Experiment:1.2

Student Name: Arijit Patra   **UID:** 23BCS11450

Branch: Computer Science & Engineering   **Section/Group:** KG-3B

Semester: 5th   **Subject Code:** 23CSP-339
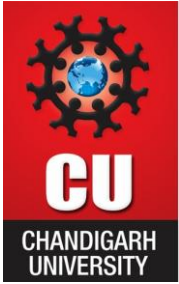
Subject Name: ADBMS

1. **Aim of the practical:**

   Author-Book Relationship Using Joins and Basic SQL Operations

   1. Design two tables — one for storing author details and the other for book details.

   2. Ensure a foreign key relationship from the book to its respective author.

   3. Insert at least three records in each table.

   4. Perform an INNER JOIN to link each book with its author using the common author ID.

   5. Select the book title, author name, and author's country.

2. Sample Output Description: The result shows the names of departments which are associated with more than two courses in the system.

3. Tool Used: SQL Server Management Studio

4. Code:

```sql
--/Problem Title: 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.

-- Create table for Faculties
CREATE TABLE Faculties (
    fac_id INT PRIMARY KEY,
    fac_name VARCHAR(100) NOT NULL
);

-- Create table for Subjects
CREATE TABLE Subjects (
    sub_id INT PRIMARY KEY,
    sub_name VARCHAR(150) NOT NULL,
    fac_id INT,
    FOREIGN KEY (fac_id) REFERENCES Faculties(fac_id)
);

-- Insert data into Faculties
INSERT INTO Faculties (fac_id, fac_name) VALUES
(10, 'Engineering'),
(20, 'Arts'),
(30, 'Sciences'),
(40, 'Commerce'),
(50, 'Law');

-- Insert data into Subjects
INSERT INTO Subjects (sub_id, sub_name, fac_id) VALUES
(201, 'Thermodynamics', 10),
(202, 'Fluid Mechanics', 10),
(203, 'Material Science', 10),
(204, 'World History', 20),
(205, 'Philosophy', 20),
(206, 'Botany', 30),
(207, 'Zoology', 30),
(208, 'Corporate Accounting', 40),
(209, 'Criminal Law', 50),
(210, 'Civil Law', 50);

-- Query to get faculties having more than 2 subjects
SELECT fac_name
FROM Faculties
WHERE fac_id IN (
    SELECT fac_id
```

```sql
    FROM Subjects
    GROUP BY fac_id
    HAVING COUNT(sub_id) > 2
);
```