

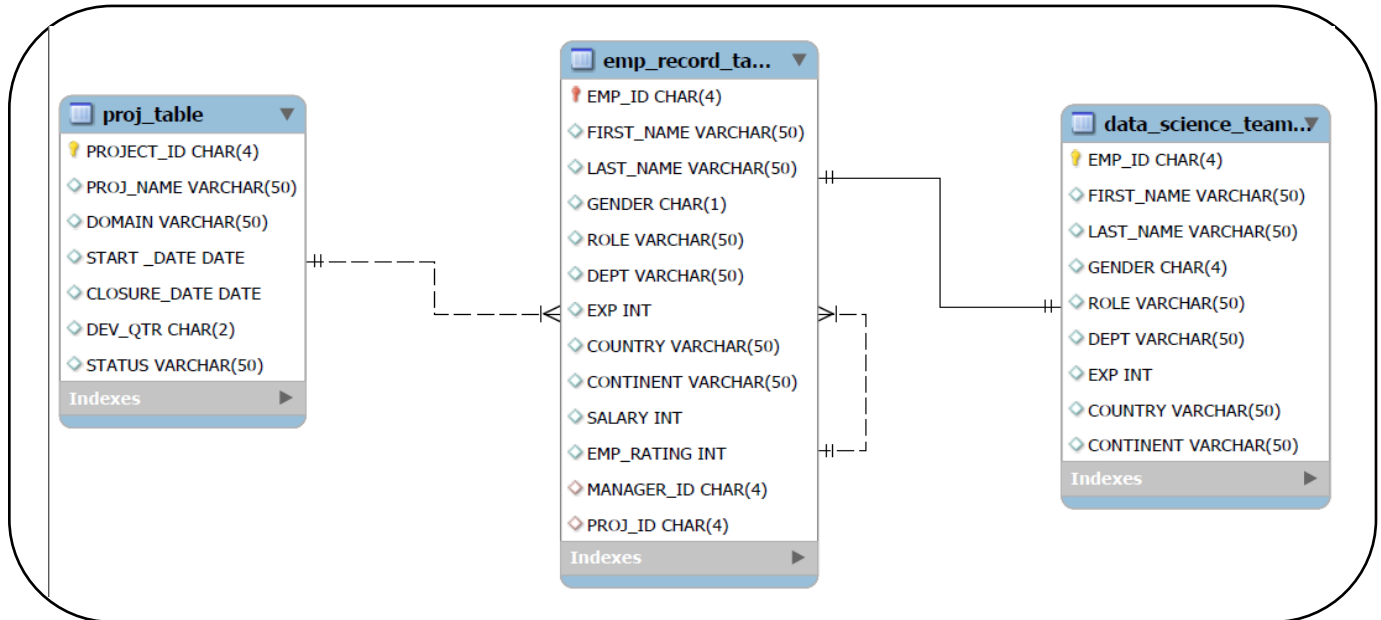
1. Create a database named *employee*, then import **data_science_team.csv**, **proj_table.csv** and **emp_record_table.csv** into the **employee** database from the given resources.

Ans:-

```
create database employee;
use employee;
```

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

Ans:-



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.

Ans:-

```
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

Ans:-

```

select emp_id, first_name, last_name, gender, dept, emp_rating,
       case
         when emp_rating < 2 then 'Less than 2'
         when emp_rating > 4 then 'Greater than 4'
         else 'Between 2 and 4'
       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.

Ans:-

```

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).

Ans:-

```

select m.first_name, m.Role, e.first_name,
       count(*) over(partition by m.first_name)
from emp_record_table e join emp_record_table m
on e.manager_id=m.emp_id;

```

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.

Ans:-

```

select * from emp_record_table where dept='Finance'
UNION
select * from emp_record_table where dept='Healthcare';

```

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.

Ans:-

```

select emp_id, first_name, last_name, role, dept, emp_rating,
       max(emp_rating) over(partition by dept) MAX_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.

Ans:-

```
select role, min(salary) Min_salary, max(salary) Max_salary 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.

Ans:-

```
select *, rank() over(order by exp desc) 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.

Ans:-

```
create view V_Emp_above6k  
as  
select emp_id, first_name, last_name, salary, country from  
emp_record_table  
where salary > 6000  
order by Country;  
  
select * from V_Emp_above6k;
```

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

Ans:-

```
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.

Ans:-

```
USE `employee`;  
DROP procedure IF EXISTS `Emp_Above3yrsExp`;  
  
USE `employee`;  
DROP procedure IF EXISTS `employee`.`Emp_Above3yrsExp`;  
;  
  
DELIMITER $$  
USE `employee` $$  
CREATE PROCEDURE Emp_Above3yrsExp()  
BEGIN  
    select * from emp_record_table where exp > 3;  
END$$  
  
DELIMITER ;  
;  
  
call Emp_Above3yrsExp();
```

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'.

Ans:-

```
USE `employee`;
DROP function IF EXISTS `Check_JobProfiles`;

DELIMITER $$
USE `employee` $$
CREATE FUNCTION Check_JobProfiles (eid varchar(5))
RETURNS varchar(100)
deterministic
BEGIN
    declare ex int;
    declare r varchar(80);
    declare vrole varchar(100);
    declare flag varchar(100);
    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;
    elseif ex > 2 and ex <=5 then
        if vrole = 'ASSOCIATE DATA SCIENTIST'
```

```
    then
        set flag = 'Yes';
    else
        set flag = 'No';
    end if;
    elseif ex <= 2 then
        if vrole = 'JUNIOR DATA SCIENTIST' then
            set flag = 'YES';
        else
            set flag = 'No';
        end if;
    end if;
    RETURN flag;
END$$

DELIMITER ;

select *, Check_JobProfiles (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.

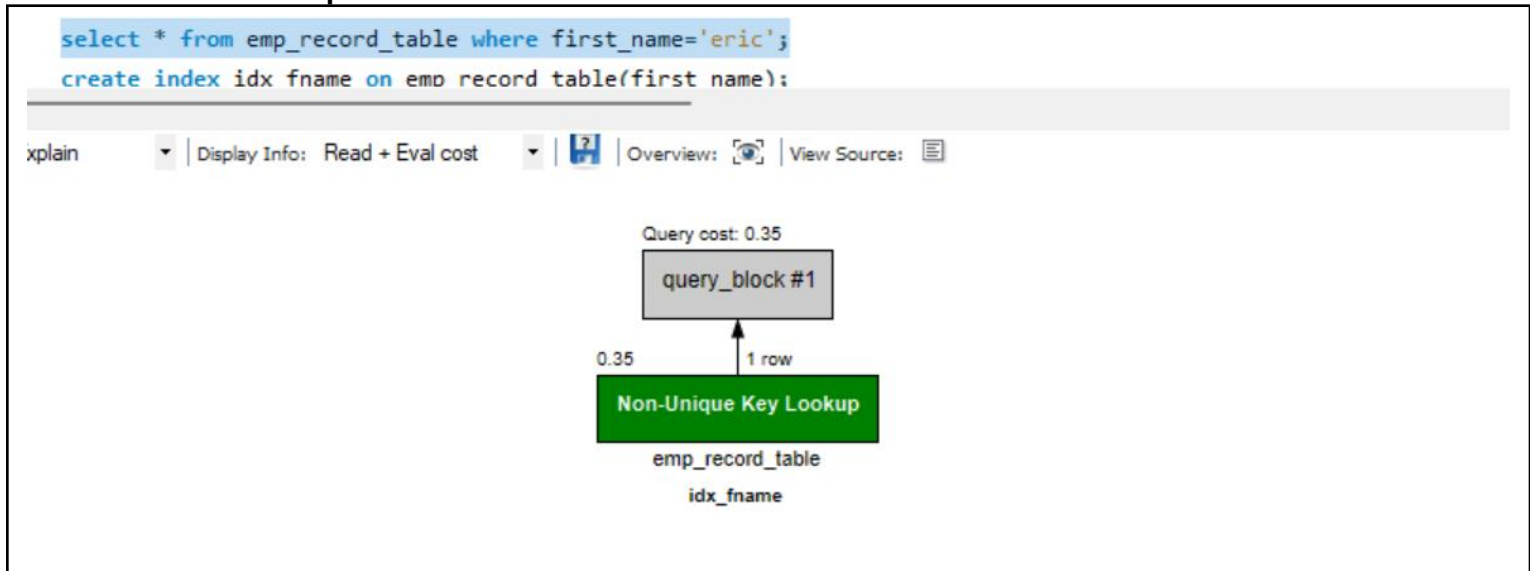
Ans:-

```
select * from emp_record_table where first_name='eric';  
create index idx_fname on emp_record_table(first_name);
```

Execution Plan without Index:-



Execution plan after Index:-



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).

Ans:-

```
select emp_id, first_name, last_name, salary, emp_rating,  
       salary * 0.05 * emp_rating as 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.

Ans:-

```
select continent, country, avg(salary) from emp_record_table  
group by continent, country with rollup;
```