


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	2313828	姓名	苏雨辰	专业	计算机科学与技术																												
项目名称	飞机客票售票信息数据库系统																																
必备环境	Java、MySQL																																
系统主要功能简介（4 分）	1. 实现用户注册、登录与注销 2. 实现机票信息查询、预定和退票操作 3. 在用户界面查询、取消订单，注册常旅客																																
系统主要页面截图（6 分）	<div><div><div><div>Flying Easily</div><div>Flying Easily Flight Booking System</div><div></div><div>Feel free to book tickets!</div><div><div>Sign in</div><div>Log in</div></div></div></div><div><div>User Information</div><div><div>Back</div><div>UID: 100000001</div><div>Username: L1011 <div>Change</div></div><div>Real Name: SU YUCHEN <div>Change</div></div><div>ID Number: 161651561 <div>Change</div></div><div>Telephone: 225562200 <div>Change</div></div><div>Email: 2313828@nankai.edu.cn <div>Change</div></div><div>Password: ***** <div>Change</div></div><div>FLYING ALLIANCE MEMBER</div><div>Urange: Lv1</div><div>Discount: 993</div></div></div></div> <div><div>My Bookings</div><table><thead><tr><th>Date</th><th>Flight No</th><th>DEP-ARR</th><th>Price</th><th>Seat No</th><th>Reservation No</th><th>Action</th></tr></thead><tbody><tr><td>2025-05-30</td><td>KE806</td><td>TSN-ICN</td><td>1141.95</td><td>1</td><td>2FBCBEE1660939F5C</td><td><div>Refund</div></td></tr><tr><td>2025-05-29</td><td>MU2278</td><td>TSN-XYY</td><td>1141.95</td><td>1</td><td>24452DDACA8B64E2</td><td><div>Refund</div></td></tr><tr><td>2025-05-29</td><td>CA2878</td><td>TSN-XYY</td><td>1141.95</td><td>1</td><td>35D75A246946F736</td><td><div>Refund</div></td></tr></tbody></table><div><div>Cancel Booking</div><div><div>?</div><div>Are you sure to cancel the following booking?</div><div>Date: 2025-05-30</div><div>Flight No: KE806</div><div>Route: TSN-ICN</div><div>Seat No: 1</div><div>Reservation No: 2FBCBEE1660939F5C</div><div>Price: 1141.95</div><div><div>确定</div><div>取消</div></div></div></div><div><div>Flight Search</div><div><div>From: TSN To: XYY Month: 5 Day: 29</div><div><div>Search</div><div>Back</div></div><div><div>Date: 2025-05-29 Flight: CA2878 From: TSN To: XYY Time: 07:15:00-09:20:00</div><div><div>Book</div></div><div><div>Date: 2025-05-29 Flight: MF8239 From: TSN To: XYY Time: 11:45:00-14:00:00</div><div><div>Book</div></div><div><div>Date: 2025-05-29 Flight: MU2278 From: TSN To: XYY Time: 18:15:00-20:15:00</div><div><div>Book</div></div><div><div>Date: 2025-05-29 Flight: BK2884 From: TSN To: XYY Time: 22:15:00-24:00:00</div><div><div>Book</div></div></div></div><div><div>Confirm Booking</div><div><div>?</div><div>Flight No: MU2278</div><div>Date: 20250529</div><div>From: TSN</div><div>To: XYY</div><div>Departure-Arrival: 18:15:00 - 20:15:00</div><div>RegNo: BE390 Type: A319-100 Age: 9.2</div><div>Seat No: 2 Price: 1141.95</div><div><div>确定</div><div>取消</div></div></div></div><div><div>Back</div></div></div></div></div></div></div>					Date	Flight No	DEP-ARR	Price	Seat No	Reservation No	Action	2025-05-30	KE806	TSN-ICN	1141.95	1	2FBCBEE1660939F5C	<div>Refund</div>	2025-05-29	MU2278	TSN-XYY	1141.95	1	24452DDACA8B64E2	<div>Refund</div>	2025-05-29	CA2878	TSN-XYY	1141.95	1	35D75A246946F736	<div>Refund</div>
Date	Flight No	DEP-ARR	Price	Seat No	Reservation No	Action																											
2025-05-30	KE806	TSN-ICN	1141.95	1	2FBCBEE1660939F5C	<div>Refund</div>																											
2025-05-29	MU2278	TSN-XYY	1141.95	1	24452DDACA8B64E2	<div>Refund</div>																											
2025-05-29	CA2878	TSN-XYY	1141.95	1	35D75A246946F736	<div>Refund</div>																											

My Bookings

Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action
2025-05-30	KE806	TSN-ICN	1141.95	1	2FBC6EE1660939P5C	<div>Refund</div>
2025-05-29	MU2278	TSN-XYY	1141.95	1	24452DDACAB8E4E2	<div>Refund</div>
2025-05-29	CA2878	TSN-XYY	1141.95	1	35D75A246946F736	<div>Refund</div>

Cancel Booking

Are you sure to cancel the following booking?

Date: 2025-05-30

Flight No: KE806

Route: TSN-ICN

Seat No: 1

Reservation No: 2FBC6EE1660939P5C

Price: 1141.95

是

否

Flight Search

From: TSN To: XYY Month: 5 Day: 29

Search

Back

Date: 2025-05-29 Flight: CA2878 From: TSN To: XYY Time: 07:15:00-09:20:00

Book

Date: 2025-05-29 Flight: MF8239 From: TSN To: XYY Time: 11:45:00-14:00:00

Book

Date: 2025-05-29 Flight: MU2278 From: TSN To: XYY Time: 18:15:00-20:15:00

Book

Date: 2025-05-29 Flight: BK2884 From: TSN To: XYY Time: 22:15:00-24:00:00

Book

Confirm Booking

Flight No: MU2278

Date: 20250529

From: TSN

To: XYY

Departure-Arrival: 18:15:00 - 20:15:00

RegNo: B8390 Type: A319-100 Age: 9.2

Seat No: 2 Price: 1141.95

确定

取消

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	1. MySQL 8.4.5			
	高级 语言	1. Java 17.0.12 LST			
连接串 分析 (6 分)		序号	名称	功能说明	取值
		1	host	数据库服务器地址	localhost
		2	port	数据库服务端口号	3306
		3	database	要连接的数据库名称	flightrightinfo
		4	Server Timezone	设置数据库服务器时区	Asia/Shanghai
		5	useUnicode	启用 Unicode 字符集支持	true
		6	Character Encoding	指定字符编码格式	utf8
		7	useSSL	启用 SSL 加密连接	true
		8	username	数据库登录用户名	root
		9	password	数据库登录密码	*****
连接串代码 (截屏) (2 分)		<pre>package sql; import java.sql.*; public class DBUtil { 6个用法 新* public static Connection getConnection() throws SQLException { 6个用法 新* String url = "jdbc:mysql://localhost:3306/flightrightinfo?serverTimezone=Asia/Shanghai&useUr String username = "root";</pre>			
备注		由于 MySQL 的 TIMESTAMP 字段会 自动转换为 UTC 时间存储，并在查询时转回当前会话时区。因此需要在连接时指定时区防止时区混乱报错。			

3. 数据库设计（14 分）

说明	(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； (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。				
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	airport	(IATAcode)	无	无
	2	aircraft	(regno)	无	无

	3	user	(uid)	无	无																																																																																																																								
	4	flight	(flightno, fldate)	(regno, Dep_IATAcodes, Arv_IATAcodes)	aircraft(regno, type, seatcapacity, age) airport (IATAcodes, ICAOcode, Position) airport (IATAcodes, ICAOcode, Position)																																																																																																																								
	5	booking	(ssvno)	(flightno, uid)	flight(flightno, fldate, regno, Dep_IATAcodes, Arv_IATAcodes, departuretime, arrivaltime, distance) user(uid, uname, realname, telno, email, IDno, password)																																																																																																																								
	6	Frequentpassenger (user 的子类)	(uid)	无	无																																																																																																																								
关系图	<p>airport (IATAcodes, ICAOcode, Position)</p> <p>aircraft(regno, type, seatcapacity, age)</p> <p>(4) user (uid, uname, realname, telno, email, IDno, password)</p> <p>frequentpassenger (uid, Urange, personaldiscount, credits)</p> <p>flight (flightno, fldate, regno, Dep_IATAcodes, Arv_IATAcodes, departuretime, arrivaltime, distance)</p> <p>booking (rsvno, flightno, uid, price, seatno, fldate)</p> <pre>erDiagram AIRCRAFT --o{ FLIGHT : operate AIRCRAFT --o{ FLIGHT : departure AIRCRAFT --o{ FLIGHT : arrival USER --o{ BOOKING : refer USER --o{ BOOKING : reserve USER --o{ FREQUENTPASSANGER : isa</pre> <p>Aircraft</p> <table><tr><td>regno</td><td><pi></td><td>Variable characters (8)</td><td><M></td></tr><tr><td>type</td><td></td><td>Variable characters (30)</td><td><M></td></tr><tr><td>seatcapacity</td><td></td><td>Integer</td><td><M></td></tr><tr><td>age</td><td></td><td>Float (8)</td><td><M></td></tr><tr><td>Identifier_1</td><td><pi></td><td></td><td></td></tr></table> <p>Flight</p> <table><tr><td>flightno</td><td><pi></td><td>Variable characters (6)</td><td><M></td></tr><tr><td>departuretime</td><td></td><td>Time</td><td></td></tr><tr><td>arrivaltime</td><td></td><td>Time</td><td></td></tr><tr><td>Identifier_1</td><td><pi></td><td></td><td></td></tr></table> <p>Airport</p> <table><tr><td>IATAcodes</td><td><pi></td><td>Characters (3)</td><td><M></td></tr><tr><td>ICAOcode</td><td></td><td>Characters (4)</td><td><M></td></tr><tr><td>Position</td><td></td><td>Characters (256)</td><td></td></tr><tr><td>Identifier_1</td><td><pi></td><td></td><td></td></tr></table> <p>User</p> <table><tr><td>uid</td><td><pi></td><td>Variable characters (15)</td><td><M></td></tr><tr><td>uname</td><td></td><td>Variable characters (20)</td><td><M></td></tr><tr><td>realname</td><td></td><td>Variable characters (256)</td><td><M></td></tr><tr><td>telno</td><td></td><td>Variable characters (20)</td><td></td></tr><tr><td>email</td><td></td><td>Variable characters (50)</td><td></td></tr><tr><td>IDno</td><td></td><td>Variable characters (20)</td><td></td></tr><tr><td>passportno</td><td></td><td>Variable characters (20)</td><td></td></tr><tr><td>Identifier_1</td><td><pi></td><td></td><td></td></tr></table> <p>Booking</p> <table><tr><td>rsvno</td><td><pi></td><td>Characters (30)</td><td><M></td></tr><tr><td>price</td><td></td><td>Float</td><td><M></td></tr><tr><td>seatno</td><td></td><td>Characters (10)</td><td><M></td></tr><tr><td>class</td><td></td><td>Characters (10)</td><td></td></tr><tr><td>date</td><td></td><td>Date</td><td></td></tr><tr><td>Identifier_1</td><td><pi></td><td></td><td></td></tr></table> <p>FrequentPassenger</p> <table><tr><td>range</td><td></td><td>Characters (20)</td><td><M></td></tr><tr><td>personaldiscount</td><td></td><td>Characters (256)</td><td><M></td></tr><tr><td>credits</td><td></td><td>Integer</td><td></td></tr></table>					regno	<pi>	Variable characters (8)	<M>	type		Variable characters (30)	<M>	seatcapacity		Integer	<M>	age		Float (8)	<M>	Identifier_1	<pi>			flightno	<pi>	Variable characters (6)	<M>	departuretime		Time		arrivaltime		Time		Identifier_1	<pi>			IATAcodes	<pi>	Characters (3)	<M>	ICAOcode		Characters (4)	<M>	Position		Characters (256)		Identifier_1	<pi>			uid	<pi>	Variable characters (15)	<M>	uname		Variable characters (20)	<M>	realname		Variable characters (256)	<M>	telno		Variable characters (20)		email		Variable characters (50)		IDno		Variable characters (20)		passportno		Variable characters (20)		Identifier_1	<pi>			rsvno	<pi>	Characters (30)	<M>	price		Float	<M>	seatno		Characters (10)	<M>	class		Characters (10)		date		Date		Identifier_1	<pi>			range		Characters (20)	<M>	personaldiscount		Characters (256)	<M>	credits		Integer	
regno	<pi>	Variable characters (8)	<M>																																																																																																																										
type		Variable characters (30)	<M>																																																																																																																										
seatcapacity		Integer	<M>																																																																																																																										
age		Float (8)	<M>																																																																																																																										
Identifier_1	<pi>																																																																																																																												
flightno	<pi>	Variable characters (6)	<M>																																																																																																																										
departuretime		Time																																																																																																																											
arrivaltime		Time																																																																																																																											
Identifier_1	<pi>																																																																																																																												
IATAcodes	<pi>	Characters (3)	<M>																																																																																																																										
ICAOcode		Characters (4)	<M>																																																																																																																										
Position		Characters (256)																																																																																																																											
Identifier_1	<pi>																																																																																																																												
uid	<pi>	Variable characters (15)	<M>																																																																																																																										
uname		Variable characters (20)	<M>																																																																																																																										
realname		Variable characters (256)	<M>																																																																																																																										
telno		Variable characters (20)																																																																																																																											
email		Variable characters (50)																																																																																																																											
IDno		Variable characters (20)																																																																																																																											
passportno		Variable characters (20)																																																																																																																											
Identifier_1	<pi>																																																																																																																												
rsvno	<pi>	Characters (30)	<M>																																																																																																																										
price		Float	<M>																																																																																																																										
seatno		Characters (10)	<M>																																																																																																																										
class		Characters (10)																																																																																																																											
date		Date																																																																																																																											
Identifier_1	<pi>																																																																																																																												
range		Characters (20)	<M>																																																																																																																										
personaldiscount		Characters (256)	<M>																																																																																																																										
credits		Integer																																																																																																																											
备注	<p>在实际工程中，会有部分表的属性改变，ER图可能会与实际不一致，以文字的 schema 为准。</p> <p>为进行存储过程控制下的更新操作，新增了表 credit_logs 作为日志表，进行备份工作。</p> <p>credit_logs (log_id, user_id, old_credits, new_credits, change_date, change_reason)</p>																																																																																																																												

4. 含有事务应用的删除操作（13 分）

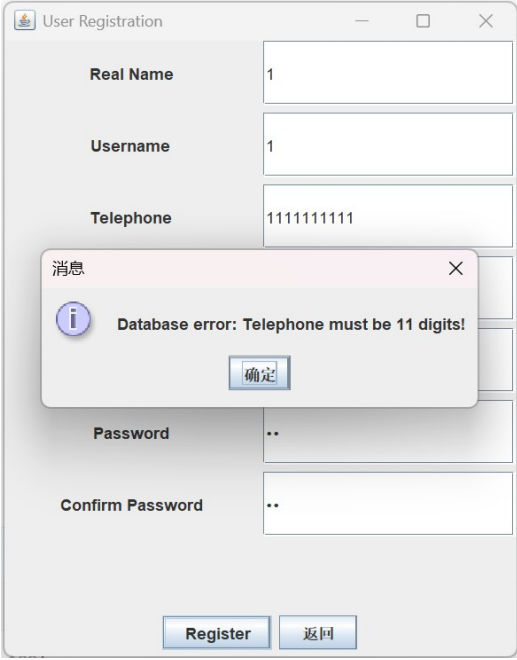
说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（2 分）该操作会涉及的表（必须含有两张或两张以上的关系表，同时以“表名”的形式给出）</p> <p>（1 分）表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>（1 分）删除条件涉及的字段描述（以“表名. 属性=? ”形式给出）</p> <p>（4 分）实现该操作的关键代码（高级语言、SQL），截图即可；（其中如果删除语句中不包含任何形式的事务应用将扣除 3 分）</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>	
功能描述（1 分）	对于现有的订单执行退票操作。该事务包括：取消订单（从 booking 删除该行）+里程回退（credits-该段里程）	
涉及的表（2 分）	booking、frequentpassenger	
表连接涉及字段（1 分）	uid 进行关联	
删除条件字段描述（1 分）	字段	规则
	p_rsvno	WHERE rsvno = p_rsvno 精确匹配订单编号（主键或唯一键）
	p_uid	WHERE uid = p_uid 精确匹配用户 ID（外键约束）
代码（4 分）	<p>事务 MySQL 定义语句</p> <pre> 1 DELIMITER // 2 CREATE PROCEDURE cancel_booking(3 IN p_rsvno VARCHAR(20), 4 IN p_uid BIGINT, 5 IN p_distance INT 6) 7 BEGIN 8 DECLARE EXIT HANDLER FOR SQLEXCEPTION 9 BEGIN 10 ROLLBACK; 11 END; 12 START TRANSACTION; 13 -- 1. 删除对应订单 14 DELETE FROM booking WHERE rsvno = p_rsvno; 15 -- 2. 减少 frequentpassenger 中的积分 16 UPDATE frequentpassenger 17 SET credits = credits - p_distance 18 WHERE uid = p_uid; 19 COMMIT; 20 END // 21 DELIMITER ; 22 </pre> <p>在程序中直接使用 CALL 调用事务 cancel_booking</p>	

