

Learn SQLite in 1 Day

By Krishna Rungta

Copyright 2019 - All Rights Reserved – Krishna Rungta

ALL RIGHTS RESERVED. No part of this publication may be reproduced or transmitted in any form whatsoever, electronic, or mechanical, including photocopying, recording, or by any informational storage or retrieval system without express written, dated and signed permission from the author.

Table Of Content

Chapter 1: What is SQLite?

1. [When to use SQLite?](#)
2. [Why use SQLite?](#)
3. [SQLite limitations and Unsupported Features](#)

Chapter 2: How to Download & Install SQLite on Windows

1. [Download & Install SQLite Package Installer](#)
2. [SQLite Studio – Manager and Administration](#)
3. [Introducing Sample database](#)

Chapter 3: How to Create, Open, Backup Database in SQLite

1. [SQLite CREATE Database](#)
2. [SQLite CREATE Database in a Specific Location using Open](#)
3. [SQLite create a database and populate it with tables from a file](#)
4. [SQLite Backup & Database](#)
5. [SQLite Drop Database](#)

Chapter 4: SQLite Create, Alter, Drop Table with Examples

1. [SQLite Create table](#)
2. [Drop table](#)
3. [Alter table](#)
4. [SQLite add columns- Using **ALTER TABLE** Command](#)
5. [SQLite Insert value into a table](#)

Chapter 5: SQLite Primary Key & Foreign Key with Example

1. [SQLite constraint](#)
2. [What is SQLite Foreign KEY?](#)

Chapter 6: SQLite Data Types with Example

1. [SQLite Storage Classes](#)
2. [SQLite Affinity Type](#)
3. [Examples of Storing Data types in SQLite:](#)

Chapter 7: SQLite Query: Select, Where, LIMIT, OFFSET, Count, Group By

1. [Reading Data with Select](#)
2. [Names and Alias](#)
3. [WHERE](#)
4. [Limiting and Ordering](#)
5. [Removing duplicates](#)
6. [Aggregate](#)
7. [Group BY](#)
8. [SQLite Query & Subquery](#)
9. [Set Operations - UNION,Intersect](#)
10. [NULL handling](#)
11. [Conditional results](#)
12. [Common table expression](#)
13. [Advanced queries](#)

Chapter 8: SQLite Join Tables: Inner, Natural, Left Outer, Cross (Examples)

1. [Introduction to SQLite JOIN Clause](#)
2. [SQLite INNER JOIN](#)
3. [SQLite JOIN ... USING](#)
4. [SQLite NATURAL JOIN](#)

5. [SQLite LEFT OUTER JOIN](#)
6. [SQLite CROSS JOIN](#)

Chapter 9: SQLite INSERT, UPDATE, DELETE Query with Example

1. [SQLite INSERT](#)
2. [SQLite Update](#)
3. [SQLite Delete](#)
4. [SQLite Conflict Clause](#)

Chapter 10: SQLite Index, Trigger & View with Example

1. [SQLite View](#)
2. [SQLite Index](#)
3. [SQLite Trigger](#)

Chapter 11: SQLite String Functions: REPLACE, SUBSTR, TRIM, ROUND (Examples)

1. [Finding the LENGTH of a string](#)
2. [Changing case with UPPER Function and LOWER Function](#)
3. [SQLite SUBSTR Function](#)
4. [Changing parts of a string with SQLite REPLACE](#)
5. [Trimming blank spaces with TRIM Function](#)
6. [Reading absolute values with ABS Function](#)
7. [Rounding values with ROUND Function](#)
8. [Finding the data type of expression with TYPEOF Function](#)
9. [Finding the last inserted record with SQLite
LAST_INSERT_ROWID](#)
10. [Getting the version of your SQLite library](#)
11. [Creating user-defined functions and aggregate user-defined
functions](#)

Chapter 1: What is SQLite?

What is SQLite?

SQLite is an open-source, embedded, relational database management system, designed circa 2000. It is a lightweight database, with zero configuration, no requirements of a server or installation. Despite its simplicity, it is laden with popular features of database management systems.

