**Experiment-1**

**Student Name:** Hemant Narain Jha **UID:** 23BCS10022

**Branch:** CSE **Section/Group:** KRG-2B

**Semester:** 5th **Date of Performance:** 28-07-25

**Subject Name:** ADBMS **Subject Code:** 23CSP-333

# Aim:

* 1. Department-Course Subquery and Access Control
     + Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
     + Insert five departments and at least ten courses across those departments.
     + Use a subquery to count the number of courses under each department.
     + Filter and retrieve only those departments that offer more than two courses.
     + Grant SELECT-only access on the courses table to a specific user.

|  |  |
| --- | --- |
| dept\_id | dept\_name |
| 1 | Computer Science |
| 2 | Mechanical Engineering |
| 3 | Electrical Engineering |
| 4 | Mathematics |
| 5 | Physics |

| **course\_id** | **course\_name** | **dept\_id** |
| --- | --- | --- |
| 101 | Data Structures | 1 |
| 102 | Algorithms | 1 |
| 103 | Operating Systems | 1 |
| 104 | Fluid Mechanics | 2 |
| 105 | Thermodynamics | 2 |
| 106 | Mechanical Vibrations | 2 |
| 107 | Circuit Theory | 3 |
| 108 | Electromagnetic Fields | 3 |
| 109 | Linear Algebra | 4 |
| 110 | Calculus | 4 |

# Objective:

* To understand how to use JOINS in SQL.
* To understand the basic SQL Queries.
* To learn how to use Sub-Queries in SQL.

## DBMS Script:

-- Create Departments table

CREATE TABLE Departments (

dept\_id INT PRIMARY KEY, -- Primary key for dept\_id

dept\_name VARCHAR(100) NOT NULL -- Department name

);

-- Create Courses table with a foreign key to Departments

CREATE TABLE Courses (

course\_id INT PRIMARY KEY, -- Primary key for course\_id

course\_name VARCHAR(100) NOT NULL, -- Course name

dept\_id INT, -- Foreign key to the Departments table

FOREIGN KEY (dept\_id) REFERENCES Departments(dept\_id) -- Foreign key constraint

);

-- Insert data into Departments table

INSERT INTO Departments (dept\_id, dept\_name) VALUES

(1, 'Computer Science'),

(2, 'Mechanical Engineering'),

(3, 'Electrical Engineering'),

(4, 'Mathematics'),

(5, 'Physics');

-- Insert data into Courses table

INSERT INTO Courses (course\_id, course\_name, dept\_id) VALUES

(101, 'Data Structures', 1), -- Course in Computer Science

(102, 'Algorithms', 1), -- Course in Computer Science

(103, 'Operating Systems', 1), -- Course in Computer Science

(104, 'Fluid Mechanics', 2), -- Course in Mechanical Engineering

(105, 'Thermodynamics', 2), -- Course in Mechanical Engineering

(106, 'Mechanical Vibrations', 2), -- Course in Mechanical Engineering

(107, 'Circuit Theory', 3), -- Course in Electrical Engineering

(108, 'Electromagnetic Fields', 3), -- Course in Electrical Engineering

(109, 'Linear Algebra', 4), -- Course in Mathematics

(110, 'Calculus', 4); -- Course in Mathematics

-- Subquery to count the number of courses offered by each department

SELECT dept\_name

FROM Departments

WHERE dept\_id IN (

SELECT dept\_id

FROM Courses

GROUP BY dept\_id

HAVING COUNT(course\_id) > 2 -- Departments offering more than 2 courses

);

-- Grant SELECT-only access to 'readonly\_user' on the Courses table

GRANT SELECT ON Courses TO readonly\_user;

## OUTPUT:

## 

1. **Learning Outcomes:**

* You will be able to write basic SQL queries.
* You will learn to perform JOINS in SQL.
* You will understand how to implement Sub-Queries.