

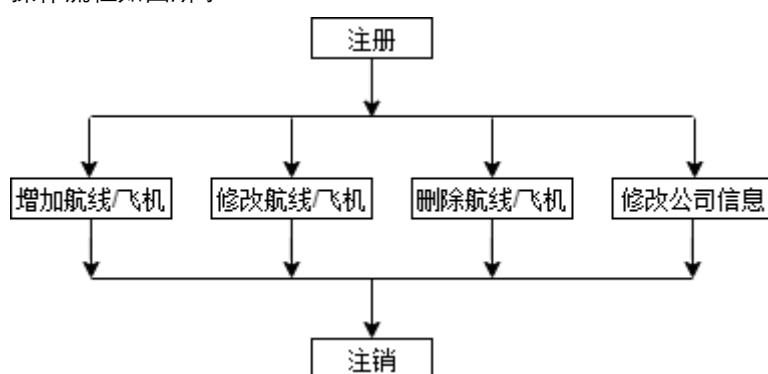
# 实验报告 第六次实验

## Hollow Man

### 一、需求分析

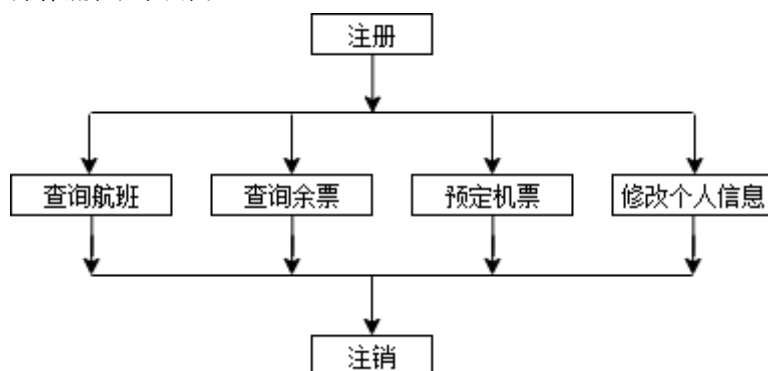
#### 1) 航空公司

操作流程如图所示：



#### 2) 客户

操作流程如图所示：



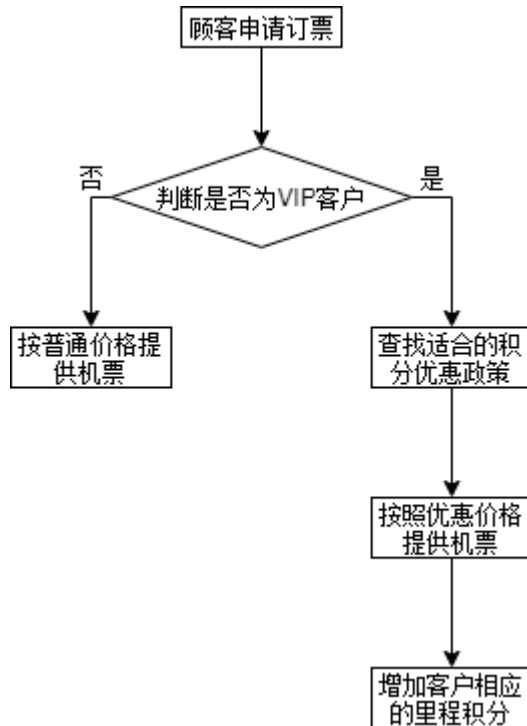
#### 3) 机场

机场需要根据航空公司提供的有关航线和飞机，安排航班以及航班的机票和客户的登机。如果航班出现晚点或者突发情况，要记录并且向有关方发送信息，对特殊 VIP 客户记录其消费信息，提供相应优惠。

#### 4) 客户订票

首先由客户提出订票申请。并由机场管理航班的机票。对于特殊 VIP 客户，出给予票价优惠以外，还要累计飞行里程。订票以后需要判断是否超员。这些情形需要客户资料，航班资料以及由航空公司提供的航线（里程）和飞机（座位数）资料等数据支持。

操作流程如图所示：

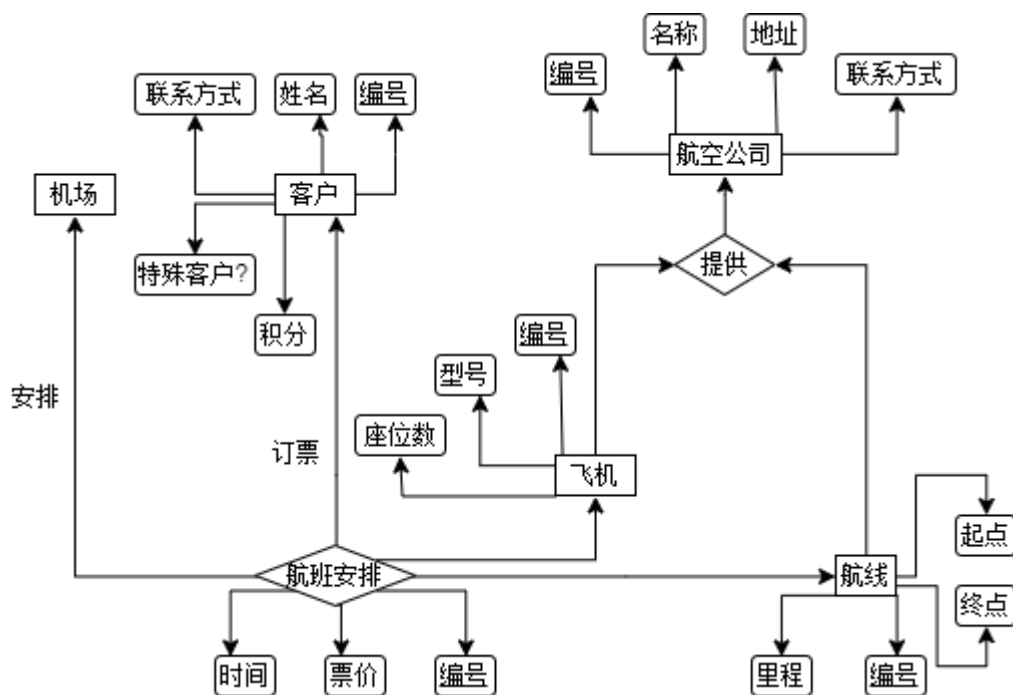


## 二、模型设计

数据库需要包含的信息为：

1. 航空公司信息
2. 客户信息
3. 飞机信息
4. 航线信息
5. 航班信息
6. 订票信息
7. 特殊 VIP 客户的积分

可用 E/R 模型表述，如下：



### 三、逻辑设计

通过 E/R 模型，将其转化为如下关系：

1. Airline(AID, Name, Addr, Cont)
2. Customer(CID, Name, Cont, IsSpec, Points)
3. Plane(PID, Type, SeatsNum, A/D)
4. Line(LID, SPosition, EPosition, Distance, A/D)
5. Flight(FID, PID, LID, Ftime, Price)
6. BookTicket(BID, FID, CID, Pay)

每个关系模式的主键用下划线标出，外键用斜体标出

### 四、优化

为了提高在表中搜索元组的速度，我们建立基于主键的索引：

1. Airline(AID)
2. Customer(CID)
3. Plane(PID)
4. Line(LID)
5. Flight(FID)
6. BookTicket (BID)

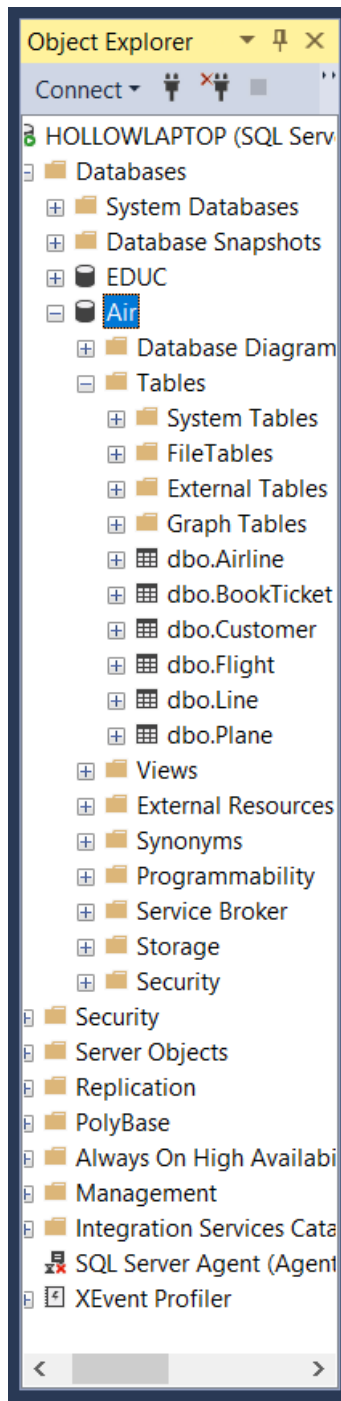
### 五、SQL 实现

首先建表：

```
CREATE TABLE Airline(--航空公司
AID varchar(20) PRIMARY KEY,--航空公司编号
Name varchar(100) NOT NULL,--名称
Addr varchar(100),--地址
Cont varchar(200)--联系方式
```

```
);  
CREATE TABLE Customer(--客户  
    CID varchar(20) PRIMARY KEY,--客户编号  
    Name varchar(50) NOT NULL,--姓名  
    Cont varchar(200),--联系方式  
    IsSpec char(1) NOT NULL DEFAULT 'N',--特殊客户标记  
    Points int NOT NULL DEFAULT 0--积分  
);  
CREATE TABLE Plane(--飞机  
    PID varchar(20) PRIMARY KEY,--飞机编号  
    Type varchar(50) NOT NULL,--型号  
    SeatsNum int NOT NULL,--座位数  
    AID varchar(20),CONSTRAINT FK_PLANE_AID  
        FOREIGN KEY(AID) REFERENCES Airline(AID)--所属航空公司  
);  
CREATE TABLE Line(--航线  
    LID varchar(20) PRIMARY KEY,--航线号  
    SPosition varchar(100) NOT NULL,--起点地址  
    EPosition varchar(100) NOT NULL,--终点地址  
    Distance real NOT NULL,--里程  
    AID varchar(20),CONSTRAINT FK_LINE_AID  
        FOREIGN KEY(AID) REFERENCES Airline(AID)--所属航空公司  
);  
CREATE TABLE Flight(--航班  
    FID varchar(20) PRIMARY KEY,--航班号  
    Ftime datetime NOT NULL,--飞行时间  
    PID varchar(20), CONSTRAINT FK_FLIGHT_PID  
        FOREIGN KEY(PID) REFERENCES Plane(PID),--执飞飞机编号  
    LID varchar(20),CONSTRAINT FK_FLIGHT_LID  
        FOREIGN KEY(LID) REFERENCES Line(LID),--航线号  
    Price real NOT NULL--票价  
);  
CREATE TABLE BookTicket(--售票  
    BID int IDENTITY(1,1) PRIMARY KEY,--交易号  
    FID varchar(20),CONSTRAINT FK_BOOKTICKET_FID  
        FOREIGN KEY(FID) REFERENCES Flight(FID),--航班号  
    CID varchar(20),CONSTRAINT FK_BOOKTICKET_CID  
        FOREIGN KEY(CID) REFERENCES Customer(CID),--客户编号  
    Pay real NOT NULL--已付金额  
);
```

执行可见表创建成功:



```
CREATE TABLE Airline(--航空公司
AID varchar(20) PRIMARY KEY,--航空公司编号
Name varchar(100) NOT NULL,--名称
Addr varchar(100),--地址
Cont varchar(200)--联系方式
);
CREATE TABLE Customer(--客户
CID varchar(20) PRIMARY KEY,--客户编号
Name varchar(50) NOT NULL,--姓名
Cont varchar(200),--联系方式
IsSpec char(1) NOT NULL DEFAULT 'N',--特殊客户标识
Points int NOT NULL DEFAULT 0--积分
);
CREATE TABLE Plane(--飞机
PID varchar(20) PRIMARY KEY,--飞机编号
Type varchar(50) NOT NULL,--型号
SeatsNum int NOT NULL,--座位数
AID varchar(20),CONSTRAINT FK_PLANE_AID
FOREIGN KEY(AID) REFERENCES Airline(AID)--所
);
--CREATE TABLE Line(--航线
--LID varchar(20) PRIMARY KEY,--航线编号
--SPosition varchar(20) NOT NULL,--起点位置
--EPosition varchar(20) NOT NULL,--终点位置
--Distance int NOT NULL,--距离
--AID varchar(20) NOT NULL,--所属航空公司
--CONSTRAINT FK_Line_AID FOREIGN KEY(AID) REFERENCES Airline(AID)
--);
```

91 %

Messages

Commands completed successfully.

Completion time: 2019-11-22T11:01:42.2361023+08:00

然后分别定义操作:

1. 航空公司操作

1) 注册

```
insert into Airline(AID,Name,Addr,Cont)
values('#AID','#Name','#Addr','#Cont');
```

这里的#AID 必须是唯一的, 否则不可以插入到数据库中。

2) 删除飞机

```
delete from Airline where(AID='#AID');
```

3) 修改公司信息

```
update Airline
set Name='#Name',Addr='#Addr',Cont='#Cout' where(AID='#AID');
```

4) 增加飞机

```
insert into Plane(PID,Type,SeatsNum,AID)
values('#PID','#Type','#SeatsNum','#AID');
```

5) 删除飞机

```
delete from Plane where(PID='#PID');
```

6) 修改飞机

```
update Plane
set Type='#Type',SeatsNum='#SeatsNum' where(PID='#PID');
```

7) 增加航线

```
insert into Line(LID,SPosition,EPosition,Distance,AID)
values('#LID','#SPosition','#EPosition','#Distance','#AID');
```

8) 删除航线

```
delete from Line where(LID='#LID');
```

9) 修改航线

```
update Line
set SPosition='#SPosition',EPosition='#EPosition',Distance='#Distance'
where(PID='#PID');
```

## 2. 客户操作

### 1) 注册

```
insert into Customer(CID,Name,Cont)
values('#CID','#Name','#Cont');
```

这里的#CID 必须是唯一的，否则不可以插入到数据库中。

### 2) 注销

```
delete from Customer where(CID='#CID');
```

### 3) 修改个人信息

```
update Customer
set Name='#Name',Cont='#Cout' where(CID='#CID');
```

### 4) 订票

订票的过程一定要做成事务，因为订票的操作应该同时只能有一个人进行。

```
create proc Book_Ticket
@FID varchar(20),
@CID varchar(6)
as
declare @TransName varchar(20)
select @TransName='Book_Ticket'
begin transaction @TransName
declare @booked int,@seat int, @IsSpec char(1)
declare @distance real,@discount real,@dist real,@price real

-- 查看是否为特殊客户，如果不是，票价不打折扣
-- 否则如果客户累计航程超过10万公里，票价打九折；超过20公里，打八折。
-- 下面的程序用来计算折扣
select @IsSpec=IsSpec,@distance=Points
from Customer where CID=@CID
select @discount=1
if @IsSpec='Y'
begin
if @distance>200000
select @distance=0.8
else if @distance>100000
select @distance=0.9
end
--- 选择出票价
select @price=Price from Flight where FID=@FID
-- 加入顾客订票信息
insert into BookTicket(FID,CID,Pay)
values(@FID,@CID,@price*@discount)
-- 将顾客的新订票里程的信息累积到用户信息里面
```

```
select @dist=Distance from Line
where LID=(Select LID from Flight where FID=@FID)
update Customer
set Points=Points+@dist where(CID=@CID);
--查看客户订票后, 是否超过可容纳的座位数目, 如果超过, 取消所有操作
select @seat=SeatsNum from Plane
where PID=(select PID from Flight where FID=@FID)
select booked=count(*) from BookTicket where FID=@FID
if @booked>@seat
rollback transaction
else
commit transaction @TransName
go
```

### 3. 机场操作

#### 1) 航班安排

```
insert into Flight(FID,PID,LID,Ftime,Price)
values('#FID','#PID','#LID','#Ftime','#Price');
```

#### 2) 通知客户航班变化

```
select [Name], Cont from Customer
where CID IN
(select CID from BookTicket where FID='#FID');
```