**ScienceQtech 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 schema employee ;

use 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

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 between 2 and 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

m.EMP\_ID, m.FIRST\_NAME, COUNT(r.EMP\_ID) AS NumberofReporters

FROM

emp\_record\_table AS r

INNER JOIN

emp\_record\_table AS m ON m.EMP\_ID = r.MANAGER\_ID

GROUP BY 1, 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 = 'healthcare'

union

select \*

from emp\_record\_table

where DEPT = 'finance' ;

/\*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

FROM emp\_record\_table;

/\*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\_Sal,

Max(SALARY) as Max\_Sal

from emp\_record\_table

group by 1;

/\*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, ROLE, DEPT, EXP,

dense\_rank() OVER(order by EXP desc ) AS 'RankOnExp'

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 Employee\_Country as

Select EMP\_ID, FIRST\_NAME, COUNTRY, SALARY

from emp\_record\_table

where SALARY > 6000;

select \* from employee\_country;

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

Take data from the employee record table.\*/

Select \*

from emp\_record\_table

where EMP\_ID in

(Select EMP\_ID

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 ExpMoreThan3yrs ()

BEGIN

select \* from emp\_record\_table where EXP > 3 ;

END ,,

DELIMITER ;

call ExpMoreThan3yrs ();

/\*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 standard\_job\_profile(EXP INT)

RETURNS VARCHAR(50)

DETERMINISTIC

BEGIN

DECLARE ROLE VARCHAR(50);

IF EXP <= 2 THEN

SET ROLE = 'JUNIOR DATA SCIENTIST';

ELSEIF EXP > 2 AND EXP <= 5 THEN

SET ROLE = 'ASSOCIATE DATA SCIENTIST';

ELSEIF EXP > 5 AND EXP <= 10 THEN

SET ROLE = 'SENIOR DATA SCIENTIST';

ELSEIF EXP > 10 AND EXP <= 12 THEN

SET ROLE = 'LEAD DATA SCIENTIST';

ELSEIF EXP > 12 AND EXP <= 16 THEN

SET ROLE = 'MANAGER';

ELSE

SET ROLE = NULL;

END IF;

RETURN ROLE;

END //

DELIMITER ;

Select

EMP\_ID,

FIRST\_NAME,

ROLE,

standard\_job\_profile (EXP) as 'Standard Profile',

CASE when ROLE = standard\_job\_profile (EXP) then 'MATCH'

else 'NO MATCH'

END as Verified

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.\*/

SELECT \* FROM emp\_record\_table

WHERE FIRST\_NAME = 'eric' ;

CREATE INDEX firstname

ON emp\_record\_table (FIRST\_NAME(50)) ;

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).\*/

Select

EMP\_ID, FIRST\_NAME, ROLE, DEPT, SALARY, EMP\_RATING ,

round(SALARY \* 0.05 \* EMP\_RATING) 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

IFNULL(CONTINENT, 'Total') AS CONTINENT,

IFNULL(COUNTRY, 'Subtotal') AS COUNTRY,

Round(AVG(salary)) AS Avg\_Sal

FROM

emp\_record\_table

GROUP BY

CONTINENT, COUNTRY WITH ROLLUP;