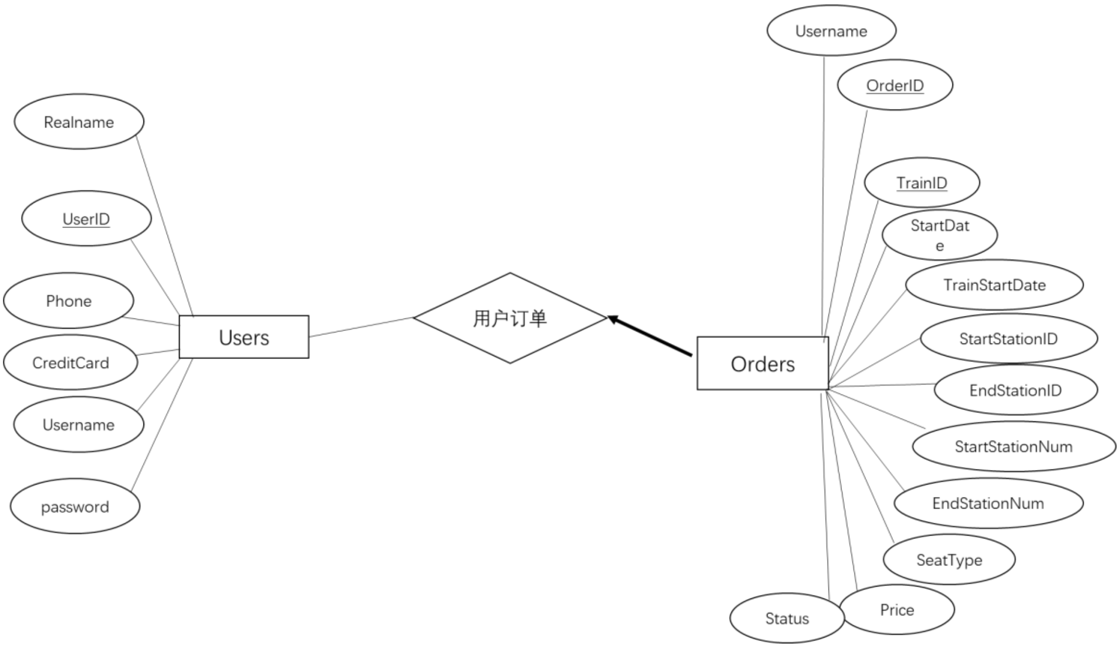
数据库 LAB2 设计文档

江洲钰、姜悦人、陈一元

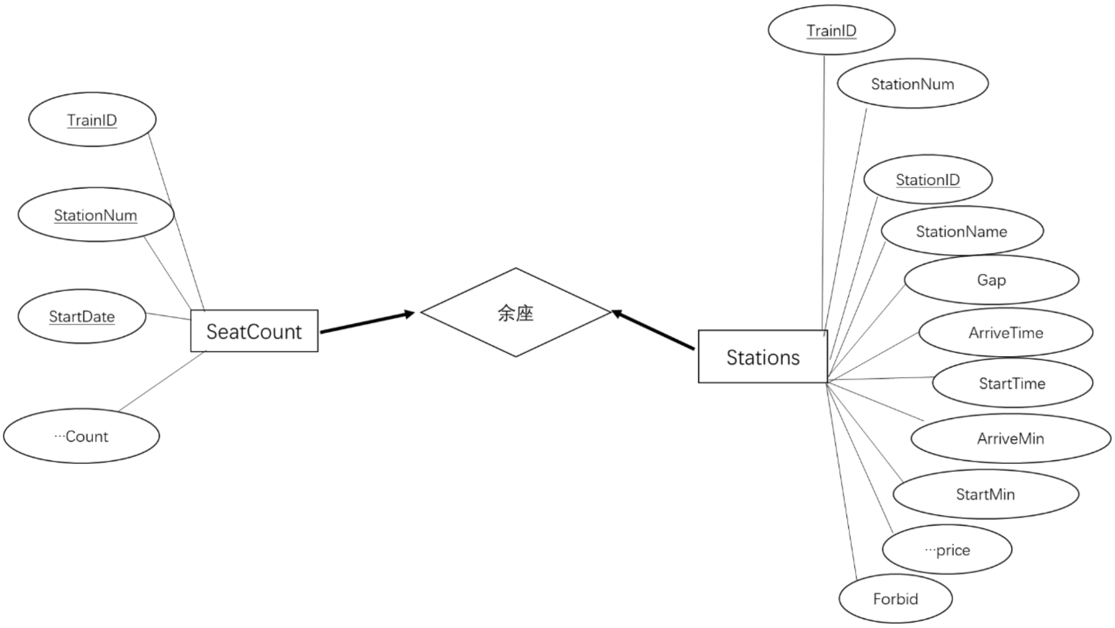
2018.11.18

一、ER 图

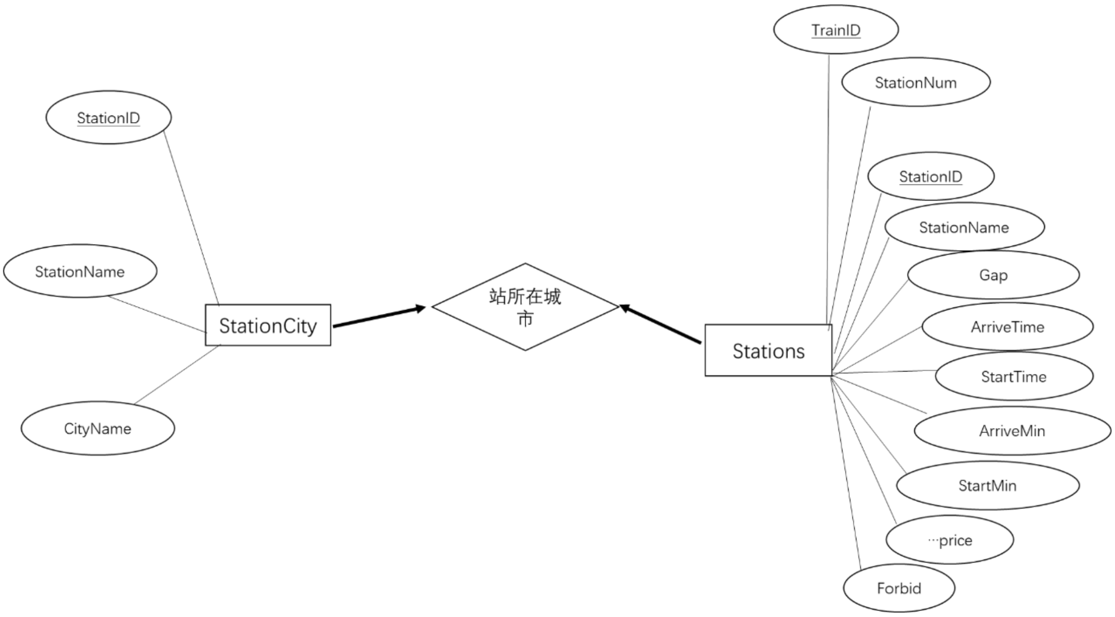
1.用户订单信息



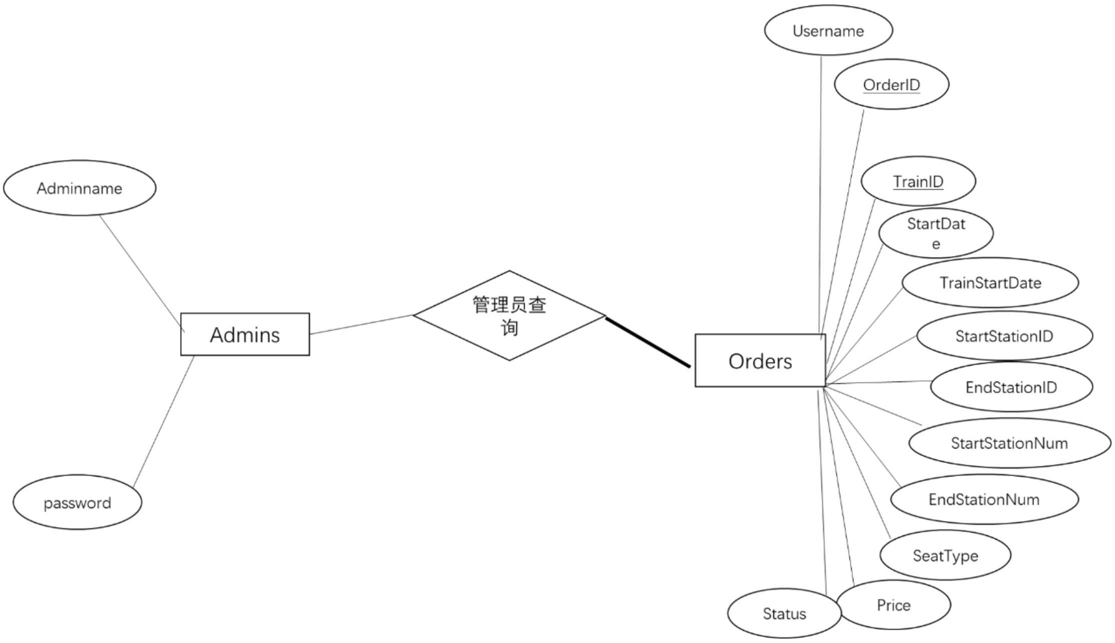
2.火车余座信息



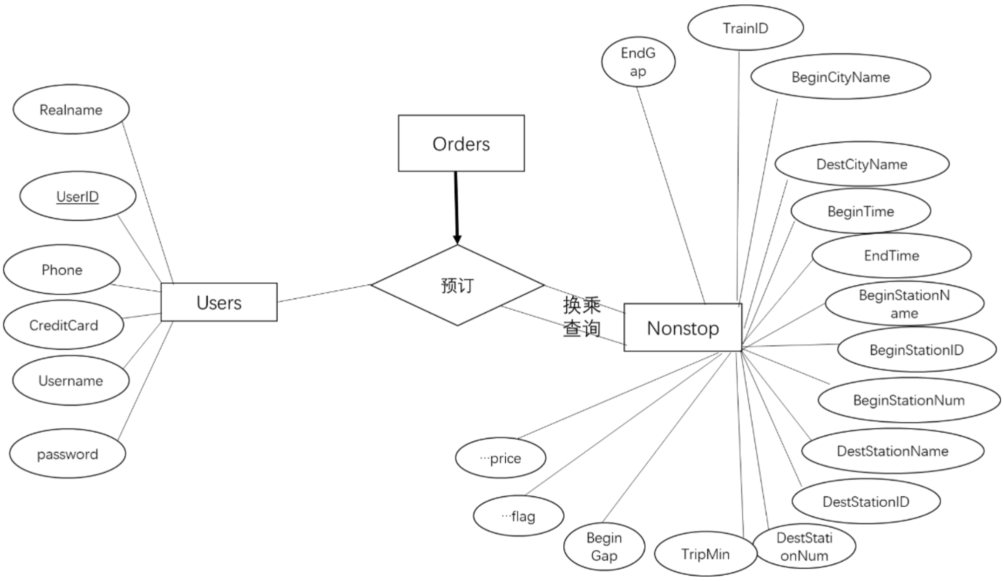
3.站所在城市信息



4. 管理员查询



5. 用户订票

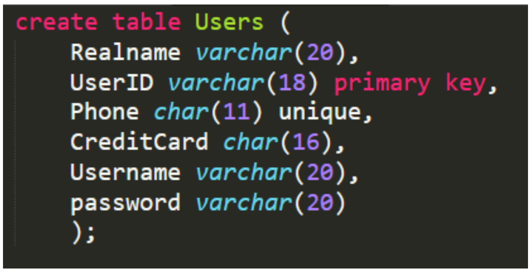


二、关系模式

1. 设计思路

考虑到查询、订票等操作的基本单位都是车站，因此我们选择以站为单位来存储信息，这样也易于做换乘查询和余座统计。

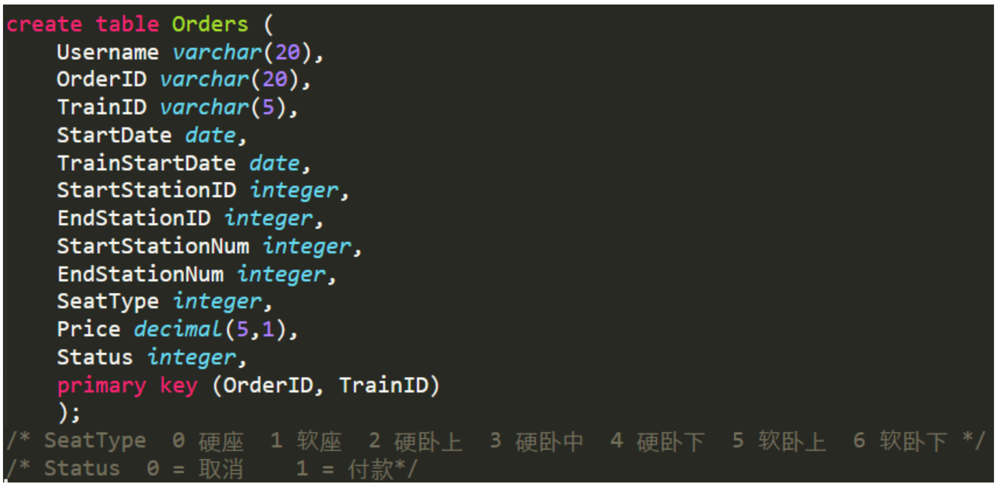
2. 用户



```
create table Users (
    Realname varchar(20),
    UserID varchar(18) primary key,
    Phone char(11) unique,
    CreditCard char(16),
    Username varchar(20),
    password varchar(20)
    );
```

Users 表存储用户的基本信息。其中以身份证号 UserID 为主键。

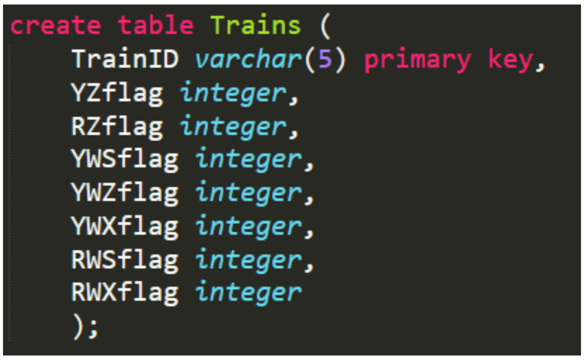
3. 订单



```
create table Orders (
    Username varchar(20),
    OrderID varchar(20),
    TrainID varchar(5),
    StartDate date,
    TrainStartDate date,
    StartStationID integer,
    EndStationID integer,
    StartStationNum integer,
    EndStationNum integer,
    SeatType integer,
    Price decimal(5,1),
    Status integer,
    primary key (OrderID, TrainID)
    );
/* SeatType  0 硬座  1 软座  2 硬卧上  3 硬卧中  4 硬卧下  5 软卧上  6 软卧下 */
/* Status  0 = 取消    1 = 付款*/
```

