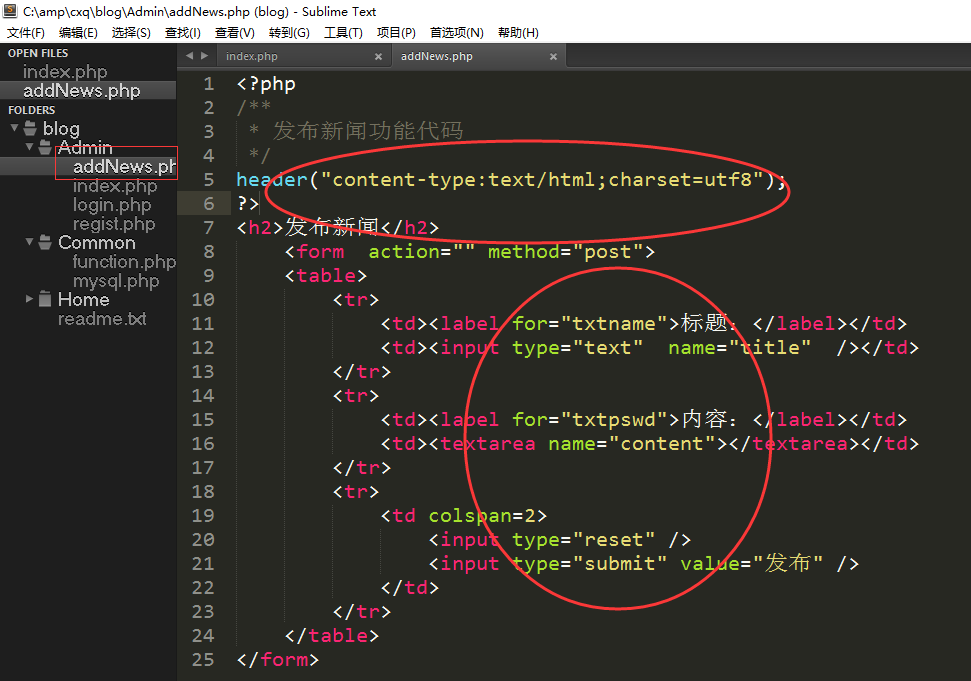
# 一、新闻模块

## 1.发布新闻

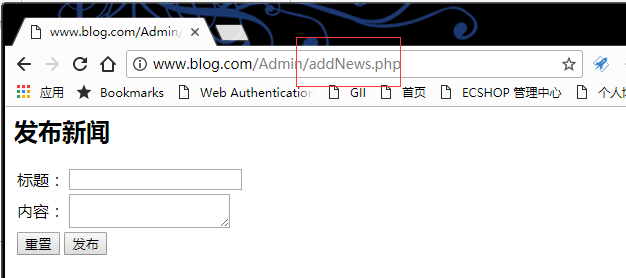
### 1.1发布新闻模板

创建/Admin/addNews.php，放入预先制作好的发布新闻模板



访问地址：<http://www.blog.com/Admin/addNews.php>

页面展示效果：

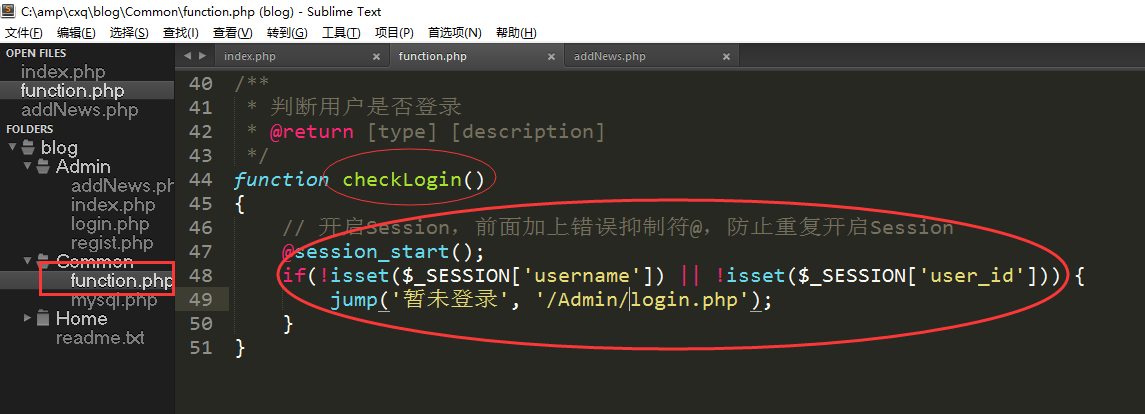


### 1.2发布新闻功能实现

#### 1.2.1封装判断用户是否登录代码

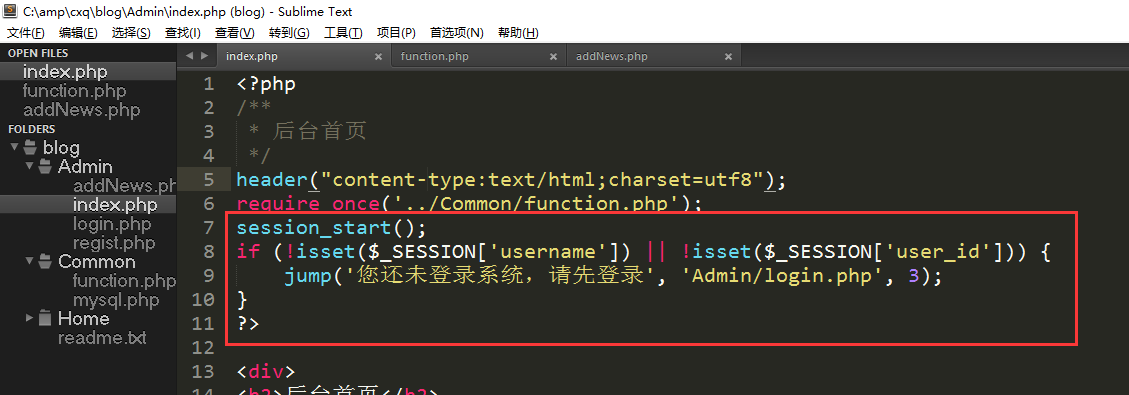
新闻发布页面也是需要判断用户是否登录的，必须是登录状态才能发布新闻，后面的一些其他功能页面也是要求必须是登录状态才可以完成，所以我们把判断用户是否登录的代码给抽离出来封装成一个自定义函数。

在/Common/function.php创建checkLogin()方法，代码如下：

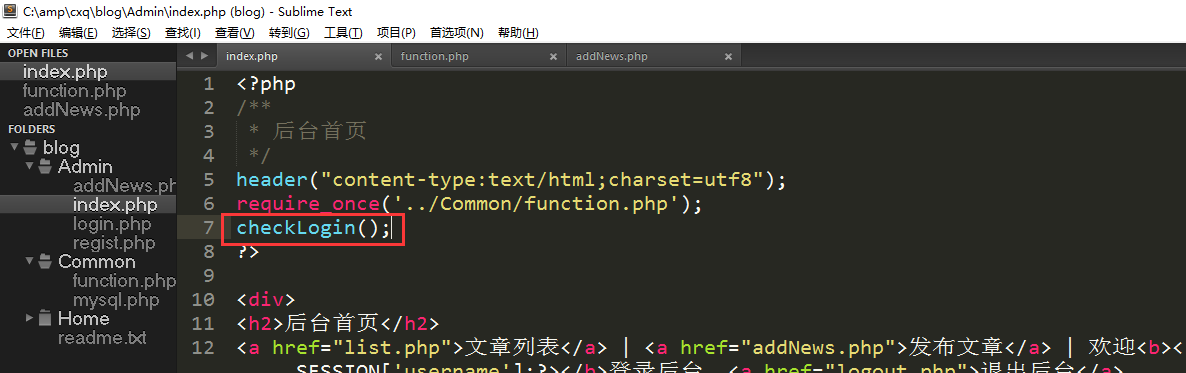


这样我们就可以把首页的代码进行简化。

Index.php修改之前的代码如下图：

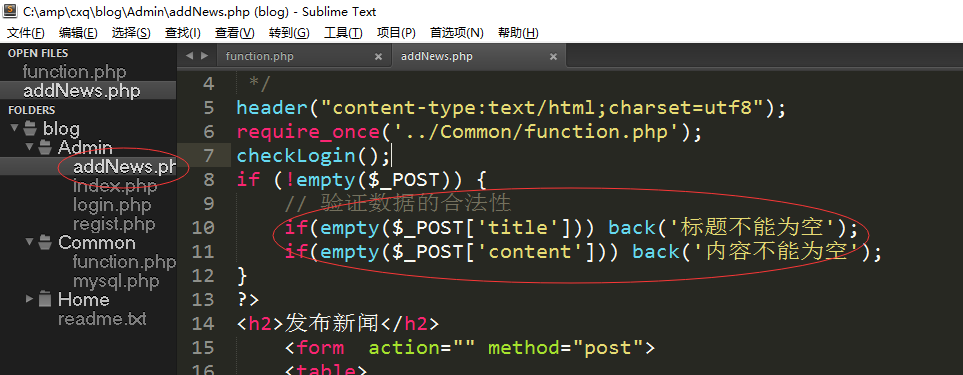


简化之后的代码：

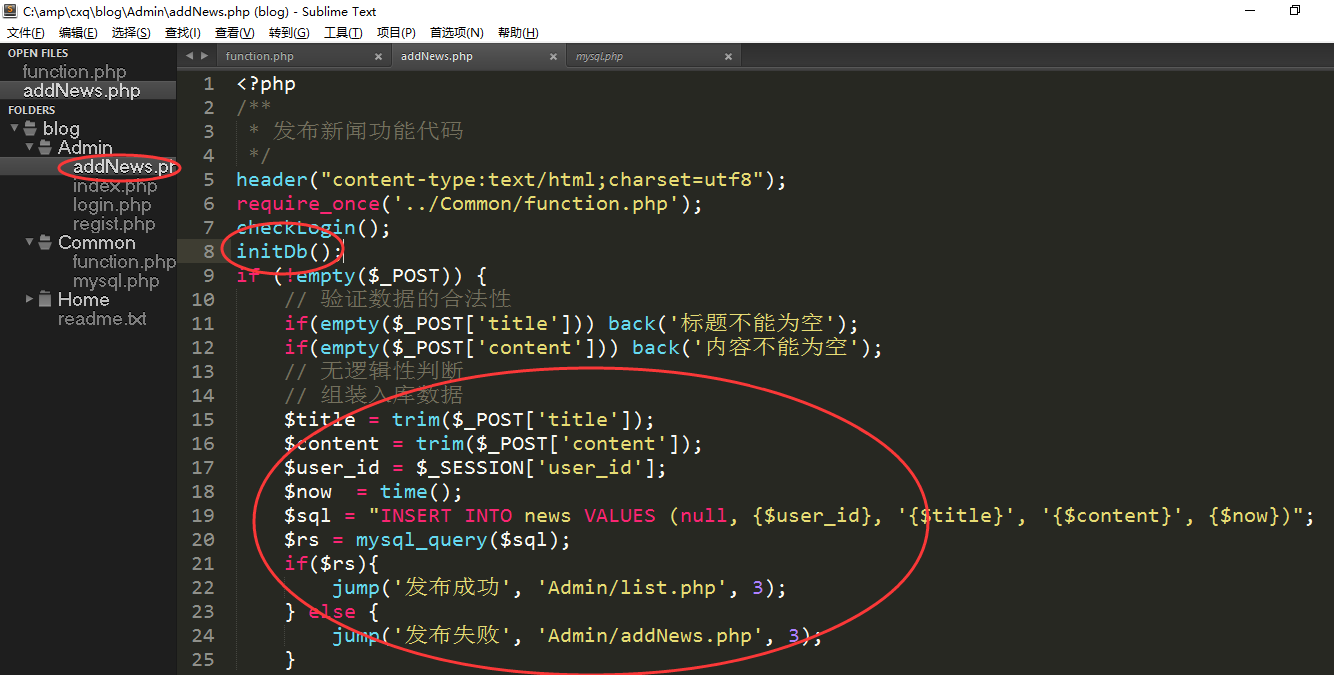


#### 1.2.2发布新闻

1) 验证数据的合法性



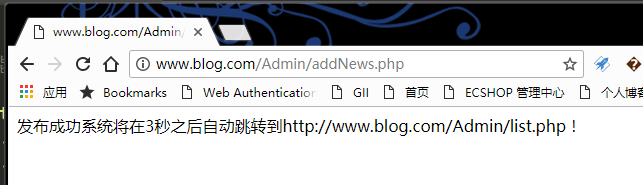
2)组装入库数据



页面呈现效果：



发布成功效果：



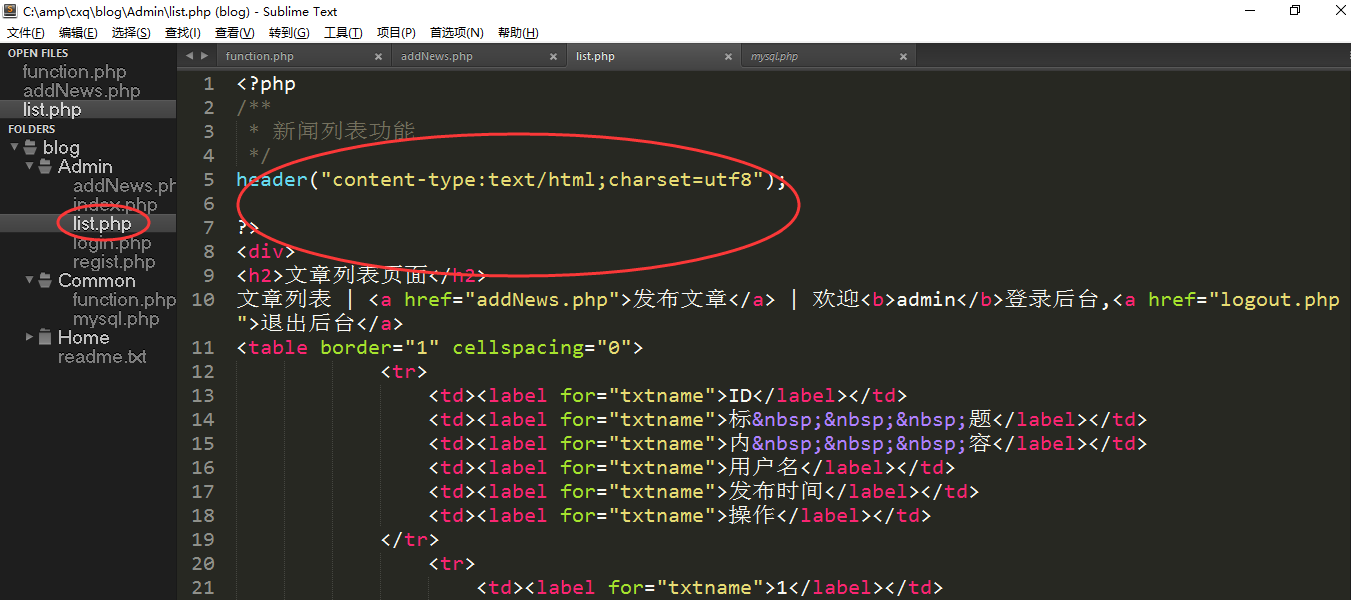
数据库呈现效果：



### 2.新闻列表

### 2.1新闻列表模板

创建/Admin/list.php，放入预先制作的列表页面。



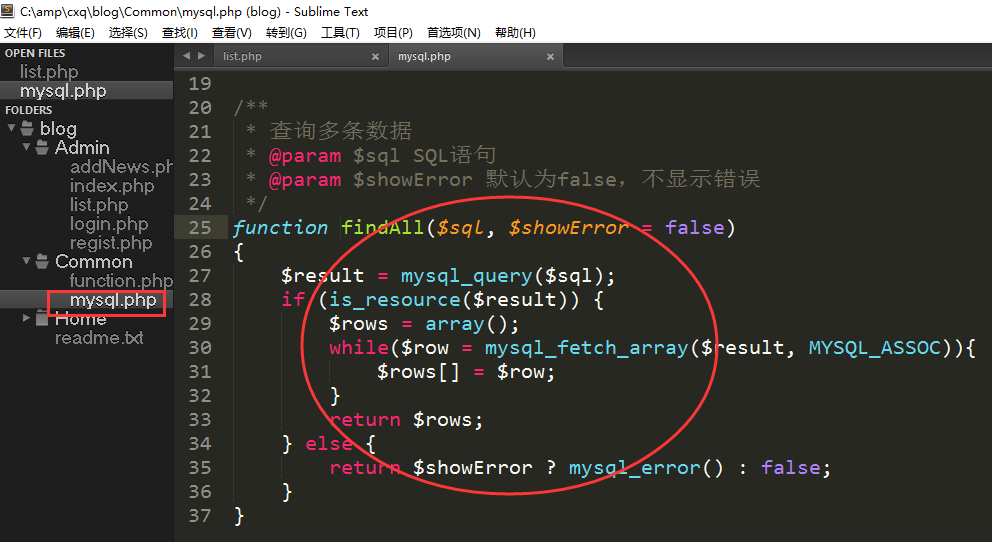
页面展示效果：



### 2.2新闻列表功能实现

#### 2.2.1封装查询多条的函数

在/Common/mysql.php封装findAll()方法，用于查询多条结果集，代码如下：



#### 2.2.2获取新闻列表

代码如下：

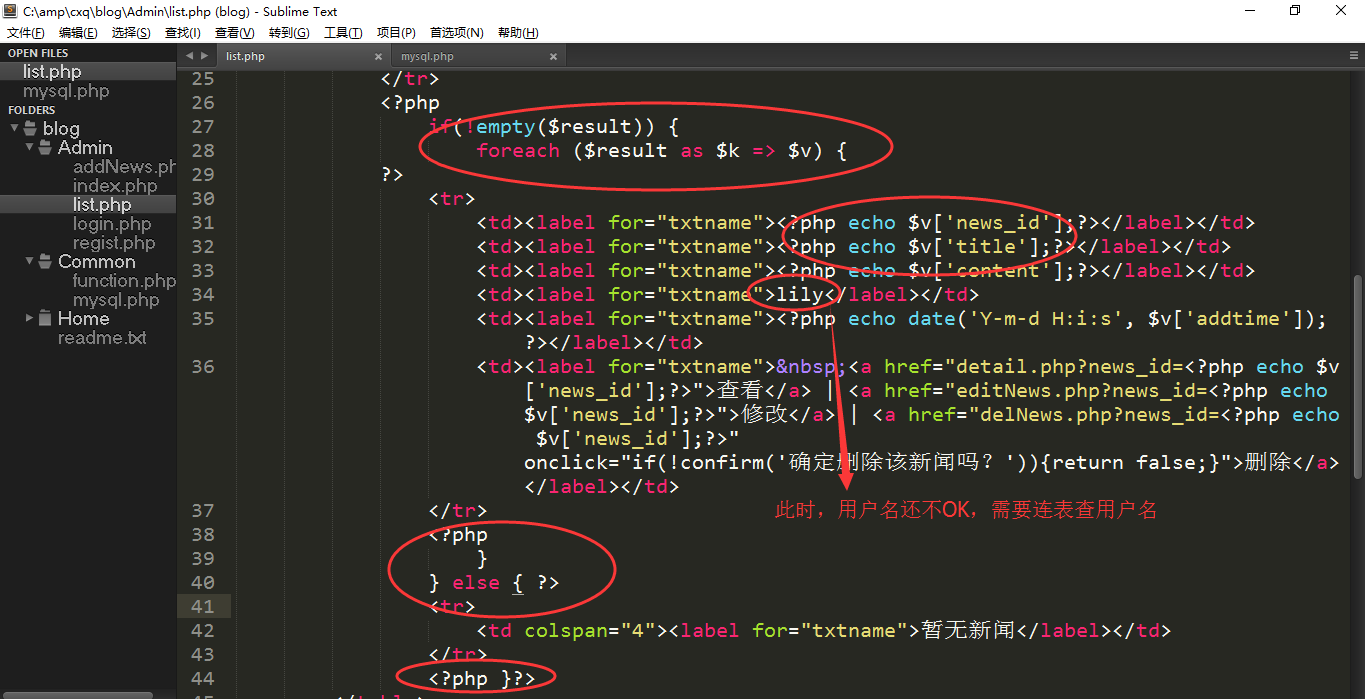


数据为二维数组，效果如下：



#### 2.2.3展示新闻列表

接下来就需要把查询到的新闻列表展示到模板上去，如果结果集为空，提示用户当前暂无新闻列表。



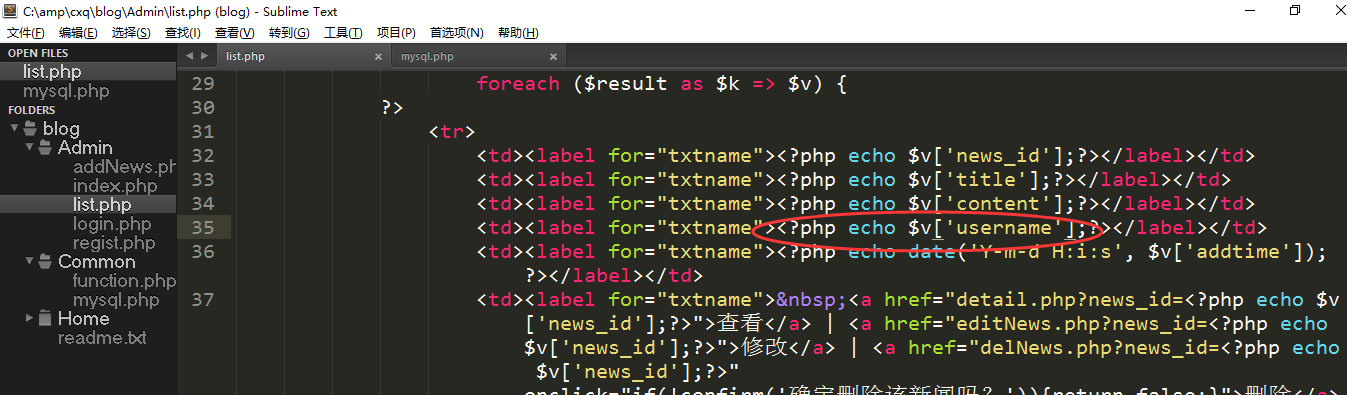
注意：给查看、修改、删除设置超链接。

#### 2.2.4连表查询用户名

新闻列表的用户名如果需要展示的话，则连表查出来，或者把用户名这一列删掉也可以，因为在后台用户只能为同一人。

连表代码如下：





页面呈现效果如下：

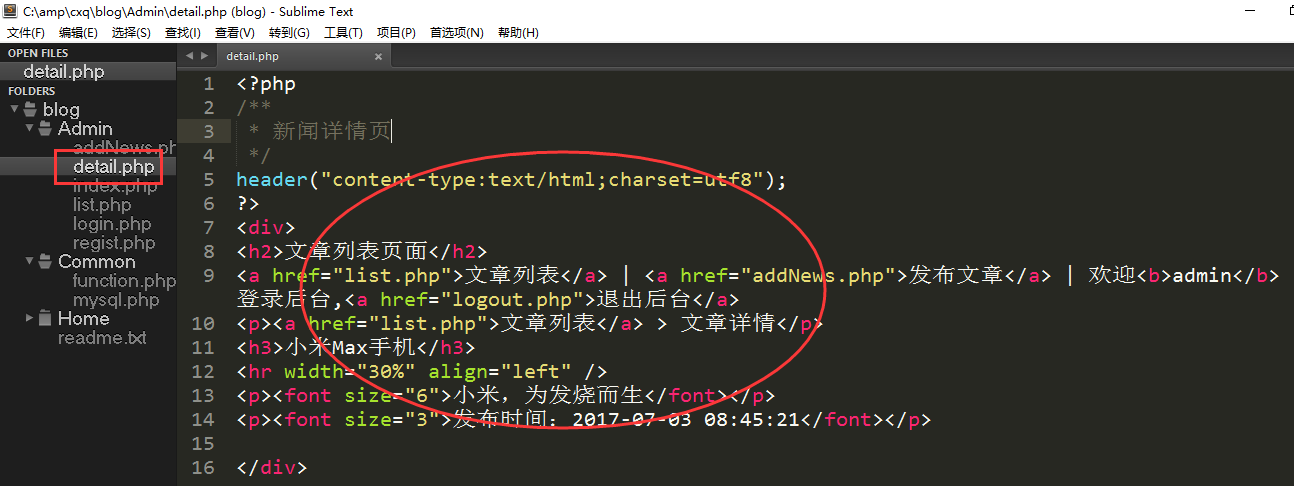


## 3.新闻详情

### 3.1新闻详情模板

创建/Admin/detail.php，放入预先制作好的模板。

代码如下：



页面展示效果如下：



### 3.2新闻详情功能实现

查询新闻详情需要根据新闻的news\_id和用户的user\_id。

为什么要加user\_id的限制呢？

答：防止用户恶意修改news\_id，去操作别人的新闻。

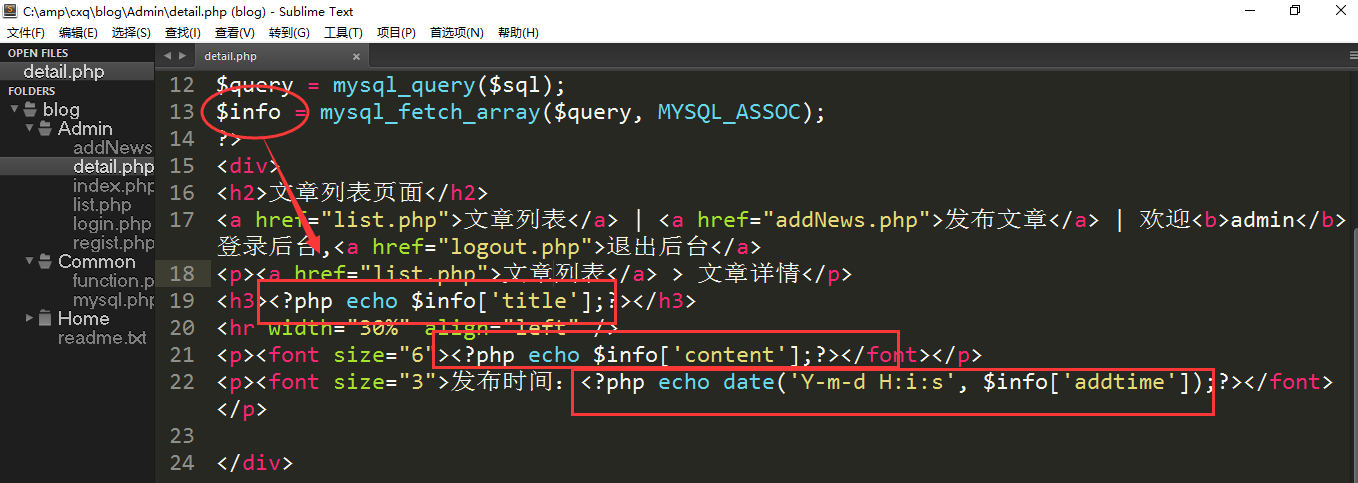
代码如下：



数据展示效果如下：



把数据$info替换原有的静态模板，代码如下：



页面展示效果如下：

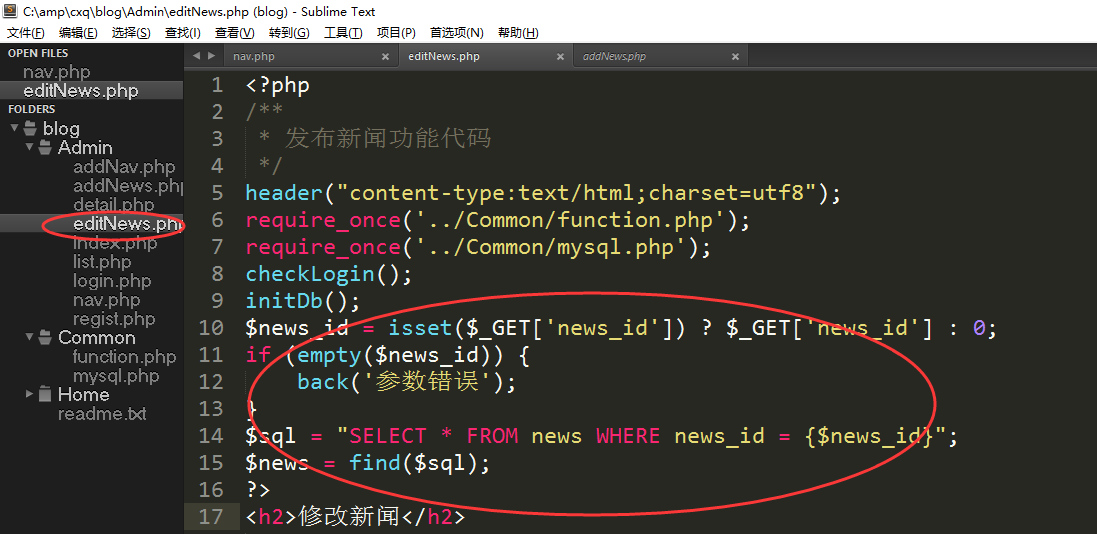


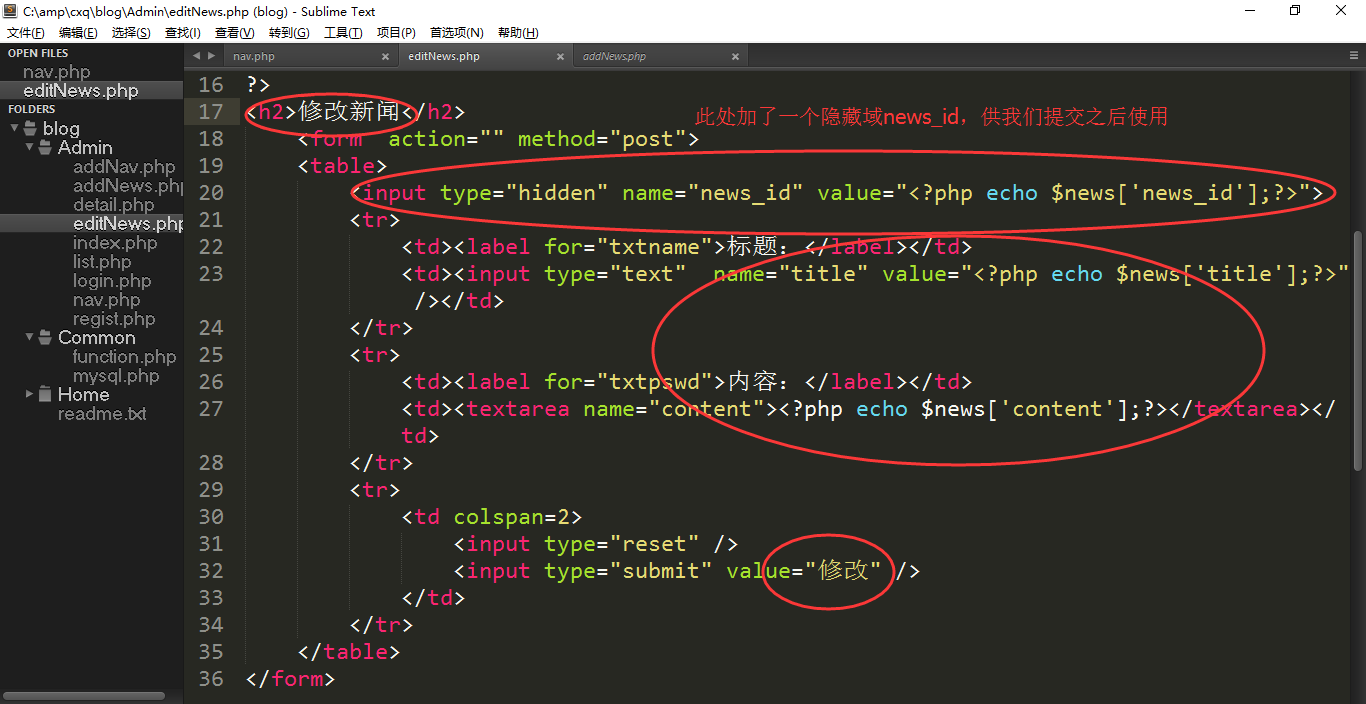
## 4.编辑新闻

点击编辑新闻，我们需要把原来旧的新闻详情给展示出来，当用户点击修改的时候把数据进行更新。

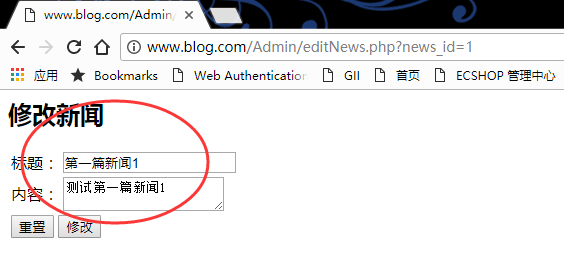
①根据新闻news\_id把新闻详情给展示出来。

创建/Admin/editNews.php，展示详情页代码如下：

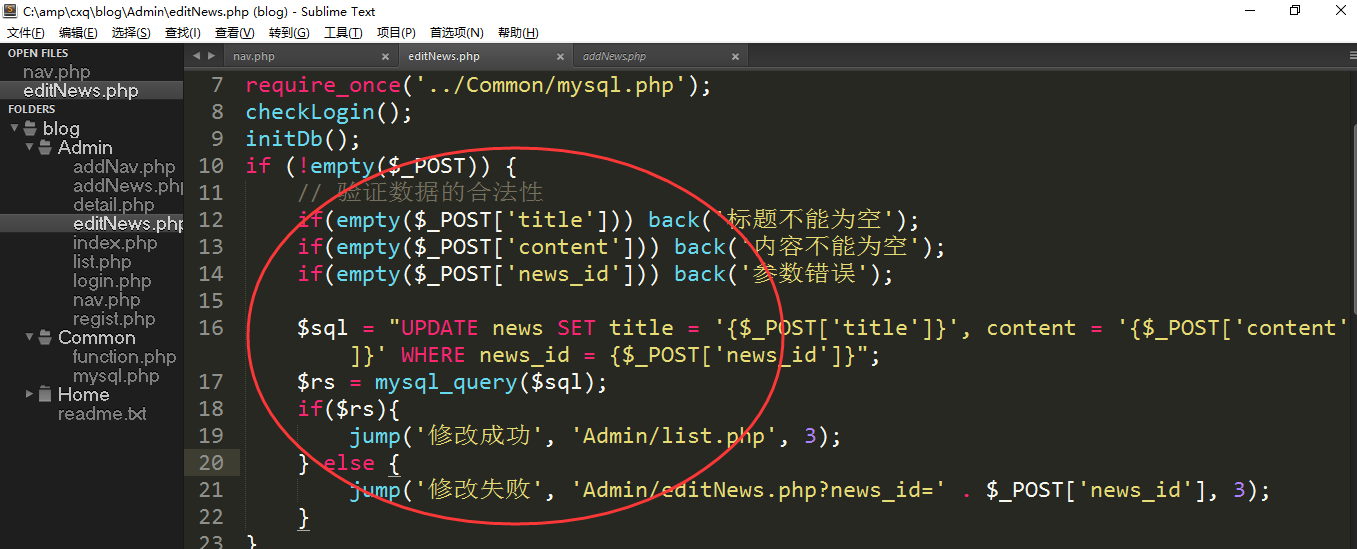




页面展示效果：



②用户提交表单，在editNews.php接收表单传值。

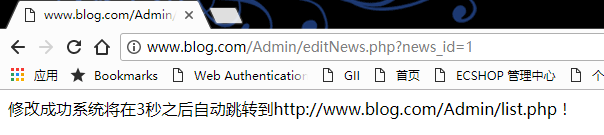


测试效果如下：

修改前：



修改中：



修改后：



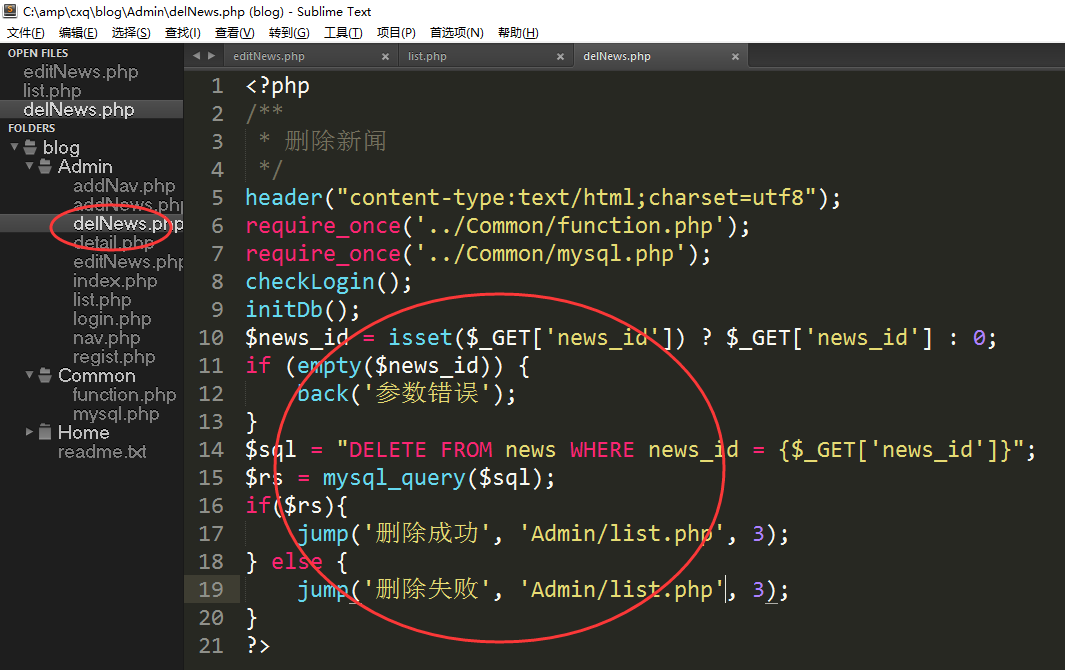
## 5.删除新闻

删除新闻也是根据news\_id来确定对应的新闻纪录，在删除之前需要给用户一个提示，是否删除？

给新闻列表的“删除”设置超链接：

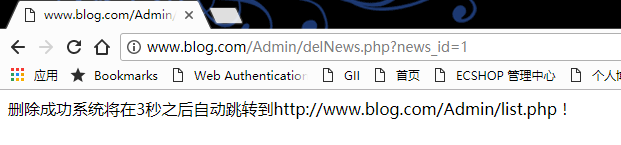


创建/Admin/delNews.php执行删除操作，代码如下：



测试效果：





此时，新闻列表已经没有第一条新闻了。



# 二、导航管理

## 1.添加导航

前台的导航，我们要能够在后台进行管理，那么就需要创建对应的导航表，保存对应的连接地址和顺序。

要求：

①导航最多有7个

②导航名称不能重复

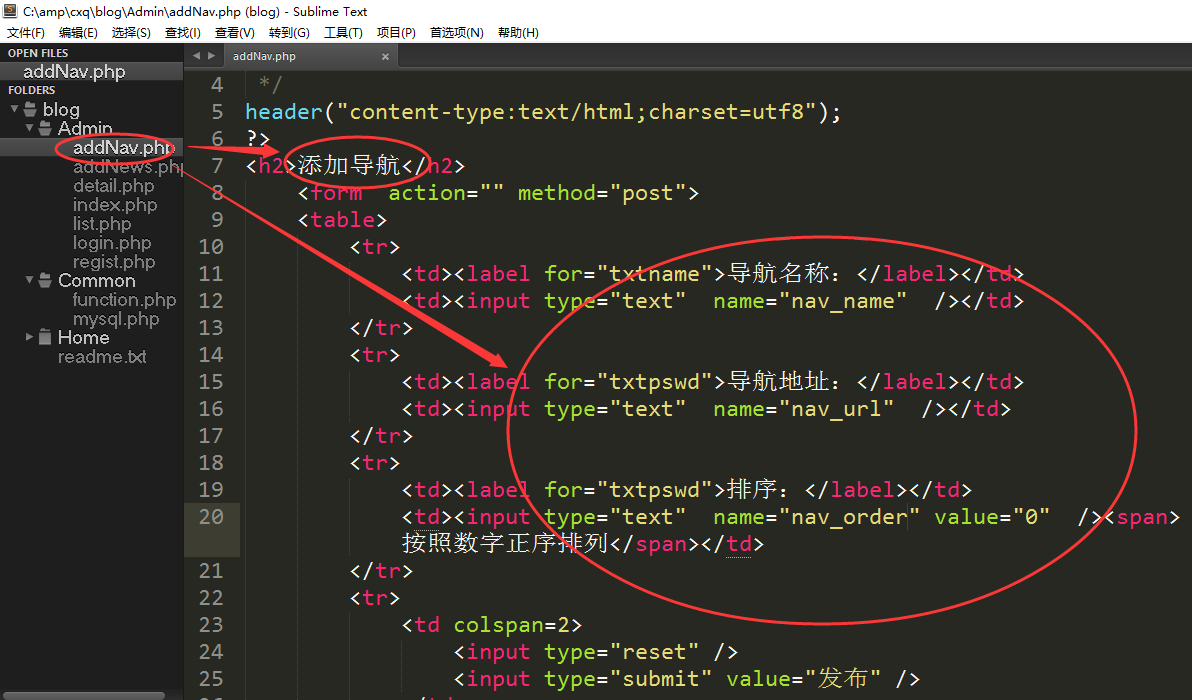
### 1.1创建导航表：

|  |
| --- |
| **CREATE** **TABLE** `nav` (  `id` **INT**(10) **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** '主键ID',  `nav\_name` **VARCHAR**(50) **NOT** **NULL** **COMMENT** '导航名称',  `nav\_url` **VARCHAR**(200) **NOT** **NULL** **COMMENT** '导航地址',  `nav\_order` **TINYINT**(3) **NOT** **NULL** **DEFAULT** '0' **COMMENT** '排序',  `update\_time` **INT**(11) **UNSIGNED** **NOT** **NULL** **DEFAULT** '0' **COMMENT** '更新时间',  **PRIMARY** **KEY** (`id`),  **UNIQUE** **INDEX** `nav\_name` (`nav\_name`)  )  **COMMENT**='导航表'  **ENGINE**=**InnoDB**  ; |

由于导航不能重复，所以我们给nav\_name导航名称字段加了一个“唯一索引”unique，保证该字段的唯一性。

### 1.2添加导航模板

创建/Admin/addNav.php，把发布新闻的模板拿过来改造一下，代码如下：



访问地址：<http://www.blog.com/Admin/addNav.php>

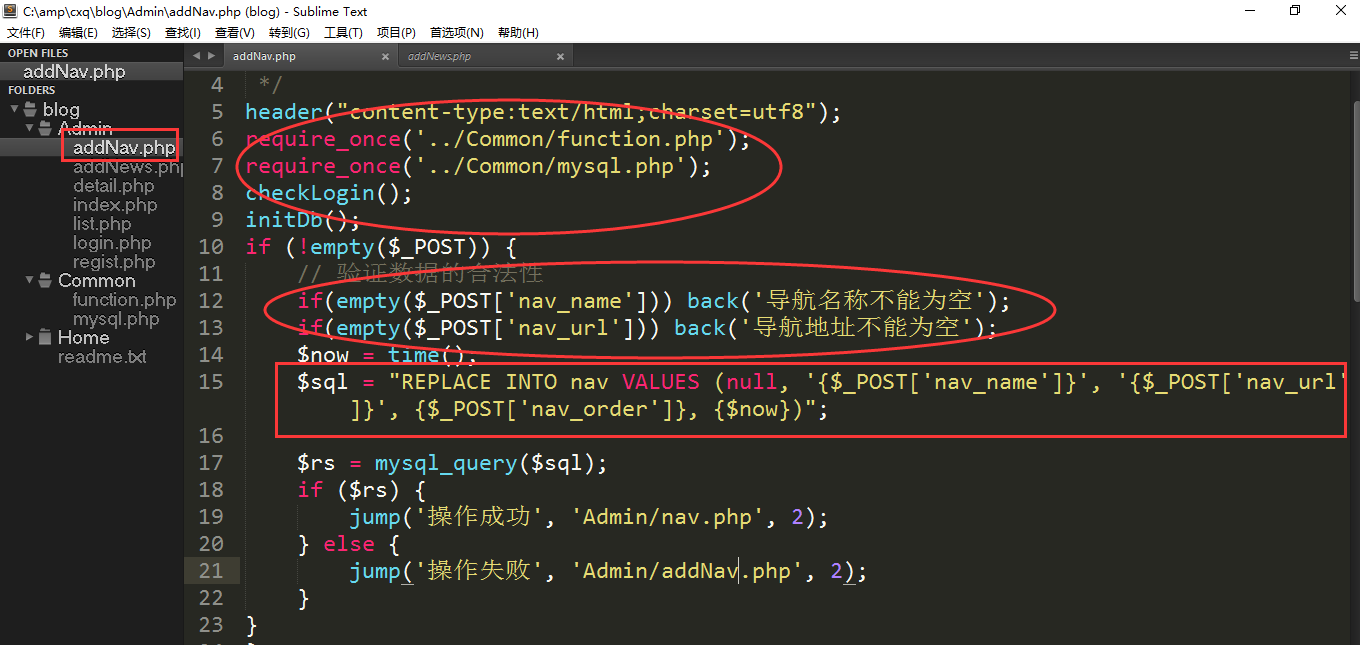
页面展示效果如下：



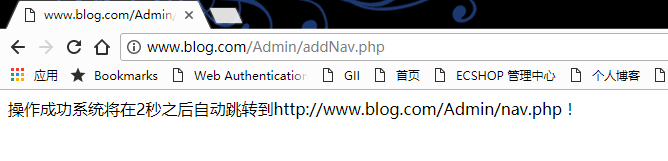
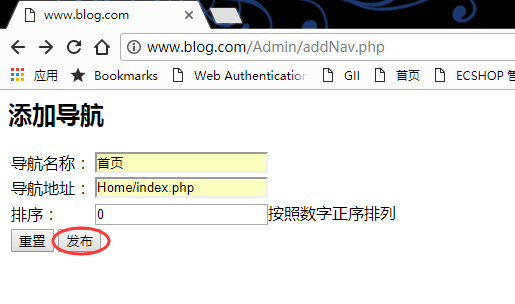
### 1.3实现添加导航功能

