

Mysql 扩展

PHP 针对 MySQL 数据库操作提供的扩展：允许 PHP 当做 MySQL 的一个客户端连接服务器进行操作。

连库基本操作

连接数据库服务器

1) 资源 `mysql_connect` (服务器地址, 用户名, 密码)

```
//建立连接
//1、主机地址：默认是自动连接端口3306
$link = mysql_connect('localhost:3306','root','root');
var_dump($link);
```



连接资源默认也是超全局的，任何地方都可以使用该资源进行数据库的其他操作

补充：`mysql_connect` 默认是对一个服务器只连接一次（如果再次运行，返回的是同一个资源），但是如果有需要连接同一个资源两次，那么可以使用第四个参数：`TRUE`

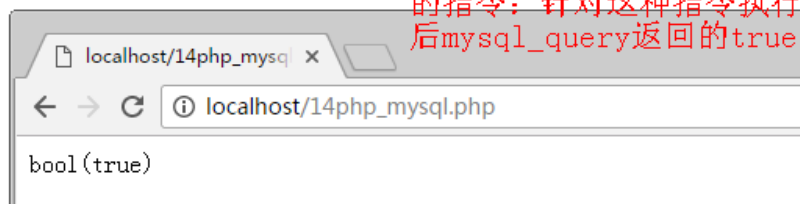
设置连接编码

保持客户端与服务器之间的沟通顺畅：同一“语言”（字符集）

1) 形式 1: `mysql_query("set names XXX")`

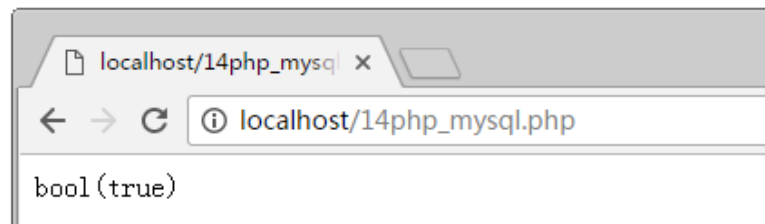
```
//设定字符集
$res = mysql_query('set names utf8',$link);
var_dump($res);
```

当前指令属于没有返回结果的指令：针对这种指令执行后mysql_query返回的true



2) 形式 2: `mysql_set_charset("XXX")`

```
// $res = mysql_query('set names utf8', $link);
$res = mysql_set_charset('utf8');
var_dump($res);
```



3) 怎么确定要使用何种编码

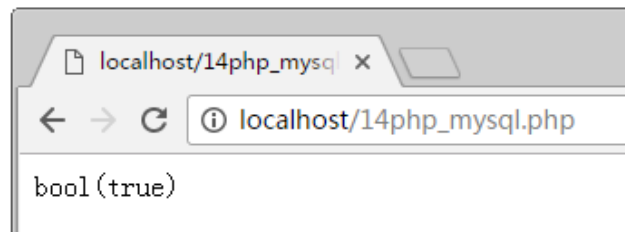
客户端当前执行脚本的界面是什么字符集，就设定成什么字符集

选定要使用的数据库

1) 形式 1: `mysql_query("use XXX")`

2) 形式 2: `mysql_select_db("XXX")`

```
// 选择数据库
$res = mysql_query('use test');
var_dump($res);
```



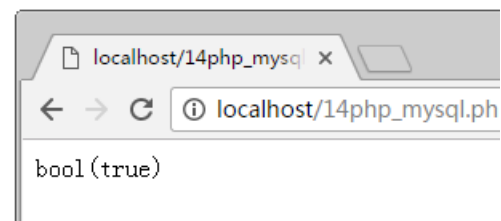
关闭连接

主动释放连接: MySQL 服务器的连接资源是有限的, 不用了需要释放 (脚本执行结束系统也会自动释放)

1) `mysql_close()`

// 数据库操作: 对数据库的增删改查

```
// 关闭资源
$res = mysql_close($link);
var_dump($res);
```



执行增删改操作

执行增删改操作

Mysql_query 函数执行结果返回的是 true 或者 false，true 代表执行成功，false 代表执行失败：失败原因分两种：一种是 SQL 指令本身语法错误；第二种是执行失败；

```
1 create database News charset utf8;
2
3 create table n_news(
4 id int primary key auto_increment,
5 title varchar(50) not null comment '新闻标题',
6 isTop tinyint not null comment '是否置顶',
7 content text comment '内容',
8 publiser varchar(20) not null comment '发布人',
9 pub_time int not null comment '发布时间'
10 )charset utf8;
```



```
管理员: 命令提示符 - mysql -uroot -proot
-> > charset utf8;
ERROR 1046 (3D000): No database selected
mysql> use News;
Database changed
mysql> create table n_news(
-> id int primary key auto_increment,
-> title varchar(50) not null comment '新闻标题',
-> isTop tinyint not null comment '是否置顶',
-> content text comment '内容',
-> publiser varchar(20) not null comment '发布人',
-> pub_time int not null comment '发布时间'
-> )charset utf8;
Query OK, 0 rows affected (0.52 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_news |
+-----+
| n_news          |
+-----+
1 row in set (0.00 sec)

mysql>
```

做个专门负责数据库初始化的脚本：之后要操作数据库包含该脚本即可

```
<?php

//做数据库初始化
header('Content-type:text/html;charset=utf-8');

//连接认证
mysql_connect('localhost:3306','root','root') or die('数据库连接失败');

//设定字符集
mysql_query('set names utf8');

//选择数据库
mysql_query('use News');
```

I

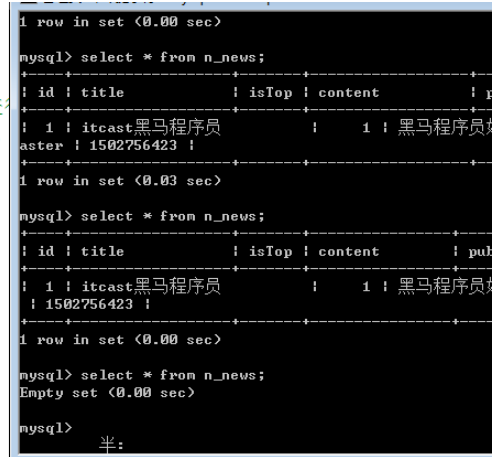
1) mysql_query("insert"): 通过 insert 指令向指定表插入对应的数据

```
5 //引入初始文件
5 include_once '16database.php';
7
3 //组织SQL指令
9 $pub_time = time();
9 $sql = "insert into n_news values(null,'itcast黑马程序员',1,'黑马程序员好样的','Itcaster',{ $pub_time})";
1
2 //执行SQL指令
3 if(mysql_query($sql)){
4 //操作成功: 通常是返回自增长ID给用户
5 echo '数据插入成功!';
5
7 }else{
3 //操作失败
9 echo '数据插入失败!';
9 }
```

I

2) mysql_query("delete"): 根据删除指令操作数据库 (表)

```
5 //引入初始文件
6 include_once '16database.php';
7
8 //组织SQL指令: 实际删除时通常也是根据ID进行
9 $sql = "delete from n_news where id = 1";
10
11 //执行SQL指令
12 if(mysql_query($sql)){
13     //操作成功
14     echo '数据删除成功!';
15 }else{
16     //操作失败
17     echo '数据删除失败!';
18 }
19 }
```



The screenshot shows a MySQL command prompt with the following commands and output:

```
mysql> select * from n_news;
+----+-----+-----+-----+
| id | title | isTop | content | publisher | pub_time |
+----+-----+-----+-----+
| 1 | itcast黑马程序员 | 1 | 黑马程序员好样的 | 1502756423 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from n_news;
+----+-----+-----+-----+
| id | title | isTop | content | publisher | pub_time |
+----+-----+-----+-----+
| 1 | itcast黑马程序员 | 1 | 黑马程序员好样的 | 1502756423 |
+----+-----+-----+-----+
1 row in set (0.03 sec)

mysql> select * from n_news;
Empty set (0.00 sec)

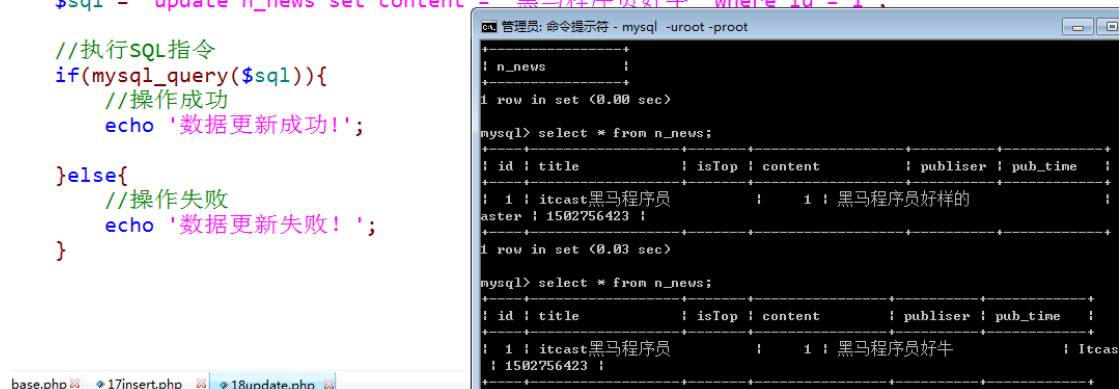
mysql>
```

3) mysql_query("update"): 根据更新指令操作数据库

```
//引入初始文件
include_once '16database.php';

//组织SQL指令: 实际更新时通常也是根据ID进行更新, 但是ID都是传递过来
$sql = "update n_news set content = '黑马程序员好牛' where id = 1";

//执行SQL指令
if(mysql_query($sql)){
    //操作成功
    echo '数据更新成功!';
}else{
    //操作失败
    echo '数据更新失败!';
}
```



The screenshot shows a MySQL command prompt with the following commands and output:

```
mysql> select * from n_news;
+----+-----+-----+-----+
| id | title | isTop | content | publisher | pub_time |
+----+-----+-----+-----+
| 1 | itcast黑马程序员 | 1 | 黑马程序员好样的 | 1502756423 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from n_news;
+----+-----+-----+-----+
| id | title | isTop | content | publisher | pub_time |
+----+-----+-----+-----+
| 1 | itcast黑马程序员 | 1 | 黑马程序员好牛 | 1502756423 |
+----+-----+-----+-----+
1 row in set (0.03 sec)
```

判断执行结果

- 1) 执行成功, 结果为 true, 只代表 SQL 指令执行成功
- 2) 执行失败, 结果为 false, 代表 SQL 指令错误或者操作对象不存在

```

L0
L1      //执行SQL指令
L2      if(mysql_query($sql)){
L3          //操作成功
L4          echo '数据删除成功!';
L5      |
L6      }else{
L7          //操作失败
L8          echo '数据删除失败!';
L9      }

```

执行查询操作

执行查询语句

1) mysql_query("select ...")

2) 其他类似查询语句，比如：show 语句，desc 语句

以上两种情况的总结：凡是执行操作希望拿到数据库返回的数据进行展示的（结果返回：数据结果）

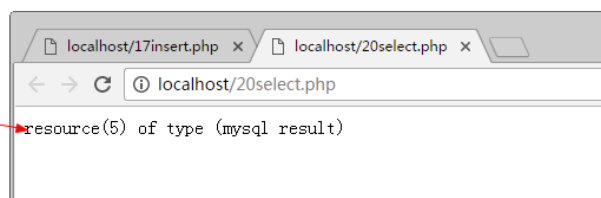
3) 执行结果的处理：成功为结果集，失败为 false

成功返回结果：SQL 指令没有错误，但是如果查询结果本身为空返回也是 true（结果集是一种资源：转换成 bool 永远为 TRUE）

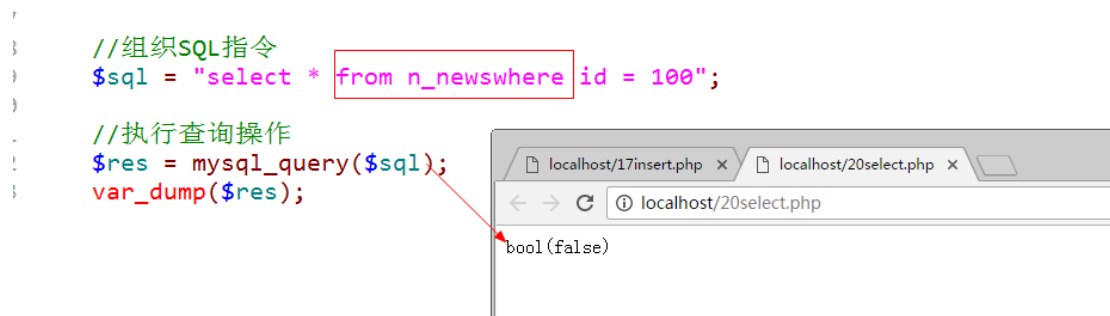
```

3      //数据库操作：查询
4
5      //引入初始化操作
6      include_once '16database.php';
7
8      //组织SQL指令
9      $sql = "select * from n_news";
10
11     //执行查询操作 操作正确
12     $res = mysql_query($sql);
13     var_dump($res);

```

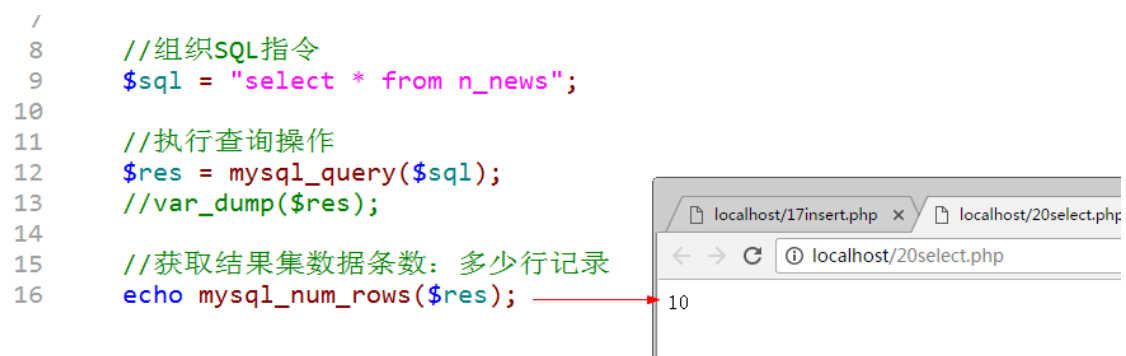


失败为 false：SQL 指令有错误



获取结果集行数

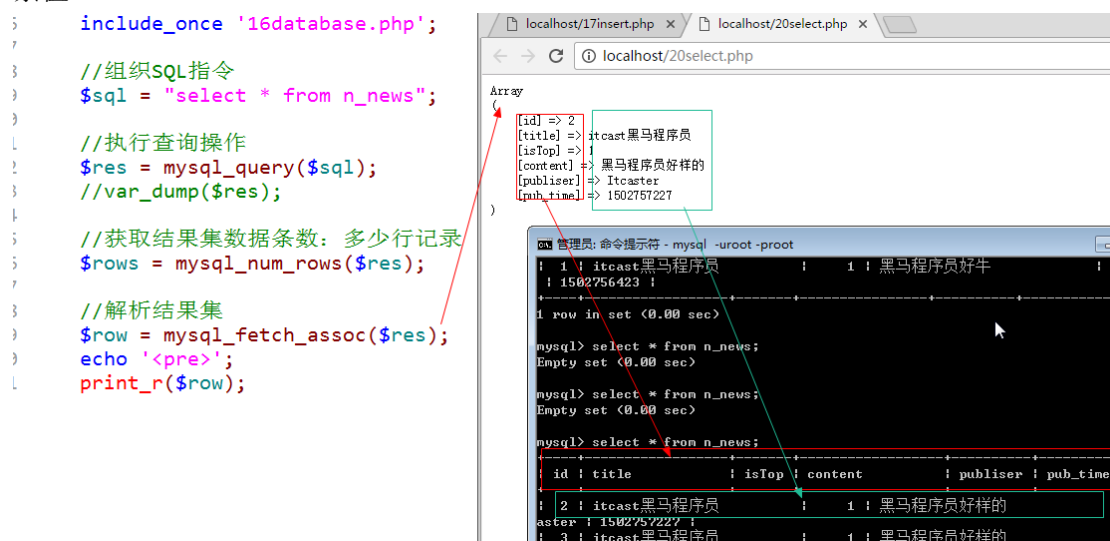
1) mysql_num_rows(): 获取结果集中到底有多少行记录



解析结果集

将一种结果集资源（PHP 不能直接使用），转换成一种 PHP 能够解析的数据格式：通过从结果集中（结果集指针：类似数组指针），按照结果集指针所在位置取出对应的一条记录（一行），返回一个数组，同时指针下移…直到指针移出结果集。

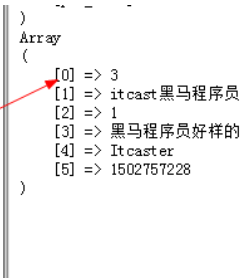
1) mysql_fetch_assoc(): 获取关联数组，表的表单名字作为数组下标，元素值作为数组元素值



2) `mysql_fetch_row()`: 获取索引数组, 只获取数据的值, 然后数组的下标从 0 开始自动索引

```
//解析结果集
$row = mysql_fetch_assoc($res);
echo '<pre>';
print_r($row);

$row = mysql_fetch_row($res);
print_r($row);
```



```
Array
(
    [0] => 3
    [1] => itcast黑马程序员
    [2] => 1
    [3] => 黑马程序员好样的
    [4] => Itcaster
    [5] => 1502757228
)
```

3) `mysql_fetch_array()`: 获取关联或者索引数组, 但是默认是同时存在: 一个记录取两次, 形成一组是关联数组, 一组是索引数组; 但是可以通过第二个参数来决定获取的方式:

`MYSQL_ASSOC` 只获取关联数组; `MYSQL_NUM` 只获取索引数组; `MYSQL_BOTH` 获取全部两种。

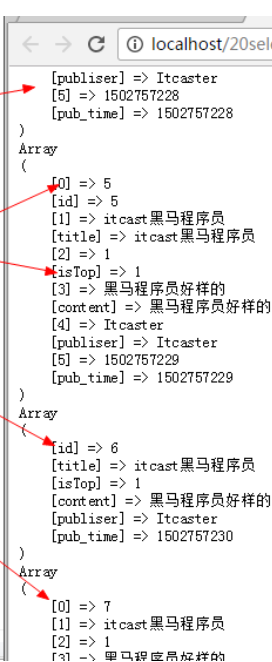
```
$row = mysql_fetch_row($res);
//print_r($row);

//MYSQL_FETCH_ARRAY
$row = mysql_fetch_array($res);
print_r($row);

//获取两种
$row = mysql_fetch_array($res,MYSQL_BOTH);
print_r($row);

//获取关联
$row = mysql_fetch_array($res,MYSQL_ASSOC);
print_r($row);

//获取索引
$row = mysql_fetch_array($res,MYSQL_NUM);
print_r($row);
```



```
[publiser] => Itcaster
[5] => 1502757228
[pub_time] => 1502757228
)
Array
(
    [0] => 5
    [id] => 5
    [1] => itcast黑马程序员
    [title] => itcast黑马程序员
    [2] => 1
    [isTop] => 1
    [3] => 黑马程序员好样的
    [content] => 黑马程序员好样的
    [4] => Itcaster
    [publiser] => Itcaster
    [5] => 1502757229
    [pub_time] => 1502757229
)
Array
(
    [id] => 6
    [title] => itcast黑马程序员
    [isTop] => 1
    [content] => 黑马程序员好样的
    [publiser] => Itcaster
    [pub_time] => 1502757230
)
Array
(
    [0] => 7
    [1] => itcast黑马程序员
    [2] => 1
    [3] => 黑马程序员好样的
```

通常结果集的操作: 一般是获取里面的所有 (全部记录)

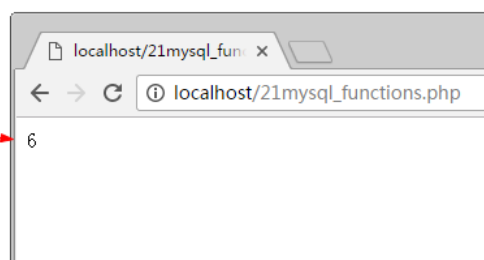
其他相关函数

有关字段信息

1) `mysql_num_fields()`: 获取一个指定结果集中所有的字段数

```
// 继续查询操作
$sql = "select * from n_news";
$res = mysql_query($sql);

// 查询当前结果集中所有的字段数量
echo mysql_num_fields($res);
```



2) mysql_field_name(): 获取一个指定结果集中指定位置字段的名称

```
// 查询当前结果集中所有的字段数量
echo mysql_num_fields($res);

// 查询指定列的字段名称
echo mysql_field_name($res, 1);
```



有关出错信息

1) mysql_error(): 获取出错对应的提示信息

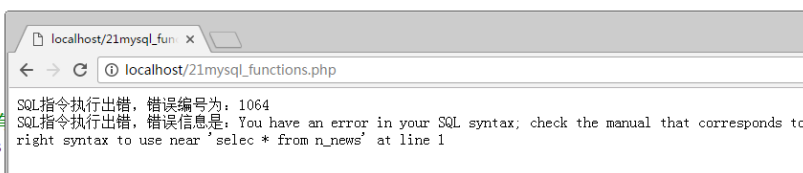
2) mysql_errno(): 获取出错对应的错误提示代号

错误的判断: 基于 Mysql_query 这个函数执行的结果: 结果返回 false 就代表执行错误。

```
$res = mysql_query($sql);

// 错误判定
if(!$res){
    // 代表结果为false
    echo 'SQL指令执行出错, 错误编号为: ' . mysql_errno() . '<br/>';
    echo 'SQL指令执行出错, 错误信息是: ' . mysql_error() . '<br/>';
    // 终止代码执行
    exit();
}
```

```
// 查询当前结果集中所有
echo mysql_num_fields($res);
```



