DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-1

Student Name: Ayush Tiwari
UID: 23BCS11366
Branch: BE-CSE
Section/Group: 23BCS_KRG-3_B
Semester: V
Date of Performance: 21/07/25
Subject Name: ADBMS
Subject Code: 23CSH-333

1. Aim:
   To create and demonstrate **relational database concepts** using two tables — WRITERS and PUBLICATIONS — and to **fetch related data** using SQL JOIN

2. Objective:

   WRITERS table – This table will store the information of different writers, such as:

   Writer ID

   Writer Name

   Nationality

   PUBLICATIONS table – This table will store the information of books or publications. Each book is written by a writer, so this table includes:

   Publication ID

   Book Title

   A reference to the writer who wrote the book

3. Code:

```
---- easy experiment -----
CREATE TABLE TBL_AUTHOR
(
    AUTHOR_ID INT PRIMARY KEY,
```

```
    AUTHOR_NAME VARCHAR(MAX),
    COUNTRY VARCHAR(MAX)
)

CREATE TABLE TBL_BOOKS
(
    BOOK_ID INT PRIMARY KEY,
    BOOK_TITLE VARCHAR(MAX),
    AUTHORID INT
    FOREIGN KEY (AUTHORID) REFERENCES TBL_AUTHOR(AUTHOR_ID)
)

INSERT INTO TBL_AUTHOR (AUTHOR_ID, AUTHOR_NAME, COUNTRY) VALUES
(1, 'George Orwell', 'United Kingdom'),
(2, 'Haruki Murakami', 'Japan'),
(3, 'J.K. Rowling', 'United Kingdom');

INSERT INTO TBL_BOOKS (BOOK_ID, BOOK_TITLE, AUTHORID) VALUES
(101, '1984', 1),
(102, 'Kafka on the Shore', 2),
(103, 'Harry Potter', 3);

SELECT B.BOOK_TITLE AS 'BOOK TITLE' , A.AUTHOR_NAME, A.COUNTRY
FROM TBL_BOOKS AS B
INNER JOIN
TBL_AUTHOR AS A
ON
B.AUTHORID = A.AUTHOR_ID


--------- medium experiment --------
CREATE TABLE Departments (
   Dept_ID INT PRIMARY KEY,
   Dept_Name VARCHAR(100) 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)
);

INSERT INTO Departments (Dept_ID, Dept_Name) VALUES
(1, 'Computer Science'),
(2, 'Mathematics'),
(3, 'Physics'),
(4, 'Chemistry'),
(5, 'English');

INSERT INTO Courses (Course_ID, Course_Name, Dept_ID) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Database Systems', 1),
(104, 'Linear Algebra', 2),
(105, 'Calculus', 2),
(106, 'Quantum Mechanics', 3),
(107, 'Thermodynamics', 3),
(108, 'Organic Chemistry', 4),
(109, 'British Literature', 5),
(110, 'World Literature', 5);

SELECT  D.Dept_ID,  D.Dept_Name, COUNT(C.Course_ID) AS Course_Count
FROM Departments D
JOIN Courses C ON D.Dept_ID = C.Dept_ID
GROUP BY D.Dept_ID, D.Dept_Name
HAVING COUNT(C.Course_ID) > 2;
```