

SQLite官网: <https://www.sqlite.org/index.html>

SQLite可视化工具: <https://sqliteexpert.com/download.html>

1.LINUX下安装SQLite

```
1 apt-get install sqlite3
2 apt-get install libsqlite3-dev
```

2.Windows下安装SQLite

2.1.下载SQLite二进制文件

前往SQLite官方网站的下载页面: SQLite Download Page

The screenshot shows the SQLite Download Page. The page has a navigation bar with links: Home, About, Documentation, Download, License, Support, Purchase. The main content area is titled "SQLite Download Page". It lists several download options under the heading "Pre-release Snapshots". The options are:

- sqlite-snapshot-202310101804.tar.gz (3.04 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.
- Source Code**:
 - sqlite-amalgamation-3430200.zip (2.55 MiB)**: C source code as an amalgamation, version 3.43.2.
 - sqlite-autoconf-3430200.tar.gz (3.03 MiB)**: C source code as an amalgamation. Also includes a "configure" script and TEA makefiles for the TCL Interface.
- Documentation**:
 - sqlite-doc-3430200.zip (10.19 MiB)**: Documentation as a bundle of static HTML files.
- Precompiled Binaries for Android**:
 - sqlite-android-3430200.aar (3.34 MiB)**: A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to drop into any Android Studio project.
- Precompiled Binaries for Linux**:
 - sqlite-tools-linux-x86-3430200.zip (2.21 MiB)**: A bundle of command-line tools for managing SQLite database files, including the `command-line shell` program, the `sqlite3_analyzer` program, and the `sqlite3_diff` program.
- Precompiled Binaries for Mac OS X (x86)**:
 - sqlite-tools-osx-x86-3430200.zip (1.58 MiB)**: A bundle of command-line tools for managing SQLite database files, including the `command-line shell` program, the `sqlite3_analyzer` program, and the `sqlite3_diff` program.
- Precompiled Binaries for Windows**:
 - sqlite-dll-win32-x86-3430200.zip (580.10 KiB)**: 32-bit DLL (x86) for SQLite version 3.43.2.
 - sqlite-dll-win64-x64-3430200.zip (1.18 MiB)**: 64-bit DLL (x64) for SQLite version 3.43.2.
 - sqlite-tools-win32-x86-3430200.zip (1.95 MiB)**: A bundle of command-line tools for managing SQLite database files, including the `command-line shell` program, the `sqlite3_analyzer.exe` program, the `sqlite3_diff.exe` program, and the `sqlite3_analyzer` program. 32-bit.




在“Precompiled Binaries for Windows”部分中, 会看到多个版本的SQLite二进制文件, 选择适合的系统架构的版本。选择下载红框中的压缩文件(它是一组用于管理SQLite数据库文件的命令行工具)。

2.2.解压缩SQLite二进制文件


解压缩下载的文件, 并将其中的文件复制到希望使用SQLite的目录或系统路径中。(例如, 可以将其解压缩到D:\sqlite目录中)

APP (D:) > sqlite

在 sqlite 中搜索


名称	修改日期 ^	类型	大小
 sqldiff.exe	2023/10/10 23:09	应用程序	589 KB
 sqlite3_analyzer.exe	2023/10/10 23:10	应用程序	2,069 KB
 sqlite3.exe	2023/10/10 23:10	应用程序	1,162 KB

- **sqlite3.exe**:

 **作用:** **sqlite3.exe** 是 **SQLite** 的交互式命令行工具, 它允许用户与 **SQLite** 数据库进行交互、执行 SQL 查询、管理数据库和表, 以及进行各种数据库操作。


详细介绍: **sqlite3.exe** 是一个命令行界面的工具, 它允许用户打开 **SQLite** 数据库文件, 并在命令行中执行 SQL 语句。可以使用它来创建数据库、创建表、插入数据、查询数据、更新数据、删除数据等等。它对于开发、测试和调试 **SQLite** 数据库非常有用, 也可以用于执行数据库备份和还原操作。可以通过在命令行中运行 **sqlite3.exe** 并指定数据库文件来启动它, 然后在交互式命令行中输入 SQL 命令。

- **sqldiff.exe**:

 **作用:** **sqldiff.exe** 是 **SQLite** 的数据库比较工具, 用于比较两个 **SQLite** 数据库文件之间的结构 and 数据差异。

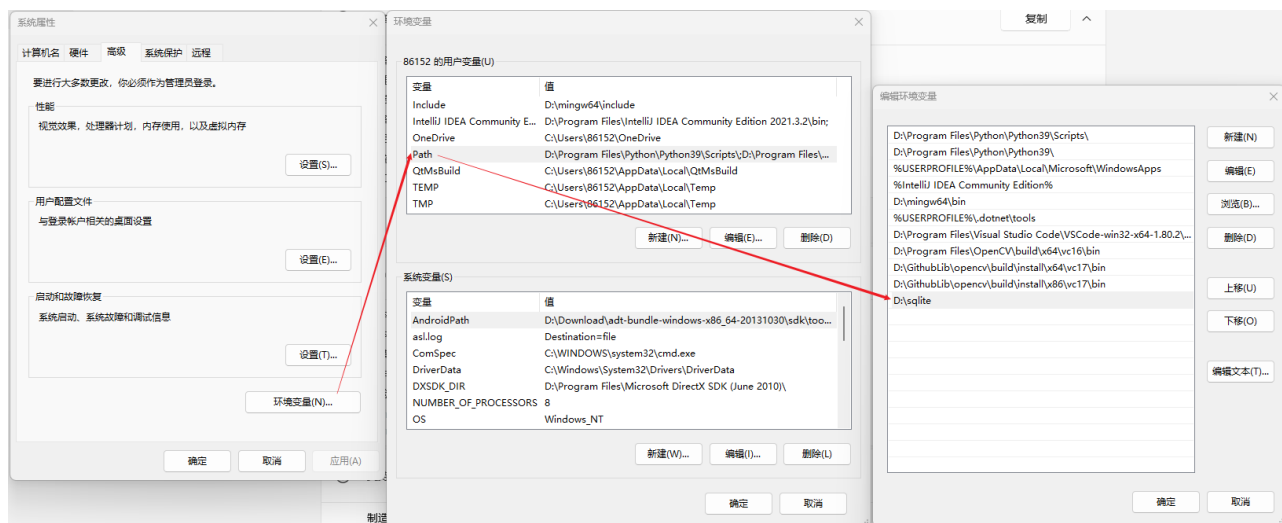
详细介绍: **sqldiff.exe** 用于比较两个 **SQLite** 数据库文件, 查找它们之间的差异, 包括数据库结构 (例如, 表、列、索引等) 和数据。这对于在不同环境中同步数据库、检测更改或合并数据库非常有用。**sqldiff.exe** 生成一个 SQL 脚本, 该脚本包含将一个数据库变更为另一个数据库的命令。可以使用该脚本来将一个数据库的更改应用到另一个数据库。

- **sqlite3_analyzer.exe**:

 **作用:** **sqlite3_analyzer.exe** 是 **SQLite** 的数据库性能分析工具, 用于分析 **SQLite** 数据库文件的性能和结构。

详细介绍: **sqlite3_analyzer.exe** 可用于检查 **SQLite** 数据库文件的结构和性能特征。它可以生成详细的报告, 包括数据库的 B 树索引结构、页面使用情况、表的大小等等。这些信息对于优化数据库查询和性能调整非常有用。另外还可以用于评估数据库文件的大小和性能, 以便在需要时采取相应的措施来改进数据库的性能。

