**BWT Task-07 Exercise**

**Submitted By: Abduhu Khan**

**ETL Task**

**There are some problems with the data and you need to remove them during the transformation phase to make data useful for everyone:**

**- An order id should always exist as an integer**

**- A product id cannot be 0**

**- We never had a product priced more than 1500 Rs. so any item with amount greater than 1500 Rs is an anomaly and it should be treated as 1500 Rs**

**- A status of an item can never be null or None, if it is then its an anomaly and item rows to be considered as fake orders and should not be kept in final data**

**- There must be duplication in final data**

**Code:**

import pandas as pd

from google.colab import files

# Load the data

file\_path = '/content/drive/MyDrive/dataset.csv'

df = pd.read\_csv(file\_path)

# Display the first few rows of the dataframe

print("Data BEFORE Transformation:")

print(df.head())

# 1. An order id should always exist as an integer

df = df.dropna(subset=['order\_id'])

df['order\_id'] = df['order\_id'].astype(int)

# 2. A product id cannot be 0

df = df[df['product\_id'] != 0]

# 3. We never had a product priced more than 1500 Rs. so any item with amount greater than 1500 Rs

#is an anomaly and it should be treated as 1500 Rs

df['amount'] = df['amount'].apply(lambda x: min(x, 1500))

# 4. Remove rows where `status` is null or None

df = df.dropna(subset=['status'])

# Display the cleaned data

print("\*\*\*\*\*\*\*\*\*\*\*\*Cleaned Data:\*\*\*\*\*\*\*\*\*\*\*\*")

print(df.head())

#Check datatype of OrderID column

print("\*\*\*\*\*\*\*\*DATATYPE\*\*\*\*\*\*\*\*\*\*\*\*")

print(df['order\_id'].dtype)

# Check for the duplicate rows

duplicates = df.duplicated()

duplicate\_rows = df[duplicates]

print("\*\*\*\*\*\*\*\*DUPLICATES CHECK\*\*\*\*\*\*\*\*\*\*\*\*")

has\_duplicates = duplicates.any()

print("Does the dataset have duplicates?", has\_duplicates)

print(df.dtypes)

# Step 6: Save the cleaned data to a new CSV file

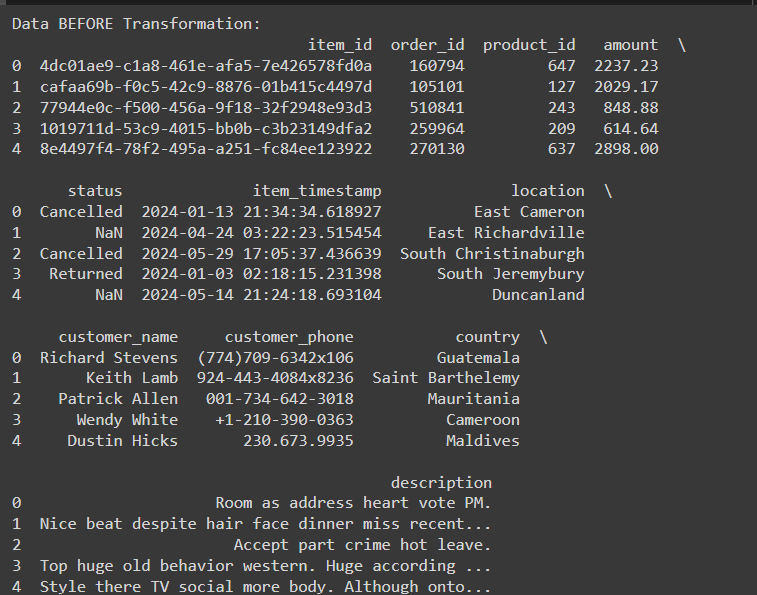
cleaned\_file\_path = '/content/cleaned\_dataset.csv'