代码如下：

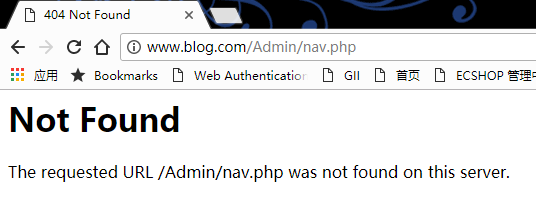
这个地方由于导航名称不能重复，所以插入语句使用了一个replace into，插入一条存在的导航名称的时候执行更新操作，以免造成冲突。



测试效果：

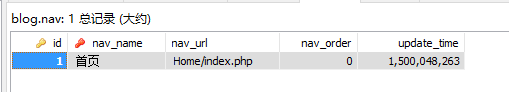


然后跳转到/Admin/nav.php



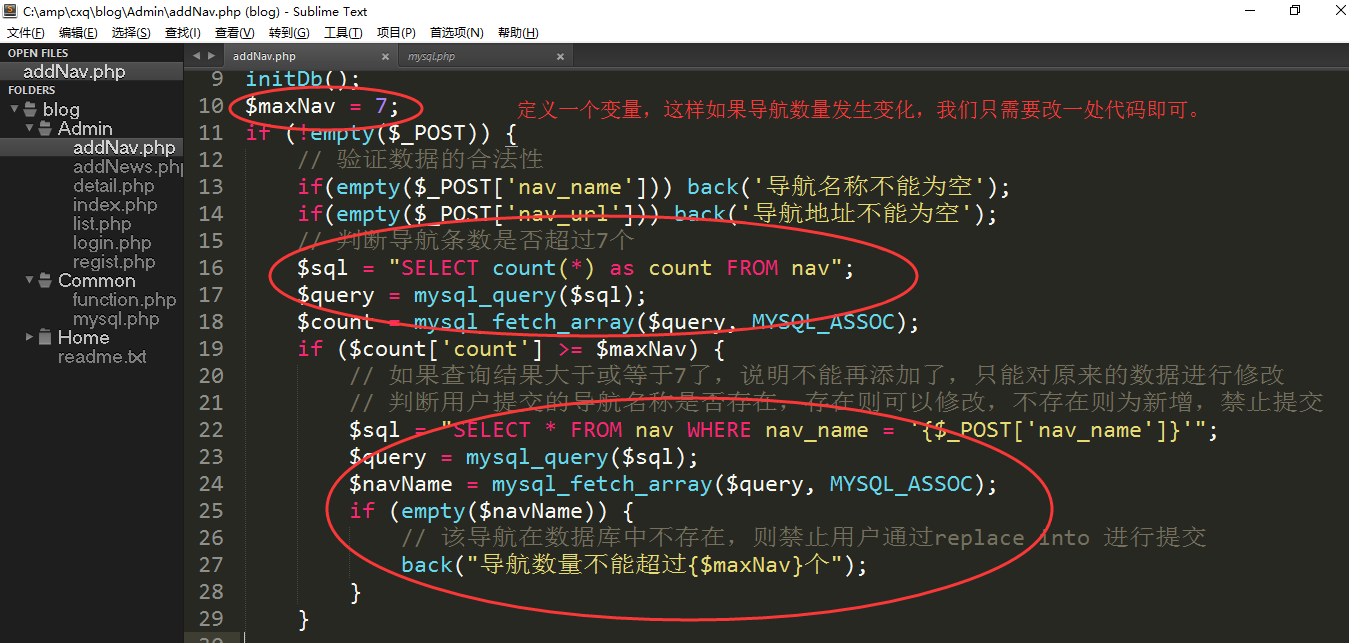
由于还没有制作导航管理（列表）页面，所以会报错nav.php找不到，这个会在随后去完善。

数据库呈现效果：



### 1.4导航总数限制

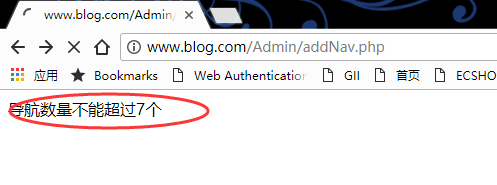
为了保证前台页面正常显示，导航的数量不能超过7个，需要在添加的时候进行判断，如果已经存在7个了则不允许再次添加新的。



