Expeiment 1

Student Name: Pankaj Tayal
UID: 23BCS13706
Branch: CSE
Section/Group: krg-3b
Semester: 5th
Date of Performance: 22/07/25
Subject Name: ADBMS
Subject Code: 23CSP-333

1. Problem Statement:

I. Easy-Level Problem

- ✓ Design two tables — one for storing author details and the other for book details.
- ✓ Ensure a foreign key relationship from the book to its respective author.
- ✓ Insert at least three records in each table.
- ✓ Perform an INNER JOIN to link each book with its author using the common author ID.
- ✓ Select the book title, author name, and author's country.

II. Medium-Level Problem

- ✓ Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
- ✓ Insert five departments and at least ten courses across those departments.
- ✓ Use a subquery to count the number of courses under each department.
- ✓ Filter and retrieve only those departments that offer more than two courses.
- ✓ Grant SELECT-only access on the courses table to a specific user.

2. Code :

Easy Question

```
CREATE TABLE TBL_AUTHOR
 (
 AUTHOR_ID INT PRIMARY KEY,
 AUTHOR_NAME VARCHAR(MAX),
```

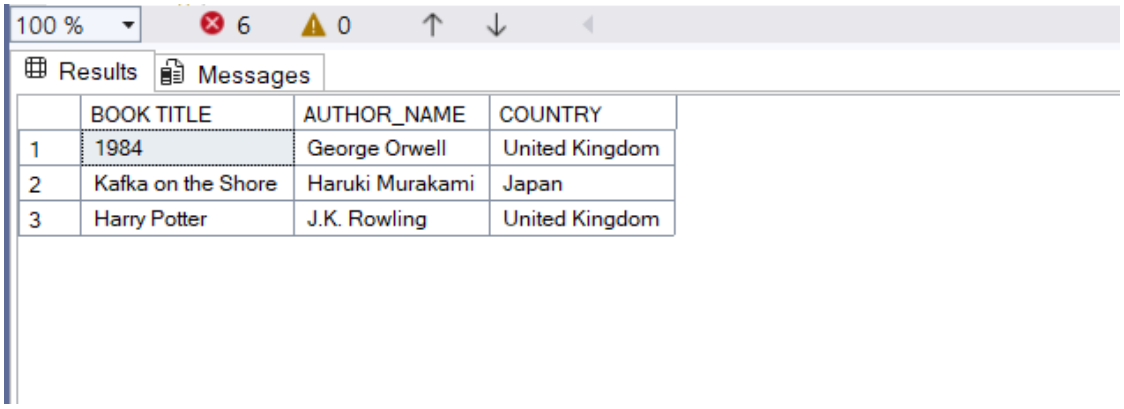
```
 COUNTRY VARCHAR(MAX)

)

CREATE TABLE TBL_BOOKS
(
 BOOK_ID INT PRIMARY KEY,
 BOOK_TITLE VARCHAR(MAX),
 AUTHORID INT
 FOREIGN KEY (AUTHORID) REFERENCES TBL_AUTHOR(AUTHOR_ID)

)

INSERT INTO TBL_AUTHOR (AUTHOR_ID, AUTHOR_NAME, COUNTRY) VALUES

(1, 'George Orwell', 'United Kingdom'),
(2, 'Haruki Murakami', 'Japan'),
(3, 'J.K. Rowling', 'United Kingdom');


INSERT INTO TBL_BOOKS (BOOK_ID, BOOK_TITLE, AUTHORID) VALUES
(101, '1984', 1),
(102, 'Kafka on the Shore', 2),

(103, 'Harry Potter', 3);


SELECT B.BOOK_TITLE AS 'BOOK TITLE' , A.AUTHOR_NAME, A.COUNTRY
FROM TBL_BOOKS AS B
INNER JOIN
TBL_AUTHOR AS A
ON
B.AUTHORID = A.AUTHOR_ID
```

Medium Question

```
CREATE TABLE Departments (
   Dept_ID INT PRIMARY KEY,
   Dept_Name VARCHAR(100) NOT NULL
```

```
);

CREATE TABLE Courses (
    Course_ID INT PRIMARY KEY,
    Course_Name VARCHAR(100) NOT NULL,
    Dept_ID INT,
    FOREIGN KEY (Dept_ID) REFERENCES Departments(Dept_ID)
);

INSERT INTO Departments (Dept_ID, Dept_Name) VALUES
(1, 'Computer Science'),
(2, 'Mathematics'),
(3, 'Physics'),
(4, 'Chemistry'),
(5, 'English');

INSERT INTO Courses (Course_ID, Course_Name, Dept_ID) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Database Systems', 1),
(104, 'Linear Algebra', 2),
(105, 'Calculus', 2),
(106, 'Quantum Mechanics', 3),
(107, 'Thermodynamics', 3),
(108, 'Organic Chemistry', 4),
(109, 'British Literature', 5),
(110, 'World Literature', 5);

SELECT  D.Dept_ID,  D.Dept_Name, COUNT(C.Course_ID) AS Course_Count
FROM Departments D
JOIN Courses C ON D.Dept_ID = C.Dept_ID
GROUP BY D.Dept_ID, D.Dept_Name
HAVING COUNT(C.Course_ID) > 2;
```

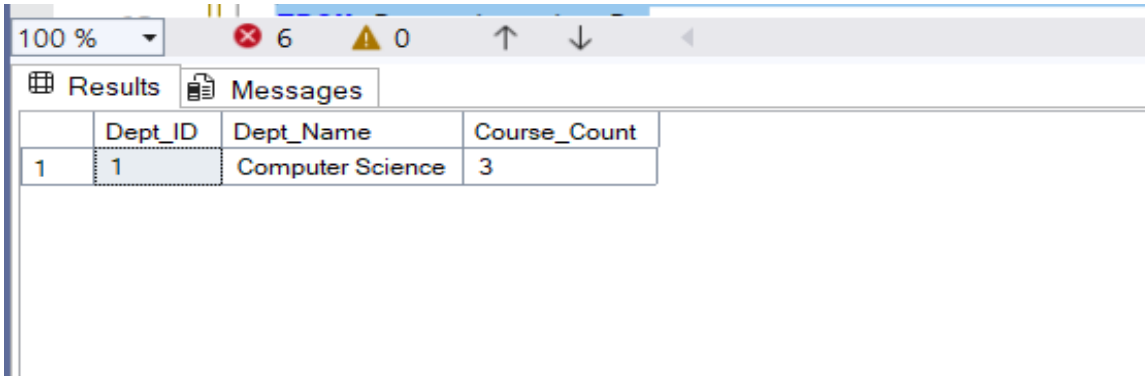
3. Output :



100 % 6 0

Results Messages

	BOOK TITLE	AUTHOR_NAME	COUNTRY
1	1984	George Orwell	United Kingdom
2	Kafka on the Shore	Haruki Murakami	Japan
3	Harry Potter	J.K. Rowling	United Kingdom



100 % 6 0

Results Messages

	Dept_ID	Dept_Name	Course_Count
1	1	Computer Science	3