

# PHP 员工管理系统

## 一、项目展示

### 1、展示员工列表

将员工信息直观展示出来，可以进行搜索、排序、翻页等常见操作。

员工信息列表

快速查询：

| ID | 姓名 | 所属部门 | 出生日期       | 入职时间       | 相关操作   |
|----|----|------|------------|------------|--------|
| 1  | 张三 | 人事部  | 1990-02-03 | 2015-08-08 | 编辑  删除 |
| 2  | 李四 | 开发部  | 1991-05-02 | 2015-07-02 | 编辑  删除 |
| 3  | 王五 | 媒体部  | 1989-07-12 | 2015-09-01 | 编辑  删除 |
| 4  | 赵六 | 销售部  | 1989-12-03 | 2014-04-12 | 编辑  删除 |
| 5  | 小明 | 开发部  | 1990-02-03 | 2015-08-08 | 编辑  删除 |

首页 上一页 下一页 尾页

添加员工 部门管理

### 2、员工搜索

可以通过员工姓名的关键字进行搜索，例如输入“张”，搜索出所有姓名中包含“张”的数据：

员工信息列表

快速查询：

| ID | 姓名 | 所属部门 | 出生日期       | 入职时间       | 相关操作   |
|----|----|------|------------|------------|--------|
| 1  | 张三 | 人事部  | 1990-02-03 | 2015-08-08 | 编辑  删除 |
| 9  | 小张 | 媒体部  | 1990-02-03 | 2015-08-08 | 编辑  删除 |

首页 上一页 下一页 尾页

添加员工 部门管理

### 3、员工列表排序

可以按照“ID”、“姓名”、“所属部门”、“出生日期”、“入职时间”这些字段进行排序，有正序和倒序两种排序方式。

| ID | 姓名 | 所属部门 | 出生日期       | 入职时间       | 相关操作   |
|----|----|------|------------|------------|--------|
| 8  | 小王 | 销售部  | 1989-12-03 | 2014-03-12 | 编辑  删除 |
| 12 | 小明 | 销售部  | 1989-12-03 | 2014-04-12 | 编辑  删除 |
| 4  | 赵六 | 销售部  | 1989-12-03 | 2014-04-15 | 编辑  删除 |
| 2  | 李四 | 开发部  | 1991-05-02 | 2015-07-02 | 编辑  删除 |
| 10 | 关羽 | 开发部  | 1991-05-02 | 2015-07-03 | 编辑  删除 |

#### 4、员工列表分页展示

可以进行“上一页”、“下一页”、“首页”、“尾页”的翻页操作：

| ID | 姓名   | 所属部门 | 出生日期       | 入职时间       | 相关操作   |
|----|------|------|------------|------------|--------|
| 11 | 王五7  | 媒体部  | 1989-07-12 | 2015-09-01 | 编辑  删除 |
| 12 | 小明   | 销售部  | 1989-12-03 | 2014-04-12 | 编辑  删除 |
| 13 | 问问   | 人事部  | 2015-10-13 | 2015-10-12 | 编辑  删除 |
| 14 | erer | 销售部  | 2015-10-12 | 2015-10-12 | 编辑  删除 |

首页 上一页 下一页 尾页

#### 5、部门管理

可以进行部门的添加、修改和删除操作：

### 部门管理

| ID | 部门名称 | 操作     |
|----|------|--------|
| 2  | 开发部  | 修改  删除 |
| 3  | 媒体部  | 修改  删除 |
| 4  | 销售部  | 修改  删除 |
| 5  | 人事部  | 修改  删除 |

添加部门:

修改部门名称:

[返回员工列表](#)

#### 6、添加员工

员工信息添加功能，通过表单让用户输入员工信息，点击“保存资料”后提交表单：

添加员工

姓名:

所属部门:

未选择

出生年月:

入职日期:

保存资料

返回列表

7、编辑员工

员工信息编辑功能，通过表单让用户修改员工信息：

编辑员工

姓名:

张三

所属部门:

人事部

出生年月:

1990-02-03

入职日期:

2015-08-08

保存资料

返回列表

8、删除员工

当在“员工列表”中点击“删除”时，删除指定的员工数据：

| ID | 姓名 | 所属部门 | 出生日期       | 入职时间       | 相关操作   |  |
|----|----|------|------------|------------|--|--|
| 1  | 张三 | 人事部  | 1990-02-03 | 2015-08-08 |  编辑 |  删除 |
| 2  | 李四 | 开发部  | 1991-05-02 | 2015-07-02 |  编辑 |  删除 |
| 3  | 王五 | 媒体部  | 1989-07-12 | 2015-09-01 |  编辑 |  删除 |
| 4  | 赵六 | 销售部  | 1989-12-03 | 2014-04-15 |  编辑 |  删除 |
| 5  | 小明 | 开发部  | 1990-02-03 | 2015-08-08 |  编辑 |  删除 |

## 二、项目概述

任何一种编程语言都需要对数据进行处理，PHP 语言也不例外。PHP 所支持的数据库类型较多，在这些数据库中，由于 MySQL 的跨平台性、可靠性、访问效率较高以及免费开源等特点，备受 PHP 开发者的青睐。接下来将运用 PHP 操作 MySQL 的一些知识，来开发一个企业员工管理系统，要求项目具备以下功能：

### 1、保存员工信息

保存员工的基本信息，包括员工姓名、所属部门、出生日期、入职时间。可以对员工信息进行添加、修改、删除

### 2、展示员工列表

将员工信息以列表的形式展示出来，以便于查看。员工列表要实现排序、分页查询、按姓名搜索的功能。

### 3、部门管理

可以部门信息进行管理，实现部门的添加、修改、删除。当添加或编辑员工时，所属部门以下拉菜单的形式选择。

## 三、数据库设计

### 1、员工信息表（表名：pre\_emp）

| 字段          | 字段类型         | 说明            |
|-------------|--------------|---------------|
| emp_id      | int unsigned | 员工表主键 ID，自动增长 |
| emp_dept_id | int unsigned | 员工所属部门 ID     |
| emp_name    | varchar(12)  | 员工姓名          |
| emp_birth   | Timestamp    | 员工出生日期        |
| emp_entry   | Timestamp    | 员工入职时间        |

### 2、部门表（表名：pre\_dept）

| 字段        | 字段类型         | 说明            |
|-----------|--------------|---------------|
| dept_id   | int unsigned | 部门表主键 ID，自动增长 |
| dept_name | varchar(12)  | 部门名称          |

上述数据表创建的 SQL 语句如下：

```
-- 创建数据库
create database `itcast`;

-- 员工表
create table `pre_emp` (
  `emp_id` int unsigned primary key auto_increment,
  `emp_dept_id` int unsigned not null comment '所属部门 ID',
  `emp_name` varchar(12) not null comment '姓名',
  `emp_birth` timestamp default '0:0:0' not null comment '出生日期',
```

```

`emp_entry` timestamp default '0:0:0' not null comment '入职时间'
)charset=utf8;
-- 部门表
create table `pre_dept`(
    `dept_id` int unsigned primary key auto_increment,
    `dept_name` varchar(12) not null unique comment '部门名称'
)charset=utf8;

```

在完成数据表的创建后，向数据库中插入几条测试数据：

```

-- 员工测试数据
insert into `pre_emp` values
(null,'1','张三','1990-02-03','2015-08-08'),
(null,'2','李四','1991-05-02','2015-07-02'),
(null,'3','王五','1989-07-12','2015-09-01'),
(null,'4','赵六','1989-12-03','2014-04-12');
-- 部门测试数据
insert into `pre_dept` values
(null,'市场部'),
(null,'开发部'),
(null,'媒体部'),
(null,'销售部'),
(null,'人事部');

```

## 四、项目文件说明

本项目的目录结构和各文件的说明如下表所示：

| 文件           | 说明                    |
|--------------|-----------------------|
| css          | 项目的 CSS 样式文件保存目录      |
| images       | 项目的图片文件保存目录           |
| js           | 项目的 JavaScript 文件保存目录 |
| lib          | 项目的函数库文件保存目录          |
| view         | 项目的 HTML 视图模板保存目录     |
| data.sql     | 项目的数据库 SQL 语句         |
| dept.php     | 部门管理功能 PHP 脚本         |
| emp_add.php  | 员工添加功能 PHP 脚本         |
| emp_edit.php | 员工编辑功能 PHP 脚本         |
| emp_del.php  | 员工删除功能 PHP 脚本         |
| index.php    | 员工列表功能 PHP 脚本         |
| init.php     | 项目初始化 PHP 脚本          |

部署项目时，先按照 data.sql 文件中的 SQL 语句创建数据库、数据表和测试数据，然后将项目代码复制到站点目录下，访问 index.php 即可。

## 五、知识梳理

### 1、数据库扩展

PHP 访问数据库提供了三种扩展，即 MySQL 扩展、MySQLi 扩展、PDO 扩展。本项目基于 MySQL 扩展实现。在使用 MySQL 扩展时，需要检查该扩展是否已经开启，未开启时，通过在 `php.ini` 中找到如下配置项取消注释即可。

```
extension=php_mysql.dll
```

### 2、连接数据库

MySQL 扩展连接数据库的函数是 `mysql_connect()`。其声明方式如下：

```
resource mysql_connect ( [ string $server [, string $username [, string $password [, bool $new_link [, int $client_flags ]]] ] )
```

其中第 1 个参数 `$server` 表示要连接的数据库地址，第 2 个参数表示操作数据库的用户名，第 3 个参数表示用户对应的密码。该函数执行后会返回一个资源类型的连接标识。

### 3、选择数据库

连接数据库后就可以选择要操作的数据库。MySQL 扩展提供了 `mysql_select_db()` 函数用于选择数据库。其声明方式如下：

```
bool mysql_select_db ( string $database_name [, resource $link_identifier ] )
```

在上诉声明中，参数 `$database_name` 表示要选择的数据库名称，可选参数 `$link_identifier` 表示 MySQL 连接，省略该参数时默认使用最近打开的连接。

### 4、执行 SQL 语句

PHP 操作数据库时，需要向数据库发送 SQL 语句。MySQL 提供了 `mysql_query()` 函数用于执行 SQL 语句，其声明方式如下：

```
resource mysql_query ( string $query [, resource $link_identifier = NULL ] )
```

在上述声明中，`$query` 表示 SQL 语句，`$link_identifier` 是可选参数，表示 MySQL 连接，当省略时使用最新打开的连接。

### 5、处理结果集

当 `mysql_query()` 函数返回的是资源类型的结果集时，需要进一步的处理，才能得到相关数据。MySQL 扩展提供了多个函数用于处理结果集，下面对其常用的函数进行介绍。

#### (1) `mysql_fetch_row()`

该函数的作用是，从结果集中读取出一行数据，以索引数组的形式返回。其声明方式如下：

```
array mysql_fetch_row ( resource $result )
```

当函数执行成功后，会自动读取下一条数据，直到结果集中没有下一条数据时为止。

#### (2) `mysql_fetch_assoc()`

该函数将从结果集中读取一行数据，以关联数组的形式返回。其声明方式如下：

```
array mysql_fetch_assoc ( resource $result )
```

当函数执行成功后，会自动读取下一条数据，直到结果集中没有下一条数据时为止。

#### (3) `mysql_fetch_array()` 函数

该函数可以看做 `mysql_fetch_row()` 函数与 `mysql_fetch_assoc()` 函数的集合体，它会将结果集中的数据分别以索引数组和关联数组的形式返回。其声明方式如下：

```
array mysql_fetch_array ( resource $result [, int $result_type ] )
```

由于该函数可以同时返回索引数组和关联数组，因此该函数提供了一个可选参数 `$result_type`，其值可以是 `MYSQL_BOTH`（默认），`MYSQL_ASSOC` 或 `MYSQL_NUM` 中的一种，其中 `MYSQL_ASSOC` 只得到关联索引，`MYSQL_NUM` 只得到数字索引。

## 六、项目实施

### 1、封装数据库函数库

数据库函数库是将一些 PHP 操作数据库的代码封装成函数，以方便调用，减少重复代码的书写。代码参见项目文件“lib\db\_function.php”。

### 2、创建项目初始化文件

项目初始化文件是项目中每个实现功能的 PHP 脚本都引入的公共文件，该初始化文件的主要作用是引入函数库、设置字符集、连接数据库等。参见项目文件“init.php”。

