

# 数据库工程作业

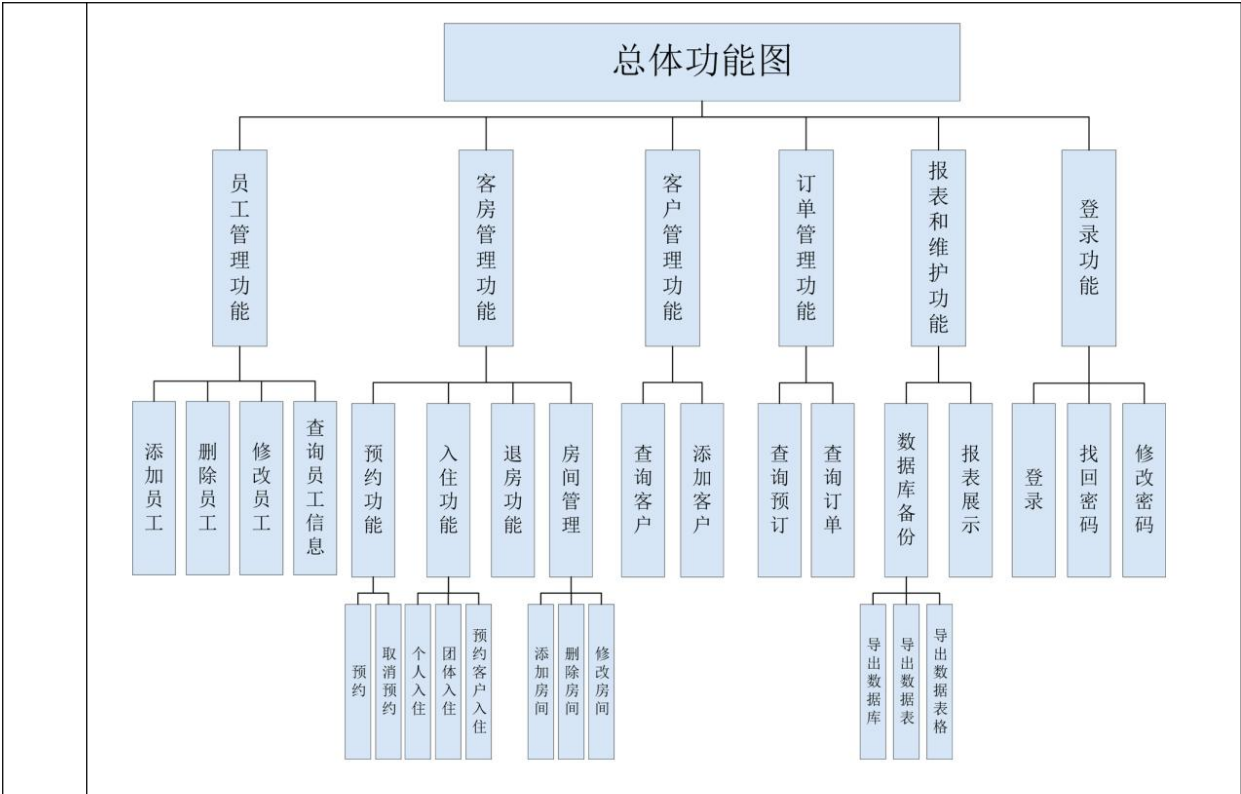
要求:

- 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

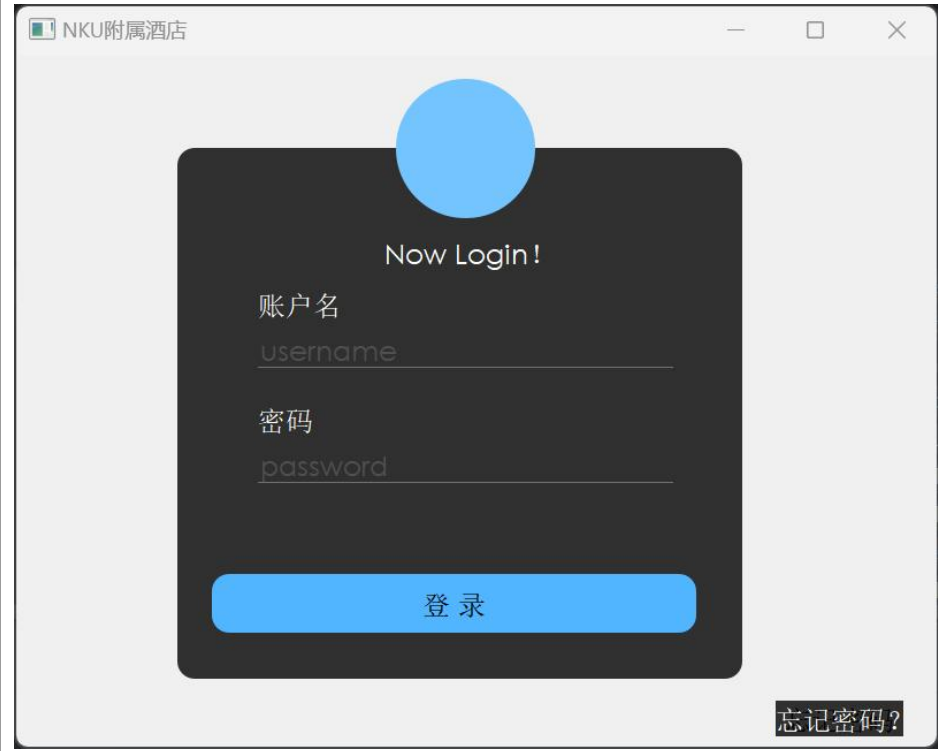
1. 项目信息（10 分）

学号	2212180	姓名	徐海滢	专业	计算机科学与技术
项目名称	酒店住房信息管理系统				
必备环境	1.python 环境（pycharm）2.mysql（navicat）3.pyQt5				
系统主要功能简介（4 分）	酒店住房信息管理系统旨在提升酒店管理效率和现代化水平。通过该信息系统，酒店可以实现信息化管理，提升运营效率，降低人工成本，并提供更好的客户服务体验。实现了登录，员工管理，客房管理，客户管理，订单管理，报表和维护六大功能。具体的功能如下图展示：				



系统主要页面截图（6分）

截 3-4 个页面即可





客房管理界面



员工管理界面



	语言	2. Pycharm2024. 1. 2			
连接串分析 (6 分)	序号	名称	功能说明	取值	
	1	host	数据库服务器的地址。localhost 表示本地服务器。	localhost	
	2	port	MySQL 数据库的端口号，默认是 3306。	3306	
	3	user	用于登录数据库的用户名	root	
	4	passwd	用户的登录密码。	123456	
	5	db	连接的数据库名称	dbdesign	
	6	charset	数据库使用的字符集	Utf8	
	7	cursorclass	pymysql.cursors.DictCursor 返回字典格式的数据。这样每行查询结果是一个字典，键是列名，值是列的值，方便数据处理。	pymysql.cursors.DictCursor	
连接串代码 (截屏) (2 分)	<pre>import pymysql from PyQt5.Qt import * from PyQt5.QtCore import Qt import re  localConfig = {     'host': 'localhost',     'port': 3306,     'user': 'root',     'passwd': '123456',     'db': 'dbdesign',     'charset': 'utf8',     'cursorclass': pymysql.cursors.DictCursor # 数据库操纵指针 }#数据库配置连接</pre>				
备注					

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	staff (员工表)	sid 员工编号	无	无
	2	client (客户表)	cid 客户编号	register_sid	staff(sid 员工编号)
	3	room (房间表)	(rid 房间编号)	无	无
	4	team (团队表)	(tid 团队编号)	check_in_sid	staff(sid 员工编号)
	5	booking_client (客户预订表)	(cid 客户编号, rid 房间)	Cid,rid	client(cid 客户编号), room(rid 房间编号)

