EXPERIMENT : 1

NAME : Charu Singla **UID: 23BCS12553**

BRANCH : BE-CSE **SECTION/GROUP : KRG-1-A**

SEMESTER : 5TH **SUBJECT CODE : 23CSP-339**

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

[EASY] Author-Book Relationship Using Joins and Basic SQL Operations

1. Design two tables – one for storing author details and the other for book details.
2. Ensure a foreign key relationship from the book to its respective author.
3. Insert at least three records in each table.
4. Perform an INNER JOIN to link each book with its author using the common author ID.
5. Select the book title, author name, and author's country.

[MEDIUM] Department-Course Subquery and Access Control.

1. Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
2. Insert five departments and at least ten courses across those departments.
3. Use a subquery to count the number of courses under each department.
4. Filter and retrieve only those departments that offer more than two courses.
5. Grant SELECT-only access on the courses table to a specific user.

2. Tools Used : Programizz Online Compiler

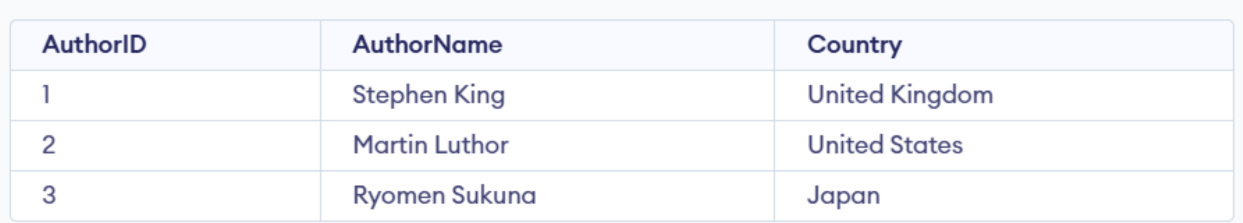
3. Code :

```
===============================
Q1: Authors & Books
===============================

Create Authors table
CREATE TABLE Authors (
   AuthorID INT PRIMARY KEY,
   AuthorName VARCHAR(100),
   Country VARCHAR(100)
);

-- Create Books table
CREATE TABLE Books (
   BookID INT PRIMARY KEY,
   Title VARCHAR(100),
   AuthorID INT,
   FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);



-- Insert data into Authors
INSERT INTO Authors (AuthorID, AuthorName, Country) VALUES
(1, 'Stephen King', 'United Kingdom'),
(2, 'Martin Luthor', 'United States'),
(3, 'Ryomen Sukuna', 'Japan');

-- Insert data into Books
INSERT INTO Books (BookID, Title, AuthorID) VALUES
(101, 'Dragon Ball', 1),
(102, 'Jujitsu Kaisen', 2),
(103, 'Magnus Carlsen', 3);

Select * from Authors;
Select * from Books;

-- Q1
SELECT
   B.Title AS BookTitle,
   A.AuthorName,
   A.Country
FROM Books AS B
INNER JOIN Authors AS A
   ON B.AuthorID = A.AuthorID;



-- ==============================
-- Q2: Departments & Courses
-- ==============================



-- Create Departments table
CREATE TABLE Departments (
   DeptID INT PRIMARY KEY,
   DeptName VARCHAR(100) NOT NULL
);

-- Create Courses table
CREATE TABLE Courses (
   CourseID INT PRIMARY KEY,
   CourseName VARCHAR(100) NOT NULL,
   DeptID INT,
   FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
);
```

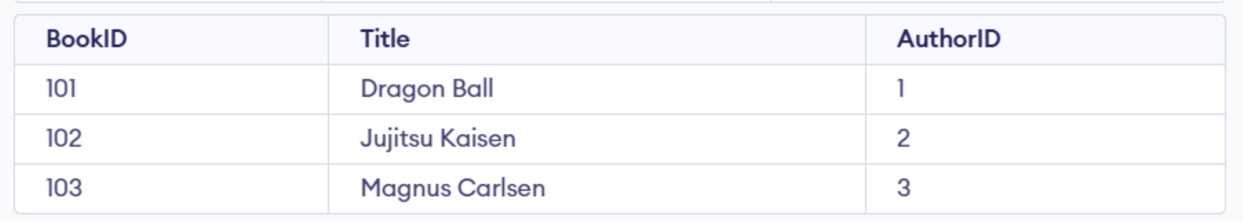
```sql
-- Insert data into Departments
INSERT INTO Departments (DeptID, DeptName) VALUES
(1, 'Computer Science'),
(2, 'AI ML'),
(3, 'ECE'),
(4, 'Mathematics'),
(5, 'Physics');

-- Insert data into Courses
INSERT INTO Courses (CourseID, CourseName, DeptID) VALUES
(101, 'Mathematics', 1),
(102, 'DAA', 1),
(103, 'Operating Systems', 1),
(104, 'Thermodynamics', 2),
(105, 'Fluid Mechanics', 2),
(106, 'DBMS', 3),
(107, 'Signals and Systems', 3),
(108, 'Linear Algebra', 4),
(109, 'Quantum Mechanics', 5),
(110, 'Classical Mechanics', 5),
(111, 'Compiler Design', 1);

Select * from Departments;
Select * from Courses;

-- Q2
SELECT
    DeptName
FROM Departments
WHERE DeptID IN (
    SELECT DeptID
    FROM Courses
    GROUP BY DeptID
    HAVING COUNT(*) > 2
);
```

