

第 1 章 项目背景

一、概述

本章介绍贯穿教材的 Web 应用系统——视频信息管理系统的构思和设计,包括系统的技术路线的选择、系统的需求分析、系统的页面设计及数据库设计等内容。

通过本章的学习,读者可以了解视频信息管理系统中要完成的任务,了解基于 PHP 的 Web 应用开发的整个流程。

二、重点与难点

重点:

- (1)对视频信息管理系统进行需求分析。
- (2)了解 Web 应用开发所需组件。

难点:

了解 Web 应用开发的整个过程。

1.1 网站开发的体系结构

基于 Web 的应用系统开发可以采用两种体系结构,一种是 C/S 结构,另外一种 B/S 结构。

(1)C/S 结构

C/S 结构,即 Client/Server(客户机/服务器)结构,通过将任务合理分配到 Client 端和 Server 端,降低了系统的通信开销,可以充分利用两端硬件环境的优势。应用此种结构需要为客户端和服务器分别编写不同的软件。比如,常用的通信软件 QQ,用户在使用 QQ 的时候需要下载客户端的 QQ 程序安装到自己的电脑上,然后通过这个客户端程序和腾讯的服务器交换数据。

(2)B/S 结构

B/S 结构,即 Browser/Server(浏览器/服务器)结构,是随着 Internet 技术的兴起,对 C/S 结构的一种变化或者改进的结构。在这种结构下,用户界面完全通过 WWW 浏览器实现,极少部分业务逻辑在前端(Browser)实现,但是主要业务逻辑在服务器端(Server)实现,这样就大大简化了客户端电脑的负荷,减轻了系统维护与升级成本和工作量,降低了用户的总

体成本。

用户通过浏览器查看网页,网页(包括静态网页、动态网页)存放在 Web 服务器上。用户通过 URL 访问服务器上的网页,服务器接到请求,通过 HTTP 的方法将网页传送给客户机,本地的浏览器将网页代码解释为一种美观、直观的形式,展现在用户面前。文字与图片是构成网页的最基本的元素,网页中还可以包括 Flash 动画、音乐、流媒体等。

一般来说,Web 服务器是一台或多台性能比较高的计算机,上面安装有 WWW 服务器软件,硬件和软件相结合,通过网络向用户提供服务。

当用户通过浏览器点击网页上的一个链接,或者在地址栏中输入一个网址的时候,其实是对 Web 服务器提出了访问请求,Web 服务器经过确认,会直接把用户请求的 HTML 文件传回给浏览器,浏览器对传回的 HTML 代码进行解释,这样用户就会在浏览器中看到所请求的页面,这个过程就是 HTML 页面的执行过程,如图 1.1 所示。

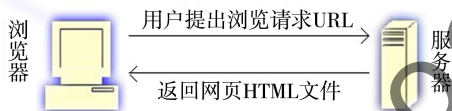


图 1.1 用户访问网页过程

总的来说,B/S 结构相对于传统的 C/S 结构,其优点在于:

①B/S 结构是一种瘦客户端模式,客户端软件只需安装浏览器,且对客户端硬件配置要求较低。

②标准统一,维护相对简单。HTML 是 Web 信息的组织方式,所有 Web 服务器和浏览器都遵循这个国际标准。开发人员可以集中在服务器端开发和维护应用程序,而服务器上的应用程序可通过网络浏览器在客户端上执行,从而充分发挥开发人员的群体优势,应用程序的维护也相对简单。

③无须开发客户端软件。浏览器软件可以从 Internet 上免费得到。

④跨平台支持。由于采用统一的通信协议,并且浏览器及服务器软件可以支持多平台,可以方便企业异构平台运行。

⑤浏览器界面易学易用。

本教材的定位就是开发一个 B/S 结构的 Web 系统。

1.2 网站开发中常用概念介绍

(1) WWW

WWW 是 World Wide Web 的缩写,也可以简称为 Web,3W 等,中文名字为“万维网”。它是存储在 Internet 计算机中数量巨大的文档的集合。这些文档彼此关联,内含文本、图形、视频、音频等,通过超文本标记语言(HTML)把这些信息组织在一起,称为超文本,再利用超级链接从一个站点跳到其他站点,使信息获取变得更加便利。

(2) URL

URL(Uniform Resource Locator,统一资源定位符)是一种地址,指定协议(如 HTTP

或 FTP)以及对象、文档、WWW 网页或其他目标在 Internet 或 Intranet 上的位置,例如:
`http://www.microsoft.com/`。

每家每户有一个门牌地址,每个网页也有一个 URL。在浏览器的地址框中输入一个 URL 或是单击一个超级链接时,就确定了要浏览的地址。

URL 有以下几种常见形式:

`ftp://219.216.128.15/`

`http://baike.baidu.com/view/8972.htm`

`http://bbs.runsky.com/bbs/forumdisplay.php?fid=38`

(3) HTTP

Internet 的基本协议是 TCP/IP 协议,然而在 TCP/IP 模型最上层的是应用层,它包含所有高层的协议。高层协议有:文件传输协议 FTP、电子邮件传输协议 SMTP 和 HTTP 协议等。

HTTP 协议(Hyper Text Transfer Protocol,超文本传输协议)是用于从 WWW 服务器传输超文本到本地浏览器的传送协议,它保证计算机正确快速地在网络上传输超文本文档。

HTTP 就是在 Internet 上传输网页的协议,它可以屏蔽掉传输的细节,对用户是透明的,网页编写者只要将精力集中在网页设计与制作上就可以了。

(4) 静态网页与动态网页

静态网页就是纯粹的 HTML 页面,网页的内容是固定的、不变的。网页一经编写完成,网页的显示效果就确定了。

动态网页是内容根据具体情况变化的网页,它一般根据网页的输入参数和数据库中内容的变化而变化。

如果在某位用户登录后,要出现一个网页,显示“你好,用户”。即张三登录后可以看到一个网页显示,“你好,张三”,而李四登录后见到的内容是“你好,李四”。如果要满足上面的要求,需要做两个静态页面,但如果有一万个用户、10 万个用户的时候,显然不可能提前做好那么多的页面,这就需要应用动态页面技术来实现这样的功能。

常用的动态网页技术有 JSP、ASP、PHP、CGI 等。可以从文件的扩展名来看一个网页文件是动态网页还是静态网页。静态网页的 URL 后缀是 htm、html、shtml、xml 等;动态网页的 URL 的后缀是 asp、aspx、jsp、php、perl、cgi 等。

(5) 网站前台和后台

网站开发的前台和后台有两种解释。第一种是从开发网站分工角度来划分,前台指的是完成网站美工、布局设计、UI 元素设计等;后台指的是负责业务逻辑处理及对数据库的操作。第二种是从用户角度来划分,前台指的是网站的普通用户可见的功能和界面;后台指的是后台管理员对数据的管理功能。本教材中提到的网站前台和后台按照第二种解释来理解。

(6) 请求

请求是由用户发起的,可以通过浏览器地址栏的形式发起,也可以通过点击超链接的方式发起,还可以通过提交表单的方式发起。在网页中请求的方式有两种,一种是 get 方式,一种是 post 方式。get 方式提交的请求数据会附加在浏览器地址栏中,比如在搜索功能、收

藏功能或者购物网站中常用此方式;post 方式数据以请求包的形式发送,当表单里有密码(登录、注册)、表单提交的数据特别大(比如发表的文章)、有文件上传功能的时候就必须要使用 post 方式。

1.3 动态网站开发所需的 Web 构件

动态网站的开发需要多种开发技术结合在一起使用,每种技术的功能各自独立而又要相互配合才能完成一个动态网站的建立,其中每一种技术称为一个 Web 构件。一个动态网站开发过程中必须用到的 Web 构件包括:客户端 IE/Firefox/Chrome 等多种浏览器,超文本标记语言 HTML,层叠样式表 CSS,客户端脚本编程语言 JavaScript/VBScript/Applet 等其中的一种,Web 服务器 Apache/Tomcat/IIS/Nginx 等其中的一种,服务器端编程语言 PHP/JSP/ASP 等其中的一种,数据库管理系统 MySQL/Oracle/SQL Server/DB2 等其中的一种。

1. 客户端浏览器

客户端浏览器是用户在客户端浏览网页时使用的软件。因为浏览器能够解析 HTML、CSS 和 JavaScript 等语言来显示网页,所以是 Web 开发中不可缺少的构件之一。对 B/S 结构的软件来说,浏览器就相当于用户端的操作界面,只要在浏览器输入不同的地址,访问不同的 Web 服务器,就形成了不同的用户操作界面。用户计算机默认都已经安装好了浏览器,所以这种图形用户界面不仅不用安装专用的客户端软件,而且只要在 Web 服务器上有一些改变,所有访问这个 Web 服务器的客户端界面通过刷新就会实时更新界面。Web 服务器还可以根据用户不同的请求,为用户返回定制的界面。所以动态网站都是通过浏览器中的图形用户界面来实现与 Web 服务器和数据库交互。但是不同的浏览器对网页的解析可能产生不同的界面效果,因此在开发 Web 应用时通常都需要用多种浏览器来进行测试。

