Experiment 2

AIM: To Demonstrate Department-Course Subquery and Access Control

Theory: Subqueries in SQL are queries nested inside another query to filter, count, or manipulate data dynamically. They allow efficient data retrieval across related tables. PostgreSQL access control is managed using roles and privileges. The GRANT and REVOKE commands define what actions different users can perform on tables, ensuring database security and proper authorization.

CODE (with output):

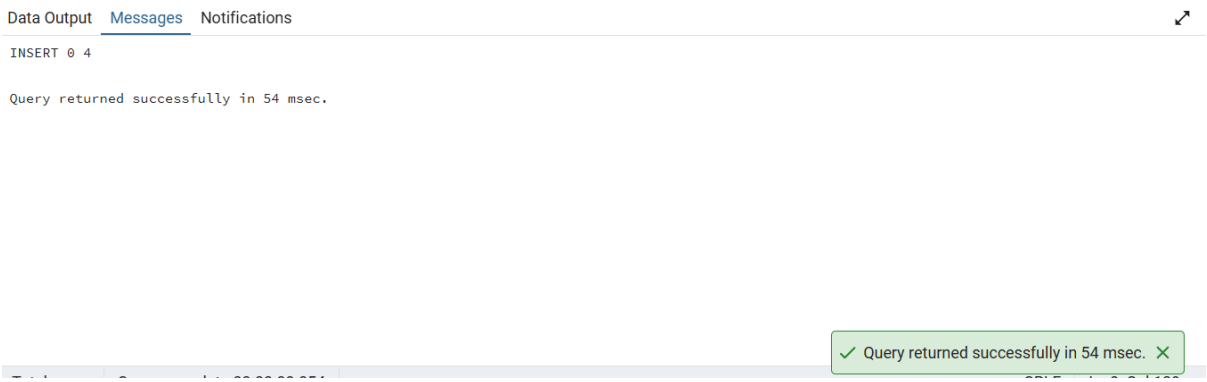
1. **CREATING THE TBALES:**

```
CREATE TABLE Department (dept_id SERIAL PRIMARY KEY, dept_name VARCHAR(100) NOT NULL);


CREATE TABLE Course (course_id SERIAL PRIMARY KEY, course_name VARCHAR(100) NOT NULL, dept_id INT REFERENCES Department(dept_id));


INSERT INTO Department (dept_name) VALUES ('Computer Science'), ('Electronics'), ('Mechanical');


INSERT INTO Course (course_name, dept_id) VALUES('DBMS', 1),('Computer Networks', 1),('Digital Logic', 2),('Thermodynamics', 3);
```

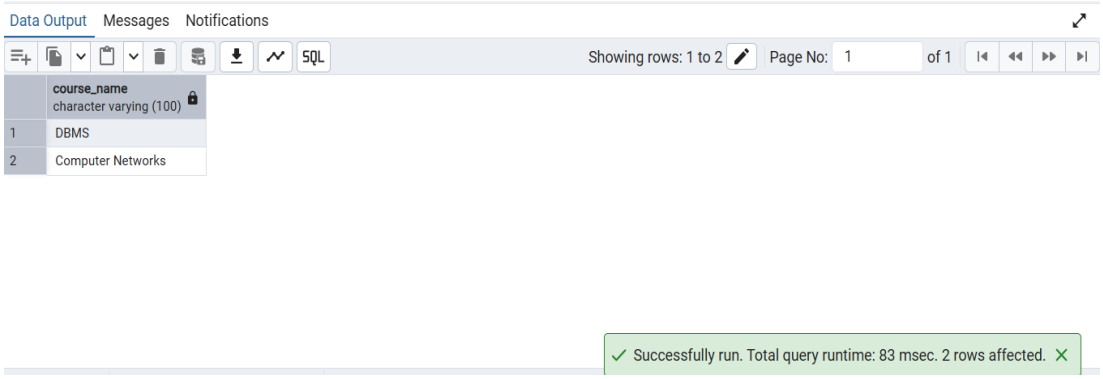


2. **Subquery examples:**

2.1 Finding all courses offered by Computer Science Department

```
SELECT course_name FROM Course WHERE dept_id = ( SELECT
dept_id FROM Department WHERE dept_name = 'Computer Science');
```

