رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

Egypt's digital pioneers' initiative

"Final Project Documentation"

Project:

Human Resources Dataset Analysis

Google Data Analyst Specialist
ONL2_DAT1_G2
Storytellers Team

Acknowledgement

As a team of students from the Egypt's Digital Pioneers Initiative, we would like to extend our deepest gratitude to all those who made this initiative possible. This remarkable program stands as a testament to Egypt's commitment to digital transformation and the strengthening of its national economy, under the generous sponsorship of the Ministry of Communications and Information Technology.

We are sincerely thankful for this invaluable training opportunity, provided free of charge, which has greatly enriched our knowledge and skills. We are also deeply appreciative of the fortune that allowed us to be part of this initiative and to benefit from its extensive resources.

We would also like to express our heartfelt thanks to all the training partners and companies who played a vital role in delivering high-quality training throughout this journey. Special thanks go to the esteemed and dedicated trainers who guided us through this enriching experience over the past six months. Their expertise, commitment, and support have truly inspired us and made this journey both enlightening and enjoyable.

With that, we are honored to present this humble work as a culmination of our learning journey. We hope it meets your expectations and reflects the effort and dedication we have invested throughout this incredible experience.

Executive Summary

This HR Data Analysis Project aims to provide insights into the workforce dynamics, employee qualifications, and organizational efficiency by leveraging data-driven decision-making. The dataset encompasses personal employee details, business details, and employees feedback collected throughout their work periods.

By analyzing these data points, we seek to: tools for deep insights of workforce, uncover trends, strengths, and areas for improvement across various aspects of the employee lifecycle. Through comprehensive analysis, we will identify key drivers of employee engagement, performance, and retention, offering strategic recommendations to enhance HR policies, talent management, and overall business performance. The findings from this study will support informed executive decision-making, ensuring alignment between workforce well-being and organizational objectives.

Project Objectives

1. Analyze Workforce Demographics & Distribution
    - Understand employee distribution across branches, roles, and qualifications.
    - Identify patterns in workforce diversity, tenure, and career progression.
2. Analyze Salary Pattern in the organization
3. Evaluate Employee Satisfaction & Work Experience
    - Assess employee perceptions of their jobs, work environment, relationships, and the life/work balance.
    - Detect trends in employee engagement and dissatisfaction over time.
4. Measure Leadership & Managerial Effectiveness
    - Examine employees' ratings of managers and leadership impact.
    - Identify leadership strengths and areas requiring improvement.
5. Identify Factors Influencing Retention & Turnover
    - Analyze historical survey responses to determine predictors of employee attrition.
    - Provide data-driven recommendations for improving retention strategies.
6. Deliver actionable insights to Optimize HR Strategies & Policies

By addressing these objectives, this project will serve as a strategic tool for enhancing employee engagement, improving managerial effectiveness, and fostering a productive and satisfied workforce.

Data Cleaning & Modeling

Here are the detailed steps performed to the dataset in order to build up the desired analysis report.

Table: Performance Rating

1- Rename columns.
   PerformanceID → Survey ID
   EmployeeID → Employee ID
   ReviewDate → Review Date
   EnvironmentSatisfaction → Environment
   JobSatisfaction → Job
   RelationshipSatisfaction → Relationships
   TrainingOpportunitiesWithinYear → Training Within Year
   TrainingOpportunitiesTaken → Training Taken
   WorkLifeBalance → Work/Life Balance
2- Unpivot the survey questions columns: (Environment – Job – Relationships - Training Within Year - Training Taken - Work/Life Balance.
3- Rename the resulted 2 columns to (question) & (answer).
4- Separate the (question) column in a new table, by Create new query for the (question) column, rename the column, convert the query into a table and rename it (Questions).
5- Rename the old table to (Survey).
6- Add an index column to the table (Questions) and rename the column to (question ID).

Question	Question ID
Environment	1
Job	2
Relationships	3
Work/Life Balance	4
Self Rating	5
Manager Rating	6
Training Within Year	7
Training Taken	8

7- Relate between the tables (Survey) and (Questions) by merge the tables to get the column (question id) in (Survey) then deletes the column (question) from it.
   These steps will help in visualizing the surveys results according to the answers' ratings.

Table: Rating Level & Satisfied Level

1- We will use Classes listed in these tables directly in charts options and DAX.
2- Delete both tables.

Table: Employee

1- Ensure that years in most recent role <= years in company.
2- Correction of the cell value related to employee ID: (9758-DE2F), where the department was mistakenly assigned as (Technology), despite the employee's job role being (Sales Executive). The department updated to (Sales).
The M language code for this modification in power query Includes adding a new column with the value to be modified, then removing the original (department) column, and finally rename the new one, as below:

```
// Update Department only for EmployeeID = "9758-DE2F"
        #"Updated Department" =
                Table.AddColumn(#"Changed Type",
                        "Updated_Department",
                        each if [EmployeeID] = "9758-DE2F" then "Sales" else [Department],
                        type text),

// Remove old "Department" column and rename new one
        #"Removed Old Column" =
                Table.RemoveColumns(#"Updated Department",
                        {"Department"}),
        #"Renamed Column" =
                Table.RenameColumns(#"Removed Old Column",
                {{"Updated_Department", "Department"}})
```

3- Rename column (Education) to (EducationLevelID) and change type to text.
4- Add new conditional column (Education Level) using the attributes from table (EducationLevel) and change type to text. We don't use (merge query) in order to delete the table (EducationLevel) later.

```
// Add new column to show the education level

    #"Added Conditional Column" = Table.AddColumn(#"Changed Type1", "Education Level",
        each if [EducationLevelID] = "1" then "No Qualification"
        else if [EducationLevelID] = "2" then "High School"
        else if [EducationLevelID] = "3" then "Bachelor"
        else if [EducationLevelID] = "4" then "Masters"
        else if [EducationLevelID] = "5" then "Doctorate"
        else null),
```

5- Change Values of Column (Ethnicity):
   Asian or Asian American → Asian/Asian American
   Black or African American → Black/African American
   Native Hawaiian → Hawaiian
   Mixed or multiple ethnic groups → Mixed-ethnic groups
   American Indian or Alaska Native → American Indian/Alaska
6- Add new conditional column (Age Category) and change type to text.

```
// Add new column to classify employees age

   #"Added Conditional Column1" = Table.AddColumn(#"Replaced Value6", "Age Category",
       each if [Age] >= 50 then "50-59"
       else if [Age] >= 40 then "40-49"
       else if [Age] >= 30 then "30-39"
       else if [Age] >= 20 then "20-29"
       else if [Age] >= 18 then "< 20"
       else null),
```

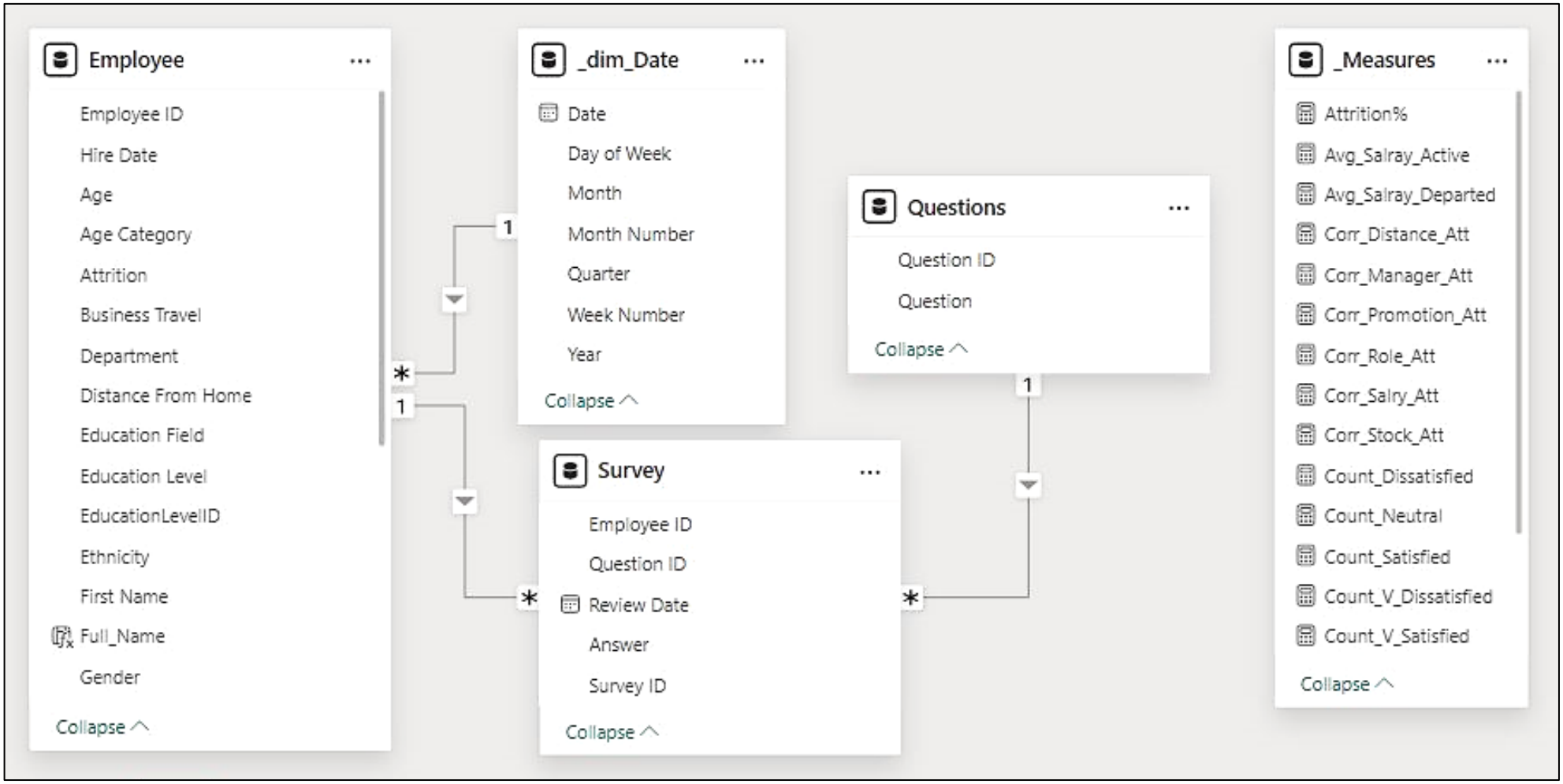
7- Replace values in column (Attrition) as follows:
   Yes → Departed
   No → Active
8- Rename columns to enhance visibility.
9- In Table view:
   - Modify column (Salary) format to currency
   - Sort the column (EducationLevel) by the column (EducationLevelID).
10- In Model View:
   Dividing the table into 2 folders:
   - (Personal) which includes all personal columns for each employee.
   - (Professional) which includes all work columns for each employee.

Table: Education Level

1- Delete Table.

The Data Model

*** schema**



Tables

Name	Description	Expression	IsHidden	IsPrivate
Employee	Table contains all the information about each employee in the the company		FALSE	FALSE
Survey	Table contains all the surveys results of all the employees of all time		FALSE	FALSE
Questions	Table to store and code the questions of the employees' surveys		FALSE	FALSE
_Measures	Table to store and organize DAX measures		FALSE	FALSE
_dim_Date	Time Table for the analysis period of time, to provide a centralized date reference that enhances time-based analysis		FALSE	FALSE
DateTableTemplate_4439d23f-be47-4c76-af94-b88c07a4e1c1		Calendar(Date(2015,1,1), Date(2015,1,1))	TRUE	TRUE
LocalDateTable_eb61007b-a943-48b9-a0bd-3a3586bec2b3		Calendar(Date(Year(MIN('_dim_Date'[Date])), 1, 1), Date(Year(MAX('_dim_Date'[Date])), 12, 31))	TRUE	FALSE
LocalDateTable_ffda792a-e4e2-4437-b266-0a08d4d4f59a		Calendar(Date(Year(MIN('Survey'[Review Date])), 1, 1), Date(Year(MAX('Survey'[Review Date])), 12, 31))	TRUE	FALSE
Model_Columns		INFO.VIEW.COLUMNS()	FALSE	FALSE
Model_Measures		INFO.VIEW.MEASURES()	FALSE	FALSE
Model_Relationships		INFO.VIEW.RELATIONSHIPS()	FALSE	FALSE
Model_Tables		INFO.VIEW.TABLES()	FALSE	FALSE

More detailed (.csv) files of the Model components attached with this documentation.

