Experiment 1

Student Name: ARPIT ANAND
UID: 23BCS12710
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:

- To create and manage relational databases LibraryDB and UniversityDB using SQL.
- To define tables with appropriate primary and foreign key constraints.
- To insert sample data into author, book, department, and course tables.
- To retrieve related data using **INNER JOIN** and **subqueries** with GROUP BY and HAVING.
- To manage user access by granting **SELECT privileges** on specific tables.

3. DBMS script and output:

Solution-(a)

```
CREATE DATABASE LibraryDB;
USE LibraryDB;

CREATE TABLE TBL_Author (
    author_id INT PRIMARY KEY,
    author_name VARCHAR(100),
    country VARCHAR(50)
);

CREATE TABLE TBL_Book (
```

```
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    author_id INT,
    FOREIGN KEY (author_id) REFERENCES Author(author_id)
);

INSERT INTO TBL_Author (author_id, author_name, country) VALUES
(1, 'J.K. Rowling', 'United Kingdom'),
(2, 'George R.R. Martin', 'United States'),
(3, 'Haruki Murakami', 'Japan');

INSERT INTO TBL_Book (book_id, title, author_id) VALUES
(101, 'Harry Potter and the Sorcerer''s Stone', 1),
(102, 'A Game of Thrones', 2),
(103, 'Kafka on the Shore', 3);

SELECT
    B.title AS Book_Title,
    A.author_name AS Author_Name,
    A.country AS Author_Country
FROM
    TBL_Book B
INNER JOIN
    TBL_Author A ON B.author_id = A.author_id:
```

BOOK_TITLE	AUTHOR_NAME	AUTHOR_COUNTRY
Harry Potter and the Sorcerer's Stone	J.K. Rowling	United Kingdom
A Game of Thrones	George R.R. Martin	United States
Kafka on the Shore	Haruki Murakami	Japan

3 rows returned in 0.01 seconds Download

Solution-(b)

```
CREATE DATABASE UniversityDB;
USE UniversityDB;

CREATE TABLE TBL_Department (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100)
);

CREATE TABLE TBL_Course (
    course_id  INT  PRIMARY  KEY,
    course_name  VARCHAR(100),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES TBL_Department(dept_id)
);

INSERT INTO TBL_Department (dept_id, dept_name) VALUES
(1, 'Computer Science'),
(2, 'Mechanical Engineering'),
(3, 'Electrical Engineering'),
(4, 'Civil Engineering'),
(5, 'Mathematics');
INSERT INTO TBL_Course (course_id, course_name, dept_id) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'DBMS', 1),
(104, 'Thermodynamics', 2),
(105, 'Fluid Mechanics', 2),
(106, 'Circuit Theory', 3),
```

```
(107, 'Power Systems', 3),
(108, 'Structural Analysis', 4),
(109, 'Linear Algebra', 5),
(110, 'Calculus', 5);

CREATE USER 'user123'@'localhost' IDENTIFIED BY 'password123';
GRANT SELECT ON UniversityDB.TBL_Course TO 'user123'@'localhost';

SELECT dept_name
FROM TBL_Department
WHERE dept_id IN (
    SELECT dept_id
    FROM TBL_Course
    GROUP BY dept_id
    HAVING COUNT(course_id) > 2
);
```

DEPT_NAME
Computer Science

1 rows returned in 0.01 seconds Download

4. Learning Outcomes:

- Understand how to design relational databases using **primary and foreign key** constraints.
- Gain hands-on experience with **SQL DDL and DML** commands for creating and manipulating tables.
- Learn to use **INNER JOINs** to combine data from related tables.
- Apply **subqueries** with aggregation (GROUP BY, HAVING) to filter complex data sets.
- Learn how to **grant user privileges** using the GRANT statement for controlled access.