

# 成长的点滴，记录与分享

此博客永不更新，包括技术文章，评论也不再回复，望海涵！请到 <http://darkbull.net>

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

## 个人资料



JGood

访问：613641次

积分：4312分

排名：第1888名

原创：74篇 转载：3篇

译文：3篇 评论：289条

[7月推荐文章汇总](#)[Android 精彩案例](#)[【独具慧眼 推荐有礼】找出您心中的技术大牛](#)[博文大赛获奖名单公布](#)[关注社区微信](#)[得下载分](#)

## Sqlite c/c++ api 学习

分类：[C/C++ Database](#)

2009-10-07 22:49

13591人阅读

[评论\(14\)](#)

[收藏](#)

[举报](#)

[sqlite](#)[api](#)[数据库](#)[sql](#)[callback](#)[insert](#)[目录\(?\)](#)[\[+\]](#)

Sqlite是我最喜欢使用的文件数据库，它小巧、方便、速度快、支持大部分标准sql语句、开源、免费……优点太多了。自从认识了她之后，我就将Access打入了冷宫，发誓以后再也不用Access了。我们可以到Sqlite的[官方网站](#)下载Sqlite的源代码，编译后的结果只有200多K，太小巧，太卡哇依了。如果嫌自己手动编译麻烦，也可以直接从官网下载二进制的可执行文件或者动态链接库。可执行文件可以用命令行的方式来创建、操作数据库；动态链接库则可以用程序来操作Sqlite数据库。今天我就来介绍一下如何使用c/c++来操作Sqlite吧~~~

## 准备工作

## 文章搜索

## 文章分类

[C/C++](#) (13)[Database](#) (4)[DotNet](#) (4)[Linux](#) (1)[Open Source](#) (4)[PHP](#) (0)[Python](#) (48)[Web](#) (2)[历史文化](#) (0)[杂七杂八](#) (17)[网络通信](#) (5)

## 文章存档

[2011年12月](#) (1)[2011年10月](#) (3)[2011年07月](#) (1)[2011年06月](#) (1)[2011年03月](#) (2)[展开](#)

## 阅读排行

我这里使用的sqlite版本是3.x, 所以首先从官网下载动态链接库[sqlite3.dll](#); 然后再下载api头文件[sqlite3.h](#)。在Code::Block中新建一个C++的控制台项目, 添加对sqlite3.dll的引用, 同时将sqlite3.h添加到项目中。我这里还使用了一个工具: [SqliteSpy](#), 用于对数据库进行可视化的操作。

## 开始使用API

下面就开始使用API吧~。Sqlite的所有api, 可以在[这里](#)查询到。

### sqlite3\_open

### sqlite3\_exec

### sqlite3\_close

看到这些函数, 我相信你大概能猜出这些函数的功能。sqlite3\_open用于打开一个数据库, 并初始化一个数据库的连接; sqlite3\_close用于关闭连接; sqlite3\_exec用于执行sql语句。用一个简单的例子来演示这些函数的使用: 下面的代码创建一个数据库test.db, 并创建数据表: test\_for\_cpp。

```
// -----  
// http://blog.csdn.net/JGood  
// 创建数据库  
// sqlite3_open, sqlite3_exec, sqlite3_close  
// -----  
  
sqlite3 *conn = NULL;    //数据库连接。  
char *err_msg = NULL;    //操作失败的原因。  
char sql[200];
```

libcurl教程

(46055)

Python模块学习 ---- sub

(40679)

Python模块学习 ---- thre

(40013)

Python模块学习 ---- date

(38651)

Python模块学习 --- urllib

(36173)

Python模块学习 ---- http

(33114)

Python模块学习 ---- re 正

(28537)

Python模块学习 ---- logg

(27406)

译: Code::Blocks手册 使

(25060)

Python模块学习 ---- struc

(24371)

## 评论排行

libcurl教程

(36)

再次封装IOCP的收获

(35)

MySQL Connector/C++

(28)