Also, the detailes tables of components: (Tables – Columns – Relationships – Measures) are included as hidden pages in the power BI project report file.

DAX

Name	Code & Description
01 Full_Name	Retrieve the full name of the employee
	`Full_Name = Employee[First Name] & " " & Employee[Last Name]`
02 Emp_Total	Retrieve the total number of employees
	`Emp_Total = CALCULATE(DISTINCTCOUNT(Employee[Employee ID]))`
03 Emp_Active	Retrieve the total number of employees that still work at the company
	`Emp_Active = CALCULATE(` `  [Emp_Total],` `  Employee[Attrition] = "active"` `)`
04 Emp_Departed	Retrieve the total number of employees that left the company
	`Emp_Departed = COALESCE(` `  CALCULATE(` `  [Emp_Total],` `  Employee[Attrition] = "Departed"` `  ),` `  0` `)`
05 Emp_Hired	Retrieve the total number of employees that have been hired
	`Emp_Hired = [Emp_Active] + [Emp_Departed]`
06 Attrition%	Retrieve the percentage of departed employees to the total
	`Attrition% = [Emp_Departed]/([Emp_Departed]+[Emp_Active])`
07 Count_V_Satisfied	Retrieve the total number of "very satisfied" answers in surveys
	`Count_V_Satisfied =` `CALCULATE(` `  COUNTROWS(Survey),` `  Survey[Answer] = 5,` `  Survey[Question ID] IN {` `  1,` `  2,` `  3,` `  4` `  }` `)`
08 Count_V_Dissatisfied	Retrieve the total number of "very dissatisfied" answers in surveys
	`Count_V_Dissatisfied =` `CALCULATE(` `  COUNTROWS(Survey),` `  Survey[Answer] = 1,`

	`` Survey[Question ID] IN { 1, 2, 3, 4 }) ```
09 Count_Satisfied	Retrieve the total number of "satisfied" answers in surveys ``` Count_Satisfied = CALCULATE(COUNTROWS(Survey), Survey[Answer] = 4, Survey[Question ID] IN { 1, 2, 3, 4 }) ```
10 Count_Neutral	Retrieve the total number of "neutral" answers in surveys ``` Count_Neutral = CALCULATE(COUNTROWS(Survey), Survey[Answer] = 3, Survey[Question ID] IN { 1, 2, 3, 4 }) ```
11 Count_Dissatisfied	Retrieve the total number of "dissatisfied" answers in surveys ``` Count_Dissatisfied = CALCULATE(COUNTROWS(Survey), Survey[Answer] = 2, Survey[Question ID] IN { 1, 2, 3, 4 }) ```
12 Corr_Stock_Att	Calculate the correlation coefficient between attrition behavior and the (stock option level) ``` Corr_Stock_Att = VAR __CORRELATION_TABLE = VALUES('Employee'[State]) VAR __COUNT = COUNTX(KEEPFILTERS(__CORRELATION_TABLE), CALCULATE(AVERAGE('Employee'[Stock Option Level]) * [Emp_Departed]) ```

```
    )
    VAR __SUM_X =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Stock Option Level]))
    )
    VAR __SUM_Y = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed])
    )
    VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Stock Option Level])
        * [Emp_Departed] * 1.
        )
    )
    VAR __SUM_X2 =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Stock Option Level]) ^ 2)
    )
    VAR __SUM_Y2 = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed] ^ 2)
    )
    RETURN
        DIVIDE(
            __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
            SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2)
            * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
            )
        )
```

13 Corr_Salary_Att

Calculate the correlation coefficient between attrition behavior and the (salary)

```
    Corr_Salary_Att =
    VAR __CORRELATION_TABLE = VALUES('Employee'[State])
    VAR __COUNT =
    COUNTX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Salary]) * [Emp_Departed])
    )
    VAR __SUM_X =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Salary]))
    )
    VAR __SUM_Y = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed])
    )
    VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Salary]) * [Emp_Departed] *
1.)
    )
    VAR __SUM_X2 =
    SUMX(
```

```
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Salary]) ^ 2)
    )
    VAR __SUM_Y2 = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed] ^ 2)
    )
    RETURN
        DIVIDE(
            __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
            SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2)
            * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
            )
        )
```

