

iOS-CoreData

视频源代码交流网址

www.1000phone.net

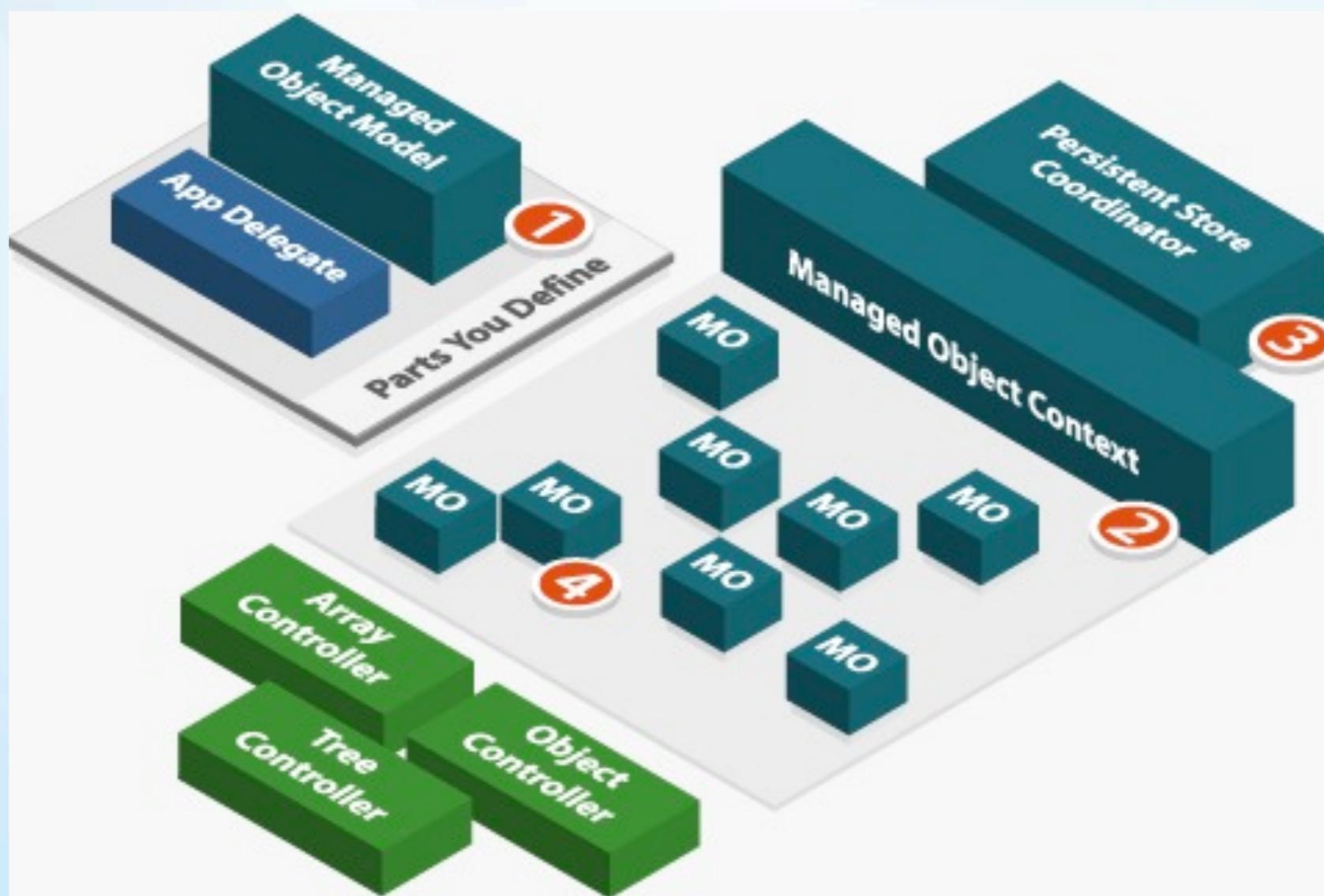
千里寻他众百度 锋自苦寒磨砺出

CoreData简介

- 1、iOS开发中实现数据持久化存储的一种重要手段。
- 2、CoreData提供了对象-关系映射的功能，能将Objective-c的对象使用数据库或者XML等方式存储。
- 3、如果使用CoreData将数据存到数据库中，不用通过sql语句来操作数据

