

# 利用 SQLXML 调用存储过程的方法

## A Method of Calling Stored Procedure by Using SQLXML

蔡瑞奎 (广州 广东商学院教育技术中心 510320)

**摘要:**本文利用 SQLXML 3.0,通过对数据库存储过程的调用,达到对数据按 XML 方式处理的目的,对数据转换很有用处。利用查询的存储过程,演示以 HTTP 方式按 XML 格式输出 SQL 数据库数据的方法。该方法为提升 SQL Server 2000 对 XML 数据文件的处理能力有很好效果。

**关键词:**SQLXML 存储过程 HTTP

### 1 引言

近年来,随着互联网的广泛流行和普及,数据处理量也随之增长,特别是处理网上各式各样的复杂数据尤为重要。为促使处理数据技术的进一步发展,人们投入更多的人力、物力研究更为完善的数据管理技术。目前,作为 Web 环境中组织数据的一种方式,常见的 HTML 描述了显示全球数据的通用方法,而 XML<sup>[1][2][3]</sup>提供了直接处理全球数据的通用方法。XML 使用一个简单而有灵活的标准格式,为基于 Web 的应用提供了一个描述数据和交换数据的有效手段。

以前的很多研究都是 SQL<sup>[4]</sup>与 XML 相互独立开来进行,本着对数据处理的相同目的,有必要把两者有机结合起来,使两者能相辅相成,在 SQL 基础上进一步发挥处理数据的更大潜能<sup>[5]</sup>。

实际上,面对庞大的 XML 数据量,有几个重要问题需要解决。例如,利用什么技术或工具可以从大型 XML 文档中抽取数据;可以在不同的 DTD 之间翻译 XML 数据;可以从多个 XML 文档中结合数据;或者可以传送大量的 XML 数据。

数据抽取、传送、合并是传统的数据库问题,这些问题的解决得益于结构化查询语言 SQL 的实现。但是 SQL 不能直接运用到 XML 中,因为 XML 中的数据结构不仅包含传统的结构化数据格式,而且还包含大量的半结构化数据。即单纯的 SQL 数据库对异构数据处理已经有较大的局限性。随着 XML 对异构数据的优异处理能力,利用 XML 结合 SQL 进行数据处理工作越来越受到关注。

### 2 SQLXML 3.0

#### 2.1 SQLXML 3.0 简介

SQLXML 3.0<sup>[6]</sup>是由微软公司开发设计的,主要有以下特点:

(1) 支持服务器端及客户端 FOR XML。SQL Server 2000 引进了 FOR XML 支持。SQLXML 3.0 允许设置服务器端和客户端查询结果的格式。

(2) XML 架构定义语言(XSD)。可以通过使用批注创建关系数据的 XML 视图来编写 XSD 架构。

(3) 支持 Updategram 或 DiffGram 格式。在现有的 XML 文档中,可以通过 Updategram(更新元)或 DiffGram(差异元)来修改(插入、更新或删除)SQL Server 2000 中的数据库数据。

(4) 大容量装载 XML 数据。可以通过 SQLXML 中的“XML 大容量装载”支持来大容量装载 XML 文档。

(5) 对 Microsoft .NET 框架的支持。SQLXML 托管类表现了 SQLXML 3.0 在 Microsoft .NET 框架中的功能。

(6) Web 服务(SOAP)支持。可以向 SQLXML 服务器发送 SOAP 请求以执行存储过程、用户定义的函数和模板。

#### 2.2 SQLXML 3.0 配置

配置设定要点:

(1) 默认 WEB 站点|右键|新建|虚拟目录,在“虚拟目录名称”填入“XMLSQL”。

(2) 安全性:使用 Windows 集成身份验证。因为

要与 SQL 登陆验证方式一致。否则,可能找不到数据源。

(3) “当前登陆使用默认数据库”选项去掉。可以方便设定要连接的数据库。

(4) 允许 SQL = ... 或 TEMPLATE = ... URL 查询。

### 3 扩展语法简介

微软公司通过增加一个 XML 语句使得 SQL Server 2000 能返回支持 XML 格式数据,它是加在 SELECT 后面来实现的。

#### 3.1 SQLXML 的 FOR XML 语句及其模式

基本语法如下:

```
SELECT FOR XML mode [, XMLDATA] [, ELEMENTS] [, BINARY BASE64]
```

其中 mode 模式变量有 RAW, AUTO, EXPLICIT 或 NESTED 等几种选择。

对于查询请求, FOR XML 模式通过返回匹配的行的 XML 格式来显示查询结果而不是以前的行集合。SQLXML 提供两个方法生成 XML 格式文档:服务器端格式和客户端格式。

##### 3.1.1 服务器端输出 XML 文档格式

此时在服务器端先执行传统的查询命令,再按 FOR XML 部分语句完成转换,并把所生成的行集合以 XML 格式送到客户端。

下表 1 显示 FOR XML 语句服务端可使用的模式:

表 1 服务器端模式表

FOR XML 模式	含义
For XML RAW	RAW 模式简单返回以“row”为标签的 XML 元素。返回的 XML 元素与每条记录相匹配,并且列名与 XML 元素的属性相匹配(除非用其他别名)。
For XML AUTO	AUTO 模式分别按元素来匹配各个表,并按照表之间的关系嵌套。缺省情况下,列即为 XML 元素的属性,如果是按“join”连接多个表的,连接的表作为子元素。它同时能识别“ELEMENTS”参数。查询语句中表的顺序决定元素和子元素的嵌套。
For XML EXPLICIT	最清晰的输出模式。EXPLICIT 模式给开发者提供最灵活的手段。它描述列与 XML 元素和属性如何匹配的特定格式查询。

##### 3.1.2 客户端输出 XML 格式文档

这时服务器端执行传统的查询而不执行后面 FOR XML 语句部分,而是中间层使用 FOR XML 转换并返回 XML 文档到客户端。

下表 2 显示 FOR XML 客户端可使用的模式:

表 2 客户端模式表

FOR XML 模式	含义
For XML RAW	当 FOR XML 在服务器端或客户端指定时产生相同结果。
For XML NESTED	与服务器端 FOR XML AUTO 模式产生的结果相类似。
For XML EXPLICIT	与服务器端的 FOR XML EXPLICIT 模式相类似。

#### 3.2 其他可选参数

FOR XML 语句中,可选参数提供对生成的 XML 格式的细节控制,见下表 3。

表 3 FOR XML 其他选项表

XMLDATA	指定生成返回 XML 文档的数据格式的 XDR(数据简化 XML),同时生成一个包含在文档里的 XML 数据架构(schema)。主要针对数字格式文档。
ELEMENTS	确保返回查询中的列名作为子元素而不是属性,它指定各列作为子元素而不是属性返回。该参数仅在 AUTO 模式下有效。
BINARY BASE64	指定以 BASE 64 编码格式的二进制数据返回,在 RAW 或 EXPLICIT 模式下,要返回二进制数据时必须提供该选项。

不过, FOR XML 查询语句返回的是 XML 结构不完整的数据格式,不能把结果直接以单个 ROOT 元素简单加入现有的 XML 文档。因为三种模式下返回的 XML 都没有根节点,因此可以使用生成存储过程使结构加以完整。

以上各种组合方式,在此不能一一示例,仅以下例做简单说明:

```
SELECT
au_id as AuthorId,
au_fname as FirstName,
```

```
au_lname as LastName
FROM authors FOR XML RAW, XMLDATA
```

如果在 SQL Query Analyzer 中,以输出文件形式获得的 XML 结果(片段):

```
<Schema name = " Schema1" xmlns = " urn : schemas
- microsoft - com : xml - data" xmlns : dt = " urn : sche
mas - microsoft - com : datatypes" >
< ElementType name = " row" content = " empty"
model = " closed" >
< AttributeType name = " AuthorId" dt : type = " string" /
>
< AttributeType name = " FirstName" dt : type = "
string" / >
< AttributeType name = " LastName" dt : type = "
string" / >
< attribute type = " AuthorId" / >
< attribute type = " FirstName" / >
< attribute type = " LastName" / >
</ ElementType >
</ Schema >
< row xmlns = " x - schema : #Schema1" AuthorId = "
409 - 56 - 7008" FirstName = " Abraham" LastName = "
Bennet" / >
< row xmlns = " x - schema : #Schema1" AuthorId = "
648 - 92 - 1872" FirstName = " Reginald" LastName = "
Blotchet - Halls" / >
< row xmlns = " x - schema : #Schema1" AuthorId = "
238 - 95 - 7766" FirstName = " Cheryl" LastName = "
Carson" / >
< row xmlns = " x - schema : #Schema1" AuthorId = "
722 - 51 - 5454" FirstName = " Michel" LastName = "
DeFrance" / >
< row xmlns = " x - schema : #Schema1" AuthorId = "
712 - 45 - 1867" FirstName = " Innes" LastName = " del
Castillo" / >
```

以上方法 authors 表的 au\_id, au\_fname, au\_lname 字段分别与 XML 格式文件的 AuthorId, FirstName, LastName 属性相对应。虽然在 SQL Query Analyzer 环境中可以获得 XML 文档,但是不直观,XML 文件内容不完整。在此基础上,本文实现一个更直观的

方法,主要是利用 SQLXML 3.0 调用存储过程,在 HTTP 下输出 XML 文件的方法。

## 4 存储过程使用方法

### 4.1 存储过程

存储过程是利用 SQL Server 所提供的 T-SQL 语言所编写的程序。T-SQL 语言是 SQL Server 提供专为设计数据库应用程序的语言,它是应用程序和 SQL Server 数据库间的主要程序式设计界面。使用 SQL 存储过程有以下好处:

(1) SQL 存储过程执行起来比 SQL 命令文本快得多。当一个 SQL 语句包含在存储过程中时,服务器不必每次执行它时都要分析和编译它。

(2) 调用存储过程,可认为是一个三层结构。这使程序易于维护。如果程序需要改动,也只要改动存储过程即可。

(3) 可以在存储过程中利用 T-SQL 的强大功能。一个 SQL 存储过程可以包含多个 SQL 语句,并可以使用变量和条件。这意味着可以用存储过程建立非常复杂的查询,以非常复杂的方式对数据库进行操作。

(4) 更为重要的是在存储过程中可以使用参数,它可以传送和返回参数。

下例是对 SQL 数据库 authors 表查询的存储过程,查询结果按作者 Author 及其属性 ID 为父元素,FirstName 为子元素的在 HTTP 上按 XML 格式显示,其根节点为 Authors。

```
CREATE PROCEDURE spXMLAuthors
AS
SELECT '< Authors >'
SELECT 1 AS Tag,
      null AS Parent,
      au_fname AS
[ Author! 1! FirstName! Element ],
      au_id AS [ Author! 1! ID ]
FROM authors FOR XML EXPLICIT
SELECT '</ Authors >'
GO
```

注意嵌套的 < FirstName > 元素是通过 [ Author! 1! FirstName! Element ] 来指定的。

### 4.2 在 HTTP 下直接调用存储过程

在 HTTP 下,对于简单的情况,我们可以在虚拟目录名称后直接输入 SQL 查询语句,如下所示:

HTTP://localhost/sqlxml?

sql = SELECT + firstname, lastname +

FROM + employees + FOR + XML + AUTO

但是,当处理复杂情况时,直接输入并不好。我们采用调用前面的存储过程来实现。如:

http://localhost/sqlxml? sql = execute + spXMLAuthors, 结果见图 1 所示。

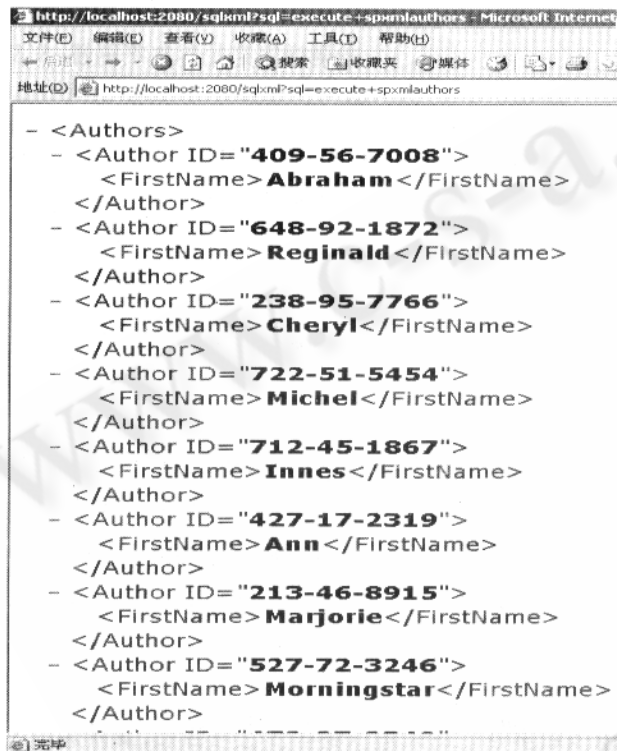


图 1 调用存储过程结果图

有了这个存储过程就不再需要创建一个冗长的 SQL 查询字符串,也就不用在 HTTP 上直接输入一长串的查询字符串来生成 XML 文档。

## 5 结论

通过 SELECT 查询语句中使用 FOR XML 查询扩展的存储过程,在 HTTP 下输出 XML 结果格式文档,达到转换数据格式的作用。其中 SQLXML 起到了桥梁的作用。该方法简单方便,功能更加强大,为 SQL 数据库数据在 HTTP 下转为 XML 格式提供另一个技术手段,为以后相关应用提供方便。不过,在实际应用中还需要

考虑数据安全性问题,以及注意多重查询的存储过程一次生成 XML 数据的复杂性。

## 参考文献

- 1 <http://www.xml.org/>
- 2 <http://www.w3.org/XML/>
- 3 Charles F. Goldfarb Paul Prescod, XML Handbook, Fourth Edition, Publishing House of Electronics Industry, 2004.
- 4 王能斌编著,数据库系统原理,电子工业出版社,2001年。
- 5 蔡瑞奎,电脑与信息技术,ASP 结合 XML 方法实现远程数据库管理,P26,2004年4月。
- 6 <http://msdn.microsoft.com/downloads/Default.aspx>