# Experiment 1

**Student Name: Yash Goel** **UID: 23BCS11498**
**Branch: B.E.CSE** **Section/Group: 23BCS-KRG-2B**
**Semester: 5th** **Date of Performance:21-07-25**
**Subject Name: ADBMS** **Subject Code:23CSP-333**

## 1. Aim:

a) Create a database schema to model the relationship between authors and books using SQL. Begin by designing two tables: one named `Authors` to store details such as the author's ID, name, and country; and another named `Books` to store details such as the book's ID, title, and a foreign key that references the corresponding author's ID. Populate each table with at least three sample records. Then, write an SQL query using an `INNER JOIN` to retrieve a list that links each book with its respective author. The final output should display the book title, the author's name, and the author's country. This exercise will demonstrate your ability to design relational tables, establish foreign key relationships, and perform join operations to combine related data across multiple tables.

**b)** You Design and implement a relational database schema to represent academic departments and the courses they offer. Start by creating two normalized tables: one for `Departments`, storing each department's ID and name, and another for `Courses`, containing course details along with a foreign key referencing the corresponding department. Populate the `Departments` table with five entries and the `Courses` table with at least ten courses distributed across those departments. Next, use a subquery to count how many courses are associated with each department, and retrieve only those departments that offer more than two courses. Finally, demonstrate the application of access control by granting `SELECT`-only permission on the `Courses` table to a specific user. This exercise evaluates your ability to normalize data, use subqueries effectively, filter based on aggregate conditions, and implement basic SQL access control.

## 2. Objective:

- To understand how to use JOINS in SQL.
- To understand the basic SQL Queries.
- To generate hierarchical reports from self-referencing tables.

## 3. DBMS script and output:

```
--EASY

CREATE TABLE AUTHOR_DETAILS(
    authID INT,
    authName VARCHAR(100),
    authCountry VARCHAR(100)
)

CREATE TABLE BOOK_DETAILS(
    bookTitle VARCHAR(100),
    authID INT
)

INSERT INTO AUTHOR_DETAILS(authID, authName, authCountry) VALUES
(1, 'Archit', 'India'),
(2, 'Dr. Park', 'USA'),
(3, 'Sherron', 'Australia')

INSERT INTO BOOK_DETAILS(bookTitle, authID) VALUES
('The Wheel Of Time', 1),
('Dear Diary', 2),
('The Dark Knight', 1),
('Sunrise', 3),
('The Summer of Salvia', 2),
('Death of Mr. Shaw', 3)

SELECT B.bookTitle AS [Book Title], A.authName AS [Author Name], A.authCountry AS [Author Country]
FROM AUTHOR_DETAILS AS A
INNER JOIN
BOOK_DETAILS AS B
ON
A.authID = B.authID
```

Results | Messages

| | Book Title | Author Name | Author Country |
|---|---|---|---|
| 1 | The Wheel Of Time | Archit | India |
| 2 | Dear Diary | Dr. Park | USA |
| 3 | The Dark Knight | Archit | India |
| 4 | Sunrise | Sherron | Australia |
| 5 | The Summer of Salvia | Dr. Park | USA |
| 6 | Death of Mr. Shaw | Sherron | Australia |

```sql
--MEDIUM

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, 'Biology'),
(5, 'History');

INSERT INTO Courses (course_id, course_name, dept_id) VALUES
(101, 'Data Structures', 1),
(102, 'Algorithms', 1),
(103, 'Operating Systems', 1),
(104, 'Linear Algebra', 2),
(105, 'Calculus', 2),
(106, 'Quantum Mechanics', 3),
(107, 'Classical Mechanics', 3),
(108, 'Genetics', 4),
(109, 'Microbiology', 4),
(110, 'World History', 5);

SELECT dept_name
FROM Departments
WHERE dept_id IN (
    SELECT dept_id
    FROM Courses
    GROUP BY dept_id
    HAVING COUNT(*) > 2
);

GRANT SELECT ON Courses TO readonly_user;
```

Results | Messages

| | dept_name |
|---|---|
| 1 | Computer Science |

## 4. Learning outcomes:

- You will be able to write basic SQL queries.
- You will learn to perform JOINS in SQL.
- You will understand how to implement foreign keys.