Key Features

- SQLite is very lightweight (it is less than 500Kb size) compare to other database management systems like SQL Server, or Oracle. SQLite is not
- a client-server database management system. It is an in-memory library that you can call and use directly. No installation and no configuration required.
- A typical SQLite database is contained on a single file on the computer disk storage with all the database objects (tables, views, triggers, etc.) included in that file. No dedicated server required.

Despite its simplicity, it is laden with popular features of database management systems.

When to use SQLite?

- If you are developing embedded software for devices like televisions, Mobile phones, cameras, home electronic devices, etc., then SQLite is a good choice.

- SQLite can handle low to medium traffic HTTP requests and manage complex session information for a website
- When you need to store an archive of files, SQLite can produce smaller size archives and with lesser metadata included than regular ZIP archives.
- If you want to do processing on some data within an application, you can use SQLite as a temporary dataset. You can load the data into an SQLite in-memory database and execute the desired queries. You can extract the data in a format you want to display in your application.
- It gives you an easy and efficient way to process using in-memory variables. For example, you are developing a program where you have to perform calculations on some records. You can create an SQLite database and insert the records there, and with only one query, you can select the records and perform calculations.
- When you need a database system for learning and training purposes, SQLite is a good fit. As we explained earlier, no installation or configuration is required. Copy the SQLite libraries in your computer, and you are ready to learn.

Why use SQLite?

Following guide will help you determine whether you should choose SQLite for your next project

- It is free. SQLite is an open source, no commercial license required to work with it.

- SQLite is cross-platform database management system. It can be used on a broad range of platforms like Windows, Mac OS, Linux, and Unix. It can also be used on a lot of embedded operating systems like Symbian, and Windows CE.
- SQLite offers an efficient way of storing data, the length of the columns is variable and is not fixed. So SQLite will allocate only space a field needs. For example, if you have a varchar(200) column, and you put a 10 characters' length value on it, then SQLite will allocate only 20 characters' space for that value and not the whole 200 space.
- A broad range of SQLite APIs – SQLite provides APIs for a broad range of programming language, for example .Net languages (Visual Basic, C#), PHP, Java, Objective C, Python and a lot of other programming languages.
- SQLite is very flexible.
 - SQLite variables are dynamically typed, meaning that the type of the variable is not determined until it is assigned a value, and not defined at the time of declaration.
 - INSERT ON CONFLICT REPLACE statement. With this statement, you can tell SQLite to try to do an insert on a table and if it found rows with the same primary keys, then update them with the values from the inserted values.
 - With SQLite, you can work at multiple databases on the same session on the same time. Just attach those databases, and then you can access all the databases' objects (tables, views, etc..) at the same time.

SQLite limitations and Unsupported Features

The following are the list of unsupported features and limitations in SQLite:

- SQLite supports neither RIGHT OUTER JOIN nor FULL OUTER JOIN. It supports only LEFT OUTER JOIN.
- Limitations in ALTER table statement: with ALTER TABLE statement in SQLite you can only add a column or rename a table (as we will see in the following tutorials). However, you can't do the following:
 - ALTER column.
 - DROP a column.
 - ADD a constraint.
- VIEWS are read-only – you can't write INSERT, DELETE, or UPDATE statements into the view. However, you can create a trigger on a view and do the INSERT, DELETE, or UPDATE statements into it.
- GRANT and REVOKE commands are not implemented in SQLite. There are only normal file access permissions implemented in SQLite. This is because SQLite reads and writes to the disk files, unlike other Database management systems.
- TRIGGERS – As we will see in the incoming tutorials, SQLite only supports FOR EACH ROW triggers, and it doesn't support FOR EACH STATEMENT triggers.

Chapter 2: How to Download & Install SQLite on Windows

SQLite offers a lot of different installation packages, depending on your operating systems. It also offers a lot of APIs for a broad range of programming languages.

