Final Project

Intro to Databases (CS195A)

This project may be completed individually or in a team of 2.

Objective: We have come all the way from drawing ER diagrams to making our own little Social Media app. As new moguls of the tech industry, we will end the semester with one final application that combines all the things we have learned so far and hopefully more. Application of the final project will be centered around a SQLite relational database, that pulls and pushes data with an interface. The interface of the application can be either web-based (Flask) or command-line. In addition to setting up the data, the app should also integrate data visualization and summary statistics in some way.

Procedure: Start by first identifying the scope of the project. What is it that you are trying to build? What problem do you wish to solve? A list of possible examples has been outlined in a previous documentation; use the list as a guide to identify your goals for the project. Once you know what you wish to build, your procedure would look something like this-

i. **Data Collection:** Source the data that you wish to use. It can be fabricated too if that is what is needed (for example, for a point-of-sale app, you can fabricate sales and inventory data). On the other hand, you can also have a dynamic data pipeline**.** Many websites allow users to continuously pull data from them (for example, up to date price of a given stock). If this is something you would like to do and are not well versed in API system for different websites, please come to office hours and we will be happy to help you out.
Key Constraints and Deliverables: a) Your database should consist of **at least 3 relational tables** with 20 rows each. b) You are not allowed to use any of the datasets we have covered in this class.

ii. **ER Diagram and Relational Model:** The next step is to create an ER diagram and relational model of your dataset. How will your data be structured? Include an E-R diagram, as well as a list of relations. The recommended textbook (O'Reilly media), provides great examples to assist if you run into issues. *Key Constraints and Deliverables*: An E-R diagram with entities, relationships (cardinality labeled) and attributes. A list of relations, including identification of primary keys and foreign keys.

iii. **Data Cleaning and Loading:** The next step is to take your dataset and load it as SQLite database. You can do that by creating a table and inserting the values, or just loading it from the CSV format using .import function. After you have loaded the database, make sure to clean your tables. This includes- fixing all the broken entries, dropping useless columns, dropping empty rows, assigning correct data types, and more. For example, if you have a table that lists the weight of each hardware tool, you want to make sure that that column is of the type INTEGER, if the value is whole numbers, or REAL. Or if your values are TEXT as "n/a" or "no value", then it really should be assigned as NULL. If your data is dynamically sourced, your script should include the ability for cleaning and loading on the fly.
Key Constraints and Deliverables: a) Avoid inserting each row one by one. b) Your final database **should be in a .sqlite or a .db format**, and it should contain all the tables you will be using for the rest of the project.

iv. **Application Functions:** Before we design the interface, we will layout the functionality of the application. In broadest terms, the functionality can be anything that you wish to do with your dataset. This includes, but is not limited to, additions, deletions, modifications, and queries. It is

very similar to what we have already done with the social media app. However, there are some key constraints that are listed below.
Key Constraints and Deliverables: a) App should **allow user to add, remove, and modify records** from all tables. b) Allow user to **make basic statistical queries** such as- Mean, min, max, median, standard deviation mean, on numerical columns. c) Allow user to **query against any column using WHERE** and list sample data. d) Allow for **2 queries that require a JOIN.** e) Allow **visualizations for key metrics** of your choice with at least two different types of plots.

v. **Application Interface:** Once you have implemented your features, the user should have the ability to use those features. Implement an interface of your choice (command-line or web-based) that allows the user to use all the functionalities listed above.
Key Constraints and Deliverables: a) Submit all files required to run your application.

vi. **Final Presentation:** During the final week of classes, you will have a 5-minute block to present your project. *Key Constraints and Deliverables*: A walk through of the database design (E-R diagram, relations) and a complete demonstration of the functionality of your project. All team members must participate.

Conclusion: The goal of this project is to get you comfortable with end-to-end implementation of databases and their use. This project is a large undertaking, so please start early. The final submission of your project should include the following:

A) Original Dataset (as CSV, or JSON, or XLS, whichever way you sourced it)
B) Clean SQLite Database
C) Python Files(s)
D) Write-up that includes the following:
    a. Description of application
    b. Full name of all members of the team
    c. E-R diagram
    d. Relational Model

Note: **No data file should have more than 500 records.** You may demo a larger data set when presenting in class, but please submit only a subset of 500 or less.

We understand the project has a lot of specific requirements, the goal is to make an app that neatly combines all these requirements instead of a disjointed app that simply satisfies these requirements. Take this as an opportunity to make something that you can present in future job applications and your GitHub account.

And don't forget to seek help where necessary.