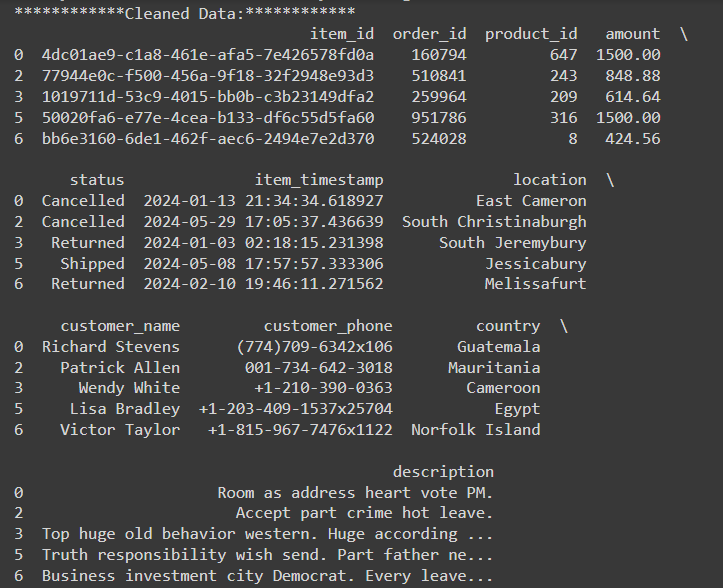
df.to\_csv(cleaned\_file\_path, index=False)

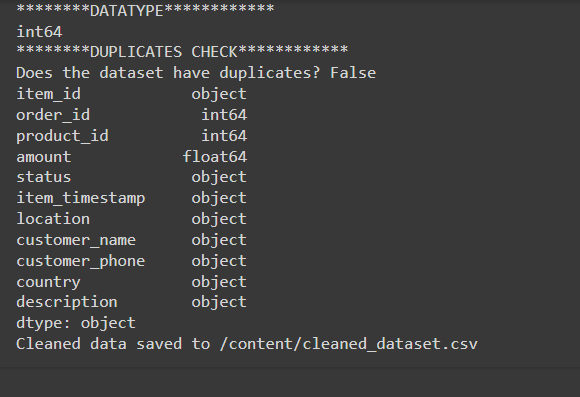
print(f"Cleaned data saved to {cleaned\_file\_path}")

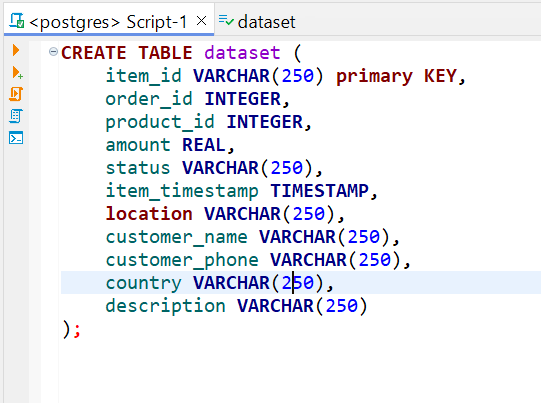
# Step 7: Download the cleaned CSV file to your local machine

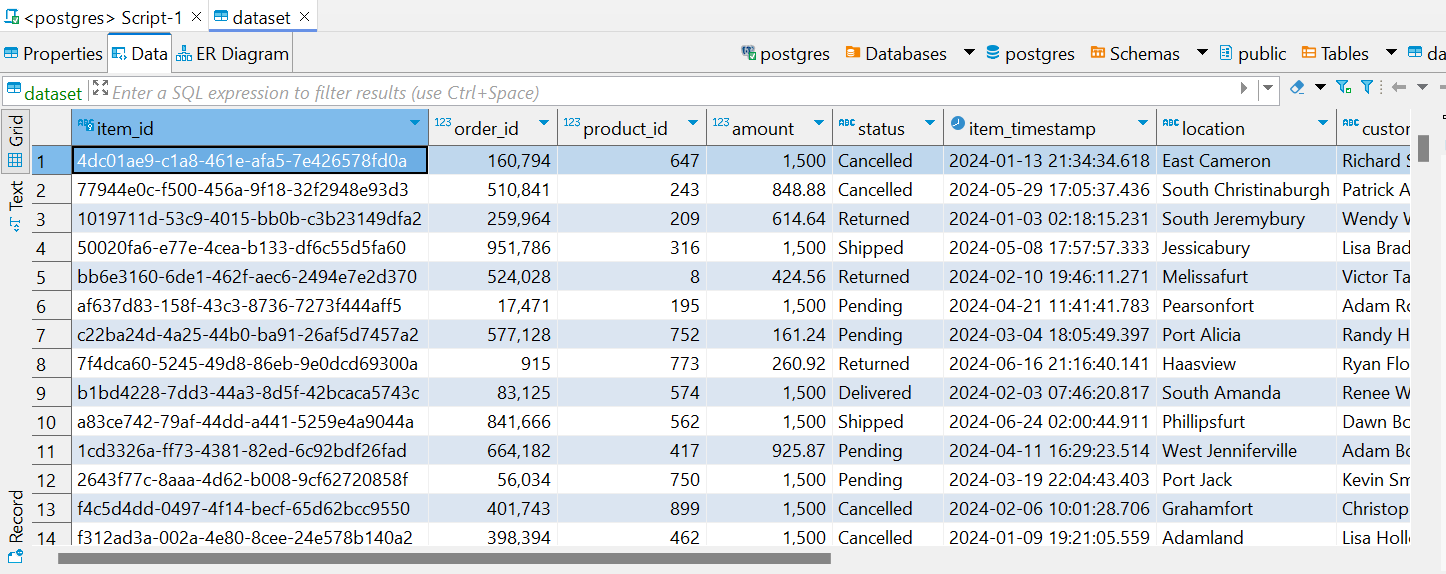
files.download(cleaned\_file\_path)

