STSCI 5060 Lab 2

(10/22/2018)

In this lab, you will learn some new useful commands in SQL and practice many basic SQL statements covered in the class. You are required to use the account (or the user name) named with your first name, which you created in Lab 1. A file called *LastName_FirstName_Lab2.log* will be created by you to document what you did in the lab. You submit this file to the course website for grading. It is due on Tuesday (11:59pm, 10/23/2018).

1. Suppose you save your files of this lab in the following directory: C:\STSCI5060\. At SQL> prompt, enter the following command to record what you do in this lab (Note: your file path cannot contain a blank):

   spool C:\STSCI5060\LastName_FirstName_Lab2.log

   Enter the following at the SQL> prompt ("/*" and "*/" are SQL comment delimiters, similar to those in SAS):

   ```
   /* Step 1.
   Create a log file to document my lab practice:
         spool C:\STSCI5060\LastName_FirstName_Lab2.log
   */
   ```

   For the rest of steps, you start each step with a comment like the following to mark the beginning of a new step (n stands for the step number).

   ```
   /* Step n. */
   ```

   If there are any sub-steps, do the same for the sub-steps too, for example:

   ```
   /* Step 5a. */
   ```

   You may also put extra information in the comments to explain what you are doing.

   Another way to add comments in Oracle is to start a single comment line with two dashes, for example:

   ```
   -- Step 1.
   -- Create a log file to document my lab practice:
   --       spool C:\STSCI5060\LastName_FirstName_Lab2.log
   ```

2. Change your output settings. Issue the following commands:

   ```
   SET pagesize 250
   SET linesize 250
   ```

3. Display today's date by entering:

   SELECT Sysdate FROM Dual;

4. Download a file called CREATEPVFC.SQL from the course website and save it to C:\STSCI5060\. Create and populate many tables together from this SQL script. Enter:

   Start C:\STSCI5060\CREATEPVFC.SQL

5. Check to see how many tables you have in your schema and display some table specific contents.
   a. Enter: SELECT * FROM Cat;
   b. Describe the Customer_T table;
   c. Display all the contents of the Customer_T table.

6. Insert new date into your tables.
   a. Use INSERT INTO to populate the Customer_T table with the following data.
      INSERT INTO Customer_T VALUES
      (020, 'Office Plus Furniture', '160 Bigler Rd.','University Park','PA',16802);
   b. Insert more data into the Customer_T.

18	Office Furniture	100 Tower Rd.	Ithaca	NY 14853
16	Lab Fixtures	11 Downtown Rd.	Ithaca	NY 14850

   c. To see if you can insert the following data into the Customer_T table. Why? Give your answer as a SQL comment by enclosing it in /* …*/ (same below).

10	HomeArt	101 Uptown Rd.	Ithaca	NY 14850

   d. For CustomerID=21, insert your own information into the Customer_T table (Note: if you do not want to use your real address, you may use a modified value but you must use your real name for the CUSTOMERNAME column).
   e. Display all the data in Customer_T table; check the order of the data in the table. What did you see?
   f. Run a query to sort the result you got from Step 6e by CustomerID in ascending order.

7. Use a query to populate a new table.

   a. Create a new table, NewCust_T, from an existing table, Customer_T;
   b. Describe NewCust_T and display all the data in the NewCust_T table. What did you see.
   c. Issue a SQL command to remove all the data form NewCust_T.
   d. Again describe NewCust_T and display all the data in the NewCust_T table.

8. Use a query to populate an existing table.
   a. Use INSERT INTO to populate NewCust_T with the data in Customer_T;
   b. Display all the data in the NewCust_T table.

c. Completely remove the NewCust_T table, including the data and table definition.
d. Try to describe NewCust_T and display all the data in NewCust_T. What happened?

9. Create a new table structure from an existing table.
   a. Enter:

```
CREATE TABLE NewCust2_T AS
SELECT * FROM Customer_T
WHERE 1=2;
```

   b. Describe NewCust2_T and display all the data in new table. Comment on why you got this result.
   c. Insert the following new data into NewCust2_T:
      017, 'Furniture Star', '16 Farm Rd.','College Station','TX', 77840
   d. Display the data in the table.

10. Update data in a table.
    a. Change the CustomerName of the customer with ID = 17 to “Furniture King” in the NewCust2_T table.
    b. Display the data in the table to verify the change.

11. Using the wildcards.
    a. Find all the information about the customers whose names start with the letter “H” in the Customer_T table.
       (Hint: SELECT * FROM …;)
    b. First find out what is stored in the Product_T table and then list all the information about all the desks in the Product_T table.

12. Using expressions in the SELECT statement.
    List ProductID, ProductStandardPrice, 30% discount price of the Product Standard Price (named *ThirtyPercentDiscountPrice*) From the Product_T.

13. Using aggregate function COUNT.
    a. Change the value of CustomerPostalCode to NULL for the customer with ID = 1 in the Customer_T.
    b. Use COUNT(*) to count the number of customers in Florida.
    c. Use COUNT(CustomerPostalCode) to count the number of customers in Florida.

14. Referencing a single-valued column when using an aggregate function. Describe what happened after you submitted the following code and what caused the error.

```
SELECT ProductID, COUNT (*)
   FROM Orderline_T
      WHERE OrderID = 1004;
```

15. Reference a single-valued column when using an aggregate function in a subquery. List the ProductDescription and the difference of a product's ProductStandardPrice from the minimum price of all the products; name this difference as Difference. (Hint: the code is very similar to the example given in the class)

16. Find out ProductDescription, ProductFinish, ProductStandardPrice information for all the desks or tables whose prices are not between $200 and $300.

17. Find out the customer names, city and state information of all the customers who are from the states of Florida, Pennsylvania, California, and New York.

18. Categorize the customers by state and find out the number of customers in each state. Your final query result should only contain the rows for the states where there are more than 2 customers and your result must be sorted by state in ascending order.

19. Create view:
    a. Enter the following:

```
CREATE VIEW Invoice_V AS
   SELECT c.CustomerID, CustomerAddress, o.OrderID, p.ProductID,
     ProductStandardPrice, OrderedQuantity
      FROM Customer_T c, Order_T o, OrderLine_T ol, Product_T p
        WHERE c.CustomerID=o.CustomerID
          AND o.OrderID = ol.OrderID
          AND p.ProductID=ol.ProductID;
```

    b. Describe Invoice_V and display all the data in the view.

20. Modify the code in Step 19 and change the view (Invoice_V) to a materialized view (Invoice_MV), and then describe the materialized view and display all the data in that view. Can you see a difference just by looking at the query results of Step 20 and Step 19b?

21. Execute the following query and look at your query result. What do you see from the Tabletype column about the two views you created above?

```
SELECT * FROM tab;
```

22. Enter the following at SQL> prompt:
    SELECT TO_CHAR(SYSDATE, 'DD-MON-YY HH:MM:SS') FROM DUAL;

23. Output your log file by entering the following at your SQL prompt:
    spool out

24. Upload your final *LastName_FirstName_Lab2.log* file to the course website.