14 Corr_Role_Att

Calculate the correlation coefficient between attrition behavior and the (job role)

```
    Corr_Role_Att =
    VAR __CORRELATION_TABLE = VALUES('Employee'[State])
    VAR __COUNT =
    COUNTX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsInMostRecentRole])
        * [Emp_Departed]
        )
    )
    VAR __SUM_X =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsInMostRecentRole]))
    )
    VAR __SUM_Y = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed])
    )
    VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsInMostRecentRole])
        * [Emp_Departed] * 1.
        )
    )
    VAR __SUM_X2 =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsInMostRecentRole]) ^ 2)
    )
    VAR __SUM_Y2 = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed] ^ 2)
    )
    RETURN
        DIVIDE(
            __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
            SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2)
            * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
            )
        )
```

15 Corr_Promotion_Att	Calculate the correlation coefficient between attrition behavior and the (No. of years since last promotion)

```
Corr_Promotion_Att =
VAR __CORRELATION_TABLE = VALUES('Employee'[State])
VAR __COUNT =
COUNTX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE(AVERAGE('Employee'[YearsSinceLastPromotion])
    * [Emp_Departed]
    )
)
VAR __SUM_X =
SUMX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE(AVERAGE('Employee'[YearsSinceLastPromotion]))
)
VAR __SUM_Y = SUMX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE([Emp_Departed])
)
VAR __SUM_XY =
SUMX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE(AVERAGE('Employee'[YearsSinceLastPromotion])
    * [Emp_Departed] * 1.
    )
)
VAR __SUM_X2 =
SUMX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE(AVERAGE('Employee'[YearsSinceLastPromotion]) ^
2)
)
VAR __SUM_Y2 = SUMX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE([Emp_Departed] ^ 2)
)
RETURN
    DIVIDE(
        __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
        SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2)
        * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
        )
    )
```

16 Corr_Manager_Att	Calculate the correlation coefficient between attrition behavior and the (No. of years with current manager)

```
Corr_Manager_Att =
VAR __CORRELATION_TABLE = VALUES('Employee'[State])
VAR __COUNT =
COUNTX(
    KEEPFILTERS(__CORRELATION_TABLE),
    CALCULATE(AVERAGE('Employee'[YearsWithCurrManager])
    * [Emp_Departed]
    )
)
VAR __SUM_X =
```

```
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsWithCurrManager]))
    )
    VAR __SUM_Y = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed])
    )
    VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsWithCurrManager])
        * [Emp_Departed] * 1.
        )
    )
    VAR __SUM_X2 =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[YearsWithCurrManager]) ^ 2)
    )
    VAR __SUM_Y2 = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed] ^ 2)
    )
    RETURN
        DIVIDE(
            __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
            SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2)
            * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
            )
        )
```

17 Corr_Distance_Att

Calculate the correlation coefficient between attrition behavior and the (distance between work and employee's home)

```
    Corr_Distance_Att =
    VAR __CORRELATION_TABLE = VALUES('Employee'[State])
    VAR __COUNT =
    COUNTX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Distance From Home])
        * [Emp_Departed]
        )
    )
    VAR __SUM_X =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Distance From Home]))
    )
    VAR __SUM_Y = SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE([Emp_Departed])
    )
    VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Employee'[Distance From Home])
        * [Emp_Departed] * 1.
        )
```

	``) VAR __SUM_X2 = SUMX(KEEPFILTERS(__CORRELATION_TABLE), CALCULATE(AVERAGE('Employee'[Distance From Home]) ^ 2)) VAR __SUM_Y2 = SUMX(KEEPFILTERS(__CORRELATION_TABLE), CALCULATE([Emp_Departed] ^ 2)) RETURN DIVIDE(__COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1., SQRT((__COUNT * __SUM_X2 - __SUM_X ^ 2) * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2))) ```
18 Avg_Salary_Departed	Calculate the average salary of departed employees ``` Avg_Salary_Departed = CALCULATE(AVERAGE(Employee[Salary]), Employee[Attrition] = "Departed") ```
19 Avg_Salary_Active	Calculate the average salary of active employees ``` Avg_Salary_Active = CALCULATE(AVERAGE(Employee[Salary]), Employee[Attrition] = "Active") ```

