Experiment 1

Student Name: Bhaskar Kumar **UID: 23BCS14337**
Branch: BE-CSE **Section/Group: KRG-3B**
Semester: 5 **Date of performance: 25-07-2025**
Subject name: ADBMS **Subject code: 23CSP-333**

Problem Title: 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.

Sample Output Description:

When the join is performed, we get a list where each book title is shown along with its author's name and their country.

Code:

```
CREATE DATABASE KRG3;

USE KRG3;


CREATE TABLE Author (
    AuthorID INT PRIMARY KEY,
    AuthorName VARCHAR(100),
    Country VARCHAR(100)
);


CREATE TABLE Book (
```

```
    BookID INT PRIMARY KEY,
    Title VARCHAR(150),
    AuthorID INT,
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID)
);

INSERT INTO Author (AuthorID, AuthorName, Country)
VALUES
(1, 'Chetan Bhagat', 'India'),
(2, 'Arundhati Roy', 'India'),
(3, 'R. K. Narayan', 'India'),
(4, 'J.K. Rowling', 'United Kingdom'),
(5, 'George R.R. Martin', 'United States'),
(6, 'Haruki Murakami', 'Japan'),
(7, 'Paulo Coelho', 'Brazil'),
(8, 'Albert Camus', 'France');

INSERT INTO Book (BookID, Title, AuthorID)
VALUES
(101, 'Five Point Someone', 1),
(102, 'The 3 Mistakes of My Life', 1),
(103, 'The God of Small Things', 2),
(104, 'Swami and Friends', 3),
(105, 'The Guide', 3),
(106, 'Harry Potter and the Philosopher''s Stone', 4),
(107, 'A Game of Thrones', 5),
(108, 'Kafka on the Shore', 6),
```

```
(109, 'The Alchemist', 7),
(110, 'The Stranger', 8);

SELECT
    B.Title AS BookTitle,
    A.AuthorName,
    A.Country
FROM
    Book B
INNER JOIN
    Author A ON B.AuthorID = A.AuthorID;
```

OUTPUT:

100 % 1 0

Results Messages

	BookTitle	AuthorName	Country
1	Five Point Someone	Chetan Bhagat	India
2	The 3 Mistakes of My Life	Chetan Bhagat	India
3	The God of Small Things	Arundhati Roy	India
4	Swami and Friends	R. K. Narayan	India
5	The Guide	R. K. Narayan	India
6	Harry Potter and the Philosopher's Stone	J.K. Rowling	United Kingdom
7	A Game of Thrones	George R.R. Martin	United States
8	Kafka on the Shore	Haruki Murakami	Japan
9	The Alchemist	Paulo Coelho	Brazil
10	The Stranger	Albert Camus	France

Problem Title: 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.

Sample Output Description:

The result shows the names of departments which are associated with more than two courses in the system.

```
-- EXPERIMENT 1(B)
USE KRG3

CREATE TABLE Employee_tbl (
   EmpId INT PRIMARY KEY,
   EmpName VARCHAR(100),
   Designation VARCHAR(100),
   Salary INT
);

CREATE TABLE department (
   DeptId INT PRIMARY KEY,
   DeptName VARCHAR(100),
   EmpId INT,
   FOREIGN KEY (EmpId) REFERENCES Employee_tbl(EmpId)
);
```

```
INSERT INTO Employee_tbl (EmpId, EmpName, Designation, Salary) VALUES
(1, 'Tanmay', 'Backend Developer', 20000000),
(2, 'Neha', 'Business Analyst', 61000),
(3, 'Karan', 'Graphic Designer', 53000),
(4, 'Priya', 'HR Executive', 64000);

INSERT INTO department (DeptId, DeptName, EmpId) VALUES
(201, 'Development', 1),
(202, 'Business Intelligence', 2),
(203, 'Creative', 3),
(204, 'Human Resources', 4);

SELECT * FROM Employee_tbl;
SELECT * FROM department;

SELECT Employee_tbl.*, department.*
FROM Employee_tbl
INNER JOIN department
ON Employee_tbl.EmpId = department.EmpId;
```

```
46  SELECT
```

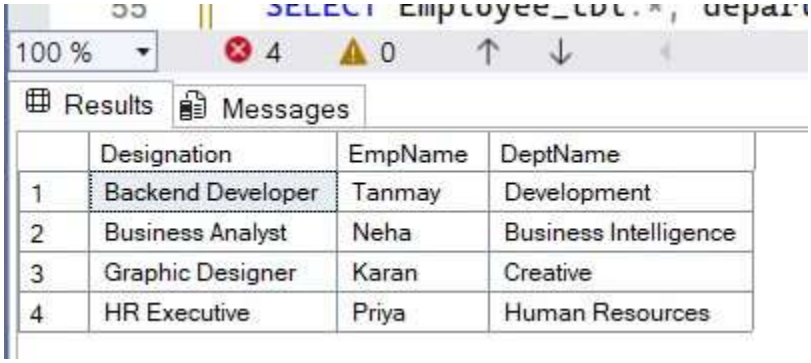
100 % 4 0

Results Messages

	EmpId	EmpName	Designation	Salary	DeptId	DeptName	EmpId
1	1	Tanmay	Backend Developer	20000000	201	Development	1
2	2	Neha	Business Analyst	61000	202	Business Intelligence	2
3	3	Karan	Graphic Designer	53000	203	Creative	3
4	4	Priya	HR Executive	64000	204	Human Resources	4

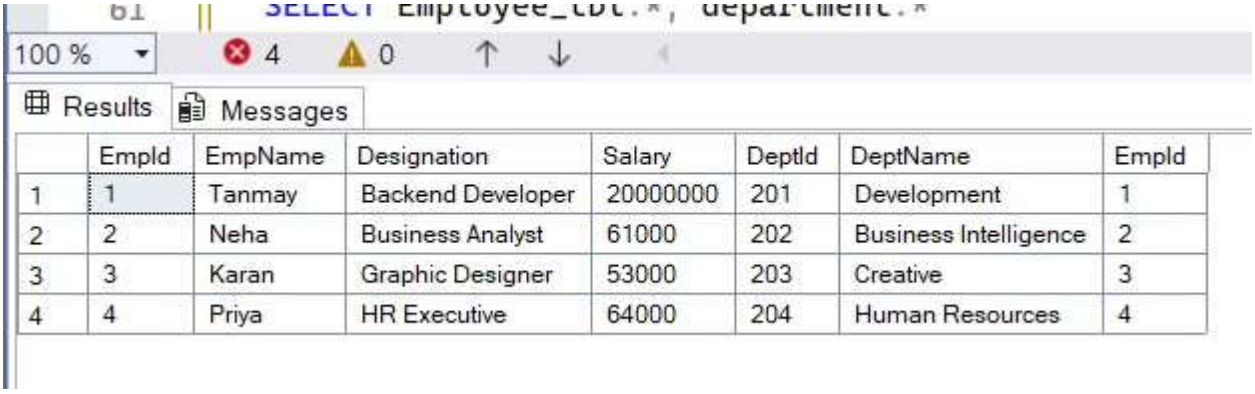
```
SELECT
  E.Designation,
```

```
 E.EmpName,
  D.DeptName
FROM Employee_tbl AS E
INNER JOIN department AS D
ON E.EmpId = D.EmpId;
```



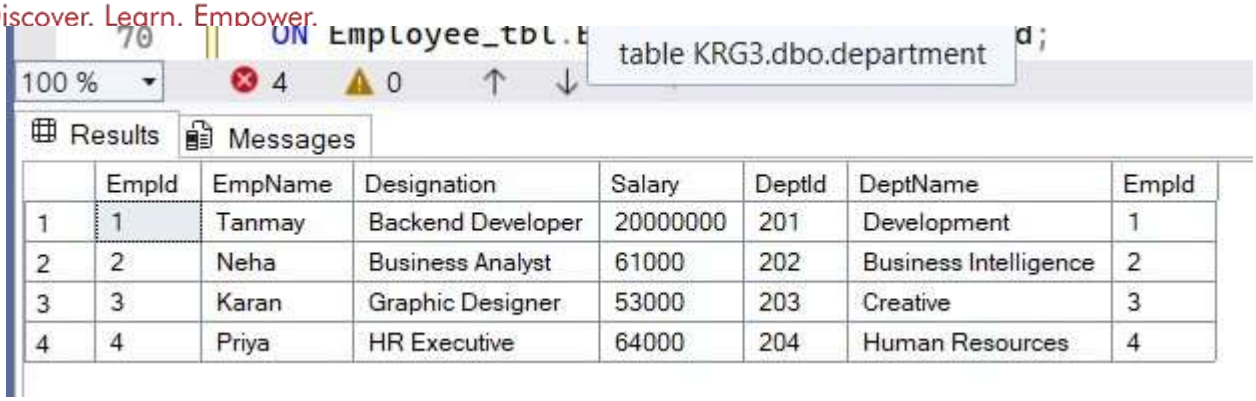
	Designation	EmpName	DeptName
1	Backend Developer	Tanmay	Development
2	Business Analyst	Neha	Business Intelligence
3	Graphic Designer	Karan	Creative
4	HR Executive	Priya	Human Resources

```
SELECT Employee_tbl.*, department.*
FROM Employee_tbl
LEFT OUTER JOIN department
ON Employee_tbl.EmpId = department.EmpId;
```



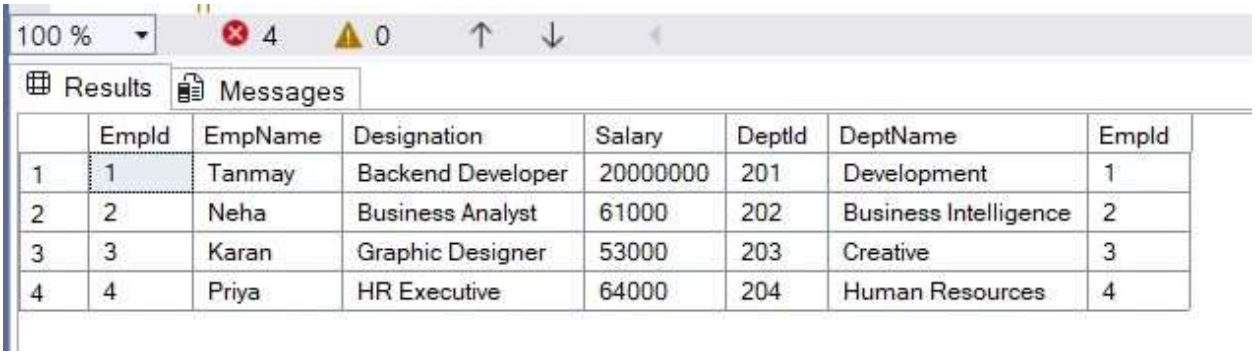
	EmpId	EmpName	Designation	Salary	DeptId	DeptName	EmpId
1	1	Tanmay	Backend Developer	20000000	201	Development	1
2	2	Neha	Business Analyst	61000	202	Business Intelligence	2
3	3	Karan	Graphic Designer	53000	203	Creative	3
4	4	Priya	HR Executive	64000	204	Human Resources	4

```
SELECT Employee_tbl.*, department.*
FROM Employee_tbl
RIGHT OUTER JOIN department
ON Employee_tbl.EmpId = department.EmpId;
```



	EmpId	EmpName	Designation	Salary	DeptId	DeptName	EmpId
1	1	Tanmay	Backend Developer	20000000	201	Development	1
2	2	Neha	Business Analyst	61000	202	Business Intelligence	2
3	3	Karan	Graphic Designer	53000	203	Creative	3
4	4	Priya	HR Executive	64000	204	Human Resources	4

```
SELECT Employee_tbl.*, department.*
FROM Employee_tbl
FULL OUTER JOIN department
ON Employee_tbl.EmpId = department.EmpId;
```



	EmpId	EmpName	Designation	Salary	DeptId	DeptName	EmpId
1	1	Tanmay	Backend Developer	20000000	201	Development	1
2	2	Neha	Business Analyst	61000	202	Business Intelligence	2
3	3	Karan	Graphic Designer	53000	203	Creative	3
4	4	Priya	HR Executive	64000	204	Human Resources	4

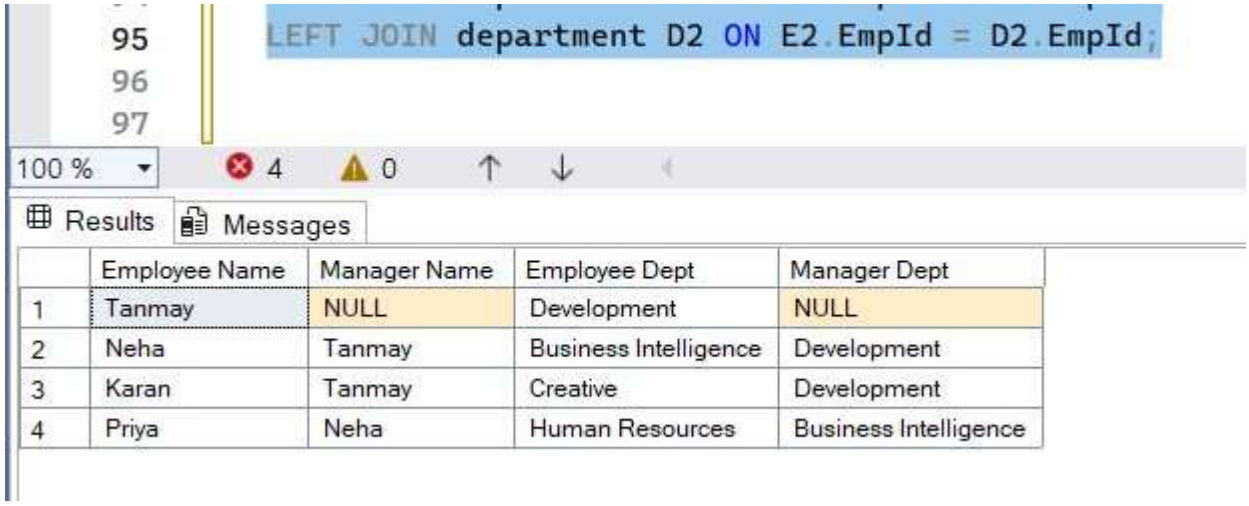
```
ALTER TABLE Employee_tbl
ADD ManagerId INT;

UPDATE Employee_tbl
SET ManagerId = NULL WHERE EmpId = 1;

UPDATE Employee_tbl
SET ManagerId = 1 WHERE EmpId IN (2, 3);

UPDATE Employee_tbl
SET ManagerId = 2 WHERE EmpId = 4;
```

```
SELECT
  E1.EmpName AS [Employee Name],
  E2.EmpName AS [Manager Name],
  D1.DeptName AS [Employee Dept],
  D2.DeptName AS [Manager Dept]
FROM Employee_tbl E1
LEFT JOIN Employee_tbl E2 ON E1.ManagerId = E2.EmpId
LEFT JOIN department D1 ON E1.EmpId = D1.EmpId
LEFT JOIN department D2 ON E2.EmpId = D2.EmpId;
```



	Employee Name	Manager Name	Employee Dept	Manager Dept
1	Tanmay	NULL	Development	NULL
2	Neha	Tanmay	Business Intelligence	Development
3	Karan	Tanmay	Creative	Development
4	Priya	Neha	Human Resources	Business Intelligence