hr Data analysis

DEPI Final Project Documentation

Team Members

Hady Mohamed Kilany

Yomna Elsayed Shehata

Shiamaa Elromh

Mohamed Yassen

Farah Essam

1. **Introduction**:

This project provides an in-depth analysis of HR data to assess key aspects such as employee performance, satisfaction, experience, and attrition. Through interactive dashboards and SQL queries, we explore various metrics, including employee count, average satisfaction, ratings, and salary trends over time. The analysis also covers employee demographics such as age, gender, ethnicity, and business travel frequency.

The project focuses on identifying relationships between employee experience (e.g., years at the company) and performance ratings, as well as the impact of training opportunities on top performers. Additionally, attrition analysis by department and job role helps pinpoint areas with high turnover, allowing HR managers to implement targeted retention strategies.

By combining data visualization and SQL-driven insights, this project equips HR managers with actionable information to enhance employee retention, optimize performance evaluations, and improve overall job satisfaction.

* **Key goals of this analysis include:**

1. **Tracking Workforce Demographics and Diversity:** Gaining insights into workforce composition by gender, ethnicity, age, and business travel frequency to support diversity and inclusion initiatives.
2. **Improving Job Satisfaction:** Assessing satisfaction metrics (e.g., environment, job, work-life balance) across various employee groups to address areas with lower satisfaction and increase engagement.
3. **Understanding Employee Performance Trends:** Analyzing self and manager ratings to identify discrepancies, trends, and areas for improvement across different departments, job roles, and demographics.
4. **Optimizing Training and Development:** Evaluating the relationship between training opportunities and performance to ensure top performers are receiving adequate professional development.
5. **Enhancing Employee Retention:** Identifying factors contributing to employee attrition, such as department, job role, or experience, to develop targeted strategies for reducing turnover.
6. **Tools and Technologies Used**
7. **Jupiter Notebook (Python)**:

* **Purpose:**

To prepare and clean the HR datasets for analysis, ensuring data consistency and readiness for exploring employee performance, satisfaction, and attrition trends.

* **Actions:**

Loaded and merged multiple datasets (employee, performance, education, satisfaction).

Cleaned data by handling missing values, removing extra spaces, and formatting columns.

Calculated key metrics like attrition year and filtered out-of-range performance reviews.

Verified and standardized data types across all datasets for accurate analysis.

1. **SQL**:

* **Purpose:**

To connect to SQL Server, upload the cleaned data, and perform advanced data analysis using SQL queries directly from the Jupyter notebook for more efficient querying and insight extraction.

* **Actions:**

Established connection to SQL Server using SQLAlchemy and ODBC.

Uploaded preprocessed dataframes as SQL tables.

Executed SQL queries to extract insights on employee performance, satisfaction, and attrition directly from the database.

1. **Power BI:**

* **Purpose:**  
  To visualize and analyze key HR metrics using Power BI dashboards, enabling HR managers to gain insights into employee performance, satisfaction, demographics, and attrition trends. These dashboards provide an interactive way to explore the data and track trends over time.
* **Actions:**
* Created interactive dashboards to visualize employee metrics such as satisfaction levels, performance ratings, and salary trends.
* Included filters for department, job role, year, state, education level, and other key demographics to allow for deeper analysis.
* Tracked and visualized trends in attrition, performance, and satisfaction over time to provide actionable insights for HR decision-making.
* **Let’s go in depth in steps in each tool:**

1. **Python:**

In this project, we used Python to handle the preprocessing of multiple HR-related datasets. The preprocessing steps are crucial to ensure the data is clean and ready for analysis. The following steps were performed**:**

### **Importing Required Libraries**

In this section, we import all the necessary libraries that will be used throughout preprocessing. These libraries provide various functions for data manipulation.**A white rectangular object with black text

Description automatically generated**

### **Data Loading and Initial Exploration**[¶](http://localhost:8888/notebooks/Final%20HR%20project.ipynb#Data-Loading-and-Initial-Exploration)

**A screenshot of a computer code

Description automatically generated**In this section, we load the various datasets required for the analysis, including education levels, employee information, performance ratings, and satisfaction levels. After loading the data, we conduct an initial exploration to inspect the structure of each dataset and identify any missing values or unique entries within key columns.

### **Checking for Missing Values:**

A screenshot of a computer program

Description automatically generatedWe checked each dataset for missing values using isnull().sum(). Fortunately, there were no missing values in any of the datasets, ensuring data integrity and reducing the need for imputation.

### **Identifying Unique Values:**

A screen shot of a computer code

Description automatically generatedTo understand the variety of data, we calculated the number of unique values for each column in the employee and performance rating datasets. This step helps identify categorical variables and spot potential anomalies in the data.

### **Preprocessing Steps**

1. **Removing Spaces from the 'EducationField' Column**
   * **Objective**: To ensure that any leading or trailing spaces in the 'EducationField' column are removed for accurate analysis and comparisons
2. **Verifying Unique Values**

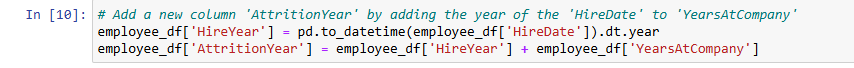
* **Objective**: To check the unique values in the 'EducationField' column after removing spaces, ensuring that the preprocessing step was successful.

A screenshot of a computer code

Description automatically generated**Summary**

* This preprocessing step is crucial for maintaining data integrity, especially when performing further analysis or when categorizing employees based on their education fields.

