

## Activity 01

- 1) Write the SQL command to create a schema named `university` if it does not already exist.
- 2) In the schema `university`, create a table `students` with the following columns:
  - `student_id` (integer, primary key)
  - `first_name` (varchar, 50 characters)
  - `last_name` (varchar, 50 characters)
  - `date_of_birth` (date)
  - `email` (varchar, 100 characters, unique)
- 3) Explain the difference between a schema and a database in SQL. Provide an example of when you would use each.
- 4) Write an SQL query to create a table `courses` in the `university` schema with the following attributes:
  - `course_id` (integer, primary key)
  - `course_name` (varchar, 100 characters)
  - `course_code` (varchar, 10 characters, unique)
  - `credits` (integer, range 1-6)
- 5) Write an SQL query to create a `course_enrollments` table with the following columns:
  - `enrollment_id` (integer, primary key)
  - `student_id` (integer, foreign key referencing `students.student_id`)
  - `course_id` (integer, foreign key referencing `courses.course_id`)
  - `enrollment_date` (date)
- 6) Write an SQL command to insert the following record into the `students` table:  
`student_id = 101, first_name = 'John', last_name = 'Doe', date_of_birth = '2000-05-15', email = 'john.doe@email.com'`
- 7) Insert a new record into the `courses` table with the following data:  
`course_id = 201, course_name = 'Database Systems', course_code = 'CS101', credits = 3`
- 8) Write an SQL command to insert a record into the `course_enrollments` table, assuming the student with `student_id = 101` and the course with `course_id = 201` are being enrolled on the date '2025-01-16'.
- 9) Write an SQL statement to update the `email` address of the student with `student_id = 101` to [new.email@example.com](mailto:new.email@example.com).
- 10) Write an SQL query to update the `credits` of the course with `course_id = 201` to 4 where the `course_name` is 'Database Systems'.

- 11) Write an SQL statement to delete the student record with `student_id = 101` from the `students` table. Comment on it.
- 12) Write an SQL command to delete the course with `course_id = 201` from the `courses` table. Comment on it.
- 13) Write an SQL statement to delete all records from the `course_enrollments` table where `student_id = 101` and `course_id = 201`.
- 14) Write an SQL statement to delete all records from the `students` table. **Note:** Ensure the referential integrity is maintained and that foreign key constraints are considered.
- 15) Write an SQL query to delete all records from the `courses` table that have no enrollments in the `course_enrollments` table.

## Activity 02

- 1) Write a SQL query to retrieve all columns and all rows from the `students` table.
- 2) Write a SQL query to retrieve the `first_name`, `last_name`, and `email` columns from the `students` table
- 3) Write a SQL query to retrieve all distinct `course_code` values from the `courses` table.
- 4) Write a SQL query to retrieve the `student_id`, `first_name`, `last_name`, and a calculated field `age` (where `age` is calculated as the difference between the current date and the `date_of_birth`) from the `students` table.
- 5) Write a SQL query to retrieve all students whose `student_id` is greater than 100 and less than or equal to 200.
- 6) Write a SQL query to retrieve all students whose `date_of_birth` is between '1995-01-01' and '2000-12-31'.
- 7) Write a SQL query to retrieve all students whose `student_id` is either 101, 102, or 103.
- 8) Write a SQL query to retrieve all students whose email starts with the letter 'j' (e.g., [john.doe@email.com](mailto:john.doe@email.com)).
- 9) Write a SQL query to retrieve all students who do not have an email address (i.e., where the `email` is NULL).
- 10) Write a SQL query to retrieve all students' `student_id`, `first_name`, and `last_name` from the `students` table, ordered by `last_name` in ascending order.
- 11) Write a SQL query to retrieve all `student_id`, `first_name`, `last_name`, and `date_of_birth` from the `students` table, ordered by `last_name` (ascending) and then by `first_name` (ascending).

### Activity 03

1. Write an SQL query to count the total number of students in the `students` table
2. Write an SQL query to calculate the total credits of all courses in the `courses` table
3. Write an SQL query to calculate the average credits of all courses
4. Write an SQL query to find the course with the smallest number of credits
5. Write an SQL query to find the course with the largest number of credits
6. Write an SQL query to count the number of students enrolled in each course
7. Write an SQL query to calculate the total credits for courses where students are enrolled:
- ~~8. Write an SQL query to find the average age of students grouped by course:~~
9. Write an SQL query to find the first and the last enrolled in each course:
10. Write an SQL query to find courses with more than 5 students enrolled: