

Database Management Systems Course

Project Assignment

General Description

In this project, students are expected to **design and develop a relational database system** that addresses a real-world business problem. The project aims to prepare students for their **graduation projects** by providing practical experience in **database analysis, design, and implementation**.

- The project can be developed **individually or in groups of up to 3 students**.
- The project consists of **two main phases**:
 1. **Database Design Phase**
 2. **Database Implementation Phase**

Phase 1 – Database Design

In the first phase, students will focus on designing a database system that solves a specific business or organizational problem. The following steps should be followed:

1. **Problem Definition and Requirements Analysis**
 - Identify a real-world problem that requires database support.
 - Clearly define **user needs** and **system requirements**.
 - Prepare a short **requirement specification document** describing the problem, the users, and the data needs.
2. **Conceptual Data Model**
 - Create an **Entity–Relationship (ER) Diagram** that models the main entities, attributes, and relationships.
 - Ensure that all business rules and constraints are represented properly.
3. **Logical Data Model**
 - Transform the conceptual model into a **logical schema** suitable for a relational database.
 - Define **primary keys, foreign keys, and relationships** explicitly.
 - Normalize all tables to an appropriate level (at least **3rd Normal Form (3NF)**).

4. Physical Data Model

- Convert the logical schema into a **physical data model** that can be implemented in a database management system.
- Use **UML class diagrams** to represent the physical model.
- Include table structures, data types, and constraints.

5. Documentation

- Submit a report including:
 - Problem description and user requirements
 - Conceptual, logical, and physical data models
 - ER and UML diagrams
 - Normalization steps and explanations

Phase 2 – Database Implementation (Updated)

In the second phase, students will implement the physical data model they developed in Phase 1. The goal of this phase is to create the actual database on a physical disk, define all tables, and populate them with sufficient sample data. Students may use **Microsoft SQL Server** or **PostgreSQL** as their database management system.

The following tasks must be completed in this phase:

1. Database Creation

- Create a new database in the selected DBMS (MS SQL Server or PostgreSQL).
- Include the necessary SQL statements for database creation in the report.

2. Table Creation

- Implement all tables defined in the physical data model.
- Use appropriate data types, primary keys, foreign keys, constraints, and indexes.
- Provide complete CREATE TABLE SQL scripts for each table.

3. Data Insertion

- Insert a sufficient amount of sample data into each table to demonstrate that the system works as intended.
- Include all INSERT INTO statements in the report.

- Ensure that inserted data follows referential integrity rules.

4. Verification of Database Structure

- Verify that all relationships, constraints, and business rules defined in Phase 1 are correctly implemented.
- Ensure consistency between the physical model and the implemented database.

5. Documentation

- Submit a report including:
 - The SQL code for creating the database and all tables
 - The SQL statements used to insert sample data
 - Descriptions of how constraints, relationships, and business rules were implemented
 - Screenshots (optional) showing the successful execution of SQL statements or populated tables

Further instructions, including advanced query requirements or stored procedure tasks, may be provided later.

Evaluation Criteria

Projects will be evaluated based on the following aspects:

- Completeness and clarity of requirements
- Correctness and normalization of data models
- Quality and accuracy of ER and UML diagrams
- Consistency between design and implementation
- Overall documentation quality and presentation