

**IBM Information**

>>> On Demand



# Materializing the Value of DB2 9 for z/OS pureXML®

*Guogen (Gene) Zhang,  
Senior Technical Staff Member, IBM SVL  
gzhang@us.ibm.com*

*April 21, 2009*



**INNOVATE.OPTIMIZE.  
PERFORM.**

# Agenda

- Challenges in today's Information Services and XML's Roles
- pureXML in DB2 9 for z/OS makes XML consumable
- Usage Scenarios and Business Value of DB2 pureXML
- Impact to DBAs
- Performance



# Information Services' Unprecedented Challenges

- Competitive pressure, ever changing requirements
- Connected cross enterprise boundaries
- Mergers and acquisitions - integration
- New regulations
- More complex business needs
- More legacy systems
- Tough economy, produce more with less
- New services, new products, time to market – applications usually a bottleneck...



# XML is Ubiquitous

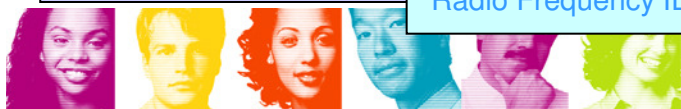
- A universal language to represent e-Business data and transactions.
  - Platform-independent, and Unicode compliant
- Data interchange and Web development
  - Web: HTML 4.0 => XHTML
  - XML for infra-structure: Web services, SOAP, SOA
  - Web 2.0 AJAX
- Documentation and content
- Database development
  - Managing XML data
  - Use XML as a flexible hierarchical data model



# Sample XML Schemas by Industry

<p><b>Banking</b>            ISO 15022, SPIFA, SPAR            IFX Standard            OFX Standard            STEP2            MISMO Standard</p> <p><b>Financial Markets</b>            FIX protocol, FIXML            Message Standard for Pos            Market Data Definition Lan            Research Information ML            Financial Products ML (FP</p> <p><b>Insurance</b>            XML for P&amp;C, Life Insuran</p> <p><b>Chemical &amp; Petroleum</b>            Chem eStandards, CyberS            PIDX Standard</p> <p><b>Healthcare</b>            HL7 Standard            DICOM Standard            SNOMED Standard</p> <p>LOINC Standard            HIPAA            SCRIPT, Mfg. Rebate Stds</p> <p>DoD XML, others</p>	<p><b>Life Sciences</b>            numerous technical standards            MIAME, MAGE, etc...            LSID, others...            HL7 Standard            DICOM Standard            CDISC Standards for ODM, LAB, ADaM, etc..</p> <p><b>Retail</b>            UCCNET including EAN-UCC            many existing standards            ePC Network &amp; standards            ARTS XML for Retail (XRETAIL)</p> <p><b>Electronics</b>            PIPs, RNIF, Business Directory, etc...            Open Access</p> <p><b>Automotive</b>            ebXML, and other B2B Standards            STAR XML</p> <p><b>Telecommunications</b>            eTOM, NGOSS, etc...            Parlay Specification</p> <p><b>Energy &amp; Utilities</b>            IEC working group14, multiple standards            CIM            Multispeak</p> <p><b>Cross Industry</b>            PDES/STEP ISO 13003            SMPI Standards            Radio Frequency ID (RFID)</p>	<p>Global Grid Forum (GGF)            Microarray Gene Expression Data Society (MGED)            Interoperable Informatics Infrastructure Consortium (I3C)            Health Level 7            Digital Imaging and Communication in Medicine            Clinical Data Interchange Standards Consortium</p> <p>Uniform Code Council, Inc            Global Commerce Initiative (GCI)            Auto-ID Center            Association of Retail Technical Standards (ARTS)            Rosetta Net</p> <p>Automotive Industry Action Group            Standards for Technology in Automotive Retail</p> <p>Telemanagement Forum (TMF)            The PARLAY Group</p> <p>International Electrotechnical Commission            Distributed Management Taskforce (DMTF.ORG)            Multispeak2.ORG</p> <p>PEDS Inc            Voluntary Interindustry Commerce Standards Association            EPCGlobal is a subsidiary of ECCnet</p>
---	--	---

It's hard to find an industry without an XML-based standard.



# Industry-specific XML Formats

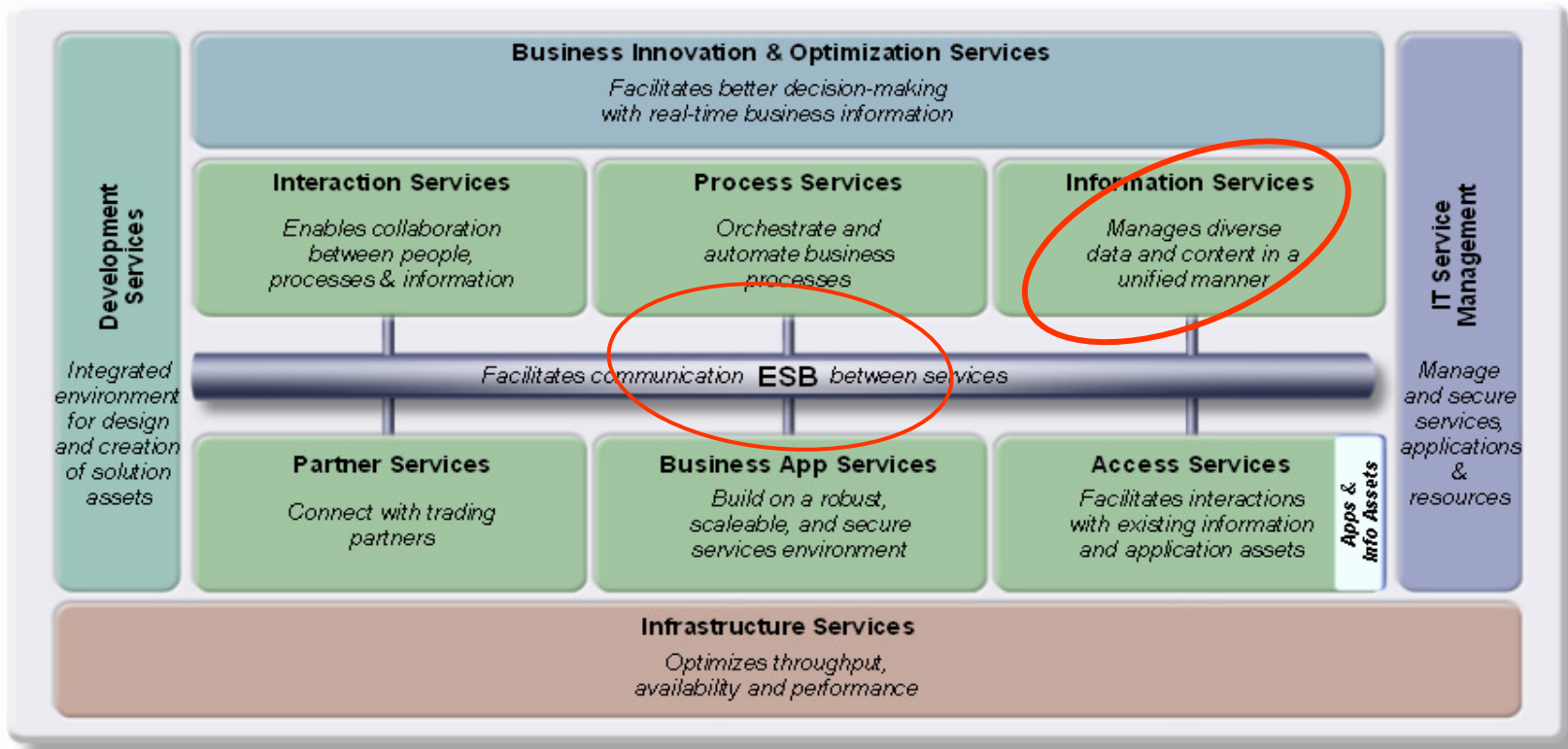
- To model complex and variable business data
- To integrate businesses and applications

