ENSF 608 Fall 2022
Department of Electrical and Software Engineering
Schulich School of Engineering

The objective of this project is to apply your understanding of course concepts, database design, and SQL queries on a real-world database application. If you are taking ENSF 607 your group should be same in both courses. The software for this you will be developing in ENSF 607.

Due: Final document for Project is due : Thursday, December 2nd, 11:59 PM on d2l

Submission: This is a team assignment. You may work in groups of 3. Your submission must be your team's own original work. Only one team member needs to submit the final files, but all names should be included on each document and within the dropbox description. Teams should verify the file upload process together- if one member uploads an incorrect file, it will impact the grade of the entire team.
You can also choose the project of your own choice or you can work on Project given in this document

There are multiple components to the project. Your submission should consist of five files:

Task	**Deadline of Submission on dropbox**
- Project Proposal	Nov 18, 2022
- A single .pdf file with your EER diagram	Nov 18, 2022
- A single .pdf file with your relational model	Nov 25, 2022
- A single .sql file to build and populate your database	Dec 2, 2022
- A single .sql file with your query demonstration	Dec 2, 2022
- An .mp4 or link to your video demonstration	Dec 2, 2022

Please upload your submission to the project D2L dropbox folder.

Weighting: You will receive a project grade out of 100%. It is worth 30% of your overall grade.

Note: In ENSF 608, you will only be graded on your database design. Make sure the video is of high quality without any background noise. Be mindful of maximum 5 minutes time for video.

Grading:
The EER diagram and relational data model should follow the formatting conventions outlined in the lecture notes. Your solution may be computer generated or hand-drawn but must be legible.

The ER diagram should include correct notation for entity types, relationship types, attributes, key attributes, relationship attributes, and cardinality. All relations should have a name, primary key, attribute(s) as necessary, and foreign key(s) as necessary. Use arrows to represent foreign keys (referential integrity).

Your SQL files will be run through MySQL Workbench. All statements must compile and execute correctly to receive marks.

Marks will be deducted for incorrect or missing information. Solutions must be neat and organized.

ENSF 608 Project Deliverables (30% of course grade)

Conceptual Database Design (33.33% of 608 project grade, 10% of course):

Based on the provided requirements narrative, design and draw an EER diagram for the described database application. Your solution should follow the model notation presented in class and should include cardinality ratios and participation constraints. Your diagram should also be accompanied by a half-page description explaining your design decisions and any assumptions that were made.

Your solution may be handwritten or typed, and you may draw your diagram by hand or by using software tools. Handwritten work may be scanned or clearly photographed. Marks will be deducted for incorrect or missing information based on the provided narrative. Solutions must be neat and organized.

Logical Database Design and Creation Code (33.33% of 608 project grade, 10% of course):

Map your conceptual schema into a relational data model, including all primary keys and referential integrity constraints (foreign keys). Then use your relational model to create a .sql script that could be used by someone else to initialize and populate your database. You are free to use the given example data or create your own.

Your solution may be handwritten or typed, and you may draw your diagram by hand or by using software tools. Handwritten work may be scanned or clearly photographed. Marks will be deducted for incorrect or missing information based on your EER diagram design. Solutions must be neat and organized.

Query Code and Video Demonstration (33.33% of 608 project grade, 10% of course):

Implement your database in MySQL Workbench. Use Zoom or another tool of your choosing to record a short demonstration of your database (5 minutes or less!). All team members should be part of the demonstration. In the demonstration, show the following elements using your database. In your submission, you should include a .sql file that contains the queries listed below:

1) Show all tables and explain how they are related to one another (keys, triggers, etc.)
2) A basic retrieval query
3) A retrieval query with ordered results
4) A nested retrieval query
5) A retrieval query using joined tables
6) An update operation with any necessary triggers
7) A deletion operation with any necessary triggers

Marks will be deducted for incorrect or missing information and may also be deducted for videos of excessive length. Videos should have clear audio. Cameras should be on or you may use a professional- looking headshot. This is your opportunity to demonstrate how your database works and why it is a correct solution for the client.

The objective of this project is to apply your understanding of course concepts, database design, and SQL queries on a real-world database application.

Submission: This is a team assignment. You may work in groups of 3. Your submission must be your team's own original work. Only one team member needs to submit the final files, but all names should be included on each document and within the dropbox description. Teams should verify the file upload process together- if one member uploads an incorrect file, it will impact the grade of the entire team.

There are multiple components to the project. Your submission should consist of five files:

Using the model for the Course Registration System (15 Marks)

In this Project, you are asked to add to the model package as needed and develop the viewController package for the Course Registration System that we have been working on during class. Your tasks are as follows:

Task 1 –Create a database for the course registration system:

- Students are supposed to be able to select their courses from the course-catalogue and register for a maximum of 6 courses.
- Each course, may be offered in 1 or more sections (offerings), must have a minimum of 8 students to run, and may have one or more prerequisites.

Task 2 - Implement a console-based menu to properly call the methods from the model (i.e. backend). The program should keep on running (i.e. should keep presenting the user with the menu) until the user quits.

To keep things simple, you can create a few objects of types Course, Student, and Course Offering in main. (i.e. simulate having a file or database). Your menu should contain the following items:

1. Populate tables for the course catalogue
2. Show all tables and explain how they are related to one another (keys, triggers, etc.)
3. A basic retrieval query
4. A retrieval query with ordered results
5. A nested retrieval query
6. A retrieval query using joined tables
7. An update operation with any necessary triggers
8. A deletion operation with any necessary triggers
9. Add and Remove course from student courses
10. View all courses in the catalogue
11. View all courses taken by the student
12. Quit

Important note 1: You must use nested queries, joins, view, update, remove, add a course

Important note 2: You must implement any missing functionality in your model. For example, you must add queries to implement remove course. Write all the assumptions you made.

Important Note 3: You can choose to develop a GUI instead of a console-based menu if you have a good handle on event-based programming.