# Python Data Engineering Exercises - Banking Domain

## Scenario 1: CSV from Database + CSV from Local

Objective: Read banking transaction data from a database table (exported as CSV) and store it in a Pandas DataFrame as well as a temporary local file. Then read another CSV file from a local location containing supplementary banking customer details, join the two datasets, apply reconciliation and transformations, and generate a report.

Fields: Account\_Number, Customer\_Name, Branch\_Code, Transaction\_Date, Transaction\_Type, Transaction\_Amount, Balance, Currency, Account\_Status

### Steps:

1. Read CSV data exported from a database into a Pandas DataFrame.

2. Save this DataFrame as a temporary CSV file in a local directory.

3. Read the input CSV file from local storage containing complementary customer details.

4. Perform an inner join on 'Account\_Number'.

5. Transformations:

- Standardize 'Customer\_Name' to title case.

- Extract the year and month from 'Transaction\_Date'.

- Convert 'Currency' to uppercase.

- Filter out transactions with 'Account\_Status' = 'Closed'.

- Calculate transaction amount in USD if multi-currency.

- Group by 'Branch\_Code' and 'Transaction\_Type' to get counts and total amounts.

6. Save the output as a CSV report to local storage.

## Scenario 2: CSV from Database + Fixed Width File from Local

Objective: Same as Scenario 1, except the supplementary input file from local storage is in fixed-width format.

Fields: Account\_Number, Customer\_Name, Branch\_Code, Transaction\_Date, Transaction\_Type, Transaction\_Amount, Balance, Currency, Account\_Status

### Steps:

1. Read CSV data from database export into a Pandas DataFrame.

2. Save it as a temporary CSV file locally.

3. Read the fixed-width file (define column widths for all fields).

4. Perform join on 'Account\_Number'.

5. Transformations:

- Strip and clean 'Customer\_Name' and remove unwanted characters.

- Format 'Transaction\_Date' to 'YYYY-MM-DD'.

- Normalize 'Transaction\_Type' values to a fixed set (Deposit, Withdrawal, Transfer, etc.).

- Group by 'Currency' to sum 'Transaction\_Amount'.

6. Save final reconciliation report to local storage.

## Scenario 3: CSV + Excel from Local → Excel Output with Summary

Objective: Read a CSV file containing branch transaction data and an Excel file containing customer account details. Join the datasets, perform transformations, and create both a detailed and a summary report in Excel.

Fields: Account\_Number, Customer\_Name, Branch\_Code, Transaction\_Date, Transaction\_Type, Transaction\_Amount, Balance, Currency, Account\_Status

### Steps:

1. Read CSV file containing branch transaction data.

2. Read Excel file containing customer account details.

3. Perform join on 'Account\_Number'.

4. Transformations:

- Ensure 'Customer\_Name' is in proper case.

- Create a 'Month' column from 'Transaction\_Date'.

- Calculate running balance by 'Account\_Number'.

- Standardize 'Currency' codes.

- Filter out inactive accounts.

5. Generate two reports:

- Detailed transaction report.

- Summary report grouped by 'Branch\_Code' showing count of transactions and total 'Transaction\_Amount'.

6. Save both reports to an Excel file with two sheets: 'Detailed\_Report' and 'Summary\_Report'.