Download & Install SQLite Package Installer

Installation packages available for Windows 10 users:

From the SQLite official website in the download section. The following screenshot allows you to download different SQLite's installation packages for Windows:

Precompiled Binaries for Windows	
sqlite-shell-win32-x86-3090200.zip (364.10 KIB)	The <code>command-line_shell</code> program (version 3.9.2). (sha1: 25d78bbba37d2a0d9b9f86ed897e454ccc94d7b2)
sqlite-dll-win32-x86-3090200.zip (398.55 KIB)	32-bit DLL (x86) for SQLite version 3.9.2. (sha1: f295747951897ecdeaa59a687e7722ec513b8a05)
sqlite-dll-win64-x64-3090200.zip (678.80 KIB)	64-bit DLL (x64) for SQLite version 3.9.2. (sha1: f1f0dc69f14bea302dee93a7d49e67ff3bb379e1)
sqlite-analyzer-win32-x86-3090200.zip (695.47 KIB)	A program to analyze how space is allocated inside an SQLite database file (version 3.9.2). (sha1: 8f86b63c9e23a70994fe2f9432979c8cb066cf34)

The command line shell program:

The highlighted download package is called the **Command-Line Program (CLP)**. CLP is a command line application that let you access the SQLite database management system and all the features of

the SQLite. Using CLP, you can create and manage the SQLite database. And it is the tool that we will use throughout the tutorial.

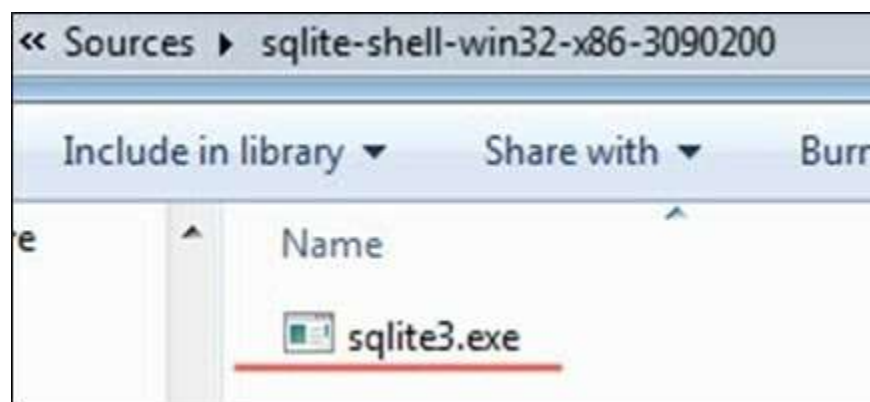
- 32-bit DLL(x86): The SQLite Database system core library for x86 platforms.
- 64-bit DLL (x64): The SQLite Database system core library for x64 platforms.

Installing the Command-Line Program (CLP) on your machine:

In the following steps, you will find the steps for how to install the Command-Line Program (CLP) on your machine:

Step 1) Download the highlighted download package from the previous image to your PC. It is a "zip" file.

