SQL Training

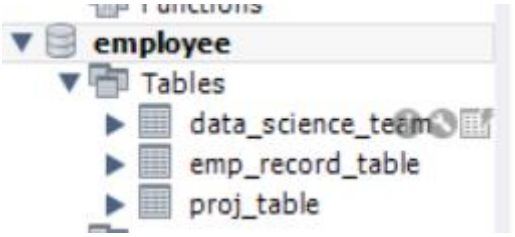
Course-End Project Assignment - 1

ScienceQtech Employee Performance Mapping

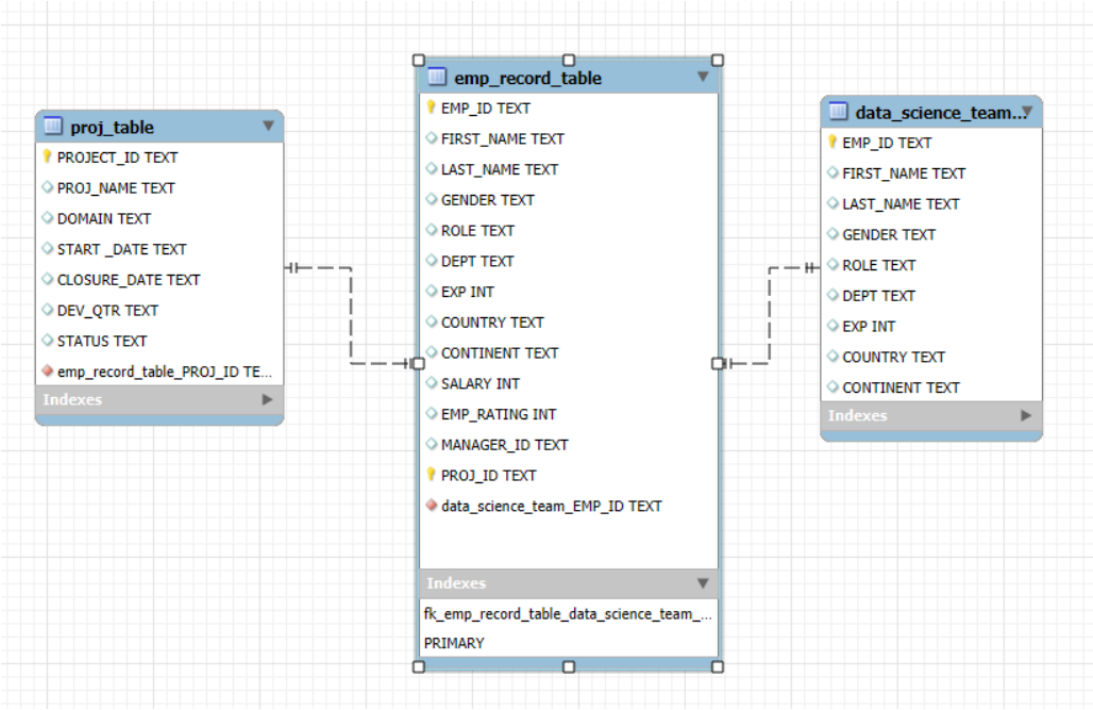
Submitted By
Aruna Rajeswari Sivaperuman

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.

```
1 •  create database employee;
```



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 AS 'DEPARTMENT'
FROM
    emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT
E001	Arthur	Black	M	ALL
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE

31 15:37:10 SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT AS 'DEPARTMENT' FROM emp_rec... 19 row(s) returned

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 AS 'DEPARTMENT',
    CASE
        WHEN EMP_RATING < 2 THEN 'Rating below 2'
        WHEN EMP_RATING > 4 THEN 'Rating Above 4'
        WHEN EMP_RATING BETWEEN 2 AND 4 THEN 'Rating between 2 and 4'
    END AS 'EMPLOYEE_RATING'
FROM
    emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT	EMPLOYEE_RATING
E001	Arthur	Black	M	ALL	Rating Above 4
E005	Eric	Hoffman	M	FINANCE	Rating between 2 and 4
E010	William	Butler	M	AUTOMOTIVE	Rating between 2 and 4
E052	Dianna	Wilson	F	HEALTHCARE	Rating Above 4
E057	Dorothy	Wilson	F	HEALTHCARE	Rating below 2
E083	Patrick	Voltz	M	HEALTHCARE	Rating Above 4
E103	Emily	Grove	F	FINANCE	Rating between 2 and 4

33 15:41:34 SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT AS 'DEPARTMENT', CAS... 19 row(s) returned

