**ScienceQtech Employee Performance Mapping**

-- Task 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;

-- Task 2 - Create an ER diagram for the given employee database.

-- Task 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;

-- Task 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

select emp\_id , first\_name , last\_name , gender , dept , emp\_rating

from emp\_record\_table

where emp\_rating < 2;

-- greater than four

select emp\_id , first\_name , last\_name , gender , dept , emp\_rating

from emp\_record\_table

where emp\_rating > 4;

-- between two and four

select emp\_id , first\_name , last\_name , gender , dept , emp\_rating

from emp\_record\_table

where emp\_rating between 2 and 4;

-- Task 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,dept

from emp\_record\_table

where dept = "FINANCE";

-- Task 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 employee. EMP\_ID, CONCAT(employee.FIRST\_NAME," ",employee.LAST\_NAME) AS EMPLOYEE\_NAME , manager.MANAGER\_ID,

CONCAT(manager.FIRST\_NAME," ",manager.LAST\_NAME) AS MANAGER\_NAME, manager.ROLE

FROM employee.emp\_record\_table employee

JOIN employee.emp\_record\_table manager

ON employee.MANAGER\_ID = manager.EMP\_ID;

-- Task 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 , dept as department from employee.emp\_record\_table

where dept = "HEALTHCARE" union

select emp\_id,first\_name , last\_name , dept as department from employee.emp\_record\_table

where dept = "FINANCE";

-- Task 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,MAX(EMP\_RATING) FROM employee.emp\_record\_table group by dept;

-- TASK 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 employee.emp\_record\_table group by role;

-- Task 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, LAST\_NAME, ROLE, DEPT, EXP, RANK() OVER (ORDER BY EXP DESC) AS RANKING FROM employee.emp\_record\_table;

-- Task 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.

SELECT\*FROM employee.emp\_record\_table;

CREATE VIEW EMPLOYEE\_SALARY\_VIEW AS SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY, SALARY FROM employee.emp\_record\_table WHERE SALARY > 6000;

SELECT\*FROM EMPLOYEE\_SALARY\_VIEW;

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

SELECT e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, e.EXP, (SELECT COUNT(DISTINCT p.EMP\_ID) FROM employee.emp\_record\_table p) AS EXP1 FROM employee.emp\_record\_table e WHERE e.EXP>10;

-- Task 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 Get\_Employee\_Exp()

BEGIN

SELECT\*FROM employee.emp\_record\_table WHERE EXP > 3;

END $$

CALL Get\_Employee\_Exp;

-- Task 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\_JOB\_PROFILE(EXP int)

RETURNS VARCHAR(100) DETERMINISTIC

BEGIN

DECLARE EMPLOYEE\_JOB\_PROFILE VARCHAR(100);

if EXP <= 2 THEN SET EMPLOYEE\_JOB\_PROFILE = 'JUNIOR DATA SCIENTIST';

elseif EXP BETWEEN 2 AND 5 THEN SET EMPLOYEE\_JOB\_PROFILE = 'ASSOCIATE DATA SCIENTIST';

elseif EXP BETWEEN 5 AND 10 THEN SET EMPLOYEE\_JOB\_PROFILE = 'SENIOR DATA SCIENTIST';

elseif EXP BETWEEN 10 AND 12 THEN SET EMPLOYEE\_JOB\_PROFILE = 'LEAD DATA SCIENTIST';

elseif EXP BETWEEN 12 AND 16 THEN SET EMPLOYEE\_JOB\_PROFILE = 'MANAGER';

END IF;

RETURN (EMPLOYEE\_JOB\_PROFILE);

END $$

SELECT EMP\_ID , FIRST\_NAME , EXP , EMPLOYEE\_JOB\_PROFILE(EXP) FROM employee.emp\_record\_table;

-- Task 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 employee.emp\_record\_table(FIRST\_NAME);

explain select EMP\_ID , FIRST\_NAME , LAST\_NAME FROM employee.emp\_record\_table where FIRST\_NAME = "ERIC";

-- Task 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 , FIRST\_NAME , LAST\_NAME , ROLE, DEPT, EXP, SALARY, EMP\_RATING, (SALARY \* 5/100)\*(EMP\_RATING) AS BONUS FROM employee.emp\_record\_table;

-- Task17: 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, COUNTRY, CONTINENT, AVG(SALARY) AS AVERAGE\_SALARY FROM employee.emp\_record\_table GROUP BY CONTINENT, COUNTRY;