

EmployeeData SQL Queries

a) Return the Shape of the Table

```
-- Number of Rows
SELECT COUNT(*) AS ROW_SIZE
FROM EmployeeData;

-- Number of Columns
SELECT COUNT(*) AS COL_SIZE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'EmployeeData';
```

Output Table:

Results		Messages	
	ROW_SIZE		COL_SIZE
1	1470	1	40

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;
```

Output Table:

	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

c) Determine Which Gender Has Higher Workforce Strength 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
Gender_rank
    FROM EmployeeData
    GROUP BY Department, Gender
) AS GenderRanks
WHERE Gender_rank = 1;
```

Output Table:

	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

```
ALTER TABLE EmployeeData
ADD AGE_BAND_COUNT INT;

UPDATE EmployeeData
SET AGE_BAND_COUNT = (
    SELECT COUNT(*)
    FROM EmployeeData AS ed2
    WHERE ed2.CF_age_band = EmployeeData.CF_age_band
);

SELECT CF_age_band, COUNT(*) AS AGE_BAND_COUNT
```

```
FROM EmployeeData
GROUP BY CF_age_band;
```

Output Table:

	CF_age_band	AGE_BAND_COUNT
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 Employees and Find the Most Frequent Marital Status

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

Output Table:

	MaritalStatus	Marital_count
1	Married	673

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

```
SELECT TOP(1) JobRole,
COUNT(CASE
    WHEN Attrition = 'Yes' THEN 1
END) * 100/COUNT(*)
AS Attrition_percent
FROM EmployeeData
GROUP BY JobRole
ORDER BY Attrition_percent DESC;
```

Output Table:

	JobRole	Attrition_percent
1	Sales Representative	39

g) Show Distribution of Employee's Promotion and Find the Maximum Chances of Promotion

```

--Distribution Of The promoted employees
SELECT
JobRole,JobLevel,PerformanceRating,YearsInCurrentRole,YearsAtCompany,YearsSinceLastPromotion,
    JobInvolvement,TrainingTimesLastYear
FROM EmployeeData
WHERE YearsSinceLastPromotion > 0
GROUP BY PerformanceRating,JobLevel,YearsSinceLastPromotion,
    YearsInCurrentRole,YearsAtCompany,JobInvolvement,TrainingTimesLastYear,JobRole

--Maximum chances of employee getting promoted.
SELECT
    AVG(CASE WHEN YearsSinceLastPromotion > 0 THEN PerformanceRating ELSE NULL
END)
    AS Promoted_performance,
    AVG(CASE WHEN YearsSinceLastPromotion = 0 THEN PerformanceRating ELSE NULL
END)
    AS Nonpromoted_performance,
    AVG(CASE WHEN YearsSinceLastPromotion > 0 THEN JobLevel ELSE NULL END)
    AS Promoted_Joblevel,
    AVG(CASE WHEN YearsSinceLastPromotion = 0 THEN JobLevel ELSE NULL END)
    AS Nonpromoted_Joblevel,
    AVG(CASE WHEN YearsSinceLastPromotion > 0 THEN YearsAtCompany ELSE NULL END)
    AS Promoted_Experience,
    AVG(CASE WHEN YearsSinceLastPromotion = 0 THEN YearsAtCompany ELSE NULL END)
    AS Nonpromoted_Experience,
    AVG(CASE WHEN YearsSinceLastPromotion > 0 THEN TrainingTimesLastYear ELSE NULL
END)
    AS Promoted_Training,
    AVG(CASE WHEN YearsSinceLastPromotion = 0 THEN TrainingTimesLastYear ELSE NULL
END)
    AS Nonpromoted_Training,
    AVG(CASE WHEN YearsSinceLastPromotion > 0 THEN JobInvolvement ELSE NULL END)
    AS Promoted_Involvement,
    AVG(CASE WHEN YearsSinceLastPromotion = 0 THEN JobInvolvement ELSE NULL END)
    AS Nonpromoted_Involvement,
    (SUM(CASE WHEN YearsSinceLastPromotion > 0 THEN 1 ELSE 0 END) * 100.0 /
COUNT(*))
    AS PromotionRate,
    AVG(YearsSinceLastPromotion) AS Avg_Promotion_Year,
    COUNT(*) AS TotalEmployees
FROM
    EmployeeData
WHERE Attrition = 'No'

```

Output Table:

1.Distribution

JobRole	JobLevel	PerformanceRating	YearsInCurrentRole	YearsAtCompany	YearsSinceLastPromotion	JobInvolvement	TrainingTimesLastYear
Sales Executive	3	3	11	12	2	2	2
Manufacturing Director	3	3	2	4	3	3	2
Sales Executive	3	3	2	4	3	3	5
Human Resources	3	3	6	20	3	2	1
Sales Executive	3	3	7	9	3	2	2
Sales Executive	3	3	7	16	3	2	6

2.Final Table

	Promoted_performance	Nonpromoted_performance	Promoted_Joblevel	Nonpromoted_Joblevel	Promoted_Experience	Nonpromoted_Experience	Promoted_Training	Nonpromoted_Training
1	3	3	2	1	8	4	2	2
Promoted_Training	Nonpromoted_Training	Promoted_Involvement	Nonpromoted_Involvement	PromotionRate	Avg_Promotion_Year	TotalEmployees		
2	2	2	2	2	61.800486618004	2	1233	

INSIGHTS / ANALYSIS :

Employees with higher joblevel , higher experience and better performance has higher chance of getting promoted. The average waiting period of promotion is 2 years. In the data 61% of employees who is working the company got promoted.

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

```
SELECT emp_no, Department, MonthlyIncome,
       DENSE_RANK() OVER (PARTITION BY Department ORDER BY MonthlyIncome DESC) AS
Income_rank
FROM EmployeeData;
```

Output Table:

	emp_no	Department	MonthlyIncome	Income_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	17888	8

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

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

Output Table:

	Department	CF_age_band	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

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 emp_no, dept.Department, YearsAtCompany, Avg_Years_In_Dept
FROM EmployeeData
LEFT JOIN (
    SELECT Department, AVG(YearsAtCompany) AS Avg_Years_In_Dept
    FROM EmployeeData
    GROUP BY Department
) AS dept
ON dept.Department = EmployeeData.Department
ORDER BY emp_no;
```

Output Table:

	emp_no	Department	YearsAtCompany	Avg_Years_In_Dept
1	STAFF-1	Sales	6	7
2	STAFF-1004	R&D	5	6
3	STAFF-1010	R&D	4	6
4	STAFF-1016	R&D	1	6
5	STAFF-1017	R&D	3	6
6	STAFF-1033	R&D	1	6
7	STAFF-1037	Sales	2	7
8	STAFF-1038	Sales	32	7
9	STAFF-1042	R&D	17	6

l) 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 Income_Rank
FROM EmployeeData
```

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

Output Table:

	Department	AvgMonthlyIncome	Income_Rank
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

```
SELECT MaritalStatus, Attrition, COUNT(*) AS marital_count
FROM EmployeeData
GROUP BY MaritalStatus, Attrition
ORDER BY marital_count DESC;
```

Output Table:

	MaritalStatus	Attrition	marital_count
1	Married	No	589
2	Single	No	350
3	Divorced	No	294
4	Single	Yes	120
5	Married	Yes	84
6	Divorced	Yes	33

INSIGHTS / ANALYSIS:

Majority of employees who are currently working in the company are married and majority of employees who left the company are single.

