**Lab 02 – Week 3 (Multi-row Functions)**

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

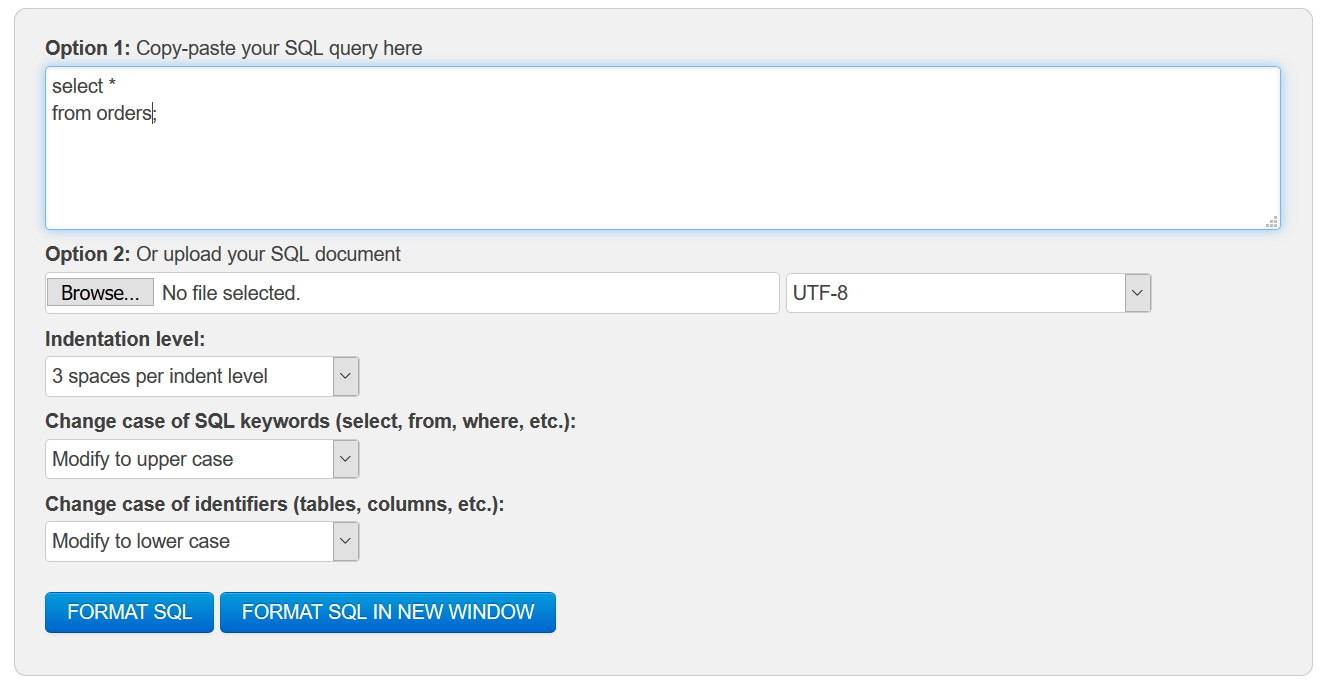
**WHERE** conditions

**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.



**Marking Scheme**

**Part**

|  |  |
| --- | --- |
| **Question** | **Weight** |
| 1 | 16% |
| 2 | 16% |
| 3 | 17% |
| 4 | 17% |
| 5 | 17% |
| 6 | 17% |

