University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 1

NAME : SARTHAK UID : 23BCS11978

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 o er, 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 o er 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(MAX),
  Country VARCHAR(MAX)
```

```
);


CREATE TABLE Book (
  Book_id INT PRIMARY KEY,
  Book_Name VARCHAR(MAX),
  Author_id INT,
  FOREIGN KEY (Author_id) REFERENCES Author(Author_id)
);

INSERT INTO Author (Author_id, Author_Name, Country) VALUES
(1, 'ABC', 'India'),
(2, 'EFG', 'US'),
(3, 'XYZ', 'China'),
(4, 'MNO', 'Japan');

INSERT INTO Book (Book_id, Book_Name, Author_id) VALUES
(11, 'Harry Potter', 1),
(12, 'A Game of Thrones', 2),
(13, 'Norwegian Wood', 3);


SELECT a.Author_Name,a.Country,b.Book_Name
  from  Author as a
     inner join
       Book as b
       ON
       a.Author_id=b.Author_id



SELECT a.*,b.*
  from  Author as a
     left outer join
       Book as b
       ON
       a.Author_id=b.Author_id

SELECT a.*,b.*
  from  Author as a
     right outer join
       Book as b
       ON
       a.Author_id=b.Author_id


---------- MEDIUM ----------

CREATE TABLE Departments (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL
);

CREATE TABLE Courses (
    CourseID INT PRIMARY KEY,
    CourseTitle VARCHAR(150) NOT NULL,
    DepartmentID INT,
```

```
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);


INSERT INTO Departments (DepartmentID, DepartmentName) VALUES
(1, 'Computer Science'),
(2, 'Mathematics'),
(3, 'Physics'),
(4, 'Chemistry'),
(5, 'Biology');

SELECT * FROM Departments;

INSERT INTO Courses (CourseID, CourseTitle, DepartmentID) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Algorithms', 1),
(104, 'Calculus I', 2),
(105, 'Linear Algebra', 2),
(106, 'Quantum Mechanics', 3),
(107, 'Classical Mechanics', 3),
(108, 'Organic Chemistry', 4),
(109, 'Cell Biology', 5),
(110, 'Genetics', 5);

SELECT * FROM Courses;

SELECT DepartmentName
FROM Departments
WHERE DepartmentID IN (
    SELECT DepartmentID
    FROM Courses
    GROUP BY DepartmentID
    HAVING COUNT(CourseID) > 2
);
```

4.Output :

[EASY]

	Author_Name	Country	Book_Name
1	ABC	India	Harry Potter
2	EFG	US	A Game of Thrones
3	XYZ	China	Norwegian Wood

	Author_id	Author_Name	Country	Book_id	Book_Name	Author_id
1	1	ABC	India	11	Harry Potter	1
2	2	EFG	US	12	A Game of Thrones	2
3	3	XYZ	China	13	Norwegian Wood	3

	Author_id	Author_Name	Country	Book_id	Book_Name	Author_id
1	1	ABC	India	11	Harry Potter	1
2	2	EFG	US	12	A Game of Thrones	2
3	3	XYZ	China	13	Norwegian Wood	3
4	4	MNO	Japan	NULL	NULL	NULL

[MEDIUM]

	DepartmentID	DepartmentName
1	1	Computer Science
2	2	Mathematics
3	3	Physics
4	4	Chemistry
5	5	Biology

	CourseID	CourseTitle	DepartmentID
3	103	Algorithms	1
4	104	Calculus I	2
5	105	Linear Algebra	2
6	106	Quantum Mechanics	3
7	107	Classical Mechanics	3
8	108	Organic Chemistry	4
9	109	Cell Biology	5
10	110	Genetics	5

	DepartmentName
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.