数据库系统应用设计

组员：刘国栋，李一苇，李鹏辉

一． 设计思路

1. 为了方便对车次和余票的统计，将车次和余座信息采用按站存储，每次的数据都是相对于前一站到该站，方便最后数据的处理，和购票后信息的筛选。

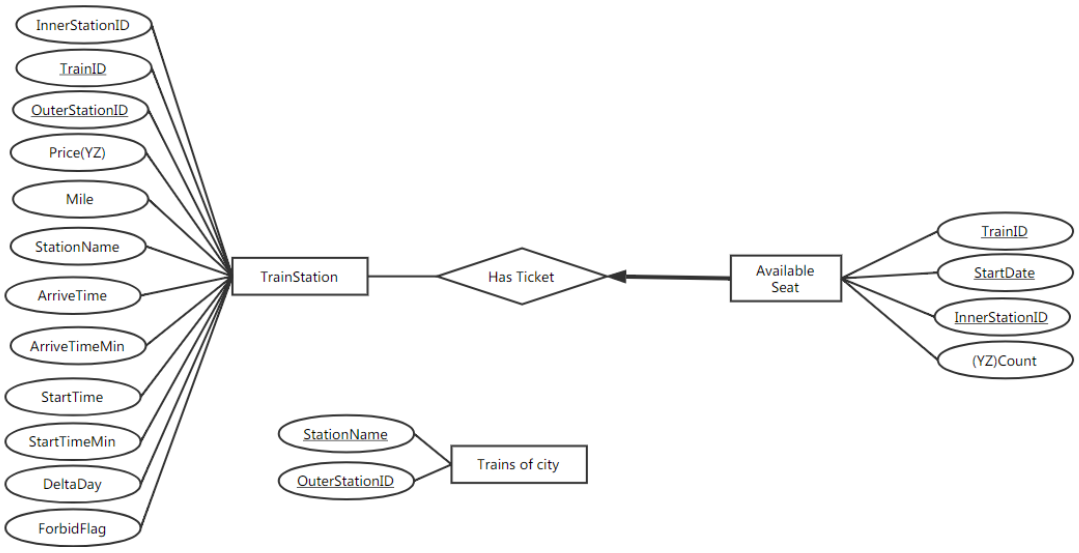
2. 因为车次每天都是重复的，所以在车次信息不需要存储时间，而余座信息是每天都是变化的，所以建立的余座信息表是根据时间存储的，求余票是根据满足要求的车次来判断是否有余票。

3. 因为用户每次数据起始点和到底地都输入的是城市名称而不是火车站名，所以建立一个城市和城市中所有火车站的表格，每次输入城市名能够关联到对应的同城火车站。

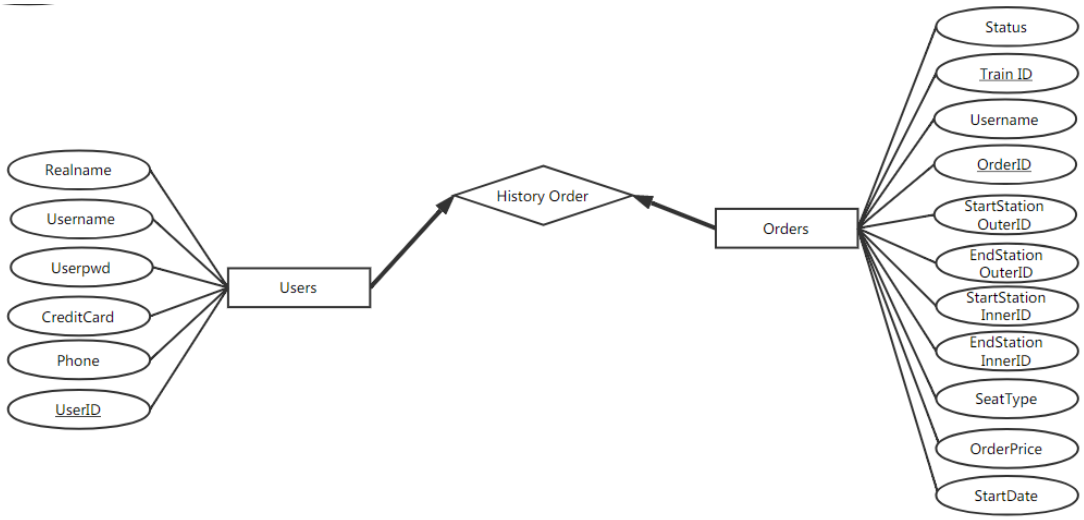
4. 用 SQL 语句生成一个全国内所有车站之间的直达火车的表单 OneTrip，每次查询余票时只需要从此表单中查找满足条件的车次。若为直达，则只需在 OneTrip 中查找满足地点（出发城市和到达城市的列车），并且联系当天的火车票信息表 AvailableSeat 判断是否有余票，将有余票的前十趟最优的方案打印。如需要换乘，则只需要添加限制条件（中间城市相同并且换乘时间满足条件），然后从 OneTrip 表中产生组合满足条件的换乘列车，并将最优的十组方案打印。

