

## c语言对sqlite3的操作函数 - Dian0dian0的博客 - CSDN博客

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

首先要安装自己的sqlite3:<https://blog.csdn.net/Dian0dian0/article/details/89158275>

原型：

```
int sqlite3_exec(sqlite3* pDB, const char *sql, sqlite_callback callback, void*para, char** errMsg);
```

功能：

编译和执行零个或多个SQL 语句，查询的结果返回给回调函数callback

输入参数：

参数	说明
sqlite3* pDB	数据库句柄
const char *sql	待执行的SQL 语句字符串，以'\0'结尾
sqlite_callback callback	回调函数，用来处理查询结果，如果不需要回调（比如做insert 或者delete 操作时），可以输入NULL
char** errMsg	返回错误信息，注意是指针的指针
返回值 int	执行成功返回SQLITE_OK，否则返回其他值，SQLITE_OK默认为零

返回值

```
#define SQLITE_OK 0 /* Successful result */

#define SQLITE_ERROR 1 /* SQL error or missing database */
#define SQLITE_INTERNAL 2 /* Internal logic error in SQLite */
#define SQLITE_PERM 3 /* Access permission denied */
#define SQLITE_ABORT 4 /* Callback routine requested an abort */
#define SQLITE_BUSY 5 /* The database file is locked */
#define SQLITE_LOCKED 6 /* A table in the database is locked */
#define SQLITE_NOMEM 7 /* A malloc() failed */
#define SQLITE_READONLY 8 /* Attempt to write a readonly database */
#define SQLITE_INTERRUPT 9 /* Operation terminated by sqlite3_interrupt()*/
#define SQLITE_IOERR 10 /* Some kind of disk I/O error occurred */
#define SQLITE_CORRUPT 11 /* The database disk image is malformed */
#define SQLITE_NOTFOUND 12 /* Unknown opcode in sqlite3_file_control() */
#define SQLITE_FULL 13 /* Insertion failed because database is full */
#define SQLITE_CANTOPEN 14 /* Unable to open the database file */
#define SQLITE_PROTOCOL 15 /* Database lock protocol error */
#define SQLITE_EMPTY 16 /* Database is empty */
#define SQLITE_SCHEMA 17 /* The database schema changed */
#define SQLITE_TOOBIG 18 /* String or BLOB exceeds size limit */
#define SQLITE_CONSTRAINT 19 /* Abort due to constraint violation */
#define SQLITE_MISMATCH 20 /* Data type mismatch */
#define SQLITE_MISUSE 21 /* Library used incorrectly */
#define SQLITE_NOLFS 22 /* Uses OS features not supported on host */
#define SQLITE_AUTH 23 /* Authorization denied */
#define SQLITE_FORMAT 24 /* Auxiliary database format error */
#define SQLITE_RANGE 25 /* 2nd parameter to sqlite3_bind out of range */
#define SQLITE_NOTADB 26 /* File opened that is not a database file */
#define SQLITE_ROW 100 /* sqlite3_step() has another row ready */
#define SQLITE_DONE 101 /* sqlite3_step() has finished executing */
```

回调函数的原型：

```
typedef int (*sqlite_callback) (void* para, int columnCount, char** columnValue, char** columnName);
```

功能：

由用户处理查询的结果

参数：

参数	说明
<b>void* para</b>	从sqlite3_exec ( ) 传入的参数指针
<b>int columnCount</b>	查询到的这一条记录有多少个字段(即这条记录有多少列)；
<b>char** columnValue</b>	查询出来的数据都保存在这里，它实际上是个1 维数组（不要以为是2 维数组），每一个元素都是一个char * 值，是一个字段内容（用字符串来表示，以'\0'结尾）；
<b>char** columnName</b>	与columnValue 是对应的，表示这个字段的字段名称。
<b>返回值 int</b>	执行成功返回SQLITE_OK，否则返回其他值,SQLITE_OK 默认等于零

回调函数的格式如下：

```
int sqlite_callback(  
    void* pv,  
    int argc,  
    char** argv,  
    char** col  
);
```

参数格式：

传给sqlite3\_exec的回调函数,用来显示查询结果  
对每一条查询结果调用一次该回调函数

参数：

pv:由sqlite3\_exec传递的初始化参数  
argc:表头的列数  
col:表头的名字数组指针  
argv:表头的数据数组指针

返回值：

1:中断查找  
0:继续列举查询到的数据

示例表：

-----+-----
id   pic   data(16进制数据)
-----+-----
1   a.jpg   00 00 00 ...
-----+-----
2   b.jpg   XX XX XX
-----+-----

对第一行数据：

argc=3 即 [0]...[2]  
argv[0]="1",argv[1]="a.jpg",argv[2]="00 00 00..."(实际16进制数据,非这里显示的字符串形式)  
col[0]="id",col[1]="pic",col[2]="data"

说明：

sqlite3\_exec() 的回调函数必须按照此格式，当然形参的名字任意。  
如果某列的数据类型不是char\*，则可以对结果执行相关的转换，如:用atoi()把结果转换为整数(integer)，如果是二进制数据，则可以直接强制类型转换，如:(void\*)argv[i]。  
该回调函数有两种返回值类型。

- 1.返回零:sqlite3\_exec() 将继续执行查询。
- 2.返回非零:sqlite3\_exec()将立即中断查询，且 sqlite3\_exec() 将返回 SQLITE\_ABORT。

示例：

```
int i;  
for(i=0; i<argc; i++)  
{  
    printf("%s\t%s\n", col[i], argv[i]);  
}
```

原文参考：<https://www.cnblogs.com/nbsofer/archive/2012/05/29/2523807.html>

文章最后发布于: 2019-04-10 13:57:30