SQL SERVER 2005 中 XML 类型数据的输入方法

Input Method of XML Type Data in SQL SERVER 2005

王海林 （山西财经大学 信息管理学院 山西太原 030006）

摘要：Microsoft SQL Server 2005 为关系数据表的字段新增了 XML 数据类型，我们可以很方便地把 XML 文档存入到 XML 类型字段中，从而使 XML 文档数据纳入到关系数据库管理系统的管理当中，实现了 XML 文档数据与关系型数据存储和管理的有机统一。但 XML 数据毕竟不同于关系型数据，它的数据输入有其特殊性，所以文中通过示例详细介绍几种 XML 数据的输入方法。

关键词：SQL SERVER 2005　XML 数据类型　XML 架构　数据输入

1　引言

由于互联网规模的不断扩大和应用的日益普及，人们通过计算机上网看新闻、查资料、下载软件、收发邮件、从事商务活动已成为寻常之事，但无论做什么，都需要接收和发送大量数据，也就是要进行大量的数据交换。而数据交换面临的最大问题是数据格式不统一，这给数据的有效使用带来了巨大的障碍，因此如何快捷、可靠、有效地获取和共享数据就成为人们所关注的问题了。人们期待着能够找到一种可以描述任何逻辑关系的数据格式来统一电子数据的存储和交换，从而不再因为数据格式的不统一而苦恼和困惑。目前，能够担当此任的就是 XML(eXtensible Markup Language，可扩展标记语言)。可以说，XML 的出现给数据交换带来了一场革命。XML 也将成为未来网络发展的基础。

XML 与平台无关的数据表示特性，使得它很快成为了互联网上数据交换的标准。为了支持这一标准，关系数据库界付出了巨大的努力，特别是新近推出的数据库产品 Microsoft SQL Server 2005、DB2 9、Oracle 10g(第 2 版)等都有了重大突破，它们都增加了 XML 数据类型。这种数据类型可以用于变量和存储过程的参数，也可以用作关系表中的字段数据类型。我们可以通过创建一个 XML 数据类型字段来存储 XML 文档，从而实现关系数据与 XML 数据共存于同一关系数据库的夙愿。

为什么使用关系数据库来存储 XML 数据呢？因为将 XML 数据存储在关系数据库中会给数据管理和查询处理带来很多好处。如 SQL Server 2005 提供了强大的查询和修改关系数据的能力，而且已经扩展到查询和修改 XML 数据。这使得可以利用在过去的版本上所进行的投资。又如，关系数据库中的索引技术已经广为人知，而且已经扩展到用于索引 XML 数据，这样就可以使用基于成本的决策来优化查询。另外，将 XML 数据与关系数据共存于同一关系数据库还有一个好处，就是 XML 数据可以与现有的关系数据以及 SQL 应用程序进行互操作，这样就可以在需要进行数据建模而又不破坏现有的应用程序的系统中引入 XML。数据库服务器还提供了管理功能来管理 XML 数据(例如备份、恢复和复制)。

但 XML 数据毕竟不同于关系数据，首先表现在关系表中数据的输入上。XML 字段的数据输入有其特殊性，我们不能在表的编辑窗口中以交互方式直接将 XML 文档数据输入到其字段对应的单元格中，必须使用其它方法。鉴于 SQL SERVER 2005、DB2 9、Oracle 10g(第 2 版)推出时间还不长，许多人对它们提供的 XML 数据输入方法还不够了解，为此本文以 SQL SERVER 2005 为例来详细介绍 XML 数据的输入方法。

2　非类型化 XML 数据的输入

SQL Server 2005 的 XML 数据类型实现了 ISO SQL－2003 标准 XML 数据类型。因此，它不仅可以存储格式良好的 XML 1.0 文档，而且还可以存储 XML 内容片段(文档的节点和元素)。在对数据进行格式良好性检查时，并不要求将 XML 数据类型绑定到 XML 架构，这类数据称为非类型化 XML 数据。

当架构不是已知先验的，并且因此而导致基于映射的解决方案不可能实现时，就可以使用非类型化的 XML。如果架构是已知的，但映射到关系数据模型非

常复杂并且难于维护，或者存在多个架构而且这些架构是后来根据外部要求绑定到数据的，也可以使用非类型化的 XML。

在介绍非类型化 XML 数据之前，我们先创建一个包含非类型化 XML 字段的关系表。该关系表为读者表 wreader。创建 wreader 表的 T－SQL 语句如下：

```
create table wreader (
  mo char(6) ,
  mame char(10),
 rcontact xml );
```

其中 mo 为读者编号，mame 为读者姓名，rcontact 为读者联系方式。mo 和 mame 为关系数据类型，而 rcontact 为 XML 数据类型。

示例一：使用 insert into 语句直接输入 XML 数据。

```
insert into wreader values('050101','张华',
'<cont>
 <phone>
  <home>7210270</home>
  <office>4040112</office>
  <cell>13310203045</cell>
 </phone>
 <address>迎泽大街 101 号</address>
 <fax>2026468</fax>
 <email>zhanghua1090@163.com</email>
</cont>')
```

语句中 XML 数据是以字符串的形式提供的。

执行结果为（如图 1）：

结果　消息

	mo	mame	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112</office><cell>133102...

图 1

用鼠标双击 rcontact 列中的单元格会显示完整的 XML 数据如下（图 2）：

示例二：可以先输入 XML 内容片段（如文档的根节点），然后通过修改语句修改 XML 字段值达到输入数据的目的，如：

```
insert into wreader values('050102','吴小军','
<cont />')
```

执行结果为（如图 3）：

```
update wreader set rcontact.modify('
insert(
 <phone>
  <office>3145787</office>
  <cell>13565710205</cell>
 </phone>,
 <address>解放南路 65 号</address>)
as first into ( /cont)[1]')
where mo='050102'
```

执行结果为（如图 4）：

示例三：示例一、二中输入的 XML 文档相当简单。如果文档很大或者很复杂，把 XML 数据键入到 INSERT 语句中是不切实际的。此时我们可以用下面介绍的方法将文档数据导入表中。假设 XML 文档文件名为 g:\whl\lmcont.xml，内容如下（图 5）：

执行下面的语句：

```
insert into wreader(mo,mame,rcontact)
select '050103','李明', * from openrowset(
  bulk 'g:\whl\lmcont.xml',
  SINGLE_BLOB) as x
```

结果为（如图 6）：

示例四：与示例二类似，也可以先输入关系数据，XML 列的数据暂时不输入（为 NULL 值），然后通过修改语句将 XML 文档数据导入表中，如：

```
insert into wreader values('050203','王国中',
null)
```

执行结果为（如图 7）：

```
update wreader
set rcontact=(
select * from openrowset(
                bulk 'g:\
whl \ zgzcont.
xml',
                SIN-
GLE_BLOB) as
x
      )
WHERE mo ='050203'
```

执行结果为（如图 8）：

3　类型化 XML 数据的输入

为了验证 XML 数据，需要定义一个 XML 架构集合，来指定可接受的 XML 元素、它们的顺序和数据类型等等。XML 架构集合是一个 W3C 行业标准并且是用 XML 编写的。

如果 XML 架构集合中有描述 XML 数据的 XML 架构，就可以将 XML 架构集合与产生类型化 XML 的 XML

rcontact1.xml | CD-790ECDEE...LQuery1.sql* | 摘要

```
<cont>
  <phone>
    <home>7210270</home>
    <office>4040112</office>
    <cell>13310203045</cell>
  </phone>
  <address>迎泽大街101号</address>
  <fax>2026468</fax>
  <email>zhanghua1090@163.com</email>
</cont>
```

图 2

结果 | 消息

	rno	rname	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112...
2	050102	吴小军	<cont />

图 3

结果 | 消息

	rno	rname	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112</office>
2	050102	吴小军	<cont><phone><office>3145787</office><cell>13565710205</cell

图 4

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <cont>
3    <phone>
4      <home>7662490</home>
5      <office>5440712</office>
6      <cell>13945204364</cell>
7    </phone>
8    <address>西矿街23号</address>
9    <email>liming3456@sohu.com</email>
10 </cont>
```

图 5

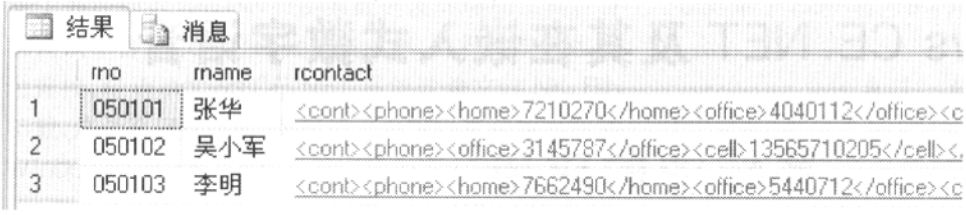
列相关联。可以使用 XML 架构来验证数据的有效性，在编译查询和数据修改语句时执行比非类型化的 XML 更精确的类型检查，以及优化存储和查询处理。

首先创建一个 XML 架构集合

```
create xml schema collection mycontact as
N'<? xml version="1.0" encoding="UTF-16"? >
<xsd: schema xmlns: xsd = " http: //www. w3. org /2001 / XMLSchema"
    xmlns = " http: //mycontact"
    elementFormDefault = " qualified"
    targetNamespace = " http: //mycontact">
  <xsd: element name = " cont" type = "contactType" />
  <xsd: complexType name = "contactType">
    <xsd: sequence>
      < xsd: element name = " phone" type = " phoneType" />
      < xsd: element name = "address" type = " xsd: string" />
      < xsd: element name = " fax" type = "xsd: string" />
      < xsd: element name = " email" type = " xsd: string" />
    </xsd: sequence>
  </xsd: complexType
  <xsd: complexType name = "phoneType">
    <xsd: sequence>
      <xsd: element name = " home" type = " xsd: string" />
      <xsd: element name = " office" type = " xsd: string" />
      <xsd: element name = "cell" type = "xsd: string" />
    </xsd: sequence>
  </xsd: complexType>
</xsd: schema>'
```

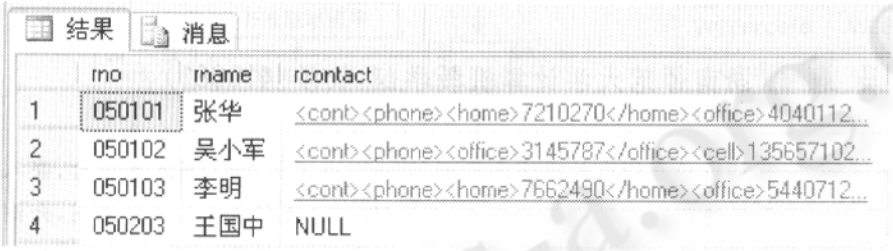
下面新建一个读者表 wreader1，其结构与 wreader 表相同，只是表中各列上增加了约束。

```
create table wreader1 (
  mo char(6) primary key,
```



	rno	rname	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112</office><c
2	050102	吴小军	<cont><phone><office>3145787</office><cell>13565710205</cell><,
3	050103	李明	<cont><phone><home>7662490</home><office>5440712</office><c

图 6



	rno	rname	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112...
2	050102	吴小军	<cont><phone><office>3145787</office><cell>135657102...
3	050103	李明	<cont><phone><home>7662490</home><office>5440712...
4	050203	王国中	NULL

图 7

	rno	rname	rcontact
1	050101	张华	<cont><phone><home>7210270</home><office>4040112</office><cell>13
2	050102	吴小军	<cont><phone><office>3145787</office><cell>13565710205</cell></phone
3	050103	李明	<cont><phone><home>7662490</home><office>5440712</office><cell>13
4	050203	王国中	<cont><phone><cell>13993274963</cell></phone><email>wangguozhong@

图 8

	rno	rname	rcontact
1	060207	王小力	<cont xmlns="http://mycontact"><phone><home>7210270</home><office...

图 9

mame char(10) not null,
rcontact XML(mycontact))

其中 rno 为主索引，mame 不允许为空，rcontact 为类型化 XML 列，它绑定到 mycontact 架构集合。

示例五：输入类型化 XML 数据。

```
insert into wreader1 values('060207','王小力',
<? xml version="1.0"? >
<cont xmlns="http://mycontact">
 <phone>
  <home>7210270</home>
  <office>4040112</office>
  <cell>13310203045</cell>
 </phone>
 <address>迎泽大街 101 号</address>
 <fax>2026468</fax>
 <email>zhanghua1090@163.com</email>
</cont>')
```

执行结果为(如图 9)：

输入数据正确，顺利通过验证。

示例六：输入类型化 XML 数据。

```
insert into wreader1 values('060508','刘一凡',
'<? xml version="1.0"? >
 <cont xmlns="http://mycontact">
  <phone>
   <home>4214020</home>
   <office>7070172</office>
   <cell>13518203640</cell>
   <xiaolingtong>8993298</xiaolingtong>
  </phone>
 <address>迎泽大街 101 号</address>
 <fax>2026468</fax>
 <email>zhanghua1090@163.com</email>
</cont>')
```

执行出错，显示错误信息为：

消息 6923，级别 16，状态 1，第 1 行

XML Validation: Unexpected element(s): http://mycontact:xiaolingtong. Location: /*:cont[1]/*:phone[1]/*:xiaolingtong[1]

(下转第 111 页)

(上接第 107 页)

原因是输入的 XML 文档数据中包含了元素 xiaolingtong,而在 mycontact 架构集合中并未定义元素 xiaolingtong,所以输入数据不正确,验证不能通过。

4 结束语

本文以 SQL Server 2005 为例介绍了 XML 数据类型、XML 架构集合的概念和创建方法以及类型化 XML 数据和非类型化 XML 数据的输入方法,全部示例均在 SQL Server 2005 系统环境中通过了测试。不过需要指出的是,其它数据库管理系统(如 DB2 9、Oracle 10g)的 XML 数据输入方法和使用的语句格式与文中所介绍的略有不同,限于篇幅,笔者不能一一介绍,感兴趣的读者可以查阅相关资料。

参考文献

1 蔡翠平,从 HTML 到 XML[M],北京:北方交通大学出版,2002.

2 Sun Microsystems. XML Schema 定义[DB/OL], http://www.huihoo.com/one_and_net/app7/14.1.htm,2003.

3 Microsoft Corporation. SQL Server 2005 联机丛书:OPENROWSET (Transact - SQL)[DB/OL], http://msdn2.microsoft.com/zh-cn/library/ms190312.aspx,2005.

4 Microsoft Corporation. SQL Server 2005 联机丛书:? CREATE XML SCHEMA COLLECTION (Transact - SQL)? [DB/OL], http://msdn2.microsoft.com/zh-cn/library/ms176009.aspx,2005.

5 Rajasekar. XML - to - SQL Query Translation [A]. 20th International Conference on Data Engineering[C]. Boston: IEEE Computer Society Press,2004: 42-53.

6 C. M. Saracco. DB2 Viper 快速入门 [EB/OL]. http://www-128.ibm.com/developerworks/cn/db2/library/techarticles/dm-0603saracco/,2006-03.