			编号)		
	6	booking_team (团队预订表)	(tid 团队编号, rid 房间编号)	Tid,rid	team(tid 团队编号), room(rid 房间编号)
	7	checkin_client (客户入住表)	(rid 房间编号, cid 客户编号)	check_in_sid	room(rid 房间编号), client(cid 客户编号), staff(sid 员工编号)
	8	checkin_team (团队入住表)	(rid 房间编号, tid 团队编号)	check_in_sid	room(rid 房间编号), team(tid 团队编号), staff(sid 员工编号)
	9	hotelorder (订单表)	(id 订单编号, start_time 开始时间, end_time 结束时间, rid 房间编号, ordertype 订单类型)	rid, register_sid	
关系图 (4)	<p>(截屏)</p> <pre>graph LR     PK[个人客户] -- 1 --&gt; 入住  R[客房]     PK -- 1 --&gt; 预订  O[订单]     TK[团体客户] -- 1 --&gt; 入住  R     TK -- 1 --&gt; 预订  O     R -- n --&gt; 管理  E[员工]     E -- m --&gt; 管理  R     E -- p --&gt; 登记  O     O -- q --&gt; 登记  E</pre> <p>个人客户属性: 客户姓名, 客户编号, 客户电话, 客户年龄, 客户性别, 入住次数, 登记时间</p> <p>团体客户属性: 客户姓名, 客户编号, 客户电话, 入住次数, 登记时间</p> <p>客房属性: 房间编号, 房间类型, 房间楼层, 房间描述, 房间图片, 房间价格</p> <p>员工属性: 登录账号, 登录密码, 员工编号, 员工姓名, 员工性别, 入职时间, 身份证号, 员工号码, 权限等级</p> <p>订单属性: 订单金额, 支付方式, 订单编号, 订单类型, 开始时间, 结束时间, 房间编号, 订单评价, 订单时间</p>				
备注					

4. 含有事务应用的删除操作 (13 分)

说明	(1 分) 简要说明该操作所要完成的功能; (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>(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>退房: 该操作用于处理客房退房, 包括个人退房和团队退房。操作会在 checkin_client 或 checkin_team 表中删除相应记录, 并在 hotelorder (完成的订单) 表中添加一条新的订单记录。</p>	
涉及的表 (2 分)	<ul style="list-style-type: none"> <li>• checkin_client (入住个人客户表)</li> <li>• checkin_team (入住团体表)</li> <li>• hotelorder (完成订单表)</li> </ul>	
表连接涉及字段 (1 分)	<p>checkin_client.rid = hotelorder.rid</p> <p>checkin_team.rid = hotelorder.rid</p>	
删除条件字段描述 (1 分)	字段	规则
	<p>checkin_client.rid = ? AND</p> <p>checkin_client.cid = ? AND</p> <p>checkin_client.start_time = ?</p> <p>AND checkin_client.end_time = ?</p>	<p>个人退房: 输入个人身份证号和房间号, 选择开始时间和结束时间, 删除相对应的住房记录</p>
	<p>checkin_team.rid = ? AND</p> <p>checkin_team.tid = ?</p> <p>checkin_team.start_time = ?</p> <p>AND checkin_team.end_time = ?</p>	<p>团体退房: 输入团队编号和房间号, 选择开始时间和结束时间删除相对应的住房记录</p>

<p>代码 (4分)</p>	<pre> def checkoutDB(self, flag, id, rid, payType, remark):     """两种方式退房"""     try:         self.db.begin() # 开始事务         if flag == '个人':             self.cursor.execute( query: "select * from checkin_client where rid=%s and cid=%s", args: (rid, id))             data = self.cursor.fetchall()             if data == ():                 QMessageBox().information(None, "提示", "没有相关入住信息!", QMessageBox.Yes)                 return False             else:                 rid_out = data[0]['rid']                 cid_out = data[0]['cid']                 stime_out = data[0]['start_time']                 etime_out = data[0]['end_time']                 money = data[0]['total_price']                 self.cursor.execute(                     query: "insert into hotelorder(id, ordertype, start_time, end_time, rid, pay_type, money, remark, register_sid)                     args: (cid_out, flag, stime_out, etime_out, rid_out, payType, money, remark, self.staff.sid))                 self.cursor.execute( query: "delete from checkin_client where rid=%s and cid=%s", args: (rid_out, cid_out))                 self.db.commit()                 QMessageBox().information(None, "提示", "本次需要支付%s" % money, QMessageBox.Yes)          elif flag == '团队':             sum = 0             for r in re.split( pattern: '[.   ]', rid):                 self.cursor.execute(                     query: "select * from checkin_team where rid=%s and tid=%s", args: (r, id))                 data = self.cursor.fetchall()                 if data == ():                     QMessageBox().information(None, "提示", "没有相关入住信息!", QMessageBox.Yes)                     return False                 else:                     rid_out = data[0]['rid']                     tid_out = data[0]['tid']                     stime_out = data[0]['start_time']                     etime_out = data[0]['end_time']                     money = data[0]['total_price']                     self.cursor.execute(                         query: "insert into hotelorder(id, ordertype, start_time, end_time, rid, pay_type, money, remark, register_sid)                         args: (tid_out, flag, stime_out, etime_out, rid_out, payType, money, remark, self.staff.sid))                     self.cursor.execute( query: "delete from checkin_team where rid=%s and tid=%s", args: (rid_out, tid_out))                     self.db.commit()                     sum = sum + int(money)                 QMessageBox().information(None, "提示", "本次需要支付%s" % str(sum), QMessageBox.Yes)             self.db.commit() # 提交事务             return True         except Exception as e:             self.db.rollback() # 回滚事务             print(e)             return False </pre>
<p>程序演示 (4分)</p>	





	<div><div><div>MainWindow</div><div><div><div><div><div><div></div><div>xhy</div><div>权限: 2</div><div>*表示需要最高权</div></div><div>搜索</div></div><div><div>客房信息</div><div>办理入住</div><div>预定登记和...</div><div>办理退房</div><div>更新房源*</div></div></div></div><div><div>办理退房</div><div><div>退房方式:<div><div><input checked="" type="radio"/> 个人</div><div><input type="radio"/> 团队</div></div></div><div>客户身份证或团队标识<input type="text" value="330304200109089178"/></div><div>起止时间:<div><div><input type="text" value="2024/6/2"/></div><div><input type="text" value="2024/6/4"/></div></div></div><div><div><div><div>提示</div><div><div><div></div><div>i</div></div><div>退房成功!</div></div><div>Yes</div></div></div></div><div>备注和评价:<input type="text"/></div><div>确认退房</div></div></div></div></div></div>
备注	

## 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)	该操作用于在数据库中添加一个新的房间记录。该操作会将新的房间信息插入到 room 表中。	
触发器描述 (2 分)	该触发器用于在插入新的房间记录后, 对某些相关表或数据进行更新或验证, 以确保数据的完整性和一致性。例如, 可以创建一个触发器来记录每次添加房间的操作日志, 或者检查插入的数据是否符合某些业务规则。	
涉及的表 (1 分)	Room	
输入数据 (2 分)	字段	规则
	rid (房间号)	必须为三位数字
	rtype (房间类型)	在几种规定的房型中选择一种
	rstorey (楼层)	必须为 1 位数字, 范围在 2 到 4 之间
	rprice (房间价格)	无
	rdesc (房间描述)	可以为空
	rpics (房间图片)	可以为空
插入操作源码 (3 分)	(截屏)	

	<pre>def addRoom(self, rid, rtype, rstorey, rprice, rdesc, rpics):     """增加房间"""     try:         # 验证房间号是否为三位数         if not (rid.isdigit() and len(rid) == 3):             QMessageBox().information(None, "提示", "房间号必须为三位数!", QMessageBox.Yes)             return False          if rid[0] != rstorey:             QMessageBox().information(None, "提示", "房间号的第一个数字必须与楼层相等!", QMessageBox.Yes)             return False          # 插入房间信息         self.cursor.execute(query="insert into room values(%s, %s, %s, %s, %s, %s)",                             args=(rid, rtype, rstorey, rprice, rdesc, rpics))         self.db.commit()         return True     except Exception as e:         print(e)         QMessageBox().information(None, "提示", "房间号已存在或其他错误!", QMessageBox.Yes)         return False</pre>
触发器源码 (3分)	<pre>DELIMITER ;; CREATE TRIGGER before_room_insert BEFORE INSERT ON room FOR EACH ROW BEGIN     DECLARE msg VARCHAR(255);      -- 检查房间号是否为三位数     IF CHAR_LENGTH(NEW.rid) != 3 OR NEW.rid NOT REGEXP '^([0-9]){3}\$' THEN         SET msg = '房间号必须为三位数!';         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;     END IF;      -- 检查房间号的第一个数字是否与楼层相等     IF LEFT(NEW.rid, 1) != NEW.rstorey THEN         SET msg = '房间号的第一个数字必须与楼层相等!';         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;     END IF;      -- 记录日志（假设有一个room_log表）     INSERT INTO room_log (rid, action, action_time)     VALUES (NEW.rid, 'INSERT', NOW());  END;; DELIMITER ;</pre>
程序演示 (4分)	说明：不违背触发器能够执行插入操作。

MainWindow

更新房源

增添房间 修改房源

xhy

权限: 2

\*表示需要最高权

搜索

客房信息

办理入住

预定登记和...