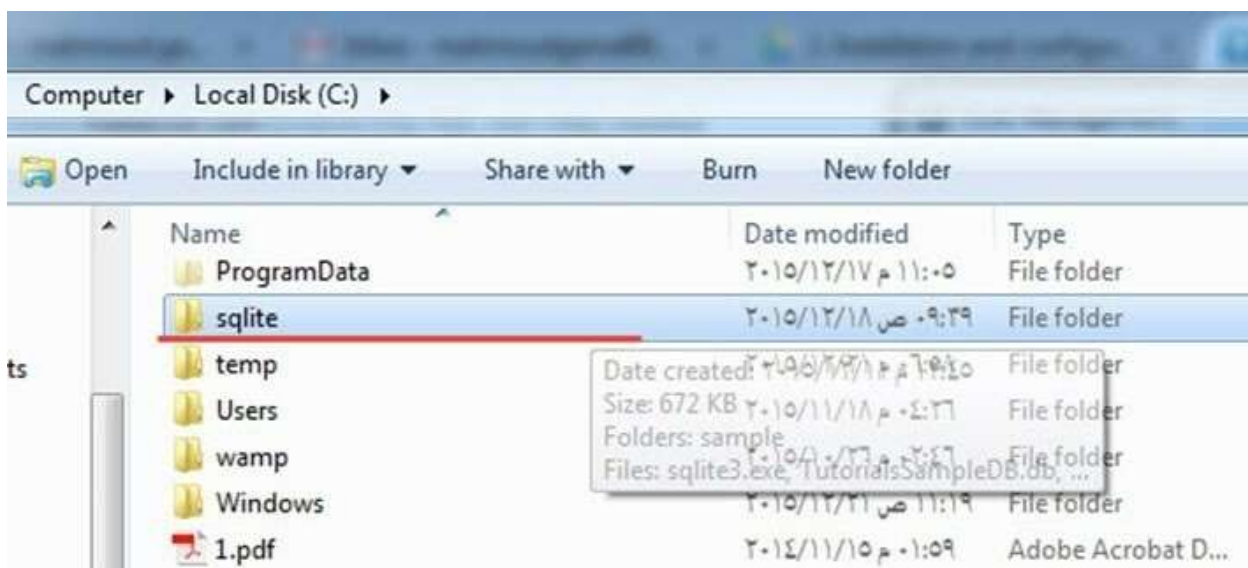
Step 2) Extract the zip file. You will find the "sqlite3.exe" in the extracted file as following:



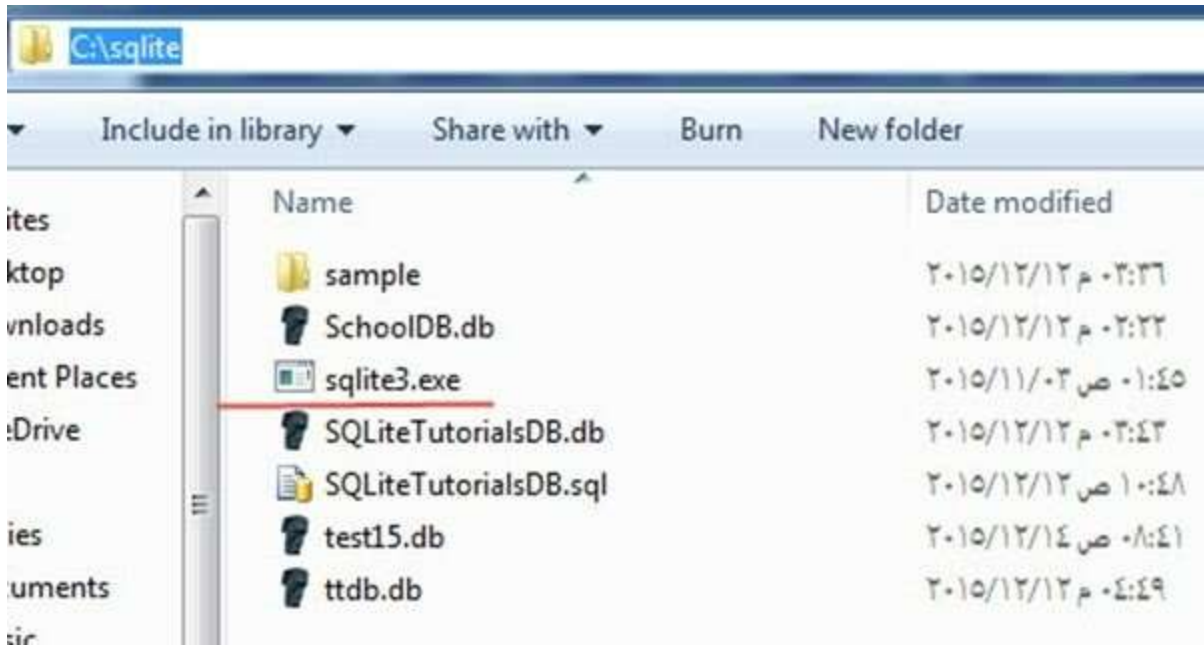
Step 3) Open My Computer, and double-click the partition "C" to navigate to it:



Step 4) Create a new directory "sqlite":



Step 5) Copy the file "sqlite3.exe" into it. This is what we will use through the tutorials to run SQLite queries:



However, there are some other packages for different purposes. They are not required. But you might need it if you are using a different OS than Windows you can get the Linux or Mac OS version of SQLite.

