## 学生选课系统

JavaWeb项目下载地址

https://ke.qq.com/course/2024761?tuin=23890943

**要求具备如下知识：**

1. Java基础
2. 前端基础（Html、css、JavaScript、Jquery）
3. mysql数据库
4. Web容器（Tomcat）
5. Jsp/Servlet

**通过该项目可以掌握哪些知识：**

1. 掌握整个项目的流程
2. 掌握开发中的简单权限控制
3. 掌握前端页面如何与后端整合
4. 宏观把握一个项目
5. 掌握echarts图表技术

# 技术选型

Web容器：Apache Tomcat8.5

编程语言：Java8

数据库连接池：DBCP

数据库操作：DBUtils

动态网页：JSP/Servlet3.1

静态网页：HTML+CSS+Jquery

数据库：MySQL5.6

数据库设计软件：Power Designer

操作系统：Window7

# 需求描述

**角色**：学生、老师、管理员

**2.1 学生：基本信息、课程信息**

1. 学生操作流程：
2. 学生登录系统
3. 选择课程并提交确认（确认之后则不能修改了）
4. 考试系统(省略)
5. 各种查询（查询各科成绩、查询总成绩、查询排名等）
6. 学生查询功能
   1. 各科成绩查询
   2. 总成绩查询
   3. 成绩排名、各科成绩排名

**2.2 老师：基本信息、所教课程**

1. 老师操作流程：
2. 登录系统
3. 课程管理（实现对课程的CURD）
4. 考试成绩录入
5. 各种查询（学生的单科成绩、总成绩、排名）
6. 各种统计（统计及格率、统计>80分以上的同学等）

**2.3 管理员：维护学生和老师信息、查询统计**

1. 管理员操作流程：
   1. 登录系统
   2. 实现对学生的CURD
   3. 实现对老师的CURD
   4. 实现各种查询、统计

登录

成功

失败

选课

评分

查询成绩

# 数据库设计

详情查看pdm图

# 项目搭建

1. 创建项目（dynamic Web project）sct
2. 建包
   1. com.hua.entity(model\vo)：student实体、课程实体等
   2. com.hua.dao（Data Access Object 数据访问对象）操作数据库
   3. com.hua.utils:工具包
   4. com.hua.servlet(web\action\controller)：存放servlet项目的文件
   5. com.hua.service(biz)：存放业务逻辑的

举例：甲🡪乙转账业务:甲转出（JDao.out()调用）🡪乙转入（YDao.in()调用），他们之间要有事务。

1. 编码

配置数据库连接文件db.properties

加载db.properties配置文件

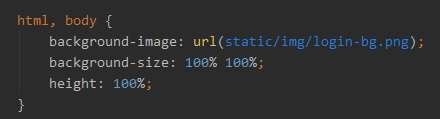
url=jdbc:mysql://localhost:3306/sct?characterEncoding=utf-8&serverTimezone=Asia/Shanghai  
username=root  
password=root  
driverClassName=com.mysql.cj.jdbc.Driver

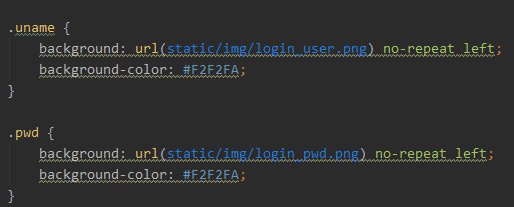
建立数据库连接，返回数据源

# 项目界面

1. 登录页面login.html

背景图片设置方法



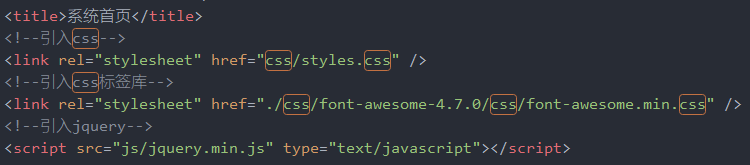


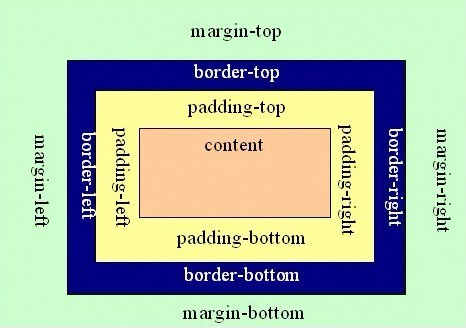
1. 首页：index.html
2. 列表页面：list.html
3. 新增或修改页面：add.html/edit.html