1. **Calculating Attrition Year**

* **Objective**: To calculate the year in which an employee is expected to leave the company based on their hire date and the number of years they have been with the company.

1. **Dropping Outdated Performance Ratings**

* **Objective**: To ensure that performance ratings are relevant by removing any ratings that fall outside the expected date range based on the calculated 'AttritionYear'.
* **Note**: If applicable, you can add code here for how you dropped the outdated performance ratings.

**Summary**

* This step is essential for accurately forecasting employee attrition and ensuring that performance evaluations are based on relevant time frames

1. **Converting Review Date to Datetime**

* **Objective**: To convert the 'ReviewDate' column in the performance rating dataframe to a datetime format, facilitating date comparisons.

1. **Merging Employee and Performance Data**

* **Objective**: To merge the employee dataframe with the performance dataframe, incorporating hire and attrition years for each employee

1. **Filtering Performance Reviews by Date Range**

* **Objective**: To filter out performance reviews that fall outside the employee's hire and attrition year range, ensuring data relevance.

1. **Dropping Unnecessary Columns**

* A screenshot of a computer code

  Description automatically generated**Objective**: To remove extra columns ('HireYear' and 'AttritionYear') from the filtered performance dataframe, simplifying the dataset.

**Output:**

A screenshot of a computer

Description automatically generated

1. **SQL Server:**

### **Connecting to SQL Server and Uploading Data**

#### Establishing a Connection to SQL Server

* + A screenshot of a computer code

    Description automatically generated**Objective**: To create a connection to the SQL Server database for storing and analyzing HR data.

#### Uploading DataFrames to SQL Server

* **A close-up of a computer code

  Description automatically generatedObjective**: To upload the processed DataFrames to SQL Server tables for further analysis and insights.

**Summary**

* This final step establishes a connection to the SQL Server and uploads the relevant Data Frames, ensuring the data is available for further analysis. Each Data Frame is uploaded to its corresponding table, with existing data replaced to maintain consistency.

**A screenshot of a computer

Description automatically generated**

#### Extracting Insights Using SQL Queries

After successfully uploading the data, we can start executing SQL queries within the notebook to extract insights from the data. The SQL queries can now be run directly against the SQL Server database, taking advantage of SQL's powerful querying capabilities to generate insights for further analysis.

This integration provides the flexibility to use both Python and SQL in a seamless workflow, enhancing data analysis efficiency.

### Salary Insights

In this section, we use SQL queries to extract insights related to employee salaries. These insights include average salary by department and job role, salary breakdown by age group and gender, and an analysis of the correlation between salary and performance ratings. The results of these queries are displayed in a scrollable format for better readability.

#### 1. Average Salary by Department and Job Role

We begin by calculating the average salary across all departments and job roles. The query computes the average salary for each department as well as the breakdown by individual job roles within each department.

* **SQL Query**:
  + First, we calculate the average salary for each department.
  + Then, we compute the average salary for each combination of department and job role.
  + Results are ordered by department and job role for easier comparison.

A white screen with red text

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 2. Salary and Experience by Age Group and Gender

Next, we investigate how salary, along with various experience-related metrics, differs by age group and gender for employees who have not left the company (i.e., **Attrition = 'No'**). The query provides average years at the company, years in the most recent role, years since last promotion, years with the current manager, and average salary for each age group and gender.

* **SQL Query**:
  + The data is grouped by age and gender.
  + The analysis includes employees who have not left the company (Attrition = 'No').

A screenshot of a computer program

Description automatically generated

A white rectangular object with black text

Description automatically generated with medium confidence

#### 3. Average Salary by Department, Job Role, Education Level, and Business Travel

We also analyze how salary varies across different departments, job roles, education levels, and business travel frequency for employees who are still active (Attrition = 'No'). This query helps us understand the impact of education and travel requirements on compensation within each role and department.

* **SQL Query**:
  + The data is grouped by department, job role, education level, and business travel frequency.
  + Employees with **Attrition = 'No'** are included in the analysis.

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 4. Correlation Between Salary and Manager Rating

Finally, we examine whether there is any correlation between salary and manager rating. This query retrieves active employees' average manager ratings, their job roles, and salaries, and orders the results by job role and salary to see if higher-performing employees tend to have higher salaries.

* **SQL Query**:
  + We calculate the average manager rating for each employee.
  + The results are grouped by job role and ordered by salary in descending order to analyze potential correlations.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

### Employee Satisfaction Insights

In this section, we utilize SQL queries to analyze employee satisfaction across various dimensions such as department, job role, age group, gender, and ethnicity. The goal is to identify trends in employee satisfaction and uncover areas where improvement may be needed. We also explore the relationship between employee satisfaction and performance ratings.

#### 1. High Satisfaction by Department and Job Role (Average >= 3)

This query groups employees by department and job role, displaying those with average satisfaction scores (environment, job, relationship, and work-life balance) greater than or equal to 3. Employees with high satisfaction are important to identify for retaining top talent.

* **Metrics Calculated**:
  + Average Environment Satisfaction
  + Average Job Satisfaction
  + Average Relationship Satisfaction
  + Average Work-Life Balance
  + Employee Count per group

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 2. Low Satisfaction by Department and Job Role (Average < 3)

This query is similar to the one above, but focuses on employees whose satisfaction scores are below 3. Identifying areas with low satisfaction helps pinpoint departments and roles where interventions may be necessary to improve employee engagement and retention.