二． ER 图设计

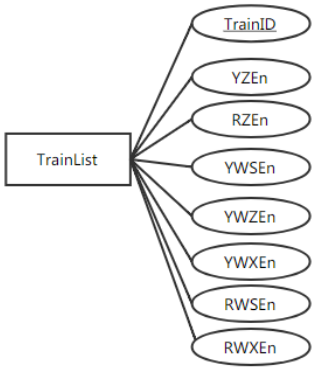
1. 火车及余票信息



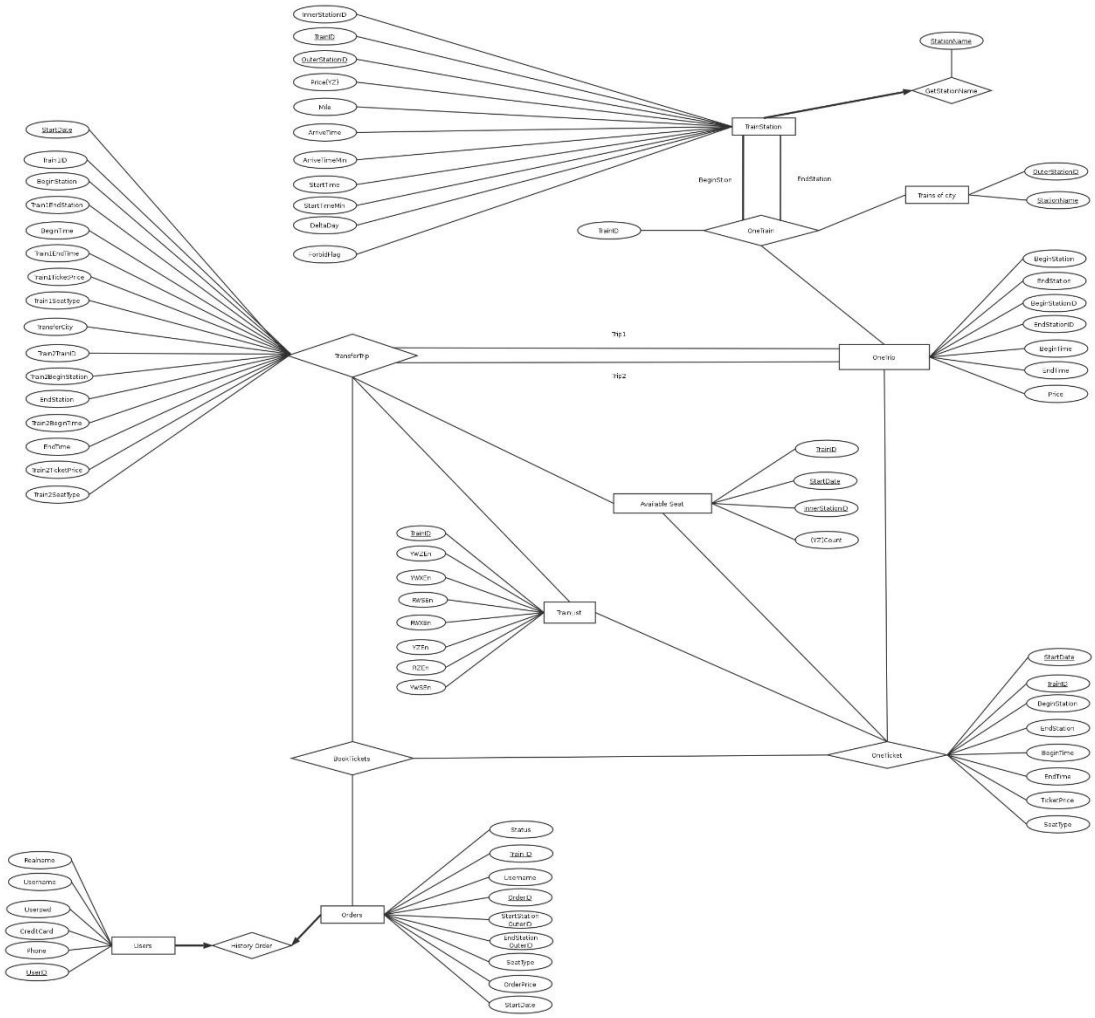
2. 用户及订单信息



3. 开设座位类型火车信息



4. 总 ER 图



三． 关系模型

1. 列车信息

1）根据设计思路，将每一趟列车按照站的排列存储，此表中包含了列车的 TrainID，此站在该趟列车中的顺序，站名等基本信息，以及从起点站到该站的各种座次的价格。

2）为了方便最后时间的排序，分别存储了 time 格式的时间和相对于列车出发日期的 00:00 的分钟数。

3）因为部分车站不能进行上下车 ,所以设置了一个 ForbidFlag 来标识此站是否可以上

下车。

```
create table TrainStation (
    InnerStationID integer,
    TrainID varchar(5),
    OuterStationID integer,
    StationName varchar(20),
    Mile integer,
    PriceYZ decimal(5,1),
    PriceRZ decimal(5,1),
    PriceYWS decimal(5,1),
    PriceYWZ decimal(5,1),
    PriceYWX decimal(5,1),
    PriceRWS decimal(5,1),
    PriceRWX decimal(5,1),

    DeltaDay integer,
    ArriveTime Time,
    StartTime Time,
    ArriveTimeMin integer,
    StartTimeMin integer,
    ForbidFlag integer,
    primary key(TrainID, OuterStationID)
```

```
);
```

2. 城市和城市的各站名

```
create table TrainsOfCity (
    StationName varchar(20),
    OuterStationID integer not null,
    primary key(CityName, OuterStationID));
```

3. 订单

包含订单状态和列车车次，以及起始点站名。

```
create table Orders (
    Status integer, /* 0 =  未付款      1 =  付款      -1 =  取消  */
    OrderID varchar(20),
    TrainID varchar(5),
    StartDate date,
    StartStationOuterID integer,
    EndStationOuterID integer,
    StartStationInnerID integer,
    EndStationInnerID integer,
    SeatType integer,/* 0 YZ   1 RZ   2 YWS   3 YWZ   4 YWX   5 RWS   6 RWX */
    OrderPrice decimal(5,1),
    Username varchar(20),

    TrainStartDate date,
```

```
primary key (OrderID, TrainID)
);
```

