



ArcelorMittal

# DB2 pureXML experiences at ArcelorMittal Gent

Davy Goethals

GSE DB2 working group

04/12/2014 Euroclear Brussel

# Overview

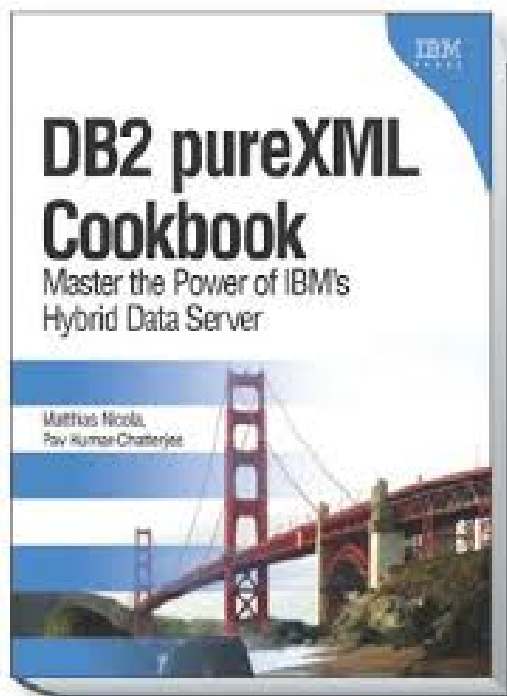
- How to start : collecting doc and choosing a case
- Producing XML documents using QMF
- Creating and loading a table with XML columns
- Querying the XML data with SQL and XPath
- Validate XML data using an XML Schema
- Create and use XML indexes

# Starting with pureXML

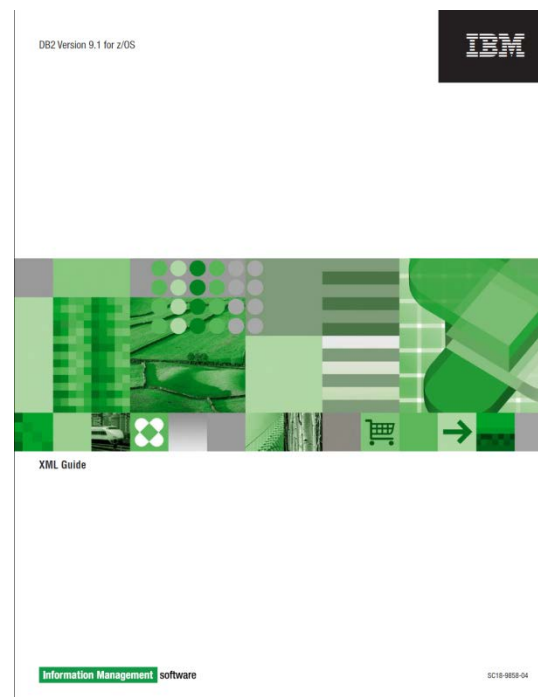


# Collecting doc

## The Bible

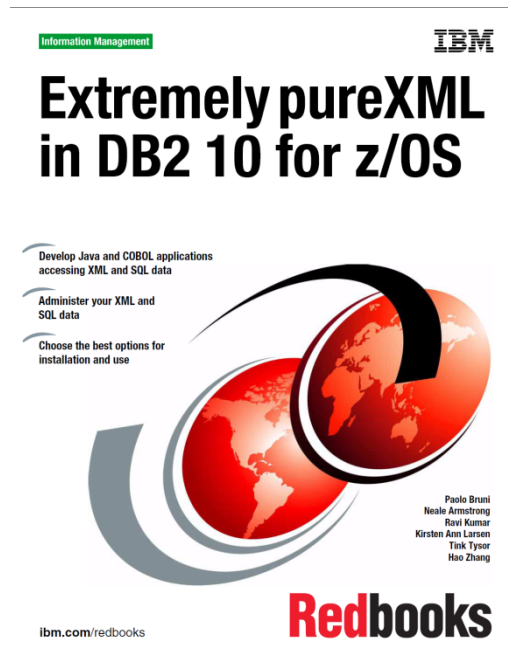


## DB2 XML Guide

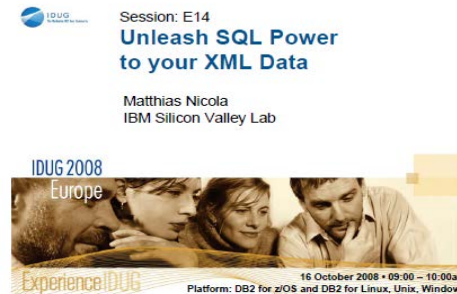


# Collecting doc

## XML Redbook



## IDUG presentations



## Collecting XML sample data

- DB2 installation samples
  - DB2.NEW.SDSNSAMP(DSNTEJ1)
  - DB2.NEW.SDSNSAMP(DSNTEJ2H)

CREATOR	NAME	COLUMN	Rows	DB2 Express C
DSN8910	PRODUCT	DESCRIPTION	4	4
DSN8910	CUSTOMER	INFO HISTORY	0	6
DSN8910	PURCHASEORDER	PORDER	0	6
DSN8910	CATALOG	CATLOG	0	0
DSN8910	SUPPLIER	ADDR	0	0

# Collecting XML sample data

- Table DSN8910.PRODUCT column DESCRIPTION

```
<product pid="100-100-01">
  <description>
    <name>Snow Shovel, Basic 22 inch</name>
    <details>Basic Snow Shovel, 22 inches wide, straight handle with D-grip</details>
    <price>9.99</price>
    <weight>1 kg</weight>
  </description>
</product>

<product pid="100-101-01">
  <description>
    <name>Snow Shovel, Deluxe 24 inch</name>
    <details>A Deluxe Snow Shovel, 24 inches wide, ergonomic curved handle with D-Grip</details>
    <price>19.99</price>
    <weight>2 kg</weight>
  </description>
</product>
```

## Collecting XML sample data

- Load IBM GSDB database in z/OS
  - “Great-outdoors-sample” : 153 tables 1Gb data
  - 1 XML column PTNR\_ORDER in table GOSALESCCT.  
PTNR\_ACTIVITY with 212 rows

```
<cust_order>
  <cust_code>          100022          </cust_code>
  <cust_order_number>  181649          </cust_order_number>
  <cust_order_date>    4/9/2007 8:43    </cust_order_date>
  <cust_unique_items>  2              </cust_unique_items>
  <cust_order_details1 product_number="61110" cust_quantity="1" cust_unit_sale_price="74.91"/>
  <cust_order_details2 product_number="99110" cust_quantity="1" cust_unit_sale_price="11.64"/>
</cust_order>
```



# Building our own PureXML case with QMF

## Collecting XML sample data

- Only two samples found so far for DB2 on z/OS
- Wanted a “richer case” to play with :
  - More complicated XML documents
  - More rows , more volume
  - With an XML schema to test validation
  - But comprehensive for DBA and developers
- Decided to build an own case based on QMF exported data in XML format to use as proof-of-concept.

## Exporting data in QMF to XML format

- EXPORT DATA TO ... (DATAFORMAT=XML)
  - Export result of QMF query to TSO SEQ file or HFS Unix file
  - Result may already contain XML columns before exporting
    - Uses z/OS XML parse services and z/OS Unicode services
  - Data exported as Unicode UTF-8 XML file (CCSID 1208)
    - Header records
    - Metadata records for each column in the result set
    - Data records for each column in the result set
  - Style sheet and XML schema provided
    - qmf\_dataset.xslt and qmf\_data.xsd

## Exporting data in QMF to XML format

- Query :
  - SELECT NAME,SALARY FROM Q.STAFF FETCH FIRST 3 ROWS ONLY
- Command :
  - EXPORT DATA TO 'SIDDAGO.STAFF.XML'(DATAFORMAT=XML
  - EXPORT DATA TO '/data/local/clipbrd/Users/siddago/staff.xml'(DATAFORMAT=XML

NAME	SALARY
SANDERS	18357.50
PERNAL	18171.25
MARENGHI	17506.70

# Exporting data in QMF to XML format

- Header records

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!-- A style sheet has been provided by the QMF product. You can find it
in the QMF SAMPLE library. Copy it to the directory where you exported
the file. The next comment is an example of a stylesheet statement
you can remove the comments and use as is. -->
```

```
<!-- ?xml-stylesheet type ="text/xsl" href="qmf_dataset.xslt" ? -->
```

```
<DataSet xmlns="http://www.ibm.com/qmf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<!-- A schema file has been provided by the QMF product. You can find it
in the QMF SAMPLE library. Copy it to the directory where you exported
the file. If you would like to use it from a different directory you
must enter an xsi:schemaLocation statement as the next statement in this
document -->
```

```
.....
```

# Exporting data in QMF to XML format

- Metadata and data records

```

<ResultSet>
  <MetaData>
    <SourceDescription/>
    <ColumnsAmount>2</ColumnsAmount>
    <ColumnDescription id="1">
      <Name>NAME</Name>
      <Label>NAME</Label>
      <Type>varchar</Type>
      <Width>9</Width>
      <Nullable>>true</Nullable>
      <Format>plain</Format>
    </ColumnDescription>
    <ColumnDescription id="2">
      <Name>SALARY</Name>
      <Label>SALARY</Label>
      <Type>decimal</Type>
      <Scale>2</Scale>
      <Precision>7</Precision>
      <Nullable>>true</Nullable>
      <Format>plain</Format>
    </ColumnDescription>
  </MetaData>
  .....
```

```

  <Data>
    <Row id="0">
      <Cell id="1">SANDERS</Cell>
      <Cell id="2">18357.50</Cell>
    </Row>
    <Row id="1">
      <Cell id="1">PERNAL</Cell>
      <Cell id="2">18171.25</Cell>
    </Row>
    <Row id="2">
      <Cell id="1">MARENGHI</Cell>
      <Cell id="2">17506.70</Cell>
    </Row>
  </Data>
</ResultSet>
<Extensions/>
</DataSet>
```

# Exporting data in QMF to XML format

```

Sessie A - [32 x 80]
Bestand  Bewerken  Beeld  Communicatie  Acties  Venster  Help

