Professor: Shamsad Parvin,PhD

Semester Project

Spring 2025

University at Buffalo

Computer Science and Engineering

CSE 4/560 Data Models and Query Language

Project Overview

Build a database to demonstrate interesting searches. This project should be a team effort, consisting of two or three people. The project is divided into two phases. It should be motivated by issues in an application domain of your interest, addressing these issues using data gathered from the domain.

Only the team leader (Member 1) needs to submit the team's work for each phase. For further directions, please check the end of the description.

Phase-1: Due 03/16/2025 **Phase-2: Due 04/27/2025**

1) General Project Requirements

1. **Work Environment:** The required language for the project is PostgreSQL
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. **References:** Include all the references that have been used to complete the project.
6. **Contribution:** We expect equal contribution to the assignment. Provide a contribution summary for each team member in the form of a table below. If the contribution is not equal, the person who contributed less will be penalized. This is to ensure fairness to other teammates.

2) Register your team (31st Jan)

You may work in a team of 2 or 3 members. To find teammates, you can also make a public post on Piazza. Please make sure that you have registered your team via Piazza before

asking for project guidance. Project discussions should only occur between you and your teammates, or between you and the course staff. Each team member must contribute to each part of the project. There will be one submission per team.

3) Tasks:

1) **Task 1:** Choose an interesting use case domain for building the database using SQL. The dataset should be substantial yet manageable. Real datasets are recommended, but program-generated "fake" datasets can be used if real ones are difficult to obtain.

   Consider the following:

   - How will the data be used?
   - What kind of queries will be asked?
   - How is the data updated?
   - Ensure the application supports both queries and updates.

2) **Task 2:** Design the database schema. Start with an E/R diagram and convert it to a relational schema. Identify any constraints that may be applicable to your use case problem and implement them using database constraints. If you plan to work with real datasets, it is important to review some real data samples to validate your

   design. Don't forget to apply database design theory and check for redundancies. Create a sample database using a small subset manually to facilitate debugging and testing, as debugging can be difficult with large datasets. It's beneficial to have different scripts to automatically create/load/alter/update/destroy the sample 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 ensure that all your relations are in Boyce-Codd Normal Form (BCNF). Provide a list of dependencies for each relation. Decompose them if the tables are not in BCNF. If you decide to keep a relation in 3NF instead of BCNF, justify the decision. Your report for this milestone should include a separate section detailing the transformation from the initial schema to the final schema where all 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:** Test your database with more than 10 SQL queries. You are required to implement the following:

   - Design 1 or 2 queries for inserting, deleting, and updating the dataset.
   - Write at least 4 select queries. Ensure that your select queries utilize different types of statements, such as "join", "order by", "group by", subquery, etc.
   - Create stored procedures for common operations such as inserting, updating, or deleting or selecting records.

   Execute these queries, capture the results, and take screenshots to demonstrate them.

6) **Task 6:** Implement a simple Transaction with Failure Handling using a Trigger. You need to create a transaction query and need to demonstrate its working. You will need to create a trigger that is executed whenever the transaction Fails. What happens when a transaction gets aborted due to a failure? Justify your reasoning.

7) **Task 7:** Identify three problematic queries (show their cost), where the performance can be improved. Provide a detailed execution plan (you may use EXPLAIN in PostgreSQL) and propose improvements using indexing strategies.

   Note: There may be no problematic queries if your dataset is sparse. In such a case, explain a common scenario where your queries can lead to poor performance when your data grows in future. Provide its execution plan and queries to create indexes to improve performance.

8) **Bonus Task (10) (Optional)** Students who are interested in building an end-to-end project or using the latest technologies in the data domain can attempt this bonus task. You can build a website to showcase the problem you solved or capture insights from the database using BI tools such as Power BI or Tableau.

   Note: Marks achieved from the bonus task will help cover lost marks in Phase 1 and Phase 2 only.

4) Submission Requirements

1) **Deadlines:** Your Phase 1 Submission is due by 11:59 PM on 03/16/2025. For each day your submission is late, there will be a 15% penalty. You must submit Phase 1 to begin work on Phase 2. Phase 2 Submission is due by 04/27/2025. Each day your Phase 2 submission is late, there will be a 15% penalty. The deadline is firmed and will not be extended.

