
数据库系统

第三、四次上机报告

姓名： 宗永承

学号： 17130188021

日期： 2019年5月19日

目录

(一)	<u>实验要求</u>	<u>2</u>
(二)	<u>实验假设</u>	<u>2</u>
(三)	<u>基本功能</u>	<u>2</u>
(四)	<u>数据库设计</u>	<u>4</u>
(五)	<u>逻辑设计</u>	<u>6</u>
(六)	<u>功能实现</u>	<u>6</u>
(七)	<u>功能展示</u>	<u>7</u>
(八)	<u>实验总结</u>	<u>10</u>
(九)	<u>程序源码</u>	<u>10</u>

(一) 实验要求

基于 MySQL，设计并实现一个简单的旅行预订系统。该系统涉及的信息有航班、大巴班车、宾馆房间和客户数据等信息。其关系模式如下：

FLIGHTS (String flightNum, int price, int numSeats, int numAvail, String FromCity, String ArivCity);

HOTELS(String hotelNum, String location, int price, int numRooms, int numAvail);

BUS(String BusNum,String location, int price, int numBus, int numAvail);

CUSTOMERS(int custID,String custName);

RESERVATIONS(String resvNum, String custID, int resvType, String resvKey);

(二) 实验假设

- 1) 在给定的一个班机上，所有的座位价格也一样；flightNum 是表 FLIGHTS 的一个主码（primary key）。
- 2) 在同一个地方的所有的宾馆房间价格也一样；hotelNum 是表 HOTELS 的一个主码。
- 3) 在同一个地方的所有大巴车价格一样；BusNum 是表 BUS 的一个主码。
- 4) custID 是表 CUSTOMERS 的一个主码。
- 5) 表 RESERVATIONS 包含着那些和客户预订的航班、大巴车或宾馆房间相应的条目，具体的说，resvType 指出预订的类型（1 为预订航班，2 为预订宾馆房间，3 为预订大巴车），而 resvNum 是表 RESERVATIONS 的一个主码，resvKey 对应于预订的三种表的某个的主键。
- 6) 在表 FLIGHTS 中，numAvail 表示指定航班上的还可以被预订的座位数。对于一个给定的航班（flightNum），数据库一致性的条件之一是，表 RESERVATIONS 中所有预订该航班的条目数加上该航班的剩余座位数必须等于该航班上总的座位数。这个条件对于表 BUS 和表 HOTELS 同样适用。

(三) 基本功能

- 1) 航班，大巴车，宾馆房间和客户基础数据的入库，更新（表中的属性也可以根据你的需要添加）。
- 2) 预定航班，大巴车，宾馆房间。（取消预定功能）

- 3) 查询航班，大巴车，宾馆房间，客户和预订信息。（取消预定功能）
- 4) 查询某个客户的旅行线路。
- 5) 检查预定线路的完整性（线路是否合理：航班时间冲突）。

(四) 数据库设计

本系统数据库建表语句如下图：

```
# 创建航班表
create table FLIGHTS(
    flightNum    varchar(5) not null,
    price        int check(price>0),
    numSeats     int check(numSeats>0),
    numAvail     int,
    FromCity     varchar(20),
    ArivCity     varchar(20),
    primary key(flightNum));

# 创建宾馆房间表
create table HOTELS(
    hotelNum     varchar(5) not null,
    location     varchar(50),
    price        int check(price>0),
    numRooms     int check(numRooms>0),
    numAvail     int,
    primary key(hotelNum));

# 创建大巴班车表
create table BUS(
    busNum       varchar(5) not null,
    location     varchar(50),
    price        int check(price>0),
    numSeats     int check(numSeats>0),
    numAvail     int,
    primary key(busNum));

# 创建客户数据表
create table CUSTOMERS(
    custID       varchar(5) not null,
    custName     varchar(10) not null,
    primary key(custID));

# 创建预定表
create table RESERVATIONS(
    resvNum      varchar(5) not null,
    custID       varchar(5) not null,
    resvType     int check(resvType in (1, 2, 3)),
    resvKey      varchar(5) check(resvKey in
        (select flightNum from FLIGHTS
         union select hotelNum from HOTELS
         union select busNum from BUS)),
    primary key(resvNum));
```

(图 1)