千里寻他众百度 锋自苦寒磨砺出

CoreData-关系图



千里寻他众百度 锋自苦寒磨砺出

CoreData-核心类

- 1、**NSManagedObjectModel**, 用来加载 coreData 数据模型文件, 所有的数据模型可以全部加载到此对象中。
- 2、**NSManagedObjectContext**, 用于操作数据模型(对象), 并监测数据模型(对象)的变化。
- 3、**NSPersistentStoreCoordinator**, 数据持久化存储协调器, 负责调度上层与底层对数据的操作。
- 4、**NSManagedObject**, 具体的数据模型对象

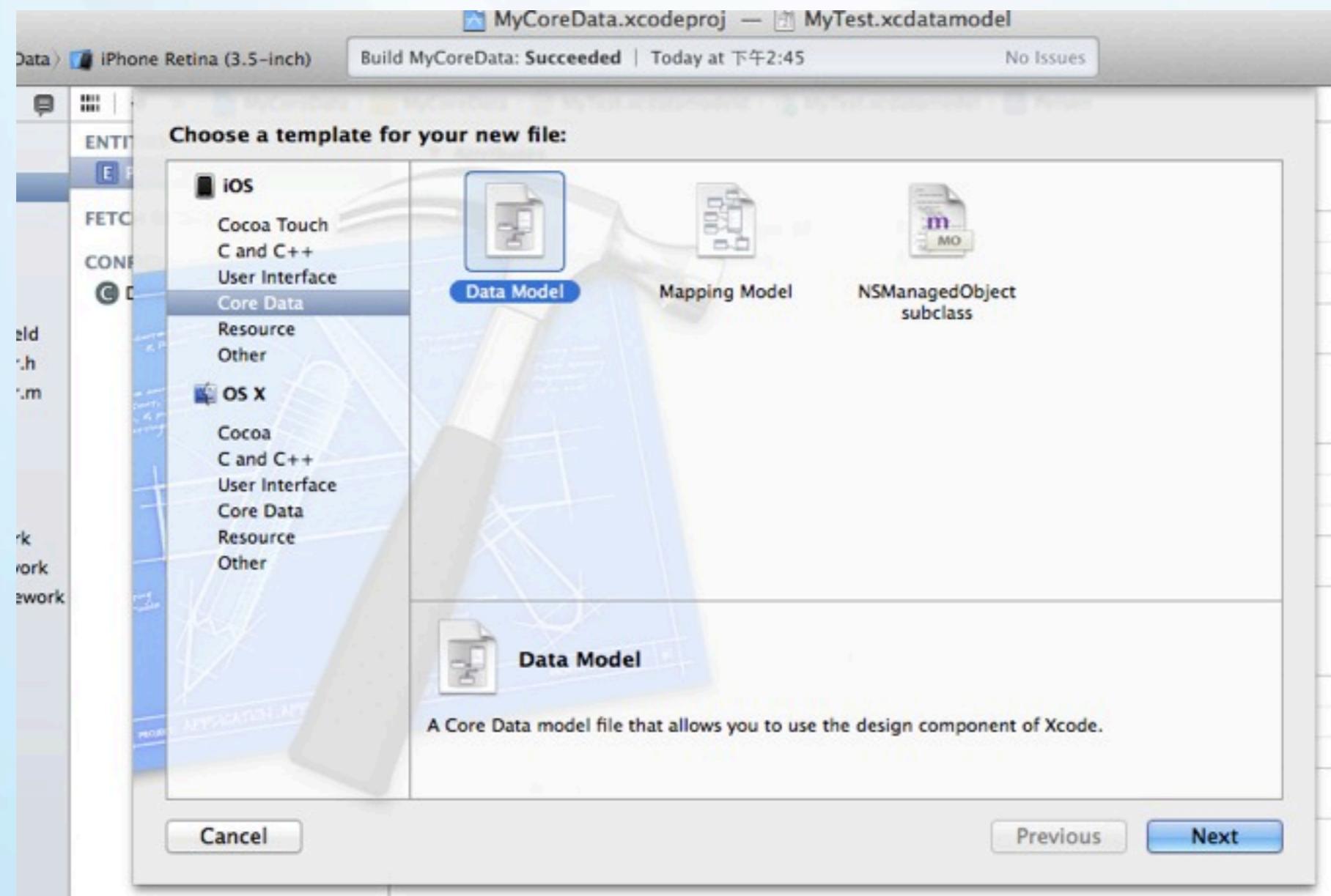
千里寻他众百度 锋自苦寒磨砺出

CoreData简介

- 5、**NSEntityDescription**, 模型描述类, 能够实例化得到具体的数据模型对象
- 6、**NSFetchRequest**, 数据查询请求类
- 7、**NSPredicate**, 通过谓词设置查询条件的类
