

基于 SQL Server XML 技术的 数据驱动 Web 设计

彭江平 (湖南大学会计学院信息管理系 410079)

摘要: SQL Server 2000 通过提供对 XML 的支持, 成为一个 XML 使能(XML-enabled)的数据库服务器。本文结合具体的实例说明 SQL Server XML 实现及其在开发数据驱动 Web 设计中的应用。

关键词: SQL Server 2000 XML XSL Web

1 现行数据驱动 Web 设计方法

最初的 Web 是由纯 HTML 文本组成, 主要实现信息的表现而忽略了信息内语义及内部结构的描述, 不能适应日益增多的信息检索与信息存档要求, 而且 HTML 的表现能力也相对有限, 不能描述矢量图形、科学符号和一些其他特殊显示效果。为了弥补 HTML 的不足, 各大软件开发商纷纷通过各种技术对 HTML 进行扩充: 通过在 Web 中插入 JavaScript 或 VBScript 脚本, 增加 Web 网面的动态功能; 通过在 Web 服务器中使用 IDC、CGI 或 ISAPI 程序, 或者在 Web 服务器中应用 ASP、PHP 与 JSP 技术, 扩充 Web 服务器的功能以实现数据访问与网页动态生成, 其中 ASP、PHP 与 JSP 技术, 是目前大多数商业网站使用的数据驱动的动态网页生成技术。

虽然可以通过多种方式从 SQL Server 2000 中返回 XML 形式的数据, 本文仅讨论基于 HTTP 协议的返回 XML 数据的方法。这类主要是借助随 SQL Server 2000 提供的 XML 支持工具“sqlisapi.dll”, 它作为一个运行于 IIS Web 服务器上的 ISAPI 筛选器应用程序, 将一般的请求由 Web 服务器自行处理, 而将与数据库关联的处理通过筛选器中的 OLE DB 重定向到 SQL Server 数据库服务器, 其工作原理如图 1 所示。

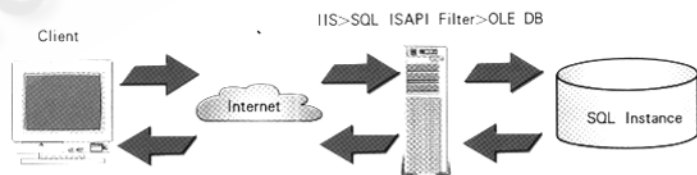


图 1 SQL Server 2000 的 XML 工作原理图

2 SQL Server 2000 的 XML 技术

在 Windows 平台下, 为了实现对数据的访问, 最直观的方法是通过数据访问对象 ADO 实现, 为了得到 XML 形式的数据一般还需要程序员设计相应的转换程序, 虽然自 ADO2.5 以后, 提供了生成 XML 数据的功能, 但因为需要通过中间层的转换过程, 效率受到一定的限制, 而利用 SQL Server 2000 的 XML 技术, 能够直接从数据库服务器中返回 XML 文档, 这一功能将为创建分布式的、数据为中心的应用系统提供有效的解决方案。

3 基于 SQL Server 2000 XML 技术的数据驱动 Web 设计实例

SQL Server 2000 通过提供对 XML 的支持, 成为一个 XML 使能(XML-enabled)的数据库服务器。通过设置 IIS 及 SQL Server 的虚拟域, 使得能通过 URL 或 XML 模板执行 SQL 查询或存储过程。

3.1 IIS 中 SQL XML 支持虚拟目录的设置

在进行具体的设置前, 首先要保证在安装 IIS 服务

的主机上安装 SQL Server 2000 的客户端，或者已将 SQL Server 2000 服务器与 IIS 安装在同一机器中。通过使用随 SQL Server 2000 提供的 SQL XML 支持的虚拟目录管理功能，可以建立某一虚拟目录与特定的 SQL Server 2000 数据库服务器之间的关联。具体的实现步履如下：

(1) 选择“程序/Microsoft SQL Server/在 IIS 中配置 SQL XML 支持”，进入“对 SQL Server 的 IIS 虚拟目录管理”，并选择管理器中相应的 Web 站点，单击右键，选择“新建/虚拟目录”，进入虚拟目录设置对话框。

(2) 在虚拟目录设置对话框的常规选项中，与一般的虚拟目录的设置方法相同，分别指定虚拟目录名及相应的物理目录。如设置虚拟目录名为“querys”，而对应的物理目录为“E:\Inetpub\wwwroot\queries”。

(3) 在安全性选项页中设置用户使用数据时所使用的的安全性设置。具体的设置方法与安装的 SQL Server 的用户设置有关。

(4) 在数据源选项页中设置所要访问的 SQL Server 数据库所在的位置及所要访问的数据库的名称。在这里，所使用的数据库与 IIS 安装在同一台机器上，而使用的数据库名称为 SQL Server 所带的示例数据库“Northwind”。

(5) 在“设置”选项页中设置该虚拟目录提供的对 SQL Server 的访问类型。各项具体设置的意义将在下面进行更详细的说明。

(6) 在“虚拟名称”选项页中，可以对虚拟目录与特定类型之间的关联关系；而在“高级”选项页中，可以查询与设置该虚拟目录所使用的 ISAPI 文件等。

通过以上六个步骤的设置，就完成了访问 SQL Server 的虚拟目录的设置。通过对管理器中列表中的某一虚拟目录双击或使用右键快捷菜单，可以对该目录的设置进行修改。

3.2 通过 URL 执行 SQL 查询

在完成 SQL Server 的 IIS 虚拟目录设置后，就可通过 URL 直接访问 SQL Server 数据库了，例如，通过在地址栏中输入下面的包含带参数的 SQL 查询语句（注意在输入时不要拆行）的 URL：

```
http://localhost/querys?sql=SELECT * FROM Customers
```

```
WHERE CustomerID='ALFKI' FOR XML AUTO &root=root
```

就可得到相应的查询结果如下代码所示，显然，通过这种方法的确能得到用 XML 表示的查询结果。

```
<?xml version="1.0" encoding="utf-8" ?>
- <root>
    <Customers CustomerID="ALFKI"
    CompanyName="Alfreds Futterkiste" ContactName="Maria
    Anders" ContactTitle="Sales Representative"
    Address="Obere Str. 57" City="Berlin" PostalCode="12209"
    Country="Germany" Phone="030-0074321" Fax="030-
    0076545" />
</root>
```

在上面使用的包含 SQL 语句的 URL 中，其中“localhost”是所对应的主机的名称，这里使用的本机，有实际应用中可以使用 IIS 服务器的 IP 地址或者相应的 DNS；而其中的“QueryString”参数“SQL”引入相应的查询语句，查询语句中的选项“FOR XML AUTO”设置查询结果 XML 表达及其格式，而另一个“QueryString”参数“root”，则设置 XML 的根元素的名称，可以通过改变设置值或取消该参数的设置查看相应的效果。

事实上，可以通过 URL 执行各种各样的 SQL 查询。假定在 SQL Server 服务器中创建了存储过程“sp_GetXml”，代码如下：

```
CREATE PROCEDURE sp_GetXml
( @CustomerID varchar(5) )
AS
BEGIN
SELECT CustomerID, CompanyName, ContactName
FROM Customers
WHERE CustomerID LIKE @CustomerID + '%' FOR
XML AUTO
END
```

则可通过如下的 URL 执行该存储过程：

```
http://localhost/northwind?sql=exec+sp_GetXml+'A'&-
root=root
```

3.3 通过 XML 模板文件访问 SQL Server 数据库

通过 URL 的查询参数执行 SQL 查询是一个很有意义的功能，但是，一方面在浏览器的地址栏中直接输入 SQL 语句是一件相当麻烦的事件，而且浏览器一般都限制了在地址栏中输入的文本的长度，这就使得某些工作在地址栏中直接输入 SQL 语句不可行；另一方面，由于可以通过 URL 执行各种各样的 SQL 语句，这就意味着用户可以直接通过 URL 实现对数据库的插入、修改与删除等操作，对数据安全性有很大影响。为此，SQL Server 的 IIS

虚拟目录还提供了通过 XML 模板文件,而在 XML 模板文件使用 SQL 语句来实现对 SQL Server 访问的选项。

通过使用 XML 模板文件,将有关对数据库的处理隐藏在相应的 XML 文件中,使所有对数据库的访问都是通过 XML ISAPI 来实现,增强了系统的安全性。在 SQL Server 的 XML 支持的设置上,应该取消虚拟目录上的“通过 URL 执行 SQL 查询”的服务。

下面通过两个具体的示例来说明如何通过 XML 模块访问 SQL Server 数据库。

要通过 XML 模板文件访问 SQL Server 数据库服务器,首先要在 SQL XML 的 IIS、虚拟目录管理器中,设置相应的虚拟目录的“设置”选项页中的“允许模板查询”,并在“虚拟名称”选项页中设置相应的映射关系。在这里的示例中,设置虚拟名称为“template”,选择类型为“template”,对应的目录为“E:\Inetpub\wwwroot\queries”。

用适当的 XML 编辑器,如书写板等创建 XML 文件“SimpleSelect.xml”,并保存在上面设置的虚拟目录中。XML 文件的代码如下:

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-
sql">
  <sql:query>SELECT CompanyName, ContactName,
ContactTitle
FROM Suppliers
FOR XML AUTO
</sql:query></ROOT>
```

然后,通过下面的 URL 则可访问以 XML 形式表达的满足条件的数据。

<http://localhost/queries/template/SimpleSelect.xml>

而且,可以在模板文件中嵌入带参数的 SQL 语句,以返回满足条件的信息,如下面的 XML 模板文件就能正确地返回满足条件“UnitPrice >10”的数据。

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-
sql">
  <sql:query>
SELECT   OrderID, UnitPrice, Quantity
FROM     [Order Details]
WHERE    UnitPrice > 10
FOR XML AUTO
</sql:query>
</ROOT>
```

事实上,也能在 XML 模板文件中嵌入执行存储过程的 SQL 语句,如下面的 XML 就能正确地执行存储过程“CategoryInfo”。

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-
sql">
  <sql:query>
exec CategoryInfo
</sql:query>
</ROOT>
```

同样,也可通过使用 XML 模板文件,方便地执行带参数的 SQL 语句或存储过程。如创建如下的 XML 模板文件“paraselect.xml”:

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-
sql">
  <sql:query>
SELECT   OrderID, UnitPrice, Quantity
FROM     [Order Details]
WHERE    UnitPrice > @Price
FOR XML AUTO
</sql:query>
</ROOT>
```

就可通过如下的 URL 进行动态查询。

<http://localhost/queries/template/paraselect.xml?Price=10>

3.4 XSL 样式表的使用

从上面的讨论可知,在设置好 SQL XML 支持的 IIS 虚拟目录后,可以通过多种方式访问 SQL Server 中的信息,但是,上面所有方法的返回结果都是 XML 形式的,这与一般习惯的 Web 页的表达方式有较大差异。为了有效地表现 XML 数据,必须使用样式表文件 XSL。下面以实例的方式来原因具体的过程。

定义如下的 XSL 样式表文件“custtable.xsl”

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/
WD-xsl">
  <xsl:template match="/">
  <html> <head> <title>Customer List</title> </head>
  <body><h1>Northwind Customers</h1>
  <table border="1">
  <tr>
  <th>Customer ID</th>
  <th>Company Name</th>
```

```

        <th>Contact</th>
        <th>Phone</th>
    </tr>
    <xsl:for-each select="root/Customers">
        <tr>
            <td><xsl:value-of select="@CustomerID"/></td>
            <td><xsl:value-of select="@CompanyName"/></td>
            <td><xsl:value-of select="@ContactName"/></td>
            <td><xsl:value-of select="@Phone"/></td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

可以通过下面的 URL 执行内嵌的 SQL 语句，并将所获得的数据用上面定义的样式表文件显示。注意，在该 URL 中使用了两个新的“QueryString”参数，其中“xsl”参数指定显示查询结果的样式表文件，而参数“contenttype”指定返回内容用 html 格式显示，如果将该参数值改为“text/xml”，则不论是否使用样式表参数，结果总是以 xml 形式显示。

```

http://localhost/demos?sql=select+'<root>';
select+*+from+Customers+as+Customer+
FOR+XML+auto;select+'</root>'+&xsl=custtable.
xsl&contenttype=text/html

```

为了在 XML 模板文件中使用样式表，需要通过标记“sql:xsl”来实现，下面在模板中使用上面定义的样式表的 XML 示例模板文件代码：

```

<root xmlns:sql="urn:schemas-microsoft-com:xml-sql"
sql:xsl="custtable.xsl" >
    <sql:query>
        SELECT *
        FROM Customers AS Customer
        FOR XML auto
    </sql:query>
</root>

```

而为了在浏览器中将 XML 模板执行的结果用网页的方式来显示，可以使用下面的 URL，要注意的是，“QueryString”参数串“contenttype=text/html”不能省略，

否则结果将仍是使用指定样式表后的 XML 文本。

```

http://localhost/querys/template/test.xml?
contenttype=text/html

```

4 方案技术展望

基于 SQL Server 2000 XML 技术设计的数据驱动 Web，与 CGI/ISAPI、ADO 及 ADO+ 等技术设计的数据驱动 Web 相比，具有下面的特点：

- (1) 与操作系统平台、浏览器厂家及版本无关的交互式数据驱动：所实现的 Web 仅需要能解析 XML 的浏览器就可使用；
- (2) 自动缓冲数据库连接：开发者不再需要通过编程来管理连接；
- (3) 快速页面生成：不需要使用 ADO 或 ADO+ 等数据访问组件，也不需要使用 ASP 代码来生成页面，因而 Web 页生成速度更快，而且通过使用数据缓冲技术，可以更加提高 Web 速度；
- (4) 管理简单：一般比相应的 ASP 实现方案代码量少，再加上通过使用可重用的 XSL 模板，将应用的表现层与商业逻辑层进行分离，简化了代码与整个方案的管理；
- (5) 方案灵活，应用面广：因为使用了 XML 技术组织数据，为开发离线应用、不同环境的应用集成、多层应用开发等提供了解决方案。

要注意的是，虽然该方案为设计数据驱动 Web 提供了一个很好的解决途径，但并不是所有的数据驱动 Web 都适合采用该方案来解决。该方面至少存在以下两方面的不足：

- ① 在服务器端必须使用特定的环境：数据库服务器必须使用 SQL Server 2000，当然意味着操作系统也必须使用 Microsoft 的 Windows 操作系统；
- ② 程序设计模式的改变：在基于 CGI/ISAPI、ASP 技术实现的数据驱动 Web 设计中，商业逻辑主要是通过组件的方式来实现，而在本方案中，商业逻辑将主要实现在数据库服务器的存储过程或 XLS 文件中。■

参考文献

- 1 刘光编者, XML 应用技巧与实例, 北京清华大学出版社, 2000. 10.
- 2 彭江平等编著, Internet/Intranet 的开发与应用, 成都电子科技大学出版社, 1999. 1.
- 3 彭江平等, 基于 Java 与 XML 的数据库应用, 北京全国第三届 Java 技术及其应用会议, 2000. 8.