CSCI - 4380 — Database Systems
Final 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, 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 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.

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're not being 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 five 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 in teams of four. You may select your own group or ask to be assigned to one. Group assignments will be posted in a Google Docs Sheet, the link to which will be posted on Piazza. Any students who haven't selected a group, and indicated their preference to the professor or TAs on Piazza, by the end of the day February 1 will be assigned a group.

Deliverables

There are four main deliverables.

A memo providing 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 two 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. It will also allow determination of the order of the final presentations.

A 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.

A zip file containing:

- your up-to-date `.SQL` file that creates your schema
- a python file `load_data.py` that will load the data into the schema
- a `readme.md` file that has instructions for any necessary database setup or configuration needed to run your python file
- any other supporting files needed to run your program

Your python file should load the data from your datasets into the database schema defined in your SQL file. It should do so in an *idempotent* manner. In other words, once the database is set up, we should be able to run your python script an arbitrary number of times without corrupting the data.

You should also be able to run the script again with newer, up-to-date versions of the datasets, and have the updates properly reflected in the database.

A zip file containing:

- your up-to-date `.SQL` file that creates your schema
- your up-to-date `load_data.py` that will populate the schema
- an `application.py` file that will serve as the entry point into your application
- a `database.py` file that contains the database-related code for your application
- a readme file with any necessary instructions for building and running your application
- any other files or source code necessary

Grading

This project will count as twenty percent (20%) 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.** (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. (10pts)
- **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 February 8.
The schema is due on Submitty by 3:59pm on Monday February 25.
The data-loading code is due on Submitty by 3:59pm on Monday March 25
The completed application is due on Submitty by 3:59 on Monday April 22

You should be prepared to present your project to the class during the lecture periods of Monday April 22 and Thursday April 25.

Late Days may not be used for project deliverables.