**《专业技能训练—数据库应用系统开发》**

**报告**

**课程设计题目：学生选课管理信息系统**

目录

[第一章 概述 1](#_Toc502578881)

[1.1 项目背景 1](#_Toc502578882)

[1.2 编写目的 1](#_Toc502578883)

[1.3 软件定义 1](#_Toc502578884)

[1.4 开发环境 2](#_Toc502578885)

[第二章 需求分析 2](#_Toc502578886)

[1.1 信息要求 2](#_Toc502578887)

[1.2 处理要求 4](#_Toc502578888)

[1.3 安全性与完整性要求 4](#_Toc502578889)

[1.3.1 安全性要求 4](#_Toc502578890)

[1.3.2 完整性要求 4](#_Toc502578891)

[第三章 概念结构设计 5](#_Toc502578892)

[3.1 E-R图 5](#_Toc502578893)

[第四章 逻辑结构设计 6](#_Toc502578894)

[4.1 关系表 6](#_Toc502578895)

[4.2 数据库 6](#_Toc502578896)

[第五章 主要模块的设计以及代码 7](#_Toc502578897)

[5.1 主要模块的设计思路、关键代码及其解释 7](#_Toc502578898)

[5.1.1 开课模块 7](#_Toc502578899)

[5.1.2 选课模块 10](#_Toc502578900)

[5.1.3成绩录入模块 13](#_Toc502578901)

[5.2前台软件与后代数据库的链接操纵过程 16](#_Toc502578902)

[第六章 总结 17](#_Toc502578903)

[6.1 所遇问题以及解决方法 17](#_Toc502578904)

[6.2 体会 17](#_Toc502578905)

[参考文献 19](#_Toc502578906)

1. 概述

1.1 项目背景

1. 项目的名称

数据库应用系统开发

1. 项目产品的名称

学生选课信息管理系统

1. 项目的开发者

王子荣

1. 项目的使用者

高校的学生、教师、以及教务管理人员

1.2 编写目的

随着招数人数的逐年增长，高校的教务信息文件也逐年累积，这使得教务人员的管理工作变得十分繁重、难以开展，导致效率很低，特别是在学生选课信息的管理这一方面，远远不能满足现实需要，为了改变现状，提高教务人员管理教务信息的效率，同时方便学生和教师对相关信息的操作，需要开发一套数字化信息管理软件，即学生选课信息管理系统，用于提高教务管理的效率，实现无纸化的办公。

1.3 软件定义

1. 学生选课信息管理系统

一个Web应用程序，主要用于学生选课的教务信息管理系统，学生登录后可以实现选课、查看成绩等操作，教师登录后可以实现查看开课班级、录入成绩等操作，教务人员登录后可以实现更高级别的操作。

* 1. 开发环境

1. 操作系统

Windows7旗舰版(64位)

1. 开发软件

JetBrains PhpStorm 2016.3.3

1. 数据库

MySQL5.6

1. 数据库可视化管理工具

Navicat Premium、phpMyadmin

1. IDE

AppServ

1. 浏览器

360安全浏览器、Google Chrome

第二章 需求分析

1.1 信息要求

1. 学生信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | 名字 | 类型 | 排序规则 | 属性 | 空 | 默认 | 额外 |
| 1 | studentNo | char(12) | utf8\_general\_ci |  | 否 | 无 |  |
| 2 | studentName | varchar(30) | utf8\_general\_ci |  | 否 | 无 |  |
| 3 | sex | enum('男', '女') | utf8\_general\_ci |  | 是 | NULL |  |
| 4 | birthday | date |  |  | 是 | NULL |  |
| 5 | phoneNumber | varchar(13) | utf8\_general\_ci |  | 是 | NULL |  |
| 6 | province | varchar(20) | utf8\_general\_ci |  | 是 | NULL |  |
| 7 | city | varchar(20) | utf8\_general\_ci |  | 是 | NULL |  |
| 8 | street | varchar(30) | utf8\_general\_ci |  | 是 | NULL |  |
| 9 | classNo | char(8) | utf8\_general\_ci |  | 是 | NULL |  |
| 10 | password | char(20) | utf8\_general\_ci |  | 否 | 123456 |  |
| 11 | email | varchar(30) | utf8\_general\_ci |  | 是 | NULL |  |

表2-1 学生表

1. 课程信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | 名字 | 类型 | 排序规则 | 属性 | 空 | 默认 | 额外 |
| 1 | courseNo | varchar(6) | utf8\_general\_ci |  | 否 | 无 |  |
| 2 | courseName | varchar(30) | utf8\_general\_ci |  | 否 | 无 |  |
| 3 | creditHour | smallint(6) |  |  | 是 | NULL |  |
| 4 | courseHour | smallint(6) |  |  | 是 | NULL |  |
| 5 | instituteNo | char(2) | utf8\_general\_ci |  | 是 | NULL |  |

表2-2 课程表

1. 开课班信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | 名字 | 类型 | 排序规则 | 属性 | 空 | 默认 | 额外 |
| 1 | courseClassNo | char(8) | utf8\_general\_ci |  | 否 | 无 |  |
| 2 | courseNo | char(6) | utf8\_general\_ci |  | 否 | 无 |  |
| 3 | courseTeacherNo | char(5) | utf8\_general\_ci |  | 是 | NULL |  |
| 4 | year | int(11) |  |  | 是 | NULL |  |
| 5 | semester | enum('2', '1') | utf8\_general\_ci |  | 是 | NULL |  |
| 6 | capacity | int(11) |  |  | 是 | NULL |  |
| 7 | enrollNumber | int(11) |  |  | 是 | NULL |  |

表2-3 开课班表

1.2 处理要求

1. 时间要求

网页的响应时间不超过2秒，支持多人同时访问

1. 空间要求

能够存储足够的教务信息

1. 精度要求

学生成绩精确到小数点后1位

1.3 安全性与完整性要求

1.3.1 安全性要求

1. 登录要求

用户输入账号密码登录，信息经过合法性检验后，与数据库中的信息进行匹配，若存在此用户，则允许登录，跳转到指定用户页面，否则禁止登录。

1. 恶意登录处理

同一IP地址，如果密码错误超过3次，则需要输入验证码才能再次登录。

1. 密码找回加密处理

通过邮箱发送找回密码的链接，点击后可以修改用户密码，其中密码的传输经过MD5加密处理。

1.3.2 完整性要求

1. 实体完整性

学院号：不为空且唯一，长度2

班级号：不为空且唯一，长度8，前4位代表年级

课程号：不为空且唯一，长度6，前2位代表所属学院，最后1位代表学分

教师号：不为空且唯一，长度5

开课班号：不为空且唯一，长度8，前4位代表年级，第5位代表学期，

学号：不为空且唯一

1. 参照完整性

班级表：外键为学院号

学生表：外键为班级号

课程表：外键为学院号

教师表：外键为学院号和课程号

开课班表：外键课程号和教师号

1. 用户自定义完整性

学生性别：男或女

课程学分：1~6

学期：1或2

限选人数：不超过3位数

第三章 概念结构设计

3.1 E-R图

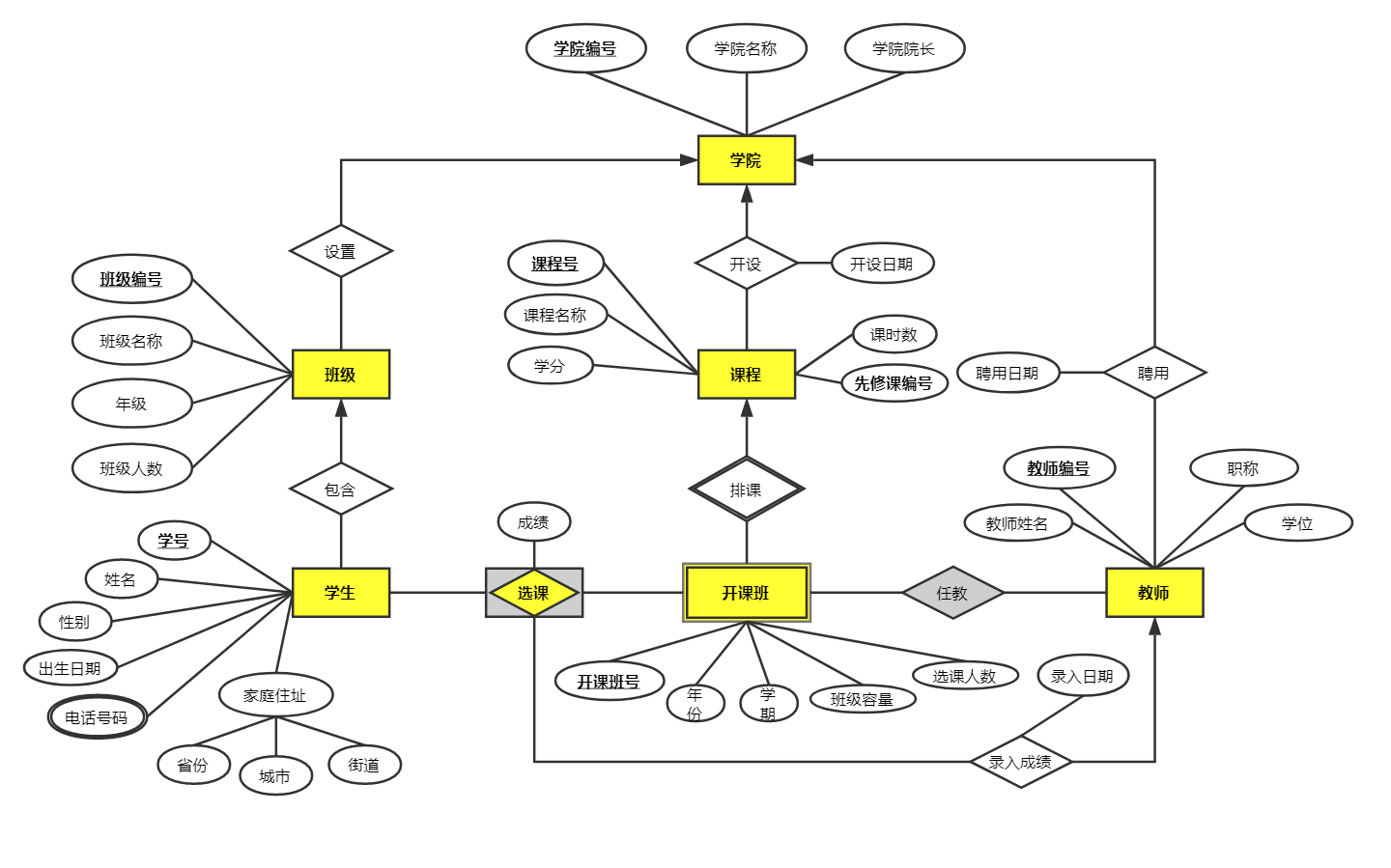


图3-1 E-R图

第四章 逻辑结构设计

4.1 关系表

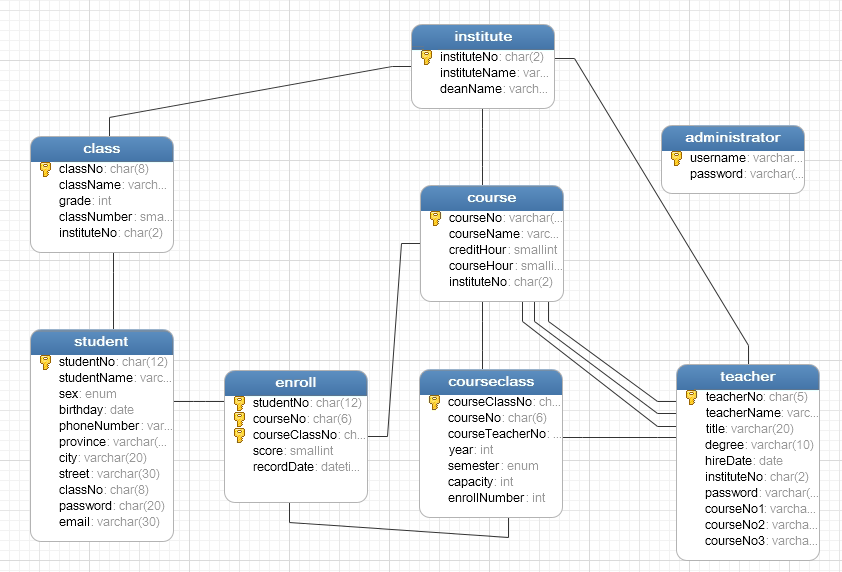


图4-1 关系表

4.2 数据库

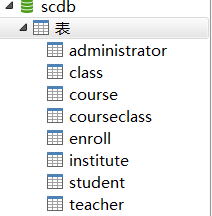


图4-2 数据库

第五章 主要模块的设计以及代码

5.1 主要模块的设计思路、关键代码及其解释

5.1.1 开课模块

1. 设计思路

* 管理员登录
* 进入开课界面
* 选择指定课程
* 填写关键信息，新增开课班

1. 关键代码及其解释
2. 前端界面显示代码

<div id="add-courseclass" class="panel panel-info"> //开课管理模块区

<div id="all-course-info" class="panel panel-primary" style="width:49%;float: left"> //可选课程区

<div class="panel-heading" >

<h4>可选课程</h4>

</div>

<div class="table-responsive"> //课程信息

//id=”table”引用了bootstrap-table的css和js文件，获取指定数据库信息进行输出

<table id="table" data-search="true" data-show-pagination-switch="true"

data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

<div id="toolbar"> //工具栏

<div class="btn-group" role="group">

<button id="add" type="button" class="btn btn-success">

<span class="glyphicon glyphicon-plus" aria-hidden="true"></span> 加入

</button>

</div>

</div>

</div>

<div id="courseclass-info" class="panel panel-primary" style="width:49%;float: right" > //开课班区

<div class="panel-heading" >

<h4>开课班</h4>

</div>

<div class="table-responsive"> //开课班信息

// id=”table2”引用了bootstrap-table的css和js文件，获取指定数据库信息进行输出

<table id="table2" data-search="true" data-show-pagination-switch="true"

data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

<div id="toolbar2"> //工具栏

<div class="btn-group" role="group">

<button id="open" type="button" class="btn btn-success">

<span class="glyphicon glyphicon-thumbs-up" aria-hidden="true"></span> 开放

</button>

<button id="mod" type="button" class="btn btn-warning">

<span class="glyphicon glyphicon-edit" aria-hidden="true"></span> 修改

</button>

<button id="del" type="button" class="btn btn-danger ">

<span class="glyphicon glyphicon-trash" aria-hidden="true"></span> 删除

</button>

</div>

</div>

</div>

</div>

效果截图：



图5-1 开课界面

1. 后端数据库处理代码

//增加行数据

if($\_POST['action']==="add"){ //获取提交数据的动作状态

try{

$arr=$\_POST['arr'];

$sql="insert into courseclass values('$arr[0]','$arr[1]','$arr[2]','$arr[3]','$arr[4]','$arr[5]','$arr[6]')";

$pds=$pdo->exec($sql);

echo $pds!=""?true:false;

}catch (Exception $exc){

echo $exc->getMessage();

}

exit;

}

//删除行数据

if($\_POST['action']==="del"){

$courseClassNos=$\_POST['courseClassNos'];

try{

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->beginTransaction(); //开启事务

foreach ($courseClassNos as $oldNo){

$sql="delete from courseclass where courseClassNo='$oldNo'";

$pds=$pdo->exec($sql);

}

$pdo->commit(); //事务提交

echo true;

}catch (Exception $exc){

$pdo->rollBack(); //事务回滚

echo false;

}

exit;

}

//修改行数据

if($\_POST['action']==="mod"){

try{

$arr=$\_POST['arr'];

$oldPk=$\_POST['oldPk'];

$sql="update courseclass set courseClassNo = '$arr[0]' , courseNo = '$arr[1]' , courseTeacherNo = '$arr[2]' , year='$arr[3]',semester='$arr[4]',capacity='$arr[5]',enrollNumber='$arr[6]' where courseClassNo='$oldPk'";

$pds=$pdo->exec($sql);

echo $pds!=""?true:false;

}catch(Exception $exc){

echo $exc->getMessage();

}

exit;

}

try{

//查询开课班信息和对应课程信息以及任课教师信息

$teacherNo=$\_GET['teacherNo'];

if($teacherNo!=""){

$sql="select \* from courseclass,teacher,course WHERE course.courseNo=courseclass.courseNo AND courseclass.courseTeacherNo='$teacherNo' AND teacher.teacherNo='$teacherNo'";

$pds=$pdo->query($sql);

$rows=$pds->fetchAll(PDO::FETCH\_ASSOC);

}else{

$sql="select \* from courseclass,course,teacher WHERE courseclass.courseNo=course.courseNo AND courseclass.courseTeacherNo=teacher.teacherNo";

$pds=$pdo->query($sql);

$rows=$pds->fetchAll(PDO::FETCH\_ASSOC);

}

}catch (Exception $exc){

echo $exc->getMessage();

}

$pds=null;

$pdo=null;

echo json\_encode($rows); //转化为json数据进行输出

效果截图：

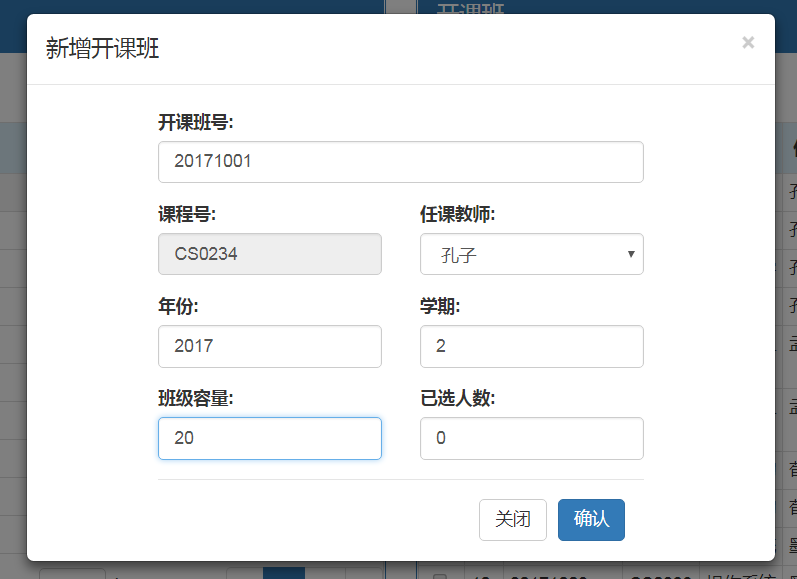


图5-2 确认开课

5.1.2 选课模块

1. 设计思路

* 学生登录
* 进入选课界面
* 选择课程并提交

1. 关键代码及其解释
2. 前端界面显示代码

<div id="personal-info" class="panel panel-primary"> //个人选课区域

<div id="optional-course" class="panel panel-primary" style="width:65%;float: left"> //可选课程区