* **Metrics Calculated**:
  + Average Environment Satisfaction
  + Average Job Satisfaction
  + Average Relationship Satisfaction
  + Average Work-Life Balance
  + Employee Count per group

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 3. High Satisfaction by Age Group, Gender, and Ethnicity (Average >= 3)

In this query, employees are grouped by age, gender, and ethnicity, and the average satisfaction scores are calculated for each group. This helps reveal any demographic patterns in employee satisfaction, ensuring that all groups are receiving a positive work experience.

* **Metrics Calculated**:
  + Average Environment Satisfaction
  + Average Job Satisfaction
  + Average Relationship Satisfaction
  + Average Work-Life Balance
  + Employee Count per group

A screenshot of a computer program

Description automatically generated

A screenshot of a white screen

Description automatically generated

#### 4. Low Satisfaction by Age Group, Gender, and Ethnicity (Average < 3)

This query identifies age, gender, and ethnicity groups with low satisfaction scores. It helps in uncovering demographic groups that may require attention to improve their work experience and engagement levels.

* **Metrics Calculated**:
  + Average Environment Satisfaction
  + Average Job Satisfaction
  + Average Relationship Satisfaction
  + Average Work-Life Balance
  + Employee Count per group



A screenshot of a computer

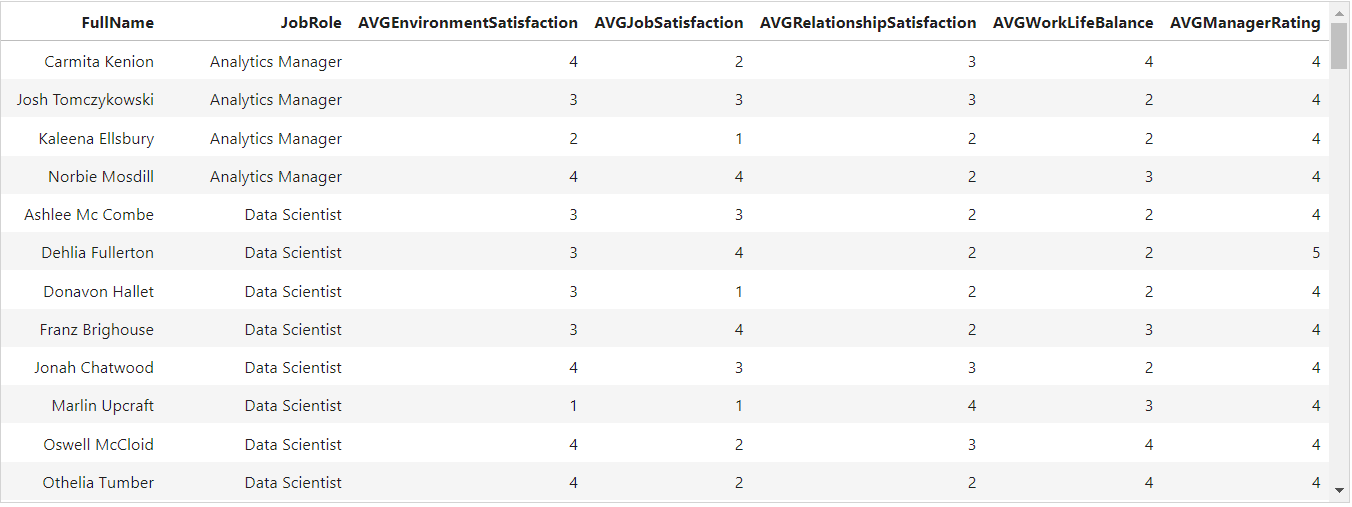
Description automatically generated

#### 5. Highly Rated Employees with Low Satisfaction

This query focuses on employees who receive high performance ratings from their managers (average rating >= 4), but report low satisfaction in terms of environment, job, relationships, or work-life balance (average satisfaction < 3 in any category). These employees are critical to identify, as they may be at risk of attrition despite being high performers.

* **Metrics Calculated**:
  + Average Environment Satisfaction
  + Average Job Satisfaction
  + Average Relationship Satisfaction
  + Average Work-Life Balance
  + Average Manager Rating





### Employee Rating Insights

In this section, we use SQL queries to gain insights into employee ratings, which include both **self-ratings** and **manager ratings**. The goal is to understand how employee ratings vary across different departments, job roles, age groups, genders, and ethnicities. We also identify areas where employee ratings may be below expectations and need attention.

#### 1. High Ratings by Department and Job Role (Average >= 3)

This query groups employees by department and job role and calculates the average self-rating and manager rating for each group. Only those groups with average ratings greater than or equal to 3 are displayed. This helps identify departments and roles with employees who consistently rate their own performance positively and receive high ratings from their managers.

* **Metrics Calculated**:
  + Average Self Rating
  + Average Manager Rating
  + Employee Count per group

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 2. Low Ratings by Department and Job Role (Average < 3)

This query focuses on employees whose average self-rating or manager rating is below 3. Identifying areas with low ratings helps pinpoint where performance may need improvement, or where additional training and support could be provided.

* **Metrics Calculated**:
  + Average Self Rating
  + Average Manager Rating
  + Employee Count per group

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 3. High Ratings by Age Group, Gender, and Ethnicity (Average >= 3)

In this query, employees are grouped by age, gender, and ethnicity, and the average self-rating and manager rating are calculated for each demographic group. This helps reveal trends in employee performance ratings across different demographics and ensures that there is no bias in performance evaluations.