如果要在任何位置使用**SQLite**命令行工具, 需要将**SQLite**添加到系统环境变量中。



2.3.验证安装

打开命令提示符，然后执行 `sqlite3 --version` 命令来验证SQLite是否正确安装，如果安装成功会看到SQLite的版本信息。

```
C:\WINDOWS\system32\cmd. X + -
Microsoft Windows [版本 10.0.22621.2428]
(c) Microsoft Corporation。保留所有权利。

C:\Users\86152>sqlite3 --version
3.43.2 2023-10-10 12:14:04 4310099cce5a487035fa535dd3002c59ac7f1d1bec68d7cf317fd3e769484790 (32-bit)

C:\Users\86152>_
```

2.4.SQLite可视化工具安装

SQLite Expert 是一个用于管理和操作 SQLite 数据库的图形化用户界面 (GUI) 工具，可以从 [SQLite Expert 官网](#) 下载最新的免费安装文件。

2.5.SQLite库文件获取

2.5.1.下载SQLite源文件和库文件

首先，需要下载SQLite的预编译二进制文件（dll文件）以及源码文件（SQLite Amalgamation文件，这是SQLite的C语言源代码文件），可以从SQLite官方网站 下载最新的版本。

SQLite Download Page

Pre-release Snapshots

[sqlite-snapshot-202310101804.tar.gz](#) (3.04 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.
(SHA3-256: b5fce126b9e81aa418d616d1b25c6b271ecbf22356455bb34c1bf33ccc56f74)

Source Code

[sqlite-amalgamation-3430200.zip](#) (2.55 MiB) C source code as an amalgamation, version 3.43.2.
(SHA3-256: af02b88c922e7306c6b9737369d0756d9e24e47741d4b315af341edd8b40)
[sqlite-autoconf-3430200.tar.gz](#) (3.03 MiB) C source code as an amalgamation. Also includes a "configure" script and TEA makefiles for the [TCL Interface](#).
(SHA3-256: a7463a45ed58849200858e514b79f7e3f5de9830047897c36e39a78a0bc75cc1)

Documentation

[sqlite-doc-3430200.zip](#) (10.19 MiB) Documentation as a bundle of static HTML files.
(SHA3-256: 7857d32739b34743c19544124eb9adc5b9e1625d8a85cd66f0104cdc12ade40)

Precompiled Binaries for Android

[sqlite-android-3430200.aar](#) (3.34 MiB) A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to drop into any Android Studio project.
(SHA3-256: 42b4a350290e21db0e7bd24a3c1c32e69a4f695192fe31b7c00186868233883)

Precompiled Binaries for Linux

[sqlite-tools-linux-x86-3430200.zip](#) (2.21 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqlite3_analyzer](#) program, and the [sqlite3_diff](#) program.
(SHA3-256: eef93c3deb1e22b4bd9ea5a01aea58186fde9e5bde494e457021b4db84f93)

Precompiled Binaries for Mac OS X (x86)

[sqlite-tools-osx-x86-3430200.zip](#) (1.58 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqlite3_analyzer](#) program, and the [sqlite3_diff](#) program.
(SHA3-256: eb838d85e162eb7be47e61bad03d783d6f043be3b8819a58aac4512ea3f9c3b)

Precompiled Binaries for Windows

[sqlite-dll-win32-x86-3430200.zip](#) (580.10 KiB) 32-bit DLL (x86) for SQLite version 3.43.2.
(SHA3-256: c103bea819ebc59310b3d711097ebc9842ca5dff5f1fc8ff9927a36dbfb8)
[sqlite-dll-win64-x64-3430200.zip](#) (1.18 MiB) 64-bit DLL (x64) for SQLite version 3.43.2.
(SHA3-256: 860c7a7e6b5c412b7e98b3cd5259b45f7cb77fde9f6dbab49f4d663ddcb)
[sqlite-tools-win32-x86-3430200.zip](#) (1.95 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqlite3_analyzer.exe](#) program, and the [sqlite3_diff.exe](#) program. 32-bit.
(SHA3-256: e10365c4d7e04a34e17ac5b0ddc4b0983f8f8e8b5493b477c0b51042fa9993f1)

