

ch12 访问 SQLite 数据库

SQLite 是一个软件库，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。

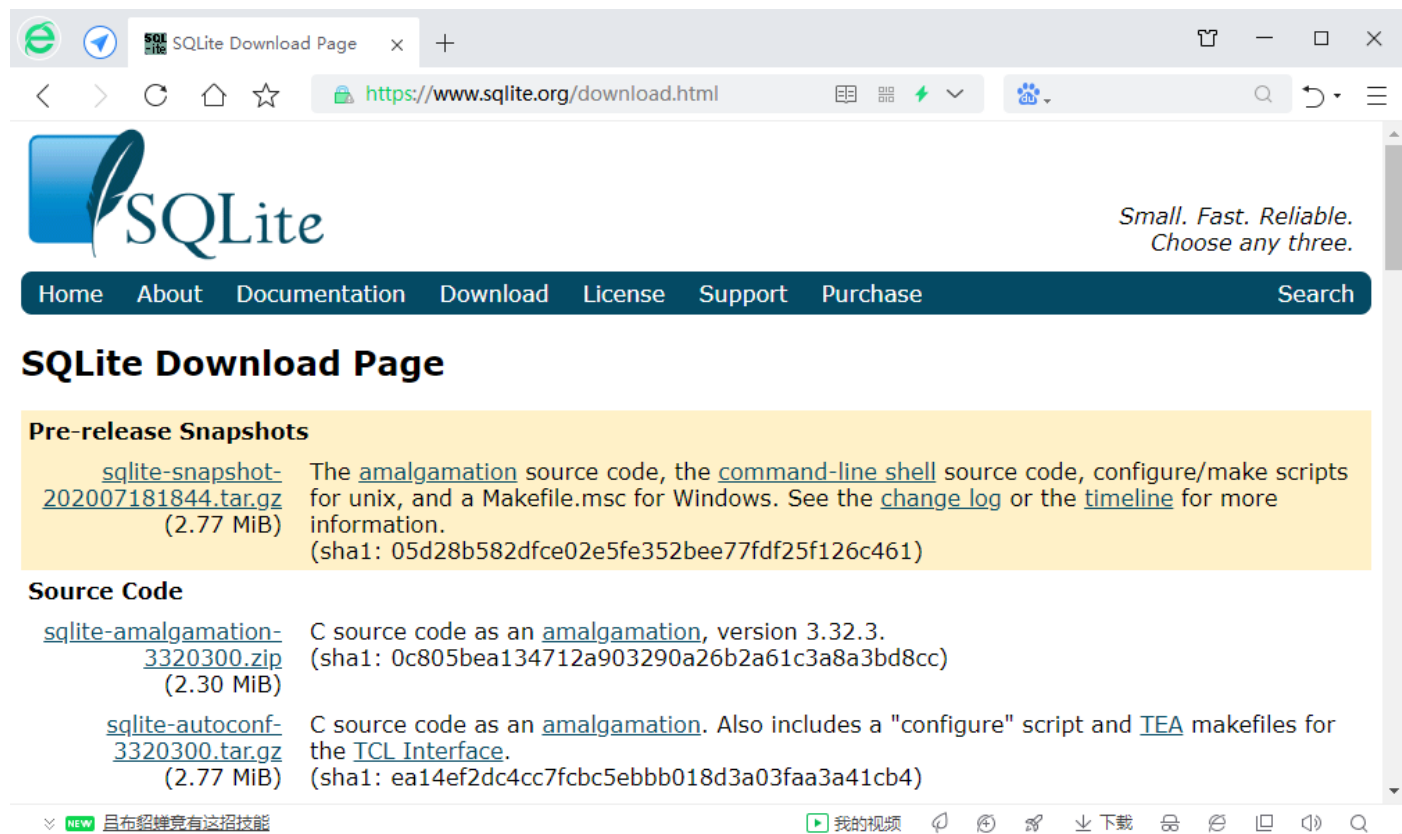
SQLite 是在世界上最广泛部署的 SQL 数据库引擎。SQLite 源代码不受版权限制。

SQLite 是一款轻型的关系型数据库管理系统，它包含在一个相对小的 C 库中。它的设计目标是嵌入式的，而且已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。它能够支持 Windows/Linux/Unix 等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java 等，还有 ODBC 接口，同样比起 Mysql、PostgreSQL 这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。SQLite 第一个 Alpha 版本诞生于 2000 年 5 月，目前最新版本是 SQLite 3.32.3。

一、SQLite3 的安装与使用

1. 下载安装

下载地址：<https://www.sqlite.org/download.html>



The screenshot shows the SQLite Download Page in a web browser. The page features the SQLite logo and the tagline "Small. Fast. Reliable. Choose any three." A navigation bar includes links for Home, About, Documentation, Download, License, Support, and Purchase. The main content area is titled "SQLite Download Page" and lists two sections: "Pre-release Snapshots" and "Source Code".

Pre-release Snapshots

- [sqlite-snapshot-202007181844.tar.gz](#) (2.77 MiB): The [amalgamation](#) source code, the [command-line shell](#) source code, configure/make scripts for unix, and a Makefile.msc for Windows. See the [change log](#) or the [timeline](#) for more information. (sha1: 05d28b582dfce02e5fe352bee77fdf25f126c461)

Source Code

- [sqlite-amalgamation-3320300.zip](#) (2.30 MiB): C source code as an [amalgamation](#), version 3.32.3. (sha1: 0c805bea134712a903290a26b2a61c3a8a3bd8cc)
- [sqlite-autoconf-3320300.tar.gz](#) (2.77 MiB): C source code as an [amalgamation](#). Also includes a "configure" script and [TEA](#) makefiles for the [TCL Interface](#). (sha1: ea14ef2dc4cc7fcbc5ebbb018d3a03faa3a41cb4)

At the bottom of the browser window, there is a status bar with a "NEW" badge, a video player icon, and various utility icons.

SQLite Download Page

<https://www.sqlite.org/download.html>

[sqlite-tools-linux-x86-3320300.zip](#) (1.95 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3_analyzer](#) program. (sha1: 8e9edf58fc6d3caf32592af0c3e39f5421fb21c7)

Precompiled Binaries for Mac OS X (x86)

[sqlite-tools-osx-x86-3320300.zip](#) (1.34 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3_analyzer](#) program. (sha1: 037dfdc13cc43768d0e7d4fa951c223f3c920342)

Precompiled Binaries for Windows

[sqlite-dll-win32-x86-3320300.zip](#) (486.71 KiB) 32-bit DLL (x86) for SQLite version 3.32.3. (sha1: b75e199f3b1f4d0d5ffe7bbe4278f98426e82e3b)

[sqlite-dll-win64-x64-3320300.zip](#) (807.17 KiB) 64-bit DLL (x64) for SQLite version 3.32.3. (sha1: 9614b03932f24c0d03fda4ef3041215f31778a26)

[sqlite-tools-win32-x86-3320300.zip](#) (1.75 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3_analyzer.exe](#) program. (sha1: e737fe2726bf8cd239e90f1d01b275d5c78a1089)

Universal Windows Platform

[sqlite-uwp-3320300.vsix](#) (7.36 MiB) VSIX package for Universal Windows Platform development using Visual Studio 2015. (sha1: 324df4190f436828a94b10a9404b972fdd6f1f6e)

<https://www.sqlite.org/2020/sqlite-tools-osx-x86-3320300.zip>

将 **Precompiled Binaries for Windows** 下的包下载下来 `sqlite-dll-win64-x64-3320300.zip`、`sqlite-tools-win32-x86-3320300.zip`

- `sqlite-dll-win64-x64-3320300.zip` 包含 `sqlite3.def`、`sqlite3.dll` 两个文件
- `sqlite-tools-win32-x86-3320300.zip` 包含 `sqlite3.exe` 等三个执行文件

下载完后 解压出来

File Explorer view of the extracted files:

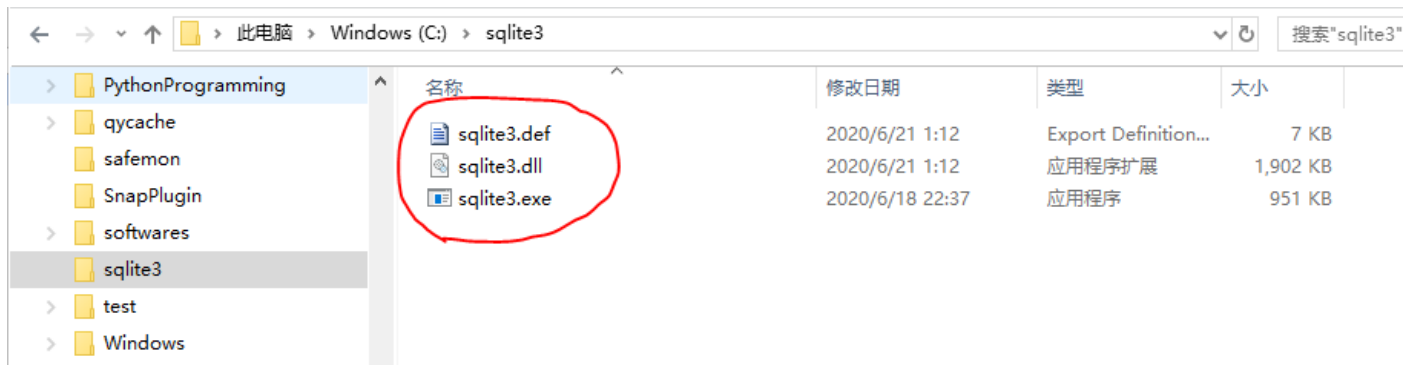
Path: 此电脑 > 桌面 > Python语言与实训 > SQLite数据库安装程序 > **sqlite-dll-win64-x64-3320300**

名称	修改日期	类型	大小
sqlite3.def	2020/6/21 1:12	Export Definition...	7 KB
sqlite3.dll	2020/6/21 1:12	应用程序扩展	1,902 KB

Path: 此电脑 > 桌面 > Python语言与实训 > SQLite数据库安装程序 > **sqlite-tools-win32-x86-3320300**

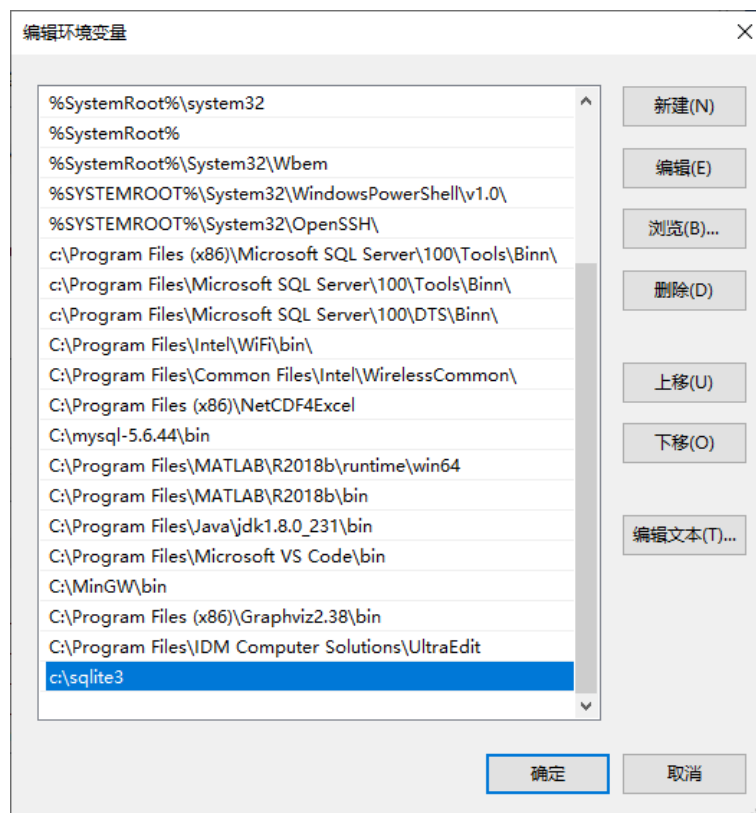
名称	修改日期	类型	大小
sqldiff.exe	2020/6/18 22:36	应用程序	505 KB
sqlite3.exe	2020/6/18 22:37	应用程序	951 KB
sqlite3_analyzer.exe	2020/6/18 22:37	应用程序	1,986 KB

解压后将上面 3 个文件拷贝到同一个目录，此处以 `c:\sqlite` 文件夹为例。



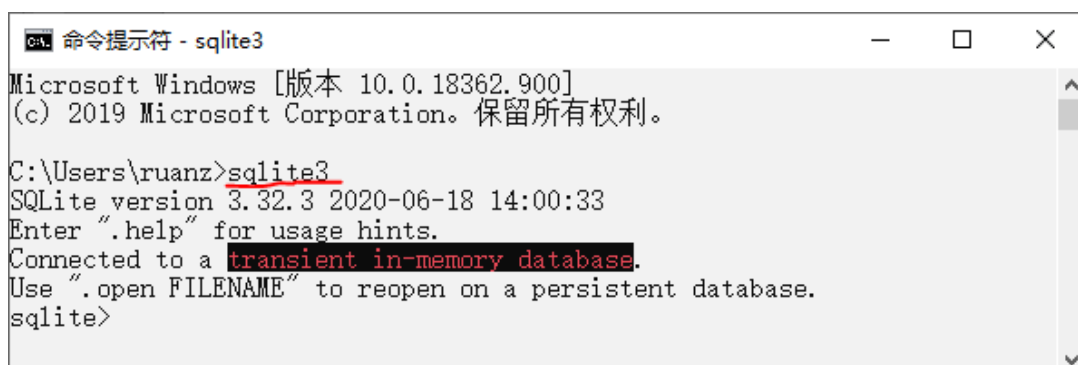
2. 配置 path 环境变量

在系统环境变量 path 中新建 “c:\sqlite3”



3. 测试安装配置

打开 cmd，在命令提示符下，输入 `sqlite3` 命令，显示如下结果，则说明 `sqlite3` 数据库已经能使用了。



二、 SQLite 命令

1. 点命令

一些 SQLite 命令以点 “.” 开头，简单实用，这些命令被称为 SQLite 的**点命令**。

如需获取可用的点命令的清单，可以在任何时候输入 “.help”，会显示各种重要的 SQLite 点命令的列表。

```
命令提示符 - sqlite3
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\ruanz>sqlite3
SQLite version 3.32.3 2020-06-18 14:00:33
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .help
.archive ...           Manage SQL archives
.auth ON|OFF           Show authorizer callbacks
.backup ?DB? FILE      Backup DB (default "main") to FILE
.bail on|off           Stop after hitting an error. Default OFF
.binary on|off         Turn binary output on or off. Default OFF
.cd DIRECTORY          Change the working directory to DIRECTORY
.changes on|off        Show number of rows changed by SQL
.check GLOB            Fail if output since .testcase does not match
.clone NEWDB           Clone data into NEWDB from the existing database
.databases             List names and files of attached databases
.dbconfig ?op? ?val?   List or change sqlite3_db_config() options
.dbinfo ?DB?          Show status information about the database
.dump ?TABLE?         Render database content as SQL
.echo on|off           Turn command echo on or off
.eqp on|off|full|...   Enable or disable automatic EXPLAIN QUERY PLAN
.excel                Display the output of next command in spreadsheet
.exit ?CODE?          Exit this program with return-code CODE
.expert               EXPERIMENTAL. Suggest indexes for queries
```

