

ASSIGNMENT 3

Create database HREmployeeDB.

- a) Return the shape of the table
- b) Calculate the cumulative sum of total working years for each department
- c) Which gender have higher strength as workforce in each department
- d) Create a new column AGE_BAND and Show Distribution of Employee's Age band group
(Below 25, 25-34, 35-44, 45-55. ABOVE 55).
- e) Compare all marital status of employee and find the most frequent marital status
- f) Show the Job Role with Highest Attrition Rate (Percentage)
- g) Show distribution of Employee's Promotion, Find the maximum chances of employee getting promoted.
- h) Show the cumulative sum of total working years for each department.
- i) Find the rank of employees within each department based on their monthly income
- j) Calculate the running total of 'Total Working Years' for each employee within each department and age band.
- k) Foreach employee who left, calculate the number of years they worked before leaving and
compare it with the average years worked by employees in the same department.
- l) Rank the departments by the average monthly income of employees who have left.
- m) Find the if there is any relation between Attrition Rate and Marital Status of Employee.
- n) Show the Department with Highest Attrition Rate (Percentage)
- o) Calculate the moving average of monthly income over the past 3 employees for each job
role.
- p) Identify employees with outliers in monthly income within each job role. [Condition :
 $\text{Monthly_Income} < Q1 - (Q3 - Q1) * 1.5$ OR $\text{Monthly_Income} > Q3 + (Q3 - Q1)$]
- q) Gender distribution within each job role, show each job role with its gender
domination.
[Male_Domination or Female_Domination]
- r) Percent rank of employees based on training times last year
- s) Divide employees into 5 groups based on training times last year [Use NTILE ()]
- t) Categorize employees based on training times last year as - Frequent Trainee,
Moderate
Trainee, Infrequent Trainee.

u) Categorize employees as 'High', 'Medium', or 'Low' performers based on their performance

rating, using a CASE WHEN statement.

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.

w) Group employees into 3 groups based on their stock option level using the [NTILE] function.

x) Find key reasons for Attrition in Company

SOLUTION

```
CREATE DATABASE HRdata;SELECT Department, TotalWorkingYears,  
SUM(TotalWorkingYears) OVER (PARTITION BY Department  
ORDER BY TotalWorkingYears ROWS BETWEEN UNBOUNDED PRECEDING  
AND CURRENT ROW) AS TotalWorkYrSum  
FROM EmployeeData  
WHERE TotalWorkingYears > 0;  
SELECT Department, TotalWorkingYears,  
SUM(TotalWorkingYears) OVER (PARTITION BY Department  
ORDER BY TotalWorkingYears ROWS BETWEEN UNBOUNDED PRECEDING  
AND CURRENT ROW) AS TotalWorkYrSum  
FROM EmployeeData  
WHERE TotalWorkingYears > 0;
```

USE HRdata;

```
CREATE TABLE EmployeeData (  
    Attrition VARCHAR(3),  
    BusinessTravel VARCHAR(20),  
    CFAgeBand VARCHAR(20),  
    CFAttritionLabel VARCHAR(20),  
    Department VARCHAR(50),  
    EducationField VARCHAR(50),  
    EmpNo VARCHAR(20),  
    EmployeeNumber INT,  
    Gender VARCHAR(10),  
    JobRole VARCHAR(50),  
    MaritalStatus VARCHAR(20),  
    OverTime VARCHAR(3),  
    Over18 VARCHAR(3),  
    TrainingTimesLastYear INT,  
    Age INT,  
    CFCurrentEmployee VARCHAR(3),  
    DailyRate INT,  
    DistanceFromHome INT,
```

```

Education VARCHAR(20),
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 EmployeeData FROM 'D:/HR Employee - HR data.csv'
WITH
(
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '0x0a',
    FIRSTROW=2
);

select * from EmployeeData;

```

a) Return the shape of the table

```

SELECT COUNT(*) AS NumOfRows FROM EmployeeData;

SELECT COUNT(*) AS NumOfColumns FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'EmployeeData';

```

Results		Messages
	NumOfRows	
1	1470	
	NumOfColumns	
1	39	

b) Calculate the 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 TotalWorkYrSum
FROM EmployeeData
WHERE TotalWorkingYears > 0;
```

Results

Messages

	Department	TotalWorkingYears	TotalWorkYrSum
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
21	HR	7	86
22	HR	7	93

c) Which gender have higher strength as workforce in each department

```
SELECT Department, Gender, Gender_count
FROM (
SELECT Department, Gender, COUNT(*) AS Gender_count,
RANK() OVER (PARTITION BY Department ORDER BY COUNT(*) DESC) AS
GenderRank
FROM EmployeeData
GROUP BY Department, Gender
) AS GenderRanks
WHERE GenderRank = 1;
```

	Department	Gender	Gender_count
1	HR	Male	43
2	R&D	Male	582
3	Sales	Male	257

d) Create a new column AGE_BAND and Show Distribution of Employee's Age band group (Below 25, 25-34, 35-44, 45-55. ABOVE 55).

