

数据库实验报告——运动会管理系统

学号:161610108 姓名:陈力以 联系方式:15601588099

专业:计算机科学与技术

Email: chenliyi@nuaa.edu.cn

摘 要: 传统数据的管理方式是通过电子表格、文档等等办公软件来处理大量的数据,这种方式存在着数据冗余,大量工作重复和繁重的缺点。通过数据库技术等信息化手段,使数据管理信息化,智能化,为决策层提供及时而有效的参考依据,从而达到提升信息管理水平的目的。在综合实验中为学校运动会设计的运动会管理系统采用 MySQL 数据库后台支持,Java 语言编写应用程序。要求对运动员,比赛项目,院系队伍等进行管理。

关键词: 数据库技术; MySQL; Java; JDBC

1 系统需求分析

本系统适用于学校运动会的报名、队伍等管理。管理员对比赛项目管理便捷,运动员报名操作简单,界面具有友好性、简单性。

针对两种层次的权限的用户:管理员、运动员。管理员和运动员都得凭借自己的账号、密码进入该系统,不同权限的用户登录该系统后可根据自己权限执行相应权限的操作。

(1) 管理员负责系统维护,具有系统的所有权限,可以对比赛项目、队伍、学生以及相关属性进行增删查改。

(2) 学生可以注册账号、报名项目、查看个人信息、查看比赛项目信息、修改密码。

1.1 系统功能模块分析

1. 用户登录模块

输入账号和密码进入该系统,如果账号或密码输入错误,会有提示并可进行重新输入。

2. 用户注册模块

输入账号和密码,若账号已存在或两次输入密码不一致则提示注册失败。

3. 学生信息模块

该模块含有学生的所有的个人基本信息,学生可以进行注册、查询等操作,管理员可以查询学生信息、添加、删除学生。

4. 项目模块

该模块含有该运动会所有的比赛项目的基本信息,学生可以查看,管理员对该模块拥有增删查改的所有权限。

5. 运动会报名管理模块

在该模块中学生可以报名项目,管理员可以查询已报名的学生信息以及其所报的项目。

6. 院系(队伍)模块

该模块含有报名参加运动会的各个院系的基本信息情况。不允许学生对其更改和删除,只允许学生进行查看,而管理员对该模块拥有增删的权限。

7. 项目分类模块

该模块含有该运动会所有的所有的运动项目的分类,不允许学生对其更改和删除,只允许学生进行查

看，而管理员对该模块拥有增删的权限。

8. 退出模块

用户登录成功后，操作进行完毕即可点击退出菜单将退出系统。

1.2 采用技术和运行环境

1. Java SE 技术

Java SE 技术实现运动会管理系统以及 Swing 组件实现可视化界面。

2. MySQL 数据库技术

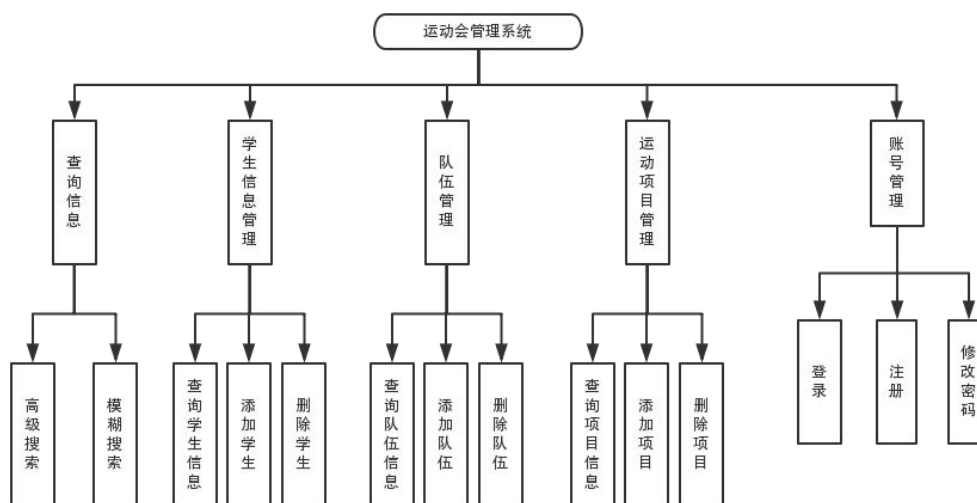
MySQL 作为后端的数据存储仓库，将数据存储起来，在前端访问数据库的时候，进行数据的访问和调用。