存储订单信息，包括订单号和火车车次，其中座位类型为整型变量，取值为 0—6，代表不同座位；设置订单状态变量，标志取消与成功，在需求 7 中显示订单状态有用。

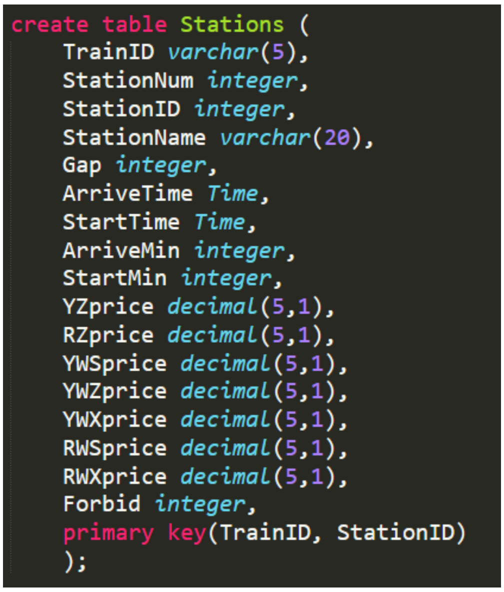
4. 火车



```sql
create table Trains (
    TrainID varchar(5) primary key,
    YZflag integer,
    RZflag integer,
    YWSflag integer,
    YWZflag integer,
    YWXflag integer,
    RWSflag integer,
    RWXflag integer
    );
```

存储车次信息，其他的座位类型 flag 标识此火车是否开设此座位，加这个变量是为了查询处理的方便。

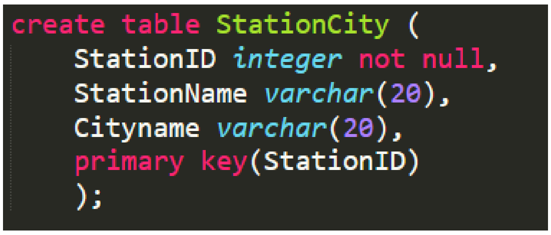
5. 与某车次相关的车站



```sql
create table Stations (
    TrainID varchar(5),
    StationNum integer,
    StationID integer,
    StationName varchar(20),
    Gap integer,
    ArriveTime Time,
    StartTime Time,
    ArriveMin integer,
    StartMin integer,
    YZprice decimal(5,1),
    RZprice decimal(5,1),
    YWSprice decimal(5,1),
    YWZprice decimal(5,1),
    YWXprice decimal(5,1),
    RWSprice decimal(5,1),
    RWXprice decimal(5,1),
    Forbid integer,
    primary key(TrainID, StationID)
    );
```

这张表存储了固定车次下的各站信息。由于某个线路上站是存在先后顺序的，而给的数据似乎往返程也并不对称，因此设置 StationNum 来标识此站在某个线路中的位置；StationID 则是唯一的车站号，通过 all-stations.csv 拿到。
由于存在跨天的情况，设置了 Gap 变量来标识，若从首发站到该站时已跨天则为 1，否则为 0，这样方便之后的日期判断；出发与到达时间也需要记录，为了计算方便，同时记录了换算为分钟的数值。
记录了不同座位的票价信息。
由于有些站不售票，设置了 Forbid 变量来标识，售票为 0，不售票为 1。需要注意的是首发站，票是按首发站到该站算的，因此首发站无票，但首发站可以上车，处理时 Forbid 设为 0。

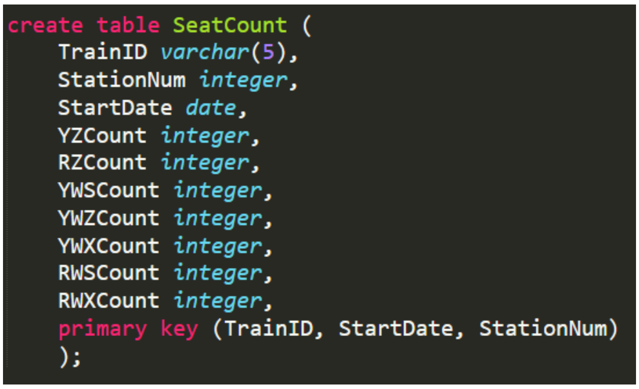
6. 车站所在城市



```
create table StationCity (
    StationID integer not null,
    StationName varchar(20),
    Cityname varchar(20),
    primary key(StationID)
    );
```

由于查询两地车次是按城市查，因此需要记录每个车站所在城市。

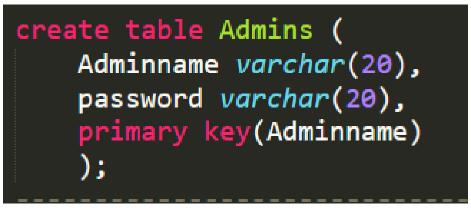
6.余座记录



```
create table SeatCount (
    TrainID varchar(5),
    StationNum integer,
    StartDate date,
    YZCount integer,
    RZCount integer,
    YWSCount integer,
    YWZCount integer,
    YWXCount integer,
    RWSCount integer,
    RWXCount integer,
    primary key (TrainID, StartDate, StationNum)
    );
```

由于如果有票则初始化为 5 张，因此需要记录日期和各座位余票，开设则初始化为 5，不开设初始化为 0。

7.管理员



```
create table Admins (
    Adminname varchar(20),
    password varchar(20),
    primary key(Adminname)
    );
```

管理员可能有多位，且拥有权限与用户不一样，因此单独设表。

三、范式分析与细化

1.Users

候选键为（UserID）,(Phone),均超键到所有属性，正常。

2.Orders

候选键为(OrderID, TrainID)

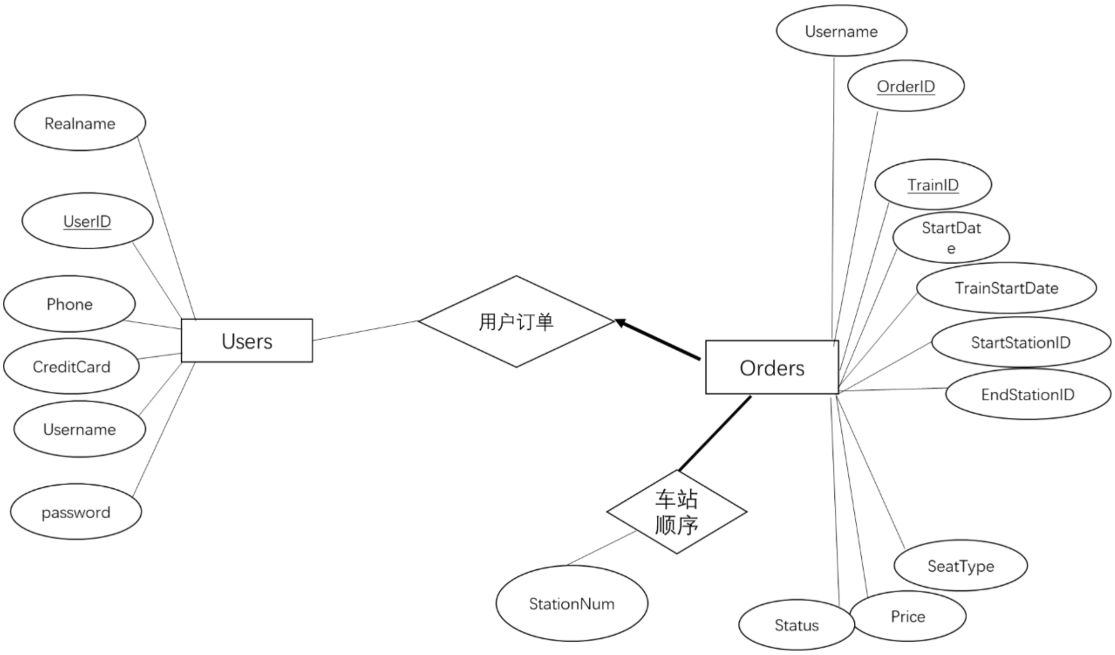
函数依赖有：

a. (OrderID, TrainID->所有属性 正常

b.（TrainID,StartStaionID）->StartStaionNum 非键传递依赖

c. (TrainID,EndStationID)->EndStationNum 非键传递依赖

处理冗余得到新的 ER 图：

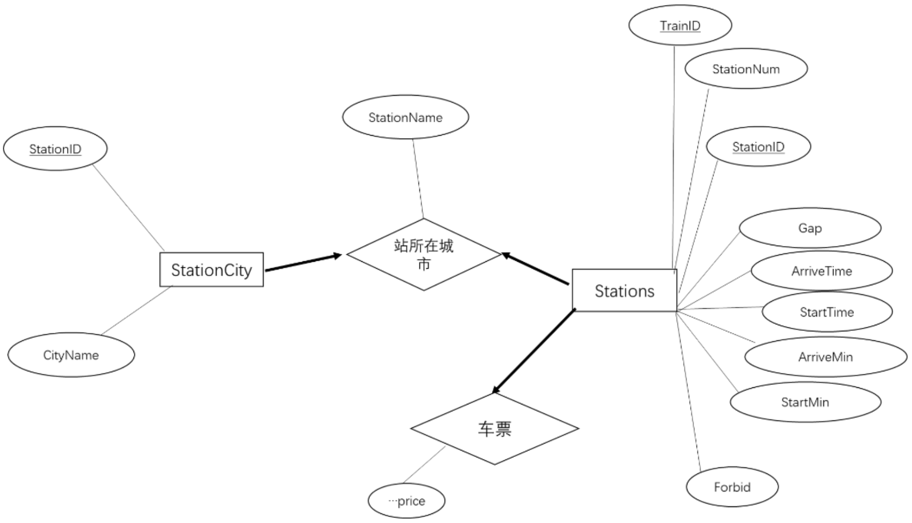


3.Trains
候选键为（TrainID）,超键到所有属性，正常。

4.Stations && StationCity
Stations 候选键为(TrainID, StationID)，(TrainID, StationNum)，(TrainID, StationName)
产生冗余的函数依赖有：
a.StationID->StationName 对于键属性的函数依赖
b.(…price)->Forbid 非键传递依赖

StationCity 候选键为（StationID）,(StationName)
产生冗余的函数依赖有：
a.StationID->StationName 对于键属性的函数依赖

处理冗余得到新的ER图：



5.SeatCount

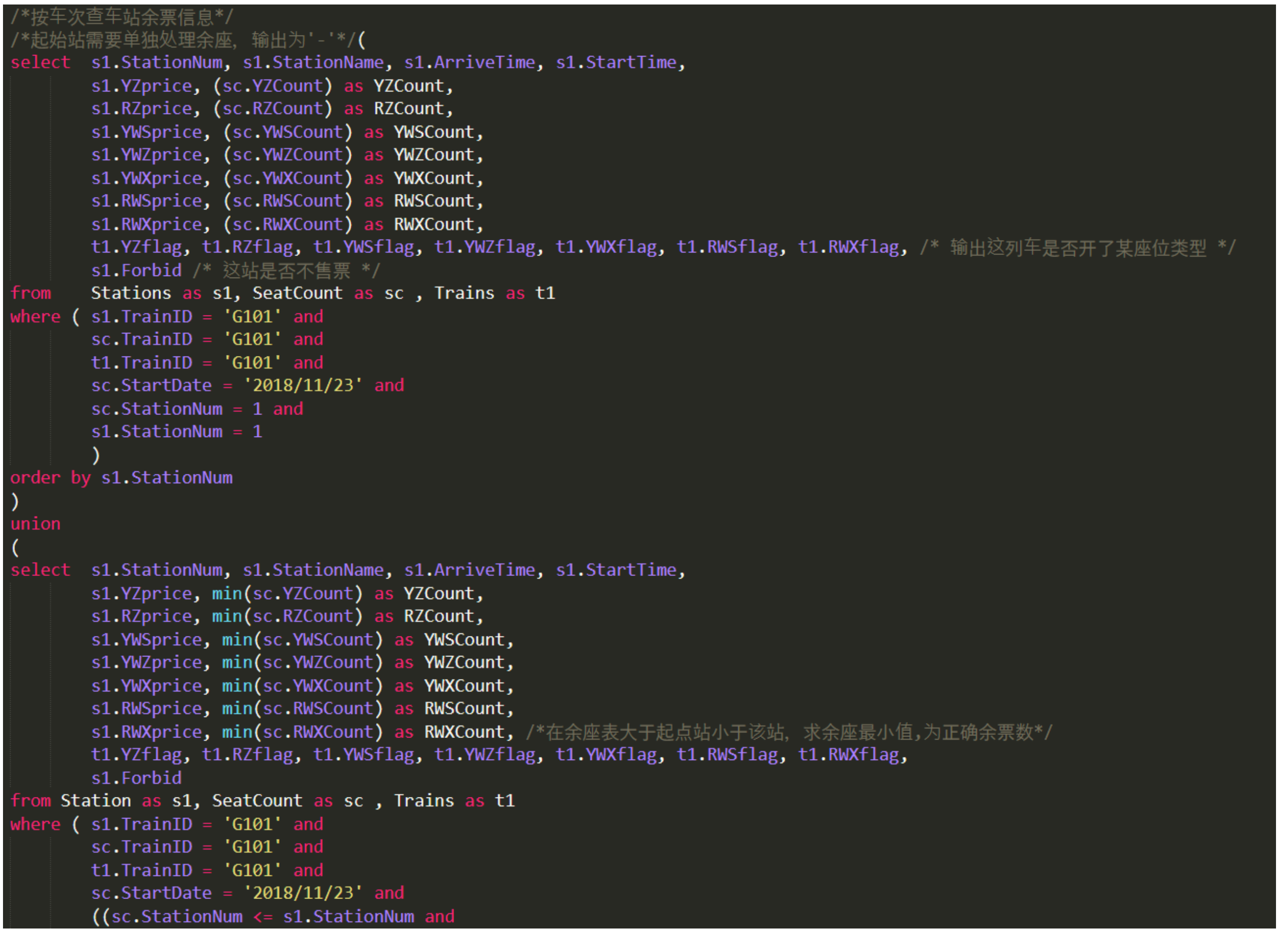
候选键为(TrainID, StartDate, StationNum)，属性正常。

6.Admins

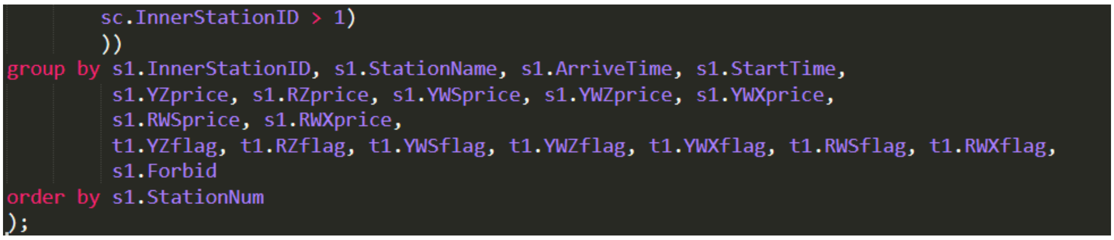
候选键为(Adminname)，属性正常。

四、SQL 查询模板

1.需求 4 查询具体车次



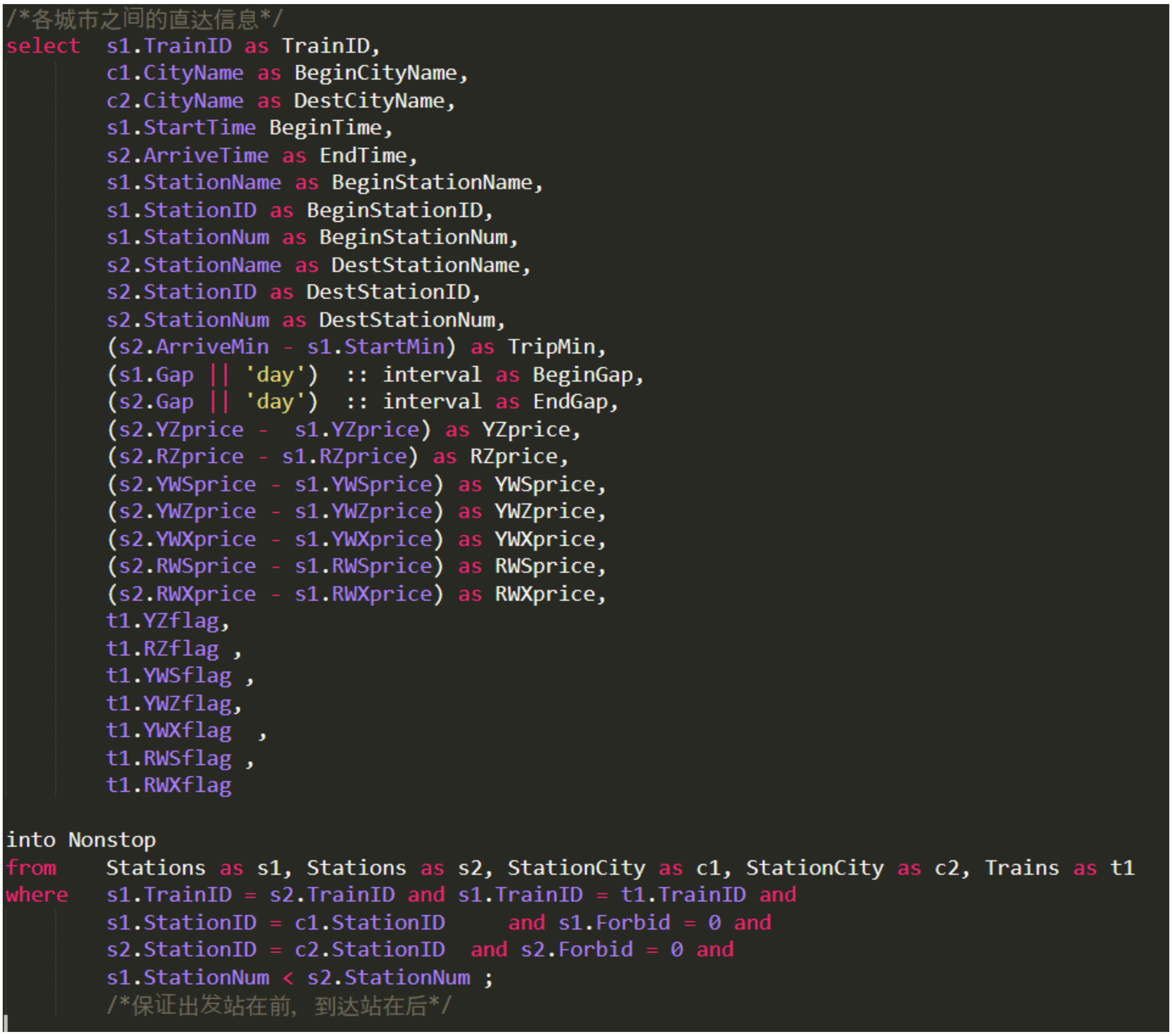
```
/*按车次查车站余票信息*/
/*起始站需要单独处理余座, 输出为'-'*/(
select  s1.StationNum, s1.StationName, s1.ArriveTime, s1.StartTime,
        s1.YZprice, (sc.YZCount) as YZCount,
        s1.RZprice, (sc.RZCount) as RZCount,
        s1.YWSprice, (sc.YWSCount) as YWSCount,
        s1.YWZprice, (sc.YWZCount) as YWZCount,
        s1.YWXprice, (sc.YWXCount) as YWXCount,
        s1.RWSprice, (sc.RWSCount) as RWSCount,
        s1.RWXprice, (sc.RWXCount) as RWXCount,
        t1.YZflag, t1.RZflag, t1.YWSflag, t1.YWZflag, t1.YWXflag, t1.RWSflag, t1.RWXflag, /* 输出这列车是否开了某座位类型 */
        s1.Forbid /* 这站是否不售票 */
from    Stations as s1, SeatCount as sc , Trains as t1
where ( s1.TrainID = 'G101' and
        sc.TrainID = 'G101' and
        t1.TrainID = 'G101' and
        sc.StartDate = '2018/11/23' and
        sc.StationNum = 1 and
        s1.StationNum = 1
        )
order by s1.StationNum
)
union
(
select  s1.StationNum, s1.StationName, s1.ArriveTime, s1.StartTime,
        s1.YZprice, min(sc.YZCount) as YZCount,
        s1.RZprice, min(sc.RZCount) as RZCount,
        s1.YWSprice, min(sc.YWSCount) as YWSCount,
        s1.YWZprice, min(sc.YWZCount) as YWZCount,
        s1.YWXprice, min(sc.YWXCount) as YWXCount,
        s1.RWSprice, min(sc.RWSCount) as RWSCount,
        s1.RWXprice, min(sc.RWXCount) as RWXCount, /*在余座表大于起点站小于该站, 求余座最小值,为正确余票数*/
        t1.YZflag, t1.RZflag, t1.YWSflag, t1.YWZflag, t1.YWXflag, t1.RWSflag, t1.RWXflag,
        s1.Forbid
from Station as s1, SeatCount as sc , Trains as t1
where ( s1.TrainID = 'G101' and
        sc.TrainID = 'G101' and
        t1.TrainID = 'G101' and
        sc.StartDate = '2018/11/23' and
        ((sc.StationNum <= s1.StationNum and
```



```
         sc.InnerStationID > 1)
         ))
group by s1.InnerStationID, s1.StationName, s1.ArriveTime, s1.StartTime,
         s1.YZprice, s1.RZprice, s1.YWSprice, s1.YWZprice, s1.YWXprice,
         s1.RWSprice, s1.RWXprice,
         t1.YZflag, t1.RZflag, t1.YWSflag, t1.YWZflag, t1.YWXflag, t1.RWSflag, t1.RWXflag,
         s1.Forbid
order by s1.StationNum
);
```