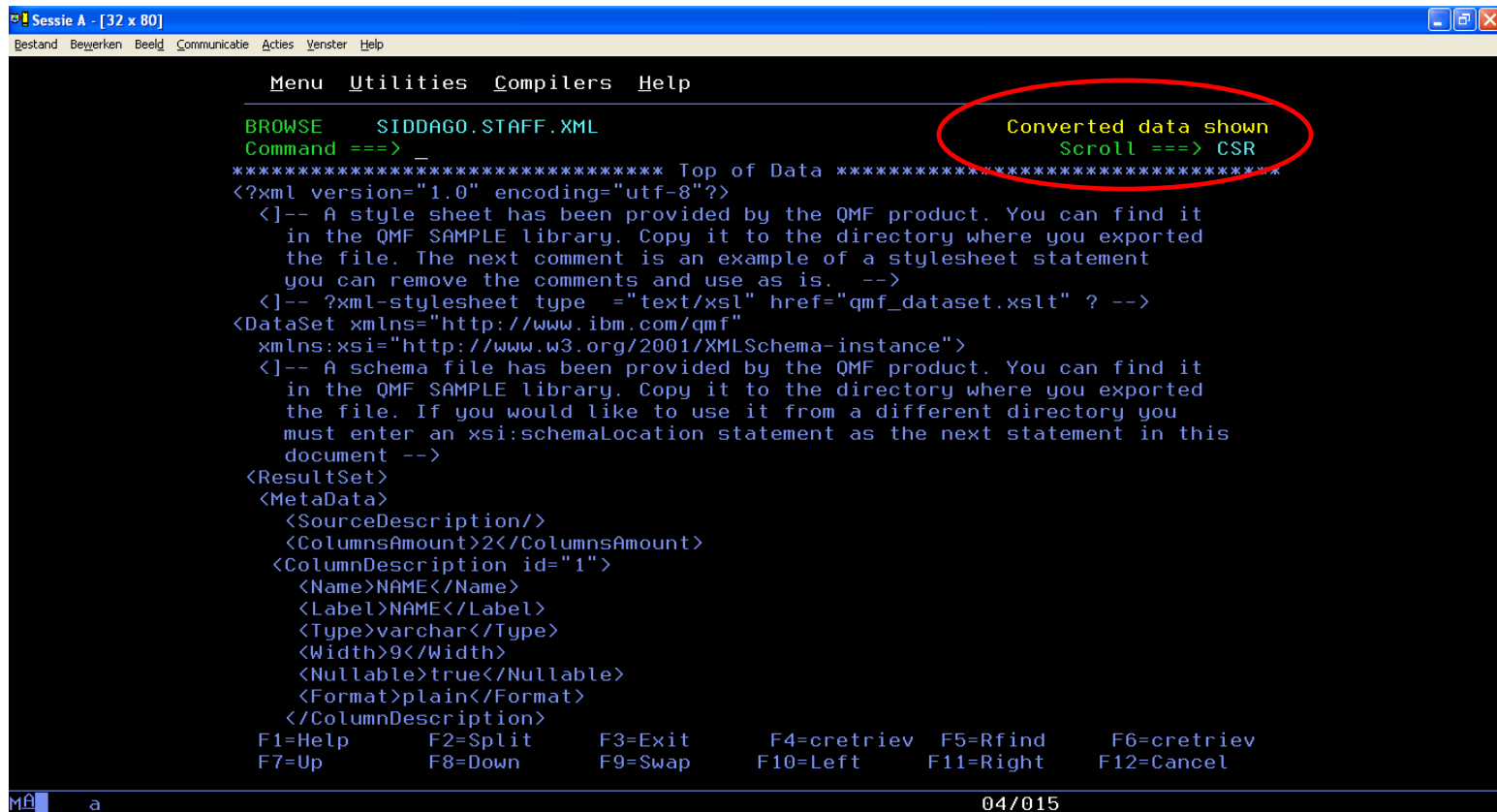
Menu  Utilities  Compilers  Help

BROWSE    SIDDAGO.STAFF.XML           Line 00000000 Col 001 080
Command ==> display utf8              Scroll ==> CSR

***** Top of Data *****
..i_%.IAEEN?>.....a>ã&ã>R..IEã
...  ÈÈ`%Á.ÈÇÁÀÈ.Ç/È.ÁÁÁ>.óÈ?INÁÁÁ.Á`ÈÇÁ.é(ã.óÈ?ÁÍÁÈ.β?Í.Á/>.ÁN>À.ÑÈ.....
...  Ñ>.ÈÇÁ.é(ã.ë (8<á.%ÑÁÈ/È`.ã?o`ÑÈ.È?.ÈÇÁ.ÁÑÈÁÁÈ?È`ÍÇÁÈÁ.`?Í.ÁÍo?ÈÈÁÁ.....
...  ÈÇÁ.ÁN%Á.èÇÁ.>ÁÍÈ.ã?_Á>È.ÑÈ./>.ÁÍ/_o%Á.?ã./ÈÈ`%ÁÈÇÁÁÈ.ÈÈ/ÉÁ_Á>È.....
...  ?Í.Á/>.ÉÁ_?ÍÁ.ÈÇÁ.Á?_Á>ÈÈ./>.Á.ÍÉÁ./È.ÑÈ.....
...  )...i_%.ÈÈ`%ÁÈÇÁÁÈ.È`oÁ...ÉÁÍÈ.ÌÈ%.ÇÉÁÑ.É_Á`À/È/ÉÁÈ.ÌÈ%È.....
..à/È/éÁÈ.Ì_>È.ÇÈÈo...IYI.Y.¿Èã...i(<èÃÇÁ_/Ñ>ÈÈ/>ÁÁ
..i_>È.ÌÈÑ..ÇÈÈo...IYI.Y.¿Èã...i(<èÃÇÁ_/Ñ>ÈÈ/>ÁÁ
...  )...ÈÁÇÁ_/ÁÑ%Á.Ç/È.ÁÁÁ>.óÈ?INÁÁÁ.Á`ÈÇÁ.é(ã.óÈ?ÁÍÁÈ.β?Í.Á/>.ÁN>À.ÑÈ.....
...  Ñ>.ÈÇÁ.é(ã.ë (8<á.%ÑÁÈ/È`.ã?o`ÑÈ.È?.ÈÇÁ.ÁÑÈÁÁÈ?È`ÍÇÁÈÁ.`?Í.ÁÍo?ÈÈÁÁ.....
...  ÈÇÁ.ÁN%Á.ñã.`?Í.Ý?ÍÁ.%Ñ.Á.È?.ÍÉÁ.ÑÈ.ÁÈ?_/_`ÑÑÁÁÁÈÁ>È.ÁNÈÁÁÈ?È`.?Í.....
...  _ÍÈÈ.Á>ÉÁÈ./>.ÌÈÑ.ÈÁÇÁ_/<?Á/ÈÑ?>.ÈÈ/ÉÁ_Á>È./È.ÈÇÁ.>ÁÍÈ.ÈÈ/ÉÁ_Á>È.Ñ>.ÈÇÑÈ....
...  Á?ÁÍ_Á>È
...  éÁÈÍ%ÈéÁÈ.....
...  (ÁÈ/à/È/
...  ë?ÍÉÁÁÁÁÈÁÈÑoÈÑ?>.....
...  ã?%í_>È_?Í>È...ã?%í_>È_?Í>È...
...  ã?%í_>àÁÈÑÈÑoÈÑ?>.ÑÁ.....
...  +/_Á.+ (á.+/_Á
...  </ÁÁ%.+ (á.</ÁÁ%
...  `oÁ.Í/ÈÁÇ/È.è`oÁ.....
...  iÑÁÈÇ...iÑÁÈÇ.....
...  +Í%%/Á%Á.ÈÈÍÁ.+Í%%/Á%Á
...  ã?È_/È.o%/Ñ>.ã?È_/È.....
...  ã?%í_>àÁÈÑÈÑoÈÑ?>.....
F1=Help  F2=Split  F3=Exit  F4=cretrieve  F5=Rfind  F6=cretrieve
F7=Up    F8=Down   F9=Swap  F10=Left   F11=Right  F12=Cancel

MÁ a                                           11/020
  
```

# Exporting data in QMF to XML format



```

Sessie A - [32 x 80]
Bestand Bewerken Beeld Communicatie Acties Venster Help

Menu Utilities Compilers Help

BROWSE      SIDDAGO.STAFF.XML
Command ==>
***** Top of Data *****
Converted data shown
Scroll ==> CSR
<?xml version="1.0" encoding="utf-8"?>
  <!-- A style sheet has been provided by the QMF product. You can find it
  in the QMF SAMPLE library. Copy it to the directory where you exported
  the file. The next comment is an example of a stylesheet statement
  you can remove the comments and use as is. -->
  <!-- ?xml-stylesheet type="text/xsl" href="qmf_dataset.xslt" ? -->
<DataSet xmlns="http://www.ibm.com/qmf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- A schema file has been provided by the QMF product. You can find it
  in the QMF SAMPLE library. Copy it to the directory where you exported
  the file. If you would like to use it from a different directory you
  must enter an xsi:schemaLocation statement as the next statement in this
  document -->
<ResultSet>
  <MetaData>
    <SourceDescription/>
    <ColumnsAmount>2</ColumnsAmount>
    <ColumnDescription id="1">
      <Name>NAME</Name>
      <Label>NAME</Label>
      <Type>varchar</Type>
      <Width>9</Width>
      <Nullable>>true</Nullable>
      <Format>plain</Format>
    </ColumnDescription>
  </MetaData>
  <Data>
    <row>
      <col>NAME</col>
      <col>NAME</col>
    </row>
  </Data>
</DataSet>

