Lab Exercise 3

Execute a Company database and apply various SQL queries.

Aim:

To Create a company database and execute the SQL queries.

Procedure:

Step 1: Go to database Home page.
Step 2: Create table with necessary entities.
Step 3:Insert Values in the existing table.
Step 4:Execute the various queries.
Step 5:Exit

Entity Relationship Diagram:

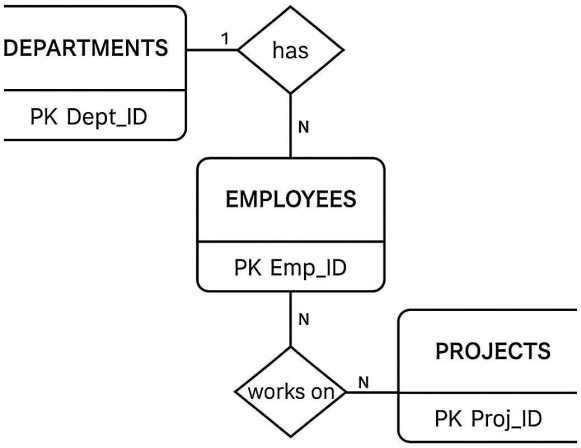


Table Creation:

```
create database company;
use company

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

```sql
CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    EmpName VARCHAR(100),
    DeptID INT,
    Salary DECIMAL(10, 2),
    JoiningDate DATE,
    Bonus DECIMAL(10, 2) NULL,          -- Bonus can be NULL (to demonstrate null handling)
    FOREIGN KEY (DeptID) REFERENCES Departments(DeptID));
```

```sql
CREATE TABLE Projects (
    ProjectID INT PRIMARY KEY,
    ProjectName VARCHAR(100));
```

```sql
CREATE TABLE EmployeeProjects (
    EmpID INT,
    ProjectID INT,
    PRIMARY KEY (EmpID, ProjectID),
    FOREIGN KEY (EmpID) REFERENCES Employees(EmpID),
    FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID));
```

Insert Values to the Table:

```sql
INSERT INTO Departments VALUES
(1, 'Development'),
(2, 'QA'),
(3, 'HR'),
(4, 'Management');
```

```sql
INSERT INTO Employees VALUES
(101, 'Alice Johnson', 1, 70000, '2020-01-15', 5000),
(102, 'Bob Smith', 1, 68000, '2019-03-10', NULL),
(103, 'Charlie Brown', 2, 60000, '2021-06-12', 3000),
(104, 'Diana Ross', 3, 55000, '2018-11-01', NULL),
(105, 'Evan Davis', 4, 90000, '2017-07-23', 7000);
```

```sql
INSERT INTO Projects VALUES
(201, 'Cloud Migration'),
(202, 'Mobile App'),
(203, 'AI Research');
```

```sql
INSERT INTO EmployeeProjects VALUES
(101, 201),
(102, 201),
(102, 202),
(103, 202),
(104, 203);
```

Output:

Query 1:

List distinct employee names working on 'Cloud Migration'

```sql
SELECT E.EmpName
FROM Employees E
JOIN EmployeeProjects EP ON E.EmpID = EP.EmpID
JOIN Projects P ON EP.ProjectID = P.ProjectID
WHERE P.ProjectName = 'Cloud Migration'
```

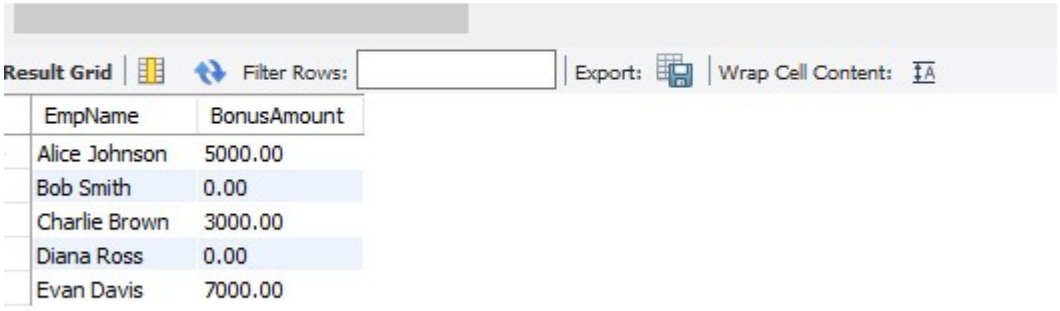
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmpName
Alice Johnson
Bob Smith

Query 2:

Show employee name and bonus; replace NULL bonus values with 0.

```
1 •  SELECT EmpName, COALESCE(Bonus, 0) AS BonusAmount
2    FROM Employees;
```



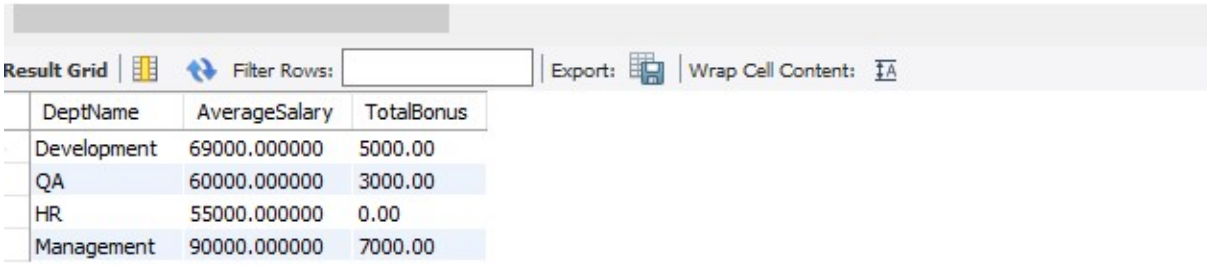
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmpName	BonusAmount
Alice Johnson	5000.00
Bob Smith	0.00
Charlie Brown	3000.00
Diana Ross	0.00
Evan Davis	7000.00

Query 3:

For each department, find the average salary and total bonus paid.

```
1 •  SELECT D.DeptName,
2           AVG(E.Salary) AS AverageSalary,
3           SUM(COALESCE(E.Bonus, 0)) AS TotalBonus
4    FROM Employees E
5    JOIN Departments D ON E.DeptID = D.DeptID
6    GROUP BY D.DeptName;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DeptName	AverageSalary	TotalBonus
Development	69000.000000	5000.00
QA	60000.000000	3000.00
HR	55000.000000	0.00
Management	90000.000000	7000.00

Query 4:

Find employees whose salary is greater than the average salary of their department.

```
SELECT EmpName, Salary, DeptID
FROM Employees E1
WHERE Salary > (
    SELECT AVG(Salary)
    FROM Employees E2
    WHERE E2.DeptID = E1.DeptID
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmpName	Salary	DeptID
Alice Johnson	70000.00	1

Query 5:

Retrieve the names of projects that have at least one employee working on them who has a bonus greater than 4000.

```
SELECT ProjectName
FROM Projects P
WHERE EXISTS (
    SELECT 1
    FROM EmployeeProjects EP
    JOIN Employees E ON EP.EmpID = E.EmpID
    WHERE EP.ProjectID = P.ProjectID
      AND E.Bonus > 4000
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProjectName
Cloud Migration

Result:

Thus the company database with SQL queries were executed successfully.