译: Code::Blocks手册 使

(20)

Python封装IOCP

(17)

Sqlite c/c++ api 学习

(14)

用Python写的图片蜘蛛人

(11)

Python模块学习 --- urllib

(7)

Python模块学习 ---- thre

(7)

Python模块学习 ---- struc

(7)

// 打开数据库, 创建连接

```
if (SQLITE_OK != sqlite3_open("test.db", &conn))
```

{

```
    printf("can't open the database.");
```

```
    exit(-1);
```

}

// 执行SQL

```
sprintf(sql, "CREATE TABLE [test_for_cpp] (/
```

```
    [id] int, [name] varchar(20), [age] int);
```

```
if (SQLITE_OK != sqlite3_exec(conn, sql, 0, 0, &err_msg))
```

{

```
    printf("operate failed: %s", err_msg);
```

```
    exit(-1);
```

}

// 关闭连接。

```
if (SQLITE_OK != sqlite3_close(conn))
```

{

```
    printf("can't close the database: %s/n", sqlite3_errmsg(conn));
```

```
    exit(-1);
```

}

```
printf("operate successfully. /n");
```

上面的例子代码都非常简单。主要的函数sqlite3\_exec比较复杂, 它的原形为:

## 推荐文章

## 最新评论

## libcurl教程

隐居士: @superyu79:使用  
CURLOPT\_TIMEOUT\_MS 设置  
连接超时, 具体见官网说明:  
ht...

## Sqlite c/c++ api 学习

igo2csdn: 谢谢

Python模块学习 ---- zipfile zip文  
yuzhu529: python压缩zip包的时  
候, 是否可以加密呢具体该用什  
么呢?

Python模块学习 ---- logging 日志  
chaos\_knight: nice thread

Python模块学习 ---- random 随机  
oranyujian: 学习了

Python模块学习 ---- random 随机  
shudou: 学习了

Python模块学习 ---- threading 多线程  
Bright-Thinking:  
@apoptosis:join是主调线程被阻  
塞, 如果要用join的话, 只能在  
seeker的run中使...

## libcurl教程

superyu79: @daofe:你好, 我也  
遇到调用curl\_easy\_perform卡  
死, 阻塞不返回, 请问你是怎么  
解决...

## libcurl教程

不要做咸鱼: @qiyating0808:怎  
么解析头部的数据呢?

python模块学习 ---- smtplib 邮件  
wenmingma: 很好很好, 谢谢!

```
int sqlite3_exec(
    sqlite3*,                               /* An open database */
    const char *sql,                         /* SQL to be evaluated */
    int (*callback)(void*,int,char**,char**), /* Callback function */
    void *,                                 /* 1st argument to callback */
    char **errmsg                             /* Error msg written here */
);
```

第三个参数是一个函数指针, 表示函数执行完毕后的回调。第四个参数是传递给回调函数的参数, 第五个参数表示执行失败时的原因。下面用一个简单的例子来演示该函数的详细使用:

```
// -----
// INSERT, SELECT
// sqlite3_exec的详细使用
// -----

sqlite3 *conn = NULL;    //数据库连接。
char *err_msg = NULL;    //操作失败的原因。
char sql[200];

// 回调函数声明。
extern int sqlite3_exec_callback(void *data, int n_columns,
    char **col_values, char **col_names);

// 打开数据库, 创建连接
sqlite3_open("test.db", &conn);
```

## 友情链接

[DarkSpy 的部落格](#)

[赖勇浩的编程私伙局](#)

[Taylor's SPACE](#)

