

# 数据库 lab2 设计文档

中国科学院大学

陈灿宇, 宋鹏皓, 黄彦杰

2018.11.22

## 1 需求分析 (lab2 说明)

### (1) 记录列车信息

需要记录车站: 始发站, 中间经停站, 终点站, 最长 20 字符的站名, 每站的发车时间和到达时间, 不同座位的票价等信息。

### (2) 记录列车座位情况

每次列车、每类座位、每天有 5 个。

### (3) 记录乘客信息

每位乘客在使用前需要注册, 登记姓名、身份证、手机号、信用卡、用户名等信息。并给出后续链接。

### (4) 支持查询具体车次

根据车次序号、日期显示具体车次的静态信息以及动态信息。并给出预订功能链接。

### (5) 查询两地之间的车次

输入出发和到达城市、日期和时间, 显示两地之间的直达列车和余票信息 (按价格升序前十) 两地之间换乘一次的列车组合和余票信息 (按价格升序前十)。换乘地必须是同一城市, 换乘地是同一车站, 那么  $1 \text{ 小时} \leq \text{换乘经停时间} \leq 4 \text{ 小时}$ , 如果换乘地是同城的不同车站, 那么  $2 \text{ 小时} \leq \text{换乘经停时间} \leq 4 \text{ 小时}$ 。发车时间  $\geq$  给定的出发时间。[换乘未实现]

