# Practical-3

**Problem Statement** - SQL Queries – all types of Join, Sub-Query and View:Write at least10 SQL queries for suitable database application using SQL DML statements: all types of Join ,Sub-Query and View

# Code : (Joins)

CREATE TABLE students ( student\_id INT PRIMARY KEY, name VARCHAR(50),

age INT,

gender VARCHAR(10), department VARCHAR(50)

);

CREATE TABLE courses ( course\_id INT PRIMARY KEY, course\_name VARCHAR(100), department VARCHAR(50)

);

CREATE TABLE enrollments ( enrollment\_id INT PRIMARY KEY, student\_id INT,

course\_id INT,

FOREIGN KEY (student\_id) REFERENCES students(student\_id), FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

INSERT INTO students (student\_id, name, age, gender, department) VALUES

(101, 'John Doe', 20, 'Male', 'Computer Science'),

(102, 'Alice Smith', 21, 'Female', 'Information Technology'), (103, 'Bob Johnson', 22, 'Male', 'Mechanical Engineering');

INSERT INTO courses (course\_id, course\_name, department) VALUES

(201, 'Database Management Systems', 'Computer Science'), (202, 'Operating Systems', 'Computer Science'),

(203, 'Data Structures', 'Information Technology');

INSERT INTO enrollments (enrollment\_id, student\_id, course\_id) VALUES

(1, 101, 201),

(2, 102, 202),

(3, 103, 203),

(4, 101, 203);

SELECT s.name, c.course\_name FROM students s

INNER JOIN enrollments e ON s.student\_id = e.student\_id INNER JOIN courses c ON e.course\_id = c.course\_id;

SELECT s.name, c.course\_name FROM students s

LEFT JOIN enrollments e ON s.student\_id = e.student\_id LEFT JOIN courses c ON e.course\_id = c.course\_id;

SELECT c.course\_name, s.name FROM courses c

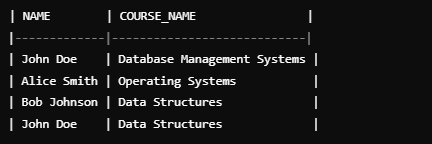
RIGHT JOIN enrollments e ON c.course\_id = e.course\_id RIGHT JOIN students s ON e.student\_id = s.student\_id;

SELECT s.name, c.course\_name FROM students s

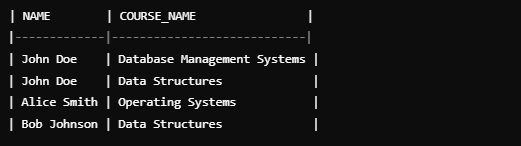
FULL OUTER JOIN enrollments e ON s.student\_id = e.student\_id FULL OUTER JOIN courses c ON e.course\_id = c.course\_id;

# Output

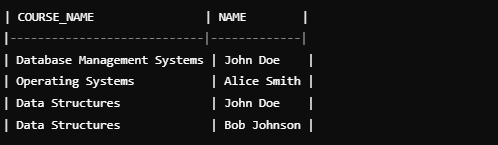
* 1. **INNER JOIN**



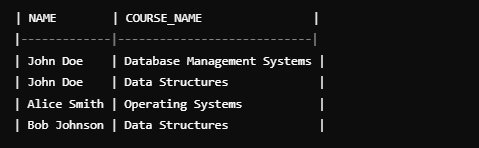
# LEFT OUTR JOIN



* 1. **RIGHT OUTER JOIN**



# FULL OUTER JOIN



* **Code : (Sub Query & View)**

SELECT s.name FROM students s

WHERE s.student\_id IN ( SELECT student\_id FROM enrollments GROUP BY student\_id

HAVING COUNT(course\_id) > 1

);

SELECT s.name FROM students s WHERE NOT EXISTS (

SELECT 1

FROM enrollments e

WHERE e.student\_id = s.student\_id

);

CREATE VIEW student\_courses AS

SELECT s.name AS student\_name, c.course\_name FROM students s

JOIN enrollments e ON s.student\_id = e.student\_id JOIN courses c ON e.course\_id = c.course\_id;

# Output :

**1. Sub Query 1 2. Sub Query 2**



# 3. View

