Answers

Write Create a Employee Department & Project Tables .Add atleast 5 records in each table.

CREATE TABLE Department (DeptID INT PRIMARY KEY, DeptName VARCHAR(50), Location VARCHAR(50));

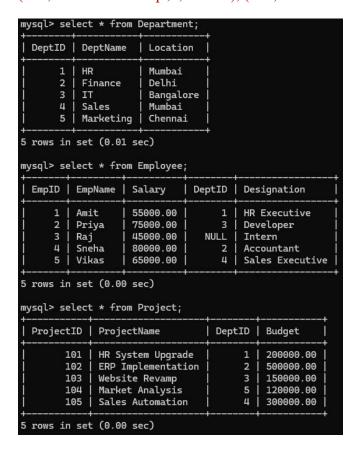
INSERT INTO Department VALUES (1,'HR','Mumbai'), (2,'Finance','Delhi'), (3,'IT','Bangalore'), (4,'Sales','Mumbai'), (5,'Marketing','Chennai');

CREATE TABLE Employee (EmpID INT PRIMARY KEY AUTO_INCREMENT, EmpName VARCHAR(50), Salary DECIMAL(10,2), DeptID INT, Designation VARCHAR(50), FOREIGN KEY (DeptID) REFERENCES Department(DeptID));

INSERT INTO Employee (EmpName,Salary,DeptID,Designation) VALUES ('Amit',55000,1,'HR Executive'), ('Priya',75000,3,'Developer'), ('Raj',45000,NULL,'Intern'), ('Sneha',80000,2,'Accountant'), ('Vikas',65000,4,'Sales Executive');

CREATE TABLE Project (ProjectID INT PRIMARY KEY, ProjectName VARCHAR(50), DeptID INT, Budget DECIMAL(12,2), FOREIGN KEY (DeptID) REFERENCES Department(DeptID));

INSERT INTO Project VALUES (101,'HR System Upgrade',1,200000), (102,'ERP Implementation',2,500000), (103,'Website Revamp',3,150000), (104,'Market Analysis',5,120000), (105,'Sales Automation',4,300000);



Write a following Queries.

1. Write a Relational Algebra expression to perform **Cartesian Product (Cross Join)** between Emp and Dept tables.

Employee × Department

2. Write a SQL query to perform an **INNER JOIN** between Emp and Dept tables displaying EmpName and DeptName.

SELECT E.EmpName, D.DeptName FROM Employee E INNER JOIN Department D ON E.DeptID=D.DeptID;

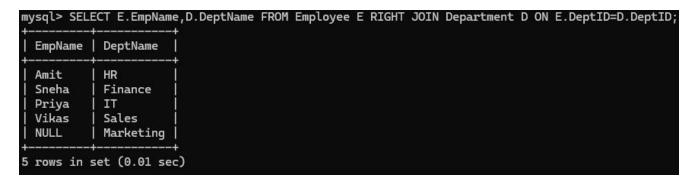
3. Write a SQL query to perform a **LEFT OUTER JOIN** between Emp and Dept tables to display all employees along with their department information, including employees without a department.

SELECT E.EmpName,D.DeptName,D.Location FROM Employee E LEFT JOIN Department D ON E.DeptID=D.DeptID;

EmpName	DeptName	Location			
Amit	HR	Mumbai			
Priya	IT	Bangalore			
Raj	NULL	NULL			
Sneha	Finance	Delhi			
Vikas	Sales	Mumbai			

4. Write a SQL query to perform a **RIGHT OUTER JOIN** between Emp and Dept tables.

SELECT E.EmpName, D.DeptName FROM Employee E RIGHT JOIN Department D ON E.DeptID=D.DeptID;



5. Write a SQL query to perform a NATURAL JOIN between Emp and Dept.

SELECT * FROM Employee NATURAL JOIN Department;

DeptID	EmpID	EmpName	Salary	Designation	DeptName	Location
1	1	Amit	55000.00	HR Executive	HR	Mumbai
3	2	Priya	75000.00	Developer	IT	Bangalore
2	4	Sneha	80000.00	Accountant	Finance	Delhi
4	5	Vikas	65000.00	Sales Executive	Sales	Mumbai

6. Write a SQL query to perform a **CROSS JOIN** between Emp and Project table.

SELECT E.EmpName, P. Project Name FROM Employee E CROSS JOIN Project P;

EmpName	ProjectName	į			
Vikas	HR System Upgrade	i			
Sneha	HR System Upgrade	1			
Raj	HR System Upgrade	į.			
Priya	HR System Upgrade	ľ			
Amit	HR System Upgrade	j)			
Vikas	ERP Implementation	Ī			
Sneha	ERP Implementation	İ			
Raj	ERP Implementation	i			
Priya	ERP Implementation	İ			
Amit	ERP Implementation	j .			
Vikas	Website Revamp	İ			
Sneha	Website Revamp	ř			
Raj	Website Revamp	j .			
Priya	Website Revamp	Ī			
Amit	Website Revamp	İ			
Vikas	Market Analysis	İ			
Sneha	Market Analysis	İ			
Raj	Market Analysis	j .			
Priya	Market Analysis	į.			
Amit	Market Analysis	ľ			
Vikas	Sales Automation	İ			
Sneha	Sales Automation	I			
Raj	Sales Automation	Ī			
Priya	Sales Automation				
Amit	Sales Automation				

7. Write a SQL query to create a new table EmpBackup with the same structure as Emp but no data.

CREATE TABLE EmpBackup AS SELECT * FROM Employee WHERE 1=0;

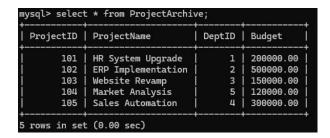
8. Write a SQL query to copy all data from Emp into EmpBackup table.

INSERT INTO EmpBackup SELECT * FROM Employee;

EmpID	EmpName	Salary	DeptID	Designation
1	Amit	55000.00	1	HR Executive
2	Priya	75000.00] 3	Developer
3	Raj	45000.00	NULL	Intern
4	Sneha	80000.00	2	Accountant
5	Vikas	65000.00	4	Sales Executive

9. Write a SQL query to create a new table ProjectArchive with the same structure and data as Project.

CREATE TABLE ProjectArchive AS SELECT * FROM Project;



10. Write a SQL query to create an **AUTO_INCREMENT sequence for EmpID in the Emp table** during table creation.

Already defined

11. Write a SQL query using a **subquery** to find all employees whose salary is greater than the **average salary of all employees**.

SELECT EmpName, Salary FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);

```
mysql> SELECT EmpName, Salary FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);
+-----+
| EmpName | Salary |
+-----+
| Priya | 75000.00 |
| Sneha | 80000.00 |
| Vikas | 65000.00 |
+-----+
3 rows in set (0.00 sec)
```

12. Write a SQL query using a **subquery in WHERE clause** to find employees working in departments located in 'Mumbai'.

SELECT EmpName FROM Employee WHERE DeptID IN (SELECT DeptID FROM Department WHERE Location='Mumbai');

13. Write a SQL query to display all departments where the number of employees is greater than 5, using a subquery.

SELECT DeptName FROM Department WHERE DeptID IN (SELECT DeptID FROM Employee GROUP BY DeptID HAVING COUNT(EmpID)>5);

mysql> SELECT DeptName FROM Department WHERE DeptID IN (SELECT DeptID FROM Employee GROUP BY DeptID HAVING COUNT(EmpID)>5); Empty set (0.00 sec)

14. Write a subqueries for find out Max salary of employee.

SELECT EmpName, Salary FROM Employee WHERE Salary=(SELECT MAX(Salary) FROM Employee);

```
mysql> SELECT EmpName, Salary FROM Employee WHERE Salary=(SELECT MAX(Salary) FROM Employee);
+-----+
| EmpName | Salary |
+----+
| Sneha | 80000.00 |
+----+
1 row in set (0.00 sec)
```

15. Write a self join query for each table.

Self join on Employee:

SELECT E1.EmpName AS Emp1,E2.EmpName AS Emp2,E1.DeptID FROM Employee E1 JOIN Employee E2 ON E1.DeptID=E2.DeptID WHERE E1.EmpID >> E2.EmpID;

mysql> SELECT E1.EmpName AS Emp1,E2.EmpName AS Emp2,E1.DeptID FROM Employee E1 JOIN Employee E2 ON E1.DeptID=E2.DeptID WHERE E1.EmpID<>E2.EmpID; Empty set (0.01 sec)

Self Join on Department:

SELECT D1.DeptName AS Dept1,D2.DeptName AS Dept2,D1.Location FROM Department D1 JOIN Department D2 ON D1.Location=D2.Location WHERE D1.DeptID<>D2.DeptID;

Self Join on Project:

SELECT P1.ProjectName AS Proj1,P2.ProjectName AS Proj2,P1.DeptID FROM Project P1 JOIN Project P2 ON P1.DeptID=P2.DeptID WHERE P1.ProjectID > P2.ProjectID;

mysql> SELECT P1.ProjectName AS Proj1,P2.ProjectName AS Proj2,P1.DeptID FROM Project P1 JOIN Project P2 ON P1.DeptID=P2.DeptID WHERE P1.ProjectID
Empty set (0.00 sec)