

目录

学生课外活动成果统计系统.....	3
一、需求分析	3
二、概念结构设计	3
三、逻辑结构设计	3
a) 课外活动成果统计表 (achievement)	3
b) 课外活动表 (activity)	4
c) 管理员表 (admin)	4
d) 学生表 (student)	4
e) 学生账号表 (student_account)	4
f) 组队信息表 (student_team)	5
g) 指导老师表 (teacher)	5
h) 团队表 (team)	5
四、具体实现	5
a) 学生课外活动成果统计系统的一些方面的数据控制功能	5
b) 操作说明	5
1、登录、注册操作	5
2、查询操作	6
3、更新操作	6
4、添加操作	6
5、删除操作	6
6、初始化操作	6
c) 核心代码部分	7
1、初始化创建表部分代码	7
2、初始化插入数据部分代码	7
3、初始化创建触发器部分代码	7
4、登录功能	7
5、检查规范问题	8
6、注册功能	8
7、切换学生端和管理员端	8
8、查询功能	8
9、更新功能	8
10、插入功能	8
11、删除功能	9
12、记录报错信息	9
13、检查 URL 功能	9
14、获取当前登录信息功能	10
d) 界面部分	10
1、登录、注册界面	10
2、管理员界面	10
3、学生界面	11
4、查询结果界面	11

5、更改、添加、删除信息界面类似，这里以更改界面为例	11
五、总结.....	11

学生课外活动成果统计系统

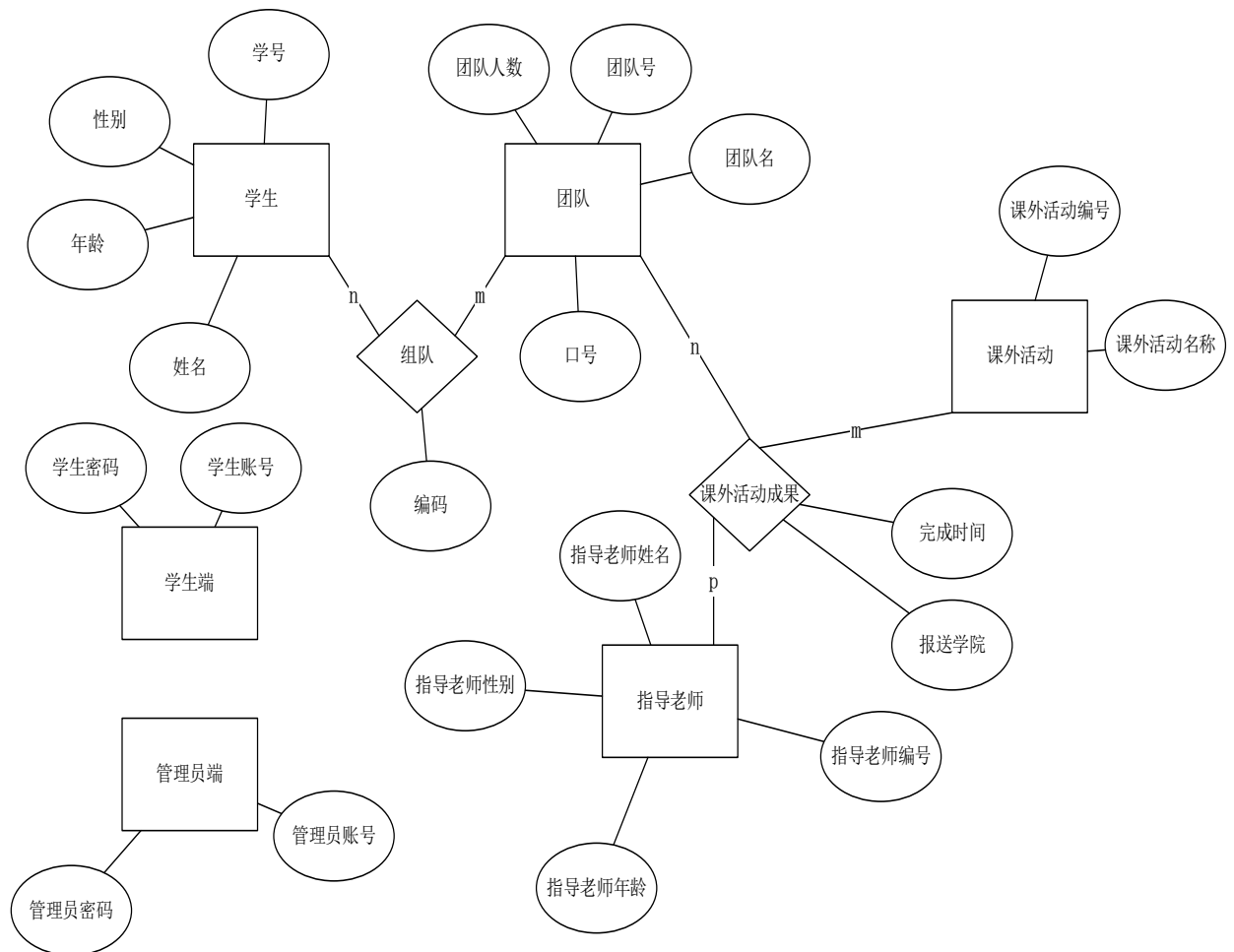
一、需求分析

随着当今大学生参加的课外活动越来越多，获得过的课外活动成果也越来越多，在实际工作中，如何提高统计这些数据的效率，如何统计可以更加方便、快捷，如何统计可以保证数据准确无误，如何在需要的时候更加快速地查找到精确数据，这些都是当今社会面临的很重要的问题。而建立一个学生课外活动成果统计系统，就是一个能够很好地解决这些问题的方法。

同时要解决以上这些问题，至少需要以下功能：

- 1、需要很好地呈现数据，系统要求有良好的人机界面
- 2、系统面向不同用户，不同用户的界面应该要求不同
- 3、原始数据增、删、查、改方便，支持多条件操作
- 4、显示出来的数据，具有合理性，便于查看
- 5、数据计算自动完成
- 6、数据在操作过程中，稳定性好

二、概念结构设计



三、逻辑结构设计

a) 课外活动成果统计表 (achievement)

属性名	含义	类型	说明
team_id	团队号	BIGINT	主键，不能为空，同时参照于团队表 (team) 的 id 属性
activity_id	课外活动编号	BIGINT	主键，不能为空，同时参照于课外活动表 (activity) 的 id 属性
time	完成时间	DATE	格式为“YYYY-MM-DD”
teacher_id	指导老师编号	BIGINT	外键，不能为空，同时参照于指导老师表 (teacher) 的 id 属性
academy	报送学院	VARCHAR (30)	不能为空

b) 课外活动表 (activity)

属性名	含义	类型	说明
id	课外活动编号	BIGINT	主键，不能为空
name	课外活动名称	VARCHAR (100)	不能为空

c) 管理员表 (admin)

属性名	含义	类型	说明
id	管理员账号	VARCHAR (30)	主键，不能为空
password	管理员密码	VARCHAR (30)	不能为空，默认为 password

d) 学生表 (student)

属性名	含义	类型	说明
id	学号	BIGINT	主键，不能为空
name	学生姓名	VARCHAR (20)	不能为空
sex	学生性别	CHAR (1)	不能为空，并且只能是‘男’或‘女’
age	学生年龄	INT	不能为空，并且大于 0，小于 120

e) 学生账号表 (student_account)

属性名	含义	类型	说明
id	学号	BIGINT	主键，不能为空，并且参照于学生表 (student) 的 id 属性

password	学生密码	VARCHAR (30)	不能为空，默认为 password
----------	------	--------------	-------------------

f) 组队信息表 (student_team)

属性名	含义	类型	说明
id	组队编号	INT	主键，不能为空，每次自增 1
student_id	学号	BIGINT	主键，不能为空，并且参照于学生表 (student) 的 id 属性
team_id	团队编号	BIGINT	主键，不能为空，并且参照于团队表 (team) 的 id 属性

g) 指导老师表 (teacher)

属性名	含义	类型	说明
id	指导老师编号	BIGINT	主键，不能为空
name	指导老师姓名	VARCHAR (20)	不能为空
sex	指导老师性别	CHAR (1)	不能为空，并且只能是‘男’或‘女’
age	指导老师年龄	INT	不能为空，并且大于 0，小于 120

h) 团队表 (team)

属性名	含义	类型	说明
id	团队编号	BIGINT	主键，不能为空
name	团队名	VARCHAR (30)	不能为空
num	团队人数	INT	不能为空
slogan	团队口号	VARCHAR (100)	不能为空

