

## 一、实验目的：

- 1、了解 DBMS 系统的功能、Web based 数据库的软件组成及工具；
- 2、掌握数据库软件的使用方法、
- 3、掌握前台页面，后台数据库的基于 WEB 的数据库操作方法。

## 二、实验要求：

- 1、安装相关软件并浏览软件自带的帮助文件和功能菜单，
- 2、掌握 PHP 构建网页以及连接数据库的方法。
- 3、掌握 phpmyadmin 创建数据库与数据表的方法；
- 4、了解 mysql 的命令以及与 php 衔接的语句
- 5、了解网页构造的原理

## 三、实验设备：

计算机、数据库管理系统如 DB2, mysql 等软件。

## 四、实验内容

- 1) 按照实验指导讲义完成要求的练习题目。
- 2) 附上相关问题的解决方案，以及展示页面

本次实验笔者完成思路如下，审批辛苦了！

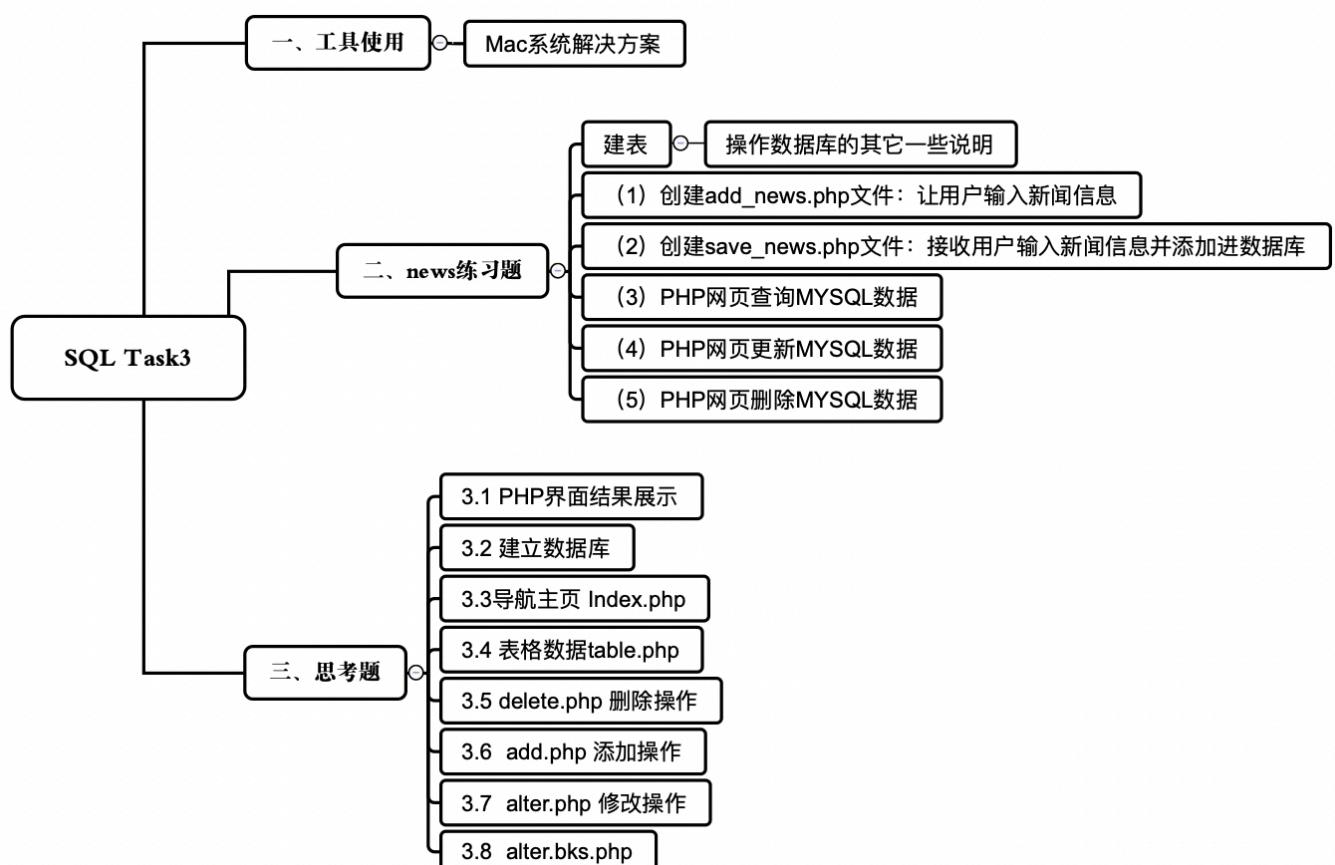


图 1:SQL Task3 思维导图

## 一、工具使用

要搭建 PHP 运行环境，有两种方式：第一种：分开安装 PHP 环境，需要服务器 Apache 或者 Nginx，数据库 MySQL 和 PHP。第二种：直接安装集成环境，服务器、数据库和 PHP 都是集成好了的。

因为笔者使用的是 mac 系统，原本想继续使用一直在用的 docker 按照第一种方式搭建 PHP 环境，但是这个过程需要配置一系列环境，因为网上教程很多不适用于新版 docker，笔者搞了很久都没有配置成功。于是搜索了很多解决方案，最后按照第二种方式，使用了 MAMP<sup>1</sup>这款工具成功配置好了 PHP 环境。

在安装了 MAMP 之后，打开却一直闪退，通过查找了很多资料，笔者才知道要将 mac 系统的 SIP 保护关闭才可以正常运行 MAMP，最后终于运行成功。同时，笔者将 Apache 端口设置为默认的 80，这样每次打开服务器就不用再输入端口了。

## 二、完成附件一实例教程，练习与该系统相关的操作

首先用 phpmyadmin 创建数据库，这一步已经很熟悉了，按照教程进行即可，表结构如图 2 所示。这里的自增长型类型在页面中被缩写为 A.I，在建表的时候输入即可。

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)	gb2312_chinese_ci		No	None		AUTO_INCREMENT	 Change  Drop  More
2	title	varchar(200)	gb2312_chinese_ci		No	None			 Change  Drop  More
3	add_time	varchar(100)	gb2312_chinese_ci		No	None			 Change  Drop  More
4	content	text	gb2312_chinese_ci		No	None			 Change  Drop  More

图 2:建表 (表格结构)

同时也可以使用类似“alter table news add reporter varchar(100) after title”的语句操作数据库，这里举例为插入变量 reporter，如图 3 所示

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)	gb2312_chinese_ci		No	None		AUTO_INCREMENT	 Change  Drop  More
2	title	varchar(200)	gb2312_chinese_ci		No	None			 Change  Drop  More
3	reporter	varchar(100)	gb2312_chinese_ci	Yes	NULL				 Change  Drop  More
4	add_time	varchar(100)	gb2312_chinese_ci		No	None			 Change  Drop  More
5	content	text	gb2312_chinese_ci		No	None			 Change  Drop  More

图 3: 操作数据库的其它一些说明 (举例)

接着通过“conn.php”文件实现 PHP 连接 MySQL 数据库，需要注意的是，笔者所使用的 mac 系统默认账户和密码都为 root。接着开始通过 PHP 网页向 MySQL 数据库添加数据，包括：

### (1) 创建 add\_news.php 文件：让用户输入新闻信息（图 4）



localhost/add\_news.php

请填写要添加新闻的信息

新闻标题

新闻内容

确定添加

<sup>1</sup> MAMP 内含 Apache 伺服器、PHP 安装套件以及 MySQL 安装套件，只要轻松点选就能安装架设网站的必备的元件，透过可视化界面的简单设定，就能在苹果电脑上架设自己专属的网站。

图 4: add\_news.php

(2) 创建 save\_news.php 文件: 接收用户输入新闻信息并添加进数据库 (图 5)



localhost/add\_news.php

新闻标题: EDG NB

新闻内容: EDG获得S11冠军! 骑士归来, 新王加冕!

确定添加

图 5: save\_news.php

点击确定添加后, 发现添加不了。笔者尝试了多种方法, 最后通过查找代码才发现在 conn.php 文件中, 需要将用户和密码都设置为初始密码 root。新闻添加成功如图 6 所示



localhost/save\_news.php

```
INSERT INTO news (title,content,add_time) VALUES ('EDG NB','EDG获得S11冠军!骑士归来, 新王加冕!', '2021-11-07')
```

图 6: 新闻添加成功

接着我们进去数据库进行验证, 可以看到图 7 所示, 数据库已经存在刚刚插入的记录。(这里的 ID 因为是自动生成的, 笔者因为前面测试了很多次, 所以这一次的插入生成的 id 是 6)



	+ Options	← →	▼	id	title	add_time	content	
		Edit		6	EDG NB	2021-11-07	EDG获得S11冠军!骑士归来, 新王加冕!	
		Check all	With selected:		Edit			

图 7: 到数据库进行验证, 添加记录成功

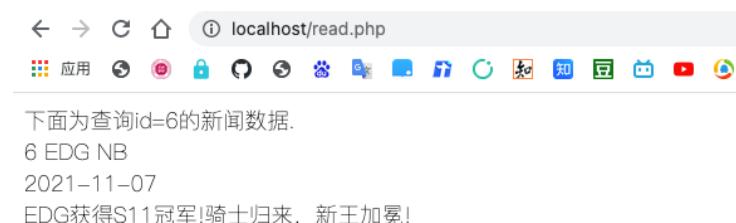
同样的方法, 我们再插入第 2 条记录, 如图 8 所示, 成功插入。

```
INSERT INTO news (title,content,add_time) VALUES ('立冬恰逢初雪, 北京号带您“云”赏雪后美景','皑皑白雪覆盖着这座城市, 凛冽寒风吹拂着大街小巷, 城市呈现出别样的景致。','2021-11-07')
```

图 8: 插入第二条记录

### (3) PHP 网页查询 MYSQL 数据

建立 read.php 文件实现普通单条数据查询, 如图 9 所示, 成功查询了 id=6 的新闻数据。



localhost/read.php

下面为查询id=6的新闻数据.

6 EDG NB

2021-11-07

EDG获得S11冠军!骑士归来, 新王加冕!

图 9: read.php 普通单条数据查询

建立 read\_2.php 实现结果不止一条数据的查询, 如图 10 所示, 成功查询了标题中含有“雪”这个字的新闻数据。



图 10: read\_2.php 结果不止一条数据的查询

#### (4) PHP 网页更新 MYSQL 数据

分为 3 个步骤。第 1 步:建立 news.php 文件,显示所有新闻让用户可选择修改 (图 11)



图 11:news.php 文件,显示所有新闻

第 2 步: 建立 edit\_news.php 文件,显示具体改的那条新闻的信息并输入改修内容 (这个文件用于接收从 news.php 传过来的 id 值,以 id 值确定要修改的新闻是那一条)。例如点击 id=8 的新闻, 进入到如图 12 所示的页面, 可以进行修改, 比如在标题中添加 (重复了) 这几个字。

请填写要修改新闻的信息

新闻标题 (重复了) 立冬恰逢初雪, 北京号  
新闻内容 皑皑白雪覆盖着这座城市, 凛冽寒风吹拂着大街小巷, 城市呈现出别样的景致。

确定修改

图 12: 新闻修改页面

第 3 步: 建立 save\_edit\_news.php 文件,用于把修改的内容更新到数据库。有了这个用于保存的文件, 点击图 12 的确定修改, 可以看到如图 13 所示的修改成功。



图 13: 新闻修改成功

进入数据库页面进行验证,从图 14 可以看到确实修改成功了,数据库已经更新了修改。

+ Options					
	← T →	↓	Id	Title	Add_time
<input type="checkbox"/>			6	EDG NB	2021-11-07
<input type="checkbox"/>			7	立冬恰逢初雪,北京号带您“云”赏雪后美景	2021-11-07
<input type="checkbox"/>			8	(重复了)立冬恰逢初雪,北京号带您“云”赏雪后美景	2021-11-07
<input type="checkbox"/>			9	号外	2021-11-07

图 14:修改后数据库验证

## (5) PHP 网页删除 MYSQL 数据

第 1 步, 在 news.php 文件基础上简单修改,显示所有新闻让用户选择删除 (图 15)。



图 15 新闻删除页面

第 2 步, 建立 del\_news.php 文件, 用于操作删除数据。点击删除 id=8 的新闻, 可以看到窗口提示删除成功。(图 16)



图 16: 新闻删除成功

再次回到新闻删除界面, 可以看到 id=8 的重复了的新闻已经被删除 (图 17)



图 17: 重复了的新闻已经被删除

### 三、针对思考题完成 PHP 和 MYSQL

#### 3.1 PHP 界面结果展示

如图 18、19 所示是针对思考题完成的 PHP 网站设计，包括导航页面和其中的子表。下面笔者讲对实验过程进行详细说明。



图 18 导航 index 页

	id	ename	city	操作
	e00	Amy	Vestal	<a href="#">修改</a> <a href="#">删除</a>
	e01	Bob	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
	e02	John	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
	e03	Lisa	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
	e04	Matt	Vestal	<a href="#">修改</a> <a href="#">删除</a>

共 5 条记录

图 19 表格 table 页

#### 3.2 建立数据库

根据指导手册的建表语句，建立数据库，如图 20 所示。需要特别注意的是，这里数据库的类型要设置为 InnoDB，不然在删除操作的时候会导致外键约束问题！

图 20: 公司数据库 (结构)

### 3.3 导航主页 Index.php<sup>2</sup>

**主页面设计：**笔者向 index 页面中添加“Welcome to Task3！”的标语，同时在该段文本下方放置表格，并设置所有元素居中放置。

**文本行设置：**在文本行前笔者添加了几行空格行<br></br>，经过不断调试最终确定加入一行空格行，使文本行位置更醒目。

```
<p>
<?php for($i=0; $i<1; $i++) {
    echo "<br> </br>";
}>
</p>

<p class="serif"> Welcome to Task3!</p>
```

**表格获取：**首先要连接数据库，引用"conn.php"文件，连接成功后获取公司数据库中所有表名的数据，如果数据库中没有表则显示“no tables”。表格表头定义为“Database”。定义变量\$res 接收数据表数量的信息，定义变量\$dbrow 接收数据表名称信息。之后利用循环逐条显示表名称。数据格中定义链接，链接名为具体的表名，链接进入到 table.php 文件中。

```
<table border="2" >
<?php
include("conn.php");

$res = mysqli_query($con, "show tables from $database");
if(!$res) die("no tables");

echo '<tr><td>公司数据表</td></tr>';
$row = mysqli_num_rows($res);
for($i=0; $i<$row; $i++) {
    $dbrow = mysqli_fetch_array($res);
    $tableName = $dbrow[0];
    $tr = '<a href="table.php?tableName=' . $tableName . '">' . $tableName . '</a>';
    echo "<tr><td>" . $tr . "</td></tr>";
}
?>
</table>
```

**CSS 美化：**（最后的效果图如上图 18 所示）

**(1) 背景图片设置：**笔者选择图片名为“index.jpg”的图片作为 index 页面的背景，因为想要这一张图片铺满页面，宽度高度都设置为 100%，背景大小设置为 cover。

**(2) 表格设置：**笔者将整张表格居中显示，字体大小设置为 60px，高度设置为 300px，宽度设置为 400px，经不断调试，这种大小比例和布局的表格更有视觉冲击性。并设置数据居中显示。

**(3) 文本设置：**笔者将这段文本的字体设置为 Times New Roman 格式，字体大小 70px，文本居中显示。

### 3.4 表格数据 table.php

在主页的公司数据表中点击链接将进入 table 页面并显示对应的数据表，table 页面由 3 个部分组成：(1) 数据表名称 (2) 数据表内容 (3) 记录数。所有元素均居中显示。

**数据获取：**连接数据库，使用\$\_GET 函数读取数据表名称，若成功则用变量\$tableName 接收返回值，若不成功则显示"ERROR"并结束运行。

<sup>2</sup> 参考：[https://blog.csdn.net/weixin\\_44176696/article/details/109447042](https://blog.csdn.net/weixin_44176696/article/details/109447042)

---

```
<form action="add.php" method="GET">
<?php
if(!isset($_GET["tableName"])) die("ERROR");
include("conn.php");
$tableName = $_GET["tableName"];

$columns = array();
?>
```

**数据表名称段落：**若上一步成功获取数据表名，则使用 echo 函数输出数据表名称，同时，笔者在表名后加入“table”，表名就更易懂了。此外，笔者还在两个单词间插入了空格，这是为了避免单词过于紧密且和表格大小不搭配。

```
<p class="serif">
    <?php echo $tableName; ?>   Table
</p>
```

**表头：**表头内容需从数据库中读取，用 mysqli\_query 函数执行“"show columns from ". \$tableName”命令，若成功则用变量\$res 接收表头内容，失败则输出"no data",接着利用循环将表头内容显示在表格上。

此外还需要添加一个表头：操作列，这个表头自己设定的，无需从数据库中读取，最后只需要再运行一下“<td>操作</td>”。至此，表头内容全部确定完毕。笔者还对表头行进行适当美化，填充了黄色背景。

```
<table border="2">
<?php

$res = mysqli_query($con, "show columns from " . $tableName);
$count=mysqli_query($con, "select count(*) from " . $tableName);
$retCount = mysqli_fetch_array($count); //获取数量

$num = $retCount[0]?$retCount[0]:0;
if(!$res) die("no data");
$row = mysqli_num_rows($res);
echo '<tr style="background-color:yellow">';
for($i=0; $i<$row; $i++) {
    $dbrow = mysqli_fetch_array($res);
    echo '<td>' . $dbrow[0] . '</td>';
    array_push($columns, $dbrow[0]);
}
echo '<td>操作</td>';
echo '</tr>';
```

**打印数据：**利用 mysqli\_query 函数执行"select \* from " . \$tableName 语句，变量\$res 接收返回值;再利用函数 mysqli\_num\_rows 确定记录数，变量\$row 接收返回值。再使用变量\$dbrow 接收 mysqli\_fetch\_array 函数对\$res 解压缩后的数据，循环遍历输出每条数据。

操作列中的数据不是由数据库中读取，建立链接进入其他页面，操作列中有两个元素，分别是「修改」和「删除」，点击「修改」进入到 alter.php 页面，点击「删除」进入到 delete.php 页面

---

```
// 打印数据
$res = mysqli_query($con, "select * from " . $tableName);
$row = mysqli_num_rows($res);
for($i=0; $i<$row; $i++) {
    echo '<tr>';
    $dbrow = mysqli_fetch_array($res);
    for($j=0; $j<count($columns); $j++) {
        echo '<td>' . $dbrow[$columns[$j]] . '</td>';
    }

    echo '<td><a href="alter.php?tableName=' . $tableName . '&primaryKeyName=' . $columns[0] . '&primaryKeyValue=' . $dbrow[0] . '">修改</a> ';
    echo '<td><a href="delete.php?tableName=' . $tableName . '&primaryKeyName=' . $columns[0] . '&primaryKeyValue=' . $dbrow[0] . '">删除</a></td>';
    echo '</tr>';
}
```

**添加数据:** 该部分为 form 表单内容, 建立一个 form 表单, 指向"add.php"文件, 方法为 GET。设置和表头数量相同的输入标签<input></input>, 各个 input 名称和表头相对应, 在操作列设置一个 submit 提交键

```
// 添加数据
echo '<tr style="background-color:yellow">';
for($j=0; $j<count($columns); $j++) {
    echo '<td><input type="text" name=' . $columns[$j] . '" style="width:120;"></input></td>';
}
echo '<td><input type="submit" style="width:120;" value="添加记录"></td>';
echo '<input type="text" name="tableName" style="display:none;" value=' . $tableName . '></input>';
echo '</tr>';
```

**记录数:** 利用 mysqli\_query 函数执行查询数据表记录数, 设置变量\$count 接收返回值, 变量\$retCount 接收解压后的结果, \$num 是最终记录数。

```
$res = mysqli_query($con, "show columns from " . $tableName);
$count = mysqli_query($con, "select count(*) from " . $tableName);
$retCount = mysqli_fetch_array($count); // 获取数量
$num = $retCount[0] ? $retCount[0] : 0;
```

表格数据打印完成后建立一个段落来输出记录数, 定义类为 “record”, 输出 “共 \$num 条记录”, 文本间添加两个空格来突出显示记录数。

```
<p class="record">
    共 &nbsp <?php echo $num;?> &nbsp 条记录
</p>
```

**CSS:** (最后效果如上图 19 所示)

(1) 背景选择 ‘table.jpg’, 背景宽度设置为 100%, 高度为 110%。(这里笔者选择的图片不是特别大, 若宽度设置为 100%, 页面下方会有一条纯色背景, 不美观, 设置为 110% 恰好可以使得背景充分显示)

(2) 同时, 设置表格居中显示和数据格居中显示。

### 3.5 delete.php 删除操作

先连接数据库, 由于 delete.php 表单 form 和 table.php 连接, 利用\$\_GET 函数获取 table.php 文件中的数据表名称\$tableName, 以及要删除的记录。delete.php 根据传递过来的主键和主键值来确定哪一条记录要被删除。确定目标记录后执行 “delete from ... where ...” 操作, mysqli\_query 函数执行该条命令, 并用\$res 变量传递结果, 若操作成功, 输出‘删除成功, 请返回’, 若操作失败, 则输出‘删除失败, 请返回’。需要特别注意的是这里的 primaryKeyValue 是字符串类型, 所以在书写查询语句的时候要使用转义字符语法。

```

<?php
include("conn.php");
$tableName = $_GET["tableName"];
$primaryKeyName = $_GET["primaryKeyName"];
$primaryKeyValue = $_GET["primaryKeyValue"];
$query = 'delete from ' . $tableName . ' where ' . $primaryKeyName . '=' . $primaryKeyValue;
$res = mysqli_query($con, $query);
if($res)
    echo "<script>window.alert('删除成功,请返回');history.back(1);</script>";
else
    echo "<script>window.alert('删除失败,请返回');history.back(1);</script>";
?>

```

**实验结果:** 因为 employee 表 eid 在 purchase 表中已经被引号了, 由于 innodb 有外键约束, 所以是删除不了的, 因此笔者在这里通过自己在后面几个步骤中添加的记录 (eid = e05) 进行展示, 删除 eid = e05, 可以看到返回 “删除成功”

eid	ename	city	操作
e00	Amy	Vestal	修改 删除
e01	Bob	Binghamton	修改 删除
e02	John	Binghamton	修改 删除
e03	Lisa	Binghamton	修改 删除
e04	Matt	Vestal	修改 删除
e05	Ferry	Guangzhou	修改 删除

共 6 条记录

删除成功,请返回

图 21 删除成功效果

**网页端验证:** 网页端刷新之后可以看到 eid=e05 的记录已经消失, 总记录数变成了 5 条, 证明删除成功。

eid	ename	city	操作
e00	Amy	Vestal	修改 删除
e01	Bob	Binghamton	修改 删除
e02	John	Binghamton	修改 删除
e03	Lisa	Binghamton	修改 删除
e04	Matt	Vestal	修改 删除

共 5 条记录

图 22 删除成功网页端验证

**数据库验证:** 进入 phpmyadmin 后查看数据表, eid=e05 的记录已经消失, 证明网页端可以进行数据库删除操作

eid	ename	city
e00	Amy	Vestal
e01	Bob	Binghamton
e02	John	Binghamton
e03	Lisa	Binghamton
e04	Matt	Vestal

e05已经被删除了

图 23 删除成功数据库验证

### 3.6 add.php 增加操作

和 delete.php 文件类似，第一步先连接数据库，首先要清楚上一个页面要往哪个数据表插入记录，因此利用 `$_GET` 函数获取 `table.php` 的表明并传递给 `$tableName`。然后从表中获取字段名并存储在 `$columns` 中，确定了字段后再用 `$_GET` 函数获取具体记录值，并将返回值存入 SQL 插入语句——“`insert into... (values)`”。最后用 `mysqli_query` 函数执行该插入语句，操作结果返回给 `$res`。若操作成功，返回“添加成功，请返回”，操作失败则返回“添加失败，请返回”。

```
// 获取字段名并且构造sql语句
$query = 'insert into ' . $tableName . ' (' ;
$res = mysqli_query($con, "show columns from " . $tableName);
$row = mysqli_num_rows($res);
$columns = array(); // 记录列名称
for($i=0; $i<$row; $i++) {
    $cname = mysqli_fetch_array($res)[0];
    array_push($columns, $cname);
    $query = $query . $cname . ',';
}
$query = substr($query, 0, strlen($query)-1); // 去掉,号
$query = $query . ') values (' ;
for($i=0; $i<count($columns); $i++) {
    $cname = $columns[$i];
    $query = $query . "'" . $_GET[$cname] . "'";
}
$query = substr($query, 0, strlen($query)-1); // 去掉,号
$query = $query . ')';
```

效果图：在 employees Table 中添加一个 `eid = e05` 的记录，可以看到返回“添加成功”

The screenshot shows a web application interface. On the left, a table titled 'employees Table' displays five rows of data. The last row, with `eid` as 'e05', `ename` as 'Ferry', and `city` as 'Shenzhen', has a yellow background. A red box highlights this row. To the right, a modal dialog box is displayed with the URL 'http://localhost/employees&eid=e05&ename=Ferry&city=Shenzhen'. The dialog contains the message '添加成功,请返回' (Add successful, return) in a red box, and a blue '确定' (Confirm) button.

图 24 添加成功效果

网页端验证：刷新网页，新添加的记录出现在数据表中

employees Table			
eid	ename	city	操作
e00	Amy	Vestal	<a href="#">修改</a> <a href="#">删除</a>
e01	Bob	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
e02	John	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
e03	Lisa	Binghamton	<a href="#">修改</a> <a href="#">删除</a>
e04	Matt	Vestal	<a href="#">修改</a> <a href="#">删除</a>
e05	Ferry	Shenzhen	<a href="#">添加记录</a>

图 25 添加成功网页端验证

数据库验证：数据库中显示了新添加的记录

图 26 添加成功 数据库验证

### 3.7 alter.php 修改操作

修改界面放一段文本，输出“修改页”。之后添加修改表格，原数据不变，用户可以直接在页面进行修改。文本段和表格均居中放置。

**文本：**先加入 4 行空格行，使得文本和表格显示在页面正中央，之后加入文本，输出“修改页”。

```
<?php for($i=0; $i<4; $i++) {
    echo "<br> </br>";
}?>
<p class="serif">
    修改页
</p>
```

**表格：**先利用 `$_GET` 函数获取数据表名，主键和主键值。再利用 `mysqli_query` 函数执行命令“`show columns from ...`”获取字段名，之后就可以输出 `alter.php` 中表格的字段。最后一列再加上「操作」字段。

```
if(!isset($_GET['tableName'])) die("loss key infomation");
if(!isset($_GET['primaryKeyName'])) die("loss key infomation");
if(!isset($_GET['primaryKeyValue'])) die("loss key infomation");

$tableName = $_GET["tableName"];
$primaryKeyName = $_GET["primaryKeyName"];
$primaryKeyValue = $_GET["primaryKeyValue"];

// 打印表头
$columns = array();
$res = mysqli_query($con, "show columns from " . $tableName);
if(!$res) die("no data");
$row = mysqli_num_rows($res);
echo '<tr style="background-color:yellow">';
for($i=0; $i<$row; $i++) {
    $dbrow = mysqli_fetch_array($res);
    echo '<td>' . $dbrow[0] . '</td>';
    array_push($columns, $dbrow[0]);
}
echo '<td>操作</td>';
echo '</tr>';
```

接着还需要获取原记录，因为这个文件的功能是修改，不是添加，所以还需要查询原来的记录，前面已经获取了主键和主键值，因此通过这些信息可以执行 `select * from...where...` 命令获取原记录。如果查询不到则输出"not found"。将获取的结果解压缩后赋值给变量 `$data`，利用循环将原记录显示在表格中，这些数据是以 `input` 形式来呈现的，如果换成别的，用户就没有接口修改数据了。输出完原记录后，还要在操作列里添加 `Submit` 提交键，用户修改数据传递到"alter.bks.php"文件。

```

// 查询该记录
$res = mysqli_query($con, "select * from " . $tableName . ' where ' . $primaryKeyName . '=' . '\'' . $primaryKeyValue . '\'');
if(!$res) die("not found");
$data = mysqli_fetch_assoc($res);

// 修改的数据字段
echo '<tr>';
for($j=0; $j<count($columns); $j++)
{
    echo '<td><input type="text" name="' . $columns[$j] . '" value=' . $data[$columns[$j]] . '></input></td>';
}
echo '<td><input type="submit" style="width:120px; value="修改"></td>';
echo '</tr>';

```

CSS:

- (1) **背景设置:** 选用'table.jpg', 同 table.php 页面相同
- (2) **文本设置:** 文本字体为"Times New Roman"格式, 字体大小为 45px, 居中放置。
- (3) **表格设置:** 表格居中

**效果图:** 对上一步添加的 eid = e05 的记录进行修改操作, 跳转到修改页, 笔者将这条记录的 city 从原先的 shenzhen 修改为 guangzhou, 点击「修改」键, 返回“修改成功”



图 27 修改成功效果

**网页端验证:** 刷新网页, 可以看到刚刚修改的记录在数据表中已发生变化 (e05 的 city 变为 Guangzhou)

employees Table				
eid	ename	city	操作	
e00	Amy	Vestal	<a href="#">修改</a>	<a href="#">删除</a>
e01	Bob	Binghamton	<a href="#">修改</a>	<a href="#">删除</a>
e02	John	Binghamton	<a href="#">修改</a>	<a href="#">删除</a>
e03	Lisa	Binghamton	<a href="#">修改</a>	<a href="#">删除</a>
e04	Matt	Vestal	<a href="#">修改</a>	<a href="#">删除</a>
e05	Ferry	Guangzhou	<a href="#">修改</a>	<a href="#">删除</a>
			<a href="#">添加记录</a>	

