

第 3 章

全程实例一：成绩查询系统

进行PHP网站开发的环境有很多，对于已经很熟悉HTML语言和PHP的设计人员甚至可以直接使用记事本进行代码的编写工作；对于新手来说可以使用Dreamweaver配合MySQL进行动态系统的开发。Dreamweaver提供了方便的图形化界面，但从PHP 7.0开始需要在编辑代码窗口中手动输入一些程序，才能够与MySQL数据库交互，实现建立数据、查询、新增记录、更新记录、删除记录等操作，实现PHP+MySQL动态系统的开发。本章将介绍如何使用Dreamweaver的服务器行为，引导读者掌握使用Dreamweaver开发PHP程序的逻辑方法。

本章的学习重点：

- 在Dreamweaver中进行PHP开发平台的搭建
- 搭建PHP成绩查询系统开发的平台
- 检查数据库记录的常见操作
- 编辑记录的常见操作

3.1 成绩查询系统环境

本节就以实例“成绩查询”系统的形式具体介绍Dreamweaver中的服务器行为的使用方法。在开始制作一个PHP网站之前，需要在Dreamweaver中定义一个新站点。在“新建站点”中可以显示现在网站的本地目录、测试的路径等信息。

3.1.1 查询系统设计

“成绩查询”系统结构主要分成用户登录入口与找回密码入口两个部分，其中index.php是这个网站的首页。

在本地的计算机设置站点服务器，除在Dreamweaver CC 2017的网站环境按F12键来浏览网页之外，还可以在浏览器输入“http://127.0.0.1/phpweb/index.php”来打开用户系统的首页index.php，其中phpweb为站点名。

本实例制作9个功能页面，各页面的功能如表3-1所示。

表3-1 网页功能表

页面	主要的功能
index.php	用来显示所有的成绩记录
conn.php	数据库连接代码
detail.php	显示详细成绩信息页面
add.php	增加成绩信息页面
update.php	更新成绩信息页面
del.php	删除成绩信息页面
saveupdate.php	更新调用的PHP代码
savedel.php	删除调用的PHP代码
saveadd.php	保存增加调用的PHP代码

(1) index.php: index.php用于浏览数据库内的记录，为detail.php提供附带URL参数ID的超级链接，便于查看详细的记录信息，如图3-1所示。



图 3-1 index.php 页面效果

(2) detail.php: detail.php用于接收由index.php传来的URL参数ID, 利用URL参数筛选数据库中的记录。更新与删除记录都是依据数据库中的关键字段ID来识别记录的, 如图3-2所示。



图 3-2 detail.php 页面效果

(3) update.php: update.php用于接收由detail.php传来的URL参数ID, 利用URL参数筛选数据库中的记录, 单击“更新”按钮调用saveupdate.php, 即可完成数据的更新并返回原网页, 制作后的效果如图3-3所示。



图 3-3 update.php 页面效果

(4) del.php: del.php用于接收由detail.php传来的URL参数ID, 利用URL参数筛选数据库中的记录, 单击“删除”按钮调用savedel.php, 即可完成数据的删除并返回首页, 制作后的效果如图3-4所示。



图 3-4 del.php 页面效果

当实现一个PHP系统功能时，提前规划网站的架构是一件很重要的事情。在我们头脑中对网站要有一个雏形，大概有哪些页面、页面间的关系如何等。数据库的架构规划也是一样的，要有哪些数据表、字段，如何跟网页配合等都是很重要的工作。

3.1.2 创建数据库

经过对前面功能的分析发现，数据库应该包括ID、姓名、年龄与成绩4个字段，所以在数据库中必须包含一个容纳上述信息的表，将数据库命名为phpweb，下面就要使用phpMyAdmin软件建立网站数据库websql作为任何数据查询、新增、修改与删除的后端支持。

创建的步骤如下：

步骤01 在浏览器中输入http://127.0.0.1/phpmyadmin/,输入MySQL的用户名和密码(xampp默认环境下可以直接登录)。

步骤02 单击“执行”按钮即可进入软件的管理界面，选择相关数据库可看到数据库中的各表，可进行表、字段的增删改，可以导入、导出数据库信息，如图3-5所示。



图 3-5 软件的管理界面

步骤03 单击 **数据库** 命令，打开本地的“数据库”管理页面，在“新建数据库”文本框中输入数据库的名称phpweb，单击后面的数据库类型下拉按钮，在弹出的下拉列表框中选择utf8_general_ci选项，如图3-6所示。



UTF8是数据库的编码格式，通常在开发PHP动态网站时Dreamweaver默认的格式就是UTF8格式，在创建数据库时也要保证数据库存储的格式和网页调用的格式一样，这里要介绍一下utf8_bin和utf8_general_ci的区别。其中ci是case insensitive，即“大小写不敏感”，a和A在字符判断中会被当作一样的；bin是二进制，a和A会被区别对待。

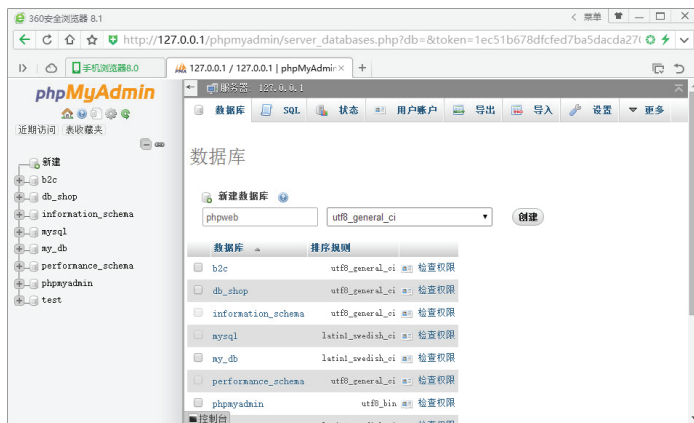


图 3-6 软件的管理界面

步骤04 单击“创建”按钮，返回“常规设置”页面，在数据库列表中就已经建立了phpweb的数据库，如图3-7所示。



图 3-7 创建后的页面

步骤05 数据库建立后还要建立网页数据所需的数据表。这个网站数据库的数据表是websql。建立数据库后，接着单击左边的phpweb数据库将其连接上，如图3-8所示。

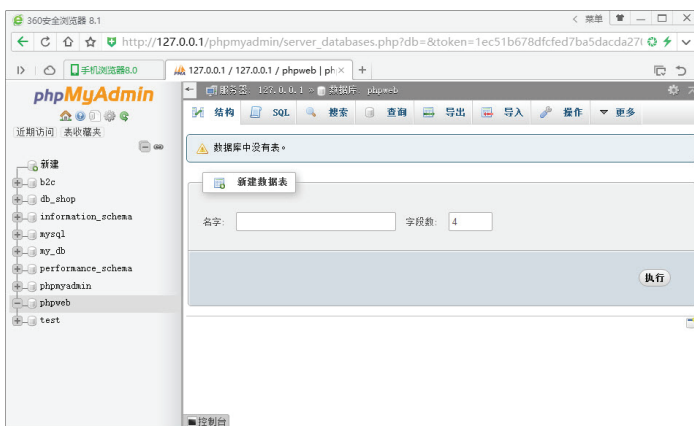


图 3-8 开始建数据表

步骤06 打开“新建数据表”页面，在“名字”文本框中输入数据表名websql，在“字段数”文本框中输入本数据表的字段数为4，表示将创建4个字段来存储数据，如图3-9所示。

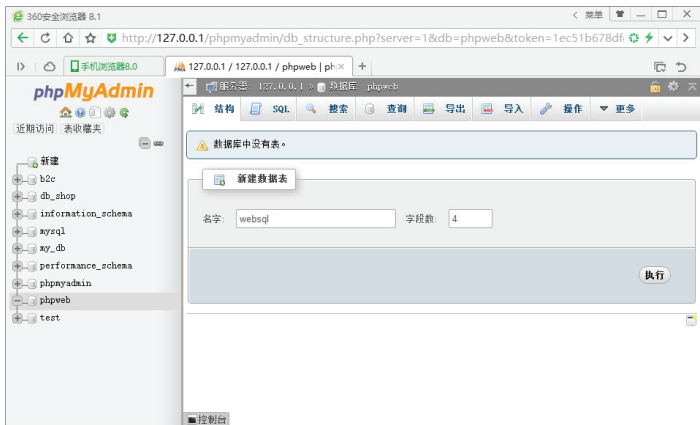


图 3-9 输入数据表名 websql 和字段数

步骤07 再单击“执行”按钮，切换到数据表的字段属性设置页面，输入数据域名以及设置数据域位的相关数据，如图3-10所示。各字段的意义如表3-2所示。这个数据表主要是记录每个人的基本数据和成绩。

表3-2 websql数据表

字段名称	字段类型	字段大小	说明
ID	int	11	自动编号
name	varchar	20	个人姓名
age	tinyint	4	个人年龄
Result	varchar	20	个人成绩

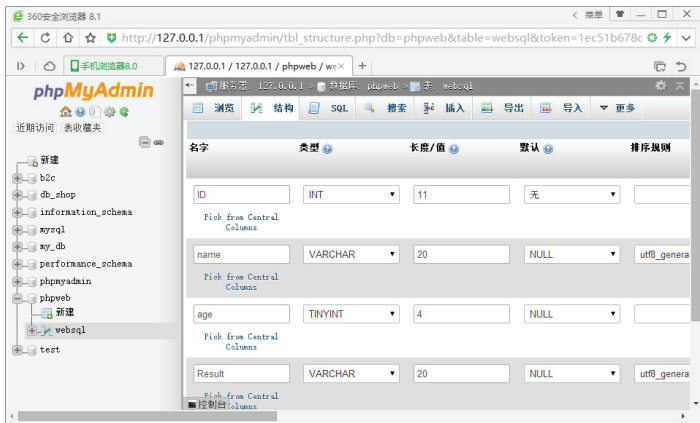


图 3-10 设置数据库字段属性

步骤08 最后单击“保存”按钮，切换到“结构”页面。实例将要使用的数据库建立完毕，如图3-11所示。

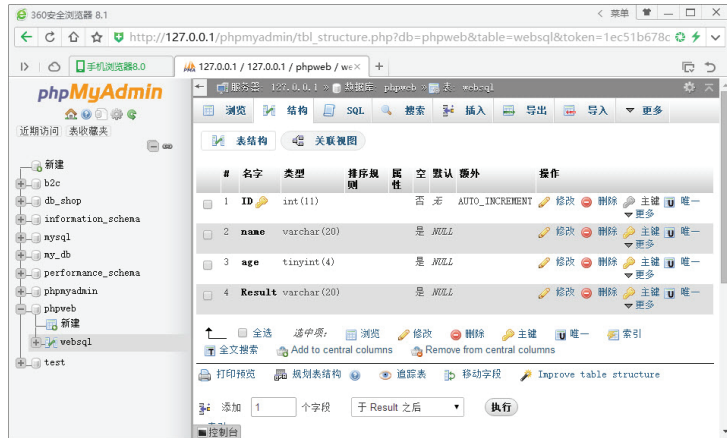


图 3-11 建立的数据库页面

步骤09 为了页面制作的调用需要，可以先在数据表里加入10笔数据。在数据表手动加入名为test1~test10的10个测试姓名，年龄和成绩可以设置不同的数据，如图3-12所示。



图 3-12 加入 10 笔数据

3.1.3 定义Web站点

在Dreamweaver CC 2017中创建一个“成绩查询”网站站点phpweb，由于这是PHP数据库网站，因此必须设置本机数据库和测试服务器，主要的设置如表3-3所示。

表3-3 站点设置的基本参数

站点名称	web
本机根目录	D:\xampp\htdocs\phpweb
测试服务器	D:\xampp\htdocs\
网站测试地址	http://127.0.0.1/phpweb/
MySQL服务器地址	D:\xampp\MySQL\data\phpweb
管理账号 / 密码	root / 空
数据库名称	phpweb

创建web站点的具体操作步骤如下：

步骤01 在D:\xampp\htdocs路径下建立phpweb文件夹（如图3-13所示），本实例所有建立的网页文件都将放在该文件夹下。

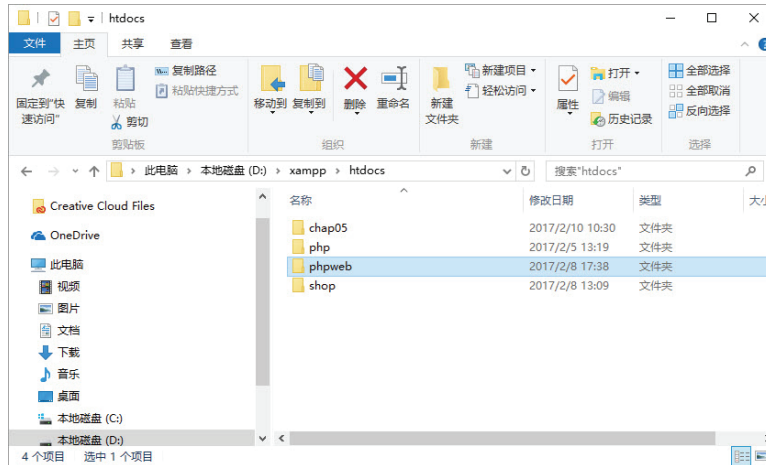


图 3-13 建立站点文件夹 phpweb

步骤02 启动Dreamweaver CC 2017，执行菜单栏中的“站点”→“管理站点”命令，打开“管理站点”对话框，如图3-14所示。

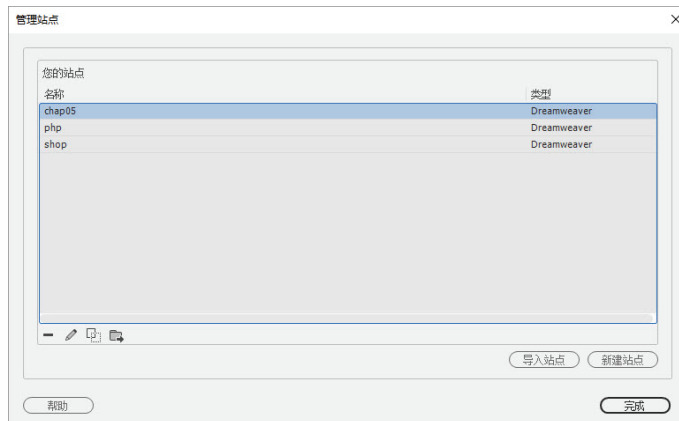


图 3-14 “管理站点”对话框

步骤03 单击“新建站点”按钮，打开“站点设置对象”对话框，左边是站点列表框，其中显示可以设置的选项。进行如图3-15所示的参数设置，设置站点名称为phpweb，本地站点文件夹地址为D:\xampp\htdocs\phpweb。

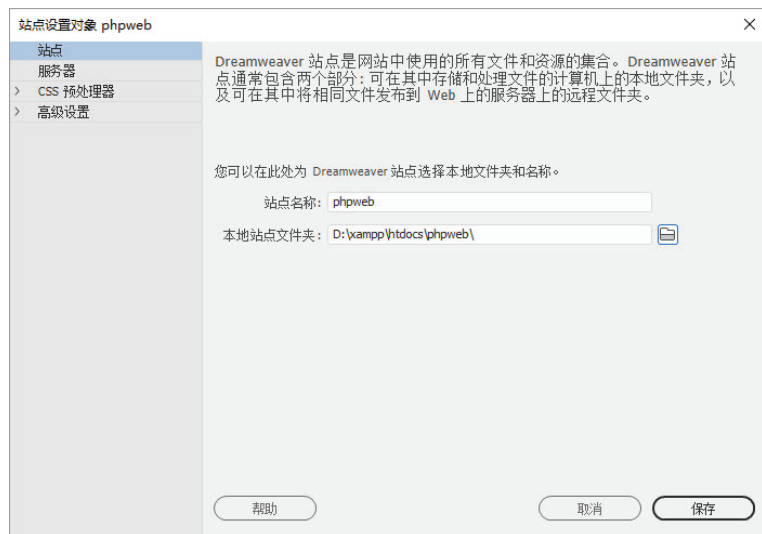



图 3-15 建立 phpweb 站点

步骤04 单击列表框中的“服务器”选项，并单击“添加服务器”按钮，打开“基本”选项卡进行如图3-16所示的参数设置。

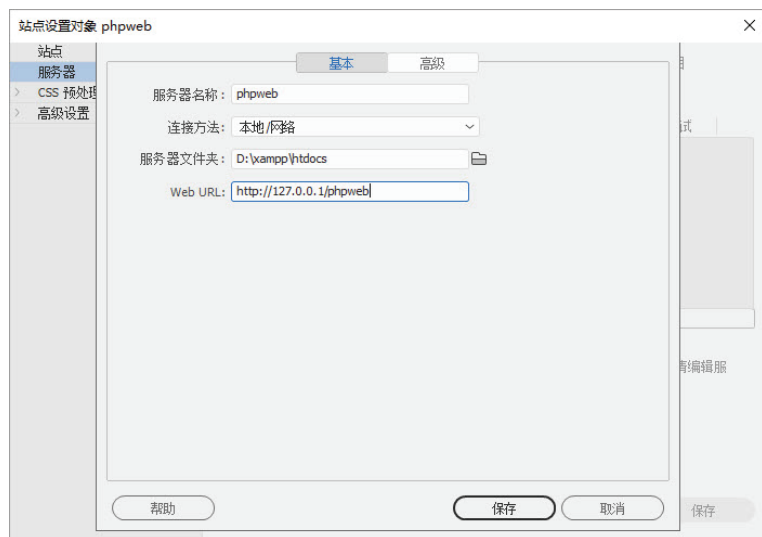


图 3-16 “基本”选项卡设置

步骤05 设置后再单击“高级”选项卡，打开“高级”服务器设置对话框，选中“维护同步信息”复选框，在“服务器模型”下拉列表框中选择PHP MySQL选项（表示是使用PHP开发的网页），其他的保持默认值，如图3-17所示。

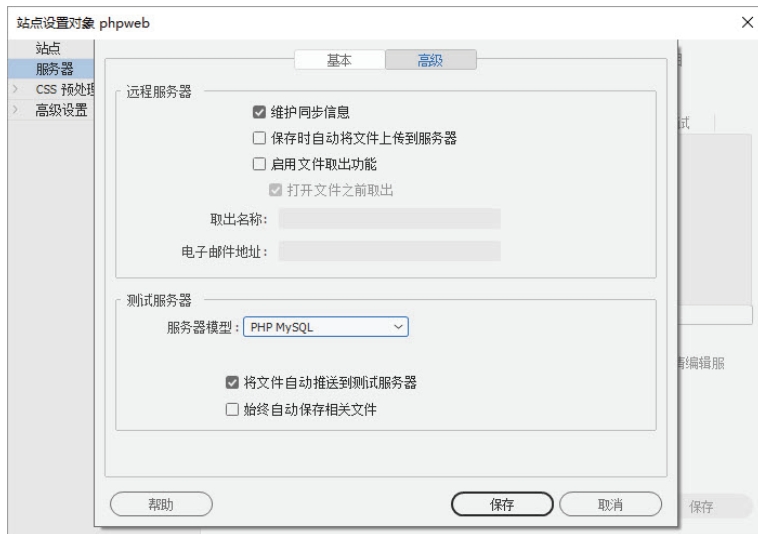


图 3-17 设置“高级”选项卡

步骤06 单击“保存”按钮，返回“服务器”设置界面，选中“测试”单选按钮，如图3-18所示。

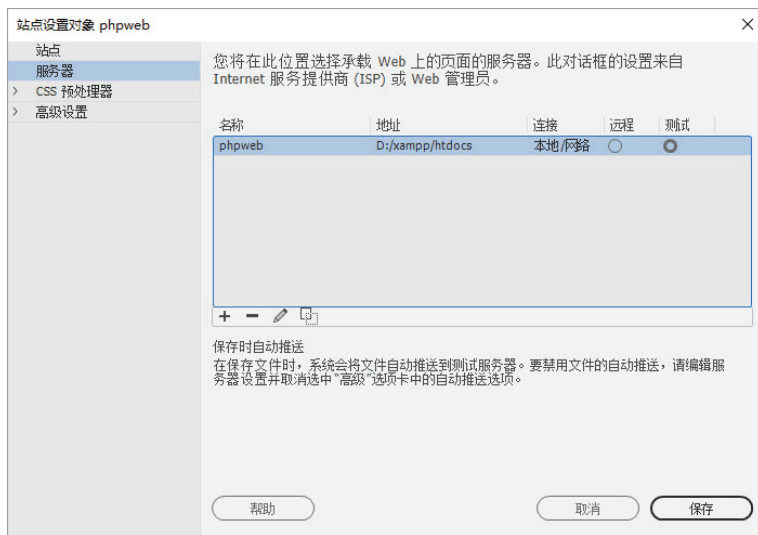


图 3-18 设置“服务器”参数

步骤07 单击“保存”按钮，完成站点的定义设置。在Dreamweaver CC 2017中就已经拥有了刚才所设置的站点Web。单击“完成”按钮，关闭“管理站点”对话框，这样就完成了Dreamweaver CC 2017测试Web站点的网站环境设置。

这里要说明一下，之所以建立Dreamweaver的站点配置，是为了方便使用Dreamweaver在开发网站时通过单击“实时视图”窗口可以在编辑窗口的上部分实时看到PHP网站运行的效果，如图3-19所示。这也是本书推荐初学者使用Dreamweaver作为PHP网站开发编辑器的主要原因之一，方便用户所见即所得，可以随时进行编辑并即时看到制作的效果。

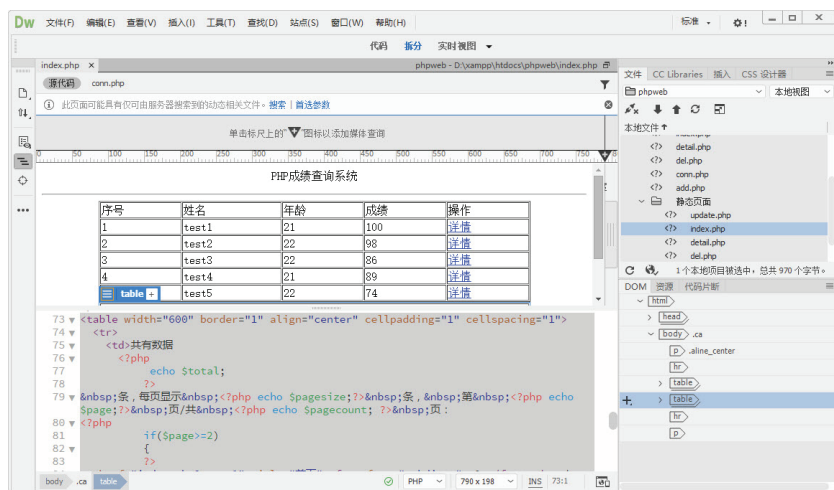


图 3-19 Dreamweaver 制作的实时窗口显示

3.1.4 建立数据库连接

完成了站点的定义后，需要将网站与前面建立的phpweb数据库建立连接。

网站与数据库的连接步骤如下：

步骤01 执行菜单栏“文件”→“新建”命令，在网站根目录下新建一个名为conn.php的网页，输入网页标题“PHP成绩查询系统”，然后执行菜单栏“文件”→“保存”命令将网页保存，如图3-20所示。

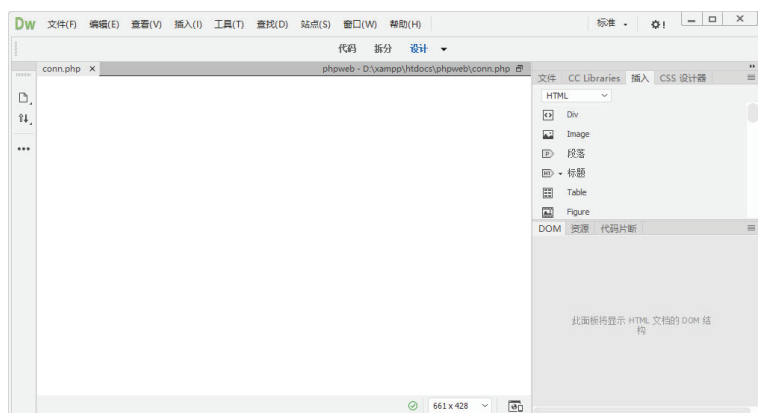


图 3-20 创建 conn.php 空白页

步骤02 conn.php文件是Dreamweaver用来存放MySQL连接设置的文件，在大多数的PHP编程文件中都喜欢用conn.php文件名来命名。打开该文件并使用“代码”视图，输入MySQL的连接代码，如图3-21所示。

