

第10章

online 影视 365 网

(Apache+PHP+ADODB+phpMyAdmin+MySQL 5.0 实现)

在全球知识经济和信息化高速发展的今天，网络化是企业发展的趋势，21世纪的人更习惯在网站上听音乐、看电影，所以企业在同领域中得到突飞猛进的发展，就必须借助网络。

当今社会进入了一个信息快速发展的社会，在网络世界浏览信息的同时也可以听听歌曲，这样既愉悦了身心，又得到了最新的市场动态，由此网络上也出现了很多的影视网站，都很受到欢迎。未来视听生活的新空间，也必然在宽带互联网上开启。VOD 的概念已经被越来越多的人所接受，逐渐成为网络发展的必然趋势之一。通过阅读本章，可以学习到：

- ▶ online 影视 365 类网站的整体设计思路及实现方法
- ▶ 应用<embed>和<object>播放影音文件
- ▶ 关闭子窗口时自动刷新父窗口
- ▶ 制作无边框的窗口
- ▶ ADODB 介绍及使用
- ▶ 更改数据库时的注意事项

10.1 开发背景

×××影视有限公司是一家以影视为主的公司。为了扩大企业规模，增强企业的竞争力，决定向多元化发展，计算机技术、网络通信技术、多媒体技术及数据存储技术的飞速发展，对人类的生产和生活方式产生了很大影响。网络传播以其特有的快速、高效、便捷的传输方式直接被人们所接受。随着多媒体数据的存储、传输和应用技术的不断成熟，以及宽带网络的不断发展，我们有理由相信宽带点播一定会成为网络内容创新的重头戏。

10.2 需求分析

根据客户要求及成本核算，本系统所要实现的功能如下：

- 网站实行会员等级管理，分“普通会员”和“高级会员”两种。
- 网站提供下载功能和在线视听功能，不同的会员等级实现的功能操作也不同。
- 网站分前台和后台两部分，没有权限，用户无法进入后台进行管理。
- 独特的点歌模块，为网站的会员提供点歌平台。
- 前期数据量比较少，可以使用小型数据库以节约成本，如果后期数据量增大，则可以更换大型数据库。

10.3 系统分析

10.3.1 系统目标

online 影视 365 网主要实现以下目标：

- 自定义的鼠标样式增强了网站界面的和谐性。
- 采用无边框窗口技术，保障系统的安全性。
- 信息搜索灵活、方便。
- 采用多媒体引用技术播放影音文件。
- 网站采用文件上传和下载技术实现图像、音频、视频文件的客户端上传和下载。

10.3.2 系统功能结构

根据 online 影视 365 网的特点，可以将其分为前台和后台两个部分设计。前台主要实现在线视听、影视音乐上传、影视音乐下载、在线聊天和在线点歌等功能；后台主要用于管理员对影视音乐目录、数据信息和上传日志进行管理。



online 影视 365 网的前台功能结构如图 10.1 所示。

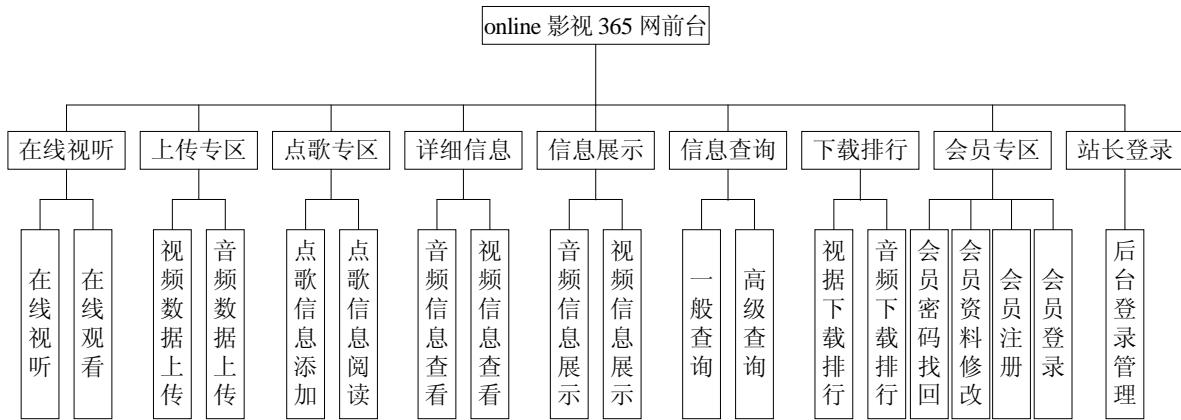


图 10.1 online 影视 365 网前台功能结构图

online 影视 365 的后台功能结构如图 10.2 所示。

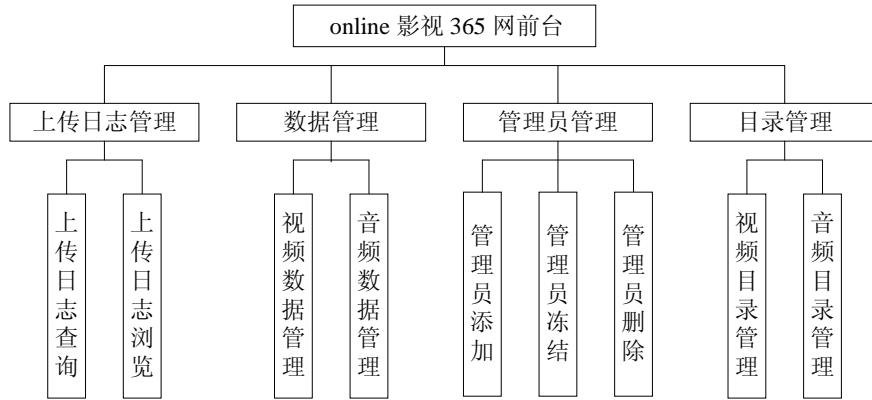


图 10.2 online 影视 365 网后台功能结构图

10.3.3 功能预览

onlie 影视 365 网由多个功能模块组成，为了让读者对本系统有个初步的了解和认识，下面列出几个典型功能的页面，其他页面参见光盘中的源程序。

影视专区页面如图 10.3 所示，该页面展示所有的视频信息及其分类。音乐专区页面如图 10.4 所示，该页面展示所有的音频信息及其分类。

最新影视页面如图 10.5 所示，该页面用于显示所有的最新上传的影视文件。注册页面如图 10.6 所示，该页面显示了游客注册时需要填写的昵称、密码和个人资料等相关内容。

PHP 项目开发全程实录



图 10.3 影视专区 (光盘\TM\10\online\list.php?action=audio)



图 10.4 音乐专区 (光盘\TM\10\online\list.php?action=video)



图 10.5 最新影视 (光盘\TM\10\online\ list.php)



图 10.6 音乐专区 (光盘\TM\10\online\ register.php)

查询页面如图 10.7 所示, 该页显示简单查询所查到的音频、视频文件列表。高级查询页面如图 10.8 所示。该页面用于模糊查询音、视频文件, 包含了多个字段, 可任意选择。



图 10.7 简单查询 (光盘\TM\10\online\show.php)

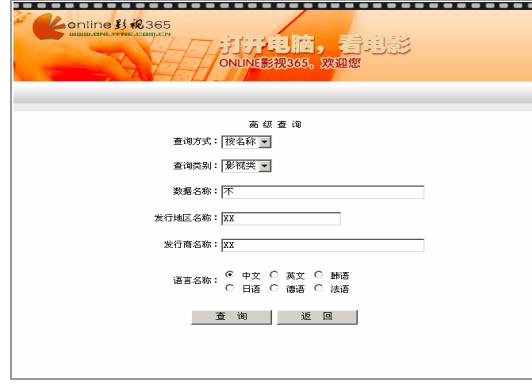


图 10.8 高级查询 (光盘\TM\10\online\high.php)

10.3.4 系统流程图

online 影视 365 网的系统流程图如图 10.9 所示。

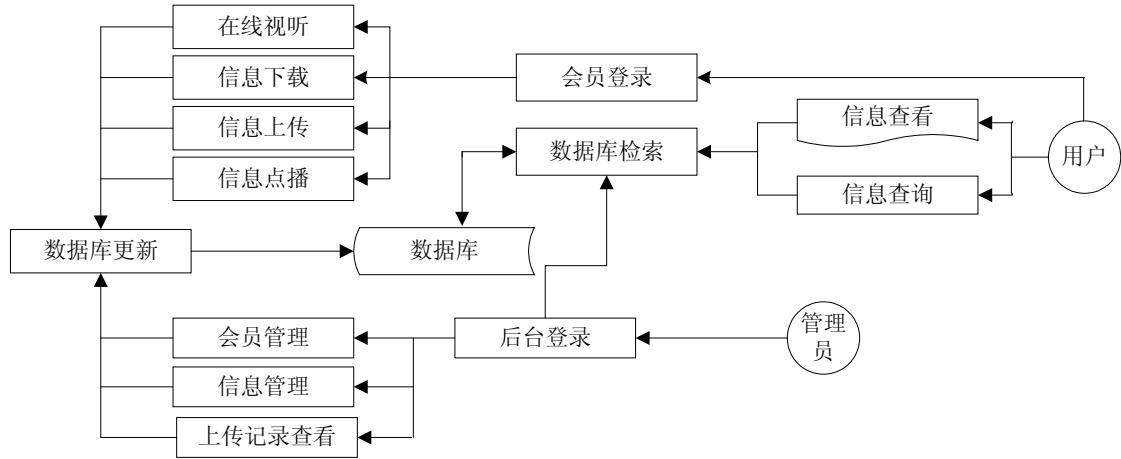


图 10.9 online 影视 365 网的系统流程图

10.3.5 开发环境

在开发 online 影视 365 网时，该项目使用的软件开发环境如下：

1. 服务器端

- 操作系统：Windows Server 2003。
- 服务器：Apache 2.0。
- PHP 软件：PHP 5.0。
- 数据库：MySQL 5.0。
- MySQL 图形化管理软件：PhpMyAdmin-2.5.5。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0。
- 分辨率：最佳效果 1024×768 像素。

2. 客户端

- 浏览器：推荐使用 IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

10.3.6 文件夹组织结构

online 影视 365 网包括前台和后台两部分，所以文件夹结构也主要由两部分组成。online 影视 365 网的组织结构图如图 10.10 所示。



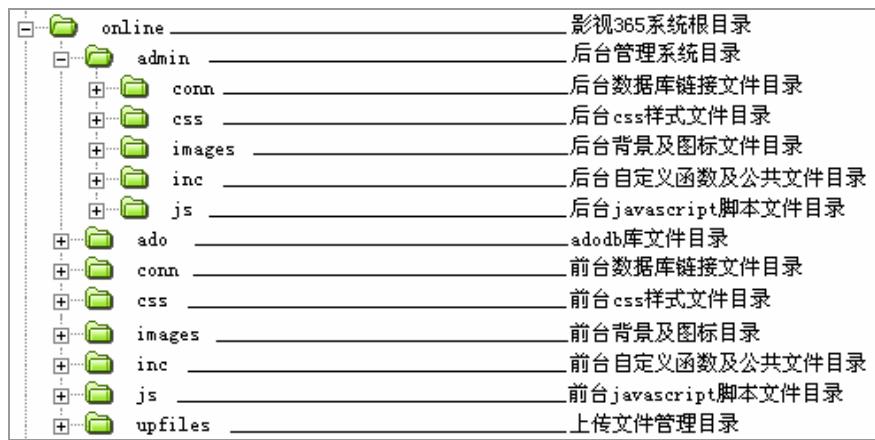


图 10.10 online 影视 365 网的文件夹组织结构图

10.4 数据库设计

10.4.1 数据库分析

本系统使用的是 MySQL 数据库，但使用的是 ado 连接方式，这是为了便于以后数据负担加重后，可以更改其他数据库，如使用 oracle。只要更改数据库而无须重新编写源程序。

10.4.2 数据库概念设计

通过需求分析和功能上的设计，本系统规划出管理员信息实体、会员信息实体、视频信息实体、音频信息实体、视频目录实体和音频目录实体。下面给出主要的实体及 E-R 图。

会员信息实体包括注册用户的详细个人信息，如果想下载或在线视听，则必须注册为会员才可以。会员信息实体 E-R 图如图 10.11 所示。

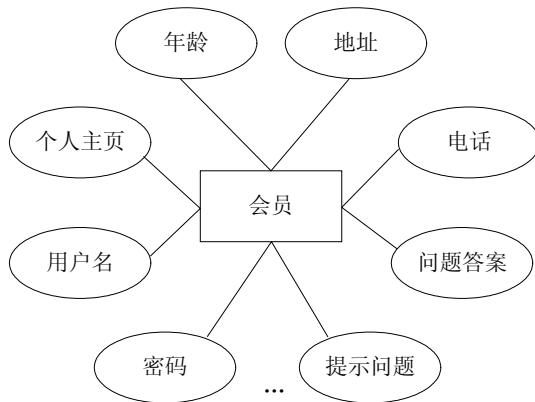


图 10.11 会员信息实体 E-R 图

视频文件实体包括视频名称、视频图片、视频文件、主要演员、导演、发行商等多项资料，视频信息实体 E-R 图如图 10.12 所示。

音频目录实体包括音频目录分类的相关信息，如目录名称，目录级别、父目录名称等，音频目录实体 E-R 图如图 10.13 所示。

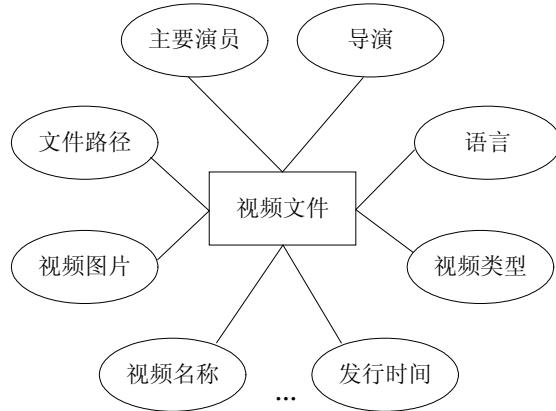


图 10.12 视频信息实体 E-R 图

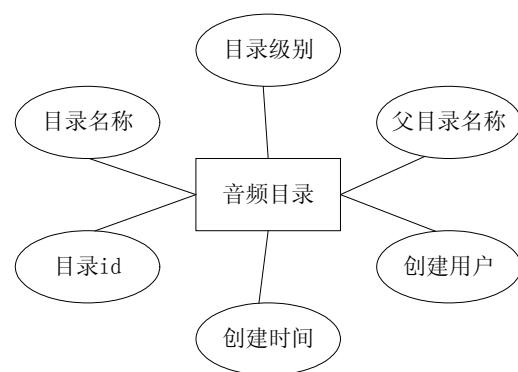


图 10.13 音频目录实体 E-R 图

10.4.3 数据库物理结构设计

online 影视 365 网的数据库中共包含 8 个数据表，如图 10.14 所示。

服务器: localhost ▶ 数据库: db_online			
表	类型	整理	说明
tb_account	MyISAM	gb2312_chinese_ci	会员信息列表
tb_audio	MyISAM	gb2312_chinese_ci	视频信息列表
tb_audiolist	MyISAM	gb2312_chinese_ci	视频目录列表
tb_grade	MyISAM	gb2312_chinese_ci	等级限定列表
tb_manager	MyISAM	gb2312_chinese_ci	管理员列表
tb_register	MyISAM	gb2312_chinese_ci	点歌信息列表
tb_video	MyISAM	gb2312_chinese_ci	音频信息列表
tb_videolist	MyISAM	gb2312_chinese_ci	音频目录列表

图 10.14 db_online 库中的数据库列表

下面来看一下各个表的结构设计。

1. tb_account (会员信息列表)

会员信息列表主要存储用户的个人信息，tb_account 表的结构如图 10.15 所示。

服务器: localhost > 数据库: db_online > 表 : tb_account						
字段	类型	整理	属性	Null	默认	说明
id	int(11)			否		自动编号
name	varchar(30)	gb2312_chinese_ci		否		会员昵称
password	varchar(30)	gb2312_chinese_ci		否		会员密码
question	varchar(50)	gb2312_chinese_ci		否		密码提示问题
answer	varchar(50)	gb2312_chinese_ci		否		密码提示答案
realName	varchar(30)	gb2312_chinese_ci		否		会员真实姓名
numbers	varchar(20)	gb2312_chinese_ci		否		身份证号
sex	varchar(10)	gb2312_chinese_ci		否		会员性别
age	int(4)			否		会员年龄
job	varchar(50)	gb2312_chinese_ci		否		所做工作
email	varchar(100)	gb2312_chinese_ci		否		E-mail
address	varchar(100)	gb2312_chinese_ci		否		联系地址
phone	varchar(20)	gb2312_chinese_ci		否		联系电话
qq	varchar(20)	gb2312_chinese_ci		否		联系QQ
http	varchar(100)	gb2312_chinese_ci		否		个人主页
counts	int(4)			否	0	上传次数
grade	varchar(10)	gb2312_chinese_ci		否	普通会员	会员级别
whether	varchar(10)	gb2312_chinese_ci		否	1	是否激活

图 10.15 会员信息列表结构

2. tb_audio (视频信息列表)

视频信息列表主要上传视频的资料，如视频名称、视频图片等。tb_audio 表的结构如图 10.16 所示。

服务器: localhost > 数据库: db_online > 表 : tb_audio						
字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
name	varchar(100)	gb2312_chinese_ci		否		视频名称
picture	varchar(200)	gb2312_chinese_ci		否		封面图片
sizes	varchar(50)	gb2312_chinese_ci		否		视频大小
grade	varchar(10)	gb2312_chinese_ci		否		视频级别
publisher	varchar(100)	gb2312_chinese_ci		否		发行公司
actor	varchar(200)	gb2312_chinese_ci		否		主要演员
director	varchar(100)	gb2312_chinese_ci		否		导演
marker	varchar(100)	gb2312_chinese_ci		否		制片
languages	varchar(20)	gb2312_chinese_ci		否		语种
type	varchar(50)	gb2312_chinese_ci		否		类型
style	varchar(50)	gb2312_chinese_ci		否		类别
froms	varchar(100)	gb2312_chinese_ci		否		视频所属地区
publishTime	date			否		发行时间
remark	varchar(1000)	gb2312_chinese_ci		否		视频介绍
property	varchar(20)	gb2312_chinese_ci		否		视频属性
address	varchar(200)	gb2312_chinese_ci		否		存储地址
username	varchar(50)	gb2312_chinese_ci		否		发布人姓名
issueDate	datetime			否		发布时间
downTime	int(4)			否	0	下载次数
lastTime	datetime			否		最后下载时间
whether	varchar(20)	gb2312_chinese_ci		否	1	是否为新品

图 10.16 视频信息列表结构



3. tb_audiolist (视频目录列表)

视频目录列表主要是上传视频所属的类型（二级目录）及类别（一级目录）。tb_audiolist 表的结构如图 10.17 所示。

字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
grade	varchar(20)	gb2312_chinese_ci		否		目录级别
name	varchar(50)	gb2312_chinese_ci		否		目录名称
father	varchar(50)	gb2312_chinese_ci		否		父级目录
userName	varchar(30)	gb2312_chinese_ci		否		创建人昵称
issueDate	datetime			否		创建时间

图 10.17 视频目录列表结构

4. tb_manager (管理员列表)

管理员列表主要用于后台管理员的资料信息管理，包括管理员名称、管理员权限等。tb_manager 表的结构如图 10.18 所示。

字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
name	varchar(50)	gb2312_chinese_ci		否		管理员昵称
password	varchar(50)	gb2312_chinese_ci		否		管理员密码
type	varchar(50)	gb2312_chinese_ci		否		管理员类型
realName	varchar(50)	gb2312_chinese_ci		否		真实姓名
issueDate	datetime			否		创建时间
whether	varchar(50)	gb2312_chinese_ci		否		是否激活

图 10.18 管理员列表结构

5. tb_register (点歌信息列表)

点歌信息列表主要存储等会员用户之间的点歌祝福，主要包括发送人、接收人、歌曲地址等。tb_register 表的结构如图 10.19 所示。

字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
name	int(4)			否		歌曲地址
fromName	varchar(50)	gb2312_chinese_ci		否		发送人姓名
toName	varchar(50)	gb2312_chinese_ci		否		接收人姓名
remark	varchar(50)	gb2312_chinese_ci		否		备注
issueDate	datetime			否		点歌时间

图 10.19 点歌信息列表结构

6. tb_video (音频信息列表)

音频信息列表主要用于音频文件的信息管理,如歌手、歌词等。tb_video 表的结构如图 10.20 所示。

服务器: localhost ▶ 数据库: db_online ▶ 表 : tb_video						
字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
name	varchar(100)	gb2312_chinese_ci		否		音频名称
picture	varchar(200)	gb2312_chinese_ci		否		封面图片
actor	varchar(100)	gb2312_chinese_ci		否		主唱
ci	varchar(50)	gb2312_chinese_ci		否		歌词作者
qu	varchar(50)	gb2312_chinese_ci		否		歌曲作者
actortype	varchar(50)	gb2312_chinese_ci		否		歌手类型
style	varchar(50)	gb2312_chinese_ci		否		音频类别
publisher	varchar(100)	gb2312_chinese_ci		否		发行商
froms	varchar(100)	gb2312_chinese_ci		否		发行地区
type	varchar(50)	gb2312_chinese_ci		否		音频类型
sizes	varchar(50)	gb2312_chinese_ci		否		音频大小
languages	varchar(20)	gb2312_chinese_ci		否		音频语言
publishTime	date			否		发行时间
remark	varchar(1000)	gb2312_chinese_ci		否		备注
property	varchar(20)	gb2312_chinese_ci		否		音频属性
address	varchar(200)	gb2312_chinese_ci		否		音频地址
userName	varchar(50)	gb2312_chinese_ci		否		发布人姓名
issueDate	datetime			否		发布时间
downTime	int(4)			否		下载次数
lastTime	datetime			否		最后下载时间
whether	varchar(50)	gb2312_chinese_ci		否	1	是否为新品

图 10.20 音频信息列表结构

7. tb_videolist (音频目录列表)

音频目录列表主要用于音频文件的分类管理,有目录名称、目录级别等。tb_videolist 表的结构如图 10.21 所示。

服务器: localhost ▶ 数据库: db_online ▶ 表 : tb_videolist						
字段	类型	整理	属性	Null	默认	说明
id	int(4)			否		自动编号
grade	varchar(50)	gb2312_chinese_ci		否		目录等级
name	varchar(50)	gb2312_chinese_ci		否		目录名称
father	varchar(50)	gb2312_chinese_ci		否		父级目录
userName	varchar(50)	gb2312_chinese_ci		否		创建人姓名
issueDate	datetime			否		创建时间

图 10.21 音频目录列表结构



10.5 前台首页设计

10.5.1 前台首页概述

online 影视 365 网的前台首页的功能模块主要包括以下 3 部分：

- ☒ 网站首部导航栏：主要有影视专区、音乐专区、上传专区和点歌专区。
- ☒ 网站左侧导航栏：包括会员登录模块、搜索模块和影视排行模块。
- ☒ 网站主显示区：包括最新上传模块、视频模块和音频模块

下面看一下本案例中提供的前台首页，该首页在本书光盘中的路径为\TM\10\online\index.php，如图 10.22 所示。



图 10.22 online 影视 365 前台首页

10.5.2 前台首页技术分析

在本章的数据库分析中已经提到，本系统所使用的连接数据库的方式为 ADODB 连接。这里，先来初步了解一下 ADODB 如何连接 MySQL。

下面以本系统使用的数据库和数据表为例，来介绍一下使用 ADODB 的基本操作步骤。

(1) 载入 (include) adodb.in.php 文件。

```
include "../ado/adodb.inc.php";
```

(2) 建立数据库类型连接。

```
$conn = &ADONewConnection("mysq"); or $conn = &NewADOConnection("mysql");
```

(3) 连接要使用的数据库。

```
$conn->PConnect("db_online"); or $conn->Connect("db_online");
```

(4) 执行 SQL 语句。

```
$recordset = $conn->Execute("select * from tb_audio");
```

(5) 判断结果，并对结果处理。

```
if($recordset) //判断execute()函数执行中是否有错误
    while(!$recordset->EOF){ //如果没有错误，则配合while语句循环输出结果
        echo $recordset->fields[0]."<br>";
        $recordset->moveNext(); //指针下移
    }
}
```

(6) 关闭连接。

```
$recordset->close();
$conn->close();
```

如果是对微软的 ado 有所了解的用户，会对上述步骤有一种亲切感。无论从字段名称，还是从功能上，两者都非常相似，所以上手会很轻松。没有接触过 ado 的用户也不用担心，用 ADODB 操作数据库主要就是这 6 个步骤，只要记住这个顺序，做几遍练习后，也会运用自如。

在本章的最后，还有对 ADODB 类库进行较详细的讲解，参见 10.14 节。

10.5.3 前台首页的实现过程

 前台首页使用的数据表：tb_account、tb_audio、tb_video

在首页中，使用到了 3 个 include 语句，将导航栏、登录框、搜索框、浏览区等主要的模块加载进来。限于篇幅，这些页面的代码不做讲解了，请自行到光盘中查找。前台首页的实现代码如下：

例程01 代码位置：光盘\TM\10\online\index.php

```
<?php
```

```
session_start(); //开启session支持
?>
<script src="js/chk.js" language="javascript"></script> //引入js脚本文件
<link rel="stylesheet" href="css/style.css" /> //引入css样式文件
<?php
    include "top.php"; //首部导航及LOGO
?>
<table border="0" cellpadding="0" cellspacing="0">
    <tr><td width="265" align="center" valign="top">
<?php
    include "left.php"; //登录、搜索框
?>
    </td><td width="605" align="center" valign="top">
<?php
    include "main.php"; //主浏览区
?>
    </td></tr></table>
```

10.6 详细信息查看模块

10.6.1 信息查看模块概述

详细信息查看模块主要包括视频详细信息查看和音频详细信息查看两个部分。查看信息功能没有权限限制，无论是游客还是会员都可以进行查看。但查看页面中的下载和在线视听按钮则是只有会员才可以看到的操作，游客只能看到“返回”按钮，用来关闭查看页面的。下面给出信息查看框架图，如图 10.23 所示。

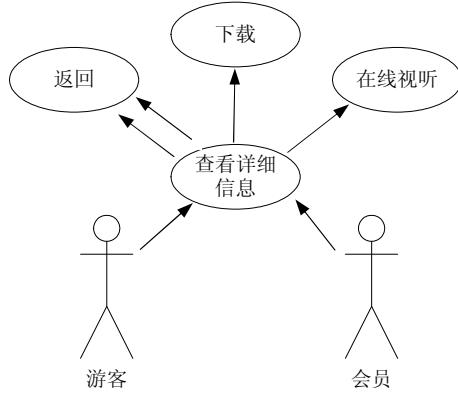


图 10.23 信息查看流程图

10.6.2 信息查看模块技术分析

在这个模块中，所要解决的技术难点就是：系统需要根据不同的浏览用户，显示不同的操作权限。

这里是使用 if 语句，配合 session 来实现其功能的。首先通过检测\$_SESSION[name]是否存在，来判断用户是否登录，如果没有登录，则跳过“在线视听”和“下载”，只显示“返回”按钮；如果用户为登录会员，则显示“在线播放”按钮，接着判断\$_SESSION[grade]是否为“高级会员”，如果登录会员为普通会员，则“下载”按钮为灰色失效状态，如果登录会员为“高级会员”，则同时显示 3 个按钮的全部功能。该功能的关键代码如下：

例程02 代码位置：光盘\TM\10\online\intro.php

```
<?php
    /* 如果用户是登录会员 */
    if($_SESSION[name]<>""){
        ?>
        <!-- 显示“播放”按钮 -->
①      <input name="Submit" type="submit" value=" 播 放 "
onclick="javascript:Wopen=open('operation.php?action=see&id=<?php echo
$rst->fields[16]; ?>',height=700,width=665,scrollbars=no');">
        <?php
            /* 如果是登录会员 */
            }if($_SESSION[name]<>""){
        ?>
            /* 如果是高级会员，则“下载”按钮为激活状态，否则为灰色 */
②            <input name="Submit" type="button" <?php if ($_SESSION[grade]<>"高级会员"){ echo "disabled";}?>
value=" 下 载 " onclick="javascript:Wopen=location='download.php?action=audio&id=<?php echo
$rst->fields[16]; ?>'";>
            <?php
                }
        ?>
        /* 显示“返回”按钮 */
③      <input name="Submit2" type="button" value=" 返 回 " class="submit"
onClick="javascript:top.window.close();"></td>
```

代码贴士

- ① open()方法可以打开一个无边框的子窗口。关于 open 方法的介绍，可以参考 10.13 章。
- ② disabled：译为无效的、不起作用的，这里是 button 按钮的属性，即不可点击状态。
- ③ top.window.close()：关闭弹出窗口。

10.6.3 信息查看的实现过程

 消息管理模块使用的数据表：tb_audio、tb_video

用户可以在信息展示页面单击  图标进入详细信息展示页面。详细信息展示页面主要是根据传递的参数进行数据库检索，并将结果集输出到浏览器中。查看页面的运行结果如图 10.24 所示。



图 10.24 详细信息页面的运行结果

在详细信息展示页面（intro.php）中，通过 id 值从数据表中提取所要显示的信息输出到页面，当文件信息输出完毕后，通过 session 值来判断访问者是游客还是会员，如果是游客，则只显示“返回”按钮；如果是会员，还要显示“在线试听”按钮和“下载”按钮。程序的关键代码如下：

例程03 代码位置：光盘\TM\10\online\intro.php

```
<?php
    session_start(); //开启session支持
    include "conn/conn.php"; //载入数据库连接文件
?>
<link rel="stylesheet" href="css/style.css"> //引入css样式文件
<!-- 显示数据详细信息 -->
...
<?php $sql="select * from tb_audio where id=".$_GET[id]; //根据id生成查询语句
        $rst = $conn->execute($sql); //执行SQL语句，返回结果集
/* 如果结果集不为空，则输出信息 */
if(!$rst->EOF){
?>
```

```

<table width="400" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="15" colspan="2">&nbsp;</td>
  </tr>
  <!-- 输出视频名称 -->
  <tr>
    <td width="131" height="20" align="right" valign="middle">名称: </td>
    <td width="269" height="20"><?php echo $rst->fields[1]; ?></td>
  </tr>
  <!-- 输出文件大小 -->
  <tr>
    <td height="20" align="right" valign="middle">大小: </td>
    <td height="20"><?php echo $rst->fields[3]; ?></td>
  </tr>
  ...
  ...
  <tr>
    <td height="30" colspan="2" align="center" valign="middle">
      /* 根据不同用户，显示不同权限 */
    </td>
  </tr>
...

```

10.7 点歌模块设计

10.7.1 点歌模块概述

用户通过主导航条进入点歌专区。在点歌专区，可以对专区内的音乐进行试听、浏览音频信息的详细内容和打开点歌页面进行点歌，进行点歌的前提条件是用户必须登录。点歌模块的用例图如图 10.25 所示。

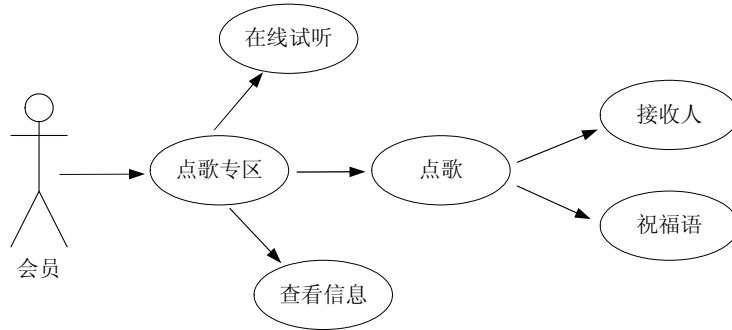


图 10.25 点歌模块的用例图

10.7.2 点歌模块技术分析

本系统的在线点歌模块实现了会员间发送祝福与点歌的功能，收到祝福的会员可以实现在线唱歌。



本节和下一节的“在线视听模块”中，将逐一介绍实现在线视听的两种方法。本节先来介绍 html 标签中的播放多媒体标签<embed>。

1. 基本语法

```
<embed src=url>
```

url 为音频或视频文件的其路径，可以是相对路径，也可以是绝对路径。

<embed>可以用来播放各种多媒体文件，格式可以是 midi、wav、mp3 等，当下主流的浏览器都支持该标签。

2. 属性设置

<embed>标签有很多属性，常用的属性及说明如表 10.1 所示。

表 10.1 <embed>常用属性及说明

属性名	说 明	举 例
autoplay	该属性规定音频或视频文件是否在下载完之后就自动播放 true：音乐下载完成后自动播放。false 下载完成后不播放	<embed src="1.mp3" autoplay=true>
loop	该属性规定音频或视频文件是否循环及循环次数 属性值为 true 时，音频或视频文件循环；属性值为 false 时，音频或视频文件不循环。如果为正整数，则为循环次数	<embed src="1.mp3" loop=true>
hidden	该属性规定控制面板是否显示，默认值为 no ture：隐藏面板；no：显示面板	<embed src="1.mp3" hidden="yes">
starttime	该属性规定音频或视频文件开始播放的时间。未定义则从文件开头播放	<embed src="1.mp3" starttime="00:10">
volume	该属性规定音频或视频文件的音量大小。未定义则使用系统本身的设定	<embed src="1.mp3" volume="10">
width	该属性规定了控制面板的宽度	<embed src="1.mp3" width="100">
height	该属性规定了控制面板的高度	<embed src="1.mp3" height="200">
title	该属性规定音频或视频文件的说明文字	<embed src="1.mp3" title="a good song">