其他函数

1) mysql_insert_id(): 获取上次插入操作所产生的自增长 ID, 如果没有自增长 ID 返回 0

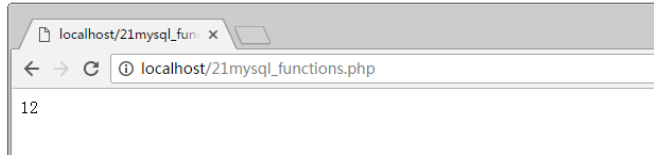

```

//获取自增长ID
$pub_time = time();
$sql = "insert into n_news values(null,'itcast好时光',2,'黑马程序员好样的，都知道','Itcaster',{ $pub_time})";

$res = mysql_query($sql);
if(!$res){
    //代表结果为false
    echo 'SQL指令执行出错，错误编号为: ' . mysql_errno() . '<br/>';
    echo 'SQL指令执行出错，错误信息是: ' . mysql_error() . '<br/>';
    //终止代码执行
    exit();
}

echo mysql_insert_id();

```



Mysql 扩展应用案例

添加新闻表单

1) 用户获取到一个可以添加新闻表单的 HTML 文件

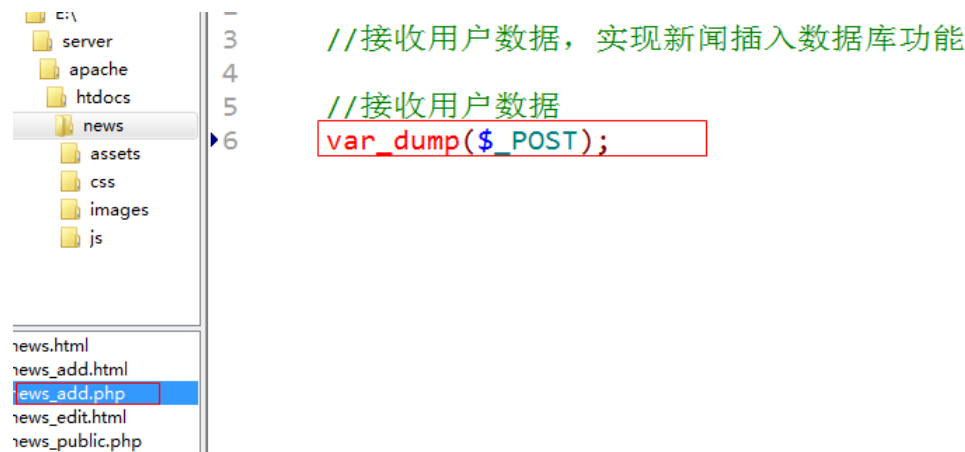
修改表单项: method 提交对象

```

<body>
<form action="news_add.php" method="post" class="definewidth m20">
<table class="table table-bordered table-hover definewidth m10">
<tr>
<td width="10%" class="tableleft">标题</td>
<td width="40%">
<input type="text" id="title" name="title"/>
</td>
<td width="10%" class="tableleft"></td>
<td width="40%">

```

2) 用户提交数据到后台指定的 PHP 脚本 (POST)



3) 后台脚本处理数据，插入到数据库：处理指 PHP 在拿到数据放入数据库之前要做很多验证工作保证数据是有效的。

3.1 接收数据

```
//var_dump($_POST);

$title = isset($_POST['title']) ? trim($_POST['title']) : ''; //title是字符串（重要）
$isTop = isset($_POST['isTop']) ? (integer)$_POST['isTop'] : 2; //数字需求，而且不重要
$content = isset($_POST['content']) ? trim($_POST['content']) : '';
$publiser = isset($_POST['publiser']) ? trim($_POST['publiser']) : '佚名'; //trim通常针对字符串
```

3.2 数据验证：合法性验证。标题不能为空，内容不能为空

```
//数据验证：标题和内容均不能为空
if(empty($title) || empty($content)){
    //标题和内容至少有一个为空
    echo '标题和内容都不能为空！';
}
```

3.3 提示用户，同时让用户重新再来（回到新增表单页面）

```
--
14 //数据验证：标题和内容均不能为空
15 if(empty($title) || empty($content)){
16     //提示同时回到提交页
17     header('Refresh:3;url=news_add.html'); //header前不能有输出，header: refresh不会阻止脚本执行
18
19     //标题和内容至少有一个为空//阻止脚本继续执行
20     exit('标题和内容都不能为空！');
21 }
```

3.4 数据入库

```
--
23 //数据入库
24 include_once 'news_public.php';
25
26 //阻止SQL指令执行
27 $time = time();
28 $sql = "insert into n_news
29     values(null,'{$title}',{$isTop},'{$content}','{$publiser}',{$time})";
30 $insert_id = my_error($sql);
31
```

4) 提示用户操作结果，并跳转到列表页（跳转到详情页）

```

31
32     //成功操作
33     //提示同时去到列表页
34     header('Refresh:3;url=news.php');
35     echo $title . ' 增加成功! ';