3. 运行环境

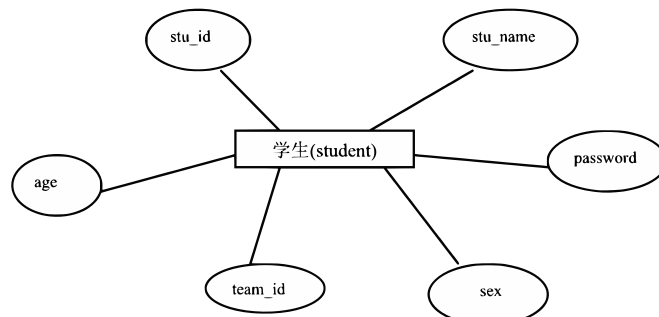
Windows 系统、Eclipse、MySQL

2 概念结构设计

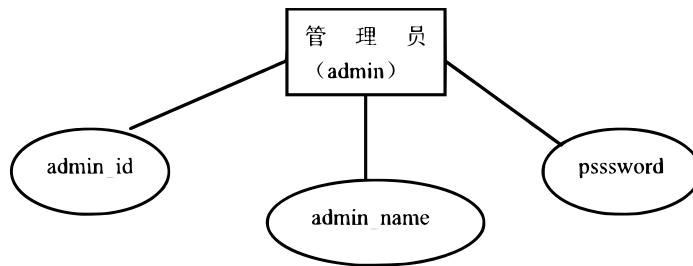
2.1 系统结构图



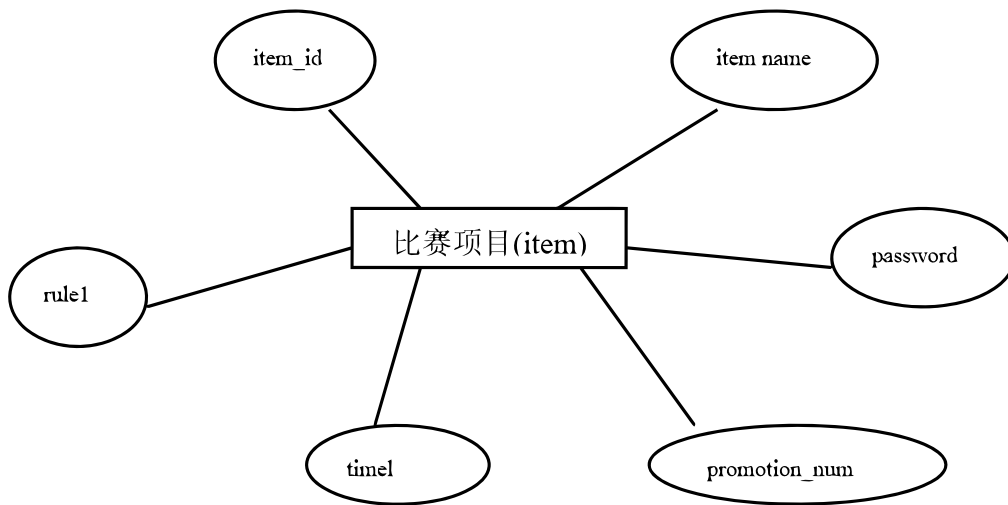
2.2 E-R 图



学生实体及其属性



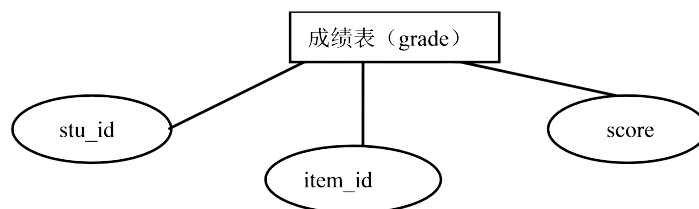
管理员实体及其属性



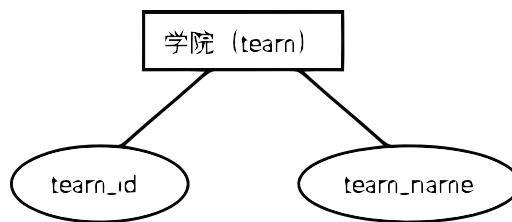
比赛项目实体及其属性



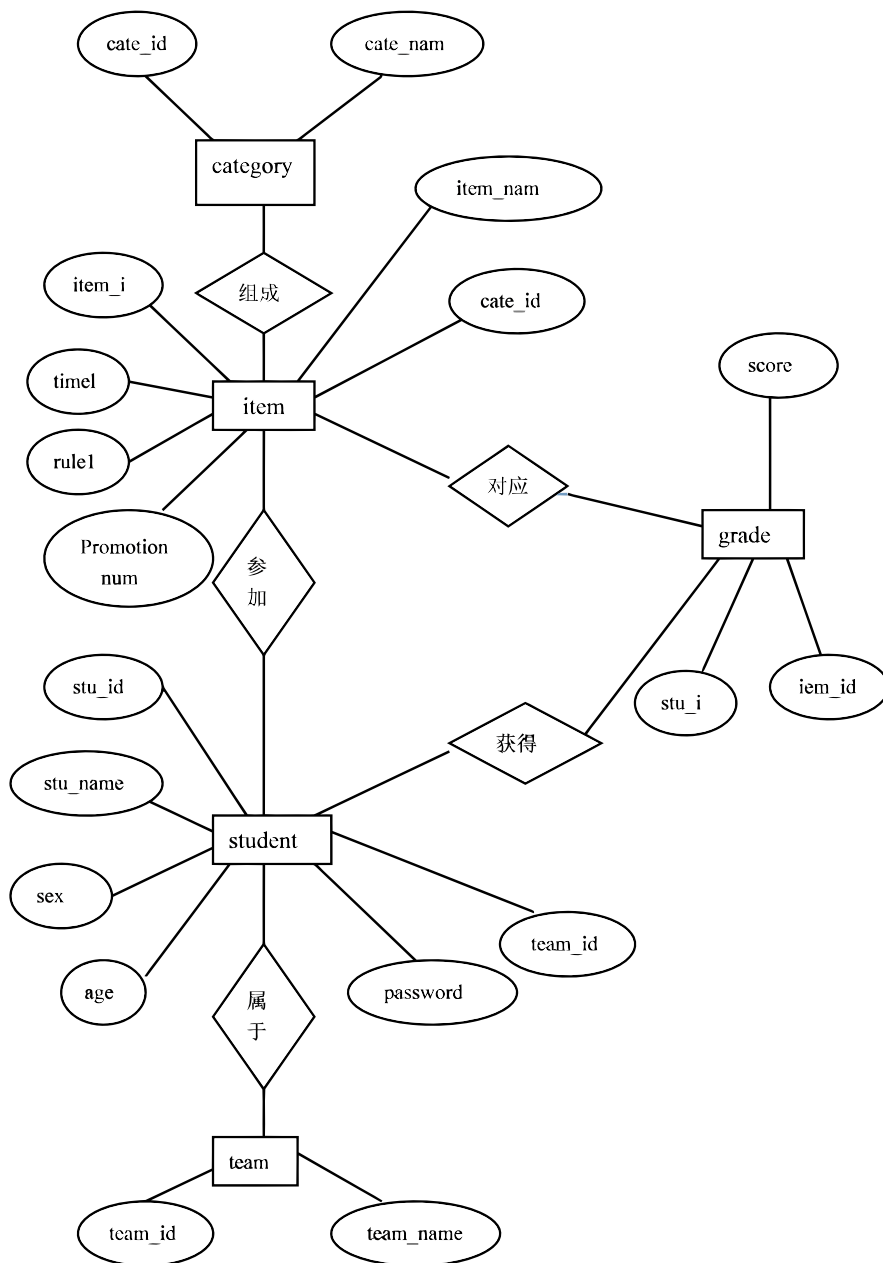
项目分类实体及其属性



成绩表实体及其属性

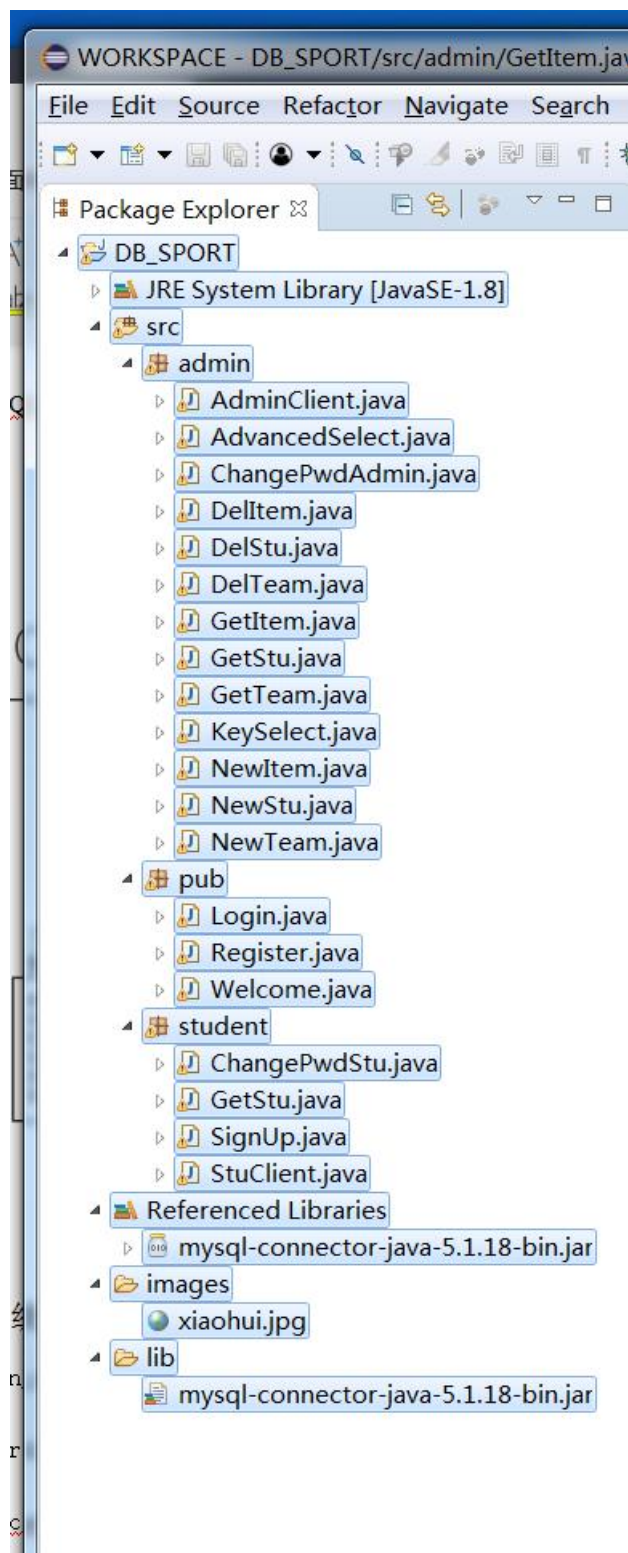


学院实体及其属性



运动会管理系统的基本 E-R 图

2.3 文件组织



3 逻辑结构设计

实下划线是主码，虚下划线是外码

管理员 (admin): (admin_id, admin_name, password);

学生 (student): (stu_id, password, stu_name, sex, age, team_id);

项目分类 (category): (cate_id, cate_name);

项目 (item): (item_id, item_name, timel, rule1, promotion_num, cate_id);

成绩 (grade): (stu_id, item_id, score);

学院 (team): (team_id, team_name);

管理员表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
admin_id	char(20)	否	主键	管理员 ID	---
admin_name	char(20)	是		管理员姓名	---
password	char(20)	是		管理员密码	---

学生表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
stu_id	char(20)	否	主键	学生 ID	---
stu_name	char(20)	是		学生姓名	---
password	char(20)	是		学生密码	学号
sex	char(2)	是		性别	---
age	smallint	是		年龄	---
team_id	char(20)	否	外键	学员号	---

项目分类表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
cate_id	char(20)	否	主键	项目分类 ID	---
cate_name	char(20)	是		项目分类名	---

项目表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
item_id	char(20)	否	主键	项目 ID	---
item_name	char(20)	是		项目名称	---
timel	char(20)	是		比赛时间	---
rule1	text	是		比赛规则	---
promotion_num	smallint	是		晋级人数	---
cate_id	char(20)	否	外键	项目分类 ID	---

成绩表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
stu_id	char(20)	否	主键	学生 ID	---
item_id	char(20)	否	主键	项目 ID	---
score	char(3)	是		得分	0

学院表:

字段名称	字段类型	是否为空	约束	字段说明	默认值
team_id	char(20)	否	主键	学院 ID	---
team_name	char(20)	是		项目号	---

4 物理结构设计

创建数据库:

```
create database sportmanager;
```

创建表:

```
create table admin
```

```
(admin_id char(20) primary key,  
  admin_name char(20),  
  password char(20)  
);
```

```
create table team
```

```
(team_id char(20) primary key,  
  team_name char(20) not null  
);
```

```
create table student
```

```
(  
  stu_id char(20) primary key,  
  password char(10),  
  stu_name char(20),  
  sex char(2) not null,  
  age smallint(2),  
  team_id char(20),  
  foreign key(team_id) references team(team_id) on update cascade on delete cascade  
);
```

```
create table category
```

```
(cate_id char(20) primary key,  
  cate_name char(20) unique  
);
```

```
create table item
```

```
(item_id char(20) primary key,  
  item_name char(20),  
  timel char(20),  
  rule1 text,  
  promotion_num smallint check(promotion_num>=0),  
  cate_id char(20) not null,  
  foreign key(cate_id) references category(cate_id) on update cascade on delete cascade
```

);

```
create table grade
(stu_id char(20) not null,
 item_id char(20) not null,
 score char(3) not null default 0,
 primary key(stu_id,item_id),
 foreign key(stu_id) references student(stu_id) on update cascade on delete cascade,
 foreign key(item_id) references item(item_id) on update cascade on delete cascade
);
```

创建索引:

```
create index team_index on team(team_id);
create index item_index on item(item_id);
create index stu_index on item(stu_id);
```

5 系统设计

5.1 登录界面



按照用户类别选择不同的数据库进行查找


```

//连接数据库
this.initialConnection();
if(type==0)
{
    try
    {
        String sql = "select * from student where stu_id= '"+id+"' and password = '"+password+"'";
        rs = stmt.executeQuery(sql);
        if(!rs.next())
        {
            JOptionPane.showMessageDialog(this, "用户名或密码错误", "错误", JOptionPane.ERROR_MESSAGE);
            this.closeConn();
            return ;
        }
        //启动学生客户端
        else
        {
            new StuClient(host,id);
            this.dispose();
        }
        this.closeConn();
    }
}

if(type==1)
{
    try
    {
        String sql = "select * from admin where admin_id= '"+id+"' and password = '"+password+"'";
        rs = stmt.executeQuery(sql);
        if(!rs.next())
        {
            JOptionPane.showMessageDialog(this, "用户名或密码错误", "错误", JOptionPane.ERROR_MESSAGE);
            this.closeConn();
            return ;
        }
        //启动管理员客户端
        else
        {
            new AdminClient(host,id);
            this.dispose();
        }
    }
}

```

5.2 注册界面

The registration window is titled "注册" (Registration). It contains the following fields and controls:

- 学号 (Student ID): Text input field.
- 密码 (Password): Text input field.
- 确认密码 (Confirm Password): Text input field.
- 姓名 (Name): Text input field.
- 性别 (Gender): Dropdown menu with "男" (Male) selected.
- 年龄 (Age): Text input field.
- 队伍 (Team): Dropdown menu with "航空宇航学院" (School of Aeronautics and Astronautics) selected.
- Buttons: "确认" (Confirm) and "重置" (Reset) at the bottom.

先检测是否有该学号，然后再插入数据库

```

{
    String sql = "select * from student where stu_id = '"+stu_id+"'";
    rs = stmt.executeQuery(sql);
    if(rs.next())
    {
        JOptionPane.showMessageDialog(this, "该学号已注册", "错误", JOptionPane.ERROR_MESSAGE);
        this.closeConn();
        return ;
    }
    String sex = (String)jcb[0].getSelectedItem();
    String team_name = (String)jcb[1].getSelectedItem();
    String team_id = map_team.get(team_name);
    String sql2 = "insert into student values('"+stu_id+"','"+password+"','"+stu_name+"','"+sex+"','"+age+"','"+team_id+"')";
    int i = stmt.executeUpdate(sql2);
    if(i==1)
    {
        JOptionPane.showMessageDialog(this, "注册成功", "提示", JOptionPane.INFORMATION_MESSAGE);
        new Login();
        this.dispose();
    }
}

catch(SQLException ea)

```

5.3 学生客户端进入界面



插入了一张图片

```

import javax.swing.*;

public class Welcome extends JLabel
{
    ImageIcon icon = new ImageIcon("images/xiaohui.jpg");

    public Welcome()
    {
        // icon.setImage(icon.getImage().getScaledInstance(2000,1000, 0));
        icon.setImage(icon.getImage().getScaledInstance(1000,1000, 0));
        this.initialFrame();
    }

    public void initialFrame()
    {
        this.setIcon(icon);
        this.setBounds(0,0,100,100);
        this.setHorizontalAlignment(JLabel.CENTER);
        this.setVerticalAlignment(JLabel.CENTER);
    }
}

```

5.4 学生客户端信息查询



显示基本信息和报名信息

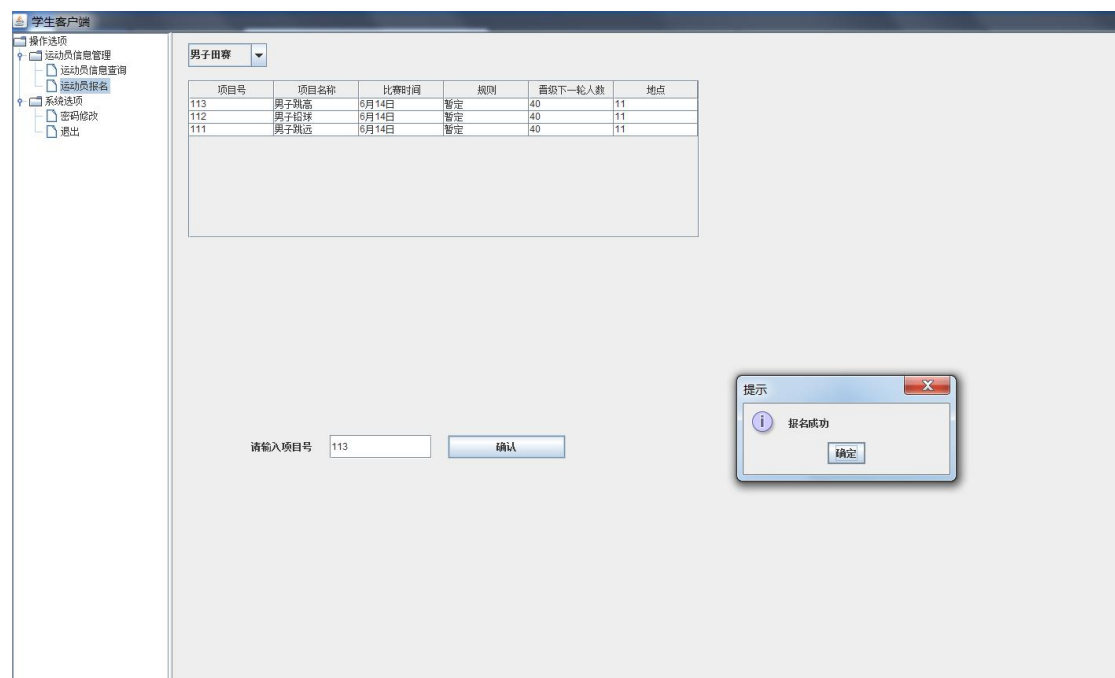
```
//显示基本信息
String sql = "select * from student where stu_id = '"+stu_id+"'";
rs = stmt.executeQuery(sql);
if(rs.next())
{
    String stu_name = rs.getString(3).trim();
    String stu_gender = rs.getString(4).trim();
    String stu_age = rs.getString(5).trim();
    String team_id = rs.getString(6).trim();

    jlArray2[0] = new JLabel(stu_id);
    jlArray2[1] = new JLabel(stu_name);
    jlArray2[2] = new JLabel(stu_gender);
    jlArray2[3] = new JLabel(stu_age);
    jlArray2[4] = new JLabel(map_team.get(team_id));
    for(int i = 0;i<5;i++)
    {
        jlArray2[i].setBounds(150,70+i*50,150,30);
        this.add(jlArray2[i]);
    }
}
```

```
String sql2 = "select item.item_id,item_name,timel,rule1,cate_id,score " +
              "from item, grade " +
              "where item.item_id = grade.item_id and grade.stu_id= '"+stu_id+"'";
rs = stmt.executeQuery(sql2);
while(rs.next())
{
    String item_id = rs.getString(1);
    String item_name = rs.getString(2);
    String timel = rs.getString(3);
    String rule1 = rs.getString(4);
    String cate_id=rs.getString(5);
    String score = rs.getString(6);
    //查询报名人数
    String sql3 = "select count(*) " +
                  "from grade " +
                  "where item_id='"+item_id+"'";

    Statement stmt1;
    stmt1 = conn.createStatement();
    ResultSet rs1;
    rs1 = stmt1.executeQuery(sql3);
    while(rs1.next())
    {
        String signup_num=rs1.getString(1);
        Vector<String> v = new Vector<String>();
        v.add(item_id);
        v.add(item_name);
    }
}
```

