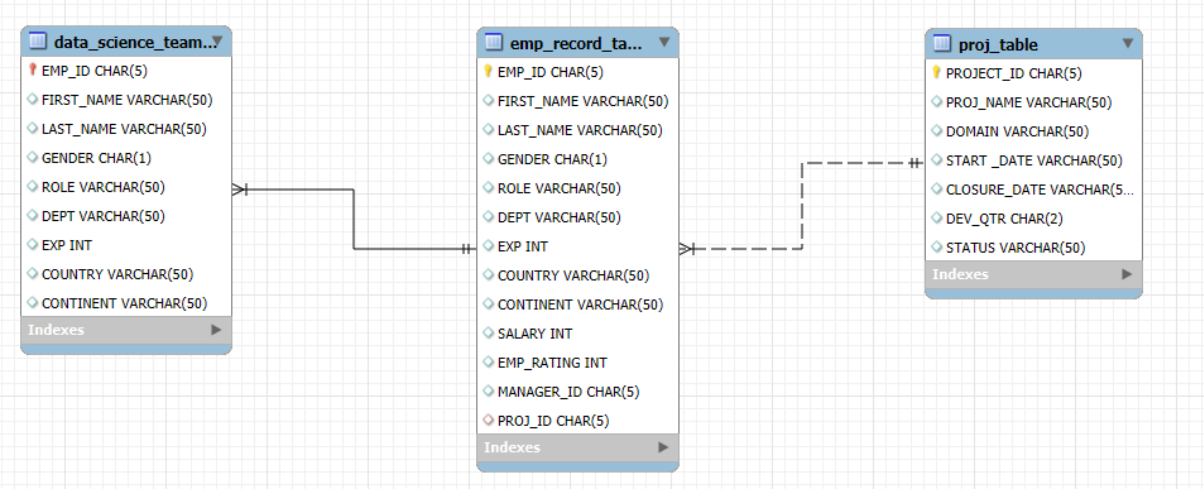
Employee Performance Mapping

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;

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 employee.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

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING,
CASE
  WHEN EMP_RATING < 2 **THEN** 'less than two'
  WHEN EMP_RATING <= 4 **THEN** 'between two and four'
  ELSE 'greater than four'
END AS Rating
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.

SELECT FIRST_NAME, LAST_NAME,DEPT, **CONCAT**(FIRST_NAME, ' ',LAST_NAME) **AS** Full_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
   M.EMP_ID,
   M.FIRST_NAME,
   M.LAST_NAME,
   COUNT(E.EMP_ID) AS Report_Count
FROM emp_record_table M
JOIN emp_record_table E
   ON M.EMP_ID = E.MANAGER_ID
GROUP BY 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.

```
SELECT *
FROM emp_record_table
WHERE DEPT = 'FINANCE'
UNION
SELECT *
FROM emp_record_table
WHERE DEPT = 'HEALTHCARE'
ORDER BY DEPT;
```

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 EMP_ID,FIRST_NAME,LAST_NAME,ROLE,DEPT,EMP_RATING, MAX(EMP_RATING) OVER
(PARTITION BY DEPT) AS MAX_RATING
FROM emp_record_table
ORDER BY EMP_RATING DESC;
```

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, CONCAT(FIRST_NAME, ' ', LAST_NAME) AS FULL_NAME, DEPT,EXP, RANK()
OVER(ORDER BY EXP DESC) AS RANK_BY_EXP
FROM emp_record_table;
```

-- OR--

```
SELECT *, RANK() OVER(ORDER BY EXP DESC) AS RANK_BY_EXP
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 OR REPLACE VIEW EMP_COUNTRY_VIEW
AS
SELECT EMP_ID, FIRST_NAME, LAST_NAME,COUNTRY,SALARY
FROM emp_record_table
WHERE SALARY>6000
ORDER BY COUNTRY;
SELECT * FROM EMP_COUNTRY_VIEW;
-- OR --
CREATE VIEW EmpSalAbove6k AS
SELECT * FROM emp_record_table WHERE salary > 6000;
SELECT * FROM EmpSalAbove6k;
```

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

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME,exp
FROM(
        SELECT * FROM emp_record_table
   WHERE EXP > 10
   ORDER BY exp
) AS EXP_GREATER_THAN_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 $$
USE `employee`$$
CREATE PROCEDURE Exp_above_3Years ()
BEGIN
        SELECT * FROM emp_record_table WHERE exp > 3;
END$$