SQL codes

Cleaning fact_Employee Table

```sql
--Remove duplicate employees
DELETE FROM fact_Employee
WHERE EmployeeID IN (
    SELECT EmployeeID FROM fact_Employee
    GROUP BY EmployeeID
    HAVING COUNT(*) > 1);
```

```sql
--Remove employees with missing value in the main coulmns
DELETE FROM fact_Employee
WHERE EmployeeID IS NULL
   OR FirstName IS NULL
   OR LastName IS NULL
   OR Gender IS NULL
   OR Age IS NULL
   OR Department IS NULL
   OR Salary IS NULL;
```

```sql
--Remove invalid salary values
DELETE FROM fact_Employee
WHERE Salary <= 0;
```

```sql
--Remove employees with unrealistic ages
DELETE FROM fact_Employee
WHERE Age < 18 OR Age > 70;
```

```sql
--Change the department of employee id(9758-DE2F) from Technology into
Sales
UPDATE fact_Employee
SET Department = 'Sales'
WHERE EmployeeID = '9758-DE2F';
```

```sql
--Cleaning Dim_SatisfiedLevel Table
DELETE FROM Dim_SatisfiedLevel
```

```
WHERE SatisfactionLevel IS NULL
   OR SatisfactionID NOT IN (1, 2, 3, 4, 5);
```

Ensuring Referential Integrity

```
07 --Remove employees with invalid education levels
DELETE FROM fact_Employee
WHERE EducationLevelID NOT IN (SELECT EducationLevelID FROM
Dim_EducationLevel);
```

```
08 --Remove performance ratings for non-existent employees
DELETE FROM fact_PerformanceRating
WHERE EmployeeID NOT IN (SELECT EmployeeID FROM fact_Employee);
```

```
09 --Normalize OverTime Values
UPDATE fact_Employee
SET OverTime = 'Yes'
WHERE OverTime ILIKE 'yes' OR OverTime ILIKE 'y' OR OverTime ILIKE
'true' OR OverTime = '1';

UPDATE fact_Employee
SET OverTime = 'No'
WHERE OverTime ILIKE 'no' OR OverTime ILIKE 'n' OR OverTime ILIKE
'false' OR OverTime = '0';
```

```
10 --Normalize Attrition Values
UPDATE fact_Employee
SET Attrition = 'Departed'
WHERE Attrition ILIKE 'yes' OR Attrition ILIKE 'y' OR Attrition ILIKE
'true' OR Attrition = '1';

UPDATE fact_Employee
SET Attrition = 'Active'
WHERE Attrition ILIKE 'no' OR Attrition ILIKE 'n' OR Attrition ILIKE
'false' OR Attrition = '0';
```

```
11 --Normalize Gender Values
UPDATE fact_Employee
SET Gender = 'Male'
WHERE Gender ILIKE 'm' OR Gender ILIKE 'male';
```

```sql
UPDATE fact_Employee
SET Gender = 'Female'
WHERE Gender ILIKE 'f' OR Gender ILIKE 'female';
```

```sql
--Normalize MaritalStatus Values
UPDATE fact_Employee
SET MaritalStatus = 'Single'
WHERE MaritalStatus ILIKE 'single';

UPDATE fact_Employee
SET MaritalStatus = 'Married'
WHERE MaritalStatus ILIKE 'married';

UPDATE fact_Employee
SET MaritalStatus = 'Divorced'
WHERE MaritalStatus ILIKE 'divorced';
```

Validating Numeric Ranges

```sql
--Validate Salary Range
DELETE FROM fact_Employee
WHERE Salary < 10000 OR Salary > 600000;
```

```sql
--Validate YearsAtCompany
DELETE FROM fact_Employee
WHERE YearsAtCompany < 0 OR YearsAtCompany > 50;
```

```sql
--Validate YearsInMostRecentRole (not exceed YearsAtCompany)
DELETE FROM fact_Employee
WHERE YearsInMostRecentRole > YearsAtCompany;
```

```sql
--Validate YearsSinceLastPromotion (not exceed YearsAtCompany)
DELETE FROM fact_Employee
WHERE YearsSinceLastPromotion > YearsAtCompany;
```

```sql
--Validate YearsWithCurrManager (not exceed YearsAtCompany)
DELETE FROM fact_Employee
WHERE YearsWithCurrManager > YearsAtCompany;
```