Made changes.

```python
import pandas as pd
import sqlite3

# 📂 Leer el archivo CSV
csv_file = "healthcare_dataset.csv"  # Ruta del archivo CSV
df = pd.read_csv(csv_file)

# Mostrar las primeras filas del archivo CSV
print("🔍 Primeras filas del archivo CSV:")
print(df.head())
```

```
🔍 Primeras filas del archivo CSV:
            Name  Age  Gender Blood Type Medical Condition Date of
Admission  \
0  Bobby JacksOn   30    Male         B-            Cancer
2024-01-31
1   LesLie TErRy   62    Male         A+           Obesity
2019-08-20
2    DaNnY sMitH   76  Female         A-           Obesity
2022-09-22
3   andrEw waTtS   28  Female         O+          Diabetes
2020-11-18
4  adrIENNE bEll   43  Female        AB+            Cancer
2022-09-19

             Doctor                     Hospital Insurance Provider  \
0     Matthew Smith             Sons and Miller         Blue Cross
1   Samantha Davies                      Kim Inc           Medicare
2  Tiffany Mitchell                     Cook PLC              Aetna
3       Kevin Wells  Hernandez Rogers and Vang,           Medicare
4    Kathleen Hanna                 White-White              Aetna

   Billing Amount  Room Number Admission Type Discharge Date
Medication  \
0    18856.281306          328         Urgent     2024-02-02
Paracetamol
1    33643.327287          265      Emergency     2019-08-26
Ibuprofen
2    27955.096079          205      Emergency     2022-10-07
Aspirin
3    37909.782410          450       Elective     2020-12-18
Ibuprofen
4    14238.317814          458         Urgent     2022-10-09
Penicillin

   Test Results
0        Normal
1  Inconclusive
```

```
2        Normal
3      Abnormal
4      Abnormal

## 2 Crear una base de datos SQLite en memoria (no se guarda en disco)
conn = sqlite3.connect(":memory:")

## 3 Cargar los datos en la base de datos en memoria
table_name = "healthcare_data"  # Nombre de la tabla temporal
df.to_sql(table_name, conn, if_exists="replace", index=False)

55500


# 4 Realizar consultas SQL directamente en los datos cargados
query = f"SELECT * FROM {table_name} LIMIT 5;"  # Consulta SQL de
ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
print("□ Resultado de la consulta SQL:")
print(df_sql)

□ Resultado de la consulta SQL:
            Name  Age  Gender Blood Type Medical Condition Date of
Admission  \
0  Bobby JacksOn   30    Male         B-            Cancer
2024-01-31
1   LesLie TErRy   62    Male         A+           Obesity
2019-08-20
2    DaNnY sMitH   76  Female         A-           Obesity
2022-09-22
3   andrEw waTtS   28  Female         O+          Diabetes
2020-11-18
4  adrIENNE bEll   43  Female        AB+            Cancer
2022-09-19

             Doctor                     Hospital Insurance Provider  \
0     Matthew Smith             Sons and Miller         Blue Cross
1   Samantha Davies                      Kim Inc           Medicare
2  Tiffany Mitchell                    Cook PLC              Aetna
3       Kevin Wells  Hernandez Rogers and Vang,           Medicare
4    Kathleen Hanna                 White-White              Aetna

   Billing Amount  Room Number Admission Type Discharge Date
Medication  \
0    18856.281306          328         Urgent     2024-02-02
Paracetamol
1    33643.327287          265      Emergency     2019-08-26
Ibuprofen
2    27955.096079          205      Emergency     2022-10-07
```

```
Aspirin
3    37909.782410          450       Elective     2020-12-18
Ibuprofen
4    14238.317814          458         Urgent     2022-10-09
Penicillin

   Test Results
0        Normal
1  Inconclusive
2        Normal
3      Abnormal
4      Abnormal
```

```python
# Renombrar la columna "Billing Amount" a "Billing_Amount"
df.rename(columns={"Billing Amount": "Billing_Amount"}, inplace=True)

# Verificar que el cambio se aplicó

print(df.head())
```

```
             Name  Age  Gender Blood Type Medical Condition Date of
Admission  \
0  Bobby JacksOn   30    Male         B-            Cancer
2024-01-31
1   LesLie TErRy   62    Male         A+           Obesity
2019-08-20
2    DaNnY sMitH   76  Female         A-           Obesity
2022-09-22
3   andrEw waTtS   28  Female         O+          Diabetes
2020-11-18
4  adrIENNE bEll   43  Female        AB+            Cancer
2022-09-19

             Doctor                     Hospital Insurance Provider  \
0     Matthew Smith             Sons and Miller         Blue Cross
1   Samantha Davies                      Kim Inc           Medicare
2  Tiffany Mitchell                    Cook PLC              Aetna
3      Kevin Wells  Hernandez Rogers and Vang,           Medicare
4    Kathleen Hanna                 White-White              Aetna

   Billing_Amount  Room Number Admission Type Discharge Date
Medication  \
0    18856.281306          328         Urgent     2024-02-02
Paracetamol
1    33643.327287          265      Emergency     2019-08-26
Ibuprofen
2    27955.096079          205      Emergency     2022-10-07
Aspirin
3    37909.782410          450       Elective     2020-12-18
Ibuprofen
```

```
4    14238.317814          458         Urgent     2022-10-09
Penicillin

   Test Results
0        Normal
1  Inconclusive
2        Normal
3      Abnormal
4      Abnormal
```

```
df.rename(columns=lambda x: x.strip().replace(" ", "_"), inplace=True)
# Reemplaza espacios por guiones bajos
print(df.columns)  # Verifica que el cambio se aplicó
```

```
Index(['Name', 'Age', 'Gender', 'Blood_Type', 'Medical_Condition',
       'Date_of_Admission', 'Doctor', 'Hospital',
'Insurance_Provider',
       'Billing_Amount', 'Room_Number', 'Admission_Type',
'Discharge_Date',
       'Medication', 'Test_Results'],
      dtype='object')
```

```
conn = sqlite3.connect(":memory:")  # Base de datos en memoria
(temporal)
df.to_sql("healthcare_data", conn, if_exists="replace", index=False)
# Sobreescribe la tabla

# Verificar que las columnas en SQLite se guardaron correctamente
query = "PRAGMA table_info(healthcare_data);"
df_columns = pd.read_sql(query, conn)
print("□ Columnas en SQLite después de renombrar:")
print(df_columns)
```

```
□ Columnas en SQLite después de renombrar:
    cid                name     type  notnull dflt_value  pk
0     0                Name     TEXT        0       None   0
1     1                 Age  INTEGER        0       None   0
2     2              Gender     TEXT        0       None   0
3     3          Blood_Type     TEXT        0       None   0
4     4   Medical_Condition     TEXT        0       None   0
5     5   Date_of_Admission     TEXT        0       None   0
6     6              Doctor     TEXT        0       None   0
7     7            Hospital     TEXT        0       None   0
8     8  Insurance_Provider     TEXT        0       None   0
9     9      Billing_Amount     REAL        0       None   0
10   10         Room_Number  INTEGER        0       None   0
11   11      Admission_Type     TEXT        0       None   0
12   12      Discharge_Date     TEXT        0       None   0
13   13          Medication     TEXT        0       None   0
14   14        Test_Results     TEXT        0       None   0
```

