MS218 – Database Technologies

Course Project

Important Dates

Project proposal (Milestone 0): Monday, October 4
Milestone 1: Monday, October 18
Milestone 2: Monday, November 1
Milestone 3: Monday, November 29

Submission and Grading

There will be **three milestones**; see **Important Dates** above for due dates. You will find the details of what to submit for each checkpoint later on.

Each project will be graded on a scale of 0-100 points. A breakdown is as follows: 30 points for submitting the required work at three checkpoints; 10 points for Milestone 0.

Teamwork

The project should be completed in **3-person teams**.

All members in a team will receive identical grades for the project. *You are required to report each team member's effort and progress in the milestone and final reports.* If there is any problem working with your team members that you cannot resolve by yourself, bring it to the instructor's attention as soon as possible. Last-minute complaints of the form "my partner did nothing" will not be entertained.

"Standard" Course Project

The "standard" project is to design a database-driven web application. GUI of web would be only on pages. Specifically, you will need to complete the following tasks over the course of this semester. Note that different members of a team can work on some of these tasks concurrently.

1. Pick your favorite data management application. It should be relatively substantial, but not too enormous. Several project ideas are described at the end of this document, but you are encouraged to come up with your own. When picking an application, keep the following questions in mind:
    1. How do you plan to acquire the data to populate your database? Use of real datasets is highly recommended. You may use program-generated "fake" datasets if real ones are too difficult to obtain.
    2. How are you going to use the data? What kind of queries do you want to ask? How is the data updated? Your application should support both queries and updates.
2. Design the database schema. Start with an E/R diagram and convert it to a relational schema. Identify any constraints that hold in your application domain, and code them as database constraints. If you plan to work with real datasets, it is important to go over some samples of real data to validate your design (in fact, you should start Task 7 below as early as possible, in parallel to Tasks 3-6). Do not forget to apply database design theory and check for redundancies.
3. Create a sample database using a small dataset. You may generate this small dataset by hand. You will find this sample database very useful in testing, because large datasets make debugging difficult. It is a good idea to write some scripts to create/load/destroy the sample database automatically; they will save you lots of typing when debugging.

4. Design a web-based user interface for your application. Think about how a typical user would use your site. Do not spend too much time on refining the look of your interface; you just need to understand the basic "flow" in order to figure out what database operations are needed in each step of the user interaction.
5. Write SQL queries that will supply dynamic contents for the web pages you designed for Task 4. Also write SQL code that modifies the database on behalf of the user. You may hard-code the query and update parameters. Test these SQL statements in the sample database.
6. Acquire the large "production" dataset, either by downloading it from a real data source or by generating it using a program. Make sure the dataset fits your schema. For real datasets, you might need to write programs/scripts to transform them into a form that is appropriate for loading into a database. For program-generated datasets, make sure they contain enough interesting "links" across rows of different tables, or else all your join queries may return empty results.
7. Test the SQL statements you developed for Task 5 in the large database. Do you run into any performance problems? Try creating some additional indexes to improve performance.

Standard Project Ideas

Below is a list of possible project ideas for which high-quality datasets exist. Of course, you are welcome to come up with your own.

Entertainment, sports, or financial websites

Examples include those that allow visitors to explore information about movies, music, sports, games, stocks, etc. There are already many commercial offerings for such purposes. While there is less room for innovation, there are plenty of examples of what a good website would look like, as well as high-quality, well-formatted datasets. For example, *IMDB* makes their movie database available (http://www.imdb.com/interfaces); historical stock quote can be downloaded and scraped from many sites such as Yahoo! and Google Finance. This project is well-suited for those who just want to learn how to build database-backed websites as beginners. You can always spice things up by adding features that you wish those websites had (e.g., different ways for summarizing, exploring, and visualizing the data).

Websites providing access to datasets of public interest

If you are interested in doing some good to society while learning databases, this project is for you. There are many interesting datasets "available" to the public, but better ways for accessing and analyzing them are still sorely needed. Here are some examples:

- Data.gov (http://www.data.gov/) has a huge compilation of data sets produced by the US government.
- The Supreme Court Database (http://scdb.wustl.edu/data.php) tracks all cases decided by the US Supreme Court.
- US government spending data (https://www.usaspending.gov/DownloadCenter/Pages/default.aspx) has information about government contracts and awards.
- Federal Election Commission (https://www.fec.gov/data/) has campaign finance data to download as well as nice interfaces for exploring the data.
- GovTrack.us (http://www.govtrack.us/developers) tracks all bills through the Congress and all votes casted by its members. The Washington Post has a nice (albeit outdated) website (http://projects.washingtonpost.com/congress/113/) for exploring this type of data (in predefined ways), but you can be creative with additional and/or more flexible exploration and analysis options. Vote Smart (https://votesmart.org/) has a wealth of additional, useful information on votes, such as issue tags, synopses and highlights.
- Each state legislature maintains its own voting records. For example, you can find North Carolina's here: http://www.ncleg.net/Legislation/voteHistory/voteHistory.html. Some states provide records in already structured formats, but for others, you may need to scrape their websites.

- The Washington Post maintains a list of datasets (http://www.washingtonpost.com/wp-srv/metro/data/datapost.html) that have been used to generate investigative news pieces. Most of these datasets hide behind some interface and may need to be scraped. Use this list for examples of what datasets are "interesting" and how to present data to the public effectively.
- Stanford Journalism Program maintains a list of curated transportation-related datasets (http://www.datadrivenstanford.org/).
- National Institute for Computer-Assisted Reporting maintains a list of datasets of public interest (http://www.ire.org/nicar/database-library/). Use this list for examples of what datasets are "interesting"—they are generally not available to the public, but there may be alternative ways to obtain them.
- Google Fusion Table (http://www.google.com/fusiontables/) hosts quite a number of datasets of public interest. It is a good place to find datasets or data sources to work on, and you can consider using it as a method of hosting your data for public access.

  Your task would be to take one of such datasets, design a good relational schema, clean up/restructure the data, and build a website for the public to explore the dataset. If you are interested in this line of projects, discuss your plan with the instructor. Some of the datasets pose significant challenges in cleansing, analysis, and visualization.

"Open" Course Project

The open option is a chance for you to build something that you really want, provided it is related to data management. You need to write a detailed project proposal, and the course staff will work with you to ensure that your project meets the minimum requirements of depth and scope. You are encouraged to build novel systems and tackle challenging problems. Your "risk factor" will be considered in grading. Because of limited time, it is important to stay focused and ensure that certain pieces of your project are completely done; it is difficult to judge a project if nothing works.

A project proposal containing:

- A description of the problem you wish to solve or the application you wish to develop, and, more specifically, what you plan to demonstrate at the end of this project.
- How it is important, interesting, and/or useful.
- Initial thoughts on how to approach the problem or build the application, including the preliminary system architecture.
- Survey of previous and/or related work and systems, including discussions of how they relate to your problem as well as their limitations and/or flaws.