为了保证数据库的一致性，预定完成后对应的 FLIGHTS、BUS、HOTELS 需要更新座位数和房间数量，采用触发器自动实现该功能。

由于本 MySQL 版本在触发器实现上不支持多个分支的 if 判断，对每个表均单独设置了触发器，创建触发器语句如下图。

```
# 预定后
# 更新航班信息
create trigger flight_update after insert on RESERVATIONS
for each row
update FLIGHTS
set     numAvail = numAvail - 1
where   new.resvType = 1 and flightNum = new.resvKey;

# 更新宾馆信息
create trigger hotel_update after insert on RESERVATIONS
for each row
update HOTELS
set     numAvail = numAvail - 1
where   new.resvType = 2 and hotelNum = new.resvKey;

# 更新大巴信息
create trigger bus_update after insert on RESERVATIONS
for each row
update BUS
set     numAvail = numAvail - 1
where   new.resvType = 3 and busNum = new.resvKey;

# 取消预定后
# 更新航班信息
create trigger flight_update2 after delete on RESERVATIONS
for each row
update FLIGHTS
set     numAvail = numAvail + 1
where   old.resvType = 1 and flightNum = old.resvKey;

# 更新宾馆信息
create trigger hotel_update2 after delete on RESERVATIONS
for each row
update HOTELS
set     numAvail = numAvail + 1
where   old.resvType = 2 and hotelNum = old.resvKey;

# 更新大巴信息
create trigger bus_update2 after delete on RESERVATIONS
for each row
update BUS
set     numAvail = numAvail + 1
where   old.resvType = 3 and busNum = old.resvKey;
```

(图 2)

(五) 逻辑设计

针对实验要求，我将旅游预定系统分为两大使用对象，一是系统管理员账户，具有对数据库系统内每个表增删改查的全部权限；二是普通用户，能够进行对航班、大巴车、宾馆房间的预定、查询、取消预定等功能。

进入系统后，选择 admin 账户登陆，显示管理员账户主菜单，可以选择 1~5 任意一个表进行后续操作，如下图：

```
D:\sql>TourBookingSystem.py
-----Welcome-----
请输入ID:admin
+-----TourBookingSystem-----+
| 1. FLIGHTS  2. BUS  3. HOTELS  4. CUSTOMERS |
| 5. Command  0. Quit |
+-----Administrator-----+
请选择服务(0/1/2/3/5):1
1. Insert 2. Delete 3. Query 4. Modify 0. Back
```

(图 3)

对于普通用户，先检查 custID 是否在本系统的 CUSTOMERS 表内，若不在，则需要登记新的用户。登记后显示用户菜单，可以选择相关操作。

```
-----Welcome-----
请输入ID:12345
欢迎新客户使用本系统
您的姓名: ZongYongcheng
+-----TourBookingSystem-----+
| 1. 预定航班/大巴车/宾馆房间 |
| 2. 查询航班/大巴车/宾馆房间预定信息 |
| 3. 取消航班/大巴车/宾馆房间预定 |
| 4. 查询旅行线路 |
| 5. 检查线路合理性 |
| 0. 退出 |
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):1
1. 预定航班 2. 预定大巴车 3. 预定宾馆房间:
```

(图 4)

(六) 功能实现

对于以上功能，使用 Python 语言连接数据库进行相关操作。在 admin.py 源代码中实现管理员的相关功能，在 dbSQL.py 内存放数据库登陆账户信息与输出格式等信息。最终实现的文件结构及全部函数如下图：

```
TourBookingSystem.py  dbSQL.py  admin.py
1  import random
2  from admin import *
3  from dbSQL import *
4
5  # 主菜单
6  # 选择预定/查询/取消预定
7  def mainMenu(): ...
22
23  # 内菜单
24  # 选择航班/酒店/大巴
25  def inMenu(str): ...
32
33  # 检查custID是否在数据库内
34  # ID为空时返回全部custID
35  def queryCustomer(ID): ...
58
59  # 插入新的用户
60  # 成功返回True, 失败返回False
61  def insertCustomer(ID, name): ...
80
81  # 用户预定
82  def CustomerService1(ID): ...
187
188  # 查询
189  # 根据ID查询全部预定信息
190  def CustomerService2(ID): ...
237
238  # 取消
239  # 根据resvNum取消相应预定信息
240  def CustomerService3(ID): ...
317
318  # 查询某个客户的旅行线路
319  def CustomerService4(ID): ...
353
354  # 检查线路合理性
355  def CustomerService5(ID): ...
400
401  # 函数入口
402  def Main(): ...
471
472  # 运行Main函数
473  Main()
```

(图 5)

(七) 功能展示

1. 预定航班

```
请选择服务(0/1/2/3/4/5):1
1. 预定航班 2. 预定大巴车 3. 预定宾馆房间:1
请输入出发/抵达城市(FromCity ArivCity):Wuhan Shanghai
所有航班信息如下:
flightNum, price, numSeats, numAvail, FromCity, ArivCity
('13384', 999, 35, 34, 'Wuhan', 'Shanghai')
请选择航班编号(输入0取消):13384
是否选择航班 13384 ?(Y/N):y
预定成功!
```

(图 6)

2. 预定班车

```
请选择服务(0/1/2/3/4/5):1
1. 预定航班 2. 预定大巴车 3. 预定宾馆房间:2
请输入城市:Wuhan
所有大巴车信息如下:
busNum, location, price, numSeats, numAvail
('31184', 'Wuhan', 160, 50, 49)
('35958', 'Wuhan', 160, 50, 50)
('38181', 'Wuhan', 160, 50, 50)
('38445', 'Wuhan', 160, 50, 50)
请选择大巴车编号(输入0取消):38445
是否选择大巴车 38445 ?(Y/N):y
预定成功!
```

(图 7)

3. 查询预定宾馆信息与取消宾馆

```
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):2
1. 查询预定航班 2. 查询预定大巴车 3. 查询预定宾馆房间:3
所有预定宾馆信息如下:
custName, custID, resvNum, hotelNum, price, location
('ZYC', '1234', '45836', '23040', 348, 'Wuhan')
('ZYC', '1234', '63159', '28184', 348, 'Wuhan')
-----查询End-----

+-----TourBookingSystem-----+
1. 预定航班/大巴车/宾馆房间
2. 查询航班/大巴车/宾馆房间预定信息
3. 取消航班/大巴车/宾馆房间预定
4. 查询旅行线路
5. 检查线路合理性
0. 退出
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):3
1. 取消预定航班 2. 取消预定大巴车 3. 取消预定宾馆房间:3
请输入预定编号:63159
查询的宾馆信息如下:
(('ZYC', '1234', '63159', '28184', 348, 'Wuhan'),)
是否取消预定(Y/N):y
已取消预定宾馆
-----取消End-----

+-----TourBookingSystem-----+
1. 预定航班/大巴车/宾馆房间
2. 查询航班/大巴车/宾馆房间预定信息
3. 取消航班/大巴车/宾馆房间预定
4. 查询旅行线路
5. 检查线路合理性
0. 退出
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):2
1. 查询预定航班 2. 查询预定大巴车 3. 查询预定宾馆房间:3
所有预定宾馆信息如下:
custName, custID, resvNum, hotelNum, price, location
('ZYC', '1234', '45836', '23040', 348, 'Wuhan')
-----查询End-----
```

(图 8)

4. 查询旅行线路


```

+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):2
1. 查询预定航班 2. 查询预定大巴车 3. 查询预定宾馆房间:1
所有预定航班信息如下:
custName, custID, resvNum, flightNum, price, FromCity, ArivCity
('zong', '11', '5651', '11254', 899, 'Beijing', "Xi'an")
('zong', '11', '95734', '12940', 1299, "Xi'an", 'Shanghai')
-----查询End-----

+-----TourBookingSystem-----+
1. 预定航班/大巴车/宾馆房间
2. 查询航班/大巴车/宾馆房间预定信息
3. 取消航班/大巴车/宾馆房间预定
4. 查询旅行线路
5. 检查线路合理性
0. 退出
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):4
Beijing -> Xi'an
Xi'an -> Shanghai
-----线路End-----

```

(图 9)

5. 检查线路合理性

用户名为 zong、ID 为 11 的顾客预定了 Beijing 到 Xi'an 和 Xi'an 到 Shanghai 的航班，也预定了在 Shanghai 和 Wuhan 的宾馆。Wuhan 不在旅行线路上，所以为不合理的旅行线路。

```

+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):2
1. 查询预定航班 2. 查询预定大巴车 3. 查询预定宾馆房间:1
所有预定航班信息如下:
custName, custID, resvNum, flightNum, price, FromCity, ArivCity
('zong', '11', '5651', '11254', 899, 'Beijing', "Xi'an")
('zong', '11', '95734', '12940', 1299, "Xi'an", 'Shanghai')
-----查询End-----

+-----TourBookingSystem-----+
1. 预定航班/大巴车/宾馆房间
2. 查询航班/大巴车/宾馆房间预定信息
3. 取消航班/大巴车/宾馆房间预定
4. 查询旅行线路
5. 检查线路合理性
0. 退出
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):2
1. 查询预定航班 2. 查询预定大巴车 3. 查询预定宾馆房间:3
所有预定宾馆信息如下:
custName, custID, resvNum, hotelNum, price, location
('zong', '11', '52858', '22385', 298, 'Shanghai')
('zong', '11', '89586', '23040', 348, 'Wuhan')
-----查询End-----

+-----TourBookingSystem-----+
1. 预定航班/大巴车/宾馆房间
2. 查询航班/大巴车/宾馆房间预定信息
3. 取消航班/大巴车/宾馆房间预定
4. 查询旅行线路
5. 检查线路合理性
0. 退出
+-----CustomerService-----+
请选择服务(0/1/2/3/4/5):5
您的旅行线路不合理

```

(图 10)

(八) 实验总结

这次数据库上机实验题目是一个面向实际应用的小项目，在我对数据库进行精心设计并持续一周的编程、调试的情况下，终于如期完成了本次实验的基本要求，完美实现了预定系统的全部功能。

本次上机实践过程中也遇到了不少问题，如可能是 MySQL 版本原因，我的数据库系统不支持多分枝的 if 语句。通过合理转化相关语句，上网查阅资料我最终都一一克服了这些障碍，并且也学习了不少新的知识，特别是对 Python 语言的使用更加熟练。

希望在今后继续加强数据库的学习，重视上机实践，编程水平更上一层楼。

(九) 程序源码（不包括数据库设计部分）

```
# TourBookingSystem.py
import random
from admin import *
from dbSQL import *

# 主菜单
# 选择预定/查询/取消预定
def mainMenu():
    print("+-----TourBookingSystem-----+")
    print("| 1. 预定航班/大巴车/宾馆房间 |")
    print("| 2. 查询航班/大巴车/宾馆房间预定信息 |")
    print("| 3. 取消航班/大巴车/宾馆房间预定 |")
    print("| 4. 查询旅行线路 |")
    print("| 5. 检查线路合理性 |")
    print("| 0. 退出 |")
    print("+-----CustomerService-----+")
    choice = input("请选择服务(0/1/2/3/4/5):")
    if choice in ["1", "2", "3", "4", "5", "0"]:
        return int(choice)
    else:
        print("请重新输入")
        return mainMenu()

# 内菜单
# 选择航班/酒店/大巴
def inMenu(str):
    choice = input("1.%s 航班 2.%s 大巴车 3.%s 宾馆房间:" % (str, str, str))
    if choice in ["1", "2", "3"]:
        return int(choice)
    else:
        print("请重新输入")
        return inMenu(str)

# 检查 custID 是否在数据库内
# ID 为空时返回全部 custID
def queryCustomer(ID):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
```

```
cursor = db.cursor()

sql = "select custID from CUSTOMERS where custID = '%s'" % (ID)
if ID=="":
    sql = "select custID from CUSTOMERS"

custs = []
try:
    # 执行 SQL 语句
    cursor.execute(sql)
    # 获取所有记录列表
    result = cursor.fetchall();
    for row in result:
        custs.append(row[0])
except:
    pass
# 关闭数据库连接
db.close()
return custs

# 插入新的用户
# 成功返回 True, 失败返回 False
def insertCustomer(ID, name):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    sql = "insert into CUSTOMERS values('%s', '%s')" % (ID, name)
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        # 提交到数据库执行
        db.commit()
        # 关闭数据库连接
        db.close()
        return True
    except:
        # 如果发生错误则回滚
        db.rollback()
        db.close()
        return False

# 查询 RESERVATIONS 表
def queryReserv(ID):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    sql = ""
    if ID == "":
        sql = "select * from RESERVATIONS"
    else:
        sql = "select * from RESERVATIONS where ID = '%s'" % (ID)

    resvs = []
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
```

```

        # 获取所有记录列表
        result = cursor.fetchall();
        for row in result:
            resvNum = row[0] #, row[1], row[2], row[3], row[4], row[5]
            resvs.append(resvNum)
            #print("%9s %5d %8d %8d %8s %8s" % (flightNum, price, numSeats,
numAvail, FromCity, ArivCity))
        except:
            print("Error!")
    # 关闭数据库连接
    db.close()
    return resvs

# 插入 RESERVATIONS 表
def insertReserv(ID, resvType, resvKey):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    resvNum = str(random.randint(1, 100000))
    while resvNum in queryReserv(""):
        resvNum = str(random.randint(1, 100000))

    sql = "insert into RESERVATIONS values('%s', '%s', %d, '%s')" % (resvNum, ID,
resvType, resvKey)
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        # 提交到数据库执行
        db.commit()
        db.close()
        return True
    except:
        # 如果发生错误则回滚
        db.rollback()
        db.close()
        return False

# 用户预定
def CustomerService1(ID):
    ch2 = inMenu("预定")
    # 预定航班
    if ch2==1:
        sql = "select * from FLIGHTS"
        try:
            FromCity, ArivCity = input("请输入出发/抵达城市(FromCity
ArivCity):").split(" ")
            if FromCity!=" " and ArivCity!=" " :
                sql = "select * from FLIGHTS where FromCity = '%s' and ArivCity =
'%s'" % (FromCity, ArivCity)
        except:
            print("输入格式有误,将查询全部航班")
            FromCity, ArivCity = "", ""
        # 查询航班
        flights = exctQueryySQL(sql)
        if len(flights)==0 :
            print("很抱歉, 没有您想要的航班")
        else :

```

```

# 显示航班信息
print("所有航班信息如下:")
print(FORMAT_FLIGHTS)
for each in flights :
    print(each)
# 预定航班
flightNum = "-1"
flightNums = [x[0] for x in flights]
while flightNum not in flightNums:
    flightNum = input("请选择航班编号(输入 0 取消):")
    if flightNum=="0":
        break
# 确认
if flightNum!="0" :
    dec = input("是否选择航班 %s ?(Y/N):" % (flightNum))
    if dec=="Y" or dec=="y" :
        insertReserv(ID, 1, flightNum)
        print("预定成功!")
    elif dec=="N" or dec=="n" :
        print("操作已取消,感谢您的使用")

# 预定大巴车
elif ch2==2:
    City = ""
    try:
        City = input("请输入城市:")
    except:
        print("输入为空,将查询全部大巴车")
        City = ""
# 查询大巴车
querySQL = "select * from BUS where location = '%s'" % (City)
buses = exctQuerySQL(querySQL)
if len(buses)==0 :
    print("很抱歉,没有您想要的大巴车")
else :
    # 显示大巴车信息
    print("所有大巴车信息如下:")
    print(FORMAT_BUS)
    for each in buses :
        print(each)

# 预定大巴车
busNum = "-1"
busNums = [x[0] for x in buses]
while busNum not in busNums:
    busNum = input("请选择大巴车编号(输入 0 取消):")
    if busNum=="0":
        break
# 确认
if busNum!="0" :
    dec = input("是否选择大巴车 %s ?(Y/N):" % (busNum))
    if dec=="Y" or dec=="y" :
        insertReserv(ID, 3, busNum)
        print("预定成功!")
    elif dec=="N" or dec=="n" :
        print("操作已取消,感谢您的使用")

# 预定宾馆

```

```

else:
    City = input("请输入城市:")
    # 查询宾馆
    querySQL = "select * from HOTELS where location = '%s'" % (City)
    hotels = exctQuerySQL(querySQL)
    if len(hotels)==0 :
        print("很抱歉, 没有您想要的宾馆")
    else :
        # 显示宾馆信息
        print("所有宾馆信息如下:")
        print(FORMAT_HOTELS)
        for each in hotels :
            print(each)
        # 预定宾馆
        hotelNum = "-1"
        hotelNums = [x[0] for x in hotels]
        while hotelNum not in hotelNums:
            hotelNum = input("请选择宾馆编号(输入 0 取消):")
            if hotelNum=="0":
                print("操作取消, 感谢您的使用")
                break
        # 确认
        if hotelNum!="0" :
            dec = input("是否选择宾馆 %s ?(Y/N):" % (hotelNum))
            if dec=="Y" or dec=="y" :
                insertReserv(ID, 2, hotelNum)
                print("预定成功!")
            elif dec=="N" or dec=="n" :
                print("操作已取消, 感谢您的使用")
        print("-----预定 End-----\n")

# 查询
# 根据 ID 查询全部预定信息
def CustomerService2(ID):
    ch2 = inMenu("查询预定")
    # 查询航班预定信息
    if ch2==1:
        querySQL = "select %s \
                    from RESERVATIONS natural join CUSTOMERS, FLIGHTS \
                    where resvKey = flightNum and resvType = 1 and custID = \
%s'" % (FORMAT+", FromCity, ArivCity", ID)
        flights = exctQuerySQL(querySQL)
        if len(flights)==0 :
            print("您没有预定任何航班")
        else :
            # 显示航班信息
            print("所有预定航班信息如下:")
            print("custName, custID, resvNum, flightNum, price, FromCity, \
ArivCity")
            for each in flights :
                print(each)

    # 查询大巴车预定信息
    elif ch2==2:
        querySQL = "select %s \
                    from RESERVATIONS natural join CUSTOMERS, BUS \
                    where resvKey = busNum and resvType = 3 and custID = '%s'" % \
(FORMAT+", location", ID)

```

```

buses = exctQuerySQL(querySQL)
if len(buses)==0 :
    print("您没有预定任何大巴车")
else :
    # 显示大巴车信息
    print("所有预定大巴车信息如下:")
    print("custName, custID, resvNum, busNum, price, location")
    for each in buses :
        print(each)

# 查询宾馆预定信息
else:
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, HOTELS \
                where resvKey = hotelNum and resvType = 2 and custID = '%s'" %
(FORMAT+", location", ID)
    hotels = exctQuerySQL(querySQL)
    if len(hotels)==0 :
        print("您没有预定任何宾馆")
    else :
        # 显示宾馆信息
        print("所有预定宾馆信息如下:")
        print("custName, custID, resvNum, hotelNum, price, location")
        for each in hotels :
            print(each)
    print("-----查询 End-----\n")

# 取消
# 根据 resvNum 取消相应预定信息
def CustomerService3(ID):
    ch2 = inMenu("取消预定")
    resvNum = input("请输入预定编号:")
    # 取消预定航班
    if ch2==1:
        while True :
            sql = "select %s \
                    from RESERVATIONS natural join CUSTOMERS, FLIGHTS \
                    where resvKey = flightNum and resvType = 1 and resvNum =
'%s'" % (FORMAT+", FromCity, ArivCity", resvNum)
            # 查询航班
            flight = exctQuerySQL(sql)
            if flight :
                # 显示航班信息
                print("查询的航班信息如下:")
                print(flight)

                cancel= input("是否取消预定(Y/N):")
                if cancel=="Y" or cancel=="y" :
                    exctDeleteSQL("delete from RESERVATIONS where resvNum =
'%s'" % (resvNum))
                    print("已取消预定航班")
                elif cancel=="N" or cancel=="n" :
                    print("操作已取消,感谢您的使用")
                    break
            else :
                resvNum=input("很抱歉,没有预定信息,请重新输入编号(输入 0 取消):")
                if resvNum=="0" :
                    break

