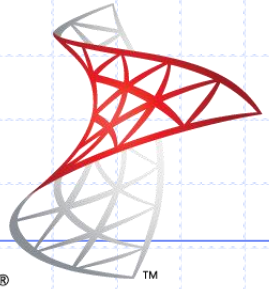

Microsoft® SQL Server® 2008 Tutorial: SELECT … FOR XML

IT 4153
Advanced Database

J.G. Zheng
Spring 2012

SPSU SOUTHERN POLYTECHNIC STATE UNIVERSITY

Relational Data and XML

- XML Document vs. XML Data
  - Document centric XML file
    - Focus on content
    - Fewer tags, less structured
  - Data centric XML file
    - Focus on data and structure
    - More tags, more structured
- Relational data (table) can be transformed to XML format (data centric XML file)

Simple Relation-to-XML Guideline

- The table becomes the root element (a complex type): may use the table name as the root element name.
- Each row (record) becomes direct child elements (complex types) under the root element.
- Each value in the row becomes (two choices)
  - an attribute of the row element (the column name becomes the attribute name, and the data becomes the attribute value), or
  - an third level child element (simple type) under the row element: the column name becomes the element name and the data becomes the text node under the element.

Example: Shippers Table

- Transforming a single table
  - The “Shippers” table in the “Northwind” database.

	Column Name	Data Type	Allow Nulls
🔑	ShipperID	int	☐
	CompanyName	nvarchar(40)	☐
	Phone	nvarchar(24)	☑

	ShipperID	CompanyName	Phone
1	1	Speedy Express	(503) 555-9831
2	2	United Package	(503) 555-3199
3	3	Federal Shipping	(503) 555-9931

Generating XML from SELECT

- SQL Server 2008
  - Directly format data into XML format using the "FOR XML" clause in SQL SELECT queries
- Example

```
SELECT * from Books FOR XML AUTO
```

FOR XML Customization

```
SELECT * FROM Shippers
FOR XML AUTO;
```

The simplest "AUTO" mode generates XML file like this. Each row in the table is transformed into an element (row element) with attributes. The table name is used as the row element name; and column names are used as attribute names. There is no root element defined.

```
<Shippers ShipperID="1" CompanyName="Speedy Express" Phone="(503) 555-9831" />
<Shippers ShipperID="2" CompanyName="United Package" Phone="(503) 555-3199" />
<Shippers ShipperID="3" CompanyName="Federal Shipping" Phone="(503) 555-9931" />
```

```
SELECT * FROM Shippers AS Shipper
FOR XML AUTO, ROOT('Shippers');
```

The table alias customizes the row element name.

The ROOT setting adds a root element with a given name.

```
<Shippers>
  <Shipper ShipperID="1" CompanyName="Speedy Express" Phone="(503) 555-9831" />
  <Shipper ShipperID="2" CompanyName="United Package" Phone="(503) 555-3199" />
  <Shipper ShipperID="3" CompanyName="Federal Shipping" Phone="(503) 555-9931" />
</Shippers>
```

FOR XML Customization: Elements

Table alias is used for the row level element name.

```
SELECT *
FROM Shippers AS Shipper
FOR XML AUTO,
ROOT('Shippers'), ELEMENTS;
```

The ELEMENTS setting uses elements instead of attributes.

```
<Shippers>
  <Shipper>
    <ShipperID>1</ShipperID>
    <CompanyName>Speedy Express</CompanyName>
    <Phone>(503) 555-9831</Phone>
  </Shipper>
  <Shipper>
    <ShipperID>2</ShipperID>
    <CompanyName>United Package</CompanyName>
    <Phone>(503) 555-3199</Phone>
  </Shipper>
  <Shipper>
    <ShipperID>3</ShipperID>
    <CompanyName>Federal Shipping</CompanyName>
    <Phone>(503) 555-9931</Phone>
  </Shipper>
</Shippers>
```

Parent-to-Child (One-to-Many)

- Put parent table in higher hierarchy

```
<?xml version="1.0" encoding="utf-8" ?>
<Shippers>
        <Shipper>
         <CompanyName>Speedy Express</CompanyName>
         <Orders>
                ...
        </Orders>
        </Shipper>
</Shippers>
```

- Put child table in higher hierarchy

```
<?xml version="1.0" encoding="utf-8" ?>
<Orders>
        <Order id="11102">
        <Shipper>
                <CompanyName>Speedy Express</CompanyName>
         </Shipper>
        </Order>
        ...
</Orders>
```

FOR XML for Multiple Tables

- Put parent table in higher hierarchy

```
SELECT ShipperId, CompanyName,
(
        SELECT OrderId, OrderDate
         FROM   Orders [Order]
         WHERE [Order].ShipVia = Shipper.ShipperID
         FOR XML AUTO, TYPE, ELEMENTS, ROOT('Orders')
)
FROM Shippers AS Shipper
ORDER BY ShipperId
FOR XML AUTO, ELEMENTS, ROOT('Shippers');
```

A sub query for inner level of XML

The join condition refer to the outer query table

Use type to specify the output of the inner sub-query is an XML data type

FOR XML for Multiple Tables

- Put child table in higher hierarchy

A sub query for inner level of XML

```
SELECT OrderId, OrderDate,
      (
            SELECT ShipperId, CompanyName
            FROM Shippers AS Shipper
            WHERE [Order].ShipVia = Shipper.ShipperID
            FOR XML AUTO, TYPE
      )
FROM   Orders [Order]
FOR XML AUTO, TYPE, ELEMENTS, ROOT('Orders')
```

Use type to specify the output of the inner sub-query is an XML data type

Key Resources

- SELECT FOR XML complete reference
  - http://msdn.microsoft.com/en-us/library/ms178107.aspx