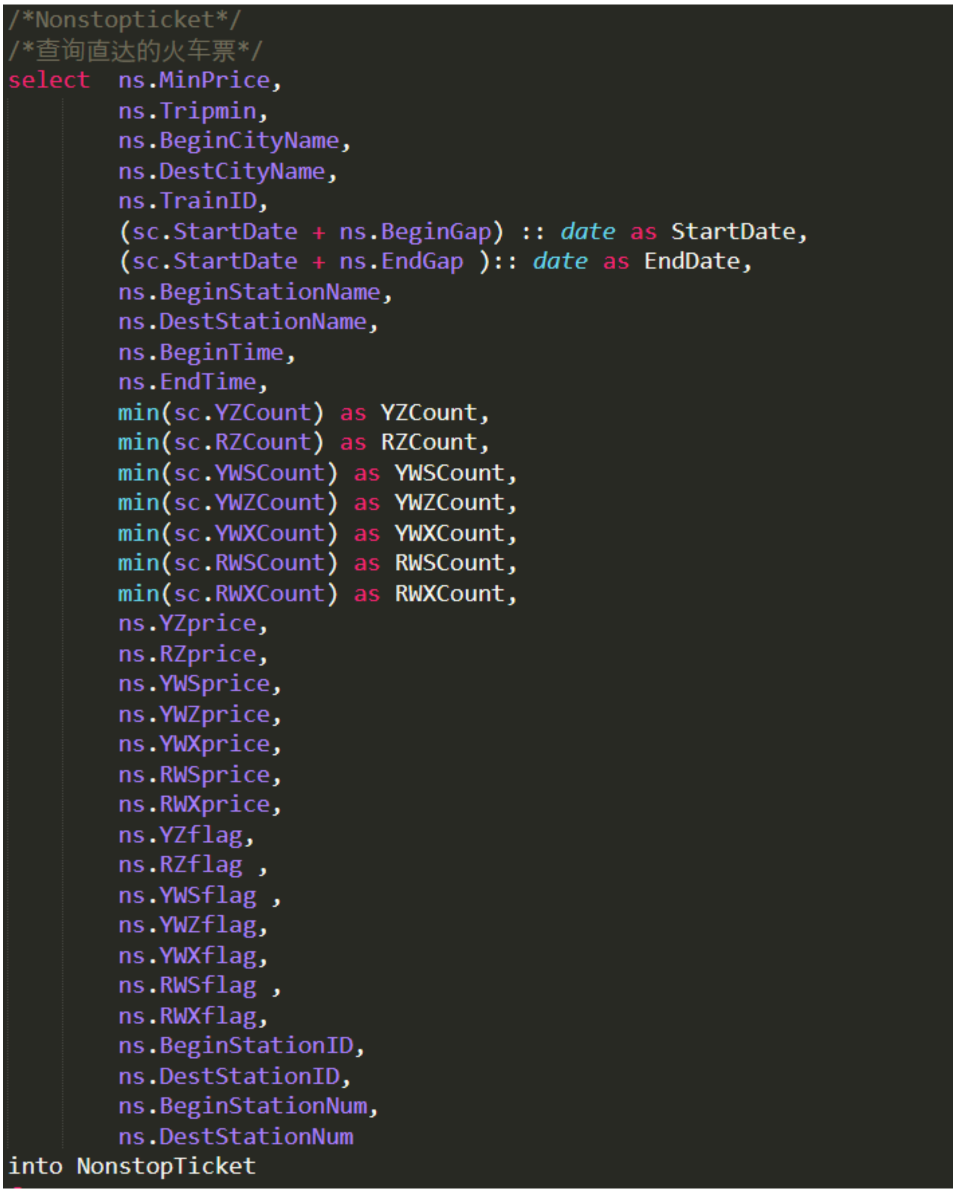
2.需求 5 查询两地之间车次

新建 NonStop 表存储各城市直达车辆信息,这样直达可以直接查询,换乘一次则是从 Nonstop 选择两班来组成，满足换乘城市既是第一班的目的地，又是第二班的起始地。

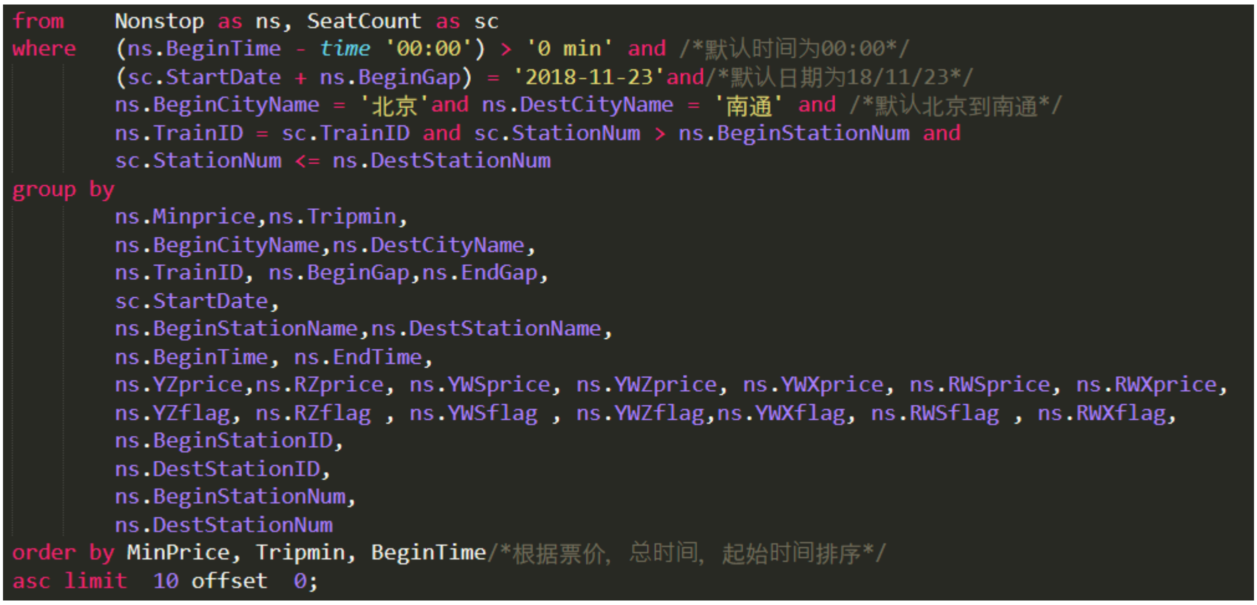


```
/*各城市之间的直达信息*/
select  s1.TrainID as TrainID,
        c1.CityName as BeginCityName,
        c2.CityName as DestCityName,
        s1.StartTime BeginTime,
        s2.ArriveTime as EndTime,
        s1.StationName as BeginStationName,
        s1.StationID as BeginStationID,
        s1.StationNum as BeginStationNum,
        s2.StationName as DestStationName,
        s2.StationID as DestStationID,
        s2.StationNum as DestStationNum,
        (s2.ArriveMin - s1.StartMin) as TripMin,
        (s1.Gap || 'day')  :: interval as BeginGap,
        (s2.Gap || 'day')  :: interval as EndGap,
        (s2.YZprice -  s1.YZprice) as YZprice,
        (s2.RZprice - s1.RZprice) as RZprice,
        (s2.YWSprice - s1.YWSprice) as YWSprice,
        (s2.YWZprice - s1.YWZprice) as YWZprice,
        (s2.YWXprice - s1.YWXprice) as YWXprice,
        (s2.RWSprice - s1.RWSprice) as RWSprice,
        (s2.RWXprice - s1.RWXprice) as RWXprice,
        t1.YZflag,
        t1.RZflag ,
        t1.YWSflag ,
        t1.YWZflag,
        t1.YWXflag  ,
        t1.RWSflag ,
        t1.RWXflag

into Nonstop
from    Stations as s1, Stations as s2, StationCity as c1, StationCity as c2, Trains as t1
where   s1.TrainID = s2.TrainID and s1.TrainID = t1.TrainID and
        s1.StationID = c1.StationID     and s1.Forbid = 0 and
        s2.StationID = c2.StationID  and s2.Forbid = 0 and
        s1.StationNum < s2.StationNum ;
        /*保证出发站在前，到达站在后*/
```

直达：

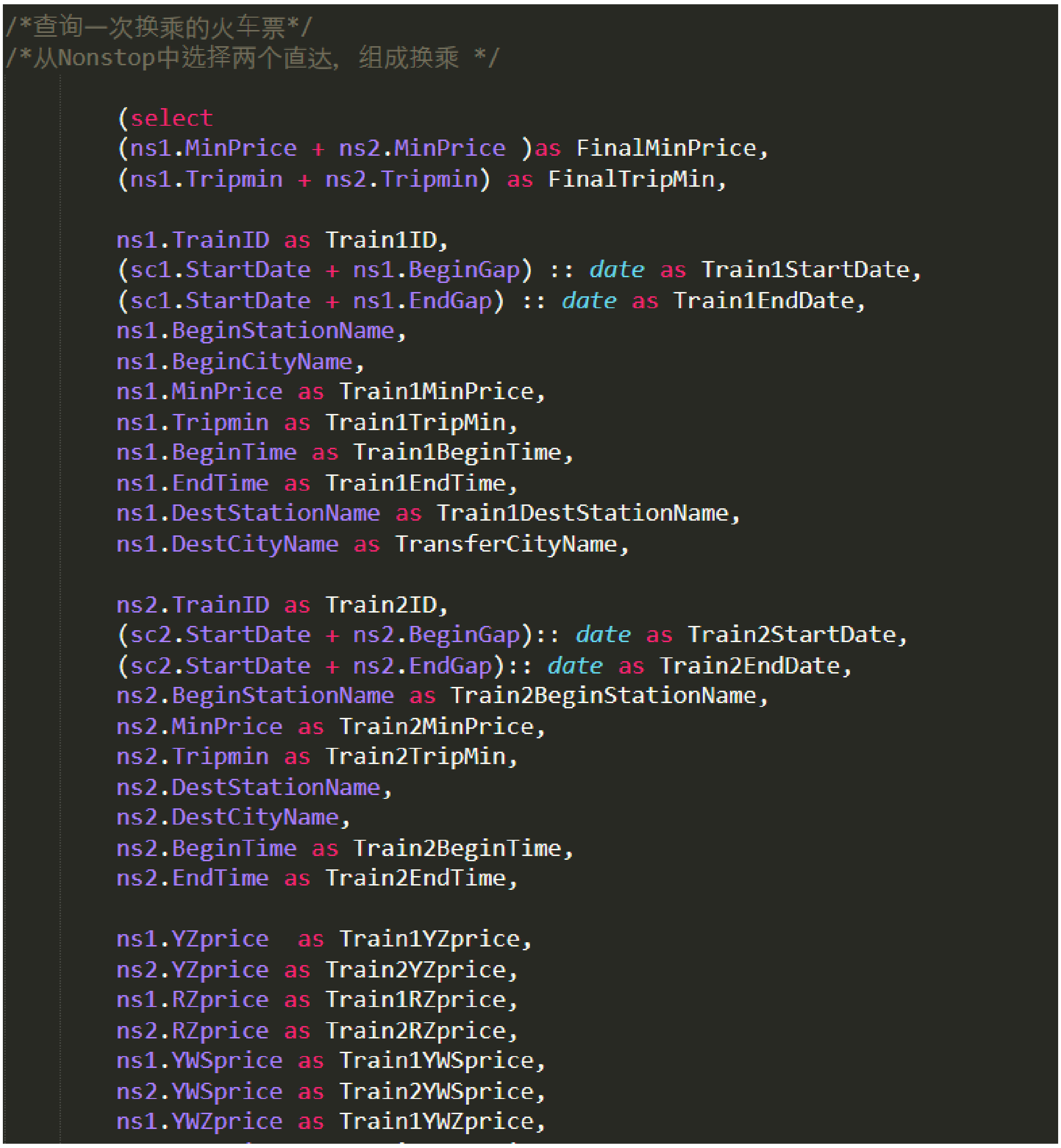


```
/*Nonstopticket*/
/*查询直达的火车票*/
select  ns.MinPrice,
        ns.Tripmin,
        ns.BeginCityName,
        ns.DestCityName,
        ns.TrainID,
        (sc.StartDate + ns.BeginGap) :: date as StartDate,
        (sc.StartDate + ns.EndGap ):: date as EndDate,
        ns.BeginStationName,
        ns.DestStationName,
        ns.BeginTime,
        ns.EndTime,
        min(sc.YZCount) as YZCount,
        min(sc.RZCount) as RZCount,
        min(sc.YWSCount) as YWSCount,
        min(sc.YWZCount) as YWZCount,
        min(sc.YWXCount) as YWXCount,
        min(sc.RWSCount) as RWSCount,
        min(sc.RWXCount) as RWXCount,
        ns.YZprice,
        ns.RZprice,
        ns.YWSprice,
        ns.YWZprice,
        ns.YWXprice,
        ns.RWSprice,
        ns.RWXprice,
        ns.YZflag,
        ns.RZflag ,
        ns.YWSflag ,
        ns.YWZflag,
        ns.YWXflag,
        ns.RWSflag ,
        ns.RWXflag,
        ns.BeginStationID,
        ns.DestStationID,
        ns.BeginStationNum,
        ns.DestStationNum
into NonstopTicket
```

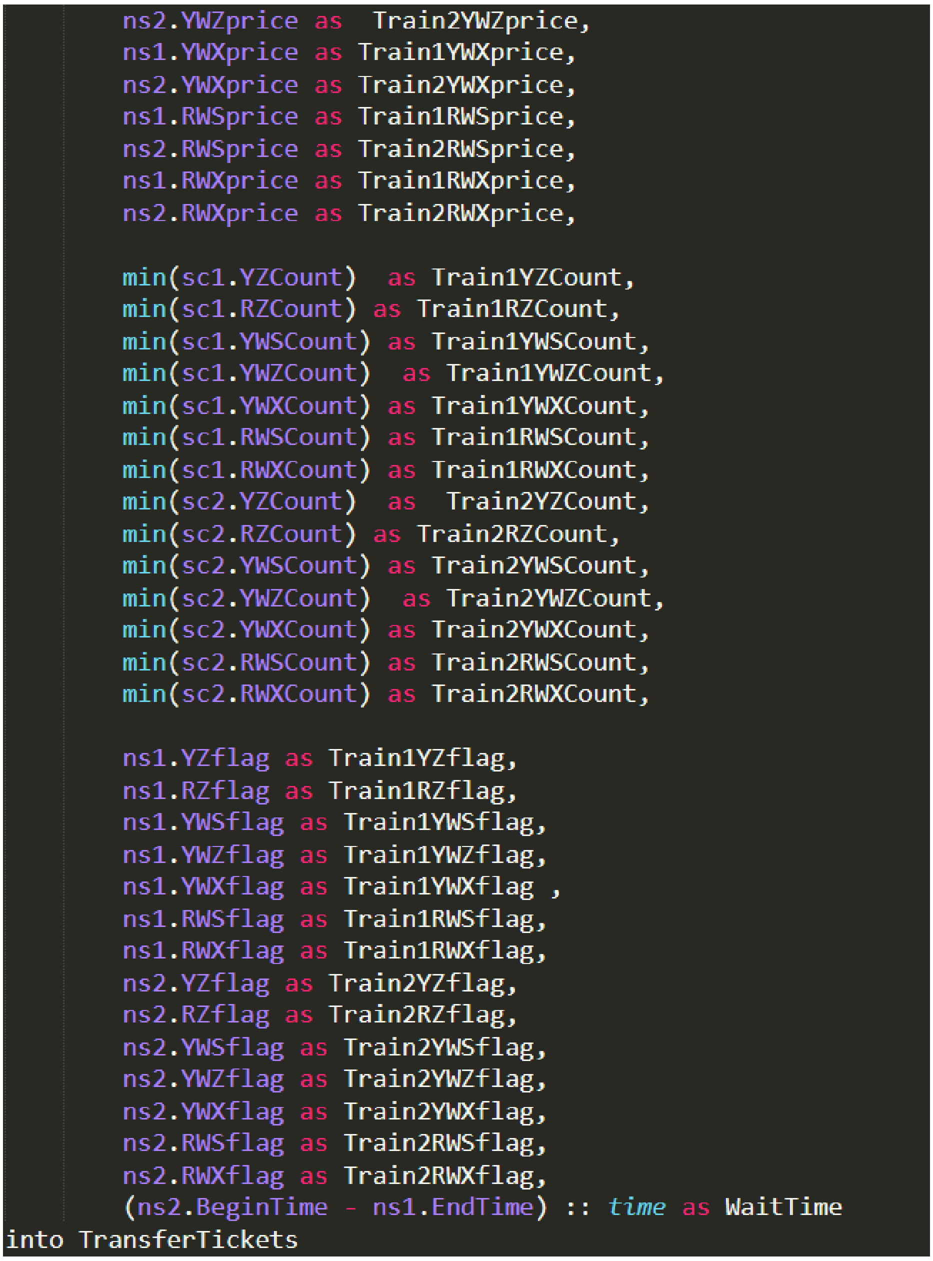


```
from    Nonstop as ns, SeatCount as sc
where   (ns.BeginTime - time '00:00') > '0 min' and /*默认时间为00:00*/
        (sc.StartDate + ns.BeginGap) = '2018-11-23'and/*默认日期为18/11/23*/
        ns.BeginCityName = '北京'and ns.DestCityName = '南通' and /*默认北京到南通*/
        ns.TrainID = sc.TrainID and sc.StationNum > ns.BeginStationNum and
        sc.StationNum <= ns.DestStationNum
group by
        ns.Minprice,ns.Tripmin,
        ns.BeginCityName,ns.DestCityName,
        ns.TrainID, ns.BeginGap,ns.EndGap,
        sc.StartDate,
        ns.BeginStationName,ns.DestStationName,
        ns.BeginTime, ns.EndTime,
        ns.YZprice,ns.RZprice, ns.YWSprice, ns.YWZprice, ns.YWXprice, ns.RWSprice, ns.RWXprice,
        ns.YZflag, ns.RZflag , ns.YWSflag , ns.YWZflag,ns.YWXflag, ns.RWSflag , ns.RWXflag,
        ns.BeginStationID,
        ns.DestStationID,
        ns.BeginStationNum,
        ns.DestStationNum
order by MinPrice, Tripmin, BeginTime/*根据票价，总时间，起始时间排序*/
asc limit  10 offset  0;
```

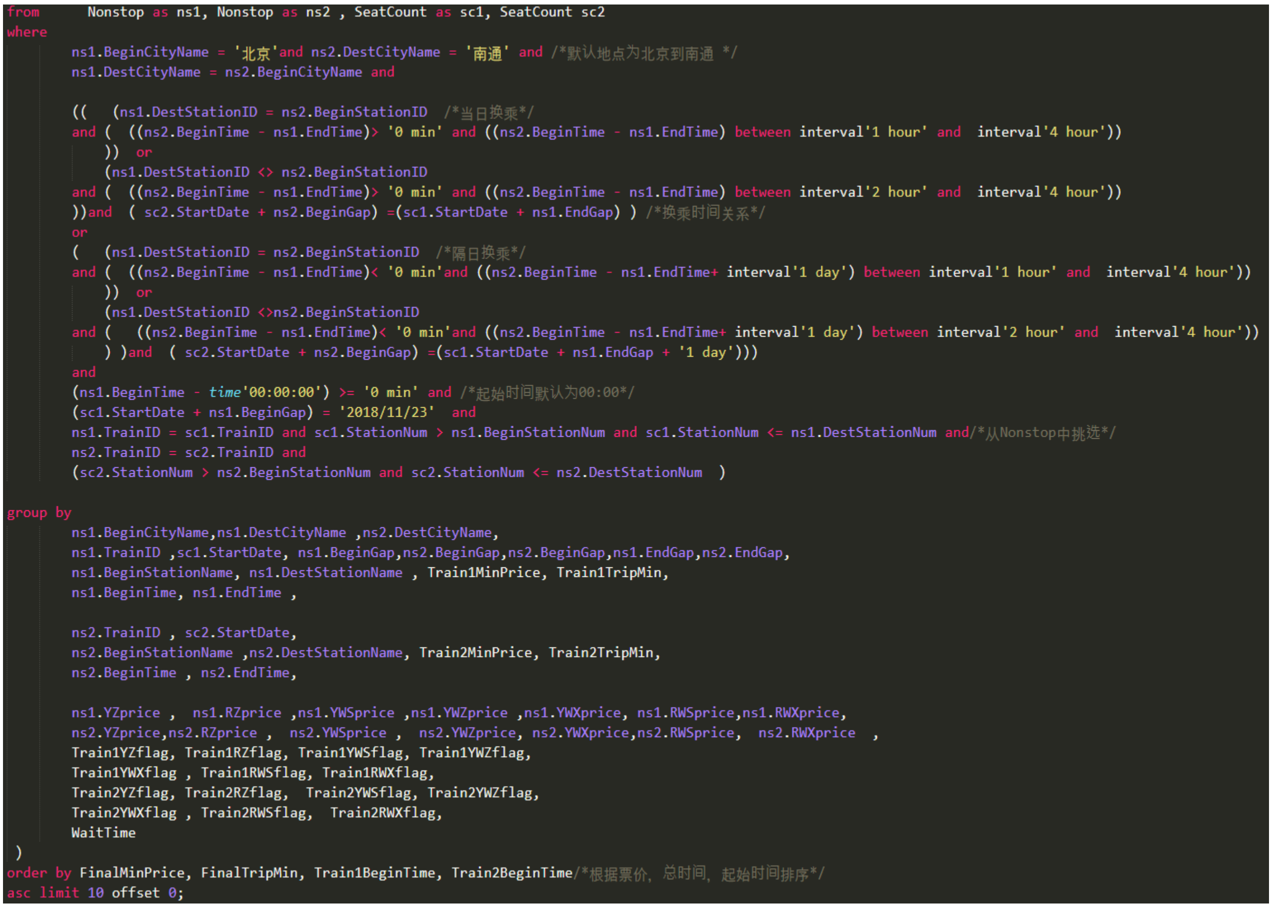
换乘：



```
/*查询一次换乘的火车票*/
/*从Nonstop中选择两个直达，组成换乘 */

        (select
        (ns1.MinPrice + ns2.MinPrice )as FinalMinPrice,
        (ns1.Tripmin + ns2.Tripmin) as FinalTripMin,

        ns1.TrainID as Train1ID,
        (sc1.StartDate + ns1.BeginGap) :: date as Train1StartDate,
        (sc1.StartDate + ns1.EndGap) :: date as Train1EndDate,
        ns1.BeginStationName,
        ns1.BeginCityName,
        ns1.MinPrice as Train1MinPrice,
        ns1.Tripmin as Train1TripMin,
        ns1.BeginTime as Train1BeginTime,
        ns1.EndTime as Train1EndTime,
        ns1.DestStationName as Train1DestStationName,
        ns1.DestCityName as TransferCityName,

        ns2.TrainID as Train2ID,
        (sc2.StartDate + ns2.BeginGap):: date as Train2StartDate,
        (sc2.StartDate + ns2.EndGap):: date as Train2EndDate,
        ns2.BeginStationName as Train2BeginStationName,
        ns2.MinPrice as Train2MinPrice,
        ns2.Tripmin as Train2TripMin,
        ns2.DestStationName,
        ns2.DestCityName,
        ns2.BeginTime as Train2BeginTime,
        ns2.EndTime as Train2EndTime,

        ns1.YZprice  as Train1YZprice,
        ns2.YZprice as Train2YZprice,
        ns1.RZprice as Train1RZprice,
        ns2.RZprice as Train2RZprice,
        ns1.YWSprice as Train1YWSprice,
        ns2.YWSprice as Train2YWSprice,
        ns1.YWZprice as Train1YWZprice,
```

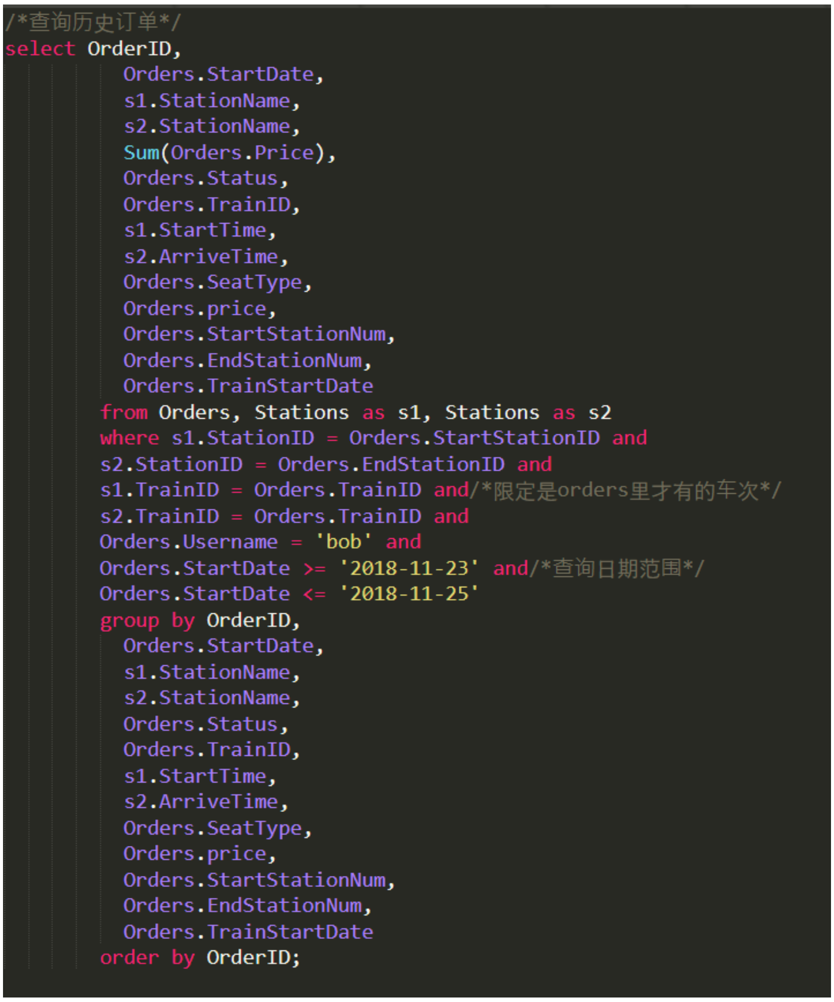


```
        ns2.YWZprice as  Train2YWZprice,
        ns1.YWXprice as Train1YWXprice,
        ns2.YWXprice as Train2YWXprice,
        ns1.RWSprice as Train1RWSprice,
        ns2.RWSprice as Train2RWSprice,
        ns1.RWXprice as Train1RWXprice,
        ns2.RWXprice as Train2RWXprice,

        min(sc1.YZCount)  as Train1YZCount,
        min(sc1.RZCount) as Train1RZCount,
        min(sc1.YWSCount) as Train1YWSCount,
        min(sc1.YWZCount)  as Train1YWZCount,
        min(sc1.YWXCount) as Train1YWXCount,
        min(sc1.RWSCount) as Train1RWSCount,
        min(sc1.RWXCount) as Train1RWXCount,
        min(sc2.YZCount)  as  Train2YZCount,
        min(sc2.RZCount) as Train2RZCount,
        min(sc2.YWSCount) as Train2YWSCount,
        min(sc2.YWZCount)  as Train2YWZCount,
        min(sc2.YWXCount) as Train2YWXCount,
        min(sc2.RWSCount) as Train2RWSCount,
        min(sc2.RWXCount) as Train2RWXCount,

        ns1.YZflag as Train1YZflag,
        ns1.RZflag as Train1RZflag,
        ns1.YWSflag as Train1YWSflag,
        ns1.YWZflag as Train1YWZflag,
        ns1.YWXflag as Train1YWXflag ,
        ns1.RWSflag as Train1RWSflag,
        ns1.RWXflag as Train1RWXflag,
        ns2.YZflag as Train2YZflag,
        ns2.RZflag as Train2RZflag,
        ns2.YWSflag as Train2YWSflag,
        ns2.YWZflag as Train2YWZflag,
        ns2.YWXflag as Train2YWXflag,
        ns2.RWSflag as Train2RWSflag,
        ns2.RWXflag as Train2RWXflag,
        (ns2.BeginTime - ns1.EndTime) :: time as WaitTime
into TransferTickets
```



