关于航空公司售票系统的调研

一、 调研报告

（一）、系统综述

航空公司售票系统，是满足用户查询、预定航班需求，实现用户信息和航班信息管理，为用户提供高效服务的系统。其主要功能有预定、取消航班，查询航班，查询执飞飞机信息，为常旅客提供积分累计和个性化服务等功能。

在数据库方面，主要存储航班信息、乘客信息、订单数据等。详细分析一下，，我们需要存储：机场、航班、订单、飞机、用户、常旅客，6 个实体。其中包含子类（常旅客为用户子类）。需要描述机场代码、位置，航班起飞到达时间、航班号，机型、机龄，还有用户的各类信息。

（二）、实体分析

1. 航班基本信息。

   包括航班号（唯一标识，例如 CA1234）、出发地（如北京首都国际机场）、目的地（如上海浦东国际机场）、出发时间（精确到分钟，如 2025-06-01 08:30:00）、到达时间、飞行时长。航班机型：记录航班所使用的飞机型号（如波音 737、空客 A320 等）。

2. 乘客信息

   基本身份信息：乘客姓名、性别、出生日期、身份证号码（或其他有效证件号码，如护照号码）、联系方式（电话号码、电子邮箱）。

   常旅客信息：对于加入航空公司常旅客计划的乘客，记录其会员编号、会员等级（如银卡、金卡、白金卡）、累计飞行里程、积分余额等信息。

3. 订单信息

   订单基本信息：订单编号（唯一标识）、乘客身份证号码（关联乘客信息）、航班号（关联航班信息）、购票时间、支付状态（已支付、未支付、支付失败）。

   订单详情：包括所购舱位、座位编号、票价金额、税费金额、实际支付金额等

   退改签记录：若订单发生退改签操作，记录退改签时间、操作类型（退票、改签）、改签后的航班信息（若为改签）、退改签费用等。

（三）、操作需求

1. 数据查询

乘客可根据出发地、目的地、出行日期等条件查询符合条件的航班信息，系统需快速返回航班列表及相关票价、座位情况等信息。航空公司员工可查询特定航班的售票情况，包括已售座位数、剩余座位数、不同舱位的销售比例等。常旅客可查询自己的会员信息、积分余额、飞行记录等。

2. 数据插入

当有新的航班计划时，管理员需将航班信息插入数据库，包括航班基本信息、机型、座位布局、票价等。乘客进行机票预订时，系统需将订单信息、乘客信息（若为新乘客）及支付信息插入相应的数据库表中。

3. 数据更新

航班状态发生变化时（如延误、取消），需及时更新航班信息表中的航班状态字段。乘客进行退改签操作后，需更新订单信息表中的相关字段，如舱位、座位编号、支付状态（若涉及费用调整）、退改签记录等。常旅客累积飞行里程或使用积分后，需更新常旅客信息表中的累计飞行里程和积分余额字段。

