

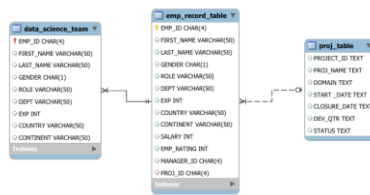
1. Create a database named *employee*, then import **data\_science\_team.csv** **proj\_table.csv** and **emp\_record\_table.csv** into the **employee** database from the given resources.

### Query –

Create Database Employee;

2. Create an ER diagram for the given **employee** database.

Screenshot –



3. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

### Query –

```
select EMP_ID,FIRST_NAME,LAST_NAME,GENDER,DEPT from emp_record_table
order By DEPT;
```

4. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:
  - less than two
  - greater than four
  - between two and four

### Query-

```
select EMP_ID,FIRST_NAME,LAST_NAME,GENDER,DEPT,EMP_RATING,
CASE
when Emp_rating < 2 then 'Low'
  when Emp_rating <= 4 then 'Avg'
  else 'High'
end as Rating_Status
from emp_record_table;
```

5. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the *Finance* department from the employee table and then give the resultant column alias as NAME.

**Query -**

```
select Concat(First_Name,' ',Last_Name) as NAME from emp_record_table
where DEPT = 'Finance';
```

6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

**Query -**

```
select m.FIRST_NAME Manager_Name,Count(*) from
emp_record_table e JOIN emp_record_table m
ON e.Manager_ID = m.EMP_ID
group by m.FIRST_NAME;
```

7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

**Query -**

```
select * from emp_record_table where DEPT = 'HEALTHCARE'
UNION
select * from emp_record_table where DEPT = 'FINANCE';
```

8. Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.

**Query -**

```
select EMP_ID,FIRST_NAME,LAST_NAME,ROLE,DEPT,EMP_RATING,
Max(Emp_Rating) over(partition by DEPT ) Max_Emp_Rating from emp_record_table;
```

9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

**Query**

```
Select role, MIN(Salary) lowest, Max(Salary) Highest from emp_record_table group by ROLE;
```

10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

**Query-**

```
select EMP_ID,FIRST_NAME,LAST_NAME,EXP, Rank() over(order by Exp desc) as RankByExp from  
emp_record_table;
```

11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

**Query-**

```
create View V_Emp_6k  
as  
select EMP_ID,FIRST_NAME,LAST_NAME,COUNTRY from emp_record_table where Salary > 6000  
ORDER bY COUNTRY;
```

```
SELECT * FROM V_Emp_6k;
```

12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

**Query –**

```
SELECT * FROM emp_record_table where Emp_ID IN (SELECT EMP_ID FROM emp_record_table WHERE  
EXP > 10);
```

13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

**Query-**

```
DELIMITER $$  
USE `employee`$$  
CREATE PROCEDURE P_Emp_3YrsExp ()  
BEGIN  
select * from emp_record_table where exp > 3;  
END$$
```

```
DELIMITER ;
```

```
call P_Emp_3YrsExp;
```

14. Write a query using stored functions in the project table to check whether the job profile assigned to each employee in the data science team matches the organization's set standard.

The standard being:

For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',

For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',

For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',

For an employee with the experience of 12 to 16 years assign 'MANAGER'.

#### Query-

```
DELIMITER $$
USE `employee`$$
CREATE FUNCTION `Task14`(eid varchar(5))
RETURNS varchar(100)
    DETERMINISTIC
BEGIN
    declare ex int;
    -- declare r varchar(80);
    declare vrole varchar(100);
    declare flag varchar(10);
    select exp, ROLE into ex, VROLE from data_science_team where emp_ID = eid;

    if ex > 12 and ex < 16 then
        if VROLE = 'Manager' then
            set flag = 'Yes';
        else
            set flag = 'No';
        end if;
        # set r = 'Manager';
    elseif ex > 10 and ex <= 12 then
        if VROLE = 'LEAD DATA SCIENTIST' then
            set flag = 'Yes';
        else
            set flag = 'No';
        end if;
        #set r = 'LEAD DATA SCIENTIST';
    elseif ex > 5 and ex <=10 then
        if VROLE = 'SENIOR DATA SCIENTIST' then
            set flag = 'Yes';
        else
            set flag = 'No';
        end if;
        #set r = 'SENIOR DATA SCIENTIST';
    elseif ex > 2 and ex <=5 then
        if VROLE = 'ASSOCIATE DATA SCIENTIST' then
            set flag = 'Yes';
        else
            set flag = 'No';
        end if;
        #set r = 'ASSOCIATE DATA SCIENTIST';
    elseif ex <= 2 then
        if VROLE = 'JUNIOR DATA SCIENTIST' then
            set flag = 'Yes';
        else
            set flag = 'No';
        end if;
        #set r = 'JUNIOR DATA SCIENTIST';
```

end if;

```
RETURN flag;
END$$
DELIMITER ;
;
SELECT *,Task14(Emp_ID) FROM data_science_team;
```

15. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is 'Eric' in the employee table after checking the execution plan.

**Query-**

```
select * from emp_record_table where First_Name='Eric';
create Index idx_emp_Fname on emp_record_table(First_Name);
select * from emp_record_table where First_Name='Eric';
```

16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

**Query -**

```
select EMP_ID,FIRST_NAME,SALARY,EMP_RATING, (Emp_rating * 0.05 * SALARY) Bonus from
emp_record_table;
```

17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

**Query-**

```
select CONTINENT,COUNTRY,Avg(Salary) from emp_record_table
group by CONTINENT,COUNTRY with rollup
order by CONTINENT,COUNTRY;
```

---

---