DELIMITER ;
-- Execute the Stored Procedure
CALL Exp_above_3Years;
```

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'.

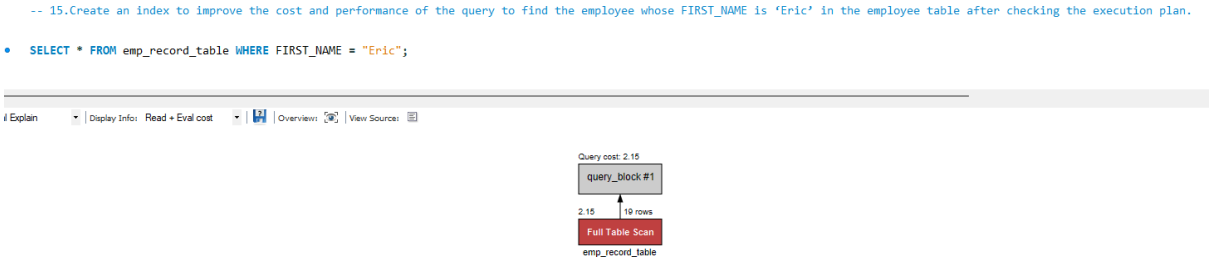
```
delimiter //
CREATE FUNCTION check_role(exp int)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
        DECLARE chck VARCHAR(40);
   IF EXP <= 2 THEN
                SET chck = "JUNIOR DATA SCIENTIST";
        ELSEIF exp > 2 AND exp <= 5 THEN
                SET chck = "ASSOCIATE DATA SCIENTIST";
        ELSEIF exp > 5 AND exp <= 10 THEN
                SET chck = "SENIOR DATA SCIENTIST";
        ELSEIF exp > 10 AND exp <= 12 THEN
                SET chck = "LEAD DATA SCIENTIST";
        ELSEIF exp > 12 AND exp <= 16 THEN
                SET chck = "MANAGER";
        END IF;
   RETURN(chck);
END //
delimiter ;

-- checking Data Science Team
SELECT  EMP_ID, FIRST_NAME, LAST_NAME, ROLE, check_role(exp)
FROM data_science_team WHERE ROLE != check_role(exp);
```

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.

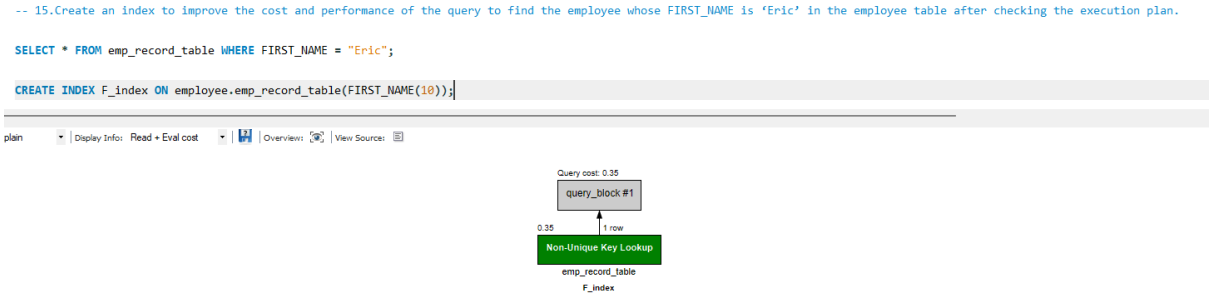
Before the index

```
SELECT * FROM emp_record_table WHERE FIRST_NAME = "Eric";
```



After creating the INDEX

```
CREATE INDEX F_index ON employee.emp_record_table(FIRST_NAME(10));
```



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).

```
SELECT EMP_ID,
CONCAT(FIRST_NAME," ",LAST_NAME) AS NAME, EMP_RATING,
SALARY,(SALARY*0.05)*EMP_RATING AS BONUS
FROM emp_record_table;
-- OR--
SELECT *, EMP_RATING * .05 * Salary 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.

```
SELECT CONTINENT, COUNTRY, AVG(SALARY) AS Avg_Salary
FROM emp_record_table
GROUP BY CONTINENT, COUNTRY WITH ROLLUP
ORDER BY CONTINENT, COUNTRY;
```