<div class="panel-heading" >

<h4>可选课程</h4>

</div>

<div class="table-responsive">

<table id="table" data-search="true" data-show-pagination-switch="true"

data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

<div id="toolbar"> //工具栏

<div class="btn-group" role="group">

<button id="sel" type="button" class="btn btn-success">

<span class="glyphicon glyphicon-hand-up" aria-hidden="true"></span> 选入

</button>

</div>

</div>

</div>

<div id="selected-course" class="panel panel-primary" style="width:34%;float: right" > //已选课程区

<div class="panel-heading" >

<h4>已选课程</h4>

</div>

<div class="table-responsive">

<table id="table2" data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

<div id="toolbar2"> //工具栏

<div class="btn-group" role="group">

<button id="del" type="button" class="btn btn-danger ">

<span class="glyphicon glyphicon-trash" aria-hidden="true"></span> 重选

</button>

</div>

</div>

</div>

</div>

效果截图：



图5-3 选课界面

1. 后端数据库处理代码

//学生选课相关操作

if($\_POST['action']==="select"){ //获取提交事件的动作状态

try{

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->beginTransaction(); //开启事务

$arr=$\_POST['arr'];

//查询开课班信息和对应课程信息

$sql="select \* from courseclass WHERE courseClassNo='$arr[0]'";

$pds=$pdo->query($sql);

$rows=$pds->fetchAll(PDO::FETCH\_ASSOC);

foreach ($rows as $row) {

if($row[enrollNumber]<$row[capacity]){ //已选人数<限选人数

$sql="update courseclass set enrollNumber=enrollNumber+1 where courseClassNo='$arr[0]'";

//更新开课班表

$pds=$pdo->exec($sql);

$sql="insert into enroll values('$arr[2]','$arr[1]','$arr[0]',null,null)"; //更新选课表

$pds=$pdo->exec($sql);

}

}

$pdo->commit(); //事务提交

echo true;

}catch (Exception $exc){

$pdo->rollBack(); //事务回滚

echo false;

}

exit;

}

5.1.3成绩录入模块

1. 设计思路

* 教师登录
* 查看开课班
* 选择学生，进行成绩录入

1. 关键代码及其解释
2. 前端界面显示代码

<div id="courseClass-info" class="panel panel-primary"> //开课班信息区

<div class="panel-heading" >

<h4>开课班</h4>

</div>

<div class="table-responsive">

//id=”table”引用了bootstrap-table的css和js文件

<table id="table" data-search="true" data-show-export="true"

data-show-columns="true" data-show-pagination-switch="true"

data-show-toggle="true" data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

</div>

<div id="enroll-info" class="panel panel-primary" style="display: none"> //开课班的学生信息区

<div class="panel-heading" >

<h4><span id="title">开课班学生信息</span></h4>

</div>

<div class="table-responsive">

//id=”table2”引用了bootstrap-table的css和js文件

<table id="table2" data-search="true" data-show-export="true"

data-show-columns="true" data-show-pagination-switch="true"

data-show-toggle="true" data-show-refresh="true" class="table-condensed">

<thead class="bg-info">

</thead>

</table>

</div>

<div id="toolbar2"> //工具栏

<div class="btn-group" role="group">

<button id="back" type="button" class="btn btn-default ">

<span class="glyphicon glyphicon-arrow-left" aria-hidden="true"></span> 返回

</button>

<button id="add" type="button" class="btn btn-success">

<span class="glyphicon glyphicon-plus" aria-hidden="true"></span> 增加

</button>

<button id="mod" type="button" class="btn btn-warning">

<span class="glyphicon glyphicon-edit" aria-hidden="true"></span> 修改

</button>

<button id="del" type="button" class="btn btn-danger ">

<span class="glyphicon glyphicon-trash" aria-hidden="true"></span> 删除

</button>

</div>

</div>

</div>

效果截图：



图5-4 开课班信息



图5-5 开课班详细信息

1. 后端数据库处理代码

//增加行数据

if($\_POST['action']==="add"){

try{

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->beginTransaction(); //开启事务

