

# Assignment HR Employee Database Analysis

---

## Initial Setup: Create Database and Table

```
-- Create the database
CREATE DATABASE HREmployeeDB;
USE HREmployeeDB;

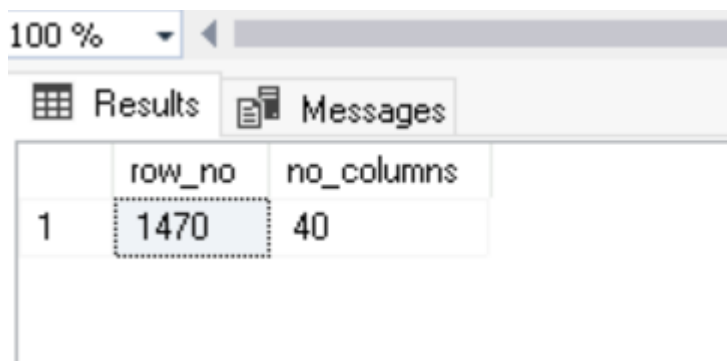
-- Drop the table if it already exists
IF OBJECT_ID('EmployeeData', 'U') IS NOT NULL
    DROP TABLE EmployeeData;

-- Create the EmployeeData table
CREATE TABLE EmployeeData (
    Attrition NVARCHAR(50),
    BusinessTravel NVARCHAR(50),
    CF_age_band NVARCHAR(50),
    CF_attrition_label NVARCHAR(50),
    Department NVARCHAR(50),
    EducationField NVARCHAR(50),
    emp_no NVARCHAR(50),
    EmployeeNumber INT,
    Gender NVARCHAR(50),
    JobRole NVARCHAR(50),
    MaritalStatus NVARCHAR(50),
    OverTime NVARCHAR(50),
    Over18 NVARCHAR(50),
    TrainingTimesLastYear INT,
    Age INT,
    CF_current NVARCHAR(50),
    DailyRate INT,
    DistanceFromHome INT,
    Education NVARCHAR(50),
    EmployeeCount INT,
    EnvironmentSatisfaction INT,
    HourlyRate INT,
    JobInvolvement INT,
    JobLevel INT,
    JobSatisfaction INT,
    MonthlyIncome INT,
    MonthlyRate INT,
    NumCompaniesWorked INT,
    PercentSalaryHike INT,
    PerformanceRating INT,
    RelationshipSatisfaction INT,
    StandardHours INT,
    StockOptionLevel INT,
    TotalWorkingYears INT,
    WorkLifeBalance INT,
    YearsAtCompany INT,
```

```
YearsInCurrentRole INT,  
YearsSinceLastPromotion INT,  
YearsWithCurrManager INT  
);  
  
-- Bulk insert data into the table  
BULK INSERT EmployeeData  
FROM 'C:\Users\Administrator\Downloads\HR_Employee1.csv'  
WITH  
(  
    FIELDTERMINATOR = ',', -- Delimiter for fields  
    ROWTERMINATOR = '0x0a', -- End of each row  
    FIRSTROW = 2           -- Skip header row  
);
```

### a) Return the Shape of the Table (Number of Rows and Columns)

```
-- Query to get the number of rows and columns  
SELECT  
    (SELECT COUNT(*) FROM EmployeeData) AS row_no,  
    (SELECT COUNT(*) FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME =  
    'EmployeeData') AS no_columns;
```



	row_no	no_columns
1	1470	40

Output:

### b) Calculate the Cumulative Sum of Total Working Years for Each Department

```
-- Query to get cumulative sum of total working years for each department  
SELECT  
    Department,  
    TotalWorkingYears,  
    SUM(TotalWorkingYears) OVER (PARTITION BY Department ORDER BY  
    TotalWorkingYears ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS  
    Cumulative_sum_year  
FROM EmployeeData;
```

100 %

Results Messages

	Department	TotalWorkingYears	Cumulative_sum_year
1	HR	1	1
2	HR	1	2
3	HR	1	3
4	HR	1	4
5	HR	2	6
6	HR	2	8
7	HR	3	11
8	HR	3	14
9	HR	4	18
10	HR	4	22
11	HR	4	26
12	HR	5	31
13	HR	6	37
14	HR	6	43
15	HR	6	49
16	HR	6	55
17	HR	6	61
18	HR	6	67
19	HR	6	73
20	HR	6	79

✓ Query executed successfully.

Output:

### c) Which Gender Has Higher Strength as Workforce in Each Department

```
-- Query to determine the gender with higher strength in each department
WITH GenderCounts AS (
    SELECT
        Department,
        Gender,
        COUNT(*) AS Gender_Count
    FROM EmployeeData
    GROUP BY Department, Gender
),
GenderSummary AS (
    SELECT
```

```

        Department,
        MAX(CASE WHEN Gender = 'Male' THEN Gender_Count ELSE 0 END) AS Male_Count,
        MAX(CASE WHEN Gender = 'Female' THEN Gender_Count ELSE 0 END) AS
Female_Count
    FROM GenderCounts
    GROUP BY Department
),
GenderWithHighestCount AS (
    SELECT
        Department,
        CASE
            WHEN Male_Count > Female_Count THEN 'Male'
            WHEN Female_Count > Male_Count THEN 'Female'
            ELSE 'Equal' -- In case of a tie
        END AS Gender_With_Highest_Count
    FROM GenderSummary
)
SELECT
    S.Department,
    S.Male_Count,
    S.Female_Count,
    G.Gender_With_Highest_Count
FROM GenderSummary S
JOIN GenderWithHighestCount G
    ON S.Department = G.Department;
```

Output:

100 %

Results Messages

	Department	Male_Count	Female_Count	Gender_With_Highest_Count
1	HR	43	20	Male
2	R&D	582	379	Male
3	Sales	257	189	Male

Query executed successfully.

## d) Create a New Column **AGE\_BAND** and Show Distribution of Employee's Age Band Group

```
-- Query to add a new column and update it with the count of age bands
```

```
ALTER TABLE EmployeeData
```

```
ADD AGE_BAND INT;
```

```
UPDATE EmployeeData
```

```
SET AGE_BAND = (
```

```
    SELECT COUNT(*)
```

```
    FROM EmployeeData AS ed2
```

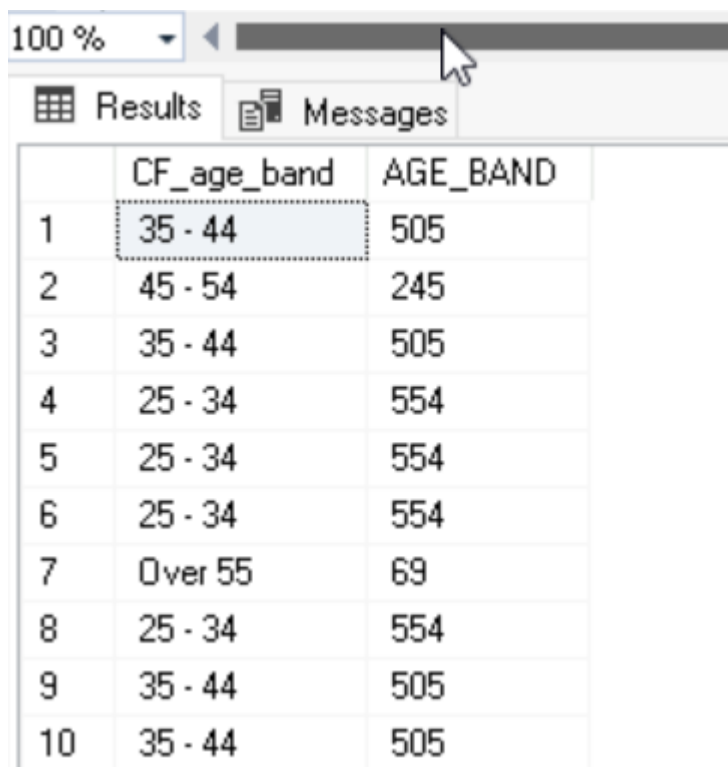
```
    WHERE ed2.CF_age_band = EmployeeData.CF_age_band
```

```
);
```

```
-- Query to display the CF_age_band and AGE_BAND columns
```

```
SELECT CF_age_band, AGE_BAND
```

```
FROM EmployeeData;
```



	CF_age_band	AGE_BAND
1	35 - 44	505
2	45 - 54	245
3	35 - 44	505
4	25 - 34	554
5	25 - 34	554
6	25 - 34	554
7	Over 55	69
8	25 - 34	554
9	35 - 44	505
10	35 - 44	505

**Output:**

## e) Compare All Marital Status of Employees and Find the Most Frequent Marital Status

```
-- Query to get the most frequent marital status
```

```
SELECT
```

```
    TOP(1)
```

```
    MaritalStatus,
```

```
    COUNT(*) AS Marital_Status_Count
```

```
FROM EmployeeData
```

```
GROUP BY MaritalStatus
ORDER BY Marital_Status_Count DESC;
```

Results Messages			
	MaritalStatus	Marital_Status_Count	Freq_Rank
1	Married	673	1

Output:

f) Show the Job Role with Highest Attrition Rate (Percentage)

```
-- Query to calculate attrition rate by job role
WITH AttritionRate AS (
    SELECT
        JobRole,
        CAST((SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0) /
COUNT(*) AS DECIMAL) AS Attrition_Percentage
    FROM EmployeeData
    GROUP BY JobRole
)
SELECT
    JobRole,
    Attrition_Percentage
FROM AttritionRate
WHERE Attrition_Percentage = (SELECT MAX(Attrition_Percentage) FROM
AttritionRate);
```

100 % Results Messages		
	JobRole	Attrition_Rate
1	Sales Representative	40

Output:

g) Show Distribution of Employee's Promotion, Find the Maximum Chances of Employee Getting Promoted

```
-- Query to show promotion chances based on years since last promotion
```

```
SELECT *
FROM EmployeeData
SELECT
    Department,
    CASE
        WHEN YearsSinceLastPromotion = 0 THEN 'Recently Promoted'
        ELSE 'Not Recently Promoted'
    END AS PromotionStatus,
    COUNT(EmployeeNumber) AS NumberOfEmployees,
    AVG(PerformanceRating) AS AvgPerformanceRating,
    AVG(JobInvolvement) AS AvgJobInvolvement,
    AVG(YearsInCurrentRole) AS AvgYearsInCurrentRole,
    AVG(YearsAtCompany) AS AvgYearsAtCompany,
    AVG(TrainingTimesLastYear) AS AvgTrainingTimesLastYear,
    AVG(Age) AS AvgAge,

    AVG(JobSatisfaction) AS AvgJobSatisfaction,
    AVG(MonthlyIncome) AS AvgMonthlyIncome,
    AVG(NumCompaniesWorked) AS AvgNumCompaniesWorked,

    AVG(YearsWithCurrManager) AS AvgYearsWithCurrManager
FROM
    EmployeeData
WHERE
    Attrition = 'No'
GROUP BY
    Department,
    CASE
        WHEN YearsSinceLastPromotion = 0 THEN 'Recently Promoted'
        ELSE 'Not Recently Promoted'
    END;
```

-- Fewer Years in Current Role: Recently promoted employees have fewer years in their current role compared to those not recently promoted, across all departments.

-- Higher Income for Not Promoted: Employees not recently promoted generally have higher average monthly incomes than those recently promoted.

-- Similar Job Involvement: The average job involvement is consistent between recently promoted and not recently promoted employees in all departments.

-- Average Age Difference: Recently promoted employees tend to be younger than those not recently promoted, regardless of department.

Output:

	Department	PromotionStatus	NumberOfEmployees	AvgPerformanceRating	AvgJobInvolvement	AvgYearsInCurrentRole	AvgYearsAtCompany	AvgTrainingTimesLastYear	AvgAge	AvgJobS
1	R&D	Recently Promoted	314	3	2	2	4	2	36	2
2	R&D	Not Recently Promoted	514	3	2	5	8	2	38	2
3	HR	Not Recently Promoted	33	3	2	4	8	2	40	2
4	HR	Recently Promoted	18	3	2	3	7	2	38	2
5	Sales	Not Recently Promoted	215	3	2	5	9	2	38	2
6	Sales	Recently Promoted	139	3	2	3	5	2	35	2

h) Show the Cumulative Sum of Total Working Years for Each Department

```
-- Query to get cumulative sum of total working years for each department
(Duplicate of query b)
SELECT
    Department,
    TotalWorkingYears,
    SUM(TotalWorkingYears) OVER (PARTITION BY Department ORDER BY
TotalWorkingYears ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS
Cumulative_sum_year
FROM EmployeeData;
```

	Department	TotalWorkingYears	Cumulative_sum_year
1	HR	1	1
2	HR	1	2
3	HR	1	3
4	HR	1	4
5	HR	2	6
6	HR	2	8
7	HR	3	11
8	HR	3	14
9	HR	4	18
10	HR	4	22

Output:

i) Find the Rank of Employees Within Each Department Based on Their Monthly Income

```
-- Query to rank employees by monthly income within each department
SELECT
```



```
EmployeeNumber,  
Department,  
MonthlyIncome,  
DENSE_RANK() OVER (PARTITION BY Department ORDER BY MonthlyIncome DESC) AS  
Income_Rank  
FROM EmployeeData;
```

Output:

100 %

Results

Messages

	EmployeeNumber	Department	MonthlyIncome	Income_Rank
1	1338	HR	19717	1
2	1625	HR	19658	2
3	1973	HR	19636	3
4	734	HR	19189	4
5	731	HR	19141	5
6	140	HR	18844	6
7	644	HR	18200	7
8	148	HR	17328	8
9	1408	HR	16799	9
10	1550	HR	16437	10

✓

Query executed successfully.

j) Calculate the Running Total of 'Total Working Years' for Each Employee Within Each Department and Age Band

```
-- Query to calculate running total of total working years within each department  
and age band  
SELECT  
EmployeeNumber,  
Department,  
CF_age_band,  
TotalWorkingYears,  
SUM(TotalWorkingYears)  
OVER (PARTITION BY Department, CF_age_band ORDER BY TotalWorkingYears ROWS  
BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Running_Total_Working_Years  
FROM EmployeeData;
```

**Output:**

100 %

Results

Messages

	EmployeeNumber	Department	CF_age_band	TotalWorkingYears	Running_Total_Working_Years
1	1865	HR	25 - 34	1	1
2	1944	HR	25 - 34	1	2
3	1467	HR	25 - 34	2	4
4	1842	HR	25 - 34	2	6
5	1858	HR	25 - 34	3	9
6	847	HR	25 - 34	4	13
7	1844	HR	25 - 34	4	17
8	1207	HR	25 - 34	5	22
9	1987	HR	25 - 34	6	28
10	869	HR	25 - 34	6	34

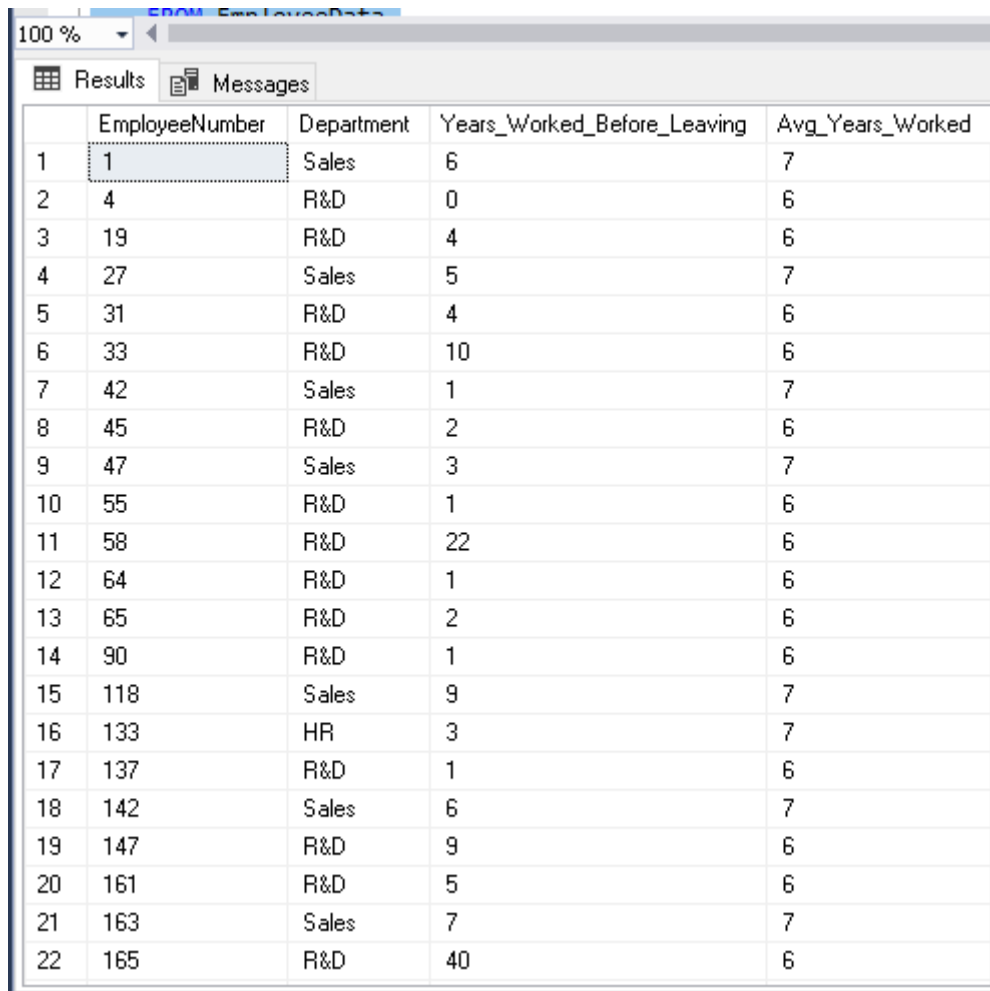
Query executed successfully.

B360E1DFB69950

### k) Calculate the Years Worked Before Leaving and Compare with the Average Years Worked in the Same Department

```
-- Query to calculate years worked before leaving and compare with average years
worked
WITH YearsWorked AS (
    SELECT
        EmployeeNumber,
        Department,
        YearsAtCompany,
        CASE
            WHEN Attrition = 'Yes' THEN YearsAtCompany
            ELSE NULL
        END AS Years_Worked_Before_Leaving
    FROM EmployeeData
),
AvgYears AS (
    SELECT
        Department,
        AVG(YearsAtCompany) AS Avg_Years_Worked
    FROM EmployeeData
    GROUP BY Department
)
SELECT
    Y.EmployeeNumber,
    Y.Department,
    Y.Years_Worked_Before_Leaving,
    A.Avg_Years_Worked
FROM YearsWorked Y
JOIN AvgYears A
    ON Y.Department = A.Department
WHERE Y.Years_Worked_Before_Leaving IS NOT NULL order by EmployeeNumber;
```

```
-- Findings
-- R&D employees often have shorter tenures before leaving compared to Sales, with
fewer years worked on average.
-- HR employees tend to stay longer before leaving compared to R&D and Sales, with
a higher average tenure.
-- Sales employees generally have longer tenures, with more consistent average
years worked before leaving.
```



	EmployeeNumber	Department	Years_Worked_Before_Leaving	Avg_Years_Worked
1	1	Sales	6	7
2	4	R&D	0	6
3	19	R&D	4	6
4	27	Sales	5	7
5	31	R&D	4	6
6	33	R&D	10	6
7	42	Sales	1	7
8	45	R&D	2	6
9	47	Sales	3	7
10	55	R&D	1	6
11	58	R&D	22	6
12	64	R&D	1	6
13	65	R&D	2	6
14	90	R&D	1	6
15	118	Sales	9	7
16	133	HR	3	7
17	137	R&D	1	6
18	142	Sales	6	7
19	147	R&D	9	6
20	161	R&D	5	6
21	163	Sales	7	7
22	165	R&D	40	6

**Output:**

## I) Rank the Departments by the Average Monthly Income of Employees Who Have Left

```
-- Query to rank departments by average monthly income of employees who have left
SELECT Department,
       AVG(MonthlyIncome) AS Avg_Monthly_Income,
       RANK

```markdown
       RANK() OVER (ORDER BY AVG(MonthlyIncome) DESC) AS Department_Rank
FROM EmployeeData
```

```
WHERE Attrition = 'Yes'
GROUP BY Department;
```

Output:

	Department	Monthly_Income	Monthly_Income_Ranking
1	Sales	5908	1
2	R&D	4108	2
3	HR	3715	3

m) Find If There Is Any Relation Between Attrition Rate and Marital Status of Employees

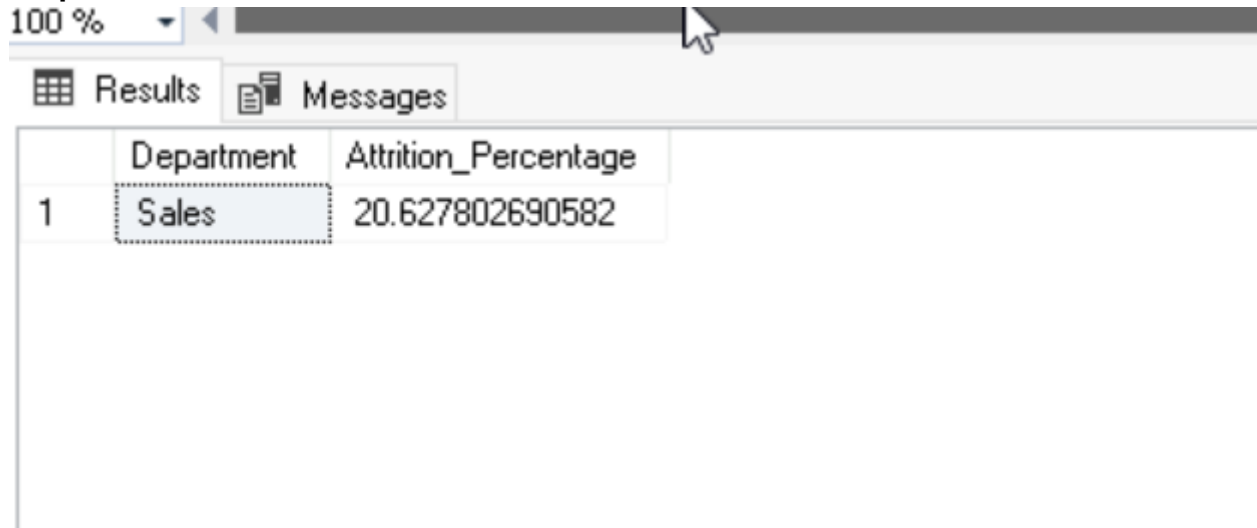
```
-- Calculate attrition percentage by marital status
SELECT
    MaritalStatus,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0) / COUNT(*) AS
    Attrition_Percentage
FROM EmployeeData
GROUP BY MaritalStatus;
```

	MaritalStatus	Attrition_Percentage
1	Single	25.531914893617
2	Divorced	10.091743119266
3	Married	12.481426448736

Output:

n) Show the Department with Highest Attrition Rate (Percentage)