* **Metrics Calculated**:
  + Average Self Rating
  + Average Manager Rating
  + Employee Count per group

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 4. Low Ratings by Age Group, Gender, and Ethnicity (Average < 3)

This query identifies age, gender, and ethnicity groups with low self or manager ratings. This helps in uncovering demographic groups that may require more attention, training, or support to improve their performance.

* **Metrics Calculated**:
  + Average Self Rating
  + Average Manager Rating
  + Employee Count per group

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

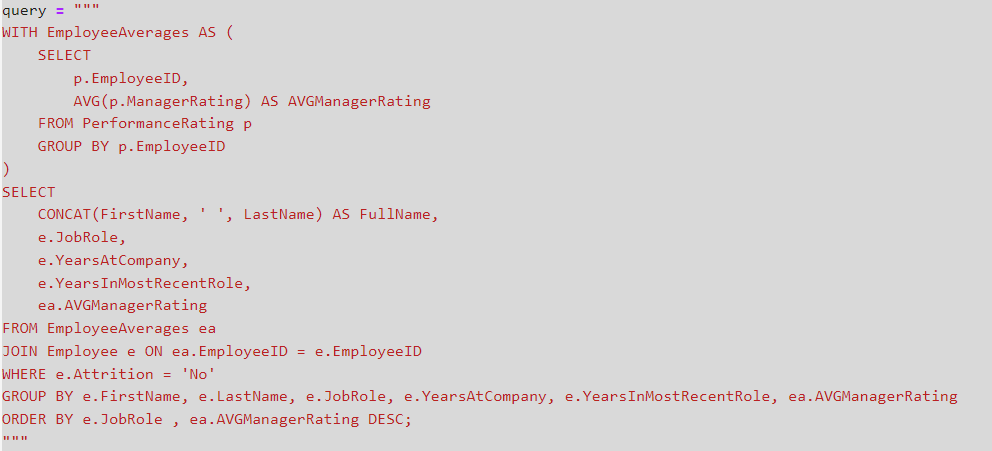
### Experience, Training, and Attrition Analysis

This section covers analysis of employee experience, training opportunities, and attrition rates. Through SQL queries, we explore the relationships between experience and performance, identify employees in long-standing roles, assess training opportunities, and calculate attrition rates by department and job role.

#### 1. Correlation Between Experience and Performance

This query explores the relationship between the number of years employees have been with the company (and in their most recent role) and their average manager ratings. It helps identify whether more experienced employees tend to perform better or if there is any observable correlation.

* **Metrics Calculated**:
  + Job Role
  + Years at Company
  + Years in Most Recent Role
  + Average Manager Rating



A screenshot of a computer

Description automatically generated

#### 2. Employees in the Same Role for Extended Periods

This query identifies employees who have stayed in the same job role for more than four years. This insight is valuable for understanding whether employees who remain in one role for a long time are being considered for promotions or other growth opportunities.

* **Metrics Calculated**:
  + Full Name
  + Job Role
  + Years in Most Recent Role

A close-up of a computer screen

Description automatically generated

A table of jobs

Description automatically generated with medium confidence

#### 3. Training Opportunities for Top Performers

Here, we examine how many training opportunities have been taken by top-rated employees (those with a manager rating of 4 or higher). Understanding whether top performers are actively engaging in training can provide insights into employee development efforts.

* **Metrics Calculated**:
  + Full Name
  + Job Role
  + Number of Training Opportunities Taken

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 4. Training Opportunities and Training Taken by Department and Job Role

This query provides an overview of the average number of training opportunities offered and the number of training opportunities taken by employees, grouped by department and job role. This helps assess the effectiveness of training programs and identify departments or roles with lower training engagement.

* **Metrics Calculated**:
  + Department
  + Job Role
  + Average Training Opportunities Within Year
  + Average Training Opportunities Taken
  + Employee Count

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

#### 5. Attrition Rate by Department and Job Role

This query calculates the attrition rate by department and job role. The attrition rate is the percentage of employees who have left the company out of the total number of employees in each department and role. This insight helps identify high-turnover areas, which may require further investigation.

* **Metrics Calculated**:
  + Department
  + Job Role
  + Total Employees
  + Employees Who Left
  + Current Employees
  + Attrition Rate (percentage)

A close-up of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

#### Displaying Results

For each query, the results are loaded into a Pandas DataFrame and displayed in a scrollable format, allowing for easier inspection of large datasets within the notebook environment.

A screenshot of a computer program

Description automatically generated

1. **Power BI:**

### Step 1: Connecting Power BI to SQL Server and Data Transformation

#### 1. **Connecting Power BI to SQL Server**

The initial step in this project involves connecting Power BI to a SQL Server database to load the HR datasets. By establishing this connection, we can pull the necessary data directly from SQL Server and ensure that the data is updated in real time when needed.

#### 2. **Data Transformation with Power Query**

Once the data is loaded, it's essential to prepare it for analysis. Using **Power Query** in Power BI, we clean and transform the data to ensure consistency and accuracy in the final analysis.

A screenshot of a computer

Description automatically generated

### Step 2: Data Modeling and Calculations in Power BI

After loading and transforming the data, the next step involves creating an efficient **data model** by connecting the relevant tables and establishing relationships between them. Additionally, we create specialized tables for handling dates and calculated measures, as well as parameters for slicers to enhance interactivity in the dashboard.