```
ALTER TABLE EmployeeData
ADD AgeBand INT;
```

```
UPDATE EmployeeData
SET AgeBand = (
SELECT COUNT(*)
FROM EmployeeData AS ed2
WHERE ed2.CFAgeBand = EmployeeData.CFAgeBand
);
```

```
SELECT CFAgeBand, COUNT(*) AS AgeBA
FROM EmployeeData
GROUP BY CFAgeBand;
```

Results Messages		
	CFAgeBand	AgeBA
1	Over 55	69
2	35 - 44	505
3	45 - 54	245
4	25 - 34	554
5	Under 25	97

e) Compare all marital status of employee and find the most frequent marital status

```
SELECT TOP(1) MaritalStatus,COUNT(MaritalStatus) AS Frequency FROM EmployeeData
GROUP BY MaritalStatus
ORDER BY Frequency DESC;
```

Results Messages		
	MaritalStatus	Frequency
1	Married	673

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

```
WITH AttrRate AS (
SELECT JobRole,
SUM(case
when attrition = 'Yes' then 1
else 0
end) * 100.0 / COUNT(*) AS Attrperc
FROM EmployeeData
GROUP BY JobRole
)
SELECT TOP(1) JobRole, Attrperc
FROM AttrRate
```

ORDER BY Attrperc DESC;

Results		Messages
	JobRole	Attrperc
1	Sales Representative	39.759036144578

g) Show distribution of Employee's Promotion, Find the maximum chances of employee getting promoted.

```
SELECT JobRole,  
AVG(YearsSinceLastPromotion) AS avg_years_since_last_promotion,  
AVG(YearsAtCompany) AS AvgYearsAtCompany,  
AVG(PerformanceRating) AS AvgPerformanceRating,  
COUNT(*) AS TotalEmployees  
FROM EmployeeData  
WHERE YearsSinceLastPromotion IS NOT NULL  
GROUP BY JobRole,PerformanceRating  
ORDER BY PerformanceRating DESC, AvgYearsAtCompany DESC,  
avg_years_since_last_promotion DESC;
```

--INSIGHT - Employees with high ~~average~~ years since last promotion, more working years, and high performance rating has the highest chance of getting promoted.

Results		Messages			
	JobRole	avg_years_since_last_promotion	AvgYearsAtCompany	AvgPerformanceRating	TotalEmployees
1	Manager	4	11	4	20
2	Healthcare Representative	5	10	4	20
3	Research Director	3	10	4	8
4	Sales Executive	3	8	4	41
5	Manufacturing Director	1	7	4	27
6	Research Scientist	1	5	4	49
7	Human Resources	0	5	4	7
8	Sales Representative	1	4	4	12
9	Laboratory Technician	0	3	4	42
10	Manager	5	15	3	82
11	Research Director	3	10	3	72
12	Sales Executive	2	7	3	285
13	Manufacturing Director	2	7	3	118
14	Healthcare Representative	2	7	3	111
15	Laboratory Technician	1	5	3	217
16	Research Scientist	1	5	3	243
17	Human Resources	1	5	3	45
18	Sales Representative	0	2	3	71

i) Find the rank of employees within each department based on their monthly income

```
SELECT EmpNo, department, MonthlyIncome,
       RANK() OVER (PARTITION BY department ORDER BY MonthlyIncome desc) AS rank
FROM EmployeeData;
```

Results		Messages		
	EmpNo	department	MonthlyIncome	rank
1	STAFF-1338	HR	19717	1
2	STAFF-1625	HR	19658	2
3	STAFF-1973	HR	19636	3
4	STAFF-734	HR	19189	4
5	STAFF-731	HR	19141	5
6	STAFF-140	HR	18844	6
7	STAFF-644	HR	18200	7
8	STAFF-148	HR	17328	8
9	STAFF-1408	HR	16799	9
10	STAFF-1550	HR	16437	10
11	STAFF-1352	HR	14026	11
12	STAFF-698	HR	10725	12
13	STAFF-1098	HR	10482	13
14	STAFF-590	HR	9950	14
15	STAFF-1744	HR	9756	15
16	STAFF-2040	HR	8837	16
17	STAFF-1419	HR	7988	17
18	STAFF-1722	HR	6430	18
19	STAFF-424	HR	6410	19
20	STAFF-760	HR	6389	20
21	STAFF-184	HR	6347	21
22	STAFF-1231	HR	6272	22

j) Calculate the running total of 'Total Working Years' for each employee within each department and age band.

