**PROJECT BLOBs and CLOBs - Input of Graphic Images into a Table**

**Objective** To create tables with SQL containing BLOB and CLOB datatypes.

***PROJECT DESCRIPTION***

Create tables, populate the tables and then perform joins that will yield records

containing information from the tables, according to the instructions rendered below.

***Information about this Project***

This lab exercise entails creating SQL scripts.

***Steps to Complete this Project***

**STEP 1**  **Open Oracle SQL Developer or Equivalent SQL IDE on Your Computer**

Launch Oracle SQL Developer or equivalent SQL application IDE.

**STEP 2**  **Create and Populate the Table(s)**

Create the SQL table(s) given below.

**[ Create Tables ]**

**[ The planets table, with comments on 2 fields ]**

**DROP TABLE planets;**

**CREATE TABLE planets**

**(pl\_ID number,**

**pl\_name varchar2(20),**

**pl\_diameter number,**

**pl\_class varchar2(20),**

**pl\_mass number, -- mass in tons x 10^21**

**pl\_dist\_fr\_sun number, -- distance x 10^6 miles**

**pl\_num\_moons number,**

**pl\_rings char(3));**

**[ The planets2 table ]**

**DROP TABLE planets2;**

**CREATE TABLE planets2 (pl\_ID number, picture BLOB);**

**[ The planets3 table ]**

**DROP TABLE planets3;**

**CREATE TABLE planets3 (pl\_id number, essay CLOB);**

**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**[The planets4 table]**

**DROP TABLE planets4;**

**CREATE TABLE planets4**

**(pl\_id number, music BLOB);**

**[ Populate the planets Table ]**

Add these records to the above table named planets.

**INSERT INTO planets**

**Values (1, 'Mercury', 4879.4,'Terrestrial',0.364,36,0,'No');**

**INSERT INTO planets**

**Values (2, 'Venus', 12103.6,'Terrestrial',5.37,67.2,0,'No');**

**INSERT INTO planets**

**Values (3, 'Earth', 12742,'Terrestrial', 6.58, 93,1,'No');**

**INSERT INTO planets**

**Values (4, 'Mars', 6779,'Terrestrial',0.708, 141.6,2,'No');**

**INSERT INTO planets**

**Values (5, 'Jupiter', 139822,'Jovian',2093,483.8,67, 'Yes');**

**INSERT INTO planets**

**Values (6, 'Saturn', 116464,'Jovian',627,890.8,62,'Yes');**

**INSERT INTO planets**

**Values (7, 'Uranus', 50724,'Jovian',95.7,1784.8,27,'Yes');**

**INSERT INTO planets**

**Values (8, 'Neptune',49244,'Jovian',113,2793.1,14,'Yes');**

**INSERT INTO planets**

**Values (9, 'Pluto', 1400 , 'Terrestrial',0.0144,3647.2,5,'No');**

**INSERT INTO planets**

**Values (10, 'PlanetX',99999 ,'Unknown',99999,9999 ,99,'No');**

**commit;**

**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**[ Populate the planets2 Table ]**

**INSERT INTO planets2 values(1, EMPTY\_BLOB());**

**INSERT INTO planets2 values(2, EMPTY\_BLOB());**

**INSERT INTO planets2 values(3, EMPTY\_BLOB());**

**INSERT INTO planets2 values(4, EMPTY\_BLOB());**

**INSERT INTO planets2 values(5, EMPTY\_BLOB());**

**INSERT INTO planets2 values(6, EMPTY\_BLOB());**

**INSERT INTO planets2 values(7, EMPTY\_BLOB());**

**INSERT INTO planets2 values(8, EMPTY\_BLOB());**

**INSERT INTO planets2 values(9, EMPTY\_BLOB());**

**INSERT INTO planets2 values(10, EMPTY\_BLOB());**

**commit;**

**[ Populate the planets3 Table ]**

**INSERT INTO planets3 values(1, EMPTY\_CLOB());**

**INSERT INTO planets3 values(2, EMPTY\_CLOB());**

**INSERT INTO planets3 values(3, EMPTY\_CLOB());**

**INSERT INTO planets3 values(4, EMPTY\_CLOB());**

**INSERT INTO planets3 values(5, EMPTY\_CLOB());**

**INSERT INTO planets3 values(6, EMPTY\_CLOB());**

**INSERT INTO planets3 values(7, EMPTY\_CLOB());**

**INSERT INTO planets3 values(8, EMPTY\_CLOB());**

**INSERT INTO planets3 values(9, EMPTY\_CLOB());**

**INSERT INTO planets3 values(10,Empty\_CLOB());**

**commit;**

**[ Populate the planets4 Table ]**

**INSERT INTO planets4 values(1, EMPTY\_BLOB());**

**INSERT INTO planets4 values(2, EMPTY\_BLOB());**

**INSERT INTO planets4 values(3, EMPTY\_BLOB());**

**INSERT INTO planets4 values(4, EMPTY\_BLOB());**

**INSERT INTO planets4 values(5, EMPTY\_BLOB());**

**INSERT INTO planets4 values(6, EMPTY\_BLOB());**

**INSERT INTO planets4 values(7, EMPTY\_BLOB());**

**INSERT INTO planets4 values(8, EMPTY\_BLOB());**

**INSERT INTO planets4 values(9, EMPTY\_BLOB());**

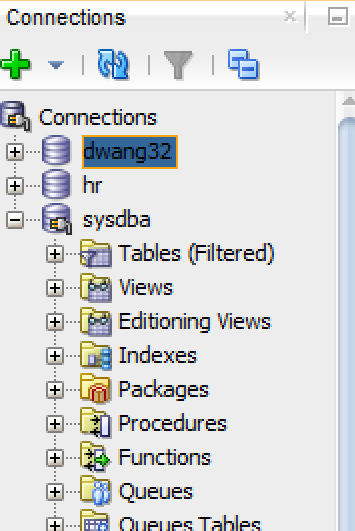
**INSERT INTO planets4 values(10,Empty\_BLOB());**

**commit;**

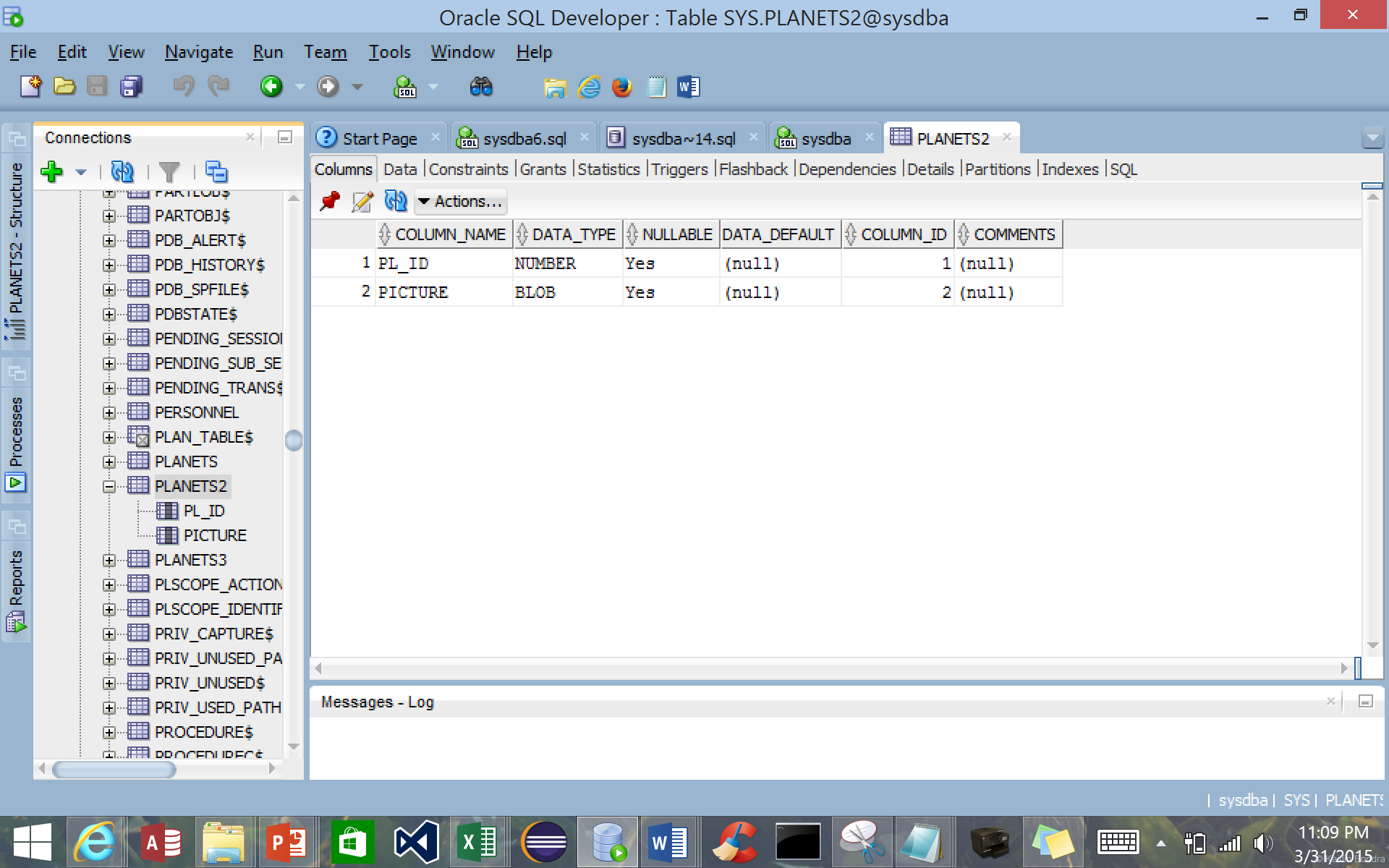
**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**STEP 3**  **Load the Pictures of the Planets**

