Processing XML with SQL on the IBM i

MMSA MEETING – April 15, 2014

Raymond A. Everhart - RAECO Design, Inc.
reverhart@raecodesign.com
Objectives

• Introduction / Overview
• Terms / Concepts
• Create DDL for table to store XML Data
• SQL / XML Functions
• Example - Create XML using SQL
• Example - Consume / Decompose Shred XML using SQL
• Use Embedded SQL in RPG to create an XML Document
• Credits / References / Other Resources
Introduction / Overview

- Available with IBM i v7r1 TR4 – April 13, 2010
- Replaces XML Extender for DB2
- Poll / Survey
  - What is your SQL proficiency?
  - What is your OS level?
  - Currently Publishing XML?
  - Currently Consuming XML?
  - Building XML with O.S.X.?

(Old School X ml - a.k.a. String Processing)
Terms / Concepts

• Hierarchal vs Relational data
  • Stable vs Evolving Requirements
  • Flexibility

• SQL Data Types
  • CLOB – Character Large Object
  • BLOB – Binary Large Object
  • XML – Internal Representation of Hierarchal Data

• Implicit Conversion
Terms / Concepts

• Document – Well formed set of hierarchical data
• Elements – Representation of information
• Attributes – Additional information about an element
• Namespaces – A method of preventing Element Naming Collisions
• Serialization – Transforming XML data into a string
• Decomposing / Shredding – Transforming Hierarchical data into Relational data
• Xpath – A method for specifying the location of elements in an XML document
CREATE TABLE Customer (Cid BIGINT NOT NULL PRIMARY KEY,
                      Info XML);
Storing XML Data into a DB2 table

INSERT INTO Customer (Cid, Info)
VALUES (1000,
' <customerinfo xmlns="http://posample.org" Cid="1000">
  <name>Kathy Smith</name>
  <addr country="Canada">
    <street>5 Rosewood</street>
    <city>Toronto</city>
    <prov-state>Ontario</prov-state>
    <pcode-zip>M6W 1E6</pcode-zip>
  </addr>
  <phone type="work">416-555-1358</phone>
</customerinfo>');
SQL / XML Functions

• **XMLAGG aggregate function** - Returns an XML sequence containing an item for each non-null value in a set of XML values.

• **XMLATTRIBUTES scalar function** - Constructs XML attributes from the arguments. This function can be used only as an argument of the XMLELEMENT function.

• **XMLCOMMENT scalar function** - Returns an XML value with the input argument as the content.

• **XMLCONCAT scalar function** - Returns a sequence containing the concatenation of a variable number of XML input arguments.
SQL / XML Functions

- **XMLDOCUMENT scalar function** - Returns an XML value that is a well-formed XML document. Every XML value that is stored in a DB2 table must be a document. This function forms an XML document from the XML value.

- **XMLELEMENT scalar function** - Returns an XML value that is an XML element. Every XML value that is stored in a DB2 table must be a document. The XMLELEMENT function does not create a document, only an element. The stored XML value must be a document formed by the XMLDOCUMENT function.

- **XMLFOREST scalar function** - Returns an XML value that is a sequence of XML elements.
SQL / XML Functions

- **XMLGROUP aggregate function** - Returns a single top-level element to represent a table or the result of a query. By default each row in the result set is mapped to a row subelement and each input expression is mapped to a subelement of the row subelement. Optionally, each row in the result can be mapped to a row subelement and each input expression to be mapped to an attribute of the row subelement.

- **XMLNAMESPACES declaration** - Constructs namespace declarations from the arguments. This declaration can be used only as an argument of the XMLELEMENT and XMLFOREST functions.

- **XMLPI scalar function** - Returns an XML value with a single processing instruction.
SQL / XML Functions

• **XMLROW scalar function** - Returns a sequence of row elements to represent a table or the result of a query. By default each input expression is transformed into a sub element of a row element. Optionally, each input expression can be transformed into an attribute of a row element.

• **XMLTEXT scalar function** - Returns an XML value that contains the value of the input argument.

• **XSLTRANSFORM scalar function** - Converts XML data into other formats, including other XML schemas.
Example - Create XML using SQL

SELECT XMLELEMENT (NAME "allProducts",
    XMLNAMESPACES (DEFAULT 'http://posample.org'),
    XMLAGG(XMLELEMENT (NAME "item", p.name)))
FROM Product p

Result:

<allProducts xmlns="http://posample.org">
    <item>Snow Shovel, Basic 22 inch</item>
    <item>Snow Shovel, Deluxe 24 inch</item>
    <item>Snow Shovel, Super Deluxe 26 inch</item>
    <item>Ice Scraper, Windshield 4 inch</item>
</allProducts>
Example - Shredding XML using SQL

DDL to create a table of sample XML data:

CREATE TABLE EMP (DOC XML)
Row 1 data:

```xml
<dept bldg="101">
  <employee id="901">
    <name><first>John</first><last>Doe</last></name>
    <office>344</office>
    <salary currency="USD">55000</salary>
  </employee>
  <employee id="902">
    <name><first>Peter</first><last>Pan</last></name>
    <phone>905-416-5004</phone>
  </employee>
</dept>
```
Example - Shredding XML using SQL

Row 2 data:
<dept bldg="114">
  <employee id="903">
    <name><first>Mary</first><last>Jones</last></name>
    <office>415</office>
    <phone>905-403-6112</phone>
    <phone>647-504-4546</phone>
    <salary currency="USD">64000</salary>
  </employee>
</dept>
SELECT X.*
FROM emp,
XMLTABLE ('$d/dept/employee' PASSING emp.doc AS "d"
COLUMNS
  empID INTEGER PATH '@id',
  firstname VARCHAR(20) PATH 'name/first',
  lastname VARCHAR(25) PATH 'name/last') AS X
Example - Shredding XML using SQL

Results:

<table>
<thead>
<tr>
<th>EMPID</th>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>901</td>
<td>John</td>
<td>Doe</td>
</tr>
<tr>
<td>902</td>
<td>Peter</td>
<td>Pan</td>
</tr>
<tr>
<td>903</td>
<td>Mary</td>
<td>Jones</td>
</tr>
</tbody>
</table>
Example – XML / SQL functions embedded in RPG

RDi Demo
Credits / References / Other Resources

- **Must Have** - IBM I Database SQL XML Programming
  

- “i Can” Blog - [http://ibmsystemsmag.blogs.com/i_can/xml/](http://ibmsystemsmag.blogs.com/i_can/xml/)

- **Using XML With DB2 for I**
  