.help 里的部分点命令

命令	描述
. backup ?DB? FILE	备份 DB 数据库（默认是 “main”）到 FILE 文件。
. bail ON OFF	发生错误后停止。默认为 OFF。
. databases	列出数据库的名称及其所依附的文件。
. dump ?TABLE?	以 SQL 文本格式转储数据库。如果指定了 TABLE 表，则只转储匹配 LIKE 模式的 TABLE 表。
. echo ON OFF	开启或关闭 echo 命令。
. exit 或 .quit	退出 SQLite 提示符。
. explain ON OFF	开启或关闭适合于 EXPLAIN 的输出模式。如果没有带参数，则为 EXPLAIN on，及开启 EXPLAIN。
. header(s) ON OFF	开启或关闭头部显示。
. help	显示消息。
. import FILE TABLE	导入来自 FILE 文件的数据到 TABLE 表中。

<code>.indices ?TABLE?</code>	显示所有索引的名称。如果指定了 TABLE 表，则只显示匹配 LIKE 模式的 TABLE 表的索引。
<code>.load FILE ?ENTRY?</code>	加载一个扩展库。
<code>.log FILE off</code>	开启或关闭日志。FILE 文件可以是 stderr（标准错误）/stdout（标准输出）。
<code>.mode MODE</code>	设置输出模式，MODE 可以是下列之一： <ul style="list-style-type: none"> ■ csv 逗号分隔的值 ■ column 左对齐的列 ■ html HTML 的 <table> 代码 ■ insert TABLE 表的 SQL 插入（insert）语句 ■ line 每行一个值 ■ list 由 .separator 字符串分隔的值 ■ tabs 由 Tab 分隔的值 ■ tcl TCL 列表元素
<code>.nullvalue STRING</code>	在 NULL 值的地方输出 STRING 字符串。
<code>.output FILENAME</code>	发送输出到 FILENAME 文件。
<code>.output stdout</code>	发送输出到屏幕。
<code>.print STRING...</code>	逐字地输出 STRING 字符串。
<code>.prompt MAIN CONTINUE</code>	替换标准提示符。
<code>.read FILENAME</code>	执行 FILENAME 文件中的 SQL。
<code>.schema ?TABLE?</code>	显示 CREATE 语句。如果指定了 TABLE 表，则只显示匹配 LIKE 模式的 TABLE 表。
<code>.separator STRING</code>	改变输出模式和 .import 所使用的分隔符。
<code>.show</code>	显示各种设置的当前值。
<code>.stats ON OFF</code>	开启或关闭统计。
<code>.tables ?PATTERN?</code>	列出匹配 LIKE 模式的表的名称。
<code>.timeout MS</code>	尝试打开锁定的表 MS 毫秒。
<code>.width NUM NUM</code>	为 "column" 模式设置列宽度。
<code>.timer ON OFF</code>	开启或关闭 CPU 定时器。
<code>.open ?OPTIONS? ?FILE?</code>	Close existing database and reopen FILE
<code>.excel</code>	Display the output of next command in spreadsheet

让我们尝试使用 `.show` 命令，来查看 SQLite 命令提示符的默认设置。



```

C:\> 命令提示符 - sqlite3

sqlite> .show
      echo: off
      eqp: off
  explain: auto
   headers: off
      mode: list
nullvalue: ""
    output: stdout
colseparator: "|"
rowseparator: "\n"
      stats: off
      width:
  filename: :memory:
sqlite>
  
```

确保 `sqlite>` 提示符与点命令之间没有空格，否则将无法正常工作。

2. 格式化输出

您可以使用下列的点命令来格式化输出为本教程下面所列出的格式：

```
sqlite>.header on
sqlite>.mode column
sqlite>.timer on
sqlite>
```

上面设置将产生如下格式的输出：

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

CPU Time: user 0.000000 sys 0.000000

3. sqlite_master 表格

主表中保存数据库表的关键信息，并把它命名为 **sqlite_master**。如要查看表概要，可按如下操作：

```
sqlite>.schema sqlite_master
```

这将产生如下结果：

```
CREATE TABLE sqlite_master (  
  
    type text,  
  
    name text,  
  
    tbl_name text,  
  
    rootpage integer,  
  
    sql text  
  
);
```

三、SQLite 语法

SQLite 是遵循一套独特的称为语法的规则和准则。本教程列出了所有基本的 SQLite 语法，向您提供了一个 SQLite 快速入门。

1. 大小写敏感性

有个重要的点值得注意，SQLite 是不区分大小写的，但也有一些命令是大小写敏感的，比如 **GLOB** 和 **glob** 在 SQLite 的语句中有不同的含义。

2. 注释

SQLite 注释是附加的注释，可以在 SQLite 代码中添加注释以增加其可读性，他们可以出现在任何空白处，包括在表达式内和其他 SQL 语句的中间，但它们不能嵌套。

SQL 注释以两个连续的 **"-" 字符**（ASCII 0x2d）开始，并扩展至下一个换行符（ASCII 0x0a）或直到输入结束，以先到者为准。

您也可以使用 **C 风格**的注释，以 **"/"** 开始，并扩展至下一个 **"*/"** 字符对或直到输入结束，以先到者为准。SQLite 的注释可以跨越多行。

```
sqlite>.help -- 这是一个简单的注释
```

3. SQLite 语句

所有的 SQLite 语句可以以任何关键字开始，如 **SELECT**、**INSERT**、**UPDATE**、**DELETE**、**ALTER**、**DROP** 等，**所有语句以分号 ";" 结束**。

四、 SQLite 控制台操作示例

1. 新建或打开一个数据库文件

SQLite 数据库文件以 db 作为文件后缀，创建数据的命令为：

sqlite> .open 数据库文件名

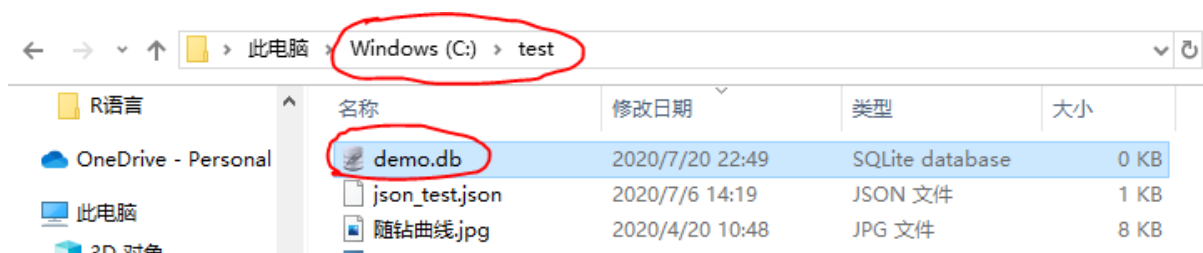
数据库文件名可以包含路径，其中反斜杠 “\” 注意转义表示为 “\\”。数据库文件如果不存在，则新建一个，否则打开已经存在的数据库。

