In [1]:

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

# List of CSV files and their corresponding table names
csv_files = [
    ('cleaned_retail_data.csv', 'retail_data'),
]

# Connect to the MySQL database
try:
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='9345',
        database='retail_sales_data'
    )
    cursor = conn.cursor()
    if conn.is_connected():
        print("Connection to MySQL is successful")
    else:
        print("Connection failed")
except mysql.connector.Error as err:
    print(f"Error: {err}")
    exit(1)

# Folder containing the CSV files
folder_path = 'D:/Data_Science/Capstone_Projects_DS/My_Capstone_Project_DA/Domain_Retail/Cleaned_Dataset_Retail-Sales-Insights'

# Function to map pandas data types to MySQL data types
def get_sql_type(dtype, column_name=None):
    if column_name == 'Phone':
        return 'VARCHAR(50)'  # Increase length to 50 characters
    elif 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'

# Read and insert each CSV file
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)

    # Print a sample of the DataFrame to ensure it loaded correctly
    print(f"Sample data from {csv_file}:")
    print(df.head())  # Check if the data looks correct

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

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

    # Ensure the 'Phone' column is properly cleaned and converted to string
    if 'Phone' in df.columns:
        df['Phone'] = df['Phone'].astype(str).str.replace('[^0-9]', '', regex=True)  # Remove non-numeric characters
        df['Phone'] = df['Phone'].apply(lambda x: x if len(x) <= 50 else None)  # Truncate to 50 characters max

    # Create the table if it doesn't exist
    columns = ', '.join([f'`{col}` {get_sql_type(df[col].dtype, col)}' for col in df.columns])
    create_table_query = f'CREATE TABLE IF NOT EXISTS `{table_name}` ({columns})'
    cursor.execute(create_table_query)

    # Insert data in smaller batches to avoid issues
    batch_size = 1000  # Set batch size
    for i in range(0, len(df), batch_size):
        batch_df = df.iloc[i:i + batch_size]
        for _, row in batch_df.iterrows():
            values = tuple(None if pd.isna(x) else x for x in row)
            sql = f"INSERT INTO `{table_name}` ({', '.join(['`' + col + '`' for col in df.columns])}) VALUES ({', '.join(['%s'] * len(row))})"
            cursor.execute(sql, values)
        conn.commit()  # Commit after each batch

    # Print the number of records inserted
    cursor.execute(f"SELECT COUNT(*) FROM {table_name}")
    count = cursor.fetchone()[0]
    print(f"Number of records inserted into {table_name}: {count}")

# Close the connection
conn.close()

print("Data insertion complete.")
```

```
Connection to MySQL is successful
Sample data from cleaned_retail_data.csv:
   Transaction_ID  Customer_ID                Name               Email  \
0       8691788.0        37249  Michelle Harrington    Ebony39@gmailcom
1       2174773.0        69749          Kelsey Hill     Mark36@gmailcom
2       6679610.0        30192         Scott Jensen    Shane85@gmailcom
3       7232460.0        62101        Joseph Miller     Mary34@gmailcom
4       4983775.0        27901        Debra Coleman  Charles30@gmailcom

         Phone                      Address        City            State  \
0  14147868010            3959 Amanda Burgs    Dortmund           Berlin
1  68528999870           82072 Dawn Centers  Nottingham          England
2  83621604490            4133 Young Canyon     Geelong  New South Wales
3  27767517240  8148 Thomas Creek Suite 100    Edmonton          Ontario
4  90982676350    5813 Lori Ports Suite 269     Bristol          England

   Zipcode    Country  ...  Product_Category  Product_Brand Product_Type  \
0  77985.0    Germany  ...          Clothing           Nike       Shorts
1  99071.0         UK  ...       Electronics        Samsung       Tablet
2  75929.0  Australia  ...             Books  Penguin Books    Childrens
3  88420.0     Canada  ...        Home Decor     Home Depot        Tools
4  48704.0         UK  ...           Grocery         Nestle    Chocolate

    Feedback Shipping_Method  Payment_Method Order_Status Ratings  \
0  Excellent         SameDay      Debit Card      Shipped     5.0
1  Excellent        Standard     Credit Card   Processing     4.0
2    Average         SameDay     Credit Card   Processing     2.0
3  Excellent        Standard          PayPal   Processing     4.0
4        Bad        Standard            Cash      Shipped     1.0

            products  Phone_Length
0     Cycling shorts            11
1         Lenovo Tab            11
2   Sports equipment            11
3      Utility knife            11
4  Chocolate cookies            11

[5 rows x 31 columns]
Number of records inserted into retail_data: 301656
Data insertion complete.
```

In [ ]: