## Lab 02 – 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 multi-row functions.

#### Submission

Your submission will consist of two files:

- (a) A single text-based SQL file with appropriate header and commenting and
- (b) An output file demonstrating that your queries work

Please ensure your SQL file runs when the entire file is executed.

Your file names should be L02-lastname-firstname

For example: L02-King-Les.sql and L02-King-Les.output or L02-King-Les.pdf, etc.

Your submission needs to be commented. At a minimum, your comments before each SQL statement should include the complete question below you are answering.

## **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)

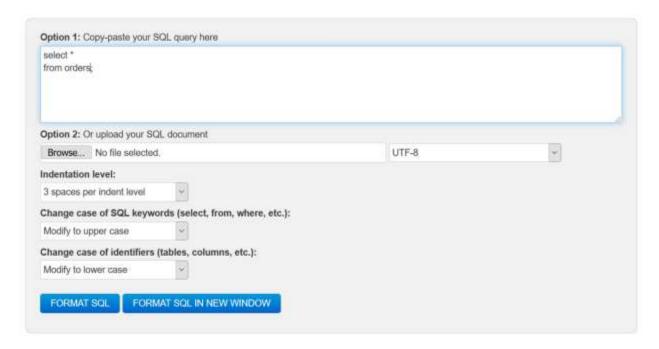
See the following sample:

```
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.



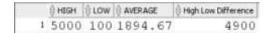
### **Tasks**

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

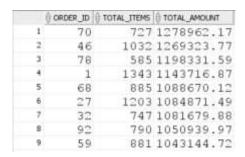
1. For each job title display the number of employees. Sort the result according to the number of employees.



 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.



3. 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.



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

| ⊕ WAR | EHOUSE_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     | 7Mexico 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                      |
|    | 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 |             | 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.

| 0.0 | ATEGORY_ID   TOTAL_AMOUNT | () AVERAGE_AMOUNT |
|-----|---------------------------|-------------------|
| 1   | 1 20041993.0              | 3 126050.27       |
| 2   | 2 13787520.6              | 2 131309.72       |
| 3   | 5 13994160.8              | 4 53209.74        |
| 9.4 | 4 4918418.1               | 5 35640.71        |

# **Example Submission**