# 2. Python SQLite Game Shop Tutorial - Create Table

## Contents

2. Python SQLite Game Shop Tutorial - Create Table ....1
SQL Tutorial ....1
SQLite with Python Tutorials ....1
Cursor Object ....1
SQLite DataTypes ....2
Creating Tables ....2
Tutorial 1: sql_2_tutorial_create_table.py ....3
Explanation ....4
Assignment 1: Add prod_desc and prod_rank Field ....5
Assignment Submission ....6

Time required: 60 minutes

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

## SQL Tutorial

- https://www.w3schools.com/sql/sql_create_table.asp

## SQLite with Python Tutorials

- SQLite Databases with Python - Full Course – FreeCodeCamp.org
- https://www.sqlitetutorial.net

## Cursor Object

Let's discuss the cursor object.

- The cursor object is used to make the connection to the database for executing text-based SQL queries.

- It acts as middleware or controller between the SQLite database connection and the SQL query. It is created after creating a connection to the SQLite database.
- The cursor is a control structure used to traverse the records of the database.
- All SQL commands are be executed using the cursor object only.

## SQLite DataTypes

- **NULL:** The value is a NULL value.
- **INTEGER:** Store a whole number.
- **REAL:** Floating-point value, for example, 3.14, the value of PI.
- **TEXT:** A text string. TEXT value stored using UTF-8, UTF-16BE or UTF-16LE encoding.
- **BLOB:** The value is a blob of data, i.e., binary data. It is used to store images and files.

The following Python types convert to SQLite types.

| **Python Types** | **SQLite Types** |
|---|---|
| None | NULL |
| int | INTEGER |
| float | REAL |
| str | TEXT |
| bytes | BLOB |

## Creating Tables

In an SQL database, data is stored in tables. Tables define a set of columns and fields, much like a spreadsheet. They contain 0 or more rows with data for each of the defined fields.

Let's create a table named **products** that tracks the following data. This is a data dictionary. A data dictionary is a great planning tool for a database.

| | |
|---|---|
| prod_id (primary key) | INTEGER |
| prod_name | TEXT |
| prod_price | REAL |

| | |
|---|---|
| prod_qty | INTEGER |

We will create a **products** table using SQLite3 in Python.

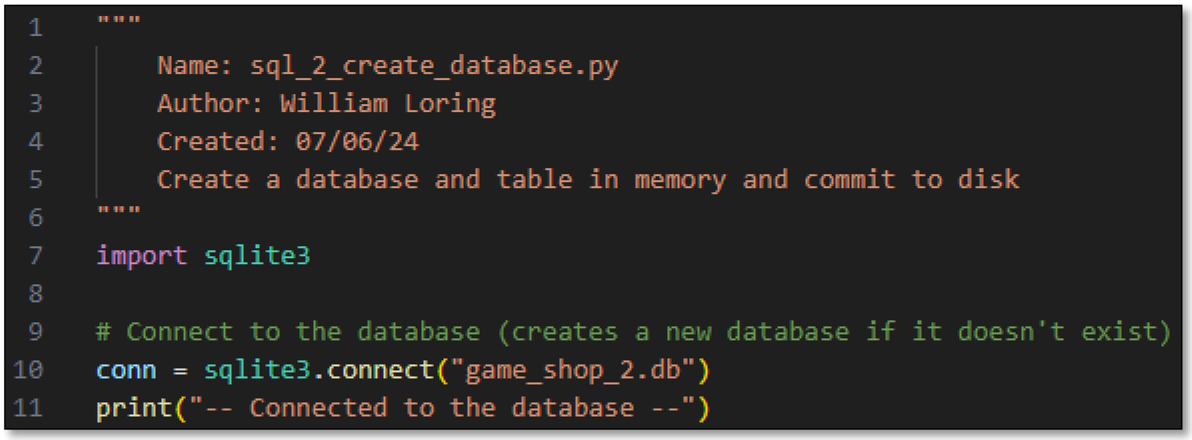
SQL is a scripting language like Python. You can assign SQL code to a String variable using a multiline string.

The result of the following code is the same as what we did earlier in DB Browser for SQLite.

## Tutorial 1: sql_2_tutorial_create_table.py

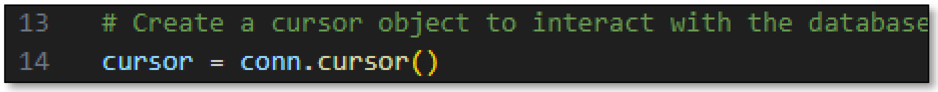
Copy your previous program and save it as **sql_2_tutorial_create_table.py**

1. The first part of the program is the same, except for the database name change.



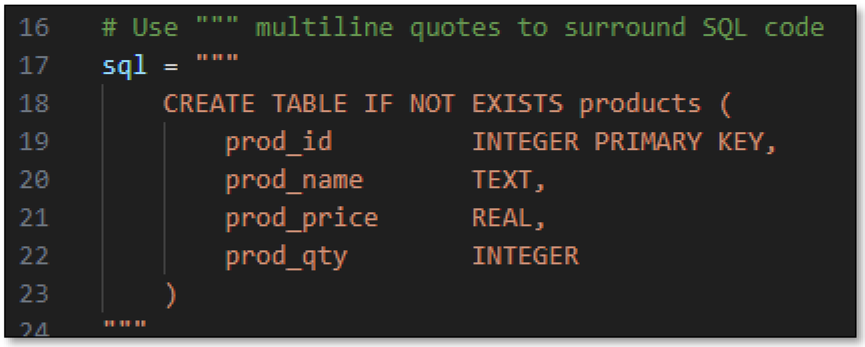
```python
"""
    Name: sql_2_create_database.py
    Author: William Loring
    Created: 07/06/24
    Create a database and table in memory and commit to disk
"""
import sqlite3

# Connect to the database (creates a new database if it doesn't exist)
conn = sqlite3.connect("game_shop_2.db")
print("-- Connected to the database --")
```

1. Create a cursor object. This is used to send SQL and other commands to the database file. It behaves much the same as the cursor when you are typing and editing a document. All the action takes place at the cursor.



