CSC 4710/6710 - HOMEWORK ASSIGNMENT 3

Please, read ENTIRE document before starting your work on this assignment.

The Order-Processing Database. Quite often a computer programmer has to access and query a database, without being given its detailed documentation. This exercise is prepared to model such scenario.

The hw3_ordersdb.sql file, available in iCollege, contains SQL table creation and data insertion commands. You should be able to quickly create the tables from the file and insert the data. The database, created for the purpose of this assignment, contains data that supports a simple order processing application for a small distribution company. It consists of five tables:

1. The CUSTOMERS table stores data about each customer, such as the company name, credit limit, and the salesperson who usually calls on the customer (i.e., the customer representative).
2. The SALESREP table stores the employee number, his/her name, age, direct supervisor (denoted as manager), year-to-date sales and other data about each salesperson. Note: Managers included in SALESREP table are direct supervisors of sales representatives - not all of them are/must be office managers.
3. The OFFICES table stores data about each of the five sales offices including the city where the office is located, the sales region to which it belongs, office manager's ID (denoted as MGR), and so on.
4. The ORDERS table keeps track of every order placed by a customer, identifying the salesperson who took the order (not necessarily the salesperson who calls on the customer), the product ordered, the ordered quantity and the total value (i.e. amount) of the order (in USD), and so on. For simplicity, each order is for only one product.
5. The PRODUCTS table stores data about each product available for sale, such as the manufacturer, product number, description, and price.

I would recommend to put substantial effort into reverse engineering the relations between the tables (from my SQL code, or simply by looking at the attributes of the created database), before you start writing your own SQL commands. You may want to look at the actual data to confirm your expectations. Even if it might seem to be time consuming at the beginning, it will save you a lot of time (and missed points) at the end.

THE TASK: Use only SQL queries (i.e., SELECT statements) to provide full information to the requests specified on the next page.

NOTE: Unless otherwise instructed, you should assume the query is asking for the **names**, and **not the id numbers** of customers, people, product or city offices; rename attributes if the meaning of the resultant table is not clear. Common issue when working with databases is preserving people's privacy-- I want you to print out ONLY the output that has been requested. Printing out unnecessary attributes will be interpreted as an error.

You should prepare two files for this assignment: (1) report and (2) SQL queries.

(1) Report includes the question number and question itself, SQL query, and the output of query. Let us assume that **Question 0** is following: **"List the names and titles of sales representatives"**. Your answer for this question should look like:

Question 0: List the names and titles of sales representatives.

SQL Query

```
select name, title
from SALESREPS;
```

Output:

```
name              title
Dan Roberts       Sales Rep
Sue Smith         Sales Rep
Paul Cruz         Sales Rep
Bob Smith         Sales Mrg
Bill Adams        Sales Rep
Sam Clark         Vp Sales
Nancy Angelli     Sales Rep
Larry Fitch       Sales Mrg
Mary Jones        Sales Rep
Tom Snyder        Sales Rep
```

(2) The SQL query file includes the queries and necessary comments. Put the question number for every query as comment.

```
# Question 0
select     name, title
from       SALESREPS;
# necessary comments…
```

Put all your queries **in sorted order** *with necessary comments* to a simple text file ([initials]_HW3_queries.txt, *e.g., BA_HW3_queries.txt*). Put your answers in your report as

described above **in sorted order** and send it as PDF ([initials]_HW3_report.pdf, *e.g., BA_HW3_report.pdf*).

For each question below, each SQL query needs to be immediately followed by ALL the output the command generated. Please, make sure to number your queries appropriately and follow the order presented below. Do not get screenshots, copy and paste all of the output as text. Correctly format your queries and your answers.

Some penalties for not sticking to these rules: (1) Using SQL commands that do not start with keyword SELECT (e.g. using a SQL command starting with CREATE TABLE/VIEW…) = 100% of your score subtracted; (2) SQL queries without results = 50% of your score subtracted; (3) Outputs only, but no SQL queries = 80% of your score subtracted; (4) Ordering of your responses does not match order of the homework questions, or not formatting, or not commenting = at least 20% of your score subtracted.

Here are the queries I want you to write in SQL and run on the above-mentioned database:

1. Show the name, age, sales and quota of the sales representative whose last name ends with letter "s".
2. List the customer company names and product descriptions of all the products each customer/company has ordered. Arrange the output ascending by the company name.
3. Show the total value of the inventory on hand for each product. Arrange in descending order by total value.
4. How many customers are there?
5. List the cities where the local offices have their targets less than $550,000.
6. List order numbers and quantities for all the orders that are over $20,000; include also the name of the salesperson who took the order and the name of the company (i.e. customer) who placed it.
7. List all the companies which have ordered any size widget (i.e., Size 1, 2, and 3 widgets), and the widget they ordered. Make sure you print out only unique pairs of attribute values.
8. List the office, city, region and amount that sales are over (or under) target for each office (if sales are over the target the number needs to be positive, if under – I want to see a negative number).
9. Are there any customers who are over their credit limit? If so, list the customer, the total amount the customer has on order, the credit limit, and the difference between total amount and credit limit.

10. What is the total order amount for each salesperson? Order output by decreasing total order amount; do not print the same names multiple times.
11. What is the total amount (i.e. value!) of orders for each salesperson whose orders total for more than $40,000? Order output by amounts, in decreasing manner.
12. List the offices and the target amounts for every office where the target for the office exceeds the sum of the individual salespeople's quotas.
13. List the salespeople whose quotas are equal to or higher than the target of the Denver sales office (note: you are not allowed to just write SQL command with "office=22" explicitly, you must use word "Denver" somewhere in your command).
14. List the names of companies who placed an order with a sales representative that is not the sales representative that usually calls on them (i.e. he/she is not specified in an appropriate record of the CUSTOMER table, as the regular sales representative for this client/company). Include also the names of these salesreps, indicating in attribute TEMPORARY_SALES_REP name of salesrep, who took the order.
15. Reverse engineer the relational design, and identify the foreign key relationships among the tables. Using alter table statements, add these constraints to your tables.

Additional info about the MySQL DBMS and homework you may find useful:

- Questions 1-14 are 5 points. Question 15 is 10 points (bonus). The homework is graded over 70 points.
- You are expected to use MySQL DBMS for this assignment. Use versions over 5.5.
- You will not get any points if you use another DBMS.
- InnoDB is the default storage engine for MySQL database management system after the version 5.5. If you are using MyISAM or some other storage engine, change it, so you can do the 15th question.
- If you cannot run MySQL, please contact the TA as soon as possible.
- Try to stick with the SQL standards, so TA can easily understand what you are doing.
- We will run your queries on our own local server, in case it does not match your output; you will be under inspection for plagiarism. DO NOT USE others' work. If you have questions, ask it to TA or me. Do not share ideas or the ways to do it. Do not do the studying part together, if you believe your answers will be similar. Do not "accidentally" send your homework to your peer(s). In case we detect plagiarism, **all parties will be responsible**. These persons will directly get an F and be reported to dean's office.