Experiment- 1

Student Name: RAVI **UID:23BCS10340**
Branch: CSE **Section/Group: KRG-3A**
Semester: 5th **Date of Performance: 18 JUL**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1) Problem Statement :

Author-Book Relationship Using Joins and Basic SQL operations Design two tables — one for storing author details and the other for book details. Ensure a foreign key relationship from the book to its respective author.

Insert at least three records in each table.

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

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

TOOLS USED : Microsoft SQL Server

SQL CODE :

```
CREATE TABLE AUTHOR(
   AUTHOR_ID INT PRIMARY KEY,
   AUTHOR_NAME VARCHAR(20),
   COUNTRY VARCHAR(20)
);

CREATE TABLE BOOK (
   BOOK_ID INT PRIMARY KEY,
   BOOK_TITLE VARCHAR(30),
   AUTHOR_ID INT,
   FOREIGN KEY (AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID)
);

INSERT INTO AUTHOR (AUTHOR_ID, AUTHOR_NAME, COUNTRY) VALUES
(1, 'Ravi', 'India'),
(2, 'Priya Mehta', 'Nepal'),
```

```
(3, 'Amit Verma', 'Sri Lanka'),
(4, 'Anjali Das', 'Bangladesh'),
(5, 'Rajeev Nair', 'Malaysia'),
(6, 'Kavita Reddy', 'Bhutan'),
(7, 'Manoj Kumar', 'Thailand');

INSERT INTO BOOK (BOOK_ID, BOOK_TITLE, AUTHOR_ID) VALUES
(101, 'Mystery of the Monsoon', 1),
(102, 'The Golden Banyan', 2),
(103, 'Whispers of the Ganges', 3),
(104, 'Songs of Sundarbans', 4),
(105, 'Spice Route Diaries', 5),
(106, 'Himalayan Echoes', 6),
(107, 'Twilight over Ayutthaya', 7);

SELECT A.AUTHOR_NAME, A.COUNTRY, B.BOOK_TITLE
FROM AUTHOR AS A
INNER JOIN BOOK AS B ON A.AUTHOR_ID = B.AUTHOR_ID;
```

OUTPUT:

AUTHOR_NAME	COUNTRY	BOOK_TITLE
Ravi	India	Mystery of the Monsoon
Priya Mehta	Nepal	The Golden Banyan
Amit Verma	Sri Lanka	Whispers of the Ganges
Anjali Das	Bangladesh	Songs of Sundarbans
Rajeev Nair	Malaysia	Spice Route Diaries
Kavita Reddy	Bhutan	Himalayan Echoes
Manoj Kumar	Thailand	Twilight over Ayutthaya

2) Problem Statement :

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.

TOOLS USED : Microsoft SQL Server

SQL CODE:

```
CREATE TABLE Department (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(100)
);
CREATE TABLE Course (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(100),
    DeptID INT,
    FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);
INSERT INTO Department (DeptID, DeptName) VALUES
(1, 'Science'),
(2, 'Arts'),
(3, 'Commerce'),
```

```
(4, 'Medical'),
(5, 'Law');

INSERT INTO Course VALUES
(101, 'Physics', 1),
(102, 'Chemistry', 1),
(103, 'Biology', 1),
(104, 'Mathematics', 1),
(105, 'History', 2),
(106, 'English', 2),
(107, 'Economics', 3),
(108, 'Business Studies', 3),
(109, 'Anatomy', 4),
(110, 'First Aid', 4),
(111, 'Indian Constitution', 5),
(112, 'Criminal Law', 5);

SELECT DeptName
FROM Department
WHERE DeptID IN (
    SELECT DeptID
    FROM Course
```

```
    GROUP BY DeptID

    HAVING COUNT(*) > 2

);

CREATE LOGIN Raviz

WITH PASSWORD = 'Ravi2005';

CREATE USER Ravi FOR LOGIN Raviz;

EXECUTE AS USER = 'Ravi';


GRANT SELECT ON Department TO Ravi;

REVOKE SELECT ON Department FROM Ravi;
```

OUTPUT :

```
DeptName
-------------------------
Science
Arts
Commerce
Law
```