(2) [sqlite-amalgamation-3430200.zip](#) 文件解压后文件内容如下，是不是非常轻便简洁。

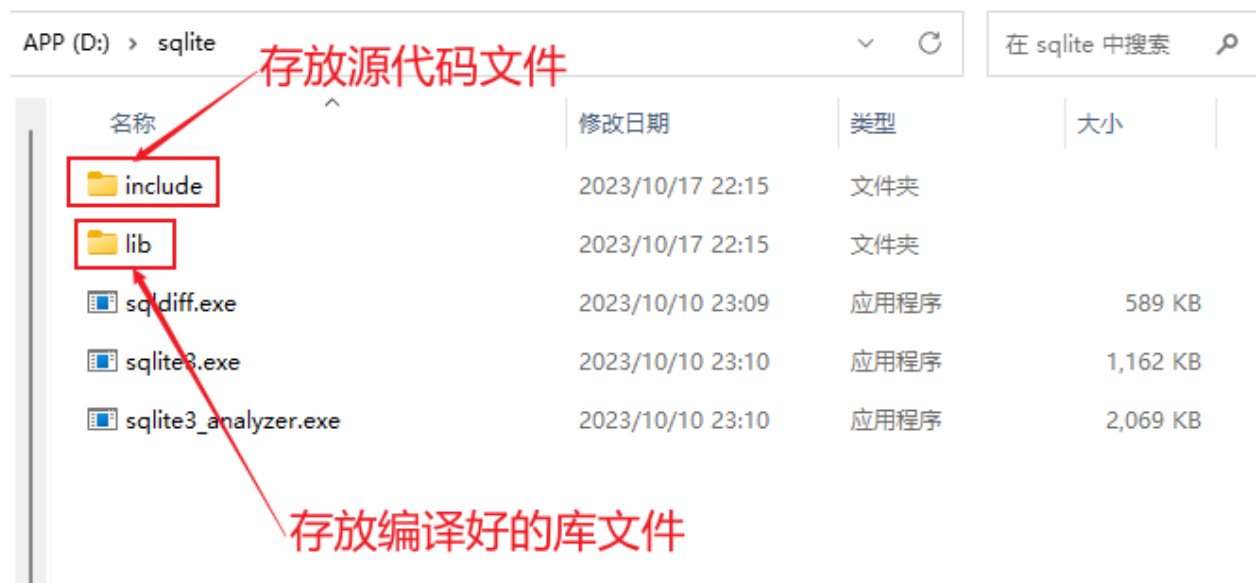
APP (D:) > sqlite > include				在 include 中搜索
名称	修改日期	类型	大小	
shell.c	2023/10/10 23:08	C 文件	863 KB	
sqlite3.c	2023/10/10 23:08	C 文件	8,641 KB	
sqlite3.h	2023/10/10 23:08	C++ Header file	614 KB	
sqlite3ext.h	2023/10/10 23:08	C++ Header file	37 KB	

(3) [sqlite-dll-win64-x64-3430200.zip](#) 解压缩后包含SQLite数据库引擎的核心库，通常以DLL（动态链接库）的形式提供。这个库文件是用来在应用程序中访问和操作SQLite数据库的。在平时开发中，需要将这个库文件与应用程序一起使用，以便应用程序能够连接到SQLite数据库。

名称	修改日期	类型	大小
sqlite3.def	2023/10/10 20:52	VCEXPRESS.DEF.1...	8 KB
sqlite3.dll	2023/10/10 20:52	应用程序扩展	2,905 KB

[sqlite3.dll](#) 是SQLite数据库引擎的二进制库，允许动态加载SQLite数据库引擎，而[sqlite3.def](#) 文件定义了 [sqlite3.dll](#) 中可供外部程序访问的函数和符号列表，以便其他程序可以在编译时正确链接到SQLite库。