#### 1. **Creating the Data Model and Connecting Tables**

To ensure smooth analysis and accurate visualizations, it's important to establish relationships between different tables based on key identifiers such as EmployeeID. This step allows for seamless data retrieval and interaction across different datasets, such as employee information, satisfaction levels, and performance ratings.

A screenshot of a computer

Description automatically generated

#### 2. **Creating a Date Table**

A **Date Table** is crucial for time-based analysis, enabling insights into trends over time (e.g., performance ratings or satisfaction scores over the years). It also helps with time intelligence calculations.

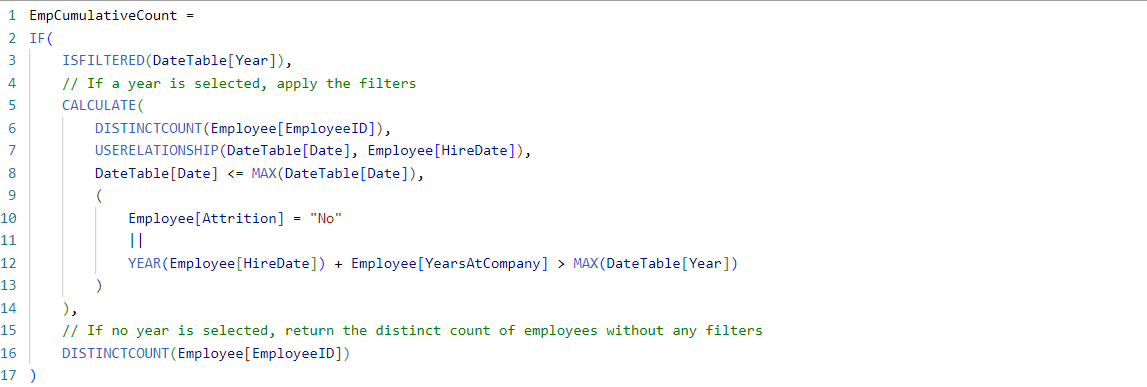
**Date Table Relationships**

The **Date Table** in this model plays a crucial role in handling time-based analysis for various employee-related events, including performance reviews, hire dates, and promotions. The table has the following relationships:

1. **Active One-to-Many Relationship with ReviewDate in PerformanceRating Table**:
   * This relationship is active and connects the Date column in the **Date Table** with the ReviewDate column in the PerformanceRating table.
   * **Purpose**: It allows for time-based analysis of performance reviews, enabling filtering and reporting by review year, month, or quarter.
2. **Inactive One-to-Many Relationship with HireDate in Employee Table**:
   * This relationship is inactive by default, connecting the Date column in the **Date Table** with the HireDate column in the Employee table.
   * **Purpose**: This relationship is used when analyzing employees based on their hire dates. It can be activated as needed using USERELATIONSHIP in DAX calculations to ensure proper filtering for hire-related analyses.
3. **Inactive Many-to-Many Relationship with LastPromotionYear in Employee Table**:
   * This is an inactive many-to-many relationship between the Date column in the **Date Table** and the LastPromotionYear in the Employee table.
   * **Purpose**: This relationship is used to track employees based on the year they were last promoted. Since the relationship is many-to-many, it handles scenarios where multiple employees share the same promotion year, and is activated through DAX expressions when needed for promotion-based analysis.

#### 3. **Creating a Calculations Table for Measures**

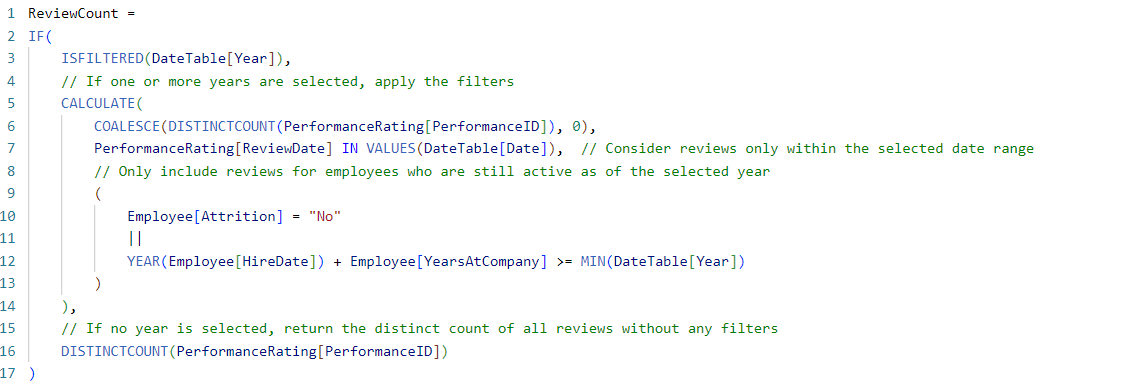
A **Calculations Table** is used to store all custom measures and calculations in a single place, keeping the data model organized. These measures are critical for performing more advanced analysis.



The EmpCumulativeCount measure is used to calculate the cumulative count of employees based on whether a specific year or range of years is selected from the DateTable. This measure ensures that only employees who were active in the selected year(s) or who left after that period are included in the count.

1. **When One or More Years are Selected**:
   * The measure uses CALCULATE to apply filtering logic based on the selected year(s) from the DateTable.
   * It uses USERELATIONSHIP to activate the relationship between DateTable[Date] and Employee[HireDate] temporarily, ensuring that employee records are filtered based on their hire date.
   * The DateTable[Date] <= MAX(DateTable[Date]) condition ensures that only employees hired on or before the selected date are counted.
   * The Attrition condition ensures that only employees who are still with the company (Attrition = "No") or employees who left the company after the selected year (determined by adding YearsAtCompany to HireYear) are included in the count.
2. **When No Year is Selected**:
   * If no year is selected, the measure defaults to returning the total distinct count of employees without applying any date filters or attrition logic.

This measure allows for dynamic filtering when a specific year or range of years is selected, providing cumulative employee counts for those who were active during the selected period. If no year is selected, it defaults to showing the total number of employees.



The ReviewCount measure is designed to calculate the distinct count of performance reviews based on whether a year or range of years is selected in the DateTable. This measure includes logic to filter reviews for employees who were still active during the selected year(s) or left the company after that period.

1. **When One or More Years are Selected**:
   * The measure uses CALCULATE to filter performance reviews based on the selected year(s).
   * It checks if the ReviewDate of each review falls within the selected date range.
   * Additionally, it ensures that only reviews for employees who were still active during the selected year(s) or who left after that year are included.
     + Active employees are determined by checking whether their Attrition status is "No" or if their calculated attrition year (hire year + years at the company) is greater than or equal to the minimum selected year.
2. **When No Year is Selected**:
   * If no specific year is selected, the measure defaults to counting all distinct performance reviews without applying any date filters or employee status conditions.

This measure is flexible, accounting for both single and multiple year selections, while ensuring that only relevant reviews from active employees (or those who left after the selected years) are included in the final count. If no year is selected, the measure returns the total distinct count of all reviews.

A computer code with many text

Description automatically generated with medium confidence

The SalarySinceLastPromotion measure calculates the average salary of employees since their last promotion, but only for employees who are still with the company or those who left after the selected year.

* **Condition**: It checks if the employee's last promotion year is before or equal to the selected year and if the employee is either still employed or left after the selected year.
* **Relationship Activation**: Uses USERELATIONSHIP to temporarily activate the relationship between LastPromotionYear and the DateTable for accurate filtering.

This measure provides the average salary of employees based on their promotion year, taking into account attrition and tenure.



The AverageSatisfactions measure calculates the average of four different satisfaction metrics for employees:

1. **Environment Satisfaction**
2. **Job Satisfaction**
3. **Relationship Satisfaction**
4. **Work-Life Balance**

* **AVERAGEX**: Iterates through each row in the PerformanceRating table, calculating the average of the four satisfaction scores.
* **COALESCE**: Ensures that if no satisfaction data is available, the measure returns 0 instead of NULL.

This measure provides a single, comprehensive satisfaction score by averaging the key satisfaction metrics for each employee.



The AverageRatings measure calculates the average of the **Manager Rating** and **Self Rating** for each employee.

* **AVERAGEX**: Iterates through the PerformanceRating table, calculating the average of the ManagerRating and SelfRating for each row.
* **COALESCE**: Ensures that if there are no ratings available (resulting in a NULL), the measure returns 0 instead of NULL.

This measure provides an overall performance rating by combining both manager and self-assessments, offering a balanced view of employee performance.

A screenshot of a computer program

Description automatically generated

A close-up of a computer screen

Description automatically generated

The CountofReviewsbySatisfaction measure counts the number of performance reviews based on the selected satisfaction metric and levels. It dynamically adjusts based on the selected satisfaction category (Environment, Job, Relationship, or Work-Life Balance) and the satisfaction levels chosen by the user.

* **VAR SelectedField**: Captures the user’s selected satisfaction type (from SatisfactionChoices).
* **VAR SelectedLevels**: Gets the list of selected satisfaction levels.
* **LevelsCount**: Checks if any satisfaction levels are selected. If none are selected, it returns 0.
* **SWITCH**: Based on the selected satisfaction category (SelectedField), the measure calculates the number of reviews for the chosen satisfaction level:
  + **Environment Satisfaction**: The active relationship is already defined, so no need for USERELATIONSHIP.
  + **Job, Relationship, and Work-Life Balance Satisfaction**: Uses USERELATIONSHIP to activate the correct relationship and filter based on the selected satisfaction levels.

This measure dynamically counts reviews based on the satisfaction metric chosen by the user, adjusting for the selected levels of satisfaction in real-time.

A screenshot of a computer program

Description automatically generated

The CountofReviewsbyRatings measure dynamically counts the number of performance reviews based on the selected **rating type** (either self-rating or manager rating) and the chosen **rating levels**.

* **VAR SelectedField**: Captures the selected rating type (from RatingChoices), which determines whether the user is looking at self-ratings or manager ratings.
* **VAR SelectedLevels**: Retrieves the specific rating levels selected by the user.
* **LevelsCount**: If no rating levels are selected, the measure returns 0.
* **SWITCH**: Based on the selected rating type (SelectedField):
  + **SelfRating**: Counts reviews based on self-rating, no need for USERELATIONSHIP since the relationship is active by default.
  + **ManagerRating**: Uses USERELATIONSHIP to temporarily activate the relationship between ManagerRating and RatingLevel[RatingID], ensuring proper filtering.

This measure allows users to filter and count reviews by different rating categories and levels, dynamically adjusting based on the user's selections.