```
-- Find the department with the highest attrition rate
SELECT TOP(1)
    Department,
    (SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0) / COUNT(*) AS
Attrition_Percentage
FROM EmployeeData
GROUP BY Department
ORDER BY Attrition_Percentage DESC;
```

**Output:**

	Department	Attrition_Percentage
1	Sales	20.627802690582

---

### o) Calculate the Moving Average of Monthly Income Over the Past 3 Employees for Each Job Role

```
-- Calculate moving average of monthly income over the past 3 employees for each
job role
SELECT
    JobRole,
    EmployeeNumber,
    MonthlyIncome,
    AVG(MonthlyIncome) OVER (PARTITION BY JobRole ORDER BY EmployeeNumber ROWS
BETWEEN 2 PRECEDING AND CURRENT ROW) AS Moving_Avg_Income
FROM EmployeeData;
```

**Output:**

	JobRole	EmployeeNumber	MonthlyIncome	Moving_Avg_Income
1	Healthcare Representative	13	5237	5237
2	Healthcare Representative	36	10248	7742
3	Healthcare Representative	40	6465	7316
4	Healthcare Representative	70	9884	8865
5	Healthcare Representative	83	10096	8815
6	Healthcare Representative	117	4152	8044
7	Healthcare Representative	119	13503	9250
8	Healthcare Representative	124	10673	9442
9	Healthcare Representative	139	5163	9779
10	Healthcare Representative	145	7484	7773

## p) Identify Employees with Outliers in Monthly Income Within Each Job Role

```
-- Identify employees with outliers in monthly income
WITH IncomeStats AS (
    SELECT
        JobRole,
        EmployeeNumber,
        MonthlyIncome,
        PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q1,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q3
    FROM EmployeeData
)
SELECT
    EmployeeNumber,
    JobRole,
    MonthlyIncome,
    CASE
        WHEN MonthlyIncome < Q1 - (Q3 - Q1) * 1.5 THEN 'Low'
        WHEN MonthlyIncome > Q3 + (Q3 - Q1) * 1.5 THEN 'High'
        ELSE 'Not an Outlier'
    END AS OutlierType
FROM IncomeStats
WHERE MonthlyIncome < Q1 - (Q3 - Q1) * 1.5
    OR MonthlyIncome > Q3 + (Q3 - Q1) * 1.5;
```

Output:

Results

Messages

	EmployeeNumber	JobRole	MonthlyIncome	OutlierType
1	1985	Laboratory Technician	6323	High
2	1737	Laboratory Technician	6472	High
3	1315	Laboratory Technician	6674	High
4	1132	Laboratory Technician	6782	High
5	944	Laboratory Technician	7403	High
6	1516	Manager	11244	Low
7	613	Manager	11557	Low
8	153	Manager	11631	Low
9	376	Manager	11849	Low
10	428	Manager	11878	Low

Query executed successfully.

q) Gender Distribution Within Each Job Role, Show Each Job Role with Its Gender Domination

```
-- Show gender distribution within each job role
WITH GenderCount AS (
    SELECT
        JobRole,
        Gender,
        COUNT(*) AS Gender_Count
    FROM EmployeeData
    GROUP BY JobRole, Gender
),
MaxGender AS (
    SELECT
        JobRole,
        MAX(Gender_Count) AS Max_Count
    FROM GenderCount
    GROUP BY JobRole
)
SELECT
    G.JobRole,
    G.Gender,
    G.Gender_Count,
    CASE
        WHEN G.Gender_Count = M.Max_Count THEN
            CASE
                WHEN G.Gender = 'Male' THEN 'Male_Domination'
                WHEN G.Gender = 'Female' THEN 'Female_Domination'
```

```
        ELSE 'No_Domination'
      END
    ELSE 'No_Domination'
  END AS Gender_Domination
FROM GenderCount G
JOIN MaxGender M
  ON G.JobRole = M.JobRole AND G.Gender_Count = M.Max_Count;
```

Output:

100 %

ResultsMessages

	JobRole	Gender	Gender_Count	Gender_Domination
1	Sales Representative	Male	45	Male_Domination
2	Sales Executive	Male	194	Male_Domination
3	Research Scientist	Male	178	Male_Domination
4	Research Director	Male	47	Male_Domination
5	Manufacturing Director	Male	73	Male_Domination
6	Manager	Male	55	Male_Domination
7	Laboratory Technician	Male	174	Male_Domination
8	Human Resources	Male	36	Male_Domination
9	Healthcare Representative	Male	80	Male_Domination

Query executed successfully

r) Percent Rank of Employees Based on Training Times Last Year

```
-- Calculate percent rank of employees based on training times last year
SELECT
  EmployeeNumber,
  TrainingTimesLastYear,
  PERCENT_RANK() OVER (ORDER BY TrainingTimesLastYear) AS Percent_Rank
FROM EmployeeData;
```



.00 %

Results Messages

	EmployeeNumber	TrainingTimesLastYear	Percent_Rank
1	1	0	0
2	56	0	0
3	58	0	0
4	90	0	0
5	125	0	0
6	178	0	0
7	246	0	0
8	275	0	0
9	353	0	0
10	361	0	0

Output:

## s) Divide Employees into 5 Groups Based on Training Times Last Year

```
-- Divide employees into 5 groups based on training times last year
SELECT
  EmployeeNumber,
  TrainingTimesLastYear,
  NTILE(5) OVER (ORDER BY TrainingTimesLastYear) AS Training_Group
FROM EmployeeData;
```

100 %

Results		Messages	
	EmployeeNumber	TrainingTimesLastYear	Training_Group
1	1	0	1
2	56	0	1
3	58	0	1
4	90	0	1
5	125	0	1
6	178	0	1
7	246	0	1
8	275	0	1
9	353	0	1
10	361	0	1

✓ Query executed successfully.

Output:


t) Categorize Employees Based on Training Times Last Year as Frequent Trainee, Moderate Trainee, Infrequent Trainee

```
-- Categorize employees based on training times last year
SELECT
  emp_no,
  TrainingTimesLastYear,
  CASE
    WHEN TrainingTimesLastYear > 4 THEN 'Frequent Trainee'
    WHEN TrainingTimesLastYear > 2 THEN 'Moderate Trainee'
    ELSE 'Infrequent Trainee'
  END AS Training_Frequency
FROM EmployeeData
ORDER BY TrainingTimesLastYear DESC;
```

Output:

column PerformanceRating(int, null)


	emp_no	TrainingTimesLastYear	Training Frequency
1	STAFF-30	6	Frequent Trainee
2	STAFF-42	6	Frequent Trainee
3	STAFF-54	6	Frequent Trainee
4	STAFF-153	6	Frequent Trainee
5	STAFF-154	6	Frequent Trainee
6	STAFF-217	6	Frequent Trainee
7	STAFF-258	6	Frequent Trainee
8	STAFF-307	6	Frequent Trainee
9	STAFF-309	6	Frequent Trainee
10	STAFF-381	6	Frequent Trainee

 Query executed successfully.

u) Categorize Employees as 'High', 'Medium', or 'Low' Performers Based on Their Performance Rating

```
-- Categorize employees based on performance rating
SELECT
  EmployeeNumber,
  PerformanceRating,
  CASE
    WHEN PerformanceRating >= 4 THEN 'High'
    WHEN PerformanceRating = 3 THEN 'Medium'
    ELSE 'Low'
  END AS Performance_Category
FROM EmployeeData;
```

Results		Messages	
	EmployeeNumber	PerformanceRating	Performance_Category
1	1	3	Medium
2	2	4	High
3	4	3	Medium
4	5	3	Medium
5	7	3	Medium
6	8	3	Medium
7	10	4	High
8	11	4	High
9	12	4	High
10	13	3	Medium

Output:  Query executed successfully.

v) Use a CASE WHEN Statement to Categorize Employees into 'Poor', 'Fair', 'Good', or 'Excellent' Work-Life Balance Based on Their Work-Life Balance Score

