CS-GY 6083-A, Principles of Database Systems, Fall 2022

Course Project: A database application with a Web front-end

General Information

This assignment is worth *25% of the overall grade*, and is roughly equivalent to 3 homeworks in terms of workload. Do not wait until the last day to get started on the project. You will not be able to finish the project one or two days before the deadline.

In this assignment you will build a database application of your choice. In part 1 you will develop a conceptual design of your application with an ER model and translate the ER model to a relational schema. We will give you feedback on part 1, which you should incorporate when working on part 2 of the project. In part 2 you will make any necessary adjustments, populate your database schema with data, and implement a simple Web application that queries and updates your database using the python package streamlit.

This assignment is to be completed in teams of two. Consult the course syllabus for a description of our academic honesty policy. Both partners are expected to equally participate in all parts of the project. You may discuss all aspects of the project with your partner. However, you may not discuss project details with your other classmates.

Action Items and Deliverables

1. **Part 1: by October 14 @ 11:59pm ET,** submit a text description of your application, an ER diagram, and a translation of your ER diagram to the relational model.
2. **Part 2, by December 16 @ 11:59pm ET**, submit your final project report, code, and presentation (slides and recording).

Submission instructions

For part 1, submit your assignment on BrightSpace. Both project partners should submit part 1 of the project by the deadline, with identical submissions. Submit all parts of the assignment (including schema.sql) as a single PDF file, and also include schema.sql as a separate file. Note that all project deliverables must be submitted by the specified deadline. You cannot use any late days for project submissions.

For part 2 submission instructions, see last section in this document.

Part 1 details

A. Be sure to choose an application that you will enjoy working with, as you will be committed to this choice for a significant part of the course. Think about whether it will be easy to find real data for this application, or to create some realistic data. We do not require that your application work with real data, but that usually makes the application more interesting.

B. You should submit a brief **text description** of the application. The text description of your application should be around 2 pages. You should:

   a. give a high-level description of your application,

   b. explicitly list all entity sets, relationship sets, and business rules that hold in your application; as you describe the business rules, you should describe all key and participation constraints,

   c. explain how you plan to acquire data for your application, and

   d. discuss how the user will interact with your database - what kinds of questions they will be able to ask.

C. Your ER model should be based on the project description. **It should have 8-10 entity sets and a similar number of relationship sets.** If you are worried that your design is too simple, or too ambitious, you are more than welcome to discuss this with the instructors by email or **during office hours**. We will check the correctness of your ER diagram against your specification of the entity sets, relationship sets, and business rules, described above.

D. We recommend drawing the **ER diagram** by hand and scanning it in. Use ER notation we covered in class, do not use any other notation, or any software that generates ER diagrams if it does not output the same notation as what we covered. Remember to include the names of both team members in your submission.

E. Translate your ER model into a relational schema. Load this schema into your PostgreSQL database to check for correctness. As a result of this step, you should create a file called **schema.sql**, containing the right drop table and create table statements. Submit that script along with the project description and the ER diagram.

We will give you feedback on your submission for Part 1. You should revise your submission using this feedback and use this revised version as the basis for Part 2 of the project. You will be required to submit a revised project description, ER diagram, and relational schema as part of the final project report.

Part 2 details

- Modify your project description, ER diagram, and schema.sql, incorporating any feedback you received from us.
- Implement a data loading procedure. This procedure can be as simple as including insert statements into your schema.sql. You may also use the psql copy command to load data into your tables, see Documentation: 12: COPY for additional information. Finally, you may write custom code to populate the tables in your database tables with data. Whatever option you choose, you should load 20-100 tuples of real or realistic data into each table in your database schema, keeping in mind that the queries you are implementing should give interesting results. Carefully describe your data loading procedure in the project report.
- Carefully think about the interactions the user will have with your application, and implement them as part of the Web front-end. Your Web interface does not need to be fancy, and you will not receive extra points based on how fancy the interface looks. However, **the user should not be asked to type in SQL queries**.
- Most interactions should involve some sort of input values. Input values may be specified using menus, radio buttons, checkboxes, scrollers, etc. Text input boxes may also be appropriate. As a result of the user's interaction, you should be generating a query that is executed against your database schema, and displaying results at the next step.
- You are required to implement at least 5 kinds of user interactions that lead to executing a query with a join, a group by, or both. You may decide to implement additional interactions that select data from 1 table at a time, or that insert, update or modify data. Still, **at least 5 of the queries you implement must involve a join, a group by, or both**.
- You will implement the Web front-end of your application in python using Psycopg2 and Streamlit library. We demonstrated how to use Psycopg2 and Streamlit during the lab, please refer to lab materials for details. Some useful pointers are below:
    - Psycopg2 official documentation: Psycopg – PostgreSQL database adapter for Python — Psycopg 2.9.1 documentation
    - Streamlit official documentation: Streamlit Docs
    - Project demo folder, including:
        - Streamlit setup on jedi: Deploying a Streamlit Application.
        - FAQ (we will update this continuously): Project FAQs

- The web-based application you develop in part 2 must run on jedi.poly.edu. Make sure that your application is deployed and running on one of the ports assigned to your team when you submit your final project report. **Make a note of the port number on which your application is deployed at the top of your project report.** We will interact with your application on jedi.poly.edu while grading your submission.
- You should prepare slides for a 3-minute presentation describing your application, and record a video with your presentation.
- Each project partner should prepare a statement of contributions: a document describing their own and their partner's contributions to the project. **This part of the assignment is to be completed individually.** Do not discuss the content of this document with your partner, and do not collaborate on this document.

Submission instructions

For part 1, submit your assignment on BrightSpace. Both project partners should submit part 1 of the project by the deadline, with identical submissions. Submit all parts of the assignment (including schema.sql) as a single PDF file, and also include schema.sql as a separate file.

For part 2, submit your assignment on BrightSpace. Both project partners should submit the complete project by the deadline, with submissions that are identical except for the contributions statement. Submit a project report, including a description of the project, the business rules, the ER model and schema.sql, as a single PDF file. Also submit schema.sql as a separate file. Include all data and code that implements your data loading procedure. Finally, include the python file that implements the UI in your submission.

Submit a zip file with the name **<netid1>_<netid2>_project.zip** (e.g., abc123_xyz456_project.zip, with project partners' netIds in alphabetical order) with the following structure (code and data are folders):

- abc123_xyz456_project (zip archive)
    - **report.pdf** - project description, business rules, ER diagram
    - **contributions.pdf** - each project partner should draft this separately, explaining their own and their partner's contribution to the project
    - **code/**
        - **schema.sql** (SQL File)
        - **project.py** (Python file)
    - **data/**
        - load.sql with insert statements and/or
        - data file(s) in CSV format and/or
        - code that implements your data loading procedure