```

```

# 取消预定大巴车
elif ch2==2:
    while True :
        sql = "select %s \
                from RESERVATIONS natural join CUSTOMERS, BUS \
                where resvKey = busNum and resvType = 3 and resvNum = '%s'" %
(FORMAT+", location", resvNum)
        # 查询大巴车
        bus = exctQuerySQL(sql)
        if bus :
            # 显示大巴车信息
            print("查询的大巴车信息如下:")
            print(bus)
            cancel= input("是否取消预定(Y/N):")
            if cancel=="Y" or cancel=="y" :
                exctDeleteSQL("delete from RESERVATIONS where resvNum =
'%s'" % (resvNum))
                print("已取消预定大巴车")
            elif cancel=="N" or cancel=="n" :
                print("操作已取消,感谢您的使用")
                break
        else :
            resvNum=input("很抱歉,没有预定信息,请重新输入编号(输入 0 取消):")
            if resvNum=="0" :
                break

# 取消预定宾馆
else:
    while True :
        sql = "select %s \
                from RESERVATIONS natural join CUSTOMERS, HOTELS \
                where resvKey = hotelNum and resvType = 2 and resvNum =
'%s'" % (FORMAT+", location", resvNum)
        # 查询宾馆
        hotel = exctQuerySQL(sql)
        if hotel :
            # 显示宾馆信息
            print("查询的宾馆信息如下:")
            print(hotel)

            cancel= input("是否取消预定(Y/N):")
            if cancel=="Y" or cancel=="y" :
                exctDeleteSQL("delete from RESERVATIONS where resvNum =
'%s'" % (resvNum))
                print("已取消预定宾馆")
            elif cancel=="N" or cancel=="n" :
                print("操作已取消,感谢您的使用")
                break
        else :
            resvNum=input("很抱歉,没有预定信息,请重新输入编号(输入 0 取消):")
            if resvNum=="0" :
                break
    print("-----取消 End-----\n")

# 查询某个客户的旅行线路
def CustomerService4(ID):
    # 查询航班预定信息

```



```

        querySQL = "select %s \
                    from RESERVATIONS natural join CUSTOMERS, FLIGHTS \
                    where resvKey = flightNum and resvType = 1 and custID = '%s'" %
(FORMAT+", FromCity, ArivCity", ID)
        flights = exctQuerySQL(querySQL)
        if len(flights) :
            # 获取航班信息
            for each in flights :
                FromCity, ArivCity = each[5], each[6]
                print("%s -> %s" % (FromCity, ArivCity))
        else :
            print("您未预定航班")
        print("-----线路 End-----\n")

# 检查线路合理性
def CustomerService5(ID):
    # 查询航班预定信息
    flag = False
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, FLIGHTS \
                where resvKey = flightNum and resvType = 1 and custID = '%s'" %
(FORMAT+", FromCity, ArivCity", ID)
    flights = exctQuerySQL(querySQL)
    cities = []
    if len(flights) :
        # 显示航班信息
        # print("custName, custID, resvNum, flightNum, price, FromCity, ArivCity")
        for each in flights :
            cities.append(each[5])
            cities.append(each[6])
    else :
        print("您未预定航班")
        return

    # 查询大巴车预定信息
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, BUS \
                where resvKey = busNum and resvType = 3 and custID = '%s'" %
(FORMAT+", location", ID)
    buses = exctQuerySQL(querySQL)
    if len(buses) :
        # 获取大巴车所在城市
        for each in buses :
            if each[5] not in cities:
                print("您的旅行线路不合理")
                return
    else :
        print("您未预定大巴车")
        return

    # 查询宾馆预定信息
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, HOTELS \
                where resvKey = hotelNum and resvType = 2 and custID = '%s'" %
(FORMAT+", location", ID)
    hotels = exctQuerySQL(querySQL)
    if len(hotels) :
        # 获取宾馆所在城市

