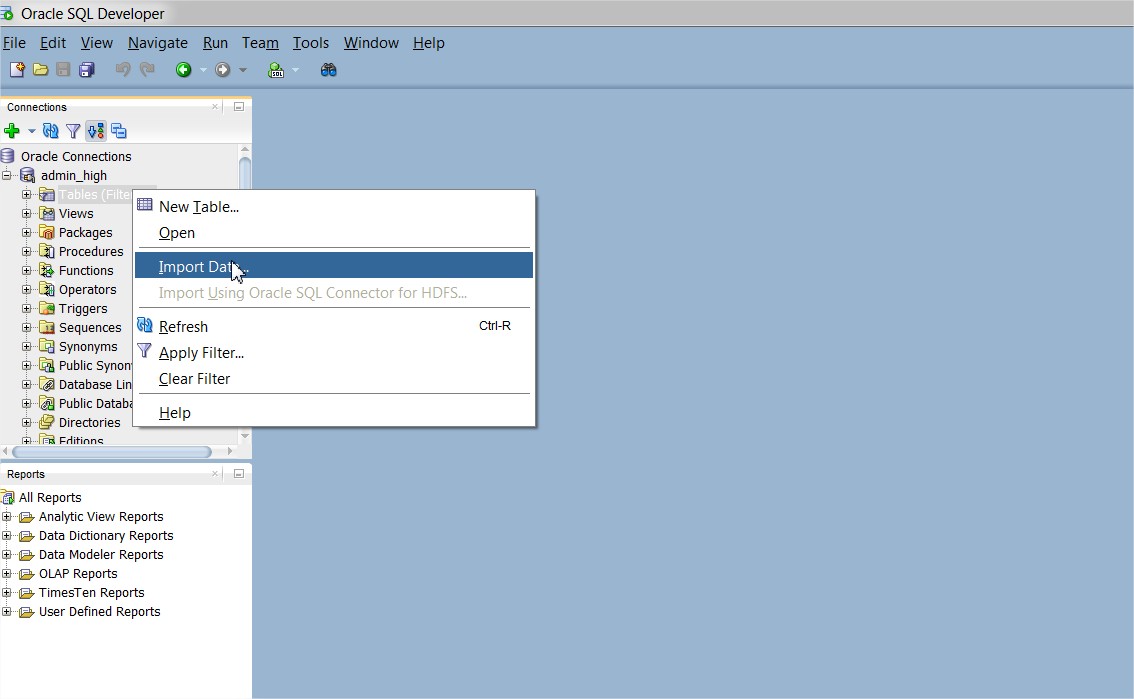


* Now you are ready to load data from the Object Store.

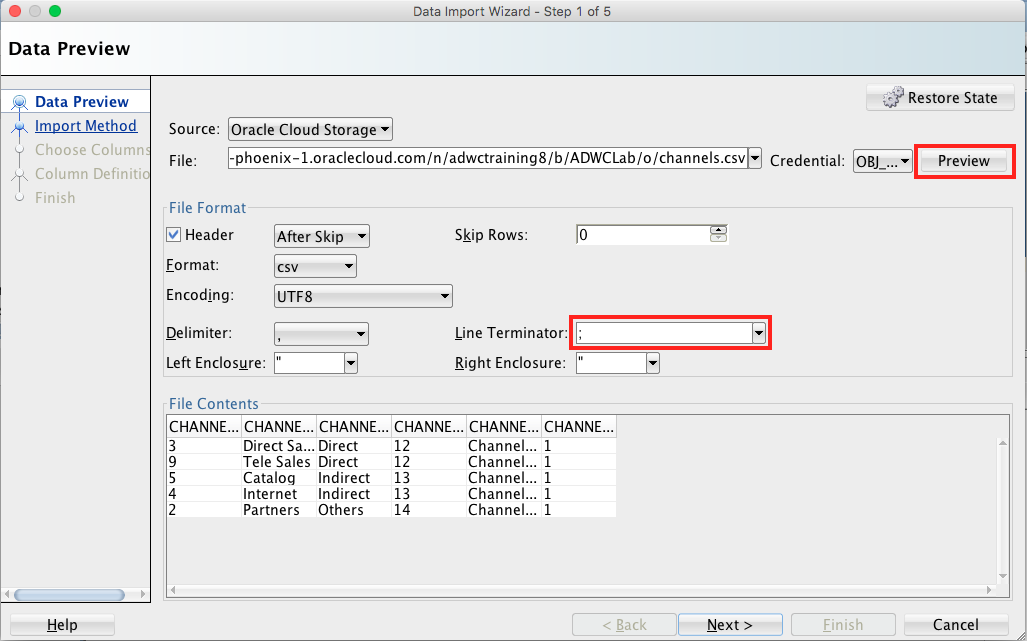
# Load data from the Object Store using the Data Import Wizard

#### ****STEP 10: Loading Data Using the Data Import Wizard in SQL Developer****

* Click ‘**Tables**’ in your user schema object tree. Clicking the right mouse button opens the context-sensitive menu in SQL Developer; select ‘**Import Data**’:



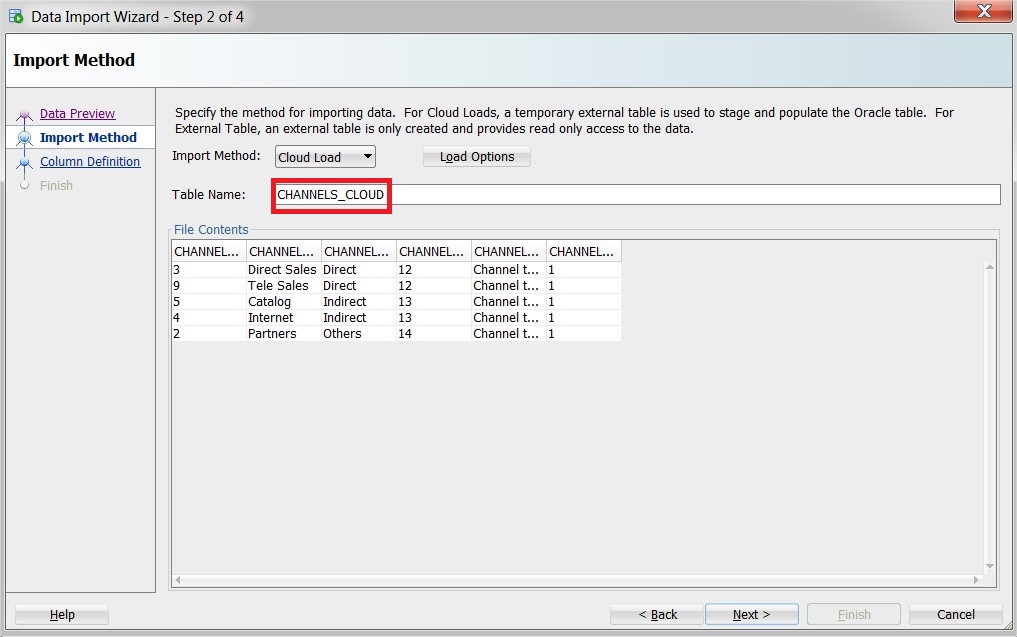
* The Data Import Wizard is started. Enter the following information:
  + Select **Oracle Cloud Storage** as source for the data load
  + Enter the URL of **channels.csv** as the file to load. You copied or constructed the URL in STEP 8 Copy the URLs of the Files on Your OCI Object Storage. For example, the URL might look something like:  
    https://objectstorage.us-phoenix-1.oraclecloud.com/n/labs/b/ADWCLab/o/channels.csv
  + Select the Credential you previously created for authentication with the Object Store, **OBJ\_STORE\_CRED**
  + You might see a read error since channels.csv has lines terminated by ";", as is sometimes the case with certain data. Manually enter a semi-colon in the **line terminator** box to properly parse the data.
  + Click the **Preview** button



