

基于商业 RDBMS 的 XML 数据交换技术

许卓明, 陶 皖, 廖述梅

(河海大学计算机及信息工程学院, 江苏 南京 210098)

摘要:介绍了三大商业关系数据库管理系统(RDBMS)Microsoft SQL Server, Oracle 和 IBM DB2 UDB 对基于 Internet 的应用间以 XML 格式交换 SQL 数据的最新技术支持;着重于如何将 XML 文档存储于 SQL 数据库中以及如何将 SQL 数据以 XML 文档发布的问题,比较了三大商业 RDBMS 的技术手段和特色,最后,对该技术进行了简要的总结与展望。

关键词:XML;数据管理;数据交换;RDBMS;SQL Server;Oracle;DB2

中图分类号:TP311.13 **文献标识码:**A **文章编号:**1000-198X(2004)01-0108-05

XML 的可扩展性、自描述性、内容与表示的分离性,以及支持模式校验等特点,使它成为表示各种数据的理想公共格式,尤其是 Internet 应用间数据交换的标准格式。但是,商业 RDBMS 在数据存储、索引、查询优化、事务管理、安全与恢复等方面提供了完善的高性能服务设施,因此,企业应用大多基于 RDBMS,大量业务数据已经并将继续以(对象-关系)格式存储于 SQL 数据库中。这样,就产生了在 Internet 应用间基于 RDBMS、以 XML 格式交换业务数据的需求。商业 RDBMS 都进行了相应的 XML 扩充,为这种数据交换提供了有效支持^[1~3]。本文着重于 XML 数据交换中,如何用 SQL 数据库存储 XML 文档、如何将 SQL 数据以 XML 格式发布等问题,介绍、比较和总结三大商业 RDBMS(Microsoft SQL Server, Oracle, IBM DB2 UDB)的最新技术支持。

1 商业 RDBMS 的 XML 数据交换技术概述

由于 XML 数据和(对象-关系)数据采用不同的数据模型,致使两者在数据模式和实例层上均存在异构性^[4],因此,为实现 XML 数据交换,RDBMS 必须进行扩充,即提供相应的数据扩充器(data extender)来实现 SQL 和 XML 之间的模式映射定义和数据实例转换。一般而言,有两种实现数据扩充器的策略^[5]:(a)关系/中间件(relational/middleware)方法——数据扩充器置于数据库的外部,典型的有 Microsoft SQL Server 和 Sybase;(b)通用服务器(universal server)方法——数据扩充器置于数据库的内部,典型的有 Oracle 和 IBM DB2。尽管早期各 RDBMS 大多以“中间件”方式支持 XML,但不同于 Microsoft SQL Server, Oracle 和 IBM DB2 UDB 已进行了充分的面向对象扩充(几乎完全支持 SQL:1999 标准)而成为对象-关系数据库系统,这为它们以“通用服务器”策略实现更自然的“本原”(native)XML 支持奠定了基础;而 Microsoft SQL Server 更侧重于保持底层的“纯正”关系模型而采用“关系/中间件”策略来支持 XML。早在 SQL/XML 标准(即 ANSI SQLX 工作组即将完成的 SQL 新部分)^[6,7]制定工作启动之前,这些 RDBMS 均开始扩充其产品^[8],以支持 XML 的存储和发布。

Microsoft 通过称为 SQLXML 的 XML 特性包(即 XML for SQL Server)来扩充 SQL Server 2000,使其成为 SQL Server 2000 Web Release^[1,9],以提供对 XML 的支持。从基于 SQLXML 1.0 的、提供基本 XML 支持的 SQL Server 2000 Web R1,到当前基于 SQLXML 3.0^[10]的 SQL Server 2000 Web R2,不断扩充对 XML 的支持。Microsoft SQL Server 通过扩展其 Transact-SQL 语句来提供(缺省或定制的)SQL-XML 映射定义和数据转换机制。

Oracle8^[2]是 Oracle 的第 1 个 XML-enabled 产品,它以 XML-SQL Utility(XSU)提供规范的(canonical)XML-SQL 映射和(很弱的)定制映射定义,以及数据转换机制。Oracle9i R1^[11]增强了 XSU 的功能,增加了新的函数,以更好地支持 XML 数据交换,率先引入了“本原”XMLType 数据类型,可实现 XML 文档的直接存储。2002 年上半年发布的 Oracle9i R2^[12]进一步提供了所谓的 XML DB 技术,其关键特性包括:XMLType 及相关的 Native

FOR XML 语法^[1]为:FOR XML(raw | auto [,elements] | explicit) [,xmldata].其中:elements 指示将非 NULL 列值映射为与列(别)名同名的子元素值;xmldata 指示在 XML 结果中,应包含一个内联模式(即用 XDR 模式语言描述的结构和数据类型信息).

