四 川 轻 化 工 大 学 实 验 报 告

院：**计算机科学与工程学院** 课程名称：**数据库系统原理** 日期：2022.5.25

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | **李万余，胡鹏，罗荣** | 组号 | 1 | 学号 | **2022** | 实验室 | A6-329 |
| 专业 | **软件工程** | 班号 | **2020级4班** | 老师签名 |  |
| 实验  名称 | **实验二、数据库的查询和维护实验** | | | | | 成绩评定 |  |
| 所用  仪器  材料 | **计算机，**  **Word 2019，**  **Visio 2016，**  **MySQL 8.0.28，**  **Navicat for MySQL 15** | | | | | | |
| 实  验  目  的  或  要  求 | **项目名称：物业管理系统**  **一、实验目的**   1. 熟练掌握使用 SQL语言进行简单表的数据查询、数据排序和数据联结查询的操作方法。 2. 熟练掌握使用 SQL语言进行数据库修改数据的操作。 3. 掌握数据库中的视图创建语句，掌握视图查询与操作语句，加深对视图作用的理解。   **二、实验内容及要求**  1.简单查询操作：包括投影、选择条件表达、数据排序、使用临时表等。  2.连接查询操作：包括等值连接、自然连接、求笛卡尔积、一般连接、外连接、内连接、左连接、右连接和自连接等。  3.数据维护操作：通过Navicat实现对数据库库的数据删除和数据修改操作。  4.创建、查看、修改和删除视图。  **具体内容及要求：**根据需求分析规划的每一个功能，编写该功能所涉及的所有查询和修改的相应SQL语句。选择1-2个用户端操作表建立视图，并进行编写相应的操作语句。 | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实  验  原  理 | **三、设计过程及结果：**  1 本实验涉及的功能  本系统由业主模块、管理员模块两部分组成。  业主模块包括由账户登录\注册、查看消息、投诉/建议、服务申请、缴费、查看房屋信息六部分组成。  管理员模块包括由查看账户、发布消息、评价管理、维修管理、服务管理五部分组成  图 1-1 系统功能模块  2 每个功能的实现  **1．用户登录/注册**  登录账户，注册账户   |  |  | | --- | --- | | （1）登录 | （2）注册 |   SQL语句：  （1）存在查询  CREATE PROCEDURE verify\_account\_up(IN value VARCHAR(50))  BEGIN  SELECT IFNULL((SELECT TRUE FROM userinfo WHERE userphone = value LIMIT 1), FALSE) AS result;  END$$  CREATE PROCEDURE verify\_account\_ue(IN value VARCHAR(50))  BEGIN  SELECT IFNULL((SELECT TRUE FROM userinfo WHERE useremail = value LIMIT 1), FALSE) AS result;  END$$  （2）密码查询  CREATE PROCEDURE get\_password\_up(IN value VARCHAR(50))  BEGIN  SELECT userpassword FROM userinfo WHERE userphone LIKE value;  END$$  CREATE PROCEDURE get\_password\_ue(IN value VARCHAR(50))  BEGINE:\MySql\_Learn\src\SQL\login.sql  SELECT userpassword FROM userinfo WHERE useremail LIKE value;  END$$  （3）注册（用户插入）  INSERT INTO userinfo(username, userpassword, userphone) VALUES ('text', 'test', '00000000000');  INSERT INTO userinfo(username, userpassword, useremail) VALUES ('text', 'test', 'test@test.test');  INSERT INTO userinfo(username, userpassword, userphone, useremail) VALUES ('text', 'test', '00000000001', 'test1@test.test');  UPDATE userinfo set isactive = 1 WHERE userphone = '00000000000';  UPDATE userinfo set isactive = 1 WHERE useremail = 'test@test.test';  执行效果截图：   |  | | --- | | （1）验证手机号： | | （2）验证邮箱： | | （3）通过手机号获取密码： | | （4）通过邮箱获取密码： |   **2．用户查看消息**  显示所有与该用户有关的消息，点击可查看详细信息：   |  | | --- | | （1）显示所有与该用户有关的消息 | | （2）显示一条消息的详细信息 |   SQL语句：   1. 查询所有与该用户有关的消息   SELECT newsid,newstitle,newstime,newsendtime,adminname FROM newsinfo,admininfo WHERE news\_adminid = adminid and newstype = 0  UNION ALL  SELECT newsid,newstitle,newstime,newsendtime,adminname FROM admininfo,newsinfo,user\_newsrelation WHERE news\_adminid = adminid and un\_newsid = newsid and un\_userid = 23  ORDER BY newstime DESC;   1. 