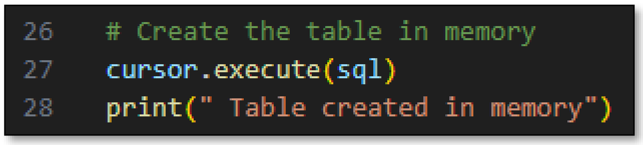
```python
# Create a cursor object to interact with the database
cursor = conn.cursor()
```

2. Assign the SQL code to a string variable. SQL is a scripting language like Python, the SQL code can be assigned to a string variable. SQL could be in a single line, the formatting below is a convention to make it easier to understand the SQL code.



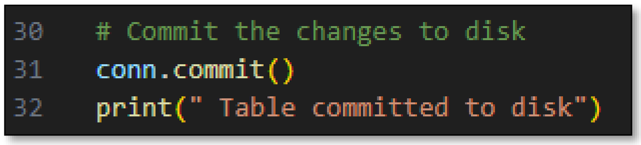
```
# Use """ multiline quotes to surround SQL code
sql = """
    CREATE TABLE IF NOT EXISTS products (
        prod_id         INTEGER PRIMARY KEY,
        prod_name       TEXT,
        prod_price      REAL,
        prod_qty        INTEGER
    )
"""
```

3. The cursor object executes the sql code.



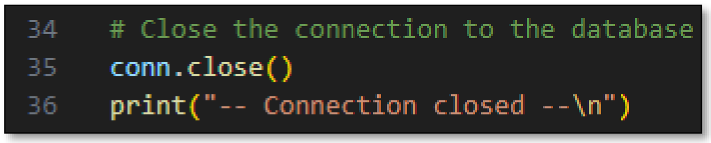
```
# Create the table in memory
cursor.execute(sql)
print(" Table created in memory")
```

4. Commit the changes to disk.



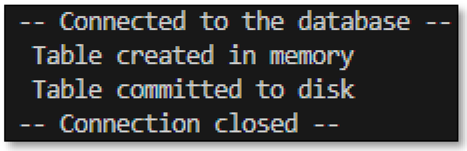
```
# Commit the changes to disk
conn.commit()
print(" Table committed to disk")
```

5. Always close the connection.



```
# Close the connection to the database
conn.close()
print("-- Connection closed --\n")
```

Example run:



```
-- Connected to the database --
 Table created in memory
 Table committed to disk
-- Connection closed --
```

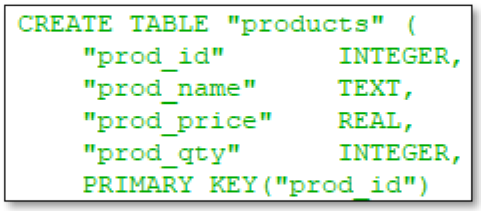
## Explanation

- The "**CREATE TABLE** “products” ..." multiline string is a SQL statement that creates a table named **products** with the columns described earlier:
    - **prod_id** of type INTEGER PRIMARY KEY
    - **prod_name** of type TEXT

    - **prod_price** of type REAL
    - **prod_qty** of type INTEGER

- A **primary key** is a unique value that provides a unique identifier for each record. A primary key is typically a sequence of whole numbers (int).
- The **cursor.executescript(SQL)** method executes the SQL commands to create the table. The **cursor.executescript()** method can execute multiple SQL statements
- **connection.commit()** writes the data to the file.

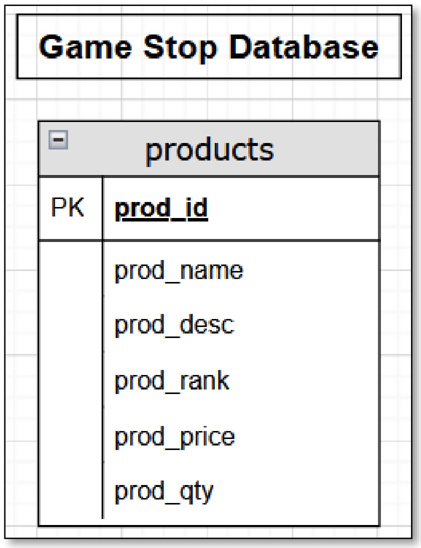
This is the SQL code.



```
CREATE TABLE "products" (
    "prod_id"       INTEGER,
    "prod_name"     TEXT,
    "prod_price"    REAL,
    "prod_qty"      INTEGER,
    PRIMARY KEY("prod_id")
```

## Assignment 1: Add prod_desc and prod_rank Field

This database diagram was drawn in app.diagrams.net  This diagram show the final version of our products table. The field **prod_desc** was added.



Here is our final data dictionary.

| Field Name | Data Type | Description |
|---|---|---|
| prod_id (primary key) | INTEGER | Unique product identifier |

|  |  |  |
|---|---|---|
| prod_name | TEXT | Product name |
| prod_desc | TEXT | Product description |
| prod_rank | INTEGER | Video game rank 1-10 |
| prod_price | REAL | Product price |
| prod_qty | INTEGER | Product quantity on hand |

1. In the tutorial program, below the **prod_name** field add the SQL code to create:
    a. field **prod_desc** of type TEXT.
    b. field **prod_rank** of type INTEGER

## Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.