# Final Design Project

**ENSF 300 Fall 2023**

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.*

**Due:** Monday, December 4th, 11:59 PM

**Submission:** This is a group assignment. 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 along with the GitHub repo link. 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:

- A single .pdf file with your EER diagram
- A single .pdf file with your relational model
- A single .sql file to build and populate your database
- A single .sql file with your query demonstration
- GitHub link must be available in the submission comment

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.

**Grading:**

The ER/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/EER 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. **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 300 Project Deliverables (30% of course grade)

**Conceptual Database Design (25% of project grade, 7.5% 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 (25% of project grade, 7.5% 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 (25% of project grade, 7.5% of course):**

Implement your database in MySQL. In your submission, you should include a .sql script for the database creation and 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

**Python Application (25% of project grade, 7.5% of course):**

Implement an application that will connect to the database to be used for database maintenance, data entry, and browsing. The application will require a log in to identify the type of user and their corresponding access level and privileges. The different type of users and their required functionality are stated at the end of the handout.

Your application must have the following:

1- Admin interface which will allow the administrator to:
    a. Type sql commands that will be executed on MySQL through your python application using the connector
    b. Provide the path and file name of an sql script file to the application, which will then run the scrip file on MySQL using the connector
2- Data entry interface, which will allow employees to:
    a. Lookup information in the database by providing search field values (not SQL commands)
    b. Insert new tuples to a specific table in the database by first selecting the table for data insertion, then either:
        i. Providing a file with information line separated, where each line represents an entry that should be made to the table of choice
        ii. Providing a sequence of entries through detailed prompts ( for example ask the user to provide art piece name, then upon entering the value ask for the data, followed by type, etc…)
    c. Update and delete tuples in the database by providing search field values. Make sure to show appropriate messages for successful updates and deletions, and also descriptive messages for failed attempts
3- Browsing interface, which will only be for data lookup. This is intended for an end user that wants to browse the database. For this type of user you can have the option of adding a skip option at the login stage to login as guest. The user might not be familiar with the data base content so use a descriptive user friendly prompts to guide the user through browsing the database.
    a. For example show an option menu for the user to select what to browse (1- art pieces, 2- exhibitions, …). This will be a multi level menu to help the user reach their selection with option to go back to the upper menu

## Question Narrative

You are tasked with developing an application for an art museum to maintain information about their art objects database. The application will interface a database that you will design and create to hold the required information, while the application will be used for data entry, maintenance, and information retrieval. The following requirements have been observed and collected from discussions with the customer:

- The museum has a collection of ART_OBJECTS. Each ART_OBJECT has a unique Id_no, an Artist (if known), a Year (when it was created, if known), a Title, and a Description. The art objects are categorized in several ways, as discussed below.
- ART_OBJECTS are categorized based on their type. There are three main types—PAINTING, SCULPTURE, and STATUE—plus another type called OTHER to accommodate objects that do not fall into one of the three main types.
- A PAINTING has a Paint_type (oil, watercolor, etc.), material on which it is Drawn_on (paper, canvas, wood, etc.), and Style (modern, abstract, etc.).
- A SCULPTURE or a statue has a Material from which it was created (wood, stone, etc.), Height, Weight, and Style.
- An art object in the OTHER category has a Type (print, photo, etc.) and Style.
- ART_OBJECTs are categorized as either PERMANENT_COLLECTION (objects that are owned by the museum) and BORROWED. Information captured about objects in the PERMANENT_COLLECTION includes Date_acquired, Status (on display, on loan, or stored), and Cost. Information captured about BORROWED objects includes the Collection from which it was borrowed, Date_borrowed, and Date_returned.
- Information describing the country or culture of Origin (Italian, Egyptian, American, Indian, and so forth) and Epoch (Renaissance, Modern, Ancient, and so forth) is captured for each ART_OBJECT.
- The museum keeps track of ARTIST information, if known: Name, DateBorn (if known), Date_died (if not living), Country_of_origin, Epoch, Main_style, and Description. The Name is assumed to be unique.
- Different EXHIBITIONS occur, each having a Name, Start_date, and End_date. EXHIBITIONS are related to all the art objects that were on display during the exhibition.
- Information is kept on other COLLECTIONS with which the museum interacts; this information includes Name (unique), Type (museum, personal, etc.), Description, Address, Phone, and current Contact_person.

Consider which elements of the functionality will need to be **stored in the database** and how they **interact** with one another. Your design may be different from others- you have flexibility in how you choose to meet the requirements of the client.

**What is the goal?**
Our goal is to create an application (command line minimum requirement, optional GUI or web application or both) for the museum to help manage their arts database and events such as exhibitions. This application will address the following 3 areas:

1. Manage art objects
    a. Add art pieces to the database
       Every animal can have different properties based on type but some properties are the same for example (Name, sex, age, RFID).
    b. Search art pieces and exhibitions.

2. User Management
    a. Add users
    b. Access control
    c. Manage users (such as block and suspend)

When designing your database, consider the following user requirements and functionalities:

All the users must log in to the application.

**Admins**
- Can add users.
- Can edit users.
- Can block users.
- Can make changes to database (modify tables attributes and constraints)

**Data entry users**
- Can add information tuples to the database as long as the information being added meets the database constraints
- Can modify existing information in the database

**End user**
- Can lookup information in the database (ideally a remote user on a website that browses the offerings of the museum, but we will have it simulated as part of the main program).