查询一条消息的详细信息   SELECT newsid,newscontent,newstime,newsendtime,adminname FROM newsinfo,admininfo WHERE news\_adminid = adminid and newsid = 1  执行效果截图：   |  | | --- | | （1）显示所有与该用户有关的消息 | | （2）显示一条消息的详细信息 |   **3. 用户发表/查看投诉/建议**  按发表时间显示该用户的投诉/建议，可发表投诉/建议   |  |  | | --- | --- | | （1）查看投诉/建议 | （2）发表投诉/建议 |   SQL语句：  （1）查询该用户发表的所有投诉/建议  SELECT c\_aid,c\_acontent,c\_atime,c\_aobject,adminname FROM complaint\_and\_acviceinfo LEFT OUTER JOIN admininfo ON c\_a\_adminid = adminid WHERE c\_a\_userid = 1;  （2）增加一条投诉/建议  INSERT INTO complaint\_and\_acviceinfo(c\_acontent,c\_atime,c\_aobject,c\_a\_userid) VALUES('xxxxxxx','2022-05-26 05:17:28','xxx', 1);  执行效果截图：   |  | | --- | | 查询该用户发表的所有投诉/建议 |   **4. 用户申请服务**  停车位查询，购买，维修任务申请，查看详情   |  |  | | --- | --- | | （1）查看拥有车位 | （2）查看未出售车位 | | （3）车位购买 | （4）查看维修任务 | | （5）查看维修详情 | （6）维修申请 |   SQL语句：  （1）查询该用户的停车位  SELECT pcarid, pcarloction, buytime, location FROM pcarinfo, houseinfo, user\_houserelation WHERE pcar\_houseid = houseid and pcar\_houseid = uh\_relation\_houseid and uh\_relation\_userid = 1;  （2）查询空车位  SELECT pcarid, pcarloction FROM pcarinfo WHERE pcarstate = 0;  （3）修改车位信息  UPDATE pcarinfo SET pcarstate = 1, pcar\_houseid = 33, buytime = localtime() WHERE pcarid = 99;  （4）查询维修任务  SELECT taskcontent,taskstate FROM taskinfo WHERE task\_houseid = 1;  （5）查询任务详情  SELECT taskcontent,taskstate,taskstarttime,taskendtime,ataskstate  FROM taskinfo LEFT OUTER JOIN admin\_taskrelation  ON admin\_task\_taskid = taskid WHERE taskid = 1;  （6）插入维修任务  INSERT INTO taskinfo(taskstate, taskcontent, task\_houseid) VALUES (0, 'test', 1);  执行效果截图：   |  | | --- | | 查询该用户的停车位 | | 查询空车位 | | 查询维修任务 | | 查询任务详情 |   **5. 用户缴费**  查看费用详情，可线上缴费   |  |  | | --- | --- | | （1）费用详情 | （2）在线缴费 |   SQL语句：  （1）查询房屋费用  SELECT \* FROM costinfo, costtypeinfo WHERE cost\_costtypeid = costtypeid AND cost\_houseid = 1;  （2） 缴费修改  UPDATE costinfo set paytime = localtime() WHERE costid = 88;  执行效果截图：   |  | | --- | | 查询房屋费用 |   **6. 用户查看房屋信息**  显示房屋详细信息   |  | | --- | | 房屋详情 |   SQL语句：  （1）查询该用户的房屋信息  SELECT houseid,location,gethousetime,movetime,adminname FROM houseinfo, user\_houserelation, admininfo WHERE uh\_relation\_houseid = houseid and house\_adminid = adminid and uh\_relation\_userid = 1;  执行效果截图：   |  | | --- | | 查询该用户的房屋信息 |   **7. 管理员账户管理**  登录管理员账户，查看上司的信息，‘删除’无用的用户账户   |  |  | | --- | --- | | （1）登录管理员账户 | （2）查看上司 | | （3）删除用户账户 | （4）查看下属 |   SQL语句：  （1）存在查询  CREATE PROCEDURE verify\_account\_ap(IN value VARCHAR(50))  BEGIN  SELECT IFNULL((SELECT TRUE FROM admininfo WHERE adminphone = value LIMIT 1), FALSE) AS result;  END$$  CREATE PROCEDURE verify\_account\_ae(IN value VARCHAR(50))  BEGIN  SELECT IFNULL((SELECT TRUE FROM admininfo WHERE adminemail = value LIMIT 1), FALSE) AS result;  END$$  （2）密码查询  CREATE PROCEDURE