实验编号： 12 **四川师大《IOS》实验报告 2018** 年 **11** 月 **28** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 数据持久化 \_

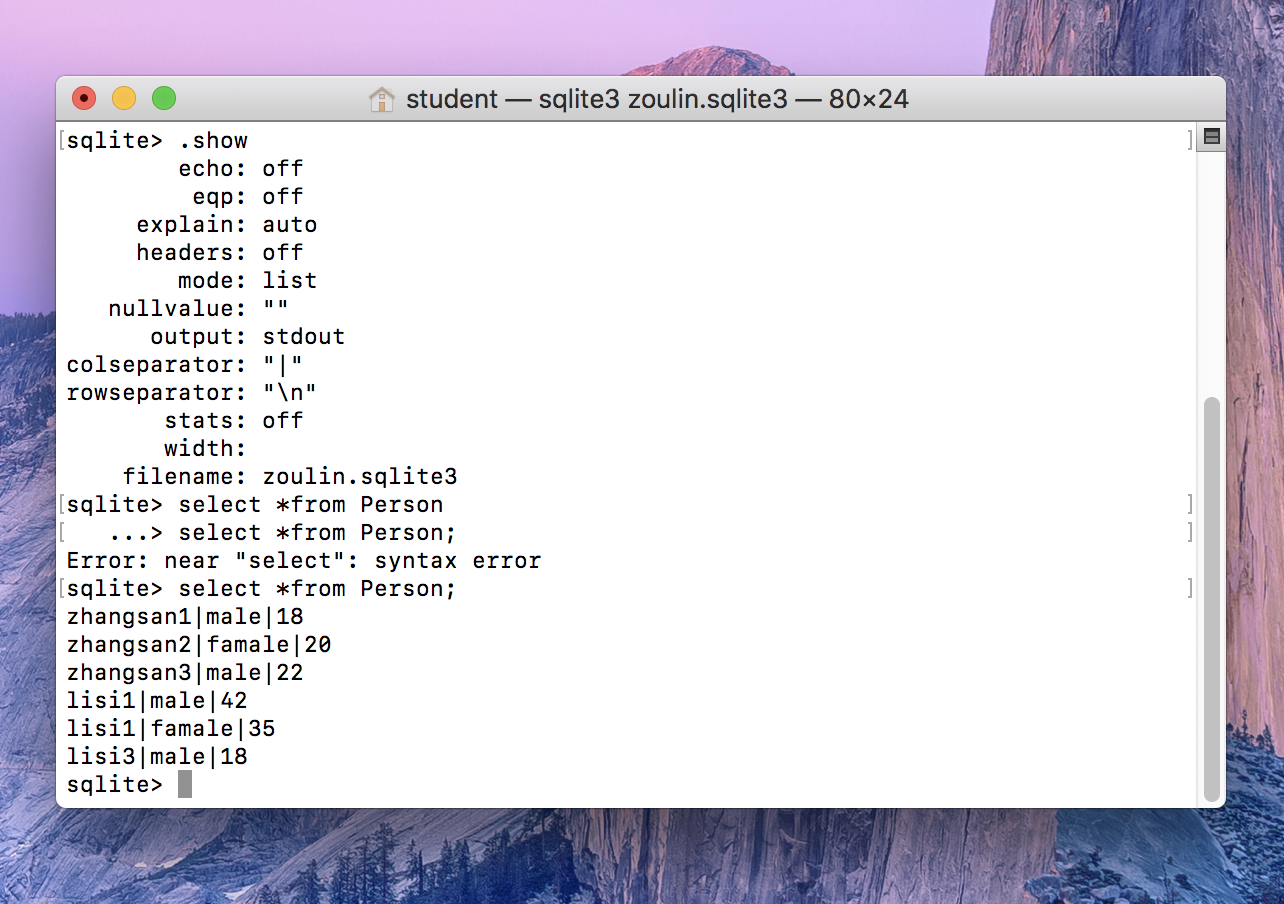
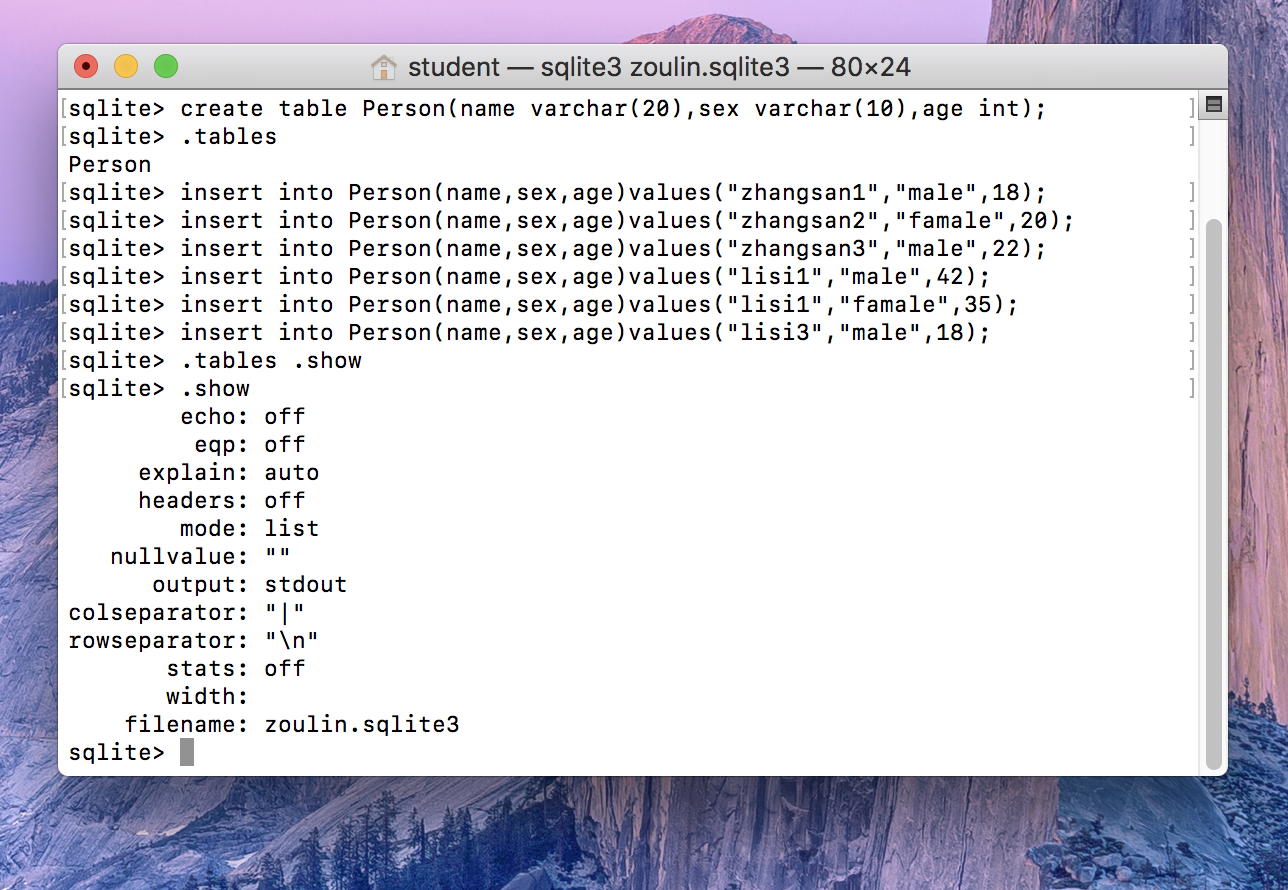
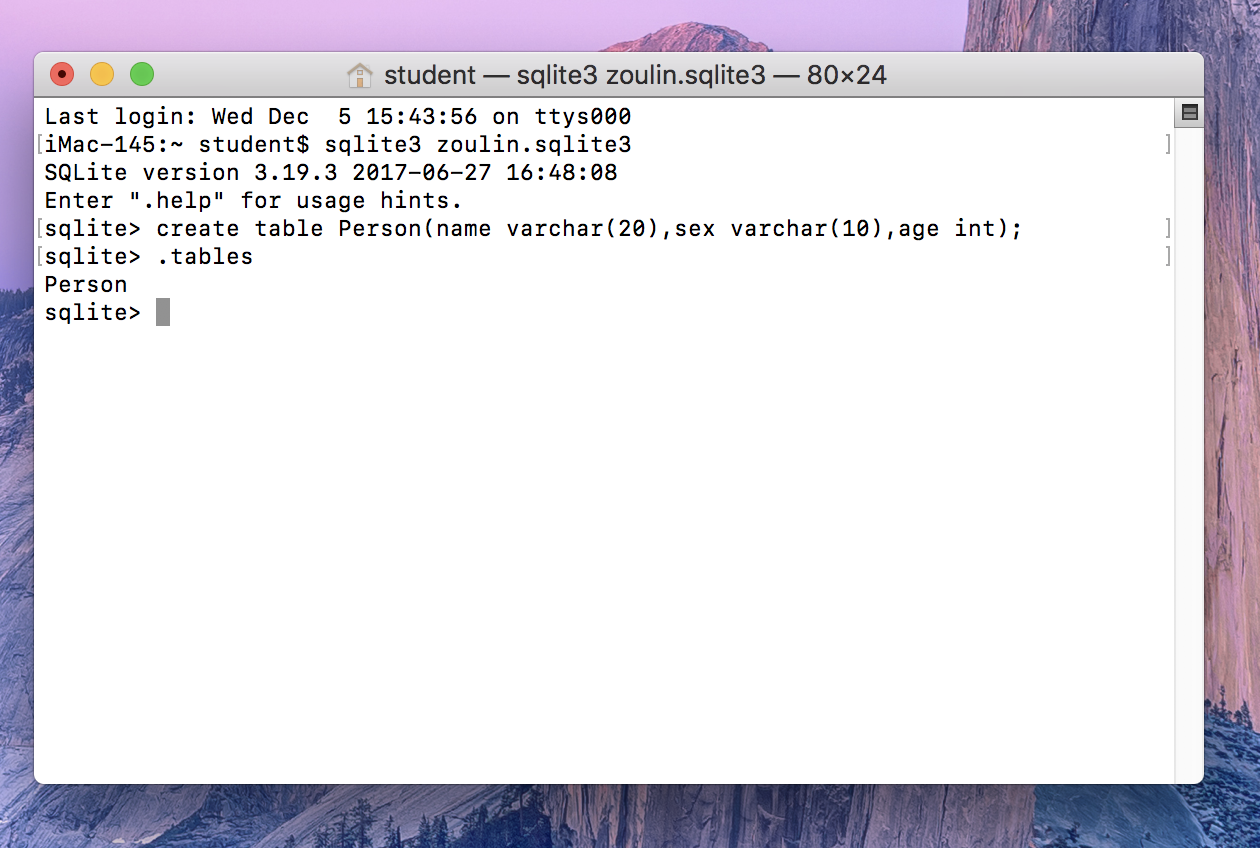
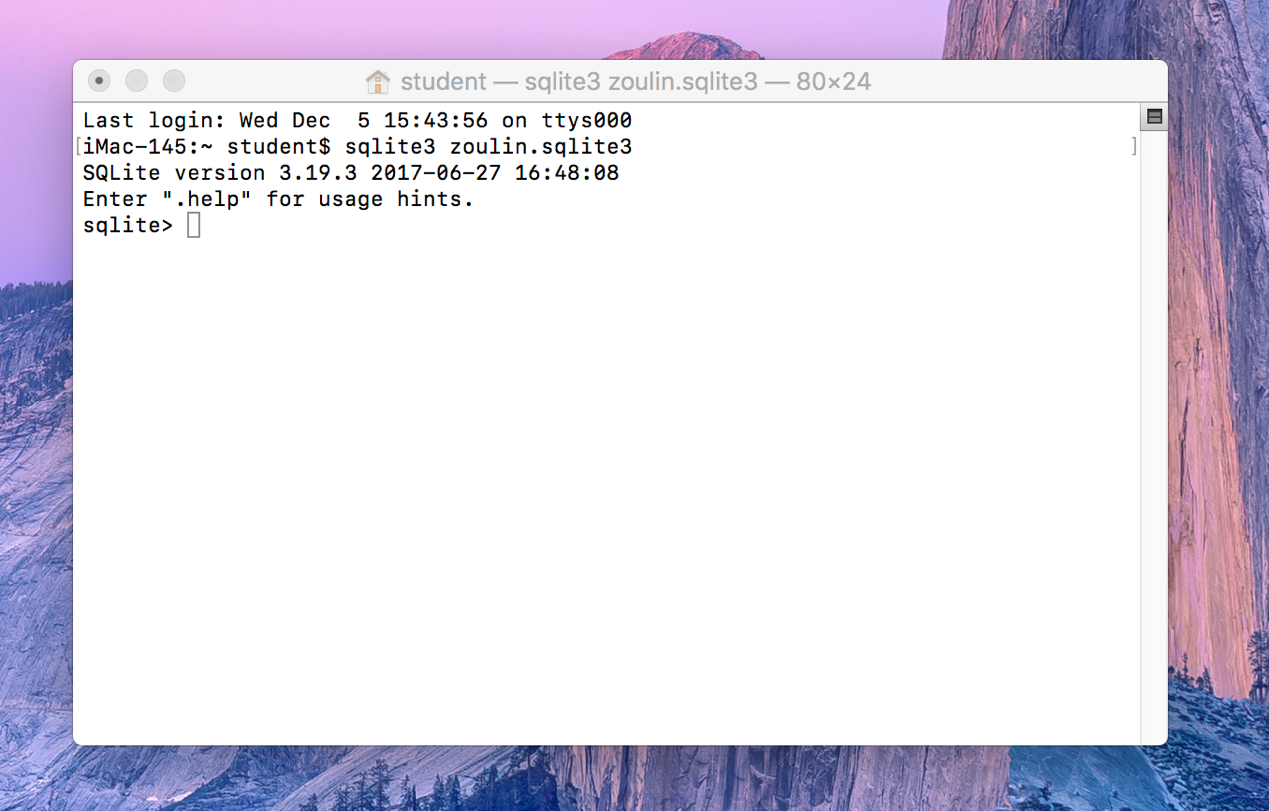
姓名：\_邹琳\_\_ 学号：\_2016110458\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 十二 \_\_\_\_\_\_**数据持久化**\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 掌握iOS中数据持久化的使用；
3. 掌握SQLite和CoreData的原理和使用；
4. 认真填写实验报告，要求附加部分运行界面和主要代码；
5. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
6. 实验内容
7. 在命令提示符下熟悉并使用SQLite命令
   1. 新建sqlite3数据库；
   2. 新建Person表；
   3. 插入新的数据；
   4. 查询数据；
8. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；
9. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
10. 在命令提示符下熟悉并使用SQLite命令
    1. 新建sqlite3数据库；
    2. 新建Person表；
    3. 插入新的数据；
    4. 查询数据；

