INFS7901 Database Principles
Project

Project Description

Each student is required to select an application that would benefit from a database and build a database application from start to finish. The web user interface is not required to cover the whole application. It should only contain a search, a delete, an insert, and an update operation (totally 4 forms).

Philosophy

Building a full database application from scratch allows you to control the process; instead of having the pieces decided for you, you must make all of the decisions by yourself. Part of this process is that you will see how design decisions made at the beginning will affect your final project. We'll have some practice of some of the skills that you'll need in labs, but this will allow you to get some more in-depth practice and a feel for what it's like when you make all of the decisions for an application.

Goals

- Deciding on an application for which database systems would be required
- Modelling the domain of the application, and defining the application functionalities
- Designing and implementing the schema
- Populating the database (this should not be the main focus of the project)
- Writing code needed to embed the database system in an application that has a minimal Graphical User Interface (GUI).

Schedule

There are a number of intermediate deadlines that you must meet in order to ensure a successful project. Descriptions for each of the required checkpoints will be described below

- 7 April: Project proposal (15% of the project grade)
- 7 April: Formal specification (35% of the project grade)

  Note: In reality you first write the project proposal, and later on write the formal specifications. It means that your design in proposal is a preliminary version of your design in formal specification. In this subject we had to postpose the deadline for project proposal, hence, you submit the two documents together. Ideally, you first write the proposal, and then the formal specification.

- May 25: Completed project (50% of the project grade)

Part 1: Project proposal

The goal of the project is to allow you to have the freedom to design your own application. A consequence of this is that we won't be telling you exactly what to do. Since we don't expect you to read our minds, this is one purpose of the checkpoints - we give you feedback on the project and tell you if you're on track to do well and if not how to get there, without penalising you heavily for not being able to read our minds. However, for those who want a bit more of an idea of what's an acceptable project, we offer the following rough guidelines. We expect that each application should eventually have:

- At least 5 entities of which at least 1 of is a weak entity.
- At least 5 relationships of which at least 1of is a Generalization/Specialization relationship.
- At least 10 queries (including selection, aggregation, nested, division, deletion and update queries) that users will be able to ask via the application interface (you only need to implement the web interface for 4 of these queries).

What to hand in and marking scheme

One pdf file that includes:

- (10 marks) Project description
    1. What is the domain, what aspects of the domain will be modeled by the database?
    2. What are the application specifications? (i.e., what functionality will the system provide)
    3. What platforms do you plan for the final project be on (i.e., MySQL, Python). Note that you are allowed to use any programming language or relational database that you please, as long as you create an application for a relational database, and that you meet all of the other requirements, and 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). We'll suggest and mostly support MySQL + Python.

- (10 marks) An E/R diagram for the database the application will use. For each entity set, identify candidate keys, and the primary key. For each relationship, identify the cardinality constraint and other constraints, such as participation constraints.

Part 2: Formal specifications

The goal of this checkpoint is for you to tell us exactly what to expect from your final application. Note that we expect your final project to have some sort of rudimentary GUI - but it doesn't have to be fancy.

What to hand in and marking scheme

One pdf that includes:

1. (5 marks) A brief **description of your project** and its main functionality.

2. (10 marks) **An updated E/R diagram (if it has changed from the initial E/R in Project Proposal)** for your project.

3. (10 marks) The **schema** derived by Mapping your ER diagram. For each table List the table definition e.g., Table1(attr1: domain1, attr2: domain2...) and specify the primary key, foreign keys, and other constraints that the table has to maintain

4. (5 marks) List of **functional dependencies** that are applicable to the table (including the ones involving the primary key). For each functional dependency, briefly describe its meaning in English.

5. (5 marks) The **normalised schema** which is in 3NF or BCNF. Give the list of tables, their primary keys, and foreign keys after normalization.

6. (10 marks) The **SQL dump** that creates all the tables in SQL. All primary keys and foreign keys must be declared appropriately. Populate each table with at least 5 tuples.

7. (5 marks) A **screenshot** of your platform performing one meaningful query. Include a brief description of the picture and the query. This is to make sure that you start the implementation of your project early in the semester.

Part 3: Completed Project

This is the main part of the project, building on what you have accomplished in the other project checkpoints in order to complete, demo and evaluate the project.

What to hand in and marking scheme

There are three components to be handed in:

1. **Code:** Project's code base and related scripts are submitted through Blackboard as one Zip file. Make sure that you hand in:
   - (10 points) All the code used in the application.
   - (10 points) A script that could be used to create all tables and data in the database

2. **Report:** Project report in PDF format, submitted through Blackboard. It should include the following items:
   - (5) Aim of the project in a paragraph

   - (20 points) **Queries:** Show the SQL code, the function (what the query does for the user within your application), and a pictorial UI (it is just a picture, don't have to implement it) using at least 4 of the following queries (each 5 point):
     a) join query
     b) division query
     c) aggregation query (function such as min, max, avg, or count)
     d) delete operation with cascade
     e) update operation
     f) Use of SQL Assertions and/or triggers functionalities in your project

   - (20 points) **GUI:** Show SQL code, the function, and the implemented web GUI for 4 forms within your project. This should include:
     a) GUI using select query
     b) GUI using update query
     c) GUI using delete query
     d) GUI using insert query

   - (5 points) **Your assessment of the Project:**
     a. Roughly how much time did you spend on the project?
     b. What did you like the best and least about the project?
     c. What helped you learn the best in the project?
     d. If you could change the way that the project was organized, what would you change?