CU
CHANDIGARH UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT -1

STUDENT NAME: ASHLIN JAMES	**UID:** 23BAI70722
BRANCH: BE-AIT-CSE	**SECTION/GROUP:** 23AIT_KRG-G1_A
SEMESTER: 5	**DATE OF PERFORMANCE:** 30 JULY, 2025
SUBJECT NAME: ADBMS	**SUBJECT CODE:** 23CSP-333

AIM:

EASY LEVEL PROBLEM:

To create author and book tables linked by a foreign key, insert sample data, and use an INNER JOIN to display each book’s title with its author’s name and country, demonstrating basic SQL joins and relational design.

MEDIUM LEVEL PROBLEM:

Create normalized tables for departments and courses linked by a foreign key, insert sample data, use a subquery to count and filter departments offering more than two courses, and grant SELECT-only access to a specific user on the courses table, demonstrating subqueries, filtering, and access control in SQL.

OBJECTIVE:

Design related tables for authors-books and departments-courses with foreign keys; insert sample data; use INNER JOIN to link books with authors, subqueries to find departments with over two courses, and grant SELECT-only access to a user—demonstrating core SQL concepts of relational design, data retrieval, and access control.

THEORY:

- Relational Databases and Tables
  - Relational databases organize data into tables (relations).
  - Each table consists of rows (records or tuples) and columns (attributes or fields).
- Keys and Relationships
  - A Primary Key uniquely identifies each record in a table.
  - A Foreign Key establishes a relationship between tables by referencing a primary key in another table.
  - These keys support efficient data storage, retrieval, and maintain data integrity through referential constraints.
- SQL as the Standard Language
  - SQL (Structured Query Language) is the standard language for creating, manipulating, and querying relational databases.
- Key SQL Operations Demonstrated
  - Table Creation and Data Insertion: Defining table schemas with appropriate columns and constraints (including foreign keys), followed by inserting sample data.
  - INNER JOIN: Combines rows from two tables based on matching key columns to retrieve linked information (e.g., books matched to authors).
  - Subqueries with Aggregation: Nested queries that summarize data (e.g., counting courses per department) and filter results using aggregate conditions.
  - Access Control: Managing database security by granting specific privileges (such as SELECT-only permission) to users, controlling data access and ensuring protection.
- Overall Conceptual Foundation
  - These concepts illustrate how to design relational database schemas that model real-world relationships using keys.
  - They show how to retrieve meaningful combined data across tables using joins and subqueries.
  - They demonstrate applying security restrictions for safe and controlled data access.
  - This approach enables organized, consistent, and secure management of interconnected data in practical database environments.

PROCEDURE:

1. Design Tables:

   - Create an Author table to store author details including AUTHOR_ID (Primary Key), AUTHOR_NAME (author's name), and Country (author's country).
   - Create a Book table to store book information with columns BOOK_ID (Primary Key), Title (book title), and AUTHOR_ID (Foreign Key referencing AUTHOR_ID in the Author table to link each book to its author).
   - Create a Department table to hold department information with columns DEPARTMENT_ID (Primary Key) and DEPARTMENT_NAME (department name).
   - Create a Course table to store course details including COURSE_ID (Primary Key), COURSE_NAME (course name), and DEPARTMENT_ID (Foreign Key referencing DEPARTMENT_ID in the Department table to associate each course with a department).

2. Insert Sample Data:
    - Insert at least three records into the Author table, including author details like ID, name, and country.
    - Insert at least three records into the Book table, linking each book to an author using AUTHOR_ID.
    - Insert five department records into the Department table with unique IDs and department names.
    - Insert at least ten course records into the Course table, assigning each course to a department using DEPARTMENT_ID.
3. Perform SQL Operations:
    - Use an INNER JOIN query to retrieve and display each book's title along with the corresponding author's name and author's country by joining the Book and Author tables on the AUTHOR_ID. This join ensures that only books with valid authors are included in the result.
    - Use a subquery with aggregation on the Course table grouped by DEPARTMENT_ID to count the number of courses per department. Filter the departments to retrieve only those having more than two courses based on the result of this subquery.
4. Apply Access Control:
    - Grant SELECT permission on the Course table to a specific user to restrict their data access to read-only. This ensures that the user can view the course data but cannot modify, insert, or delete any records in the table.

CODE:

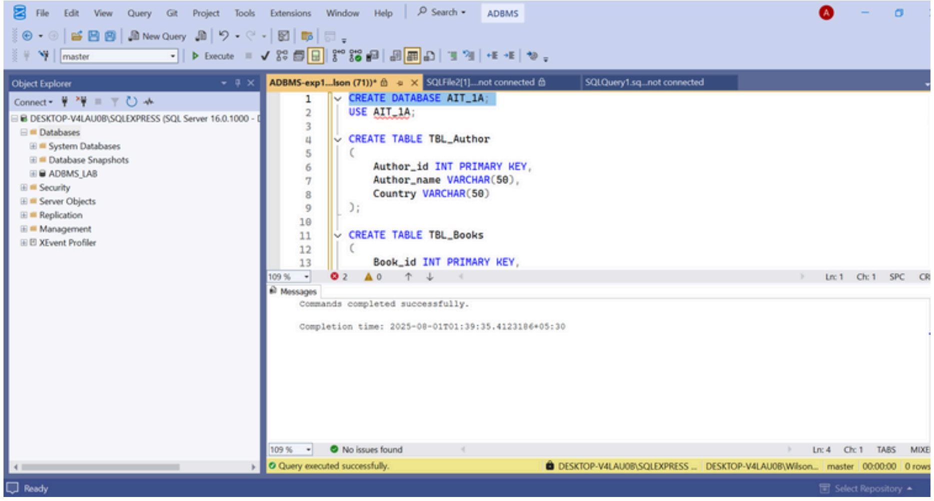
```
--Easy Level Problem

CREATE DATABASE AIT_1A;
USE AIT_1A;

CREATE TABLE TBL_Author
(
   Author_id INT PRIMARY KEY,
   Author_name VARCHAR(50),
   Country VARCHAR(50)
);

CREATE TABLE TBL_Books
(
   Book_id INT PRIMARY KEY,
   Book_title VARCHAR(30),
   AuthorId INT,
   FOREIGN KEY (AuthorId) REFERENCES TBL_Author(Author_id)
);

INSERT INTO TBL_Author (Author_id, Author_name, Country) VALUES
(1, 'Isabel Allende', 'Chile'),
(2, 'Harper Lee', 'United States'),
(3, 'Chimamanda Ngozi Adichie', 'Nigeria'),
(4, 'Neil Gaiman', 'United Kingdom'),
(5, 'Paulo Coelho', 'Brazil'),
(6, 'Salman Rushdie', 'India');

INSERT INTO TBL_Books (Book_id, Book_title, AuthorId) VALUES
(101, 'The House of Spirits', 1),
(102, 'To Kill a Mockingbird', 2),
(103, 'Half of a Yellow Sun', 3),
(104, 'American Gods', 4),
(105, 'The Alchemist', 5),
(106, 'Fury', 6);

SELECT
   B.Book_title AS [Book Name],
   A.Author_name AS [Author Name],
   A.Country AS [Country]
FROM TBL_Books AS B
INNER JOIN TBL_Author AS A
   ON B.AuthorId = A.Author_id;
```

```
--Medium Level Problem

USE AIT_1A;

CREATE TABLE Department
(
 DeptID INT PRIMARY KEY,
 DeptName VARCHAR(100)
);

CREATE TABLE Course
(
 CourseID INT PRIMARY KEY,
 CourseName VARCHAR(100),
 DeptID INT,
 FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);

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

INSERT INTO Course VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Quantum Mechanics', 2),
(104, 'Electromagnetism', 2),
(105, 'Linear Algebra', 3),
(106, 'Calculus', 3),
(107, 'Organic Chemistry', 4),
(108, 'Physical Chemistry', 4),
(109, 'Genetics', 5),
(110, 'Computer Networks', 1),
(111, 'Linux/Unix systems', 1),
(112, 'Matrix', 3),
(113, 'Space Physics', 2);

SELECT D.DeptName,
 (SELECT COUNT(*)
 FROM Course C
 WHERE C.DeptID = D.DeptID) AS CourseCount
FROM Department D;
```

```
SELECT DeptID, DeptName,
 (SELECT COUNT(*)
 FROM Course
 WHERE Course.DeptID = Department.DeptID) AS CourseCount
FROM Department
WHERE (SELECT COUNT(*)
 FROM Course
 WHERE Course.DeptID = Department.DeptID) > 2;

CREATE LOGIN Ashlin WITH PASSWORD='#23BAI70722';
CREATE USER Ashlin FOR LOGIN Ashlin;
GRANT SELECT ON Course TO Ashlin;
```

OUTPUT:

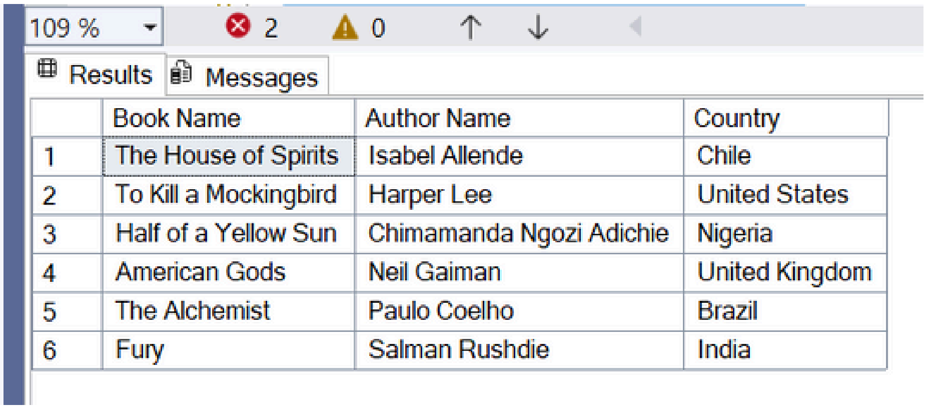
- EASY LEVEL PROBLEM



```
CREATE DATABASE AIT_1A;
USE AIT_1A;

CREATE TABLE TBL_Author
(
    Author_id INT PRIMARY KEY,
    Author_name VARCHAR(50),
    Country VARCHAR(50)
);

CREATE TABLE TBL_Books
(
    Book_id INT PRIMARY KEY,
```

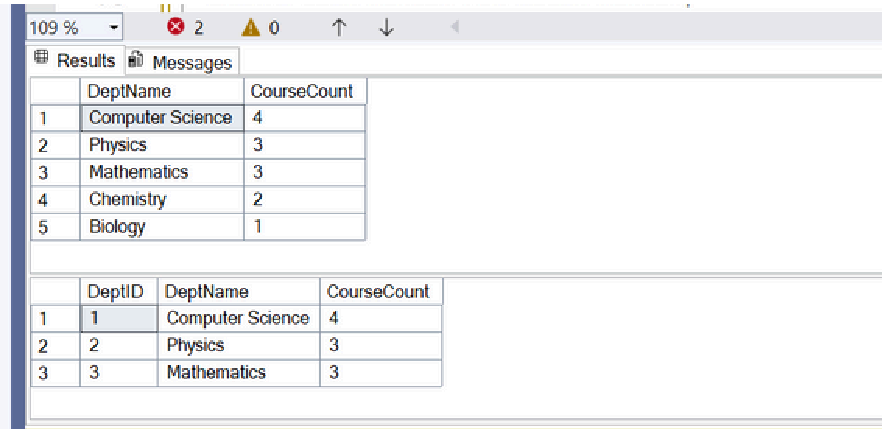
```
Commands completed successfully.

Completion time: 2025-08-01T01:39:35.4123186+05:30
```

Query executed successfully.