get\_password\_ap(IN value VARCHAR(50))  BEGIN  SELECT adminpassword FROM admininfo WHERE adminphone LIKE value;  END$$  CREATE PROCEDURE get\_password\_ae(IN value VARCHAR(50))  BEGIN  SELECT adminpassword FROM admininfo WHERE adminemail LIKE value;  END$$  （3）上司查询  SELECT adminid, adminname, adminphone, adminemail  FROM admininfo WHERE adminid = (  SELECT padminid FROM admininfo WHERE adminid = 1);  （4）删除用户账户  UPDATE userinfo SET isactive = 0 WHERE userid = 3;  （5）查询下属  SELECT adminid, adminname, adminphone, adminemail  FROM admininfo WHERE padminid = 21;  执行效果截图：   |  | | --- | | 存在查询 | | 密码查询 | | 上司查询 | | 下属查询 |   **8. 管理员发发布消息**  管理员向用户发送消息   |  |  | | --- | --- | | （1）未回复投诉/建议 | （2）已回复投诉/建议 | | （3）发公告 | （4）回复投诉/建议 | | （5）催收费用 | （6）维修提醒 |   SQL语句：  （1）查询未回复信息  SELECT c\_aid,c\_a\_userid,c\_acontent,c\_aobject,c\_atime  FROM complaint\_and\_acviceinfo WHERE c\_a\_adminid IS NULL;  （2）查询已回复投诉建议  SELECT c\_aid,c\_a\_userid,c\_acontent,c\_aobject,c\_atime  FROM complaint\_and\_acviceinfo WHERE c\_a\_adminid IS NOT NULL;  （3）发公告  INSERT INTO newsinfo(news\_adminid,newstitle,newscontent,newstime,  newsendtime,newstype)  VALUES (12, 'test', 'test', localtime(), localtime(), 0);  ## 发单人消息  DELIMITER $$  CREATE PROCEDURE send\_news(in inadmin INT, in inuser INT,  in innews INT, in intitle TEXT,  in incontent TEXT, in inendtime TIMESTAMP)  BEGIN  INSERT INTO newsinfo(news\_adminid,newstitle,newscontent,  newstime,newsendtime,newstype, newsid)  VALUES (inadmin, intitle, incontent, localtime(),  inendtime, 2, innews);  INSERT INTO user\_newsrelation(un\_newsid, un\_userid)  VALUES (innews, inuser);  END$$  DELIMITER ;  执行效果截图：   |  | | --- | | (1) 查询未回复投诉/建议 | | (2) 查询已回复投诉/建议 |   **9. 管理员更新房屋信息**  管理员修改所有权   |  |  | | --- | --- | | （1）空房查询 | （2）空房出售 | | （3）房屋转手 |  |   SQL语句：  （1）空房查询  SELECT houseid, location, house\_adminid FROM houseinfo  WHERE housestate = 0;  （2）新房出售  DELIMITER $$  CREATE PROCEDURE updata\_new(IN inhouse INT, IN inuser INT, IN inadmin INT, in content TEXT)  BEGIN  ##修改房屋状态，添加用户  UPDATE houseinfo SET housestate = 1 WHERE houseid = inhouse;  INSERT INTO user\_houserelation(uh\_relation\_userid, uh\_relation\_houseid, gethousetime)  VALUES(inuser, inhouse, localtime());  ##记录修改  INSERT INTO updatarelation(updatacontent, updatatime, updata\_houseid, updata\_adminid)  VALUES(content, localtime(), inhouse, inadmin);  END$$  DELIMITER ;  （3）旧房转卖  DELIMITER $$  CREATE PROCEDURE updata\_old(IN inhouse INT, IN inuser INT, IN inadmin INT, in content TEXT)  BEGIN  ##清空该房屋的费用，车位  UPDATE costinfo SET paytime = localtime() WHERE paytime IS NULL;  UPDATE costinfo SET istrue = 0 WHERE cost\_houseid = inhouse;  UPDATE pcarinfo SET pcarstate = 0, buytime = NULL, pcar\_houseid = NULL WHERE pcar\_houseid = inhouse;  ##添加记录  INSERT INTO user\_houserelation(uh\_relation\_userid, uh\_relation\_houseid, gethousetime)  VALUES(inuser, inhouse, localtime());  ##记录修改  INSERT INTO updatarelation(updatacontent, updatatime, updata\_houseid, updata\_adminid)  VALUES(content, localtime(), inhouse, inadmin);  END$$  DELIMITER ;  执行效果截图：   |  | | --- | | 空房查询 |   **10. 