CU CHANDIGARH UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 1.1

Student Name:Kushagra Chakraborty	UID: 23BAI70105
Branch: BE-AIT-CSE	Section/Group: 23AML-1 (B)
Semester: 5th	Date of Performance: 22 July 2025
Subject Name: ADBMS	Subject Code: 23CSP-333

1. Experiment Name:
   To design and manipulate a University Database using SQL that involves creating relations Tables for Students, Courses, Enrollments and Professors, inserting and retrieving data Using JOINs, managing access control with GRANT/REVOKE, and handling transactions Control using COMMIT and ROLLBACK.

2. Objective:

   Easy-Level Problem
   Problem Title: Author-Book Relationship Using Joins and Basic SQL Operations Procedure (Step-by-Step):
   Design two tables — one for storing author details and the other for book details.
   1. Ensure a foreign key relationship from the book to its respective author.
   2. Insert at least three records in each table.
   3. Perform an INNER JOIN to link each book with its author using the common author ID.
   4. Select the book title, author name, and author’s country.

   Medium-Level Problem
   Problem Title: Department-Course Subquery and Access Control Procedure (Step-by-Step):
   1. Design normalised 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.

3. Code:

----EASY LEVEL PROBLEM CODE----

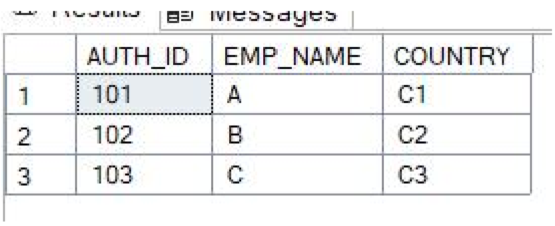
```
CREATE TABLE AUTHOR(
AUTH_ID INT PRIMARY KEY,
EMP_NAME VARCHAR(MAX),
COUNTRY VARCHAR(MAX)
)

create table book
(
BOOK_ID INT PRIMARY KEY,
BOOK_NAME VARCHAR(MAX),
AUTHID INT
FOREIGN KEY (AUTHID) REFERENCES AUTHOR(AUTH_ID)
)
INSERT INTO AUTHOR(AUTH_ID,EMP_NAME,COUNTRY) VALUES(102,'B','C2'),(103,'C','C3')
INSERT INTO book(BOOK_ID,BOOK_NAME,AUTHID)
VALUES(1,'a',101),(2,'b',102),(3,'c',103)
select*from book

SELECT* FROM AUTHOR
select* from AUTHOR as A
inner join
book as b on
```

----MEDIUM LEVEL PROBLEM CODE----

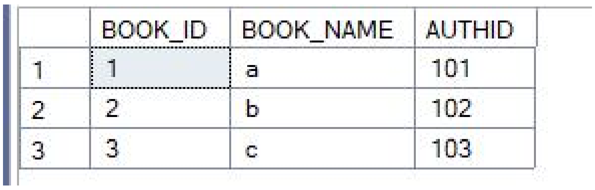
```
        CREATE TABLE Department ( DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100)
        ) ;
        CREATE TABLE Course (
        CourseID INT PRIMARY KEY,
        CourseName VARCHAR(100), DepartmentID INT,
        FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
      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

        -- Insert departments
        INSERT INTO Department (DepartmentID, DepartmentName) VALUES (1, 'Computer
Science'),
        (2, 'Mechanical Engineering'),
        (3, 'Electrical Engineering'),
        (4, 'Mathematics'),
        (5, 'Civil Engineering');

        -- Insert courses

        INSERT INTO Course (CourseID, CourseName, DepartmentID) VALUES (101, 'Data
Structures', 1),
        (102, 'Operating Systems', 1),
        (103, 'DBMS', 1),
        (104, 'Thermodynamics', 2),
        (105, 'Fluid Mechanics', 2),
        (106, 'Circuit Theory', 3),
        (107, 'Digital Electronics', 3),
        (108, 'Linear Algebra', 4),
        (109, 'Calculus', 4),
        (110, 'Structural Analysis', 5);

        SELECT
        DepartmentName, (SELECT COUNT(*)
        FROM Course
        WHERE Course.DepartmentID = Department.DepartmentID) AS CourseCount FROM
    Department;

        SELECT
        DepartmentName FROM
        Department WHERE
        (SELECT COUNT(*)
        FROM Course
           WHERE Course.DepartmentID =

    Department.DepartmentID) > 2; GRANT SELECT ON Course TO

    readonly_use;
```

4. Output:

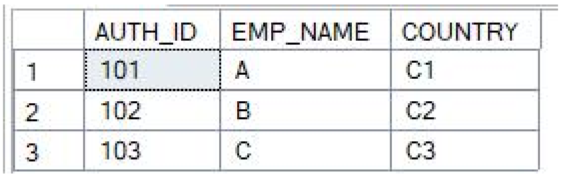
----easy level problem output----



	AUTH_ID	EMP_NAME	COUNTRY
1	101	A	C1
2	102	B	C2
3	103	C	C3

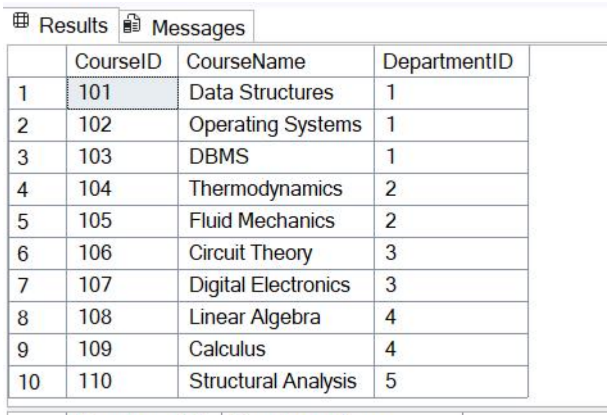


	BOOK_ID	BOOK_NAME	AUTHID
1	1	a	101
2	2	b	102
3	3	c	103

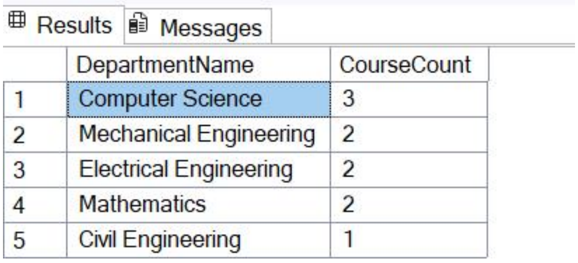


	AUTH_ID	EMP_NAME	COUNTRY
1	101	A	C1
2	102	B	C2
3	103	C	C3

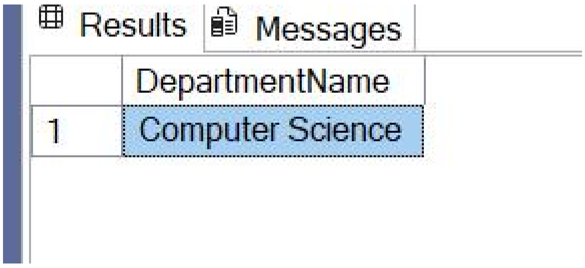
----medium level problem output----



	CourseID	CourseName	DepartmentID
1	101	Data Structures	1
2	102	Operating Systems	1
3	103	DBMS	1
4	104	Thermodynamics	2
5	105	Fluid Mechanics	2
6	106	Circuit Theory	3
7	107	Digital Electronics	3
8	108	Linear Algebra	4
9	109	Calculus	4
10	110	Structural Analysis	5



	DepartmentName	CourseCount
1	Computer Science	3
2	Mechanical Engineering	2
3	Electrical Engineering	2
4	Mathematics	2
5	Civil Engineering	1



	DepartmentName
1	Computer Science

4. Learning Outcomes:

- Understanding Database Normalization:
- Mastering Subqueries in SQL:
- Filtering Data Using Conditions:
- Data Access Control:
- Working with One-to-Many Relationships: