

# 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	2112066	姓名	于成俊	专业	密码科学与技术
项目名称	学校管理系统				
必备环境	1. MySQL 8.0.32 for Win64 on X86_64 2. Java version “19” 2022-09-20 3. Eclipse IDE for Java Developers - 2022-09				
系统主要功能简介（4 分）	系统实现了三种用户登录，分别为教师，本科生，管理者。（若未选择身份，系统会弹窗）。教师登录进入教师界面，可以进行开设课程、查看已开设课程、查看某个课程所有学生的成绩等操作；本科生登录进入本科生界面，可以进行选课、退课、查询成绩、参加社团等操作。管理者进入管理者界面，可以进行增加社团、删除社团、查看学院、评优学生、查看所有课程、学生、老师、社团，以及查看参加某个社团的所有学生等操作。				
系统主要页面截图（6 分）	截 3-4 个页面即可 1. 登录界面 				

## 2. 教师界面

学校管理系统

姓名: 穆里尼奥

课程名称

课程号: 396

上课时间

开设课程

查询已开设课程

查询学生成绩

职工号: 100086

课程号: 396

上课时间

职称: 教授

课程名称

课程代号

上课时间

密码学基础

367

周五上午八点

数据库系统

396

周五上午九点

爱情社会学

1314

周三早上八点

## 3. 本科生界面

学校管理系统

姓名 李二麻子

学号 2113666

专业 心理学

称谓 普通学生

选课

退课

查询成绩

参加社团

课程名称

成绩

密码学基础

89.0

数据库系统

67.0

4. 管理者界面

学校管理系统

增加社团

删除社团

查看学院

评优学生

查看所有课程

查看所有学生

查看所有老师

查看所有社团

查看社团学生

姓名	职工号	职称	所属学院	入职时间
穆里尼奥	100086	教授	网络空间安全学院	2020-09-08
瓜迪奥拉	191909	教授	计算机学院	2003-09-08
全拓禹	202367	教授	周恩来政府管理学院	2008-09-23

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			
		2. MYSQL WORKBENCH			
	高级 语言	1. Java			
		2. SQL			
连接串 分析 (6 分)		序 号	名称	功能说明	取值
		1	<code>DriverManager. registerDriver (new com.mysql. jdbc.Dri ver());</code>	告诉 java 要连接什么数据库，这里用到了 JDBC 中的 DriverManager 类	
		2	<div>1. <code>Url</code> 2. <code>User</code> 3. <code>password</code></div>	用于获取连接所需要的参数，即相应的数据库、用户名和密码	<div>1. <code>"jdbc:mysql://localhost:3306/schoolmanagement"</code></div>

				system" 2. "root" 3. "3973433"
	3	Connection conn	获取连接的数据库，这里依旧用到了 JDBC 中的 DriverManager 类	DriverManager .getConnection(url, user, password);
	4	Statement stmt	获取数据库操作对象，从而进行增删改查等操作	conn.createStatement();
连接串代码 (截屏) (2 分)	<pre>//定义下面需要的对象 Connection conn=null; Statement stmt=null; //因为下面要关闭 所以创建变量不能在try中 try {     //第一步 注册驱动     DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());     //第二步 获取连接     String url="jdbc:mysql://localhost:3306/school management system";//数据库     String user="root"; //用户名     String password="3973433"; //密码     conn= DriverManager.getConnection(url, user, password);     System.out.println("数据库连接对象"+conn);     //第三步获取数据库操作对象     stmt =conn.createStatement();  } catch (SQLException e) {     // TODO Auto-generated catch block     e.printStackTrace(); }</pre>			
备注	图片中的密码为错误密码（为保障安全）			

### 3. 数据库设计（14 分）

说明	<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>				
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	用户（账号，密码）	账号	无	无
	2	学院（学院名称，成立时间）	学院名称	无	无
	3	教师（姓名，职称，入职时间，职工号，所属学院）	职工号	所属学院	学院（学院名称）
	4	专业（名称，培养目标，开设学院）	名称	开设学院	学院（学院名称）

	5	管理者（账号，密码）	账号	账号	用户（账号）
	6	学生（学号，姓名，入学时间，性别，专业，称谓）	学号	（学号，专业）	用户（账号） 专业（名称）
	7	课程（课程名称，课号，开课老师，上课时间）	课程号	开课老师	教师（职工号）
	8	社团（社团名称，成立时间）	社团名称	无	无
	9	参加（社团名称，学生学号）	（学生学号、社团名称）	（学生学号、社团名称）	学生（学号） 社团（社团名称）
	10	成绩（课号，学生学号，成绩）	（课号、学生学号）	（课号、学生学号）	课程（课号） 学生（学号）
关系图 (4)	<pre> erDiagram     USER   --o{ TEACHER : 参照1     USER   --o{ STUDENT : 参照2     USER   --o{ CLUB : 参照3     TEACHER   --o{ CLUB : 教授     TEACHER   --o{ COLLEGE : 归属     STUDENT   --o{ CLUB : 参加     STUDENT   --o{ COURSE : 选课     STUDENT   --o{ MAJOR : 所属     CLUB   --o{ COLLEGE : 归属     CLUB   --o{ STUDENT : 参加     COLLEGE   --o{ MAJOR : 归属     COURSE   --o{ MAJOR : 归属     </pre>				
备注	该数据库设定学生的账号为学号，教师的账号为职工号，所以学生表的学号和教师表的职工号参照于用户表的账号。				

#### 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）	管理者输入社团名称，点击删除，若该社团现有的学生人数小于 2 人，则删除成功，否则不成功。	
涉及的表 （2 分）	社团（社团名称，成立时间） 参加（社团名称，学生学号）	
表连接涉及字段 （1 分）	社团. 社团名称=参加. 社团名称	
删除条件 字段描述 （1 分）	字段	规则
	社团. 社团名称	管理者输入一个社团名称，若符合要求则删除这个社团。
	参加. 社团名称	随社团. 社团名称删除（设置了级联更新）
	参加. 学生学号	随社团. 社团名称删除（设置了级联更新）
	社团. 成立时间	随社团. 社团名称删除
代码 （4 分）	<pre> public void delete_nameclub(String name, Connection conn) {     PreparedStatement stmt = null;     ResultSet rs = null;     try {         //开始事务         conn.setAutoCommit(false);         // 执行SQL查询语句         String sql = "SELECT COUNT(*) FROM 参加 WHERE 参加的社团名称=?";         stmt = conn.prepareStatement(sql);         stmt.setString(1, name); //设置参数         rs = stmt.executeQuery();         int count=0;         if(rs.next()) {             count=rs.getInt(1);         }         //判断学生数量是否大于2         if (count &gt; 2) {             JOptionPane.showMessageDialog(null,"删除不成功，该社团的学生数量大于2", "错误信息", JOptionPane.ERROR_MESSAGE); // 弹出对应的弹窗             // 事务回滚             conn.rollback();         } else {             //执行删除操作             String deleteClubQuery = "DELETE FROM 社团 WHERE 社团名称 = '" + name + "'";             stmt.executeUpdate(deleteClubQuery);             //提交事务             conn.commit();             JOptionPane.showMessageDialog(null,"删除成功", "提示信息", JOptionPane.INFORMATION_MESSAGE); // 弹出对应的弹窗         }         stmt.close();     } catch (Exception e) {         //处理异常         e.printStackTrace();         try {             // 事务回滚             if (conn != null) {                 conn.rollback();             }         } catch (SQLException ex) {             ex.printStackTrace();         }     } } </pre>	
程序演示 （4 分）		

1. 执行操作前的“参加”表

Result Grid		Filter Rows:
参加的社团名称	学生学号	
篮球社团	2113666	
篮球社团	2113737	
篮球社团	2119780	
足球社团	2113737	
NULL	NULL	

2. 删除足球社团（由于人数小于 2，删除成功）

Information

i

人数小于:

删除

删除名称:

足球社团

删除

确定

Information

提示信息

i

删除成功

确定

3. 删除篮球社团（由于人数大于 2，删除不成功）

Information

i

人数小于:

删除

删除名称:

篮球社团

删除

确定



#### 4. 删除之后的“参加”表

参加的社团名称	学生学号
篮球社团	2113666
篮球社团	2113737
篮球社团	2119780
NULL	NULL

备注

由于参加表中的社团名称参照于社团表中的社团名称，所以我又设置了级联更新，来使删除操作成功。



## 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）	教师开设课程时，如果该教师开设多于两门课程或课程时间冲突，则开课不成功。	
触发器描述 （2 分）	课程表设置了触发器，当教师开设课程时，即向课程表插入新课程。触发器功能为在插入新课程之前，要检查新课程时间是否与原课程表中的课程时间冲突和原本该教师开设的课程数量是否大于 1。若满足条件，则会发出开课不成功的信息，若不满足条件，则开课成功。	
涉及的表 （1 分）	课程（课程名称，课号，开课老师，上课时间）	
输入数据 （2 分）	字段	规则
	课程. 上课时间	不能与原课程表中的上课时间相同
	课程. 开课老师	如果新开设课程的老师，在原来的课程表中出现次数小于 2，则插入成功。
插入操作 源码 （3 分）	<pre> public void add_course(int id,int tid, String name, String time, Connection conn) {      // 创建 PreparedStatement 对象     PreparedStatement pstmt = null;     try {         // 为 SQL 语句中的参数赋值         String sql = "INSERT INTO 课程 (课程名称, 课程代号, 开课老师, 上课时间) VALUES (?, ?, ?, ?)";         pstmt = conn.prepareStatement(sql);         pstmt.setString(1, name); // 课程名称         pstmt.setInt(2, id); // 课程代号         pstmt.setInt(3, tid); // 开课老师         pstmt.setString(4, time); // 上课时间          // 执行 SQL 语句，并获得受影响的行数         int rowsAffected = pstmt.executeUpdate();          // 输出受影响的行数         System.out.println(rowsAffected + " row(s) affected");     } catch (SQLException e) {         String errorMsg = e.getMessage(); // 获取异常信息         JOptionPane.showMessageDialog(null, errorMsg, "错误信息", JOptionPane.ERROR_MESSAGE); // 弹出对应的弹窗     } finally {         try {             if (pstmt != null) {                 pstmt.close();             }         } catch (SQLException e) {             e.printStackTrace();         }     } } </pre>	

触发器源码  
(3分)

```
CREATE DEFINER='root'@'localhost' TRIGGER `开设课程` BEFORE INSERT ON `课程` FOR EACH ROW BEGIN
    declare msg1 varchar(90);
    declare msg2 varchar(90);
    IF (SELECT COUNT(*) FROM 课程 WHERE 开课老师 = NEW.开课老师)>1 THEN
        set msg1 = "同一个老师最多只能开设两个课程!";
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg1;
    end if;
    IF(SELECT COUNT(*) FROM 课程 WHERE NEW.上课时间 = 课程.上课时间)>0 THEN
        set msg2 = "课程时间冲突";
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg2;
    end if;
END
```

程序演示  
(4分)

说明：不违背触发器能够执行插入操作。

学校管理系统

姓名:	全拓禹	课程名称	计算机原理	开设课程
职工号:	202367	课程号:	121211	查询已开设课程
职称:	教授	上课时间	周天晚上十点	查询学生成绩

提示信息  
i 开设成功  
确定

程序演示  
(4分)

说明：违背触发器要求，不能够执行插入操作，系统报错。  
1.

	<div><div>学校管理系统</div><div><table><tr><td>姓名:</td><td>全拓禹</td><td>课程名称</td><td>计算机原理</td><td>开设课程</td></tr><tr><td>职工号:</td><td>202367</td><td>课程号:</td><td>121211</td><td>查询已开设课程</td></tr><tr><td>职称:</td><td>教授</td><td>上课时间</td><td>周五上午八点</td><td>查询学生成绩</td></tr></table></div></div> <div><div>错误信息</div><div><div>X</div>课程时间冲突</div><div>确定</div></div>	姓名:	全拓禹	课程名称	计算机原理	开设课程	职工号:	202367	课程号:	121211	查询已开设课程	职称:	教授	上课时间	周五上午八点	查询学生成绩
姓名:	全拓禹	课程名称	计算机原理	开设课程												
职工号:	202367	课程号:	121211	查询已开设课程												
职称:	教授	上课时间	周五上午八点	查询学生成绩												
2.																
	<div><div>学校管理系统</div><div><table><tr><td>姓名:</td><td>穆里尼奥</td><td>课程名称</td><td>计算机原理</td><td>开设课程</td></tr><tr><td>职工号:</td><td>100086</td><td>课程号:</td><td>8666</td><td>查询已开设课程</td></tr><tr><td>职称:</td><td>教授</td><td>上课时间</td><td>周三晚上九点</td><td>查询学生成绩</td></tr></table></div></div> <div><div>错误信息</div><div><div>X</div>同一个老师最多只能开设两个课程!</div><div>确定</div></div>	姓名:	穆里尼奥	课程名称	计算机原理	开设课程	职工号:	100086	课程号:	8666	查询已开设课程	职称:	教授	上课时间	周三晚上九点	查询学生成绩
姓名:	穆里尼奥	课程名称	计算机原理	开设课程												
职工号:	100086	课程号:	8666	查询已开设课程												
职称:	教授	上课时间	周三晚上九点	查询学生成绩												
备注	违背触发器的条件不同，弹出的弹窗也不一样。															

