



Autonomous Data Warehouse Cloud

Updated: June 19, 2018

Lab 10: Machine Learning and Notebooks

In this lab you will be using the new Oracle Machine Learning notebook application provided with ADWC. This browser-based application provides a web interface to run SQL queries and scripts, which can be grouped together within a notebook. Notebooks can be used to build single reports, collections of reports and dashboards. OML provides a simple way to share workbooks with other OML users.

Objectives

- Create OML Users
- Exploring OML Home Page
- Run a SQL Statement
- Sharing Notebooks

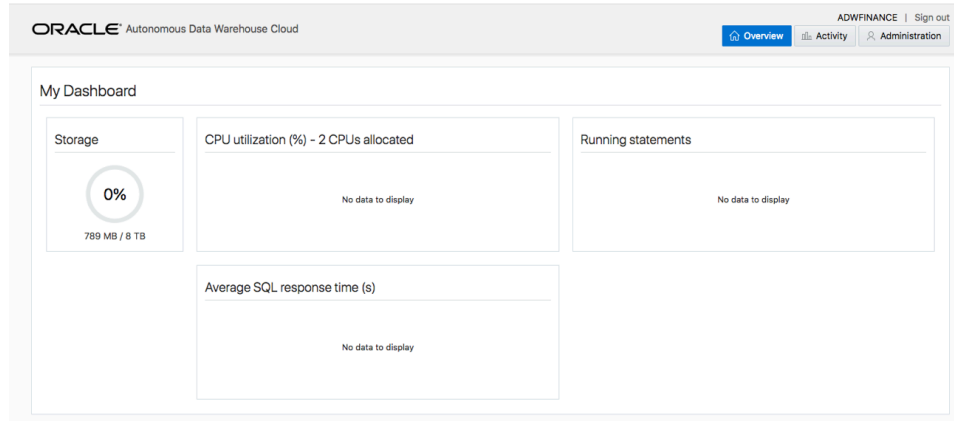
Required Artifacts

- The lab requires an Oracle Autonomous Data Warehouse Cloud subscription.
- Locate your **Cloud Account Name**, **Username**, and **Password**.

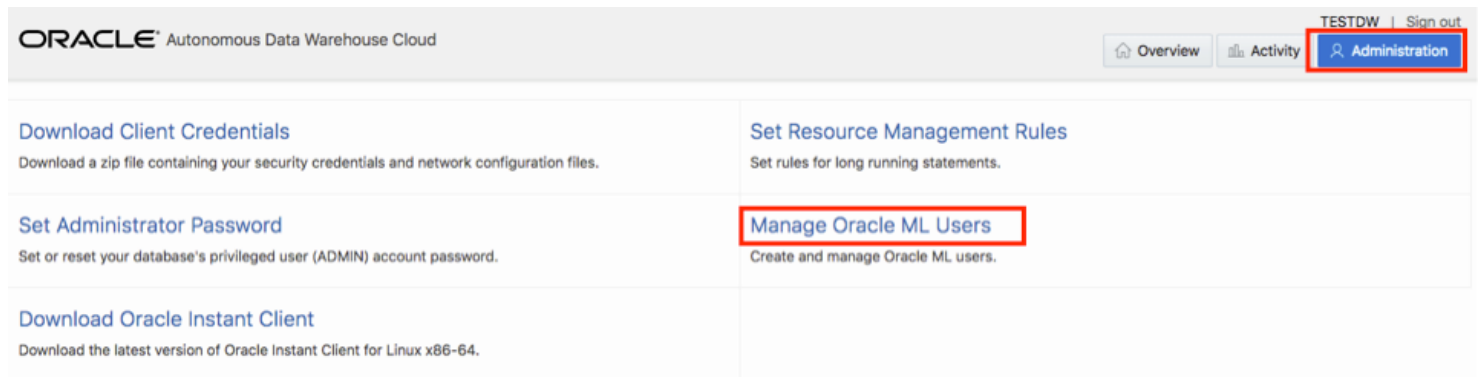
Lab Steps

Step 1: Creating OML Users

- Sign in to the ADW **Service Console** using the instructions from **Lab 2 - Step 2: Sign in to ADW Service Console**.



- Go to the **Administration** tab and click **Manage Oracle ML Users** to go to the OML user management page. This page will allow you to manage OML users.



- This will open a new tab within your browser that prompts you for a username and password.
- Enter **admin** as the username and use the password you specified when provisioning your ADWC instance.

Sign In

Tenant

Database

* Username

* Password

Note that you do not have to go to this page using the same steps every time, you can bookmark the URL and access it directly later.

ORACLE[®] Machine Learning User Administration

Users

Show All Users

User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		18/02/2018 08:29	Open

- Click the **Create** button to create a new OML user. Note that this will also create a new database user. This newly created user will be able to use the OML notebook application.

Note that you can also enter an email address to send an email confirmation to your user. For this workshop you can use your own personal email address.

- Enter the required information for this user, name the user as **omluser1**.
- Click the **Create** button, in the top-right corner of the page, to create the user.

Create User

* Username	<input type="text" value="omluser1"/>
First Name	<input type="text" value="OML"/>
Last Name	<input type="text" value="User1"/>
* Password	<input type="password" value="....."/>
* Confirm Password	<input type="password" value="....."/>
Email Address	<input type="text" value="oml.user1@oracle.com"/>
<input checked="" type="checkbox"/> Email account details to user	

- If you supplied a valid email address, a welcome email should arrive within a few minutes to your Inbox.
- Below is the email which each user receives welcoming them to the OML application. It includes a direct link to the OML application for that user which they can bookmark.

Dear OML User1,

A new Oracle Machine Learning account has been created for you.

Here are the details about your new account:

Account Details



Username: omluser1
Password: Welcome1!Welcome1
First Name: OML
Last Name: User1
Email: keith.laker@oracle.com

Click [here](#) to sign in and change your password as soon as possible.

/td>

- After you click **Create** you will see that user: **omluser1** is listed in the Users section.

The screenshot shows the Oracle Machine Learning User Administration interface. At the top, there is a navigation bar with the Oracle logo and the text 'Machine Learning User Administration'. Below this, a blue notification banner reads 'User Created'. Underneath the banner, the heading 'Users' is displayed. To the left of the heading are buttons for '+ Create', 'X Delete', and a checkbox for 'Show All Users'. To the right is a search box labeled 'Search...'. Below these elements is a table with the following columns: 'User Name', 'Full Name', 'Role', 'Email', 'Created On', and 'Status'. The table contains two rows: one for 'ADMIN' (System Administrator, 2/18/18 8:29 AM, Open) and one for 'OMLUSER1' (OML User1, Developer, keith.laker@oracle.com, 3/19/18 1:50 PM, Open).

User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		2/18/18 8:29 AM	Open
OMLUSER1	OML User1	Developer	keith.laker@oracle.com	3/19/18 1:50 PM	Open

- Using the same steps, create another user named **omluser2**.

User Created

Users

[+ Create](#) [X Delete](#) Show All Users

User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		2/18/18 8:29 AM	Open
OMLUSER1	OML User1	Developer	keith.laker@oracle.com	3/19/18 1:50 PM	Open
OMLUSER2	OML User2	Developer	keith.laker@oracle.com	3/19/18 1:58 PM	Open

- You will use these two users later in this workshop.

Exploring the OML Home Page

Sign-on to OML

- Using the link from your welcome email, from Oracle Global Accounts, you can now sign-in to OML. Copy and paste the application link from the email into your browser and sign-in to OML.
- If you have not specified an email address you can click the **Home** icon on the top right of Oracle Machine Learning **User Administration** page to go to OML home page.