Also, you can get the documentation or source code from there if you wish. You can also get the API for Windows Phone 8 or .Net and other programming languages.

Here are some other different packages for different purposes:

- The Source Code and some alternative Source Code Formats – The complete source code that made up the SQLite.
- The documentation – The documentation of the SQLite as HTML pages. It is the same online documentation, but downloadable as HTML page so that you can open them offline.
- Precompiled Binaries for Linux. Precompiled
- Binaries for Mac OS X (x86).
- Precompiled Binaries for Windows Phone 8 – SDK and components to develop an application for Windows Phone 8 that

uses SQLite databases.

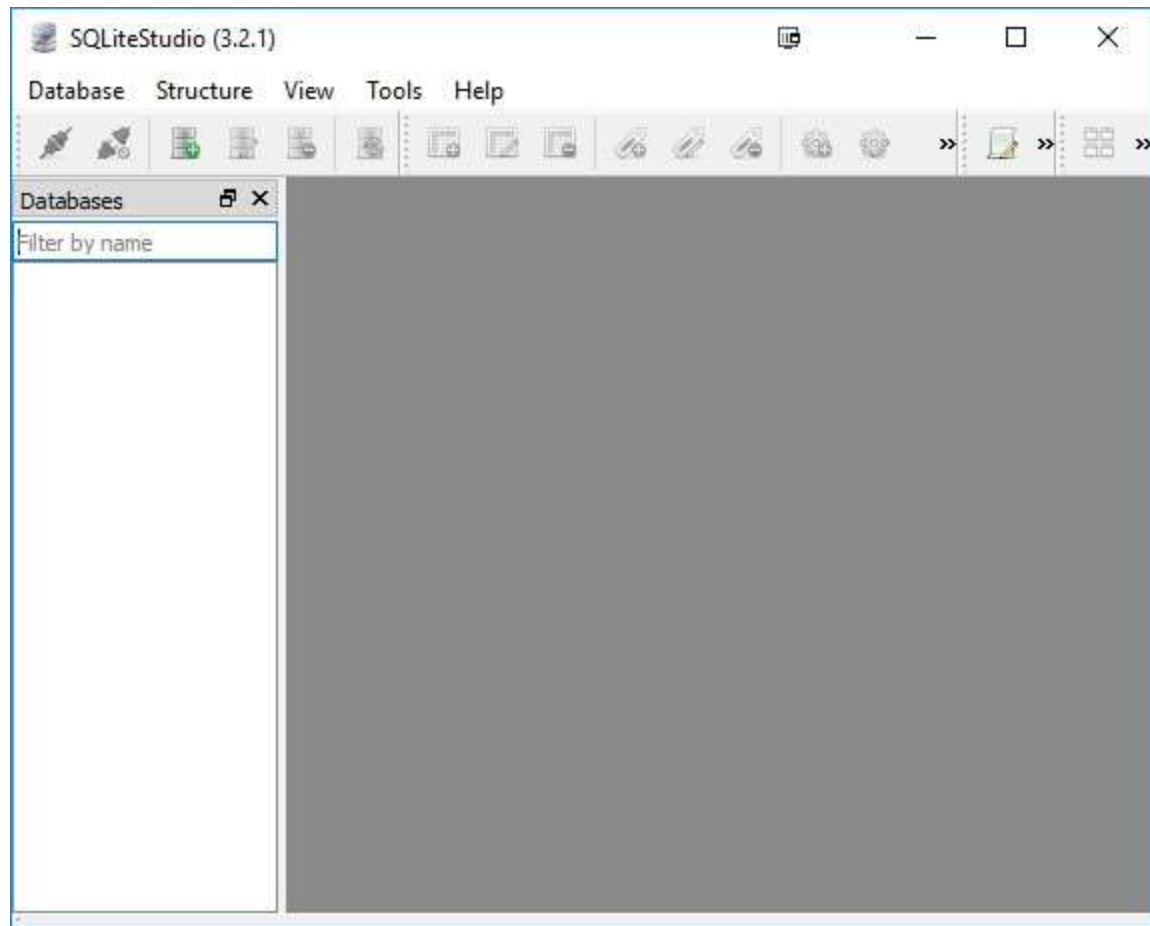
- Precompiled Binaries for Windows Runtime – SDK and other components for developing an application to connect to SQLite databases for the Windows Runtime platforms.
- Precompiled Binaries for .NET – these are some set of DLLs and .NET libraries that you can use them from .NET application to connect to SQLite databases.