4. 用户

包含用户的基本信息，如姓名，电话，银行卡等。

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

5. 余座表

包含列车发车当天的日期，列车的 TrainID 和前一站到此站的余座信息

```
create table AvailableSeat (
    TrainID varchar(5),
    StartDate date,
    InnerStationID integer,
    YZCount integer,
    RZCount integer,
    YWSCount integer,
    YWZCount integer,
```

YWXCount integer,

RWSCount integer,

RWXCount integer,

primary key (TrainID, StartDate, InnerStationID)

);

四． 范式分析

1. TrainList

候选键是 TrainID 只存在的函数依赖是超键到所有属性，所以没有数据冗余。

2. TrainStation

候选键是{TrainID, InnerStationID}, {TrainID, StationName}, {TrainID, OuterStationID}

存在以下函数依赖：

1) {InnerStationID, TrainID} -> 所有属性 此依赖正常。

2) OuterStationID -> StationName 对于键属性的函数依赖 用 BCNF 消除

3) {PriceYZ, …, PriceRWX} -> ForbidFlag 非键的传递依赖 用 3NF 消除

4) ArriveTimeMin -> {ArriveTime, DeltaDay} 非键的传递依赖 用 3NF 消除

5) StartTimeMin -> {StartTime, DeltaDay} 非键的传递依赖 用 3NF 消除

3. TrainOfCity

候选键是{CityName, OuterStationID}, {CityName, StationName}

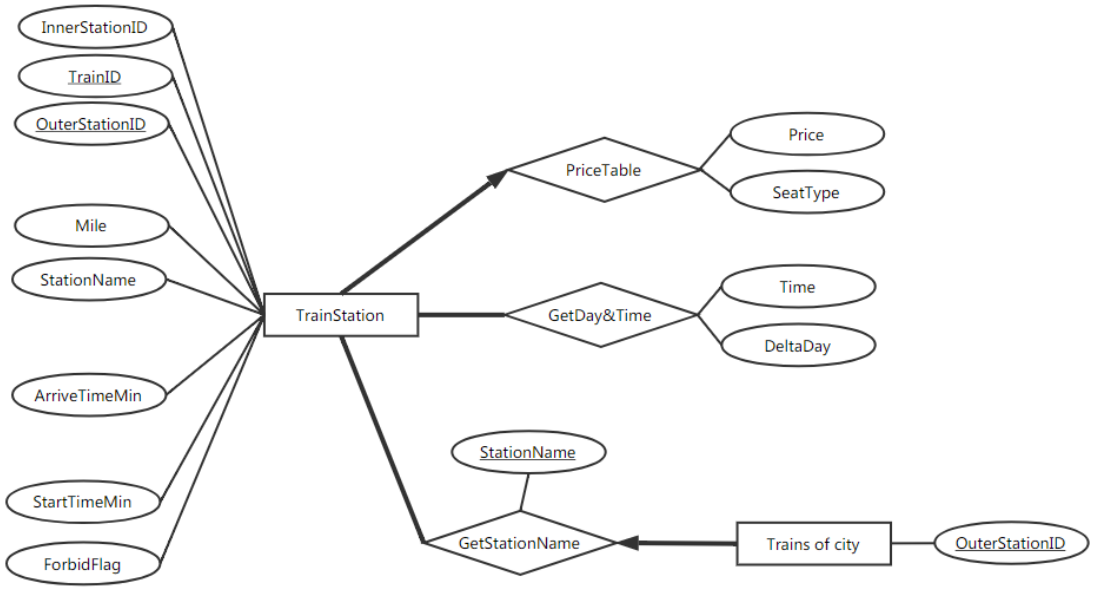
存在以下函数依赖：

1) {CityName, OuterStationID} -> 所有属性 此依赖正常。

2) {CityName, StationName} -> 所有属性 此依赖正常

3) OuterStationID -> StationName 对于键属性的函数依赖 用 BCNF 消除

处理好 Trainstation 和 Trainsofcity 的数据冗余的新的 ER 图如下所示：



4. Orders

候选键是{OrderID, TrainID}

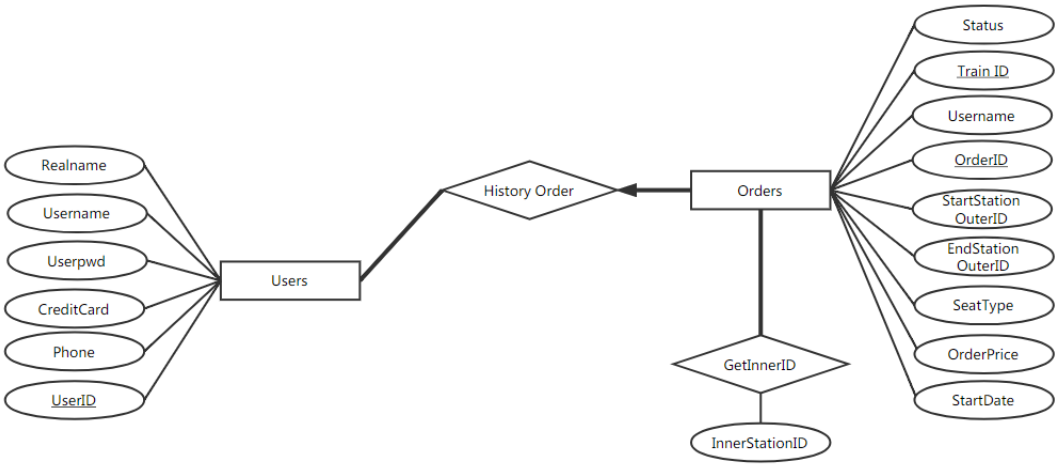
存在以下函数依赖：

1) {OrderID, TrainID} -> 所有属性 此依赖正常。

2) {TrainID, StartOuterID} -> StartInnerID 非键传递依赖 用 3NF 消除

3) {TrainID, EndOuterID} -> EndInnerID 非键传递依赖 用 3NF 消除

处理好 Orders 的数据冗余的新的 ER 图如下所示：



4. Users

候选键是{UserID}, {Phone}

存在以下函数依赖：

1) {UserID} -> 所有属性　此依赖正常。

2) {Phone} -> 所有属性　此依赖正常。

5. AvailableSeat

候选键是{TrainID, StartDate, InnerStationID}

存在以下函数依赖：

1) {TrainID, StartDate, InnerStationID} -> 所有属性　此依赖正常。

五． SQL 语句

1. 新建 OneTrip 表

记录全国所有城市之间的直达车辆信息

/*Table OneTrip*/

/*全国所有城市之间的直达车辆信息，显示起始站和结束站，和其他和车次相关的信息*/

```
/*此表单只在内部使用，不需要显示*/
select ts1.TrainID as TrainID,c1.CityName as BeginCityName,
    ts1.StationName as BeginStationName,
    ts1.OuterStationID as BeginStationOut, ts1.InnerStationID as
BeginStationID,
    ts1.StartTime BeginTime,ts2.ArriveTime as EndTime,
    ts2.StationName as EndStationName,ts2.OuterStationID as EndStationOut,
    ts2.InnerStationID as EndStationID,  c2.CityName as EndCityName,
    (ts2.ArriveTimeMin - ts1.StartTimeMin) as TripTimeMin,
    (ts1.Deltaday || 'day')  :: interval as BeginDeltaDay,
    (ts2.DeltaDay || 'day')  :: interval as EndDeltaDay,
    (ts2.PriceYZ -  ts1.PriceYZ) as PriceYZ, (ts2.PriceRZ - ts1.PriceRZ) as
PriceRZ,
    (ts2.PriceYWS - ts1.PriceYWS) as PriceYWS, (ts2.PriceYWZ -
    ts1.PriceYWZ) as PriceYWZ,
    (ts2.PriceYWX - ts1.PriceYWX) as PriceYWX, (ts2.PriceRWS -
    ts1.PriceRWS) as PriceRWS,
    (ts2.PriceRWX - ts1.PriceRWX) as PriceRWX,
    tl1.YZen, tl1.RZen , tl1.YWSen , tl1.YWZen,
    tl1.YWXen  , tl1.RWSen , tl1.RWXen
into OneTrip
from TrainStation as ts1, TrainStation as ts2, TrainsOfCity as c1, TrainsOfCity as
```

```
c2, TrainList as tl1
where     ts1.TrainID = ts2.TrainID and ts1.TrainID = tl1.TrainID and /*保证是同一
辆车*/
ts1.OuterStationID = c1.OuterStationID   and ts1.ForbidFlag = 0 and /*排除不能
上下车的站*/
 ts2.OuterStationID = c2.OuterStationID   and ts2.ForbidFlag = 0 and
 ts1.InnerStationID < ts2.InnerStationID ;/*保证 出发站 在前，到达站在后*/
```

1） 将不卖座位的价格均设置为 0

```
/*排除掉因为中途不能上下车 fobitflag = 1*/
/*导致的价格为负数                          */
/*将负数转化为正数                          */
 update OneTrip
      set PriceYz = 0
      where PriceYz < 0;
 update OneTrip
      set PriceYWS = 0
      where PriceYWS < 0;
 update OneTrip
      set PriceYWZ = 0
      where PriceYWZ < 0;
 update OneTrip
      set PriceYWX = 0
```

```
        where PriceYWX < 0;
    update OneTrip
        set PriceRZ = 0
        where PriceRZ < 0;
    update OneTrip
        set PriceRWS = 0
        where PriceRWS < 0;
    update OneTrip
        set PriceRWX = 0
        where PriceRWX < 0;
```

2. 为 OneTrip 表加入 MinPrice 属性

```
    *增加 OneTrip 一列，为最低价格                */
    /*记录价格中最低的一列                        */
    alter table OneTrip add MinPrice decimal(5,1) default 9000.0;/*价格最便宜的*/
    /*更新 MinPrice                               */
    update OneTrip
    set MinPrice = PriceYZ
    where MinPrice > PriceYZ and PriceYZ > 0;
    update OneTrip
    set MinPrice = PriceYWS
    where MinPrice > PriceYWS and PriceYWS> 0;
    update OneTrip
```

```
set MinPrice = PriceYWZ
where MinPrice > PriceYWZ and PriceYWZ > 0;
update OneTrip
set MinPrice = PriceYWX
where MinPrice > PriceYWX and PriceYWX > 0;
update OneTrip
set MinPrice = PriceRZ
where MinPrice > PriceRZ and PriceRZ > 0;
update OneTrip
set MinPrice = PriceRWS
where MinPrice > PriceRWS and PriceRWS > 0;
update OneTrip
set MinPrice = PriceRWX
where MinPrice > PriceRWX and PriceRWX > 0;
```

