

第 13 章 网店购物车

一个网站开展了电子商务，把琳琅满目的商品呈现在用户的面前。这也就实现了电子商务在线交易的前提。怎样才能为用户提供友好完整的购物流程呢？至关重要的一步就是让用户放置选购商品的购物车功能。利用 Dreamweaver 为网站设计编写一套网店购物车系统需要做哪些安排？需要做哪些操作？其实并不复杂，简单的电子商务之旅从这开始。

13.1 购物信息库的设计

一个简单的网店购物车系统必须具备一套完善的购物信息库。在这个库中通常会包括两个表：商品信息表（tb_goods）、购物信息表（tb_shopping）。下面就具体来介绍整个购物信息库（Access 数据库）的设计流程。

（1）单击【开始】|【程序】|【Microsoft Office】|【Microsoft Office Access 2003】命令，打开【Microsoft Access】窗口。

（2）单击【新建文件】|【空数据库】命令，弹出【文件新建数据库】对话框。可以根据实际情况，保存到相应位置。并在【文件名】文本框中输入数据库文件名，例如“shopping.mdb”，如图 13.1 所示。

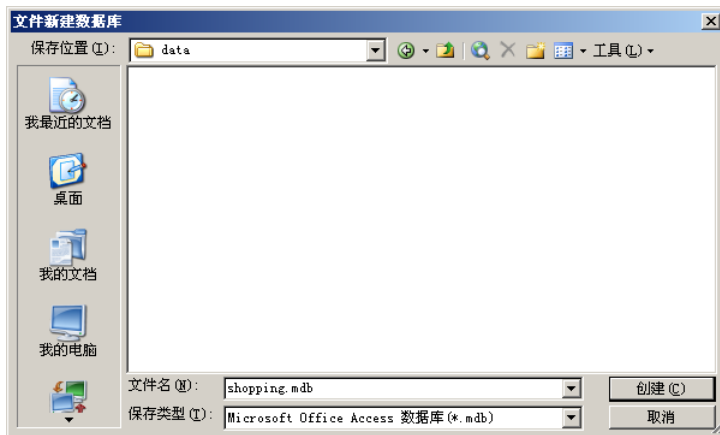


图 13.1 【文件新建数据库】对话框

（3）单击【创建】按钮，弹出【shopping:数据库（Access 2000 文件格式）】窗口，如图 13.2 所示。



图 13.2 【shopping:数据库 (Access 2000 文件格式)】窗口

(4) 单击【使用设计器创建表】按钮，弹出【表 1: 表】设计窗口。单击【文件】|【保存】命令，把表进行重新命名，例如“tb_goods”。

(5) 在【tb_goods: 表】设计窗口中按照表 13.1 所示定义所有字段信息。

表 13.1 【tb_goods: 表】所有字段信息

字段名称	数据类型	说明
id	自动编号	商品编号
Title	文本	商品名称
Price	数字	交易价格
saler	文本	卖家称呼
img	文本	图片说明

(6) 右击“id”字段，选择【主键】命令，为表设置主键，如图 13.3 所示。

(7) 单击窗口右上角的【关闭】按钮，系统会提示保存所做的修改。返回到【shopping:数据库 (Access 2000 文件格式)】窗口，会发现所设计的表“tb_goods”已经存在于当中，如图 13.4 所示。

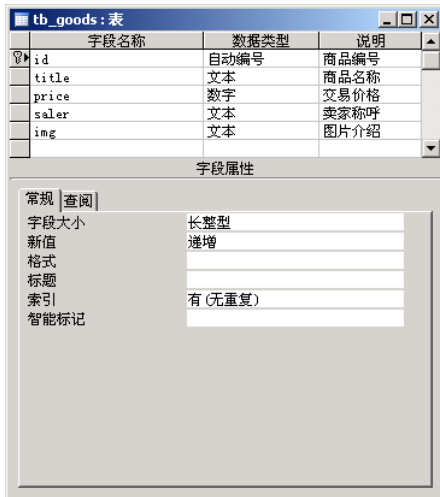


图 13.3 为表设置主键



图 13.4 【shopping:数据库 (Access 2000 文件格式)】窗口

(8) 在实际操作中，可以双击打开“tb_goods”表记录管理窗口。在这个窗口中，可以选择录入、编辑或者删除一些记录信息，如图 13.5 所示。



图 13.5 “tb_goods”表记录管理窗口

(9) 单击【使用设计器创建表】按钮，弹出【表 1: 表】设计窗口。单击【文件】|【保存】命令，把表进行重新命名，例如“tb_shopping”。

(10) 在【tb_shopping: 表】设计窗口中按照表 13.2 所示定义所有字段信息。

表 13.2 【tb_shopping: 表】所有字段信息

字段名称	数据类型	说明
id	自动编号	交易编号
goodid	数字	商品编号
dt	日期/时间	交易时间

(11) 选择“dt”字段，在【默认值】文本框中输入“now()”。目的是把当前时间做为记录新增时，“dt”字段的默认值，如图 13.6 所示。

(12) 右击“id”字段，选择【主键】命令，为表设置主键，如图 13.7 所示。

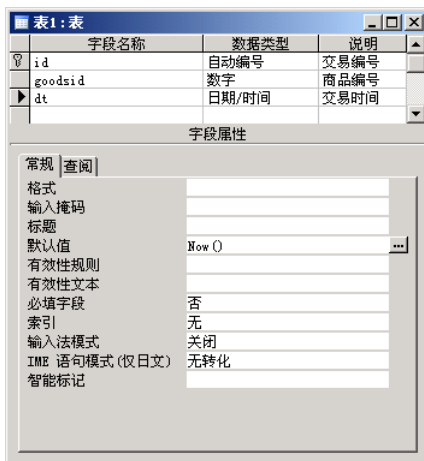


图 13.6 “dt”字段的默认值设置

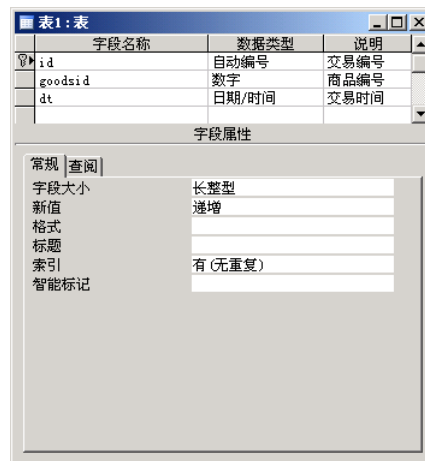


图 13.7 为表设置主键

(13) 单击窗口右上角的【关闭】按钮，系统会提示保存所做的修改。返回到【shopping: 数据库(Access 2000 文件格式)】窗口，会发现所设计的表“tb_shopping”已经存在于当中，如图 13.8 所示。

(14) 打开 Dreamweaver 的设计界面。单击【文件】|【新建】命令，新建一个标准 ASP 文件。

(15) 单击【窗口】|【数据库】命令，弹出【应用程序】面板，如图 13.9 所示。

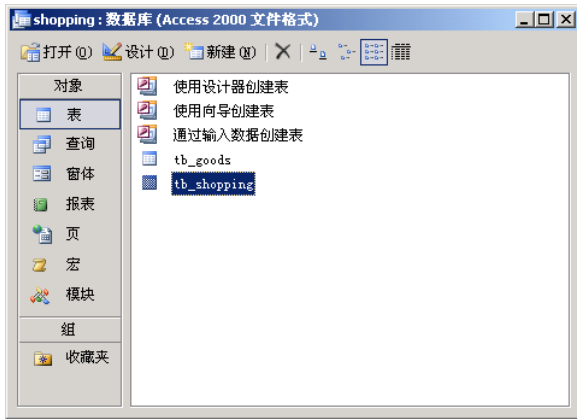


图 13.8 【shopping:数据库 (Access 2000 文件格式)】窗口

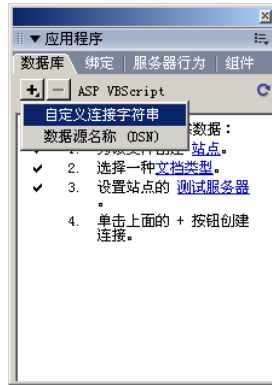


图 13.9 【应用程序】面板

(16) 单击【+】|【自定义连接字符串】命令，弹出【自定义连接字符串】对话框。在【连接名称】文本框中输入“conn”。在【连接字符串】文本框中输入“Driver={Microsoft Access Driver (*.mdb)};DBQ=i:\site\data\shopping.mdb”。选中【Dreamweaver 应连接】选项组下的【使用此计算机的驱动程序】选项，如图 13.10 所示。

(13) 单击【测试】按钮，如果连接创建成功，将弹出如图 13.11 所示的对话框。

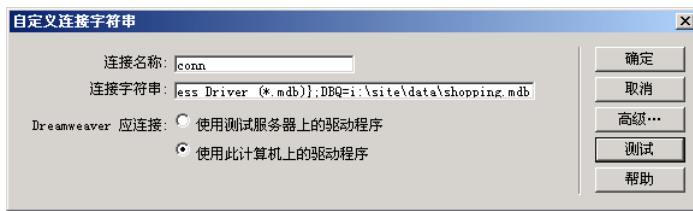


图 13.10 【自定义连接字符串】对话框



图 13.11 连接创建成功提示

(14) 单击【确定】按钮，关闭【自定义连接字符串】对话框。返回【应用程序】面板，向导将自动在其下拉列表中添加一条【conn】的列表选项，如图 13.12 所示。

(15) 单击【窗口】|【文件】命令，弹出【文件】面板。将看到在站点目录下会自动创建一个名为“Connections”的文件夹。在其下将会新增一个名为“conn.asp”的文件，如图 13.13 所示。

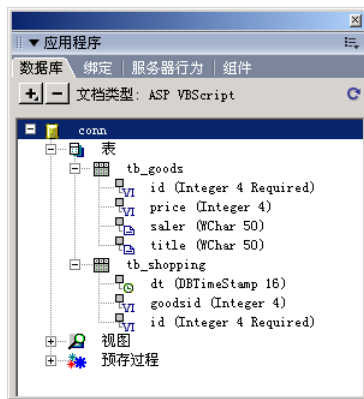


图 13.12 【应用程序】面板

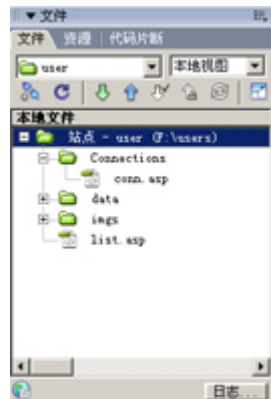


图 13.13 【文件】列表面板

13.2 商品展示的实现

商品展示功能用来显示商品的基本信息，如商品的图片、名称、定价和卖家名称。通过展示功能，帮助用户选择商品。

13.2.1 商品展示页面设计

商品展示页面读取所有商品信息，并分页显示商品信息。商品展示页面设计如下所示。

(1) 打开 Dreamweaver 的设计界面。单击【文件】|【新建】命令，新建一个名为“list.asp”的标准 ASP 文件。

(2) 单击 Dreamweaver【插入】工具栏中【常用】|【表格】命令。在弹出的【表格】对话框中进行相应设置，在【文档】窗口中插入了一些表格用于定位。

(3) 根据实际情况，可以对特定的单元格进行宽度和背景图片的指定，来修饰表格的外观。在此基础上，再对单元格行适当的排版，录入相应的文字信息具体效果如图 13.14 所示。



图 13.14 商品展示页面排版

(4) 单击【窗口】|【绑定】命令，弹出【应用程序】面板，如图 13.15 所示。

(5) 单击【+】|【记录集(查询)】命令，弹出【记录集】对话框。在【名称】文本框中输入“rs”。在【连接】下拉列表中选择【conn】选项。【表格】下拉列表中选择【tb_goods】选项，如图 13.16 所示。

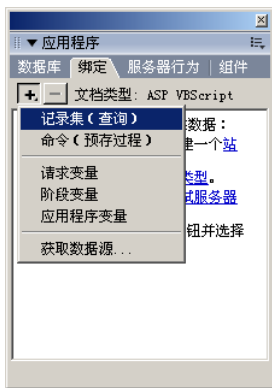


图 13.15 【绑定】命令

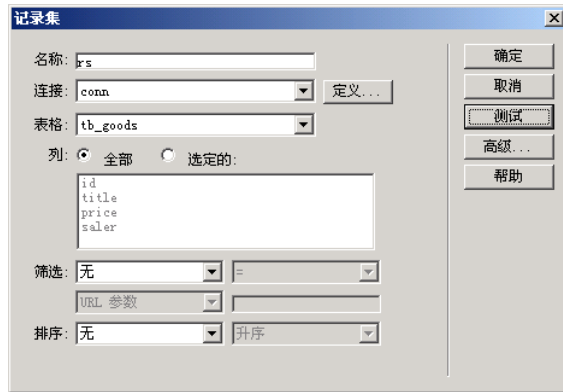


图 13.16 【记录集】对话框

(6) 单击【测试】按钮，如果记录集创建成功，将弹出如图 13.17 所示的对话框。

(7) 单击【确定】按钮，关闭【记录集】对话框。返回【应用程序】面板，向导将自动在其下拉列表中添加一条【rs】的列表选项，如图 13.18 所示。

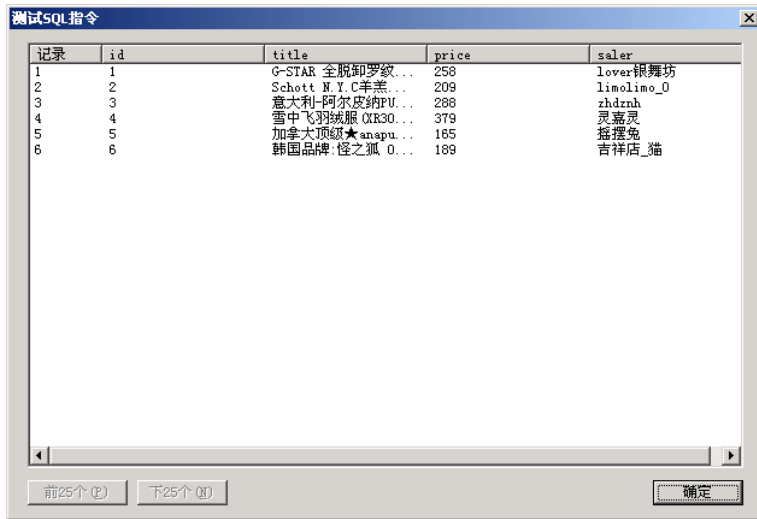


图 13.17 记录集创建测试

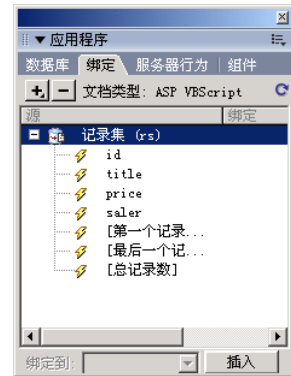


图 13.18 【应用程序】面板

(8) 在显示“商品名称”所对应的表格单元格中单击鼠标。在【应用程序】面板中选中【记录集 (rs)】|【title】选项。单击【插入】按钮，把“title”字段插入到单元格中。用上述同样的方法，在“交易价格”和“卖家称呼”所对应的单元格中插入记录集字段，如图 13.19 所示。



图 13.19 插入所有记录集字段

(9) 选中用于显示商品“图片说明”的图片对象，在【属性】检查器中【源文件】文本框中输入“<%=rs.Fields.Item("img").Value)%>”，如图 13.20 所示。

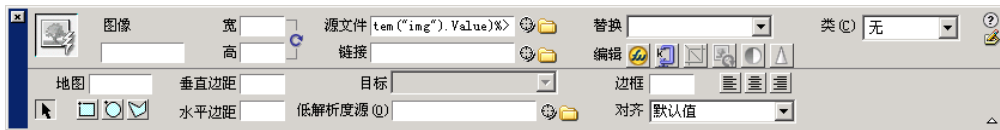


图 13.20 “图片说明”对象

(10) 选中“购买”按钮，在【属性】检查器中【链接】文本框中输入“buy.asp?id=<%=rs.Fields.Item("id").Value)%>”，如图 13.21 所示。

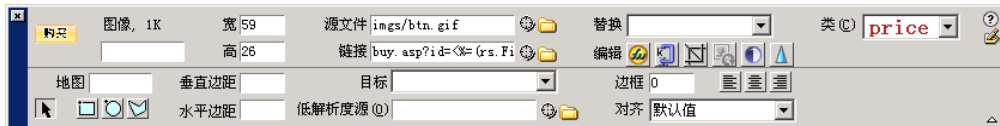


图 13.21 “购买”按钮

(11) 单击【窗口】|【服务器行为】命令，弹出【应用程序】面板，如图 13.22 所示。

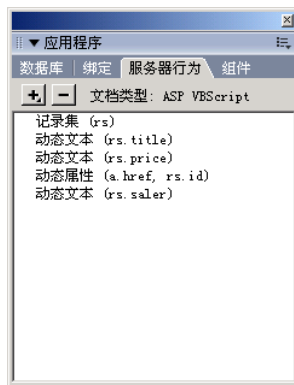


图 13.22 【应用程序】面板

(12) 直接通过鼠标或者借助【标签选择器】选中表格内用于显示数据的表格，如图 13.23 所示。

```
<body><table><tr><td><table>
```

图 13.23 【标签选择器】

(13) 单击【+】|【重复区域】命令，弹出【重复区域】对话框。在【记录集】下拉列表中选择【rs】选项。在【显示】选项组文本框中输入“5”，如图 13.24 所示。单击【确定】按钮，关闭对话框。

(14) 把鼠标定位到用于放置分页信息的区域。单击【插入】|【应用程序对象】|【显示记录计数】|【记录集导航状态】命令，弹出【Recordset Navigation States】对话框。在【Recordset】下拉列表中选择【rs】选项，如图 13.25 所示。

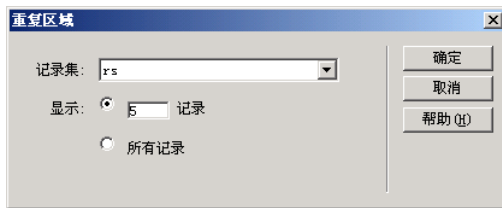


图 13.24 【重复区域】对话框

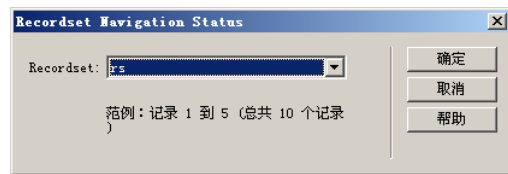


图 13.25 【Recordset Navigation States】对话框

(15) 单击【确定】按钮，关闭对话框。Dreamweave 将在文档窗口中自动插入一排用于记录统计的信息，如图 13.26 所示。



图 13.26 插入记录集导航状态

(16) 把鼠标定位到用于放置分页控制的按钮区域。单击【插入】|【应用程序对象】|【记录集分页】|【记录集导航条】命令，弹出【记录集导航条】对话框。在【记录集】下拉列表中选择【rs】选项。在【显示方式】选项组中选中【文本】选项，如图 13.27 所示。

