

Final Test: Databases and interoperability with C#

The text:

Requirements for UVIC Management System

UVIC wants to develop a new system for managing various aspects of their operations. They require a comprehensive database to store information about students, courses, professors, classrooms, departments, enrollments, teaching assignments, and class schedules. Additionally, the university employs different types of staff (administrative, maintenance, security) each requiring specific additional information.

The university intends to develop an initial part of the application to evaluate if the developer's knowledge is sufficient to cover all their needs.

Application Requirements

The university requires a Console Application developed in C# to interact with the database. The application should have a main menu and provide the following functionalities:

1. **Connect to the database and return the version of MariaDB.**
2. **Add a new row:** If the user selects this option, they need to specify the table, and a new hardcoded row will be added to that table.
3. **Select table:** Ask the user which table they want to list and display the rows from that table.
4. **Drop the last element of a specific table:** Ask for the table and delete the last row.
5. **Select values of two related tables:** Ask for the two tables and list the related values (e.g., students and the courses they are enrolled in).

Additionally, in the second part, the university expands the system to include login functionalities where:

- **Students** can view their timetable, completed and missing assignments, and grades.
- **Professors** can add, remove, and modify student grades, add assignments, and view their teaching schedule.

Hiba Nejari

- **Administrative staff** can manage student enrollments and professor assignments, as well as view a list of the professors, students, and add/delete professors.
- **Maintenance staff** can view and manage maintenance tasks.
- **Security staff** can view and manage security shifts.

Implementation Details

To implement these requirements, follow these steps:

1. **Create the Database:** ✓
 - Use MariaDB to create the database and tables from the relational schema.
 - Insert at least 5 rows for each table.
2. **Develop the Console Application:** ✓
 - Create a new C# Console Application.
 - Create two classes: one for the main code and another for database connection.
 - Implement a menu with the specified options.

Relational Schema :

- **Departments:** DepartmentID (PK), DeptName, Building
- **Students:** StudentID (PK), FirstName, LastName, DateOfBirth, Email
- **Courses:** CourseID (PK), CourseName, CourseDescription, CreditHours, DepartmentID (FK)
- **Professors:** ProfessorID (PK), FirstName, LastName, DepartmentID (FK), Email
- **Classrooms:** ClassroomID (PK), RoomName, Capacity, Location
- **Enrollments:** StudentID (PK, FK), CourseID (PK, FK), EnrollmentDate, Grade
- **TeachingAssignments:** ProfessorID (PK, FK), CourseID (PK, FK), Semester
- **ClassSchedules:** ScheduleID (PK), CourseID (FK), ClassroomID (FK), DayOfWeek, Time

Hiba Nejari

- **DepartmentsCourses** : DepartmentID, CourseID
- **Staff (abstract)**: StaffID (PK), FirstName, LastName, DepartmentID (FK)

AdministrativeStaff: StaffID (PK, FK), Position, OfficeNumber

MaintenanceStaff: StaffID (PK, FK), SkillSet

SecurityStaff: StaffID (PK, FK), Shift, ClearanceLevel