### 3、员工列表展示功能

展示员工列表的功能开发十分简单，将员工数据从数据库中查询后展示即可。编写脚本文件“index.php”，其关键代码如下：

```
1 <?php
2 //员工列表功能
3 require './init.php'; //项目初始化文件
4 //准备 SQL 语句
5 $sql = "select * from pre_emp left join pre_dept on emp_dept_id=dept_id";
6 //获取员工信息数据
7 $emp_info = db_fetch_all($sql);
8 //加载视图页面，显示数据
9 require './view/list.html';
```

接下来编写 HTML 模板文件“view\list.html”，用于输出数据并展示。关键代码如下：

```
<table>
    <tr>
        <th>ID</th>
        <th>姓名</th>
        <th>所属部门</th>
        <th>出生日期</th>
        <th>入职时间</th>
        <th>相关操作</th>
    </tr>
    <?php if(empty($emp_info)): ?>
        <tr><td colspan="6">查询的结果不存在! </td></tr>
    <?php else: foreach($emp_info as $v): ?>
    <tr>
        <td><?php echo $v['emp_id']; ?></td>
        <td><?php echo $v['emp_name']; ?></td>
        <td><?php echo $v['dept_name']; ?></td>
```

```

        <td><?php echo date('Y-m-d',strtotime($v['emp_birth'])); ?></td>
        <td><?php echo date('Y-m-d',strtotime($v['emp_entry'])); ?></td>
        <td>
            <a href="emp_edit.php?id=<?php echo $v['emp_id']; ?>">编辑</a>
            <a href="emp_del.php?id=<?php echo $v['emp_id']; ?>">删除</a>
        </td>
    </tr>
<?php endforeach; endif; ?>
</table>

```

#### 4、员工列表排序功能

在学习 MySQL 时，我们通过 SQL 语句中的 order by 可以实现排序。开发员工列表排序功能也是通过 order by 实现的。其关键代码如下：

```

1  //函数：生成排序 SQL
2  function make_sort_sql($order,$sort){
3      //允许排序的字段
4      $fields = array('id','dept_id','name','birth','entry');
5      //判断$order 是否存在于合法字段列表中
6      if(in_array($order,$fields)){
7          //判断$sort 是否为合法值
8          if(in_array($sort,array('asc','desc'))){
9              return "order by `emp_$order` $sort";
10         }
11     }
12     return '';
13 }
14 //排序功能
15 $order = input_get('order');
16 $order_sort = input_get('sort');
17 $order_sql = make_sort_sql($order,$order_sort); //生成排序 SQL
18 //准备 SQL 语句
19 $sql = "select * from pre_emp left join pre_dept on emp_dept_id=dept_id $order_sql";

```

在 HTML 模板中需要为字段添加排序链接，以实现点击链接通过 GET 参数实现排序，代码如下：

```

<tr>
    <?php
    //反转排序值
    $order_sort = ($order_sort=='desc'? 'asc': 'desc');
    ?>
    <th><a href="?order=id&sort=<?php echo $order_sort; ?>">ID</a></th>
    <th><a href="?order=name&sort=<?php echo $order_sort; ?>">姓名</a></th>
    <th><a href="?order=dept_id&sort=<?php echo $order_sort; ?>">所属部门</a></th>
    <th><a href="?order=birth&sort=<?php echo $order_sort; ?>">出生日期</a></th>
    <th><a href="?order=entry&sort=<?php echo $order_sort; ?>">入职时间</a></th>
    <th>相关操作</th>

```



&lt;/tr&gt;

## 5、员工列表搜索功能

在员工列表中可以按照员工姓名进行搜索，我们可以通过 SQL 语句中的 like 实现该功能。其关键代码如下：

```
1 //函数：生成搜索 SQL
2 function make_search_sql($keyword){
3     $keyword = db_escape(filter($keyword));
4     return "and emp_name like '%$keyword%'";
5 }
6 //搜索功能
7 $search = input_get('search');
8 $where_sql = $search ? make_search_sql($search) : '';
9 //准备 SQL 语句
10 $sql = "select * from pre_emp left join pre_dept on emp_dept_id=dept_id where 1=1 $where_sql
$order_sql";
```