```
SELECT Department, CFAgeBand, TotalWorkingYears,
SUM(TotalWorkingYears) OVER (PARTITION BY Department, CFAgeBand
ORDER BY TotalWorkingYears ROWS BETWEEN UNBOUNDED PRECEDING
AND CURRENT ROW) AS TotalWorkYrSum
```

```
FROM EmployeeData
WHERE TotalWorkingYears > 0;
```

Results		Messages		
	Department	CFAgeBand	TotalWorkingYears	TotalWorkYrSum
1	HR	25 - 34	1	1
2	HR	25 - 34	1	2
3	HR	25 - 34	2	4
4	HR	25 - 34	2	6
5	HR	25 - 34	3	9
6	HR	25 - 34	4	13
7	HR	25 - 34	4	17
8	HR	25 - 34	5	22
9	HR	25 - 34	6	28
10	HR	25 - 34	6	34
11	HR	25 - 34	6	40
12	HR	25 - 34	6	46
13	HR	25 - 34	6	52
14	HR	25 - 34	7	59
15	HR	25 - 34	8	67
16	HR	25 - 34	8	75
17	HR	25 - 34	8	83
18	HR	25 - 34	9	92
19	HR	25 - 34	10	102
20	HR	25 - 34	11	113
21	HR	25 - 34	12	125
22	HR	25 - 34	14	139

k) For each employee who left, calculate the number of years they worked before leaving and compare it with the average years worked by employees in the same department.

```
SELECT EmpNo, dept.Department, YearsAtCompany, AvgYears
FROM EmployeeData
LEFT JOIN (
SELECT Department, AVG(YearsAtCompany) AS AvgYears
FROM EmployeeData
```

GROUP BY Department
) AS dept
 ON dept.Department = EmployeeData.Department
 ORDER BY EmpNo;

	EmpNo	Department	YearsAtCompany	AvgYears
1	STAFF-1	Sales	6	7
2	STAFF-10	R&D	1	6
3	STAFF-100	Sales	15	7
4	STAFF-1001	R&D	2	6
5	STAFF-1002	R&D	9	6
6	STAFF-1003	Sales	9	7
7	STAFF-1004	R&D	5	6
8	STAFF-1005	R&D	12	6
9	STAFF-1006	R&D	2	6
10	STAFF-1007	R&D	8	6
11	STAFF-1009	R&D	4	6
12	STAFF-101	R&D	0	6
13	STAFF-1010	R&D	4	6
14	STAFF-1011	R&D	4	6
15	STAFF-1012	R&D	0	6
16	STAFF-1013	R&D	10	6
17	STAFF-1014	R&D	16	6
18	STAFF-1015	R&D	8	6
19	STAFF-1016	R&D	1	6
20	STAFF-1017	R&D	3	6
21	STAFF-1018	R&D	8	6
22	STAFF-1019	R&D	4	6

I) Rank the departments by the average monthly income of employees who have left

```
SELECT Department,avg(MonthlyIncome) AvgMonthlyIncome,
RANK() OVER(ORDER BY avg(MonthlyIncome) DESC)
AS IncomeRank
FROM EmployeeData
WHERE Attrition = 'Yes'
GROUP BY Department
```

Results		Messages	
	Department	AvgMonthlyIncome	IncomeRank
1	Sales	5908	1
2	R&D	4108	2
3	HR	3715	3

m) Find the if there is any relation between Attrition Rate and Marital Status of Employee.

```
SELECT MaritalStatus, COUNT(*) AS MAR_COUNT,
SUM(CASE WHEN attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
attritionpercent
FROM EmployeeData
GROUP BY MaritalStatus
ORDER BY MAR_COUNT DESC;
```

	MaritalStatus	MAR_COUNT	attritionpercent
1	Married	673	12.481426448736
2	Single	470	25.531914893617
3	Divorced	327	10.091743119266

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

```
WITH attritionRate AS (
  SELECT Department,
    SUM(CASE WHEN attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
attritionPCT
  FROM EmployeeData
  GROUP BY Department)
```

```
)
SELECT TOP 1 Department, attritionPCT
FROM attritionRate
ORDER BY attritionPCT DESC;
```

Results		Messages
	Department	attritionPCT
1	Sales	20.627802690582

--Insight - Sales department has the highest attrition rate in the company.

o)Calculate the moving average of monthly income over the past 3 employees for each job role.

