**-- Week 1 : Beginner Level**

**-- Problem 1: Calculate the average monthly salary for each department.**

SELECT

    Department,

    AVG(MonthlySalary) AS Average\_MonthlySalary

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    Department;

This query will return a list of departments and their average monthly salary. In the result set, each row is department and average monthly salary of employees in this department.

**-- Problem 2: Identify the employee with the highest performance score in the IT department.**

SELECT

    Name,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

WHERE

    Department = 'IT'

ORDER BY

    PerformanceScore DESC

LIMIT 1;

This will output the name of IT employee who has highest performance score

**-- Problem 3: Count the number of employees in each department.**

SELECT

    Department,

    COUNT(\*) AS Number\_of\_Employees

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    Department;

The output of this query will be a list department with the count of employees in each department. The result set displays a row for each department showing the count of employees in it.

**-- Problem 4: Find the employee who has been with the company the longest.**

SELECT

    Name,

    HireDate

FROM

    epes.employee\_performance\_evaluation

ORDER BY

    HireDate ASC

LIMIT 1;

This query will return the name and hire date of the employee with longest tenure at your company.

**-- Problem 5: Determine the average performance score across all employees.**

SELECT

    AVG(PerformanceScore) AS Average\_Performance\_Score

FROM

    epes.employee\_performance\_evaluation;

This query will return the average performance score from all employees in our table. Makes a value cut for the average performance score of all workers as One

**-- Problem 6: List all employees who have a performance score above 80.**

SELECT

    Name,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

WHERE

    PerformanceScore > 80;

Output for this query would list down details like name and score of employees who have surpassed a performance rating more than 80.

**-- Problem 7: Calculate the total monthly salary expenditure for the Finance department.**

SELECT

    SUM(MonthlySalary) AS Total\_Monthly\_Salary\_Expenditure

FROM

    epes.employee\_performance\_evaluation

WHERE

    Department = "Finance";

This query will return the total salary cost to this specific department for all employees. This gives you a simple total of the amount designate for all salaries related to that department.

**-- Week 2 : Intermediate Level**

**-- Problem 1: Create a list of employees who have been with the company for more than 10 years but have a performance score below 50.**

SELECT

    Name,

    YearsAtCompany,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

WHERE

    YearsAtCompany > 10

    AND PerformanceScore < 50;

The result of this query will list the names, years of service, and performance scores of employees who meet both conditions: they have been with the company for over 10 years and have a performance score under 50.

**-- Problem 2: Determine the average number of years at the company for each job title.**

SELECT

    JobTitle,

    AVG(YearsAtCompany) AS Average\_Years\_With\_Company

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    JobTitle;

It will return a list of job titles and the average years employed within each title. Each row is made up of a job title and the average time that an employee holds this position for.

**-- Problem 3: Find the top 3 highest-paid employees in the HR department.**

SELECT

    EmployeeID, Name, Department, MonthlySalary

FROM

    epes.employee\_performance\_evaluation

WHERE

    Department = "HR"

ORDER BY

    MonthlySalary DESC

LIMIT 3;

This query will display the employee IDs, names of those top 3 paid employees with their departments and salary will be displayed on a monthly basis in HR department. These top 3 employees are represented in each row.

**-- Problem 4: Calculate the total number of years all employees have worked at the company combined.**

SELECT

    SUM(YearsAtCompany) AS Total\_Years

FROM

    epes.employee\_performance\_evaluation;

This query will return a single value which is the sum of all employee years at company. Overall, this gives you an impression on the total experience of all employees.

**-- Problem 5: Generate a list of employees who were hired before the year 2000.**

SELECT

    EmployeeID, Name, HireDate

FROM

    epes.employee\_performance\_evaluation

WHERE

    HireDate < "2000-01-01";

The query result will display the employee IDs, names and hire dates of employees hired before 2000. Here, each row is a way to satisfy this criterion per employee.

**-- Problem 6: Find the department with the highest average performance score.**

SELECT

    Department

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    Department

ORDER BY

    AVG(PerformanceScore) DESC

LIMIT 1;

It will return the department name which has highest Average Performance Score. With the highest Mean performance score by department.

