

基于 ASP 技术的数据库浏览工具



学 院 计算机科学与技术

专 业 计算机科学与技术

年 级 2003 级

姓 名 李宇鹏

指导教师 张坤龙

2007 年 6 月 20 日

摘 要

数据库在各个行业中的应用就越来越广泛，对数据库内容的查询是最常用的数据库操作。但是对于最终用户来说，使用专门的 SQL 查询语言或者开发专门的查询应用程序存在很多困难，他们希望能够通过 Web 界面来查询远程的数据库。另外，对于数据库应用开发人员来说，一个简单但是功能丰富的数据库浏览工具也能成为其开发和测试时的得力帮手。

论文通过对 ASP 技术和同类软件的深入了解，设计并实现一个基于 ASP 技术的数据库浏览工具 ASPDBE(ASP-based DataBase Explorer)。通过该工具数据库管理员可以将自己管理的数据库发布到 Web 上，随后数据库用户就可以随时随地通过浏览器来浏览数据库中的内容。

关键字：数据库浏览；ASP 技术；ASPDBE。

目 录

第一章	绪论	1
1.1	数据库的浏览.....	1
1.2	课题的意义.....	2
1.3	国内外现状.....	2
1.4	论文的结构组织.....	4
第二章	ASP 技术	5
2.1	ASP 简介.....	5
2.2	ASP 脚本基础.....	5
2.3	ASP 内建对象.....	6
2.4	ASP 的运行环境.....	7
2.5	ASP, PHP, JSP 比较	7
第三章	ASPDBE 的设计	8
3.1	体系结构.....	8
3.2	需求分析.....	11
3.3	概要设计.....	12
第四章	ASPDBE 的实现	17
4.1	用 ASP 连接数据库	17
4.2	参数传递.....	18
4.3	分页浏览.....	19
4.4	排序.....	22

4.5	验证码实现.....	23
第五章	总结.....	26
5.1	完成工作与收获.....	26
5.2	未来工作的展望.....	26
	参考文献.....	27
	外文资料	
	中文译文	
	致 谢	

基于 ASP 技术的数据库浏览工具

第一章 绪论

1.1 数据库的浏览

1.1.1 数据库浏览的概念

现阶段数据库已经是必不可少的工具，数据库浏览就是把数据库中的数据按一定规则显示给用户的过程。数据库浏览可以通过 web 表格，应用前端或 SQL 接口等方式实现，这些方法最终解释为 SQL 命令输入到数据库管理工具进行相应的操作，并把结果显示给用户。

1.1.2 浏览的对象

数据库浏览的主要对象是用户的数据和元数据。用户的数据顾名思义就是用户保存的，对用户具有实际意义的数据，是用户的数据。而元数据需要详细介绍一下。

元数据也是数据，数据本身也可以作为被描述的对象，这些描述它的数据就是元数据。在信息系统中一般把数据看成是独立的信息单元，不管这里的“数据”是一本书、一个网页、或者一个虚拟的 URL 地址。元数据可以出现在数据内部、独立于数据、伴随着数据或与数据包裹在一起。

传统的书目数据、产品目录、人事档案等都是元数据。在数据库中元数据主要存储表信息，数据库信息。离开元数据的数将是一盘散沙，无法提供有效的检索和处理。

在 ASPDBE 中允许用户访问用户数据和有限的元数据，例如表的字段的定义等。管理员则可以访问所有元数据。

1.1.3 数据库浏览的方式

在 ASPDBE 功能实现前，确定使用哪种浏览方式是很重要的，因为浏览方式不是像它看起来一样简单，数据库浏览方式决定 ASPDBE 的用户群体，和软件特征。

数据浏览的方式主要有 web 表格，应用前端或 SQL 接口等方式。前两种方式的用户一般是初级用户，如：客户，旅行代理商等，而是用 SQL 接口方式的用户一般是高级用户，应用程序员，数据库管理员等^[1]。

其中 web 表格面向的用户群体可以不受空间的限制，可以在世界各地，只要有 Internet 网就可以使用。这种方式可以提供比 SQL 接口方式更简单，更便于操作的人性化界面，为那些对 SQL 语言不了解的用户提供了方便，简单的

浏览工具。

基于 web 的数据库浏览工具的数据源一般在远程的服务器上,用户通过浏览器与服务器进行通讯。数据库管理员在服务器端对数据进行发布。以上这些特性使数据库可以在全球范围内被访问,使数据库更加广泛的应用在各个行业。

在 ASPDBE 中,面向的是 web 用户。这就要求它灵活性高,所以在本系统中应该选择 web 表格的浏览方式,对于用户的个别需求,在显示记录时也可以采取表单的形式。

1.2 课题的意义

数据库在各个行业中的应用就越来越广泛,对数据库内容的查询是最重要也是最常用的数据库操作。但是对于最终用户来说,使用专门的 SQL 查询语言或者开发专门的查询应用程序存在很多困难,他们希望能够通过 Web 界面来查询远程的数据库。另外,对于数据库应用开发人员来说,一个简单但是功能丰富的数据库浏览工具也能成为其开发和测试时的得力帮手。

课题的目标就是满足用户的这种要求,设计并实现一个基于 ASP 技术的数据库浏览工具。通过该工具,数据库管理员可以将自己管理的数据库发布到 Web 上,随后数据库用户就可以随时随地通过浏览器来浏览数据库中的内容。

在课题的开发过程中,可以学习动态网页与数据库相结合的应用程序设计的相关知识与方法。在课题中可以使我了解本课题在现阶段国内外的水平。

1.3 国内外现状

1.3.1 Access

Access 是微软公司 OFFICE 软件中的一个组件,是一个小型的数据库。Access 数据库能够进行数据表设计、可视查询设计、SQL 查询语言、窗体设计、报表设计、Web 页设计、宏设计和 VBA 程序设计,Access 数据库作为面向对象的关系型数据库有着各种完善的功能。

Access 主要采用应用前端的浏览方式对数据库进行浏览。Access 的浏览界面类似于 HTML 中 iframe 框架的结构。界面分为两侧,左侧显示用户可选的选项,右图则显示用户查询的结果,如图 1-1 所示。

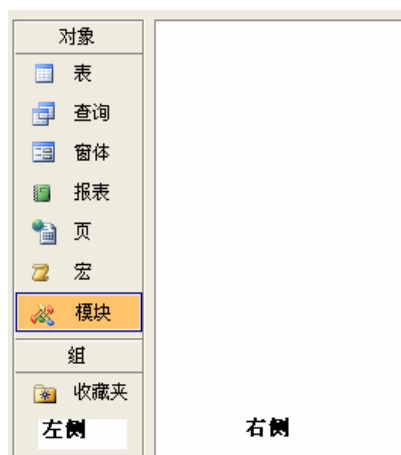


图 1-1 Access 浏览界面

1.3.2 Oracle

ORACLE 是一种适用于大型、中型和微型计算机的关系数据库管理系统, 它使用 SQL(Structured query language)作为它的数据库语言。ORACLE 数据库由三种类型的文件组成: 数据库文件、日志文件和控制文件。

数据字典是由 ORACLE 自动建立并更新的一组表, 这些表中记录用户的姓名、描述表和视图以及有关用户权限等的信息。数据字典是只读的, 只允许查询。也就是说数据字典是一种数据库资源, 每个用户都可以访问数据字典, DBA 可通过数据字典来监视 ORACLE RDBMS 的使用, 并帮助用户完成其应用。ORACLE RDBMS 本身也要利用数据库字典来管理和控制整个数据库。

1.3.3 SQL server 2000

SQL Server 是 Microsoft Windows Server System 的核心组件。SQL Server 是基于服务器端的中型的数据库, 可以适合大容量数据的应用, 在功能上管理上也要比 Access 要强得多。