(正则表达式)

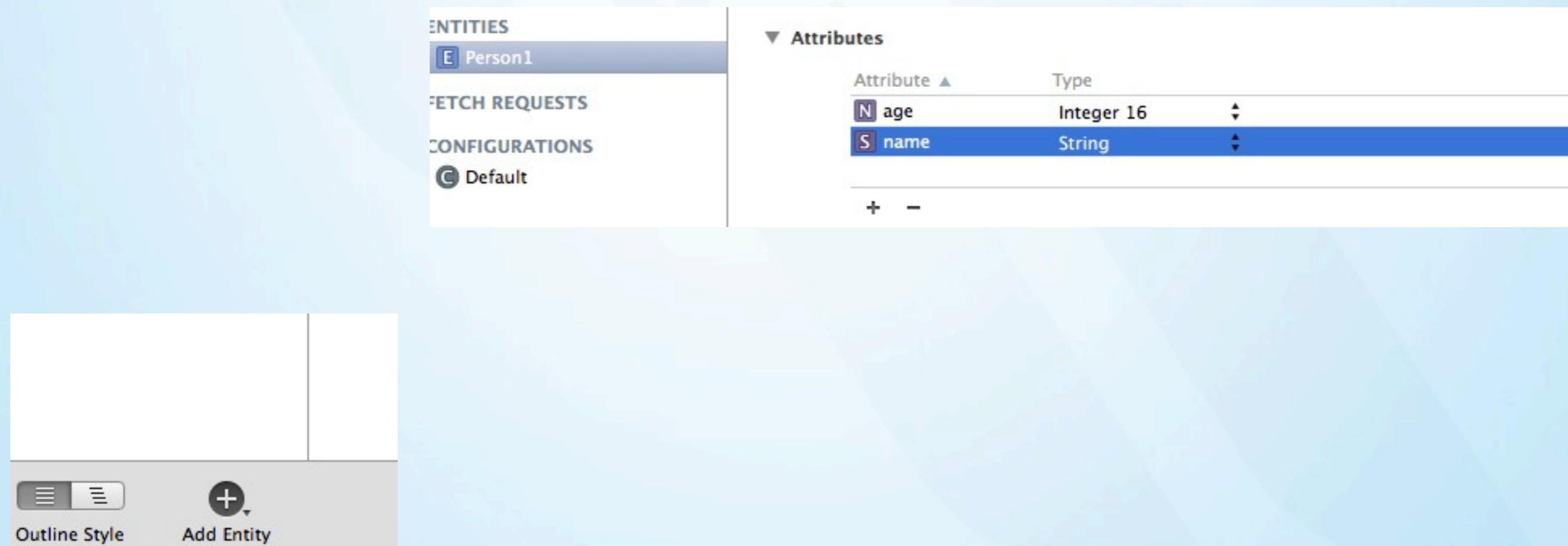
千里寻他众百度 锋自苦寒磨砺出

CoreData的使用：创建CoreData文件



千里寻他众百度 锋自苦寒磨砺出

CoreData的使用：添加CoreData模型



千里寻他众百度 锋自苦寒磨砺出

CoreData的使用：加载CoreData文件

```
//获取到coreData文件的路径，并转换成url
    NSString *coreDataPath = [[NSBundle mainBundle]
pathForResource:@"MyTest" ofType:@"momd"];
//加载coreData文件中的数据，加载文件中所有数据模型
    NSManagedObjectModel *model =
[[NSManagedObjectModel alloc] initWithContentsOfURL:
[NSURL fileURLWithPath:coreDataPath]];
```