5.5 学生客户端报名



先查询项目，再输入编号报名

```

v_head.add("地点");
try
{
    String sql = "select * from item where cate_id = '"+map_cate.get(jcb.getSelectedItem())+"' order by item_id desc";
    rs = stmt.executeQuery(sql);
    while(rs.next())
    {
        String item_id = rs.getString(1).trim();
        String item_name = rs.getString(2).trim();
        String item_time = rs.getString(3);
        String rule = rs.getString(4);
        String promotion_num = rs.getString(5);
        String place = rs.getString(6);

        Vector<String> v = new Vector<String>();
        v.add(item_id);
        v.add(item_name);
        v.add(item_time);
        v.add(rule);
        v.add(promotion_num);
        v.add(place);
        v_data.add(v);
        v_itemid.add(item_id);
    }
    rs.close();
}

if(e.getSource()==jb)
{
    String item_id = jtfArray[1].getText().trim();
    if(item_id.equals(""))
    {
        JOptionPane.showMessageDialog(this,"项目号不得为空","错误",JOptionPane.ERROR_MESSAGE);
        return ;
    }
    if(!v_itemid.contains(item_id))
    {
        JOptionPane.showMessageDialog(this,"项目号不在列表中","错误",JOptionPane.ERROR_MESSAGE);
        return ;
    }
    try
    {
        this.initialConnection();
        String sql2 = "select * from grade where stu_id = '"+stu_id+"' and item_id = '"+item_id+"'";
        rs = stmt.executeQuery(sql2);
        if(rs.next())
        {
            JOptionPane.showMessageDialog(this,"你已经报过此项目","错误",JOptionPane.ERROR_MESSAGE);
            return ;
        }
        rs.close();
        //插入成绩表（暂时无成绩）
        String sql3 = "insert into grade(stu_id,item_id) values('"+stu_id+"','"+item_id+"')";
        int i = stmt.executeUpdate(sql3);
        if(i==1)

```

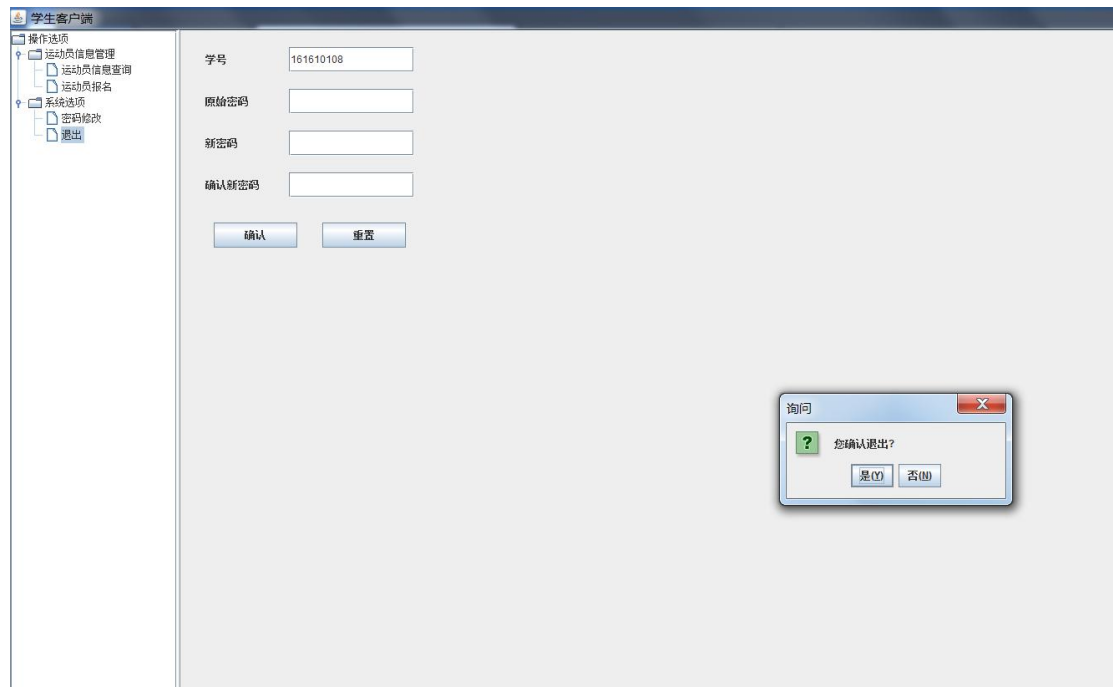
5.6 学生客户端修改密码

先检查是否两次输入新密码一致，再进行更改

```
String newPwd = jpfArray[1].getText().trim();
if(newPwd.equals(""))
{
    JOptionPane.showMessageDialog(this, "请输入新密码", "错误", JOptionPane.ERROR_MESSAGE);
    return ;
}
String newPwd1 = jpfArray[2].getText().trim();
if(!newPwd1.equals(newPwd))
{
    JOptionPane.showMessageDialog(this, "密码确认与新密码不同", "错误", JOptionPane.ERROR_MESSAGE);
    return ;
}

String sql = "update student set password = '"+newPwd+"' where stu_id = '"+user_name+"' and password = '"+oldPwd+"'";
int i = stmt.executeUpdate(sql);
if(i==0)
{
    JOptionPane.showMessageDialog(this, "修改失败，请检查您的用户名和密码是否正确", "错误", JOptionPane.ERROR_MESSAGE);
    return ;
}
else
{
    JOptionPane.showMessageDialog(this, "密码修改成功", "提示", JOptionPane.INFORMATION_MESSAGE);
    new Login();
    sc.dispose();
}
```

5.7 学生客户端退出系统



```
int i = JOptionPane.showConfirmDialog(jpy, "您确认退出?", "询问", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);  
if(i==0)  
{  
    System.exit(0);  
}
```

5.8 管理员客户端进入界面



同学生客户端

5.9 管理员客户端学生信息查询



```

this.initialConnection();
//查询学生单表
String sql = "select * from student where stu_id = '"+stu_id+"'";
rs = stmt.executeQuery(sql);
if(rs.next())
{
    this.removeAll();
    String stu_name = rs.getString(3).trim();
    String stu_gender = rs.getString(4).trim();
    String stu_age = rs.getString(5).trim();
    String team_id = rs.getString(6).trim();

    jlArray2[0] = new JLabel(stu_id);
    jlArray2[1] = new JLabel(stu_name);
    jlArray2[2] = new JLabel(stu_gender);
    jlArray2[3] = new JLabel(stu_age);
    jlArray2[4] = new JLabel(map_team.get(team_id));
    this.initialFrame();
    for(int i = 0;i<5;i++)
    {
        jlArray2[i].setBounds(150,70+i*50,150,30);
        this.add(jlArray2[i]);
    }
    this.repaint();
}
else

```

5.10 管理员客户端添加学生

管理员客户端

操作选项

- 学生信息管理
 - 学生信息查询
 - 添加学生
 - 删除学生
- 项目管理
 - 项目信息查询
 - 添加项目
 - 删除项目
- 院系设置
 - 队伍信息查询
 - 增加队伍
 - 删除队伍
- 查询功能
 - 模糊搜索
 - 高级搜索
- 系统选项
 - 密码修改
 - 退出

学号

姓名

性别

年龄

队伍

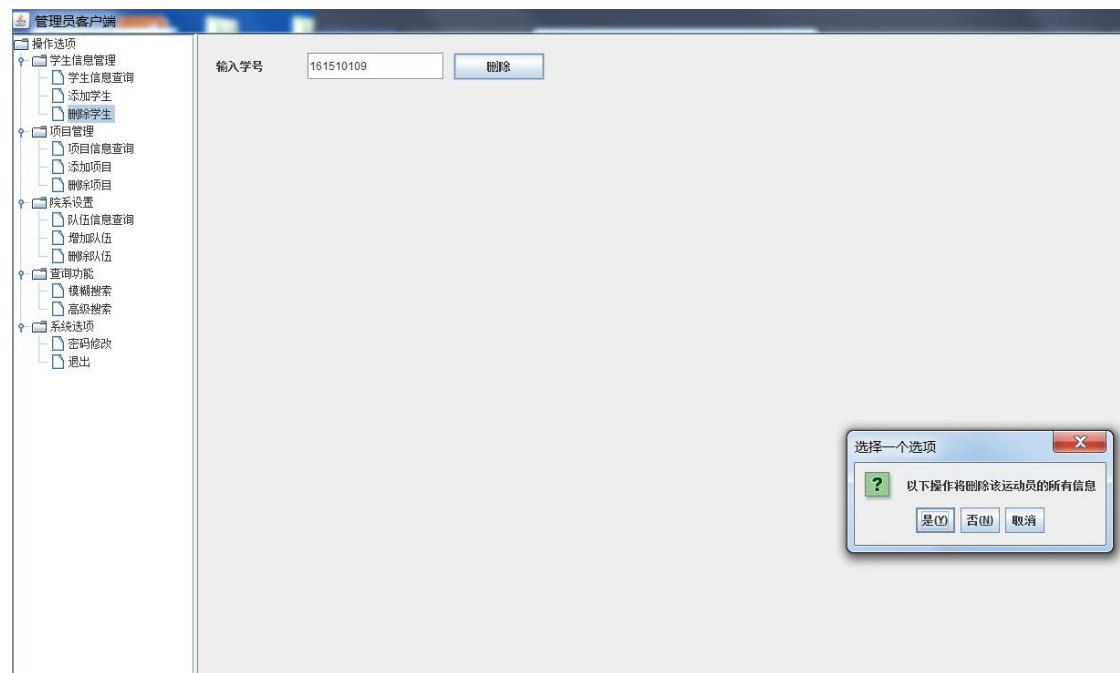
确认

重置

先查看学号是否已存在，再进行添加

```
try
{
    String sql = "insert into student "
        + "values('"+stu_id+"','"+stu_id+"','"+stu_name+"','"+stu_gender+"','"+stu_age+"','"+team_id+"')";
    int i = stmt.executeUpdate(sql);
    if(i==1)
    {
        JOptionPane.showMessageDialog(this, "添加成功", "正确", JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(this, "添加失败", "错误", JOptionPane.INFORMATION_MESSAGE);
    }
}
catch(SQLException e)
```

5. 11 管理员客户端删除学生

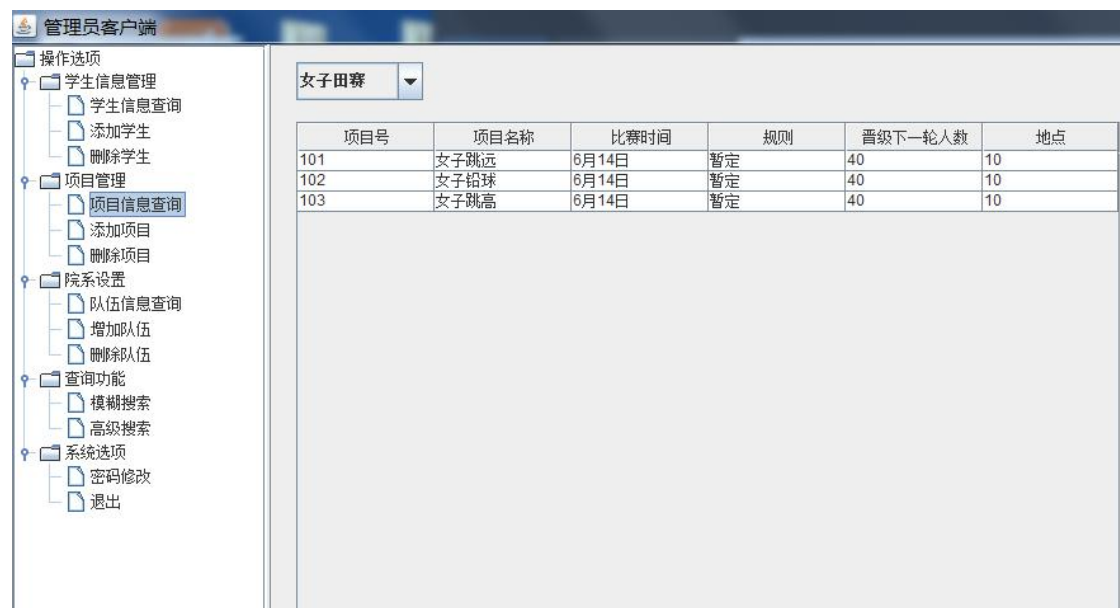


```

{
    this.initialConnection();
    try
    {
        String sql = "delete from student where stu_id = '"+stu_id+"'";
        int j = stmt.executeUpdate(sql);
        if(j==1)
        {
            JOptionPane.showMessageDialog(this, "删除成功", "提示", JOptionPane.INFORMATION_MESSAGE);
        }
        else
        {
            JOptionPane.showMessageDialog(this, "删除失败,请检查学号是否正确", "提示", JOptionPane.INFORMATION_MESSAGE);
        }
    }
    catch(SQLException ea)
    {
    }
}

```

5. 12 管理员客户端项目信息查询



```
try
{
    String sql = "select * from item where cate_id = '"+map_cate.get(jcb.getSelectedItem())+"' order by item_id";
    rs = stmt.executeQuery(sql);
    while(rs.next())
    {
        String item_id = rs.getString(1).trim();
        String item_name = rs.getString(2).trim();
        String item_time = rs.getString(3);
        String rule = rs.getString(4);
        String promotion_num = rs.getString(5);
        String place = rs.getString(6);

        Vector<String> v = new Vector<String>();
        v.add(item_id);
        v.add(item_name);
        v.add(item_time);
        v.add(rule);
        v.add(promotion_num);
        v.add(place);
        v_data.add(v);
        v_itemid.add(item_id);
    }
    rs.close();
}
```

5.13 管理员客户端添加项目

管理员客户端

操作选项

- 学生信息管理
 - 学生信息查询
 - 添加学生
 - 删除学生
- 项目管理
 - 项目信息查询
 - 添加项目
 - 删除项目
- 院系设置
 - 队伍信息查询
 - 增加队伍
 - 删除队伍
- 查询功能
 - 模糊搜索
 - 高级搜索
- 系统选项
 - 密码修改
 - 退出

项目编号

项目名称

时间

晋级人数

项目分类

规则

确认

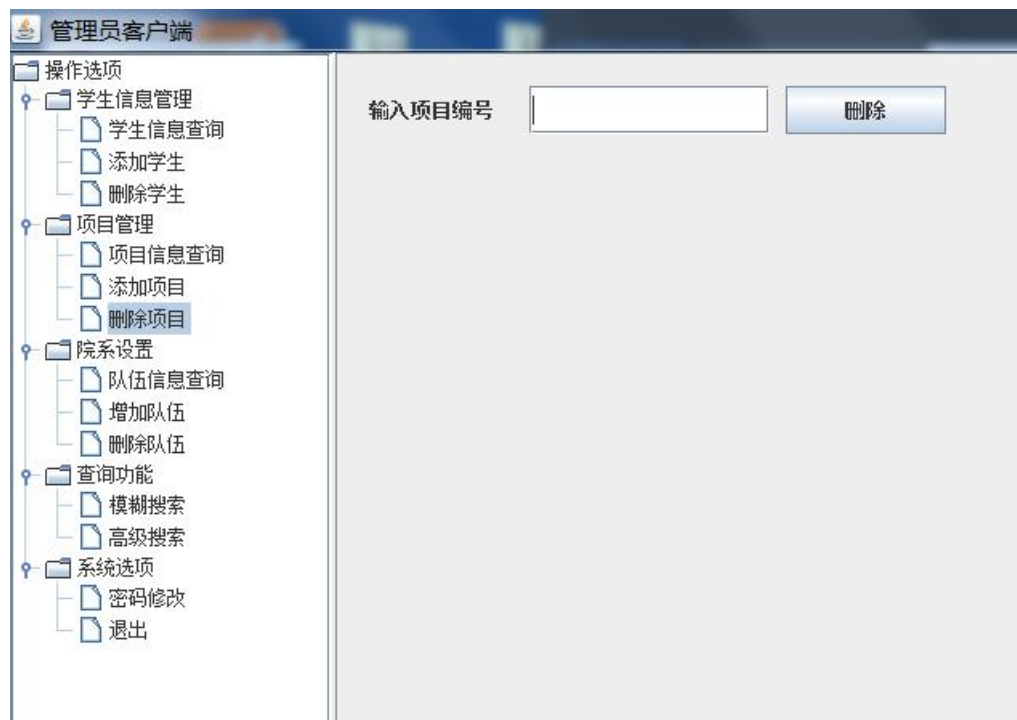
先查看项目是否已存在，然后再添加

```

String sql = "select * from item where item_id = '"+item_id+"'";
rs = stmt.executeQuery(sql);
if(rs.next())
{
    JOptionPane.showMessageDialog(this, "该项目已经存在", "错误", JOptionPane.ERROR_MESSAGE);
    this.closeConn();
    return ;
}
rs.close();
String sql2 = "select * from item where item_name = '"+item_name+"'";
rs = stmt.executeQuery(sql2);
if(rs.next())
{
    JOptionPane.showMessageDialog(this, "项目名称与已有项目名称重复", "错误", JOptionPane.ERROR_MESSAGE);
    this.closeConn();
    return ;
}
rs.close();
String sql3 = "insert into item "
    + "values('"+item_id+"','"+item_name+"','"+time+"','"+rule+"','"+promotion_num+"','"+cate_id+"') ";
int i = stmt.executeUpdate(sql3);
if(i==1)
{
    JOptionPane.showMessageDialog(this, "添加成功", "提示", JOptionPane.INFORMATION_MESSAGE);
}
else
{
    JOptionPane.showMessageDialog(this, "添加失败", "错误", JOptionPane.INFORMATION_MESSAGE);
}
}