程序代码：

运行结果：



1. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；

程序代码：

//

// ViewController.swift

// shiyan12-2

//

// Created by student on 2018/12/8.

// Copyright © 2018年 zoulin. All rights reserved.

//

import UIKit

import CoreData

class ViewController: UIViewController {

@IBOutlet weak var tfName: UITextField!

@IBOutlet weak var tfSex: UITextField!

@IBOutlet weak var tfAge: UITextField!

@IBOutlet weak var tfPhone: UITextField!

//设置代理,强转为自己的代理对象

let appDelegate = UIApplication.shared.delegate as! AppDelegate

//保留上下文

let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext

override func viewDidLoad() {

super.viewDidLoad()

print(NSHomeDirectory())

// Do any additional setup after loading the view, typically from a nib.

}

//添加

@IBAction func addClick(\_ sender: Any) {

let person = Person(context:context)

person.name = tfName.text

person.sex = tfSex.text

person.age = tfAge.text

person.phone = tfPhone.text

//保存

appDelegate.saveContext()

}

//更新，需要先完成查询

@IBAction func update(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

p.phone = tfPhone.text

p.sex = tfSex.text

p.age = tfAge.text

//修改完成后需要保存

// appDelegate.saveContext()

}

}

//删除,需要先完成查询

@IBAction func deleteClick(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

context.delete(p)

//删除完成后需要保存

// appDelegate.saveContext()

}

}

//查询

@IBAction func queryClick(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

tfPhone.text = p.phone

tfSex.text = p.sex

tfAge.text = p.age

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*多mvc\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//

// ViewController.swift

// shiyan12-2

//

// Created by student on 2018/12/8.

