import pandas as pd import psycopg2 from psycopg2.extras import execute_values import os import re

File and table config

file_path = 'C:/Users/moham/Desktop/Data Analysis/2025 Data Analysis/SQL Zero - Hero/Indian Election Result 2024/constituencywise_details.csv' table_name = 'constituencywise_details'

Map pandas dtype to SQL

def get_sql_type(dtype): if pd.api.types.is_integer_dtype(dtype): return 'INTEGER' elif pd.api.types.is_float_dtype(dtype): return 'REAL' elif pd.api.types.is_bool_dtype(dtype): return 'BOOLEAN' elif pd.api.types.is_datetime64_any_dtype(dtype): return 'TIMESTAMP' else: return 'TEXT'

try: # ☑ Database connection conn = psycopg2.connect(host='localhost', user='postgres', password='mohammad', dbname='Indian Election Result', port='5432') cursor = conn.cursor() print("☑ Connected to the database.")

```
# ☑ File check
if not os.path.exists(file_path):
    raise FileNotFoundError(f"❌ File not found: {file_path}")

# ☑ Load and clean DataFrame
df = pd.read_csv(file_path, encoding='ISO-8859-1')
df = df.where(pd.notnull(df), None)

# ☑ Clean column names
df.columns = [re.sub(r'\W+', '_', col.strip()) for col in df.columns]

# ☑ Add auto-increment ID
df.insert(0, 'id', range(1, len(df) + 1))

# ☑ Create SQL table
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},
        PRIMARY KEY ("id")
    );
"""
cursor.execute(create_table_query)
print(f"🛠 Table `{table_name}` created or already exists.")

# ☑ Prepare insert query
column_names = ', '.join([f'"{col}"' for col in df.columns])
insert_query = f'INSERT INTO "{table_name}" ({column_names}) VALUES %s'
```

```
# ☑ Sanitize data: replace '%' to prevent formatting errors
data = [
    [str(item).replace('%', ' percent') if isinstance(item, str) else item for
item in row]
    for row in df.values.tolist()
]

# ☑ Bulk insert
execute_values(cursor, insert_query, data)
conn.commit()
print(f"☑ Inserted {len(data)} rows into `{table_name}`.")
```

except psycopg2.Error as db_err: print(f" ❗ Database error: {db_err}") except Exception as ex: print(f" ❗ General error: {ex}") finally: if 'cursor' in locals(): cursor.close() if 'conn' in locals(): conn.close() print(" 🔒 Database connection closed.")