

Department–Course Subquery and Access Control

Aim

To create normalized Department and Course tables, insert sample data, retrieve departments offering more than two courses using subquery, and manage access control using DCL commands.

Part A — Create Department and Course Tables with Normalization (up to 3NF)

Theory

- 1NF: No repeating groups, atomic values.
- 2NF: Every non-key attribute fully depends on the primary key.
- 3NF: No transitive dependency — non-key attributes depend only on the key.

In this design:

- Department details are stored in Departments table.
- Course details are stored in Courses table, with a foreign key linking to Departments.

SQL Code

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

Expected Output

```
1  CREATE TABLE Departments (  
2      dept_id INT PRIMARY KEY,  
3      dept_name VARCHAR(50) UNIQUE NOT NULL  
4  );  
5  
6  CREATE TABLE Courses (  
7      course_id INT PRIMARY KEY,  
8      course_name VARCHAR(100) NOT NULL,  
9      dept_id INT,  
10     FOREIGN KEY (dept_id) REFERENCES Departments(dept_id)  
11  );  
12
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 89 msec.

Part B — Insert Sample Data into Department and Course Tables

Theory

We insert sample data ensuring:

- At least 5 departments
- At least 10 courses
- Foreign keys are valid

SQL Code

```
INSERT INTO Departments (dept_id, dept_name) VALUES
```

```
(1, 'Computer Science'),  
(2, 'Electrical'),  
(3, 'Mechanical'),  
(4, 'Civil'),  
(5, 'Electronics');
```

```
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);
```

Expected Output

```
13  ▾ INSERT INTO Departments (dept_id, dept_name) VALUES  
14  (1, 'Computer Science'),  
15  (2, 'Electrical'),  
16  (3, 'Mechanical'),  
17  (4, 'Civil'),  
18  (5, 'Electronics');  
19  
Data Output  Messages  Notifications  
INSERT 0 5  
Query returned successfully in 75 msec.
```

```
20  ▾ INSERT INTO Courses (course_id, course_name, dept_id) VALUES  
21  (101, 'DBMS', 1),  
22  (102, 'Operating Systems', 1),  
23  (103, 'Power Systems', 2),  
24  (104, 'Digital Circuits', 2),  
25  (105, 'Thermodynamics', 3),  
26  (106, 'Fluid Mechanics', 3),  
27  (107, 'Structural Engineering', 4),  
28  (108, 'Surveying', 4),  
29  (109, 'Embedded Systems', 5),  
30  (110, 'VLSI Design', 5);  
31  
Data Output  Messages  Notifications  
INSERT 0 10  
Query returned successfully in 79 msec.
```

Part C — Retrieve Departments Offering More Than Two Courses Using Subquery

Theory

A subquery in the WHERE clause is used to find department IDs having more than two courses. The GROUP BY and HAVING clauses filter results.

SQL Code

```
SELECT dept_name
FROM Departments
WHERE dept_id IN (
    SELECT dept_id
    FROM Courses
    GROUP BY dept_id
    HAVING COUNT(course_id) > 1
);
```

Sample Output

```
1  CREATE TABLE Departments (
2      dept_id INT PRIMARY KEY,
3      dept_name VARCHAR(50) UNIQUE NOT NULL
4  );
5
6  CREATE TABLE Courses (
7      course_id INT PRIMARY KEY,
8      course_name VARCHAR(100) NOT NULL,
9      dept_id INT,
10     FOREIGN KEY (dept_id) REFERENCES Departments(dept_id)
11 );|
12
```

Data Output [Messages](#) Notifications

CREATE TABLE

Query returned successfully in 89 msec.

Part D — Grant SELECT Access on Courses Table Using DCL

Theory

DCL (Data Control Language) manages access rights in the database. GRANT is used to give permissions, REVOKE to take them back. Here, we grant only SELECT permission on the Courses table to viewer_user.

SQL Code

```
GRANT SELECT ON Courses TO viewer_user;
```

Expected Output

```
40  Create USER Khushi_23BD010018;
41  GRANT SELECT ON Courses TO Khushi_23BD010018;
42
```

Data Output [Messages](#) Notifications

GRANT

Query returned successfully in 133 msec.

```
42 REVOKE SELECT ON Courses FROM Khushi_23BD010018;
```

Data Output Messages Notifications

REVOKE

Query returned successfully in 67 msec.

Conclusion

In this practical, we:

- Applied normalization (up to 3NF) for Department and Course tables.
- Inserted meaningful sample data.
- Retrieved filtered results using subquery with aggregation.
- Managed access control using DCL commands.