ITWS-4960/6960 — Database Systems
Graduate Student Project

Summary

For this project, you will find two publicly available datasets that share a common attribute (e.g., Zipcode), create a normalized schema describing the structure of the data, and produce an application that can populate your schema with the data, and run queries on the data, producing useful output. Students taking the course at the graduate level will have to use a non-relational database system for a portion of their schema (and the related data), requiring their application to interface with both relational and non-relational components.

Objective

There are several objectives for this assignment.

- Gain an awareness of the scope of datasets publicly available for research purposes
- Demonstrate an ability to understand the structure of a dataset, as well as an ability to apply that understanding to create an effective schema
- Apply concepts learned during class to query the data, and extend those concepts to create an application allowing users to do the same

Description

There are a number of different sources of publicly available data. Both the State of New York and the Federal Government provide hundreds of datasets. There are numerous other sources of open data as well, but those two will get you started. Please pay attention to licenses for any datasets you use. Data itself is generally not copyrightable, but schemas are, and there may be terms of service for accessing the data itself.

Select two datasets that are robust enough to be interesting (a dataset with only four columns and a few thousand rows probably doesn't qualify). They should share a common attribute (or set of attributes). Create a SQL schema for your data, making sure that it's appropriately normalized. Graduate students will also need to separate part of that schema into a separate (non-relational) database system (e.g., MongoDB or even a collection of `.yaml` files). The type of system you choose should be appropriate for the part of the data you choose to store there.

Create an application in a modern programming language of your choice that will load the dataset into a Postgres database defined by your schema (and non-relational database). Take some time to explore the data by running some SQL queries. Once you have an idea of some of the more interesting aspects of the data, create an interface for your application that will allow the user to explore the data as well.

Your application shouldn't reimplement the wheel. You don't need to provide the user with a way to do whatever they want. It should provide more of a self-guided tour, rather than a detailed map. It should provide interactivity beyond simply allowing the user to run one of five or six static queries, but it doesn't have to allow them to write their own queries.

For example, there might be a dataset giving the results of health inspections of restaurants in New York. Your application might allow the user to see which restaurants in their area had violations, or how often a given restaurant received a violation, or whether restaurants in a certain area get more violations than other areas.

The interface can be text-based. If you want to go further and provide visualizations, that's fantastic, but it isn't within the scope of the project. Your application should be able to be built easily, the data loaded easily, and used easily.

You will demonstrate your application for the class in a five to ten minute presentation, in which you will discuss your choice of datasets, outline the design of your schema, and demonstrate the types of queries your application can perform.

You may work either individually or in teams of two.

Deliverables

There are two main deliverables.

A memo providing the following information:

- Your name (or both names, if working as a team)
- The datasets you plan on using
  - The location of the data
  - Any relevant license information
  - How you plan to join the two datasets
- What programming language you plan on using for the project

The memo will be due before the rest of the project and will serve as a way to make sure the project scope is appropriate. It will also allow determination of the order of the final presentations.

A zip file containing:

- the source code for your application
- a readme file explaining
  - What data you used and where you got it
  - How to build your application
  - How to load the data into the application
  - How to run and use the application to explore the data
- The datafiles

If you want, everything but the memo, datafiles, and readme can be delivered via a github or gitlab repository (this is preferable, but not required).

Grading

This project will count as fifteen percent (30%) of your total grade.

Points will awarded for the following:

- **Schema design and definition.** Does your schema accurately and effectively store the data, is it appropriately normalized, did you choose appropriate datatypes? (25pts)
- **Application correctly loads the data.** (10pts)
- **Application facilitates exploration of the data.** A user should be able to use your application to explore your chosen datasets. (35pts)
- **Application conforms to best-practices.** Your code should be clear and the components of your application well-organized. It shouldn't contain any SQL-injection vulnerabilities, and the database should be reasonably configured. (15pts)
- **In-class presentation** (15pts)

Note that if your application doesn't correctly load the data, exploration of the data will likely be impossible, so while loading the data is only worth ten points, if your application doesn't load the data, it's unlikely you'll earn many of the points for facilitating exploration of the data.

Due Dates

The memo is due via LMS by 11:59pm on November 3.

You should be prepared to present your project to the class during the lecture by 6:00pm on December 13. Your deliverables must be submitted via LMS by 11:59pm on December 8.