	<pre>if (choice == JOptionPane.YES_OPTION) { try (Connection conn2 = DBUtil.getConnection()) { // 调用存储过程 cancel_booking(rsvno, uid, distance) CallableStatement cs = conn2.prepareCall(sql: "{CALL cancel_booking(?, ?, ?)}"); cs.setString(parameterIndex: 1, rsvno); cs.setLong(parameterIndex: 2, uid); cs.setInt(parameterIndex: 3, distance); cs.execute(); cs.close(); } }</pre>																																																								
程序演示（4分）	<p>初始的行程和里程积分如下：</p> <div><div>FLYING ALLIANCE MEMBER</div><div>Urange: Lv1</div><div>Discount:</div><div>Credits: 1809</div></div> <div><div>My Bookings</div><table><tr><th>Date</th><th>Flight No</th><th>DEP-ARV</th><th>Price</th><th>Seat No</th><th>Reservation No</th><th>Action</th></tr><tr><td>2025-05-29</td><td>MF8239</td><td>TSN-XIY</td><td>1141.95</td><td>1</td><td>CD12E2B1F22A54CF</td><td>Refund</td></tr><tr><td>2025-05-29</td><td>KE806</td><td>TSN-ICN</td><td>938.40</td><td>1</td><td>E809644A82566080</td><td>Refund</td></tr></table></div> <p>执行 refund 退票操作，显示订单信息：</p> <div><div>My Bookings</div><table><tr><th>Date</th><th>Flight No</th><th>DEP-ARV</th><th>Price</th><th>Seat No</th><th>Reservation No</th><th>Action</th></tr><tr><td>2025-05-29</td><td>MF8239</td><td>TSN-XIY</td><td>1141.95</td><td>1</td><td>CD12E2B1F22A54CF</td><td>Refund</td></tr><tr><td>2025-05-29</td><td>KE806</td><td>TSN-ICN</td><td>938.40</td><td>1</td><td>E809644A82566080</td><td>Refund</td></tr></table><div><div>Cancel Booking</div><div><div>?</div><div>Are you sure to cancel the following booking?</div><div>Date: 2025-05-29</div><div>Flight No: MF8239</div><div>Route: TSN-XIY</div><div>Seat No: 1</div><div>Reservation No: CD12E2B1F22A54CF</div><div>Price: 1141.95</div><div>是(Y) 否(N)</div></div></div></div> <p>退票完成，行程信息与积分更新：</p> <div><div>My Bookings</div><table><tr><th>Date</th><th>Flight No</th><th>DEP-ARV</th><th>Price</th><th>Seat No</th><th>Reservation No</th><th>Action</th></tr><tr><td>2025-05-29</td><td>KE806</td><td>TSN-ICN</td><td>938.40</td><td>1</td><td>E809644A82566080</td><td>Refund</td></tr></table><div><div>FLYING ALLIANCE MEMBER</div><div>Urange: Lv1</div><div>Discount:</div><div>Credits: 816</div></div></div>	Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action	2025-05-29	MF8239	TSN-XIY	1141.95	1	CD12E2B1F22A54CF	Refund	2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund	Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action	2025-05-29	MF8239	TSN-XIY	1141.95	1	CD12E2B1F22A54CF	Refund	2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund	Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action	2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund
	Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action																																																		
	2025-05-29	MF8239	TSN-XIY	1141.95	1	CD12E2B1F22A54CF	Refund																																																		
	2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund																																																		
Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action																																																			
2025-05-29	MF8239	TSN-XIY	1141.95	1	CD12E2B1F22A54CF	Refund																																																			
2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund																																																			
Date	Flight No	DEP-ARV	Price	Seat No	Reservation No	Action																																																			
2025-05-29	KE806	TSN-ICN	938.40	1	E809644A82566080	Refund																																																			
备注	此部分 UI 已更新，最终情况以最后为准。（UI 不影响该部分功能的正常执行）。																																																								

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)	注册时在 mysql 语句层面验证输入的手机号、邮箱、IDno 的合法性。	
触发器描述 (2 分)	要求输入手机号必须为 11 位, 邮箱必须含有@, ID 身份证号必须为 18 位。(这里用手机号做示例)。	
涉及的表 (1 分)	user	
输入数据 (2 分)	字段	规则
	telno	通过 textfield 获取输入的电话号码, 将其直接作为参数整合进 INSERT 语句, 在 MYSQL 构建的触发器中, 若发现 telno 不是 11 位, 会触发异常, 返回报错信息, 弹出提示弹窗。
插入操作源码 (3 分)	<pre> ResultSet rs = stmt.executeQuery(sql: "SELECT MAX(uid) FROM user"); long newUid = 100000001L; if (rs.next() && rs.getLong(columnIndex: 1) > 0) { newUid = rs.getLong(columnIndex: 1) + 1; } PreparedStatement ps = conn.prepareStatement(sql: "INSERT INTO user(uid, realname, uname, telno, email, IDno, password) " + "VALUES (?, ?, ?, ?, ?, ?, ?)"); ps.setLong(parameterIndex: 1, newUid); ps.setString(parameterIndex: 2, fields[0].getText()); ps.setString(parameterIndex: 3, fields[1].getText()); ps.setLong(parameterIndex: 4, telno); ps.setString(parameterIndex: 5, email); ps.setLong(parameterIndex: 6, IDno); ps.setString(parameterIndex: 7, password1); String realname=fields[0].getText(); ps.executeUpdate(); JOptionPane.showMessageDialog(parentComponent: this, message: "Registration successful! UID: " + newUid); </pre> <p>插入操作的源代码。首先获取已经创建的最大 uid+1, 通过获取 textfield 输入的数据, 使用 setxxx 对对应位置的进行赋值, 使用 INSERT...VALUE()... 进行插入操作。</p>	

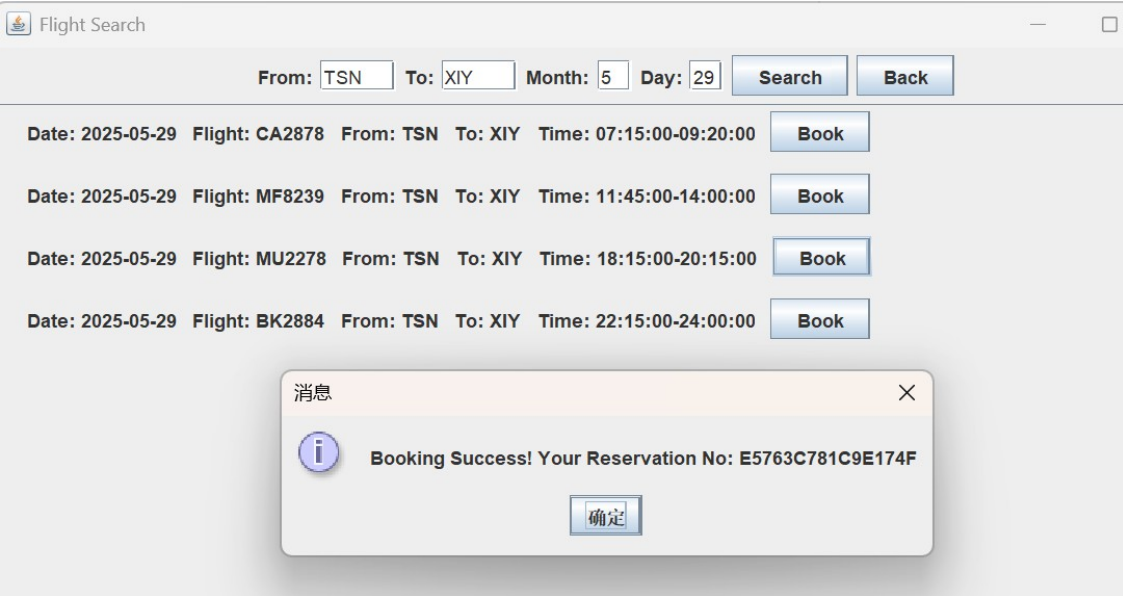
	<p>使用 try-catch 进行异常捕获，如果输入的 telno 位数不正确会引发数据库异常，我们通过对应的 message text 的信息，进行对应的异常处理。这里的操作是弹出消息框，同时不会执行提交写入操作。（其他的 email 和 IDno 的异常处理类似）。</p> <pre> } catch (NumberFormatException ex) { JOptionPane.showMessageDialog(parentComponent: this, message: "Invalid number format: " + ex.getMessage()); } catch (SQLException ex) { // 捕获数据库触发器的错误信息 String errorMsg = ex.getMessage(); if (errorMsg.contains("telno_length")) { JOptionPane.showMessageDialog(parentComponent: this, message: "Database error: Telephone must be 11 digits!"); } else if (errorMsg.contains("email_format")) { JOptionPane.showMessageDialog(parentComponent: this, message: "Database error: Invalid email format!"); } else if (errorMsg.contains("idno_length")) { JOptionPane.showMessageDialog(parentComponent: this, message: "Database error: ID number must be 18 digits!"); } else { JOptionPane.showMessageDialog(parentComponent: this, message: "Database error: " + errorMsg); } ex.printStackTrace(); } }</pre>
触发器源码 (3分)	<pre>BEGIN IF LENGTH(CAST(NEW.telno AS CHAR)) != 11 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'telno_length'; END IF; END</pre> <p>检查输入的 telno 长度，如果长度不是 11，数据库报错，message text 标记为 telno_length。</p>
程序演示 (4分)	<p>说明：不违背触发器能够执行插入操作。 输入的所有数据符合要求，提交成功。</p> 
程序演示	<p>说明：违背触发器要求，不能够执行插入操作，系统报错。 输入错误位数得到 telno，点解 Register 后，弹出消息框显示：位数不是 11 位。此操作</p>

<p>(4分)</p>	<p>不会写入数据库，只有当输入符合要求后才会提交数据库。</p> <div data-bbox="292 239 810 896"></div>
<p>备注</p>	<p>此部分仅针对 telno、email、IDno 进行数据库层面的触发器设置，realname，和 uname 不作检查。对密码是在 java 程序层面进行检查，只有当两次输入密码一致才能进入到数据库插入的操作。</p>

6. 存储过程控制下的更新操作（18 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明该存储过程所要完成的功能；</p> <p>（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）</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>（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；</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>（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。</p>	
功能描述 （1 分）	通过存储过程 update_credits 实现用户积分更新功能，在用户成功预订航班后，根据飞行距离增加用户积分	
存储过程功能描述 （1 分）	根据用户 ID 和飞行距离参数，如果用户是常旅客，更新相应用户的积分值。	
涉及的关系表 （2 分）	users、booking、flight	
表连接涉及字段 （1 分）	booking.uid = user.uid booking.flightno = flight.flightno booking.fldate = flight.fldate	
更改字段 （2 分）	字段	规则
	user.credits	存储过程 update_credits 控制更新 输入验证：距离 > 0（否则抛出异常）；用户必须存在（否则抛出异常） 事务保护：整个更新在事务中完成
更新代码 （3 分）	Booking 插入	price：基于飞行距离计算（距离*1.15）；seatno：自动查找最小可用座位号；事务保护：与积分更新在同一事务中
	该部分代码在预订记录插入之后，事务提交之前。构建了一个 CallableStatement，用来调用 update_credits。所有操作包括绑定用户 ID 和飞行距离参数、执行存储过程、异常处理均包含在其中。在完成调用后关闭 CallableStatement 资源。	