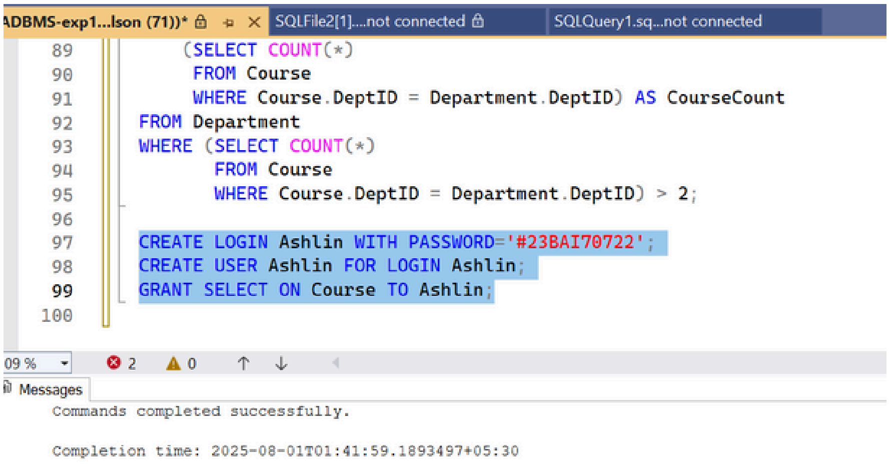
	Book Name	Author Name	Country
1	The House of Spirits	Isabel Allende	Chile
2	To Kill a Mockingbird	Harper Lee	United States
3	Half of a Yellow Sun	Chimamanda Ngozi Adichie	Nigeria
4	American Gods	Neil Gaiman	United Kingdom
5	The Alchemist	Paulo Coelho	Brazil
6	Fury	Salman Rushdie	India

- <u>MEDIUM LEVEL PROBLEM</u>



	DeptName	CourseCount
1	Computer Science	4
2	Physics	3
3	Mathematics	3
4	Chemistry	2
5	Biology	1

	DeptID	DeptName	CourseCount
1	1	Computer Science	4
2	2	Physics	3
3	3	Mathematics	3


ADBMS-exp1....lson (71))*
SQLFile2[1]....not connected
SQLQuery1.sq...not connected
09 %
2
0
Messages

```
        (SELECT COUNT(*)
         FROM Course
         WHERE Course.DeptID = Department.DeptID) AS CourseCount
    FROM Department
    WHERE (SELECT COUNT(*)
           FROM Course
           WHERE Course.DeptID = Department.DeptID) > 2;

    CREATE LOGIN Ashlin WITH PASSWORD='#23BAI70722';
    CREATE USER Ashlin FOR LOGIN Ashlin;
    GRANT SELECT ON Course TO Ashlin;
```

```
Commands completed successfully.

Completion time: 2025-08-01T01:41:59.1893497+05:30
```

<u>LEARNING OUTCOMES:</u>

1. Understand the fundamentals of designing a relational schema that accurately models entities and relationships found in real-world systems, ensuring organized, normalized, and consistent data storage.
2. Demonstrate the ability to create SQL tables with appropriate primary keys and establish links between tables using foreign keys, reflecting true-to-life connections (such as authors and books, or departments and courses).
3. Develop proficiency in writing SQL JOIN queries (such as INNER JOIN, LEFT JOIN, etc.) to combine and retrieve meaningful information from multiple related tables, supporting more comprehensive data analysis.
4. Apply basic database security principles by restricting or granting user access, specifically using GRANT and REVOKE statements to control who can view or manipulate data in specific tables, enhancing overall data protection.