办理退房

更新房源\*

输入要增设的房间信息:

房间号:

房间号

房间类型:

请选择...

房间楼层:

请选择...

房间单价(元/晚)

单价

描述:

描述

展示图片选择:

图片路径

浏览

确认增加

MainWindow

更新房源

增添房间 修改房源

xhy

权限: 2

\*表示需要最高权

搜索

客房信息

办理入住

预定登记和...

办理退房

更新房源\*

输入要增设的房间信息:

房间号:

401

房间类型:

总统套房

房间楼层:

4

房间单价(元/晚)

1000

描述:

华总统套房, 体现您的身份

展示图片选择:

图片路径

浏览

确认增加



说明：违背触发器要求，不能够执行插入操作，系统报错。

1. 添加的房号已经存在

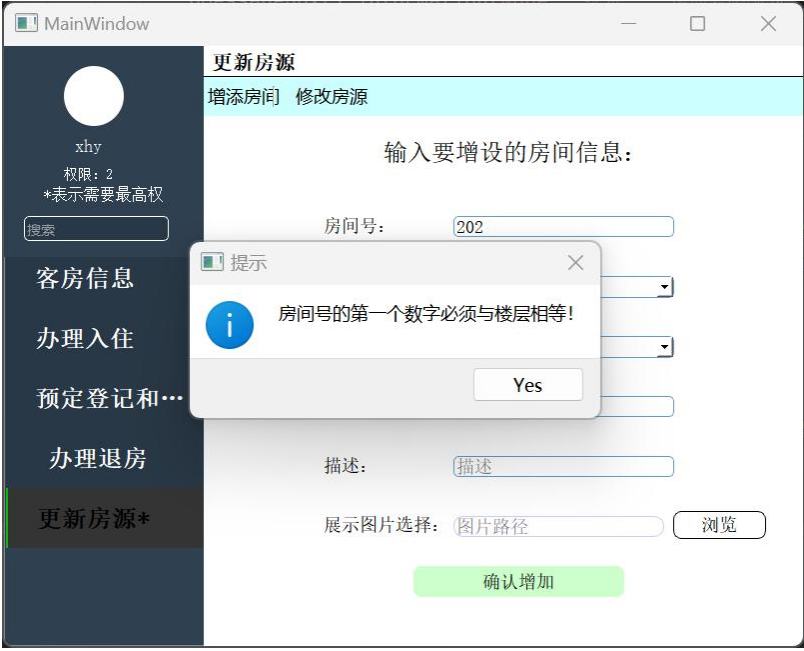


2. 房间号格式错误 不为三位数

程序  
演示  
(4  
分)



3.房间号与层数不匹配



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）	在一个酒店管理系统中实现对员工信息的修改。特别是当修改员工 ID（sid）时，同时更新其他相关表中的外键。	
存储过程功能描述 （1分）	当 staff 中的信息	
涉及的关系表 （2分）	<ul style="list-style-type: none"><li>• staff</li><li>• checkin_client</li><li>• checkin_team</li><li>• client</li><li>• hotelorder</li><li>• team</li></ul>	
表连接涉及字段 （1分）	checkin_client.check_in_sid = staff.sid checkin_team.check_in_sid = staff.sid client.register_sid = staff.sid hotelorder.register_sid = staff.sid team.check_in_sid = staff.sid	
更改字段 （2分）	字段	规则
	sid - 员工 ID	新员工 ID 替换旧员工 ID，并更新其他表中的相关外键字段。
	sname - 员工姓名	新员工姓名替换旧员工姓名
	susername - 用户名	新用户名替换旧用户名
	spassword - 密码	新密码替换旧密码
	srole - 权限	新的权限取代旧的权限
	Sphone-电话号码	新电话号码取代旧电话号码
	ssex - 性别	不可以更改的字段
	sidcard - 身份证号	
更新代码	（截屏）	



