Experiment – 1

Student Name: KJ Pritpal Singh   **UID: 23BCS14164**
Branch: BE-CSE   **Section/Group: KRG_1A**
Semester: 5th   **Date of Performance: 17-07-25**
Subject Name: ADBMS   **Subject Code: 23CSP-333**

1. Aim:

Q1) 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.
- Expected Output: Each book title along with its author's name.

Q2) 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
- Display only departments that offer more than 2 courses.
.

2. Requirements (Hardware/Software):
MySQL, PostgreSQL, Oracle, or SQL Server

3. DBMS Script & Output:

[EASY LEVEL]

CREATE DATABASE ADBMS_2027;

--USE ADBMS_2027;

```sql
CREATE TABLE TBL_AUTHOR(AUTHOR_ID INT PRIMARY KEY,
 AUTHOR_NAME VARCHAR(30));
CREATE TABLE TBL_BOOK(BOOK_ID INT PRIMARY KEY,
BOOK_TITLE VARCHAR(30),
AUTHOR_ID INT,
FOREIGN KEY (AUTHOR_ID) REFERENCES TBL_AUTHOR(AU-
    THOR_ID));

INSERT INTO TBL_AUTHOR (AUTHOR_ID, AUTHOR_NAME) VALUES
(1, 'C.J. Date'),
(2, 'Silberschatz'),
(3, 'A. Tanenbaum');

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

SELECT * FROM TBL_BOOK;
SELECT * FROM TBL_AUTHOR;

SELECT B.BOOK_TITLE , A.AUTHOR_NAME
FROM TBL_BOOK AS B
INNER JOIN
TBL_AUTHOR AS A
ON
B.AUTHOR_ID = A.AUTHOR_ID;
```

[MEDIUM LEVEL]

```sql
CREATE TABLE TBL_DEPARTMENT (DEPT_ID INT PRIMARY KEY,
    DEPT_NAME VARCHAR(30));

CREATE TABLE TBL_COURSE(COURSE_ID INT PRIMARY KEY,
    COURSE_NAME VARCHAR(30), DEPT_ID INT, FOREIGN KEY
    (DEPT_ID) REFERENCES TBL_DEPARTMENT(DEPT_ID));


INSERT INTO TBL_DEPARTMENT (DEPT_ID, DEPT_NAME) VALUES
(1, 'Computer Science'),
(2, 'Electrical Engineering'),
(3, 'Mechanical Engineering'),
(4, 'Civil Engineering'),
(5, 'Mathematics');

INSERT INTO TBL_COURSE (COURSE_ID, COURSE_NAME, DEPT_ID)
    VALUES
(101, 'Data Structures', 1),
(102, 'Algorithms', 1),
(103, 'Operating Systems', 1),
(104, 'Circuits', 2),
(105, 'Digital Logic', 2),
(106, 'Thermodynamics', 3),
(107, 'Fluid Mechanics', 3),
(108, 'Surveying', 4),
(109, 'Calculus', 5),
(110, 'Linear Algebra', 5),
(111, 'Discrete Math', 5);


SELECT DEPT_NAME
FROM TBL_DEPARTMENT
WHERE DEPT_ID IN (
    SELECT DEPT_ID
```

```
        FROM TBL_COURSE
        GROUP BY DEPT_ID
        HAVING COUNT(*) > 2
    );
```

4. **Output:**

[EASY LEVEL]

	BOOK_TITLE	AUTHOR_NAME
1	Database Systems	C.J. Date
2	Operating Systems	Silberschatz
3	Computer Netwo...	A. Tanenbaum
4	Advanced Datab...	C.J. Date
5	Modem OS	Silberschatz

[MEDIUM LEVEL

	DEPT_NAME
1	Computer Science
2	Mathematics

5. **LEARNING OUTCOMES:**

- Gained practical experience in table creation and record manipulation in databases.
- Learned how to join tables and extract relevant values efficiently.
- Understood implementation of INNER JOINs to integrate and analyze related data.