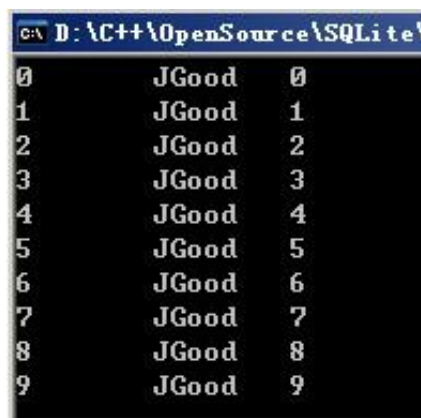
```
//添加10条记录
for (int i = 0; i < 10; i++)
{
    // 执行SQL
    sprintf(sql, "INSERT INTO [test_for_cpp] ([id], [name], [age]) /
        VALUES (%d, '%s', %d)", i, "JGood", i);
    sqlite3_exec(conn, sql, 0, 0, &err_msg);
}

// 查询
sprintf(sql, "SELECT * FROM [test_for_cpp]");
sqlite3_exec(conn, sql, &sqlite3_exec_callback, 0, &err_msg);

// 关闭连接。
sqlite3_close(conn);

//... ...
/** sqlite3_exec的回调。
 *
 * 向控制台打印查询的结果。
 *
 * @param in data 传递给回调函数的数据。
 * @param in n_columns sqlite3_exec执行结果集中列的数量。
 * @param in col_values sqlite3_exec执行结果集中每一列的数据。
 * @param in col_names sqlite3_exec执行结果集中每一列的名称。
 */
```

```
* @return 状态码。  
*/  
int sqlite3_exec_callback(void *data, int n_columns, char **col_values, char  
**col_names)  
{  
    for (int i = 0; i < n_columns; i++)  
    {  
        printf("%s/t", col_values[i]);  
    }  
    printf("/n");  
  
    return 0;  
}
```



0	JGood	0
1	JGood	1
2	JGood	2
3	JGood	3
4	JGood	4
5	JGood	5
6	JGood	6
7	JGood	7
8	JGood	8
9	JGood	9

下面是执行的结果：

### 使用事务

在上面的例子中，我们向数据库里插入了10条数据，然后再从数据库里读出来。细心的你不知道有没有发现，在执行Insert的时候，并没有使用同一个事务。在很多情况下，我们需要使用事务来保证对数据库操作的原子性。

Sqlite是支持事务的, 而且对事务的使用非常简单: 使用sql语句"begin;"表示事务开始, "rollback;"表示事务的回滚, "commit;"表示事务的提交。下面我们对上面例子中的代码作一下修改, 给Insert操作添加事务支持:

```
//... ...

bool is_succed = true;
sqlite3_exec(conn, "begin;", 0, 0, 0); // 开启事务
//添加10条记录
for (int i = 0; i < 10; i++)
{
    // 执行SQL
    sprintf(sql, "INSERT INTO [test_for_cpp] ([id], [name], [age]) /
        VALUES (%d, '%s', %d)", i, "JGood", i);
    if (SQLITE_OK != sqlite3_exec(conn, sql, 0, 0, &err_msg))
    {
        is_succed = false;
        break;
    }
}

if (is_succed)
    sqlite3_exec(conn, "commit;", 0, 0, 0); // 提交事务
else
    sqlite3_exec(conn, "rollback;", 0, 0, 0); // 回滚事务

//... ...
```

## 使用sql参数

基本上, 使用sqlite3\_open, sqlite3\_close, sqlite3\_exec这三个函数, 可以完成大部分的工作。但还不完善。上面的例子中, 都是直接以sql语句的形式来操作数据库, 这样很容易被注入。所以有必要使用sql参数。

sqlite3\_prepare

sqlite3\_bind\_\*

sqlite3\_step

sqlite3\_column\_\*

struct sqlite3\_stmt

sqlite3\_finalize

sqlite3\_prepare用来编译sql语句。sql语句被执行之前, 必须先编译成字节码。sqlite3\_stmt是一个结构体, 表示sql语句编译后的字节码。sqlite3\_step用来执行编译后的sql语句。sqlite3\_bind\_\*用于将sql参数绑定到sql语句。sqlite3\_column\_\*用于从查询的结果中获取数据。sqlite3\_finalize用来释放sqlite3\_stmt对象。代码最能说明函数的功能, 下面就用一个例子来演示吧~~

```
// -----  
// http://blog.csdn.net/JGood  
// sqlite3_prepare, sqlite3_bind_*, sqlite3_step, sqlite3_column_*,  
// sqlite3_column_type  
// sqlite3_stmt, sqlite3_finalize, sqlite3_reset  
// 查询
```