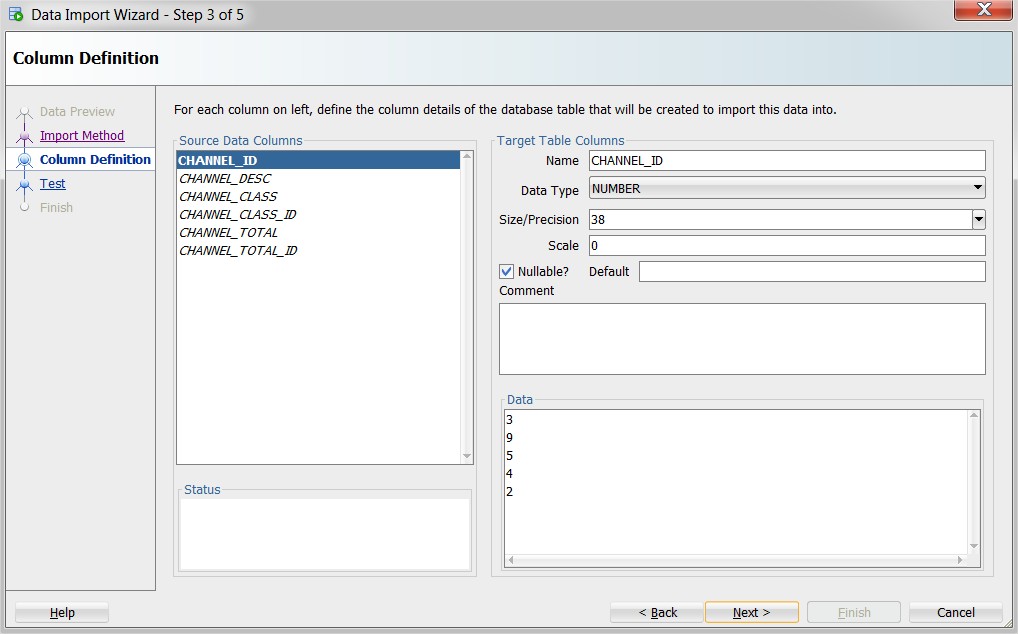
When you are satisfied with the data preview, click **NEXT**.

Note: If you see an object not found error here, your user may not be set up properly to have data access to the object store. Please contact your Cloud Administrator.

* On the Import Method page, you can click on Load Options to see some of the available options. For this exercise, leave the options at their defaults. Enter CHANNELS\_CLOUD as the table name and click **NEXT** to advance to the next page of the wizard.

