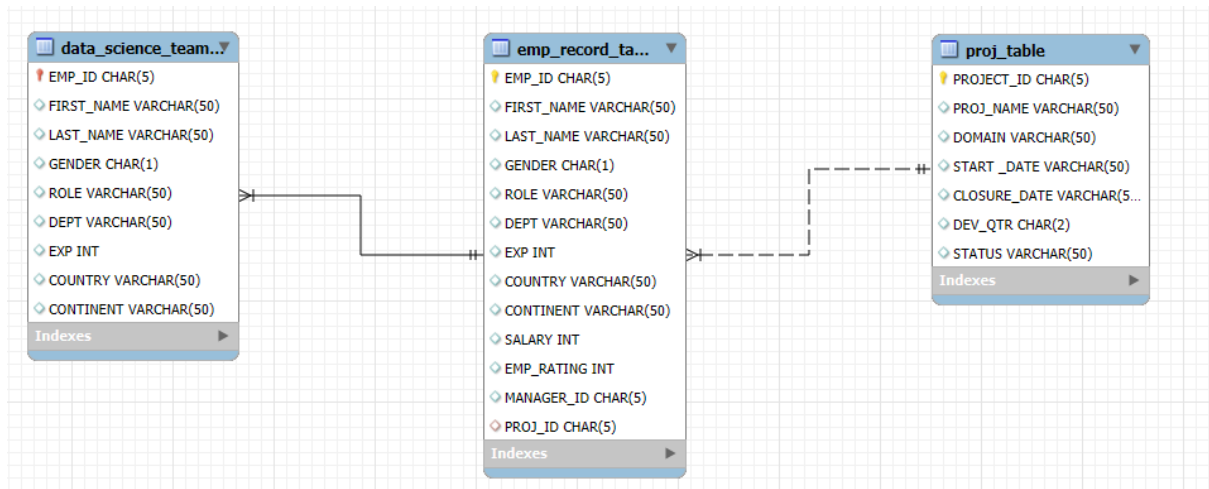


## Employee Performance Mapping

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;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E010	William	Butler	M	AUTOMOTIVE
E204	Karene	Nowak	F	AUTOMOTIVE
E428	Pete	Allen	M	AUTOMOTIVE
E532	Claire	Brennan	F	AUTOMOTIVE
E005	Eric	Hoffman	M	FINANCE
E103	Emily	Grove	F	FINANCE
E403	Steve	Hoffman	M	FINANCE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E505	Chad	Wilson	M	HEALTHCARE
E245	Nian	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL
E478	David	Smith	M	RETAIL
E583	Janet	Hale	F	RETAIL
E612	Tracy	Norris	F	RETAIL
E620	Katrina	Allen	F	RETAIL
E640	Jenifer	Jhones	F	RETAIL
NULL	NULL	NULL	NULL	NULL

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;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING	Rating
E001	Arthur	Black	M	ALL	5	greater than four
E005	Eric	Hoffman	M	FINANCE	3	between two and four
E010	William	Butler	M	AUTOMOTIVE	2	between two and four
E052	Dianna	Wilson	F	HEALTHCARE	5	greater than four
E057	Dorothy	Wilson	F	HEALTHCARE	1	less than two
E083	Patrick	Voltz	M	HEALTHCARE	5	greater than four
E103	Emily	Grove	F	FINANCE	4	between two and four
E204	Karene	Nowak	F	AUTOMOTIVE	5	greater than four
E245	Nian	Zhen	M	RETAIL	2	between two and four
E260	Roy	Collins	M	RETAIL	3	between two and four
E403	Steve	Hoffman	M	FINANCE	3	between two and four
E428	Pete	Allen	M	AUTOMOTIVE	4	between two and four
E478	David	Smith	M	RETAIL	4	between two and four
E505	Chad	Wilson	M	HEALTHCARE	2	between two and four
E532	Claire	Brennan	F	AUTOMOTIVE	1	less than two
E583	Janet	Hale	F	RETAIL	2	between two and four
E612	Tracy	Norris	F	RETAIL	4	between two and four
E620	Katrina	Allen	F	RETAIL	1	less than two
E640	Jenifer	Jhones	F	RETAIL	4	between two and four

