Experiment 1

Student Name: Muskan

UID:23BAI70172

Branch: BE-AIT-CSE

Section/Group:23AML-1(A)

Semester:5th

Date of Performance:23 july,2025

Subject Name:ADBMS

Subject Code:23CSP-333

EASY LEVEL

1. Problem Title: Author-Book Relationship Using Joins and Basic SQL Operations.

2. Procedure (Step-by-Step): Design two tables — one for storing author details and the other for book details.

a. Ensure a foreign key relationship from the book to its respective author.
b. Insert at least three records in each table.
c. Perform an INNER JOIN to link each book with its author using the common author ID.
d. Select the book title, author name, and author's country.

3. Sample Output Description: When the join is performed, we get a list where each book title is shown along with its author's name and their country.

4. SQL Commands:

a. Create the database:

```
create database AIT_1A;

use AIT_1A;
```

b. Create tables for author and books

```sql
CREATE TABLE author (
    author_id INT PRIMARY KEY,
    author_name VARCHAR(MAX),
    experience INT,
    country VARCHAR(50)
);

CREATE TABLE books (
    book_id INT PRIMARY KEY,
    book_name VARCHAR(MAX),
    author_id INT FOREIGN KEY(author_id) REFERENCES author(author_id)
);
```

c. Insert the values in the tables:

```sql
INSERT INTO author VALUES (1, 'Aman', 10, 'India');
INSERT INTO author VALUES (2, 'Naman', 20, 'UK');
INSERT INTO author VALUES (3, 'Mantu', 5, 'USA');
INSERT INTO author VALUES (4, 'Andrew', 7, 'Canada');
INSERT INTO author VALUES (5, 'Nandu', 3, 'India');

INSERT INTO books VALUES (101, 'Learn SQL', 1);
INSERT INTO books VALUES (102, 'Data Science 101', 2);
INSERT INTO books VALUES (103, 'Advanced Python', 3);
INSERT INTO books VALUES (104, 'Database Basics', 1);
INSERT INTO books VALUES (105, 'Machine Learning', 4);
```

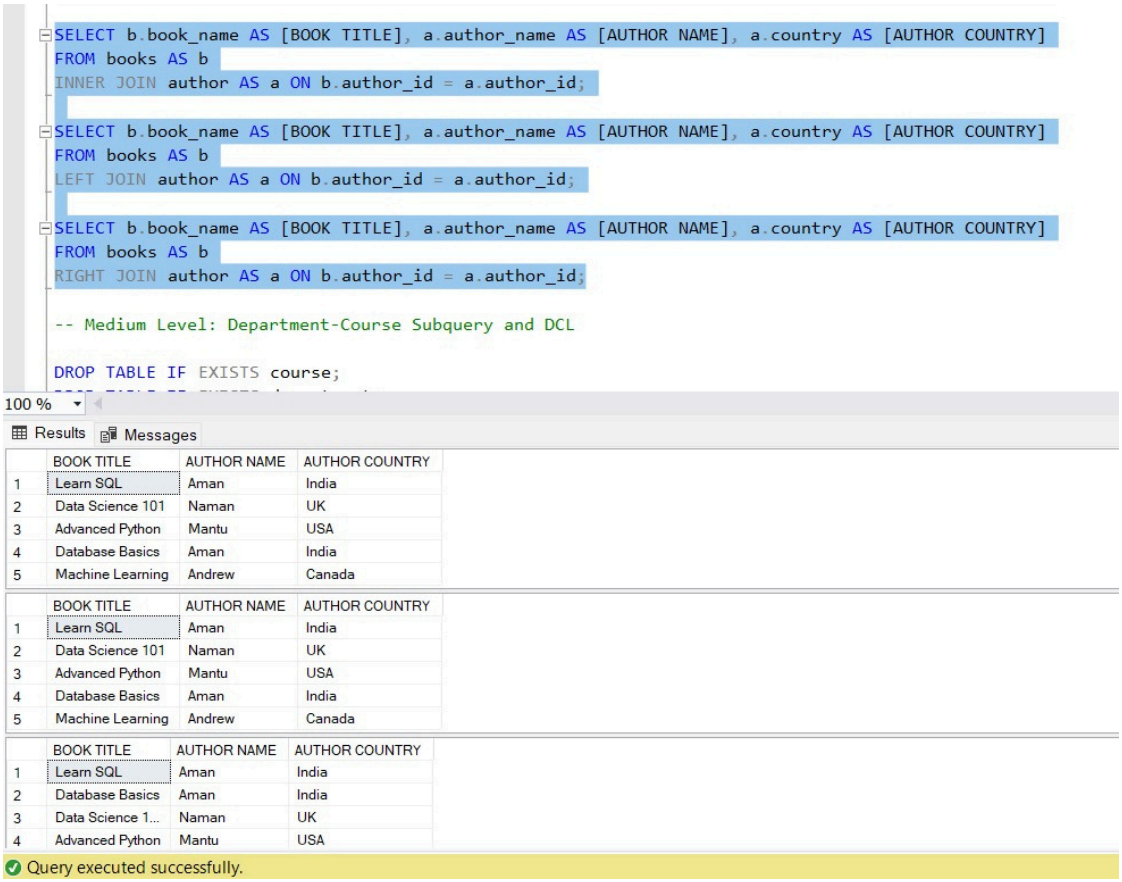
d. Selecting book title,author name and author's country using joins

```sql
SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
INNER JOIN author AS a ON b.author_id = a.author_id;

SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
LEFT JOIN author AS a ON b.author_id = a.author_id;

SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
RIGHT JOIN author AS a ON b.author_id = a.author_id;
```

5. Outputs:



```
SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
INNER JOIN author AS a ON b.author_id = a.author_id;

SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
LEFT JOIN author AS a ON b.author_id = a.author_id;

SELECT b.book_name AS [BOOK TITLE], a.author_name AS [AUTHOR NAME], a.country AS [AUTHOR COUNTRY]
FROM books AS b
RIGHT JOIN author AS a ON b.author_id = a.author_id;

-- Medium Level: Department-Course Subquery and DCL

DROP TABLE IF EXISTS course;
```

100 %

Results | Messages

	BOOK TITLE	AUTHOR NAME	AUTHOR COUNTRY
1	Learn SQL	Aman	India
2	Data Science 101	Naman	UK
3	Advanced Python	Mantu	USA
4	Database Basics	Aman	India
5	Machine Learning	Andrew	Canada

	BOOK TITLE	AUTHOR NAME	AUTHOR COUNTRY
1	Learn SQL	Aman	India
2	Data Science 101	Naman	UK
3	Advanced Python	Mantu	USA
4	Database Basics	Aman	India
5	Machine Learning	Andrew	Canada

	BOOK TITLE	AUTHOR NAME	AUTHOR COUNTRY
1	Learn SQL	Aman	India
2	Database Basics	Aman	India
3	Data Science 1...	Naman	UK
4	Advanced Python	Mantu	USA

Query executed successfully.

6. Learning Outcomes:

a. Learnt how to create and use the database.

b. Learnt how to create tables and insert values in that.

c. Learnt to perform the joins for retrieving the combined data.

MEDIUM LEVEL

1. **Problem Title:** Course Subquery and Access Control
2. **Procedure (Step-by-Step):**
   a. Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
   b. Insert five departments and at least ten courses across those departments.
   c. Use a subquery to count the number of courses under each department and Filter and retrieve only those departments that offer more than two courses.
   d. Grant SELECT-only access on the courses table to a specific user.
3. **Sample Output Description:**
   The result shows the names of departments which are associated with more than two courses in the system.
4. **SQL Commands:**
   a. Create the tables:

```sql
CREATE TABLE department (
    deptid INT PRIMARY KEY,
    dept_name VARCHAR(50) NOT NULL
);

CREATE TABLE course (
    courseid INT PRIMARY KEY,
    coursename VARCHAR(100) NOT NULL,
    deptid INT,
    FOREIGN KEY(deptid) REFERENCES department(deptid)
);
```

b. Insert values into tables:

```sql
INSERT INTO author VALUES (1, 'Aman', 10, 'India');
INSERT INTO author VALUES (2, 'Naman', 20, 'UK');
INSERT INTO author VALUES (3, 'Mantu', 5, 'USA');
INSERT INTO author VALUES (4, 'Andrew', 7, 'Canada');
INSERT INTO author VALUES (5, 'Nandu', 3, 'India');

INSERT INTO books VALUES (101, 'Learn SQL', 1);
INSERT INTO books VALUES (102, 'Data Science 101', 2);
INSERT INTO books VALUES (103, 'Advanced Python', 3);
INSERT INTO books VALUES (104, 'Database Basics', 1);
INSERT INTO books VALUES (105, 'Machine Learning', 4);
```

c. Use a subquery to count the number of courses under each department and two retrieve those who offer more that 2 courses

```sql
SELECT d.deptid, d.dept_name, COUNT(*) AS count
FROM course c
JOIN department d ON c.deptid = d.deptid
GROUP BY d.deptid, d.dept_name
HAVING COUNT(*) > 2;
```

d. Grant and revoke access on the courses table to a specific user.

```sql
-- DCL: User Creation and Permission Test

CREATE LOGIN Muskan WITH PASSWORD = '@1Muskan@1';
CREATE USER Muskan FOR LOGIN Muskan;
GRANT SELECT, INSERT, UPDATE, DELETE ON course TO Muskan;
INSERT INTO course VALUES (115, 'exempler Chemistry', 5);
REVOKE INSERT ON course FROM Muskan;
EXECUTE AS USER = 'Muskan';
INSERT INTO course VALUES (113, 'Modern abc', 4);
REVERT;
```

