DSCI 551 – Fall 2023

Project Guideline

In this project, you are asked to design and implement a sample database system. Here are general requirements.

- The system should support a data model, which can be relation (as in MySQL) or JSON (as in Firebase and MongoDB), or any other model of your choice. Users of the system will structure their data using the model provided by the system.
- The system should have its **own** query language which should be different from existing query languages, including SQL, and queries provided by Firebase and MongoDB. It is ok that the language is like natural language, e.g., "find employees who are at least 25 years old".
- The query language should support projection (selecting a subset of rows), filtering (selecting rows), join (e.g., combining multiple data tables), grouping, aggregation, and ordering.
- The system should also provide commands for inserting, deleting, and updating the data. These commands can be like that in existing database systems.
- You are free to decide how you store the data in a data model (e.g., you may store a table in a file), and how you implement the data modification commands above.
- Your system should not load the entire database into the memory and process queries and data modifications on the entire database. Instead, you should assume that the database may potentially handle a large amount of data that might not fit in the main memory.
- You should implement an interactive command line interface (similar to MySQL, MongoDB, sftp clients, etc.) for users to interact with the systems, issue commands, and get results. As an example, suppose your database is called MyDB, your interface may look like:
    - MyDB > create table person(a int);
    - MyDB > Table created.
    - MyDB > insert into …
    - MyDB > find employees who are at least 25 years old
    - MyDB > …
    - MyDB > exit
- You should show how to create a database using your system to store multiple real-world data sets, and how the queries and data modifications work on the data sets.
- Your dataset should be some existing real-world dataset available on the Web. For example, Kaggle, google, etc. are good places to find such datasets。
    - https://datasetsearch.research.google.com/
    - https://www.kaggle.com/
- You can form a project team of no more than 3 people. However, there are additional requirements if your project group has more than one person.
    - If your team has two people, you need to design two database systems, one for storing relational database, the other for NoSQL data. The data set for NoSQL should be different from the ones used for relational.

    - If your team has three people, you also need to design two database systems, one for relational, the other for NoSQL data; and use different datasets. In addition, you are expected to build a web application that demonstrates the functions of your database systems. Students in the past have used framework, like Flask, with success.
- There is no restriction on programming languages that can be used for the project.

The project will be done in phases.

- Proposal (due 9/22, 10 points): tell us what you plan to do (basic design of system, data model, query language, data modification, and ideas on how to implement them. Tell us your project group members, for each member, indicate the background (including undergraduate major, skills, etc.) and responsibilities for the project. Note choose your team members wisely, as your project is supposed to be a teamwork and all members should contribute to the project equally.
- Midterm progress report (due 10/13, 5 points): tell us your progress so far and the challenges you encountered.
- Demo (11/21, 5 points): Give live demo of your project. All project members should be present during the demo. Before demo, please prepare slides talking about your project design. The presentation will be about 5 minutes and the demo 10 minutes. Note we may also use part of 11/28 meeting for your project presentation. Note two hours of 11/28 class will be used for your comprehensive exam.
- Final report (due 12/8, 10 points): the final report should be comprehensive, details your design and implementation.
- Implementation (due 11/21, 70 points): note your project should be fully implemented before the demo. You should include in your final report a link to Google drive where you will upload your project codebase and documentations. Make sure you give access to your project folder.
- Note proposal, midterm report, and final report will need to be uploaded to course web site (Blackboard for afternoon section, D2L for morning sections including DEN). Submission entries will be announced before the deadline is approaching.