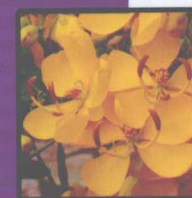
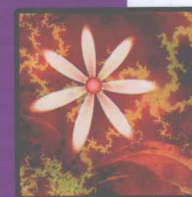


点石成金

应用开发 典型实例

黄永晔 刘 晖 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Ajax应用开发典型实例

黄永晔 刘 晖 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要通过多个Ajax技术制作的实例,向读者逐步深入地介绍使用Ajax技术建设Web 2.0网站的方法、技巧和详细步骤。全书共分为13章,由浅入深、循序渐进地介绍基于Ajax组合查询实例、调查程序实例、天气预报实例、会员注册登录验证、股票价格查询、无刷新新闻系统、滚动翻页、上传文件进度条、Ajax调用Web Service应用实例、RSS阅读器实例等的详细制作过程。

本书适合网页程序开发人员、网站建设人员、网页培训班学员、自学读者及大中专院校相关师生作为学习用书或参考教材。

黄永晔 编著

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Ajax应用开发典型实例/黄永晔,刘晖编著. —北京:电子工业出版社,2008.7

ISBN 978-7-121-06696-2

I. A… II. ①黄…②刘… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆CIP数据核字(2008)第069852号

责任编辑:吴源姜影

印刷:北京天竺颖华印刷厂

装订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

开本:787×1092 1/16 印张:14.25 字数:360千字

印次:2008年7月第1次印刷

定价:27.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：（010）88254396；（010）88258888

传 真：（010）88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路173信箱

电子工业出版社总编办公室

邮 编：100036

欢迎与我们联系

为了方便与我们联系，我们已开通了网站（www.medias.com.cn）。您可以在本网站上了解我们的新书介绍，并可通过读者留言簿直接与我们沟通，欢迎您向我们提出您的想法和建议。也可以通过电话与我们联系：

电话号码：（010）68252397。

邮件地址：webmaster@medias.com.cn

前言

本书通过多个Ajax技术制作实例，详细讲解了应用Ajax技术制作网站的技术和技巧，使读者通过具体实例的学习，直观快捷地掌握利用Ajax技术制作不同类型网站的技巧和实战经验。本书是专门针对网站设计用户开发的初中级教材，由浅入深，条例清晰，内容全面，阐述权威，技术含量高，非常适合作为自学参考书和大专院校网站设计教材。

本书主要特点

系统全面：本书通过大量实例，详细全面地介绍了使用Ajax技术建设Web 2.0网站的方法、技巧和详细步骤。

结构清晰：本书结构清晰完整，代码简洁，通过大量的实例制作说明，让读者熟悉并掌握Ajax技术。

实用性强：本书的实例均具有针对性、实用性，通过对实例制作流程的详细讲解，使读者能快速掌握各种编写工具软件的使用方法，帮助读者理解教程内容并加深印象。

本书结构安排

全书一共13章，各章节实例经过精心挑选，涉及的知识广，能让读者更全面地掌握Ajax技术的应用。

全书章节结构安排如下：

第1章：通过制作一个Ajax组合查询实例，可以直接对数据库进行增加、删除、修改、查询等操作，而且可以保持网站的持续服务。

第2章：通过对一个已有的调查程序进行改造，添加Ajax技术，让参与调查的人可以在同一个页面上提交调查问卷，并查看结果。

第3章：通过制作一个天气预报查询实例，学习使用Ajax调用某些大型网站天气频道提供的RSS数据在本地的页面上查询天气情况。

第4章：通过制作一个会员注册登录验证的实例，学习利用Ajax技术实现用户注册时的用户名检测和E-mail地址唯一性验证。

第5章：通过制作一个搜索提示功能，了解Ajax技术给用户带来的更好使用体验。使用搜索提示，可以看到一些关键字的提示，以及大概搜索结果的提示。

第6章：通过制作一个无限级菜单的效果实例，学习使用Ajax技术解析XML文件内容的多种方法。

第7章：通过制作一个无刷新图片切换特效实例，学习jQuery的基础及工作原理，并通过实例讲述了jQuery制作JavaScript程序的基本步骤，让用户体会到jQuery带来的高效率。

