

NSArray

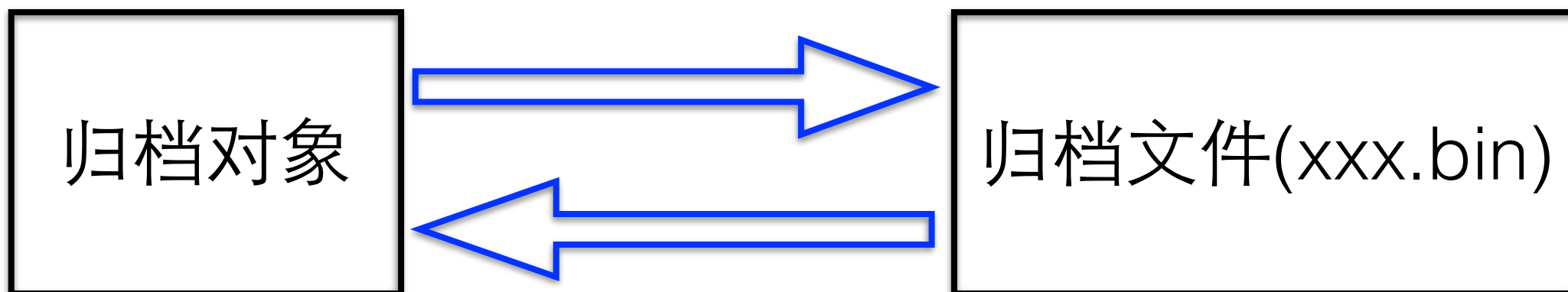
NSDictionary 第一个元素

name = "Bob"  
age = 19

NSDictionary 第二个元素

name = "Jonny"  
age = 20

归档 / 序列化



反归档 / 反序列化

## 数据库 personDB.db

表一： person

id	name	age	class
1	Jonny	19	class1
2	Bob	17	class2
3	Maggie	18	class3

表二： details

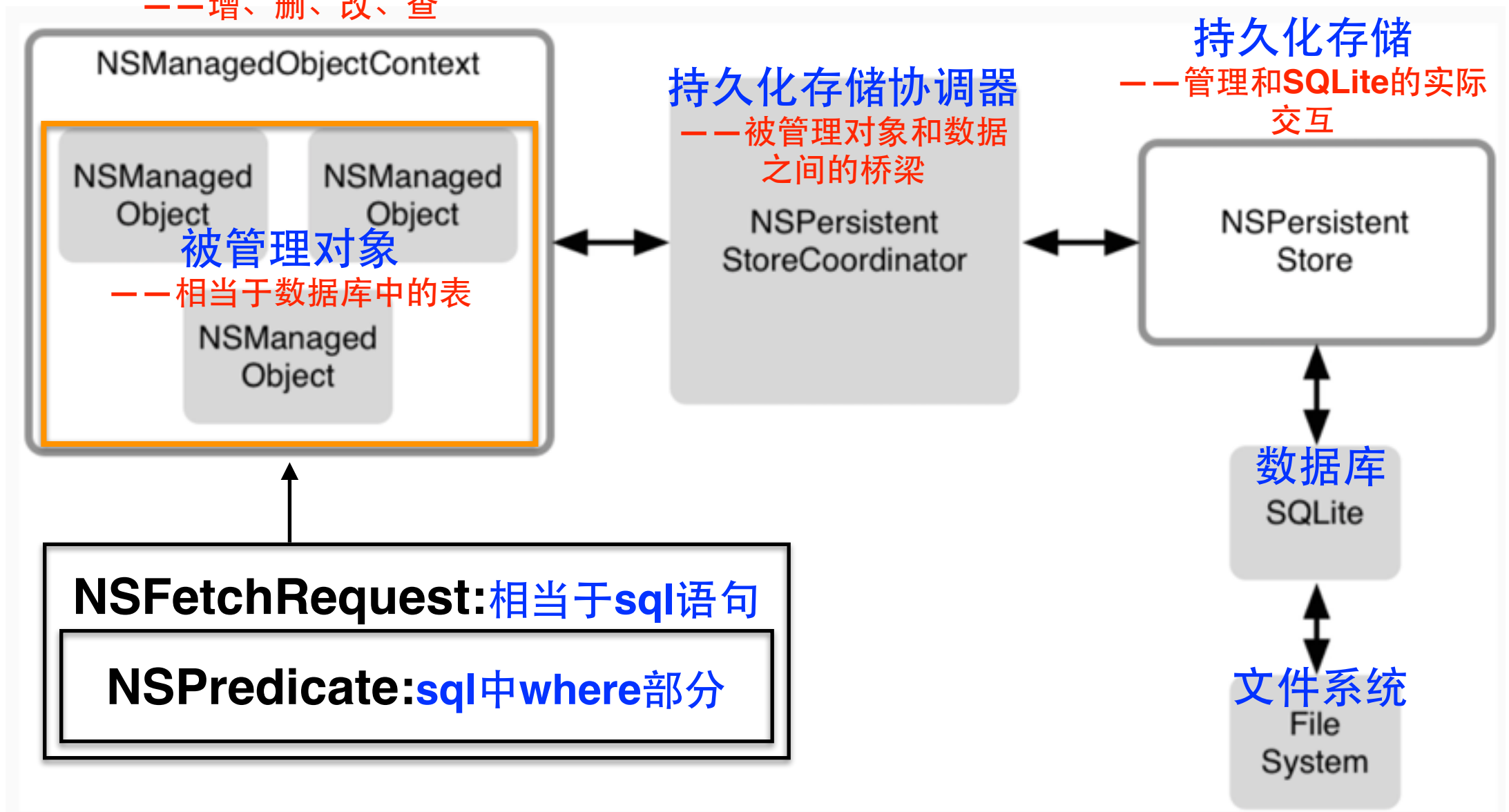
id	name	father	country	birth place
1	Jonny	Tom	Amerian	New York
2	Bob	Jerry	Canada	Ottawa
3	Maggie	Jack	Australia	Sydney