```

Mysql 扩展应用案例

显示新闻列表

将数据库中所有的新闻信息获取出来，然后讨要 HTML 提供的显示模板把数据全部按照指定的位置和格式输出显示。

- 1、 从数据库获取所有的新闻数据：PHP 脚本

```

3     //获取所有新闻内容并显示
4
5     //操作数据库获取数据
6     include_once 'news_public.php';
7
8     //组织SQL获取所有新闻信息
9     $sql = "select * from n_news";
10    //执行
11    $res = my_error($sql);           //拿到结果集

```

从结果集中取出所有的记录：一次取一条，一条一个数组；然后将所有的记录放到一个数组中：形成一个典型的二维数组

```

13    //循环遍历所有结果
14    $news = array();           //保存取出的记录（数组）
15
16    //方案1：获取结果集中记录数，然后for循环
17    /*
18    $num = mysql_num_fields($res);
19    for($i = 0;$i < $num;$i++){
20        $news[] = mysql_fetch_assoc($res);
21    }
22    */
23
24    //方案2：利用while循环，每次取到数据之后判断保存数据的结果，只要结果不为false，那么
25    while($row = mysql_fetch_assoc($res)){
26        $news[] = $row;
27    }

```

- 2、 有一个已经做好的 HTML 模板能够显示数据：前端

```

8     //print_r($news);
9
10    //包含显示模板（HTML）
11    include_once 'news.html';|

```

- 3、 PHP 想办法将数据在 HTML 中显示（二者结合）：PHP 包含 HTML

```
</thead>
<?php foreach($news as $k => $new):?>
<tr>
    <td><?php echo $k + 1;?></td>
    <td><?php echo $new['title'];?></td>
    <td><?php echo date('Y年m月d日 H时i分s秒',$new['pub_time']);?></td>
    <td><?php echo $new['publiser'];?></td>
    <td><?php echo $new['isTop'] == 1 ? '是' : '否';?></td>
    <td style="text-align:center;">
        <a href="news_edit.html">修改</a>&nbsp;  |
        <a href="javascript:" onclick="">删除</a>&nbsp;  
    </td>
</tr>
<?php endforeach;?>
</table>
```

注意细节：

- 1、 数据的显示要通过循环遍历输出：需要明确 PHP 是如何保存数据
- 2、 数据的格式需要按照需求来显示：列表、时间、是否置顶

Mysql 扩展应用案例

删除新闻

删除新闻就是通过用户选中某一个新闻点击删除实现从数据库剔除。

- 1、 让用户看到所有的新闻：已经完成（新闻显示列表：news.php→news.html）
- 2、 给用户提供一个对应的点击按钮：已经存在
- 3、 实现点击针对指定的新闻

```
<td><?php echo $new['isTop'] == 1 ? '是' : '否';?></td>
<td style="text-align:center;">
    <a href="news_edit.html">修改</a>&nbsp;  |
    <a href="news_delete.php?id=<?php echo $new['id'];?>" onclick="return
    confirm('是否确定删除当前新闻? ');">删除</a>&nbsp;  
</td>
</tr>
<?php endforeach;?>
```

- 4、 有一个专门的 PHP 脚本负责响应用户的点击：点哪条删哪条

4.1 接收要删除的新闻 ID

```
2
3 //新闻管理：删除指定新闻
4 header('Content-type:text/html;charset=utf-8');
5
6 //接收要删除的新闻ID
7 $id = isset($_GET['id']) ? (integer)$_GET['id'] : 0; //0不会存在
8 if($id == 0) {
9     header('Refresh:3;url=news.php');
10    echo '当前要删除的新闻不存在!';
11    exit;
12 }
```

4.2 调用数据库操作删除数据

```

13
14 //删除数据
15 include_once 'news_public.php';
16
17 //组织SQL并执行
18 my_error('delete from n_news where id = ' . $id);
19
20 |

```

4.3 提示删除成功，并且刷新当前界面

```

18 my_error('delete from n_news where id = ' . $id);
19
20
21 //提示
22 header('Refresh:3;url=news.php');
23 echo '当前新闻删除成功!';

```

Mysql 扩展应用案例

修改新闻

修改新闻指用户通过某个取到看到想修改的新闻，然后点击之后能够修改对应的新闻信息，最后点击保存实现修改更新到数据库。

- 1、 需要看到想编辑新闻：已经完成：新闻列表 news.php→news.html
- 2、 可以点击实现查看：列表页中为每个新闻增加点击链接

```

<td><?php echo $new['publiser']?></td>
<td><?php echo $new['isTop'] == 1 ? '是' : '否';?></td>
<td style="text-align:center;">
    <a href="news_edit.php?id=<?php echo $new['id'];?>">修改</a>&nbsp;|
    <a href="news_delete.php?id=<?php echo $new['id'];?>" onclick="return
    confirm('你确定要删除当前新闻吗?');">删除</a></td>

```

- 3、 需要一个 PHP 脚本能够处理这件事
- 3.1 创建 PHP 文件，同时接受要编辑的 ID

```

2
3 //查询某个新闻信息放到对应的表单中（编辑第一步）
4
5 header('Content-type:text/html;charset=utf-8');
6
7 //接收要编辑的新闻ID
8 $id = isset($_GET['id']) ? (integer)$_GET['id'] : 0; //0不会存在
9 if($id == 0) {
10     header('Refresh:3;url=news.php');
11     echo '当前要编辑的新闻不存在!';
12     exit;
13 }

```

- 3.2 通过 ID 获取对应的新闻信息

```
//获取当前ID对应的新闻信息
include_once 'news_public.php';
$sql = "select * from n_news where id = {$id}";
$res = my_error($sql);
$new = mysql_fetch_assoc($res); // $new一定是一个数组吗?
```

4、 需要一个 HTML 模板来展示当前要编辑的数据：PHP 包含 HTML

4.1 包含模板 HTML

```
$sql = "select * from n_news where id = {$id}";
$res = my_error($sql);
$new = mysql_fetch_assoc($res); // $new一定是一个数组吗?

//加载模板
include_once 'news_edit.html';
```

I

4.2 在模板表单中显示对应的数据

```
<td width="10%" class="tableleft">标题</td>
<td width="40%">
    <input type="text" id="title" name="title" value="<?php echo
    $new['title'];?>" />
</td>
<td width="10%" class="tableleft"></td>
<td width="40%">
    </td>
</tr>
<tr>
<td width="10%" class="tableleft">是否置顶</td>
<td width="10%">
    <input type="radio" name="isTop" <?php echo $new['isTop'] == 1 ?
    'checked="checked"' : '' ;?> value="1"/>是
    <input type="radio" name="isTop" <?php echo $new['isTop'] != 1 ?
    'checked="checked"' : '' ;?> value="2"/>否
</td>
<td width="10%" class="tableleft"></td>
<td>
    下面还有一个内容
</td>
```

I

5、 需要修改前端提供的更新表单：表单提交对象（不限）

5.1 需要修改表单提交对象

```
</head>
<body>
    <form action="news_update.php" method="post" class="definewidth m20">
        <table class="table table-bordered table-hover definewidth m10">
            <tr>
                <td width="10%" class="tableleft">标题</td>
```

5.2 更新操作，是后台根据表单提交的数据来选择某个新闻进行更新：表单中必须包含一个唯一能确定记录的新闻：ID，而 ID 不需要给用户看，需要增加隐藏表单

```
</head>
<body>
    <form action="news_update.php" method="post" class="definewidth m20">
        <input type="hidden" name="id" value="<?php echo $new['id'];?>" />
        <table class="table table-bordered table-hover definewidth m10">
            <tr>
```

6、 需要一个 PHP 脚本能够接受修改后的数据并保存到数据库

6.1 增加更新数据的 PHP 脚本

```
rectory Clip 1 2 3 4 5 6
新加卷
E:\
server
apache
htdocs
news
assets
css
1 <?php
2
3 //更新数据到数据库
4 header('Content-type:text/html;charset=utf=8');
5
6 //
```

6.2 接收用户提交的数据，同时进行数据验证：标题和内容均不能为空

```
>
6 //接收数据并验证
7 $id = isset($_POST['id']) ? intval($_POST['id']) : 0;
8 $title = isset($_POST['title']) ? trim($_POST['title']) : ''; //title是字符串（重要）
9 $isTop = isset($_POST['isTop']) ? intval($_POST['isTop']) : 2; //数字需求，而且不重要
10 $content = isset($_POST['content']) ? trim($_POST['content']) : '';
11
12 //数据验证：标题和内容均不能为空
13 if(empty($title) || empty($content)){
14     //提示同时回到提交页
15     header('Refresh:3;url=news.php'); //header前不能有输出，header: refresh不会阻止脚本执行
16
17     //标题和内容至少有一个为空//阻止脚本继续执行
18     exit('标题和内容都不能为空！');
19 }
```

6.3 组织 SQL 更新到数据库

```
>
1 //组织SQL更新到数据库
2 include_once 'news_public.php';
3 $sql = "update n_news set title = '{$title}',isTop = {$isTop},content='{$content}' where id = {$id}";
4 my_error($sql);
5
```

6.4 提示结果

```
24
25 //提示成功
26 header('Refresh:3;url=news.php');
27 echo '当前新闻更新成功！';
```