(3  
分)

```
def modifyStaff(self, row, column, value):
    """通过存储过程更新员工信息"""
    SQL_COLUMN = ['sid', 'sname', 'ssex', 'stime', 'susername', 'spassword', 'srole', 'sidcard', 'sphone']
    try:
        # 获取需要修改的员工信息
        self.cursor.execute("select * from staff")
        data = self.cursor.fetchall()
        sid_selected = data[row]['sid']
        column_name = SQL_COLUMN[column]

        # 禁止更新性别和身份证号
        if column_name in ['ssex', 'sidcard']:
            QMessageBox().information(None, "提示", "性别和身份证号不能更新!", QMessageBox.Yes)
            return False

        if column_name == 'sid':
            # 更新 sid 时, 调用存储过程
            new_sid = value
            self.cursor.callproc('update_staff', (new_sid, data[row]['sname'], data[row]['stime'], data[row]['susername'], data[row]

    else:
        # 非 sid 更新的情况, 调用存储过程
        update_query = f"UPDATE staff SET {column_name} = %s WHERE sid = %s"
        self.cursor.execute(update_query, (value, sid_selected))
        self.db.commit()
        return True
except Exception as e:
    print(e)
    return False
```

创建  
存储  
过程  
源码  
(3  
分)

(截屏)

```
DELIMITER //

CREATE PROCEDURE update_staff(
    IN p_sid INT,
    IN p_sname VARCHAR(50),
    IN p_stime DATE,
    IN p_susername VARCHAR(50),
    IN p_sp password VARCHAR(50),
    IN p_srole VARCHAR(20),
    IN p_sphone VARCHAR(20)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    START TRANSACTION;

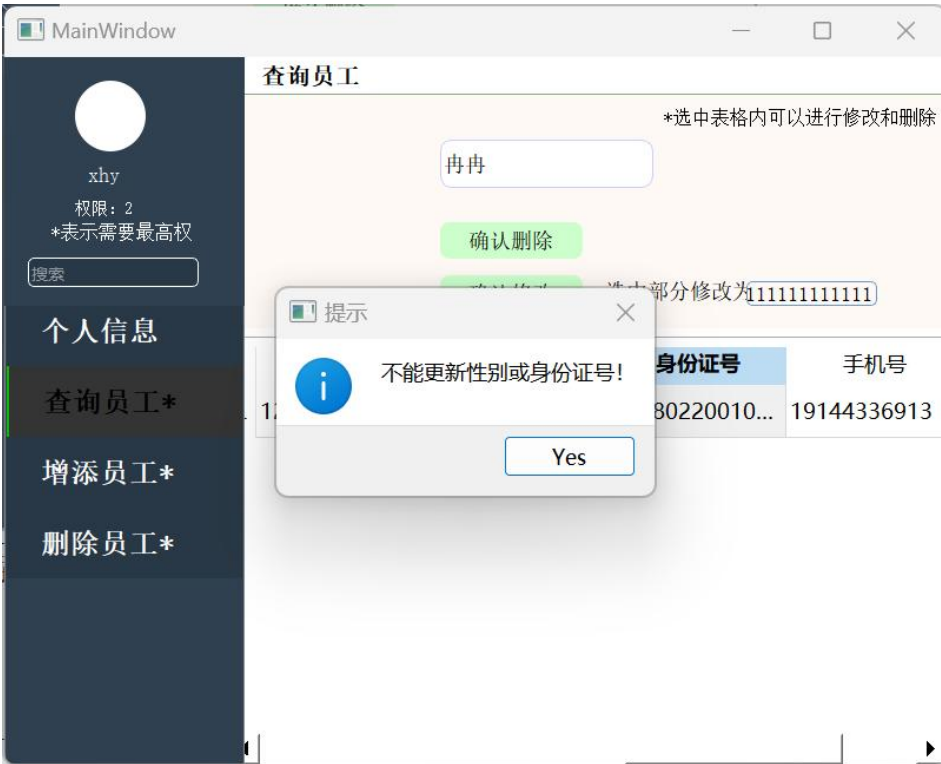
    -- 禁用外键约束
    SET FOREIGN_KEY_CHECKS = 0;
```

	<pre>-- 更新其他表中的外键 UPDATE checkin_client SET check_in_sid = p_sid WHERE check_in_sid = (SELECT sid FROM staff WHERE sid = p_sid) UPDATE checkin_team SET check_in_sid = p_sid WHERE check_in_sid = (SELECT sid FROM staff WHERE sid = p_sid) UPDATE client SET register_sid = p_sid WHERE register_sid = (SELECT sid FROM staff WHERE sid = p_sid) UPDATE hotelorder SET register_sid = p_sid WHERE register_sid = (SELECT sid FROM staff WHERE sid = p_sid) UPDATE team SET check_in_sid = p_sid WHERE check_in_sid = (SELECT sid FROM staff WHERE sid = p_sid)  -- 更新 staff 表中的信息，但不包括性别和身份证号 UPDATE staff SET sname = p_sname,     stime = p_stime,     susername = p_susername,     spassword = p_spasword,     srole = p_srole,     sphone = p_sphone WHERE sid = p_sid;  -- 启用外键约束 SET FOREIGN_KEY_CHECKS = 1;  COMMIT; END //</pre> <p>DELIMITER ;</p>
存储过程执行源码 (1分)	<p>(截屏)</p> <pre>self.cursor.callproc('update_staff', (new_sid, data[row]['sname'], data[row]['stime'], data[row]['susername'], data[row]['spasword'], data[row]['srole'], data[row]['sphone']))</pre>
程序演示 (2分)	<p>说明：不违背存储过程，能够执行更新操作</p> 



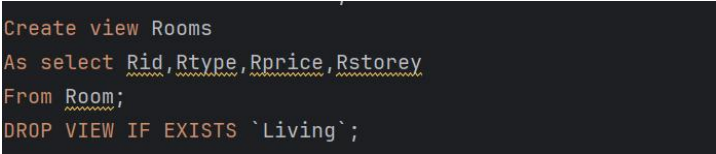
程序  
演示  
(2  
分)

说明：违背存储过程，系统报错；



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）	该操作通过创建一个视图来简化复杂查询，使得客房信息查询更加高效和易于管理。该视图将综合展示客房的详细信息，包括房间类型、价格、楼层等，以及当前的入住状态。查询视图与订房功能链接，可以通过查询窗口进行订房。
视图功能描述（1 分）	视图综合了房间表(room)、入住个人客户表(checkin_client)和入住团队表(checkin_team)的信息，提供每个房间的详细信息及其当前的入住状态。注意，每次搜索之后下一次搜索之前需要点击重置再进行搜索。
涉及的关系表（2 分）	<ul style="list-style-type: none"> <li>room</li> <li>checkin_client</li> <li>checkin_team</li> </ul>
表连接字段（1 分）	<p>room.rid = checkin_client.rid</p> <p>room.rid = checkin_team.rid</p>
创建视图代码	<p>（截屏）</p>  <pre>Create view Rooms As select Rid,Rtype,Rprice,Rstorey From Room; DROP VIEW IF EXISTS `Living`;</pre>

