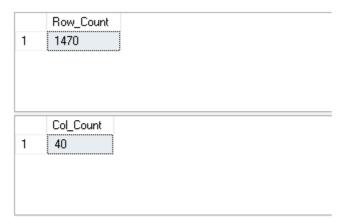# Assignment 3: SQL

a) Return the Shape of the Table

```sql
SELECT COUNT(*) AS Row_Count
FROM EmployeeData;

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

	Row_Count
1	1470

	Col_Count
1	40

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

```sql
SELECT Department,
       TotalWorkingYears,
       SUM(TotalWorkingYears) OVER (
           PARTITION BY Department
           ORDER BY TotalWorkingYears
           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
       ) AS Running_Sum
FROM EmployeeData;
```

	Department	TotalWorkingYears	Running_Sum
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

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

```sql
SELECT Department,
       Gender AS Gender_Domination,
       EmpCount AS Emp_count
FROM (
    SELECT Department,
           Gender,
           COUNT(*) AS EmpCount,
           RANK() OVER (
               PARTITION BY Department
               ORDER BY COUNT(*) DESC
           ) AS rn
    FROM EmployeeData
    GROUP BY Department, Gender
) AS ranked
WHERE rn = 1;
```

	Department	Gender_Domination	Emp_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

```sql
-- Add a new column for age band
ALTER TABLE EmployeeData
ADD AGE_BAND INT;

-- Update the AGE_BAND column
UPDATE EmployeeData
SET AGE_BAND = (
    SELECT COUNT(*)
    FROM EmployeeData AS e
    WHERE e.CF_age_band = EmployeeData.CF_age_band
);

-- Show distribution of age bands
SELECT CF_age_band,
       COUNT(*) AS AGE_BAND_COUNT
FROM EmployeeData
GROUP BY CF_age_band;
```
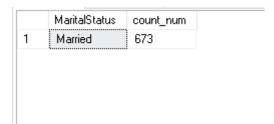
	CF_age_band	AGE_BAND_COUNT
1	Over 55	69
2	35 - 44	505
3	45 - 54	245
4	25 - 34	554
5	Under 25	97

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

```
select top(1) MaritalStatus,COUNT(*) as count_num
from EmployeeData
group by MaritalStatus
order by count_num desc;

--Insight: Most of the employees working are married.
```

	MaritalStatus	count_num
1	Married	673

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

```
SELECT TOP(1) JobRole,
       (total_yes * 100.0 / total_count) AS Attrition_percent
FROM (
    SELECT JobRole,
           COUNT(CASE WHEN Attrition = 'Yes' THEN 1 END) AS total_yes,
           COUNT(*) AS total_count
    FROM EmployeeData
    GROUP BY JobRole
) jobs
ORDER BY Attrition_percent DESC;
```

	JobRole	Attrition_percent
1	Sales Representative	39.759036144578

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

```
-- Distribution of promotions
SELECT YearsSinceLastPromotion, COUNT(*) AS Promoted_Emp
FROM EmployeeData
GROUP BY YearsSinceLastPromotion
ORDER BY Promoted_Emp DESC;
```

	YearsSinceLastPromotion	Promoted_Emp
1	0	581
2	1	357
3	2	159
4	7	76
5	4	61
6	3	52
7	5	45
8	6	32
9	11	24
10	8	18

Insight: This Shows that most promoted employees where last promoted less than 1 year ago.

```
-- Average metrics based on job role and performance rating
SELECT  JobRole,
        PerformanceRating,
        AVG(YearsInCurrentRole) AS avgCurrentRoleYears,
        AVG(YearsAtCompany) AS avgWorkYears,
        AVG(TrainingTimesLastYear) AS avgTrainingTime,
        AVG(YearsSinceLastPromotion) AS avgGapBetweenPromotions
FROM EmployeeData
GROUP BY JobRole, PerformanceRating
ORDER BY avgGapBetweenPromotions ASC;
```

	JobRole	PerformanceRating	avgCurrentRoleYears	avgWorkYears	avgTrainingTime	avgGapBetweenPromotions
1	Human Resources	4	2	5	2	0
2	Laboratory Technician	4	2	3	2	0
3	Sales Representative	3	1	2	2	0
4	Research Scientist	3	3	5	2	1
5	Human Resources	3	3	5	2	1
6	Research Scientist	4	3	5	2	1
7	Manufacturing Director	4	5	7	2	1
8	Sales Representative	4	3	4	4	1
9	Laboratory Technician	3	3	5	3	1
10	Manufacturing Director	3	4	7	2	2

Insight: The table shows that employees with most perfomance rating ang most work experience has the higher chance of promotion.

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

