Advanced Database Management Systems Lab

Basic Lab Program

- ☐ **Refer this video for installation of mysql**
  - ☐ **How to install MySQL 8.0.35 Server and Workbench latest version on Wi...**
  - ☐ **Note**: During the installation process, at one step you will be asked to set the **MySQL Root Password** and **Repeat Password**. Ensure that both the password is always set to **"admin."** While installing on your personal computer, you can choose any password, but when installing in the computer lab, make sure the password is set to "**admin**."
- ☐ **Once the installation is completed, create a path for sql in Environment Variables.**
  - ☐ https://www.youtube.com/watch?v=iPjPFjyOKmw

Steps to be followed once the installation is complete

- ☐ **To check the version of mysql**
  - ☐ In command prompt type
    - ☐ **mysql –version**
- ☐ **To start the server**
  - ☐ **mysql -u root -p**
  - ☐ It will ask for a password, then enter the password.
  - ☐ If the password is correct, then it will show the success message.
- ☐ **To check the number of databases we have**
  - ☐ **show databases;**
  - ☐ It will display the table of the database.
- ☐ **To create a database**
  - ☐ **create database databasename;**
    - ☐ **Example:** create database **school**;
- ☐ **Using a Particular database**
  - ☐ Since we have created a database called **school**, we have to use the **banking** database as follows
    - ☐ **use database;**

First Question to be executed and write all the commands for the following question

Problem Statement

- ☐ **Create a database for a simple school management system. The system should have the following tables:**
  - ☐ Students
  - ☐ Courses
  - ☐ Enrollments
  - ☐ Instructors

Perform the following tasks step-by-step using SQL commands.

Tasks

- ☐ **Create the Database and Tables**
  - ☐ **Create a database named SchoolDB.**
  - ☐ Create the Students table with the following columns:
  - ☐ StudentID (Primary Key)
  - ☐ FirstName
  - ☐ LastName
  - ☐ BirthDate
  - ☐ Gender
  - ☐ EnrollmentDate
- ☐ **Create the Courses table with the following columns:**
  - ☐ CourseID (Primary Key)
  - ☐ CourseName
  - ☐ Credits

- ☐ **Create the Enrollments table with the following columns:**
  - ☐ EnrollmentID (Primary Key)
  - ☐ StudentID (Foreign Key referencing Students.StudentID)
  - ☐ CourseID (Foreign Key referencing Courses.CourseID)
  - ☐ Grade
- ☐ **Create the Instructors table with the following columns:**
  - ☐ InstructorID (Primary Key)
  - ☐ FirstName
  - ☐ LastName
  - ☐ HireDate
- ☐ **Insert Data into Tables**
  - ☐ Insert at least 3 records into the Students table.
  - ☐ Insert at least 3 records into the Courses table.
  - ☐ Insert enrollment records for students in various courses into the Enrollments table.
  - ☐ Insert at least 2 records into the Instructors table.
- ☐ **Query Data**
  - ☐ Retrieve all student records.
  - ☐ Retrieve all courses with their course names and credits.
  - ☐ Retrieve all enrollments showing StudentID, CourseID, and Grade.
  - ☐ Retrieve all students enrolled in a specific course (use a JOIN operation).
- ☐ **Update Data**
  - ☐ Update a student's last name.
  - ☐ Update a course's credits.
  - ☐ Update a grade for a specific enrollment.

- ☐ **Delete Data**
  - ☐ Delete a student record.
  - ☐ Delete a course record.
  - ☐ Delete an enrollment record.
- ☐ **Use Basic SQL Functions**
  - ☐ Count the total number of students.
  - ☐ Find the average grade for a specific course.
  - ☐ Retrieve the maximum and minimum grades from the Enrollments table.
- ☐ **Apply Constraints**
  - ☐ Add a NOT NULL constraint to the FirstName and LastName columns in the Students table.
  - ☐ Add a UNIQUE constraint to the CourseName column in the Courses table.
  - ☐ Add a CHECK constraint to ensure that Grade in the Enrollments table is between 0 and 100.
- ☐ **Use the LIKE Operator**
  - ☐ Retrieve students whose first names start with 'A'.
  - ☐ Retrieve courses whose names contain the word 'Math'.