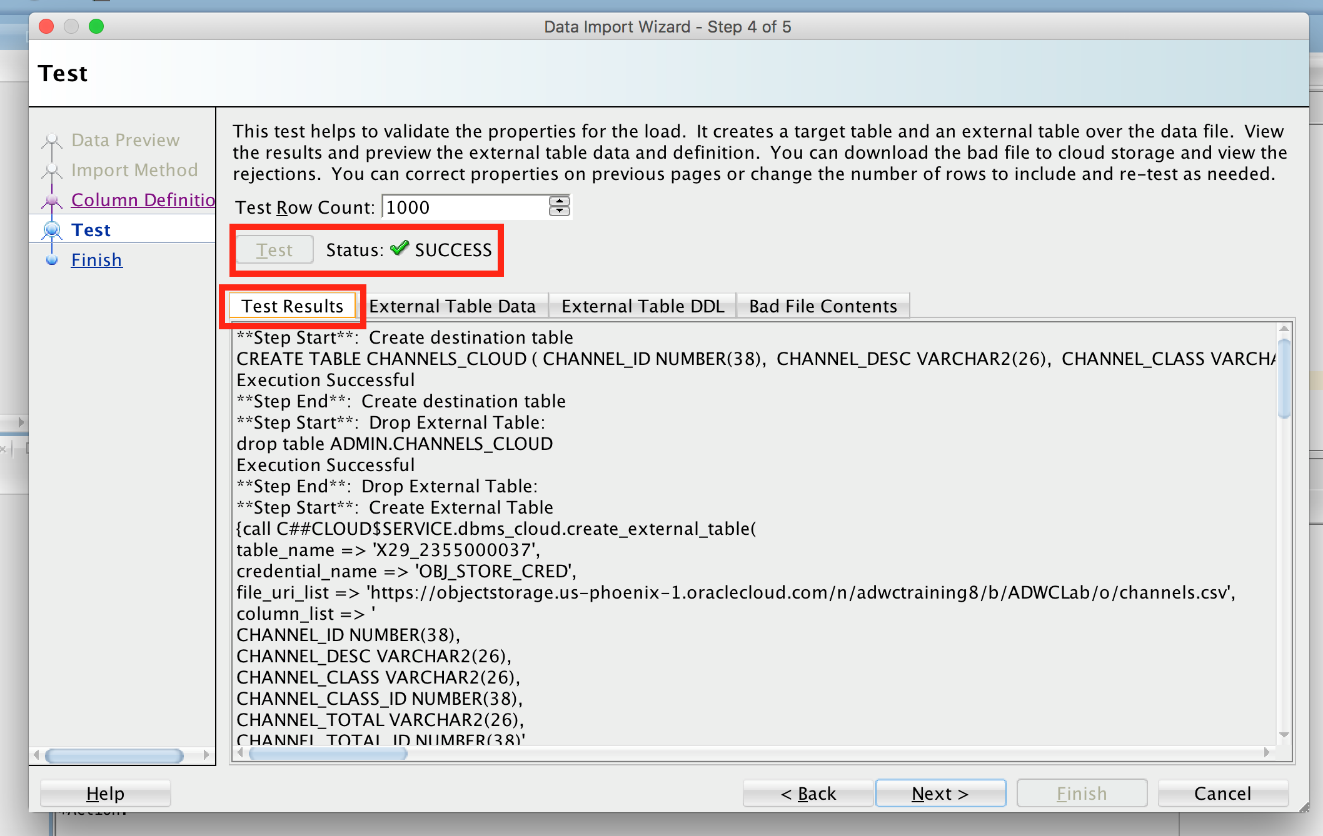


* On the Column Definition page, you can control how the fields of the file map to columns in the table. You can also adjust certain properties such as the Data Type of each column. This data needs no adjustment, we can simply proceed by clicking **Next**.



* The last screen before the final data load enables you to test a larger row count than the sample data of the beginning of the wizard, to see whether the previously made decisions are satisfying for your data load. Note that we are not actually loading any data into your database during these Tests. Click **TEST** and look at the Test Results log, the data you would load, any mistakes and what the external table definition looks like based on your inputs.

When done with your investigation, click **NEXT**.



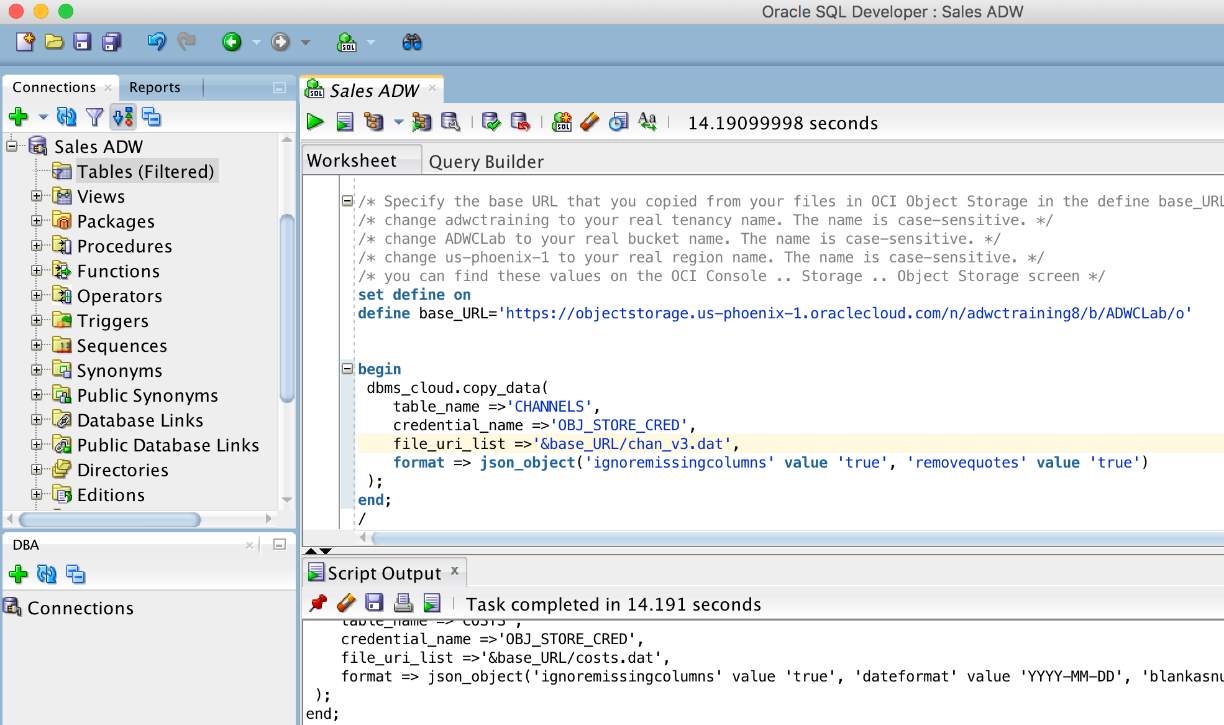
* The final screen reflects all your choices made in the Wizard. Click **FINISH** when you are ready to load the data into the table CHANNELS\_CLOUD.

# Load data from the Object Store using DBMS\_CLOUD

#### ****STEP 11: Loading Data Using the PL/SQL Package, DBMS\_CLOUD****

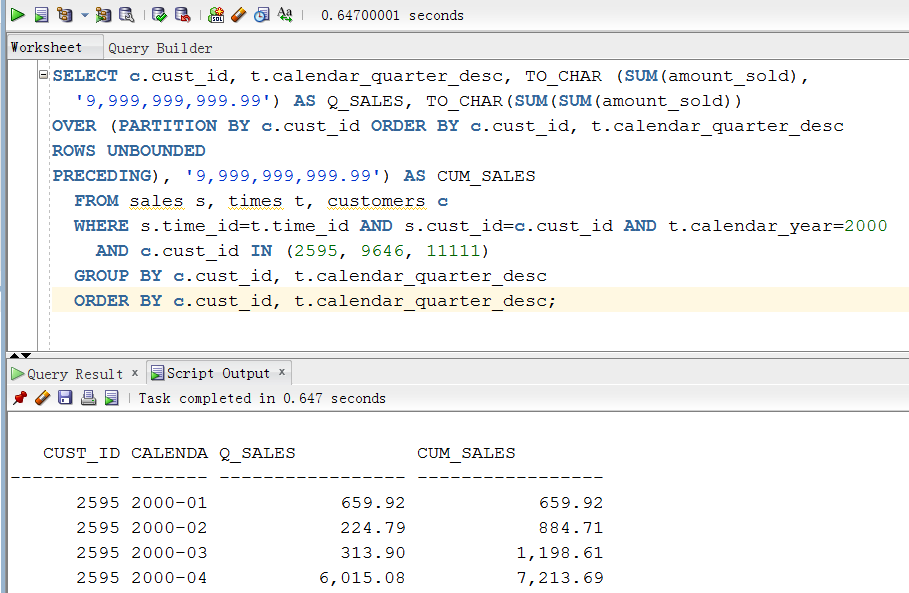
As an alternative to the wizard-guided data load, you can use the PL/SQL package **DBMS\_CLOUD** directly. This is the preferred choice for any load automation.

* Connected as your user in SQL Developer, copy and paste [this code snippet](https://raw.githubusercontent.com/oraclehubteam/workshops/master/load_data.txt) to SQL Developer worksheet. This scripts uses the **copy\_data** procedure of the **DBMS\_CLOUD** package to copy the data in the source files to the target tables you created before.
  + At the top of the script, specify the Object Store base URL in the definition of the **base\_URL** variable. You have copied and saved this URL in the step "Copy the URLs of the Files on Your OCI Object Storage" above.
  + For the **credential\_name** parameter in the **copy\_data** procedure, it is the name of the credential you defined in the step "Create a Database Credential for Your User" above. You can use that credential.
  + For the **format** parameter, it is a list of DBMS\_CLOUD options (you can read more about these options [here](https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/user/dbmscloud-reference.html)).
  + Run the script.



* + You have successfully loaded the sample tables. You can now run any sample query in the [relational analytics](https://docs.oracle.com/database/122/DWHSG/part-relational-analytics.htm#DWHSG8493) section of the Oracle documentation. For example, to analyze the cumulative amount sold for specific customer IDs in quarter 2000, you could run the query in [this code snippet](https://raw.githubusercontent.com/oraclehubteam/workshops/master/query_tables.txt).

[Click Here](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/introduction-data-warehouse-concepts.html#GUID-452FBA23-6976-4590-AA41-1369647AD14D) to read more about Data Warehousing.



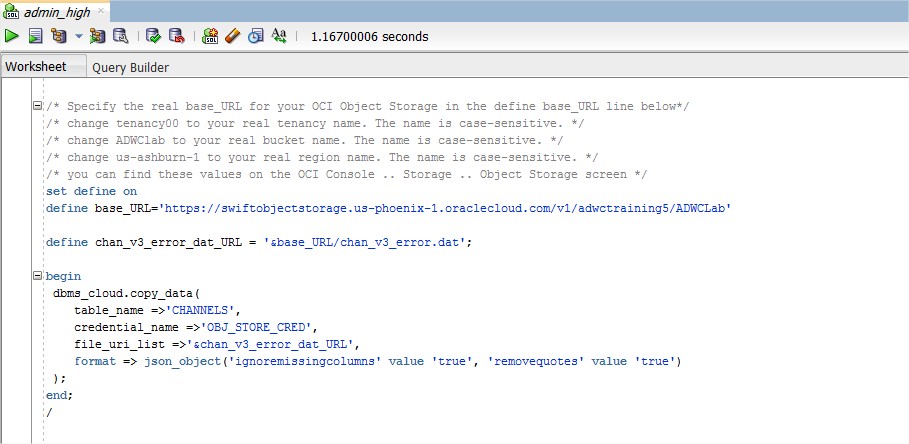
#### ****STEP 12: Troubleshooting DBMS\_CLOUD data loads****

* Connected as your user in SQL Developer, run the following query to look at past and current data loads.

select \* from user\_load\_operations;

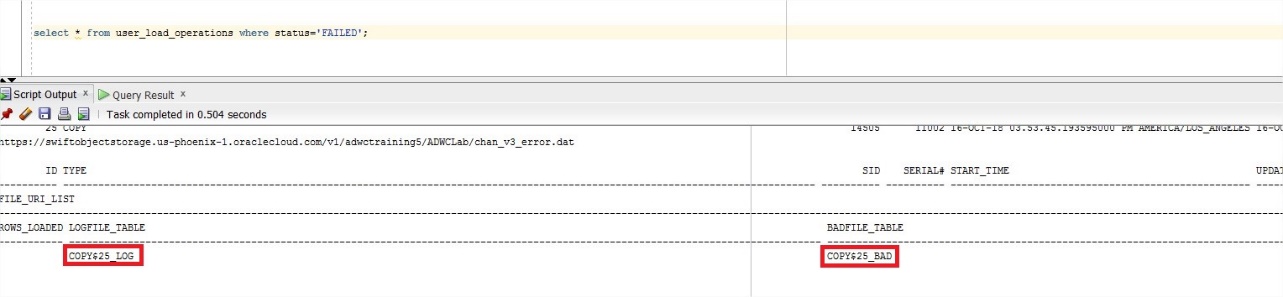
Notice how this table lists the past and current load operations in your schema. Any data copy and data validation operation will have backed up records in your Cloud.

* For an example of how to troubleshoot a data load, we will attempt to load a data file with the wrong format (chan\_v3\_error.dat). Specifically, the default separator is the | character, but the channels\_error.csv file uses a semicolon instead. To attempt to load bad data, copy and paste [this code snippet](https://raw.githubusercontent.com/oraclehubteam/workshops/master/load_data_with_errors.txt) to a SQL Developer worksheet and run the script as your user in SQL Developer. Specify the URL that points to the **chan\_v3\_error.dat** file. You have copied and saved the URL in the step "Copy the URLs of the Files on Your OCI Object Storage" above. Expect to see see "reject limit" errors when loading your data this time.