```sql
SELECT *
FROM (
    SELECT emp_no,
           Department,
           MonthlyIncome,
           DENSE_RANK() OVER (
               PARTITION BY Department
               ORDER BY MonthlyIncome DESC
           ) AS rank_
    FROM EmployeeData
) AS ranked
WHERE rank_ <= 5;
```

	emp_no	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-259	R&D	19999	1
7	STAFF-1035	R&D	19973	2
8	STAFF-1191	R&D	19943	3
9	STAFF-226	R&D	19926	4
10	STAFF-787	R&D	19859	5

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

```sql
SELECT Department,
       emp_no,
       TotalWorkingYears,
       SUM(TotalWorkingYears) OVER (
```

```
            PARTITION BY Department
            ORDER BY TotalWorkingYears
            ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
        ) AS running_Work_sum
FROM EmployeeData
WHERE TotalWorkingYears > 0;
```

	Department	emp_no	TotalWorkingYears	running_Work_sum
1	HR	STAFF-1714	1	1
2	HR	STAFF-1865	1	2
3	HR	STAFF-1944	1	3
4	HR	STAFF-566	1	4
5	HR	STAFF-1842	2	6
6	HR	STAFF-1467	2	8
7	HR	STAFF-1858	3	11
8	HR	STAFF-1746	3	14
9	HR	STAFF-1844	4	18
10	HR	STAFF-470	4	22

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

```
WITH YearsWorked AS (
    SELECT
        emp_no,Department as Department_worked,YearsAtCompany AS
YearsWorkedBeforeLeaving
    FROM EmployeeData
    WHERE Attrition = 'Yes'

),
AverageYearsByDepartment AS (
    SELECT Department,
        AVG(YearsAtCompany) AS AvgYearsWorked
    FROM EmployeeData
    GROUP BY Department
)
select*
from YearsWorked LEFT JOIN AverageYearsByDepartment
on AverageYearsByDepartment.Department = YearsWorked.Department_worked;
```

	emp_no	Department_worked	YearsWorkedBeforeLeaving	Department	AvgYearsWorked
1	STAFF-1	Sales	6	Sales	7
2	STAFF-1004	R&D	5	R&D	6
3	STAFF-1010	R&D	4	R&D	6
4	STAFF-1016	R&D	1	R&D	6
5	STAFF-1017	R&D	3	R&D	6
6	STAFF-1033	R&D	1	R&D	6
7	STAFF-1037	Sales	2	Sales	7

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

```sql
SELECT Department,
       AvgMonthlyIncome,
       RANK() OVER (
           ORDER BY AvgMonthlyIncome DESC
       ) AS income_rank
FROM (
    SELECT Department,
           AVG(MonthlyIncome) AS AvgMonthlyIncome
    FROM EmployeeData
    WHERE Attrition = 'Yes'
    GROUP BY Department
) AS left_emp;
```

	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

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

	MaritalStatus	Attrition	emp_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

Insight: Attrition rate is highest for Single employees and majority of working people are married.

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

```sql
SELECT TOP(5) Department,
              yes_count * 100 / yes_count AS Attrition_percent
FROM (
```

```sql
    SELECT Department,
            COUNT(CASE WHEN Attrition = 'Yes' THEN 1 END) AS total_yes,
            COUNT(*) AS yes_count
    FROM EmployeeData
    GROUP BY Department
) jobs
ORDER BY Attrition_percent DESC;
```

	Department	Attrition_percent
1	R&D	100
2	HR	100
3	Sales	100

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

```sql
SELECT emp_no,
        MonthlyIncome,
        AVG(MonthlyIncome) OVER (
            ORDER BY MonthlyIncome
            ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
        ) AS runningIncome
FROM EmployeeData;
```

	emp_no	MonthlyIncome	runningIncome
1	STAFF-701	1009	1009
2	STAFF-1012	1051	1030
3	STAFF-1056	1052	1037
4	STAFF-1876	1081	1048
5	STAFF-1928	1091	1056
6	STAFF-243	1102	1064
7	STAFF-1273	1118	1072
8	STAFF-1974	1129	1079
9	STAFF-411	1200	1092
10	STAFF-1270	1223	1105

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

