DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Experiment-1

Student Name: Himanshu Gupta **UID:** 23BCS10889
Branch: CSE **Section/Group:** KRG-2B
Semester: 5th **Date of Performance:** 28-07-25
Subject Name: ADBMS **Subject Code:** 23CSP-333

1. AIM:

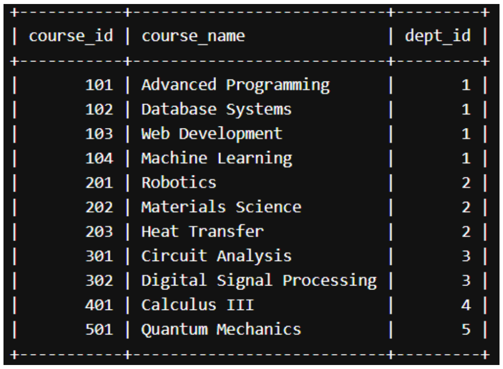
a.) Department-Course Subquery and Access Control

- 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.

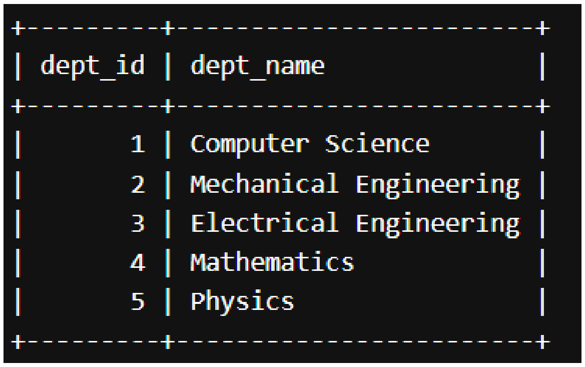
2. OBJECTIVE:

- To understand how to use JOINS in SQL.
- To understand the basic SQL Queries.
- To learn how to create foreign keys in SQL.

3. TABLES:



```
+-----------+--------------------------+---------+
| course_id | course_name              | dept_id |
+-----------+--------------------------+---------+
|       101 | Advanced Programming     |       1 |
|       102 | Database Systems         |       1 |
|       103 | Web Development          |       1 |
|       104 | Machine Learning         |       1 |
|       201 | Robotics                 |       2 |
|       202 | Materials Science        |       2 |
|       203 | Heat Transfer            |       2 |
|       301 | Circuit Analysis         |       3 |
|       302 | Digital Signal Processing |       3 |
|       401 | Calculus III             |       4 |
|       501 | Quantum Mechanics        |       5 |
+-----------+--------------------------+---------+
```



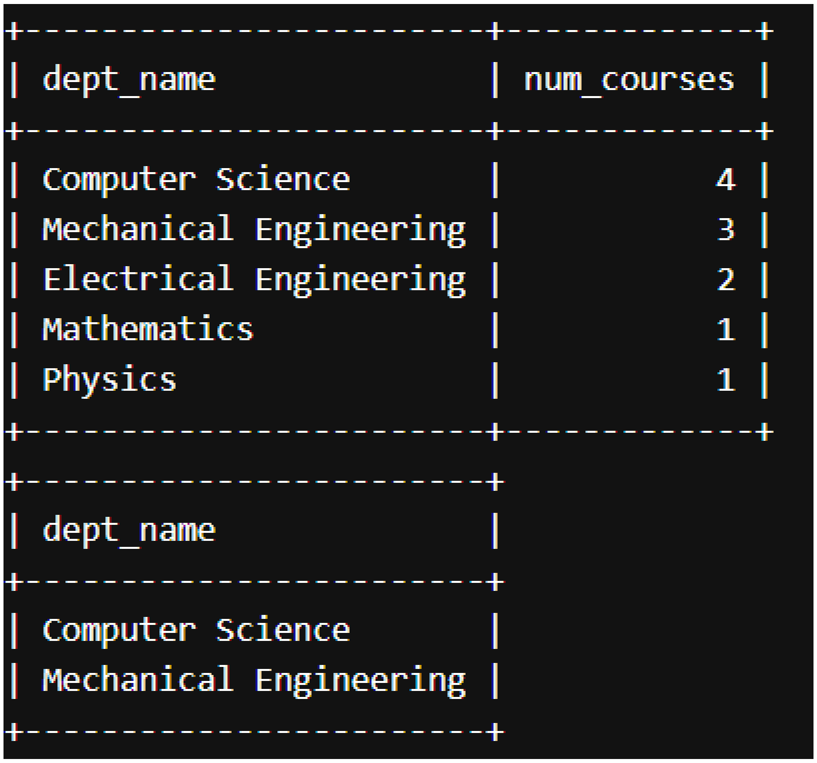
```
+---------+------------------------+
| dept_id | dept_name              |
+---------+------------------------+
|       1 | Computer Science       |
|       2 | Mechanical Engineering |
|       3 | Electrical Engineering |
|       4 | Mathematics            |
|       5 | Physics                |
+---------+------------------------+
```

4. DBMS:

```
-- Department table
CREATE TABLE Department (
 dept_id INT PRIMARY KEY,
 dept_name VARCHAR(100)
);

-- Course table with a foreign key to Department
CREATE TABLE Course (
 course_id INT PRIMARY KEY,
 course_name VARCHAR(100),
 dept_id INT,
 FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);

-- Insert data into Department
INSERT INTO Department (dept_id, dept_name) VALUES
(1, 'Computer Science'),
(2, 'Mechanical Engineering'),
(3, 'Electrical Engineering'),
(4, 'Mathematics'),
(5, 'Physics');

-- Insert data into Course
INSERT INTO Course (course_id, course_name, dept_id) VALUES
(101, 'Advanced Programming', 1),
(102, 'Database Systems', 1),
(103, 'Web Development', 1),
(104, 'Machine Learning', 1), -- CS now has 4 courses
(201, 'Robotics', 2),
(202, 'Materials Science', 2),
(203, 'Heat Transfer', 2), -- Mechanical now has 3 courses
(301, 'Circuit Analysis', 3),
(302, 'Digital Signal Processing', 3),
(401, 'Calculus III', 4),
(501, 'Quantum Mechanics', 5); -- Physics has 1 course


Select * from Course;
Select * from Department;


SELECT
    d.dept_name,
    (SELECT COUNT(c.course_id) FROM Course c WHERE c.dept_id = d.dept_id) AS num_courses
FROM
    Department d;
```

```
SELECT dept_name
FROM Department
WHERE dept_id IN (
 SELECT dept_id
 FROM Course
 GROUP BY dept_id
 HAVING COUNT(course_id) > 2
);

-- Grant SELECT-only access on the courses table to a specific user

GRANT SELECT ON Course TO readonly_user;
```

5. OUTPUT:



```
+------------------------+-------------+
| dept_name              | num_courses |
+------------------------+-------------+
| Computer Science       |           4 |
| Mechanical Engineering |           3 |
| Electrical Engineering |           2 |
| Mathematics            |           1 |
| Physics                |           1 |
+------------------------+-------------+
+------------------------+
| dept_name              |
+------------------------+
| Computer Science       |
| Mechanical Engineering |
+------------------------+
```

6. LEARNING OUTCOMES:

- Basic SQL queries.
- JOINS in SQL.
- Implementation of Sub-Queries.