商务活动中 OPENXML 对关系数据库表的更新

Updating the Relative Database Table Using OPENXML in Business Affairs

赵俊岚 （呼和浩特内蒙古财经学院计算机信息管理系 010070）

摘要：本文着重介绍了 OPENXML 特性，并给出了通过 OPENXML 将 XML 格式的数据打开成一个表的方法。文中还给出了使用 OPENXML 特性，利用 XML 文档的数据对关系数据库中表实施插入、修改与删除的方法。

关键词：SQL Server2000　XML　关系数据库　OPENXML

1　OPENXML 简介

SQL Server2000 所呈现的最显著的新特征之一，就是它提供了与 XML 的直接集成。下一代的 Web 和 Enterprise 应用程序将会更多的使用 XML 来提供数据交换功能。因此，将 XML 集成到 SQL Server 中是大有裨益的。

标准的 SQL Server2000 中包括的 XML 特性主要为：SELECT 语句中的 FOR XML 子句、OPENXML、XML 视图。其中 OPENXML 是在内存中的 XML 文档上提供行集 Transact－SQL 的关键字。OPENXML 是与表或视图相似的行集提供程序。OPENXML 通过提供 XML 文档内部表示法的行集视图，进而允许访问 XML 数据，此时的 XML 就好像是关系或视图一样。当然，行集中的记录也可以存储在数据库的表中。使用行集中的记录就好像在一个 SELECT INTO 语句的 FROM 子句中使用表名称或视图名称一样。

```
SELECT *
INTO XML_Result_Table
FROM OPENXML (@idoc, '/ROOT/Customer',1)
WITH (CustomerID int 'ID',
      CustomerName varchar (50) 'NAME')
```

其中：@idoc 参数是希望用 OPENXML 对其进行处理的 XML 文档内部表示的一个句柄，并且通过调用 sp_xml_preparedocument 系统存储过程返回此文档句柄；'/ROOT/Customer' 为 XML 文档的 Xpath 表达式，用来标识将要处理的节点为 ROOT 节点下的 Customer 节点；1 为 Flags 参数，用于指定行集的列与 XML 节点之间的映射类型为属性中心型映射，即只映射 Customer 元素节点的属性值，此外 Flags 参数还可以取 0、2 等值，分别表示默认属性中心型映射、元素中心型映射等不同的映射类型；WITH 为可选项，用于指定将要生成行集的架构，这里指定了行集架构由名为 CustomerID、CustomerName 两个属性列组成，并具体定义了这两个属性列的类型，以及这两个属性列到 XML 文档节点 ID、NAME 的映射。

OPENXML 主要的用途在于向数据库中的表插入、更新或删除数据，该数据库的数据是基于源 XML 文档的。这一功能是十分强大的。对于两家忙于数据交换的商务公司，则可以相互在文档中收发 XML 数据（如 Microsoft BizTalk 2000），并且可以用这些 XML 文档中的数据去更新自己数据库的数据，若这两家公司有一个已知的数据格式，通过 OPENXML 技术处理过程将会相当简单。

由于 OPENXML 是行集提供程序，因此大多在行集提供程序的 Transact－SQL 中出现 OPENXML。下面给出一个将 XML 格式的数据转换成为一个行集的示例：

```
declare @idoc int
declare @txtXML varchar(1000)
--创建一个 XML 文档的内部描述
set @txtXML = '<Root>
      <download ID="1" Filename="ayako_katagiri.exe"/
>
      <download ID="2" Filename="beauti.mid"/>
      ......
      </Root>'
--调用系统存储过程，为@idoc 返回一个整数句柄，并将
@idoc 句柄指向@txtXML 变量所对应的 XML 文档 EXEC sp
_xml_preparedocument @idoc OUTPUT, @txtXML
--执行 SELECT，检索 OPENXML 所提供的行集中的列
值
SELECT *
FROM OPENXML (@idoc, '/Root/download',1)
WITH (ID# int '@ID',
    NAME varchar(50) '@Filename')
--从内存中移除 XML 文档
EXEC sp_xml_removedocument @idoc
```

执行结果为：

```
ID# NAME
```


©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

1 ayako_katagiri. exe

2 beauti. mid

……

该示例中，将 XML 文档中根元素 < Root > 下的 < download > 元素与行集中的行（记录）对应，< download > 元素中的两个属性 ID 和 Fielname 与行集中的列（字段）对应，并分别命名为 ID#和 NAME 输出。

2 商务数据交换

现有某零售商数据库所保存的一些数据，其中用到一个 Customers 客户关系数据表、一个 Products 产品供应关系数据表及一个 Type 产品类型关系数据表。

2.1 定义关系表的 schema：

```
--定义 Customers 关系数据表
CREATE TABLE [dbo].[Customers] (
  [CustomerID][int] NOT NULL ,
  [CustomerName] [varchar] (50) NULL ,
  [CustomerAddress] [varchar] (50) NULL
) ON [PRIMARY]
GO
--定义 Products 关系数据表
CREATE TABLE [dbo].[Products] (
  [ProductID] [int] NOT NULL ,
  [CustomerID] [int] NOT NULL ,
  [TypeID] [varchar] (50) NULL ,
  [ProductName] [varchar] (50) NULL
) ON [PRIMARY]
GO
--定义 Type 关系数据表
CREATE TABLE [dbo].[Type] (
  [TypeID] [int] NOT NULL ,
  [TypeName] [varchar] (50) NULL ,
) ON [PRIMARY]
GO
--定义 Customers 关系数据表上的触发器
CREATE TRIGGER trg_Customers ON [dbo].[Customers]
FOR DELETE
AS
DELETE FROM Products
WHERE CustomerID in (
  SELECT CustomerID
  FROM DELETED)
GO
```

上面定义了名为 Customers（含有：客户号、客户名、客户地址）、Products（含有：产品号、客户号、类型号、产品名）、Type（含有：类型号、类型名）的三个关系数据表，并定义了 Customers 关系数据表上的删除触发器 trg_Customers，用于删除那些产品（Products）关系数据表中客户（CustomerID）已不存在的产品。

2.2 初始源 XML 文档

为了利用 OPENXML，使得数据库中保存的信息能够反映出源 XML 文档中的数据信息，需先给出一个初始的源 XML 文档如下：

```
declare @ txtXML varchar(8000)
set @ txtXML = ' <? xml version = " 1.0 " encoding = "
GB2312 " ? >
    < ROOT >
      < CUSTOMER ID = "1" >
        < NAME > Sam </NAME >
        < ADDRESS > 25 Hillfoot Dr, Howwod </ADDRESS >
        < PRODUCTS >
          ……
        </PRODUCTS >
      </CUSTOMER >

  < CUSTOMER ID = "2" >
    < NAME > Jacquie </NAME >
    < ADDRESS > 3 station Court , Glengarnock </ADDRESS >
    < PRODUCTS >
      ……
    </PRODUCTS >
  </CUSTOMER >
      < CUSTOMER ID = "3" >
        < NAME > Glen </NAME >
        < ADDRESS > 21 corseford Ave, Wishaw </ADDRESS >
        < PRODUCTS >
          ……
        </PRODUCTS >
      </CUSTOMER >
    </ROOT > '
```

此 XML 文档以 < ROOT > 为根元素，其中包含了三组 < CUSTOMER > 记录，并将该文档赋给了变量 @ txtXML，以备后面调用。

3 OPENXML 技术对关系数据库表的更新

我们的目的是要对关系数据库加以更新，以反映出源 XML 文档中的数据改动。为此，创建如下存储过程 sp_Up-

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

dateCustomers，类似的也可以创建存储过程 sp_UpdateProducts 和 sp_UpdateType，它们均会将 XML 源数据文档作为一个参数。

可以对上面所示的 XML 文档使用这三个过程，可实现对产品(Products)和客户(Customers)数据进行初始加载(即 insert 功能)或数据更新(即 update 功能)。这些过程将使用 OPENXML 来生成用于操作的行集。sp_UpdateCustomers 过程将会检查客户记录是否存在，如果不存在，就会增加该客户记录；如果已存在的话，该过程将执行一个更新操作。

一旦对关系数据表进行了插入和更新，sp_UpdateCustomers 过程就会去检查在数据库中是否存在源 XML 中所不存在的客户记录，Customers 关系数据表上的删除触发器将负责删除那些客户不再存在的产品。

过程的源代码如下所示：

```
CREATE PROCEDURE sp_UpdateCustomers
( @txtXML varchar(8000) )
AS
declare @idoc int
--建立一个 XML 文档的内部描述
EXEC sp_xml_preparedocument @idoc OUTPUT , @txtXML
--若关系数据表中对应 XML 文档的 Customer 记录不存在，则插入该记录
INSERT Customers
SELECT CustomerID, CustomerName, CustomerAddress
FROM OPENXML(@idoc,'ROOT/CUSTOMER',1)
WITH (CustomerID int '@ID',
      CustomerName varchar (50) 'NAME',
      CustomerAddress varchar (50) 'ADDRESS')
WHERE CustomerID Not IN (
          SELECT CustomerID
          FROM Customers)
--与 XML 文档比较，对于关系数据表中已经存在的 Customer 记录，用 XML 文档中数据对其进行更新
UPDATE Customers
SET     CustomerName = O. CustomerName,
        CustomerAddress = O. CustomerAddress
FROM ( SELECT CustomerID, CustomerName, CustomerAddress
        FROM OPENXML (@idoc, 'ROOT/CUSTOMER',1)
        WITH (CustomerID int '@ID',
              CustomerName varchar (50) 'NAME',
              CustomerAddress varchar (50) 'ADDRESS')
) O
WHRER Customers. CustomerID = O. CustomerID
--删除在关系数据表中存在，而在 XML 文档中不存在的 Customer 记录
DELETE
 FROM Customers
 WHERE CustomerID NOT IN (
  SELECT CustomerID
  FROM OPENXML (@idoc, 'ROOT/CUSTOMER',1)
  WITH (CustomerID int '@ID') )
GO
```

sp_UpdateCustomers 过程中包含了对数据库表进行插入、修改、删除三方面的功能。sp_UpdateProducts、sp_UpdateType 过程的创建与 sp_UpdateCustomers 过程的创建相似，这里省略。

3 结束语

OPENXML 是用于从 XML 源中生成行集的方法。可以将 OPENXML 用于任何其他的行集提供者(如关系数据表或视图)可以使用的地方。OPENXML 的最大好处是能够使用包含在源 XML 文档中的数据来更新数据库(即能够使用 OPENXML 向数据库中插入、更新及删除数据)。由于能够以一种通用的通信模式来影响商务活动数据，这将使得 OPENXML 会被广泛应用于接口程序的开发中。

参考文献

1 Microsoft Corporation. SQL Server 2000 体系结构与 XML/Internet 支持[M]，清华大学出版社，2001.8。

2 Marci Frohock Garcia. SQL Server 2000 系统管理员宝典[M]，清华大学出版社，2001.7。

3 Paul J. Burke. SQL Server 2000 XML 高级编程[M]，中国电力出版社，2003.1。

4 Heather Williamson. XML 技术大全[M]，机械工业出版社，2002.1。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn