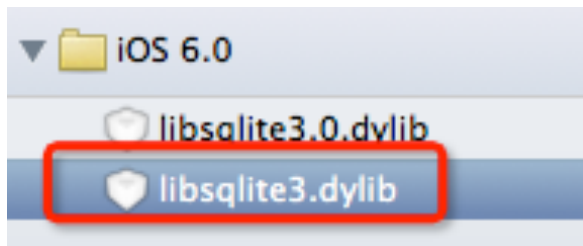
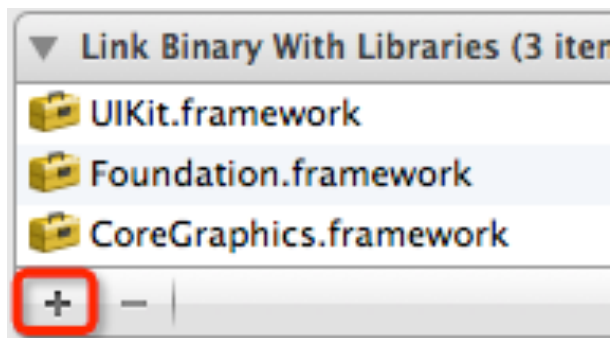


SQLite3

- 在iOS中使用SQLite3，首先要添加库文件libsqlite3.dylib和导入主头文件



```
#import <sqlite3.h>
```

创建、打开、关闭数据库

- 创建或打开数据库

// path是数据库文件的存放路径

```
sqlite3 *db = NULL;
```

```
int result = sqlite3_open([path UTF8String], &db);
```

- 代码解析：

- ◆ `sqlite3_open()`将根据文件路径打开数据库，如果不存在，则会创建一个新的数据库。如果`result`等于常量**SQLITE_OK**，则表示成功打开数据库
- ◆ `sqlite3 *db`：一个打开的数据库实例
- ◆ 数据库文件的路径必须以C字符串(而非NSString)传入

- 关闭数据库：`sqlite3_close(db);`

执行不返回数据的SQL语句

- 执行创表语句

```
char *errorMsg = NULL; // 用来存储错误信息
```

```
char *sql = "create table if not exists t_person(id integer primary key autoincrement,  
name text, age integer);";
```

```
int result = sqlite3_exec(db, sql, NULL, NULL, &errorMsg);
```

- 代码解析：

- ◆ **sqlite3_exec()**可以执行任何SQL语句，比如创表、更新、插入和删除操作。但是一般不用它执行查询语句，因为它不会返回查询到的数据
- ◆ **sqlite3_exec()**还可以执行的语句：
 - ① 开启事务：begin transaction;
 - ② 回滚事务：rollback;
 - ③ 提交事务：commit;

带占位符插入数据

```
char *sql = "insert into t_person(name, age) values(?, ?);";  
sqlite3_stmt *stmt;  
if (sqlite3_prepare_v2(db, sql, -1, &stmt, NULL) == SQLITE_OK) {  
    sqlite3_bind_text(stmt, 1, "母鸡", -1, NULL);  
    sqlite3_bind_int(stmt, 2, 27);  
}  
if (sqlite3_step(stmt) != SQLITE_DONE) {  
    NSLog(@"插入数据错误");  
}  
sqlite3_finalize(stmt);
```

- 代码解析：
- ◆ `sqlite3_prepare_v2()`返回值等于**SQLITE_OK**，说明SQL语句已经准备成功，没有语法问题

带占位符插入数据

- ◆ `sqlite3_bind_text()`: 大部分绑定函数都只有3个参数

- ① 第1个参数是`sqlite3_stmt *`类型

- ② 第2个参数指占位符的位置，**第一个占位符的位置是1，不是0**

- ③ 第3个参数指占位符要绑定的值

- ④ 第4个参数指在第3个参数中所传递数据的长度，对于C字符串，可以传递-1代替字符串的长度

- ⑤ 第5个参数是一个可选的函数回调，一般用于在语句执行后完成内存清理工作

- ◆ `sqlite_step()`: 执行SQL语句，返回`SQLITE_DONE`代表成功执行完毕

- ◆ `sqlite_finalize()`: 销毁`sqlite3_stmt *`对象

查询数据

```
char *sql = "select id,name,age from t_person;";
sqlite3_stmt *stmt;
if (sqlite3_prepare_v2(db, sql, -1, &stmt, NULL) == SQLITE_OK) {
    while (sqlite3_step(stmt) == SQLITE_ROW) {
        int _id = sqlite3_column_int(stmt, 0);
        char *_name = (char *)sqlite3_column_text(stmt, 1);
        NSString *name = [NSString stringWithUTF8String:_name];
        int _age = sqlite3_column_int(stmt, 2);
        NSLog(@"id=%i, name=%@, age=%i", _id, name, _age);
    }
}
sqlite3_finalize(stmt);
```

- 代码解析

- ◆ **sqlite3_step()**返回SQLITE_ROW代表遍历到一条新记录
- ◆ **sqlite3_column_***()用于获取每个字段对应的值，第2个参数是字段的索引，从0开始