管理员其他管理**  查看停车位状态，缴费状态   |  |  | | --- | --- | | （1）已出售停车位 | （2）未出售停车位 | | （3）未缴费 | （4）已缴费 |   SQL语句：  （1）查询未购车位  SELECT pcarid,pcarloction FROM pcarinfo WHERE pcarstate<>2;  （2）查询已购车位  SELECT pcarid,pcar\_houseid,buytime,pcarloction FROM pcarinfo where pcarstate=2;  （3）查询已缴费  SELECT costid,cost\_houseid,costmoney,coststarttime,paytime,costtype FROM costinfo,costtypeinfo WHERE cost\_costtypeid = costtypeid and paytime is NOT NULL;  （4）查询未缴费  SELECT costid,cost\_houseid,costmoney,coststarttime,costtype FROM costinfo,costtypeinfo WHERE cost\_costtypeid = costtypeid and paytime is NULL;  执行效果截图：   |  | | --- | | 查询未购车位 | | 查询已购车位 | | 查询已缴费 | | 查询未缴费 |   **11. 管理员管理维修任务**  分配维修任务   |  |  | | --- | --- | | （1）未接取任务 | （2）已开始任务 | | （3）已完成任务 | （4）修改任务状态 |   SQL语句：  （1）未接取任务查询  SELECT task\_houseid,taskcontent FROM taskinfo where taskstate = 0;  （2）已开始任务查询  SELECT task\_houseid,taskstarttime,taskcontent,ataskstate  FROM taskinfo,admin\_taskrelation where taskid=admin\_task\_taskid  AND taskstate = 1;  （3）已完成任务查询  SELECT task\_houseid,taskstarttime,taskendtime,taskcontent  FROM taskinfo,admin\_taskrelation where taskid=admin\_task\_taskid  AND taskstate = 2;  （4）接取未开始任务  DELIMITER $$  CREATE PROCEDURE receive\_task(IN intask INT, IN inadmin INT)  BEGIN  UPDATE taskinfo SET taskstate = 1 WHERE taskid = intask;  INSERT INTO admin\_taskrelation (ataskstate, taskstarttime,  admin\_task\_taskid, admin\_task\_adminid)  VALUES(0, localtime(), intask, inadmin);  END$$  DELIMITER ;  （5）维修完成  DELIMITER $$  CREATE PROCEDURE successful\_task(IN inatr INT)  BEGIN  UPDATE taskinfo SET taskstate = 2 WHERE taskid = (  SELECT admin\_task\_taskid FROM admin\_taskrelation  WHERE admin\_taskid = inatr);  UPDATE admin\_taskrelation SET ataskstate = 1, taskendtime = localtime()  WHERE admin\_taskid = inatr;  END$$  DELIMITER ;  （6）维修失败  DELIMITER $$  DELIMITER $$  CREATE PROCEDURE fail\_task(IN inatr INT)  BEGIN  UPDATE taskinfo SET taskstate = 0 WHERE taskid = (  SELECT admin\_task\_taskid FROM admin\_taskrelation  WHERE admin\_taskid = inatr);  UPDATE admin\_taskrelation SET ataskstate = 2  WHERE admin\_taskid = inatr;  END$$  DELIMITER ;  执行效果截图：   |  | | --- | | 未接取任务查询 | | 已开始任务查询 | | 已完成任务查询 | |
| 附  原  始  实  验  记  录 | **四、实验总结**  1. 在用户查看消息时，不知道什么消息属于该用户，通过添加消息-用户管理表（user\_newsrelation）解决该问题  2. 维修任务与用户无关，应与房屋关联  3. 在更新房屋信息时，考虑到新住户不能查看旧住户的缴费信息但缴费信息又不能删除，通过在缴费信息表（costinfo）中添加费用是否为当前住户(0:否, 1:是)（istrue）字段解决  4. 考虑到实际功能需求，对管理员模块做了大幅度修改  5. 有些功能在写程序时还将修改  6. 数据存在一些问题 |