例如，在 c:\test 创建 SQLite 数据库文件 demo.db，则执行如下命令：

sqlite> .open c:\\test\\demo.db

```
命令提示符 - sqlite3
c:\test>sqlite3
SQLite version 3.32.3 2020-06-18 14:00:33
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open c:\\test\\demo.db
sqlite>
```

得到：



名称	修改日期	类型	大小
demo.db	2020/7/20 22:49	SQLite database	0 KB
json_test.json	2020/7/6 14:19	JSON 文件	1 KB
随钻曲线.jpg	2020/4/20 10:48	JPG 文件	8 KB

也可以使用命令查看已附加的数据库文件: **.databases**

sqlite> .databases

```
命令提示符 - sqlite3
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open c:\\test\\demo.db
sqlite> .databases
main: c:\test\demo.db
sqlite>
```

2. 在数据库中新建表

使用 CREATE 语句创建表。


```
sqlite> CREATE TABLE student(id INTEGER PRIMARY KEY, name VARCHAR(20), sex VARCHAR(5),age INT);
```

则在 demo.db 数据库里创个表：

```
命令提示符 - sqlite3
sqlite> .open demo.db
sqlite> CREATE TABLE student(id INTEGER PRIMARY KEY, name VARCHAR(20), sex VARCHAR(5),age INT);
sqlite> .tables
student
sqlite>
```

可以用点命令 **.tables** 查看 demo.db 数据库中的表。

3. 在表中添加一些记录

使用 INSERT 语句添加记录。

一次插入一条记录：

```
sqlite> INSERT INTO student VALUES(1, "张三","男",20);
```

一次插入多条记录：

```
sqlite> INSERT INTO student VALUES(2, "李四","男",45),(3,"吴小英","女",19);
```

```
命令提示符 - sqlite3
sqlite> INSERT INTO student VALUES(1, "张三","男",20);
sqlite> INSERT INTO student VALUES(2, "李四","男",45),(3,"吴小英","女",19);
sqlite>
```

4. 查询记录

使用 SELECT 语句查询记录。

```
sqlite> SELECT * FROM student;
```

```
命令提示符 - sqlite3
sqlite> SELECT * FROM student;
1|张三|男|20
2|李四|男|45
3|吴小英|女|19
sqlite>
```

5. 设置格式化查询结果

点命令 **.mode column** 将设置为列显示模式， **.header** 将显示列名。

```
命令提示符 - sqlite3
sqlite> SELECT * FROM student;
1|张三|男|20
2|李四|男|45
3|吴小英|女|19
sqlite> .header on
sqlite> SELECT * FROM student;
id|name|sex|age
1|张三|男|20
2|李四|男|45
3|吴小英|女|19
```

```
命令提示符 - sqlite3
sqlite> .mode column
sqlite> SELECT * FROM student;
id      name      sex      age
-----
1       张三      男       20
2       李四      男       45
3       吴小英    女       19
sqlite>
```

给

6. 显示表结构

sqlite> .schema [table]

```
命令提示符 - sqlite3
1       张三      男       20
2       李四      男       45
3       吴小英    女       19
sqlite> .schema
CREATE TABLE student(id INTEGER PRIMARY KEY, name VARCHAR(20), sex VARCHAR(5), age I
NT);
sqlite>
```

7. 其他操作参考

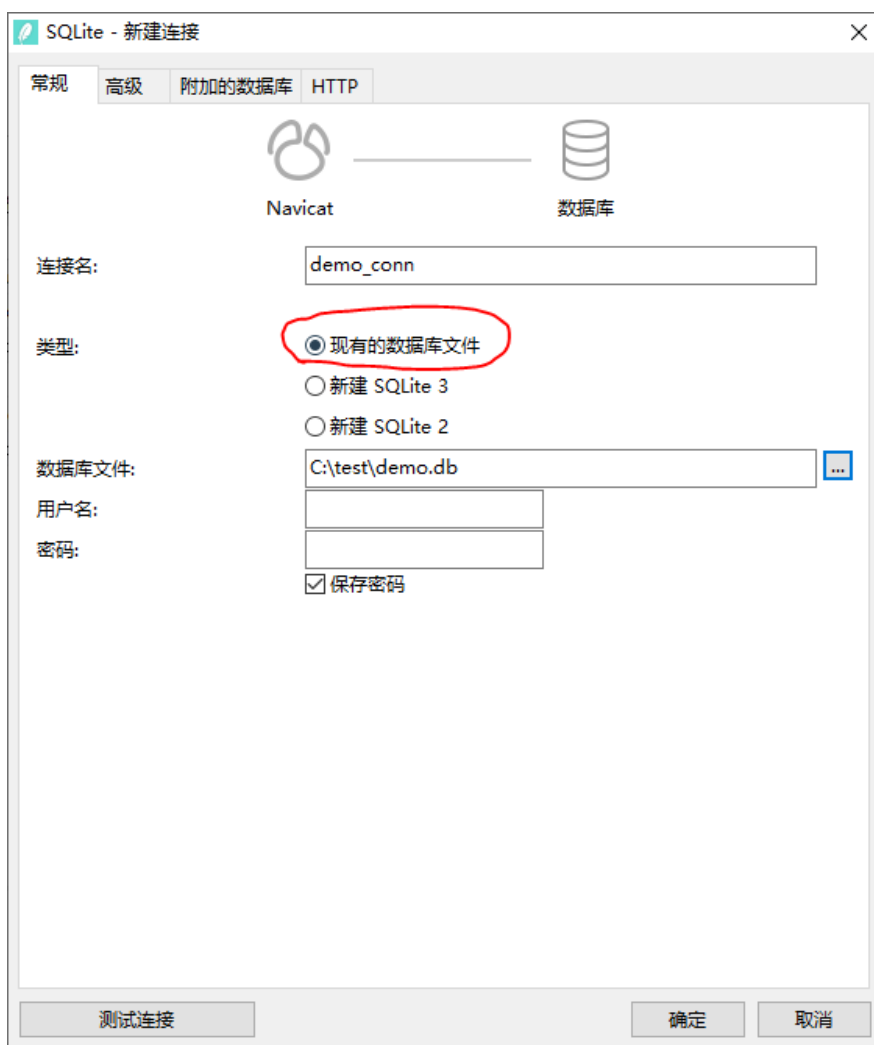
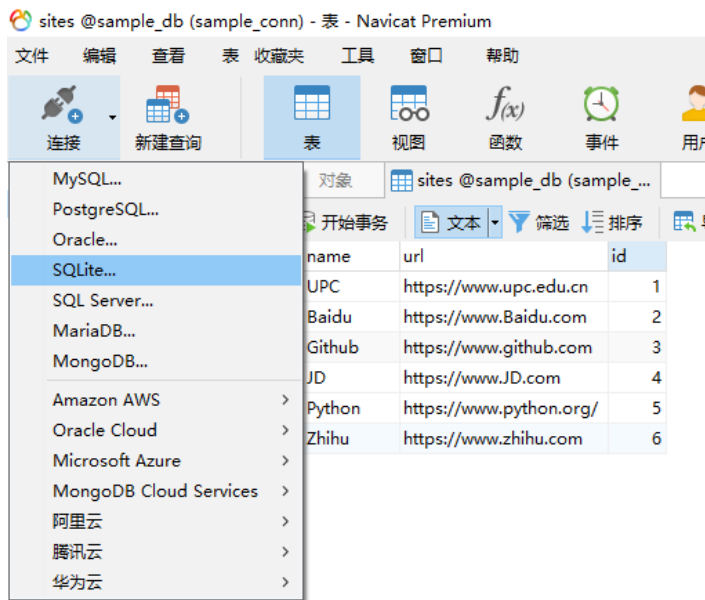
- 修改表结构，增加列：
sqlite> alter table mytable add column email text not null " collate nocase;
- 创建视图：
sqlite> create view nameview as select * from mytable;
- 创建索引：
sqlite> create index test_idx on mytable(value);

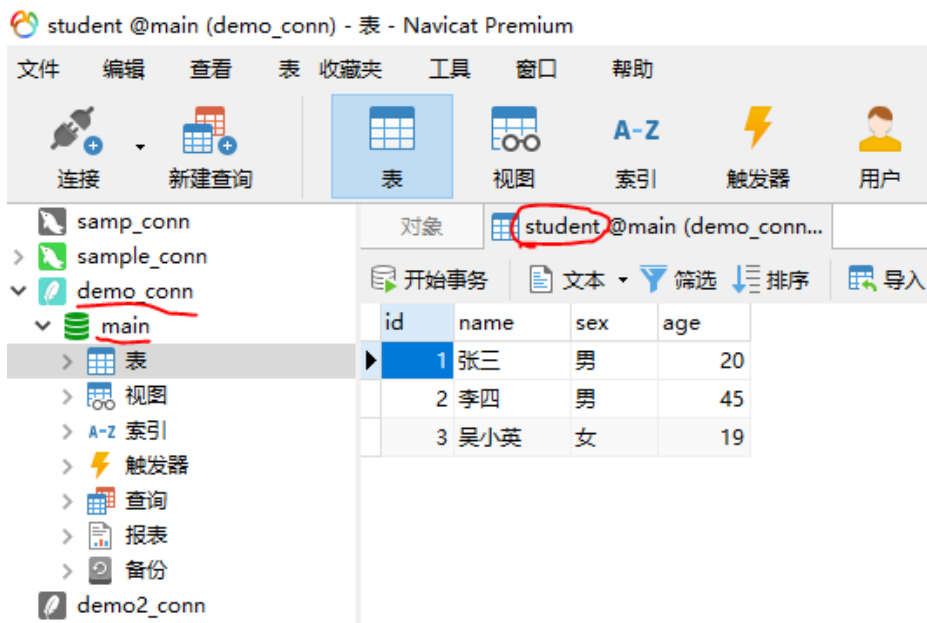
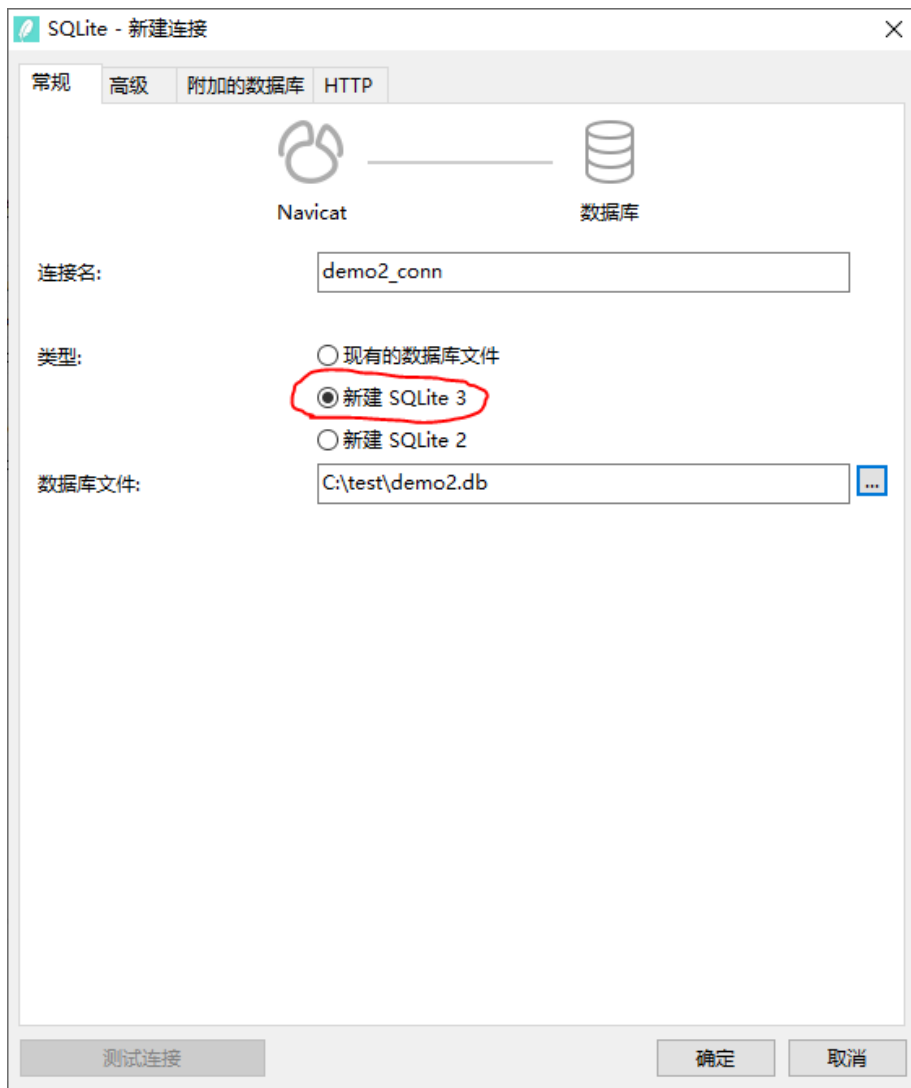
- 显示表结构：
sqlite> .schema [table]
- 获取所有表和视图：
sqlite > .tables
- 获取指定表的索引列表：
sqlite > .indices [table]
- 导出数据库到 SQL 文件：
sqlite > .output [filename]
sqlite > .dump
sqlite > .output stdout
- 从 SQL 文件导入数据库：
sqlite > .read [filename]
- 格式化输出数据到 CSV 格式：
sqlite >.output [filename.csv]
sqlite >.separator ,
sqlite > select * from test;
sqlite >.output stdout
- 从 CSV 文件导入数据到表中：
sqlite >create table newtable (id integer primary key, value text);
sqlite >.import [filename.csv] newtable
- 备份数据库：
/* usage: sqlite3 [database] .dump > [filename] */
sqlite3 mytable.db .dump > backup.sql
- 恢复数据库：
/* usage: sqlite3 [database] < [filename] */
sqlite3 mytable.db < backup.sql