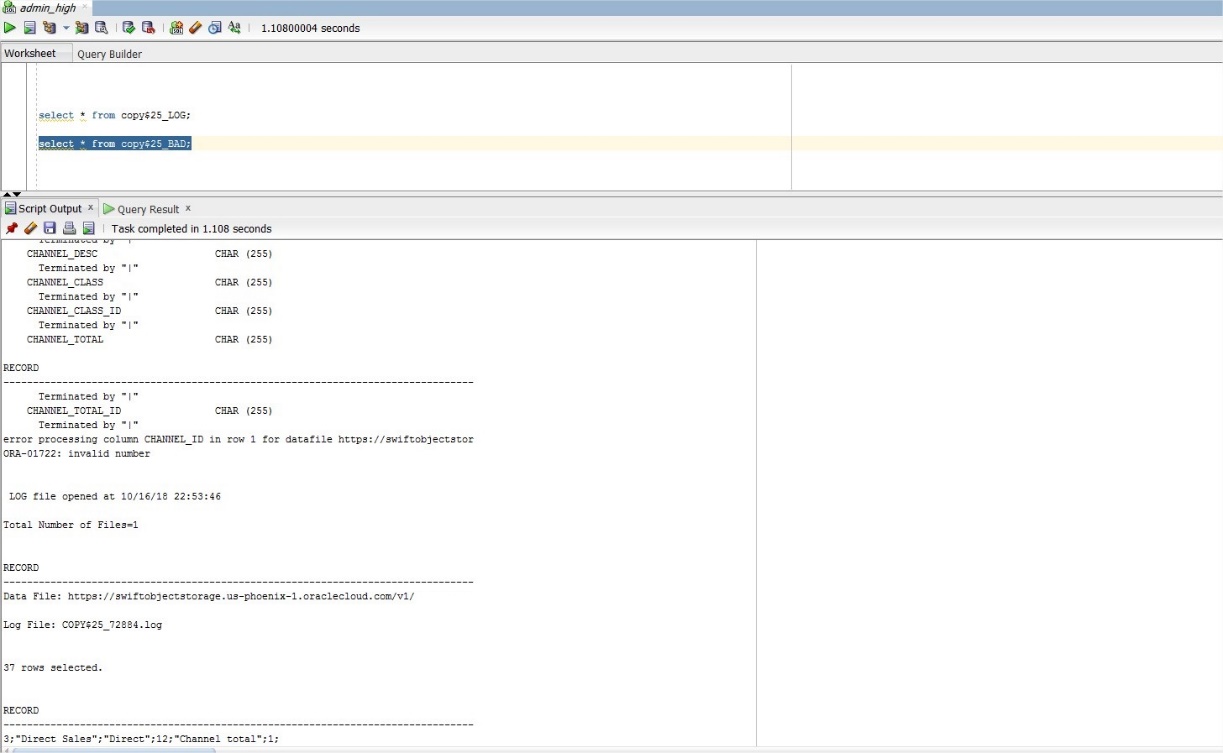
* Run the following query to see the load that errored out.

select \* from user\_load\_operations where status='FAILED';



A load or external table validation that errors out is indicated by status=FAILED in this table. Get the names of the log and bad files for the failing load operation from the column **logfile\_table** and **badfile\_table**. The logfile\_table column shows the name of the table you can query to look at the log of a load operation. The column badfile\_table shows the name of the table you can query to look at the rows that got errors during loading.

* Query the log and bad tables to see detailed information about an individual load. In this example, the names are copy$25\_log and copy$25\_bad respectively.



* To learn more about how to specify file formats, delimiters, reject limits, and more, review the [DBMS\_CLOUD Package Format Options](https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/user/dbmscloud-reference.html)
* Keep your SQL Deveoper open and move to the next lab - Querying External Data.