```

5. 14 管理员客户端删除项目

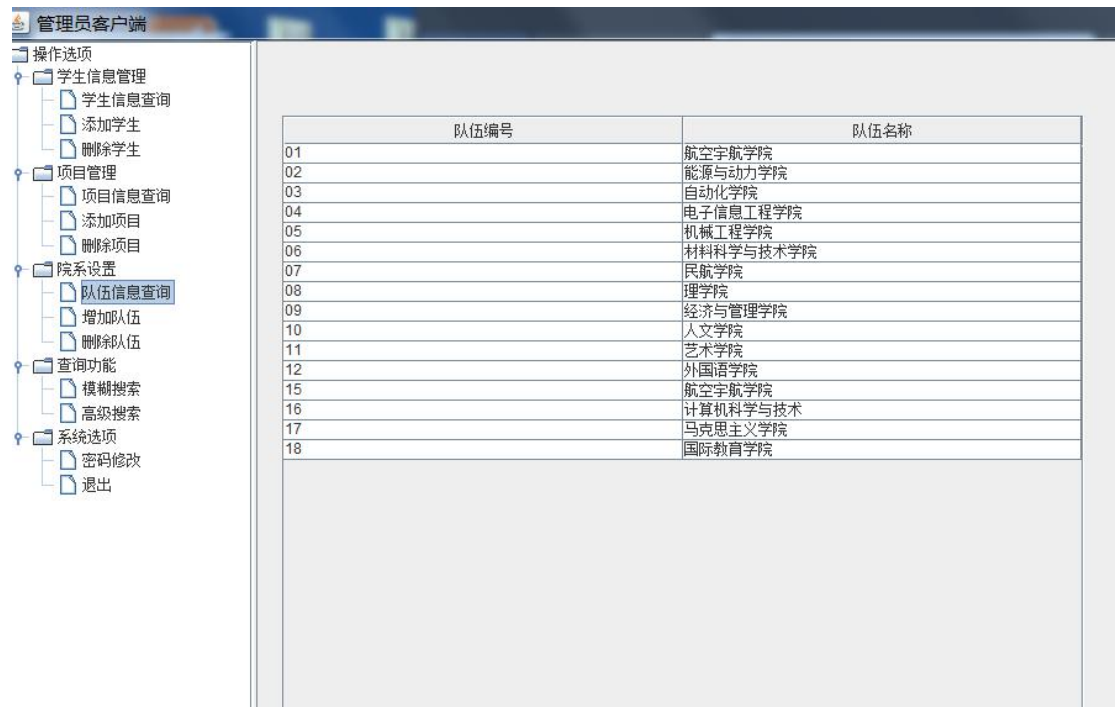


```

try
{
    String sql = "delete from item where item_id = '"+item_id+"'";
    int j = stmt.executeUpdate(sql);
    if(j==1)
    {
        JOptionPane.showMessageDialog(this, "删除成功", "提示", JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(this, "删除失败,请检查编号是否正确", "提示", JOptionPane.INFORMATION_MESSAGE);
    }
}
}

```

5. 15 管理员客户端队伍信息查询



```

try
{
    String sql = "select * from team order by team_id";
    rs = stmt.executeQuery(sql);
    while(rs.next())
    {
        String team_id = rs.getString(1).trim();
        String team_name = rs.getString(2).trim();

        Vector<String> v = new Vector<String>();
        v.add(team_id);
        v.add(team_name);

        v_data.add(v);
        v_itemid.add(team_id);
    }
    rs.close();
}

```

5.16 管理员客户端增加队伍

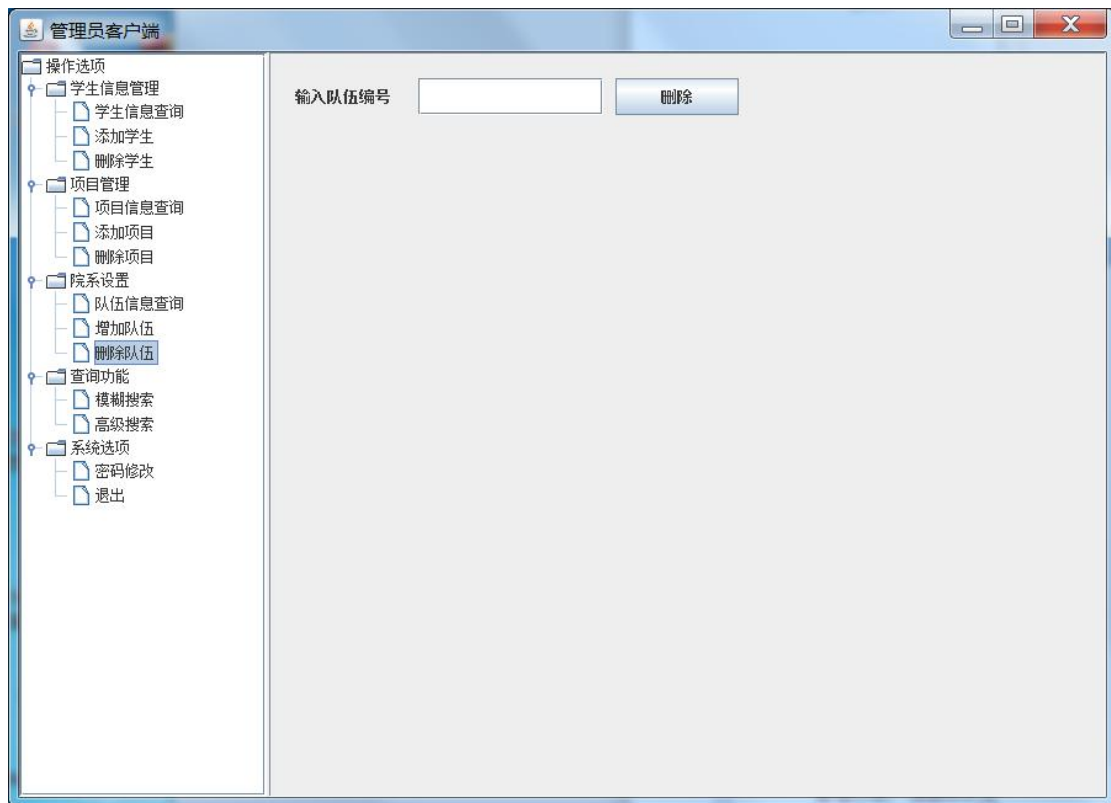


先检查是否已有该队伍，再添加

```
try
{
    String sql = "select * from team where team_id = '"+team_id+"'";
    rs = stmt.executeQuery(sql);
    if(rs.next())
    {
        JOptionPane.showMessageDialog(this, "该院系编号已存在", "错误", JOptionPane.ERROR_MESSAGE);
        this.closeConn();
        return ;
    }
    rs.close();
    String sql2 = "select * from team where team_name = '"+team_name+"'";
    rs = stmt.executeQuery(sql2);
    if(rs.next())
    {
        JOptionPane.showMessageDialog(this, "院系名称错误", "错误", JOptionPane.ERROR_MESSAGE);
        this.closeConn();
        return ;
    }
    rs.close();
    String sql3 = "insert into team values('"+team_id+"','"+team_name+"')";
    int i = stmt.executeUpdate(sql3);
    if(i==1)
    {
        JOptionPane.showMessageDialog(this, "添加成功", "提示", JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {

```

5. 17 管理员客户端删除队伍



```
this.initialConnection();
try
{
    String sql = "delete from team where team_id = '"+team_id+"'";
    int j = stmt.executeUpdate(sql);
    if(j==1)
    {
        JOptionPane.showMessageDialog(this, "删除成功", "提示", JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(this, "删除失败,请检查编号是否正确", "提示", JOptionPane.INFORMATION_MESSAGE);
    }
}
catch(SQLException ea)
{
}
```

5. 18 管理员客户端模糊搜索



为提高搜索效率，运用 locate 函数将时间缩短

```

i
    v_data.clear();
    this.initialConnection();
    String sql = "select distinct item.item_name,student.stu_id,student.stu_name,student.age,student.sex " +
        "from grade,item,student " +
        "where " +
        "( " +
        "student.stu_id=grade.stu_id and item.item_id=grade.item_id" +
        ") and " +
        "( " +
        "locate('"+key+"',stu_name)>0"+
        ")";
    rs = stmt.executeQuery(sql);
    while(rs.next())
    {
        Vector<String> v = new Vector<String>();
        String item_name = rs.getString(1).trim();
    }

```

5. 19 管理员客户端高级搜索

管理员客户端

操作选项

- 学生信息管理
 - 学生信息查询
 - 添加学生
 - 删除学生
- 项目管理
 - 项目信息查询
 - 添加项目
 - 删除项目
- 院系设置
 - 队伍信息查询
 - 增加队伍
 - 删除队伍
- 查询功能
 - 模糊搜索
 - 高级搜索
- 系统选项
 - 密码修改
 - 退出

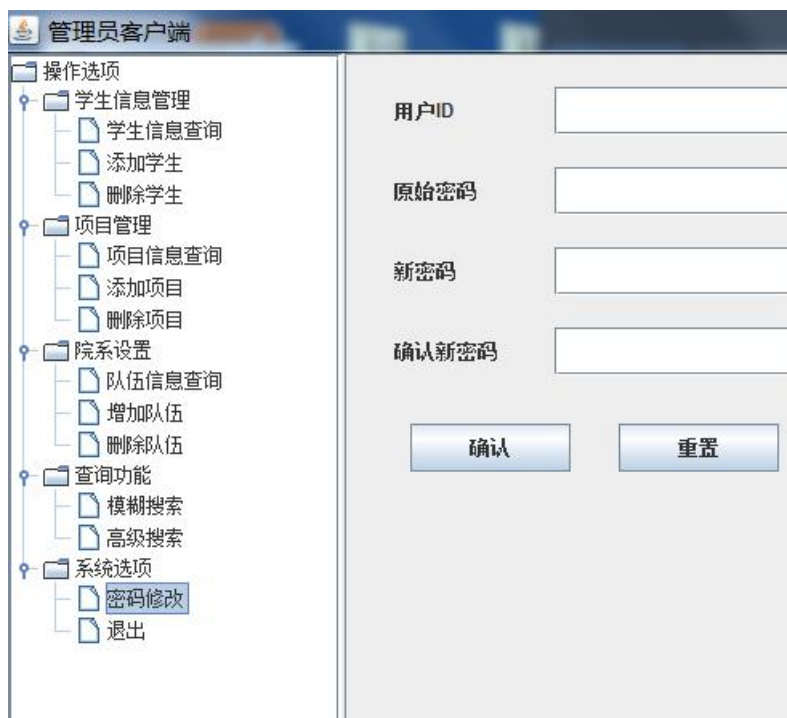
性别: 男 出生年份: 1998 所属院系: 航空... 报名项目: 男子跳远 确定

项目名称	学号	姓名	年龄	性别	所属队伍
男子跳远	151410241	赫连建家	21	男	航空字航学院
男子跳远	151410351	欧启	21	男	航空字航学院
男子跳远	151410436	邱涛	21	男	航空字航学院
男子跳远	151410906	董翰	21	男	航空字航学院
男子跳远	151411116	福广志	21	男	航空字航学院
男子跳远	151411246	官和	21	男	航空字航学院
男子跳远	151411251	于伯	21	男	航空字航学院
男子跳远	151411811	奎圣和	21	男	航空字航学院
男子跳远	151411916	殷进	21	男	航空字航学院
男子跳远	151412416	史毅冠	21	男	航空字航学院
男子跳远	151420201	李岩中	21	男	航空字航学院
男子跳远	151420451	曾建家	21	男	航空字航学院
男子跳远	151420741	姜和	21	男	航空字航学院
男子跳远	151421341	司徒勤霖	21	男	航空字航学院
男子跳远	151421406	柴逸军	21	男	航空字航学院
男子跳远	151421746	青致谦	21	男	航空字航学院
男子跳远	151422226	董志	21	男	航空字航学院
男子跳远	151422501	吴子	21	男	航空字航学院
男子跳远	151422946	万侯鹏洋	21	男	航空字航学院
男子跳远	151423006	陆博	21	男	航空字航学院
男子跳远	151430121	赫连民	21	男	航空字航学院
男子跳远	151430146	郝松基	21	男	航空字航学院
男子跳远	151430421	步敬树	21	男	航空字航学院
男子跳远	151430916	竺青	21	男	航空字航学院
男子跳远	151431106	门冠	21	男	航空字航学院
男子跳远	151431536	苗建家	21	男	航空字航学院
男子跳远	151431541	廖先	21	男	航空字航学院
男子跳远	151431601	项豪	21	男	航空字航学院
男子跳远	151431831	戎成	21	男	航空字航学院
男子跳远	151432016	昌博	21	男	航空字航学院
男子跳远	151432146	风顺	21	男	航空字航学院
男子跳远	151432331	慕盛	21	男	航空字航学院
男子跳远	151432636	邓德	21	男	航空字航学院
男子跳远	151433111	鲁胜	21	男	航空字航学院
男子跳远	151440806	太史刚	21	男	航空字航学院
男子跳远	151440811	卢伟刚	21	男	航空字航学院

```
String sql = "select distinct item.item_name,student.stu_id,student.stu_name,student.age,student.sex,team.team_name " +
            "from grade,item,student,team " +
            "where student.stu_id=grade.stu_id and item.item_id=grade.item_id and student.team_id=team.team_id ";
if(!sex.equals("不限"))
{
    sql += " and sex = '"+sex+"'";
}
if(!age.equals("不限"))
{
    sql += "and age = '"+age+"'";
}
if(!team_id.equals("不限"))
{
    sql += "and student.team_id = '"+team_id+"'";
}
if(!item_id.equals("不限"))
{
    sql += "and grade.item_id = '"+item_id+"'";
}
rs = stmt.executeQuery(sql);
while(rs.next())
{
    String itemname = rs.getString(1).trim();
```

字段为不限时就不在 string 后添加该字段，选定时则需要添加在查找字段中，即在 where 里限定条件。

5.20 管理员修改密码

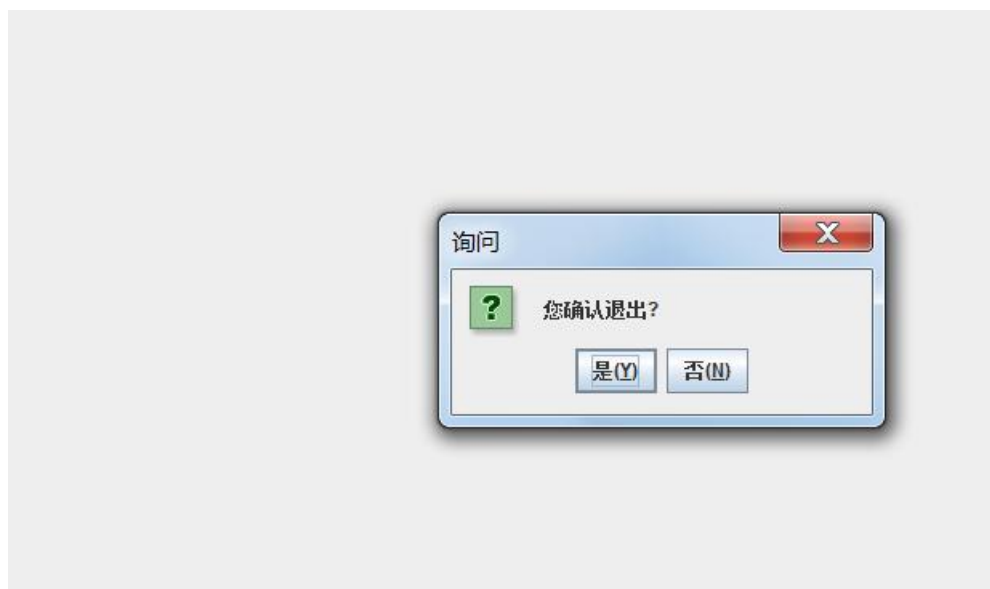


```

try {
    stmt.executeUpdate(sql);
    int i = stmt.executeUpdate(sql);
    if(i==0)
    {
        JOptionPane.showMessageDialog(this, "修改失败, 请检查您的用户名和密码是否正确", "错误", JOptionPane.ERROR_MESSAGE);
        return ;
    }
    else
    {
        JOptionPane.showMessageDialog(this, "密码修改成功", "提示", JOptionPane.INFORMATION_MESSAGE);
        new Login();
        ac.dispose();
    }
}

