Final Project Instructions

Course: Relational Databases (Fall 2025) - PROG2111

Total Marks: 100 Group

Work: 3 Members

Project Overview:

In this final project, you will work as a team of four to design, implement, and programmatically interact with a relational database. You will choose a use case that requires relational database support, model the data, design an Entity-Relationship Diagram (ERD), normalize the data, implement the database using SQL Data Definition Language (DDL), and perform Create, Read, Update, and Delete (CRUD) operations using C# and ADO.NET MySQL API.

Objectives:

By the end of this project, students will be able to:

1. Model a relational database for a real-world use case.
2. Create an ER diagram based on their data model.
3. Normalize the database to 3rd Normal Form (3NF).
4. Use SQL DDL to create tables with appropriate keys and constraints.
5. Write C# programs that interact with the MySQL database to perform CRUD operations.
6. Demonstrate programmatic access to the database using MySQL API.

Project Phases:

Phase 1: Project Idea & Use Case (5 Marks)

- Choose a real-world scenario where a relational database is essential (e.g., a library management system, hospital management system, e-commerce site).
- Create a GitHub repository and add all members to it
- Make sure all members have access before you submit Phase 1 document
- Follow the date and timeline for submission on eConestoga

Phase 2: Data Modeling & ER Diagram (20 Marks)

- Identify the key entities, attributes, and relationships in your use case.
- Develop an Entity-Relationship Diagram (ERD) using appropriate diagramming tools.
- The ERD must include all entities, their attributes, primary keys, and relationships between them.

Deliverable: ERD in PDF format.

Phase 3: Normalization (15 Marks)

- Normalize your database to 3rd Normal Form (3NF).
- Submit documentation showing the process of normalizing the database from 1NF to 3NF.
- Clearly explain how you handled redundant data and ensured that data dependencies are logically organized.

Deliverable: Normalization report.

Phase 4: DDL Statements (20 Marks)

- Based on the ERD, write SQL DDL statements to create the database schema.
- Ensure that all necessary primary keys, foreign keys, and integrity constraints are included.

Deliverable: SQL script with DDL statements to create tables.

Phase 5: Programmatic Access & CRUD Operations (30 Marks)

- Develop C# programs that connect to the MySQL database using the ADO.NET
- Perform CRUD operations (Create, Read, Update, Delete) for the entities defined in your ERD.
- Ensure proper error handling and transaction control.
- Use the ADO.NET MySQL library to establish the connection and perform operations.

Deliverables: C# programs with CRUD functionality.

Phase 6: Final Presentation & Code Walkthrough (10 Marks) Present

your database project to the class. The presentation should:

1. Explain the project use case.
2. Walk through the ER diagram.
3. Demonstrate CRUD operations using the C# programs.
4. Discuss any challenges and how your team overcame them.

Project Guidelines:

Teamwork: Work as a cohesive team of three members. Each member should have a defined role, such as data modeling, coding, database design, etc.

Documentation: Keep a log of each team member's contributions, and submit it with the final deliverables.

Programming Language: All programmatic access should be done using C#, and SQL queries should be executed through the MySQL API.

Database: Use MySQL as the relational database system.

Assessment Criteria:

Clarity of the use case: 5 Marks

Quality of the ERD and normalization: 35 Marks

Correctness of the DDL statements: 20 Marks

Functionality and completeness of the C# programs: 30 Marks

Presentation quality and project explanation: 10 Marks

Submission Format:

- Upload all deliverables (ERD, DDL, C# programs, normalization report) in a zip file.

- Include the name of each team member and their contribution in the project log

GitHub Repo

- Create GitHub repo and Submit the link with Phase 1
- Make sure all the members have access to the GitHub before submitting Phase 1 document
-