四、具体实现

a) 学生课外活动成果统计系统的一些方面的数据控制功能

- 1、采用学生端和管理员端两端等方式，提供数据的安全性保护。
- 2、使用在增删改查操作前先满足数据是否合理等基本要求，以及增加一些触发器等方式，提供数据的完整性检查，将数据控制在有效的范围内，并保持数据之间满足一定的关系。
- 3、选用 MySQL 的 InnoDB 存储引擎，其存储引擎采用两段锁协议，会根据隔离级别在需要的时候自动加锁，并且所有的锁都是在同一时刻被释放，满足数据库并发控制的情况。
- 4、在遇到一些特殊的数据库破坏的情况，可以使用管理员端的恢复数据的功能进行恢复。

b) 操作说明

- 1、登录、注册操作

1. 登录：直接在登录界面输入对应的账号密码，并且无论是学生账号还是管理员账号，都可以在登录界面进行登录，后台会根据账号类型，进行页面的切换，学生账号进入学生端，管理员账号进入管理员端。
2. 注册
 - a) 注册学生账号

注册学生账号至少需要用户名，姓名，性别和年龄，密码不填写的话，默认为 password，如果注册失败，会给出相应的提示信息，例如“注册失败，请更换用户名重试”等提示信息。
 - b) 注册管理员账号

注册管理员账号只需要用户名和密码，密码不填写，默认也是 password，错误提示信息与注册学生账号时类似。
- 2、 查询操作

查询操作有查询学生，查询老师，查询团队，查询活动，查询成果和查询组队信息六个功能，每个查询功能操作类似，将以其中一个进行描述。

例如查询学生信息，可以通过学号，姓名，性别，年龄四个中的任意组合都可以查询到对应的学生信息。

提交按钮则是提交给后台，查询对应的信息，如果查询到相应的结果，则会以表格的形式呈现。重置按钮可以将输入框中的数据全部清除。

表格内容也会相对于查找的要求进行相应的优化，例如在查询团队的时候，输入对应的团队号查询后，在输出查询结果的时候，会将对应的团队名一起输出。
- 3、 更新操作

更新操作有更新学生信息，更新老师信息，更新团队信息，更新活动信息，更新成果信息，更新组队信息六个功能，每个更新功能操作也类似，将以其中一个进行描述。

例如更新学生信息操作，首先通过第一个表单，学号，姓名，性别和年龄的任意组合确定需要更新的信息，在通过新学号，新姓名，新性别和新年龄的任意组合进行更新，当更新成功与否，都会有相应的信息提示出来。

提交按钮和重置按钮与查询操作中的相同，将不再赘述。
- 4、 添加操作

添加操作包括添加学生信息、添加老师信息、添加团队信息、添加活动信息、添加成果信息和添加组队信息。

以添加学生信息为例，输入想要添加的对应的学号，姓名，性别和年龄，然后系统进行验证，然后再进行添加操作，添加的成功与否也会有相应的信息提示出来。
- 5、 删除操作

删除操作包括删除学生信息、删除老师信息、删除团队信息、删除活动信息、删除成果信息和删除组队信息。

以删除学生信息为例，输入想要删除的对应的学号，姓名，性别和年龄的任意组合，锁定数据库中的数据，进行删除操作，删除的成功与否也会有相应的信息提示出来。
- 6、 初始化操作

在管理员端的个人信息界面，提供数据库格式化的操作，可以将数据恢复到前一段时间的状态。

c) 核心代码部分

- 1、初始化创建表部分代码：使用 SQL 语句以及表名变量，直接创建表，以及一些属性，和属性的一些要求（不能为空、属性类型、主键、外键等）。

```
96 CREATE TABLE $student_team_table
97 (
98     `id` INT NOT NULL AUTO_INCREMENT ,
99     `student_id` BIGINT NOT NULL ,
100     `team_id` BIGINT NOT NULL ,
101     PRIMARY KEY (`id`, `student_id`, `team_id`) ,
102     FOREIGN KEY (`student_id`) REFERENCES $student_table(`id`) ON DELETE CASCADE ON UPDATE CASCADE,
103     FOREIGN KEY (`team_id`) REFERENCES $team_table(`id`) ON DELETE CASCADE ON UPDATE CASCADE
104 )
105 ENGINE = InnoDB;
```

- 2、初始化插入数据部分代码：使用 SQL 语句以及表名变量，给表插入数据。

```
360 INSERT
361 INTO $team_table(`id`, `name`, `num`, `slogan`)
362 VALUES
363 ("1000", "光速队", "4", "单量铺路，金额致富!"),
364 ("1001", "出单队", "3", "我出单！我喜欢!"),
365 ("1002", "精英队", "5", "精领天下，英才神话!"),
366 ("1003", "超能勇士队", "4", "超越自我，超越梦想！还有一个团结"),
367 ("1004", "跃起动力队", "5", "让激情在动力中飞跃现在!"),
368 ("1005", "野狼盟队", "4", "我能!"),
369 ("1006", "胜羽队", "6", "胜利的羽毛永远归顺于我!");
```

- 3、初始化创建触发器部分代码：指定创建一个触发器的名称，在什么情况下，以及是语句型还是行型，然后再给出定义，这个触发器需要做哪些工作。

```
855 CREATE TRIGGER `Team_num_update` AFTER UPDATE ON `student_team`
856     FOR EACH ROW BEGIN
857     UPDATE
858         team
859     SET
860         num = num + 1
861     WHERE NEW.team_id = id;
862     UPDATE
863         team
864     SET
865         num = num - 1
866     WHERE OLD.team_id = id;
867     END;
```

- 4、登录功能：先在学生账号表中，查找是否与之匹配的信息，如果有，则登录到学生端，如果没有，再去查找管理员账号中查找，如果有，则登录到管理员端，如果没有就提示登录失败信息。

```
15 //进行登录
16 $login_user = get_user();
17 $login_password = get_password();
18 if ($login_user != "" && $login_password != "") {
19     $find_bool = find($link, $login_user, $login_password);
20     if ($find_bool == true) { //如果在学生端查找成功 ...
21     } else {
22         $find_admin_bool = find_admin($link, $login_user, $login_password);
23         if ($find_admin_bool == true) { //如果在管理员端查找成功 ...
24         }
25     } else { //如果都没有成功 ...
26     }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
```

- 5、检查规范问题：以注册为例，先检查学号是否为数字，以及名字长度是否符合规范，性别是不是只有男或女，还有年龄是否在指定范围内，只有都通过才能注册，否则输出对应的提示信息。

- 6、注册功能：将注册输入框中的数据获取出来，进行插入账号表操作，如果插入成功，则跳转登录界面，如果失败，输出相应的提示信息。

- 7、 切换学生端和管理员端:检测登录名中是否是管理员,如果是,则跳转管理端,否则跳转到学生端。

- 8、 查询功能：采用字符串拼接的方式，配合变量，可以整合出一套特定的 select 语句，无论是 where 部分，还是完成的 select 语句，然后再进行数据库查询，如果有结果，则输出对应表格信息，否则输出相应提示信息。

- 9、更新功能：采用字符串拼接的方式，配合变量，可以整合出一套特定的 update 语句，无论是 where 部分，还是完整的 update 语句，然后再进行数据库更新，如果有成功，则输出更新成功信息，否则输出相应提示信息。

- 10、 插入功能：采用字符串拼接的方式，配合变量，可以整合出一套特定的 insert 语句，无论是 where 部分，还是完整的 insert 语句，然后再进行数据库插入，如果有成功，则输出插入成功信息，否则输出相应提示信息。


```

156         if ($id != '' && $team_id != '') {
157             $values_section = <<<STRING
158             ("{$id}", "{$team_id}");
159             STRING;
160             $query = 'INSERT INTO ' . STUDENT_TEAM_TABLE . '(student_id, team_id) VALUES ' . $values_section;
161             $result = mysqli_real_query($link, $query);
162 >         if ($result == true) { ...
164 >         } else { ...
166     }

```

- 11、 删除功能：采用字符串拼接的方式，配合变量，可以整合出一套特定的 delete 语句，无论是 where 部分，还是完整的 delete 语句，然后再进行数据库删除，如果有成功，则输出删除成功信息，否则输出相应提示信息。

```

156     if ($id != '' || $name != '') {
157         $where_section = ($id == '' ? '' : (' AND id="' . $id . '"')) . ($name == '' ? '' : (' AND name="' . $name . '"'));
158         $where_section = 'WHERE ' . substr($where_section, 5);
159         $query = 'DELETE FROM ' . ACTIVITY_TABLE . ' ' . $where_section;
160         $result = mysqli_real_query($link, $query);
161
162 >         if ($result == true) { ...
164 >         } else { ...
166     }

