DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CU CHANDIGARH UNIVERSITY Discover. Learn. Empower.

Experiment 1

Student Name: Souradeep Banerjee

UID: 23BAI70654

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Date of Performance: 22nd July, 2025

Subject Name: ADBMS

Subject Code: 23CSP-333

1. **AIM:**

   To design and manipulate a University Database using SQL that involves creating relational tables for Students, Courses, Enrollments, and Professors, inserting and retrieving data using JOINs, managing access control with GRANT/REVOKE, and handling transaction control using COMMIT and ROLLBACK.

2. **Tools Used:**

   SQL Server Management Studio 21 (SSMS) code editor.

3. **Experiment:**

   1. **Easy-Level Problem:** Author-Book Relationship Using Joins and Basic SQL Operations.
      - 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.

   2. **Medium-Level Problem:** Department-Course Subquery and Access Control.
      - 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.

4. Solution:

Easy-Level

```
use Arka;

SELECT name, database_id, create_date
FROM sys.databases;
GO

-- Table author & book_table
create table author(
   AUTHOR_ID int primary key,
   AUTHOR_NAME varchar(20)
)
alter table author add COUNTRY varchar(20);

create table book_table(
   BOOK_ID int primary key identity(200,2),
   BOOK_NAME varchar(20),
   AUTHO_ID int
   foreign key(AUTHO_ID) references author(AUTHOR_ID)

)


-- Operation into author
insert into author values(120, 'Ruskin Bond'), (130, 'Robert Greene'), (145,
'Dale Carnegie');
insert into author values(250, 'Robert Frost');
update author set COUNTRY = 'India' where AUTHOR_ID = 120;
update author set COUNTRY = 'USA' where AUTHOR_ID in (130, 145, 250);
select A.AUTHOR_ID as [Author Id], A.AUTHOR_NAME as 'Author Name', A.COUNTRY as
'Country' from author as A;
select * from author;

-- Operation on book_table
insert into book_table(BOOK_NAME, AUTHO_ID) values('Influence', 145), ('Room on
the Roof', 120), ('Blue Umbrella', 120), ('Human Nature', 130);

select B.BOOK_ID as [ID], B.BOOK_NAME as [Name], B.AUTHO_ID as [Author Id] from
book_table as B


-- INNER JOIN
select B.BOOK_NAME as [Book Name], A.AUTHOR_NAME as [Author Name], A.COUNTRY as
'Country' from book_table as B INNER JOIN author as A on A.AUTHOR_ID =
B.AUTHO_ID;
```

Medium-Level

```sql
--Create Table Dept and Course
create table dept(
   Dept_Id smallint PRIMARY KEY identity(201,3),
   Dept_Name varchar(12)
)

create table course(
   Dept smallint FOREIGN KEY references dept(Dept_Id),
   Course varchar(12)
)

-- Operation on dept
insert into dept(Dept_Name) values('AI&ML'), ('CSE'), ('Bio-Tech'), ('Finance'),
('Psychology');
select * from dept;

-- Operation on Course
insert into course values(201, 'Data Science'), (201, 'Neural Net'),
                                    (207, 'Biology'),
                                    (204, 'Full Stacks'),
                                    (210, 'Economics'), (210, 'Socio-Psycho'),
                                    (213, 'Socio-Psycho'), (213, 'Psychology');
select * from course;


select C.Dept, C.Course, D.Dept_Name as [Department Name] from course as C INNER
JOIN dept as D ON C.Dept = D.Dept_Id;


-- Count of Courses offered by various Dept SUBQUERY
select D.Dept_Name as Department, D.Dept_Id, C.COUNT as 'COUNT' from dept as D
INNER JOIN
   (select Dept, COUNT(Dept) as 'COUNT' from course group by Dept) as C on
D.Dept_Id = C.Dept;

select D.Dept_Name as Department, D.Dept_Id from dept as D INNER JOIN
   (select Dept, COUNT(Dept) as 'COUNT' from course group by Dept) as C on
D.Dept_Id = C.Dept
   where C.COUNT >= 2;


-- GRANT DCL Command
create login tt_login with password = '1234';
GO

create user tt_user for login tt_login;
GO

GRANT select on dbo.course to tt_user;
GO
```

5. **Output:**

Easy-Level

Author Table

	Author Id	Author Name	Country
1	120	Ruskin Bond	India
2	130	Robert Greene	USA
3	145	Dale Carnegie	USA
4	250	Robert Frost	USA

Books Table

	ID	Name	Author Id
1	200	Influence	145
2	202	Room on the Roof	120
3	204	Blue Umbrella	120
4	206	Human Nature	130

INNER JOIN

	Book Name	Author Name	Country
1	Influence	Dale Carnegie	USA
2	Room on the Roof	Ruskin Bond	India
3	Blue Umbrella	Ruskin Bond	India
4	Human Nature	Robert Greene	USA

Medium-Level

Department Table

	Dept_Id	Dept_Name
1	201	AI&ML
2	204	CSE
3	207	Bio-Tech
4	210	Finance
5	213	Psychology

Course Table

	Dept	Course
1	201	Data Science
2	201	Neural Net
3	207	Biology
4	204	Full Stacks
5	210	Economics
6	210	Socio-Psycho
7	213	Socio-Psycho
8	213	Psychology

Subquery to Count

	Department	Dept_Id	COUNT
1	AI&ML	201	2
2	CSE	204	1
3	Bio-Tech	207	1
4	Finance	210	2
5	Psychology	213	2

Department with more than 2 courses

	Department	Dept_Id
1	AI&ML	201
2	Finance	210
3	Psychology	213

6. **Learning Outcomes:**
    - Learn't about SQL Basic Operations.
    - Learn't about various types of JOINS such as FULL JOIN, INNER JOIN, LEFT & RIGHT JOIN.
    - Learn't about foreign key and its implementation in actual scenario.
    - Learn't how to perform subquery and implement filter along with subquery.
    - Learn't about basic TCP command such as GRANT.