

# Part 1: Database Creation

## Task 1

Create a database named **university\_db**

---

# Part 2: Table Creation

## Task 2: departments table

Create a table named **departments** with the following columns:

- dept\_id as integer (primary key)
- dept\_name as text (not null)

## Task 3: courses table

Create a table named **courses** with the following columns:

- course\_id as integer (primary key)
- course\_name as text (not null)
- credits as integer (not null, default 3)
- dept\_id as integer (foreign key)

The foreign key must reference **departments(dept\_id)** and use **on delete cascade**.

## Task 4: students table

Create a table named **students** with the following columns:

- student\_id as integer (primary key)
- student\_name as text (not null)

## Task 5: enrollments table (M:N)

Create a table named **enrollments** to link students and courses.

- enrollment\_id as integer (primary key)
- student\_id as integer (foreign key)

- course\_id as integer (foreign key)

Both foreign keys must reference their respective tables and use **on delete cascade**.

### Task 6: **health\_records** table (1:N)

Create a table named **health\_records** to store student health info:

- record\_id as integer (primary key)
- student\_id as integer (foreign key)
- blood\_group as text

The foreign key must reference **students(student\_id)** and use **on delete cascade**.

---

## Part 3: Data Insertion

### Task 7: Insert data

Insert the following data:

- 3 departments:
  - computer science
  - mathematics
  - physics
- 4 courses:
  - database systems — 4 credits — computer science
  - algorithms — 3 credits — computer science
  - calculus — 4 credits — mathematics
  - quantum mechanics — 5 credits — physics
- 5 students:
  - alice
  - bob
  - charlie
  - samy
  - eva
- Enroll students in multiple courses using the **enrollments** table:
  - alice — database systems, algorithms
  - bob — database systems
  - charlie — algorithms, calculus
  - samy — calculus, quantum mechanics
  - eva — no courses

- Add health records for some students using the **health\_records** table:
    - alice — A+
    - bob — B+
    - charlie — O-
    - samy — AB+
    - eva — A-
- 

## Part 4: Joins and Cascading Actions

### Task 8: inner join

1-Write a query to display each student's name along with their blood group.

2-Write a query to display each student's name along with their blood group courses they are enrolled in. Only include students who are enrolled in at least one course.

### Task 9: left join

Write a query to display all students, including those who are not enrolled in any course. Show student name and course name (course name can be NULL if not enrolled).

### Task 10: right join

Write a query to display all courses, including those that have no students enrolled. Show course name and student name (student name can be NULL if no student is enrolled).

### Task 11: cascading delete on departments

Delete one department and observe the effect on the related courses and enrollments. Explain what happens to students' enrollments.

### Task 12: cascading delete on students

Delete one student and observe the effect on the **enrollments** and **health\_records** tables.

---

## Part 5: Aggregate Functions

### Task 13:Aggregate Functions

Get the following:

- Total number of students
- Average credits of all courses
- Highest credit
- Lowest credit
- Total number of students who have A+ as blood group