在 HTML 模板中需要创建同于输入搜索内容的表单，其关键代码如下：

```
<div class="search">
    <form>
        快速查询: <input type="text" name="search"/>
        <input type="submit" value="提交"/>
    </form>
</div>
```

## 6、员工列表分页功能

当员工数据越来越多时，我们需要为列表进行分页，以便于查看。实现分页查询的关键在于 SQL 语句中的 limit 的使用。首先创建分页链接类，方便在 HTML 中显示翻页链接，如“上一页”、“下一页”。请参见项目文件“lib\page\_function.php”。

接下来在员工列表功能中添加分页显示功能，其关键代码如下：

```
1 require './lib/page_function.php'; //引入分页函数库
2 //函数：生成 LIMIT SQL
3 function make_limit_sql($page,$page_size){
4     return 'limit '.(($page-1)*$page_size).','.$page_size;
5 }
6 //排序功能
7 $order = input_get('order');
8 $order_sort = input_get('sort');
9 $order_sql = make_sort_sql($order,$order_sort); //生成排序 SQL
10 //分页功能
11 $page_size = 5; //每页显示 5 条信息
12 $page_count = db_fetch_column('select count(*) from pre_emp'); //查询所有记录条数
13 $page_max = ceil($page_count/$page_size); //计算最大页码值
14 $page = (int)input_get('page'); //获取当前访问的页码
15 $page = max($page,1); //页码值最小为 1
```

```

16 $page = min($page, $page_max); //页码值最大为$max_page
17 $page_html = makePageHtml($page, $page_max); //调用函数生成分页链接
18 $limit_sql = make_limit_sql($page, $page_size); //拼接 SQL
19 //准备 SQL 语句
20 $sql = "select * from pre_emp left join pre_dept on emp_dept_id=dept_id $limit_sql";

```

上述代码通过调用分页链接生成函数 `makePageHtml()` 生成分页的 HTML 结果，我们需要在 HTML 模板中输出结果，其关键代码如下：

```

<div class="page">
    <?php echo $page_html ?>
</div>

```

## 7、部门管理功能

部门管理功能主要包括部门添加、部门修改、部门删除，其功能简单，可以在一个 PHP 脚本中实现，具体请参见“`dept.php`”。

在实现时，可以通过 GET 传参的方式，来区分不同的操作，如以下代码所示：

```

<?php
//部门管理功能
require './init.php'; //项目初始化文件
$action = input_get('action'); //获取操作参数
//部门名称修改
if($action=='rename'){
    .....

//添加部门
}elseif($action=='add'){
    .....

//删除部门
}elseif($action=='del'){
    .....
}

//查询部门列表
$sql = "select * from pre_dept";
$dept_info = db_fetch_all($sql);
//加载视图页面，显示数据
require './view/dept.html';

```

## 8、员工添加功能

实现员工添加功能主要通过收集表单数据后执行 `insert` 语句来完成。具体请参见“`emp_add.php`”。该脚本通过判断是否有 POST 数据来进行不同的处理，当没有 POST 数据时，显示 HTML 表单，当有 POST 数据时，说明用户提交了表单，则进行表单处理。其关键代码如下：

```

1  <?php
2  //员工添加功能
3  require './init.php'; //项目初始化文件
4  //表单处理
5  if($_POST){
6      //执行员工添加操作

```

```

7      .....
8  }
9
10 //获取部门信息
11 $sql = "select * from pre_dept";
12 $dept_info = db_fetch_all($sql);
13 //加载视图页面，显示数据
14 require './view/emp_add.html';

```

在 HTML 模板中，显示一个员工添加的表单，其中部门的选择使用下拉菜单来展现。其关键代码如下：

```

<table>
    <tr><th>姓名: </th><td><input type="text" name="emp_name" /></td></tr>
    <tr><th>所属部门: </th><td><select name="emp_dept_id">
        <option value="0">未选择</option>
        <?php foreach($dept_info as $v): ?>
            <option value="<?php echo $v['dept_id']; ?>" >
                <?php echo $v['dept_name'] ?></option>
        <?php endforeach; ?>
    </select></td></tr>
    <tr><th>出生年月: </th>
    <td><input id="birth" name="emp_birth" type="text"></td></tr>
    <tr><th>入职日期: </th>
    <td><input id="entry" name="emp_entry" type="text"></td></tr>
    <tr><td colspan="2" class="td-btn">
        <input type="submit" value="保存资料" />
        <input type="button" value="返回列表" onclick="location.href='index.php'" />
    </td></tr>
</table>