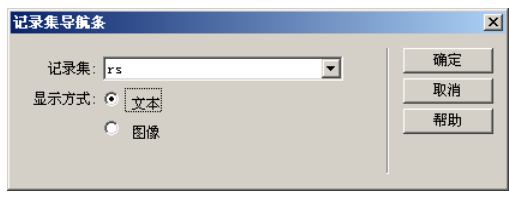


图 13.27 【记录集导航条】对话框

(17) 单击【确定】按钮，关闭对话框。Dreamweave 将在文档窗口中自动插入一排用于翻页控制的导航条，如图 13.28 所示。

启动 Internet Explorer，在地址栏中输入“http://localhost/list.asp”。将会看到如图 13.29 所示的效果。



图 13.28 插入记录集导航条



图 13.29 查看“http://localhost/list.asp”

13.2.2 生成代码分析——商品分页列表页面 (list.asp)

商品分页列表页面 (list.asp)，完整代码如下：

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="Connections/conn.asp" -->
<%
Dim rs
Dim rs_numRows
Set rs = Server.CreateObject("ADODB.Recordset")
rs.ActiveConnection = MM_conn_STRING
rs.Source = "SELECT * FROM tb_goods"
rs.CursorType = 0
rs.CursorLocation = 2
rs.LockType = 1
rs.Open()
rs_numRows = 0

```

'定义记录集变量
'定义记录当前记录位置的变量
'建立记录集对象
'指定数据库链接字符串
'获取记录

```

%>
<%
Dim Repeat1__numRows           '定义显示记录的总数目变量
Dim Repeat1__index            '定义显示记录的数目变量
Repeat1__numRows = 5
Repeat1__index = 0
rs_numRows = rs_numRows + Repeat1__numRows
%>
<%
' *** Recordset Stats, Move To Record, and Go To Record: declare stats variables
Dim rs_total                   '定义记录总数变量
Dim rs_first                   '定义初始行变量
Dim rs_last                    '定义结束行变量
' set the record count
rs_total = rs.RecordCount      '获取记录集总数
' set the number of rows displayed on this page
If (rs_numRows < 0) Then
    rs_numRows = rs_total
Elseif (rs_numRows = 0) Then
    rs_numRows = 1
End If
' set the first and last displayed record
rs_first = 1                   '设置开始行为 1
rs_last = rs_first + rs_numRows - 1 '计算结束行
' if we have the correct record count, check the other stats
If (rs_total <> -1) Then       '如果记录集总数正确
    If (rs_first > rs_total) Then
        rs_first = rs_total
    End If
    If (rs_last > rs_total) Then '如果结束行数大约记录总数
        rs_last = rs_total
    End If
    If (rs_numRows > rs_total) Then '如果当前记录行数大于记录总数
        rs_numRows = rs_total
    End If
End If
%>
<%
' *** Recordset Stats: if we don't know the record count, manually count them
If (rs_total = -1) Then       '如果获取的记录总数为-1
    ' count the total records by iterating through the recordset
    rs_total=0                '设置记录总数为 0
    While (Not rs.EOF)        '依次循环访问每个记录
        rs_total = rs_total + 1 '记录总数加 1
        rs.MoveNext           '访问下一个记录
    Wend
    ' reset the cursor to the beginning
    If (rs.CursorType > 0) Then
        rs.MoveFirst          '访问第一个记录
    Else

```

```

rs.Requery '重新查询
End If
' set the number of rows displayed on this page
If (rs_numRows < 0 Or rs_numRows > rs_total) Then
    rs_numRows = rs_total
End If
' set the first and last displayed record
rs_first = 1 '设置开始行为 1
rs_last = rs_first + rs_numRows - 1

If (rs_first > rs_total) Then
    rs_first = rs_total
End If
If (rs_last > rs_total) Then '如果结束行数大约记录总数
    rs_last = rs_total
End If
End If
%>
<%
Dim MM_paramName
%>
<%

```

以下代码实现跳转到指定记录分页。

```

Dim MM_rs
Dim MM_rsCount
Dim MM_size
Dim MM_uniqueCol
Dim MM_offset
Dim MM_atTotal
Dim MM_paramsIsDefined
Dim MM_param
Dim MM_index
Set MM_rs = rs
MM_rsCount = rs_total
MM_size = rs_numRows
MM_uniqueCol = ""
MM_paramName = ""
MM_offset = 0
MM_atTotal = false
MM_paramsIsDefined = false
If (MM_paramName <> "") Then
    MM_paramsIsDefined = (Request.QueryString(MM_paramName) <> "")
End If
%>
<%
' *** Move To Record: handle 'index' or 'offset' parameter
if (Not MM_paramsIsDefined And MM_rsCount <> 0) then
    ' use index parameter if defined, otherwise use offset parameter
    MM_param = Request.QueryString("index")
    If (MM_param = "") Then

```

```

MM_param = Request.QueryString("offset")
End If
If (MM_param <> "") Then
    MM_offset = Int(MM_param)
End If
' if we have a record count, check if we are past the end of the recordset
If (MM_rsCount <> -1) Then
    If (MM_offset >= MM_rsCount Or MM_offset = -1) Then ' past end or move last
        If ((MM_rsCount Mod MM_size) > 0) Then ' last page not a full repeat region
            MM_offset = MM_rsCount - (MM_rsCount Mod MM_size)
        Else
            MM_offset = MM_rsCount - MM_size
        End If
    End If
End If
' move the cursor to the selected record
MM_index = 0
While ((Not MM_rs.EOF) And (MM_index < MM_offset Or MM_offset = -1))
    MM_rs.MoveNext '访问下一个记录
    MM_index = MM_index + 1
Wend
If (MM_rs.EOF) Then
    MM_offset = MM_index ' set MM_offset to the last possible record
End If
End If
%>
<%
' *** Move To Record: if we dont know the record count, check the display range
If (MM_rsCount = -1) Then
    ' walk to the end of the display range for this page
    MM_index = MM_offset
    While (Not MM_rs.EOF And (MM_size < 0 Or MM_index < MM_offset + MM_size))
        MM_rs.MoveNext '访问下一个记录
        MM_index = MM_index + 1
    Wend
    ' if we walked off the end of the recordset, set MM_rsCount and MM_size
    If (MM_rs.EOF) Then
        MM_rsCount = MM_index
        If (MM_size < 0 Or MM_size > MM_rsCount) Then
            MM_size = MM_rsCount
        End If
    End If
End If
' if we walked off the end, set the offset based on page size
If (MM_rs.EOF And Not MM_paramsIsDefined) Then
    If (MM_offset > MM_rsCount - MM_size Or MM_offset = -1) Then
        If ((MM_rsCount Mod MM_size) > 0) Then
            MM_offset = MM_rsCount - (MM_rsCount Mod MM_size)
        Else
            MM_offset = MM_rsCount - MM_size
        End If
    End If

```

