第三次数据库实验 姓名:曾凡浩 学号：20009200713

- 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 location, int price, int numRooms, int numAvail)；

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

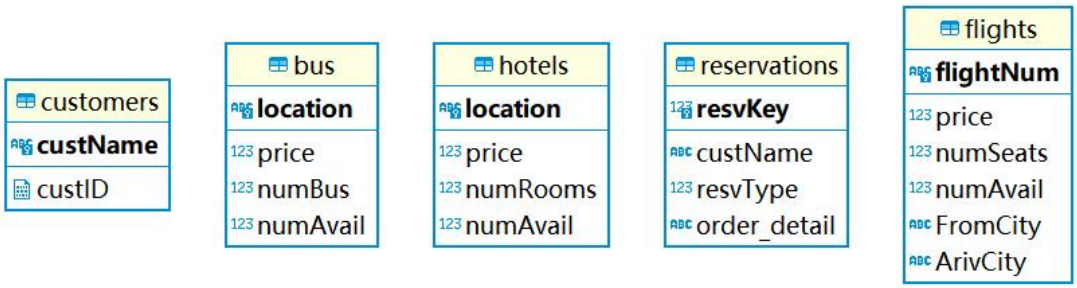
CUSTOMERS(String custName,custID)；

RESERVATIONS(String custName, int resvType, String resvKey)

为简化系统的复杂度，对所实现的应用系统作下列假设：

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；location 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；location 是表 BUS 的一个主码。
4. custName 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 resvKey 是表 RESERVATIONS 的一个主码。
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。

- 数据库设计



其中 custName bus.location hotels.location resvKey flightNum 分别为各 table 的主码，在最开始思考的时候，有想过是否需要用外键来对各表的取值域进行约束，但各 table 中的属性都显得独立，只有 reservations 上的 custName 的取值域有约束，该属性的取值必须是在

customers 上面登记过的 custName，但在程序设计的过程中在用户发送订票请求的时候，会单独判断该用户是否是在数据库中已经出现，所以就没有设置外键约束。在题设中对于 reservations 这个 table 只有 resvKey , custNaeme , resvType，但仅仅有着四个属性的话对于实现对用户航线进行查询的时候就无法知道其具体航班号，对于其预定的 bus 和 hotel 也将不知道其具体位置，所以为了后续功能的实现，添加了一个 order_detail 属性，当 resvType 是 1 的时候，就存入航班信息，其他情况就存入预定 bus 或者 hotel 的地理位置。

- 程序实现
  - 程序的分层

本次程序的实现一共分为了四个层其中分别是 dao 层，model 层，service 层，rescources 层。其中各层的作用如下：

Dao 层：分为了 dao 接口以及 baseDao 。 dao 接口中封装了本次实验中对数据库进行增删改查的具体方法，例如对数据的更新，查询航班等等。BaseDao 是负责最底层的操作，包括数据库的连接，最基础的增删改查，以及资源的释放。

Model 层：分为了 bus , customers , flights , hotels , reservations。其实就是分别对应了数据库设计中各个 table，其中的私有变量也就对应各个 table 中的属性，以方便在后续对数据库进行插入，以及查询航班等功能的实现。

Service 层：对于 dao 中封装的方法进行具体的实现。

Rescources 层：demo 其中放置一些静态方法，来实现界面的现实，在对于数据库进行连接的时候是需要保存配置文件来告诉驱动数据库名，用户名，密码，但是数据库信息的配置文件只能放在 src 根目录所以在实验过程中遇到一个问题。

Main 函数通过 swith-case 来实现对不同功能的分隔

  - 具体功能实现
    - 更新数据

首先调用 demo 中的静态方法展现菜单后，将对于航班信息，大巴信息，酒店信息，客户信息的更新和修改进行分隔。下面以航班信息为例，选择好种类之后显示具体功能是插入数据还是修改数据，首先会创建一个 flight 对象，通过 set 方法将对象的信息补全，如果是插入就需要先在数据库中查找是否该航班是否已经存在，因为航班号是主码，当无此航班时才能插入；更新于此类似，得数据库中已有该航班才能允许更新。对于大巴信息，酒店信息，客户信息的更新操作于此同理。

