### 3.3 数据与系统

本节内容按4课时设计。

**第一课时**

**【**教学重点**】**了解tkinter模块，编写简单可交互的GUI程序。

**【**教学难点**】**掌握数据传递过程（GUI程序界面诸对象及文本文件）。

**【教学过程】**

**一、引入**

1.学生预习，阅读第67页任务一 探究“旅行线路”之“活动1 初看“旅行线路”，了解“旅行线路”程序界面及功能，了解图形用户界面（GUI）。完成填空，以实现在窗口上添加一个“查询线路”按钮的功能。

2.上机操作：运行程序，观看运行结果，对照理解每个语句的作用

from tkinter import \* #导入tkinter模块

root = Tk() #创建一个窗口

root.title("旅行线路") #设置窗口标题

root.geometry('300x120') #设置窗口大小

root.resizable(0,0) #禁止调整窗口大小

#的窗口上建一个“添加线路”按钮

Button(root, text="添加线路", relief="solid",width=10,height=2).pack()

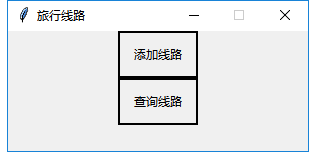
#在屏幕上建一个Button按钮。



为实现在窗口上添加一个“查询线路”按钮的功能，应该加上一条语句：

Button(root, text="查询线路", relief="solid",width=10,height=2). pack()

运行检验效果：**详见配套资源“旅行线路.py”。**



老师检查学生完成情况。可视化编程中，引用模块比较陌生，设计环节多，相互关联多，出错可能性大增。教师要发现问题，及时解决，破除学生因屡屡失败而产生的畏难情绪。

**二、图形用户界面（GUI）**

图形用户界面（Graphical User Interface，GUI）又称图形用户接口，是指采用图形方式显示的计算机操作用户界面。在Python 语言中，tkinter是实现图形用户界面的模块，可以快速创建GUI应用程序。我们需要先导入模块，上面所示的代码用“from tkinter import \*”导入tkinter模块，然后就可以运用相应代码创建矩形容器（Frame）、文本框（Entry）、命令按钮（Button）、文本标签（Label）等交互对象。

在现代程序设计中，我们用“对象”建立计算机世界的实体与客观世界的实体相一致的映射关系，问题的解决方案可以用一些彼此交互的对象来形象化表示。比如在“旅行线路”例子中，我们利用Python提供的按钮类创建按钮对象，通过单击按钮的交互方式完成代码的执行过程。

**三、编程实现“添加线路”功能（活动2）**

**（详见配套资源“添加线路.py”）**

from tkinter import \*

root = Tk() #创建一个窗口

root.title("添加线路") #设置窗口标题

root.geometry('600x100') #设置窗口大小

root.resizable(0,0) #禁止调整窗口大小

var=StringVar() #定义StringVar()类型

def intomap(): #Button按钮激发函数

c=open("旅行线路.txt",'a+') #以追加模式打开文件

c.write(var.get()+"\n") #在text文件末尾添加text里的内容

c.close #关闭文件

#在窗口上建一个文本标签

Label(root, text='请输入线路', font=('Arial', 10)).pack()

#在窗口上建一个文本框

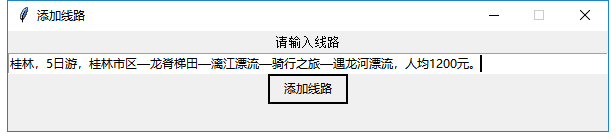
Entry(root,textvariable=var,width=550).pack()

#在窗口上建一个Button按钮

Button(root, text="添加线路", command =intomap,relief="solid",width=10).pack()

root.mainloop()

运行效果如下：



**四、文件与Python读写文本文件**

师：文件是存储在外存储设备中的相关数据的集合。在解决问题的过程中，我们可以根据需要将数据组织在一起形成队列、树或图等逻辑结构，这时的数据存储在内存中，当退出程序时，相关数据也将丢失。如果需要更持久地存储数据，我们可以将数据以文件的形式存储到外存储设备中。在活动2中，“线路数据”是用文件的形式存储在计算机的磁盘里的。

师：在Python语言中，用open函数把外存储器里的文本文件（.txt）打开，用write方法将内存中的数据存储到外存储器的文件中，文件操作结束时一定要用close方法关闭文件。

Python还提供了读写电子文档和电子表格等多种类型文件的模块，可以很方便地建立起不同计算工具间的联系。

在数据管理技术的发展历程中，文件的使用有助于长期保存数据，并且实现由专门的软件对数据进行管理，从而减轻人工管理数据的负担。

数据的存储传统上是使用单独的没有关联的文件。比如在活动2 中，程序“添加线路.py”使用文件“旅行线路.txt”。现在，我们也可以将一些文件建立关联，形成数据库文件。

**五、小结**

1. tkinter模块的基本操作：引用，创建窗体，窗体大小，标题内容；在窗体上创建标签，文本框，按钮；为按钮单击事件编写代码。

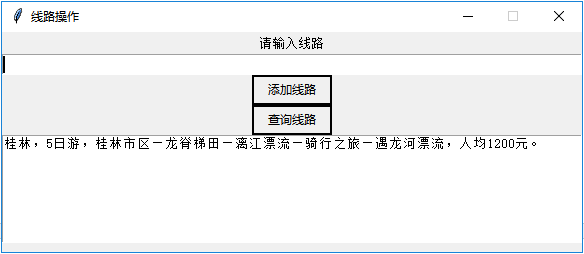
2.对文本文件的操作：将内容写入文本文件。

**六、练习**

【扩展活动2例程】活动2只有一个“添加线路”的功能，请在此例程基础上，增加一个“查询线路”的按钮，并实现功能：单击该按钮，在文本框中显示文本文件"旅行线路.txt"中的已有全部内容。

【附】参考程序界面如下**（详见配套资源“添加线路与查询线路.py”）**

：



**第二课时**

**【**教学重点**】**Access数据库简单操作 。

**【**教学难点**】**Access数据库简单操作。

**【教学过程】**

**一、引入**

WPS表格数据操作：学生预习，阅读第69页“任务二 走近火车票订票系统”之“活动1 查找车次”，查找厦门—成都之间通过武汉（含武昌、汉口）转乘信息，厦门至武汉全部车次信息（9个车次），武汉至成都全部车次信息（24个车次）。

将网上信息复制到WPS表格，整理、存储相关数据，填写图3.3.3。

**（详见配套资源“厦门至成都车次信息.xlsx”）**

**二、数据库**

【做一做】学生用Access软件对数据表进行简单数据操作：增加、修改、删除、保存。

在教科书配套资源中找到并打开“厦门至成都.mdb”数据库文件，将“厦门至武汉车次”和“武汉至成都车次”这两个表的数据补充完整，所需数据从图3.3.3中取得。

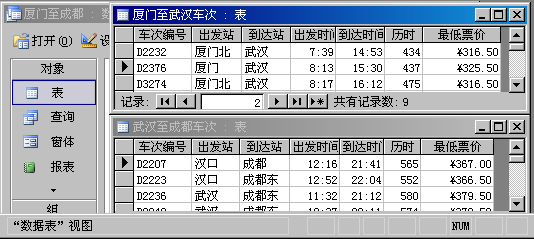


图3.3.4 Access管理数据库界面

数据库是以一定的组织方式存储在计算机中的相互关联的数据集合。

1.数据表

在数据库中，数据是以二维表的形式组织存储的，称为表。如图3.3.4所示的 “武汉至成都车次”表，表中的一列称为一个字段，一个字段对应车次实体的一个属性，每个字段都有一个名字，称为字段名，如车次编号、出发站等。表中的一行称为一条记录，一条记录对应于一个车次实体，保存有关实体属性的数据。可以唯一确定一条记录的字段称为主键，如车次编号。若干个车次数据的集合就组成了“武汉至成都车次”表。我们可以对数据表进行插入、删除、更新等操作。

2.数据库管理系统

在数据库技术中，数据是如何被科学地组织和存储的？又是如何被高效地获取和维护的？完成这些任务的就是数据库管理系统。数据库管理系统是定义、创建、维护数据库的一种工具，它允许用户管理和控制数据库中的数据。常见的数据库管理系统有Sybase、ORACLE、MySQL、Access等。

**三、小结**

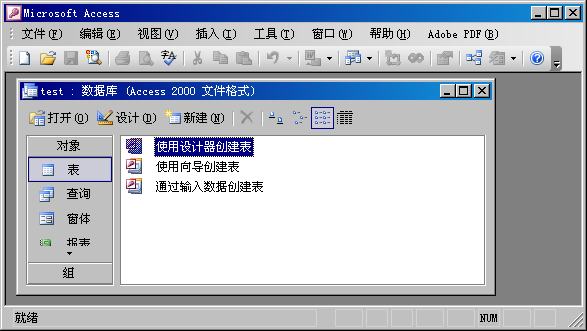
1.体会WPS表格操作与Access数据库的表操作的异同，以便将基本概念（字段、记录）和操作（追加、插入、修改、删除），顺利实现从电子表格向数据库牵移。

2.利用数据库管理系统Access，创建数据库文件（扩展名为\*.mdb）,在数据库文件中创建数据表。通过切身操作，理顺三者之间的关系，建立起相应概念。

**四、练习**

【创建Access数据库及表】

1.启动Access软件，新建一空白数据库文件，保存到桌面，取名为test.mdb。



2.双击创建表结构，按下表依次创建7个字段，并设置数据类型和范围。

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **范围** |
| 车次编号 | 文本 | 字段大小:10 |
| 出发站 | 文本 | 字段大小:20 |
| 到达站 | 文本 | 字段大小:20 |
| 出发时间 | 日期/时间 | 格式:短时间 |
| 到达时间 | 日期/时间 | 格式:短时间 |
| 历时 | 数字 | 字段大小:整型 |
| 最低票价 | 货币 | 格式:货币 |

3.设置完成后保存数据表，表名取为“火车车次信息表”，并定义“车次编号”为主键。

4.登陆中国铁路12306订票官网，查找后天从北京到上海的所有火车信息，选择前3条记录录入或复制到表中。



**第三课时**

**【**教学重点**】**Access表间关系及查询。

**【**教学难点**】**Access表间关系及查询。

**【教学过程】**

**一、引入**

学生预习，阅读第71页“任务三 探究订票系统功能”之“活动1 查询换乘方案”，尝试在Access中建立表之间的“关系”，利用“关系”解决问题。

在查找换乘方案时，“厦门至武汉车次”表中的字段“到达站” 与“武汉至成都车次”表中的“出发站”是有对应关系的。请你打开教科书配套资源中的“厦门至成都.mdb”文件，利用“关系”菜单创建如图3.3.5所示的表间关系：

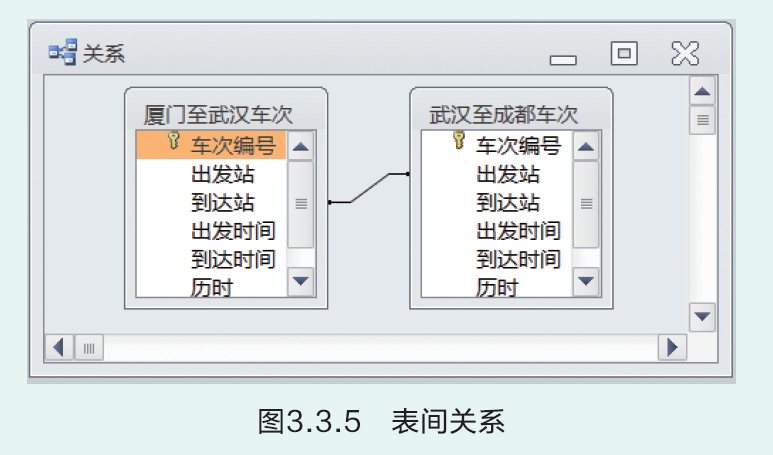
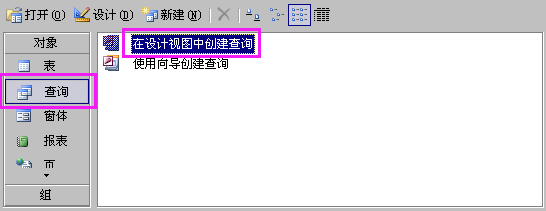


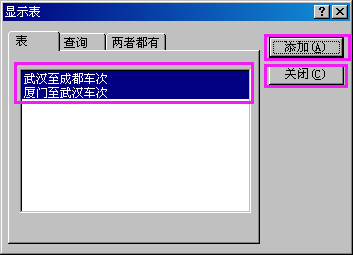
图3.3.5 表间关系

**二、表间关联与“查询设计”**

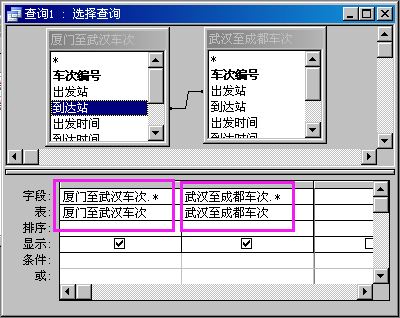
关系就像绳子将两个“表”联结在一起，我们可以利用“查询”面板中的“查询设计”选项按照提示创建选择查询。



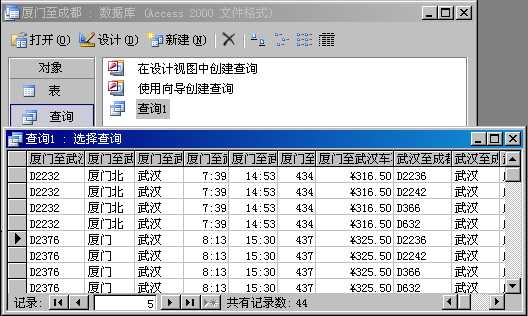
双击，弹出对话框。



选中2个表，单击“添加”按钮，再单击“关闭”按钮。



再分别选中2个表的全部字段，保存为“查询1”。



双击“查询1”，即可看到查询结果（一个新的表），共有44条记录。可以看出，查询结果正是两表中的数据通过关系（厦门至武汉车次.到达站=武汉至成都车次.出发站）进行关联的结果。

该结果有两大问题：

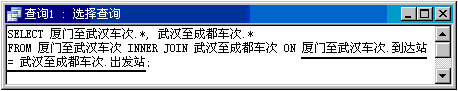
一是时间上不合理，没有考虑换乘时间的合理性。例如：在图3.3.6第1条记录中，从厦门北到达武汉的时间是14:53，而从武汉前往成都东的出发时间是11:32，时间错位。

二是中转地点不合理。表“厦门至武汉车次”与表“武汉至成都车次”中录入的都是以武汉（含汉口、武昌、武汉三个车站）为中转站的车次数据，武汉三站之间地铁换乘，也仅需半小时，因此，以“厦门至武汉车次.到达站=武汉至成都车次.出发站”为关系建立的查询是不可取的，它将许多可能合理的换乘方案排除了，如坐D2232从厦门北到武汉站，然后坐武汉市内地铁到武昌站，再换乘T246次从武昌到成都东，这条可行方案被排除在外了。

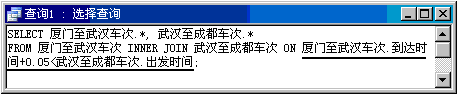
改进：将关系（厦门至武汉车次.到达站=武汉至成都车次.出发站）修改为关系（厦门至武汉车次.到达时间+0.05 < 武汉至成都车次.出发时间），式中0.05单位为天，即0.05\*24小时=1.2小时，预留了武汉三站之间的换乘时间及出站、进站时间。



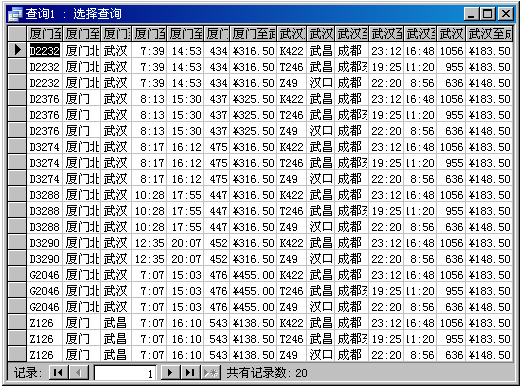
打开“查询1”的设计视图，在空白处点鼠标右键，单击“SQL视图”：



直接在里面修改SQL查询语句为：



保存修改退出查询设计视图，双击“查询1”，结果如下：



共有20条记录。这个结果基本上都可作为换乘方案。

活动2 满足个性化需求

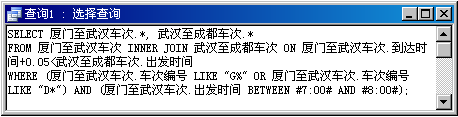
在12306网站中有出发时间段和车型等选择项，可以满足不同旅客的出行要求。同样，也可以在Access中设置查询条件，解决个性化需求的问题。在查找换乘方案时，可以在“查询设计器”窗口的“条件”栏中设置查询的准则。

**三、查询准则**

如果只是简单地查找某个字段为特定值的记录，只要在该字段对应的“条件”栏中输入该值即可；如果不仅是查找某个特定值，在准则中可以使用表达式。在表达式中可通过操作符设置查询范围。