User Created


Users


[+ Create](#) [X Delete](#) Show All Users

User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		2/18/18 8:29 AM	Open
OMLUSER1	OML User1	Developer		3/20/18 5:13 PM	Open
OMLUSER2	OML User2	Developer		3/20/18 5:14 PM	Open

- Use your new user account **omluser1**:

Sign In

Tenant 

Database 

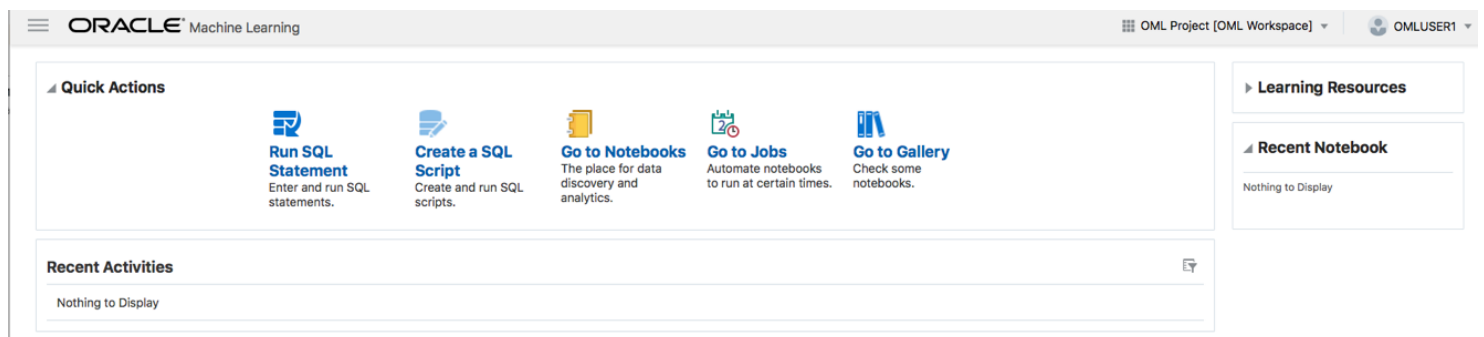
* Username

* Password

- Once you have successfully signed-in to OML the application home page will be displayed.

OML Home Page

- The grey menu bar at the top of the screen provides links to the main OML menus for the application (left corner) and the workspace/project and user maintenance on the right-hand side.



The screenshot shows the Oracle Machine Learning home page. At the top, there is a grey header bar with the Oracle Machine Learning logo on the left and the current workspace 'OML Project [OML Workspace]' and user 'OMLUSER1' on the right. Below the header, the main content area is divided into several panels. On the left, there is a 'Quick Actions' panel with five icons: 'Run SQL Statement' (with a sub-description 'Enter and run SQL statements.'), 'Create a SQL Script' (with a sub-description 'Create and run SQL scripts.'), 'Go to Notebooks' (with a sub-description 'The place for data discovery and analytics.'), 'Go to Jobs' (with a sub-description 'Automate notebooks to run at certain times.'), and 'Go to Gallery' (with a sub-description 'Check some notebooks.'). To the right of the 'Quick Actions' panel, there are two more panels: 'Learning Resources' and 'Recent Notebook', both showing 'Nothing to Display'. At the bottom left, there is a 'Recent Activities' panel, also showing 'Nothing to Display'.

- On the home page the main focus is the “**Quick Actions**” panel.
- The main icons in this panel provide shortcuts to the main OML pages for running queries and managing your saved queries.



Run SQL Statement

Enter and run SQL statements.



Create a SQL Script

Create and run SQL scripts.



Go to Notebooks

The place for data discovery and analytics.



Go to Jobs

Automate notebooks to run at certain times.



Go to Gallery

Check some notebooks.

- All your work is automatically saved – i.e. there is no “Save” button when you are writing scripts and/or queries.

Key concepts

What is a Workspace? A workspace is an area where you can store your projects. Each workspace can be shared with other users so they can collaborate with you. For collaborating with other users, you can provide different levels of permission such as Viewer, Developer and Manager – these will be covered in more detail later in this workshop. You can create multiple workspaces.

What is a Project? A project is a container for organizing your notebooks. You can create multiple projects.

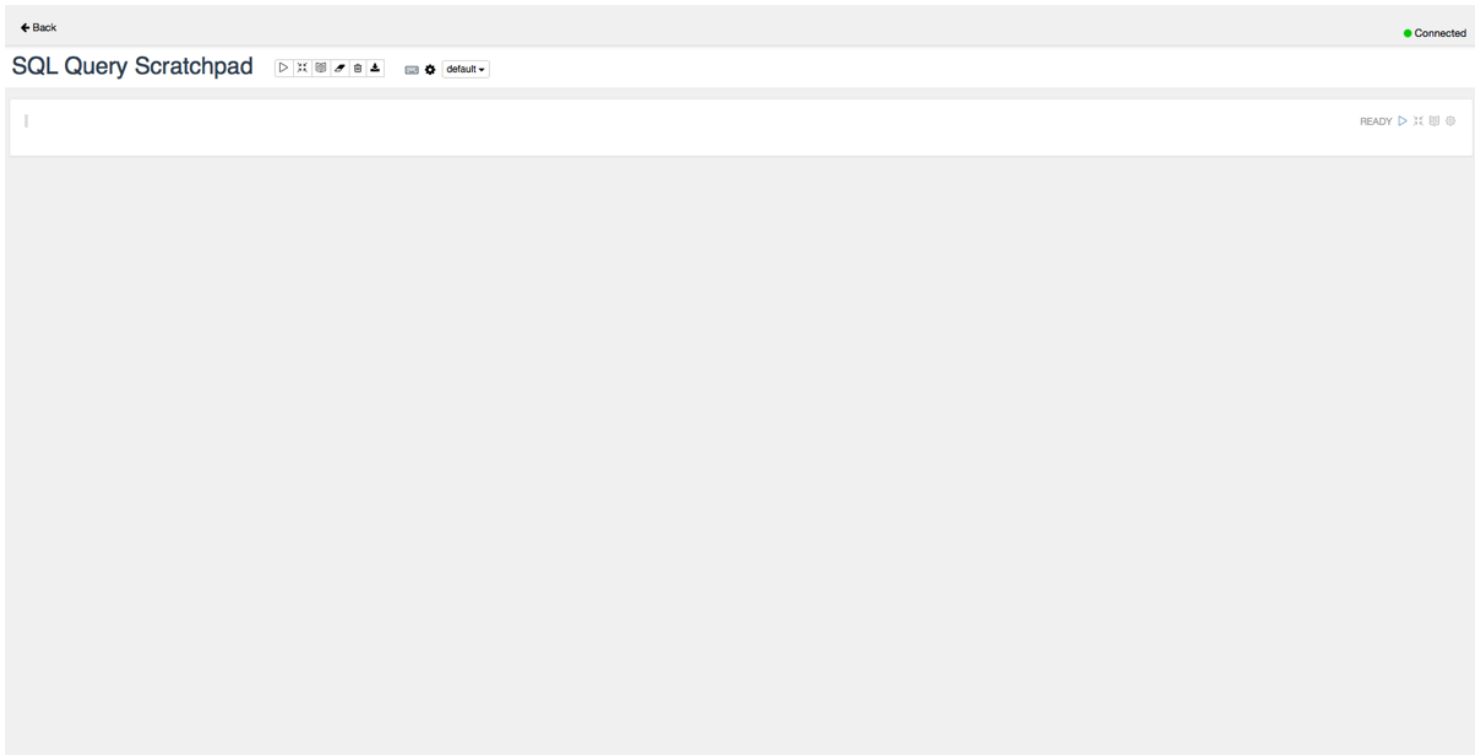
What is a Notebook? A notebook is a web-based interface for building reports and dashboards using a series of pre-built data visualizations which can then be shared with other OML users. Each notebook can contain one or SQL queries and/or SQL scripts. Additional non-query information can be displayed using special markdown tags (an example of these tags will be shown later).