2. 超文本标记语言 HTML

HTML(Hyper Text Mark-up Language)即超级文本标记语言或超文本链接标记语言,是构成网页文档的主要语言,也是一种规范、一种标准,是网站软件开发必不可少的 Web 构件之一。每一个 Web 开发者都需要对它熟练掌握。

HTML 文档是一个放置了标记(tags)的 ASCII 文本文件,通常它的文件扩展名是 .html 或 .htm。生成一个 HTML 文档主要有以下三种途径:第一种,手工直接编写(例如:文本编辑器记事本或其他 HTML 的编辑工具 Dreamweaver、Editplus 等);第二种,通过某些格式转换工具将现有的其他格式文档(例如:Word 文档)转换为 HTML 文档;第三种,由 Web 服务器在用户访问时动态地生成。

HTML 语言是通过利用各种“标记”来标识文档的结构和超链接、图片、文字、段落、表单等信息,再通过浏览器读取 HTML 文档中这些不同的标签来显示页面,形成用户的操作界面。虽然 HTML 语言描述了文档的结构格式,但不能精确地定义文档信息必须如何显示和排列,所以用户端最终的界面效果取决于 Web 浏览器本身的显示风格及其对标记的解释能力。这就是为什么同一文档在不同的浏览器中展示的效果会不一样。

图 1.2 是一段简单的 HTML 代码运行在浏览器中所呈现的网页效果。

Document Heading

This is the first paragraph. [HTML](#) 不是一种编程语言，而是一种超文本标记语言。标记语言是使用一套标记标签描述网页。

This is the second paragraph.

图 1.2 网页效果

代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Document</title>
</head>
<body>
  <h1>Document Heading</h1>
  <p>This is the first paragraph.
  <a href="http://www.w3.org/html/">HTML</a>
  不是一种编程语言，
  而是一种超文本标记语言。
  标记语言是使用一套标记标签描述网页。
  </p>
  <p>This is the second paragraph.</p>
</body>
</html>
```

3. 层叠样式表 CSS

层叠样式表 CSS(Cascading Style Sheets)也称级联样式表，是一种为网站添加布局效果的出色工具，可定义 HTML 元素如何被显示，可以有效地对页面进行布局 and 美化，设置字体、颜色、背景和其他效果等来实现更加精确的样式控制。CSS 不能离开 HTML 独立工作。

CSS 是由 W3C 的 CSS 工作组创建和维护的。和 HTML 一样，也算是一种标记语言，因此也不需要编译，还是直接由浏览器解释执行的。所以在不同的浏览器中展示的效果也会不一样，开发者同样要遵守 W3C 制定的标准。

CSS 包含了一些 CSS 标记，可以直接在 HTML 文件中使用，也可以写到后缀名是 .css 的文本文件中。只要对相应的代码做一些简单的修改，就可以改变同一页面的不同部分，或者改变网页的整体表现形式，或者改变多个不同页面的外观和布局。

如图 1.3 所示是给图 1.2 中的网页增加了 CSS 样式后的效果。

Document Heading

This is the first paragraph. HTML 不是一种编程语言，而是一种超文本标记语言。标记语言是使用一套标记标签描述网页。

This is the second paragraph.class="ex"

图 1.3 加 CSS 样式后网页效果

添加了 CSS 代码的 HTML 文档如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Document</title>
  <style type="text/css">
    h1 {color: #00ff00; text-decoration: underline;text-align: center;}
    body{color: red;}
    p{text-indent: 30px;}
    p.ex{color: rgb(0,0,255);font:italic bold 15px/30px Times;}
    a{font-family: Times; text-decoration: none;}
  </style>
</head>
<body>
  <h1>Document Heading</h1>
  <p>This is the first paragraph.
  <a href="http://www.w3.org/html/">HTML</a>
  不是一种编程语言，
  ◆ 而是一种超文本标记语言。
  标记语言是使用一套标记标签描述网页。
  </p>
  <p class="ex">This is the second paragraph.class="ex" </p>
</body>
</html>
```

4. 客户端脚本编程语言 JavaScript

HTML 用来在页面中显示数据，CSS 用来对页面进行布局与美化，而客户端脚本语言 JavaScript 则是一种有关因特网浏览器行为的编程，是用来编写网页的功能特效的，能够实现用户和浏览器之间的互动性，这样才能有能力传递更多的动态网站内容。客户端脚本编程语言有许多种，如 JavaScript、VBScript、JScript、Applet 等，都可以开发同样的交互式 Web 网页。而 Web 开发中使用最多、浏览器支持最好、案例最丰富的是 JavaScript 脚本语

言,并且像 Ajax 和 jQuery 框架等技术也都是基于 JavaScript 开发的产品。

JavaScript 是为网页设计者提供的一种编程语言,可以在 HTML 页面中放入动态的文本,能够对事件进行反应(如鼠标单击移动等事件操作),可读取并修改 HTML 元素、元素属性和元素中的内容,并被用来验证数据。

CSS 样式表和客户端脚本语言结合使用,能够使 HTML 文档与用户具有交互性和动态变换性,通常称为 DHTML(Dynamic HTML,动态 HTML)。样式表和客户端脚本都是直接由浏览器解释执行的,所以同一文档在不同的浏览器中展示的效果也会不一样,编写 JavaScript 代码时也要遵循 W3C 标准。JavaScript 程序可以直接嵌入到 HTML 页面文档中,也可以写在一个后缀名为 .js 的文本文件中,之后再链接到 HTML 页面。

如下面代码所示,是使用 JavaScript 脚本代码对某一用户注册页面中的用户名是否为空进行验证。如果用户在提交页面表单内容时未填写用户名,则将出现如图 1.4 所示的提示对话框。

```
<script>
function check()
{
    var uname=document.fm.username.value;
    if(uname==null || uname=="")
    {
        alert("用户名不能为空!");
        return false;
    }
    return true;
}
</script>
```

在上述的代码中, fm 是表单的名字, form 中需要添加 name 属性,取值为 fm,同时还要添加 onSubmit 事件。当点击提交按钮时,将执行函数 check()。如果 check() 函数返回 false, 表单不会提交到处理页面;如果 check() 函数返回 true, 表单才能提交给处理页面。



图 1.4 用户名输入空值时的验证

5. Web 服务器

Web 服务器的主要功能是提供网上信息浏览服务,传送网页给浏览器浏览。当用户在客户端使用浏览器,并通过 URL 请求 Web 服务器管理下的 HTML 文件时,Web 服务器软件则会在它有权管理的目录中,寻找用户请求的 HTML 网页文件。如果用户请求的文件存在,则直接把网页中的内容代码响应给客户端请求的浏览器。如果用户请求动态页面,

Web 服务器把动态响应的产生委托给一些其他的程序(例如 PHP 脚本)的服务器端技术。浏览器在收到服务器返回的代码后,逐条解释成网页,显示给用户查看。Web 服务器的功能展示如图 1.5 所示。

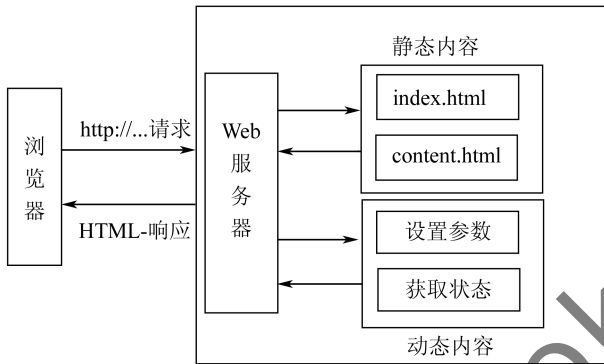


图 1.5 Web 服务器的功能展示

所谓“发送服务请求”指的是客户端想要得到某个服务(例如,想要浏览网页),而向服务器发送请求,服务器得到请求后,就会将请求的结果返回给客户端,这样就构成了一个完整的流程。服务器不知疲倦地工作,不停地响应来自任何地方的不同服务请求,在权限允许的情况下将数据源源不断地发送出去。

在 Internet 中,Web 服务器和浏览器通常位于两台不同的机器上,也许它们之间相隔千万里。但在开发阶段,一般都会将个人计算机作为 Web 服务器,即本地服务器。访问远程或本地的 Web 服务器没有什么差别,其工作原理是不变的。目前可用的 Web 服务器有 Apache、NGINX、IIS、Tomcat、Weblogic 等。在 WAMP 中使用的 Apache 服务器,它是世界上使用排名第一的 Web 服务器,它可以运行在几乎所有广泛使用的计算机平台上。它是开源软件,不断有人来为它开发新的功能、新的特性,修改原来的缺陷。Apache 的特点是简单,速度快,性能稳定。

6. 服务器端编程语言

服务器端编程语言是提供访问商业逻辑的途径以供客户端应用的程序,是需要通过安装应用服务器解析的,而应用服务器又是 Web 服务器的一个功能模块,需要和 Web 服务器安装在同一个系统中。所以服务器端编程语言的使用是用来协助 Web 服务器工作的编程语言,也可以说是对 Web 服务器功能的扩展,并外挂到 Web 服务器上一起工作。

如果用户向服务器请求的是一个脚本程序,如 .php 文件或者 .jsp 文件或者 .asp 文件,服务器会调用相应的引擎把 PHP 等代码转换成模板代码(html/css/javascript),再将结果返回给用户。因为 Web 服务器本身是不能解析这个脚本程序的,那么服务器除了要安装 Web 服务器 Apache 之外,还要安装可以解析脚本程序的应用程序服务器软件(例如 PHP 应用服务器),并在 Apache 服务器中配置来自客户端的 PHP 文件的请求,就可以在服务器端使用 PHP 应用服务器来解析 PHP 程序了。因为 PHP 应用服务器会理解并解释 PHP 代码的含义,这样就可以根据用户的不同请求进行操作,也就是通过 PHP 程序的动态处理,解

释成不同的 HTML 静态代码响应给用户。当然返回给客户端浏览器的只是一个很单纯的静态 HTML 网页。动态网站在用户端是看不到 PHP 程序源代码的,在一定程度上起到了保护代码的作用。

7. 数据库管理系统

现在的动态网站都是基于数据库的编程,任何程序的业务逻辑实质上都是对数据的处理操作。如果网站的数据是保存在服务器端的数据库中,则需要为服务器安装数据库管理系统(如 MySQL),用来存储和管理网站中的内存数据。MySQL 服务器和 Apache 服务器可以安装在同一台计算机上,也可以分开来安装,通过网络相连即可。由于 Apache 服务器是无法连接或者操作 MySQL 服务器的,所以也要安装 PHP 应用服务器。这样 Apache 服务器就可以委托 PHP 应用服务器,通过解释 PHP 脚本程序去连接或者操作数据库,完成用户的请求。数据库的功能和 Web 操作形式如图 1.6 所示。

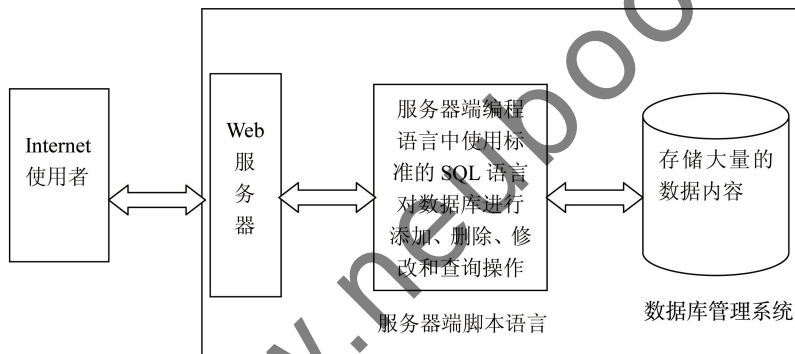


图 1.6 数据库的功能和 Web 操作形式

数据库管理系统这种软件也有好多种,都是使用标准的 SQL 访问和处理数据库中的数据。例如 Oracle、DB2、SQL Server、MySQL、Sybase、SQLite、Access 等。本教材主要介绍 MySQL 数据库管理系统。MySQL 是一个关系型数据库,是一个真正多用户、多线程的关系数据库服务器,和 PHP 一样都是开源免费的软件。主要特点是执行效率与稳定性高,操作简单、易用,所以用户众多,同时也提供网页形式的操作 PHPMyAdmin 管理界面和多种图形管理界面,简单易学,管理方便。MySQL 和 PHP 是真正的黄金组合,是网站开发首选的数据库管理系统。

1.4 视频信息管理系统的构思与设计

本课程以“视频信息管理系统”作为课程内容讲解的主线,系统功能完善,页面美观,界面友好,易于激发读者的学习兴趣。视频信息管理系统为读者呈现了基于 PHP 技术的 Web 应用从设计、开发到部署的各个环节,使读者通过实际项目的应用更好地进行开发实践,极具学习价值。

1.4.1 需求说明

本教材中介绍的视频信息管理系统分为普通用户、注册用户和管理员三类用户,每类用户要求实现如下功能:

普通用户的功能包括浏览首页信息、分栏目浏览视频信息、全站搜索视频信息并分页浏览、浏览视频详细信息、在线播放视频、下载视频、登录、注册的功能。普通用户功能模块结构图如图 1.7 所示。

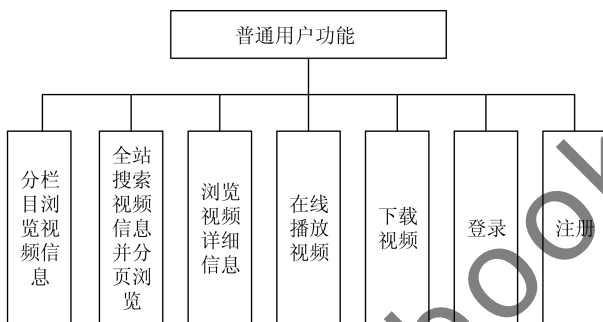


图 1.7 普通用户功能模块结构图

注册用户的功能除了具有普通用户的功能以外,还可以实现给视频评分、给视频留言、收藏视频、修改个人密码、修改个人信息、修改历史评价记录以及注销的功能。注册用户功能模块结构图如图 1.8 所示。

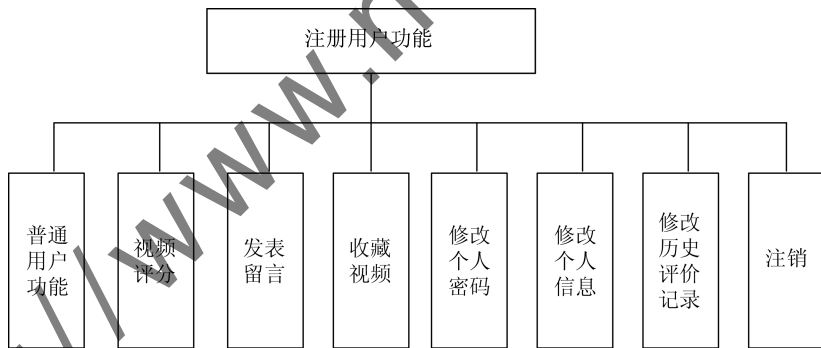


图 1.8 注册用户功能模块结构图

管理员模块包括管理员欢迎界面、视频类别管理、视频信息管理、注册用户信息管理、留言管理、修改个人密码以及管理员登录、管理员退出等功能。其中视频类别管理包括视频类别浏览,视频类别添加、修改和删除;视频信息管理包括视频信息分页浏览,视频信息添加、修改、删除和搜索;注册用户管理包括注册用户信息分页浏览、注册用户信息修改、注册用户信息删除;留言管理包括分页浏览用户发表过的留言、删除留言记录的功能。管理员功能模块结构图如图 1.9 所示。