SQL Server 主要采用 SQL 接口的浏览方式对数据库进行浏览。SQL Server 提供了 SQL 查询分析器, 用户直接输入 SQL 语句对数据库进行操作。SQL Server 也提供了对象浏览器, 用户可以直接查看对象信息。SQL Server 2000 的界面截图如图 1-2 所示。

从上面的介绍可以看出 SQL Server 2000 更适合经验丰富, 对 SQL 语句非常熟悉的数据库管理员使用, 但不适合普通用户使用。因为大多数情况下使用数据库浏览工具的用户是不懂 SQL 语句的。ASPD BE 面向的用户恰恰是这些普通用户, 所以 ASPD BE 的界面是基于 web 的, 更适合普通用户的查询。

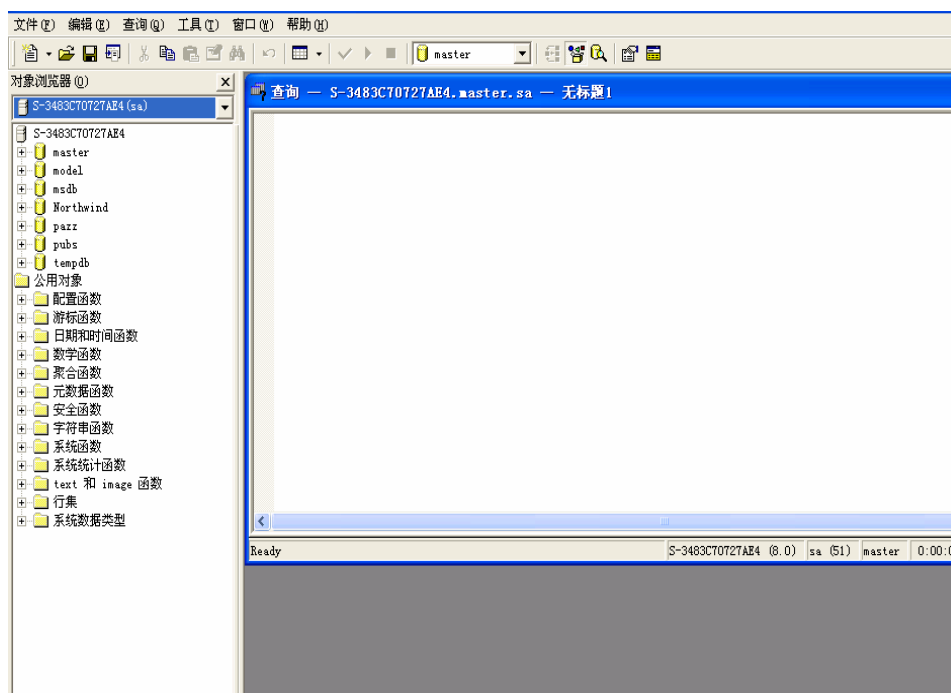


图 2-1 SQL Server 2000 界面图

1.4 论文的结构组织

论文的目标是使用 ASP 技术设计与实现一个使用方便、功能全面的名为 ASPDBE（ASP-based DataBase Explorer）的数据库浏览工具。

本文第一章绪论，阐述了本课题的意义，并对国内外同类软件进行比较。本文第二章 ASP 技术中，详细介绍 ASPDBE 中的主要技术——ASP 技术的脚本基础和内建对象，并对 ASP，PHP，JSP 进行比较。本文第三章 ASPDBE 的设计，详细介绍了系统的设计过程，包括体系结构的选择，功能模块的划分。第四章 ASPDBE 的实现，介绍了 ASPDBE 实现过程中遇到的主要技术问题和解决方法。第五章总结，对所作的工作进行总结并对未来的工作进行展望。

第二章 ASP 技术

2.1 ASP 简介

ASP 技术是 Microsoft 公司开发的一套全新的服务器端脚本程序环境，是微软用来建立动态网页的解决方案。它可以根据客户端的不同请求，在服务器端经过相应的 ASP 程序处理生成不同的静态 HTML 页面并传回给浏览器。ASP 有很多优点，这些优点是它成为当今使用最多的脚本设计环境。

ASP 内含于 IIS 3.0 以上的版本之中。通过 ASP 我们可以结合 HTML 网页、ASP 指令和 ActiveX 元件建立动态、交互且高效的 WEB 服务器应用程序。有了 ASP 你就不必担心客户的浏览器是否能运行你所编写的代码，因为所有的程序都将在服务器端执行，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。ASP 的执行过程如图 2-1 所示：

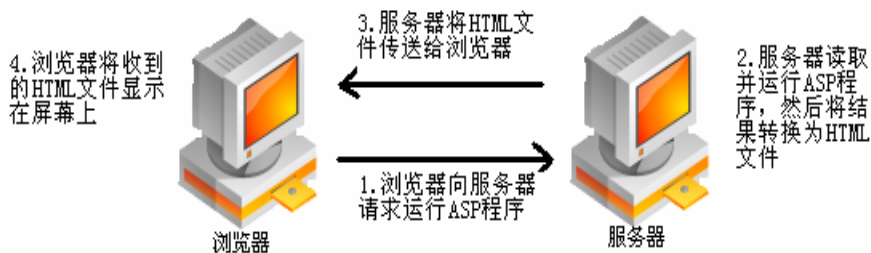


图 2-1 ASP 程序的执行过程

2.2 ASP 脚本基础

ASP 本身并不是一种脚本语言，它只是提供了一种使镶嵌在 HTML 页面中的脚本程序得以运行的环境。ASP 的脚本语言可以是 VBScript 或者 Jscript，也可以是两者的结合，所以任何一种文本编辑器都能编辑 ASP 的脚本，当然不同的编辑器的开发调试效率是不一样的。

VBScript 是 IIS 默认的脚本语言，当然可以在 IIS 中更改默认的脚本语言。如果改成 Jscript，那么默认的脚本语言就是 Jscript。本文代码主要用 VBScript 编写。

1. VBScript

Microsoft Visual Basic Scripting Edition 是程序开发语言 Visual Basic 家族的最新成员，它将灵活的脚本应用于更广泛的领域，包括 Microsoft Internet Explorer 中的 Web 客户机脚本和 Microsoft Internet Information Server 中的

Web 服务器脚本。

VBScript 使用 ActiveX Script 与宿主应用程序对话。使用 ActiveX Script, 浏览器和其他宿主应用程序不再需要每个脚本部件的特殊集成代码。

2. JScript

JScript 是一种解释型的、基于对象的脚本语言。JScript 不是其他语言的精简版, 也不是任何事物的简化。不过, 它有其局限性。例如, 用户不能使用该语言来编写独立运行的应用程序, 并且没有对读写文件的内置支持。此外, JScript 脚本只能在某个解释器或“宿主”上运行, 如 ASP、Internet 浏览器或者 Windows 脚本宿主。

2.3 ASP 内建对象

ASP 内置 Application、ASPErrors、Request、Response、Session、ObjectContext、Server 等 7 个对象, 而且每个对象有各自的属性、方法、集合或事件。下面介绍这 7 个对象。

1) Response 对象

这个对象可用来决定何时或如何将输出由服务器端传送至浏览器端。

2) Request 对象

这个对象可以用来获取由浏览器端返回服务器端的数据, 如浏览者输入的窗体数据、存储在浏览者本机计算机的 Cookie 等。

3) ASPErrors 对象

这个对象可让您取得 Script 发生错误的信息。

4) Server 对象

这个对象提供了服务器端最基本的属性及方法。

5) Application 对象

当有数个浏览者同时存取网页时, 这个对象可用来记录不同浏览器端共享的变量, 因为无论有几个浏览者同时存取网页, 都只会产生一个 Application 对象的案例。Application 对象默认的生命周期起始于 Web 服务器开始执行时, 终止于 Web 服务器结束执行时, 或超过 20 分钟没有任何浏览器读取网页。

6) Session 对象

这个对象也是用来记录浏览器端的变量的, 但和 Application 对象不同的是 Application 对象记录了所有浏览器端共享的变量, 而 Session 对象则是记录了个别浏览器端专用的变量。Session 对象默认的生命周期起始于 Web 服务器开始执行时, 终止于 Web 服务器结束执行时, 或超过 20 分钟没有任何浏览器读取网页。

7) ObjectContext 对象

这个对象用来链接 ASP 和 Microsoft Transaction Server。

2.4 ASP 的运行环境

由于 ASP 是一种服务器端的脚本语言，所以并不是所有的 web 服务器都具有这个功能。ASP 是由 Microsoft 公司推出的，当然在目前也只有 Microsoft 公司推出的服务器能实现 ASP 的强大功能，其他的一些服务器在安装了 ASP 组建后才能实现这个功能。

Microsoft 公司推出的支持 ASP 的 Web 服务器有以下几个：

Microsoft Internet Information Server version （简称 IIS）；

Microsoft Peer Web Services；

Microsoft Personal WebServer （简称 PWS）。

其中前两个主要是为 Microsoft 公司的 Windows NT/2000/XP 系统开发的，当然它们的功能、稳定性及安全性都要强的多。目前大型的站点大多是用 IIS 作为服务器，而 PWS 是为一般的 PC 用户开发的。

2.5 ASP, PHP, JSP 比较

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C,Java 和 Perl 语言的语法，并耦合 PHP 自己的特性,使 WEB 开发者能够快速地产出动态页面。PHP 可以编译成具有与许多数据库相连接的函数。PHP 与 MySQL 是现在绝佳的组合。但 PHP 提供的数据库接口支持彼此不统一，这是 PHP 的弱点。

JSP (java server page) 是一种用来产生动态网页内容的规范，以 Java 的技术为主，并具有跨平台和跨网页服务器支持的特性。这样的技术在网络上的应用相当广泛。JSP 也可以嵌入在静态的 HTML 网页中，来产生动态网页。

在目前 windows 操作系统在市场占很大份额，ASP 执行于 windows 服务器平台，这个特点使 ASP 具有其它两种技术无法比较的优势，可以在 LAN 和 Internet 之上建立一种统一的软件构件框架。其中 ActiveX 服务器组件具有无限可扩充性，可以使用多种程序设计语言编写所需的 ActiveX Server Component。它能够让软件开发者很方便、快速地在 Internet、Intranet 网络环境里，制作或提供生动活泼的内容与服务、编写功能强大的应用程序。这些优点正是 ASPDBE 所需的，因为 ASPDBE 是基于 web 的数据库浏览工具，这就要求使用更加强大动态网页的功能，ASP 是好的选择。

第三章 ASPDBE 的设计

3.1 体系结构

在系统设计之前，决定采用哪种体系结构来实现是非常重要的。有一个好的体系结构会有助于程序的实现并且能更好的实现所要求的功能。下面介绍 B/S 模式、C/S 模式以及二层结构与三层结构。

3.1.1 B/S 模式和 C/S 模式比较

1. B/S 模式

B/S 就是 Brower/Server 的缩写，在这种模式下用户通过浏览器直接访问服务器，用户的工作界面在浏览器中实现。客户机只要安装了浏览器就可以在任何联网的地方访问服务器。浏览器没有限制，可以是 Internet Explorer、Netscape Navigator 等，而服务器端可以使用 SQL Server 2000、Oracle、Access 等多种数据库。这种模式的优点很多，比如，客户端可以实现零维护，不需要安装其他软件，而在服务器端管理员可以轻松管理用户使用的数据。B/S 模式最大的缺点就是对网络依赖太高，一旦网络出现问题就会造成系统瘫痪。B/S 模式应用系统网络结构图如图 3-1 所示。

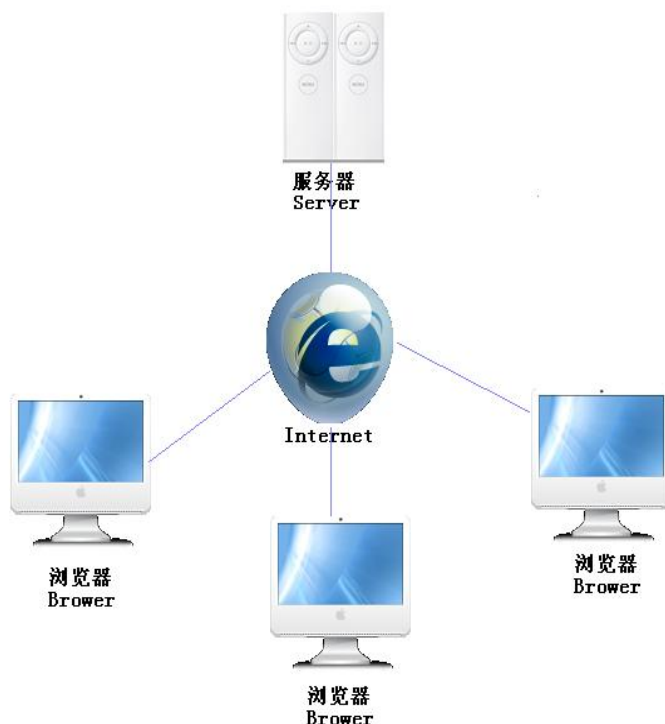


图 3-1 B/S 模式应用系统网络结构图

2. C/S 模式

C/S（Client/Server）结构，即大家熟知的客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。目前大多数应用软件系统都是 Client/Server 形式的两层结构。

客户应用程序是系统中用户与数据进行交互的部件。服务器程序负责有效地管理系统资源，如管理一个信息数据库，其主要工作是当多个客户并发地请求服务器上的相同资源时，对这些资源进行最优化管理。中间件负责联结客户应用程序与服务器管理程序，协同完成一个作业，以满足用户查询管理数据的要求。C/S 模式应用系统网络结构图如图 3-2 所示。

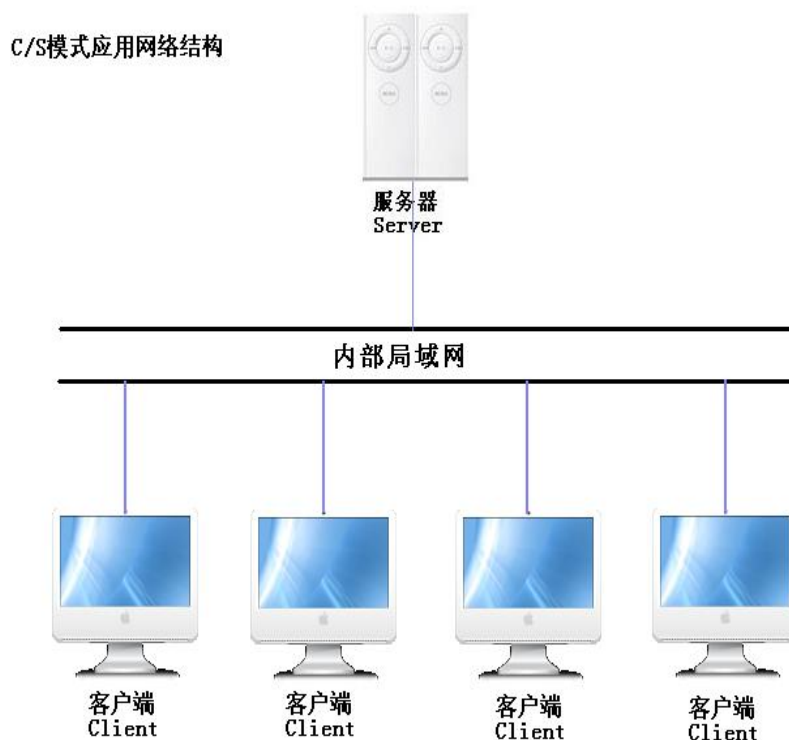


图 3-2 C/S 模式应用系统网络结构图

3.1.2 两层结构与三层结构

1. 两层结构

两层结构就是指客户端应用程序层和数据库服务器层。数据库服务器主要负责数据库的管理和维护等，而应用程序负责向数据库服务器层，然后通过一定的界面提供给用户，供用户对数据库进行浏览，查询，修改等操作。这种结构处理数据快，应用程序简单，但是客户端必须安装客户端程序，即庞大而且不便于维护。这种两层结构如图 3-3 所示。