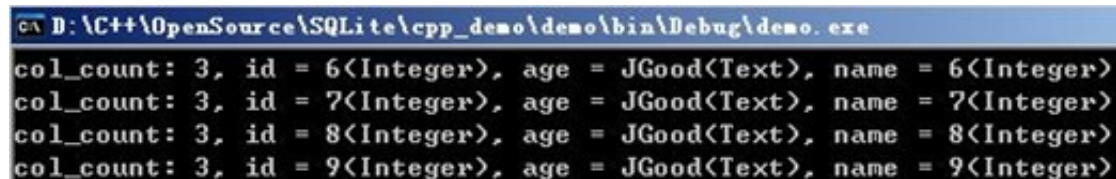
```
// -----  
sqlite3 *conn = NULL;  
sqlite3_stmt *stmt = NULL;  
const char *err_msg = NULL;  
// 列数据类型  
char col_types[][10] = { "", "Integer", "Float", "Text", "Blob", "NULL" };  
  
sqlite3_open("test.db", &conn);  
sqlite3_prepare(conn, "SELECT * FROM [test_for_cpp] WHERE [id]>", -1, &stmt,  
&err_msg);  
sqlite3_bind_int(stmt, 1, 5);  
  
while (SQLITE_ROW == sqlite3_step(stmt))  
{  
    int col_count = sqlite3_column_count(stmt); // 结果集中列的数量  
  
    const char *col_0_name = sqlite3_column_name(stmt, 0); // 获取列名  
    int id = sqlite3_column_int(stmt, 0);  
    int id_type = sqlite3_column_type(stmt, 0); // 获取列数据类型  
  
    const char *col_2_name = sqlite3_column_name(stmt, 2);  
    int age = sqlite3_column_int(stmt, 2);  
    int age_type = sqlite3_column_type(stmt, 2);  
  
    const char *col_1_name = sqlite3_column_name(stmt, 1);
```

```
char name[80];
strncpy(name, (const char *)sqlite3_column_text(stmt, 1), 80);
int name_type = sqlite3_column_type(stmt, 1);

// 打印结果
printf("col_count: %d, %s = %d(%s), %s = %s(%s), %s = %d(%s)/n",
       col_count, col_0_name, id, col_types[id_type], col_2_name, name,
       col_types[name_type], col_1_name, age, col_types[age_type]);
}

sqlite3_finalize(stmt); // 释放sqlite3_stmt
sqlite3_close(conn);
```

这段代码查询id号大于5的所有记录, 并显示到控制台, 最后效果为



```
D:\C++\OpenSource\SQLite\cpp_demo\demo\bin\Debug\demo.exe
col_count: 3, id = 6<Integer>, age = JGood<Text>, name = 6<Integer>
col_count: 3, id = 7<Integer>, age = JGood<Text>, name = 7<Integer>
col_count: 3, id = 8<Integer>, age = JGood<Text>, name = 8<Integer>
col_count: 3, id = 9<Integer>, age = JGood<Text>, name = 9<Integer>
```

## 其他函数

在上面的例子中, 还使用了其他的一些函数, 如: `sqlite3_column_count`用于获取结果集中列的数量; `sqlite3_column_name`用于获取列的名称; `sqlite3_column_type`用于获取列的数据类型; `sqlite3_errcode`用于获取最近一次操作出错的错误代码; `sqlite3_errmsg`用于获取最近一次操作出错的错误说明。sqlite的api中还有很多的函数, 有了上面的基础, 相信你通过查询官方的文档, 能迅速掌握本文未介绍的api。