F1=Help      F2=Split    F3=Exit      F4=cretriev  F5=Rfind    F6=cretriev
F7=Up        F8=Down     F9=Swap      F10=Left    F11=Right   F12=Cancel

MA a 04/015
  
```

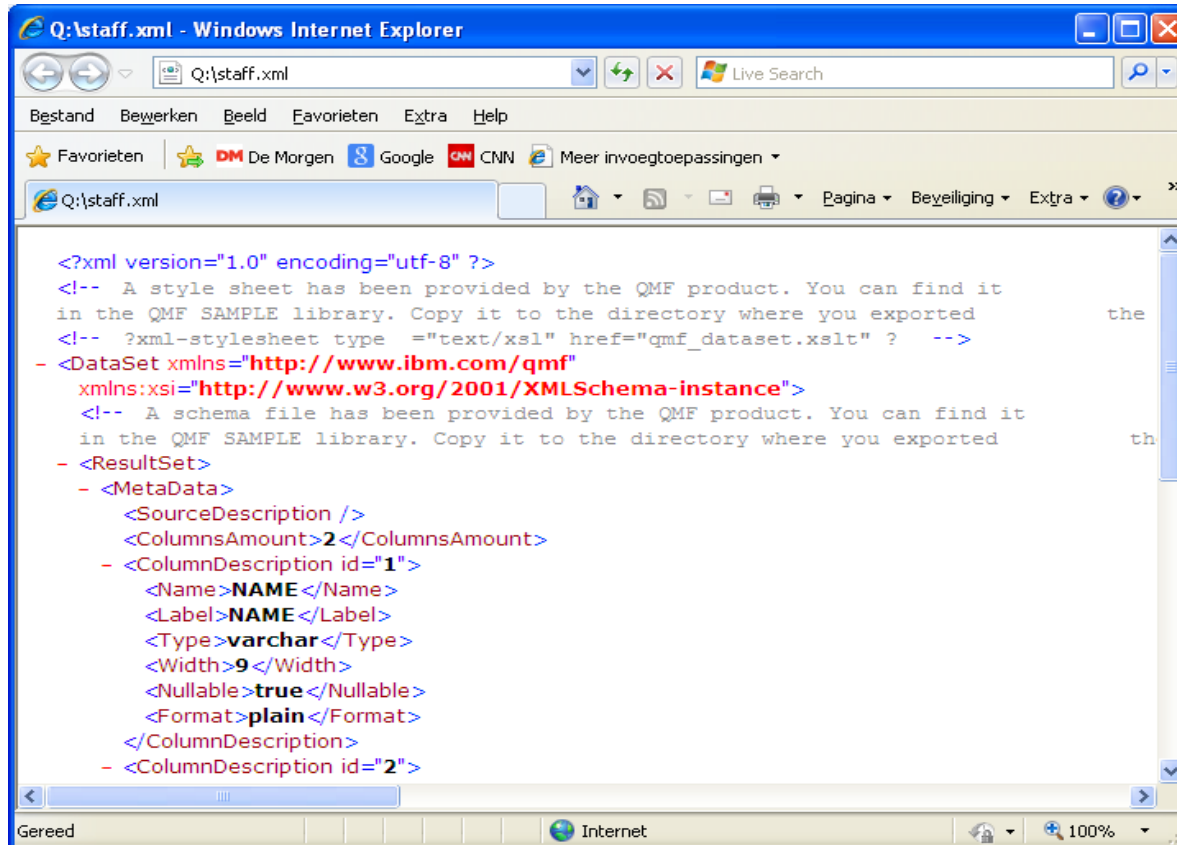


## Exporting data in QMF to XML format

- Sharing XML files between mainframe and Windows platform
  - Use FTP (sequential file or HFS file)
  - Use DFS shared directory (HFS only)
    - DFS : IBM Distributed File Service for z/OS
- Export data to DFS share:  
/data/local/clipbrd/Users/siddago/staff.xml
  - 3270 screen : Use TSO ISHELL command + ISPF  
browse/edit
  - Windows : use Notepad or Internet Explorer or other XML  
product
  - Edit .xml file : easiest is from Windows because of UTF-8  
format

# Exporting data in QMF to XML format

- Windows Internet Explorer



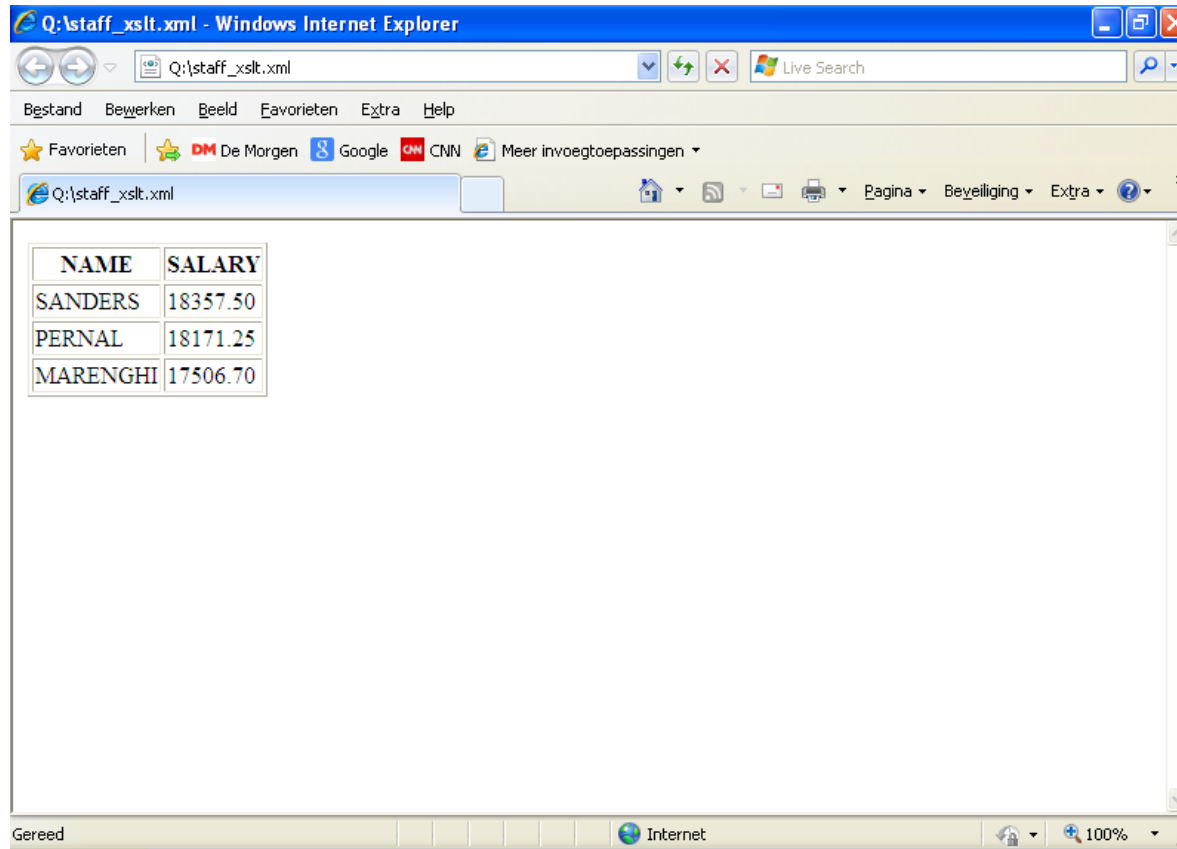
```

