雇员管理系统开发文档

1. 需求分析

该项目实现功能：

1. 管理员可以登录到管理界面
2. 可以对雇员进行增加
3. 可以对雇员进行删除
4. 可以对雇员进行修改
5. 可以对雇员进行检查（分页显示）

可以统计查阅次数

1. 画出uml时序图
2. 界面设计
3. 先设计数据库

如果是大型的项目的时候，有很多的表格，那么我们就要用表名\_字段名的形式，如admin\_id

1. 管理员表 admin

create table admin(

id int primary key,

name varchar(32) not null,

password varchar(64) not null

);

create table employee(

id int primary key auto\_increment,

name varchar(64) not null,

level tinyint /\*1表示1级工\*/,

email varchar(64) not null,

salary float

);

添加一些初始化数据

insert into employee (name, level,email,salary) value(‘zs’,1,’zs@qq.com’,2200);

。。。

向管理员里面也添加一个数据

insert into admin(id,name,password) value(5,’admin’,md5(‘123’));

代码阶段：

创立项目 empManage

用户管理系统的程序框架图

登录界面 首先要有个 login.php界面 然后有个处理界面 loginProcess.php

合法的话就再进入 empMain界面

1、先不进行数据库的验证

要求，如果登录不成功，在login页面提示红色信息

在管理页面提供一个超链接，可以退出界面

1. 进行数据库验证 记得取出res->row 的时候，切记，使用assoc还是row，

取出字段的时候比如name，password等的，记得使用assoc，取出方式按照row[0]，1,2等方式，用row

1. 在管理页面显示登录成功的管理员的名字
2. 在用户列表页面empList.php显示所有的用户信息 以表格形式列出
3. 4的步骤显示出所有的信息，紧接着做!!!!分页!!!!!

在分页中有几个变量是必须的

$pageNow 现在显示的是第几页 用户输入

$pageCount 共有几页 在程序中计算出来，根据以下两条计算 向上取整 ceil函数，若整除就取整，若不能够整除，那么向上加1；

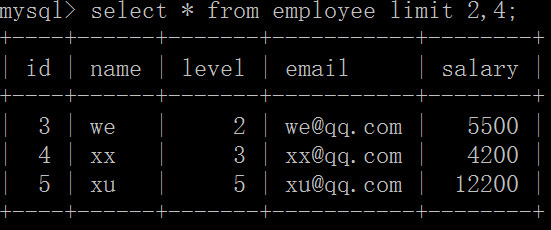
$rowCount 共有多少条记录 数据库中获取 通过count（\*）获取，但是有时候有的属性会为空，一般都通过count(id)来获取总共多少条记录

$pageSize 每一页显示多少条 程序员自己设定

分页设计:



Limit是从0开始计算的，后面的是表示取出几条



EmpManage1.0版本完成了基本的要求，登录，以及分页查看显示等功能

测试一下分页的效率，在实际开发中，数据量很大，因此我们测试当数据量大时，是否可以满足用户需求，模拟20w

数据。

Mysql自我复制

insert into employee(name,level,email,salary) select name,level,email,salary from employee;

事实证明，十几万数据的时候，超链接过多，因此浏览器崩溃。

现在设置一个页面只显示10条数据，以及上一页下一页的显示方法

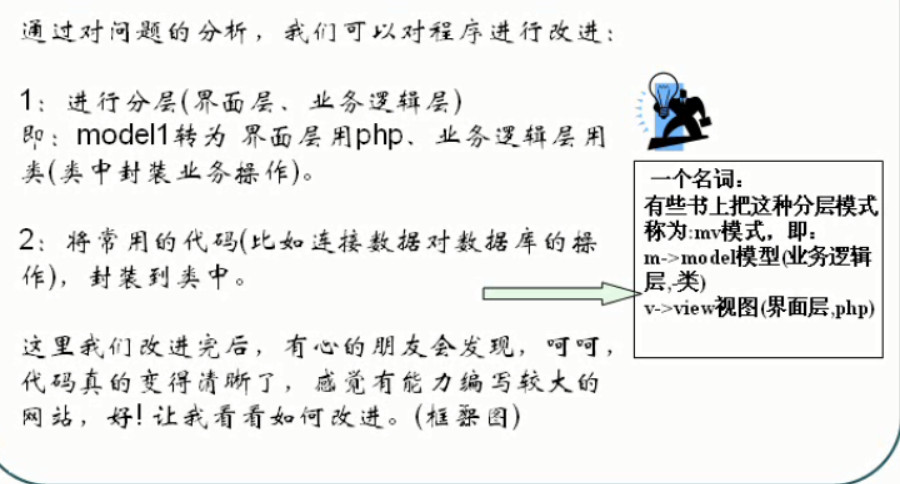
存在问题，界面和业务逻辑放在一起了，称作model1模式

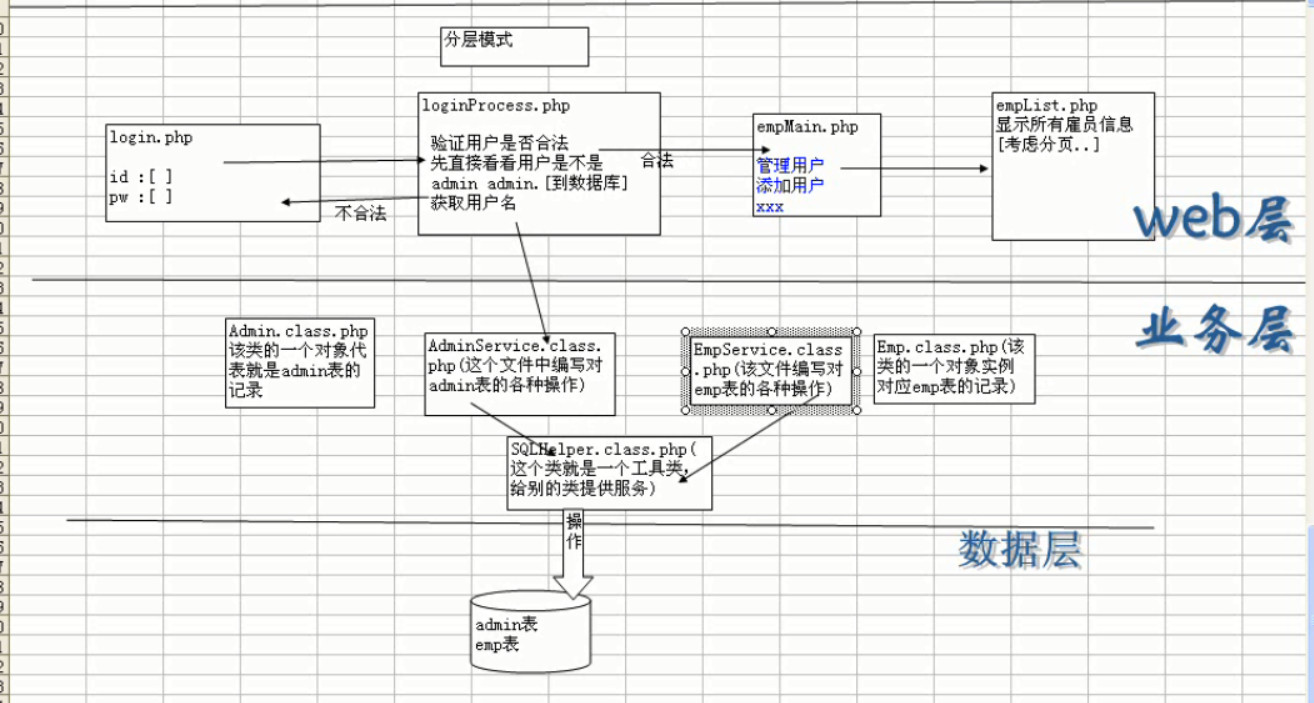
网站结构问题：login.php以及empList.php中都有对数据库的操作，代码有重复

Model2模式 分层模式 界面和业务逻辑分离，结构清晰，利于分工开发

Model1->分层模式->mvc模式

Mvc模式：强制把数据的输入，数据的处理，数据的显示分开





每一个实体表（订单，商品等表）都对应两个php文件 表名.class.php（表的相关记录） 以及 表名service.class.php(对表的操作，调用工具类)

工具类 SQLiTool.class.php对数据库进行操作

1. 在什么类里面进行分页功能 应该在EmployeeService.class.php中写
2. 通过分析，我们知道为了完成分页，我们的empList.php需要两个重要的数据 ，一个是pageCount还有一个是分页需要显示的数据

