CPSC 2221 - Database Systems

Group Project - Implementation of a Relational Database

Total Marks: 100

REQUIREMENTS OF THE PROJECT

This course includes an obligatory group project. The project will be completed in groups of **four.** The aim of the project is to apply your skills to build a database application from start to finish to present a real database driven solution for a real-life business. Furthermore, students are required to use all the concepts that they will learn while taking the course. The primary emphasis in class is teaching database concepts with some hands-on instructions. You will be graded based on the result of a demo, a group report, a personal report, and meeting of intermediate milestones. The project will either be conducted using Java/JDBC or using web design and PHP. The grade of the final project will be based on the rubric mentioned on the last page of this document. All the deliverables must be submitted with a cover page. The template of a cover page is posted on the Piazza.

The milestones will be submitted through Brightspace. You can use Piazza to report/solve problems among the members.

GOALS OF THE PROJECT

- Deciding on an application for which database systems would be required
- Defining the application functionalities
- Designing and implementing the schema
- Populating the database
- Writing the code required to embed the database system in an application having a Graphical User Interface (GUI).

Focus on the database aspects and the design of the schemas, as opposed to spending extensive time on sample data and the GUI.

Avoid the examples provided in the textbook, course notes, or lectures. Don't use any examples that you've worked with before. It can't be something from another course, the Internet, your friend's work from a previous offering of the course, etc. It must be your team's original work.

My Advice: Please read the instructions carefully to make sure that you don't miss anything.

Start early; don't wait for the few days before the deadline to get started.

The project will have **five** milestones:

MILESTONE 1 - PROJECT PROPOSAL

Marks: 10

The goal of the project is to allow you to have the freedom to design your application. However, for those who want a bit more of an idea of what's an acceptable project, I offer the following rough guidelines. I expect that each application should eventually have:

- At least 5 entities and 5 relationships
- The ability to handle multiple classes of users (eg. customers and bank employees)
- At least 10 queries that users will be able to ask via the application interface.

What to submit?

Each group must submit a document, namely, *project title_PP.docx* with the following information:

- A cover page
- Description of Project : This should include the information about the domain and application specifications (i.e., what functionality will the system provide).
- Specify the platforms that you plan for the project (i.e., MySQL, PHP). You are allowed to use any programming language or relational database, as long as you do all programming and query writing yourself (e.g., you can't use a platform that's going to write your SQL for you). ***I am also providing some documentation using PHP and MySQL.***
- Any other assumption that you are considering.

MILESTONE 2 - ER DIAGRAMS AND SCHEMA

Marks: 20

The aim is to do modeling using ER diagram and Schema for your database. While working, you may go through several refinements. But, you must submit only the final version. Each group must provide following:

- A cover page
- ER diagram – No handwritten diagrams will be accepted. For ER diagrams, you can use Gliffy, Microsoft Visio). Your ERD should include:
    - At least 5 entity sets. There must be at least a weak entity, self referencing entity and isA hierarchy. For each entity set:
        - Identify Primary and Candidate keys.
        - At least 5 relationships. For each relationship, identify the cardinality constraint and other constraints, such as participation constraints.

- The schema derived by translating your ERD into relational Model. For each table:
  - List the table definition e.g., Table1(attr1: domain1, attr2: domain2…)
  - Specify the primary key, foreign keys, and other constraints that the table must maintain

MILESTONE 3 – FD and Normalization

Marks: 20

Each group must provide following:

- A cover page
- Functional Dependencies
  - List all functional dependencies that are applicable to the table.
  - List all candidate keys
- Normalization
  - Normalize each of your tables, if necessary, to be in 3NF or BCNF. Give the list of tables, their primary keys, and foreign keys after normalization.
- The SQL DDL to create all the tables in SQL. All primary keys and foreign keys must be declared appropriately.
- Populate each table with at least 5 tuples. Show the instance of each relation after inserting the tuples.

Note : There must be some kind valid FDs in at least two relations other than PK and CK. In case, you observe that no relations have FDs then add some attributes intentionally to show FD. It is must that your schema go through the normalization process.

MILESTONE 4– COMPLETE PROJECT FILES

Marks: 40

Each group must submit following:

- A cover page
- All the code used in the application and step by step instructions on how to install your application. This is required so that Marker can install and confirm the working of your application.
- Script that could be used to create all tables and data in the database
- A short description of what the project accomplished
- A list of the SQL queries used

MILESTONE 5– IN CLASS DEMOS

Marks: 10

Each group must present the developed application in 15- 20 minutes during the class. This presentation should explain and summarize the application. Each group will have 15- 20 minutes. During the presentation, you must show the following queries:

1. **[1 Mark]** *Projection query*: Create one query of this category and provide an interface for the user to specify the projection conditions to be returned.
   Example:
   SELECT Field_01
   FROM Table_01

2. **[1 Mark]** *Selection query* : Create one query of this category and provide an interface for the user to specify the selection conditions to be returned. Example:
   SELECT Field_01
   FROM Table_01
   WHERE Field_02 >= 0

3. **[1 Mark]** *Join query:* Pick one query of this category, which joins at least two tables and performs a meaningful query, and provide an interface for the user to choose this query (e.g. join the Customers and the Transactions table to find the phone numbers of all customers who has purchased a specific item).

4. **[2 Mark]** *Division query:* Pick one query of this category and provide an interface for the user to choose this query (e.g. find all the customers who bought all the items).

5. **[2 Mark]** *Aggregation query:* Pick two queries that require the use of distinct aggregation (min, max, average, or count are all fine).

6. **[1 Mark]** *Nested aggregation with group-by:* Pick one query that finds some aggregated value for each group (e.g. the average number of items purchased per customer).

7. **[1 Marks]** *Delete operation:* Implement a cascade-on-delete situation. Provide an interface for the user to specify some input for the deletion operation. Based on input, deletion should be performed.

8. **[1 Marks]** *Update Operation:* Provide an interface for the user to specify some input for the update operation.

9. **[3 Bonus Marks]** *Extra features:* Create unique features that your application supports. Some examples are using Bootstrap, implementing Triggers and Privileges in the database.