**(1)** Find your table planets2 from your connection.

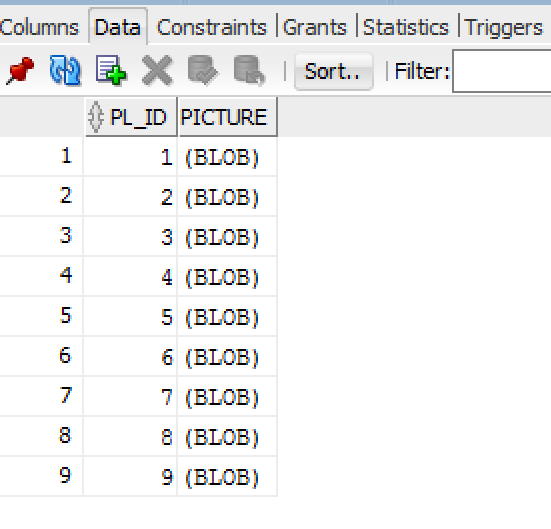


**(2)** Click on the table to reveal its structure.



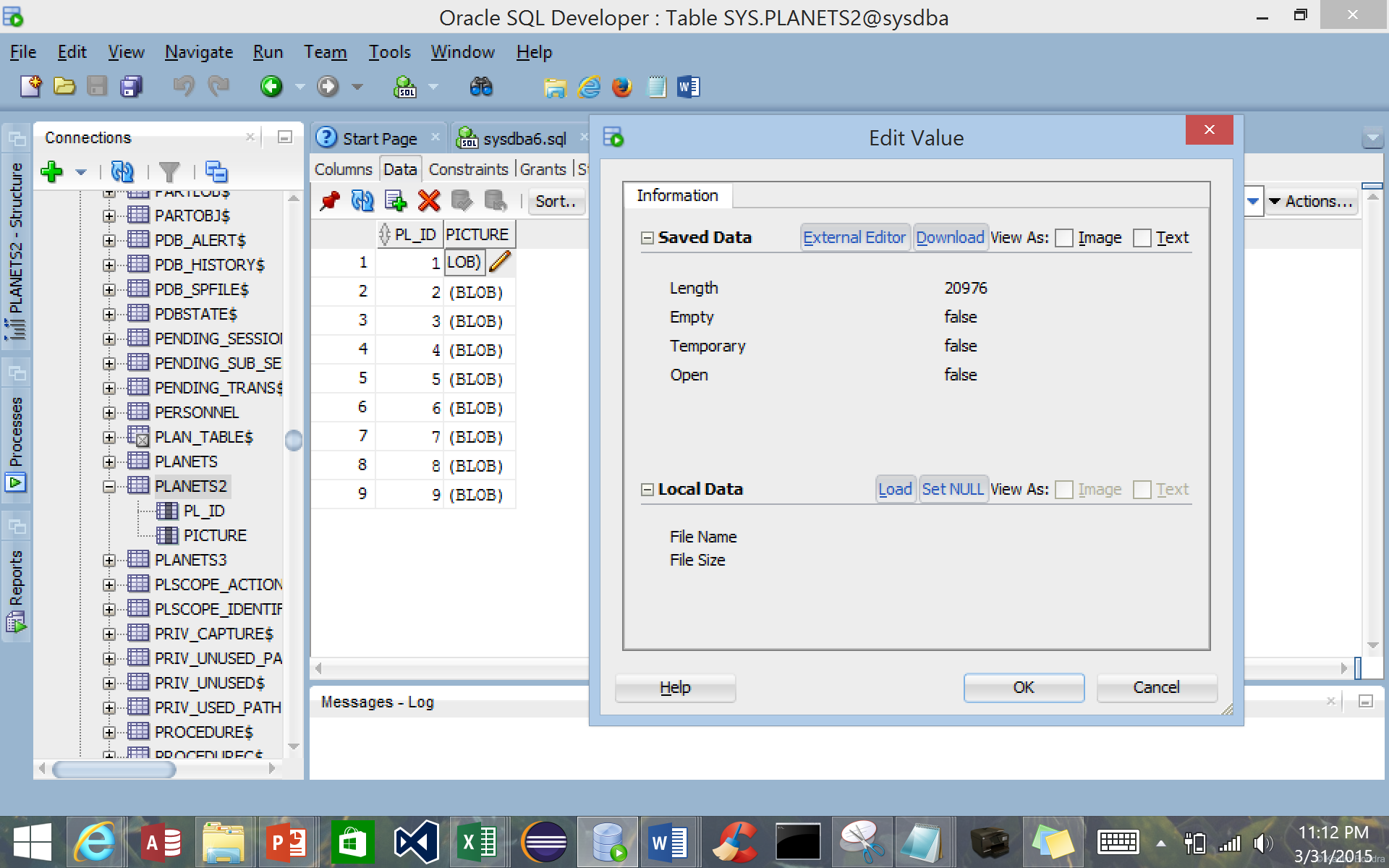
**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**(3)** Click on the Data tab located to the right of the Columns tab.



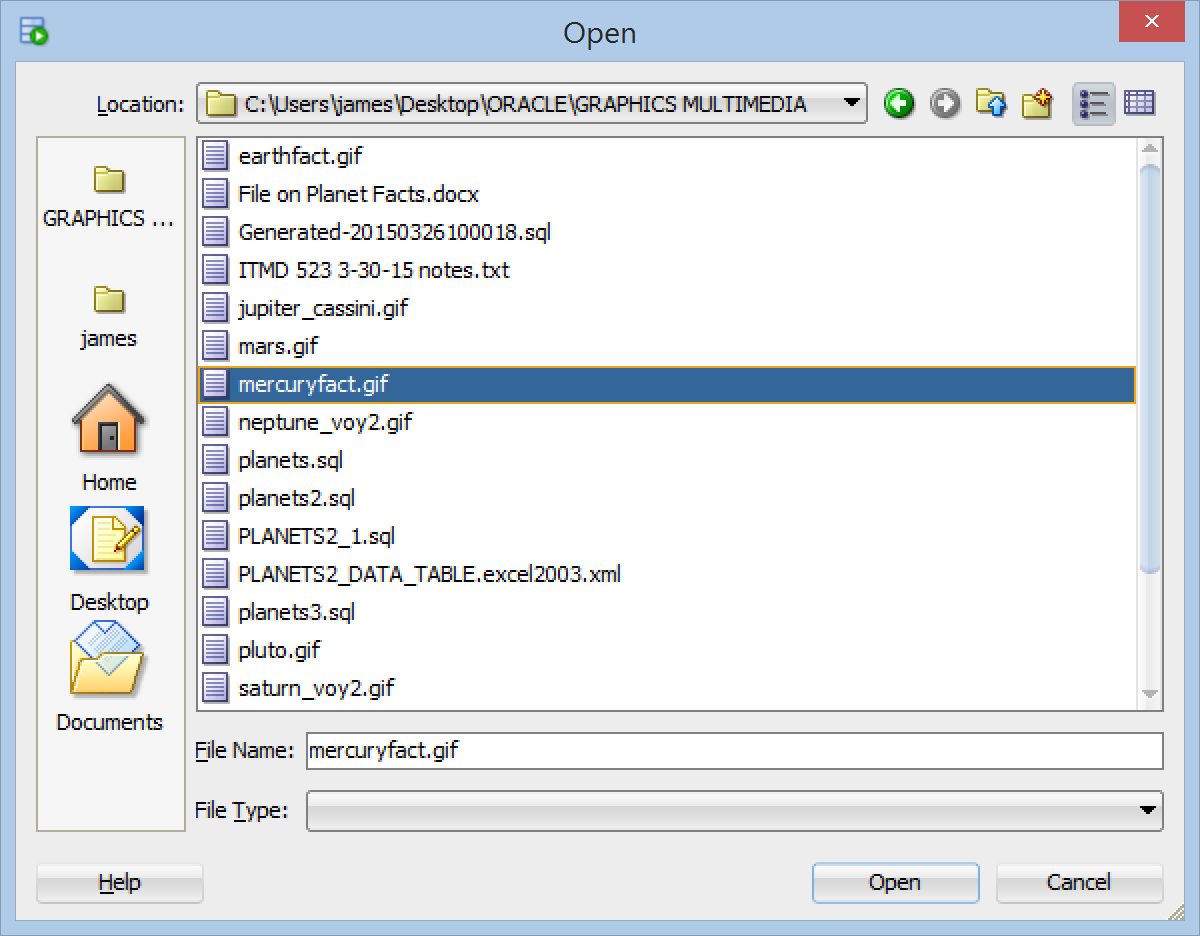
**(4)** Double click on the Picture field’s first record (BLOB). Click on the pencil to open

the Edit Value window.

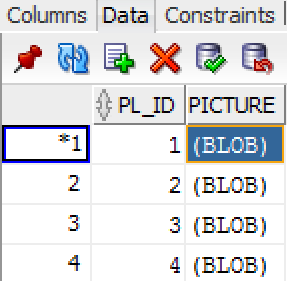


**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**(5)** Click on Load to find the image associated with the record. Then click OK.



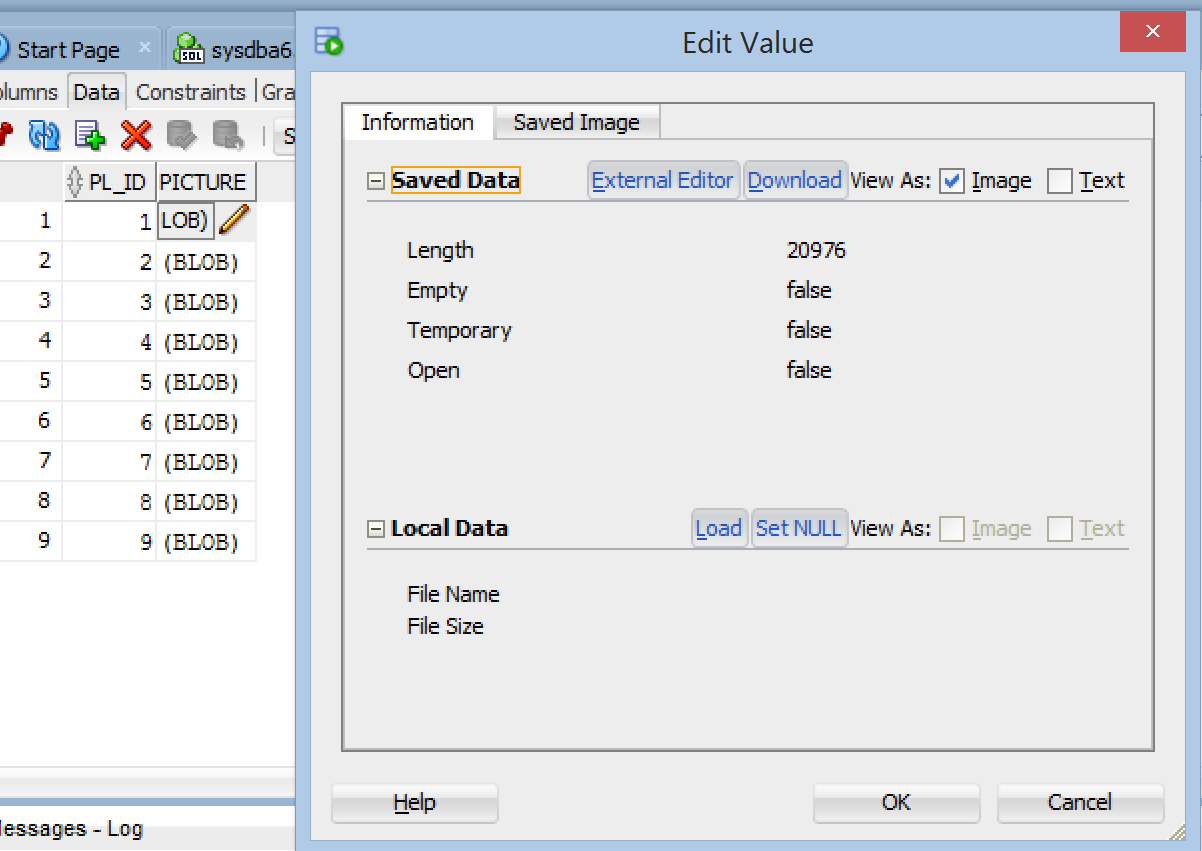
**(6)** You will see an asterisk to the left of the row number.

