

## **CSCI 4850/8856 Project**

### **Fall 2025**

Your project entails designing and building a database for an application of **your choice**. You must use SQLite or MySQL to implement your project.

CSCI4850 students: This project can be done in teams of at most two students.

CSCI8856 students: This project must be done individually.

The project consists of the following deliverables where undergraduates the first two files (60+40) and graduate all files (50+30+20):

#### **1: Data Definition with E-R Diagram:**

**All students must submit a PDF file.**

#### **2: SQL Queries**

**1. All students must submit the DB file.**

**2. All the query statements used to create, insert, update, query the database.**

#### **3: Python Script file or a language of your choice.**

#### **4: Deadline: 11:59 pm of Friday 12/12/2025**

### **1 – Data Definition**

There should be at least six relations (more is better) in your relational data model. The application must be rich enough to support the subsequent part of the project. The relations must be set up so as to allow complex queries such as set operators, nested queries, aggregation and group by. You can choose any of the project list here (<https://www.db-book.com/university-lab-dir/lab-exercises-projects.html>), like retailer database, real estate database, if you find it hard to think of an application scenario.

#### **Hand In:**

1. An English description of your application. (5|5)
  - a. The description must explain all details of your applications such as what kind of data is being stored, what types of queries are expected, and how do these queries help your application.
  - b. Your description must include an explanation of your E-R diagram along with various decisions you made about how to model your application.
  - c. Your description must also discuss the correspondence between the E-R model and the relational model by discussing the decisions you made about the primary keys, mapping cardinalities, participating constraints, weak entities and specialization hierarchies. Your document must include, for each E-R feature, how it is captured in the relational model.
2. An *Entity - Relationship* diagram for your system. (30|25)
  - a. **You will be graded on how many of the E-R features you used in your application.**
    - i. Different types of attributes.
    - ii. Mapping Cardinalities.

- iii. Participation constraints.
  - iv. Weak entities
  - v. Specialization/generalization hierarchies and constraints.
3. The SQL specification of your relational database, expressed in the SQL Data Description Language. This specification must include a CREATE TABLE for each of your relational tables. You must specify a primary key for each table and define any foreign key constraints (20|15)
  4. Sample instances of your relational tables, containing some sample data. (5|5)

## 2 – SQL Queries

Populate your database with sample data. There should be enough data and it should have sufficient structure, so that the queries have enough data.

Then write a variety of SQL queries. There should be around two dozen queries per student. They should range from routine to sophisticated, and some should be challenging. Make sure you use all of simple as well as advanced SQL query types discussed in class from Chapters 3 and 4.

Hand in: SQL files like the textbook website, table\_creation.sql, data\_insertion.sql, query.sql with comments. *The files must be uploaded to Canvas.*

1. All SQL queries are required to create your tables and populate them with data.
2. Each of your SQL queries, and the results it produced. For each query, also provide a comment of English description of what it is doing.

Each student will write his/her own distinct queries. Your grade will depend on the diversity of your queries -- create tables, queries that show that constraints (foreign key, primary key, unique, and check) are enforced, all types of join expressions discussed in Chap 4 including outer joins, renaming, string operations, Group by, aggregating, having, and order by, Set union, intersection, and minus, nested queries: for set membership, set comparison, testing for empty relation and absence of duplicate tuples, nested queries in the From clause, With Clause, Deletion of tuples, updating of tuples, inserting of tuples selected from one table into another table.

## 3 – Python script (20 points)

This involves the following requirements– (This part may subject to change in around week 10)

- 1) A Python or language of your choice script to execute some queries on the database you created in Part 2. Design up to 5 queries that demonstrate different key aspects of your system. Write a Python script that connects to your database, executes the queries, and outputs the query results. A simple implementation can “hard code” the queries in the Python script, with no input accepted from the user. When the Python script is executed, the output is simply printed on the screen.