2) **Submission details:** Project deliverables should be submitted via Brightspace. There should be one final submission per group by Team Leader. 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.

   https://www.ieee.org/conferences/publishing/templates.html

   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, https://www.ieee.org/conferences/publishing/templates.html

   A demo video, a database SQL dump, and all other files within the zip file.

5) Deliverables [Total 200 points + 10 point for Bonus Part]

5.1 Phase-1: Due date: 03/16/2025 (100 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 report should contain the following sections:

1) **Project details:** Name of your project, your team, and all team members, everyone's UB id (e.g. shamsadp)

2) **Problem Statement [20 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 [20 points]:** Who will use your database? Who will administer the database? You are encouraged to give a real-life scenario.

4) **E/R diagram [20 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 **[40 Points]**

Define a list of relations and their attributes.

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

Note : Minimum 8 relations for 460 (Underdard) students and 10 relations for 560 (Grad) Students. The number of records in each table should be a minimum of 3000.

5.2 Phase-2: Due date: 04/27/2025 (100 Points)

Start planning for tasks 5-8. 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) Test your database with more than 10 SQL queries. You are required to implement the following:
[25 Points]

- Design 1 or 2 queries for inserting, deleting, and updating the dataset.
- Write at least 4 select queries. Ensure that your select queries utilize different types of statements, such as "join", "order by", "group by", subquery, etc.
- Create procedures/functions for common operations such as inserting, updating, or deleting or selecting records.

Execute these queries, capture the results, and take screenshots to demonstrate them.

2) **Transaction & Triggers:** Implement a simple Transaction with Failure Handling using a Trigger. You need to create a transaction query and need to demonstrate its working. You will need to create a trigger that is executed whenever the transaction Fails. What happens when a transaction gets aborted due to a failure? Justify your reasoning.
[25 Points]

3) **Indexing & Query Execution Analysis:** Identify 3 problematic queries (show their costs), where the performance can be improved. Provide a detailed execution plan (you may use EXPLAIN in PostgreSQL) and propose improvements using indexing strategies. **[25 Points]**

Note: There may be no problematic queries if your dataset is sparse. In such a case, explain a common scenario where your queries can lead to poor performance when your data grows in future. Provide its execution plan and queries to create indexes to improve performance.

4) **Presentation and Demo [15 marks]:** Record an 8–15 minute video about your project's demo to Brightspace. Our TAs will contact you if they have questions related to your project. 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.

5) **Project Report [10 marks]:** Write the report and state clearly the contribution from each team member. 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 highlight 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 include a readme.txt file to describe how you built tables and imported data.

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.

5.3 Submission for Bonus Task: (10 Points)

As part of the bonus task, students must include a link to their published dashboard within the project report. Additionally, each visualization should be accompanied by a brief analysis (2-3 sentences) explaining its insights.

- Data Connection & Queries (5 pts) → SQL queries used to extract data.
- Dashboard Design (3 pts) → Effective visual representation & interactivity.
- Insights & Explanation (2 pts) → Clear interpretation of data trends.

CSE Demo Days (Optional)

If you get interesting results, we would encourage you to share your project with the public in terms of participating in the CSE Demo Days. CSE Demo Days is a semester event where you can highlight your project results. This is a fun Event. Send us your prior results before 27th April. Selected teams will have to prepare a poster and present it.

Final Project Directions

Below is a list of possible domains for your final project:

- Healthcare
- Finance
- Education
- Retail
- Environmental Monitoring
- Sports

Note: It is not mandatory to select a domain from the above list. This list only provides direction on topics you can choose from.

Once you come up with a use case in a domain of your choice, the next step is to select a database that is suitable for solving your use case. Below are a few open-source database websites:

- **IMDb:** Makes their movie database available here.
- **Data.gov:** A huge compilation of datasets produced by the US government is available here.
- **The Supreme Court Database:** Tracks all cases decided by the US Supreme Court. Details can be found here.
- **US Government Spending Data:** Information about government contracts and awards is available for download here.
- **Federal Election Commission:** Campaign finance data can be downloaded from here. Their "disclosure portal" also provides nice interfaces for exploring the data, which can be accessed here.
- **Historical Stock Quotes:** Can be downloaded and scraped from many sites such as Yahoo! Finance and Google Finance.
- **Open-Source Datasets:** You are allowed to use any open-source datasets.