3. 新建 OneTicket 查询直达的火车票

```
/*Oneticket*/
/*查询一次直达的火车票，输入出发时间，起点*/
/*和终点城市，出发日期                    */
select   ot.MinPrice, ot.TripTimemin,
         ot.BeginCityName,ot.EndCityName,
         ot.TrainID,
         (ast.StartDate + ot.BeginDeltaDay) :: date as StartDate,
```

```
        (ast.StartDate + ot.EndDeltaday ):: date as EndDate,
        ot.BeginStationName,ot.EndStationName,
        ot.BeginTime, ot.EndTime,
        min(ast.YZCount) as YZCount,      min(ast.RZCount) as RZCount,
        min(ast.YWSCount) as YWSCount,
        min(ast.YWZCount) as YWZCount, min(ast.YWXCount) as YWXCount,
        min(ast.RWSCount) as RWSCount,
        min(ast.RWXCount) as RWXCount,
        ot.PriceYZ,ot.PriceRZ, ot.PriceYWS, ot.PriceYWZ, ot.PriceYWX,
        ot.PriceRWS, ot.PriceRWX,
        ot.YZen, ot.RZen , ot.YWSen , ot.YWZen,
        ot.YWXen, ot.RWSen , ot.RWXen
into OneTicket
from    OneTrip as ot, AvailableSeat as ast
where   (ot.BeginTime – time :1 ) > '0 min' and /*默认用户要求时间为 06:30：*/
        (ast.StartDate + ot.BeginDeltaDay) = : 2 and/*默认时间为 17/11/30*/
        ot.BeginCityName = : 3 and ot.EndCityName = : 4   and /*在此 sql 语句中实
        现为北京到长沙*/
        ot.TrainID = ast.TrainID and ast.InnerStationID > ot.BeginStationID and
        ast.InnerStationID <= ot.EndStationID                    /*保证内部的站的
        顺序*/
group by
```

```
        ot.BeginCityName,ot.EndCityName,
        ot.TrainID, ot.BeginDeltaday,ot.EndDeltaday,
        ast.StartDate,
        ot.BeginStationName,ot.EndStationName,
        ot.BeginTime, ot.EndTime,ot.Minprice,ot.TripTimemin,
        ot.PriceYZ,ot.PriceRZ, ot.PriceYWS, ot.PriceYWZ, ot.PriceYWX,
        ot.PriceRWS, ot.PriceRWX,ot.TripTimeMin, ot.TripTimemin,
        ot.YZen, ot.RZen , ot.YWSen , ot.YWZen,
        ot.YWXen, ot.RWSen , ot.RWXen
    order by MinPrice, TripTimemin, BeginTime/*根据最低价格，旅程时间，出发时间排序
    */
    asc limit   10 offset   0;
```

