X.M.L. DB

Pellizzaro Massimiliano

Milano, Settembre 2003

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Agenda

- XMLType: inserting xml data into Oracle 9i
  - plain xml
  - using schemas
  - Appling xslt
- Extracting xml data from relational data tables
  - plain data
  - using xslt
- Using XML DB to view data using http
- Loading Servlets into Oracle 9i
- Using XSU
- A practical example: H3G data stage

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

XMLType

- It is a new data type introduced by Oracle 9i
- It can be used in strore proc. as input param, returned value, variable
- As instance of if represent an xml document
- Once created a table/column, you can use XMLType as a constructor :
  - XMLType(<xmlDOC>…</xmlDOC>)
  - XMLType(clob)
  - XMLType(blob)
  - …
- XMLType has many useful function:
  - getStringVal()
  - getClobVal()
  - getNumberVal()
  - …

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Create xml repository

- Create relation table with XMLType as a column
- Create XMLTaype table

```
CREATE TABLE ProvaXML
(
 KEYVALUE INTEGER PRIMARY KEY,
 XMLCOLUMN xmltype
);
```

```
CREATE TABLE XMLTABLE OF XMLTYPE ;
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Simple uploading xml file

- UTL_FILE_DIR = 'C:\max\tmp\xmldir (ora.ini)
- **create or replace directory** XMLDIR **as** 'C:\max\tmp\xmldir';
- grant read on directory xmldir to public with grant option;
- **create or replace function** getDocument(filename **in varchar2**) **return clob is**

```
  xclob clob;
  xbfile bfile;
begin
  xbfile := bfilename('XMLDIR',filename);
  dbms_lob.open(xbfile);
  dbms_lob.createtemporary(xclob,TRUE,dbms_lob.session);
  dbms_lob.loadfromfile(xclob,xbfile,dbms_lob.getlength(xbfile));
  dbms_lob.close(xbfile);
  return xclob;
end getDocument;
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Let’s go back in prova.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ADDRESS_BOOK>
  <ADDRESS>
        <NAME>Micheal Daconta</NAME>
        <STREET>4296 Razon Hill Road</STREET>
        <CITY>Beleton</CITY>
        <STATE>VA</STATE>
        <ZIP>22712</ZIP>
  </ADDRESS>
  <ADDRESS>
        <NAME>Max Pellizzaro</NAME>
        <STREET>3256 West 8th</STREET>
        <CITY>Larned</CITY>
        <STATE>KS</STATE>
        <ZIP>67775</ZIP>
  </ADDRESS>
</ADDRESS_BOOK>
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Inserting and selecting data

- Inseting data: with default check
  - **insert into** xmltable **values**(**XMLTYPE**(getdocument('prova.xml')));
  - **insert into** provaxml (keyvalue,xmlcolumn) **values**(1,XMLTYPE(getdocument('prova.xml')));
- Use of XMLSchema
  - Use a schema not registered
  - Use a registered schema
- Non registered Schema:
  - xmltype(xmlData IN clob, schama IN varchar2:= NULL, validated IN number:=0, wellformed IN number:=0) return self as a result
  - xmltype(xmlData IN varchar2, schama IN varchar2:= NULL, validated IN number:=0, wellformed IN number:=0) return self as a result

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Use a registered schema

- Register the schema:
  - xdb.dbms_xmlschema.registerschema(
    'http://www.maxpellizzaro.com/xsd/provaschema.xsd',
    GetDocument('provaschema.xsd'));
- Define a table/column as a schema validating

```
CREATE TABLE ProvaSchemaXML OF XMLTYPE
XMLSCHEMA  "http:///www.max.pellizzaro.com/xsd/provaschema.xsd"
ELEMENT "ADDRESS_BOOK"
```