<?xml version="1.0" encoding="utf-8" ?>
<!-- A style sheet has been provided by the QMF product. You can find it
in the QMF SAMPLE library. Copy it to the directory where you exported
the
-->
<!-- ?xml-stylesheet type = "text/xsl" href="qmf_dataset.xslt" ? -->
- <DataSet xmlns="http://www.ibm.com/qmf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<!-- A schema file has been provided by the QMF product. You can find it
in the QMF SAMPLE library. Copy it to the directory where you exported
th
-->
- <ResultSet>
- <MetaData>
<SourceDescription />
<ColumnsAmount>2</ColumnsAmount>
- <ColumnDescription id="1">
<Name>NAME</Name>
<Label>NAME</Label>
<Type>varchar</Type>
<Width>9</Width>
<Nullable>true</Nullable>
<Format>plain</Format>
</ColumnDescription>
- <ColumnDescription id="2">

```

# Exporting data in QMF to XML format

- Use of style sheet `<?xml-stylesheet type="text/xsl" href="qmf_dataset.xslt" ?>`



# Creating and loading a table with XML columns

- Building a case :
  - DB2 meta data table with a description of 2571 existing DB2 tables from our environment in QMF exported XML format

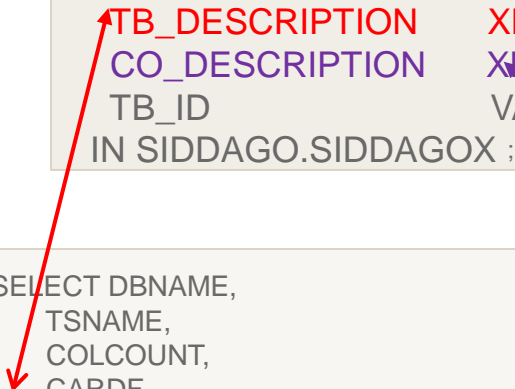
```
CREATE TABLE SIDDAGO.TABLE_XML
(TB_CREATOR      VARCHAR(128) NOT NULL,
 TB_NAME        VARCHAR(128) NOT NULL,
 TB_DESCRIPTION  XML          NOT NULL,
 CO_DESCRIPTION XML          NOT NULL,
 TB_ID          VARCHAR(20)  NOT NULL)
IN SIDDAGO.SIDDAGOX ;
```

T0000001.xml  
T0000002.xml  
.....

C0000001.xml  
C0000002.xml  
.....

```
SELECT DBNAME,
       TSNAME,
       COLCOUNT,
       CARDF,
       AUDITING,
       CREATEDTS,
       TRANSLATE(REMARKS,'A','&'),
       TRANSLATE(LABEL,'A','&')
FROM SYSIBM.SYSTABLES
for this table
```

```
SELECT COLNO,
       NAME AS COLNAME,
       COLTYPE,
       LENGTH,
       SCALE,
       NULLS,
       TRANSLATE(REMARKS,'A','&')
FROM SYSIBM.SYSCOLUMNS
for all columns of this table
```



# Creating and loading a table with XML columns

- Building a case :
  - QMF REXX SCRIPT
    - Create the T<sub>xxxxxxx.xml</sub> and C<sub>xxxxxxx.xml</sub> files
      - Run query on sysibm.systables and sysibm.syscolumns for each table
      - Export data in XML format
    - Create input SYSREC file for LOAD utility with file-reference-variables :

```
Q , APPLICANT , /data/local/clipbrd/Users/siddago/T0000001.xml , /data/local/clipbrd/Users/siddago/C0000001.xml , 0000001
Q , COMMAND_SYNONYMS , /data/local/clipbrd/Users/siddago/T0000002.xml , /data/local/clipbrd/Users/siddago/C0000002.xml , 0000002
Q , DSQ_RESERVED , /data/local/clipbrd/Users/siddago/T0000003.xml , /data/local/clipbrd/Users/siddago/C0000003.xml , 0000003
Q , ERROR_LOG , /data/local/clipbrd/Users/siddago/T0000004.xml , /data/local/clipbrd/Users/siddago/C0000004.xml , 0000004
Q , INTERVIEW , /data/local/clipbrd/Users/siddago/T0000005.xml , /data/local/clipbrd/Users/siddago/C0000005.xml , 0000005
Q , OBJECT_DATA , /data/local/clipbrd/Users/siddago/T0000006.xml , /data/local/clipbrd/Users/siddago/C0000006.xml , 0000006
Q , OBJECT_DIRECTORY , /data/local/clipbrd/Users/siddago/T0000007.xml , /data/local/clipbrd/Users/siddago/C0000007.xml , 0000007
Q , OBJECT_REMARKS , /data/local/clipbrd/Users/siddago/T0000008.xml , /data/local/clipbrd/Users/siddago/C0000008.xml , 0000008
Q , ORG , /data/local/clipbrd/Users/siddago/T0000009.xml , /data/local/clipbrd/Users/siddago/C0000009.xml , 0000009
Q , PARTS , /data/local/clipbrd/Users/siddago/T0000010.xml , /data/local/clipbrd/Users/siddago/C0000010.xml , 0000010
```

# Creating and loading a table with XML columns

- LOADING the table
  - XML files are always parsed and must contain valid XML
  - Add PRESERVE WHITESPACE if desired
  - HFS ,PDS or PDSE
  - SYSREC must be in UTF-8 (CCSID 1208 same as .xml files)
  - Provide DISCARD file to discard records with XML parsing errors  
SQLCODE = -20398

```
LOAD DATA RESUME YES LOG NO
UNICODE CCSID(01208,00000,00000)
INTO TABLE SIDDAGO.TABLE_XML
(TB_CREATOR      CHAR,
 TB_NAME         CHAR,
 TB_DESCRIPTION  CHAR CLOBF,
 CO_DESCRIPTION  CHAR CLOBF,
 TB_ID           CHAR)
FORMAT DELIMITED
SORTDEVT 3390
WORKDDN(TSYSUT1,TSORTOUT)
DISCARD DDN(TSYSDDISC)
ERRRDDN(TSYSERR)
MAPDDN(TSYSMAP)
```

# Creating and loading a table with XML columns

- Compressing the XML table spaces

- ALTER TABLESPACE  
SIDDAGO.XTAB0001 PART 1  
COMPRESS YES
- ALTER TABLESPACE  
SIDDAGO.XTAB0002 PART 1  
COMPRESS YES

```
REORG TABLESPACE SIDDAGO.XTAB0001
NOSYSREC
COPYDDN(TSYSCOPY)
SHRLEVEL REFERENCE
DRAIN_WAIT 20 RETRY 120 RETRY_DELAY 60
TIMEOUT TERM
SORTDEVT 3390
WORKDDN(TSYSUT1,TSORTOUT)
STATISTICS TABLE(ALL) INDEX(ALL)
REPORT YES
```

	.xml files	XML TS	XML TS compress ed	Compressi on ratio
T files	190 MB	50 MB	4,5 MB	89 %
C files	1200 MB	90 MB	21 MB	76 %

# Querying the XML data with SQL and XPath



# Querying the XML data with SQL and XPath

- Using XMLEXISTS function : get all tables of database DSQ1STBB

<pre> &lt;ResultSet&gt;   &lt;MetaData&gt;     &lt;SourceDescription/&gt;     &lt;ColumnsAmount&gt;8&lt;/ColumnsAmount&gt;     &lt;ColumnDescription id="1"&gt;       &lt;Name&gt;DBNAME&lt;/Name&gt;       &lt;Label&gt;DBNAME&lt;/Label&gt;       &lt;Type&gt;varchar&lt;/Type&gt;       &lt;Width&gt;24&lt;/Width&gt;       &lt;Nullable&gt;true&lt;/Nullable&gt;       &lt;Format&gt;plain&lt;/Format&gt;     &lt;/ColumnDescription&gt;     &lt;ColumnDescription id="2"&gt;       &lt;Name&gt;TSNAME&lt;/Name&gt;       &lt;Label&gt;TSNAME&lt;/Label&gt;     &lt;/ColumnDescription&gt;   &lt;/MetaData&gt;   &lt;Data&gt;     &lt;Row id="0"&gt;       &lt;Cell id="1"&gt;DSQ1STBB&lt;/Cell&gt;       &lt;Cell id="2"&gt;DSQ1STBT&lt;/Cell&gt;       &lt;Cell id="3"&gt;5&lt;/Cell&gt;       &lt;Cell id="4"&gt;1.000E+01&lt;/Cell&gt;       &lt;Cell id="5"&gt;C&lt;/Cell&gt;       &lt;Cell id="6"&gt;1990-03-12-09.51.52.580000&lt;/Cell&gt;       &lt;Cell id="7"&gt;QMF SAMPLE TABLE&lt;/Cell&gt;       &lt;Cell id="8" /&gt;     &lt;/Row&gt;   &lt;/Data&gt; &lt;/ResultSet&gt;           </pre>	Table Q.APPLICANT T0000001.xml
---	-----------------------------------

```

SELECT TB_CREATOR,TB_NAME
FROM TABLE_XML
WHERE XMLEXISTS('declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row[@id="0"]/
Cell[@id="1" and text()="DSQ1STBB"]'
PASSING TB_DESCRIPTION )
          
```

```

SELECT TB_CREATOR,TB_NAME
FROM TABLE_XML
WHERE XMLEXISTS('declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row/
Cell[./@id="0" and ./@id="1" and ./text()="DSQ1STBB"]'
PASSING TB_DESCRIPTION )
          
```

```

SELECT TB_CREATOR,TB_NAME
FROM TABLE_XML
WHERE XMLEXISTS('declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row[@id="0"]/
Cell[@id=/DataSet/ResultSet/MetaData/
ColumnDescription [Name="DBNAME"]/@id and text()
="DSQ1STBB"]' PASSING TB_DESCRIPTION)
          
```

# Querying the XML data with SQL and XPath

- Using XMLEXISTS function : get all tables of database DSQ1STBB

TB_CREATOR	TB_NAME
Q	APPLICANT
Q	INTERVIEW
Q	ORG
Q	PARTS
Q	PRODUCTS
Q	PROJECT
Q	SALES
Q	STAFF
Q	SUPPLIER

# Querying the XML data with SQL and XPath

- Using XMLQUERY function : get number of columns and cardinality of table Q.APPLICANT

```

<ResultSet>
  <MetaData>
    <SourceDescription/>
    <ColumnsAmount>8</ColumnsAmount>
    <ColumnDescription id="1">
      <Name>DBNAME</Name>
      <Label>DBNAME</Label>
      <Type>varchar</Type>
      <Width>24</Width>
      <Nullable>>true</Nullable>
      <Format>plain</Format>
    </ColumnDescription>
    <ColumnDescription id="2">
      <Name>TSNAME</Name>
      <Label>TSNAME</Label>
    </ColumnDescription>
    .....
  </MetaData>
  <Data>
    <Row id="0">
      <Cell id="1">DSQ1STBB</Cell>
      <Cell id="2">DSQ1STBT</Cell>
      <Cell id="3">5</Cell>
      <Cell id="4">1.000E+01</Cell>
      <Cell id="5">C</Cell>
      <Cell id="6">1990-03-12-09.51.52.580000</Cell>
      <Cell id="7">QMF SAMPLE TABLE</Cell>
      <Cell id="8" />
    </Row>
  </Data>
</ResultSet>
  
```

Table Q.APPLICANT  
T0000001.xml