## 字符串编码

在官网上查看Sqlite的api的时候, 发现有很同函数的名称都非常相似, 只是最后添加了”\_16”, 如: sqlite3\_open和sqlite3\_open16, sqlite3\_errmsg和sqlite3\_errmsg16, 等等。其实添加了”16”后缀的函数, 主要用于支持utf-16编码的字符串。如sqlite3\_open16可以接收utf-16编码的数据库路径。

在sourceforge上, 有一个开源的项目sqlitex, 它封装了这些api, 使对sqlite数据库的操作更加方便。sqlitex的源代码非常的简单, 感兴趣的同学可以下载下来自己研究。

- **参考文档:**
- **Sqlite官网:** <http://www.sqlite.org/>
- **Sqlite中文社区:** <http://www.sqlite.com.cn/>

上一篇 在ASP.NET中使用gzip压缩

下一篇 锻炼身体

主题推荐

sqlite

c++

api

控制台

源代码

猜你在找

sqlite3用法详解草稿

FastDb client-server模式

同步对象Event的用法

\_\_win32控件(2)\_\_教你自学掌握所有控件的玩法

DirectX游戏编程入门——第二部分（游戏编程工具箱）

C++访问sqlite3实践

高性能日志服务

vc 对话框修改菜单栏的背景颜色

wchar\_t内置还是别名？小问题一则

windows和linux中程序栈空间



## 免费注册 免费学编程

kaikeba.com

开课吧特级导师详细讲解编程开发 先注册,后学习,一切全免费噢!

查看评论

9楼 [igo2csdn](#) 2014-07-02 00:52发表



谢谢

8楼 [haithink](#) 2013-11-28 16:38发表



嗯, 文章不错!

7楼 [jiandan\\_sz](#) 2013-07-07 22:28发表



写得很清晰, 有收获, 谢谢!

6楼 [卫斯](#) 2011-11-03 09:45发表



我之前从某个网易的博客上看到这篇文, 貌似没有声明是转载的, 不管怎样, 这篇文是非常好的, 程序员会写代码的很多, 但是要清晰的把自己的思路讲出来却是不容易的。

Re: [JGood](#) 2011-11-03 11:43发表



回复cubepeng: 是原创的

5楼 [卫斯](#) 2011-11-03 09:37发表



您这事原文吗?

4楼 [xxw168](#) 2011-05-01 09:20发表



[e01]

收藏

3楼 [余独爱莲](#) 2011-04-06 17:37发表



[e01] 从中获益, 解决了问题, 感谢楼主!

2楼 [tinystone](#) 2010-10-12 16:29发表



LZ, 我有跟一楼一样的问题。C++项目添加DLL的引用, 是不是好要将sqlite3.dll放在工程的debug文件夹下面? 用Visual Studio2010怎么添加引用你知道吗?

1楼 [xddy1009](#) 2010-03-13 13:34发表



lz 问一下如何添加对sqlite3.dll的引用。谢谢[e01]

Re: [JGood](#) 2010-03-13 20:23发表



回复 xddy1009: 如果是gcc, 使用-l来指定要使用的dll. 在codeblocks中, 可以通过 settings->compiler and debugger settings -> Linker settings一设置.

Re: [xddy1009](#) 2010-03-14 23:59发表



回复 JGood: 哦, 呵呵, 我是用VC++的, 主要是为了得到SQLite 类型的数据库文件内容, 现在已经行了, 不过你这篇文章帮了我不少忙啊.

Re: [JGood](#) 2010-03-15 00:03发表



回复 xddy1009: [e04]

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

### 核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack  
VPN   Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery  
BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity  
Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack

[FTC](#) [coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#) [Rackspace](#) [Web App](#) [SpringSide](#) [Maemo](#)  
[Compuware](#) [大数据](#) [apttech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#) [HBase](#) [Pure](#) [Solr](#)  
[Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) [webmaster@csdn.net](#) 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2014, CSDN.NET, All Rights Reserved 