INFS7901 Database Principles
Project Guidelines

Project Description

In this project, 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 mandatory for this project, but can play an important role if you would like to obtain bonus points. The project requires two deliverables, where marking schemes are provided in both parts and bonus points are explained in Part 2.

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 your own decisions at the beginning and can experience how they will affect the execution of your project. We will prepare practical sessions for the key skills needed, but this project will further facilitate more in-depth practices and take you on a tour of a database architect's responsibilities when designing 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;
- Writing code needed to embed the database system in an application;
- (Optional) Building a database application with Graphical User Interface (GUI).

For time management, please note that populating the database is only for ensuring necessary query operations and is not the main focus of the project.

Schedule

There are two important deadlines that you must meet in order to ensure a successful project:

- 17 April: Project proposal (40% of the project grade)
- 25 May: Completed project (60% 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. With the project proposal, we can give you feedback on the project and tell you if you are on the right track – and suggest improvements if not. As an overarching picture, we expect that each project should eventually have:

- At least five entities, where at least one of them is a weak entity.
- At least five relationships, where at least one of them is for generalization/specialization.
- At least ten queries (including selection, aggregation, nested, division, deletion and update queries) that the database users will be able to operate with.

Marking Scheme (In 100-point Scale)

The goal of this milestone is for you to tell us exactly what to expect from your final application. Please submit one PDF file that includes the following information.

1. (15 marks) A project description that explains:

- What is the domain, what aspects of the domain will be modeled by the database?
- What are the application specifications, i.e., what functionalities will the system provide?
- What platforms do you plan for the final project be on. We strongly recommend MySQL + Python. Meanwhile, you are allowed to use any programming language or relational database that you prefer, as long as you create an application for a relational database, meet all requirements, and you do all programming and query writing yourselves.

2. (25 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.

3. (25 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. (10 marks) List of **functional dependencies** that are applicable to the table (including the ones involving the primary key). For each functional dependency, clearly describe its meaning.

5. (10 marks) The **normalized schema** which is in 3NF or BCNF. Provide 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.

要自己在phpmyadmin里create table，然后进行一些query操作(应该average之类的就好了)

Part 2: Completed Project (Code and 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.

Marking Scheme (In 100-point Scale)

There are two components to be handed in, namely the project final report and the source code. The requirements for both components are introduced below. More information on bonus points from optional GUI implementation can be found at the end of this instruction.

Project Report

The final report should be one PDF file describing the information below.

1. (10 marks) A brief statement on the aim and background of the project.

2. (50 marks) **Query Demonstration:** Show the function (what the query does for the user within your application), the SQL code, and a pictorial UI (if there is one) of at least five of the following queries (10 marks each):

- Join query
- Division query
- Aggregation query (functions such as min, max, average or count)
- Aggregation with group-by (aggregated value for each group)
- Delete operation with Cascade
- Update operation

3. (10 marks) A conclusion that briefly describe the following:

- Roughly how much time did you spend on each part of the project?
- What aspects do you like the most and the least in this project?
- What helped you learn the best in the project?
- What suggestions do you have for a beginner in database design?

Code

The project's code base and related scripts are submitted through Blackboard as one Zip file. Please ensure:

1. (5 marks) The file contains all the code used in the application.

2. (5 marks) The file contains scripts that could be used to create all tables and data in the database.

3. (20 marks) All the queries included in the Query Demonstration section can be executed and produce the same results as shown in the report (4 marks each).

(Optional) Bonus Points – GUI and Usability Features

So far, if all the requirements are met, one should be able to obtain full (100) marks. However, if you would like to go deeper with database applications, and want to secure some bonus points for Part 2 of your project, you can go ahead with the optional GUI development. The maximum bonus for this part is 10 points to be added to your Part 2's final scores. Please note the final scores are strictly capped to 100. For example, if you scored 89/100 for your report+code and 10 in the bonus section, your final score for Part 2 will be 99. If you scored 95/100 for your report+code and 10 in the bonus section, your final score for Part 2 will be 100. The marks will be evaluated based on:

1. (5 marks) Use of a Responsive Web Design for your front-end implementation. For marking, please include GUI screenshots in the Query Demonstration of your report and upload your code.

2. (5 marks) Use of SQL assertions and/or triggers functionalities with GUI in your project. For marking, please add corresponding content in Query Demonstration of your report following the required format.