**Experiment No: 4**

**Date : 13/02/2025**

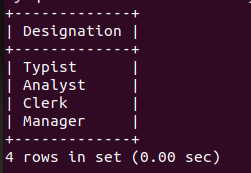
**Familiarization of Subquery, Joins, Views and Set Ooperations.**

**Consider the following Database Schema**

**Employee(ID character 5, DeptID numeric 2, Name character 15, Designation character 15, Basic numeric 10,2 , Gender character 1)**

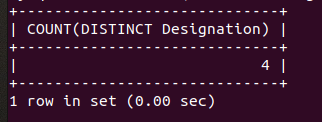
1. Display the different designations existing in the organisation.

>>SELECT DISTINCT Designation FROM EMPLOYEE;



2. Display the number of different designations existing in the organisation.

>>SELECT COUNT(DISTINCT Designation) FROM EMPLOYEE;



3. Display employee details with calculated DA, HRA, and NET\_SAL.

>>SELECT

ID,

Name,

Designation,

DeptID,

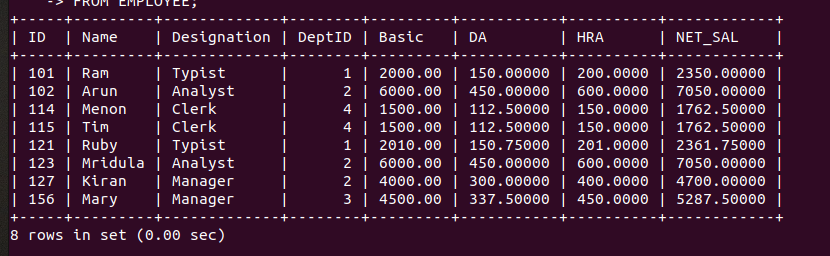
Basic,

(Basic \* 0.075) AS DA,

(Basic \* 0.10) AS HRA,

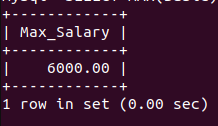
(Basic + (Basic \* 0.075) + (Basic \* 0.10)) AS NET\_SAL

FROM EMPLOYEE;



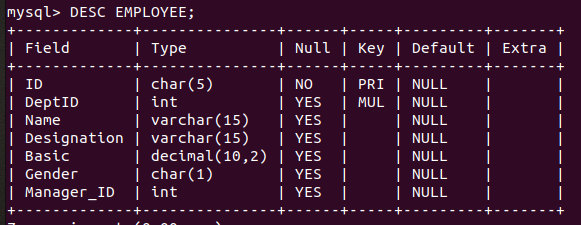
4. Display the maximum salary given for female employees.

>>SELECT MAX(Basic) AS MaxSalary FROM EMPLOYEE WHERE Gender = 'F';



5. Add a column 'ManagerID' into the Employee table.

>>ALTER TABLE EMPLOYEE ADD Manager\_ID INT;



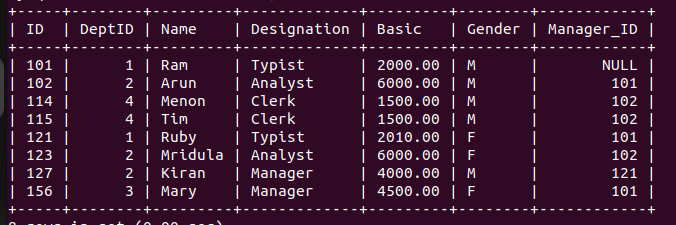
6. Update ManagerID values.

>>UPDATE EMPLOYEE SET Manager\_ID = NULL WHERE ID = 101;

>>UPDATE EMPLOYEE SET Manager\_ID = 101 WHERE ID IN (102, 121, 156);

>>UPDATE EMPLOYEE SET Manager\_ID = 102 WHERE ID IN (123, 114, 115);

>>UPDATE EMPLOYEE SET Manager\_ID = 121 WHERE ID = 127;



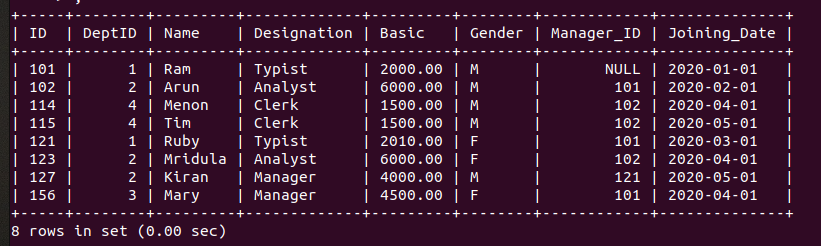
7. Add a column 'JoiningDate' and update values.

>>ALTER TABLE EMPLOYEE ADD Joining\_Date DATE;

-- Update with appropriate values (example: '2020-01-01')

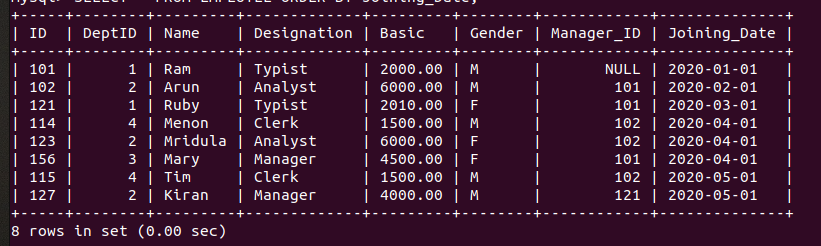
>>UPDATE EMPLOYEE SET Joining\_Date = '2020-01-01' WHERE ID = 101;

-- Repeat for other employees



8. Display employee details according to their seniority.

>>SELECT \* FROM EMPLOYEE ORDER BY Joining\_Date ;



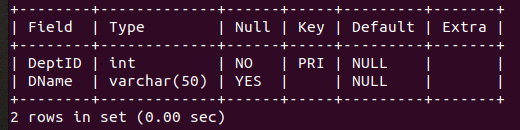
9. Create DEPARTMENT table.

>>CREATE TABLE Department (

DeptID INT PRIMARY KEY,

DName VARCHAR(50)

);



10. Insert values into the DEPARTMENT table.

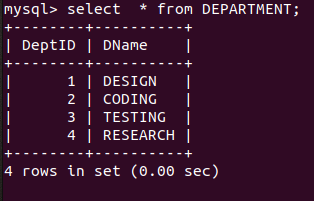
>>INSERT INTO Department (DeptID, DName) VALUES

(1, 'DESIGN'),

(2, 'CODING'),

(3, 'TESTING'),

(4, 'RESEARCH');

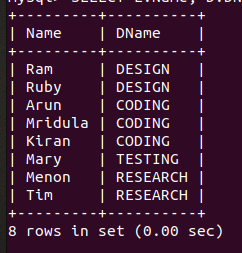


11. Display employee name and department name.

>>SELECT E.Name, D.DName

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID;



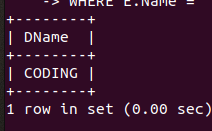
12. Display the department name of employee Arun.

>>SELECT D.DName

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

WHERE E.Name = 'Arun';



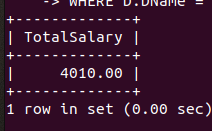
13. Display the salary given by the DESIGN department.

>>SELECT SUM(Basic) AS TotalSalary

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

WHERE D.DName = 'DESIGN';

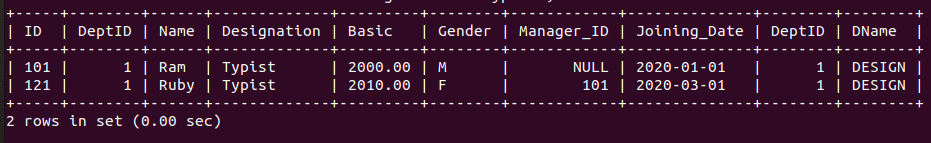


14. Display details of typists working in DESIGN department.

>>SELECT \* FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

WHERE E.Designation = 'Typist' AND D.DName = 'DESIGN';



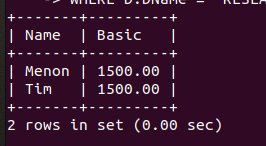
15. Display salaries of employees working in RESEARCH department.

>>SELECT E.Name, E.Basic

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

WHERE D.DName = 'RESEARCH';



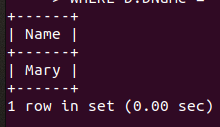
16. List female employees working in TESTING department.

>>SELECT E.Name

FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

WHERE E.Gender = 'F' AND D.DName = 'TESTING';

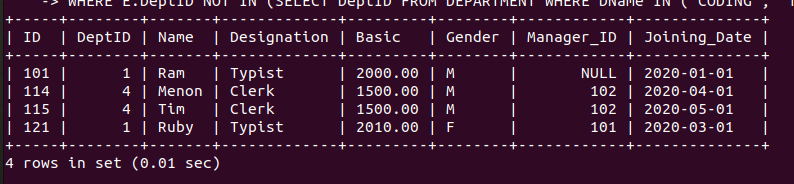


17. Display employees not working in CODING or TESTING department.

>>SELECT \* FROM EMPLOYEE E

WHERE E.DeptID NOT IN (SELECT DeptID FROM DEPARTMENT WHERE DName IN

('CODING', 'TESTING'));



18. Display the department giving maximum salary.

>>SELECT D.DeptName

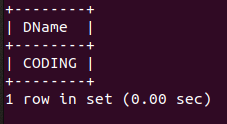
FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

GROUP BY D.DName

ORDER BY SUM(E.Basic) DESC

LIMIT 1;



19. Display the departments with the minimum number of employees.

>>SELECT D.DName

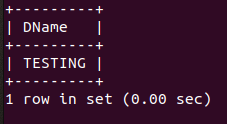
FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DeptID = D.DeptID

GROUP BY D.DName

ORDER BY COUNT(E.ID) ASC

LIMIT 1;



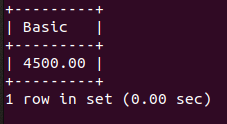
20. Display the second maximum salary.

>>SELECT DISTINCT Basic

FROM EMPLOYEE

ORDER BY Basic DESC

LIMIT 1 OFFSET 1;



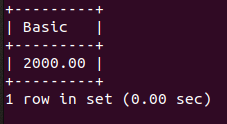
21. Display the second minimum salary.

>>SELECT DISTINCT Basic

FROM EMPLOYEE

ORDER BY Basic ASC

LIMIT 1 OFFSET 1;



22. Display employees getting salary greater than the average salary of their department.

>>SELECT E.Name

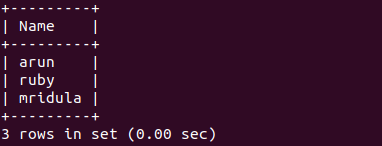
FROM EMPLOYEE E

JOIN (SELECT DeptID, AVG(Basic) AS AvgSalary FROM EMPLOYEE GROUP BY DeptID)

AS DeptAvg

ON E.DeptID = DeptAvg.DeptID

WHERE E.Basic > DeptAvg.AvgSalary;



23. Display employees working under manager Ram.

>>SELECT E.Name

FROM EMPLOYEE E

JOIN EMPLOYEE M ON E.ManagerID = M.ID

WHERE M.Name = 'Ram';