**-- Problem 7: Analyse the distribution of monthly salaries across different departments and identify any significant disparities.**

SELECT

    Department,

    AVG(MonthlySalary) AS Average\_Salary,

    MIN(MonthlySalary) AS Min\_Salary,

    MAX(MonthlySalary) AS Max\_Salary,

    STDDEV(MonthlySalary) AS Salary\_StandardDeviation

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    Department

ORDER BY

    Average\_Salary DESC;

The output of this query will display every department with:

* Average monthly salary
* Minimum and maximum salaries — per month
* Standard deviation of the monthly wages

**-- Week 3 : Advanced Level**

**-- Problem 1: Identify the top 5 employees with the highest performance scores across all departments and list their department, job title, and years at the company.**

SELECT

    EmployeeID,

    Name,

    Department,

    JobTitle,

    DATEDIFF(CURDATE(), HireDate) / 365 AS Years\_At\_Company,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

ORDER BY

    PerformanceScore DESC

LIMIT 5;

This query would output the top 5 employees with their respective EmployeeID, Name, Department, JobTitle , YearsAtCompany, and PerformanceScore. Each row is a top performing employee, where the data gives insight about their department, role and how long they work there.

**-- Problem 2: Determine the impact of department and job title on performance scores by analysing the average performance score for each combination.**

SELECT

    Department,

    JobTitle,

    AVG(PerformanceScore) AS Average\_Performance\_Score

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    Department, JobTitle

ORDER BY

    Department, JobTitle;

It gives the average performance score at each unique department and job title combined. This row represents specific job title and Department along with average Performance score within it This accommodates for an average cross-section of different roles across each department, so to enable a better in-depth comparison on the impact that department and job title has with employee performance.

**-- Problem 3: Create a performance evaluation report that ranks employees by performance score, including their department, job title, and years at the company.**

SELECT

    EmployeeID,

    Name,

    Department,

    JobTitle,

    DATEDIFF(CURDATE(), HireDate) / 365 AS Years\_At\_Company,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

ORDER BY

    PerformanceScore DESC;

This will return a list of employees ordered by their performance scores. The data contains the ID, name, department, job title and years at company of each employee along with their individual performance score. This report provides a full data of how each employee is performing throughout the organization, as it highlights top performers and allows individual performance comparisons across departments or job titles.

-- Problem 4: Develop a recommendation system to suggest potential promotions based on years at the company, job title, and performance score.

SELECT

    EmployeeID,

    Name,

    Department,

    JobTitle,

    DATEDIFF(CURDATE(), HireDate) / 365 AS Years\_At\_Company,

    PerformanceScore

FROM

    epes.employee\_performance\_evaluation

WHERE

    (DATEDIFF(CURDATE(), HireDate) / 365) >= 5 -- Employees with at least 5 years at the company

  AND PerformanceScore >= 85 -- Employees with a performance score of 85 or above

  AND JobTitle IN ("Sales Manager", "System Administrator", "Mechanical Engineer", "HR Manager", "Recruiter",

  "Customer Service Manager", "Content Creator", "Software Engineer", "Accountant", "Data Scientist",

  "HR Coordinator", "Support Agent", "Sales Representative", "Account Manager", "Network Engineer",

  "Security Analyst", "IT Support", "Marketing Manager", "Marketing Specialist",

  "Finance Manager", "Civil Engineer", "Financial Analyst")

  -- Specific job titles that are considered for promotion

ORDER BY

    PerformanceScore DESC, Years\_At\_Company DESC;

It gives a list of employees eligible to be promoted. Every row is of an employee who:

* On-board for at least 5 years.
* Performance score of 85 or greater, and
* Job titles that may be promoted to one of the specified types.

-- Problem 5: Perform a year-wise analysis of hiring trends, identifying the number of employees hired each year and any notable patterns.

SELECT

    YEAR(HireDate) AS Hire\_Year,

    COUNT(EmployeeID) AS Number\_Of\_Employee\_Hires

FROM

    epes.employee\_performance\_evaluation

GROUP BY

    YEAR(HireDate)

ORDER BY

    Hire\_Year;

This gives an output in which it provides the number of employees are been hired year-wise. Each row in this data set covers a specific year and the number of total employees hired that year.

-- Problem 6: Generate a report to identify employees whose salaries are below the department average and have a high performance score (above 80).

SELECT

    e.EmployeeID,

    e.Name,

    e.Department,

    e.MonthlySalary,

    e.PerformanceScore

FROM

    epes.employee\_performance\_evaluation e

JOIN (

    SELECT

        Department,

        AVG(MonthlySalary) AS Avg\_Department\_Salary

    FROM

        epes.employee\_performance\_evaluation

    GROUP BY

        Department

) Dept\_Avg ON e.Department = Dept\_Avg.Department

WHERE

    e.MonthlySalary < Dept\_Avg.Avg\_Department\_Salary AND e.PerformanceScore > 80

ORDER BY

    e.Department, e.PerformanceScore DESC;

This query will return the employees who:

* Get paid less than their department's average salary for that month
* High performance score - more than 80