

数据库工程作业

要求:

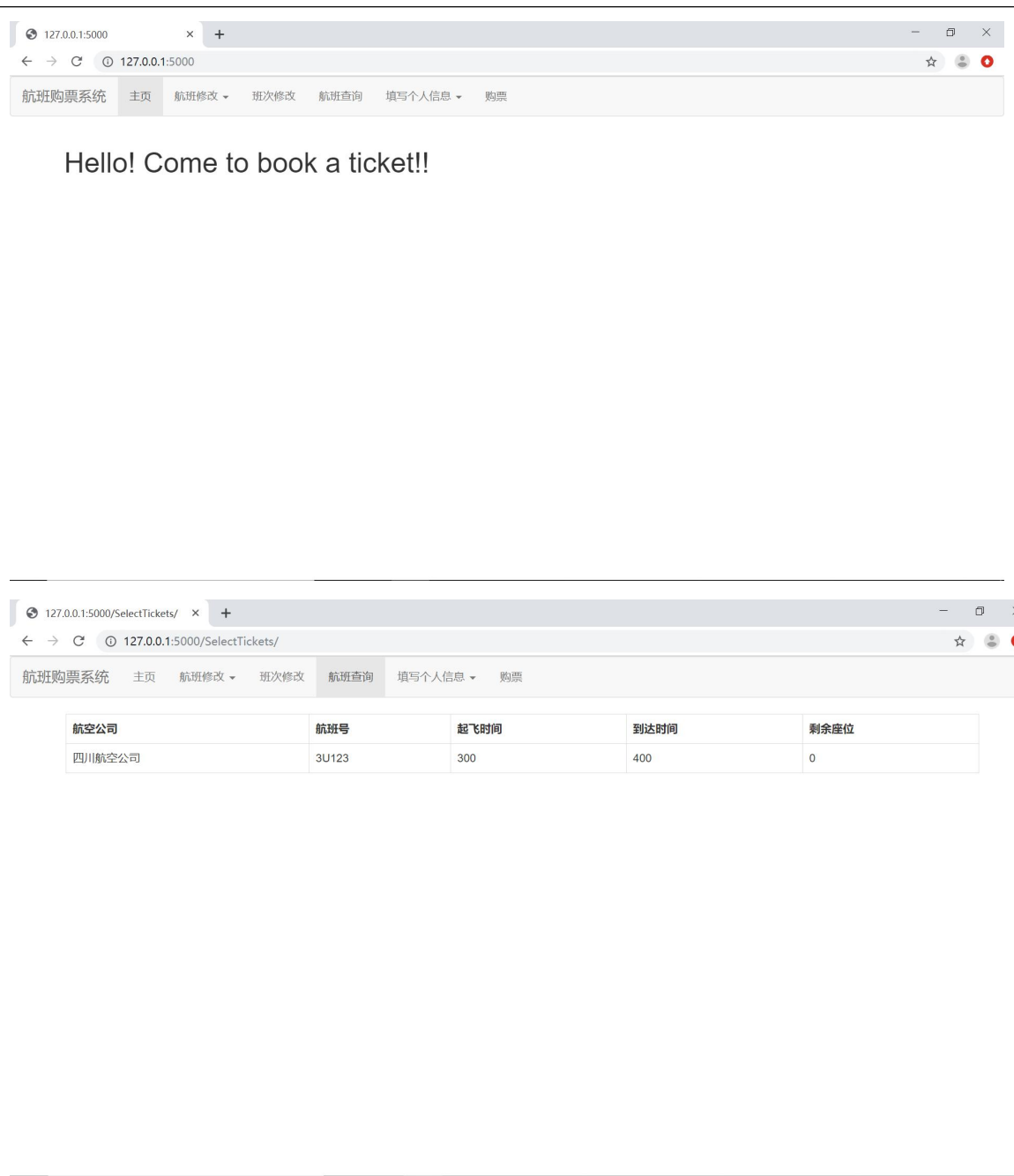
- 1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
- 2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL 等），不要使用桌面数据库。
- 3. 报告中所列举的四种操作，每种操作举一个例子即可。
- 4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。
- 5. 作业完成后，请将工程作业报告和程序上传到学院网站，并联系助教老师进行系统说明和演示，回答相关问题。

工程作业报告

1. 项目信息（10 分）

学号		姓名		专业	
项目名称	航班管理及购票系统				
必备环境	Python3 Flask 及相关拓展 Mysql80				
系统主要功能简介（4分）	航班管理订票系统 <ul style="list-style-type: none">- 添加/删除航班- 添加班次- 航班查询- 购票- 填写/更新个人信息				
系统主要页面截图	截 3-4 个页面即可				

图
(6
分)



127.0.0.1:5000/AddCN/

127.0.0.1:5000/AddCN/

航班购票系统 主页 航班修改 班次修改 航班查询 填写个人信息 购票

Nationality:

English Name:

Tel:

ID:

Chinese Name:

Ethnic:

Submit

127.0.0.1:5000/UpdateTimetable/

127.0.0.1:5000/UpdateTimetable/

航班购票系统 主页 航班修改 班次修改 航班查询 填写个人信息 购票

航班号:

请填写此字段。

出发日期:

起飞时间:

到达时间:

总座位数:

submit

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. 后台数据库使用 MySQL
		2. 端口：3306，用户名：root
		3. 数据库：domestic_flight, 字符集：utf8mb4
	高级 语言	1. Pycharm Professional+Python3.7+virtualenv
		2. http://127.0.0.1/5000
		3. 使用 Flask 及其拓展 flask_wtf、flask_bootstrap、flask_nav 编写页面、表单
		4. 使用库 pymysql 连接数据库，并对数据库进行操作

	序号	名称	功能说明	取值
连接串分析 (6分)	1	class addflights(Flaskform)	定义表单 addflights, 继承自 Flaskform	
	2	AirlineCode=StringField('航空公司编号:', validators=[DataRequired()])	为表单添加属性 AirlineCode, 文本域, 填写表单时不能为空, 填写时的提示为“航空公司编号:”	
	3	@app.route('/addFlights/', methods=['GET', 'POST']) def addFlights():	定义一个路由, 在 127.0.0.1/addFlights 下可以访问 addFlights 函数, 及其定义的表单, 并进行填写	
	3	form=addflights()	form 为表单 addflights 的一个实例	form.data 中各项为向表单中填写的值
	4	AirlineCode=form.data['AirlineCode']	将表单中 AirlineCode 的值赋给一个变量	表单中输入的值
	5	pymysql.connect	用 pymysql 连接 mysql	
	6	sql="insert into ... values(%s);"	定义 sql 语句, %s 为待填参数	
	7	execute(sql, [参数列表])	执行 sql 语句	
	8	temp=add_flight([参数列表])	调用执行数据库操作的函数	返回 “Success” 或错误信息
	9	conn=conn.cursor()	获得一个执行光标	

<p>连接串代码 (截屏) (2 分)</p>	<pre> class addflights(FlaskForm): AirlineCode = StringField('航空公司编号:', validators=[DataRequired()]) DeptAirport = StringField('出发机场编号:', validators=[DataRequired()]) ArriveAirport = StringField('到达机场编号:', validators=[DataRequired()]) FlightNo = StringField('航班号:', validators=[DataRequired()]) submit = SubmitField('submit', validators=[DataRequired()]) @app.route('/addFlights/', methods=['Get', 'Post']) def addFlights(): form = addflights() if form.validate_on_submit(): AirlineCode = form.data['AirlineCode'] DeptAirport = form.data['DeptAirport'] ArriveAirport = form.data['ArriveAirport'] FlightNo = form.data['FlightNo'] temp = add_flight(AirlineCode, DeptAirport, ArriveAirport, FlightNo) return temp return render_template('form.html', form=form) def add_flight(AirlineCode, DeptAirport, ArvAirport, FlightNo): conn = pymysql.connect(host="127.0.0.1", port=3306, user="root", password="mysql", database="Domestic_Flight", charset="utf8mb4") cursor = conn.cursor() sql = "insert into flight(flightno,dept_airport,arv_airport,code) VALUES (%s, %s, %s,%s);" str_ = '' try: # 执行SQL语句 cursor.execute(sql, [FlightNo, DeptAirport, ArvAirport, AirlineCode]) # 提交事务 conn.commit() str_ = 'Success' except Exception as e: # 有异常, 回滚事务 print('traceback.format_exc():\n%s' % traceback.format_exc()) conn.rollback() str_ = str(e) cursor.close() conn.close() return str_ </pre>
<p>备注</p>	

3. 数据库设计 (14 分)

<p>说明</p>	<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>
-----------	--

	<p>物理模型：</p> <pre> erDiagram passenger --o{ ticket : "has" chinese --o{ passenger : "is a" foreigner --o{ passenger : "is a" ticket --o{ flight : "has" flight --o{ airline : "has" flight --o{ airport : "has" flight --o{ timetable : "has" passenger { varchar(50) id_passportno PK varchar(20) nationality bigint(0) tel varchar(50) ename } chinese { varchar(50) id_passportno PK varchar(20) cname char(5) ethnic } foreigner { varchar(50) id_passportno PK varchar(100) visano } ticket { char(7) flightno PK int(0) deptdate smallint(0) seat varchar(50) id_passportno FK } flight { char(7) flightno PK char(4) dept_airport FK char(4) arv_airport FK char(3) code } airline { char(3) code PK varchar(1024) airlinename } airport { char(4) airportno PK varchar(100) airportname } timetable { char(7) flightno PK int(0) deptdate smallint(0) depttime smallint(0) arvtime smallint(0) seats_total smallint(0) seats_current } </pre>
备注	

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）	删除没有已售出机票的航班
涉及的表（2 分）	flight, timetable, ticket
表连	flight.flightno=timetable.flightno=ticket.flightno

接 涉 及 字 段 (1 分)		
删 除 条 件 字 段 描 述 (1 分)	字段	规则
	flightno	如果刚班次没有尚未启程的售票，则可以删除，因为该数据库中所有的删除都是采用的级联模式，所以删除该航班时可以自动删除该班次的所有航班
代 码 (4 分)	(截屏)	
	<pre>drop procedure if exists delete_flight; create procedure delete_flight(flightno_ char(7)) begin if exists(select * from ticket where flightno=flightno_) then signal sqlstate 'HY000' set message_text='请先退回机票'; else delete from flight where flightno=flightno_; end if; end;</pre> <pre>class deleteflights(FlaskForm): FlightNo = StringField('航班号:', validators=[DataRequired()]) submit = SubmitField('submit', validators=[DataRequired()])</pre>	


```
def delete_flight(FlightNo):
    conn = pymysql.connect(
        host="127.0.0.1",
        port=3306,
        user="root",
        password="mysql",
        database="Domestic_Flight",
        charset="utf8mb4"
    )
    cursor = conn.cursor()
    sql = "call delete_flight(%s);"
    try:
        cursor.execute(sql, FlightNo)
        conn.commit()
        str_ = 'Success'
    except Exception as e:
        # 有异常，回滚事务
        print('traceback.format_exc():\n%s' % traceback.format_exc())
        conn.rollback()
        str_ = str(e)

    cursor.close()
    conn.close()
    return str
```

```
@app.route('/deleteFlights/', methods=['Get', 'Post'])
def deleteFlights():
    form = deleteflights()
    if form.validate_on_submit():
        FlightNo = form.data['FlightNo']
        temp = delete_flight(FlightNo)
        return temp
    return render_template('form.html', form=form)
```

程序演示
(4分)

删除班次 CA123

1. 没有已售出的票，及没有未该航班添加班次或者已添加但尚无人购票

对象		timetable @domestic_flight (MySQL...				
开始事务		文本 · 筛选 ↓ 排序 导入 导出				
flightno	deptdate	depttime	arvtime	seats_total	seats_current	
CA123	20200102	1200	1400	2	2	

提交待删除的航班号

← → ↻ 127.0.0.1:5000/deleteFlights/ ☆ ⓘ

航班购票系统

航班号:

CA123

submit

删除成功

127.0.0.1:5000/deleteFlights/ × +

← → ↻ ⓘ 127.0.0.1:5000/deleteFlights/ ☆ ⓘ

Success

2.有已售出的票 (seats_total<>seats_current)

对象 timetable @d... buyT @dome... MySQL80 - ... * 无标题 - ...

开始事务 文本 筛选 排序 导入 导出

flightno	deptdate	depttime	arvtime	seats_total	seats_current
CA123	20200102	1200	1400	1	0

127.0.0.1:5000/deleteFlights/ × +

← → ↻ ⓘ 127.0.0.1:5000/deleteFlights/ ☆ ⓘ

航班购票系统

航班号:

CA123

submit

删除失败

127.0.0.1:5000/deleteFlights/ × +

← → ↻ ⓘ 127.0.0.1:5000/deleteFlights/ ☆ ⓘ

(1644, '请先退回机票')

备注	事务部分（含提交和回滚）由 pymysql 完成
----	--------------------------

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)	根据查询所得到的结果购买机票, 并获得对应座位号, 同时更改飞机上的剩余座位, 在 ticket 表中添加相应信息	
触发器描述 (2 分)	如果当前班次已无余座, 则不能购票 如果该乘客尚未填写个人信息, 则不能购票, 需要先填写个人信息	
涉及的表 (1 分)	timetable, ticket, passenger	
输入数据 (2 分)	字段	规则
	flightno(航班号)	timetable 中存在的 flightno, 且对应当前作为数量 (seats_current) 不为 0, 非空
	deptdate(出发日期)	timetable 中存在的 deptdate, 且对应当前作为数量 (seats_current) 不为 0, 非空, 8 位数字格式
	id_passportno(身份证号/PassportNo.)	非空, 若 passenger 表中不存在这个人的信息, 则将其插入到 passenger 表中
插入操作源码 (3 分)	(截屏)  <pre> drop procedure if exists buyT ; create procedure buyT (FlightNo_ char(5), DeptDate_ integer, ID_PassportNo_ varchar(30)) begin declare seat char(5); set seat=(select seats_current from timetable where flightno=FlightNo_ and deptdate=DeptDate_); insert into ticket(flightno,deptdate,seat,id_passportno) values(FlightNo_,DeptDate_,seat,ID_PassportNo_); set seat=seat-1; update timetable set seats_current=seat where flightno=FlightNo_ and deptdate=DeptDate_; end; </pre>	

```

class buyticket(FlaskForm):
    FlightNo = StringField('航班号:', validators=[DataRequired()])
    DeptDate = StringField('出发日期:', validators=[DataRequired()])
    ID_PassportNo = StringField('身份证号/PassportNo.:', validators=[DataRequired()])
    submit = SubmitField('submit', validators=[DataRequired()])

```

```

def buy(FlightNo, DeptDate, ID_PassportNo):
    conn = pymysql.connect(
        host="127.0.0.1",
        port=3306,
        user="root",
        password="mysql",
        database="Domestic_Flight",
        charset="utf8mb4"
    )
    cursor = conn.cursor()
    sql = "call buyT(%s,%s,%s);"
    try:
        cursor.execute(sql, [FlightNo, DeptDate, ID_PassportNo])
        conn.commit()
        str_ = 'Success'

    except Exception as e:
        # 有异常, 回滚事务
        print('traceback.format_exc():\n%s' % traceback.format_exc())
        conn.rollback()
        str_ = str(e)
    cursor.close()
    conn.close()
    return str_

```

```

@app.route('/BuyTickets/', methods=['Get', 'Post'])
def BuyTickets():
    form = buyticket()
    if form.validate_on_submit():
        FlightNo = form.data['FlightNo']
        DeptDate = form.data['DeptDate']
        ID_PassportNo = form.data['ID_PassportNo']
        temp = buy(FlightNo, DeptDate, ID_PassportNo)
        return temp
    return render_template('form.html', form=form)

```

触发 (截屏)

器源
码
(3
分)

```
drop trigger if exists buy_add;
delimiter //
create trigger buy_add
before insert on ticket for each row
begin
    declare seat smallint(4);
    set seat=(select seats_current from timetable
              where flightno=new.flightno
              and deptdate=new.deptdate);
    if seat<1 then
        signal sqlstate 'HY000' set message_text='该班次机票已售罄';
    end if;
    if new.id_passportno not in (select id_passportno from passenger
                                where id_passportno=new.id_passportno )
    then
        signal sqlstate 'HY000' set message_text='请先填写个人信息';
    end if;
end;//
delimiter ;
```

说明：不违背触发器能够执行插入操作。

对象

timetable @dom...

buyT @domestic...

MySQL80 - 命令...

开始事务

文本

筛选

排序

导入

导出

flightno	deptdate	depttime	arvtime	seats_total	seats_current
CA123	20200102	1200	1400	1	1

程序
演示
(4
分)

127.0.0.1:5000/BuyTickets/

航班购票系统

航班号:

出发日期:

身份证号/PassportNo.:

127.0.0.1:5000/BuyTickets/

127.0.0.1:5000/BuyTickets/

Success

对象 timetable @d... buyT @dome... MySQL80 - ... * 无标题 - ...

开始事务 文本 筛选 排序 导入 导出

flightno	deptdate	depttime	arvtime	seats_total	seats_current
CA123	20200102	1200	1400	1	0

对象 timetable @d... buyT @dome...

开始事务 文本 筛选 排序 导入 导出

flightno	deptdate	seat	id_passportno
CA123	20200102	1	12010000000

说明：违背触发器要求，不能够执行插入操作，系统报错。
票已售罄

对象 timetable @d... buyT @dome... MySQL80 - ... * 无标题 - ...

开始事务 文本 筛选 排序 导入 导出

flightno	deptdate	depttime	arvtime	seats_total	seats_current
CA123	20200102	1200	1400	1	0

127.0.0.1:5000/BuyTickets/

127.0.0.1:5000/BuyTickets/

航班购票系统

航班号:
CA123

出发日期:
20200102

身份证号/PassportNo.:
12010000007

submit

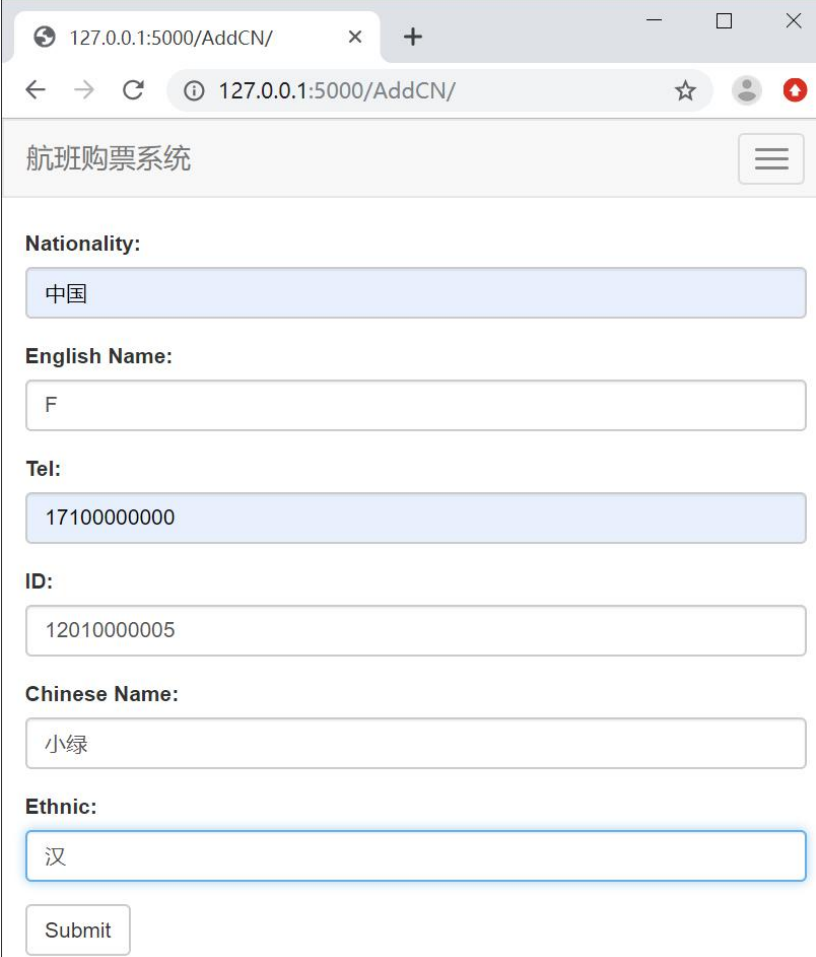
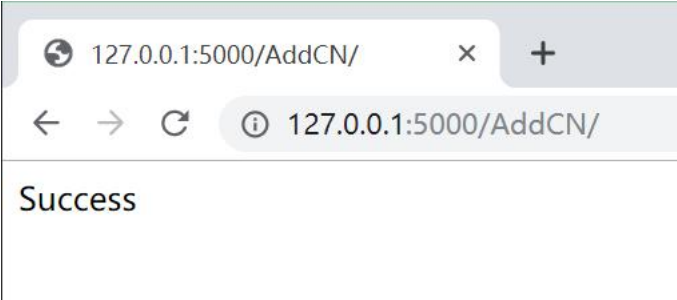
	<div><div>127.0.0.1:5000/BuyTickets/ × +</div><div>← → ↻ ⓘ 127.0.0.1:5000/BuyTickets/</div><div>(1644, '该班次机票已售罄')</div></div> <div>3.没有填写个人信息</div> <div><div>127.0.0.1:5000/BuyTickets/ × +</div><div>← → ↻ ⓘ 127.0.0.1:5000/BuyTickets/</div><div>航班购票系统 主页 航班修改 ▾ 班次修改 航班查询</div><div>航班号:<div>CA123</div></div><div>出发日期:<div>20200102</div></div><div>身份证号/PassportNo.:<div>350000000</div></div><div>submit</div></div> <div><div>127.0.0.1:5000/BuyTickets/ × +</div><div>← → ↻ ⓘ 127.0.0.1:5000/BuyTickets/</div><div>(1644, '请先填写个人信息')</div></div>
备注	

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）	个人信息的存储和更新	
存储过程功能描述 （1分）	<p>如果表中已存在这个人的身份证号或者护照号，则使用 update 更新，如果不存在，则使用 insert 插入</p> <p>中国籍和非中国籍人士需要到相应板块填写个人信息，如果填写错误，则给出提示信息</p>	
涉及的关系表 （2分）	passenger, chinese, foreigner	
表连接涉及字段 （1分）	<p>passenger. id_passportno=chinese. id_passportno</p> <p>passenger. id_passprotno=foreigner. id_passportno</p>	
更改字段 （2分）	字段	规则
	passenger. nationality	当表中已存在表单中输入的身份证号或护照号(id_passportno)
	passenger. tel	
	passenger. ename	
	chinese. cname	
	chinese. ethnic	
更新代码 （3分）	（截屏）	

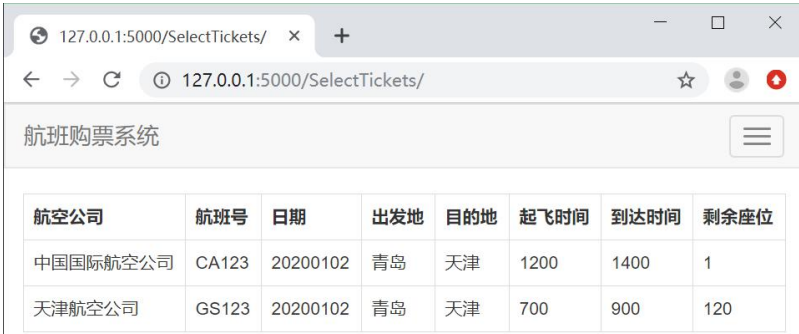
	<pre> drop procedure if exists information_updating_cn; create procedure information_updating_cn(enname_ varchar(50),cnname_ varchar(20),nationality_ varchar(20), tel_ bigint(11),id_passportno_ varchar(50),ethnic_ char(5)) begin if nationality_<>'中国' then signal sqlstate 'HY000' set message_text='国籍填写错误，非中国籍人士请到foreigner板块填写'; end if; if exists (select * from passenger where id_passportno=id_passportno_) then update passenger set tel=tel_ where id_passportno=id_passportno_; update passenger set nationality=nationality_ where id_passportno=id_passportno_; update passenger set enname=enname_ where id_passportno=id_passportno_; else insert into passenger(id_passportno,tel,nationality,enname) values (id_passportno_,tel_,nationality_,enname_); end if; if exists (select * from chinese where id_passportno=id_passportno_) then update chinese set cnname=cnname_ where id_passportno=id_passportno_; update chinese set ethnic=ethnic_ where id_passportno=id_passportno_; else insert into chinese(id_passportno,cnname,ethnic) values(id_passportno_,cnname_,ethnic_); end if; end; </pre> <pre> def add_CN(EnName, CNName, Nationality, Tel, ID, Ethnic): conn = pymysql.connect(host="127.0.0.1", port=3306, user="root", password="mysql", database="Domestic_Flight", charset="utf8mb4") cursor = conn.cursor() sql = "call information_updating_cn(%s,%s,%s,%s,%s,%s);" str_ = '' try: cursor.execute(sql, [EnName, CNName, Nationality, Tel, ID, Ethnic]) conn.commit() str_ = 'Success' except Exception as e: # 有异常，回滚事务 print('traceback.format_exc():\n%s' % traceback.format_exc()) conn.rollback() str_ = str(e) cursor.close() conn.close() return str_ </pre>
创建 存储 过程 源码 (3)	(截 屏)