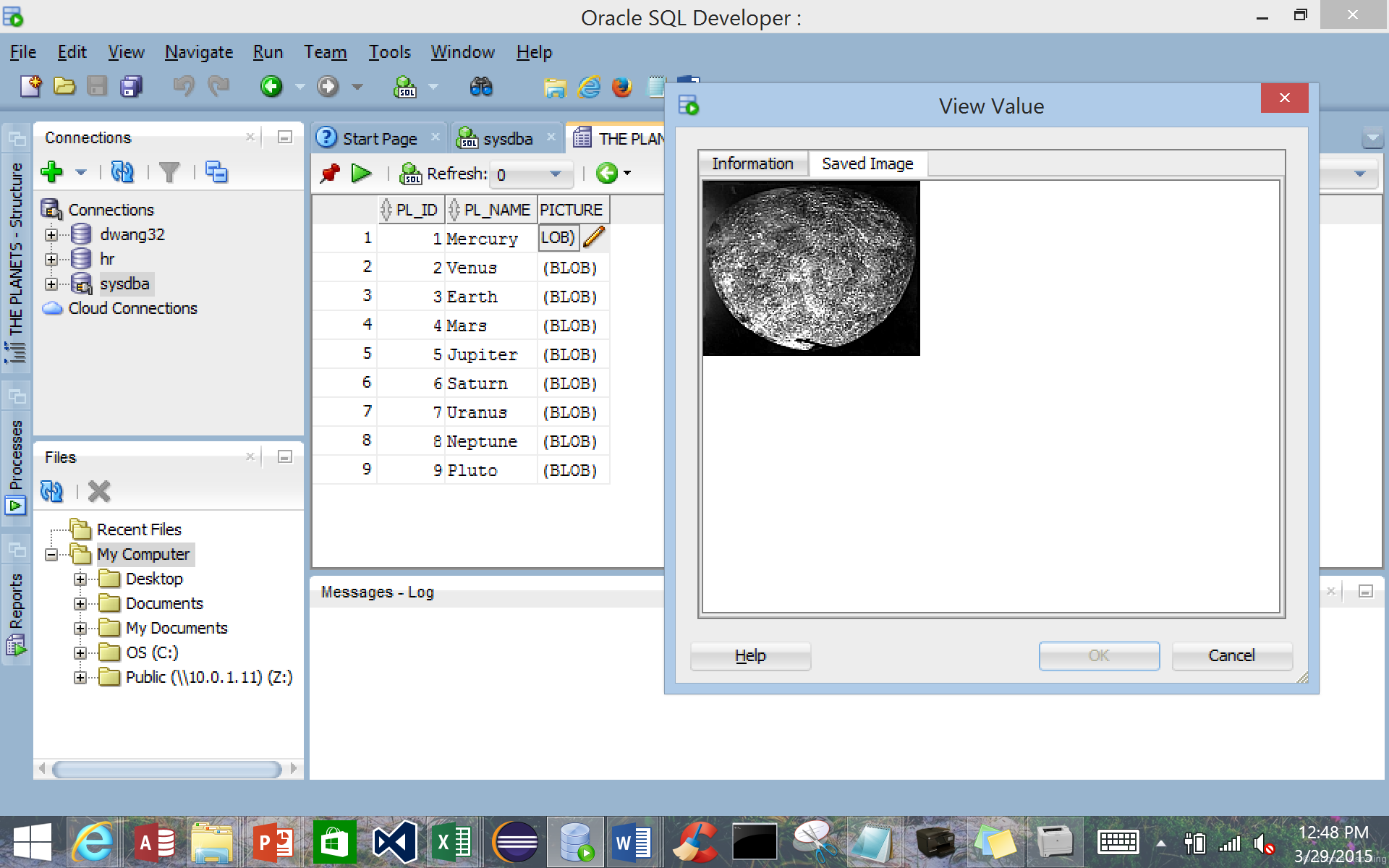


**(7)** After you have loaded all your images, commit your changes [ F11 ] by clicking on the icon  .

**(8)** You will be able to view the image! Just click on the record’s BLOB field, check View As Image in the Edit Value Window, and the saved image appears.

**PROJECT : PL / SQL - Retrieving Data with Select Statements**





**PROJECT : PL / SQL - Retrieving Data with Select Statements**

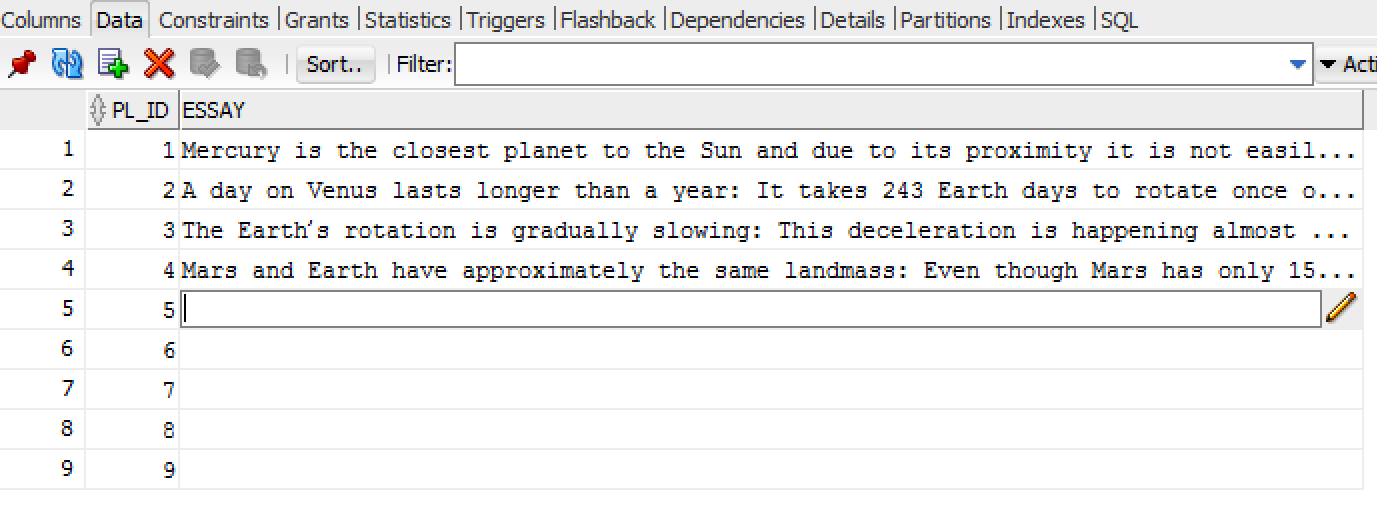
**STEP 4**  **Load the Essay Field in the planets3 Table**