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(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS 'NAME'
FROM
    emp_record_table
WHERE
    DEPT = 'FINANCE';
```

NAME
Eric Hoffman
Emily Grove
Steve Hoffman

34 15:45:39 SELECT CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS 'NAME' FROM emp_record_table ... 3 row(s) returned

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,
    e.LAST_NAME,
    e.ROLE,
    COUNT(r.EMP_ID) AS NUM_OF_REPORTERS
FROM
    emp_record_table e
JOIN
    emp_record_table r ON e.EMP_ID = r.MANAGER_ID
GROUP BY
    e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE;
```

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

35 15:48:06 SELECT e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE, COUNT(r.EMP_ID) AS NUM_... 6 row(s) returned

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
    EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
    emp_record_table
WHERE
    DEPT = 'Healthcare'

UNION

SELECT
    EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
    emp_record_table
WHERE
    DEPT = 'Finance';
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E505	Chad	Wilson	M	HEALTHCARE
E005	Eric	Hoffman	M	FINANCE
E103	Emily	Grove	F	FINANCE
E403	Steve	Hoffman	M	FINANCE

```
36 15:53:21 SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table WHERE ... 7 row(s) returned
```

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.

```sql
SELECT
    e.EMP_ID,
    e.FIRST_NAME,
    e.LAST_NAME,
    e.ROLE,
    e.DEPT AS DEPARTMENT,
    e.EMP_RATING,
    d.MAX_DEPT_RATING
FROM
    emp_record_table e
JOIN (
    SELECT
        DEPT,
        MAX(EMP_RATING) AS MAX_DEPT_RATING
    FROM
        emp_record_table
    GROUP BY DEPT
) d ON e.DEPT = d.DEPT;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPARTMENT	EMP_RATING	MAX_DEPT_RATING
E001	Arthur	Black	PRESIDENT	ALL	5	5
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
E103	Emily	Grove	MANAGER	FINANCE	4	4
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5

```
37 15:54:34 SELECT e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE, e.DEPT AS DEPARTMENT, ... 19 row(s) returned
```

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.

```sql
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

38 15:55:07 SELECT ROLE, MIN(SALARY) AS MIN_SALARY, MAX(SALARY) AS MAX_SALARY FROM emp ... 6 row(s) returned

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

```sql
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    ROLE,
    DEPT,
    EXP,
    RANK() OVER (ORDER BY EXP DESC) AS EXP_RANK
FROM
    emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	EXP_RANK
E001	Arthur	Black	PRESIDENT	ALL	20	1
E083	Patrick	Voltz	MANAGER	HEALTHCARE	15	2
E103	Emily	Grove	MANAGER	FINANCE	14	3
E428	Pete	Allen	MANAGER	AUTOMOTIVE	14	3
E583	Janet	Hale	MANAGER	RETAIL	14	3
E612	Tracy	Norris	MANAGER	RETAIL	13	6
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	12	7
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	11	8
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	9	9

30 15:34:38 SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP, RANK() OVER (ORD... 19 row(s) returned

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 high_salary_employees_by_country AS
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    COUNTRY,
    SALARY
FROM
    emp_record_table
WHERE
    SALARY > 6000;
```

39 15:56:58 CREATE VIEW high_salary_employees_by_country AS SELECT EMP_ID, FIRST_NAME, LAST_NA... 0 row(s) affected

```
SELECT * FROM high_salary_employees_by_country;
```

EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
E001	Arthur	Black	USA	16500
E005	Eric	Hoffman	USA	[illegible]
E010	William	Butler	FRANCE	[illegible]
E057	Dorothy	Wilson	USA	7700

42 15:59:57 SELECT * FROM high_salary_employees_by_country LIMIT 0, 1000 12 row(s) returned

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

```sql
SELECT EMP_ID,CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS 'NAME',ROLE,EXP
FROM emp_record_table
WHERE EMP_ID IN (
    SELECT EMP_ID
    FROM emp_record_table
    WHERE EXP > 10
);
```

EMP_ID	NAME	ROLE	EXP
E001	Arthur Black	PRESIDENT	20
E005	Eric Hoffman	LEAD DATA SCIENTIST	11
E010	William Butler	LEAD DATA SCIENTIST	12
E083	Patrick Voltz	MANAGER	15

45 16:07:50 SELECT EMP_ID,CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS 'NAME',ROLE,EXP FROM emp... 8 row(s) returned