```
SELECT EmpNo, MonthlyIncome,
AVG(MonthlyIncome) OVER (PARTITION BY JobRole ORDER BY MonthlyIncome ROWS
BETWEEN 3 PRECEDING AND
CURRENT ROW) AS Moving_average_income
FROM EmployeeData;
```

Results		Messages	
	EmpNo	MonthlyIncome	Moving_average_income
1	STAFF-369	4000	4000
2	STAFF-942	4014	4007
3	STAFF-1501	4035	4016
4	STAFF-1552	4069	4029
5	STAFF-659	4089	4051
6	STAFF-1062	4107	4075
7	STAFF-1630	4148	4103
8	STAFF-117	4152	4124
9	STAFF-1022	4240	4161
10	STAFF-807	4244	4196
11	STAFF-793	4335	4242
12	STAFF-1829	4448	4316
13	STAFF-526	4522	4387
14	STAFF-551	4523	4457
15	STAFF-721	4553	4511
16	STAFF-1981	4617	4553
17	STAFF-390	4741	4608
18	STAFF-1033	4777	4672
19	STAFF-287	4876	4752
20	STAFF-2009	4878	4818
21	STAFF-1903	5033	4891
22	STAFF-1597	5063	4962

p) Identify employees with outliers in monthly income within each job role.

[Condition :Monthly_Income < Q1 - (Q3 - Q1) * 1.5 OR Monthly_Income > Q3 + (Q3 - Q1)]

```
WITH income_outliers AS (
SELECT EmpNo, JobRole, 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 EmpNo, JobRole, MonthlyIncome, Q1, Q3
FROM income_outliers
WHERE MonthlyIncome < Q1 - (1.5 * (Q3 - Q1)) OR MonthlyIncome > Q3 + (1.5 * (Q3 - Q1));

```

<div> <div>Results</div> <div>Messages</div> </div>					
	EmpNo	JobRole	MonthlyIncome	Q1	Q3
1	STAFF-1985	Laboratory Technician	6323	2379.5	3880.5
2	STAFF-1737	Laboratory Technician	6472	2379.5	3880.5
3	STAFF-1315	Laboratory Technician	6674	2379.5	3880.5
4	STAFF-1132	Laboratory Technician	6782	2379.5	3880.5
5	STAFF-944	Laboratory Technician	7403	2379.5	3880.5
6	STAFF-1516	Manager	11244	16339.5	19129.25
7	STAFF-613	Manager	11557	16339.5	19129.25
8	STAFF-153	Manager	11631	16339.5	19129.25
9	STAFF-376	Manager	11849	16339.5	19129.25
10	STAFF-428	Manager	11878	16339.5	19129.25
11	STAFF-1048	Manager	11904	16339.5	19129.25
12	STAFF-1321	Manager	11916	16339.5	19129.25
13	STAFF-381	Manager	11996	16339.5	19129.25
14	STAFF-96	Research Scientist	6220	2386	3902.5
15	STAFF-679	Research Scientist	6322	2386	3902.5
16	STAFF-347	Research Scientist	6545	2386	3902.5
17	STAFF-691	Research Scientist	6646	2386	3902.5
18	STAFF-1830	Research Scientist	6854	2386	3902.5
19	STAFF-1847	Research Scientist	6962	2386	3902.5
20	STAFF-86	Research Scientist	9724	2386	3902.5
21	STAFF-131	Sales Executive	13872	5025.25	8538.75
22	STAFF-1056	Sales Representative	1052	2250	2878.5

q) Gender distribution within each job role, show each job role with its gender domination.[Male_Domination or Female_Domination]

```
SELECT JobRole,Gender_Count,Gender
FROM (
SELECT JobRole,Gender,COUNT(*) Gender_Count,
RANK() OVER(PARTITION BY JobRole ORDER BY COUNT(*) DESC)
AS gender_rank
FROM EmployeeData
GROUP BY JobRole,Gender
) AS _
WHERE gender_rank = 1;
```

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

r) Percent rank of employees based on training times last year

