Disclaimer and caution: This script/file was generated via generative AI

Thursday February 13, 2025, without prejudice

Prompt:

'Write python code that loads a python dataframe to an SQLite3 database named "Patient_record" as a table "Liver_patients". To confirm loading, query the database to print the first entries in the table.'

Response:

'Certainly! Below is a well-structured Python code that demonstrates how to load a pandas DataFrame into an SQLite3 database named "Patient_record" as a table "Liver_patients". The code also includes a query to print the first entries in the table to confirm that the loading was successful.'

```
import pandas as pd
import sqlite3

def load_dataframe_to_sqlite(df, db_name, table_name):
    """
    Load a pandas DataFrame into an SQLite database as a specified table.

    Parameters:
    df (pd.DataFrame): The DataFrame to load into the database.
    db_name (str): The name of the SQLite database file.
    table_name (str): The name of the table to create in the database.
    """
    try:
        # Establish a connection to the SQLite database
        with sqlite3.connect(db_name) as conn:
            # Load the DataFrame into the SQLite table
            df.to_sql(table_name, conn, if_exists='replace', index=False)
            print(f"DataFrame loaded into table '{table_name}' in database
'{db_name}'.")

    except Exception as e:
        print(f"An error occurred while loading the DataFrame: {e}")

def query_table(db_name, table_name):
    """
    Query the first entries of a specified table in an SQLite database.

    Parameters:
    db_name (str): The name of the SQLite database file.
    table_name (str): The name of the table to query.
    """
    try:
        # Establish a connection to the SQLite database
        with sqlite3.connect(db_name) as conn:
            # Execute a query to select the first 5 entries from the table
            query = f"SELECT * FROM {table_name} LIMIT 5;"
```

```
            result = pd.read_sql_query(query, conn)
            print("First entries in the table:")
            print(result)

    except Exception as e:
        print(f"An error occurred while querying the table: {e}")

# Example usage
if __name__ == "__main__":
    # Create a sample DataFrame
    data = {
        'Patient_ID': [1, 2, 3],
        'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [30, 45, 50],
        'Diagnosis': ['Hepatitis', 'Cirrhosis', 'Fatty Liver']
    }
    liver_patients_df = pd.DataFrame(data)

    # Load the DataFrame into the SQLite database
    db_name = 'Patient_record.db'
    table_name = 'Liver_patients'
    load_dataframe_to_sqlite(liver_patients_df, db_name, table_name)

    # Query the table to confirm loading
    query_table(db_name, table_name)
```

Code Explanation:

```
Import Required Libraries: The code begins by importing the necessary libraries:
pandas for DataFrame manipulation and sqlite3 for database operations.

Function load_dataframe_to_sqlite:
    Takes a DataFrame, database name, and table name as parameters.
    Establishes a connection to the SQLite database and loads the DataFrame into
the specified table.
    Uses if_exists='replace' to overwrite the table if it already exists.
    Includes error handling to catch and print any exceptions that occur during
the process.

Function query_table:
    Takes a database name and table name as parameters.
    Queries the first five entries of the specified table and prints them.
    Includes error handling for potential issues during the querying process.

Example Usage:
    A sample DataFrame is created with patient data.
    The DataFrame is loaded into the SQLite database.
    The table is queried to display the first entries.
```

This code adheres to recent programming practices, includes clear variable names, and is structured for readability and maintainability.