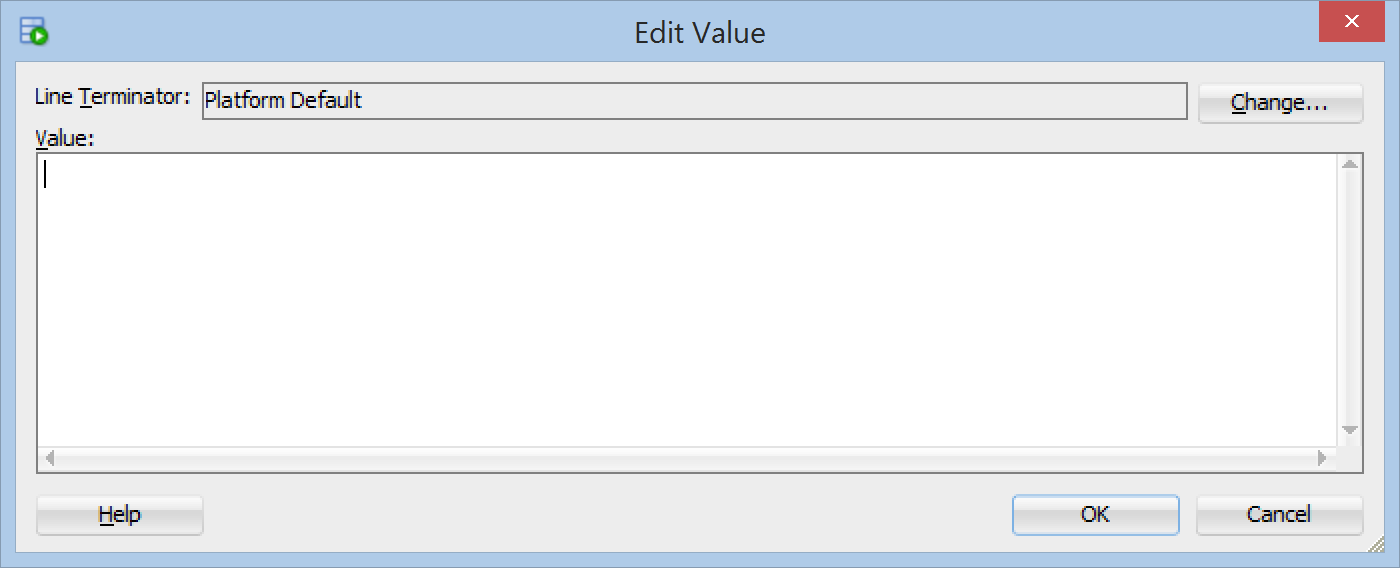
**(1)** Find the file "File on Planets Fact." This file will contain information on each of the nine planets.

**(2)** It is better to update the essay field for each record with the information per planet; alternatively, try loading the essay CLOB field **the same way the**

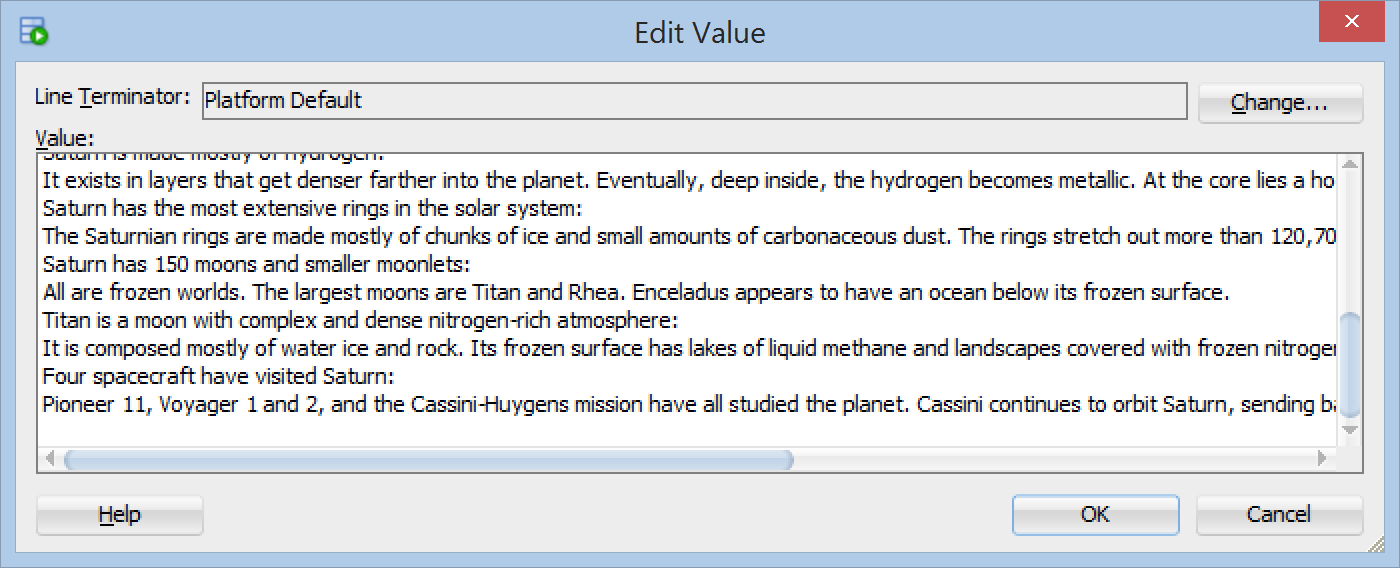
**images** were loaded for the planets2 table. Double click the pencil. The Edit

value will appear; place the text for the planet into the box and click OK.