当调用函数的时候，函数返回结果集很重要，不返回肯定是没问题的，但是占用资源，因此必须要返回，但是过早的返回，调用者调用的时候就会调用一个空集，因此很重要！！！

解决办法：用数组将结果集里面的东西放到数组里面

分页功能见项目：做一个分页工具类紧接着可以调用，最后只需要调用参数就可以返回各个对象，然后从对象中取出所需要的相关信息即可

思路：设计一个pageDiv.class.php

Emp-MANAGE 2.0版本完成了分层模式的分页功能

增加功能以删除某个id雇员 将删除id的请求递交给自身empList

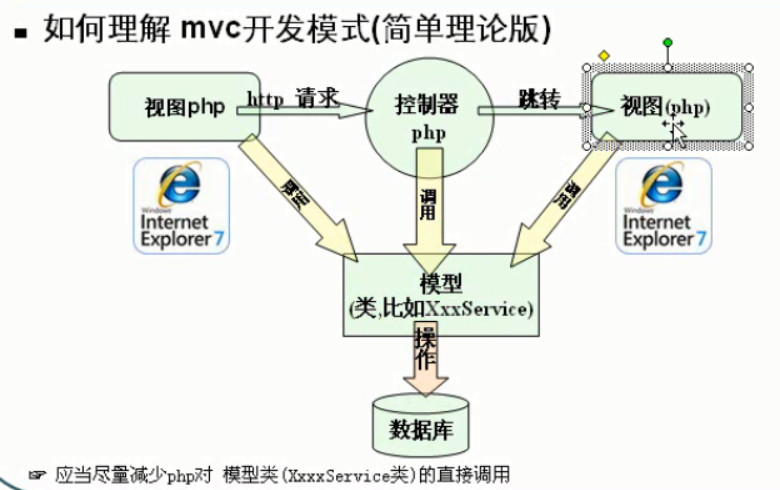
需要做一个标志位，区分对待是要删除还是分页显示

如果真的将删除请求直接提交给empList，则好的结构又被破坏，又会变得臃肿，因此引出了MVC模式

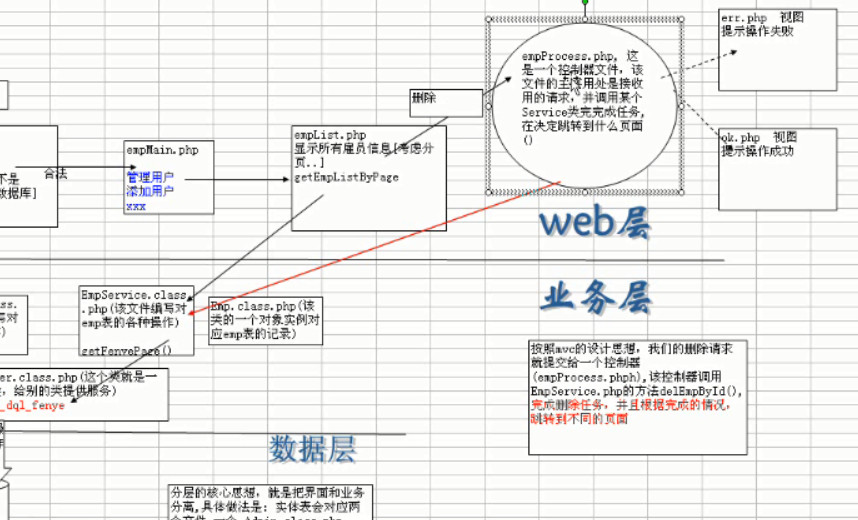
M（模型）V（视图）C（控制器）

1. mvc是一种软件设计模式（类似练武的一种套路） 一定程度上约束了程序员的自由
2. mvc具体每个字母的含义 M处理业务逻辑 （类似这个项目里面的Empservice.php 实体类 工具类） V视图（界面 login 以及 empList.php） C控制器（接收用户的请求，并调用某个service方法，然后跳转到下一个任务）
3. mvc的核心思想，强制将程序员在编写项目的时候，把数据的输入、数据的的处理以及数据的输出分开

注意：尽量减少php对于service类的直接调用



控制器也是在web层



按照mvc的思想，即将删除id的请求交给控制器来完成（empProcess.php）—该控制器调用empService中的删除方法即可，完成相关情况，跳转到不同的页面

MVC小结：

1. 首先控制器接收用户的请求，并决定应该在调用哪个模型（service/）来进行处理；
2. 然后调用模型来处理用户的请求并返回数据；
3. 最后控制器用相应的视图显示模型返回的数据，并通过浏览器呈现给用户。

缺点：加大调试难度，因为有了层的概念；小项目不适宜用mvc模式

一般最不容易出错的地方是：工具类

很容易出错的是service类，以及web层的数据交接的地方

Web/service结构和mvc结构不冲突

addEmp.php 提供一个界面

是不是每一个请求，都需要一个对应的控制器来响应？

原则上是可以的，但是没有这样的必要，可以将同一逻辑的请求，交给一个控制器即可

updateEmp.php 修改用户的界面，里面有待修改用户的相关信息

不让其修改id号码 设置 readonly=”readonly”;即可

在系统中加入上次登陆的时间

<?php

if(!empty($\_COOKIE['lastVisit'])){

echo "上次登陆的时间是：".$\_COOKIE['lastVisit'];

setcookie("lastVisit",date("Y-m-d H:i:s"),time()+3600\*24\*30);

}else{

echo "欢迎登陆";

setcookie("lastVisit",date("Y-m-d H:i:s"),time()+3600\*24\*30);

}

?>