```

```

        for each in hotels :
            if each[5] not in cities:
                print("您的旅行线路不合理")
                return
    else :
        print("您未预定宾馆")
        return

    print("您的旅行线路合理")

# 函数入口
def Main():
    while True :
        print("-----Welcome-----")
        ID = input("请输入 ID:")
        # 系统管理员
        if ID=="admin" :
            while True :
                ch1 = Menu()
                # 退出管理系统
                if ch1==0:
                    print("-----已退出管理系统-----\n")
                    break
                elif ch1==5:
                    while True:
                        print("Please input MySQL instructors in ONE line(Input
exit to quit)")
                        command = input()
                        if command=="exit":
                            break
                        if exctSQL(command):
                            print("成功执行")
                        else:
                            print("执行失败")
                    break

            # 增删查改
            choice = input("1.Insert 2.Delete 3.Query 4.Modify 0.Back\n")
            if choice=="1":
                # 添加航班/大巴/宾馆/客户
                Insert(ch1)
            elif choice=="2":
                # 删除航班/大巴/宾馆/客户
                Delete(ch1)
            elif choice=="3":
                # 查询航班/大巴/宾馆/客户
                Query(ch1)
            elif choice=="4":
                # 修改航班/大巴/宾馆/客户
                print("暂未开放")
            else :
                pass

        # 退出
        elif ID=="exit" :
            break

# 用户端服务

```

```

else :
    if ID not in queryCustomer(ID):
        print("欢迎新客户使用本系统")
        name = input("您的姓名: ")
        if insertCustomer(ID, name) :
            print("您的信息已登记")

    while True:
        ch1 = mainMenu()
        # 选择预定/查询/取消预定
        if ch1==1 :
            CustomerService1(ID)
        elif ch1==2 :
            CustomerService2(ID)
        elif ch1==3 :
            CustomerService3(ID)
        elif ch1==4 :
            CustomerService4(ID)
        elif ch1==5 :
            CustomerService5(ID)
        else :
            print("-----已退出用户服务-----\n")
            break

# 运行 Main 函数
Main()

# admin.py
from dbSQL import *

def Menu():
    print("+-----TourBookingSystem-----+")
    print("| 1.FLIGHTS  2.BUS  3.HOTELS  4.CUSTOMERS |")
    print("| 5.Command  0.Quit                               |")
    print("+-----Administrator-----+")
    choice = input("请选择服务(0/1/2/3/5):")
    if choice in ["1","2","3","4","5","0"]:
        return int(choice)
    else:
        print("请重新输入")
        return Menu()

# 插入
def Insert(ch1):
    # 插入航班
    if ch1==1:
        flightNum, price, numSeats, FromCity, ArivCity = input("请输入增加的航班信息
(flightNum price numSeats FromCity ArivCity):").split(" ")
        sql = "insert into FLIGHTS values('%s', %d, %d, %d, '%s', '%s')" %
(flightNum, int(price), int(numSeats), int(numSeats), FromCity, ArivCity)
        # 插入大巴
        elif ch1==2:
            busNum, location, price, numSeats = input("请输入增加的大巴车信息(busNum
location price numSeats):").split(" ")
            sql = "insert into BUS values('%s', '%s', %d, %d, %d)" % (busNum, location,
int(price), int(numSeats), int(numSeats))

```

```

# 插入宾馆
elif ch1==3:
    hotelNum, location, price, numRooms = input("请输入增加的宾馆信息(hotelNum
location price numRooms):").split(" ")
    sql = "insert into HOTELS values('%s', '%s', %d, %d, %d)" % (hotelNum,
location, int(price), int(numRooms), int(numRooms))
    # 插入客户
else:
    custID, custName = input("请输入增加的客户信息(custID custName):").split(" ")
    sql = "insert into HOTELS values('%s', '%s')" % (custID, custName)

if exctInsertSQL(sql):
    print("插入成功")
else :
    print("插入失败")

# 删除
def Delete(ch1):
    sql = ""
    # 删除航班
    if ch1==1:
        flightNum = input("请输入删除的航班编号(flightNum):")
        sql = "delete from FLIGHTS where flightNum = '%s'" % (flightNum)
    # 删除大巴
    elif ch1==2:
        busNum = input("请输入删除的大巴编号(busNum):")
        sql = "delete from BUS where busNum = '%s'" % (busNum)
    # 删除宾馆
    elif ch1==3:
        hotelNum = input("请输入删除的宾馆编号(hotelNum):")
        sql = "delete from HOTELS where hotelNum = '%s'" % (hotelNum)
    # 删除客户
    else :
        custID = input("请输入删除的客户编号(custID):")
        sql = "delete from CUSTOMERS where custID = '%s'" % (custID)
    if exctDeleteSQL(sql):
        print("删除成功")
    else :
        print("删除失败")

# 查询(包含/不包含预定)
def Query(ch1):
    order = input("是否包含预定信息(Y/N)?")
    if order=="Y" or order=="y":
        # 查询航班预定信息
        if ch1==1:
            querySQL = "select %s \
                        from RESERVATIONS natural join CUSTOMERS, FLIGHTS \
                        where resvKey = flightNum and resvType = 1" % (FORMAT+",
FromCity, ArivCity")
            flights = exctQuerySQL(querySQL)
            if len(flights)==0 :
                print("没有预定任何航班")
            else :
                # 显示航班信息
                print("所有预定航班信息如下:")
                print("custID, custName, resvNum, flightNum, price, FromCity,
ArivCity")

```

```
        for each in flights :
            print(each)

# 查询大巴车预定信息
elif ch1==2:
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, BUS \
                where resvKey = busNum and resvType = 3" % (FORMAT+",
location")
    buses = exctQuerySQL(querySQL)
    if len(buses)==0 :
        print("没有预定任何大巴车")
    else :
        # 显示大巴车信息
        print()
        print("所有预定大巴车信息如下:")
        print("custID, custName, resvNum, busNum, price, location")
        for each in buses :
            print(each)

# 查询宾馆预定信息
elif ch1==3:
    querySQL = "select %s \
                from RESERVATIONS natural join CUSTOMERS, HOTELS \
                where resvKey = hotelNum and resvType = 2" % (FORMAT+",
location")
    hotels = exctQuerySQL(querySQL)
    if len(hotels)==0 :
        print("没有预定任何宾馆")
    else :
        # 显示宾馆信息
        print("所有预定宾馆信息如下:")
        print("custID, custName, resvNum, hotelNum, price, location")
        for each in hotels :
            print(each)

# 查询客户预定信息
else:
    querySQL = "select custID, custName, resvType, resvNum \
                from RESERVATIONS natural join CUSTOMERS order by custID"
    custs = exctQuerySQL(querySQL)
    if len(custs)==0 :
        print("没有客户预定")
    else :
        # 显示客户信息
        print("所有客户预定信息如下:")
        print("custID, custName, resvType, resvNum")
        for each in custs :
            print(each)

else:
    # 查询航班
    if ch1==1:
        querySQL = "select * from FLIGHTS"
        flights = exctQuerySQL(querySQL)
        if len(flights)==0 :
            print("没有航班")
        else :
            # 显示航班信息
```

```
        print("所有航班信息如下:")
        print("flightNum, price, numSeats, numAvail, FromCity, ArivCity")
        for each in flights :
            print(each)

# 查询大巴车
elif ch1==2:
    querySQL = "select * from BUS"
    buses = exctQuerySQL(querySQL)
    if len(buses)==0 :
        print("没有大巴车")
    else :
        # 显示大巴车信息
        print("所有大巴车信息如下:")
        print("busNum, location, price, numSeats, numAvail")
        for each in buses :
            print(each)

# 查询宾馆
elif ch1==3:
    querySQL = "select * from HOTELS"
    hotels = exctQuerySQL(querySQL)
    if len(hotels)==0 :
        print("没有宾馆")
    else :
        # 显示宾馆信息
        print("所有宾馆信息如下:")
        print("hotelNum, location, price, numRooms, numAvail")
        for each in hotels :
            print(each)

# 查询客户
else:
    querySQL = "select * from CUSTOMERS"
    custs = exctQuerySQL(querySQL)
    if len(custs)==0 :
        print("没有客户")
    else :
        # 显示客户信息
        print("所有客户信息如下:")
        print("custID, custName")
        for each in custs :
            print(each)

# dbSQL.py
import pymysql

DataBaseUser = "root"
DataBasePassword = "1234"
DataBaseName = "TourBooking"

FORMAT = "custName, custID, resvNum, resvKey, price"
FORMAT_FLIGHTS = "flightNum, price, numSeats, numAvail, FromCity, ArivCity"
FORMAT_BUS = "busNum, location, price, numSeats, numAvail"
FORMAT_HOTELS = "hotelNum, location, price, numRooms, numAvail"
```

```
# 执行 MySQL 添加
# 成功返回 True, 失败返回 False
def exctInsertSQL(sql):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        db.commit()
        # 关闭数据库连接
        db.close()
        return True
    except:
        # 如果发生错误则回滚
        db.rollback()
        db.close()
        return False

# 执行 MySQL 删除
# 成功返回 True, 失败返回 False
def exctDeleteSQL(sql):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        db.commit()
        # 关闭数据库连接
        db.close()
        return True
    except:
        # 如果发生错误则回滚
        db.rollback()
        db.close()
        return False

# 执行 MySQL 查询
# 返回全部所有表
def exctQuerySQL(sql):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()

    result = ()
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        # 获取所有记录列表
        result = cursor.fetchall()
    except:
        # 如果发生错误则回滚
        db.rollback()
    finally:
```

```
        # 关闭数据库连接
        db.close()
    return result

# exctUpdateSQL(sql):

def exctSQL(sql):
    # 打开数据库连接
    db = pymysql.connect("localhost", DataBaseUser, DataBasePassword, DataBaseName)
    # 使用 cursor()方法获取操作游标
    cursor = db.cursor()
    try:
        # 执行 SQL 语句
        cursor.execute(sql)
        db.commit()
        # 关闭数据库连接
        db.close()
        return True
    except:
        # 如果发生错误则回滚
        db.rollback()
        db.close()
    return False
```