4. 新建 TransferTickets 求同日一次换乘列车

```
/*Transfertickets*/
/*查询一次换乘的火车票，输入出发时间，起点*/
/*和终点城市，出发日期                    */
/*中途换乘城市相同                        */
/*从 OneTrip 中挑选两个直达，组成换乘      */
/*同一天换乘的情况                        */
(select
        (ot1.MinPrice + ot2.MinPrice )as FinalMinPrice,
```

```
(ot2.TripTimemin + ot2.TripTimemin) as FinalTripTimeMin,

ot1.TrainID as Train1ID,

(ast1.StartDate + ot1.BeginDeltaday) :: date as Train1StartDate, (ast1.StartDate
+ ot1.EndDeltaDay) :: date as Train1EndDate,

ot1.BeginStationName,ot1.BeginCityName, ot1.MinPrice as Train1MinPrice,
ot1.TripTimemin as Train1TripTimeMin,

ot1.BeginTime as Train1BeginTime, ot1.EndTime as Train1EndTime,


ot1.EndStationName as Train1EndStationName,

ot1.EndCityName as TransferCityName,


ot2.TrainID as Train2ID,

(ast2.StartDate + ot2.BeginDeltaday):: date as Train2STartDate,(ast2.StartDate
+ ot2.EndDeltaday):: date as Train2EndDate,

ot2.BeginStationName as Train2BeginStationName,ot2.MinPrice as
Train2MinPrice, ot2.TripTimemin as Train2TripTimeMin,

ot2.EndStationName, ot2.EndCityName,

ot2.BeginTime as Train2BeginTime, ot2.EndTime as Train2EndTime,


ot1.PriceYZ as Train1PriceYZ, ot2.PriceYZ as Train2PriceYZ, ot1.PriceRZ as
Train1PriceRZ, ot2.PriceRZ as Train2PriceRZ,

ot1.PriceYWS as Train1PriceYWS, ot2.PriceYWS as Train2PriceYWS,
```

```
ot1.PriceYWZ as Train1PriceYWZ, ot2.PriceYWZ as  Train2PriceYWZ,
ot1.PriceYWX as Train1PriceYWX,ot2.PriceYWX as Train2PriceYWX,
ot1.PriceRWS as Train1PriceRWS,ot2.PriceRWS as Train2PriceRWS,
ot1.PriceRWX as Train1PriceRWX, ot2.PriceRWX as Train2PriceRWX,

min(ast1.YZCount) as Train1YZCount, min(ast1.RZCount) as
Train1RZCount,
min(ast1.YWSCount) as Train1YWSCount, min(ast1.YWZCount) as
Train1YWZCount, min(ast1.YWXCount) as Train1YWXCount,
min(ast1.RWSCount) as Train1RWSCount, min(ast1.RWXCount) as
Train1RWXCount,

min(ast2.YZCount) as Train2YZCount, min(ast2.RZCount) as
Train2RZCount,
min(ast2.YWSCount) as Train2YWSCount, min(ast2.YWZCount) as
Train2YWZCount, min(ast2.YWXCount) as Train2YWXCount,
min(ast2.RWSCount) as Train2RWSCount, min(ast2.RWXCount) as
Train2RWXCount,
ot1.YZen as Train1YZen, ot1.RZen as Train1RZen, ot1.YWSen as
Train1YWSen, ot1.YWZen as Train1YWZen,
ot1.YWXen as Train1YWXen ,ot1.RWSen as Train1RWSen, ot1.RWXen as
Train1RWXen,
```

```
ot2.YZen as Train2YZen, ot2.RZen as Train2RZen, ot2.YWSen as
Train2YWSen, ot2.YWZen as Train2YWZen,
ot2.YWXen as Train2YWXen, ot2.RWSen as Train2RWSen, ot2.RWXen as
Train2RWXen,
(ot2.BeginTime - ot1.EndTime) :: time as WaitTime
into TransferTickets
from OneTrip as ot1, OneTrip as ot2 , AvailableSeat as ast1,
AvailableSeat ast2
where
ot1.BeginCityName =: 1 and ot2.EndCityName = : 2 and /*默认地点为长沙到哈
尔滨 */
ot1.EndCityName = ot2.BeginCityName and

( (ot1.EndStationOut = ot2.BeginStationOut
and ( ((ot2.BeginTime - ot1.EndTime)> '0 min' and ((ot2.BeginTime-
ot1.EndTime ) between interval'1 hour' and interval'4 hour'))
)) or
(ot1.EndStationOut <>ot2.BeginStationOut
and ( ((ot2.BeginTime - ot1.EndTime)> '0 min' and ((ot2.BeginTime -
ot1.EndTime ) between interval'2 hour' and interval'4 hour'))
)) ) /*换乘时间关系*/
and
```

```
        (ot1.BeginTime – time：3 ) >= '0 min' and /*起始时间默认为 06:00*/
        (ast1.StartDate + ot1.BeginDeltaday) = ：4 and ( ast2.StartDate +
        ot2.BeginDeltaday) = (ast1.StartDate + ot1.EndDeltaday) and/*同一天换乘的情
        况*/
        ot1.TrainID = ast1.TrainID and ast1.InnerStationID > ot1.BeginStationID and
        ast1.InnerStationID <= ot1.EndStationID and/*从 OneTrip 中挑选*/
        ot2.TrainID = ast2.TrainID and
        (ast2.InnerStationID > ot2.BeginStationID and ast2.InnerStationID <=
        ot2.EndStationID )

group by
        ot1.BeginCityName,ot1.EndCityName ,ot2.EndCityName,
        ot1.TrainID ,ast1.StartDate, ot1.BeginDeltaday, ot2.BeginDeltaday,
        ot2.BeginDeltaday, ot1.EndDeltaday, ot2.EndDeltaday,
        ot1.BeginStationName, ot1.EndStationName , Train1MinPrice,
Train1TripTimeMin,
        ot1.BeginTime, ot1.EndTime ,

        ot2.TrainID , ast2.StartDate,
        ot2.BeginStationName ,ot2.EndStationName, Train2MinPrice,
Train2TripTimeMin,
        ot2.BeginTime , ot2.EndTime,
```

```
        ot1.PriceYZ , ot1.PriceRZ ,ot1.PriceYWS ,ot1.PriceYWZ ,ot1.PriceYWX,
        ot1.PriceRWS,
        ot1.PriceRWX,
        ot2.PriceYZ,ot2.PriceRZ , ot2.PriceYWS , ot2.PriceYWZ, ot2.PriceYWX,
        ot2.PriceRWS, ot2.PriceRWX ,
        Train1YZen, Train1RZen, Train1YWSen, Train1YWZen,
        Train1YWXen , Train1RWSen, Train1RWXen,
        Train2YZen, Train2RZen, Train2YWSen, Train2YWZen,
        Train2YWXen , Train2RWSen, Train2RWXen,
        WaitTime
)union(
/*换乘时间为隔日                                    */
        select
        (ot1.MinPrice + ot2.MinPrice )as FinalMinPrice,
        (ot2.TripTimemin + ot2.TripTimemin) as FinalTripTimeMin,
        ot1.TrainID as Train1ID,
        (ast1.StartDate + ot1.BeginDeltaday) :: date as Train1StartDate, (ast1.StartDate
        + ot1.EndDeltaDay) :: date as Train1EndDate,
        ot1.BeginStationName,ot1.BeginCityName, ot1.MinPrice as Train1MinPrice,
        ot1.TripTimemin as Train1TripTimeMin,
        ot1.BeginTime as Train1BeginTime, ot1.EndTime as Train1EndTime,
```

```
ot1.EndStationName as Train1EndStationName,
ot1.EndCityName as TransferCityName,

ot2.TrainID as Train2ID,
(ast2.StartDate + ot2.BeginDeltaday):: date as Train2STartDate,(ast2.StartDate
+ ot2.EndDeltaday):: date as Train2EndDate,
ot2.BeginStationName as Train2BeginStationName,ot2.MinPrice as
Train2MinPrice, ot2.TripTimemin as Train2TripTimeMin,
ot2.EndStationName, ot2.EndCityName,
ot2.BeginTime as Train2BeginTime, ot2.EndTime as Train2EndTime,

ot1.PriceYZ as Train1PriceYZ, ot2.PriceYZ as Train2PriceYZ, ot1.PriceRZ as
Train1PriceRZ, ot2.PriceRZ as Train2PriceRZ,
ot1.PriceYWS as Train1PriceYWS, ot2.PriceYWS as Train2PriceYWS,
ot1.PriceYWZ as Train1PriceYWZ, ot2.PriceYWZ as Train2PriceYWZ,
ot1.PriceYWX as Train1PriceYWX,ot2.PriceYWX as Train2PriceYWX,
ot1.PriceRWS as Train1PriceRWS,ot2.PriceRWS as Train2PriceRWS,
ot1.PriceRWX as Train1PriceRWX, ot2.PriceRWX as Train2PriceRWX,

min(ast1.YZCount) as Train1YZCount, min(ast1.RZCount) as
Train1RZCount,
```