第8章：通过制作一个自动刷新的股票价格查询应用实例，详细讲解了利用jQuery强大的选择器，实现后台程序和前台界面的交互功能。

第9章：通过制作一个无刷新页面新闻系统实例，实现了新闻的无刷新浏览，提高了用户的阅读体验，也方便了日后的更新和维护。

第10章：通过制作一个滚动翻页的评论系统实例，介绍了动态生成JavaScript脚本的技术，同时再次利用jQuery强大的扩展功能，重新定义了文本输入框的作用。

第11章：通过制作上传文件进度条应用实例，学习使用jQuery和jQuery.MutiFile插件完成设计工作，并了解如何利用jQuery来解析XML的内容。

第12章：通过制作两个利用Ajax调用Web Service应用的实例，学习如何使用jQuery框架对Web Service的调用，并对Web Service做了详尽的介绍和分析。

第13章：介绍使用JSON实现跨域的数据读取，并利用jQuery的UI插件完成了RSS阅读器的显示。

本书内容体现了“实用”的思路，读者可以根据教程中的实例进行扩展或将代码“移植”，直接制作出适合自己需要的程序，而且，可以充分利用现有的插件或类来编写程序，这样可以尽最大可能减少程序开发的时间，提高效率。

本书配套资料

本书的配套资料中提供了书中涉及的所有实例的相关素材文件、原始文件和最终的源文件，分别放置在对应的文件夹（第xx章）中。

由于本书篇幅所限，加之时间仓促，书中难免有错漏之处，敬请广大读者批评指正。

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（<http://www.hxedu.com.cn>），在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 组合查询实例	1
1.1 实例分析	1
1.2 制作步骤	2
1.3 本章小结	21
第2章 调查程序实例	22
2.1 实例分析	23
2.2 制作步骤	25
2.3 本章小结	33
第3章 天气预报查询实例	34
3.1 实例分析	34
3.2 制作步骤	35
3.3 本章小结	46
第4章 会员注册登录验证实例	47
4.1 实例分析	47
4.2 制作步骤	48
4.3 本章小结	62
第5章 搜索提示应用实例	63
5.1 实例分析	63
5.2 制作步骤	65
5.3 本章小结	78
第6章 无限级菜单应用实例	79
6.1 实例分析	79
6.2 制作步骤	80
6.3 本章小结	94
第7章 无刷新图片切换特效实例	95
7.1 实例分析	95

7.2	什么是jQuery	96
7.3	制作步骤	101
7.4	本章小结	109
第8章 股票价格查询应用实例		
8.1	实例分析	110
8.2	制作步骤	110
8.3	本章小结	128
第9章 无刷新页面新闻系统实例		
9.1	实例分析	129
9.2	制作步骤	130
9.3	本章小结	147
第10章 滚动翻页的评论系统		
10.1	实例分析	148
10.2	制作步骤	149
10.3	本章小结	168
第11章 上传文件进度条应用实例		
11.1	实例分析	169
11.2	制作步骤	170
11.3	本章小结	189
第12章 Ajax调用Web Service应用实例		
12.1	Web Service简介	190
12.2	制作步骤	194
12.3	本章小结	204
第13章 RSS阅读器实例		
13.1	RSS阅读器分析	205
13.2	制作步骤	206
13.3	本章小结	220

第1章 组合查询实例

本章导读

本章通过制作一个Ajax组合查询实例，可以直接对数据库进行增加、删除、修改、查询等操作，而且可以保持网站的持续服务。本章的学习重点是：

- 了解XMLHttpRequest对象的应用
- 与服务器端异步交互的操作
- 构建SQL查询语句

1.1 实例分析

在网站应用中，经常使用到ASP+Access数据库，在线维护Access数据库中的数据是件非常头疼的事情，因为Access没有像SQL Server中的查询分析器（Query Analyzer）那样的工具，要对数据库进行程序外的操作，必须把数据库下载到本地来才行。如果有一个工具能在网页上直接管理Access数据库，直接对数据库进行增加、删除、修改、查询等操作，那就省去了不少工作，更重要的是，可以保持网站的持续服务。于是，一个基于Ajax的Access管理员（Access Manager）就诞生了。如图1-1所示。