Running a SQL Statement

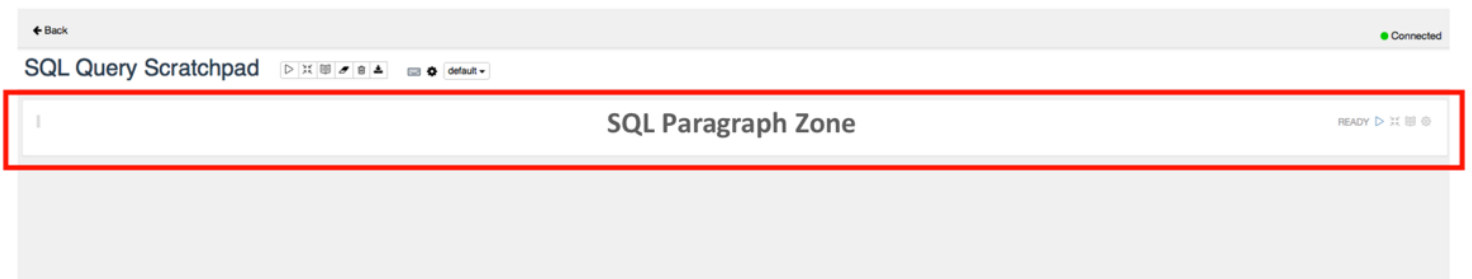
- From the home page click on the **Run SQL Statement** link in the Quick Actions panel to open a new SQL query scratchpad.

The screenshot shows the Oracle Machine Learning interface. At the top, there is a header with the Oracle logo and 'Machine Learning' text. On the right, there are two dropdown menus: 'OML Project [OML Workspace]' and 'OMLUSER1'. Below the header, there is a 'Quick Actions' panel with five icons and their corresponding text: 'Run SQL Statement' (Enter and run SQL statements), 'Create a SQL Script' (Create and run SQL scripts), 'Go to Notebooks' (The place for data discovery and analytics), 'Go to Jobs' (Automate notebooks to run at certain times), and 'Go to Gallery' (Check some notebooks). To the right of the Quick Actions panel, there are two more panels: 'Learning Resources' and 'Recent Notebook' (Nothing to Display). Below the Quick Actions panel, there is a 'Recent Activities' panel (Nothing to Display).

- The following screen should appear:



- The white panel below the main title (SQL Query Scratchpad – *this name is automatically generated*) is an area known as “paragraph”. Within a scratchpad you can have multiple paragraphs. Each paragraph can contain one SQL statement or a SQL script.



- You need to enter the SQL in the **SQL Paragraph Zone**. Click on the below link to get the SQL and paste it to the SQL Paragraph zone.
 - [OML SQL](#)
- Here is the SQL (do not copy/paste the below SQL as it may contain line breaks, use the above link instead)

```

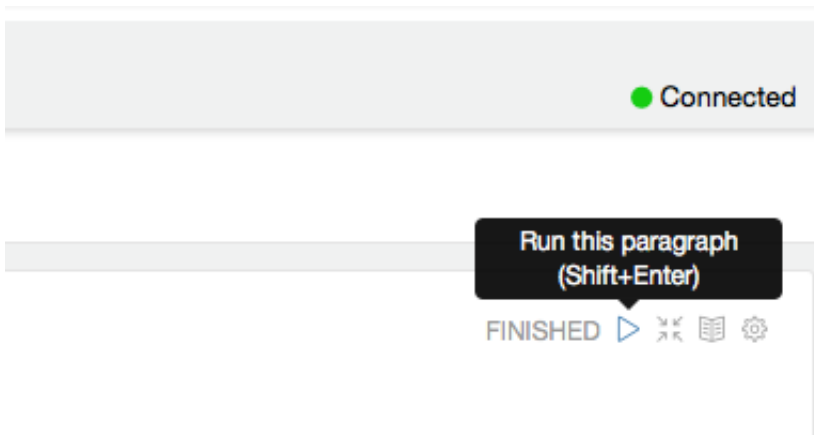
SELECT
p.prod_category_desc,
t.calendar_year as year,
t.calendar_month_desc as Month,
TRUNC(SUM(amount_sold)) as revenue,
TRUNC(AVG(SUM(amount_sold)) over (PARTITION BY t.calendar_year ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 2 PRECEDING)) as avg_3M_revenue,
TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 5 PRECEDING)) as avg_6M_revenue,
TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 11 PRECEDING)) as avg_12M_revenue
FROM sh.sales s, sh.times t, sh.products p
WHERE s.time_id = t.time_id
AND s.prod_id = p.prod_id
AND prod_category_desc = 'Electronics'
GROUP BY p.prod_category_desc, t.calendar_year, calendar_month_desc
ORDER BY p.prod_category_desc, t.calendar_year, calendar_month_desc;

```

- Your screen should now look like this:



- Press the **Run this paragraph** icon to execute the SQL statement.



- The results will be displayed in a tabular format:

← Back Connected

SQL Query Scratchpad

```

SELECT
  p.prod_category_desc,
  t.calendar_year as year,
  t.calendar_month_desc as Month,
  TRUNC(SUM(amount_sold)) as revenue,
  TRUNC(AVG(SUM(amount_sold)) over (PARTITION BY t.calendar_year ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 2 PRECEDING)) as avg_3m_revenue,
  TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 5 PRECEDING)) as avg_5m_revenue,
  TRUNC(AVG(SUM(amount_sold)) over (ORDER BY p.prod_category_desc, t.calendar_month_desc ROWS 11 PRECEDING)) as avg_12m_revenue
FROM sh.sales s, sh.times t, sh.products p
WHERE s.time_id = t.time_id
AND s.prod_id = p.prod_id
AND prod_category_desc = 'Electronics'
GROUP BY p.prod_category_desc, t.calendar_year, calendar_month_desc
ORDER BY p.prod_category_desc, t.calendar_year, calendar_month_desc;

```

FINISHED

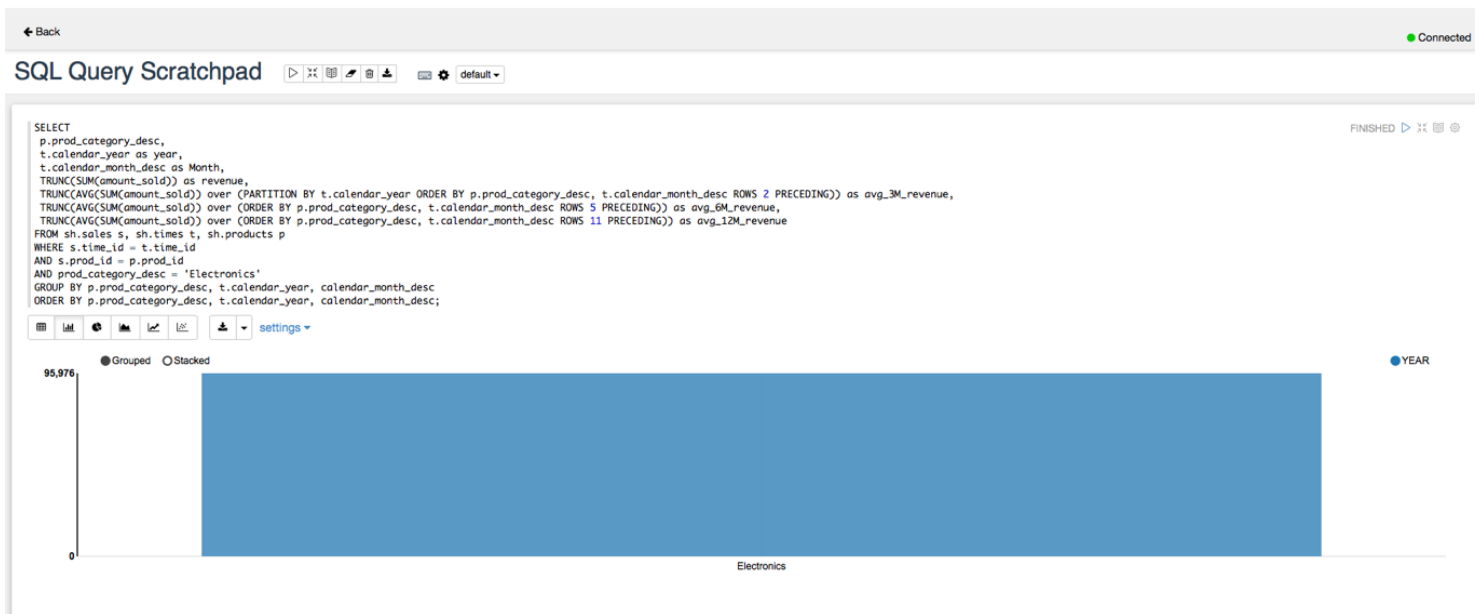
PROD_CATEGORY_DESC	YEAR	MONTH	REVENUE	AVG_3M_REVENUE	AVG_6M_REVENUE	AVG_12M_REVENUE
Electronics	1,998	1998-01	151,647	151,647	151,647	151,647
Electronics	1,998	1998-02	183,034	167,341	167,341	167,341
Electronics	1,998	1998-03	131,373	155,351	155,351	155,351
Electronics	1,998	1998-04	168,357	160,922	158,603	158,603
Electronics	1,998	1998-05	133,325	144,352	153,547	153,547
Electronics	1,998	1998-06	177,123	159,602	157,477	157,477
Electronics	1,998	1998-07	157,758	156,069	158,495	157,517
Electronics	1,998	1998-08	134,857	156,513	150,432	154,659
Electronics	1,998	1998-09	151,299	147,905	153,753	154,286

Changing the Report Type

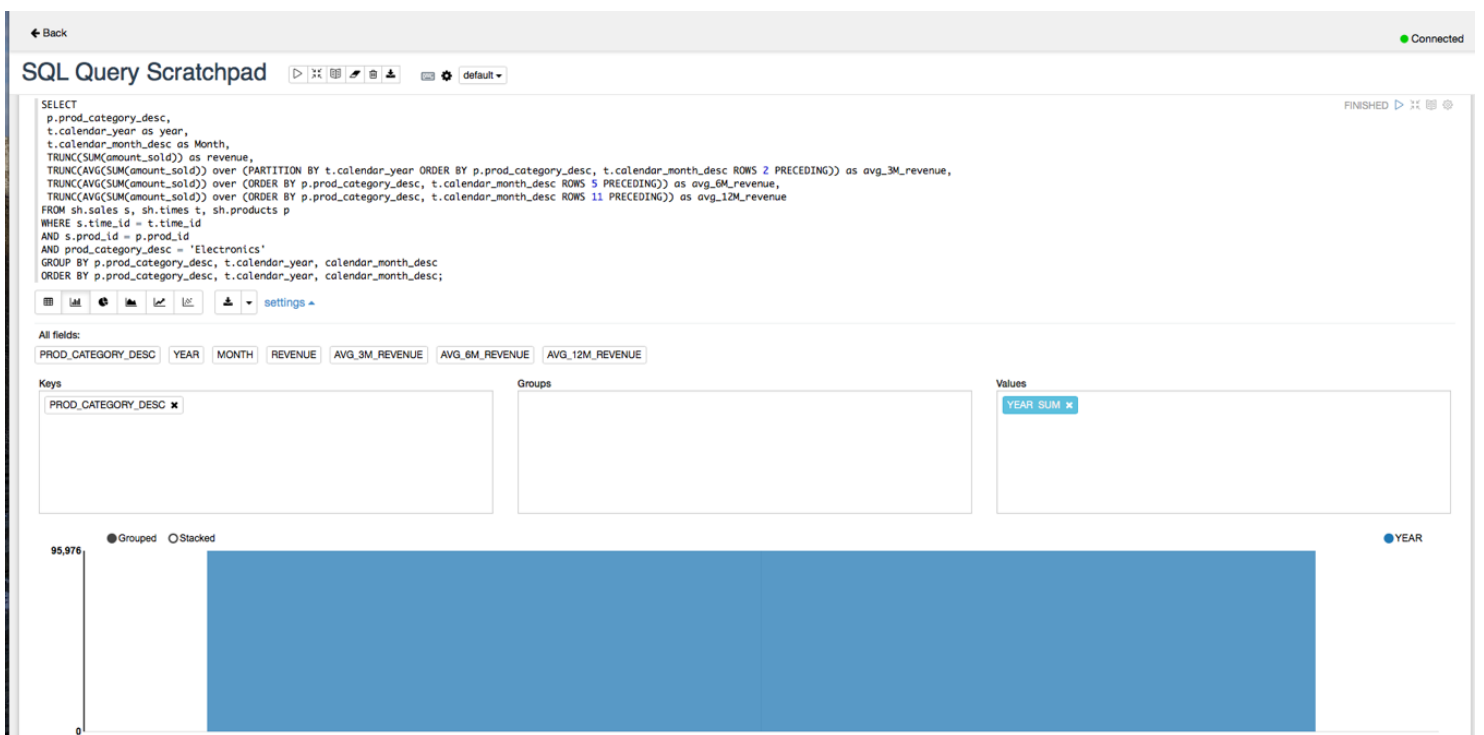
- Using the report menu bar you can change the table to a graph and/or export the result set to a CSV or TSV file.



- When you change the report type to one of the graphs, then a **Settings** link will appear to the right of the menu which allows you to control the layout of columns within the graph.
- Click on the bar graph icon to change the output to a bar graph (see below)



- Click on the **Settings** link to unfold the settings panel for the graph.

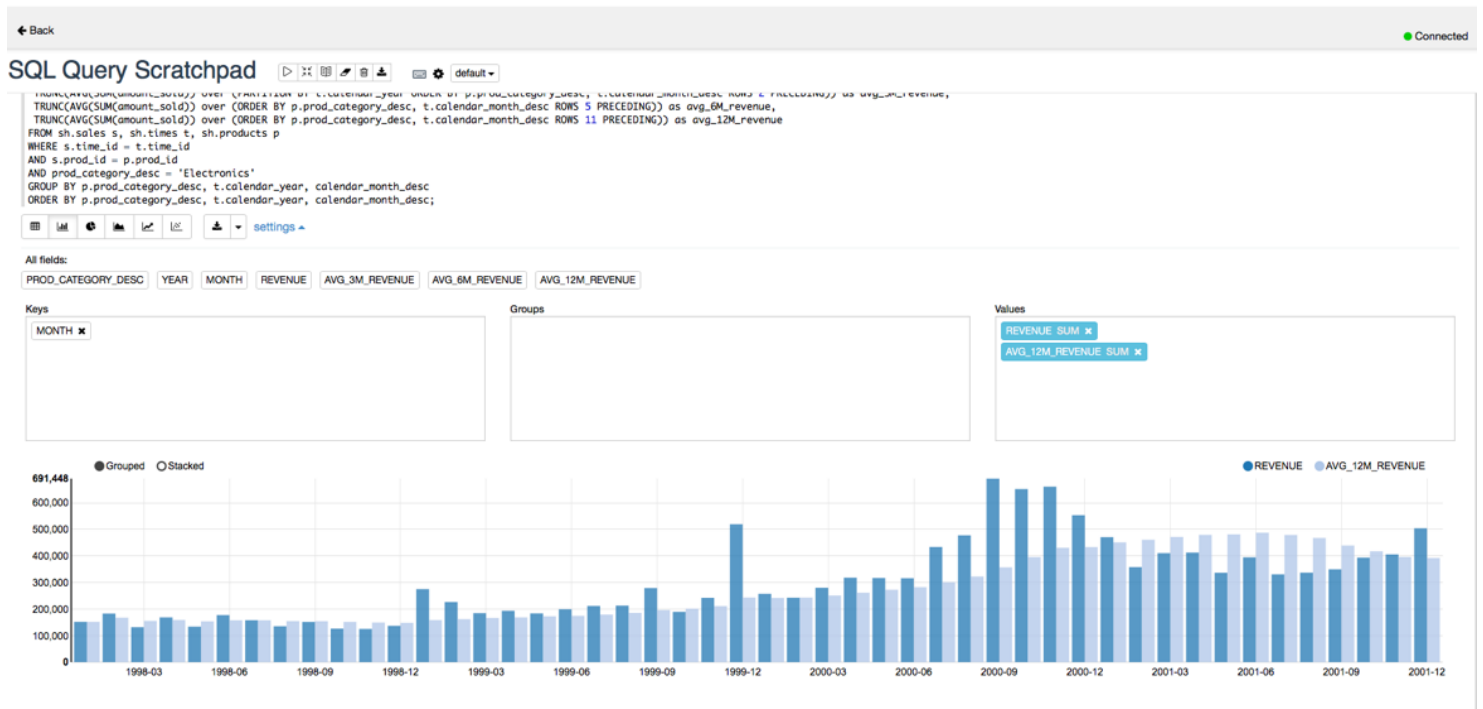


- To add a column to one of the **Keys**, **Groups** of **Values** panels just drag and drop the column name into the required panel.
- To remove a column from the Keys, Groups of Values panel just click on the “x” next to the column name displayed in the relevant panel.

