[SQLite 插入大量数据慢的解决方法](http://www.cnblogs.com/likebeta/archive/2012/06/15/2551466.html)

sqlite 插入数据很慢的原因：sqlite在没有显式使用事务的时候会为每条insert都使用事务操作，而sqlite数据库是以文件的形式存在磁盘中，就相当于每次访问时都要打开一次文件，如果对数据进行大量的操作，时间都耗费在I/O操作上，所以很慢。  
解决方法是显式使用事务的形式提交：因为我们开始事务后，进行的大量操作的语句都保存在内存中，当提交时才全部写入数据库，此时，数据库文件也就只用打开一次。  
我在没有显式使用事务形式插入100条数据时用了12.226s；用显式事务形式，插入100条只用了0.172s，插入1000000条也才34.891s，相关很大吧。  
**显式使用事务的例子：**

[复制代码](javascript:void(0);)

#include <iostream>

#include <windows.h>

using namespace std;

#include "sqlite/sqlite3.h"

int main()

{

sqlite3\* db;

int nResult = sqlite3\_open("test.db",&db);

if (nResult != SQLITE\_OK)

{

cout<<"打开数据库失败："<<sqlite3\_errmsg(db)<<endl;

return 0;

}

else

{

cout<<"数据库打开成功"<<endl;

}

char\* errmsg;

nResult = sqlite3\_exec(db,"create table fuck(id integer primary key autoincrement,name varchar(100))",NULL,NULL,&errmsg);

if (nResult != SQLITE\_OK)

{

sqlite3\_close(db);

cout<<errmsg;

sqlite3\_free(errmsg);

return 0;

}

string strSql;

strSql+="begin;\n";

for (int i=0;i<100;i++)

{

strSql+="insert into fuck values(null,'heh');\n";

}

strSql+="commit;";

//cout<<strSql<<endl;

SYSTEMTIME tm\_s;

GetLocalTime(&tm\_s);

nResult = sqlite3\_exec(db,strSql.c\_str(),NULL,NULL,&errmsg);

SYSTEMTIME tm\_e;

GetLocalTime(&tm\_e);

if (nResult != SQLITE\_OK)

{

sqlite3\_close(db);

cout<<errmsg<<endl;

sqlite3\_free(errmsg);

return 0;

}

cout<<"start:"<<tm\_s.wMinute<<":"<<tm\_s.wSecond<<":"<<tm\_s.wMilliseconds<<endl;

cout<<"end :"<<tm\_e.wMinute<<":"<<tm\_e.wSecond<<":"<<tm\_e.wMilliseconds<<endl;

return 0;

}