## 

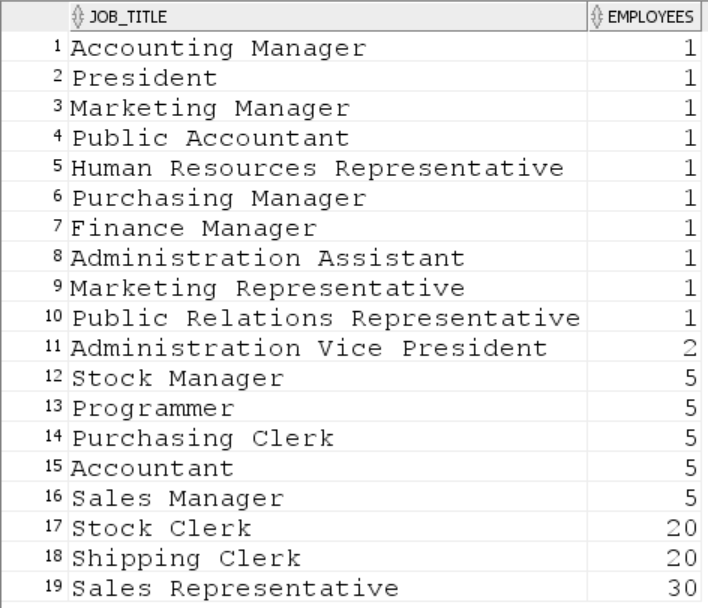
**Grade Policy**

* 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.

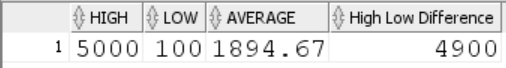
## **Tasks**

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

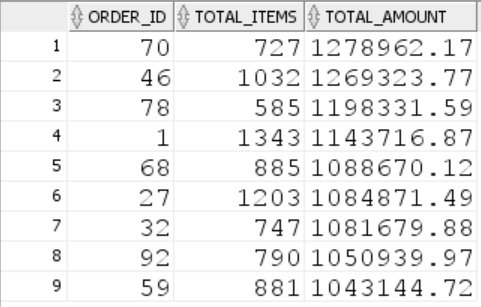
1. For each job title display the number of employees. Sort the result according to the number of employees.



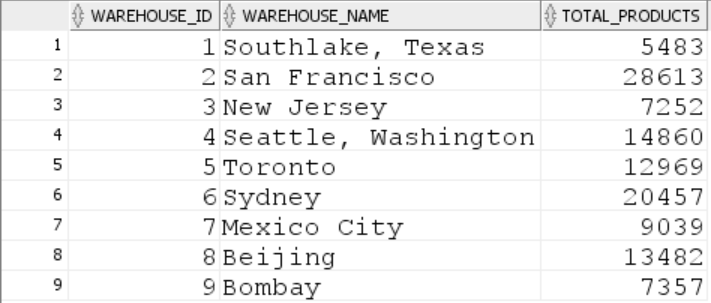
1. Display the highest, lowest, and average customer credit limits. Name these results high, low, and average. Add a column that shows the difference between the highest and the lowest credit limits named “High and Low Difference”. ***Round*** the average to 2 decimal places.



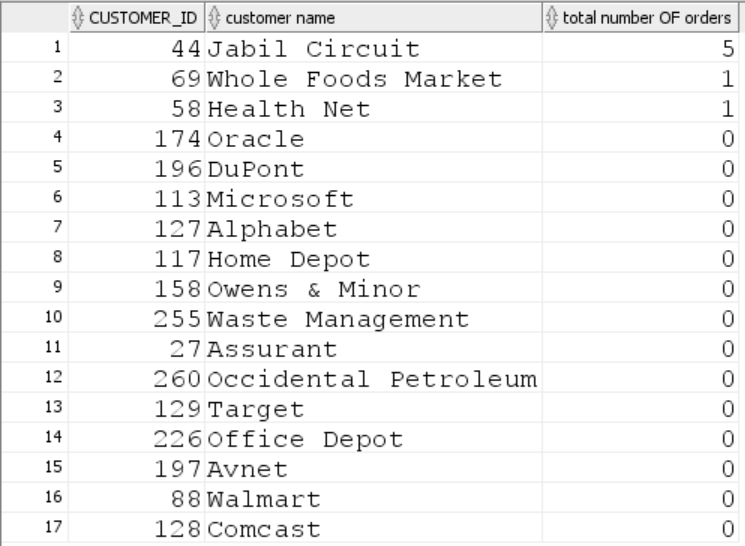
1. Display the order id, the total number of products, and the total order amount for orders with the total amount over $1,000,000. Sort the result based on total amount from the high to low values.



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



1. 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.



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

