

# 数据库工程作业

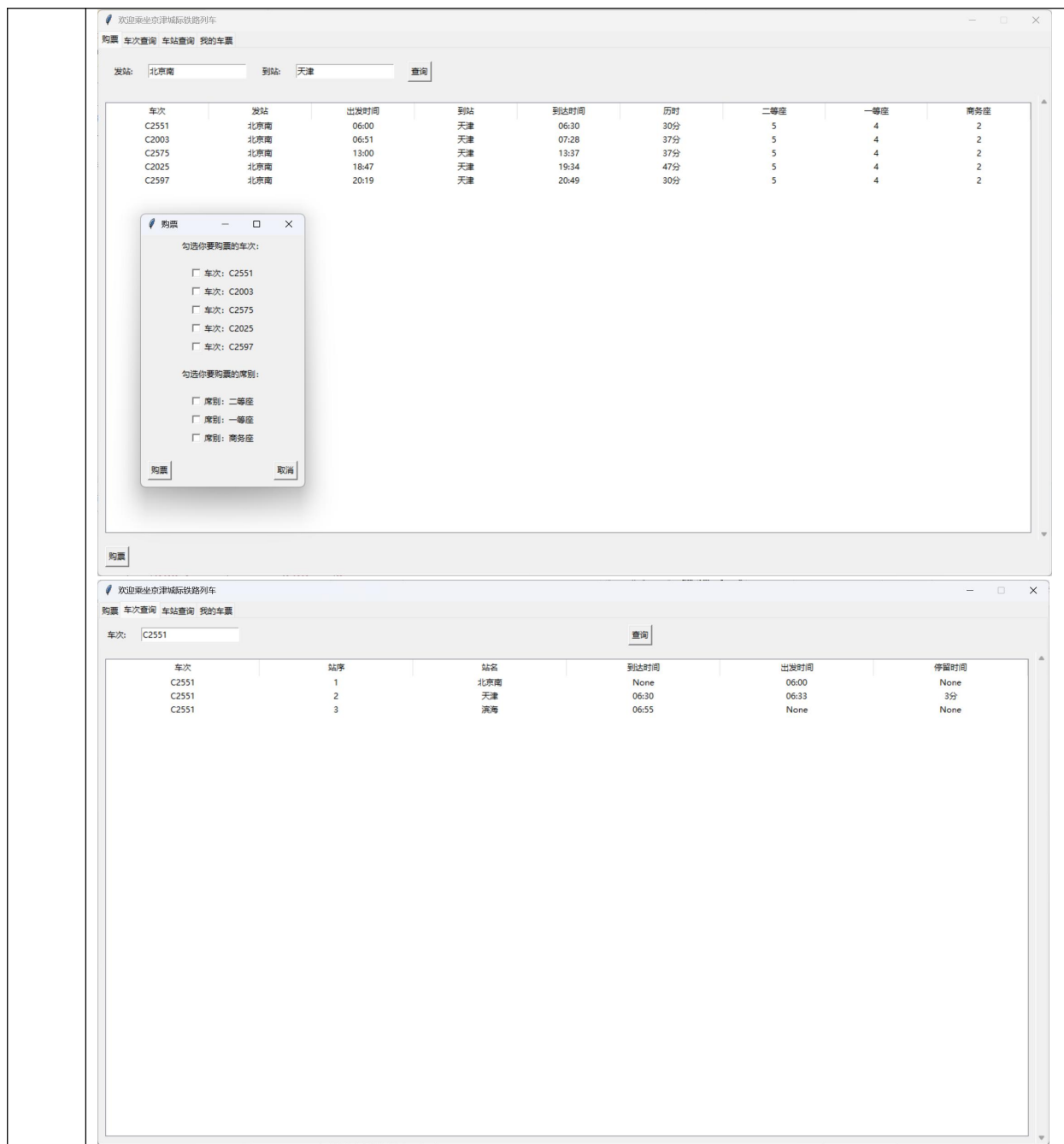
## 要求:

1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
2. 开发模式(B/S或C/S)、开发高级语言任选,后台数据库使用大型数据库管理系统(SQL Server、Oracle、MySQL等)，**不要使用桌面数据库**。
3. 报告中所列举的四种操作，**每种操作举一个例子即可**。
4. 作业成绩按照报告中的标准评分，**程序只实现报告中涉及的部分即可**。
5. 作业完成后，**请将工程作业报告和程序打包提交给助教老师，并联系助教老师进行系统说明和演示，回答相关问题**。

## 工程作业报告

### 1. 项目信息（10分）

学号	2313546	姓名	蒋衲言	专业	信息安全
项目名称	京津城际铁路列车模拟订票系统				
必备环境	Python, MySQL				
系统主要功能简介（4分）	该项目实现了一个简易版的模拟京津城际铁路列车订票系统，主要功能有注册身份信息、模拟购票和退票、输入车站名查询该车站的车次信息、输入车次查询该车次的时刻表。				
系统主要页面截图（6分）	<p>截 3-4 个页面即可</p> 				



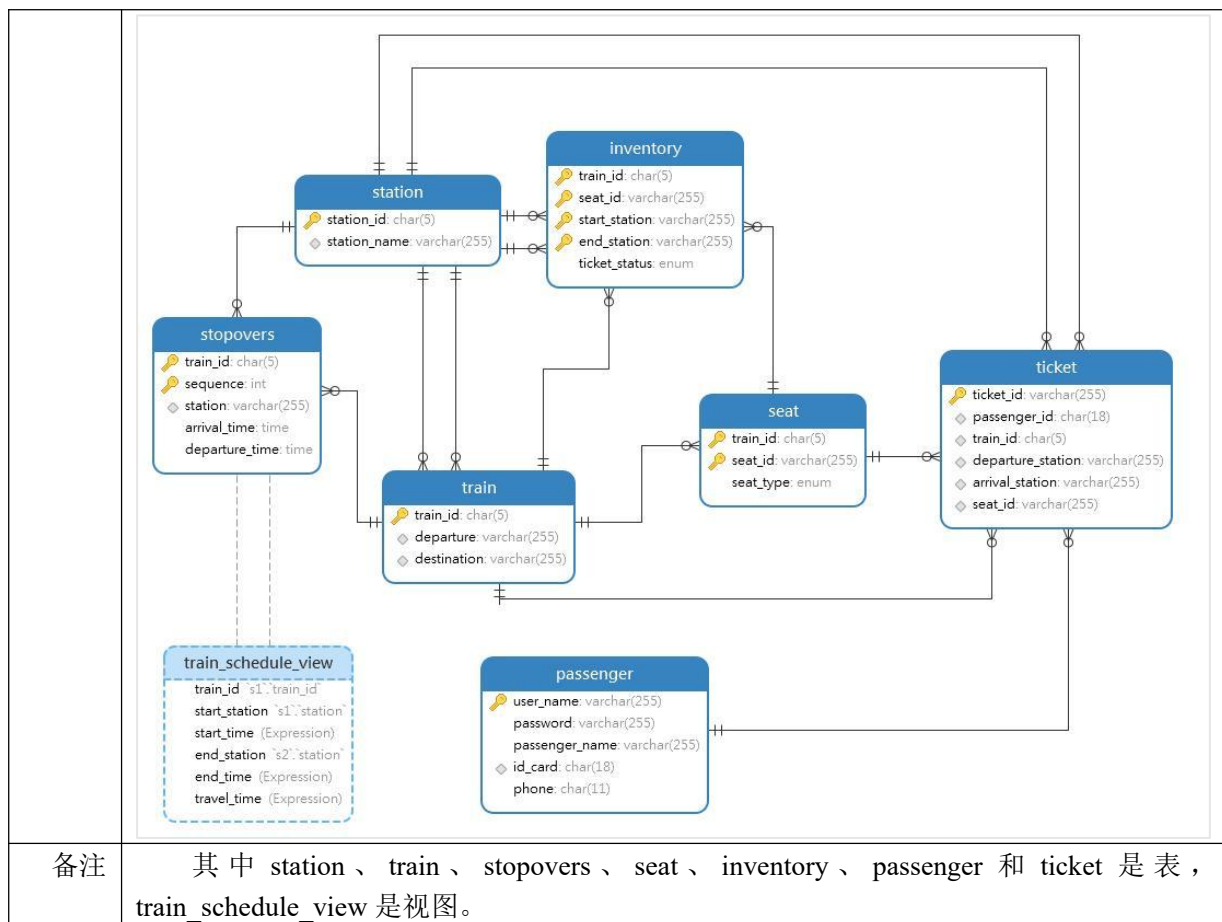
## 2. 系统配置（10 分）

说明		(2 分) 请说明系统配置情况（后台数据库，高级语言）； (8 分) 请使用连接串连接高级语言和数据库，并分析字符串的各个部分。		
配置 步骤 2 分	DBMS	MySQL		
	高级语言	Python		
连接串 分析 (6 分)	序号	名称	功能说明	取值
	1	connect_db()	在 login_register.py 中与 MySQL 进行连接,从而	无

			进行对应的注册、登录、修改密码等操作。	
	2	connect_db()	在 ticket.py 中与 MySQL 进行连接，从而进行对应的购票、退票等操作。	无
连接串代码 (截屏) (2 分)	<pre># 连接数据库 def connect_db(): 8 用法     return pymysql.connect(         host='localhost',         user='root',         password='c[REDACTED]',         database='ticket_booking'     )</pre>			
备注	数据库主机名叫 localhost，用户名叫 root，数据库名叫 ticket_booking，密码这里就不展示了。			

### 3. 数据库设计 (14 分)

说明	<p>（10 分）按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“（字段 1，字段 2，……，字段 n）”的形式给出，被参照字段以“表名（字段 1，字段 2，……，字段 n）”的形式给出；</p> <p>（4 分）一般 DBMS 都可以为数据库生成关系图，请将该图片截屏并粘贴到表格中。</p>				
数据表 （10）	创建顺序	数据表名称	主键	参照属性	被参照表及属性
	1	station	station_id	无	
	2	train	train_id	departure	station.station_name
				destination	station.station_name
	3	stopovers	(train_id,sequence)	train_id	train.train_id
				station	station.station_name
	4	seat	(train_id,seat_id)	train_id	train.train_id
	5	inventory	(train_id,seat_id,start_station,end_station,ticket_status)	train_id	train.train_id
				seat_id	seat.seat_id
				start_station	station.station_name
				end_station	station.station_name
	6	passenger	user_name	无	
	7	ticket	ticket_id	passenger_id	passenger.id_card
				train_id	train.train_id
				departure_station	station.station_name
arrival_station				station.station_name	
seat_id				seat.seat_id	
关系图 （4）	（截屏）				



备注 其中 station、train、stopovers、seat、inventory、passenger 和 ticket 是表，train\_schedule\_view 是视图。

#### 4. 含有事务应用的删除操作（13 分）

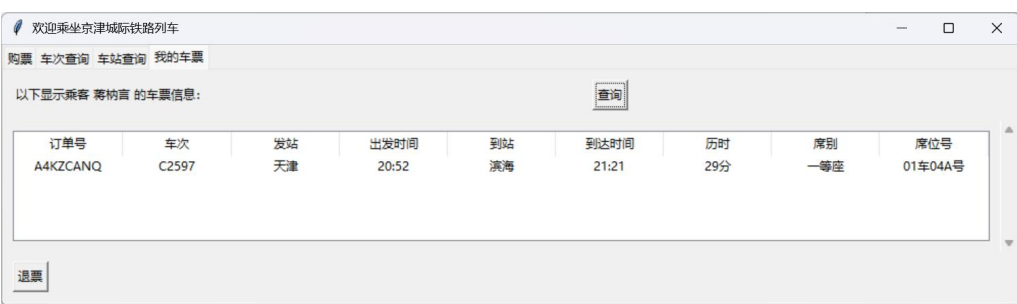
说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（2 分）该操作会涉及的表（必须含有两张或两张以上的关系表，同时以“表名”的形式给出）</p> <p>（1 分）表连接涉及字段描述（描述方式为“表 1.属性=表 2.属性”）</p> <p>（1 分）删除条件涉及的字段描述(以“表名.属性=? ”形式给出)</p> <p>（4 分）实现该操作的关键代码（高级语言、SQL），截图即可；（其中如果删除语句中不包含任何形式的事务应用将扣除 3 分）</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>	
功能描述 (1 分)	<p>该操作主要实现退票功能，用户可在“我的车票”里查询已购车票，点击下方“退票”按钮，选择对应的车票即可进行退票。退票后，数据库里相应的余票信息也会发生改变。</p>	
涉及的表 (2 分)	ticket,inventory	
表连接涉及字段 (1 分)	<p>ticket.train_id=inventory.train_id</p> <p>ticket.departure_station=inventory.start_station</p> <p>ticket.arrival_station=inventory.end_station</p> <p>ticket.seat_id=inventory.seat_id</p>	
删除条件 字段描述 (1 分)	字段	规则
	ticket.ticket_id	选中订单号 ticket_id 后进行退票，在表 ticket 按照 ticket_id 属性进行索引，删除相应行。
代码 (4 分)	<p>1. 退票操作的关键 Python 代码</p> <pre> if messagebox.askyesno("确认退票", f"是否确认退票以下订单? \n{' '.join(selected_tickets)}"):     conn = connect_db()     cursor = conn.cursor()     try:         # 开始事务         cursor.execute("START TRANSACTION")         tid = selected_tickets[0]         updateRefundSQL = """             UPDATE             inventory             SET             ticket_status = 'available'             WHERE             train_id = (SELECT train_id FROM ticket WHERE ticket_id = %s)             AND             seat_id = (SELECT seat_id FROM ticket WHERE ticket_id = %s)             AND             (SELECT sequence FROM stopovers WHERE station = start_station             AND train_id = (SELECT train_id FROM ticket WHERE ticket_id = %s)) &gt;=             (SELECT sequence FROM stopovers WHERE station = (SELECT departure_station FROM ticket WHERE ticket_id = %s)             AND train_id = (SELECT train_id FROM ticket WHERE ticket_id = %s))             AND             (SELECT sequence FROM stopovers WHERE station = end_station             AND train_id = (SELECT train_id FROM ticket WHERE ticket_id = %s)) &lt;=             (SELECT sequence FROM stopovers WHERE station = (SELECT arrival_station FROM ticket WHERE ticket_id = %s)             AND train_id = (SELECT train_id FROM ticket WHERE ticket_id = %s));             """         cursor.execute(updateRefundSQL, (tid, tid, tid, tid, tid, tid, tid, tid))         cursor.execute("DELETE FROM ticket WHERE ticket_id = %s", (tid,))         messagebox.showinfo("提示", "退票成功")         conn.commit() # 成功-&gt;事务提交         popup.destroy() # 关闭弹窗         search_user_ticket() # 刷新主界面     except Exception as e:         conn.rollback() # 失败-&gt;事务回滚         messagebox.showerror("错误", f"退票失败: {e}") </pre>	

2. 退票操作的触发器（用于更新余票信息）

名称	触发	插入	更新	删除
update_inventory_after_refund	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

```
1 BEGIN
2 CREATE TEMPORARY TABLE tmp1 AS
3 SELECT * FROM inventory;
4
5 CREATE TEMPORARY TABLE tmp2 AS
6 SELECT * FROM inventory;
7
8 CREATE TEMPORARY TABLE tmp3 AS
9 SELECT * FROM inventory;
10
11 CREATE TEMPORARY TABLE tmp4 AS
12 SELECT * FROM inventory;
13
14
15 UPDATE inventory
16 SET ticket_status = 'available'
17 WHERE train_id = OLD.train_id
18 AND seat_id = OLD.seat_id
19 AND (
20 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = OLD.train_id) <=
21 (SELECT sequence FROM stopovers WHERE station = OLD.departure_station AND train_id = OLD.train_id)
22 AND
23 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = OLD.train_id) <=
24 (SELECT sequence FROM stopovers WHERE station = OLD.arrival_station AND train_id = OLD.train_id)
25 AND
26 (SELECT ticket_status FROM tmp1 WHERE train_id = OLD.train_id AND seat_id = OLD.seat_id AND tmp1.start_station = inventory.start_station
27 AND tmp1.end_station = OLD.departure_station) = 'available')
28
29 OR
30 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = OLD.train_id) >=
31 (SELECT sequence FROM stopovers WHERE station = OLD.departure_station AND train_id = OLD.train_id)
32 AND
33 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = OLD.train_id) >=
34 (SELECT sequence FROM stopovers WHERE station = OLD.arrival_station AND train_id = OLD.train_id)
35 AND
36 (SELECT ticket_status FROM tmp2 WHERE train_id = OLD.train_id AND seat_id = OLD.seat_id AND tmp2.start_station = OLD.arrival_station
37 AND tmp2.end_station = inventory.end_station) = 'available')
38
39 OR
40 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = OLD.train_id) <=
41 (SELECT sequence FROM stopovers WHERE station = OLD.departure_station AND train_id = OLD.train_id)
42 AND
43 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = OLD.train_id) >=
44 (SELECT sequence FROM stopovers WHERE station = OLD.arrival_station AND train_id = OLD.train_id)
45 AND
46 (SELECT ticket_status FROM tmp3 WHERE train_id = OLD.train_id AND seat_id = OLD.seat_id AND tmp3.start_station = inventory.start_station
47 AND tmp3.end_station = OLD.departure_station) = 'available')
48
49 OR
50 (SELECT ticket_status FROM tmp4 WHERE train_id = OLD.train_id AND seat_id = OLD.seat_id AND tmp4.start_station = OLD.arrival_station
51 AND tmp4.end_station = inventory.end_station) = 'available')
52
53
54
```

使用乘客蒋柄言的账号进行登录，查询车票信息，发现有 C2597 次天津至滨海一等座，数据库表 ticket 里也有相应行。



程序演示  
(4 分)

ticket_id	passenger_id	train_id	departure_station	arrival_station	seat_id
A4KZCANQ	510703200501010101	C2597	天津	滨海	01车04A号

表 inventory 里相应席位也显示 “sold”。

train_id	seat_id	start_station	end_station	ticket_status
C2597	01车04A号	北京南	塘沽	sold
C2597	01车04A号	北京南	天津	available
C2597	01车04A号	北京南	滨海	sold
C2597	01车04A号	塘沽	滨海	sold
C2597	01车04A号	天津	塘沽	sold
C2597	01车04A号	天津	滨海	sold

进行退票：

退票

勾选你要退票的订单号:

☒ 订单号: A4KZCANQ

退票取消

提示

退票成功

确定

再次查询车票信息，发现该车票已经消失，数据库表 ticket 里相应行也被删除了。

欢迎乘坐京津城际铁路列车

购票 车次查询 车站查询 我的车票

以下显示乘客 蒋纳言 的车票信息:

查询

订单号	车次	发站	出发时间	到站	到达时间	历时	席别	席位号
-----	----	----	------	----	------	----	----	-----

退票

ticket_id	passenger_id	train_id	departure_station	arrival_station	seat_id
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

表 inventory 里相应席位也更新为“available”。

train_id	seat_id	start_station	end_station	ticket_status
C2597	01车04A号	北京南	塘沽	available
C2597	01车04A号	北京南	天津	available
C2597	01车04A号	北京南	滨海	available
C2597	01车04A号	塘沽	滨海	available
C2597	01车04A号	天津	塘沽	available
C2597	01车04A号	天津	滨海	available

备注

以上是一个完整的事务操作，不可分割。任何一个环节失败则整个事务进行回滚；否则提交。



## 5. 触发器控制下的添加操作（20 分）

说明	(1 分) 简要说明该操作所要完成的功能; (2 分) 简要说明该触发器所要完成的功能 (1 分) 该操作会涉及的表 (以“表名”的形式给出)。 (2 分) 该操作输入数据以及输入数据应该满足的条件, 如: 数值范围、是否为空; (6 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可; (8 分) 如何执行该操作, 按所述方法能够正常演示程序则给分。	
功能描述 (1 分)	该操作主要实现购票功能, 用户根据出发地和目的地进行车次查询, 点击“购票”, 选择车次和席别, 系统将随机生成订单号和座位号, 将车票的信息添加到数据库中。购票后, 数据库里相应的余票信息也会发生改变。若余票不足时和同一人买相同车次、相同席别、相同出发地和目的地的车票时, 无法购票成功。	
触发器描述 (2 分)	1. no_duplicate_ticket 该触发器主要是用于避免购相同的车票, 当同一人买相同车次、相同席别、相同出发地和目的地的车票时, 抛出错误, 无法购票成功。 2. update_inventory_after_book 该触发器主要是用于更新余票状态, 当购买 X 车次 A 至 B 区间的座位 S 时, X 车次的座位 S 在与 A 至 B 区间有交集的区间上均显示 “sold”。	
涉及的表 (1 分)	ticket,inventory	
输入数据 (2 分)	字段	规则
	ticket_id	随机生成的不重复的 8 位大写字母和数字组成的字符串
	passenger_id	均不为空, 并且都有外键约束
	train_id	
	departure_station	
	arrival_station	
	seat_id	
插入操作源码 (3 分)	<pre>1. 判断是否还有余票 if messagebox.askyesno("确认购票", f"是否确认购票以下车次席别? \n{'', ' '.join(selected_trains)} {'', ' '.join(selected_seatType)}"):     train_no = selected_trains[0]     seat_type = selected_seatType[0]     conn = connect_db()     cursor = conn.cursor()     queryInventorySQL = """         SELECT seat_id         FROM inventory NATURAL JOIN seat         WHERE train_id = %s AND start_station = %s AND end_station = %s             AND seat_type = %s AND ticket_status = 'available';     """     cursor.execute(queryInventorySQL, (train_no, dep_station, arr_station, seat_type))     results_seat = cursor.fetchall()     if len(results_seat) == 0:         messagebox.showerror("错误", "购票失败: 余票不足!")         return  2. 生成订单号、分配座位号、添加购票信息</pre>	



	<pre>try:     # 开始事务     cursor.execute("START TRANSACTION")     book_seat = results_seat[0][0] # 取出座位号      # 生成随机订单号     def generate_random_string(length=8):         whetherDuplicate = True         chars = "AAAAAAAA"         while whetherDuplicate:             chars = string.ascii_letters + string.digits # 包含大小写字母和数字             chars = chars.upper()             cursor.execute("SELECT * FROM ticket WHERE ticket_id = %s", chars)             results = cursor.fetchall()             if len(results) == 0:                 whetherDuplicate = False          return ".join(random.choices(chars, k=length))      random_str = generate_random_string()      cursor.execute("SELECT id_card FROM passenger WHERE user_name = %s", username)     pID = cursor.fetchall()[0][0]      cursor.execute("INSERT INTO ticket VALUES (%s,%s,%s,%s,%s,%s)",                    (random_str, pID, train_no, dep_station, arr_station, book_seat))     messagebox.showinfo("提示", "购票成功")     conn.commit() # 成功-&gt;事务提交     popup.destroy() # 关闭弹窗     search_ticket() # 刷新主界面 except Exception as e:     conn.rollback() # 失败-&gt;事务回滚     messagebox.showerror("错误", f"购票失败: {e}")</pre>																					
触发器源 码 (3分)	<p>1. no_duplicate_ticket</p> <table><tr><th>名称</th><th>触发</th><th>插入</th><th>更新</th><th>删除</th></tr><tr><td>I no_duplicate_ticket</td><td>BEFORE</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>update_inventory_after_book</td><td>AFTER</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>update_inventory_after_refund</td><td>AFTER</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input checked="" type="checkbox"/></td></tr></table> <p>定义</p> <pre>1 BEGIN 2 IF EXISTS( 3     SELECT * FROM ticket WHERE passenger_id = NEW.passenger_id AND train_id = NEW.train_id 4     AND departure_station = NEW.departure_station 5     AND arrival_station = NEW.arrival_station 6 ) 7 THEN 8     SIGNAL SQLSTATE '45000' 9     SET MESSAGE_TEXT = 'Error: You are trying to book a duplicate ticket!'; 10 END IF; 11 END</pre> <p>2. update_inventory_after_book</p>	名称	触发	插入	更新	删除	I no_duplicate_ticket	BEFORE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	update_inventory_after_book	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	update_inventory_after_refund	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
名称	触发	插入	更新	删除																		
I no_duplicate_ticket	BEFORE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																		
update_inventory_after_book	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																		
update_inventory_after_refund	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>																		

名称	触发	插入	更新	删除
no_duplicate_ticket	BEFORE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
update_inventory_after_book	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
update_inventory_after_refund	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

定义

```
1 BEGIN
2 UPDATE inventory
3 SET ticket_status = 'sold'
4 WHERE train_id = NEW.train_id
5 AND seat_id = NEW.seat_id
6 AND (
7 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = NEW.train_id) <=
8 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id)
9 AND
10 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) <=
11 (SELECT sequence FROM stopovers WHERE station = NEW.arrival_station AND train_id = NEW.train_id)
12 AND
13 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) >
14 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id))
15 OR
16 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = NEW.train_id) >=
17 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id)
18 AND
19 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) >=
20 (SELECT sequence FROM stopovers WHERE station = NEW.arrival_station AND train_id = NEW.train_id)
21 AND
22 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) <
23 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id))
24 OR
25 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = NEW.train_id) <=
26 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id)
27 AND
28 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) >=
29 (SELECT sequence FROM stopovers WHERE station = NEW.arrival_station AND train_id = NEW.train_id))
30 OR
31 ((SELECT sequence FROM stopovers WHERE station = start_station AND train_id = NEW.train_id) >=
32 (SELECT sequence FROM stopovers WHERE station = NEW.departure_station AND train_id = NEW.train_id)
33 AND
34 (SELECT sequence FROM stopovers WHERE station = end_station AND train_id = NEW.train_id) <=
35 (SELECT sequence FROM stopovers WHERE station = NEW.arrival_station AND train_id = NEW.train_id))
36 );
37 END
```

说明：不违背触发器能够执行插入操作。  
还是登录乘客蒋杬言的账号，查询武清至天津的车票，假如购买 C2575 次商务座。

购票 车次查询 车站查询 我的车票

发站: 武清 到站: 天津 查询

车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座
C2575	武清	13:23	天津	13:37	14分	5	4	2
C2025	武清	19:20	天津	19:34	14分	5	4	2

购票

程序演示  
(4 分)

购票

勾选你要购票的车次:

☒ 车次: C2575  
☐ 车次: C2025

勾选你要购票的席别:

☐ 席别: 二等座  
☐ 席别: 一等座  
☒ 席别: 商务座

购票 取消

购票成功，再次查询武清至天津车票，发现余票成功扣减。在“我的车票”里也新增  
了该车票，数据库相应的余票信息也进行了更新。




	<div><div>欢迎乘坐京津城际铁路列车</div><div>购票 车次查询 车站查询 我的车票</div><div>发站: 武清 到站: 天津 查询</div><table><tr><th>车次</th><th>发站</th><th>出发时间</th><th>到站</th><th>到达时间</th><th>历时</th><th>二等座</th><th>一等座</th><th>商务座</th></tr><tr><td>C2575</td><td>武清</td><td>13:23</td><td>天津</td><td>13:37</td><td>14分</td><td>5</td><td>4</td><td>1</td></tr><tr><td>C2025</td><td>武清</td><td>19:20</td><td>天津</td><td>19:34</td><td>14分</td><td>5</td><td>4</td><td>2</td></tr></table><div>购票</div></div> <div><div>欢迎乘坐京津城际铁路列车</div><div>购票 车次查询 车站查询 我的车票</div><div>以下显示乘客 蒋杓言 的车票信息: 查询</div><table><tr><th>订单号</th><th>车次</th><th>发站</th><th>出发时间</th><th>到站</th><th>到达时间</th><th>历时</th><th>席别</th><th>席位号</th></tr><tr><td>BRU32PWL</td><td>C2575</td><td>武清</td><td>13:23</td><td>天津</td><td>13:37</td><td>14分</td><td>商务座</td><td>01车01F号</td></tr></table><div>退票</div></div> <table><tr><th>train_id</th><th>seat_id</th><th>start_station</th><th>end_station</th><th>ticket_status</th></tr><tr><td>C2575</td><td>01车01F号</td><td>军粮城北</td><td>滨海</td><td>available</td></tr><tr><td>C2575</td><td>01车01F号</td><td>北京南</td><td>军粮城北</td><td>sold</td></tr><tr><td>C2575</td><td>01车01F号</td><td>北京南</td><td>天津</td><td>sold</td></tr><tr><td>C2575</td><td>01车01F号</td><td>北京南</td><td>武清</td><td>available</td></tr><tr><td>C2575</td><td>01车01F号</td><td>北京南</td><td>滨海</td><td>sold</td></tr><tr><td>C2575</td><td>01车01F号</td><td>天津</td><td>军粮城北</td><td>available</td></tr><tr><td>C2575</td><td>01车01F号</td><td>天津</td><td>滨海</td><td>available</td></tr><tr><td>C2575</td><td>01车01F号</td><td>武清</td><td>军粮城北</td><td>sold</td></tr><tr><td>C2575</td><td>01车01F号</td><td>武清</td><td>天津</td><td>sold</td></tr><tr><td>C2575</td><td>01车01F号</td><td>武清</td><td>滨海</td><td>sold</td></tr></table>	车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座	C2575	武清	13:23	天津	13:37	14分	5	4	1	C2025	武清	19:20	天津	19:34	14分	5	4	2	订单号	车次	发站	出发时间	到站	到达时间	历时	席别	席位号	BRU32PWL	C2575	武清	13:23	天津	13:37	14分	商务座	01车01F号	train_id	seat_id	start_station	end_station	ticket_status	C2575	01车01F号	军粮城北	滨海	available	C2575	01车01F号	北京南	军粮城北	sold	C2575	01车01F号	北京南	天津	sold	C2575	01车01F号	北京南	武清	available	C2575	01车01F号	北京南	滨海	sold	C2575	01车01F号	天津	军粮城北	available	C2575	01车01F号	天津	滨海	available	C2575	01车01F号	武清	军粮城北	sold	C2575	01车01F号	武清	天津	sold	C2575	01车01F号	武清	滨海	sold
车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座																																																																																													
C2575	武清	13:23	天津	13:37	14分	5	4	1																																																																																													
C2025	武清	19:20	天津	19:34	14分	5	4	2																																																																																													
订单号	车次	发站	出发时间	到站	到达时间	历时	席别	席位号																																																																																													
BRU32PWL	C2575	武清	13:23	天津	13:37	14分	商务座	01车01F号																																																																																													
train_id	seat_id	start_station	end_station	ticket_status																																																																																																	
C2575	01车01F号	军粮城北	滨海	available																																																																																																	
C2575	01车01F号	北京南	军粮城北	sold																																																																																																	
C2575	01车01F号	北京南	天津	sold																																																																																																	
C2575	01车01F号	北京南	武清	available																																																																																																	
C2575	01车01F号	北京南	滨海	sold																																																																																																	
C2575	01车01F号	天津	军粮城北	available																																																																																																	
C2575	01车01F号	天津	滨海	available																																																																																																	
C2575	01车01F号	武清	军粮城北	sold																																																																																																	
C2575	01车01F号	武清	天津	sold																																																																																																	
C2575	01车01F号	武清	滨海	sold																																																																																																	
程序演示 (4分)	<p>说明：违背触发器要求，不能够执行插入操作，系统报错。</p> <p>乘客蒋杓言再次尝试购买武清至天津 C2575 次商务座，发现购票失败。</p> <div><div>错误</div><div>购票失败: (1644, 'Error: You are trying to book a duplicate ticket!')</div><div>确定</div></div>																																																																																																				
备注	以上是一个完整的事务操作，不可分割。任何一个环节失败则整个事务进行回滚；否则提交。																																																																																																				

## 6. 存储过程控制下的更新操作（18 分）

说明	（1 分）简要说明该操作所要完成的功能； （1 分）简要说明该存储过程所要完成的功能； （2 分）说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述） （1 分）表连接涉及字段描述（描述方式为“表 1.属性=表 2.属性”） （2 分）该操作会修改字段（以“表名.字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等； （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （5 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述 （1 分）	此功能较为简单，就是输入用户名、原密码和新密码，来修改账号的密码。并且要求新密码不能和原密码一致。	
存储过程 功能描述 （1 分）	传入新密码和用户名，更新表 passenger 的 password。	
涉及的关系表（2 分）	passenger,ticket	
表连接涉及 及字段 （1 分）	passenger.id_card=ticket.passenger_id	
更改字段 （2 分）	字段	规则
	password	要求新密码不能和原密码一致即可。
更新代码 （3 分）	1. 更新操作  <pre> # 插入用户信息到数据库 try:     connection = connect_db()     cursor = connection.cursor()      # 执行 SQL 查询，验证用户名和密码     query = "SELECT * FROM passenger WHERE user_name = %s AND password = %s"     cursor.execute(query, (username, previousPassword))      # 获取查询结果     result = cursor.fetchone()     if result: # 如果找到匹配的记录         # 插入数据的 SQL 语句         sql = "CALL updatePassword(%s, %s)"         cursor.execute(sql, (newPassword, username))         connection.commit()         cursor.close()         connection.close()          # 修改密码成功         messagebox.showinfo("提示", "修改密码成功！")         new_window1.destroy() # 关闭修改密码界面     else:         messagebox.showerror("登录失败", "用户名或密码错误")  except Exception as e:     messagebox.showerror("错误", f"修改失败: {e}") </pre>	

	<div>2. 触发器</div> <div><table><tr><th>名称</th><th>触发</th><th>插入</th><th>更新</th><th>删除</th></tr><tr><td>new_differ_old</td><td>BEFORE</td><td><input type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td></tr></table><div>定义<pre>1 BEGIN 2     IF NEW.`password` = OLD.`password` THEN 3         SIGNAL SQLSTATE '45000' 4         SET MESSAGE_TEXT = 'Error: New password must differ from the original one!'; 5     END IF; 6 END</pre></div></div>	名称	触发	插入	更新	删除	new_differ_old	BEFORE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																			
名称	触发	插入	更新	删除																																										
new_differ_old	BEFORE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																										
创建存储过程源码 (3 分)	<pre>1 CREATE DEFINER='root'@'localhost' PROCEDURE `updatePassword`(IN newPwd VARCHAR(255),username VARCHAR(255)) 2 BEGIN 3     UPDATE passenger SET password = newPwd WHERE user_name = username; 4 END</pre>																																													
存储过程执行源码 (1 分)	<pre># 获取查询结果 result = cursor.fetchone() if result: # 如果找到匹配的记录     # 插入数据的 SQL 语句     sql = "CALL updatePassword(%, %s)"     cursor.execute(sql, (newPassword, username))     connection.commit()     cursor.close()     connection.close()  # 修改密码成功 messagebox.showinfo("提示", "修改密码成功!") new_window1.destroy() # 关闭修改密码界面 else:     messagebox.showerror("登录失败", "用户名或密码错误")</pre>																																													
程序演示 (2 分)	<div>说明：不违背存储过程，能够执行更新操作 查看表 passenger，乘客蒋纳言的原密码是 cr353888，想要将密码改为 123456，尝试修改发现修改成功。</div> <table><tr><th>user_name</th><th>password</th><th>passenger_name</th><th>id_card</th><th>phone</th></tr><tr><td>a</td><td>0</td><td>测试账号</td><td>0</td><td>0</td></tr><tr><td>BigBouBo</td><td>tiananmen568</td><td>郭正平</td><td>120110200405059999</td><td>13577777777</td></tr><tr><td>bozi123</td><td>wyb999zlx</td><td>王怡博</td><td>411326200510103333</td><td>13566666666</td></tr><tr><td>ClarenceWise</td><td>479479xy</td><td>熊耀</td><td>530629200512126666</td><td>18888888888</td></tr><tr><td>jry050608</td><td>cr353888</td><td>蒋纳言</td><td>510703200501010101</td><td>13550837630</td></tr><tr><td>loong</td><td>wyl123</td><td>温延龙</td><td>XXXXXXXXXXXXXXXXXX</td><td>15555555555</td></tr><tr><td>zhangsb</td><td>123</td><td>张翔鸣</td><td>220101199911112222</td><td>13900000000</td></tr><tr><td>zzzzzzhangsan</td><td>zhang123abc</td><td>张三</td><td>51011120000101111X</td><td>13333333333</td></tr></table> <div><div><div>修改密码</div><div><div>用户名:</div><div>jry050608</div></div><div><div>原密码:</div><div>cr353888</div></div><div><div>新密码:</div><div>*****</div></div><div><div>请再次确认新密码:</div><div>*****</div></div><div><div>修改密码</div><div>关闭</div></div></div><div><div>提示</div><div><div>修改密码成功!</div><div>确定</div></div></div></div> <div>此时表 passenger 里的信息也成功更新。</div>	user_name	password	passenger_name	id_card	phone	a	0	测试账号	0	0	BigBouBo	tiananmen568	郭正平	120110200405059999	13577777777	bozi123	wyb999zlx	王怡博	411326200510103333	13566666666	ClarenceWise	479479xy	熊耀	530629200512126666	18888888888	jry050608	cr353888	蒋纳言	510703200501010101	13550837630	loong	wyl123	温延龙	XXXXXXXXXXXXXXXXXX	15555555555	zhangsb	123	张翔鸣	220101199911112222	13900000000	zzzzzzhangsan	zhang123abc	张三	51011120000101111X	13333333333
user_name	password	passenger_name	id_card	phone																																										
a	0	测试账号	0	0																																										
BigBouBo	tiananmen568	郭正平	120110200405059999	13577777777																																										
bozi123	wyb999zlx	王怡博	411326200510103333	13566666666																																										
ClarenceWise	479479xy	熊耀	530629200512126666	18888888888																																										
jry050608	cr353888	蒋纳言	510703200501010101	13550837630																																										
loong	wyl123	温延龙	XXXXXXXXXXXXXXXXXX	15555555555																																										
zhangsb	123	张翔鸣	220101199911112222	13900000000																																										
zzzzzzhangsan	zhang123abc	张三	51011120000101111X	13333333333																																										



	<table><tr><th>user_name</th><th>password</th><th>passenger_name</th><th>id_card</th><th>phone</th></tr><tr><td>a</td><td>0</td><td>测试账号</td><td>0</td><td>0</td></tr><tr><td>BigBouBo</td><td>tiananmen568</td><td>郭正平</td><td>120110200405059999</td><td>13577777777</td></tr><tr><td>bozi123</td><td>wyb999zqx</td><td>王怡博</td><td>411326200510103333</td><td>13566666666</td></tr><tr><td>ClarenceWise</td><td>479479xy</td><td>熊耀</td><td>530629200512126666</td><td>18888888888</td></tr><tr><td>jry050608</td><td>123456</td><td>蒋纳言</td><td>510703200501010101</td><td>13550837630</td></tr><tr><td>loong</td><td>wyl123</td><td>温延龙</td><td>XXXXXXXXXXXXXXXXXX</td><td>15555555555</td></tr><tr><td>zhangsb</td><td>123</td><td>张栩鸣</td><td>220101199911112222</td><td>13900000000</td></tr><tr><td>zzzzzzhangsan</td><td>zhang123abc</td><td>张三</td><td>51011120000101111X</td><td>13333333333</td></tr></table>	user_name	password	passenger_name	id_card	phone	a	0	测试账号	0	0	BigBouBo	tiananmen568	郭正平	120110200405059999	13577777777	bozi123	wyb999zqx	王怡博	411326200510103333	13566666666	ClarenceWise	479479xy	熊耀	530629200512126666	18888888888	jry050608	123456	蒋纳言	510703200501010101	13550837630	loong	wyl123	温延龙	XXXXXXXXXXXXXXXXXX	15555555555	zhangsb	123	张栩鸣	220101199911112222	13900000000	zzzzzzhangsan	zhang123abc	张三	51011120000101111X	13333333333
user_name	password	passenger_name	id_card	phone																																										
a	0	测试账号	0	0																																										
BigBouBo	tiananmen568	郭正平	120110200405059999	13577777777																																										
bozi123	wyb999zqx	王怡博	411326200510103333	13566666666																																										
ClarenceWise	479479xy	熊耀	530629200512126666	18888888888																																										
jry050608	123456	蒋纳言	510703200501010101	13550837630																																										
loong	wyl123	温延龙	XXXXXXXXXXXXXXXXXX	15555555555																																										
zhangsb	123	张栩鸣	220101199911112222	13900000000																																										
zzzzzzhangsan	zhang123abc	张三	51011120000101111X	13333333333																																										
程序演示 (2 分)	<p>说明：违背存储过程，系统报错</p> <p>如果再次尝试将密码改为 123456，发现修改失败，提示新密码必须与原密码不同。</p> <div><div> 错误</div><div></div><div> 修改失败: (1644, 'Error: New password must differ from the original one!')</div><div>确定</div></div>																																													
备注	修改密码首先需要确保原密码输入正确，然后保证新密码与原密码不同才可修改成功。																																													