### (6) 预订车次座位

每个车次显示车次、出发日期、出发时间、出发车站、到达日期、到达时间、到达车站、座位类型、本次车票价。点击确认生成, 取消返回。

### (7) 查询订单和删除订单

给定出发日期范围, 显示历史订单列表及相关信息。

### (8) 管理员登录

## 2 ER 图

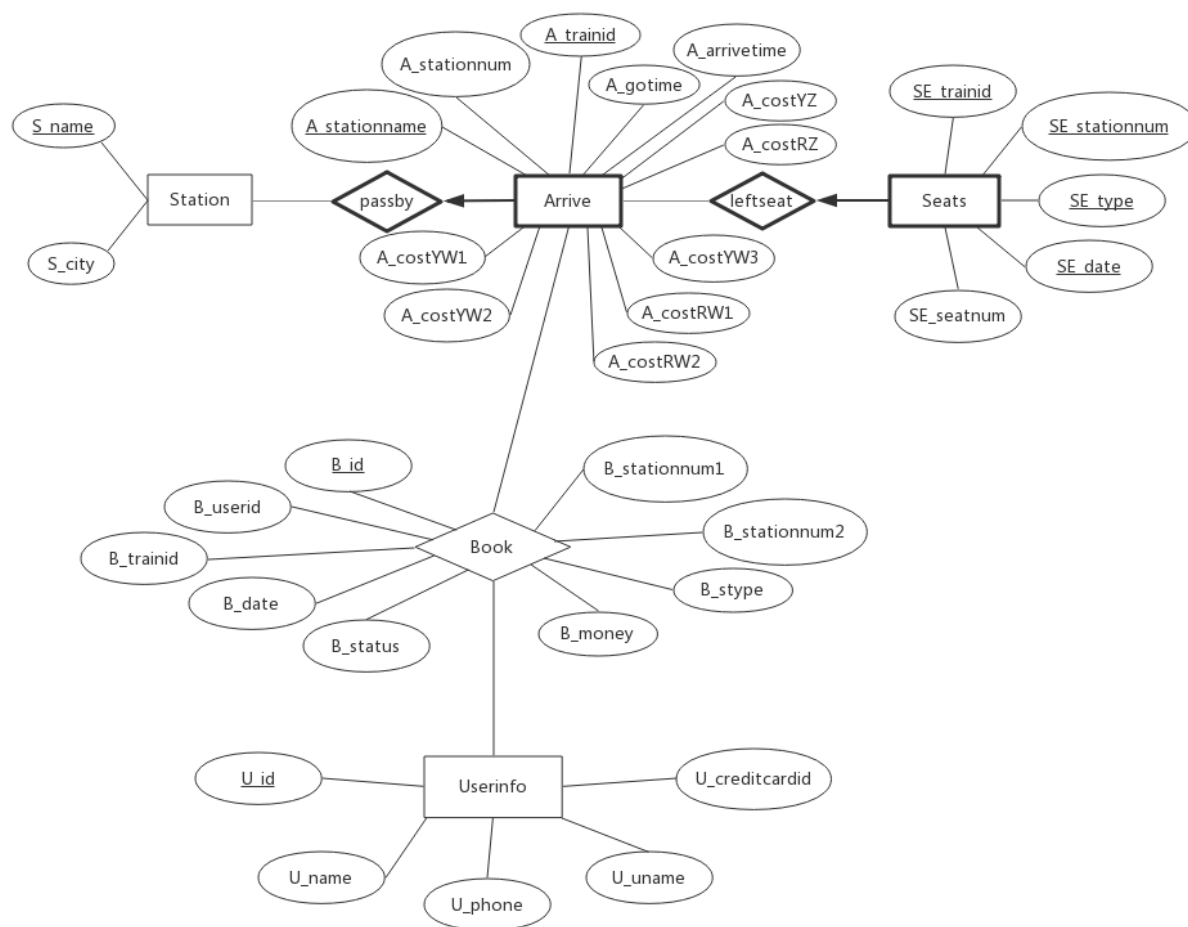


图 1: ER 图

我们一共设立了四个实体集和一个联系集。

(1) 实体 Station 用于存储车站的信息，其中有两个属性，name 是车站名，city 是车站所在的城市名。

(2) 实体 Arrive 是一个弱实体，它存储的是一辆列车经过一个车站的信息，所以它的主实体是 Station，因为有了车站才有这个实体的存在。这个实体一共有 12 个属性。trainid 是列车号，stationname 和 stationnum 分别是经过的车站的名字还有车站在这趟列车的次序。arrivetime 和 gotime 分别表示的是该列车到达该站的时间和离开该站的时间。其余的属性则表示的是从始发站到当前站的各种座位的价格。

(3) 实体 Seats 也是一个弱实体，它表示一辆列车到一个车站时车上的座位信息。由于 Seats 只有在列车到站时才有值，所以它的主实体是 Arrive。Seats 实体一共有 5 个属性。trainid 和 stationnum 分别表示列车和车站的信息。type 则是座位的类型，有硬/软座，硬卧（上/中/下），软卧（上/下）这几种类型。date 表示日期，而 seatnum 则表示这种类型的座位还剩余的数量。

(4) 实体 Userinfo 存储用户的信息，它有 5 个属性。id 表示用户的身份证号，name 和 uname 分别是姓名和用户名，phone 则是电话号码，剩下的 creditcardid 则是信用卡号。

(5) 除此之外还有一个联系集 Book，它存储所有的订单信息。其一共有 9 个属性，包括订单号 id，用户 id，列车 id，日期 date，订单状态 status，出发站 stationnum1，终点站 stationnum2，座位类型 stype 以及花费 money。

