

# 1. Python SQLite Game Shop Tutorial

## Contents

1. Python SQLite Game Shop Tutorial .....	1
SQL Online Tutorial .....	1
SQLite with Python Tutorials .....	1
SQLite and Python .....	2
Self-Contained .....	2
Zero-Configuration .....	3
Transactional .....	3
Single-Database .....	3
CRUD (DY) .....	3
Tutorial 1: Install SQLite DB Browser for SQLite .....	4
Tutorial 2: sql_1_tutorial_create_database.py .....	5
Assignment Submission .....	5

Time required: 60 minutes

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

## SQL Online Tutorial

- [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)
- [https://www.w3schools.com/sql/sql\\_syntax.asp](https://www.w3schools.com/sql/sql_syntax.asp)
- [https://www.w3schools.com/sql/sql\\_create\\_db.asp](https://www.w3schools.com/sql/sql_create_db.asp)

## SQLite with Python Tutorials

- [SQLite Databases with Python - Full Course](#) – FreeCodeCamp.org
- <https://www.sqlitetutorial.net>

## SQLite and Python

SQLite is a self-contained, file-based SQL database. SQLite comes bundled with Python and can be used in any of your Python applications without having to install any additional software.

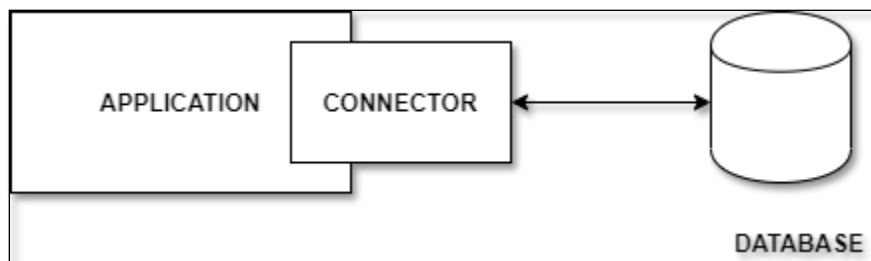
Databases offer numerous functionalities by which one can manage large amounts of information easily over the web and high-volume data input and output over a typical file such as a text file. SQL is a query language and is very popular in databases. Many websites use MySQL. SQLite is a “light” version that works over syntax very similar to SQL.

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine on the world wide web. Python has a library to access SQLite databases, called `sqlite3`, intended for working with SQLite.

SQLite has the following features.

1. Serverless
2. Self-Contained
3. Zero-Configuration
4. Transactional
5. Single-Database file

SQLite does not require a server to run. The SQLite database is joined with the application that accesses the database. SQLite database read and write directly from the database files stored on disk and applications interact with that SQLite database.



### Self-Contained

SQLite does not need any external dependencies like an operating system or external library. This feature of SQLite helps especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

## Zero-Configuration

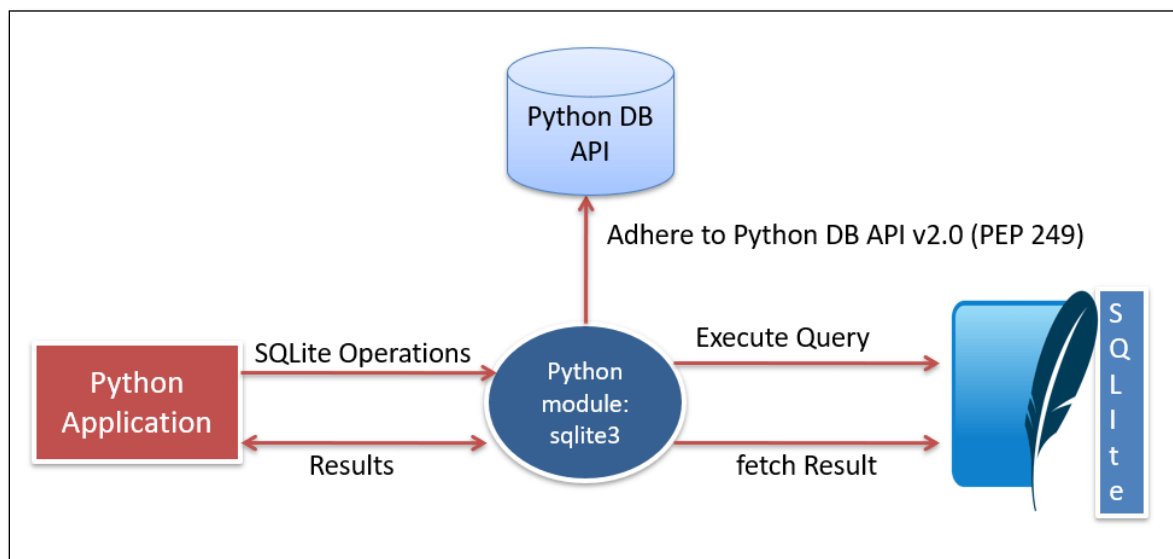
Zero-configuration means no setup or administration needed. Because of the serverless architecture, you don't need to "install" SQLite before using it. There is no server process that needs to be configured, started, and stopped.

