余票表

remain\_ticket

(date, train\_num, departure, destination,

business\_remain, first\_remain, second\_remain, noSeat\_remain)

路线表 车次-站名-站号

route

(train\_num, station\_name, station\_num)

车厢

carriage

(type, name, seat\_quantitiy)

列车表 车次-车厢

train

(train\_num, business\_num, first\_num, second\_num)

列车运行时刻表

timetable

(train\_num, station\_name, station\_num, arrive\_time, departure\_time)

不同类型车厢的座位数和几排几座一一对应，建seat 表

business\_seat

(seat\_num, row, column)

first\_seat

(seat\_num, row, column)

second\_seat

(seat\_num, row, column)

ticket

每列车有8/16个车厢，一个车厢对应一种座位类型，无座只在二等座车厢出售，商务座/一等座/二等座/无座可以按40/80/100/20来设定，也可以自定义，数目不要太离奇就可以。

购票

选择：日期，出发地，目的地

显示符合条件的车次信息

筛选route表，返回所有满足的车次，及其在timetable中存储的对应出发时间和到达时间（出发时间：利用departure和train\_num查找timetable，查看departure\_time，到达时间：利用destination和train\_num查找timetable，查看arrive\_time）

选择：上述返回结果列表中某一车次

余票查询：

根据日期、车次、出发地、目的地，查找remain\_ticket表，返回其对应的商务座、一等座、二等座，无座的剩余票数

选择购票：日期、车次、出发地、目的地，座位类型

此时需要加锁！处理并发！

余票更新：

查询route表，将出发地与目的地之间的所有车站取出，为集合A。

事务开始 加锁FOR UPDATE

遍历remain\_ticket表，

将该车次对应的所有列表项中

1. departure和destination中任意一个出现A中站名的
2. departure为出发地站名
3. destination为目的地站名

筛选出的所有项中对应的座位类型中余票

检查所有的余票都是>0的

自减，更新数据库

UPDATE完毕 事务提交

返回购票成功的信息

购票成功打印车票：

日期、车次、出发时间、出发地站名、目的地站名，票价，自动分配座位

票价如何计算？

商务座票价=1.234\*里程

一等座票价 = 0.775×里程

二等座票价 = 0.485×里程

里程数=timetable中（destination对应的distance-departure对应的distance）

如何根据余票张数计算得到座位？

车厢号=（某类型车厢的总座位数-该车厢类型的余票数）／一个该类型车厢的座位数+1+该类型车厢之前的其他类型车厢总数

座位号=（某类型车厢的总座位数-该车厢类型的余票数）%一个该类型车厢的座位数+1

查对应车厢类型的seat表，返回（row，column）

1. 建表
2. 填入数据
3. 测试余票