Experiment 1

Student Name: Deepanshu Mandhyan **UID: 23BCS12327**
Branch: CSE **Section/Group: KRG 3-A**
Semester: 5th **Date of Performance:24/07/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Aim:

The University DBMS manages enrollments, courses, and faculty with secure access and transactions using CRUD, JOINs, subqueries, and permissions.

a. Author-Book: Uses JOINs for linking authors and books.

b. Department-Course: Uses subqueries and access control for course mapping.

2. Objective:

- Build **LibraryDB** and **UniversityDB** in SQL.
- Set up tables with **primary** and **foreign keys**.
- Insert data into **author**, **book**, **department**, and **course** tables.
- Use **JOINs**, **subqueries**, **GROUP BY**, and **HAVING** to fetch data.
- Grant **SELECT** access to specific users.

3. DBMS script and output:

Solution-(a)

```
CREATE DATABASE DigitalLibrary;


USE DigitalLibrary;


CREATE TABLE WriterMaster (
    WriterID INT PRIMARY KEY,
    WriterName VARCHAR(100),
```

```
    Nationality VARCHAR(50)
);

CREATE TABLE PublicationMaster (
    PubID INT PRIMARY KEY,
    PubTitle VARCHAR(120),
    WriterID INT,
    FOREIGN KEY (WriterID) REFERENCES WriterMaster(WriterID)
);

INSERT INTO WriterMaster VALUES
(1, 'Neil Gaiman', 'UK'),
(2, 'Margaret Atwood', 'Canada'),
(3, 'Paulo Coelho', 'Brazil');

INSERT INTO PublicationMaster VALUES
(201, 'American Gods', 1),
(202, 'The Handmaid''s Tale', 2),
(203, 'The Alchemist', 3),
(204, 'Coraline', 1);

SELECT
    P.PubTitle AS Publication_Title,
    W.WriterName AS Author,
    W.Nationality
FROM
    PublicationMaster P
JOIN
```

WriterMaster W ON P.WriterID = W.WriterID;

	Publication_Title	Author	Nationality
1	American Gods	Neil Gaiman	UK
2	The Handmaid's Tale	Margaret Atwood	Canada
3	The Alchemist	Paulo Coelho	Brazil
4	Coraline	Neil Gaiman	UK

Figure(a): JOIN Operation

Solution-(b)

```
CREATE DATABASE AcademyDB;

USE AcademyDB;

CREATE TABLE TBL_Division (
    division_id INT PRIMARY KEY,
   division_name VARCHAR(100)
);

CREATE TABLE TBL_Subject (
subject_id INT PRIMARY KEY,
subject_name VARCHAR(100),
division_id INT,
FOREIGN KEY (division_id) REFERENCES TBL_Division(division_id)
);

INSERT INTO TBL_Division (division_id, division_name) VALUES
(1, 'Information Technology'),
(2, 'Civil Engineering'),
(3, 'Electronics'),
(4, 'Physics'),
(5, 'Statistics');

INSERT INTO TBL_Subject (subject_id, subject_name, division_id) VALUES
(201, 'Programming Basics', 1),
(202, 'Database Systems', 1),
(203, 'Operating Systems', 1),
(204, 'Building Design', 2),
(205, 'Construction Materials', 2),
```

```
(206, 'Signals and Systems', 3),
(207, 'Microprocessors', 3),
(208, 'Quantum Mechanics', 4),
(209, 'Probability Theory', 5),
(210, 'Statistical Methods', 5);

CREATE LOGIN academy_user WITH PASSWORD = 'Secure@123';
CREATE USER academy_user FOR LOGIN academy_user;

GRANT SELECT ON TBL_Subject TO academy_user;

SELECT division_name
FROM TBL_Division
WHERE division_id IN (
   SELECT division_id
   FROM TBL_Subject
   GROUP BY division_id
   HAVING COUNT(subject_id) > 2
);
```

	division_name
1	Information Technology

Figure (b): SQL Operation

4. Learning Outcomes:

- Design databases with primary and foreign keys.
- Use SQL DDL/DML to manage tables.
- Join related tables using INNER JOIN.
- Filter data with subqueries, GROUP BY, HAVING.
- Grant access using GRANT command.