用 XML DiffGram 对 SQL 数据库操作的方法

A Method of Operating SQL Database by Using XML DiffGram

蔡瑞奎　（广州 广东商学院教育技术中心 510320）

摘要：本文主要利用 XML DiffGram，通过 HTTP 实现对 SQL 数据库表的删除、插入和更新等操作。与一般操作数据库方法不同，该方法直接使用 XML 文件对 SQL Server 2000 的数据进行操作。在 SQLXML 3.0 支持下，该法是 XML 文件直接操作数据库的新方法。

关键词：DiffGram　SQLXML　HTTP

1　引言

目前，作为 Web 环境中组织数据的一种方式，常见的 HTML 描述了显示全球数据的通用方法，而 XML[1][2][3] 提供了直接处理全球数据的通用方法。XML 使用一个简单而有灵活的标准格式，为基于 Web 的应用提供了一个描述数据和交换数据的有效手段。

一般地，如果数据是高度结构化的，具有已知的架构，则关系模型[4] 可能对于数据存储最为有效。而如果结构是灵活的，即半结构化，非结构化或未知的，则必须适当地考虑如何对此类数据进行建模。为确保使用结构化和语义标记的数据的可移植性，需要独立于平台的模型，此时 XML 是一种很好的选择。

虽然 SQL Server 2000 具有强大的数据处理能力，可针对半结构化和非结构化的数据管理开发出功能丰富的应用程序。不过，使用 SQLXML 3.0 可以更广泛地访问其中的 XML 功能。本着对各种数据处理的相同目的，必须把 SQL 和 XML 有机结合起来，使两者能相辅相成，灵活处理运用，使二者达到完美的结合。SQL 的数据库存储管理能力加上 XML 数据结构表达的优异能力，使得 XML 结合 SQL 进行数据库工作越来越受到关注和重视。

对 SQL 数据库数据的操作，可以有很多方法，除在 SQL Server 上直接操作外，还可以使用其他高级编程语言来实现对数据库的操作，它们特别常见于与数据库管理应用系统相关的软件设计中。本文主要利用 XML DiffGram（可简称为差异元）的特性，通过 HTTP 实现对 SQL 数据库表的操作。

2　DiffGram 基本工作原理

SQL Server 2000 结合 SQLXML 3.0 为我们提供了强大的 XML 数据管理功能。这些功能致力于关系数据和 XML 数据之间的映射。

当运行基于 WEB 的数据应用时，一般使用数据集（DataSet）读取数据，然后对数据作些更新再送回数据库保存最终数据。如果当数据集里有成千上万的记录时，发送整个数据集就相当费时。此时，最好的办法是找出数据集更新过的行并仅仅发送更新过的数据回数据库，而不是整个数据集送回数据库。这正是 DiffGram 最有优势的地方。

当用户要更改数据时，客户应用程序就会将一个 DiffGram 发送到 XML Web 服务中。DiffGram 描述了已经插入、修改或删除了哪些行，即告诉 XML Web 服务更改了哪些数据。而后 XML Web 服务使用 XML DiffGram 来更新数据库。

DiffGram 工作在 AXSD（Annotated 注解 XSD）或 XDR 架构所提供的 XML 视图上。它使用在架构上提供的 XML 与数据结构相“映射”的关系，当然架构必须有信息来匹配与数据库表相对应的 XML 元素和属性，以便让 DiffGram 使用来更新数据库表。

3　DiffGram 格式和处理逻辑

DiffGram 是一个具有 XML 序列化格式的 XML 流。DiffGram 文件很详细地描述了数据集里表和行的状态，包含每一行当前的数据，以及被修改的行和未解决的错误初始值，所以有些冗长。

3.1 DiffGram 的格式

一般的 DiffGram 格式如下：

```
<? xml version = "1.0" ? >
<diffgr:diffgram
xmlns:msdata = " urn:schemas - microsoft - com:xml - msdata"
xmlns:diffgr = " urn:schemas - microsoft - com:xml - diffgram - v1"
xmlns: xsd = " http://www. w3. org/2001/XMLSchema" >
    <DataInstance >
      ...
    </DataInstance >
    [ <diffgr:before >
      ...
    </diffgr:before > ]
    [ <diffgr:errors >
      ...
    </diffgr:errors > ]
</diffgr:diffgram >
```

DiffGram 格式包含以下几块：

<DataInstance >：元素名 DataInstance 主要用于数据描述目的。该块包含所有改变后的相关数据，也可能包括没有被修改的数据。如果 diffgr:hasChanges 属性没指定，DiffGram 处理逻辑就会忽略该块里的元素。

<diffgr:before >：可选项，包括要更新或删除的原始记录元素的内容。所有要被 DiffGram 更新或删除的数据库表的内容必须作为顶级元素出现在 <before >块里。

<diffgr:errors >：可选项，被 DiffGram 处理逻辑忽略。

另外，DiffGram 还有一些属性：

Annotations（注解）：这些注解在 DiffGram 的名字空间" urn:schemas - microsoft - com:xml - diffgram - 01" 里定义。

Id：该属性用于在 <before > 和 <DataInstance > 块里构成元素对。diffgr:id 属性用于在当前行和原始行之间建立链接。

hasChanges：对插入和更新操作，DiffGram 必须指定该属性值为 " Inserted " 或 " Modified " 。如果该属性没有指定，在 <DataInstance > 里相应的元素就被处理逻辑忽略，并且不执行更新。diffgr:hasChanges 属性可以很快找到哪些记录被删除、插入或者只是更新。

parentID:DiffGram 中，该属性用于在元素之间指定父子关系。它仅出现在 <before > 块里。当要更新时，它被 SQLXML 使用。父子关系用于确定 DiffGram 里元素被处理的顺序。

hasErrors:表示在 <DataInstance > 块里有错误。该元素放在 <diffgr:errors > 里。

3.2 DiffGram 处理逻辑简介

DiffGram 处理逻辑使用某些规则来确定一个操作是插入、删除还是更新操作。这些规则简单介绍如下，见表 1：

表 1 处理逻辑说明表

操作	描述
插入	当一个元素出现在 <DataInstance > 块里但并不在相应的 <before > 块，且该元素 diffgr:hasChanges 的属性值为 inserted 时，DiffGram 就是一个插入操作。DiffGram 把在 <DataInstance > 块里指定的记录内容插入到数据库相应的表里。如果 diffgr:hasChanges 属性没有指明，元素被处理逻辑忽略，不执行插入操作。
更新	当在 <before > 块里有一个元素与 <DataInstance > 块里有相对应的元素，即两个元素具有相同的 diffgr:id 属性值，并且元素的 diffgr:hasChanges 属性值在 <DataInstance > 块里指定为 modified 时，DiffGram 是一个更新操作。如果元素的 diffgr:hasChanges 属性在 <DataInstance > 块里没指定，处理逻辑返回一个错误。如果 diffgr:parentID 在 <before > 块里指定，则由 parentID 指定元素的父子关系来确定记录被更新的顺序。
删除	当一个元素出现在 <before > 块，但不出现在相应的 <DataInstance > 块里时，该 DiffGram 是一个删除操作。此时，DiffGram 从数据库里删除在 <before > 块里指定的相应记录内容。如果 diffgr:parentID 是在 <before > 块里指定，则由 parentID 指定元素的父子关系来确定记录被删除的顺序。

4 SQLXML 3.0 配置要点

配置 SQLXML 3.0[5] 在此就不详细介绍，仅介绍配置要点。设定一个虚拟名称 xmlvn（类型为 template 模板型），如下图 1 所示。

5 示例演示

在 SQLXML 3.0 支持下，先构造演示用的数据库

表,再加入部分具体数据,然后编写 DiffGramSchema. xml,之后编写操作 XML DiffGram 代码,最后在 URL 上执行 HTTP 操作。

图 1　配置 SQLXML 3.0

5.1 数据库结构及代码

构造两个数据库表 Cust 和 Ord 并加入部分内容:

```
CREATE TABLE Cust(
CustomerID nchar(5) Primary Key,
CompanyName nvarchar(40) NOT NULL ,
ContactName nvarchar(60) NULL)
CREATE TABLE Ord(
OrderID int Primary Key,
CustomerID nchar(5) Foreign Key REFERENCES Cust(CustomerID))
INSERT INTO Cust (CustomerID, CompanyName, ContactName) VALUES
('GDKND', '广东康能电脑有限公司', '李德能')
INSERT INTO Cust (CustomerID, CompanyName, ContactName) VALUES
('GZHYD', '广州华业电脑设备厂', '王经理')
INSERT INTO Cust (CustomerID, CompanyName, ContactName) VALUES
('GZJYD', '广州佳盈电脑有限公司', '廖经理')
INSERT INTO Ord(OrderID, CustomerID) VALUES(1, 'GDKND')
INSERT INTO Ord(OrderID, CustomerID) VALUES(2, 'GZHYD')
INSERT INTO Ord(OrderID, CustomerID) VALUES(3, 'GZJYD')
```

而 DiffGram 架构代码如下:

DiffGramSchema. xml

```
<? xml version = "1.0" encoding = "gb2312" ? >
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
xmlns:sql = "urn:schemas-microsoft-com:mapping-schema" >
<xsd:annotation >
<xsd:documentation >
```