## 7. 含有视图的查询操作（15 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明建立的该视图的功能；</p> <p>（2 分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）</p> <p>（1 分）简要说明表连接涉及的字段（以“表 1.属性=表 2.属性”）</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
操作功能描述（1 分）	此部分的功能就是根据用户输入的发站和到站查询车次信息和余票信息。
视图功能描述（1 分）	筛选出所有车次能够出票的合理区间，以及该区间的出发时间、到达时间和历时。
涉及的关系表（2 分）	inventory,seat,stopovers
表连接字段（1 分）	<p>inventory.train_id=seat.train_id=stopovers.train_id</p> <p>inventory.seat_id=seat.seat_id</p>
创建视图代码（3 分）	<pre> CREATE OR REPLACE ALGORITHM = UNDEFINED DEFINER = `root`@`localhost` SQL SECURITY DEFINER VIEW `ticket_booking`.`train_schedule_view` AS SELECT `s1`.`train_id` AS `train_id`, `s1`.`station` AS `start_station`, substring_index( `s1`.`departure_time`, ':', 2 ) AS `start_time`, `s2`.`station` AS `end_station`, substring_index( `s2`.`arrival_time`, ':', 2 ) AS `end_time`, concat( IF (( floor(( time_to_sec( timediff( `s2`.`arrival_time`, `s1`.`departure_time` )) / 3600 )) &gt; 0 ), concat( floor(( time_to_sec( timediff( `s2`.`arrival_time`, `s1`.`departure_time` )) / 3600 ), '小时' ), ), IF ((( floor(( time_to_sec( timediff( `s2`.`arrival_time`, `s1`.`departure_time` )) / 60 )) % 60 ) &gt; 0 ), concat( floor(( time_to_sec( timediff( `s2`.`arrival_time`, `s1`.`departure_time` )) / 60 )) % 60 ), '分' ), )) AS `travel_time` FROM ( `stopovers` `s1` JOIN `stopovers` `s2` ON (( `s1`.`train_id` = `s2`.`train_id` ))) WHERE ( `s1`.`sequence` &lt; `s2`.`sequence`); </pre>



查询代码  
(3 分)

```
try:
    conn = connect_db()
    cursor = conn.cursor()

    # SQL 查询车票
    query = """
        SELECT
            train_id,start_station,start_time,end_station,end_time,travel_time,
            (SELECT num_of_seats
             FROM (SELECT train_id,COUNT(seat_id) AS num_of_seats FROM inventory NATURAL JOIN seat
                  WHERE start_station = %s
                     AND end_station = %s
                     AND ticket_status = 'available'
                     AND seat_type = '二等座'
                  GROUP BY train_id
                 ) AS count_seat
             WHERE count_seat.train_id = train_schedule_view.train_id
            ) AS ZE,
            (SELECT num_of_seats
             FROM (SELECT train_id,COUNT(seat_id) AS num_of_seats FROM inventory NATURAL JOIN seat
                  WHERE start_station = %s
                     AND end_station = %s
                     AND ticket_status = 'available'
                     AND seat_type = '一等座'
                  GROUP BY train_id
                 ) AS count_seat
             WHERE count_seat.train_id = train_schedule_view.train_id
            ) AS ZY,
            (SELECT num_of_seats
             FROM (SELECT train_id,COUNT(seat_id) AS num_of_seats FROM inventory NATURAL JOIN seat
                  WHERE start_station = %s
                     AND end_station = %s
                     AND ticket_status = 'available'
                     AND seat_type = '商务座'
                  GROUP BY train_id
                 ) AS count_seat
             WHERE count_seat.train_id = train_schedule_view.train_id
            ) AS SW
        FROM
            train_schedule_view
        WHERE
            start_station = %s
            AND end_station = %s
        ORDER BY
            start_time ASC;

        """

    cursor.execute(query, (dep_station, arr_station, dep_station, arr_station, dep_station, arr_station, dep_station, arr_station))
    results = cursor.fetchall()

    # 清空表格
    for row in tree3.get_children():
        tree3.delete(row)

    # 插入查询结果
    for row in results:
        train_no, departure_station, d_time, arrival_station, a_time, travel_time, second_class, first_class, business_class = row
        checkbox_vars2[train_no] = tk.BooleanVar(value=False)

        tree3.insert("", 'end', iid=train_no, values=(
            train_no, departure_station, d_time,
            arrival_station, a_time, travel_time, second_class, first_class, business_class,
            "\n" if checkbox_vars2[train_no].get() else "" # 初始为空
        ))

    cursor.close()
    conn.close()
except Exception as e:
    messagebox.showerror("错误", f"数据库查询失败: {e}")
```

输入发站和到站，点击“查询”，即可查询到车票信息。二等座、一等座、商务座下方的数字代表的是余票数量。

欢迎乘坐京津城际铁路列车

购票 车次查询 车站查询 我的车票

发站:  到站:

车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座
C2551	北京南	06:00	天津	06:30	30分	5	4	2
C2003	北京南	06:51	天津	07:28	37分	5	4	2
C2575	北京南	13:00	天津	13:37	37分	5	4	1
C2025	北京南	18:47	天津	19:34	47分	5	4	2
C2597	北京南	20:19	天津	20:49	30分	5	4	2

购票

欢迎乘坐京津城际铁路列车

购票 车次查询 车站查询 我的车票

发站:  到站:

车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座
C2570	军粮城北	12:18	天津	12:31	13分	5	4	2

购票

欢迎乘坐京津城际铁路列车

购票 车次查询 车站查询 我的车票

发站:  到站:

车次	发站	出发时间	到站	到达时间	历时	二等座	一等座	商务座
C2562	滨海	09:17	天津	09:39	22分	5	4	2
C2570	滨海	11:55	天津	12:31	36分	5	4	2
C2596	滨海	20:31	天津	20:53	22分	5	4	2

购票

视图筛选出了所有车次能够出票的合理区间，以及该区间的出发时间、到达时间和历时，在视图的基础上再进行车票查询就方便了许多。