《Python语言程序设计》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 马昆 | 班级 | 2206831 | | | 学号 | 2206831544 |
| 实验名称 | 实验八 **Python数据库应用** | | | | | | |
| 实验时间 | 2023年 06 月 02 日 | | | 成绩 |  | | |
| **实验目的：**   1. 掌握如何访问SQLite数据库 2. 了解访问其他关系型数据库的方法   **实验环境：**  1、硬件环境：联想拯救者Y7000P-2022。  2、软件环境：  1)软件：Anaconda,spyder  2)操作系统：Windows 11 专业工作站版  **实验内容：**  编程练习：  1、设计一个用户注册信息管理系统。系统使用SQLite数据库存储注册用户数据，可以从数据库中读出数据进行查看、修改、删除和添加用户信息。  **实验过程及结果记录：**   1. 编程练习：   设计一个用户注册信息管理系统。系统使用SQLite数据库存储注册用户数据，可以从数据库中读出数据进行查看、修改、删除和添加用户信息。   |  | | --- | | 代码  """  查看、修改、删除和添加用户信息  """  import pymysql as mysql  # 用户类  class User:  # 构造方法  def \_\_init\_\_(self, id, name, username, password, gender, phone):  self.id = id  self.name = name  self.username = username  self.password = password  self.gender = gender  self.phone = phone  # toString 方法  def \_\_str\_\_(self):  return "编号：%d，姓名：%s，用户名：%s，密码：%s，性别：%s，手机号：%s"\  %(self.id,self.name,self.username,self.password,self.gender,self.phone)    # 获取连接对象  def get\_conn():  conn = mysql.connect(host = '127.0.0.1' # 连接名称  ,user = 'manager' # 用户名  ,passwd = 'mk20010624' # 密码  ,port = 3306 # 端口，默认为3306  ,db = 'test' # 数据库名称  ,charset = 'utf8' # 字符编码  )  return conn  # 查询所有用户  def queryAll():  conn = get\_conn()  cur = conn.cursor() # 生成游标对象  sql = "select \* from `admin`" # SQL语句  cur.execute(sql) # 执行SQL语句  datas = cur.fetchall() # 通过fetchall方法获得数据  user\_list = []  for data in datas:  user = User(id = data[0], name = data[1], username = data[2], password = data[3], gender = data[4], phone = data[5])  user\_list.append(user)  cur.close() # 关闭游标  conn.close() # 关闭连接  return user\_list  # 添加用户  def add(user):  conn = get\_conn()  cur = conn.cursor() # 生成游标对象  sql = "insert into `admin` (name, username, password, gender, phone) values ('%s','%s','%s','%s','%s')"\  % (user.name, user.username, user.password, user.gender, user.phone)  cur.execute(sql)  conn.commit() # 提交  cur.close() # 关闭游标  conn.close() # 关闭连接    # 更新用户  def edit(user):  conn = get\_conn()  cur = conn.cursor() # 生成游标对象  sql = "update `admin` set name = '%s', username = '%s', password = '%s', gender = '%s', phone = '%s' where id = %d"\  % (user.name, user.username, user.password, user.gender, user.phone, user.id)  cur.execute(sql)  conn.commit() # 提交  cur.close() # 关闭游标  conn.close() # 关闭连接  # 删除用户  def delete(id):  conn = get\_conn()  cur = conn.cursor() # 生成游标对象  sql = "delete from `admin` where id = %d" % id  cur.execute(sql)  conn.commit() # 提交  cur.close() # 关闭游标  conn.close() # 关闭连接    # 启动系统函数  def system():  while True:  print("---------------------------------")  print("1、查看所有用户")  print("2、增加一个用户")  print("3、修改一个用户")  print("4、删除一个用户")  print("0、退出系统")  user\_in = input("请输入对应服务的编号：")  match user\_in:  case "1":  # 查询所有用户  user\_list = queryAll()  for user in user\_list:  print(user)  case "2":  # 添加用户  user = User(id = 0,name = input("请输入姓名："), \  username = input("请输入用户名："), \  password = input("请输入密码："), \  gender = input("请输入性别："), \  phone = input("请输入手机号："))  add(user)  case "3":  # 修改用户  user = User(id = int(input("请输入编号：")),  name = input("请输入姓名："), \  username = input("请输入用户名："), \  password = input("请输入密码："), \  gender = input("请输入性别："), \  phone = input("请输入手机号："))  edit(user)  case "4":  delete(int(input("请输入编号：")))  case "0":  print("欢迎下次使用，再见~")  break  case \_:  print("未知选项，请重新输入")      # 启动系统  system() | | 运行结果1    运行结果2    运行结果3    运行结果4    运行结果5    运行结果6 | | | | | | | | |
| **实验总结：**  通过本次实验，复习了一些之前学到的知识，例如程序的流程控制语句、面向对象编程、函数式编程、以及SQL语句相关的知识，并且还新学了一个新的流程控制语句，类似于其他编程语言的switch-case语句，是Python 3.10 新增的 match-case语句。使用起来比用if-else方便多了，本次实验是要求使用sqlite数据库，而我们课堂上教学使用的是Mysql数据库，所以本次实验使用了Mysql作为了数据库，也学习了第三方库pymysql的基本使用，能够实现简单的增删改查，并且使用这些知识完成了一个控制台应用，做了一个能够持久化存储的小程序，非常有成就感。 | | | | | | | |
|  | | | | | | | |