1 Retrieve the details of all patients diagnosed with Cancer Disease.

```
# 4 Realizar consultas SQL directamente en los datos cargados

query = f"SELECT Name, Age, Gender, Medication, Doctor,
Billing_Amount, Medication FROM {table_name} WHERE Medical_Condition =
'Cancer';"  # Consulta SQL de ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
print("□ Resultado de la consulta SQL:")
print(df_sql)
```

```
□ Resultado de la consulta SQL:
                    Name  Age  Gender   Medication
Doctor  \
0          Bobby JacksOn   30    Male  Paracetamol     Matthew Smith

1          adrIENNE bEll   43  Female   Penicillin    Kathleen Hanna

2     CHrisTInA MARtinez   20  Female  Paracetamol    Suzanne Thomas

3       ChRISTopher BerG   58  Female  Paracetamol       Heather Day

4       mIchElLe daniELs   72    Male  Paracetamol       John Duncan

...                  ...  ...     ...          ...               ...

9222      tIFfANy miller   78    Male    Ibuprofen      Jaime Valdez

9223        DEBRa MIller   17    Male    Ibuprofen  Samantha Russell

9224    mIcHAeL SanTiAgo   58  Female    Ibuprofen     Andrea Fields

9225     keNNEtH alvarez   80    Male      Aspirin     Andrew Conner

9226    STepHAniE oliVer   82    Male   Penicillin      Gary Leblanc


      Billing_Amount   Medication
0       18856.281306  Paracetamol
1       14238.317814   Penicillin
2       45820.462722  Paracetamol
3       19784.631062  Paracetamol
4       12576.795609  Paracetamol
...              ...          ...
9222    17217.325440    Ibuprofen
9223    43230.028453    Ibuprofen
9224    45767.175201    Ibuprofen
9225    45653.802310      Aspirin
9226    17350.543524   Penicillin
```

```
[9227 rows x 7 columns]
```

Explanation: This query selects the relevant details of patients diagnosed with 'Kidney Disease' by filtering the rows where the Diagnosis column matches 'Cancer'. The selected columns include name, age, gender, Medication, doctor, and the bill amount.

1. Total bill amount by hospital - Calculate the total bill amount generated by each hospital

```
query = f"SELECT Hospital, SUM(Billing_Amount) AS TOTAL_BILL FROM
{table_name} GROUP BY Hospital;"  # Consulta SQL de ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
print("🔍 Resultado de la consulta SQL:")
print(df_sql)

🔍 Resultado de la consulta SQL:
                            Hospital    TOTAL_BILL
0                        Abbott Inc  38052.041917
1                        Abbott Ltd  29877.586483
2         Abbott Moore and Williams,  24799.596339
3      Abbott and Thompson, Sullivan  16738.569765
4         Abbott, Peters and Hoffman  37684.793727
...                              ...           ...
39871             and Zimmerman Sons  32706.652625
39872      and Zuniga Davis Carlson,  42867.041298
39873   and Zuniga Francis Peterson,  33689.630726
39874                and Zuniga Sons  33950.170483
39875     and Zuniga Thompson, Blake  22067.428763

[39876 rows x 2 columns]
```

Explanation: This query groups the data by Hospital and calculates the total bill amount (SUM(Billing_Amount)) for each hospital. The result provides the sum of bill amounts for every hospital in the dataset.

1. Patients discharged after a specific date - List the patients who were discharged after January 1, 2024.

```
query = f"SELECT Name, Medical_Condition,  Discharge_Date  FROM
{table_name} WHERE  Discharge_Date > '2024-01-01';"  # Consulta SQL de
ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
print("🔍 Resultado de la consulta SQL:")
print(df_sql)

🔍 Resultado de la consulta SQL:
                Name Medical_Condition Discharge_Date
```

```
0      Bobby JacksOn            Cancer     2024-02-02
1        cathy sMaLl            Asthma     2024-01-19
2        mIchael LiU      Hypertension     2024-04-22
3          Kim ScOtt            Asthma     2024-05-04
4     MicHAEl MillEr          Diabetes     2024-02-10
...              ...               ...            ...
4282    ChRIs huGHeS           Obesity     2024-03-14
4283  mIsTy RICharDs      Hypertension     2024-04-23
4284   briTtNeY York           Obesity     2024-02-04
4285   JEssIcA WHiTe         Arthritis     2024-01-04
4286    jAMES GARCiA         Arthritis     2024-04-29

[4287 rows x 3 columns]
```

Explanation: This query filters the data to retrieve patients whose Discharge_Date is later than January 1, 2024. The output includes patient ID, name, diagnosis, and discharge date for each relevant patient.

1. Average age of patients by treatment - Calculate the average age of patients receiving each type of treatment

```
query = f"SELECT Medical_Condition, AVG(Age) AS AVERAGE_AGE  FROM {table_name} GROUP BY Medical_Condition;"  # Consulta SQL de ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
print("□ Resultado de la consulta SQL:")
print(df_sql)
```

```
□ Resultado de la consulta SQL:
  Medical_Condition  AVERAGE_AGE
0         Arthritis    51.565320
1            Asthma    51.575830
2            Cancer    51.558795
3          Diabetes    51.554170
4      Hypertension    51.741915
5           Obesity    51.240277
```

Explanation: This query groups the data by Treatment and calculates the average age (AVG(Age)) of patients receiving each type of treatment. It helps in understanding the age distribution for each treatment type.

1. Patients with the highest bill amount - Find the top 5 patients with the highest bill amounts.

```
query = f"SELECT Name, Billing_Amount  FROM {table_name} ORDER BY Billing_Amount;"  # Consulta SQL de ejemplo
df_sql = pd.read_sql(query, conn)

# Mostrar el resultado de la consulta
```

```
print("🔍 Resultado de la consulta SQL:")
print(df_sql)
```

```
🔍 Resultado de la consulta SQL:
                    Name  Billing_Amount
0             james lUnA    -2008.492140
1           EMMA savAGE    -1660.009373
2             joHn hahN    -1520.420555
3         JOsEPH robBins    -1428.843941
4           tErRy WILSOn    -1316.618581
...                    ...             ...
55495  kathRYN GoNzales    52211.852966
55496    DAVId SanDOvaL    52271.663747
55497       kARen klInE    52373.032374
55498       kARen klInE    52373.032374
55499     tOdd CARrILlO    52764.276736

[55500 rows x 2 columns]
```

Explanation: This query sorts the data by Bill_Amount in descending order (DESC), and limits the result to the top 5 records. This helps identify the patients who have the highest treatment costs.

```
# 5 Cerrar la conexión cuando ya no se necesite
conn.close()
```