

数据库工程作业

工程作业报告

1. 项目信息（10分）

学号	2213912	姓名	罗劲	专业	密码科学与技术
项目名称	公益网吧管理系统				
必备环境	MySql 8.0.36 /Java with jdk1.7				
系统主要功能简介（4分）	实现了一个公益网吧管理系统，首先里面具有管理者的账号信息，用于登录系统；管理者可以对电脑信息实现增加、删除的操作；也可以对访客进行信息的登记、删除和修改的操作；还有安排访客与对应电脑的匹配，并登记到安排表中，可以通过访问安排表知道哪个顾客正在用哪一台电脑。				
系统主要页面截图（6分）	<div></div> <div></div> <div></div> <div></div>				

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			
	高级	1. Sql			
	语言	2. Java			
连接串 分析 (6分)	序号	名称	功能说明	取值	
	1	驱动	加载驱动	Jdbc	

	2	连接地址	连接数据库地址	localhost
	3	端口号	连接时所用的端口号	3306
	4	用户名	数据库的用户名	root
	5	密码	数据库的密码	*****
	6	数据库名	要连接的数据库名字	cybercafe
	连接串代码 (截屏) (2分)			
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	administrator	account	无	无
	2	category	cid	无	无
	3	computer	comid	cid	category(cid)
	4	customer	customerid	无	无
	5	vipc	customerid	无	无
	6	arrange	comid、customerid	comid、customerid	computer(comid)、customer(customerid)
关系图(4)					
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）	实现了对用户信息、用户与电脑之间的安排信息的批量删除操作。	
涉及的表 （2分）	arrange 表和 computer 表	
表连接涉及字段 （1分）	computer.comid=arrange.comid	
删除条件 字段描述 （1分）	字段	规则
	computer.comid	删除时选中了 computer.comid，就直接删除 computer.comid 对应的 computer 表中的行信息
	arrange.comid	删除对应的 computer 时，若有人正在用该 computer，即 arrange 表中有该 computer 的信息，则删除 arrange 表中相应的信息。

<p>代码 (4 分)</p>	<pre> boolean delComputer() throws SQLException { this.conn.setAutoCommit(false); boolean flag = false; String sql1 = "delete from arrange where comid=?"; String sql2 = "delete from computer where comid=?"; PreparedStatement Pst1 = null; PreparedStatement Pst2 = null; try { Pst1 = this.conn.prepareStatement(sql1); Pst2 = this.conn.prepareStatement(sql2); for(int i = 0; i < ComputerInfo.inforNum; ++i) { if (ComputerInfo.comList[i].isSel.isSelected()) { Pst1.setString(parameterIndex: 1, ComputerInfo.comList[i].lComid.getText()); Pst2.setString(parameterIndex: 1, ComputerInfo.comList[i].lComid.getText()); System.out.println(ComputerInfo.comList[i].lComid.getText()); Pst1.addBatch(); Pst2.addBatch(); } } Pst1.executeBatch(); Pst2.executeBatch(); this.conn.commit(); flag = true; } catch (SQLException var13) { SQLException e = var13; SQLException e = var13; try { e.printStackTrace(); this.conn.rollback(); System.out.println("失败"); } catch (SQLException var12) { SQLException e1 = var12; e1.printStackTrace(); } } finally { Pst1.close(); Pst2.close(); } this.conn.setAutoCommit(true); return flag; } </pre>
<p>程序演示 (4 分)</p>	<p>打开电脑信息窗口，删除某一的电脑信息，则在用户和电脑之间的安排信息也将被删除</p>
<p>备注</p>	

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)	向用户和电脑安排信息的表中插入信息	
触发器描述 (2 分)	向用户和电脑安排信息的表中插入信息之前, 先对外键 comid 和 customerid 进行相应的检查, 若外键 comid 在 computer 表中不存在, 触发器报错; 若外键 customerid 在 customer 表中不存在, 则触发器也报错; 若插入的电脑还在被别人使用, 即电脑还在 runtime 之内, 也进行报错处理。以上报错均不会正常执行, 并且会提示操作者插入失败。如果信息都存在且正常则插入可以正常进行。	
涉及的表 (1 分)	arrange, computer, customer	
输入数据 (2 分)	字段	规则
	arrange.comid	comid 的值需要在 computetr 表中存在, 否则异常
	arrange.customerid	customerid 若在对应表中不存在, 就插入失败
	arrange.stime	若某一电脑还在 runtime 之内, 则不能加入安排

<p>插入操作 源码 (3 分)</p>	<pre> boolean addArranges(String comid, String customerid, String stime) { String sql = "insert into arrange values(?,?,?)"; PreparedStatement pstmt = null; int row = 0; SQLException e; try { pstmt = this.conn.prepareStatement(sql); pstmt.setString(parameterIndex: 1, comid); pstmt.setString(parameterIndex: 2, customerid); pstmt.setString(parameterIndex: 3, stime); row = pstmt.executeUpdate(); addArrange.getInstance().lTips.setText("添加上机成功"); } catch (SQLException var9) { e = var9; addArrange.getInstance().lTips.setText("错误: 不存在该电脑或顾客上机冲突"); e.printStackTrace(); } try { pstmt.close(); } catch (SQLException var8) { e = var8; e.printStackTrace(); } return row == 1; } </pre>
<p>触发器源 码 (3 分)</p>	<pre> delimiter ;; CREATE TRIGGER `addarr` BEFORE INSERT ON `arrange` FOR EACH ROW BEGIN if NEW.comid not in (select comid from computer) or NEW.customerid not in (select customerid from customer) or EXISTS (select * from (select arrange.comid,customerid,stime,(DATE_ADD(stime,INTERVAL runtime MINUTE))as endtime from arrange left join computer on arrange.comid=computer.comid) t1 where new.stime BETWEEN t1.stime and t1.endtime)#新加入的安排在旧电脑中有重复 THEN SIGNAL SQLSTATE 'HY000'; #一种异常 end if; </pre>
<p>程序演示 (4 分)</p>	<p>不违背触发器能够执行插入操作: 若插入的安排信息中的电脑存在且用户信息存在与否则都没事, 且安排时间不冲突, 则插入成功。</p>
<p>程序演示 (4 分)</p>	<p>违背触发器要求, 不能够执行插入操作, 系统报错: 若插入的安排信息中电脑不存在或者安排时间与其他电脑安排时间冲突, 则插入失败。</p>
<p>备注</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）	实现对电脑（computer）信息的插入和更新操作。	
存储过程功能描述（1分）	<p>插入前进行分析增加或修改的电脑信息内容是否符合要求：</p> <p>如果插入的电脑类型存在，则进行判断电脑是否存在，电脑存在则进行对电脑信息的更新，电脑信息不存在则插入新的电脑；</p> <p>如果插入的电脑类型不存在，则向类型表中插入新的类型，并且判断电脑是否存在于表中，若存在则更新原电脑信息，若不存在则向表中插入新的信息。</p> <p>以上操作不会影响到 arrange 表中的信息。</p>	
涉及的关系表（2分）	computer、category	
表连接涉及字段（1）	computer.cid=category.cid	
更改字段（2分）	字段	规则
	computer.cid	若插入电脑和原电脑编号和名称一致，更新电脑 cid；否则插入新电脑
	computer.runtime	若插入电脑和原电脑编号和名称一致，更新电脑 runtime；否则插入新电脑
	computer.buydate	若插入电脑和原电脑编号和名称一致，更新电脑不同 buydate；否则插入新电脑
	computer.comid	若检测到新值，插入电脑
	computer.comname	若检测到新值，插入电脑
	category.cid	若插入的 cid 不存在，则 category 表中插入 cid 和 cname
	category.cname	若插入的 cname 不存在，则 category 表中插入 cid 和 cname