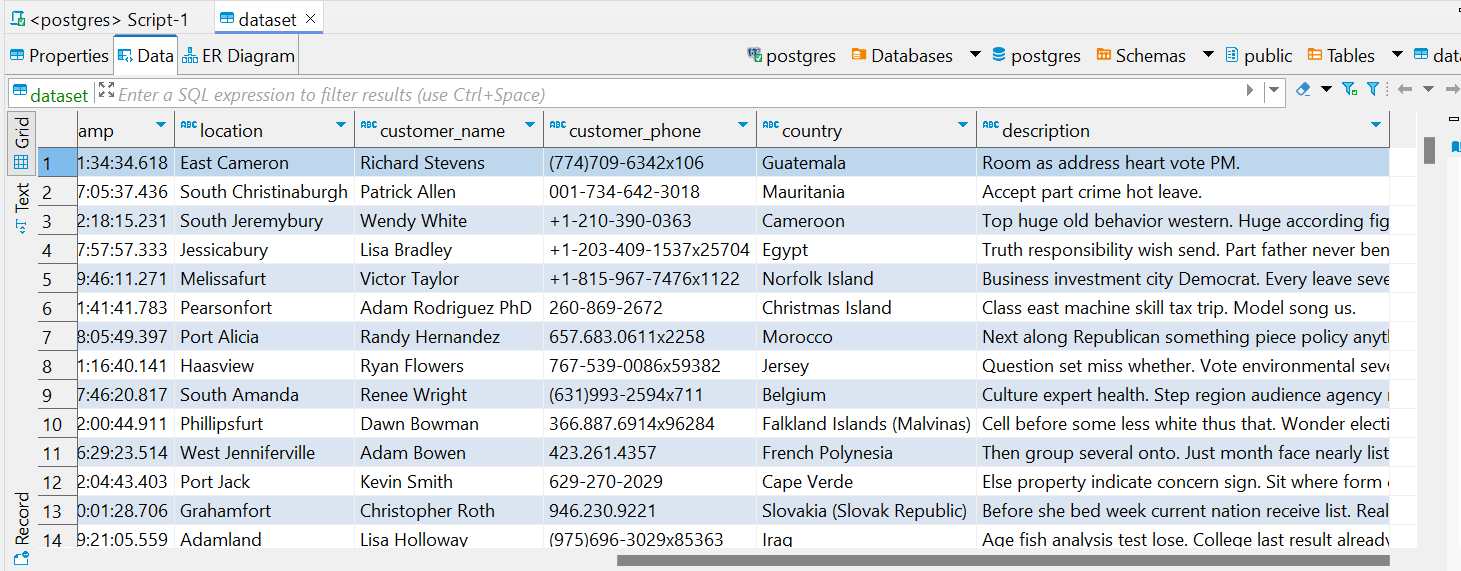
****

****

****

****

****

****

**Explanation:**

A CSV file was given for which the ETL tasks had to be performed which in this instance were to read data from dataset.csv file using pandas or PySpark. But in the above mentioned code, the data was extracted using pandas and then the necessary transformations were carried out on the csv based on the task requirements and the data was loaded in the database to view the transformed data and to use it further.