

Mini Project 01

Edit and Filter Students (SQL UPDATE and SELECT WHERE)

CSC-361 Database System Design

Dr. Qixin Deng

Due by: Feb. 22, 2026

Overview. In this mini project, you will extend the course demo database application by implementing two realistic and commonly used database features:

1. **Edit a student record** using **DML UPDATE**
2. **Filter the student list** using **SELECT ... WHERE** with optional conditions

Goal: practice the full workflow

HTML Form → FastAPI Route → Database Function → SQL.

You must finish this project Independently!

Part A: Edit Student

You must implement the ability to edit an existing student record.

Web UI Requirements

- Each student row in the table must include an **Edit** control.
- The Edit control may reveal an inline form (recommended) or link to a separate edit form.
- The form must clearly correspond to the correct student.

Server Requirements

- Implement a FastAPI route similar to:

`POST /students/{student_id}/edit`

- The route must receive form data and call a database update function:

`update_student(student_id, major, gpa, credits, city, ...)`

- Use a parameterized SQL UPDATE statement.
- The SQL must include a WHERE clause to update exactly **one** student.
- You must allow editing of **major, gpa, credits, and city**.

Part B: Filter Students

You must implement filtering of the student list using SQL `WHERE` conditions.

Web UI Requirements

- Add a small filter form above the student table.
- The form must use the HTTP `GET` method.
- All fields must be optional.
- You must support **at least two optional filters**:
 - One categorical or text-based filter (e.g., `major`)
 - One numeric filter (e.g., minimum GPA)

If a filter field is left blank, it must not restrict the result set. If all fields are blank, all students should be displayed.

Server Requirements

- Implement a function such as:

```
fetch_students_filtered(major=None, min_gpa=None)
```
- Use `SELECT ... FROM Student` with an optional `WHERE` clause.
- Combine multiple conditions using `AND`.

Scope control: Do **not** implement joins, subqueries, pagination, or advanced UI elements. Keep it simple and correct.

Submission Requirements

Submit a single `.zip` file containing the following items:

1. **Source Code** (your full project folder).
2. **Report (PDF)** explaining all modifications you made.
 - Provide sufficient explanation for each change to demonstrate your design idea.
 - Submitting code means you fully understand what you submitted.
3. **Short Demo Video** (screen recording).
 - Record the entire screen.
 - Show the project open in your IDE.
 - Manually run the project.
 - Show **two examples for Part A** and **two examples for Part B**.