EXPERIMENT-02

• <u>AIM:-</u> To design a normalized academic schema (up to 3NF) for managing departments and their courses, populate it with meaningful sample data, query departments offering more than two courses using a subquery, and implement access control using Data Control Language (DCL).

• THEORY:-

- ✓ **Normalization (3NF)**: The Third Normal Form eliminates transitive dependencies. A relation is in 3NF if it is in 2NF and no transitive functional dependency exists between non-prime attributes.
- **✓** Relational Model Design:
- ✓ **Departments** table holds unique department data.
- ✓ **Courses** table associates each course with exactly one department via a foreign key.
- ✓ Subqueries:

A subquery is a query nested inside another SQL query. It helps in filtering, transforming, or summarizing data based on related conditions.

✓ Access Control using DCL:

Data Control Language statements like GRANT manage user privileges. Granting SELECT access ensures a user can view but not modify data.

• <u>CODE:-</u>

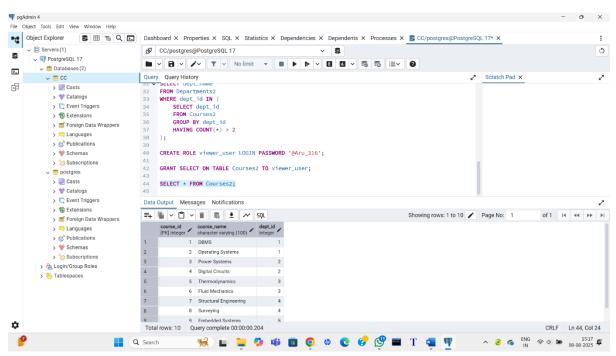
```
-- Drop if exists for clean re-execution
DROP TABLE IF EXISTS Courses;
DROP TABLE IF EXISTS Departments;
```

```
-- Create Departments table
CREATE TABLE Departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50) UNIQUE NOT NULL
);
-- Create Courses table
CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100) NOT NULL,
    dept_id INT NOT NULL,
```

```
FOREIGN KEY (dept id) REFERENCES Departments(dept id) ON
     DELETE CASCADE
     );
   > INSERTION OF DATA:
     -- Insert Departments
     INSERT INTO Departments (dept id, dept name) VALUES
     (1, 'Computer Science'),
     (2, 'Electrical'),
     (3, 'Mechanical'),
     (4, 'Civil'),
     (5, 'Electronics');
     -- Insert Courses
     INSERT INTO Courses (course id, course name, dept id) VALUES
     (101, 'DBMS', 1),
     (102, 'Operating Systems', 1),
     (103, 'Power Systems', 2),
     (104, 'Digital Circuits', 2),
     (105, 'Thermodynamics', 3),
     (106, 'Fluid Mechanics', 3),
     (107, 'Structural Engineering', 4),
     (108, 'Surveying', 4),
     (109, 'Embedded Systems', 5),
     (110, 'VLSI Design', 5);
     -- Insert Courses if more than 2 courses
SELECT dept name
FROM Departments
WHERE dept id IN (
  SELECT dept id
  FROM Courses
  GROUP BY dept id
  HAVING COUNT(*) > 2
     -- Grant Access to the user
GRANT SELECT ON TABLE Courses TO viewer user;
```

);

• OUTPUTS:-



• LEARNING OUTCOMES:-

- ➤ Understand and apply **3NF normalization** in database design.
- ➤ Use **foreign key constraints** to maintain referential integrity.
- ➤ Write **subqueries** using GROUP BY and HAVING to analyze relationships.
- ➤ Implement access control using GRANT statements in PostgreSQL.
- ➤ Handle **real-world schema modeling** and data organization tasks efficiently.