	<pre>// 调用存储过程更新积分 CallableStatement call = conn2.prepareCall(sql: "{CALL update_credits(?, ?)}"); call.setLong(parameterIndex: 1, uid); call.setInt(parameterIndex: 2, distance); call.execute(); call.close(); conn2.commit(); JOptionPane.showMessageDialog(parentComponent: this, message: "Booking Success! Your Reservation No: " + rsvno); }catch (Exception ex) { try {</pre>
创建 存储 过程 源码 (3 分)	<pre>1 DELIMITER // 2 ● CREATE PROCEDURE update_credits1(3 IN p_uid BIGINT, 4 IN p_distance INT 5) 6 BEGIN 7 DECLARE EXIT HANDLER FOR SQLEXCEPTION 8 BEGIN 9 ROLLBACK; 10 RESIGNAL; 11 END; 12 -- 参数校验 13 IF p_distance <= 0 THEN 14 SIGNAL SQLSTATE '45000' 15 SET MESSAGE_TEXT = 'Distance must be positive'; 16 END IF; 17 -- 检查用户是否存在 18 IF NOT EXISTS (SELECT 1 FROM user WHERE uid = p_uid) THEN 19 SIGNAL SQLSTATE '45000' 20 SET MESSAGE_TEXT = 'User does not exist'; 21 END IF; 22 START TRANSACTION; 23 -- 更新用户积分 (假设每公里0.1积分) 24 UPDATE users 25 SET credits = credits + p_distance 26 WHERE uid = p_uid; 27 COMMIT; 28 END // 29 DELIMITER ;</pre> <p>这段 MySQL 代码创建了一个名为 update_credits1 的存储过程，用于根据用户行驶距离自动计算并更新积分。它接收用户 ID 和行驶距离作为输入参数，首先进行参数有效性检查（确保距离为正且用户存在），然后在事务保护下执行积分更新操作。过程中使用了异常处理机制，确保在发生错误时自动回滚事务，同时通过自定义错误信号提供明确的业务异常反馈。这种设计模式将积分计算逻辑集中管理，既保证了数据一致性，又提高了代码可维护性，适用于需要根据用户行为动态调整积分的应用场景。</p>

<p>存储过程执行源码 (1分)</p>	<pre>// 插入预订记录 PreparedStatement ins = conn2.prepareStatement(sql: "INSERT INTO booking VALUES (?, ?, ?, ?, ?, ?);"); ins.setString(parameterIndex: 1, rsvno); ins.setString(parameterIndex: 6, date); ins.setString(parameterIndex: 2, flightNo); ins.setLong(parameterIndex: 3, uid); ins.setInt(parameterIndex: 5, seatno); ins.setDouble(parameterIndex: 4, price); ins.executeUpdate(); ins.close(); // 调用存储过程更新积分 CallableStatement call = conn2.prepareCall(sql: "{CALL update_credits(?, ?)}"); call.setLong(parameterIndex: 1, uid); call.setInt(parameterIndex: 2, distance); call.execute(); call.close();</pre> <p>此部分为插入部分，将数据对应插入 booking 表，下面对应的就是存储部分进行里程积分的更新。</p>
<p>程序演示 (2分)</p>	<p>说明：不违背存储过程，能够执行更新操作 正常情况下，只要点击确认后都会自动弹出购票成功的弹窗，显示与订单号。返回后进入 Bookings 界面可以看到刚刚预定的订单。</p> 
<p>程序演示 (2分)</p>	<p>说明：违背存储过程，系统报错； 由于正常情况下不会报错我们在调用 update_credits1 时进行一个小的处理：uid+999（目前没有这个 uid），模拟随机数据错误。当按下“确定”后立即报错，返回查看 booking 和 information，credits 不变，booking 里没有新的订单，说明该过程能很好地识别存储中的错误，进行报错。</p>

	<div><div><div>From: TSN To: XIY Month: 5 Day: 29 Search Back</div><div><div>Date: 2025-05-29 Flight: CA2878 From: TSN To: XIY Time: 07:15:00-09:20:00 Book</div><div>Date: 2025-05-29 Flight: MF8239 From: TSN To: XIY Time: 11:45:00-14:00:00 Book</div><div>Date: 2025-05-29 Flight: MU2278 From: TSN To: XIY Time: 18:15:00-20:15:00 Book</div><div>Date: 2025-05-29 Flight: BK2884 From: TSN To: XIY Time: 22:15:00-24:00:00 Book</div></div><div><div>Error</div><div><div>X</div><div>Booking failed: User account not found</div><div>确定</div></div></div></div></div>
备注	

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)	查询 uid 用户的所有订单信息, 包括订单号、航班号、日期、出发到达时间、票价、座位号等信息。
视图功能描述 (1 分)	在 booking 界面我需要获取 booking 表里地订单号、价格等信息和 flight 表里的出发到达航班号等信息。因此需要将它们合并为视图, 它们通过 flightno 和 fldate 连接, 生成一张虚拟的表。
涉及的关系表 (2 分)	booking, flight
表连接字段 (1 分)	<p>booking.flightno = flight.flightno</p> <p>booking.fldate = flight.fldate</p>
创建视图代码 (3 分)	<pre> 1 -- 创建用户预订视图 2 • CREATE OR REPLACE VIEW v_user_bookings AS 3 SELECT 4 b.fldate, 5 b.flightno, 6 CONCAT(f.DEP_IATACode, '-', f.ARV_IATACode) AS route, 7 f.distance, 8 b.price, 9 b.seatno, 10 b.rsvno, 11 b.uid, 12 f.departuretime, 13 f.arrivaltime, 14 f.regNo AS aircraft_regno, 15 a.type AS aircraft_type 16 FROM booking b 17 JOIN flight f 18 ON b.flightno = f.flightno 19 AND b.fldate = f.fldate 20 JOIN aircraft a 21 ON f.regNo = a.regno; </pre> <p>通过 CREATE VIEW 新建视图, 将两张表重命名从里分别取出属性。两张表通过 fldate 和 flightno 进行连接, 同时 join 根据 regno 连接 aircraft 获取机型信息。</p>

查询代码 (3分)	<pre>try (Connection conn = DBUtil.getConnection()) { // 使用视图查询 (新增aircraft_type字段) String sql = "" SELECT fldate, flightno, route, distance, price, seatno, rsvno, departuretime, arrivaltime, aircraft_regno, aircraft_type FROM v_user_bookings WHERE uid = ? ORDER BY fldate DESC ""; PreparedStatement ps = conn.prepareStatement(sql); ps.setLong(parameterIndex: 1, uid); ResultSet rs = ps.executeQuery(); while (rs.next()) { String fldate = rs.getString(columnLabel: "fldate"); String flightNo = rs.getString(columnLabel: "flightno"); String route = rs.getString(columnLabel: "route"); } }</pre> <p>由于已经在数据库中构建好视图，这里在 SQL 语句中直接使用 FROM+名称即可。由于订单信息可能会有很多行，这里通过 while(rs.next())循环遍历所有数据，在每一行将得到的数据依次添加，最终得到类似表格的订单信息显示效果。</p>																														
程序演示 (4分)	<p>最终查询结果如下，所有信息均能正确显示。</p>  <table><thead><tr><th>Date</th><th>Flight No</th><th>Route</th><th>Departure</th><th>Arrival</th><th>Aircraft</th><th>Price</th><th>Seat</th><th>Reservation</th><th>Action</th></tr></thead><tbody><tr><td>2025/-05-30</td><td>KE806</td><td>TSN-ICN</td><td>12:20</td><td>15:10</td><td>A330-300</td><td>\$1141.95</td><td>1</td><td>9F4246F0D4238672</td><td>Refund</td></tr><tr><td>2025/-05-29</td><td>MF8239</td><td>TSN-XYY</td><td>11:45</td><td>14:00</td><td>B737-800</td><td>\$1141.95</td><td>1</td><td>FE71CA9B38532AB59</td><td>Refund</td></tr></tbody></table>	Date	Flight No	Route	Departure	Arrival	Aircraft	Price	Seat	Reservation	Action	2025/-05-30	KE806	TSN-ICN	12:20	15:10	A330-300	\$1141.95	1	9F4246F0D4238672	Refund	2025/-05-29	MF8239	TSN-XYY	11:45	14:00	B737-800	\$1141.95	1	FE71CA9B38532AB59	Refund
Date	Flight No	Route	Departure	Arrival	Aircraft	Price	Seat	Reservation	Action																						
2025/-05-30	KE806	TSN-ICN	12:20	15:10	A330-300	\$1141.95	1	9F4246F0D4238672	Refund																						
2025/-05-29	MF8239	TSN-XYY	11:45	14:00	B737-800	\$1141.95	1	FE71CA9B38532AB59	Refund																						
备注	<p>点击右侧的 Refund 按钮会执行退票操作，包括删除数据库中这条记录、还原里程积分。由于个部分完成时间不同，此部分 booking 的 ui 与之前的有所差异，最终程序 ui 以此版本为准。</p>																														

备注：

项目说明：

1. 本项目为航班票务信息系统，包括数据库导出文件 flightticketinfo.sql 和 Java 代码。
2. 使用步骤：
 - a. 在 MySQL Workbench 中导入 flightticketinfo.sql 到数据库 flightticketinfo
 - b. 修改 DBUtil.java 中的用户名/密码为你本地 MySQL 的登录信息
 - c. 编译并运行 Java 项目

建议使用 JDK 8+，确保本机安装了 MySQL 数据库。