

Lab 02 – Week 3 (Multi-row Functions)

Lab Due: October 12, 11:59pm

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 L02_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 02 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;
```

To save time, you can write all SQL statement in your SQL developer. To make sure that your SQL statements style follows the standard SQL style guide, copy and paste your SQL statement onto the following website and click on “**FORMAT SQL**” or “**FORMAT SQL IN NEW WINDOW**”.

<https://www.freeformatter.com/sql-formatter.html#ad-output>

You can also upload your SQL file. See the setting in the following image. Have SQL keywords (SELECT, INSERT, UPDATE, etc.) uppercase and user defined objects and identifiers (tables, columns, etc.) lowercase.

The screenshot shows the SQL Formatter website interface. It has two main options: 'Option 1: Copy-paste your SQL query here' and 'Option 2: Or upload your SQL document'. Under Option 1, there is a text area containing the SQL query: 'select * from orders;'. Under Option 2, there is a 'Browse...' button, a 'No file selected.' message, and a 'UTF-8' dropdown menu. Below these, there are settings for 'Indentation level' (set to '3 spaces per indent level') and 'Change case of SQL keywords (select, from, where, etc.):' (set to 'Modify to upper case'). There is also a 'Change case of identifiers (tables, columns, etc.):' (set to 'Modify to lower case'). At the bottom, there are two buttons: 'FORMAT SQL' and 'FORMAT SQL IN NEW WINDOW'.

Marking Scheme

Part

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

Grade Policy

- Submissions with errors do not get any marks. (They get zero.)
 - 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.

- For each job title, display the number of employees. Sort the result according to the number of employees.

	JOB_TITLE	EMPLOYEES
1	Accounting Manager	1
2	President	1
3	Marketing Manager	1
4	Public Accountant	1
5	Human Resources Representative	1
6	Purchasing Manager	1
7	Finance Manager	1
8	Administration Assistant	1
9	Marketing Representative	1
10	Public Relations Representative	1
11	Administration Vice President	2
12	Stock Manager	5
13	Programmer	5
14	Purchasing Clerk	5
15	Accountant	5
16	Sales Manager	5
17	Stock Clerk	20
18	Shipping Clerk	20
19	Sales Representative	30

- Display the highest, lowest, and average customer credit limits. Name these results high, low, and average. Add a column that shows the difference between the highest and the lowest credit limits named "High and Low Difference". **Round** the average to 2 decimal places.

	HIGH	LOW	AVERAGE	High Low Difference
1	5000	100	1894.67	4900

- Display the order id, the total number of products, and the total order amount for orders with the total amount over \$1,000,000. Sort the result based on total amount from the high to low values.

	ORDER_ID	TOTAL_ITEMS	TOTAL_AMOUNT
1	70	727	1278962.17
2	46	1032	1269323.77
3	78	585	1198331.59
4	1	1343	1143716.87
5	68	885	1088670.12
6	27	1203	1084871.49
7	32	747	1081679.88
8	92	790	1050939.97
9	59	881	1043144.72

4. Display the warehouse id, warehouse name, and the total number of products for each warehouse. Sort the result according to the warehouse ID.

	WAREHOUSE_ID	WAREHOUSE_NAME	TOTAL_PRODUCTS
1	1	Southlake, Texas	5483
2	2	San Francisco	28613
3	3	New Jersey	7252
4	4	Seattle, Washington	14860
5	5	Toronto	12969
6	6	Sydney	20457
7	7	Mexico City	9039
8	8	Beijing	13482
9	9	Bombay	7357

5. For each customer, display customer number, customer full name, and the total number of orders issued by the customer.
- If the customer does not have any orders, the result shows 0.
 - Display only customers whose customer name starts with 'O' and contains 'e'.
 - Include also customers whose customer name ends with 't'.
 - Show the customers with highest number of orders first.

	CUSTOMER_ID	customer name	total number OF orders
1	44	Jabil Circuit	5
2	69	Whole Foods Market	1
3	58	Health Net	1
4	174	Oracle	0
5	196	DuPont	0
6	113	Microsoft	0
7	127	Alphabet	0
8	117	Home Depot	0
9	158	Owens & Minor	0
10	255	Waste Management	0
11	27	Assurant	0
12	260	Occidental Petroleum	0
13	129	Target	0
14	226	Office Depot	0
15	197	Avnet	0
16	88	Walmart	0
17	128	Comcast	0

6. Write a SQL query to show the total and the average sale amount for each category. **Round** the average to 2 decimal places.

	CATEGORY_ID	TOTAL_AMOUNT	AVERAGE_AMOUNT
1	1	20041993.03	126050.27
2	2	13787520.62	131309.72
3	5	13994160.84	53209.74
4	4	4918418.15	35640.71