



中华人民共和国审计署

National Audit Office of the People's Republic of China



站内搜索 站外 搜索

设为首页 | 联系我们 | English

当前位置: 首页 > 审计资讯 > 理论与方法

浅谈SQL Server对XML的支持及其在审计中的应用

暴益铭 陈佳声（审计署沈阳办）

【时间: 2013年02月20日】

【来源: 】

【字号: 大 中 小】

在开展计算机审计的过程中, 审计人员除了面对SQL Server、Oracle、Access、Excel 等常见的数据库外, 难免遇到一些审计中不常见的“非主流”格式电子数据。如, 笔者所在的审计组在开展高效节能空调推广资金审计时, 需采集某企业空调在“家电下乡”活动中享受补贴数据并与节能推广数据进行比对筛查是否有重复现象, 而该企业通过商务部门“家电下乡”信息管理系统导出的产品享受补贴的数据却是XML文件格式。因此, 审计人员需要对该XML文件进行处理以“为我所用”。

一、XML技术简介

XML是可标记扩展语言eXtensible Markup Language的缩写, 它将用户接口与结构化数据分隔开来, 实现数据与显示的分离, 使其逐渐成为因特网上数据表示和数据交换的新标准。用XML进行数据管理具有以下几个特点:

- (一) XML是纯文本, 开发简单;
- (二) 基于内容的数据标识可格式化及可扩展性;
- (三) 可用CSS、XSL自定义数据的显示方式;
- (四) 与平台无关, 可在Windows、Linux等任意平台任选一解析器进行解析。

如, “家电下乡”信息系统中提供给生产厂家的享受补贴明细XML文件内容如下(部分):

```

<?xml version="1.0" encoding="gb2312"?>
<ROOT>
  <PRODUCT>
    <PD_NAME>空调</PD_NAME>
    <PD_ID>8020971</PD_ID>
    <PD_TYPE>6</PD_TYPE>
    <PD_SERIAL>NB2990050126</PD_SERIAL>
  </PRODUCT>
  <PRODUCT>
    <PD_NAME>空调</PD_NAME>
    <PD_ID>8020971</PD_ID>
    <PD_TYPE>6</PD_TYPE>
    <PD_SERIAL>NB2990050226</PD_SERIAL>
  </PRODUCT>
  .....
</ROOT>

```

每个XML文档都由XML序言开始, 在前面的代码中的第一行就是XML序言, 这一行代码会告诉解析器或浏览器这个文件应该按照XML规则进行解析。

在XML序言后面的是XML的根元素。在上面的例子中, 此元素就是, 而是根元素的结尾标记(即结束标签)。任何XML文档都只能有一个根元素。根元素里面, 可以有若干个子元素, 子元素里面可以有子子元素, 以此类推。如, 上例中的便是子元素, 共有两个, 代表两件产品。而每一个下又有四个子元素, 分别是产品名称, 产品编号, 产品类型和用于标识产品的唯一编号。

XML是一种“元”语言, 它定义了一套标准, 而世界上任何个人和组织均可以基于这套标准建立适合自己使用的标记语言。如, 用于地理信息系统的地理标记语言GML (Geography Markup Language), 用于金融行业的金融产品标记语言FpML (Financial Products Markup Language), 在开展资源环境审计或金融审计中均有可能遇见, 且对它们的处理都遵循统一的标准。

二、SQL Server数据库对XML的支持

SQL Server 对XML的支持源自2005版。用户可以创建这样的表, 它在关系列之外还有一个或多个 XML 类型的列; 此外, 还允许带有变量和参数。为了更好地支持 XML 模型特征(例如文档顺序和递归结构), XML 值以内部格式存储为大型二进制对象 (BLOB)。

用户将一个XML数据存入数据库的时候, 可以使用这个XML的字符串, SQL Server会自动的将这个字符串转化为XML类型, 并存储到数据库中。相应的, T-SQL语句也提供了大量对XML操作的功能来配合SQL Server中XML字段的使用。

(一) 定义XML字段并插入XML文件。

XML字段的定义即是在创建数据表时, 对要用于存储XML数据的列, 指定其数据格式为XML。如, 要创建一个存放家电下乡空调补贴明细的

- 政府信息公开指南
- 政府信息公开目录
- 政府信息公开申请
- 政府信息公开年报

审计资讯

- 近日要闻
- 要闻回顾
- 特派办动态
- 审计结果公告
- 国际交流
- 综合论坛
- 审计播报
- 图片新闻
- 机关动态
- 地方动态
- 审计工作通告
- 案件披露
- 理论与方法

网上服务

- 考试信息
- 资料下载
- 审计知识
- 网站地图
- 报刊订阅
- 教育培训
- 成绩查询
- 公共信息
- 历史上的今天
- 投稿信箱及投稿须知

互动交流

- 信访举报
- 图文直播
- 意见征集
- 投稿信箱
- 地方在线直播
- 在线访谈
- 网上调查
- 审计长信箱
- 地方在线访谈

表, 包含两个字段, 一是存放空调型号的字段 二是存放该型号空调享受家电下乡补贴明细XML字段的字段。SQL语句如下:

```
CREATE TABLE 家电下乡空调补贴明细
(
  型号 NVARCHAR(255) PRIMARY KEY,
  明细 XML NOT NULL
)
```

创建完表之后, 即可插入数据。如, 型号为“KF—25U2—B”的空调, 其明细存放在“KF—25U2—B.xml”文件中, SQL语句如下:

```
DECLARE @doc XML;
SELECT @doc=BULKCOLUMN FROM OPENROWSET(
  BULK N'D:/KF-25U2-B.xml',
  SINGLE_BLOB) AS x

INSERT INTO 家电下乡空调
VALUES ('KF-25U2-B', @doc)
```

运行结果如下:

(二) SQL Server对XML字段的查询操作。

在定义了一个XML类型的数据之后, 我们最常用的就是查询操作, 下面我们来介绍如何使用SQL语句来进行查询操作。

在T—SQL中, 提供了两个对XML类型数据进行查询的函数, 分别是query(xquery)和value(xquery, dataType)。其中, query(xquery)得到的是带有标签的数据, 而value(xquery, dataType)得到的则是标签的内容。下面, 我们分别使用这两个函数来进行查询。

1. 使用query(xquery)查询。

该函数通过传递某一元素的“XPath路径”为参数, 查找该路径下的元素。如, 我们需要获取上例中的第一个产品()的唯一编号, 可以发现其是根元素下的子元素。因此, 其路径为“(/ROOT/PRODUCT/PD_SERIAL)[1]”, “[1]”代表第一个产品。查询语句为:

```
DECLARE @doc xml;
DECLARE @DocHandle int
SET @doc=(SELECT 明细 FROM 家电下乡空调)
SELECT @doc.query('/ROOT/PRODUCT/PD_SERIAL)[1]')
```

运行结果如下图:

2. 使用value(xquery, dataType)查询。

该函数通过传递某一元素的“XPath路径”和要返回值的数据类型为参数, 查找该路径下的元素的值。如, 在上例中, 要查找第一个产品()的唯一编号的值, 其数据类型是字符串型。查询语句如下:

```
DECLARE @doc xml;
DECLARE @DocHandle int
SET @doc=(SELECT 明细 FROM 家电下乡空调)
SELECT @doc.value('/ROOT/PRODUCT/PD_SERIAL)[1]', 'NVARCHAR(MAX)')
```

运行结果如下图:

使用上述两种方法能够较快地定位XML中某一个元素的值, 但是在实际运用过程中, 如果需要大量获取数据则显得比较吃力。如, 某型号空调家电下乡补贴明细中含有300产品信息, 即元素下有300个子元素, 此时如果仍一个个地查询出产品的编码, 效率是非常低的。

对于这种情况, SQL Server数据库提供了一种将XML文件转换成关系型数据表的方法。

(三) 使用OPENXML将XML转换成关系型数据表。

OPENXML通过XML文档提供行集视图。由于OPENXML是行集提供程序, 因此可在会出进行行集提供程序(如表、视图或OPENROWSET函数)的Transact-SQL语句中使用OPENXML。其语法表示如下:

```
OPENXML( idoc int [ in ], rowpattern nvarchar [ in ], [ flags byte [ in ] ] )
[ WITH (SchemaDeclaration) ]
```

参数说明:

idoc: XML文档的内部表式形式的文档句柄。通过调用 sp_xml_preparedocument创建XML文档的内部表式形式。

rowpattern: XPath模式, 用来标识要作为行处理的节点(这些节点在XML文档中, 该文档的句柄由idoc参数传递)。

flags: 指示应在XML数据和关系行集间使用映射以及如何填充溢出列。flags为可选输入参数, 可以是下列值之一。

字节值	说明
0	默认为“以属性为中心”的映射。
1	使用“以属性为中心”的映射。可以与 XML_ELEMENTS 一起使用。这种情况下, 首先应用“以属性为中心”的映射, 然后对所有未处理的列应用“以元素为中心”的映射。
	使用“以元素为中心”的映射。可以与 XML_ATTRIBUTES 一起使

2	用。这种情况下，首先应用“以属性为中心”的映射，然后对所有未处理的列应用“以元素为中心”的映射。
8	可与 XML_ATTRIBUTES 或 XML_ELEMENTS 组合使用（逻辑或）。在检索的上下文中，该标志指示不应将已使用的数据复制到溢出属性 @mp:xmltext。

SchemaDeclaration: 用来描述要转成的关系型数据表的表结构声明。

如，要将上述空调家电下乡推广明细的XML文件转换成关系型数据表，其语句如下：

```
DECLARE @doc xml;
DECLARE @DocHandle int
DECLARE @model nvarchar(255)
SET @model='KF-25U2-B'
SET @doc=(SELECT 明细 FROM 家电下乡空调 WHERE 型号=@model)
EXEC sp_xml_preparedocument @DocHandle OUTPUT, @doc
SELECT @model AS 型号, *
FROM OPENXML (@DocHandle, 'ROOT/PRODUCT',2)
WITH (PD_NAME nvarchar(max),
      PD_ID nvarchar(max),
      PD_TYPE nvarchar(max),
      PD_SERIAL nvarchar(max))
EXEC sp_xml_removedocument @DocHandle
```

运行结果如下图：