```
End If
End If
' reset the cursor to the beginning
If (MM_rs.CursorType > 0) Then
    MM_rs.MoveFirst '访问第一个记录
Else
    MM_rs.Requery '重新查询
End If
' move the cursor to the selected record
MM_index = 0
While (Not MM_rs.EOF And MM_index < MM_offset)
    MM_rs.MoveNext '访问下一个记录
    MM_index = MM_index + 1
Wend
End If
%>
<%
' *** Move To Record: update recordset stats
' set the first and last displayed record
rs_first = MM_offset + 1
rs_last = MM_offset + MM_size
If (MM_rsCount <> -1) Then
    If (rs_first > MM_rsCount) Then
        rs_first = MM_rsCount
    End If
    If (rs_last > MM_rsCount) Then
        rs_last = MM_rsCount
    End If
End If
' set the boolean used by hide region to check if we are on the last record
MM_atTotal = (MM_rsCount <> -1 And MM_offset + MM_size >= MM_rsCount)
%>
<%
' *** Go To Record and Move To Record: create strings for maintaining URL and Form parameters
Dim MM_keepNone
Dim MM_keepURL
Dim MM_keepForm
Dim MM_keepBoth
Dim MM_removeList
Dim MM_item
Dim MM_nextItem
' create the list of parameters which should not be maintained
MM_removeList = "&index="
If (MM_paramName <> "") Then
    MM_removeList = MM_removeList & "&" & MM_paramName & "="
End If
MM_keepURL=""
MM_keepForm=""
MM_keepBoth=""
MM_keepNone=""
```

```

' add the URL parameters to the MM_keepURL string
For Each MM_item In Request.QueryString
    MM_nextItem = "&" & MM_item & "="
    If (InStr(1,MM_removeList,MM_nextItem,1) = 0) Then
        MM_keepURL = MM_keepURL & MM_nextItem & Server.URLEncode(Request.QueryString(MM_item))
    End If
Next
' add the Form variables to the MM_keepForm string
For Each MM_item In Request.Form
    MM_nextItem = "&" & MM_item & "="
    If (InStr(1,MM_removeList,MM_nextItem,1) = 0) Then
        MM_keepForm = MM_keepForm & MM_nextItem & Server.URLEncode(Request.Form(MM_item))
    End If
Next
' create the Form + URL string and remove the initial '&' from each of the strings
MM_keepBoth = MM_keepURL & MM_keepForm
If (MM_keepBoth <> "") Then
    MM_keepBoth = Right(MM_keepBoth, Len(MM_keepBoth) - 1)
End If
If (MM_keepURL <> "") Then
    MM_keepURL = Right(MM_keepURL, Len(MM_keepURL) - 1)
End If
If (MM_keepForm <> "") Then
    MM_keepForm = Right(MM_keepForm, Len(MM_keepForm) - 1)
End If
' a utility function used for adding additional parameters to these strings
Function MM_joinChar(firstItem)
    If (firstItem <> "") Then
        MM_joinChar = "&"
    Else
        MM_joinChar = ""
    End If
End Function
End Function
%>
<%

```

以下代码设置分页导航信息。

```

Dim MM_keepMove
Dim MM_moveParam
Dim MM_moveFirst           '定义第一页链接字符串变量
Dim MM_moveLast           '定义最后一页链接字符串变量
Dim MM_moveNext           '定义下一页链接字符串变量
Dim MM_movePrev           '定义上一页链接字符串变量
Dim MM_urlStr
Dim MM_paramList
Dim MM_paramIndex
Dim MM_nextParam
MM_keepMove = MM_keepBoth
MM_moveParam = "index"
' if the page has a repeated region, remove 'offset' from the maintained parameters
If (MM_size > 1) Then

```

```

MM_moveParam = "offset"
If (MM_keepMove <> "") Then
    MM_paramList = Split(MM_keepMove, "&")
    MM_keepMove = ""
    For MM_paramIndex = 0 To UBound(MM_paramList)
        MM_nextParam = Left(MM_paramList(MM_paramIndex), InStr(MM_paramList(MM_paramIndex), "=") - 1)
        If (StrComp(MM_nextParam, MM_moveParam, 1) <> 0) Then
            MM_keepMove = MM_keepMove & "&" & MM_paramList(MM_paramIndex)
        End If
    Next
    If (MM_keepMove <> "") Then
        MM_keepMove = Right(MM_keepMove, Len(MM_keepMove) - 1)
    End If
End If
End If
' set the strings for the move to links
If (MM_keepMove <> "") Then
    MM_keepMove = Server.URLEncode(MM_keepMove) & "&"
End If
MM_urlStr = Request.ServerVariables("URL") & "?" & MM_keepMove & MM_moveParam & "="
MM_moveFirst = MM_urlStr & "0"
MM_moveLast = MM_urlStr & "-1"
MM_moveNext = MM_urlStr & CStr(MM_offset + MM_size)
If (MM_offset - MM_size < 0) Then
    MM_movePrev = MM_urlStr & "0"
Else
    MM_movePrev = MM_urlStr & CStr(MM_offset - MM_size)
End If
%>

```

以下代码显示商品展示页面。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>商品展示页面</title>
<style type="text/css">
<!--
body,td,th {
    font-family: 宋体;
    font-size: 12px;
    color: #000000;
}
.price{font-size:18px;color:#AC2C2B; font-weight:bold;}
.STYLE1 {color: #FFFFFF}
-->
</style></head>
<body>
<table width="500" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>

```



```
        </td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td height="31" align="center" background="imgs/titlebg.gif">&nbsp;</td>
  </tr>
</table>
</body>
</html>
```

以下代码关闭记录集对象，清空记录集对象。

```
<%
rs.Close()
Set rs = Nothing
%>
```

13.2 购物功能的实现

购物功能的实现，可以通过 2 个页面进行处理。分别是商品购买页面（buy.asp）、购物车页面（buy_ok.asp）。

13.2.1 商品购买页面（buy.asp）

商品购物页面实现商品购买功能，将指定商品的 id 添加到数据表中。商品购物页面如下所示。

（1）打开 Dreamweaver 的设计界面。单击【文件】|【新建】命令，新建一个名为“buy.asp”的标准 ASP 文件。

（2）单击 Dreamweaver【插入】工具栏中【常用】|【表格】命令。在弹出的【表格】对话框中进行相应设置，在【文档】窗口中插入了一些表格用于定位。

（3）根据实际情况，可以对特定的单元格进行宽度和背景图片的指定，来修饰表格的外观。在此基础上，再对单元格行适当的排版，录入相应的文字信息具体效果如图 13.30 所示。



图 13.30 录入成功信息提示页面排版

(4) 单击【窗口】|【服务器行为】命令，弹出【应用程序】面板，如图 13.31 所示。

(5) 单击【+】|【命令】命令，弹出【命令】对话框。在【名称】文本框中输入“rs”。在【连接】下拉列表中选择【conn】选项。在【类型】下拉列表中选择【插入】选项。在【SQL】文本框中输入“INSERT INTO tb_shopping (goodsid) VALUES (“&request.querystring("id")&”)”，如图 13.32 所示。

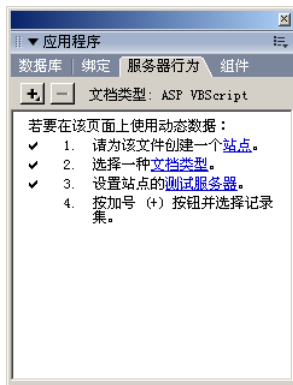


图 13.31 【应用程序】面板



图 13.32 【命令】对话框

(6) 单击【确定】按钮，关闭【命令】对话框。返回【应用程序】面板，向导将自动在其下拉列表中添加一条【命令 (rs)】的列表选项，如图 13.33 所示。

(7) 单击 Dreamweaver【插入】工具栏中【HTML】|【文件头：刷新】命令。在弹出的【刷新】对话框中【延迟】文本框中输入“2”。在【操作】选项组中选中【转到 URL:】选项，并在其对应的文本框中输入“buy_ok.asp”，如图 13.34 所示。



图 13.33 【应用程序】面板

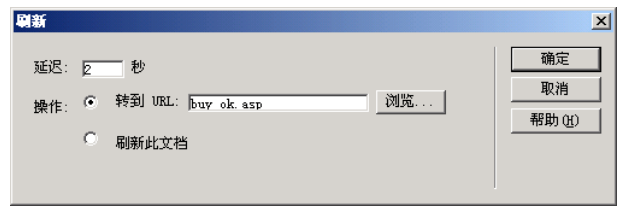


图 13.34 【刷新】对话框

(8) 单击【确定】按钮,关闭对话框。

13.2.2 购物车页面 (buy_ok.asp)

在商品购买页面中, 商品添加成功后 2 秒后, 跳转到购物车页面。购物车页面设计如下所示。

(1) 打开 Dreamweaver 的设计界面。单击【文件】|【新建】命令, 新建一个名为“buy_ok.asp”的标准 ASP 文件。

(2) 单击 Dreamweaver 【插入】工具栏中【常用】|【表格】命令。在弹出的【表格】对话框中进行相应设置, 在【文档】窗口中插入了一些表格用于定位。

(3) 根据实际情况, 可以对特定的单元格进行宽度和背景图片的指定, 来修饰表格的外观。在此基础上, 再对单元格行适当的排版, 录入相应的文字信息具体效果如图 13.35 所示。



图 13.35 购物车页面排版

(4) 单击【窗口】|【绑定】命令, 弹出【应用程序】面板, 如图 13.36 所示。

(5) 单击【+】|【记录集(查询)】命令, 弹出【记录集】对话框。在【名称】文本框中输入“rs”。在【连接】下拉列表中选择【conn】选项。在【SQL】文本框中输入“SELECT * FROM tb_goods a,tb_shopping b WHERE a. id=b. id”, 如图 13.37 所示。

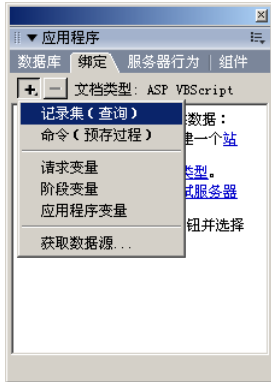


图 13.36 【绑定】命令

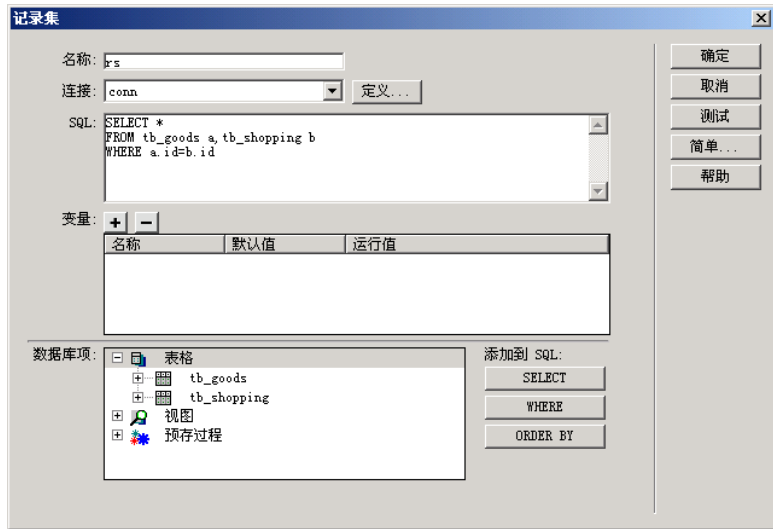


图 13.37 【记录集】对话框

(6) 单击【测试】按钮，如果记录集创建成功，将弹出如图 13.38 所示的对话框。

(7) 单击【确定】按钮，关闭【记录集】对话框。返回【应用程序】面板，向导将自动在其下拉列表中添加一条【rs】的列表选项，如图 13.39 所示。

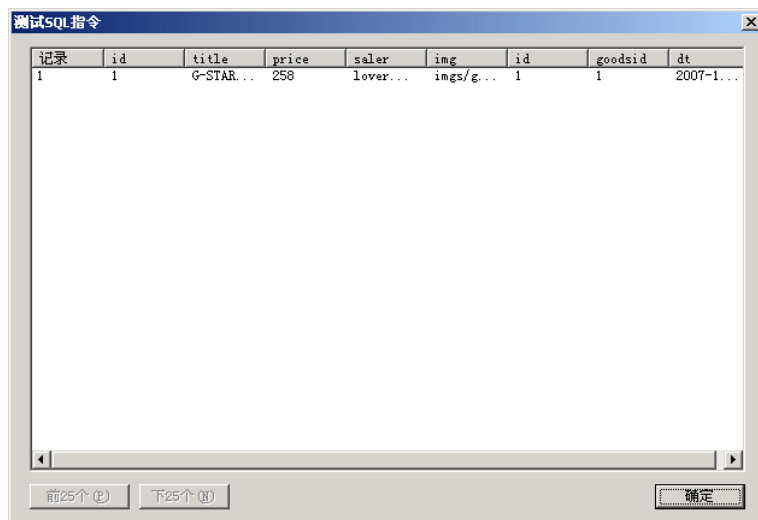


图 13.38 记录集创建测试

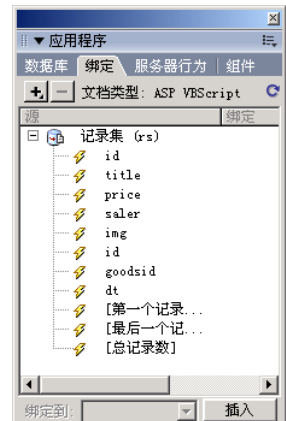


图 13.39 【应用程序】面板

(8) 在显示“商品名称”所对应的表格单元格中单击鼠标。在【应用程序】面板中选中【记录集 (rs)】|【title】选项。单击【插入】按钮，把“title”字段插入到单元格中。用上述同样的方法，在“交易价格”和“卖家称呼”所对应的单元格中插入记录集字段，如图 13.40 所示。



图 13.40 插入所有记录集字段

(9) 选中用于显示商品“图片说明”的图片对象，在【属性】检查器中【源文件】文本框中输入“`<%=rs.Fields.Item("img").Value%>`”，如图 13.41 所示。

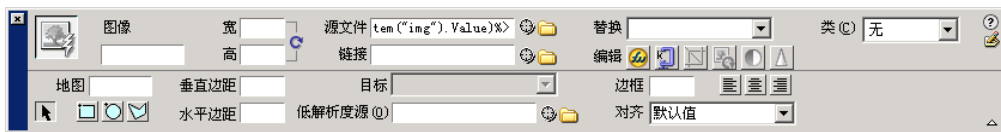


图 13.41 “图片说明”对象

(10) 按照上述同样的方法，分别选中“继续”按钮和“结算”按钮。在其对应的【属性】检查器中【链接】文本框内输入“list.asp”和“pay.asp”。

(11) 单击【窗口】|【服务器行为】命令，弹出【应用程序】面板，如图 13.42 所示。

(12) 直接通过鼠标或者借助【标签选择器】选中表格内用于显示数据的表格，如图 13.43 所示。

(13) 单击【+】|【重复区域】命令，弹出【重复区域】对话框。在【记录集】下拉列表中选择【rs】选项。在【显示】选项组中选中【所有记录】选项，如图 13.44 所示。

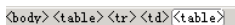
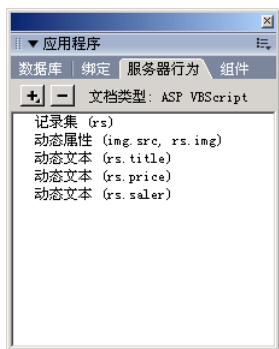


图 13.42 【应用程序】面板

图 13.43 【标签选择器】

图 13.44 【重复区域】对话框

(14) 单击【确定】按钮，关闭对话框。

13.2.3 测试购物功能

启动 Internet Explorer，在地址栏中输入“http://localhost/list.asp”。打开商品展示页面，如图 13.45 所示。单击【购买】按钮，进入 buy.asp 页面。商品信息会自动录入到数据库，如图 13.46 所示。



图 13.45 查看“http://localhost/add.asp” 图 13.46 查看“http://localhost/buy.asp”
经过 2 秒的防恶意刷新处理，系统会自动转到购物车 (buy_ok.asp) 页面，如图 13.47 所示。



图 13.47 查看“http://localhost/list.asp”


```

        <td height="50" bgcolor="#EDE8DD" style="padding:5px;">&nbsp;</td>
    </tr>
    <tr>
        <td height="31" align="center" background="imgs/titlebg.gif">&nbsp;</td>
    </tr>
</table>
</body>
</html>

```

13.2.5 生成代码分析——购物车页面（buy_ok.asp）

购物车页面（buy_ok.asp），完整代码如下：

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="Connections/conn.asp" -->
<%
Dim rs                                '定义记录集变量
Dim rs_numRows                        '定义记录当前记录位置的变量
Set rs = Server.CreateObject("ADODB.Recordset") '建立记录集对象
rs.ActiveConnection = MM_conn_STRING '指定数据库连接字符串
rs.Source = "SELECT * FROM tb_goods a,tb_shopping b WHERE a.id=b.id"
rs.CursorType = 0
rs.CursorLocation = 2
rs.LockType = 1
rs.Open()                             '获取记录
rs_numRows = 0
%>
<%
Dim Repeat1__numRows                 '定义显示记录的总数目变量
Dim Repeat1__index                   '定义显示记录的数目变量
Repeat1__numRows = -1
Repeat1__index = 0
rs_numRows = rs_numRows + Repeat1__numRows
%>

```

以下代码显示购物车页面。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>购物车页面</title>
<style type="text/css">
<!--
body,td,th {
    font-family: 宋体;
    font-size: 12px;
    color: #000000;
}
.price{font-size:18px;color:#AC2C2B; font-weight:bold;}
.STYLE1 {color: #FFFFFF}

```


13.3 结算功能的实现

13.3.1 结算功能页面设计

结算功能页面统计购物车的所有价钱，并显示总价。结算功能页面设计过程如下所示。

(1) 打开 Dreamweaver 的设计界面。单击【文件】|【新建】命令，新建一个名为“pay.asp”的标准 ASP 文件。

(2) 单击 Dreamweaver【插入】工具栏中【常用】|【表格】命令。在弹出的【表格】对话框中进行相应设置，在【文档】窗口中插入了一些表格用于定位。

(3) 根据实际情况，可以对特定的单元格进行宽度和背景图片的指定，来修饰表格的外观。在此基础上，再对单元格行适当的排版，录入相应的文字信息具体效果，如图 13.48 所示。



图 13.48 录入成功信息提示页面排版

(4) 选中“返回”按钮，在其对应的【属性】检查器中【链接】文本框内输入“list.asp”，如图 13.49 所示。

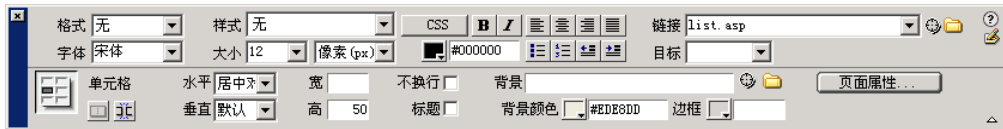


图 13.49 “返回”按钮【属性】检查器

(5) 单击【窗口】|【绑定】命令，弹出【应用程序】面板，如图 13.49 所示。

(6) 单击【+】|【记录集(查询)】命令，弹出【记录集】对话框。在【名称】文本框中输入“rs”。在【连接】下拉列表中选择【conn】选项。在【SQL】文本框中输入“SELECT SUM(a. price) as totalmoney FROM tb_goods a,tb_shopping b WHERE a. id=b. id”，如图 13.50 所示。

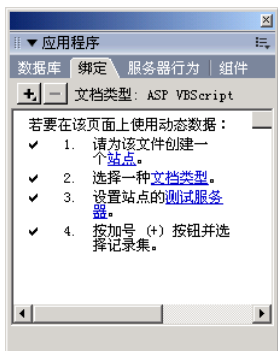


图 13.49 【应用程序】面板

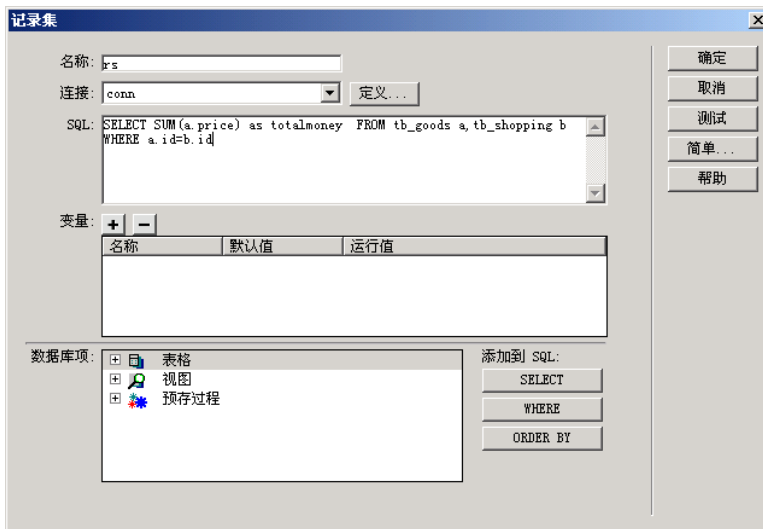


图 13.50 【命令】对话框

(7) 单击【测试】按钮，如果记录集创建成功，将弹出如图 13.51 所示的对话框。

(8) 单击【确定】按钮，关闭【记录集】对话框。返回【应用程序】面板，向导将自动在其下拉列表中添加一条【rs】的列表选项，如图 13.52 所示。

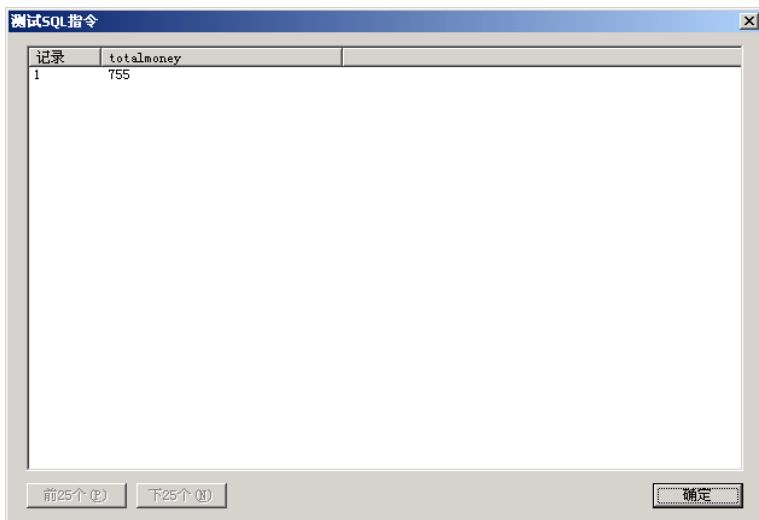


图 13.51 记录集创建测试

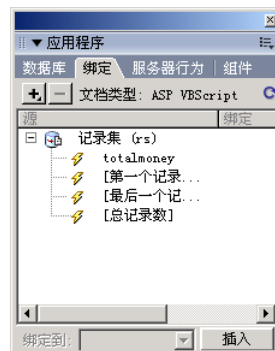


图 13.52 【应用程序】面板

(9) 在显示“结算统计”所对应的表格单元格中单击鼠标。在【应用程序】面板中选中【记录集 (rs)】|【totalmoney】选项。单击【插入】按钮，把“totalmoney”字段插入到单元格中，如图 13.53 所示。

13.3.2 测试结算功能

启动 Internet Explorer，在地址栏中输入“http://localhost/buy_ok.asp”。打开购物车页面，如图 13.54 所示。



图 13.53 插入记录集字段



图 13.54 查看“http://localhost/buy_ok.asp”

单击【结算】按钮，进入 pay.asp 页面。系统会自动对购物车内的所有商品进行结算，如图 13.55 所示。



图 13.46 查看“http://localhost/buy.asp”

13.3.3 生成代码分析——结算功能页面（pay.asp）

结算功能页面（pay.asp），完整代码如下：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="Connections/conn.asp" -->
```



```
</tr>
<tr>
  <td height="31" align="center" background="imgs/titlebg.gif">&nbsp;</td>
</tr>
</table>
</body>
</html>
```

以下代码关闭记录集对象，并清空记录集对象。

```
<%
rs.Close()
Set rs = Nothing
%>
```