```

5. 21 管理员退出系统



```

e if(id.equals("42"))

int i = JOptionPane.showConfirmDialog(jpy, "您确认退出?", "询问", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
if(i==0)
{
    System.exit(0);
}

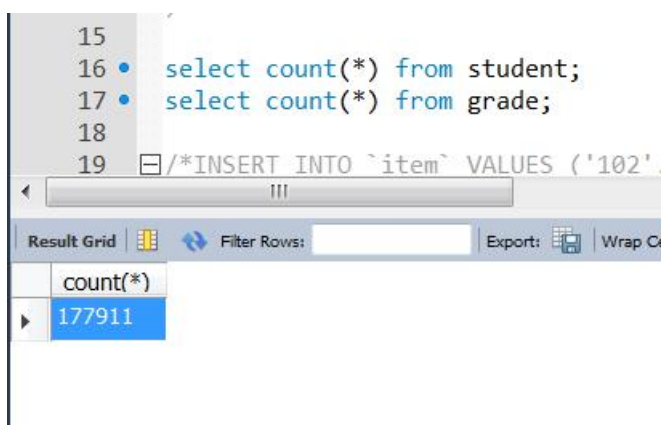
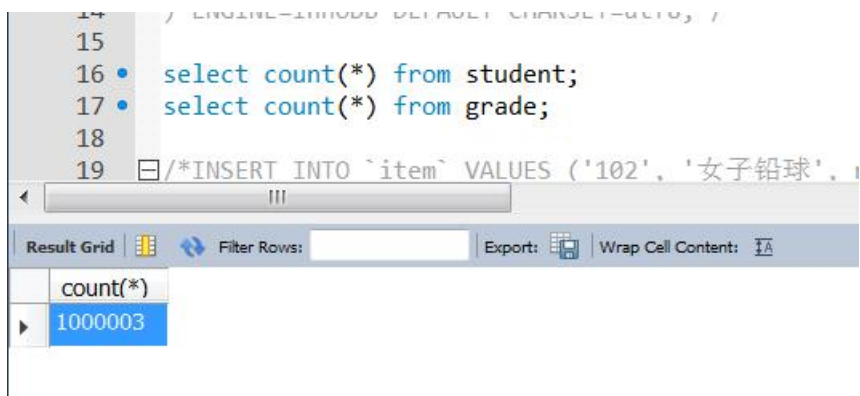
```

6 百万条数据

```
1 • INSERT INTO `student` VALUES ('021410101', '021410101', '微生兴', '男', '22', '02');
2 • INSERT INTO `student` VALUES ('021410102', '021410102', '覃安', '男', '21', '02');
3 • INSERT INTO `student` VALUES ('021410103', '021410103', '满平保', '男', '23', '02');
4 • INSERT INTO `student` VALUES ('021410104', '021410104', '钮建家', '男', '23', '02');
5 • INSERT INTO `student` VALUES ('021410105', '021410105', '闵博', '男', '21', '02');
6 • INSERT INTO `student` VALUES ('021410106', '021410106', '应进', '男', '22', '02');
7 • INSERT INTO `student` VALUES ('021410107', '021410107', '咸裕', '男', '23', '02');
8 • INSERT INTO `student` VALUES ('021410108', '021410108', '南门壮', '男', '23', '02');
9 • INSERT INTO `student` VALUES ('021410109', '021410109', '岳谦', '男', '23', '02');
10 • INSERT INTO `student` VALUES ('021410110', '021410110', '成裕', '男', '23', '02');
11 • INSERT INTO `student` VALUES ('021410111', '021410111', '第五洁梅', '女', '22', '02');
12 • INSERT INTO `student` VALUES ('021410112', '021410112', '公良绍功', '男', '21', '02');
13 • INSERT INTO `student` VALUES ('021410113', '021410113', '包缺仪', '女', '21', '02');
14 • INSERT INTO `student` VALUES ('021410114', '021410114', '卓绍功', '男', '21', '02');
15 • INSERT INTO `student` VALUES ('021410115', '021410115', '扶寒', '女', '22', '02');
16 • INSERT INTO `student` VALUES ('021410116', '021410116', '从保', '男', '22', '02');
17 • INSERT INTO `student` VALUES ('021410117', '021410117', '鄢岩中', '男', '21', '02');
18 • INSERT INTO `student` VALUES ('021410118', '021410118', '空继启', '男', '22', '02');
19 • INSERT INTO `student` VALUES ('021410119', '021410119', '侯泽', '男', '23', '02');
20 • INSERT INTO `student` VALUES ('021410120', '021410120', '莫启', '男', '22', '02');
21 • INSERT INTO `student` VALUES ('021410121', '021410121', '慕可', '女', '21', '02');
22 • INSERT INTO `student` VALUES ('021410122', '021410122', '殷环', '女', '21', '02');
23 • INSERT INTO `student` VALUES ('021410123', '021410123', '易妹', '女', '22', '02');
24 • INSERT INTO `student` VALUES ('021410124', '021410124', '项东文', '男', '22', '02');

1 • INSERT INTO `grade` VALUES ('021410116', '111', '76');
2 • INSERT INTO `grade` VALUES ('021410121', '202', '63');
3 • INSERT INTO `grade` VALUES ('021410126', '202', '70');
4 • INSERT INTO `grade` VALUES ('021410136', '211', '87');
5 • INSERT INTO `grade` VALUES ('021410141', '111', '53');
6 • INSERT INTO `grade` VALUES ('021410146', '111', '77');
7 • INSERT INTO `grade` VALUES ('021410206', '201', '96');
8 • INSERT INTO `grade` VALUES ('021410211', '211', '93');
9 • INSERT INTO `grade` VALUES ('021410221', '113', '76');
10 • INSERT INTO `grade` VALUES ('021410226', '113', '92');
11 • INSERT INTO `grade` VALUES ('021410231', '211', '58');
12 • INSERT INTO `grade` VALUES ('021410236', '112', '91');
13 • INSERT INTO `grade` VALUES ('021410241', '201', '92');
14 • INSERT INTO `grade` VALUES ('021410246', '112', '63');
15 • INSERT INTO `grade` VALUES ('021410301', '211', '98');
16 • INSERT INTO `grade` VALUES ('021410306', '112', '74');
17 • INSERT INTO `grade` VALUES ('021410311', '212', '59');
18 • INSERT INTO `grade` VALUES ('021410316', '113', '54');
19 • INSERT INTO `grade` VALUES ('021410321', '112', '59');
20 • INSERT INTO `grade` VALUES ('021410326', '212', '76');
21 • INSERT INTO `grade` VALUES ('021410331', '101', '64');
22 • INSERT INTO `grade` VALUES ('021410336', '102', '82');
23 • INSERT INTO `grade` VALUES ('021410341', '202', '66');
24 • INSERT INTO `grade` VALUES ('021410346', '113', '95');
```

暴力插入一百万条数据



7 结束语

通过本学期的数据库原理实验课，把书上的理论知识都在自己的电脑上实际运用了一遍，并且真正的把数据库应用到一个管理系统中。从一开始的建表就需要注意一些表项冗余或者函数依赖的问题，主码和外码的设计在数据处理中非常重要。之前也几乎没有怎么做过界面，综合实验中也学会了划分面板、树结构等 Swing 的东西，觉得非常的有意思，也增强了交互性。在完成的过程中也参考了很多资料，比如 JDBC、Swing 之类的。但是自己做的系统还是有一个不完善的地方，添加队伍以后没能立刻刷新面板上的所有队伍信息，改了好几次也没有成功，如果时间足够的话，我会再查查和做界面有关的资料。

致谢

在完成本学期数据库原理实验中，得到了老师的指导和同学们的帮助，对数据库管理系统的具体实现有了更深刻的理解，在此致以诚挚的谢意！

参考文献:

- [1] 王珊. 《数据库系统概论》. 第 5 版[M]. 高等教育出版社:2014.
- [2] CSDN 博客
- [3] 百度知道