Report: Historical Data Transformation

For: Data Analyst Intern(PeopleBox)
Prepared by: Sangam Sharma
sngmshrmaa@gmail.com

Introduction

This report explores the feasibility of using Microsoft Excel to analyse the given data and to transform the columnar data into a row-based historical versioning system for insertion into data warehouse of the organisation.

Objective: Transform current employee data from a columnar format into a historical, row-based versioning format suitable for database storage.

Task Overview: Your task is to convert an input CSV file containing employee data into a structured format representing historical records of employee compensation, engagement, and performance reviews. The new format requires transforming columnar data into a row-based historical versioning system for insertion into our data warehouse.

Methodology

Reading Input Data: The code begins by reading the input Excel file specified by input_file_path into a Pandas DataFrame using the pd.read_excel() function.

Sorting Data: The DataFrame is sorted by 'Employee Code' and 'Date of Joining' columns using sort_values().

Setting 'End Date': An 'End Date' column is added to the DataFrame, initialized with a far-future date (January 1, 2100). This column will be used to determine the end date of each employee's tenure.

Iterating through Rows: The code iterates through each row of the DataFrame to set the 'End Date' based on the next employee's 'Date of Joining'. It checks if the current employee code matches the next employee code. If they match, it sets the 'End Date' of the current row as the day before the next employee's 'Date of Joining'.

Transforming Data to Row-based Format: The code then iterates through each row of the DataFrame again, this time transforming the data into a row-based format suitable for analysis. For each employee, it generates rows corresponding to each year of their tenure in the organization. It fills in details such as 'Employee Code', 'Manager Employee Code', 'Last Compensation', 'Compensation', 'Last Pay Raise Date', 'Variable Pay', 'Tenure in Org', 'Performance Rating', 'Engagement Score', 'Effective Date', and 'End Date' for each row.

Converting to DataFrame: The list of dictionaries containing the transformed data is converted into a new DataFrame using pd.DataFrame().

Dropping Unnecessary Columns: The 'End Date' column is dropped from the DataFrame as it's no longer needed.

Saving Transformed Data: The transformed data is saved to a new CSV file specified by output_file_path using to_csv(). The index parameter is set to False to prevent saving the DataFrame index.

Printing Completion Message: Finally, a message "Done" is printed to indicate that the transformation process is completed.

Main Block: The if __name__ == "__main__": block ensures that the transformation function transform_data() is executed only when the script is run directly, not when it's imported as a module. It specifies the input and output file paths and calls the transform_data() function.

Assumptions

The provided Python program makes several assumptions to transform the data from the input Excel file to a CSV file. These assumptions are important to understand as they define the behavior and output of the transformation process. Here are the assumptions made by the program:

Sorting: The program sorts the data based on the 'Employee Code' and 'Date of Joining' columns. This assumption implies that the input data may not be sorted initially, and sorting is necessary for the subsequent processing steps.

End Date Calculation: The program calculates the 'End Date' for each employee's tenure by looking at the next employee's 'Date of Joining'. It assumes that the data is structured such that each employee's record is followed by the record of the next employee, sorted by 'Employee Code' and 'Date of Joining'. This assumption ensures that the 'End Date' accurately reflects the tenure duration.

Tenure Calculation: The program calculates the tenure of each employee in the organization by subtracting the 'Date of Joining' from the current date.

Yearly Transformation: The program transforms the data into a row-based format where each row represents a year of tenure for each employee. It assumes a yearly interval for the transformation, incrementing by 366 days (accounting for leap years).

Default Duration: If the tenure of an employee exceeds one year, the program assumes a default duration of one year for each transformed row. It limits the duration of each row to one year for simplicity.

Output File Format: The program assumes the output format to be a CSV file. It saves the transformed data to a CSV file specified by the output_file_path.

Understanding these assumptions is crucial for ensuring that the input data aligns with the expectations of the program and that the output meets the desired format and content. Any deviations from these assumptions may require adjustments to the program or preprocessing of the input data.

Access the Github Repository to access the code using the link:

https://github.com/SngmShrmaa1/Peoplebox_Data_Analyst_Intern_Assignment