码 ( 3 分)	
查 询 代 码 ( 3 分)	<p>(截屏)</p> <pre> def showRoom(self,rtype,rstate,rstorey,rstarttime,rendtime,price_bottom,price_up):     """ 根据条件检索房间"""     print(rstarttime, rendtime)     if rstate == 0:         self.cursor.execute( query: "select * from room where rtype like %s and rstorey like %s and rprice between %s and %s",                                args: (rtype,rstorey,int(price_bottom),int(price_up)))         data1 = self.cursor.fetchall()         return data1     elif rstate == 1:         self.cursor.execute(             query: "select rid from room where rtype like %s and rstorey like %s and rprice between %s and %s",             args: (rtype, rstorey, int(price_bottom), int(price_up)))         data = self.cursor.fetchall()         list_data = []         for i in range(len(data)):             crid = data[i]['rid']             self.cursor.execute(                 query: "select * from checkin_client as A where (A.rid=%s) and (A.end_time&gt;%s and A.start_time&lt;%s "                        "or A.end_time&gt;%s and A.start_time&lt;%s or A.start_time&lt;=%s and A.end_time&gt;=%s or A.start_time&gt;=%s and A.end_time&lt;=%s",                 , args: (crid, rstarttime, rstarttime, rendtime, rendtime, rstarttime, rendtime,rstarttime,rendtime))             data1 = self.cursor.fetchall()             self.cursor.execute(                 query: "select * from checkin_team as A where (A.rid=%s) and (A.end_time&gt;%s and A.start_time&lt;%s "                        "or A.end_time&gt;%s and A.start_time&lt;%s or A.start_time&lt;=%s and A.end_time&gt;=%s or A.start_time&gt;=%s and A.end_time&lt;=%s",                 , args: (crid, rstarttime, rstarttime, rendtime, rendtime, rstarttime, rendtime,rstarttime,rendtime))              data2 = self.cursor.fetchall()             self.cursor.execute(                 query: "select * from booking_client as A where (A.rid=%s) and (A.end_time&gt;%s and A.start_time&lt;%s "                        "or A.end_time&gt;%s and A.start_time&lt;%s or A.start_time&lt;=%s and A.end_time&gt;=%s or A.start_time&gt;=%s and A.end_time&lt;=%s",                 , args: (crid, rstarttime, rstarttime, rendtime, rendtime, rstarttime, rendtime,rstarttime,rendtime))             data3 = self.cursor.fetchall()             self.cursor.execute(                 query: "select * from booking_team as A where (A.rid=%s) and (A.end_time&gt;%s and A.start_time&lt;%s "                        "or A.end_time&gt;%s and A.start_time&lt;%s or A.start_time&lt;=%s and A.end_time&gt;=%s or A.start_time&gt;=%s and A.end_time&lt;=%s",                 , args: (crid, rstarttime, rstarttime, rendtime, rendtime, rstarttime, rendtime,rstarttime,rendtime))             data4 = self.cursor.fetchall()             if data1 == () and data2 == () and data3 == () and data4 == ():                 list_data.append(crid)         ret = []         for i in range(len(list_data)):             rid_ret = list_data[i]             self.cursor.execute( query: "select * from room where rid=%s", (rid_ret))             ret = ret + self.cursor.fetchall()         return ret </pre>
程 序 演 示 ( 4 分)	<p>1.首先查询三楼的所有房间，结果如下</p>

xhy

权限: 2

\*表示需要最高权

搜索

客房信息

办理入住

预定登记和...

办理退房

更新房源\*

客房信息

房型: 大床房 房间状态 ☐ 只看空闲

楼层: 3 预计入住时间 2024/6/2

— 2024/6/12

价格范围: —

搜索 重置

房间号:303 楼层:3 房间号:305 楼层:3

类型:大床房 类型:大床房

描述:无 价格:258 描述:设施新 价格:258

立即订购 立即订购

2.接着查询三楼所有空闲的大床房，发现房间 305 不见了。因为该段时间内 305 房有人入住

xhy

权限: 2

\*表示需要最高权

搜索

客房信息

办理入住

预定登记和...

办理退房

更新房源\*

客房信息

房型: 大床房 房间状态 ☒ 只看空闲

楼层: 3 预计入住时间 2024/6/2

— 2024/6/12

价格范围: —

搜索 重置

房间号:303 楼层:3

类型:大床房

描述:无 价格:258

立即订购

3. 若点击立即订购按钮，则跳转到办理入住界面

	<div><div><div>MainWindow</div><div><div><div><div><div><div></div></div><div><div>xhy</div><div>权限: 2</div><div>*表示需要最高权</div></div><div>搜索</div></div><div><div>客房信息</div><div>办理入住</div><div>预定登记和...</div><div>办理退房</div><div>更新房源*</div></div></div></div><div><div>办理入住</div><div>散客入住 团队入住 预定入住</div><div><div>客户信息</div><div>客户姓名: <input type="text" value="姓名"/></div><div>客户身份证: <input type="text" value="身份证"/></div><div>客户手机号: <input type="text" value="手机号"/></div><div>客户年龄: <input type="text" value="年龄"/></div><div>客户性别: <input type="radio"/> 男 <input type="radio"/> 女</div></div><div><div>入住信息</div><div>房间号: <input type="text" value="303"/></div><div>预计退房时间: <input type="text" value="2024/6/12"/></div><div>备注: <input type="text"/></div><div>确认入住</div></div></div></div></div></div>
备注	