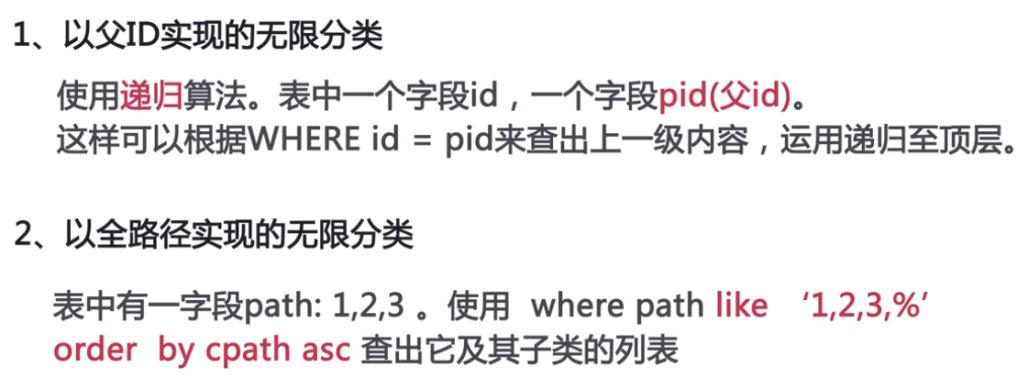
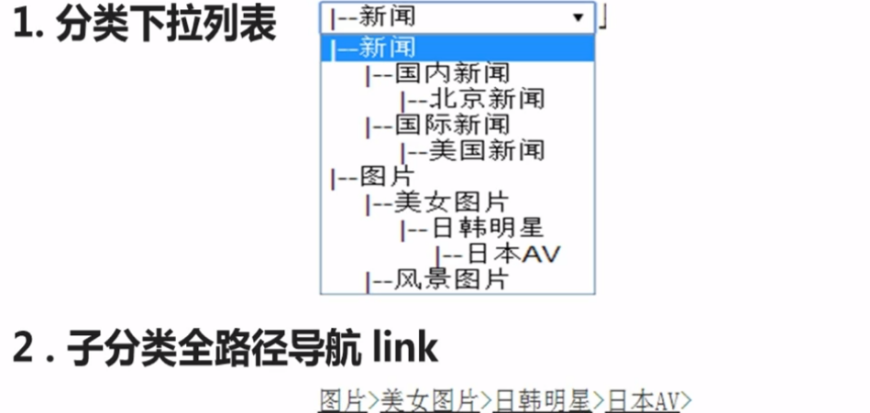
# 递归无极限分类

## 思路



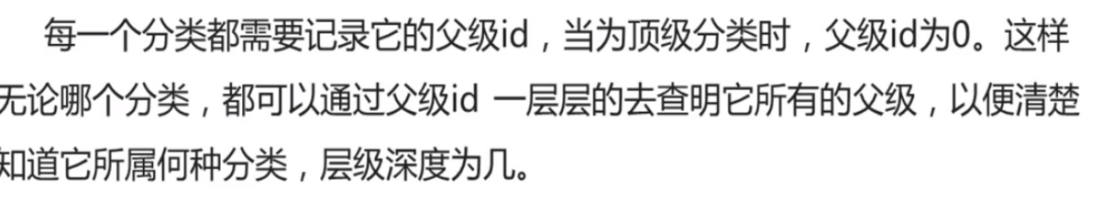
## 用处



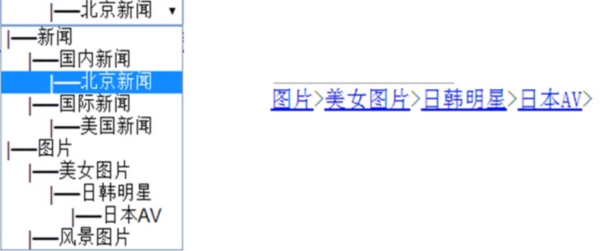
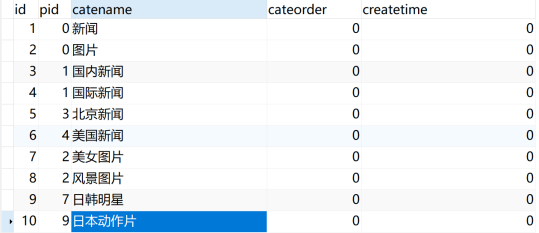
## 递归



## 原理



## SQL



/\*  
 Navicat Premium Data Transfer  
  
 Source Server : 127.0.0.1  
 Source Server Type : MySQL  
 Source Server Version : 50553  
 Source Host : localhost:3306  
 Source Schema : test1  
  
 Target Server Type : MySQL  
 Target Server Version : 50553  
 File Encoding : 65001  
  
 Date: 09/10/2019 23:29:59  
\*/  
  
SET NAMES utf8mb4;  
SET FOREIGN\_KEY\_CHECKS = 0;  
  
-- ----------------------------  
-- Table structure for sessions  
-- ----------------------------  
DROP TABLE IF EXISTS `sessions`;  
CREATE TABLE `sessions` (  
 `session\_id` int(11) NULL DEFAULT NULL,  
 `session\_data` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  
 `session\_expires` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL  
) ENGINE = MyISAM CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic;  
  
-- ----------------------------  
-- Records of sessions  
-- ----------------------------  
INSERT INTO `sessions` VALUES (0, 'use|s:3:\"sun\";age|i:233;email|s:11:\"sun@fox.com\";', '1570630054');  
  
-- ----------------------------  
-- Table structure for sort  
-- ----------------------------  
DROP TABLE IF EXISTS `sort`;  
CREATE TABLE `sort` (  
 `id` int(11) NOT NULL,  
 `pid` int(11) NULL DEFAULT NULL,  
 `catename` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci NULL DEFAULT NULL,  
 `cateorder` int(11) NULL DEFAULT NULL,  
 `createtime` int(11) NULL DEFAULT NULL,  
 PRIMARY KEY (`id`) USING BTREE  
) ENGINE = MyISAM CHARACTER SET = utf8 COLLATE = utf8\_general\_ci ROW\_FORMAT = Dynamic;  
  
-- ----------------------------  
-- Records of sort  
-- ----------------------------  
INSERT INTO `sort` VALUES (1, 0, '新闻', 0, 0);  
INSERT INTO `sort` VALUES (2, 0, '图片', 0, 0);  
INSERT INTO `sort` VALUES (3, 1, '国内新闻', 0, 0);  
INSERT INTO `sort` VALUES (4, 1, '国际新闻', 0, 0);  
INSERT INTO `sort` VALUES (5, 3, '北京新闻', 0, 0);  
INSERT INTO `sort` VALUES (6, 4, '美国新闻', 0, 0);  
INSERT INTO `sort` VALUES (7, 2, '美女图片', 0, 0);  
INSERT INTO `sort` VALUES (8, 2, '风景图片', 0, 0);  
INSERT INTO `sort` VALUES (9, 7, '日韩明星', 0, 0);  
INSERT INTO `sort` VALUES (10, 9, '日本动作片', 0, 0);  
INSERT INTO `sort` VALUES (0, NULL, NULL, NULL, NULL);  
  
SET FOREIGN\_KEY\_CHECKS = 1;

## 树形思路

根据已知的pid进行搜索递归找出所有相关的

外道的先循环-->内道的先循环--->外道的进行循环

## 递归树形

<?php  
*define*("host","localhost");  
*define*("user","root");  
*define*("pwd","root");  
*define*("db","test1");  
*define*("port",3306);  
  
$conn=new mysqli(*host*,*user*,*pwd*,*db*,*port*);

<?php  
  
include "conn.php";  
  
function getTree(*$pid*,&*$result*=[],*$spac*=0){  
 global $conn;  
 *$spac*=*$spac*+2;  
 $sql="select \* from sort where pid=*{$pid*}";  
 $res=*mysqli\_query*($conn,$sql);  
 while($row=*mysqli\_fetch\_assoc*($res)){  
 $id=$row['id'];  
 $row['catename']=*str\_repeat*('&nbsp;&nbsp;',*$spac*).$row['catename'];  
 *$result*[]=$row;  
 getTree($id,*$result*,*$spac*);  
 }  
 return *$result*;  
}  
  
function display(*$sid*=0){  
 $res=getTree(0);  
 echo "<select name='cate'>";  
 foreach($res as $k=>$v){  
 if(*$sid*==$v['id']){  
 echo "<option selected>{$v['catename']}</option>";  
 }else{  
 echo "<option >{$v['catename']}</option>";  
 }  
 }  
 echo "</select>";  
}  
  
echo "<br>";  
display(3);