5.Output:

```sql
SELECT d.deptid, d.dept_name, COUNT(*) AS count
FROM course c
JOIN department d ON c.deptid = d.deptid
GROUP BY d.deptid, d.dept_name
HAVING COUNT(*) > 2;

-- DCL: User Creation and Permission Test

CREATE LOGIN Muskan WITH PASSWORD = '@1Muskan@1';
CREATE USER Muskan FOR LOGIN Muskan;
GRANT SELECT, INSERT, UPDATE, DELETE ON course TO Muskan;
INSERT INTO course VALUES (111, 'exempler Chemistry', 5);
REVOKE INSERT ON course FROM Muskan;
EXECUTE AS USER = 'Muskan';
INSERT INTO course VALUES (113, 'Modern abc', 4);
```

.00 %

Results | Messages

	deptid	dept_name	count
1	1	Computer Science	3
2	3	Physics	3

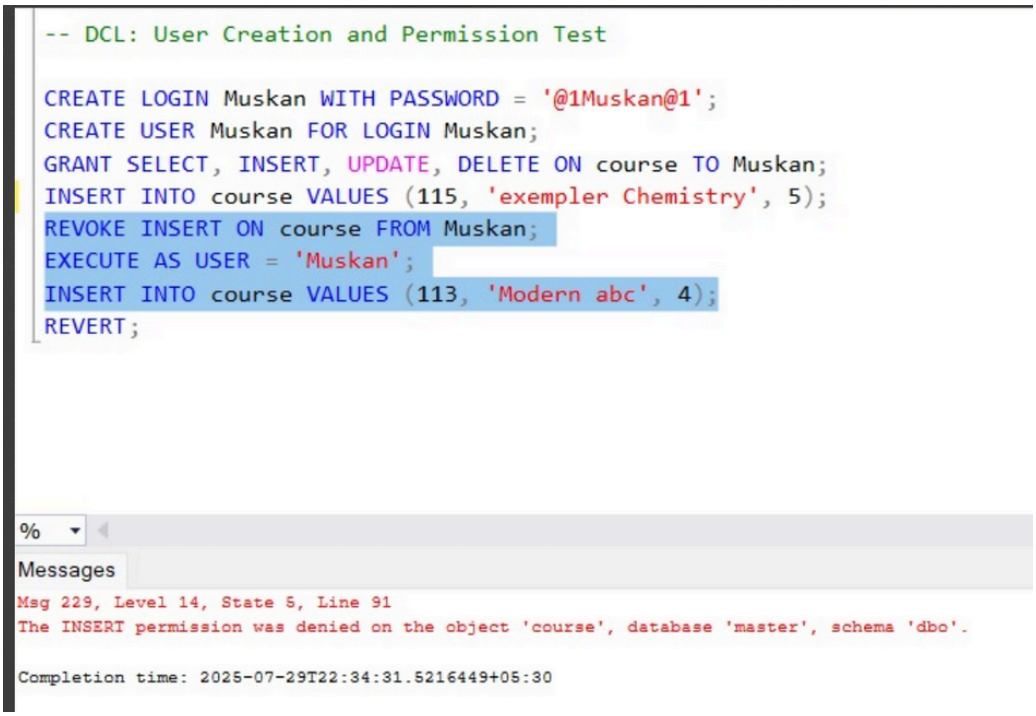
```sql
CREATE LOGIN Muskan WITH PASSWORD = '@1Muskan@1';
CREATE USER Muskan FOR LOGIN Muskan;
GRANT SELECT, INSERT, UPDATE, DELETE ON course TO Muskan;
INSERT INTO course VALUES (115, 'exempler Chemistry', 5);
REVOKE INSERT ON course FROM Muskan;
EXECUTE AS USER = 'Muskan';
INSERT INTO course VALUES (113, 'Modern abc', 4);
REVERT;
```

100 %

Messages

```
(1 row affected)

Completion time: 2025-07-29T22:34:05.9887245+05:30
```



6. Learning Outcomes:

a. Learnt how to normalized tables with foreign key relationship.

b. Learnt subqueries to insert multiple records across database tables.

c. Learnt to perform database access permission to specific users.