```
-- Science Qtech Employee Performance Mapping --

create database employee;

use employee;

select * from data_science_team;

select * from emp_record_table;

select * from proj_table;

ALTER TABLE `employee`.`data_science_team`

MODIFY COLUMN `EMP_ID` VARCHAR(10) NOT NULL,

ADD PRIMARY KEY (`EMP_ID`);

ALTER TABLE `employee`.`emp_record_table`

MODIFY COLUMN `EMP_ID` VARCHAR(10) NOT NULL,

ADD PRIMARY KEY (`EMP_ID`);

ALTER TABLE `employee`.`proj_table`

MODIFY COLUMN `PROJECT_ID` VARCHAR(10) NOT NULL,

ADD PRIMARY KEY (`PROJECT_ID`);

-- 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;

-- 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 OR EMP_RATING > 4 OR (EMP_RATING >= 2 AND EMP_RATING <= 4);

-- 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';


-- 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, COUNT(r.EMP_ID) AS num_reporters

FROM emp_record_table e

LEFT JOIN emp_record_table r ON e.EMP_ID = r.MANAGER_ID

GROUP BY e.EMP_ID, e.FIRST_NAME, e.LAST_NAME;


-- 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

FROM emp_record_table

WHERE DEPT = 'healthcare'

UNION

SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT

FROM emp_record_table

WHERE DEPT = 'finance';


-- 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 e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE, e.DEPT, e.EMP_RATING, d.MAX_EMP_RATING

FROM emp_record_table e

JOIN (

   SELECT DEPT, MAX(EMP_RATING) AS MAX_EMP_RATING

   FROM emp_record_table

   GROUP BY DEPT

) d ON e.DEPT = d.DEPT;


-- 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;


-- 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, EXP,
    RANK() OVER (ORDER BY EXP DESC) AS experience_rank
FROM emp_record_table;

-- 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_view AS
SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY
FROM emp_record_table
WHERE SALARY > 6000;

select * from high_salary_employees_view;

-- 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 emp_record_table
WHERE EXP > 10;

-- 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 GetExperiencedEmployees()
BEGIN
   SELECT EMP_ID, FIRST_NAME, LAST_NAME
   FROM emp_record_table
   WHERE EXP > 3;
END;
//
DELIMITER ;

-- 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 --
-- TheFor 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'.
```

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
   CASE
      WHEN EXP <= 2 THEN 'JUNIOR DATA SCIENTIST'
      WHEN EXP <= 5 THEN 'ASSOCIATE DATA SCIENTIST'
      WHEN EXP <= 10 THEN 'SENIOR DATA SCIENTIST'
      WHEN EXP <= 12 THEN 'LEAD DATA SCIENTIST'
      ELSE 'MANAGER'
   END AS EMP_RANK
FROM emp_record_table
LIMIT 0, 1000;


-- 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_firstname ON emp_record_table (FIRST_NAME(50));
SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric';


-- 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, SALARY, EMP_RATING,
   0.05 * SALARY * EMP_RATING AS BONUS
FROM emp_record_table;


-- 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;
```