CSCI 6622: Database Systems | Fall 2019
Course Project
Due 12/03/19

Instructions

- This is a project that must be completed by each individual in the class; group work is <u>NOT</u> permitted.

- The project will be considered one week late if handed in by 12/10/19; it will be considered two weeks late if handed in by 12/17/19. ***PROJECTS HANDED IN AFTER 12/17/19 WILL NOT BE ACCEPTED***.

- All materials must be submitted via Blackboard per the instructions listed below. ***You do <u>NOT</u> have to turn in a printed copy of any materials for this project.***

- The database you create must be implemented in MySQL. Databases created in other database management systems will not be accepted.

Computer Maintenance Service Database

Your task is to create a relational database for a computer maintenance service. The service sends technicians out to businesses to maintain their computer systems or fix them when they have problems. The company's services are broken down into the following three categories: hardware services (e.g., hard drive replacement, RAM installation, monitor repair, etc.); software services (e.g., operating system upgrades, software installation, etc.); and disaster recovery (virus/spyware removal, data recovery, etc.). Each service is charged at a specific hourly rate. Each customer is identified by a unique customer ID number and the database stores the business name, street address, city, state, ZIP code, phone number, and the first name, last name and email address of the primary contact at the business. All services are delivered at appointments set for a specific date and time. Each appointment may involve one or more repair services being rendered to exactly one customer (for example, a customer can have RAM installed in three computers and have a virus removed from one computer in the same appointment). The amount of time required for each service is tracked in hours; hours can be billed in quarter-hour increments (for example, installing a hard drive can be billed at 0.75 hours to represent 45 minutes of time). Each appointment is serviced by a technician who works for the company. Each technician is identified by a unique technician ID number and the database stores the first and last name of each technician, along with his or her mobile phone number and the date the technician was hired. Each appointment is assigned exactly one technician; a technician may have many appointments. All prices in the database are tracked in US dollars ($). You can assume that all customers are located in the United States.

When you create this database in MySQL, you will populate it with a relational instance that contains sufficient data to answer the following queries:

1. List all services offered by the company, sorted alphabetically by category and then alphabetically by the service name.

2. List the names of all customers who had hard drives replaced in the last 30 days.

3. List the names of all customers who bought both hardware and software services (either in the same appointment or separate appointments).

4. List all technicians who have performed operating system upgrades and the total number of hours each technician spent on those upgrades.

5. List all customers who bought data recovery services but did not buy any other item.

6. List the names of the customer (or customers) who spent the greatest dollar amount in a single appointment.

7. List each category, the name of the customer who has spent the most on services in that category, and the total amount of money that customer spent *in that category*.

8. List the names of each technician, along with the number of hours the technician has spent on appointments and the amount of money billed for the technician's hours.

9. List the name and ZIP code for all customers who have spent more than $500 (this could be spent over several different appointments, not just a single appointment).

10. Show the ZIP code where the largest number of appointments were (i.e., the highest count of appointments for customers located in a specific ZIP code).

You need to turn in the following materials on Blackboard for this project:

1. You must submit a word processing document that contains the following items:

    a. A description of any assumptions you're making in the design of this database.

    b. An Entity-Relationship (ER) diagram of your database using the format shown in class. This should be either created in a graphics program or neatly drawn on paper and photographed. All graphics should be inserted into your word-processing document (not attached separately).

    c. A relational schema with all primary and foreign keys indicated in the manner of Elmasri Figure 5.7.

2. You must submit a MySQL dump file (.sql file) that will import the database and relational instance you created for this project. The dump file should follow these guidelines:

    a. The database you create should be named as follows:
        [your UNH username]_project. For example, a student with the UNH username jsmith2 would name the database as ***jsmith2_project***. Your dump file should create a database with this name when it is imported.

    b. The dump file should be created as a self-contained file with the following name: [your UNH username]_dump.sql. For example, a student with the UNH username jsmith2 would name the dump file as ***jsmith2_dump.sql***. This dump file should create the entire database schema and populate it with data from the relational instance you design to illustrate your queries.

    c. The database created from this dump file should not create any views (just tables and data).

3. You must submit a script file (.sql file) that executes the 10 queries described above. The script file you submit should follow these guidelines:

    a. Your script file should be named as follows: [your UNH username]_queries.sql. For example, a student with the UNH username jsmith2 would name the script file as ***jsmith2_queries.sql***.

    b. Your script file should include the necessary syntax to ensure that it runs queries against the specific database you created for this project (for example, a database named *jsmith2_project*).

    c. Please ensure the relational instance you provide in the MySQL dump file contains enough data to answer all 10 queries. In particular, none of your queries should return an empty table when executed.

    d. You may use views or temporary tables to answer your queries. If you do this, your script file must contain the proper SQL syntax for creating these.

PLEASE NOTE: As with all assignments where you are turning in files to create databases and execute queries on them, I strongly encourage you to test your database import and script file several times to make sure everything works as expected before you submit materials for your project. If your submitted database fails to import or the queries in your script file fail to execute, you will lose points off your final grade.