SQLite Example

環境
- iOS 6.1
- Xcode 4.6.2

命令列Objective-C程式使用SQLite3

寫兩個簡單的命令列來測試查詢和寫入sqlite database，資料庫的建立可以使用sqlite3指令做到。

testSqlite1.m 查詢sqlite3 database

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

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        sqlite3* database;
        if(sqlite3_open([@"/Users/tzuyichao/lab/objectivec/test.sqlite" UTF8String],
&database) == SQLITE_OK) {
            NSLog( @"Connected..." );
            const char* sqlStatement = "select * from song";
            sqlite3_stmt* compiledStmt;
            if(sqlite3_prepare_v2(database, sqlStatement, -1, &compiledStmt, NULL) ==
SQLITE_OK) {
                NSLog( @"Go Query..." );
                while(sqlite3_step(compiledStmt) == SQLITE_ROW) {
                    NSString* name = [NSString stringWithUTF8String:(char *)
sqlite3_column_text(compiledStmt, 1)];
                    NSString* description = [NSString stringWithUTF8String:(char *)
sqlite3_column_text(compiledStmt, 2)];
                    NSLog( @"Name: %@", name );
                    NSLog( @"Desc: %@", description );
                }
            }
            sqlite3_finalize(compiledStmt);
        }
        sqlite3_close(database);
    }
    return 0;
}
```

Compile的指令是
clang -fobjc-arc -lsqlite3 testSqlite1.m -o testSqlite1

testSqlite2.m 寫入一筆資料到sqlite3 database

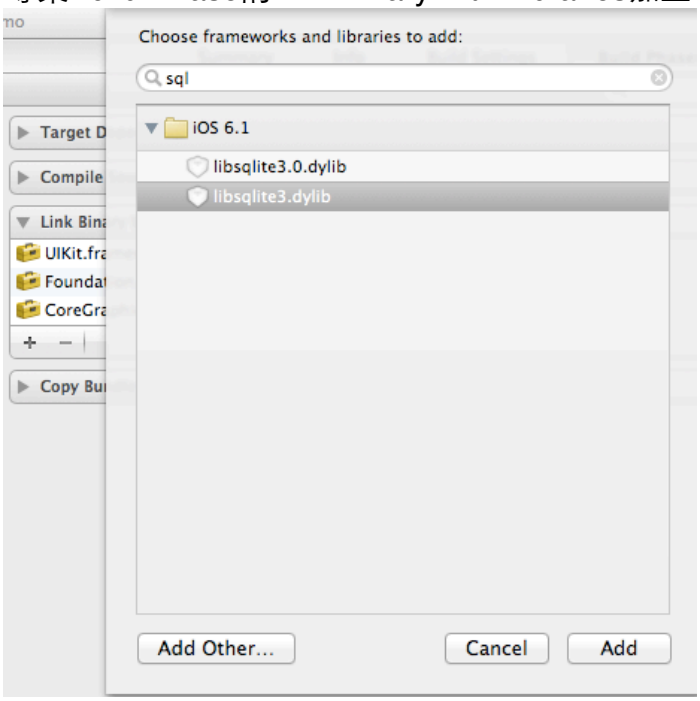
```
#import <Foundation/Foundation.h>
#import <sqlite3.h>

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        sqlite3* database;
        if(sqlite3_open([@"/Users/tzuyichao/lab/objectivec/test.sqlite" UTF8String],
&database) == SQLITE_OK) {
            NSLog( @"Connected..." );
            const char* sqlStatement = "insert into song (name, description) values
('test', 'test')";
            sqlite3_stmt* compiledStmt;
            sqlite3_prepare_v2(database, sqlStatement, -1, &compiledStmt, NULL);
            if(sqlite3_step(compiledStmt) == SQLITE_DONE) {
                NSLog( @"Inserted..." );
            } else {
                NSLog( @"Insrt Failed..." );
            }
            sqlite3_finalize(compiledStmt);
        }
        sqlite3_close(database);
    }
    return 0;
```

```
}
```

Compile的指令是
clang -fobjc-arc -lsqlite3 testSqlite2.m -o testSqlite2

iOS app做法

- 準備sqlite3
專案Build Phase的Link Binary with Libraries加上libsqlite3.dylib



接下來就是import sqlite3的include file和設定sqlite database需要的property

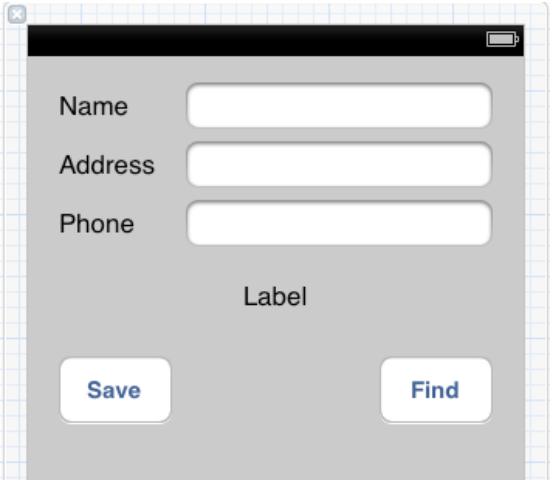
```
#import <UIKit/UIKit.h>
#import <sqlite3.h>

@interface ViewController : UIViewController

@property (strong, nonatomic) IBOutlet UITextField* name;
@property (strong, nonatomic) IBOutlet UITextField* address;
@property (strong, nonatomic) IBOutlet UITextField* phone;
@property (strong, nonatomic) IBOutlet UILabel* status;

-(IBAction)saveData;
-(IBAction)findContact;

// Database function
@property (strong, nonatomic) NSString* databasePath;
@property (nonatomic) sqlite3* contactDB;

@end
```

- 建立database和table
因為是iOS app我們必須要在第一次執行的時候建立sqlite database並且建立table。這個功能就做在自己寫的initView method，在viewDidLoad method使用。關鍵在於取得可以用來放置sqlite database file的位置之後，透過NSFileManager檢查檔案是否存在，就可以用來

判斷是否需要進行sqlite database的建立和tables的建立。當然如果有升版異動資料庫的時候這邊就會更複雜。

```
-(void)initView {
    NSString* docsDir;
    NSArray* dirPaths;

    // get documnet paths
    dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES);
    docsDir = dirPaths[0];
    NSLog( @"docsDir is %@", docsDir );

    // Build path to the database file
    [self setDatabasePath:[[NSString alloc]initWithString:[docsDir
stringByAppendingPathComponent:@"contacts.db"]]];
    NSLog( @"Database Path: %@", [self databasePath] );

    NSFileManager* fileMgr = [NSFileManager defaultManager];

    // 如果database file不存在就要產生database file和產生table
    // 如果這裡initial失敗，記得把模擬器的app移除，否則檔案存在，有可能table沒建立
    // 下次進來就不會重新initial，就會debug半天找不出問題為啥寫入查詢都失敗
    if( [fileMgr fileExistsAtPath:[self databasePath]] == NO ) {
        const char* dbPath = [[self databasePath] UTF8String];
        if(sqlite3_open(dbPath, &_contactDB) == SQLITE_OK) {
            char* errMsg;
            const char* sql_stmt = "CREATE TABLE IF NOT EXISTS CONTACTS (ID INTEGER
PRIMARY KEY AUTOINCREMENT, NAME TEXT, ADDRESS TEXT, PHONE TEXT)";
            if(sqlite3_exec(_contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK) {
                [[self status] setText:@"Failed to create database"];
                NSLog( @"Failed to create database" );
            }
            sqlite3_close(_contactDB);
        } else {
            NSLog( @"Failed to open/create database" );
            [[self status] setText:@"Failed to open/create database"];
        }
    }
}

- (void)viewDidLoad
{
    [super viewDidLoad];
     // Do any additional setup after loading the view, typically from a nib.
    [self initView];
}
```

- 處理寫入資料庫和查詢資料庫事件



我們要在Save button的event handler處理寫入資料庫；在Find button的event handler處理透過name查詢address, phone的功能，而Label則是用來顯示訊息使用。
寫入資料庫的部分

```
-(void)saveData {
    NSLog( @"save data called" );
    NSLog( @"Database Path: %@", [self databasePath] );
    const char* dbpath = [_databasePath UTF8String];
    sqlite3_stmt* statement;

    if(sqlite3_open(dbpath, &_contactDB) == SQLITE_OK) {
        NSString* insertSQL = [NSString stringWithFormat:@"INSERT INTO
CONTACTS (NAME, ADDRESS, PHONE) VALUES (\'%@\', \'%@\', \'%@\')",
self.name.text, self.address.text, self.phone.text];
        NSLog( @"%@", insertSQL );
        const char* insert_stmt = [insertSQL UTF8String];

        sqlite3_prepare_v2(_contactDB, insert_stmt, -1, &statement,
NULL);
        if(sqlite3_step(statement) == SQLITE_DONE) {
            self.name.text = @"";
            self.address.text = @"";
            self.phone.text = @"";
            [[self status]setText:@"add contact success"];
            NSLog( @"add contact success" );
        } else {
            [[self status]setText:@"Failed to add contact"];
            NSLog( @"Failed to add contact" );
        }
        sqlite3_finalize(statement);
        sqlite3_close(_contactDB);
    }
}
```

查詢的部分

```
-(void)findContact {
    NSLog( @"find contact called" );
    NSLog( @"Database Path: %@", [self databasePath] );
    const char* dbpath = [_databasePath UTF8String];
    sqlite3_stmt* statement;

    if(sqlite3_open(dbpath, &_contactDB) == SQLITE_OK) {
        NSString* querySQL = [NSString stringWithFormat:@"SELECT
address, phone FROM CONTACTS WHERE name=\'%@\'", self.name.text];
        NSLog( @"%@", querySQL );
        const char* query_stmt = [querySQL UTF8String];
        if(sqlite3_prepare_v2(_contactDB, query_stmt, -1, &statement,
NULL) == SQLITE_OK) {
            if(sqlite3_step(statement) == SQLITE_ROW) {
                // 查到了
                NSString* addressFiled = [[NSString
alloc]initWithUTF8String:(const char*)sqlite3_column_text(statement,
0)];
                NSString* phoneFiled = [[NSString
alloc]initWithUTF8String:(const char*)sqlite3_column_text(statement,
1)];
                [[self address]setText:addressFiled];
                [[self phone]setText:phoneFiled];
                [[self status]setText:@"Match Found"];
```

```
            } else {
                // 沒查到鬼東西
                [[self status] setText: @"Match Not Found"];
                [[self address] setText:@""];
                [[self phone] setText:@""];
            }
        } else {
            NSLog( @"prepare failed" );
        }
        sqlite3_finalize(statement);
        sqlite3_close(_contactDB);
    }

}
```

Reference

http://www.techotopia.com/index.php/An_Example_SQLite_based_iOS_6_iPhone_Application