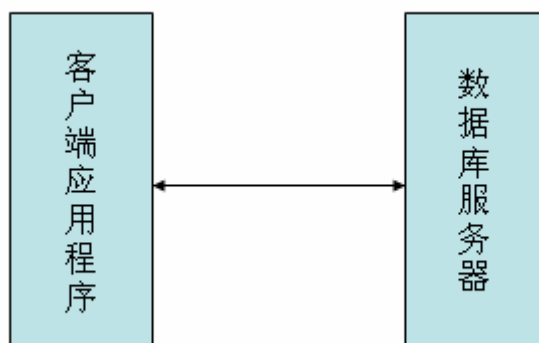


图 3-3 两层结构

2. 三层结构

三层结构也就是把传统 C/S 模式中的服务器部分分解为一个数据服务器与一个或多个应用服务器(Web 服务器),从而构成一个三层结构的客户服务器体系

第一层客户机是用户与整个系统的接口。客户的应用程序精简到一个通用的浏览器软件。

第二层 Web 服务器将启动相应的进程来响应这一请求,并动态生成一串 HTML 代码,其中嵌入处理的结果,返回给客户机的浏览器。

第三层数据库服务器负责协调不同的 Web 服务器发出的 SQL 请求,管理数据库。这种三层结构如图 3-4 所示。

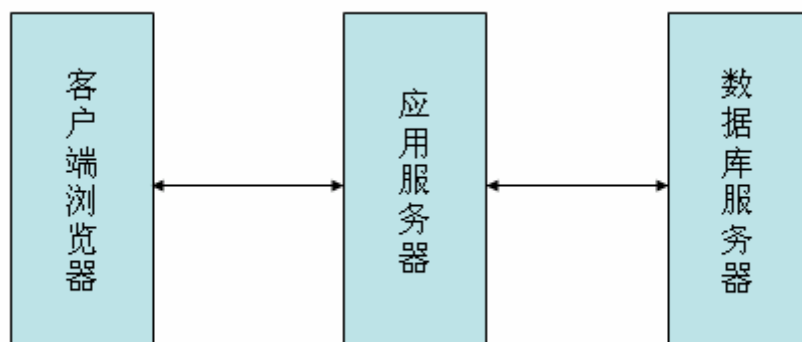


图 3-4 三层结构

3.2 需求分析

3.2.1 系统需求

在现在动态网页技术与数据库技术发展迅速的时代,许多人更希望可以用简单的、便于操作的 web 网页来浏览数据库。在许多行业中,浏览工具可以使信息快速,准确的被查询,比如财务可以随时通过浏览工具查看近期的财务支出,经理可以方便的查询员工的入职信息,超市可以准确的现实每样商品的售价以及进价。这些在现实中的应用还很多,基于 web 的数据库浏览工具已经成为计算机系统中必不可少的组成部分。

3.2.2 用例图

经过分析,绘制了本系统的用例图,图 3-5 清楚地显示了本系统所实现的功能,包括用户以及管理员拥有的权限操作。现在一一介绍一下。

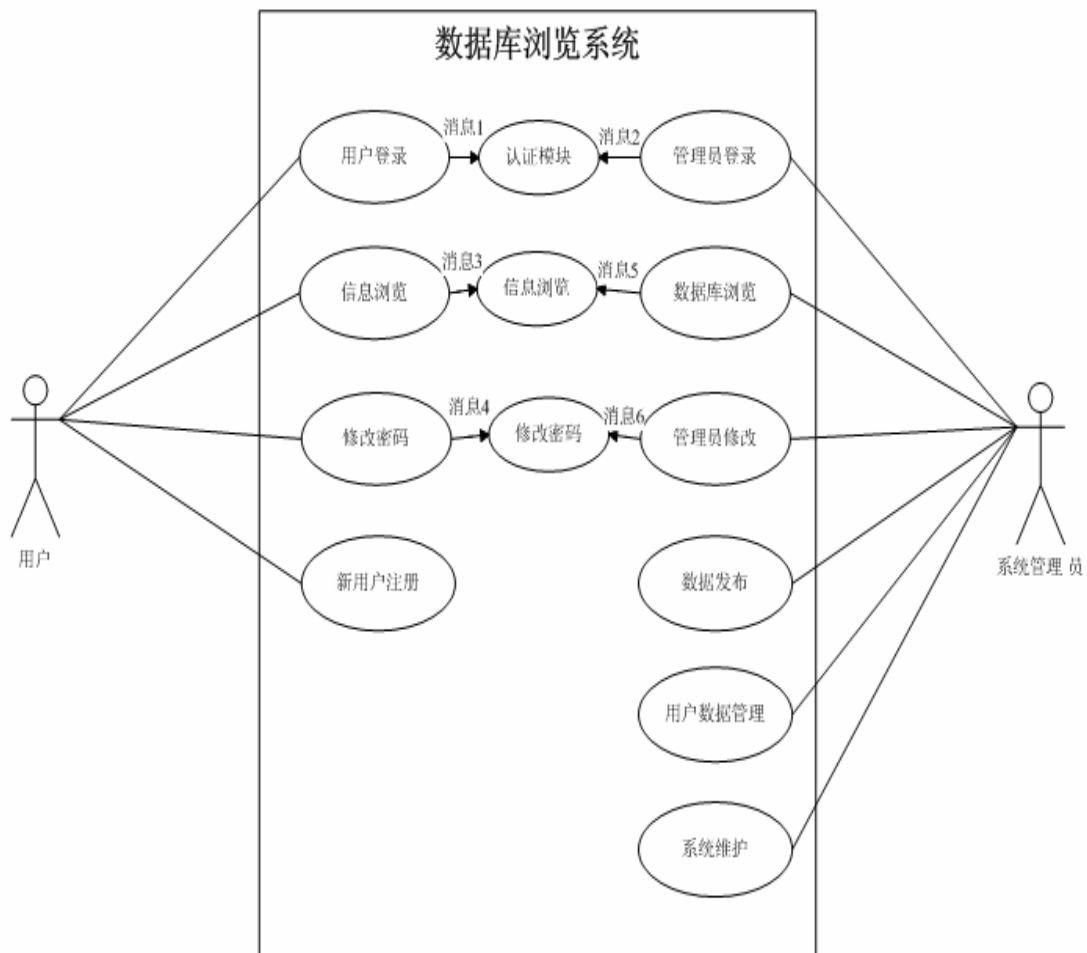


图 3-5 用例图

- 用户登陆：用户通过界面登陆。
- 修改密码：用户修改自己的登陆密码。
- 新用户注册：新用户申请账号。
- 信息浏览：新用户通过网页向服务器提交浏览请求，服务器端向用户发送请求内容，并且通过网页显示出来。
- 管理员登陆：用管理员账号登陆，拥有管理员权限。
- 管理员修改密码：与普通用户密码修改一样，管理员可以修改自己的密码，需要原密码的认证。
- 数据库浏览：管理员通过网页向服务器提交浏览请求，服务器端向用户发送请求内容，并且通过网页显示出来。
- 数据管理：管理员对数据进行修改、删除、添加等操作。
- 用户数据管理：管理员有权限对用户数据进行管理，比如可以修改用户权限，禁止用户浏览等。
- 系统维护：管理员对系统的服务器，数据库，网页框架进行必要的维护。

3.3 概要设计

3.3.1 体系结构的选择

在系统开发的初期，选择一个好的体系结构是至关重要的，体系结构直接影响系统开发功能实现与程序设计，所以一个合理的体系结构在系统开发初期非常重要。由本章第一节中介绍的 B/S 与 C/S 的比较以及两层结构与三层结构的比较中可以看到 B/S 模式比 C/S 灵活，客户端要求也比较低，而且三层结构与两层结构相比较也具有非常大的优势。而由上一节介绍的本系统需要实现的功能可以看出本系统的应用范围比较广泛，灵活性高，要求各客户端配置低，并且支持远程访问。这就确定了 ASPDBE 的体系结构应该选择 B/S 三层结构。ASPDBE 的三层结构图如图 3-6 所示。



图 3-6 ASPDBE 的三层结构

ASPDBE 中主要由用户、管理员、WEB 网站、DBMS 和数据库组成，这几个组成部分之间的关系如图 3-7 所示。

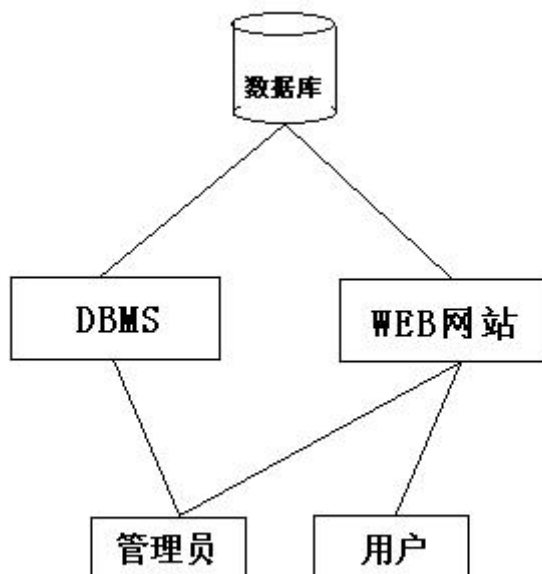


图 3-7 ASPDBE 组件关系

- **数据库**：数据源，存放用户所浏览数据。用户不能直接访问，只能通过 WEB 网站访问。
- **DBMS**：数据库管理系统，管理员可以直接通过 DBMS 访问以及管理数据库。
- **WEB 网站**：用户通过本地浏览器，直接访问。对数据库的浏览界面通过 WEB 网站发送给本地浏览器。

3.3.2 界面设计

ASPDBE 是基于动态网页技术的数据库浏览工具，ASPDBE 的所有页面是通过用户的操作动态生成的，这些操作主要是查询操作。

ASPDBE 的操作流程大致可以分为两个部分，一部分完成对数据库和表的选择，换句话说就是选择操作对象，另一部分对操作对象进行操作。这就要求 ASPDBE 中除了登录页面，需要一个主界面来完成以上两部分功能。说到这里，我们首先会想到采用 HTML 中的 frame 框架来实现设想的界面。Frame 框架主要作用是把浏览器分为几个部分每个部分显示不同的页面。在 ASPDBE 中我们把浏览器分为两个部分：第一部分是左框架，主要用来显示主目录，提示用户选择数据库或者表。第二部分是右框架，在右框架中，当用户进行操作后动态生成相应的页面，返回用户操作的结果。

Frame 框架使用户的使用变得简单，使 ASPDBE 的功能变得集中，大大提高了用户使用的效率。ASPDBE 界面框架如图 3-8 所示。

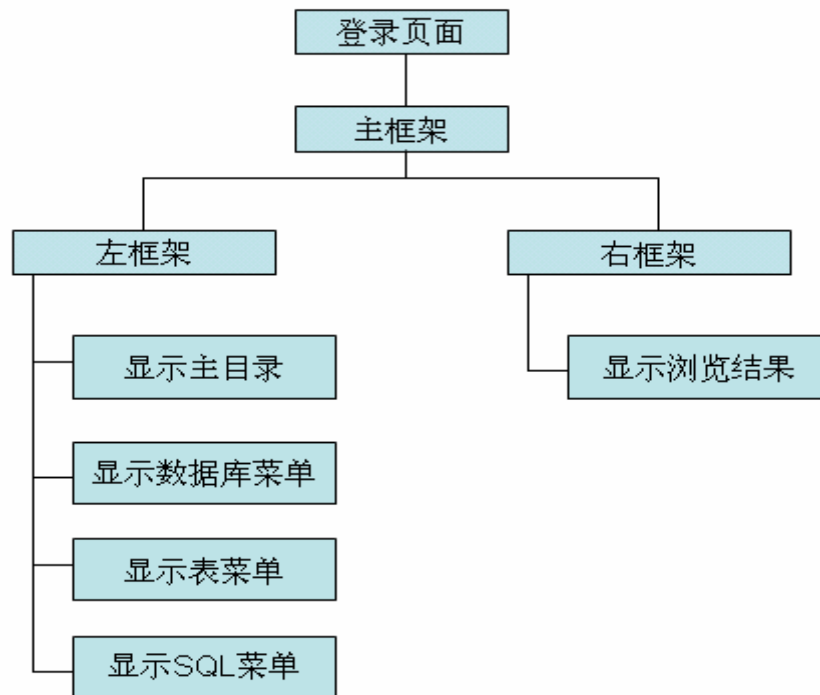


图 3-8 ASPDBE 界面框架图

- 1) 显示数据库菜单：显示用户可以选择用来浏览的数据库，利用下拉菜单实现。
- 2) 显示表菜单：显示用户可以选择用来浏览的表。
- 3) 显示 SQL 菜单：为用户提供直接输入 SQL 语句的界面。
- 4) 显示浏览结果：当用户完成操作后，把用户所要的结果显示出来。

3.3.3 模块设计

利用图 3-8 中的界面框架，将 ASPDBE 划分为三大模块：登陆模块、数据库操作模块和表操作模块。ASPDBE 的功能在这三个模块里实现。ASPDBE 的功能模块如图 3-9 所示。

1. 登录模块：包括认证模块和注册模块。

- 1) 登录认证：提供用户登录界面并对用户名，密码和验证码进行检验。
- 2) 用户注册：提供新用户注册页面，提示用户输入个人信息并提交注册。

2. 数据库操作模块：主要实现选择数据库和执行 SQL 语句的功能。

- 1) 选择数据库：因为在 ASPDBE 中有一个或多个数据库可以访问，所以需要选择需要浏览的数据库。在左框架中利用下拉框实现。
- 2) 执行 SQL 语句：用户可以直接输入 SQL 语句进行查询。为了数据库的安全，并且 ASPDBE 是数据库浏览工具，所以，这里的 SQL 语句仅限

查询语句，输入其他语句无效。

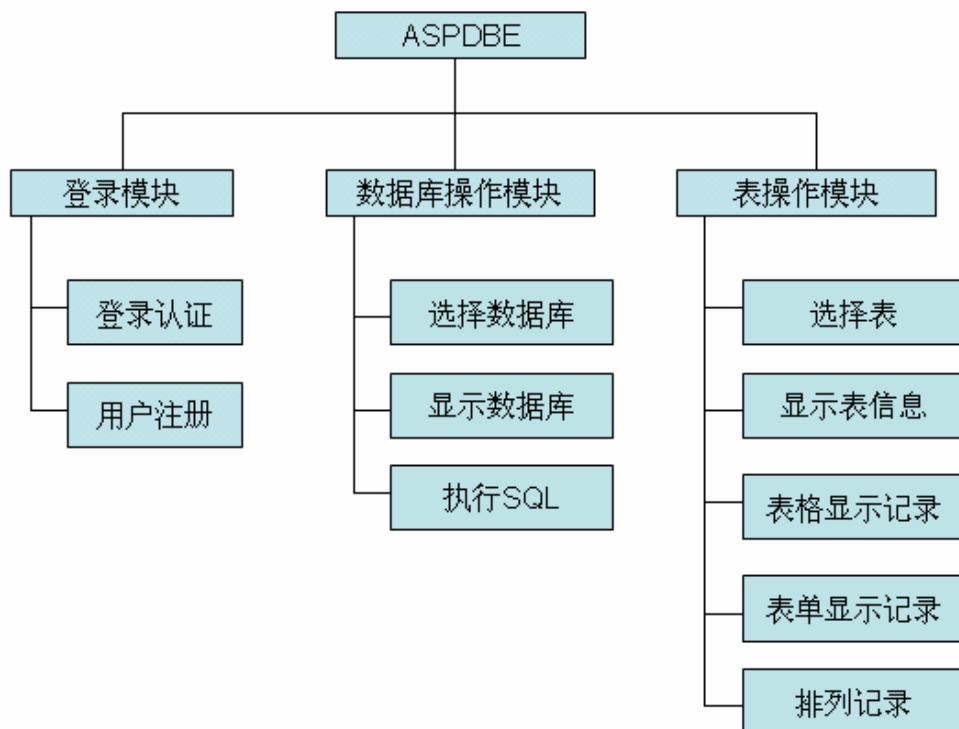


图 3-9 ASPDBE 模块划分

3. 表操作模块

- 1) 选择表：每个数据库中都有不同的表，用户可以点击主目录中显示的表来选择需要浏览的对象
- 2) 显示表信息：ASPDBE 通过表格的形式显示表信息。
 - 字段名
 - 字段类型
 - 是否空值
 - 是否主键
 - 默认值
- 3) 显示记录：显示相应表中记录，一般用户主要查询这些内容。根据用户的不同需求，ASPDBE 使用表格式显示和表单式显示。
- 4) 排列记录：对显示的记录，按照用户要求的顺序排序并显示出来。

3.3.4 系统数据库设计

在 ASPDBE 的设计中，只需要在数据库中建立两个表，一个是用户登录信息表，另一个是用户资料表。因为 ASPDBE 访问的数据没有特定对象，只

有当管理员进行发布时才添加新的表格，而这两个表是用来验证用户登录和记录用户信息的，所以 ASPDBE 中只初始化这两个表。

1. 用户登录信息表

这个表存储用户登录信息，用来验证用户登录。具体定义如表 3-1 所示。

ID	字段名	字段类型	长度	是否关键字	是否空值
1	Uname	Char	30	是	否
2	Pwd	Char	30		否
3	Rank	Int	1		否

表 3-1 用户登录信息表

2. 用户资料表

这个表存储用户个人资料，具体定义如表 3-2 所示。

ID	字段名	字段类型	长度	是否关键字	是否空值
1	Id	Int	自增	是	否
2	Name	Char	30		否
3	Age	Int	3		否
4	Address	Char	60		否
5	Sex	Char	5		否

表 3-2 用户资料表

第四章 ASPDBE 的实现

4.1 用 ASP 连接数据库

4.1.1 OLE-DB、ODBC 和 ADO

微软新推出了 OLE-DB 数据访问技术，来实现全球数据访问。它为数据存储驱动程序提供了一个低级接口，可以在特定的数据存储里连接和处理数据。关键一点在于，OLE-DB 驱动程序（或者是提供者）给所有使用它的客户应用提供了一个标准接口，此接口与诸如 ADO(ActiveX Data Objects,ActiveX 数据对象)这样的高级数据访问技术兼容。由于 OLE-DB 提供者提供了同样的标准接口，所以，ADO 可以相同的方式与它对话，因而可以访问 OLE-DB 提供者的任何数据存储。

ODBC（Open Database Connectivity，开放数据库连接）是微软连接到数据库的最新方法，其设计方法与 OLE-DB 相同。许多软件生产商都支持它。

ADO 提供了一组对象，我们可以将它们与数据存储一起使用，因为它适用于驱动程序。其中的三个主要的对象为：Recordset、Command 和 Connection。

Recordset 对象用于以标准的行和列的表格记录集的格式来创建和操纵数据本身。对于实际上不是数据库的数据存储（例如通常以非结构化方式存储信息的应用程序），在数据返回前，是作为记录集建立的，如 Microsoft Indexing Service。

Command 对象用于定义传递到数据存储的命令。它可以是存储过程的名称和参数，或者只是作为 SQL 字符串的查询。通常，只是在运行需要对其传递参数的存储过程的地方，或者运行需要在参数中返回值的存储过程的地方，才使用 Command 对象。

Connection 对象用于定义 ADO 和数据存储之间的连接。它可以以几种方式完成。通常，我们首先显式定义并打开一个连接，但在需要的情况下，Recordset 对象可以透明方式自动打开和使用预定义的 Connection 对象。

三者的具体关系如图 4-1 所示：

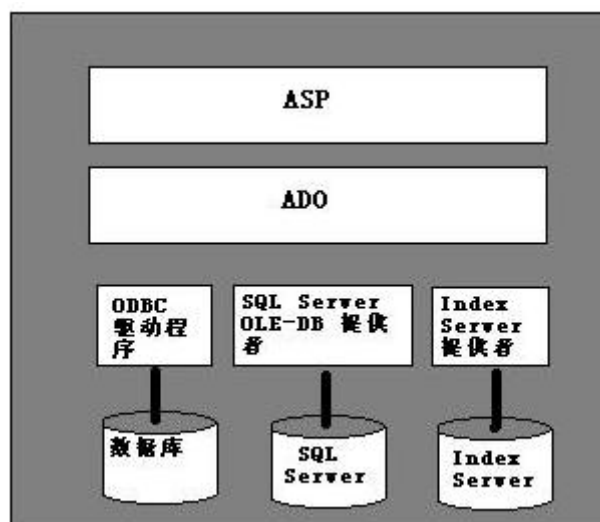


图 4-1 OLEDB、ODBC 和 ADO 三者关系

1. 利用 ODBC 驱动连接 SQL Server

在连接数据库时需要使用 ADO 中的 connection 对象。首先，创建一个新的 connection 对象 conn，再用 ODBC 驱动进行连接。

```
<%
Set conn = Server.CreateObject("ADODB.Connection")
Conn.ConnectionString="Driver={SQLServer};server=servername      ;
uid=sa ;pwd=pass;database=dbname;"
conn.open
%>
```

2. 利用 OLE-DB 驱动连接数据库

据微软表示，OLE-DB 驱动程序的效率比 ODBC 驱动程序佳，当使用 OLE-DB 连接数据库时把上面的代码中的 Conn.ConnectionString 替换成如下代码：

```
<%
Conn.ConnectionString="Provider=SQLOLEDB;
datasource=machinename;initialcatalog=dbname;
userid=sa;password=pass;"
%>
```

4.2 参数传递

在 ASPDBE 的设计中，不可能把所有内容写在一个网页中。当多个网页有

某些联系时，就会用到一些共同的参数，比如当用户登陆时，首页中提示用户输入用户名和密码，当点击提交时就会跳转到一个检测文件，这个文件对于用户是不可见的，这时首页必需把用户的信息传递给检测文件，这里就使用了参数传递这个概念。

第二章中介绍了 session 对象和 application 对象，这两个对象都是存储浏览器端变量的。通过这两种对象就可以实现参数的传递。举个例子，比如一个网页的功能是显示当前用户的个人信息，这时就要求获取用户的用户名来检索。获取用户名的方式就可以用 session 对象实现，在用户登录时，把用户名保存在 session 中`<% session(username)="username"%>`，当需要调用时只需用`<% username=session(username)%>`就把用户名传递过去了。这里等号前面的都是变量。Application 对象的使用方法类似，只是作用范围不同。

Request 对象是参数传递最常用的方法，request 对象可以获取由浏览器端返回服务器端的数据，例如浏览者所输入的表单数据、存储在浏览者本机计算机上的 cookie 等。Request 对象中 Cookies 集合用来获取 cookie 中的数据，而 Form 对象和 QueryString 对象用来获取用户输入的表单数据。

在获取表单数据时，根据 HTML 语言中的`<FORM>`表单中的`<OPTION>`标记区别使用 Form 对象和 QueryString 对象。

当 option=get 时使用

```
<%request.querystring("username")%>
```

当 option=post 时使用

```
<%request.form("username")%>
```

当不确定什么 option 的值时可以省略 querystring 和 form 直接使用

```
<%request ("username")% >
```

这样的话，Request 对象就会现在 querystring 集合中搜索 username，如找到就返回变量的值，如未找到就继续到 form 集合中搜索。

4.3 分页浏览

由于用户所查询的记录可能非常多，不能在一页中显示出来，这就要求我们使用分页浏览。分页浏览就是把所有记录按一个常数分别显示在不同页里，通常这个常数表示每页最多能显示的记录的个数，在 ASPDBE 中我们设定这个常数为 10。在分页浏览中，应该实现“上一页”和“下一页”的跳转，并且可以提供页码供用户选择需要跳转到的页。下面给出分页浏览的实现。

4.3.1 ADO 的 Recordset 对象

在分页浏览中主要遇到两个问题：第一个，如何控制分页。第二个，如何

控制分页的输出。其中，控制分页就是确定在什么地方分页，分几页。而分页输出的控制就是确定用户翻页的方式。

ASPDBe 使用 ADO 的 Recordset 对象来实现分页浏览。这种方法对开发环境的要求不高，在记事本环境下就可以实现。它实现起来也比较容易，用到的对象也很少。

该方法手工建立 Recordset 对象，利用其 `pagesize`（每页指定显示记录数），`pagecount`（总页码数）和 `absolutePage`（当前页码数）属性来控制分页的输出。首先，把记录保存在新建的名为 `rs` 的 Recordset 对象中，然后通过

<code>rs.PageSize = 10</code>	'每页记录条数'
<code>iCount = rs.RecordCount</code>	'记录总数'

语句获取记录总数并且定义每页显示的记录数。有了 `pagesize` 与 `RecordCount` 的值后，可以很容易确定总页数。这里只需要用简单的除法就可以解决。但是如果 `RecordCount` 不是 `pagesize` 的整数倍时如何确定页数和最后一页的记录数有成为问题。这个问题可以用 Recordset 对象中的 `pagecount` 解决，`pagecount` 利用 `pagesize` 和 `RecordCount` 自动生成总页数。当当前页的值等于最后一页时，利用 $x = iCount - (maxpage - 1) * iPageSize$ 公式就可以计算出最后一页的记录数。该方法控制分页时采用 `<a href=` 直接带页码参数的方法来控制翻页。也就是把每一页的页码做成超链接，当用户点击时直接跳转到相应得页，此时本页向目标页传递页码参数。例如，网页 `example_1.asp` 实现了分页，其源代码如下：

```
<% set rs=server.CreateObject("adodb.recordset")
sql="select * from xxx"
rs.open sql,conn,1,3           '读取数据库
.....

rs.PageSize = 10               '每页记录条数
iCount=rs.RecordCount         '记录总数用 iCount 表示
iPageSize=rs.PageSize
maxpage=rs.PageCount
page=request("page")
if Not IsNumeric(page) or page="" then    '设置 page 的值
    page=1
else
    page=cint(page)
end if
```

```

if page<1 then
    page=1
elseif page>maxpage then
    page=maxpage
end if
rs.AbsolutePage=Page      ‘把页码赋给 absolutePage 属性从而知当
                           前页的首条记录号

if page=maxpage then
    x=iCount-(maxpage-1)*iPageSize ‘这里处理了最后一页记录的个数
else
    x=iPageSize
end if
i=0
end if
%>

.....

<% do while not rs.eof and i< rs.pagesize ‘循环输出
    i=i+1
%>

.....

<%if page-1>0 then%>
<a href="example.asp?page=<%=page-1%>">上一页</a>
<%else%>
<font color=666666>上一页</font>
<%end if%>

<%if page+1<=maxpage then%>
<a href="example.asp?page=<%=page+1%>">下一页</a>
<%else%>
<font color=666666>下一页</font>
<%end if%>

</font>共<%=iCount %>条记录</div></TD>

```

4.3.2 其它分页方法

1. 用 Grid 控件设计分页

只需拖 DTC 中的 Recordset 控件和 Grid 控件到 ASP 网页中就可以。但是这种方法的缺点是必须用它给定的格式显示，而不能自由控制表格的显示格式。

2. DHTML

利用 DHTML 中表的数据绑定特性来控制记录的分页显示。数据记录显示在一个 HTML 表中。缺点就是你的翻页方法将被限制为一种特定的方式：只能“上页”和“下页”而不能“首页”和“末页”。由于是在客户端控制翻页，所以，这种方法是速度最快的，但遗憾的是它只能在支持 DHTML 的浏览器上使用。

4.4 排序

当用户浏览记录时，用户会要求显示的记录可以按用户想要的顺序排序，比如成绩的高低，年龄的大小，修改时间等。为了满足这个需求，ASPDBe 采用点击字段生成排序结果。具体方案是这样的，在显示的记录中，把字段名设为触发排序的开关，也就是设置一个链接，当用户点击字段名时，调用排序并最终把结果显示给用户。一般默认按降序排列，当用户再次点击时按升序排列。

这种排序一般是按一个或多个字段名排序，将 Recordset 对象进行排序的方法有两种，一种是使用 Recordset 对象中的 sort 属性。另一种是使用 SQL 语句，下面就分别介绍这两种方法，并给出实现代码。

1. SORT 属性

Sort 属性的格式如下所示：

```
Rs.Sort= "字段 1 DESC 字段 2 ASC..."
```

其中 ASC 表示升序，DESC 表示降序。排在前面的字段优先。

2. SQL 语句

使用 SQL 语句排序，其实就是用 order by 方法。格式如下所示：

```
Select * from tablename order by 字段 1 DESC , 字段 2 ASC...
```

其中 ASC 表示升序，DESC 表示降序。排在前面的字段优先。

上述方法在实际情况中还有一些问题。比如，当用户需要排序时，而且有不只一页的记录时，用户一般情况下要求把所有记录按某字段排序。这就要求在程序实现时，不能把排序语句写在输出代码中，而必须把排序代码写在输出代码之前。代码如下：

```

<%
    ‘这里连接数据库，取 recordset 对象
    .....
    Rs.sort= “字段 1 DESC 字段 2 ASC...”

    ‘这里是上一节介绍的分页现实代码
    .....
%>

```

当使用 **ORDER BY** 排序时也一样，只需在 **select** 语句中加入 **ORDER BY**。两种方法都比较简单，但是 **SQL** 的效率会高一些。

由于排序是动态的，也就是说当用户点击时才进行相应的排序。这就要求我们把字段名设置为链接，重复调用同一个文件，并且返回要调用的字段名。

4.5 验证码实现

验证码的出现是为了防止恶意的登录给服务器带来的巨大压力。比如，有人使用遍历法不断试探登录，这样对系统安全也构成危险。或者有人故意使用错误用户名密码不断登录，这样就给服务器带来很大压力。

显而易见，每次用户登录时显示的验证码应该是不相同的，也就是说验证码的生成应该是没有规律的。验证码一般是 4 位随机数，也有网站使用图片或数字与字母的混合体作为验证码。在 **ASPDBE** 中，使用 4 位随机数作为验证码，下面介绍验证码的实现。

4.5.1 相关函数

1. Rnd 函数

Rnd 函数返回一个小于 1 但大于或等于 0 的值。**Rnd[(number)]** 中

- 如果 **number** 的值是小于 0，则每次 **Rnd** 生成都使用 **number** 作为随机数种子得到的相同结果。
- 大于 0 序列中的下一个随机数。
- 等于 0 最近生成的数。
- 省略 序列中的下一个随机数

要产生指定范围的随机整数时使用以下公式：

$$\text{Int}((\text{upperbound} - \text{lowerbound} + 1) * \text{Rnd} + \text{lowerbound})$$

其中 **upperbound** 是上限，**lowerbound** 是下限。

在 **ASPDBE** 中，由于要求的验证码是四位数字，所以这个公式的上限和下限应该是数字 0 和 9 的 **ANSI** 字符。

2. Chr 函数

Chr 函数

返回与指定的 ANSI 字符代码相对应的字符。

Chr(charcode) 其中 charcode 参数是可以标识字符的数字。

4.5.2 代码实现

首先实现验证码功能，需要确定是用什么方法。从上面介绍的 Rnd（）函数的功能可以想到，应该使用一个 4 次循环，每次循环生成一位验证码，循环结束时就得到了 4 位验证码。在验证码实现过程中有个问题，如何保存生成的验证码和用户输入的验证码。如果通过表单传参，只能把用户输入的验证码传递给检测文件，而自动生成的验证码却不能，这样就无法对二者进行比较。ASPDBE 中使用 Session 对象来解决这个问题，只需把自动生成的验证码存储到 Session 中，在检测文件中取出即可。

在本系统中实现验证码时，需要在登录界面显示要输入的验证码，并且在 check.asp 里进行检验。这里 check.asp 是 ASPDBE 中检验用户名、密码以及验证码的 ASP 文件。假设登录界面为 index.asp，index.asp 中有关验证码的代码如下：

```
<%  
    ‘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，该生成器具有根据系统计时器得到的种子。  
    Randomize  
  
    Do While Len(rndnum)<4                ‘获取一个 4 位的随机数  
  
        num1=CStr(Chr((57-48)*rnd+48))    ‘随机码是上限 57，下限 47 的  
                                           随机数并且随机数作为 ANSI  
                                           码转换为字符  
  
        rndnum=rndnum&num1                ‘连接 4 位字符  
    loop  
    session("verifycode")=rndnum        ‘保存在 session 中，名字为  
    verifycode  
%>
```

```
<input name="verifycode" type="text" value="" size="14" />  
'这段代码在网页中显示验证码并提示用户输入验证码  
<%=session("verifycode")%>  
<input type="hidden" name="verifycode2"  
value="<%=session("verifycode")%>">
```

check.asp 中的代码如下所示:

```
<% dim verifycode,verifycode2  
'提取原验证码  
verifycode=trim(Request.Form("verifycode"))  
  
'提取用户输入的验证码  
verifycode2=trim(Request.Form("verifycode2"))  
  
'比较二者的值，并且判断  
if verifycode<>verifycode2 then  
  
response.write"<SCRIPT language=JavaScript>  
alert('您输入的验证码不正确。');"</SCRIPT>"  
response.write"location.href='../index.asp'"</SCRIPT>"  
founderr=true  
  
else  
'把 session 中 verifycode 置空值  
session("verifycode")=""  
%>
```

这样就实现了 4 位的数字验证码。

第五章 总结

5.1 完成工作与收获

通过比较多种动态网页开发技术，参考查阅相关的中英文资料，本文最终使用 ASP 技术设计与实现了 ASPDBE 数据库浏览工具。本文所作的主要工作和成果总结如下：

1. 深入学习了 ASP 技术。构建了 ASP 运行环境，通过对 ASP 开发的网页源代码的阅读，了解了开发中使用的许多关键技术。学习了 HTML 语言，使用网页开发工具设计并制作了系统用到的网页。

2. 提出了 B/S 三层结构的设计方案。本文通过对多种动态网页开发技术（ASP，PHP，JSP），以及目前已有的类似软件（ACCESS，PHPADMIN，SQL Server 等）进行比较，和运用软件工程的相关知识，提出了 B/S 三层结构的设计方案，并将其运用在实际系统开发中。

3. 在系统实现过程中，运用系统设计时确定的方案，实现了各个模块的功能。在实践中体会了系统开发流程，对一些关键技术也有了新的体会，开阔了眼界。

5.2 未来工作的展望

目前市场上已经有很多同类软件，在 ASPDBE 的设计与实现中，参考了其它软件的设计方案和功能。但是由于时间有限和经验不足，ASPDBE 还有许多需要完善的地方。比如，ASPDBE 是数据库浏览工具，这在功能上有局限性，在未来的工作中应该把 ASPDBE 发展成为数据库管理工具。另外还有一些功能需要改进，比如，在排序时可以按照多个字段进行排序。

参考文献

- [1] 龙马工作室. ASP+SQL Server 组建动态网站实例精讲[M]. 北京邮电出版社. 2004.
- [2] 邹建. 中文版 SQL Server2000 开发与管理应用实例[M]. 人民邮电出版社. 2005.
- [3] Alex Homer .ASP 3.0 专业 Web 技术. 中国邮电出版社[M]. 2000.
- [4] Ceri, S., Fraternali,P,et al. Architectural issues and solutions in the development of data-intensive Web applications[J]. CIDR. 2003.
- [5] Ceri, S. and Manolescu, I.Constructing and integrating data-centric Web applications: Methods, tools, and techniques (tutorial) [J]. Very Large Databases Conference. 2003. 1151.
- [6] 宋颜浩, 费文华. ASP 与相关数据库技术高级指南[J]. 水利电力出版社.
- [7] 黄斯伟. 网页样式设计 CSS 使用详解[M]. 人民邮电出版社.
- [8] 邓文渊, 陈惠贞, 陈俊荣. ASP 与网页数据库设计[J]. 中国铁道出版社. 2004.
- [9] 高守传. 精通 SQL—结构化查询语言详解[M]. 人民邮电出版社. 2007.
- [10] Turley P, Wood D . SQL Server 2005 深入开发系列—SQL Server 2005 Transact-SQL[J]. 清华大学出版社. 2007.
- [11] 吕少华. Web 数据库开发与应用实例(ASP+SQL Server) [M]. 科学出版社, 2001.
- [12] 美·索尔. 即时应用 CGI / Perl[M]. 北京: 人民邮电出版社, 2002.
- [13] 马长青, 刘中合, 刘贤喜. 基于 Web 数据库技术的系统开发[J]. 山东农业大学学报```自然科学版, 2005.
- [14] 赵春泉. 客户/服务器开发工具实用手册[R]. 清华大学出版社. 2002: 221-223
- [15] Marco Brambilla, Stefano Ceri et al. Process modeling in Web applications [J]. TOSEM, 2006, 15(4): 360-409

致 谢

经过这学期的努力，毕业设计也接近尾声。这篇论文不只凝结了我一个人的汗水，也凝结了所有帮助、支持我的老师和同学的汗水。

在这里我首先要向张坤龙老师表示最真诚的感谢。张老师学识渊博、治学严谨，无论在学习上还是生活中都是我应该学习的榜样。在整个毕设过程中，张老师严格要求，积极指导。每周的例会，张老师督促我们的进度，解答我们的问题，帮助我们拓宽思路。当我在生活中遇到困难，张老师用前辈的眼光指导我，纠正我的不足。在张老师身上除了学到知识以外，我学到了做学术应该有的精神，让我受益匪浅。再一次感谢张老师，祝他工作顺利，身体健康。

感谢同组的同学，这个学期里，大家互相帮助，互相鼓励。大家走过了四年，也走过了这段大学最后时光，祝他们毕业设计取得好的成果。

感谢所有的任课老师，从各位老师身上学到的知识使我终身受用。感谢学院王英老师，在我大学最艰苦的时期给我帮助。

谢谢！