EXPERIMENT - 1

Name: Neha Sharma	**UID**: 23BCS10766
Branch: CSE	**Section**: KRG1-B
Semester: 5	**Date of Performance**: 22/05/2025
Subject: ADBMS	**Subject Code**: 23CSP-333

Question 1 : Easy-Level Problem

Problem Title: Author-Book Relationship Using Joins and Basic SQL Operations

Task:

You are required to design and query a relational database involving two tables:

1. Create two tables:
    - Authors (to store details like author ID, name, and country)
    - Books (to store details like book ID, title, and author ID)
2. Establish a foreign key relationship between the Books table and the Authors table using the author_id.
3. Insert at least three sample records into each table.
4. Write an SQL INNER JOIN query that retrieves:
    - Book title
    - Author name
    - Author's country

Expected Output:

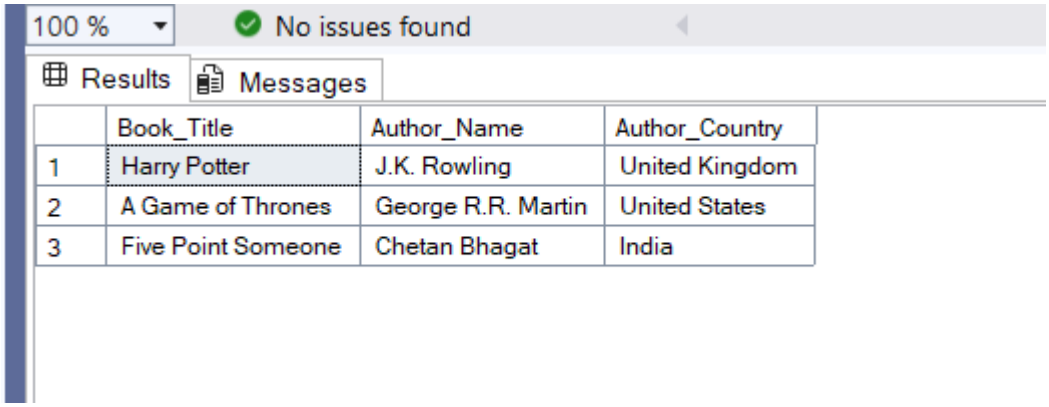
The output of the join should display each book's title along with the name and country of its respective author.

Solution :

Program Code:

```
CREATE TABLE Authors (
    author_id INT PRIMARY KEY,
    name VARCHAR(100),
    country VARCHAR(100)
);

CREATE TABLE Books (
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    author_id INT,
    FOREIGN KEY (author_id) REFERENCES Authors(author_id)
);

INSERT INTO Authors (author_id, name, country) VALUES
(1, 'J.K. Rowling', 'United Kingdom'),
(2, 'George R.R. Martin', 'United States'),
(3, 'Chetan Bhagat', 'India');

INSERT INTO Books (book_id, title, author_id) VALUES
(101, 'Harry Potter', 1),
(102, 'A Game of Thrones', 2),
(103, 'Five Point Someone', 3);

SELECT Books.title AS Book_Title,
Authors.name AS Author_Name,
Authors.country AS Author_Country
FROM Books
INNER JOIN
Authors ON Books.author_id = Authors.author_id;
```

Output:



	Book_Title	Author_Name	Author_Country
1	Harry Potter	J.K. Rowling	United Kingdom
2	A Game of Thrones	George R.R. Martin	United States
3	Five Point Someone	Chetan Bhagat	India

Question 2 : Medium - Level Problem

Problem Title: Department-Course Subquery and Access Control

Task:

You are required to design a relational database schema involving departments and the courses they offer.

1. Create two normalized tables:

- Departments: to store department details.
- Courses: to store course details and associate each with a department.

2. Establish a **foreign key relationship** between Courses and Departments.
3. Insert **at least five departments** and **ten or more courses** across those departments.
4. Write an SQL **subquery** to count the number of courses offered by each department.
5. Retrieve the **names of only those departments** that offer **more than two courses**.
6. Finally, grant **SELECT-only access** on the Courses table to a specific user (e.g., readonly_user).

Expected Output:

A list of department names that have more than two associated courses in the system.

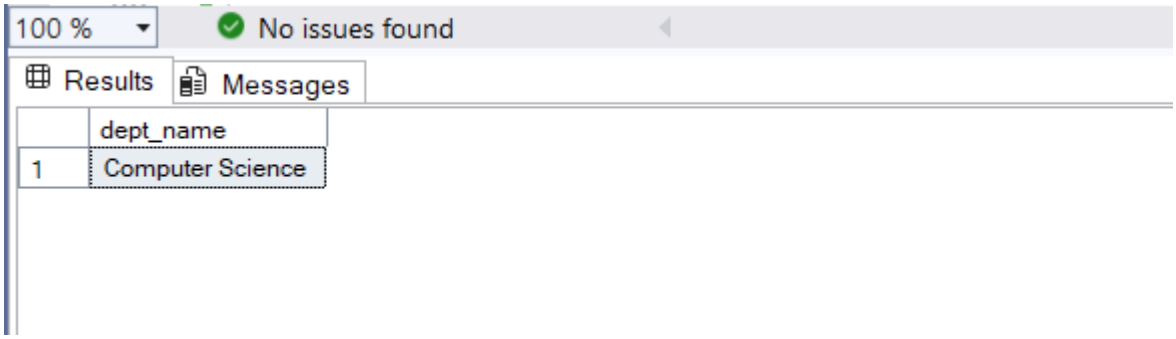
Solution :

Program Code:

```
CREATE TABLE Department (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100)
);
CREATE TABLE Course (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);
INSERT INTO Department (dept_id, dept_name) VALUES
(1, 'Computer Science'),
(2, 'Electronics'),
(3, 'Mechanical'),
(4, 'Mathematics'),
(5, 'Civil');
INSERT INTO Course (course_id, course_name, dept_id) VALUES
(101, 'Data Structures', 1),
(102, 'Algorithms', 1),
```

```
(103, 'Operating Systems', 1),
(104, 'Digital Electronics', 2),
(105, 'Microprocessors', 2),
(106, 'Thermodynamics', 3),
(107, 'Fluid Mechanics', 3),
(108, 'Calculus', 4),
(109, 'Linear Algebra', 4),
(110, 'Structural Engineering', 5);

SELECT dept_name
FROM Department
WHERE dept_id IN (
 SELECT dept_id
 FROM Course
 GROUP BY dept_id
 HAVING COUNT(*) > 2
);
create login login_users with password = 'Neha@123';

create user neha_103 for login login_users;

grant select on courses to neha_103;
```

Output:


100 %
No issues found
Results
Messages
dept_name
1
Computer Science