CIS560 Class Project – Database Application

General project description and goal

The goal of the class project is to build a database-driven web information system. This includes:

1. Finding an application for which database systems would be required
2. Modeling the domain of the application, and defining the application functionalities
3. Designing an E/R diagram and implementing the corresponding schema in MySql
4. Populating the database and exploring other aspects such as key and referential integrity constraints, and triggers.
5. Writing interactive queries and modifications on the database.
6. Writing the code needed to embed the database system in the application
7. Developing an interface for your database application.

You are free to propose your own project topics (some examples of possible topics are provided at the end of this document). However, we encourage you to pick an application that you will enjoy working with, as you'll be stuck with it for the whole semester! It's going to be especially nice if you pick a useful/realistic application where you can populate your database using real, as opposed to fabricated, data.

Minimum project requirements

Structural requirements: your database should be substantial, but not enormous. The minimal structural requirements that the database model should meet are listed below:

- The database should include **5 distinct, significant entity sets** (i.e., classes of objects), with different kinds of data (strings, integers, etc.).
    - Note that this is not the number of tables, but precisely the number of different entities/objects that you will model - each entity in a model will be converted into one or more tables in the corresponding database design.
    - Also note that a superclass entity along with all of its subclass entities count as a single entity toward the five-entity requirement.
- It should also include at least **4 relationships**
    - at least **one** of these relationships must be a legitimate **many-one** relationship.
    - at least **one** of these relationships must be a legitimate **many-many** relationship.

Functionality - your system's functionality should include:

- Insert, delete, update records into the database
- Search database and list records
- Key and referential integrity constraints
- Interface for accessing your application

Amount of data:

- Your database should contain a reasonable amount of data, so that interesting user scenarios can be shown in your final demo. In general, you should have at least 12 rows per table, with additional rows as needed to make your database a reasonable prototype. It is anticipated that at least one of your tables will grow substantially larger than this to support the set of queries and reports in a meaningful way. An example of how to "fabricate" realistic data will be provided in a class assignment, later in the semester.

While the minimum requirements for class projects should be met, we encourage students to consider incorporating more advanced features into their projects. Such features will also look good on your resumes and will definitely pay off when you'll be looking for jobs.

Project milestones

There will be several project milestones to help you make and assess progress. The milestones will correspond roughly to the 6 project goals listed above. Each milestone will be marked by a short report, which should state your project accomplishments at that time, and unexpected obstacles you may have encountered, if any. In some cases, we may have to revise the project plans accordingly.

1. **Project proposal** - The goal of the initial project proposal is to get you think of an interesting database application and come up with a plan for the project.

2. **Database design** – Once you decide on an application, you will next focus on the database design (this includes developing an E/R diagram, producing a set of relations, determining functional dependencies and ensuring each relation is in Boyce-Codd Normal Form with respect to these dependencies).

3. **Database implementation**: In this step, you will implement your database in MySql. This includes creating the tables in your database schema and populating these tables with 5-10 records of realistic data. You should also discuss how you will gather/generate additional data for your project. Additionally, you will need to support all necessary constraints and triggers to ensure referential integrity within your database.

4. **Queries and reports**: You will write at least 10 queries (seven questions and three reports) that are "interesting," substantially-different, and structurally-different, i.e., operations that require some of the more complex SQL constructs such as subqueries, aggregates, set operators, etc. (The difference between questions and reports is shown with some examples in the Project Ideas section below.)
5. **Code development and interface**: at this step, you will report on your progress in writing the code needed to embed the database system in the application, and/or progress on developing a web based interface for your database application.
6. **Final project submission, presentation and demo**.

Note: More details about the specific requirements of each milestone will be provided when you'll be asked to submit your report on that particular milestone.

Project ideas

Note: these are not specific proposals, only topic areas. You need to refine the topic and decide on the exact functionalities.

Library domain:
You have a nice collection of items that you would like to lend. You need to keep track of who is borrowing what, or what copies of what item you have in what condition. You would also like to be able to organize your collection in a variety of ways. You can support various questions such as who is borrowing item X, or which items that have property Y are available. A typical report could list what each person has borrowed, in alphabetical order by person, with the items borrowed by each person in alphabetical order by name.

Retail-business related:
Here, you could model a clothing store, a food store, etc. You would probably want to keep track of inventory, employee hours, or customer special orders. You might want to be able to answer questions such as how many items of type X do we currently have in stock. An example of a typical report would be a listing of the current inventory, organized by category, in reverse order of quantity of each item.

Service-based-business-related
Some examples in this category include catering business, a doctor's or vet's office, a repair business. You'd want to keep track of appointments or bookings, patient's or customer's history, etc. Your system can support questions such as what are the bookings/appointments for today or for some particular date? As a report, you might want to list, for each customer, what repairs have been done (or what appointments he/she has had), in chronological order.

Lesson/Course-related

In this domain, you can model the activities of an organization, or of a small group of individuals (or a business) offering courses (in some sport, for example, or dog training classes, or skiing lessons, etc.) You want to keep track of which lessons/courses still have openings, or what the prerequisites are for a particular course. As an example question, a user might ask: are there still any openings in the coming semester's Tuesday-Thursday section/offering of course X? A useful report would provide a list the courses in alphabetical order, listing the current enrollees in each course in alphabetical order by last name.

Sports or tournament domain:
In this domain, you can model a sports league (e.g., baseball, football, soccer) or a sports club, especially one that organizes and/or participates in tournaments, loans equipment, schedules games/matches, etc. You want to keep track of players, various player or team statistics, which team is playing where and when, who is supposed to officiate matches, etc. A sample question in this domain could be: for what team does a player X play or what were the scores of all the matches that X played in? A typical report might ask for the top players with respect to a particular statistic, for each team, or for a schedule, in chronological order, showing who is playing who, where, and when.

Movies playing in theaters:
In this domain you would be modeling entities movies, their actors, directors, genres, playing times, reviews. There exist several sources on the WWW from which you can get data to populate such a database. You can support various queries such as finding specific playing times, finding movies directed by a given director. You can also support updates to the reviews section of the database (e.g., viewers giving their own opinions). Another functionality could be to provide personal profiles of people (i.e., the movies they like) and recommendations of movies based on profiles of viewers with similar tastes.

Online bookstore:
In this domain you would be modeling entities such as books, their authors, topics (which may be a complex hierarchy). You may also model various attributes of the authors, the institutions they belong to, etc. You will support a buy/sell service of new books, used books, books used in specific university courses. A personal profile, similar to the one for movies is also a possibility.

Apartment renting:
This domain would require modeling apartments and their properties, areas of town and their various properties (e.g., bus lines, crime rate distance from various landmarks). There are many apartment data (landlord websites, postings) that can be used to populate your database. You would provide an interface for offering apartments for rent, finding apartments. You could search

apartments based on location, price, number of bedrooms, and sort them based on prices, posting date, distance. Visualize a list of apartments on a map, as an more advanced feature.

Database of web sites:
In this domain, your basic entities would be web sites. You can model their various properties: topic, url, organization to which they belong, other sites they point to, reviews of the site, quality, cost of accessing, etc. Users can add new sites that they found to the database, add opinions about existing sites, or search for sites that may be of interest for them. See Yahoo.com for a partial inspiration.

Other domains: social networks, bookmarks, blogs, images, music, sports, etc.

For more ideas, take a look at any Web shopping site. They all have a similar theme: products, customers, orders, shopping baskets, etc., and typically make for an interesting and appropriately sized application.

If you're still having trouble thinking of an application or if you are not sure your selected topic makes a good project, please feel free to consult with the course instructor. Coming up with a good design in the first place will pay off greatly as the project advances.

Project Tip

Start early – Remember that there is a lot going on at the end of the semester. Poor time management is not an excuse for a low quality project.

- “Unexpected issues”
- Data collection can take some time
- Schema may change as you dig more into the project
- Web space, database access will take some time to get familiar with, if you are not already.

Acknowledgments: The project has been adapted from similar projects required for courses on database systems offered by Arthur Keller at UCSC, Alon Halevy at University of Washington, Kevin Chang at UIUC and Thomas C. Bressoud at Denison University.

Grading: We will evaluate your work based on conceptual cleanness, how well you adhere to good practice of E/R design (as discussed in class), how well you translate your E/R diagram into relational schema, and how well you present both the E/R diagram and relational schema.