Experiment - 1

Student Name: Manit Saxena UID: 23BCS13532

Branch: BE-CSE Section/Group: KRG-1 (B)

Semester: 5th **Date:** 29-07-25

Subject Name: ADBMS Subject Code: 23CSP-333

Aim:

Q1. Author-Book Relationship Using Joins and Basic SQL Operations

Implement the following operations:

- o Design two tables one for storing author details and the other for book details.
- o 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.
- o Select the book title, author name, and author's country.
- o Expected Output: Each book title along with its author's name and country.

Q2. Department-Course Subquery and Access Control

Implement the following operations:

- Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
- o Insert five departments and at least ten courses across those departments.
- o Use a subquery to count the number of courses under each department.
- o Display only departments that offer more than 2 courses.

Objective:

Q1: Author-Book Relationship Using Joins and Basic SQL Operations

The objective of this task is to establish a relationship between authors and books using fundamental SQL operations. Two relational tables will be designed - one to store author details and another to hold book information. A foreign key will be implemented in the books table to reference the corresponding author's ID, ensuring referential integrity. The goal is to insert at least three records into each table and then use an INNER JOIN to combine data based on the shared author ID. The final output should present a list of book titles along with the respective author's name and country, demonstrating how join operations can be used to fetch related data across tables.

Q2: Department-Course Subquery and Access Control

This task focuses on creating a normalized database structure for departments and the courses they offer, emphasizing the use of subqueries and basic access filtering. Two tables will be designed - one for departments and another for courses - with a foreign key linking each course to its respective department. A minimum of five departments and ten courses will be inserted to provide sufficient data for analysis. A subquery will be employed to count the number of courses associated with each department. The final query will display only those departments that offer more than two courses, demonstrating how subqueries and filtering can be used for meaningful data extraction and control.

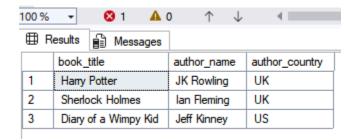
DBMS Code & Output:

Q1: Author-Book Relationship Using Joins and Basic SQL Operations

```
create database KRG 1B;
use KRG 1B;
-- Authors table
create table Authors (
  author id int primary key,
  name varchar(100),
  country varchar(50)
);
-- Books table with a foreign key referencing Authors
create table Books (
  book_id int primary key,
  title varchar(100),
  author id int,
  foreign key (author id) references Authors(author id)
);
-- Sample data for Authors
insert into Authors
values
  (1, 'JK Rowling', 'UK'),
  (2, 'Ian Fleming', 'UK'),
  (3, 'Jeff Kinney', 'US');
-- Sample data for Books
insert into Books
values
  (101, 'Harry Potter', 1),
  (102, 'Sherlock Holmes', 2),
  (103, 'Diary of a Wimpy Kid', 3);
-- Output inner join
select
  b.title as 'book title',
  a.name as 'author name',
```

Discover. Learn. Empower.

a.country as 'author_country'
from
Books b
inner join
Authors a on b.author_id = a.author_id;



Q2: Department-Course Subquery and Access Control

use KRG 1B;

```
-- Departments table
create table Departments (
  dept_id int primary key,
  dept_name varchar(100)
);
-- Courses table with foreign key referencing Departments
create table Courses (
  course_id int primary key,
  course_name varchar(100),
  dept id int,
  foreign key (dept_id) references Departments(dept_id)
);
-- Data for Departments
insert into Departments
values
  (1, 'Computer Science'),
  (2, 'Electrical Engineering'),
  (3, 'Mechanical Engineering'),
  (4, 'Civil Engineering'),
  (5, 'Mathematics');
-- Data for Courses
insert into Courses
values
  (201, 'Data Structures', 1),
  (202, 'Operating Systems', 1),
  (203, 'Database Systems', 1),
  (204, 'Circuits', 2),
  (205, 'Signals and Systems', 2),
  (206, 'Thermodynamics', 3),
  (207, 'Fluid Mechanics', 3),
  (208, 'Statics', 4),
  (209, 'Linear Algebra', 5),
  (210, 'Calculus', 5);
```

```
-- Subquery to get departments offering more than 2 courses select
d.dept_id,
d.dept_name
from
Departments d
where (
select count(*)
from
```

Courses c where

 $c.dept_id = d.dept_id$

Query executed successfully.

```
100 % ✓ No issues found

☐ Results ☐ Messages

☐ dept_id dept_name

1 1 1 Computer Science
```