Changing the layout of the graph

With the graph settings panel visible:

- Remove all columns from the both the Keys and Values panels.
- Drag and drop **MONTH** into the **Keys** panel
- Drag and drop **REVENUE** into the **Values** panel
- Drag and drop **AVG 12M REVENUE** into the **Values** panel
- The report should now look like the one shown below.



Tidying up the report:

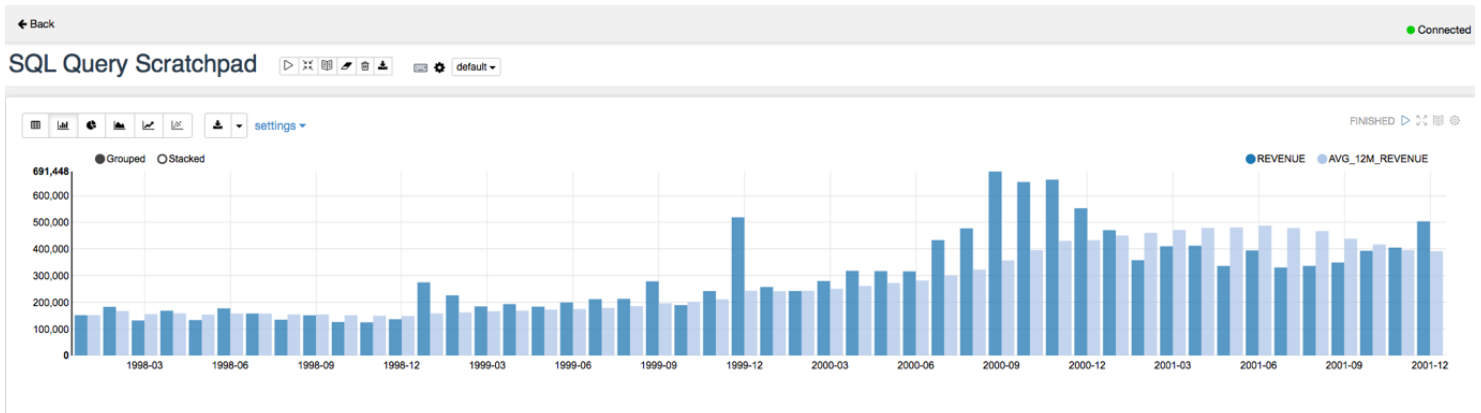
- Click on the **Settings** link to hide the layout controls.
- Click on the **“Hide editor”** button which is to the - right of the **“Run this paragraph”** button.

● Connected

Hide editor

FINISHED ▶ ⌵ ⌶ ⚙

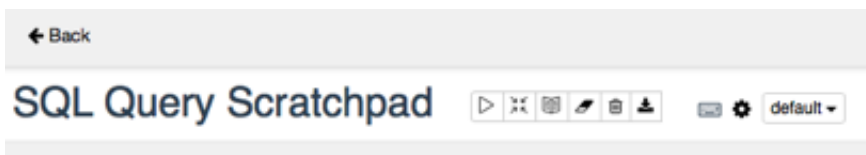
- Now only the output is visible.



Saving the Scratchpad as a New Notebook

The SQL Scratchpad in the previous section is simply a default type notebook with a system generated name.

But we can change the name of the scratchpad we have just created **SQL Query Scratchpad**.



- Click on the **Back** link in the top left corner of the Scratchpad window to return to the OML home page.
- Notice that in the **Recent Activities** panel there is a potted history of what has happened to your SQL scratchpad **Notebook**.

Quick Actions

- Run SQL Statement**
Enter and run SQL statements.
- Create a SQL Script**
Create and run SQL scripts.
- Go to Notebooks**
The place for data discovery and analytics.
- Go to Jobs**
Automate notebooks to run at certain times.
- Go to Gallery**
Check some notebooks.

Learning Resources

Recent Notebook

Nothing to Display

Recent Activities

today

OML User1 created SQL Query Scratchpad notebook in OML Project [OML Workspace]
3/20/18 8:50 PM

- Click on **Goto Notebooks** in the **Quick Actions** panel.

Quick Actions

- Run SQL Statement**
Enter and run SQL statements.
- Create a SQL Script**
Create and run SQL scripts.
- Go to Notebooks**
The place for data discovery and analytics.
- Go to Jobs**
Automate notebooks to run at certain times.
- Go to Gallery**
Check some notebooks.

Learning Resources

Recent Notebook

SQL Query Scratchpad

- The Notebooks page will be displayed:

Notebooks

Edit Create Duplicate Save as Template Delete Import Version

Name	Comment
SQL Query Scratchpad	Scratch pad notebook to run SQL statements.

- Let's rename our SQL Scratchpad notebook to something more informative.
- Click on text in the **“comments”** column to select the scratchpad so we can rename it.
- After you click, the **SQL Query Scratchpad** will become selected and the menu buttons above will activate.

Notebooks

Edit Create Duplicate Save as Template Delete Import Version

Name	▲ Comment
SQL Query Scratchpad	Scratch pad notebook to run SQL statements.

- Click on the **Edit** button to pop-up the settings dialog for this notebook and enter the information as shown in the image below (*note that the connection information is read-only because this is managed by Autonomous Data Warehouse Cloud*):

Edit Notebook ✕

Name *

Comment

Connection

- Click **OK** to save your notebook. You will see that your SQL Query Scratchpad notebook is now renamed to the new name you specified.

Notebooks

✎ Edit + Create 📄 Duplicate 📄 Save as Template ✕ Delete 📄 Import 📄 Version Search... 🔍				
Name	▲ Comment	Last Update	Updated By	Connection Group
Sales Analysis Over Time	Sales analysis bar chart	3/20/18 8:55 PM	OMLUSER1	Global

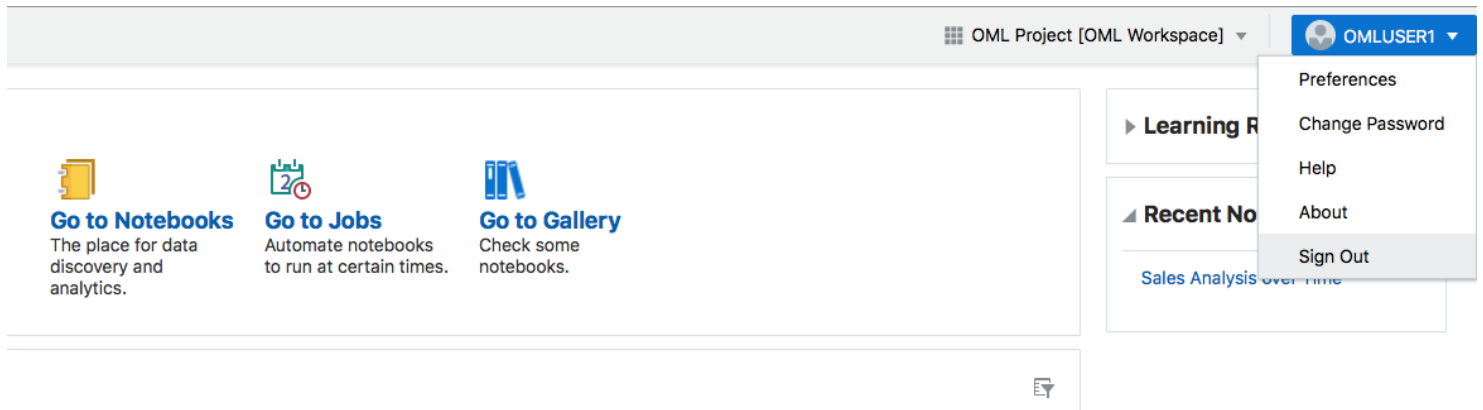
Sharing notebooks

By default, when you create a notebook it's only visible to you.