```

## 9、员工修改功能

员工修改功能的 HTML 页面和员工添加类似，其不同之处在于，员工添加页面是一个空的 HTML 表单，而员工修改页面是将待修改的员工的信息先展示到表单中，可以用户让用户来修改，修改后提交则通过“update”SQL 语句进行数据的更新操作。

实现员工编辑功能的具体代码请参见“emp\_edit.php”。其关键代码如下：

```

1  <?php
2  //员工编辑功能
3  require './init.php'; //项目初始化文件
4  $id = (int)input_get('id');
5  //表单处理
6  if($_POST){
7      //更新员工数据
8  }
9
10 //获取员工原来的信息
11 $sql = "select * from pre_emp left join pre_dept on emp_dept_id=dept_id where emp_id=$id";

```

```

12 $emp_info = db_fetch_row($sql);
13 //获取部门信息
14 $sql = "select * from pre_dept";
15 $dept_info = db_fetch_all($sql);
16 //加载视图页面，显示数据
17 require './view/emp_edit.html';

```

在 HTML 模板中，需要将员工原本的数据显示到表单中，员工所属的部门的下拉菜单也应该默认选中。其关键代码如下：

```

<form method="post">
<table>
    <tr><th>姓名: </th><td>
        <input type="text" name="emp_name"
        value="<?php echo $emp_info['emp_name']; ?>" /></td></tr>
    <tr><th>所属部门: </th><td><select name="emp_dept_id">
        <option value="0">未选择</option>
        <?php foreach($dept_info as $v): ?>
            <option value="<?php echo $v['dept_id']; ?>"
            <?php if($v['dept_id']==$emp_info['emp_dept_id']) echo "selected"; ?> >
            <?php echo $v['dept_name'] ?></option>
        <?php endforeach; ?>
    </select></td></tr>
    <tr><th>出生年月: </th><td><input id="birth" name="emp_birth" type="text"
    value="<?php echo date('Y-m-d',strtotime($emp_info['emp_birth'])); ?>"></td></tr>
    <tr><th>入职日期: </th><td><input id="entry" name="emp_entry" type="text"
    value="<?php echo date('Y-m-d',strtotime($emp_info['emp_entry'])); ?>"></td></tr>
    <tr><td colspan="2" class="td-btn">
        <input type="submit" value="保存资料" />
        <input type="button" value="返回列表" onclick="location.href='index.php'" />
    </td></tr>
</table>
</form>

```

## 10、员工删除功能

在实现员工删除时，需要先获取要删除的员工的 ID，然后通过“delete”SQL 语句实现数据删除即可。实现该功能的代码如下：

```

1 <?php
2 //员工删除功能
3 require './init.php'; //项目初始化文件
4 $id = (int)input_get('id');
5 $sql = "delete from pre_emp where emp_id=$id";
6 db_query($sql);
7 //删除后返回员工列表
8 header('Location:index.php');

```

### 11、测试

项目功能开发完成后，进入测试阶段，对项目中的主要功能进行测试。主要包括以下方面：

- ✓ 员工列表是否正常显示
- ✓ 按照“ID、姓名、所属部门、出生日期、入职时间”是否可以正序、降序排列
- ✓ 员工姓名搜索功能是否正常，如搜索“张”可以搜索出“张三”、“小张”。
- ✓ 员工列表分页是否准确无误，当给定 page 参数为“-1”、“abc”等非法值时是否处理。
- ✓ 部门管理功能是否正常实现“部门添加”、“部门修改”、“部门删除”。
- ✓ 员工添加功能是否可以正确添加，有无漏项未保存。
- ✓ 员工修改功能是否可以正确修改，所属部门是否可正常修改，有无漏项未保存。
- ✓ 员工删除功能是否可以正确删除数据

程序在浏览器中的运行效果如下：