// Copyright © 2018年 zoulin. All rights reserved.

//

import UIKit

import CoreData

class ViewController: UIViewController {

@IBOutlet weak var tfName: UITextField!

@IBOutlet weak var tfSex: UITextField!

@IBOutlet weak var tfAge: UITextField!

@IBOutlet weak var tfPhone: UITextField!

//设置代理,强转为自己的代理对象

let appDelegate = UIApplication.shared.delegate as! AppDelegate

//保留上下文

let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext

override func viewDidLoad() {

super.viewDidLoad()

print(NSHomeDirectory())

// Do any additional setup after loading the view, typically from a nib.

}

//添加

@IBAction func addClick(\_ sender: Any) {

let person = Person(context:context)

person.name = tfName.text

person.sex = tfSex.text

person.age = tfAge.text

person.phone = tfPhone.text

//保存

appDelegate.saveContext()

}

//更新，需要先完成查询

@IBAction func update(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

p.phone = tfPhone.text

p.sex = tfSex.text

p.age = tfAge.text

//修改完成后需要保存

// appDelegate.saveContext()

}

}

//删除,需要先完成查询

@IBAction func deleteClick(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

context.delete(p)

//删除完成后需要保存

// appDelegate.saveContext()

}

}

//查询

@IBAction func queryClick(\_ sender: Any) {

//先构造一个请求对象

let fetch:NSFetchRequest<Person> = Person.fetchRequest()

//断言

fetch.predicate = NSPredicate(format: "name=%@", tfName.text!)

let persons = try? context.fetch(fetch)