五、 使用 Navicat 或 SQLiteStudio 管理 SQLite 数据库

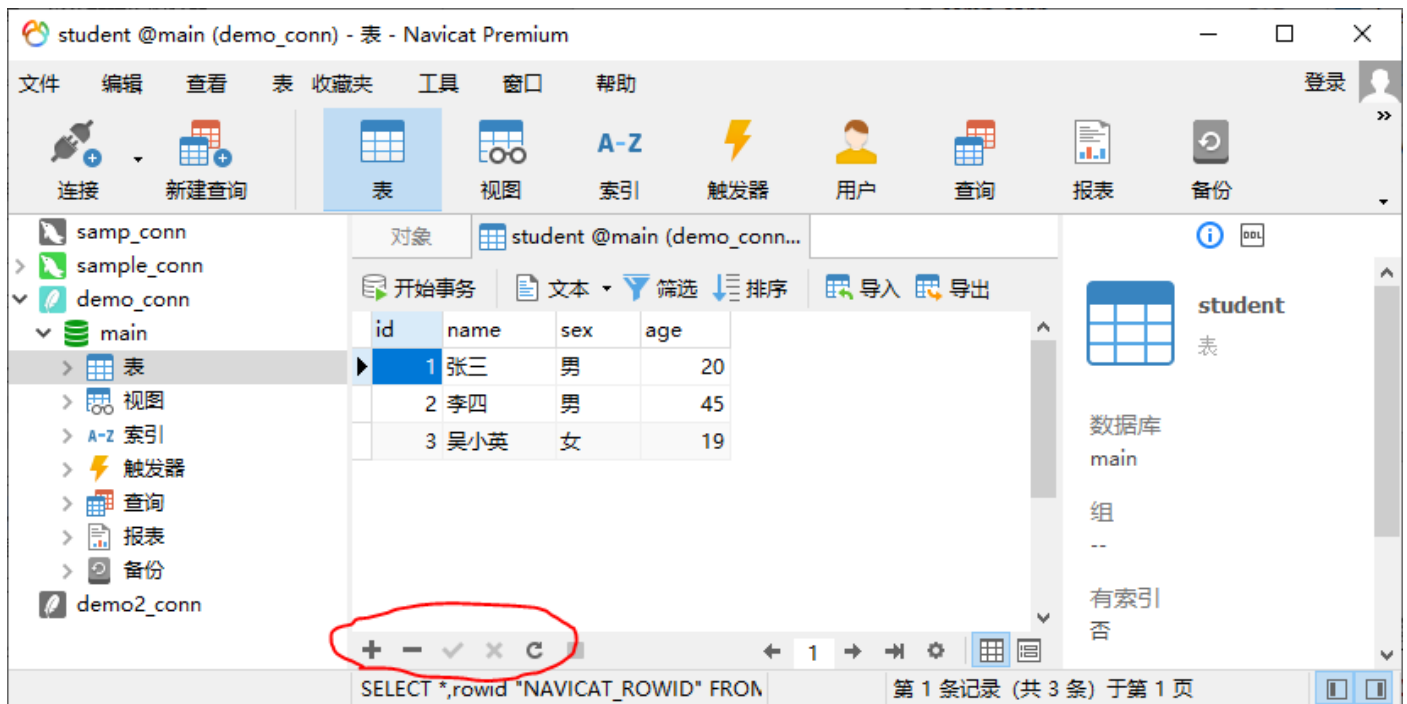
1. 使用 Navicat 管理 SQLite 数据库

- (1) 连接已有数据库或新建数据库





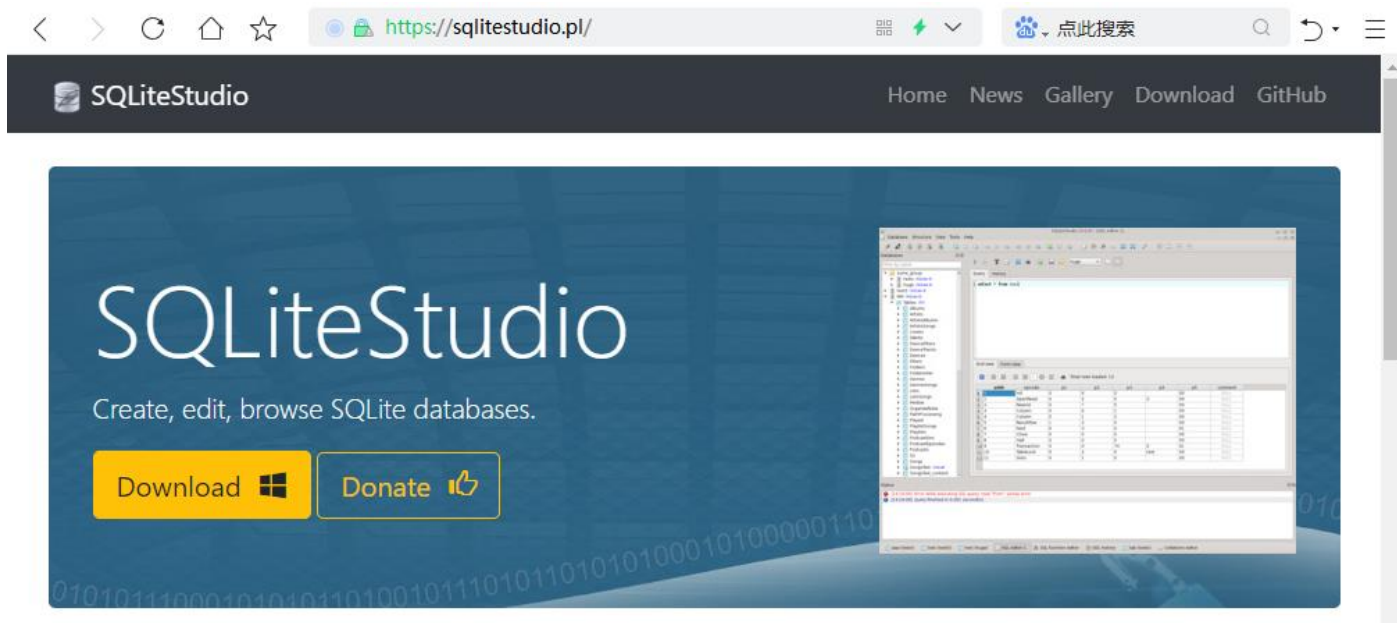
(2) 记录增删改及其刷新操作



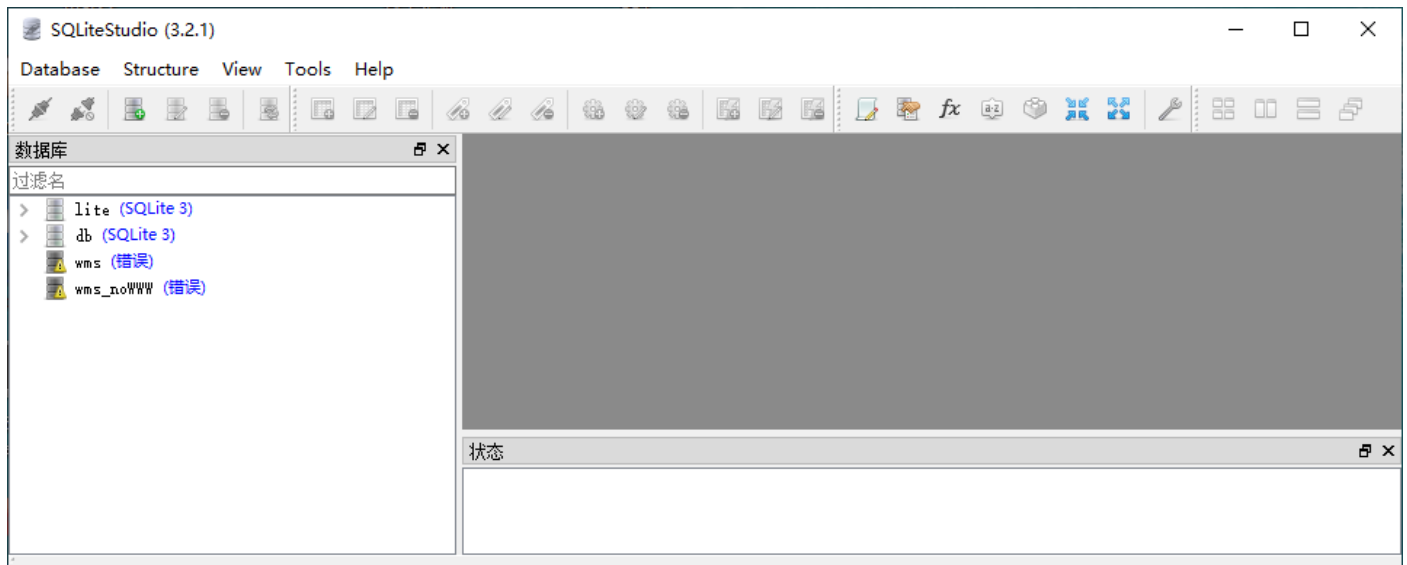
2. 使用 SQLiteStudio 管理 SQLite 数据库

(1) 下载 SQLiteStudio 并安装

