**IOS 数据存储之 FMDB 详解**

FMDB是用于进行数据存储的第三方的框架，它与SQLite与Core Data相比较，存在很多优势。

FMDB是面向对象的，它以OC的方式封装了SQLite的C语言API，使用起来更加的方便，不需要过多的关心数据库操作的知识。但是它本身也存在一些问题，比如跨平台，因为它是用oc的语言封装的，所以只能在ios开发的时候使用，如果想实现跨平台的操作，来降低开发的成本和维护的成本，就需要使用比较原始的SQLite。

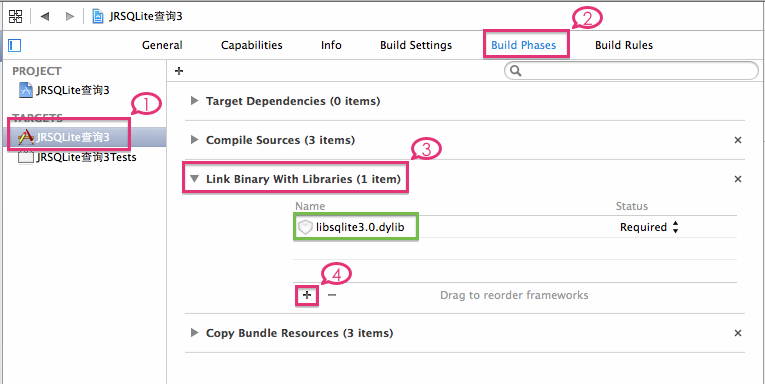
Core Data是ORM的一种体现，使用Core Data需要用到模型数据的转化，虽然操作简单，不需要直接操作数据库，但是性能没有直接使用SQLite高。但是SQLite使用的时候需要使用c语言中的函数，操作比较麻烦，因此需要对它进行封装。但是如果只是简单地封装，很可能会忽略很多重要的细节，比如如何处理并发以及安全性更问题。

因此，在这里推荐使用第三方框架FMDB，它是对libsqlite3框架的封装，用起来的步骤与SQLite使用类似，并且它对于多线程的同时操作一个表格时进行了处理，也就意味着它是线程安全的。FMDB是轻量级的框架，使用灵活，它是很多企业开发的首选。  
FMDB中重要的类

FMDatabase：一个FMDatabase对象就代表一个单独的SQLite数据库，用来执行SQL语句

FMResultSet：使用FMDatabase执行查询后的结果集

FMDatabaseQueue：用于在多线程中执行多个查询或更新，它是线程安全的

[](http://s4.51cto.com/wyfs02/M02/6F/7B/wKiom1Wd4uyzKVCHAADJY9nw8A8478.png)

**FMDB使用步骤**

1. 下载FMDB文件 fmdb下载地址 ，将FMDB文件夹添加到项目中

2. 导入sqlite框架，导入FMDatabase.h文件

3.与SQLite使用步骤类似，需要获取数据库文件路径，然后获得数据库，并打开数据库，然后数据库进行操作，最后关闭数据库。代码如下所示：

1. // ViewController.m
2. // JRFMDB
3. //
4. // Created by jerehedu on 15/6/18.
5. // Copyright (c) 2015年 jerehedu. All rights reserved.
6. //
8. #**import** "ViewController.h"
9. #**import** "FMDatabase.h"
11. **@interface** ViewController ()
12. @property (nonatomic, strong) FMDatabase \*db;
14. @end
16. @implementation ViewController
18. - (**void**)viewDidLoad
19. {
20. [**super** viewDidLoad];
21. //导入sqlite框架，导入FMDB文件夹
23. //1.获得数据库文件的路径
24. NSString \*doc=[NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES) lastObject];
25. NSString \*fileName=[doc stringByAppendingPathComponent:@"student.sqlite"];
26. NSLog(@"fileName = %@",fileName);
28. //2.获得数据库
29. FMDatabase \*db = [FMDatabase databaseWithPath:fileName];
31. //3.打开数据库
32. **if** ([db open]) {
33. NSLog(@"ok");
35. //4.创表
36. BOOL result=[db executeUpdate:@"CREATE TABLE IF NOT EXISTS t\_student (id integer PRIMARY KEY AUTOINCREMENT, name text NOT NULL, age integer NOT NULL);"];
37. **if** (result) {
38. NSLog(@"创表成功");
39. }**else**{
40. NSLog(@"创表失败");
41. }
42. }
43. self.db=db;
45. //插入数据
46. [self insertStu];
47. [self deleteStu:6];
48. [self updateStu:@"apple7\_name" :@"7777"];
50. [self queryStu];
51. [self dropStu];
52. [self insertStu];
53. [self queryStu];
55. //6.关闭数据库
56. [self.db close];
57. }
59. #pragma mark 插入数据
60. -(**void**)insertStu
61. {
62. **for** (**int** i=0; i<10; i++)
63. {
64. NSString \*name = [NSString stringWithFormat:@"1apple%i\_name",i];
65. **int** age = arc4random()%3+20;
67. //1. executeUpdate : 不确定的参数用?来占位 (后面参数必须都是oc对象)
68. [self.db executeUpdate:@"INSERT INTO t\_student (name, age) VALUES (?,?);",name,@(age)];
70. //2. executeUpdateWithFormat : 不确定的参数用%@、%d等来占位 （参数为原始数据类型）
71. // [self.db executeUpdateWithFormat:@"insert into t\_student (name, age) values (%@, %i);",name,age];
73. //3. 数组
74. // [self.db executeUpdate:@"INSERT INTO t\_student (name, age) VALUES (?,?);" withArgumentsInArray:@[name,@(age)]];
75. }
77. }
79. #pragma mark 删除数据
80. -(**void**)deleteStu:(**int**)idNum
81. {
82. //a. executeUpdate : 不确定的参数用?来占位 (后面参数必须都是oc对象)
83. // [self.db executeUpdate:@"delete from t\_student where id=?;",@(idNum)];
85. //b. executeUpdateWithFormat : 不确定的参数用%@、%d等来占位
86. // [self.db executeUpdateWithFormat:@"delete from t\_student where name=%@;",@"apple9\_name"];
87. }
89. #pragma mark 销毁表格
90. -(**void**)dropStu
91. {
92. [self.db executeUpdate:@"drop table if exists t\_student;"];
94. //4.创表
95. BOOL result=[self.db executeUpdate:@"CREATE TABLE IF NOT EXISTS t\_student (id integer PRIMARY KEY AUTOINCREMENT, name text NOT NULL, age integer NOT NULL);"];
96. **if** (result) {
97. NSLog(@"再次创表成功");
98. }**else**{
99. NSLog(@"再次创表失败");
100. }
101. }
103. #pragma mark 修改数据
104. -(**void**)updateStu:(NSString \*)oldName :(NSString\*)newName
105. {
106. // [self.db executeUpdateWithFormat:@"update t\_student set name=%@ where name=%@;",newName,oldName];
107. [self.db executeUpdate:@"update t\_student set name=? where name=?",newName,oldName];
108. }
110. #pragma mark 查询数据
111. -(**void**)queryStu
112. {
113. //1.执行查询语句
114. // FMResultSet \*resultSet = [self.db executeQuery:@"select \* from t\_student;"];
115. FMResultSet \*resultSet = [self.db executeQuery:@"select \* from t\_student where id<?;",@(14)];
117. //2.遍历结果集合
118. **while** ([resultSet next]) {
119. **int** idNum = [resultSet intForColumn:@"id"];
120. NSString \*name = [resultSet objectForColumnName:@"name"];
121. **int** age = [resultSet intForColumn:@"age"];
122. NSLog(@"id=%i ,name=%@, age=%i",idNum,name,age);
123. }
125. }
127. - (**void**)didReceiveMemoryWarning
128. {
129. [**super** didReceiveMemoryWarning];
130. // Dispose of any resources that can be recreated.
131. }
133. @end

【编辑推荐】

1. [关闭iOS 8.3验证 iOS新版系统占有率超40%](http://mobile.51cto.com/news-483594.htm)
2. [iOS开发者须知：你在开发过程中可能会遇到的问题](http://mobile.51cto.com/hot-483597.htm)
3. [iOS 9 beta 3发布，处处有新意](http://mobile.51cto.com/iphone-483601.htm)