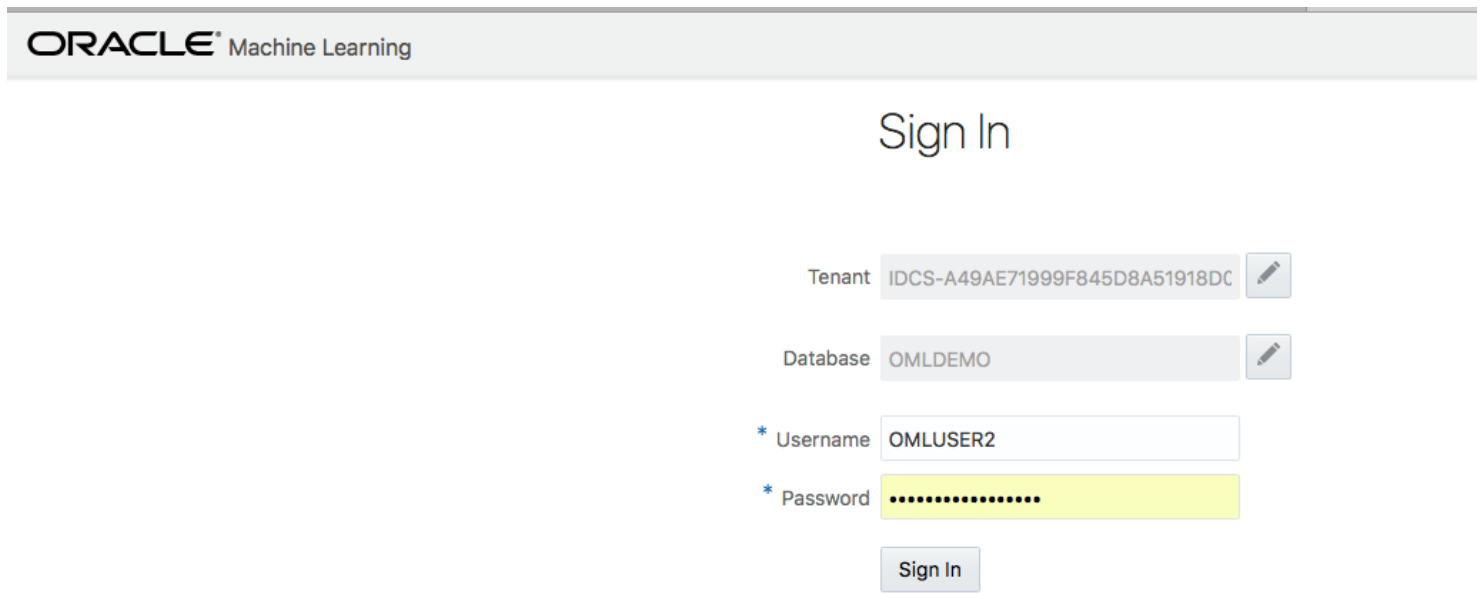
To make it available to other users you need to share the workspace containing the notebook. You can create new workspaces and projects to organize your notebooks for ease of use and to share with other users.

To demonstrate the sharing process let's begin by logging in to OML as our second OML (**omluser2**) user and checking if any notebooks are available.

- Click on your user name in the top right corner (**omluser1**) and select **Sign Out**.



- Now sign-in as OML user **omluser2** using the password you entered at the beginning of this workshop:



- Notice that you have no activity listed in the **Recent Activities** panel on your OML home page and you don't have any notebooks.

ORACLE Machine Learning

OML Project [OML Workspace] OMLUSER2

Quick Actions

- Run SQL Statement**
Enter and run SQL statements.
- Create a SQL Script**
Create and run SQL scripts.
- Go to Notebooks**
The place for data discovery and analytics.
- Go to Jobs**
Automate notebooks to run at certain times.
- Go to Gallery**
Check some notebooks.

Recent Activities
Nothing to Display

Hint – click on the **Go to Notebooks** link in the **Quick Actions** panel:

Notebooks

Edit Create Duplicate Save as Template Delete Import Version

Name	Comment	Last Update	Updated By	Connection Group
No data to display.				

- Repeat the previous steps to logout of OML and sign into OML as **omluser1**.

Changing Workspace Permissions

From the OML home page:

- Click on link **OML Project (OML Workspace)** link in the top right corner on the OML home page to display the workspace-project menu.
- Select “**Workspace Permissions**”

OML Project [OML Workspace] OMLUSER1

- Recent Projects
- Select Project...
- New Project...
- Manage Workspaces...
- Workspace Permissions...**

- The permissions dialog box will appear – see below.
- In the dialog box next to the “**Add Permissions**” text type “**OMLUSER2**” (use uppercase).
- Set the permission type to “**Viewer**” (this means read-only access to the workspace, project and notebook).

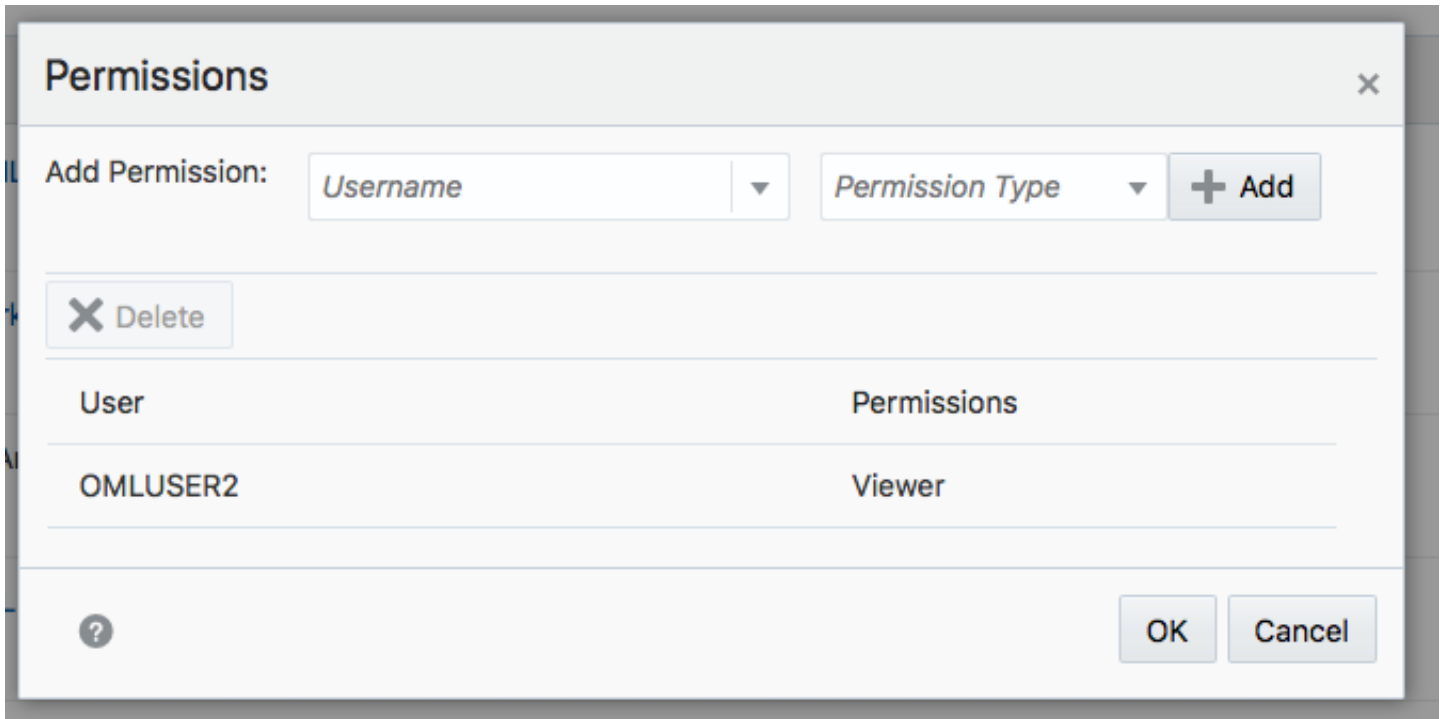
Note that:

A **Developer** would have read-only access to the workspace, project but could add new notebooks, update and delete existing notebooks and schedule jobs to refresh a notebook.

