ITWS - 4250/6250 — Database Applications and Systems

Final Project

Summary

For this project, you will find multiple publicly available datasets that share common attributes (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—including the ability to refresh the data—and run queries on the data, producing useful output.

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, using concepts learned during class, to create an effective database 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 eligible for copyright protection (at least in the United States), 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). *Students taking the course for graduate credit need to select two additional datasets, for a total of four.* They should share a common attribute (or set of attributes). Create a SQL schema for your data, making sure that it's appropriately normalized. *Students taking the course for graduate credit need to use a non-relational database and schema for some portion of their data.*

Create an application in Python 3 that will load the dataset into a Postgres database defined by your schema. The loading process should be able to be re-run with updated datasets to refresh the data in the 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 re-implement 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 (you will not be graded on the appearance of your interface). 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 short 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.

All work will be done either individually or in teams of two.

Deliverables

There are four main deliverables:

- Project Memo
- Database Schema
- Project Code
- Presentation

Project Memo

The memo should provide the following information:

- The names of the members of your team
- The datasets you plan on using
  - The location of the data
  - Any relevant license information
  - How you plan to join the datasets

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, as well as making sure you all have teams early enough to complete the project.

Database Schema

You will submit a single **SQL file** that can be run to create the schema for your database. The SQL file will also be due before the rest of the project and will be graded at that time so that feedback can be incorporated into the final product. *Students taking the course for graduate credit will need to bear in mind that some portion of their data will be stored in a non-relational database. That aspect of the project is not due with this deliverable.*

Project Code

The `final-project` directory in the class git repo will eventually contain several files, along with an `instructions.md` file, which will provide more specific directions for how your code should be organized, and how it will be run during grading. Your project code submission should follow those instructions.

Presentation

While technically nothing needs to be submitted for the presentation, it is a separate part of the project with its own deadline. The presentation should last around five minutes. It should briefly discuss your chosen datasets and give an overview of your schema. It should then demonstrate the main parts of your application. It should also discuss any major challenges you faced. You probably won't have time to exhaustively demonstrate every feature, so plan accordingly. You should also be prepared to answer questions.

Grading

This project will count as thirty 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.** *Note that this includes the code required by students taking the course for graduate credit for loading the data into the non-relational database of their choice.* (20pts)
- **Application facilitates exploration of the data.** A user should be able to use your application to explore your chosen datasets. (25pts)
- **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 (or their non-relational equivalents). (20pts)
- **In-class presentation** (10pts)

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 twenty 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 on Submitty by 11:59pm on Friday October 9
- The schema is due on Submitty by 11:59pm on Monday October 26.
- The data-loading code is not formally due, but students should aim to have it completed by Wednesday November 25.
- The completed application is due on Submitty by 11:59 on Sunday December 6.
- You should be prepared to present your project to the class during the lecture period on Wednesday December 9.

Late Days may not be used for project deliverables.