#### 4. **Creating Parameters for Slicers (Satisfaction and Ratings)**

To allow users to dynamically filter the data based on employee satisfaction type and different performance rating, we create **parameters** for these metrics and link them to slicers in the dashboard.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Step 3: Creating an Interactive Dashboard with Multiple Views

In this final phase, we create an interactive Power BI dashboard that allows users to switch between different views—**Employee Overview**, **Satisfaction Overview**, and **Ratings Overview**—using buttons. The dashboard also includes reset buttons to clear filters and return to the default view, providing a smooth and user-friendly experience for exploring various HR metrics.

#### 1. **Creating Multiple Views**

The dashboard features three distinct views that provide comprehensive insights into the HR data:

* **Employee Overview**:
  + Displays key metrics such as total employee count, average salary, demographics (age, gender, ethnicity), and employment trends.
  + Includes visuals like bar charts, pie charts, and line graphs for a clear breakdown of employee data.
* **Satisfaction Overview**:
  + Focuses on employee satisfaction metrics, including environment, job, relationship, and work-life balance satisfaction levels.
  + Allows users to filter satisfaction metrics by department, job role, and other key categories.
* **Ratings Overview**:
  + Provides insights into performance ratings, including manager and self-ratings.
  + Includes the ability to filter by rating levels, job role, and department, with charts illustrating ratings over time and across demographic categories.

#### 2. **Switching Views Using Buttons**

To enhance interactivity, we include buttons that allow users to seamlessly switch between the different dashboard views. Each button corresponds to one of the views:

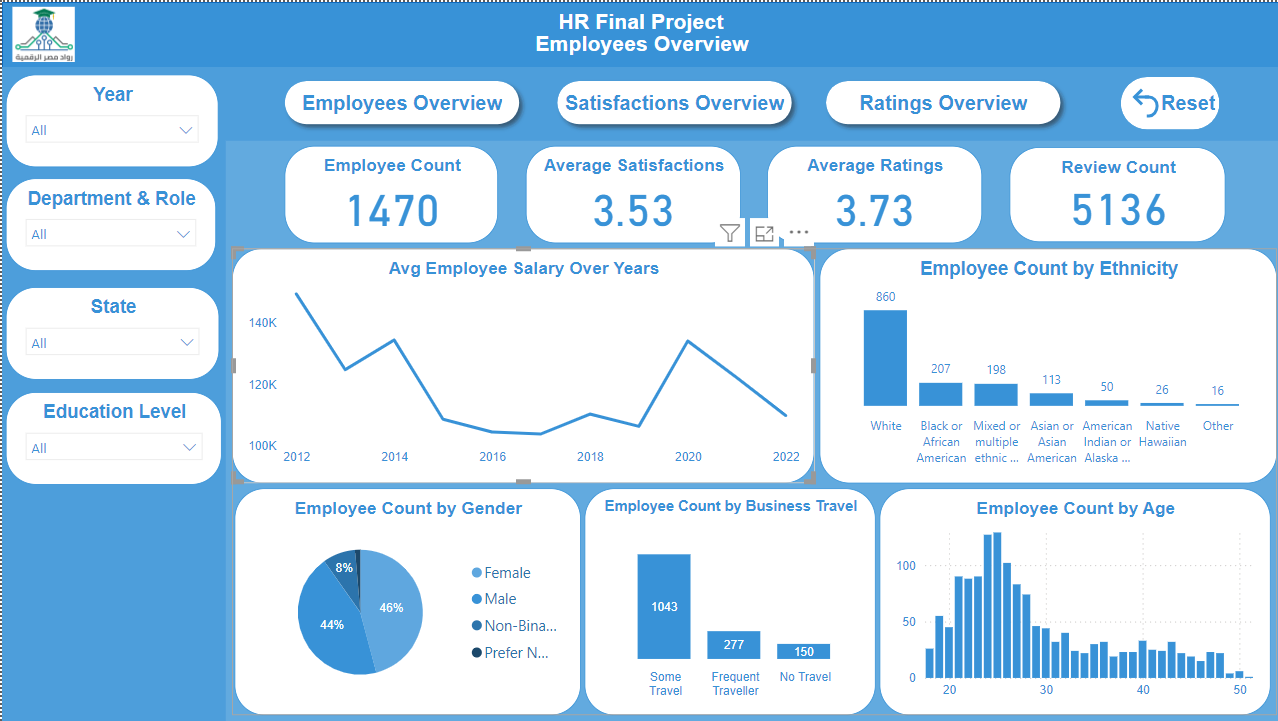
This functionality ensures users can easily navigate between the different views without overwhelming the dashboard with too many visuals at once.