图 28 修改成功网页端验证

**数据库验证:** 回到数据库, 也可以看到刚刚修改的记录已发生变化 (e05 的 city 变为 Guangzhou)

图 29 修改成功数据库验证

### 3.8 alter.bks.php

从 alter.php 文件中的 form 表单中获取要修改的记录信息，包括数据表名，主键和主键值之后构造 SQL 更新语句，更改数据库记录。更新语句需要表名、字段名和值，上一步都已经获取，构造语句“update... set where ...”并执行。操作成功返回'修改成功,请返回'，操作失败则返回'修改失败,请返回'。

```
// 获取字段名并且构造sql语句
$query = 'update ' . $tableName . ' set ';
$res = mysqli_query($con, "show columns from " . $tableName);
$row = mysqli_num_rows($res);
for($i=0; $i<$row; $i++) {
    $cname = mysqli_fetch_array($res)[0];
    $query = $query . $cname . '=' . $_GET[$cname] . ',';
}
$query = substr($query, 0, strlen($query)-1);
$query = $query . ' where ' . $primaryKeyName . '=' . '\'' . $primaryKeyValue . '\'';
```

## 五. 问题分析（碰到什么问题，如何解决）

**1、工具使用：**因为笔者使用的是 mac 系统，原本想继续使用一直在用的 docker 按照第一种方式搭建 PHP 环境，但是这个过程需要配置一系列环境，因为网上教程很多不适用于新版 docker，笔者搞了很久都没有配置成功。于是搜索了很多解决方案，最后按照第二种方式，使用了 MAMP 这款工具成功配置好了 PHP 环境。在安装了 MAMP 之后，打开却一直闪退，通过查找了很多资料，笔者才知道要将 mac 系统的 SIP 保护关闭才可以正常运行 MAMP，最后终于运行成功。同时，笔者将 Apache 端口设置为默认的 80，这样每次打开服务器就不用再输入端口了。

**2、连接数据库：**通过“conn.php”文件实现 PHP 连接 MYSQL 数据库时，笔者所使用的 mac 系统默认账户和密码都为 root，这跟实验教程给的空账号密码不一样，笔者通过查阅了很多资料才发现问题出在这个地方，通过修改账号密码解决了问题。

**3、删除操作：**本次实验最大的 bug 来自于对 delete.php 的编写，主要有两个问题。一个是因为主键是字符串类型，所以写查询语句的时候需要合理使用转义字符，\$primaryKeyName . '=' . '\'' . \$primaryKeyValue . '\'', 不然就会导致语句出错。第二个问题是外键约束问题，因为 employee 表 eid 在 purchase 表中已经被引号了，由于 innodb 有外键约束，所以是删除不了的。笔者在这两个地方纠结了很久，最后通过助教和老师的帮助解决了问题。

**4、PHP 语言：**本次实验还有挑战是对 PHP 语言的陌生，因此在部分语法问题上纠结花

---

了大量的时间，后面经过系统性地看书和查找博客借鉴学习，就很容易发现一些语法问题。比如在该数据库中，两个表格中的数据都具有一定的函数依赖关系，因此不能任意对表信息进行添加和修改，如不能在 employee 表中添加重复的 eid 信息，这就需要对添加的内容进行条件筛选，只有与表之前 eid 信息不重复的才能够添加成功。

## 六. 实验心得

本次实验通过使用 PHP 语言，连接了 HTML 与 MySQL，将 MySQL 的数据可视化，并通过 HTML 页面对 MySQL 的数据库进行修改。这不仅锻炼了我们对 MySQL 语言的熟练使用，还让我们对网页设计和内容呈现有了更深的理解。数据库内核的设计是根本。前端对数据库的展现也至关重要，能否充分利用数据库的功能，能否设计的稳定、便捷，决定了一个数据库究竟能发挥出多大的作用，以及他给用户最终带来的使用价值。

在数据输入和修改方面，首先需要找出数据库中表信息的函数依赖关系，然后在添加和修改的时候对函数依赖关系的条件加以限定，防止之后的操作破坏原表的函数依赖关系。但在真正的实践当中，我们还是首先应该对原数据进行分析，然后找出其中的函数依赖关系，对表进行合理的拆分。在设计页面时，应该将这些函数依赖关系的限定条件加入其中。因此，我们还十分有必要学习和掌握如何对表信息进行逻辑分析。

总体来看本次的实验，笔者花了挺多时间，最主要的还是对于知识的不熟悉，然后在具体的应用的过程当中，也遇到了挺多 bug。通过了这次的实验学习到了很多东西，但是还有很多地方可以改进，例如在网页界面的呈现上还是比较的简单。继续加油吧！

## 七. 诚信承诺

本组成员郑重承诺在项目实施的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本组成员承担。

签名：