### 3 范式细化与关系模式

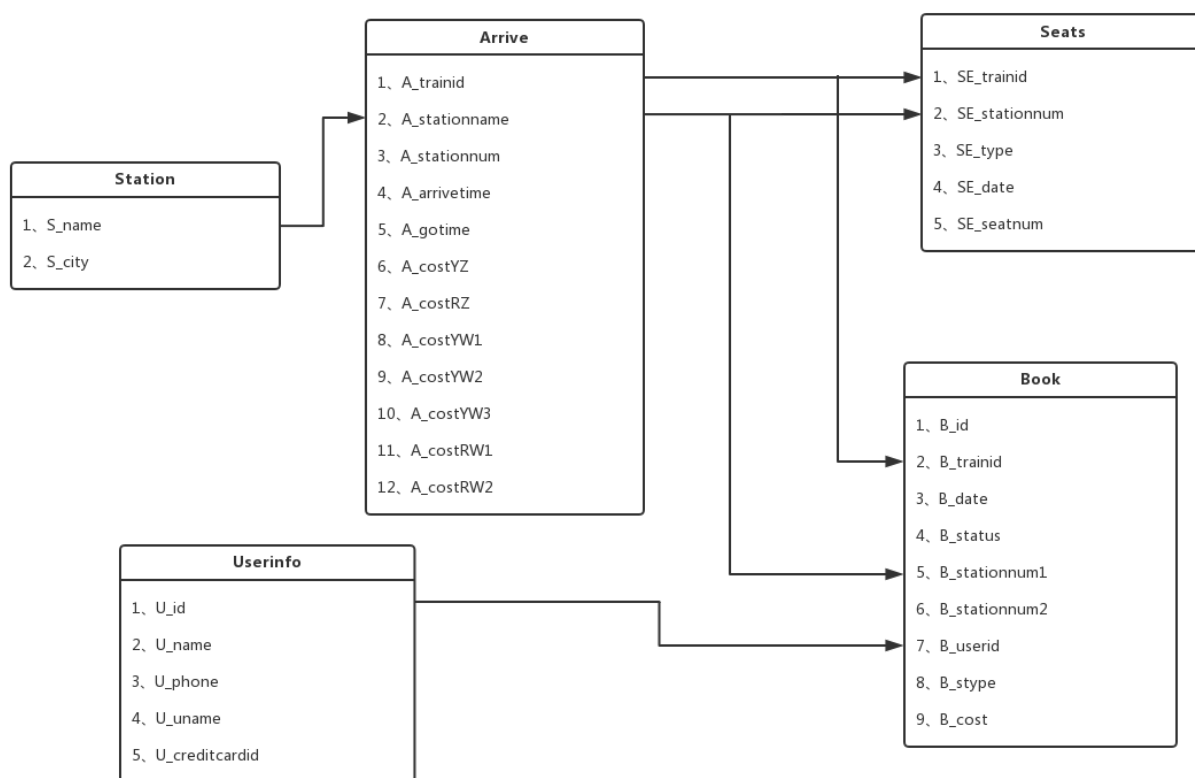


图 2: 关系表及其相互关联

分析每个 table 后可以得到范式细化的结果。

首先考虑部分依赖，部分依赖是存在一个非键属性函数依赖于候选键的子集，这表示候选键的子集就已经能确定这个非键属性了。要消除部分依赖，就是让每一个非键属性都必须由候选键里所有的属性共同确定，比如上述 Arrive 实体集中，候选键是 trainid 和 stationname，我们做到的是只有当 trainid 和 stationname 都已经直到的时候才能得出唯一的 arrivetime、gotime 和各种座位价格。所以我们满足了这点，不存在部分依赖，属于 2NF。

而对于非键传递依赖，它与部分依赖的不同仅在于函数依赖的对象不仅包含了候选键的部分，还有非键的属性。但由于我们设计种的非键属性之间不存在依赖或者对应关系，所以也不存在非键传递依赖，故属于 3NF。

最后是关于键属性的函数依赖，它的情况是存在一个候选键的子集函数依赖与包含了另一个候选键子集的集合。但由于我们设计的每张表种候选键的属性都是各自独立的，所以很容易就可以得出不存在对于键属性的函数依赖。综上所述，我们所设计的模型满足 BCNF。

根据以上讨论，我们的建表方案如下：

---

```
1 create table Station
2 (
3     S_Name          varchar(20) primary key,
4     S_City          varchar(20) not null
5 );
```

---

```
1 create table Arrive
2 (
3     A_TrainId       varchar(6) not null,
4     A_StationName   varchar(20) not null,
5     A_StationNum    integer not null,
6     A_ArriveTime    time,
7     A_GoTime        time,
8     A_CostYZ        float,
9     A_CostRZ        float,
10    A_CostYW1        float,
11    A_CostYW2        float,
12    A_CostYW3        float,
13    A_CostRW1        float,
14    A_CostRW2        float,
15    primary key (A_TrainId, A_StationNum),
16    foreign key (A_StationName) references Station(S_Name)
17 );
```

---

```
1 create table UserInfo
2 (
3     User_Id         char(18) primary key,
4     U_Name          varchar(20) not null,
5     U_Phone         char(11) not null,
6     U_UName         varchar(20) not null,
7     U_CreditCardId  char(16) not null
8 );
```

---

```
1 create table Book
2 (
3     B_Id            SERIAL primary key,
4     B_UserId        char(18) not null,
5     B_TrainId       varchar(6) not null,
6     B_Date          date not null,
7     B_StationNum1   integer not null,
8     B_StationNum2   integer not null,
9     B_SType         seaS_Type not null,
10    B_Money          integer not null,
11    B_Status         status_type not null,
12    foreign key (B_TrainId, B_StationNum1) references Arrive(A_TrainId, A_StationNum),
13    foreign key (B_UserId) references UserInfo(User_Id)
14 );
```

---

---

```

1 create table Seats
2 (
3     S_TrainId      varchar(6) not null,
4     S_PStationNum  integer not null,
5     S_Type         seaS_Type not null,
6     S_Date         date not null,
7     S_SeatNum      integer not null,
8     primary key (S_TrainId, S_PStationNum, S_Type, S_Date),
9     foreign key (S_TrainId, S_PStationNum) references Arrive(A_TrainId, A_StationNum)
10 );

```

---

## 4 SQL 查询语句的模板

下面给出本次试验比较复杂的查询语句的模板, 下面的查询语句基本都需要前端配合才能完成相应的工作。

### (1) 给定某车次以及某两站站序, 查询余票

由于 Seats 表是存储的是某天一个车次相邻两站票被订购的次数。未被订就不存在记录, 所以下面查询方式需要配合前端操作才能得到余票还有多少。注意 'K474'、1、13、'2017-12-01' 是参数。

---

```

1 with T1(T1_Type, T1_SeatNum) as
2 (SELECT S_Type, S_SeatNum
3     FROM Seats
4     WHERE S_TrainId = '$trainid'
5           AND S_PStationNum >= 1
6           AND S_PStationNum < $sta_num
7           AND S_Date = '$thedata')
8
9 SELECT T1_Type, MAX(T1_SeatNum)
10 FROM T1
11 GROUP BY T1_Type;

```

---

### (2) 给定两地和时间, 查询所有可以乘坐的列车 (按硬座票价升序)

下面的查询方法没有考虑余票, 因此需要前端进一步配合。'天津'、'苏州' 和 '00:00' 是可以配置的参数。

---

```

1 with S1(S1_TrainId, S1_StationNum) as
2 (select Passby.P_TrainId, Passby.P_StationNum
3     from Passby, Station
4     where Passby.P_StationName = Station.S_Name
5           and Station.S_City = '天津'),
6
7 S2(S2_TrainId, S2_StationNum) as
8 (select Passby.P_TrainId, Passby.P_StationNum
9     from Passby, Station