	<pre>BEGIN UPDATE computer set cid=mycid,buydate=mybuydate,runtime=myruntime where comid=mycomid; set rtn=1; END; ELSE BEGIN set rtn=2; END; end if; END; ELSE BEGIN INSERT into computer values (mycomid,mycomname,mycid,mybuydate,myruntime); set rtn=3; END; end if; END; end if; END ;; delimiter ;</pre>
存储过程执行源码 (1分)	<pre>int addComputer(String comid, String comname, String catename, String cateid, String buydate, String comprice) { System.out.println(comid); System.out.println(comname); String sql0 = "set @comid='" + comid + "'"; String sql1 = "set @comname='" + comname + "'"; String sql2 = "set @cateid='" + cateid + "'"; String sql3 = "set @buydate='" + buydate + "'"; String sql4 = "set @comprice='" + comprice + "'"; String sql5 = "set @catename='" + catename + "'"; String sql6 = "call addcom(@comid,@comname,@cateid,@buydate,@comprice,@catename,@rtn)"; String sql7 = "select @rtn"; int rtn = 0; try { Statement stmt = this.conn.createStatement(); stmt.executeQuery(sql0); stmt.executeQuery(sql1); stmt.executeQuery(sql2); stmt.executeQuery(sql3); stmt.executeQuery(sql4); stmt.executeQuery(sql5); stmt.executeQuery(sql6); ResultSet rst = stmt.executeQuery(sql7); if (rst.next()) { rtn = rst.getInt(columnIndex: 1); } stmt.close(); } catch (SQLException var18) { SQLException e = var18; e.printStackTrace(); } return rtn; }</pre>
程序演示 (2分)	<p>不违背存储过程，能够执行更新操作：</p> <p>选择管理电脑界面，点击修改或添加按钮，输入电脑编号与电脑名称、电脑类型编号和电脑类型对应或全新电脑类型的电脑，更新内容成功。</p>
程序演示 (2分)	<p>违背存储过程，系统报错：</p> <p>选择管理电脑界面，点击修改或添加按钮，输入电脑编号与电脑名称相悖，或电脑类型编号与电脑类型相悖的电脑，更新失败，系统报错。</p>
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）	利用电脑名称的部分信息，获得安排记录
视图功能描述 （1 分）	实现了对安排的查询，可以通过查询电脑名称的部分信息，就得出相应电脑的所有安排信息
涉及的关系表 （2 分）	arrange, computer, category
表连接字段（1 分）	arrange.comid=computer.comid computer.cid=category.cid
创建视图代码 （3 分）	<pre> DROP VIEW IF EXISTS `arr`; CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `arr` AS (select comname,cname,customerid,stime,arrange.comid from arrange left join computer on arrange.comid=computer.comid left join category on computer.cid=category.cid) ; ----- -- View structure for index_com ----- DROP VIEW IF EXISTS `index_com`; CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `index_com` AS (select comid,comname,computer.cid,cname,buydate,runtime from computer left join category on computer.cid=category.cid) ; </pre>

<p>查询代码 (3 分)</p>	<pre>int searchArrange(String key) throws Exception { if (key.equals("")) { return this.getArrange(); } else { String sql = "select * from arr where comname like '%" + key + "%'"; Statement stmt = this.conn.createStatement(); ResultSet rst = stmt.executeQuery(sql); System.out.println(sql); int num = 0; while(rst.next()) { String comname = rst.getString(columnIndex: 1); String cname = rst.getString(columnIndex: 2); String customerid = rst.getString(columnIndex: 3); String stime = rst.getString(columnIndex: 4); String comid = rst.getString(columnIndex: 5); ArrangeInfo.setArrange(num++, comname, cname, customerid, stime, } stmt.close(); ArrangeInfo.inforNum = num; return num; } } }</pre>
<p>程序演示 (4 分)</p>	<p>打开管理电脑，在 TextField 中输入相应值，会在下方显示出相应的查询结果。</p>
<p>备注</p>	