例2 用 FOR XML 将从基表 Customers 和 Orders 中查询到的订货用户信息以 XML 形式发布的脚本如图3所示,可能的结果为:< Row CustNo = "201 "ItemNo = "703 " OrderDate = "2002-12-10 "> .

3 Oracle9i R2 对 XML 数据交换的支持

3.1 XML 存储

Oracle9i R2^[12]支持 XML Schema 及 XML 文档基于 XML Schema 的全局(针对整个数据库)或局部(针对某个特定用户)约束与校验.Oracle9i R2 的 XML DB 中的 XMLType 提供了 XML 的“本原”存储,它有下列存储形式选择:(a)CLOB 形式(非结构化存储,可原样保持 XML 文档的完整性);(b)对象-关系表形式(本原的结构化存储,将一个 XML 文档拆分成底层 SQL:1999 风格的对象-关系结构,且保持文档的 DOM 保真度,即:存储的 DOM 与检索出的 DOM 完全一致);(c)混合存储(以上两种形式的混合).XMLType 存储模式的选择对处理 XMLType 提取的应用开发者是完全透明的.

例3 设图1的 XML 文档用图4的 XML Schema 文件 OrdSchema.xsd 来约束,用 CREATE TABLE 创建基表 Orders(OrderID ,OrderDate ,CustNo ,CustName)来存储图1的 XML 文档数据的相应脚本如图5所示.

3.2 XML 发布

XML DB 主要通过一组内置的 SQL/XML 风格函数来实现 XML 发布^[12].借助 SYS_XMLGEN() 函数,可有效地从 SQL 结果中生成一个有效的 XML 文档(XMLType 类对象).SYS_XMLGEN() 能从一个变量或 SQL Object 类型中生成 XML 文档,若是一个变量则生成一个只有单个根元素的简单文档,若是一个 SQL Object 类型则产生每个根元素都可以有一个子元素(子元素反映了 Object 类型所定义的属性)的复杂文档.SYS_XMLAGG() 函数可将 SYS_XMLGEN() 生成的一系列文档组合为一个文档.

例4 用 SYS_XMLGEN() 函数从基表 Orders (OrderID , CustNo , OrderName , OrderPrice) 和 Customers (CustNo ,CustName ,CustAdd)中提取数据生成 XML 文档的相应脚本见图6.

4 DB2 UDB V8 对 XML 数据交换的支持

4.1 XML 存储

IBM DB2 UDB 的 XML Extender 提供两种可选的 XML 关系存储方案:(a)XML Column:一个 XML 文档完整地作

```
SELECT Customers ,CustNo ,Orders ,ItemNo ,OrderDate
FROM Customers ,Orders
WHERE Customers . CustNo = Orders . CustNo
ORDER BY Customers .custNo
FOR XML raw ;
```

图3 用 FORXML 实现 XML 发布
Fig.3 XML publication with FORXML

```
<xs:element name="OrderList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Order">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OrderDate" type="xs:date">
              xdb:SQLName="OrderDate"
              xdb:SQLType="date"/>
            <xs:element name="Customer">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="CustNo">
                    type="xs:integer"
                    xdb:SQLName="CustNo"
                    xdb:SQLType="number"/>
                  ...
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute name="OrderNo" type="xs:integer">
    xdb:SQLName="OrderID"
    xdb:SQLType="number"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
```

图4 XML Schema 文件 OrdSchema.xsd 片段
Fig.4 OrdSchema.xsd fragment of XML Schema file

```
CREATE TABLE Orders( info XMLTYPE )
XMLSCHEMA 'http://www.oracle.com/xdb/OrdSchema.xsd'
ELEMENT 'OrderList' ;
```

图5 用 CREATE TABLE 创建基表存储 XML 数据
Fig.5 Creating a table to store XML data with
CREATE TABLE

```
CREATE TYPE CustomerOrder AS Object( "CustName"
  Varchar2(16); OrderID Number(6), OrderName Varchar2(14),
  "CustNo" Number(6), OrderPrice Number(10,2));
//产生生成 XML 文档所需的 Object 类型
CREATE VIEW OrderNameView AS
  SELECT CustName SYS_XMLGEN( Customer( CustName ,
  SELECT Order( OrderID ,OrderName ,CustNo ,OrderPrice )
  FROM Orders Y WHERE X.CustNo = Y.CustNo )
  As CustomerOrder )
  FROM Customers X ;
//利用 SYS_XMLGEN() 产生 XMLType 类型的实例
```

