1. **Title:** Salary Distribution Across Job Roles

Description: This query analyzes the salary distribution for each job role by calculating the minimum, maximum, and average salaries. The results are grouped by job role and sorted in descending order based on the average salary. This helps in comparing how different roles are compensated within the organization.

```
SELECT Job Role, MIN(Salary) AS MinSalary, MAX(Salary) AS MaxSalary, AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY JobRole
ORDER BY AvgSalary DESC;
```

2. Title: Employee Count by Education Level and Job Role

Description: This query counts the total number of employees for each combination of education level and job role in the "Employees" table. The results are grouped by education level and job role, and ordered first by education level and then by the number of employees in descending order. This helps in understanding the distribution of employees across various education levels and job roles.

```
SELECT Education Level, Job Role, COUNT(*) AS Total Employees
FROM Employees
GROUP BY Education Level, Job Role
ORDER BY EducationLevel, COUNT(*) DESC;
```

3. **Title:** Attrition Rate by Sales Job Role

Description: This query calculates the total number of employees, the number of employees who have left (attrition), and the attrition rate for job roles that contain "Sales" in their title. The data is grouped by job role, and the attrition rate is expressed as a percentage of employees who have left compared to the total number of employees in each "Sales" job role. This helps in understanding employee retention within sales-related positions.

```
SELECT JobRole, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS AttritionRate
FROM Employees
WHERE JobRole LIKE '%Sales%'
GROUP BY JobRole;
```

4. **Title:** Promotions by Ethnicity and Job Role

Description: This query counts the total number of employees and the number of promotions received, grouped by ethnicity and job role. The results are ordered by ethnicity and then by the number of promotions in descending order. This helps in analyzing promotion trends within different ethnic groups and job roles across the organization.

```
SELECT Ethnicity, JobRole, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Promotion = 'Yes' THEN 1 ELSE 0 END) AS PromotionsCount
FROM Employees
GROUP BY Ethnicity, JobRole
ORDER BY Ethnicity, PromotionsCount DESC;
```

5. **Title:** Average Salary by State

Description: This query calculates the average salary for employees in each state, grouped by state. The results are sorted in descending order of average salary, providing insights into regional salary variations across the organization.

```
SELECT State, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY State
ORDER BY AverageSalary DESC;
```

6. **Title:** Average Salary by Years at Company

Description: This query calculates the average salary for employees based on their years of tenure at the company. The results are grouped by the number of years employees have been with the company and sorted in ascending order of tenure, helping to analyze how salaries correlate with length of employment.

```
SELECT YearsAtCompany, AVG(Salary) AS AverageSalary FROM Employees
GROUP BY YearsAtCompany ORDER BY YearsAtCompany;
```

7. **Title:** Attrition Rate Based on Overtime Status

Description: This query calculates the total number of employees, the number of employees who left (attrition), and the attrition rate based on overtime status. The data is grouped by whether employees work overtime or not, with the attrition rate expressed as a percentage of employees who left compared to the total in each overtime category. This helps to understand the relationship between overtime and employee retention.

```
SELECT OverTime, COUNT(*) AS TotalEmployees, SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount, (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS AttritionRate FROM Employees GROUP BY OverTime;
```

8. **Title:** Attrition Rate by Stock Option Level

Description: This query calculates the total number of employees, the number of employees who left (attrition), and the attrition rate based on their stock option level. The data is grouped by stock option level, and the attrition rate is expressed as a percentage of employees who left relative to the total number in each stock option category. This provides insights into the impact of stock options on employee retention.

```
SELECT StockOptionLevel, COUNT(*) AS TotalEmployees, SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount, (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS AttritionRate FROM Employees GROUP BY StockOptionLevel;
```

9. **Title:** Average Salary by Department

Description: This query calculates the average salary for employees within each department, grouping the results by department. The data is ordered in descending order of average salary, providing insights into how different departments are compensated across the organization.

```
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY Department
ORDER BY AverageSalary DESC;
```

10. **Title:** Employee Distribution by Education Field and Job Role

Description: This query counts the total number of employees for each combination of education field and job role. The results are grouped by education field and job role, and ordered by education field and then by the total number of employees in descending order. This helps in understanding how employees are distributed across different educational backgrounds and job roles within the organization.

```
SELECT EducationField, JobRole, COUNT(*) AS TotalEmployees
FROM Employees
GROUP BY EducationField, JobRole
ORDER BY EducationField, COUNT(*) DESC;
```

11. **Title:** Attrition Rate Based on Business Travel Status

Description: This query calculates the total number of employees, the number of employees who left (attrition), and the attrition rate based on business travel status. The results are grouped by whether employees travel for business or not, with the attrition rate represented as a percentage of employees who left relative to the total number in each business travel category. This helps in analyzing the impact of business travel on employee retention.

```
SELECT BusinessTravel, COUNT(*) AS TotalEmployees,
      SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount,
      (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS
AttritionRate
FROM Employees
GROUP BY BusinessTravel;
```

12. **Title:** Attrition Rate by Marital Status

Description: This query calculates the total number of employees, the number of employees who left (attrition), and the attrition rate based on marital status. The data is grouped by marital status, and the attrition rate is calculated as a percentage of employees who left compared to the total number of employees in each marital status category. This helps in examining the relationship between marital status and employee retention.

```
SELECT MaritalStatus, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS
AttritionRate
FROM Employees
GROUP BY MaritalStatus;
```

13. **Title:** Average Salary by Gender

Description: This query calculates the average salary for employees, grouped by gender. It helps to analyze the salary distribution between different gender groups within the organization.

```
SELECT Gender, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY Gender;
```

14. **Title:** Average Salary by Education Level

Description: This query calculates the average salary for employees based on their education level. The results are grouped by education level and ordered in ascending order, providing insights into how compensation varies across different educational qualifications within the organization.

```
SELECT EducationLevel, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY EducationLevel
ORDER BY EducationLevel;
```

15. **Title:** Attrition Rate by Age

Description: This query calculates the total number of employees, the number of employees who left (attrition), and the attrition rate for each age group. The data is grouped by age and ordered in ascending order. The attrition rate is expressed as a percentage of employees who left compared to the total number in each age group, offering insights into how attrition varies across different age ranges.

```
SELECT Age, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS
AttritionRate
FROM Employees
GROUP BY Age
ORDER BY Age;
```

16. **Title:** Employee and Performance Ratings

Description: This SQL query retrieves a comprehensive view of employee performance by selecting key data points from two tables: the `Employee` table (including EmployeeID, JobRole, and Education) and the `PerformanceRating` table (including ManagerRating and SelfRating). By joining these tables based on the EmployeeID, the query allows for an analysis of employee roles, educational background, and performance feedback.

```
SELECT e.EmployeeID, e.JobRole, e.Education, p.ManagerRating, p.SelfRating
FROM Employee e
JOIN PerformanceRating p ON e.EmployeeID = p.EmployeeID;
```

17. **Title:** Analyzing Job Satisfaction Across Departments

Description: This query calculates the average job satisfaction for each department by joining the `Employee`, `Department`, and `PerformanceRating` tables. It groups the results by department name.

```
SELECT d.Dempartment_Name, AVG(p.JobSatisfaction) AS AvgJobSatisfaction
FROM Employee
JOIN Department d ON e.Department = d.Department_ID
JOIN PerformanceRating p ON e.EmployeeID = p.EmployeeID
```

18. **Title:** Job Satisfaction by Training Opportunities

Description: This query calculates the average job satisfaction based on the availability of training opportunities within the year. It groups the data by training opportunities and provides insights into how training impacts employee satisfaction.

```
SELECT TrainingOpportunitiesWithinYear, AVG(JobSatisfaction) AS AvgJobSatisfaction
FROM PerformanceRating
GROUP BY TrainingOpportunitiesWithinYear;
```

19. **Title:** Job Satisfaction by Years at Company

Description: This query calculates the average job satisfaction for employees based on their years at the company by joining the `EmpoYearsInfo` and `PerformanceRating` tables. It helps analyze how tenure impacts job satisfaction

```
SELECT y.YearsAtCompany, AVG(p.JobSatisfaction) AS AvgJobSatisfaction
FROM EmpoYearsInfo y
JOIN PerformanceRating p ON y.EmployeeID = p.EmployeeID
```

20. **Title:** Employee Attrition Analysis

Description: This query retrieves data on employees who have left the company, focusing on their gender, age, job role, overtime status, salary, and attrition status. It helps in understanding key characteristics of employees who experience attrition.

```
SELECT e.Gender, e.Age, e.JobRole, e.OverTime, e.Salary, e.Attrition
FROM Employee e
WHERE e.Attrition = 'Yes';
```

21. Title: Attrition Rate by Job Role

Description:

This query calculates the attrition rate for each job role in the company. It groups employees by their job role, counts the total number of employees per role, calculates how many employees have left (attrition), and then computes the attrition rate as a percentage for each role. This provides insights into the turnover rates across different job functions within the organization.

```
SELECT JobRole, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS
AttritionRate
FROM Employee
GROUP BY JobRole;
```

22. Title: Attrition Count by Job Satisfaction Level

Description:

This query calculates the attrition count for employees based on their job satisfaction level. It joins the `PerformanceRating` and `Employee` tables and groups the results by job satisfaction.

```
SELECT JobSatisfaction, COUNT(*) AS TotalEmployees,
      SUM(CASE WHEN e.Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM PerformanceRating p
JOIN Employee e ON p.EmployeeID = e.EmployeeID
GROUP BY JobSatisfaction;
```

23. Title: Attrition Count by Overtime Status

Description:

This query calculates the attrition count for employees based on their overtime status. It groups the results by whether employees worked overtime and counts the total employees and those who left (attrition) for each group.

```
SELECT OverTime, COUNT(*) AS TotalEmployees,
      SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM Employee
GROUP BY OverTime;
```

24. Title: Average Years for Promotion by Department

Description:

This query calculates the average number of years since the last promotion for employees in each department. It joins the `Employee`, `Department`, and `EmpoYearsInfo` tables, and groups the results by department name.

```
SELECT d.Dempartment_Name, AVG(y.YearsSinceLastPromotion) AS AvgYearsForPromotion
FROM Employee e
JOIN Department d ON e.Department = d.Department_ID
JOIN EmpoYearsInfo y ON e.EmployeeID = y.EmployeeID
GROUP BY d.Dempartment_Name;
```

25. Title: Employees with Long-Term Roles Matching Years at Company

Description:

This query retrieves employees whose years in their most recent role match their total years at the company, with a minimum of 3 years in that role. It joins the `Employee` and `EmpoYearsInfo` tables and filters based on these criteria.

```
SELECT e.EmployeeID, e.JobRole, y.YearsAtCompany, y.YearsInMostRecentRole
FROM Employee e
JOIN EmpoYearsInfo y ON e.EmployeeID = y.EmployeeID
WHERE y.YearsInMostRecentRole = y.YearsAtCompany and YearsInMostRecentRole > 3
```

26. Title: Average Promotion Time by Years with Current Manager

Description:

This query calculates the average time since the last promotion for employees, grouped by the number of years they've been with their current manager. It retrieves data from the `EmpoYearsInfo` table and aggregates the promotion time based on the years spent with the current manager.

```
SELECT y.YearsWithCurrManager, AVG(y.YearsSinceLastPromotion) AS AvgPromotionTime
FROM EmpoYearsInfo y
GROUP BY y.YearsWithCurrManager;
```

27. Title: Average Promotion Time by Manager Rating

Description:

This query calculates the average time since the last promotion for employees, grouped by their manager rating. It joins the `PerformanceRating` and `EmpoYearsInfo` tables to aggregate the promotion time based on the rating given by the employee's manager.

```
SELECT p.ManagerRating, AVG(y.YearsSinceLastPromotion) AS AvgPromotionTime
FROM PerformanceRating p
JOIN EmpoYearsInfo y ON p.EmployeeID = y.EmployeeID
GROUP BY p.ManagerRating;
```

28. Title: Average Salary by Education Level

Description:

This query calculates the average salary of employees based on their education level. It groups the results by education and orders them accordingly to give insights into salary trends for different education categories.

```
SELECT e.Education, AVG(e.Salary) AS AvgSalary
FROM Employee e
GROUP BY e.Education
order by Education
```

29. Title: Employee Retention and Attrition by Stock Option Level

Description:

This query calculates employee retention and attrition based on stock option levels. It groups employees by their stock option level and counts the total number of employees, the number of retained employees (Attrition = 'No'), and the number of employees who left (Attrition = 'Yes').

```
SELECT StockOptionLevel, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'No' THEN 1 ELSE 0 END) AS RetainedEmployees,
        SUM(CASE WHEN Attrition = 'YES' THEN 1 ELSE 0 END) AS AttritionEmployees
FROM Employee
GROUP BY StockOptionLevel;
```

30. Title: Average Salary by Business Travel Status

Description:

This query calculates the average salary of employees based on their business travel status. It groups the results by business travel status, providing insights into salary variations for employees who travel frequently versus those who do not.

```
SELECT BusinessTravel, AVG(Salary) AS AvgSalary
FROM Employee
GROUP BY BusinessTravel;
```

31. Title: Average Salary by State

Description:

This query calculates the average salary of employees based on their state of residence. It groups the results by state, offering insights into salary differences across different geographical locations.

```
SELECT State, AVG(Salary) AS AvgSalary
FROM Employee
GROUP BY State;
```

32. Title: Job Satisfaction by Commuting Distance

Description:

This query examines the relationship between employees' commuting distance and their job satisfaction. It joins the `Emplo_Distance` and `PerformanceRating` tables and calculates the average job satisfaction for employees based on their commuting distance status.

```
SELECT e.DistanceStatus, AVG(p.JobSatisfaction) AS AvgJobSatisfaction
FROM Emplo_Distance e
JOIN PerformanceRating p ON e.EmployeeID = p.EmployeeID
GROUP BY e.DistanceStatus;
```

33. Title: Attrition Count by Commuting Distance Status

Description:

This query calculates the attrition count for employees based on their commuting distance status. It joins the `Emplo_Distance` and `Employee` tables, groups the results by commuting distance status, and counts the number of employees who have left (attrition) for each commuting distance group.

```
SELECT e.DistanceStatus, COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN emp.Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM Emplo_Distance e
JOIN Employee emp ON e.EmployeeID = emp.EmployeeID
GROUP BY e.DistanceStatus;
```

34. Title: Total Employees with Far Commuting Distance by Department

Description:

This query calculates the total number of employees with a "far" commuting distance, grouped by their department. It joins the `Emplo_Distance`, `Employee`, and `Department` tables to retrieve and count the employees who fall under the 'Far' commuting distance status within each department.

```
SELECT d.Dempartment_Name, COUNT(*) AS TotalFarDistantEmployees
FROM Emplo_Distance e
JOIN Employee emp ON e.EmployeeID = emp.EmployeeID
JOIN Department d ON emp.Department = d.Department_ID
WHERE e.DistanceStatus = 'Far'
GROUP BY d.Dempartment_Name;
```