```
-- Categorize employees based on work-life balance score
SELECT
    EmployeeNumber,
    WorkLifeBalance,
    CASE
        WHEN WorkLifeBalance = 1 THEN 'Poor'
        WHEN WorkLifeBalance = 2 THEN 'Fair'
        WHEN WorkLifeBalance = 3 THEN 'Good'
        ELSE 'Excellent'
    END AS Work_Life_Balance_Category
FROM EmployeeData;
```

**Output:**

Results		Messages	
	EmployeeNumber	WorkLifeBalance	Work_Life_Balance_Category
1	1	1	Poor
2	2	3	Good
3	4	3	Good
4	5	3	Good
5	7	3	Good
6	8	2	Fair
7	10	2	Fair
8	11	3	Good
9	12	3	Good
10	13	2	Fair


✓ Query executed successfully.

### w) Group Employees into 3 Groups Based on Their Stock Option Level Using the NTILE Function

```
-- Group employees into 3 groups based on stock option level
SELECT
    EmployeeNumber,
    StockOptionLevel,
    NTILE(3) OVER (ORDER BY StockOptionLevel) AS StockOption_Group
FROM EmployeeData;
```

100 %

Results		Messages	
	EmployeeNumber	StockOptionLevel	StockOption_Group
1	1	0	1
2	4	0	1
3	5	0	1
4	8	0	1
5	12	0	1
6	15	0	1
7	19	0	1
8	27	0	1
9	28	0	1
10	30	0	1

Output:  Query executed successfully.

## x) Find Key Reasons for Attrition in Company

```
-- Analyze key reasons for attrition in the company
SELECT
  Attrition,
  YearsAtCompany,
  COUNT(*) AS Total_Employees,
  AVG(Age) AS Avg_Age,
  AVG(MonthlyIncome) AS Avg_MonthlyIncome,
  AVG(WorkLifeBalance) AS Avg_WorkLifeBalance,
  AVG(JobSatisfaction) AS Avg_JobSatisfaction,
  AVG(EnvironmentSatisfaction) AS Avg_EnvironmentSatisfaction,
  AVG(JobInvolvement) AS Avg_JobInvolvement,
  AVG(PerformanceRating) AS Avg_PerformanceRating
FROM EmployeeData
WHERE Attrition = 'Yes'
GROUP BY Attrition, YearsAtCompany
ORDER BY Attrition, YearsAtCompany, Total_Employees DESC;

-- Findings
-- Sales and R&D departments have the highest attrition rates.
-- Sales Executives and Research Scientists are the most affected job roles.
-- Single(marital status) employees have a higher likelihood of leaving.
-- Overtime is a significant factor contributing to higher attrition.
-- Younger employees are more likely to leave.
-- Employees with less experience tend to leave more frequently.
```

Output:

ResultsMessages

	Department	EducationField	Gender	JobRole	MaritalStatus	OverTime	AttritionCount	AvgAgeAttrition	AvgMonthlyIncomeAttrition	AvgYearsAtCompany	AvgDistanceFrom	
1	R&D	Life Sciences	Male	Laboratory Technician	Single	No	8	33	2322	2	11	
2	R&D	Life Sciences	Male	Laboratory Technician	Single	Yes	6	35	3695	3	10	
3	Sales	Marketing	Male	Sales Executive	Single	Yes	6	35	8262	7	11	
4	R&D	Life Sciences	Female	Research Scientist	Single	Yes	5	29	2673	5	9	
5	R&D	Life Sciences	Male	Research Scientist	Single	Yes	4	32	2798	4	11	
6	R&D	Medical	Female	Healthcare Representative	Married	No	4	46	7998	15	17	
7	R&D	Medical	Male	Research Scientist	Single	Yes	4	28	2008	2	15	
8	R&D	Medical	Male	Research Scientist	Married	Yes	4	29	3483	2	3	
9	R&D	Technical Degree	Male	Research Scientist	Single	No	4	35	2500	5	1	
10	Sales	Life Sciences	Female	Sales Executive	Single	Yes	4	40	8537	3	17	
11	Sales	Life Sciences	Male	Sales Executive	Single	Yes	4	45	9616	10	17	
12	Sales	Life Sciences	Male	Sales Executive	Married	No	3	38	11086	15	3	
13	Sales	Life Sciences	Female	Sales Representative	Single	No	3	27	2286	1	5	
14	Sales	Life Sciences	Female	Sales Executive	Married	No	3	40	6312	6	19	
15	Sales	Marketing	Female	Sales Executive	Single	No	3	35	5788	4	17	
16	Sales	Marketing	Female	Sales Representative	Single	Yes	3	31	2771	2	11	