



链滴

PHP 无限分类的具体原理分析

作者: [maixiaojie](#)

原文链接: <https://ld246.com/article/1559037884906>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



对于初学PHP语言的朋友来说，PHP还有很多东西需要深入的探究。只有在不断的问题处理中才能掌握其中的真挚。什么是PHP无限分类呢？就像windows下新建一个文件夹，在新建的文件夹下又可以新建一个文件夹，这样无限循环下去，无限分类也是这样，父类可以分出它子类，子类又可以分出它的子，这样一直无限循环下去。

那PHP又是如何实现它的无限分类的呢？如何把它的各个分类——列出来呢？

首先我们假设有这样的一个三级分类，新闻→PHP新闻→PHP6.0出来了。

如果我们要查找“PHP6.0出来了”这条新闻，我们先点击新闻，然后再点击PHP新闻就可以查出来，也就是说我们可以通过祖父类一级一级地往下找，反过来我们只要知道一个子类的父类，就可以把查找出来了。这样我们在设计数据库时就可以多设计一个父类id的字段就可以实现PHP无限分类的功能了。

```
//我们建一个表"class"
CREATE TABLE `class` (
  `id` int(11) NOT NULL auto_increment COMMENT '分类id',
  `f_id` int(11) NOT NULL COMMENT '父id',
  `name` varchar(25) collate gbk_bin NOT NULL COMMENT '分类名称',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=gbk COLLATE=gbk_bin AUTO_INCREMENT=1;
```

```
//首先我们往数据库里插入‘新闻’这个大分类，因为‘新闻’是最大分类，上面没有父类了，所以把它的f_id设置为0。
INSERT INTO `class` (`id`, `f_id`, `name`) VALUES(1, 0, '新闻'); //id这个字段是自动增长的，可以不值。
```

```
//然后我们再往数据库里插入‘PHP新闻’这个分类，它的父类‘新闻’的id是1，所以它的f_id设置1。
INSERT INTO `class` (`id`, `f_id`, `name`) VALUES(2, 1, 'PHP新闻');
```

```
//然后我们再往数据库里插入‘PHP6.0出来了’这个分类，它的父类‘PHP新闻’的id是2，所以它的_id设置为2。
```

```
INSERT INTO `class` (`id`, `f_id`, `name`) VALUES(3, 2, 'PHP6.0出来了');
```

//同理，我们可以这样一直往下插入分类，也就达到了PHP无限分类。

//我们可以发现插入一个分类的原则关键是找到这个分类的父类的id，然后作为这个分类的f_id字段值。

//假设要插入跟‘新闻’同一个级别的分类‘技术’，也就是说它也是最大分类，上面没有父类了，么它的f_id也设置为0;

```
INSERT INTO `class` (`id`, `f_id`, `name`) VALUES(4, 0, '技术');
```

//在‘技术’下面又有一个分类‘PHP技术’，那么我们怎么插入呢，首先找到‘PHP技术’的父类‘技术’的id，然后作为自己的f_id字段的值。

```
INSERT INTO `class` (`id`, `f_id`, `name`) VALUES(5, 4, 'PHP技术');
```

我们已经知道如何往数据库里插入各个分类了，那又如何把各个分类罗列出来呢？

```
1. < ?php
2. header("Content-type:text/html;charset=utf-8");
3. $db=new mysqli("localhost","root","","news_php100");
   //实例化一个数据库连接。使用这个前一定要确保已经加载了mysqli类库，
   或者用mysql_connect这个方式连接。
4. if(mysqli_connect_errno()){
5. echo "链接失败:".mysqli_connect_error();
6. exit(); }
7. $db->query("set names utf8");
8. $result=$db->query("select name from class where f_id=0");
   //查找f_id=0的分类，也就是查找每一个大类。
9. while($row=$result->fetch_assoc()){
10. echo $row['name']." < br>"; //这样就把每个大类循环出来了。
11. }
12. //同样我们可以把新闻的子类循环出来。
13. $result=$db->query("select * from class where f_id=1");
   //查找f_id=1的分类，也就是查找‘新闻’的子类。
14. while($row=$result->fetch_assoc()){
15. echo $row['name']."
16. "; //这样就把‘新闻’的子类循环出来了。注意：只是子类，不包括孙子类。
17. }
18. //写到这里，我们会发现一个问题，如果这个分类是10级分类，难道我们要写
   10个循环把它每个子类循环出来？如果是更多级分类呢，这样写显然是不现实的。
19. //那又有什么办法解决呢？我们可以写一个递归的函数，把f_id作为参数传入，
```

不断循环每一个f_id的值，也就是说把每一个f_id值的子类循环出来。

```
20. //首先我们把各个分类的值保存在一个二维数组中，在下面的递归函数里有用。
```

```
21. $result=$db->query("select * from class");
```

```
22. while($row=$result->fetch_assoc()){
```

```
23. $arr[]=array($row[id],$row[f_id],$row[name]); //每一行保存一个
   分类的id,f_id,name的信息。
```

```
24. }
```

```
25. function fenlei($f_id=0){ // $f_id初始化为0，也就是从最大分类开始循环。
```

```
26. global $arr; //声明$arr为全局变量才可在函数里引用。
```

```
27. for($i=0;$i< count($arr);$i++){ //对每个分类进行循环。
```

```
28. if($arr[$i][1]==$f_id){ // $arr[$i][1]表示第$i+1个分类的f_id的值。
```

开始\$f_id=0,也就是把f_id=0的分类输出来。

```
29. echo $arr[$i][2]." < br>"; // $arr[$i][1]表示第$i+1个分类的name的值。
```

```
30. fenlei($arr[$i][0]); // $arr[$i][1]表示第$i+1个分类的id的值。进行递归
```

```
    , 也就是把自己的id作为f_id参数把自己的子类再循环出来。  
31. }  
32. }  
33. }  
34. fenlei(); //使用这个函数.  
35. ?>
```

以上代码示例为我们具体讲解了有关PHP无限分类的原理及使用方法，希望能对大家有所帮助。