

DATABASE MANAGEMENT SYSTEMS

Lab Task — DDL | Subqueries | Joins

DBMS Lab

Subject DBMS Lab	Total Marks 10	Time Allowed 2.5 Hours	Tool Oracle / MySQL
---------------------	-------------------	---------------------------	------------------------

Covered Topics: DDL Commands | DML | Subqueries & Nested Queries | Joins

Note: Use the following four tables for all tasks. Create tables in the order shown (respecting foreign key dependencies).

Database Schema — University Course Management

FACULTY

Column	Type	Constraint
faculty_id	INT	PRIMARY KEY
faculty_name	VARCHAR(80)	NOT NULL
email	VARCHAR(100)	UNIQUE, NOT NULL
department	VARCHAR(60)	—
hire_date	DATE	—

COURSE

Column	Type	Constraint
course_id	INT	PRIMARY KEY
course_name	VARCHAR(100)	UNIQUE, NOT NULL
credits	INT	CHECK (credits BETWEEN 1 AND 6)
faculty_id	INT	FOREIGN KEY → FACULTY
fee	DECIMAL(10,2)	—

STUDENT

Column	Type	Constraint
student_id	INT	PRIMARY KEY
student_name	VARCHAR(80)	NOT NULL
gpa	DECIMAL(3,2)	CHECK (gpa BETWEEN 0 AND 4)
city	VARCHAR(50)	—

Column	Type	Constraint
enroll_year	INT	—

ENROLLMENT

Column	Type	Constraint
enrollment_id	INT	PRIMARY KEY
student_id	INT	FOREIGN KEY → STUDENT
course_id	INT	FOREIGN KEY → COURSE
grade	VARCHAR(2)	—
semester	VARCHAR(20)	NOT NULL

Part A — DDL

Q1

Create the FACULTY and COURSE tables with all appropriate data types, primary key constraints, a UNIQUE constraint on email and course_name, and a foreign key from COURSE.faculty_id referencing FACULTY.faculty_id. Insert at least 5 records into each table.

Q2

Add a new column phone VARCHAR(15) to the FACULTY table. Then add a column classroom VARCHAR(20) to the COURSE table. Finally, drop the classroom column you just added using ALTER TABLE.

Q3

Create the STUDENT and ENROLLMENT tables. Include a CHECK constraint ensuring gpa BETWEEN 0 AND 4 and another ensuring credits BETWEEN 1 AND 6. Insert at least 6 students and 8 enrollment records ensure at least one student is enrolled in more than two courses, and at least one student has no enrollment.

Part B — Subqueries & Nested Queries

Q4

Write a query to display the names of all students whose GPA is greater than the average GPA of all students in the STUDENT table. Use a single-row subquery.

Q5

Find all students who are enrolled in the same course(s) as the student named 'Ayesha Noor'. Display student names only. Use a subquery with the IN operator.

Q6

Display each course's `course_name` and fee along with the maximum fee among all courses taught by the same faculty member. Use a correlated subquery in the `SELECT` clause.

Q7

List all faculties who teach at least one course with a fee greater than 15,000. Use the `EXISTS` keyword in your subquery.

Part C — Joins and Their Types

Q8

Using an INNER JOIN, display each student's name, the course they are enrolled in, and the semester. Only include students who have at least one enrollment record.

Q9

Using a LEFT OUTER JOIN, list all students along with the names of courses they are enrolled in. Include students who have no enrollment their course columns should show NULL.

Q10

Write a query using a RIGHT OUTER JOIN to display all courses and the names of students enrolled in them. Include courses that have no students enrolled yet.

Q11

Using a SELF JOIN on the FACULTY table, display each faculty member's name alongside the name of another faculty who belongs to the same department. Use aliases f1 and f2 and exclude rows where both aliases refer to the same person.

Q12

Using JOIN + GROUP BY + SUM(), display each faculty member's name and the total fee of all courses they teach. Sort the result from highest to lowest total fee. Only include faculties who teach at least one course.

Q13

Using JOIN + GROUP BY + HAVING, list all students who are enrolled in more than one course. Display the student's name and the total number of courses they are enrolled in.