

**CS34800: Project1**  
**Due Date: 11:59PM, Saturday, October 20 2018**

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

Note:

1. The schema definition of these tables and sample test data are provided in **tables.sql** and **data.sql**, respectively.
2. You should finish all your work in **answer.sql**.
3. We provide **test.sh** script to test your sql queries on sample test data, but we will use different data when grading. Feel free to modify/add your sample test data in order to polish your queries. (You need to verify your answer by yourselves if you change the sample test data. In this case, test.sh may not work.)
4. How to use **test.sh**. First, set the variables “username” and “pword” in the script with your oracle account with “@csora” and your password. Second, execute the **tables.sql** and **data.sql**. Then
  - a. “./test.sh” test all 10 queries;
  - b. “./test.sh \$(Query\_number)” test one specific query. For example, “./test.sh 1”
5. Submit your answers via Blackboard.
6. **Do not use PL/SQL for this homework, just a main SQL select statement per question (subqueries are allowed).**
7. DO NOT delete/change the “-- Query[0-9]\*” comment in the **answer.sql** file.
8. Grading: We will use script to grade your projects. There will be **no** partial credit for each query.
9. For those “select top X” problems, assume that there is no tie condition that will influence the result. That is, the values in the column that may affect the result of selecting top X are distinct.
10. For the typo “PENDDING”, please accept and use it to pass the test script.

**Tables in Project1:**

*Customer (CustomerId, FirstName, LastName, Address)*

*Company (CompanyId, CompanyName, Address, State)*

*Retailer (RetailerId, RetailerName, Address, State)*

*Product (ProductId, ProductName, Category, CompanyId, ExFactoryPrice)*

*Orders (OrderId, CustomerId, RetailerId, ProductId, Count, UnitPrice, OrderDate, Status)*

*RetailerInventory (ProductId, RetailerId, TotalStock)*

(Notice: We name a table “Orders” here because we can’t name any table “Order” in Oracle.)

1. (10 points) Count numbers of products in each category.

Output columns: *Category, NumberOfProducts*

Sort by: *NumberOfProducts* in descending order

<i>Category</i>	<i>NumberOfProducts</i>
-----------------	-------------------------

2. (10 points) Calculate the average ex-factory price of products in each category

Output columns: *Category, AveragePrice*

Sort by: *AveragePrice* in descending order

Note: The *AveragePrice* should round to and keep 2 digits after decimal point. (e.g. 200.1967 should be shown as 200.20)

<i>Category</i>	<i>AveragePrice</i>
-----------------	---------------------

3. (10 pts) Display all the *ProductId* and *ProductName* of products produced by companies in CA or IN

Output columns: *ProductId, ProductName*

Sort by: *ProductId* in ascending order

<i>ProductId</i>	<i>ProductName</i>
------------------	--------------------

4. (10 pts) Calculate how many more products retailers need to prepare. That is, for each retailer, if any product has  $\$(TotalStock) < \$(sum\ of\ all\ counts\ for\ a\ product\ in\ Orders\ table\ with\ status\ "PENDING")$ , the retailer needs to prepare more of that product. Display the *RetailerId*, *RetailerName*, *ProductId* and *PrepareNumber* (the number of products to be prepared). Assume that all the product-retailer pairs in *Orders* table can be found in *inventory* table.

Output columns: *RetailerId, ProductId and PrepareNumber*

Sort by: *RetailerId* in ascending order

<i>RetailerId</i>	<i>ProductId</i>	<i>PrepareNumber</i>
-------------------	------------------	----------------------

5. (10 pts) Find the names of the top 3 best-seller products ordered by the total number of items sold in descending order (include all states). Display also the total number of the items sold.

Output columns: *ProductName, NumberOfItemsSold*

Sort by: *NumberOfItemsSold* in descending order

<i>ProductName</i>	<i>NumberOfItemsSold</i>
--------------------	--------------------------

6. (10 pts) Find the top 3 customers that spend most money in our data. Display the FirstName, LastName and TotalAmount

Output columns: *FirstName, LastName, TotalAmount*

Sort by: *TotalAmount* in descending order

Note: The *TotalAmount* should round to and keep 2 digits after decimal point as Q2

<i>FirstName</i>	<i>LastName</i>	<i>TotalAmount</i>
------------------	-----------------	--------------------

7. (10 pts) Find the top 2 retailers that received the most orders from customers.

Display the RetailerId, RetailerName and NumberOfOrders

Output columns: *RetailerId, RetailerName, NumberOfOrders*

Sort by: *NumberOfOrders* in descending order

<i>RetailerId</i>	<i>RetailerName</i>	<i>NumberOfOrders</i>
-------------------	---------------------	-----------------------

8. (10 pts) Find the top 2 retailers that have the highest total net profit. Display the RetailerId, RetailerName and the NetProfit. (For simplicity, we define the net profit per piece of product as the UnitPrice – ExFactoryPrice)

Output columns: *RetailerId, RetailerName, NetProfit*

Sort by: *NetProfit* in descending order

Note: The *NetProfit* should round to and keep 2 digits after decimal point as Q2

<i>RetailerId</i>	<i>RetailerName</i>	<i>NetProfit</i>
-------------------	---------------------	------------------

9. (10 pts) Find the top 3 companies that sell most products (Measured by *Count* in Orders table. A company can have multiple products). Display the *CompanyId, CompanyName* and *TotalCount*

Output columns: *CompanyId, CompanyName, TotalCount*

Sort by: *TotalCount* in descending order

<i>CompanyId</i>	<i>CompanyName</i>	<i>TotalCount</i>
------------------	--------------------	-------------------

10. (10 pts) Find the top 2 retailers visited by most distinct customers. (Note: one customer can appear multiple times in Orders table with same retailer. In this case, we only count this customer once for this retailer.) Display the RetailerId, RetailerName and NumberOfDistinctCustomers

Output columns: *RetailerId, RetailerName, NumberOfDistinctCustomers*

Sort by: *NumberOfDistinctCustomers* in descending order

<i>RetailerId</i>	<i>RetailerName</i>	<i>NumberOfDistinctCustomers</i>
-------------------	---------------------	----------------------------------

**Submission instructions:**

Please submit via Blackboard the following:

Your SQL script (**answer.sql**). It should contain the 10 SQL queries and look like the following:

```
-- Query1  
Select.....  
  
.....  
-- Query10  
Select.....
```

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