 在工作中，我总是使用编译后sqlite3可执行程序，进行数据库的创建；其实我们操作sqlite数据库，  
也可是使用源代码级别的形式去操作数据库。我在网上查找，对部分代码进行修改，添加上注释，  
形式如下代码，其中包含对数据表的创建，添加数据，查询数据功能，代码如下：

1. #include <stdio.h>
2. #include <sqlite3.h>
3. //查询的回调函数声明
4. int select\_callback(void \* data, int col\_count, char \*\* col\_values, char \*\* col\_Name);
5. int main(int argc, char \* argv[])
6. {
7. const char \* sSQL1 = "create table users(userid varchar(20) PRIMARY KEY, age int, birthday datetime);";
8. char \* pErrMsg = 0;
9. int result = 0;
10. // 连接数据库
11. sqlite3 \* db = 0;
12. int ret = sqlite3\_open("./test.db", &db);
13. if( ret != SQLITE\_OK ) {
14. fprintf(stderr, "无法打开数据库: %s", sqlite3\_errmsg(db));
15. return(1);
16. }
17. printf("数据库连接成功!\n");
18. // 执行建表SQL
19. sqlite3\_exec( db, sSQL1, 0, 0, &pErrMsg );
20. if( ret != SQLITE\_OK ){
21. fprintf(stderr, "SQL error: %s\n", pErrMsg);
22. sqlite3\_free(pErrMsg);
23. }
24. // 执行插入记录SQL
25. result = sqlite3\_exec( db, "insert into users values('张三',20,'2011-7-23');", 0, 0, &pErrMsg);
26. if(result == SQLITE\_OK){
27. printf("插入数据成功\n");
28. }
29. result = sqlite3\_exec( db, "insert into users values('李四',20,'2012-9-20');", 0, 0, &pErrMsg);
30. if(result == SQLITE\_OK){
31. printf("插入数据成功\n");
32. }
33. // 查询数据表
34. printf("查询数据库内容\n");
35. sqlite3\_exec( db, "select \* from users;", select\_callback, 0, &pErrMsg);
36. // 关闭数据库
37. sqlite3\_close(db);
38. db = 0;
39. printf("数据库关闭成功!\n");
40. return 0;
41. }
42. int select\_callback(void \* data, int col\_count, char \*\* col\_values, char \*\* col\_Name)
43. {
44. // 每条记录回调一次该函数,有多少条就回调多少次
45. int i;
46. for( i=0; i < col\_count; i++){
47. printf( "%s = %s\n", col\_Name[i], col\_values[i] == 0 ? "NULL" : col\_values[i] );
48. }
49. return 0;
50. }

编写好代码后，我们需要进行编译。  
我使用的ubuntu10.04自带的gcc编译器gcc4.4.3进行编译。  
如果你使用 $ gcc sqlitetest.c  
进行编译时，会出现如下的错误：  
/tmp/ccyxvdme.o: In function `main':  
sqlitetest.c:(.text+0x31): undefined reference to `sqlite3\_open'  
sqlitetest.c:(.text+0x48): undefined reference to `sqlite3\_errmsg'  
sqlitetest.c:(.text+0xa5): undefined reference to `sqlite3\_exec'  
sqlitetest.c:(.text+0xd6): undefined reference to `sqlite3\_free'  
sqlitetest.c:(.text+0x102): undefined reference to `sqlite3\_exec'  
sqlitetest.c:(.text+0x12e): undefined reference to `sqlite3\_exec'  
sqlitetest.c:(.text+0x15a): undefined reference to `sqlite3\_exec'  
sqlitetest.c:(.text+0x166): undefined reference to `sqlite3\_close'  
collect2: ld returned 1 exit status  
这是因为gcc找不到定义sqlite3\_open等等的头文件，经过在网上查找资料，  
找到了解决方法，使用如下的方法进行编译：  
$ gcc -o hello -L /usr/local/sqlite-autoconf-3070400/lib -I/usr/local/sqlite-autoconf-307040/include sqlitetest.c  -lsqlite3  
上面的编译主要意义时-L 代码你安装sqlite3类库所在的路径， -I代表安装sqlite3的头文件路径 而-l表示  
可执行程序的名称  
经过上面的编译，即可成功。  
执行结果如下：  
~$ ./hello   
数据库连接成功!  
插入数据成功  
插入数据成功  
查询数据库内容  
userid = 张三  
age = 20  
birthday = 2011-7-23  
userid = 李四  
age = 20  
birthday = 2012-9-20  
数据库关闭成功!