NAME-ROUSHAN KUMAR, UID-23BCC70013, SUB-ADBMS

EXPERIMENT-02

- **AIM:- Part A**

  Title: Create Department and Course Tables with Normalization (up to 3NF)

- **CODE:-**

```
-- 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
        );
```

- AIM :- **Part B :-**
  Title: Insert Sample Data into Department and Course Tables

- **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);
  ```

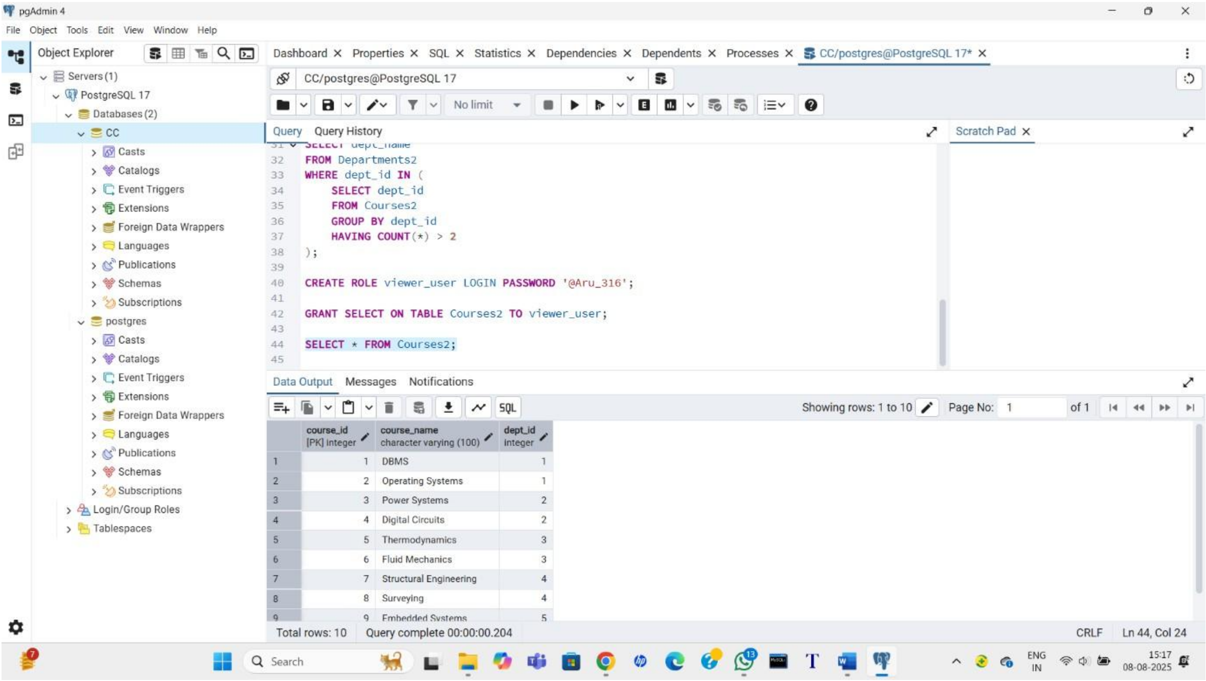
  -- Insert Courses if more than 2 courses

```
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:-**



```sql
SELECT dept_name
FROM Departments2
WHERE dept_id IN (
    SELECT dept_id
    FROM Courses2
    GROUP BY dept_id
    HAVING COUNT(*) > 2
);

CREATE ROLE viewer_user LOGIN PASSWORD '@Aru_316';

GRANT SELECT ON TABLE Courses2 TO viewer_user;

SELECT * FROM Courses2;
```

	course_id [PK] integer	course_name character varying (100)	dept_id integer
1	1	DBMS	1
2	2	Operating Systems	1
3	3	Power Systems	2
4	4	Digital Circuits	2
5	5	Thermodynamics	3
6	6	Fluid Mechanics	3
7	7	Structural Engineering	4
8	8	Surveying	4
9	9	Embedded Systems	5

Total rows: 10   Query complete 00:00:00.204

- **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.