ASSIGNMENT 3

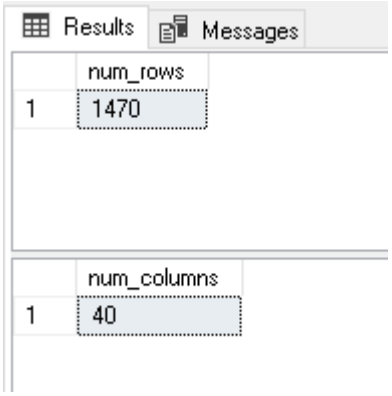
Table creation and inserting data

```
DECLARE @Databasename NVARCHAR(50) = 'HREmployeeDB'

IF NOT EXISTS(SELECT 1 FROM SYS.databases WHERE NAME = @Databasename)

BEGIN
	DECLARE @SQL NVARCHAR(MAX) = 'CREATE DATABASE '+
QUOTENAME(@Databasename)
	EXEC sp_executesql @SQL;
END

USE HREmployeeDB;

CREATE TABLE employee (
   Attrition VARCHAR(20),
   BusinessTravel VARCHAR(26),
   CF_age_band VARCHAR(20),
   CF_attrition_label VARCHAR(35),
   Department VARCHAR(50),
   EducationField VARCHAR(50),
   emp_no VARCHAR(20) PRIMARY KEY,
   EmployeeNumber INT,
   Gender VARCHAR(6),
   JobRole VARCHAR(50),
   MaritalStatus VARCHAR(10),
   OverTime VARCHAR(3),
   Over18 VARCHAR(3),
   TrainingTimesLastYear INT,
   Age INT,
   CF_current 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,
    YearsWithCurrentManager INT
);

BULK INSERT employee
FROM 'C:\Users\Administrator\Downloads\HREmployee.csv'
WITH(
    FIELDTERMINATOR = ',',  -- Field terminated by '|',';','\t'
    ROWTERMINATOR = '0x0a', -- Carriage & New Line Character'\r\n','\n','0x0a'(line feed)
    FIRSTROW = 2
);
```

– a) Return the shape of the table

```
SELECT COUNT(*) AS num_rows FROM employee
SELECT COUNT(*) AS num_columns
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'employee'
```



RESULT: Total no. of rows = 1470, total no. of columns = 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 cumulativeSumYear
FROM employee
WHERE TotalWorkingYears > 0
```

Results | Messages

	Department	TotalWorkingYears	cumulativeSumYear
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

RESULT: Cumulative sum is calculated

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

```
with gender as (SELECT Department,Gender,COUNT(*) AS genderCount,
RANK() OVER(PARTITION BY Department ORDER BY COUNT(*) DESC) as rank from
employee group by Department,Gender)
select Department, Gender, genderCount from gender where rank=1
```

Results | Messages

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

RESULT: In each department, Male gender has higher strength as workforce

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

```
ALTER TABLE employee
ADD AGE_BAND VARCHAR(20);
UPDATE employee
SET AGE_BAND = CASE
   WHEN Age < 25 THEN 'Under 25'
   WHEN Age BETWEEN 25 AND 34 THEN '25-34'
   WHEN Age BETWEEN 35 AND 44 THEN '35-44'
   WHEN Age BETWEEN 45 AND 54 THEN '45-54'
   WHEN Age >= 55 THEN '55 and above'
   ELSE 'Unknown'
END;
```

	AGE_BAND
1	35-44
2	55 and above
3	35-44
4	25-34
5	45-54
6	45-54
7	25-34
8	45-54
9	35-44
10	45-54

RESULT: Created age band

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

```
SELECT TOP(1) MaritalStatus,COUNT(*) AS count
FROM employee
GROUP BY MaritalStatus
ORDER BY count DESC
```

Results | Messages

	MaritalStatus	count
1	Married	673

RESULT: The most frequent marital status is married with a count of 673

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

```
SELECT TOP(1) JobRole,
totalyes * 100/total AS attritionPercent
FROM (
	SELECT JobRole,
	COUNT(CASE
		WHEN Attrition = 'Yes' THEN 1
	END) AS totalyes,
	COUNT(*) total
	FROM employee
	GROUP BY JobRole
) as _
ORDER BY attritionPercent DESC
```

	JobRole	attritionPercent
1	Sales Representative	39

RESULT: The job role with highest attrition rate is Sales Representative.

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

```
SELECT YearsSinceLastPromotion, COUNT(*) AS EmployeeCount
FROM employee
GROUP BY YearsSinceLastPromotion
ORDER BY YearsSinceLastPromotion;