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';
```

FIRST_NAME	LAST_NAME	DEPT	Full_Name
Eric	Hoffman	FINANCE	Eric Hoffman
Emily	Grove	FINANCE	Emily Grove
Steve	Hoffman	FINANCE	Steve Hoffman

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	Report_Count
E103	Emily	Grove	2
E428	Pete	Allen	3
E083	Patrick	Voltz	3
E001	Arthur	Black	5
E583	Janet	Hale	3
E612	Tracy	Norris	2

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4	E001	NULL
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E103	P105
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	NULL
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	MAX_RATING
E001	Arthur	Black	PRESIDENT	ALL	5	5
E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
E103	Emily	Grove	MANAGER	FINANCE	4	4
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
E612	Tracy	Norris	MANAGER	RETAIL	4	4
E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	RETAIL	4	4
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	3	4
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E245	Nian	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4
E583	Janet	Hale	MANAGER	RETAIL	2	4
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4

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;
```

ROLE	MIN_SALARY	MAX_SALARY
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	5500	7700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

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;
```

EMP_ID	FULL_NAME	DEPT	EXP	RANK_BY_EXP
E001	Arthur Black	ALL	20	1
E083	Patrick Voltz	HEALTHCARE	15	2
E103	Emily Grove	FINANCE	14	3
E428	Pete Allen	AUTOMOTIVE	14	3
E583	Janet Hale	RETAIL	14	3
E612	Tracy Norris	RETAIL	13	6
E010	William Butler	AUTOMOTIVE	12	7
E005	Eric Hoffman	FINANCE	11	8
E057	Dorothy Wilson	HEALTHCARE	9	9
E204	Karene Nowak	AUTOMOTIVE	8	10
E260	Roy Collins	RETAIL	7	11
E052	Dianna Wilson	HEALTHCARE	6	12
E245	Nian Zhen	RETAIL	6	12
E505	Chad Wilson	HEALTHCARE	5	14
E403	Steve Hoffman	FINANCE	4	15
E478	David Smith	RETAIL	3	16
E532	Claire Brennan	AUTOMOTIVE	3	16
E620	Katrina Allen	RETAIL	2	18
E640	Jenifer Jhones	RETAIL	1	19

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
E103	Emily	Grove	CANADA	10500
E245	Nian	Zhen	CHINA	6500
E583	Janet	Hale	COLOMBIA	10000
E010	William	Butler	FRANCE	9000
E204	Karene	Nowak	GERMANY	7500
E428	Pete	Allen	GERMANY	11000
E260	Roy	Collins	INDIA	7000
E612	Tracy	Norris	INDIA	8500
E001	Arthur	Black	USA	16500
E005	Eric	Hoffman	USA	8500
E057	Dorothy	Wilson	USA	7700
E083	Patrick	Voltz	USA	9500

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	EXP
E005	Eric	Hoffman	11
E010	William	Butler	12
E612	Tracy	Norris	13
E103	Emily	Grove	14
E428	Pete	Allen	14
E583	Janet	Hale	14
E083	Patrick	Voltz	15
E001	Arthur	Black	20

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;

```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5	NULL	NULL
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E428	P204
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	NULL
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4	E001	NULL
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5	E428	P204
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2	E583	P109
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E583	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E103	P105
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4	E001	NULL
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2	E001	NULL
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	NULL

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** chkck **VARCHAR**(40);

**IF** EXP <= 2 **THEN**

        SET chkck = "JUNIOR DATA SCIENTIST";

**ELSEIF** exp > 2 **AND** exp <= 5 **THEN**

        SET chkck = "ASSOCIATE DATA SCIENTIST";