测试效果如下：



点击发布之后效果：

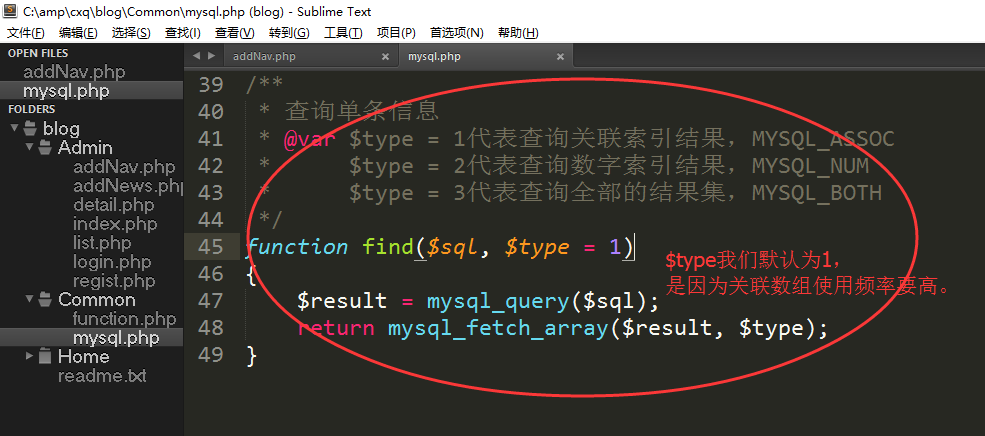


### 1.5封装查询单条数据的函数

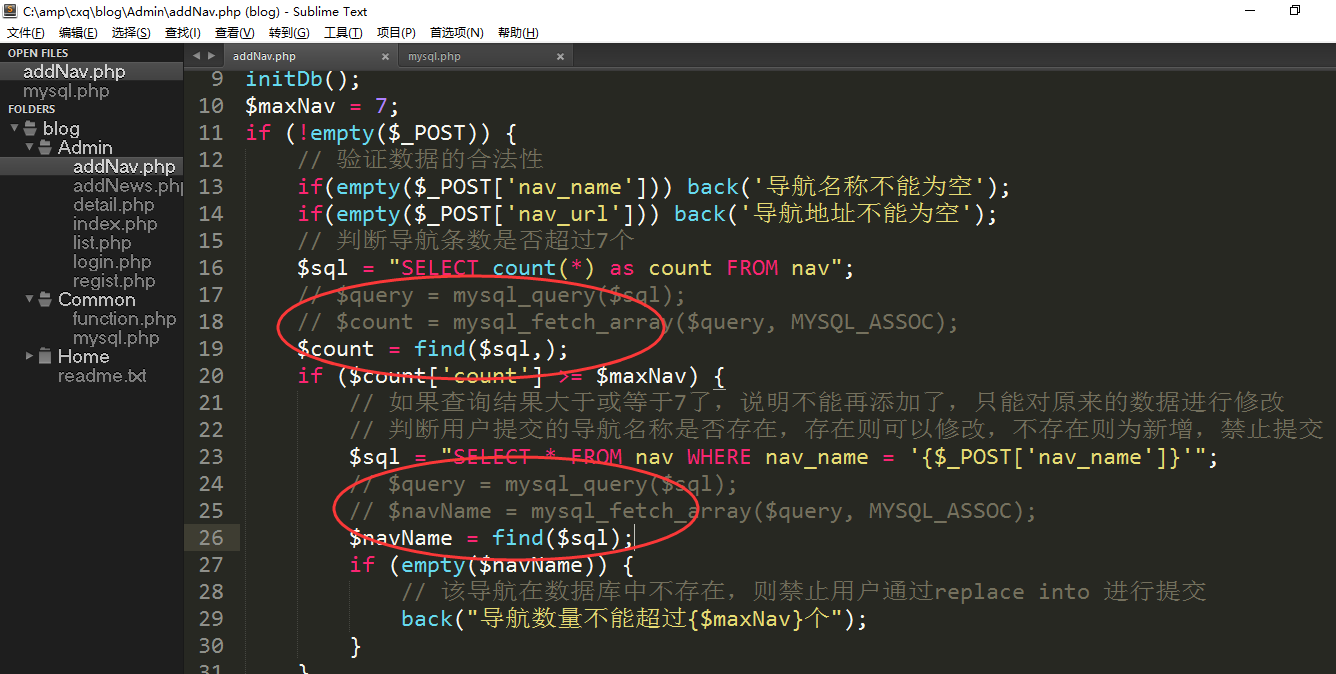
在以上的操作我们经常使用如下两行代码来获取一条记录信息，为了避免代码重复，我们可以把这两行代码进行封装。

|  |
| --- |
| $query = mysql\_query($sql);  $navName = mysql\_fetch\_array($query, MYSQL\_ASSOC); |

封装到/Common/mysql.php 下面的find()方法。



此时我们的添加导航addNav.php的代码则可以优化为如下：



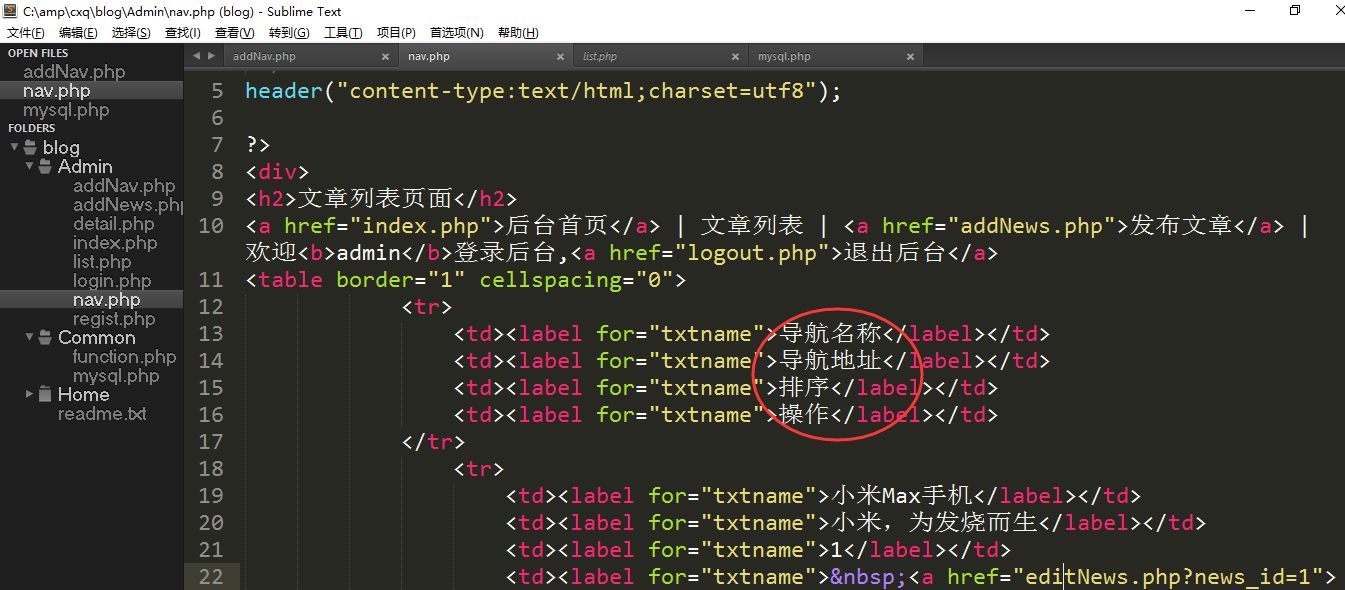
效果是一样的。

## 2.导航列表

### 2.1制作导航列表模板

同样导航列表模板，我们可以使用新闻列表来制作。

先去创建/Admin/nav.php，然后把新闻列表的模板放入文件中，进行适当修改，代码如下图所示：



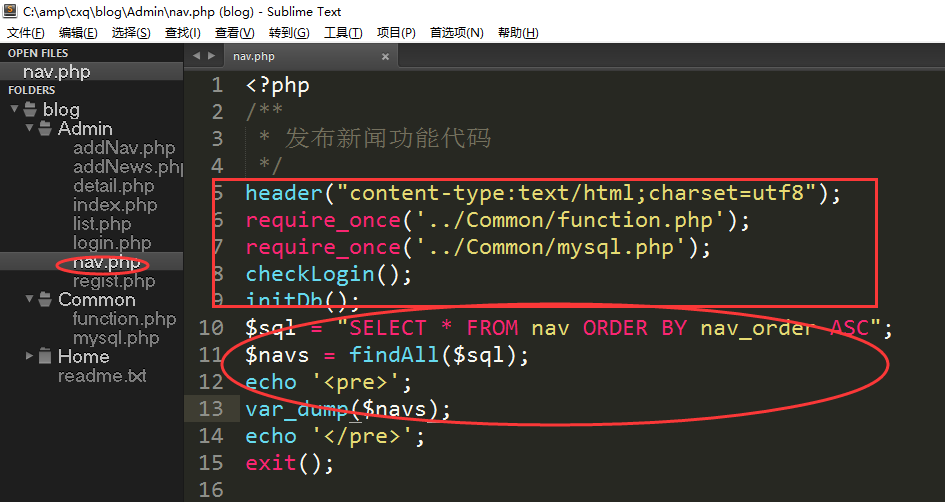
页面呈现效果：



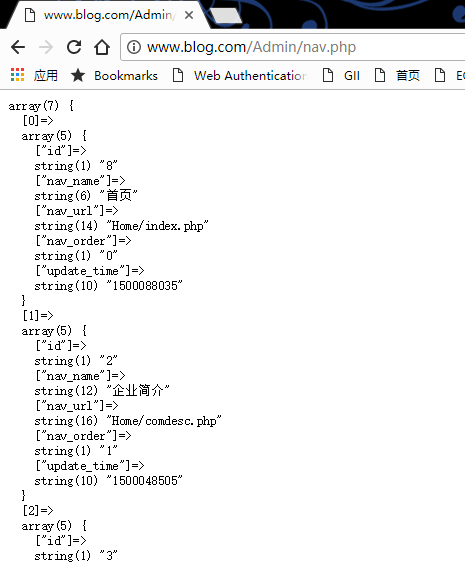
### 2.2导航列表功能实现

查询导航列表的时候需要注意的是按照“nav\_order”字段进行升序排列。

代码如下图所示：



返回的是一个二维数据，数据展示效果：



把结果进行遍历，在页面进行展示，代码如下：



页面展示效果如下：



## 3.导航修改和删除

自行维护，课堂上不再演示。原理跟新闻的修改和删除是一样的，作为扩展内容，同学们自由发挥即可。