if let p = persons?.first {

tfPhone.text = p.phone

tfSex.text = p.sex

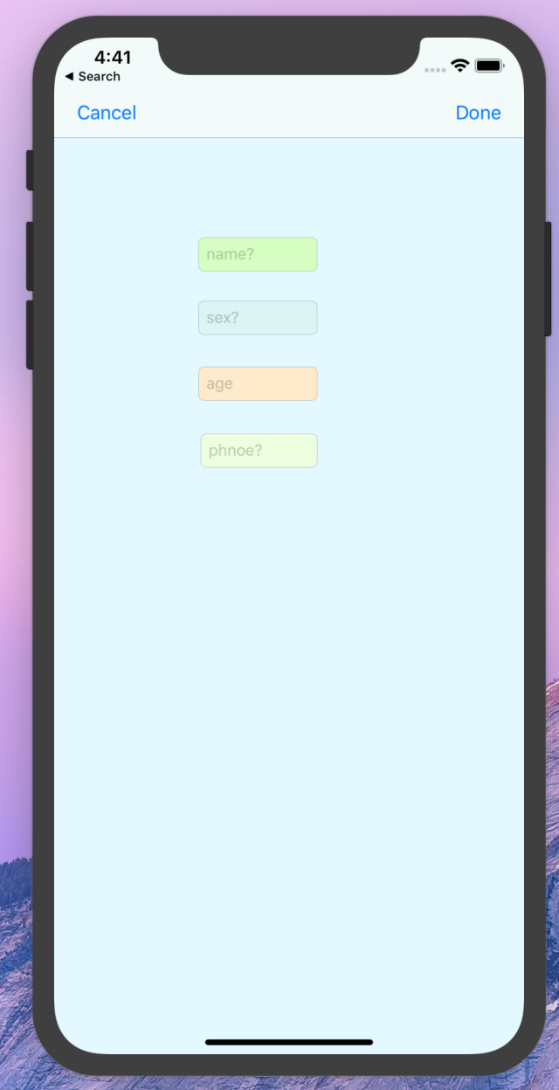
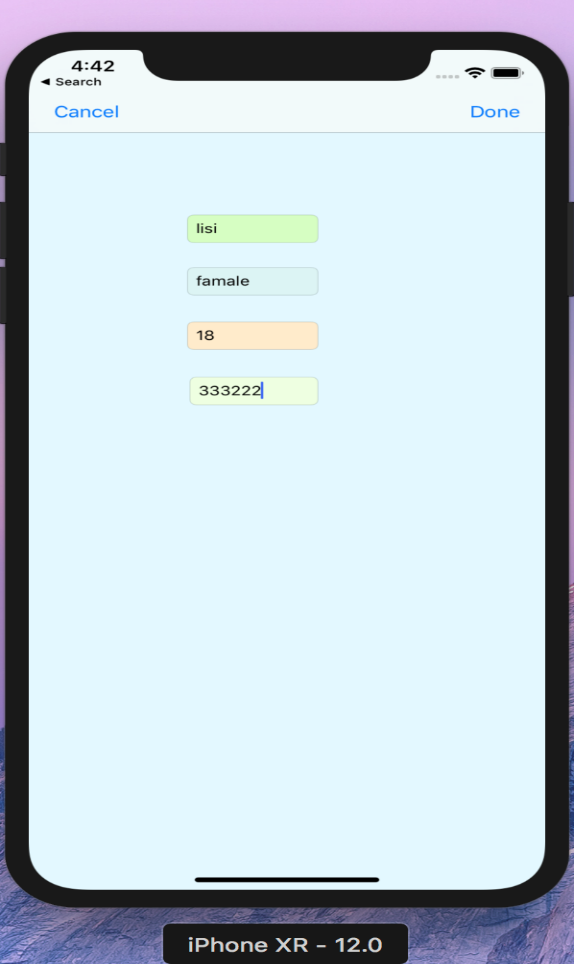
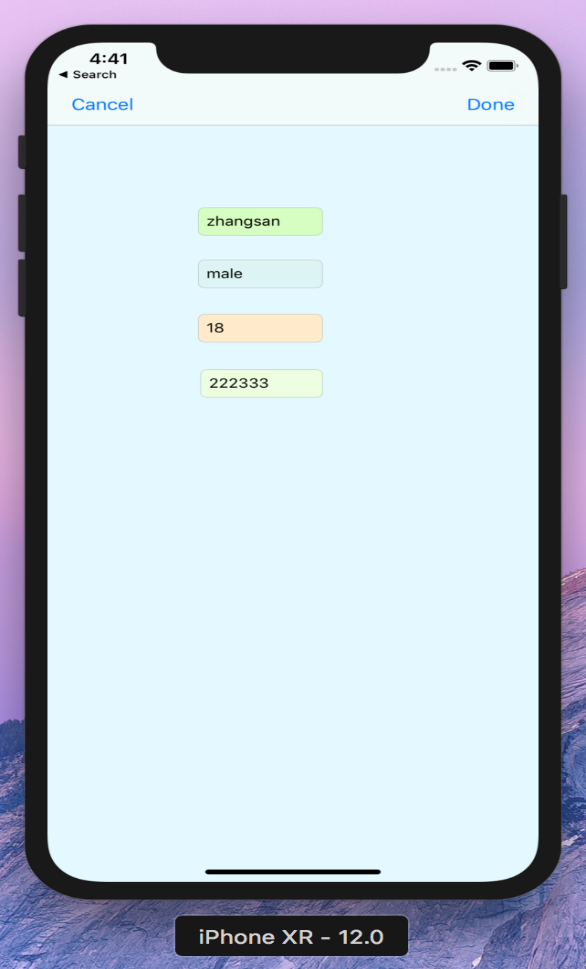
