University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 1

NAME : ADITI ARYA UID : 23BCS11535

BRANCH : BE-CSE SECTION/GROUP : KRG_1A

SEMESTER : 5TH SUBJECT CODE : 23CSP-339

SUBJECT NAME : ADBMS

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.

[MEDIUM] Department-Course Subquery and Access Control.

1. Design normalized tables for departments and the courses they o er, 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 o er more than two courses.
5. Grant SELECT-only access on the courses table to a specific user.

2. Tools Used : SQL Server Management Studio

3. Code :

```
---------- EASY ----------

CREATE TABLE TBL_AUTHOR_DETAILS(
 AUTHOR_ID INT PRIMARY KEY,
 AUTHOR_NAME VARCHAR(50),
 COUNTRY VARCHAR(50)
```

```
);

CREATE TABLE TBL_BOOK_DETAILS(
 BOOK_ID INT PRIMARY KEY,
 BOOK_TITLE VARCHAR(MAX),
 AUTHORID INT
 FOREIGN KEY (AUTHORID) REFERENCES TBL_AUTHOR_DETAILS(AUTHOR_ID)
);

INSERT INTO TBL_AUTHOR_DETAILS VALUES (1,'AMAN','INDIA');
INSERT INTO TBL_AUTHOR_DETAILS VALUES (2,'MARK','USA');
INSERT INTO TBL_AUTHOR_DETAILS VALUES (3,'KANG','CHINA');
SELECT * FROM TBL_AUTHOR_DETAILS;



INSERT INTO TBL_BOOK_DETAILS VALUES (1,'JAVA HANDS ON',1);
INSERT INTO TBL_BOOK_DETAILS VALUES (2,'FB MARKETPLACE',2);
INSERT INTO TBL_BOOK_DETAILS VALUES (3,'MOON DANCE',3);
SELECT * FROM TBL_BOOK_DETAILS;



SELECT BD.BOOK_TITLE, AD.AUTHOR_NAME, AD.COUNTRY
FROM
TBL_AUTHOR_DETAILS AS AD
INNER JOIN
TBL_BOOK_DETAILS AS BD
ON
AD.AUTHOR_ID = BD.AUTHORID ;

---------- MEDIUM ----------

CREATE TABLE TBL_DEPARTMENTS (
DEPT_ID INT PRIMARY KEY,
DEPT_NAME VARCHAR(100) NOT NULL
);

CREATE TABLE TBL_COURSES (
COURSE_ID INT PRIMARY KEY,
COURSE_NAME VARCHAR(150) NOT NULL,
DEPT_ID INT,
FOREIGN KEY (DEPT_ID) REFERENCES TBL_DEPARTMENTS(DEPT_ID)
);

INSERT INTO TBL_DEPARTMENTS VALUES
(1, 'COMPUTER SCIENCE'),
(2, 'MATHEMATICS'),
(3, 'PHYSICS'),
(4, 'CHEMISTRY'),
(5, 'BIOLOGY');

SELECT * FROM TBL_DEPARTMENTS;

INSERT INTO TBL_COURSES VALUES
(101, 'Data Structures', 1),
```

```
(102, 'Operating Systems', 1),
(103, 'Algorithms', 1),
(104, 'Calculus I', 2),
(105, 'Linear Algebra', 2),
(106, 'Quantum Mechanics', 3),
(107, 'Classical Mechanics', 3),
(108, 'Modern Poetry', 4),
(109, 'Cell Biology', 5),
(110, 'Genetics', 5);

SELECT * FROM TBL_COURSES;


SELECT DEPT_NAME
FROM TBL_DEPARTMENTS
WHERE DEPT_ID IN (
SELECT DEPT_ID
FROM TBL_COURSES
GROUP BY DEPT_ID
HAVING COUNT(COURSE_ID) > 2
    );
```
4.

Output :

[EASY]

	AUTHOR_ID	AUTHOR_NAME	COUNTRY
1	1	AMAN	INDIA
2	2	MARK	USA
3	3	KANG	CHINA

	BOOK_ID	BOOK_TITLE	AUTHORID
1	1	JAVA HANDS ON	1
2	2	FB MARKETPLACE	2
3	3	MOON DANCE	3

	BOOK_TITLE	AUTHOR_NAME	COUNTRY
1	JAVA HANDS ON	AMAN	INDIA
2	FB MARKETPLACE	MARK	USA
3	MOON DANCE	KANG	CHINA

[MEDIUM]

	DEPT_ID	DEPT_NAME
1	1	COMPUTER SCIENCE
2	2	MATHEMATICS
3	3	PHYSICS
4	4	CHEMISTRY
5	5	BIOLOGY

	COURSE_ID	COURSE_NAME	DEPT_ID
1	101	Data Structures	1
2	102	Operating Systems	1
3	103	Algorithms	1
4	104	Calculus I	2
5	105	Linear Algebra	2
6	106	Quantum Mechanics	3
7	107	Classical Mechanics	3
8	108	Modern Poetry	4
9	109	Cell Biology	5
10	110	Genetics	5

	DEPT_NAME
1	COMPUTER SCIENCE

5. 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.