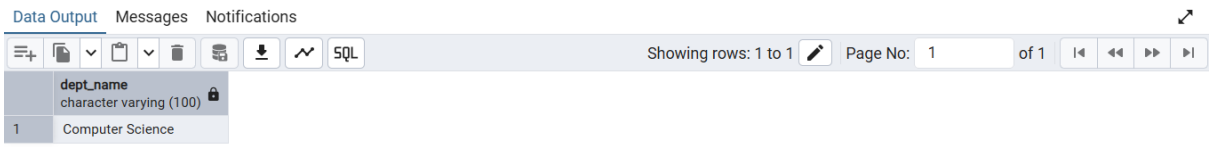
OUTPUT:



2.2 Find departments that offer more than 1 course

```
SELECT dept_name FROM Department d WHERE (SELECT COUNT(*)
FROM Course c WHERE c.dept_id = d.dept_id) > 1;
```

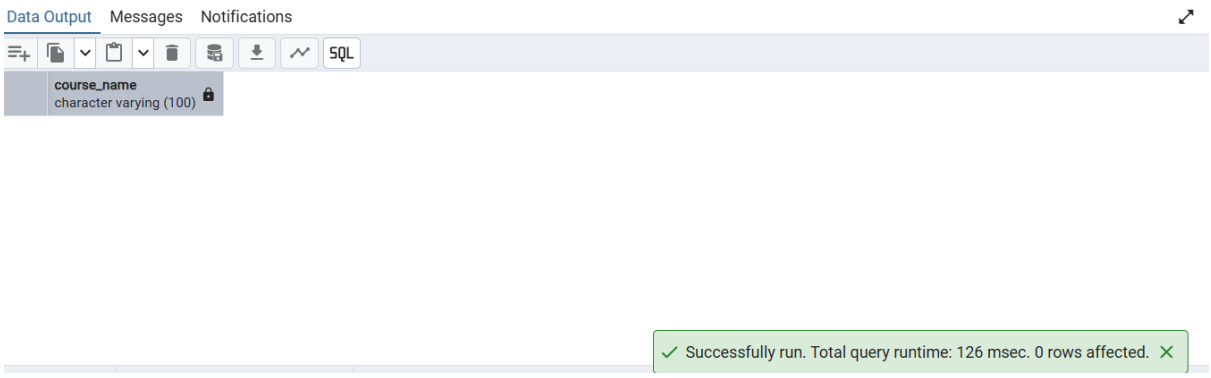
OUTPUT:



2.3 Find courses not assigned to any department

```
SELECT course_name FROM Course WHERE dept_id NOT IN (SELECT
dept_id FROM Department);
```

OUTPUT:

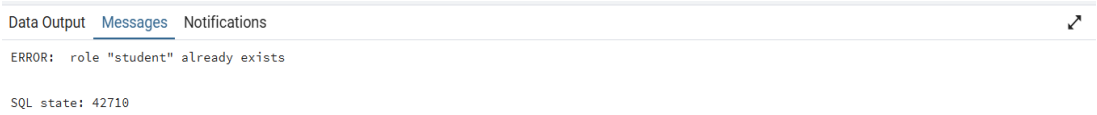


3. **ACCESS CONTROL:**

3.1 Create Roles

```
CREATE ROLE student LOGIN PASSWORD 'stud123';
CREATE ROLE professor LOGIN PASSWORD 'prof123';
```

OUTPUT:

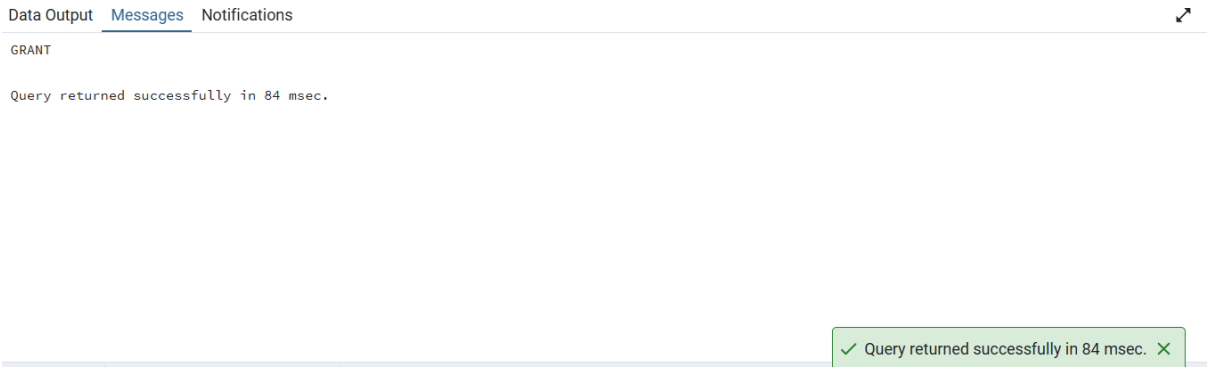


3.2 Grant privileges

GRANT SELECT ON Course TO student; -- students can only read courses

GRANT INSERT, UPDATE ON Course TO professor; -- professors can add/update courses

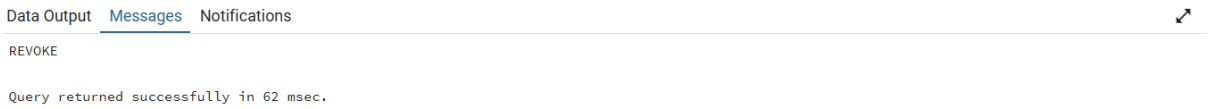
OUTPUT:



3.3 Revoke a privilege

REVOKE UPDATE on Course FROM student;

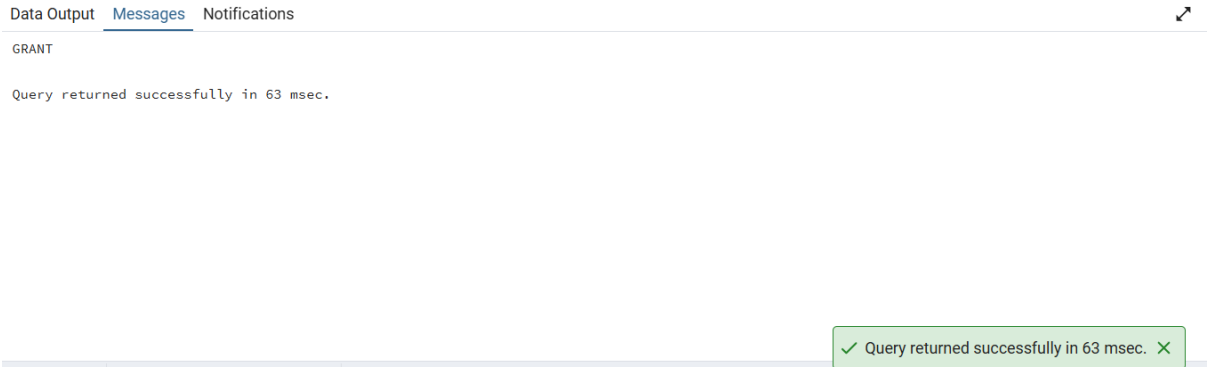
OUTPUT:



3.4 Grant schema usage

GRANT USAGE on SCHEMA public TO Student;

OUTPUT:



4.OPTIONALS(only to be used I case the above commands are not running because of pre existence):

DROP Table Department;

DROP Table Course;

Learning Outcomes

- Understood how **subqueries** can be applied to filter and analyze relational data.
- Learned how to **create roles** and manage **user privileges** using GRANT and REVOKE in PostgreSQL.
- Gained practical knowledge of combining **data retrieval** with **database security** in a single workflow.