**ELSEIF** exp > 5 **AND** exp <= 10 **THEN**

        SET chkck = "SENIOR DATA SCIENTIST";

**ELSEIF** exp > 10 **AND** exp <= 12 **THEN**

        SET chkck = "LEAD DATA SCIENTIST";

**ELSEIF** exp > 12 **AND** exp <= 16 **THEN**

        SET chkck = "MANAGER";

**END IF**;

**RETURN**(chkck);

**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);

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	check_role(exp)
--------	------------	-----------	------	-----------------

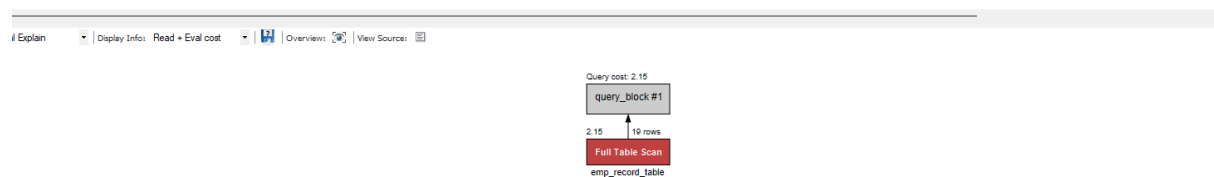
15. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is 'Eric' in the employee table after checking the execution plan.

**Before the index**

**SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = "Eric";**

-- 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.

• **SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = "Eric";**



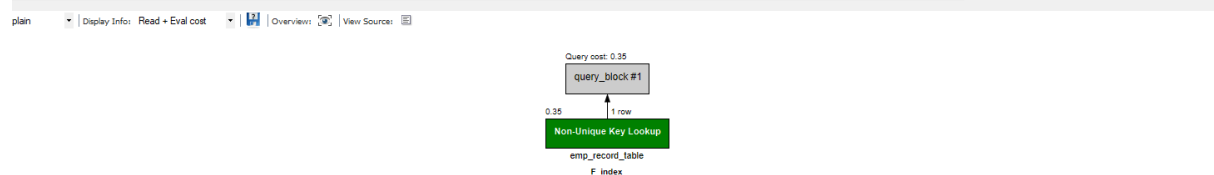
**After creating the INDEX**

**CREATE INDEX F\_index ON employee.emp\_record\_table(FIRST\_NAME(10));**

-- 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.

**SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = "Eric";**

**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
```

EMP_ID	NAME	EMP_RATING	SALARY	BONUS
E001	Arthur Black	5	16500	4125.00
E005	Eric Hoffman	3	8500	1275.00
E010	William Butler	2	9000	900.00
E052	Dianna Wilson	5	5500	1375.00
E057	Dorothy Wilson	1	7700	385.00
E083	Patrick Voltz	5	9500	2375.00
E103	Emily Grove	4	10500	2100.00
E204	Karene Nowak	5	7500	1875.00
E245	Nian Zhen	2	6500	650.00
E260	Roy Collins	3	7000	1050.00
E403	Steve Hoffman	3	5000	750.00
E428	Pete Allen	4	11000	2200.00
E478	David Smith	4	4000	800.00
E505	Chad Wilson	2	5000	500.00
E532	Claire Brennan	1	4300	215.00

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;
```

CONTINENT	COUNTRY	Avg_Salary
NULL	NULL	7463.1579
ASIA	NULL	6250.0000
ASIA	CHINA	6500.0000
ASIA	INDIA	6166.6667
EUROPE	NULL	7950.0000
EUROPE	FRANCE	9000.0000
EUROPE	GERMANY	7600.0000
NORTH AMERICA	NULL	8525.0000
NORTH AMERICA	CANADA	7000.0000
NORTH AMERICA	USA	9440.0000
SOUTH AMERICA	NULL	5600.0000
SOUTH AMERICA	COLOMBIA	5600.0000