4. **Output:**

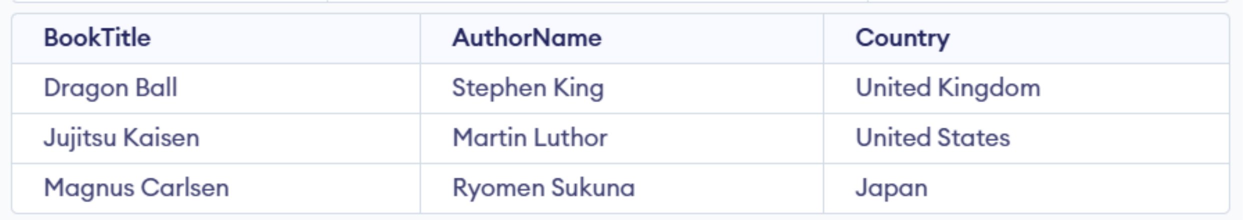
[EASY]



AuthorID	AuthorName	Country
1	Stephen King	United Kingdom
2	Martin Luthor	United States
3	Ryomen Sukuna	Japan

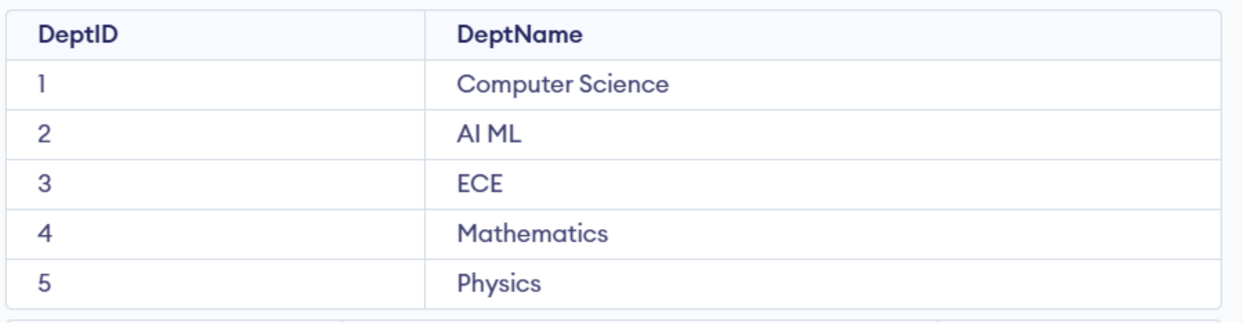


BookID	Title	AuthorID
101	Dragon Ball	1
102	Jujitsu Kaisen	2
103	Magnus Carlsen	3

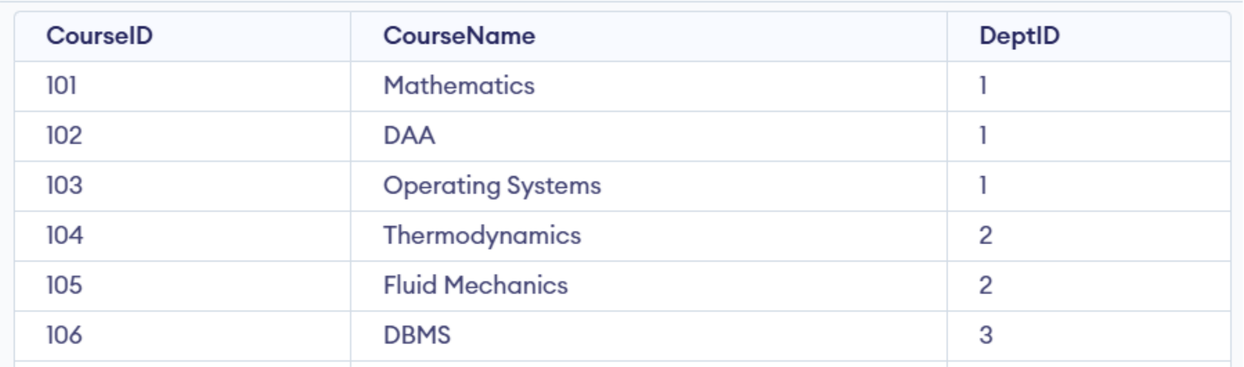


BookTitle	AuthorName	Country
Dragon Ball	Stephen King	United Kingdom
Jujitsu Kaisen	Martin Luthor	United States
Magnus Carlsen	Ryomen Sukuna	Japan

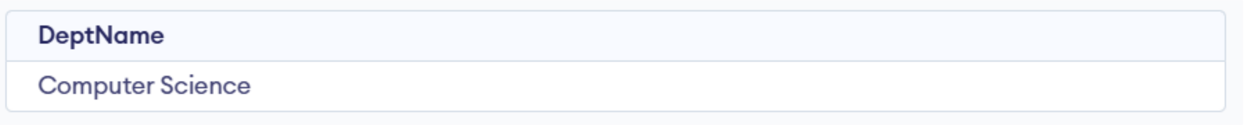
[MEDIUM]



DeptID	DeptName
1	Computer Science
2	AI ML
3	ECE
4	Mathematics
5	Physics



CourseID	CourseName	DeptID
101	Mathematics	1
102	DAA	1
103	Operating Systems	1
104	Thermodynamics	2
105	Fluid Mechanics	2
106	DBMS	3



DeptName
Computer Science

5. Learning Outcomes :

- Learn how to define and create relational database tables using CREATE TABLE syntax. Understand the use of data types like INT and VARCHAR.
- Gain practical knowledge of establishing a primary key for uniquely identifying records.
- Understand how to create and enforce foreign key relationships to maintain data integrity between related tables (Books → Authors).
- Develop the ability to use INNER JOIN to combine data from multiple tables based on a common key (e.g. author_id).
- Understand how to design normalized relational tables with foreign key constraints for real-world entities like departments and courses.
- Gain proficiency in inserting multiple records into related tables using the INSERT INTO statement.
- Learn how to use subqueries with GROUP BY and HAVING to aggregate data and apply conditional logic.
- Apply filtering logic to retrieve records from a parent table based on results from a subquery on a related child table.