```
SELECT EmpNo, TrainingTimesLastYear,  
       percent_rank() OVER (ORDER BY TrainingTimesLastYear) as percent_rank  
FROM EmployeeData ORDER BY percent_rank DESC;
```

Results		Messages	
	EmpNo	TrainingTimesLastYear	percent_rank
1	STAFF-42	6	0.95643294758339
2	STAFF-54	6	0.95643294758339
3	STAFF-30	6	0.95643294758339
4	STAFF-153	6	0.95643294758339
5	STAFF-154	6	0.95643294758339
6	STAFF-217	6	0.95643294758339
7	STAFF-258	6	0.95643294758339
8	STAFF-309	6	0.95643294758339
9	STAFF-307	6	0.95643294758339
10	STAFF-381	6	0.95643294758339
11	STAFF-384	6	0.95643294758339
12	STAFF-446	6	0.95643294758339
13	STAFF-492	6	0.95643294758339
14	STAFF-479	6	0.95643294758339
15	STAFF-477	6	0.95643294758339
16	STAFF-533	6	0.95643294758339
17	STAFF-581	6	0.95643294758339
18	STAFF-626	6	0.95643294758339
19	STAFF-749	6	0.95643294758339
20	STAFF-727	6	0.95643294758339
21	STAFF-832	6	0.95643294758339
22	STAFF-850	6	0.95643294758339

s) Divide employees into 5 groups based on training times last year [Use NTILE ()]

```
SELECT EmpNo, TrainingTimesLastYear,  
       ntile(5) OVER (ORDER BY TrainingTimesLastYear DESC) AS TrainingGroup  
FROM EmployeeData;
```

Results		Messages	
	EmpNo	TrainingTimesLastYear	TrainingGroup
1	STAFF-30	6	1
2	STAFF-42	6	1
3	STAFF-54	6	1
4	STAFF-153	6	1
5	STAFF-154	6	1
6	STAFF-217	6	1
7	STAFF-258	6	1
8	STAFF-307	6	1
9	STAFF-309	6	1
10	STAFF-381	6	1
11	STAFF-384	6	1
12	STAFF-446	6	1
13	STAFF-477	6	1
14	STAFF-479	6	1
15	STAFF-492	6	1
16	STAFF-533	6	1
17	STAFF-581	6	1
18	STAFF-626	6	1
19	STAFF-727	6	1
20	STAFF-749	6	1
21	STAFF-832	6	1
22	STAFF-850	6	1

t) Categorize employees based on training times last year as - Frequent Trainee, Moderate Trainee, Infrequent Trainee.

```
SELECT EmpNo, TrainingTimesLastYear,
CASE
    when TrainingTimesLastYear > 6 then 'Frequent Trainee'
    when TrainingTimesLastYear between 3 and 6 then 'Moderate Trainee'
    else 'Infrequent Trainee'
END AS trainee_category
FROM EmployeeData;
```

Results		Messages	
	EmpNo	TrainingTimesLastYear	trainee_category
1	STAFF-1	0	Infrequent Trainee
2	STAFF-2	3	Moderate Trainee
3	STAFF-4	3	Moderate Trainee
4	STAFF-5	3	Moderate Trainee
5	STAFF-7	3	Moderate Trainee
6	STAFF-8	2	Infrequent Trainee
7	STAFF...	3	Moderate Trainee
8	STAFF...	2	Infrequent Trainee
9	STAFF...	2	Infrequent Trainee
10	STAFF...	3	Moderate Trainee
11	STAFF...	5	Moderate Trainee
12	STAFF...	3	Moderate Trainee
13	STAFF...	1	Infrequent Trainee
14	STAFF...	2	Infrequent Trainee
15	STAFF...	4	Moderate Trainee
16	STAFF...	1	Infrequent Trainee
17	STAFF...	5	Moderate Trainee
18	STAFF...	2	Infrequent Trainee
19	STAFF...	3	Moderate Trainee
20	STAFF...	3	Moderate Trainee
21	STAFF...	5	Moderate Trainee
22	STAFF...	4	Moderate Trainee

u) Categorize employees as 'High', 'Medium', or 'Low' performers based on their performance rating, using a CASE WHEN statement.