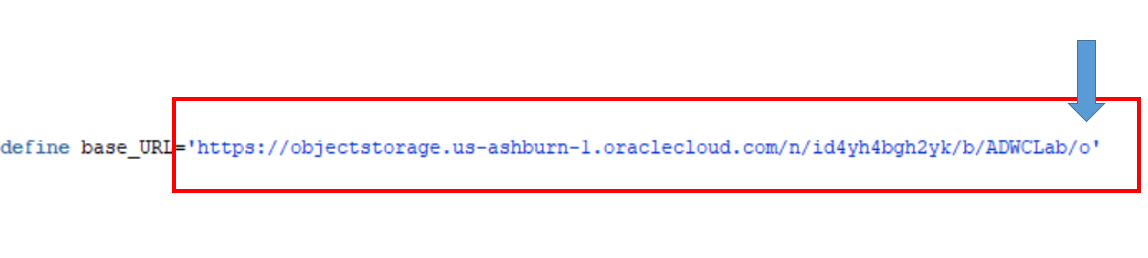
**Querying External Data**

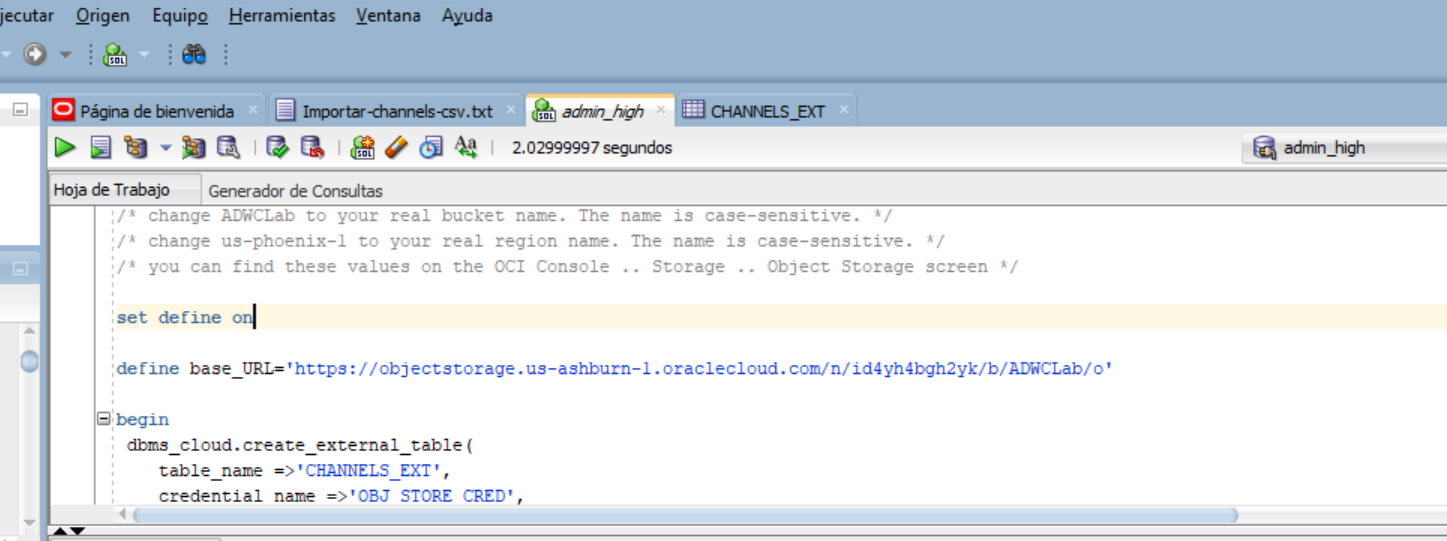
**STEP 1: Create External Tables with DBMS\_CLOUD**

* Connected as your user in SQL Developer, copy and paste this code 🡪 [create\_external\_tables.txt](https://raw.githubusercontent.com/oraclehubteam/workshops/master/create_external_tables.txt) to SQL Developer worksheet.

Use the **create\_external\_table** procedure of the **DBMS\_CLOUD** package to create external tables on the files staged in your object store. Note that you are still using the same credential and the URLs of files on OCI Object Storage you used when loading data in the previous lab.

* + At the top of the script, specify the Object Store base URL in the definition of the **base\_URL**variable.

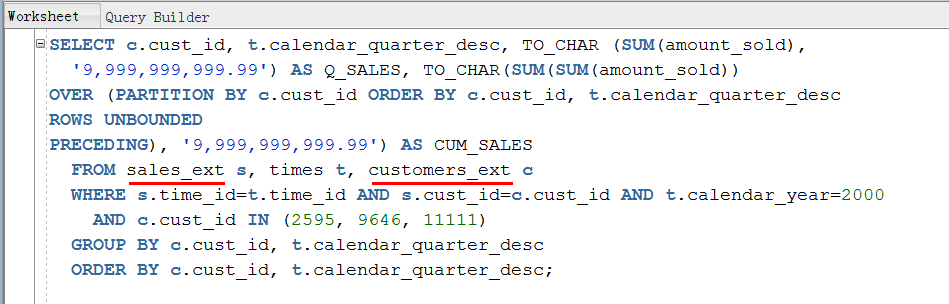




* + **Run the script**.

Now you have **external tables** for the sample data pointing to files in the object store. Any query against the external tables will return the same result as against the original tables.

**STEP 2: Querying External Data**

* Copy and paste this code query-->[external\_data.txt](https://raw.githubusercontent.com/oraclehubteam/workshops/master/external_data.txt) to SQL Developer worksheet. We only replaced the original table names <TABLE\_NAME> with <TABLE\_NAME\_EXT> in the sample query.   
  

**Run the script**. You will now see the same query result as before, from data pulled directly from the Object Store