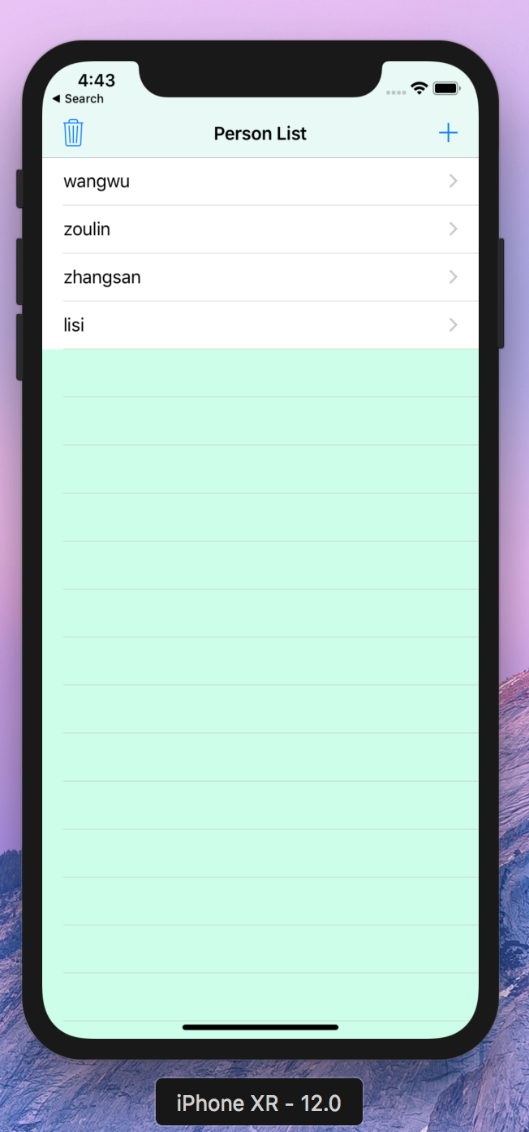
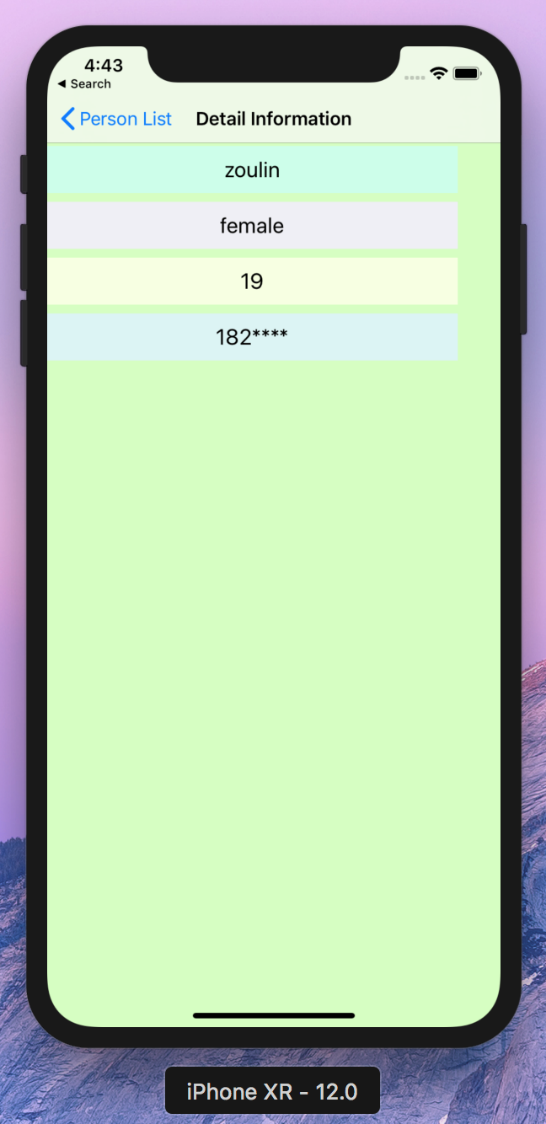
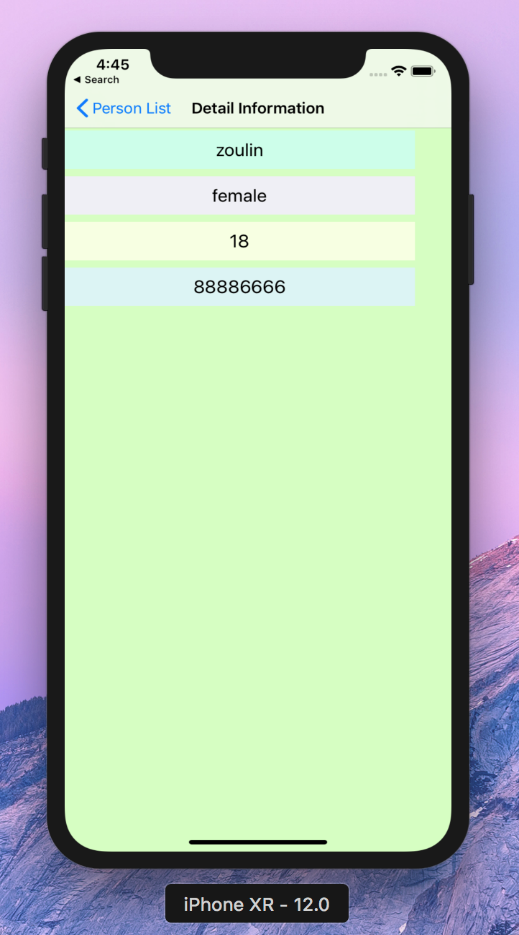
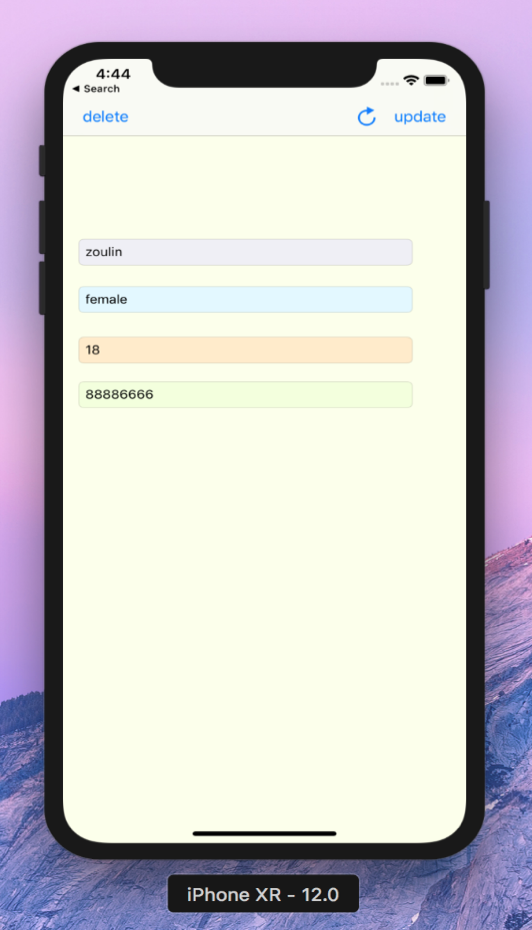
tfAge.text = p.age

