Python SQL Basics Project in SQLite

Contents

Time required: 90 minutes

- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Your Store in Python SQLite

You created your store in the last SQL Basics tutorial. We will take that SQL code and put it into SQLite with Python.

This list is a reminder of what you created in the previous tutorial. We want the same results in the Python SQLite version.

- Create your own store!
- Your store should sell one type of thing, like clothing or bikes, whatever you want your store to specialize in.
- Create a table for all the items in your store.
- A least 5 columns for the kind of data you think you'd need.
- You should sell at least 10 items
- Use select statements
    - Display all records
    - Order your items by price and show at least one statistic about the items.
    - Display all products that have less than a quantity of 10 items.

Take SQL to Python SQLite

This is a basic guide to implementing SQL code into SQLite with Python. This is just to get you started.

You can improve upon this assignment by using some of the techniques we used last semester.

Import SQLite in Python:

Start by importing the SQLite library in your Python script or Jupyter notebook.

```
import sqlite3
```

Connect to SQLite Database: Establish a connection to an SQLite database. If the database does not exist, it will be created.

```
# Connect to SQLite database (creates a new database if it doesn't exist)
connection = sqlite3.connect('your_database.db')
```

Create a Cursor Object: Create a cursor object to interact with the SQLite database.

```
cursor = connection.cursor()
```

Execute SQL Statements: Use the cursor to execute SQL statements. You can create tables, insert data, and perform various database operations.

```
# Drop the product table as you are testing your SQL script.
SQL = "DROP TABLE IF EXISTS your_table"
cursor.execute(SQL)
# Example: Creating a table
SQL ="""
    CREATE TABLE IF NOT EXISTS your_table (
        column1 datatype1,
        column2 datatype2,
        ...
    )
"""
cursor.execute(SQL)

# Example: Inserting data
SQL = "INSERT INTO your_table (column1, column2) VALUES (?, ?)",
    ('value1', 'value2')

cursor.execute(SQL)
```

Commit Changes: After executing SQL statements, commit the changes to the database.

```
connection.commit()
```

Query the Database: You can retrieve data from the database using SELECT statements.

```
SQL = 'SELECT * FROM your_table'
cursor.execute(SQL )
data = cursor.fetchall()

for row in data:
    print(row)
```

Close the Connection: Once you are done with database operations, close the connection.

```
connection.close()
```

With Handler

A with handler allows better use in functions.

```python
# ------------------------ CREATE TABLE ----------------------------------#
    CodiumAI: Options | Test this method
    def create_table(self):
        """Create database and table if not exists."""
        # If everything inside the with sqlite3.connect is successful,
        # connect.commit() and connect.close() are automatically called
        # when the with statement exits
        # Establish a connection to the database file using a
        # with context manager
        with sqlite3.connect(self.database) as connection:

            # Create a cursor object to interact with the database
            cursor = connection.cursor()

            # SQL statement to drop 'products' table if it exists
            # SQL statement to create the 'products' table
            SQL = """
            DROP TABLE IF EXISTS products;
            CREATE TABLE products (
                prod_id       INTEGER PRIMARY KEY,
                prod_name     TEXT,
                prod_desc     TEXT,
                prod_price    REAL,
                prod_rank     INTEGER,
                prod_qty      INTEGER
            );
            """

            # Execute the SQL script
            cursor.executescript(SQL)

        # Changes are committed automatically after the with handler exits
        # All connections are closed
```

Assignment Submission

1. Attach the sql code for the project. Save the file with an sql extension.
2. Attach a screenshot showing the successful completion of the project in the SQLite Database Browser.
3. Submit in Blackboard.