SQLite Studio – Manager and Administration

There are lots of SQLite management tools that make working with SQLite databases easier. Instead of creating and managing databases using a command line, these tools provide a set of GUI tools that let you create and manage the database.

The official SQLite website has dozens of such tools listed; you can view them from here: [SQLite Management Tools](#). Here is the recommended one

SQLite Studio: It is a portable tool that doesn't require an installation. It supports both SQLite3 and SQLite2. You can easily import and export data to various formats like CSV, HTML, PDF, JSON. Its open source and supports Unicode.



Introducing Sample database

In the following steps, we will create the sample database that we will use throughout the tutorials:

Step 1) Open a text file and paste the following commands into it:

```
CREATE TABLE [Departments] (  
    [DepartmentId] INTEGER NOT NULL PRIMARY KEY,  
    [DepartmentName] NVARCHAR(50) NULL  
);  
INSERT INTO Departments VALUES (1, 'IT');  
INSERT INTO Departments VALUES (2, 'Physics');  
INSERT INTO Departments VALUES (3, 'Arts');  
INSERT INTO Departments VALUES (4, 'Math');
```

```

CREATE TABLE [Students] (
    [StudentId] INTEGER          PRIMARY KEY NOT NULL,
    [StudentName] NVARCHAR(50) NOT NULL,
    [DepartmentId] INTEGER      NULL,
    [DateOfBirth] DATE NULL,
    FOREIGN KEY(DepartmentId) REFERENCES
Departments(DepartmentId)
);
INSERT INTO Students VALUES(1, 'Michael', 1, '1998-10-12'); INSERT INTO Students
VALUES(2, 'John', 1, '1998-10-12'); INSERT INTO Students VALUES(3, 'Jack', 1, '1998-10-12');
INSERT INTO Students VALUES(4, 'Sara', 2, '1998-10-12'); INSERT INTO Students VALUES(5,
'Sally', 2, '1998-10-12'); INSERT INTO Students VALUES(6, 'Jena', NULL, '1998-10-12');
INSERT INTO Students VALUES(7, 'Nancy', 2, '1998-10-12'); INSERT INTO Students
VALUES(8, 'Adam', 3, '1998-10-12'); INSERT INTO Students VALUES(9, 'Stevens', 3, '1998-
10-12'); INSERT INTO Students VALUES(10, 'George', NULL, '1998-10-12');

CREATE TABLE [Tests] (
    [TestId] INTEGER NOT NULL PRIMARY KEY, [TestName]
    NVARCHAR(50) NOT NULL,
    [TestDate] DATE NULL
);
INSERT INTO [Tests] VALUES(1, 'Mid Term IT Exam', '2015-10-18');
INSERT INTO [Tests] VALUES(2, 'Mid Term Physics Exam', '2015-10-23');
INSERT INTO [Tests] VALUES(3, 'Mid Term Arts Exam', '2015-10-10');
INSERT INTO [Tests] VALUES(4, 'Mid Term Math Exam', '2015-10-15');

CREATE TABLE [Marks] (
    [MarkId] INTEGER NOT NULL PRIMARY KEY, [TestId]
    INTEGER NOT NULL, [StudentId] INTEGER NOT NULL,
    [Mark] INTEGER NULL,
    FOREIGN KEY(StudentId) REFERENCES Students(StudentId), FOREIGN KEY(TestId)
    REFERENCES Tests(TestId)
);

INSERT INTO Marks VALUES(1, 1, 1, 18);
INSERT INTO Marks VALUES(2, 1, 2, 20);