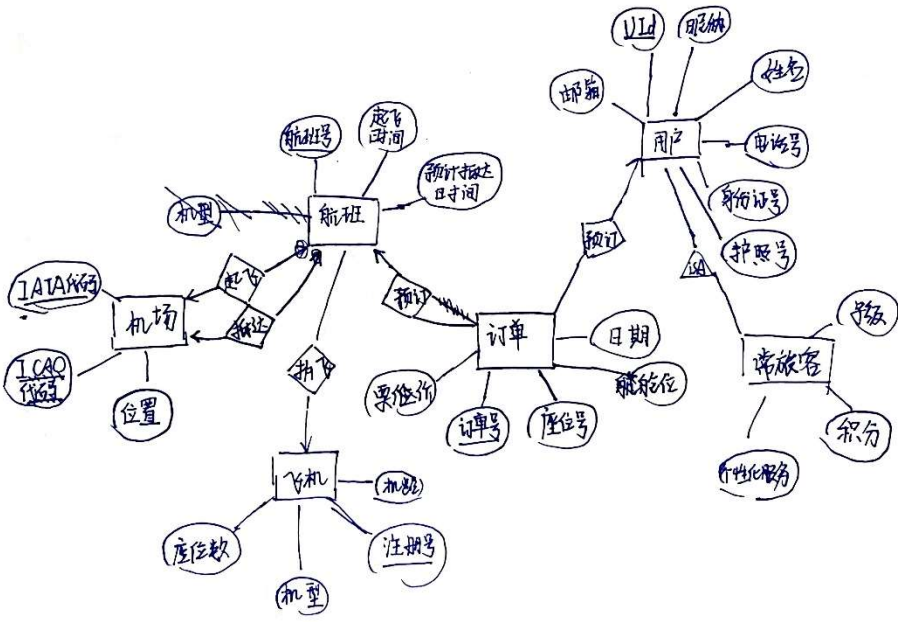
4. 数据删除

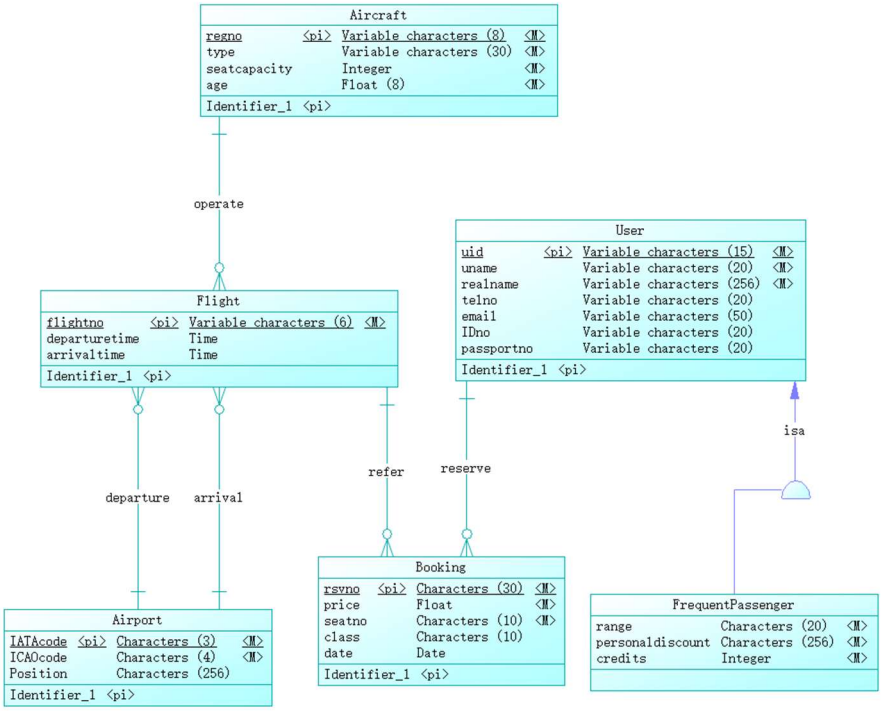
对于已过期且不再需要保留的航班信息（如历史航班），在满足一定的保留期限后，管理员可将其从数据库中删除。对于已完成且无后续业务关联的订单（如已退票且超过一定时间），可进行删除操作，但需确保数据的删除不会影响数据分析和统计。

（四）、实体关系分析

用户和常旅客之间是父类与子类的关系。用户预定订单，一个用户可以预定多个订单，且一个订单有且只有一个用户预定。订单信息用于航班，一个订单有且只有一个对应航班，航班内有多个订单。航班由飞机执飞，飞机可以不执行航班，但航班必须有且只有一个飞机执行。航班出发到达参照机场信息，航班与出发/到达机场是多对一关系。

二、绘制 ER 图





三、ER 图转换成关系模式

1. 按照逻辑自己生成

User(uid(主键)， uname, realname, telno, email, IDno, passportno);

FrequentPassenger(uid(主键), range, personaldiscount, credits);使用 ER 模式存储

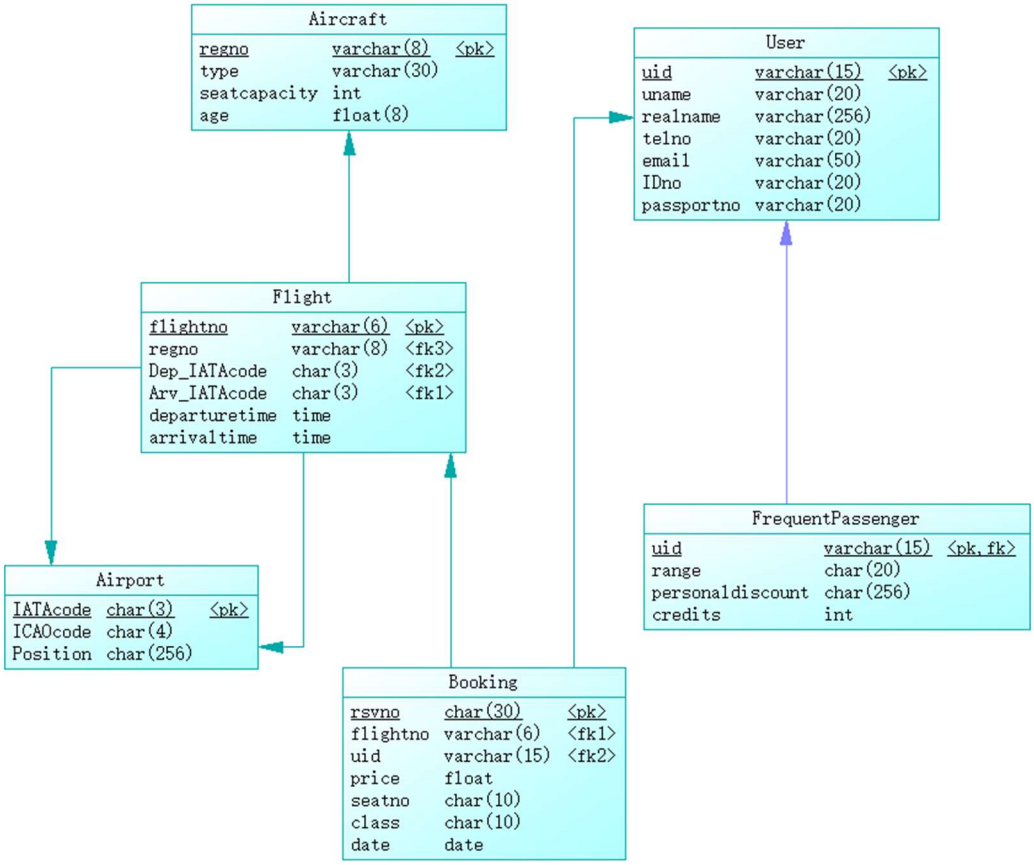
Booking(resvno(主键), flightno(外键， 参照 Flight),uid(外键， 参照 User), price, seatno, class, date);

Flight(flightno(主键), regno(外键， 参照 Aircraft), IATAcode(外键， 参照 Airport), Air_IATAcode(外键， 参照 Airport), departuretime, arrivaltime);

Aircraft(regno(主键), type, seatcapacity, age);

Airport(IATAcode(主键), ICAOcode, Position):

2. powerdisigner 自动生成



四、用 SQL 语句创建关系

1.根据自己的关系模式手动生成

```
create table Aircraft(
    regno                varchar(8) not null,
    type                 varchar(30) not null,
    seatcapacity         int not null,
    age                  float(8) not null,
    primary key (regno));

create table Airport(
    IATAcode             char(3) not null,
    ICAOcode             char(4) not null,
    Position             char(256),
    primary key (IATAcode));

create table Booking(
```

```
    rsvno                char(30) not null,
    flightno             varchar(6) not null,
    uid                  varchar(15) not null,
    price                float not null,
    seatno               char(10) not null,
    class                char(10),
    date                 date,
    primary key (rsvno));

create table Flight(
    flightno             varchar(6) not null,
    regno                varchar(8) not null,
    Dep_IATAcode         char(3) not null,
    Arv_IATAcode         char(3) not null,
    departuretime        time,
    arrivaltime          time,
    primary key (flightno));

create table User(
    uid                  varchar(15) not null,
    uname                varchar(20) not null,
    realname             varchar(256) not null,
    telno                varchar(20),
    email                varchar(50),
    IDno                 varchar(20),
    passportno           varchar(20),
    primary key (uid));

create table FrequentPassenger(
    uid                  varchar(15) not null,
    range                char(20) not null,
    personaldiscount     char(256) not null,
    credits              int not null,
    primary key (uid));
```

