CS 417 Semester Project

Tom Kelliher

1 General Description and Goal

The goal of the semester project is to implement an application of a database system. This includes:

1. Finding an application for which a database system would be required.
2. Modeling the domain of the application, and defining the application functionalities.
3. Designing and implementing the schema.
4. Populating the database.
5. Building the application.

Projects will be done singly or with a partner. Students are encouraged to pick an application domain that excites them, provided it has a nontrivial database component.

2 What Should be in the Project?

Every project should illustrate the following features:

1. Conceptual design of the domain.
2. Schema design for the application.
3. Sophisticated querying.
4. Database updates.
5. Use of multiple views.

A web front end to the application, using Flask, is expected.

3 Guidelines for the Semester Project

Here are some guidelines that should help you make project decisions. Of the following, the first three guidelines are the most important.

1. Effective use: Try to make effective use of as much of the course material as possible. As we progress through this course, try to incorporate what we learn into your projects. For example, use joins, aggregates, indexes, triggers, etc.

2. Completeness: The project must be complete and stable enough for a good demonstration at the end of the course. This requirement is very important. If you are unable to give a reasonable demonstration, your project grade will suffer greatly.

3. Database size: All project databases must be of nontrivial size. You may interpret nontrivial based on your application.

4. Practicality: The application you build should be of practical use to a sizable community (at least in the foreseeable future).

5. Innovation: The more innovative ideas you include in your project, the more credit you will receive.

6. Miscellany: In addition to the above, your project grade will take into account factors such as teamwork, overall effort, timeliness, answers to questions about the project, justification of design decisions, and intermediate and final project reports.

3.1 Groups Decided

One member from each group must send me an email with:

1. The names of the members of the group.

2. The group name. (The name must adhere to the traditional Linux username conventions, consisting of one to eight alphanumeric characters with the first character being non-numeric.)

Document deliverables should be submitted as MS Word documents. Image deliverables, if they can't be inserted into a document, the preferred way of submitting them, should be submitted as PNGs or JPGs.

Be sure to consider the following in selecting your partners:

1. Project interests.

2. Working style.

3. Goals.

4. Target grade.

5. Availability outside of class.

6. Strengths and weaknesses.

Do not be shy about interviewing potential partners — you are forming a short-term business venture.

3.2 Proposals

Students should submit via Canvas a proposal containing:

1. Project description: 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 is the role of each project member in the project?

4. Schedule: what are the landmarks in your work.

5. Other, more specific comments if appropriate. Include all these items in a single Word document.

The proposals are meant to get you to start thinking about the project, get into groups and create a plan. All group proposals must be unique.

3.3 E/R Design and Relational Schema

Each group must submit:

1. E/R diagrams. Use `dia` with the `ER` sheet or some other software tool to produce a professional-looking E/R diagram.

2. Relational schemas. Again, use `dia` or another software tool. If using `dia`, use the `Database` sheet.

3. A written description detailing exactly what data their database will contain. This description should be fairly thorough.

The E/R diagram, relational schema, and written description should all be included in a single Word document.

3.4 Formal Specification

Each group must submit as a single Word document:

1. Begin by including your Proposal and E/R Design and Relational Schema deliverables, incorporating any changes you've made on your own or as a result of assessment feedback.

2. A description of the data you will have in the final application.

   This means revisiting your design and figuring out for good what tables you need to create, taking into account the principles of good design and your desired functionality.

   In addition to the description, write the SQL DDL that you will be use to create your tables. Include the DDL as an appendix in your document.

3. A description of the data constraints that exist and how you are modeling them.

   Describe the constraints, and show them in the DDL for your tables.

4. Describe the functionality the final application will have and describe the user interface. Include typical use cases for all classes of users (unprivileged user, administrative users, etc.).

5. An updated division of labor.

3.5 Sample End-to-End Application

Each group must submit:

1. A couple of screen shots of their application running a simple query: getting the data from the database, and putting it into a simple version of the final application of the project.

   These screen shots and explanations should be included in a single Word document.

2. The URL of your project's GitHub repository, as a separate submission in Canvas.

3.6 Completed Project

Each group must have a demo of their project. By the project completion deadline, each group must submit by the deadline (**no** working after the deadline):

1. A short description of the finished project and a description of how your final project differed from your formal specifications, if at all.

   These should be submitted as a single Word document.

2. The URL of your project's GitHub repository, as a separate submission in Canvas.