某同学想乘坐高铁或动车出行，他还希望乘坐7:00～8:00的火车出发，只需在查询表达式中追加条件：

WHERE (厦门至武汉车次.车次编号 LIKE "G%" OR 厦门至武汉车次.车次编号 LIKE "D\*") AND (厦门至武汉车次.出发时间 BETWEEN #7:00# AND #8:00#);



查询执行结果如下：



图3.3.7 体验个性化出行

数据库的建立、使用和维护等工作仅靠数据库管理系统还远远不够，还要有专门的人员来完成，这些人被称为数据库管理员。数据库管理员通过数据库管理系统可以对数据库进行操作和维护。但对于广大普通用户来说，很难像专业的数据库管理员那样操作数据库，而具有良好交互性的数据库应用系统可以满足人们简单、方便地使用数据的需求。

**四、数据库系统**

数据库系统是由数据库、数据库管理系统（及其应用开发工具）、数据库应用系统、数据库管理员和用户组成的存储、管理、处理和维护数据的系统。数据库主要用于存储数据，需要足够大的内存储器和外存储器等硬件平台的支持。软件部分除了操作系统和数据库管理系统外， 还包括支持特定应用环境开发的软件工具以及开发完成的数据库应用系统。12306网站是一个大型数据库应用系统，其他如各类财务管理系统、人事管理系统、图书管理系统等软件也都是数据库应用系统。

**五、小结**

1.数据表间关系，控制着多表联合查询时数据之间基本的组织方式。

2.SQL语句是查询的“硬核”，其常见格式参考如下：

SELECT 厦门至武汉车次.\*, 武汉至成都车次.\*

FROM 厦门至武汉车次 INNER JOIN 武汉至成都车次 ON 厦门至武汉车次.到达时间+0.05<武汉至成都车次.出发时间

WHERE (厦门至武汉车次.车次编号 LIKE "G%" OR 厦门至武汉车次.车次编号 LIKE "D\*") AND (厦门至武汉车次.出发时间 BETWEEN #7:00# AND #8:00#);

**六、练习**

请在“体验个性化出行”查询基础上，增加一列，显示旅行总票价。效果如下：



【附】查询表达式如下：

SELECT 厦门至武汉车次.\*, 武汉至成都车次.\*, 厦门至武汉车次.最低票价+武汉至成都车次.最低票价 AS 总票价

FROM 厦门至武汉车次 INNER JOIN 武汉至成都车次 ON 厦门至武汉车次.到达时间+0.05<武汉至成都车次.出发时间

WHERE (厦门至武汉车次.车次编号 Like "G%" Or 厦门至武汉车次.车次编号 Like "D\*") And (厦门至武汉车次.出发时间 Between #7:00# And #8:00#);

**第四课时**

**教学重难点**

**【**教学重点**】**旅行小助手功能、界面、代码设计。

**【**教学难点**】**旅行小助手代码设计。

**【教学过程】**

**一、引入**

学生预习，阅读第73页“任务四 设计 “旅行小助手系统”之“活动1 设计系统功能和应用程序界面”，填写第74页的表3.3.1。教师检查，并评讲交流填写情况。

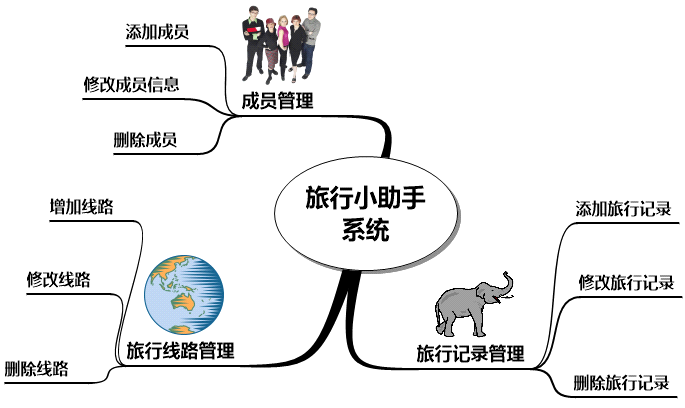


表3.3.1 系统界面首页中的对象及对应的功能

|  |  |
| --- | --- |
| 对象类别 | 对应的功能 |
| “成员信息管理”单选钮 | 单击选中“成员信息表”，并查看其内容 |
| “旅行线路管理”单选钮 | 单击选中“旅行线路表”，并查看其内容 |
| “旅行记录管理”单选钮 | 单击选中“旅行记录表”，并查看其内容 |
| “增加”按钮 | 单击在选中的数据表中增加记录 |
| “修改”按钮 | 单击修改选中的数据表中选中的记录 |
| “删除”按钮 | 单击删除选中的数据表中选中的记录 |
| “查询”按钮 | 单击创建查询 |
| “旅行组团”按钮 | 单击创建旅行组团 |