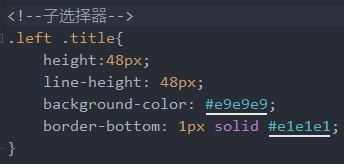
# 项目界面与代码整合

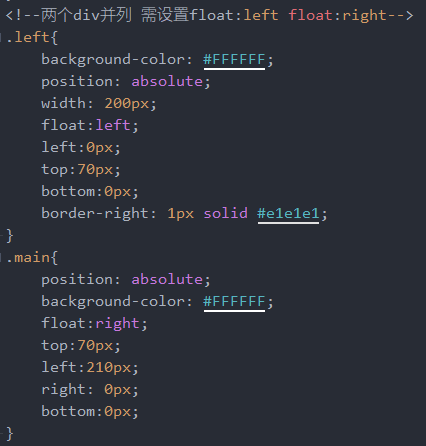
详解视频

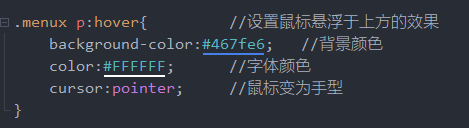
这部分学到的知识

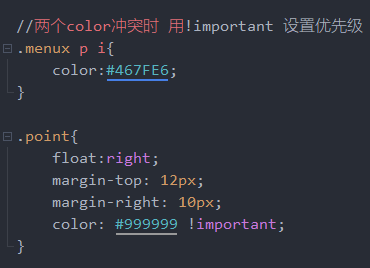


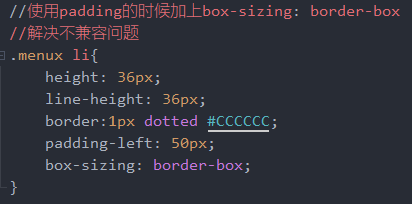


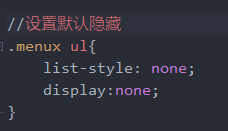


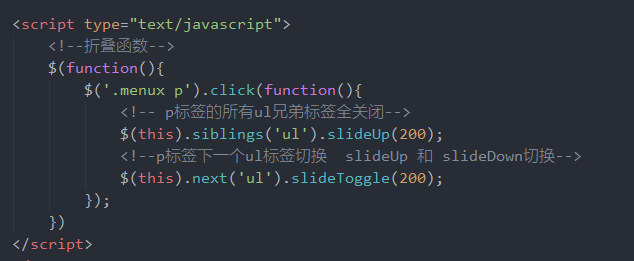


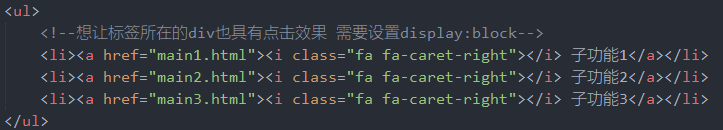




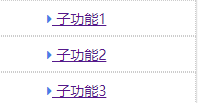
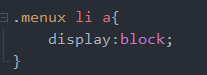




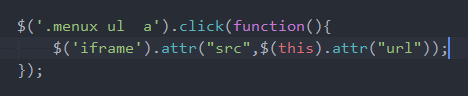


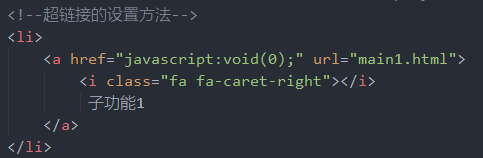


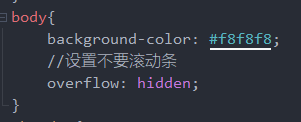
让他成为一个块



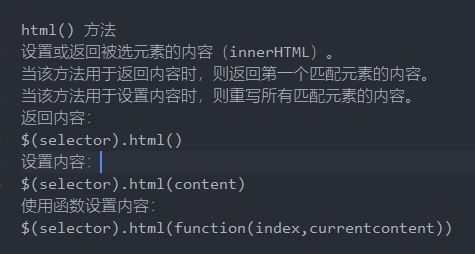


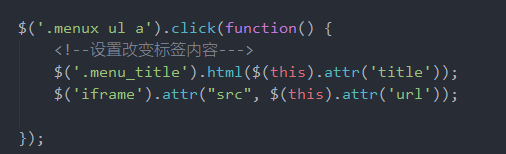


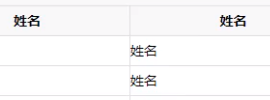




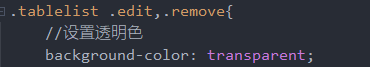


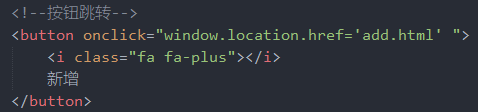


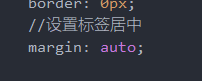


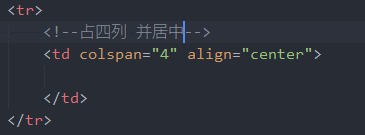
 















# 登录

管理员：

老师：

学生：

1、假设：后端设计数据库中账号字段的时候，有规律可循：

管理员：1001开头 1001001、1001002

老师：2001开头 2001001、2001002

老师：3001开头 3001001、3001002

2、假设：后端设计数据库中账号字段没有规律，此时需要添加一个登录类型：



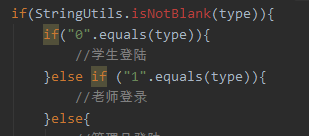
Md5加密 双重加密

package com.hua.utils;  
  
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;  
public class MD5 {  
  
 /\*  
 \* 1.一个运用基本类的实例 MessageDigest 对象开始被初始化。该对象通过使用 update 方法处理数据。 任何时候都可以调用 reset  
 \* 方法重置摘要。 一旦所有需要更新的数据都已经被更新了，应该调用 digest 方法之一完成哈希计算。 对于给定数量的更新数据，digest  
 \* 方法只能被调用一次。 在调用 digest 之后，MessageDigest 对象被重新设置成其初始状态。  
 \*/  
 public static String encrypByMd5(String context) {  
 try {  
 MessageDigest md = MessageDigest.*getInstance*("MD5");  
 md.update(context.getBytes());// update处理  
 byte[] encryContext = md.digest();// 调用该方法完成计算  
  
 int i;  
 StringBuffer buf = new StringBuffer("");  
 for (int offset = 0; offset < encryContext.length; offset++) {// 做相应的转化（十六进制）  
 i = encryContext[offset];  
 if (i < 0)  
 i += 256;  
 if (i < 16)  
 buf.append("0");  
 buf.append(Integer.*toHexString*(i));  
 }  
 return buf.toString();  
 //System.out.println("32result: " + buf.toString());// 32位的加密  
 //System.out.println("16result: " + buf.toString().substring(8, 24));// 16位的加密  
 } catch (NoSuchAlgorithmException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 return null;  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println(MD5.*encrypByMd5*("e10adc3949ba59abbe56e057f20f883e"));  
 }  
}

导入commons-lang3-3.8.1包

注意导入的包是：import org.apache.commons.lang3.StringUtils;

该方法判断比较全面



# 登录权限过滤

PermissionFilter可以过滤是否登录的权限！

继承 java.Servlet.Filter

1. import javax.servlet.\*;
2. import javax.servlet.annotation.WebFilter;
3. import javax.servlet.annotation.WebInitParam;
4. import javax.servlet.http.HttpServletRequest;
5. import javax.servlet.http.HttpServletResponse;
6. import java.io.IOException;
7. *//拦截所有*
8. @WebFilter(urlPatterns = {"/\*"},initParams = {
9. @WebInitParam(name="exclude",value="/sct1/,/login.jsp,/login,/noprivilege.jsp")
10. })
11. public class PermissionFilter implements Filter {
12. public static String excludeString;
13. @Override
14. public void init(FilterConfig filterConfig) throws ServletException {
15. excludeString=filterConfig.getInitParameter("exclude");
16. }
17. @Override
18. public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws
19. IOException, ServletException {
20. *//先将req requ 转换*
21. HttpServletRequest httpServletRequest=(HttpServletRequest) servletRequest;
22. HttpServletResponse httpServletResponse=(HttpServletResponse) servletResponse;
23. Object user=httpServletRequest.getSession().getAttribute("user");
24. String uri=httpServletRequest.getRequestURI();
25. if(isExist(uri)){
26. filterChain.doFilter(httpServletRequest,httpServletResponse);
27. }else{
28. if(user!=null){
29. filterChain.doFilter(httpServletRequest,httpServletResponse);
30. }else {
31. httpServletResponse.sendRedirect("noprivilege.jsp");
32. }
33. }
34. }
35. public static boolean isExist(String uri){
36. *//最后url的结尾语excludeString匹配*
37. String[] arr=excludeString.split(",");
38. boolean flag=false;
39. for(String i:arr){
40. if(uri.endsWith(i)){
41. flag=true;
42. }
43. }
44. return flag;
45. }
46. @Override
47. public void destroy() {
48. }
49. }

导入jstl包



退出界面



调用windows弹框 重定位



# 权限的功能设置

**管理员：**

1. 学生管理：CURD
2. 老师管理：CURD、（老师课程设置）
3. 课程管理：CURD
4. 统计查询：（及格率、每个科目等等统计）

**老师：**

1. 登录
2. 录入分数
   1. 先查询自己所教的课程(管理员已经设置过了)
   2. 查询所教课程下的学生（学生角色选课的过程，学生登录后做选课）
   3. 最后录入评分,提交即可（老师角色评分即可）
3. 查询统计（统计及格率、所教科目统计等等）

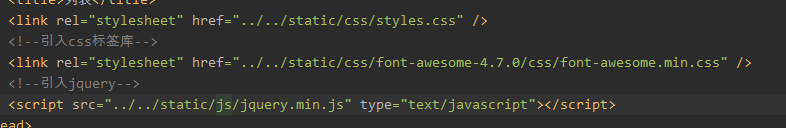
**学生：**

1. 登录
   1. 登录操作
2. 选课
   1. 查询所有的课程
   2. 选课其实就是选择老师（因为课程关联老师）
3. 考试（考试系统…），老师录入分数之后。（6b步骤）
4. 查询考试成绩、查询排名等

# 学生管理

1. 列表查询

路径问题

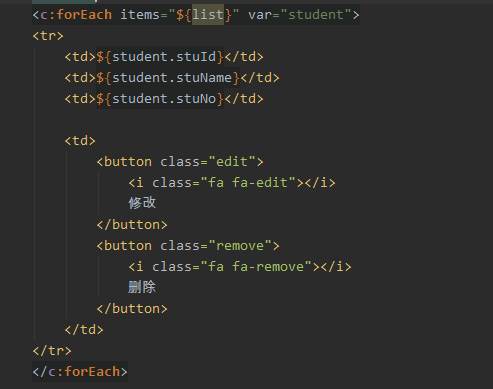


../表示往上退一级

servlet路径问题 写一个Filter 设置一个域对象

1. package com.hua.utils;
2. import com.sun.deploy.net.HttpRequest;
3. import javax.jws.WebService;
4. import javax.servlet.\*;
5. import javax.servlet.annotation.WebFilter;
6. import javax.servlet.http.HttpServletRequest;
7. import javax.servlet.http.HttpServletResponse;
8. import java.io.IOException;
9. @WebFilter(urlPatterns = {"/\*"})
10. public class PathFilter implements Filter {
11. @Override
12. public void init(FilterConfig filterConfig) throws ServletException {
13. }
14. @Override
15. public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {
16. HttpServletRequest httpServletRequest=(HttpServletRequest)servletRequest;
17. HttpServletResponse httpServletResponse=(HttpServletResponse)servletResponse;
18. String path=httpServletRequest.getContextPath();
19. String basePath = httpServletRequest.getScheme()+"://"+httpServletRequest.getServerName()+":"+httpServletRequest.getServerPort()+path+"/";
20. System.out.println(basePath);
21. httpServletRequest.setAttribute("basePath",basePath);
22. filterChain.doFilter(httpServletRequest,httpServletResponse);
23. }
24. @Override
25. public void destroy() {
26. }
27. }

采用jstl和el表达式 循环添加表格



1. 新增
2. <button class="edit" type="button" onclick="***window***.history.back(-1)">  
    <i class="fa fa-arrow-left"></i>  
    返回  
   </button>