```

SELECT XMLCAST(
XMLQUERY('declare default element namespace
"http://www.ibm.com/qmf";
$doc/DataSet/ResultSet/Data/Row[@id="0"]/
Cell[@id="3"]'
PASSING TB_DESCRIPTION AS "doc") AS INTEGER) AS
COLCOUNT
,XMLCAST(
XMLQUERY('declare default element namespace
"http://www.ibm.com/qmf";
$doc/DataSet/ResultSet/Data/Row[@id="0"]/
Cell[@id="4"]'
PASSING TB_DESCRIPTION AS "doc") AS FLOAT) AS CARDF
FROM TABLE_XML
WHERE TB_CREATOR = 'Q' AND TB_NAME = 'APPLICANT'
  
```

COLCOUNT

CARDF

5

1.000E+01

# Querying the XML data with SQL and XPath

- Using XMLTABLE function : get all columns of table Q.STAFF

```

<ResultSet>
  <MetaData>
  </MetaData>
  <Data>
    <Row id="0">
      <Cell id="1">1</Cell>
      <Cell id="2">ID</Cell>
      <Cell id="3">SMALLINT</Cell>
      <Cell id="4">2</Cell>
      <Cell id="5">0</Cell>
      <Cell id="6">N</Cell>
      <Cell id="7" />
    </Row>
    <Row id="1">
      <Cell id="1">2</Cell>
      <Cell id="2">NAME</Cell>
      <Cell id="3">VARCHAR</Cell>
      <Cell id="4">9</Cell>
      <Cell id="5">0</Cell>
      <Cell id="6">Y</Cell>
      <Cell id="7" />
    </Row>
    <Row id="2">
      <Cell id="1">3</Cell>
      <Cell id="2">DEPT</Cell>
      <Cell id="3">SMALLINT</Cell>
      <Cell id="4">2</Cell>
    </Row>
    .....
  </Data>
</ResultSet>
  
```

Table Q.STAFF  
C0002571.xml

```

SELECT X.* FROM TABLE_XML,
XMLTABLE('declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row'
PASSING CO_DESCRIPTION
COLUMNS
"COLNAME" VARCHAR(30) PATH 'declare default element
namespace
"http://www.ibm.com/qmf";
Cell[@id = /DataSet/ResultSet/MetaData/
ColumnDescription[Name="COLNAME"]/@id]' AS X
WHERE TB_CREATOR = 'Q' AND TB_NAME = 'STAFF'
  
```

COLNAME
ID
NAME
DEPT
JOB
YEARS
SALARY
COMM

# Querying the XML data with SQL and XPath

- Using XMLTABLE function : get all column meta data of table Q.STAFF

```

<ResultSet>
  <MetaData>
</MetaData>
  <Data>
    <Row id="0">
      <Cell id="1">1</Cell>
      <Cell id="2">ID</Cell>
      <Cell id="3">SMALLINT</Cell>
      <Cell id="4">2</Cell>
      <Cell id="5">0</Cell>
      <Cell id="6">N</Cell>
      <Cell id="7" />
    </Row>
    <Row id="1">
      <Cell id="1">2</Cell>
      <Cell id="2">NAME</Cell>
      <Cell id="3">VARCHAR</Cell>
      <Cell id="4">9</Cell>
      <Cell id="5">0</Cell>
      <Cell id="6">Y</Cell>
      <Cell id="7" />
    </Row>
    <Row id="2">
      <Cell id="1">3</Cell>
      <Cell id="2">DEPT</Cell>
      <Cell id="3">SMALLINT</Cell>
      <Cell id="4">2</Cell>
    </Row>
    .....
  </Data>
</ResultSet>

```

Table Q.STAFF  
C0002571.xml

```

SELECT X.* FROM TABLE_XML,
XMLTABLE(XMLNAMESPACES(DEFAULT
'http://www.ibm.com/qmf'),
'$doc/DataSet/ResultSet/Data/Row'
PASSING CO_DESCRIPTION AS "doc"
COLUMNS
"COLNO" INTEGER PATH 'Cell[@id = "1"]'
,"COLNAME" VARCHAR(30) PATH 'Cell[@id = "2"]'
,"COLTYPE" VARCHAR(10) PATH 'Cell[@id = "3"]'
,"LENGTH" INTEGER PATH 'Cell[@id = "4"]'
,"SCALE" INTEGER PATH 'Cell[@id = "5"]'
)AS X
WHERE TB_CREATOR = 'Q' AND TB_NAME = 'STAFF'

```

# Querying the XML data with SQL and XPath

- Using XMLTABLE function : get all column meta data of table Q.STAFF

COLNO	COLNAME	COLTYPE	LENGTH	SCALE
1	ID	SMALLINT	2	0
2	NAME	VARCHAR	9	0
3	DEPT	SMALLINT	2	0
4	JOB	CHAR	5	0
5	YEARS	SMALLINT	2	0
6	SALARY	DECIMAL	7	5
7	COMM	DECIMAL	7	5

# Validating XML data using an XML Schema

## Validating XML data using an XML Schema

- Validating the XML documents in our TABLE\_XML table
  - QMF provides default schema “qmf\_data.xsd” (EBCDIC)
  - Must first be registered in XSR
  - DB2 V9 : only user controlled validation
    - through DSN\_XMLVALIDATE function
      - SYSIBM.DSN\_XMLVALIDATE : SQL Builtin function

```
SELECT
SYSIBM.DSN_XMLVALIDATE( XMLSERIALIZE(TB_DESCRIPTION AS CLOB), 'SYSXSR.QMF_01) ,
SYSIBM.DSN_XMLVALIDATE( XMLSERIALIZE(CO_DESCRIPTION AS CLOB), 'SYSXSR.QMF_01')
FROM TABLE_XML
WHERE TB_ID = '0000001'
```

```
INSERT INTO TABLE_XML  SELECT TB_CREATOR,TB_NAME,
SYSIBM.DSN_XMLVALIDATE(XMLSERIALIZE(TB_DESCRIPTION AS CLOB),'SYSXSR.QMF_01),
SYSIBM.DSN_XMLVALIDATE(XMLSERIALIZE(CO_DESCRIPTION AS CLOB),'SYSXSR.QMF_01'),
TB_ID FROM TABLE_XML_TEMP
```



# Validating XML data using an XML Schema

- QMF default schema “qmf\_data.xsd” XSD = XML Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.ibm.com/qmf"
  xmlns:QMF="http://www.ibm.com/qmf"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="DataSet">
    <xs:annotation>
      <xs:documentation>
        Root element of Data Set (data
        and extensions descriptions as well)
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="QMF:ResultSet"/>
        <xs:element name="Extensions">
          <xs:annotation>
            <xs:documentation>
              Additional information about
              formatting of this Data Set
            </xs:documentation>
          </xs:annotation>