```

```
10     where Passby.P_StationName = Station.S_Name
11         and Station.S_City = '苏州'),
12
13 T1(T1_TrainId) as
14 (select S1.S1_TrainId
15     from S1, S2
16     where S1.S1_TrainId = S2.S2_TrainId
17         and S1.S1_StationNum < S2.S2_StationNum
18 ),
19
20 T3(Tp_trainid, Tp_stationname, Tp_stationnum, Tp_arrivetime,
21     Tp_gotime, Tp_moneyyz, Tp_moneyrz, Tp_moneyyw1, Tp_moneyyw2,
22     Tp_moneyyw3, Tp_moneyrw1, Tp_moneyrw2, Tt1_trainid, Ts_name, s_city) as
23 (select *
24     from Passby, T1, Station
25     where Passby.P_TrainId = T1.T1_TrainId
26         and Passby.P_StationName = Station.S_Name
27         and (Station.S_City = '天津'
28             or Station.S_City = '苏州')),
29
30 T4(T4_id, yz, rz, yw1, yw2, yw3, rw1, rw2) as
31 (
32 select Tp_trainid, Max(Tp_moneyyz)-Min(Tp_moneyyz),
33             Max(Tp_moneyrz)-Min(Tp_moneyrz),
34             Max(Tp_moneyyw1)-Min(Tp_moneyyw1),
35             Max(Tp_moneyyw2)-Min(Tp_moneyyw2),
36             Max(Tp_moneyyw3)-Min(Tp_moneyyw3),
37             Max(Tp_moneyrw1)-Min(Tp_moneyrw1),
38             Max(Tp_moneyrw2)-Min(Tp_moneyrw2)
39 from T3
40 group by Tp_trainid
41 ),
42
43 T5(T5_id,T5_StationName, T5_GoTime, T5_StationNum) as
44 (
45 select T1.T1_TrainId, Passby.P_StationName, Passby.P_GoTime, Passby.P_StationNum
46     from T1, Passby, Station
47     where P_TrainId = T1_TrainId
48         and P_StationName = Station.S_Name
49         and Station.S_City = '天津'
50 ),
51
52 T6(T6_id, T6_StationName, T6_ArriveTime, T6_StationNum) as
53 (
54 select T1.T1_TrainId, Passby.P_StationName, Passby.P_ArriveTime, Passby.P_StationNum
55     from T1, Passby, Station
56     where P_TrainId = T1_TrainId
57         and P_StationName = Station.S_Name
58         and Station.S_City = '苏州'
```

```
59 )
60
61 -- order by yz;
62 select DISTINCT
63 T4.T4_id,
64 T5.T5_StationName,
65 T5.T5_GoTime,
66 T6.T6_StationName,
67 T6.T6_ArriveTime,
68 T4.yz,
69 T4.rz,
70 T4.yw1,
71 T4.yw2,
72 T4.yw3,
73 T4.rw1,
74 T4.rw2,
75 T5.T5_StationNum,
76 T6.T6_StationNum
77     from T4, T5, T6
78     where T4.T4_id = T5.T5_id
79           and T6.T6_id = T5.T5_id
80           and T5_GoTime > '00:00'
81
82 order by T4.yz
83 ;
```

---

### (3) 订票退订时, 发生的操作

---

```
1 -- BOOK
2 INSERT INTO
3     book(B_UserId,B_TrainId,B_Date,B_StationNum1,B_StationNum2,B_SType,B_Money,B_Status)
4     VALUES('$uid', '$trainid', '$date', $stnum1, $stnum2, '$seattype', $money, 'normal');
5 -- CANCAL
6 UPDATE book SET b_status = 'cancelled'
7     WHERE b_id = '$bookid';
```

---

同时, 订票和退订时, 对表 TicketInfo 也有操作, 订票时:

---

```
1 select T_SeatNum
2 from TicketInfo
3     where T_TrainId='K474'
4           and T_PStationNum = 1
5           and T_Type = 'YZ'
6           and T_Date = '2017-12-01';
7 -- 前端检查返回结果, 如果为空
8 insert into
9 TicketInfo(T_TrainId , T_PStationNum , T_Type , T_Date , T_SeatNum )
10 values ( 'K474' , 1 , 'YZ' , '2017-12-01' , 1 ) ,
11 ('K474' , 1 , 'YZ' , '2017-12-01' , 2) ;
```

```
12 -- 如果不空[NEW_NUM] <= [NUM] + 1
13 update TicketInfo set T_SeatNum = [NEW_NUM]
14     where T_TrainId='K474'
15         and T_PStationNum = 3
16         and T_Type = ' YZ '
17         and T_Date = ' 2017-12-01 ' ;
```

---

退订时:

---

```
1 select T_SeatNum
2 from TicketInfo
3     where T_TrainId='K474'
4         and T_PStationNum = 1
5         and T_Type = ' YZ '
6         and T_Date = ' 2017-12-01 ' ;
7
8 -- 前端检查返回结果,如果是1
9 delete from TicketInfo
10     where T_TrainId='K474'
11         and T_PStationNum = 1
12         and T_Type = ' YZ '
13         and T_Date = ' 2017-12-01 ' ;
14 -- 否则 [NEW_NUM] <= [NUM] - 1
15 update TicketInfo
16 set T_SeatNum = [NEW_NUM]
17     where T_TrainId='K474'
18         and T_PStationNum = 1
19         and T_Type = ' YZ '
20         and T_Date = ' 2017-12-01 ' ;
```

---

注: 以上 sql 语句除了 createtable 之外, 均嵌入在前端的 php 文件中, 和前端配合完成某个工作。