$arr=$\_POST['arr'];

$sql="insert into enroll values('$arr[0]','$arr[1]','$arr[2]',null,null)"; //更新选课表

$pds=$pdo->exec($sql);

$sql="update courseclass set enrollNumber=enrollNumber+1 where courseClassNo='$arr[2]'";

//更新开课班表

$pds=$pdo->exec($sql);

$pdo->commit(); //事务提交

echo true;

}catch(Exception $exc){

$pdo->rollBack(); //事务回滚

echo false;

}

exit;

}

//修改行数据

if($\_POST['action']==="mod"){

try{

$arr=$\_POST['arr'];

$sql="update enroll set score='$arr[3]',recordDate='$arr[4]' where studentNo='$arr[0]' AND courseNo='$arr[1]' AND courseClassNo='$arr[2]'";

$pds=$pdo->exec($sql);

echo $pds!=""?true:false;

}catch(Exception $exc){

echo $exc->getMessage();

}

exit;

}

//删除行数据

if($\_POST['action']==="del"){

try{

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->beginTransaction(); //开启事务

$arr=$\_POST['arr'];

$sql="delete from enroll where studentNo='$arr[0]' AND courseNo='$arr[1]' AND courseClassNo='$arr[2]'";

$pds=$pdo->exec($sql);

$sql="update courseclass set enrollNumber=enrollNumber-1 where courseClassNo='$arr[2]'"; //更新开课班表

$pds=$pdo->exec($sql);

echo $pds!=""?true:false;

$pdo->commit();

}catch (Exception $exc){

$pdo->rollBack();

echo $exc->getMessage();

}

exit;

}

//如果能够获取提交过来的班级号

if(isset($\_GET['courseClassNo'])){

//查询该开课班的学生信息

try{

$courseClassNo=$\_GET['courseClassNo'];

$sql="select \* from enroll,student,class,institute where class.instituteNo=institute.instituteNo

AND student.classNo = class.classNo AND enroll.courseClassNo = '$courseClassNo'

AND enroll.studentNo=student.studentNo";

$pds=$pdo->query($sql);

$rows=$pds->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode($rows);

}catch (Exception $exc){

echo $exc->getMessage();

}

exit;

}

try{

//查询个人选课信息及其相应课程信息

$studentNo=$\_GET['studentNo'];

$sql="select \* from enroll,course,teacher,courseclass,student where student.studentNo=enroll.studentNo AND enroll.studentNo = '$studentNo' AND enroll.courseNo=course.courseNo

AND teacher.teacherNo = courseclass.courseTeacherNo

AND courseclass.courseClassNo=enroll.courseClassNo";

$pds=$pdo->query($sql);

$rows=$pds->fetchAll(PDO::FETCH\_ASSOC);

}catch (Exception $exc){

echo $exc->getMessage();

}

$pds=null;

$pdo=null;

echo json\_encode($rows);

5.2前台软件与后代数据库的链接操纵过程

1. 连接数据库
2. 代码

$dsn="mysql:host=localhost:3306;dbname=scdb";

try{

$pdo=new PDO($dsn,'root','admin0714'); //创建PDO对象，建立服务器连接

}catch (Exception $exc){

echo $exc->getMessage();

}

1. 解释

任何需要对数据库进行操作的页面，都统一引用该文件代码与数据库进行连接，防止多次定义导致错误后难以发现，还方便后期修改数据库的连接。

1. 对数据库进行查找和更新操作

利用PDO的相关语句，对数据库进行操作，可用try-catch语句包含在内，能发现异常并作出适当的处理，对完整性要求，可开启事物，这都是PDO语句可完成的操作。

1. 终止连接
2. 代码

$pds=null;

