University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT:1

NAME : SATYA PRAKASH SWAIN **UID : 23BCS101172**
BRANCH : BE-CSE **SECTION : KRG_1A**
SEMESTER : 5TH **SUBJECT : 23CSP-339**
SUBJECT NAME : ADBMS

1. AIM:-

[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(
A_ID INT PRIMARY KEY,
A_NAME VARCHAR(10),
A_COUNTRY VARCHAR(10),
);
INSERT INTO AUTHOR(A_ID,A_NAME,A_COUNTRY)
VALUES(2,'VANSH','INDIA'),(3,'SHANTNU','AMERICA'),(4,'DEEPAK','RUSSIA');
```

```
CREATE TABLE BOOK(
A_ID INT ,
BOOK_ID INT PRIMARY KEY,
BOOK_NAME VARCHAR(20),
FOREIGN KEY (A_ID) REFERENCES AUTHOR(A_ID)
);
INSERT INTO BOOK(A_ID,BOOK_ID,BOOK_NAME) VALUES(2,100,'HOLY
GRAIL'),(3,101,'A ROOM'),(4,102,'THE ROBBERY');
SELECT * FROM AUTHOR;
SELECT * FROM BOOK;
SELECT
BOOK.BOOK_ID,
BOOK.BOOK_NAME,
AUTHOR.A_NAME,
AUTHOR.A_COUNTRY
FROM
        BOOK
INNER JOIN
        AUTHOR
ON
        BOOK.A_ID = AUTHOR.A_ID;
SELECT
BOOK.BOOK_NAME,
AUTHOR.A_NAME,
AUTHOR.A_COUNTRY
FROM
BOOK
INNER JOIN
AUTHOR
ON
BOOK.A_ID = AUTHOR.A_ID;



--MEDIUM--
CREATE TABLE DEPARTMENT (
   DEPT_ID INT PRIMARY KEY,
   DEPT_NAME VARCHAR(50)
);
CREATE TABLE COURSE (
   COURSE_ID INT PRIMARY KEY,
   COURSE_NAME VARCHAR(50),
   DEPT_ID INT,
   FOREIGN KEY (DEPT_ID) REFERENCES DEPARTMENT(DEPT_ID)
);
INSERT INTO DEPARTMENT (DEPT_ID, DEPT_NAME) VALUES
(1, 'Computer Science'),
(2, 'Mathematics'),
```

```
(3, 'Physics'),
(4, 'Chemistry'),
(5, 'English');
INSERT INTO COURSE (COURSE_ID, COURSE_NAME, DEPT_ID) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Calculus I', 2),
(104, 'Linear Algebra', 2),
(105, 'Classical Mechanics', 3),
(106, 'Quantum Physics', 3),
(107, 'Organic Chemistry', 4),
(108, 'Inorganic Chemistry', 4),
(109, 'English Literature', 5),
(110, 'Creative Writing', 5);
SELECT * FROM DEPARTMENT;
SELECT * FROM COURSE;
SELECT
   DEPT_NAME,
   (SELECT COUNT(*)
    FROM COURSE
    WHERE COURSE.DEPT_ID = DEPARTMENT.DEPT_ID) AS COURSE_COUNT
FROM
   DEPARTMENT
WHERE
   (SELECT COUNT(*)
    FROM COURSE
    WHERE COURSE.DEPT_ID = DEPARTMENT.DEPT_ID) >= 2;
```

4.OUTPUT:-

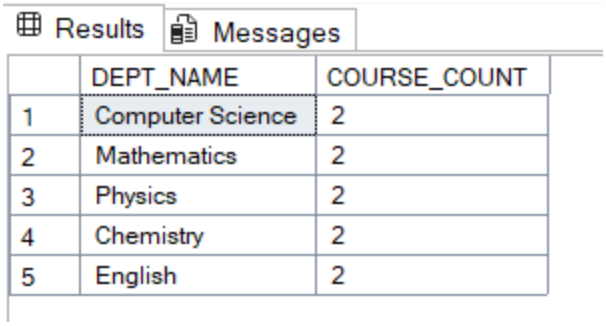
[EASY]

Results | Messages

	BOOK_ID	BOOK_NAME	A_NAME	A_COUNTRY
1	100	HOLY GRAIL	VANSH	INDIA
2	101	A ROOM	SHANTNU	AMERICA
3	102	THE ROBBERY	DEEPAK	RUSSIA

	BOOK_NAME	A_NAME	A_COUNTRY
1	HOLY GRAIL	VANSH	INDIA
2	A ROOM	SHANTNU	AMERICA
3	THE ROBBERY	DEEPAK	RUSSIA

[MEDIUM]



Results | Messages

	DEPT_NAME	COURSE_COUNT
1	Computer Science	2
2	Mathematics	2
3	Physics	2
4	Chemistry	2
5	English	2

5.LEARNING OUTCOMES:-

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