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

```
SELECT TOP(1) Department,
    (COUNT(CASE WHEN Attrition = 'Yes' THEN 1 END) * 100) / COUNT(*) AS
Attrition_Percent
FROM EmployeeData
GROUP BY Department
ORDER BY Attrition_Percent DESC;
```

Output Table:

	Department	Attrition_Percent
1	Sales	20

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

```
SELECT emp_no,MonthlyIncome,JobRole,
AVG(MonthlyIncome) OVER(PARTITION BY JobRole ORDER BY MonthlyIncome
    ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
AS Moving_average_income
FROM EmployeeData
```

Output Table:

	emp_no	MonthlyIncome	JobRole	Moving_average_income
1	STAFF-369	4000	Healthcare Representative	4000
2	STAFF-942	4014	Healthcare Representative	4007
3	STAFF-1501	4035	Healthcare Representative	4016
4	STAFF-1552	4069	Healthcare Representative	4039
5	STAFF-659	4089	Healthcare Representative	4064
6	STAFF-1062	4107	Healthcare Representative	4088

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

```
SELECT emp_no, JobRole, MonthlyIncome
FROM (
    SELECT emp_no, JobRole, MonthlyIncome,
        PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q1,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q2,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q3
    FROM EmployeeData
) AS IncomePercentiles
WHERE MonthlyIncome < Q1 - ((Q3 - Q1) * 1.5) OR MonthlyIncome > Q3 + (1.5 * (Q3 - Q1));
```

Output Table:

	emp_no	JobRole	MonthlyIncome
1	STAFF-1985	Laboratory Technician	6323
2	STAFF-1737	Laboratory Technician	6472
3	STAFF-1315	Laboratory Technician	6674
4	STAFF-1132	Laboratory Technician	6782
5	STAFF-944	Laboratory Technician	7403
6	STAFF-1516	Manager	11244
7	STAFF-613	Manager	11557
8	STAFF-153	Manager	11631
9	STAFF-376	Manager	11849

q) Gender Distribution Within Each Job Role

```
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
```

Output Table:

	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

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

```
SELECT emp_no, TrainingTimesLastYear,
    PERCENT_RANK() OVER (ORDER BY TrainingTimesLastYear) AS training_percentage
FROM EmployeeData
ORDER BY training_percentage DESC;
```

Output Table:

	emp_no	TrainingTimesLastYear	training_percentage
1	STAFF-1037	6	0.95643294758339
2	STAFF-1025	6	0.95643294758339
3	STAFF-1009	6	0.95643294758339
4	STAFF-1079	6	0.95643294758339
5	STAFF-1092	6	0.95643294758339
6	STAFF-1131	6	0.95643294758339
7	STAFF-1322	6	0.95643294758339
8	STAFF-1315	6	0.95643294758339

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

```
SELECT emp_no, TrainingTimesLastYear,
       NTILE(5) OVER (ORDER BY TrainingTimesLastYear DESC) AS training_tile
FROM EmployeeData;
```

Output Table:

	emp_no	TrainingTimesLastYear	training_tile
1	STAFF-1009	6	1
2	STAFF-1025	6	1
3	STAFF-1037	6	1
4	STAFF-1079	6	1
5	STAFF-1092	6	1
6	STAFF-1131	6	1
7	STAFF-1201	6	1
8	STAFF-1242	6	1

t) 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;
```

Output Table:

	emp_no	TrainingTimesLastYear	Training Frequency
1	STAFF-1	0	Infrequent Trainee
2	STAFF-10	3	Moderate Trainee
3	STAFF-100	2	Infrequent Trainee
4	STAFF-1001	2	Infrequent Trainee
5	STAFF-1002	2	Infrequent Trainee
6	STAFF-1003	0	Infrequent Trainee
7	STAFF-1004	2	Infrequent Trainee
8	STAFF-1005	2	Infrequent Trainee
9	STAFF-1006	0	Infrequent Trainee

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

```
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
```

Output Table:

	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
9	STAFF-1006	3	Medium Performer

v) Categorize Employees into 'Poor', 'Fair', 'Good', or 'Excellent' Work-Life Balance

