**Historical Data Transformation**

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

**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 using Python. The new format requires transforming columnar data into a row-based historical versioning system for insertion into our data warehouse.

**Key Instructions:**

1. **Effective and End Dates**:
   * Derive 'Effective Date' and 'End Date' for each historical record.
   * Ensure the 'End Date' is one day before the next 'Effective Date' to avoid overlap.
   * For the latest record of an employee, assign a far-future date (e.g., 2100-01-01) as the 'End Date'.
2. **Data Transformation**:
   * Transform columnar data related to compensation, engagement, and review into a row-based format.
   * Each row should represent a specific period with consistent data.
   * If data for a range is missing, inherit values from the most recent past record for the same employee.
3. **Data Copying**:
   * Maintain unchanged values for fields without associated dates across different records.
   * Ensure all relevant data from the input file is accurately reflected in the output format.
4. **Output Format**:
   * The output should be a CSV file formatted for historical data analysis, including fields for employee identifiers, compensation, dates, performance ratings, and engagement scores.
5. **Documentation**:
   * Briefly document your approach and any assumptions made during the transformation process.

**Evaluation Criteria:**

* Accuracy of the transformation based on the provided rules.
* Clarity and efficiency of the documentation and code
* Ability to handle missing data and date ranges appropriately.