```
from sqlalchemy import create_engine
import urllib.parse

import pandas as pd

# Load the dataset (Replace with your actual Excel file path)
file_path = input("Enter the file path: ")
df = pd.read_excel(file_path, sheet_name="")


df = pd.read_excel(file_path, sheet_name="")

# Filter for European data
df_europe = df[df["WHO Region"].str.contains("Europe", case=False,
na=False)].copy()
df_europe = df_europe[["WHO Country Name", "City or Locality",
"Measurement Year", "PM2.5 (µg/m3)", "PM10 (µg/m3)", "NO2 (µg/m3)"]]

# List of European capitals (update if needed)
european_capitals = {
   "France": "Paris",
   "Germany": "Berlin",
   "Italy": "Roma",
   "Spain": "Madrid",
   "Switzerland": "Bern",
   "United Kingdom": "London",
   "Netherlands": "Amsterdam",
   "Belgium": "Brussel",
   "Sweden": "Stockholm",
   "Norway": "Oslo",
   "Denmark": "Copenhagen",
   "Finland": "Helsinki",
   "Poland": "Warszawa", 'Iceland': 'Reykjavik',
   "Portugal": "Lisboa",
   "Austria": "Wien",
   'Albania': 'Tirana',
   "Greece": "Athens",
   "Ireland": "Dublin", 'Bosnia and Herzegovina': 'Sarajevo',
   "Czechia": "Prague", 'Georgia': 'Tbilisi',
   "Hungary": "Budapest", 'Luxembourg': 'Luxembourg',
```

```
    "Slovakia": "Bratislava",
    "Romania": "Bucharest",
    "Bulgaria": "Sofia",
    "Serbia": "Belgrade",
    "Croatia": "Zagreb",
    "Slovenia": "Ljubljana", 'Montenegro': 'Podgorica',
    "Lithuania": "Vilnius", 'Cyprus': 'Nicosia',
    "Latvia": "Riga",
    "Estonia": "Tallinn",
    "Ukraine": "Kyiv"
}

# Filter df_europe to keep only capital cities
df_capitals = df_europe[df_europe["City or
Locality"].isin(european_capitals.values())]

# Save new dataset
df_capitals.to_csv("", index=False)

# Sort data by country and year
df_capitals_sorted = df_capitals.sort_values(by=["WHO Country Name",
"Measurement Year"])
# Save sorted dataset
df_capitals_sorted.to_csv("", index=False)

import pandas as pd
import pymysql
import numpy as np

# ✅ Load CSV
csv_file_path = ""
df = pd.read_csv(csv_file_path)

# ✅ Ensure column names exactly match MySQL table
df.rename(columns={
    'WHO Country Name': 'WHO Country Name',
    'City or Locality': 'City or Locality',
    'Measurement Year': 'Measurement Year',
    'PM2.5 (µg/m3)': 'PM2.5 (µg/m3)',
    'PM10 (µg/m3)': 'PM10 (µg/m3)',
```

```
    'NO2 (µg/m3)': 'NO2 (µg/m3)'
}, inplace=True)

# ✅ Convert NaN to None (NULL in SQL)
df = df.replace({np.nan: None})

# ✅ Connect to MySQL
conn = pymysql.connect(
        host="your_host",
        user="your_username",
        password="your_password",
        database="your_database"
)
cursor = conn.cursor()

# ✅ Define SQL Query with Correct Backticks
insert_query = """
INSERT INTO PollutionData (`WHO Country Name`, `City or Locality`,
`Measurement Year`, `PM2.5 (µg/m3)`, `PM10 (µg/m3)`, `NO2 (µg/m3)`)
VALUES (%s, %s, %s, %s, %s, %s);
"""

# ✅ Insert Each Row from CSV into MySQL
for _, row in df.iterrows():
    cursor.execute(insert_query, tuple(row))

# ✅ Commit Changes and Close Connection
conn.commit()
cursor.close()
conn.close()

print("✅ CSV successfully inserted into MySQL!")
```