Web 数据库开发与应用 (ASP+SQL Server)

吕少华 边 宇 编著

版社 举 <u>k</u>

2001

本书是以当前最流行的 ASP+SQL Server 的形式,介绍如何构建一个功能 完善的网络数据库。

全书分为两篇。第1篇为 Web 数据库基础篇,包括第1章~第7章,读者可 以从中了解和掌握一个完整的 Web 数据库的开发设计步骤;BBS 论坛的设计 与开发;后台数据库的选择、建立和管理;ODBC 数据源的建立;ASP 技术以及 ASP 程序设计技巧。

第 2 篇为 Web 数据库实例篇,该篇用了 3 章的篇幅,详细介绍基于 Web 的 文档资料管理系统和搜索引擎,并给出了 9 个实用程序片断,这对 Web 数据库 开发者具有重要的参考价值。

本书语言流畅、条理清晰,既有理论又有实例,适合 Web 数据库开发人员 作为参考手册。

图书在版编目(CIP)数据

Web 数据库开发与应用(ASP+SQL Server)/吕少华等编著. 一北京: 科学出版社,2001.6 ISBN 7-03-009430-1

I.W… I. 吕… I. 万维网一数据库一程序设计 Ⅳ. TP393.4

中国版本图书馆 CIP 数据核字(2001)第 037612 号

44 学 生 施 社 出版 北京东黄城根北街16 号 邮政编码 100717

北京门头沟胶印厂印刷

科学出版社发行 各地新华书店经销

2001年7月第一版	开本:787×1092 1/16
2001 年 7 月第一次印刷	印张:24 3/4
印数:1-5 000	字数:601 920

定价:32.00元

前言

最近几年, Internet 技术得到持续、迅猛的发展,已经成为计算机产业的一个技术热点, 促进 Internet 高速发展的因素之一就是 WWW 技术。虽然 WWW 已经诞生多年了,但长期 以来,它基于一种简单的传输协议,向用户发送的是完全静态的文本与图像。

在引入了可以运行与用户交互的脚本的 Web 服务器后,使得原来的 Web 页面更加精彩。 人们熟悉的客户/服务器开发模式逐渐引用到 Web 上。ASP(Active Server Page)这种动态 服务网页技术的出现,更推动了客户/服务器的开发模式的发展,由于其简便,可操作性强, 已经被广大的 Web 服务系统与 Web 数据库开发者所利用。

ASP 是一套微软公司开发的服务器端的脚本环境, ASP 内含于 IIS3.0 以上版本中, 通 过 ASP 可以结合 HTML 网页、ASP 指令和 ActiveX 控件建立动态、交互且高效的 Web 服 务器应用程序。有了 ASP 开发者不用再担心用户端的浏览器是否能运行所编写的代码, 因 为所有的程序都是在服务器端执行,包括所有嵌在普通 HTML 中的脚本程序。当程序执行 完毕后,服务器仅将执行的结果返回给客户浏览器,这样也就减轻了客户端浏览器的负担, 大大提高了交互的速度。

但仅有 ASP 技术是无法开发一个完整的 Web 服务系统,还需要服务器端后台数据库的 支持来存储相关的信息资料。把 ASP 这样的动态技术与服务器端的后台数据库有机结合起 来,就组成了现在非常流行的动态 Web 数据库。目前,网络上现有的 BBS 论坛、网上购 物、资料查询、聊天室、留言簿等交互系统,其中大部分都是通过 ASP 和后台数据库结合 的动态技术来实现的。

服务器端的后台数据库目前最流行的是 Microsoft SQL Server,其最新版本是 Microsoft SQL Server 2000 版本。它的功能强大,操作起来比其他一些数据库又相对简单,所以应用 十分广泛。当然,Microsoft SQL Server 并不是唯一的选择,功能更强大,在大型网站中应 用更多的是 ORACLE 数据库,但是它的安装及使用更加专业化和复杂化,需要专业的数据 库人员来操作;另一方面是,一般的中小型网站并不需要这样庞大的数据库后台服务,Microsoft SQL Server 完全可以满足,而且由于 Microsoft SQL Server 是微软公司的产品,与 Windows 系列的服务器操作系统之间采用无缝的底层结合技术,具有更高的安全机制。因 此,在本书中讲述的所有实例都是以 Microsoft SQL Server 为后台数据库。

一般情况下,一个 Web 数据库系统的工作过程是这样的:首先客户端向服务器发出请 求指令,服务器端收到请求指令之后,首先对客户端的身份进行验证,确认后执行相应的 程序,通过服务器后台系统,对数据库进行客户端发出的相关操作,然后再次通过服务器 后台系统把执行的结果返回到服务器的前台,动态生成网页,通过网络再传给客户端的浏 览器,客户就可以看到发出的请求指令的执行结果了。从上面的过程可以看出,大部分的 程序执行都是在服务器端完成的。

要开发一个完整的 Web 数据库系统,需要多方面的技术,首先需要熟悉 ASP 编程语言 及运行环境、后台数据库的开发技术以及一些相关的网络知识。但是,要在短时间内熟悉 所有这些技术是不可能的,而且对于一些初学者来说,更是感到无从下手,不知从哪里学起,一些 Web 数据库的开发实例对于初学者往往可以起到很好的作用,这也正是本书的出发点。对于 Web 数据库系统的高级开发人员,本书也有很好的参考价值。

从技术角度分析,一个完善的 Web 数据库系统由两个模块构成:前台应用程序和后台的数据库支持。在具体的开发过程中,需要编写前台的应用程序,也就是 ASP 程序,用 VBScript 来实现,还需要设置好前台应用程序的运行环境,也就是 Windows 2000 Server 和 IIS5.0;开发后台的数据库,主要包括设计合理的数据库表,系统运行之后,做好数据库的 日常备份和维护工作。

本书正是按照 Web 数据库系统的开发步骤来讲述相关的技术知识的。本书共分两个部分,第1部分通过笔者自己开发的一个 BBS 论坛实例介绍 Web 数据库的初步知识, 使读者对 Web 数据库系统有一个初步的认识和理解, 为后面的开发实例作一些准备工作; 第2部分是本书的提高部分, 列举了几个完善的 Web 数据库系统的实例, 通过讲述这些数据库系统的开发过程, 介绍开发技巧和经验。具体各章的内容分别如下:

第1章:介绍 Web 数据库的初步知识,包括开发技术的发展历史,使读者对 Web 数据 库系统从整体上有一个初步的了解和认识。

第2章:本章从使用者的角度介绍一个 BBS 论坛系统,首先介绍了整个论坛的应用和 安全性考虑,然后从技术角度介绍了 BBS 论坛的模块结构,最后对 BBS 论坛的功能进行 评价。在此,读者将看到一个 Web 数据库系统的开发成果的应用,了解 Web 数据库系统可 以实现哪些功能。

第3章:构建后台数据库,首先介绍 Microsoft SQL Server 的基础知识,讲述 Microsoft SQL Server 的安装、管理和使用,并举例说明如何建立一个实用的数据库以及数据库表, 着重介绍了 Microsoft SQL Server 中的数据类型和 SQL 语句,这对于 Web 数据库的开发是 至关重要的,最后,讲述了 SQL Server 数据库的高级管理、数据备份、日常管理等方面的 知识。通过本章的学习,将学会如何操作 Microsoft SQL Server 数据库,如何建立一个满足 实际应用的数据库表。

第4章:本章主要讲述 ASP 程序的开发准备工作,首先介绍 Windows 2000 Server 下 如何正确配置 IIS5.0,以保证开发出来的 ASP 程序顺利地安全运行;然后简单介绍了 ASP 程序的开发编辑工具 Microsoft FrontPage2000 的使用;最后简要介绍了 HTML 的语法,因为 ASP 的页面程序中必须嵌套一定的 HTML 语句,了解 HTML 的语法是很必要的。

第5章:本章着重介绍 VBScript, VBScript 是 ASP 的缺省脚本语言,开发 ASP 程序, 必须了解 VBScript 的使用,因此本章从变量、运算符、数据类型、循环语句、条件语句、 对象和函数几个方面对 VBScript 的语法结构进行全面介绍。

第6章: 了解了 VBScript 的语法结构后, 必须学习 ASP 的对象, 以完成一些功能更强 大的操作, 比如数据库的连接等等。本章着重介绍 ASP 的 7 个内嵌对象。

第7章:学习了 Microsoft SQL Server 的操作和管理,学习了配置 ASP 程序的运行环 境以及 ASP 的缺省脚本语言和内嵌对象后,读者将具备开发简单的 Web 数据库的能力。本 章将详细介绍笔者开发的 BBS 论坛的源程序的编写过程,从中介绍笔者的一些开发经验和 技巧。

第8章: 在很多时候需要在线管理一些文档,而不同的文档如果分不同的保密级别, 公开的对象也不同,这样的问题应如何解决?如何通过 Web 数据库系统来实现呢?本章将 详细介绍一个在线的文档管理系统。

第9章:介绍一个 Web 服务器内部的搜索引擎,讲述如何通过 Web 数据库的形式来实现。

第 10 章: 实用程序片断。本章从 3 个方面介绍程序段,即源代码说明、使用说明、结果显示。这对 Web 数据库开发者具有重要的参考价值。

本书的样例代码及相关文件均以自由下载方式提供,详细内容请参见本书附录 D。 由于时间仓促,书中错误和不妥之处在所难免,恳请读者批评指正。

目 录

第1篇 Web 数据库基础篇

第1章	We	əb 数据库基础3
1.1	Wet	o 数据库的概念
1.2	Web	o 数据库系统的发展历程5
1	.2.1	管理信息系统
1	.2.2	动态 Web 数据库的发展7
1.3	后台	3数据库选择10
1.4	操作	「系统配置
1.5	Web	•数据库的设计步骤12
第2章	BB	S 论坛实例
2.1	论坛	5介绍17
2	.1.1	用户登录17
2.	.1.2	本站精华
2.	.1.3	分类讨论区
2.	.1.4	修改个人资料
2.	.1.5	网友查询32
2.	1.6	个人信箱
2.	.1.7	聊天室
2.	1.8	离开本论坛
2.2	安全	≥性介绍
2.3	论坛	后的整体结构
2.4	论坛	动能比较
第3章	构到	建后台数据库
3.1	SQL	- Server 7.0 概述
3.2	SQL	- Server 7.0 的安装
3.	2.1	SQL Server 在不同操作系统上的安装
3.	2.2	安装 SQL Server 7.0 的系统要求
3.	2.3	SQL Server 7.0 的安装过程
3.	2.4	常见的安装问题
3.3	SQL	. Server 的使用
3.	3.1	SQL Server Service Manager
3.	3.2	SQL Server Enterprise Manager
3.	3.3	SQL Server Query Analyzer

3	.3.4	SQL Server 的其他工具	62
3	.3.5	SQL Server 使用小结	63
3.4	设ì	┼ SQL Server 数据库	63
3	.4.1	数据库设计	64
3	.4.2	设计 BBS 论坛后台数据库	66
3	.4.3	SQL Server 7.0 中的数据类型	71
3.5	建立	Σ数据库	80
3	.5.1	使用 SQL Server Enterprise Manager 建立数据库(硬操作)	
3	.5.2	建立数据库的 SQL 语句(软操作)	
3	.5.3	通过 SQL Server Enterprise Manager 操作数据表(硬操作)	
3	.5.4	使用 SQL 语句操作数据表(软操作)	
3	.5.5	两种建立数据库方法的比较	93
3.6	SQI	_ 语言	93
3	.6.1	使用 SELECT 语句进行数据库查询	94
3	.6.2	对数据表进行更新和删除操作	
3.7	管理	₤ SQL Server 数据库	105
3	.7.1	SQL Server 中可用的向导	
3	.7.2	数据库文件和事务日志文件的管理	
3	.7.3	数据库安全	
3	.7.4	建立 SQL Server 数据库用户	
3.8	数携	居备份	
3	.8.1	数据备份的几种基本方式	
3	.8.2	BBS 论坛数据备份策略	113
3	.8.3	备份及恢复过程说明	
第4章	AS	P 程序准备	120
4.1	配置	程 IIS	
4	.1.1	IIS 的概念	
4	.1.2	IIS 的软件要求	
4	.1.3	改变自定义错误信息	
4.2	ASI	? 开发工具介绍	
4.3	HT	ML 语法概要	
4	.3.1	表单 Form 的使用	
4	.3.2	表格 Table 的使用	
4	.3.3	超链接的目标属性 target	
第5章	VB	Script 编程基础	
		а. • , А /лт	

5.1 VB	lScript 介绍	. 138
5.2 VB	Script 中的变量	. 139
5.2.1	声明变量	. 139

4	5.2.2	变量命名规则	
5.3	数携		
5.4	VB		
5.5	VB	Script 运算符	
4	5.5.1	运算符优先级	
5.6	循玎	· · · · · · · · · · · · · · · · · · ·	
4	5.6.1	DoLoop 循环	
4	5.6.2	· WhileWend 语句	
4	5.6.3	ForNext 语句	
4	5.6.4	For EachNext 语句	
5.7	VB	Script 条件语句	
4	5.7.1	IfThenElse 语句	
4	5.7.2	Select Case 语句	
5.8	在、	/BScript 中使用对象	
4	5.8.1		
4	5.8.2	Function 过程	
4	5.8.3	过程的数据进出	
4	5.8.4	在代码中使用 Sub 和 Function 过程	152
5.9	VB	Script 函数	
4	5.9.1	数学函数	
5	5.9.2	格式化函数	
4	5.9.3	日期时间函数	
4	5.9.4	字符串操作函数	
4	5.9.5	变体子类型	
第6章	AS	P 对象	156
6.1	Req	uest 对象	156
e	5.1.1	使用 QueryString 集合	
e	5.1.2	使用 Form 集合	
6.2	使用	引 Response 对象	
e	5.2.1	Write 方法	
e	5.2.2	End 方法和 Clear 方法	
ć	5.2.3	Redirect 方法	
6.3	Ses	sion 对象	
e	5.3.1	Session 对象简介	
(5.3.2	Session 对象的属性	
(5.3.3	Session 事件	
6.4	Coo	kie 对象	169
ť	5.4.1	Cookie 的工作原理	
(5.4.2	创建多个 Cookie	

	6.5 A	pplication 对象	
	6.5.	1 Application 对象简介	
	6.5.2	2 使用 Application 对象	
	6.6 A	SP 数据库组件	
	6.6.	I ActiveX 数据对象概述	
	6.6.2	2 设置 ODBC 数据源	
	6.6.3	3 使用 ActiveX 数据对象对数据库进行存储操作	
	6.7 用	记录集显示记录	
	6.7.	1 记录集游标和锁定类型	
	6.7.2	2 操作记录集的高级方法	
	6.7.3	3 遍历记录集	
	6.7.4	4 返回记录数目	
	6.7.	5 对记录集进行分页	
	6.7.6	5 把记录取到数组中	
	6.7.3	7 指定记录集的最大容量	
	6.8 本	章小结	
第 7	章 B	BS 论坛系统	197
	7.1 登	₹录系统	
	7.1.1	Ⅰ 用户身份验证	
	7.1.2	2 用户注册系统	
	7.1.3	3 密码提示系统	
	7.2 本	站精华 site.asp	
	7.3 分	·类讨论区	
	7.4 资	料修改系统	
	7.5	友查询系统	
	7.6 个	·人信箱系统	
	7.7 密	。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。	
	7.8 离	5开论坛	

第2篇 Web 数据库应用实例

第8章	文	档资料管理系统	
8.1	管理	里系统的任务	
8.2	使用	用稿件管理系统	
8.3	稿件	牛信息数据库结构设计	
8	.3.1	数据表结构设计	270
8	.3.2	稿件信息数据库各数据表之间的关系	274
8.4	ASI	P 程序开发	
8	.4.1	程序开发准备	
8	.4.2	稿件信息数据库前台程序的编写	
第9章	搜	索引擎实例	305
9.1	概论	龙	305
9.2	搜索	 家引 擎 的使用	306
9	.2.1	搜索文件(关键词搜索)	
9	.2.2	文件夹搜索	
9	.2.3	所有文件搜索列表	
9	.2.4	dido 搜索引擎总体结构图	
9.3	DID	OO 搜索引擎的开发	
9	.3.1	数据库设计	
9	.3.2	前台程序设计	
第 10 章	实	5月程序片断	335
10.1	计	数器	
1	0.1.1	代码说明	
10.2	地:	址重新定向	
10.3	显	示当前文件目录	
10.4	列	举浏览器中的 cookie	
10.5	留	言簿	
10.6	投	票程序	
10.7	随	机图片显示	
10.8	表	单限制	
10.9	页	面刷新程序	353
附录A	SQ	化语言	356
附录 B	We	eb 数据库网络资源	369
附录 C	SQ	ℓL 中的保留字	376
附录 D	文作	件下载使用说明	378

第1篇 Web 数据库基础篇

20 世纪 90 年代开始兴起的国际互联网,极大地改变了我们的生活方式。它不仅摆脱 了传统的空间概念,使得地理上相隔遥远的双方可以轻而易举地相互传递数据和信息。也 使得不计其数的各种各样的网站如雨后春笋般地不断涌现,从网上教育到在线娱乐,从新 闻资讯到电子商务,几乎无所不包。

但是,一个网站仅仅拥有简单的静态主页是远远不够的。互联网上任何一个成功的站 点都必然拥有一个或多个强大的后台数据库资源。网络上浩如烟海的数据库资源无疑是网 络如此令人趋之若鹜的重要原因之一。因此,建立一个网络数据库对于一个成功的网站而 言是极其重要的。

任何一个功能完善的网络数据库都需要各个方面的技术支持。尽管构建一个网络数据 库的技术方案有很多种,但基本上都包括前台和后台两方面的技术。在众多的构成方式中, 我们选用的是当前最流行的 ASP+SQL Server 形式,也就是前台采用浏览器形式,后台数 据库采用 Microsoft SQL Server 7.0,我们暂且把这种方式称为 Web 数据库。作为 Web 数据 库开发人员,在开发出一套具有某种功能的 Web 数据库系统之前,首先必须熟练掌握各个 方面的技术。

本书的第1篇将通过一个完整的 BBS 论坛讲述 Web 数据库的构建技术,从中你可以 学到以下几个方面的知识:

- 1. 一个完整 Web 数据库的开发设计步骤
- 2. 后台数据库的建立和管理
 - ◆ SQL Server 7.0 的安装与管理
 - ◆ 数据库表的建立
 - ◆ SQL 语句的使用
 - ◆ 数据库的管理
 - ◆ 数据的备份
- 3. ASP 程序准备
 - ◆ 配置 IIS
 - ♦ ASP 开发工具
 - ♦ HTML 语法概要
 - ◆ ODBC 数据源的要领与设定
- 4. Active Server Page 技术
 - ◆ ASP 简介
 - ◆ ASP 脚本语言 VBScript

- ◆ ASP 中的对象
- 5. BBS 论坛设计与开发
 - ◆ 总体设计
 - ◆ 功能介绍
 - ◆ 程序解释

第1章 Web 数据库基础

随着计算机技术的不断发展和进步,网上浏览者的要求也在不断提高,为了适应这种 变化,任何一个网站都不再局限于仅发布静态的页面,而是开发动态的交互式页面,使网 上浏览者得到更加完善、快捷的服务。

目前,大多数网站不同程度都有后台的数据库支持,用来存放用户和服务器数据,如 果再加上一定的前台程序,就能构成一个完整的数据存储、查询系统。由于大多数的网上 冲浪者几乎都是通过客户端的浏览器来获取信息,访问静态或动态的页面,或者访问远程 数据库,所以,大多数系统都是基于浏览器的,这种系统我们称为 Web 数据库系统。

本章主要介绍 Web 数据库的一些基础知识:

- ◆ Web 数据库的概念
- ◆ Web 数据库系统的发展历程
- ◆ 后台数据库的选择
- ◆ 操作系统配置
- ◆ Web 数据库的设计步骤

1.1 Web 数据库的概念

Web 数据库目前在因特网上有大量的应用,比如,网站的留言簿、自由论坛、网上商 店、网上考试、虚拟人才市场等等,这些系统几乎都是采用 Web 数据库这种方式来实现。 使用 Web 数据库来开发系统的最大优势就是它的所谓"瘦客户"方式,也就是说,用户无 需在自己的客户端上安装任何与所需存取或操作的数据库系统对应的客户端软件,只需通 过 Web 浏览器(如微软公司的 Internet Explorer 和网景公司的 Netscape Communicator)便 可完成对数据库数据的常用操作。这种方式的优点是用户不必再去学习复杂的数据库知识 和数据库软件的使用方法,只需掌握基本的网络操作,如填写、提交表单等(这对绝大多 数网络用户来说是不困难的)就可以轻而易举地从任何一台可以连接因特网的计算机上来 访问数据库。

正是因为这种优势所在,开发 Web 数据库越来越成为一种需要:组建一个网站几乎必须有自己的数据库系统,用来存放重要的数据和信息,以供必要的时候加以调用。另外, 采用这种 Web 数据库方式可以大大提高服务器系统维护者的工作效率。比如,一个新闻发 布网页,如果采用手工方式的话,当有新的新闻信息时,系统维护者必须亲自去改动前台 的 Web 页面,不但繁琐而且跟不上信息的更新速度。如果编写一段前台的应用程序,并把 它和后台数据库联系起来,每天系统自动把新闻信息存入后台数据库中,当用户浏览时, 又会自动从后台数据库中调出,自动实时更新信息,而这一切都是系统自动完成的,不仅 方便了用户,也给系统维护者节省了大量的工作。

一般的 Web 数据库运行方式如图 1-1 所示,从图中可以看出组成一个 Web 数据库的必要硬件元素:后台数据库、Web 服务器、客户端浏览器以及联系客户端和服务器之间的网络(Internet 或 Intranet,也就是因特网或者局域网)。这些元素只不过是 Web 数据库存在的基础,而构建硬件环境并不是 Web 数据库系统开发者的任务,他要考虑的重要问题是,运行一个 Web 数据库系统所必须的软件要求。



图 1-1 Web 数据库示意图

注意: 图 1-1 中的后台数据库可以安装在 Web 服务器上或者使用专门的数据库 服务器,后台数据库的类型很多,比如 Microsoft Access、Microsoft SQL Server、 Oracle 8i 等等。在本书中,Web 数据库的后台数据库类型都是采用 Microsoft SQL Server 7.0,因此在介绍后台数据库时,都以 SQL Server 7.0 为例来说明,至 于选择它的原因可以参考第3章中的介绍。

正如上面所提到的一样,Web数据库采用的是一种"瘦客户"方式,客户端不用安装 任何专用的软件,只需要有平常网上冲浪所必需的Web浏览器就可以了。有一点Web常 识的读者都知道,Web浏览器接收的其实只是文本形式的HTML代码(包括一些动态脚本 语言),通过逐条解释HTML代码来组织各种文本和多媒体对象,以一定的方式显示在客 户端的浏览器界面上,传递信息给客户。

换一个角度说,如果客户端只有浏览器的话,Web 服务器只能相应发出文本形式的 HTML 代码(包括一些动态脚本语言),其他所有类型的代码是无法在客户端执行的。要 实现动态的、交互的浏览形式,Web 服务器就必须具有动态生成 HTML 代码的能力。

Web 数据库系统的运行方式如图 1-2 所示。



图 1-2 Web 数据库的运行方式

从图中可以看出用户通过客户端浏览器来访问一个 Web 数据库系统的过程。客户端的 浏览器首先发出请求(比如,在浏览器的地址栏中敲入某一个 Web 数据库系统的登录网页 地址),服务器根据请求向客户端发回相应的 HTML 文件,这时客户端将显示出这一 Web 数据库系统的登录界面,假设在这个登录界面上需要用户输入用户名和密码,则当用户输 入完毕并选择提交之后,信息将通过因特网(或者局域网)传递到服务器端,这时服务器 端的相应程序启动并自动执行,向后台的数据库发出相关的数据操作指令,后台数据库的 引擎将执行该指令,进行相应的数据存储或查询操作,并将结果返回到服务器的运行程序, 这时服务器端的程序根据从后台数据库返回的数据进行判断,生成相应的 HTML 文件,并 通过因特网(或者局域网)返回到客户端的浏览器,客户将看到相应的信息。在上面的例 子中,如果客户所输入的用户名和密码正确,也就是说,在后台的数据库中存在,该客户 的注册信息将会得到登录成功的信息。

从上面所述的 Web 数据库运行方式中,可以看出运行一个完善的 Web 数据库系统所 需要的软件元素是客户端必须有能够解释执行 HTML 代码的浏览器(如微软公司的 Internet Explorer 和网景公司的 Netscape Communicator)。在 Web 服务器上必须有完成一定功能并 自动生成 HTML 代码的程序,这是 Web 数据库系统开发者的主要工作,需要开发者来编 写程序,占相当大的工作量。另外,后台数据库需要能够自动完成数据操作的指令(存储 和查询等),这对于 Microsoft SQL Server 7.0 来说是完全可以胜任的。

可以看出,编写服务器端应用程序在开发 Web 数据库系统中有很重要的地位,应用程序的效果直接影响对客户的服务质量和服务方式。这样的应用程序必须具有以下几项功能:

- ◆ 建立与后台数据库的联系
- ◆ 能够给后台数据库发出数据操作指令
- ◆ 能够对客户发出的请求作出响应
- ◆ 能够处理客户反馈的数据
- ◆ 能够处理从数据库返回的信息

1.2 Web 数据库系统的发展历程

由前一节知道,服务器端应用程序实际上起到一个桥梁作用,使客户和后台数据库之间建立联系,交互数据和信息。狭义上的 Web 数据库系统就是指这样的服务器端应用程序。

在本节中所讲的 Web 数据库系统实际上是狭义的。

1.2.1 管理信息系统

讲述 Web 数据库系统的发展历程,就必须提到管理信息系统 MIS,因为最初的 Web 数据库系统就是从管理信息系统发展而来的。

MIS(Management Information System) 是管理信息系统的简称,它的发展主要经过两个阶段: Client/Server(客户机/服务器,简称 C/S)模式和 Browser/Server(浏览器/服务器,简称 B/S)模式。

1. Client/Server(客户机/服务器)模式

以前的 MIS 绝大部分是采用 C/S 模式, 它建立在局域网基础上, 采用安全性能比较好的网络协议, 如图 1-3 所示。



图 1-3 客户机/服务器模式

客户机主要由 PC 或工作站组成。客户端需要安装完整的用户界面程序,以加强服务 器和客户端之间的交互性。但是由于此应用程序与所用的操作系统平台有关,所以不适合 于在多平台环境下运行,而管理人员需要花费大量的时间和精力开发和维护。

随着 Web 技术的广泛应用, C/S 模式已无法满足人们的需要,而且传递单向信息的静态 Web 页面无法提供充分的交互功能,供用户查询甚至更新。同时企业也无法将企业动态 信息及时发布到网络上,这就导致出现了 B/S 模式的 MIS。

2. Browser/Server(浏览器/服务器)模式

采用 B/S 模式后,在客户端只需安装一个通用的浏览器就可以了,同时 B/S 的大部分 功能都在服务器上实现,故大大降低了日常维护的工作量。并且,由于客户端不必安装用 户界面程序,用户的操作就变得异常简单,也就是说,用户只要会用浏览器就行。

由于该模式基于 Internet 和 Intranet,采用 TCP/IP 协议,使得信息能够很方便地在网上 发布,故服务的领域不再局限于局域网内,而可以通过因特网和世界各地建立联系。

1.2.2 动态 Web 数据库的发展

动态 Web 数据库是本书讲述的重点,它的发展也经历了好几个阶段。

1. CGI 技术

CGI 是 Common Gateway Interface 的简称,意即公共网关接口,是外部应用程序与 Web 服务器之间的接口标准,利用它可使数据库与 WWW 集成,实现动态的 Web 数据库应用。 CGI 的应用流程如图 1-4 所示。



图 1-4 CGI 应用流程

客户端通过 IE 等浏览器在表格中输入查询条件(或其他要求)并提交给 Web 服务器; 服务器将查询条件及客户信息置于标准输入或一组环境变量中,并运行相应的 CGI 程序; CGI 程序依据标准输入或环境变量向数据库服务器提出查询要求,执行相应的 SQL 语句; 数据库服务器返回查询结果给 Web 服务器,并通过 HTML 页面返回至客户端的浏览器。

使用诸如 VB、C/C++、PERL 等语言都可以编写 CGI 程序,由于 CGI 比较通用,所 以在 DOS、Windows、Windows NT、UNIX 等操作系统下都可以运行。但是,CGI 程序的 开发需要一些底层开发经验,故难度较大。而且,CGI 一般都是一个独立的可执行程序, 每当客户端输入一个请求时,就必须激活一个 CGI 进程,这样用户数量的增加必然大大剥 夺系统的资源,使得 Web 服务器的效率降低。

目前,CGI 技术最为成熟,历史最为悠久,并且也得到了广泛的应用,但其效率低、 编程难一直是其应用的最大困难。

2. IDC 技术

IDC(Internet Database Connector, Internet 数据库连接器)是微软提供的一个 API (Application Program Interface,应用程序接口),实质上是微软的 Web 服务器 IIS(Internet Information Server)的一个 ISAPI(Internet Server API)应用程序 httpodbc.dll。它可通过 ODBC(Open Database Connectivity,开放数据库连接)接口访问各种各样的数据库。

IDC 使用两种文件来控制数据库操作和返回 HTML 页面。一种是.idc 文件,它包括连接 ODBC 数据源的信息、需要执行的 SQL 语句和.htx 文件名和路径。另一种就是.htx 文件,它相当于一个 HTML 模板,依照 IDC 执行的 SQL 语句的查询结果来构造出一个 HTML 文件返回给用户的浏览器。

IDC 应用流程如图 1-5 所示。



图 1-5 IDC 应用流程

IDC 是针对 CGI 的低效率而提出的一种基于 ISAPI 的动态 Web 数据库应用技术。因 其以动态链接库(DLL)的形式存在,在启动时被加入内存,和 Web 服务器处于同一进程 空间,故占用系统资源少,执行效率高。另外,IDC 的编程相当容易,只需掌握一定的 HTML 语言和 SQL 语言就可以设计出良好的.htx 和.idc 文件,适用于简单的 Web 数据库应用开发。

但由于 IDC 基于微软的 IIS,因此跨平台性相当差。

3. ASP 技术

ASP (Active Server Page,动态服务器页面) 是基于 Mircosoft NT 4.0 和 IIS 3.0 以上版 本或 Personal Web Server on Win9X 的开放式脚本开发环境,是最新的动态 Web 数据库应 用开发方案。利用它可以实现复杂的 Web 应用,开发动态、高效和交互的 Web 服务器端 应用程序。它具有以下特点:

- ◆ 完全与 HTML 相融合,使得开发环境变得相当容易,而且灵活性得以提高。
- ♦ ASP 能够利用任何一种脚本语言,如 VBScript(默认脚本语言)和 JavaScript 等。
- ◆ ASP 的制作工具可用纯文本编辑器(如 Notepad),但专业开发人员一般使用 Microsoft Visual InterDev 6.0。
- ◆ ASP 是文本文件, 扩展名为.asp, 包括文本、HTML 标记和 ASP 文件特有的脚本 语句 (用<%和%>括起来)。
- ◆ 支持面向对象。
- ◆ 利用 ASP 开发出的脚本程序运行在服务器端,所以它对浏览器种类没有限制,而 且也防止了源程序被不良动机者偷看,提高了安全性。
- ◆ 支持多用户、多线程。
- ◆ ASP 采用 ADO (ActiveX Data Objects) 组件,通过 ODBC 驱动程序连接数据库。

如图 1-6 所示, ASP 的流程为: IE 浏览器向 Web 服务器提交查询, 申请访问一个 ASP 文件; Web 服务器调用 ASP 文件并执行; 用 ADO 组件通过 ODBC 访问数据库, 将结果传递到用户的浏览器。



图 1-6 ASP 应用流程

ASP 是一种十分具有发展前途的动态数据库应用技术。在许多网站上都能看到。ASP 提供了五个内嵌对象: Request, Response, Server, Session 和 Application, 相当于已经封装好了的类,它为用户访问各种资源提供了极大的方便。

正是由于前面所述的优点,ASP 在推出后短短一年多的时间内,就当仁不让地抢占了 开发 Web 应用程序的主导地位。如今,ASP 已成为 Web 应用程序开发的主流环境。据统 计,当今世界上有将近半数的 Web 站点采用了 ASP 技术,而且这一比例还在不断地扩大。

基于以上原因,我们几乎没有理由不采用 ASP 技术来实现 Web 数据库的动态交互。 因此,ASP 技术也是本书中开发 Web 数据库系统的主要工具。

VBScript 是 ASP 默认的脚本语言,尽管它是由微软倡导推出的而得到了大多数开发者 的青睐,但也恰恰是这个原因造成了麻烦。当今的浏览器市场主要由 Netscape 的 Navigator 和微软的 IE 所垄断,虽然 IE 对 VBScript 和 JavaScript 同时支持,但 Navigator 却不支持 VBScript。因此,在客户端浏览器不统一或不知道的情况下,我们在开发 Web 页面编写非 ASP 的脚本时,理所当然地更倾向于使用不依赖于浏览器平台的 JavaScript 作为脚本语言。 而 ASP 脚本由于是在服务器端执行的,不存在浏览器的兼容问题,所以编写时究竟是采用 VBScript 还是 JavaScript 作脚本语言是无关紧要的。在本书中,所编写的程序实例都是采 用 VBScript 来完成的。

为了进一步说明 ASP 程序的运行原理,在这里举一个简单的例子,如图 1-7 所示。



图 1-7 ASP 运行原理

图中所列的 ASP 文件是一个最简单的 ASP 文件,其中只有一条 ASP 语句:

<% response.write "您好" %>

这里的 response 是 ASP 中的一个内嵌对象,具体解释见 6.2 节"使用 Response 对象"。 当服务器端执行这个 ASP 文件时,将自动生成 HTML 代码,如图中所列,那也是一个最 简单的 HTML 代码,通过浏览器执行将得到一个"您好"的信息。

利用 ASP 可以开发高效、快捷和安全的服务器端应用程序,可以实现复杂的 Web 应用,这是目前最为流行的 Web 数据库开发方式,也是本书所要讲述的主要内容之一。

1.3 后台数据库选择

要开发一个 Web 数据库当然离不开后台数据库的支持,在数据库的选择上,不同的要求可以选择不同的数据库来实现。

数据库技术发展到今天,关系数据库已经完全占据了统治地位。世界上最为流行的关系数据库管理系统(Database Management System, DBMS)主要由两家巨头公司垄断: Oracle 和 Microsoft。

在开发数据库软件领域,Oracle 公司无疑是业内的老大。如今该公司最重要的主导产 品是 Oracle 8i,号称全球第一套基于 Internet 的关系数据库管理系统。1999 年在全球的 DBMS 产品销售市场中,Oracle 占据了几乎 50%的市场份额。但是,Oracle 的 DBMS 产品 主要面向大型的企业级数据库应用,它的高效和强大功能使得它的售价几乎是同类产品中 最高的。对于实现一些中小型的 Web 数据库系统而言,其昂贵的价格和管理、使用的复杂 性是不切实际的。

Microsoft 公司是世界公认的应用软件开发霸主,其关系数据库软件产品主要有 Access 和 SQL Server。

Access 是微软的 Office 系列办公软件家族的一员,它自 1992 年发布以来,已逐步成为桌面数据库领域的领导产品,拥有极其广泛的用户。其主要优点是学习容易,使用简单。 虽然只是一个桌面型的 DBMS 产品,却具有面向对象的可视化设计,能够利用 Web 来进 行检索和发布数据,实现了与 Internet 的连接。

毕竟 Access 只是一种针对桌面办公系统设计的数据库产品,其对网络检索和操作的支持是有限的。首先,它最多只能支持四个并发用户进行存取;其次,它最大只能支持 2GB 大小的数据库,这对于一些需要大量数据存储的系统显然是不够的。所以,人们把目光投向了 SQL Server。SQL Server 几乎具有开发 Web 数据库系统所需要的全部优点:

- ◆ 更高的性能和分时性。很多情况下, SQL Server 能提供比 Access 数据库更好的性能。并且,在 Windows NT 的支持下, SQL Server 可以极为高效地并行处理查询(在处理用户请求的单个进程中使用多个本地线程),同时也将添加更多用户时的附加内存需要量降低至最小。
- ◆ 提高了可用性。使用 SQL Server,可以在正在使用数据库时对它进行递增的或完 全的动态备份。因此,不必强迫用户为了备份数据库而退出数据库。也就是说, 数据库可以每日每夜不间断地运行。
- ◆ 改进的安全性。SQL Server 集成了 Windows NT 操作系统的安全性,为网络和数

据库提供同一个登录过程。这使得能够管理复杂的安全方案。服务器上的一个 SQL Server 数据库也更好地被保护起来,因为未授权的用户不能直接访问数据库 文件,而必须先访问服务器。

- ◆ 即时的可恢复性。当系统出现故障时(例如操作系统崩溃或电源突然断电等), SQL Server 具有一个自动恢复机制,可以在几分钟内将数据库恢复到一致性的最后状态,并且不需要数据库管理员干预。
- ◆ 可靠的发布数据和事务。对于支持要求严格应用程序的系统而言,事务处理是很 重要的,例如银行系统和联机订货输入系统。SQL Server 通过事务日志支持最小 的事务,这样就保证了在事务中进行的所有更改要么提交要么恢复。即使在系统 出现故障时,以及多个用户正在进行复杂更新时,也能保证数据库事务的一致性 和可恢复性。SQL Server 将一个事务中的所有数据库更改都当作单个的工作单 元。根据定义,要么安全地完成整个事务,并且在数据库中体现所有完成的更新, 要么恢复该事务,撤销对数据库进行的所有更改。
- ◆ 基于服务器的处理能力。微软从一开始就把 SQL Server 设计为客户/服务器数据 库。数据和索引保存在单个服务器计算机上,很多客户计算机通过网络经常访问 这个服务器计算机。SQL Server 通过将结果发送给客户机之前在服务器上处理数 据库查询,减少了网络通讯量。这样,客户/服务器应用程序在最佳的位置──服 务器上运行。
- ◆ 经济划算。SQL Server 比 Oracle 要便宜得多,尽管它的性能也许不如 Oracle 出色, 但对于中小型企业数据库而言,它已经完全能够满足用户的需要。因此,在本书 中所讲述的所有开发实例都是基于 SQL Server 数据库的。

目前 SQL Server 的最新版本是 7.0 版,随着 SQL Server 7.0 的发行,使得 SQL Server 的功能更加强大,几乎可以满足所有用户的需求,而且可以在 NT Server 和 Windows 9x 上运行,非常通用。

采用 ASP+SQL Server 的方式来开发 Web 数据库无疑是目前最为流行的一种方式,具 有高效、快捷、安全等特性,而且开发人员完全是使用上层语言来开发程序,不需要了解 任何关于底层的技术,完全可以实现复杂的 Web 应用,实现用户和数据库的快速交互,是 一种最可选的方式。

1.4 操作系统配置

选择好开发的技术和数据库系统之后,这些系统都必须运行在一定的操作系统环境 中。开发 Web 数据库系统必须对操作系统有一定的了解。虽然几乎在所有的操作系统上都 可以构建 Web 数据库系统,但不同操作系统上的 Web 数据库在适用性和安全性等方面都 有很大的差别。

一般的数据库系统中都存储有很重要的数据,这些数据的安全是开发者必须考虑的问题。因此,一般都选择 Windows NT Server。笔者在这里使用的是 Windows 2000 Server 操

作系统,也就是 NT 5.0 版本 (见图 1-8)。由于 SQL Server 和 NT Server 采用无缝连接,使 系统的数据库具有更高的安全性。对于 NT Server,要完全理解并掌握它,并不是一件容易 的事情。关于 NT Server 不是本书要介绍的重点,你可以参考其他有关 NT Server 的书籍。 因为在开发 ASP 程序和安装使用 SQL Server 数据库时,必须对 NT Server 有一定的了解, 熟悉其中的一些操作。



图 1-8 Windows 2000 Server 管理工具程序组

1.5 Web 数据库的设计步骤

要开发一个功能完善的 Web 数据库,并发布到因特网上供用户访问,这是一项很复杂的工作,需要有完善的设计步骤,分阶段进行。首先,必须掌握 Web 数据库涉及到的各项 技术。Web 数据库系统的开发人员必须认识到,开发一个 Web 数据库系统远比简单地编一 段程序复杂。

因为开发一个系统,往往需要几个开发者互相合作,最终完成一个系统的开发。在具体开发之前,必须做好规划工作,需要有前期的调查,做好计划和分工,明白开发的系统应该具备什么样的功能,有什么样的安全防护措施等等,在时间和人员上做好安排和规划。

一般的 Web 数据库系统开发步骤如图 1-9 所示。



图 1-9 Web 数据库的设计步骤

如图 1-9 所示, Web 数据库的开发可以分为三个步骤:前期准备,开发阶段以及测试 与维护。在每一个阶段又有不同的设计步骤。这些步骤是开发一个 Web 数据库系统的一般 步骤,其中的每一步都不可缺少。下面逐一介绍每一步骤中应该完成的工作。

1. 前期准备阶段

这一阶段中虽然工作量不是很大,但是对后面的开发步骤有着重要的指导意义,如果 规划的不完善,就容易造成后面的开发混乱,影响工作进展和工作效率。

(1) 前期调查、整体规划

这一步骤的工作往往需要一个项目的主管或者项目的策划者来完成,首先确定一个项目,然后进行相关的调查。比如本书所列的 BBS 论坛开发实例,在得到这个项目之后,首先应该做一些相关的调查,列出一个基于 Web 数据库系统的 BBS 论坛应该具有的功能,也可以参照使用 Telnet 访问的 BBS 方式,找出:

◆ 哪些功能是一个 BBS 系统必须具有的;

- ◆ 哪些功能是可选择的;
- ◆ 哪些功能需要改进;
- ◆ 需要新添哪些功能;

◆ 开发的 BBS 系统将来用于何种需要,具有哪一些功能才可以满足用户的需要。

定出了这些功能,相当于制定了项目的任务书,也就是要最终达到的工作目标。当然, 这些初步设计的功能,在实际的开发过程中根据开发情况可以做一定的调整,但大体上是 遵循初步设计的。

确定了工作目标之后,需要对整个项目作一个整体规划,甚至作出一个时间进度表,确定工作中的每一阶段应该完成的任务,安排好各项工作的先后顺序,排出最优的工作顺序。

(2) 配置硬件、安装软件

作好整个项目的规划之后,就可以进行具体工作了。首先需要构造工作环境,否则不可能开始真正的工作。

硬件的配置一般不需要 Web 数据库系统的开发人员来完成,一个固定的网站,它的硬件和网络环境是固定的,必要时可进行系统的升级。比如,运行一个 SQL Server 数据库对系统有一定的要求(具体要求见第3章),系统配置太低的话,无法正常安装 SQL Server,也就无法进行工作,而且要作为网络中的 Web 服务器,要提高用户的访问速度的话,应该选择更高级的硬件系统,提供更多的系统资源。

有了硬件环境之后,还需要安装相应的软件。首先,一个稳定、高效的操作系统是必须的,大多数网络 Web 服务器都采用 Windows NT Server,当然也有其他的操作系统。但 这不是本书讨论的主要问题,本书中的所有实例都是在 Windows 2000 Server (NT5.0)环 境中开发的,所以首先应该在服务器上安装 NT Server 5.0 操作系统。

安装好操作系统后,要根据前台程序的类型来安装必要的开发工具和执行软件。本书 的实例都是采用 ASP 编写的, IIS 4.0 以上的版本都可以运行 ASP 程序,而在安装 Windows 2000 Server 时,系统已经默认安装了 IIS 5.0,所以这里不需要再安装。有了程序代码的运 行软件,对于开发者来说,一个功能强大的编辑软件是必须的,虽然 Windows 自带的记事 本就可以用来编写程序代码,但由于功能单一,不能满足开发者的要求。笔者在开发实例 时,采用的是 Microsoft FrontPage 2000 开发工具,因此需要首先安装此开发工具。

安装好前台工具之后,接下来就是安装后台的数据库软件了。开发者可以根据自己的 实际需要来选择相应的数据库软件,本书中的实例都是采用 Microsoft SQL Server 7.0,所 以也需要安装。有关 SQL Server 的安装和使用,请参见第3章"构建后台数据库"。

2. 开发阶段

在构建好硬件和软件环境之后,就可以开始具体的开发工作。从图 1-9 中可以看出, Web 数据库系统的开发阶段主要分为后台数据库设计和前台程序的编写调试。这两项工作 需要相互协调,因为后台的数据库是前台程序的数据来源,而前台程序起到后台数据库与 客户端之间的桥梁作用。为了达到最优化的设计,有时需要对后台数据库和前台程序同时 进行调整。 (1) 后台数据库整体设计

所谓数据库的设计就是要确定数据库所要完成的任务以及如何去完成,也就是确定信息数据以何种方式来存放。必须明确希望从设计的数据库中得到什么信息,由此可以确定 需要用什么主题来保存相关事件(对应于数据库中的表)和需要什么事件来保存每一个主题(对应于数据库中的字段)。

只有设计好数据库的整体结构(表与字段)之后,才能进行相关的应用程序开发和调 试。

(2) 前台程序的开发调试

前台应用程序主要起一个桥梁作用,也必须具有一定的安全保护功能,能够对不同的 用户进行判断,以不同的方式来对待,从而保护数据库中的数据不受破坏。前台程序的开 发是一个 Web 数据库系统开发中最重要也是工作量最大的环节,几乎大部分工作量都集中 在前台程序的开发和调试上。本书中的前台程序都是采用 ASP 编写的,虽然 ASP 是一种 内嵌的脚本语言,但是在某种程度上和其他的(例如 C++等等)程序语言在编写风格上是 相似的。

在使用 ASP 编写前台程序时,应该注意以下几点:

- ◆ 有清晰的设计思路和良好的设计风格,程序代码要简洁、紧凑。
- ◆ 建立最优化思想,尽量减少程序中不必要的代码。
- ◆ 对于复杂的程序段需要画出流程图。
- ◆ 考虑不同类型的访问者,保证合法用户能够正常使用,而又能最大限度地限制非 法用户的访问请求。
- ◆ 编写代码时有良好的风格,程序代码要整齐,层次分明,有助于查找错误的程序 代码,提高工作效率。

3. 测试与维护阶段

一个 Web 数据库系统开发完毕之后,并不能马上发布到因特网上供用户访问,还需要 一定阶段的测试。

(1) 后期测试、完善漏洞

进行后期测试是为了找出程序中的漏洞,因为在开发程序代码的过程中不可能完全考虑到所有可能的情况,而一旦系统开始运行,发现问题再进行补救的话,会造成一定的损失,并影响用户的访问。所以,应尽量在正式运行之前找出所有的漏洞,这时一般可以把 系统发布在一个小的局域网内,供一些特殊的指定用户访问,以完成测试。

(2)发布站点、正常运行

通过测试阶段之后,程序得到了进一步的完善,可以发布到因特网上,供所有的用户 访问。在这一阶段需要做一定的宣传工作,以提高系统的访问量,并严密监视系统的运行 状况,一旦出现问题,马上给予解决。当系统正常运转一段时间之后,就可以稍微放松对 系统的监视,改为定期的查看和观察。 (3) 日常维护、数据管理

当 Web 数据库系统开始正常运转之后,还需要定期的管理与维护。因为后台数据库中存储有大量的数据信息,一旦破坏就很难再恢复,所以要定期进行数据的备份,以防人为或自然的破坏。

本节叙述了一个完整 Web 数据库系统的设计开发过程,可以看出,在设计过程中必须 遵循一定的原则和要求,建立最优化的思想,以最少的工作量得到最优的成果。在 Web 数 据库系统正常运转之后,仍然需要定期的维护与数据备份。

第2章 BBS 论坛实例

第1章介绍了 Web 数据库的基本知识,说明了开发 Web 数据库应该掌握的技术和知识。 在具体介绍各方面的技术之前,本章首先介绍一个笔者自己开发的 BBS 论坛,展示一个完 整 BBS 论坛的全貌,阐述一个 BBS 论坛应该具有的功能,以及如何使用这个论坛,比如 用户登录、个人资料修改、阅读或者查询文章、发表文章等等,从而说明开发一个真正 Web 数据库应该注意和考虑的方面。

在后面的章节中,对开发技术的介绍都是以这个 BBS 论坛中的代码为实例来说明的。 希望读者不要仅仅以一个使用者的身份来看待这个论坛,而是以一个开发者的身份来仔细 考虑论坛中涉及到的各个环节。假想如果自己要开发这样的一个论坛,要达到这样的功能, 需要学习哪些方面的技术,可以采用什么样的方案,哪种方案是最实用又最安全的等等。 在阅读本章的过程中,或许读者会产生很多疑问,带着这些疑问去学习后面章节中的知识, 就可以更加迅速地掌握。这也是为什么在介绍基础知识之前,首先介绍这个 BBS 论坛。

2.1 论坛介绍

经常上网的读者一定有使用 CTERM 登录一些 BBS 站点的经历,比如,目前全国最大的 BBS 站点水木清华。其中提供了各项功能,比如精华讨论区、分类讨论区、个人工具箱、个人信箱、聊天室等等。你有没有想过自己也可以开发这样的一个系统呢? 使用 ASP 和后台的 SQL Server 数据库,几乎完全可以实现这些功能。当然,由于构成的方式不同,实现的技术也不同,功能上不可能完全相同。

在这一节中,你将看到一个具有完整功能的 BBS 论坛全貌,这个论坛是笔者花了相当 长一段时间,进行不断改进和完善得到的最终成果。由于还没有向读者介绍任何关于构成 这个论坛的具体技术问题,所以这里只介绍这个论坛的功能和使用方法,而不介绍具体的 技术构建。在后面的第7章中,会结合这里的实例对涉及到的技术细节进行详细说明。

2.1.1 用户登录

当这个论坛发布到因特网上后,任何一位可以连接的用户都可以登录这个论坛,使用 者只需在自己的浏览器地址栏中输入相应的地址就可以了。这里,因为所有的调试都是在 本地机上完成的,所以在后面的图例中读者看到的只是一个主机名(http://wjy)而不是一 个完整的 URL 地址,但这里显示的和远程访问的用户看到的是完全一样的。这里的所有文 件都存放在 Web 服务器发布目录中的/bbs 子目录下,所以在地址后面要输入相应的子目录 名称和相应的文件(http://wjy/bbs/login.htm),如图 2-1 所示。

3 9 . Microsoft hab	met Ephrei		Part of the
R	and the second second	g un	
今后間・今~日	CONTRACTOR		Classification and
http://whyft	absflogin.htm		
1			2
			्य
	ab var b	JA DDO A I-	્સ
	欢迎九	。他pert-BBS论坛	
			× 1
			្រី
	田古夕		÷.]
	用户名	patt	2
	郡福	***	4.
			• *
		•	
		A F B B B	
		本码提示	j.ee
		<u>五 日本小</u>	
		新用户注册	<u>ب</u>
			ž.
		la de la companya de	

图 2-1 用户登录界面(login.htm)

这里的 login.htm 是这个 BBS 论坛的主登录界面,所有的用户都必须通过这里登录到 论坛,进一步来享受各个方面的服务。

如图 2-1 所示,在这里用户有三个选择:

- ◆ 如果你是一位老用户,那只需要在如图 2-1 所示相应的文本框中输入"用户名" 和"密码",然后就可以登录到本论坛。
- ◆ 如果你是一位老用户,但是不小心忘记了密码,这里提供密码提示的功能,只需要选择下面的"密码提示"链接就可以了,相应的提示问题和答案是在用户注册时由用户自己设置的。
- ◆ 如果你是一位新用户,那么你必须经过相应的注册才能登录到本论坛,这里选择 相应的"新用户注册"链接就可以了。
- 1. 老用户登录

首先来看老用户的登录过程: 必须在如图 2-1 所示相应的文本框中输入"用户名"和 "密码", 然后单击下面的"登录"按钮; 如果输入错误, 可以单击"重写"按钮来清空文 本框中的内容; 如果没有在文本框中输入任何内容, 单击"登录"按钮时会弹出如图 2-2 所示的警告对话框。



图 2-2 警告对话框

此时必须单击"确定"按钮,在相应的文本框中输入完整的内容。

当然,如果"用户名"和"密码"都输入了,单击"登录"按钮,将不会弹出如图 2-2 所示的警告框,而是登录到论坛的主菜单 choice.asp,如图 2-3 所示。这说明在登录页面中输入"pett"的用户已经成功登录到本论坛,可以进一步选择相应的服务。

●读作集中 Microsoft Internet Explorer	
http://wjy/bbs/choice.asp	🔤 🕅 Minister and Tanana and T
PETT-BBS会传-主菜单	
欢迎您,亲爱的pe	:tt,这是您第35次访问本站
主义年	đ.Գ.
	大汉县 本社会会人类记忆区 御司到周
<u>分类讨论区</u>	在这里是年始的音(历史时论区)派与论风 读和发表文章:
<u>查询网友</u>	在这里想可以查词任何一位已经在本站注册 的用户-
<u>聊天室</u>	这里是本站的事天室,在这里您可以畅所欲 言,结识更多的朋友:

图 2-3 论坛主菜单(choice.asp)

从图 2-3 中可以看到本 BBS 论坛的主菜单,在窗口的最上方显示了对用户的欢迎词和 用户访问本站的次数。同时显示了各个子菜单:本站精华、分类讨论区、修改个人资料、 查询网友、个人信箱(有无新邮件),聊天室和离开本论坛。用户根据自己的需要选择相应 的选项,将得到相应的页面内容。这些子菜单的内容将在下一节具体介绍。

如果输入的用户名和密码在后台数据库中不存在,或者不匹配,那么将不能登录本论 坛,而是被自动转到如图 2-4 所示的 relogin.htm 页面,要求你重新登录。

Male 1 Street and the second section of the second sec	ha for her
	」 - 2013年後に注意した。第三部の第三部の第三部の 「1997年によって、1997年に第三部の第三部の第三部の第三部の第三部の第三部の第三部の第三部の第三部の第三部の
The story / web of the store of	
对不起! 您不能	追登录此BBS论坛

·川丞重周 [.	i.
1. 您是否填错了你的ID	
2 伤患否记错了磨碍。	
3 你是不勿略了Cane ^楼 。	
	F 凄和ズ幼繁阑易联系 201
4. XII * 32. 94 (X 1*) E 32.0 9 (9.32.0 9	
*	F啓录
×	1 2 4 4
用户名	
TA	
14	
本码提示	新田户注册
<u><u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u></u>	<u>#1/13/</u>
	No. 12 CONTRACTOR OF AN

图 2-4 重新登录页面 (relogin.htm)

任何错误的用户名和密码都不能登录到本论坛,必须重新登录。如果不小心填错了用 户名和密码,则可以重新填写,再次登录。

2. 新用户注册

因为只有在系统数据库中已经注册的用户才能登录到本论坛,所以新用户必须首先注册。只需要在主登录界面上单击"新用户注册"就可以登录到如图 2-5 所示的注册页面, 在相应的文本框中输入相应的内容就可以了。

3 E A FEHL Shorosoft	Internet Explorer	
	Contractor - Action Address	
	2 A CLARKER CON	· · · · · · · · · · · · · · · · · · ·
http://wjyfbl	os/regist.htm	
5.对于打*	号的项目是您必须填写的项目。	<u> </u>
账号	benfoz *	
真实姓名	teoteo *	
昵称	taotao *	2 2
生日	77.11.26	κ ά
密码	****	
密码验证	****	
提示问题	hehe *	(如"吃了没有?")
提示答案	hehe +	(如"还没吃呢")
通讯地址	tsinghua university	
工作单位	tsinghua universtiy	
联系电话	62777890	56
BP机号	191-1102939	51
E-mail	benf ox@263. net	
网上寻呼号	1234987	
个人签名档	hehe	ž
备注	hehe	4
	•	
	Provide the second s	
	**************************************	5.5.4.5.5.5.7.5.1.5.5.5
)		

图 2-5 新用户注册 (regist.htm)

每一位新用户都必须按照要求正确填写上面的注册项,如果带"*"号的项目为空或者 两次输入的密码不同,在单击"提交"按钮时,将弹出如图 2-6 所示的警告对话框,指明 此用户没有填写"真实姓名",而且两次输入的密码不相同。必须进行改正,否则无法注册。

Microsoft	Internet Explorer	13
		1.00
1.1		1.00
Constant Section	and the second second second	-
	CHARACTER PROPERTY	141
		Louis - S
		Q. 4

图 2-6 填写注册单时的警告对话框

在这里,我们输入的用户资料如图 2-5 中所示。账号为 "benfox",这里的账号就是用 户在本论坛中的身份标识。有些内容是可选的,可以选择填写,其中的个人签名档是指用 户在发表文章时的签名,在用户发表文章的过程中,系统会自动加入用户的个人签名档。 登录过水木清华站的读者可以很容易明白这个问题,在个人签名档中可以放入一些自己喜 欢的语句或者其他任何符号。

如图 2-5 所示,当按要求正确填写好个人资料后,单击"提交"按钮,将把用户填写 的所有内容经过判断递交到后台数据库中,并且显示出"注册成功"的信息,如图 2-7 所 示,并且注册好的用户可以马上登录本论坛,只需要在图中所示的文本框中输入"用户名" 和"密码"就可以成功登录到如图 2-3 所示的论坛主菜单。

🗿 詳冊成功:Microsoft Internet Explorer	
http://wjy/bbs/finish.asp	2000 4000
PETT-BBS给#-注册)	式功 🗳
患的个人资料已经	安全输入到数据库中/
谢谢您	5的支持!
登	录论坛
用户名	
密码	
R <	全球中写
	an a

图 2-7 注册成功(finish.asp)

并不是任何用户资料都可以递交到数据库中,因为用户名是每一位用户在本论坛中的 惟一标识,所以不能有重复的用户名。假如再次递交如图 2-5 所示的信息,也就是再次注 册一个用户名为"benfox"的用户,将显示出如图 2-8 所示的信息,用户的资料无法写入后 台数据库中,注册失败。这时需要用户退回到前面的 regist.htm 页面,重新选择一个新的用 户名。



图 2-8 账号已有人使用(finish.asp)

比较图 2-7 和图 2-8 可以发现,其实浏览器的地址栏中显示的地址和文件都是

http://wjy/bbs/finish.asp。这说明在不同的情况下,同一个 ASP 文件生成的页面是完全不同 的,这也就是所谓的动态页面的"动态"所在。

3. 密码提示

有时,用户可能会忘记自己的密码,论坛系统提供密码提示功能。例如上面的用户 "benfox"注册过程中,在如图 2-5 所示的注册表中填写了提示问题"神仙? 妖怪?"和提 示答案"呵呵",这里假定"benfox"用户忘记了自己的密码,需要密码提示,这时"benfox" 用户只需要在如图 2-1 的画面单击"密码提示"链接,浏览器将打开如图 2-9 所示的页面。

● 新樹樹示 Microsoft Internet Explorer	南區町
the site site the	
http://dido.it.tsinghua.edu.crybbs/password.htm	
PETT-BBS##-密码提示	
	2
请输入用户名	
输入账号 benfox	
e) 🛪 🖉 turnt	JOSSICO,

图 2-9 密码提示页面(password.htm)

如图 2-9 所示,在文本框中输入 "benfox", 然后单击图中的"提交"按钮, 浏览器将显示出如图 2-10 所示的画面。

🚡 影輕得示 - Microsoft Internet Explorer	100 100 100
THE BIG BIG BIG ING	
http://wjy/bbs/password1.asp	CALL 1
РЕТТ-ВВ5發行-回答问题	
您设定的问题是:	
神仙?妖怪?	
请输入问题的答案 呵呵	
(W)之() (1) (新安治)	
新用户注册	i i i i i i i i i i i i i i i i i i i

图 2-10 回答问题(password1.asp)

该画面显示出了"benfox"用户在注册时设定的问题"神仙?妖怪?",此时只需在图中的文本框中输入问题的答案,填写好答案之后,单击"提交"按钮就可以了。

当然,在这里查询的是"benfox"用户的密码,而且在这里我们知道"benfox"用户是

在此之前已经成功注册的用户,也就是说后台数据库中已经保存有用户名为"benfox"的记录,故在这里可以显示出设定的问题。假如在图 2-9 中输入一个数据库中不存在的用户名 "moongirl",则不可能显示出相应的设定问题,显示的画面如图 2-11 所示。



图 2-11 不存在的用户名(password1.asp)

如图 2-10 所示, "benfox"用户将设定问题的答案"呵呵"提交到 password2.asp, 首 先系统对问题的答案和数据库中保存的设定答案进行对照,如果答案正确则得到如图 2-12 所示的画面,其中显示出了"benfox"用户的密码"benfox",并提醒用户牢记。如果提交 的答案与数据库中保存的答案不相同,也就是输入了不正确的答案,则不会给出正确的信 息,如图 2-13 所示。

-

图 2-12 正确的答案显示出密码(password2.asp)



图 2-13 对错误的答案给出的提示(password2.asp)

虽然大多数用户都无须使用密码提示功能,但对于一个功能健全的 BBS 论坛, 必须具有这样的功能。

至此,介绍了用户登录和注册的过程,以及密码提示功能的使用。所有这些都只是对 用户的身份进行验证。在确认了用户的身份之后,用户就可以进入到如图 2-3 所示的论坛 主菜单。

2.1.2 本站精华

假如当前的用户是"pett",则显示的主菜单和图 2-3 完全相同。进一步选择其中的"本站精华"链接,将看到本站的各项统计数字。

如图 2-14 所示,目前本站共有 206 人注册,3090 人次访问本站,人均访问次数 15, 日平均访问次数 108。并且列出了当前论坛中的十大热门话题以及相应的讨论区,可以给 用户提供参考。这个页面可以帮助用户了解本站的规模,也可以了解到大家正在讨论的热 门话题,方便用户进一步选择相应的讨论区。

ጋ ለ β ቋ ም. Stem at Interact Explore:			
Anton http://why/bbs/ste.asp			
PETT-85504-本站精华			i S
本站目前共有206人注册 人均访问次数:15	共有3090人次 日平均访问	(访问本站 量: 108	
古口十大名	也门绪题		**************************************
主题;	计绘画	土幸赦 ()	ž.
我怎么才能找到兼职工作?	社会信息	70	ž.
清华的哪个地方最适合和女朋友约会?	休朗娱乐	52	¥.
要求增加爱情讨论区	本估讨论	48	ш К
怎么回事呢?	本位讨论	39	ж: a
爱是什么?	加性感性	38	**i 6-
我好无聊	知性感性	35	5. *
今年的新生学费多少?	文化教育	34	7
数据库问题求教!help!	电脑技术	32	4
大家好!	的被耻	30	8
我要运动!我要健壮!	休肉娱乐	28	18. 18.
哪里可以打网球!	休用娱乐	23	11 11
灌水一篇	创成版	22	97 81
这个系统还不错	的彼斯	20	p. 3.
[」	菜单		i i
			2

图 2-14 本站精华(site.asp)

图 2-14 所示的各项数字都是随时更新的,如果当前用户隔一段时间再次访问这个页面, 其中的各项数字或许就会有所变化,或许会出现新的热门话题。

阅读完"本站精华"中的内容之后,用户可以点击"返回主菜单"链接,返回到如图 2-3 所示的主菜单。

2.1.3 分类讨论区

在图 2-3 所示的主菜单中选择"分类讨论区"链接,浏览器将显示出如图 2-15 所示的 分类讨论区选单。从图中可以看到,当前本论坛中共有 7 个分类讨论区:休闲娱乐、本站 讨论、社会信息、知性感性、测试版、文化教育、电脑技术。同时,显示各讨论区中当前 的文章数目、版主和本版简介。各个讨论区的顺序是以文章的数目按降序排列的。

♪武将業単 - Starosoft Inte	met Explorer		A LL
	olive an	* 4 6	STATES
DOTATION COMMAN	****		
- Maria - Maria - Maria	÷	>类讨论	E
讨论区	目前交音级	K ź	本版简合
休闲娱乐	19	honeydue	传递娱乐动态,丰富你我生活!
本站讨论	11	xinrr	讨论本站的建设情况,希望大家提 出宝贵意见,我们会及时改正。
社会信息	6	guyajin	发布各种社会信息,比如交通、工 作、二手货交易等等。
知性感性	4	bruckner	知性感性
测试版	4	benfox	这是本站的 测试版, 大家可以尽情 灌水
<u>文化教育</u>	2	pett	文化和數實
电脑技术	2	realowen	讨论计算机科学的发展及各项计算 机技术的发展
			i i
		巡回主来早	
		and the the constant of the later of	
0			Salation () () () () ()

图 2-15 选择分类讨论区(board.asp)

在图 2-15 中,用户可以选择相应的讨论区来阅读该讨论区中的文章,而且也可以给相 应版的版主去信,也可以选择"返回主菜单"链接,返回如图 2-3 所示的主菜单。由于本 论坛还处在完善的阶段,并没有真正发布到 Web 站点上,所以各个讨论区的文章还很少。

在这里选择其中的"休闲娱乐"讨论区(当然也可以选择其他的讨论区,虽然内容不同,但浏览的形式是相同的),单击"休闲娱乐"链接,得到如图 2-16 所示的页面。从图中可以看到,休闲娱乐讨论区目前共有 14 篇文章,每一页显示 5 篇,所以共分 3 页。当前显示的是第 2 页,仔细观察发表时间,可以看到是按照时间降序排列的,最新的文章总是排在最前面。另外,在视窗的下面有三排工具选项,极大地方便了用户阅读,增加了讨论区的可读性。单击"【前一页】"链接可以翻到前一页,单击"【下一页】"链接可以翻到下一页,也可以选择直接跳到任意页,只需在"跳到第()页"中填入需要的页码,单击"提交"按钮就可以了。

ининавый, b. Store Secto Sur (Флан	soll Internet Explorer	ti Albad.		
4 81 0	2 1 Comp (1 bbs/main.asp?pid=2	un (1992 19- 2) 3 5 5 6 0		(? ???)
nenn-beog	誌-休闲娱	乐讨论区		N.
		共有3页,共有14篇文章 当前页面: 2	i i	· **
编号	(1) #	文章标题	发表时间	110
6	wjy	re: 希里可以娱乐?	2000-8-30 23:06:18	
7	benfox	re: 希里可以娱乐?	2000-8-30 23:05:57	
8	guyajin	re: 希里可以娱乐?	2000-8-30 23:02:36	
9	pett	那里可以娱乐?	2000-8-30 23:01:58	
10	vjy	淮水一篇	2000-8-29 1:48:45	
Ľ	前一页】	【下一页】	跳到第一页	皮
作者搜索pett		<u>美</u> 主题搜索	新史 讨论区导航 —	
	豊力式調査	2000	返回主菜单	
			<u></u>	

图 2-16 休闲娱乐讨论区(main.asp)

从图 2-16 中可以看到"作者搜索"和"主题搜索"两种搜索方式,下面分别用实例说 明这两种搜索功能。如图 2-16 所示,在"作者搜索"后面的文本框中输入"pett",然后单 击"提交"按钮,将得到如图 2-17 所示的页面。

• (21) • 🔸	/wiy/bbs/author.asp	}##: @##: @##: @##: @ ## : ?##-2	
ett BHS	ab (b 休闲城	吴乐 讨论区-作者搜索	
		当前页面: 1	
		共有2页,7篇文章	
编号	作者	文章标题	发表时间
1	pett	ne:那位兄弟恶毒和族去踢球啊?	2000-8-30 23:10:11
2	pett	那位兄弟恶毒和疲去踢球啊?	2000-8-30 23:09:30
3	pett	ne: 哪里可以娱乐?	2000-8-30 23:06:55
4	pett	那里可以娱乐?	2000-8-30 23:01:58
5	pett	re: 休闲娱乐	2000-8-28 15:45:52
		【下一页】 跳到第	页 🗱
多杏油			

图 2-17 作者搜索(main.asp)

图 2-17 中显示的风格和图 2-16 是一样的,不同的是文章的作者都是"pett",文章是
按照发表的先后时间降序排列的。用户"pett"共在休闲娱乐讨论区发表了7篇文章,分为2页,当前是第1页。最重要的是每一篇文章的标题都代表一个链接,点击之后可以阅读 文章的全文(这种功能在图 2-16 所示的讨论区也有)。由于这里显示的是第1页,所以没 有显示出"【前一页】"的链接。如果有很多页的话,可以选择直接跳到指定的页码。另外, 在"作者查询"后面的文本框中输入其他作者,可以进一步进行作者查询。阅读完文章之 后,可选择"返回休闲娱乐讨论区"链接,返回到如图 2-16 所示的休闲娱乐讨论区。

图 2-16 中还有一种"主题搜索"功能,在"主题搜索"后面的文本框中输入"哪里可 以娱乐?"然后单击"提交"按钮,就得到如图 2-18 所示的搜索结果。

inter 🕑 hittig	://wty/bbs/title_asp?pid=2			1 64
ste -BBS	548-64- 休闲娱乐	讨论区- 主题搜索		
		当前页面: 1		
		共有1页,5篇文章		
编号	作者	文章标题	发表时间	
1	pett	ne: 希望可以娱乐?	2000-8-30 23:06:55	
2	wjy	no:哪里可以娱乐?	2000-8-30 23:06:18	
3	benfox	re: 那里可以娱乐?	2000-8-30 23:05:57	
4	guyajin	re:那里可以娱乐?	2000-8-30 23:02:36	
5	pett	那里可以娱乐?	2000-8-30 23:01:58	
新本海		医 返回休闲娱乐讨论区		

图 2-18 主题搜索(main.asp)

可以看到符合"哪里可以娱乐?"这一主题的文章共有 5 篇,仅有 1 页,没有其他可 选择的页码。所以,没有显示出"【前一页】"和"【后一页】"的链接,也没有显示出直接 跳到任意页码的文本框。但仍可以进一步进行主题查询,也可以选择"返回休闲娱乐讨论 区"链接返回到如图 2-16 所示的休闲娱乐讨论区。

在如图 2-16 所示的讨论区界面,还可以看到一个"讨论区导航"下拉菜单,这是为了 方便用户随时切换讨论区设置的。单击"讨论区导航"下拉菜单,如图 2-19 所示。

讨论区导航 💌	
讨论区导航	
电脑技术	
社会信息	
XKX	
知性感性	

图 2-19 讨论区导航

用户只需单击其中的任意一个讨论区就可以切换到指定的讨论区。例如,选择其中的 "本站讨论",如图 2-20 所示,该讨论区当前页面 1 包含 5 篇文章。用户也可以使用"作

3.04.04.03	issí, h. Maras	oft Internet Explorer		
Beschike 175				
[###@)	http://wjy/bk	us/main.asp		
FET	T-EDS# (·-本站讨·	论讨论区	
			共有2页,共有7篇文章 当前页面: 1	
	编号	作者	支章标题	发表时间
	1	pett	<u>re: 3名田事呢?</u>	2000-8-31 11:05:27
	2	pett	计多谋需责意见!	2000-8-31 11:04:27
	3	pett	re: 3 2 0 7 2?	2000-8-28 23:29:24
	4	pett	<u>息名回事呢?</u>	2000-8-28 23:27:09
	5	000	re: 約) 截	2000-8-23 20:00:15
			【下一页】	跳到第 月 一月一月一月一月一月一月一月一月一月一月一月一月一月一月一月一月一月一月一
作者	f搜索		主题搜索	建築 讨论区 导航 三
			Reconstant.	返回主菜单

者搜索"和"主题搜索"功能,切换到其他的讨论区。

图 2-20 本站讨论讨论区(main.asp)

图 2-20 中文章的标题是按照发表时间按降序排列的,同一主题的文章并没有放在一起, 为了方便用户阅读,左下方的"主题方式阅读"选项按钮将实现把同一主题的文章放在一 起。单击"主题方式阅读"按钮,将得到如图 2-21 所示的排列方式。

ng og berger som	ofi Internet Explore Discussion Reference s/main.asp	0. 90 0 108 - Ont 0- 9- 3 (5 - 7 (5	1. 1999 1997 1997
err-ersé (-本站讨;	论讨论区	
		共有2页,共有7篇文章 当前页面: 1	i
编号	(i *	交景标题	发表时间
1	000	ne: (#) (4)	2000-8-23 20:00:15
2	xixi	<u>rə: 🕬 🙀 </u>	2000-8-23 20:00:13
3	xixi	<u>(18)</u>	2000-8-23 20:00:12
4	pett	请多提盆贵意见!	2000-8-31 11:04:27
5	p ett	<u>ne: 3407%?</u>	2000-8-31 11:05:27
		【下一页】	跳到第 页
作者搜索		主题搜索	讨论区导航
	MAR E		返回主菜单

图 2-21 主题方式阅读(main.asp)

注意:由于这里只是对本论坛系统作一个演示,所以每一页的文章数设置为5篇。 在编程中,这只是一个变量参数而已,需要调节的话,直接改动参数的值就可以 了,十分方便。具体设置见7.3节中的程序。

可以看到,图 2-21 所显示的文章排列是按照标题分类的,在同一标题内又按照时间的 降序排列,这种显示方式极大地方便了用户阅读对同一主题进行讨论的文章。注意,这时 的选项按钮变成了"时间方式阅读",单击此按钮即可返回到如图 2-20 所示的时间显示方 式。

注意:这里,虽然显示的是不同的讨论区或者不同的阅读方式,但观察浏览器的 地址栏,可以看到其中的文件都是 main.asp。这也能说明 ASP 的强大功能所在, 选择不同的参数,可以显示出完全不同的内容。

作为一个 BBS 论坛的用户,当然不只满足于阅读别人发表的文章,也需要自己发表文章来参与论坛中某一话题的讨论或者新发表一篇文章。在图 2-21 的中下方可以看到一个"发表文章"按钮,单击此按钮,浏览器将显示发表文章的界面,如图 2-22 所示。



图 2-22 发表文章界面(write.asp)

用户只需要输入文章标题和文章内容,然后单击"提交"按钮,系统会自动把文章内 容存储到后台的数据库中,用户的界面也会自动返回到如图 2-21 所示的阅读界面,但这时 显示的界面和图 2-21 所示的界面略有不同,最新发表的文章已经添加到文章列表。

在图 2-21 中,用户只能看到文章的标题列表,如需要阅读全文的话,只需要点击相应 的文章标题就可以了。例如,单击"祝本站早日发布!"标题,将看到这篇文章的全文,如 图 2-23 所示。



图 2-23 全文阅读 (fulltext.asp)

从图中可以看到关于本篇文章的全部内容,发信人:pett,信区:本站讨论,标题:祝 本站早日发布!时间:2000-8-31 15:33:28,文章内容:希望本站早日发布!各位网友可以 进行交流!结交更多的朋友!个人说明档:要想富,少生孩子多种树!来源于: 169.254.102.247。

阅读完全文之后,用户可以单击下方的"回复"按钮,参与当前主题的讨论。如在图 2-23 所示的界面中单击"回复"按钮,将得到如图 2-24 所示的回复文章界面。

■ X 表文① - Microsoft Internet Explorer ZACD - CALE: #2500 ACC/0 FSND: - CE-06	rr Ssi
A RUL O O O O O O O O O O O O O O O O O	
PETT-BOSette-本站讨论讨论区	
回复文章	
标题: RE: 根本站早日发布1	
谢谢各位的支持! 通 我们一定会加倍努力!	
	ł

图 2-24 回复文章界面(write.asp)

回复文章界面和发表文章的界面十分相似,其实两个界面的文件是一样的,都是 write.asp,可以参见 7.3 节中的程序解释。只是图 2-24 的标题栏中已经填写好了文章的标 题,并且注明了是回复文章 "RE"。其他的方面和图 2-22 是完全相同的。用户填写好文章 内容之后,单击"提交"按钮就可以提交文章到后台数据库中,浏览器也返回到图 2-21 所 示的讨论区界面,在其中的文章列表中可以看到回复的文章标题前面会有一个 "re"标志。

至此,介绍了本 BBS 论坛的核心部分——分类讨论区。从上面的介绍可以看到,构成

一些功能的 ASP 文件并不多,却可以实现强大的功能:可以选择不同的讨论区;不同的讨论区之间可以很方便地进行切换;可以选择"主题方式阅读"和"时间方式阅读"两种阅读方式;还可以进行"作者"和"主题"搜索;在显示文章标题列表时,采用了分页的显示方式,大大增加了本论坛的可读性;在任何一个讨论区,用户都可以方便地阅读任何一篇文章的全文;可以很方便地回复文章参与某个话题的讨论;也可以发表文章新开一个话题。这里实现的功能和一般的 BBS 系统 (如水木清华)讨论区中的功能几乎完全相同。

2.1.4 修改个人资料

论坛中任何一位用户,都有可能需要修改自己的个人资料。个人资料的修改十分简单, 只需要返回到如图 2-3 所示的论坛主菜单界面,在其中选择"修改个人资料"链接,就可 以登录到个人资料的修改界面,如图 2-25 所示。

9	http://wiv	
_	CONTRACTOR OF STREET	/bbs/modify.asp
		pett,你的个人资料设置如下,
ļ	真实姓名	pett ¥
E	见称	huhu*
4	±日	79.2.7
Į	密 码	ar a construction of the c
ş	空码验证	*****
1	是示问题	hehe*(如 "吃了没有?")
1	邊示答案	hehe *(如"还没吃呢")
ì	直讯地址	tsinghua university
	工作单位	tsinghua university
1	联系电话	62771234
1	BP机号	191-1234567
1	E-mail	pett@263.net
I	网上寻呼号	14377596
	个人签名档	要想言,少生孩子多种树t
÷	备注	阿阿 ···································

图 2-25 个人资料修改界面(modify.asp)

如图 2-25 所示,正确输入密码之后,系统就会以表单的形式显示出数据库中现有的当前用户资料。当用户修改完毕后,单击"提交"按钮,将显示如图 2-26 所示的窗口,提示修改成功。

∎ 往册成功 Microsoft Internet Explorer	ME E
THE MAD SHO CHELLED AND AND A	
ARE BALON FRANKING	
MINO A http://dido.it.tsinghua.edu.cn/bbs/update.asp	
PETT-BB5卷始-修改成功	
您的个人资料已经安全输入到数:	据库中1
返回主菜单	

图 2-26 修改个人资料成功(update.asp)

2.1.5 网友查询

通过"网友查询"可以很方便地得到论坛中每一位用户的一些统计资料,例如上站次数,发表文章的数目,上次的登录时间和退出时间,个人说明档及有无新邮件等。利用这样的查询功能,可以很方便地掌握一些好友的情况,方便相互之间的联系和了解。

网友查询的使用界面十分简单,在如图 2-3 所示的论坛主菜单中选择"网友查询"链接,浏览器将显示出如图 2-27 所示的查询界面,用户只需输入查询的用户名,就可以得到 所查询的用户详细资料。

]四友合連 - Successoft Internet Explorer	Part a Part
http://wjy/bbs/queryfriend.asp	
PETT-BB5枪任-网友查询	
输入用户名	
<u>Elec</u>	
返回主菜单	Į
	herenet /

图 2-27 网友查询界面(queryfriend.asp)

例如, 查询一位用户名为 "guyajin"的用户, 在文本框中输入之后, 单击"查询"按 钮, 显示出的查询结果如图 2-28 所示。可以看到"guyajin"用户的一些统计资料: guyajin(惠 惠) 共上站 12 次, 发表过 17 篇文章。 上次在[2001-3-26 13:52:17] 从[166.111.168.202]到 本站一游, 离线时间[因在线上或非常断线不详]。 [有新邮件]。

个人说明档如下:我爱萝卜和青菜!

同样可以查询本论坛中存在的其他任意一位用户的资料。在图 2-28 所示的文本框中输入用户名,就可以进一步查询。

♪国友登训 - Microsoft Internet Explorer	
	这种性的 在"我们的"
E PARA PARA	0 2 4 S 4 4 7
http://dido.it.tsinghua.edu.cn/bbs/queryfriend.asp	
PETT-BBS给tt-网友查询	
guyajin(急急) 共上站12次,发表过17篇;	丈 ≢。
上次在[2001-3-26 13:52:17]从[166.11 一游,	1.168.202]到本站
离线时间 [因在线上或非常断线不详]。	[有新邮件]。
个人说明档如下。	
我爱萝卜和青菜	
输入用户名	
10 Mail	

图 2-28 查询结果显示(queryfriend.asp)

如果所输入的用户名在本论坛的注册用户中并不存在,例如,输入的用户名 "haha" 并不是一位注册用户,则显示出如图 2-29 所示的提示信息:用户 haha 不存在!请重试!

ጋ ዞፋራ ሳሳት Microsoft Internet Explorer	
The second seco	20 (1911) (1917)
返回主擎单	
	E V M

图 2-29 不存在的用户(queryfriend.asp)

查询完毕后,点击"返回主菜单"链接就可以返回到如图 2-3 所示的论坛主菜单,以选择其他的子菜单。

2.1.6 个人信箱

为了方便论坛中的注册用户彼此之间通信和联系,在图 2-3 所示的论坛主菜单中,有 一个"个人信箱"子菜单,在这个子菜单中,任何一位注册用户都可以收取别的用户的来 信,也可以给任何一位用户寄信,并且可以很方便地回复别的用户的来信。

在图 2-3 中的登录用户是"pett",所以选择"个人信箱"子菜单后,显示的是当前登录用户"pett"的个人信箱,如图 2-30 所示。"pett"用户共有 2 封来信,而且有 1 封未

读,也就是有1封新邮件,从信件列表中还可以看到寄信者、来信的主题、发信时间以及 信件的状态。

3.网友开阔、5	ficcosoft Internet i	sploter					
		Alle College A					
j eb re	N 8 8 6	ent run ()					
Jenn D ha	p://wjy/bbs/letterbox	(.asp					
РЕТТ-Е	PETT-BESGE - 个人信箱						
		pelt个人	信箱				
	pet	t您有2封信,其 [·]	中有 1封新邮件	20 20 20			
编号	寄信者	主题	发信时间	状态			
1	guyajin	<u>你好吗</u>	2000-8-25 21:59:33	已读			
2	wjy	<u>Re:吃了吗</u>	2000-8-28 15:01:09	未读			
		寄信给					
		Million States					
		<u>返回主</u> 家	<u> </u>				
				Wast 4			

图 2-30 pett 的个人信箱(letterbox.asp)

点击主题上的链接,就可以很方便地阅读其中的新信件,例如点击"Re:吃了吗",将 看到用户 wjy 在 2000-8-28 15:01:09 寄来的信件全部内容,如图 2-31 所示。



图 2-31 阅读信件内容(fullletter.asp)

如果需要回复的话,只需单击图中所示的"回复"按钮,回复信件的界面如图 2-32 所示。

2 1976/11 Storest Internet Diplore	
PETT-BEG能任- 回复wjy的来信	<u></u>
标题: Re:Re:吃了吗	
吃的不懂吗,小伙子! 至 内容,要注意减肥啊!	
	ž. Marine – Marine – Marine – Marine Marine – Marine Marine –

图 2-32 回复邮件(wirteletter.asp)

图 2-32 中的标题内容是系统自动加上去的,是回复用户 wjy 的来信,回复信件的标题 是相同的。当然,如果需要的话,可以修改信件的标题。书写好信件内容后,单击"提交" 按钮,信件将发出,提交到后台的数据库中,收信人在登录本论坛时,就可以看到。发出 信件之后,浏览器自动返回到个人信箱界面,如图 2-33 所示,可以看到其中第 2 封信的状态已经变为"已读",而且显示的信息是"pett 你有 2 封信,其中有 0 封新邮件"。

<u>]</u> Mኦለመ · ·	icrosoft Internet (Khjorer		
С. С. С. –	p://wjy/bbs/letterbox	640 (C)	AND S PLOY	- (29)
PETT-3	B5枪任-个	人信箱		1
		pelt个人	信箱	Ŕ
	pe t	t您有2封信,其:	中有 0封新邮件	
编号	寄信者	主题	发信时间	状态
1	guyajin	<u>你好吗</u>	2000-8-25 21:59:33	已读
2	₩ју	<u>Re:吃了吗</u>	2000-8-28 15:01:09	已读
		寄信给guyajin		
		III.		
		返回主卖	—— 《单	-
				1 Intranet III

图 2-33 回复邮件完毕(letterbox.asp)

下面示例说明如何直接发信给其他用户。

在如图 2-33 所示的文本框中输入收信者的用户名,这里输入的是"guyajin",然后单击文本框下方的"书写邮件"按钮,浏览器将显示如图 2-34 所示的书写邮件界面。比较图 2-34 和图 2-32 可以发现,其实两者的显示风格是一样的,地址栏中的文件也都是

writeletter.asp。在这里,标题栏中系统并没有给出默认的标题,需要用户输入。输入完毕后, 单击"提交"按钮,信件就会自动提交到后台数据库中,浏览器也自动返回到如图 2-30 所 示的个人信箱界面。



图 2-34 寄信给其他用户

通过个人信箱,论坛中的每一位用户都可以快捷地收发信件,很方便地和其他用户进 行联系,在阅读信件时,设置的回复功能更是方便用户回复收到的来信。

2.1.7 聊天室

在如图 2-3 所示的主菜单中,有一个"聊天室"子菜单,单击"聊天室"链接,就可 以进入本论坛的聊天室,在其中可以面对面地和其他用户聊天,更快捷地进行交流。

注意:这里只留下这样一个链接,但这并不表示聊天室和整个论坛系统之间没有 任何联系。在编写程序代码的时候就会明白,这些不同的模块之间需要有一定的 数据传输和身份验证,也可以说,有一定的独立性但又存在一定的内在联系。聊 天室的开发实例不再具体介绍。

2.1.8 离开本论坛

在图 2-3 所示的主菜单中,最后一个子菜单是"离开本论坛",当用户选择本菜单之后, 浏览器将自动返回到登录界面。这里并不是一个简单的超链接,当用户选择了这个选项之 后,系统会自动终止用户的登录信息,用户无法再返回到论坛中的任何一个页面,无法继 续从论坛的数据库中读写任何数据。如果需要的话,只有重新登录。这一项功能主要是利 用 ASP 中的一个对象来实现的,具体参见 7.8 节中的程序介绍。

2.2 安全性介绍

随着计算机网络技术的发展,通过网络,人们可以方便、快捷地进行交流,交换信息 和文件。网络上也出现了很多基于网络的系统,比如上一节介绍的 BBS 论坛系统,论坛中 的每一位用户都是通过网络来访问并登录的,所以,安全性是开发者需要考虑的重要问题 之一。在编写程序代码的时候,必须考虑非法的访问者,以及如何保护合法用户的个人资 料等问题。

在编写程序时,加强安全性保护是必要的,但有时开发者需要折衷考虑,因为安全性 加强几乎总是伴随着合法用户使用系统的能力减弱。对于那些超级的入侵手段,在本论坛 系统的开发中没有必要考虑,那是计算机安全专家考虑的问题。总之,应综合考虑系统的 安全性,基本的预防措施是必须的,另外,必须经常检查系统的运行情况,及时发现问题, 避免造成更多的破坏。

对于本论坛系统采取的保护措施主要有:

- ◆ 只有登录的合法用户才能享受本论坛的服务。任何一位没有登录的用户,即使在 这个地址栏中敲入了本系统中的任何一个文件地址(除登录界面外),浏览器都会 自动定向到登录界面(login.htm)。这项保护功能的实现见 7.1.1 节"用户身份验 证"。
- ◆ 用户在修改个人资料时,需要输入密码进行验证,否则无法修改。
- ◆ 后台数据库设置严格的身份验证,保护数据不受破坏。
- ◆ 后台数据库中的数据进行定期备份,即使系统遭到入侵,也可以在最短的时间内 恢复原来的数据,从而将损失程度降到最小。见 3.8 节"数据备份"部分。
- ◆ 任何一个系统都有自己的弱点,所以 BBS 论坛的系统管理员必须了解该论坛自身的弱点,经常进行检查和防护,并及时发现新的漏洞,加以解决。

2.3 论坛的整体结构

图 2-35 示出了一个 BBS 论坛的整体结构。从中读者可以清楚地看到整个论坛的结构 设计。

图 2-36 和图 2-37 分别示出了个人信箱结构和分类讨论区结构。



图 2-35 BBS 论坛的整体结构



图 2-36 个人信箱结构



图 2-37 分类讨论区结构图

2.4 论坛功能比较

现将本 BBS 论坛和水木清华 BBS 的功能比较列于表 2-1。

表 2-1 Pett-BBS 论坛与水木清华比较

项目	本论坛	水木清华	备注
登录界面	有	有	基本相同
主菜单	有	有	本论坛子菜单少
本站精华	有	有	本论坛需要积累
本日十大话题	有	有	基本相同
查询网友 ·	有	有	显示的信息相同
聊天室	有	有	省去了具体细节
个人信箱	有	有	功能基本相同
修改个人资料	有	有	修改个人信息
分类讨论区	有	有	本论坛中的分类讨论区较少
主题搜索	有	有	基本相同
作者搜索	有	有	基本相同
发表文章	有	有	方式相同
短信息功能	无	有	Web 形式开发难度大
回复文章	有	有	基本相同
管理系统	无	有	希望读者能开发管理系统

当然,本 BBS 论坛与水木清华这样的 BBS 相比,还是有很大的差别,但是,基本的功能都已经具备,完全可以投入实际的应用。

第3章 构建后台数据库

在 Web 数据库系统中,最重要的底层支持是后台数据库系统,选择安全性好、功能强 大的数据库可以帮助开发者自动完成很多工作。在后台数据库的构建过程中,最重要的就 是数据库的设计,每一独立的 Web 数据库系统都需要一个专门的后台数据库来支持,用于 存储各种类型的数据。数据库结构的优化程度直接影响前台 ASP 代码的开发效率。

Microsoft 公司的 SQL Server 虽然不是功能最强大、应用最广泛的数据库系统,但由于 其和 Microsoft 的操作系统无缝结合,使用和操作简便易行,对于一般规模的网站或企业数 据库都可以完全胜任,因此以下的叙述都是基于 SQL Server 7.0。

3.1 SQL Server 7.0 概述

SQL Server 是一个功能完善的数据库管理系统,随着 Microsoft SQL Server 7.0 的发行, 使得 SQL Server 的功能更加强大,它能为任何规模的企业提供良好的数据管理方法。几乎 可以满足所有用户的需求,而且可以在 NT Server 和 Windows 9x 上运行,非常通用。SQL Server 以其强大、灵活以及易于使用的特性雄居业界榜首。

Microsoft SQL Server 是一个客户/服务器关系式数据库系统。客户/服务器关系式数据 库系统的确切含义包含两个部分:客户/服务器部分和关系式数据库系统部分。

客户/服务器也称作分布式计算技术,它指的是程序的所有数据处理过程,不像基于桌面、小型或大型的计算机系统那样发生在一台单独的计算机上。SQL Server 的不同部分可以运行于两个或者更多的计算机上。比如,SQL Server 程序的数据库引擎部分(数据和程序代码驻留的位置,完成大多数工作的地方)驻留在一个指定的服务器计算机上,而程序的另一部分客户界面则驻留在用户的台式机中。客户/服务器系统的组件通过网络进行通信。

关系式数据库系统是指 SQL Server 采用的一种数据库结构类型,它广泛地被软件开发 者所采用。一个关系数据库可以分成多个数据表,每个表进一步分成行(记录)和列(字 段)。

关系式数据库系统包括下列特征:

◆ 所有关系式数据库系统都使用 SQL 语言对数据库中的数据进行操作。SQL 语言是 IBM 公司于 20 世纪 70 年代开发的,已经成为一个工业标准。本章 3.5 和 3.6 节中 对 SQL 语句进行详细解释和说明。SQL Server 不但完全适应 SQL 语言的标准, 而且对 SQL 语言也有所扩展,增加了重要的新特性,正因为这一点,SQL Server 中包含的 SQL 版本也称为 Transact-SQL。

- ◆ 关系式数据库在许多方面都类似于一个高端编程语言。事实上,可以用 SQL Server 创建出满足机构需要的各种类型的业务应用程序。
- ◆ 关系数据库系统的数据存储和用来处理数据的编程逻辑是彼此分开的。在一个两 层客户/服务器结构中,一个 SQL Server 仅仅用来保存和处理数据,而另一个 SQL Server 则用来保存编程代码和逻辑,这些编程代码和逻辑用于处理前面那个 SQL Server 中保存的数据。
- ◆ 数据库完整性对于一个数据库系统是最关键的。关系式数据库系统有许多防止引入坏数据的特性。
- ◆ 关系数据库的设计使得修改十分容易,可以适应每天都可能发生的变化。
- ◆ 与相同数据在许多不同记录中重复出现的平面文件数据库不同,关系式数据库系统把大多数多余的数据排斥在外,减少了存储空间的需求,提高了数据访问的速度。
- ◆ 数据访问可以由应用程序本身进行限制,运行多级别的安全保护。

SQL Server 是一个真正的客户/服务器关系式数据库,可以利用它设计满足不断变化信息需求的分布式数据库系统。

3.2 SQL Server 7.0 的安装

SQL Server 的安装并不复杂,但是在安装过程中需要注意一些问题,否则将影响 SQL Server 的正常运行,甚至出现安装错误。

3.2.1 SQL Server 在不同操作系统上的安装

到目前为止, SQL Server 所支持的操作系统平台依然是清一色的微软产品: Windows NT Server, Windows NT Workstation, Windows 95/98, Windows 3.X 以及 MS-DOS。由于 后两种操作系统现在已很少有人使用,故这里主要叙述 SQL Server 对前三种操作系统的支持。

1. 在 Windows 95/98 上安装

因为 Windows 95/98 是面向个人用户的操作系统,并非服务器平台,所以在 95/98 上安装 SQL Server 时只能安装它的单机版本(即在安装时只能选择 SQL Server Desktop 安装方式,详见 3.2 节 SQL Server 7.0 的安装)。也就是说,在 95/98 上安装的 SQL Server 不具有数据库服务器的功能,无法提供网络服务。但是,我们还是可以将 SQL Server 所支持的前台工具完全安装到此平台中。

2. 在 Windows NT Workstation 上安装

Windows NT Workstation 是 NT 的工作站版本,把 SQL Server 7.0 安装在该平台上以后,基本上可以支持从网络上进行数据存取,也就是具有了一个服务器的环境。同时,也可以

将 SOL Server 7.0 的前台管理工具完全安装在该平台上。

3. 在 Windows NT Server 上安装

如果使用的是 NT Server 4.0 操作系统,那么在安装 SQL Server 7.0 前,必须将 NT 升 级至 Service Pack 4 (需要具有中文环境)。一般的 SQL Server 7.0 (中文标准版) 安装 CD 中,在 B 碟中会包含 NT Server 中文版的 Service Pack 4。

在 NT Server 中安装 SQL Server 7.0 时,我们通常选择 SQL Server Standard 安装方式, 使其具有网络数据库服务器的功能。

如果使用的是最新的 Windows 2000 操作系统, 那么 SQL Server 的安装基本上与在 NT 上的安装是一致的, 因为 Windows 2000 实质上就是 NT 的 5.0 版本。其中, Windows 2000 Professional 是 NT Workstation 的 5.0 版本, Windows 2000 Server 和 Windows 2000 Advanced Server 是 NT Server 的 5.0 版本。所以, 在 Windows 2000 上安装 SQL Server 7.0 时也应注意 针对操作系统版本的区别而选择不同的安装方式。

以上涉及到的 SQL Server 的两种安装方式(Desktop 和 Standard 方式),在安装 SQL Server 之前,安装程序会有相应的图形界面供用户选择。

3.2.2 安装 SQL Server 7.0 的系统要求

安装 SOL Server 7.0 的最低系统要求是:

- ◆ CPU: Intel 32 位, 基于 80x86 的 CPU(例如 Pentium, Pentium Pro 系列, 或 Pentium II 以上更高主频 CPU)。
- ◆ 内存:至少 32M,建议 64M 或更高。
- ◆ 硬盘空间:最小安装需 80M,典型安装要求 185M,完全安装需 210M。
- ◆ 文件格式: FAT 格式和 NTFS 格式,建议使用 NTFS 格式,因为它具有 NT Server C2 的安全级别。
- ◆ 平台网络环境:一般可采用 Windows 缺省的网络环境。如果要使 SQL Server 7.0 支持网络存取,或通过网际网络进行远程管理,可在安装 SQL Server 的平台中设 置好 TCP/IP 协议,其余采用标准的缺省设置即可。

3.2.3 SQL Server 7.0 的安装过程

笔者在这里使用的操作系统是 Windows 2000 Server, 故选择安装标准版。

注意: 在安装 SQL Server 之前, 要确认已经安装了 Microsoft Internet Explorer 4.10 或者更高的版本。如果还没有安装, 则使用 SQL Server 安装盘上提供的 Internet Explorer 的副本。

下面说明安装 SQL Server 的步骤。

(1) 以具有管理权限的管理者身份登录到 Windows 2000 Server (NT 5.0)。

(2) 确保前台没有运行其他程序, 例如 NT Event Viewer(事件查看器)、Registry Editor (注册表编辑器)或者其他的 NT 管理工具。也假定服务器没有运行任何其他版本的 SQL Server.

(3) 如果从 CD 上安装 SQL Server,那么应该遵循下列步骤来开始 SQL Server Setup 程序:

在 CD 驱动器中放入 SQL Server CD。几秒钟后, SQL Server 程序将自动启动,显示的 画面如图 3-1 所示。



图 3-1 SQL Server 的自启动画面

为了启动 SQL Server Setup, 必须选择 "Install SQL Server 7.0 Components" (安装 SQL Server 7.0 组件), 并在图 3-2 所示的画面选择希望加载的组件。



图 3-2 安装 SQL Server 7.0 组件选择

如果关闭了 NT Server 的 CD 自启动特性,那么可以手工启动 SQL Server Setup 程序。

方法是浏览 SQL Server 安装 CD 的根目录,并且运行其中的 Setup.bat 程序,将出现如图 3-1 所示的初始安装画面。

(4) 如果从网络上安装的话,那么应浏览安装文件夹的根目录,并运行 Setup.bat。

(5) 在如图 3-2 所示的安装 SQL Server 7.0 组件选择画面中,如果希望在 NT 系统上 (包括 Windows 2000 Server 和 Windows 2000 Advance Server)加载 SQL Server,则选择 "Database Server-Standard Edition"(数据库服务器-标准版);如果希望在 Windows 9x 上加 载 SQL Server,则选择"Database Server-Desktop Edition"(数据库服务器-桌面版)。

(6)出现安装的准备画面,如图 3-3 所示,安装程序将自动对安装系统环境进行测试。



图 3-3 安装测试

(7) 出现 "Select Install Method"(选择安装方法) 画面,如图 3-4 所示。一般情况下,都会选择在本地计算机上安装。如果是这样,只需要选择 Local Install-Install to the Local Machine (本地安装-安装到本地机)的默认值选项。如果希望安装到远程计算机上,可以选择 "Remote Install-Install to a Remote Machine"(远程安装-安装到远程机)。确定后单击 Next (下一步) 按钮。



图 3-4 安装方法选择画面

(8) 安装程序将自动进行操作系统的测试,以确定是否能够在本地机上安装 SQL Server 7.0 标准版。

注意:标准版只能安装在 NT 4.0 以上的服务器操作系统上,否则只能安装桌面版。 如果当前的操作系统是 Windows 2000 Professional,则弹出如图 3-5 所示的警 告提示框,显示出警告信息: "The Standard Edition server component cannot be installed on Windows NT WorkStation using this CD. Only client components will be available for installation" (使用这张 CD,标准版的服务器组件无 法安装在 Windows NT WorkStation 上,只能安装客户端组件)。



图 3-5 警告提示框

(9) 单击"确定"按钮将出现欢迎画面,如图 3-6 所示。SQL Server 的安装程序可以 帮助使用者完成整个安装进程的向导。单击 Next,继续进行安装。



图 3-6 安装程序的欢迎画面

(10) 将出现 Software License Agreement (软件许可协议) 画面,如图 3-7 所示。如 果接受这个协议,请单击 Yes (是)继续安装。否则将退出安装。



图 3-7 安装许可协议

(11) 单击 Yes 按钮出现 User Information(用户信息) 画面,如图 3-8 所示。在这里, 输入使用者和公司的名称。输入完这个信息之后,可以单击 Next 继续进行安装。





(12) 依次出现两个 SQL Server 安装画面(见图 3-9 和图 3-10)。在这里需要输入 10 位 CD Key。这个数字在 CD 包装的黄色贴条上可以找到。输入号码后,单击 OK 继续安装。



图 3-9 输入 CD Key



图 3-10 安装画面

(13) 出现 Setup Type(安装类型)选择画面,如图 3-11 所示。在这里必须选择执行 哪一种安装类型。



图 3-11 选择安装方式

SQL Server 提供了三种不同类型的安装方法: Typical (标准)、Minimum (最小化)和 Custom (自定义)。

- ◆ Typical(标准) 选择这个选项,客户实用程序将使用所有的默认安装选项。这 意味着安装者所作的决策最少。该选项自动安装 SQL Server、Books Online、Quick Tour 和 What's New。该选项大约占用 163MB 磁盘空间。它不会自动加载 Full Text Search 软件、开发工具和范例程序。建议安装者一般情况下选择这种安装方式。
- ◆ Minimum(最小化) 这个选项加载用来操作 SQL Server 的最少数量文件,使用 所有的默认安装选项。除了不加载 Management Tools、Books Online、Quick Tour

和 What's New 之外, Minimum 安装和 Typical 安装类似。该选项大约占 74MB 磁 盘空间。

- Custom(自定义) 这个选项不仅使用户能够准确地选择自己希望加载的 SQL Server 组件,而且可以选择许多不同的 SQL Server 选项,而不是仅仅自动接受默 认选项。自定义安装包括下列选择:
 - ◆ 网络协议
 - ◆ 字符集
 - ◆ 排序顺序
 - ◆ 安装的实用程序
 - ◆ 安装哪个在线文件
 - ◆ 在何处放置程序文件
 - ◆ 在何处放置数据文件
 - ♦ SQL Server Services 使用什么服务账号
 - ◆ 是否自动启动 SQL Services
 - ♦ 使用哪个程序组放置 SQL Server 图标

选择 Custom 安装提供了最大的选择灵活性,安装者可以随意选择自己需要的组件。在 这里假定选择 Typical (标准)安装方式。单击 Next 继续。

(14) 出现 Select Components(选择组件)画面,如图 3-12 所示。安装者必须在这里 选择那些希望加载的组件。注意,在画面左面的是 Components(组件),画面右面是 Sub-Components(子组件)。首先通过单击复选框来选择组件,如果合适的话,再通过单击 复选框来选择子组件。在为安装选择合适的组件时,可以通过观察画面底部的可用磁盘空 间来计量安装的空间大小。完成这个画面的选择之后,单击 Next 继续安装。



图 3-12 选择组件

(15) 出现 Character Set/Sort Order/Unicode Collation (字符集/排序顺序/Unicode 整理) 画面,如图 3-13 所示。从画面上方的下拉式列表框中选择合适的 Character Set 和 Sort Order。 从画面底部的列表框中选择 Unicode Collation。在右下方的复选框中还可以选择 Unicode 字符是否区分大小写。完成这个画面的选择之后,单击 Next 继续安装。

Own	aut Sar 936 Chinese (simplified)		
3 0	t Droben (Inclusion) and the second	nes All provided 1	
(Unics	es Calation		14-12-1 14-12-1
	Binary Order General Unicode Catalan		
	Chinese Bopomofo (Taiwan) Conneces due fu dean Chinese Stroke Count (Taiwan)	 I Michelsenson I Gine secondere 	

图 3-13 选择合适的 Character Set、Sort Order 和 Unicode Collation

(16) 将出现 Network Libraries (网络库)画面,如图 3-14 所示。对于安装的每一个 Network Library,必须输入相应的信息。通常情况下,默认值就可以了。选择好这个画面 之后,单击 Next 继续安装。



图 3-14 选择合适的网络库

(17) 出现 Services Accounts (服务账号),如图 3-15 所示。在这里,如果希望为每个 SQL Server 服务指定相同的 Service Account,要确保选择 "Use the same account for each

service. Auto start SQL Server Service"(所有服务使用相同的账号,自动启动 SQL Server 服务)。然后,可以选择 Use a Domain User account (使用域用户账号),在 Username (用户 名)文本框中输入合适的服务账号名称,在 Password 文本框中输入相应的密码,在 Domain 文本框中输入账号所在的 NT Server 域的名称。



图 3-15 选择 SQL Server 使用的服务账号

如果希望为服务账号使用 Local System (本地系统)账号,而不是 Domain User (域用 户)账号,则选择 Use the Local System account (使用本地系统账号)单选按钮。如果选择 了该选项,那么所安装的 SQL Server 就不能和任何其他 SQL Server 进行通信了。

如果希望为每个 SQL Server 服务使用不同的服务账号,那么必须选择 "Customize the setting for each service"(为每个服务自定义设置)单选按钮,然后分别为每个服务填写 Service Settings(服务设置)。

完成选择之后,单击 Next 继续安装。

(18)出现"选择授权模式"画面,如图 3-16 所示。这一步骤中的设置很重要,其中的"每服务器有(⊻)4 个同时连接"表示每个服务器同时可以有几个连接,这在不同的 SQL Server 之间通信时十分重要。默认的连接是 2 个,单击"添加许可证"按钮,可以改变同时连接的数目。设置好之后,单击"继续"按钮,继续安装。

(19)现在将出现复制文件的画面,如图 3-17 所示,表示安装程序已经开始安装 SQL Server 到计算机中。在大约 5~10 分钟之后(当然这取决于计算机的速度),安装便可完成。

(20)最后的画面是要求重新启动服务器。如果可以重启而不会打断正在访问的用户, 那么可以立刻重启。在重启计算机之后, SQL Server 就安装完毕,如果选择的是自启动服 务的话, SQL Server 已经启动,并开始服务。

至此, SQL Server 安装完毕,可以开始使用 SQL Server 了。



图 3-16 选择授权模式



图 3-17 复制文件

3.2.4 常见的安装问题

在安装 SQL Server 的过程中,可能会遇到一些最常见的问题。

1. 使用了错误版本的 NT Server

使用不正确版本的 NT Server,或者使用一个旧的 Service Pack,将会引起大量的问题, 其中许多问题都很难诊断。如果发现使用了错误的版本,那么就应卸载 SQL Server,把 NT Server 升级到可接受的版本,然后重新加载 SQL Server。当然如果使用的是最新的 Windows 2000 Server 或者 Windows 2000 Advanced Server,就不会出现这一问题。

2. 在安装过程中打开了文件

在安装过程中,SQL Server 会替代一些 NT Server 的文件。如果在安装过程中这些文件碰巧打开了,那么它们将引起屏幕上显示一个重要错误信息。这也就是在安装 SQL Server 的时候不能运行其他程序的原因。如果你确定发现有一个或更多的 NT 实用程序正打开着,

那么应关闭它, 然后单击在信息上显示的 Retry 按钮。如果不起作用, 那么就只有退出 SQL Server 的安装, 再试一次。

3.3 SQL Server 的使用

在正确安装了 SQL Server 之后,就可以使用 SQL Server 了。在这一节中,将介绍如何 使用 SQL Server 的管理工具,主要介绍与 Web 数据库开发有重要关系的几个管理工具。

3.3.1 SQL Server Service Manager

如果在安装的过程中设置了自启动的话,这时 SQL Server 就已经开始正常运行于服务器的后台了,在 Windows 视窗底部的任务栏中可以看到**2**,这是 SQL Server Service Manager (SQL Server 服务管理器)的运行标志。SQL Server 服务管理工具实际上是 SQL Server 的众多管理工具中的一个,用于启动、终止和暂停 SQL Server (MSSQL Server)、SQL Server Agent 和 Microsoft 公司的 Distributed Transaction Coordinator (MSDTC)。

双击 SQL Server 服务管理器在任务栏中的标志,将打开 SQL Server 服务管理器的界面, 如图 3-18 所示。

[≓] O SQL Server Service Manager	Fer
Service SQLS erver Agent	n 🎆
	Contrin
A Summer	
🗳 san katala	
A CONTRACT OF A	

图 3-18 SQL Server 服务管理器界面

也可以从 SQL Server 的程序组中启动 SQL Server 服务管理器。从"开始"菜单中选择 "程序", 然后选择 Microsoft SQL Server 7.0 ,最后选择 Service Manager 就可以启动 SQL Server 服务管理器,如图 3-19 所示。

SQL Server 服务管理器在任务栏中的图标取决于服务器所处的状态。服务器的三种状态是:

の停止	白圆环内有一个红色方块
の暂停	白圆环内有两条黑杠
D启动	白圆环内有一个绿色三角形



图 3-19 从程序组启动 SQL Server 服务管理器

在需要启动、暂停或者停止 SQL Server 的服务时,只需要在如图 3-18 所示的界面上单击相应的操作按钮就可以了。

注意:在 SQL Server 服务管理器中,不同的服务需要分别启动,所以在启动时需要使用 Services 后面的下拉菜单选择不同的服务来分别启动,如图 3-20 所示。

SQLServerågent	
<u>Histributed</u> Transaction For MSSQLServer Microsoft Search SQLServerAgent	or di

图 3-20 不同的服务选择

如果要使用 SQL Server 数据库,或者通过 ODBC 来访问数据库,就必须确定 SQL Server 已经启动。

另外,在 SQL Server 的启动过程中有可能遇到如图 3-21 所示的问题。

	second second statistics and second
SQL Server Service Manager	
	1. July 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.

图 3-21 SQL Server 启动失败

从图中显示的信息可以看出,由于登录失败而无法启动服务。在前面的安装过程中已 经讲述过,必须是有管理权限的用户才能登录到 NT Server,在安装过程中如果使用者是 NT 系统用户,出现如图 3-21 所示的错误,很有可能是更改了系统管理员的密码。下面讲述解决这个问题的方法。 最简单的方法莫过于把系统管理员的密码再改回去,改为安装时的系统管理员密码, 就可以启动 SQL Server 了。但是,这种方法显然是不切合实际的,因为为了服务器的安全 保护,系统管理员的密码一般都会定期更换,如果这样的话,就会限制系统管理员对密码 的更改。

另外一种方法就是改变 SQL Server 服务的登录密码,改为和当前系统管理员登录相同的密码。改动的方法如下:

(1) 在"开始"菜单中选择"程序">"管理工具">"服务", 如图 3-22 所示。



图 3-22 启动"管理工具"中的"服务"选项

(2) 服务界面如图 3-23 所示,在界面右边列出了本地服务器提供的所有服务,拖动 最右边的滚动条,可以找到 MSSQL Server 一项,它的登录身份是 Administrator。如果 Administrator 的密码改变的话, SQL Server 将无法正常登录。

K %					F F
	1 P				a
1	P 67	0 - 0-0-0 12-			2 <u>60001178</u>] 1
Internet Connectio	为道		手动	LocalSystem	1
Intersite Messaging	允许		己兼用	LocalSystem	
PSEC Policy Agent	22	色息动	自动	LocalSystem	1
Kerberos Key Distri	产生		巳業用	LocalSystem	
Ser		已启动	自动	LocalSystem	822
Logical Disk Manager	逻辑	已启动	身勒	LocalSystem	22.
Logical Disk Manage	- 雅盘…		手动	LocalSystem	[]
Message Queuing	为分	巴尼勒	自动	LocalSystem	
Messenger	发送	已启动	自动	LocalSystem	
Microsoft Search		已息勒	目动	LocalSystem	R.
					1.47 0.19
Met Logan	X 70		于动	LocalSystem	
NetMeeting Remote	元正…		于动	LocalSystem	
Network Connections	11日本	E1840	于动	LocalSystem	
The Network DUE	黄洪		于取	Locarsystem	DEA
Reality in Control Control	52	a a 2	于初	Locatoystam	
	, 7 (R	C-1640	テ切	Locallystem	\$
Performance Logs 4	463	망 회 관 수	子和り	Locacyscem	Canal State
Burnt Sources	842 187	드러워) 분 함과	요리	Locallyscom	
California Service	19.4	三島市		LocalSustem	
	1. O gate				

图 3-23 服务界面

(3) 把鼠标放在 MSSQL Server 服务项目条上,单击鼠标右键,将弹出如图 3-24 所示

的选项列表。



图 3-24 选项列表

(4) 在如图 3-24 所示的选项列表中,选择"属性"选项,打开属性设置对话框,单击"登录"选项卡,切换到登录属性界面,如图 3-25 所示。可以看出, SQL Server 的登录 身份是 Administrator,而不是本地系统账户,实际上,当 Administrator 的密码改变时,所 有登录身份是 Administrator 的服务都必须重新设置登录密码,否则无法正常启动。在"密码"和"确认密码"后面的文本框中输入新的密码,然后单击"确定"按钮。

M550LServer的星性(本	地计算机)		
」元叶服务与题目	ite o		
Q	. Administrator		

	ne and a state state state of the state of the State of the state of	, in the state of the State of the state of	
		己启用	
	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	0 1 10 1 0	
			62

图 3-25 登录属性设置对话框

注意: 在如图 3-24 所示的选项列表中,如果选择"启动"选项,将重新启动 SQL Server,如果密码设置正确,就不会出现如图 3-21 所示的错误。

(5) 在如图 3-23 所示的服务界面中拖动右边的滚动条,可以找到 SQL Server Agent 服务,处理方法和上面的步骤一样,设置正确的 Administrator 密码。

完成这些步骤之后,不会再出现如图 3-21 所示的错误画面。

从上面的叙述可以看出, SQL Server Service Agent (SQL Server 服务管理器) 主要用 于 SQL Server 服务的启动、暂停和停止操作,可以选择自动启动。如果要对 SQL Server 进行任何数据操作,必须确保 SQL Server 的服务已经启动。如果 SQL Server 服务不能正常 启动的话,可以根据显示的错误提示信息找出发生错误的原因,然后进行相应的配置,以 确保 SQL Server 服务的正常运行。

3.3.2 SQL Server Enterprise Manager

SQL Server Enterprise Manager 是 SQL Server 的核心管理部分,可以使用它来完成管理 SQL Server 的大部分工作。下面介绍有关 SQL Server Enterprise Manager 的一些内容。

1. SQL Server Enterprise Manager 的使用界面

要对 SQL Server 进行管理或者数据操作,首先应该启动 SQL Server Enterprise Manager。 在 Windows 视窗中单击"开始">"程序">Microsoft SQL Server>Enterprise Manager,就可以启动 SQL Server Enterprise Manager 的视窗界面,如图 3-26 所示。



图 3-26 SQL Server Enterprise Manager 视窗界面

该图顶部的菜单栏中有三个选项:控制台、窗口和帮助。

◆ "控制台"菜单中只有一个子菜单:退出 Enterprise Manager。

 ◆ "窗口"菜单的功能和微软公司其他多文档管理界面应用程序中的窗口菜单功能 相似,利用它可以水平平铺打开的窗口或层叠窗口,还有一些其他的功能,如图 3-27 所示。

AND IN THE R. LEWIS CO., LANSING MICH.										e de	
 A second s		manula									79
				-84 -84							10 .
	7	k	di Ale	i i i Diae			97			194	<u>, 7</u> , 7-
			5			a servit			A.P	47	ete.



◆ "帮助"菜单中提供 Enterprise Manager 的帮助信息,从中可以找到一些关于 Enterprise Manager 的信息。有关 SQL Server 7.0 的更详细信息,需要参照图 3-19 所示的 Microsoft SQL Server 程序组中的 Books Online。

Enterprise Manager 的主菜单栏如图 3-28 所示,从左边开始,依次是操作、查看和 Tools (工具) 三个主菜单以及一些工具按钮。其中:



图 3-28 Enterprise Manager 主菜单

 ◆ "操作"菜单中的选项随控制树的层次不同而不同,当控制树处于数据库层时, 操作菜单中的选项如图 3-29 所示。实际上,在控制树的相应层上单击鼠标右键, 将弹出同样的选项框。



图 3-29 操作菜单

- ◆ "查看"菜单中提供了一系列不同的可用视图,从中可以选择大图标、小图标、 列表视图或详细资料视图,还可以选择关联的属性。
- ◆ "Tools"菜单提供了 SQL Server 的所有工具和向导(见图 3-30)。可以选择备份数据库、启动、停止和配置同步复制,而且还可启动许多工具,如 SQL Server Query Analyzer, SQL Server Profiler 等,其实这些工具有些是包含在图 3-19 所示的 SQL Server 程序组中的。



图 3-30 Tools 菜单

对于 Tools 菜单中每一个子菜单的功能和使用方法,这里不再赘述;但对于其中 重要的子菜单,见本小节后面部分的介绍。

◆ 在 Tools 菜单右边的是一些常用的工具按钮,这些工具按钮基本上都是一些快捷 按钮。通过这些快捷按钮,可以启动菜单栏下经常使用的菜单选项。各个按钮的 具体功能如图 3-28 所示。

Enterprise Manager 视窗界面的左面是 SQL Server 的控制树,从中可以清楚地看到 SQL Server 中各个对象和操作的层次关系,如图 3-31 所示。最高层的目录是 Microsoft SQL Servers,下面一层是 SQL Server Group,这里只显示了一个组:WJY (Windows NT),如果本服务器中的 SQL Server 与其他 SQL Server 之间建立了联系,则会显示出相应的组。再进一层的目录显示的就是 WJY 组中的所有内容 (见图 3-26):

Databases(数据库) Data Transformation Service(数据转换服务) Management(管理) Replication Monitor(复制监视) Security(安全) Support Services(支持服务)

所有的项目都还有进一步的选择,对于 Web 数据库开发者来说,常用到的是数据库和 安全两个项目,这里只介绍数据库中的子选项。



图 3-31 SQL Server 的层次关系

◆ Databases (数据库)中列出了本地 SQL Server 中包含的所有数据库列表,其中有 很多是系统数据库,用来存放有关 SQL Server 本身的系统信息,当然用户所创建 的数据库也包含在其中。例如,图 3-31 中所示的 pettbbs 数据库就是笔者新建的 数据库,点击数据库图标,可以看到关于当前数据库的所有子选项,主要有:

Diagrams(图表) Tables(表) Views(视图) Stored Procedures(存储程序) Users(用户) Roles(角色) Rules(规则) Defaults(默认值) User Defined Data Types(用户定义的数据类型) Full-Text Catalogs(全文目录)

根据控制树提供的层次关系,用户可以方便快捷地对数据库进行各种管理和操作,熟

悉各个层次中包含的内容,从而更迅速地找到需要的对象。

2. SQL Server Enterprise Management 的功能

使用 SQL Server Enterprise Management 可以处理以下的事务:

- ◆ 针对 SQL Server, SQL Agent, SQL Mail, Distributed Transaction Coordinator(DTC), Microsoft Search 等作启动、组态设定和停止等操作;
- ◆ 管理数据库备份;
- ◆ 管理数据库;
- ♦ 管理数据库维护工作;
- ◆ 管理登入与许可;
- ◆ 管理复制;
- ◆ 管理表格, Views (视图), Stored Procedures (存储程序), Triggers (触发), Indexed (索引), rules (规则), defaults (预设值)和 User Defined Data Types (用户自定 义数据类型);
- ◆ 排列规划工作;
- ♦ 产生 Web 网页;
- ◆ 生成 SQL Scripts。

在 SQL Server Enterprise Manager 中,可以完成几乎所有的管理和操作任务,具体的操作步骤和有关数据操作的步骤,将在 3.5 节中具体介绍,对数据库的管理参见 3.7 节。

3.3.3 SQL Server Query Analyzer

在 Enterprise Manager 的 Tools 菜单(见图 3-30)中,有一个 SQL Server Query Analyzer 选项。SQL Server Query Analyzer 的图形界面如图 3-32 所示,可以用来分析一个或多个查询的执行计划、查看数据结果和推荐的索引。

SQL Server Query Analyzer 是一个非常容易使用的工具,它也可以通过图 3-19 所示的 SQL Server 程序组来启动。

从图 3-32 中可以看出, SQL Server Query Analyzer 的图形界面中有上下两个窗格, 需要的时候,可以在上面的窗格中输入 SQL 语句,例如,在上面的窗格中输入:

select * from board

执行之后在下面的窗格中将显示出 SQL 语句相应的执行结果。

说明: 这是一个最简单的 SQL 语句, 含义是获取 board 表中的所有内容。关于 SQL 语句的详细解释, 可以参考附录 A。另外, board 表是第 2 章所述 BBS 论坛的后 台数据库中的分类讨论区信息表, 具体结构见 3.4.2 节 "设计 BBS 论坛后台数据库"。

对于所输入的 SQL 语句,有以下 3 种执行方法可供选择:

◆ Standard Execution(标准执行, Ctrl+E)以文本格式显示结果。图 3-32 显示了查询的标准执行界面。

é sut Server garav Analysia			No. 1
			10.00
the result of the wire Advances and the	b schot * franct. *		bes to be
	pettbbs		6. 200 C
select * from letter	1918 Million at Arrest Rey 1. Strengther and Annal		
			2
			analasing a si
poster	receiver		
ouvalin	Dett		
pett	guvajin		<u>.</u>
pett	Viy		
pett	guyajin		
y .	pett		1006
	2000-08-28 15:01:00	169.254.102.247	
ធាធា	hunu		4894
	Children V. Mithematika Mithiana I.	Second States and States	Character
	AND AND INC. AND		100000
August Au			Service Work not
		. الاشتخصيت الشيبيني	
		CALIFORNIA STATE	

图 3-32 SQL Server Qurey Analyzer 图形界面

◆ Execute to Grid (网格执行, Ctrl+D)的格式看起来会更加友好, 如图 3-33 所示, 它提供一种电子表格或表状结构 (行列方式排列)。

SUEServer Duery Au	alyzea					PST
		FANNELLY	、 教練術で			
Query WirsenGh	s.w.ty. Administrati	r (untilled) select*	komb*			
B D N [14]			pettbb	. 2		建设的研
lect * fr¢m bo	ard					
	bei	nfox		这是本站的测试版	,大家可以	2000-1
ĸ	re	aloven		讨论计算机科学的	发展及各项	2000-1
L	gu	7ajin		发布各种社会信息	,比如交通	2000-1
T	pe	35		文化和教育		2000-1
ê	xi	JEL		讨论本站的建设情	况,希望大	2000-1
ŧ	br	ickner		知性感性		2000-1
F	ho	neydue		传递娱乐动态,丰	言你我生活!	2000-1
Results Grid	and the second sec					
a da a an	$\sum_{i=1}^{l_{\mathrm{res}}} \frac{ h_{\mathrm{res}} }{ h_{\mathrm{res}} } \sum_{i=1}^{l_{\mathrm{res}}} \frac{ h_{\mathrm{res}} }{ h_{\mathrm{res}}} } \sum_{i=1}^{l_{\mathrm{res}}} \frac{ h_{\mathrm{res}}} }{ h_{\mathrm{res}}} } \sum_{i=1}^{l_{\mathrm{res}}} \frac{ h_{\mathrm{res}$	5 (K.D. A.S 5				2011 24 Page
2049 - 1980 - 1						31 J S

图 3-33 网格方式执行结果

◆ Execution Plan (计划执行, Ctrl+L) 给出了这种执行方法的实际执行过程, 如图 3-34 所示。



图 3-34 产生的执行计划

由于在这里还没有讲述任何关于 SQL 语句的知识,所以不再对 SQL Server Query Analyzer 作进一步的叙述,但这并不表示 SQL Server Query Analyzer 不重要。相反,在 Web 数据库系统的开发过程中,使用最多的 SQL Server 工具就是 SQL Server Query Analyzer, 它可以测试 ASP 程序中内嵌的 SQL 语句,并使其最优化。

关于 SQL Server Query Analyzer 的进一步使用方法,可以参考 3.5 节与 3.6 节中 SQL 语句的调试。

3.3.4 SQL Server 的其他工具

在如图 3-19 所示的 SQL Server 程序组中,除了前面介绍的 3 个工具外,还有一些其他的工具,在这里简要加以介绍。

1. Books Online

微软公司的在线帮助已经成了一种标准,微软的应用程序、服务程序和编程语言都采 用这种帮助系统。SQL Server Books Online 是一个优秀资源。这里的内容融合了相关产品 的知识。

SQL Server Books Online 使用起来也十分方便,总目录是一种树形结构,可以很容易 找到相关的帮助条目,而且还提供了不同的查找方式。

2. Import and Export Data

Import and Export Data(导入和导出数据)其实是 SQL Server 的一个向导,通过 DTS (Data Transformation Services,数据转换服务程序)向导可以借助 OLE DB 和 ODBC 轻松 导入和导出异构信息,可以在不同的关系数据库之间复制数据库方案和数据。
3. Performance Monitor

SQL Server Performance Monitor(性能监视器)集成了 Windows NT 的 Performance。 它能提供小到分钟级的活动,并能解决任何性能问题,它还是诊断系统可能出现的问题的 一条途径。另外, SQL Server Performance Monitor 还能指明提高和改善性能的办法。

4. Profiler

SQL Server Profiler 是一个不错的实用程序,它能实时地为服务器的活动提供连续的记录。SQL Server Profiler 监视着 SQL Server 产生的一切事件,并且能根据用户的指定条件对这些事件进行筛选,还可以把这些跟踪结果显示到屏幕或者保存到文件或表中。

5. Client Network Utility

SQL Server Client Configuration 用于为 DB-Library、Net-Library 和一切用户自定义的网络连接管理客户配置。

是否需要配置一个客户 Net-Library 与一个服务器进行通信是可选的。在 Microsoft NT 或者 Windows 9x 的操作系统上安装 SQL Server 7.0 时,将安装一个叫 Named Pipes (命名 管道)的默认客户 Net-Library。

3.3.5 SQL Server 使用小结

通过以上的介绍,基本认识了 SQL Server 的各个管理程序,对于 Web 数据库系统开发 者来说,其中最重要和最常用的是: SQL Server Service Manager (SQL Server 服务管理器), SQL Server Enterprise Management 和 SQL Server Query Analyzer。从上面的叙述可以知道, SQL Server 服务管理器主要用来启动、暂停和停止 SQL Server 服务; Enterprise Management 是管理 SQL Server 数据库以及进行数据操作的界面,通过它可以完成对数据库的大部分操 作; SQL Server Query Analyzer 是测试和执行 SQL 语句的工具,Web 数据库系统开发人员 可以用它来预调内嵌于 ASP 程序内的 SQL 语句。

3.4 设计 SQL Server 数据库

使用 SQL Server 的目的是按照一定的方式存储数据信息,并且能够按照指定的方式查询数据。要存储数据,则必须首先建立属于用户的数据库。通过前面的介绍,我们了解了 SQL Server 数据库的基本使用方法。在这一节中,将着重介绍如何建立一个 SQL Server 数据库。

在第2章中详细介绍了一个 BBS 论坛,其中的数据表结构虽然在前一章中已经给出, 但是具体的设计过程以及设计中应该注意的问题才是最重要的。本节讲述的数据库设计将 以前一章所述 BBS 论坛的后台数据库为示例。 3.4.1 数据库设计

在创建一个完整的关系型数据库之前,合理的数据库结构设计是相当重要的,它是创 建一个能够有效、准确、及时完成所需功能的数据库的基础。如果事先没有好的设计,在 后面的 ASP 程序开发过程中将不得不对数据库中的表格进行修改。不合理的数据库结构可 能会增加 SQL 语句的设计难度,甚至有可能无法从数据库中查询出想要的信息。因此,数 据库结构的好坏将直接影响前台程序的开发效率。下面简要介绍如何设计一个结构合理的 数据库。

设计数据库的基本步骤如下:

(1)确定新建数据库所要存储的数据,并按照一定的规则对数据类型进行分组。这一步可以认为是总体设计。

(2)根据数据类型的分组规划数据库中需要建立的表。基本上,每个表存储同一规则 的数据。

(3) 根据每组中的数据分类,确定各个表中所需的字段。

(4)为每个表明确一个具有惟一值的字段。这一步对某些表的设计可能不必要,但涉 及到惟一查询的表必须有这样的字段。

(5)确定各个表之间的关系。

- (6) 优化数据库结构的设计。
- (7) 输入数据并新建其他数据库对象。

对于不同功能的数据库在设计步骤上会有所不同,但基本上都需要遵循这样的步骤顺 序。

下面是对设计中的每一个步骤的详细说明。

1. 总体设计

所谓数据库的设计就是要确定数据库所要完成的任务以及如何去完成。因此,我们在 设计前必须明确将来希望从设计的数据库中得到什么信息,由此可以确定需要用什么主题 来保存相关事件(对应于数据库中的表)和需要什么事件来保存每一个主题(对应于数据 库中的字段)。

了解数据库就需要设计者与将要使用数据库的用户进行交流,讨论需要数据库解决的问题。在 Web 数据库系统的开发过程中,数据库的使用者就是前台程序开发者,因此在设计时应该遵循一定的原则:

(1) 了解前台程序的要求;

(2) 表体结构要优化,使用最少的表,存储最多的数据类型;

(3)关于同一对象的数据,尽量放置在一个表中。

了解前台程序的要求,就是列出前台程序需要往数据库中存储哪些数据和信息,并希 望以何种方式得到数据记录;还要列出所有的数据类型,并根据所属对象的不同进行分类。 所谓表体结构的优化,就是尽量减少一个数据库中表的数目,这样可以提高查询速度,降低 SQL 语句的设计难度。对于同一对象,在数据库中尽量放置在一个表中,这样有利于数据分类,便于查询。

2. 规划数据库中的数据表

规划数据表可能是数据库设计过程中最难处理也最为重要的步骤。这是因为,从第一步了解数据库任务的过程中,所获得的结果不一定能提供用于生成所有数据表的结构线索。

在设计表时,应该按照以下原则对信息进行分类。

(1) 表中不应该包含重复信息,并且信息不应该在表之间复制。

也就是说,为了尽量提高数据库的效率并保持数据库信息的一致性,应尽量避免数据 的冗余。

这样做的好处是很明显的:如果每条信息只保存在一个表中,那就只须在一处进行更新,这样效率显然更高,同时也消除了包含不同信息的重复项的可能性。但是有一点例外,就是两个表之间的连接字段的信息。

(2) 每个表应该只包含关于一个主题的信息。

这是因为,如果每个表只包含关于一个主题的信息,则可以独立于其他主题维护每个 主题的信息。

3. 确定数据表中的字段

每个表中都包含关于同一个主题的信息,并且表中的每个字段应包含关于该主题的各 个事件。例如,"用户表"中包含了用户的姓名、登录账号、登录密码、电子邮件等信息。 在决定每个表各应包含哪些字段时,要注意以下几点:

- (1) 每个字段要与表的主题相关。
- (2) 不包含推导或计算的数据(表达式的计算结果)。
- (3) 包含所需的所有信息。
- (4) 以最小的逻辑部分来保存信息。

4. 明确有惟一值的字段

为了惟一地标识数据表中的每一条记录,必须确定每个数据表中要有一个或一组字段 在整个表中始终具有惟一确定的值。因为我们必须要使用这样的字段来标识数据表中的每 一条记录,这样的字段就叫做该数据表的"主关键字"。为表设计了主关键字后,为了确保 惟一性, SQL Server 会自动避免任何重复值或者 NULL 值进入主关键字字段。一般的数据 库表中可以设置一个自动编号字段,用来标识表中的每一条记录。

5. 确定表之间的关系

因为已经将信息分配到各个表中,并且已经定义了主关键字字段,所以需要通过某种 方式告诉 SQL Server 如何以有意义的方法将相关的信息重新结合到一起。用户如果进行上 述操作,必须定义表之间的关系,也就是要确定数据表两两之间的关系(一对多或者多对 多)。这是通过将两个表中逻辑上需要相关连的字段联系起来而实现的。这两个(或两组) 字段通常内容是相同的,因为需要靠这样的字段来实现多个表之间具有相同信息字段内容 的一致性。

在定义关系时,如果其中一个字段在它所在的表中是主关键字,而在另一个表中的那 个字段不是主关键字,那么这时建立起来的就是一对多(1~∞)关系。原因很明显,因为 主关键字字段在其所属的表中具有惟一确定的值,而非主关键字字段则不一定。如果两个 字段在各自的表中都不是主关键字,那么创建的关系就是多对多(∞~∞)关系了。

在 SQL Server 中,打开 Enterprise Manager 后,点击想要创建关系的数据库前面的+ 号,在展开的目录项中选择 diagrams,就会进入创建数据表关系的画面。

6. 优化设计

在设计完需要的表、字段和关系之后,应该检查一下该设计并找出任何可能存在的不 足之处。这是因为,在设计时改变数据库的设计要比更改已经填满数据的表容易得多。

可以再用 SQL Server 在数据库中初步新建几个表,指定好表之间的关系,并在每个表中输入一些记录,看看能不能用该数据库获得我们所需的结果。

Web 数据库系统的开发过程中,数据库结构的优化有可能贯穿系统开发始末,在前台 程序的开发过程中,由于角度的改变,可能会产生新的概念和思想,随时可以对数据库结 构进行优化。

7. 输入数据

初步认定表的结构已达到了设计要求之后,就可以向表中添加数据了,一般情况下, Web 数据库系统中的后台数据库并不需要直接往其中输入任何数据,其中的数据基本上都 来自用户。但是在 Web 数据库系统的开发和调试过程中是没有任何用户的,所以为了开发 和调试的需要,可以事先在数据库中直接输入一些记录。

这里讲述的都是数据库设计的基本步骤和规则,具体的设计过程并非这样简单,需要 很多繁琐的操作步骤。下面将讲述如何在 SQL Server 数据库中设计并建立一个 BBS 论坛的 后台数据库。

3.4.2 设计 BBS 论坛后台数据库

1. 总体设计

首先应从总体上了解开发的 BBS 论坛所应具有的功能,可以参照第2章中介绍的 BBS 论坛系统。从功能上分析,所要存储的数据基本上分为以下几类:

- ◆ 用户注册资料
- ◆ 讨论区的情况
- ◆ 用户信件
- ◆ 表的文章

2. 规划数据表

数据库中的表主要根据所要存储数据的分类情况来决定。对数据进行分组时,要掌握 一定的原则。具体对 BBS 论坛数据进行分组时,在总体设计中,对所要存储的数据已经分 为了 4 组,而且符合数据的分组原则,因此,这里不打算对数据重新进行分组。根据这些 数据分组,制定如下的 4 个表:

- ◆ 用户资料表(Member 表),用于存储有关每一个用户的个人数据。由于其中存储 了所有用户的个人信息,因此需要更高的安全级别。
- ◆ 分类讨论区信息表(Board 表),用于存储每一个分类讨论区的信息,包括版主、 简介等字段。
- ◆ 文章表(Message 表),用于存储各个讨论区中的文章,包括发表者、分属的讨论 区、文章内容等字段。这是本数据库中最重要的表。
- ◆ 信件表(Letter 表),用于存储论坛用户之间的信件,包括寄信者、收信者、信件 主题、信件内容等字段。

根据以上的数据表规划,在数据库中建立相应的表。下面一步就是设计各个数据表中 的字段以及各个字段的属性。

3. 确定数据表中的字段

对于 Web 数据库系统中使用到的所有数据,都必须在数据表中找到相应的字段来存储, 因此,设计数据表的第一个原则就是必须完整。首先需要确定各个字段的字段名,字段名 一般都选取和所代表的字段有相近含义的英文单词或者缩写,或者任何可以让所有开发人 员容易接受的名称。当然也可以选择中文作为字段名,但使用中文字段在不同的操作系统 之间进行数据传输有时会出现意想不到的错误,一般不采用这样的方法来定义字段名。

注意:数据库语言和编程语言有一些关键词。关键词是在某一环境下能够促使某一动作发生的单词。为了避免冲突和产生错误,在命名表、列及其他对象时应避免使用关键词名词。建议最好使用汉语拼音的全称或缩写来命名表、列及其他对象。

数据表设计中最关键的一步是定义各个字段的数据类型, SQL Server 7.0 中提供了 23 种数据类型, 用户也可以自己定义数据类型, 对于每一种数据类型的解释见下一小节。字段数据类型的选择要根据字段中所存储的数据长度和类型来确定。

下一步就是设计各个字段的数据长度,应根据存储的数据长度最大值来确定字段的数据长度。还需要确定字段是否可以为空,也就是是否可以为 NULL 值。由于一般在前台的 程序中对用户所发送的信息都有一定的限制,因此在数据库表中,对字段的值是否可以为 空不必作太严格的限制。

注意:一般在调试前台程序之前,都将数据表的字段设置为可以为空,这样可以 方便前台程序的调试。这是因为,如果在前台的程序中向数据表的一个不能为空 的字段中发送空值,会出现数据库的操作错误,影响查找前台程序的运行错误。 按照以上的步骤,笔者所设计的 BBS 论坛数据表分别如表 3-1~表 3-4 所示。

			·····		· · · · · · · · · · · · · · · · · · ·	····
字段名	含义	数据类型	长度	精度	允许空	备注
Regnum	登录编号	Int	4	10		自动编号
Lybid	用户账号	Nvarchar	50	0	N	
Lybname	真实姓名	Nvarchar	50	0	Ν	
Nickname	昵称	Nvarchar	50	0	Ν	
Birthday	生日	Nvarchar	50	0	Y	
Password	密码	Nvarchar	50	0	Ν	
Question	提示问题	Nvarchar	50	0	Ν	
Answer	提示答案	Nvarchar	50	0	Ν	
Address	住址	Nvarchar	50	0	Y	
Workunit	工作单位	Nvarchar	50	0	Y	
Telephone	电话	Nvarchar	50	0	Y	
Bpnumber	寻呼机	Nvarchar	50	0	Y	
Email	E-Mail	Nvarchar	50	0	Y ·	
Icq	网络寻呼号	Nvarchar	50	0	Y	
Pen	个人说明档	Ntext	100	0	Y	
Other	备注	Nvarchar	50	0	Y	
Addtime	注册时间	Smalldatetime	4	0	N	Now ()
Addip	注册 IP	Nvarchar	50	0	Ν	自动获取
Vtimes	访问次数	Int	4	10	N	
Fabiaonum	发表文章数目	Int	4	10	N	
Intime	登录时间	Smalldatetime	4	0	Ν	Now ()
Outtime	退出时间	Smalldatetime	4	0	N	Now ()
Inip	登录 IP	Nvarchar	50	0	N	自动获取

表 3-1 用户资料表 (Member 表)

表 3-1 所示的用户资料表主要包括用户两个方面的信息:个人资料和上站统计。个人资料中主要包括用户的注册信息,如真实姓名、登录账号、密码、提示问题、联系方法等 字段。上站统计中主要包括访问的次数、发表文章数目、登录和退出的时间、上站的 IP 地 址等字段。

字段名	含义	数据类型	长度	精度	允许空	备注
Bdnum	讨论区编号	Int	4	0	N	自动编号
Boardname	讨论区名称	Nvarchar	50	0	Ν	
Bdowner	版主	Nvarchar	50	0	Ν	
Bdintro	讨论区简介	Ntext	300	0	Ν	
Createtime	创建时间	Smalldatetime	4	0	N	Now ()

表 3-2 分类讨论区信息表 (Board 表)

表 3-2 所列的分类讨论区信息表,包括各个分类讨论区的信息,主要有讨论区名称、 版主、讨论区简介、创建时间等字段。

字段名	含义	数据类型	长度	精度	允许空	备注
Addid	文章编号	Int	4	10	N	自动编号
Addname	文章作者	Nvarchar	50	0	N	
Message	文章内容	Nvarchar	1000	0	Ν	
Addtime	发表时间	smalldatetime	4	0	N	Now ()
Addip	作者 IP	Nvarchar	50	0	Ν	自动获取
Boardtype	讨论区	Nvarchar	50	0	Ν	
Title	文章主题	Nvarchar	50	0	Ν	
Re	是否为回复文章	Nvarchar	10	0	N	

表 3-3 文章表 (Message 表)

表 3-3 所示是各个讨论区中发表的文章,这是数据库系统中最重要的一个表,用户发 表的所有文章都存储在这个表中,其中主要包括文章编号、文章作者、文章主题、文章内 容、所属的讨论区等字段。

表 3-4	信件表	(Letter	表)

字段名	含义	数据类型	长度	精度	允许空	备注	
Poster	寄信者	Nvarchar	50	0	N		
Receiver	收信者	Nvarchar	50	0	Ν		
Title	信件主题	Nvarchar	50	0	N		
Message	信件内容	Nvarchar	50	0	Ν		
Addtime	发信时间	Smalldatetime	4	0	N	Now ()	
Addip	发信 IP	Nvarchar	50	0	N	自动获取	
Read	阅读次数	Int	4	10	N	默认值为1	
Addid	信件编号	Nnt	4	10	N	自动编号	

表 3-4 所列是论坛用户之间的来往信件,主要包括寄信者、收信者、信件主题、信件 内容、发信时间、发信 IP、阅读次数等字段。

4. 明确有惟一值的字段

这是对数据表进行完善的重要一步。在对数据表进行数据操作时,必须有一个判断不 同记录的标准。一般情况下,不同记录之间存在一个不同的字段值,但可能这个字段在不 同的记录之间并不是同一个字段,因此必须找出一个每条记录都具有惟一值的字段。

正如上面的表中所列,每一个数据表中都有一个编号字段。用户资料表(Member 表) 中的惟一值字段是登录编号(Regnum),分类讨论区信息表(Board 表)中是讨论区编号字 段(Bdnum),文章表(Message 表)中是文章编号字段(Addid),信件表(Letter 表)中 具有惟一值的字段是信件编号(Addid)。

5. 确定表之间的关系

要把不同表中的记录联系起来,就必须弄明白这些表之间的关系,其实主要就是确定 这些表中具有相同值的字段。这个 BBS 论坛数据库中的 4 个表主要是通过两个关键字段来 建立联系的,一个是用户的登录账号,根据账号可以从 Member 表中查找个人资料及相关 的统计信息,也可以从 Letter 表中查询收到的信和发出的信,最重要的是可以从 Message 表中查找此账号所发表的文章。另一个是讨论区字段,根据文章所属的讨论区可以从 Board 表中找到本讨论区中的相关信息。

具体的表之间的关系如图 3-35 所示。



图 3-35 各个数据表之间的关系

6. 优化设计

对表的设计进行优化是很必要的。但是,有时数据表的设计者在设计表的初期总感觉 数据表中的结构和字段非常完善,找不出应该优化的地方,主要的原因在于没有使用。在 前台的程序编写过程中,有时会对数据库的字段和结构提出一定的要求。这里的 BBS 论坛 数据表是经过优化之后的结果。

7. 输入数据

设计数据表之后,一般要往数据表中输入一些记录,主要用于对数据表进行测试,另 一方面也方便前台程序的调试。对于一个空数据表,前台程序在进行调试时容易出现问题。

至此,一个 BBS 论坛的后台数据库结构已经设计完成。在设计过程中应该注意的问题 在上面的设计步骤中有详细叙述,这些原则和问题应该引起数据库设计者的注意。在下面 的小节中,将详细叙述字段的数据类型选择。

3.4.3 SQL Server 7.0 中的数据类型

在设计数据库中表的时候,表体的结构设计是很重要的,但表中字段的数据类型也是 很重要的一个方面。一般需要定义数据类型的情况有如下4种:

- ◆ 建立表格字段
- ◆ 申请局部变量
- ◆ 申请存储过程里的局部变量
- ◆ 转换数据类型

SQL Server 系统中提供了 23 种数据类型,可以分为以下 9 大类,如表 3-5 所示。

编号	数据类型	描述	1
		int:	以4个字节来存储正负数,可存储范围为:-2^31至2^31-1
1	整数	smallint:	以 2 个字节来存储正负数,存储范围为:-2^15 至 2^15-1
		tinyint:	是最小的整数类型, 仅用 1 字节, 范围:0 至 2^8-1
		real:	位数为 24,用 4 个字节,数字范围:-3.04E+38 至 3.04E+38
2	浮点	float:	用 8 个字节来存储数据,最多可为 15 位,范围为:-1.7E+308 至
			1.7E+308
		numeric:	表示的数字可以达到 38 位,存储数据时所用的字节数目会随着使用
3	精确数值		权与位数的多少变化
		decimal:	和 numeric 几乎相同
		char:	长度是设定的,最短为1字节,最长为8000个字节,不足的长度用
			空格补上
		nchar:	长度是设定的,最短为1字节,最长为4000个字节,不足的长度用
			空格补上,存储一个字符需要 2 个字节
4	子付	varchar:	长度也是设定的,最短为1字节,最长为8000个字节,尾部的空格
			会去掉
		nvarchar:	长度是设定的,最短为1字节,最长为4000个字节,尾部的空白会
			去掉,存储一个字符需要2个字节

表 3-5 SQL Server 中的数据类型分类

(续表)

编号	数据类型	描述	
		datetime:	表示时间范围可以表示从 1753/1/1 至 9999/12/31,时间可以表
	口地时间		示到 3.33/1000 秒,使用 8 个字节
2 口朔时间	口舟門門	smalldatetime:	表示时间范围可以表示从 1900/1/1 至 2079/12/31,使用 4 个字
			节
		bit:	bit 数据类型代表 0,1 就是表示 true,false,占用 1byte
		timestamp:	该数据类型在每一个表中是惟一的。当表中的一个记录更改时,
	胜白		该记录的 timestamp 字段会自动更新
0	付化	uniqueidentifier:	用于识别数据库里面许多个表的惟一一个记录
		binary:	固定长度的二进制码字符串字段,最短为1,最长为8000
		varbinary:	与 binary 差异为数据尾部是 00 时, varbinary 会将其去掉
		text:	长宽也是设定的,最长可以存放 2G 的数据
-	立大和困伤	ntext:	长度是设定的,最短为1字节,最长为2G,尾部的空白会去掉,
'	又中和图像		存储一个字符需要2个字节
		image:	为可变长度的二进制码字符串,最长 2G
		money: 记录	金额范围为:一922337203685477.5808 至 922337203685477.5807,
8	货币		需要8个字节
		smallmoney:	记录金额范围为:-214748.3648 至 214748.3647, 需要 4 个字节

1. 整数数据类型

整数类型是用来定义存储对象的各种数据类型中的第一种。它被用来存储所有数字。 可以对整数直接进行算术运算而不使用函数。在每个整数类型允许的取值范围内,不管多 大的数字,其存储空间大小总是一样的。

(1) int

int (或 integer) 是 3 种整数数据类型中的第一种,存储-2³¹~2³¹-1 的所有正负整数,大约 43 亿个数。这个范围是从-2147483648~2147483648。每个 int 类型的值按 4 个字节存储,其中 31 位用于表明数的长度和大小,另有 1 位表示符号(正或负)。

(2) smallint

smallint 是第二种整数数据类型,可以存储-32768~32768 的整数。smallint 类型的数据存储占 2 个字节,其中的 15 位用来表示大小,另一位表示符号。

(3) tinyint

用 tinyint 存储结构可以存储 0~255 的正整数。每一数值占 1 个字节。

2. 浮点数据类型

浮点数据类型是几种可用来定义像"列"这样的存储结构的数值类型的第二组。与整数数据类型不同,浮点类型可用来存储十进制小数。

但要注意,浮点类型会遇到舍入误差的麻烦。所谓舍入误差是指存储在数值数据类型

中的数值只能精确到指明的精确位。例如,假若精确位是 15,大于 15 位的数虽然也能被存储,但会用前 15 位数字来不准确地代表这个数值存储。当然,舍入误差只对数字中最无足轻重的那几位数起作用。你可以在浮点类型所允许的精确度内存储数字。

注意: Microsoft 把像浮点类型这样的数据类型称为近似数值类型,因为它们只能存存储储结构允许限度内所代表的值。应避免对那些近似数值类型存储的数据进行比较,如在 WHERE 语句中,因为所载入的那些大于精确度的数值会在存储过程中被舍入。

(1) real

浮点类型中的第一种是 real, 它用 4 个字节来存储。可以在 real 类型中存储精确到第 7 位的正负小数。范围是-3.04E+38 至 3.04E+38。

(2) float[(n)]

浮点类型中的第二种是 float,如果其中的数 n 被省略,则它用 8 个字节来存储数据。可以在浮点型中存储精确到第 15 位的正负小数,范围是-1.7E+308~1.7E+308。

将 n 定在 1~7 的范围时,实际上是定义了一个 real 类型;当这一范围为 8~15 时,这一数据类型有相同的属性,就像 n 被忽略了一样。

3. 精确数值数据类型

decimal[(p[, s])]和 numeric[(p[, s])]

与 float 或 real 类型有所不同的是, decimal 或 numeric 类型可以提供小数所需的实际存储空间。精确的比例和数位由参数 p 和 s 来决定。这两种类型可以用 2~17 个字节来存储 10^{38} -1~-10³⁸ 的数值。

参数中的 p 表示可供存储的值在小数点左右的总位数。s 用来表示小数点右边的数位, s 必须小于或者等于 p,其中 p 的缺省值是 38, s 的缺省值是 0。

注意:如果不从命令行启动 SQL Server 改变精度的话, numeric 和 decimal 类型允 许的最大精度是 28 位。

4. 字符数据类型

在设计表时,常会使用字符类型来定义列或其他存储结构。字符类型可使表存储比数 值型广泛得多的各种符号。它可用来存储字母、数字符号、各种特殊符号(例如?和>等)。 在为列或其他存储单位输入字符数据时,须加上单引号或双引号。

(1) char

以 char 类型存储的每一个字符或符号占用 1 个字节。括号中的数字指明了所有字符所 占的存储空间。例如,如果将某列定义为 char (15),那么列中的每一个值都占 15 个字节, 可存储 15 个字符。输入的字符如少于 15 个, SQL Server 会自动在最后一个字符后增加空 格。

可以定义 char (n) 类型来容纳多达 8000 个 ANSI 字符。记住,列值总是容纳指定的 字符数,SQL Server 会自动在 char 值后添加空格来填满设定好的空间。 注意:如果某列被定义为 char 类型并设定为 NULL 型, 它将被视作 varchar 型。

(2) nchar

nchar 工作特性是 char 类型的真值。区别在于此类型每列最多容纳 4000 个 Unicode 字 符。

(3) varchar

可以用 varchar 类型来存储长度最多可达 8000 个字符的变长字符串。与 char 类型不同 的是,存储空间随行中各列所存储的字符数不同。

例如,若将某列定义为 varchar (15),则每一行的相关列中最多可以存储 15 个字符。 除非每个列值的长度都是 15 个字符,否则列值的最后会加上空格。用 varchar 来存储可能 有不同长度的列值会节省不少空间。也可以用关键词 Char Varying 来定义一个 varchar 类型。

(4) nvarchar

同样, nvarchar 的工作特性也在于它是 varchar 的真值。区别是列中字符的最大数目是 4000 个 Unicode 字符。

在下面的例子中,表中的两列分别被定义为 char 和 varchar 类型。插入行中的每列只存储了两个字符。结果第一列被加入了 3 个空格共占 5 个字节,第二列只用了 2 个字节来存储两个字符。例中所展示的行,其各列的值看起来是一样的,这样就掩盖了存储结构上的不同。

create table string exp

在下面的例子里,插入行所容纳的列值比数据类型所规定的最大长度分别多出两个和 一个字符。例中的 select 语句部分显示插入行的列值被截短了,只保留了前 5 个字符。而 且在插入行的时候得不到任何有关截断的信息。这一点应该引起注意。

```
inSert into string_exp
values ('abcdefg','abcdef')
select * from string_exp
char1 char2
.....
ab cd
abcde abcde
(2 row (s) affected)
```

因此使用字符数据类型时,应该注意以下两点:

- ◆ 当某列被定义为 char 或 varchar 类型时,以后列中所插入值的最大长度就已事先 被说明了。SQL Server 将自动截断超出最大长度的字符,但在执行时并没有任何 提示信息。
- ◆ 使用 char 类型时,列的已规定的全部空间都会被填满,不足部分会自动加入额外的空格。做报表时若引用这些类型列,输出的列也许无法排列整齐。如果额外加入的空格可能造成麻烦,可以用修饰语句将数据转换为 varchar 类型。

5. 日期和时间数据类型

datetime 和 smalldatetime 是用来存储日期和时间的数据类型。用日期和时间类型来存储日期和时间比用 char 或 varchar 类型方便得多。如果在这样的一个数据类型中存储了数据,可以很轻松地显示这些数据,因为 SQL Server 会自动用人们所熟悉的方式显示。也可以用特别的日期时间功能来使用以这种方法存储的值。

如果用 char、varchar 类型存储时间和日期或者在数值类型中存储时间,日期和时间值 在显示时不会自动以方便的格式出现。

(1) datetime

在这一类型中,可以存储从公元 1753 年 1 月 1 日到公元 9999 年 12 月 31 日的时间和 日期。

datetime 类型所占的存储空间为 8 个字节。SQL Server 使用前 4 个字节来存储 1900 年 1 月 1 日以前或以后的天数,负数表示该日期之前,正数则代表这天以后的日期。后 4 个 字节用来存储子夜 12:00 以后的毫秒数。

注意: datetime 值精确到 1/300 秒 (3.333 毫秒),以下的值将被忽略。例如, 1~3 毫秒都被视为 0 毫秒, 4~6 毫秒都被视为 3 毫秒。

在 datetime 中存储数值时,缺省的显示方式是 MMM DD YYYY hh:mm AM/PM,例如 Feb 18 1978 11:25AM。在一个 insert 语句或其他语句中使用 datetime 值时,必须在其前后 加上单引号。可以随意选择将日期或者时间放在前面,因为 SQL Server 能正确地识别和存 储每一部分的值。

在输入 datetime 值时,可以使用大写或小写字符来存储日期,在月、日、年之间使用一个或多个空格。当输入不带日期的时间时,时间的缺省值为 1900 年 1 月 1 日。如果输入一个没有时间的日期,则时间的缺省值为 12:00AM。如果两者都缺省,缺省值为 1900 年 1 月 1 日 12:00AM。

可以使用几种方法来输入日期。SQL Server 都能正确地识别和存储。你可以按字母方 式输入日期,使用月份的简写或者全名,在日和年之间可以用也可以不用逗号(,)。

若省略了年份中表示世纪的两位数字,后两位小于 50 则被视为 21 世纪(即前两位默 认为 20),如果大于 50 则被视为 20 世纪(即前两位默认为 19)。例如,输入年份 49,完 整的年份被存为 2049 年;输入年份 78,完整的年份就是 1994。如果要输入的年份与缺省 值不同,就必须输入完整的年份。此外,如果输入日期值中没有天,则缺省值就是当月的 第一天。如果将 SQL Server 设置为美国英语, datetime 缺省的输入顺序就是月、日、年。 当然这个顺序是可以改变的。在下面的例子中具体说明这个问题。

create table timetest (time1 datetime,time2 datetime) insert into timetest values ('Feb 18 78','780218') select * from timetest 结果:

time1

time2

1978-02-18 00:00:00.000 1978-02-18 00:00:00.000

(1 row (s) affected)

使用 datetime 数据类型时,如果插入一个值为空字符串的列,并且这个列定义为 datetime 列,将不会得到所希望的一个空值(NULL)。当中间没有插入字符的两个单引号 用作日期和时间数据类型的值时,值 January 1,1900 和午夜 12 点总是会被 SQL Server 插入。 必须按照时间单元为小时、分、秒的顺序来输入时间,在多个时间单元之间必须用冒号(:) 作为分隔符来使一系列数字被认为是一个时间而不是一个日期值。可以用 AM 或 PM (小 写大写都可以)来指定为午夜前或午夜后。

在秒前使用一个小数点和使用一个冒号,解释之后的结果是不一样的。后跟一单个数 字的小数点指定了十分之一秒;二位数字被解释为百分之一秒;三位数字被解释为千分之 一秒。冒号后的数字则被解释为千分之一秒。

(2) smalldatetime

smalldatetime 是用于定义存储结构的另一种日期和时间数据类型。用 smalldatetime 数 据类型可以存储从公元 1900/1/1 到 2079/6/6 年之间的日期和时间。一个 smalldatetime 数据 类型值的总存储空间为 4 个字节。SQL Server 使用 2 个字节存储从基础日期 1/1/1900 以来 的天数。另外 2 字节存储用午夜之后的秒数表示的时间。smalldatetime 的精度为 1 秒。使 用 smalldatetime 可以存储在有限范围里的和 datetime 相比时精度更低的值。

6. 特定数据类型

Transact-SQL 包含一系列用于数据存储的特殊数据类型,虽然大多数时候会用更传统的数据类型存储数据,如整型、浮点型、字符型和日期时间型,其实几乎 90%的数据都是用这 4 种数据类型来存储的。但也许要存储仅需表现"真"或"假"、"是"或"不是"的数据,这时也许需要一个更用户化的解决方法。

(1) bit

bit 数据类型用来存储那些可以只用两种状态表示的信息,bit 数据类型被存储在一个 位之中。因此,只有两种可能的值(0和1)被存储。如果在一个数据存储结构中的一个表 列输入任何不是 0 或 1 的值,都将被存储为 1, bit 类型不允许定义 NULL 值。

(2) timestamp

如果用 timestamp 数据类型定义一个表的列,则无论是插入一新的行还是更新已存在的行,一个计数值将自动添加到这个 timestamp 列。不能输入一个确切的值到定义为 timestamp 的列中去。

由 SQL Server 插入到一个 timestamp 列中的计数值说明了 SQL Server 已经完成操作的 顺序。输入到一个 timestamp 列中的值是用一个 varbinary (8)格式而不是 datetime 或 smalldatetime 格式存储的,缺省情况下允许在一个 timestamp 列中使用 NULL 值。一个 timestamp 值不是日期和时间,但在表和数据库中它总是惟一的,一个表只能定义一个列为 timestamp。

注意:时常用 timestamp 来保证一行可惟一地被识别。如果正在一行中更新列, 通常在 Update 语句的 Where 子句中指定 timestamp 域。这就确保了只更新表中的 一行,保证惟一性,因为 SQL Server 可能在任何插入一行和更新一行的时候维护 和更新它。在 Where 子句中常用 timestamp 防止两个人更新相同的行。因为 timestamp 在对这行作出更新的时候会自动地更新,这样可以肯定不会覆盖别人。 如果别人更新你正在处理的行,当他或她保存了更新时,这行的 timestamp 将被 更新,它将不再匹配你的拷贝。当你提出更新命令来保存所作的改变时,Where 子句将会发生错误,因为它找不到你所指定的需检索的行。timestamp 是行识别符 保持惟一的最好的 SQL Server 维护方法。

(3) uniqueidentifier

uniqueidentifier 是一个 16 字节全局性惟一识别符(GUID),其主要用途是在从许多不同数据库的许多不同表中收集数据时保持惟一性。

允许对 uniqueidentifier 数据类型进行的操作是比较操作(=和◇)、ISNOT、NULL 和 ISNULL 操作,不允许进行非等值比较或算术运算。所有用于列的限制和属性除 IDENTITY 外均可用于 uniqueidentifier 数据类型,每个 GUID 允许有一个惟一值。

在 Transact-SQL 中,一个新的 uniqueidentifier 是用函数 NEWID()产生的。

(4) binary (n)

可以用 binary 数据类型来存储由 8000 字节组成的位值。在括号中指定一个从 1~8000 的整数来定义位图的长度。必须指定 binary 列的大小,至少为 1 个字节,但可以存储全部为 0 的位图。

必须输入一个前面为 0X 的二进制值,可以使用字符 0~9 和 A~F。例如,输入 AD 的 值,形成 0XAD。如果输入的值大于所定义的长度,多余的长度被忽略,值也被在最小有 效数字后用 0 补齐。

(5) varbinary (n)

可以用 varbinary 数据类型来存储大到 255 个字节组成的位值,用在括号中指定的整数 定义所有位值中的最大长度,从 1~8000 字节,必须指定 binary 列的大小至少为 1 个字节, 但可以存储全为 0 的位图。

与 binary 数据类型不同, varbinary 数据类型的存储长度只能限制到实际值的长度, 像

binary 数据类型一样, 必须输入前面冠有 0X 的二进制值, 可使用字符 0~9 和 A~F, 如果输入的值比定义的最大长度大, 该值被截断。

7. 文本和图像数据类型

可用 text 或 image 数据类型来存储大于 2G 字节的字符型或二进制数据。

(1) text

用 text 数据类型来存储大量的文本,存储在文字段中的字符是典型的可直接输出到显示设备(比如显示器和打印机)上的字符。可以用文本类型数据存储从1到2147483647个字节的数据。

(2) ntext

在 NULL 和 ntext 之间主要的区别是 Unicode 字符需要 16 位的存储。

(3) image

可以用 image 数据类型来存储大的位值,从1 到 2147843647 字节长度。例如,在一个 表行单个列值中存储雇员照片、图像目录册或图形。通常不直接用 insert 语句输入存储在 image 列中的数据。

(4) 使用 text 和 image

在使用 select 语句时,存储为 text 和 image 数据类型的值就像显示其他列一样显示。

在用 select 语句显示定义为 image 数据类型的列中数据时,这些数据不会自动从其 ASCII 字符中转译出来。存储在定义为 text 数据类型列中的数据在用 select 语句输出时,可 自动转换为 ASCII 字符。Image 列不能直接用于输出,它可以送到另一个程序或是客户系 统上允许的程序显示对它进行的处理。

8. 货币数据类型

(1) money

money 数据类型用于存储货币值,用 money 数据类型存储的数据值由一个整数部分和 一个小数部分组成,用 2 个 4 字节整数存储。可以用 money 数据类型存储范围从 -922337203685477.5808 到 922337203685477.5807 的值。money 数据类型值的精度是万分 之一货币单位,一些前端工具用 money 数据类型显示存储的值时截取最小的分币。

(2) smallmoney

smallmoney 数据类型存储货币范围比 MONEY 数据类型更小。可用 smallmoney 数据 类型存储值的范围从-214748.3648 到 214748.3647。smallmoney 数据类型存储的数据值由一 整数部分和一小数部分组成,用 4 个字节存储。像存储定义为 money 数据类型的表列中的 值一样,一些前端工具显示 smallmoney 数据类型存储的值时截取到最小的货币单位。

当将值添加到定义为 money 或 smallmoney 的表列时,必须在最大位数字前加一美元符 号(\$) 或定义货币单位的符号。

9. 用户自定义数据类型——sysname

sysname 是用户自定义的数据类型, 它被定义为 nvarchar (128), 意味着 128 个 unicode 字符或 256 字节, 不允许为 NULL 值。

SQL Server 允许用户依据系统数据自定义数据类型,如表 3-6 所示。每个被创建的用户自定义数据类型都作为一行,通过 sp_addtype 命令以另一名称代表 SQL Server 所提供的数据类型,并被添加到 systypes 表中。

用户自定义的数据类型	映射的系统数据类型
Binary varing	barbinary
Character	char
Character varing	varchar
Dec	decimal
Integer	int
Double precision	float

表 3-6 数据类型映射

建立用户自定义数据类型的命令如下:

sp_addtype typename, phystype[, nulltype]

其中:

- ◆ typename 为用户自定义数据类型的名称,在每个数据库中是惟一的,但不同用户 自定义的数据类型可拥有相同的数据类型定义;
- ◆ phystype 为 SQL Server 提供的数据类型名称;
- ◆ nulltype 决定用户自定义数据类型是否允许 NULL 值,用户可指定允许 NULL 或 NOT NULL,若不指定,SQL Server 会取用当前 "ANSI null default"数据选项的 设定。

定义表的主要工作是指明表列的数据类型。Transact-SQL 可使你定义一些数据类型, 包括字符、数字和位图。也可以根据需要,为在数据表中使用而必须存储的程序、表和使 用数据表做其他工作。

一个表至少定义1列,最多不超过1024列。行的长度也是有限制的,最多为8060个 字节。

注意:可以用 image、ntext 和 text 数据类型来避免行长 8060 个字节的限制。用 image 和 text 数据类型定义的列是存储在表外的。

数据类型是表中列定义的第一个特性。列的数据类型控制着这一列中所存储信息的类型。数据类型栏紧随着"列名"栏,用一个关键词来定义它。在定义了某列的数据类型后, 它就被作为一个持久而不变的特性存储起来了。

3.5 建立数据库

在设计好数据库之后,下一步是在数据库系统中实现设计,建立相应的数据库。建立 数据库的方法有两种:一种是通过 SQL Server Enterprise Manager 建立数据库;一种是使用 SQL 语句中的 Create 命令来建立数据库。前一种方法可以称作硬操作,而后一种方法可以 称作软操作。本节将详细介绍如何使用这两种方法建立一个完整的数据库。

注意: 有关 SQL 语言的介绍见 3.6 节。在这里读者只需要了解建立数据库的 SQL 语句的结构就可以了。

3.5.1 使用 SQL Server Enterprise Manager 建立数据库(硬操作)

在 3.3 节中已经初步介绍了 SQL Server Enterprise Manager 的一些基本概念和使用方法。下面将详细介绍如何通过 Enterprise Manager 建立一个完整的数据库。

注意: 这里所要建立的数据库对象就是前一节中设计的 BBS 论坛数据库。

(1) 首先通过"开始"菜单中的 SQL Server 7.0 程序组启动 Enterprise Manager,得到 如图 3-26 所示的 Enterprise Manager 视窗界面,下面的所有操作步骤都是在这个界面中完成的。

注意:在进行这一步骤之前,必须确保 SQL Server Service Manager 中的 MSSQL Server 处于启动状态,此时 Windows 状态栏中显示的是绿色的三角形图标

(2) 在 SQL Server Group 中找到数据库所在的位置,一般是本地机上,连接到本地机上的 SQL Server。点击各个项目前面的 "+"号,展开所有的结构层次,得到的层次结构如 图 3-36 所示。



图 3-36 本地 SQL Server 的层次结构

(3) 在 Databases 选项上单击鼠标右键,将弹出如图 3-37 所示的快捷菜单。



图 3-37 快捷菜单

(4) 在该菜单中选择 New Database 选项,将弹出如图 3-38 所示的 Database Properties (数据库属性)对话框。

General Trensacti	n Lov		abaum ti s a		
d - 1	新建的教授	库名称			
Det theus filles					
文件名称	文件	的截盘位置	文件的初始	的大小	文件增长
C3					
		u ger til des Kiese			<u> </u>
Tile					
Y	1478-0	uk j ^u kai	ne file úze-	:fina	* *****
Y Tile T Satesatival Tile gave SUs X Tile gravit The arguint	ntoor 1	■k Z	ns fils size entrictad fil	2.11 M.2. 	
Y Tile To Antoentreal Antoentreal Antoentreal To Antoentreal Ant	******		nar file size- rantriciak fil prist Chief filmer	stinia artika artika	
YI Jile Fartssattosi Artssattosi Si Sussattosi Si Sussatto	n 2 a 2 a		nin file tize missionia fil generativa fil	x nasa seseba B	

图 3-38 Database Properties 对话框 (General 选项)

其中有 General (普通)和 Transaction Log (事务日志)两个选项。

(5) 在图 3-38 所示的 Name 文本框中输入新建的数据库名称。这里输入 "pettbbs" 作为 BBS 论坛的数据库名称。

(6) 在 General 界面中间的 Database files 框中,将出现数据库文件名称、文件的磁盘 位置、文件的初始大小以及文件增长的默认值。在 General 界面的左下方是数据库文件是 否自动增长以及增长方式的选择,右下方是是否限制数据库文件的大小以及设定的最大数 值。

注意: 在新建数据库时,这些默认的选项是在安装 SQL Server 7.0 的过程中默认的,如果在安装过程中没有改变自动的设定,则采用系统的默认值。当然,这些

值在图 3-38 和图 3-39 所示的对话框中都可以改变。

默认值分别是:

File name (文件名称): pettbbs_Data。

Location (文件路径): d:\server2000\mssql7\data\pettbbs_Data.MDF。

Initial size (MB) (初始大小): 1MB。

File growth (文件增长): Primary。

单击显示这些默认值的区域,就可以重新设置这些默认值。

Automatically grow file (文件自动增长):允许。

In megabytes (以兆字节增长): 非默认选项, 1MB。

By percent (以百分比增长):默认选项,10%。

Unrestricted filegrowth(不限制文件增长)

Restrict filegrowth (MB) (限制文件增长): 2MB。

(7) 切换到如图 3-39 所示的 Transaction Log (事务日志)界面,其中显示的是有关新 建数据库的事务日志文件。



图 3-39 Transaction Log (事务日志)选项

可以重新设定的参数和默认值如下:

File name (文件名称): pettbbs_Log。

Location (文件路径): d:\server2000\mssql7\data\pettbbs_Log.LDF。 Initial size (MB) (初始大小): 1MB。

单击显示这些默认值的区域,就可以重新设置这些默认值。

Automatically grow file (文件自动增长): 允许。

In megabytes (以兆字节增长):非默认选项,1MB。 By percent (以百分比增长):默认选项,10%。 Unrestricted filegrowth (不限制文件增长) Restrict filegrowth (MB) (限制文件增长): 2MB。

可以看出,除文件增长选项之外,事务日志文件和数据库文件的参数都一样。各个参数的具体值,参照图 3-39 中所示。一般在不了解各个参数的具体含义时,可以采用默认值。

注意: SOL Server 中每一个数据库都有一个单独的事务日志文件。

(8) 设置好数据库文件和事务日志文件的参数之后,单击"确定"按钮,将建立一个 新的数据库。如果新建的数据库名称在现有的数据库系统中已经存在,则 SQL Server 将弹 出如图 3-40 所示的警告框,提示用户新建的数据库已经存在。

prost reasons and reasons a reason for the	alle a set of			AND ADDRESS OF MANY COMPANY	Contraction of the local division of the
Microsoft 9	5QL-DM() (OD8C	SQLState:	42000)	
S - 142 - 1	10.20				
57A	Finery 196	-	the barry and the		August.
					u di Anna Alexandri
t fr. a	M. Cak		ATT MAR	17 . A.L.	
		Contraction of the			

图 3-40 数据库已经存在的警告框

(9) 在如图 3-36 所示的数据库层次目录中,单击"Database"前面的"+"号,展开 后,将看到本地数据库系统中的所有数据库,如图 3-41 所示。

ė 🍘 🖸	atabases
Ē Ē Ī	distribution
- b - (master
_ ⊕_ (model
D	msdb
e (Northwind
E E	pett 👔
Đ - (pettbbs
_ ⊡ • (pubs
j 🗄 🕻	tempdb

图 3-41 Database 目录

(10) 在图 3-41 中可以看到在前面步骤中新建的数据库 pettbbs,其中的 pett 数据库也 是笔者建立的。其余的 distribution、master、model、msdb、Northwind、pubs 和 tempdb 都 是系统数据库,用来存放有关本地数据库系统的所有系统信息,包括用户许可、数据库结 构等等。有关这些系统数据库中数据表的结构,可以参考相关的书籍。

(11)完成以上这些步骤之后,就可以对新建的数据库进行相应的操作了。在相应的数据库目录上单击鼠标右键,将弹出如图 3-42 所示的操作选项,可以选择新建数据库或者删除等操作。



图 3-42 对数据库进行操作

通过上面步骤建立的数据库只是一个空数据库,其中并没有存放任何用户可以调用的 数据信息。需要进一步建立相应的设计好的数据表,具体操作步骤参见 3.5.3 小节。

3.5.2 建立数据库的 SQL 语句(软操作)

在 3.3.3 节中已经介绍了 SQL Server Query Analyzer 的使用方法,从中也知道 Query Analyzer 是用来运行 SQL 语句的工具。下面介绍如何使用 SQL 语句在数据库系统中新建 一个数据库。

单击 Windows 桌面上的"开始" > "程序" > Microsoft SQL Server>Query Analyzer,将 启动如图 3-43 所示的视窗界面。

Ú,	iery W	Winiaster.W7	Y Administrat	or (untitled) c	reate databa	se*		
5		(<u>8</u> D.	- <u>6</u> ., ., 5			naster		
ree	ite dat	abase pet	tbbs					
						9829 747		
he	CREATE	DATABASE	process i	s allocating	1.00 MB	on disk	pettbbs'	•
(he	CREATE	DATABASE	process i	s allocating	, 0.49 MB	on di s k	'pettbb s	logʻ.
	1129.00	1122823-00		10.000		a areas Mil		A LODGERSON
	Results							and the state of the
isperto B	e ill or class		ALCO DOM:	2a. (1835).	ann a' griper An		1 AND THE REAL PROPERTY OF	Na (367

图 3-43 使用 SQL 语句新建数据库

在上面的窗体中输入相应的 SQL 语句,下面的窗体中将显示出相应的执行结果。 这里新建的数据库名称是 "pettbbs",相应的 SQL 语句是:

create database pettbbs

如图 3-43 上面的窗体中所显示,其中的蓝色语句表示的是系统命令语句。

注意: SQL 语句中的系统命令对大小写不敏感。Create 和 creatE 的作用是一样的。

执行后,在下面的窗体中将显示出相应的执行结果:

The CREATE DATABASE process is allocating 1.00 MB on disk 'pettbbs'.

The CREATE DATABASE process is allocating 0.49 MB on disk 'pettbbs_log'.

显示出这样的结果,表明已经成功建立一个数据库。

这里,在 SQL 语句中没有对新建的数据库作任何参数设定,因此建立的数据库的参数 都采用默认值。执行完语句之后,在 SQL Server Enterprise Manager 的界面中选择"刷新" 操作,将看到新建的数据库。

使用 SQL 语句建立数据库和通过 SQL Server Enterprise Manager 建立数据库的效果是一样的,都可以对新建的数据库作进一步的操作。

3.5.3 通过 SQL Server Enterprise Manager 操作数据表(硬操作)

新建好一个数据库之后,需要在数据库中建立设计好的表,用来存储一定格式的数据 信息。通过 SQL Server Enterprise Manager 可以在选定的数据库中建立数据表,具体的步骤 如下:

(1) 在如图 3-41 所示的 Database 目录中单击新建的 "pettbbs" 数据库前面的 "+"号, 显示出 pettbbs 数据库所包含的内容, 如图 3-44 所示。



图 3-44 pettbbs 数据库目录中的内容

(2) 单击图 3-44 中所示的 Tables 选项,在右边的区域将显示出 pettbbs 数据库中已经 存在的数据表。右边的区域显示数据表 4 个方面的内容: Name(表名)、Owner(所有者)、 Type(类型)、Create Date(建立日期)。如果是新建的数据库,其中会有 19 个类型为 System 的系统表。用户新建数据表的类型为 User。

(3) 在右边区域中或者在图 3-44 所示的 Tables 项目上单击鼠标右键,将弹出一个操 作对话框。

(4) 在对话框中选择 "New Table" 操作,将弹出如图 3-45 所示的 "选择名称"对话框。在其中的文本框中输入相应的数据表名称,单击 "确定"按钮。这里要建立的是 BBS 论坛中的用户表 (Member 表)。



图 3-45 输入表的名称

注意:如果新建数据表的名称在数据库中已经存在,单击"确定"按钮时将弹出 相应的警告框。

(5) 若输入的数据表名称不冲突,将出现如图 3-46 所示的设计界面。在前面的表 3-1 中我们已经设计好 Member 字段和表体结构。这里只需要进行相应的输入就可以了。

	ago table ino	-mber				Plant announced					B	~
	0 2 6				5 - A							
			(') _		2					325		
rega	rigin	int	4	10	0							Т
lybi	đ	nvercher	50	0	0							T
lyba		nvercher	50	0	0							
ni che	2.40 t	nvercher	50	0	0							
birt	hdey	nver cher	50	0	0	V				100.16		
pass	word	nver cher	50	0	0							
ques	tion	nver cher	50	0	0							
40.57	ar	nvar char	50	0	0							
addr	455	nvar char	50	0	0	V						
w or k	nani t	nvar char	50	0	0	V						
tele	phone	nver cher	50	0	0							
bpnu	aber	nver cher	50	0	0						1	
an ai	1	nver cher	50	0	0	V						
icq		nver cher	50	0	0	V						
pen		ntext	16	0	0							
othe	T	aver cher	50	0	0							
eddt	ime	smalldatetis	4	0	0							
addi	P	nver cher	50	O	0							
vtim		int	4	10	0							
febi	LODUD	int	4	10	0							
inti	ae	malldatetie	4	0	0							
outt	ime	smalldetetie	4	0	0							
inip		nver cher	50	0	0							
498 ¹⁹⁶		7.165.18 ¹ .07.101.18 ¹ .01.18	ь <u>,</u> м., н.,	ke_er_pe_			*_+:_*;_*	6.95 9 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1	ادر ه <u>م</u>		ຢູ່ສູ່ຢູ່ສູ	1

图 3-46 设计新建的数据表

在其中的"列名"一列中输入字段名,并选择相应的数据类型,如图 3-47 所示,设置 长度、精度、小数位以及允许空。

1111	(1) (1) (1)
money	
ncher	
ntext	
numeric	
nvar char	
real	
smalldatet:	

图 3-47 选择数据类型

(6)完成所有的输入后,单击左上角的保存按钮,保存设计好的数据表。然后关闭设计窗口,结束数据表的设计。

完成上面的步骤之后,在 SQL Server Enterprise Manager 视窗右边的表区域中将显示出 新建的数据表。

(7) 在相应的数据表项目上单击鼠标右键,将弹出如图 3-48 所示的快捷菜单。可以 通过其中的选项对数据表进行相应的操作。



图 3-48 对数据表进行操作的选项

- ◆ 选择"New Table"可以新建一个数据表。
- ◆ 选择 "Design Table" 将打开如图 3-46 所示的设计界面,可以修改对表体和字段的 设计。
- ◆ 选择"Open Table"将打开数据表,在此可以往数据表中输入数据。
- ◆ 选择"复制"、"删除"或"重命名"可以对数据表进行复制、删除和重命名操作。
- ◆ 选择"属性"选项,将弹出相应表的属性对话框,可以设定本表的一些参数。

这些选项中最常用的是"Open Table"。它的两个子选项如图 3-49 所示。



图 3-49 Open Table 选项的子选项

这两个子选项的不同之处在于显示的记录条数。

Return Top...(返回前...条):选择此项,将弹出如图 3-50 所示的对话框,可以填入任意大小的数字。单击"OK"按钮后,打开的界面类似于图 3-51,惟一不同的是显示的行数 是受限制的。

The second secon	the second se	and a second	A COLORADO A COMPLEX ADDRESS
Alternation of the second			S (1.9)
rsumper of ROWS			T CALLER AND S
man and state the second states of a solar the shares of the	CITAGE OF A CONTRACT OF A DATA STATE OF A DATA STATE	the setter description of the setter of the	SIGNATION DE LA CARACTERISTA
	11 19 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		A CONTRACTOR OF THE OWNER
A DECK PERMIT		1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.	Construction of the second
the second second second second second	and the second second second second second	States Property in the	1. 18 A. 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18 1. 18
and the second	in of some in fairly	1 Contraction Provession	an Martin Providence Alexand
Willing Co. Contractor of particular international strand strand			STOCKIC STOCKED
· 5925305		SPARE CONTRACTOR 201	Sale address
		SOME SHORE STORE	Market All and a second
1919909888		- Contraction of the second se	Sector and the sector of the s
- A state of the university of the second state of the state of	instruction in the second s		1000 Contraction (1000 Contrac
March Rettle America and the	Contraction of the second	er fin de la samuel	Contraction of the second
	and the second	Carl Street Street Street Street	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Carl States and States and States	THE REPORT OF THE REPORT OF THE REPORT OF THE	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A Designation of the second second

图 3-50 输入需要显示的行数

Return all rows (返回所有的行): 选择此项,将在如图 3-51 所示的界面中显示数据表中的所有行。

(8) 在这里选择 "Return all rows"子选项,将打开如图 3-51 所示的界面。其中显示 了 member 表中的所有内容。可以在这里往数据表中添加记录。

	' [1-5 ² -5 H	Table Scenober*					- ku	177
	La (6 8 2.3					
		14.877 P. 1660				AND NOT THE	826 TT 1653	1.8M-30
15	12	pett	pett	huhu	790207	123	beeh	heat
12	13	haha	pett	huhu	791248	123	heeh .	edf
÷.,	14	۲ ί۳	*j7	v.jy	790308	123	hehe	hehe
2	15	guyejin	边宇		790218	453525	土人最不爱喝作	と職業
	18	benfor	美瑞	朱熹理	770808	benfox	神仙?妖怪?	- M M
	19	benfox1	鼻聋	來裏彈	770608	benfox	神仙?妖怪?	
Ø			N & X & A				****	A.30
	96-97		State State	1847-1		Section of the sectio		18

图 3-51 打开数据表 (member 表)

图 3-51 显示了本 BBS 论坛中的所有注册用户,由于此论坛没有正式发布,其中的用户都只是用来测试系统而建立的。

注意: 在输入数据时, 要注意相应字段的数据类型, 如果输入的数据不符合设定的数据类型, 输入后 SQL Server 将弹出如图 3-52 所示的提示框。

			194. 1	and the second
۶QI	Server	Enterprise	Manager	
			a it is a first state of the second	
	×			
				A CALLER AND
1.	Restor		and the second	
5 4 S	2. e .			
		6.1	and the second secon	

图 3-52 输入的数据不符合设定的数据类型

至此,已经建立起一个完整的数据表,在数据表中输入数据之后,就可以进行相应的 数据操作,比如查询、删除等。

注意: 在建立好数据表之后,一般需要往数据表中输入一定数量的记录,以方便 调试前台的 ASP 程序。

BBS 论坛的其他三个数据表 Board 表、Message 表和 Letter 表的建立步骤和上面的步骤相同,只是表体结构和字段不同而已。

3.5.4 使用 SQL 语句操作数据表 (软操作)

3.5.3 小节中的所有操作步骤也可以用 SQL 语句来完成,下面举例说明如何使用 SQL

语句来建立数据表并往其中输入数据。

1. 建立数据表

(1) 和 3.5.2 小节中使用 SQL 语句建立数据库类似,首先需要启动 SQL Server Query Analyzer。打开的画面如图 3-53 所示。

Query _WTY.pettbbs.WTY_Administrator(unitiled)_create.table.ine*	
create table member	
(lybid Int NOT NU11)	
	201
	BECCO D
The command(s) completed successfully.	R
	23
	.
	C PROPERTY OF
Results	
Query bach surgested	Driz, Col S

图 3-53 SQL Server Query Analyzer 界面

(2)一个数据库系统中一般会存在多个数据库,而所要建立的数据表应该从属于某一 个数据库,因此在执行相应的 SQL 语句之前,必须选择相应的数据库。在图 3-53 上面的 工具栏中,选择其中的 DB 下拉菜单。该下拉菜单中的内容如图 3-54 所示,其中显示的是 当前数据库系统中存在的所有数据库,包括系统数据库。

pettbbs	7855
distribution	
master	
model	1
asdb	
Northwind	
pett	
post () de la	
pettbbsl	
pubs	
tempdb	
(Refresh)	

图 3-54 DB 下拉菜单

选择其中的"pettbbs"数据库,建立的数据表将都从属于"pettbbs"数据库。

(3) 在上面的窗体中输入相应的 SQL 语句,如图 3-53 所示,执行的结果显示在下面的窗体中。

SQL 语句为:

create table member

(lybid Int NOT NULL)

相应的执行结果为:

The command (s) completed successfully.

个	执行这一条语句,将在"pettbbs"数据库 字段,字段名为 lybid,数据类型为 Int,不	中建立一个 member 表,建立的表中仅有一 允许空值 NOT NULL。
	注意:新建的数据表中最少必须有一个等	F段,否则 SQL Server 将出现错误信息。
	Create Table 命令的基本语法如下:	
	Create Table table_name	
	(column_name datatype {identity NOT NULL N	JLL})
	[,]	
	下面给出建立 BBS 论坛数据库 member 表	表的 SQL 语句。
٠	建立用户资料表(member 表)	
	create table member	
	(Regnum int (4) not null,	登录编号
	lybid Nvarchar (50) not null,	用户账号
	lybname Nvarchar (50) not null,	真实姓名
	nickname Nvarchar (50) not null,	昵称
	password Nvarchar (50) not null,	密码
	birthday Nvarchar (50) null,	生日
	question Nvarchar (50) not null,	提示问题
	answer Nvarchar (50) not null,	提示答案
	address Nvarchar (50) null,	住址
	workunit Nvarchar (50) null,	工作单位
	telephone Nvarchar (50) null,	联系电话
	bpnumber Nvarchar (50) null,	寻呼机
	email Nvarchar (50) null,	E-mail 地址
	ICQ Nvarchar (50) null,	网上寻呼号
	other Ntext null,	其他信息(备注)
	pen Ntext (100) null,	个人签名档
	Vtimes int default 1,	来访次数
	addtime smalldatetime not null,	注册时间
	addip NNvarchar not null,	注册 IP
	fabiaonum int default 0,	发表文章数
	intime smalldatetime not null,	登录时间
	outtime smalldatetime not null,	退出时间
	inip nNvarchar not null	登录 IP
)	
٠	建立讨论区信息表(board 表)	
	create table board	

(bdnum int not null,	讨论区编号
boardname Nvarchar (50) not null,	讨论区名称
bdowner Nvarchar (50) not null,	版主
bdintro Ntext (300) not null,	讨论区简介
createtime smalldatetime not null	创建时间
)	
◆ 建立文章表(message 表)	
create table message	
(addid int not null,	消息编号
addname Nvarchar (50) not null,	文章作者
message Nvarchar (1000) null,	文章内容
addtime smalldatetime not null,	发表时间
addip Nvarchar (50) not null,	作者 IP
boardtype Nvarchar (50) not null,	所属的讨论区
title Nvarchar (50) not null,	文章主题
re Nvarchar (10) not null	是否为回复文章
)	
◆ 建立信件表(letter 表)	
create table letter	
(poster Nvarchar (50) not null,	寄信者
receiver Nvarchar (50) not null,	收信者
title Nvarchar (50) not null,	信件主题
message Nvarchar (50) not null,	发信内容
addtime Nvarchar (50) not null,	信件时间
addip Nvarchar (50) not null,	发信 IP
read Nvarchar (50) not null,	阅读次数
addid Nvarchar (50) not null,	信件编号
)	

在 SQL Server Query Analyzer 中执行上面的程序段,将在 pettbbs 数据库中建立前面设 计好的数据表。

2. 往数据表中插入数据

可使用 SQL 中的 INSERT 语句将行添加到数据表中。下面的示例将向上面建立的 Message 表中添加一条记录。

Insert into message

(addname, title, addtime, addip, boardtype, message, re)

Values ('pett', 'test', '2000-08-23 19:57:00', '192.168.1.11', '测试版', 'hehe',1)

在 SQL Server Query Analyzer 中执行这一条语句,相应的执行结果是:

1 row (s) affected

出现这样的结果,说明 SQL 语句已经成功执行,并且在数据表中添加了一行。

INSERT 语句也是前台的 ASP 程序中经常嵌套的 SQL 语句之一。在 BBS 论坛系统中, 新用户注册、发表文章、发送信件等需要往数据库中添加数据记录的操作都是通过 INSERT 语句来完成的。

INSERT 语句的一般语法是:

INSERT INTO table name

(field_name,...)

VALUES (field_value, ...)

使用 INSERT 语句需要注意以下几点:

- ◆ VALUES 清单的数值必须和字段的数据类型匹配, 否则 INSERT 语句将中断执行。
- ◆ 在 INSERT 语句中可以省略表中的一些字段,但前提是这些字段必须允许 NULL, 否则同样出现执行错误。
- ◆ 可以省略字段名的列表,也就是上面所示程序中的第二行,这是因为,SQL Server 将按照表中的固定字段顺序插入相应的值。一般不鼓励这样做。
- ◆ 字段名和相应的值一定要匹配。当字段名较多时,经常会出现匹配上的错误,这 在调试前台 ASP 程序的过程中是一种经常出现的错误。
- ◆ 凡字符串值都必须用单引号,否则将认为是一个变量,或者出现执行错误。对于 数字类型的值不需要加引号,否则将被认为是数字式的字符串。
- ◆ 这里插入的所有值都是确定的。一般嵌套在 ASP 程序内 SQL 语句的值有可能是 各种变量,程序的编写将更加复杂,需要在编写程序的过程中更加仔细认真,不 出现任何的书写错误。

注意:要想插入多行可以利用子查询。关于子查询的概念将在 3.6 节介绍。

在 BBS 论坛系统中,一些地方使用了 INSERT 语句,例如需要往数据表中插入新的记录,用户提交个人资料、发表文章、发送信件等。

比如,用户填写好个人信息之后,接受资料的页面中使用了 INSERT 的 SQL 语句,如 下所示:

insert into member

(lybid,lybname,nickname,password,birthday,question,answer,address,workunit,telephone,bpnumber,email ," &"icq,other,pen,addip,vtimes) "_____

&" values ("'&request.form ("lybid") &"',"*&request.form ("lybname") &"',"*&request.form ("nickname") &"',"*&request.form ("password") &"',"_ & ""*&request.form ("birthday")

&"',""&request.form ("question") &"',""&request.form ("answer") &"',""&request.form ("address") &"',"_ &"""&request.form ("workunit") &"',""&request.form ("telephone") &"',""&request.form ("bpnumber")

&"',"&request.form ("email") &"'," &""*&request.form ("icq") &"',"&request.form ("other") &"',"&request.form ("pen") &"',"&request.servervariables ("remote addr") &"',1) "

其中的 request.form 和各种连接符号都是 ASP 语句, 因为这条 SQL 语句是嵌套在 ASP 程序中的, 所以要满足 ASP 程序的语法。具体的解释见第 7 章。

3.5.5 两种建立数据库方法的比较

由上面的叙述可以看出,通过两种途径都可以新建数据库和数据表,并对数据库中的 数据进行一定的操作。在平时进行数据库操作时这两种方式都可以采用,如果对 SQL 语句 很熟悉,可以使用 SQL 语句的软操作完成指定的操作,方便快捷;对于 SQL Server 初学者, 最好还是使用 SQL Server Enterprise Manager 的图形界面来完成指定操作。

新建数据库或者数据表时,可以使用 Create 语句来完成,也可以通过图形界面来完成。 但使用 SQL 语句更加方便快捷,能够提高工作效率。尤其在创建多字段的数据表时,逐一 设置每个字段的数据类型、数据长度、允许空等属性,这种机械式的重复劳动让人感到十 分繁琐。而如果采用 SQL 语句,相同属性的字段可以使用复制、粘贴操作来减少输入的工 作量,这在一定程度上提高了工作效率。

在往数据库中加入数据记录时, INSERT 语句一次只能添加一条记录,如果需要输入 大量数据的话,这种做法显然不合理,而且需要注意 VALUES 清单中值和相应字段的数据 类型相匹配。这种情况下,建议使用 SQL Server Enterprise Manager 图形界面来添加数据记 录。

3.6 SQL 语言

SQL语言是位于加利福尼亚 San Jose 的 IBM 实验室于 20 世纪 70 年代后期开发出来的, 它的含义是结构化查询语言(Structured Query Language)。它最初是为 IBM 的 DB2 产品设 计的。SQL 语言不同于同时期出现的过程化语言,它是一种非过程化语言,它是建立关系 数据库管理系统(RDBMS)的基础。

注意: SQL 语言的所谓非过程化是指"是什么",而不是"怎么办"。在 SQL 语言中,描述的是检索、删除或者插入数据等,但不说明这些过程如何进行。

美国国家标准协会 (ANSI) 和国际标准化组织 (ISO) 一直在推动 SQL 标准的工业化。 SQL 语言最初成为 ANSI 标准是在 1986 年。1989 年的 ANSI 标准 (常被称为 SQL-89) 对 SQL 语言作了进一步的扩展和规范。后来随着计算机技术的发展和需要, SQL 语言还有 SQL-92, SQL-96 以及最新的 SQL-99 标准, 一般的 SQL-99 标准定义的 SQL 语言也称为 SQL-3。经过多次完善和补充, SQL 语言已成为操作和检索关系数据库的事实上的标准语 言。 注意:尽管标准化组织为数据库的设计者们制定了统一的标准,但几乎所有的数据库产品都和 ANSI 标准在一定程度上存在差别。因此,在采用 SQL 语言的不同实现方式时,应该注意它们之间的区别。

SQL语言的一个最大优点就是它是一种真正的跨平台语言和跨产品语言。它被编程者称为高水平的语言,只需要几行程序代码就可以完成大量的工作。

尽管从 SQL 的字面(Structured Query Language, 结构化查询语言)上讲 SQL 只是一种查询语言,但这一点往往引起误解。实际上,通过 SQL 语言不仅可以从数据库中查询数据,而且可以在数据库中创建表、增加数据、删除数据、将数据连接在一起,根据数据库的变化触发相应的处理过程,将用户的查询存储在程序和数据库中。归纳起来,一般 SQL 编程者和数据库管理员可以做以下的事情:

- ◆ 修改数据库的结构
- ◆ 改变系统的安全性设置
- ◆ 修改用户访问数据库表的权限
- ◆ 查询数据库的信息
- ◆ 更新数据库的内容

SQL 中最常用的语句是 SELECT 语句,同样,在 Web 数据库系统的开发过程中,内嵌 的 SQL 语句中最多的也是 SELECT 语句。SELECT 语句从数据库检索数据,并将结果提供 给用户。除此之外, SQL 还提供了创建新数据库、表、字段和索引的语句,以及插入和删 除记录的语句。ANSI SQL 还提供了一组数据操纵的核心函数。许多数据库系统也提供了 保证数据库完整性和增强系统安全性的工具。

注意:Web数据库开发者主要将 SQL 语句内嵌于 ASP 程序中,根据内嵌的 SQL 语句通过 ASP 中的数据库对象对数据库进行相应的操作。Web 数据库系统的客户 应用程序中,一般使用最多的是 SELECT 语句,也使用少量的 INSERT (插入数据)语句。但是在开发 Web 数据库系统的管理程序时,可能需要内嵌其他的 SQL 语句,如 CREATE (创建)、ALERT (更新)、DELETE (删除)、UPDATE (修改)等命令。最重要也最复杂的语句是 SELECT 语句,学好 SELECT 语句后,就 很容易掌握其他 SQL 语句。

3.6.1 使用 SELECT 语句进行数据库查询

SQL 的主要功能之一是实现数据库查询。如果你熟悉 Internet 搜索引擎,那么肯定熟 悉查询了。用户可以使用查询来取得满足特定条件的信息。例如,如果想找到有关 ASP 信息的全部站点,可以连接到 Yahoo!并执行一个对 Active Sever Pages 的搜索。在输入这个查询后,会收到一个列表,表中包括所有其描述中包含搜索表达式的站点。

多数 Internet 引擎允许逻辑查询。在逻辑查询中,可以包括特殊的运算符,如 AND、 OR 和 NOT,可以使用这些运算符来选择特定的记录。也就是说,用户可以用 AND 来限制 查询结果。例如,执行一个对 Active Sever Pages AND SQL 的搜索,将得到其描述中同时 包含 Active Sever Pages 和 SQL 的记录。

如果需要扩展查询的结果,可以使用逻辑操作符 OR。例如,需要执行一个搜索,查找所有的其描述中包含 Active Sever Pages OR SQL 的站点,这时,用户可以使用 OR 来得到包括所有其描述中同时包含两个表达式或其中任何一个表达式的站点列表。

如果想从搜索结果中排除特定的站点,可以使用 NOT。例如,查询"Active Sever Pages" AND NOT"SQL"将返回一个列表,列表中的站点包含 Active Sever Pages,但不包含 SQL。 当必须排除特定的记录时,可以使用 NOT。

用 SQL 执行的查询与用 Internet 搜索引擎执行的搜索非常相似。当执行一个 SQL 查询时,通过使用包括逻辑运算符的查询条件,可以得到一个记录列表。此时查询结果可能来自一个表或多个表。

SQL 查询的句法非常简单。为了说明 SELECT 语句的使用方法,我们使用在前面小节 中建立的 BBS 论坛数据库,此数据库中数据表的结构和字段如表 3-1 至表 3-4 所列,在后 面所述的 SQL 语句中都以这 4 个数据表为示例。

注意: SQL 语句其实十分容易,几乎完全是命令式的,需要什么样的结果只需要 输入一行命令就可以得到。但是,在使用的过程中还需要注意 SQL 语句的基本语 法结构。另外,本小节所举的 SQL 示例都以前台的 ASP 程序中所要嵌套的 SQL 语句为主。了解这些 SQL 语句,对于编写前台的 ASP 程序有很大帮助。

注意: 在 BBS 论坛的前台程序中,在 17 个页面中使用了 28 条 SELECT 语句,可 见 SELECT 语句在前台程序中的重要性。

SELECT 语句的作用是从单个或多个数据表中查询并显示数据。SELECT 是使用最频 繁的 SQL 命令。许多复杂的数据操作有时仅需要一条精选的 SQL 语句,这种 SELECT 语句的交互作用正是 SQL 语句的巨大功能所在。

SELECT 语句可以分为单表查询和多表查询,也可以在一个查询上构造另外一个查询。 这里首先介绍单表查询。

1. 单表查询

单表查询的 Select 语句基本语法如下:

SELECT	[DISTINCT] {* 字段名[,字段名,…]}
FROM	表名[或表单别名]
[WHERE	条件
[AND/OR	条件, [AND/OR 条件, …]]]
[GROUP BY	列表达式[,列表达式,…]
[HAVING	条件]]
[ORDER BY	{列,[,列,…]}]

方括弧中是非强制性的选项,括弧{}中是可选择的选项,而三连点…表示可替换前面的选项。SELECT 语句的附加子句隐藏在后面的部分中。从一个 SQL 语句到另一个 SQL

语句的执行过程中可能产生细微变化,但所有语句最起码应该包含这些必须的部分。 SELECT 命令在执行过程的顺序如下:

FROM	指定存取的表
WHERE	根据某些条件过滤行
GROUP BY	从一组行中构成一个单独的行
HAVING	根据某些条件过滤组
SELECT	指定输出哪些结果
ORDER BY	决定输出行的顺序

下面举一个最简单的 SELECT 示例来说明 SELECT 语句的概念。

查询 1: 列出 Message 表中的所有内容。

SELECT 语句:

Select * from Message

上面的 SQL 语句将显示出 Message 表中的所有行和列。在 SELECT 语法中,此 SELECT 语句是最简单的。这里的 Message 表是前面 BBS 论坛数据库中的用户信息表,其中存储所 有讨论区中的所有文章。

当然,编写好了的 SELECT 语句只有经过执行才能得到所需要的结果。SQL 语句需要 在 SQL Server 的引擎数据库中执行,执行 SQL 语句的界面就是前面所介绍的 SQL Server Query Analyzer。选择"开始"菜单,在"程序"中的 Microsoft SQL Server 程序组中选择 Query Analyzer,将启动 Query Analyzer,调试和执行 SQL 语句。

执行查询1语句的结果如图 3-55 所示。

and and the	State Services	arran and a said she was a she was a said	en la constante de la sec La constante de la sec		885245855.F.		en de la Salar de Cara e de Colora da Cara de C	Construction of the second
and the shift of		San Ser Sec.		1. A. B.	pettbbs			3. 2. 199 (* 1
ect * f	con message	4						
		CONTRACTOR AND	0.200 (N.C.)	57.0 MA	4024 M 12	of the second	22727-22	man and a
YP 4001624	17	a da transmissione	A SCOLLARS WHEN	Barris and Street State	(1).700 (1) (1)	CHEROLOGY WORK	Hall & Star week the second	
	Ganadaroa ang (152			Electrony and the			2000 08-22	10.57.00.000
	pecc			CR9C			2000-08-23	19:57100.000
	pecc			XDE dahati			2000-00-23	19:39:00.000
	• 19			CULARI			2000-00-23	19:59:00.000
	Dianyu			edz		1	2000-00-23	20100100.000
	. #JY						2000-08-23	20:00:00.000
	huhu			gdge			2000-08-23	20:00:00.000
	haha			dgegt			2000-08-23	20:00:00.000
	hehe			test			2000-08-23	20:00:00.000
	xixi			set			2000-08-23	20:00:00.000
	xixi			stet			2000-08-23	20:00:00.000
	-888			Ste			2000-08-23	20:00:00.000
	8	2010 12:00 8:00	and the second		34. A.	1		
Sec.		and the company of the second s	NOT TRADUCT OF THE TAX	CALL THREE AND AND A	4075-2000 clish-40/055	NO. 2 TO MARK MARK	CONTRACTOR OF AN AND A DECK	and the second se
Revalts G	rid and a start		200 an an 10 kg	10 10 200	and Constants	6. 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	A REAL PROPERTY OF	

图 3-55 查询 1 的执行

由于 Message 表中的记录很多,这里无法一一列出所有的记录,但通过这个实例,可 以对 SELECT 语句有一个初步的认识。 查询 2: 用户身份验证。 SELECT 语句:

Select * From member Where lybid=""&name&" and password=""&password&"

这是对登录用户身份进行验证的 SQL 语句,根据用户输入的用户名和密码,在用户资料表中查找符合条件的记录。如果语句的执行结果不为空,也就是说在数据表中存在符合条件的记录,说明数据库中存在当前用户的资料,登录的是 BBS 论坛的合法用户,根据这条语句的执行结果,ASP 程序会给出相应的处理。

本条语句中的 lybid 和 password 是 member 中的两个字段名,分别代表用户名和登录密码, 而单引号中的 name 和 password 分别是 ASP 程序中的两个变量, "&&"是对变量的一种引用方法。在把 SQL 语句输送到 SQL Server 数据库之前, ASP 程序会自动用变量的值来代替单引号内的内容。

假设目前登录用户输入的用户名和密码分别是 pett 和 123, 那真实的 SQL 语句将是:

Select * From member Where lybid='pett'and password='123'

注意: 这里的密码 123 虽然是数字,但其在数据库中的数据类型为字符串型,因此需要加单引号。这是因为,密码中可能包括字母。

这里的 Where 子句可以用来过滤不满足条件的所有记录。在上面的语句中使用了运算符 "="和逻辑连接符 "and"。

Where 子句中还可以使用下面的谓词:

- ◆ 运算符:=, ◇ (不等于), <, >, >=, <=。
- ◆ BETWEEN expression1 and expression2: BETWEEN 具有相容性,返回指定字段值 在表达式1和表达式2之间的所有记录。
- ◆ IN (element1, element2, ...): 返回和清单中指定元素具有相同值的所有记录。
- ◆ LIKE string_expression: 用于模糊匹配,可用的通配符有%、[]、_(下划线),比 如:

 LIKE '%pett%'
 表示任何包含 pett 的字符串;

 LIKE ' [A-M]%'
 表示首字母在 A 到 M 之间的所有字符串;

 LIKE ' [ABD]%'
 表示以 ABD 开头的所有字符串;

 LIKE ' [A-FU]%'
 表示首字母在 A 到 F 之间或者是 U 的所有字符串;

 LIKE ' [^p]%'
 表示首字母不是 p 的所有字符串;

 LIKE ' p_tt'
 其中的_可以是任何一个单独的字母,例如 pett 是满足条件的, 但 peett 不满足。

◆ 逻辑连接符 and 和 or: 用于连接多个搜索参数。

查询 3: 查出当前用户的访问次数。

SELECT 语句:

٠

select vtimes from member where lybid=""&session ("user") &"

这一条语句和查询 2 的语句很相似,不同的是在这里用 vtimes 代替了*,这里不返回 满足条件的记录的所有列,只返回 vtimes 字段。这里的 vtimes 字段是 member 表中的一个 字段名,含义是用户的访问次数。单引号内的 session ("user")是 ASP 的一个对象,具体 解释见 6.3 节 "session 对象"。

查询 4: 查询当前用户新收到的信件的数目。

SELECT 语句:

select count (receiver) as lnum from letter where receiver="&name&" and unread=0

这里的 letter 是信件表,其中存储的是 BBS 论坛用户之间的来往信件。这条语句的含 义是:计算 letter 表中 receiver 字段为"&name&"且 unread 字段为 0 的记录数目,并以 lnum 字段表示这个结果。这里的 count 是 SQL 合计函数,计算记录中 receiver 的数目,也就是 相当于计算符合条件的记录的数目。

SQL Server 中的函数很多,有字符串函数、数学函数、日期时间函数、系统函数、Niladic 函数以及合计函数。

SQL Server 中可用的合计函数如表 3-7 所示。

函数名	说明
AVG ([ALL DISTINCT] expression)	Expression 中所有值的平均值
COUNT([ALL DISTINCT] expression *)	Expression 中非 NULL 值个数。如果指定*则计算 NULL 个
	数
MAX ([ALL DISTINCT] expression)	Expression 中的最大值
VARP (expression)	Expression 中所有给定值的统计方差
STEDVP (expression)	Expression 中所有给定值的统计标准差
MIN ([ALL DISTINCT] expression)	Expression 中的最小值
SUM ([ALL DISTINCT] expression)	Expression 中所有值的总和
VAR (expression)	Expression 中所有给定值的统计方差
STEDV (expression)	Expression 中所有给定值的标准差

表 3-7 合计函数

通过添加关键字 DISTINCT,系统将只总计不同值,默认情况是 ALL。还有一种情况 是 COUNT (*),它计算满足条件的所有记录的数目。

注意:在 SQL 语句中使用合计函数,可以很方便地得到关于数据表中记录的各种统计数据。在 ASP 程序中,只需要直接嵌套 SQL 语句就可以,不再需要前台的程序进行任何处理。另外,合计函数并非只能用于 SELECT 语句中,其他语句中也能使用合计函数。

在查询 4 的 SELECT 语句中使用了 COUNT 函数,用来计算满足条件的记录数目,也
就是当前用户的新信件数目。

假设当前用户是 pett,则相应的 SELECT 语句是:

select count (receiver) as lnum from letter where receiver='pett' and unread=0

在 Query Analyzer 中的执行结果如图 3-56 所示。

[5] SUEScreent(uscyAnalyzer [Quecy wppetbls:W0YAdiometrator (initited) select cond(ms*]	國家國
	and a render
	P. 4. 841
select count(receiver) as inum from letter where receiver='pett' and unread=0	2
	31
	A
	1
	assistent a
Ernits Grid	A R Continues
	STREET, STREET
an a	

图 3-56 查询 4 的结果

如果删除语句中的 as lnum,此语句也能正常运行,结果中将显示出 No column name, 这在前台的 ASP 程序中不方便调用。因此一般在使用合计函数进行统计时,都使用 as 指 定统计结果的名称,以方便调用。

可以看出,使用合计函数可以很方便地对数据库中的数据记录进行各种实时统计。在 BBS 论坛中,需要一些常见的统计数字,以了解此论坛的规模大小。比如,当前注册人数、 截至当前有多少人访问本站、人均访问次数、日均访问次数等,统计这 4 项数字的 SELECT 语句如下:

SELECT count (lybid) as data1,sum (vtimes) as data2,abs (sum (vtimes)/count (lybid)) as data3,abs (sum (vtimes) /DateDiff ('dy','2000-8-24',getdate ())) as data4 from member

在这条语句中使用了 5 个 SQL 函数,即 COUNT()、SUM()、ABS()、DATEDIFF()、 GETDATE(),还有一个除法运算符/,其中的 COUNT()和 SUM()是合计函数,在上面已经 介绍过。ABS()是取绝对值,属于数学函数。SQL 中的数学函数如表 3-8 所示。其中所列 的数学函数根据输入值执行计算并返回一个数值。DATEDIFF()和 GETDATE()都是时间日 期函数, SQL 中的时间日期函数如表 3-9 所示。其中给定的表达式是时间数据类型。

函数名	说明	
ABS ()	计算绝对值	
ACOS ()	计算反余弦值	
ASIN ()	计算反正弦值	

表 3-8 数学函数

函数名	说明
ATAN ()	计算反正切值
DEILIN ()	大于或等于输入值的最小整数值
COS ()	计算三角余弦值
COT ()	计算三角余切值
EXP ()	计算指数值
FLOOR	小于或等于输入值的最大整数值
LOG ()	计算自然对数值
LOG10 ()	计算以 10 为底的对数值
PI ()	常数 pi
POWER (A,B)	A 的 B 次幂
SIGN (A)	A>=0,计算值为 1; A<0,计算值为 0
SIN ()	计算三角正弦值
SQUARE ()	计算平方值
SQRT ()	计算平方根
TAN ()	计算三角正切值

运算符+、-、*、/和%(取模)也都可以用于数值表达式。

表 3-9 时间日期函数

函数名	说明
DATEADD (E,I,B)	将整数 I 按照 E 数添加到 B 上
DATE (E,A,B)	A 和 B 之间差值的 E 数
DATENAME (E,A)	具有 A 的 E 名的字符串
DATEPART (E,A)	A 的 E 部分
GETDATE ()	当前的日期和时间
DAY (A)	A 对应的天数,以整数表示
MONTH (A)	A 对应的月份,以整数表示
YEAR (A)	A 中的年份, 以整数表示

表 3-9 中的参数 A 和 B 都必须是时间数据类型的数据, 而参数 E 则表示指定时间日期 数据类型的某一个部分。表 3-10 列出了可以指定的数据部分代码。

代码	说明	取值范围	
уу	年	1753~9999	
qq	季	1~4	
mm	月	1~12	

表 3-10 可以指定的数据部分代码

(续表)

代码	说明	取值范围
dy	一年中的某一天	1~366
dd	一月中的某一天	1~31
wk	星期	1~53
dw	工作日	1~7
hh	小时	0~23
mi	分	0~59
SS	秒	0~59
ms	毫秒	0~999

注意:使用 SQL 中的函数时,一定要注意要求的数据类型和返回的数据类型,使用不符合条件的数据类型进行运算,将出现运行错误。在嵌套到 ASP 程序之前,必须在 SQL Server Query Analyzer 中进行调试,能够正确执行之后再插入到 ASP 程序中。

除上面所列 3 种类型的函数之外, SQL Server 中还有字符串函数、系统函数以及 NILADIC 函数, 这里不再介绍, 需要时可以参考相关的书籍。

如果数据类型允许的话,SQL 中的函数可以嵌套使用,比如统计日均访问人次的 abs (sum (vtimes) /DateDiff ('dy','2000-8-24',getdate ())),其中就有三层函数嵌套。

查询 5: 查询属于某一个讨论区的所有文章,并按时间降序排列。

SELECT 语句:

select * from message where boardtype=""&session ("board") &" order by addtime DESC

这条 SQL 语句中除前面已经介绍过的 where 子句外,还使用了 order by 子句。order by 子句将记录集按指定的一个字段或多个字段排序。默认排列顺序是升序,如果希望按降序 排列,则需要添加 DESC 来指定。如果希望使记录集具有一定的顺序,那么需要在 SQL 语 句中使用 order by 子句。order by 后所跟的是一个或多个字段名。

假设指定的讨论区是"休闲娱乐",则相应的 SELECT 语句为:

select * from message where boardtype='休闲娱乐' order by addtime DESC

执行结果是返回 message 表中所有属于"休闲娱乐"讨论区的文章,并按照发表时间 的降序排列,这样可以保证用户浏览时首先看到的是最新的文章。

查询 6: 查询出属于某一讨论区的所有文章,同一主题的文章放在一起,并按时间的 降序排列。

SELECT 语句:

select * from message where boardtype=""&session ("board") &" order by title,addtime DESC

(续表)

查询 6 和查询 5 的 SQL 语句十分相近,不同的是 order by 子句中的字段名,查询 6 是 以 title 和 addtime 排序的。也就是说,首先按主题的降序排列,这样同一主题的文章将放 在一起,主题是按照首字的首字母顺序排列的,先英文主题后中文主题;在同一个主题组 内,所有文章都是按时间先后顺序降序排列。这在 BBS 论坛中是用户可选的一种阅读方式。

注意:使用 order 子句按照指定的多个字段进行排序时,最重要的是考虑各个字段的先后顺序,先按照哪个字段,然后再按照哪个字段,位置颠倒的话,查询的结果 会有很大差别。

BBS 论坛中还有一些类似的 SQL 语句。比如,用户的个人信箱显示页,其中需要显示 出当前用户所有收到的信,并且要标明每一封信的状态(已读或者未读),相应的 SQL 语 句如下:

select * from letter where receiver=""&session ("user") &" order by addtime

这个查询语句与查询 6 的语句十分相似,只有 where 子句和 order by 子句中的字段不同而已。

查询 7: 查询 BBS 论坛中当前的十大讨论话题。 SELECT 语句:

select top 10 count (addid) as tnum, title, board type from message group by title, board type order by tnum DESC

这是一个比较复杂的 SELECT 语句,其中包含了 group by 和 order by 子句。

关键字 top 用来限制返回的行数。在本查询中,只要求返回 BBS 论坛中当前的十大话题,因此使用 top 10 来限制仅仅输出前 10 条记录,也就是所需要的十大话题。在上面的 SELECT 语句中,首先需要统计各个话题在各个讨论区中的文章数(这里不考虑属于不同 讨论区的同一话题),因此这里首先使用 group by 子句进行分组,分组的字段是 title 和 boardtype,这里的 title 和 boardtype 字段的先后顺序并不十分重要,最终按照文章数进行排序之后,两种先后顺序得到的结果是相同的。

group by 子句用来对数据表中的记录进行分组,以便于对各种类型的数据进行统计。

2. 多表查询

查询 8: 列出各个讨论区的情况,包括讨论区名称、版主、简介、当前文章数目。 SELECT 语句:

select boardname,mesnum,bdowner,bdintro from (SELECT boardtype,count (*) as mesnum from message group by boardtype) a,board b where a.boardtype=b.boardname order by mesnum DESC

此查询所需要的数据来自两个表中,分别是 message 表和 board 表。这需要涉及到多 表查询;另外,这个语句中使用了子查询,如图 3-57 所示。

lect boardness man	num bdormer bdintre	from (SELECT boardtune o	
ssage group by bos	rdtype) a, board b	where a boardtype=b.boardr	name order by mesnum DESC
and the state of the second		and the second	readility of which came and some second to
<u> Securitania</u>	14.14.17 (S.1.14) (S		
ale e service de la companya de la c De la companya de la c		bildman	
闲嫨乐	14	hone ydue	autorite 传递娱乐动态,丰富
闲娱乐 站讨论	14 10	backart honeydue xinrr	有通娱乐动态,丰富 讨论本站的建设情况
闲嫨乐 站讨论 试版	14 10 4	bacear films honeydue xinrr benfox	有道规乐动态,丰富 传递娱乐动态,丰富 讨论本站的建设情况 这是本站的测试版,
闲娱乐 闲娱乐 试版 会信息	14 10 10 4 4	honeydue xinrr benfox guyajin	1440000 传道娱乐动态,丰富 讨论本站的建设情况 这是本站的潮试版, 发布各种社会信息,
闲娱乐 站讨论 试版 云信息 性感性		bonegue kinrr benfox guyajin bruckmer	传道娱乐动态,车置 讨论本站的建设情况 这是本站的潮试版, 发布各种社会信息。 如任感性
用媒乐 站讨论 试版 会信息 性感性 酶技术		honeydue xintr benfox guyajin bruckner realowen	有道规系动态,本言 待道规系动态,本言 讨论本站的建设情况 这是本站的潮试版, 发布各种社会信息, 如性感性 讨论计算机科学的发
用漢系 站讨论 試版 会信息 性感性 脑技术 仏教育	14 10 1 1 3	bottomat. honeydue xinrr benfox guyajin bruckner realowen pett	■10年20 传道娱乐动态,丰富 讨论本站的建设情况, 这是本站的潮试版, 发布各种社会信息, 知性感性 讨论计算机科学的发 文化和教育
構築系 始け论 式版 合信息 性感性 酶技术 化教育	14 10 4 3 1	boneydue koneydue xinrt benfox guywjin bruckner realowen pett	特遇援承动态,丰富 特遇援承动态,丰富 讨论本站的建设情况, 这是本站的潮试版, 发布各种社会信息, 知觉感性 讨论计算机科学的发 文化和教育
和娱乐 始讨论 试版 会信息 性感性 微性 化較育		honeydue xintr benfox guyajin bruckner realowen pett	●通知 ● ●通知 和 本 本 前 本 本 前 書 讨论本 本 的 准 读 使 很 这是本 本 的 潮 试 服 , 发 布 各 种 社 会 信 息 , 如性 感性 讨论计算机科学的发 文化和 軟育

图 3-57 查询 8 的执行结果

这个 SELECT 语句的设计过程可以分下面两个步骤:

(1) 首先统计各个讨论区中的文章的数目,也就是上面语句中()的部分:

SELECT boardtype,count (*) as mesnum from message group by boardtype

很容易理解这一条语句的构造,其中使用了合计函数 COUNT()和一个 GROUP BY 子句。具体的查询过程可以描述为:首先按讨论区分组,然后统计每一组中的记录条数,以 mesnum 字段返回。执行结果如图 3-58 所示。

	sype, count	(*) a.s m	eshum I	rom messi	nde Gront	ру роал	acype
in the second of			e contra		1		M
2000-01-02-02-02-02-02-02-02-02-02-02-02-02-02-	10 You down 10 You do 10 You		10				
试版			4				
自脑技术			1				
L 会信息			4				
(化教育			1				
闲娱乐			14				
0件感性			3				

图 3-58 子查询的执行结果

从图 3-58 下面的窗格中可以看到子查询的执行结果,生成的结果和一个数据表的构造 是一样的,有字段名和记录(列和行)。一个查询生成的结果可以称为一个视图,视图和表 几乎一样,可以对其进行进一步的查询。

(2)结合子查询的结果从 board 表中查询各个讨论区的相关内容,即 bdowner、bdintro、 mesnum、boardname。这一步骤涉及到两个表之间的查询,关键是要求各个讨论区的信息 与文章数目统计相匹配,因此在查询语句中使用了一个 where 子句。

select boardname, mesnum, bdowner, bdintro from (SELECT boardtype, count (*) as mesnum from message group by boardtype) a, board b where a boardtype=b.boardname order by mesnum DESC

在语句中,为了调用方便,给子查询生成的视图和 board 表分别定义了一个别名 a 和 b。 在 where 子句的条件中,判断的字段必须标明是哪个表或者视图中的字段,一般用 table name.column name 来表示。

在这里,调用子查询的方法和调用一个表的方法是一样的,只需把子查询的语句用括 弧()括起来就可以了。利用子查询可以完成复杂的统计查询,把复杂的查询分成若干步 骤来实现。

设计带有子查询的查询时,首先需要调试子查询语句,然后把子查询生成的结果(视图)当作一个表来对待,再进行相关的查询。对两个表进行相关查询要比单独对一个表进 行查询要复杂一些,最重要的是要标明字段的来源。

查询 9:查询出文章总数大于 10 篇的讨论区。

SELECT 语句:

SELECT boardtype,count (*) as mesnum from message group by boardtype having count (*) >10

这一语句和查询 8 中的子查询很相近,不同的是,这里添加了一个 HAVING 子句用于 限制输出的行数,限制只输出文章数大于 10 的讨论区。

执行的结果如图 3-59 所示。

boardtype 부터문학 14

图 3-59 查询 9 的输出结果

这里使用的 HAVING 子句很简单,用来限制返回的统计数字。在前面的查询中,用来 作限制条件的子句是 where 子句,这里不能使用 where 子句。where 子句只能根据表中的字 段值来作一定的限制,不能对统计结果作限制。如果执行语句:

SELECT boardtype,count (*) as mesnum from message group by boardtype a where mesnum>10

那么在 SQL Server Query Analyzer 中将显示出如下的错误信息:

Server: Msg 156, Level 15, State 1, Line 1

Incorrect syntax near the keyword 'where'.

错误信息的第一行表示错误的等级。对于 SQL 语句执行过程中出现的错误, SQL Server

都会报出其等级,以表示此错误的严重程度,具体解释可以参考关于 SQL Server 的书籍。 第二行的含义是关键字'where'出现错误语法,这是 SQL 初学者容易犯的错误,由于 mesnum 不是 message 表中的字段,因此 where 子句出现运行错误。

如果要使用 where 子句,可以通过子查询来实现,相应的 SQL 语句如下:

select * from (SELECT boardtype,count (*) as mesnum from message group by boardtype) a where a.mesnum>10

这里是把子查询的结果当作一个视图来对待,并且定义了一个别名 a,这里不定义别名 a 的话,语句执行时将出现运行错误。其中的 where 子句的条件可以改写为 where mesnum>10,因为此语句中只涉及到一个表,字段名可以不表明来源,但 where 前面的 a 是不可省略的,必须定义视图的别名。

3.6.2 对数据表进行更新和删除操作

BBS 论坛中提供个人资料的修改功能,在页面中,需要对数据表中当前用户的记录进行更新,可以使用 UPDATE 语句。

1. 使用 UPDATE 语句对记录进行更新

UPDATE 语句非常简洁明了,只需要指定所更新的表、要更新的字段、字段的新值以 及相应的行。

在用户登录 BBS 论坛时,将使当前用户的来访次数加1。例如,使用户 pett 的来访次数加1,相应的 UPDATE 语句为:

update member set vtimes=vtimes+1 where lybid='pett'

如果省略其中的 where 子句, 将更新 member 表中的所有记录。

2. 使用 DELETE 语句删除记录

要从一个数据表中删除记录,可以使用 DELETE 语句。例如,删除用户 pett 的个人资料,相应的 DELETE 语句是:

delete member where lybid='pett'

同样,如果省略其中的 where 子句,则将删除掉 member 表中的所有记录。因此,需要小心使用 delete 语句,以免造成无法恢复的数据损失。

完整的 SQL 语法很多,在有限的篇幅内,不可能进行详细的解释和说明,读者可以参照有关 SQL 的书籍。另外,附录 A 中列出了 SQL 语法的参考。

3.7 管理 SQL Server 数据库

数据库管理是一个很复杂的概念,真正管理好一个数据库需要做各个方面的工作,包

括数据库文件管理、日志文件管理、数据库安全、登录用户管理、数据备份等等。其中的 数据备份是数据库管理的日常工作之一,也是很重要的一项工作,将在 3.8 节详细介绍。 这一节主要介绍数据库安全和登录用户的管理。

3.7.1 SQL Server 中可用的向导

对数据库进行管理时,一般都使用 SQL Server 管理工具 Enterprise Manager 来实现。 关于 SQL Server Enterprise Manager 可以参照 3.3.2 节的介绍,其中有很多向导(Wizard) 可以帮助管理者方便地完成各种管理操作。在 Enterprise Manager 的视窗中选择 Tools 菜单 的 Wizard 选项,将弹出 SQL Server 中所有可用的向导,如图 3-60 所示。



图 3-60 可用的向导(Wizard)

从图 3-60 中可以看到总共 4 种类型的向导:数据库向导(Database)、数据转换服务向导(Data Transformation Services)、管理向导(Management)和复制向导(Replication)。 在前面的小节中建立数据库时,已经使用了新建数据库向导。在建立数据库的过程中,可以设置数据库文件和事务日志文件的存放路径和大小限制。在添加数据库的登录用户时, 需要使用登录向导(Create Login Wizard)。进行数据库备份时,需要使用备份向导(Backup Wizard)。进行数据输入输出时,需要使用数据输入向导(DTS Export Wizard)和输出向导 (DTS Import Wizard)。

3.7.2 数据库文件和事务日志文件的管理

一般在新建数据库的时候,已经设定了数据库文件和事务日志文件的一些参数。当数 据库建立并使用之后,对这些文件仍可以重新进行参数设置。

在 Enterprise Manager 的视窗界面中双击指定数据库的标志,或者用鼠标右键单击指定数据库的标志,在弹出的选项框中选择"属性"选项,都将弹出指定数据库的属性对话框。 假设指定的数据库为 pettbbs,则弹出的对话框如图 3-61 所示。



图 3-61 pettbbs 数据库属性对话框(pettbbs Properties)

和建立数据库时的界面(见图 3-38)不同,这里多出了两个选项: Options 和 Permissions。 其中的 General 和 Transaction Log 选项的设置方法和新建数据库时一样,这里不再重复。 Options 选项中可以设置存取(Access)和设置(Setting)参数。

Permissions 选项中可以设置登录用户对数据库的各种操作权限。

3.7.3 数据库安全

SQL Server 的安全性是多方面的,包括网络操作系统和 SQL Server 本身。任何人访问 SQL Server 或 SQL Server 数据库存储的数据前,SQL Server 都需要鉴别登录用户的身份,以确定试图登录的用户是不是一个已授权的用户。如果不能通过身份验证,SQL Server 就 拒绝访问;如果通过,则允许访问。

SQL Server 支持两种不同类型的用户鉴别: Windows NT Authentication (Windows NT 鉴别)和SQL Server Authentication(SQL Server 鉴别)。如果使用 Windows NT Authentication,则采用 NT Server 的内置安全性来鉴别用户。如果使用 SQL Server Authentication,那么表示采用 SQL Server 中的内置安全性来鉴别用户。

数据库管理中的安全模式就是指 SQL Server 鉴别登录用户身份的模式。SQL Server 提供了两种安全模式: Windows NT 鉴别方法和 SQL Server Mix Authentication (SQL Server 混合鉴别安全模式)。模式和鉴别方法是两个不同的概念,不要相互混淆。

1. 登录 ID

某一位用户想要访问 SQL Server 时,这个用户必须以某种形式提供一个登录 ID。这个 登录 ID 好比是打开 SQL Server 的钥匙,它使用户能够访问由 SQL Server 管理的特定数据 库。如果用户账号使用 Windows NT 鉴别,那么用户的 SQL Server 登录 ID 就是 NT Server 的登录 ID。如果用户的账号使用 SQL Server 鉴别,则必须手工输入登录 ID 和密码才能访 问 SQL Server。

2. 数据库用户 ID

登录用户在访问某个由 SQL Server 管理的数据库之前,必须具有这个数据库对应的数据库用户 ID。每一个单独的数据库包括一个系统表,这个系统表包含了能访问这个数据库的所有数据库用户 ID。如果某个用户没有访问某个数据库的数据库用户 ID,则不能访问这个数据库。一般情况下,用户的登录 ID 和数据库用户 ID 是一样的。

用户登录 SQL Server 时必须通过以下 4 个步骤 (见图 3-62):



图 3-62 用户登录 SQL Server 时必须通过的四个步骤

步骤1 用户通过的第一个步骤是用户网络操作系统。尽管这个系统一般是 NT Server, 但也可以是任何能与 NT Server 和 SQL Server 共存的网络操作系统。一般情况下,用户必须提供一个有效的网络账号和密码来接受鉴别和登录到网络上。

步骤 2 用户访问了网络之后,下一步就是用 Windows NT 鉴别或者 SQL Server 鉴别 来鉴别用户。要使用 SQL Server 鉴别用户,必须提供一个有效的登录 ID 和密码给 SQL Server。如果是用 Windows NT 鉴别用户,那么用户的 NT Server 网络账号就是 SQL Server 登录 ID。这也就是 SQL Server 和 NT Server 之间无缝结合的表现。实际上,用户不必输入 登录 ID, SQL Server 能自动确定登录 ID。

步骤3 用户能够登录和访问 SQL Server 并不意味着就可以访问 SQL Server 管理的任何数据库。因此,下一步用户必须接受所要访问的数据库的鉴别。只有在数据库系统表中存在 ID 的用户,才能访问这个数据库。SQL Server 数据库的管理者可以给某一位用户分配访问所有数据库的权限,或者只给用户分配访问一两个数据库的权限。

步骤 4 使用户有权访问某个数据库,还是不能自动授权给用户访问存储在数据库中的数据,用户必须有特定的权限,并且权限和对象必须一一对应。

可以看出,在访问 SQL Server 数据库时,用户必须通过许多步骤。尽管这4个步骤使 未经授权的用户很难访问数据库,可以保证数据不会轻易遭到破坏,但也增加了设置 SQL Server 安全性的难度。根据对 SQL Server 数据库安全性的要求,设置安全性有可能是一件 比较复杂和耗时的工作。

在实际操作中,作为以 NT Server 管理者(Administrator)登录 SQL Server 的管理者, 在登录过程中并不会遇到这么多复杂的步骤,因为 SQL Server 的安全性可以受 NT Server 来保护。

注意:这里所讲述的 SQL Server 数据库用户和 Web 数据库系统中的用户是两个不同的概念。这里的 SQL Server 数据库用户是指直接对数据库进行各种操作的用户。 而 Web 数据库系统中的用户则是通过浏览器来对数据库进行访问,并不能随意对 数据库进行操作,只能通过前台程序提供的功能来完成数据的操作。

3.7.4 建立 SQL Server 数据库用户

很多情况下,在进行协同开发中需要多人合作对数据库进行管理和操作,但不允许多 人拥有管理者等级的权限,那样容易造成数据管理上的混乱。必须建立一些权限等级较低 的用户。下面介绍如何在 SQL Server Enterprise Manager 中新建登录账号。

(1) 首先通过 SQL Server 程序组启动 Enterprise Manager。在界面中展开 Security 选项,选择其中的 Login 选项,在右面的窗格中将显示出 SQL Server 数据库的所有当前用户 ID,在其中单击鼠标右键,将弹出如图 3-63 所示的选项对话框。

New Logina .
与出列联()
NATERCO
帮助(出)

图 3-63 新建登录 ID 选项对话框

(2) 在如图 3-63 所示的选项对话框中选择 New Login (新的登录 ID 选项),将弹出 如图 3-64 所示的对话框。

(3) 在图 3-64 所示的 General 选项中需要输入用户账号,并在 Windows NT Server 鉴别模式和 SQL Server 鉴别模式中选择一种身份鉴别模式。如果选择前一种模式,还需要选择所属的 NT 域,并设置安全性;如果选择后一种模式,则需要输入用户密码,单击"确定"后,会弹出确认(Confirm)密码框。在默认值中,需要指定默认的语言和数据库,根据当前用户需要进行相关的设置。正确设置后,单击最上方的 Server Roles(服务角色)选项,如图 3-65 所示。

SQL Server Login Propert	ies - New Logín	2 2 B
		1997 - 199 1 (* 1997)
WERNIGHT AL Server Belat	Deteber Access	
14 June Press Press	DATE REAL PLAN	
	用户账号10	
	ten in de la compañía	
Authentigati 2-16		
C Madere B	anthantinet las find	
Star Starity		
C	at access 2700	
C los		
	of all and the relation of	
	Charles and the second second	rene la serie de la serie d
C SOL Serve	entheatization SQL 3	
Eastword:	日本 語為基礎的語言。	and the second
a state of the second		the second second second
	a de view contraints a second	1 1 - S - S - S - S - S - S - S - S - S
	Real Strength Court and	some toy use
	aster.	
		And the second second
Lengunge:	(Default)	• 100 a
		and she water
		i f mm f
ALL PROPERTY.		

图 3-64 General 选项



图 3-65 Server Roles 选项

(4) 在如图 3-65 所示的 Server Roles 选项中设置新建用户的角色权限,比如其中的系统管理权限(System Administrator)等等。当光标移动到某一选项上时,在下面的描述框(Description)中就有相应的说明。当需要分配给新建用户某一权限时,选中选项前面相应的方框就可以了。设置好相应的权限之后,单击上面的 Database Access 选项,切换到如图 3-66 所示的数据库访问界面。

SQL Server Logi	n Properties - New Login	
General Ser	ur Lales Detaine Access	
Detabers		Contraction of the second
A 30	cify which databases can b	a accound by this
	Batohus Bow	
	model	
	i 🛐 msdb	
	pettbbs1	
CANE C	D pubs	
] 🗗 tempdb	
Det	abase rolas for 'pettbbs':	
1	rmit in databare role	· · · · · · · · · · · · · · · · · · ·
	🗗 public	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	db_owner	
) ab_accessadmin dh securitzedmin	
	db ddladmin	· Propertake

图 3-66 Database Access 选项

(5) 在如图 3-66 所示的界面中,允许指定新建用户可以访问的数据库,可以选择多 个数据库,当选中某一数据库之后,在下面的数据库角色中对所选定的数据库指定角色。 在上面的窗格中列出的是当前系统中的所有数据库,在下面的窗格中列出的是公共用户 (public)、数据所有者(db_owner)等角色。同样,根据新建用户的需要,正确设置好对 各个数据库的访问权限,设置完毕后,可以再切换到另外的两个选项,对前面的设置进行 一定的更改。所有设置完成之后,单击"确定"按钮。

(6) 如果在 General 选项中选择的是 SQL Server 模式,则出现密码确认对话框,在 Confirm New Password 框中再次输入设置的密码,然后单击 "OK" 按钮。

完成上面的步骤之后,就建立了一个新的登录用户。在右面的窗格中将出现新建登录 用户的列表。如果需要对用户的权限和角色重新进行设置,则可以选中用户,然后单击右 键,选择"属性"选项,就可以重新进行设置。

在 SQL Server 的向导(Wizard)中有一个 Create Login Wizard(新建登录用户向导), 如图 3-60 所示,使用这个向导可以更明了地新建登录用户,但建立的步骤比直接在 Enterprise Manager 中要多。了解新建登录用户需要注意的事项后,不必要再使用向导来新 建登录用户。因此,对 SQL Server Create Login Wizard 不再具体介绍。

在 Web 数据库系统的开发过程中,新建 SQL Server 数据库登录用户是必须的,在前台的 ASP 程序与数据库连接时,同样需要用户身份认证。出于安全性考虑,一般不会把系统管理员的账号和密码插入在前台的 ASP 程序中,每一个单独系统中的用户只需能访问相应的后台数据库就可以了,并不需要对其他数据库操作的任何权限。因此,一般情况下,需要给 SQL Server 中的每一个数据库新建一个专门用户,以供前台 ASP 程序访问数据库。

3.8 数据备份

造成数据库丢失或损坏的原因一般有以下几种:硬盘损坏或失窃,管理员不小心进行 了不适当的删除操作,某些不法分子的恶意破坏,自然灾害,计算机病毒等。所以,针对 不同的情况,我们有必要同时进行好几种方式的备份。

3.8.1 数据备份的几种基本方式

1. SQL Server 数据库的备份

SQL Server 的备份总共有 4 种方式:

- ◆ 完整数据库备份。就是将某个数据库整个备份成一个备份文件。
- ◆ 事务日志备份。事务日志是一个单独的文件,它记录了自上一次完整数据库备份 或事务日志备份以来,对数据库数据的所有修改。如果数据库较大,不便于进行 频繁的完整数据库备份,就可以在进行一次完整数据库备份之后只安排事务日志 备份。
- ◆ 差异备份。差异备份只备份那些自上次完整数据库备份以来所改变的部分,它的 优点是备份和恢复速度都比较快(相对事务日志备份和完整数据库备份而言)。
- ◆ 数据库文件备份。在 SQL Server 中,一个数据库可以由好几个文件构成,这些文件可以分别存放在不同的驱动器中,这样既可以使不断增大的数据库不必受硬盘空间的限制,又可以提高数据检索的效率。当一个数据库非常大,一个晚上都备份不完时,就可以把备份工作分成几次进行,每次只备份一个文件。

2. 利用 NT Server 进行备份

除了使用 SQL Server 来备份外,我们也可以使用 NT 提供的备份工具。在防止数据丢 失方面,NT 的磁盘管理器具有强大的功能,它支持 RAID (廉价磁盘冗余阵列)的第 0、1、 5 级。

所谓廉价磁盘冗余阵列,是指将多个小容量的廉价驱动器(例如我们所用的普通硬盘) 组合在一起使用,通过数据的冗余来提高安全性。

- ◆ RAID 0 级是指将多个驱动器上的自由空间组合成一个无校验的带区集,在带区集上保存数据时,即使某一个成员能保存整个文件,一个文件也绝对不会仅仅分布在一个成员上。
- ◆ RAID 1 级是指把一个驱动器上的某一分区在另一个驱动器上建立一个完全的镜像。进行写操作时,数据将向两个磁盘中写入一模一样的数据,读取时可以从两个磁盘同时读取。这样做的好处是显而易见的,当一个驱动器损坏时,可以由它的镜像来进行恢复。
- ◆ RAID 5 级是指有校验的磁盘带区化。类似于 RAID 0 一样进行带区化,但在数据

写入带区集时,数据的校验信息也将被同时写入,并且校验信息总是与相应的数据存放在不同的带区成员上(也就是存放在不同的驱动器中),这样,当磁盘阵列中一个磁盘发生故障时,该盘上的数据可根据其他盘上的校验信息进行恢复。

具体的磁盘带区化和 RAID 各级的实现,可参考有关 NT Server 的书籍。

除 RAID 外,我们也可以使用 NT 提供的备份工具来设置计划进行备份,NT 的备份工具在"附件">"系统工具"中。

3. 第三方备份工具

除了使用以上两种备份方式外,我们还可以采用别的公司开发的备份软件。笔者常用 的是 Norton 公司的磁盘备份工具——Ghost。该软件可以直接将磁盘上的某个分区或整个 硬盘克隆成一个镜像文件(可压缩),然后只要我们把这镜像文件存放在别处,那么当该分 区或硬盘出现问题甚至毁坏时,就可以使用 Ghost 在另一硬盘或分区上快速地把这个镜像 文件还原,并且保证跟原来的一模一样。

3.8.2 BBS 论坛数据备份策略

首先需要分析 BBS 论坛数据库的大小。BBS 论坛数据库中并没有太大的信息量。假设 把我们这个 BBS 论坛数据库发布到因特网上供用户访问,这里来计算一下一年的信息量。

以每天 1000 篇文章计算,假设每一条记录平均包含 500 个中文字符(每个英文字符和 数字占一个字节,每个中文字符占两个字节),则每一条记录平均占 1KB,1000 条记录则 占 1M。也就是说,BBS 论坛数据库中每天记录增长需要的空间是 1M,那一年总共需要的 空间就是 365M。另外,还有其他几个表以及事务日志。所以,BBS 论坛数据库每年需要 的磁盘空间最大不会超过 1G。BBS 论坛数据库的管理人员可以随时删除一些垃圾记录,以 节省磁盘空间。

从规模而言,这个 BBS 论坛数据库属于一个小型数据库,因此执行一次完整数据库备 份不会耗费多少时间。为了提高安全性,SQL Server 要求事务日志只能备份在与完整数据 库不同的带区成员中。也就是说,事务日志备份需要 RAID 来支持,必须拥有至少 3 个硬 盘,才能实现 RAID 0 级,才能进行事务日志备份。因此,就 BBS 论坛数据库来说,进行 事务日志备份既不易实现也没有太大的必要。

笔者在这里建议,在 SQL Server 中设定备份计划,每天进行一次完整数据库备份(最 好把备份时间设置在夜间用户访问量最少的时候,这样可以在最小的程度上影响用户对数 据库的访问)。在必要的阶段,可以增加完整数据库备份的频率,可每天进行几次备份。假 如因为某种原因后台数据库出现了致命错误,也可将数据库恢复至一天前的状态,最多只 会损失掉一天的文章记录。

此外, SQL Server 对数据库进行完整备份时是生成一个备份文件, 默认存放在 SQL Server 安装目录 (通常名叫 MSSQL7)下的 BACKUP 文件夹中 (当然也可以在别的分区或 目录, 但不能是映射的网络驱动器)。所以,为了确保数据安全,可以利用 NT Server 或 Windows 2000 的备份工具,在每一次 SQL Server 完成完整数据库备份后,把生成的备份文 件在另一台计算机上再复制一份。这样,即使其中一个备份文件遭到损坏,仍可用它在另

一台机器上的拷贝来恢复。

上面所说的只是利用同一个 SQL Server 对数据库进行备份。但是,如果本地计算机上的 SQL Server 系统发生了崩溃,完全不能恢复,又该如何挽救呢?

解决这个问题有两种方案:

- ◆ 使用论坛数据库的备份文件在另一个 SQL Server 中进行恢复。可以在本地计算机 重新安装一个 SQL Server 系统,然后用论坛数据库备份文件的一个拷贝进行恢复, 在新的 SQL Server 中生成一个和原来一模一样的论坛 fengzh 数据库。
- ◆ 事先在网络上的另一台计算机上安装一个新的 SQL Server,然后使用 SQL Server 的"数据转换服务"功能(Data Transfer Service)将论坛数据库整个导出至新的 SQL Server 中。这样,两个 SQL Server 中就都具有了完全相同的论坛数据库。

显然,第二种方法实际上是又一种备份方式,而且这无疑是最好的一种备份方法。这 是因为,当原来的论坛数据库(甚至是整个 SQL Server)崩溃后,可以直接启用另一个 SQL Server 中的论坛数据库,对于整个 Web 数据库系统所要做的只是修改一下发布网页的服务 器上 ODBC 数据源中所设置的 SQL Server 主机名称(关于 ODBC 数据源的设置见 6.6.2"设 置 ODBC 数据源")。

这两种方法在 SQL Server Enterprise Manager 上可以很方便地实现。

注意:无论是哪一种方法,都必须先在新的 SQL Server 中新建一个和原来论坛数 据库所使用的一模一样的登录 ID(也就是前一节中建立的 User 用户),否则数 据库的导出将无法顺利进行!

3.8.3 备份及恢复过程说明

下面详细介绍上一节提到的备份过程。无论哪一种,都需要先确认 SQL Server 服务是 否已经启动(默认为自动启动),如果没有,则使用 SQL Server 7.0 程序菜单中的"SQL Server Service Manager"进行启动。然后打开 SQL Server 的"Enterprise Manager"。

1. 在本地机上进行数据库备份

把"Enterprise Manager"左边的 SQL Server Group 展开,将显示该 Group 中已在此 SQL Server 中注册了的所有 SQL Server(包括自己)。现在以本机上的 SQL Server 为例进行说明。 展开 SQL Server 中的"Databases",单击右键,如图 3-67 所示,在出现的菜单中选择

"所有任务"选项中的"Backup Database"子选项,将出现如图 3-68 所示数据库备份画面。



图 3-67 Backup Database 级联菜单



图 3-68 数据库备份

选好要备份的数据库并选中"Database-complete"(完整数据库存备份)后,单击"Add" 按钮,弹出图 3-69 所示的备份设备选择对话框。

Choose Backup Destination	100
Choose the file name or backup device to u	36
for the backup operation. Beckup deviced	Cath .
be created for files that you use frequent	
The season of th	
	C States -
d:\server2000\mssql7\BACKUP\	1226
Contraction Meride: 1 (1997) - 1997 - 1997 - 1997	Let Z
	SCAT LAN

图 3-69 选择备份文件的存放目录

设置完毕单击"OK"按钮,将回到图 3-68 所示的界面。注意图 3-68 界面中的"Overwrite" 属性,第一个选项"Append to media"是指每执行一次新的备份时,以前的备份都将保留; 第二个选项"Overwrite existing media"是指新的备份将会覆盖掉原来的备份。显然,如果 选第一项,随着时间的推移和备份次数的增加,备份文件会越来越大,因为它包含了数据 库在不同时间的每一个备份;如果选第二项,备份文件的大小就始终与原数据库大小同步。

设置完毕,单击"确定"按钮,完整数据库备份就开始执行了。如果要设置定期自动 备份,则可在图 3-68 的"Schedule"中具体设置。单击"Schedule"按钮,将弹出如图 3-70 所示的备份计划设置对话框。



图 3-70 指定备份计划对话框

在图 3-70 中,有4 种可以选择的备份类型:

- ◆ 当 SQL Server Agent 启动时进行备份(Start automatically when SQL Server Agent starts)。
- ◆ 当 CPU 空闲时进行备份(Start whenever the CPU(s) become idle)。
- ◆ 仅备份一次(One time),可以设置任意一个时刻。
- ◆ 定时备份(Recurring),可以选择周期性的数据库备份计划,时间段可以通过单击 "Change"按钮来设置。

选择好备份的类型之后,单击"OK"按钮,将返回到图 3-68 所示的主界面。

2. 在另一个 SQL Server 中恢复

首先使用合法的用户 ID 登录到另外一个 SQL Server 系统,然后把备份的数据库文件的一个拷贝复制到这个 SQL Server 所在的计算机上。启动 Enterprise Manager,展开其中的选项,选择"Databases",单击右键,在出现的选项对话框中选择"所有任务"选项中的"Restore database",将出现如图 3-71 所示的窗口。



图 3-71 在另一个 SQL Server 中恢复数据库



单击 "Select Devices" 按钮,选择要恢复的数据库文件,将弹出图 3-72 所示的对话框。

图 3-72 从磁盘进行恢复

单击"Add"按钮,弹出如图 3-73 所示对话框。

Ehoose Restore Desti	nation	K
		910.
THE SALE AND THE		
Q	d:\server2000\mssql7\BACKUP\	
		2-343 IS

图 3-73 选择要恢复备份文件

当备份文件选择完毕后,逐步确认后将返回到图 3-71 的界面。这时"Backup number" 选项将会变亮,在这里选择最近的一次备份(因为如果这个完整数据库备份文件在当初备 份时"overwrite"属性选的是"Append to media",那么它就很可能包含了从那时到现在的 每一次备份)。

然后单击"确定",数据库恢复过程开始执行。

3. 将数据库导出到另一个 SQL Server 系统

(1) 注册目标 SQL Server

要把数据库从一个 SQL Server 导出到另一个 SQL Server,必须先在本机的 SQL Server 中对目标 SQL Server 进行注册,才能将两个 SQL Server 相连接。并且,在目标 SQL Server 中应事先拥有一个登录 ID,并且此 ID 最好具有 "System administrator"(系统管理员)的 角色,这样可以保证能够自如地访问目标 SQL Server。

在"Enterprise Manager"左边的目录栏中选中"SQL Server Group",单击右键,在弹出的选项对话框中选择"New SQL Server Registration",将进入注册向导,该向导会给出网络中的 SQL Server 列表,选择目标 SQL Server 后,再填写正确的登录 ID 和密码,就可以将目标 SQL Server 注册在本机的"SQL Server Group"中了。

(2) 导出数据库

首先确定在目标 SQL Server 中已经新建了登录 ID, 同时还要确定目标 SQL Server 的 SQL Server 服务已经启动。然后在本机 SQL Server Enterprise Manager 中选择 "Databases" 中需要备份的数据库标志,单击右键,在出现的选项对话框选择 "所有任务"中的 "Export Data..."选项。这时将弹出 "Date Transfer Services Export Wizard" (数据转换服务输出向导) 的界面,单击 "下一步" 按钮,出现如图 3-74 所示的界面。



图 3-74 设置数据库的导出源

单击"下一步"按钮,如图 3-75 所示选择目标服务器并填写用户名和密码。在这里, 使用 SQL Server 验证模式来访问目标 SQL Server,填写登录 ID 和密码(在目标 SQL Server 中应事先拥有一个登录 ID,并且此 ID 最好具有"System administrator"的角色)。在 "Database"一项中选择"<new>",这时系统会要求你输入新的数据库名称,这里以 "pettbbs2"命名。



图 3-75 设置目标服务器

单击"下一步"按钮,进入选择导出方式的界面,如图 3-76 所示。

, DISExport W	izard		
Specify Tel Specify w the date	ble Copy or Query whether to copy one or source.	more tables or the :	results of a query free
	s Setting Set		
	9) Q(
		<u>≺ t</u> ¥0) T	-903 RM

图 3-76 导出方式的选择

设置好之后,单击"下一步"按钮,在下面的步骤中一般选择默认选项,就可以顺利 完成数据库的导出操作了。

第4章 ASP 程序准备

在开发 ASP 程序之前,需要做一些相应的准备工作。首先,所有的 ASP 程序都是在 服务器端执行的,程序开发者必须对服务器的操作系统以及 IIS 的设置相当了解。ASP 程 序需要一定的调试和运行环境,因此,需要对服务器端的 IIS 进行一定的配置。其次,为 调试开发的 ASP 程序,一个好的程序编辑软件是必不可少的。在本章中,向读者推荐微软 的产品 FrontPage 2000,并作简单的介绍。最后,对于一个 ASP 程序中需要嵌入大量的 HTML 语句,以便实现一定的功能和达到最佳的显示效果,因此了解一定的 HTML 语言的语法也 是必要的。

如果对这些内容已经很熟悉,可以跳过这一章,直接进行下一章的学习。

4.1 配置IIS

4.1.1 IIS 的概念

IIS 是 Internet Information Server (Internet 信息服务器)的缩写,它是微软公司主推的服务器,最新的版本是 Windows 2000 里面包含的 IIS 5。IIS 与 Window NT Server 完全集成 在一起,因而用户能够利用 Windows NT Server 和 NTFS (NT File System, NT 的文件系统) 内置的安全特性,建立强大、灵活而安全的 Internet 和 Intranet 站点。

IIS 支持 HTTP (Hypertext Transfer Protocol,超文本传输协议)、FTP (File Transfer Protocol,文件传输协议) 以及 SMTP 协议,通过使用 CGI 和 ISAPI, IIS 可以得到高度的 扩展。

IIS 支持与语言无关的脚本编写和组件,通过 IIS,开发人员就可以开发新一代动态的、 富有魅力的 Web 站点。IIS 不需要开发人员学习新的脚本语言或者编译应用程序, IIS 完全 支持 VBScript、JScript 开发软件以及 Java,它也支持 CGI 和 WinCGI,以及 ISAPI 扩展和 过滤器。

IIS 支持服务器应用的 Microsoft BackOffice 系列。Microsoft BackOffice 系列包括以下 内容:

- ◆ Microsoft Exchange Server 客户/服务器通讯和群组软件
- ◆ Microsoft Proxy Server 代理服务器
- ◆ 用于连接 IBM 企业网络的 Microsoft SNA Server
- ◆ 用于集中管理分布式系统的 Microsoft Systems Management Server
- Microsoft Commercial Internet System (MCIS)

IIS 的设计目的是建立一套集成的服务器服务,用于支持 HTTP、FTP 和 SMTP,它集成了现有产品能够提供快速且可扩展的 Internet 服务器。

IIS 响应性极高,同时系统资源的消耗也最少。IIS 的安装、管理和配置都相当简单, 这是因为 IIS 与 Windows NT Server 网络操作系统紧密地集成在一起。另外, IIS 还使用与 Windows NT Server 相同的 SAM (Security Accounts Manager,安全性账号管理器),对于管 理员来说, IIS 使用诸如 Performance Monitor 和 SNMP (Simple Nerwork Management Protocol,简单网络管理协议) 之类的 NT 现有管理工具。

IIS 支持 ISAPI,使用 ISAPI 可以扩展服务器功能,而使用 ISAPI 过滤器可以预先处理 和事后处理储存在 IIS 上的数据。用于 32 位 Windows 应用程序的 Internet 扩展可以把 FTP、 SMTP 和 HTTP 协议置于容易使用且任务集中的界面中,这些界面大大简化 Internet 应用程 序的使用。IIS 也支持 MIME (Multipurpose Internet Mail Extensions,多用途 Internet 邮件扩 展),它可以为 Internet 应用程序的访问提供一个简单的注册项。

IIS 的一个重要特性是支持 ASP。IIS 3.0 版本以后引入了 ASP,可以很容易地张贴动态 内容和开发基于 Web 的应用程序。对于诸如 VBScript 和 JScript 的开发软件,或者由 Visual Basic、Java 和 Visual C++开发的应用程序组件,以及现有的 CGI 和 Win CGI 脚本开发的应 用程序, IIS 都提供强大的本地支持。

4.1.2 IIS 的软件要求

运行 IIS 所需要的软件是带有最新升级服务包的 Windows NT Server 4,升级的 Windows NT4 Service Pack 可以在微软公司的网站(http://www.microsoft.com/iis)上获取。IIS 需要 Windows NT Server 支持,如果必要的话,还需要数据库系统(DBMS)和电子邮件系统。

Internet 服务器所需要的典型软件如表 4-1	1
----------------------------	---

	及中1 memer 服务船所需要的兴堂私厅
软件	版本
	带有最新升级服务包的 WindowsNTServer4+SP3 或者更新的版本
服务器软件	Microsoft Internet Information Server4
HTML 编辑软件	FrontPage、文本编辑器或者更新版本

表 4-1 Internet 服务器所需要的典型软件

通常连线的 Web 服务器会使用到许多数据库服务,实现这些功能都要求一个支持 ODBC 的数据库系统。

Web 服务器是 Web 数据库系统的心脏和灵魂。执行动态网页请求是 Web 服务器的核 心工作,没有 Web 服务器的支持,也就失去了后台数据库与客户之间的桥梁,客户无法获 得任何直接服务。也正是由于采用了服务器端的技术,使得 ASP 能创建可以为任何浏览器 执行的独立于客户平台的网页。作为 ASP 运行环境的 IIS(Internet 信息服务器)。在安装 Windows 2000 Server 时已经是默认的安装组件,不再需要进行任何安装工作,但是正确使 用和配置 IIS 对于创建自己的 ASP 项目将奠定一个良好的基础和开端。

最新的 IIS 版本是 IIS5.0, 它可以提供多种 Internet 信息服务, 但其中 WWW 服务为客

户端浏览器提交 Web 页,并允许访问 ASP 页面,是最重要的服务。从 Web 的发展角度来 看,WWW 服务也是所有服务中最重要的服务。

对于 IIS 中 WWW 服务的管理对整个站点或者对整个 Web 数据库系统的成功与否都具 有决定性意义。在 Windows 2000 Server 的视窗界面中,单击"开始",选择"程序"中的 "管理工具"程序组,其中有一项为"Internet 信息服务",运行它就可以对本服务器中的 IIS 进行管理和配置。IIS 的管理界面如图 4-1 所示。



图 4-1 IIS 管理界面

通过图 4-1 所示的管理界面,对 IIS 中的 WWW 服务进行管理非常简便。在图中所示的 "默认 Web 站点"上单击鼠标右键,将弹出如图 4-2 所示的选项框。



图 4-2 选项框

如图 4-2 所示,选择"新建">"虚拟目录"操作,将出现"虚拟目录创建向导",使 用该向导很容易新建一个虚拟目录。 实际上, 虚拟目录的建立和管理是 IIS 管理的一个很重要方面, 它也是网站开发和 Web 数据库系统开发中很重要的一个概念。使用虚拟目录有以下几个优点:

- ◆ 虚拟目录隐藏了有关目录结构的重要信息。在 Web 浏览中,用户通过选择 Directory/file,总能找到 Web 页的来源。如果在 Web 页中使用实际的硬盘文件路 径,将暴露有关站点目录结构的重要信息,可能导致某些人攻击系统,使系统安 全受到损害,但使用了虚拟目录后,就可以通过使用自定义的名称来代替实际的 目录名称,从而隐藏了具体的目录结构信息。
- ◆ 虚拟目录允许在不对页面代码作任何改动的情况下将 Web 页从一台服务器转移到 另一台服务器,只要两台服务器具有相同的虚拟目录即可。这样就大大方便了系 统的开发和移植。

选择"新建">"虚拟目录"操作之后,将首先弹出"虚拟目录创建向导"的欢迎画面, 只需要单击"下一步"就可以了。具体的画面这里不再给出。

下一步将出现如图 4-3 所示的选择虚拟目录别名画面,在这里可以选择一个和系统内 容相关的别名,或者任意的别名。这里笔者将为开发的 BBS 系统建立一个别名,因此输入 了"BBS"的别名。输入完毕后,单击"下一步"继续。



图 4-3 选择虚拟目录别名

出现 Web 站点内容目录界面,单击文本框右面的"浏览"按钮,可以选择本地计算机 中任何一个实际存在的目录,选择完毕后,单击"确定"按钮,路径名称将自动写入文本 框中,如图 4-4 所示。



图 4-4 选择实际的目录

下一步将出现访问权限的设置界面,如图 4-5 所示,可以看到其中的访问权限设置总 共有 5 项:

- ◆ 读取。这是最基本的访问权限,只有设置了这个权限,普通用户才可以访问虚拟 目录中的页面。
- ◆ 运行脚本。这一项是 ASP 程序开发的基本权限要求,只有具有这样的权限,存放 在虚拟目录中的 ASP 文件才能在客户浏览器发出请求之后运行相应的 ASP 文件, 否则将出现错误信息。
- ◆ 执行。这里主要是指执行一些服务器应用程序,如 CGI 程序等。
- ◆ 写入。是指用户是否有往此虚拟目录写入文件的权限。一般不设置这样的权限。
- ♦ 浏览。这里的浏览主要是指对目录结构的浏览权限。



图 4-5 访问权限设置

下一步将出现虚拟目录创建完毕的画面,只需要单击其中的"完成"按钮,就最终完成一个虚拟目录的设置。这时在默认 Web 站点的目录中将看到相应的虚拟子目录,如图 4-6 所示。



图 4-6 新建的虚拟目录

通过上面的步骤,在客户端浏览器访问本服务器时,c:\intpub\wwwroot\pettbbs 目录被 映射为/bbs 目录。在图 4-5 所示的访问权限设置中,可以看出此目录中的文件被允许执行, 但禁止读写。这时,在访问者浏览器的地址栏中显示的是 http://域名(或 IP 地址)/bbs/, 而不是 http://域名(或 IP 地址)/pettbbs/,这样就隐藏了实际的目录结构。从图 4-6 中可以 看到,实际存在的目录显示的是文件夹>>>> 标志,而虚拟目录是。一般的 HTML 页面只 需要请求读取就可以了,而 ASP 文件都需要被执行访问,所以在访问权限的设置中,需要 设置运行脚本一项。

在创建虚拟目录的过程中,会设置目录的访问权限。创建好的虚拟目录的访问权限也 可以通过改变属性来修改。

在 Internet 信息服务管理器中找到相应的虚拟子目录,单击右键,将弹出如图 4-7 所示的选项对话框。



图 4-7 对虚拟目录进行操作

医胚性			1 10
		a contraction of the second	
Marine Caracteria		esta latera	
	Q		
	Postal a secondaria (secondaria)		
		k et al.	<u>Real second</u>
	C:\Inetpub\wwwroot	\bbs	an a
fr <mark>a</mark> stered)			
ling select a little in the	BBS		
		See Press	
	纯脚本		
	中(共用的)		60) (
	and the second second		
			(d) - R R

从图中可以看到,对虚拟目录可以做很多操作,包括删除、重命名、刷新、编辑属性等。选择其中的"属性"选项,将弹出如图 4-8 所示的虚拟目录属性界面。

图 4-8 虚拟目录属性设置界面

从图中可以看到, 虚拟目录的属性主要包括5个方面:

(1) 虚拟目录

这是经常需要改变的属性界面之一,如图 4-8 所示,从中可以改变此虚拟目录所映射 的实际路径,对于访问权限也可以做新的设置。一般当硬盘上的实际文件路径改变时,只 需要在这个界面中改变虚拟目录的相应映射路径就可以了,而不需要重新建立一个新的虚 拟目录。

(2) 文档

文档界面是设置虚拟目录的默认文档界面,具体设置画面在图 4-8 中选择"文档"选项就可以看到。设置了默认文档之后,访问者在访问本虚拟目录时,只需要在地址栏中输入完整的目录名称就可以启动默认的文档,否则,在目录后需要输入对应的文件全名。

(3) 目录安全性

目录安全性对于任何一个 Web 站点来说都是一个很重要的问题。IIS 的安全系统和 Windows NT 的安全系统是紧密结合在一起的。但是这里应该设置为运行匿名访问,不然 Internet 上的其他用户是无法访问本 Web 站点的。

(4) HTTP 头

通过此选项界面可以设置本 Web 站点的 HTTP 头,这一选项一般不需要设置,采用默认值就可以了。

(5) 自定义错误信息

自定义错误信息可以在客户访问本站点出现错误时给出错误类型界面,一般也不需要 设置,采用默认值。

至此,对 IIS 的配置就算完成了。可以把编辑好的 ASP 文件放入相应的目录下,然后 在浏览器中运行,如果正常运行的话,说明 IIS 的配置没有问题。

ASP 程序的编辑工具很多,任何能够编辑文本的工具都可以用来编辑 ASP 文件。当然, 不同的编辑工具编写时的方便程度不同。在后面将介绍一个好的 ASP 文件编辑工具。当然, 最简单的就是使用 Windows 的记事本进行编辑。打开它,然后输入下面的这个程序:

<HTML>

<HEAD>

<TITLE>IIS 配置测试</TITLE>

</HEAD>

<BODY>

<%

```
Response.write "IIS 配置正确! "
```

%>

```
</BODY>
```

</HTML>

保存文件为 test.asp, 把文件存储到前面设置的 IIS 目录 c:\intput\wwwroot\pettbbs。然 后打开浏览器,在地址栏中调用该文件,调用的方式是: http://域名(或者 IP 地址或者主 机名)/bbs/test.asp。笔者在这里输入的是 http://wjy/bbs/test.asp, 其中 wjy 是主机名。如果 一切正常的话,浏览器上将显示出:

IIS 配置正确!

如果出现错误信息,则说明 IIS 的配置多少有一些问题。可能出现的错误情况有如下 4 种:

错误信息:无法与服务器建立连接(见图 4-9)

Microsoft Internet Explorer	
生活的 化和时间间的 经进行利益的 化合金	

图 4-9 无法与服务器建立连接

解 释:出现这样错误的原因是,所要访问的 Web 服务器已经关闭。 解决方法:启动如图 4-2 所示的默认 Web 站点选项,可以看到,如果 IIS 没有运行, 启动选项应当处于激活状态,点击后就可以启动服务进程。

错误信息:无法找到网页

HTTP 404 - 无法找到文件

- 解 释:浏览器无法找到调用的文件。
- 解决方法:这是因为在浏览器地址栏中输入了错误的地址或者错误的文件名。这样的 错误在调试 ASP 程序时经常会由于不小心而出现。解决办法很简单,检查 并输入正确的文件名和地址就可以了,当然前提是调用的文件实际存在。
- 错误信息: 网页无法显示 HTTP 403.1 禁止访问: 禁止可执行访问 您试图从目录中执行 CGI、ISAPI 或其他可执行程序,但该目录不允许执 行程序。
- 解 释:调用的文件在其路径下禁止执行。
- 解决方法: 这是由于没有正确设置 ASP 文件所在目录的访问权限。回头看一看前面叙述的内容,设置可执行的权限就可以了。
- 错误信息: 在浏览器中出现的是 asp 文件文本内容, 而不是执行的结果
- 解 释:由于 ASP 文件必须由 IIS 来执行,因此访问时只能通过在浏览器的地址栏 中输入 http://的方式来调用,这和普通的 HTML 文件是不同的。不能直接 使用打开文件命令。只需要在地址栏中输入正确的 URL 地址就可以了。这 一点对于 ASP 的初学者来说是应该注意的。

4.1.3 改变自定义错误信息

在 ASP 程序的调试过程中,经常会输错文件地址或文件名,这样就会出现如上所述的 HTTP404-无法找到该文件。浏览器的显示如图 4-10 所示。

A.C.B.MH40 Storovoll Internet (spinrer PS)	
4.8/new/document/advance/ad_gsuery_result.asp	
无法找到网页	
您正在被素的网页可能已经最除、更名或暂时不可用。	
谉豐試下 列腺作:	
 如果您在"纯丝"花中能入了阿瓦迪社,请检查其斯写是否正确。 打开 100.111.34.8 主页,得找指向所需信息的感觉。 单击后更安锐觉试其他能捷。 	
ЮТР 404 - 无法找到这种 Internet 信息服务	
• 详细信息: Bicrosoft 支持	
	n 29

图 4-10 HTTP 404---无法找到网页

这样的错误画面的可读性很差,而在程序的调试过程中又是不可避免的。而且作为一个正式的运行中的服务器,远程的客户端浏览器也有可能输入错误的文件地址,而得到如图所示的错误画面。为了增加可读性,只需要改变 IIS 的设置就可以了,也就是改变"默认 Web 站点"的属性。

如同上面的操作,打开 Internet 服务管理器,右键"默认 Web 站点",在弹出的选项菜单中选择属性,并且在属性界面中选择"自定义错误信息"标签,得到如图 4-11 所示的属性界面。



图 4-11 "自定义错误信息"设置界面

从图 4-11 中可以找到编号为 404 的 HTTP 错误,选中后,点击"编辑属性(E)"按钮, 将得到如图 4-12 所示的"错误映射属性"编辑界面。

错误映射起性
E:\Inetpub\wwwroot\error.htm
化算机算法 计全部的问题 计变换 网络马克
A second s
(1) 10 元 10 10 元 10 元 10 元 10 元 10 元 10 元

图 4-12 "错误映射属性"编辑界面

在图 4-12 所示的"错误映射属性"编辑界面中的"消息类型"下拉菜单中有三个选项可以选择,分别是:"默认值"、"文件"和"URL"。选择其中的"文件",然后点击"浏览"按钮,选择"文件(E)",然后点击"确定"按钮。

返回图 4-11 所示的界面后,点击"确定"按钮,自定义错误信息的属性设置完毕。这样当客户端向 IIS 服务器提交一个不存在的文件请求时,将显示自定义的 error.htm。

这里的 error.htm 是笔者编写的自定义错误文件,显示效果如图 4-13 所示。



图 4-13 自定义的错误文件

当然读者在这里可以编写自己的自定义错误文件,在其中写入任意信息。

注意: 自定义错误文件在编写时,如果需要在其中插入其他的文件,比如: 图 片文件,则必须使用绝对地址,否则将会出错,不能找到该文件。

4.2 ASP 开发工具介绍

在上一节中,为了测试 IIS 的配置用 Windows 的记事本编辑了一个简单程序。尽管用 记事本完全可以进行 ASP 程序的开发,但如果想高效地开发 Web 数据库管理系统,还应当 选择合适的工具。笔者在开发 ASP 程序的过程中采用的开发工具是 Microsoft FrontPage 20000,它的功能十分强大,尽管它的很多功能都体现在静态网页制作和站点管理上,但对 于 ASP 文件的开发也是十分便捷的。例如,在开发 Web 数据库系统时,用户交互表单的使 用是必需的,而使用 FrontPage 20000 可以快捷地生成表单,不需要开发者自己输入相应的 代码。在这里,只对它作简单的介绍。对于它的实际使用细节,就需要在实践中摸索了。 FrontPage 20000 专门的参考书也很多。

对于 ASP 程序开发者,使用最多的就是如图 4-14 所示的开发界面。笔者在开发过程 中之所以采用 FrontPage 20000 作为开发工具,主要是由于 FrontPage 2000 具有如下三个方 面的功能。

(1) 强大的编辑功能。

FrontPage 20000 在编辑网页方面功能强大,它以图形化形式编辑网页,通过"所见即 所得"方式使网页的编辑变得非常简单,特别是在制作复杂的大型网页时显得尤为方便。 它几乎可以实现 HTML 的所有功能,这是其他软件无法比拟的。



图 4-14 Microsoft FrontPage 20000 的使用界面

在 ASP 程序的编辑过程并没有很多网页编辑任务,但是使用 FrontPage 20000 可以很 容易找到一个 ASP 文件中代码所在的行数和列数。在调试 ASP 程序时,浏览器出现的错 误信息中一般都会指出是哪一行出现了错误,在 FrontPage 20000 中可以很容易找到指定的 行。另外,编辑菜单中的查找替换功能可以给开发者节省更多的重复工作,提高工作效率。 FrontPage 20000 中的查找功能不但可以找出当前文件中的指定代码,也可以找出当前站点 中所有文件中的代码,并以列表的形式列出。开发者在修改不同文件中相同代码时,使用 此项功能可以找出所有的相同代码,不至于遗忘哪一个文件(见图 4-15)。



图 4-15 FrontPage 20000 的查找功能

显示时,HTML 的代码是蓝色,而 VBScript(ASP 的默认脚本语言)代码为红色,很

容易区分。

(2) 强大的管理功能。

FrontPage 20000 管理 Web 站点的功能也非常强大,它不仅可以新建一个 Web 站点,还可以导入、复制、删除、发布一个 Web 站点。另外,FrontPage 20000 还具有对各种 Web 站点内部内容进行管理的功能,包括文件夹功能、报表管理功能、导航管理功能、任务管理功能。

(3) 强大的辅助功能。

FrontPage 20000 还具有很强大的辅助功能,它可以帮助开发人员实现一些比较难以实现的功能。比如,FrontPage 20000 可以很简单地在 Web 站点中建立一个搜索表单,而如果用人工建立一个搜索表单,其工作量非常大,也非常复杂。

另外,在 ASP 程序的开发过程中需要用到很多 HTML 元素,这些代码有时开发者并 不能全部记住,查找其他资料也非常麻烦,这时可以利用 FrontPage 20000 强大的网页编辑 功能,通过"所见即所得"方式插入相关的对象,然后查看自动生成的 HTML 源代码,就 可以从中找到需要的代码。

由于 FrontPage 20000 确实具有强大的编辑功能和网站管理功能,因而是 ASP 开发者 可选的开发工具之一。

4.3 HTML 语法概要

ASP 语句和 HTML 语句可以相互嵌套,达到综合的效果。在使用 ASP 开发 Web 数据 库系统时,HTML 代码是必要的。关于 HTML 语言的语法很多,在这里不作全面介绍,读 者可以参考相关的书籍。但是,有一些 HTML 语句是 Web 数据库系统中最常用的,主要是 表单 Form 和表格 Table。在这一节中,简要介绍 Form 和 Table 语句的使用方法。

4.3.1 表单 Form 的使用

HTML 的 Form 是主页间传递信息的最主要手段,也是用户和 ASP 充分实现交互和信息传递的最可靠手段。因为标准 HTML 语言适合任何类型的浏览器,所以利用 Form 来传递用户信息是可靠的。当然也可以选择其他方法,例如 Java Applet、ActiveX 控件等等。如果希望所创建的网站是针对 Internet 上更多层次上的使用者,建议还是以 HTML 的 Form 为主,这样可以使网站拥有更多的浏览者。

一般地,凡需要用户输入一定的信息或者选择时,都通过使用表单来完成。例如第 2 章介绍的 BBS 论坛系统,在用户登录、新用户注册以及发表文章、书写信件等需要用户输 入信息的地方,都使用了表单。

下面通过实例详细说明 Form 的基本语法结构。

第2章中图 2-1 显示的是 BBS 论坛的登录界面,其 HTML 源代码如下:

<html>

<head>

<title>登录界面</title> </head> <body bgcolor="#ffffff0"> <form method="POST" action="choice.asp" name="input"> 欢迎光临 pett-BBS 论坛 <div align="center"> 用户名 <input name="lybid" style="font-size: 9pt;border: 1 solid #800080"size="20">

密码

<input name="password" style="font-size: 9pt;border: 1 solid #800080" size="20"
type="password">

<input type="submit" value="登录" name="B3" style="background-color:
#800080; color: #FFFF00; font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-style:
ridge; border-color: #FFFF00">

<input type="reset" value="重写" name="B4" style="background-color: #800080; color: #FFFF00; font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-style: ridge; border-color: #FFFF00">

</center>

</form>

</body>

</html>

注意: login.htm 的源代码中还有一个输入判断程序在这里略去了,见 7.1 节 "登 录系统"和 10.8 节 "表单限制"。

Login.htm 的源代码实际上是 Form 和 Table 语句的嵌套,使用 Table 语句使表单中的各个文本框的显示更加有序。从上面的源代码可以看到 Form 的基本语法结构。

```
<form action="url" method=*>
```

<input type=submit> <input type=reset>

</form>

其中,*=GET 或者 POST 表单中提供给用户的输入形式为:

<input type=* name=**>

*=text, password, checkbox, radio, image, hidden, submit, reset

其中各个参数的含义见表 4-2。

参数	含义	
Text	文字 输 入	
Password	密码输入	
Checkbox	复选框	
Radio	单选按钮	
Image	图片	
Hidden	隐藏	
Submit	提交	
Reset	重置	

表 4-2 表单类型

下面举例说明每一种类型的使用方法。

1. 文字输入(Text)和密码输入(Password))

**=表单域的名称

<input type=*>

<input type=* value=**>

<form action=choice.asp method=POST>

您的姓名:
```
<input type=text name=网址><br>
密码:
<input type=password name=密码><br>
<input type=submit value="发送"><input type=reset value="重设">
</form>
```

2. 复选框(Checkbox)和单选按钮(RadioButton)

```
<form action=post.asp method=POST>
<input type=checkbox name=水果 1 value=香蕉>
        香蕉
<input type=checkbox name=水果 2 value=苹果 checked>
        苹果
<input type=checkbox name=水果 3 value=橘子>
       橘子
<input type=submit><input type=reset>
</form>
<form action=post.asp method=POST>
<input type=radio name=水果>
       Banana
<input type=radio name=水果 checked>
        Apple
<input type=radio name=水果 value=橘子>
       Orange
<input type=submit><input type=reset>
</form>
```

3. 隐藏表单的元素

<form action=post.asp method=POST> <input type=hidden name=add value=next> 这是一个隐藏元素。 <input type=submit><input type=reset> </form>

4. 下拉菜单(Selectable Menu)

<form action=post.asp method=POST> <select name=fruits>

<option>Banana

<option selected>Apple

```
<option value=My_Favorite>Orange
</select>
<input type=submit><input type=reset>
</form>
<select size=**>
<form action=post.asp method=POST>
<select name=fruits size=3>
         <option>Banana
         <option selected>Apple
         <option value=My_Favorite>Orange
         <option>Peach
</select>
<input type=submit><input type=reset>
</form>
          <select size=** multiple>
<form action=post.asp method=POST>
<select name=fruits size=3 multiple>
         <option selected>Banana
         <option selected>Apple
         <option value=My Favorite>Orange
         <option selected>Peach
</select>
<input type=submit><input type=reset>
</form>
```

注意:用 Ctrl 键配合鼠标实现多选(和 MS-WINDOWS 的 File Manager 一样)。

5. 文本区域

<textarea name=* rows=** cols=**> ... <textarea> <form action=post.asp method=POST> <textarea name=comment rows=5 cols=60> </textarea> <P> <input type=submit><input type=reset> </form>

4.3.2 表格 Table 的使用

Table 语句在 ASP 程序中虽然不是必需的,但使用表格可以合理显示页面中的各个对

象,可以按照一定的要求排列页面的元素,而且如果把表格的边框设为 0 的话,在浏览器 中是看不见表格存在的。另外,在 Web 数据库系统中查询的结果需要以列表形式显示,使 用 Table 语句无疑是最佳的选择。

在 4.3.1 节中所举的 login.htm 实例源代码中就使用了 Table 语句。 Table 语句的基本语法结构如下:

<tr>

其中:

定义表格

定义行

之/td>定义列

注意: Table 语句中,列是在行中定义的,也就是说,

包含在七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>七/tr>

除这些基本语法结构外,Table 语句还有很多属性参数,比如背景颜色、对齐方式、边 框颜色、边框尺寸、单元格间距等,ASP 编程者必须熟悉这些参数值。可能有些读者对这 些参数值还不是很了解,可以参考相应的资料和书籍。另外,可以使用前面提到的 FrontPage 等可见即可得的编辑器,通过鼠标操作设置表格的属性,而不需要了解具体的语法。

4.3.3 超链接的目标属性 target

熟悉 HTML 语句的读者对超链接一定很清楚,实际上,HTML 语言的最主要功能就是 通过不同页面之间的超链接来完成的。超链接的语法这里不再叙述,这里重点介绍超链接 的目标属性 target 参数的设置方法,大致有以下几种参数设置:

 功能:新窗口
 功能:本窗口
 功能:父窗口
 功能:整个浏览器窗口
 功能:新的窗口
 功能:frame 框架中的相应窗口

上面介绍的表单、表格和超链接目标属性是 ASP 程序中必不可少的 HTML 代码,掌握这些语法对 ASP 编程是很有用的。

第5章 VBScript 编程基础

VBScript 是 Active Sever Pages 的缺省脚本语言。与一些复杂的编程语言相比,VBScript 只是一种脚本语言,语法要相对简单得多。但其在语法、变量使用、循环语句等方面还是 有一定的要求,掌握 VBScript 的基本语法,可以为 Active Server Pages 编程奠定良好的基础。另一方面,如果不熟悉 VBScript 语句,在调试程序过程中往往会出现不容易检查出来的语法错误,会大大降低工作效率。因此在调试 ASP 程序之前,必须对 VBScript 的基本语 法有一定的了解和认识。

本章将主要介绍 VBScript 的语法,为 ASP 编程奠定一定的基础。将依次讨论以下内容:

- ◆ 变量类型和使用
- ◆ 运算符
- ◆ 数组
- ◆ 控制流
- ◆ 函数和子进程
- ♦ 注释

5.1 VBScript 介绍

VBScript 与 BASIC 语言有密切关系。如果熟悉 BASIC,将会发现 VBScript 很容易学 习和使用。VBScript 是 Microsoft Visual Basic 的简化版本,但 VBScript 和 Visual Basic 之间 有一些重要差异:

- ◆ 变量类型和作用范围不同
- ◆ 不能使用某些内置函数
- ◆ 增加了一些函数和对象
- ◆ 处理错误的方式不同
- ◆ 进行排错调试的方式不同

VBScript 是一种脚本语言,意味着这种语言比 BASIC 语言的成熟版本容易使用又难于 使用。编制简单的程序时,脚本语言是容易使用的。脚本语言的句法比较简单。但是,简 单的句法也使开发复杂的应用程序变得很困难。

可以在 HTML 文件中直接嵌入 VBScript 脚本,这样能扩展 HTML 的功能,使它可以 对用户的操作作出反应、进行交互,不再仅仅是一种页面格式语言。VBScript 是 Active Server Pages 的缺省语言。当然也可以在 ASP 网页中使用其他脚本语言,如 JavaScript、Perl 或者 REXX(由 IBM UK 实验室的 Michael Cowlishaw 所创建的一种语言)。本书所有的 ASP 例子 程序都使用 VBScript。

VBScript 既可以作为客户端编程语言,也可以作为服务器端编程语言。客户端编程语 言可以由浏览器解释执行。Java 和 JavaScript 也可以作为客户端编程语言。当以这些语言 中任意一种编制的一个程序下载到一个兼容的浏览器中时,浏览器将自动执行该程序。

客户端编程语言的优点是浏览器完成了所有的工作,这可以减轻服务器的负担。而且 客户端程序运行起来比服务器端程序快得多。当一个浏览器的用户执行了一个操作时,不 必通过网络对其作出响应,客户端程序就可以作出响应。

一般来说,可以解释 VBScript 脚本的浏览器只有 Microsoft Internet Explorer,例如 Netscape Navigator 将忽略 VBScript 脚本,这意味着目前没有充分理由把 VBScript 作为客 户端编程语言。也不允许花费时间编写出只有少数浏览器可以解释的程序。

注意: Internet Explorer 和 Netscape Navigator 都可以解释 JavaScript 程序,因此一般客户端编程语言都采用 JavaScript,而服务器端的编程语言都采用 VBScript。

VBScript 也可以作为服务器端编程语言。服务器端编程语言是在服务器上执行的。服务器为一个站点提供文件,而浏览器接收这些文件。服务器端编程语言执行站点主机上的所有操作。

用 VBScript 作为服务器端编程语言的好处是 VBScript 不受浏览器的限制。VBScript 脚本在网页通过网络传送给浏览器之前被执行。Web 浏览器收到的只是标准的HTML文件。

当创建一个 ASP 网页时,应该用 VBScript 作为服务器端编程语言。

5.2 VBScript 中的变量

和所有的编程语言相同,在使用 VBScript 进行编程时,也需要一些变量来传递数值, 作为循环语句中的判断等等。VBScript 中只有一种基本数据类型,即 variant,因此所有的 变量数据类型都是 variant。variant 是一种特殊的变量类型,它可以代表许多其他类型的变量。

5.2.1 声明变量

可以使用 Dim 语句、Public 语句和 Private 语句在 Script 中显式声明变量。例如:

Dim a

一次声明多个变量时,不同的变量名之间用逗号隔开。例如:

Dim a,b,c,d

在 VBScript 程序中,使用一个变量之前不必进行声明。例如:

<HTML>

<HEAD>

<TITLE> Simple VBScript</TITLE>

</HEAD>

<BODY>

<%hello=" Hello World! " %>

<%=hello%>

</BODY>

</HTML>

这是一个十分简单的 ASP 网页,在第 4 章中已经介绍过调用 ASP 网页的一些注意事项,这里不再赘述。

在这个例子中,变量 hello 并没有经过声明,但是它被赋值"Hello World!",这个值将 在浏览器中输出。这个 ASP 网页可以工作得很好。

然而,如果编写长而复杂的程序时,应该先统一声明变量然后再使用。这样做的好处 是使程序脚本易于调试。如果产生输入错误,程序在执行过程中有可能产生其他结果。例 如:

<HTML>

```
<HEAD></HEAD>
```

Dim hello

```
Hello="Hello World! "
```

=Heloo

%>

```
</HTML>
```

这段程序本来的功能是输出 Hello 变量的值,但由于程序编写者的输入错误,把 Hello 误写为 Heloo,程序仍然能正确执行,只是输入的是 Heloo 变量的值,但 Hello 变量没有赋值,因此浏览器中将不会得到任何输出结果。这样的隐形错误有时在程序调试中很难查找。

为了避免这样的错误,可以使用<%OPTION EXPLICIT%>语句,例如:

```
<%OPTION EXPLICIT%>
<HTML>
<HEAD></HEAD>
<%
Dim hello
Hello="Hello World! "
=Heloo
%>
</HTML>
```

在这个例子中,OPTION EXPLICIT 语句强制所有的变量必须专门声明。Dim 语句声明了变量 Hello。但后面的 Heloo 变量并不是声明过的变量,因此程序在执行过程中将出现错误信息: Variable is undefined: "Heloo"

注意: OPTION EXPLICIT 语句的位置很特殊,必须把 OPTION EXPLICIT 语句作 为一个 ASP 网页的起始语句之一。它必须出现在任何 HTML 标识或其他 VBScript 命令之前。否则,该语句将被视为非法语句。

5.2.2 变量命名规则

在 VBScript 中声明变量时,必须遵循其中的标准命名规则:

- ◆ 变量名的第一个字符必须是字母;
- ◆ 变量名中不能包含嵌入的句点;
- ◆ 变量名的长度不能超过 255 个字符;
- ◆ 变量在被声明的变量域内必须惟一;
- ◆ 变量的作用域和存活期由声明的位置决定。

在 VBScript 程序中,可能会有自定义的函数或者子过程,在函数或者子过程中一般都 有变量定义,这些变量称为过程级变量,当函数或者子过程的运行结束时,这些变量的生 命期也就结束了。在函数或者子过程之外声明的变量成为脚本级或者模块级,在整个程序 的运行过程中变量都有效存在。也就是说,所有脚本级变量或者模块级变量都可以在声明 该变量的 WEB 页面的所有过程中使用,它的生命周期从浏览器分析该变量开始,到浏览 器从该页面上移走结束。

5.3 数据类型

DIM 语句声明一个变量时,并没有定义变量的数据类型,这是因为 VBScript 中所有变量都是 variant 型变量。

当需把数字赋值给一个变量时,可以直接把该数字分配给该变量,不用引号标记。当 需要把字符串赋值给一个变量时,直接把字符串分配给该变量,用引号标记。一个 variant 型变量可以自动与赋给它的值的类型保持一致。

注意:如果把数字赋值给变量时使用引号标记,则将把这个数字当成字符串来处理。

通常,不必考虑变量如何自动转变为具体的子类型。有一个函数可以输出变量的子类型。例如:

<%Hello="Hello World! "%> <%=TYPENAME(Hello)%> <%=vartype(Hello)%>

<%Hello=3%>

<%=TYPENAME(Hello)%>

<%=vartype(Hello)%>

这一段程序的输出如下:

String8Integer2

在这个例子中,变量 Hello 被分配了两个不同数据类型的值。对每种类型的值,VBScript 函数 TYPENAME()和 VARTYPE()都可以显示该变量所代表的实际数据的子类型。例如, 当给变量 Hello 分配字符串数据 "Hello World!"时,variant 型变量的子类型是 string,相应 的 Vartype()函数的输出是 8。当给变量 Hello 分配整数 3 时, Hello 的子类型是 integer,相 应的 Vartype()函数的输出是 2。

这个例子说明 variant 型变量实际代表的数据是有子类型的。variant 包含的数据子类型 见表 5-1。函数 TYPENAME()和 VARTYPE()都可以显示一个变量的子类型。不同的是, TYPENAME()显示的是具体的子类型,而 VARTYPE 显示的是子类型的数字代码,对应的 子类型见表 5-2。

不同的变量子类型可以代表的数据有不同的范围。表 5-1 列出了变量子类型和它们的 数据范围的完整信息。

数据子类型	描述
Empty	未初始化的 variant,对于数值变量值为 0,对于字符串变量值为""
Null	不包含任何数据的 variant
Boolean	TRUE 或 FALSE
Byte	0 到 255 之间的整数
Error	包含错误号
Integer	-32 768 到 32 767 之间的整数
Long	-2 147 483 648 到 2 147 483 647 之间的整数
Single	单精度浮点数
	负数:-3.402823E38 到-1.401298E-45
	正数: 1.401298E-45 到 3.402823E38
Double	双精度浮点数
	负数:-1.79769313486232E308 到-4.94065645841247E-324
	正数: 4.94065645841247E-324 到 1.79769313486232E308
Currency	-922 337 203 685 477.5808 到 922 337 203 685 477.5807
Date	包含日期的数字,从 100 年 1 月 1 日到 9999 年 12 月 31 日
object	任何对象
string	变长字符串,最大长度为20亿个字符

表 5-1 数据子类型

注意: string 子类型的数据范围非常之大,你可以把整个网页甚至整部小说保存 在一个字符串变量中。有时把网页保存在字符串中是很有用的。

同样,使用 Vartype 函数也可以得到变量的数据子类型,Vartype 函数的返回值所代表的变量子类型如表 5-2 所列。

Vartype()返回值	对应的变量子类型
0	Empty
1	Null
2	Integer
3	Long
4	Single
5	Double
6	Currency
7	Date
8	String
9	Object
10	Error
11	Boolean
12	Variant(指示变量数组)
13	非 ActiveX(OLE)自治控件
17	Byte
8192	数组

表 5-2 Vartype 函数的返回值对照

在所有数据子类型中,需要注意的是空变量和空值变量,这是两种不同的变量子类型。 在给一个变量赋值之前,该变量是空的。用函数 ISEMPTY()可以检测一个变量是否为空。 如下例所示:

<% a="Hello World! " %> a:<%=ISEMPTY(a)%> b:<%=ISEMPTY(b)%>

在这个例子中,检测了两个变量 a 和 b,看它们是否为空。因为 a 已经被分配了一个 值,当把变量 a 作为函数 ISEMPTY()的参数时,该函数返回 FALSE。而 b 没有分配数据, 因此以 b 为参数时,函数 ISEMPTY()返回 TRUE。

空变量与被分配了长度为0的字符串或数字0的变量是不同的。例如,语句 a=" "和 a=0 都给变量 a 分配了一个值。在这两种情况下,该变量都不是空的。只有一个从未被赋过值

的变量才是空变量。

空变量与包含空值的变量也是不同的。这也许有些奇怪,一个被分配了空值 NULL 的 变量不是空变量,而是有值 NULL。如果想使一个已经分配了值的变量成为空变量,必须 使用关键字 EMPTY。例如:

a=EMPTY

5.4 VBScript 常数

VBScript 可以声明常量,用于代替数字或者字符串。常量与变量有相似之处;但是, 一旦声明了一个常量,该常量的值将不能被改变。例如:

<%

CONST Hello="世界你好! "

CONST Bdate=#21-8-00#

%>

在这个例子中,把"世界你好!"这个字符串赋值给常量 Hello。因为 Hello 是一个常量,在脚本中将不能给 Hello 分配新值。如果试图改变一个常量的值,将收到错误信息:Illegal assignment error。当然,也可以使用逗号来同时定义多个常量。

注意:字符串文字应包含在双引号""之间,这也是区别数值型常数和字符串型常数最明显的方法。日期文字和时间文字必须包含在两个#号之间。

5.5 VBScript 运算符

VBScript 有一套完整的运算符,包括算术运算符、比较运算符、连接运算符和逻辑运算符。

5.5.1 运算符优先级

当表达式包含多个运算符时,将按预定顺序计算每一部分,这个顺序被称为运算符 优先级。可以使用括号越过这种优先级顺序,强制首先计算表达式的某些部分。运算时, 总是先执行括号中的运算符,然后再执行括号外的运算符。但是,在括号中仍遵循标准运 算符优先级。

当表达式包含多种运算符时,首先计算算术运算符,然后计算比较运算符,最后计算 逻辑运算符。所有比较运算符的优先级相同,即按照从左到右的顺序计算比较运算符。算 术运算符和逻辑运算符的优先级如表 5-3 所示,由上至下优先级降低。

算术这	算符	比较运算符	3	逻辑运	算符
描述	符号	描述	符号	描述	符号
求幂	^	等于	=	逻辑非	Not
负号	-	不等于	\diamond	逻辑与	And
乘	*	小于	<	逻辑或	Or
除	/	大于	>	逻辑异或	Xor
整除	١	小于等于	<=	逻辑等价	Eqv
求余	Mod	大于等于,	>=	逻辑隐含	Imp
加	+	对象引用比较	Is		
减	-				
字符串连接	&				

表 5-3 VBScript 运算符

当乘号与除号同时出现在一个表达式中时,按从左到右的顺序计算乘、除运算符。同 样,当加与减同时出现在一个表达式中时,按从左到右的顺序计算加、减运算符。

字符串连接运算符&不是算术运算符,但是在优先级顺序中它排在所有算术运算符之 后和所有比较运算符之前。Is运算符是对象引用比较运算符,它并不比较对象或对象的值, 而只是进行检查,判断两个对象引用是否引用同一个对象。

连接运算符&用于强制两个表达式进行字符串连接,用法是: result=expression1& expression2。当任一 expression 不是字符串时,它将被转换为 String 子类型。如果两个表达式都为Null, result 也为 Null。然而,如果仅有一个 expression 为 Null,则它和其他表达式连接时按零长度字符串("")处理。任何表达式为Empty时也按零长度字符串处理。

5.6 循环语句

VBScript 中的循环语句是最主要的编程方式,对象中各个参数的次序显示都是通过循 环语句来完成的,另一方面,多数计数也都使用循环语句来完成。VBScript 中循环语句的 语法非常简单,和一般编程语言中的循环语句相近。

循环用于重复执行一组语句。循环可分为 3 类: 一类在条件变为 False 之前重复执行语句, 一类在条件变为 True 之前重复执行语句, 另一类按照指定的次数重复执行语句。

在 VBScript 中可使用下列循环语句:

- ♦ Do...Loop: 当(或直到)条件为 True 时循环。
- ◆ While...Wend: 当条件为 True 时循环。
- ◆ For...Next: 指定循环次数,使用计数器重复运行语句。
- ◆ For Each...Next: 对于集合中的每项或数组中的每个元素, 重复执行一组语句。

5.6.1 Do...Loop 循环

当条件为 True 时或条件变为 True 之前重复执行某语句块。Do...Loop 语句的语法结构 如下:

Do [{While | Until} condition] [statements] [Exit Do] [statements]

Loop

也可以使用下面的语法结构:

Do

[statements] [Exit Do] [statements]

Loop [{While | Until} condition]

Do...Loop 语句的语法有以下部分:

- ◆ condition 数值或字符串表达式,其值为 True 或 False。如果 condition 为Null, 则 condition 被当作 False。
- ◆ statements 当 condition 为 True 时被重复执行的一条或多条命令。
- ◆ Exit Do 仅可以用在 Do...Loop 控制语句中,提供另一种退出 Do...Loop 的方法。 可以在 Do...Loop 语句中的任何位置放置任意个 Exit Do。Exit Do 通 常与条件判断语句(如 If...Then)一起使用,将控制传递给紧随在 Loop 语句后面的语句。

当用于嵌套 Do...Loop 中的时候, Exit Do 将控制传递给其所在循环的上一层嵌套循环。

5.6.2 While...Wend 语句

当指定的条件为 True 时,执行一系列语句。While...Wend 语句的语法结构如下:

While condition

[statements]

Wend

While...Wend 语句的语法有以下部分:

- ◆ condition 数值或字符串表达式,其计算结果为 True 或 False。如果 condition 为 Null,则 condition 被当作 False。
- ◆ statements 在条件为 True 时执行的一条或多条语句。

如果 condition 为 True,则 statements 中所有 Wend 语句之前的语句都将被执行,然后 控制权将返回到 While 语句,并且重新检查 condition。如果 condition 仍为 True,则重复执行上面的过程。如果不为 True,则从 Wend 语句之后的语句继续执行程序。

While...Wend 循环可以是多层嵌套结构。每个 Wend 与最近的 While 语句对应。

提示: Do...Loop 语句提供一种结构化与适应性更强的方法来执行循环。

5.6.3 For...Next 语句

以指定次数重复执行一组语句。

```
For counter = start To end [Step]
[statements]
[Exit For]
[statements]
Next
```

For...Next 语句的语法包含以下部分:

counter	用做循环计数器的数值变量。这个变量不能是数组元素或用户自定义类
	型的元素。
start	counter的初值。
end	counter 的终值。
Step	counter 的步长。如果没有指定,则 Step 的默认值为 1。
statements	For 和 Next 之间的一条或多条语句,将被执行指定次数。

Step 参数可以是正数或负数。step 参数值决定循环的执行情况,如表 5-4 所示:

值	如果…则循环执行
正数或0	counter <= end
负数	counter >= end

表 5-4 step 参数值

当循环启动并且所有循环中的语句都执行后, step 值被加到 counter 中。这时, 或者 循环中的语句再次执行(基于循环开始执行时同样的测试), 或者退出循环并从 Next 语句 之后的语句继续执行。

提示: 在循环中改变 counter 的值,将会使程序代码的阅读和调试变得更加困难。

Exit For 只能用于 For Each...Next 或 For...Next 结构中,提供另一种退出循环的方法。 可在语句的任意位置放置任意个 Exit For 语句。Exit For 经常和条件判断语句一起使用(例 如 If...Then),并立即将控制权转移到 Next 之后的语句。

可以将一个 For...Next 循环放置在另一个 For...Next 循环中,组成嵌套循环。每个循环

中的 counter 要使用不同的变量名。下面的结构是正确的:

```
For I = 1 To 10
For J = 1 To 10
For K = 1 To 10
...
Next
Next
```

Next

5.6.4 For Each...Next 语句

对数组或集合中的每个元素重复执行一组语句。For Each...Next 语句的语法结构如下:

```
For Each element In group
```

[statements]

[Exit For]

[statements]

Next [element]

For Each...Next 语句的语法包含以下部分:

element	用来枚举集合或数组中所有元素变量。对于集合, element 可能是 variant
	变量、通用 Object 变量或任意指定的Automation 对象变量。对于数组,
	element 只能是 Variant 变量。
group	对象集合或数组的名称。
statements	对于 group 中的每一项执行的一条或多条语句。
	如果 group 中有至少一个元素,就会进入 For Each 块执行。一旦进入循
	环,便首先对 group 中第一个元素执行循环中的所有语句。只要 group
	中还有其他的元素,就会对每个元素执行循环中的语句。当 group 中没
	有其他元素时退出循环,然后从 Next 语句之后的语句继续执行。
Exit For	只能用于 For EachNext 或 ForNext 控制结构中,提供另一种退出循环
	的方法。可在循环的任意位置放置任意个 Exit For 语句。Exit For 经常和
	条件判断语句一起使用(例如 IfThen),并立即将控制权转移到 Next
	之后的语句。

可以将一个 For Each...Next 循环放置在另一个 For Each...Next 循环之中,组成嵌套式 For Each...Next 循环。但是每个循环的 element 必须是惟一的。

注意:如果省略 Next 语句中的 element,则程序仍会像已包含它一样继续执行。如果 Next 语句在其相应的 For 语句之前出现,则会产生错误。

5.7 VBScript 条件语句

使用条件语句和循环语句可以控制 Script 的流程。使用条件语句可以编写进行判断和 重复操作的 VBScript 代码。在 VBScript 中可使用以下条件语句:

◆ If...Then...Else 语句

◆ Select Case 语句

5.7.1 If...Then...Else 语句

根据表达式的值有条件地执行一组语句。具体语法结构如下:

If condition Then statements [Else elsestatements]

或者,使用块形式的语法:

If condition Then

[statements]

[ElseIf condition-n Then [elseifstatements]]...

[eisenstatements]] . .

[Else

[elsestatements]]

End If

If...Then...Else 语句的语法包含以下部分:

condition	一个或多个下面两种类型的表达式:
	数值或字符串表达式,其运算结果是 True 或 False。如果 condition
	是Null,则 condition 被视为 False。
	形如 TypeOf objectname Is objecttype 的表达式。objectname 是任何
	对象的引用,而 objecttype 则是任何有效的对象类型。如果
	objectname 是 objecttype 所指定的一种对象类型,则表达式为 True;
	否则为 False。
statements	condition 为 True 时执行的一条或多条(以冒号分开)语句。
condition-n	同 condition。
elseifstatements	相关的 condition-n 为 Ture 时执行的一条或多条语句。
elsestatements	如果前面没有 condition 或 condition-n 表达式为 False 时,执行的一
	条或多条语句。

对于短小简单的测试,可以使用单行形式(第1种语法)。但块形式(第2种语法)提供了比单行形式更强的结构化与适应性,比较容易阅读、维护及调试。

注意: 在单行语法中, 可以执行多条语句作为 If...Then 判断的结果, 但所有语句 必须在同一行上并且以冒号分开, 如下列语句所示:

If A > 10 Then A = A + 1 : B = B + A : C = C + B

当程序运行到 If 块(第2种语法)时,将测试 condition。如果 condition 是 True,则执行 Then 之后的语句。如 condition 是 False,则每个 ElseIf 部分的条件式(如果有的话)会依次计算并加以测试。当找到某个为 True 的条件时,则其相关的 Then 之后的语句会被执行。如果没有一个 ElseIf 语句是 True(或没有 ElseIf 子句),则将执行 Else 之后的语句。执行 Then 或 Else 之后的语句以后,将继续执行 End If 之后的语句。

Else 和 Elself 子句都是可选的。在 If 块中可以放置任意多个 Elself 子句,但是都必须 在 Else 子句之前。If 块语句可以被嵌套,即被包含在另一个 If 块语句之中。

要判断某个语句是否是一个 If 块,可检查 Then 关键字之后是什么。如果在 Then 的同 一行之后还有其他非注释的内容,则此语句就是单行形式的 If 语句。

If 块语句必须是某一行的第一条语句,并且必须以 End If 语句结束。

5.7.2 Select Case 语句

根据表达式的值执行几组语句之一。其语法结构如下:

```
Select Case testexpression
[Case expressionlist-n
[statements-n]] . . .
[Case Else expressionlist-n
[else statements-n]]
```

End Select

Select Case 语句的语法包含以下部分:

testexpression	任意数值或字符串表达式。
expressionlist -n	如 Case 出现则必选。它是一个或多个表达式的分界列表。
statements-n	当 testexpression 与 expressionlist-n 中的任意部分匹配时,执行的一
	条或多条语句。
else statements -n	当 testexpression 与 Case 子句的任何部分不匹配时,执行的一条或
	多条语句。

如果 testexpression 与任何 Case expressionlist 表达式匹配,则执行此 Case 子句和下一个 Case 子句之间的语句,对于最后的子句,则会执行该子句到 End Select 之间的语句,然 后控制权会转到 End Select 之后的语句。如 testexpression 与多个 Case 子句中的 expressionlist 表达式匹配,则只有第一个匹配后的语句被执行。

Case Else 用于指示若在 testexpression 和任何其他 Case 选项的 expressionlist 之间未找 到匹配,则执行 elsestatements。虽然不是必要的,但最好是将 Case Else 语句置于 Select Case

块中,以处理不可预见的 testexpression 值。如果没有 Case expressionlist 与 testexpression 匹配且无 Case Else 语句,则继续执行 End Select 之后的语句。

Select Case 语句可以是嵌套的,每一层嵌套的 Select Case 语句必须有与之匹配的 End Select 语句。

5.8 在 VBScript 中使用对象

在 VBScript 中,过程被分为两类: Sub过程和Function过程。

5.8.1 Sub 过程

Sub 过程是包含在 Sub 和 End Sub 语句之间的一组 VBScript 语句,执行操作但不返回 值。Sub 过程可以使用参数(由调用过程传递的常数、变量或表达式)。如果 Sub 过程无任 何参数,则 Sub 语句必须包含空括号()。

下面的 Sub 过程使用两个固有的(或内置的) VBScript 函数,即MsgBox和InputBox, 来提示用户输入信息。然后显示根据这些信息计算的结果。计算由使用 VBScript 创建的 Function 过程完成。此过程在以下讨论之后演示。

```
Sub ConvertTemp()
temp = InputBox("请输入华氏温度。", 1)
MsgBox "温度为 " & Celsius(temp) & " 摄氏度。"
```

End Sub

5.8.2 Function 过程

Function 过程是包含在 Function 和 End Function 语句之间的一组 VBScript 语句。 Function 过程与 Sub 过程类似,但是 Function 过程可以返回值。Function 过程可以使用参数(由调用过程传递的常数、变量或表达式)。如果 Function 过程无任何参数,则 Function 语句必须包含空括号()。Function 过程通过函数名返回一个值,这个值是在过程的语句中赋 给函数名的。Function 返回值的数据类型总是 Variant。

在下面的示例中, Celsius 函数将华氏度换算为摄氏度。Sub 过程 ConvertTemp 调用此 函数时,包含参数值的变量被传递给函数,换算结果返回到调用过程并显示在消息框中。

```
Sub ConvertTemp()

temp = InputBox("请输入华氏温度。", 1)

MsgBox "温度为 " & Celsius(temp) & " 摄氏度。"

End Sub

Function Celsius(fDegrees)

Celsius = (fDegrees - 32) * 5 / 9
```

End Function

5.8.3 过程的数据进出

给过程传递数据的途径是使用参数。参数被作为要传递给过程的数据占位符。参数名可以是任何有效的变量名。使用 Sub 语句或 Function 语句创建过程时,过程名之后必须紧跟括号,括号中包含所有参数,参数间用逗号分隔。例如,在下面的示例中,fDegrees 是传递给 Celsius 函数的值的占位符:

Function Celsius(fDegrees)

Celsius = (fDegrees - 32) * 5 / 9

End Function

要从过程获取数据,必须使用 Function 过程。请记住,Function 过程可以返回值;Sub 过程不返回值。

5.8.4 在代码中使用 Sub 和 Function 过程

调用 Function 过程时,函数名必须用在变量赋值语句的右端或表达式中。例如:

```
Temp = Celsius(fDegrees)
```

或

```
MsgBox"温度为"&Celsius(fDegrees)&"摄氏度。"
```

调用 Sub 过程时,只需输入过程名及所有参数值,参数值之间使用逗号分隔。不需使用Call语句,但如果使用了此语句,则必须将所有参数包含在括号之中。

下面的示例显示了调用 MyProc 过程的两种方式。一种使用 Call 语句:另一种则不使用。两种方式效果相同。

Call MyProc(firstarg, secondarg)

MyProc firstarg, secondarg

请注意,当不使用 Call 语句进行调用时,括号被省略。

5.9 VBScript 函数

要在程序中实现更全面的功能,进行更复杂的运算,更灵活地实现程序的功能,必须 掌握一些重要的函数,从某种意义上说,对 VBScript 函数的掌握和熟悉程度反映了用户的 编程水平。

5.9.1 数学函数

数学函数如表 5-5 所示。

表 5-5 VBScript 数学函数

函数	说明
Atn(A)	返回 A 的反正切
Cos(A)	返回 A 的余弦
Sin(A)	返回 A 的正弦
Tan(A)	返回 A 的正切
Abs(A)	返回 A 的绝对值
Exp(A)	返回 A 以 e 为底的对数
Log(A)	返回 A 的自然对数
Rnd(A)	返回一个随机数
Sgn(A)	返回 A 的正负符号
Sqr(A)	返回 A 的平方值

5.9.2 格式化函数

VBScript 中的格式化函数如表 5-6 所示。

表 5-6	VBScript	格式化函数
-------	----------	-------

函数	说明
Cboo(expression)	将 expression 转换成布尔子类型
Cbyte(expression)	将 expression 转换成单字节子类型
Cdate(expression)	将 expression 转换成日期子类型
CDbl(expression)	将 expression 转换成双字节子类型
Chr(expression)	将 expression 转换成一个字符
CInt(expression)	将 expression 转换成整数子类型
CLng(expression)	将 expression 转换成长整数子类型
Csng(expression)	将 expression 转换成单精度实数子类型
Cstr(expression)	将 expression 转换成字符串子类型
Hex(expression)	将 expression 转换成十六进制字符
Int(expression)	将 expression 取整
Fix(expression)	将 expression 取整
Oct(expression)	将 expression 转换成八进制字符
Val(expression)	将 expression 转换成一个数

5.9.3 日期时间函数

VBScript 中的日期时间函数如表 5-7 所示。

表 5-7 VBScript 日期时间函数

函数	说明
Date()	返回系统的当前日期
DateSerial(year,month,day)	返回日期子类型
DateValue(String)	将字符型转换成日期型
Day(Date)	返回给定日期中的天
Hour(Time)	返回给定日期中的小时
Minute(Time)	返回给定日期中的分钟
Month(Date)	返回给定日期中的月数
Now()	返回当前系统的日期和时间
Second(Time)	返回给定时间的秒数
Time()	返回当前系统的时间
TimeSrial(Hour,minute,sec)	返回时间子类型
TimeValue(Date)	将字符串转换成时间型
WeekDay(Date)	返回给定日期的周数
Year(Date)	返回给定日期的年份

5.9.4 字符串操作函数

字符串操作函数如表 5-8 所示。

表 5-8 VBScript 字符串操作函数

函数	说明
Asc(string)	返回第一个字符的 ANSI 符号
InStr(string1, string2)	返回 string2 中 string1 的位置
Lcase(string)	将 string 转换成小写
Left(string,num)	返回字符串左边的 num 个字符
Len(string)	返回字符串的长度
Ltrim(string)	去掉 string 左边的空格
Mid(string,start,len)	返回 string 中从第 start 个字符开始的 len 个字符
Right(string,num)	返回 string 右面的 num 个字符
Str(num)	将 num 转换成字符串型
Rtrim(string)	去掉字符串右边的空格
StrComp(string1, string2)	比较两个字符串
String(len,char)	创建含有 len 个字符的字符串
Trim(string)	除去字符串两端的空格
Ucase(string)	将 string 转换为大写

5.9.5 变体子类型

变体子类型函数如表 5-9 所示。

表 5-9 VBScript 变体子类型函数

函数	说明
IsArray(Var)	如果 Var 是数组,则返回 true
IsDate(Var)	如果 Var 是日期,则返回 true
IsEmpty(Var)	如果 Var 没有值,则返回 true
IsNull(Var)	如果 Var 是空值,则返回 true
IsNumber(Var)	如果 Var 是个数字,则返回 true
IsObject(Var)	如果 Var 是一个 Object,则返回 true
VarType(Var)	将 Var 的类型作为一个整数返回

VBScript 的函数都很简单,直接调用就可以了,但在使用过程中需要注意输入值的类型限制。对于具体的语法解释,参考相应的资料。

第6章 ASP 对象

熟悉了 VBScript 的语法后,还需要了解 ASP 的对象,使用这些对象可以完成功能更强 大的操作,比如和数据库的连接、前台页面信息的处理等等。这里我们先浏览一下内嵌对 象:

- ◆ Application 对象 被用来存储一个应用中所有用户共享的信息,例如可以利用 application 对象实现不同用户间的信息传递。
- ◆ Request 对象 可以用来访问所有从浏览器到服务器间的信息,也就是说,可以利 用 request 对象来接受用户在前台页面 form 中的信息。
- ◆ Response 对象 用来将信息发送回浏览器。可以利用 response 对象将脚本语言运行结果输出到浏览器上。
- ◆ Server 对象 提供运用许多 server 端的应用函数。例如,可以利用 server 对象来 控制脚本语言在超过时限前的运行时间。也可以利用 server 对象来创建其他对象 的实例。
- ◆ Session 对象 用来存储一些普通用户滞留期间的信息,可以用 Session 对象来储 存一个用户访问站点的滞留时间。
- ◆ ObjectContext 对象 可以用来控制 Active Server Pages 的执行。这种执行过程由 Microsoft Transaction Server (MTS) 来管理。

内嵌对象不同于正常对象。利用内嵌对象的脚本时,不需要首先创建一个实例。在整 个网站的应用中,内嵌对象的所有方法、集合以及属性都是自动可访问的。

下面逐一介绍这些对象的使用方法和使用过程中应该注意的问题。

6.1 Request 对象

Request 对象是 Active Server Pages 编程中使用最频繁的对象之一,它对应于 HTTP(超 文本链接协议)请求。Request 对象的使用使得 Web 服务器端和 Web 客户端紧密地联系在 一起。和绝大部分 Active Server Pages 对象一样, Request 对象也包括集合、属性以及方法, 利用 Request 对象的集合、属性和方法可以接受任何浏览器到网站的请求。

Request 对象的调用语法结构如下:

Request[.Collection] ("variable")

其中, Collection 集合可以被指定为 From 或者 QueryString, Variable 就是提交页面中 使用对象的变量名。

6.1.1 使用 QueryString 集合

1. 接受单个的 Query 字段

QueryString 是 Active Server 搜索的第一个对象集合。这是因为, QueryString 是从 HTTP 的 Get 方法中得到的,而 Get 方法是 HTTP 中缺省的请求方法。

应用中通常在 URL 地址后面带有 Query 字段来传递参数,而在接收页中用 QueryString 集合来对 Query 字段进行访问。如下面的例子所示,可以在指定的链接中使用?带 Query 字段来传递参数信息:

<html>

<head>

<title>超链接</title>

</head>

<body>

Query 字段示例

</body>

</html>

注意:上面实例的文件名为 test.asp,存放在服务器端的/bbs 虚拟目录中,接收字段信息的页面 test1.asp 也是放在同一目录下,所以在链接中不需要写入完整的地址和路径。

在这个例子中,超链接的地址是 test1.asp。但这个链接包括一个 Query 字段"id=1"。 通常,也可以在浏览器的地址栏中直接键入地址和文件名后面带上 Query 字段来进行信息 传递,例如在浏览器的地址栏中输入:

http://wjy/bbs/test1.asp?id=1

对于服务器端来说,这二者是没有区别的。

可以在 test1.asp 文件中用 Active Server Pages 的 Request 对象的 QueryString 集合来接 收这些信息。要想接收一个特定的 Query 字段内容,只需要写出 Query 字段的名称就可以 了。在 test1.asp 中,可以用 Request.querystring("id")来访问 id 字段的内容。Test1.asp 的 内容如下:

```
<html>
<head>
<title>接收字段</title>
</head>
<body>
<%=request.querystring("id")%>
</body>
```

</html>

当在 test.asp 中点击 Query 字段示例的超链接时,浏览器将显示出 test1.asp 的内容,将显示出数字 1。

在进行菜单选择时,Query 字段是非常有用的。如果在一个主页中选择不同的链接, 传递的参数值不同,但又希望在一个页面中处理,使用Query 字段便可以实现这样的功能。 例如:

```
<!---page.asp-->
<html>
<head>
<title>翻页实例</title>
</head>
<body bgcolor="#ccccff">
<%
if requst.querystring("operator")~"" then
=request.querystring("operator")
end if
%>
<a href="page.asp?operator=previous">前一页</a>
<a href="page.asp?operator=next">后一页</a>
</body>
</html>
```

这是在 Web 数据库编程中经常采用的分页显示实例中的一部分代码,可以看到,采用 这样的方式可以实现对多条记录的分页显示。从上面的代码可以看出,页面中提供两个超 链接,即前一页和后一页,超链接的对象文件都是自身,但传递的参数不同,根据收到的 不同参数,可以选择显示相应区段的记录,达到分页的效果。

注意:对多条记录进行分页显示需要完整的代码,这里只是用来说明在分页显示 中用到 Query 字段的部分。完整的分页实例见 7.3 节"分类讨论区"在后面的章 节中可以找到。另外,这里超链接的对象是自身文件,这样的传递在 ASP 编程中 是允许的。

2. 使用 Encode 操作

如果通过 Query 字段传递的是一个很长的字符串,例如:

测试语句

那么当在 test1.asp 中使用 Request.QueryString("message")方法来显示 message 的值时,可能会发现屏幕上显示的信息不是预料中的输出结果:

这是一个 Query 字段的测试语句。

而是出现了一些奇怪的字符串。这是因为,当 Query 字段有一些特殊字符,例如空格和标 点符号等,仅仅用" "括住的字符串信息是不能被正确传递的。这时必须使用 Encode 操作对 字符串进行处理。

Encode 操作对应的特定方法是 Server.URLEncode(), 例如:

<a href="test1.asp?message=<%=Servr.URLEncode("这是一个 Query 字段的测试语句. ")%> ">测试语 句

这时在 test1.asp 中使用<%=request.querystring("message")%>时就能得到正确的结果。 需要注意的是,不要对 Query 字段的名称以及等号进行 Encode 操作,而只需对其值进行操 作就可以了。

进行 Encode 操作后, Query 字段的内容如下:

这是一个 Query 字段的测试语句。

%D5%E2%CA%C7+%D2%BB%B8%F6Query%D7%D6%B6%CE%B5%C4%B2%E2%CA%D4%D3%E F%BE%E4%2E

其中的空格变成了+号,而所有的汉字都变成了相应的编码。 Active Server Page 会自动对上面的字串进行解码,最后得到正确的结果。

3. 使用 Query 字段传递多个参数

有时需要在一个链接中一次传递多个参数,这时同样可以采用 Query 字段来完成这样的功能。对上例中的 test1.asp 进行一定的修改,如下所示:

<html>

```
<head>
```

```
<title>超链接</title>
```

</head>

<body bgcolor="#ccccff">

```
<a href="test1.asp?id=1&name=pett">Query 字段示例</a>
```

```
</body>
```

</html>

从第6行代码可以看出,在这个超链接中一次传递了两个参数,即"id=1"和"name=pett",不同的参数之间用连接符&连接。显然,通过这样的方法可以一次传递更多的参数。

注意:这里的连接符号"&"十分重要。&符号可以看作是一个参数变量的结束,如果不使用&符号而使用别的符号来代替,比如使用分号(;),则其中的1; name=pett将视为一个完整的字符串作为参数"id"的值。

相应的接收这两个参数的 test1.asp 文件内容如下:

<html>

<head>

<title>超链接</title>

</head>

<body bgcolor="#ccccff">

<%=request.querystring("id")%>

<%=request.querystring("name")%>

</body>

</html>

在浏览器中显示的效果如图 6-1 所示。

∋超链接 - Microsoft Internet Explorer	
ANNO INCO INCO AND DOM IN	6.6
	365 2
http://wjy/bbs/test1.asp?id=18name=pett	
Jack Contract of the second	
	net /

图 6-1 test1.asp 的结果

当在 Query 字段中将同一个参数名写多遍时,可以利用一个参数对应多个值。例如:

<a href="test1.asp?message=<%=Server.URLEncode("这是传递的第1条信息。")%> &message=<%=Server.URLEncode("这是传递的第2条信息。")%>">Query 字段测试

在这里,参数 message 对应两个值,即"这是传递的第1条信息。"和"这是传递的第 2条信息。"这时,千万不能以其他编程语言中的变量思想来看待这个问题,以为后面的第 2 个参数值会覆盖前面的第1 个参数值,实际上这里的 message 是一个对象,可以包含一系 列的值,可以理解为一个数组。Request 对象的 Count 属性可以在这里用来检查到底有多少 个值。Pvalue 属性用来返回 QueryString 集合中的元素值。

4. 输出 QueryString 集合中的所有参数

使用循环语句可以输出 QueryString 集合中的所有参数。 下面的脚本是利用 For Each 循环:

<%

FOR EACH QSParam IN Request.QueryString Response.Write("
&QSParam&"=") Response.Write(Request.QueryString(QSParam) NEXT

%>

除了使用 FOR EACH 循环外,还可以使用 FOR...NEXT 循环来进行这种操作。这样的话,首先需要利用 Count 属性来了解具体值的数量,例如下面这个例子:

<%

FOR I=1 TO Request.QueryString.Count

Response.Write("
"&Request.QueryString(i))

NEXT %>

当然也可以利用最简单的方法来接受没有解码的 Query 字段。见下面这个例子:

<%=Request.QueryString%>

5. 不能使用 Query 字段的场合

Query 字段在主页间传递参数时非常有用,但有两种场合最好不要使用:

- ◆ 传递隐含信息
- ◆ 传递大量信息

这是因为,从某种意义上说 Query 字段根本无法隐含,这就意味着如果用来传递用户的密码就十分糟糕,任何人都可以看见它。另外,对于不同的浏览器来说,传递的数据量最大值有不同的限制。例如,IE4.0 无法处理超过 200 个字符的 Query 字段,当 URL 链接带的 Query 字段超过这个长度时,浏览器就无法正确理解它们。

总之,利用 Query 字段传递的信息应当是简洁的和非保密的,如果有大量数据需要传递,那么可以利用 HTTP 中 Form 的 Hidden (隐藏表单)类型。HTTP 协议在传递表单时比 传递 Query 字段有效得多。

6.1.2 使用 Form 集合

1. 显示 From 集合中所有字段的值

和 QueryString 集合中所使用的方法一样,通常可以用循环来显示上个页面 Form 中所 包含的内容。

(1) For Each 方法

<%

FOR EACH name IN Request.Form Response.Write(""&name&"=") Response.Write(Request.form(name)& "") %> 使用上面的程序段可以显示出表单传递的键名和相应的值,并且是分行显示。

注意:在 Form 集合中,大小写是通用的。两者的使用没有任何区别。

(2) For Next 方法

<%

FOR I=1 to Request.Form.Count

Response.Write(""&Request.Form(i)& "")

NEXT %>

使用上面的 FOR NEXT 方法也可以显示所有表单值。

2. Form 域中对应于 CheckBox 和 Radio 提交表单值的处理

有时, Form 的某个域对应多个值,例如,用 Checkbox 类型来实现多项选择,以及用 Radio 类型来实现单项选择。当运行多选时,可以调用 Request.Form("name").Count 来得到 复选框中的选择数目而不是所有 Form 域提交的数目。

3. 测试 Form 域是否存在

在前面曾经提到过检查用户是否填写了正确的注册信息。如果有些填写的域不希望用 户空置,那么可以利用下面的简单脚本来检验:

<%

```
If request.form("Username") > " " then
```

```
Response.write("注册成功")
```

Else

```
Response.write("必须输入姓名")
```

End if

%>

当然,这个验证程序的前提是假设提交的页面中有名称为 Username 的 Form。这个脚本测试用户是否填写了姓名项,检查过程是与一个 0 长度的字段进行比较,如果用户填写 了,则不等式成立,进行用户注册,如果没有填写,则不等式不成立,显示提示填写的信息,还可以使用 Redirect 方法引导用户返回上页。关于 Response 对象的使用方法,将在下面部分介绍。

6.2 使用 Response 对象

Response 对象负责管理服务器端和客户端的交互。Response 对象用于向客户端浏览器 发送数据。用户可以使用该对象将服务器端的数据用 HTML 超文本的格式发送到客户端的 浏览器。该对象的功能和 Request 对象的功能恰恰相反。Request 对象是用于得到用户提交的数据,而 Response 对象用于将服务器端的数据发送到用户的浏览器,这两者都是实现动态的基础。

Response 对象有以下的几种方法:

- ♦ AddHeader
- AppendToLog
- BinaryWrite
- Clear
- End
- Flush
- Redirect
- Write

除 Write 方法外,其他方法都是 Response 对象较高级的特征。在前面的例子中已经使用了 Response.Write 方法来向客户端返回信息。这一节介绍 Write 方法、End 方法、Clear 方法和 Redirect 方法在 ASP 编程中的应用。

6.2.1 Write 方法

Response 对象的 Write 方法是 ASP 编程中最基本的方法,它向客户端的输出流执行写的操作。它的语法结构很简单:

Response.write(字符串变量)

同时,可以用<%=字符串变量%>方法来完成相同的工作。

注意: <%和%>必须配对使用。

6.2.2 End 方法和 Clear 方法

End 方法标志着一个 Response 输出流的结束, 其语法结构为:

Response.end

它通常被用于在判断用户提交的表单有不合法内容后结束程序,相当于子程序中的 End Sub 语句的用法。

Clear 方法用于将客户端的浏览器清屏,其语法结构为:

Response.clear

6.2.3 Redirect 方法

Response 的 Redirect 方法可以用来引导用户从一个主页到另一个主页上。在判断用户

提交的表单中有不正确的信息时,自动将用户返回注册界面。

在一般的密码判断程序中都使用这一语句。当程序判断出用户的身份不合法时,可以 调用 Redirect 语句,强制非法用户离开当前的页面,或者根据用户的不同身份引导到不同 的页面。Redirect 语句的语法结构为:

Response.redirect "相对路径/文件全名"

Redirect 语句一般和判断语句(IF)嵌套使用,根据不同的客户选择引导到相应的页面。

6.3 Session 对象

在创建自己的交互网页时,也许会遇到一个问题:如果用户访问的是一系列的网页,如何才能和用户保持联系,跟踪并为用户提供特定的服务呢? Session 对象将完成这样的一系列工作。

6.3.1 Session 对象简介

当 HTTP 协议工作时,用户发出请求,服务器在接到请求信号后作出响应,这样用户 和服务器端一问一答进行通信。这种用户和服务器端进行的联系是非连续性的。在服务器 端完成响应用户请求后,服务器端不能持续与该浏览器保持联系。这样,每一个新的用户 请求都是单独存在的,当同一个用户在不同主页间转换时,服务器端没有办法对用户进行 跟踪。Session 的引用将弥补这个缺陷,利用它,可以使一个用户在多个主页间切换的时候 也能保存他的信息。

在使用时,Session 实际上指的是一系列变量的集合,它们从一个页面到另一个页面跟 踪访问者,直到用户离开为止。通常,每个访问者在访问期间都会单独获得一个 Session。

我们可以使用 Session 对访问者的信息进行跟踪,譬如访问者的姓名、嗜好等等,从而 动态地创建具有用户个人特色的网页。例如,使用 Session 还可以创建虚拟购物篮,这样用 户能把购物信息保存在 Session 中。用户可以随时把喜欢的商品放入"购物篮"中,在离开 网站时一起定购。

ASP 中 Session 对象的使用很简单,如果在一个用户的 Session 中存储了信息,只要简 单地直接调用 Session 对象就可以了。

Session 指的就是访问者从到达某个特定主页到离开为止的那段时间,每个访问者都会 单独获得一个 session。

Session 是利用 Cookie 进行信息处理的(参见 6.4 节 "Cookies 对象"),当用户首先进行了请求后,服务端就在用户浏览器上创建了一个 Cookie,当这个 Session 结束时,就意味着这个 Cookie 过期了。为这个用户创建的 Cookie 的名称是 ASPSESSIONID。这个 Cookie 的惟一目的就是为每一个用户提供不同的身份认证。

Session 变量自己不会存放在用户浏览器上。不过,ASPSESSIONID 这个 Cookie 需要 使用 Session 变量。Server 使用 ASPSESSIONID Cookie 来将特定的用户和特定的 Session

信息联系起来。没有 Cookie 的话, Server 就不会了解到每一个特定用户在网站中移动的信息。

利用SessionID变量存储ASPSESSIONID Cookie和直接对名为ASPSESSIONID Cookie 赋值有很大不同。微软利用了一个复杂的数学算法对SessionID进行加密,以防止黑客猜测出SessionID的值并且依据这个值获得不该获得的身份或权限。

注意:可以用两种方法屏蔽掉 SessionID: 一种是将全站进行屏蔽,另外一种是将 一个单独 Active Server Page 进行相应屏蔽。

如果想要将整个站点的 Session 操作进行屏蔽,可以使用 Internet Service Manager,从 Application 设置对话框点击 Active Server Pages 表,并且取消对 Enable Session State 选项的 选择即可。

还可以在特定 Active Server Page 的首行加入使之屏蔽的语句来进行这种操作:

<% EnableSessionState=False %>

由于 Session 对象使用了 Cookie,那么它的兼容性就受到了限制,一些老的浏览器显 然是不行的,新的浏览器如 NetScape 4.0 也提供了屏蔽 Cookie 的选项。

由于 Cookie 不能适用于所有浏览器,那么在开发程序时就必须注意,如果网站定位于 大众通用,就必须考虑各种不同的用户情况。不过现在确实有可以取代 Cookie 来进行身份 认证的方法。

注意:当前浏览器是否发送一个 Cookie 在 URL 是区分大小写的,因此最好使用 同样的大小写方式。

6.3.2 Session 对象的属性

1. SessionID 属性

SessionID 属性用于返回当前用户的惟一标识符。服务器将自动给每一个 Session 分配 不同的编号。如下例所示:

```
<HTML>
```

```
<HEAD><TITLE>SessionID</TITLE></HEAD>
```

```
<BODY>
```

```
<%=sessionID%>
```

%>

</BODY>

</HTML>

当执行这一段程序时,浏览器将显示出当前 Session 的编号。

2. TimeOut 属性

服务器端怎样判断一个 Session 是否结束了呢?也就是说,当前的客户端浏览器是否已

经转移到其他网站。如果客户端浏览器一直没有提出请求或者刷新主页长达 20 分钟,那么 服务器端将默认为用户已经离开了。这种策略就使得服务端可以释放对用户进程进行跟踪 时使用的资源。

这里的 20 分钟是 Session 对象的默认值。当然,对于不同网站所需要的时间长度不同, 对于需要用户长时间停留的网站,应该增长这个时间,但对于访问量大,信息随时更新的 网站,可以适当缩短这个时间限制。

通过 Session 对象的 TimeOut 属性可以很容易设置 Session 存在的时间。重新定义 TimeOut 的语句如下:

<% Session.Timeout=60 %>

注意:可以在 Internet 信息服务管理器中改变默认值。在当前目录的应用程序配置中可以设置这个默认值。

当用户的 Session 时间过期后,如果用户刷新了主页,那么将被认为是新的访问者,所 有以前的 Session 信息会全部丢失。也可以利用 Abandon 方法来消除一个 Session。这里再 引入一个 SessionID 属性,这将自动分别为每一个新的 Session 分配不同的编号。

3. 自定义属性

在 ASP 程序中可以自定义 Session 变量,使用方法很简单。例如:

<HTM>

```
<HEAD><TITLE>Session</TITLE></HEAD>
```

<BODY>

```
<%Session "user")= " pett"%>
```

```
<%response.write(Session("user"))
```

</BODY>

</HTML>

这个例子的显示结果为

pett

当定义了 Session 对象的属性后,属性数据就可以被 Session 用户提取,而无论浏览器 处于哪一个页面中,直到 Session 过期。

在 ASP 程序中可以定义多个不同的 Session 属性,用来存储客户端用户的多个数据。 几乎所有 Session 存储的内容存放在 Content 集合中。因此,下面两个语句是等效的:

<% Session("user")= "pett" %>

<% Session.Contents("user")= "pett" %>

和前面所叙述的集合一样,仍然可以利用 Count 属性来检查 Content 集合的数量。同样,可以利用 FOR EACH 或 FOR ...NEXT 循环来显示 Content 所有内容。下面的例子使用了这

些方法:

```
<%
Session("User")= "pett"
Session("pwd")= "123"
%>
Session 对象的 Content 集合共有<%=Session.Content.Count%>项。
<%
FOR EACH thing IN Contents
Response.Write("<br>"&thing&Session.Contents(thing))
NEXT
%>
<%
FOR I=1 to Session.Contents.Count
Response.Write("<br>"&Session.Contents(i))
NEXT
```

在这段程序中定义了两个 Session 变量,即 User 和 pwd,然后,依次通过 FOR EACH 和 FOR...NEXT 循环将这两个字段内容显示出来,和前面的章节中所述十分类似。

显示结果为:

```
Session 对象的 Content 集合共有 2 项。
User=pett
pwd=123
pett
123
```

6.3.3 Session 事件

和其他对象不同的是, Session 对象中有事件(Event)。共两种: 一个是 Session_OnStart 事件, 当一个 Session 开始时被触发; 还有一个是 Session_OnEnd 事件, 当一个 Session 结 束时被触发。在一个脚本中你可以和其中一个并且只能和其中一个事件关联。

在事件触发时下面这些脚本语句被执行。这两个脚本位于特定的文件 Global.asa。这个 文件位于网站应用的根目录,它包括了一些通用程序段和网站应用。Global.asa 文件有如下 结构:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
END SUB
</SCRIPT>
```

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

SUB Application_OnEnd

END SUB

</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

SUB Session_OnStart

END SUB

</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

SUB Session_OnEnd

END SUB

</SCRIPT>

Global.asa 包括 4 个脚本,这里面有一个是根据 Session_OnStart 触发,另一个是根据 Session OnEnd 触发。

请注意,Global.asa 使用了微软的 HTML 拓展<SCRIPT>标记语法来限制脚本,也就是 说,必须使用<SCRIPT>标记来引用这两个事件而不能用<%和%>符号引用。例子中 Global.asa 使用的是 VBScript,当然也可以使用其他脚本语言。

在 Global.asa 中不能有任何输出语句,无论是 HTML 的语法还是 Response.Write()方法都不行,Global.asa 任何情况下也不能进行显示。

只需要在 Global.asa 中添加一些希望执行的脚本,那么只要一创建 Session,这些脚本 就会自动执行,例如:

<SCRIPT LANGUAGE=VBScipt RUNAT=Server>

SUB Session_OnStart

Session("Username")= "Unknow"

```
Session("Userpassword")= "Unknow"
```

END SUB

</SCRIPT>

这个脚本将"Unkown"值赋给了 Username 和 UserPassword 变量。这个例子将在任何一个 Session 变量创建的时候执行。

Session_Onstart 脚本可以用于多种目的。例如,希望访问者必须浏览某一个主页。下面的例子在用户进程开始时进行了这种引导,这里面使用了 Response.redirect 方法:

<Script Language=VBScript RUNAT=Server>

SUB Session_OnStart

MyHomepage="/homepage.asp"

RequestPage=Request.ServerVariables("SCRIPT_NAME")

IF NOT (STRCOMP(MyHomePage,RequestPage,vbTextCompare)=0) THEN

Response.Redirect MyHomePage

END IF

END SUB

</SCRIPT>

在这个脚本中,用户请求和主页路径进行比较,如果不一样,用户就被自动引导到该 主页。

下面的例子使用了 Session_OnStart 和 Session_OnEnd:

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

SUB Session_OnStart

Response.AppendToLog Session.SessionID&" Starting"

END SUB

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

SUB Session_OnEnd

Response.AppendToLog Session.SessionID&" Ending"

END SUB

</SCRIPT>

在这个例子中,当用户的 Session 开始时,日志文件中记录了该用户的 Session 开始信息;当用户的 Session 结束时,日志文件就记录了该用户的 Session 结束信息。这样,就可以作很多判断统计,例如每个人的停留时间、站上现在有多少人等等。这对于站点设计和 定位就很有裨益。

6.4 Cookie 对象

Cookie 是一种发送到客户端浏览器的文本串句柄,并保存在客户机磁盘上,可以用来 在某个 Web 站点会话期间持久地保持数据。

浏览器用一个或多个限定的文件支持 Cookie。这些文件在 Windows 操作系统中叫做 Cookie 文件,网站用来在上面存储 Cookie 数据。服务器端程序开发者可以在这些 Cookie 文件中插入信息。这样就带来了另外的一个问题,如果服务器端程序存在一些不利于客户 的程序,有可能会侵犯用户的隐私,甚至带来更大的破坏。

目前有些 Cookie 是临时的,还有一些则是持续的。例如,一些 Cookie 被 Active Sever Pages 用来跟踪用户进程直到用户离开网站,另外一些 Cookie 则保持在 Cookie 文件中直到 用户返回时又进行调用。

6.4.1 Cookie 的工作原理

Cookies 将通过 HTTP Headers 来从服务器端返回到浏览器上。服务器端首先在响应中 利用 Set-Cookie header 来创建一个 Cookie,浏览器后面请求的 cookie header 中就会返回这 个 Cookie 来完成浏览器的认证。 假设创建了一个名字为 UserName 的 Cookie 来包含访问者的信息, 创建 Cookie 时, Server 的 Header 就像下面这样:

Set-Cookie: UserName=BILL+Gates;path=/;domain=aspsite.com; expires=Tuesday,01-Jan-99 00:00:01 GMT

这个 Header 就在浏览器电脑上的 Cookie 文件中添加了一条记录。浏览器将名字为 UserName 的 Cookie 赋值为 Bill Gates。请注意,这个 Cookie 的值是进行了 URL-encoded 操作的。

后来,Header 通知浏览器将 Cookie 通过请求以忽略路径的方式返回服务器端,因此, 一个 Cookie 设定后,其应用的所有文件就必须在同一个目录下。例如,如果开始指定的路 径是/private 目录,那么 Cookie Header 就可以请求文件/private/mypage.asp,而/mypage.asp 由于路径变动就无法利用这个 Cookie 了。

domain 属性能够在浏览器端对 Cookie 发送进行更多限制。在这个例子中, Cookie 只能传到指定的服务器上,而决不会跑到 www.yahoo.com 或者其他什么网站。

注意:现在的浏览器在判断 Cookie 的路径时区分大小写,这就意味着,如果路径 是/private,那么以/PRIVATE 路径方式就无法进行这个 Cookie 的调用和认证。

最后, Expires 标记限定了 Cookies 的过期时间,在例子 Header 中,限定浏览器将该 Cookie 保存到 1999 年 1 月 1 日第 1 秒,实际上,当浏览器接受的 Cookie 很多时,还会自 动进行删除。

浏览器创建了一个 Cookie 后,对每一个针对该网站的请求都会在 Header 中带上这个 Cookie,也就是每当满足该路径的情况下这个 Cookie 都会有效。不过,对于其他网站的请 求 Cookie 是绝对不会跟着发送的。浏览器会这样一直发送到 Cookies 过期为止。Cookie Header 如下:

cookie: username: Bill+Gates

可以在 Active Server Pages 中创建和读取 Cookies。当利用 Active Server Pages 创建了 一个 Cookie 之后,你就可以使用 Response 对象的 Cookie 集合了。你可以创建两种 Cookie; 一种是单值的,另一种可以认为是 Cookie 字典类型,即允许存在多个键值对。

创建单值的相对简单,脚本如下:

<% Response.Cookies("Username")= "Bill Gates"

Response.Cookies("Username").Expires="Jan 1,1999"

%>

这个脚本的工作一目了然,将名字为 Username 的 Cookie 赋值为 Bill Gates, 同时将 过期时间限定为 1999 年 1 月 1 日。需要说明的是, Expires 属性如果不进行赋值,那么默 认的就是用户一离开网站就过期。

由于这个例子脚本创建的是 Header 的部分,那么你就必须在你的 Active Server Pages 的任何输出语句之前进行这个脚本的操作,或者使用 Buffer 输出。
前面的脚本是创建一个 Cookie 的简单示例,只是使用了最常用的 Expires 属性,其实还有许多其他属性也可以自行设置,下面是一个比较完整的例子:

<%

Response.Cookies("Username")= "Steve Jobs" Response.Cookies("Username").Expires="Jan 1, 1999" Response.Cookies("Username").Path="/examples" Response.Cookies("Username").Domain="aspsite.com" Response.Cookies("Username").Secure=True %>

这个脚本例子和前面的其实没有什么区别,不过有3个附加的属性需要解释:

- ◆ Path 属性是用来更加严格地限定浏览器发送 Cookie 的。在这个例子中,只有针对 /examples 目录请求的 Header 中才携带 Cookie 信息,例如针对/examples/hello.asp 以及/examples/chapter16/hello.asp 的请求都会在 Header 中携带 Cookie 信息,如果 是浏览器针对/hello.asp 的请求,就不会携带该 Cookie 信息。Path 属性的默认值是 该 Cookie 创建的 Active Server Pages 所在的路径。也就是说,即便不做指定,也 不会跨过目录发送 Cookie。
- ◆ Domain 属性限定了 Cookie 发送的网站。例子中的 aspsite.com 说明 Cookie 可以被 发送到 www.aspsite.com 或者 beetle.aspsite.com 或者 yeah.aspsite.com 等等。同样, 作为默认值是该 Cookie 创建的网站。
- ◆ 最后是 Secure 属性,顾名思义,该属性设为 True 则传递中就实行了加密算法。如果你正在使用安全接口层,那么你就可以使用这个属性。

在一个 Active Server Page 中读取 Cookie, 你只需要使用 Request 对象的 Cookies 集合。 例如, 输出一个 Cookie 值, 那么脚本如下:

<%=Request.Cookies("Username") %>

这个脚本将名字为 Username 的 Cookie 值进行了输出。和以前一样,你依然可以利用 For Each 循环或者利用 Count 属性和 For ...Next 循环结合的方式来将 Cookie 集合的所有属 性值显示出来。下面这个例子的运行结果应当无须解释了:

<%

For EACH thing IN Request.Cookies

Response.write("
"&thing&Request.Cookies(thing))

NEXT

%>

6.4.2 创建多个 Cookie

你当然还可以创建不止一个 Cookie, 只要在 Response 对象的 Cookies 集合中简单地定

义多个名称就可以了。不过,许多浏览器对一个指定网站就限定了3到4个 Cookie。

创建多个 Cookie 还有一种选择,就是创建一个 Cookie 字典,一个 Cookie 字典中可以 含有多个键值对。下面是一个字典的例子:

<%

Response.Cookies("User")("Name")= "Bill Gates" Response.Cookies("User")("Password")= "billions" %>

这个脚本创建了一个名为 User 的 Cookie 字典,其中含有两个键,分别是 Name 和 Password。当这么一个 Cookie 字典创建时,请求的 Header 中是这样的信息:

Set-Cookie:User=Name=Bill+Gates&Password=billions

一个名字为 User 的 Cookie 创建了,其中含有两个键值对,这意味着所有的键和相应的值都在一个大的 Cookie 中。

接受这样的 Cookie 值,你还可以利用以前的 Response 对象的 Cookies 集合,既可以将 其全部显示(这样显示就是没有经过解码的 Header 中的源代码,也就是上面 Header 中的 信息,一般都是用于调试工作),也可以按每一个键的相应名称显示相应值,如下例所示, 无须解释结果:

<%=Request.Cookies("User") %> <%=Request.Cookies("User")("Name")%> <%=Request.Cookies("User")("Name")%>

注意:利用 Cookie 技术传递诸如密码这样的信息要特别小心,因为一般说来这种 信息是未经加密的。当然,如果你的网站有安全接口层技术,也可以进行加密传 输。但是,在浏览器端该信息还是存放在文本文件中,是可以进行修改的。

如果希望知道一个 Cookie 是否是一个 Cookie 字典,可以使用 HasKeys 属性。例如, 下面的脚本如果返回值为 True,那么就是一个 Cookie 字典:

<%=Request.Cookies("User").HasKeys %>

6.5 Application 对象

ASP 的 Application 对象用来存储所有用户的共享信息,通过 Application 对象可以实现 不同用户之间的信息传递。

6.5.1 Application 对象简介

一个 Application 就是在硬盘上的一组主页以及 ASP 文件, 当一个 Active Server Pages

加入了一个 Application 后,那么它就拥有了作为单独主页所无法拥有的属性。下面是 Active Server Pages 的 Application 的一些特性:

◆ 数据可以在 application 内部共享,因此可以覆盖多个用户。

- ◆ 一个 Application 包含的事件可以触发某些 Application 脚本。
- ◆ 一个对象的实例可以被整个 Application 共享。
- ◆ 特殊的 Application 可以用 Internet Service Manager 来设置而获得不同属性。
- ◆ 单独的 Application 可以隔离出来在某一客户端的内存中运行,因此,如果某一位 用户的 Application 遭到破坏,不会影响其他用户。
- ◆ 可以停止一个 Application 而不会影响到其他用户。

一个网站可以创建多个 Application。可以根据针对个别任务的一些 ASP 文件创建个别的 Application。例如,可以创建一个 Application 用于所有的公用用户,而另外创建一个 Application 限制在系统管理员的范围。

Application 的目录由 Internet Service Manager 来设定,设定的目录以及其子目录均属 于这个 Application。不过,如果某一个子目录也进行了设定,将被认为是另一个 Application, 换句话说,两个 Application 可以重叠交错。

创建了一个新的 Application 后,就可以在 Application 设置区域进行属性设置,例如限 定是否使用 Session 和进行 Buffer 输出等等。

另外,进行这种设置之后就可以使用自己的 Global.asa 文件来创建自己的事件。

6.5.2 使用 Application 对象

Application 对象拥有与 Application 所有相关的集合、方法和事件。下面的部分讲述如何使用 Application 对象来创建 application 变量和 application 事件。

1. Application 变量

一个 application 变量包含的数据可以在整个 application 中被所有用户享用。Application 对象包括任何类型,包括队列和对象。application 变量和 Session 变量的区别有以下两点:

- ◆ 和 Session 不同的是, Application 变量不需要 Cookies。网站不需要利用 Application 变量来跟踪用户进程。这就意味着使用起来很安心,可以适应任何浏览器。
- ◆ 和一个 Session 变量不同的是, application 变量可以被多个用户共享。从一个用户 接受到的 application 变量可以传递给另外的用户。

Application 的应用方式:

- ◆ application 变量可以用来在每个主页上显示动态信息。例如,对每一个页面进行 当日新闻的更新。
- ♦ Application 可以用来记录广告条被点击的时间和次数。
- ◆ Application 可以从数据库中读取数据,例如,可以从网站的数据库中查询某些数据信息,然后利用 application 变量显示在多个页面上。

- ♦ Application 变量可以用来统计目前上站人数。
- ◆ application 变量可以用于网站上不同用户间的通讯,这样就可以创建多用户游戏 以及多用户聊天室。
- (1) 创建和读取 Application 变量。

Application 变量的创建方法和 Session 变量的创建方法一样,例如:

```
<HTML>
<HEAD><TITLE>Application 示例</TITLE></HEAD>
<BODY>
<%
Application("user")= "pett"
%>
<%=Application("Greeting")%>
</BODY>
</HTML>
```

在这个示例中创建了一个名称为 user 的 Application 变量,赋值为 "pett"。然后这个变量值 "pett" 将显示在浏览器上。

一旦变量被赋值,它就可以在任何 ASP 文件中显示出来,即使是另外的客户端。例如, 这时候另外一个用户调用下面的 ASP 文件,也可以显示 "pett" 信息:

```
<HTML>
<HEAD><TITLE>另外一页</TITLE></HEAD>
<BODY>
<%=Application("pett") %>
</BODY>
</HTML>
```

由于 Application 变量是对于多用户共享的,因此与 Session 变量有着本质的区别。同时, Application 变量不会因为某一个甚至全部用户离开就消失,一旦建立了 Application 变量,那么它就会一直存在,直到网站关闭或者这个 Application 被卸载。

由于 Application 变量创建后不会自己消亡,它占用系统内存,因此需要斟酌使用,以 免降低服务器对其他请求的响应速度。

注意: Application 变量终止的情况有 3 种:服务被终止,Global.asa 被改变,或者 该 Application 被卸载。

Application 变量也存在一定的问题:如果多人同时调用,程序的运行将会出现错误。 例如下面这个例子,假如有两个用户同时点击,将会发生程序冲突:

<%

NumClicks=Application("BannerClicks")

NumClicks=NumClicks+1

Application("BannerClicks")=NumClicks

%>

有一种方法可以解决这个问题,分别是 Lock 方法和 UnLock 方法,用来暂时防止其他 用户改变 Application 变量。那么前面的例子正确形式应该如下:

<%

Application.Lock NumClicks=Application("BannerClicks") NumClicks=NumClicks+1 Application("BannerClicks")=NumClicks Application.UnLock %>

脚本的第一行将 Application 进行 Lock 操作,这样其他用户就无法对 Application 变量 进行修改,直到 UnLock 操作或者该 ASP 文件执行结束。

注意, Lock 方法不是永远不让其他用户进行 Application 变量的修改, 而是让修改过程 连续有序, 防止冲突。

(2) 显示所有 Application 对象。

绝大部分 Application 变量存放于 contents 集合中,一旦创建了一个新的 Application 变量,其实就是在 contents 集合中添加了一项。例如,下面两个脚本是等效的:

<%Application("user")= "pett"> <%Application.contents("user")= "pett">

由于 Application 变量存放在集合里面,如果想要全部显示,可以和前面的集合一样使用循环语句来实现,这里不再赘述。

2. Application 事件

与 Session 一样, Application 也有两个事件,即 Application_OnStart 事件和 Application_OnEnd 事件,前者在 Application 开始时触发,后者在结束时触发。

Application 的开始时间从第一个请求 Application 的页面被调用算起。

一个 Application 的 OnStart 事件肯定是在 Session_Start 事件之前。不过, Application 不会像 Session 那样在一个新用户请求后就触发, Application 只触发一次, 就是第一个用户的第一次请求。

Application_OnEnd 事件只有在服务终止或者该 Application 卸载时才会触发。例如,在 Internet Service Manager 中关闭了网络服务,那么 Application_OnEnd 事件就会触发;如果 针对单独目的使用 Application,这个事件可以通过 Application 利用 Unload 按钮卸载时进行 触发。一个 Application OnEnd 事件肯定发生在 Session OnEnd 事件之后。

Application OnStart 和 Application_OnEnd 事件都是触发一个也是惟一一个脚本程序。

而且,这些事件都在一个文件中,那就是在 Session 事件中提到的 Global.asa 文件。关于 Global.asa 文件,这里不再赘述。

6.6 ASP 数据库组件

这一节将介绍 Activex 数据对象(ADO)。首先是 ADO 对象的概述, 然后一步一步地介绍使用 ADO 从数据库中读取数据和向数据库中存储数据的方法。

6.6.1 ActiveX 数据对象概述

使用 Activex 数据对象 (ADO),可以对来自许多数据提供者的数据进行读取和写入操作。例如,可以使用 ADO 来访问 Microsoft Access、Microsoft SQL 和 Oracle 数据库中的数据。甚至可以使用 ADO 从 Microsoft Excel 表格中读取数据。

Activex 数据对象共有 7 种独立的对象。下面列出这 7 种对象的名字,并简要介绍了它 们的功能:

- ◆ 连接对象。代表与一个数据源的惟一对话。例如,可以用连接对象来打开一个对 Microsoft SQL Sever 的连接。
- ◆ 记录集对象。代表来自一个数据提供者的一组记录。例如,可以用一个记录集对 象来修改一个 SQL Sever 表中的记录。
- ◆ 域对象。代表一个记录集中的一个域。
- ◆ 命令对象。代表一个命令。例如,可以用命令对象执行一个 SQL 存储过程或带参数的查询。
- ◆ 参数对象。代表 SQL 存储过程或带参数查询中的一个参数。
- ◆ 属性对象。代表数据提供者的具体属性。
- ◆ 错误对象。代表 ADO 错误。

6.6.2 设置 ODBC 数据源

这一小节将介绍如何在 ASP 网页中使用 ADO。在前面的章节中已经讲述过服务器的 配置。本书是在 Microsoft SQL Sever 中使用 ADO。在继续下面的内容之前, Microsoft SQL Sever 必须与当前的 Web 服务器安装在同一台机器上或者网络上的另一台机器上。要了解 安装和配置 Microsoft SQL Sever 的详细内容,请参阅第3章。

在能够使用 ADO 之前,首先必须建立一个数据源。数据源包含了如何与一个数据提供者进行连接的信息。在这种情况下,你将使用数据源与 Microsoft SQL Sever 建立连接。数据源共有 3 类型:用户数据源、系统数据源和文件数据源。使用 SQL Server 作为 Web 库系统的后台数据服务,需要建立系统数据源。

注意: 在建立一个系统数据源之前, 应确保 SQL Sever 服务已启动。通过 Microsoft SQL Sever 程序组中的 SQL Server 服务管理器可以确定 SQL 服务是否在运行。

1. ODBC 概述

ODBC(Open DataBase Connectivity,开放数据库连接)是一个标准协议,允许应用程序连接到各种外部数据库服务器或文件。Microsoft Jet 数据库引擎使用的 ODBC 驱动程序 允许访问 Microsoft SQL Server 和一些其他外部数据库。

ODBC 是 Microsoft 为应用程序开发人员提供的,它采用单一的 API 来辐射各种不同的 数据库引擎、关系的和非关系的数据库管理系统(DBMS, DataBase Manager System)。ODBC API 的宗旨是给应用开发人员提供相似的功能,而不管所访问的数据类型是什么,这一目标是通过要求每个 ODBC 驱动程序必须达到某个预定义的一致性要求级别来实现的。考察 一种 ODBC 驱动程序时,这个驱动程序必须遵守核心的 ODBC 规范,这一要求保证应用开发人员可以得到相同的功能,而不管它访问的是什么数据库。如果所访问的数据库格式不能直接支持这些核心功能,ODBC 驱动程序就要模拟这些功能。

ODBC 的 API 是远离数据源的抽象层,这些调用为开发人员提供了一种工作方法,让 他们能使用 ODBC 中定义的相同 SQL 语法来操作不同的数据源。此外,连接数据源的方 法、从数据源接受错误信息的过程以及标准的登录接口,在不同的驱动程序下都是相同的。 为了达到此通用性,驱动程序的核心 API 必须满足如下的一致性要求:

(1) ODBC 核心 API 的一致性要求

- ◆ 分配和释放环境、连接及语句句柄。
- ◆ 连接至数据源,在一个连接中使用多个语句。
- ◆ 准备和执行 SQL 语句及立即执行 SQL 语句。
- ◆ 在 SQL 语句及结果列中制定参数的存储空间。
- ◆ 从结果集合中检索数据,检索与结果集合有关的信息。
- ◆ 获取出错信息。

(2) ODBC 第一级的一致性要求

- ◆ 对核心 API 功能的支持。
- ◆ 使用驱动程序有关的对话框来连接数据源。
- ◆ 设置和获取语句以及连接选项的值。
- ◆ 发送部分或所有参数值。
- ◆ 获取一个结果列值中的一部分或全部(对长数据有用)。
- ◆ 获取类别信息(如列、特殊列、统计和表)。
- ◆ 获取与驱动程序及数据源功能有关的信息,如支持的数据类型、标量函数和 ODBC 函数等等。

(3) ODBC 第二级的一致性要求

- ◆ 对核心及第一层的 API 的支持。
- ◆ 浏览连接信息并列出可用的数据源。
- ◆ 发送参数数值组,检索结果列值数组。
- ◆ 使用可移动的光标。

- ◆ 检索一个 SQL 的本地形式。
- ◆ 检索类别信息(授权、关键字和过程)。
- ◆ 调用转换 DLL。

由于每个数据库引擎都会使用略有不同的机制来访问数据,或是采用了稍加变化的 SQL语法,或是使用了一个自定义的、独有的应用程序,因此一致性要求是很重要的。

2. ODBC 数据源

ODBC 数据源是由数据源名称(Data Source Name, 缩写为 DSN)来引用的,它可以 通过 Windows 的控制面板来创建。

ODBC 数据源位于支持开放式数据库连接性协议的程序或数据库中,它包括 Microsoft SQL Server (但并不只限于此)。例如, SQL Server 数据源就是 SQL Server 数据库 (即位于服务器上的 SQL 数据库)以及访问该服务器所需的网络信息。定义 ODBC 数据源的信息 会视其 ODBC 驱动程序的要求而不同。

ODBC 桌面数据库驱动程序所需文件安装在 Windows 2000 或更低版本的 Windows 9x 的\Windows\System 目录下,或 Windows 2000 Server Family 或 Windows NT 的\Windows \System32 目录下。如果 ODBC 文件已预先安装在不同的目录之中,则要注意确保使用 \Windows\System (或 System32) 目录下的较新文件。ODBC 数据源结构如图 6-2 所示。



图 6-2 ODBC 数据源结构

3. 创建一个系统 DSN

要创建一个数据源,首先应给服务器提供数据库的名字、通信所需的 ODBC 驱动程序 以及(如果是活动数据服务器)网络中的地址。还要注意,如果要使用 ADO,那么要创建 的 DSN 必须是一个系统的 DSN 或文件的 DSN,而不是一个用户的,它们之间的不同点在 于系统 DSN 对于 NT 服务器上的所有用户都是可以访问的,包括所有的 NT 系统服务。

设置 DSN 的方法很多,可以使用 Visual InterDev 来创建一个系统 DSN,当然最常见的 还是通过 NT 系统中的 ODBC 数据源来进行设置。下面介绍如何通过 NT 系统中的 ODBC 数据源设置一个 DSN,和后台的 SQL Server 建立连接。

(1) 首先选择 NT 系统"开始"菜单"程序组"中的"管理工具"选项,然后选择其中的 ODBC 数据源,将启动数据源管理器,如图 6-3 所示。



图 6-3 数据源管理器

(2) 在图 6-3 所示的视窗中选择"系统 DSN",如图 6-4 所示。



图 6-4 系统 DSN 选项

(3) 如图 6-4 所示视窗中所显示的系统数据源都是笔者已经建立的系统数据源。要新 建一个 DSN,请单击"添加"按钮,如图 6-5 所示。



图 6-5 选择驱动程序

(4) 不同的数据库需要不同的数据源驱动程序,在这里要根据后台数据库来选择相应的驱动程序。在这里的后台数据库是 Microsoft SQL Server,因此选择 SQL Server 驱动程序。 单击"完成"按钮,继续操作。

注意:在设置数据源之前,必须确保要建立连接的 SQL Server 对象处于启动状态。

(5) 在图 6-6 所示的对话框中,需要填写数据源名称、说明和所要连接的 SQL Server 服务器。这里的数据源名称就是 DSN 名称,在 ASP 编程中所引用的 DSN 名称就是指在这 里设置的名称。对于此数据源的描述,可以输入相关的说明语句,也可以不输入任何信息。 所要连接的 SQL Server 服务器可能是本地机也可能是网络中的其他 SQL Server 服务器,通 过下拉菜单可以选择,这里选择的是(local),也就是本地机上的 SQL Server 服务器。正 确设置后,单击"下一步"。



图 6-6 与 SQL Server 建立连接

(6) 如图 6-7 所示,在登录 SQL Server 之前,必须通过 SQL Server 的身份验证。这 里提供了两种方式:一种是 Windows NT 验证,一种是 SQL Server 验证。这里,一般采用 SQL Server 验证。Windows NT 验证实际上就是以系统管理员身份登录 SQL Server,在 ASP 程序中容易泄漏系统管理员的密码,因此采用 SQL Server 验证。这里的用户名和密码是在 SQL Server 中创建的,关于 SQL Server 的用户创建参考第 3 章。

建立新的数据费到 501 Serv	(er	X
	e gestaar oo soo gestaar gestaar dhi	28.41
A DOCRES		
al de set		WHEG)
TOUR FOREN		
NE STORUT DDBI		
Concent Ter	A CONTRACTOR OF	
MICON Serve V		
	at DO! pett	
	*****	1962 -
	· · · · · · · · · · · · · · · · · · ·	
· 在19月1日中世界	清控的现在分词 网络二个	- 和助

图 6-7 SQL Server 身份验证

(7)如果输入的用户名和密码正确,将成功登录到 SQL Server 服务器,出现如图 6-8 所示的设置面板,根据其中的提示进行相应的设置,在这里最重要的是设置默认的数据库,因为一般在一个 SQL Server 服务器中都存在多个数据库,而每一个数据源只能连接到一个数据库,所以在这里必须设置默认的数据库。设置完成后,单击"下一步"。



图 6-8 登录成功

(8)出现如图 6-9 所示的数据源设置面板,根据其中的项目提示进行相应的设置。一般情况下,采用默认设置就可以了。设置完成后,单击"完成",结束数据源的设置步骤。



图 6-9 数据源设置

(9) 出现如图 6-10 所示的数据源参数报表,可以单击其中的"测试数据源"对新建 立的数据源进行测试。如图 6-11 所示。



图 6-10 数据源配置成功



图 6-11 SQL Server 数据源测试结果

通过上面的测试之后,可以确定已经成功建立了一个新的数据源。这样,就可以在前 台的 ASP 程序中通过这个数据源对后台的默认数据库进行相应操作,具体操作步骤和方法 见 6.7 节。

6.6.3 使用 ActiveX 数据对象对数据库进行存储操作

下面将举例说明如何使用 ADO 对 Microsoft SQL Sever 数据库进行存储操作。读者在做好所有的配置后,可以尝试下面的例子,从中可以看到从 ASP 网页中访问 Microsoft SQL Sever 的基本过程,从而对 ADO 有进一步的理解和认识。另一方面,可以通过这个例子检测服务器的配置是否正确。

下面这段程序把字符串"测试成功!"插入到 SQL Server 数据库的一个 test 表中, 然 后从表中取出这个数据并在浏览器中显示。

<HTML>

<HEAD><TITLE> ADO Example </TITLE></HEAD>

<BODY>

<%

Set MyConn=Server.CreateObject("ADODB.Connection")

MyConn.Open "DSN=test,uid=pett,pwd=123"

MyConn.Execute "INSERT INTO test (testcolomn) VALUES ('测试成功!')"

Set RS=MyConn.Execute("SELECT * FROM test")

Response.Write(RS("testcolomn"))

MyConn.Close

Set MyConn=nothing

%>

</BODY>

</HTML>

使用这个例子之前,首先需要在 SQL Server 数据库中建立一个名为 test 的表。数据表的创建步骤可参考第 3 章。

在这一段程序中首先创建了与数据库的连接对象。接下来调用连接对象的 open 方法, 打开了一个对数据库的连接。在 open 方法中用到了在上一小节所建立的系统数据源,以建 立对数据库的连接。

一旦打开了一个连接,就可以用这个连接执行 SQL 语句。在这个例子中,用 Execute 方法执行了两个 SQL 语句。首先用 INSERT 语句把字符串"测试成功!"输入到数据库中。 然后用 SELECT 语句从表中取出了这个字符串。SQL 语句的相关介绍也可以参考第 3 章。

如果系统配置正确,字符串"测试成功!"将会显示在浏览器窗口中。该字符串先被插 入数据库,又被取出并显示。

与数据库的所有通信都要通过一个打开的连接来进行。在对一个数据库进行数据插入 和读取之前,必须先打开与这个数据库的连接。这种打开和关闭一个连接的操作与打电话 的过程有相似之处。在能够与 SQL Sever 进行通信之前,必须先建立连接。

下面将叙述如何使用 ADO 连接对象。首先讲述如何打开和关闭一个对数据库的连接。 然后讲述如何通过一个打开的连接执行 SQL 语句。

1. 打开和关闭数据库连接

要打开与一个数据库的连接,首先需要创建一个连接对象。创建了连接对象之后,就 可以调用连接对象的 open 方法来打开一个连接。下面的程序将建立一个连接对象:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

MyConn.Open"dsn=test;uid=pett;pwd=123"

MyConn.Execute "INSERT test (testcolumn) VALUES ('测试成功!') "

MyConn.Close

Set Myconn=nothing

%>

注意: 在建立连接对象并使用 Open 方法打开一个数据库时,必须提供正确的登录 ID 和密码,这个登录 ID 和密码是在建立 ODBC 数据源时设定的。

在这个例子中, 创建了连接对象的一个实例 MyConn。然后调用连接对象的 open 方法。 接下来, 调用连接对象的 Execute 方法, 执行了一个 SQL 语句。最后关闭这个连接。

使用完一个连接后,应该关闭。这就像挂断一个电话,从而释放占用的线路。连接对 象的 Close 方法可以关闭一个连接。关闭了一个连接后,你就不能再用这个连接与数据库 进行通信。因此,依赖这个连接支持的其他对象也不能再与数据库相互通信。

2. 通过打开的连接执行 SQL 语句

通过打开的连接执行 SQL 语句要用到 Execute 方法。这个方法有两种形式:一种形式 用来从数据库中返回信息;当不需要返回信息时,则使用另一种形式。

下面的例子演示了如何使用 Execute 方法执行一个没有返回结果的 SQL 语句:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

MyConn.Execute "INSERT test (testColumn) VALUES('测试成功!') "

MyConn.Close

%>

在这个例子中,用 Execute 方法执行了一个 SQL INSERT 语句。因为没有返回结果, Execute 方法不使用括号。

可以用 Execute 方法从一个查询返回结果,如下例所示:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

Set RS=MyConn.Execute("SELECT * FROM test")

MyConn.Close

%>

在这个例子中,使用了 Execute()方法以返回一个 SQL SELECT 查询的结果。与上一个 例子不同,这个 Execute 方法使用了括号。当需要返回结果时,千万别忘了使用括号,否 则会收到错误信息 Expected end of statement。

这个 SQL 查询的结果被读取到记录集对象的一个实例 RS 中。该记录集由 Execute()方 法自动创建。

在 Web 数据库系统中,要频繁地对后台数据库进行各种操作,因此建立数据库的连接 对象在 ASP 程序中是最常用的语句。

6.7 用记录集显示记录

记录集可以用来代表表中的记录。与表一样,一个记录集包含一条或多条记录(行), 每个记录包括一个或多个域(字段)。在任何时刻,只有一条记录是当前记录。

要创建一个记录集对象,可以使用连接对象的 Execute()方法。当使用 Execute()方法从 一个数据库返回查询结果时,一个记录集对象会被自动创建。比如:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

Set RS=MyConn.Execute("SELECT * FROM test")

RS.Close

MyConn.Close

%>

在这个例子中,使用一个 SQL SELECT 语句从一个表 test 中返回了所有记录。Execute() 方法返回一个记录集。在这个脚本中,该记录集被分配给变量 RS,然后关闭了这个记录集。 最后,关闭与数据库的连接。

记录集中的每一条记录都对应表 test 中的一条记录。要显示记录集中的所有记录,只 要简单地做一个循环就可以,如下例所示:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection") MyConn.Open "Dsn=test;uid=pett;pwd=123" Set RS=MyConn.Execute(" SELECT * FROM Test") WHILE NOT RS.EOF

```
Response.Write("<BR>"&RS("testcolumn"))
```

RS.MoveNext

WEND

RS.Close

MyConn.Close

%>

在这个例子中,WHILE...WEND 循环用来扫描记录集 RS 中的每一条记录,把每个记录的 testcolumn 域输出到浏览器。这段脚本显示了表 test 中的所有记录。

当一个记录集对象中收集了数据时,当前记录总是第一条记录。在上面的例子中,调用了记录集对象的 MoveNext 方法,使当前记录移到下一条记录。当所有的记录都显示完时,记录集对象的 EOF 属性值将变为 true,从而退出 WHILE...WEND 循环。

一个记录集对象有一个域集合,包含一个或多个域对象。一个域对象代表表中的一个 特定字段。例如,在上面的脚本中,使用了表达式 RS("testcolumn")来显示字段 testcolumn。 实际上,可以通过许多途径显示一个字段的值。下面的每一个表达式都可以显示字段 testcolumn 的值:

RS("testcolumn") RS(0) RS.Fields("testcolumn") RS.Fields(0) RS.Fields.Item("testcolumn")

RS.Fields.Item(0)

注意:既可以通过字段名也可以通过顺序号来指定一个字段。例如,可以使用 RS("testcolumn")或 RS(0)来代表字段 testcolumn。两种方法起到同样的效果,因为 字段 testcolumn 对应表中的第一个字段(第一个字段的顺序号是0)。

当不知道一个记录集中的字段名时,通过顺序号指定一个域是有用的。例如,下面的 ASP 脚本显示了一个表中所有记录的所有字段:

```
<HTML>
```

```
<HEAD><TITLE>Show All Rows And Columns</TITLE></HEAD>
```

<BODY>

<%

Set MyConn=Sever.CerateObject("ADODB.Connection")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

Set RS=MyConn.Execute("SELECT * FROM test")

%>

<TABLE BORDER=1>

<TR>

```
<% FOR i=0 to RS.Fields.Count-1 %>
     <TH><% =RS(i).Name *></TH>
<% Next %>
</TR>
<% While Not RS.EOF %>
<TR>
<% FOR i=0 TO RS.Fields.Count-1 %>
    <TD><% = RS(i) %></TD>
<% Next %>
</TR>
<%
RS.MoveNext
WEND
RS.Close
MyConn.Close
%>
</TABLE>
</BODY>
</HTML>
```

在这个例子中, 域集合的 Count 属性用来返回该记录集中域的数目。Name 属性用来返回每个域的名字。两个 FOR...NEXT 循环用来对记录集中的所有字段进行操作。不论表中有多少记录和字段, 它们将被全部显示。

6.7.1 记录集游标和锁定类型

可以用 4 种类型的游标打开一个记录集。游标决定对一个记录集进行什么操作,游标 还决定了其他用户可以对一个记录集进行什么样的改变。下面列出了游标的不同类型和限 制:

- ◆ adOpenFowardOnly 使用前向游标,只能在记录集中向前移动。
- ◆ adOpenKeyset 使用 Keyset 游标,可以在记录集中向前或向后移动。如果另一个 用户删除或改变了一条记录,记录集中将反映这个变化。但是,如果另一个用户 添加了一条新记录,新记录不会出现在记录集中。
- ◆ adOpenDynamic 使用动态游标,可以在记录集中向前或向后移动。其他用户造成的记录任何变化都将在记录集中有所反映。
- ◆ adOpenStatic 使用静态游标,可以在记录集中向前或向后移动。但是,静态游标 不会对其他用户造成的记录变化有所反映。

在缺省情况下,当打开一个记录集时,将用前向游标打开它。这意味着只能用 MoveNext 方法在记录集中向前移动。对记录集的其他操作将不支持。 前向游标的好处是它比较快。无论何时,如果前向游标可以实现要求,就应该使用前 向游标。如果需要用功能更强的游标打开记录集,可以使用如下的脚本:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection") Set RS=Sever.CreateObject("ADODB.RecordSet") MyConn.Open "Dsn=test;uid=pett;pwd=123" RS.Open "SELECT * FROM Test",MyConn,adOpenDynamic RS.Close MyConn.Close

要用一种特定的游标打开记录集,必须显式地创建这个记录集,然后用该游标类型打 开它。要做到这一点,首先应创建记录集对象的一个实例。接下来,要用 Open 方法通过 一个连接和一种游标类型来打开这个记录集。在这段脚本中,用连接对象 MyConn 和一个 动态游标打开了记录集 RS。

打开记录集时,也可以指定锁定类型。当不止一个用户同时试图改变一个记录时,锁 定类型决定了数据库应如何处理。可以指定下面4种锁定类型:

- ♦ adLockReadOnly 指定不能修改记录集中的记录。
- ◆ adLockPessimistic 指定在编辑一个记录时立即锁定它。
- ♦ adLockOptimstic 指定只有调用记录集的 Update 方法时才锁定记录。
- ◆ adLockBatchOptimstic 指定记录只能成批地更新。

在缺省情况下,记录集使用只读锁定。要指定不同的锁定类型,可以在打开记录集时 包含这些锁定常量之一。这里有一个例子:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT * FROM Test", MyConn, adOpenDynamic, adLockPessimistic

RS.Close

MyConn.Close

```
%>
```

这个脚本与上一个基本相同,只是增加了锁定类型。当打开记录集 RS 时,将使用 adLockPessimistic 锁定。这意味着这个记录集中的记录可以被修改。

最后,打开一个记录集时,可以指定一个 Options 参数。Options 参数标明用来打开记 录集的命令字符串的类型。告诉 ADO 被执行字符串内容的有关信息有助于高效地执行该 命令字符串。

可以使用下面的常量作为 Options 参数:

- ♦ adCMDTable 被执行的字符串包含一个表的名字。
- ◆ adCMDText 被执行的字符串包含一个命令文本。
- ◆ adCMDStoredProc 被执行的字符串包含一个存储过程名。
- ◆ adCMDUnknown 不指定字符串的内容(这是缺省值)。

在下面的脚本中, Options 参数用来告诉 ADO, 命令字符串的内容是命令文本:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

```
Set RS=Sever.CreateObject("ADODB.RecordSet")
```

```
MyConn.Open "Dsn=test;uid=pett;pwd=123"
```

```
RS.Open "SELECT * FROM Test", MyConn, adOpenDynamic, adCMDText
```

RS.Close

MyConn.Close

%>

6.7.2 操作记录集的高级方法

到现在为止,讲述了如何用 SQL 来修改记录集中的记录。实际上,还可以使用许多记录集方法修改记录集中的记录。下面列出了每种方法的简要说明:

- ◆ AddNew 向记录集中添加一条新记录。
- ◆ CancelBatch (当记录集处在批量更新模式时)取消一批更新。
- ♦ CancelUpdate (调用 Update 之前)取消对当前记录所做的所有修改。
- ♦ Delete 从记录集中删除一条记录。
- ◆ Update 保存对当前记录所做的修改。
- ◆ UpdateBatch (当记录集处于批量更新模式时)保存对一个或多个记录的修改。

例如,使用 AddNew 方法向一个打开的记录集中添加一条空记录:

<%

```
Set MyConn=Sever.CreateObject("ADODB.Connection")
```

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT * FROM Test", MyConn, adOpenDynamic, adCMDText

RS.AddNew

RS("Testcolumn")= "A new column"

RS.Update

RS.Close

MyConn.Close

%>

在这个脚本中,用 AddNew 方法创建了一条新记录。接着,新记录的 Testcolumn 域被

赋值"A new column"。最后,调用 Update 方法保存新记录。要使用这些方法,记录集必 须以只读方式以外的其他锁定方式打开。

也可以使用 SQL INSERT 语句向一个表中添加新记录,以取代 AddNew 方法。通常, 使用 SQL 比使用上述方法要好一些,因为 SQL 更加灵活。

6.7.3 遍历记录集

记录集对象有许多在记录之间移动的方法。这些方法的大部分只有在记录集以某种特定的游标打开时才可以使用。下面列出了记录集对象的一些方法及其功能说明:

- ◆ Move NumRecords 在记录集中向前或向后移动指定数目的记录数。
- ◆ MoveFirst 移动到记录集的第一条记录。
- ◆ MoveNext 移动到记录集的下一条记录。
- ◆ MovePrevious 移动到记录集中的上一条记录。
- ◆ MoveLast 移动到记录集的最后一条记录。

记录集对象还有许多属性,它们对遍历记录集是有用的。同样,许多属性需要特定的 游标类型:

◆ AbsolutePosition 用来设置或读取当前记录在记录集中的位置顺序号。

- ◆ BOF 标明当前位置在记录集的第一条记录之前。
- ◆ EOF 标明当前位置在记录集的最后一条记录之后。
- ◆ RecordCount 表示一个记录集的记录总数。

例如,假设想在一个记录集中反向移动,那么就可以使用 MoveLast 和 MovePrevious 方法以及 BOF 属性。下面的 ASP 网页演示了该如何实现:

<HTML>

<HEAD><TITLE> Backwards Recordset </TITLE></HEAD>

<BODY>

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT * FROM Test",MyConn,adOpenStatic

RS.MoveLast

WHILE NOT RS.EOF

Response.Write("
"&RS("Testcolumn"))

WEND

RS.Close

MyConn.Close

%>

</BODY>

</HTML>

在这个例子中,用静态游标打开记录集。然后通过使用 MoveLast 和 MovePrevious 方 法显示该记录集中的所有记录,直至到达记录集的开头。BOF 属性用来检测何时到达记录 集的开头。

可以用这种方法在一个记录集中反向移动,但是很难想象有什么理由需要这样做。使用 SQL 语言自身对查询结果进行排序是有效的。如果需要,可以使用 ORDER BY 子句对记录进行排序。

6.7.4 返回记录数目

可以用记录集对象的 RecordCount 属性来确定一个记录集中的记录总数。但是,应该 慎重使用这个属性,因为这个属性在许多情况下效率极低。

对一个用前向游标打开的记录集,不能使用 RecordCount 属性。必须打开一个效率较低的游标才可以使用这个属性,如下例所示:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT Testcolumn FROM Test", MyConn, adOpenStatic

Response.Write(RS.RecrdCount)

RS.Close

MyConn.Close

这个脚本输出表 Test 中的记录数目, RecordCount 属性用来返回这个数字。为了能使用 RecordCount 属性,记录集用一个静态游标打开。

通常,需要一个记录数目的惟一原因是想确定是否至少有一条记录满足特定的条件。 例如,想检查一下是否一个人输入了合法的口令。在这种情况下,可以查询一个保存口令 的表,用 RecordCount 属性确定是否有口令存在。如果 RecordCount 大于 0,说明有口令存 在,否则说明没有,即该口令是非法的。

但是,用 EOF 属性检测一个查询是否返回了结果将会更好。当用前向游标打开了一个 记录集时,可以使用 EOF 属性。这里有一个例子:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT * FROM Password_Table

^{%&}gt;

WHERE Password="&Request.Form("Password"),Myconn

IF RS.EOF THEN

Response.Write("The password you entered is invalid. ")

ELSE

Response. Write("Welcome to our Web site! ")

END IF

RS.Close

MyConn.Close

%>

在这个例子中,用 EOF 属性检测一个查询是否有返回结果。如果 EOF 属性是 true,说明用户输入的口令不在保存口令的表中。

在某些情况下,的确需要返回记录数目。例如,也许想显示在站点上注册的用户总数。 但是,同样应该避免使用 RecordCount 属性。取而代之,可以执行一个 SQL COUNT (*) 查询,如下例所示:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT COUNT(*) Mycount FORM Password_Table", MyConn

%>

There are <% =RS("Mycount")%> registered users at the Web site.

<%

RS.Close

MyConn.Close

%>

请注意这个查询中如何使用字段别名 Mycount。通过给集合函数 COUNT(*)提供一个名 字,可以在输出查询结果时使用这个名字。

6.7.5 对记录集进行分页

比如在一个 BBS 论坛系统中,需要显示出某一讨论区中的所有文章,但是有很多文章。 在这种情况下,不希望把所有的文章显示在同一个 ASP 页中。这时允许用户进行翻页是一 个不错的办法。

记录集对象有三个属性用于此目的。可以使用这些属性把一个记录集中的记录分成许 多逻辑页。通过把记录集中的记录进行分页,可以一次只显示记录集中的一部分。下面列 出了这些属性:

♦ AbsolutePage 指定当前的页。

- ♦ PageCount 返回记录集中的逻辑页数。
- ◆ PageSize 指定一个逻辑页中的记录个数,缺省值是10。

要把一个记录集分成多个页,首先应使用 PageSize 属性指定一页中的记录个数,然后 就可以用 AbsolutePage 属性移动到一个特定的页。最后,PageCount 属性可以用来反回总 页数。相应的代码如下:

```
<HTML>
```

```
<HEAD><TITLE> Recordset With Pages </TITLE></HEAD>
<BODY>
<%
'指定当前的页码
IF Request.QueryString("MOVE")="NEXT" THEN
 Session("CurrentPage")=Session("CurrentPage")+1
END IF
IF Request.QueryString("MOVE")="PREV" THEN
 Session("CurrentPage")=Session("CurrentPage")-1
END IF
IF Session("CurrentPage")="" THEN
 Session("CurrentPage")=1
END IF
%>
<H1>Current Page: <%=Session("CurrentPage")%></H1>
<HR>
<%
'打开一个连接和记录集
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB.RecordSet")
MyConn.Open "Dsn=test;uid=pett;pwd=123"
RS.Open "SELECT title FROM message", MyConn,adOpenStatic
' 定义每一页中的文章数
RS.PageSize=5
'定义当前的页码
RS.AbsolutePage=Session("CurrentPage")
'显示当前页面中的记录
WHILE NOT RS.EOF AND NumRows<RS.PageSize
%>
<BR>Product Name: <%=RS("title")%>
<%
RS.MoveNext
```

```
NumRows=NumRows+1
WEND
%>
<HR>
<% IF Session("CurrentPage")>1 THEN %>
<A HREF="pages.asp?MOVE=PREV"> [PREV] </A>
<% END IF %>
<% IF Session("CurrentPage")<RS.PageCount THEN %>
<A HREF="pages.asp?MOVE=NEXT"> [NEXT] </A>
<% END IF %>
<%
RS.Close
MyConn.Close
%>
</BODY>
</HTML>
```

这个 ASP 网页一次显示表 message 中的 5 条记录,一个 session 变量 CurrentPage 被用 来跟踪记录集的当前页。当用户单击 NEXT 时,就显示下一页记录,当用户单击 PREV 时,就显示上一页。

6.7.6 把记录取到数组中

在特定情况下,需要把记录集中的记录取到数组中。例如,如果需要修改一个记录集中的数据,但不想改变记录集中的记录本身,这时就可以先把记录取到数组中。

要把一个记录集中的记录分配给一个数组,应使用记录集对象的 GetRows()方法。这 里有一个例子:

<%

Set MyConn=Sever.CreateObject("ADODB.Connection")

Set RS=Sever.CreateObject("ADODB.RecordSet")

MyConn.Open "Dsn=test;uid=pett;pwd=123"

RS.Open "SELECT MyFirstCol,MySecondCol FORM Test",MyConn

MyArray=RS.GetRows()

RS.Close

MyConn.Close

%>

在这个脚本中,记录集 RS 包含的所有记录被分配给数组 MyArray。这个数组由 GetRows()方法自动创建并添入数据。GetRows()方法创建了一个二维数组,第 1 个下标代 表字段数,第 2 个下标代表记录数。

下面的脚本可以用来显示数组 MyArray 中的内容:

```
<%
```

```
FOR i=0 TO UBOUND(MyArray,2)
%>
<BR> First Column:<%=MyArray(0,i)%>
<BR> Second column:<%=MyArray(1,i)%>
<%
```

NEXT

```
%>
```

VBScript 函数 UBOUND()用来确定数组第二维的维数。FOR...NEXT 循环用来扫描数组的所有元素。这个数组代表一个有两个字段的记录集。当该数组的第一个下标值是 0 时,代表第一个字段,当第一个下标值是 1 时,代表第二个字段。

6.7.7 指定记录集的最大容量

假设想显示在站点上发布的最后 10 条信息,但是不能超过 10 条信息;或者想显示 3 个连接,但是不超过 3 个。如何限制读取到一个记录集中的记录数?记录集对象有一个属 性用于此目的。通过使用 MaxRecords 属性,可以限制从一个数据库查询返回到一个记录集 中的记录数目。如下例所示:

<%

```
Set MyConn=Sever.CreateObject("ADODB.Connection")
```

Set RS=Sever.CreateObject("ADODB.RecordSet")

```
MyConn.Open "Dsn=test;uid=pett;pwd=123"
```

RS.MaxRecords=10

RS.Open "SELECT Testcolumn FROM Test", MyConn

WHILE NOT RS.EOF

Response.Write("
"&RS("Testcolumn"))

RS.MoveNext

WEND

RS.Close

MyConn.Close

%>

在这个脚本中, MaxRecords 属性被设为 10。当记录集 RS 被打开时,不会有超过 10 个记录被取到 RS 中。即使表中有 10 000 条记录,也只有 10 条记录被取出。使用 MaxRecords 属性时,必须在打开记录集之前设置该属性值。记录集打开之后,该属性将成为只读的。

6.8 本章小结

本章介绍了 ASP 的几个主要对象,并重点讨论了记录集对象。叙述了如何在 ASP 网页中使用记录集对象显示表中的记录,还叙述了如何用不同类型的游标和锁定打开记录集。 使用这些对象,可以完成对数据库的绝大部分操作。

第7章 BBS论坛系统

本章将详细介绍 BBS 论坛系统,此论坛的功能在第2章已经介绍过了。

7.1 登录系统

每一个 BBS 论坛系统首先需要一个登录系统,以确认用户的合法身份。一般要求登录 用户输入用户名和密码,然后根据数据库中的用户资料表进行身份验证。如果数据库中存 在此用户且密码正确,则通过验证;否则登录失败,要求用户重新输入。

登录系统中主要包括3个模块:

◆ 用户身份验证模块

- ◆ 用户注册模块
- ◆ 密码提示模块

具体流程可以参考图 2-35。

7.1.1 用户身份验证

用户身份验证模块是登录系统中最重要的模块,其中包括两个文件: Login.htm 和 Choice.asp。在 Login.htm 中要求用户输入用户名和密码,然后提交到 Choice.asp 对用户输 入的信息进行处理。处理的过程就是打开后台数据库,从用户资料表中进行查询,验证当 前输入的信息在用户资料中是否存在。

1. Login.htm

Login.htm 是 BBS 论坛用户的主登录界面,只有通过此通道才能正确登录到 BBS 论坛中。登录界面在浏览器中的效果可以参照第2章中的介绍。此文件并不是一个 ASP 文件,只是一个普通的 HTML 文件,但在登录系统中是不可缺少的。在此文件中提供两个输入表单,要求用户输入用户名和密码,还提供了另外两个模块的链接。

<html>

<head>

<title>登录界面</title>

为了减轻服务器端的负荷,充分利用客户端的处理能力,在把一些客户端用户提交的 信息向服务器传递之前一般先要进行验证。这样不但可以校验客户输入的信息,防止出现 一些由于粗心造成的错误,而且也可以对用户进行一定的限制,使不符合条件的信息不会 发送到服务器端,而是把错误限制在了客户端,提高了相应速度,节省了客户的网上传输 量,减轻了服务器端的工作量。同时,也可以提高用户访问的效率,随时对用户的错误进 行提醒。因此,对于需要用户输入信息的页面往往进行客户信息验证。下面就是一段信息 验证程序: <script language="JavaScript"> ' 在客户端对客户信息进行提交前的验证 function getstring(str)' 定义函数 getstring() { rstr="": for(i=0;i<str.length;i++) {if(str.charAt(i)!="") rstr=rstr+str.charAt(i); } return rstr; } function submitclick()' 定义函数 submitclick() Ł cnt=0; titlestr="[请注意]\n\n";' 定义字符串 titlestr errstr="您填写用户名和密码的时候: \n\n":' 定义字符串 errstr tmpzcm=getstring(document.input.lybid.value); ' 获取表单中 lybid 的输入值 if(tmpzcm=="")'判断 lybid 的输入值是否为空,如果为空则执行下面的语句块 {cnt=cnt+1; errstr=errstr+cnt+"."+"请输入您的姓名\n";'在 errstr 字符串中添加一条语句 } else'如果不为空 document.input.lybid.value=tmpzcm;' 重新赋值 tmply=getstring(document.input.password.value); ' 获取表单中 password 的输入值 if(tmply=="")'判断 password 的输入值是否为空,如果是则执行下面的语句块 {cnt=cnt+1; errstr=errstr+cnt+"."+"请输入您的口令\n";'在 errstr 字符串中添加一条语句 } else'如果不为空 document.input.password.value=tmply;'重新赋值 if(cnt==0) return true; '如果 cnt=0,则两个表单的值都不为空,返回 true else '如果不满足条件 {window.alert(errstr); '弹出警告窗口, 窗口中显示 errstr 字符串 return false; ' 返回 false }

}

</script>' JavaScript 语句结束

这是一段使用 JavaScript 编写的客户端脚本程序。尽管 VBScript 是 Active Server Page 的默认脚本语言,并且服务器端大多数也都采用 VBScript,但另一方面却存在麻烦。这是 因为,VBScript 是 Microsoft 推出的产品,IE3.0 以上的版本都支持 VBScript 的脚本,但 Netscape 对 VBScript 并不兼容,而 Netscape 在当前浏览器市场上还占有一定的份额,也有 一部分用户在使用 Netscape 浏览器,如果使用 VBScript 来编写这样的客户端信息验证程序, Netscape 将忽略这些语句,从而起不到验证作用。为了考虑更多的用户群体,只能采用 IE 和 Netscape 两种浏览器都支持的脚本语言 JavaScript 来编写这样的客户端程序。

这是一段非常简单的程序,其中定义了两个函数,对用户输入的信息进行提交前的验证。程序的运行过程是这样的:用户填写好表单后,点击"提交"按钮,将激发函数 submitclick,运行此函数,然后顺序读取表单中的各个输入值,判断输入值是否为空或者 是否满足设定的某种条件,满足的话返回 true,否则返回 false,并且添加提示信息,最后 如果所有条件都满足的话,就允许将变量提交到下一个处理页面,否则弹出显示提示信息 的警告框,提醒用户改正输入的信息。

</head>

<body bgcolor="#ffffff0">

<form method="POST" action="choice.asp" name="input">

欢迎光临 pett-BBS 论坛

<div align="center">

<tableborder="0"cellpadding="5"cellspacing="5"width="80%">

用户名

<palign="left">

<input name="lybid" style="font-size: 9pt; border: 1solid#800080" size="20">

<fontstyle="font-size:14px">

密码

```
<palign="left">
<input name="password" style="font-size: 9pt; border: 1solid#800080" size="20" type="password">
<input type="submit" value="登录" name="B3" OnClick="returnsubmitclick();" style="background-color:
 #800080;color: #FFFF00; font-family:楷体 GB2312; font-size: 12pt;font-weight: bold;border-style:
 ridge; border-color:#FFFF00">
<inputtype="reset"value="重写"name="B4"style="background-color:#800080;color:#FFFF00;font-family:
 楷体 GB2312;font-size:12pt;font-weight:bold;border-style:ridge;border-color:#FFFF00">
</div>
</form>
<palign="center"><ahref="password.htm">密码提示</a>
<palign="center">~ahref="regist.htm">新用户注册</a>~/p>
</body>
</html>
```

上面这段是纯粹的 HTML 代码,熟悉 HTML 语言的读者不难看懂,这里不再介绍每一句代码的具体含义。其中主要使用了 HTML 中的 Form (表单)和 Table (表格),最重要的代码就是加粗的部分,采用 Post 方式把表单中的输入值发送到 Choice.asp。另外,这 里要注意 Form 中 Input 的属性值,为了能在浏览器中得到更理想的效果,需要设置一定的 属性值,具体参数不再赘述,可以参考有关 HTML 语言的书籍。

2. Choice.asp

这个文件是保证系统安全的最重要文件。每一位用户的登录信息都发送到这个页面来处理。和前面介绍的 Login.htm 文件不同,该文件中加入了大量的 VBScript 代码来完成一定的要求的功能。和编写所有的程序一样,在编写 Choice.asp 文件之前,也需要先绘制程序流程图,这样有利于理清思路,提高工作效率。Choice.asp 中的程序流程如图 7-1 所示。

首先考虑进入这个页面的所有可能用户:

- (1) 直接进入 Choice.asp 的使用者。
- (2) 从其他选段返回 Choice.asp 的用户。
- (3) 提交了错误信息的用户。
- (4) 提交了正确信息的用户。



图 7-1 Choice.asp 程序流程

可以看出,其中第(1)和第(3)类用户都属于非法用户,必须有防止这样的用户登录的程序段。对于直接进入 Choice.asp 的用户,没有任何表单信息和 Session 信息,在本程序所包含的 protect.inc 安全程序中将会被自动引导到登录界面;对于从其他选段返回 Choice.asp 的用户,由于有 Session 对象跟踪,因此可以通过安全程序的验证;对于提交了错误信息的用户(也就是在数据库中没有找到相应的用户资料),也将被自动引导到登录界面;对于提交了正确信息的用户,则能够通过数据库的验证,将显示出主菜单,成功登录,同时在数据库中记录下当前用户的来访时间、来访 IP 以及累计来访次数。

本 BBS 论坛系统中,在站内是通过 Session 对象来完成对用户的跟踪的,关于 Session 对象的使用方法在第 6 章已经介绍过。从程序的角度来看,进入 Choice.asp 的输入有以下 四种情况:

Request.form="" and Session("user")= "" Request.form="" and Session("user")<> "" Request.form<>"" and Session("user")= "" Request.form<>"" and Session("user")<> ""

注意: Request.form 是指 Login.htm 中的用户名和密码两个表单的值。Session("user") 是指对用户进行跟踪的对象参数。

根据程序流程编写的程序代码如下:

<!--#INCLUDE VIRTUAL=" protect.inc" -->

这里的 protect.inc 是一个安全程序,也可以称为密码保护程序文件,它和当前文件在

同一路径下,其中包含一定的 VBScript 代码,在执行过程中插入当前位置。

密码保护文件被包含在需要进行密码保护的每一个网页中。它检查 session 变量中的用 户名和密码是否存在,或者完成一定的操作。如果不存在,就自动引导用户到主登录界面, 要求用户输入名字和密码。

这样的密码保护文件具有公共使用性,在所有需要密码保护的文件中都可以包含,可以避免相同代码的重复。关于 protect.inc 文件的源代码见 7.7 节 "密码保护程序"。

<html>

<head>

<title>选择菜单</title>

</head>

<body bgcolor="#fffff0">

<%

Set Conn=Server.CreateObject("ADODB.Connection") ' 创建数据库连接对象

Conn.open"dsn=bbs;uid=pett;pwd=123"

- '通过 ODBC 数据源打开数据库,这里的 dsn 是指数据源名称,uid 和 pwd 则分别是访问数据库的用 户名和密码。(关于 ODBC 数据源的设置,可以参考前面 ASP 中有关 ADO 对象的介绍。)
 - if request.form("lybid") <> "" then '判断表单值是否为空,不为空则执行语句

dim aname, apassword ' 定义两个变量

- aname=replace(trim(request.form("lybid")),""","@") '进行用户名字符代换
- apassword=replace(trim(request.form("password")),""","@")
- '进行用户名、密码处理

sq="select * from member where lybid=""&aname&" and password=""&apassword&""

'进行查询操作的 SQL 语句,从用户资料表 member 中查找和输入值相同的记录。

set rs=Conn.execute(sq) '执行上面的 SQL 语句

%>

这里的字符代换语句是必不可少的,可以防止 SQL 语句的漏洞。这样的漏洞在 ASP 初学者的编程中经常出现,对于熟悉 ASP 程序和 SQL 语句的用户,也可能利用这样的程 序漏洞非法入侵。

这里使用的是 VBScript 中的 replace 字符串函数,把输入表单中的用户名和密码中的 单引号(')替换为符号@。这是因为,单引号(')是 SQL 语句中字符串的标识符,在成对的 单引号内的任何字符,数据库引擎都可能认为是字符串,包括 SQL 语句中的命令。上面代 码中的 SQL 语句是:

select * from member where lybid=""&aname&" and password=""&apassword&"

其中的 aname 和 apassword 都是定义的 VBScript 变量,程序在执行时这里会自动加入 变量的值。假如一位合法用户输入这样的信息:

用户名: pett

密 码: 123

则相应的 SQL 语句就是:

select * from member where lybid='pett' and password='123'

这一 SQL 语句在数据库中可以正确执行,查询的结果就是用户名(lybid)和密码 (password)分别为 pett 和 123 的记录,任何一项不满足都不会返回正确结果,从而起到 了验证用户身份的作用。

如果非法用户在输入中加有单引号的话,查询的结果就不是这样了。假如输入如下的 信息:

用户名: a'or 1=1 or

密码:='

如果不进行符号替换的话,相应的 SQL 语句就是:

select * from member where lybid='a' or 1=1 or 'and password='="

可以看出,这样的 SQL 语句在语法上是完全正确的,也可以在数据库引擎中正确执行, 由于人为加入了一个"或"条件(or 1=1),而这一条件永远是成立的,这样就屏蔽了用户 名和密码的验证作用,将顺利通过程序的验证,成功登录到 BBS 论坛系统。这样的漏洞如 果被别有用心的用户发现,往往可能带来不可挽回的损失。

另一方面,这样的漏洞并不可怕,只要知道就可以堵住。解决问题的关键就在于单引 号,要通过这个步骤,必须在文本框中输入单引号,如果使用替换语句用其他的任意符号 来代替,就可以完全堵住这个漏洞。

注意:从这一点可以看出,要想保护 ASP 程序的安全并不是一件容易的事情,首先需要开发者本身反复验证自己的程序,从中找出漏洞,并及时加以解决。否则 等系统正式发布到网络上,可能就晚了。这一漏洞也是笔者在实际编程过程中发现的。

<% if rs.EOF then ' 判断查询结果是否为空,如果为空执行下面的语句 response.redirect"login.htm" ' 自动返回到主登录界面 Login.htm end if%>

<% session("pwd")=request.form("password")

session("user")=request.form("lybid")%> ' 给 Session("user")赋值

这里的 session 变量没有定义变量的过期时间,而是采用默认的时间长度,也就是 20 分钟。在 20 分钟内,用户没有任何请求的话,将自动过期,禁止访问其他的页面。

<% sq1="update member setvtimes=vtimes+1,intime=now(),

inip=""&request.servervariables("remote_addr")&" where lybid=""&aname&""

'SQL 语句,增加来访次数,记录本次来访时间和 IP 地址。

```
conn.execute(sq1)%>' 执行 SQL 语句
```

```
<% set rs=nothing '关闭 rs 对象
```

end if %>

```
<% sql="select vtimes from member where lybid="&session("user")&"""</p>
```

set vrs=conn.execute(sql)%>

'获取当前用户的来访次数

font face="方正舒体">font color="#CCCC80">PETT-BBS 论坛-

```
</font><font face="楷体_GB2312" size="5" color="#0000FF">主菜单</font>
```

欢迎您,亲爱的

```
<%=session("user")%>,这是您第<%=vrs("vtimes")%>次访问本站</font>
```

'显示当前用户的来访次数

```
<% set vrs=nothing%>
```

```
<% lsql="select count(receiver) as lnum from letter where receiver=""&aname&" and read=0"%>
'统计当前用户的新信件数目
```

<% set rst=conn.execute(lsql)%>

```
<div align="center">
```

<center>

```
<font style="font-size: 14px" face="华文行楷" size="4"
```

```
color="#FF00FF">主菜单</font>
```

```
<font style="font-size: 14px" color="#FF00FF" size="4" face="华文行楷">
简介</font>
```

```
<a href="site.asp"><font style="font-size: 20px" face="楷体
GB2312" color="#000000" size="4">本站精华</font></a>
```

 在这里介绍本站的统计情况,本日十大话题,访问人次和注册人数等等。

<font style="font-size: 20px"

```
color="#000000" face="楷体_GB2312" size="4">分类讨论区</font></a>
```

 在这里是本站的各个分类讨论区,您可以阅读和发表文章!

```
</tr>
```

```
<font style="font-size: 20px" color="#000000" face="楷体
_GB2312" size="4"><a href="modify.asp">修改个人资料</a></font>
<font style="font-size: 14px" color="#000000">
这里您可以修改自己的个人资料! </font>
vidth="45%" align="center"><a href="queryfriend.asp"><font style="font-size: 20px"
color="#000000" face="楷体_GB2312" size="4">查询网友</font></a>
width="55%" align="center" valign="middle"><font style="font-size: 20px"
color="#000000" face="楷体_GB2312" size="4">查询网友</font></a>
width="color="#000000">
color="#000000" face="楷体_GB2312" size="4">查询网友</font></a>
```

个人信箱

<%if rst("lnum")<>0 then%>

```
 <font style="font-size: 20px" color="#FF0000" face="楷体_GB2312" size="4">(有
```

```
<%=rst("lnum")%>封新邮件) </font>
```

<% end if %>

```
<%set rst=nothing%>
```

'如果新邮件数目不为零的话,显示数目

 这是您的个人信箱,在这里您可以阅读别的用户给你寄来的信件,也可以给其他的用户发信!

聊天室

 这里是本站的聊天室,在这里您可以畅所欲言,结识更多的朋友!

离开本论坛

点击这里将退出 本论坛!

```
</center>
</div>
<% Conn.close
set Conn=nothing '关闭数据库连接
%>
</body>
</html>
3. protect.inc 文件
```

这是上面 Choice.asp 中所包含的密码保护程序,其 VBScript 代码如下:

```
<% if session("user")="" and request.form("lybid")="" then
```

response.redirect"login.htm"

end if%>

可以看出,程序中的代码很简单,在这里主要是向读者说明这样包含文件的使用方法。 本论坛系统中,大部分页面只有合法用户才能浏览,这样页面就需要密码保护,使用同一 个包含文件就可以了,见 7.7 节 "密码保护程序"。

以上介绍的这 3 个文件组成了 BBS 论坛系统的用户身份验证系统,最重要的文件是 Choice.asp 文件,其中采用严密的防范措施来维护论坛系统的安全,同时也保证了论坛系 统中的合法用户。

7.1.2 用户注册系统

最新登录 BBS 论坛的用户当然无法正确登录,因为在后台数据库中尚不存在任何资料, 无法提供用户名和密码。为了能让更多的用户使用,需要提供新用户的注册功能。本模块 就是完成这一功能的,包括两个文件: regist.htm 和 finish.asp。

1. regist.htm

这个文件和前面的登录文件 login.htm 在构造上十分相似。由 JavaScript 限制程序和 Form 表单组成。不同的是, regist.htm 中提供了大量的表单, 需要用户输入相应的信息。 有网上注册新用户经历的读者对这样的页面应该很熟悉。下面将给出 regist.htm 文件中 JavaScript 限制程序的源代码。不再给出纯粹的 HTML 代码。

```
<script language="JavaScript">
function getstring(str)
{
    rstr="";
    for(i=0;i<str.length;i++)
    {        if(str.charAt(i)!="")
        rstr=rstr+str.charAt(i);
```
```
}
  return rstr;
}
function submitclick()
{
 cnt=0;
  titlestr="[请注意] \n\n";
  errstr="您填写个人资料的时候: \n\n";
  tmpzcm=getstring(document.input.lybid.value);
  if(tmpzcm="")
  { cnt=cnt+1;
    errstr=errstr+cnt+"."+"请输入您的代号\n":
  }
  else document.input.lybid.value=tmpzcm;
     tmply=getstring(document.input.lybname.value);
  if(tmply=="")
  { cnt=cnt+1;
    errstr=errstr+cnt+"."+"请输入您的真实姓名\n";
  }
  else document.input.lybname.value=tmply;
  tm1=getstring(document.input.password.value);
  if(tm1="")
  { cnt=cnt+1;
    errstr=errstr+cnt+"."+"请输入您的密码\n";
  }
  else document.input.password.value=tm1;
  tm2=getstring(document.input.spassword.value);
  if(tm2=="")
  { cnt=cnt+1;
    errstr=errstr+cnt+"."+"请输入密码验证\n";
  }
  else document.input.spassword.value=tm2;
      if(tm1!=tm2)
  {cnt=cnt+1
   errstr=errstr+cnt+"."+"两次输入的密码不相同\n";
   }
  tm3=getstring(document.input.question.value);
  if(tm3=="")
  { cnt=cnt+1;
```

```
errstr=errstr+cnt+"."+"请输入提示问题,以方便您找回丢失的密码\n";
}
else document.input.question.value=tm3;
    tm4=getstring(document.input.answer.value);
if(tm4=="")
{ cnt=cnt+1;
    errstr=errstr+cnt+"."+"请输入提示答案\n";
}
else document.input.answer.value=tm4;
if(cnt==0)return true;
else
{ window.alert(errstr);
    return false;
}
```

```
}
```

```
</script>
```

可以看出,这一段程序和 Login.htm 中的信息验证程序十分相近,只是相应表单域的 名称不同。还有一点就是这里加入了密码判断,验证用户两次输入的密码是否相同。另外, 在填写用户资料的时候,需要用户注意一些相关的问题都应该在这里说明,当然这一点对 于程序开发者来说并不是至关重要的。

注册用户在这一页面中填写好相应的信息,通过限制程序后,将把所有的信息以 Post 方式发送到 finish.asp 文件来处理,并存储到后台数据表中。

2. finish.asp

这个程序很简单,首先判断提交的用户名在当前用户资料表中是否存在,因为用户名 是不同登录用户在论坛系统中的唯一标识,不允许重复,如果雷同,则需要用户重新选择 用户名。然后把接受到的所有表单信息写入后台的用户资料表中就可以了,使用一条 Insert 语句就可以完成。相应的源代码如下:

```
<%if request.form("lybid")="" then
```

```
response.redirect"regist.htm"
```

end if%>

```
'判断是否有信息提交,如果没有的话,将自动引导回注册界面。
```

```
<html>
```

<head>

<title>注册成功</title>

</head>

<body bgcolor="#fffff0">

<% Set myconn=Server.CreateObject("ADODB.connection")

```
myconn.open"dsn=bbs;uid=pett;pwd=123"
```

'建立数据库连接并打开

sql="select * from member where lybid="&request.form("lybid")&""

set rs=myconn.execute(sql)

'查询是否有相应的用户名

If rs.EOF then '判断语句,如果没有雷同,执行下面的语句

insq="insert into member

(lybid,lybname,nickname,password,birthday,question,answer,address,workunit,telephone,bpnumber,emai l,"__

&"icq,other,pen,addip,vtimes)"_

&" values

("%request.form("lybid")&"',"%request.form("lybname")&"',"%request.form("nickname")&"',"%request.form("password")&"',"

&"""&request.form("birthday")&"',"&request.form("question")&"',"&request.form("answer")&"',"&req uest.form("address")&"',"

&"""&request.form("workunit")&"',"&request.form("telephone")&"',"&request.form("bpnumber")&"'," &request.form("email")&"',"

&"""&request.form("icq")&"',"&request.form("other")&"',"&request.form("pen")&"',"&request.serverva riables("remote_addr")&"',1)"

'添加新用户记录的 SQL 语句

注意:这一语句虽然很长,但结构很简单。这里需要注意的是 ASP 中用来表示换 行的下划线(-),当一个字符串很长,在一行中无法放置的时候,可以拆为几个 字符串,然后用字符串连接符&连接起来,但换行时需要使用下划线来标识。

myconn.execute(insq) '执行上面的 SQL 语句

%>

PETT-BBS 论坛--

注册成功

您的个人资料已经安全输入到数 据库中!

<% else%> '如果已经存在相同的用户名,则执行下面的语句

对不起,此账号已经有人 使用!

请另外选择账号

'提示用户名已经存在

<% end if%>

<% rs.Close

set myconn=nothing

%> '关闭数据库连接

</body>

</html>

可以想象这个程序的执行过程,首先打开数据库连接,进行用户名判断,然后执行插 入新纪录的操作。后面的许多程序结构上都十分相似。

7.1.3 密码提示系统

一个功能完善的登录系统一般都提供这项功能,个别用户可能会丢失密码,通过密码 提示系统可以得到系统的密码提示。在用户注册时,都需要用户输入密码提示问题和答案, 这些就是为了给用户自己提供方便。当用户丢失密码时,只需要输入用户名,并正确回答 提示问题,系统就会自动显示出用户的密码。

密码提示系统共有三个文件: password.htm、password1.asp 和 password2.asp。下面分别进行介绍。

1. password.htm

这也是一个 HTML 文件,需要输入用户名,并提交到 password1.asp 中进行相应的处理。文件的源代码很简单,仅提供一个表单用于输入用户名,同样也需要一个 JavaScript 的限制程序。源代码这里不再列出。读者可以根据第 2 章中的功能介绍来自己编写相应的代码。

2. password1.asp

password1.asp 用来接受 password.htm 提交的用户名,并根据提交的用户名从数据表中 查询出当前用户的密码提示问题,显示出一个回答问题的表单框。相应的源代码如下:

```
<%if request.form("lybid")="" then
```

response.redirect"password.htm"

end if%>

'判断提交的表单值是否为空,如果为空,则自动引导回前一页面中。

<html>

<head>

<title>密码提示</title>

注意:这里需要插入一个 JavaScript 的限制程序,由于和前面提到的限制程序十分相似,这里省略了源代码。读者可以参考前面的限制程序来编写这里的限制程序。

</head> <body bgcolor="#fffff0"> PETT-BBS 论坛--

回答问题

<form method="POST" action="password2.asp" name="input">

<div align="center">

<% Set myconn=Server.CreateObject("ADODB.connection")</p>

myconn.open"dsn=bbs;uid=pett;pwd=123"%>

<%sql="select * from member where lybid="%request.form("lybid")&""

set rs=myconn.execute(sql)

'建立数据库连接,查询当前用户的记录

session("user")=request.form("lybid")

if rs.eof then%> '判断输入的用户名是否存在,如不存在,则执行下面的语句

很抱歉! 你输入的

```
<%=session("user")%>不存在! </font>
```

<%else%>

您设定的问题是:

<%=rs("question")%>

' 显示密码提示问题

<center>

请输入问题的答案</center>

<input name="answer" style="font-size: 9pt;border: 1 solid #800080"</pre>

size="20">

<center>

<input type="submit" value="提交" name="B3" OnClick="return submitclick();"

style="background-color: #800080; color: #FFFF00; font-family: 楷体_GB2312; font-size: 12pt;

font-weight: bold; border-style: ridge; border-color: #FFFF00">

<input type="reset" value="全部重写" name="B4" style="background-color: #800080; color: #FFFF00;

font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-style: ridge; border-color: #FFFF00">

</center>

</div>

</form>

'提供回答问题的文本输入框

<%end if%>

<% myconn.close

set rs=nothing%>

' 关闭数据库连接

新用户注册

</body>

</html>

3. password2.asp

password2.asp 比较 password1.asp 中提交问题的答案和用户资料表的正确答案,判断二 者是否吻合,如果吻合则显示出当前用户的密码,否则提示答案不正确。相应的程序代码 如下:

```
<%if request.form("answer")="" or session("user")="" then
```

response.redirect"password.htm"

end if%>

'判断接受的表单值和 session 变量是否为空

<html>

<head>

```
<title>密码提示</title>
```

</head>

<body bgcolor="#fffff0">

<div align="center">

<% a=session("user")%>

<%Set myconn=Server.CreateObject("ADODB.connection")

```
myconn.open"dsn=bbs;uid=pett;pwd=123"%>
```

<%sql="select * from member where lybid=""&a&"""

set rs=myconn.execute(sql)

b=rs("answer")

'和数据库建立连接,并查询出当前用户存在于数据表中的正确答案

c=request.form("answer")

if b<>c then%> '判断提交的答案是否正确,如果不正确则显示相应信息

很抱歉! 你输入的答案不符, 请重试! <%else%> '如果正确,则查询出密码

PETT-BBS 论坛-

密码提示

<%=session("user")%>,您的

密码是: <%=rs("password")%>

'显示出当前用户的密码

请牢记!

</div>

<%end if%>

<% myconn.close

set rs=nothing%>

'关闭数据库连接

重新登录

</body>

</html>

7.2 本站精华 site.asp

在前面介绍的 Choice.asp 文件中,当合法用户登录时,将通过身份验证,成功登录到 BBS 论坛系统,显示出论坛的主菜单。具体显示效果请参考第2章。其中第一个子菜单就 是"本站精华",这个菜单的功能是通过一个文件来完成的,它对论坛系统中的各类数据进 行详细统计,最重要的是从所有讨论区中查询出当前最热门的十大话题,也就是讨论的文 章数最多的十个话题。相应的源代码如下:

<!--#INCLUDE VIRTUAL="pws.inc" -->

这里所包含的是 BBS 论坛的安全保护文件 pws.inc,这个文件在 7.7 节中将具体介绍。 < html>

<head>

<title>选择菜单</title>

</head>

<body bgcolor="#fffff0">

PETT-BBS 论坛--

本站精华

本站统计

<%

Set Conn=Server.CreateObject("ADODB.Connection")

Conn.open"dsn=bbs;uid=pett;pwd=123"

'建立数据库连接对象,打开数据库

sq="SELECT count(lybid) as love1,sum(vtimes) as love2,int(sum(vtimes)/count(lybid)) as love3,"

sq2="int(sum(vtimes)/DateDiff('d','2000-8-24',date())) as love4 from member"

```
sq=sq&sq2
```

' 定义 SQL 语句, 查询出总的注册人数、来访人次、人均来访次数和日均来访次数

注意: 这是一个比较复杂的 SQL 语句, 详细解释请参考第 3 章。

```
set rs=Conn.execute(sq)%>
```

```
<center>
```

本站目前共有 <%=rs("love1")%>人注册

共有

<%=rs("love2")%>人次访问本站

人均访问次数: <%=rs("love3")%>

```
<font color="#ff0000" face="楷体_2312">日平均访问量:
<%=rs("love4")%></font>
```

'显示出相应的统计数据

```
</center>
```

<% set rs=nothing%>

```
<% sql="select top 10 * from (select count(åddid) as tnum,title,boardtype from message group by title,boardtype) order by tnum DESC"
```

'从文章表 message 中查询出当前的十大热门话题

注意: 这一条 SQL 语句在第 3 章中也有详细解释。

set rs=conn.execute(sql)%>

```
<font face="方正舒体" color="#FF0000" size="5">当日十大热门话题</font>
<center>
```

```
<font face="宋体" color="#ff0000" size="3">主题: </font>
```

讨论区

文章数目 <% do while i<10%> '循环和判断语句,得到蓝白相间的表格背景 bgcolor="#f3f3ff" <%end if%>> <%=rs("title")%> <%=rs("boardtype")%> font color="#0000ff" size="3">%=rs("tnum")%>/font> <%i=i+1 rs.movenext loop%> '显示出查询出的十大热门话题的主题、所属讨论区和当前的文章数 <% set rs=nothing Conn.close set Conn=nothing %> </center>

返回主菜单

</body>

</html>

7.3 分类讨论区

BBS 论坛主菜单中最重要的一个选项就是分类讨论区菜单,这也是最复杂的一个程序 模块,比前面的程序更难设计。主要包括文章阅读和文章发表两个字模块。文章阅读中提 供主题和作者查询功能,还提供了方便的文章回复功能。该区包括的主要文件有:board.asp, main.asp, author.asp, title.asp, fulltext.asp, write.asp 和 fabiao.asp。下面逐一介绍这 7 个文 件的源代码。

1. board.asp 文件

board.asp 的作用是列出所有讨论区以及相关信息,程序结构并不复杂。由于有关每一 个讨论区的详细信息都存储在 board 表中,而当前每个讨论区中的文章数目需要在文章表 message 中统计,因此对这一段程序来说,最关键的就是编写其中的 SQL 语句。全部程序 代码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

```
' 密码保护文件
```

<html>

<head>

<title>选择讨论区</title>

</head>

<body bgcolor="#ffffff0">

```
<font face="方正舒体"><font color="#CCCC80">PETT-BBS 论坛</font>-
```

分类讨论区

<%

Set Conn=Server.CreateObject("ADODB.Connection")

Conn.open"dsn=bbs;uid=pett;pwd=123" 建立数据库连接对象,并打开数据库

sql="select boardname, mesnum, bdowner, bdintro from "____

&"(SELECT boardtype,count(title) as mesnum from board,"_

&"message where board.boardname=message.boardtype group by boardtype)"_____

&" a, board b where a board type=b. board name order by mesnum DESC" 4 相应的 SQL 语句

注意:这一条 SQL 语句涉及到了多表查询,详细解释见前面关于 SQL 语句的章节。

```
set rs=conn.execute(sql)%> '执行 SQL 语句
```

```
<font face="方正舒体" color="#FF0000" size="6">分类讨论区</font>
```

<center>

讨论区

目前文章数

版主

本版简介


```
<% do while not rs.eof%>
   bgcolor="#f3f3ff"
        <%end if%>> '使单元格的背景色间隔显示,美化显示效果
      <font face="楷体 2312" color="#ff0000" size="4"><a
 href="main.asp?board=<%=rs("boardname")%>"><%=rs("boardname")%></a></font>
                   '传递讨论区名称参数
      >>font face="方正舒体" color="#0000ff"
 size="4"><%=rs("mesnum")%></font>
      <font color="#0000ff" size="4"><a
 href="mailto:<%=rs("bdowner")%>@263.net"><%=rs("bdowner")%></a></font>
      <font style="font-size: 14px"
 color="#000080"><%=rs("bdintro")%></font>
      <%i=i+1
   rs.movenext
     loop
     conn.close
     set rs=nothing
     set conn=nothing%>
' 以表格形式显示出查询的结果
   </center>
```

```
<a href="choice.asp">返回主菜单</a>
```

</body>

board.asp 中提供了所有讨论区的链接,实际上所有讨论区链接的目标文件都是 main.asp,只是传递的参数不同。main.asp 接受这些参数,并根据参数进行查询操作,显示 出相应讨论区中的文章。

2. main.asp 文件

main.asp 文件是此模块中最关键的文件,在这个程序中同时实现了多项功能,因此程 序的流程也比较复杂。在编写程序时,首先需要设计程序的流程图,然后再根据流程图编 写相应的程序。 main.asp 的程序流程如图 7-2 所示。



图 7-2 main.asp 程序流程图

程序的源代码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

```
'密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件
```

<html>

<head>

```
<title>BBS 论坛</title>
```

</head>

<body bgcolor=#fffff0>

```
<%if session("board")="" then
```

```
session("board")="测试版"
```

```
end if
```

'判断 session("board")变量是否为空,如果为空则选择默认值

```
注意: 这里的 session("board")变量是用来存放用户的讨论区选择,保证从表中 选择出同一讨论区的文章。
```

if request.form("titleway")="" and request.form("timeway")="" then

session("readway")="timeway"

```
'判断用户是否选择了阅读方式,如果没有选择则采用默认的时间阅读方式 elseif request.form("titleway")<>"" then
```

```
session("readway")=request.form("titleway")
'如果用户选择主题阅读方式,则赋值 session(' readway')变量为 titleway
 else
session("readway")=request.form("timeway")
                                      ' 否则选择时间阅读方式
 end if
 if request.querystring("board") <> "" then
    session("board")=request.querystring("board")
end if
                   '如果接受到相关的讨论区选择参数不为空,赋值给 session(' board')变量
注意:从下面开始的代码用来实现分页显示,使用 ASP 中的 recordset (记录集)
对象可以很容易地实现数据记录的分页显示。为了功能更完整,还需要提供前一
页、下一页的按钮以及直接跳到某一页的功能。关于 recordset 对象在 6.7 节已
经介绍过,这里不再赘述。
if request.form("pagenum") >0 then
                                         '判断用户是否指定了页数
session("currentpage")=request.form("pagenum")
 elseif request.querystring("pid")=1 then
session("currentpage")=session("currentpage")-1
                                         '如果用户选择了前一页,则页数减1
 elseif request.querystring("pid")=2 then
session("currentpage")=session("currentpage")+1
                                          '如果用户选择了后一页,则页数加1
 else
session("currentpage")=1
                                          '如果没有选择,则默认值为第1页
end if
%>
<font face="方正舒体"><font color="#CCCC80">PETT-BBS 论坛</font>-</font><font</pre>
 face="楷体 GB2312" size="5" color="#0000FF"><%=session("board")%>讨论区</font>
set myconn=server.createobject("ADODB.connection")
<%
    set rs=server.createobject("ADODB.recordset")
                                   '建立数据库连接对象和 recordset 对象,打开数据库
myconn.open"dsn=bbs;uid=pett;pwd=123"
    rs.cursorlocation=3
   if session("readway")="timeway" then
    sql="select * from message where boardtype=""&session("board")&" order by addtime DESC"
  else
    sql="select * from message where boardtype=""&session("board")&" order by title,addtime DESC"
                                   '根据不同的阅读方式定义相应的 SQL 查询语句
   end if
                                   '执行 SOL 语句
   rs.open sql,myconn
                                   ' 定义每一页显示的记录数
   rs.pagesize=5
%>
```

共有<%=rs.pagecount%>页,共有<%=rs.recordcount%>篇文章

当前页面: <%=session("currentpage")%>
'显示总页数、文章数和当前的页数
<% dim a,b,c,d
a=rs.pagesize
b=session("currentpage")
c=a*b
d=a*(b-1)
%>
<center></center>
编号
作者
文章标题
发表时间
<%
do while not rs.eof
i=i+1
%>
<% if numrows>=d and numrows <c then%=""></c>
<%=i%>
<font <="" color="#0000ff" face="万止舒体" td="">
size="3"><%=rs("addname")%>

<a addid")%="" href="fulltext.asp?addid=<%=rs(">" target="windows resource">
<%1f rs("re")="re" then%>
10:
<%end it%><%=rs("title")%>

·

```
<font color="#0000ff" size="3"><%=rs("addtime")%></font>
      <% end if%>
 <%
rs.movenext
  numrows=numrows+1
  loop
  %>
                     ' 结束分页显示程序
  </center>
   <div align="center">
    <center>
  <% if session("currentpage")>1 then%>
  <a href="main.asp?pid=1">【前一页】</a>
  <% end if%>
                    '如果当前不是第1页,则显示前一页的链接,提交 pid 参数
   <% if session("currentpage")<rs.pagecount then%>
  <a href="main.asp?pid=2">【下一页】</a>
  <% end if%>
                              '如果当前不是最后一页,则显示下一页的链接
   <% if rs.pagecount>1 then%>
                               '如果总页数大于1,则直接跳到某一页的表单
<form method="POST" action="main.asp">
跳到第<input type="text" name="pagenum" size="20">页<input type="submit" value="提交"
 name="B1">
</form>
<%end if%>
   <form method="POST" action="author.asp">
r>作者搜索<input type="text" name="author" size="20"><input type="submit" value="提交"</p>
```

' 作者捜索表单

name="B1">

</form>

<form method="POST" action="title.asp">

>主题搜索<input type="text" name="title" size="20"><input type="submit" value="提交"

name="B1">

</form>

' 主题搜索表单

<form name="navigation">

<select name="list"

onChange="window.location.href=document.navigation.list.options[document.navigation.list.selectedIndex

].value"

size="1">

<option selected>讨论区导航</option>

<%osql="select boardname from board"

```
set ors=myconn.execute(osql)
```

do while not ors.eof

if ors("boardname") <> session("board") then%>

<option

value="main.asp?board=<%=ors("boardname")%>"><%=ors("boardname")%></option>

<%end if

ors.movenext

```
loop
```

set ors=nothing%>

</select>

'生成下拉菜单,用于各个讨论区之间的切换,方便用户浏览

注意:这是一段混杂的程序段,其中使用了 JavaScript 和 VBScript 两种代码, 但其中的 VBScript 是在服务器端执行的, JavaScript 则是在客户端执行。用户 在下拉菜单中选择了某一讨论区之后,将自动切换到相应的讨论区,不需要提供 按钮,这一自动功能是通过上面程序中的 JavaScript 来实现的。而其中的 VBScript 代码则是从讨论区表中选择出所有的讨论区名称,显示出除当前讨论区 以外的所有讨论区列表。

```
</form>
<%
```

rs.close

myconn.close

' 关闭数据库连接

<%if session("readway")="timeway" then%>

<form method="POST" action="main.asp">

<input type="hidden" name="titleway" value="titleway">

<input type="submit" value="主题方式阅读" name="B1">

</form>

%>

<% if session("readway")="titleway" then%>

<form method="POST" action="main.asp">

<input type="hidden" name="timeway" value="timeway">

```
<input type="submit" value="时间方式阅读" name="B1">
```

</form>

<%end if%> '如果当前的阅读方式是主题方式,则显示时间方式阅读的按钮

注意: 这里的阅读方式选择是通过一个隐藏表单实现的。当需要传递一些已知的 参数时,隐藏表单方式优于直接的参数传递方式。

<form method="POST" action="write.asp">

<input type="submit" value="发表文章" name="B1">

</form> '显示发表文章按钮

返回主菜单

</center>

</div>

</body>

3. author.asp

main.asp 程序提供了作者查询功能,从 main.asp 的源程序代码可以看到, author.asp 是 用来接收提交的作者字符串,在 author.asp 中,根据接收的作者字符串从数据表中查询出指 定作者发表的所有文章。

author.asp 程序比起上面的 main.asp 来要简单一些, 打开数据库从文章表 message 中进 行相应的查询, 同样使用分页显示查询的结果。程序源代码如下:

<!--#INCLUDE VIRTUAL="pwd.inc" -->

```
'密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件
<html>
<head>
<title>主题搜索</title>
</head>
<body bgcolor=#fffff0>
<%
  if request.form("author") <> "" then
     session("author")=request.form("author")
  end if
  if request.form("pagenum") >> 0 then
     session("currentpage")=request.form("pagenum")
  end if
  if request.querystring("pid")=1 then
     session("currentpage")=session("currentpage")-1
  end if
  if request.querystring("pid")=2 then
     session("currentpage")=session("currentpage")+1
  end if
%>
<font face="方正舒体" color="#CCCC80">Pett-BBS 论坛-</font>
                 <font style="font-size: 18px" color="#ff0000"><%=session("board")%></font>
                 <font style="font-size: 18px" color="#0000ff">讨论区-作者搜索</font>
<font color="#ff0000">当前页面: <%=session("currentpage")%></font>
<% set myconn=server.createobject("ADODB.connection")
   set rs=server.createobject("ADODB.recordset")
   myconn.open"dsn=bbs;uid=pett;pwd=123"
   rs.cursorlocation=3
   sql="select * from message where boardtype=""&session("board")&" and
  addname=""&session("author")&"" order by addtime DESC"
   rs.open sql,myconn
   rs.pagesize=5
   %>
<font color="#ff0000">共有<%=rs.pagecount%>页,<%=rs.recordcount%>篇文章
  </font>
   <% dim a,b,c,d
   a=rs.pagesize
   b=session("currentpage")
```

第7章 BBS论坛系统 c=a*b d=a*(b-1) %> 编号 作者 文章标题 发表时间 <% do while not rs.eof i=i+1%> <% if numrows>=d and numrows<c then%> <%=i%> <%=rs("addname")%> <a href="fulltext.asp?addid=<%=rs("addid")%>"> <% if rs("re")="re" then%> re: <%end if%> <%=rs("title")%> <%=rs("addtime")%> <% end if %> <% rs.movenext numrows=numrows+1 loop

%>

>

```
<% if session("currentpage")>1 then%>
```

```
<a href="author.asp?pid=1">【前一页】</a>
```

<% end if%>

<% if session("currentpage")<rs.pagecount then%>

```
<a href="author.asp?pid=2">【下一页】</a>
```

<% end if%>

>

```
<% if rs.pagecount>1 then%>
```

```
<form method="POST" action="main.asp">
```

```
跳到第<input type="text" name="pagenum" size="20">页<input type="submit" value="提交" name="B1">
```

</form>

```
<%end if%>
```

```
<form method="POST" action="author.asp">
```

```
作者查询<input type="text" name="author" size="20"><input type="submit" value="提交"</p>
```

```
name="B1">
```

</form>

```
<%
rs.close
myconn.close
%>
```

返回<%=session("board")%>讨论区<P>

</body></html>

4. title.asp

title.asp 和上面的 author.asp 程序十分相似,不同的是查询方式,这里所查询的是同一

```
主题的文章,同样,查询结果也使用分页显示,以增加可读性。
   注意:相同主题查询和相同作者查询都是在同一个分类讨论区中进行的。这里的
    讨论区跟踪是通过 session('board' )来实现的。
    <!--#INCLUDE VIRTUAL="pwd.inc" -->
    '密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件
    <html>
    <head>
    <title>主题搜索</title>
    </head>
    <body bgcolor=#fffff0>
    <%
      if request.form("title") <> "" then
         session("title")=request.form("title")
      end if
      if request.form("pagenum") <> 0 then
        .session("currentpage")=request.form("pagenum")
      end if
      if request.querystring("pid")=1 then
         session("currentpage")=session("currentpage")-1
      end if
      if request.querystring("pid")=2 then
         session("currentpage")=session("currentpage")+1
      end if
    %>
    <font face="方正舒体" color="#CCCC80">Pett-BBS 论坛:</font>
                    <font style="font-size: 18px" color="#ff0000"><%=session("board")%></font>
                    <font style="font-size: 18px" color="#0000ff">讨论区-主题搜索</font>
    <font color="#ff0000">当前页面:
    <%=session("currentpage")%></font>
    <% set myconn=server.createobject("ADODB.connection")
       set rs=server.createobject("ADODB.recordset")
       myconn.open"dsn=bbs;uid=pett;pwd=123"
       rs.cursorlocation=3
       sql="select * from message where boardtype="&session("board")&" and
     title=""&session("title")&" order by addtime DESC"
       rs.open sql,myconn
       rs.pagesize=5
```

%>

共有<%=rs.pagecount%>页,<%=rs.recordcount%>篇文章

<% dim a,b,c,d

a=rs.pagesize

b=session("currentpage")

c=a*b

d=a*(b-1)

%>

编号

作者

文章标题

发表时间

<%

do while not rs.eof

i=i+1%>

<% if numrows>=d and numrows<c then%>

<%=i%>

```
<font face="方正舒体" color="#0000ff"
```

```
size="3"><%=rs("addname")%></font>
```

<a

href="fulltext.asp?addid=<%=rs("addid")%>">

```
<% if rs("re")="re" then%>
```

re:

<%end if%>

<%=rs("title")%>

<%=rs("addtime")%>

```
<% end if %>
 <%
rs.movenext
  numrows=numrows+1
  loop
  %>
  <% if session("currentpage")>1 then%>
  <a href="title.asp?pid=1">【前一页】</a>
  <% end if%>
  <% if session("currentpage")<rs.pagecount then%>
  <a href="title.asp?pid=2">【下一页】</a>
  <% end if%>
  <
<% if rs.pagecount>1 then%>
<form method="POST" action="title.asp">
 跳到第<input type="text" name="pagenum" size="20">页<input type="submit" value="提交"
 name="B1">
</form>
<%end if%>
<form method="POST" action="title.asp">
 主题查询<input type="text" name="title" size="20"><input type="submit" value="提交"
 name="B1">
</form>
<a href="main.asp">返回<%=session("board")%>讨论区</a><P>
   <%
   rs.close
   myconn.close
   %>
 </body></html>
```

5. fulltext.asp

在上面的不同阅读方式和查询中,所有的结果都只显示了主题、作者和所属讨论区的 列表。在主题上提供了全文阅读的链接,当需要阅读全文时,只需要单击相应的主题就可 以打开新的窗口,显示文章的全文内容。

在程序中,根据提交的文章编号参数,从文章表 message 中查询出相应的记录,然后显示出文章的全文和相关信息,并且从用户资料表中,查询出文章作者的个人说明文档和 相关信息,最后以表格形式显示出所有的查询内容。相应的源代码如下:

<!--#INCLUDE VIRTUAL="pwd.inc" -->

'密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件

<html>

<head>

<title>全文阅读</title>

</head>

```
<body bgcolor="#fffff0">
```

<% set conn=server.createobject("ADODB.connection")

conn.open"dsn=bbs;uid=pett;pwd=123"

sql="select * from message where addid="&request.querystring("addid")&""

set rs=conn.execute(sql)

```
sql1="select pen from member where lybid=""&rs("addname")&"""
```

```
set rst=conn.execute(sql1)
```

%>

```
<font face="方正舒体"><font color="#CCCC80">PETT-BBS 论坛</font>-</font></font</pre>
```

```
face="楷体_GB2312" size="5" color="#0000FF">全文阅读</font>
```

<div align="center">

<center>

```
<font color="#0000FF" face="楷体 GB2312">发信人:
```



```
<font color="#000000" face="宋体"><%=rs("addname")%></font><font color="#0000FF" face="楷体 GB2312">信区: </font>
```

<%=session("board")%>

```
<font color="#0000FF" face="楷体_GB2312">标&nbsp;题:
</font>
```

```
<font color="#000000" face="宋体
"><%=rs("title")%></font>
```

时 间: <%=rs("addtime")%> <%=rs("message")%> 个人说明档: font color="#000080"><%=rst("pen")%>

来源于:


```
<%=rs("addip")%>
```



```
<form method="POST" action="write.asp">
```

<input type="hidden" name="rewrite"</pre>

```
value="<%=rs("title")%>">&nbsp;
```

<input type="submit" value="回复" name="B2">

```
<a href="fulltext.asp?addid=<%=request.querystring("addid")-1%>"><input
type="button" value="上一篇" name="B3"></a>&nbsp;
```

```
<a href="fulltext.asp?addid=<%=request.querystring("addid")+1%>"><input
type="button" value="下一篇" name="B1"></a>
```

```
</form>
```

6. write.asp

write.asp 提供了输入文章的表单,其中主要有一个判断程序,如果是发表新文章的话, 将直接显示出相应的表单,如果是回复文章,则自动写入文章的标题,并且自动加上"Re" 回复标识。输入完毕后,表单中的信息将提交到 fabiao.asp 进行处理。

<!--#INCLUDE VIRTUAL="pwd.inc" -->

'密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件

<html>

<head>

<title>发表文章</title>

</head>

<body bgcolor="#fffff0">

PETT-BBS 论坛-

<%=session("board")%>讨论

区

<% if

request.form("rewrite") <> "" then%>

回复文章

<%else%>

发表文章

<%end if%>

<center>

<form method="POST" action="fabiao.asp">

标题:

<%if request.form("rewrite") >"" then%>

RE:

<%end if%>

<input type="text" name="title" size="20"

<% if request.form("rewrite") >"" then%>

```
value="<%=request.form("rewrite")%>">
  <%end if%>
  内容: <textarea rows="3" name="message" cols="50"></textarea>
  <input type="hidden" name="re"
     <% if request.form("rewrite") >"" then%>
           value="re"
     <% else%>
           value="00"
     <% end if%>>
     <input type="submit" value="提交" name="B1"> <input type="reset" value="全部重写"
 name="B2">
</form>
</center>
</body>
</html>
7. fabiao.asp
```

fabiao.asp 用来处理 write.asp 中提交的发表文章的信息,把文章的标题、内容存入后台数据库,并记录下发表的时间、IP 地址,另外在 member 表中累加当前作者发表的文章数。 用户在前台的页面中是不会看到这一文件,因为当相应的数据存储和更新后,将自动引导到 main.asp 页面,也就是返回到讨论区中。

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

'密码保护程序,和前面的程序保护 pwd.inc 同样的密码保护文件

<html>

<head>

```
<title>全文阅读</title>
```

</head>

<body bgcolor="#ffffff0">

<% set conn=server.createobject("ADODB.connection")

```
conn.open"dsn=bbs;uid=pett;pwd=123"
```

sql="insert into message (addname,title,message,boardtype,addip,re) "_

&"values("'&session("user")&"',"'&request.form("title")&"',"'&request.form("message")&"',"_

&"""&session("board")&"',"&request.servervariables("remote_addr")&"',"&request.form("re")&"')"
sql1="update member set fabiaonum=fabiaonum+1 where lybid=""&session("user")&"'"
conn.execute(sql)

conn.execute(sql1)

%>

文章发表成功! 谢谢您的支

持!

<% conn.close

set conn=nothing

response.redirect"main.asp"

%>

</body>

</html>

7.4 资料修改系统

BBS 论坛用户在一定的时候可能需要修改自己的个人资料,所以,一个 BBS 论坛系统 应该具备这样的功能。本 BBS 论坛的个人资料修改系统由 modify.asp 和 update.asp 两个文 件组成。

1. modify.asp

modify.asp 文件主要是以表单形式显示数据表中当前用户已有的个人资料,显示方式 和注册系统中 regist.htm 的形式类似,不同的是这里的表单已具有一定的初始值,而且不提 供用户账号的输入,也就是说一个用户的账号名称是不能修改的。用户修改完后,将把所 有的信息提交到 update.asp 文件,然后传送到后台数据库中。

另外,和 regist.htm 文件一样,在这个文件中也需要一个输入限制程序,以限定用户必须填写一些项目。这里省略了这个限制程序,可以参考前面对表单限制程序的介绍。除限制程序外, modify.asp 文件的源代码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

<title>修改个人资料</title>

</head>

<body bgcolor=#fffff0>

<% Set Conn=Server.CreateObject("ADODB.Connection")</p>

Conn.open"dsn=bbs;uid=pett;pwd=123"

sq="select * from member where lybid=""&session("user")&"' and password="&session("pwd")&"'' set rs=Conn.execute(sq)%>

<%=session("user")%>,你的个人资料设置如下:

```
第 7 章 BBS 论坛系统
                                                                             235
<form method="POST" action="update.asp" name="input">
 <div align="center">
   <center>
  <font style="font-size: 14px" color="#000080">真实姓名</font>
     <input name="lybname" style="font-size: 9pt; border: 1 solid #800080" size="20"
 value="<%=rs("lybname")%>"><font style="font-size: 14px">*</font>
   <font style="font-size: 14px" color="#000080">昵&nbsp;称</font>
     <input name="nickname" style="font-size: 9pt; border: 1 solid #800080"
 size="20" size="20" value="<%=rs("nickname")%>">*
   <font style="font-size: 14px" color="#000080">生&nbsp;日</font>
     <font style="font-size: 14px"><input name="birthday" style="font-size: 9pt; border: 1 solid
 #800080" size="20" size="20" value="<%=rs("birthday")%>"></font>
   <font style="font-size: 14px" color="#000080">密&nbsp;码</font>
     <font style="font-size: 14px"><input name="password" style="font-size: 9pt; border: 1 solid
 #800080" size="20" type="password" size="20" ></font>*
    <font style="font-size: 14px" color="#000080">密码验证</font>
     <font style="font-size: 14px"><input name="spassword" style="font-size: 9pt; border: 1 solid
 #800080" size="20" type="password" size="20" ></font>*
    <font style="font-size: 14px" color="#000080">提示问题</font>
     <input name="question" style="border: 1 solid #800080" size="25"
 value="<%=rs("question")%>">*<font style="font-size: 14px" color="#FF00FF">(如"吃了没
 有? ")</font>
    <font style="font-size: 14px" color="#000080">提示答案</font>
      <input name="answer" style="border: 1 solid #800080" size="25"
 value="<%=rs("answer")%>">*<font style="font-size: 14px" color="#FF00FF">(如"还没吃呢")
```

```
</font>
```

通讯地址

<input name="address" style="border: 1 solid #800080" size="40" value="<%=rs("address")%>">

font style="font-size: 14px" color="#000080">工作单位

<input name="workunit" style="border: 1 solid #800080" size="33"

```
value="<%=rs("workunit")%>">
```

联系电话

<input name="telephone" size="16" style="border: 1 solid #800080"

value="<%=rs("telephone")%>">

```
<font style="FONT-SIZE: 14px" color="#000080">BP 机号</font><input name="bpnumber" style="border: 1 solid #800080" size="16"</td>value="<%=rs("bpnumber")%>">
```

```
<font style="FONT-SIZE: 14px" color="#000080">E-mail</font><input name="email" style="border: 1 solid #800080" size="26"</td>value="<%=rs("email")%>">
```

网上寻呼号

<input name="icq" size="16" style="border: 1 solid #800080" value="<%=rs("icq")%>">

个人签名档

textarea cols="45" name="pen" rows="2" style="font-size: 9pt; background-color: #F3F3FF; color: #800000; border: 1 solid #000000"><%=rs("pen")%></textarea>

备 注

```
<textarea cols="45" name="other" rows="2" style="font-size: 9pt; background-color: #FFFFFF;</td>color: #800000; border: 1 solid #000000"><%=rs("other")%></textarea></div><input type="submit" value="提交" name="B1"style="background-color: #800080;</td>color: #FFFF00; font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-style: ridge;border-color: #FFFF00" OnClick="return submitclick();"><input type="reset" value="全部重写" name="B2"style="background-color: #800080; color: #FFFF00;</td>font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-style: ridge;<input type="reset" value="全部重写" name="B2"style="background-color: #800080; color: #FFFF00;</td>font-family: 楷体_GB2312; font-size: 12pt; font-weight: bold; border-color: #FFFF00;</t
```

conn.close

set conn=nothing

%>

2. update.asp

update.asp 用于处理 modify.asp 中提交的用户资料,并且把数据表中的用户资料进行更新。update.asp 文件十分简单,仅仅执行一个 SQL update 语句。update.asp 文件的源代码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

```
<title>注册成功</title>
```

</head>

<body bgcolor="#fffff0">

<% session("pwd")=request.form("password")

Set myconn=Server.CreateObject("ADODB.connection")

myconn.open"dsn=bbs;uid=pett;pwd=123"

sql="update member set "_

- &"lybname=''&request.form("lybname")&''',nickname='''&request.form("nickname")&''',password='''&req uest.form("password")&''',"_
- &"birthday='"&request.form("birthday")&"',question='"&request.form("question")&"',answer=''&request.form("answer")&"',answer=''&request.form("address")&"',"_
- &"workunit=""&request.form("workunit")&"",telephone=""&request.form("telephone")&"",b pnumber=""&request.form("bpnumber")&"",email=""&request.form("email")&"","_

&"icq="'&request.form("icq")&"',other='"&request.form("other")&"',pen='"&request.form("pen")&"',addip ='"&request.servervariables("remote_addr")&"' "_

&"where lybid=""&session("user")&" and password=""&session("pwd")&"""

myconn.execute(sql)

%>

font face="方正舒体">font color="#CCCC80">PETT-BBS 论坛-

```
</font><font face="楷体 GB2312" size="5" color="#0000FF">修改成功</font>
```

您的个人资料已经安全输入到数 据库中!

<%

set myconn=nothing

%>

```
<a href="choice.asp">返回主菜单</a>
```

</body>

</html>

7.5 网友查询系统

一般的 BBS 论坛系统都具有网友查询功能,以方便论坛用户之间互相交流和了解。这 里的网友查询系统并不显示用户的所有个人资料,仅仅显示所查询用户的一些上站情况, 例如上站次数、发表文章数、上次登录时间和离开时间、目前是否在站上等等。该系统仅 仅包含一个文件: queryfriend.asp。

queryfriend.asp 文件结构也十分简单,根据当前用户输入的用户名,从用户资料表中查询出相应用户的部分个人资料,以一定的形式显示给当前用户。

queryfriend.asp 的源程序如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

'密码保护程序,和前面的程序保护 pws.inc 同样的密码保护文件

<html>

<head>

<title>网友查询</title>

</head>

<body bgcolor="#fffff0">

PETT-BBS 论坛--网友查询

<% set conn=server.createobject("ADODB.connection")

conn.open"dsn=bbs;uid=pett;pwd=123"

```
if request.form("friendname") "" then
     sql="select * from member where lybid="%request.form("friendname")&""
     sql1="select * from letter where receiver=""&request.form("friendname")&" and read=1"
  set rst=conn.execute(sql1)
  set rs=conn.execute(sql)
%>
<%if not rs.eof then%>
<div align="center">
 <font color="#0000FF" face="楷体</pre>
 _GB2312"><%=rs("lybid")%>(<%=rs("nickname")%>)
      共上站<%=rs("vtimes")%>次,发表过<%=rs("fabiaonum")%>篇文章。
      </font>
         <center>
   <font color="#0000FF" face="楷体_GB2312">上次在[<%=rs("intime")%>]
       从[<%=rs("inip")%>]到本站一游,
       </font>
   <font color="#0000FF" face="楷体_GB2312">离线时间[
       <% if rs("intime")<rs("outtime") then%>
         <%=rs("outtime")%>
                          '因在线上或非常断线不详
         <%else%>
         <%end if%>
         ].
       <% if not rst.eof then%>
       [有新邮件]。
       <%end if%>
       </font>
   个人说明档如下:
```

<%=rs("pen")%> </center> </div> <%else%> 用户<%=request.form("friendname")%> 不存在! 请重试! <% end if%> <% set rst=nothing set rs=nothing end if%> <form method="post" action="queryfriend.asp"> 输入用户名<input type="text" name="friendname">

```
<input type="submit" value="查询" name="B2">&nbsp;&nbsp;
```

</form>

```
<a href="choice.asp">返回主菜单</a>
```

<% conn.close

set conn=nothing

%>

</body>

</html>

7.6 个人信箱系统

为了更好地方便论坛用户之间的通信和联系,应该设置个人信箱系统。这个系统以列 表形式显示当前用户收到的信件,并且统计出信件总数和新邮件数目,在每一个邮件的列 表中标明状态(已读或者未读),方便用户查找未读的新邮件。要想向别的用户发送邮件也 很方便,只需要输入收信人的用户名,然后书写邮件内容就可以了。

本系统由 4 个文件组成: letterbox.asp, writeletter.asp, fullletter.asp, post.asp。

1. Letterbox.asp

letterbox.asp 是一个比较复杂的程序,主要表现在其中 SQL 语句的设计上。在这个文件中需要统计出当前用户的所有信件数目和新信件的数目,而程序判断信件是否为新收到的标识就是阅读次数,如果阅读次数(read 字段)为0,说明邮件未读。查询出来结果后,以列表形式列出信件的标题、发信人、收信时间和状态,点击邮件标题上的连接即可进行 全文阅读。

另外,在这个文件中提供了书写邮件的表单,只需要输入收信人的账号。文件的源代 码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

<meta http-equiv="Content-Language" content="zh-cn">

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>网友查询</title>

</head>

<body bgcolor="#fffff0">

<% set conn=server.createobject("ADODB.connection")

conn.open"dsn=bbs;uid=pett;pwd=123"

```
set rs=server.createobject("ADODB.recordset")
```

sql="select * from letter where receiver=""&session("user")&" order by addtime"

sql1="select count(read) as newletternum from letter where receiver="&session("user")&" and read=0"

rs.cursorlocation=3

rs.open sql,conn

```
set rst=conn.execute(sql1)
```

%>

PETT-BBS 论坛-个人信箱

<%=session("user")%>个人信 箱

<%if rs.eof then%>

<%=session("user")%>您目 前没有信件

<%else%>

<%=session("user")%>您有 <%=rs.recordcount%>封信,其中有

<%=rst("newletternum")%>封新邮件 <div align="center"> 编号 寄信者 主题 发信时间 状态 <% do while not rs.eof i=i+1%> font color="#0000FF" face="楷体_GB2312">%=i%> <%=rs("poster")%> <a href="fullletter.asp?addid=<%=rs("addid")%>"><%=rs("title")%> <%=rs("addtime")%> <%if rs("read") >0 then %> 已读 <%else%> 未读 <%end if%> <%rs.movenext loop end if set rst=nothing set rs=nothing%> </div> <% set rs=nothing %>
```
<form method="post" action="writeletter.asp">
```

寄信给<input type="text" name="receiver">

<input type="submit" value="书写邮件" name="B2">

</form>

返回主菜单

```
<% conn.close
```

set conn=nothing

%>

</body>

</html>

2. fullletter.asp

fulleter.asp 提供邮件的全文阅读功能,根据 letterbox.asp 中连接提供的邮件编号,从信件表查询出相应的信件记录,以表格形式显示出邮件全文,并且累加当前信件的阅读次数。相应的源代码如下:

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

```
<meta http-equiv="Content-Language" content="zh-cn">
```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>全文阅读</title>

</head>

```
<body bgcolor="#fffff0">
```

<% set conn=server.createobject("ADODB.connection")

conn.open"dsn=bbs;uid=pett;pwd=123"

sql="select * from letter where addid="&request.querystring("addid")&""

sql1="update letter set read=read+1 where addid="&request.querystring("addid")&""

```
conn.execute(sql1)
```

```
set rs=conn.execute(sql)
```

%>

PETT-BBS 论坛-阅读信件 <div align="center">

<center>

寄信人:

<%=rs("poster")%>

```
<font color="#0000FF" face="楷体 GB2312"></font>
```


标

题:

<font color="#000000" face="宋体

"><%=rs("title")%>

发信时间:

```
<font color="#000000" face="宋体
```

```
"><%=rs("addtime")%></font>
```

font style="font-size: 14px"

color="#ff0000"><%=rs("message")%>

来源于:

<%=rs("addip")%>

<form method="POST" action="writeletter.asp">

<input type="hidden" name="receiver" value="<%=rs("poster")%>">

<input type="hidden" name="title" value="<%=rs("title")%>">

<input type="submit" value="回复" name="B2">

<a href="fullletter.asp?addid=<%=request.querystring("addid")-1%>"><input

```
type="button" value="上一封" name="B3"></a>&nbsp;
```

```
<a href="fullletter.asp?addid=<%=request.querystring("addid")+1%>"><input
```

```
type="button" value="下一封" name="B1"></a>
</form>

</ta
```

%>

</body>

</html>

3. writeletter.asp

writeletter.asp 提供了书写邮件的表单,用户只需要输入信件主题和内容就可以了,表 单内容将提交到 post.asp 中进行处理。

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>书写信件</title>

</head>

<body bgcolor="#ffffff0">

```
<font face="方正舒体"><font color="#CCCC80">PETT-BBS 论坛</font>-</font><font
face="楷体_GB2312" size="5" color="#0000FF">
```

<%if request.form("title")<>"" then%>

回复<% =request.form("receiver") %>的来信

<%else%>

寄信给<% =request.form("receiver") %>

<%end if%>

<form method="POST" action="post.asp">

标题: <input type="text"
name="title" size="20"</pre>

<% if request.form("title")<>"" then%>

value="Re:<%=request.form("title")%>"

<% end if%>

>

```
<font style="font-size:14px" color="#ff0000">内容: </font><textarea rows="2"</pre>
```

```
name="message" cols="43"></textarea>
```

<input type="hidden" name="receiver"</pre>

```
value="<%=request.form("receiver")%>">
```

```
<input type="submit" value="提交" name="B1">&nbsp;&nbsp;<input type="reset" value="全部重
```

```
写" name="B2">
```

</form>

```
<center><font style="font-size: 14px" color="#ff0000"><a href="letterbox.asp">返回信箱
```

</center>

</body>

</html>

4. post.asp

post.asp 处理从上一文件接受的表单信息,并且把这些信息发送到后台数据库中,仅仅 是一个简单的 SQL Insert 语句。

```
<!--#INCLUDE VIRTUAL="pwd.inc" -->
```

<html>

<head>

```
<meta http-equiv="Content-Language" content="zh-cn">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

```
<meta name="ProgId" content="FrontPage.Editor.Document">
```

```
<title>全文阅读</title>
```

```
</head>
```

<body bgcolor="#ffffff0">

```
<% set conn=server.createobject("ADODB.connection")
```

conn.open"dsn=bbs;uid=pett;pwd=123"

sql="insert into letter (poster, receiver, title, message, addip, read) "_

```
&"values(""&session("user")&"',""&request.form("receiver")&"',""&request.form("title")&"',"&request.for
m("message")&"',"
```

```
&"""&request.servervariables("remote_addr")&"",1)"
```

```
conn.execute(sql)
%>
<font color="#FF0000" face="楷体 GB2312" size="5">信件已经发出!你的朋友很快
 就会收到的! </font>
<div align="center">
 <center>
 <a href="letterbox.asp">返回信箱</a>
    </center>
</div>
<%
     conn.close
  set conn=nothing
  response.redirect"letterbox.asp"
%>
</body>
</html>
```

7.7 密码保护程序

关于密码保护程序的概念在前面已经介绍过,这里不再赘述。这里的密码保护程序就 是前面大部分页面所包含的文件 pwd.inc。在这个文件中首先判断 Session("user")和 Session("pwd")是否为空。如为空,则使用 response.redirect 方法引导到 login.htm 页面;如 不为空,则建立数据库连接,并判断这个用户名和密码在数据库中是否存在,如不存在(即 记录为空),则也引导到 login.htm 页面。可以看到包含这个密码保护程序的页面都可以严 格受到保护,任何非法登录的用户无法进入。

pwd.inc 的源代码如下:

```
<% if session("user")="" or session("pwd")="" then
response.redirect"login.htm"
else
Set Conn=Server.CreateObject("ADODB.Connection")
Conn.open"dsn=bbs;uid=pett;pwd=123"
sq="select * from member where lybid="&session("user")&" and password="&session("pwd")&""
```

set rs=Conn.execute(sq)

if rs.EOF then

response.redirect"login.htm"

end if

set rs=nothing

conn.close

set conn=nothing

end if

%>

7.8 离开论坛

设置离开论坛功能的主要目的是:记录用户的离开时间,并且释放 Session 变量,使 Session 变量过期。程序的结构很简单,quit.asp 源代码如下:

```
<%@ LANGUAGE = "VBScript" %>
```

```
<%response.buffer=true%>
```

<%

```
set conn=server.createobject("ADODB.connection")
```

```
conn.open"dsn=bbs;uid=pett;pwd=123"
```

```
sql="update member set outtime=now() where lybid=""&session("user")&"""
```

conn.execute(sql)

conn.close

```
set conn=nothing
```

session("user")=""

response.Expires=0

response.redirect "login.htm"

%>

第2篇 Web 数据库应用实例

第1篇通过 BBS 论坛系统地讲述了如何使用 ASP+SQL Server 来构建 Web 数据库,读 者对 Web 数据库已经有了一个初步的了解。从本篇开始,读者就可以尝试开发一个 Web 数据库,以检验自己对 Web 数据库开发技术的掌握程度。

但从另一个方面来说,不同的 Web 数据库会涉及到不同的技术侧面,对技术的要求也 有所不同。比如,对于前面所讲述的 BBS 论坛实例,要求达到一定的功能,还要具有一定 的保密性,因此,在开发过程中所有用户身份验证都是通过 ASP 中的 session 来限定的。 但如果需要设计具有复杂保密级别、不同权限用户、应用性更强的 Web 数据库系统,对开 发者则会有更高的要求。

为了进一步提高读者的开发水平,本篇将讲述两个更高级的 Web 数据库系统。其一是 基于 Web 的文档资料管理系统,其中设定了不同权限的用户,对不同的文档有不同的密级, 并利用 cookie 来跟踪用户,从而提高了数据库系统的安全性。其具体的设计、开发过程见 第 8 章。其二是搜索引擎。怎样从技术角度来构建一个搜索引擎,会涉及到设计、开发、 维护等等问题,详细论述见第 9 章。在第 10 章中,给出了 9 个实用程序片断,对 Web 数 据库开发者有重要的参考价值。

第8章 文档资料管理系统

本章将着重讲述一个基于 Web 的文档资料管理系统,也是一个 Web 数据库系统,同样 是通过 ASP+SQL Server 来开发。本章中的实例是某一杂志社的文档资料管理系统,了解了 这个文档管理系统之后,就会发现它的适用性是很广的,只要改动界面就能作为其他的用途。

8.1 管理系统的任务

要开发一个应用管理系统,首先应弄清楚要达到什么样的目的,也就是要明确开发的 任务。这个文档管理系统的主要任务是管理某一杂志社的文档,其中要存放历年来此杂志 社的稿件及其相关信息,包括作者的基本个人资料,校稿者的基本个人资料,稿件的题目、 摘要、密级等信息。要求此管理系统的用户能够通过上网查询所需的稿件信息,并根据用 户权限提供相应的全文下载服务。

稿件正文的安全密级总共分为五级:0——完全公开;1——所有编辑人员都可查看;2 ——同部门所有编辑人员可查看;3——同部门专家可查看;4——仅稿件校稿者可查看。 这五个安全级别依次递增。

相应地,用户根据其权限也分为五级:0—友情用户;1—普通编辑;2—专家; 3—系统管理员;一1—非法用户。其中友情用户指的是杂志社内不具有编辑、专家、 系统管理员身份或者杂志社外的合法用户,但却有权访问稿件信息数据库并可以下载安全 级别为 0(完全公开)的稿件全文的用户,而非法用户是指由于某种特殊原因而被系统管 理员暂时中止了对稿件全文的访问权限的用户。

由于稿件信息的输入是一项工作量巨大的任务,所以由稿件作者上网输入并提交自己 的稿件信息,将自己的稿件正文上传至服务器。由于作者提交上来的稿件信息还需经过编 辑或者管理员处理(如为稿件编号等)才能存入正式的在网上发布的数据库中,所以还需 要一个临时的稿件信息数据库来专门存放作者刚提交上来的未经处理的稿件信息。为提高 正式稿件信息数据库的存取效率和安全性,临时稿件信息数据库设计为一个单独的数据库。 这样,作者提交自己的稿件信息时就不需访问正式的稿件信息数据库了,从而减少了正式 稿件信息数据库的负荷。

8.2 使用稿件管理系统

根据上一节提出的设计要求,笔者开发了一个实用的符合设计要求的稿件查询、管理 系统。在这里,首先介绍如何从一个用户的角度来使用这个稿件管理系统。至于如何具体 实现技术构建,如何开发程序和设计后台数据库,见 8.4 节 "ASP 程序开发"将在后面的 章节再讲述。

此稿件管理系统是在网络上发布的,应该做到每一位用户都能很方便地找到系统的入口。这里的做法就是在杂志社的主页上加上系统的连接,如图 8-1 所示。



图 8-1 杂志社主页

当然,这个杂志社的主页只是一个虚拟的杂志社,一个真正的杂志社的主页肯定不会 这么简单,但这个主页从技术角度能足以说明这个稿件管理系统的入口所在。由于系统的 登录链接就在杂志社的主页上,所以,对于任意一位用户,无论是低级用户还是超级用户, 或输入稿件的初次用户,都可以很容易找到系统的登录界面。

从图 8-1 中可以看出,系统的登录链接并不是一个而是 4 个,这样设计的主要目的是 为了满足原定的设计要求,对于不同权限的用户设计不同的登录界面。

注意:这里把整个系统设计为四个模块,并不意味着必须分成四个部分。其实, 可以把所有的用户登录界面都做到一起,只要通过一个简单的用户身份验证程序, 就可以做到对不同的用户区别对待,但是这会使整个系统显得十分复杂,不利于 读者分析和学习。

下面就带领读者去看一看系统不同部分都是什么样子的,都能实现什么样的功能。

1. 公共查询界面(模块1)

首先单击图 8-1 中的第一个超级链接,就可以打开系统的第一个登录界面——公共查询界面,如图 8-2 所示。

从图 8-2 中可以看出,公共查询界面并不需要任何身份验证,任何一位网络用户都可 以访问这个界面并查询。但这并不说明整个系统是完全公开的,因为查询的结果是受限制 的,从下面的查询结果中就可以看到这一点。

DIDU Microsoft Internet	txplorer			Harrison State Land	
	A		e (6- 4)	<u> .</u>	
http://166.111.17	0.110/document/que	ry/query.html	-		
DIDO杂志社>>稿件查询界的	a 17		44		
	稿件查	词界	đ		~
稿件题目:		an in the second s	many Arraitaine		
稿件作者: 校稿编辑:					
关键字:				e 与	⊂或
稿件完成时间。	2001到现在		v	a Autor Autoria	
稿件类别:	€ 学生稿件	C教师	稿件		
	○ 其他稿件				
	ANNESS CONTRACTOR			182 (12.16)	ned of
				T	

图 8-2 公共查询界面

这是稿件信息的主查询页面。在这里,总共为访问者提供了 6 个查询条件选项:稿件 题目、稿件作者、校稿编辑、关键字、稿件完成时间、稿件类别。其中,稿件完成时间和 稿件类别选项分别是以下拉莱单和单选按钮的方式提供的,其余选项都是以文本框方式填 写。所以,当用户在提交查询条件的时候,"稿件完成时间"和"稿件类别"为必选项,即 使用户不作任何选择,这两个表单也会有默认值。从图 8-3 中可以看到,"稿件完成时间" 提供了 4 个时间段供用户选择: 1998 年,1999 年,2000 年,2001 年到现在。"稿件类别" 则要求用户必须选择自己所要查询的到底是学生来的稿件还是教师的稿件或其他类型的稿 件。查询条件中的关键字最多可以输入 3 个,而且提供"与"、"或"两种逻辑关系。其余 3 个查询条件用户可填可不填,用户不填的项目在检索时就不会被列入查询条件中。

2001年到现在 🛒
2001年到现在
2000年
1999年
1998年

图 8-3 稿件完成时间区段选择

这样设计使得检索时至少具有两个查询条件,从而对查询可能命中的记录数目进行一 定的控制,并且,这两个查询条件对于绝大多数用户而言都是必要的。 在如图 8-2 所示的界面中,单击"开始查询"按钮,此时没有输入任何查询条件,提 交的就是默认的查询条件,也就是完成时间在 2001 年并且作者为学生的稿件,得到的结果 如图 8-4 所示。

	no constante		
http://166.111.170.110	D/document/query/query_result.asp	<u>), , , , , , , , , , , , , , , , , , , </u>	
条件的稿件总篇	数: 25		
这是第1页			
张鑫	·····································	影	2001-2-10
阿刘	网络现状分析	段清	2001-1-12
起单	moDom法购与使用	段清	2001-1-2
阿刘	网络硬件选择	刘高	2001-3-1
禁明	<u>如何确定自己的人生道路</u>	陈桂	2001-2-15
陈辉	<u>她何提高自己的地位</u>	<u> 長分</u>	2001-2-20
畫杰	原材料市场分析	<u> 吴珺</u>	2001-1-25
作別	<u>文人嗜好</u>	老发	2001-3-12
起其	全民维身,利国利民	末頭	2001-2-16
刘东	两上聊天纪史	段清	2001-2-10
	<u>01</u> <u>02</u> <u>03</u>		
	重新查询		
	http://166.111.170.114 条件的稿件总篇 这是第1页 米島 阿刘 差単 阿刘 黎明 陈輝 蓋杰 件別 赵其 刘东	大きのうくまま、ため、ない、ない、ない、ない、ない、ない、ない、ない、ない、ない、ない、ない、ない、	No. No.

图 8-4 查询结果

从图 8-4 中可以看到查询的结果,共返回了 25 条信息,分 3 页显示,当前显示的是第 1 页,也就是 1~10 篇符合查询条件的稿件,这里列出了稿件题目、作者、稿件编辑和完 成时间,从这些简要的信息中,用户就可以找到自己需要的稿件信息。从图 8-4 中还可以 看到不同页码的超链接,单击就可以翻到对应的页面,方便用户浏览。从图中还可以看到 稿件题目和稿件编辑都为超级链接的形式,单击就可以看到更详细的信息。

在如图 8-4 所示的界面中,单击第1篇稿件的篇名链接"如何上网",将自动打开如图 8-5 所示的稿件摘要界面,从中可以看到稿件的摘要和关键字。在这种界面下,用户无法看 到稿件的全文,也就是说,在这个公共稿件查询模块中用户并不能得到稿件的全文,只能 得到简要的信息。

同样,在如图 8-4 所示的界面中单击相应的稿件编辑链接,也将打开一个新的窗口, 如图 8-6 所示,显示出相应的编辑的简要信息,包括编辑姓名、所属部门、办公电话和电 子邮件地址。通过这些信息,普通用户就可以与稿件编辑取得联系。

] 箱件确要 _ Microsoft Internet Explorer	N R
稿件《如何上网》的摘要:	
文中详细论述了,在当前的网络硬件环境下,如何上网?并对不同的用户, 不同的分析。	做了
关键字:电脑,电话,modem,局域网,网卡	
[关闭窗口]	

图 8-5 稿件摘要界面

 网络部	68142578	benfox#263. net	

图 8-6 编辑信息

公共稿件的查询界面已经介绍完了,读者可能会觉得这个模块实现起来并不困难,和 第1篇中介绍的 BBS 论坛的显示文章模块大同小异。确实是这样,这个模块中的核心技术 就是分页显示程序,分页显示程序在 Web 数据库的前台程序中广泛应用,十分重要。下面 将介绍加上用户身份验证的查询系统。

2. 权限用户查询模块 (模块 2)

权限用户查询模块主要是为具有一定权限的用户设计的,因为杂志社中的稿件具有不同的保密级别,而不同的用户又具有不同的保密级别。这个模块的功能就是能够使所有用户都能查询到自己权限以内的稿件信息。

但是,这里的用户身份必须通过系统管理员的验证才能成为合法用户。

在如图 8-1 所示的主页上单击"权限用户登录"链接,浏览器中将显示如图 8-7 所示 的权限用户登录界面。因为这里并不是公共的查询模块,不允许匿名访问,所以用户必须 输入用户名和密码,以通过系统对用户的身份验证。

当然,如果在当前的数据库中并没有当前用户的身份认证,就必须进行注册申请。在 如图 8-7 所示的界面中单击"注册申请",将得到如图 8-8 所示的新用户注册页面,在这里, 新用户必须如实填写自己的相关信息,否则无法通过系统管理员的认证,从而无法完成注 册。

予用户資彙 - Microsoft Internet Oxplorer	
http://166.111.170.110/document/advance/login.asp	and Louise Mg
DIDO杂志社>>权限用户登录	
D9D0稿件系统权限用户登录	
用户名:	
密码。 注册申请	

图 8-7 权限用户登录

新用户注册
3 223
请填写真实个人资料 真实姓名: 边想
「「「周部门: 网络部 ● 「「周部门: 网络部 ● 「「「「「「「」」」」 「「「「」」」」
新技木部 程序部 数据序部 国内部 国际部 计算机图像部

图 8-8 新用户注册界面

用户所要填写的个人资料很简单,包括真实姓名、E-mail 地址和所属部门,其中的所 属部门是以下拉菜单的形式供用户选择的,其选项涉及杂志社内的所有部门,包括网络部、 软件部、硬件部、新技术部、程序部、数据库部、国内部、国际部、计算机图像部、网络 文化部、总编室、社内领导。从这里可以看出,只有杂志社内的用户具有注册申请资格, 社外的用户没有这样的资格,也无法通过系统管理员的身份认证。 假设当前有一位属于网络部的编辑,填入了如图 8-8 中所示的真实个人信息,单击"提 交申请"按钮后,将出现如图 8-9 所示的页面。



图 8-9 注册申请成功提交

窗口中出现提示"申请成功提交",这表明申请已经成功提交到系统管理员手中,下一步就是等待系统管理员确认。

假如,系统管理员在收到"边慧"编辑的注册申请后,立即通过了身份验证,这时, 新用户会在自己的信箱中发现系统管理员寄来的用户名和密码以及相关的权限信息。获得 了确认之后,就可以在如图 8-7 所示的权限用户登录界面中输入用户名和密码,登录此查 询系统。

同样,在查询窗口中输入查询条件,就可以得到如图 8-4 所示的查询结果。同样,查 询结果也是以分页方式显示,并且稿件题目和校稿编辑都是以超链接的形式提供,单击相 应的超链接,就可以得到进一步的详细信息。

假如单击图 8-4 中的第1篇稿件题目,则弹出如图 8-10 所示的窗口。该窗口中显示了 稿件《如何上网》的摘要和关键字,和图 8-5 惟一不同的地方就是窗口中提供了"正文链 接",单击"正文链接",就可以下载本稿件的正文文件。

3 稿件執要 Microsoft Internet Explorer 際	
论文 《如何上网》的摘要。	
文中详细论述了,在当前的网络硬件环境下,如何上网?并对不同的用户,做了 不同的分析。	
关键字:电脑,电话,modem,局域网,网卡	
正文链	e
[关闭窗口]	

图 8-10 权限用户查询结果

假如当前的用户需要查询稿件的正文文件,则单击图 8-10 中的"正文链接"将弹出如 图 8-11 所示的文件下载窗口。下一步如何保存文件到本地磁盘上,这里就不再赘述。



图 8-11 文件下载对话框

注意:在权限用户查询系统中得到的查询结果并非都提供正文下载,也就是说, 在如图 8-10 所示的窗口中,并不是所有的结果稿件都有正文下载链接,而是当前 用户权限以内的稿件才有正文下载。具体的权限分配方式 8.4 节有详细说明。

至此,权限用户的查询系统就介绍完了,通过比较以上两个模块可以发现,这两个模 块有很多的相同点,不同的就是正文下载。公共查询界面只能得到稿件的摘要,而无法得 到稿件的正文下载,而在权限用户查询系统上,登录用户可以得到自己权限内的稿件正文。

3. 稿件输入系统(模块3)

作为一个正常运行的杂志社,随时都可能有作者将新的稿件发送到稿件信息数据库中。 如果所有稿件的所有信息都由稿件管理员输入,这将是一项非常艰苦的工作。所以最佳的 考虑是让作者通过网络来输入自己的稿件信息,这样既提高了效率又减轻了稿件管理员的 负担。

假如有一位作者想向杂志社投稿,就可以在杂志社的主页上单击"稿件信息输入"链接,从而得到如图 8-12 所示的窗口。

从图 8-12 中可以看出,稿件作者必须输入有关个人和稿件的详细信息,包括姓名、联 系电话、用户名、通讯地址、稿件中英文题目、完成时间、中文关键字、英文关键字、中 文摘要、英文摘要。这里设置的"校稿编辑"一项是为了维护信息的完整性,因为每一份 稿件在数据库中都有"校稿编辑"一项,而且某些情况下,稿件信息的输入是由编辑来完 成的。当然,一般的稿件用户不必填写"校稿编辑"一栏。

■学生未大论文論人 Microsoft Internet Explorer	
Republic http://166.111.170.110/documenc/imput/imput/asp	a second that is the second
请输入真实的稿件信息	
	ana and the second seco
植针央义团日:	
校稿编辑:	3
李成时间. · · · · · · · · · · · · · · · · · · ·	I—90
中文关键字,	以","分隔
孝 文关備字:	以 "." 分隔
	e 🕺
英文编要:	×

图 8-12 稿件输入窗口

填写完上面的必要信息后,单击"提交信息"按钮,如果填入的信息符合要求,将发送到数据库中,并出现如图 8-13 所示的窗口,提示提交成功。

]提父成功 - Marosoft Internet Explorer
and a second
E ALCONTRACTOR
http://166.111.170.110/document/input_
· 您输入的稿件信息已经成功提交(
请将您的稿件压霜后上传
以用户名作为文件名上传。
码ffzip又ff上t#
at at a state of the state of t
[返回]

图 8-13 稿件正文上载

稿件信息输入到数据库后,还需要作者或者编辑上载稿件正文文件,这里需要把稿件 的文件压缩为 zip 文件,并且以上面的用户名为文件名。在本地磁盘上做完这些操作之后, 单击"浏览"按钮,就可以找到本地磁盘上的文件,然后单击"上载稿件"按钮,稿件将 上载到服务器。

至此,稿件输入模块就介绍完了,使用这个系统,作者或者编辑可以很方便地输入稿 件的信息并且上载稿件正文。

4. 稿件数据库管理系统(模块 4)

稿件数据库在运行过程中需要不断地维护和更新,这些工作要由管理员来完成,因此 需要稿件数据库管理系统。笔者开发的这个管理系统具有高度的保密性和安全性,使用方 便,操作简单,所有操作都是可视化的,即使不懂数据库技术的人也可以担任此稿件数据 库的管理员。

下面详细讲述如何完成稿件数据库的管理操作。

首先,需要进行系统管理员的身份认证,因此需要输入用户名和密码。在图 8-1 所示的杂志社主页上单击"管理员界面",将得到如图 8-14 所示的登录窗口,需要输入用户名和密码。

■论文教紙所管理员登录 Micro 公式	asoft Internet Explorer
http://166.111.170.110	0/document/manager/login.htm
稿件数词	居库管理员登录
用户名。	
密码.	

图 8-14 系统管理员登录

假如现在有一位管理员,输入正确的用户名和密码后,将顺利通过系统的身份验证, 进入管理界面,完成相应的管理。

注意:只有具有管理权限的用户才能登录此系统。能登录管理系统的用户分为两 类,一类是稿件管理员,一类是系统管理员。这两类管理员拥有不同的权限:稿 件管理员对作者信息表、编辑信息表和正式稿件信息表拥有添加权限(所谓对正 式稿件信息表的添加权限也只是指能够将临时稿件表中的原始信息进行处理后加 入正式库中),对任何表都没有修改和删除权限;而系统管理员则拥有一切权限。

假设当前有一位"benfox"的稿件管理员,登录稿件数据库管理系统,在图 8-14 所示

的窗口中输入用户名和密码后,将进入如图 8-15 所示的稿件数据库管理界面,其中的两个 下拉菜单中的内容如图 8-16 所示。

+ ## - + - • • • • • • • • • • • • • • • • •	-1400
http://166.111.170.110/document/manager/login_result.asp	
您要管理哪一个表, 学生作者信息表 您要对该表进行何种操作, 添加记录	开始操作

图 8-15 稿件数据库管理界面(一)

学生作者信息表 💌
学生作者信息表
其他作者信息表
编辑信息表
旧戶信息表
临时稿件信息表

添加记录 💌
漆加记录
修改记录
删除记录

(a) 信息表选项下拉菜单

(b) 操作选项下拉菜单

图 8-16 稿件数据库管理界面中的下拉菜单

从图 8-16 可以看出,信息表选项中共有 7 个选项,分别是学生作者信息表、教师作者 信息表、其他作者信息表、编辑信息表、正式稿件信息表、用户信息表、临时稿件信息表; 操作选项中共有 3 个选项,分别是添加记录、修改记录、删除记录。也就是说,对每一个 表都有 3 种不同的操作,在此管理界面中可以完成 21 种管理操作。

不要忘记,在前面已经提示过,稿件管理员没有修改和删除的权限,也没有管理用户 信息表和临时信息表的权限,就是说,当前的登录用户"benfox"稿件管理员只能完成 5 种管理操作。

假设当前登录用户"pett"为系统管理员,同样,在如图 8-14 所示的登录界面中输入 用户名和密码,也进入稿件数据库管理界面,由于权限不同,因此管理界面和图 8-15 所示 的界面有所不同,如图 8-17 所示。

从浏览器的地址栏中可以看出,图 8-17 和图 8-15 的稿件数据库管理界面都是同一个 asp 文件,但是显示出来的界面有不同之处。由于系统管理员具有更高的权限,因此,在图 8-17 中有"稿件输入情况管理"的超链接,而稿件管理员的管理界面中则没有这个链接。 除此之外,两个下拉菜单中的选项是相同的。但是系统管理员具有更多的权限,系统管理 员和稿件管理员所能完成的操作对比如表 8-1 和表 8-2 所示。

】病件发酵连管理界值。Microsoft Internet Explorer	ner
http://166.111.170.110/document/manager/login_result.asp	
你專會理嗎一个表, 学生作者信息表 漸	
你要对该表进行何种操作。 「漆茄记录 編	
	[稿件输入情况管理]

图 8-17 稿件数据库管理界面 (二)

表 8-1	槁仵管理贞操作权限

	学生作者	教师作者	其他作者		正式稿件	用户	临时稿件
	信息表	信息表	信息表	信息表	信息表	信息表	信息表
添加记录	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	×	×
修改记录	×	×	×	×	×	×	×
删除记录	×	×	×	×	×	×	×

表 8-2 系统管理员操作权限

	学生作者 信息表	教师作者 信息表	其他作者 信息表	编辑 信息表	正式稿件 信息表	用户 信息表	临时稿件 信息表
	~	~	\checkmark	~	~	~	×
修改记录	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	×
删除记录	×	×	×	×	\checkmark	\checkmark	\checkmark

注意: "√"表示有操作权限; "×"表示没有操作权限。

为了更好地说明管理员的管理操作过程,把表 8-2 中的所有可执行操作进行编号,如表 8-3 所示。

表 8-3 管理操作编号

	学生作者	教师作者	其他作者		正式稿件	用户	 临时稿件
	信息表	信息表	信息表	信息表	信息表	信息表	信息表
添加记录	A1	B1	C1	D1	E1	F1	×
修改记录	A2	B2	C2	D2	E2	F2	×
删除记录	×	×	×	×	E3	F3	G3

下面简要介绍完成这些操作的过程。

◆ A1 往学生作者信息表中添加记录。

只需要在图 8-15 或者图 8-17 所示的管理员操作界面的下拉菜单选择相应的选项, 然 后单击"开始操作"按钮,将弹出如图 8-18 所示的界面窗口。

▲加速率	Microsoft Internet Explorer	
	C/ « × B A ··· F · K ··	B- *
inn o	http://166.111.170.110/document/manage	
作者姓名		
用户名:		
通讯地址		
稿件所在	部门, 网络部	
稿件完成	时间, 例, 2001-5-3)
	[继续添加]	
	[返回表管理界面]	
		i En l

图 8-18 添加作者信息

管理员填入必要的信息后,单击"写入数据库"按钮,新的作者就添加到数据库了。 稿件管理员与系统管理员都具有完成这个操作的权限。

◆ B1 往教师作者信息表中添加记录。

和 A1 操作基本相同,只是添加到不同的数据表中,具体操作不再赘述。

◆ C1 往其他作者信息表中添加记录。

和以上两个操作相同。

◆ D1 添加编辑信息表。

和以上操作基本相同,不同的是提交编辑信息的表单项目。

注意:以上操作都是基本相同的,只有一些小的地方不同。在把信息提交到后台 数据库之前,前台有信息检查程序对所提交的信息进行限制,避免出现用户名重 复。

◆ E1 往正式稿件信息表中添加记录。

在管理员操作界面中选择相应的下拉菜单选项,进入添加界面,如图 8-19 所示。



图 8-19 添加正式稿件信息

这里所显示的是临时稿件列表,同样采取分页显示,管理员只需单击每一篇稿件后面的"添加入库"链接,就可以把当前稿件从临时信息表中添加到正式稿件信息表中。图 8-19 中显示出总共有1篇新提交的稿件,单击"添加入库"链接后的窗口如图 8-20 所示。

】着什信息。Microsoft Internet Ex	plorer
	W IAD HOG
http://166.111.170.110/c	ocument/manager/check_temp1.asp?id=zke
	K A
第二十四日本(1937年7月末1日)	
	2001-3-6
	学生作者稿件 💌
	科培、学习、 电波
	network, study, computer
	[近向表管理更而]

图 8-20 稿件信息确认

在这里,需要系统管理员或者稿件管理员输入稿件编号,修改相关信息,如果确认无 误,单击"提交入库"按钮。同样,在提交入库之前,前台程序会对其中的一些信息进行 检查,主要是检查其中的"校稿编辑"一项,如果这里提交的"校稿编辑"在编辑数据表 中不存在,则不允许完成提交。

编辑信息无误的话,信息将提交到后台的正式稿件信息表中,同时返回完成操作的提示信息,并检查正文是否存在。添加完后,可以返回图 8-19 所示的窗口,继续完成另一篇 新稿件的提交操作,如果没有新稿件,可以进行其他操作。

◆ F1 添加新用户。

添加新用户操作与添加编辑信息操作基本相同,不同的是,在添加新用户时需要设定 用户的权限和级别,如图 8-21 所示。

▲ 添加记录 Successift Internet Explorer	Mar
THE OLD SHE OWN THE SHO	$\sim \mu_{\rm c}$
	- B
1997 Attp://166.111.170.110/document/manager/e 📰 💞	
田古夕	
工作证号,	
· 密码:	
所属部门,网络部	
用户类别。 普通編結 📓	
特权用户, 🖀 📓 系统管理员, 🖀 🔳	
「法法法	
「「「「「」」」	
[返回表管理界面]	3

图 8-21 添加用户操作

从图 8-21 可以看出,需要填入的信息主要有用户名、真实姓名、工作证号、密码。另 外要选择所属部门,下拉菜单中的选项包括杂志社内的所有部门,这里不再详细列举。用 户类别主要有五级,即普通编辑、专业编辑、专家、部门领导、社内领导,对于不同的用 户身份应该具有不同的级别选择。最重要的信息就是设定当前用户是否具有高级权限,也 就是稿件管理员和系统管理员两种高级权限。

填入了正确的信息后,单击"写入数据库"按钮就可以了。

注意:在前面的权限用户查询模块中,讲述过如何申请一个新用户,主要是用户 把申请信息发送给系统管理员,系统管理员经过确认后,就可以通过图 8-21 所示 的添加界面,把用户添加为正式的用户,并设置其权限。

注意: 下面列出的一些操作, 只有系统管理员才能执行, 稿件管理员不再拥有完成操作的权限。

♦ A2、B2、C2 修改作者信息表。

对于已经添加到作者表中的作者信息,由于信息在不断变化,其中一些可能需要修改, 这时就需要进行修改操作。 进行操作时,只需要在图 8-17 所示的稿件数据库管理界面中,选择相应的表选项和修 改操作选项,然后会出现要求输入作者的用户名的窗口界面,在其中输入需要修改的作者 的用户名后,就会以表单形式返回作者信息,方便系统管理员修改,修改完毕后,提交入 库就可以了。

◆ D2 修改编辑信息。

由于工作变动,对编辑的信息需要进行改动和更新,需要输入新的信息或者修改原有 的信息。同修改作者操作不同的是,在稿件数据库管理界面选定相应的表选项和操作选项 后,系统会返回所有表中的所有编辑信息列表,直接单击"修改链接",就会以表单形式返 回选定编辑的详细信息。修改完后,提交入库就可以了。

◆ E2 修改正式稿件信息。

对于稿件信息的修改和作者信息的修改操作大体是一样的,输入稿件的编号后,系统 同样以表单形式返回选定稿件的所有信息。修改完成后,提交入库。

◆ F2 修改用户信息。

用户信息的修改操作和编辑信息的修改操作相同,这里不再赘述。

◆ E3 删除正式稿件信息。

日积月累,正式稿件信息表中的信息可能过多,这时需要按照年代进行删除。删除的 界面如图 8-22 所示,只需要输入年份和稿件选项,然后单击"删除稿件"就可以了。



图 8-22 删除正式稿件信息

◆ F3 删除用户操作。

选择删除用户操作后,系统将返回数据库中的所有用户列表,如图 8-23 所示。

从图 8-23 可以看出,所有用户是按用户的姓氏拼音来排序的,并且列出了用户的简要 信息,如果确认删除的话,单击相应用户信息后的"删除"就可以了。

】 用FEMIK:Merrosoft Internet Explorer	and a second
	的复数 行 计分析时
CERTIFICATE DATE OF A DESCRIPTION OF A D	Sale of the second second
http://166.111.170.110/document/inanager/del_user.asp	
用户表中共有3位用户,接戴氏斜音排序如下表	
总共1页,这是第1页	
L CARE THE STORE	
1 00011 30 20020.0	
I NE EN	
<u>01</u>	

图 8-23 用户列表

为了保证信息不被误删除,单击"删除"后将弹出确认窗口,如图 8-24 所示。



图 8-24 确认删除

如果不能确认,单击"取消"就可以了,因为一旦删除之后,用户的信息就不能再恢 复,只能重新输入。

◆ G3 删除临时稿件信息。

当稿件信息提交到正式稿件信息表之后,就没有在临时稿件信息表中存在的必要,故 需要删除。删除操作界面如图 8-25 所示。

在稿件数据库管理界面中选择临时稿件信息删除操作后,系统将返回临时稿件信息表 中的所有稿件信息,如图 8-25 所示。

从图中可以看到,系统提供了完备的删除方式,稿件信息以分页方式显示,并且管理 员可以选择"按提交时间先后"和"用户名"两种方式排序,可以删除其中的某一篇稿件 信息,也可以全部删除,而且还提供删除时是否删除正文文件选项。

到此,稿件数据库的管理操作就全部介绍完了。可以看出,稿件数据库的管理模块是 4 个模块中最复杂、最完善的一个,当然,相应的程序编写也会更加复杂。使用这个稿件 数据库的管理系统,可以很方便地完成所有的管理操作。

.共有6篇提 .共1页,	这的稿件(这是第1页	接提交时间先后 [非序 <u>按用户名推升</u>	Ε).			
							AS RES
						A state of the second s	
				5.4.5		T MARK	
		ning han an ing the second			I MANER IN COLUMN AND C		80374180

图 8-25 删除临时稿件信息

下面来整体回顾一下稿件数据库的结构。整个稿件数据库的结构如图 8-26~8-30 所示。



图 8-27 公共查询模块结构



图 8-28 权限用户查询模块结构



图 8-29 稿件信息输入模块结构



图 8-30 稿件数据库管理模块结构

在阅读完稿件数据库系统的介绍后,不知读者有何看法。如果要读者去开发这样的一 个系统,是否心里有数呢?是否知道如何安排工作进度,如何完成关键程序的编写?其实 在本章的开头也提过,这个稿件数据库系统也是一个 Web 数据库系统,因此它开发的过程 和一般的 Web 数据库系统的开发是一样的。

在本书第1篇中已经有了一个完整的 Web 数据库的开发实例。在这里向读者提一个小小的要求,读者不妨设想一下一个 Web 数据库系统的结构和技术构建,对本书第1篇中的设计开发过程做一个回顾,然后做一个设计,并且和第2篇中笔者的开发过程做一番比较,

这样读者在开发时的体会会更深一些。

本稿件数据库系统的开发符合一般 Web 数据库的开发步骤,首先需要设计后台数据库的结构,这里的后台数据库同样采用 SQL Server 数据库。

8.3 稿件信息数据库结构设计

开发一个 Web 数据库,最重要的一步就是后台数据库的结构设计,而后台数据库设计 原则中最重要的一条就是必须符合整个系统的需要。具体的数据库设计原则详见第1章。 这里主要讲述笔者的设计过程和设计结果。

8.3.1 数据表结构设计

根据前面所述的设计思路,按照6个主题为稿件信息数据库(库名为 document)共设 计了6个数据表:一个稿件信息表、三个作者信息表、一个编辑信息表、一个用户信息表。 下面叙述表结构的具体设计。

1. 三个作者信息表

学生作者信息表(student 表),用于存储稿件作者为在校学生的相关信息;教师作者 信息表(teacher 表),用于存储稿件作者为教师身份或者研究员的作者信息;其他作者信息 表(other 表),用于存储所有不属于前两类的作者信息。这三个表分别负责存储这三类作 者的个人信息。三个表中的字段结构如表 8-4 所示。

字段名	数据类型	长度	主关键字	说明
Name	varchar	8		姓名
Login	char	8	\checkmark	用户名
finish_time	datetime	8		完成稿件时间
Section	varchar	50		稿件所在部门
Address	varchar	50		通信地址
Telcode	varchar	50		联系电话

表 8-4 稿件作者信息表

◆ 字段的确定。

姓名、用户名是最基本的个人信息,其中用户名能够惟一地确定每一个作者,所以选 择它来作为主关键字。另外设定了一个 finish_time 字段来存放作者完成稿件的时间信息。 而 section 字段的设定是为了存放稿件的所属部门。

这里设置的用户名字段是一个十分关键的字段,是区别不同作者的惟一标志,因为作 者的真实姓名有可能相同,但用户名可以设置为不同。另外,从上一节的介绍中已经知道, 作者在上传稿件正文的时候是以自己的用户名为稿件正文的文件名的。有了用户名字段, 也方便系统管理员把一些特殊作者添加到系统的用户表中。 ◆ 字段数据类型的确定。

由于每个作者的姓名、用户名和稿件所投部门都可能不同,所以采用可变长度的字符 型。完成稿件的时间理所当然应采用时间日期类型。

注意:在 SQL Server 中,每个数字和英文字母的长度都是1个字节(byte),而 每个汉字的长度是2个字节。这一点在设置字段长度时应该引起读者注意。

2. 编辑信息表 (editor 表)

该表主要记录了杂志社内所有编辑的主要个人信息,包括的字段如表 8-5 所示。

字段名称	数据类型	长度	主关键字	说明			
name	varchar	8					
editor_id	char	5	\checkmark	工作证号			
section	varchar	50		所属部门			
email	varchar	50		电子邮箱			
phone	char	8		办公电话			
editor_rank	varchar	10		级别			

表 8-5 编辑信息表 (editor 表)

◆ 字段的确定。

类似于作者表,设置了姓名、工作证号、所属部门、级别四个基本信息,其中的级别 分为五级,即普通编辑、专业编辑、专家、部门领导、社内领导。由于要给查询稿件的用 户提供与校稿者的联系办法,所以又设置了 email 和 phone 两个字段。

至于数据类型的确定,也和前面类似。

3. 用户信息表 (userinfo 表)

记录有权访问稿件全文的用户相关信息。包含的字段见表 8-6。

	表 8-6 用户	悟思表(U	serinto 表)		
字段名称	数据类型	长度	主关键字	说明	默认值
username	varchar	20		用户名	
truename	varchar	10		真实姓名	
user_id	char	5	\checkmark	用户编号	
password	varchar	50		密码	
section	varchar	50		所属部门	
privilege	bit	1		稿件管理员	0
administrator	bit	1		系统管理员	0
rank	int	4		用户权限级	0

表 8-6 用户信息表(userinfo 表)

◆ 字段的确定。

为严格管理,除了用户名外,用户在第一次申请时必须同时提供自己的真实姓名。为

惟一确定每位用户,设置了"user_id"(用户编号)字段并将其设为主关键字字段。 "privilege"和 "administrator"标记了该用户是否具有管理员身份。管理员分为两种:稿 件管理员和系统管理员。

注意:前者只具有将作者提交上来的稿件进行处理并添加至正式稿件数据库中的 权限。而后者则具有对数据库的一切权限,为最高级的用户。

"rank"则记录该用户访问稿件正文的权限,即普通编辑、专业编辑、专家、部门领导、社内领导等共五类。

◆ 字段数据类型的确定。

表 8-6 中前五个字段数据类型的确定跟其他表类似,这里主要介绍后面三个字段。 "privilege"和 "administrator"只可能有"是"和"否"两种值,所以采用了 SQL Server 的 bit 型,这种数据类型只有"0"和"1"两个值,是 SQL Server 提供的专门用于处理布 尔型字段的数据类型。

注意: 稿件管理员的 "privilege" 字段值为 1, 而系统管理员的 "privilege" 和 "adiministrator" 两个字段的值都为 1。

由于 "rank" 字段需要记录的是 5 个表示优先级的值,所以这里采用了 int 型(即整数型)。在决定某用户是否有权限访问某篇稿件的全文时,将该字段的值与稿件的保密级别进 行比较。

◆ 字段默认值的设定。

为避免管理员的误操作而使某用户具有不该拥有的权限,将以下三个字段的默认值都 设为 0: "privilege", "administrator", "rank",即最低的权限。

4. 稿件信息表 (document 表)

记录了有关正式稿件的所有信息,包含的字段如表 8-7 所示。

字段名称	数据类型	 长度	主关键字	默认值	说明
Document_id	char	7	~		稿件编号
finish_time	datetime	8			完成时间
author_id	char	6			作者编号
editor_id	char	5			校稿编辑工作证号
chinese_title	varchar	100			中文题目
english_title	varchar	300			英文题目
chinese_keyword	varchar	200			中文关键字
english_keyword	varchar	300			英文关键字
chinese_abstract	text	16			中文摘要
english_abstract	text	16			英文摘要
Document_rank	varchar	20			稿件类别
secret_rank	int	4		4	保密级别

表 8-7 稿件信息表 (document 表)

◆ 字段的确定。

稿件编号惟一确定了每篇稿件,所以设置为主关键字。考虑到要为用户提供根据稿件 完成时间范围进行查询的功能,所以加入了"finish time"字段来存储稿件的具体完成时间。

该表中只包含了作者编号和校稿编辑工作证号,而没有记录作者姓名和编辑姓名。这 是因为,在进行跨表查询时需要这两个字段对作者表和编辑表进行连接,而这两个字段正 是各自表中的主关键字,因此可以分别从作者表和编辑表中准确地检索出稿件作者个人信 息和编辑个人信息。这就避免了前面所提到的数据冗余。虽然作者和校稿编辑的个人资料 都可以跟稿件信息一块存放在稿件信息表中,从而简化查询,但违背了关系数据库的设计 思想,导致了相当多的数据冗余,容易造成数据库各表内容的不一致性。稿件的中英文题 目、关键字和摘要属于必要信息,当然应当包含在表中。

因为稿件数据库的查询界面中需要为用户提供按稿件类别查询的服务,所以表中还包 含了一个"document_rank"字段。这里稿件的类别指的是学生作者稿件、教师作者稿件和 其他作者稿件。

"secret_rank"字段记录的是每篇稿件的保密级别,前面已经提到过,这里就不再多说了。

◆ 字段数据类型的确定。

完成时间依然是采用日期型。各文本字段根据其各自的长度特征适当地采用了字符型 或可变长度字符型。需要特别指出的是中英文摘要字段,因为它们所记录的文字信息相对 较长,所以采用了 text(文本)类型。

5. 临时稿件信息数据库表

保密级别由于需要与用户的权限级别进行比较,所以也采用了整数类型。

除了正式的稿件信息数据库外,为存储作者输入的稿件原始信息,需要另外新建一个临时的稿件信息数据库(库名为 temp_document)。该数据库中只有一个与库同名的数据表: temp_document 表。

temp_document 表包含的字段如表 8-8 所示。

字段名称	数据类型	长度	主关键字	默认值	说明
id	int	6			临时稿件编号
tel	varchar	50			作者电话
address	varcha	200			作者通讯地址
chinese_title	varchar	100			中文题目
english_title	varchar	300			英文题目
chinese_keyword	vachar	200			中文关键字
english_keyword	varchar	300			英文关键字
chinese_abstract	text	16			中文摘要
english_abstract	text	16			英文摘要

表 8-8 临时稿件信息数据库表(temp_document 表)

						(续表)
字段名称	数据类型	长度	主关键字	默认值	说明	
author_id	char	6	~		作者编号	
name	varchar	8			作者姓名	
finish_time	datetime	8			完成时间	
input_time	datetime	8		getdate()	提交时间	
input_ip	varchar	15			IP 地址	
have_add	bit	1		0	当前状态	

◆ 字段的确定。

由于该表保存的是作者提交的原始稿件信息,与稿件信息表(document 表)相比字段 内容设计当然有所不同。前面说过,这些信息是要经过稿件管理员处理才能提交到正式的 document 表中去的。其中"name"字段保存的是作者姓名,这一信息其实最终并不提交至 稿件库中,它只是用来方便稿件管理员对作者的信息进行核对。"finish_time"(完成时间) 虽然也是由作者输入的,但由于作者填写的这一信息往往不太规范,所以稿件管理员有修 改此信息的权力,作者填写的完成时间只是作为参考。

"input_time"和"input_ip"都是出于安全保障而设计的字段,因为必须考虑到某个 有不良企图或者搞恶作剧的人向临时稿件信息数据库提交垃圾信息这类不愉快事情的发 生。为方便系统管理员对始作俑者进行追查,该表中设计这两个字段分别用来记录每一条 稿件记录提交上来时的时间和来源 IP,这将为追查过程提供最重要的线索。"input_time" 字段采用了 SQL Server 的 getdate()函数作为默认值,在每插入一条新记录时便自动向该字 段填入当前时间(精确到秒)。

"have_add"字段负责记录该条稿件记录是否经过稿件管理员的处理被提交到了正式 的稿件库中。其默认值设置为 0,即逻辑值"False",每一条由作者提交上来的稿件记录插 入该表时,该字段便会填入这个值。待稿件管理员处理完这条稿件记录并提交至正式库中 之后,前台与数据库交互的程序就会自动将这个值改为 1,即"True"。

8.3.2 稿件信息数据库各数据表之间的关系

作者表、编辑表与稿件信息表之间都是一对多关系。因为作者表和编辑表中的每一条 记录都有可能在稿件信息表中对应一条或者多条记录。而用户信息表从本质上来说与其余 的表并没有什么特殊的逻辑关系,理论上是应该将它分离出稿件信息库的。但是由于需要 建立的是一个基于 Web 的数据库,其任务主要是为 Internet 用户从网上访问提供后台的数 据支持,用户对数据库的所有访问操作都是通过 ASP 脚本语言来实现的,前台程序在访问 某个数据库之前都必须先打开一个通向该数据库的连接(这个过程在第 1 篇已经详细介 绍),所以,如果将用户信息表从稿件信息数据库中分离出去,另建一个用户数据库,那么 在用户访问稿件信息数据库时,为了验证用户的身份和权限(这一过程在用户访问时会频 繁发生),前台程序就不得不同时建立分别对应两个数据库的连接,这既增加了服务器的负 荷又降低了对用户的响应速度,结果反而得不偿失。所以,综合考虑,最好将用户信息表 归入稿件信息数据库中,这样既不会出现前面所说的情况,也便于管理。 这里需要强调的是,临时稿件信息数据库不存在上面所说的问题。这是因为,作者提 交稿件这一操作并不是每天都发生,通常只是在作者初次提交时操作,故不必考虑对服务 器负荷的影响。并且,这一操作与正式的稿件信息数据库没有什么联系,所以可以把它独 立出来做成另一个库。这样做还有一个明显的好处:与正式稿件信息分开保存,有效地提 高了原始稿件信息的安全性。

下面再谈一谈关系的建立。

虽然建立的是关系数据库,但这个数据库是一个需要通过前台程序来访问的后台数据 库。Web数据库系统最基本的要求之一就是界面的友好性。所以,如果依靠在 SQL Server 中为各个数据表建立关系来控制数据库数据信息的一致性,那么当用户(比如稿件管理员) 对数据库进行的操作违反了一致性要求或者做了别的一些非法操作时,前台程序就会自动 返回 SQL Server 的出错信息,而这些信息对于多数用户来说又是很难懂的,这就造成了系统界面的不友好。

因此,在 SQL Server 中的稿件信息数据库本体里,不对表之间的关联进行设置。而且 在设计表结构时为方便后面的管理,把作者信息分为三个类型的表进行存放。由于关系数 据库在设定外关键字时要求子表(稿件表)的每一条记录在父表(如作者表、编辑表)中 都要有相关联的记录,而稿件信息全部存放在稿件表中,稿件表中只有一部分记录对应着 一个作者表,这就不符合外关键字的定义。所以最好采用另一种方式,就是使用前台的 ASP 脚本程序来进行控制,通过前台 ASP 程序对数据库进行访问获得必要的信息来识别用户进 行的数据库操作是否合法。这样做既保证了数据库信息的一致性(参照完整性),又可以按 照设想为用户定制错误信息,让用户界面变得更友好。

关于如何在 SQL Server 中创建新的数据库和数据表,在本书的第1篇中已有详细的介 绍,这里不再赘述。在建立好数据库后,还要进一步做好数据库的维护和备份工作,制定 一定的备份计划是必要的。

在进行前台程序调试之前,也必须设置系统 ODBC 数据源,否则前台程序无法对数据 库进行访问。

关于 ASP 程序如何维持各表之间的参照完整性,将在8.4 节介绍。

8.4 ASP 程序开发

我们知道,后台数据库是以一定的组织方式来存储数据和信息的。但是直接在数据库 的管理工具中进行查询、修改、删除、新添记录十分不方便,结果也不容易调用,更不适 合网络上多人同时进行操作。而前台 ASP 程序可以在网络上替代数据库管理工具进行查询、 修改、删除等操作,ASP 程序可以动态地生成页面,然后通过网络传输,具有更高的安全 性。

有关 ASP 程序开发,这里仅仅说明笔者开发整个稿件数据库系统的过程和结果。

8.4.1 程序开发准备

在编写 ASP 程序之前,首先要做好准备工作。要检查 ASP 程序的运行环境是否存在, 和后台数据库之间的通信是否畅通。如果没有满足这些条件,需要首先就设置开发环境。 ASP 程序的开发环境是 IIS 4.0 以上,具体设定操作详见第4章,这里不再赘述。

配置好服务器后,需要通过 ODBC 数据源建立同 SQL Server 数据库的连接,也就是要 建立系统 DSN。首先在 SQL Server 中建立新的用户,然后建立连接,定义 DSN 名称。

注意:一般情况下,一台服务器上 ODBC 数据源中有不止一个同数据库的连接, 而每一个连接都对应 SQL Server 中的特定数据库,为了安全,一般的做法是为每 一个数据库建立专门的用户名和密码。

这里笔者建立了两个数据库连接,一个是与正式稿件数据库(document)的连接,DSN 名称是 document; 一个是同临时稿件数据库(temp_document)的连接,DSN 名称是 temp_document。并且为这两个数据库建立了同一个访问账号,login: dido, password: 123。 这在程序中是必要的。

在进行具体代码编写之前,首先要设计好整个系统的结构。这里的结构主要是从技术 角度考虑的,可以参考图 8-26 至图 8-30 所示的结构图。但是要注意这两种结构图之间的 区别。

8.4.2 稿件信息数据库前台程序的编写

整个稿件信息数据库的前台系统主要由以下的几个模块组成:

- ◆ 允许所有匿名用户访问的公开稿件信息数据库查询系统。在这里最多只能查看到 稿件的摘要。
- ◆ 只有授权用户才能访问的稿件信息数据库查询系统。进入该系统需提供用户名和 密码,每个用户都只能下载那些他有权限查看的稿件正文。
- ◆ 作者稿件输入系统。作者通过它输入自己稿件的相关信息,并且将稿件正文电子 版文件上传至服务器。
- ◆ 稿件信息数据库管理系统。这是稿件信息数据库管理人员对稿件信息数据库进行 维护和管理的一套系统。稿件信息数据库管理系统是最复杂的一个部分。

1. 公开的稿件查询系统 (模块 1)

这个模块总共由 query.html, sql.asp, query_result.asp, show.asp 和 show_editor.asp 五 个网页组成,它们都位于 query 目录下。在该杂志社的主页上提供了登录这套系统的超链 接(见图 8-1),所有 Internet 用户都可以匿名登录该系统来查询稿件信息。但最多只能查 看到每篇稿件的摘要,不能看到稿件的正文。

(1) query.html

这是稿件信息的主查询页面。在这里,总共为访问者提供了6个查询条件选项:稿件题目,稿件作者,校稿编辑,关键字,稿件完成时间,稿件类别。其中,稿件完成时间和稿件类别选项分别是以下拉菜单和单选按钮的方式提供的,其余选项都是以文本框方式填

写。所以,当用户在提交查询条件的时候,"稿件完成时间"和"稿件类别"为必选项目。 "稿件完成时间"提供了4个时间段供用户选择:1998年,1999年,2000年,2001年到 现在。"稿件类别"则要求用户必须选择自己所要查询的到底是学生来的稿件还是教师或其 他类型的稿件。其余四个查询条件用户可填可不填,用户不填的项目在检索时就不会被列 入查询条件中。"稿件完成时间"和"稿件类别"这两项是以选择方式提供的,所以在提交 查询时这两个查询条件都各自会有一个被选中的值。

这样设计的原因是,使得检索时至少具有两个查询条件,从而可以强制对查询所有命 中的记录数目进行一定的控制,并且,这两个查询条件对于绝大多数用户而言都是必要的。

当用户输入完自己的查询条件后,单击下方的"开始查询"按钮,用户输入的查询条件便会被提交给 sql.asp 文件进行处理。

关于 query.html 的源代码这里不再列出,其中主要包含一个表单和相应的表单限制程序。

(2) sql.asp

sql.asp 文件是一个纯 ASP 文件,它并不返回一个用户能够从浏览器中直接看到的信息,因此在上面的介绍中,读者看不到这个文件的执行。该程序只负责完成一些后台任务:接收用户从 query.html 提交上来的查询条件,并对其进行适当的整理,构成一条 SQL 查询语句,然后将该语句作为一个字符串,并将其作为一个 cookie 写入客户端,接着对页面进行重定向,执行 query_result.asp 文件。

源代码如下:

<%

'本文件用于接收用户提交的查询条件来生成相应的 SQL 查询语句。

'接收用户提交的查询条件。

dim title,auth,tut,finitime,key1,key2,key3,ran,begtime,endtime,relation

relation=request.form("relation")

begtime=""

endtime=""

titl=""

auth=""

tut=""

finitime=""

ran=""

titl=request.form("title")

auth=request.form("author")

tut=request.form("tutor")

finitime=request.form("finish_time")

key1=request.form("key1")

key2=request.form("key2")

key3=request.form("key3")

ran=request.form("rank")

'根据用户选择的查询时间范围生成查询所需的起始时间和截止时间。

select case finitime

case "7"

begtime=#1/1/1998#

endtime=#12/31/1998#

case "8"

begtime=#1/1/1999# endtime=#12/31/1999#

case "9"

begtime=#1/1/2000#

endtime=#12/31/2000#

case "0"

begtime=#1/1/2001#
endtime=date()

end select

dim exec

'指定 SQL 查询语句。

exec="select

document.chinese_title,document.document_id,document.finish_time,"&ran&".name,editor.name,editor_id from "_

&"document,"&ran&",editor where"__

&" (document.author_id="&ran&".login)"_

&" and (document.editor_id=editor.editor_id)"

'&" and (document.finish_time between ""&begtime&"" and ""&endtime&"")"

'判断用户填写了哪一项,并将该项列入查询条件中。

if titl∽"" then

exec=exec&" and (document.chinese_title like '%"&titl&"%')"

end if

if relation="and" then

if key1<>"" then

exec=exec&" and (document.chinese_keyword like '%"&key1&"%')"

end if

if key2 <> "" then

exec=exec&" and (document.chinese_keyword like '%"&key2&"%')"

end if

if key3 <>"" then

exec=exec&" and (document.chinese_keyword like '%"&key3&"%')"

end if

elseif (key1 "" or key2 "" or key3 "") then

```
exec=exec&" and "
```

```
if key1<>"" then
         exec=exec&"((document.chinese keyword like '%"&key1&"%')"
     end if
     if key2 <>"" then
         exec=exec&" or (document.chinese keyword like '%"&key2&"%')"
     end if
     if key3<>"" then
         exec=exec&" or (document.chinese keyword like '%"&key3&"%')"
     end if
     exec=exec&")"
 end if
 if auth <>"" then
    exec=exec&" and ("&ran&".name=""&auth&"")"
 end if
 if tut<>"" then
    exec=exec&" and (editor.name=""&tut&"")"
 end if
'将最后生成的 SQL 查询语句作为 cookie 返回到客户端,方便随时调用。
response.cookies("query")=exec
'response.write exec
'最后将页面重定向到 query result.asp 列出检索结果。
response.redirect "query result.asp"
```

%>

(3) query result.asp

该文件用来返回查询结果。该文件先读取在 sql.asp 中写入客户端的 cookie 值,即获得 根据用户输入的查询条件生成的 SQL 查询语句,然后根据这个语句从数据库中查出相应的 记录。

为了节省服务器资源,减少用户对服务器资源的占用,在这里对检索所能返回的记录 数目进行控制。当检索到符合查询条件的稿件超过 50 篇时,系统将告知用户:"符合条件 的稿件太多,请缩小查询范围后再重新查询。"

为使整个查询结果页面更为友好,采用了 ADO 中的 Recordset 对象的分页技术,即当 检索得到的记录比较多时(多于 10 篇),就将得到的所有记录进行分页处理。关于分页如 何实现,在第1篇中的 BBS 实例中已详细介绍,这里不再赘述。

在这里,使用客户端的 cookie 对象来保存客户提出的查询要求。其实,从技术实现的 角度来看,使用 session 也是可以的,但是 session 变量有一个过期时间,也就是说,如果 采用 session 变量,将不容易确定 session 的存在时间。过期时间太短的话,可能没有浏览 完一页中的信息 session 就已经过期,如果太长的话,又从一定程度上降低了整个系统的安 全性。
因此, cookie 无疑是解决这个问题的最佳选择。将 SQL 语句直接存放在用户的硬盘上, 需要时直接从用户硬盘读取,既节省了服务器资源,又不受变量寿命的限制。这是因为, 只要不设定 cookie 变量到期的绝对时间,系统就默认为当会话结束时自动清除 cookie,也 就是直到用户离开网站或将这个浏览器窗口关闭,这个 cookie 才会自动地从用户的硬盘上 清除掉。当然,如果用户回到 query.html 页面,改变原来的查询条件再进行新的查询, sql.asp 文件便会再执行一遍, cookie 中的值(那个 SQL 语句字符串)将会被刷新,重定向到 query result.asp 后执行的就是新的查询。

query result.asp 中的程序源代码如下:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>稿件查询结果</title>

<script language="JavaScript">

function viewItem(lid)

' 定义一个新弹出窗口, 在新窗口中执行 show.asp 文件的 javascript 函数。

{

window.open("show.asp?id="+lid,"","height=400,width=500,resizable=yes,scrollbars=yes,status=no,toolba r=no,menubar=no,location=no");

}

function viewItem2(lid2)

'与上面类似,只是在新窗口中执行 show_editor.asp 文件。

{

window.open("show_editor.asp?id="+lid2,"","height=160,width=700,resizable=yes,scrollbars=yes,status=n o,toolbar=no,menubar=no,location=no");

}

</script>

</head>

<% '下面对检索到的记录进行分页显示。

mypage=request.querystring("whichpage") '先检查用户需要显示的是哪一页的记录。

if mypage="" then '如果为空,说明用户为首次进入该页面,显示出第1页的记录。

mypage=1

end if

mypagesize=request.querystring("pgsize") '获取每页显示的记录数。

if mypagesize="" then '如果为空,则控制每页显示 10 条。

mypagesize=10

end if

%>		
<body background="r</td><td>nanager/bgleaf.bmp" text="#000080"></body>		
<% sql=request.cookies("query")	' 获取在 sql.asp 中写入客户端的 SQL 语句的 cookie。	
rem response.write sql		
set conn=server.createobject("adodb.c	onnection")	
conn.open "dsn=document;uid=dido;p	owd=dido"	
set rs=server.createobject("adodb.reco	rdset") '创建记录集对象。	
rs.CursorLocation=3		
rs.open sql,conn,3,3	'打开记录集,执行 SQL 语句。	
if rs.EOF then	'如果检索不到符合条件的记录。	
%>	' 对不起,没有您要查找的稿件。	
<%	'如果检索到的记录多于 100 条。	
elseif rs.recordcount>100 then		
%>	' 符合条件的稿件太多,请缩小查询范围再重新查询。	
<% set rs=nothing	'释放记录集。	
else	' 如果检索结果少于 100 条。	
%>		
'	符合查询条件的稿件总篇数:<%=rs.recordcount%>	
<% rs.movefirst	记录集指针往下移动1条记录。	
rs.pagesize=mypagesize	指定每页显示的记录数。	
maxcount=cint(rs.pagecount) '	获得总页数。	
rs.absolutepage=mypage	指定记录集指针当前所在的记录集对象的绝对页数(第几页)为	
	mypage 变量的值。	
howmanyrecs=0		
howmanyfields=rs.fields.count-1		
response.write "总共"&maxcount&"页	,这是第"&mypage&"页	
%>		
<div align="center"></div>		
<center></center>		
<table border="1" bordercolor="#FFFFF</td><td>F" cellspacing="0" height="65" width="675"></table>		
>		
<td align="center" bgcolor<="" td="" width="45"><td>="#99CCFF" height="30">序号</td></td>	<td>="#99CCFF" height="30">序号</td>	="#99CCFF" height="30"> 序号
<td align="center" bgcolor<="" td="" width="95"><td>="#99CCFF" height="27">作者</td></td>	<td>="#99CCFF" height="27">作者</td>	="#99CCFF" height="27"> 作者
<td align="center" bgcold<="" td="" width="418"><td>or="#99CCFF" height="27">稿件题目</td></td>	<td>or="#99CCFF" height="27">稿件题目</td>	or="#99CCFF" height="27"> 稿件题目
<td align="center" bgcold<="" td="" width="100"><td>or="#99CCFF" height="27">稿件编辑</td></td>	<td>or="#99CCFF" height="27">稿件编辑</td>	or="#99CCFF" height="27"> 稿件编辑

```
</font>
 <font color="#800080">完成时间
 </font>
                              按前面设定的每页大小来显示检索结果。
 <%
                             ٠
 do while not rs.EOF and howmanyrecs<rs.pagesize %>
 <font size="2"
 color="#0000FF"><%=rs.AbsolutePosition%></font>
 <font size="2"
 color="#0000FF"><%=rs(3)%></font>
 <a href="javascript:viewItem('<%=rs("document_id")%>')">
 <font size="2" color="#0000FF"><%=rs("chinese title")%></font></a>
 <a href="javascript:viewItem2('<%=rs("editor_id")%>')">
 <font size="2" color="#0000FF"><%=rs(4)%></font></a>
 <font size="2"
 color="#0000FF"><%=rs("finish time")%></font>
 <%
     rs.movenext
    howmanyrecs=howmanyrecs+1
   loop
 rs.close
                       '关闭记录集。
 set rs=nothing
                       '释放记录集。
%>
</center>
</div>
<P align=center>
<% pad="0"
  scriptname=request.ServerVariables("script name")
                                   ' 获取当前 ASP 文件的绝对路径。
                                    '显示各页的链接页码。
 for counter=1 to maxcount
      if counter>=10 then
        pad=""
      end if
      ref="<a
 href=""&scriptname&"?whichpage="&counter&"&pgsize="&mypagesize&"'>"&pad&counter&"</a>"
      response.write ref&" "
 if counter mod 10=0 then
                                     控制使得每行只显示 10 个页码的链接。
```

```
response.write"<br/>
end if
next
%>
</P>
epalign=center><a href="query.html"><font color="#FF6666">重新查询</font></a>
<% end if %>
<% end if %>
<% set conn=nothing
' 释放数据库连接。
%>
</body>
</html>

(4) show.asp
在 query_result.asp 根据查询条件检索出所有符合条件的稿件记录并以列表形式给出
```

时,其实已经将每条稿件记录的稿件编号查出,并且稿件题目是以超链接的形式提供的, 同时提供参数 id,也就是稿件的编号。由于稿件编号是稿件信息表的主关键字,每个编号 惟一确定一篇稿件, show.asp 就是根据这个稿件编号到稿件信息表中去取出这篇稿件的摘 要的。

```
show.asp 中的程序源代码如下:
```

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>稿件摘要</title>

</head>

<body text="#000080">

```
<%
```

```
dim id,sql '从超链接的 Query 字段参数中获取稿件的编号。
```

' 根据稿件编号查出该稿件的题目和摘要。

sql="select chinese_abstract, chinese_title, chinese_keyword from document where

document_id=""&id&"""

id=request("id")

```
set conn=server.createobject("adodb.connection")
```

conn.open "dsn=document;uid=dido;pwd=dido"

set rs=server.createobject("adodb.recordset")

rs.CursorLocation=3

rs.open sql,conn,3,3

</html>

(5) show_editor.asp

阅读完稿件摘要后,如果用户对这篇稿件非常感兴趣,可以与该稿件的校稿编辑直接 联络。单击稿件题目后面的校稿编辑姓名,同时将弹出一个新窗口,这个窗口将执行 show_editor.asp 程序,给出该校稿编辑的个人信息和联系方法。具体实现过程与 show.asp 相同。

show_editor.asp 的源代码与 show.asp 的源代码几乎相同,这里不再重复。 综上所述,公开稿件信息数据库查询系统的基本结构如图 8-31 所示。



图 8-31 公开稿件信息数据库查询

2. 仅供授权用户访问的稿件查询系统(模块 2)

公开的稿件信息数据库查询系统是为所有网络用户提供稿件信息交流服务,由于杂志 社内的所有稿件都具有一定的保密级别,所以从公开查询界面登录的用户都不能查看到任 何一篇稿件的全文。为方便杂志社内部或者授权用户之间的稿件信息交流,需要开发一个 供内部授权用户访问的稿件查询系统。

本模块所有文件都在 advance 文件夹中, 共有 10 个文件: login.htm, login_result.asp, register.htm, register_result.asp, ad_query.asp, ad_sql.asp, ad_query_result.asp, show.asp, link.asp, show_editor.asp。另外还有一个名叫 document 的文件夹, 用来存放所有稿件正文 电子版文件的目录,该文件夹中又有三个名为"students"、"editor"、"others" 的子文件夹,

分别用来存放三类稿件的正文文件。

(1) login.htm

这是用户登录该系统的登录界面,在浏览器中显示的效果可参考前面的图 8-7。本文件 中的源代码也很简单,中间包含表单代码和表单限制程序。源代码在这里不再赘述。

(2) login_result.asp

该文件接收用户在 login.htm 中提交的用户名和密码, 然后在稿件信息数据库的用户信息表中进行查询核对, 如果两项相匹配, 则认为是合法用户, 并将用户名和密码赋值给两个 cookie, 写入用户的硬盘中, 便于在后面的每一个页面中对该用户的身份和权限进行识别。然后重定向至 ad_query.asp 文件, 让用户进入主查询页面。如果验证未通过, 则显示错误信息, 要求重新输入用户名和密码。

cookie 其实是以文本文件的形式存放在客户端的,也就是说,尽管表示 cookie 的代码 非常难懂,但理论上用户最终还是可以直接在机器上修改自己的 cookie 来欺骗服务器,冒 充他人以获取那些自己本来无权访问的资源。所以,最后只将用户名和密码这两项信息作 为 cookie 写入客户端,其余涉及到个人权限验证的相关信息全都通过用户名和密码来从数 据库中获得,而不是从客户端获得,这就避免了安全隐患。

在接收提交来的用户名和密码时,还有一个安全问题,就是利用单引号"'"来屏蔽查 询语句中的查询条件,这个问题在前面的 BBS 部分也有详细介绍。解决办法就是把 SQL 语句中的"'"通过 replace 函数替换掉,保证 SQL 语句中不出现单引号。

<%

' 获取用户提交来的用户名和密码。

dim username, password

username=replace(Trim(request.form("username")),""","@")

'如果用户名中包括单引号,则将单引号替换成@符号。

password=replace(Trim(request.form("password")),""","@")

dim exec

exec="select * from userinfo where username=""&username&" and password=""&password&""

set conn=server.createobject("adodb.connection")

conn.open "dsn=document;uid=dido;pwd=dido"

set rs=conn.execute(exec)

if not rs.EOF then '如果验证通过,则将用户名和密码以 cookie 形式写入客户端。

response.cookies("user")("username")=username

response.cookies("user")("password")=password

set conn=nothing

response.redirect "ad_query.asp"

%> <%

' 如果验证未通过,则提示重新输入用户名和密码。

else%> '显示重新登录的表单,这里省去了表单的代码。

<%end if%>

<%

set rs=nothing

set conn=nothing

%>

(3) register.htm

此文件也是一个表单页面,从中需要输入一些用户的基本资料,当然,也需要相应的 表单限制程序。这里不再赘述源代码。

注意: 在这个页面中还设置了一个隐含提交的参数 "filled",以标记用户已经填写并提交了个人信息。它具有一个"True"值并和个人信息一并提交给 register result.asp.

(4) register_result.asp

该文件负责接收用户从 register.htm 提交的个人信息,并将这些信息写到服务器 document 目录下一个名为 "newuser.txt"的文本文件中。这样,数据库系统管理员只需每 天查看一下这个文本文件,就知道当天有哪些新用户进行了注册,然后分别对他们进行身 份验证。

在刚打开这个页面时,程序将检查那个隐含提交的参数"filled",如果值为"True", 则说明用户是通过在 register.htm 中填写个人信息并提交后才调用这个页面的,否则就是直 接在地址栏中输入本页面的 URL 而调用该页面的,这当然是非法的访问方式,将会被重定 向到 register.htm。

本文件的程序主要是创建一个文件对象,把接受的表单值写入指定路径下的文本文件中。这里不再列出源代码。

(5) ad_query.asp

本文件和模块1中的 query.html 基本相同, 惟一的区别是在文件头部加入了访问控制。 即在程序的开始部分就从客户端获取该用户的 cookie。如果用户名不为空,则说明该用户 是通过了身份验证而进入本页面的, 否则将被重定向至 login.htm; 如果用户名不为空, 但 被系统管理员设定为暂时终止其使用权限的用户, 也将被重定向到 login.htm。

这里仅仅列出文件头部的访问控制程序代码。

if request.cookies("user")("username")="" then '如果用户名 cookie 为空,说明未经过登录而直接 访问本页面。 response.redirect "login.htm" '重定向到登录界面。 end if %> <% '打开数据库,验证该用户是否为非法用户(即被暂时禁用)。	<% '访问控制	,防止用尸未经	登录而直接访问本页面。	
访问本页面。 response.redirect "login.htm" '重定向到登录界面。 end if %> <% '打开数据库,验证该用户是否为非法用户(即被暂时禁用)。	if request.cookies("user")("use	rname")="" then	'如果用户名 cookie 为空,	说明未经过登录而直接
response.redirect "login.htm" ' 重定向到登录界面。 end if %> <% '打开数据库,验证该用户是否为非法用户(即被暂时禁用)。			访问本页面。	
end if %> <% '打开数据库,验证该用户是否为非法用户(即被暂时禁用)。	response.redirect "login.htm"		'重定向到登录界面。	
%> <% 打开数据库,验证该用户是否为非法用户(即被暂时禁用)。	end if			
<% '打开数据库,验证该用户是否为非法用户(即被暂时禁用)。	%>			
	<%	'打开数据库,	验证该用户是否为非法用户(即被暂时禁用)。

```
set conn=server.createobject("adodb.connection")
```

conn.open "dsn=document;uid=dido;pwd=dido"

dim username, password, sqlcheck

username=request.cookies("user")("username")

password=request.cookies("user")("password")

sqlcheck="select rank from userinfo where username=""&username&"' and password=""&password&"'" set user rs=conn.execute(sqlcheck)

if user_rs("rank")<0 then '如果该用户的权限级别小于 0,说明为非法用户,重定向到登录界面。

set user_rs=nothing

set conn=nothing

response.redirect "login.htm"

else

set user_rs=nothing

set conn=nothing

end if

%>

(6) ad_sql.asp

其作用和代码与模块1中的 sql.asp 相同。区别也是在文件头部加入了访问控制。

(7) ad query result.asp

同模块 1 中的 query_result.asp,用于显示检索结果。区别也是在文件头部加入了访问 控制。

(8) show.asp

作用与公开稿件信息数据库查询系统中的 show.asp 大致相同,主要用于显示稿件的摘要。区别有两点:

- 文件头部加入了访问控制。
- 根据 cookie 从数据库的用户信息表中查出该用户的职称、部门和工作证号等信息, 再查出当前这篇稿件的保密级别,两组信息相互比较,从而判断该用户是否具有 查看此稿件全文的权限,如果有,则显示稿件正文的下载链接(链接目标为 link.asp, 并同时传递两个分别记录此稿件编号和类别的 Query 字段参数),否则将不显示此 链接。

(9) link.asp

这是一个执行下载任务的程序。编写这个程序的主要目的是为了维护系统的安全。网络上,文件传输主要有两种协议: FTP 和 HTTP。

FTP 方式要求服务器端启动 FTP 服务,并把所有要提供下载服务的文件放到 FTP 服务的发布目录下。只需在下载链接中直接指定要下载文件的 URL(以"ftp://"开头)便可。

但是,FTP 方式对我们这套系统有一个致命的安全缺陷:从访问者权限角度而言,这种方 式只有匿名访问和拒绝匿名访问两种访问控制,无法满足安全需要(因为要求每篇稿件对 每个用户都有不同的权限)。也就是说,假如采用匿名访问方式,那么用户完全可以不通过 我们在 show.asp 中提供的下载链接,而直接在浏览器地址栏中输入下载文件或文件目录的 URL 来直接下载文件。更可怕的是,甚至整个 FTP 发布目录中所有稿件正文文件都会轻易 地暴露给用户。假如我们拒绝用户以匿名方式登录 FTP 服务器,虽然可以阻止用户直接输 入 FTP 服务器发布目录的 URL 来进行访问,但这样又会导致用户在单击 show.asp 中提供 的下载链接时弹出 NT 的用户验证窗口,不仅要求用户在这里再输一次 NT 的用户名和密 码,而且还得在 FTP 服务器上再将用户信息表中所有用户注册为 NT 的用户,既麻烦又很 不友好,又造成了两份用户表,维护起来十分麻烦。

如果是 HTTP 方式,则只需将要提供下载的稿件正文文件放至 Web 站点发布目录下的 任一级目录中,同样可通过其 URL 进行下载。采用 HTTP 方式有个最大的优点:如果不允 许目录浏览,那么用户输入站点的 URL 就不能看到站点发布目录中的文件。虽然用户仍然 是以匿名访问方式来下载,但是可以采取一些特定手段来隐藏稿件正文存放的路径信息。 当然,还可以把正文文件和稿件信息一块存放在稿件信息数据库中,这样从逻辑上来说显 然更直观,但同样也会带来问题:使数据库体积变得庞大,检索时将严重降低数据库效率; 无法对正文文件进行单独的备份和管理。

在此,笔者采用 HTTP 协议,这就使得隐藏稿件正文路径变得尤其重要。

为了隐藏稿件正文的路径信息,用户单击检索结果列表中的稿件题目链接后,弹出的 新窗口中没有地址栏和状态栏,从而使得用户单击下载链接后至少不会明显地从浏览器上 直观地看到下载链接的 URL。但这不起任何作用,因为要得到链接的 URL 地址很容易, 如果直接将待下载正文文件的 URL 赋给该链接,那么正文文件的 URL 就会暴露。

link.asp 文件的主要作用是将页面重定向到待下载正文的 URL。从 show.asp 页面给出 的下载链接中将会传递该下载文件的稿件编号(也就是文件名)和稿件类别这两个 Query 字段参数,然后 link.asp 根据这两个参数来确定应该将页面重定向到哪里。因为 link.asp 是 一个 ASP 文件,它是在服务器端执行的,所以这个重定向过程是在服务器端完成的,用户 在客户端看不见此过程。在测试过程中,使用了"网络蚂蚁"、"网络吸血鬼"和"网络蚊 子"等流行的下载工具进行试验,看到的都只是 link.asp 的 URL 和那两个 Query 字段参数, 正文文件的真实路径没有暴露。

除了隐藏文件真实路径外,在这个页面的头部还要有一段代码访问安全控制,不允许 在地址栏中输入 URL 来直接访问。同时,在进行重定向之前,也要将用户的个人信息与稿 件保密级别进行比较,以确定该用户是否真的有权下载。

文件的源代码如下:

```
<%
```

'访问控制。

```
if request.cookies("user")("username")="" then
  response.redirect "login.htm"
```

```
end if
```

%>

```
<% if request.cookies("user")("rank")<0 then</pre>
        response.redirect "login.htm"
     end if
 %>
 <html>
 <head>
 <title>下载稿件正文</title>
 </head>
 <body>
 <% set conn=server.createobject("adodb.connection")</pre>
     conn.open"dsn=document;uid=dido;pwd=dido"
                                                     '从数据库中查出当前用户的权限信息。
    dim username, password, sqlcheck
        username=request.cookies("user")("username")
        password=request.cookies("user")("password")
        sqlcheck="select section,rank,user id from userinfo where username=""&username&" and
password=""&password&"""
        set user rs=conn.execute(sqlcheck)
   if not user rs.EOF then
                                  '如果用户表中有该用户,再去查出保密级别。
dim id,kind
    id=request("abc")
                                  '从下载链接的 Query 字段参数中获得文件名。
    kind=request("rank")
                                 '从下载链接的 Query 字段参数中获得该稿件的类别。
dim sql1,sql2
    sql1="select secret_rank,author_id,editor_id,document_rank from document where
           document_id="%id&""
           set rs=conn.execute(sql1)
    sql2="select section from "&rs("document_rank")&" where login=""&rs("author id")&"""
           set auth_rs=conn.execute(sql2)
   dim secret_rank,editor_id,document_rank,author_section
    secret_rank=rs("secret_rank")
    editor_id=rs("editor_id")
   'author_section=auth_rs("section")
  set auth_rs=nothing
  set rs=nothing
                                '根据稿件的类别到不同的目录下找出该稿件的正文文件。
  if secret rank=0 then
             select case kind
                       case "undergraduate"
                            response.redirect "document/student/"&id&".zip"
                       case "graduate"
                            response.redirect "document/editor/"&id&".zip"
```

case "doctor"

response.redirect "document/others/"&id&".zip"

end select

elseif Trim(editor_id)=Trim(user_rs("user_id")) then

select case kind

case "undergraduate"

response.redirect "document/student/"&id&".zip"

case "graduate"

response.redirect "document/editor/"&id&".zip"

case "doctor"

response.redirect "document/others/"&id&".zip"

end select

elseif Trim(author_section)=Trim(user_rs("section")) then

if rs("secret_rank")<4 then

if cint(user_rs("rank"))>=cint(secret_rank) then

select case kind

case "undergraduate"

response.redirect "document/student/"&id&".zip"

case "graduate"

response.redirect "document/editor/"&id&".zip"

case "doctor"

response.redirect "document/others/"&id&".zip"

end select

end if

end if

else

```
response.write("对不起,您没有下载此稿件的权限!")
```

end if

else

```
set user_rs=nothing
```

set conn=nothing

response.redirect "login.htm"

end if

```
set user_rs=nothing
```

set conn=nothing

%>

</body>

</html>

注意:采用这种下载方式,可以完全防止用户找到正文的实际路径,从而起到安 全保护作用。这种方法同样可以在开发其他 Web 数据库系统时采用。

(10) show editor.asp

与模块1的 show_editor.asp 相同。

3. 作者稿件输入系统(模块3)

随时都可能有作者将新的稿件发送到稿件信息数据库中。如果所有稿件的信息都由稿 件管理员输入,这将是一项非常艰苦的工作。所以,最佳的考虑是让作者通过网络来输入 自己的稿件信息,这样既提高了效率又减轻了稿件管理员的负担。

作者稿件输入系统总共由三个页面文件组成: author_input.htm, add_to_temp.asp, upload.asp。它们存放在 document 目录下的 document_input 文件夹中。

(1) author input.htm

包含输入稿件以及稿件作者详细信息的表单。同样有表单限制程序。

(2) add_to_temp.asp

处理用户输入的表单信息,将这些信息写入临时稿件信息数据库。 在将信息写入临时稿件信息数据库前,程序将自动进行相关的验证:

- ◆ 检查参数 "filled" 是否为 "True"。如果是,则说明用户是先在 input.htm 页面输入了稿件信息并提交后才进入本页面的。如果不是,则说明用户是通过在浏览器 地址栏中直接输入本页面的 URL 来访问的,这是非法访问,将被重定向。
- ◆ 检查临时稿件表中是否存在相同用户名的稿件,如果存在,则拒绝处理刚才提交的稿件信息,并提醒用户立刻与系统管理员联系。如果不存在,则进行下一项检查。
- ◆ 检查"稿件完成时间"一项是不是合法的日期格式字符,如果不是,则拒绝处理, 并提示用户返回修改前面输入的日期。如果输入日期合法,则将所有提交的稿件 信息插入稿件信息数据库,然后要求用户上载稿件正文的 ZIP 文件(为便于管理, 要求以用户名命名)。

文件的源代码如下:

```
<html>
```

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>提交成功</title>

</head>

<body >

<% if request.form("filled") <> "True" then '如果 input.htm 中表单项 filled 值不为 True,说明为正常合法用户。 response.redirect "student input.htm" end if dim login ' 接收表单传递过来的用户名。 login=request.form("login") '将用户名生成 cookie 返回到客户端。 response.cookies("login")=login set conn=server.createobject("adodb.connection") conn.open "dsn=temp_document;uid=users;pwd=bluesky" sql="select login from temp_document where login=""&login&""" set rs=conn.execute(sal) if rs.EOF then '进一步检查提交过来的完成时间格式是否合法,如果不合法则要求返回重填。 if not IsDate(request.form("finish time")) then %> <>>请输入有效的时间后再提交! <% else '如果时间格式合法,则执行以下操作接收 author input.htm 表单传递过来的稿件信息。 dim chinese title,english title,tutor name,finish time,chinese keyword,english keyword,chinese abstract,english abstract dim et,ek,ea,ea1,ca1,who,name name=request.form("name") chinese_title=request.form("chinese_title") english title=request.form("english title") '将接收来的英文题目中的单引号换成两个单引号,以免系统错误解释 SOL 语句。 et=replace(english_title,""","""") tutor name=request.form("tutor name") '因为传递过来的完成时间是字符型,这里把它转换成日期型。 finish_time=FormatDateTime(request.form("finish_time"),vbShortDate) chinese keyword=request.form("chinese keyword") english_keyword=request.form("english_keyword") ek=replace(english keyword,""",""") chinese abstract=request.form("chinese abstract") '将英文摘要中的回车符和空格符换成 html 格式的换行符,否则从网页显示出来将没有换行和空格。 cal=replace(chinese_abstract,Chr(13)&Chr(10),"
") ca1=replace(ca1," "," ") english abstract=request.form("english_abstract") ea=replace(english_abstract,""","""")

```
eal=replace(english abstract,Chr(13)&Chr(10),"<br>")
   ea1=replace(ea1," "," ")
                                          '记录下提交稿件信息的 IP 地址。
   who=request.serverVariables("REMOTE ADDR")
                                          '指定插入操作的 SOL 语句。
dim add
 add="insert into temp_document (chinese_title,english_title,tutor_name,finish_time,chinese_keyword,"_
 &"english_keyword,chinese_abstract,english abstract,login,input ip,name)
 values("%chinese_title&"',"%et&"',"
 &"""&tutor_name&"",""&finish time&"",""&chinese keyword&"",""&ek&"",""&ca1&"","
         &"""&ea1&"','"&login&"','"&who&"','"&name&"')"
  ' response.write add
   conn.execute(add)
    response.write("您输入的信息已经成功提交!")
 %>
 <font size="2">
 请将您的稿件压缩后上传</font>
 <font size="2">以</font><b><font color="#FF0000" size="3">用户名</font></b><font size="2">作
 为文件名上传</font>。
 <form name="frmUpload" Method="Post" Enctype="multipart/form-data" Action="Upload.asp">
 <font size="3"><b>稿件</b></font><font
 size="3"><b>zip 文件上传</b></font>
     <input type="file" name="FileName">
 <TR>
     <td COLSPAN="2" ALIGN="right" width="302"
 valign="middle">     
         <INPUT TYPE="Submit" VALUE="上载稿件"></TD>
 </TR>
 </form>
 <%
       end if
   else
             '如果数据库中已有该用户名的信息,则给出警告信息%>
 <font size="2">
 '数据库中已存在该用户名对应的稿件,你的稿件不能提交。</font>
 <font size="2">请立即发邮件给下面的地址与系统管理员联系。</font>
```

<% end if %>

<div align="center">

<center>

[返回

]

</center>

</div>

</body>

<% set rs=nothing

```
set conn=nothing%>
```

</html>

(3) upload.asp

接收作者上载的稿件文件,将其放在服务器上一个特定的文件夹中(暂时把文件存放 于站点发布目录下的 temp_document 文件夹中),这个文件夹不一定要求在站点的发布目录 下,它可以是服务器上任一个驱动器(映射的网络驱动器除外)中的文件夹。如果在该文 件夹中已存在一个同名文件,系统将拒绝上载,并向用户提出警告,提示用户立刻通过 e-mail 与系统管理员联系。

在这个程序中,采用了文件上传组件,名叫"aspcnUP"。这里介绍一下关于 asp 组件 的知识。使用 asp 组件可以帮助服务器完成很多任务,而不同的组件一般都会具有不同的 功能,通常都采用 VB 做开发工具。常见的 asp 组件有上传组件、聊天组件、画图组件等 等。网络上也有很多免费的组件源代码,使用这些组件,可以弥补 ASP 程序的不足。本例 中使用的上传组件,也是采用网络上的免费组件。

upload.asp 的源代码如下:

```
<%@ Language="VBScript" %>
<html>
<head>
<title>上传结果</title>
</head>
<body>
<%
If Request.ServerVariables("REQUEST_METHOD") = "POST" Then
```

```
Set objUpload = Server.CreateObject("aspcn.upload") '设定文件最大长度不超过 3M,默认值为 100K。
objupload.maxsize=3000000 '如果目标文件存在,则进行覆盖。默认为 1,表示不能覆盖。
objupload.overwrite=1 '设定上传到服务器的路径,这里必须为绝对路径,默认值为 c:\。
objupload.path="c:\Inetpub\wwwroot\example\document\temp_document\"
'调用 upload 方法保存文件,这里的路径视实际情况而定。
```

```
varResult = objUpload.Upload
```

If varResult="OK" Then '如果 varResult 返回 OK,则文件上载成功。 response.write("您的正文已成功上载!")

' 上传出错。

```
Else
```

response.write varResult '显示错误信息。

response.write("上传失败! ")

End If

Set objUpload=Nothing

End If

%>

[关闭窗口]

</body>

</html>

4. 稿件信息数据库管理系统(模块4)

在整个稿件信息数据库系统的四个子系统中,这个模块是文件最多、功能要求最强的, 程序设计也最为复杂。

稿件信息数据库管理系统总共包含 29 个页面文件。它们都位于 document 目录下的 manager 文件夹中。这些文件分别完成对稿件信息数据库和临时稿件信息数据库中各表的 添加、修改、删除等功能。

在介绍具体程序之前,首先介绍一下此模块的整体程序结构,以帮助读者理清各个文件之间的关系(见图 8-32)。



图 8-32 稿件数据库管理模块程序结构(一)

(1) login.htm

这是用于管理员登录的界面,提供输入用户名和密码的表单和相应的表单限制程序。 这里不再列出源代码。

(2) login_result.asp

login_result.asp 文件中的程序用来接受 login.htm 中提交的用户名和密码, 然后打开数 据库连接, 判断是否为合法用户, 如果是合法用户, 再进一步对登录用户的权限进行判断。 如果是稿件管理员(用户表中"privilege"字段值为 1), 就向客户端写入一个名为"privilege" 而值为"True"的 cookie; 如果是系统管理员,则向客户端写入一个名为"administrator" 而值为"True"的 cookie。往后的所有程序就是根据这两个 cookie 来判断该管理员所能进 行的所有操作的。

身份验证完毕,将显示出管理各信息表的界面,如前面的图 8-15 和图 8-17 所示,稿 件数据库管理模块程序结构如图 8-33 所示。



图 8-33 稿件数据库管理模块程序结构(二)

(3) operation_echo.asp

前面的管理界面提交的两个参数将提交到 operation_echo.asp 文件中,该程序总是先生成一个 cookie 来记录表名,然后将页面重定向至 add.asp。

'访问控制,防止未经过登录而直接访问本页面。 <% if request.cookies("user")("privilege") <> "True" then response.redirect "login.htm" end if %> <html> <head> <title>操作响应</title> </head> <body text="#000080" background="bgleaf.bmp"> <% dim tablename.operation tablename="" operation="" tablename=request.form("tablename") '接收由 login result.asp 提交的管理员要操作的表的名称。 operation=request.form("operation") '接收要执行的操作的名称。 '将表名作为 cookie 返回客户端,以备随时调用。 response.cookies("tablename")=tablename if operation="add" then '如果要进行添加操作,则重定向到 add.asp。 response.redirect "add.asp" elseif operation="del" then '如果要进行删除操作,并且具有系统管理员身份。 if request.cookies("user")("administrator")="True" then if tablename="userinfo" then '如果要删除用户表中的记录,则重定向到 del_user.asp。 response.redirect "del user.asp" elseif tablename="temp document" then '如果要删除临时稿件表中的记录,则重定向到 manager tempA.asp. response.redirect "manager tempA.asp" else '如果要删除其他数据表中的记录,则重定向到 del_ choice.asp. response.redirect "del choice.asp" end if '如果不具有系统管理员身份,则返回错误信息。 else response.write("对不起,您没有删除权限。") end if '如果要进行修改操作,并且具有系统管理员身份。 elseif operation="modify" then if request.cookies("user")("administrator")="True" then '根据不同的表重定向到不同的页面。 select case tablename case "document" '如果是正式稿件表,则重定向到 modify_document.asp。 response.redirect "modify document.asp" '如果是用户表,则重定向到 modify_userinfo.asp。 case "userinfo" response.redirect "modify_userinfo.asp"

case "editor"	'如果是编辑表,则重定向到 modify_editor.asp。
	response.redirect "modify_editor.asp"
case "temp_document"	'如果是临时稿件表,则返回错误信息(因为该表不能 修 改)。
	response.write("临时稿件信息表中的记录不能被修改!")
case else	'如果是作者表,则重定向到 modify_author.asp。
	response.redirect "modify_author.asp"
end select	
else	'如果不具有系统管理员身份,则返回报错信息。
response.wri	te("对不起,您没有修改的权限。")
end if	
end if	

%>

返回表管理界面

</body>

</html>

(4) add.asp

add.asp 程序从客户端获取目标表名的 cookie 和名为 "administrator"的 cookie,进行 权限判断后显示出相应的表单。例如表名是 userinfo,将显示出一个包含了用户信息表所有 字段项目的表单(表名为 document 时例外),管理员可在这里填写某个新用户的相关信息。

不同的表单将被提交给不同的 ASP 程序进行处理。因为数据库中三个作者表的字段名 是完全相同的,所以三个作者表的表单都将被提交给 add_author.asp。 用户表的表单将提 交给 add_user.asp,编辑表的表单将提交给 add_editor.asp。

add.asp 中的程序源代码将不列出。add.asp 中包含一系列判断语句,根据不同的目标表显示相应的表单。下面介绍各个表单的程序。

(5) add_author.asp

add_author.asp 程序用于添加作者,由于稿件的作者分为三类,因此不同类别的作者信 息输入到不同的表中。在程序中,首先根据类别的不同选择相应的目标表,程序首先接受 提交的表单值,然后利用一个"insert" SQL 语句把表单值插入相应的数据表中。完成这样 操作的语句在前面已介绍过,这里不再赘述。

(6) add_user.asp

此文件用来接受 add.asp 中发送的新用户信息,并把这些信息存储到后台数据库的用户 表中,完成添加新用户的功能。这里省去程序源代码。

(7) add_editor.asp

添加编辑信息。程序代码类似于上面的 add_user.asp。

注意: 在向各个表中添加新记录前, 程序都将自动检查表中是否已经存在相同的

记录(即主关键字字段发生冲突),如果已有相同的记录存在,则拒绝添加。

(8) check temp.asp

check_temp.asp 是添加稿件信息模块中的一个程序,该程序检索出临时稿件表中 have_add 字段值为 0(意味着尚未添入正式库)的所有记录,并同时检查每条记录是否在 temp_document 文件夹中拥有相应的稿件正文。因为通常临时稿件表中尚未加入正式稿件 库的记录不会超过 200 条,为了使管理员操作更为直观方便,这里采用了与稿件查询系统 相同的方式,把检索结果以记录分页(每页 10 条记录)形式列出。在每条稿件记录后都有 一个"添加入库"的链接,目标程序为 check temp1.asp。

此文件中的程序包含一条 SQL 语句,以时间顺序列出临时稿件数据库中所有 have_add 字段值为 0 的记录,并且以分页形式显示所有记录。

(9) check temp1.asp

该程序通过"添加入库"链接中的 Query 参数获得稿件作者的用户名,再以此为线索 从临时稿件库中检索出此稿件的全部原始信息,以在浏览器中显示,并且稿件管理员可以 修改"编辑工作证号"、"完成时间"、"稿件类别"和"稿件保密级别",需要管理员输入"稿 件编号"。确认所有信息无误后,将把这些信息存储到正式稿件数据库中,而这一步由 submit.asp 程序来完成。

该程序代码也很普通,根据用户名从数据库中检索出目标记录,然后把这些信息显示 出来。完成这样功能的程序对于读者也很熟悉了,这里不再重复说明。

(10) submit.asp

submit.asp 中的程序先检查各项信息是否正确, 然后再写入正式稿件信息数据库, 同时 把临时稿件表中该记录的 have_add 字段值改为 1。对于正文文件, 则以用户名重新命名, 根据管理员输入的"稿件类别"信息, 从 temp_document 文件夹中把稿件正文复制到 advance 目录下的相应文件夹中。

其中拷贝文件部分的源代码如下:

<% set conn1=server.createobject("adodb.connection")

conn1.open"dsn=temp_document;uid=users;pwd=bluesky"

'在临时稿件表中将该临时稿件记录标记为"已提交入库"。

conn1.execute exec1

'检查临时稿件正文目录中是否有该稿件的正文文件,如果有,则复制一份到供下载的正式稿件正文 目录中去。

set fs=Server.CreateObject("Scripting.FileSystemObject")

if fs.FileExists("e: \temp article\"&author id&".zip") then

select case document rank '如果是学生作者稿件。

case "student"

fs.CopyFile"e: \temp_article\"&author_id&".zip",

"e:\document\u\"&document_id&".zip" '如果是教师作者稿件。

case "graduate"

fs.CopyFile" e: \temp_article\"&author_id&".zip",

"e:\document \g\"&document_id&".zip" '如果是其他作者稿件。

case "doctor"

fs.CopyFile" e: \temp_article\"&author_id&".zip",

"e:\document \d\"&document_id&".zip"

end select

else '如果未发现正文文件,则提示管理员。

%>

请注意:由于作者未上传正文,故刚才的操作仅是向稿件数据库中添加了该稿件的简要信息。
br>

<% end if

set fs=nothing

%>

(11) modify_document.asp

如果要修改的是稿件信息表中稿件的记录,页面将重定向到 modify_document.asp 文件。由于稿件信息表中的记录一般来说非常多,所以管理员只能通过在这个页面中输入该稿件的编号来检索出该稿件的所有信息以进行修改。

因此,在此文件中显示一个接受稿件编号的表单,然后把表单信息发送到 modify_document1.asp。

(12) modify document1.asp

modify_document1.asp 接收 modify_document.asp 传递过来的稿件编号, 然后打开稿件 信息数据库进行查找, 如果发现该稿件, 则以表格形式返回该稿件的所有信息, 并且这些 信息都将显示在表单文本框中, 方便系统管理员直接进行修改。修改完毕后, 将提交给 modify_document2.asp, 该程序负责把表格中所有的信息重新写入数据库中。

此处不再列出源代码。

(13) modify_document2.asp

modify_document2.asp 用来将修改后的稿件信息存储到数据库中,使用一个 "update" SQL 语句来更新数据库中的记录。

(14) modify_userinfo.asp

如果要修改用户信息表,则被重定向到 modify_userinfo.asp。因为用户表中的用户记录 较少,最多只有 100 多条,所以这里把所有用户的记录也分页显示,每页 10 条记录。在每 条记录后都有一个名为"修改"的超链接,管理员只需单击某用户记录后的这个"修改" 链接,就会激活 modify_userinfo1.asp 程序。 (15) modify userinfo1.asp

modify_userinfo1.asp 根据接受到的用户编号,查出该用户的所有信息,并且以表单形 式显示。修改完毕,信息将提交给 modify_userinfo2.asp。

(16) modify_userinfo2.asp

该程序用于更新用户记录。

(17) modify_editor.asp

如果要修改编辑信息表,则被重定向到 modify_editor.asp。整个修改方式和过程基本与 修改用户表相同,程序代码也大体相同,不再重复介绍。由 modify_editor1.asp 找出并显示 编辑的详细记录,修改后的信息提交给 modify_editor2.asp 重新写入库中。

(18) modify author.asp

如果是修改某个作者信息表,则被重定向到 modify_author.asp。由于作者表中的作者 记录通常也比较多,所以修改方式和过程基本上与修改稿件表相同。modify_author1.asp 负 责接收管理员在 modify_author.asp 中输入的作者用户名,根据表名 cookie 从相应表中查出 该作者信息并显示在网页上,再由 modify_author2.asp 接收修改后的信息并重新写入库中。

(19) del user.asp

del_user.asp 用来完成删除用户操作,以分页方式显示出所有的用户列表,并且在每条 用户记录后显示"删除"链接,用来把用户的编号参数传递到 del_user1.asp。

(20) del user1.asp

del_user1.asp 中的程序根据接受到的 Query 字段获取用户的编号,从而将此用户从用 户表中删除。其中的代码如下:

<%

'访问控制,如果不是系统管理员,则要求重新登录。

if request.cookies("user")("administrator") <> "True" then

response.redirect "login.htm"

end if

%>

<%

dim id,exec

id=request.QueryString("id") '从 del_user.asp 的"删除"超链接中的 Query 字段参数获得用户编号。 exec="delete from userinfo where user_id='"&id&""" ' 定义根据工作证号删除该用户的 SQL 语句。

set conn=server.createobject("adodb.connection")

conn.open "dsn=document;uid=dido;pwd=dido"

conn.execute exec

%>

<%

set conn=nothing

%>

从代码可以看出,在浏览器中也看不到这个程序的执行过程,因为上面的程序不返回 任何信息到浏览器中。

(21) manager_tempA.asp

此文件是删除临时稿件数据库程序中的一部分,用来详细列出临时稿件数据库中的所 有记录。首先检索出表中所有记录的主要字段信息,并显示这些稿件是否具有正文文件以 及是否已被添加入库。在每条记录的后面都有一个"删除"链接,单击它就会激活 manager tempB.asp。可以看出,和上面的删除用户操作大体相同。

(22) manager_tempB.asp

此文件中的程序与上面的 del_user1.asp 中的程序很相似,不同的是,manager_tempB.asp 中的程序在删除数据库信息的同时删除正文文件。程序源代码如下:

<%

' 访问控制,如果不是系统管理员,则要求重新登录。

```
if request.cookies("user")("administrator") >"True" then
```

response.redirect "login.htm"

end if

%> <%

' 获取临时稿件作者用户名,根据用户名删除该条记录。

dim id,exec

id=request.QueryString("id")

exec="delete from temp_document where login="%id&"""

```
set conn=server.createobject("adodb.connection")
```

conn.open "dsn=temp_document;uid=users;pwd=bluesky"

conn.execute exec

```
set fs=server.createobject("Scripting.FileSystemObject")
```

'检查服务器上是否有该提交的正文,如果有,则一并删除。

if fs.FileExists("e: \temp_document\"&id&".zip") then

fs.DeleteFile"e: \temp_document\"&id&".zip"

end if

set fs=nothing

%>

<%

set conn=nothing

response.redirect "manage_tempA.asp"

' 自动返回临时稿件记录列表。

%>

(23) del_choice.asp

此文件中的程序用来完成删除稿件信息操作,显示一个表单程序供管理员选择,提供 输入年代的文本框,而且根据下拉菜单选择属于哪种用户群的稿件。在删除稿件的同时也 删除相应的用户记录。表单提交的目标文件是 del_article.asp。

```
(24) del article.asp
```

```
del article.asp 负责执行删除过程。源代码如下:
```

```
<%
```

'访问控制,如果不是系统管理员,则要求重新登录。

if request.cookies("user")("administrator") >"True" then

```
response.redirect "login.htm"
```

end if

```
%>
```

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>删除完毕</title>

</head>

<body>

```
<% set conn=server.createobject("adodb.connection")
```

```
conn.open "dsn=document;uid=dido;pwd=dido"
```

```
dim gyear, delyear, kind, exec, kind1, sql
```

```
gyear=Trim(request.form("year"))&"-1-1"
```

```
'根据系统管理员在 del choice.asp 中输入的年份生成一个完整的从1月1日开始的日期。
```

```
if IsDate(gyear) then
```

```
delyear=FormatDateTime(gyear,vbShortDate) '将日期字符串转换成日期型。
```

kind=request.form("kind")

sql="select finish_time from document where finish_time<=""&delyear&"""

```
set rs=conn.execute(sql)
```

```
if rs.EOF then '如果没有该日期之前的记录,则返回错误信息。
```

'稿件数据库中不存在<%=delyear%>以前的稿件。

```
%>
```

<% elseif kind="all" then

'如果有该日期之前的记录,并且系统管理员选择了删除该日期前的"所有稿件",则执行删除。 exec="delete from document where finish_time<=""&delyear&""

conn.execute(exec)

%>

<%=delyear%>之前的所有稿件已被全部删除。

<% else

'如果有该日期之前的记录,则删除该类稿件。

exec="delete from document where (document_rank='"&kind&") and

(finish_time<='"&delyear&"')"

conn.execute(exec)

%>

注意: 这里列出的只是重要部分的代码。

至此,稿件数据库管理系统中的所有程序都介绍完了,作为开发提高部分,本章省去 了很多可能重复的部分代码,没有列出所有文件中的源代码,而仅仅列出了其中的一小部 分,但是又是很重要的部分。希望读者能注意这一点。

第9章 搜索引擎实例

本章将详细介绍一个搜索引擎开发实例。搜索引擎对于经常上网的读者来说肯定不陌 生。要找到所需的资料但又不知道地址的时候,几乎所有的上网者都会想到搜索引擎。网 络上大大小小的搜索引擎很多,比如 Sina、Sohu 等等。输入关键词之后,大多数情况下都 能很快得到自己的结果。当然,这样的一个搜索引擎开发起来并不是一件简单的事情。

另外一种情况是,浏览者知道需要的资料在哪个网址可以找到,但由于站点上的页链 接错综复杂,无法在很短的时间内找到需要的网址。久而久之,浏览者就会对这个网址失 去兴趣,这个网站上的浏览者也就越来越少了,这对于网站开发者来说是一个很严重的问题。

对于一个正常运转的 Web 服务器来说,它的工作目的就是及时、全面地向浏览者提供 各种各样的信息,因此,服务器上的 Web 文件和内容在不断更新。但是,开发者并不能做 到所有的信息网页都有明显的链接导向,让浏览者轻松找到信息。

要解决这个问题其实很简单,和在网络上有搜索引擎一样,开发者可以开发基于本地 服务器的搜索引擎。这样,浏览者只需要输入搜索关键词,就可以得到相应关键词的所有 网页链接,从而大大方便了浏览者,增加网站的访问人数,提高网站的知名度。

9.1 概述

顾名思义,搜索引擎就是要进行搜索。但是,进行全面的文件搜索将耗费很多系统资源。而且对于一个正在运行的网站来说,同时访问的人数很多,同时进行文件搜索的浏览 者也可能不止一个,这样将耗费大量的系统资源,降低服务器对其他正常访问者的回应速 度。必须解决这个问题,不可能对每一位用户提出的搜索关键词都进行一次全面的搜索。 如何解决这个问题,如何构建本地服务器上的搜索引擎,从下面的实例中可以得到一些启 示。

假如在一台存在很多文档资料的计算机上需要找到一个或者一些文件,而仅仅知道文 件名或者甚至只知道文件所涉及的内容。在这种情况下,可以逐个磁盘、逐个文件夹手工 查找,逐一打开每一个文件,显然,这将会浪费大量的时间。当然,可以使用计算机中的 搜索工具,输入关键词或者文件名来得到所需文件的地址目录。多数计算机使用者都会采 用后一种方法。但是,对于存在成千上万个文件的计算机,全面搜索一次也要花费相当长 的一段时间,而且要耗费很多系统资源。就我们要开发的本地搜索引擎来说,对于浏览者 提交的关键词,必须经过逐一搜索筛选,才能返回浏览者需要的结果,但这必然会耗费服 务器的系统资源。这两者之间的矛盾如何解决呢?我们可以从网络搜索中找到答案。

当浏览者使用 Sohu、Sina 之类的搜索引擎进行搜索时,输入关键词后,在很短时间内 就可以得到需要的结果。等待的时间都是在网络瓶颈上,而不是服务器的回应时间。当我 们向搜索服务器提出一个关键词后,服务器是否立即从网络上进行搜索,然后把搜索的结 果返回来呢?不是这样的。事实上,所有的搜索引擎都有一个庞大的后台数据库支持,事 先按照一定的分类原则把很多关键词的网址资料输入到数据库中,当收到浏览者的搜索请 求后,只是把服务器数据库中的网址资料传输给用户,而服务器与数据库之间的数据传输 是很快的。所以,构建后台数据库就可以解决开发本地服务器搜索引擎中的矛盾。

建立一个关键词与相关路径资料的数据库后,当浏览者提出搜索要求时,服务器直接 从数据库中读取信息发给浏览者,而不再进行全面的文件搜索,既及时准确地传送信息, 又节省大量的系统资源。这样的一个数据库其实也是一个 Web 数据库。

但是,建立后台数据库来支持搜索引擎将带来新的问题,那就是信息资料的更新。对 于一个 Web 服务器,网页文件每天都在进行更新,相应地,数据库中关键词的路径资料将 会变得陈旧,因此必须定期更新。数据库的更新可以设在访问量少的时候,或者利用别的 服务器完成,然后进行系统切换,这样将不会影响 Web 服务器的运行。

通过 ASP 和 SQL Server 可以实现这样的一个数据库系统,下面我们将详细介绍如何构 建和开发这样的一个 Web 数据库系统,如何实现本地服务器的搜索引擎。

9.2 搜索引擎的使用

按照 9.1 节中介绍的开发思想,笔者开发了一个用于本地服务器的搜索引擎。 如图 9-1 所示,本搜索引擎分为三个部分:

- ◆ 关键词搜索(搜索文件)
- ◆ 文件夹搜索
- ◆ 列出所有文件



图 9-1 搜索引擎主界面

下面将逐一介绍其使用方法。

9.2.1 搜索文件(关键词搜索)

图 9-1 所示是 DIDO 搜索引擎的主界面,从窗口中的三个链接中点击第一个链接将进 行关键词搜索,关键词搜索窗口如图 9-2 所示。

] dido - 根案引擎 - Microsoft Internet Explorer
http://166.111.170.110/searchie
请输入欲搜索的字串: dido
◎标题形式 C 文件名形式
请输入搜索条件

图 9-2 关键词搜索界面

关键词搜索界面分为上下两个部分,用户可以在上半部分输入和选择搜索条件,下面 部分用来显示搜索的结果。从图中可以看到,在进行搜索之前需要输入关键词,并且选择 搜索的方式。关键词在单行文本框输入,而搜索方式有两种选择:一种是标题形式,一种 是文件名。前者将返回所有符合条件的文件的标题列表,而后者将返回所有符合条件的文 件的文件名列表。

注意:这里进行的搜索是一种字符串搜索,也就是说,判断一个网页文件是否满 足要求的唯一标准就是文件中是否包含搜索关键词。

下面将进行具体的搜索操作。输入如图 9-2 所示的关键词 "dido",并且选择"标题形式",然后点击"搜索"按钮,这时将弹出如图 9-3 所示的提示窗口。

3 请等待	Microsoft Internet Explorer	ㅋ미ㅈ
		_
	st over total th	
		-1

图 9-3 提示窗口

注意: 在笔者所开发的这个搜索引擎中,支持的后台数据库事先没有存入任何关 键词信息,只有对每一个关键词搜索过后,才存入后台数据库中。因此,对一个 关键词进行第一次搜索,系统将全面地逐一搜索,把结果返回给用户,同时把结 果存入后台数据库中;当进行第二次搜索时,将直接从数据库中读取,而不再重 新搜索。这样设置并不影响此搜索引擎的实用性,因为在投入实用之前完全可以 由开发者输入一些关键词的信息。由于进行全面搜索需要一段时间,因此在这里 需要对用户进行提示。

正如图 9-3 中的提示所述,如果系统很快返回搜索结果,说明数据库中已经存有当前 的关键词信息,否则系统将进行全面搜索,而搜索一遍需要的时间一般为几十秒。从开发 者的角度来看,设置这样的提示窗口将使系统显得更加友好。提示窗口在 20 秒后会自动关 闭。

下面看看"dido"关键词的搜索结果,如图 9-4 所示。

💁 dido - 想家引擎 - Microsoft Internet Explorer		
	e 12-6	
http://166.111.170.110/searchjet/default.asp		entra di Principalita di Santa di Santa Santa di Santa
请输入欲搜系的子串, d1do		
◎标题形式 ○文件名形式		
lind.		
树皮 1:1. 计 / 田		
授款 0100 的结禾:		
🗟 <u>/lunwen</u>		
■ DIDO杂志社		l.
~		
🖾 <u>/searchjet</u>		
■ <u>DIDO搜索引擎</u>		
搜索共用了26秒		
		33
Ø1780		

图 9-4 "dido"关键词的搜索结果

共有两个文件满足要求,分别位于/lunwen 和/searchjet 文件夹下,两个文件的标题分别 是"DIDO 杂志社"和"DIDO 搜索引擎"。另外可以看出,所有的搜索结果都是以超链接 的形式提供的,点击相应的标题链接,将打开相应的文件,比如点击"DIDO 搜索引擎", 将打开如图 9-1 所示的窗口。另外一个页面文件"DIDO 杂志社"在前面也介绍过,见图 8-1。

在搜索结果的最后一行显示了搜索的时间是 26 秒,也就是说进行一次全面搜索所花费

的时间是26秒钟。

如果搜索"dido"关键词时,选择的搜索方式是"文件名形式",得到的搜索结果则如 图 9-5 所示。



图 9-5 "文件名形式"的搜索结果

其实这两种方式的搜索结果是一样的,不同的只是结果的显示方式。图 9-5 中显示的 是符合条件的文件的文件名,点击相应的链接,同样也可以打开相应的页面文件。比较图 9-4 与图 9-5,发现搜索结果不同的地方还有搜索时间,"文件名形式"搜索花费的时间是 20 秒,而"标题形式"搜索需要花费 26 秒,显然后者花费的时间要多,这是因为后者要 对文件的标题进行检查。

结果中出现搜索时间报告,说明当前搜索的关键词是第一次搜索。如果再次进行"dido" 关键词的搜索,将会很快返回搜索结果,如图 9-6 所示,而且没有搜索时间报告。这说明 搜索结果是从数据库中得到,而不是直接搜索得到,也说明关键词"dido"的搜索结果已 经保存到数据库中。



图 9-6 从数据库中读取的搜索结果

从以上叙述可以看出,关键词搜索的功能是找出服务器发布目录下所有包含搜索关键 词的文件,并且按顺序列出它们的地址链接,方便浏览者访问。

下面介绍如何进行文件夹的搜索。

9.2.2 文件夹搜索

在如图 9-1 所示的主界面中点击第二个"搜索子文件夹"链接,将打开"文件夹搜索"的窗口,如图 9-7 所示。

❑文件★提案 Microsoft Internet Explorer 下下下
Reality Martin Martin Martin Constant States
2 × 2 ∧ r × 2
A http://166.111.170.110/searchjet/abc1.asp
请输入欲搜索文件夹名称:
○列出子文件夹 ○ 不列出子文件夹 ◎ 列出所有文件名称
E-alas
请输入搜索条件!

图 9-7 搜索子文件夹窗口

比较图 9-7 和图 9-2 可以看出,搜索子文件夹的界面和搜索文件的界面十分相似,都 是分为上下两个部分,在上面的部分中输入搜索条件,在下面的部分中显示搜索结果。

从图 9-7 中可以看出,有三种搜索方式可以选择:

- ◆ 列出子文件夹。选择此方式将返回所有的子文件夹列表。
- ◆ 不列出子文件夹。选择此方式只返回包含的下一级子文件夹列表,而不返回更低级别的子文件夹。
- ◆ 列出所有文件名称。选择此方式将返回当前文件夹下的所有文件名称列表,不显示任何一个下一级子文件夹中的文件。
- 1. 列出子文件夹

在图 9-7 所示的窗口中输入搜索条件 "searchjet",也就是当前介绍的搜索引擎系统文件所在的文件夹,搜索的结果如图 9-8 所示。

3. 文件未得来 Microsoft Internet Explorer (日本) 文代化: Microsoft Internet Explorer (日本)
请输入欲搜索文件夹名称, searchjet の列出子文件夹C不列出子文件夹C列出所有文件名称
/searchjet下的文件夹及其所有子文件夹如下,
example1 example11 example2 example21 example22
3

图 9-8 "searchjet" 搜索结果(一)

searchjet 文件夹下共有两个子文件夹,即 example1 和 example2,而这两个子文件夹下 又分别有两个子文件夹,即 example11、example12 和 example21、example22。这四个文件 夹下不存在更低级的文件夹。

注意:输入要进行搜索的文件夹名称时,如果文件夹位于根目录下,则直接输入 文件夹名称就可以了;如果不位于根目录下,则需要输入完整的路径名称,例如 searchjet/example1;如果不输入任何字符,默认为根目录,将返回根目录下的所 有文件夹列表。 2. 不列出子文件夹

同样的搜索条件 "searchjet" 下,如果选择 "不列出子文件夹" 选项,则搜索的结果如 图 9-9 所示。

主文件未提案 Microsoft Internet Explorer 图画层
THO MED ANY WHAT I'VE AND
- #8
148100 D http://166.111.170.110/searchjet/abc1.asp
请输入欲搜索文件夹名称, searchjet
○列出子文件夹 ◎ 不列出子文件夹 ○列出所有文件名称
/searchjet下的所有文件夹如下,
C:\Inetpub\www.oot\searchjet\example1 C:\Inetpub\www.oot\searchjet\example2
· · · · · · · · · · · · · · · · · · ·

图 9-9 "searchjet" 搜索结果(二)

比较图 9-8 与图 9-9 可以看出,"不列出子文件夹"搜索方式返回的搜索结果中,只有 example1 和 example2 两个子文件夹,而不列出更低等级的子文件夹。

提示:一般在搜索所需要的文件夹时,首先需要知道目标文件夹的准确名称,但 是有时可能只知道一个大致的名称,这时可以用"不列出子文件夹"的搜索方式 先返回根目录下的所有文件夹,获得文件夹名称后,然后再使用"列出子文件夹" 的搜索方式,进行进一步的详细搜索,以获得目标文件夹的位置。

3. 列出所有文件名称

搜索文件夹的最终目的是为了搜索所需要的文件。当使用上面的两种文件夹搜索方式 获得目标文件夹后,就可以使用这种搜索方式列出所有的文件。具体搜索过程如图 9-10 所 示。

从图 9-10 可以看出,结果中仅仅是"searchjet"文件夹下的所有文件名称列表,但不包括"searchjet"文件夹下的子文件夹名称。

注意:图 9-10 中所列出的文件名称列表,其实就是本搜索引擎系统所涉及到的所 有文件。



图 9-10 "searchjet"搜索结果(三)

这个搜索的结果仅仅是文件的名称列表,而没有以超链接的形式提供,这和 9.2.1 节中 的文件搜索结果不同。这样设计的原因是:在进行文件搜索时,所有的搜索结果都包含搜 索关键词,浏览者有可能需要浏览列出的每一个文件,以超链接的形式列表将大大方便浏 览者;而这里的文件夹搜索则不同,浏览者的搜索目标是一个文件,并不是文件夹下的所 有文件,当浏览者找到目标文件并需要浏览时,根据文件夹的路径很容易就可以得到目标 文件的地址链接。

9.2.3 所有文件搜索列表

在进行文件搜索时,可能没有任何线索可以使用,这就需要提供所有文件列表搜索。 dido 搜索引擎提供了这种功能,只需要在图 9-1 所示的主界面中点击"列出所有文件"的 超链接,就会返回服务器根目录下所有文件的列表,并且是以超链接的形式提供,如图 9-11 所示。



图 9-11 所有文件列表

搜索的结果中列出了 Web 服务器发布主目录下的所有文件和文件夹,点击相应的文件 链接就可以浏览相应的文件,对于浏览器可以打开的文件,将直接打开,例如图 9-11 中的 help.gif。对于不能直接打开的文件,点击后将出现下载文件的对话框,例如图 9-11 中的 mdatabase.zip,如果点击的话,将弹出如图 9-12 所示的对话框。



图 9-12 从服务器下载 mdatabase.zip 文件

对于图 9-11 所示的文件夹列表,点击链接将打开相应的文件夹,返回选定文件夹下的 所有内容,包括子文件夹和文件列表。点击其中的 "searchjet" 链接,得到的搜索结果如图 9-13 所示。



图 9-13 "searchiet" 文件夹列表

图 9-13 所示的结果与图 9-8 至图 9-10 所示的结果一样,反映了"searchjet"文件夹下的所有内容。该结果也是以超链接形式提供的,点击其中的文件和文件夹与在图 9-11 中点击链接的处理方法一样。另外,还提供了"返回上一层文件夹"的链接,点击该链接将返回图 9-11 所示的根目录列表。当然,点击"example1"或者"example2"文件夹,将进一步展开。

通过这样的方式,浏览者可以查看 Web 服务器上的所有文件,但这对于 Web 服务器上 需要保密的文件将是一种威胁。因此,这样的文件不宜于向匿名用户开放,只适合向权限 用户或者系统管理员开放,以便于系统管理员查看服务器上的文件情况,及时进行管理操 作。

至此,dido 搜索引擎的使用就全部结束了,从上面的介绍可以看出,使用这个搜索引 擎,任何一个网络浏览者都可以找到 Web 服务器上发布目录下的任何一个文件,并且浏览 或者下载。当然,在实际操作中不能向访问者公开所有的文件和资料。只需要设置一些文 件或者文件夹的访问权限,就可以限制匿名用户的访问。
注意: 在 4.1.2 节中介绍了如何创建一个虚拟目录。所建立的虚拟目录可以是服务 器本地磁盘或者局域网内的任意文件夹,也就是说,一般情况下虚拟目录和 Web 服务器的主目录并不是同一物理路径。本搜索引擎在进行搜索时,所有的程序都 是在服务器上完成的,搜索过程也是依照物理路径进行逐一搜索的,因此搜索的 结果中将不包括虚拟目录下的文件和文件夹。

9.2.4 dido 搜索引擎总体结构图

dido 搜索引擎系统中共包含 15 个页面文件和两个图片文件,在图 9-10 中有详细的列表,各个页面文件之间的关系以及 dido 搜索引擎的总体结构如图 9-14 所示。



图 9-14 dido 搜索引擎结构图

该结构图主要分为上下两个部分,如图中的方框所示。上面的方框中包含的是 9.2.1 节中介绍的"关键词搜索"部分,首先从主界面进入"文件搜索"部分。上面的方框有两 层含义:一是其中的所有文件都属于一个子系统,二是其中的所有文件都是在框架页面 default.asp 中显示的。在进行文件搜索时,首先在 title.asp 中输入搜索条件,搜索条件将发 送到 abc.asp 进行处理, 主要是看要求进行何种方式的搜索, 如果是"文件标题列表"方式, 将把搜索关键词发送到 search2.asp, 如果是"列出文件名"方式, 将把搜索关键词发送到 search1.asp。这就是"文件搜索"子系统的运行过程。

下面介绍一个"文件夹搜索"子系统的运行过程,这个子系统中涉及到的所有文件如 图 9-14 下面的方框中所示。这里的方框和上面的方框同样有两层含义,不同的是这里的框 架页面文件是 abc1.asp。当在主界面中点击"文件夹搜索"的链接后,将开始运行"文件 夹搜索"子系统,在 title1.asp 中输入搜索条件,提交后,搜索条件将发送到 abc2.asp 进行 处理,根据搜索条件的不同,把搜索关键词发送到不同的页面。如果是选择"列出子文件 夹",将把搜索关键词发送到 allfold.asp,如果是选择"不列出子文件夹",将把搜索关键词 发送到 listsfd.asp,否则将发送到 searchfile1.asp。

最后,总结构图中还有一个最小的子系统,就是所有文件列表,如图中 listdir.asp 所示,当在主界面中选择"列出所有文件"链接时,将自动执行 listdir.asp,列出所有的文件。

笔者在进行此系统的设计时,正是按照这样的总体结构图框架来组织文件的。

9.3 DIDO 搜索引擎的开发

本章开头已经提到过,这里的搜索引擎实际也是一个 Web 数据库系统,因此,它的设 计、开发过程同其他 Web 数据库系统的设计、开发过程是相同的。首先要进行数据库设计, 然后是程序的开发,最后要注意数据库的维护。下面将按照这样的步骤进行设计和开发。

9.3.1 数据库设计

dido 搜索引擎系统中的数据库设计是本书所有实例中最简单的一个,因为它的表结构 非常简单。笔者在这里开发的数据库的名称是 searchresult,其中共包含两个表: searchresult 表与 searchresult1 表。

searchresult 表用来存放以"文件名列表"形式的搜索结果,而 searchresult1 表则用来存放"以文件标题"形式的搜索结果,这两个表中的字段是一样的,如表 9-1 所示。

			and the second	
字段名	数据类型	说明	初值	备注
searchstring	char	 搜索关键词		
searchresult	varchar	捜索结果		最大长度限制
searchtime	datetime	结果得到时间	now()	最新记录插入时间
searchtimes	int	搜索次数	1	访问者的搜索次数
Searchip	char	访问者 IP		提交新关键词的用户地址

表 9-1 searchresult 表

下面对表的设计进行说明。

searchstring 用来存放关键词的字段,因此数据类型为 char (文本类型)。 searchresult 为相应关键词的搜索结果,因此数据类型为 varchar。需要说明的是,由 于搜索的结果可能是一个很长的字段,因此需要最大长度限制。

searchtime 记录搜索结果的最近更新时间,因此数据类型为 datetime(日期时间类型)。 在插入新记录时,这个字段的默认值是 now(),也就是插入时的时间。在对搜索结果进行更 新时,同时也更新 searchtime 字段值。

searchtimes 记录当前关键词的搜索次数,除第一次外,所有关键词在进行搜索时都 直接从数据库中返回信息,每一次存取都进行累加。从这个字段值可以知道哪些关键词是 热门关键词,这个字段的初值设为1。

searchip 用来记录访问者的 IP, 主要防止一些非法访问者提交不正常的关键词, 通过 IP 地址就可以找到提交信息的来源。

设计好数据表的结构后,就可以开始在 SQL Server 数据库中建立数据库。关于建立数据库和创建新的数据表的方法,在第3章中有详细叙述,这里不再赘述。

建立好 searchresult 数据库后,需要在 SQL Server 中为这个数据库建立一个专门的账号, 然后利用系统管理工具中的 ODBC 数据源建立与 searchresult 数据库的联系,并把这个系统 数据源命名为 search。做好这些工作后,就可以编写 ASP 程序了,通过这个数据源接口可 以实现对 searchresult 数据库的读取。

9.3.2 前台程序设计

在编写前台程序之前要做一些准备工作,这在前面的实例中已经讲述过,主要是要设置好 ASP 程序的运行环境,并且选择一定的编辑工具。笔者的 ASP 运行环境是 IIS 5.0,也就是说 ASP 程序只能在服务器端运行,而运行结果是在客户端的浏览器中显示的。

配置好运行环境后,在服务器的发布主目录下建立一个子目录来存放搜索引擎系统的 所有文件,以和别的文件区别。比如,笔者在开发过程中建立的目录就是 C:\Inetpub\wwwroot\searchjet。

建立好目录后,首先需要做好整个系统的规划工作,规划好所有子系统之间的关系。 笔者在开发之前所做的总结构设计如图 9-14 所示。

下面的工作就是按照总体设计步骤逐一调试。

调试 ASP 程序同调试其他程序一样,需要掌握一定的技巧,其中一些技巧在前面已经 介绍过了,这里再作一个小结:

- ◆ 逐步调试。ASP 程序的运行可以分为很多步骤,可以逐步调试。比如,先输出 SQL 语句,然后把输出的 SQL 语句直接拷贝到 SQL Server 中执行,如果顺利通过,那说明这以前的程序步骤是没有问题的,即使有错误,也能从系统报出的错误提示中分析出可能的程序错误。
- ◆ 程序参考。ASP 的默认脚本语言是 VBScript, VBScript 中包含很多函数、语句、 对象及参数,这些不可能都一一记在脑子中,因此在程序开发中必须有比较好的 程序参考手册和一些常用的程序片断。在本书第 10 章列出了一些常用的程序片 断,可以供读者参考。

下面介绍笔者开发 dido 搜索引擎的过程。

1. 系统主界面 welcome.htm

这是引擎系统的第一个页面,其中提供进入所有子系统的链接,是一个纯页面文件, 中间不包括任何脚本,如图 9-1 所示,这里不再列出代码。

2. 关键词搜索系统框架文件 default.asp

default.asp 也是一个纯页面文件,用来构成关键词搜索系统的框架。从下面的代码可以 看出,浏览器窗口被分为上下两个部分,上面部分的默认页面是 title.asp,下面部分的默认 文件是 please.htm。窗口界面如图 9-2 所示。

<html>

<head>

<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>dido--搜索引擎</title>

</head>

<frameset rows="135,*">

<frame name="header" scrolling="no" noresize target="main" src="title.asp">

<frame name="main" src="please.htm" target="_blank" scrolling="auto">

注意:上面的两行黑体代码是很重要的,从中可以看出上下两个部分的默认页面文件,上下两个 frame 的名称分别是 header 和 main,需要记住下面的 frame 名称,因为在后面的文件中会用到。

```
<noframes>
```

<body>

此网页使用了框架,但您的浏览器不支持框架。

</body>

</noframes>

</frameset>

</html>

使用框架页面将使界面更加直观,在上面的框架中输入搜索条件,将在下面的窗口中 返回搜索结果,无论进行何种搜索,所有的结果都显示在一个页面中。这样的设计方便程 序的移植。一般在正式运行的网站上,搜索引擎可能只是整个页面的一小部分,这样只需 要稍微改动就可以了。

3. 搜索条件输入文件 title.asp

这个文件主要用来接受搜索条件。从上面的 default.asp 中可以看出, title.asp 实际是上面部分框架的默认文件, 在浏览器中的效果见图 9-2。

在编写文件时,主要是插入表单组件,包括一个单行文本框和两个单选按钮。单行文

本框用来输入"搜索关键词",两个单选按钮用来选择"搜索方式"。另外,在代码中嵌套 了 JavaScript 程序,当访问者提交搜索程序后,将弹出提示窗口 wait.htm。提示窗口的代码 在下面的步骤中介绍。

title.asp 文件的源代码如下:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>title</title>

注意: 这里定义了一个 JavaScript 函数 hehe(),此函数的作用是打开一个新的浏览 器窗口以在其中打开 wait.htm 文件,并且设置了这个新窗口的一个相关属性,窗 口的高度为 200,宽度为 400。但是这个函数需要激发才会执行。激发事件就是提 交操作,在下面的代码中有相关的语句。

```
<script language="JavaScript">
```

function hehe()

{

window.open("wait.htm","","height=200,width=400,resizable=yes,scrollbars=yes,status=no,toolbar=no, menubar=no,location=no");

```
}
```

</script>

```
<base target="main">
```

</head>

<body>

<form action="abc.asp" method="post" name=searchform target="main">

注意: 表单的提交方式是"post"(提交方式决定目标文件接受信息的方式), 接受信息的目标文件是"abc.asp",表单的名称是"searchform",提交的目标框 架是"main"(也就是在上面的 default.asp 中设置的下面部分的框架, title.asp 中 的信息提交后,将发送到 abc.asp,并且在下面的框架中显示运行结果)。

请输入欲搜索的字串: <input type="text"

size="20" name="SearchString" >

' "搜索关键词"输入文本框。

<input type="radio" value="V1" name="R1">标题形式<input type="radio" value="V2" checked name="R1">文件名形式

'选择搜索方式的单选按钮。

<input

type="submit" value="搜索" onclick="javascript:hehe()">

注意:这里是一个提交按钮,但是其中有一句代码 onclick="javascript:hehe()",当 点击这个提交按钮时将激发一个事件,执行 hehe()函数,打开窗口。

</form>

</body>

</html>

4. abc.asp

这个文件用来接受上面 title.asp 中提交的表单信息,并对其进行分析处理,再把信息 发送到其他文件,进行进一步的处理。文件中的程序很简单,仅仅是一个判断程序,根据 单选按钮的值重新定向页面,如下所示:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>New Page 1</title>

</head>

<body>

```
<% session("searchstring")=request.form("searchstring")
```

'把"searchstring"表单提交的信息赋值给一个 session("searchstring"),以便在后面的程序中调用。

if request.form("R1")="V1" then

response.redirect "search2.asp"

'如果表单"R1"的值为"V1",说明选择的是"标题"搜索方式,应重新定向到 search2.asp,因为 search2.asp 中包含进行"标题"搜索的程序。

else

response.redirect "search1.asp"

'如果表单"R1"的值为"V2",说明选择的是"文件名"搜索方式,应重新定向到 search1.asp,因为 search1.asp 中包含进行"文件名"搜索的程序。

```
end if
```

%>

</body>

</html>

5. search1.asp

search1.asp 是搜索引擎中的核心文件,也是最复杂的文件之一,它可以完成"文件名 形式"搜索。它的整体设计思路如图 9-15 所示。



图 9-15 search1.asp 中的程序结构

在编写一个程序之前,必须先设计出程序的整体框架,然后依照框架进行代码的编写。 注意:调试程序时,有时需要对单独的一段代码进行调试,有时需要整体运行, 但一般遵循从简单到复杂、从整体到局部的顺序。也就是说,应该最先编写程序 的整体流程,然后再详细插入每一段的详细代码。

下面以笔者编写的程序为例,详细介绍如何依照图 9-15 的流程编写相应的程序。

<%

SearchString = session("SearchString")

'从 session("searchstring")获取关键词,这相当于程序整体流程中的①。

taketime=now() '记录程序开始运行的时间,用变量 taketime 来记录。

%> <%

Head = "站内搜索引擎" ' 定义进行搜索的函数和子程序。

Function UnMapPath(Path)

UnMapPath = Replace(Mid(Path, Len(Server.MapPath("/")) + 1), "\", "/")

End Function

' Len()函数返回字符串内字符的数目,或存储一变量所需的字节数。Mid()函数从字符串中返回指 定数目的字符。Replace()函数返回字符串,其中指定数目的某子字符串被替换为另一个子字符串。 Function SearchFile(f,s)

Set fo = fs.OpenTextFile(f)

content = fo.ReadAll

fo.Close

SearchFile = InStr(1, content, S, vbTextCompare) > 0

End Function

- ' 打开文件(f),并且读取其中的所有内容。InStr()函数返回字符串 S 在另一字符串 content 中第一次出现的位置。参数意义分别为: 1,表示开始计算的位数,content 是被搜索的字符串,S 是关键 词字符串,vbTextCompare 是指进行文本比较。
- SearchFile(f,s)是进行文件搜索的核心函数,首先打开一个文件,读取其中的所有内容,赋值给一 个字符串 content,然后比较搜索关键词在 content 里是否出现,如果出现,则表示此文件符合搜索 条件,否则不符合。

Function FolderLink(fd)

vPath = UnMapPath(fd.Path)

FolderLink = "" & vPath & ""

FolderLink = " " & FolderLink

End Function

' 定义 FolderLink(fd)函数,此函数的功能是为结果中的文件夹增加链接,并且在文件夹名称前增加 "文件夹"的图片使结果形象化。

Function FileLink(f)

vPath = UnMapPath(f.Path)

FileLink = "" & f.Name & ""

FileLink = " "& FileLink & ""

End Function

```
' 定义 FileLink(f)函数,此函数的功能是为结果中的文件列表增加超链接,并且在文件名的前面增加
"文件"的图片,同样使结果形象化。
```

```
Sub SearchFolder( fd, s )
found = False
For each f In fd.Files
pos = InStrRev(f.Path, "." )
If pos > 0 Then
ext = Mid(f.Path, pos + 1 )
Else
ext = ""
End If
If LCase(ext) = "htm" Then
If SearchFile( f, s ) Then
```

If found = False Then

```
found = True
```

Response. Write FolderLink(fd) & "<P>"

```
allstring=allstring&FolderLink(fd)&""
```

End If

Response.Write FileLink(f)

allstring=allstring&Filelink(f)

End If

End If

Next

For each sfd In fd.SubFolders

SearchFolder sfd, s

Next

End Sub

%>

定义 SearchFolder(fd,s)子程序,在此子程序中调用了上面定义的四个函数,对目标文件夹中的所有 文件进行全面搜索,返回结果。

注意:函数和子程序只有在调用时才会被执行。从下面才开始<html>标记。

<html>

<head>

```
<meta http-equiv="Content-Type"
```

```
content="text/html; charset=gb_2312-80">
```

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<title>站内搜索</title>

<base target="_blank">

</head>

<body>

<hr>

<% If SearchString <> "" Then

Set Conn=Server.CreateObject("ADODB.Connection")

Conn.open"dsn=search;uid=pett;pwd=123"

sq="select * from searchresult where searchstring=""&searchstring&"""

set rs=Conn.execute(sq)

'打开 search 数据库,执行 SQL 查询语句,从数据库中查询所输入的搜索关键词,然后进行判断, 相当于程序流程中的②。

if rs.EOF then

'如果搜索结果为空,也就是说数据库中没有当前搜索的关键词记录,则开始对所有的文件进行重新搜索。

%>

<%

allstring=""

On Error Resume Next

Set fs = Server.CreateObject("Scripting.FileSystemObject")

Set fc = fs.GetFolder(Server.MapPath("/"))

Response.Write "<H2>搜索 " & SearchString & " 的结果: </H2><P>"

For Each sfc In fc.SubFolders

fold1="/"&sfc.name

```
Set fd = fs.GetFolder(server.mappath(""&fold1&""))
```

SearchFolder fd, SearchString

'调用上面定义的子函数 SearchFolder 进行搜索。

set fd=nothing

Next

根据当前关键词搜索根目录下的所有文件,此段程序相当于程序流程中的③。

insq="insert into searchresult (searchstring, searchresult, searchip)"_

&" values (""&searchstring&"', ""&allstring&"', ""&request.servervariables("remote_addr")&"')"

conn.execute(insq)

```
'把搜索的结果存储到数据库中,此部分相当于程序流程中的④。
```

taketime=3600*hour(now())+60*minute(time)+second(now())-3600*hour(taketime)-60*minute(taketime)-s econd(taketime)

```
利用时间函数计算搜索所耗费的时间。由于记录了程序开始运行的时间,因此很容易计算搜索时间。
```

%>

```
<font color="#FF0000">搜索共用了<%=taketime%>秒</font>
```

<%

```
else
```

```
如果在数据库中找到当前关键词的记录,则执行下面的程序。
```

Response.Write "<H2>搜索 " & SearchString & " 的结果: </H2><P>"

response.write rs("searchresult")

输出数据库中的相关记录,此程序相当于程序流程中的⑥。

sq1="update searchresult set searchtimes=searchtimes+1,

```
searchip=""&request.servervariables("remote_addr")&" where searchstring=""&searchstring&"""
```

conn.execute(sq1)

```
更新数据库中的记录,累计查询次数,记录访问者的 IP 地址,这段程序相当于程序流程中的⑤。
end if
```

```
set rs=nothing
```

set conn=nothing

End If

%>

<hr>

</body>

</html>

注意: ASP 程序中, 一条长字符串需要换行时, 利用'_和'&进行连接, 因此要注意列出的程序(尤其是定义 SQL 语句的程序行)是否有换行标记。

6. search2.asp

此文件中的程序用来进行"标题形式"的搜索, search1.asp 和 search2.asp 这两个文件 包含的程序十分相似,程序的整体结构也是相同的,因此对于相同的部分,这里不再重复 说明。

<%

```
Head = "dido--搜索引擎"
SearchString = session("SearchString")
```

Function UnMapPath(Path)

```
UnMapPath = Replace(Mid(Path, Len(Server.MapPath("/")) + 1), "\", "/")
```

End Function

```
以上的部分和 search l.asp 中相同。
```

```
Function SearchFile( f, s, title )
```

```
Set fo = fs.OpenTextFile(f)
```

```
content = fo.ReadAll
```

fo.Close

SearchFile = InStr(1, content, S, vbTextCompare) > 0

If SearchFile Then

```
pos1 = InStr(1, content, "<TITLE>", vbTextCompare)
```

```
pos2 = InStr(1, content, "</TITLE>", vbTextCompare)
```

title = ""

If pos1 > 0 And pos2 > 0 Then

```
title = Mid( content, pos1 + 7, pos2 - pos1 - 7 )
```

End If

End If

End Function

' 此处定义的 SearchFile 函数与 search1.asp 中的 SearchFile 函数有所不同,这里除对文件进行比较搜索外,还要找出符合条件的文件的标题。实现的方法是取出<TITLE>和</TITLE>之间的字符串,这 几乎是 search2.asp 与 search1.asp 中唯一不同的地方。

```
Function FolderLink( fd )
```

```
vPath = UnMapPath( fd.Path )
```

```
FolderLink = "<A HREF=""" & vPath & "">" & vPath & "</A>"
   FolderLink = "<IMG SRC=folder.gif Align=TextTop> "& FolderLink
End Function
Function FileLink( f, title )
   vPath = UnMapPath( f.Path )
   If title = "" Then title = f.Name
   FileLink = "<A HREF=""" & vPath & """>" & title & "</A>"
   FileLink = "<UL><IMG SRC=file.gif Align=TextTop> " & FileLink & "</UL>"
End Function
                                            ' 定义链接函数。
Sub SearchFolder( fd, s )
   found = False
   For each f In fd.Files
      pos = InStrRev(f.Path, "." )
      If pos > 0 Then
          ext = Mid(f.Path, pos + 1)
      Else
          ext = ""
      End If
      If LCase(ext) = "htm" Then
          If SearchFile( f, s, title ) Then
              If found = False Then
                 found = True
                 Response.Write FolderLink(fd) & "<P>"
                 allstring=allstring&FolderLink(fd)&""
              End If
              Response.Write FileLink(f, title)
              allstring=allstring&FileLink(f,title)
          End If
       End If
   Next
   For each sfd In fd.SubFolders
       SearchFolder sfd, s
   Next
                                               定义子程序供调用。
End Sub
%>
<html>
<head>
<meta http-equiv="Content-Type"
```

```
content="text/html; charset=gb 2312-80">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<title>站内搜索</title>
<base target=" blank">
</head>
<body>
<hr>
<% If SearchString <> "" Then
 Set Conn=Server.CreateObject("ADODB.Connection")
    Conn.open"dsn=search;uid=pett;pwd=123"
        sq="select * from searchresult1 where searchstring=""&searchstring&"""
                                     建立数据库连接,搜索当前关键词。
        set rs=Conn.execute(sq)
  if rs.EOF then
                                      如果数据库中没有记录,则执行搜索程序。
    %>
        <%
    taketime=now()
allstring=""
On Error Resume Next
   Set fs = Server.CreateObject("Scripting.FileSystemObject")
   Set fc = fs.GetFolder( Server.MapPath("/") )
 Response.Write "<H2>搜索 " & SearchString & " 的结果: </H2><P>"
   For Each sfc In fc.SubFolders
fold1="/"&sfc.name
Set fd = fs.GetFolder(server.mappath(""&fold1&""))
   SearchFolder fd, SearchString
 set fd=nothing
                                    将搜索结果输入后台数据库中。
 Next
                                  t
 insq="insert into searchresult1 (searchstring,searchresult,searchip)"
    &" values (""&searchstring&"', ""&allstring&"', ""&request.servervariables("remote_addr")&"")"
      conn.execute(insq)
taketime=3600*hour(now())+60*minute(time)+second(now())-3600*hour(taketime)-60*minute(taketime)-s
econd(taketime)
 %>
 <font color="#FF0000">搜索共用了<%=taketime%>秒</font>
<%
 else
                              ' 如果数据库中有相应的记录,则直接输出结果。
  Response.Write "<H2>搜索 " & SearchString & " 的结果: </H2><P>"
      response.write rs("searchresult")
      sq1="update searchresult1 set
```

searchtimes=searchtimes+1,searchtime=now(),searchip=""&request.servervariables("remote addr")&""

```
where searchstring=""&searchstring&"""
```

```
conn.execute(sq1)

    更新数据库记录。

      end if
    set rs=nothing
    set conn=nothing
End If
</body>
```

</html>

%> <hr>

至此,"关键词搜索"部分的程序就介绍完了,读者不妨回顾一下上面的程序,回想一 下其中的程序结构,用到了哪些函数,有哪些技巧。下面将介绍搜索引擎系统的第二部分 ——"文件夹"搜索。

```
7. abc1.asp
```

abcl.asp 是一个纯页面文件,用来构成"文件夹"搜索系统的框架。

<html>

<head>

```
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=gb2312">
```

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>文件夹搜索</title>

</head>

<frameset rows="150,*">

<frame name="header" scrolling="no" noresize target="main" src="title1.asp">

<frame name="main" src="please.htm">

'从上面的代码中可以看出,上下两个框架中的默认文件分别是 title1.asp 和 please.htm。

```
<noframes>
```

```
<body>
```

此网页使用了框架,但您的浏览器不支持框架。

</body>

</noframes>

</frameset>

</html>

8. title1.asp

这个文件的作用主要是用来输入搜索条件,从上面的 abcl.asp 中可以看出, title1.asp

实际是上面部分框架的默认文件,在浏览器中的效果见图 9-7。

在编写文件时,主要是插入表单组件,包括一个单行文本框和三个单选按钮。单行文本框用来输入"搜索关键词",三个单选按钮用来选择"搜索方式"。

title1.asp 文件的源代码如下:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>请输入欲搜索文件夹名称</title>

<base target="main">

</head>

<body>

<form action="abc2.asp" method="post" name=searchform target="main">

注意: 这里的 form 的属性, 表单中的值将以 "post" 方式发送到 abc2.asp 进行处理, 并且目标框架为 "main", 也就是 abc1.asp 中下面的框架。

请输入欲搜索文件夹名称: <input type="text" size="20" name="SearchString">input type="radio" value="V1" name="R1">列出子文件夹<input type="radio" value="V2" checked name="R1">不列出子文件夹<input type="radio" value="V3" checked name="R1">列出所有文件名称

<input type="submit" value="搜索" onclick="javascript:hehe()">

</form>

</body>

</html>

9. abc2.asp

这个文件用来接受上面 title1.asp 中提交的表单信息,并对其进行分析处理,再把信息 发送到其他文件,进行进一步的处理。

该文件中包含一个简单的判断程序,根据单选按钮的值重新定向页面。

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>New Page 1</title>

</head>

<body>

```
<%
```

session("searchstring")=request.form("searchstring")

'把"searchstring"表单提交的信息赋值给一个 session("searchstring"),以便在后面的程序中调用。 if request.form("R1")="V1" then

response.redirect "allfold.asp"

'如果表单"R1"的值为"V1",说明选择的是"列出子文件夹"搜索方式,应重新定向到 allfold.asp,因为 allfold.asp 中包含列出所有子文件夹的搜索程序。

end if

if request.form("R1")="V2" then

response.redirect "listsfd.asp"

'如果表单"R1"的值为"V2",说明选择的是"不列出子文件夹"搜索方式,应重新定向到 listsfd.asp,因为 listsfd.asp 中包含不列出更低级子文件夹的搜索程序。

else

response.redirect "searchfile1.asp"

'如果表单"R1"的值为"V3",说明选择的是"列出所有文件"搜索方式,应重新定向到 searchfile1.asp,因为 searchfile1.asp 中包含列出文件夹下所有文件的搜索程序。

end if

%>

</body>

</html>

10. listsfd.asp

listsfd.asp 用来列出某一文件夹下的所有文件夹,但不列出更低级的子文件夹。

<HTML>

<BODY>

<%response.write "/"&session("searchstring")%>下的所有文件夹如下:

<% fold1="/"&session("searchstring")

'从 session("searchstring")中获得需要进行搜索的文件夹名称。

On Error Resume Next

Set fs = Server.CreateObject("Scripting.FileSystemObject")

Set fd = fs.GetFolder(Server.MapPath(""&fold1&""))

For Each sfd In fd.SubFolders

Response.Write sfd & "
"

Next

可以看出,仅仅使用一个简单循环就可以列出文件夹中的所有子文件夹。建立一个文件对象,然后直接利用文件夹的子文件夹属性就可以得到子文件夹的名称。

注意:并不列出更低等级的文件夹。运行的结果可参照图 9-9。

%>

</BODY>

</HTML>

11. allfold.asp

allfold.asp 文件中的程序用来列出某个文件夹下的所有文件夹,包括所有等级的子文件 夹。这个程序使用了一个递归算法,在设计这个程序之前,希望读者能够回顾一下上面的 listsfd.asp 文件中的程序,对其中的程序段仔细研究一下,就会明白如何把上面的程序改写 为我们现在需要的程序。

首先介绍一下递归程序的设计思想。我们知道,文件夹是一种树状结构,也就是说, 每一个文件夹下都可能包括子文件夹,而其中的子文件夹可能包括更低一级的子文件夹, 因此,在程序设计中,从根目录开始对每一个子文件夹都继续进行搜索,直到找不到子文 件夹为止。

下面给出利用递归列出所有子文件夹的程序源代码。

<%

```
Sub ListFolder( fd )
                                      ' 定义 ListFolder 子程序, fd 为一 Folder 对象。
   Response.Write "<UL>"
                                      ' HTML 格式调整语句。
  For Each sfd In fd.SubFolders
     Response.Write sfd.Name & "<BR>"
                                      '列出 fd 的下一级子文件夹 sfd。
     ListFolder sfd
                                      '以子文件夹 sfd 为参数进行递归调用。
     Next
     Response.Write "</UL>"
   End Sub
%>
<HTML>
<BODY>
<% response.write "/"&session("searchstring")%>下的文件夹及其所有子文件夹如下:
<%fold1="/"&session("searchstring")
   On Error Resume Next
   Set fs = Server.CreateObject("Scripting.FileSystemObject")
   Set fd = fs.GetFolder( Server.MapPath(""&fold1&""))
'建立文件对象,调用子程序 ListFolder, 搜索出 fold1 下的所有文件夹。
  ListFolder fd
%>
</BODY>
</HTML>
```

12. searchfile1.asp

searchfile1.asp 是文件夹搜索的第三种方式,利用下面的程序将列出一文件夹对象下的 所有文件,但不包括子文件夹中的文件。

<HTML>

<BODY>

<% response.write "/"&session("searchstring")%> 文件夹的所有档案

<%

fold1="/"&session("searchstring")

On Error Resume Next

Set fs = Server.CreateObject("Scripting.FileSystemObject")

Set fd = fs.GetFolder(Server.MapPath(""&fold1&""))

For Each f In fd.Files

Response.Write f.Name & "
"

Next

'建立文件对象,利用循环语句列出文件夹对象下的所有文件。

%>

</BODY>

</HTML>

至此,搜索引擎系统中的第二个部分——"文件夹搜索"就介绍完了,其中最重要的 是递归算法,这一点应该引起读者的注意。在处理多维对象时,一般都需要使用递归算法。

13. listdir.asp

listdir.asp 是本搜索引擎中最后一个部分中惟一的一个文件,中间也是利用递归算法, 但不是一次列出所有的文件,而是列出同一文件夹下的子文件夹和文件及其链接。点击其 中的文件夹链接,将列出详细内容,点击文件链接可以浏览文件的内容。

源代码如下:

<%

Function UnMapPath(Path)

UnMapPath = Replace(Mid(Path, Len(Server.MapPath("/")) + 1), "\", "/")

End Function '定义函数 UnMapPath(path)。

Function MakeFolderHREF(AspPath, Path)

URLPath = Server.URLEncode(UnMapPath(Path))

```
MakeFolderHREF = "<A HREF=""" & AspPath & "?Folder=" & URLPath & """>"
```

End Function

Function MakeFileHREF(Path)

MakeFileHREF = " <a "="" """="" &="" href="" server.htmlencode(unmappath(path))="">
End Function
%> '定义添加文件夹和文件链接的函数。
<html></html>
<body></body>
<%
Folder = Request("Folder")
If Folder = "" Then Folder = "/"
AspPath = Request.ServerVariables("PATH_INFO")
Response.Write " <h2>文件夹 " & Folder & "<hr/></h2> "
On Error Resume Next
Set fs = Server.CreateObject("Scripting.FileSystemObject")
Set fd = fs.GetFolder(Server.MapPath(Folder))
If Folder <> "/" Then ' 是否有上一层文件夹
HREF_Path = MakeFolderHREF(AspPath, fd.ParentFolder.Path)
Response.Write HREF_Path & "[上一层文件夹] "
End If '输出[上一层文件夹]。
For Each sfd In fd.SubFolders
HREF = " "
HREF = HREF & MakeFolderHREF(AspPath, sfd.Path)
Response.Write HREF & sfd.Name & " "
Next '输出文件夹。
Response.Write " "
For Each f In fd.Files
HREF = " " & MakeFileHREF(f.Path)
Response.Write HREF & f.Name & " "
Next '输出文件。
%>

至此,dido 搜索引擎的所有内容都介绍完了。在本章的开头部分,曾提过本搜索引擎 中数据库的自动更新问题,由于 Web 服务器下的文件在不断更新和变化,因此,数据库中 关于关键词的搜索结果也需要随之更新。应该编写一个自动更新程序,在服务器访问量低 的时候运行,完成数据库的更新工作,或者在另外的备份服务器上运行。

希望读者能自己动手来编写这个自动更新程序,在此笔者讲述一下大体的思路:首先 与数据库建立连接,读出数据表中的所有记录,然后按照关键词的顺序依次重新进行搜索, 把搜索结果再依次存入数据库中,更新数据表中原来的搜索记录。

第10章 实用程序片断

任何一个功能强大的 Web 数据库系统中都有很多复杂的程序,但是这些复杂的程序都可以分解为一些简单的程序模块。笔者在开发 Web 数据库系统的过程中,最大感受是经常需要查阅一些小的程序片断,将它们改动后嵌到整个程序中去。这样做的好处很多:

- ◆ 可以直接使用现成的程序模块,节省开发时间。
- ◆ 在 ASP 程序开发中,开发者不必记住所有的 VBScript 函数,尤其是一些不常用的 函数。
- ◆ 这些做好的小程序模块都是经过实践考验的,包含很多技巧,从中也可以学到很 多编程思想。

笔者平时很注意这些小的程序片断的搜集和整理,这里把它们列出来并加以详细注释, 以方便读者实际应用。

在本章中,从三个方面来介绍程序段,即源代码说明、使用说明、结果显示。源代码 说明中将显示出完整的源代码;使用说明中主要说明此程序模块的使用方法、适用场合等 等;而结果显示中主要给出此程序段的显示效果,并且有详细注释。

10.1 计数器

在很多的网站中,都希望有一个站点计数器,以记录访问本网站的人次,这时,就需要一个计数器程序。读者在访问有计数器的站点时,如果留意的话,就会发现,有很多是 使用 CGI 程序来完成的,但是其实只需要在本站的首页文件中插入一段小小的计数器程序 就可以了。

10.1.1 代码说明

1. 计数器一

简单的站点计数器程序如下:

<%

Set fs=CreateObject("Scripting.FileSystemObject")

' 创建一个 Scripting.FileSystemObject 对象, 对服务器上的文件进行操作。

Set ts=fs.OpenTextFile("c:\MyCount.txt")

' 指定目标文件,这里所设的文件是 c 盘根目录下的 MyCount.txt 文件,当然这里也可以设置为虚拟 路径,和页面文件一样放在服务器的发布目录下,但是考虑安全还是放在指定的实际路径下。这里 的 MyCount.txt 文件就是记录访问次数的文本文件。

C=ts.ReadLine	'读出文本文件中的数字,并把它赋值给变量 C。
Response.write"您是第"	
Response.write C+1	
	'输出当前的访问次数。
Response.write"位光临本站"	
ts.Close	' 关闭对象。
Set ts=fs.CreateTextFile("c:\MyCount.txt")	' 重新建立新的文件对象。
TS.WriteLine(C+1)	'往记录文件中写入新的数字记录。
ts.Close	'关闭文件对象。
Set fs=nothing	
%>	

这段程序的运行过程十分简单,当程序执行时,首先打开记录文件,读出记录数并输 出,然后累加,再次写入记录文件。

把上面这段程序插入以.asp 为后缀的页面文件中,当浏览者每次加载这个.asp 文件时,都会执行上面的程序,从而起到页面计数的作用,并且显示当前浏览者是第几位访问者。 程序执行的结果如图 10-1 所示。



图 10-1 计数器显示效果

实例代码参考随书光盘中的计数器一。

2. 计数器二

如图 10-1 所示的效果已经能够达到计数器的基本要求,但是很多情况下希望能以一种 更活泼、更美观的方式来显示数字,例如使用图片来显示这些数字。这时,需要对上面计 数器一中的程序进行改进,改进后的源代码如下:

<%

Set fs=CreateObject("Scripting.FileSystemObject")

Set ts=fs.OpenTextFile("c:\MyCount.txt") .

C=ts.ReadLine

Response.write"您是第"

'以上的程序和计数器一中的是一样的。

'下面的循环语句将把数字转换为图片的文件名。

For i=1 To Len(C) Response.write"" ' 使用 MID 字符串函数逐一取出 C 中每一位上的数字。 Next ' 结束循环。 Response.write"位光临本站" ts.Close Set ts=fs.CreateTextFile("c:\MyCount.txt") TS.WriteLine(C+1) ts.Close Set fs=nothing %>

把这一段程序插入到页面之前,需要制作 10 幅数字图片,并且依次存为 count0.gif~ count9.gif,如图 10-2 所示。

图 10-2 计数图片

然后把这些文件保存在指定的 images 目录下就可以了。 程序运行的执行结果如图 10-3 所示,与图 10-1 相比效果上有了很大的改进。

💁 计数器	- Microsoft Int	ernet Explor	
」文件(6)			ж.
今 后退	• • • 🐼	ð 6 😳	in .*
	http://166.11		
您是第	订的利息	的位光临	本站

图 10-3 图片计数器显示效果

实例代码参考随书光盘中的计数器二。

3. 计数器三

读者实际应用上面的程序段后,就会发现该程序段仅仅起到一个页面计数的作用,而 不是站点计数。也就是说,如果浏览者不断刷新当前页面的话,计数器就会不断累加,这 样的话,将起不到实际站点计数的作用。

因此,需要进一步对上面的程序进行改进,对于已经访问过的用户不再重复计数。改

进的源代码如下:

|--|

Set fs=CreateObject("Scripting.FileSystemObject")

Set ts=fs.OpenTextFile("c:\MyCount.txt")

C=ts.ReadLine

Response.write"您是第"

For I=1 To Len(C)

Response.write"<img src="

response.write "images/count"

response.write MID(C,I,1)&".gif>"

Next

Response.write"位光临本站"

ts.Close

```
if session("image")="" then
```

'判断 session("image")是否为空,对当前的浏览者进行判断,是否是第一次访问。如果是第一次访问,则进行累加,否则不运行累加程序。

Set ts=fs.CreateTextFile("c:\MyCount.txt")

TS.WriteLine(C+1)

ts.Close

end if

```
session("image")="done"
```

'给 session("image")赋值,标明当前浏览者已经访问过。

Set fs=nothing

%>

以上程序主要是对浏览者进行判断,也就是通过 session("image")来进行标记。关于 session 对象的使用在前面已详细介绍过,这里不再赘述。

实例代码参考随书光盘中的计数器三。

10.2 地址重新定向

在很多情况下,对于不同的浏览者系统需要作出不同的反应,也就是说需要对不同的 浏览者显示出不同的页面。对于没有通过系统身份验证的用户,在访问未授予权限的页面 时,将会被自动定向到指定的页面,也就是地址重新定向。

地址重新定向程序十分简单,主要结合其他判断语句一起使用,比如下面的源代码:

<%

week = DatePart("w", Date()) - 1

'利用 DatePart 函数得到当前是星期几。DatePart 函数返回的值代表当前是一周中的第几天,而按照 西方的惯例,周日是每周的第一天,因此需要减去1。

WhichPage = "week" & week & ".htm"

Response.Redirect WhichPage

'使用 Response.Redirect 对地址进行重新定向。

%>

这段程序一般放在页面代码的开头,也就是<html>标记之前。本例的执行结果如图 10-4 所示。



图 10-4 地址重新定向

图 10-4 的地址栏中,显示的地址是 week6.htm,而实际输入的是 fruit.asp, fruit.asp 就 是嵌有重新定向程序的页面文件,同时也说明当天是星期六。

这里的程序选择条件是日期,每周中不同的天显示不同的页面。当然,这里的条件是 最重要的,可根据实际情况进行修改。

10.3 显示当前文件目录

在程序中进行条件判断时,有时需要得到当前文件所在的相对目录,然后在程序中调用。虽然当前文件的相对目录在浏览器地址栏中可以看到,但是后台程序在运行时不能随 意调用。显示当前文件相对目录的程序如下:

<%

Function GetCurrentPath()	•	定义函数 GetCurrentPath()。
Dim Path, pos	'	定义变量 Path 和 pos。

```
      Path = Request.ServerVariables("PATH_INFO")'获得当前文件的完整相对目录信息,包括文件名。

      response.write Path
      '输出当前文件的完整相对目录信息。

      pos = InStrRev(Path, "/")
      GetCurrentPath = Left(Path, pos - 1)
      '在完整的相对目录信息中去掉文件名。

      End Function
      '函数结束。

      %>

      <HTML>

      BODY>
      '在文件正文中调用函数 GetCurrentPath.

      'AC件ITAL>

      'AC件ITAL>
```

这段程序的输出结果,如图 10-5 所示。



图 10-5 显示当前目录

10.4 列举浏览器中的 cookie

对于 cookie 对象,在前面的章节中已详细介绍,这里的程序可以进一步加深对 cookie 的理解,熟悉对 cookie 对象的调用,熟练掌握 cookie 语句的使用。具体程序如下:

```
<HTML>
<BODY>
列举 Cookies:
<UL>
<%
If Request.Cookies.Count = 0 Then '判断是否存在 Cookie。
Response.Write "本浏览器之中未储存有任何 Cookie"
Else '列举所有的 Cookie
For Each cookie In Request.Cookies
Response.Write cookie & "="
Response.Write Request.Cookies(cookie) & "<P>" '列举 Cookie 中的所有单字。
```

```
Response.Write "<UL>"
For Each key in Request.Cookies(cookie)
Response.Write key & "="
Response.Write Request.Cookies(cookie)(key) & "<P>"
Next
Response.Write "</UL>"
Next ' 利用循环语句列出所有的 Cookie 值。
End If
%>
</UL>
```

其实,这些 cookie 都是存储在浏览者的客户端,保存在系统盘的 TemporaryInternetFile 目录下。利用这个程序可以读出存放在客户端的所有 cookie 信息。执行结果如图 10-6 所示。

🕽 http://166.111.1 🕅 💌 🗊
Marine Marine 286
列举 Cookies,
Love=eat
Name=pett

图 10-6 列举浏览器中的所有 cookie

图 10-6 显示了笔者浏览器中的所有 cookie 值, cookie(love)的值为 eat, 而 cookie(name) 的值为 pett。

10.5 留言簿

在很多情况下,可能需要存储一些简单信息,但又没有必要使用复杂的专门数据库来 完成。使用*.txt 文本文件也可以存储信息,完成交互功能。下面向读者介绍一个简单的留 言簿程序,所有的程序都在一个页面文件 newlyb.asp 中。源代码如下:

<html>

<head>

<title>留言簿</title>

<STYLE type=text/css>BODY { FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt; LINE-HEIGHT: 18px} TD { FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt; LINE-HEIGHT: 18px } SELECT { FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt } INPUT { FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt} .font {}A {TEXT-DECORATION: none; TEXT-TRANSFORM: none} A:hover { TEXT-DECORATION: underline} .sgxing { FONT-SIZE: 12pt} </STYLE> ' 网页样式标记。 </head> <body background="bg9.gif"> 留言簿 <hr width=85% color=#CC0099> '程序代码。 <%if request.form("message")<>"" then '判断是否有提交的表单信息。 a=request.form("title") b=request.form("message") user=request.form("name") '给提交的表单信息加上格式修饰。 a="<div align=center><center>" &"标题: "&a&" b="font color=#0000FF>留言内容:"&b&" user=""&user user=user&"7-" user=user&date&"留言</div></hr color-#CC0099>" Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject") '建立 Scripting.FileSystemObject 对象。 lujing=server.mappath("\note\note.txt") ' 指定存储信息的文本文件路径。 Set MyTextFile=MyFileObject.OpenTextFile(""&lujing&"",8,TRUE) MyTextFile.WriteLine(a) MyTextFile.WriteLine(b) MyTextFile.WriteLine(user) MyTextFile.Close '将提交的表单信息写入文本文件。 set MyFileObject=nothing end if %> <% Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject") lujing=server.mappath("\note\note.txt")

Set MyTextFile=MyFileObject.OpenTextFile(""&lujing&"")

While not MyTextFile.AtEndOfStream

Response.Write(MyTextFile.ReadLine)

Wend

MyTextFile.Close '再次建立 Scripting.FileSystem 对象,读取文本文件中的所有信息,并输出。 %>

对我的主页如果有任何的意见请您留言:

<div align="center">

<center>

<form method="post" action="newlyb.asp" name="input">

```
<span class="fontlittle"><font color="#0000FF" size="2">请输入姓名:
```


<input name="name" style="background-color: #ffffff; font-size: 9pt; width: 179; height: 23; border: 1px solid #9a9999" size="20">

```
<span class="fontlittle"><font color="#0000FF" size="2">请输入主题:
```



```
<input name="title" size="6" style="background-color: #ffffff; font-size: 9pt; width: 182; height: 23; border: 1px solid #9a9999">
```

```
<span class="fontlittle"><font color="#0000FF" size="2">请输入留言内容:
```



```
<textarea rows="2" name="message" cols="23"></textarea>
```

```
<input border="0" cache src="newlyb1.gif" type="image" width="60" height="41">
```

'用于提交信息的表单。

</form>

</center>

</div>

</body>

</html>

从上面的程序可以看出,表单中的信息实际上是提交到自身进行处理,这样就可以把 表单页面和处理页面结合在一起使用,简化了程序结构。使用以上程序时,需要在相应的 指定路径下建立 note.txt 文件,用于存储留言信息。

注意:这里的路径是虚拟路径,并不是磁盘上的实际路径。

建立好文本文件后,把上面的 newlyb.asp 文件放在服务器的发布目录下,就可以拥有 自己的留言簿了。在其他页面上提供 newlyb.asp 的链接, 浏览者就可以访问这个留言簿, 当没有任何留言时,画面如图 10-7 所示。

国 留古牌 Micro	oft Internet Explorer	ininiimen seedii kin kiituseen si of olii	
and the second	a an	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1996 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	e
Silve Seco	/ * 8 ^ .	r	a B. 0
http:/	/166.111.170.110/note/nev	wlyb.asp 📕	
	切主	*	
	W P	A	define 🙎
			<u>i ik</u> š
对我的主页的复数	任何的意见诸然错言:		
and the second sec		a langi	
请输入姓名	i pett		
请输入主题			
108	- 我们去放风筝	ar a	
增加人們1	M41		10.5
	e Maria		
			5 AU
1			

图 10-7 留言簿(没有留言)

假设目前有一位用户留言, 留言内容如图 10-7 中所示, 点击"留言", 留言内容将提 交到服务器端的文本文件中, 并且显示出来, 如图 10-8 所示。

] 태금, 御 - Merosoft tr	renet Explorer		ast Constant
Contraction of the second s			- K-
	留言	*	
	様型:春天 輩言内書:我们	来了! 去放风等吧 p=tt于2001-4-	15 🖀 🗃
对我的主页的是有任何。	n a. u. a a a a a a		
谱输入姓名; 谱输入主题;			

图 10-8 留言簿 (查看留言)

可能读者注意到,这里的提交按钮不再是常见的"提交"按钮,而是图片形式,这样 显得更加活泼,增加了页面的可读性。

10.6 投票程序

在很多站点中,都希望征求浏览者对某件事物(如网站刚推出的栏目)的看法,或者 对某一个舆论焦点的态度,这时用一个简单的投票程序就可以实现。下面将介绍一个这样 的投票程序。

这个投票程序也只有一个页面文件,也就是说所有的程序代码都是放在一个文件中。 在这个程序中,为了直观地表现每一参选项目当前的得票数,以及与其他参选项目的比较, 利用长度与当前得票数成比例的线条来表示每一参选项目的相对值。因为在 HTML 语法中 图片的大小可以指定,所以可以利用这一点来改变线条的长度。假如共有 n 个参选项目, 首先建立 n 个文本文件,用来存放每一个参选项目的得票数。当程序执行时,先读出文本 文件中每一个参选项目的得票数,然后从这 n 个数据中选出最大的,并且计算出每一个数 据与最大数据的相对值,在页面中插入 n 个线条图片用于表示每一个参选项目的相对得票 数,把得票最多的参选项目的线条长度定义为 100 像素,其他参选项目的线条长度按照前 面的相对值来确定。然后在页面中建立 n 个单选按钮,方便浏览者对自己支持的参选项目 投票,当然,每一个单选按钮对应一个参选项目,并且把这 n 个单选按钮定义为同一名称, 并且从 1 到 n 进行编号,当提交时,表单值将提交到文件本身的程序,根据提交的值就可 以找到相应的参选项目,然后把这个参选项目的得票数加 1。

另外,为了防止同一浏览者重复投票,插入了限制语句。

这里所举的实例中参选项目总数为 4, 文件名为 vote.asp。完整的源代码如下:

<html>

<head>

```
<meta http-equiv="Content-Language" content="zh-cn">
```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

```
<meta name="ProgId" content="FrontPage.Editor.Document">
```

```
<STYLE type=text/css>BODY {
```

```
FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt; LINE-HEIGHT: 18px
```

```
}
```

TD {

FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt; LINE-HEIGHT: 18px

}

SELECT {

FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt

}

```
INPUT {
FONT-FAMILY: Arial 宋体; FONT-SIZE: 9pt
}
.font {
}
A {
TEXT-DECORATION: none; TEXT-TRANSFORM: none
}
A:hover {
TEXT-DECORATION: underline
}
.sgxing {
FONT-SIZE: 12pt
}
</STYLE>
                            '指定页面样式。
<title>投票</title>
</head>
<body background="bg9.gif">
<font color="#FF0000">你对我的主页的感觉如何,请发表你的意见! </font>
<form method="POST" action="vote.asp" name="vote" >
注意: 表单的内容递交到自身程序。
<%dim a1,a2,a3,a4,b1,b2,b3,b4,c,n1,n2,n
c=100
Set fs=server.CreateObject("Scripting.FileSystemObject")
                                                  '建立 Scripting.FileSystemObject 对象。
lujing=server.mappath("/note/count1.txt")
Set ts1=fs.OpenTextFile(""&lujing&"")
a1=ts1.ReadLine
ts1.Close
              '在本程序中,需要记录对主页的四种意见,因此需要依次读取四个文件中的数字。
lujing=server.mappath("/note/count2.txt")
Set ts2=fs.OpenTextFile(""&lujing&"")
a2=ts2.ReadLine
ts2.Close
lujing=server.mappath("/note/count3.txt")
Set ts3=fs.OpenTextFile(""&lujing&"")
a3=ts3.ReadLine
ts3.Close
lujing=server.mappath("/note/count4.txt")
```

```
Set ts4=fs.OpenTextFile(""&lujing&"")
a4=ts4.readline
ts4.Close
                 '当然,也可以把四个数据放在同一个文件中。
if a1>a2 then
   n1=a1
   else
   n1=a2
   end if
if a3>a4 then
   n2=a3
   else
   n2=a4
end if
if n1>n2 then
   n=n1
else
   n=n2
                         '选出四个数据中的最大值。
end if
b1=c*a1/n
b2=c*a2/n
b3=c*a3/n
b4=c*a4/n
                         '计算出每一个线条图片的长度数据。
set fs=nothing%>
<%if session("vote")="" and request.form("choice")<>""then
                                                            '判断用户是否为第一次投票。
Set fs=server.CreateObject("Scripting.FileSystemObject")
          select case request.form("choice")
           case "c1": lujing=server.mappath("/note/count1.txt")
            case "c2": lujing=server.mappath("/note/count2.txt")
            case "c3": lujing=server.mappath("/note/count3.txt")
            case "c4": lujing=server.mappath("/note/count4.txt")
         end select
                                                '选择需要更新的记录文件。
Set ts=fs.CreateTextFile(""&lujing&"",True)
select case request.form("choice")
            case "c1": TS.WriteLine(a1+1)
            case "c2": TS.WriteLine(a2+1)
            case "c3": TS.WriteLine(a3+1)
            case "c4": TS.WriteLine(a4+1)
          end select
                                                '更新选定的记录数据。
```

ts.Close

Set fs=nothing Session("vote")= "done" end if '程序段结束,显示表单内容和目前得票结果。 %> <div align="center"> <input type="radio" value="c1" checked name="choice">做得 好棒 <img src="line.gif" width=<%=b1%> height=8> <%=a1%> 1 指定相应的线条图片的长度。 <input type="radio" name="choice" value="c2">挺不错的 <img src="line.gif" width=<%=b2%> height=8> <%=a2%> <input type="radio" name="choice" value="c3">马马虎虎 <img src="line.gif" width=<%=b3%> height=8> <%=a3%> <input type="radio" name="choice" value="c4">还需改进 <img src="line.gif" width=<%=b4%> height=8> <%=a4%> <center> <input type="submit" name="submit" value="投票"</pre>

```
' 增加 Windows 提示窗口,加深浏览者印象。
```

</wd>

 <% if session("vote")="" then %>
 onclick=window.alert("谢谢您投了宝贵的一票") '如果是第一次投票的话,感谢浏览者的支持。
 <//>
 <//>
 <//>
 <//>
 <//>
 <//>
 <//>
 <//>
 <//div>
 <//div
 <//div

在使用这一段程序之前,需要制作一个线条图片,如图 10-9 所示,存为 line.gif,并且 和 vote.asp 放在同一发布目录下。

图 10-9 线条图

从上面的程序可以发现,这段程序的可移植性很强,只需要改变其中的参选项目,就 可以了,如图 10-10 所示。



图 10-10 投票窗口

10.7 随机图片显示

页面上的图片如果可以不断更新,将会吸引更多浏览者访问,但是,要通过手工更新 是不现实的。这时可以考虑一个随机程序,首先制作多幅图片,保存在指定的发布目录下, 然后随机显示,这样,浏览者每次访问时都可能看见不同的图片,从而达到吸引浏览者的 目的。

随机图片显示程序 random.asp 十分简单,直接调用 VBScript 中的随机函数就可以了。 random.asp 的源程序如下:

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

<meta name="GENERATOR" content="Microsoft FrontPage 4.0">

<meta name="ProgId" content="FrontPage.Editor.Document">

<title>随机图片显示</title>

</head>

<body>

<%Randomize

No = Int((9 - 0 + 1) * Rnd) + 0

产生0~9之间的随机数。

下面说明 Randomize 与 Rnd 函数的用法。

Rnd[(number)] 返回一个随机数。

number 参数可以是任意有效的数值表达式。Rnd 函数返回一个小于 1 但大于或等于 0 的值。number 的值决定了 Rnd 生成随机数的方式。

因为每一次连续调用 Rnd 函数时都用序列中的前一个数作为下一个数的种子,所以对 于任何最初给定的种子都会生成相同的数列。

在调用 Rnd 之前,先使用无参数的 Randomize 语句初始化随机数生成器,该生成器具 有基于系统计时器的种子。

要产生指定范围的随机整数,可以使用以下公式:

Int((upperbound - lowerbound + 1) * Rnd + lowerbound)

这里, upperbound 是此范围的上界, 而 lowerbound 是此范围的下界。

Randomize 使用 number 参数初始化 Rnd 函数的随机数生成器, 赋给它新的种子值。如果省略 number, 则使用系统计时器返回的值作为新的种子值。

如果不使用 Randomize,则第一次调用 Rnd 函数(无参数)时,它将使用相同的数字 作为种子值,随后使用最后生成的数值作为种子值。 Response.Write "随机数: "&No & "
"

response.write "随机图片:<img border=0 src=../images/count"

response.write No&".gif>"

```
'根据随机数的不同而显示不同的图片。
```

%>

</body>

</html>

显示效果如图 10-11 所示。



图 10-11 随机图片显示

在实际使用时,只需要把例中的数字图片换为相应的图片文件就可以了,但要注意, 为了简化程序,所有的随机图片文件名称应该具有某种编号特征,以和程序中产生的随机 数建立联系,便于程序中对图片文件的调用。这样的程序多用于站点中的广告图片显示。

10.8 表单限制

Web 数据库系统中,有很大一部分程序都是对用户提交的表单信息进行处理,但有时 浏览者提交的表单信息中缺少一些必要的内容。为了对提交的表单内容进行限制,可以在 页面中插入一些小程序,对表单的值进行限制。当然,这些程序都是客户端程序,也就是 说程序是在客户端运行的,不消耗服务器端的资源。

最常用的一段 javascript 表单限制程序如下:

```
<script language="JavaScript">
function getstring(str)
{
rstr="";
for(i=0;i<str.length;i++)
{ if(str.charAt(i)!="")
```
```
rstr=rstr+str.charAt(i);
  }
  return rstr;
}
'首先建立一个读取字符串的函数。
function submitclick()
{
  cnt=0;
  titlestr="[请注意] \n\n";
  errstr="";
  tmpzcm=getstring(document.input.lybid.value);
  if(tmpzcm="")
' 检查 document 表单中的 input 单行文本框 lybid 的值是否为空。
'如果为空的话,指出相应的表单项。
   { cnt=cnt+1;
     errstr=errstr+cnt+"."+"请输入您的姓名\n";
  }
  else document.input.lybid.value=tmpzcm;
  tmply=getstring(document.input.password.value);
  if(tmply=="")
'检查 document 表单中的 input 单行文本框 password 的值是否为空。
   { cnt=cnt+1;
     errstr=errstr+cnt+"."+"请输入您的口令\n";
   }
   else document.input.password.value=tmply;
   if(cnt==0)return true;
   else
   { window.alert(errstr);
     return false;
   }
}
```

上面的程序片断只是定义了一个函数,必须在页面中调用,函数才能被执行。一般都 是在浏览者提交表单时调用这些函数进行判断,调用函数的语句是 OnClick="return submitclick();",这一语句嵌在提交按钮的属性语句中。

假如本例中浏览者忘记输入密码项,也就是说 password 文本框的值为空,这时如果提 交, 弹出的警告窗口则如图 10-12 所示。



图 10-12 表单限制程序的警告窗口

在实际应用中调用这段程序,只需把上面的函数插在页面的任何部分,改动其中的文 本框名称就可以了。

10.9 页面刷新程序

在一些 ASP 系统中,可能会即时发布一些信息,比如用 ASP 编写的聊天室程序,都 是通过不断刷新页面来读取新的信息。但是,刷新时间的长度很难确定,时间太长,影响 信息更新速度,时间太短页面将不断刷新,影响浏览者查看。刷新时间最好在客户端可控, 从而使浏览者在客户端得到希望的效果。

一般的刷新程序都是采用框架网页来完成的,把刷新程序放置在其中的一个框架中, 被刷新的页面放在另一个框架中,如图 10-13 所示。



图 10-13 刷新程序

整个框架文件的源代码如下:

<HTML>

<HEAD><TITLE>页面刷新程序</TITLE>

</HEAD>

<frameset cols="40%,*">

<Frame src="fresh.htm" Marginwidth=1 Marginheight=1</pre>

Scrolling="no">

<Frame Name="main" src="main.asp">

注意: 这里的框架名称和目标文件是很重要的, 是下面刷新程序中的调用参数。

</Frameset>

</HTML>

刷新程序 fresh.htm 的所有源代码如下:

<html>

<head></head>

<script language="vbscript">

<!--

Dim ival

Sub newscheck() ' 定义一个子程序。

'Dim i

If post1.interval.value = Empty Then

ival = 60000 '设定刷新时间的默认值。

Else

```
ival = CInt(post1.interval.value)*1000
```

'读取表单 interval 的值,并且换算为刷新时间,因为浏览器中的刷新时间是以毫秒为单位。

注意: Clnt 与 Fix 和 Int 函数不同,后两者将数字的分数部分截尾取整,而不是四 舍五入。当分数部分恰好为 0.5 时,Clnt 函数通常将其四舍五入为最接近的偶数。例如,0.5 被四舍五入为 0, 而 1.5 被四舍五入为 2。

End If

Parent.main.Navigate "main.asp" 制新指定框架中的制定文件。

```
SetTimeout "newscheck", ival
```

End Sub

SetTimeout "newscheck", ival --> </script> <body>
 <h2>刷新程序</h2> <hr> <hr> <FORM NAME="post1" METHOD="POST"> 每隔 <INPUT TYPE="TEXT" NAME="interval" SIZE="3" MAXLENGTH="3" VALUE="60">秒更新一次(预设值 60 秒) </form> </body></html>

在图 10-13 中,控制刷新时间的程序在窗口左边的框架中,而被刷新的窗口位于窗口 右面的框架中。

本章列出了在 Web 数据库开发中经常需要用到的一些源代码,这些代码对于开发者来 说是至关重要的。读者在开发过程中也应该养成积累代码的习惯,对于一些重要的程序段 应该搜集起来,需要时直接拿过来使用,这样就可以提高编程的效率。

附录A SQL 语言

A.1 数据库管理

```
1. 创建数据库
```

```
CREATE DATABASE database name
[ ON [PRIMARY]
[ <filespec> [,...n]]
[, <filegroup> [,...n]]
1
[LOG ON { <filespec> }]
[FOR LOAD | FOR ATTACH ]
其中, <filespec> :: =
([NAME = logical file name,])
FILENAME = 'os file name'
[, SIZE = size]
[, MAXSIZE = { max size |UNLIMITED } ]
[, FILEGROWTH = growth increment])[,...n]
<filegroup> :: = FILEGROUP filegroup_name <filespec> [,...n]
2. 修改数据库
ALTER DATABASE database
{ ADD FILE<filespec> [,...n] [TO FILEGROUP filegroup_name]
ADD LOG FILE <filespec> [,...n]
REMOVE FILE logical file_name
ADD FILEGROUP filegroup name
| REMOVE FILEGROUP filegroup_name
| MODIFY FILE <filespec>
| MODIFY FILEGROUP filegroup name filegroup property
}
其中, <filespec> :: =
(NAME = 'logical_file_name'
[, FILENAME = 'os_file_name']
[,SIZE = size]
[, MAXSIZE = { max size | UNLIMITED }]
```

```
[, FILEGROWTH = growth_increment])
```

3. 删除数据库

DROP DATABASE database_name [,...n]

```
1. 创建表
```

```
CREATE TABLE table name
( { <column definition>
column name AS computed column expression
< table_constraint >
}[,...n]
)
[ ON {filegroup | DEFAULT} ]
[TEXTIMAGE ON { filegroup | DEFAULT } ]
其中, <column definition> :: = { column_name data_type }
[NULL | NOT NULL ]
[ IDENTITY [(seed, increment )
[NOT FOR REPLICATION]]]
[ROWGUIDCOL]
[ < column_constraint > :: = [CONSTRAINT constraint_name ]
{ { PRIMARY KEY | UNIQUE }
[ CLUSTERED | NONCLUSTERED ]
[ WITH [ FILLFACTOR = fillfactor ]
1
[ ON {filegroup | DEFAULT} ]
[FOREIGN KEY]
REFERENCES ref table
[(ref_column)]
[ NOT FOR REPLICATION]
| DEFAULT constant expression
| CHECK [NOT FOR REPLICATION]
 (logical expression)
}
][...n]
其中  :: = [ CONSTRAINT constraint_name]
```

```
{ [ {PRIMARY KEY | UNIQUE }
[ CLUSTERED | NONCLUSTERED]
\{(column[,...n])\}
[ WITH [ FILLFACTOR = fillfactor]
1
[ ON {filegroup | DEFAULT} ]
1
| FOREIGN KEY
[(column[,...n])]
REFERENCES ref_table [ (ref_column[,...n])]
[NOT FOR REPLICATION]
CHECK [NOT FOR REPLICATION]
(search conditions)
}
2. 修改表
ALTER TABLE table
{
[WITH CHECK | WITH NOCHECK ]
{ [ ALTER COLUMN column name
{
[new_data_type [(precision[, scale])]
[NULL | NOT NULL ]]
|[ { ADD | DROP} ROWGUIDCOL ]
}
]
ADD
{ [ <column definition> ]
column_name AS computed column expression
[  ]
}[,...n]
DROP
{ [CONSTRAINT] constraint
| COLUMN column
}[,...n]
| {CHECK | NOCHECK } CONSTRAINT
{ ALL | constraint[,...n]}
| {ENABLE | DISABLE} TRIGGER
```

```
{ ALL | trigger[,...n]}
}
其中, < column_definition> :: = { column_name data_type }
[NULL | NOT NULL ]
[ IDENTITY [ (seed[, increment])
[NOT FOR REPLICATION ]]]
[ROWGUIDCOL]
[ <column_constraint> :: = [CONSTRAINT constraint name]
{ { PRIMARY KEY | UNIQUE }
[ CLUSTERED | NONCLUSTERED]
[ WITH [FILLFACTOR = fillfactor]
1
[ ON {filegroup | DEFAULT} ]
[FOREIGN KEY]
REFERENCES ref table
[(ref_column)]
[NOT FOR REPLICATION]
| DEFAULT constant_expression [WITH VALUES]
| CHECK [ NOT FOR REPLICATION ]
(logical_expression)
}
][,...n]
其中, <table_constraint> :: = [ CONSTRAINT constraint_name ]
{ [ { PRIMARY KEY | UNIQUE }
[ CLUSTERED | NONCLUSTERED ]
{ (column[,...n]) }
[ WITH [ FILLFACTOR = fillfactor ]
1
[ON {filegroup | DEFAULT}]
1
FOREIGN KEY
[(column[,...n])]
REFERENCES ref_table [ (ref_column[,...n])]
[ NOT FOR REPLICATION ]
DEFAULT constant expression
[FOR column]
| CHECK [ NOT FOR REPLICATION ]
```

(logical_expression)

}

3. 删除表

DROP TABLE table_name

A.3 索引管理

1. 创建索引

CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name ON table (column [,...n]) [WITH [PAD_INDEX] [[,] FILLFACTOR = fillfactor] [[,] IGNORE_DUP_KEY] [[,] DROP_EXISTING] [[,]STATISTICS_NORECOMPUTE]] [ON filegroup]

2. 删除索引

DROP INDEX 'table.index' [,...n]

A.4 视图管理

1. 创建视图

CREATE VIEW view_name [(column [,...n])] [WITH ENCRYPTION] AS select_statement [WITH CHECK OPTION] 2. 修改视图

ALTER VIEW view_name [(column [,...n])] [WITH ENCRYPTION] AS select statement [WITH CHECK OPTION]

3. 删除视图

DROP VIEW {view} [,...n]

A.5 触发器管理

1. 创建触发器

```
CREATE TRIGGER trigger_name
ON table
[WITH ENCRYPTION]
{ {FOR {[,] [DELETE] [,] [INSERT] [,] [UPDATE] }
[WITH APPEND]
[NOT FOR REPLICATION]
AS
sql_statement [ ...n ]
}
| {FOR { [,] [INSERT] [,] [UPDATE]}
[WITH APPEND]
[NOT FOR REPLICATION]
AS
{ IF UPDATE (column)
[ { AND | OR } UPDATE (column) ]
[...n]
| IF (COLUMNS_UPDATED() {bitwise_operator} updated_bitmask)
{ comparison_opertor} column_bitmask [ ...n]
}
sql_statement [ ...n ]
}
}
2. 修改触发器
ALTER TRIGGER trigger_name
ON table
```

[WITH ENCRYPTION] { { FOR { [,] [DELETE] [,] [UPDATE] [,][INSERT]} [NOT FOR REPLICATION]

AS sql_statement [...n] } | {FOR { [,] [INSERT] [,] [UPDATE]} [NOT FOR REPLICATION] AS { IF UPDATE (column) [{AND | OR } UPDATE (column)] [...n] | IF (COLUMNS UPDATED() {bitwise_operator} updated_bitmask) { comparison_operator } column_bitmask [...n] } sql statement [... n] } } 3. 删除触发器

```
DROP TRIGGER {trigger} [,...n]
```

A.6 存储过程管理

1. 创建存储过程

```
CREATE PROC[EDURE] procedure_name [ ; number]
[ { @parameter data_type } [VARYING] [ = default ] [OUTPUT]
] [,...n]
[ WITH
{ RECOMPILE
| ENCRYPTION
| RECOMPILE, ENCRYPTION
}
]
[ FOR REPLICATION ]
AS
sql_statement [ ...n ]
2. 修改存储过程
ALTER PROC[EDURE] procedure_name [ ; number ]
```

```
[ { @parameter data_type } [ = default] [OUTPUT]
][,...n]
[ WITH
{ RECOMPILE
| ENCRYPTION
| RECOMPILE, ENCRYPTION
}
1
[FOR REPLICATION]
AS
sql_statement [ ...n]
3. 执行存储过程
[[EXEC[UTE]]
{ [ @return_status = ]
{ procedure_name [; number] | @procedure_name_var
}
1
[ [ @parameter = ] value |@variable [OUTPUT] | [DEFAULT] ] [ ,...n ]
[WITH RECOMPILE]
4. 删除存储过程
DROP PROCEDURE { procedure } [,...n]
```

A.7 规则管理

1. 创建规则

CREATE RULE rule AS condition_expression

2. 绑定规则

sp_bindrule [@rulename =] 'rule' ,

[@objname =] 'object_name'

[,[@futureonly =] 'futureonly_flag']

3. 解除绑定的规则

sp_unbindrule [@objname =] 'object_name',

```
[, [@futureonly = ] 'futureonly_flag']
```

4. 删除规则

DROP RULE {rule} [,...n]

A.8 默认管理

1. 创建默认

CREATE DEFAULT default AS constant_expression

2. 绑帮定默认

sp_bindefault [@defname =] 'default' ,

[@objname =] 'object_name'

[, [@futureonly =] 'futureonly_flag']

3. 解除帮定的默认

sp_unbindefault [@objname =] 'object_name'

```
[, [@futureonly = ] 'futureonly_flag' ]
```

4. 删除默认

```
DROP DEFAULT {default } [,...n]
```

A.9 查询语句

SELECT [ALL | DISTINCT]
[TOP n [PERCENT] [WITH TIES]] <select_list>
[INTO new_table]
[FROM <table_source>]
[WHERE <search_condition>]
[[GROUP BY [ALL] group_by_expression [,...n]]
[HAVING <search_conditions>]
[WITH { CUBE | ROLLUP }]]
[ORDER BY { column_name [ASC | DESC] } [,...n]]
[COMPUTE
{ { AVG | COUNT | MAX | MIN | SUM } (expression) } [,...n]
[BY expression [,...n]]
[FOR BROWSE]
[OPTION (<query_hints>)]
其中 <select list> :: =

```
{ [ { <table_or view> | table alias } . ]*
| {column_name | expression | IDENTITYCOL | ROWGUIDCOL }
[[AS] column alias]
| new column name = IDENTITY (data type, seed, increment)
| GROUPING (column name)
| {table_name | table_alias} . RANK
| column alias = expression
expression column name
} [ ,...n ]
<table_sources> :: =
{ 
(select_statement) [AS] table_alias [ (column_alias [,...n])]
<table_or_view> CROSS JOIN <table_or_view>
{ { INNER
| { FULL | LEFT | RIGHT }
[OUTER][<join hints>][JOIN]
} <table_or_view>ON <join_condition>
}
| <rowset_function>
} [ ,...n]
 :: =
{ table_name [ [AS] table_alias ] [WITH ( <table_hints> [ ...n] ) ]
view_name [ [AS] table_alias] }
<table_hints> :: =
{ INDEX (index_name | index_id [,...n])
| FASTFIRSTROW
| HOLDLOCK
| NOLOCK
| PAGLOCK
| READCOMMITTED
| READPAST
| READUNCOMMITTED
| REPEATABLEREAD
ROWLOCK
SERIALIZABLE
| TABLOCK
| TABLOCKX
```

```
| UPDLOCK
}
<join_hints> :: = { HASH | LOOP | MERGE }
<query_hints> :: =
{ { HASH | ORDER } GROUP
| { CONCAT | HASH | MERGE } UNION
|FAST number_rows
| FORCE ORDER
MAXDOP number
| ROBUST PLAN }
<join_condition> :: =
{ table_name | table_alias | view_name } . column_name
<logical_operator>
{ table_name | table_alias | view_name } .column_name
< logical operator > :: = \{ = | > | < | > = | < = | < > |! = |! < |! > |
<rowset_function> :: =
{ CONTAINSTABLE [ [AS] table_alias ]
(table, { column | * },'<contains_search_condition>')
[FREETEXTTABLE [ [AS] table_alias]
(table, { column | * }, 'freetext_string')
| OPENQUERY (linked_server, 'query')
| OPENROWSET
('provider_name',
{ 'datasource', 'user_id'; 'password'
| 'provider_string' },
| [catalog.][schema.]object_name
| 'query' } )
<search conditions> :: =
{ [NOT] <predicate> [ { AND | OR } [NOT] <predicate> ]
| CONTAINS
({column | •}, '<contains_search_condition>')
FREETEXT
({column | •}, 'freetext_string')
| fulltext_table.gulltext_key_column = fulltext_table.[KEY]
} [ ,...n]
<predicate> :: =
| expression \{ = | <> | ! = | > = | !> | < | <= | ! < \} expression
string_expression [NOT] LIKE string_expression
```

```
[ESCAPE 'escape_character']
| expression [NOT] BETWEEN expression AND expression
| expression IS [NOT] NULL
| expression [NOT] IN (subquer | expression [ ,...n ] )
| expression { = | <> | ! = | > | > = | !> | < | < = | ! < }
{ ALL | SOME | ANY } (subquery)
}</pre>
```

A.10 许可管理

1. 语句许可是授权管理

GRANT {ALL | statement [,...n] } TO security_account [,...n]

2. 对象许可授权管理

GRANT

```
{ALL [PRIVILEGES] | permission [ ,...n ] }
```

```
\{ [ (column[,...n]) ] ON \{ table | view \} \}
```

```
| ON \{table | view\} [ (column[,...n]) ]
```

```
| ON {stored_procedure | extended_procedure}
```

```
}
```

```
TO security_account [ ,...n ]
```

```
[WITH GRANT OPTION]
```

```
[ AS {group | role} ]
```

3. 语句许可授权管理

REVOKE {ALL | statement[,...n] FROM security_account[,...n]

4. 对象许可授权管理

```
REVOKE [GRANT OPTION FOR]
| ALL [PRIVILEGES] | permission [,...n] }
{ [ (column [,...n] ) ] ON {table | view }
| ON {table | view} [ (column[,...n] ) ]
| {stored_procedure | extended_procedure }
}
{ TO | FROM }
security_account[,...n]
[CASCADE]
```

[AS {group | role }]

5. 语句许可否定管理

DENY{ALL | statement [,...n] } TO security_account [,...n]

6. 对象许可否定管理

DENY

{ALL [PRIVILEGES] | permission [,...n] }

{ [(column[,...n])] ON {table | view}

 $|ON \{table | view\} [(column[,...n])]$

| ON {stored_procedure | extended_procedure }

}

```
TO security_account[,...n]
```

[CASCADE]

附录 B Web 数据库网络资源

B.1 Web 数据库实例

为了便于读者参考,下面列出了网络中当前使用 ASP 开发的 Web 数据库实例。

1. 中国教育热线(http://www.eol.com.cn/eol_index/index.asp)

一个面向教育的大型中文 Internet 信息服务站点,是一个依托"中国教育与科研计算机 互联网——CERNET",进行教育增值信息开发的 Internet 网络信息平台。其目标是在 Internet 上建立一个满足教育需求、真正活跃的虚拟社会,其宗旨在于为教育领域的广大消费者和 从业者提供一个自由而充分地利用 Internet 进行信息交流的环境。

2. FM365 (http://fm365.com)

联想公司打造的大型综合型网站。

3. 听吧 (http://www.tingbar.com/)

听吧音乐网络有限公司的宗旨是利用网络技术的契机推动中国音乐市场的繁荣发展。 听吧音乐网站(www.tingbar.com)于 2000 年初成立, 2000 年 4 月正式开通。此网拥有中 国最为丰富的音乐数据库,收集了近 20 年来国内外出版的各种 CD 资料,并为音乐爱好者 提供在线试听服务。

4. 博士网(http://www.helpwork.net/)

包含书库、代码、软件、控件和驱动等,是一个网上学习和交流的好地方。

5. 三九健康网(http://www.999.com.cn/)

三九健康网是三九网络旗下的一个综合性健康网站,包含了目前健康概念下的所有内容,拥有最强大的专业搜索引擎,最个性化的专业主页空间,最即时的专业领域动态和信息,最准确和热点的话题及调查,并本着科学和严谨的态度,组建遍布全国的网上诊疗系统和有效的电子商务系统。

6. 中关村在线(http://www.zol.com.cn/)

ZOL 是中关村在线(Zhongguancun On Line)的缩写。此网站立足于中关村,并将中关村 作为中国 IT 行业的窗口。真正做到第一时间反映中关村价格变化和 IT 业界动态,包含各 电脑配件、外设价格及中关村的市场动态。

7. 运博赴美留学网(http://embark.cscse.edu.cn/)

中国留学服务中心(中国留学信息网)与美国 Embark.com 公司合作,将 Embark.com 在线留学申请系统全面引入中国,为有意赴美留学的广大学子提供更方便、快捷、经济的 网上留学申请,并为赴美学子提供相关咨询和代收申请费服务。

8. 中国考试信息中心(http://KaoShiLa.yeah.net/)

包含升学、英语、计算机资格认证等各种考试信息,还可以进行高考和四级六级英语 成绩查询,还有各种考试的资料软件下载,是一个方便的网站。

9. 学生公社(http://www.univerciti.com/)

一个为学生服务的网站,包括同学录、求职、社团、课题、创业、留学、书籍、校刊 高考和考研等考试信息以及论坛聊天等内容。

10. 蔚蓝网络书店(http://www.welan.com/)

一个网上书店,数目齐全,价格合理,开通多条配送站点。

11. 前程无忧网(http://www.51job.com/)

2000 年 2 月,51job.com、《前程周刊》和润安(R&A)信息顾问有限公司结成"前程 无忧"联盟,向业界提供包括网络招聘、报纸广告和猎头在内的全方位综合人事招聘服务, 是迄今为止国内第一个集多种媒介优势的人力资源服务系统。求职者可以更有效、更快捷 地在职位库中搜索到适合自己的职位。

12. 卓越网(http://www.joyo.com/)

由国内顶尖 IT 企业联想、金山共同投资的优秀电子商务网站,主营图书、音像、软件、 游戏、玩具等文化娱乐消费产品,同时提供丰富而具纵深感的相关内容与资讯。自 2000 年 月 5 月 11 日正式发布后,以其精品、特价、快捷配送等优势迅速树立起良好的商业网站形 象,并已跻身于中国最出色的电子商务网站之列。

13. 当当书店(http://www.dangdang.com/)

一个网上书店,是一个完整的 Web 数据库实例。

14. 8848 网上超市(http://www.8848.com/)

著名的电子商务实例。全用 ASP 程序开发,其中的程序组织结构是很好的学习实例。 当然,在现在的网络上还有很多用 ASP 开发的 Web 数据库实例,读者可以访问这些站 点,从而增加对 Web 数据库系统的理解。

B.2 ASP 与 SQL 资料网站

在开发程序时,开发者往往需要更多的帮助来解决各个方面的问题,而这些问题的答案大多都能从网络上找到,因为现在网络上有很多关于 ASP 和 SQL Server 的技术站点。

1. 中国动感技术网络——ASP 技术网站 (http://www.chinaasp.com/)

1998 年 6 月,由从事 ASP 开发的飞鸟等网友自发组织成立了"飞鸟之家"个人站点, 对 ASP 技术在国内的应用进行研究和交流,并不断地吸纳相关网友参与实践和理论性的探 讨。1999 年 3 月,为了使 ASP 技术在国内得到更好的发展和应用,决定以半商业化、半兴 趣化模式发展 ASP 技术讨论站点,并申请国际域名 CHINAASP.COM,组建一个国内专业 的 ASP 技术讨论站点,其中包含文章专栏、FAQ、组件、论坛等栏目,汇集各类网络技术, 方便网友查找技术资料。

2. ASP 中华网 (http://www.aspcn.com/)

提供 ASP 教程、ActiveX 组件和论坛等,其中的产品均有免费版本,并且提供全部源 代码。现在比较有影响力的产品是[aspcnUP 上传组件]。

3. 中国 ASP 学习站 (http://www.learnasp.com.cn/)

提供 ASP 教程、ASP 论坛、技术文章、组件、源代码。

4. ASP 网络之门 (http://www.asp.com.cn/asp/)

包含 ASP 搜索、链接、代码、源程序、ASP 论坛等。

5. ASP 启航工作室 (http://www.aspchina.com/)

包含 asp, php, sql server 等网络编程技术和实例,以及源程序的下载。

6. ASP 互助联盟 (http://www.joyeasy.net/)

成立于 1999 年 6 月 20 日,以个人主页的形式存在了两个月。鉴于联盟发展的形式,申请了国际域名及虚拟主机。此站采取会员制,包括 ASP 技术论坛、ASP 教程、ASP 资料等。

7. 中国 ASP (http://www.chinaasp.com/)

包含论坛、BBS、CHAT、聊天室、留言本和站点访问统计报告系统等。

8. ASP 动态网站 (http://kujile.shangdu.net/)

包含 ASP 资料、源码、VB 脚本、JAVA 脚本、精选教程和编程必备工具等。

9. 飞鹰 ASP 网 (http://www.aspcool.com/)

包含 ASP 教程、ASP 程序下载、免费留言板、免费计数器等。

10. ASP 动感网络 (http://active.silversand.net/)

优秀个人网站,包括 ASP 教学、程序源代码和论坛等。

11. ASP 网络之星 (http://www.coolbel.com/wangchuang)

介绍 asp、dhtml、Web 数据库知识、有相关网络教程及 asp 程序。

12. ASP 专题网 (http://go.163.com/~painter)

包含 asp, sql, vbscript, javascript 等语言的代码,软件和教程,还有论坛、留言板等。

13. ASP 动感先锋 (http://wodeail.on.net.cn/)

包含 ASP 学习资料、技巧文摘、ASP 下载、论坛等。全新版本的论坛,较高的人气, 完全为解决问题而准备。

14. asp 在线 (http://asp.isme.net/)

包含 ASP 教程、ASP 编程、ASP 参考、建站技术、ASP 下载。但是必须先登录才可进入。

15. ASP 动感技术网络站点 (http://www.active.com.cn/default.asp)

讨论 asp, nt, sql, java, xml, php, delphi。

16. ASP 下载特区 (http://aspnet.on.net.cn/)

专业的 ASP 下载站点,提供大量 ASP 的源码,组件及教程下载。

17. 网魂 (go18.163.com/~ng/)

ASP, PHP 等专题及软件下载,还有部分电子书籍。

18. ASP CHINA (http://yup.2699.com/)

包含 ASP 教程、源码、技术文章,以及 ASP, VBS, VJS 在线参考手册。

19. 香港 CGI 及 ASP 信息综合网 (http://k666.126.com/)

包含免费软件、免费论坛、计数器、聊天室、邮件列表、六合彩信息等。

20. 第二频道-ASP 专卖店 (http://www.tv2nd.com/)

ASP 程序下载,网站后台数据库定制,ASP 技巧,程序相互交易和动感论坛等 。

21. ASP 学园 (http://easp.126.com/)

ASP 教程, 文章, 例程, 组件, 下载等。

22. ASP 资讯站 (http://landinfo.163.net/)

包含 ASP 教程、编程、源代码、组件、工具和下载等,以及 asp, powerbuilder, VBScript 和 Delphi 等教程。

23. 动感小屋 (go18.163.com/~peterzhj/)

这是一个个人网站,包含 ASP 简介、精选文章、源代码、实例和论坛等。

24. ASP (http://cuteasp.126.com/) [GB]

关于 ASP 方面的学习经验和精彩事例。

25. ASP 爱好者 (http://aspfans.yeah.net/)

一个专业的 ASP 编程网站,包括大量源程序、组件、技术文章和教材等。

26. ASP 三角洲 (http://www.asp800.net)

包含 ASP 教程、SQL 教程、源代码下载、聊天室实例等。

27. ASP 视窗 (http://aspwindows.yeah.net/)

此网站以 asp 为核心,并介绍 JAVASCRIPT、VBSCRIPT、css、sql 和以网页制作相关内容。

28. ASP 天地 (http://elewolf.home.chinaren.com/)

大量 ASP 程序下载,所有程序均经过站长亲自测试,配有演示。

29. ASP 实战宝典 (http://phphelp.easthome.net/)

ASP 的站点,内容新,包含 ASP 简介、实例、组件、源代码、教程、论坛及相关资源。

30. ASP 精品屋 (http://aspage.yeah.net/)

从国内海外搜集 ASP 最新最好的资料,同样包含 ASP 实例、组件源代码软件工具和 论坛等。

31. ASP 资源登记查询中心 (http://aspindex.126.com/)

ASP 资源,免费下载,个人网站,免费 ASP 空间,免费代码和论坛等。

32. 蜘蛛精 ASP 日报 (http://dailyasp.yeah.net/)

蜘蛛精翻译的 ASP 资料,包含源代码、教程和相关资源等。

33. ASP 天空 (http://aspsky.easthome.net/)

提供 ASP 文章、源程序、控件,以及 flash 教程、html 制作技巧等。

34. asp/php 网络编程 (http://asp2001.yeah.net/)

小黑侠的主页,内容包括网站建设:ASP,PHP,MYSQL等。

35. ASP 时代 (http://asptimes.126.com/)

包含 asp 教学、下载、论坛和聊天室等。还有一个不错的 CGI 世界。

36. ASP 加油站 (http://towncat.126.com/)

包含 ASP, PHP, NT 和 SQL Server 等, 提供技巧、源代码。

37. 顶尖 ASP 技术园 (http://djweb.go.163.com/)

是一个人网站,关于 asp 的站点,里面有很多 ASP 的文章、ASP 教程、组件。

38. 完全 ASP (http://yesasp.yeah.net/)

一个全新的 ASP 站点,丰富的源程序下载。

39. ASP 文摘站 (http://aspnow.yeah.net/)

个人网站,收集了大量的 ASP 文章,并且提供源码下载,列有综合文章。

40. ASP 资源网 (http://superasp.yeah.net/)

详细介绍 ASP 编程,动态网页制作。

41. 第五空间 (http://dq94.126.com/)

网络开发技术、数据库应用、图像设计技巧,如 Flash, firework 和 asp 等。

42. ASP 网站 (http://ksren.126.com/)

关于 ASP 的文章、技巧、代码下载。

43. jaky 的 ASP 空间 (http://jaky.2911.net/)

本主页是一个专业介绍asp与Web数据库技术的网站,已经实现数据库管理。包含ASP, VBScript, CGI, PowerBuilder, Delphi 等编程技术和自编软件,编程开发,实用工具等软件下载。

44. 深入浅出 ASP (http://flyto.8u8.com/)

深入浅出学习 ASP 及其相关知识。

45. 新海动力天地 (http://www.newsea.net/board/sboard2.php3)

提供 PHP, ASP, Java, Javascript 各类源代码教程及论坛。

46. 动能网站制作指南(http://sayyes2000.yeah.net/)

一个对 ASP 进行全面介绍的网站。

47. 孤狼 ASP (http://wolfwph.yeah.net/)

介绍 asp, html, css 等, 包含免费资源, 源码下载。

48. ASP 情缘网 (http://www.aspweb.onchina.net/)

包含 ASP 教程、源代码、免费资源、论坛和 Javascript 等。

49. ASP 极品斋恶狼工作室 (http://happybook.myrice.com/)

一个 ASP 教学与交流的网站。

- 50. VB&ASP (http://www.itshoper.com/aspvb.htm)
- 一个个人网站,包含 VB 和 ASP 的资料及编程技巧。
- 51. asp 专卖店 (http://wuqing.coolbel.com/)
- 关于 ASP 程序、文章、技巧、组件、数据库、ASP 资源的网站。
- 52. 轻松 ASP (http://www.coolbel.com/lonewolf/)
- 一个网页制作爱好者的资源宝库, 收藏了以 ASP 为主的网页制作资料。
- 53. 先锋 ASP 精华网 (http://ksk999.yeah.net/)
- 最新的 ASP 文章、源码、组件、教程,还包括一些原创作品 。
- 54. asp 空间站 (http://asphelper.yeah.net/)
- 提供 asp 技术教程、组件下载、源码。
- 55. ASP 步步高 (http://realwax.coolbel.com/)
- 为 ASP 初学者提供一个入门的指导,并提供本人学习 ASP 的经验。
- 56. ASP 资源网 (http://myhome.py.gd.cn/keven/index.asp)
- 这里是 ASP 的世界,有大量的教程、代码下载。
- 57. ASP 酷站 (http://web.nyist.net/~yasp/)
- 包含 ASP 源程序、基础语法、软件下载、教材和相关工具等。
- 58. 命古射日 ASP 空间 (http://mgshsun.myrice.com/)
- 包含自动友情链接系统、搜索引擎注册、教程下载、软件下载及贺卡等。
- 59. Flash&Asp (http://leonfs.top263.net/)
- flash&Asp 论坛,由佛山市第一中学和浙江大学光电系联办。
- 60. 至爱 ASP (http://oyes.coolbel.com/)
- ASP、CGI、JAVA、DHTML、FLASH、VBSCRIP。

附录 C SQL 中的保留字

在本书的第3章曾经提到过 SQL Server 中的保留关键字,在 SQL Server 中进行数据库 设计时,数据库名称、数据表名称、字段名称都不能使用这些关键字。

一旦出现关键字冲突,程序运行时会出现非法操作,而且这样的错误很难查找。因此, 在设计时一定避免使用这些关键字。

ABSOLUTE	ANY	ALL	ASC
ADD	ARE	Alphanumeric	ASSERTION
ADMINDB	AS	AUTHORIZATION	ALTER
ALTERT ABLE	AUTO	And	Avg
•	INCREMENT		
As			
BEGIN	Between	BINARY	BIT
BIT_LENGTH	BOOLEAN	BOTH	BY
BYTE			
CASCADE	CATALOG	CHAR	CHARACTER
CHAR_LENGTH	CHECK	CLOSE	CLUSTERED
CONVERT	Count	COUNTER	COLUMN
COMPRESSION	COMMIT	COMP	CONNECT
CONSTRAINT	CONSTRAINTS	CONNECTION	CONTAINER
COALESCE	CONTAINS	COLLATE	COLLATION
CREATE	CURRENCY	CURRENT_DATE	CURRENT_TIME
CURRENT_USER	CURSOR		
DATABASE	DATETIME	DATE	DEC
DECIMAL	DECLARE	DELETE	DESC
DISCONNECT	DISTINCT	DISTINCTROW	DAY
DISALLOW	DOMAIN	DOUBLE	DROP
EXEC	Eqv	EXECUTE	EXIST
EXTRACT	FALSE	FETCH	FIRST
FLOAT	FLOAT8	FOREIGN	FROM
FROM Clause	GENERAL	GRANT	GROUP
GUID	HAVING	HOUR	
IDENTITY	IEEESINGLE	IEEEDOUBLE	IGNORE
IMAGE	Imp	INSENTITIVE	INSERT
INPUT	INSERTINTO	INT	INTEGER
INTEGER4	INTERGER1	In	INDEX
INDEXCREATED	INNER	INTERVAL	INTO
Is	ISOLATION		
JOIN	LONGTEXT	KEY	LOWER
LANGUAGE	LAST	LEFT	Level
Like	LOGICAL	LOGICAL1	LONG
LONGBINARY	LOGNCHAR	MATCH	Max
MEMO	Min	MINUTE	MOD

MONEY	MONTH		
NATIONAL	NCHAR	NONCLUSTERED	Not
NTEXT	NULL	NUMBER	NUMERIC
NVARCHAR	Outer	OUTPUT	OWNERACCESS
OCTET_LENGTH	OLEOBJECT	ON	OPEN
POTION	Or	ORDER	PAD
PARAMETERS	PARTIAL	PASSWORD	PERCENT
PIVOT	POSITION	PRECISION	PROC
PROCEDURE	PREPARE	PRIMARY	PRIVILEGES
PUBLIC			
REAL	REFERENCES	RESTRICT	REVOKE
RIGHT	ROLLBACK	SCHEMA	SECOND
SELECT	SELECTSCHEMA	SELECTSECURITY	SET
SHORT	SINGLE	SIZE	SMALLDATETIME
SMALLINT	SMALLMONEY	SOME	SPACE
SQL	SQLCODE	SQLERROR	SQLSTATE
STDEV	STDEVP	STRING	SUBSTRING
SUM	SYSNAME	SYSTEM_USER	TABLE
TABLEid	TEMPORARY	TEXT	TIME
TIMESTAMP	TIMEZONE HOUR	TIMEZONE MINUTE	TINYING
ТО	ТОР	TRAILING	TRANSACTION
TRANSFORM	TRANSLATE	TRANSLATION	TRIM
TURE			
UINON	UNIQUE	UNIQUEIDINTIFIER	UNKNOWN
UPDATE	UPDATEIDENTITY	UPDATEOWNER	UPDATESECURITY
USAGE	USER	USING	VALUE
VALUES	VARYING	VARBINARY	VARCHAR
VAR	VAP	VIEW	WHEN
WHENEVER	WHERE	WITH	WORK
XOR	YEAR	YESNO	ZONE

-

附录 D 文件下载使用说明

为了方便读者浏览实例,快速建立后台数据库,并从读者的经济利益考虑,本书的样 例代码及相关文件以自由下载方式提供。需要者请访问北京科海培训中心网站:

http://www.khp.com.cn

1. 数据库导入

本书中的所有实例均以两种文件形式提供:

◆ SQL Server 数据库备份文件

◆ access 数据库文件

所有的数据库文件在 database 目录下都可以找到。

SQL Server 数据库备份文件使用 SQL Server Enterprise 中的数据库恢复向导,就可以恢复,具体的恢复操作可以参考第3章。

对于提供的 access 数据库文件, 使用 SQL Server 中的 DTS Import/Export Wizard (数据 转换服务向导)可以实现 access 数据库到 SQL Server 数据库的转换, 具体的转换操作过程 如下:

(1) 打开 SQL Server Enterprise,建立同本地 SQL Server 的连接,然后,在"tools"
 菜单中选择 "Import Data",将运行 DTS Import/Export Wizard (数据转换服务向导),如图 D-1 所示。



图 D-1 DTS Import/Export Wizard (数据转换服务向导)

(2) 在图 D-1 中单击"下一步"按钮,得到的如图 D-2 所示的对话框。需要选择 Data Source (数据库类型),这里在下拉菜单中选择"Microsoft Access",下拉菜单如图 D-3 所示。选择好数据库类型后,单击"File name"后的 按钮,选择目标文件。这里以 BBS 论坛实例的后台数据库文件 bbs.mdb 为例来说明。图中的"Username"、"Password"不需要填写任何信息。

: DIS Import/Export Wize	ard	R
Choose a Bata Source From where do you of the following so	e want to copy data? You can copy data from or ources.	•
Quin Fourier 9		
To comment. You uppy new	inited a statistical and provide it was many and convey The specify opportunity splitters	d 7, 7
fores	E:\database\bbs.mdb	
energy Constant		
	Barro	
	KL-POT-SON RA	

图 D-2 选择数据库文件



图 D-3 数据库类型下拉菜单

选择好数据库文件后,单击图 D-2 中的"下一步"按钮,出现如图 D-4 所示的对话框。

Legender -		rosoft OLE DB P	rovider for SQ	LServer	
O					
O					
4	amana.				
E					

图 D-4 目标数据库设置

(3) 在图 D-4 所示的目标数据库设置对话框中,选择目标 L Server,这里选择的是 "DIDO",在两种身份验证方式中选择其中的任何一个都可以,这里选择的是"User Windows Authentication",在"Database"(目标数据库)下拉菜单中选择"<new>",如图 D-5 所示。将弹出如图 D-6 所示的 Create Database(新建数据库)对话框,在其中输入 Name (数据库名称),并且设置数据库文件和日志文件的大小限制,然后单击"OK"按钮。



图 D-5 选择目标数据库

将返回图 D-4 所示的目标数据库设置对话框,并且单击"下一步"按钮,将得到如图 D-7 所示的对话框。







图 D-7 选择转换方式

(4) 在如图 D-7 所示的"选择转换方式"对话框中,有两种转换方式可以选择:

- Copy table(s) and view(s) from the source database
- Use a query to specity the data to transer

选择第一种转换方式,将完全复制原来的数据库到 SQL Server 中。

选择第二种转化方式,将允许使用 SQL 语句的查询方式进行数据转换,这样的方式比前一种更加灵活。

这里采用第一种方式。

选择好转换方式后,单击"下一步"按钮。将得到如图 D-8 所示的对话框。

्र, DTS Im	port/Export Wizare	1		
Select You cop (t Source Tables can choose one or y the schema and c .) to transform th	and Views r more tables or views to copy. data as it appears in the source he data using ActiveX scripts.	You can e or click	
		ale to make the		
	board	(bbs)[dbo][board]		
	letter member	(bio) (dbo) (letter) (bio) (oten (or subset)		
	message	[bbs].[dbo].[message]	<u></u>	
	Seera 1	The second s		

图 D-8 Select Source Tables and Views (选择数据库表和视图)

(5) 在图 D-5 中列出了 bbs.mdb 中所有的数据表,在这里可以浏览数据库中每一表中的所有内容,只需要选中相应的表,然后单击 "Preview" 按钮。这里选中的是 member 表,单击 "Preview" 按钮后弹出如图 D-9 所示的对话框。

ain de mail@75851 1	and and a second se I second	and and a second se I second	nen en la company a desente desente de la company de la El company de la company de
13	huhu	pett	huhu
14	wju	wiju	wiju
15	guyajin	边宇	惠惠
18	benfox	吴珺	笨狐猩
19	benfox1	吴珺	笨狐猩
20	ivshaohua	昌少华	呼呼
21	rain	wxrain	rain
22	уурргг	ур	УP

图 D-9 View Data (浏览数据)

从图 D-9 中可以看到 member 表中的所有内容,单击 "OK" 按钮将返回图 8 中所示的 对话框。这里选中数据库表,然后单击"下一步"按钮。将得到如图 D-10 所示的对话框。

(6) 在图 D-10 中需要选择数据库转换计划,这里选择 Run immediately,如图 D-10 所示。

+ DTS	Import/Export Wizard		
Sev	e, schedule, and repli Specify if you want to say replicate the data or sche later time.	cate package we this DTS package. You may al edule the package to be executed	so at a
	When V Bur Inneshidy Seheddli DTS package Occurs every T dig(s) s	C Lise replication to publich due for later execution LOBD 40.	
	Save	SQL Server SQL Server Meta Data Servio Siguatured Storage File Moull Basis File	53
	104 - 1 <u>< F</u>	-#@) <u>T-#@)</u> RM	J #Bb J

图 D-10 数据转移计划选择

单击"下一步"按钮,将出现本次数据库转换的描述,如图 D-11 所示。单击"完成" 将开始数据转换。



图 D-11 本次数据库转换操作描述

(7)数据转换需要一定的时间来完成,如图 D-12 所示,将显示数据转换的进度。当数据转换完成时,将提示数据库转换完成。

An office the subscription of the		
Microsoft Account	Missonak SQL Ser	v a r
higher		
	in ter calls	
Helur:		
Step Name	Status	
Create Table [bbs] (dbo] [board] Step	Complete	1
Copy Data from board to [bbs].[dbo].[board] Step	Running (0)	
Create Table [bbs].[dbo].[letter] Step	Complete	
Opy Data from letter to [bbs].[dbo].[letter] Step	Waiting	
Create Table [bbs].[dbo].[member] Step	Complete	
Ocopy Data from member to [bbs].[dbo].[member] Step	Waiting	
Create Table [bbs].[dbo].[message] Step	Complete	
		D

图 D-12 数据转移进度

至此,已经完成了 bbs.mdb 数据库的转换操作,已经在 SQL Server 中新建了一个 BBS 数据库。

其他的三个数据库文件 document.mdb、temp_document.mdb 和 search.mdb 的转换操作 同上。

2. 建立 ODBC 数据源

(1) 在 SQL Server 中建立 Login

按照程序的设置,在 SQL Server 中新建立两个用户,pett 和 dido,其密码分别设置为 123 和 dido,并具有访问数据库的权限。其中的 pett 用于 bbs 论坛数据库和搜索引擎数据 库的访问,而 dido 则用于稿件数据库和临时稿件数据库的访问。

(2) 建立 ODBC 数据源

建立的方法在本书的前面的章节中有详细的介绍,但数据源的名称必须符合程序中的 要求。分别为:

bbs 论坛数据库: dsn=bbs;uid=pett;pwd=123

正式稿件数据库: dsn=document;uid=dido;pwd=dido

临时稿件数据库: dsn=temp_document;uid=dido;pwd=dido

搜索引擎数据库: dsn=search;uid=pett;pwd=123

当然用户名和数据源名称读者也可以任意设置,但是必须改变程序中相关语句,在改变时,建议采取 Frontpage 中的全站范围内查找替换。

如果读者在本地机上还没有安装 SQL Server 的话,可以直接用下载的 ACCESS 数据库

文件建立数据源。

3. 改变存放文件的目录

本书中的稿件数据库实例中,需要操作一些文件,因此需要一些目录。这些文件和目录都已经放置在 example 目录下,这里假定读者的主发布目录是 c:\Inetpub\wwwroot\。因此,如果主发布目录是其他路径的话,则需要改变相关程序中的语句。

其中以下的文件中的 c:\Inetpub\wwwroot\example\document\temp_document\要替换为相应的目录,此目录用于存放临时稿件,因此最好不要放置在发布目录下,需要另建目录。

example/document/document_input/UPLOAD_check.asp

example/document/manager/del_all_temp.asp

example/document/manager/submit.asp

example/document/manager/check_temp.asp

example/document/manager/check_document3.asp

example/document/document_input/UPLOAD.asp

example/document/manager/manager_tempA.asp

example/document/manager/submit1.asp

example/document/manager/manager_tmepB.asp

example/document/newuser.txt 文件用于存放用户提交的申请信息,因此如果要改变其存放路径的话,需要更改 example/document/advance/register result.asp 中的目录设置。

如有问题请与作者联系, E-mail: <u>pett@263.net</u>。