下载地址: <https://sqlitestudio.pl/>



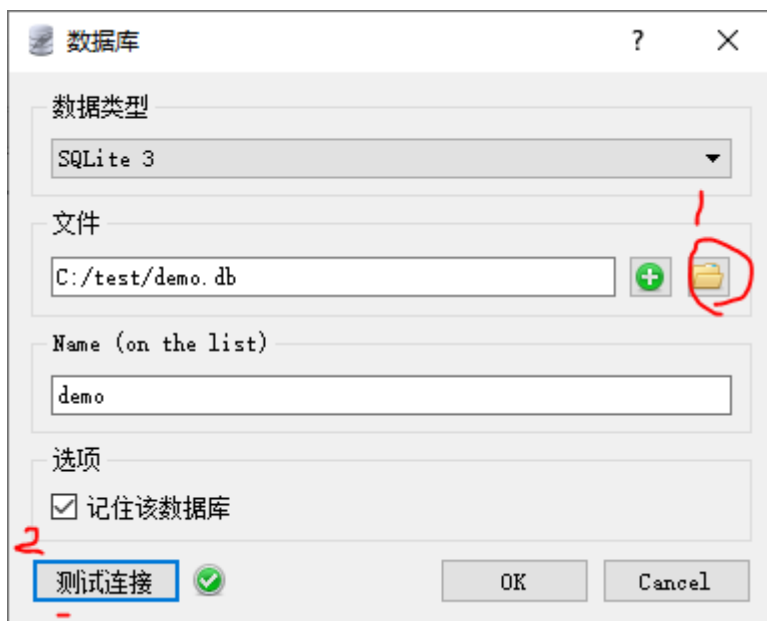
将下载的 SQLiteStudio-3.2.1.zip 解压缩后, 直接执行 SQLiteStudio.exe 就可以使用 SQLiteStudio 了, 免安装。

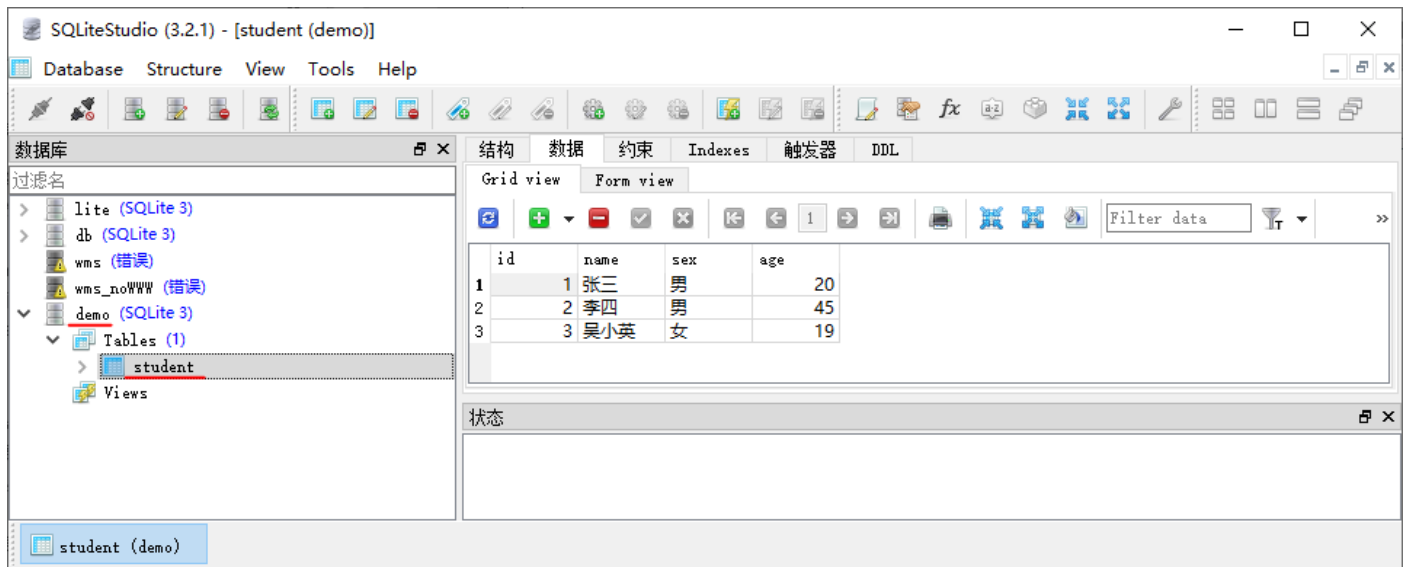


(2) 建立数据库连接

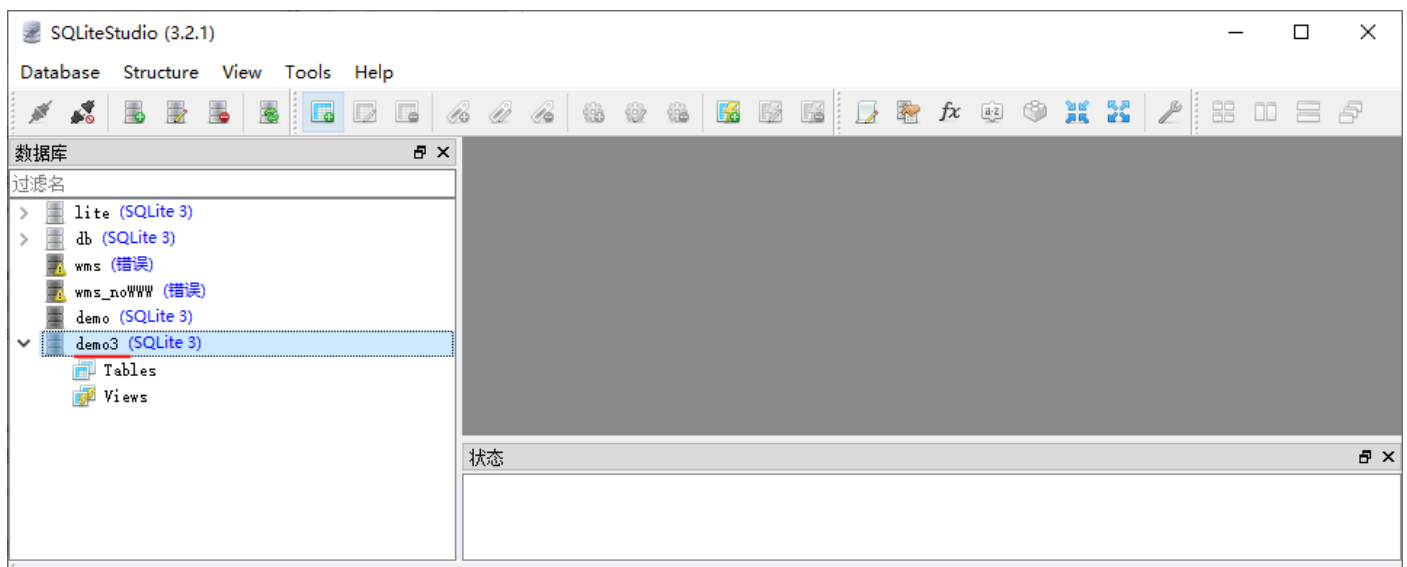
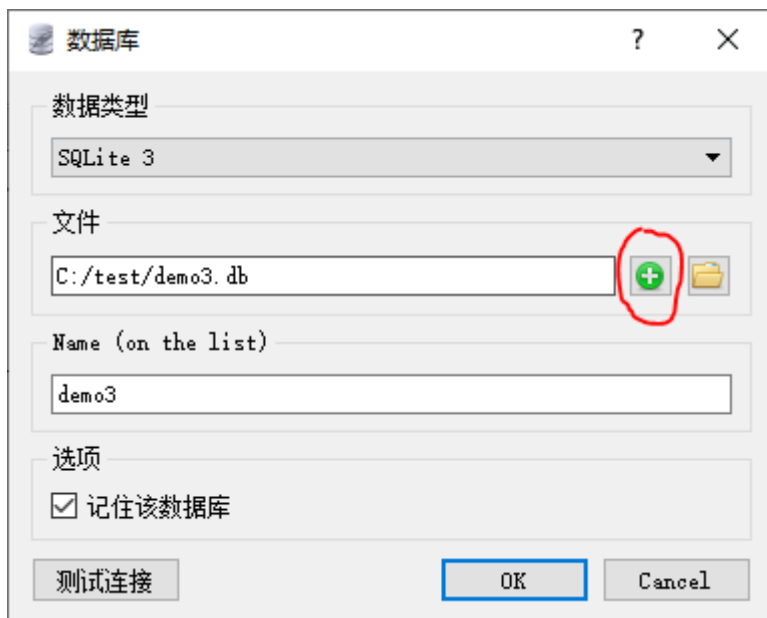


