

**CS34800: Project1**  
**Due Date: 11:59PM, Friday, February 9, 2018**

Given the following entities, provide the SQL queries corresponding to the questions below:

Note: The schema definition of these tables and sample data are provided in **tables.sql** and **data.sql**, respectively. **Do not use PL/SQL for this homework, just a main SQL select statement per question (subqueries are allowed).** You should use your Purdue Oracle account to create and test the queries. Submit your answers via Blackboard.

*Customer*(CustomerId, FirstName, LastName, Address, PhoneNo)  
*Supplier*(SupplierId, FirstName, LastName, Address, PhoneNo)  
*Product*(ProductId, ProductName, Category, SupplierId, ProductDescription)  
*Inventory*(ProductId, TotalStock, LastUpdatedOn)  
*Orders*(OrderId, CustomerId, OrderDate, Status)  
*OrderItems*(OrderId, ProductId, Quantity, UnitPrice, Discount)

Note: The sample output for each query below is for illustration only, i.e., the sample output below can be different from the output of a correct SQL query running over the sample data provided with the handout.

1. For each product, find the average discount given on that product.

Output columns: *ProductName*, *AverageDiscount*

Sort by: *AverageDiscount* in descending order

Note: The average discount should be displayed in the following format xx.xx (e.g., 80.37) and all the products should be shown. A product with no purchases should show an average discount of zero.

<i>ProductName</i>	<i>AverageDiscount</i>
<i>Product1</i>	<i>90.00</i>
<i>Product2</i>	<i>79.50</i>
<i>Product3</i>	<i>75.00</i>

2. For each phone number, say N, in the customers table, retrieve the number of customers having the same phone number N. Do not display a phone number if it is associated with only one customer (i.e., the *CountOfCustomers* column below should be greater than one).

Output columns: *PhoneNo*, *CountOfCustomers*

Sort by: *CountOfCustomers* in descending order

<i>PhoneNo</i>	<i>CountOfCustomers</i>
<i>111</i>	<i>40</i>
<i>222</i>	<i>7</i>
<i>333</i>	<i>2</i>

3. For the products without any stock, display the product id, product name as well as the suppliers' first and last names of those products.

Output columns: *ProductId, ProductName, FirstName, LastName*

Sort by: *ProductId (ascending)*

<i>ProductId</i>	<i>ProductName</i>	<i>FirstName</i>	<i>LastName</i>
<i>1</i>	<i>iPhone 7</i>	<i>Smith</i>	<i>Michael</i>
<i>2</i>	<i>iPhone 6</i>	<i>George</i>	<i>Adam</i>

4. For each customer, display the customer id, last name, and the average discount applied on the purchases of that customer. Ignore the customers who did not purchase any product. The average discount should be shown as a percentage value (i.e., do not use the Quantity and the UnitPrice attributes).

Output columns: *CustomerId, LastName, AverageDiscount*

Sort by: *CustomerId (ascending)*

<i>CustomerId</i>	<i>LastName</i>	<i>AverageDiscount</i>
<i>1</i>	<i>Michael</i>	<i>20.00</i>
<i>2</i>	<i>Adam</i>	<i>23.07</i>

5. Find the names of the three best-seller products ordered by the total number of items sold in descending order. Display also the total number of the items sold.

Output columns: *ProductName, NumberOfItemsSold*

Sort by: *NumberOfItemsSold (descending)*

<i>ProductName</i>	<i>NumberOfItemsSold</i>
<i>IPhone</i>	<i>5687</i>
<i>Mac Pro</i>	<i>4573</i>

6. Find customer phone numbers with ‘delayed’ orders (Use Attribute Status in Table Orders). Display also the customer first name and last name.

Output columns: *FirstName, LastName, PhoneNo.*

Sort by: *FirstName (ascending)*

<i>FirstName</i>	<i>LastName</i>	<i>PhoneNo</i>
<i>Adam</i>	<i>Smith</i>	<i>111</i>
<i>Monica</i>	<i>George</i>	<i>222</i>

7. Find the top four customers who generate the maximum revenue. The revenue due to a specific customer is the total money that was paid to the store by that customer. Display the revenue along with the corresponding customer id.

Output columns: *CustomerId, Revenue*

Sort by: *Revenue (descending)*

<i>CustomerId</i>	<i>Revenue</i>
<i>6</i>	<i>10990.34</i>
<i>140</i>	<i>9908.11</i>
<i>2</i>	<i>9189.09</i>
<i>4</i>	<i>4563.01</i>

8. Retrieve the top three product categories that are purchased by different customers, i.e., a customer purchasing three items from one category will be counted as one customer for that category.

Output columns: *Category, DistinctNumberOfCustomers*

Sort by: *DistinctNumberOfCustomers (descending)*

<i>Category</i>	<i>DistinctNumberOfCustomers</i>
<i>Smart phones</i>	<i>189</i>
<i>Books</i>	<i>154</i>
<i>Clothing</i>	<i>98</i>

9. Find the customers with orders of total amount exceeding \$1000, where these orders are in the 'delayed' status. Display the first names and last names of the customers as well as order ids and the total amount of the orders. The total amount should consider the discounts applied.

Output columns: *FirstName, LastName, OrderId, TotalAmount*

Sort by: *TotalAmount(descending)*

<i>FirstName</i>	<i>LastName</i>	<i>OrderId</i>	<i>TotalAmount</i>
<i>Monica</i>	<i>Witt</i>	<i>15</i>	<i>1198.91</i>
<i>Adam</i>	<i>Smith</i>	<i>130</i>	<i>1002.45</i>

10. Find the suppliers who supply at least three different products. Display the supplier ids and last names. Display also the number of the distinct supplied products. Two products are different if they have different product ids.

Output columns: *SupplierId, LastName, NumberOfDifferentProducts*

Sort by: *NumberOfDifferentProducts (descending)*

<i>SupplierId</i>	<i>LastName</i>	<i>NumberOfDifferentProducts</i>
01	Lidong	54
02	Chris	39

### Submission instructions:

Please submit via Blackboard a zip file containing the following files:

1. Your SQL script containing the 10 SQL queries. Name it **project1\_YourEmailAlias.sql**; you should put a comment before each query in the following format:

--□--□ Query1

Select.....

--□--□ Query2

Select.....

--□--□ Query10

Select.....

2. A README file containing your first name, last name, and your Purdue email address.