```

```
INSERT INTO Marks VALUES(3, 1, 3, 16);
INSERT INTO Marks VALUES(4, 2, 4, 19);
INSERT INTO Marks VALUES(5, 2, 5, 14);
INSERT INTO Marks VALUES(6, 2, 7, 20);
INSERT INTO Marks VALUES(7, 3, 8, 20);
INSERT INTO Marks VALUES(8, 3, 9, 20);
```

Step 2) Save the file as "**TutorialsSampleDB.sql**" in the following directory "**C:\sqlite**".

Step 3) Open the Windows Command Line tool (cmd.exe) from the start menu, type "**cmd**" and open it.

Step 4) It will open in the default path, you need to navigate to the "**C:\sqlite**" folder we had created earlier in this tutorial by the following command "**cd "C:\sqlite"**":

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\MGA>cd "C:\sqlite"
C:\sqlite>
```

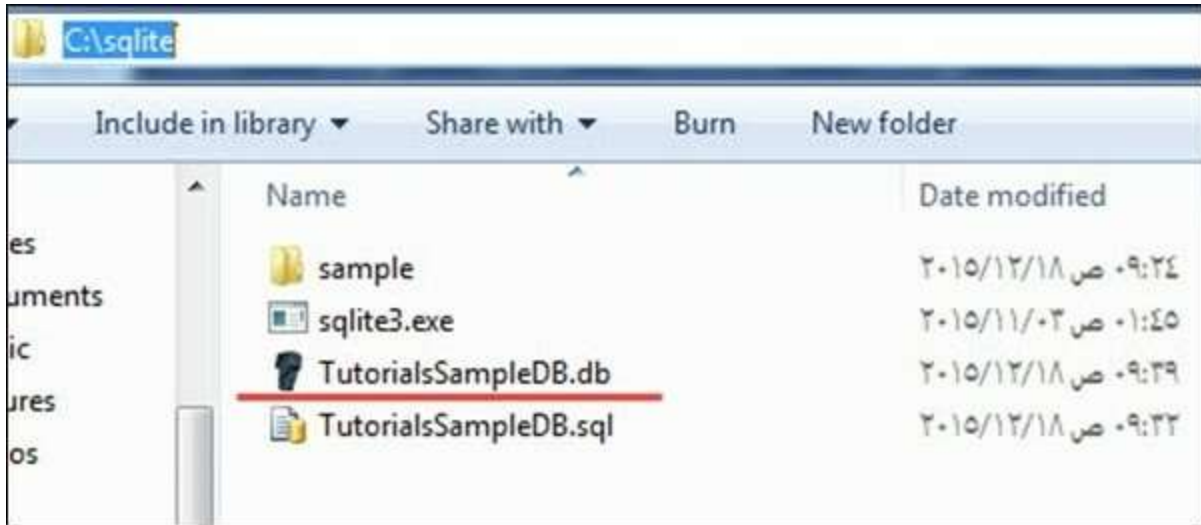
Step 5) Write the following command,

```
sqlite3 TutorialsSampleDB.db < TutorialsSampleDB.sql
```

The command should be completed successfully, and you should see no output after that command as the following screenshot:

```
C:\sqlite>sqlite3 TutorialsSampleDB.db < TutorialsSampleDB.sql
C:\sqlite>_
```

Step 6) You should now be able to see the database file "**TutorialsSampleDB.db**" created in the directory "**C:\sqlite**":



Buy Now \$9.99