SELECT emp_no, YearsSinceLastPromotion, PerformanceRating, PercentSalaryHike,
case
	when YearsSinceLastPromotion>=(select avg(YearsSinceLastPromotion) from
employee)
	and PerformanceRating>=(select avg(PerformanceRating) from employee)
	and PercentSalaryHike>=(select avg(PercentSalaryHike)from employee)
	then 'yes'
	else 'no'
end as 'Chance of promotion'
from employee
order by YearsSinceLastPromotion desc,PerformanceRating desc, PercentSalaryHike desc
```

Results | Messages

	YearsSinceLastPromotion	EmployeeCount
1	0	581
2	1	357
3	2	159
4	3	52
5	4	61
6	5	45
7	6	32
8	7	76

	emp_no	YearsSinceLastPromotion	PerformanceRating	PercentSalaryHike	Chance of promotion
1	STAFF-244	15	4	25	yes
2	STAFF-1204	15	4	23	yes
3	STAFF-329	15	3	18	yes
4	STAFF-1042	15	3	16	yes
5	STAFF-1278	15	3	15	yes
6	STAFF-569	15	3	15	yes
7	STAFF-1293	15	3	14	no
8	STAFF-1307	15	3	13	no
9	STAFF-162	15	3	13	no

RESULT : The chances of promotion calculated based on YearsSinceLastPromotion, PerformanceRating and PercentSalaryHike

– h) Show 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 cumulativeSumYear
```

FROM employee
WHERE TotalWorkingYears > 0

Results | Messages

	Department	TotalWorkingYears	cumulativeSumYear
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

RESULT : Cumulative sum of years for each department calculated

--i) Find the rank of employees within each department based on their monthly income
SELECT emp_no, Department,MonthlyIncome,
DENSE_RANK() OVER(PARTITION BY Department ORDER BY MonthlyIncome DESC)
AS rankIncome
FROM employee

Results | Messages

	emp_no	Department	MonthlyIncome	rankIncome
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

RESULT: Rank of employees based on monthly income calculated

--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 runningtotal
FROM employee
WHERE TotalWorkingYears > 0
```

Results | Messages

	Department	AGE_BAND	TotalWorkingYears	runningtotal
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

RESULT: Running total calculated

--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 emp_no, dept.Department, YearsAtCompany, avgYears
FROM employee LEFT JOIN
(
        SELECT Department, AVG(YearsAtCompany) as avgYears
        FROM employee
        GROUP BY Department
) as dept
ON dept.Department = employee.Department
where Attrition='Yes'
```

Results | Messages

	emp_no	Department	YearsAtCompany	avgYears
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
10	STAFF-1052	R&D	1	6

RESULT: RESULT: Comparison done between each employee who left, the number of years they worked before leaving and the average years worked by employees in the same department.

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

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

Results | Messages

	Department	AvgMonthlyIncome	Avg_income_rank
1	Sales	5908	1
2	R&D	4108	2
3	HR	3715	3

RESULT: Ranked the employees by average monthly income

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

```
SELECT MaritalStatus,Attrition,COUNT(*) as maritalCount
FROM employee
GROUP BY  MaritalStatus,Attrition
ORDER BY maritalCount DESC
```

Results | Messages

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

RESULT: Single has more attrition rate

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

```
SELECT TOP(1) Department,
(COUNT(CASE
        WHEN Attrition = 'Yes' THEN 1
END) * 100 ) / COUNT(*) AS attritionRate
```

```
FROM employee
GROUP BY Department
ORDER BY attritionRate DESC
```

Results | Messages

	Department	attritionRate
1	Sales	20

RESULT: Sales department has highest attrition rate

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

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

Results | Messages

	emp_no	JobRole	MonthlyIncome	movingAverageIncome
1	STAFF-1014	Healthcare Representative	10388	10388
2	STAFF-1022	Healthcare Representative	4240	7314
3	STAFF-1024	Healthcare Representative	10999	8542
4	STAFF-1033	Healthcare Representative	4777	6672
5	STAFF-1034	Healthcare Representative	6385	7387
6	STAFF-1062	Healthcare Representative	4107	5089
7	STAFF-1082	Healthcare Representative	8722	6404
8	STAFF-1088	Healthcare Representative	8823	7217
9	STAFF-1092	Healthcare Representative	10322	9289
10	STAFF-1099	Healthcare Representative	7119	8754

RESULT: Calculated 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 :
--Monthly_Income < Q1 - (Q3 - Q1) * 1.5 OR Monthly_Income > Q3 + (Q3 - Q1)]

```
SELECT JobRole, MonthlyIncome
FROM(
        SELECT 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 employee
) _
WHERE MonthlyIncome < Q1 - (Q3 - Q1) * 1.5 OR MonthlyIncome > (Q3 + (Q3 - Q1))
```

Results | Messages

	JobRole	MonthlyIncome
1	Sales Executive	13872
2	Healthcare Representative	13964
3	Healthcare Representative	13966
4	Manufacturing Director	13973
5	Manager	14026
6	Manager	14118
7	Research Director	14275
8	Research Director	14336
9	Research Director	14411
10	Research Director	14732

RESULT: Identified employees with outliers in monthly income within each job role

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

```
SELECT JobRole,Gender
FROM (
	SELECT JobRole,Gender,
	RANK() OVER(PARTITION BY JobRole ORDER BY COUNT(*) DESC)
	AS genderRank
	FROM employee
	GROUP BY JobRole,Gender
) AS _
WHERE genderRank = 1
```

Results | Messages

	JobRole	Gender
1	Healthcare Representative	Male
2	Human Resources	Male
3	Laboratory Technician	Male
4	Manager	Male
5	Manufacturing Director	Male
6	Research Director	Male
7	Research Scientist	Male
8	Sales Executive	Male
9	Sales Representative	Male

RESULT: each job role with its gender domination.

--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 employee
order by training_percentage desc
```

Results | Messages

	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
9	STAFF-1311	6	0.95643294758339
10	STAFF-1297	6	0.95643294758339

RESULT: Found percent rank of employees based on training times last year

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

```
SELECT emp_no,TrainingTimesLastYear,
NTILE(5) OVER(ORDER BY TrainingTimesLastYear)
AS trainingGroup
FROM employee
```

Results | Messages

	emp_no	TrainingTimesLastYear	trainingGroup
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

RESULT: Divided employees into 5 groups based on training times last year

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

```
SELECT emp_no,TrainingTimesLastYear,
CASE
        WHEN TrainingTimesLastYear > 4 THEN 'Frequent Trainee'
```

```
        WHEN TrainingTimesLastYear > 2 THEN 'Moderate Trainee'
        ELSE 'Infrequent Trainee'
END AS 'Employee category'
FROM employee
ORDER BY TrainingTimesLastYear DESC
```

Results | Messages

	emp_no	TrainingTimesLastYear	Employee 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

RESULT : Categorized employees based on training times last year

--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'
FROM employee
ORDER BY PerformanceRating DESC
```

Results | Messages

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

RESULT: Categorized employees as 'High', 'Medium', or 'Low' performers based on their performance

--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 emp_no,WorkLifeBalance,
CASE
        WHEN WorkLifeBalance > 3 THEN 'Excellent'
        WHEN WorkLifeBalance > 2 THEN 'Good'
        WHEN WorkLifeBalance > 1 THEN 'Fair'
        ELSE 'Poor'
END AS 'WorkLifeBalance'
FROM employee
ORDER BY PerformanceRating DESC
```

Results | Messages

	emp_no	WorkLifeBalance	WorkLifeBalance
1	STAFF-1010	1	Poor
2	STAFF-1035	3	Good
3	STAFF-1056	3	Good
4	STAFF-10	2	Fair
5	STAFF-103	3	Good
6	STAFF-1080	3	Good
7	STAFF-1092	3	Good
8	STAFF-1028	2	Fair
9	STAFF-11	3	Good
10	STAFF-1100	3	Good

RESULT: Categorized 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.

```
SELECT StockOptionLevel,
NTILE(3) OVER(ORDER BY StockOptionLevel DESC)
AS 'Stock RANK'
FROM employee
```

Results | Messages

	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
10	3	1

RESULT: Grouped employees into 3 groups based on their stock option level

--x) Find key reasons for Attrition in Company

```
SELECT JobSatisfaction,Age,EnvironmentSatisfaction,MonthlyIncome,PercentSalaryHike,
    COUNT(*) AS TotalEmployees,
    SUM(CASE WHEN Attrition = 'Yes' THEN 1 ELSE 0 END) AS AttritionCount
FROM employee
GROUP BY
JobSatisfaction,EnvironmentSatisfaction,Age,MonthlyIncome,PercentSalaryHike;
```

Results | Messages

	JobSatisfaction	Age	EnvironmentSatisfaction	MonthlyIncome	PercentSalaryHike	TotalEmployees	AttritionCount
1	1	25	1	4031	13	1	1
2	1	25	1	4400	12	1	1
3	1	26	1	2042	14	1	1
4	1	29	1	2335	15	1	1
5	1	29	1	2362	13	1	1
6	1	32	1	4200	22	1	1
7	1	33	1	2707	20	1	1
8	1	33	1	3348	11	1	1
9	1	20	1	2973	19	1	1
10	1	33	1	5324	15	1	1

RESULT: The relation between attrition and JobSatisfaction,Age,EnvironmentSatisfaction,MonthlyIncome,PercentSalaryHike calculated