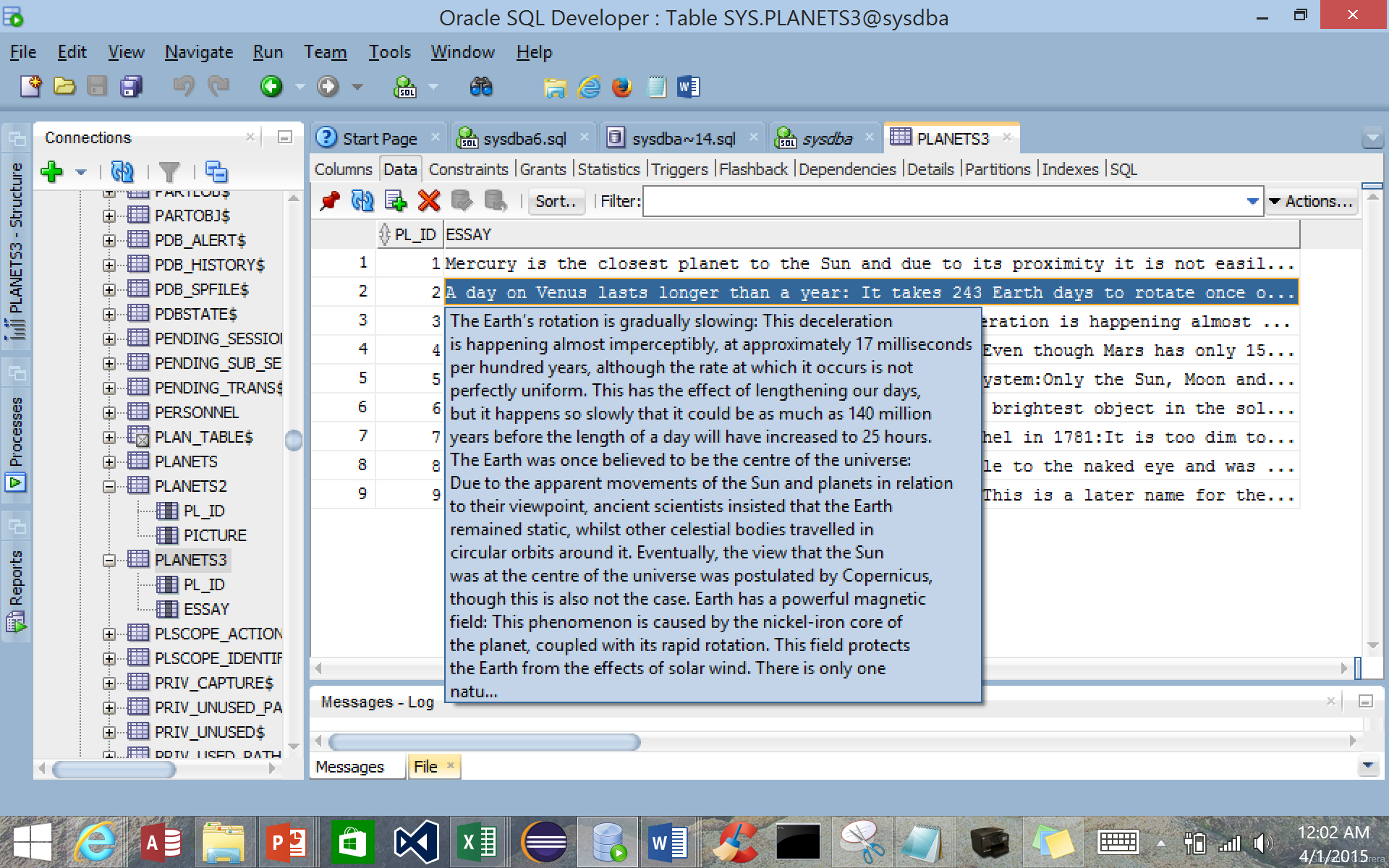




**PROJECT : PL / SQL - Retrieving Data with Select Statements**



**(3)** As you hover over the record, you should see a box pop - up with the appropriate text.



**STEP 5 Load the Music files in the planets4 Table**

Visit the Web site [**https://www.purple-planet.com/**](https://www.purple-planet.com/) for royalty free music clips. Populate at least two records with **mp3** files.