min(ast1.YWSCount) as Train1YWSCount, min(ast1.YWZCount) as Train1YWZCount, min(ast1.YWXCount) as Train1YWXCount,

min(ast1.RWSCount) as Train1RWSCount, min(ast1.RWXCount) as Train1RWXCount,

min(ast2.YZCount) as Train2YZCount, min(ast2.RZCount) as Train2RZCount,

min(ast2.YWSCount) as Train2YWSCount, min(ast2.YWZCount) as Train2YWZCount, min(ast2.YWXCount) as Train2YWXCount,

min(ast2.RWSCount) as Train2RWSCount, min(ast2.RWXCount) as Train2RWXCount,

ot1.YZen as Train1YZen, ot1.RZen as Train1RZen, ot1.YWSen as Train1YWSen, ot1.YWZen as Train1YWZen,

ot1.YWXen as Train1YWXen ,ot1.RWSen as Train1RWSen, ot1.RWXen as Train1RWXen,

ot2.YZen as Train2YZen, ot2.RZen as Train2RZen, ot2.YWSen as Train2YWSen, ot2.YWZen as Train2YWZen,

ot2.YWXen as Train2YWXen, ot2.RWSen as Train2RWSen, ot2.RWXen as Train2RWXen,

(ot2.BeginTime - ot1.EndTime + time'24:00:00') :: time as WaitTime

```
from    OneTrip as ot1, OneTrip as ot2 , AvailableSeat as ast1, AvailableSeat ast2
where
        ot1.BeginCityName = :1 and ot2.EndCityName = :2 and /*默认为长沙到哈尔滨*/
        ot1.EndCityName = ot2.BeginCityName and
        (    (ot1.EndStationOut = ot2.BeginStationOut   /*隔日换乘*/
        and  (    ((ot2.BeginTime - ot1.EndTime)< '0 min'and ((ot2.BeginTime -
        ot1.EndTime+ interval'1 day') between interval'1 hour' and  interval'4 hour'))
            ))  or
            (ot1.EndStationOut <>ot2.BeginStationOut
        and  (      ((ot2.BeginTime - ot1.EndTime)< '0 min'and ((ot2.BeginTime -
        ot1.EndTime+ interval'1 day') between interval'2 hour' and  interval'4 hour'))
            ) ))
        and

        (ot1.BeginTime -  time :3 ) >= '0 min' and /*出发时间默认为 06:00*/
        (ast1.StartDate + ot1.BeginDeltaday) =: 4 and      ( ast2.StartDate +
        ot2.BeginDeltaday) =(ast1.StartDate + ot1.EndDeltaday + '1 day') and
        ot1.TrainID = ast1.TrainID and ast1.InnerStationID > ot1.BeginStationID and
        ast1.InnerStationID <= ot1.EndStationID and
        ot2.TrainID = ast2.TrainID and
        (ast2.InnerStationID > ot2.BeginStationID and ast2.InnerStationID <=
        ot2.EndStationID )
```

```
group by
        ot1.BeginCityName,ot1.EndCityName ,ot2.EndCityName,
        ot1.TrainID ,ast1.StartDate, ot1.BeginDeltaday, ot2.BeginDeltaday,
        ot1.EndDeltaday, ot2.EndDeltaday,
        ot1.BeginStationName, ot1.EndStationName , Train1MinPrice,
Train1TripTimeMin,
        ot1.BeginTime, ot1.EndTime ,


        ot2.TrainID , ast2.StartDate,
        ot2.BeginStationName ,ot2.EndStationName, Train2MinPrice,
Train2TripTimeMin,
        ot2.BeginTime , ot2.EndTime,
        ot1.PriceYZ , ot1.PriceRZ ,ot1.PriceYWS ,ot1.PriceYWZ ,ot1.PriceYWX,
        ot1.PriceRWS,
        ot1.PriceRWX,
        ot2.PriceYZ, ot2.PriceRZ , ot2.PriceYWS , ot2.PriceYWZ, ot2.PriceYWX,
        ot2.PriceRWS, ot2.PriceRWX ,
        Train1YZen, Train1RZen, Train1YWSen, Train1YWZen,
        Train1YWXen , Train1RWSen, Train1RWXen,
        Train2YZen, Train2RZen, Train2YWSen, Train2YWZen,
        Train2YWXen , Train2RWSen, Train2RWXen,
```

```
        WaitTime
)
order by FinalMinPrice, FinalTripTimeMin, Train1BeginTime, Train2BeginTime/*根据最低
价格，旅程时间，出发时间排序*/
asc limit 10 offset 0;
```

