基于数据库的 XML 存储技术设计和实现

Design and Realize For The Storage Technique Of XML Based Database

冯启蒙 （西安工业大学 材料与化学学院 710032）
王振辉 （西京学院 经济系 陕西 西安 710123）
王振铎 （西京学院 经济系 陕西 西安 710123）

摘要：首先对企业由数据集成到数据挖掘的新需求进行了分析。在此基础上，提出了采用 XML 与数据库技术来简化数据仓库的建立和操作。重点探讨了 XML 在数据库中存储技术的实现。然后结合 Oracle 9i 和企业客户数据文件，给出了 XML 在数据库中存取的关键技术和主要代码，最后对数据库存储 XML 文件在数据处理和数据检索等方面的应用优势进行了总结。

关键词：数据仓库 数据库 XML 存储技术

1 问题的提出

近年来，一些信息化应用较早的企业为了整合内部的异构数据源，解决"信息孤岛"问题，采用了数据集成技术，建立多种不同体系结构的综合信息服务系统，以满足数据共享和全局应用的开展。其中，多以 XML 技术作为数据交换和传输的标准，在应用便利的同时也产生了大量的 XML 文档。随着数据量的增加，XML 文档的管理和 XML 作为数据存储的缺陷－查询性能低下的特点暴露无疑，并且由于文件系统的约束，文件大小和并发性都不能满足用户的要求。数据集成主要解决了全局检索的问题，随着企业对主题数据的分析能力的提升，企业对数据集成平台提出了更高的要求。不仅要解决数据共享的问题，同时又要满足知识发现的要求，帮助企业决策者发现数据间的关联和变化趋势，这样就需要对基于 XML 的数据集成系统进行升级改造以适应目前新的需求。

数据库技术有着严格的理论基础，丰富的查询语言和广泛的应用。同时，能够保证数据的完整性，安全性和访问并发性的要求。知识发现也就是在海量数据中进行数据的挖掘，而数据挖掘的基础是利用数据库技术建立和管理数据仓库。

数据仓库的数据来自分布在企业内部的多种异构数据源中，通过 XML 集成这些异构数据后，为数据仓库提供了一个统一格式的数据，简化了数据仓库获取数据和转化数据的过程。同时数据库技术的应用，保证了数据检索的效率，同时通过 DBMS 软件统一管理数据，使得数据的完整性和安全性得到了保障。一个好的数据仓库环境是数据挖掘的催化剂，这两种技术相辅相成，必将大大提升企业数据的挖掘效率。图 1 为基于 XML 和数据库技术建立数据仓库的流程。

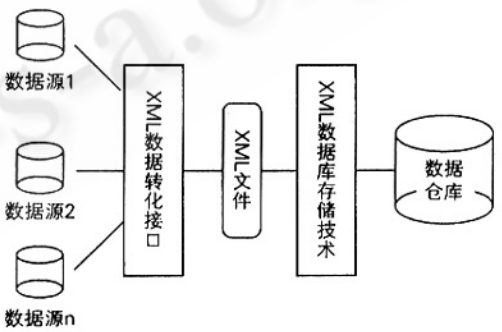

图 1 数据仓库数据采集流程图

在图 1 中，多个分布在不同物理位置的异构数据源通过 XML 数据转化接口，生成统一格式的 XML 数据文件。经过 XML 格式统一后的数据，简化了数据仓库直接从数据源经过抽取、转换、清洁数据的过程。而 XML 与数据库技术结合的基础就是如何用数据库技术来存储管理 XML 数据。XML 文档模式与数据库存储

模式的不同决定其不能直接载入到数据仓库中，于是也就引出了本文讨论的问题——基于数据库的 XML 文件存储技术。

2 XML 在数据库中的存储技术

XML 在数据库中能够正确的存储，要解决的主要问题是如何实现 XML 与数据库模式的相互映射。数据通过一定的 XML 转换和处理从而可以为传统的基于关系模型的数据库所支持。其中关键的技术在于数据及数据之间关系的映射，即如何处理 DTD 与数据库定义语言(DDL)间的转换。这样的映射通常分为两大类：模板驱动和模式驱动。

2.1 模板驱动的映射

以模板驱动的映射中，没有预先定义文档结构和数据库结构之间的映射关系，而是使用将命令语句内嵌入模板的方法，让数据传输中间件来处理该模板。下面是用 Oracle 的 xsql 文件返回图书数据。

```
<? xml version = "1.0"? >
<xsql:query xmlns:xsql = "urn:oracle-xsql" connection = "demo"
  Select title, author, description, cost from booklist where year = {@year}
</xsql:query >
```

当数据传输中间件处理到该文档时，每个 SELECT 语句都将被各自的执行结果所替换，从而得到某年的图书目录。这种以模板驱动的映射可以相当的灵活，但是目前以模板驱动的映射只支持从一个关系型数据库转换成 XML 文档的情况。

2.2 模型驱动的映射

在以模型驱动的映射中，利用 XML 文档结构将对应的数据模型显式或隐式地映射成数据库的结构，而且反之亦然。它的缺点是灵活性不够，但是却简单易用，这是因为它是基于具体的数据模型来进行映射的，通常能够为用户实现很多的转换工作。目前在 XML 文档数据视图模型中比较成熟的技术是表格模型。

