1. **How many employees are in the dataset?**

SELECT COUNT(\*) AS TotalEmployees

FROM Employee;

1. **What is the average age of employees?**

SELECT AVG(Age) AS AverageAge

FROM Employee;

1. **What is the distribution of employees by gender?**

SELECT Gender, COUNT(\*) AS Count

FROM Employee

GROUP BY Gender;

1. **What is the most common education level among employees**

SELECT TOP 1 E.Education, EL.EducationLevel, COUNT(\*) AS Count

FROM Employee E

JOIN EducationLevel EL ON E.Education = EL.EducationLevelID

GROUP BY E.Education, EL.EducationLevel

ORDER BY Count DESC;

1. **What is the most common job roles in the company**

SELECT TOP 5 JobRole, COUNT(\*) AS Count

FROM Employee

GROUP BY JobRole

ORDER BY Count DESC;

1. **How many employees left vs. stayed?**

SELECT

CASE

WHEN Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END AS AttritionStatus,

COUNT(\*) AS EmployeeCount

FROM

Employee

GROUP BY

CASE

WHEN Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END;

1. **Average job satisfaction for leavers vs. stayers**

SELECT

CASE

WHEN e.Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END AS AttritionStatus,

AVG(p.JobSatisfaction) AS AvgJobSatisfaction

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

CASE

WHEN e.Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END;

1. **Average manager and self-rating by attrition**

SELECT

CASE

WHEN e.Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END AS AttritionStatus,

AVG(p.ManagerRating) AS AvgManagerRating,

AVG(p.SelfRating) AS AvgSelfRating

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

CASE

WHEN e.Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END;

1. **Attrition Rate by Marital Status**

SELECT

MaritalStatus,

CAST(SUM(CAST(Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

MaritalStatus;

1. **Attrition Rate by Education Level**

SELECT

el.EducationLevel,

CAST(SUM(CAST(e.Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee e

JOIN

EducationLevel el ON e.Education = el.EducationLevelID

GROUP BY

el.EducationLevel;

1. **Relationship Between OverTime and Attrition**

SELECT

OverTime,

COUNT(CASE WHEN Attrition = 1 THEN 1 END) AS AttritionCount,

COUNT(\*) AS TotalEmployees,

COUNT(CASE WHEN Attrition = 1 THEN 1 END) \* 1.0 / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

OverTime;

1. **Average Distance from Home by Attrition**

SELECT

CASE

WHEN Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END AS AttritionStatus,

AVG(DistanceFromHome\_KM) AS AvgDistanceFromHome

FROM

Employee

GROUP BY

CASE

WHEN Attrition = 1 THEN 'Left'

ELSE 'Stayed'

END;

1. **Number of employees in each department**

SELECT

Department,

COUNT(\*) AS EmployeeCount

FROM

Employee

GROUP BY

Department;

1. **Average job satisfaction score per department**

SELECT

e.Department,

AVG(p.JobSatisfaction) AS AvgJobSatisfaction

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

e.Department;

**15. Average manager rating by department**

SELECT

e.Department,

AVG(p.ManagerRating) AS AvgManagerRating

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

e.Department;

1. **Departments with the highest average work-life balance**

SELECT

e.Department,

AVG(p.WorkLifeBalance) AS AvgWorkLifeBalance

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

e.Department

ORDER BY

AvgWorkLifeBalance DESC;

1. **Demographic and Compensation Insights**

SELECT

Gender,

AVG(Salary) AS AvgSalary

FROM

Employee

GROUP BY

Gender;

1. **Average salary by stock option level**

SELECT

StockOptionLevel,

AVG(Salary) AS AvgSalary

FROM

Employee

GROUP BY

StockOptionLevel;

1. **Average environment satisfaction per state**

SELECT

e.State,

AVG(p.EnvironmentSatisfaction) AS AvgEnvironmentSatisfaction

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

e.State;

20. **Number of employees by self-rating performance**

SELECT

SelfRating,

COUNT(\*) AS Count

FROM

PerformanceRating

GROUP BY

SelfRating;

1. **Relationship between training opportunities taken and job satisfaction**

SELECT

TrainingOpportunitiesTaken,

AVG(JobSatisfaction) AS AvgJobSatisfaction

FROM

PerformanceRating

GROUP BY

TrainingOpportunitiesTaken

ORDER BY

TrainingOpportunitiesTaken;

1. **Average salary by gender**

SELECT

Gender,

AVG(Salary) AS AvgSalary

FROM

Employee

GROUP BY

Gender;

1. **Average tenure (YearsAtCompany) for each department**

SELECT

CASE

WHEN Age < 30 THEN '<30'

WHEN Age BETWEEN 30 AND 40 THEN '30-40'

ELSE '>40'

END AS AgeGroup,

CAST(SUM(CAST(Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

CASE

WHEN Age < 30 THEN '<30'

WHEN Age BETWEEN 30 AND 40 THEN '30-40'

ELSE '>40'

END;

1. **Which age group has the highest attrition rate**

SELECT

CASE

WHEN Age < 30 THEN '<30'

WHEN Age BETWEEN 30 AND 40 THEN '30-40'

ELSE '>40'

END AS AgeGroup,

CAST(SUM(CAST(Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

CASE

WHEN Age < 30 THEN '<30'

WHEN Age BETWEEN 30 AND 40 THEN '30-40'

ELSE '>40'

END;

1. **Average time since last promotion by department**

SELECT

Department,

AVG(YearsSinceLastPromotion) AS AvgYearsSinceLastPromotion

FROM

Employee

GROUP BY

Department;

1. **Which job roles have the highest attrition rates**

SELECT

JobRole,

CAST(SUM(CAST(Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

JobRole

ORDER BY

AttritionRate DESC;

1. **Average performance rating (SelfRating) by job role**

SELECT

e.JobRole,

AVG(p.SelfRating) AS AvgSelfRating

FROM

Employee e

JOIN

PerformanceRating p ON e.EmployeeID = p.EmployeeID

GROUP BY

e.JobRole;

**28. What percentage of employees were promoted in the last 3 years**

SELECT

COUNT(CASE WHEN YearsSinceLastPromotion <= 3 THEN 1 END) \* 1.0 / COUNT(\*) AS PromotedInLast3Years

FROM

Employee;

1. **Relationship between promotions and attrition**

SELECT

CASE

WHEN YearsSinceLastPromotion <= 3 THEN 'Promoted Recently'

ELSE 'Not Promoted Recently'

END AS PromotionStatus,

CAST(SUM(CAST(Attrition AS INT)) AS FLOAT) / COUNT(\*) AS AttritionRate

FROM

Employee

GROUP BY

CASE

WHEN YearsSinceLastPromotion <= 3 THEN 'Promoted Recently'

ELSE 'Not Promoted Recently'

END;