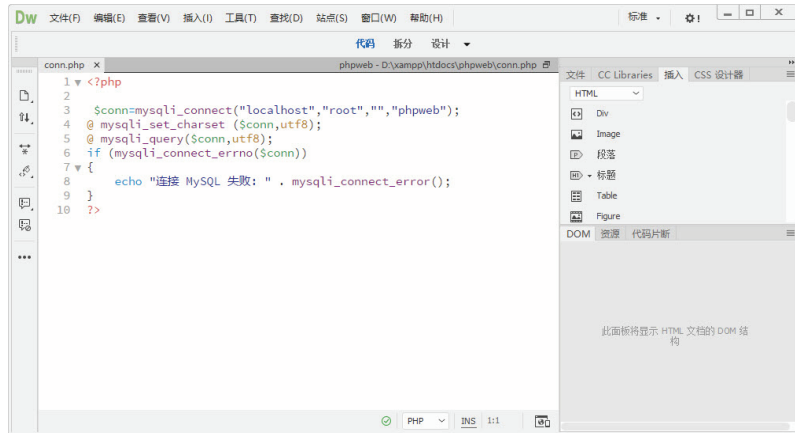


图 3-21 数据库连接设置

在这个文件中定义了与MySQL服务器的连接（mysqli_connect函数），包括以下内容：

```
<?php
//建立数据库连接;
$conn=mysqli_connect("localhost","root","","phpweb");
//设置字符为utf-8, @抑制字符变量的声明提醒。
@ mysqli_set_charset ($conn,utf8);
@ mysqli_query($conn,utf8);
//如果连接错误显示错误原因。
if (mysqli_connect_errno($conn))
{
    echo "连接 MySQL 失败: " . mysqli_connect_error();
}
?>
```

其中，表示数据库连接的代码为：

- localhost: MySQL服务器的地址。
- phpweb: 连接数据库的名称。
- root: MySQL用户名。
- 空白“”：MySQL用户密码。这里使用环境默认是空白。

连接后才能对数据库进行查询、新增、修改或删除的操作。如果在网站制作完成后将文件上传至网络上的主机空间时发现网络上的MySQL服务器访问的用户名、密码等方面与本机设置的有所不同，可以直接修改conn.php文件。

3.2 动态查询功能

使用PHP+Dreamweaver可以利用Dreamweaver软件自带的动态服务器行为快速建立一些基本动态功能，但PHP 7.0之后由于废弃了mysql.dll（推荐使用mysqli或者pdo_mysql），就无法再使用这些基础功能了。需要在Dreamweaver中单独编写PHP代码来实现对MySQL数据库的操作，主要包括创建记录集、插入记录、更新记录、重复区域、显示区域和记录集分页等常用的动态服务器行为。

3.2.1 创建新记录集

在每个需要查看数据库记录的页面中皆需为其建立一个MySQL数据库的查询“记录集（查询）”，从而可以让Dreamweaver知道，目前这个网页中所需要的是数据库中的哪些数据。即便需要的内容一样，在不同网页也需要单独建立。同一个数据库只需建立一次MySQL连接，但我们可以为同一个MySQL数据库连接建立多个“记录集”，配合筛选的功能达到某个记录集只包含数据库中符合某些条件的记录。

下面以系统的实例实现来说明，具体的步骤如下：

步骤01 执行菜单栏“文件”→“新建”命令，在网站根目录下新建一个名为index.php的网页，输入网页标题“PHP成绩查询系统”，然后执行菜单栏“文件”→“保存”命令将网页保存。

步骤02 打开index.php文件后，使用“代码”视图，输入代码如下：

```
<?php
    $sql1=mysqli_query($conn,"select * from websql order by ID asc limit ");
    //设置websql数据表按ID升序排序查询出所有数据
    while($info1=mysqli_fetch_array($sql1))
    //使用mysqli_fetch_array查询所有记录集，并定义为$info1
    {
    }
?>
```

字段的功能说明如表3-4所示。

表3-4 字段与功能说明

字段	说明
mysqli_query函数	一般用\$sql作为变量定义
mysqli_fetch_array函数	选择所建立记录集作为数组
order by ID asc 排序	是否依照某个字段值进行排序。比如，在新闻系统中需要把新的新闻放到前面位置，就可以使用排序的功能

记录集使用到的就是SELECT语句，因为查询出来的结果可能会有很多条，所以称为记录集（合），而“筛选”部分则对应WHERE子句。

其程序具体分析如下：

(1) 第1行定义了查询数据库的SQL语句。使用所定义的SQL语句对数据库执行查询操作（mysqli_query()），此时返回的结果是资源标识符，还不能被使用。

(2) 第3行将前面查询的结果以关系型数组的形式（mysqli_fetch_array()）传至变量\$info1，然后就可以使用\$info1_记录集名称[‘字段名称’]来取得记录集字段值。

3.2.2 显示所有记录

要将记录集内的记录（数据库中的数据）直接显示到网页上，实现的步骤如下：

步骤01 在“文件”面板中打开index.php，在网页中制作一个如图3-22所示的2×5表格，然

后在表格的列<td>代码中输入如下代码:

```
<td><?php echo $info1['ID'];?></td><!--显示ID字段-->
<td><?php echo $info1['name'];?></td><!--显示name字段-->
<td><?php echo $info1['age'];?></td><!--显示age字段-->
<td><?php echo $info1['Result'];?></td><!--显示Result字段-->
```

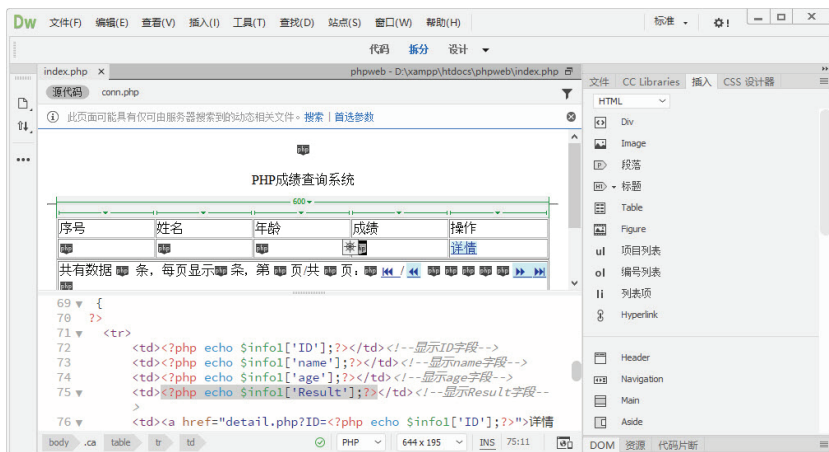


图 3-22 绑定字段

步骤02 将序号、姓名、年龄、成绩4个字段分别输入相应的单元格后，单击 **实时视图** 按钮。视图所呈现的效果与使用浏览器打开的网页一样，原本仅显示{记录集名称.字段名称}的部分将会显示出记录集内的记录，这里由于没有设置完整的重复命令，运行只能显示空白的第一条记录集，如图3-23所示。

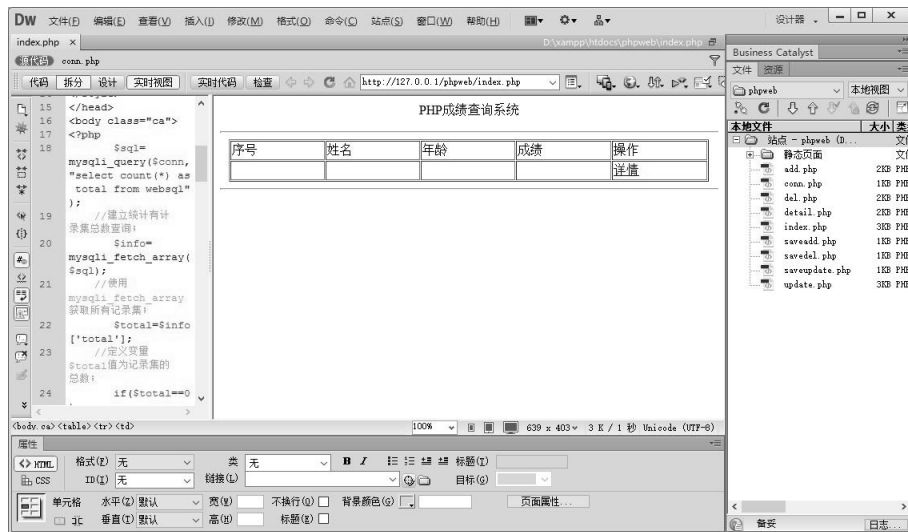


图 3-23 显示一条空白的数据

步骤03 再单击一次 **实时视图** 按钮，将页面切换到 **代码** 视图。我们来看{记录集名称.字段名称}部分的代码。之所以要确认选取的标签为 <tr>，是因为重复区域会使用while循环包围所作用的范围。而需要重复的仅是第2行的表格，在HTML中表格的行是使用 <tr>标签。确认选取的标签正确，执行时才不会发生错误。运行时将显示所有的记录集，如图3-24所示。

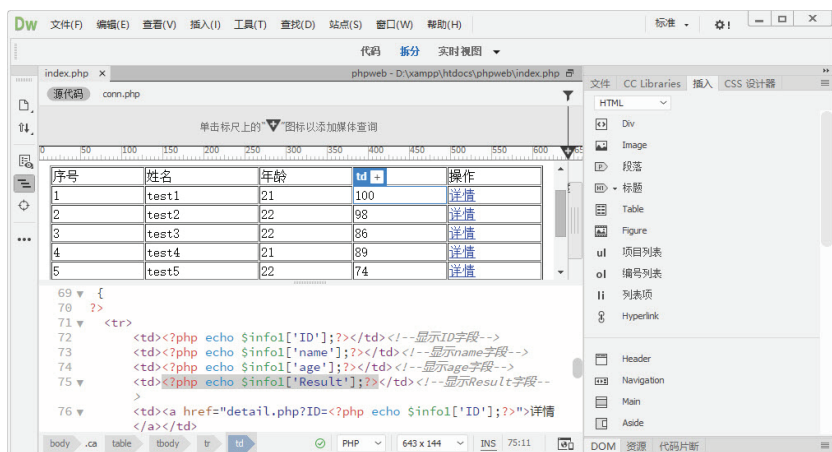


图 3-24 显示所有的记录集

3.2.3 记录集的分页

上一节已经可以查询记录集中所有的记录了，那剩下的记录如何显示出来呢？下面就介绍记录集分页功能的实现方法。

步骤01 在index.php页面代码前面加入分页统计查询的PHP代码如下：

```
<?php
    $sql=mysqli_query($conn,"select count(*) as total from websql");
    //建立统计有记录集总数查询;
    $info=mysqli_fetch_array($sql);
    //使用mysqli_fetch_array获取所有记录集;
    $total=$info['total'];
    //定义变量$total值为记录集的总数;
    if($total==0)
    {
        echo "本系统暂无任何查询数据!";
    }
    //如果记录总数为0则显示无数据;
    else
    {
        ?>
```

步骤02 对第一个显示所有记录查询进行页码划分，实现的代码如下：

```
<?php

    $pagesize=5;
    //设置每页显示5条记录;
    if ($total<=$pagesize) {
        $pagecount=1;
        //定义$pagecount初始变量为1页;
    }
    if (($total%$pagesize) !=0) {
        $pagecount=intval ($total/$pagesize) +1;
```

```

        //取页面统计总数为整数;
    }else{
        $pagecount=$total/$pagesize;

    }
    if((@ $_GET['page'])==""){
        $page=1;
        //如果总数小于5则页码显示为1页;
    }else{
        $page=intval($_GET['page']);
        //如果大于5条则显示实际的总数;
    }
    $sql1=mysqli_query($conn,"select * from websql order by ID asc limit
    ".($page-1)*$pagesize.",$pagesize ");
    //设置websql数据表按ID升序排序查询出所有数据;
    while($info1=mysqli_fetch_array($sql1))
        //使用mysqli_fetch_array查询所有记录集,并定义为$info1;
    {
    ?>

```

这里将分页功能定义为一个记录集,通过统计之后即可进行调用,作为本范例中建立记录集导航条。

3.2.4 显示记录计数

在页面下方输入“共有数据X条,每页显示Y条,第A页/共B页:”,建立起记录集导航条,以便让用户了解有多少页记录,当前正在浏览第几页。

显示记录计数的步骤如下:

步骤01 在index.php中建立一个表格,将“共有数据X条,每页显示Y条,第A页/共B页:”中输入显示记录数的PHP代码如下:

```

<table width="600" border="1" align="center" cellpadding="1" cellspacing="1">
<tr>
<td>共有数据
<?php
    echo $total;//显示总页数;
    ?>
    &nbsp;条,每页显示&nbsp;?<?php echo $pagesize;//打印每页显示的总条数; ?>&nbsp;条,
    &nbsp;第&nbsp;?<?php echo $page;//显示当前页码; ?>&nbsp;页/共&nbsp;?<?php echo
    $pagecount;//打印总页码数 ?>&nbsp;页;
    <?php
        if($page>=2)
            //如果页码数大于等于2则执行下面程序
        {
            ?>
            <a href="index.php?page=1" title="首页"><font face="webdings"> 9 </font></a> /
            <a href="index.php?id=<?php echo $id;?>&amp;page=<?php echo $page-1;?>" title="
            前一页"><font face="webdings"> 7 </font></a>
            <?php

```

```

    }
    if($pagecount<=4){
        //如果页码数小于等于4执行下面程序
        for($i=1;$i<=$pagecount;$i++){
            ?>
            <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
            <?php
            }
        }else{
            for($i=1;$i<=4;$i++){
                ?>
                <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
                <?php }?>
                <a href="index.php?page=<?php echo $page-1;?>" title="前一页"><font
                face="webdings"> 8 </font></a> <a href="index.php?id=<?php echo
                $id;?>&amp;page=<?php echo $pagecount;?>" title="尾页"><font face="webdings"> :
                </font></a>
                <?php }?></td>
            </tr>
        </table>
    }

```

对于加入的PHP代码含义在代码中都单独进行了标注。

步骤02 完成后，当我们浏览该网页时便会出现当前共有几条数据，每页显示的条数，当前是第几笔到第几笔的提示文字，如图3-25所示。

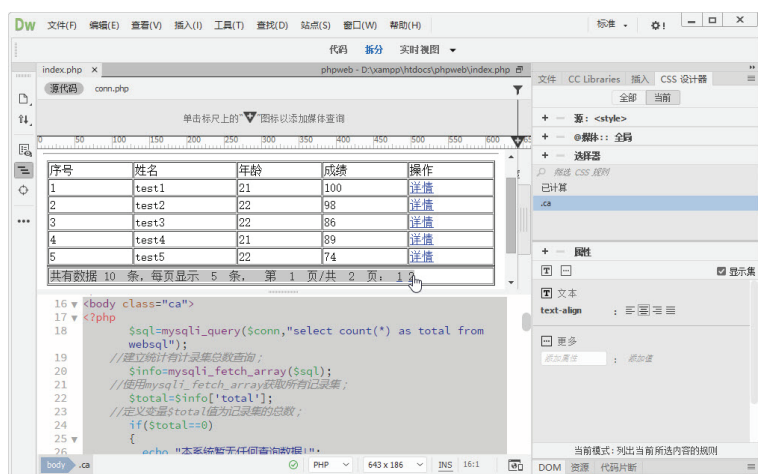


图 3-25 建立导航条效果

3.2.5 显示详细信息

通常一个动态网站的数据量是比较大的，很多时候并不会一开始就将数据库所有字段、记录都显示出来。例如一个新闻系统，在首页只会显示新闻的日期与标题，更详细的新闻内容需要选择标题后进入另一个页面才能显示。假设显示新闻标题的页面是index.php，而显示详细新闻内容的网页名称为detail.php。当在index.php中单击标题的链接后，此时该超链接会带着一个参数到detail.php，网址类似于detail.php?ID=1。多出的ID=1是一个变量名为ID、值为1的URL参数。当detail.php收到


ID=1的URL参数后，便利用这个URL参数在建立记录集时筛选所指定的新闻记录，并将记录详细信息显示在网页上，这样就构成了一个简单的新闻系统架构。要筛选指定的记录可以在SQL中使用WHERE子句。

详细页的制作步骤如下：

步骤01 使用Dreamweaver CC 2017创建一个空白detail.php页面并保存。index.php中选择用来连接到详细信息页面的部分（其实就是选择要在哪里建立超级链接），在本例中选择的是单击“详情”文字链接，如图3-26所示。



图 3-26 单击首页链接到 detail.php

步骤02 在下面的“属性”面板中找到建立链接的部分，并单击“浏览文件”图标, 如图3-27所示。

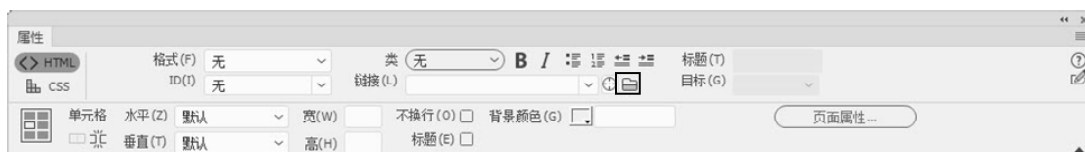


图 3-27 建立链接设置

步骤03 在弹出的对话框中选择用来显示详细记录信息的页面detail.php，如图3-28所示。

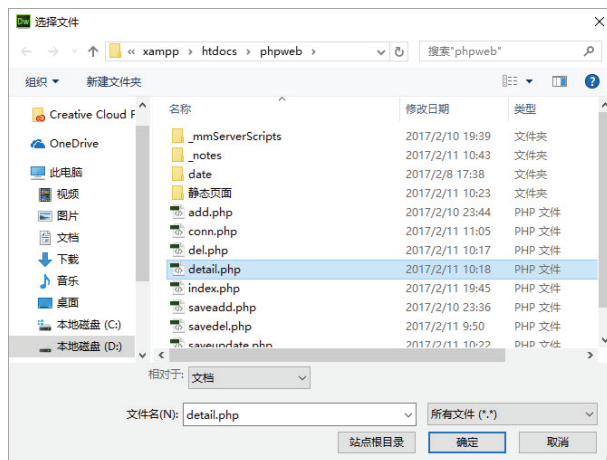


图 3-28 设置链接的文件

步骤04 如果只是这样，那只会是单纯的超级链接并没有附带URL参数，因此要设置超级链接要附带的URL参数的名称与值。本例将参数名称命名为ID，接收上一页传递过来的ID值。

步骤05 地址变成detail.php?ID=<?php echo \$info1['ID'];?>，如图3-29所示。

```
<td><a href="detail.php?ID=<?php echo $info1['ID'];?>">详情</a></td>
<!--设置跳转并传递ID值-->
```



图 3-29 完成后的链接地址

步骤06 设置完成后，可以在浏览器打开index.php页面。在浏览器下方的状态栏上可以看到每一条记录的链接都带着URL参数ID，其值是每条记录的ID，如图3-30所示。

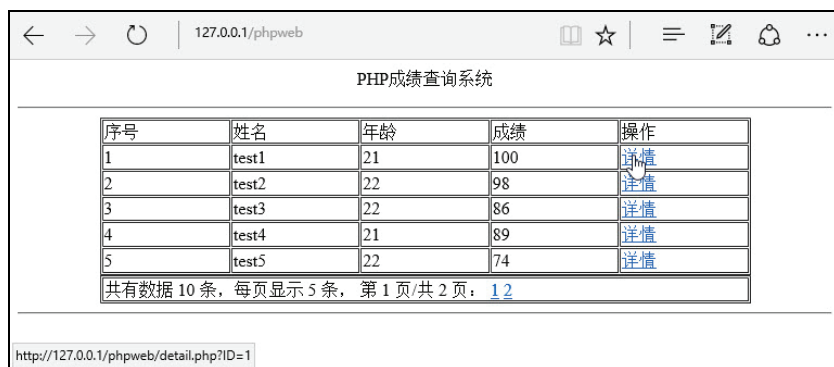


图 3-30 单击链接的属性显示

前面已经完成index.php页面的制作，下面来设计接收URL参数的detail.php页面，看看如何用收到的参数来筛选指定的记录。

步骤01 打开detail.php页面后输入的PHP代码如下：

```
<?php require_once('conn.php'); ?>
//调用数据库连接文件实现连接
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>PHP成绩查询系统</title>
<style type="text/css">
.aline_center {
    text-align: center;
}
.ca {
    text-align: center;
}
</style>
</head>
<body class="ca">
<p class="aline_center">PHP成绩查询系统</p>
<hr />
<table width="600" border="1" align="center" cellpadding="1" cellspacing="1">
<tr>
<td>序号</td>
<td>姓名</td>
```

```

        <td>年龄</td>
        <td>成绩</td>
        <td>编辑</td>
    </tr>
    <?php
    $ID=@ $_GET['ID'];
    //设置$ID值为前页传过来的ID值,使用$_GET[]函数实现
    $sql=mysqli_query($conn,"select * from websql where ID='".$ID."'");
    //建立数据库条件查询,查询条件为ID='".$ID."'
    $info=mysqli_fetch_array($sql);
    ?>
    <tr>
        <td><?php echo $info['ID'];?></td>
        <td><?php echo $info['name'];?></td>
        <td><?php echo $info['age'];?></td>
        <td><?php echo $info['Result'];?></td>
        <td><a href="update.php?ID=<?php echo $info['ID'];?>">更新</a> / <a
href="del.php?ID=<?php echo $info['ID'];?>">删除</a></td>
    </tr>
</table>
<hr />
<p>&nbsp;</p>
</body>
</html>

```

步骤02 记录集建立完毕后,可以把各个字段“插入”页面相应的单元格中,完成的页面如图3-31所示。

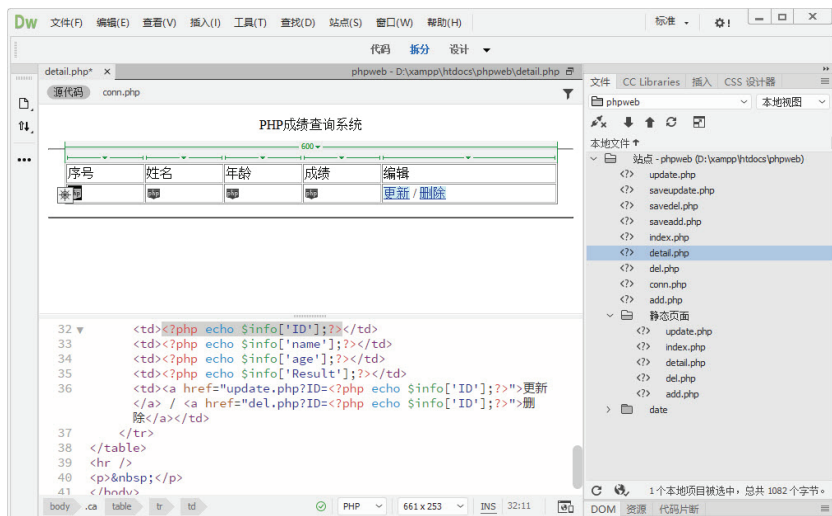


图 3-31 制作的详细页面

步骤03 完成后直接按F12键在浏览器中打开detail.php,发现内容是空白的,如图3-32所示。这是怎么回事呢?因为在网址后面没有带URL参数,当然记录集里就不会有任何内容了。

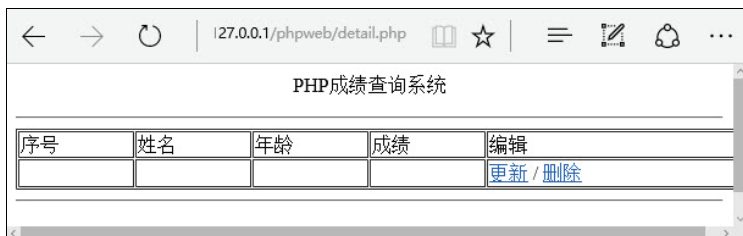


图 3-32 显示为空白

步骤04 直接在网址后加上URL变量ID，其值可以选1~10的任何一个值，如这里输入6，然后按Enter键，网页显示的结果如图3-33所示。

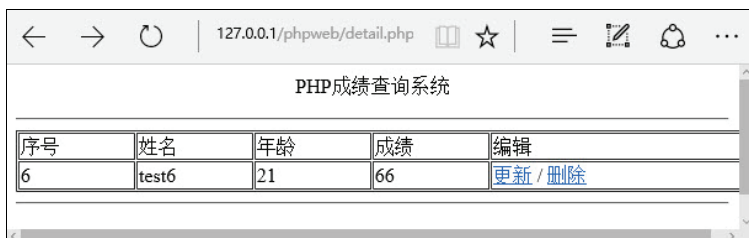


图 3-33 URL 参数 ID=6 时的详细页面

步骤05 在index.php中，每一条记录的网址都带有特定的参数链接到detail.php，如图3-34所示。

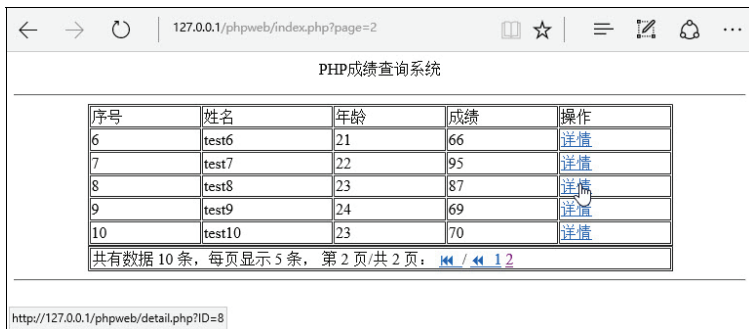


图 3-34 单击编号链接

这里如果不以“详情”作为主链接也是可以的，也可以使用标题作为链接。单击某个新闻标题即可打开相应的详细页面采用的就是这种技术。

3.3 编辑记录功能

数据库记录在页面上的显示、重复、分页、计数、显示详细信息的操作已经介绍完毕，本节将介绍在Dreamweaver CC 2017中进行增加、修改以及删除记录的操作。

3.3.1 增加记录功能

在数据表websql中有4个字段，其中ID字段为主键且附加了“自动编号”属性，因此在新增记

录时不必考虑ID字段，只需增加3个值即可。

实现的步骤如下：

步骤01 创建一个空白的php网页，并命名为add.php，先添加一个表单，再插入一个4×2表格，键入相关提示后依序放上3个文本字段、两个按钮，完成后如图3-35所示。

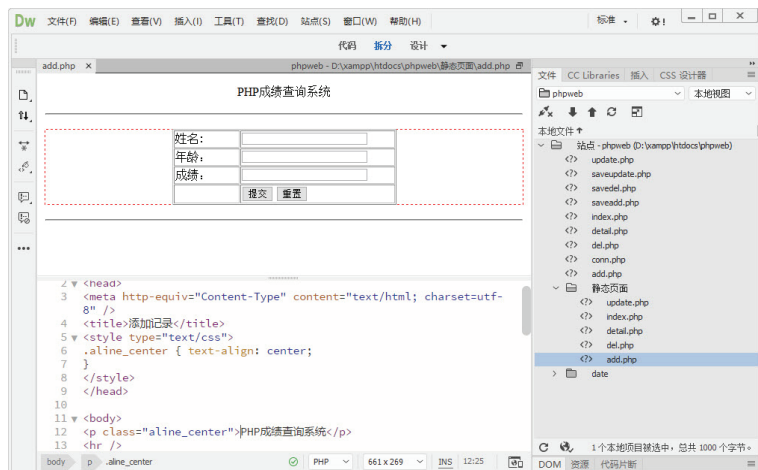


图 3-35 建立表单并设计网页

当需要新增、更新记录时，网页中需要有一个表单且表元素必须置于表单内，在单击按钮后只有在表单内的元素会被以POST或GET的方式传递。

步骤02 插入3个文本字段，并分别选择各个文本字段，在“属性”面板为其命名，分别是姓名name、年龄age、成绩Result，注意在设计时要与记录集字段名称一一对应，如图3-36所示为成绩Result的文本框属性设置。



图 3-36 命名文本域



当表单元素的命名与记录集字段相符合时，在做“新增记录”“更新记录”时PHP会自动将表单元素与记录集字段相匹配，同时也方便编程人员在做修改时能快速找到相应的字段。

步骤03 设置完成后，在页面中加入一个跳转和判断的命令，其中action="saveadd.php"用于提交表单后提交到saveadd.php代码页进行处理，onSubmit="return chkinput(this)"是在本页的前面加入表单验证的功能，即不能提交空白数据，具体所加入的位置如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>添加记录</title>
<style type="text/css">
```



```

</form>
<hr />
<p>&nbsp;</p>
</body>
</html>

```

步骤04 提交到的saveadd.php网页是纯PHP代码，主要功能是将提交的表单数据插入MySQL数据库中，实现的PHP代码如下：

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<?php
include("conn.php");
$name=$_POST['name'];
$age=$_POST['age'];
$result=$_POST['Result'];
mysqli_query($conn,"insert into websql (name,age,result) values ('$name',
'$age','$result')");
echo "<script>alert('添加成功!');history.back();</script>";
//用script脚本语言实现插入后提醒“添加成功”，history.back();实现跳转回原页的功能
?>

```

步骤05 直接按F12键在浏览器中打开网页，输入值如图3-37所示，单击“提交”按钮尝试新增一记录。



图 3-37 输入记录数据

步骤06 单击“提交”按钮后，网址将从add.php转至index.php。单击网页下方的分页导航条的“最后一页”链接（这里是单击数字3链接），便可以看到刚才新增的记录，如图3-38所示。



图 3-38 增加记录后的效果

3.3.2 更新记录功能

更新记录功能是指将数据库中的旧数据根据需要进行更新的操作。这里我们会用前面已经使用到的detail.php文件。

更新记录功能的操作步骤如下：

步骤01 打开detail.php网页后，选择链接文字“更新”，如图3-39所示。

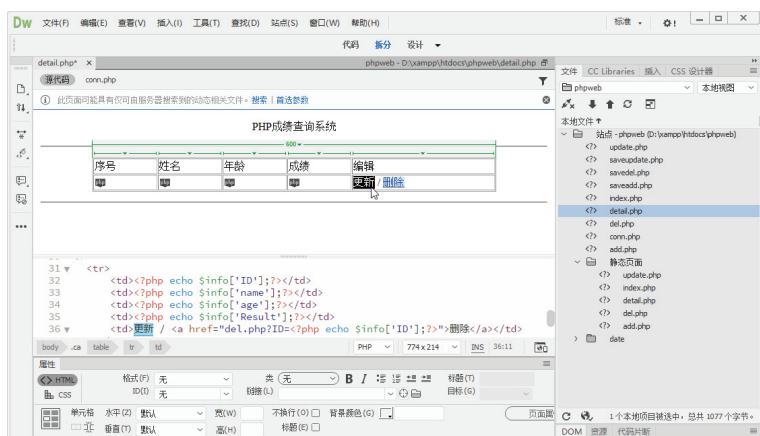


图 3-39 选择链接文字

步骤02 在“属性”面板中单击如图3-40所示的“浏览文件”图标，为其建立附带URL参数的超级链接。



图 3-40 单击“浏览文件”图标

步骤03 输入用来更新记录使用的update.php页面，为其建立名称为ID、值是记录集ID字段值的URL参数。

步骤04 完成后的链接地址如图3-41所示。

```
update.php?ID=<?php echo $info['ID'];?>
//传递ID到update.php页面
```

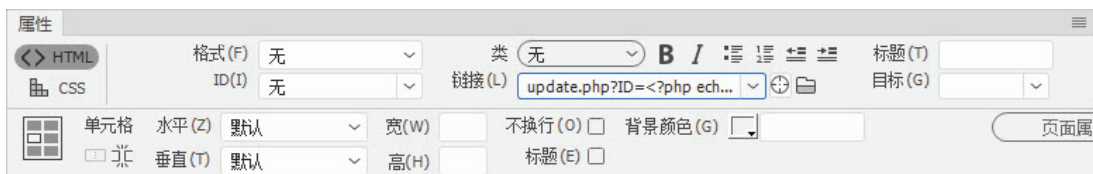


图 3-41 传递 ID 至 update.php

步骤05 创建update.php空白文档，该页面的设计与详细信息页面detail.php相同，都是要利用接收到的URL参数筛选指定记录。使用PHP设置查询记录集代码如下：

```
<?php
    $ID=@ $_GET['ID'];
    $sql=mysqli_query($conn,"select * from weysql where ID='".$ID."'");
    $info=mysqli_fetch_array($sql);
?>
```