```sql
SELECT JobRole,
        MonthlyIncome
FROM (
    SELECT JobRole,
            MonthlyIncome,
            PERCENTILE_CONT(.25) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
(PARTITION BY JobRole) AS Q1,
```

```
            PERCENTILE_CONT(.50) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
  (PARTITION BY JobRole) AS Q2,
            PERCENTILE_CONT(.75) WITHIN GROUP (ORDER BY MonthlyIncome) OVER
  (PARTITION BY JobRole) AS Q3
      FROM EmployeeData
  ) boxy
  WHERE MonthlyIncome < Q1 - ((Q3 - Q1) * 1.5)
     OR MonthlyIncome > Q3 + (1.5 * (Q3 - Q1));
```

	JobRole	MonthlyIncome
1	Laboratory Technician	6323
2	Laboratory Technician	6472
3	Laboratory Technician	6674
4	Laboratory Technician	6782
5	Laboratory Technician	7403
6	Manager	11244
7	Manager	11557
8	Manager	11631
9	Manager	11849
10	Manager	11878

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

```
SELECT JobRole,
       Gender AS Gender_Domination,
       EmpCount AS no_count,
       CASE
           WHEN Gender = 'Male' THEN 'Male_Domination'
           WHEN Gender = 'Female' THEN 'Female_Domination'
       END AS Domination
FROM (
    SELECT JobRole,
           Gender,
           COUNT(*) AS EmpCount,
           RANK() OVER (
               PARTITION BY JobRole
               ORDER BY COUNT(*) DESC
           ) AS rn
    FROM EmployeeData
    GROUP BY JobRole, Gender
) AS ranked
WHERE rn = 1;
```

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

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

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

	emp_no	TrainingTimesLastYear	percent_
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
9	STAFF-1311	6	0.95643294758339
10	STAFF-1297	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) AS training_grp
FROM EmployeeData;
```

	emp_no	TrainingTimesLastYear	training_grp
1	STAFF-1	0	1
2	STAFF-1003	0	1
3	STAFF-1006	0	1
4	STAFF-1022	0	1
5	STAFF-1069	0	1
6	STAFF-1107	0	1
7	STAFF-1108	0	1
8	STAFF-1133	0	1
9	STAFF-1156	0	1
10	STAFF-1162	0	1

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

```sql
SELECT emp_no,
       TrainingTimesLastYear,
       CASE
           WHEN TrainingTimesLastYear > 4 THEN 'Frequent Trainee'
           WHEN TrainingTimesLastYear > 2 THEN 'Moderate Trainee'
           ELSE 'Infrequent Trainee'
       END AS Category
FROM EmployeeData
ORDER BY TrainingTimesLastYear DESC;
```

	emp_no	TrainingTimesLastYear	Category
1	STAFF-1009	6	Frequent Trainee
2	STAFF-1025	6	Frequent Trainee
3	STAFF-1037	6	Frequent Trainee
4	STAFF-1079	6	Frequent Trainee
5	STAFF-1092	6	Frequent Trainee
6	STAFF-1131	6	Frequent Trainee
7	STAFF-1201	6	Frequent Trainee
8	STAFF-1242	6	Frequent Trainee
9	STAFF-1243	6	Frequent Trainee
10	STAFF-1283	6	Frequent Trainee

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

```sql
SELECT emp_no,
       PerformanceRating,
       CASE
           WHEN PerformanceRating > 3 THEN 'High Performance'
           WHEN PerformanceRating > 1 THEN 'Medium Performance'
           ELSE 'Low Performance'
       END AS Performance
FROM EmployeeData
ORDER BY PerformanceRating DESC;
```

	emp_no	PerformanceRating	Performance
1	STAFF-1010	4	High Performance
2	STAFF-1035	4	High Performance
3	STAFF-1056	4	High Performance
4	STAFF-10	4	High Performance
5	STAFF-103	4	High Performance
6	STAFF-1080	4	High Performance
7	STAFF-1092	4	High Performance
8	STAFF-1028	4	High Performance
9	STAFF-11	4	High Performance
10	STAFF-1100	4	High Performance