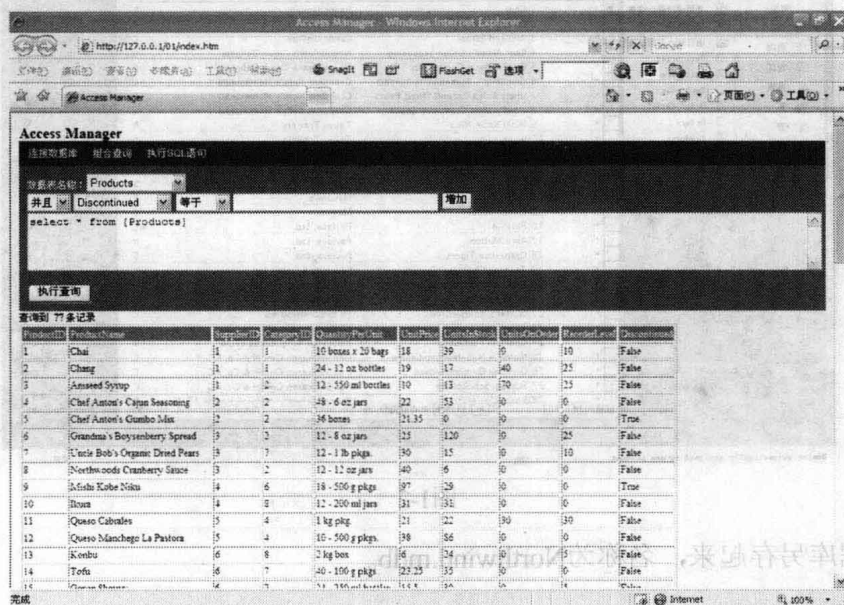


图1-1

Access Manager主要的工作：

- (1) (后台) 分析数据库结构，列出数据库中的表和字段，以XML格式返回给前台使用
- (2) (后台) 执行SQL查询，返回XML结果

- (3) (后台) 执行SQL命令, 返回处理结果
- (4) (前台) 连接指定的数据库, 获取数据库表结构
- (5) (前台) 根据获取的表结构组合SQL语句, 进行查询
- (6) (前台) 执行SQL语句, 并返回结果

本例制作完成后效果如图1-1所示。

1.2 制作步骤

1.2.1 制作数据库

为了节省读者时间, 本节实例直接选择微软Office自带的示例数据库Northwind.mdb作为操作的对象。实际上, 这个Access Manager是一个通用的Access数据库管理工具, 可以管理任何mdb数据库。不过请注意, 实例源码中未对中文做处理, 因此, 所建的数据库的表名称及字段名称不能使用中文。

1. 从配套资料的本章目录下打开Northwind.mdb数据库, 可以看到在该示例数据库中已经填写了很多资料, 因此可以直接拿来使用, 如图1-2所示。如果读者安装的是英文版的Access, 则可以自己直接创建该数据库。

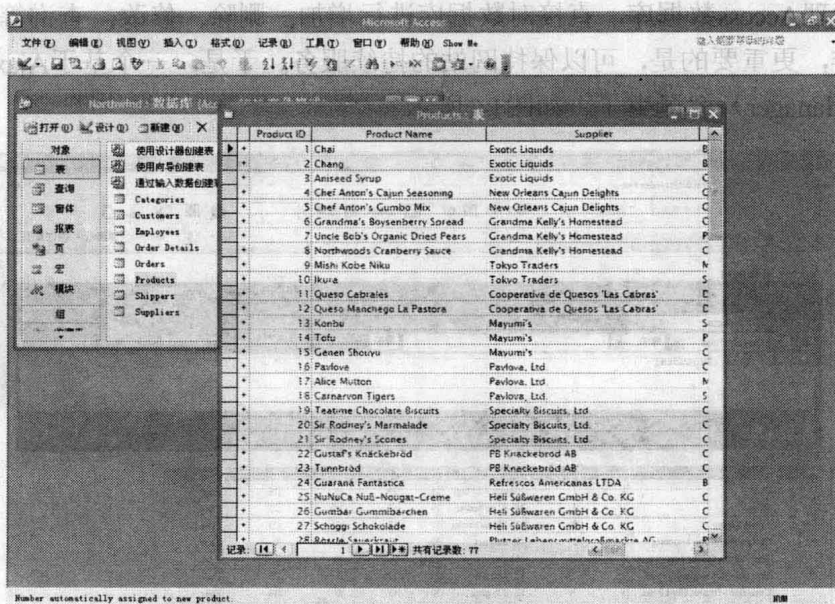


图1-2

2. 将数据库另存起来, 名称为Northwind.mdb。

1.2.2 制作查询页面

制作界面是一项很有讲究的工作, 一个好的界面可以让用户很轻松的接受你的作品。在这个应用程序中, 我们使用Dreamweaver制作了三个表格和一个标签栏, 用于表现此软件的不同功能。这三个表格用不同的<DIV>来标记, 利用标签栏的链接来控制同时只能显示一个表格, 具体步骤如下:

1. 打开Dreamweaver CS3, 新建一个空白的HTML文件, 如图1-3所示。

2. 将文件另存为index.htm, 在文档中输入Access Manager几个字, 打开“CSS样式”面板, 单击面板中的按钮, 打开“新建CSS规则”对话框, 设置选择器类型为“类”, 名称为“title”, 如图1-4所示。

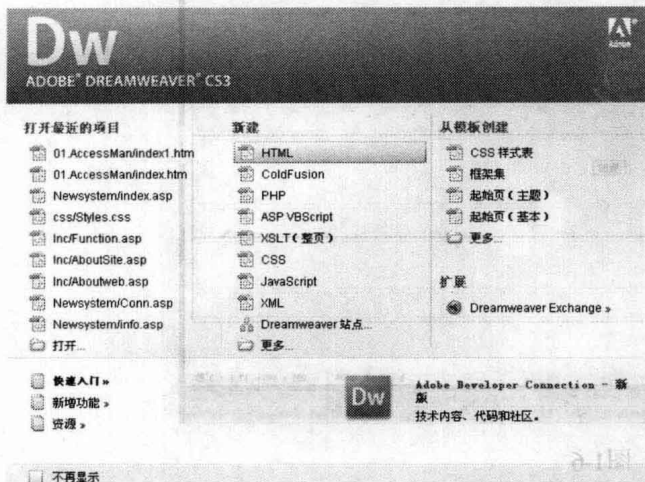


图1-3

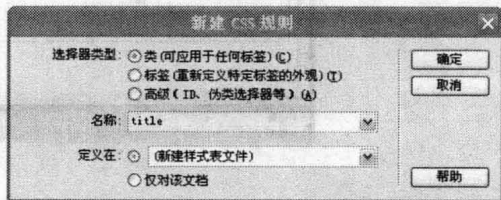


图1-4

3. 将样式表文件存储为menu.css, 定义title规则如下, 并应用于刚才输入的英文上, 完成后如图1-5所示。

```
.title {  
    font-size: 18px;  
    font-weight: bold;  
}
```