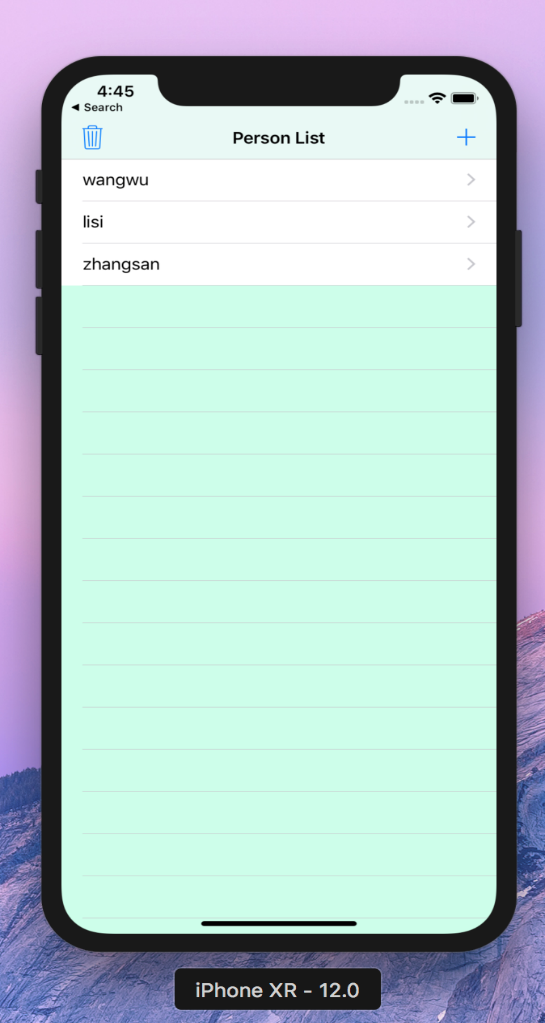
}

}

}

运行结果：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：<https://github.com/Paass12/linzou.git>

这次实验主要考察了数据库的基本实现。了解了三种方法，掌握了两种方法。

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。