分)	<pre> drop procedure if exists information_updating_cn; create procedure information_updating_cn(enname_ varchar(50),cnname_ varchar(20),nationality_ varchar(20), tel_ bigint(11),id_passportno_ varchar(50),ethnic_ char(5)) begin if nationality_<>'中国' then signal sqlstate 'HY000' set message_text='国籍填写错误，非中国籍人士请到foreigner板块填写'; end if; if exists (select * from passenger where id_passportno=id_passportno_) then update passenger set tel=tel_ where id_passportno=id_passportno_; update passenger set nationality=nationality_ where id_passportno=id_passportno_; update passenger set enname=enname_ where id_passportno=id_passportno_; else insert into passenger(id_passportno,tel,nationality,enname) values (id_passportno_,tel_,nationality_,enname_); end if; if exists (select * from chinese where id_passportno=id_passportno_) then update chinese set cnname=cnname_ where id_passportno=id_passportno_; update chinese set ethnic=ethnic_ where id_passportno=id_passportno_; else insert into chinese(id_passportno,cnname,ethnic) values(id_passportno_,cnname_,ethnic_); end if; end; </pre>
存储 过程 执行 源码 (1 分)	<p>(截屏)</p> <pre> class CN(FlaskForm): Nationality = StringField('Nationality:', validators=[DataRequired()]) EnName = StringField('English Name:', validators=[DataRequired()]) Tel = StringField('Tel:', validators=[DataRequired()]) ID = StringField('ID:', validators=[DataRequired()]) CNName = StringField('Chinese Name:', validators=[DataRequired()]) Ethnic = StringField('Ethnic:', validators=[DataRequired()]) submit = SubmitField('Submit') @app.route('/AddCN/', methods=['Get', 'Post']) def Add_CN(): form = CN() if form.validate_on_submit(): EnName = form.data['EnName'] CNName = form.data['CNName'] Nationality = form.data['Nationality'] Tel = form.data['Tel'] ID = form.data['ID'] Ethnic = form.data['Ethnic'] return add_CN(EnName, CNName, Nationality, Tel, ID, Ethnic) return render_template('form.html', form=form) </pre>
程序 演示 (2 分)	说明：不违背存储过程，能够执行更新操作

	<div data-bbox="287 197 1107 1151"></div> <div data-bbox="287 1247 967 1545"></div>
程序 演示 (2 分)	说明：违背存储过程，系统报错；

	<div><div>127.0.0.1:5000/AddCN/ × +</div><div>← → ↻ ⓘ 127.0.0.1:5000/AddCN/</div><div>航班购票系统</div><div><div>Nationality:</div><div>英国</div></div><div><div>English Name:</div><div>F</div></div><div><div>Tel:</div><div>17100000000</div></div><div><div>ID:</div><div>12010000005</div></div><div><div>Chinese Name:</div><div>小绿</div></div><div><div>Ethnic:</div><div>汉</div></div><div>Submit</div></div> <div><div>127.0.0.1:5000/AddCN/ × +</div><div>← → ↻ ⓘ 127.0.0.1:5000/AddCN/ ☆</div><div>(1644, '国籍填写错误, 非中国籍人士请到foreigner板块填写')</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）	<p>输入出发日期、出发地、目的地，查询符合条件的班次，输出航空公司、航班号、日期、出发地、目的地、起飞时间、到达时间、剩余座位等信息</p> 
视图功能描述（1 分）	实现表的合并，并将表中的机场代码更换为地点，便于用户查找
涉及的关系表（2 分）	timetable、flight、airport、airline
表连接字段（1 分）	<p>timetable.flightno=flight.flightno</p> <p>flight.dept_airpot=airport.airportno</p> <p>flight.arv_airport=airport.airportno</p> <p>flight.code=airline.code</p>
创	（截屏）

建
视
图
代
码
(3
分)

```
drop view if exists search_tickets;
create view search_tickets
as
select airlinename,t1.flightno as flightno,t1.deptdate as deptdate,
       depttime,arvtime,seats_current,dept,arv
from
(select airlinename,flightno,deptdate,depttime,arvtime,seats_current,
       airportname as dept
from timetable natural join flight natural join airline left outer join
       airport on flight.dept_airport=airport.airportno) as t1
left outer join
(
select flightno,deptdate,airportname as arv
from timetable natural join flight natural join airline left outer join
       airport on flight.arv_airport=airport.airportno
) as t2
on t1.flightno=t2.flightno and t1.deptdate=t2.deptdate
```

查
询
代
码
(3
分)


(截屏)

```
def select_flights(DeptDate, DeptAirport, ArvAirport): # 查timetable
    conn = pymysql.connect(
        host="127.0.0.1",
        port=3306,
        user="root",
        password="mysql",
        database="Domestic_Flight",
        charset="utf8mb4"
    )
    cursor = conn.cursor()

    sql = "select *\
          from search_tickets\
          where deptdate=%s and dept=%s and arv=%s;"

    try:

        cursor.execute(sql, [DeptDate, DeptAirport, ArvAirport])
        rs = cursor.fetchall()
        return render_template('selectform.html', rs=rs) # 返回表格
    except Exception as e:
        print('traceback.format_exc():\n%s' % traceback.format_exc())
        conn.rollback()
        str_ = str(e)
        return str_
```

	<pre> class SelectTicket(FlaskForm): DeptDate=StringField('出发日期:', validators=[DataRequired()]) DeptAirport = StringField('出发地:', validators=[DataRequired()]) ArvAirport = StringField('目的地:', validators=[DataRequired()]) submit = SubmitField('submit', validators=[DataRequired()]) @app.route('/SelectTickets/', methods=['Get', 'Post']) def SelectTickets(): form = SelectTicket() if form.validate_on_submit(): DeptDate = form.data['DeptDate'] DeptAirport = form.data['DeptAirport'] ArvAirport = form.data['ArvAirport'] temp = select_flights(DeptDate, DeptAirport, ArvAirport) return temp return render_template('form.html', form=form) </pre>
<p>程序演示 (4分)</p>	<p>1.</p> 

127.0.0.1:5000/SelectTickets/

127.0.0.1:5000/SelectTickets/

航班购票系统

航空公司	航班号	日期	出发地	目的地	起飞时间	到达时间	剩余座位
中国国际航空公司	CA123	20200102	青岛	天津	1200	1400	1
天津航空公司	GS123	20200102	青岛	天津	700	900	120

2.

127.0.0.1:5000/SelectTickets/

127.0.0.1:5000/SelectTickets/

航班购票系统

出发日期:
20200105

出发地:
青岛

目的地:
天津

submit

127.0.0.1:5000/SelectTickets/

127.0.0.1:5000/SelectTickets/

航班购票系统

航空公司	航班号	日期	出发地	目的地	起飞时间	到达时间	剩余座位
天津航空公司	GS123	20200105	青岛	天津	700	900	100

备注