Fall 2023

Semester Project

University at Buffalo

CSE 4/560 Data Models and Query Language

Project Overview

Build up a database to demonstrate interesting searches. This project shouldbe a team effort, and each team should contain three or two people. The project is broken down into two phases. The project is expected to be motivated by issue(s) in an application domain of your interest, and addressing these issues using data gathered from the domain.

For each phase, only the team leader (Member 1) needs to submit your work.

1.Tasks:

1. **(Task1**) :Select one interesting use case domain, building your database using SQL. It should be relatively substantial, but not too enormous. Several project ideas are described at the end of this document. However, these ideas aimto support you start thinking, and you are encouraged to come up with your own choice of use case. Please keep the following points in mind: a) while real datasets are highly recommended, you may also use program-generated"fake" datasets if real ones are too difficult to obtain; b) How will you use the data? What kind of queries do you want to ask? Howis the data updated? Your application should support both queries and updates.
2. **(Task2)**Design the database schema. Start with an E/R diagram and convert itto a relational schema. Identify any constraints that may be applicable in your use case problem and implement them using database constraints. If you plan to work with real datasets, it is important to go over some real data samples to validate your design. Do not forget to apply database design theory and check for redundancies. Create a sample database using a small subset by hand to facilitate debugging and testing because large datasets make debugging difficult. It is good for different scripts to automatically create/load/alter/update/destroy thesample database.

3. (**Task 3**) Acquire the large ”production” dataset, either by downloading it from a real data source or generating it using a program. Make sure the dataset fits your schema. You might need to write programs/scripts to transform them into a suitable form for loading into a database for real datasets. For program-generated datasets, make sure they contain interesting enough ”links” across rows of different tables to show the results of different Advanced SQL queries learned in class.

4. (**Task 4**)You are required to make sure all of your relations are in Boyce-Codd Normal Form. Provide a list of dependencies for each relation. Decomposethem if the tables are not in BCNF. If you decide to keep it in 3NF instead of BCNF, justify the decision for a particular relation. Your report for this milestone should contain a separate section with the details ofthe transformation from the initial schema to the final schema where the relations are in BCNF.

Note: This is quite possible that your initial schema is already in BCNF, and in that case, you need to provide us the functional dependencies and convince us that the relations are already in BCNF.

5. **(Task 5)** Do you specifically run into any problems while handling the larger dataset? Did you try to adopt some indexing concepts to resolve this? Briefly describe the questions you faced and how you solved them.

6. (**Task 6**)Test your database with more than 10 SQL queries. You are supposed to design 1 or 2 queries for each inserting, deleting, and updating operation inyour dataset. And please write select queries no less than 4 queries. Your select queries should be in different types of statements, for example, you can use ”join”, ”order by”, ”group by”, subquery, etc. Get your execution results and take screenshots to show them.

7. (**Task 7**)Query execution analysis: identify three problematic queries (show their cost), where the performance can be improved. Provide a detailed execution plan (you may use EXPLAIN in PostgreSQL) on how you plan to improve these queries. You can find the EXPLAIN tool in Figure 1.

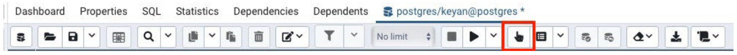

Figure 1: EXPLAIN tool

8. **Bonus Task [10]** Build a running website that links with your database to display/visualize query and query results. You can use any language.

2. General Project Requirements

1. **Work Environment:** The required language for the project is PostgresSQL
2. **Programming:** Prepare yourself to program by learning from the course textbooks and online resources.
3. **Academic Integrity: You will get an automatic F for the course if you violate the academic integrity policy. See the course syllabus for more detail.**
4. **Project Phases:**
   This project is divided into two phases.
5. **Teams:** For the duration of the project you may work in groups of two or three. Please make sure that you have registered your team via Piazza, which must be completed before asking for project guidance. Project discussion should only occur between you and your teammate, or you and course staff. Each team member must contribute to each part of the project. There will be **one submission per team.**

3. Submission Requirements

1. **Deadlines:** Your submission for **phase -1** is due by 11:59 PM, 10/20/2023. **Phase-2** submission due is 12/8/2023. For each day your submission is late, there will be a 25% penalty. You must submit **Phase 1** to begin work on **Phase 2**. Please start the project as soon as possible.

Submission: Project deliverables should be submitted via Brightspace. There should be one final submission per group. You can submit multiple times before the deadline but we will grade the final submission. For **Phase 1** report submission you are required to submit a pdf file named **member1_member2_member3_phase_1.pdf in** IEEE/ACM format.