(4) 把这些解压缩后的文件整理一下，以下是我的存放方式，统一放在 D:\sqlite 路径下。



D:\sqlite 目录下文件存放方式如下：

- D:\sqlite\include 目录存放 `sqlite-amalgamation-3430200.zip` 解压后的文件。
- D:\sqlite\lib\x64 目录存放 `sqlite-dll-win64-x64-3430200.zip` 解压后的文件。
- D:\sqlite\lib\x86 目录存放 `sqlite-dll-win32-x86-3430200.zip` 解压后的文件。

2.5.2.编译SQLite3源码得到sqlite3.lib静态库

(1) 找到 Visual Studio 安装目录下的 lib.exe 工具路径，将该路径添加到系统环境变量中：（系统架构为64位选Hostx64目录，32位选Hostx86目录）

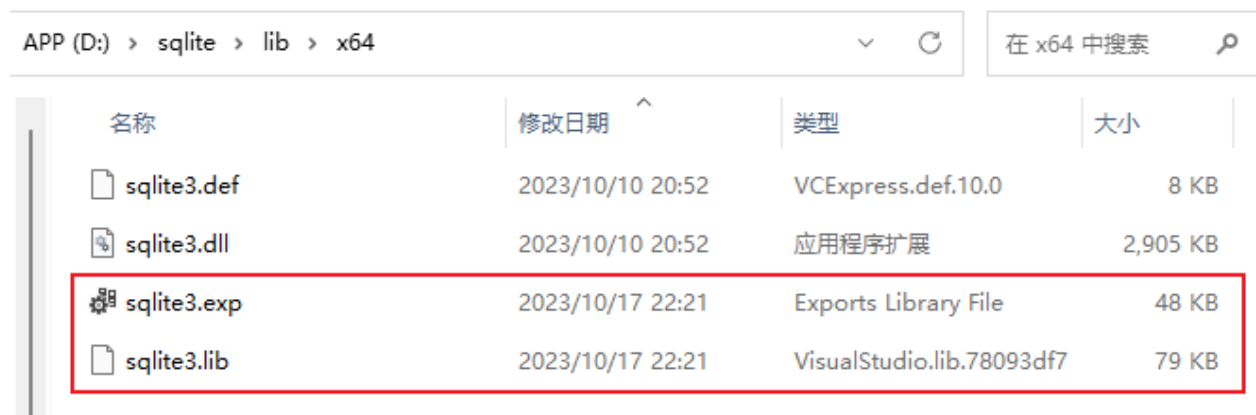
1. D:\Program Files\Visual Studio 2022\VC\Tools\MSVC\14.35.32215\bin\Hostx64\x64

2. 上述路径表示 Visual Studio 2022 中的编译工具链是为在 64 位 Windows 操作系统上构建 64 位应用程序和库而设计的。这使我们能够使用 Visual Studio 来开发和构建适用于 64 位环境的软件。（其它路径同理）

(2) 在上面的 D:\sqlite\x64 路径下打开终端，执行命令：LIB /MACHINE:X64 /DEF:sqlite3.def

1. /MACHINE:X64：指定生成的库文件将用于 64 位体系结构，即 x64 架构。

2. /DEF:sqlite3.def：指定用于生成库文件的定义文件（.def文件）。.def 文件通常包含了库文件的导出函数和符号信息。（通过def和dll文件生成lib文件）



(3) 然后会生成sqlite3.exp和sqlite3.lib两个文件。

(4) 切换到 D:\sqlite\lib\x86 路径下并打开终端执行命令：LIB /MACHINE:X86 /DEF:sqlite3.def，将会生成用于32位的库文件。

2.5.3.lib.exe工具的作用

(1) lib.exe 是 Microsoft Visual Studio 中的库工具，它的主要用途是用于创建、管理和处理静态链接库 (Static Link Library, .lib 文件)。

