STSCI 5060 Lab 3

(11/5/2018)

In this lab, you will use Oracle's graphical user interface, SQL Developer, to practice the advanced SQL queries covered in the class. You are required to use the account (user name) named with your first name, which was created in Lab 2.

Part One: Become Familiar with SQL Developer

1. Start SQL Developer by double clicking the shortcut to sqldeveloper.exe on the desktop if there is one, or go to "All Programs" or the folder in which you put the SQL Developer program, to start it.

2. Create a connection to an Oracle database (It is a necessary step in order to use the Oracle database in SQL Developer):
   - Right click "Connections" in the upper left corner of the screen
   - Click "New Connection…"
   - The "New / Select Database Connection" window opens.
     - Enter a Connection Name using your first name.
     - Enter the username (also your first name, which is your database account name or user ID) and the password you created in Lab 2.
     - Click the "Test" button at the bottom of the window and you should see "Status: Success" (lower left corner).
     - Click "Connect" to connect to your Oracle database.
     - Your new connection should be added to the Connections Panel.
     - You will do all your practice on the "Worksheet" under your connection tab.

3. Your instructor will briefly demonstrate the above steps and a few other important points of SQL Developer.

4. **Important points for creating your lab report:**

   - You are required to submit a lab report, which has two parts, your script output and the contents of your worksheet, i.e. your SQL script. You execute your SQL queries by clicking the "**Run Script**" button or by pressing the F5 key (**Important to remember:** **highlight the code you want to run first; otherwise, you will run the whole script on your worksheet, which may make undesirable changes to your database**). At the end of your lab practice, save your script output as a text file (by clicking the save button right below the "Script Output" label) and name it as "last name_ first name_lab3_script_output.txt" (for example, Smith_John_lab3_script_output.txt). Edit this file to insert a question number at the beginning of each practice step. Leave some whitespace between any two practice outputs.

- Save the contents of your worksheet by clicking File→Save with a file name consisting of your last name, an underscore, first name and "_lab3_worksheet.sql" like Smith_John_lab3_worksheet.sql.

Part Two: Practice Advanced SQL Queries

1. Document your lab practice by typing

   -- STSCI 5060 Lab 3
   --Student Name: FirstName LastName

   --P1.

   Note: For every practice step, you start a new paragraph separated with a blank line. Each paragraph must start with "--P# " (where # stands for the practice step number).

   Set your output line size to 500 columns.
   Run the following statements on the Worksheet to begin documenting your lab practice:

   ```
   SELECT user FROM Dual;
   SELECT Sysdate FROM Dual;
   ```

2. Run the same SQL script as in Lab2, i.e., createpvfc.sql, in your worksheet window so that you have some brand new database tables to start with.

3. Use two methods, a WHERE clause and an INNER JOIN…ON clause, to find the names, their states and postal codes, order IDs and order dates for the customers who have placed an order. Compare and comment on the results you got; you can enter your comments on the worksheet by using "--" or "/*…*/". You should put your comments right after the related query. Do the same for the rest of this lab if required.

4. Practice OUTER JION queries. Use LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN to find the customer ID, name, and order number for all customers, including customer information even for the customers who have not ordered anything. Compare and comment on the results you got from the three OUTER JOIN methods. Based on your

results, point out the names of the customers who have never ordered so far. (Note: you must use the keyword "ON" before the condition clause in Oracle.)

5. You are required to use a view (or views) to obtain to obtain your final results with breaking down an otherwise more complicated query into simpler steps. Your business goal is to find out the customers who put orders that have a total order amount more than $600 for a single product. You need to display customer ID, name, address, city, state, postal code, order ID, product description, the quantity ordered, standard product price, and total order amount. Use an alias called "Total_Amount" for the total order amount.

6. Use CROSS JOIN to do a Cartesian Join although for most times you want to avoid such a join. Find out all possible combinations of the product description and order date from the two tables involved.

7. Use a subquery in the FROM clause. List product description, product finish, and product standard price of the products that have a standard price less than the average price of all the products. Your subquery should form a derived table used in the FROM clause of the outer query. Use an alias to represent the average price.

8. Write a correlated subquery containing the LIKE keyword and a wildcard. List all the orders (i.e., OrderIDs) that include furniture that has a finish type starting the word "Natural" and with a standard price greater than $300.

9. Practice on ensuring transaction integrity with the ROLLBACK statement. The steps below can run together as a SQL script.
   A. Create a new table called "ProductNew_T" that is an identical copy of the Product_T table using a query.
   B. Update the standard price of Dining Table to $1800 in the new table.
   C. Confirm the change by displaying the whole table.
   D. Issue the ROLLBACK command.
   E. Confirm the ROLLBACK result by display the whole table again.

   Now you add a COMMIT statement to above code so that the ROLLBACK command does not have an effect. Run the updated code and confirm the results.

10. Obtain information from the database dictionary. Write a query to find out the owners, table names, statuses, and user stats of all the tables in the database. (Hint: you may want to describe the table to find out the column names you need.)

11. Create a function called get_price on table Product_T to return the standard price of a specified product ID. After you have created and compiled the function, you can call the function by using a SELECT statement and providing a product ID. Use your function to get the standard prices of product IDs 1, 2, 3, and 4. You can run your SELECT statements together as a group.

    Bonus practice (earn up to 5 extra points): Write a query, in which you must use the function you just created, to find out the standard price, product description, and product finish of product ID 5.

12. Create a procedure called update_price on table Product_T so that you can update the standard price of a product by providing the product ID and new price. After you have created and compiled the procedure, you can execute the procedure by using an EXECUTE statement. Use your procedure to update the standard prices of the following product ID and new price pairs. After the updates, confirm your result with a query.

Product ID	New Price
1	818
2	828
3	838

13. Create a trigger for an update event of the OrderedQuantity column of the OrderLine_T table to trigger an automatic insertion into an audit table called Orders_Audit_T **after** the event. First, you need to create the Orders_Audit_T table with the following column specifications:

    Order_ID Number(11,0), which is the primary key
    Quantity_Before Number(10,0)
    Quantity_After Number(10,0)
    Username Varchar2(50)

    Name the trigger Order_After_Update. After you have compiled your trigger, update OrderLine_T with the following updates:

OrderID	ProductID	New OrderedQuantity
1001	1	8
1003	3	6
1004	6	20
1007	1	6

List the contents of Orders_Audit_T and OrderLine_T tables to confirm your updates and the trigger results.

When you are done, save the worksheet and script output contents as instructed above. Use 7-Zip (or a similar tool) to compress your two files into one zip file (lastName_firstName_Lab3.7z). Submit your zip file to course website by the end of Saturday (11:59pm, 11/10/2018).