**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**STEP 6**  **Perform the Following Join**

(a) Perform the following join between the planets, planets2 and planets3

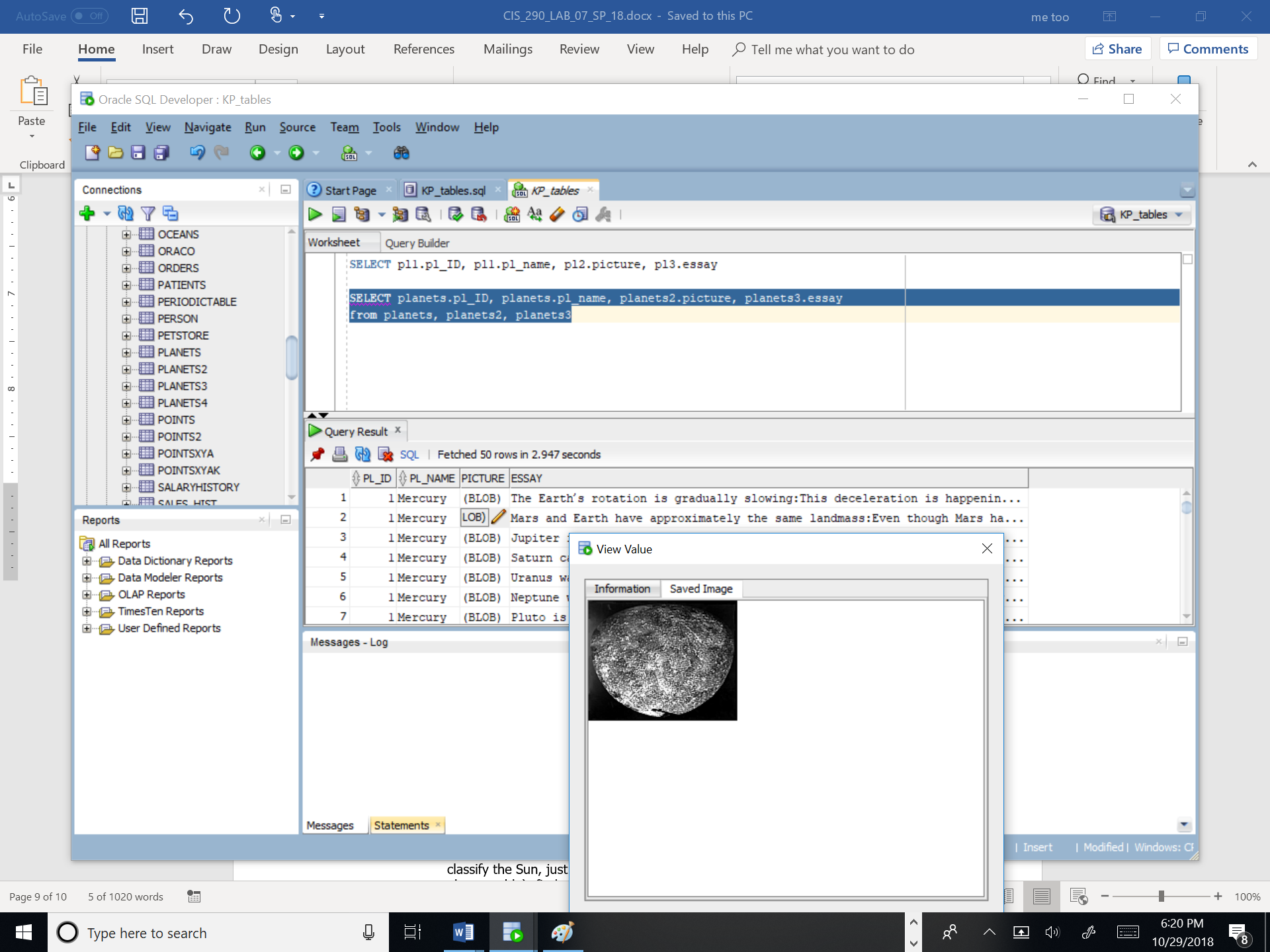
tables.

**SELECT pl1.pl\_ID, pl1.pl\_name, pl2.picture, pl3.essay**

**FROM planets pl1, planets2 pl2, planets3 pl3**

**WHERE pl1.pl\_ID = pl2.pl\_ID and pl1.pl\_ID = pl3.pl\_ID;**

You will see the following:



(b) Perform the following join between the planets, planets2 and planets4

tables.

**SELECT pl1.pl\_ID, pl1.pl\_name, pl2.picture, pl3.essay**

**FROM planets pl1, planets2 pl2, planets4 pl4**

**WHERE pl1.pl\_ID = pl2.pl\_ID and pl1.pl\_ID = pl4.pl\_ID;**

**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**STEP 7**  **Create a Report on the Join**

Create a report based on the first join conditions(s) given above. Include your **own name** in the title as the originator of the report.

**STEP 8**  **Insert a New Record for All Three Tables**

Insert two new records for all three tables, with respect to the SUN

( if you classify the Sun, just put in the word Star for the class field in the planets table ) : find an image of the SUN ( preferably a gif file to be consistent ) , and some facts for the SUN; and with respect to Ceres, an asteroid that is now considered a dwarf planet, and is located in the Asteroid belt.

**STEP 9**  **Perform a Natural Join Between All Three Tables**

Now, construct a Natural Join between all three tables.

**SELECT \* from planets NATURAL JOIN planets2 NATURAL JOIN planets3**

**STEP 10**  **Listen to the Music and View the Images.**

See if you can visualize the images and hear the music! Take a snapshot of the report generated for the 3 - table join.

**STEP 11 Questions and Reflections Concerning this Database Project**

Now that you have completed this lab project, review the questions below to reflect on the procedures and settings that you utilized as you followed the steps to complete the project. Place your responses in your lab submittal document.

**(1)** Managers of media services that offer video and audio media files for viewing and listening, such as YouTube, NetFlix, Amazon Prime, Hulu, and Google Play, have a tremendous task of storing the media, auditing the files, uploading new media and removing media. Customers are becoming more discriminatory, especially parents. How would you structure queries to filter media content that is geared toward children versus adults, comedies versus dramas and documentaries versus independent films? Give examples to support your answer.

**(2)** Many cultural institutions, such as art museums, planetariums, orchestras and zoos, also maintain databases containing files ( customer / member information, media files, image files ( photographs ) , digital news articles ) that also must be maintained. Construct a database containing various tables that will classify such files ( as mentioned here ) for Century City’s 21st Century exhibition on the history of the city. Be descriptive as to the attributes of each table.

**(3)** Intellectual Property includes patents, trademarks, copyrights and trade secrets. Perform research as to the legal lifetime of these four important intellectual properties in the United States.

**PROJECT : PL / SQL - Retrieving Data with Select Statements**

**(4)** When conducting research in astronomy, there are often new discoveries made that may add to, corroborate or contradict known data. You are the database manager for NASA. How are you do deal with these discoveries and their placement into the appropriate files?

**(5)** Would it be preferable to store astronomical data in NoSQL files or in

traditional SQL files? Support your answer.