Diffgram 客户/定单 Schema 架构示例。

```
</xsd:documentation >
<xsd:appinfo >
<sql:relationship name = "CustomersOrders" parent = "Cust"
parent-key = "CustomerID" child-key = "CustomerID"
child = "Ord" / >
</xsd:appinfo >
</xsd:annotation >
<xsd:element name = "Customer" sql:relation = "Cust" >
<xsd:complexType >
<xsd:sequence >
<xsd:element name = "CompanyName" type = "xsd:string" / >
<xsd:element name = "ContactName" type = "xsd:string" / >
<xsd:element name = "Order" sql:relation = "Ord"
sql:relationship = "CustomersOrders" >
<xsd:complexType >
<xsd:attribute name = "OrderID" type = "xsd:int"
sql:field = "OrderID" / >
<xsd:attribute name = "CustomerID" type = "xsd:string" / >
</xsd:complexType >
</xsd:element >
</xsd:sequence >
<xsd:attribute name = "CustomerID" type = "xsd:string"
sql:field = "CustomerID" / >
</xsd:complexType >
</xsd:element >
</xsd:schema >
```

进行所有操作示例的 all-operate. xml 文件代码如下,包括删除、插入、更新等操作:

```
<? xml version = "1.0" encoding = "gb2312" ? >
```

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql" sql:mapping-schema="DiffGramSchema.xml">
  <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
<!--Customer2 要更新的内容-->
  <DataInstance>
    <Customer diffgr:id="Customer2" msdata:rowOrder="1"
diffgr:hasChanges="modified"
CustomerID="GZHYD">
      <CompanyName>广州金伦飞电脑有限公司</CompanyName>
      <ContactName>蔡先生</ContactName>
      <Order diffgr:id="Order2" msdata:rowOrder="1"
msdata:hiddenCustomerID="GZHYD"
CustomerID="GZHYD" OrderID="2" />
    </Customer>
<!--end of Customer2 更新-->
<!--插入第三条记录 Customer4-->
    <Customer diffgr:id="Customer4" msdata:rowOrder="3"
diffgr:hasChanges="inserted"
CustomerID="GZDBD">
      <CompanyName>广州都邦电脑有限公司</CompanyName>
      <ContactName>杨先生</ContactName>
      <Order diffgr:id="Order4" msdata:rowOrder="3"
diffgr:hasChanges="inserted"
msdata:hiddenCustomerID="GZDBD"
CustomerID="GZDBD" OrderID="4" />
    </Customer>
  </DataInstance>
<!--end of 插入 Customer4-->
<!--删除 customer1-->
  <diffgr:before>
    <Order diffgr:id="Order1" msdata:rowOrder="0"
msdata:hiddenCustomerID="GDKND"
CustomerID="GDKND" OrderID="1" />
    <Customer diffgr:id="Customer1" msdata:rowOrder="0"
CustomerID="GDKND">
      <CompanyName>广东康能电脑有限公司</CompanyName>
      <ContactName>李德能</ContactName>
    </Customer>
<!--end of 删除 customer1-->
<!--需更新的 Customer2-->
    <Customer diffgr:id="Customer2" msdata:rowOrder="1"
      CustomerID="GZHYD">
      <CompanyName>广州华业电脑设备厂</CompanyName>
      <ContactName>王经理</ContactName>
    </Customer>
  </diffgr:before>
</diffgr:diffgram>
</ROOT>
```

5.2 使用 XML 对 SQL 操作示例

由于在 SQLXML 3.0 下 DiffGram 是作为一个模板文件，可通过 HTTP 发送 DiffGram，因此可保存为一个文件并作为一个 XML 模板文件在 URL 上执行。

利用 SQLXML 3.0 可以设虚拟目录名称为 operate，在其后接虚拟名称 xmlvn，直接输入上面的 all-operate.xml 文件。该 XML 文件包含删除、更新、插入等操作。如下所示：

http://localhost/operate/xmlvn/all-operate.xml

6 结论

在 SQL Server 2000 中，要在 HTTP 下对数据库进行操作，可以在 SQLXML 3.0 的支持下使用 XML DiffGram 文件实现对数据库数据操作，它是 XML 格式文件操作 SQL 数据库新的技术手段。该方法主要从 XML 文件角度出发，利用 XML DiffGram 里面数据不同状态的描述来控制 SQL 数据库数据，而不是通过常见的 DELETE，INSERT，UPDATE 等命令来对 SQL 数据库进行操作。利用该方法，使得 XML 的功能更加强大，显示它不是单纯的描述数据结构的功能，而且能够更好、更有效地控制 SQL 数据库里的数据。

参考文献

1 http://www.xml.org/

2 http://www.w3.org/XML/

3 Charles F. Goldfarb? Paul Prescod，XML Handbook，Fourth Edition，Publishing House of Electronics Industry，2004.

4 王能斌编著，数据库系统原理，电子工业出版社，2001 年。

5 http://msdn.microsoft.com/downloads/