- Insert data…

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Searching for data: XPath

- **XPath is a W3C recommendation for navigating XML documents**
- **XPath model XML document as a tree of nodes**
- **You biuld an XPath expression to locate a portion of the document**
- **Some examples (abbreviation)**
  - / -> is a location root path
  - /ADDRESS_BOOK/NAMES -> children path
  - /ADDRESS_BOOK/NAMES[1] -> first element
  - /ADDRESS_BOOK/NAMES[1]@name -> attribute element
  - /ADDRESS_BOOK/NAMES[1]/text() -> text element
  - /ADDRESS_BOOK/NAMES[@name = “ciccio”] -> evaluating expression
  - /ADDRESS_BOOK/NAMES[name=“ciccio”] -> evaluating expression
  - /ADDRESS_BOOK[position() = last()] -> evaluating expression

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Selecting data

- select * from
  - Return XMLTypes

- existsNode
  - **select** existsNode(**value**(X),'/ADDRESS_BOOK/ADDRESS') **FROM** xmltable X;
  - It returns 0 or 1 value
  - The most common use is a where condition: **select** count(*) **FROM** xmltable X where existsNode(**value**(X),'/ADDRESS_BOOK/ADDRESS') =1

- extractValue
  - select extractValue(value(X),'/ADDRESS_BOOK/ADDRESS/CITY') FROM xmltable X
  - It returns **only one** node value
  - It can be used in where condition

- extract
  - select extract(value(X),'/ADDRESS_BOOK/ADDRESS/CITY') FROM xmltable X
  - It is used to extract a collection of value for each rows

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Other functions:

- xmlsequence
  - It converts a fragment, returned by extract, into a XMLType
- updatexml
  - update xmltable t set value(t) = updateXML(value(t),'xpathexpression','xpathvalue')
- ….

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Applying xslt

- xsl is a valid xml document, so it can be stored in a data table as an XMLType
- XMLType has a tranform( ) method
  - select **value(t).transform(xmltype(getdocument('prova.xsl')))** from xmltable

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Using XML DB : Java API

```
{
      String query = "select A.XMLCOLUMN.getClobVal() XMLC from provaxml A";
      OraclePreparedStatement stmt =
   (OraclePreparedStatement)conn.prepareStatement(query);
      ResultSet rs = stmt.executeQuery();
      OracleResultSet ors = (OracleResultSet) rs;

      while(ors.next())
      {
        // XMLType xml = XMLType.createXML(ors.getOPAQUE(1));
        CLOB clb = ors.getCLOB(1);
        String xml=clb.getSubString(1,clb.getBufferSize());
        System.out.println(xml);
       }
}
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

What we have learned

- Inserting XML document into a DB
- Validating XML document
- Extract XML document or portion of it

- NEXT Step: generating data as xml
  - SQLX Functions
  - DBMS_XMLGEN
  - SQL Functions
  - XSQL Servlet
  - XSU

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Generate simple XML node : XMLELEMENT

Øselect XMLELEMENT("Emp",e.ename || ' ' || e.job) as name from emp e

NAME

<Emp>SMITH CLERK</Emp>
<Emp>ALLEN SALESMAN</Emp>
<Emp>WARD SALESMAN</Emp>
<Emp>JONES MANAGER</Emp>
<Emp>MARTIN SALESMAN</Emp>
<Emp>BLAKE MANAGER</Emp>
<Emp>CLARK MANAGER</Emp>
<Emp>SCOTT ANALYST</Emp>
<Emp>KING PRESIDENT</Emp>
<Emp>TURNER SALESMAN</Emp>
<Emp>ADAMS CLERK</Emp>

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Generate nested nodes

Ø select XMLELEMENT("Emp",
XMLELEMENT("name",e.ename),XMLELEMENT("job",e.job)) from emp e

```
<Emp>
  <name>SMITH</name>
  <job>CLERK</job>
</Emp>

<Emp>
  <name>ALLEN</name>
  <job>SALESMAN</job>
</Emp>

<Emp>
  <name>WARD</name>
  <job>SALESMAN</job>
</Emp>
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Generate XML attributes