```
SELECT EmpNo,PerformanceRating,  
CASE  
    when PerformanceRating >= 4 then 'High Performer'  
    when PerformanceRating = 3 then 'Medium Performer'  
    else 'Low Performer'  
END AS PERFCAT  
FROM EmployeeData;
```

Results		Messages	
	EmpNo	PerformanceRating	PERFCAT
1	STAFF-1	3	Medium Performer
2	STAFF-2	4	High Performer
3	STAFF-4	3	Medium Performer
4	STAFF-5	3	Medium Performer
5	STAFF-7	3	Medium Performer
6	STAFF-8	3	Medium Performer
7	STAFF...	4	High Performer
8	STAFF...	4	High Performer
9	STAFF...	4	High Performer
10	STAFF...	3	Medium Performer
11	STAFF...	3	Medium Performer
12	STAFF...	3	Medium Performer
13	STAFF...	3	Medium Performer
14	STAFF...	3	Medium Performer
15	STAFF...	3	Medium Performer
16	STAFF...	3	Medium Performer
17	STAFF...	3	Medium Performer
18	STAFF...	3	Medium Performer
19	STAFF...	3	Medium Performer
20	STAFF...	3	Medium Performer
21	STAFF...	3	Medium Performer
22	STAFF...	4	High Performer

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.

```
SELECT EmpNo, WorkLifeBalance,
CASE
    when WorkLifeBalance = 1 then 'Poor'
    when WorkLifeBalance = 2 then 'Fair'
    when WorkLifeBalance = 3 then 'Good'
    else 'Excellent'
END AS WLBCategory
FROM EmployeeData;
```

	EmpNo	WorkLifeBalance	WLBCategory
1	STAFF-1	1	Poor
2	STAFF-2	3	Good
3	STAFF-4	3	Good
4	STAFF-5	3	Good
5	STAFF-7	3	Good
6	STAFF-8	2	Fair
7	STAFF-10	2	Fair
8	STAFF-11	3	Good
9	STAFF-12	3	Good
10	STAFF-13	2	Fair
11	STAFF-14	3	Good
12	STAFF-15	3	Good
13	STAFF-16	2	Fair
14	STAFF-18	3	Good
15	STAFF-19	3	Good
16	STAFF-20	3	Good
17	STAFF-21	2	Fair
18	STAFF-22	2	Fair
19	STAFF-23	3	Good
20	STAFF-24	3	Good
21	STAFF-26	2	Fair
22	STAFF-27	3	Good

w) Group employees into 3 groups based on their stock option

```
SELECT EmpNo, StockOptionLevel,  
NTILE(3) OVER (ORDER BY StockOptionLevel DESC) AS StockOp  
FROM EmployeeData;
```

Results		Messages	
	EmpNo	StockOptionLevel	StockOp
1	STAFF-10	3	1
2	STAFF-83	3	1
3	STAFF-84	3	1
4	STAFF-107	3	1
5	STAFF-117	3	1
6	STAFF-121	3	1
7	STAFF-159	3	1
8	STAFF-161	3	1
9	STAFF-195	3	1
10	STAFF-244	3	1
11	STAFF-262	3	1
12	STAFF-306	3	1
13	STAFF-283	3	1
14	STAFF-332	3	1
15	STAFF-386	3	1
16	STAFF-403	3	1
17	STAFF-417	3	1
18	STAFF-438	3	1
19	STAFF-453	3	1
20	STAFF-456	3	1
21	STAFF-532	3	1
22	STAFF-551	3	1

x) Find key reasons for Attrition in Company

```
SELECT ReasonForAttrition,
COUNT(*) AS CountOfAttrition,
COUNT(*) * 100.0 / (SELECT COUNT(*) FROM EmployeeData WHERE Attrition = 'Yes') AS
PercentageOfAttrition
FROM ( SELECT
CASE
WHEN WorkLifeBalance <= 2 THEN 'Poor Work-Life Balance'
WHEN DistanceFromHome > 20 THEN 'High Distance from Home'
WHEN JobSatisfaction <= 2 THEN 'Low Job Satisfaction'
WHEN MonthlyIncome < (SELECT AVG(MonthlyIncome) FROM EmployeeData) THEN 'Low
Monthly Income'
ELSE 'Other Reasons'
END AS ReasonForAttrition
FROM
EmployeeData WHERE Attrition = 'Yes'
) AS attrition_reasons
GROUP BY ReasonForAttrition ORDER BY CountOfAttrition DESC;
```

***--INSIGHT -Employees with poor work life balance have high percentage of attrition.
Other reasons are poor worklife balance , low job satisfaction, high distance from home,
low monthly income.***

	ReasonForAttrition	CountOfAttrition	PercentageOfAttrition
1	Poor Work-Life Balance	83	35.021097046413
2	Low Job Satisfaction	62	26.160337552742
3	Low Monthly Income	46	19.409282700421
4	High Distance from Home	28	11.814345991561
5	Other Reasons	18	7.594936708860