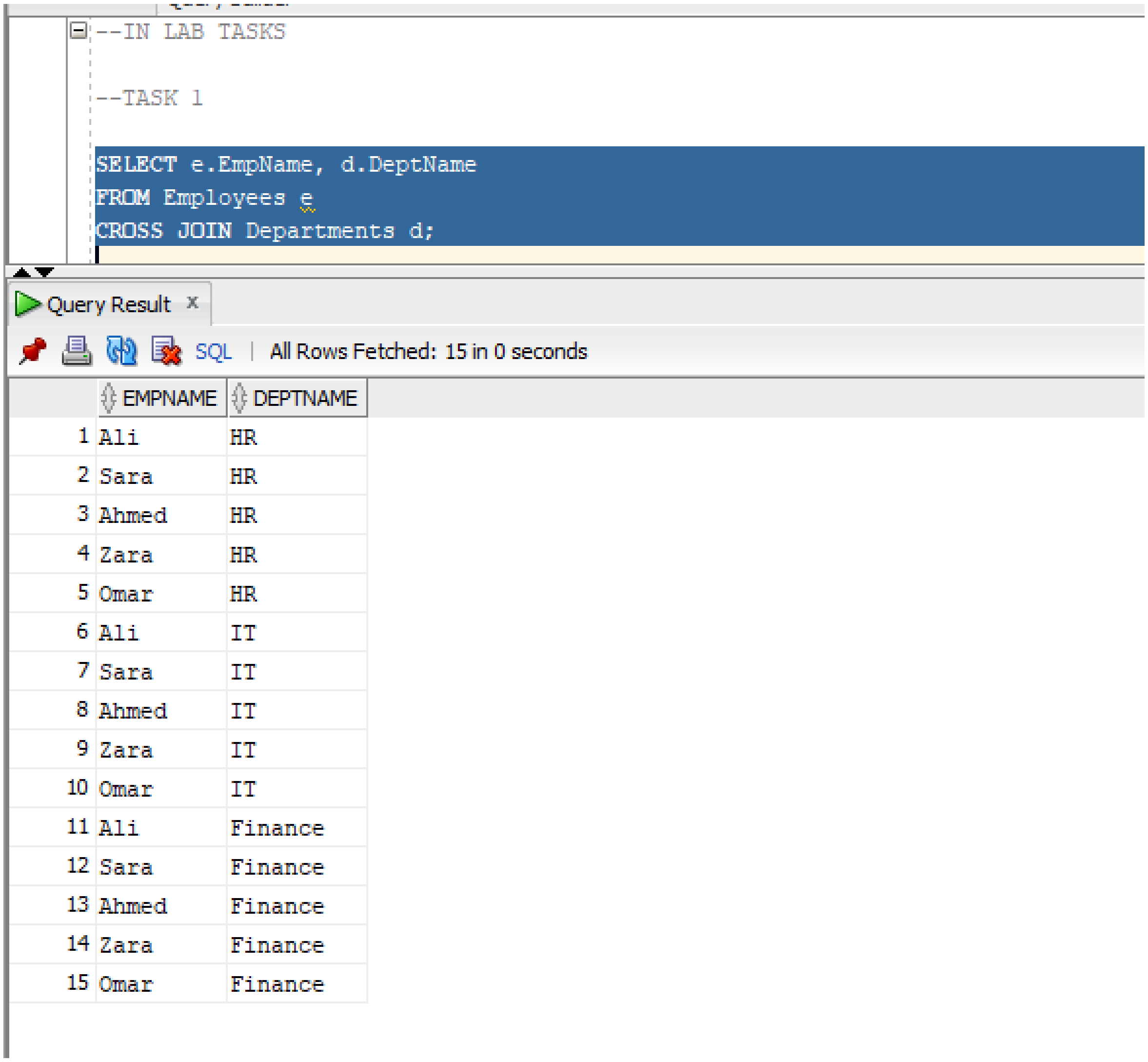
IN-LAB 5 TASKS DB

23K-0071

UROOJ BALOCH

TASK 1

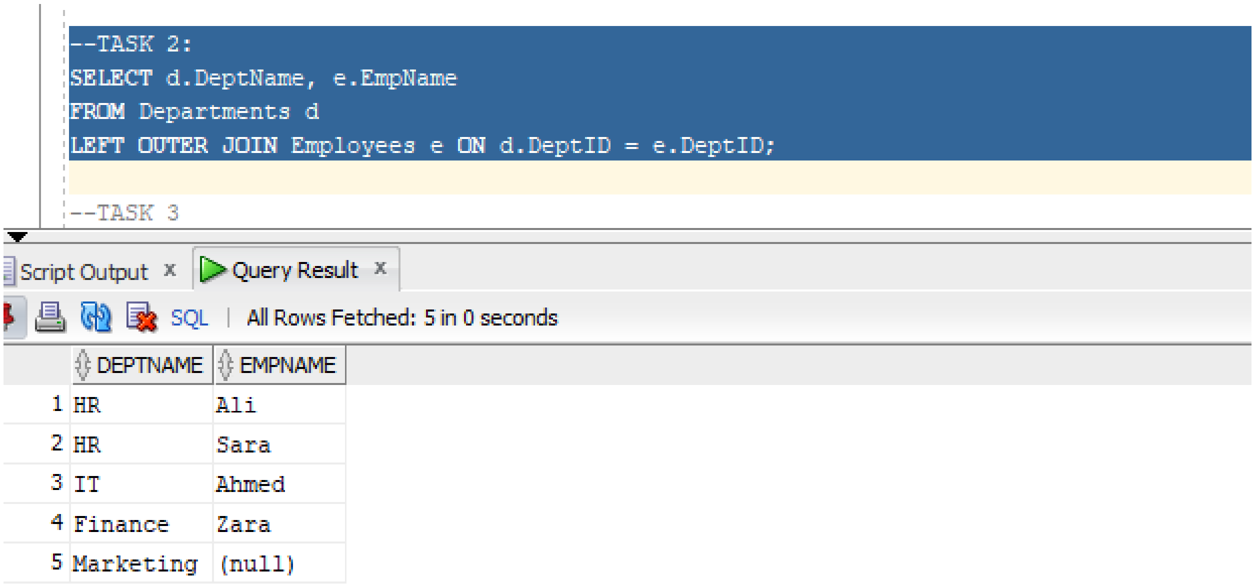


```
--IN LAB TASKS

--TASK 1

SELECT e.EmpName, d.DeptName
FROM Employees e
CROSS JOIN Departments d;
```

All Rows Fetched: 15 in 0 seconds

	EMPNAME	DEPTNAME
1	Ali	HR
2	Sara	HR
3	Ahmed	HR
4	Zara	HR
5	Omar	HR
6	Ali	IT
7	Sara	IT
8	Ahmed	IT
9	Zara	IT
10	Omar	IT
11	Ali	Finance
12	Sara	Finance
13	Ahmed	Finance
14	Zara	Finance
15	Omar	Finance

TASK 2:

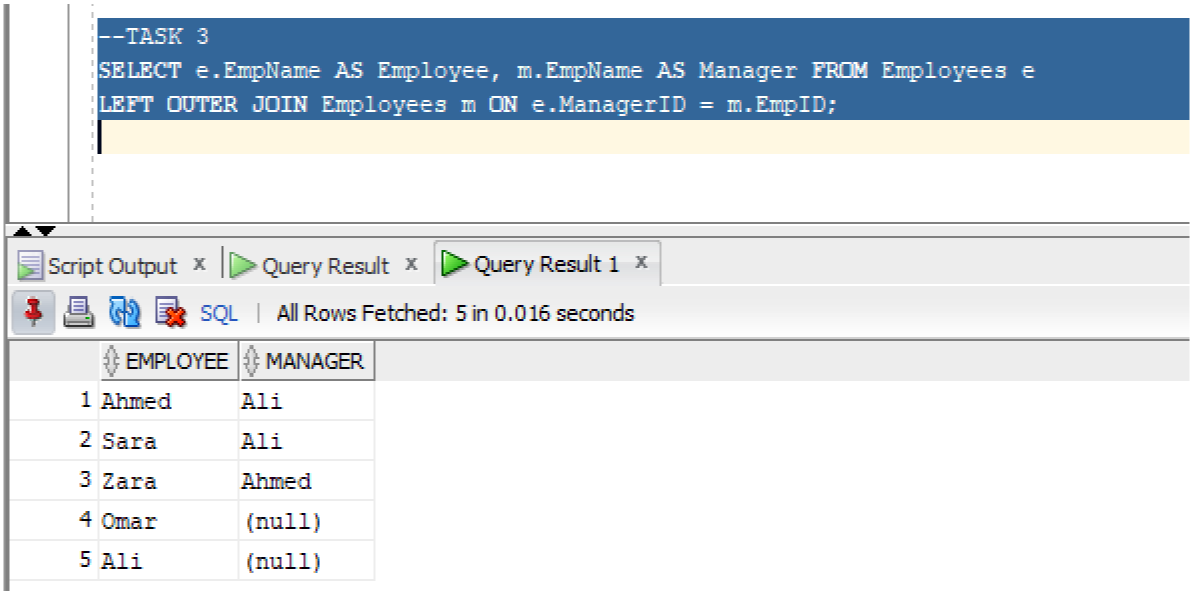


```
--TASK 2:
SELECT d.DeptName, e.EmpName
FROM Departments d
LEFT OUTER JOIN Employees e ON d.DeptID = e.DeptID;

--TASK 3
```

All Rows Fetched: 5 in 0 seconds

	DEPTNAME	EMPNAME
1	HR	Ali
2	HR	Sara
3	IT	Ahmed
4	Finance	Zara
5	Marketing	(null)

TASK 3:

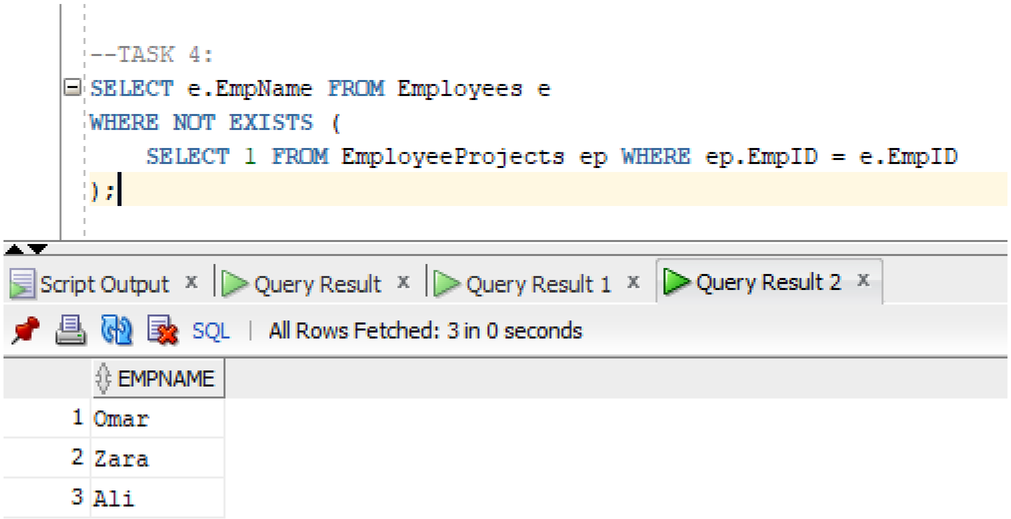


```
--TASK 3
SELECT e.EmpName AS Employee, m.EmpName AS Manager FROM Employees e
LEFT OUTER JOIN Employees m ON e.ManagerID = m.EmpID;
```

All Rows Fetched: 5 in 0.016 seconds

	EMPLOYEE	MANAGER
1	Ahmed	Ali
2	Sara	Ali
3	Zara	Ahmed
4	Omar	(null)
5	Ali	(null)

TASK 4:

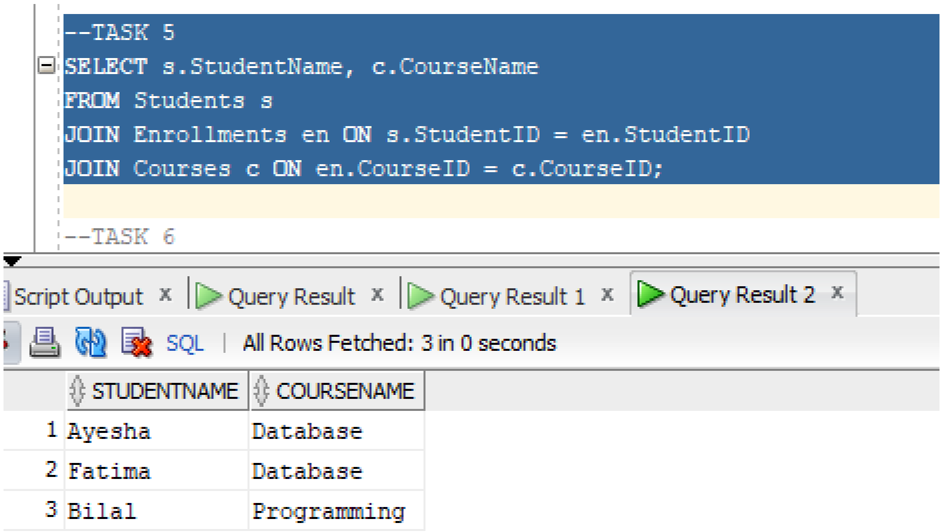


```
--TASK 4:
SELECT e.EmpName FROM Employees e
WHERE NOT EXISTS (
    SELECT 1 FROM EmployeeProjects ep WHERE ep.EmpID = e.EmpID
);
```

All Rows Fetched: 3 in 0 seconds

	EMPNAME
1	Omar
2	Zara
3	Ali

TASK 5:

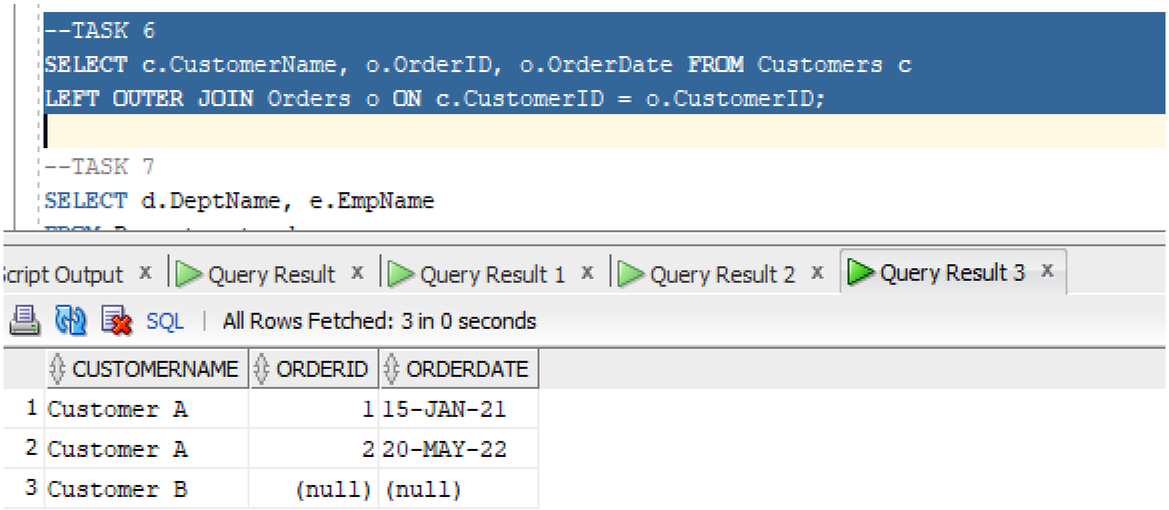


```
--TASK 5
SELECT s.StudentName, c.CourseName
FROM Students s
JOIN Enrollments en ON s.StudentID = en.StudentID
JOIN Courses c ON en.CourseID = c.CourseID;

--TASK 6
```

All Rows Fetched: 3 in 0 seconds

	STUDENTNAME	COURSENAME
1	Ayesha	Database
2	Fatima	Database
3	Bilal	Programming

TASK 6:

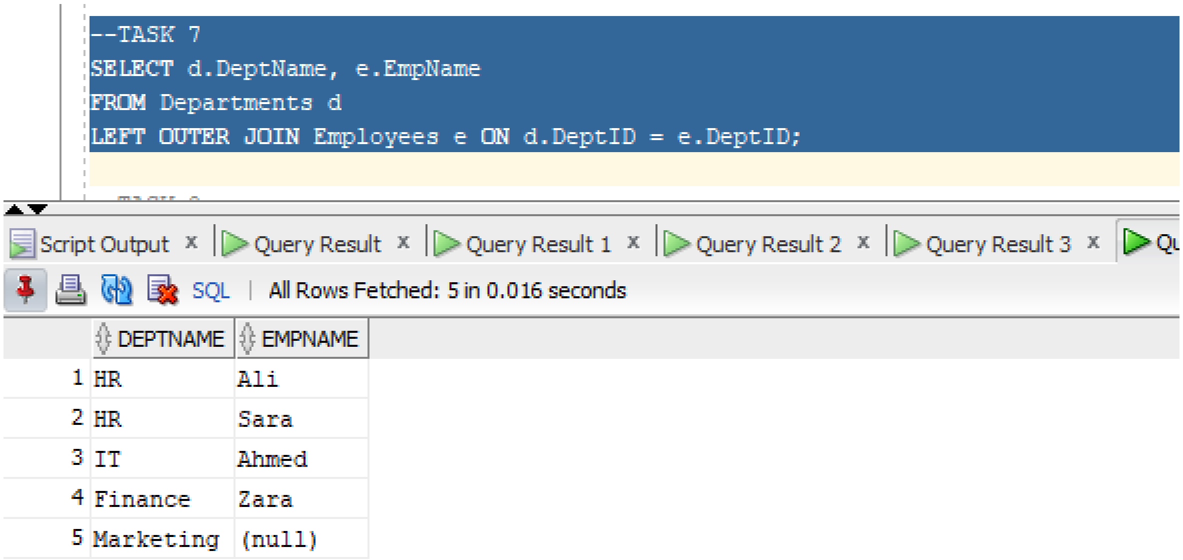


```
--TASK 6
SELECT c.CustomerName, o.OrderID, o.OrderDate FROM Customers c
LEFT OUTER JOIN Orders o ON c.CustomerID = o.CustomerID;

--TASK 7
SELECT d.DeptName, e.EmpName
```

All Rows Fetched: 3 in 0 seconds

	CUSTOMERNAME	ORDERID	ORDERDATE
1	Customer A	1	15-JAN-21
2	Customer A	2	20-MAY-22
3	Customer B	(null)	(null)

TASK 7:

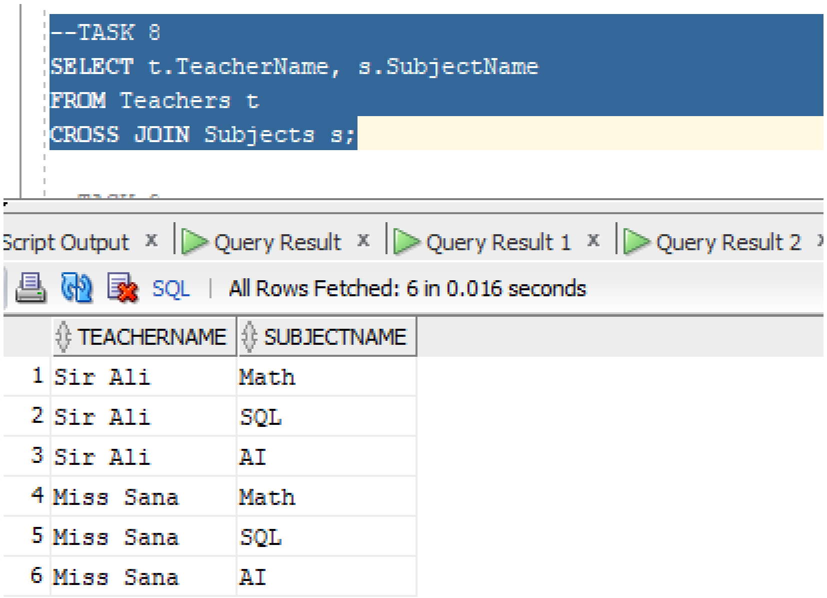


```
--TASK 7
SELECT d.DeptName, e.EmpName
FROM Departments d
LEFT OUTER JOIN Employees e ON d.DeptID = e.DeptID;
```

All Rows Fetched: 5 in 0.016 seconds

	DEPTNAME	EMPNAME
1	HR	Ali
2	HR	Sara
3	IT	Ahmed
4	Finance	Zara
5	Marketing	(null)

TASK 8:

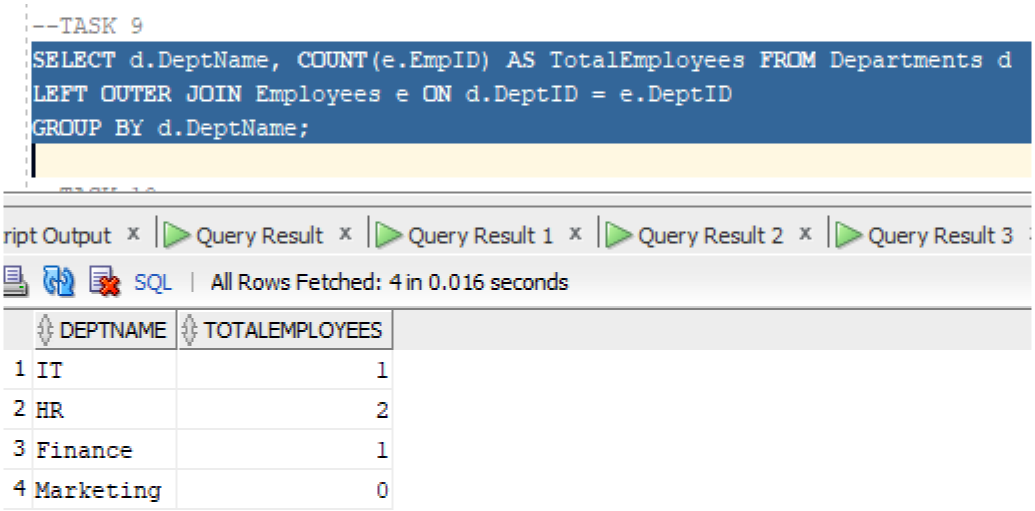


```
--TASK 8
SELECT t.TeacherName, s.SubjectName
FROM Teachers t
CROSS JOIN Subjects s;
```

All Rows Fetched: 6 in 0.016 seconds

	TEACHERNAME	SUBJECTNAME
1	Sir Ali	Math
2	Sir Ali	SQL
3	Sir Ali	AI
4	Miss Sana	Math
5	Miss Sana	SQL
6	Miss Sana	AI

TASK 9:

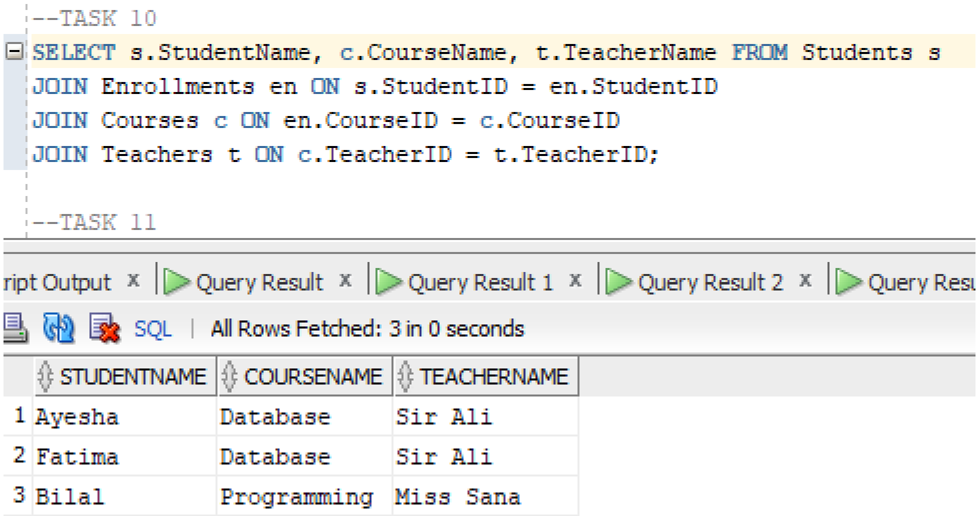


```
--TASK 9
SELECT d.DeptName, COUNT(e.EmpID) AS TotalEmployees FROM Departments d
LEFT OUTER JOIN Employees e ON d.DeptID = e.DeptID
GROUP BY d.DeptName;
```

All Rows Fetched: 4 in 0.016 seconds

	DEPTNAME	TOTALEMPLOYEES
1	IT	1
2	HR	2
3	Finance	1
4	Marketing	0

TASK 10:



```
--TASK 10
SELECT s.StudentName, c.CourseName, t.TeacherName FROM Students s
JOIN Enrollments en ON s.StudentID = en.StudentID
JOIN Courses c ON en.CourseID = c.CourseID
JOIN Teachers t ON c.TeacherID = t.TeacherID;

--TASK 11
```

All Rows Fetched: 3 in 0 seconds

	STUDENTNAME	COURSENAME	TEACHERNAME
1	Ayesha	Database	Sir Ali
2	Fatima	Database	Sir Ali
3	Bilal	Programming	Miss Sana

-- Departments

CREATE TABLE Departments (DeptID INT PRIMARY KEY, DeptName VARCHAR2(50));

-- Employees

CREATE TABLE Employees (EmpID INT PRIMARY KEY, EmpName VARCHAR2(50), DeptID INT, Salary NUMBER(10,2), ManagerID INT, HireDate DATE, FOREIGN KEY (DeptID) REFERENCES Departments(DeptID), FOREIGN KEY (ManagerID) REFERENCES Employees(EmpID));

-- Projects

CREATE TABLE Projects (ProjID INT PRIMARY KEY, ProjName VARCHAR2(50));

-- EmployeeProjects

CREATE TABLE EmployeeProjects (EmpID INT, ProjID INT, PRIMARY KEY (EmpID, ProjID), FOREIGN KEY (EmpID) REFERENCES Employees(EmpID), FOREIGN KEY (ProjID) REFERENCES Projects(ProjID));

-- Students

```
CREATE TABLE Students ( StudentID INT PRIMARY KEY, StudentName VARCHAR2(50), City VARCHAR2(50) );

-- Teachers

CREATE TABLE Teachers ( TeacherID INT PRIMARY KEY, TeacherName VARCHAR2(50), City VARCHAR2(50) );

-- Courses

 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR2(50), TeacherID INT, FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID) );

-- Enrollments

 CREATE TABLE Enrollments ( StudentID INT, CourseID INT, PRIMARY KEY (StudentID, CourseID), FOREIGN KEY (StudentID) REFERENCES Students(StudentID), FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) );

-- Subjects

CREATE TABLE Subjects ( SubjectID INT PRIMARY KEY, SubjectName VARCHAR2(50) );

-- TeacherSubjects

CREATE TABLE TeacherSubjects ( TeacherID INT, SubjectID INT, PRIMARY KEY (TeacherID, SubjectID), FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID), FOREIGN KEY (SubjectID) REFERENCES Subjects(SubjectID) );

-- Customers

 CREATE TABLE Customers ( CustomerID INT PRIMARY KEY, CustomerName VARCHAR2(50) );

-- Orders

CREATE TABLE Orders ( OrderID INT PRIMARY KEY, CustomerID INT, OrderDate DATE, FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) );

--INSERTING SAMPLE DATA TO PERFORM TASKS

– Departments
```

```
 INSERT INTO Departments VALUES (1, 'HR'); INSERT INTO Departments VALUES (2, 'IT');
INSERT INTO Departments VALUES (3, 'Finance'); INSERT INTO Departments VALUES (4,
'Marketing');

SELECT * FROM Departments;

-- Employees

 INSERT INTO Employees VALUES (1, 'Ali', 1, 60000, NULL, DATE '2020-05-10'); INSERT
INTO Employees VALUES (2, 'Sara', 1, 45000, 1, DATE '2021-03-15'); INSERT INTO
Employees VALUES (3, 'Ahmed', 2, 70000, 1, DATE '2019-07-20'); INSERT INTO Employees
VALUES (4, 'Zara', 3, 30000, 3, DATE '2022-01-01'); INSERT INTO Employees VALUES (5,
'Omar', NULL, 40000, NULL, DATE '2021-09-10'); SELECT * FROM Employees;

-- Projects

 INSERT INTO Projects VALUES (1, 'ERP System'); INSERT INTO Projects VALUES (2,
'Website'); INSERT INTO Projects VALUES (3, 'Audit'); SELECT * FROM Projects;

-- EmployeeProjects

INSERT INTO EmployeeProjects VALUES (2,1); INSERT INTO EmployeeProjects VALUES
(3,2); SELECT * FROM EmployeeProjects;

-- Students

 INSERT INTO Students VALUES (1, 'Ayesha', 'Lahore'); INSERT INTO Students VALUES (2,
'Bilal', 'Karachi'); INSERT INTO Students VALUES (3, 'Fatima', 'Lahore'); SELECT * FROM
Students;

-- Teachers

 INSERT INTO Teachers VALUES (1, 'Sir Ali', 'Lahore'); INSERT INTO Teachers VALUES (2,
'Miss Sana', 'Karachi'); SELECT * FROM Teachers;

-- Courses

INSERT INTO Courses VALUES (1, 'Database', 1); INSERT INTO Courses VALUES (2,
'Programming', 2); SELECT * FROM Courses;

-- Enrollments
```

```sql
 INSERT INTO Enrollments VALUES (1,1); INSERT INTO Enrollments VALUES (2,2); INSERT
INTO Enrollments VALUES (3,1); SELECT * FROM Enrollments;

-- Subjects

 INSERT INTO Subjects VALUES (1, 'Math'); INSERT INTO Subjects VALUES (2, 'SQL');
INSERT INTO Subjects VALUES (3, 'AI'); SELECT * FROM Subjects;

-- TeacherSubjects

INSERT INTO TeacherSubjects VALUES (1,1); INSERT INTO TeacherSubjects VALUES (1,2);
INSERT INTO TeacherSubjects VALUES (2,3); SELECT * FROM TeacherSubjects;

-- Customers

 INSERT INTO Customers VALUES (1, 'Customer A'); INSERT INTO Customers VALUES (2,
'Customer B'); SELECT * FROM Customers;

-- Orders

 INSERT INTO Orders VALUES (1,1, DATE '2021-01-15'); INSERT INTO Orders VALUES (2,1,
DATE '2022-05-20'); SELECT * FROM Orders;

--IN LAB TASKS

--TASK 1

SELECT e.EmpName, d.DeptName FROM Employees e CROSS JOIN Departments d;

--TASK 2:

 SELECT d.DeptName, e.EmpName FROM Departments d LEFT OUTER JOIN Employees e
ON d.DeptID = e.DeptID;

--TASK 3

SELECT e.EmpName AS Employee, m.EmpName AS Manager FROM Employees e LEFT
OUTER JOIN Employees m ON e.ManagerID = m.EmpID;

--TASK 4:

 SELECT e.EmpName FROM Employees e WHERE NOT EXISTS ( SELECT 1 FROM
EmployeeProjects ep WHERE ep.EmpID = e.EmpID );
```

```
--TASK 5

 SELECT s.StudentName, c.CourseName FROM Students s JOIN Enrollments en ON
s.StudentID = en.StudentID JOIN Courses c ON en.CourseID = c.CourseID;

--TASK 6

SELECT c.CustomerName, o.OrderID, o.OrderDate FROM Customers c LEFT OUTER JOIN
Orders o ON c.CustomerID = o.CustomerID;

--TASK 7

SELECT d.DeptName, e.EmpName FROM Departments d LEFT OUTER JOIN Employees e
ON d.DeptID = e.DeptID;

--TASK 8

SELECT t.TeacherName, s.SubjectName FROM Teachers t CROSS JOIN Subjects s;

--TASK 9

SELECT d.DeptName, COUNT(e.EmpID) AS TotalEmployees FROM Departments d LEFT
OUTER JOIN Employees e ON d.DeptID = e.DeptID GROUP BY d.DeptName;

--TASK 10

SELECT s.StudentName, c.CourseName, t.TeacherName FROM Students s JOIN
Enrollments en ON s.StudentID = en.StudentID JOIN Courses c ON en.CourseID =
c.CourseID JOIN Teachers t ON c.TeacherID = t.TeacherID;
```