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

```sql
SELECT emp_no,
       WorkLifeBalance,
       CASE
           WHEN WorkLifeBalance > 3 THEN 'Excellent'
           WHEN WorkLifeBalance > 2 THEN 'Good'
           WHEN WorkLifeBalance > 1 THEN 'Fair'
           ELSE 'Poor'
       END AS work_life_balance
FROM EmployeeData
ORDER BY WorkLifeBalance DESC;
```

	emp_no	WorkLifeBalance	work-life balance
1	STAFF-102	4	Excellent
2	STAFF-1029	4	Excellent
3	STAFF-1045	4	Excellent
4	STAFF-1074	4	Excellent
5	STAFF-101	4	Excellent
6	STAFF-1081	4	Excellent
7	STAFF-1097	4	Excellent
8	STAFF-1106	4	Excellent
9	STAFF-1128	4	Excellent
10	STAFF-1131	4	Excellent

w) Group Employees into 3 Groups Based on Their Stock Option Level Using the NTILE Function
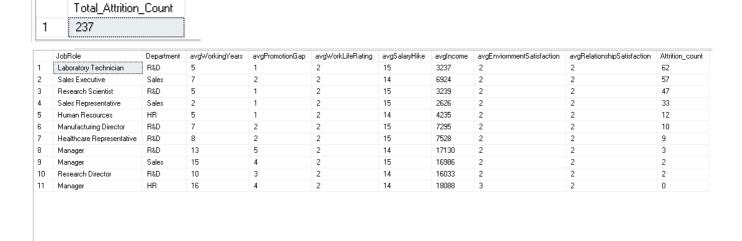
```sql
SELECT emp_no,
       StockOptionLevel,
       NTILE(3) OVER (ORDER BY StockOptionLevel) AS Stock_Level
FROM EmployeeData
ORDER BY Stock_Level;
```

	emp_no	StockOptionLevel	Stock_Level
1	STAFF-1	0	1
2	STAFF-100	0	1
3	STAFF-1003	0	1
4	STAFF-1004	0	1
5	STAFF-1012	0	1
6	STAFF-1016	0	1
7	STAFF-1017	0	1
8	STAFF-1018	0	1
9	STAFF-102	0	1
10	STAFF-1022	0	1

x) Find Key Reasons for Attrition in the Company

```
select COUNT(Attrition) as Total_Attrition_Count
from EmployeeData
where Attrition = 'Yes' ;

select JobRole,Department,
AVG(YearsAtCompany) as avgWorkingYears,
AVG(YearsSinceLastPromotion) as avgPromotionGap,
AVG(WorkLifeBalance) as avgWorkLifeRating,
AVG(PercentSalaryHike) as avgSalaryHike,
AVG(MonthlyIncome) as avgIncome,
AVG(EnvironmentSatisfaction) as avgEnviornmentSatisfaction,
AVG(RelationshipSatisfaction) as avgRelationshipSatisfaction,
COUNT(case when Attrition = 'Yes' then 1 end) Attrition_count
from EmployeeData
group by JobRole,Department
order by Attrition_count desc;
```

	Total_Attrition_Count
1	237

	JobRole	Department	avgWorkingYears	avgPromotionGap	avgWorkLifeRating	avgSalaryHike	avgIncome	avgEnviornmentSatisfaction	avgRelationshipSatisfaction	Attrition_count
1	Laboratory Technician	R&D	5	1	2	15	3237	2	2	62
2	Sales Executive	Sales	7	2	2	14	6924	2	2	57
3	Research Scientist	R&D	5	1	2	15	3239	2	2	47
4	Sales Representative	Sales	2	1	2	15	2626	2	2	33
5	Human Resources	HR	5	1	2	14	4235	2	2	12
6	Manufacturing Director	R&D	7	2	2	15	7295	2	2	10
7	Healthcare Representative	R&D	8	2	2	15	7528	2	2	9
8	Manager	R&D	13	5	2	14	17130	2	2	3
9	Manager	Sales	15	4	2	15	16986	2	2	2
10	Research Director	R&D	10	3	2	14	16033	2	2	2
11	Manager	HR	16	4	2	14	18088	3	2	0

Insight: Departments with lowest of avg salary, avg enviornment satisfaction and avg relationship satisfaction had the highest attrition count.