
ASSIGNMENT 3

Analyzing employee attrition, employee performance, salary patterns and job roles within the company using SQL

a) Return the shape of the table

```
SELECT COUNT(*) AS NumberOfRows
FROM EmployeeData

SELECT COUNT(*) AS NumberOfCols
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'EmployeeData'
```

	NumberOfRows
1	1470

	NumberOfCols
1	41

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 cumulativeSum
FROM EmployeeData
WHERE TotalWorkingYears > 0
```

				61	HR	33	657
				62	HR	35	692
	department	TotalWorkingYears	cumulativeSum	63	HR	36	728
1	HR	1	1	64	R&D	1	1
2	HR	1	2	65	R&D	1	2
3	HR	1	3	66	R&D	1	3
4	HR	1	4	67	R&D	1	4
5	HR	2	6	68	R&D	1	5
6	HR	2	8	69	R&D	1	6
7	HR	3	11	70	R&D	1	7
8	HR	3	14	71	R&D	1	8
9	HR	4	18	72	R&D	1	9
10	HR	4	22	73	R&D	1	10

Result:

The cumulative sum of total working years is calculated for each department

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

```
SELECT Department,Gender,COUNT(*) AS EmployeeCount
FROM EmployeeData
GROUP BY Department,Gender
ORDER BY Department,EmployeeCount DESC;
```

	Department	Gender	Gender_count
1	R&D	Male	582
2	R&D	Female	379
3	Sales	Male	257
4	Sales	Female	189
5	HR	Male	43
6	HR	Female	20

Result:

From the query result, Male is dominated in all the departments

d) Create a new column AGE_BAND and Show Distribution of Employee's Age band group

```
ALTER TABLE EmployeeData
ADD AGE_BAND varchar(30);
```

```

UPDATE EmployeeData
SET AGE_BAND = CASE
    WHEN Age < 20 THEN 'Under 20'
    WHEN Age BETWEEN 20 AND 29 THEN '20-29'
    WHEN Age BETWEEN 30 AND 39 THEN '30-39'
    WHEN Age BETWEEN 40 AND 49 THEN '40-49'
    WHEN Age BETWEEN 50 AND 59 THEN '50-59'
    ELSE '60 and above'
END;

```

```

SELECT AGE_BAND, COUNT(*) AS EmployeeCount
FROM EmployeeData
GROUP BY AGE_BAND
ORDER BY AGE_BAND;

```

	AGE_BAND	EmployeeCount
1	20-29	309
2	30-39	622
3	40-49	349
4	50-59	168
5	60 and above	5
6	Under 20	17

Result:

Created a new column and employee age group distribution is made.

Most employees are in the group 30-39 age

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

```

SELECT TOP(5) MaritalStatus, COUNT(*) AS Marital_count
FROM EmployeeData
GROUP BY MaritalStatus
ORDER BY Marital_count DESC

```

	MaritalStatus	Marital_count
1	Married	673
2	Single	470
3	Divorced	327

Result:

Married individuals are most common in the company with count of 673

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

```
SELECT top(3) JobRole,
(SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS AttritionRate
FROM EmployeeData
GROUP BY JobRole
ORDER BY AttritionRate DESC
```

	JobRole	AttritionRate
1	Sales Representative	39.759036144578
2	Laboratory Technician	23.938223938223
3	Human Resources	23.076923076923

Result:

From the query, Sales Representatives has highest attrition rate with about 40 %

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

```
SELECT
CASE
WHEN YearsInCurrentRole < 2 THEN '< 2 Years'
WHEN YearsInCurrentRole BETWEEN 2 AND 4 THEN '2 - 4 Years'
WHEN YearsInCurrentRole BETWEEN 5 AND 7 THEN '5 - 7 Years'
ELSE '> 7 Years'
END AS 'Promotion ',
```

```

COUNT(*) AS 'Employee Count'
FROM EmployeeData
GROUP BY
CASE
    WHEN YearsInCurrentRole < 2 THEN '< 2 Years'
    WHEN YearsInCurrentRole BETWEEN 2 AND 4 THEN '2 - 4 Years'
    WHEN YearsInCurrentRole BETWEEN 5 AND 7 THEN '5 - 7 Years'
    ELSE '> 7 Years'
END
ORDER BY 'Employee Count' DESC;

```

	Promotion	Employee Count
1	2 - 4 Years	611
2	< 2 Years	301
3	5 - 7 Years	295
4	> 7 Years	263

Result:

From the query, promotion chances are maximum within 2 to 4 years of joining a post

h) Calculate cumulative sum of total working years for each department

```

SELECT Department, EmployeeNumber, TotalWorkingYears,
SUM(TotalWorkingYears) OVER (PARTITION BY Department ORDER BY EmployeeNumber)
AS CumulativeTotalWorkingYears
FROM EmployeeData
ORDER BY Department, EmployeeNumber;

```

	Department	EmployeeNumber	TotalWorkingYears	CumulativeTotalWork
1	HR	103	16	16
2	HR	133	7	23
3	HR	140	30	53
4	HR	148	23	76
5	HR	177	8	84
6	HR	184	12	96
7	HR	321	7	103
8	HR	424	9	112
9	HR	470	4	116
10	HR	566	1	117
11	HR	590	11	128
12	HR	608	8	136
13	HR	644	32	168
14	HR	665	10	178
15	HR	698	16	194
16	HR	731	23	217

Result:

Cumulative sum of total working years is calculated for each department

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

```
SELECT Department, EmployeeNumber, MonthlyIncome,
RANK() OVER (PARTITION BY Department ORDER BY MonthlyIncome DESC) AS Rank
FROM EmployeeData
ORDER BY Department, Rank;
```

	Department	EmployeeNumber	MonthlyIncome	Rank
1	HR	1338	19717	1
2	HR	1625	19658	2
3	HR	1973	19636	3
4	HR	734	19189	4
5	HR	731	19141	5
64	R&D	259	19999	1
65	R&D	1035	19973	2
66	R&D	1191	19943	3
67	R&D	226	19926	4
68	R&D	787	19859	5

Result:

Data is ranked based on the monthly income of employees for each department

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

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

	Department	AGE_BAND	TotalWorkingYears	TotalWorkYrSum
1	HR	20-29	1	1
2	HR	20-29	1	2
3	HR	20-29	1	3
4	HR	20-29	3	6
5	HR	20-29	3	9
6	HR	20-29	4	13
7	HR	20-29	6	19
8	HR	20-29	6	25
9	HR	20-29	6	31

Result:

Running total for total working years is calculated for each age band group and each department

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.

```
SELECT EmployeeNumber,Department,TotalWorkingYears AS YearsWorked,
AVG(TotalWorkingYears) OVER (PARTITION BY Department) AS AvgYearsInDepartment,
(TotalWorkingYears - AVG(TotalWorkingYears) OVER (PARTITION BY Department)) AS
DifferenceFromAvg
FROM EmployeeData
WHERE Attrition = 'Yes'
ORDER BY Department;
```

	EmployeeNumber	Department	YearsWorked	AvgYearsInDepartment	DifferenceFrc
1	1098	HR	24	6	18
2	133	HR	7	6	1
3	1467	HR	2	6	-4
4	1714	HR	1	6	-5
5	1747	HR	6	6	0
6	1818	HR	6	6	0
7	1842	HR	2	6	-4
8	1844	HR	4	6	-2
9	1944	HR	1	6	-5
10	566	HR	1	6	-5
11	590	HR	11	6	5
12	608	HR	8	6	2
13	593	R&D	4	8	-4
14	58	R&D	23	8	15
15	582	R&D	15	8	7
16	584	R&D	8	8	0
17	587	R&D	12	8	4

Result:

Compared the years worked by the employees before leaving the company with the average working years for each department.

Negative value indicates that the employee leaved before the average working years of the department .

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

```
SELECT Department,AVG(MonthlyIncome) AS AverageMonthlyIncome,
RANK() OVER (ORDER BY AVG(MonthlyIncome) DESC) AS Rank
FROM EmployeeData
WHERE Attrition = 'Yes'
GROUP BY Department
ORDER BY AverageMonthlyIncome DESC;
```

	Department	AverageMonthlyIncome	Rank
1	Sales	5908	1
2	R&D	4108	2
3	HR	3715	3

Rank:

**Ranked the departments based on the average income of employees left.
Sales department has highest average**

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


```

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

```

	MaritalStatus	TotalEmployees	TotalAttrition	AttritionRate
1	Single	470	120	25.531914893617
2	Married	673	84	12.481426448736
3	Divorced	327	33	10.091743119266

Result:

From the table , most of the employees left are single

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

```

SELECT Department,
(SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS AttritionRate
FROM EmployeeData
GROUP BY Department
ORDER BY AttritionRate

```

	Department	AttritionRate
1	R&D	13.839750260145
2	HR	19.047619047619
3	Sales	20.627802690582

Result:

From the table, Sales Department has the hughes attrition rate with 20%

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

```

SELECT emp_no,MonthlyIncome,

```

AVG(MonthlyIncome) OVER(ORDER BY MonthlyIncome ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)

AS Moving_average_income

FROM EmployeeData

	emp_no	MonthlyIncome	Moving_average_income
1	STAFF-701	1009	1009
2	STAFF-1012	1051	1030
3	STAFF-1056	1052	1037
4	STAFF-1876	1081	1061
5	STAFF-1928	1091	1074
6	STAFF-243	1102	1091
7	STAFF-1273	1118	1103
8	STAFF-1974	1129	1116
9	STAFF-411	1200	1149
10	STAFF-1270	1223	1184
11	STAFF-30	1232	1218

Result:

The moving average of monthly income is calculated using avg() over()

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)]

SELECT *

FROM(

SELECT EmployeeNumber,JobRole,MonthlyIncome,
 PERCENTILE_CONT(.25) WITHIN GROUP(ORDER BY MonthlyIncome) OVER() AS Q1,
 PERCENTILE_CONT(.5) WITHIN GROUP(ORDER BY MonthlyIncome) OVER() AS Q2,
 PERCENTILE_CONT(.75) WITHIN GROUP(ORDER BY MonthlyIncome) OVER() AS Q3
 FROM EmployeeData

) as _

WHERE MonthlyIncome < Q1 - (Q3 - Q1) * 1.5 OR MonthlyIncome > (Q3 + (Q3 - Q1))

	EmployeeNumber	JobRole	MonthlyIncome	Q1	Q2	Q3
1	131	Sales Executive	13872	2911	4919	8379
2	431	Healthcare Representative	13964	2911	4919	8379
3	1661	Healthcare Representative	13966	2911	4919	8379
4	1005	Manufacturing Director	13973	2911	4919	8379
5	1352	Manager	14026	2911	4919	8379
6	568	Manager	14118	2911	4919	8379
7	901	Research Director	14275	2911	4919	8379
8	520	Research Director	14336	2911	4919	8379
9	1539	Research Director	14411	2911	4919	8379
10	1424	Research Director	14732	2911	4919	8379
11	84	Manager	14756	2911	4919	8379

Result:

Employees with outliers in monthly income within each job role are identified

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

```

SELECT JobRole,
SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0 END) AS MaleCount,
SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) AS FemaleCount,
CASE
    WHEN SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0 END) > SUM(CASE WHEN
Gender = 'Female' THEN 1 ELSE 0 END) THEN 'Male_Domination'
    WHEN SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) > SUM(CASE WHEN
Gender = 'Male' THEN 1 ELSE 0 END) THEN 'Female_Domination'
    ELSE 'Equal Distribution'
END AS GenderDominance
FROM EmployeeData
GROUP BY JobRole
ORDER BY JobRole;
```

	JobRole	MaleCount	FemaleCount	GenderDominance
1	Healthcare Representative	80	51	Male_Domination
2	Human Resources	36	16	Male_Domination
3	Laboratory Technician	174	85	Male_Domination
4	Manager	55	47	Male_Domination
5	Manufacturing Director	73	72	Male_Domination
6	Research Director	47	33	Male_Domination
7	Research Scientist	178	114	Male_Domination
8	Sales Executive	194	132	Male_Domination
9	Sales Representative	45	38	Male_Domination

Result:

Male domination is seen in all the department in terms of count of employees

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

```
SELECT EmployeeNumber, TrainingTimesLastYear,
PERCENT_RANK() OVER (ORDER BY TrainingTimesLastYear) AS PercentRank
FROM EmployeeData
ORDER BY PercentRank desc;
```

	EmployeeNumber	TrainingTimesLastYear	PercentRank
1	1037	6	0.95643294758339
2	1025	6	0.95643294758339
3	1009	6	0.95643294758339
4	1079	6	0.95643294758339
5	1092	6	0.95643294758339
6	1131	6	0.95643294758339
7	1322	6	0.95643294758339
8	1315	6	0.95643294758339

Result:

Percent rank is found based on training times last year and highest is 95 and lowest is 0

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

```

SELECT EmployeeNumber, TrainingTimesLastYear,
NTILE(5) OVER (ORDER BY TrainingTimesLastYear) AS TrainingGroup
FROM EmployeeData
ORDER BY TrainingGroup, TrainingTimesLastYear;

```

	EmployeeNumber	TrainingTimesLastYear	TrainingGroup
1	1	0	1
2	1003	0	1
3	1006	0	1
4	1022	0	1
5	1069	0	1
6	1107	0	1
7	1108	0	1
8	1133	0	1

Result:

Employees are divided into 5 groups using ntile() based on the number of training last year

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

```

SELECT EmployeeNumber, TrainingTimesLastYear,
CASE
    WHEN TrainingTimesLastYear >= 5 THEN 'Frequent Trainee'
    WHEN TrainingTimesLastYear BETWEEN 2 AND 4 THEN 'Moderate Trainee'
    ELSE 'Infrequent Trainee'
END AS TraineeCategory
FROM EmployeeData

```

	EmployeeNumber	TrainingTimesLastYear	TraineeCategory
1	1	0	Infrequent Trainee
2	10	3	Moderate Trainee
3	100	2	Moderate Trainee
4	1001	2	Moderate Trainee
5	1002	2	Moderate Trainee
6	1003	0	Infrequent Trainee
7	1004	2	Moderate Trainee
8	1005	2	Moderate Trainee

Result:

Employees are categorized based on the trainings attempted last year into 3 groups

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

```

SELECT emp_no, PerformanceRating,
CASE
    WHEN PerformanceRating > 3 THEN 'High Performer'
    WHEN PerformanceRating > 1 THEN 'Medium Performer'
    ELSE 'Low Performer'
END AS 'Performance Ranking'
FROM EmployeeData

```

	emp_no	PerformanceRating	Performance Ranking
1	STAFF-1	3	Medium Performer
2	STAFF-10	4	High Performer
3	STAFF-100	3	Medium Performer
4	STAFF-1001	3	Medium Performer
5	STAFF-1002	3	Medium Performer
6	STAFF-1003	3	Medium Performer
7	STAFF-1004	3	Medium Performer
8	STAFF-1005	3	Medium Performer

Result:

Categorized 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.

```
SELECT EmployeeNumber,WorkLifeBalance,
CASE
    WHEN WorkLifeBalance = 1 THEN 'Poor'
    WHEN WorkLifeBalance = 2 THEN 'Fair'
    WHEN WorkLifeBalance = 3 THEN 'Good'
    WHEN WorkLifeBalance = 4 THEN 'Excellent'
END AS WorkLifeBalanceCategory
FROM EmployeeData
ORDER BY EmployeeNumber;
```

	EmployeeNumber	WorkLifeBalance	WorkLifeBalanceCategory
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

Result:

Employees are categorized based on the work life balance value

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

```
SELECT EmployeeNumber,StockOptionLevel,
```

```

NTILE(3) OVER (ORDER BY StockOptionLevel) AS StockOptionGroup
FROM EmployeeData
ORDER BY EmployeeNumber

```

	EmployeeNumber	StockOptionLevel	StockOptionGroup
1	1	0	1
2	2	1	2
3	4	0	1
4	5	0	1
5	7	1	2
6	8	0	2
7	10	3	3
8	11	1	3

Result:

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

x) Find key reasons for Attrition in Company

```

SELECT JobRole,MaritalStatus,OverTime,
COUNT(*) AS TotalEmployees,
SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS TotalAttrition,
SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS AttritionRate,
AVG(TotalWorkingYears) AS AvgTotalWorkingYears,
AVG(MonthlyIncome) AS AvgMonthlyIncome,
AVG(DistanceFromHome) AS AvgDistanceFromHome,
AVG(EnvironmentSatisfaction) as AvgEnvironmentSatisfaction
FROM EmployeeData
GROUP BY JobRole, MaritalStatus,OverTime
ORDER BY AttritionRate DESC;

```


	MaritalStatus	OverTime	TotalEmployees	TotalAttrition	AttritionRate	AvgTotalWorkingYears	AvgMonthlyIncome	AvgDistanceFromHome	AvgEnvironmentSatisfaction
ive	Single	Yes	14	11	78.571428571428	2	2333	8	2
cian	Single	Yes	19	13	68.421052631578	7	3564	8	2
ive	Divorced	Yes	3	2	66.666666666666	5	2551	7	2
	Single	Yes	32	20	62.500000000000	11	7366	11	2
cian	Divorced	Yes	13	7	53.846153846153	7	2980	8	3
t	Single	Yes	35	18	51.428571428571	6	2990	7	2
s	Divorced	Yes	8	4	50.000000000000	8	3786	8	2
ive	Married	Yes	7	3	42.857142857142	9	2855	7	2
ive	Single	No	24	10	41.666666666666	4	2733	7	2
cian	Married	Yes	30	11	36.666666666666	7	3416	10	2
t	Married	Yes	42	12	28.571428571428	7	3369	9	2

Result:

Based on the output, the following can be concluded

- ☐ Employees with overtime work have higher attrition rate
- ☐ Single employees from every department have more chances of attrition
- ☐ Distance from home is a factor for attrition
- ☐ Employees with monthly income less than 4000 have higher attrition rate