A **Manager** would have read-only access to the workspace, can create, update and delete projects, add new notebooks, update and delete existing notebooks and schedule jobs to refresh a notebook.

The screenshot shows a 'Permissions' dialog box. At the top, the title 'Permissions' is displayed with a close button (X). Below the title bar, there is a section 'Add Permission:' with a dropdown menu showing 'OMLUSER2' and a '+ Add' button. A 'Permission Type' dropdown menu is open, showing options: 'Permission Type', 'Manager', 'Developer', and 'Viewer'. Below this is a 'Delete' button with an 'X' icon. A table with columns 'User' and 'Permissions' is shown, with the text 'No data to display.' below it. At the bottom, there is a help icon (question mark) and 'OK' and 'Cancel' buttons.

- Click the **Add** button to add the user **omluser2** as a read-only viewer of the workspace.
- Your form should look like this:

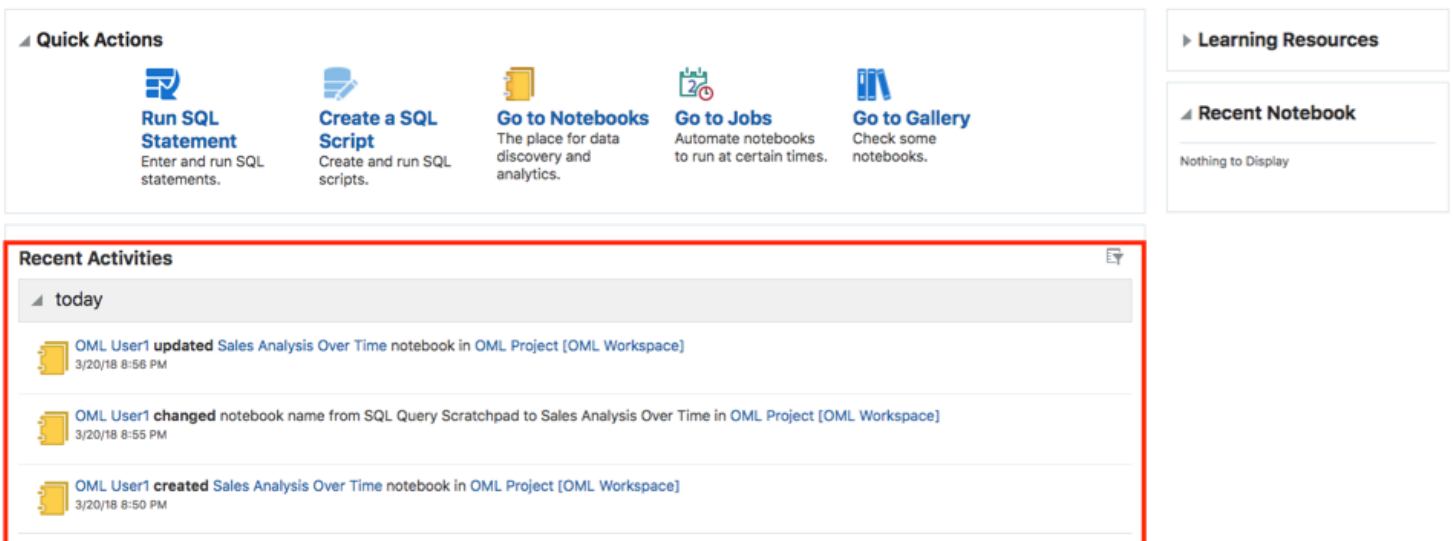


- Finally, click the **OK** button.

Accessing shared notebooks

Now repeat the process you followed at the start of this section and sign-out of OML and sign-in to OML again as user **omluser2**.

First thing to note is that the **Recent Activities** panel below the **Quick Links** panel now shows all the changes user **omluser1** made within the workspace OML-Workspace.



- As user **omluser2** you can now run the **Sales Analysis Over Time** notebook by clicking on the blue-

linked text in the **Recent Activities** panel (*note that your recent activity will be logged under the banner labelled "Today"*).

Recent Activities

yesterday



OML User1 updated Sales Analysis over Time notebook in OML Project [OML Workspace]

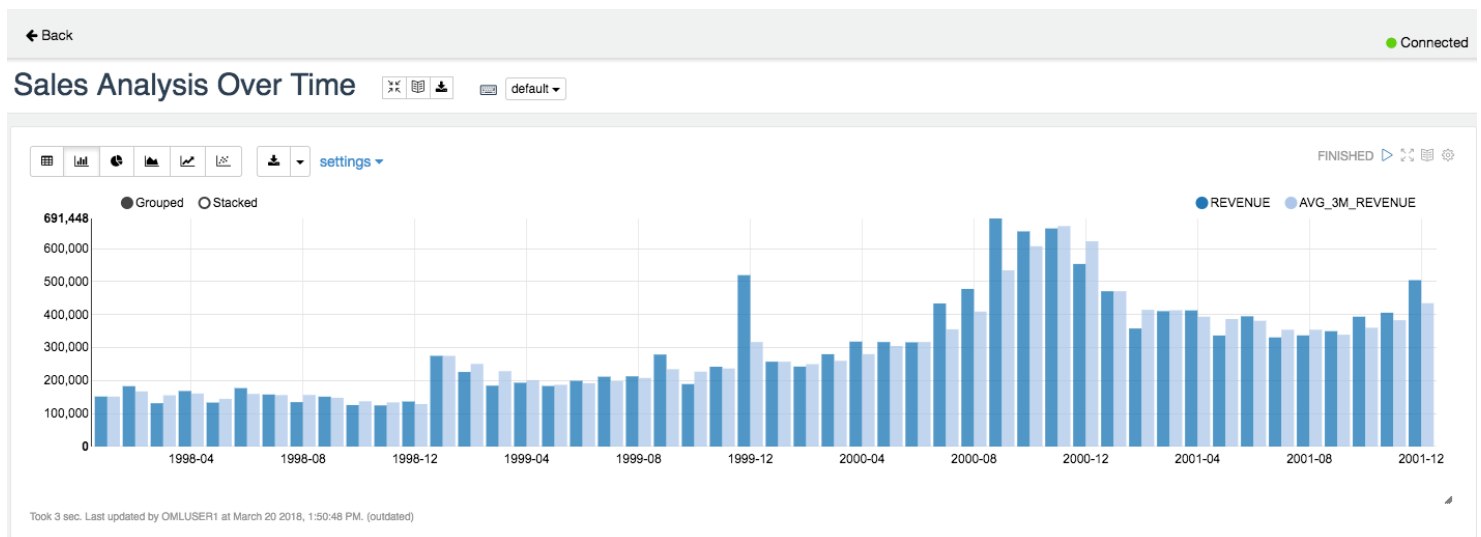
19/03/2018 16:38



OML User1 created Latest Sales Report notebook in OML Project [OML Workspace]

19/03/2018 15:44

- The notebook will now open:



Creating and running SQL scripts

- Log out from user OMLUSER2 and log in as OMLUSER1.
- The **“Run SQL Statement”** link on the home page allows you to run a single query in a paragraph. To be able to run scripts you can use the **Create a SQL Script** link on the home page.