```

```
.....
<xs:sequence>
  <xs:element name="Name"
    type="xs:string"/>
  <xs:element name="Label"
    type="xs:string"
    minOccurs="0"/>
  <xs:element name="Type"
    type="QMF:DataType"/>
  <xs:choice>
    <xs:sequence>
      <xs:element name="Scale"
        type="xs:integer"/>
      <xs:element name="Precision"
        type="xs:integer"/>
    </xs:sequence>
    <xs:element name="Width"
      type="xs:integer"/>
  </xs:choice>
  <xs:element name="Nullable"
    type="QMF:Nullable"
    minOccurs="0"/>
  <xs:element name="Format"

```

## Validating XML data using an XML Schema

- Registering the QMF schema in the XSR
  - Using the XSR stored procedures
    - SYSPROC.XSR\_REGISTER (C)
    - SYSPROC.XSR\_ADDSCHEMADOC (C)
    - SYSPROC.XSR\_COMPLETE (Java)
    - SYSPROC.XSR\_REMOVE (C)
  - Using DB2 CLP commands on Windows platform
    - register xmlschema (register primary schema document)
    - add xmlschema (only if additional schema documents)
    - complete xmlschema (complete registration)
  - XSR must be installed (DB2 DSNTIJRT installation job)

# Validating XML data using an XML Schema

- Registering the QMF schema in the XSR
  - Copy QMF910.SDSQSAPE(DSQ1SCEM) to qmf\_data.xsd' on Windows with FTP

```
db2 => connect to DB2GD

Database Connection Information

Database server      = DB2 z/OS 9.1.5
SQL authorization ID = SIDDAGO
Local database alias = DB2GD

db2 => register xmlschema 'qmf_data.xsd' from file://Q:\qmf_data.xsd as SYSXSR.QMF_01
DB20000I The REGISTER XMLSCHEMA command completed successfully.
db2 => complete xmlschema SYSXSR.QMF_01
DB20000I The COMPLETE XMLSCHEMA command completed successfully.
db2 => quit
DB20000I The QUIT command completed successfully.
```

```
db2 => ? register xmlschema
REGISTER XMLSCHEMA schema-URI FROM content-URI
[WITH properties-URI] [AS relational-identifier][Sub-document-clause]
[COMPLETE [WITH schema-properties-URI][ENABLE DECOMPOSITION]]

Sub-document-clause:
  ADD document-URI FROM content-URI [WITH properties-URI] ...

db2 => ? complete xmlschema
COMPLETE XMLSCHEMA relational-identifier
[WITH schema-properties-URI][ENABLE DECOMPOSITION]
```

# Validating XML data using an XML Schema

- Enabling Java stored procedures on z/OS for XSR\_COMPLETE
  - Download “DB2 9 for z/OS XSR Setup and Troubleshooting” from web
  - One additional problem : exception 'java.lang.UnsatisfiedLinkError: DSNNVBAT

```
//V91AWLJA PROC DB2SSN=V91A,NUMTCB=2,APPLENV=WLMJAVA
//TCBNUM1 EXEC PGM=DSNX9WLM,TIME=NOLIMIT,
// PARM='&DB2SSN,&NUMTCB,&APPLENV',
// REGION=0M
//STEPLIB DD DSN=DSN910.SDSNLOAD2,DISP=SHR
//          DD DSN=DSN910.SDSNLOAD,DISP=SHR
//          DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=DSN.RUNLIB.LOAD,DISP=SHR add unauthorized lib
//JAVAENV DD DSN=WLMJAVA.JSPENV,DISP=SHR
//JSPDEBUG DD SYSOUT=A
//CEEDUMP DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

- SYSIBM.DSN\_XMLVALIDATE fails with sqlcode-904 rc00D50005

```
SETPROG LPA,ADD,MODNAME=(GXLIMODV),DSNAME=SYS1.SIEALNKE after each IPL
```

# Create and use XML indexes

# Create and use XML indexes

```
SELECT TB_CREATOR,TB_NAME
FROM TABLE_XML
WHERE XMLEXISTS('declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row[@id="0"]/
Cell[@id="1" and text()="DSQ1STBB"]'
PASSING TB_DESCRIPTION )
```

```
CREATE INDEX SIDDAGO.I_TABLE_XML_01
ON SIDDAGO.TABLE_XML
(TB_DESCRIPTION )
GENERATE KEY USING
XMLPATTERN 'declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row/
Cell'
AS SQL VARCHAR(254)
NOT PADDED
USING STOGROUP SYSDEFLT
BUFFERPOOL BP2
CLOSE YES ;
```

```
CREATE INDEX SIDDAGO.I_TABLE_XML_02
ON SIDDAGO.TABLE_XML
(TB_DESCRIPTION )
GENERATE KEY USING
XMLPATTERN 'declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row/
Cell/@id'
AS SQL VARCHAR(2)
NOT PADDED
USING STOGROUP SYSDEFLT
BUFFERPOOL BP2
CLOSE YES ;
```

```
CREATE INDEX SIDDAGO.I_TABLE_XML_03
ON SIDDAGO.TABLE_XML
(TB_DESCRIPTION )
GENERATE KEY USING
XMLPATTERN 'declare default element namespace
"http://www.ibm.com/qmf";
/DataSet/ResultSet/Data/Row/
Cell/text()'
AS SQL VARCHAR(254)
NOT PADDED
USING STOGROUP SYSDEFLT
BUFFERPOOL BP2
CLOSE YES ;
```

## Create and use XML indexes

- Eligible XML indexes : 3 conditions
  - The data type of the index and the predicate must be compatible
    - VARCHAR versus DECFLOAT
  - Text nodes must be treated the same way in the index and predicate
    - [element=value] versus [element/text()=value]
    - Ex : [Cell="DSQ1STBB"] versus [Cell/text()="DSQ1STBB"]
  - The index must “contain” the query predicate
    - Index pattern is equally or less restrictive than the predicate pattern
- Use “LEAN” indexes for best performance
- Consider the use of UNIQUE indexes
  - To force uniqueness across and within all XML documents of an XML column

## Create and use XML indexes

- Check explain PLAN\_TABLE column ACESSTYPE

ACCESS TYPE	ACCESNAME	INDEX ONLY	
M		N	Multiple index scan
DX	I_TABLE_XML_03	Y	XML index scan
DX	I_TABLE_XML_02	Y	XML index scan
DI		N	Intersection

- Check explain DSN\_STATEMNT\_TABLE column PROCSU

PROCMS	PROCSU	TOTAL_COST
2	118	6.021E+00

659 MS

338764 SU

without indexes



## Conclusion

PureXML is fun ! Try it out

Questions ?