5. 按车次查询站点和余座信息

```
(
select   ts.InnerStationID, ts.StationName, ts.ArriveTime, ts.StartTime,
        ts.PriceYZ, (ast.YZCount) as YZCount,
        ts.PriceRZ, (ast.RZCount) as RZCount,
        ts.PriceYWS, (ast.YWSCount) as YWSCount,
        ts.PriceYWZ, (ast.YWZCount) as YWZCount,
        ts.PriceYWX, (ast.YWXCount) as YWXCount,
        ts.PriceRWS, (ast.RWSCount) as RWSCount,
        ts.PriceRWX, (ast.RWXCount) as RWXCount,
        tl.YZEn, tl.RZEn, tl.YWSEn, tl.YWZEn, tl.YWXEn, tl.RWSEn, tl.RWXEn, /*  输出
这列车是否开了某座位类型  交给网页输出'-' */
        ts.ForbidFlag /*是否在这站不售票  如果是  网页输出  '-' */
from TrainStation as ts, AvailableSeat as ast , TrainList as tl
where ( ts.TrainID = 'G351' and
        ast.TrainID = 'G351' and
```

```
            tl.TrainID = 'G351' and
            ast.StartDate = '2017/11/26' and
            ast.InnerStationID = 1 and /*单独处理起始站的情况(不能买到起始站的票) 网页
输出全为'-' */
            ts.InnerStationID = 1
            )
order by ts.InnerStationID
) /*上一部分完全是因为起始站需要单独处理余座的情况*/
union
(
select   ts.InnerStationID, ts.StationName, ts.ArriveTime, ts.StartTime,
            ts.PriceYZ, min(ast.YZCount) as YZCount,
            ts.PriceRZ, min(ast.RZCount) as RZCount,
            ts.PriceYWS, min(ast.YWSCount) as YWSCount,
            ts.PriceYWZ, min(ast.YWZCount) as YWZCount,
            ts.PriceYWX, min(ast.YWXCount) as YWXCount,
            ts.PriceRWS, min(ast.RWSCount) as RWSCount,
            ts.PriceRWX, min(ast.RWXCount) as RWXCount,
            tl.YZEn, tl.RZEn, tl.YWSEn, tl.YWZEn, tl.YWXEn, tl.RWSEn, tl.RWXEn,
            ts.ForbidFlag
from TrainStation as ts, AvailableSeat as ast , TrainList as tl
where ( ts.TrainID = 'G351' and
```

```
        ast.TrainID = 'G351' and
        tl.TrainID = 'G351' and
        ast.StartDate = '2017/11/26' and
        ((ast.InnerStationID <= ts.InnerStationID and /*在余座表大于起点(第 1 站)小于第
i 站的情况下求所有余座 Count 的最小值,即这条路段的最小值*/
        ast.InnerStationID > 1)
        ))
group by ts.InnerStationID, ts.StationName, ts.ArriveTime, ts.StartTime,
         ts.PriceYZ, ts.PriceRZ, ts.PriceYWS, ts.PriceYWZ, ts.PriceYWX,
         ts.PriceRWS, ts.PriceRWX,
         tl.YZEn, tl.RZEn, tl.YWSEn, tl.YWZEn, tl.YWXEn, tl.RWSEn, tl.RWXEn,
         ts.ForbidFlag
order by ts.InnerStationID
);
```

6. 查询历史订单

```
select OrderID, Orders.StartDate, TS1.StationName, TS2.StationName, Sum(OrderPrice),
Status, TS1.StartTime
from Orders, TrainStation as TS1, TrainStation as TS2
where TS1.OuterStationID = Orders.StartStationOuterID and
        TS2.OuterStationID = Orders.EndStationOuterID and
        TS1.TrainID = Orders.TrainID and
        TS2.TrainID = Orders.TrainID and /*限定是 orders 里才有的车次*/
```

```
        Orders.UserId = 'lgd' and
        Orders.StartDate >= '2017-11-25' and /*查询范围是起点到终点*/
        Orders.StartDate <= '2017-11-25'
group by OrderID, Orders.StartDate, TS1.StationName, TS2.StationName, Status
order by OrderID;
```

7. 管理员信息

```
/*9_1 总订单数*/
select count(distinct OrderID)
from orders;
/*9_2 总票价*/
select sum(OrderPrice)
from orders;
/*9_3 最热点车次排序*/
select TrainID, count(*) as OrderTimes
from orders
group by TrainID
order by OrderTimes desc limit 10;
/*9_4 当前注册用户列表*/
select *
from users;
/*9_5 查看每个用户的订单*/
```

/*refers to req_8*/

8. 冲突列车查询（一个用户不能同时坐两趟车）

/* 输入：当前 userid trainid startinnerstationid endinnerstationid whichday st1->st2*/

/* 输出 orderid trainid startstationname endstationname startdate starttime arrivetime st3->st4*/

```
select od.orderid, od.trainid, ts3.stationname, ts4.stationname, od.startdate, ts3.starttime,
ts4.arrivetime
from orders as od,
    trainstation as ts1, trainstation as ts2, trainstation as ts3, trainstation as ts4
where
    od.userid = 'lyw' and
    ts1.trainid = 'G1' and ts2.trainid = 'G1' and
    od.status != -1 and
    od.startdate + ts1.deltaday = '2017-11-30' and
    ts3.trainid = od.trainid and ts4.trainid = od.trainid and
    ts1.innerstationid = '1' and
    ts2.innerstationid = '2' and
    ts3.innerstationid = od.startstationinnerid and
    ts4.innerstationid = od.endstationinnerid and
    (
```

```
      ((ts1.starttimemin >= ts3.starttimemin) and (ts1.starttimemin   <= ts4.arrivetimemin))
or   /*st3---st1--->st4   (st2) */
      ((ts2.starttimemin >= ts3.starttimemin) and (ts2.arrivetimemin <= ts4.arrivetimemin))
or   /*(st1)   st3---st2--->st4 */
      ((ts1.starttimemin <= ts3.starttimemin) and (ts4.arrivetimemin <= ts2.arrivetimemin))
/*st1   st3------>st4     st2*/
      )
;
```