表三

.....

## 被管理对象上下文

——增、删、改、查



1. **data model(数据模型文件.xcdatamodel):**相当于数据库文件;
2. 数据模型对应OC中的**NSManagedObjectModel(被管理的对象模型类)**:管理数据库中的所有“表”

**目标：** 使用Core data解决方案中的NSFetchedResultsController控制器来控制table view的数据显示

**实现功能：** 好友的插入、查询、删除和更新

**准备工作：**

1. 创建工程，选择“User Core Data”选项
2. 导入<CoreData/CoreData.h>
3. 添加Entity(实体),并添加属性、设置属性的类型
4. 创建被管理对象类(NSManagedObject)

功能一： 添加 / 插入好友记录

1. ContactTableViewController

2. AddContactViewController



## 功能一：添加／插入好友记录

### 2. AddContactViewController

取消	Title	保存
名字		
<input type="text" value="Jonny"/>		
签名		
<input type="text" value="hello world"/>		

点击“保存”按钮

1. 保存到sqlite

**步骤1. 在类2中**，实现第2个类中的save点击函数

- 1.1 从结果控制器中获取一个空的模型对象
- 1.2 将用户输入的名字和签名赋值给该空对象
- 1.3 使用上下文将赋值完毕的模型对象保存到数据库文件中
- 1.4 退出当前控制器

## 功能一：添加／插入好友记录

### 2. AddContactViewController



The screenshot shows a mobile app interface for adding a contact. At the top is a title bar with a '取消' (Cancel) button on the left and a '保存' (Save) button on the right. Below the title bar are two text input fields. The first field is labeled '名字' (Name) and the second field is labeled '签名' (Signature). Both fields are currently empty.

点击“保存”按钮

2. 结果控制器监听到sqlite的变化

**步骤2.** 在**类1**中，实现“结果控制器”中的四个协议方法

2.1 实现针对行的监听方法(此时type为insert)

2.2 实现针对section的监听方法(此时type为insert)

2.3 实现“结果控制器”将要发生变化的监听方法

2.4 实现“结果控制器”已经发生变化的监听方法



## 功能一：添加／插入好友记录

### 2. AddContactViewController

The screenshot shows a mobile app interface for adding a contact. At the top, there's a header bar with a '取消' (Cancel) button on the left, a 'Title' in the center, and a '保存' (Save) button on the right. Below the header, there are two text input fields. The first field is labeled '名字' (Name) and the second field is labeled '签名' (Signature). Both fields are currently empty.