```
SELECT emp_no, WorkLifeBalance,
       CASE
         WHEN WorkLifeBalance > 3 THEN 'Excellent WorkLifeBalance'
         WHEN WorkLifeBalance > 1 THEN 'Fair WorkLifeBalance'
         ELSE 'Poor WorkLifeBalance'
       END AS WorkLifeBalance_Ranking
FROM EmployeeData
```

Output Table:

	emp_no	WorkLifeBalance	WorkLifeBalance Ranking
1	STAFF-1	1	Poor WorkLifeBalance
2	STAFF-10	2	Fair WorkLifeBalance
3	STAFF-100	3	Fair WorkLifeBalance
4	STAFF-1001	3	Fair WorkLifeBalance
5	STAFF-1002	3	Fair WorkLifeBalance
6	STAFF-1003	3	Fair WorkLifeBalance
7	STAFF-1004	3	Fair WorkLifeBalance
8	STAFF-1005	3	Fair WorkLifeBalance
9	STAFF-1006	2	Fair WorkLifeBalance

w) Group Employees into 3 Groups Based on Stock Option Level

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

Output Table:

	StockOptionLevel	Stock RANK
1	3	1
2	3	1
3	3	1
4	3	1
5	3	1
6	3	1
7	3	1
8	3	1
9	3	1

x) Find Key Reasons for Attrition in Company

```
SELECT
  Department,
  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(CASE WHEN Attrition = 'Yes' THEN MonthlyIncome ELSE NULL END)
  AS AttritionIncome,
  AVG(CASE WHEN Attrition = 'No' THEN MonthlyIncome ELSE NULL END)
  AS NonAttritionIncome,
  AVG(CASE WHEN Attrition = 'Yes' THEN DistanceFromHome ELSE NULL END)
  AS AttriDistanceFmHome,
  AVG(CASE WHEN Attrition = 'No' THEN DistanceFromHome ELSE NULL END)
  AS NonAttriDistanceFmHome,
  AVG(CASE WHEN Attrition = 'Yes' THEN YearsAtCompany ELSE NULL END)
  AS AttritionExperience,
  AVG(CASE WHEN Attrition = 'No' THEN YearsAtCompany ELSE NULL END)
  AS NonAttritionExperience,
```

```
AVG(CASE WHEN Attrition = 'Yes' THEN YearsSinceLastPromotion ELSE NULL END)
AS AttritionPromotionYr,
AVG(CASE WHEN Attrition = 'No' THEN YearsSinceLastPromotion ELSE NULL END)
AS NonAttritionPromotionYr,
AVG(CASE WHEN Attrition = 'Yes' THEN WorkLifeBalance ELSE NULL END)
AS AttritionWorkLife,
AVG(CASE WHEN Attrition = 'No' THEN WorkLifeBalance ELSE NULL END)
AS NonAttritionWorkLife,
AVG(CASE WHEN Attrition = 'Yes' THEN JobSatisfaction ELSE NULL END)
AS AttritionSatisfaction,
AVG(CASE WHEN Attrition = 'No' THEN JobSatisfaction ELSE NULL END)
AS NonAttritionSatisfaction
FROM EmployeeData
GROUP BY Department
```

Output Table:

	Department	TotalEmployees	TotalAttrition	AttritionRate	AttritionIncome	NonAttritionIncome	AttriDistanceFmHome	NonAttriDistanceFmHome
1	Sales	446	92	20.627802690582	5908	7232	10	8
2	HR	63	12	19.047619047619	3715	7345	13	7
3	R&D	961	133	13.839750260145	4108	6630	10	8

AttritionExperience	NonAttritionExperience	AttritionPromotionYr	NonAttritionPromotionYr	AttritionWorkLife	NonAttritionWorkLife	AttritionSatisfaction	NonAttritionSatisfaction
5	7	2	2	2	2	2	2
4	7	0	2	2	2	2	2
4	7	1	2	2	2	2	2

INSIGHTS / ANALYSIS :

-- INSIGHT : Employees who leave the company generally earn less, have longer distance to travel,work experience is less.Some department doesn't provide proper promotion which is a factor for attrition too.Sales department has highest attrition rate and R&D has lowest.