In [1]:

```
import pandas as pd
import mysql.connector
import os

# List of CSV files and their corresponding table names
csv_files = [
    ('cleaned_customers.csv', 'cleaned_customers'),
    ('cleaned_geolocation.csv', 'cleaned_geolocation'),
    ('cleaned_order_items.csv', 'cleaned_order_items'),
    ('cleaned_order_payments.csv', 'cleaned_order_payments'),
    ('cleaned_order_reviews.csv', 'cleaned_order_reviews'),
    ('cleaned_orders.csv', 'cleaned_orders'),
    ('cleaned_products.csv', 'cleaned_products'),
    ('cleaned_sellers.csv', 'cleaned_sellers')
]

# Connect to the MySQL database
conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='9345',  # Replace with your MySQL password
    database='project_olist'
)
cursor = conn.cursor()

# Folder containing the CSV files
folder_path = 'D:/Data_Science/Capstone_Projects_DS/My_Capstone_Project_DA/Domain_E-commerce/Cleaned-Dataset_Olist_Marketplace_Sales'

def get_sql_type(dtype):
    if pd.api.types.is_integer_dtype(dtype):
        return 'INT'
    elif pd.api.types.is_float_dtype(dtype):
        return 'FLOAT'
    elif pd.api.types.is_bool_dtype(dtype):
        return 'BOOLEAN'
    elif pd.api.types.is_datetime64_any_dtype(dtype):
        return 'DATETIME'
    else:
        return 'TEXT'

for csv_file, table_name in csv_files:
    file_path = os.path.join(folder_path, csv_file)

    # Read the CSV file into a pandas DataFrame
    df = pd.read_csv(file_path)

    # Handle duplicate columns if they exist (e.g., 'total_order_value_x' or 'total_order_value_y')
    if 'total_order_value_x' in df.columns:
        df.drop(columns=['total_order_value_x'], inplace=True)

    if 'total_order_value_y' in df.columns:
        df.rename(columns={'total_order_value_y': 'total_order_value'}, inplace=True)

    # Replace NaN with None to handle SQL NULL
    df = df.where(pd.notnull(df), None)

    # Debugging: Check for NaN values
    print(f"Processing {csv_file}")
    print(f"NaN values before replacement:\n{df.isnull().sum()}\n")

    # Clean column names (removing any spaces)
    df.columns = [col.replace(' ', '_').replace('-', '_').replace('.', '_') for col in df.columns]

    # Generate the CREATE TABLE statement with appropriate data types
    columns = ', '.join([f'`{col}` {get_sql_type(df[col].dtype)}' for col in df.columns])
    create_table_query = f'CREATE TABLE IF NOT EXISTS `{table_name}` ({columns})'
    cursor.execute(create_table_query)
    print(f"Created table {table_name} if not already exists.")

    # Insert DataFrame data into the MySQL table
    values = [tuple(None if pd.isna(x) else x for x in row) for _, row in df.iterrows()]

    # Insert data in batches to improve performance
    batch_size = 1000
    sql = f"INSERT INTO `{table_name}` ({', '.join(['`' + col + '`' for col in df.columns])}) VALUES ({', '.join(['%s'] * len(df.columns))})"
    for i in range(0, len(values), batch_size):
        batch = values[i:i+batch_size]
        cursor.executemany(sql, batch)
        conn.commit()
        print(f"Inserted batch {i // batch_size + 1} into {table_name}.\n")

# Close the connection
conn.close()
print("Database connection closed.")
```