**Comprehensive Database Project Development**

**Course:** Database Systems / Advanced Database Systems  
**Instructor:** Dr. Polla Fattah  
**Due Date:** 2025-03-01

**Objective**

The goal of this assignment is to apply database design and development principles in a real-world scenario by creating a **non-trivial** database-driven application. This project will involve:

* **Understanding client requirements**
* **Designing an efficient database schema**
* **Implementing CRUD operations**
* **Managing user authentication & roles**
* **Creating complex queries for reporting and aggregation**
* **Utilizing views, triggers, and stored procedures**
* **Optimizing database performance**
* **Ensuring security best practices**

This project will help you gain hands-on experience in database system development, SQL programming, and database management system (DBMS) features.

**Project Guidelines**

Each team (or individual) must select a **realistic** use case and implement a **fully functional** database system. Below are the requirements:

**1. Project Selection and Client Requirements**

* Identify a **real-world** application scenario (e.g., hospital management, university portal, online marketplace, inventory system, hotel booking system, etc.).
* Collect and document **client requirements** as if you are building a real system for an organization.
* Prepare a **Software Requirements Specification (SRS)** document, including:
  + **System objectives**
  + **User roles & privileges**
  + **Entity relationships**
  + **Functional requirements**
  + **Non-functional requirements**

**2. Database Design and Implementation**

* Design the **Entity-Relationship (ER) Diagram** using **Crow’s Foot Notation**.
* Convert the ER model into a **relational schema**.
* Implement the database using **PostgreSQL / MySQL / Oracle**.
* Ensure **proper normalization** (at least 3NF).

**3. CRUD Operations**

* Implement **Create, Read, Update, and Delete (CRUD)** operations for **all required entities**.
* Create **stored procedures/functions** for common database operations.
* Implement **transaction handling** for multi-step processes.

**4. User Authentication & Authorization**

* Implement **user authentication (login/logout)** using hashed passwords.
* Define **user roles and permissions** (e.g., Admin, Manager, Regular User).
* Ensure that **only authorized users** can perform specific actions.

**5. Complex Queries & Reporting**

* Implement **aggregate queries** (SUM, COUNT, AVG, MIN, MAX, etc.).
* Use **JOINs and subqueries** for reporting.
* Generate **custom reports** based on user input (e.g., sales reports, activity logs).
* Implement **parameterized queries** to allow filtering of reports dynamically.

**6. Views & Stored Procedures**

* Create **multiple database views** for different user groups (e.g., a sales dashboard for managers).
* Write **stored procedures** to automate common tasks (e.g., sending notifications, generating invoices).

**7. Triggers & Constraints**

* Implement **triggers** to enforce business logic (e.g., updating stock levels when an order is placed).
* Define **constraints** (CHECK, UNIQUE, FOREIGN KEY, PRIMARY KEY) to ensure **data integrity**.

**8. Indexing & Performance Optimization**

* Use **indexes** to improve query performance.
* Analyze **query execution plans** to optimize performance.
* Avoid **redundant data retrieval** using **materialized views** if needed.

**9. Security & Backup**

* Apply **data encryption** where necessary (e.g., storing passwords securely).
* Implement **role-based access control (RBAC)**.
* Create **database backup & recovery strategies** (e.g., automated backups, snapshots).

**Submission Requirements**

Each team must submit the following deliverables:

**1. Documentation (Report)**

* **Project Proposal (2-3 pages)**
* **Software Requirements Specification (SRS)**
* **ER Diagram & Database Schema Design**
* **Use Case Diagrams & Description of System Functionalities**
* **Authentication and Role-Based Access Implementation**
* **Query Optimization Strategies Used**

**2. SQL Implementation Files**

* SQL script to **create the database schema**.
* SQL scripts for **CRUD operations, views, stored procedures, triggers, and indexes**.

**3. Sample Data & Queries**

* Populate the database with **at least 50-100 records per table**.
* Provide at least **5 complex queries** with sample results.

**4. Source Code of Web or CLI Application (Optional)**

* If you build a web or command-line interface, submit the **source code** and README.
* The application must connect to the database and perform **real-time CRUD operations**.

**5. Final Presentation & Demonstration**

* A **10-15 minute** live demonstration of the project.
* Explain **system architecture, design decisions, and challenges faced**.

**Grading Criteria**

| **Criteria** | **Marks** |
| --- | --- |
| Project Requirements & Documentation | 20 |
| Database Design & ER Model | 15 |
| CRUD Operations Implementation | 10 |
| User Authentication & Role Management | 10 |
| Advanced Queries & Reporting | 10 |
| Views, Stored Procedures, and Triggers | 10 |
| Indexing & Performance Optimization | 10 |
| Security Measures | 10 |
| Presentation & Demonstration | 5 |
| **Total** | **100** |