基于XML的信息存储与检索

沈 艺 (南京师范大学图书馆计算机室 210097)

摘要：本文在分析XML (eXtensible Markup Language，可扩展标记语言) 组成特点的基础上，以SQL Server 2000应用为例，讨论了XML在信息存储和检索应用中的具体实现。

关键词：XML SQL 信息存储 情报检索 半结构化数据

1 引言

XML (eXtensible Markup Language，可扩展标记语言) 作为SGML (Standard Generalized Markup Language，标准通用标记语言) 的子集被W3C (World Wide Web Consortium) 认可，并成为一个推荐标准。XML具有可扩展性、简单性、开放性、互操作性及支持多国语言等特性，避免了HTML (Hypertext Markup Language，超文本标记语言) 的局限性，成为网络环境下结构化信息描述和管理的有效工具。

利用成熟数据库技术对纷繁繁杂的互联网信息进行有效的存储和检索，是当今网络和数据库领域共同关注的问题。为了解决这个问题，许多著名的数据库管理系统都力求将XML作为一个面向电子信息资源管理的标准化架构，提供对XML的集成，实现基于XML文档的定义、存储和检索。本文以SQL Server 2000为例说明如何将XML格式的文档存储在数据库中，以及在浏览器中利用XML模板访问数据库。

2 XML与数据库

XML是一种元语言，它包含一组基本规则。利用这些规则可以创建特定的标记语言，这些标记不是描述信息的显示方式，而是描述信息本身。XML包含三个要素：

(1) DTD (Document Type Definition，文档类型定义) 或XML Schema (XML大纲)，它们定义了XML文件的元素、元素属性以及元素和元素属性之间的关系，实现统一的XML数据表示以及数据的相互集成；

(2) XSL (eXtensible Stylesheet Language，可扩展样式语言) 用于规定XML文档呈现的形式，说明各个标记的显示方式，实现XML文档的转换和格式化，使得数据与其表现形式相互独立；

(3) XLL(eXtensible link Language，可扩展链接语言) 将进一步扩展目前Web上已有的简单链接，实现多方向链接，且链接可以存在于对象层，而不仅仅是页面。

XML支持结构化数据，可以详细地定义某个数据对象的数据结构。例如为了描述一本书，可以定义书的作者、标题、ISBN、出版社等。这种XML数据容易按作者、标题等排序，查询也很方便。XML可以从不同的来源集成或组合数据，也可以将多个应用程序生成的数据纳入同一个XML文件。互联网中数据的结构和数据元素的类型会不断改变，新的数据类型也会不断产生。由于数据模式经常改变，互联网中数据很难采用单一模式进行管理，互联网数据实际上是一种半结构化数据。半结构化数据的特点是数据表示形式不规则，不符合某一固定的格式。因此，对传统的数据库来说，半结构化数据是难以直接进行管理的，而借助XML，也可以有效实现半结构化信息的处理。如果信息以XML形式提供，则数据结构和数据内容都是可分析的，可以进行功能强大的查询。

以数据库观点，XML文档可看作数据库，它的DTD看作是数据库的模式。数据库的模式描述了数据库结构，也就是数据库管理的数据实体的类型、特征和实体间的联系；DTD描述了XML文档的结构，定义了所允许的元素类型、属性和实体，并表述它们组合方式的约束条件。但数据库与XML文档是有区别的，数据库的数据结构性很强，而XML更适合描述半结构化数据。XML可以描述扩展的关系模型和面向对象的数据模型，因此，关系数据库中的数

据可以在不丢失原始语义的情况下转化为XML文档，利用关系数据库存储XML文档，为了消除半结构化数据与二维数据之间的差别，可能会丢失半结构化数据的部分信息，这是我们在实际应用中必须注意的问题。

3 XML文档存储

如果信息是XML形式，则数据结构和数据内容都是可分析的。从安全和效率考虑，从互联网上获得的数据一般不直接存入数据库中，而先存储在XML文档中，然后再从XML文档中提取数据信息存入到关系数据库中。在SQL Server 2000中OPENXML提供了在关系数据库表中存储XML文档的功能，其语法格式为：

OPENXML (idoc int [in],rowpattern nverchar [in],[flags byte [in]] [WITH(SchemaDeclaration | TableName)]

参数idoc为XML文档内部映射的文件指针；rowpattern表示Xpath模式，用于确定哪一节点将被处理成关系表的行；flags指明XML文档与关系表数据行之间的匹配关系；SchemaDeclaration为表单的模式定义；TableName给定表名。SQL Server 2000已建有数据表"图书"，定义为：图书=(书名，丛书名，类别，ISBN，价格，出版者，责任者，开本，页数，出版日期)。下面给出一个例子，使用OPENXML将描述二种图书的XML文档转换成关系表。

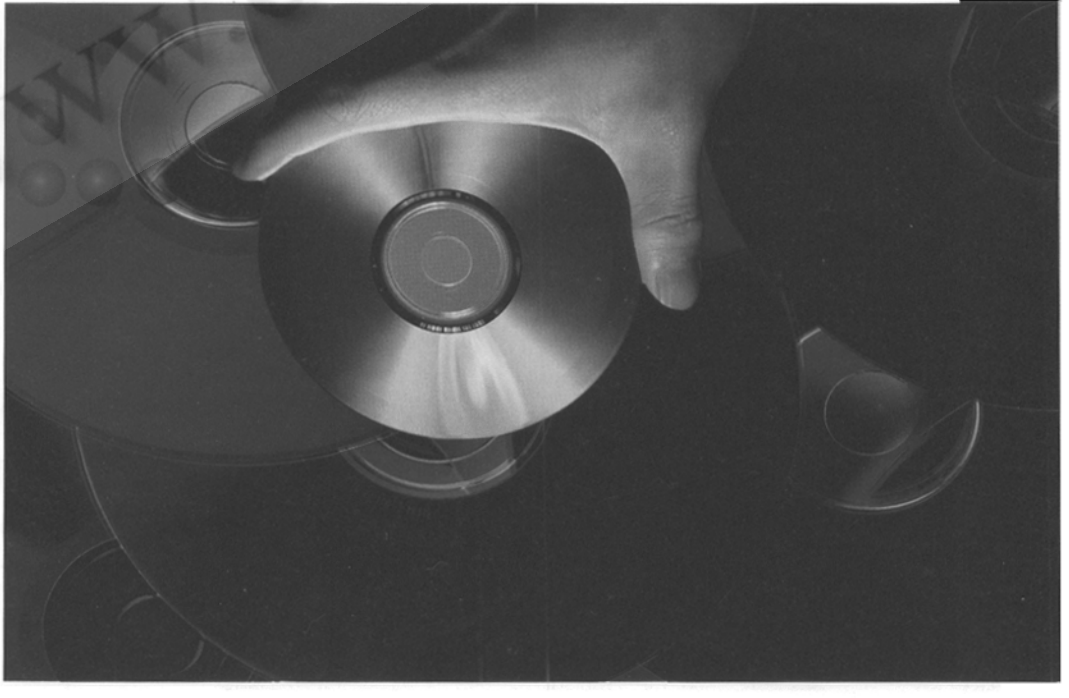
```
Declare @idoc int
Declare @doc varchar(1000)
Set @doc='
<root>
<图书 书名="迈向新的国际金融体系：亚洲金融危机后的思考" 丛书名="国际经济热点译丛" 类别="F" isbn="7-200-04007-X" 价格="12.00" 出版者="北京出版社" 责任者="易臣格瑞" 开本="21" 页数="200" 出版日期="2000-01-01" />
<图书 书名="21世纪教师与父母必读" 丛书名="21世纪教育的四大支柱丛书" 类别="G" isbn="7-200-03778-8" 价格="15.00" 出版者="北京出版社" 责任者="孙云晓" 开本="21" 页数="336" 出版日期="2000-01-01" />
</root>
exec sp-xml-preparedocument @idoc output, @doc
select *
from openxml (@idoc,root/图书 2)
with (书名 varchar(100) '@书名',
      丛书名 varchar(100) '@丛书名',
      类别 char(10) '@类别',
      isbn char(13) '@isbn',
      价格 numerical(9) '@价格',
      出版者 varchar(50) '@出版者',
      责任者 varchar(50) '@责任者',
      开本 char(6) '@开本',
      页数 numerical(5) '@页数',
      出版日期 datetime '@出版日期')
```

4 信息检索

SQL Server 2000允许在URL中使用SQL语句向SQL Server 2000提交查询，并以XML文档形式返回查询结果。例如在浏览器地址栏中输入http://202.119.108.116/xml?sql=SELECT+*+FROM+图书+WHERE+责任者='孙云晓'+FOR+XML+AUTO&root=root，则返回如下结果：

```
<?xml version="1.0" encoding='utf-8'?>
-<root>
  <图书 书名="21世纪教师与父母必读" 丛书名="21世纪教育的四大支柱丛书" 类别="G" isbn="7-200-03778-8" 价格="15.00" 出版者="北京出版社" 责任者="孙云晓" 开本="21" 页数="336" 出版日期="2000-01-01" />
</root>
```

虽然在浏览器中直接执行SQL语句简单快捷，但出于对数据库系统安全的考虑，这种方式在大部分互联网环境中是不适用的，因为最终用户通过浏览器直接对数据库进行insert，update甚至是delete操作，会给数据库安全带来极大的威胁。为此SQL Server 2000提供了XML模板功能，通过URL来访问指定服务器端XML模板，把SQL语句或Xpath查询请求隐藏在XML模板中。这也就提供了一种安全快捷的信息发布方法。下面的代码是一个通过XML模板发布书目



信息的例子，它们由两个文档组成，一个是查询模板book.xml，另一个是决定book.xml显示式样的book.xsl文档。

Book.xml文档内容：

```
<? xml version=′ 1.0′ encoding= ″GB2312″? >
<book xmlns: sql= ′urn: schemas-microsoft-com: xml-sql′
sql: xsl= ′book.xsl′ >
<sql: query>
SELECT书名, 责任者FROM 图书 FOR XML AUTO
</sql: query>
</book>
```

book.xsl文档内容：

```
<?xml version= ′1.0′ encoding= ″GB2312″? >
<xsl: stylesheet xmlns:xsl= ′http://www.w3.org/1999/XSL/Transform′ >
<xsl: template match= ′/′
<xsl: apply-templates />
</xsl: template>
<xsl: template match=′ book′ >
<tr>
<td><xsl: value-of select = ′@ 书名′ /></td>
<td><b><xsl: value-of select= ′@ 责任者′ /></b></td>
</tr>
</xsl: template>f<xsl:template match= ′/′ >
<html>
<body>
<table board= ′2′ style=′ with: 200;′ >
<tr><th>书目信息</th></tr>
<tr><th>书名</th><th>作者</th></tr>
<xsl: apply-templates select=′ 图书′ />
</table>
</body>
</html>
</xsl: template >
</xsl: stylesheet>
```

在浏览器地址栏里输入http://IISSERVER/xml/template/book.xml?contenttype=text/html，就可以将所有的书目信息简单地列出来。

5 结束语

XML文档的有效存储和检索，是互联网资源有效管理的关键。随着信息技术的发展，会有越来越多的数据库管理系统支持XML应用。XML作为一种半结构化数据模型，与数据库结合起来，实现更为精确的信息整理和获取。随着对XML研究的不断深入，新的成果不断出现，XML必将会被广泛地接受，也必将会对互联网应用，尤其是数字图书馆应用起到非常重要的作用。

参考文献

1 http://www.w3.org/xml
2 http://www.w3.org/1999/XSL/Transform
3 http://www.sqlmag.com/
4 Rick Jelliffe著，XML&SGML参考手册，人民邮电出版社，2000.10。