(2) lib.exe 具有以下主要功能和用途：

1. 创建静态链接库：lib.exe 可用于将多个目标文件（通常是 .obj 文件）合并成一个静态链接库文件，以便在编译和链接应用程序时使用。
2. 添加和删除目标文件：可以使用 lib.exe 向现有的库文件中添加新的目标文件，也可以从库文件中删除不再需要的目标文件。
3. 生成导入库：lib.exe 通常用于生成库文件的导入版本 (import library)，以便在应用程序编译时链接到动态链接库 (.dll 文件)。
4. 列出库文件内容：可以使用 lib.exe 来列出库文件中包含的目标文件和符号信息。
5. 更新库文件：如果库中的一个或多个目标文件已被更新，可以使用 lib.exe 来重新构建库文件，以确保它包含最新的目标文件。

2.5.4.添加SQLite库和头文件

1. 将下载的SQLite头文件复制到项目目录中。
2. 在Visual Studio中，右键单击项目，选择“属性”打开属性窗口。
3. 在属性窗口中，展开“VC++ 目录 > 常规”，然后将SQLite头文件的目录路径添加到“外部包含目录”中。
(D:\sqlite\include)
4. 在属性窗口中，展开“VC++ 目录 > 常规”，然后将SQLite的dll文件所在的目录路径添加到“库目录”中。
(D:\sqlite\lib\x64)
 - 如果库目录是D:\sqlite\lib\x64就在当前工程目录下添加 D:\sqlite\lib\x64 目录下的sqlite3.dll文件。
 - 如果库目录是D:\sqlite\lib\x86就在当前工程目录下添加 D:\sqlite\lib\x86 目录下的sqlite3.dll文件。
5. 在属性窗口中，展开“链接器 > 输入”，然后将SQLite的库文件（通常是sqlite3.dll）添加到“附加依赖项”中。

3.SQLite交叉编译

1. 从sqlite官方网上下载sqlite原码，网址<http://www.sqlite.org/download.html>
2. tar -zxvf sqlite-autoconf-3071600.tar.gz -C ./ 解压到当前目录下
3. 进入到原码目录下
4. ./configure --host=/usr/local/hfctool/bin/arm-linux-gnueabi --prefix=/usr/local/
 - --host=交叉编译器目录
 - --prefix=需要安装的目录
5. 打开Makefile文件，找到-DPACKAGE_STRING="sqlite 3.7.16"位置，去掉"3.7.16"前面的空格（擦，编译了半天都没有编译过去，后来网上找到答案！）
6. make && make install
7. 编译安装完成后，会在安装目录下生成四个文件夹：bin include lib share

- 8. 将bin下的文件下载到开发板的/bin/目录下，将lib目录下的所有文件下载到开发板/lib/目录下
- 9. include中的头文件交叉编译文件时会用到，将include目录下的文件复制到交叉编译器目录下的include/目录下
- 10. 在开发板上终端下输入sqlite3命令

4.打开数据库

原型：

```
1 int sqlite3_open(const char *filename, sqlite3 **ppDb)
```

该接口函数用来打开一个指向 SQLite 数据库文件的连接，返回一个用于其他 SQLite 程序的数据库连接对象。如果没有名为 filename 的数据库文件，则创建一个名为 filename 的数据库文件。如果已经存在名为filename的数据库文件，则打开该数据库文件。

返回值：(列举几个相对重要的返回值)

宏	值	含义
SQLITE_OK	0	Successful result
SQLITE_ERROR	1	SQL error or missing database
SQLITE_INTERNAL	2	Internal logic error in SQLite
SQLITE_PERM	3	Access permission denied
SQLITE_ABORT	4	Callback routine requested an abort
SQLITE_BUSY	5	The database file is locked
SQLITE_LOCKED	6	A table in the database is locked
SQLITE_NOMEM	7	A malloc() failed
SQLITE_READONLY	8	Attempt to write a readonly database
SQLITE_INTERRUPT	9	Operation terminated by sqlite3_interrupt()
SQLITE_NOTADB	26	File opened that is not a database file
SQLITE_ROW	100	sqlite3_step() has another row ready
SQLITE_DONE	101	sqlite3_step() has finished executing

