**HISTORICAL DATA TRANSFORMATION**

**Approach:**

**1. Input Data Reading:**

* Used the pandas library to read the input CSV file into a DataFrame.
* Specified the columns containing date information to be parsed as datetime objects.

**2. Data Transformation:**

* Iterated through each row of the input dataframe.
* For each employee, processed compensation, review, and engagement data, creating rows for each period with consistent data.
* Calculated effective and end dates for each historical record, avoiding overlap by setting the end date one day before the next effective date.
* Added rows to the output list based on the specified rules.

**3. Output Data Creation:**

* Created a new DataFrame from the output list, defining column names such as Employee Code, Manager Employee Code, Last Compensation, Compensation, Last Pay Raise Date, Variable Pay, Tenure in Org, Effective Date, and End Date.

**4. Output CSV Writing:**

* Used pandas to write the output DataFrame to a new CSV file(output.csv)

**Assumptions:**

**1. Column Naming:**

* Assumed consistent naming conventions for columns in the input file (Compensation, Review, Engagement).

**2. Date Formatting:**

* Assumed that date columns in the input file are formatted consistently and could be parsed into datetime objects.

**3. Variable Pay and Tenure in Org:**

* Introduced placeholders for 'Variable Pay' and 'Tenure in Org' as they were not explicitly provided in the input data.

**4. Handling Missing Data:**

* Assumed that if data for a range is missing, values should be inherited from the most recent past record for the same employee.