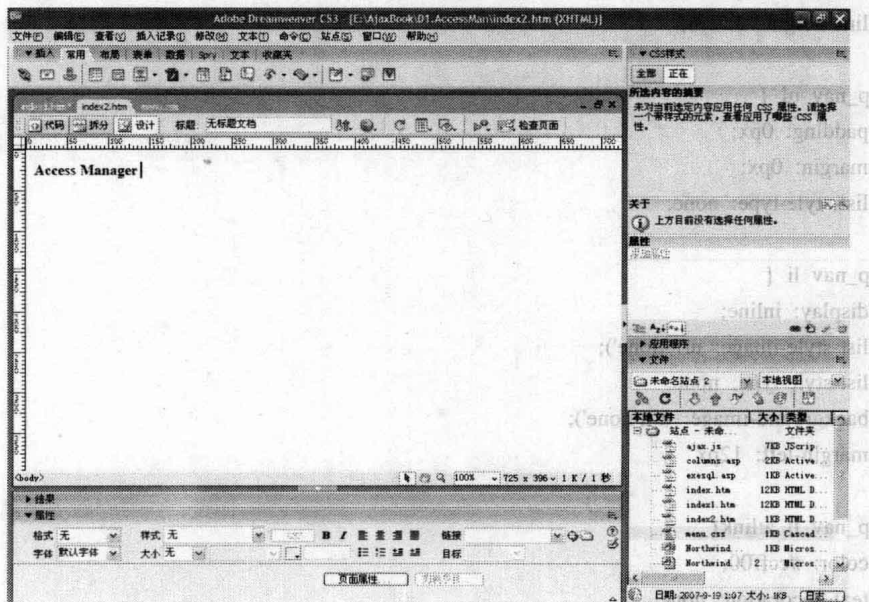


图1-5

4. 在页面中插入几个<DIV>标签, 并分别在<DIV>标签中插入一些表单元素, 如图1-6所示。

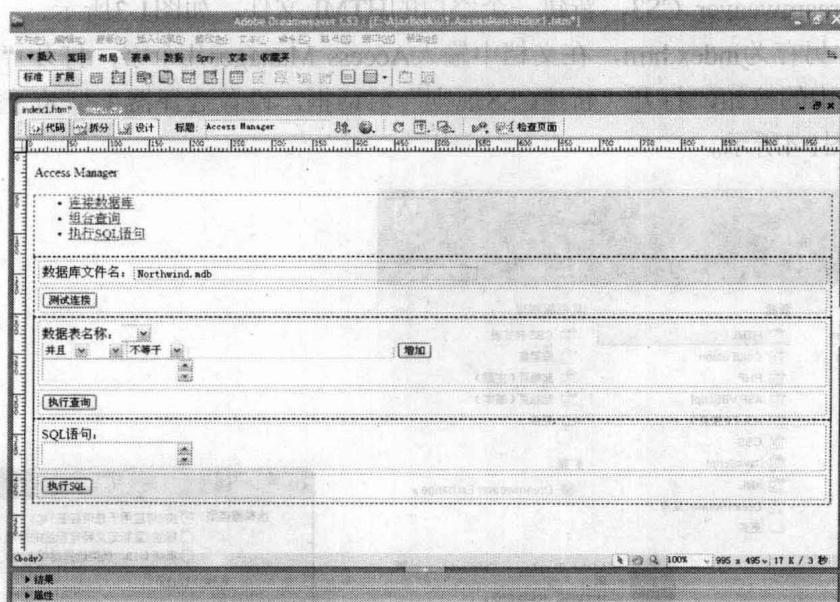


图1-6

5. 使用CSS样式控制整个页面的显示，CSS样式表文件menu.css制作完成后代码如下：

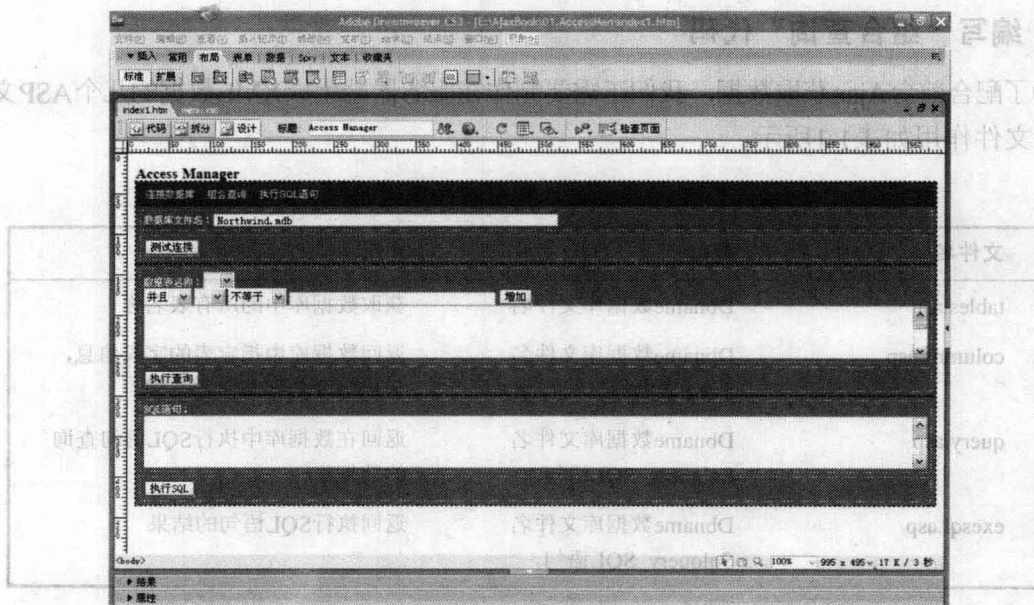
```
/* Menu 1 */
#top_nav {
    background-color: #4c4c4c;
    width: 100%;
    text-transform: uppercase;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    line-height: 2.5em;
}
#top_nav ul {
    padding: 0px;
    margin: 0px;
    list-style-type: none;
}
#top_nav li {
    display: inline;
    list-style-image: url('none');
    list-style-type: none;
    background-image: url('none');
    margin-left: 12px;
}
#top_nav li a:link {
    color: #ccff00;
    text-decoration: none;
    padding-bottom: 2px;
}
#top_nav li a:visited {
    color: #ccff00;
```

```

text-decoration: none;
}
#top_nav li a:hover {
color: #FFFF00;
border-bottom-width: 2px;
border-bottom-style: solid;
border-bottom-color: #FFFF00;
}td,div {
font-size: 12px;
}
.Main {
background-color: #5E7589;
color: #FFFFFF;
}
.title {
font-size: 18px;
font-weight: bold;
}
textarea {
width: 100%;
height: 65px;
}