a) 连接至已有数据库

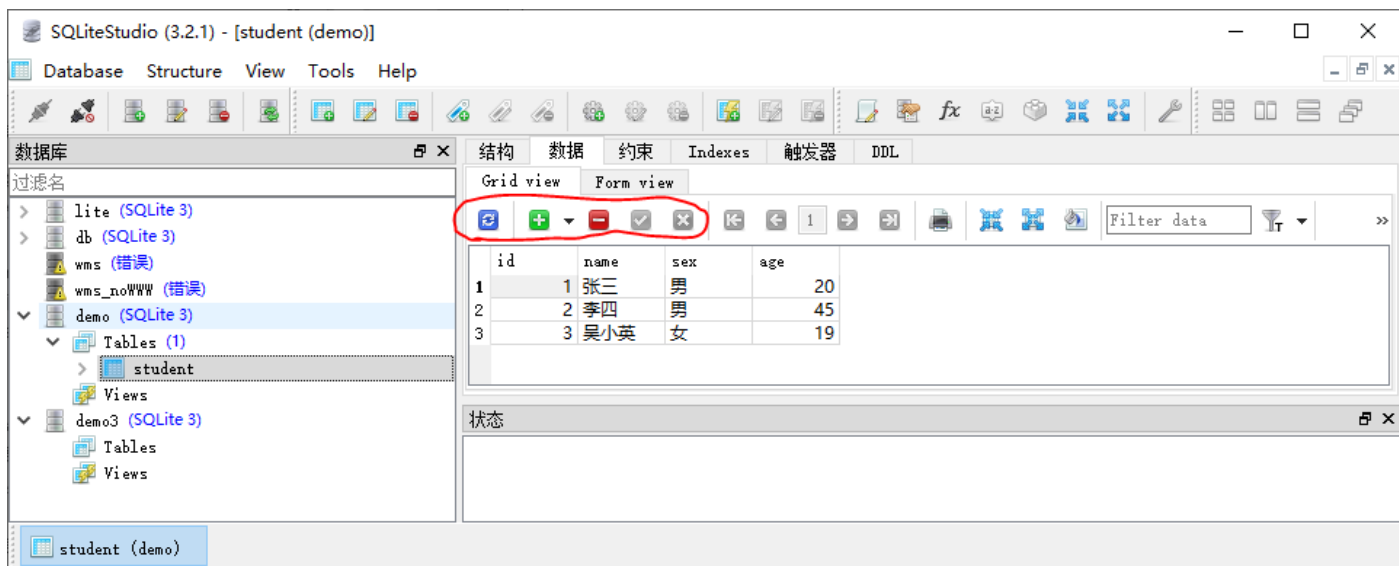




b) 新建数据库并建立连接



(3) 记录增、删、改、提交、回滚及刷新等操作



六、 Python 中访问 SQLite 数据库

SQLite 3 是 Python 3 预装的、相当完备、无需配置的基于 SQL 的数据库管理系统。要使用 SQLite，只需导入 sqlite3 库，并使用 Python 标准化数据库 API 来编程，而不用处理其他工作，比如：安装数据库、配置等等。

Python 数据库 API 提供了一种标准机制，可以针各种各样的数据库管理系统，包括 SQLite。不管使用什么后台数据库，代码所遵循的过程都是一样的：**连接 -> 创建游标 -> 交互（利用游标，使用 SQL 管理数据） -> 提交/回滚 -> 关闭**。

1. SQLite 数据库访问示例一：读取数据库服务器的当前日期

```
import sqlite3  # 导入 sqlite3 库

conn = sqlite3.connect(r'c:\test\demo.db')  # 1、建立与数据库的连接
cursor = conn.cursor()  # 2、创建数据游标
cursor.execute("select date('NOW')")  # 3、执行一些 SQL 操作
print(cursor.fetchone())

conn.commit()  # 4、提交所做的修改，使修改永久保留
conn.close()  # 5、完成时关闭链接
```

某次执行程序，输出结果为： ('2020-07-21',)

2. SQLite 数据库访问示例二：创建表、添加记录和查询记录

```
import sqlite3 # 导入 sqlite3 库

db_path = r'c:\test\demo.db' # 数据库名
sql1 = "drop table if exists books;"
sql2 = ""
create table books(
    id integer primary key autoincrement unique not null,
    name text not null,
    price integer,
    publish_date date not null
);
""""

conn = sqlite3.connect(db_path) # 连接数据库
cursor = conn.cursor() # 获取游标

# 先删除表（如果存在），然后创建表
cursor.execute(sql1) # 执行 SQL
cursor.execute(sql2)

# 插入数据
sql3 = "insert into books (name,price,publish_date) values (?,?,?)" # ? 为占位符
cursor.execute(sql3, ('Java', 123.23, '2012-12-03')) # 执行 SQL
cursor.execute(sql3, ('C++', 83.23, '2013-02-03'))
conn.commit() # 提交修改

# 查询记录
sql4 = "SELECT * FROM books"
cursor.execute(sql4)
# 返回一个元素为 tuple（元组）的 list，表示记录集
records = cursor.fetchall()
for row in records:
    print(row)

conn.close() # 关闭数据库连接
```

执行后，输出结果：

(1, 'Java', 123.23, '2012-12-03')

(2, 'C++', 83.23, '2013-02-03')

在 SQLiteStudio 中可以看到，demo.db 数据库中新增了表 books，其中有两条记录：

