

ScienceQtech Employee Performance Mapping.

The task to be performed:

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.

```
create database employee;
```

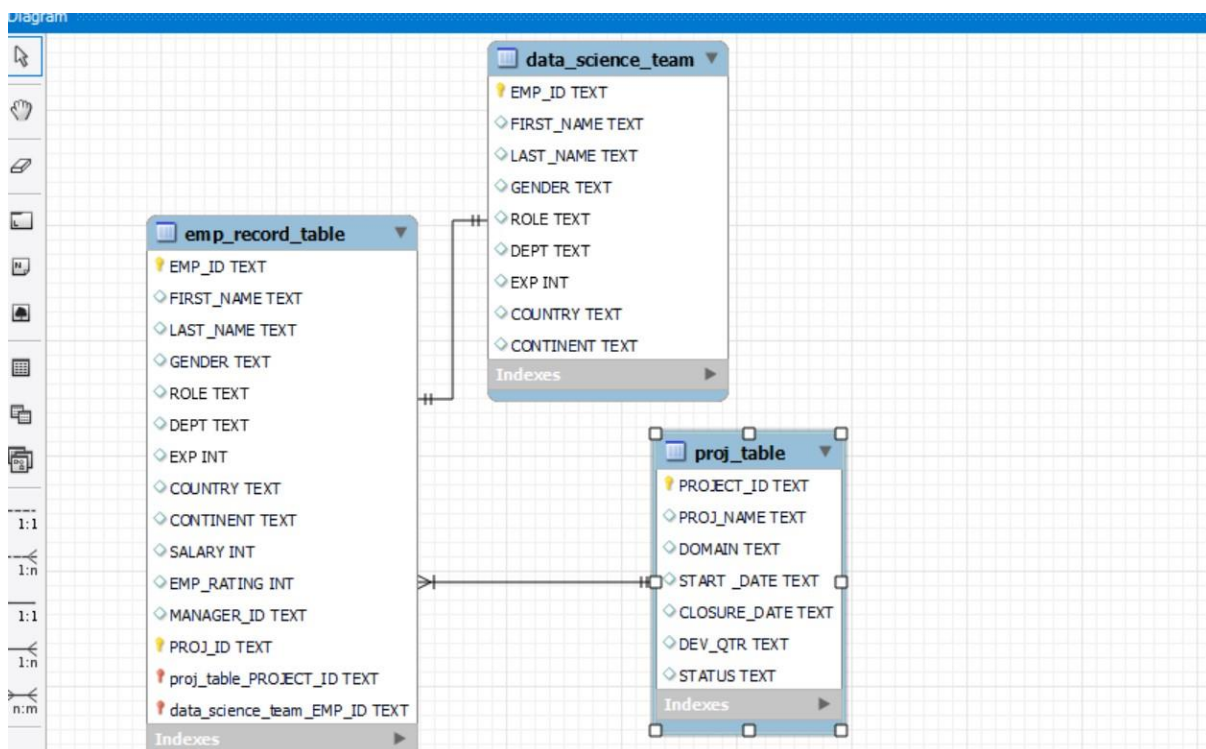
```
use employee;
```

```
select * from data_science_team;
```

```
select * from emp_record_table;
```

```
select * from proj_table;
```

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



#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.

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

#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

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table  
where EMP_RATING < 2;
```

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table  
where EMP_RATING > 4;
```

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table  
where EMP_RATING > 2 and EMP_RATING < 4;
```

#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.

```
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).

```
SELECT  
e.EMP_ID,  
e.FIRST_NAME,  
COUNT(DISTINCT r.EMP_ID) AS number_of_reporters  
FROM  
emp_record_table e  
inner join  
emp_record_table r ON e.EMP_ID = r.MANAGER_ID  
GROUP BY
```

```
e.EMP_ID, e.FIRST_NAME
```

```
ORDER BY
```

```
number_of_reporters DESC;
```

#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.

```
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.

```
SELECT m.EMP_ID, m.FIRST_NAME, m.LAST_NAME, m.ROLE, m.DEPT, m.EMP_RATING,  
max(m.EMP_RATING) OVER (PARTITION BY m.DEPT) AS "MAX_DEPT_RATING" FROM  
emp_record_table m ORDER BY DEPT;
```

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.

```
SELECT
```

```
    ROLE,
```

```
    MIN(salary) AS min_salary,
```

```
    MAX(salary) AS 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.

```
select EMP_ID, FIRST_NAME, EXP, Rank() over(order by EXP) as 'rank' 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.

```
create view empview as select EMP_ID, FIRST_NAME, LAST_NAME, DEPT, COUNTRY from  
emp_record_table where SALARY > 6000;
```

```
select * from empview;
```

12. Write a nested query to find employees with experience of more than ten years.

```
select * 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.

```
DELIMITER //
```

```
create procedure GETEMPEXP()
```

```
begin
```

```
select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP from emp_record_table where EXP > 3;
```

```
end //
```

```
DELIMITER ;
```

```
call GETEMPEXP();
```

14. 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'.

```
DELIMITER //
```

```
CREATE FUNCTION Employee_ROLE(EXP int)
```

```
RETURNS VARCHAR(40)
```

```
DETERMINISTIC
```

```
BEGIN
```

```

DECLARE ROLE VARCHAR(40);

IF EXP>12 AND 16 THEN SET Employee_ROLE="MANAGER";

ELSEIF EXP>10 AND 12 THEN SET Employee_ROLE ="LEAD DATA SCIENTIST";

ELSEIF EXP>5 AND 10 THEN SET Employee_ROLE ="SENIOR DATA SCIENTIST";

ELSEIF EXP>2 AND 5 THEN SET Employee_ROLE ="ASSOCIATE DATA SCIENTIST";

ELSEIF EXP<=2 THEN SET Employee_ROLE ="JUNIOR DATA SCIENTIST";

END IF;

RETURN (ROLE);

END //

SELECT EXP,ROLE 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.

```

CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(20));

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).

```

update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING));

SELECT * 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.

```

SELECT EMP_ID,FIRST_NAME,LAST_NAME,SALARY,COUNTRY,CONTINENT,
AVG(salary)OVER(PARTITION BY COUNTRY) as AVG_salary_IN_COUNTRY,COUNT(*)OVER(PARTITION
BY COUNTRY) as COUNT_IN_COUNTRY FROM emp_record_table;

```