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.

```sql
DELIMITER //

CREATE PROCEDURE GetExperiencedEmployees()
BEGIN
    SELECT
        EMP_ID,
        FIRST_NAME,
        LAST_NAME,
        ROLE,
        DEPT,
        EXP,
        SALARY
    FROM
        emp_record_table
    WHERE
        EXP > 3;
END //

DELIMITER ;
```

46 16:12:16 CREATE PROCEDURE GetExperiencedEmployees() BEGIN SELECT EMP_ID, FIRST_NAME, ... 0 row(s) affected

```sql
CALL GetExperiencedEmployees();
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	SALARY
E001	Arthur	Black	PRESIDENT	ALL	20	16500
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	11	8500
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	12	9000
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	6	5500
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	9	7700
E083	Patrick	Voltz	MANAGER	HEALTHCARE	15	9500

47 16:13:59 CALL GetExperiencedEmployees() 15 row(s) returned

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'.

```sql
DELIMITER //
CREATE FUNCTION GetStandardRole(exp INT)
RETURNS VARCHAR(30)
DETERMINISTIC
BEGIN
    DECLARE standard_role VARCHAR(30);

    IF exp <= 2 THEN
        SET standard_role = 'JUNIOR DATA SCIENTIST';
    ELSEIF exp > 2 AND exp <= 5 THEN
        SET standard_role = 'ASSOCIATE DATA SCIENTIST';
    ELSEIF exp > 5 AND exp <= 10 THEN
        SET standard_role = 'SENIOR DATA SCIENTIST';
    ELSEIF exp > 10 AND exp <= 12 THEN
        SET standard_role = 'LEAD DATA SCIENTIST';
    ELSEIF exp > 12 AND exp <= 16 THEN
        SET standard_role = 'MANAGER';
    ELSE
        SET standard_role = 'UNDEFINED';
    END IF;
    RETURN standard_role;
END //

DELIMITER ;
```

✓	48	16:19:36	CREATE FUNCTION GetStandardRole(exp INT) RETURNS VARCHAR(30) DETERMINISTIC BEGIN	DEC...	0 row(s) affected

```sql
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    EXP,
    ROLE AS ASSIGNED_ROLE,
    GetStandardRole(EXP) AS STANDARD_ROLE,
    CASE
        WHEN ROLE = GetStandardRole(EXP) THEN 'MATCH'
        ELSE 'MISMATCH'
    END AS STATUS
FROM
    Data_science_team;
```

EMP_ID	FIRST_NAME	LAST_NAME	EXP	ASSIGNED_ROLE	STANDARD_ROLE	STATUS
E005	Eric	Hoffman	11	LEAD DATA SCIENTIST	LEAD DATA SCIENTIST	MATCH
E010	William	Butler	12	LEAD DATA SCIENTIST	LEAD DATA SCIENTIST	MATCH
E052	Dianna	Wilson	6	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST	MATCH
E057	Dorothy	Wilson	9	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST	MATCH
E204	Karene	Nowak	8	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST	MATCH

49 16:22:10 SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, ROLE AS ASSIGNED_ROLE, GetStan... 13 row(s) returned

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.

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

53 16:31:35 CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(20)) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```sql
EXPLAIN SELECT *
FROM emp_record_table
WHERE FIRST_NAME = 'Eric';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	emp_record_table	NULL	ref	idx_first_name	idx_first_name	83	const	1	100.00	Using where

54 16:32:30 EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric' 1 row(s) returned

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).

```sql
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    SALARY,
    EMP_RATING,
    (0.05 * SALARY * EMP_RATING) AS BONUS
FROM
    emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	EMP_RATING	BONUS
E001	Arthur	Black	16500	5	4125.00
E005	Eric	Hoffman	8500	3	1275.00
E010	William	Butler	9000	2	900.00
E052	Dianna	Wilson	5500	5	1375.00

55 16:35:01 SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, (0.05 * SALARY * ... 19 row(s) returned

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

```sql
SELECT
    CONTINENT,
    COUNTRY,
    ROUND(AVG(SALARY),2) AS AVG_SALARY
FROM
    emp_record_table
GROUP BY
    CONTINENT, COUNTRY
ORDER BY
    CONTINENT, COUNTRY;
```

CONTINENT	COUNTRY	AVG_SALARY
ASIA	CHINA	6500.00
ASIA	INDIA	6166.67
EUROPE	FRANCE	9000.00
EUROPE	GERMANY	7600.00
NORTH AMERICA	CANADA	7000.00

57 16:37:10 SELECT CONTINENT, COUNTRY, ROUND(AVG(SALARY),2) AS AVG_SALARY FROM emp_rec... 7 row(s) returned