    - 预订信息

对于预定信息，首先必须预定的用户已经在数据库 customers 表中登记过，才会让其进行预定操作。如果该用户存在，则显示预定菜单在菜单中进行预定操作种类的选择，是预定航班，预定大巴等等，在选择完功能后会将数据库中该类信息全部打印输出，以预定航班为例，会输出所有的航班号，航班价格，座位总数等等，其实就是数据库中 flight 表中的所有属性。然后通过航班号来进行预定（大巴等就是 location，其实就是各表的主码），完成后需要再 flight 表中将 NumAvail 减一来模拟下单，并且在 reservations 表中插入一个元组，来模拟生成订单信息。当然再下单的时候会判断 NumAvail 是否为 0，当没票的时候自然无法下单。其余下单同理。

- 查询信息

  查询信息的时候是可选择查询什么信息，航班？大巴？等，选中之后就直接输出数据库中该表的所有信息。这么做其实是有一些偷懒，按道理应该还需要实现通过 flightNum 等详细信息来进行查询，但由于在实现下单功能的时候会输出所有的信息，已经写好了一个现成的方法，所以就在这直接套用。要实现按信息查询也很简单，也可通过%来进行模糊搜索而不需要精准匹配。

- 查询路线

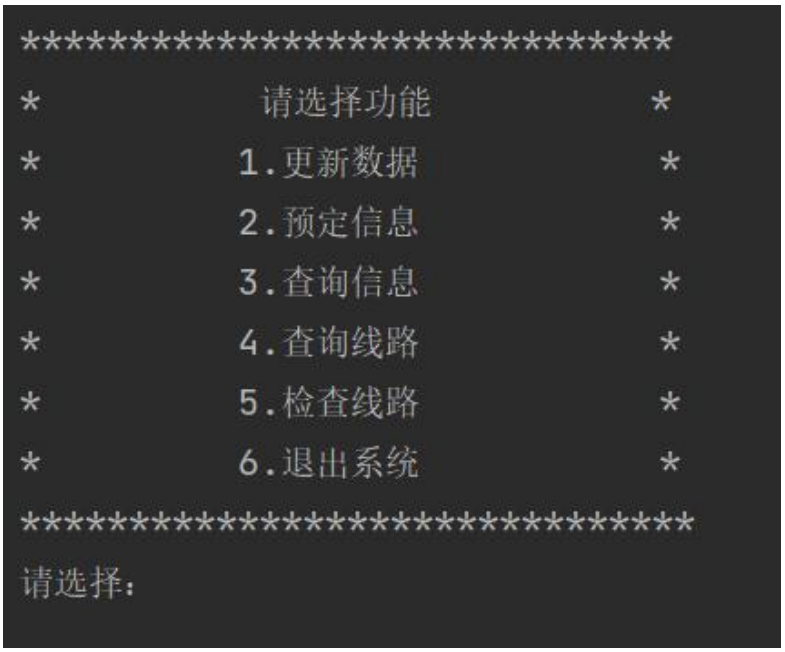
  这个功能的实现就相对于之前的功能复杂，同样首先会判断需要查询的用户名是否是数据库中已有的用户。较之前变难的地方在于，查询线路我们需要保证线路的关联性，即输出的信息应该是从起始站到终点站，而不是简单的将该用户的所有航班查到之后简单的输出 FromCity 和 ArivCity。首先先用一个 list 来记录该用户的所有航班信息，然后只有起始站和终点站才只会出现一次，而中转站一定会在 FromCity 和 ArivCity 中都有记录。知道起始站后通过起始站的到达站再作为起始站在寻找下一站，直到找到终点站。

- 检查线路

  我们首先输入起点城市和终点城市，然后通过<flight>的 list 将数据库中所有航班的 FromCity 和 ArivCity 记录下来，先做一个简单的判断，如果说起点城市和终点城市都包含在其中，那么线路一定是不完整的。如果是在其中的就看这连个结点之间能否联通，于是我通过一个递归输入起始城市，通过查询表看起能到什么中转城市，再将中转城市作为起始城市去递归，如果说能在某一航班的 ArivCity 就是终点城市那么证明两个地方的线路是完整的，否则就是不完整的。

- 成果展示

  - 起始界面



- 更新数据(以 flight 为例)
  - 插入

```
**********************************
*                请选择功能              *
*            1.更新航班数据              *
*            2.更新大巴数据              *
*            3.更新宾馆数据              *
*            4.更新客户数据              *
**********************************
请选择：
1
请选择功能：1：数据入库   2：更新数据
1
请输入：
(String)flight_number:
MA1234
(int)price:
987
(int)numSeats:
100
(String)FromCity:
XXX
(String)ArivCity:
ZZZ
 Insert Success!!!
**********************************
```

2	CA1312	999	50	50	zunyi	chengdu
3	MA1234	987	100	100	XXX	ZZZ

◆ 更新

```
**********************************
*                请选择功能              *
*             1.更新航班数据             *
*             2.更新大巴数据             *
*             3.更新宾馆数据             *
*             4.更新客户数据             *
**********************************
请选择:
1
请选择功能：1：数据入库   2：更新数据
2
请输入:
(String)flight_number:
MA1234
(int)price:
99999
(int)numSeats:
345
(String)FromCity:
HHHHH
(String)ArivCity:
XXXXX
Update Success!!!
```

3	MA1234	99,999	345	100	HHHHH	XXXXX

■ 预订信息

```
请输入客户名称：
zfh
*********************************
*           请选择功能          *
*         1.预定航班            *
*         2.预定大巴            *
*         3.预定宾馆            *
*********************************
请选择：
1
(flightNum):CA1301     (price):1899     (numSeats):50     (numAvail):38     (FromCity):beijing     (ArivCity):guangzhou
(flightNum):CA1302     (price):1599     (numSeats):50     (numAvail):49     (FromCity):guangzhou     (ArivCity):shanghai
(flightNum):CA1303     (price):899     (numSeats):50     (numAvail):49     (FromCity):shanghai     (ArivCity):shenzhen
(flightNum):CA1304     (price):1499     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):beijing
(flightNum):CA1305     (price):799     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):shanghai
(flightNum):CA1306     (price):999     (numSeats):50     (numAvail):50     (FromCity):xian     (ArivCity):beijing
(flightNum):CA1307     (price):999     (numSeats):50     (numAvail):50     (FromCity):shandong     (ArivCity):beijing
(flightNum):CA1308     (price):999     (numSeats):50     (numAvail):50     (FromCity):nanjing     (ArivCity):xian
(flightNum):CA1309     (price):999     (numSeats):50     (numAvail):50     (FromCity):chengdu     (ArivCity):nanjing
(flightNum):CA1310     (price):999     (numSeats):50     (numAvail):50     (FromCity):chongqing     (ArivCity):chengdu
(flightNum):CA1311     (price):999     (numSeats):50     (numAvail):50     (FromCity):dali     (ArivCity):zunyi
(flightNum):CA1312     (price):999     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):chengdu
(flightNum):MA1234     (price):99999     (numSeats):345     (numAvail):100     (FromCity):HHHHH     (ArivCity):XXXXX
```

```
flight choise:
MA1234
Order flight success!!!
```

MA1234	99,999	345	99	HHHHH	XXXXX

6	zfh	1	6	MA1234

■ 查询信息

```
*********************************
*           请选择功能          *
*         1.查询航班            *
*         2.查询大巴            *
*         3.查询宾馆            *
*         4.查询客户            *
*         5.查询预定            *
*********************************
请选择：
1
(flightNum):CA1301     (price):1899     (numSeats):50     (numAvail):38     (FromCity):beijing     (ArivCity):guangzhou
(flightNum):CA1302     (price):1599     (numSeats):50     (numAvail):49     (FromCity):guangzhou     (ArivCity):shanghai
(flightNum):CA1303     (price):899     (numSeats):50     (numAvail):49     (FromCity):shanghai     (ArivCity):shenzhen
(flightNum):CA1304     (price):1499     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):beijing
(flightNum):CA1305     (price):799     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):shanghai
(flightNum):CA1306     (price):999     (numSeats):50     (numAvail):50     (FromCity):xian     (ArivCity):beijing
(flightNum):CA1307     (price):999     (numSeats):50     (numAvail):50     (FromCity):shandong     (ArivCity):beijing
(flightNum):CA1308     (price):999     (numSeats):50     (numAvail):50     (FromCity):nanjing     (ArivCity):xian
(flightNum):CA1309     (price):999     (numSeats):50     (numAvail):50     (FromCity):chengdu     (ArivCity):nanjing
(flightNum):CA1310     (price):999     (numSeats):50     (numAvail):50     (FromCity):chongqing     (ArivCity):chengdu
(flightNum):CA1311     (price):999     (numSeats):50     (numAvail):50     (FromCity):dali     (ArivCity):zunyi
(flightNum):CA1312     (price):999     (numSeats):50     (numAvail):50     (FromCity):zunyi     (ArivCity):chengdu
(flightNum):MA1234     (price):99999     (numSeats):345     (numAvail):99     (FromCity):HHHHH     (ArivCity):XXXXX
*********************************
```

■ 查询线路

```
*****************************
*          请选择功能          *
*         1.更新数据           *
*         2.预定信息           *
*         3.查询信息           *
*         4.查询线路           *
*         5.检查线路           *
*         6.退出系统           *
*****************************
请选择：
4
请输入客户名：
zfh
zfh route:
beijing===>guangzhou===>shanghai===>shenzhen
*****************************
```

	custName	resvType	resvKey	order_detail
1	zfh	1	1	ca1301
2	zfh	2	2	beijing
3	zfh	3	3	beijing
4	zfh	1	4	ca1302
5	zfh	1	5	ca1303

■ 检查线路完整性

```
Start:
beijing
End:
shenzhen
[guangzhou, shanghai, shenzhen]
Route is complete!!!
*****************************
*          请选择功能          *
*         1.更新数据           *
*         2.预定信息           *
*         3.查询信息           *
*         4.查询线路           *
*         5.检查线路           *
*         6.退出系统           *
*****************************
请选择：
5
Start:
beijing
End:
chendu
Route is not complete!!!
```

	flightNum	price	numSeats	numAvail	FromCity	ArivCity
1	CA1301	1,899	50	38	beijing	guangzhou
2	CA1302	1,599	50	49	guangzhou	shanghai
3	CA1303	899	50	49	shanghai	shenzhen
4	CA1304	1,499	50	50	zunyi	beijing
5	CA1305	799	50	50	zunyi	shanghai
6	CA1306	999	50	50	xian	beijing
7	CA1307	999	50	50	shandong	beijing
8	CA1308	999	50	50	nanjing	xian
9	CA1309	999	50	50	chengdu	nanjing
10	CA1310	999	50	50	chongqing	chengdu
11	CA1311	999	50	50	dali	zunyi
12	CA1312	999	50	50	zunyi	chengdu
13	MA1234	99,999	345	99	HHHHH	XXXXX

- 实验总结：

对数据库的操作往往是一个程序中极其小的一部分，所以我们要学会在高级程序语言中能够实现对数据库实现增删改查，我们可以使用 ODBC , JDBC ,嵌入式 SQL 三种途径来进行实现。但是如今对于大部分情况很少会使用嵌入式 SQL。对于 ODBC 是适用于 C/C++语言，不太适合于 Java,例如因为 c 有指针，但 java 没有指针。但是仍然可以通过接口的方式使用 ODBC，但这样其实还是不好，不如直接使用 JDBC，完全由 Java 语言书写的驱动程序。通过本次实验让我学到了如何在 java 语言中使用 JDBC 去连接数据库，然后写出最基础的增删改查操作，在对其进行封装，使得能有一些更加便捷的方法，比如本次实验的查询线路，插入数据等等。在实验中也遇到了一些问题，比如说由于 sql 语句要采用字符串的拼接，所以常常会导致 sql 语法错误等。其实本次实验就已经比较相似于后端的开发，和他不同的仅仅是没有获得客户端的请求，已经对于客户端请求的处理。总的来说本次实验收获颇丰。