**Lab Assignment 6: CSV In depth**

You are working as a data engineer for a large retail company. Your team is responsible for

processing and analyzing sales data from multiple stores across the country. The sales data is

stored in CSV files, and each file represents sales data for a specific month and year. Each CSV

file has the following columns:

* Date (in the format "YYYY-MM-DD")
* Store ID (a unique alphanumeric code)
* Product ID (a unique alphanumeric code)
* Quantity sold (an integer representing the number of products sold on that date)

The "product\_names.csv" file has two columns: "Product ID" and "Product Name," and it

contains the mapping for all products in the sales data.

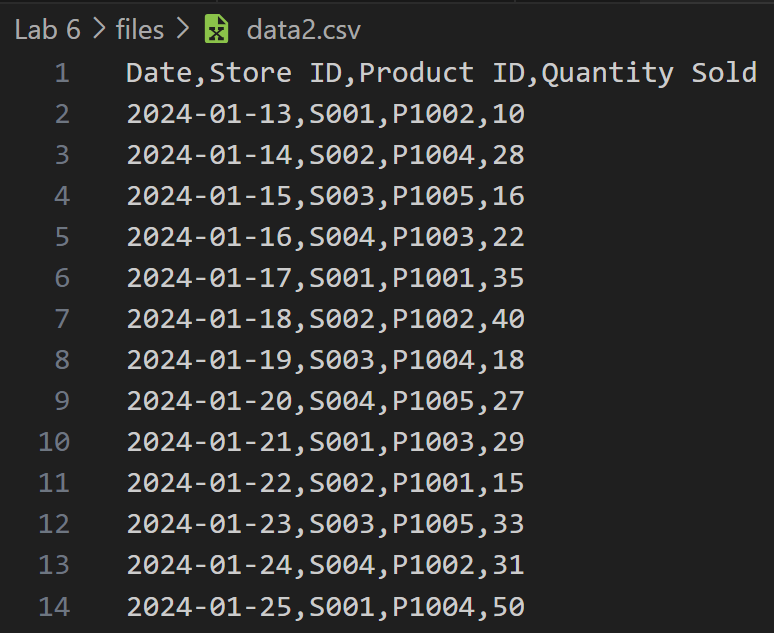
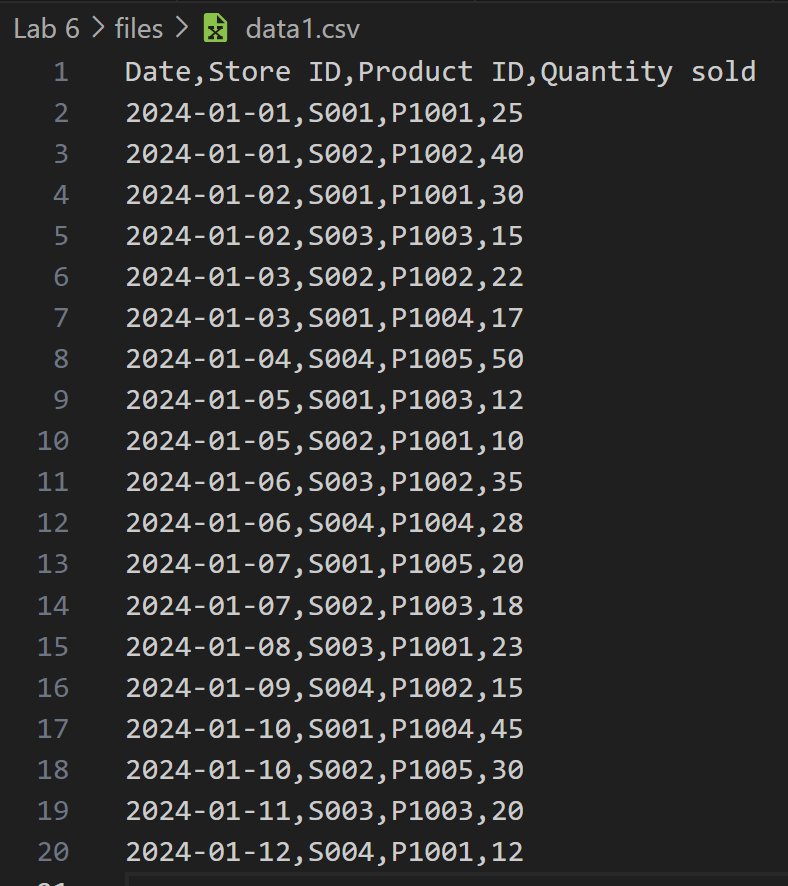
Your task is to write a Python program that performs the following operations:

* Read the sales data from all the CSV files in a given directory and its subdirectories.
* Calculate the total sales (quantity sold) for each product across all stores and all months.
* Determine the top 5 best-selling products in terms of the total quantity sold.

Create a new CSV file named "sales\_summary.csv" and write the following information into it:

* Product ID
* Product Name
* Total Quantity Sold
* Average Quantity Sold per month (considering all months available in the data

Data Set –



Code –

import os

import pandas as pd

# Directory path where the sales data CSV files are stored

sales\_data\_dir = 'D:/5th Lab/Python/Lab 6/files'

product\_names\_file = 'D:/5th Lab/Python/Lab 6/product\_names.csv'

output\_file = 'sales\_summary.csv'

# Read product names mapping file

product\_names\_df = pd.read\_csv(product\_names\_file)

# Initialize an empty DataFrame to store aggregated sales data

all\_sales\_data = pd.DataFrame()

# Walk through the directory and read all sales CSV files

for root, dirs, files in os.walk(sales\_data\_dir):

    for file in files:

        if file.endswith('.csv'):

            file\_path = os.path.join(root, file)

            sales\_data = pd.read\_csv(file\_path)

            # Convert the 'Date' column to datetime for further analysis if needed

            if 'Date' in sales\_data.columns:

                sales\_data['Date'] = pd.to\_datetime(sales\_data['Date'], format='%Y-%m-%d', errors='coerce')

            # Append the current file's sales data to the aggregated data

            all\_sales\_data = pd.concat([all\_sales\_data, sales\_data], ignore\_index=True)

# Check if there is data to process

if not all\_sales\_data.empty:

    # Group by Product ID to calculate total sales across all stores and months

    if 'Product ID' in all\_sales\_data.columns and 'Quantity sold' in all\_sales\_data.columns:

        total\_sales = all\_sales\_data.groupby('Product ID')['Quantity sold'].sum().reset\_index()

        # Merge with product names

        total\_sales = pd.merge(total\_sales, product\_names\_df, on='Product ID', how='left')

        # Calculate the number of months in the data (unique year-months in 'Date')

        if 'Date' in all\_sales\_data.columns:

            all\_sales\_data['YearMonth'] = all\_sales\_data['Date'].dt.to\_period('M')

            months\_count = all\_sales\_data['YearMonth'].nunique()

            # Calculate average sales per month for each product

            total\_sales['Average Quantity Sold per Month'] = total\_sales['Quantity sold'] / months\_count

            # Sort by total sales and get the top 5 best-selling products

            top\_5\_sales = total\_sales.nlargest(5, 'Quantity sold')

            # Save the summary to a new CSV file

            top\_5\_sales[['Product ID', 'Product Name', 'Quantity sold', 'Average Quantity Sold per Month']].to\_csv(output\_file, index=False)

            print(f"Sales summary written to {output\_file}")

        else:

            print("Error: 'Date' column is missing or incorrect in the data.")

    else:

        print("Error: Required columns 'Product ID' or 'Quantity sold' are missing in the data.")

else:

    print("No sales data found.")

Output –