```

6. 完成后的页面效果如图1-7所示, 具体的CSS规则设置内容请参考配套资料中的源文件。请注意, 所有的<DIV>均在属性检查器中设置了相应的名称。



7. 将视图转换为代码视图, 在<head>区添加如下代码:

```

<script language="javascript">
function showPanel(id){
document.getElementById('dbconn').style.display='none';

```

```
document.getElementById('query').style.display='none';
document.getElementById('sql').style.display='none';
document.getElementById(id).style.display="";
}
</script>
```

8. 在代码视图找到<body>标签，修改如下：

```
<body onLoad="showPanel('dbconnn')">
```

9. 保存文件，按【F12】快捷键，在浏览器中显示效果如图1-8所示。

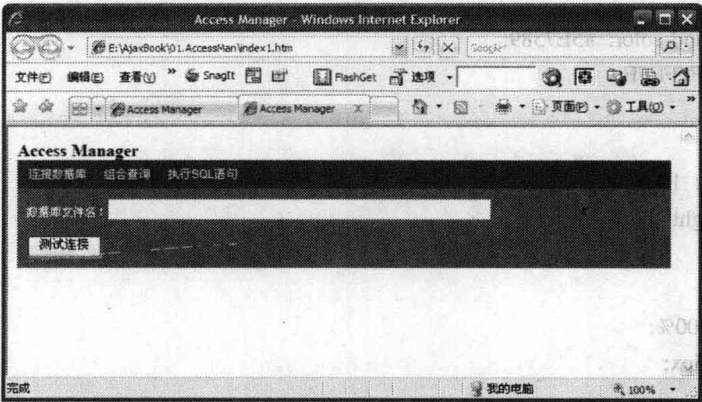


图1-8

这样查询页面就制作完成了。

1.2.3 编写“组合查询”代码

为了配合前台Ajax获取数据，我们还需要制作在服务器端生成XML数据的几个ASP文件，这几个文件作用如表1-1所示。

表1-1 ASP文件参数作用说明

文件名	参数	作用
tables.asp	Dbname数据库文件名	获取数据库中的所有表名称
columns.asp	Dbname数据库文件名 Table表名称	返回数据库中指定表的字段信息，包括字段名称和字段类型
query.asp	Dbname数据库文件名 Sqlquery SQL语句	返回在数据库中执行SQL语句查询的数据集
exesql.asp	Dbname数据库文件名 Sqlquery SQL语句	返回执行SQL语句的结果

请注意，以上每个文件都是以XML格式返回数据，以tables.asp为例，在浏览器中查看其返回的数据源码如图1-9所示。



图1-9

table.asp页面的源代码如下:

```
<%
Response.ContentType="application/xml"
Response.Expires=0
Response.Write("<?xml version='1.0' encoding='utf-8' ?>")
%>
<AccessManager>
<%
    DB_Name=request("dbname")
    if inStr(DB_Name,"/")=0 and inStr(DB_Name,"\")=0 then
        DB_Name=server.mappath(DB_Name)
    end if
    Set CONN = Server.CreateObject("ADODB.CONNECTION")
'on error resume next
    CONN.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & DB_Name
    if err.number<>0 then
        response.write "<error>Error"&err.description&"</error>"
        err.clear
    end if
    Const adSchemaTables = 20
    Const adSchemaColumns = 4
    dim rstSchema
    Set rstSchema = conn.OpenSchema(adSchemaTables)
    Do Until rstSchema.EOF
        If rstSchema("TABLE_TYPE")="TABLE" then
            response.write " <record table='""&rstSchema("Table_name")&""'"></record>"&vbCrLf
        End If
        rstSchema.MoveNext
    Loop
    set rstSchema=Nothing
    Call closeDB( )
%>
<%
SUB closeDB( )
    If IsNull(rs)=False Then
        Set rs=Nothing
```

```

End If
CONN.close( )
Set CONN=nothing
End SUB
%>
</AccessManager>

