Experiment 1

Student Name: Jay shankar kumar UID: 23BCS10408
Branch: CSE Section/Group: KRG_1B
Semester: 5th
Subject Name: ADBMS Subject Code: 23CSP-333

1.Aim of the practical

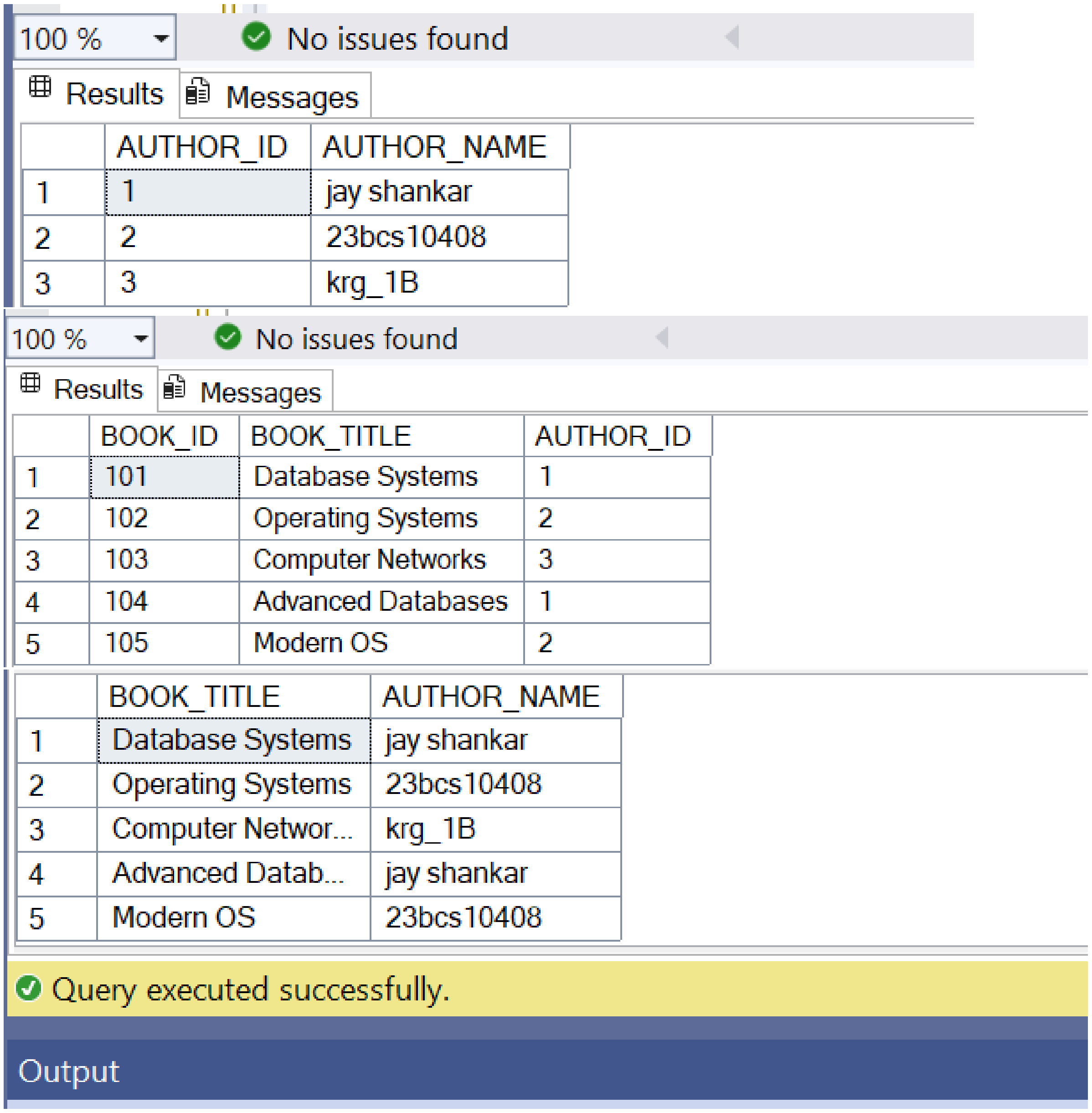
[EASY] Author-Book Relationship Using Joins and Basic SQL Operations

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.

CODE: ------------------------EASY------------------------

```
CREATE TABLE TB_AUTHOR(
AUTHOR_ID INT PRIMARY KEY,
AUTHOR_NAME VARCHAR(30));

CREATE TABLE TB_BOOK(
BOOK_ID INT PRIMARY KEY,
BOOK_TITLE VARCHAR(30),
AUTHOR_ID INT,
FOREIGN KEY (AUTHOR_ID) REFERENCES
TB_AUTHOR(AUTHOR_ID));

INSERT INTO TB_AUTHOR (AUTHOR_ID, AUTHOR_NAME)
VALUES
(1, 'jay shankar'),
(2, '23bcs10408'),
(3, 'krg_1B');

INSERT INTO TB_BOOK (BOOK_ID, BOOK_TITLE, AUTHOR_ID)
VALUES
(101, 'Database Systems', 1),
(102, 'Operating Systems', 2),
(103, 'Computer Networks', 3),
(104, 'Advanced Databases', 1),
(105, 'Modern OS', 2);

SELECT * FROM TB_BOOK;
SELECT * FROM TB_AUTHOR;
SELECT B.BOOK_TITLE ,
A.AUTHOR_NAME
FROM TB_BOOK AS B
INNER JOIN
TB_AUTHOR AS A
ON
B.AUTHOR_ID = A.AUTHOR_ID;
```

OUTPUT:


100 % No issues found

Results Messages

	AUTHOR_ID	AUTHOR_NAME
1	1	jay shankar
2	2	23bcs10408
3	3	krg_1B

100 % No issues found

Results Messages

	BOOK_ID	BOOK_TITLE	AUTHOR_ID
1	101	Database Systems	1
2	102	Operating Systems	2
3	103	Computer Networks	3
4	104	Advanced Databases	1
5	105	Modern OS	2

	BOOK_TITLE	AUTHOR_NAME
1	Database Systems	jay shankar
2	Operating Systems	23bcs10408
3	Computer Networ...	krg_1B
4	Advanced Datab...	jay shankar
5	Modern OS	23bcs10408

Query executed successfully.

Output

[MEDIUM] Department-Course Subquery and Access Control.

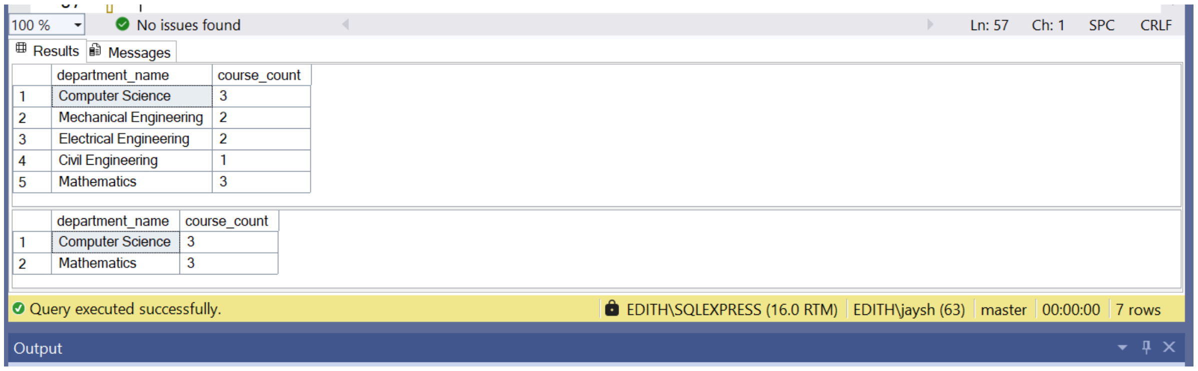
1. Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
2. Insert five departments and at least ten courses across those departments.
3. Use a subquery to count the number of courses under each department.
4. Filter and retrieve only those departments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

CODE:

```
-----------------------------MEDIUM--------------------------------
-- Step 1: Create Tables
CREATE TABLE Departments (
   department_id INT PRIMARY KEY,
   department_name VARCHAR(100) NOT NULL
);

CREATE TABLE Courses (
   course_id INT PRIMARY KEY,
   course_name VARCHAR(100) NOT NULL,
   department_id INT,
   FOREIGN KEY (department_id) REFERENCES
Departments(department_id)
);

-- Step 2: Insert Data into Departments
INSERT INTO Departments (department_id, department_name) VALUES
(1, 'Computer Science'),
(2, 'Mechanical Engineering'),
(3, 'Electrical Engineering'),
(4, 'Civil Engineering'),
(5, 'Mathematics');

-- Step 3: Insert Data into Courses
INSERT INTO Courses (course_id, course_name, department_id) VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
(103, 'Machine Learning', 1),
(104, 'Thermodynamics', 2),
(105, 'Fluid Mechanics', 2),
(106, 'Circuits and Systems', 3),
(107, 'Control Systems', 3),
(108, 'Structural Analysis', 4),
(109, 'Linear Algebra', 5),
(110, 'Calculus', 5),
(111, 'Probability Theory', 5);
```

```
-- Step 4: Count Number of Courses per Department

SELECT
    department_name,
    (SELECT COUNT(*)
     FROM Courses c
     WHERE c.department_id = d.department_id) AS course_count
FROM Departments d;

-- Step 5: Filter Departments Offering More Than 2 Courses
SELECT
    department_name,
    (SELECT COUNT(*)
     FROM Courses c
     WHERE c.department_id = d.department_id) AS course_count
FROM Departments d
WHERE (SELECT COUNT(*)
       FROM Courses c
       WHERE c.department_id = d.department_id) > 2;
```

OUTPUT:



```
100 %    No issues found    Ln: 57   Ch: 1   SPC   CRLF
Results   Messages
```

	department_name	course_count
1	Computer Science	3
2	Mechanical Engineering	2
3	Electrical Engineering	2
4	Civil Engineering	1
5	Mathematics	3

	department_name	course_count
1	Computer Science	3
2	Mathematics	3

```
Query executed successfully.    EDITH\SQLEXPRESS (16.0 RTM) | EDITH\jaysh (63) | master | 00:00:00 | 7 rows
Output
```

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.