图6 用 SYS_XMLGEN() 提取基表数据生成 XML 文档
Fig.6 Abstracting data from tables to produce an
XML document with SYS_XMLGEN()

```
< DAD >
< dtidid > OrderList < /dtidid >
< Xcollection >
< prolog > ? xml version = "1.0" ? < /prolog >
< doctype > ! DOCTYPE OrderList OrderList.DTD < /doctype >
< root_node >
  < element_node name = "OrderList ">
    < RDB_node >
      < table name = "Order" key = "OrderNo" />
      < table name = "ItemList" key = "OrderNo" />
      < condition > Order . OrderNo = ItemList . OrderNo
      < /condition >
    < /RDB_node >
    < element_node name = "Order" multi_occurrence = "yes">
      < attribute_node name = "OrderNo">
        < RDB_node >
          < table Name = "Order" />
          < column name = "OrderNo" type = "varchar( 10 )" />
        < /RDB_node >
      < /attribute_node >
      < element_node name = "ItemList">
        < element_node name = "item" multi_currence = "yes">
          < element_node name = "itemNo">
            < text_node >
              < RDB_node >
                < table name = "ItemList" />
                < column name = "itemNo"
                  type = "varchar( 10 )" />
              < /RDB_node >
            < /text_node >
          < /element_node >
        < /element_node >
      ...
    < /element_node > < ! --The end of itemList-->
  < /element_node > < ! --The end of Order-->
< /element_node > < ! --The end of OrderList-->
< /root_node >
< /Xcollection >
< /DAD >
```

图 7 使用 RDB_node 模式的 DAD 文件(片断)
Fig.7 DAD file(fragment)in RDB_node mode

5 总结与展望

Microsoft SQL Server 2000 以扩展 SQL 语句为主要策略,使用户充分利用已有的 SQL 知识,以最小编程量实现应用间以 XML 为中介格式的数据交换;Oracle9i R2 则以“本原化”为主要特色,提供独特的 XML DB 技术,这更有利于支持用户采用标准的 XML 技术来开发 Internet 应用;IBM DB2 UDB V8 虽因尚不支持 XML Schema 受到广泛批评,但其在引入 SQL/XML 风格函数方面却走在前头,这向支持 SQL/XML 标准^[6]迈出了坚实的一步.从功能角度来说,尽管各 RDBMS 对 XML 数据交换的实现技术各异,但它们均能满足 XML 数据交换的需求.从性能角度来说,目前难以有一个统一的评价结论,因为客观地评价各家的性能“好坏”应在一个广泛接受的公共基准^[15]下进行.此外,由于 XML 只提

为一个用户定义类型(XMLCLOB,XMLVARCHAR 或 XMLDBCLOB)的列值存储于 DB2 表中或以 XMLFile 类型存储在操作系统的文件系统中,此时 DAD 用于定义哪些 XML 元素/属性应被索引;(b)XML Collection:一个 XML 文档被拆分为结构化数据存储到由若干 DB2 表组成的一个或多个 XML Collection 中,此时 DAD 用于定义 XML 元素/属性与 DB2 表之间的映射关系.定义了 DAD 后,通过用户自定义函数将 XML 数据存储于数据库中.

例 5 用图 7 的 DAD 文件(片断)定义一个 XML Collection,把图 1 的 XML 文档存储于两个 DB2 表 Order(OrderNo ,CustNo ,CustName ,CustAdd)和 ItemList (OderNo ,ItemNo ,ItemName ,ItemPrice)中.

4.2 XML 发布

首先要定义 DAD 文件,然后用 XML Extender 的存储过程 dxGenXML() 和 dxRetrieveXML() 实现将 SQL 数据以 XML 文档发布^[3].此外,IBM DB2 UDB V8 的内置 SQL/XML 函数^[14]如:XMLEMENT(),XMLATTRIBUTES() 分别用于建立 XML 元素和属性),XMLAGG() 将一组元素组合成一个元素森林)等,使用户用熟悉的 SQL 操作发布 XML 文档^[14].

例 6 用函数 XMLEMENT(),XMLATTRIBUTES() 和 XMLAGG() 将基表 Orders (OrderNo , CustNo , CustName ,OrderDate)中的数据进行 XML 发布,相应脚本见图 8,可能的结果见图 9.

```
SELECT XML2CLOB( XMLEMENT( NAME" OrderLIST ",
XMLAGG( XMLEMENT( NAME" Order ",
XMLATTRIBUTES( OrderNo ),XMLEMENT( NAME" OrderDate " ),
XMLAGG( XMLEMENT( NAME" Customer ",
XMLEMENT( NAME" CustNo " ),
XMLEMENT( NAME" CustName "))))))
FROM Orders ;
```

图 8 用内置 SQL/XML 函数实现基表数据的 XML 发布
Fig.8 Publishing table data in XML with built-in SQL/XML functions

```
< OrderList >
< Order OrderNo = '1001' >
  < OrderDate > 2003-07-11 < /OrderDate >
  < Customer >
    < CustNo > 301 < /CustNo > < CustName > Mark < /CustName >
  < /Customer >
< /Order >
< /OrderList >
```

图 9 结果 XML 文档实例
Fig.9 Example of generated XML document

供语法层的互操作性,因此,XML数据交换只有在交换双方共同遵守公共 DTD 或 XML Schema 模式定义的前提下才能“不失真”地进行。

标准化组织和 RDBMS 厂商正在共同制定的 SQL/XML 标准,旨在规范和统一 SQL 数据库与 XML 之间的映射关系和数据转换机制。这有利于向用户提供规范、统一的 XML 数据交换技术,有利于提高应用的可移植性和(跨平台的)互操作性,有利于开发一个真正广泛接受的产品比较评价基准。可以预计,RDBMS 厂商在支持 SQL/XML 标准方面将一如既往地展开激烈竞争,但这种竞争的最大赢家将是广大的用户。

参考文献:

- [1] MICHAEL R. State-of-the-art XML support in RDBMS: Microsoft SQL Server's XML features[J]. IEEE Data Engineering Bulletin, 2001, 24(2): 3—11.
- [2] BANERJEE S, KRISHNAMURTHY V, KRISHNAPRASAD M, *et al.* Oracle8i—the XML enabled data management system[A]. In: Proc of the 16th Int'l. Conf. on Data Engineering (ICDE 2000) [C]. New York: IEEE Computer Society Press, 2000. 561—568.
- [3] CHENG Josephine, XU Jane. IBM DB2 XML Extender: an end-to-end solution for storing and retrieving XML documents[EB/OL]. <http://www-4.ibm.com/software/data/db2/extenders/xmlxt/xmlxtbroch.pdf>, 2000-02/2003-08-15.
- [4] 许卓明,刘琴,董逸生. 基于关系数据库的 XML 存储技术评述[J]. 计算机工程与应用, 2003, 39(21): 197—200.
- [5] PHIL H. Databases: an evaluation & comparison[M]. Buckinghamshire, UK: Bloor Research Ltd, 2001. 27—32.
- [6] EISENBERG A, MELTON J. SQL/XML is making good progress[J]. ACM SIGMOD Record, 2002, 31(2): 89—95.
- [7] EISENBERG A, MELTON J. SQL/XML and the SQLX informal group of companies[J]. ACM SIGMOD Record, 2001, 30(3): 105—108.
- [8] 刘琴,许卓明,刘传汉. 基于 XML 的商业 DBMS 技术评述[J]. 计算机工程与设计, 2002, 23(11): 24—28.
- [9] HOWLETT S, JENNINGS D. SQL Server 2000 and XML: developing XML-enabled data solutions for the Web[EB/OL]. <http://msdn.microsoft.com/msdnmag/issues/02/01/SQLXML/default.aspx>, 2002-01/2003-08-15.
- [10] THILMANY C. SQLXML 3.0: build data-driven Web Services with updated XML support for SQL Server 2000[EB/OL]. <http://msdn.microsoft.com/msdnmag/issues/02/05/SQLXML3/default.aspx>, 2002-05/2003-08-15.
- [11] SCARDINA M V. XML Support in Oracle9i[EB/OL]. <http://otn.oracle.com/tech/xml/pdf/Oracle9iXML1.pdf>, 2000-12/2003-08-15.
- [12] BANERJEE S. Oracle XML DB technical white paper release 9.2[EB/OL]. http://otn.oracle.com/tech/xml/xmlldb/pdf/XMLDB_Technical_Whitepaper.pdf, 2002-01/2003-08-15.
- [13] IBM Corporation. IBM DB2 Universal Database version 8.1: what's new[EB/OL]. <http://www-3.ibm.com/software/data/db2/udb/pdfs/db2q0.pdf>, 2002/2003-08-15.
- [14] IBM Corporation. Efficient ways to publish DB2 data as XML: using the new SQL/XML features in DB2 V8[EB/OL]. http://www6.software.ibm.com/reg/devworks/dw-db2sqlx-i?S_TACT=103AMW68&S_CMP=dwjs, 2002/2003-08-15.
- [15] SCHMIDT A, WAAS F, KERSTEN M, *et al.* Why and how to benchmark XML databases[J]. ACM SIGMOD Record, 2001, 30(3): 27—32.

Commercial RDBMS-based data exchange in XML-form

XU Zhuo-ming, TAO Wan, LIAO Shu-mei

(College of Computer and Information Engineering, Hohai Univ., Nanjing 210098, China)

Abstract: An introduction is given to the updated XML supports, provided by three major commercial relational database management systems (RDBMSs), i.e. Microsoft SQL Server, Oracle, and IBM DB2, for SQL data exchange in XML-form through the Internet. Comparisons focusing on the issues of how to store XML documents in SQL databases and how to publish SQL data as XML documents are made from the angles of technical means and features of the RDBMSs. Finally, a conclusion on the technology and the trend of its future development are presented in brief.

Key words: XML; data management; data exchange; RDBMS; SQL Server; Oracle; DB2