Getting Started with Scripts

On the OML home page click **“Create a SQL Script”** link within the **“Quick Actions”** panel.

▲ Quick Actions



Run SQL Statement

Enter and run SQL statements.



Create a SQL Script

Create and run SQL scripts.

- A new **SQL scratchpad** will be created with the **%script** identifier already selected, this identifier allows you to run multiple SQL statements.

← Back

SQL Script Scratchpad



```
%script
```

Notice that the script paragraph does not have any menus to control the display and formatting of the output. You can, however, use SQL SET commands to control how data is formatted for display.

Creating and running a SQL script

In this section we are going to use a script from a SQL pattern matching tutorial on the free Oracle livesql.oracle.com site:

https://livesql.oracle.com/apex/livesql/file/tutorial_EWB8G5JBSHAGM9FB2GL4V5CAQ.html

The PL/SQL script shows how to use the SQL pattern matching **MATCH_RECOGNIZE** feature for sessionization analysis based on **JSON** web log files.

- You need to enter a PL/SQL script. Click on the below link to get the script and paste it to the **%script** paragraph:
 - [OML Script](#)
- Here is the PL/SQL script (Do not copy/paste from here as the lines will be broken - use the above link instead)

```

%script
/*
First step is to create the JSON table that will hold our data from our session log file. The log will provide the details of the time and account/user id.

Oracle Autonomous Data Warehouse Cloud supports storing JSON documents inside the database. Use following code to create a table to store the transaction log which is in JSON format
*/
BEGIN
    EXECUTE IMMEDIATE 'drop table json_sessionization';
    EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'CREATE TABLE json_sessionization(session_doc CLOB, CONSTRAINT "VALID_JSON" CHECK (session_doc IS JSON) ENABLE)';
END;
/
/*
Next step is to add some data to our JSON table using the normal JSON notation of key-value pairs.
*/

BEGIN
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"1","user_id":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"2","user_id":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"11","user_id":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"12","user_id":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"22","user_id":"Sam"}'')';

```

```

EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"23","user_id
":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"32","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"34","user_id
":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"43","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"44","user_id
":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"47","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"48","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"53","user_id
":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"59","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"60","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"63","user_id
":"Mary"}'')';
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{"time_id":"68","user_id
":"Sam"}'')';
EXECUTE IMMEDIATE 'COMMIT';
END;

```

/

/*

Using the new JSON SQL notation we can query our data using a simple SELECT statement
*/

```

SELECT
  TO_NUMBER(j.session_doc.time_id) as time_id,
  j.session_doc.user_id as user_id
FROM json_sessionization j;

```

/*

Using MATCH_RECOGNIZE clause to create a sessionization report. The aim is to count the number of events within each session and calculate the duration of each session.

How do we do that? We can use some of the other built-in measures such as FIRST() and LAST() to extract values from our resultset and we can calculate new values such as the duration of a session.

In our code we will compute some new measures:

count(*) returns the number of events within a session

first(tstamp) returns the start time of each session

last(tstamp) returns the end time of each session

last(tstamp) - first(tstamp) calculates the duration of each session

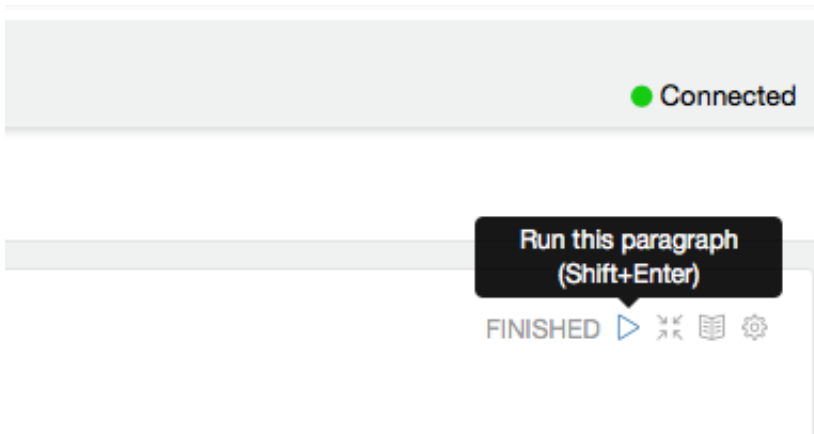
*/

```
SELECT
  userid,
  session_id,
  no_of_events,
  start_time,
  end_time,
  session_duration
FROM (SELECT
      TO_NUMBER(j.session_doc.time_id) as time_id,
      j.session_doc.user_id as userid
    FROM json_sessionization j)
MATCH_RECOGNIZE(
  PARTITION BY userid ORDER BY time_id
  MEASURES match_number() as session_id,
            COUNT(*) as no_of_events,
            FIRST(b.time_id) start_time,
            LAST(s.time_id) end_time,
            LAST(s.time_id) - FIRST(b.time_id) session_duration
  ONE ROW PER MATCH
  PATTERN (b s+)
  DEFINE
    s as (time_id - PREV(time_id) <= 10)
);
```

- After pasting the above code into the script paragraph it should look something like this:

```
← Back ● Connected  
SQL Script Scratchpad ▶ ⌘ ⌚ ⌕ ⚙ default  
  
/*script  
/*  
First step is to create the JSON table that will hold our data from our session log file. The log will provide the details of the time and account/user id.  
  
Oracle Autonomous Data Warehouse Cloud supports storing JSON documents inside the database. Use following code to create a table to store the transaction log which is in JSON format  
*/  
BEGIN  
  EXECUTE IMMEDIATE 'drop table json_sessionization';  
  EXCEPTION  
  WHEN OTHERS THEN NULL;  
END;  
/  
  
BEGIN  
  EXECUTE IMMEDIATE 'CREATE TABLE json_sessionization(session_doc CLOB, CONSTRAINT "VALID_JSON" CHECK (session_doc IS JSON) ENABLE)';  
END;  
/  
/*  
Next step is to add some data to our JSON table using the normal JSON notation of key-value pairs.  
*/  
  
BEGIN  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':1,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':2,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':11,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':12,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':22,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':23,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':32,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':34,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':43,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':44,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':47,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':48,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':53,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':59,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':60,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':63,'user_id':''Mary''})';  
EXECUTE IMMEDIATE 'INSERT INTO json_sessionization VALUES (''{''time_id'':68,'user_id':''Sam''})';  
EXECUTE IMMEDIATE 'COMMIT';  
END;  
/  
/*  
Using the new JSON SQL notation we can query our data using a simple SELECT statement  
*/  
SELECT
```

- You can then run the script/paragraph and the output will appear below the code that makes up the script.



- The result should look something like this:

SQL Script Scratchpad

▶ ⌂ 🗑️ 📄 📄 default ▾

```

SELECT
  userid,
  session_id,
  no_of_events,
  start_time,
  end_time,
  session_duration
FROM (SELECT
  TO_NUMBER(j.session_doc.time_id) as time_id,
  j.session_doc.user_id as userid
  FROM json_sessionization j)
MATCH_RECOGNIZE(
  PARTITION BY userid ORDER BY time_id
  MEASURES match_number() as session_id,
  COUNT(*) as no_of_events,
  FIRST(b.time_id) start_time,
  LAST(s.time_id) end_time,
  LAST(s.time_id) - FIRST(b.time_id) session_duration
  ONE ROW PER MATCH
  PATTERN (b s+)
  DEFINE
  s as (time_id - PREV(time_id) <= 10)
);

```

```

44 Mary
47 Sam
48 Sam
53 Mary
59 Sam

```

```

TIME_ID USER_ID
60 Sam
63 Mary
68 Sam

```

17 rows selected.

USERID	SESSION_ID	NO_OF_EVENTS	START_TIME	END_TIME	SESSION_DURATION
Mary	1	2	1	11	10
Mary	2	4	34	63	29
Sam	1	4	2	32	30
Sam	2	3	43	48	5
Sam	3	3	59	68	9