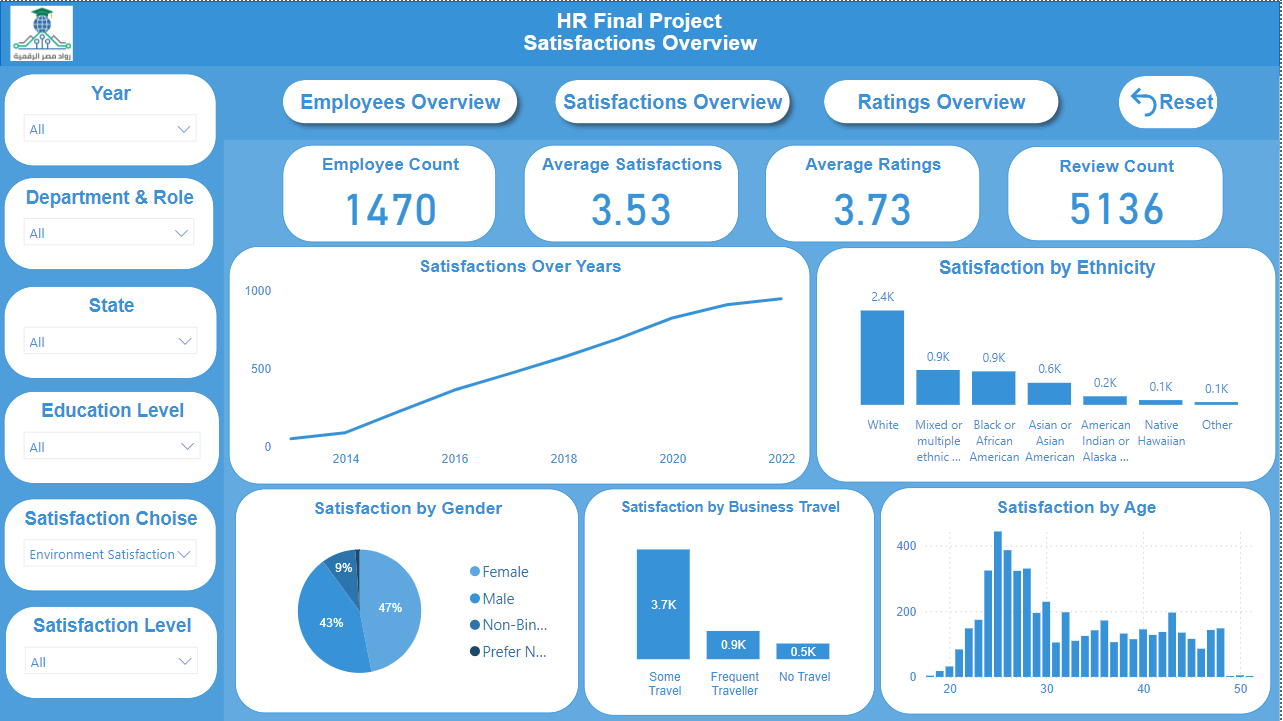
#### 3. **Adding Reset Buttons**

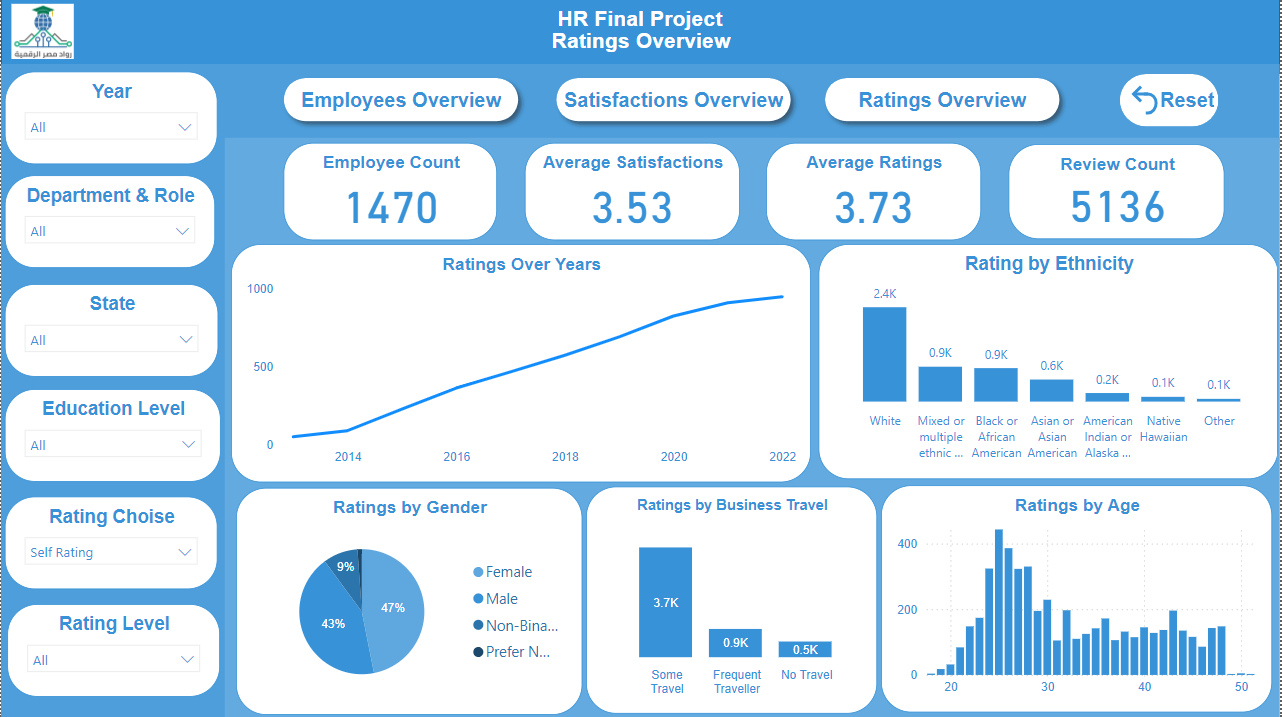
To further improve the user experience, we add **Reset Filters** buttons that allow users to clear any applied filters and return to the default state of the dashboard:

#### 4. **Final Dashboard Design**

* The dashboard is designed with clear visuals and intuitive navigation. Each view provides a focused analysis, with buttons allowing users to explore different aspects of the HR data (employees, satisfaction, and ratings).
* Users can apply filters such as department, job role, and satisfaction/rating levels, then easily switch views or reset the dashboard for further exploration.





****