```

在这里，所有的XML都使用了UTF-8的编码。在Ajax使用UTF-8编码的好处在于不用担心中文乱码的问题。如果新建一个网站，并且打算使用Ajax技术，那最好使用UTF-8编码，以免以后遇到各种乱码或编码转换的问题。

有了这些数据，我们就可以构建自己的组合查询了。

由于Ajax的核心技术是XMLHttpRequest的应用，但在各浏览器中实现XMLHttpRequest的方法不尽相同，例如在IE中，实现XMLHttpRequest通过ActiveX对象，所以我们制作了一个能兼容IE、Mozilla Firefox、Opera等浏览器的JavaScript文件。此文件的主要功能是重定义Window.XMLHttpRequest对象，让Ajax程序在各浏览器中都可以用统一的XMLHttpRequest对象实现服务器端的交互，提高Ajax程序对浏览器的兼容性，后面的例子基本都会用到这个文件Ajax.js。此文件的主要部分如下：

```

// IE support 在IE下重定义XMLHttpRequest对象
if (window.ActiveXObject && !window.XMLHttpRequest)
{
    window.XMLHttpRequest = function( )
    {
        var msxmls = new Array('Msxml2.XMLHTTP.5.0','Msxml2.XMLHTTP.4.0','Msxml2.
XMLHTTP.3.0','Msxml2.XMLHTTP','Microsoft.XMLHTTP');
        for (var i = 0; i < msxmls.length; i++)
        {
            try
            {
                return new ActiveXObject(msxmls[i]);
            }
            catch (e)
            {
            }
        }
        return null;
    };
}

// Gecko support
/* Gecko引擎已经内置了XMLHttpRequest，直接使用就可以了 */

// Opera support
//此处省略Opera浏览器的支持代码，请到Ajax.js中查看...

```

以后只需在<head>和</head>之间加上如下一行，写程序就不用判断浏览器了。

```
<script language="javascript" type="text/javascript" src="ajax.js"></script>
```

1. 下面我们就在前台页面上写连接数据库获取表架构的代码。打开index.htm页面，将视图转换为代码视图，在<head>区的<script language="javascript">...</script>语句中添加以下代码：

```
var xmlhttp; //定义一个全局的变量
function testConn( ){
    var dbname=document.getElementById('dbname').value;
    var url = 'tables.asp?dbname='+dbname;
    xmlhttp=new XMLHttpRequest( );
    try{
        xmlhttp.onreadystatechange = showTables; //当http状态改变，就执行showTables函数
        xmlhttp.open("GET", url, true);
        //使用同步模式发送请求，如果是false，则为异步模式，即浏览器直到返回数据才会响应
        xmlhttp.send(null);
    }catch(e){
        alert('XML error:\n\n'+e.message);
    }
}

function showTables( ){ //此函数将返回的数据中的表名称添加到一个下拉框中
    if(xmlhttp.readyState == 4){ //由于我们使用的是同步模式，只有readyState为4才表示数据返回完毕
        if (xmlhttp.status == 200 || xmlhttp.status == 0){
            //只有HTTP状态返回200代码的时候才是正常的数据库
            // 取得XML的DOM对象
            var xmlDoc = xmlhttp.responseXML;
            // 取得XML文档的根
            try{
                var root = xmlDoc.documentElement;
                var records=root.getElementsByTagName('record');
                document.getElementById("tableList").options.length =0; //先将下拉列表清空
                document.getElementById("tableList").options.add(new Option("--请选择数据表-
                ", ""));
                //再添加一个提示选项
                for(var i=0;i<records.length;i++){
                    addItem(document.getElementById("tableList"),records[i].getAttribute
                    ('table'));
                }
                //逐项加入XML中读取的数据到第二个标签的“数据表名称”下拉框中
            }
            showPanel('query'); //切换到第二个标签
        }catch(e){//错误处理，提示错误信息
            alert('XML error:\n\n'+e.message);
        }
    }
}
```

2. 然后在第一个<DIV>标签的【测试连接】按钮上增加一个onclick事件，在单击此按钮的时候，就开始取tables.asp生成的XML数据，并将其中的表名称加到第二个标签的下拉列表tableList中。其修改后的代码如下：