Lab 03 – Week 4 (Sub-Queries)

This week's lab continues using the SELECT command and learning the interfaces for both SQL Developer and introduces the use of single-line functions.

Submission

Your submission will be a single text-based SQL file with appropriate header and commenting. Please ensure your file runs when the entire file is executed in SQL Developer.

Create a new Worksheet in SQL Developer. Save the file as L03_ID#_LASTNAME.sql

Your submission needs to be commented and include the question number and the solutions.

Using comments to number the question answers, write the SQL code to complete the following tasks.

Example Submission

```
-- **********************
-- Name: Your Name
-- ID: #########
-- Date: The current date
-- Purpose: Lab 03 DBS311
-- **********************


-- Q1 SOLUTION --
SELECT * FROM TABLE;


-- Q2 Solution –
SELECT * FROM TABLE;
```

Style Guide

Your SQL should be written using the standard coding style:

- all keywords are to be upper case,
- all user-defined names are to be lower case, (example: table and field names)
- there should be a carriage return before each major part of the SQL statements (i.e. before SELECT, FROM, WHERE and ORDER BY)

See the following sample:

```
SELECT columns
FROM tables
WHERE conditions
ORDER BY column1, column2;
```

Marking Scheme

Question	Weight
1	16%
2	16%
3	17%
4	17%
5	17%
6	17%

Grade Policy

- Submissions with errors do not get any marks.
  - o Execute your *.sql* file using the "Run Script" button to make sure there is no errors in your file.
- If your result in a question does not match the sample output results, you do not get any marks.
- You do not receive marks for the missing solutions.

Tasks

For each question, the title of columns and the output result must match the provided output result in that question.

1. Write a SQL query to display the last name and hire date of all employees who were hired before the employee with ID 107 got hired but after **March** 2016. Sort the result by the hire date and then employee ID.

	LAST_NAME	HIRE_DATE
1	Hayes	06-APR-16
2	Henderson	10-APR-16
3	Simmons	10-APR-16
4	Stevens	21-APR-16
5	Henry	21-APR-16
6	Kennedy	23-APR-16
7	Ferguson	24-APR-16
8	Coleman	01-MAY-16
9	Boyd	11-MAY-16
10	Sanders	18-MAY-16
11	Cox	21-MAY-16
12	Rose	23-MAY-16
13	Warren	24-MAY-16

2. Write a SQL query to display *customer name* and *credit* limit for customers with lowest credit limit. Sort the result by customer ID.

NAME	CREDIT_LIMIT
1 Raytheon	100
2 Plains GP Holdings	100
3 US Foods Holding	100
4 AbbVie	100
5 Centene	100
6 Community Health Systems	100
7 Alcoa	100
8 International Paper	100
9 Emerson Electric	100
10 MGM Resorts International	100
11 Farmers Insurance Exchange	100

3. Write a SQL query to display the product ID, product name, and list price of the highest paid product(s) in each category. Sort by category ID and the product ID.

	CATEGORY_ID	PRODUCT_ID	PRODUCT_NAME	LIST_PRICE
1	1	228	Intel Xeon E5-2699 V3 (OEM/Tray)	3410.46
2	2	133	PNY VCQP6000-PB	5499.99
3	3	190	Supermicro X10SDV-8C-TLN4F	948.99
4	4	50	Intel SSDPECME040T401	8867.99

4. Write a SQL query to display the category ID and the category name of the most expensive (highest list price) product(s).

	CATEGORY_ID	CATEGORY_NAME
1	5	Storage

5. Write a SQL query to display product name and list price for products in category 1 which have the list price less than the lowest list price in ANY category. Sort the output by top list prices first and then by the product ID.

	PRODUCT_NAME	LIST_PRICE
1	Intel Xeon E5-2687W	710.99
2	Intel Xeon E5-2680 V2	701.95
3	Intel Core i7-980	699.99
4	Intel Core i7-7820X	678.75
5	Intel Core i7-3930K	660
6	Intel Xeon E5-2630 V4	647.99
7	Intel Xeon E5-2630 V3	629.99
8	Intel Core i7-4930K	624.04
9	Intel Core i7-4790K	620.95
10	Intel Xeon E5-2640 V2	608.95
11	Intel Xeon E5-1650 V4	601.99
12	Intel Xeon E5-1650 V4 (OEM/Tray)	594.99
13	Intel Xeon E5-2630 V3 (OEM/Tray)	589.99
14	Intel Xeon E5-2630 V2	588.95
15	Intel Xeon E5-1650 V3	564.89
16	Intel Core i7-5930K	554.99

6. Display the maximum price (list price) of the category(s) that has the lowest price product.

	MAX(LIST_PRICE)
1	8867.99