千里寻他众百度 锋自苦寒磨砺出

将数据的存储方式设定为数据库

```
//初始化数据存储协调器
NSPersistentStoreCoordinator *pStroe =
[[NSPersistentStoreCoordinator alloc]
initWithManagedObjectModel:model];
NSError *error = nil;
//创建数据库的路径
NSString *path = [NSHomeDirectory()
stringByAppendingString:@"/Documents/test.sqlite"];
//将coreData文件数据映射到数据库
NSPersistentStore *store = [pStroe
addPersistentStoreWithType:NSSQLiteStoreType configuration:nil
URL:[NSURL fileURLWithPath:path] options:nil error:&error];
if (!store) {
    NSLog(@"error:%@",error.localizedDescription);
}
```

千里寻他众百度 锋自苦寒磨砺出

操作数据：添加

```
//创建操作数据的对象
NSManagedObjectContext *conntext = [[NSManagedObjectContext alloc] init];
//设置属性
conntext.persistentStoreCoordinator = pStroe;
/************添加数据*******/
//通过模型描述类，创建object对象
Person *myPerson = [NSEntityDescription
insertNewObjectForEntityForName:@"Person"
inManagedObjectContext:conntext];
myPerson.name = @"jiang";
myPerson.age = [NSNumber numberWithInteger:20];
NSError *insertError = nil;

Person *second = [NSEntityDescription
insertNewObjectForEntityForName:@"Person"
inManagedObjectContext:conntext];
second.name = @"test";
second.age = [NSNumber numberWithInteger:21];
//通过context的save方法保存数据,数据会保存到数据库
if (![conntext save:&insertError]) {
    NSLog(@"insert error:%@",error.localizedDescription);
}
```

千里寻他众百度 锋自苦寒磨砺出

操作数据：查询

```
//查询请求类，能查询到所有Person model的数据
NSFetchRequest *request = [NSFetchRequest fetchRequestWithEntityName:@"Person"];
NSError *fetchError;
NSArray *result = [context executeFetchRequest:request error:&fetchError];
 NSLog(@"person count:%d", result.count);
// NSLog(@"my result:%@", result);
for (Person *p in result) {
    NSLog(@"person:%@", p);
}
```

千里寻他众百度 锋自苦寒磨砺出

操作数据：修改

```
//根据查询条件，查询出想要更改的数据模型后，更改数据，再写回
    //设置查询条件
    NSPredicate *predicate = [NSPredicate
predicateWithFormat:@"name like %@", @"test"];
    request = [[NSFetchRequest alloc] init];
    //创建查询请求
    [request setEntity:[NSEntityDescription entityForName:@"Person"
inManagedObjectContext:conntext]];
    [request setPredicate:predicate];
    //得到查询结果
    NSArray *changeResult = [conntext executeFetchRequest:request
error:nil];
    for (Person *changeP in changeResult) {
        //修改
        changeP.name = @"superman";
    }
    //回写
    [conntext save:nil];
```

千里寻他众百度 锋自苦寒磨砺出

操作数据：删除

```
//删除
Person *last = [changeResult lastObject];
[conntext deleteObject:(NSManagedObject *)last];
//回写
[conntext save:nil];
```

千里寻他众百度 锋自苦寒磨砺出

谢谢大家！

- 千锋互联：www.1000phone.com
- 千锋3G学院：www.mobiletrain.org
- 千锋嵌入式学院：www.embedtrain.org

千里寻他众百度 锋自苦寒磨砺出