除了这些属性外，<embed>标签还可以面板的外观进行设置，对面板的背景色和前景色进行设置，感兴趣的读者可以查找相关的资料。

10.7.3 点歌模块的实现过程

■ 消息管理模块使用的数据表：tb_register、tb_video

单击“点歌专区”超链接，首先进入到点歌专区页面，页面中的音乐试听和信息查看在其他模块中都有介绍，这里主要看一下点歌功能。单击小图标，将进入到点歌信息页面，点歌页面的运行结果如图 10.26 所示。





图 10.26 点歌模块的运行结果

在点歌信息页面,需要用户添加祝福语和接收人名称。点歌页面涉及到的 HTML 表单元素如表 10.2 所示。

表 10.2 点歌页面中的主要表单元素

名 称	类 型	重 要 属性	含 义
list	form	action="Register.asp?action=add"	表单
toname	text	id="toname" size="30"	接收人名称
remark	textarea	cols="40" rows="5" id="remark"	简要介绍
id	Hidden	value=" php echo \$_GET[id] ?>"</td <td>所点歌曲的地址</td>	所点歌曲的地址
submit	button	onClick="register()"	“点歌”按钮
submit2	button	onClick="javascript:top.window.close()"	“返回”按钮

在点歌页面，将前页传过来的歌曲 id、接收人、祝福语等信息组成 insert 语句存储到数据库中，不管保存成功与否，都将关闭当前子窗口，回到父窗口中进行操作。程序代码如下：

例程04 代码位置：光盘\TM\10\online\give.php

```
<?php
    session_start(); //开启session支持
    ...
    if($_POST[toname] <> ""){ //如果接收人不为空
        $id=$_POST[id];
        $toname=$_POST[toname];
        $from=$_SESSION[name];
        $remark=$_POST[remark];
        <!-- 生成insert语句 -->
        $sql="insert into tb_register Values(\".$id.\",\".$from.\",\".$toname.\",\".$remark.\",\".date(\"Y-m-d H:i:s\").\")";
        $rst = $conn->execute($sql); //执行SQL语句
    }
    <?
    <script language="javascript">
    <?php
        if(!($rst == false)){
    ?>
            alert("点歌信息保存成功");
    <?php
        }else{
    ?>
            alert("点歌失败");
    <?php
        }
    ?>
        top.window.close();
    </script>
    <?php
    }
    ?>
```

10.8 在线视听模块设计

10.8.1 在线视听模块概述

在线视听主要包括视频数据在线观看和音频数据在线收听两个部分。在不同的页面都可以直接进入视听页面，包括首页、搜索结果页、内容介绍页和列表页。但必须是登录会员才可以，游客是没有权限的。下面给出在线视听流程图如图 10.27 所示。

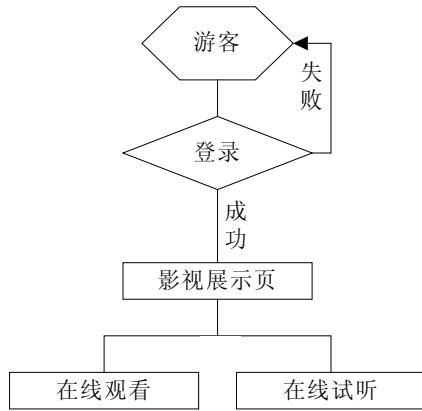


图 10.27 在线试听流程图

10.8.2 在线视听模块技术分析

在 10.7.2 中学习了如何使用<embed>多媒体标签来播放影音文件。本节来学习另一种方法:<object>标签。

使用<object>标签来定义一个嵌入的对象。该对象向 HTML 页面添加多媒体。此元素运行插入到 html 页面中的对象的数据和参数，以及可用来显示和操作数据的代码。<object>标签的基本格式为：

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" height="68" id="MediaPlayer1" width="460">
    <param name="ShowStatusBar" value="-1">
    <param name="Filename" value="<?php echo $_GET[id]?>">
</object>
```

- classid: 定义嵌入 Windows Registry 中或某个 URL 中的类的 ID 值，此属性可用来制定浏览器中包含的对象的位置。
- param: 可定义用于对象的 run-time 设置。

<object>是微软专门为 IE 浏览器打造的、可以扩展外部应用程序及插件的对象标签，它和<embed>的不同之处在于：<object>只支持以 IE 技术为核心的浏览器系列，对其他的浏览则无效。而且如果要使用<object>来播放多媒体，那么需要安装相应的播放插件，如 realplay、quicktime 等。

10.8.3 在线视听模块的实现过程

- 在线视听模块使用的数据表：tb_audio、tb_video

会员可以在首页、信息展示页面或者详细信息展示页面单击“观看”或“试听”按钮进行在线视听操作。页面运行结果如图 10.28 所示。



图 10.28 在线视听模块的运行结果

以视频详细信息展示页面为例，用户可以单击 图标，打开视频播放页面（see.php）进行在线观看。视频播放页面主要是根据传递的参数进行数据库检索，并将对应数据的硬盘存储地址作为多媒体文件的引用地址，当页面加载完毕后，视频数据将自动播放，实现在线观看功能。在线播放页面的主要代码如下：

例程05 代码位置：光盘\TM\10\online\see.php

```
<?php
    session_start(); //开启session支持
?>
<body>
    <!-- 使用object对象播放视频/音频文件 -->
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" width="665" height="500" id="MediaPlayer1" >
    <!-- 设置自动播放 -->
    <param name="AutoStart" value="-1">
        <!-- 设置状态栏-->
    <param name="ShowStatusBar" value="-1">
        <!-- 播放文件路径 -->
    <param name="Filename" value="upfiles/audio/<?php echo $_GET[id]; ?>">
</object>
</body>
```

10.9 后台首页设计

10.9.1 后台首页概述

根据用户对各个功能模块的使用频率和重要程度，本系统的首页面中要显示的模块主要有两部分：

(1) 网站左侧导航栏：包括各个管理模块及分类。

- 目录管理模块：主要包括视频目录管理和音频目录管理两个部分。
- 数据管理模块：主要包括视频数据管理和音频数据管理两个部分。
- 会员管理模块：包括会员等级管理、会员信息冻结、会员信息删除 3 个部分。
- 上传信息管理模块：主要用于浏览和查询用户上传操作的详细内容。
- 管理员设置模块：主要包括管理员信息添加、删除、冻结 3 个部分。

(2) 网站主显示区：显示各个模块的操作及结果。

下面看一下本案例中提供的后台首页，该首页在本书光盘中的路径为\TM\10\online\admin\index.php，如图 10.29 所示。



图 10.29 online 影视 365 后台首页

10.9.2 后台首页技术分析

本系统的后台首页采用的布局结构为二分栏布局。左侧导航栏清楚、明白地显示了后台管理员所能使用到的功能。当单击任意功能按钮时，在主显示区显示对应的操作界面，和该功能模块下的子功能。页面简练、结构清晰、浏览方便，二分栏的主要特点被表现的淋漓尽致。关于二分栏的详细介绍，请看 3.6.3 节和 4.7.1 节。

10.9.3 后台首页的实现过程

 后台首页使用的数据表: tb_manager

下面来看一下后台首页的实现过程。在首页代码中, 对每项功能权限的判断都是在 left.php 中完成的, 只要在左侧导航栏的位置载入 left.php 即可。在主显示区, 根据参数返回的不同值, 显示不同的功能界面。程序的关键代码如下:

例程06 代码位置: 光盘\TM\10\online\admin\main.php

```
<?php
    session_start();                                //开启session支持
    include "inc/check.php";                         //载入权限检查文件
    include "conn/conn.php";                          //载入数据库链接文件
?>
<link href="css/style.css" rel="stylesheet"/>      //引入css样式文件
<script src="js/admin_js.js" language="javascript"></script>
<center>
...
<?php
/* 载入网站左侧导航栏 */
    include "left.php";
?>
...
<div style="height:35px;">&nbsp;</div>
<?php
/* 根据不同的参数, 显示不同的功能界面 */
if(isset($_GET[action])){
    switch ($_GET[action]){
        case "audioList":                           //视频目录名称添加
            include "a_list.php";
            break;
        case "videoList":                           //音频目录添加
            include "v_list.php";
            break;
        case "audio":                               //视频数据添加
            include "audio.php";
            break;
        case "video":                               //音频数据添加
            include "video.php";
            break;
        case "grade":                               //会员等级设置
            include "grade.php";
            break;
        case "member":                             //会员数据设置
            include "member.php";
            break;
        case "log":                                 //上传日志管理
            include "log.php";
            break;
        case "manager":                            //管理员设置
            include "manager.php";
            break;
    }
}
?>
```



```

        include "manager.php";
        break;
    }
?>      </td></tr>
<tr><td height="15" bgcolor="#f0f0f0">&nbsp;</td></tr>
</table>
</td></tr>
</table>
</center>

```

10.10 目录管理模块设计

10.10.1 目录管理模块概述

目录管理模块主要包括视频目录管理和音频目录管理两个部分。管理员可以通过后台管理导航进入对应的目录管理页面。

如果管理员进入了视频目录管理页面（audiolist.php），那么在该页面内可以打开目录添加页面进行目录添加操作，也可以删除相应的目录。鉴于目录信息比较简短，本例中没有提供目录修改的功能。目录管理模块的用例图如图 10.30 所示。

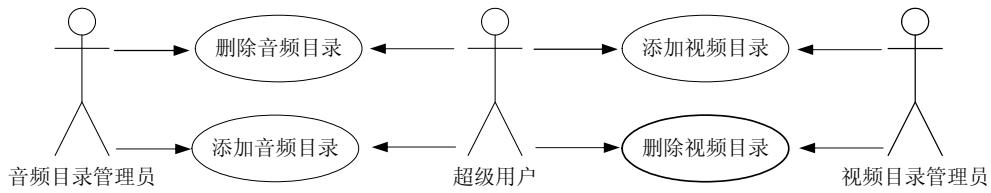


图 10.30 视频目录管理用例图

说明：由于视频目录管理和音频目录管理实现的方法基本类似，因此，本章重点讲解视频目录管理模块实现的方法，关于音频目录管理模块的实现方法请参见本书附赠源码光盘。

10.10.2 目录管理模块技术分析

在本系统中，有个功能经常要被用到：就是在添加目录、歌曲时，经常会碰到重名的问题。对于这个问题，可以写一个自定义函数，并将它放到单独的一个文件中，方便其他页面调用，如本系统中的/inc/func.php，就是专门用来存储自定义函数的文件。

下面来看一下自定义函数 is_chk()。

例程07 代码位置：光盘\TM\10\online\admin\inc\func.php

```

//判断目录名是否重复
//$f_fields: 字段名
//$tablename: 数据表名
//$f_str: 要查找的字段
function is_chk($f_fields,$tablename,$f_str){

```

```

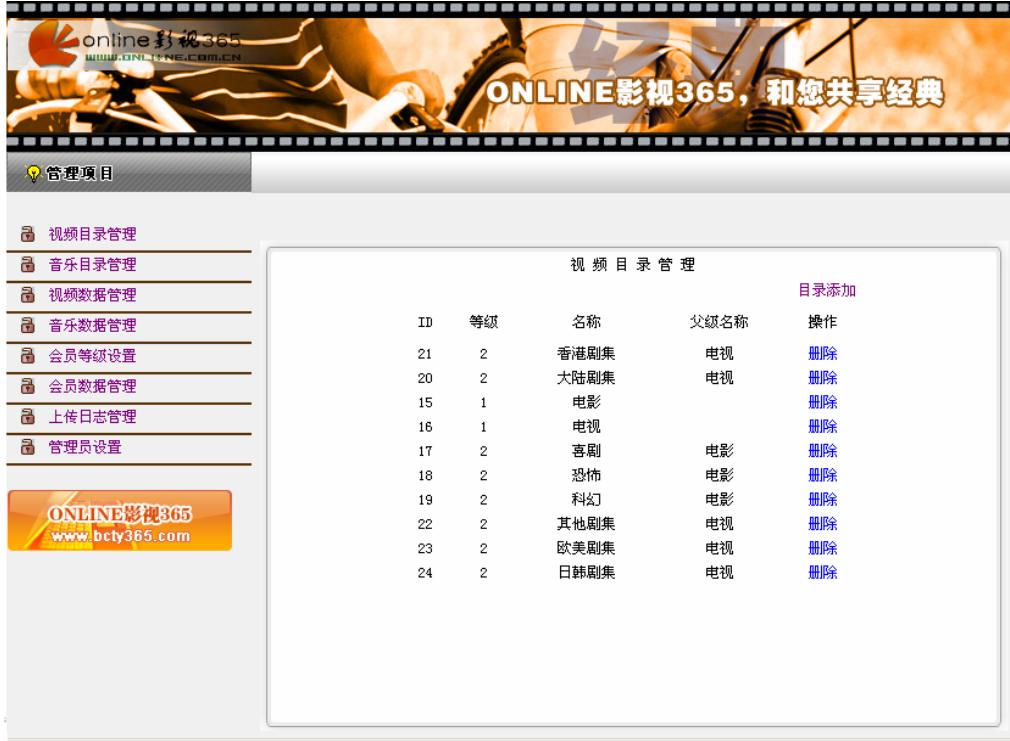
$conn = &ADONewConnection('mysql');           //建立mysql连接
$conn->PConnect("localhost","root","root","db_online"); //连接"db_online"数据库
$is_chk = true;                           //声明一个boolean型变量
$is_sqlstr = "select $f_fields from $tablename"; //查找字段
$is_rst = $conn->execute($is_sqlstr);      //循环输出记录集
while(!$is_rst->EOF){
    **** 对比字段中的值 ****
    if($f_str == $is_rst->fields[0]){
        /* 如果字段在数据库中找到, 说明重名了, is_chk设置成false */
        $is_chk = false;
        break;
    }
    $is_rst->MoveNext();                     //将指针指向下一条记录
}
return $is_chk;                           //返回变量$is_chk
}

```

10.10.3 目录管理的实现过程

 目录管理使用的数据表: tb_audiolist、tb_videolist

单击左侧导航栏中的“视频目录管理”超链接, 可以打开视频目录管理页面, 在管理页面中, 有一个“目录添加”超链接、所有的一、二级目录信息和对应的“删除”按钮。视频管理页面的运行结果如图 10.31 所示。



视频目录管理				
目录添加				
ID	等级	名称	父级名称	操作
21	2	香港剧集	电视	删除
20	2	大陆剧集	电视	删除
15	1	电影	电视	删除
16	1	电视	电视	删除
17	2	喜剧	电影	删除
18	2	恐怖	电影	删除
19	2	科幻	电影	删除
22	2	其他剧集	电视	删除
23	2	欧美剧集	电视	删除
24	2	日韩剧集	电视	删除

图 10.31 视频目录管理的运行结果

视频目录管理页面的代码很简单：使用 while 循环将数据库中的记录全部显示出来，并在每输出一条记录后，添加一个“删除”操作。目录管理页面的实现代码如下：

例程08 代码位置：光盘\TM\10\online\admin\a_list.php

```
<?php
    session_start();                                //开启session支持
    include "conn/conn.php";                         //载入数据库链接文件
    include "inc/check.php";                         //载入权限检查文件
    /* 输出视频目录数据表中的全部数据 */
    $l_sqlstr = "select * from tb_audiolist";
    $l_rst = $conn->execute($l_sqlstr);

?
...
<tr>
/* 目录添加 */
<td height="10" colspan="5" align="right" valign="middle"><a href="#" onclick="javascript:Wopen=open('operation.php?action=audiolist','添加目录','height=500,width=665,scrollbars=no');">目录添加</a></td>
</tr>
<!-- 显示目录相关信息的字段名 -->
<tr>
    <td height="30" align="center" valign="middle">ID</td>
    <td height="30" align="center" valign="middle">等级</td>
    <td height="30" align="center" valign="middle">名称</td>
    <td height="30" align="center" valign="middle">父级名称</td>
    <td height="30" align="center" valign="middle">操作</td>
</tr>
<?php
    while(!$l_rst->EOF){
?
<tr>
    <!-- 输出目录id -->
    <td height="18" align="center" valign="middle"><?php echo $l_rst->fields[0]; ?></td>
    <!-- 输出目录等级 -->
    <td height="18" align="center" valign="middle"><?php echo $l_rst->fields[1]; ?></td>
    <!-- 输出目录名称 -->
    <td height="18" align="center" valign="middle"><?php echo $l_rst->fields[2]; ?></td>
    <!-- 输出目录父级名称 -->
    <td height="18" align="center" valign="middle"><?php echo $l_rst->fields[3] ?></td>
    <!-- 输出“删除”超链接 -->
    <td height="18" align="center" valign="middle"><a href="del_list_chk.php?action=audiolist&id=<?php echo $l_rst->fields[0]; ?>" onclick="return del_chk();">删除</a>
    </td>
</tr>
<?php
    /* 指针下移一条记录 */
    $l_rst->MoveNext();
}
?
</table></td>
</tr>
```



```

</table></td>
</tr>
</table>

```

10.10.4 目录添加的实现过程

■ 目录添加使用的数据表: tb_audiolist、tb_videolist

单击视频目录页面中的“目录添加”按钮, 将打开目录添加页面。目录添加页面的运行结果如图 10.32 所示。



图 10.32 视频添加目录的运行结果

视频目录添加页面 (audiolist.php) 中主要包含一个添加目录表单, 表单的主要元素如表 10.3 所示。

表 10.3 视频目录添加页面涉及到的 HTML 表单元素

名 称	类 型	重 要 属性	含 义
list	form	method="post" action="audiolist_chk.php"	视频目录添加表单
grade	select	name="grade" OnChange="check()" <option value="1" selected>一级目录</option> <option value="2">二级目录</option>	目录等级
father	select	<?php \$l_sqlstr = "select * from tb_audiolist where grade = '1"'; \$l_rst = \$conn->execute(\$l_sqlstr); while(!\$l_rst->EOF){ ?> <option value=<?php echo \$l_rst->fields[2]; ?>><?php echo \$l_rst->fields[2]; ?>></option> <?php \$l_rst->MoveNext(); ?>	父级目录名称
names	text	type="text" id="names"	目录名称
Submit	submit	class="submit" value="添 加" onclick="return n_chk();"	“添加”按钮
Submit2	button	class="submit" value="返 回" onClick="javascript:top.window.close()"	“返回”按钮



当管理员添加信息提交后，系统转到处理页中（audiolist_chk.php）进行添加操作。首先判断输入目录名称是否与已存在的名称重复，如果重复，回到上一步。如果不重复，则向数据表中添加新记录。视频目录添加的程序代码如下：

例程09 代码位置：光盘\TM\10\online\admin\audiolist_chk.php

```
<?php
    session_start(); //开启session支持
    include "inc/chec.php"; //载入权限检查文件
    include "conn/conn.php"; //载入数据库连接文件
    include "inc/func.php"; //载入自定义函数文件
    /* 使用自定义函数is_chk(), 判断用户名是否重复 */
    if(is_chk("name","tb_audiolist",$_POST[names]) == false){
        echo "<script>alert('名称重复');history.go(-1);</script>"; //如果重复，退到上一步操作
        exit();
    }
    /* 生成insert插入语句 */
    $a_sqlstr = "insert into tb_audiolist (grade,name,father,userName,issueDate)
values($_POST[grade],$_POST[names],$_POST[father],'$_SESSION[admin]',".date("Y-m-d H:i:s").")";
    /* 判断返回结果 */
    if($conn->execute($a_sqlstr) == false)
    /* 如果添加失败，返回上一步 */
        echo "<script>alert('添加失败');history.go(-1);</script>";
    else
    /* 如果添加成功，则刷新父窗口，关闭子窗口 */
        echo "<script>top.opener.location.reload();alert('添加成功');window.close();</script>";
?
>
```

10.10.5 目录删除的实现过程

■ 目录删除使用的数据表：tb_audiolist、tb_videolist

视频目录的删除功能是根据传递的目录 id 号将数据库中对应的记录找到，并删除，程序代码如下：

例程10 代码位置：光盘\TM\10\online\admin\del_list_chk.php

```
<?php
    session_start(); //开启session支持
    include "inc/chec.php"; //载入权限检查文件
    include "conn/conn.php"; //载入数据库链接文件
    if($_GET[action] == "audiolist")
        $t_name = "tb_audiolist"; //如果action等于audiolist
    else if($_GET[action] == "videolist")
        $t_name = "tb_videolist"; //要访问的数据表就为tb_audiolist
    $sqlstr = "delete from ".$t_name." where id =".$_GET[id]; //如果action等于videolist
    /* 如果删除成功，则返回到当前页，否者，退到上一步 */
    if($rst = $conn->execute($sqlstr) == false){
        echo "<script>alert('删除错误！ ".$sqlstr.");history.go(-1);</script>";
    }else
        echo "<script>alert('删除成功');location='$_SERVER[HTTP_REFERER]';</script>";
?
>
```



10.11 数据管理模块设计

10.11.1 数据管理模块概述

数据管理模块主要包括视频数据管理和音频数据管理两个部分。管理员可以通过后台管理导航进入对应的数据管理页面，如视频数据管理页面（audio.php）。在数据管理页面，用户可以打开数据添加页面或删除对应的数据。数据管理模块用例图如图 10.33 所示。

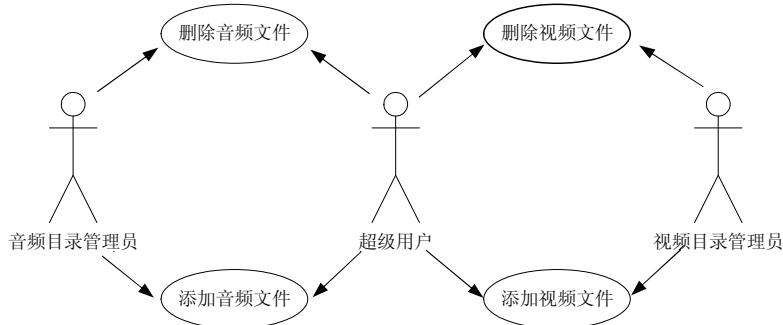


图 10.33 数据管理模块用例图

 说明：由于视频数据管理和音频数据管理实现的方法基本类似，因此，本章重点讲解视频数据管理模块实现的方法，关于音频数据管理模块的实现方法请参见本书附赠源码光盘。

10.11.2 数据管理模块技术分析

在本模块中，主要运用的就是文件上传技术。关于文件上传方面的知识，请参见第 8 章博客管理系统中的 8.7 节，里面有对预定义变量\$_FILES 的详尽介绍，和如何判断图片类型和大小的相关方法。

在这个模块中，略有不同的是使用了一个自定义函数来判断文件名后缀。下面来看看这个函数的实现过程：

例程 11 代码位置：光盘\TM\10\online\admin\inc\func.php

```

//判断文件后缀
//$f_type: 允许文件的后缀类型
//$f_upfiles: 上传文件名
function f_postfix($f_type,$f_upfiles){
    $is_pass = false;
    $tmp_upfiles = split(".", $f_upfiles);
    $tmp_fix = $tmp_upfiles[count($tmp_upfiles) - 1];
    for($num = 0; $num < count($f_type); $num++){
        if(strtolower($tmp_fix) == $f_type[$num])
            $is_pass = true;
    }
}
    
```

//设置一个boolean型变量\$is_pass
 //拆分上传文件名，存储到变量\$tmp_upfiles中
 //取得上传文件名后缀
 //使用for循环，输出允许使用的后缀类型
 //如果上传文件后缀包括在其中，说明类型相符
 //将\$is_pass变量赋值为true



```

        return $is_pass; //返回变量$is_pass
    }

```

代码贴士

① `split("\\.", $f_upfiles)`: `split()`函数为字符分割函数，它有两个参数：分隔符、要分割字符串。返回结果为分后后的字符串数组。例如，`$tmp=split("c","placepublicslice")`，那么`$tmp`值就为`$tmp[0] = "plac"`、`$tmp[1] = "epubli"`、`$tmp[3] = "sli"`、`$tmp[4] = "e"`。

② `count($tmp_upfiles)`: `count()`函数的作用是取得数组个数。

③ `strtolower($tmp_tix)`: `strtolower()`函数返回参数字符串的小写格式，如 `strtolower("AbC")`，输出等于 `abc`。

说明：尽可能地使用 PHP 函数库的函数。因为自己写的函数，无论是从算法设计，还是从功能实现上，都无法和 PHP 自带的函数相比较。不要试着去找寻它们的错误，这是非常不明智的做法。

10.11.3 数据添加的实现过程

■ 数据添加使用的数据表：tb_audio、tb_video

管理员可以通过单击视频数据管理页面的“数据添加”按钮打开数据添加页面（audio.php），在数据添加页面填写了表单之后，将表单提交给本页，本页会根据提交的表单数据在数据库中添加对应的记录。数据添加页面的运行结果如图 10.34 所示。

The screenshot shows a web page titled '打开电脑，看电影' (Open Computer, Watch Movie) from 'ONLINE 影视365' (www.online.com.cn). The main title is '视频数据添加' (Video Data Addition). The form fields include:

- 名称: 勇儿**自强 *
- 封面图片: E:\Documents and 浏览... *(最大700K)
- 资源上传: E:\Documents and 浏览... *(最大30M)
- 等级: 无限制级 *
- 发行商: XX影视娱乐公司 *
- 主要演员: 李某某 *
- 导演: 张XX *
- 制片人: 吴XX *
- 语言: 中文 英文 韩语
 日语 德语 法语 *
- 二级分类: 其他剧集 *
- 一级分类: 电影 *
- 发行国家: 中国 *
- 发行时间: 2008-01-01 *
- 新品: 是 否 *
- 简介: 立志片，很值得一看

At the bottom are '添加' (Add) and '返 回' (Back) buttons.

图 10.34 视频数据添加页面的运行结果

在视频数据添加页面中涉及到的 HTML 表单元素如表 10.4 所示。

表 10.4 视频添加页面涉及到的 HTML 表单元素

名称	类型	重要属性	含义
List	form	name="list" method="post" action="dataadd_chk.php" enctype="multipart/form-data"	表单
Names	text	id="names" size="30"	数据名称
Picture	text	id="picture" size="15"	图片名称
Address	text	id="address" size="15"	数据位置
Grade	select	<option value="一级">一级</option> <option value="二级">二级</option> <option value="三级">三级</option> <option value="无限制级" selected="selected">无限制级</option> <option value="禁片">禁片</option>	数据等级
Publisher	text	id="publisher" size="30"	发行商
Actor	text	id="actor" size="30"	主要演员
Director	text	id="director" size="30"	导演
Maker	text	id="marker" size="30"	制片人
Language	radio	<input type="radio" name="language" value="中文" checked> 中文 <input type="radio" name="language" value="英文">英文 <input type="radio" name="language" value="韩语">韩语 <input type="radio" name="language" value="日语">日语 <input type="radio" name="language" value="德语">德语 <input type="radio" name="language" value="法语"> 法语 *	语种
Style	select	<?php \$a_sqlstr="select * from tb_videoList where grade='2'"; \$a_rst = \$conn->execute(\$a_sqlstr); while(!\$a_rst->EOF){ ?> <option value="<?php echo \$a_rst->fields[2]; ?>"><?php echo \$a_rst->fields[2]; ?></option> <?php \$a_rst->movenext(); ?> ?>	类别
Types	select	<?php \$t_sql = "select * from tb_audiolist where grade = '1"'; \$t_rst = \$conn->execute(\$t_sql); while(!\$t_rst->EOF){ ?> <option value="<?php echo \$t_rst->fields[2]; ?>"><?php echo \$t_rst->fields[2]; ?></option> <?php \$t_rst->movenext(); ?> ?>	类型



续表

名称	类型	重要属性	含义
From	text	id="from" size="30"	发行国家
publishtime	text	loadCalendar(list.publishtime);	发行时间
news	radio	<input name="news" type="radio" value="1" checked>是 <input name="news" type="radio" value="0">否	新品参数
remark	textarea	class="submit" onclick="return a_chk();"	简要介绍
Submit	button	class="submit" onclick="return a_chk();"	“添加”按钮
Submit2	button	OnClick="javascript:top.window.close()>"	“返回”按钮

视频管理处理页（dataadd_chk）主要分3部分。图片文件格式及图片大小判断、视/音频格式及文件大小判断，信息都准确无误后，则向数据库中添加新记录。处理页处理的程序代码如下：

例程12 代码位置：光盘\TM\10\online\admin\dataadd_chk.php

```
<?php
    session_start(); //开启session支持
    include "inc/check.php"; //载入权限检查文件
    include "conn/conn.php"; //载入数据库链接文件
    include "inc/func.php"; //载入自定义函数文件
    $p_type = array("jpg", "jpeg", "bmp", "gif"); //允许上传的图片文件格式
    $f_type = array("avi", "rm", "rmvb", "wav", "mp3", "mpg"); //允许上传的视频、音频格式
    $audio_path = "./upfiles\\audio"; //视频上传位置
    $video_path = "./upfiles\\video"; //音频上传位置
    $picture_path = ""; //图片上传路径
    $file_path = ""; //文件上传路径
```

在图片类型判断中，首先判断图片大小是否在允许范围之内，接着使用自定义函数 f_postfix() 来判断图片后缀是否符合要求，如果没有问题，则根据 time() 函数，生成图片名，并保存到变量 \$picture_path 中。

例程13 代码位置：光盘\TM\10\online\admin\dataadd_chk.php

```
/* 判断图片大小，是否大于0k，小于700k */
① if($_FILES[picture][size] > 0 and $_FILES[picture][size] < 700000){
    /* 如果图片格式正确 */
    ② if(($postf = f_postfix($p_type,$_FILES[picture][name])) != false){
        /* 使用time()函数生成文件名 */
        ③ $picture_path = time().".".postf;
        /* 如果action值为a时，说明上传为视频图片 */
        if($_POST[action] == "a"){
            /* 如果生成的临时文件正确，则使用move_uploaded_file() 函数上传 */
            if($_FILES[picture][tmp_name])
                move_uploaded_file($_FILES[picture][tmp_name],$audio_path."\\".$picture_path);
            ④ else{
                echo "<script>alert('上传图片失败！');history.go(-1);</script>";
                exit();
            }
            /* 如果上传图片为音频图片 */
        } else if($_POST[action] == "v"){
    }
```

```

if($_FILES[picture][tmp_name])
    move_uploaded_file($_FILES[picture][tmp_name],$video_path."\\".$picture_path);
else{
    echo "<script>alert('上传图片失败！');history.go(-1);</script>";
    exit();
}
}
}else{
    echo "<script>alert('上传图片格式错误！');history.go(-1);</script>";
    exit();
}
/* 如果图片大于700KB,说明图片太大，超出要求，返回上一步 */
}else if($_FILES[picture][size] > 700000){
    echo "<script>alert('上传图片大小超出范围！');history.go(-1);</script>";
    exit();
}
else{
    $picture = "";
}
*****

```

代码贴士

- ① 使用预定义变量的\$_FILES[name][size]属性判断上传文件的大小。如果文件大于 pho.ini 中设置的上传文件的最大值，那么 size 将永远返回 0，而不是文件的实际大小。
- ② 使用自定义函数 f_postfix() 判断上传文件的后缀。
- ③ 返回当前 UNIX 的时间戳
- ④ 使用 move_uploaded_file() 上传函数要注意：它的第一个参数是上传到服务器中的临时文件名，而不是文件本身后，是用户自定义的名称。要取得临时文件名用 \$_FILES[name][tmp_name] 来获取。

对上传文件的判断和上传图片的流程基本相同，也是先判断文件大小是否符合上传文件的范围要求，接着判断上传文件后缀，最后使用 time() 函数生成文件名，保存到变量 \$address_path 中。下面是判断视频文件的流程。

例程 14 代码位置：光盘\TM\10\online\admin\dataadd_chk.php

```

/* 判断上传文件类型与大小 */
if($_FILES[address][size] > 0){                                //判断上传文件大小
    if($_POST[action] == "a"){                                    //判断是视频文件还是音频文件
        if($_FILES[address][size] < 300000000){                  //判断文件是否超出限定大小
            if(($postf = f_postfix($f_type,$_FILES[address][name])) != false){
                $file_path = time().".".postf;                    //生成新的文件名
                if($_FILES[address][tmp_name])                      //如果临时文件正确
                    /* 使用move_uploaded_file() 函数上传文件 */
                    move_uploaded_file($_FILES[address][tmp_name],$audio_path."\\".$file_path);
            }
        }
    }
}

```

```

        }else{
            echo "<script>alert('上传文件格式错误！');history.back(-1);</script>";
            exit();
        } }else{
            echo "<script>alert('上传文件大小错误！');history.go(-1);</script>";
            exit();
        }
    /* 如果是音频文件 */
    else if($_POST[action] == "v"){
        /* 判断文件大小 */
        if($_FILES[address][size] < 10000000){
            /* 判断文件后缀 */
            if(($postf = f_postfix($f_type,$_FILES[address][name])) != false){
                $file_path = time().".". $postf;
            /* 判断临时文件是否生成 */
                if($_FILES[address][tmp_name])
            /* 使用move_uploaded_file() 函数上传文件 */
                move_uploaded_file($_FILES[address][tmp_name],$video_path."\\".$file_path);
            else{
                echo "<script>alert('上传文件错误！');history.go(-1);</script>";
                exit();}}else{
                    echo "<script>alert('上传文件格式错误！');history.back(-1);</script>";
                    exit();}}else{
                        echo "<script>alert('上传文件大小错误！');history.go(-1);</script>";
                        exit();}}}
    }else{
        echo "<script>alert('没有上传文件或文件大于300M');history.go(-1);</script>";
        exit();
    }
}

```

如果上传图片和上传的文件都没有问题，则对表单的其他信息进行处理。生成 insert 插入语句，添加新记录：

例程15 代码位置：光盘\TM\10\online\admin\dataadd_chk.php

```

/* 视频文件和音频文件相同的信息 */
$names = $_POST[names];           //视频名称
$grade = $_POST[grade];           //级别
$sizes = $_FILES[address][size];   //大小
$publisher = $_POST[publisher];    //发行商
$actor = $_POST[actor];           //演员
$language = $_POST[language];      //语言
$style = $_POST[style];           //类别
$types = $_POST[types];           //类型
$froms = $_POST[from];           //发行国家
$publishtime = $_POST[publishtime]; //首映时间
$news = $_POST[news];             //是否为新片
$remark = $_POST[remark];         //备注

/***************/
if($_POST[action] == "a"){

```



```

/* 确认父级目录 */
/* 表单值，视频文件特有的属性 */
$director = $_POST[director]; //提取导演
$marker = $_POST[marker]; //制片人
/* 生成insert语句 */
$a_sqlstr = "insert into tb_audio
(name,picture,sizes,grade,publisher,actor,director,marker,languages,type,style,froms,publishtime,bool,remark,property,address,u
sername,issueDate)
values('$names','$picture_path','$sizes','$grade','$publisher','$actor','$director','$marker','$language','$types','$style','$from','$publ
ishtime','$news','$remark','管理员','$file_path','$SESSION[admin]','".date("Y-m-d H:i:s").")";
}
else if($_POST[action] == "v"){ //如果上传文件为音频文件
$actortype = $_POST[actortype]; //歌曲类型
$ci = $_POST[ci]; //作词
$qu = $_POST[qu]; //作曲
/* 生成insert语句 */
$a_sqlstr = "insert into tb_video
(name,picture,actor,ci,qu,actortype,type,style,publisher,froms,sizes,languages,publishTime,remark,property,address,u
serName,issueDate)
values('$names','$picture_path','$actor','$ci','$qu','$actortype','$types','$style','$publisher','$froms','$sizes','$language','$publishti
me','$remark','管理员','$file_path','$SESSION[admin]','".date("Y-m-d H:i:s").")";
}
else{
echo "<script>alert('错误');window.close();</script>";
exit();
}
/* 上传图片和文件 */
*****
$a_rst = $conn->execute($a_sqlstr);
if(!$a_rst == false)
echo "<script>top.opener.location.reload();alert('添加成功');window.close();</script>";
else
echo "<script>alert('添加失败');history.go(-1);</script>";
?>

```

10.11.4 数据删除的实现过程

 数据删除使用的数据表：tb_audio、tb_video

视频数据删除主要是根据传递的文件 id 值，将数据库中的记录删除，同时，使用 unlink 函数删除服务器端的真实数据。程序代码如下：

例程16 代码位置：光盘\TM\10\online\admin\del_video_chk.php

```

<?php
session_start(); //开启session支持
include "conn/conn.php"; //载入数据库链接文件
include "inc/chec.php"; //载入权限判断文件
$file_path = "..\upfiles\video\"; //生成文件路径
$s_sqlstr = "select * from tb_video where id = ".$_GET[id]; //生成SQL语句
$s_rst = $conn->execute($s_sqlstr);
if(!$s_rst == false){
/* 删除相应的图片文件和音频、视频文件 */

```

```

if(unlink($file_path.$s_rst->fields[16]) and unlink($file_path.$s_rst->fields[2])){
/* 生成删除语句 */
$d_sqlstr = "delete from tb_video where id = ".$_GET[id];
/* 执行删除语句 */
$d_rst = $conn->execute($d_sqlstr);
if(!$d_rst == false){
    echo "<script>alert('删除成功');location='main.php?action=video';</script>";
}else{
    echo "<script>alert('删除失败');history.go(-1);</script>";
}}else{
    echo "<script>alert('删除失败');history.go(-1);</script>";
}
?>

```

10.12 管理员设置模块设计

10.12.1 管理员设置模块概述

管理员设置模块主要包括管理员信息添加、管理员信息删除及管理员信息冻结 3 个部分。管理员可以通过后台管理功能导航进入管理员设置页面（member.php）。在管理员设置页面中，管理员可以进行管理员信息的添加操作，或进行会员信息的冻结或删除等操作。管理员设置用例图如图 10.35 所示。

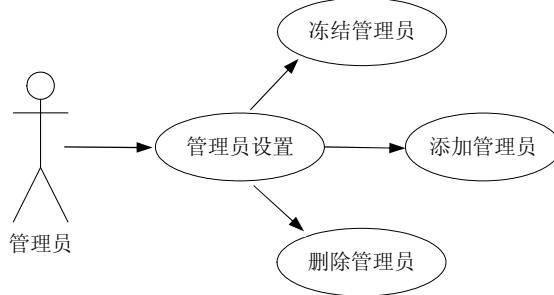


图 10.35 管理员管理用例图

10.12.2 管理员设置模块技术分析

管理员可以单击管理员设置页面的“冻结”或“解冻”按钮，对管理员信息进行冻结或解冻的操作。当管理员登录时，首先查看数据库中的 whether 字段，如果为 1，则说明该用户处于激活状态，否者，将显示提示窗口，告诉该用户，此账号已被冻结等提示信息。

对管理员信息的冻结或解冻操作主要是根据传递的管理员的 id，将数据库中的 whether 字段更新。程序代码如下：

例程17 代码位置：光盘\TM\10\online\admin\m_freeze_chk.php

```

<?php
    session_start(); //开启session支持

```



```

include "conn/conn.php"; //引入数据库链接文件
include "inc/chec.php"; //引入权限检查文件
/* 根据whether值，进行取反操作 */
if($_POST[whether] == "1")
    $wt = "0";
else if($_POST[whether] == "0")
    $wt = "1";
else{
    echo "<script>alert('非法操作!');history.go(-1);</script>";
    exit();
}
/* 根据管理员id和赋值后的$wt，生成update语句 */
$o_sqlstr = "update tb_manager set whether = '".$wt."' where id = ".$_POST[id];
/* 执行update语句 */
$o_rst = $conn->execute($o_sqlstr);
if(!$o_rst == false){
    echo "<script>alert('操作成功');location='main.php?action=manager';</script>";
}
?>

```

该方法实现很简单，但却十分实用，在使用会员制的网站上，在可以发帖、回复的论坛里，都可以使用这个方法来管理。

10.12.3 管理员添加的实现过程

管理员添加使用的数据表：tb_manager

管理员可以通过管理员设置页面打开管理员信息添加页面（addmanager.php）。在管理员信息添加页面中，管理员可以填写表单并将表单提交到处理页进行数据处理，以完成管理员信息添加的操作。管理员添加页面的运行结果如图 10.36 所示。



管理 员 信 息 添加	
管理员名称：	<input type="text" value="视频数据管理员"/>
管理密码：	<input type="password" value="*****"/>
密码确认：	<input type="password" value="*****"/>
管理权限：	视频数据管理员 <input type="button" value="▼"/>
真实姓名：	<input type="text" value="spcr"/>
<input type="button" value="添 加"/> <input type="button" value="返 回"/>	

图 10.36 管理员添加页面的运行结果

管理员添加页面涉及到的 HTML 表单元素如表 10.5 所示。

表 10.5 管理员添加页面涉及到的 HTML 表单元素

名称	类型	重要属性	含义
list	form	method="post" action="addmanager_chk.php"	表单
names	text	id="names" size="30"	管理员名称
password	password	id="password" size="30"	密码
password2	password	id="password2" size="30"	密码确认
grade	select	<option value="视频目录管理员" selected>视频目录管理员</option><option value="音频目录管理员">音频目录管理员</option><option value="视频数据管理员">视频数据管理员</option><option value="音频数据管理员">音频数据管理员</option><option value="会员数据管理员">会员数据管理员</option><option value="会员等级管理员">会员等级管理员</option>	管理权限
realname	text	id="realname" size="30"	真实姓名
Submit	button	class="submit" onClick="check();"	“添加”按钮
Submit2	button	value="返回" class="submit" onClick="javascript:top.window.close()"	“返回”按钮

管理员信息添加的数据处理主要是根据提交的表单数据，在数据库中添加对应的记录，首先判断输入姓名是否存在，如果存在，显示提示语，并返回到上一步。如果不存在这种错误，生成 insert 语句，添加新管理员。程序代码如下：

例程18 代码位置：光盘\TM\10\online\admin\addmanager.php

```

<?php
    session_start();
    //开启session支持
    include "conn/conn.php";
    //载入数据库链接文件
    include "inc/chec.php";
    //载入权限检查文件
    $a_sql="select * from tb_manager where name='".$POST[names]."';//检查是否有重名
    $a_rst = $conn->execute($a_sql);
    /* 如果管理员昵称存在，弹出提示框，并返回到上一步。 */
    if(!$a_rst->EOF)
        echo "<script>alert('该名称的管理员已经存在，请更换名称');history.go(-1);</script>";
    else{
        /* 生成insert语句 */
        $a_sqlstr="insert into tb_manager
values('".$POST[names]."', '".$POST[password]."', '".$POST[grade]."', '".$POST[realname]."', '".$date("Y-m-d")."', '1')";
        /* 执行insert语句 */
        $a_rst1 = $conn->execute($a_sqlstr);
        /* 如果管理员添加成功 */
        if(!($a_rst1 == false)){
            ?>
            <script>
            /* 刷新父窗口，弹出提示框，最后删除当前窗口 */
            top.opener.location.reload();
            alert("管理员添加成功");
            top.window.close();
            </script>
        }
    }
<?php
}

```

```

        else
            echo "<script>alert('添加失败'.".$_a_sqlstr.");history.go(-1);</script>";
    }
?>

```

10.12.4 管理员删除的实现过程

■ 管理员删除使用的数据表: tb_manager

管理员信息删除主要是根据传递的管理员的 id, 将数据库中对应的数据删除。程序代码如下:

例程19 代码位置: 光盘\TM\10\online\admin\del_mfreeze_chk.php

```

<?php
    session_start();                                //开启session支持
    include "conn/conn.php";                         //载入数据库链接文件
    include "inc/chec.php";                          //载入权限检查文件
    $d_sqlstr = "delete from tb_manager where id = ".$_GET[id]; //生成delete语句
    if(!($d_rst = $conn->execute($d_sqlstr)) == false) //判断删除成功
        echo "<script>alert('删除成功');location='main.php?action=manager';</script>";
    else
        echo "<script>alert('删除失败');history.go(-1);</script>";
?>

```

10.13 开发技巧与难点分析

在本系统中, 多个地方使用了无边框窗口和刷新父级窗口的技术。下面来学习一下有关这两方面的知识要点。

10.13.1 无边框窗口

平时看到的浏览器窗口都是有边框的, 如果偶尔有一个无边框的窗口弹出, 那么将会更加引人注目。实现无边框窗口使用的是 JavaScript 脚本中的 open 方法, open 方法的常用格式如下:

```
window.open(url,name,features,replace);
```

open 方法中的参数解释如表 10.6 所示。

表 10.6 open 方法的参数说明

属性 值	说 明
url	在弹出窗口中要打开的地址
name	要打开窗口的名字。在使用target属性时会用到, 可以为空
features	列举的窗口特征。如果不写任何参数, 那么默认的参数多数情况下是关闭的
replace	一个boolean值, 指出是否url替换当前内容。该参数可以省略不写

下面应用 window 对象的 open 方法打开一个指定大小的窗口。代码如下：

```
<script language="javascript">
    wopen = open("list.php","query","height=500,width=665,scrollbars=no,toolbar=yes,location=yes");
</script>
```

运行结果如图 10.37 所示。



图 10.37 使用 open 方法弹出窗口

除了上面实例中给出的几个窗口特征，表 10.7 还列出了其他几个常用的特征参数及说明。

表 10.7 浏览器窗口的外观样式

属性 值	说 明
left	新窗口的左坐标
top	新窗口的上坐标，配合 left 可以定位一个窗口的弹出位置
resizable	是否可以通过拖拽来调整新窗口的大小，默认为 no
status	在新窗口中是否显示状态栏，默认为 no

注意：窗口特征是用等号赋值、逗号分割的，在逗号或等号前后不要加上空格，因为在有些浏览器中会显示错误。

10.13.2 刷新父级窗口和关闭子窗口

在打开的子窗口进行一系列的操作，如添加、删除数据后，就要对当前窗口进行关闭，最后返回父级窗口中，但父级窗口中的数据可能仍然是没有添加或删除之前的记录数，这时，就可以调用 JavaScript 脚本，对父窗口进行刷新了。这种对父级窗口的刷新只有对弹出窗口才有效，使用时要注意。该脚本的使用格式为：

```
top.opener.location.reload();
```

opener 属性是子窗口的最高层 window 对象才有的属性，配合 location 对象可以对父窗口进行刷新或重载。location 对象的两个常用方法有 href 方法和 reload 方法。href 方法可以刷新父窗口内容，使用格式如下：

```
top.opener.location.href="operation.php";
```

关闭子窗口使用如下代码即可：

```
window.close();
```

不过这里要注意的是：close()方法在关闭子窗口时，不弹出提示就直接关掉浏览器了，但对于父级窗口，则弹出提示框，经确认后才可以关闭浏览器。

10.14 ADODB 连接数据库技术专题

PHP 为各种版本的数据库定义了庞大的函数群来支持，而且支持的很完整，那为什么本系统要使用 ADODB 来对数据库进行操作呢？原因在于 ADODB 的优势：一次编写，不需修改。也就是说，当系统使用的数据库需要更换时，只要更新数据库、数据表即可，不需要再重新编写源码了，这为后期维护节约了一大笔开销，而且不需要长时间关闭网站，这是 PHP 函数所无法比拟的。本章的技术专题就来着重学习一下 ADODB 技术。

10.14.1 ADODB 简介

由於 PHP 存取函数没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时，会带来大量的代码修复工作。这时，就需要一组函数库来隐藏不同资料库函数界面间的差异，让开发者可以很简单地去切换资料库，这就是 ADODB。ADODB 目前支持 MySQL、Oracle、Microsoft SQL Server、Sybase、PostgreSQL、Foxpro、Access、ADO 和 ODBC 等大多数知名的、不知名的数据库类型。

10.14.2 ADODB 的安装

要使用 ADODB 来操作数据库，首先就要获取和安装 ADODB。读者只要到网上下载 ADODB 包，解压到 Web 服务器目录下即可。

 注意：要使用 ADODB，使用的 PHP 必须是 4.01 以上的版本。

10.14.3 ADODB 常用函数及说明

在 10.4.2 节中已经介绍过了使用 ADODB 的一般步骤，在实际使用中，需要根据情况进行不同的处理，这时就需要对步骤及功能进行扩展，下面就来介绍几种经常会用到的函数。

1. PConnect(\$host,\$user,\$password, \$database)

- \$host: 数据库服务器地址。
- \$user: 数据库用户名。
- \$password: 数据库密码。



- \$database: 连接的数据库。

如果连接成功回传 true，失败则回传 false。

2. SelectLimit(\$sql,\$numrows=-1,\$offset=-1)

- \$sql: 执行的 sql 语句。

- \$numrows: 显示的记录条数。

- \$offset: 从第几条记录开始计算。

这个函数和 MySQL 里的 limit 子句相似，但要注意，limit 后面的两个参数的顺序是 (limit offset,numrows)，这个函数是完全相反的。

3. ErrorMsg()

回传错误信息。当 execute() 函数出现错误时，使用 ErrorMsg() 函数可以查看最后出错的相关信息。使用方法如下：

```
...
$recordset = $conn->Execute("select * from tb_audio")
if(!$recordset) //判断execute()函数执行中是否有错误
    echo $conn->ErrorMsg();
else{
    ...
}
```

即使没有出现错误，ErrorMsg() 也会回传最后的状态值，这种情况下一般不需要输出。

4. RecordCount()

回传结果集里的记录数。使用方法如下：

```
...
$sql = 'select * from tb_account';
$rs = $conn->Execute($sql); //执行SQL语句
echo $rs->RecordCount(); //输出查询结果数
...
```

5. FieldCount()

回传结果集里的字段数。使用方法同上。

6. MoveNext()、Move()、MoveFirst()、MoveLast()

- MoveNext(): 移动指针到下一条记录。

Move(\$to): 移动指针到指定的列 (\$to)。这里使用的是绝对定位，如 \$to 是 10，就移动到结果集中的第 10 条记录，如果 \$to 大于记录总数，指针将移动到最后。注意：部分数据库不支持想后移动指针。

- MoveFirst(): 实际上就是 Move(0)，指针移动到结果集头部，部分数据库不支持该函数。

MoveLast(): 指针移动到结果集的尾部，也可以表示为 Move(\$recordCount() - 1)，部分数据库不支持此函数。

最后介绍一个 ADODB 中的常量。



7. \$ADODB_FETCH_MODE

该常量决定了结果集以哪种方式将数据回传。在有些数据库中，使用\$recordset->fields[]仅支持字段编号为索引，即 0、1、2 等等，不支持字段名（如 id, user）做索引，这时，可以使用该常量进行设置，该常量的值有：

define('ADODB_FETCH_DEFAULT',0);	//系统默认值
define('ADODB_FETCH_NUM',1);	//以字段编号为索引
define('ADODB_FETCH_ASSOC',2);	//以字段名称为索引
define('ADODB_FETCH_BOTH',3);	//同时用时编号和名称

这时上面的步骤可以改为：

```
...
$conn = &ADONewConnection("mysq");
$conn->PConnect("db_online");
$ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;
$recordset = $conn->Execute("select * from tb_audio");
if(!$recordset) //判断execute()函数执行中是否有错误
    echo $conn->ErrorMsg();
else{
    while(!$recordset->EOF){ //如果没有错误，则配合while语句循环输出结果
        echo $recordset->fields[id].<br>;
        $recordset->moveNext(); //指针下移
    }
}
...
...
```

这里建议将\$ADODB_FIELDS_MODED 的值固定为 ADODB_FETCH_NUM 及 ADODB_FETCH_ASSOC，不建议使用 ADODB_FETCH_BOTH，因为许多驱动程序并不支持该常量。

ADODB 的常用函数就介绍这么多了，如果读者感兴趣，可以下载 ADODB 手册来学习更多内容。

10.14.4 不同数据库之间的转换

使用 ADODB 的优势就是可以在以后的升级中，更换不同的数据库而不用大量的改动源程序。但不同数据库之间，有的 SQL 语句或语法是不相同的，这时就要做一些小幅的调整。

(1) 在 conn.php 文件中，更改连接的数据库类型和参数。

```
$conn = &ADONewConnection('mysql'); //建立MySQL连接
$conn->PConnect("localhost","root","root","db_online"); //连接"db_online"数据库
```

如使用 SQL Server 时，上面的链接语句更改为：

```
$conn = &ADONewConnection('mssql');
$conn->PConnect("localhost","sa","","db_online");
```

(2) 注意包含时间类型的 SQL 语句。

不同的数据库对于时间的表示法是不同的，如 MySQL 使用时间类型时，就像使用字符类型一样，



放到单引号里就可以了，但 Access 则需要放到“#”之间。ORACLE 虽然也是使用单引号，但它的格式为“日-月-年”，而其他数据库大多都是“年-月-日”。

(3) 尽量使用通用的 SQL 语句。

尽管国际标准化组织 (ISO) 在 1987 年创建了 SQL 的一个标准，但在各种 SQL 之间仍然存在差别。所以为了增加通用性，最好了解一下数据库间的不同。例如，MySQL 中的 limit 子句，在 oracle 中就不被支持。

10.15 本章总结

online 影视 365 网实现了一个在线播放平台。该网站具备一个影视网站所需要的所有的基本功能，包括影视文件的上传、下载、播放及介绍。还学习了 js 脚本语言的一些实用技巧，最后，介绍了本章的核心内容：ADODB 连接数据库技术。希望通过本章的学习，读者不仅可以建立一个自己的影视网，而且可以熟练使用 ADODB 来连接任意一个数据库。有条件的读者可以试着将本系统的数据库改为 SQL Server 或 ORACLE，来增加对 ADODB 的了解及熟练程度。

