Homework 1 : SQL and MySQL

Getting Started

In this lab you will create and query a simple SQL database. Please download and install MySQl on your computer (or you may use an online MySQL platform such as https://paiza.io). Using your favorite editor, create a file named **`hw1.sql`**; this will be the only file you submit.

Description

You are assigned to help an e-commercial company to maintain online shopping records using a database.

Part 1:
You need create 6 tables, including any appropriate primary/foreign key constraints. The schemas of the tables are as follows.

Customer(C_Id: number, C_Name: string, Credit_Limit:number, Income_Level:char, Gender:char)
In table Customer, C_Id is the id of the customer, C_name is the name of the customer. There are three possible income levels recorded as: Low(L), Median(M), High(H). In attribute *Gender*, we use F for female, and M for male.

Inventories(Product_Id: number, Warehouse_Id:number, Quantity_On_Hand:number)

Orders(Order_Id:number, Order_Data:date, C_Id:number, Order_Status:string)
In Orders, there are two possible statuses of Order_Status: Complete and Processing. A "Processing" order means that the order is placed online, but the products have not yet been shipped to the customer.

OrderItems(Order_Id:number, Product_Id:number, Unit_Price:number, Quantity:number)
Table OrderItems records the products in each order. Here, Unit_Price is the price paid by the customers.

ProductInformation(Product_Id:number, Warranty_Period:number,Purchased_Price:numer)
Table ProductInformation records the information of products. Warranty_Period is in number of days. For simplicity's sake, here we assume that the warranty starts from the date the order is placed. Purchased_Price is the price paid when the product is bought.

WareHouses(Warehouse_Id:number, Location:string)

(FYI: If you install MySQL on your computer, run "Create Table …" at MySQL prompt and see errors like " No database selected", then you will need create a database first and use it. E.g.

```
mysql > create database db1;
mysql > use db1;
)
```

After creating the tables, please insert records into the tables so that the tables are as follow:

Table1. Customers

C_Id	C__Name	Credit_Limit	Income_Level	Gender
1	Jamas	1000	L	F
2	Christan	2000	M	M
3	Saywer	5000	H	F
4	Kropy	5000	H	M
5	Lock	2000	M	M
6	Mando	1000	L	F

Table2. Inventories

Product_Id	Warehouse_Id	Quantity_On_Hand
1	1	14
2	1	25
3	3	9
4	2	67
5	2	50
6	1	9
7	3	90
8	3	20
9	3	35

Table3. Orders

Order_Id	Order_Date	C_Id	Order_Status
1	2019-09-01	1	Processing
2	2019-08-27	2	Complete
3	2019-06-20	3	Complete
4	2019-08-01	4	Complete
5	2019-08-31	1	Processing
6	2019-09-01	4	Processing
7	2019-08-20	6	Complete
8	2019-08-11	2	Complete

Table4. OrderItems

Order_Id	Product_Id	Unit_Price	Quantity
1	1	80	1
1	2	90	1

2	1	70	1
3	4	20	2
3	2	100	3
3	8	160	1
4	4	20	12
5	2	100	2
5	8	250	1
6	9	160	10
7	5	100	3
7	7	250	1
8	4	35	2

Table5. ProductInformation

Product_Id	Warranty_Period	Purchased_Price
1	90	60
2	120	80
3	365	100
4	30	15
5	365	70
6	90	300
7	60	200
8	90	150
9	120	100

Table6. WareHouses

Warehouse_Id	Location
1	Los Angles
2	Chicago
3	New York

Part 2:

Write the following queries in SQL. For each question, you can use multiple SQL statements if needed:

1) Return the names of customers whose income level are marked as Median(M), and return the average credit limit for these customers.
2) Return the name of the customer who has placed more orders than other customers; return the name of the customer who has bought more items than any other customers (Including all processing and complete orders)
3) Return the Location of the warehouse which has the largest number of products in stock.
4) Return the name of the female customer who has ever placed an order (orders) containing at least 3 items.
5) Return the sale revenue of all complete orders; return the profit for all complete orders. ** *sale revenue* is defined as the amount of the price of all sold items; *profit* is defined as the sale

price minus the purchasing_price of all sold items.

6) Return the names of the customers who have placed orders valued more than half of his/her credit limit.
7) List the <order_id, c_id, order_data, warranty_period> for all the sold products in completed orders whose warranties have already expired.
8) Return the names of the customers who have placed at least 2 orders in August 2019. (Including all processing and completed orders)
9) List the names of the customers who have placed at least one order including items from at least 2 different warehouses.
10) Return the names of the customers whose orders never have any item from the warehouse located at "New York".
11) Which item has the largest sale quantity for the customers of each income level?
12) Which item does the largest number of customers buy? Return the Id of the item.

Part 3:

Do the following updates using SQL . For each question, you can use multiple SQL statements if needed:

1. Decrease the credit limit of all female customers by 50%.
2. If there exist any "processing" orders that exceed the customer's credit limits, delete such orders.
3. Change the status of all "processing" orders to "Complete" and update the ***inventories*** table.
4. We decide not selling product with product_id = 3 anymore, please delete all the related information of this product in all the relevant tables.
5. Increase the warranty period by 30 days for all the products stored in warehouse located at "Chicago".

Part 4:

Congratulations! You have finished all your work and please write SQL code to **Drop** all tables you've created.

Submission

You only need submit the script 'hw1.sql' . Before your final submission, make sure the file is executable, and all your results are correct . Comment your code; mark each query with its number. To run all the code in your hw1.sql file, use the following command at MySQL prompt:

```
mysql > source C:\your directory\hw1.sql
```