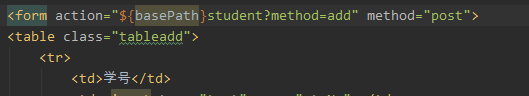
返回按钮

按钮要加类型 不然会默认为submit

在列表中添加完数据 直接重定向回来 直接显示



尽量吧 路径写完整



表单验证



去下载jquery validate包 然后引入（用于表单验证 重点）



1. (function( factory ) {
2. if ( typeof define === "function" && define.amd ) {
3. define( ["jquery", "../jquery.validate"], factory );
4. } else {
5. factory( jQuery );
6. }
7. }(function( $ ) {
8. */\**
9. \* Translated default messages for the jQuery validation plugin.
10. \* Locale: ZH (Chinese, 中文 (Zhōngwén), 汉语, 漢語)
11. \*/
12. $.extend($.validator.messages, {
13. required: "这是必填字段",
14. remote: "请修正此字段",
15. email: "请输入有效的电子邮件地址",
16. url: "请输入有效的网址",
17. date: "请输入有效的日期",
18. dateISO: "请输入有效的日期 (YYYY-MM-DD)",
19. number: "请输入有效的数字",
20. digits: "只能输入数字",
21. creditcard: "请输入有效的信用卡号码",
22. equalTo: "你的输入不相同",
23. extension: "请输入有效的后缀",
24. maxlength: $.validator.format("最多可以输入 {0} 个字符"),
25. minlength: $.validator.format("最少要输入 {0} 个字符"),
26. rangelength: $.validator.format("请输入长度在 {0} 到 {1} 之间的字符串"),
27. range: $.validator.format("请输入范围在 {0} 到 {1} 之间的数值"),
28. max: $.validator.format("请输入不大于 {0} 的数值"),
29. min: $.validator.format("请输入不小于 {0} 的数值")
30. });
31. }));
32. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成



转发

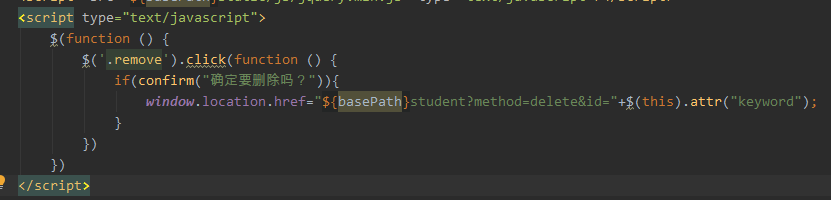
*request.getRequestDispatcher("/two").forward(request, response);*

重定向

response.sendRedirect("two");

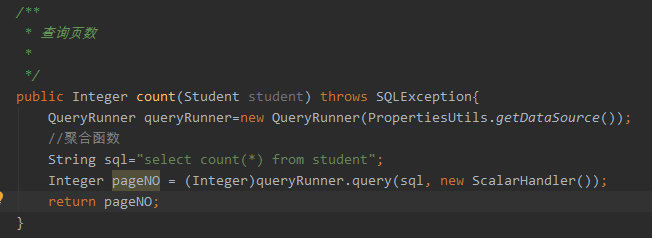


删除的时候必须要写确认 行业潜规则

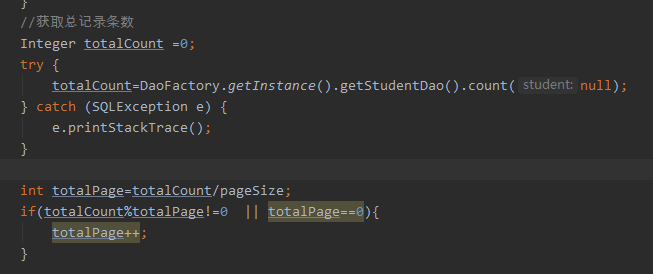


解决乱码问题采用过滤器

1. 组合条件查询
2. 分页



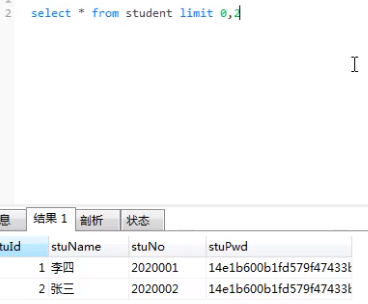
很坑的点



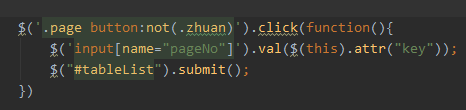
当记录数不够一页时，计算所得的总页数为0，页数不能为零，所以判断条件加上一个，totalPage==0;

查询数据库前两条数据

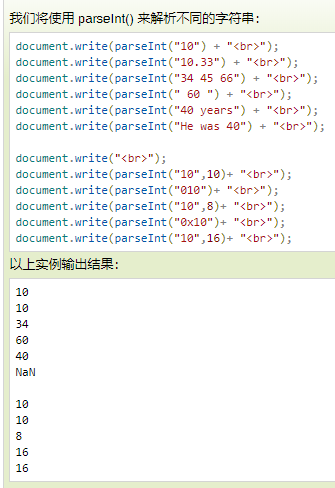
第一个参数叫起始条数 第二个参数交页面大小



排除其他按钮用 not



Js parseInt( )方法



解决乱码问题采用过滤器

# 分页技术实现

分页：数据库分页查询语句+掌握jsp+java等综合技术

**常见第一种**



**常见第二种**



**常用分页技术：**

内存分页:一次性将数据查询出来放入内存，然后进行分页，每页的数据都是在内存中存储，所以查询效率高。适用于小数据量，要求查询效率较高。

数据库分页：每次从数据库中查询出来一次的数据，然后放入内存，如果点击下一页，则重新从数据库查询下一页数据放入内存，上一页内存数据就被GC回收。适用于数据量较大的，查询效率较低。

**分页技术点**

****

页面大小(PageSize)：已知设置的10；

总记录条数（TotalCount）数据库查询而来；

总页数(totalPage)：TotalCount /pageSize = 85;

如果：totalCount%pageSize!=0 +1

如果：totalCount%pageSize==0 +0

页码数（PageNo）：默认为1，

下一页 pageNo+1（尾页的时候，不加1了），

上一页pageNo-1（首页的时候不减1）

尾页：pageNo = totalPage

首页：pageNo = 1

分页的封装

1. <%@ page language="java" contentType="text/html; charset=UTF-8"
2. pageEncoding="UTF-8"%>
3. <table class="page">
4. <td>
5. <button type="button" key="1">首页</button>
6. <button type="button" <c:if test="${pageInfo.firstPage == true}">disabled</c:if> key="${pageInfo.prePage}">上一页</button>
7. <button type="button" <c:if test="${pageInfo.lastPage == true}">disabled</c:if>  key="${pageInfo.nextPage}">下一页</button>
8. <button type="button" key="${pageInfo.totalPage}">尾页</button>
9. <input  type="text"  class="page-no"  id="page-no" value="${pageInfo.pageNo}" />
10. <button type="button" class="zhuan">跳转</button>
11. 总记录条数${pageInfo.totalCount}条，当前${pageInfo.pageNo}/${pageInfo.totalPage}页，每页${pageInfo.pageSize}条数据
12. </td>
13. </table>
14. <script type="text/javascript">
15. $(function () {
16. $('.page .zhuan').click(function(){
17. if(parseInt($('#page-no').val())<=parseInt("${pageInfo.totalPage}")){
18. $('input[name="pageNo"]').val($('#page-no').val());
19. $("#tableList").submit();
20. }else{
21. alert("不能大于总页数");
22. }
23. });
24. $('.page button:not(.zhuan)').click(function(){
25. $('input[name="pageNo"]').val($(this).attr("key"));
26. $("#tableList").submit();
27. })
28. })
29. */\*function go(page) {*
30. window.location.href="${basePath}student?method=list&pageNo="+page;
31. }\*/
32. </script>

# 组合条件查询

组合条件：一个条件、两个条件、三个及其以上

输入几个条件，按照条件查询结果（满足查询的条件与）。

组合条件查询出来的结果是&&的关系

|  |
| --- |
| -- 按照ID查询  select \* from student where stuId = 1  -- 当前端录入学生姓名的时候  select \* from student where stuName like '%2%'  -- 当前端录入姓名与学号的时候 ，执行的如下SQL  select \* from student where stuName like '%2%' and stuNO like '%1%'  -- 通常情况下：字符串都按照like 模糊查询较多，数字：（=或者范围居多） |

StuDao 中 list方法

1. public PageInfo<Student> list(Student student, PageInfo<Student> pageInfo) throws SQLException{
2. QueryRunner queryRunner=new QueryRunner(PropertiesUtils.getDataSource());
3. String \_sql="";
4. List<Object> \_list=new ArrayList<Object>();
5. if(student.getStuId()!=null){
6. \_sql+="and stuid=?";
7. \_list.add(student.getStuId());
8. }
9. if(StringUtils.isNoneBlank(student.getStuName())){
10. \_sql+="and stuname=?";
11. \_list.add("%"+student.getStuName()+"%");
12. }
13. if(StringUtils.isNoneBlank(student.getStuNo())){
14. \_sql+="and stuno=?";
15. \_list.add("%"+student.getStuNo()+"%");
16. }
17. *//list转数组*
18. Object[] arr=new Object[\_list.size()];
19. for(int i=0;i<\_list.size();i++){
20. arr[i]=\_list.get(i);
21. }
22. String sql="select \*from student where 1=1 "+\_sql+"limit "+(pageInfo.getPageNo()-1)\*pageInfo.getPageSize()+","+pageInfo.getPageSize();
23. List<Student> list = queryRunner.query(sql, new BeanListHandler<>(Student.class),arr);
24. pageInfo.setList(list);
25. pageInfo.setTotalCount(this.count(student));
26. return pageInfo;
27. }

# 老师管理

1. 列表查询
2. 新增
3. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成

1. 组合条件查询
2. 分页

# 课程管理

1. 列表查询
2. 新增
3. 修改

修改分两步：1）先查询，跳转到待修改页面，2）修改信息，然后提交完成

1. 组合条件查询
2. 分页

**查询条件：课程ID、课程名称、老师姓名**

**列表中的数据来源于course表与teacher表**

|  |  |  |
| --- | --- | --- |
| 课程ID | 课程名称 | 老师姓名 |
| 1 | Java编程 | 张平1 |
| 2 | HTML编程 | 张平1 |

课程与老师的关系

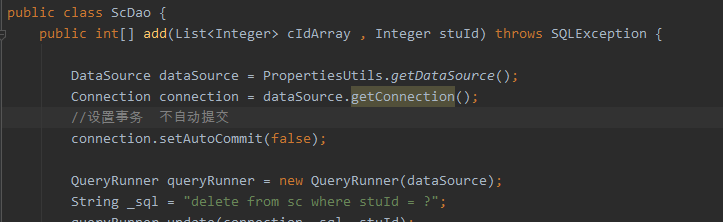
N:1

主外键关系：

主表（Teacher 1）、从表（Course：N）

# 学生角色

1. 登录
2. 学生选课



选课前 把数据先清空

# 老师角色

1. 登录
2. 评分
   1. 先查询自己所教的课程(管理员已经设置过了)
   2. 查询所教课程下的学生（学生角色选课的过程，学生登录后做选课）
   3. 最后录入评分,提交即可（老师角色评分即可）

# 查询统计

1. 数据录入
2. 查询统计
   1. **管理员-成绩区间统计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 科目 | <60） | [60-70] | (70-85] | (85-100] |
| Java | 1 | 10 | 5 | 6 |
| Html | 1 | 10 | 5 | 6 |
| Javascript | 1 | 10 | 5 | 6 |

* 1. **管理员-及格率**

|  |  |  |
| --- | --- | --- |
| 科目 | 及格率 | 参加考试人数 |
| Java | 72% | 10 |
| Html | 75% | 10 |
| Javascript | 84% | 10 |

* 1. **老师-成绩区间统计(所教科目)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 科目 | <60 | 60-70 | 70-85 | 85-100 |
| Java | 1 | 10 | 5 | 6 |

* 1. **老师-及格率**

|  |  |  |
| --- | --- | --- |
| 科目 | 及格率 | 参加考试人数 |
| Java | 72% | 10 |

* 1. **老师/学生-排名（共有查询）**
  2. **学生-自己分数**

# 报表统计

图表只是数据的一种更为直观的展示形式而已，一般要求及时没有图表，数据表格的统计也是要有的，图表都是给领导看的。

**下载apache echarts 文件zip**

<https://github.com/apache/incubator-echarts/tree/4.7.0/dist>

拷贝dist目录到项目中

使用引入*echarts.min.js*

|  |
| --- |
| <script type=*"text/javascript"* src=*"*${basePath}*static/echarts/echarts.min.js"*></script> |

