Experiment 1

Student Name:Gourav Sharma   **UID: 23BCS10857**
Branch: CSE   **Section/Group: KRG 3-A**
Semester: 5th   **Date of Performance:24/07/2025**
Subject Name: ADBMS   **Subject Code: 23CSP-333**

1. **Aim**: University Database System helps in managing student enrollments, course allocations, and professor assignments effectively. The system also demonstrates secure access control and transaction safety. This includes CRUD operations, JOIN queries, and database-level user permission management.
   a. Author-Book Relationship Using Joins and Basic SQL Operations
   b. Department-Course Subquery and Access Control

2. **Objective:**

   The objective of the given SQL code is to demonstrate how to model and query relational data in a university-like database system using foundational and intermediate SQL concepts. It covers the creation of tables for authors, books, departments, and courses, and showcases how to retrieve combined data using JOINs, as well as how to apply subqueries for conditional filtering based on grouped data. The code emphasizes practical understanding of entity relationships and efficient data retrieval strategies.

3. **DBMS script and output:**

   Solution-(a)
   ```
   --Easy-Level Problem
   CREATE TABLE TBL_AUTHOR (
      AuthorID INT PRIMARY KEY,
      AuthorName VARCHAR(50),
      Country VARCHAR(50)
   );

   CREATE TABLE TBL_BOOK (
      BookID INT PRIMARY KEY,
      Title VARCHAR(100),
      AuthorID INT,
      FOREIGN KEY (AuthorID) REFERENCES TBL_AUTHOR(AuthorID)
   );

   INSERT INTO TBL_AUTHOR (AuthorID, AuthorName, Country) VALUES
   (1, 'Gourav', 'India'),
   (2, 'Bunty', 'Nepal'),
   (3, 'Himanshu', 'Bhutan'),
   (4, 'Bharat', 'India'),
   (5, 'Mehak', 'Nigeria');
   ```

```sql
INSERT INTO TBL_BOOK (BookID, Title, AuthorID) VALUES
(101, 'Mastering DBMS', 1),
(102, 'DAA for Beginners', 2),
(103, 'Competitive Coding Secrets', 3),
(104, 'Algorithm Design Made Easy', 4),
(105, 'Problem Solving in C++', 5);

SELECT B.Title, A.AuthorName, A.Country
FROM TBL_BOOK B
INNER JOIN TBL_AUTHOR A ON B.AuthorID = A.AuthorID;
```

Title	AuthorName	Country
Mastering DBMS	Gourav	India
DAA for Beginners	Bunty	Nepal
Competitive Coding Secrets	Himanshu	Bhutan
Algorithm Design Made Easy	Bharat	India
Problem Solving in C++	Mehak	Nigeria

Solution-(b)

```sql
--Medium-Level Problem

CREATE TABLE TBL_DEPARTMENT (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(50)
);

CREATE TABLE TBL_COURSE (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(100),
    DeptID INT,
    FOREIGN KEY (DeptID) REFERENCES TBL_DEPARTMENT(DeptID)
);

INSERT INTO TBL_DEPARTMENT (DeptID, DeptName) VALUES
(1, 'Computer Science'),
(2, 'Information Technology'),
(3, 'Electronics'),
(4, 'Mechanical'),
(5, 'Civil');

INSERT INTO TBL_COURSE (CourseID, CourseName, DeptID) VALUES
(101, 'DBMS', 1),
```

```
(102, 'DAA', 1),
(103, 'Competitive Coding', 1),
(104, 'Web Development', 2),
(105, 'Cyber Security', 2),
(106, 'Data Structures', 2),
(107, 'Digital Circuits', 3),
(108, 'Microprocessors', 3),
(109, 'Thermodynamics', 4),
(110, 'Structural Analysis', 5);

SELECT DeptName
FROM TBL_DEPARTMENT
WHERE DeptID IN (
    SELECT DeptID
    FROM TBL_COURSE
    GROUP BY DeptID
    HAVING COUNT(CourseID) > 2
);
```

DeptName
Computer Science
Information Technology

4. Learning Outcomes (What I have Learnt):

- Understand the creation of relational tables using **primary and foreign key constraints**.
- Gain hands-on experience with **INNER JOIN** to retrieve related data across tables.
- Learn to use **subqueries with GROUP BY and HAVING** clauses for data aggregation and filtering.
- Develop the ability to represent **one-to-many relationships** between entities (e.g., Authors → Books, Departments → Courses).
- Enhance skills in performing **basic and intermediate-level SQL queries** involving selection, joins, and subqueries.