许多中间件软件包都采用表格模型在 XML 和关系型数据库之间进行转换。它把 XML 的模型看成是一个单独的表格或者是一系列的表格。也就是说，XML 的文档的结构和下面的例子相类似，其中在单个表格的情况下，<database>并不出现：

```
<database >
  <table >
    <row >
      <column1 >... </column1 >
      <column2 >... </column2 >
    </row >
  </table >
</database >
```

3 应用实例

下面具体介绍如何实现 XML 在数据库中的存储，以我们做过的 CRM(客户关系管理系统)系统为例，它是用 XML 作为 Oracle9i 数据仓库的数据源，将 XML 数据存储到数据仓库中，用来对客户的信息进行相关主题分析。其详细实现如下。

3.1 将 XML 文档映射为关系模式

为方便开发者的工作，ORACLE 公司推出了相应的开发工具：ORACLE XML UTILITY，将整个 XML 文档元素对应为一个表，XML 文档内嵌元素对应为 ORACLE 的对象类型，这样可用 SQL－XML 实现数据表与 XML 文档的一对一转换，不过在实现上必须使用面向对象的数据表或者改造 XML 文档以配合数据表。下面举例进行简单介绍。

这是一个客户档案的 XML 文档，用于记录客户信息。

```
<? xml version = "1.0" encode = gb2312 >
<customer >
  <id >200509100001 </id >
  <name >李刚 </name >
  <gender >男 </gender >
  <contact >
    <phone >13689298715 </phone >
    <email >9502wzh@163.com </email >
    <address >西安市金花南路 5 号 </address >
  </contact >
</customer >
```

为了实现数据库信息与 XML 文档的对应，在 ORACLE 中应建立与此 XML 文档相对应的表，生成数据库的 SQL 语句为：

```
CREATE TABLE CUSTOMER
```

```
( ID CHAR(14) NOT NULL,
  NAME VARCHAR2(8) NOT NULL,
  GENDER CHAR(2)NOT NULL,
  CONTACT CONTACT_INFO_TYPE
)
```

其中的 CONTACT 字段对应 XML 文档中的嵌套元素(nested element)contact,在这里通过定义对象类型实现。

```
CREATE TYPE CONTACT_INFO_TYPE AS OBJECT
  (PHONE VARCHAR2(16),
   EMAIL VARCHAR2(30),
   ADDRESS VARCHAR2(30))
```

3.2 XML 在数据库中的存储

Oracle 通过建立数据表与 XML 文档的对应完成 XML 与 SQL 数据的映射,并由建立对象类型来实现 XML 嵌套元素。JAVA 类 ORACLEXMLSAVE 提供了方便的 XML 数据存储功能。

```
import java.sql.*;
import oracle.xml.sql.dml.OraclexmlSave;
public class testXMLInsert
{
  public static void main(string args[]) throws SQLException
  {
    connection conn = getConnection ("user","pass");
    OracleXMLSave sav=new OracleXMLSave(conn,"user.CUSTOMER");
    //把全部 XML 文档作为第一个参数执行
    sav.insertXML(args[0]);
    sav.close();
  }
  ..
}
```

Oracle 中不支持 XML 元素的属性(attribute)的存储,比如:

<customer type="wholesale">中的 type,这个问题可以通过把属性 type 转换为一个元素来解决,比如:

```
<customer>
  <type>wholesale</type>
...
</customer>
```

3.3 从数据库中返回 XML 格式数据

假设表中已经有了数据,现在要取出数据生成 XML 文档,以一段简单的 JAVA 程序作为示例,使用 JDBC 访问数据库。

```
import oracle.jdbc.driver.*;
import oracle.xml.sql.query.Oracle-XMLQuery;
import java.lang.*;
import java.sql.*;
class testXMLSQL
{
  public static void main(String[] args)
  {
    try
      {
      //建立连接
      Connection conn = getConnection ("user","pass");
      //建立查询类
      OracleXMLQuery qry = new Oracle-XMLquery(conn,"SELECT * FROM CUSTOMER");
      //得到 XML 字串
      String str=qry.getXMLString();
      //输出 XML 文档
      System.out.println(str);
      //关闭查询
      qry.close()
      }
  }
}
```

其中 getconnetion 为数据库连接函数

```
private static Connection getConnection(String username,String password) throws SQLException
{
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
Connection conn=null;
conn = DriverManager.getConnction("jdbc:oracle:thin:@myhost:1521:orcl",username,password);Return conn;
}
```

(下转第 38 页)

(上接第 34 页)

4 总结

根据 XML 标准化思路,结合数据库的优势,提出了数据库管理 XML 文件的数据存储方案,并且通过 Oracle 9i 大型数据库为例说明了数据存储转换的实现过程,设计了 XML 文件到数据库表模式的转换、存储、检索等具体操作,进一步优化了数据交换、传递的性能。利用 XML 技术解决了异构数据获取问题,完成了数据格式的统一,大大简化了建立数据仓库的数据的获取,转化和处理过程。通过对 XML 在数据库中存储技术的论述,使得两种数据技术达到了优势互补。随着这两种技术的相互借鉴和结合,数据的标准化工作和数据挖掘技术也会更加日臻完善。

参考资料

1 孟小峰,Web 数据:数据库技术面临的机遇与挑战[EB/OL],www. tongji. edu. cn.

2 邓东华、杨宗凯、乐春晖,基于 XML 的三层 c/s 模型[J],计算机系统应用,2001,(3):34 -36。

3 郑阿奇,ORACLE 实用教程[M],北京 电子工业出版社,2003。

4 仇丽青、赵庆祯,基于 XML 的数据仓库系统[J],计算机系统应用,2004,(2):12 -14。