点击“保存”按钮

3. 更新table view  
界面

**步骤3.** 在**类1**中，实现table view中的三个协议方法

3.1 实现table view中返回行数的协议方法

3.2 实现table view中返回section的协议方法

3.3 实现table view中设置行的协议方法

## 功能二：删除好友记录

1. ContactTableViewController

2. AddContactViewController

点击“Edit”按钮



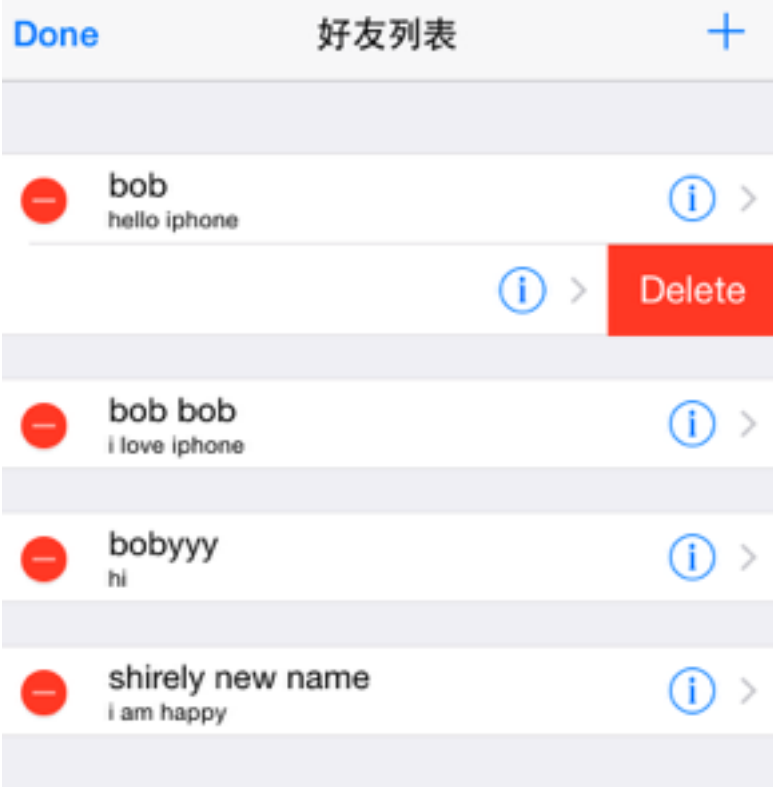
# 功能二：删除好友记录

1. ContactTableViewController



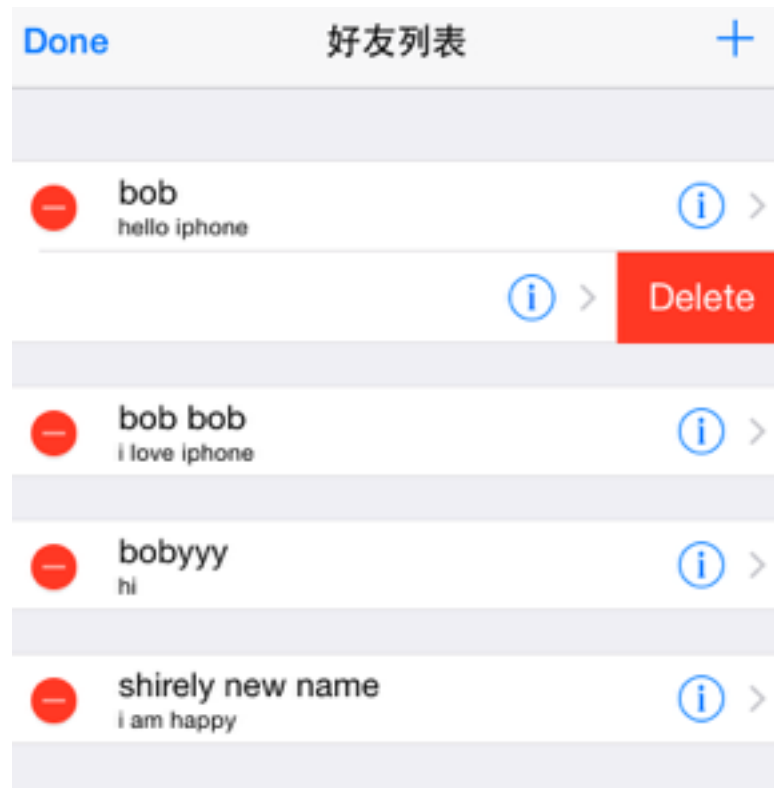
点击“-”按钮

1. ContactTableViewController



## 功能二：删除好友记录

1. ContactTableViewController



点击  
"Delete"

1. ContactTableViewController



sqlite

**步骤1:** 当点击“Delete”时，在**类1**中，

1.1 实现table view的一个和编辑行有关的协议方法

1.1.1 通过“结果控制器”来获取被管理对象上下文

1.1.2 通过被管理对象上下文删除indexPath位置的对象

1.2.3 通过被管理对象上下文调用save函数，将数据从数据库中删除

## 功能三：更新好友记录

### 1. ContactTableViewController

### 2. AddContactViewController



点击“i”按钮



- 步骤1:** 在table view编辑状态下，当点击“i”按钮时，在**类1**中，
- 1.1 实现table view的一个和点击accessory按钮有关的协议方法
    - 1.1.1 通过“结果控制器”来获取indexPath位置的Contact对象
    - 1.1.2 创建类2的对象，并使用自定义的初始化方法，将上面获取的对象传给类2
  - 1.2 从类1跳转到类2

## 功能三：更新好友记录

2. AddContactViewController

取消 Title 保存

名字

bob 1111

签名

hello iphone 1111

点击  
“保存”

1. ContactTableViewController

Done	好友列表	+
—	bob 1111 hello iphone 1111	(i) >
—	bob bob i love iphone	(i) >
—	bobyyy hi	(i) >
—	shirely new name i am happy	(i) >

**步骤2:** 当点击“保存”时，在**类2**中，

2.1 添加save触发方法的逻辑

2.1.1 通过传入的contact对象是否为空，来判定是否是从编辑状态过来的； 如果contact对象为空，做插入操作； 否则只做更新操作

sqlite