## Transactional

Transactional means they are atomic, consistent, isolated, and durable (ACID). All transactions in SQLite are fully ACID-compliant. In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

## Single-Database

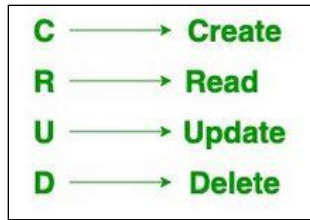
SQLite is a single database that means it allows a single database connection to access multiple database files simultaneously. These features bring many nice features like joining tables in different databases or copying data between databases in a single command. SQLite also uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.



---

## CRUD (DY)

A database provides four major operations, usually referred to as CRUD.



The corresponding SQL commands. Notice that SQL commands are typically in UPPER CASE. This is a convention; SQL does not require UPPER CASE. It makes the SQL commands easier to pick out of an SQL statement

**CREATE:** Create new tables and records

**SELECT:** Select all or selected fields and records

**UPDATE:** Modify existing tables and records

**DELETE:** Delete existing tables and records

## Tutorial 1: Install SQLite DB Browser for SQLite

This is a handy tool to look at, troubleshoot, and manipulate your database. We will use it to learn the basics of SQL.

1. Go to <https://sqlitebrowser.org>
2. Go to the **Download** tab.
3. Download the **Windows PortableApp → DB Browser for SQLite - PortableApp**
4. Double Click the installation file. Click **Next**.
5. Click **Install**. Click **Finish**.
6. You will find a new folder: **SQLiteDatabaseBrowserPortable**
7. This folder can be moved anywhere, the program will work just fine.
8. In the folder you will find **SQLiteDatabaseBrowserPortable.exe**
9. Double Click the file. Click **OK** on the warning if there is one.

We will use this program to look at our database to confirm that it is working.

## Tutorial 2: sql\_1\_tutorial\_create\_database.py

When we connect to an SQLite database, we are accessing data that resides in a file on our local computer.

1. Please create the following Python program. **sql\_1\_tutorial\_create\_database.py**
2. To use SQLite in Python, import the built-in **sqlite3** library

```
1  """
2      Name: sql_1_create_database.py
3      Author: William Loring
4      Created: 07/06/24
5      Create a database file on disk
6  """
7  import sqlite3
```

3. Create a connection object using the **connect()** method and pass the name of the database file you want to access. If there is a file with that name, it will open that file. Otherwise, Python will create a file with the given name.  
The connection object represents the database in memory.

```
9  # Connect to the database (creates a new database if it doesn't exist)
10 conn = sqlite3.connect("game_shop_1.db")
11 print("-- Connected to the database --")
```

4. Always close the connection to the database when the operation is done.

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

Please keep copies of each tutorial as you go in case something goes wrong. Use the name shown in the tutorial screenshot.

Example run:

```
-- Connected to the database --
-- Connection closed --
```

---

## Assignment Submission

1. Attach the program files.

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