```

- 12、 记录报错信息：根据输入框中的数据，采用特定编号的形式，避免产生很多种组合，而需要特定地去输出相应的内容，大大减少了重复代码量，以及提高了开发效率。

```

169     if ($id != '' || $name != '' || $sex != '' || $age != '') {
170         $index = 0;
171         if ($id != '') $index += 1000;
172         if ($name != '') $index += 100;
173         if ($sex != '') $index += 10;
174         if ($age != '') $index += 1;
175         $error_info = '';
176         if ($index / 1000 >= 1) { //记录是否是$id问题
177             $error_info = $error_info . ',id=\'' . $id . '\'';
178             $index -= 1000;
179         }
180 >         if ($index / 100 >= 1) { //记录是否是$name问题 ...
183         }
184 >         if ($index / 10 >= 1) { //记录是否是$sex问题 ...
187         }
188 >         if ($index >= 1) { //记录是否是$age问题 ...
191         }
192         $error_info = substr($error_info, 1);
193         //记录报错信息
194         $error_info = '在' . $error_info . '的条件下查询不到数据';

```

- 13、 检查 URL 功能：登录系统需要有用户名和密码，利用这两个点进行检查 URL 是否合理，然后还需要避免其中一些操作会变动 URL，只有两个条件都满足，才说明系统 URL 是合理的。

```

278 } else { //检测URL是否规范
279     $start = strpos($url_user_id, "user=") + 5;
280     $end = strpos($url_user_id, "&id=");
281     $name = substr($url_user_id, $start, $end - $start);
282     $start = strpos($url_user_id, "id=") + 3;
283     $id = substr($url_user_id, $start);
284     if ($name != "" && $id != "") {
285         $bool = true;
286     }
287 }

```

- 14、 获取当前登录信息功能：使用 JavaScript 检查 URL 中的姓名以及账号，然后操作 DOM，修改界面中的部分信息，达到获取当前登录的信息，也保证了在系统中操作的合理性以及完善性。

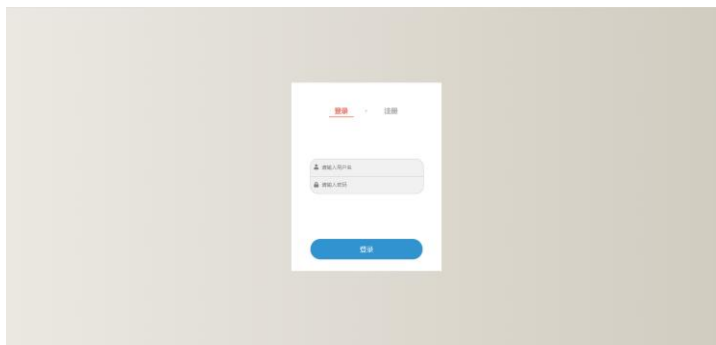
```

1 function addName() {
2     var name = UrlParm.parm("user");
3     var node_id_name = document.getElementById("name");
4     var node_name = document.createTextNode(name);
5     node_id_name.appendChild(node_name);
6
7     var id = UrlParm.parm("id");
8     var node_id_id = document.getElementById("id");
9     var node_id = document.createTextNode(id);
10    node_id_id.appendChild(node_id);
11 }
12
13 addLoadEvent(addName);

```

d) 界面部分

1、 登录、注册界面



2、 管理员界面



3、 学生界面



4、 查询结果界面



5、 更改、添加、删除信息界面类似，这里以更改界面为例



五、 总结

这一次的数据库课程设计我选用的是 HTML+CSS+JavaScript+PHP+MYSQL 组合，通过 HTML 来实现主要大体的框架，CSS 完成页面的展示部分，JavaScript 则操作一些 DOM，PHP 用来连接 MYSQL 数据库以及操作数据库。

整体来说，这次课程设计对我来说还是挑战蛮大的，没有选择那些已经很熟练的 C++ 和 Java 去完成界面和操作等问题，而是选择了全新的方面去开发页面，选择了全新的语言去操作数据库。

这几天一直坎坎坷坷，遇到了一些从来没有考虑过的问题，比如当时没有想到使用学生端和管理端分离的方法使整个系统更加完善，也没有及时想到先创建一个模板页面再通过修改模板页面来提高开发系统的效率，还有之前还直接用一些 SQL 语句来增删改查，后来才想到用变量和预加载的方式来生成动态的 SQL 语句。

不过也正是遇到了这些问题，才体现出课程设计的重要性，完成数据库的课程设计，不仅仅只是需要对数据库有很好的理解，还要有其他知识来进行补充。总的来说，有付出就会有收获，在这次课程设计期间，我收获满满，让我看到了我的不足，也明白了需要不断完善自己的重要性。