- select XMLELEMENT("Emp", XMLATTRIBUTES(e.ename,e.job as "work")) from emp e

```
<Emp ENAME="WARD" work="SALESMAN"/>
<Emp ENAME="JONES" work="MANAGER"/>
<Emp ENAME="MARTIN" work="SALESMAN"/>
<Emp ENAME="BLAKE" work="MANAGER"/>
<Emp ENAME="CLARK" work="MANAGER"/>
<Emp ENAME="SCOTT" work="ANALYST"/>
<Emp ENAME="KING" work="PRESIDENT"/>
<Emp ENAME="TURNER" work="SALESMAN"/>
<Emp ENAME="ADAMS" work="CLERK"/>
<Emp ENAME="JAMES" work="CLERK"/>
<Emp ENAME="FORD" work="ANALYST"/>
<Emp ENAME="MILLER" work="CLERK"/>
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Other SQLX functions:

- XMLForest()
- XMLSequence()
- XMLConcact()
- XMLAgg()

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Using java Api

```
try
    {
     String query = "select XMLELEMENT(\"Emp\", XMLATTRIBUTES(e.ename,e.job
as \"work\")).getClobVal() from emp e";
     OraclePreparedStatement stmt =
(OraclePreparedStatement)conn.prepareStatement(query);
     ResultSet rs = stmt.executeQuery();
     OracleResultSet ors = (OracleResultSet) rs;

     while(ors.next())
     {
       CLOB clb = ors.getCLOB(1);
       String xml=clb.getSubString(1,clb.getBufferSize());
       System.out.println(xml);
      }
     }
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Using DBMS_XMLGEN package

- XMLGEN is a build in package that allows to work with relational data in and XML out
- It allows to transform and “select” into a xml document

- function:
  dbms_xmlgen.getXML('select * from scott.emp’)
- java:

```
String query = "select dbms_xmlgen.getXML('select * from scott.emp') from dual";
  ….. ;
 while(ors.next())
 {
   CLOB clb = ors.getCLOB(1);
   String xml=clb.getSubString(1,clb.getBufferSize());
   System.out.println(xml);
  }
```


© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Additional function

- SYS_XMLGEN ( )
- SYS_XMLAGG ( )

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

URI: accessing data using http://..

- Oracle has a built in web server
- You can access web server using:
  - http://localhost:8080/
- xdbconfig.xml shows the configuration of the data base
- You can access to xmlconfig using ftp server connected as xdb user
- The most important build in servlet is : DBURIServlet

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Using DBURIServlet

- http://localhost:8080/SCOTT/EMP
- http://localhost:8080/'any valid xpath expression'
  - http://localhost:8080/oradb/SCOTT/EMP/ROW[EMPNO=7369]

```
<?xml version="1.0" ?>
- <ROW>
  <EMPNO>7369</EMPNO>
  <ENAME>SMITH</ENAME>
  <JOB>CLERK</JOB>
  <MGR>7902</MGR>
  <HIREDATE>17-DEC-80</HIREDATE>
  <SAL>800</SAL>
  <DEPTNO>20</DEPTNO>
  </ROW>
```

- It is possible to retireve a single text value:
  - http://localhost:8080/oradb/SCOTT/EMP/ROW[EMPNO=7369]/ENAME/text()

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Using and installing Servlet

- Write a servlet
- Compile the servlet
- Load the servlet:
  - loadjava –grant public –u scott/tiger –r test.class
- Configure xdbconfig.xml
- try….

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

Oracle enterprise manager

- Viewing demo…

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

XSU

- Xml Sql Utility
  - transform data from database to XML
  - Generate DTDs
  - Generate Schema
  - Transform XML with XSL
  - extract data from XML and inserting into table/column
  - extract data from XML and insert / update appropriate value

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com

XSU command line:

- java OracleXML getXML –user “scott/tiger” “select * from emp”
- java OracleXML putXML -user “scott/tiger” –filename “tmp/prova.xml” “emp”
- The two classes that makes up XSU:
  - oracle.xml.sql.query.OracleXMLQuery
  - oracle.xml.sql.dml.OracleXMLSave

© Copyright IBM Corporation 2002

PDF created with pdfFactory Pro trial version www.pdffactory.com