## 6. 存储过程控制下的更新操作（18分）

说明	(1分) 简要说明该操作所要完成的功能; (1分) 简要说明该存储过程所要完成的功能; (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) (1分) 表连接涉及字段描述(描述方式为“表 1. 属性=表 2. 属性”) (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; (6分) 实现该操作的关键代码(高级语言、SQL),截图即可; (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。	
功能描述(1分)	管理员可以将所有课程的平均成绩超过 90 的学生评为优秀学生	
存储过程功能描述(1分)	根据传入的学生学号,在课程表中,用 loop 循环,检查该学生是否选课且是否所有课程成绩的平均值大于 90,若满足条件,则将该学生的称谓设为“优秀学生”,否则发出“不符合更新条件!”。若该学生之前已是优秀学生,则发出“该学生已经是优秀学生!”的信息。	
涉及的关系表(2分)	学生(学号,姓名,入学时间,性别,专业,称谓) 成绩(课号,学生学号,成绩)	
表连接涉及字段(1)	成绩. 学生学号=学生. 学号	
更改字段(2分)	字段	规则
	学生. 称谓	若满足条件。将学生. 称谓设置为“优秀学生”
更新代码(3分)	<pre> IF course_count &gt; 0 AND grade_sum/course_count &gt; 90 THEN   UPDATE 本科生 SET 称谓 = '优秀学生' WHERE 学号 = student_id;   SELECT CONCAT(student_name, ' 已被更新为优秀学生') AS message;         </pre>	
创建存储过程源码(3分)		

```

CREATE DEFINER='root'@'localhost' PROCEDURE `评优`(IN student_id INT)
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE msg VARCHAR(45);
    DECLARE course_code INT;
    DECLARE grade float;
    DECLARE grade_sum float DEFAULT 0;
    DECLARE course_count INT DEFAULT 0;
    DECLARE student_name char(10);
    DECLARE student_title VARCHAR(45);
    DECLARE cur CURSOR FOR SELECT 课号, 成绩 FROM 成绩 WHERE 学生学号 = student_id;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    SELECT 姓名 INTO student_name FROM 本科生 WHERE 学号 = student_id;
    SELECT 称谓 INTO student_title FROM 本科生 WHERE 学号 = student_id;
    IF student_title = '优秀学生' THEN
        SELECT '该学生已经是优秀学生' AS message;

        set msg = "该学生已经是优秀学生!";
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
    ELSE
        OPEN cur;
        score_loop: LOOP
            FETCH cur INTO course_code, grade;
            IF done THEN
                LEAVE score_loop;
            END IF;



            SET grade_sum = grade_sum + grade;
            SET course_count = course_count + 1;
        END LOOP;

        IF course_count > 0 AND grade_sum/course_count > 90 THEN
            UPDATE 本科生 SET 称谓 = '优秀学生' WHERE 学号 = student_id;

            SELECT CONCAT(student_name, ' 已被更新为优秀学生') AS message;
        ELSE
            SELECT CONCAT(student_name, ' 不符合更新条件') AS message;
            set msg = "不符合更新条件!";
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
        END IF;

        CLOSE cur;
    END IF;
END

```

存储过程执行源码 (1分)	<pre>public void apprais_student(int id, Connection conn) {      java.sql.CallableStatement stmt=null;     try {         // 创建CallableStatement对象         stmt = conn.prepareCall("{call 评优(?)}");         // 设置输入参数         stmt.setInt(1, id); // 学生学号          // 执行Stored Procedure         stmt.execute();         JOptionPane.showMessageDialog(null, "评优成功!", "提示信息", JOptionPane.INFORMATION_MESSAGE); // 弹出对应的弹窗     } catch (Exception e) {         String errorMsg = e.getMessage(); // 获取异常信息         JOptionPane.showMessageDialog(null, errorMsg, "错误信息", JOptionPane.ERROR_MESSAGE); // 弹出对应的弹窗     } finally {         try {             if (stmt != null) {                 stmt.close();             }         } catch (SQLException e) {             // handle exception         }     } }</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)	<p>说明：违背存储过程，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）	管理者输入社团名称，可查看所有参加该社团的学生信息。
视图功能描述（1 分）	视图为动态视图，涉及参加表和学生表，根据社团名称参数，来显示参加该社团的所有学生信息。为了实现动态视图，参数为 function
涉及的关系表（2 分）	<p>参加（社团名称，学生学号）</p> <p>学生（学号，姓名，入学时间，性别，专业，称谓）</p>
表连接字段（1 分）	学生.学号=参加.学号
创建视图代码（3 分）	<p>1. 视图</p>  <pre>CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `new_view` AS select `本科生`.`学号` AS `学号`, `本科生`.`姓名` AS `姓名`, `本科生`.`性别` AS `性别`, `本科生`.`所学专业` AS `所学专业`, `本科生`.`入学时间` AS `入学时间`, `本科生`.`称谓` AS `称谓` from (`本科生` join `参加` on((`本科生`.`学号` = `参加`.`学生学号`))) where (`参加`.`参加的社团名称` = `return_clubname`())</pre> <p>2. function</p>  <pre>CREATE DEFINER=`root`@`localhost` FUNCTION `return_clubname`() RETURNS char(20) CHARSET utf8mb3 NO SQL BEGIN RETURN @clubname; END</pre> <p>3. 存储过程（为了可以给视图传递参数，我采用了存储过程来实现查询视图。）</p>



	<pre>CREATE DEFINER=`root`@`localhost` PROCEDURE `传参数`(inout clubname char(20)) BEGIN set @clubname=clubname; select*from new_view; END</pre>
查询代码 (3 分)	<pre>public void showview(String clubname, Connection conn) {     java.sql.CallableStatement stmt=null;     ResultSet rs=null;     try {         // 创建CallableStatement对象         stmt = conn.prepareCall("{call 传参数(?)})");         // 设置输入参数         stmt.setString(1, clubname); // 社团名称          // 执行Stored Procedure         rs=stmt.executeQuery();         // 处理查询结果         StringBuilder sb = new StringBuilder();         sb.append("学号").append(" ").append("姓名").append(" ").append("性别").append(" ").append("所学专业").append(" ").append(" ").append("\n");         while (rs.next()) {             int id = rs.getInt("学号");             String name=rs.getString("姓名");             String sex=rs.getString("性别");             String major=rs.getString("所学专业");             String time=rs.getString("入学时间");             String title=rs.getString("称谓");             sb.append(id).append(" ").append(name).append(" ").append(sex).append(" ").append(major).append(" ").append(title).append("\n");         }         // 将结果设置到JTextArea中         ManagerPanel.getInstance().textArea.setText(sb.toString());     } catch (Exception e) {         e.printStackTrace();     } finally {         try {             if (rs != null) {                 rs.close();             }             if (stmt != null) {                 stmt.close();             }         } catch (SQLException e) {             // handle exception         }     } }</pre>
程序演示 (4 分)	



	<div><div>学校管理系统</div><div><div><div>增加社团</div><div>删除社团</div><div>查看学院</div><div>评优学生</div></div><div><div>查看所有课程</div><div>查看所有学生</div><div>查看所有老师</div><div>查看所有社团</div><div>查看社团学生</div></div></div><table><tr><th>学号</th><th>姓名</th><th>性别</th><th>所学专业</th><th>入学时间</th></tr><tr><td>2113666</td><td>李六</td><td>男</td><td>心理学</td><td>2009-03-13</td></tr><tr><td>2113737</td><td>张三</td><td>男</td><td>可视化编程</td><td>2019-09-08</td></tr><tr><td>2119780</td><td>李湘</td><td>男</td><td>密码科学与技术</td><td>1999-09-08</td></tr></table><div><div>评优学生</div><div><div><div>i</div><div>请输入社团名称</div><div>篮球社团</div><div>查看</div></div><div>确定</div></div></div></div>	学号	姓名	性别	所学专业	入学时间	2113666	李六	男	心理学	2009-03-13	2113737	张三	男	可视化编程	2019-09-08	2119780	李湘	男	密码科学与技术	1999-09-08
学号	姓名	性别	所学专业	入学时间																	
2113666	李六	男	心理学	2009-03-13																	
2113737	张三	男	可视化编程	2019-09-08																	
2119780	李湘	男	密码科学与技术	1999-09-08																	
备注	为了可以给视图传递参数，我采用了存储过程来实现查询视图。																				