1.4.2 界面设计

视频管理系统作为大众使用的 Web 应用系统,其界面风格应该遵循清晰美观、用户体验良好的原则。系统从首页开始,通过点击不同的超级链接、按钮或者模态框转入其他页

面,各个页面之间的迁移关系如图 1.10 和图 1.11 所示。

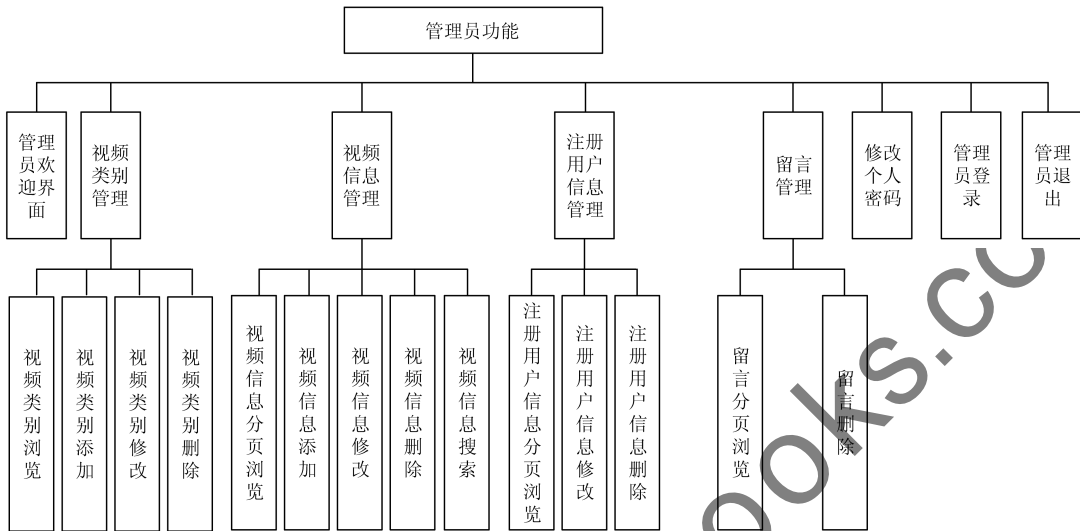


图 1.9 管理员功能模块结构图

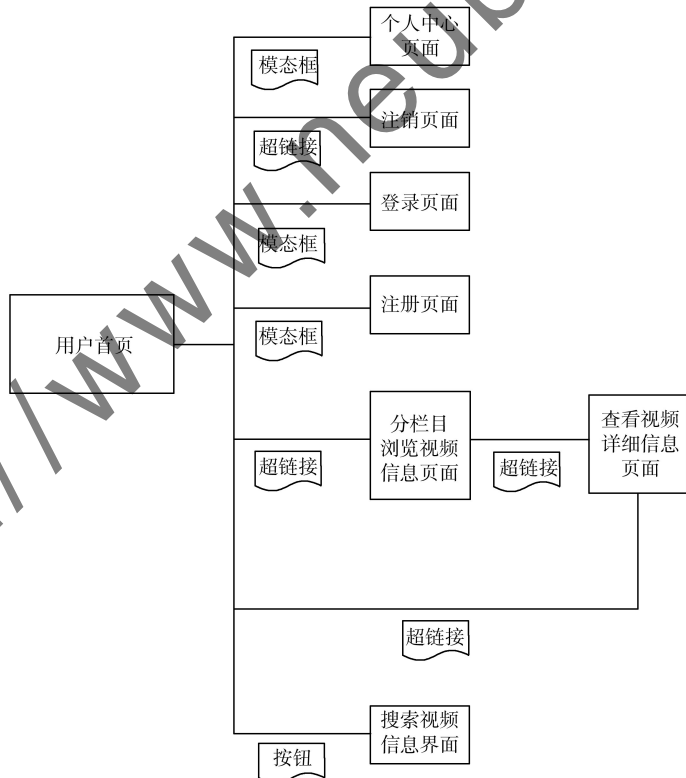


图 1.10 用户功能的页面迁移图

下面将给出网站几个核心功能的运行界面截图。视频信息管理系统的首页显示导航、幻灯片、主要栏目的视频列表和排行等信息,首页中还提供了登录、注册和搜索的超链接,如图 1.12~图 1.15 所示。

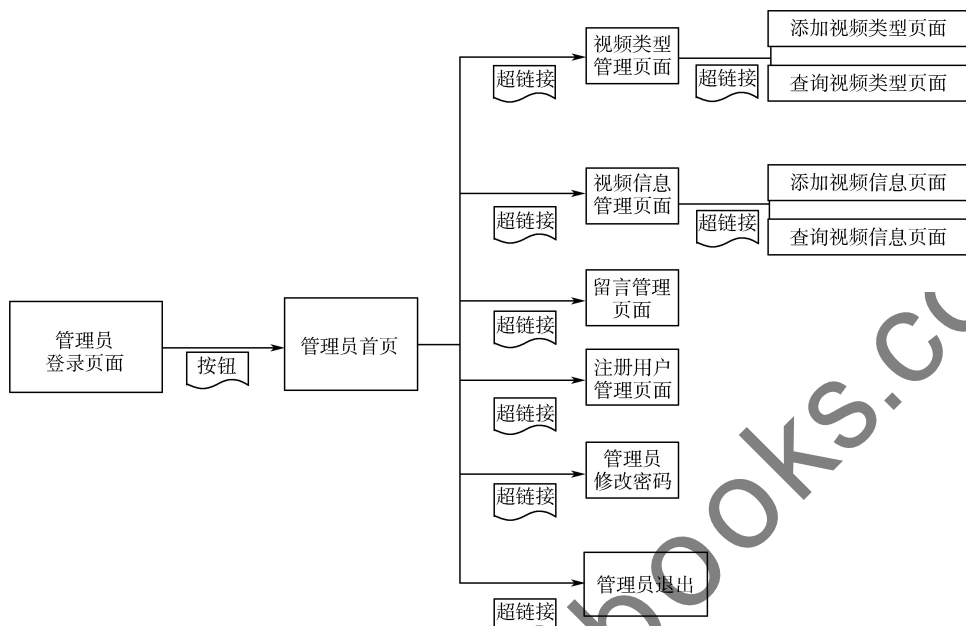


图 1.11 管理员功能的页面迁移图



图 1.12 首页-导航



图 1.13 首页-滚动幻灯片



图 1.14 首页-主要栏目列表



图 1.15 首页-点击排行和下载排行

如果用户成功登录后点击视频标题进入视频内容页(此处以原创栏目的“操作系统”课程视频为例来展示内容页的详细信息),如图 1.16~图 1.19 所示。在此页面用户可以了解该视频的专栏信息、点击次数、下载次数、上传时间以及上传人等详细信息,还可以下载影片,为视频发表留言、收藏以及评分等。



专栏	原创
上传时间	2015-12-08 14:28:26
点击次数	35
上传者	zhouhaibo
下载次数	2
有事找站长	意见箱
下载和收藏	点击下载 点击这里收藏
评分	<input type="button" value="评价影片"/> <input type="button" value="评价"/>

计算机操作系统

影片评分: 5.0

图 1.16 视频内容页-信息介绍

内容简介

操作系统是计算机系统的基本组成部分,是整个计算机的基础和核心。它对下操纵硬件的动作,控制各种资源的分配和使用。扩充硬件的功能,对上为用户程序和其他软件、工具提供环境和服务,方便用户使用。由于操作系统处于这样一种特别重要的地位,因而,操作系统课程就是各大院校设计计算机软件和计算机应用专业的一门必修课程。

图 1.17 视频内容页-内容简介

留言列表

序号	内容	评论人	发表时间
1	推荐	zhangsan	2016-03-02 14:29:59
2	讲解细致	zhangsan	2015-12-09 16:15:07
3	推荐	zhangsan	2016-03-02 14:27:56

图 1.18 视频内容页-留言列表

您的留言

图 1.19 视频内容页-留言框

在视频详细内容页,用户可以点击视频标题在线播放视频,页面如图 1.20 所示。



图 1.20 在线播放视频页面

后台管理员用户登录成功后将进入管理员首页面。管理员实现对信息的管理功能,包括注册用户信息的修改、删除和搜索,视频类别信息和视频信息的增加、修改、删除和搜索,留言信息的删除等。

下面以视频信息的管理功能为例来展示网站的后台管理功能界面,其他管理功能界面类似。图 1.21 为管理员添加视频信息的界面,在此页面管理员可以为新增视频填写视频名称,选择视频类型,上传视频海报,填写视频简介和下载地址等信息。

请填写视频信息

视频名称	<input type="text" value="videoname"/>
视频类型	<input type="text" value="电影"/>
视频海报	<input type="button" value="选择文件"/> 未选择任何文件
视频简介	<input type="text"/>
下载地址	<input type="text" value="address"/>
<input type="button" value="添加"/> <input type="button" value="重置"/>	

图 1.21 添加视频信息页面

图 1.22 为视频信息的管理界面,管理员可以按照视频名称搜索视频,可以修改某一特定视频的信息,也可以删除视频。

请输入视频名称:

序号	视频名称	类别	添加时间	海报	操作
1	iOS	原创	2015-10-29 15:44:43		修改 删除
2	易语言	原创	2015-10-29 15:45:26		修改 删除
3	3Dmax	原创	2015-10-29 15:46:01		修改 删除
4	大圣归来	动画片	2015-10-29 15:48:17		修改 删除

图 1.22 视频信息管理页面

1.4.3 数据库设计

根据视频信息管理系统功能分析,本系统需要创建一个数据库(这里命名为 neuvideo)。neuvideo 数据库中共需要 7 张数据库表,分别是 users 表、admins 表、videotype 表、videos 表、comments 表、levels 表和 collect 表。其中,users 表用来存储注册用户的信息;admins 表用来存储管理员的用户名和密码信息;videotype 表用来存储视频类别信息;videos 表用来存储视频信息;comments 表用来存储用户对视频发表的留言信息;levels 表用来存储用户对视频的评分信息;collect 表用来存储用户收藏视频的信息。数据库的详细设计将在第 2 章中具体介绍。

1.5 开发技术与运行环境的选择

1.5.1 开发技术的选择

开发 Web 应用系统的技术很多,目前流行的 Web 开发技术包括 PHP、ASP、.NET 和 JSP 等。本教材中的视频信息管理系统是基于 B/S(浏览器/服务器)模式,以 PHP 开发技术和 MySQL 数据库为基础,结合 HTML、CSS、Bootstrap 等技术实现的交互式视频信息管理系统。

PHP(Professional Hypertext Preprocessor)是一种运行于服务器端的 HTML 嵌入式脚本描述语言。PHP 借鉴了 C、Java、Perl 等传统计算机语言的特性和优点,并结合自己的特性,使得 Web 开发者能够快速地编写出动态页面。PHP 是完全免费的开源产品,并且易学易用。PHP 可以很好地支持 Internet 协议和多种数据库的操作,经常和 MySQL 搭配使用。

1.5.2 运行环境的选择

基于 PHP 的 Web 开发环境有很多,本教材选择了 WAMP 作为开发、调试和运行程序的平台。

(1)WAMP

WAMP 是 Windows 下的 Apache、MySQL 和 PHP 的集成环境,是一组常用来搭建动态网站或者服务器的开源软件,本身都是各自独立的程序,但是因为常被放在一起使用,拥有了越来越高的兼容度,共同组成了一个强大的 Web 应用程序平台。Apache 是最通用的网络服务器;MySQL 是带有基于网络管理附加工具的关系数据库;PHP 是流行的对象脚本语言,它包含了多数其他语言的优秀特征来使得它的网络开发更加有效。与 JSP 和 ASP 相比,PHP 具有简易性、高安全性、开发速度快和执行灵活等优点,并且在 WAMP 平台中开发的项目,在软件方面的投资成本较低,运行稳定。WAMP 软件安装过程参见附录 A。

(2)MySQL

MySQL 是一个跨平台的开源关系型数据库管理系统,目前,MySQL 被广泛地应用在 Internet 的中小型网站中。由于其体积小、速度快,尤其是开放源代码的特点,许多中小型网站都选择 MySQL 作为网站的数据库。

(3)Apache

Apache HTTP Server(简称 Apache)是最流行的 Web 服务器端软件之一,是 Apache 软件基金会的一个开放源代码的网页服务器。它可以在大多数计算机操作系统中运行。Web 系统是客户端/服务器模式的,所以应该有服务器程序和客户端程序两部分。常用的服务器程序就是 Apache;常用的客户端程序是浏览器(如 IE、Chrome 等)。Apache 主要用来接收 Web 客户端用户发来的请求。Apache 收到请求后,将客户端要求的页面内容返回给客户端,如果出现错误,就返回错误代码。但 Apache 只能处理 HTML 请求,诸如 JSP、PHP 和 ASP 的请求则需要配置其他相应的服务器才能解析处理。

1.5.3 初识 PHP

1. 什么是 PHP

开发 Web 应用系统的技术很多,目前流行的 Web 开发技术包括 PHP、ASP、.NET 和 JSP 等等。PHP(Professional Hypertext Preprocessor)是一种运行于服务器端的 HTML 嵌入式脚本描述语言。PHP 借鉴了 C、Java、Perl 等传统计算机语言的特性和优点,并结合自己的特性,使得 Web 开发者能够快速编写出动态页面。PHP 是完全免费的开源产品,并且易学易用。PHP 可以很好地支持 Internet 协议和多种数据库的操作,经常和 MySQL 搭配使用。PHP 的创始人是丹麦人 Rasmus Lerdorf,生于 1968 年,于 1995 年创造了 PHP。当前 PHP 的最高版本是 PHP 7.0。

使用 PHP 进行 Web 应用程序开发有以下优势:

(1)易学易用

PHP 可以内嵌到 HTML 中,以脚本语言为主,内置丰富的函数,语法简单,是一个弱类型语言,学习方便。有 C、Java 等基础的开发者都很容易理解 PHP 的语法,相对于 JSP 等要

容易入门得多。集成开发环境容易搭建配置,开发软件也非常多样。

(2) 成本低、应用广泛

PHP 是开源软件,PHP 的运行环境 LAMP 平台(Linux、Apache、MySQL 和 PHP)也都是免费的,这种框架结构可以为网站经营者节省很大开支,所以很多中小型企业网站采用 PHP 开发。

(3) 执行速度快

占用资源少,速度快,内嵌 zend 加速引擎,性能稳定快速。

(4) 支持面向对象

同时支持面向过程和面向对象两种开发模式,用户可以自行选择。

(5) 支持广泛的数据库

可操作多种主流与非主流数据库,如 MySQL、Access、SQL Server、Oracle、DB2 等,其中 PHP 与 MySQL 是目前最佳的组合,它们的组合可以跨平台运行。

(6) 跨平台性

PHP 几乎支持所有的操作系统,并且支持 Apache、IIS、Nginx 等多种 Web 服务器。

由于 PHP 的以上优势,所以 PHP 的应用领域非常广阔,比较常见的应用有中小型网站的开发、大型网站的业务逻辑结果展示、Web 办公管理系统、硬件设备的数据获取、电子商务应用、企业级应用开发,以及微信公众号和小程序等。

2. 第一个 PHP 文件

PHP 是嵌入 HTML 中的语言,HTML 负责呈现网页内容,PHP 负责业务逻辑,我们先来看一个简单的 PHP 文件,代码如下:

first.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>我的第一个 PHP 文件</title>
</head>
<body>
  <? php
    echo "<font color='blue'>你好世界! </font>";
  ? >
  <br>
  <? php
    for( $ i=1; $ i<=6; $ i++){
      echo "<h $ i>";
      echo "你好世界!";
      echo "</h $ i>";
    }
  ? >
</body>
</html>
```

运行效果如图 1.23 所示。第一行显示的是蓝色字“你好世界!”。我们看到了 PHP 代

码中混有 HTML 代码,对于 PHP 来说,HTML 代码只是字符串。第二行到第七行使用 HTML 的标题元素从大到小显示了“你好世界!”。我们可以看到复杂的业务逻辑需要 PHP 的参与,这样简化了代码。其中 \$ 美元符号是 PHP 中变量的标识,echo 是 PHP 中用来进行输出的语句结构。



图 1.23 第一个 PHP 文件运行界面

实践练习

简答题

1. 试说明软件开发中的 B/S 架构和 C/S 架构之间的区别。
2. 列举常见的 Web 服务器和数据库服务器。
3. 列举你所熟知的动态网页程序设计语言。
4. 试说明动态网站开发所需要的 Web 构件。
5. 默认情况下,Apache 服务器的配置文件名、MySQL 服务器以及 PHP 预处理器的配置有哪些?
6. 简述 Apache 和 PHP 的工作流程。

第 2 章 项目数据库设计

一、概述

通过第 1 章内容的介绍,我们已经了解课程项目(视频信息管理系统)需要完成的功能,接下来就要对项目功能的具体实施进行设计,包括数据库设计和页面设计两部分工作。本章将介绍数据库设计这一部分。动态网站的数据基本上都是存储于数据库中的,而且是允许修改和维护的。本章将介绍 MySQL 数据库的基础知识、Navicat 软件的使用、MySQL 数据库的常见操作(数据库连接与关闭,创建数据库,创建数据表,数据表内容的简单管理)、数据表设计、数据字段属性、SQL 语句设计等内容,并以用户管理子系统项目中用到的数据表为例讲解创建数据库和创建表的流程,最后完成课程项目的所有数据表的设计工作。

二、重点与难点

重点:

- (1)数据库在 Web 开发中的地位。
- (2)MySQL 数据库的常见操作。

难点:

- (1)MySQL 数据库的常见操作。
- (2)使用 Navicat 操作数据库。

【项目描述】设计视频信息管理系统数据库

创建视频信息管理系统后台数据库 neuvideo,在数据库中创建系统中 7 张数据表:用户信息表 users、管理员信息表 admins、视频类型表 videotype、视频信息表 videos、评论信息表 comments、评分表 levels 和收藏表 collect。

(1)用户信息表 users

users 表用来存储注册用户的信息,字段包括用户 ID、用户名、密码、性别、电话、上传的头像、电子邮箱、注册时间和更新时间。users 表结构如表 2.1 所示。

表 2.1

用户信息表 users

列名	数据类型	约束	备注
uid	int(11)	PRIMARY KEY AUTO_INCREMENT	用户 ID
uname	varchar(40)	NOT NULL	用户名
password	varchar(50)	NOT NULL	密码
gender	tinyint(4)	NOT NULL	性别
tel	varchar(20)	NOT NULL	电话
photo	varchar(50)	NOT NULL	上传的头像
email	varchar(50)	NOT NULL	电子邮箱
createtime	datetime	NOT NULL	注册时间
updatetime	timestamp	NOT NULL	更新时间

(2) 管理员信息表 admins

admin 表是用来存储管理员的信息, 字段包括管理员用户 ID、用户名、密码。admins 表结构如表 2.2 所示。

表 2.2

管理员信息表 admins

列名	数据类型	约束	备注
adminid	int(11)	PRIMARY KEY AUTO_INCREMENT	管理员用户 ID
adminname	varchar(40)	NOT NULL	管理员用户名
password	varchar(50)	NOT NULL	密码

(3) 视频类型表 videotype

videotype 表用来存储视频类型信息, 该信息在添加视频时会用到。字段包括视频类型 ID、视频类型名。videotype 表结构如表 2.3 所示。

表 2.3

视频类型表 videotype

列名	数据类型	约束	备注
tid	int(11)	PRIMARY KEY AUTO_INCREMENT	视频类型 ID
typename	varchar(20)	NOT NULL	视频类型名

(4) 视频信息表 videos

videos 表用来存储视频信息, 字段包括视频 ID、视频名称、视频类型、视频海报图片、视频简介、视频上传时间、上传人、点击量、下载量、下载地址。videos 表结构如表 2.4 所示。

表 2.4

视频表 videos

列名	数据类型	约束	备注
vid	int(11)	PRIMARY KEY AUTO_INCREMENT	视频 ID
videoname	varchar(30)	NOT NULL	视频名称
tid	int(11)	NOT NULL FOREIGN KEY	视频类型
pic	varchar(30)	NOT NULL	视频海报图片

(续表)

列名	数据类型	约束	备注
intro	varchar(2000)	NOT NULL	视频简介
uploaddate	datetime	NOT NULL	视频上传时间
uploadadmin	int(11)	NOT NULL	上传者
hittimes	int(11)	NOT NULL DEFAULT '0'	点击量
downtimes	int(11)	NOT NULL DEFAULT '0'	下载量
address	varchar(200)	NOT NULL	下载地址

(5) 评论信息表 comments

comments 表用来存储登录后的用户对视频发表的评论信息, 字段包括评论 ID、评论内容、评论发表的时间、发表评论的用户 ID、被评论的视频 ID。comments 表结构如表 2.5 所示。

表 2.5 评论信息表 comments

列名	数据类型	约束	备注
cid	int(11)	PRIMARY KEY AUTO_INCREMENT	评论 ID
content	varchar(600)	NOT NULL	评论内容
cdate	datetime	NOT NULL	评论日期
uid	int(11)	NOT NULL FOREIGN KEY	评论用户
vid	int(11)	NOT NULL FOREIGN KEY	评论视频

(6) 评分表

levels 表用来存储登录后的用户对视频的评分, 字段包括级别 ID、被评分的视频 ID、打分的用户 ID、评分的分值。levels 表结构如表 2.6 所示。

表 2.6 评分表 levels

列名	数据类型	约束	备注
lid	int(11)	PRIMARY KEY AUTO_INCREMENT	级别 ID
vid	int(11)	NOT NULL FOREIGN KEY	视频 ID
uid	int(11)	NOT NULL FOREIGN KEY	打分用户 ID
score	int(11)	NOT NULL	评分

(7) 收藏表 collect

collect 表用来存储用户收藏的视频, 字段包括收藏 ID、被收藏的视频 ID、打分的用户 ID。collect 表结构如表 2.7 所示。

表 2.7 收藏表 collect

列名	数据类型	约束	备注
clid	int(11)	PRIMARY KEY AUTO_INCREMENT	收藏 ID
vid	int(11)	NOT NULL FOREIGN KEY	视频 ID
uid	int(11)	NOT NULL FOREIGN KEY	收藏用户 ID

2.1 MySQL 数据库基础知识

数据库是以结构化格式组织的数据的集合,这些数据集合是需要通过某种程序或者软件创建和维护的,MySQL 就是众多软件中的一员,SQL Server 和 Oracle 也是其中的成员。值得一提的是,MySQL 本身并不是数据库,它只是用来创建、维护和管理数据库的计算机软件,这些软件被称为数据库管理系统(DBMS)。通常情况,开发人员就是通过数据库管理系统来维护数据库中的数据。

MySQL 是一款免费的软件,它基于开放软件的理念,提供免费和低成本的数据解决方案。重要的是,它在性能、安全和稳定性方面,完全可以满足大多数 Web 开发的需要。MySQL 的特点就是灵活而不是强大,和 Oracle 等 DBMS 相比,它本身并不庞大,但仍然集中了大量特性,使其快速而高效。对于中小规模的数据库需求来说,它已足够。

MySQL 是基于关系型的数据库,当前,大多数数据库都是关系数据库。关系数据库是数据库类别中的一种,它将数据组织成表,并表现为表与表之间的关系。通过这种关系,数据库管理系统可以从不同表中提取某种特定的数据集。

2.1.1 MySQL 数据库的存储引擎

“存储引擎”从字面理解,“存储”的意思是存储数据;“引擎”一词来源于发动机,它是发动机中的核心部分。在软件工程领域,相似的称呼有“游戏引擎”“搜索引擎”,它们都是相应程序或系统的核心组件。

数据库引擎是用于存储、处理和保护数据的核心服务。我们知道关系型数据库的数据是存在表里的,可以将表理解为由行和列组成的表格,类似于 Excel 的电子表格的形式。表是在存储数据的同时,还要组织数据的存储结构,而这些数据的组织结构就是由存储引擎决定的。

现实生活中,由于不同业务产生了不同的数据,这些数据有的可能偏重于被频繁查询,有的要求增删速度快,有的则对事务、索引、外键有特殊规定。这样一来,就对存储数据的数据表要求使用不同的数据组织结构,也就是存储引擎。MySQL 支持多种引擎,常用的引擎如表 2.8 所示。

表 2.8

MySQL 数据库引擎介绍

数据库引擎	功能
MyISAM	MyISAM 强调快速读取操作,拥有较高的插入、查询速度,但不支持事务和外键
CSV	逻辑上由逗号分隔数据的存储引擎。它会在数据库子目录里为每个数据表创建一个 .CSV 文件。这是一种普通文本文件,每个数据行占用一个文本行。CSV 存储引擎不支持索引
MEMORY(HEAP)	所有数据置于内存的存储引擎,拥有极高的插入、更新和查询效率。但是会占用和数据量成正比的内存空间,并且其内容会在 MySQL 重新启动时丢失

(续表)

数据库引擎	功能
ARCHIVE	非常适合存储大量的独立的、作为历史记录的数据。因为它们不经常被读取。Archive 拥有高效的插入速度,但其对查询的支持相对较差
InnoDB	InnoDB 支持事务处理和外键约束,这两点都是 ISAM 和 MyISAM 两个引擎所没有的。所以,如果数据库设计需要这两个特性中的一者或者两者,那就可以选择使用 InnoDB 引擎

可以通过查看数据库引擎的 SQL 命令来查看当前 MySQL 使用的是哪一种引擎,命令如下:

```
SHOW ENGINES;
```

在 WAMP 集成开发环境中,MySQL 的默认数据库引擎是 InnoDB。

2.1.2 MySQL 数据库的数据类型

在创建一个表时,需要指定字段的数据类型,MySQL 根据字段的数据类型来决定如何存储数据。比如一个字段用来存放班级学生人数,就要使用无符号的 INT 数据类型。又如,一个字段用来存放用户提交的意见或建议,就应该考虑使用 VARCHAR 类型或者 TEXT 类型。本节将介绍 MySQL 常用的数据类型。

1. 数值型

表 2.9 列出了 MySQL 数据库的常用数值类型。

表 2.9 数值型数据类型

MySQL 数据类型	字节	说明
TINYINT	1 字节	存储从 $-2^8 \sim 2^8 - 1$ 范围的整数
SMALLINT	2 字节	存储从 $-2^{15} \sim 2^{15} - 1$ 范围的整数
INT	4 字节	存储从 $-2^{31} \sim 2^{31} - 1$ 范围的整数
BIGINT	8 字节	存储从 $-2^{63} \sim 2^{63} - 1$ 范围的整数
FLOAT	4 字节	存储从 $-3.40E+38 \sim 3.40E+38$ 范围的浮点型数
DOUBLE	8 字节	存储从 $-1.79E+308 \sim 1.79E+308$ 范围的浮点型数
DEC(p,q) 或 DECIMAL(p,q)		定点精度和小数位数。p 为精度,指定小数点左边和右边一共可以存储的十进制数字的最大个数。q 为小数位数,指定小数点右边可以存储的十进制数字的最大个数。q 的默认值为 0

2. 字符串类型

字符串类型数据由汉字、英文字母、数字和各种符号组成。表 2.10 列出了 MySQL 数据库的常用字符串类型。

表 2.10

字符串型数据类型

MySQL 数据类型	说明
CHAR(N)	固定长度的字符串类型,N 表示字符串的最大长度,取值范围为 0~255
VARCHAR(N)	可变长度的字符串类型,N 表示字符串的最大长度,取值范围为 0~65535
TEXT	0~65535 个字节,可存储大文本
LONGTEXT	0~2 ³² -1(4294967295)个字节,可存储超大文本(约为 2GB)
BINARY(N)	固定长度的二进制字符数据(如图片、音乐、视频文件等),N 表示最大长度,取值范围为 1~8000

字符串类型的常量两端要用单引号括起来。如:'hello','how are you','18274'都是字符串类型的数据。

3. 日期时间类型

表 2.11 列出了 MySQL 常用的日期时间数据类型。

表 2.11

日期时间型数据类型

MySQL 数据类型	格式	含义
DATE	YYYY-MM-DD	日期,表示 1000-01-01~9999-12-31 之间的日期
TIME	HH:MM:SS	时间,表示 -838:59:59~838:59:59 之间的时刻
DATETIME	YYYY-MM-DD HH:MM:SS	日期时间,表示 1000-01-01 00:00:00~9999-12-31 23:59:59 之间的日期时间
YEAR	YYYY	年份类型,1901~2155 之间的某一年

日期类型的常量两端要用单引号括起来。日期类型常量有英文数字格式、数字加分隔符格式和纯数字格式三种。采用英文数字格式时,月份可用英文全名或缩写形式,不区分大小写。例如 2015 年 10 月 25 日可采用以下几种输入格式:

'Oct 25 2015' /* 英文数字格式 */

'2015-10-25'或 '2015/10/25' /* 纯数字加分隔符 */

'20151025' /* 纯数字格式 */

DATETIME 类型输入时间部分可以采用 12 小时格式或 24 小时格式。使用 12 小时格式时要加上 AM 或 PM,以便说明是上午还是下午。在时与分之间可以使用冒号作为分隔符。例如,要表示 2015 年 10 月 25 日 下午 3 点 28 分 56 秒,可以用如下形式输入:

'2015-10-25 3:28:56 PM' /* 12 小时格式 */

'2015-10-25 15:28:56' /* 24 小时格式 */

4. 数据完整性约束

设计数据库时,可以对表中的一些字段设置约束条件。常用的约束条件有 6 种:主键约束(PRIMARY KEY)、外键约束(FOREIGN KEY)、唯一性约束(UNIQUE)、默认值约束(DEFAULT)、非空约束(NOT NULL)以及检查约束(CHECK)。

(1) PRIMARY KEY

主键约束。比如商品信息表中的商品号,每个商品都有一个确切的商品号,并且每个商品的商号都不会重复。那么商品号就可以定义为主键,用以唯一地确定一个商品记录。有些

表中一个列上的取值不足以唯一地标识一个元组,那么可以将多个列组合在一起作为主键。

主键上的取值既不允许重复,也不允许为空。添加数据的时候,如果违反这两条中的任一规则,会提示出错。

关系模型的实体完整性要求每张表必须有且仅有一个主键。

(2) FOREIGN KEY

外键约束。可以设置表中的一个(或多个)属性,让它引用某个表的主键(通常情况下),使表中的属性集的取值受它引用的属性集取值的约束,这个受约束的属性集称为外键。使用形式为:FOREIGN KEY (外键列) REFERENCES 表名(主键列)。例如:视频信息表 video 中的视频类型号 tid 列是一个外键列,它需要参照视频类型表 videotype 中的主键 tid 列的取值范围而取值,即 video 中的 tid 只能取 videotype 表中已有的视频类型编号,其外键定义为:FOREIGN KEY (tid) REFERENCES videotype (tid)。MySQL 中需设置存储引擎是 InnoDB,才支持外键约束。

(3) UNIQUE

唯一值约束。限定某一列的取值唯一。换句话说就是列的取值不能重复。比如顾客的身份证号这列,每个人的身份证号都不相同,如果这一列上出现重复值必定是错误的。那么可以设定身份证号的约束为 UNIQUE,重复的数据将不允许出现在这一列上。

(4) DEFAULT

默认值约束。可以设定某列的默认值。使用形式为:DEFAULT 常量。

(5) NOT NULL

非空约束。限定某列的取值不允许为空(NULL)。假设顾客信息表中的顾客姓名这一列不允许为空,那么可以设定姓名的约束为 NOT NULL,这样如果在姓名列上出现空的数据,DBMS 会提示出错。

(6) CHECK

检查约束。用以限定列的取值范围。使用形式为 CHECK(约束表达式)。例如限定商品价格大于 0,在创建表时可以声明 CHECK(价格>0)。

除了以上 6 种约束之外,MySQL 还扩展了标准 SQL 中的完整性约束,增加了 AUTO_INCREMENT 约束,设定该约束的属性列的值会自动增加。一般会在整型的主键字段设定此约束,默认从整数 1 开始,每次增加 1。

2.1.3 MySQL 字符集与字符序

MySQL 由瑞典 MySQL AB 公司开发,默认情况下,使用的是 latin1 字符集(西欧 ISO_8859_1 字符集的别名)。由于 latin1 字符集是单字节编码,而汉字是双字节编码,由此可能导致 MySQL 数据库不支持中文字符串查询,或者发生中文字符串乱码等问题。为了避免问题出现,我们需要了解字符集和字符序的相关概念,并进行必要的字符集和字符序的设置。

字符(character)是人类语言最小的表意符号,如‘A’、‘B’等。给定一系列字符,并对每个字符赋予一个数值,用数值来代表对应的字符,这个数值就是字符的编码(character encoding)。例如,假设给字符‘A’赋予整数 65,给字符‘B’赋予整数 66,则 65 就是字符‘A’的编码,66 就是字符‘B’的编码。

给定一系列字符并赋予对应的编码后,所有这些“字符和编码对”组成的集合就是字符

集(character set)。MySQL 提供了多种字符集,例如 latin1、utf8、gbk 等。不同的字符集支持不同地区的字符。例如,latin1 支持西欧字符、希腊字符等,gbk 支持中文简体字符,big5 支持中文繁体字符,utf8 几乎支持世界上所有国家的字符。由于每种字符集支持的字符个数各不相同,各种字符集占用的存储空间也不相同。

字符序(collation)是指在同一字符集内字符之间的比较规则。只有确定字符序后,才能在一个字符集上定义什么是等价的字符,以及字符之间的大小关系。一个字符集可以包含多种字符序,每个字符集有一个默认的字符序(default collation)。MySQL 字符序命名的规则是:以字符序对应的字符集名称开头,以国家名居中(或以 general 居中),以 ci、cs 或 bin 结尾。以 ci 结尾的字符序表示大小写不敏感,以 cs 结尾的字符序表示大小写敏感,以 bin 结尾的字符序表示按二进制编码值比较。例如,latin1 字符集有 latin1_swedish_ci、latin1_general_cs 和 latin1_bin 等字符序,其中在字符序 latin1_swedish_ci 规则中,字符‘a’和‘A’是等价的。

为了更好地支持中文检索以防止乱码问题,建议 MySQL 的字符集设置为 gbk 简体中文字符集或者 utf8 字符集。可以通过修改 my.ini 配置文件来修改 MySQL 默认的字符集。具体操作可以将 my.ini 配置文件中的 client 选项组中的 default-character-set 参数修改为 utf8,将 mysqld 选项组中的 character-set-server 参数值修改为 utf8,重启 MySQL 服务,这些字符集设置将生效。

2.1.4 MySQL 数据库的 SQL 语法基础

SQL 的英文全称是 Structured Query Language(结构化查询语言),使用它可以完成对数据库的管理操作。比如数据查询、添加、修改和删除等。

1. 添加数据

可以使用 SQL 语句的 INSERT 命令向表中插入数据,INSERT 语句格式是:

```
INSERT INTO 表名[(列名列表)]  
VALUES(值列表)[, (值列表), ...]
```

此语句将一行或多行新记录插入到指定表中。其中值列表的值应该与列名列表的列一一对应。列名列表中未出现的属性列将取空值。如果列名列表省略,则值列表应该与基本表的列一一对应。

例 2.1 向 videotype 表中插入如下三条记录。

```
INSERT INTO videotype VALUES (null, '电影'), (null, '电视剧'), (null, '动画片');
```

2. 更新数据

更新数据即修改表中数据的操作,SQL 中更新数据的命令是 UPDATE,其语法格式为:

```
UPDATE <表名>  
SET <列名 1>=<表达式 1> [, <列名 2>=<表达式 2>, ...]  
[WHERE <条件>]
```

其中,SET 子句指定要更新的列并计算更新后的值,如果需要更新多列,则用逗号分隔;利用 WHERE 子句可以实现条件更新。

例 2.2 更新 videotype 表中 tid 为 4 的记录信息。

```
UPDATE videotype set typename='科技' where tid=4;
```

3. 删除数据

SQL 删除数据的命令是 DELETE。其一般格式为：

```
DELETE FROM <表名>
[WHERE <条件>]
```

该语句的功能是从指定表中删除满足<条件>的所有元组。如果省略 WHERE 子句，表示删除表中全部数据。

例 2.3 删除 videotype 表中 tid 为 6 的记录信息。

```
DELETE FROM videotypewhere tid=6;
```

4. 查询数据

查询的 SQL 命令是通过使用 SELECT 语句来实现，其功能是从数据库中检索满足条件的数据。数据的来源可以是一张表，也可以是多张表。查询的结果是由 0 行（没有满足条件的数据）或多行记录组成的一个记录集合（结果集），并且可以选择一个或多个字段作为输出字段。SELECT 语句还能实现对查询的结果进行排序或汇总等，该语句具有灵活的使用方式。其一般格式是：

```
SELECT [DISTINCT]<字段 1 [as 别名 1]>[, <字段 2 [as 别名 2]>]... /* 查询哪些列 */
FROM <表名>[, <表名>]... /* 来自于哪些表 */
[WHERE <检索条件>] /* 根据什么条件 */
[GROUP BY <字段> [HAVING <组提取条件>]] /* 按哪列分组及分组的过滤条件 */
[ORDER BY <列名 2>[ASC|DESC]] /* 按一个或多个字段排序查询 */
[LIMIT count|number1,number2]; /* 对结果记录显示的限制 */
```

该语句的功能是从 FROM 子句中的表或视图中找出满足 WHERE 子句中条件的元组，再将这些元组在 SELECT 子句中规定的列上做投影，最后得到一个结果关系。其余子句都是对得到的结果关系进行再处理。

- GROUP BY 子句将结果关系按<列名 1>的值进行分组
- HAVING 子句用于指定组的选择条件。
- ORDER BY 子句将结果关系中的数据按<列名 2>的值进行升序或降序排列。
- LIMIT 是 MySQL 对标准的 SQL 查询语句的一个功能扩充，它能限定显示查询结果的行，可以是前 count 行记录，也可以是中间的从 number1 到 number2 的记录行。

例 2.4 模糊搜索 videos 表中视频名包含“特工”的视频信息。

```
SELECT * FROM videos WHERE videoname like '%特工%';
```

例 2.5 查询点击率排行在前 10 名的视频信息。

```
SELECT * FROM videos ORDER BY hittimes DESC LIMIT 0,10;
```

例 2.6 查询所有视频的视频信息，视频类型显示类型名。

```
SELECT * FROM videos JOIN videotype ON videos.tid=videotype.tid;
```

2.1.5 数据库用户权限管理

1. MySQL 用户权限系统介绍

MySQL 的最好特性之一是支持复杂的权限系统。权限是对特定对象执行特定操作的权利,它与特定用户相关。其概念非常类似于文件的权限。当在 MySQL 中创建一个用户时,就赋予了该用户一定的权限,这些权限指定了该用户在本系统中可以做什么与不可以做什么。

最少权限原则可以用来提高任何计算机系统的安全性。它是一个基本的,但又是非常重要的,而且容易被忽略的原则。该原则是:一个用户应该拥有能够执行分配给他的任务的最低级别的权限。例如:只需要在网站上运行查询,用户并不需要 root 用户所拥有的所有权限。因此,应该为该网站应用创建另一个用户,他只有访问网站数据库的必要权限,如查询。

2. 设置用户与权限

一个 MySQL 系统可能有许多用户。为了安全起见,root 用户通常只用作管理目的。对于每个需要使用该系统的用户,应该为他们创建一个账号和密码。这些用户名和密码尽量不要与 MySQL 之外的用户名称和密码相同。对于系统用户和 MySQL 用户最好使用不同的密码,尤其对 root 用户更应该这样。

为用户设置密码不是必须的,但是强烈建议为所有创建的用户设定密码。要建立一个 Web 数据库,最好为每个网站应用程序建立一个用户。

可以使用下面的命令创建一个用户 neuuser,密码也是 neuuser。

```
CREATE USER 'neuuser'@'localhost' IDENTIFIED BY 'neuuser';
```

赋予用户权限使用 GRANT 命令,其语法如下:

```
GRANT 权限列表[(列名)] ON 对象名  
TO 用户名 [IDENTIFIED BY '密码']  
[WITH GRANT OPTION]
```

(1)权限列表的权限可以分为四个级别:全局级、数据库级、表级和列级。一般权限是限定到表级的,如果想限定到列级,可以使用“权限名(列名列表)”的形式,比如“grant select (name,price) on book to user1”就是把 book 表中 name 列和 price 列的查询权限授予 user1。一般情况下,INSERT、UPDATE、DELETE、SELECT、CREATE、ALTER、DROP、INDEX 权限适用于一般用户。

(2)对象名是权限所应用于的数据库或表。“*.*”表示权限作用于当前服务器中的所有数据库的所有表,这一权限是全局级权限。“数据库名.*”表示权限作用于特定数据库的所有表,这是数据库级别的权限。“数据库名.表名”指定权限作用于特定数据库的单张表,这是表级别的权限。表名(列名)用来限定权限到特定的列,属于列级权限。如果输入命令时正在使用一个数据库,则表名本身将被解释成当前数据库中的一个表。

(3)用户名是用户登录 MySQL 所使用的用户名,它不必与登录系统时所使用的用户名相同,它还可以包含一个主机名,如用户名@localhost 或用户名@somewhere.com。

(4)WITH GRANT OPTION 选项,如果指定,则表示允许指定的用户向别人授予自己所拥有的权限。

例如,下面的 GRANT 语句实现将 neuvideo 数据库的全部表的增删改查权限授予用户

neuuser。

```
GRANT SELECT , INSERT , DELETE , UPDATE ON neuvideo. * TO 'neuuser'@'localhost' IDENTIFIED BY 'neuuser';
```

收回用户权限的命令是 REVOKE。它的语法与 GRANT 很相似：

```
REVOKE 权限列表[(列名)] ON 对象名 FROM 用户名
```

如果已经给出了 WITH GRANT OPTION 子句,可以按如下方式撤销它(以及所有其他权限):

```
REVOKE ALL PRIVILEGES, GRANT FROM 用户名
```

2.2 使用 Navicat 管理数据库

Navicat 的安装软件可以从 <https://www.navicat.com/> 网站下载安装。Navicat Premium 是一套数据库管理工具,可以同时连接到 MySQL、MariaDB、SQL Server、SQLite、Oracle 和 PostgreSQL 数据库。Navicat Premium 可满足现今数据库管理系统的使用功能,包括存储过程、事件、触发器、函数、视图等。本节将介绍 Navicat 管理 MySQL 数据库的方法。

2.2.1 启动 Navicat 并配置数据库连接

1. 启动 Navicat

首先启动 Wamp Server 程序。Wamp Server 启动后,MySQL 服务器也会随之启动,如果 MySQL 服务器未启动,可以通过选择 Wamp Server|MySQL|service|Start service 来启动。如果需要重新启动 MySQL 服务器,则选择 Wamp Server|MySQL|service|Restart service。如果需要重新关闭 MySQL 服务器,则选择 Wamp Server|MySQL|service|Stop service。

MySQL 服务器启动后,选择【开始】→【程序】→【Navicat Premium 12】或直接点击桌面快捷方式,启动 Navicat,主界面如图 2.1 所示。

2. 配置连接

在图 2.1 中,点击工具栏中的【Connection】图标,在下拉菜单中选择【MySQL】,弹出配置连接的窗口,如图 2.2 所示。

在图 2.2 中的【General】选项卡中,输入连接名(myCon),MySQL 服务器的主机名或 IP 地址(localhost)、端口号(3306)、连接 MySQL 时所用的用户名(root)和密码(空,不需填写),点击连接测试【Test Connection】按钮,出现“连接成功”的提示后,点击【OK】按钮。在左侧连接栏中,双击【myCon】连接,界面如图 2.3 所示。

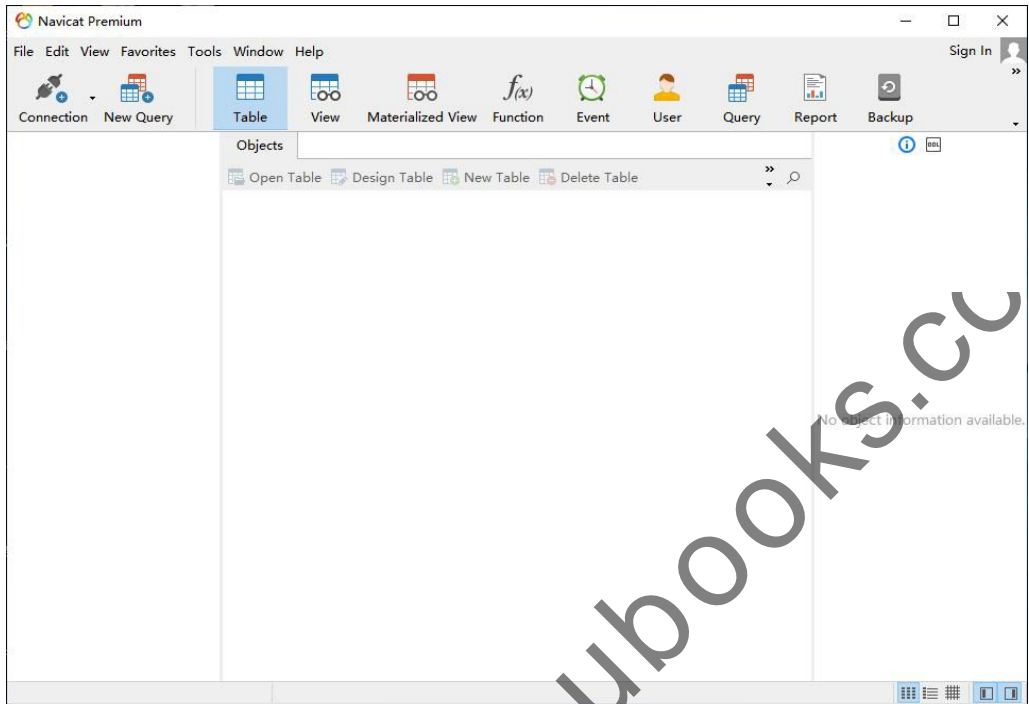


图 2.1 Navicat 主界面



图 2.2 配置连接的窗口