**5. To implement various data flow transformations that are commonly used in ETL processes.**

import pandas as pd

# Sample dataset

data = {

'ID': [1, 2, 3, 4, 5, 6],

'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank'],

'Age': [25, 30, 35, 40, 22, 29],

'Country': ['USA', 'UK', 'USA', 'India', 'Canada', 'UK'],

'Sales': [200, 450, 300, 800, 150, 400]

}

df = pd.DataFrame(data)

# 1. Character Map Transformation - Convert Name column to uppercase

df['Name'] = df['Name'].str.upper()

print("\nCharacter Map Transformation:\n", df)

# 2. Multicast Transformation - Create copies of dataset

df\_copy1 = df.copy()

df\_copy2 = df.copy()

print("\nMulticast Transformation - Two copies created")

# 3. Conditional Split Transformation - Filter rows where Sales > 300

df\_filtered = df[df['Sales'] > 300]

print("\nConditional Split Transformation:\n", df\_filtered)

# 4. Aggregation Transformation - Total sales per country

df\_aggregated = df.groupby('Country')['Sales'].sum().reset\_index()

print("\nAggregation Transformation:\n", df\_aggregated)

# 5. Sorting Transformation - Sort by Sales in descending order

df\_sorted = df.sort\_values(by='Sales', ascending=False)

print("\nSorting Transformation:\n", df\_sorted)

# 6. Derived Column Transformation - Categorize Sales as High or Low

df['Sales Category'] = df['Sales'].apply(lambda x: 'High' if x > 300 else 'Low')

print("\nDerived Column Transformation:\n", df)