## 线形思路

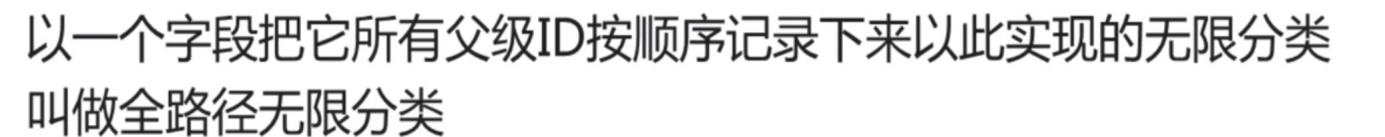
根据子id进行找出名字拼接好就可以了

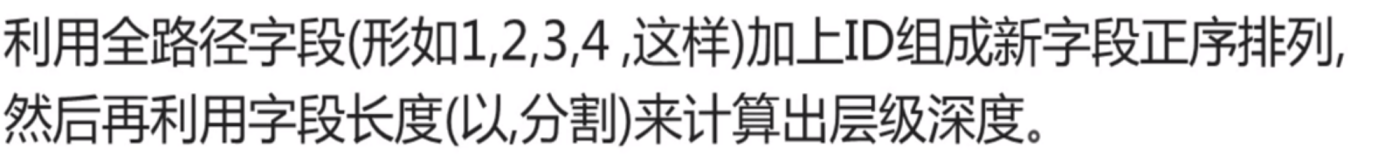
## 递归线形

<?php  
*define*("host","localhost");  
*define*("user","root");  
*define*("pwd","root");  
*define*("db","test1");  
*define*("port",3306);  
  
$conn=new mysqli(*host*,*user*,*pwd*,*db*,*port*);

<?php  
include "conn.php";  
  
function getList(*$cid*=0,&*$result*=[]){  
 global $conn;  
 $sql="select \* from sort where id=*{$cid*}";  
 $res=*mysqli\_query*($conn,$sql);  
 $row=*mysqli\_fetch\_assoc*($res);  
  
 if($row){  
 *$result*[]=$row['catename'];  
 getList($row['pid'],*$result*);  
 }  
 *krsort*(*$result*);  
 return *$result*;  
}  
  
function display(*$id*=10){  
 $res=getList(*$id*);  
 $str="";  
 foreach ($res as $k=>$v){  
 if($k==0){  
 $str.="<a href='conn.php'>"."{$v}".'</a>';  
 }else{  
 $str.="<a href='conn.php'>"."{$v}".'</a>'.'>';  
 }  
 }  
  
 return $str;  
}  
  
$str=display(10);  
echo $str;

# 全路径无限极分类





## 数据库





## 树形思路

path是相关联的

排序是按照fullpath进行先排好去的--->拼接path与id

这种方法对于数据库的id是有要求的，我估计应用汇比价局限！

缩进是按照path的个数进行控制空格的个数

## 树形

<?php  
include "conn.php";  
  
function getTreeTree(){  
 global $conn;  
 $result=[];  
 $sql="select id,catename,concat(ifnull(path,''),',',id) as fullpath from full order by fullpath asc";  
 $res=*mysqli\_query*($conn,$sql);  
 while($row=*mysqli\_fetch\_assoc*($res)){  
 $deep=*explode*(",",$row['fullpath']);  
 $row['catename']=*str\_repeat*("&nbsp;&nbsp;",*intval*(*count*($deep))\*2).$row['catename'];  
 $result[]=$row;  
 }  
 return $result;  
}  
  
  
function display(){  
 $res=getTreeTree();  
 echo "<select name='aa'>";  
 foreach($res as $k=>$v){  
 echo "<option>{$v['catename']}</option>";  
 }  
 echo "</select>";  
}  
  
display();

## 线形思路

按照path进行拼接即可

## 线形

<?php  
  
include "conn.php";  
  
function getLink(*$id*){  
 $str='';  
 global $conn;  
 $sql="select \* from full where id=*{$id*}";  
 $res=*mysqli\_query*($conn,$sql);  
 $row=*mysqli\_fetch\_assoc*($res);  
 $selfname=$row['catename'];  
 $arr=*explode*(',',$row['path']);  
 foreach($arr as $k=>$v){  
 $sql="select catename from full where id=*{*$v}";  
 $sql=*mysqli\_query*($conn,$sql);  
 $catename=*implode*('',*mysqli\_fetch\_assoc*($sql));  
 if($k==*count*($arr)-1){  
 $str=$str."<a href='conn.php'>"."$catename"."</a>"."><a href='conn.php'>$selfname</a>";  
 }else{  
 $str.="<a href='conn.php'>"."$catename"."</a>".">";  
 }  
 }  
 return $str;  
}  
  
$res=getLink(10);  
echo "$res";