



Experiment 1

Name: Bhuvan Dhiman
Branch: BE-CSE
Semester: 5th Semester
Subject Code: 23CSP-333

UID: 23BCS14067
Section/Group: KRG_3A
Subject: ADBMS

1. AIM:

University Database System helps in managing student enrollments, course allocations, and professor assignments effectively. The system also demonstrates secure access control and transaction safety. This includes CRUD operations, JOIN queries, and database-level user permission management.

A. Easy-Level Problem

Problem Title: Author-Book Relationship Using Joins and Basic SQL Operations

Procedure (Step-by-Step):

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.

B. Medium-Level Problem

Problem Title: Department-Course Subquery and Access Control

Procedure (Step-by-Step):

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.

2. DBMS CODE:

Solution A:

```
CREATE TABLE AUTHOR(  
  AUTHOR_ID INT PRIMARY KEY,  
  AUTHOR_NAME VARCHAR(20),  
  COUNTRY VARCHAR(20)  
)  
CREATE TABLE BOOK (
```

```

BOOK_ID INT PRIMARY KEY,
BOOK_TITLE VARCHAR(20),
AUTHOR_ID INT
FOREIGN KEY (AUTHOR_ID)
REFERENCES AUTHOR(AUTHOR_ID)
)
INSERT INTO AUTHOR
VALUES
(1,'ANDREZ', 'POLAND'),
(2,'MARTIN', 'USA'),
(3,'AMAN', 'INDIA'),
(4,'CARL', 'AUSTRALIA');

```

```

INSERT INTO BOOK
VALUES
(101,'SONG OF ICE AND FIRE', 2),
(102,'THE WITCHER', 1),
(103,'MONEY', 3),
(104,'SWORD', 4);

```

```

SELECT A.AUTHOR_NAME, A.COUNTRY , B.BOOK_TITLE
FROM AUTHOR AS A
INNER JOIN
BOOK AS B
ON
A.AUTHOR_ID = B.AUTHOR_ID

```

Output:

100 %

No issues found

Results

Messages

	AUTHOR_NAME	COUNTRY	BOOK_TITLE
1	MARTIN	USA	SONG OF ICE AND FIRE
2	ANDREZ	POLAND'	THE WITCHER
3	AMAN	INDIA	MONEY
4	CARL	AUSTRALIA	SWORD

Solution B:

```
CREATE TABLE Department (  
    DeptID INT PRIMARY KEY,  
    DeptName VARCHAR(100)  
);
```

-- Create Course Table

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

-- Insert Departments

```
INSERT INTO Department VALUES  
(1, 'Computer Science'),  
(2, 'Physics'),  
(3, 'Mathematics'),  
(4, 'Chemistry'),  
(5, 'Biology');
```

-- Insert Course

```
INSERT INTO Course VALUES  
(101, 'Data Structures', 1),  
(102, 'Operating Systems', 1),  
(103, 'Quantum Mechanics', 3),  
(104, 'Electromagnetism', 2),  
(105, 'Linear Algebra', 3),  
(106, 'Calculus', 3),  
(107, 'Organic Chemistry', 4),  
(108, 'Physical Chemistry', 4),  
(109, 'Genetics', 5),  
(110, 'Molecular Biology', 5);
```

```
SELECT DeptName  
FROM Department  
WHERE DeptID IN (
```

```
SELECT DeptID
```

```
FROM COURSE
GROUP BY DeptID
HAVING COUNT(*) > 2
)
```

```
CREATE LOGIN BHUVAN
WITH PASSWORD = 'Bhuvan';
USE EMPLOYEES;
CREATE USER Bhuvan FOR LOGIN BHUVAN;
GRANT SELECT ON Course TO BHUVAN;
```

Output:

100 %	✓ No issues found
Results	Messages
	DeptName
1	Mathematics