For the **Phase 2** final submission, you are required to submit a **zip** file containing all the required deliverables. The zip file must be named: **member1_member2_member3_phase_2.zip**. It should contain a PDF for your project report named Phase2report.pdf in IEEE/ACM format, demo video, a database SQL dump, and all other files within the zip file

4. Deliverables [Total 100 points]

4.1 Phase-1: Due date: 10/20/2023 (**30 Points**)

You are required to hand in a report that contains the overview of your project proposal. The overview can change slightly as we go over the course, but the central theme should be intact. The proposal should consist of two or more pages describing the problem you plan to solve, outlining how you plan to solve it, and describing what you will ”deliver” for the final project. Your reportshould contain the following sections:

1. **Project details**: Name of your project, your team, and all team members,everyone’s UB id(not the UB number);
2. **Problem Statement [5 points]:** Form a title and problem statement that clearly state the problem and questions you are trying to answer. Why do you need a database instead of an Excel file? Additionally:
   a. Discuss the background of the problem leading to your objectives. Why is it a significant problem?
   b. Explain the potential of your project to contribute to your problem domain. Discuss why this contribution is crucial.
3. **Target user [5 points]** : Who will use your database? Who will administer thedatabase? You are encouraged to give a real-life scenario;
4. **E/R diagram [10 points]**: Draw an E/R diagram for your database and briefly describe the relationships between different tables. (Do not draw the figure by hand, you may use any tools to design or generate your E/R diagram
5. **Tasks** (3&4) should be completed **[10 Points]**

NOTE: Define a list of relations and their attributes.

1. Indicate the primary key and foreign keys (if any) for each relation. Justify your choice;
2. Write a detailed description of each attribute (for each table), its purpose, and datatype;
3. Indicate each attribute’s default value (if any) or if the attribute can beset to ’null’;
4. Explain the actions taken on any foreign key when the primary key (that the foreign key refer to) is deleted (e.g., no action, delete cascade, set null,set default).

4.2 Phase-2: Due date: 12/8/2023 (**70 Points**)

Start planning for tasks 5-7. The detailed description and demonstration of your work on each of these tasks should be presented in the Project description and demo presentation video.

1. Do you specifically run into any problems while handling the larger dataset? Did you try to adopt some indexing concepts to resolve this? Briefly describe the questions you faced and how you solved them. **[5 Points]**

2. Test your database with more than 10 SQL queries. You are supposed to design 1 or 2 queries for each inserting, deleting, and updating operation inyour dataset. And please write select queries no less than 4 queries. Your select queries should be in different types of statements, for example, you can use ”join”, ”order by”, ”group by”, subquery, etc. Get your execution results and take screenshots to show them**. [20 Points]**

3. Query execution analysis: identify three problematic queries (show their cost), where the performance can be improved. Provide a detailed execution plan (you may use EXPLAIN in PostgreSQL) on how you plan to improve these queries**. [10 points]**

4. **Presentation and Demo [15 marks] :** Record an 8-15-minute video about your project's demo to Brightspace. Besides, you need to prepare a live presentation as well. It can be similar to your recording but our TAs will attend your live demo and ask questions related to your project. We will fix each checkpoint's presentation dates after the corresponding deadline.

3. Project Report [20 marks]:

write the report and state clearly the contribution from each team member. It should be 5-8 pages in IEEE/ACM . This is your final submission of the project. The complete report should be there.
Your complete report should contain:

1. Details of all the tasks required to perform in this project. Please high- light any new assumptions, E/R diagram, and list of tables (if they have changed since Phase 1 that you have added/edited).

2. Create a file *create.sql* which will create all the tables in your database. Load these relations from data files (tab or comma-separated files). The tab or comma-separated files can be created by you (dummy values) or other sources. Create a *load.sql* file for bulk loading. Create a *readme.txt* file that states your data source. Put *create.sql*, *load.sql*, all the '.dat' files (or *.csv* files, or data files in any other format), and a *readme.txt* file into a sub-directory. If you generate and import your data through some scripts, you do not have to create *create.sql* or *load.sql*, but please also include a *readme.txt* file to describe how you built tables and imported data.

***Final Project Demo**. You will need to present your system's working demo at the end of the semester. Instructions on how to sign up for the demo will be given during the second to last week of the class. You are also encouraged to stay in touch with the TAs (we will assign a TA for each team) to discuss your project and get their feedback on how to improve.

Example Application Scenarios

Please follow the links for the details.

- IMDb makes their movie database available https://www.imdb.com/interfaces/.
- Data.gov http://www.data.gov/ has a huge compilation of data sets produced by the US government.
- The Supreme Court Database (http://scdb.wustl.edu/data.php tracks all cases decided by the US Supreme Court.
- US government spending data (https://files.usaspending.gov/database_ download/) has information about government contracts and awards.
- Federal Election Commission (http://www.fec.gov/disclosure.shtml) has campaign finance data to download; their “disclosure portal” (http: //www.fec.gov/pindex.shtml) also provides nice interfaces for exploring the data.
- historical stock quotes can be downloaded and scraped from many sites such as Yahoo! and Google Finance.
- You are allowed to use any open-source datasets.