5.执行SQL命令

原型：

```
1 int sqlite3_exec(sqlite3*, const char *sql, sqlite_callback, void *data, char **errmsg)
```

该接口函数提供了一个执行 SQL 命令的快捷方式，SQL 命令由 sql 参数提供，可以由多个 SQL 命令组成。参数说明：

- 参数 `sqlite3` 是打开的数据库对象名称；
- 参数 `sql` 作为 SQL 命令传入；
- 参数 `sqlite_callback` 是一个回调；
- 参数 `data` 作为回调函数 `sqlite_callback` 的第一个参数；
- 参数 `errmsg` 用来保存 SQL 命令是错误信息。

6.关闭数据库

原型：

```
1 int sqlite3_close(sqlite3*)
```

该接口函数用来关闭之前调用 `sqlite3_open()` 打开的数据库连接。所有与连接相关的语句都应在连接关闭之前完成。如果还有查询没有完成，`sqlite3_close()` 将返回 `SQLITE_BUSY` 禁止关闭的错误消息。

7.操作SQLite数据库

```
1 #include <stdio.h>
2 #include <sqlite3.h>
3 #include <stdlib.h>
4
5 int callback(void *para, int f_num, char **f_value, char **f_name)
6 {
7     int i;
8     for (i = 0; i < f_num; ++i)
9     {
```



```

10     printf("%s:%s\t", f_name[i], f_value[i]);
11 }
12 printf("\n");
13 return 0;
14 }
15
16 int main()
17 {
18     sqlite3 *db;
19     char *errmsg;
20     int ret;
21
22     // 打开数据库
23     ret = sqlite3_open("config.db", &db);
24     if (ret != SQLITE_OK)
25     {
26         perror("open database failed!!!\n");
27         return -1;
28     }
29
30     // 创建数据表
31     ret = sqlite3_exec(db, "CREATE TABLE if not exists config (id INTEGER AUTO
INCREMENT,\
32                                     applicationName
33                                     varchar(50),\
34                                     applicationService
35                                     varchar(50),\
36                                     applicationType
37                                     varchar(50));",
38                                     NULL, NULL, &errmsg);
39     if (ret != SQLITE_OK)
40     {
41         perror("create table config failed!!!\n");
42         sqlite3_free(errmsg);
43         return -2;
44     }
45
46     // 插入数据
47     ret = sqlite3_exec(db, "INSERT into config
(applicationName,applicationService,applicationType)
VALUES('mxapp2','myir.cn.mxapp','qt');", NULL, NULL, &errmsg);

```

```
45     if (ret != SQLITE_OK)
46     {
47         perror("insert data failed!!!\n");
48         sqlite3_free(errormsg);
49         return -3;
50     }
51
52     // 查询数据
53     printf("after insert data:\n");
54     sqlite3_exec(db, "SELECT * FROM config;", callback, NULL, &errormsg);
55     printf("\n");
56
57     // 删除数据
58     ret = sqlite3_exec(db, "delete from config where applicationName='mxapp2';", NULL,
59     NULL, &errormsg);
60     if (ret != SQLITE_OK)
61     {
62         perror("delete data failed!!!\n");
63         sqlite3_free(errormsg);
64         return -4;
65     }
66
67     // 查询数据
68     printf("after delete data:\n");
69     sqlite3_exec(db, "SELECT * FROM config;", callback, NULL, &errormsg);
70     printf("\n");
71
72     // 关闭数据库
73     sqlite3_close(db);
74
75     return 0;
76 }
```