Python Scripts for project Data

Python Version : 3.12

MySql Version : 8.0

IDE :- PyCharm

MySql Credentials:-

      User   : root

      Password : ---------

      Database : Event_project

      Tables Name: society_data, area_data, resident, Near_data

1) **society_data - Collected from society**
2) **area_data - Marathi Data**
3) **Resident - 78k data + area_data**
4) **Near_data**

Python Libraries :

1) **pandas**
2) **Numpy**
3) **mysql.connector**

1) Import Data in NEAR_DATA Table

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

excel_file_path = 'output.xlsx'

data = pd.read_excel(excel_file_path)

db_connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='-----',
    database='Event_project'
)

cursor = db_connection.cursor()

create_table_query = '''
CREATE TABLE IF NOT EXISTS near_DATA (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(2000),
    Mobile_No TEXT,
    Address TEXT,
    Event_Name TEXT
)
'''
cursor.execute(create_table_query)

for _, row in data.iterrows():
    row = row.where(pd.notnull(row), None)

    insert_query = '''
    INSERT INTO near_DATA (name, First_Name, Middle_Name, Last_Name,
Mobile_No, Address, Event_Name)
    VALUES (%s, %s, %s, %s,%s,%s,%s)
    '''
    cursor.execute(insert_query,
(row['Name'],row['First_Name'],row['Middle_Name'],row['Last_Name'],
row['Mobile_No'], row['Address'], row['Event_Name']))

db_connection.commit()

cursor.close()
db_connection.close()
```

2) Split the full name in separate columns (First Name, Middle Name, Last Name)

```python
import pandas as pd

def separate_names(name):
    parts = name.split()
    if len(parts) == 1:
        return pd.Series([parts[0], '', ''])
    elif len(parts) == 2:
        return pd.Series([parts[0], '', parts[1]])
    elif len(parts) == 3:
        return pd.Series([parts[0], parts[1], parts[2]])
    elif len(parts) == 4:
        return pd.Series([parts[0], f"{parts[1]} {parts[2]}", parts[3]])
    else:
        return pd.Series([name, '', ''])

df = pd.read_excel('Master Data.xlsx')

df[['First Name', 'Middle Name', 'Last Name']] = df['Name'].apply(separate_names)

df.to_excel('output.xlsx', index=False)
```

3) Import Near Data in resident Table

```python
import pandas as pd
import mysql.connector
import numpy as np

site_df = pd.read_excel("area.xlsx")

site_df.replace({np.nan: None}, inplace=True)

db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '------',
    'database': 'Event_project',
}

connection = mysql.connector.connect(**db_config)
cursor = connection.cursor()

insert_query = """
    INSERT INTO resident (
        uid, color,assembly, part_no, booth_no, srno, name, First_Name,
Middle_Name, Last_Name, Full_Name,First_and_last_Name ,
Last_and_First_Name, Gender, Age,
        Repeated, Dead, Mobile_1, Mobile_2, Star_Voter, Personnel, Cast,
Influence, New_Address, Society, Flat_No, Party_Name,
        Voter_Id, Booth, Address, Addresschange, House_No, Extraz_Check_1,
Assembly_No, last_modified_by, sa
    ) VALUES (
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s
    )
"""

for index, row in site_df.iterrows():
    insert_values = tuple(row)
    try:
        cursor.execute(insert_query, insert_values)
        connection.commit()  # Commit after each successful insertion
    except mysql.connector.Error as err:
        print(f"Error inserting data: {err}")
        print(f"Failed data: {insert_values}")

cursor.close()
connection.close()
```

4) Import Data into society_data

```python
import pandas as pd

import mysql.connector
import numpy as np

site_df = pd.read_csv("April.csv")

site_df.replace({np.nan: None}, inplace=True)

db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '------',
    'database': 'Event_project',
    'charset': 'utf8mb4',
    'collation': 'utf8mb4_unicode_ci'
}

connection = mysql.connector.connect(**db_config)
cursor = connection.cursor()

insert_query = """
    INSERT INTO resident (
        id, color, assembly, part_no, booth_no, srno, name,
First_Name, Middle_Name, Last_Name, Gender, Age,
        Repeated, Dead, Mobile_1, Mobile_2, Star_Voter, Personnel,
Cast, Influence, New_Address, Society, Flat_No, Party_Name,
        Voter_Id, Booth, Address, Addresschange, House_No,
Extraz_Check_1, Assembly_No, last_modified_by, sa
    ) VALUES (
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s
    )
"""

check_query = "SELECT COUNT(*) FROM resident WHERE name = %s"

for index, row in site_df.iterrows():
    name = row['name']
    cursor.execute(check_query, (name,))
    result = cursor.fetchone()
```

```python
        if result[0] == 0:
            insert_values = tuple(row)
            try:
                cursor.execute(insert_query, insert_values)
            except mysql.connector.Error as err:
                print(f"Error: {err}")
                continue

    connection.commit()
    cursor.close()
    connection.close()
```

5) Import 70k data in the resident table

```python
import pandas as pd

import mysql.connector
import numpy as np

site_df = pd.read_csv("70k data.csv")

site_df.replace({np.nan: None}, inplace=True)

db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '------',
    'database': 'Event_project',
    'charset': 'utf8mb4',
    'collation': 'utf8mb4_unicode_ci'
}

connection = mysql.connector.connect(**db_config)
cursor = connection.cursor()

insert_query = """
    INSERT INTO resident (
        id, color,assembly, part_no, booth_no, srno, name,
First_Name, Middle_Name, Last_Name, Gender, Age,
        Repeated, Dead, Mobile_1, Mobile_2, Star_Voter, Personnel,
Cast, Influence, New_Address, Society, Flat_No, Party_Name,
        Voter_Id, Booth, Address, Addresschange, House_No,
Extraz_Check_1, Assembly_No, last_modified_by, sa
    ) VALUES (
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s
```

```python
        )
    """

    for index, row in site_df.iterrows():
        insert_values = tuple(row)
        try:
            cursor.execute(insert_query, insert_values)
        except mysql.connector.Error as err:
            print(f"Voter_id")
            continue

connection.commit()

cursor.close()
connection.close()
```