2.Power Designer 自动生成

```
/*==============================================================*/
/* DBMS name:      MySQL 5.0                                    */
/* Created on:     2025/4/10 22:42:52                           */
/*==============================================================*/


drop table if exists Aircraft;

drop table if exists Airport;

drop table if exists Booking;

drop table if exists Flight;

drop table if exists FrequentPassenger;

drop table if exists User;

/*==============================================================*/
/* Table: Aircraft                                              */
/*==============================================================*/
create table Aircraft
(
   regno                varchar(8) not null,
   type                 varchar(30) not null,
   seatcapacity         int not null,
   age                  float(8) not null,
   primary key (regno)
);

/*==============================================================*/
/* Table: Airport                                               */
/*==============================================================*/
create table Airport
(
   IATAcode             char(3) not null,
   ICAOcode             char(4) not null,
   Position             char(256),
   primary key (IATAcode)
);
```

```
/*==============================================================*/
/* Table: Booking                                               */
/*==============================================================*/
create table Booking
(
   rsvno                char(30) not null,
   flightno             varchar(6) not null,
   uid                  varchar(15) not null,
   price                float not null,
   seatno               char(10) not null,
   class                char(10),
   date                 date,
   primary key (rsvno)
);

/*==============================================================*/
/* Table: Flight                                                */
/*==============================================================*/
create table Flight
(
   flightno             varchar(6) not null,
   regno                varchar(8) not null,
   Dep_IATAcode         char(3) not null,
   Arv_IATAcode         char(3) not null,
   departuretime        time,
   arrivaltime          time,
   primary key (flightno)
);

/*==============================================================*/
/* Table: FrequentPassenger                                     */
/*==============================================================*/
create table FrequentPassenger
(
   uid                  varchar(15) not null,
   range                char(20) not null,
   personaldiscount     char(256) not null,
   credits              int not null,
   primary key (uid)
);
```

```
/*==============================================================*/
/* Table: User                                                  */
/*==============================================================*/
create table User
(
   uid                  varchar(15) not null,
   uname                varchar(20) not null,
   realname             varchar(256) not null,
   telno                varchar(20),
   email                varchar(50),
   IDno                 varchar(20),
   passportno           varchar(20),
   primary key (uid)
);

alter table Booking add constraint FK_refer foreign key (flightno)
      references Flight (flightno);

alter table Booking add constraint FK_reserve foreign key (uid)
      references User (uid);

alter table Flight add constraint FK_arrival foreign key (Dep_IATAcode)
      references Airport (IATAcode);

alter table Flight add constraint FK_departure foreign key (Arv_IATAcode)
      references Airport (IATAcode);

alter table Flight add constraint FK_operate foreign key (regno)
      references Aircraft (regno);

alter table FrequentPassenger add constraint FK_isa foreign key (uid)
      references User (uid);
```

五、查询样例

1. 单表查询示例：查询年龄大于 5 年的飞机

```
SELECT *

FROM Aircraft

WHERE age > 5;
```

2.多表连接查询示例：查询航班号为 CX9100 的执飞飞机机型；

```
SELECT type
FROM Aircraft natural join Flight
WHERE flightno='CX8100'
```

3. 多表嵌套查询示例：查询预订了特定航班（如 CA123）的乘客姓名

```
SELECT realname
FROM User
WHERE uid IN ( SELECT uid
    FROM Booking
    WHERE flightno = 'CA123');
```

4.EXISTS 查询示例：查询所有订过航班的用户的真实姓名和邮箱

```
SELECT u.realname, u.email
FROM User u
WHERE EXISTS (
    SELECT 1
    FROM Booking b
    WHERE b.uid = u.uid
);
```

5.聚合操作查询示例：查询姓名为张三的所有订单数量和总价

```
SELECT COUNT(*) AS order_count, SUM(price) AS total_price
FROM Booking
WHERE uid = (
    SELECT uid
    FROM User
    WHERE realname = '张三'
);
```

六、分析与比较

经过上述操作，我发现使用 power designer 自动生成的模式与自己手动转化出来的模式在整体含义上没有区别。由于在 ER 关系图中，实体间的关系都没有属性，因此无需对特例进行特殊转换，对于 sql 的语句生成也没有影响。

对于生成的 sql 语句，自动生成的有一些附加语句，例如：drop table if exists Aircraft;这句话指如果存在 Aircraft 的表，把它删除，防止重复情况影响下一步建表。由于我在使用 power design 时详细分析了属性取值问题，在一些属性下勾选了非空“Mandatory”选项，因此部分属性后有 not null。在一开始手写 sql 语句时，我并没有考虑，在后来才添加上。在末尾还有诸如 alter table Booking add constraint FK_refer foreign key (flightno) references Flight (flightno); 的语句。它的意思是给 Booking 表添加一个外键约束，确保预订信息中的航班编号 flightno 必须存在于 Flight 表中。这是为了满足数据库的参照完整性，防止外键属性没有在另一个表中出现。