



University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 1

NAME : Maheshwar

UID: 23BCS11580

BRANCH : BE-CSE

SECTION/GROUP : KRG_2A

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 : SQL Server Management Studio



3. Code

(Easy)

```
CREATE TABLE Author (  
    author_id INT PRIMARY KEY,  
    author_name VARCHAR(30),  
    nationality VARCHAR(30)  
);  
  
CREATE TABLE Book (  
    book_id INT PRIMARY KEY,  
    title VARCHAR(50),  
    author_id INT,  
    FOREIGN KEY (author_id) REFERENCES Author(author_id)  
);  
  
INSERT INTO Author (author_id, author_name, nationality)  
VALUES  
    (1, 'Maheshwar', 'India'),  
    (2, 'Raghav', 'Japan'),  
    (3, 'Manan', 'South Korea'),  
    (4, 'Rohit', 'North Korea'),  
    (5, 'Abhinav', 'Italy');  
  
INSERT INTO Book (book_id, title, author_id)  
VALUES  
    (1001, 'Advanced Data Structures', 1),  
    (1002, 'C++ Programming', 2),  
    (1003, 'Operating System', 1),  
    (1004, 'System Design', 4),  
    (1005, 'Mathematics', 5);  
  
SELECT * FROM Author;  
SELECT * FROM Book;  
  
SELECT  
    A.author_id AS [Author ID],  
    A.author_name AS [Author Name],  
    A.nationality,  
    B.book_id AS [Book ID],  
    B.title AS [Book Title]  
FROM Author A  
JOIN Book B ON A.author_id = B.author_id;
```

(Medium)

```
CREATE TABLE Departments (  
    DeptID INT PRIMARY KEY,  
    DeptName VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY,  
    CourseName VARCHAR(100) NOT NULL,  
    DeptID INT FOREIGN KEY REFERENCES Departments(DeptID)  
);  
  
INSERT INTO Departments (DeptID, DeptName) VALUES  
    (1, 'Computer Science'),  
    (2, 'Mechanical'),  
    (3, 'Civil'),  
    (4, 'Electrical'),  
    (5, 'Mechatronics');  
  
INSERT INTO Courses (CourseID, CourseName, DeptID) VALUES  
    (101, 'Data Structures', 1),  
    (102, 'DBMS', 1),  
    (103, 'Operating System', 1),  
    (104, 'Fluid Mechanics', 2),  
    (105, 'Workshop', 2),  
    (106, 'Highway Engineering', 3),  
    (107, 'RCC', 3),  
    (108, 'Digital Electronics', 4),  
    (109, 'Alternating Current', 4),  
    (110, 'Robotics', 5),  
    (111, 'AI', 5);  
  
SELECT d.DeptName,  
    (SELECT COUNT(*)  
     FROM Courses c  
     WHERE c.DeptID = d.DeptID) AS CourseCount  
FROM Departments d;  
  
SELECT d.DeptName  
FROM Departments d  
WHERE (SELECT COUNT(*)  
       FROM Courses c  
       WHERE c.DeptID = d.DeptID) > 2;
```



4. Output

(Easy)

	Author ID	Author Name	nationality	Book ID	Book Title
1	1	Maheshwar	India	1001	Advanced Data Structures
2	2	Raghav	Japan	1002	C++ Programming
3	1	Maheshwar	India	1003	Operating System
4	4	Rohit	North K...	1004	System Design
5	5	Abhinav	Italy	1005	Mathematics

(Medium)

	DeptName	CourseCount
1	Computer Science	3
2	Mechanical	2
3	Civil	2
4	Electrical	2
5	Mechatronics	2

	DeptName
1	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.