**Schema**

Station  
表示所有车站组成的实体集

ID：车站编号，主键，可选择all\_stations.txt前面的数字

Name：车站名

City：所在的城市名

Train  
表示所有车次的列车组成的实体集

Number：车次号，如G101

STime：起始时间，如G101的06:44

ETime：终止时间，如G101的12:38

SStation：首发站的ID

EStation：终点站的ID

Schedule  
排班/计划表，单条记录表示某趟列车会到达某站，在该站会提供某种席次的座位

ID：排班表中某一条目的编号，主键

//Key Constraint + Total participation: Train.Number和Station.ID为外键

SeatType：提供的某种席次，如G101在德州站所提供的、144.5元的硬座

TotalPrice：该列车从始发站到该站的票价，如上面的144.5

TotalTime：历时，如G101中到达德州东的81（分）

ArriveTime：到达该站的时刻，如G101到达德州东的08:05

DepartTime：从该站发车的时刻，如G101从德州东发出的08:13

Rank：该站排在该趟列车的第几站，如G101德州东前面的3（为站点排序）

Ticket  
票池，管理余票和已经售出的票

ID：票号，主键

Date：发车日期

WhichSeat：哪一座位，取值{A, B, C, D, E, F}，乘客可以购买前5种，即某一席次下的5张票，F为不可购买，即某些站下的横杠

Status：该段票是否售出，Available或Sold

Int\_Start：该张票的区间开始端，外键引用Schedule的ID，表示某趟列车的某席位在发车日期为Date这天，从Int\_Start到Int\_End这段可以购买或者已经售出（状态为Status）

Int\_End：该张票的区间结束端，外键引用Schedule的ID

比如，G101次列车在4月24日的硬座席位从北京南到上海虹桥的Schedule.ID依次为21~31，初始时有一张Ticket.ID=40的票，它的Int\_Start为21，Int\_End为31，有人买了这张票的泰安到徐州东这段，那么从票库中删除Ticket.ID=40的这张票，根据泰安在G101中的Rank为5，徐州东的Rank为7，找到它们对应的Schedule.ID=25, 27，还需在票库中增加三张Status分别为Available, Sold, Available的票

退票时需要检查该ID的Ticket的前后段是否可以合并

Order  
订单，可以包含两张票

ID：订单号，主键

Status：订单状态，Normal或Canceled，如果订单中的票的Date都在查询的时刻之后，则可以取消

HasTicket  
票和订单的联系，每张票至多有一个订单与之对应，每个订单中至多含有两张票

外键：Ticket.ID, Order.ID

Passenger  
乘客

Phone：电话，主键

User：用户名，候选键

Name：姓名

History  
用户的历史订单，用户和订单的联系，每个订单有且仅有一个用户与之对应

外键：Order.ID, Passenger.Phone