步骤06 将页面中应该有的表单、文本字段、按钮设置完成，在“绑定”面板中将记录集内的字段拖动至页面上各对应的文本字段中，如图3-42所示。由于ID是主键，不能随便变更主键的值，因此这里的ID部分是直接显示为文本字段，其他为输入文本。



图 3-42 绑定字段

步骤07 同样需要对在单击“更新”按钮时进行一个表单判断，不能让姓名、年龄、成绩三个字段为空，完成整页的代码如下：

```
<?php require_once('conn.php'); ?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>更新页面</title>
<style type="text/css">
.aline_center { text-align: center;
}
.dc {
text-align: center;
}
</style>
</head>

<body>
<p class="aline_center">PHP成绩查询系统</p>
<hr />
<script language="javascript">
function chkinput(form)
{
if (form.name.value=="")
{
alert("请输入姓名!");
form.name.select();
return(false);
}
if (form.age.value=="")
{
alert("请输入年龄!");
form.age.select();
return(false);
}
```

```

    }
    if(form.Result.value=="")
    {
        alert("请输入年龄!");
        form.Result.select();
        return(false);
    }
    return(true);
}
</script>
<form name="form1" method="post" action="saveupdate.php" onSubmit="return
chkinput(this)">
<table width="600" border="1" align="center" cellpadding="1" cellspacing=
"1">
<tr>
<td>序号</td>
<td>姓名</td>
<td>年龄</td>
<td>成绩</td>
</tr>
<?php
$ID=@ $_GET['ID'];
$sql=mysqli_query($conn,"select * from websql where ID='".$ID."'");
$info=mysqli_fetch_array($sql);
?>
<tr>
<td><?php echo $info['ID'];?></td>
<input name="ID" type="hidden" id="ID" value="<?php echo $info['ID']?>"
size="16" />
<!--加入ID隐藏域使用表单传递时能把ID值正确传给下一页-->
<td><input name="name" type="text" id="name" value="<?php echo
$info['name']?>" size="16" /></td>
<td><input name="age" type="text" id="age" value="<?php echo
$info['age']?>" size="16" /></td>
<td><input name="Result" type="text" id="Result" value="<?php echo
$info['Result']?>" size="16" /></td>
</tr>
<tr>
<td colspan="4" class="dc"><input type="submit" name="button" id="button"
value="更新" />
<input type="reset" name="button2" id="button2" value="重置" /></td>
</tr>
</table>
</form>
<hr />
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>

```

步骤08 创建saveupdate.php网页，使用update函数实现对表格数据的更新，具体的PHP代码如下：

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<?php
    $ID=$_POST['ID'];
    $name=$_POST['name'];
    $age=$_POST['age'];
    $Result=$_POST['Result'];
    include("conn.php");
    mysqli_query($conn,"update weysql set name='$name',age='$age',Result=
'$Result' where id='$ID'");
    echo "<script>alert('修改成功!');history.back();</script>";
?>
```

步骤09 最后在浏览器中打开index.php，选择最后一笔记录到详情页面detail.php，再在详情页面选择“更新”链接，如图3-43所示。



图 3-43 选择“更新”链接

步骤10 在update.php中可以修改姓名、年龄与成绩的字段值，而ID文本字段是不能被修改的，更改完成后单击“更新”按钮，如图3-44所示。



图 3-44 修改数据

步骤11 返回到index.php，检查记录是否被正确更新，如图3-45所示。

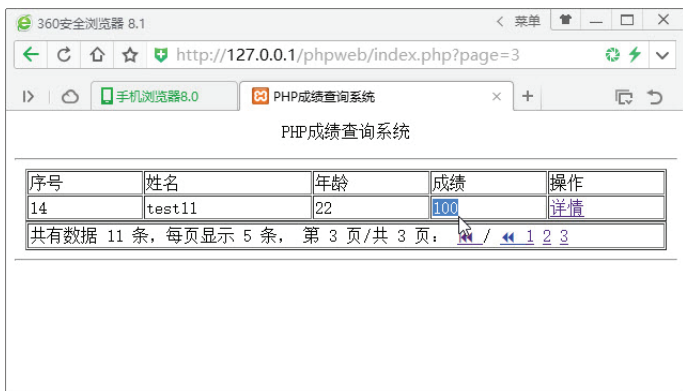


图 3-45 完成更新的功能页面

这部分的程序代码与插入记录基本相同，差别只在于隐藏字段的名称不同，使用的是UPDATE语句。

3.3.3 删除记录功能

删除记录功能是指将数据从数据库中删除，使用delete（删除记录）命令即可实现。具体的实现步骤如下：

步骤01 使用超级链接带着URL参数转到删除页面del.php。首先在detail.php中选中“删除”，在“属性”面板中建立链接，如图3-46所示。

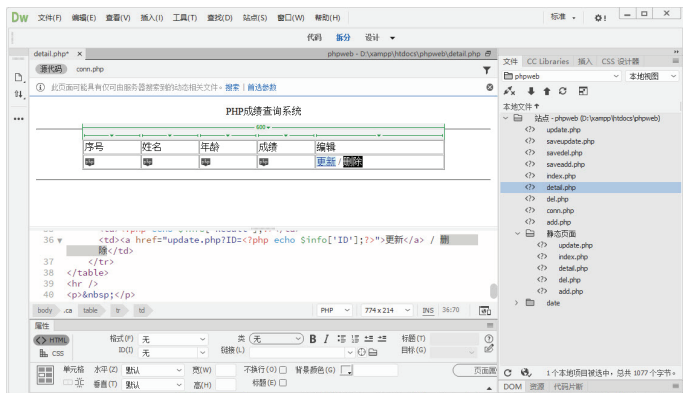


图 3-46 设置“删除”链接

步骤02 在“属性”面板中单击如图3-47所示的“浏览文件”图标，为其建立附带URL参数的超级链接。



图 3-47 单击“浏览文件”图标

步骤03 输入用来删除记录使用的del.php页面，为其建立名称为ID、值是记录集ID字段值

的URL参数。

步骤04 完成后的链接地址如图3-48所示。

```
del.php?ID=<?php echo $info['ID'];?>
//传递ID到del.php页面
```



图 3-48 传递 ID 至 del.php

步骤05 创建del.php空白文档，使用PHP设置查询记录集代码，实现功能的PHP代码已经在网页中标注出来：

```
<?php require_once('conn.php'); ?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>删除页面</title>
<style type="text/css">
.aline_center { text-align: center;
}
.dc {
text-align: center;
}
</style>
</head>
<body>
<p class="aline_center">PHP成绩查询系统</p>
<hr/>
<form name="form1" method="post" action="savedel.php">
<table width="600" border="1" align="center" cellpadding="1" cellspacing="1">
<tr>
<td>序号</td>
<td>姓名</td>
<td>年龄</td>
<td>成绩</td>
</tr>
<?php
$ID=$_GET['ID'];
$sql=mysqli_query($conn,"select * from websql where ID='".$ID."'");
$info=mysqli_fetch_array($sql);
?>
// 查询记录集记录
<tr>
<td><?php echo $info['ID'];?></td>
<input name="ID" type="hidden" id="ID" value="<?php echo $info['ID'];?>" size="16" />
<!--加入ID隐藏域使用表单传递时能把ID值正确传给下一页-->
```

```

        <input type="hidden" name="id" value="<?php echo $info['ID'];?>">
        <td><?php echo $info['name']?></td>
        <td><?php echo $info['age']?></td>
        <td><?php echo $info['Result']?></td>
    </tr>
    <tr>
        <td colspan="4" class="dc"><input type="submit" name="button"
id="button" value="删除" /></td>
    </tr>
</table>
</form>
<hr />
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>

```

步骤06 最后创建savedel.php网页代码，使用delete命令实现数据库的删除操作。并使用header("location:index.php");命令实现删除成功后跳转到首页。

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<?php
include("conn.php");
$ID=$_POST['ID'];
mysqli_query($conn,"delete from websql where id='$ID'");
echo "<script>alert('删除成功!');</script>";
header("location:index.php");
?>

```

本章学习了最基本的PHP常用服务器行为的操作和使用，并且了解了原始程序代码的意义。在后面的章节中，如用户管理系统、留言簿管理系统、新闻管理系统等的实现都将用到这些基本操作。