$pdo=null;

echo json\_encode($rows);

1. 解释

操作结束后，主动终止与数据库的连接，避免对数据库过多连接导致错误的发生，通过转化查询结果为json格式的数据并结合目前互联网上流行的高效可用的框架bootstrap，可直接把数据输出到表格中。

第六章 总结

6.1 所遇问题以及解决方法

1. 字符乱码

问题描述：

获取数据库中的数据输出到网页后，呈现为乱码

解决办法：

修改MySQL配置，把服务器端和客户端的默认字符集统一设置为UTF8

1. SESSION数据丢失

问题描述：

在页面进行跳转之后，SESSION数据会丢失，导致新页面无法获取到旧页面的SESSION数据

解决办法：

在需要用到SESSION数据的页面的头部填写session\_start()函数

1. bootstrap框架异常

问题描述：

在使用bootstrap框架后，一些CSS或JS功能无法正常使用

解决办法：

调整css和js的引用顺序，遵循先CSS后JS的原则，jquery.js先于bootstrap.js

6.2 体会

经过这次专业技能训练，我学到了很多知识，其中，我学会Web服务器的配置和使用，能够在Shell中使用一些简单的命令进行文件操作，使用了一些当前比较流行的框架，例如：bootstrap和jquery，借助这些框架能够快速地完成网页设计的工作，还使用Dreamweaver、PhpStorm、Navicat Premiun等工具，能极大地提高效率，不仅如此，我还掌握了HTML、CSS、JavaScript、PHP等语言，由此可见，要完成一个Web信息管理系统，涉及到的知识领域很广，需合理规划项目设计的流程，这样做起来才有条不紊，尽管整个过程花费了不少的精力，但我这次做的学生选课信息管理系统还只是勉强及格，我认为还有许多地方不够好，一些细节的地方没有更好地完善，还有许多没有考虑到的地方，随着学习的深入，我发现要学习的知识真的很多，而我仅仅学了些基础，相信在以后的学习中，我能继续加强自己的能力，务必把项目设计做得更完美。

参考文献

[1] 万常选，廖国琼，吴京慧，等. 数据库系统原理与设计. 北京：清华大学出版，2009.

[2] 刘云生. 数据库系统分析与实现. 北京：清华大学出版社，2009.

[3] 孙建伶，林怀忠. 数据库原理与应用. 北京：高等教育出版社，2006.

[4] Garcia-Molina H，ULLman J D，Widom J. 数据库系统全书（Database Systems：the CompleteBook），岳丽华，杨冬青，龚育昌，等译. 北京：机械工业出版社，2003.

[5] 张海蕃，牟永敏. 软件工程导论. 北京：清华大学出版社，2013.

[6] 孙飞显，孙俊玲，马杰. MySQL数据库实用教程. 北京：清华大学出版社，2015.

[7] 郑阿奇. MySQL实用教程[M]. 电子工业出版社，2014.

[8] 深入浅出MySQL数据库开发，优化与管理维护[M]. 人民邮电出版社，2008.

[9] 朝阳，简. MySQL性能调优与架构设计[M]. 电子工业出版社，2009.

[10] Scbuartz B，Zaitsev P，Tkaebenko V. 高性能MySQL[J]. 2011.

[11] David Sklar，著.；安道，译. PHP学习手册/（美）. 北京：中国电力出版社，2017.5.

[12] 牟奇春，汪剑主编. PHP动态网站开发项目教程[专著]. 北京：人民邮电出版社，2017.01.

[13] 陆凯主编. PHP网站开发实用技术[专著]. 北京：人民邮电出版社，2016.06.

[14] 刘西杰，张婷著. HTML CSS JavaScript网页制作从入门到精通[专著]. 北京：人民邮电出版社，2016.07.

[15] 李晓斌等编著. PHP+MySQL+Dreamwea网站建设[专著]. 北京：机械工业出版社，2016.10.