**二、设计系统数据库**

活动2 设计系统数据库，了解Python访问Access数据库的步骤

“旅行小助手系统”数据库（旅行小助手.mdb）：

内有3个数据表：成员信息表、旅行线路表、旅行记录表

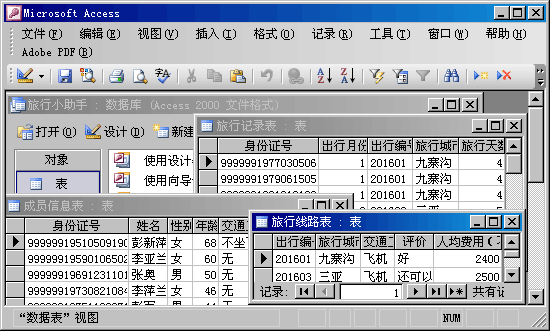
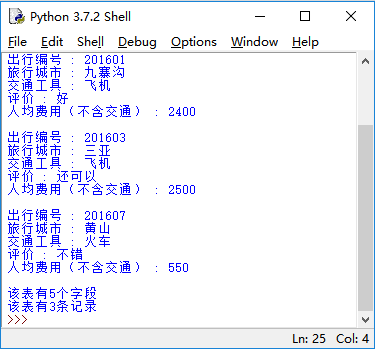


图3.3.8 “旅行小助手系统”数据库

创建了数据库之后，我们可以利用数据库及数据库管理系统对数据进行计算和分析，解决一些简单问题。但如果我们想要更高效、更便捷地对数据进行管理和使用，还需要专业人员用程序设计语言编写应用程序，实现更复杂的自动计算和分析。数据库应用系统可以提供简单、方便的操作界面，让我们能够利用系统解决更多的问题。

数据库应用系统的设计一般包括系统总体结构设计、代码设计、数据库（文件）设计、输入/输出（I/O）设计、模块功能与处理过程设计等步骤。“旅行小助手系统”在完成了系统界面设计和数据库设计之后，还需要经过编写应用程序、调试等环节才能成为真正的应用系统。

【做一做】请打开教科书配套资源中的“Python访问mdb.py”文件，运行程序，观察运行结果；阅读程序代码，体会Python访问mdb数据库的基本步骤。



用Python访问Access数据库一般可分为5个步骤：

（1）建立与数据库的连接；

利用win32com.client模块的COM组件访问功能，通过ADODB访问Access的mdb文件，因此先要安装与Python和操作系统匹配的pywin32安装包。

import win32com.client

conn = win32com.client.Dispatch(r"ADODB.Connection")

#DSN = 'PROVIDER = Microsoft.Jet.OLEDB.4.0;DATA SOURCE = 旅行小助手.mdb' #Access2007以前

DSN = 'PROVIDER = Microsoft.ACE.OLEDB.12.0;DATA SOURCE = 旅行小助手.mdb' #Access2007及以后

conn.Open(DSN)

（2）根据需要产生记录集；

rs = win32com.client.Dispatch(r'ADODB.Recordset')

tablename = '旅行线路表'

rs.Open('[' + tablename + ']', conn, 1, 3)

（3）访问记录集，处理数据；

#rs.MoveFirst() #光标移到首条记录

while not rs.EOF:

for i in range(rs.Fields.Count):

print(rs.Fields[i].Name, ":", rs.Fields[i].Value) #字段名：字段内容

print(end='\n')

rs.MoveNext() #光标移到下条记录

print('该表有'+str(rs.Fields.Count)+'个字段')

print('该表有'+str(rs.RecordCount)+'条记录')

（4）根据需要把处理好的数据更新到数据库中；

（5）断开与数据库的连接。

conn.Close()

**三、小结**

1.旅行小助手功能、界面、代码设计。

2.用Python访问mdb文件数据表的基本操作步骤。

**四、练习**

打开教科书配套资源中的“旅行小助手系统.py”文件，运行程序，观察运行结果；阅读程序代码，体会程序功能：查看数据表、增加记录、修改记录、删除记录；尝试能否开发实现其他功能。