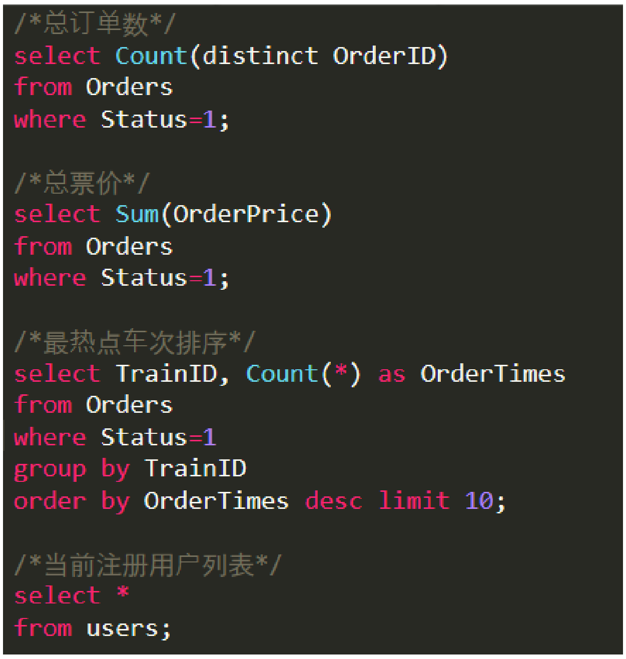
```
from      Nonstop as ns1, Nonstop as ns2 , SeatCount as sc1, SeatCount sc2
where
        ns1.BeginCityName = '北京'and ns2.DestCityName = '南通' and /*默认地点为北京到南通 */
        ns1.DestCityName = ns2.BeginCityName and

        ((   (ns1.DestStationID = ns2.BeginStationID  /*当日换乘*/
        and (  ((ns2.BeginTime - ns1.EndTime)> '0 min' and ((ns2.BeginTime - ns1.EndTime) between interval'1 hour' and  interval'4 hour'))
            ))  or
            (ns1.DestStationID <> ns2.BeginStationID
        and (  ((ns2.BeginTime - ns1.EndTime)> '0 min' and ((ns2.BeginTime - ns1.EndTime) between interval'2 hour' and  interval'4 hour'))
        ))and  ( sc2.StartDate + ns2.BeginGap) =(sc1.StartDate + ns1.EndGap) ) /*换乘时间关系*/
        or
        (   (ns1.DestStationID = ns2.BeginStationID  /*隔日换乘*/
        and (  ((ns2.BeginTime - ns1.EndTime)< '0 min'and ((ns2.BeginTime - ns1.EndTime+ interval'1 day') between interval'1 hour' and  interval'4 hour'))
            ))  or
            (ns1.DestStationID <>ns2.BeginStationID
        and (   ((ns2.BeginTime - ns1.EndTime)< '0 min'and ((ns2.BeginTime - ns1.EndTime+ interval'1 day') between interval'2 hour' and  interval'4 hour'))
            ) )and  ( sc2.StartDate + ns2.BeginGap) =(sc1.StartDate + ns1.EndGap + '1 day')))
        and
        (ns1.BeginTime - time'00:00:00') >= '0 min' and /*起始时间默认为00:00*/
        (sc1.StartDate + ns1.BeginGap) = '2018/11/23'  and
        ns1.TrainID = sc1.TrainID and sc1.StationNum > ns1.BeginStationNum and sc1.StationNum <= ns1.DestStationNum and/*从Nonstop中挑选*/
        ns2.TrainID = sc2.TrainID and
        (sc2.StationNum > ns2.BeginStationNum and sc2.StationNum <= ns2.DestStationNum  )

group by
        ns1.BeginCityName,ns1.DestCityName ,ns2.DestCityName,
        ns1.TrainID ,sc1.StartDate, ns1.BeginGap,ns2.BeginGap,ns2.BeginGap,ns1.EndGap,ns2.EndGap,
        ns1.BeginStationName, ns1.DestStationName , Train1MinPrice, Train1TripMin,
        ns1.BeginTime, ns1.EndTime ,

        ns2.TrainID , sc2.StartDate,
        ns2.BeginStationName ,ns2.DestStationName, Train2MinPrice, Train2TripMin,
        ns2.BeginTime , ns2.EndTime,

        ns1.YZprice ,  ns1.RZprice ,ns1.YWSprice ,ns1.YWZprice ,ns1.YWXprice, ns1.RWSprice,ns1.RWXprice,
        ns2.YZprice,ns2.RZprice ,  ns2.YWSprice ,  ns2.YWZprice, ns2.YWXprice,ns2.RWSprice,  ns2.RWXprice  ,
        Train1YZflag, Train1RZflag, Train1YWSflag, Train1YWZflag,
        Train1YWXflag , Train1RWSflag, Train1RWXflag,
        Train2YZflag, Train2RZflag,  Train2YWSflag, Train2YWZflag,
        Train2YWXflag , Train2RWSflag,  Train2RWXflag,
        WaitTime
 )
order by FinalMinPrice, FinalTripMin, Train1BeginTime, Train2BeginTime/*根据票价, 总时间, 起始时间排序*/
asc limit 10 offset 0;
```

3. 需求 7：查询历史订单



```
/*查询历史订单*/
select OrderID,
          Orders.StartDate,
          s1.StationName,
          s2.StationName,
          Sum(Orders.Price),
          Orders.Status,
          Orders.TrainID,
          s1.StartTime,
          s2.ArriveTime,
          Orders.SeatType,
          Orders.price,
          Orders.StartStationNum,
          Orders.EndStationNum,
          Orders.TrainStartDate
        from Orders, Stations as s1, Stations as s2
        where s1.StationID = Orders.StartStationID and
        s2.StationID = Orders.EndStationID and
        s1.TrainID = Orders.TrainID and/*限定是orders里才有的车次*/
        s2.TrainID = Orders.TrainID and
        Orders.Username = 'bob' and
        Orders.StartDate >= '2018-11-23' and/*查询日期范围*/
        Orders.StartDate <= '2018-11-25'
        group by OrderID,
          Orders.StartDate,
          s1.StationName,
          s2.StationName,
          Orders.Status,
          Orders.TrainID,
          s1.StartTime,
          s2.ArriveTime,
          Orders.SeatType,
          Orders.price,
          Orders.StartStationNum,
          Orders.EndStationNum,
          Orders.TrainStartDate
        order by OrderID;
```

4. 需求 8：管理员



```
/*总订单数*/
select Count(distinct OrderID)
from Orders
where Status=1;

/*总票价*/
select Sum(OrderPrice)
from Orders
where Status=1;

/*最热点车次排序*/
select TrainID, Count(*) as OrderTimes
from Orders
where Status=1
group by TrainID
order by OrderTimes desc limit 10;

/*当前注册用户列表*/
select *
from users;
```