INFS7901 Database Principles
Project

Project Description

Students in **teams of 4** are required to select an application that would benefit from a database and build a database application from start to finish.

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 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

- March 17: Project proposal (10% of the project grade)
- April 14: Formal specification (20% of the project grade)
- May 26: Completed project (50% of the project grade)
- May 26: Class Presentation (20% of the project grade)

Part 1: Project proposal (one report per group)

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.

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 (one report per group)

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** 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 a 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 (group code + group report + individual report)

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. **Project Report**: To evaluate your project better, we will ask you to create a RiPPLE learning resource reporting the outcome of your project. The learning resource should start with a few sentences describing the aim of the project. It should then include a video to showcase the capabilities of your project. The video is to be uploaded to any video-sharing website such as YouTube or Vimeo. You can then include the video in your learning resource on RiPPLE. Make sure that the video is less than 10 minutes and demonstrates the following in the context of your project:

     - **Queries:** Demo a meaningful feature within the context of your application that uses each of the following types of queries.
       - (10 points) Join query
       - (10 points) Division query
       - (10 points) Aggregation query (functions such as min, max, average or count)
       - (20 points) Aggregation with group-by (aggregated value for each group)
       - (10 points) Delete operation with Cascade
       - (10 points) Update operation:

   - **Usability features**
     - (20 points) Use of a Responsive Web Design for your front-end implementation.
     - (20 points) Use of SQL Assertions and/or triggers functionalities in your project.

   - **Competencies**
     - (10 points) A broad understanding of the topics covered in this course
     - (20 points) Effective communication skills in presenting the outcomes of your project
     - (10 points) Effective interaction between team members

3. **The individual reports**: Your individual reports are submitted through Blackboard and will remain confidential and will not be shared with your group member(s). For each of the following four bullets, consider writing roughly one paragraph.

- (5 points) **Final Achievement Assessment**:
    - Did you fall short or exceed your initial expectations?
    - How well did you work together and what areas could have been improved?
    - Comment on how group roles and responsibilities were defined

- (5 points) **Group Assessment:**
    - for each partner, evaluate performance, effort, accomplishments on a scale of 1-5 where 1 is "met them once, I think, they weren't around" and 5 is "captain, CEO, chief-architect, cheer-leader, all around good person and the project would have been significantly less without them". Please be sure to justify, briefly, your assessment/evaluation.

- (5 points) **Self-Assessment:**
    - For yourself, evaluate performance, effort, accomplishments on a scale of 1-5 similar to how you evaluated your partners. justify, briefly, your assessment/evaluation.
    - What were your contributions in terms of code, design, group moral, etc. Specifically, what did you do for the group and for yourself?
    - What aspects, if any, of your effort and work are you most proud of? What aspects, if any, are most in need of improvement?
    - What do you think your fellow team members will say about your achievements and efforts?

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

Part 4: Class Presentation (group video)

We will ask you to deliver a 6-min presentation of your project in class to your peers. This will be a recorded video that will be played in class. The presentation should provide a broad overview of your project and highlight the most impressive features of your project.

Class Presentation's marking scheme

- (20 points) Content of the presentation
- (10 points) Visual appeal of the presentation
- (10 points) Comprehension / knowledge of topic
- (10 points) Presentation skills
- (10 points) Group dynamics during the presentation
- (10 points) Time management