	Elements + Attributes
<b>HL7 CDA 3</b> Health Level 7, Clinical Document Architecture	<b>945 + 477</b>
<b>STAR</b> Standards for Technology in Automotive Retail (OAGIS)	<b>77319 + 625</b>
...	...
<b>FpML 4.2:</b> Financial products Markup Language	<b>1867 + 196</b>
<b>FIXML 4.4:</b> Financial Information eXchange Protocol	<b>619 + 2593</b>



# SOA and Reference Model

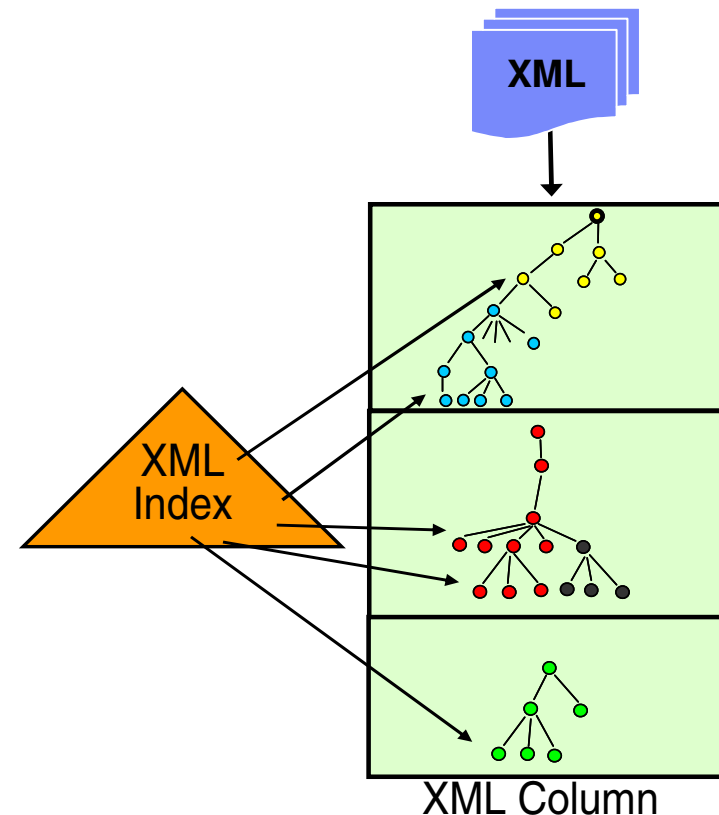
- *Improves system interoperability through open SOA standards*
- *Provides greater agility among people, processes and information*
- *Supports rapid and efficient process optimization, change and deployment*



# pureXML makes XML consumable

- Managing XML data the same way as relational data

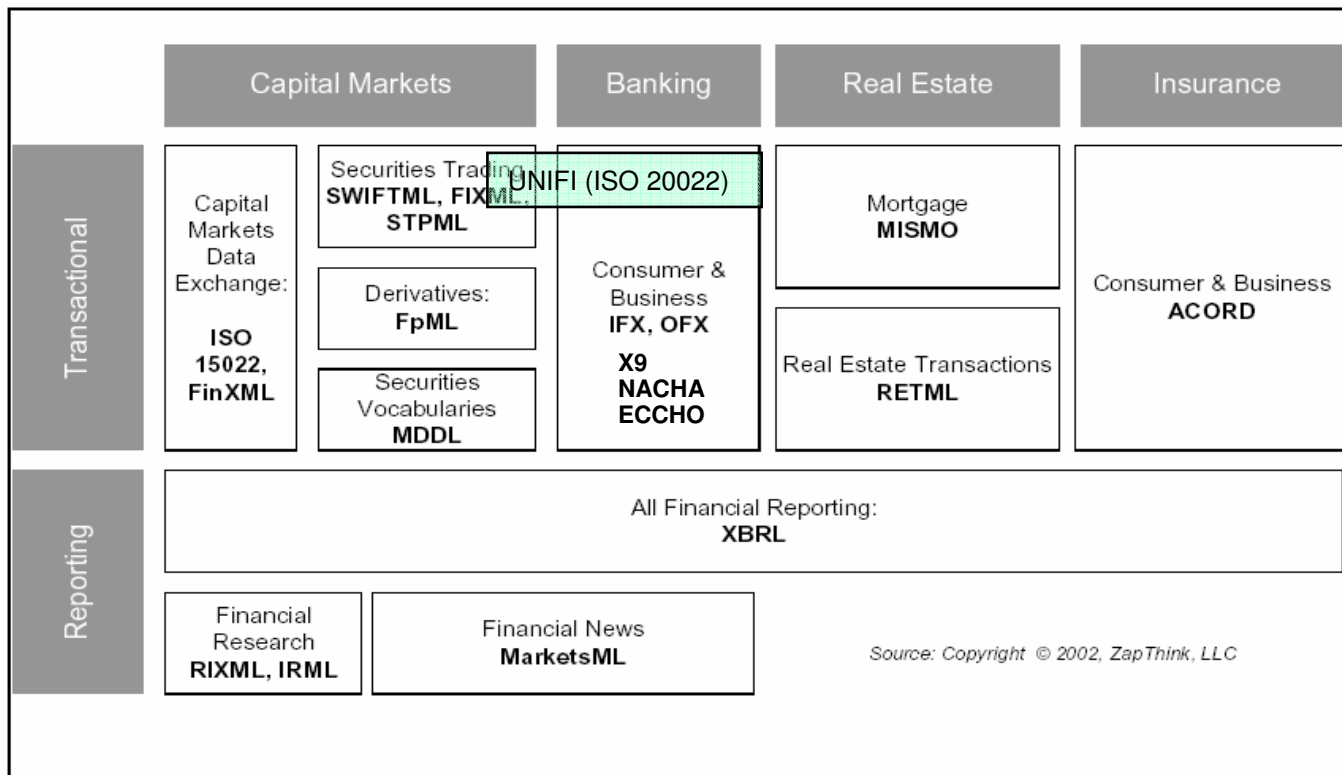
- Create tables with XML columns or alter table add XML columns
- Insert XML data, optionally validated against schemas
- Create indexes on XML data
- Efficiently search XML data
- Extract XML data
- Present XML data in relational view
- Construct XML documents from relational and XML data
- Handle XML in all the utilities and tools





# Usage Scenarios: Directly Processing XML

- XML for **data in motion** - storing what is exchanged
  - SEPA/UNIFI, ACORD, FIXML, FpML, MIMSO, XBRL, ...
  - DJXDM, HR-XML, HL7, ARTS, HIPAA, NewsML, XForms, ...
  - Insurance policy, contract, purchase order, emails, etc.

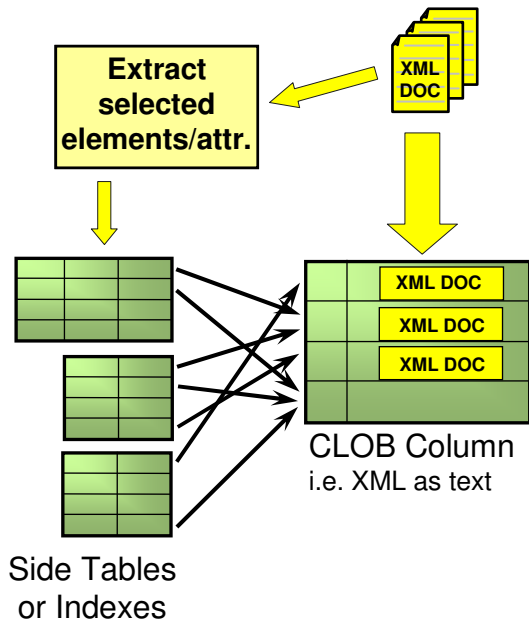


## Financial Services XML Standards



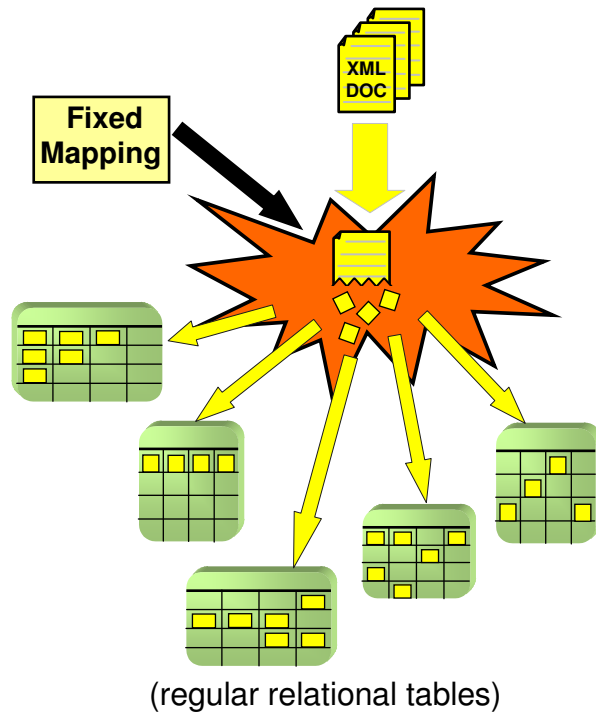
# XML Storage: Old and New

## Unstructured XML storage: XML as text



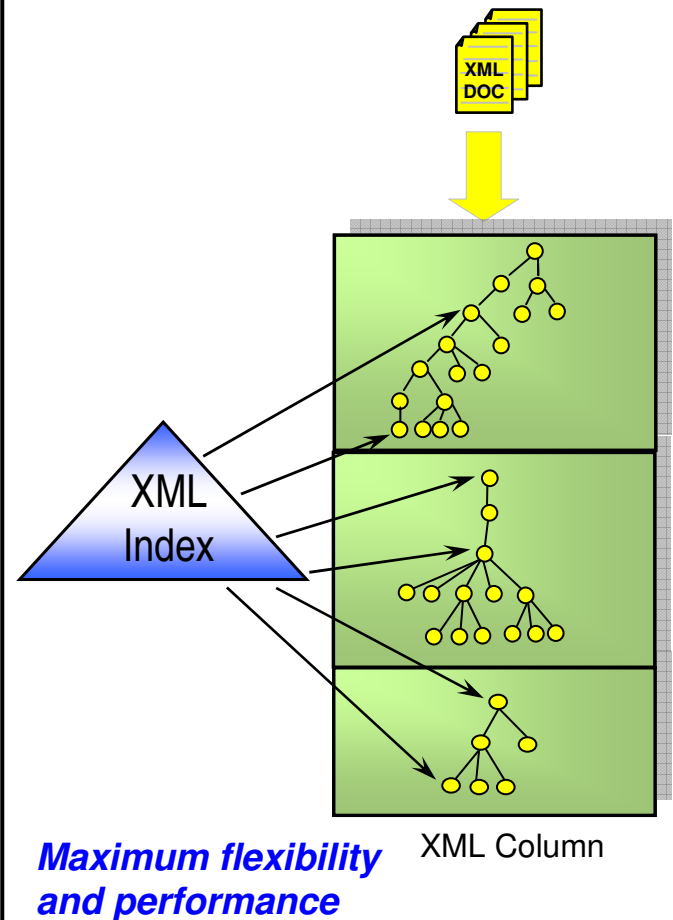
Any sub-document level access requires XML parsing – **slow**.

## Shredding: XML → Relational



Mapping prevents XML schema changes, and is often **too complex**. XML reconstruction is **slow**.

## DB2 9 pureXML: XML as XML



**Maximum flexibility and performance**



# To Represent a Purchase Order

```
<?xml version="1.0" encoding="UTF-8"?>
<ipo:purchaseOrder
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ipo="http://www.example.com/IPO" orderDate="1999-12-01">
  <shipTo exportCode="1" xsi:type="ipo:UKAddress">
    <name>Helen Zoe</name>
    <street>47 Eden Street</street>
    <city>Cambridge</city>
    <postcode>CB1 1JR</postcode>
  </shipTo>
  <billTo xsi:type="ipo:USAddress">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <items>
    <item partNum="833-AA">
      <productName>Lapis necklace</productName>
      <quantity>1</quantity>
      <USPrice>99.95</USPrice>
      <comment>Want this for the holidays!</comment>
      <shipDate>1999-12-05</shipDate>
    </item>
    <item partNum="926-AA">
      <productName>Baby Monitor</productName>
      <quantity>1</quantity>
      <USPrice>39.98</USPrice>
      <shipDate>1999-12-21</shipDate>
    </item>
  </items>
</ipo:purchaseOrder>
```

## PURCHASEORDER

POID	CID	...
------	-----	-----

## ORDERITEM

POID	PID	ITEMNO	...
------	-----	--------	-----

## CUSTOMER

CID	...
-----	-----

## ADDRESS

ADDID	...
-------	-----

## PRODUCT

PID	...
-----	-----

## PHONE

CID	PHONENUMBER
-----	-------------



# Decomposition: A schema change...

*“Employees are now allowed to have multiple phone numbers...”*

```
<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>408-463-4963</PHONE>
    <PHONE>415-010-1234</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>
```

## Requires:

- Normalization of existing data !
- Modification of the mapping
- Change of applications

## Phone

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

## Department

DEPTID	DEPTNAME
15	Sales

Costly!

## Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750



# Generate a Relational Schema for FpML



**X** In Relational: 485+ tables

**✓** In pureXML:  
*create table T(ID int, trade XML);*



## Relational Normalization not always Required or Desirable

- Normalization to eliminate redundancy to:
  - Reduce storage requirements. (used to be expensive)
  - Avoid update anomaly. (what if I don't update?)
  - Avoid inconsistency. (I need personalization)
- Fact: rigid table structures and 3NF also cause:
  - Programming overhead in normalization/denormalization.
  - Performance problems due to joins.
  - Difficulties in schema evolution.
- Consider XML normalization



# Using pureXML (1) -- XML Type and DDL

```
CREATE TABLE PURCHASEORDERS (  
    PONUMBER    VARCHAR(10) NOT NULL,  
    PODATE      DATE NOT NULL,  
    STATUS      CHAR(1),  
    XMLPO       XML)  
[ IN MYDB.MYTS ];
```

```
CREATE VIEW VALIDPURCHASEORDERS as  
    SELECT PONUMBER, PODATE, XMLPO  
        FROM PURCHASEORDERS  
        WHERE STATUS = 'A';
```

```
ALTER TABLE PURCHASEORDERS  
    ADD REVISEDXMLPO XML;
```

```
CREATE INDEX IDX1 ON PURCHASEORDERS(XMLPO) GENERATE KEYS  
    USING XMLPATTERN '//items/item/desc'  
    AS SQL VARCHAR(100);
```



## Using pureXML (2) -- DML (I/U/D)

```
EXEC SQL BEGIN DECLARE SECTION;  
    SQL TYPE IS XML AS CLOB(1M) xmlPo;  
EXEC SQL END DECLARE SECTION;
```

```
INSERT INTO PurchaseOrders VALUES ('200300001',  
    CURRENT DATE, 'A', :xmlPo);
```

```
INSERT INTO PurchaseOrders VALUES('200300002', CURRENT DATE, 'A',  
    XMLPARSE(DOCUMENT :vchar PRESERVE WHITESPACE) );
```

```
INSERT into PurchaseOrders VALUES( '200300001', CURRENT DATE, 'A',  
    XMLPARSE(DOCUMENT DSN_XMLValidate(:lobPo,'SYSXSR.myPOSchema'))  
);
```

```
UPDATE PurchaseOrders SET XMLpo = :XMPpo_new  
    WHERE ponumber = '12345';
```

```
DELETE FROM PurchaseOrders WHERE ponumber = '12345' OR  
    XMLEXISTS('/purchaseOrder/items/item[shipDate < "2000-01-01"]' PASSING  
    XMLPO);
```





## Using pureXML (3) -- DML (Select)

```
SELECT XMLpo INTO :xmlPo  
FROM PurchaseOrders  
WHERE ponumber = '200300001';
```

```
SELECT XMLQUERY('declare namespace  
ipo="http://www.example.com/IPO";  
/ipo:purchaseOrder/items/item/quantity' PASSING XMLpo)  
FROM PurchaseOrders;
```

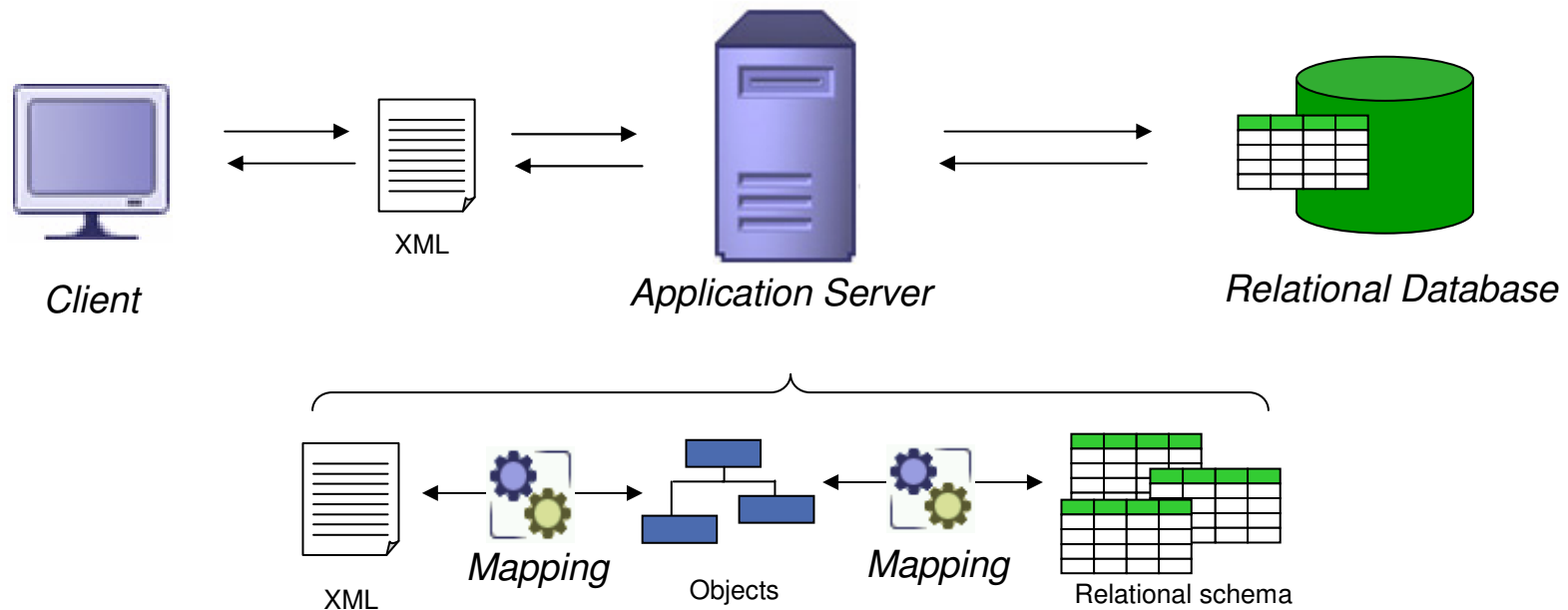
```
SELECT XMLPO  
FROM PurchaseOrders  
WHERE XMLEXISTS('//items/item[desc = "Shoe"]' PASSING XMLpo);
```

```
SELECT XMLPO  
FROM PurchaseOrders, Product  
WHERE XMLEXISTS('//items/item[desc = $n]' PASSING XMLpo,  
Product.name as "n");
```



# Usage Scenarios: XML and the Web

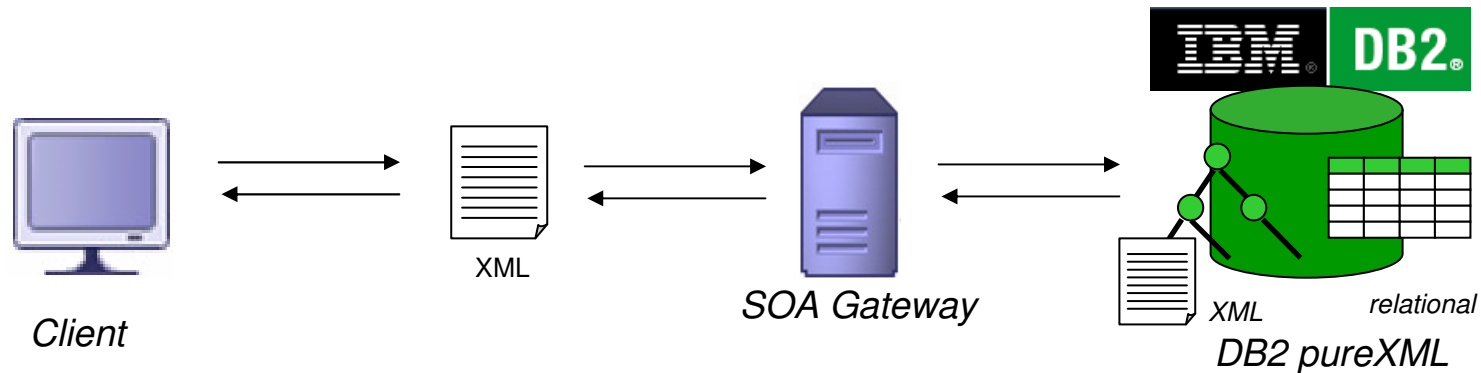
## - before DB2 pureXML



- XML to object mapping and object to relational mapping
  - Very costly
  - Complex
  - Inflexible



# DB2 pureXML and the Web



- End-to-End XML
  - No expensive object mapping
  - Pass Thru XML from/to database
- SOA-Gateway
  - Device/application to handle network protocols, security, reliability, performance
  - Easy to manage
    - Simple pre- and post-processing of XML – e.g. via XSLT or XQuery



# Generating web pages XHTML or PDF

```
<?xml version="1.0" encoding="UTF-8" ?>
<ns1:getEmployeeRecordHTMLResponse
  xmlns:ns1="http://www.myNamespace.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <EMPNO>000130</EMPNO>
    <FIRSTNAME>DELORES</FIRSTNAME>
    <MIDINIT>M</MIDINIT>
    <LASTNAME>QUINTANA</LASTNAME>
    <WORKDEPT>C01</WORKDEPT>
    <PHONENO>4578</PHONENO>
    <HIREDATE></HIREDATE>
    <JOB>ANALYST</JOB>
    <EDLEVEL>16</EDLEVEL>
    <SEX>F</SEX>
    <BIRTHDATE></BIRTHDATE>
    <SALARY>73800.00</SALARY>
    <BONUS>500.00</BONUS>
    <COMM>1904.00</COMM>
    <EMPNO>000130</EMPNO>
    <PHOTO_FORMAT>gif</PHOTO_FORMAT>
    <PICTURE>
      R01G0Dlh0gDJAPcAAJtfYJJISnotMYU2O11GR4d3eHZoaYI
    </PICTURE>
    <EMP_ROWID></EMP_ROWID>
    <EMPNO>000130</EMPNO>
    <RESUME_FORMAT>html</RESUME_FORMAT>
    <RESUME>
      Resume: Delores M. Quintana
      ...
    </RESUME>
    <EMP_ROWID></EMP_ROWID>
  </row>
</ns1:getEmployeeRecordHTMLResponse>
```

Output XSLT

The screenshot shows a Mozilla Firefox browser window displaying the output of an XSLT transformation. The page is titled "Employee Record:" and contains a table of employee details and a resume section. The table includes fields like EMPNO, FIRST NAME, MIDDLE INITIAL, LAST NAME, WORK DEPARTMENT, PHONE NUMBER, JOB, EDUCATION LEVEL, SEX, SALARY, BONUS, and COMMISSION. The resume section is titled "Resume: Delores M. Quintana" and contains personal and department information.

Employee Record:	
EMPNO:	000130
FIRST NAME:	DELORES
MIDDLE INITIAL:	M
LAST NAME:	QUINTANA
WORK DEPARTMENT:	C01
PHONE NUMBER:	4578
JOB:	ANALYST
EDUCATION LEVEL:	16
SEX:	F
SALARY:	73800.00
BONUS:	500.00
COMMISSION:	1904.00

**Resume:**

Resume: Delores M. Quintana

**Personal Information**

Address: 1150 Eglinton Ave  
Mellonville, Idaho 83725  
Phone: (208) 871-9933  
Birthdate: September 15, 1925  
Sex: Female  
Marital Status: Married  
Height: 5'2"  
Weight: 120 lbs.

**Department Information**

Employee Number: 000130  
Dept. Number: C01  
Manager: Sally Rvan  
Position: Analyst  
Phone: (208) 885-4578  
Hire Date: 1971-07-28

**Education**

1965 Math and English, B.A.  
Adelphi University



Any text format  
including XML,  
PDF



# Using pureXML


-- construct XML from relational

Can construct XHTML for web pages.

```
SELECT XMLDOCUMENT(  
  XMLELEMENT(NAME "hr:Department",  
    XMLNAMESPACES('http://example.com/hr' as "hr"),  
    XMLATTRIBUTES (e.dept AS "name" ),  
    XMLCOMMENT('names in alphabetical order'),  
    XMLAGG(XMLELEMENT(NAME "hr:emp", e.lname)  
      ORDER BY e.lname )  
  ) ) AS "dept_list"
```

FROM employees e  
GROUP BY dept;

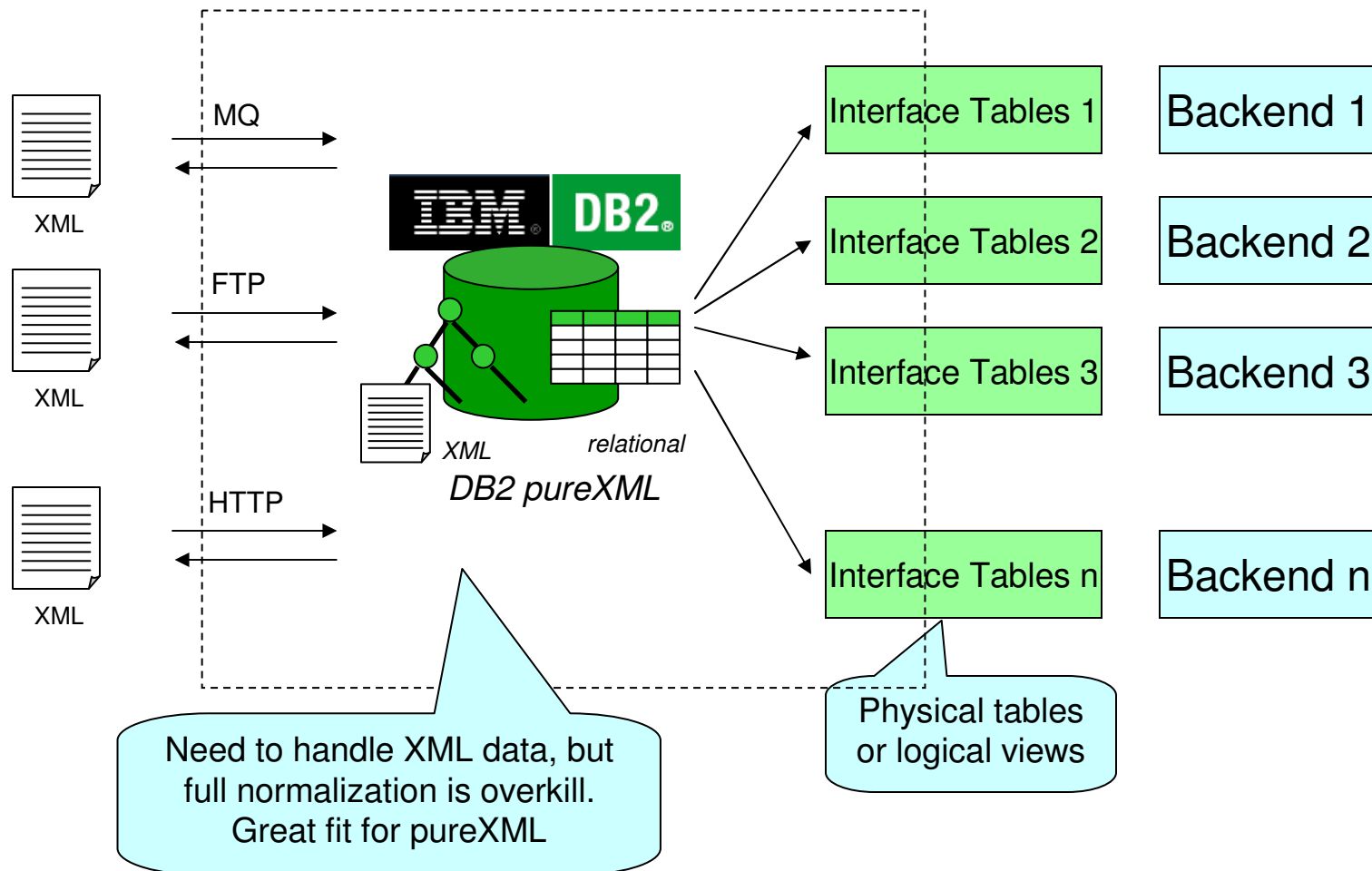
firstname	lname	dept
Sean	Lee	Shipping
Michael	Johnson	Accounting
Vicky	Oppenheimer	Shipping
Christine	Martin	Shipping



```
<?xml version="1.0" encoding="UTF-8">  
<hr:Department xmlns:hr="http://example.com/hr"  
  name="Shipping">  
  <!-- names in alphabetical order -->  
  <hr:emp>Lee</hr:emp>  
  <hr:emp>Martin</hr:emp>  
  <hr:emp>Oppenheimer</hr:emp>  
</hr:Department>
```



# Usage Scenario: XML as Front to Backend/Core Systems



# Using pureXML -- Leveraging the Power of SQL

```
CREATE VIEW ORDER_VIEW AS
SELECT PO.POID, X.*
FROM PurchaseOrders PO,
     XMLTABLE( '//item' PASSING PO.XMLPO
              COLUMNS  "orderDate"      DATE      PATH '../../@orderDate',
                        "shipTo City"    VARCHAR(20) PATH '../..//shipTo/city',
                        "shipTo State"   CHAR(2)   PATH '../..//shipTo/state',
                        "Part #"         CHAR(6)   PATH '@partnum',
                        "Product Name"   CHAR(20)  PATH 'productName',
                        "Quantity"       INTEGER   PATH 'quantity',
                        "US Price"       DECIMAL(9,2) PATH 'USPrice',
                        "Ship Date"      DATE      PATH 'shipDate',
                        "Comment"        VARCHAR(60) PATH 'comment' ) AS X;
```

**XMLTable function was delivered after V9 GA PTF UK33493**

```
SELECT "Product Name", "shipTo State",
       SUM("US Price" * "Quantity") AS TOTAL_SALE
FROM ORDER_VIEW
GROUP BY "Product Name", "shipTo State";
```

Can join with other tables.

```
SELECT "shipTo City", "shipTo State",
       RANK() OVER(ORDER BY SUM("Quantity")) AS SALES_RANK
FROM ORDER_VIEW
WHERE "Product Name" = 'Baby Monitor'
GROUP BY "shipTo State", "shipTo City"
ORDER BY SALES_RANK;
```



# Using pureXML

-- Construct new doc from existing doc

```
SELECT XMLDocument(  
  XMLElement(NAME "invoice",  
    XMLAttributes( '12345' as "invoiceNo"),  
    XMLQuery ('/purchaseOrder/billTo' PASSING xmlpo),  
    XMLElement(NAME "purchaseOrderNo",  
      PO.ponumber)  
    XMLElement(NAME "amount",  
      XMLQuery  
        ('fn:sum(/purchaseOrder/items/item/xs:decimal(USPrice))'  
        PASSING xmlpo) )  
  ) )  
FROM PurchaseOrders PO,  
WHERE PO.ponumber = '200300001';
```

```
<?xml version="1.0" encoding="utf-8" ?>  
<invoice invoiceNo = "12345">  
  <billTo country="US">  
    <name>Robert Smith</name>  
    ...  
  </billTo>  
  <purchaseOrderNo>200300001</purchaseOrderNo>  
  <amount>188.93</amount>  
</invoice>
```





# Usage Scenario: DB2 as Web Services Consumer

- Invoke light-weight SOAPHTTP UDF from SQL, consume response in the same query conveniently.
- Tooling support for DB2 in RAD
  - Create a SQL UDF specifically for a WS operation from WSDL
  - Creates 'wrapper' SQL UDF to hide XML/relational mapping.

## How much is EUR1000 worth in USD?

```
SELECT 1000 *
XMLCAST( XMLQUERY('$d//*:ConversionRateResult' PASSING
XMLPARSE (DOCUMENT
DB2XML.SOAPHTTPNV(
'http://www.webserviceX.NET/CurrencyConvertor.asmx',
'http://www.webserviceX.NET/ConversionRate',
'<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<ConversionRate xmlns="http://www.webserviceX.NET/">
<FromCurrency>EUR</FromCurrency>
<ToCurrency>USD</ToCurrency>
</ConversionRate>
</soap:Body>
</soap:Envelope>')
) AS "d") AS DECIMAL(10,5))
FROM SYSIBM.SYSDUMMYU#
```



## ... and DB2 as Web Service Provider

- Tooling integrated in IBM Data Studio
  - Query builder
    - Integrated Query Editor for SQL and XQuery
    - Graphical SQL Builder
  - Stored Procedure builder
    - Integrated debugger for Java and SQL PL procedures
    - Run and test procedures with DB2 and IDS
- Rapid generation of Web services from database operations using intelligent defaults
- No programming required
- Assembles a “ready-to-deploy” Data Web Services (DWS) Web application
- Integrated deploy and test environment



# Data Studio Tooling for Data Web Services

1. Develop Statements

2. Create Service

3. Drag 'n drop  
Service assembly

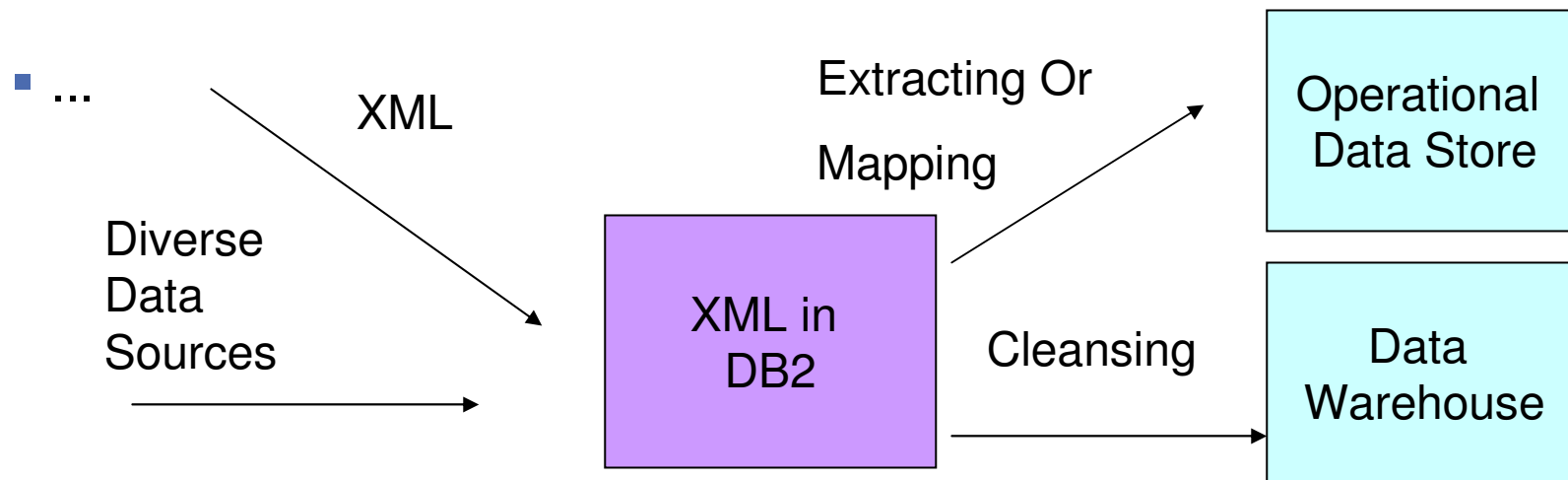
4. Deploy Service

5. Test Service



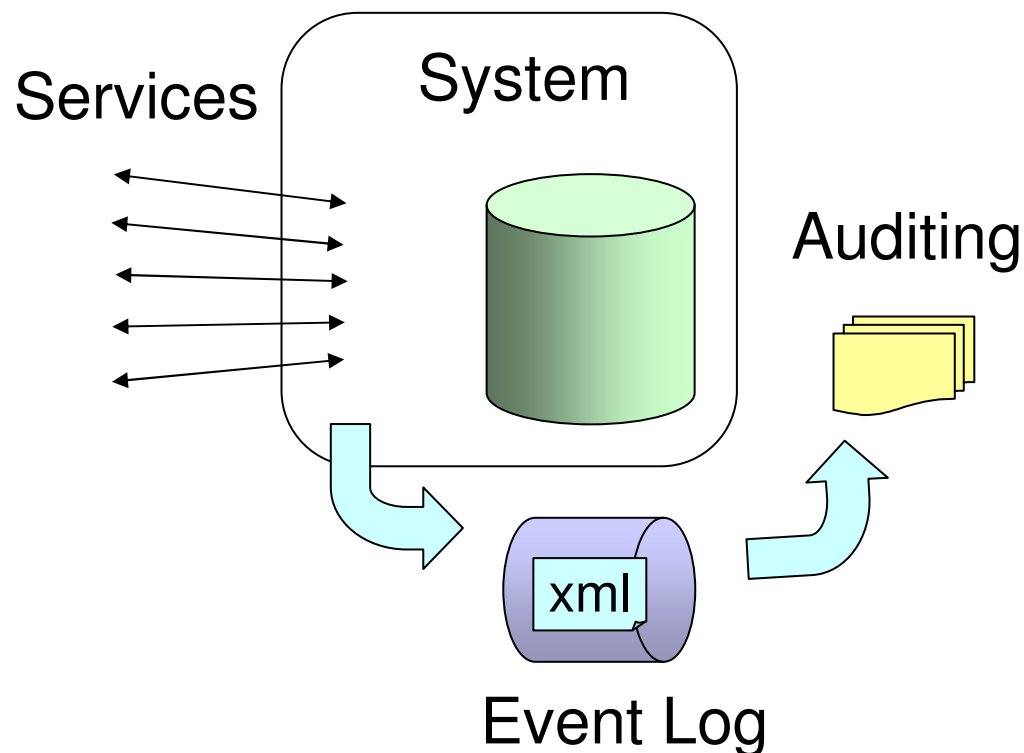
# Usage Scenarios: Business Monitoring

- Data analysis of intermittent process status
  - Immediate access to arriving data via XML tools and SQL
- Data warehouse – propagating data
- Auditing



# Usage Scenario: Event Log for Auditing

- Event log keeps records of diverse events, including:
  - TIMESTAMP, SESSIONID, USERID, REQUEST, RESULTSTATUS, SOURCE, EVENTTYPE, ACTION, RESPONSE etc.
- An XML column can contain the diverse data specific to each event type.
- Many XML indexes can be created to serve diverse query needs with good performance.



# XML Indexes: examples

CREATE INDEX IDX1 ON PURCHASEORDERS(XMLPO)

Generate Keys Using XMLPATTERN

**'/purchaseOrder/shipTo/name'**

as SQL VARCHAR(40);

CREATE INDEX IDX2 ON PURCHASEORDERS(XMLPO)

Generate Keys Using XMLPATTERN

**'/purchaseOrder/billTo/name'**

as SQL VARCHAR(40);

CREATE INDEX IDX3 ON PURCHASEORDERS(XMLPO)

Generate Keys Using XMLPATTERN

**'/purchaseOrder/items/item/@partNum'**

as SQL VARCHAR(10);

CREATE INDEX IDX4 ON PurchaseOrders(XMLPO)

Generate Keys Using XMLPATTERN

**'/purchaseOrder/items/item/desc'**

as SQL VARCHAR(100);

CREATE INDEX IDX5 ON PurchaseOrders(XMLPO)

Generate Keys Using XMLPATTERN

**'/purchaseOrder/items/item/USPrice'**

as SQL DECFLOAT;

```
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    ...
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    ...
  </billTo>
  <comment>Hurry, my lawn is going wild!</comment>
  <items>
    <item partNum="872-AA">
      <desc>Lawnmower</desc>
      <quantity>1</quantity>
      <USPrice>148.95</USPrice>
      <comment>Confirm this is electric</comment>
    </item>
    <item partNum="926-AA">
      <desc>Baby Monitor</desc>
      <quantity>1</quantity>
      <USPrice>39.98</USPrice>
      <shipDate>2003-05-21</shipDate>
    </item>
  </items>
</purchaseOrder>
```



## Usage Scenarios: Relational Schema vs. XML

- Versatile schemas and enable **end-user customizable** applications.
  - very difficult to reflect in a relational schema
- Object persistence (single XML column v.s. many tables)
  - avoiding mapping of complex XML into many relational tables
- Sparse attribute values (null v.s. absence)
  - Product features (e.g. vehicle features)



# Scenario: End-user Customizable Apps

- ISVs or large enterprise ITs need to serve diverse needs.
- Information such as product spec, customer info varies drastically.
- Use one XML column to contain the customizable data.

User View	XML Representation	XPath to refer to the values	Display
product spec unit unit price Color Size Battery Voltage Standards	<pre>&lt;product&gt;   &lt;spec&gt;     &lt;unit&gt;box&lt;/unit&gt;     &lt;unitprice&gt;8.5&lt;/unitprice&gt;     &lt;color&gt;black&lt;/color&gt;     &lt;size&gt;8 x 10&lt;/size&gt;     &lt;standards&gt;       &lt;std&gt;ISO910001&lt;/std&gt;     &lt;/standards&gt;   &lt;/spec&gt; &lt;/product&gt;</pre>	<pre>/product/spec/unit /product/spec/unitprice /product/spec/color /product/spec/size /product/spec/battery /product/spec/voltage /product/spec/standards/std</pre>	Using XSLT or XForms or your own framework based on XHTML





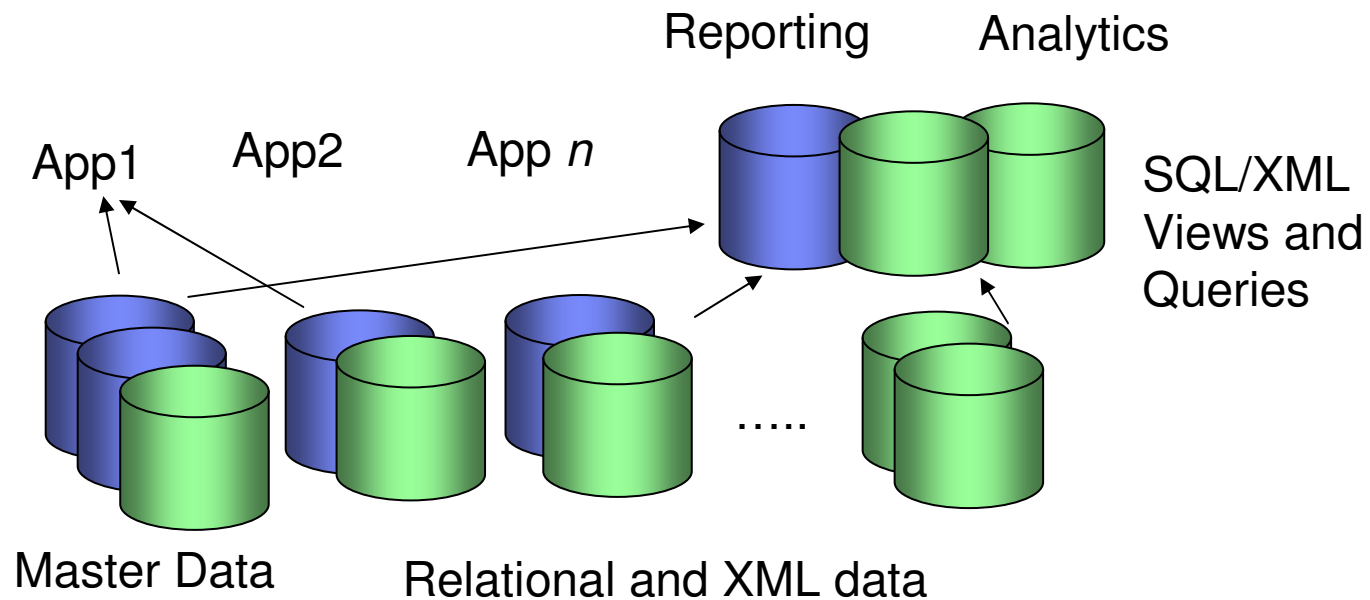
## Scenario: Object Persistence

- JPA and JAXB: Java persistence architecture, and Java Architecture for XML Binding
  - Access and process XML data without having to know XML or XML processing
- Custom-made persistence: Temenos T24 core banking product
  - `create table SCF_SECU000 (recid varchar(100), xmlrecord XML);`
  - Search on both the ID and XML columns.
  - It results in better performance.



# Hybrid relational and XML database

- XQuery and SQL/XML liberate XML data – they eliminate the need of (relational) normalization/denormalization at DB interface level, dramatically improving app dev productivity.



# Tools

- Tool choices:
  - IBM Data Studio
  - Rational Data Architect
  - Rational Application Developer
  - .NET
  - QMF
  - SPUFI
  - CLP (Command Line Processor)
- They can help to perform:
  - Schema registration, validation
  - Mapping relational to XML schema for XML generation



Data - Project1/Script1.sql - IBM Data Studio Developer

File Edit Navigate Search Project Data Run Script Window Help

Manage Licenses

Data Project Explorer

Project1 (SAMPLE:jdbc:db2://localhost:)

- SQL Scripts
  - Script1.sql
- Stored Procedures
- User-Defined Functions
- Web Services
- XML
- WSDL

Data Source Explorer

Configuration Repositories

Database Connections

- EOSDB205 (DB2 for z/OS V9 (New-F...))
  - EOSDB205
    - Databases
    - Roles
    - Schemas [Filtered]
      - DDS0075
        - Aliases
        - Auxiliary Tables
        - Dependencies
        - Jars
        - MQTs
        - Packages
        - Sequences
        - Stored Procedures
        - Synonyms
        - Tables
          - BOOK\_AUTHOR
          - BOOK\_CONTENT
          - CATALOG
          - CUST
          - IRXSXD
          - IVS\_TEST
          - MYTABLE

```

SELECT *
FROM PURCHASEORDER
WHERE XMLEXISTS('declare default element namespace "http://posample.org";
 $po/PurchaseOrder[@Status="Unshipped"]' PASSING PORDER AS "po")
  
```

Outline

- SELECT Statement

Properties Problems Error Log Model Report SQL Results Bookmarks

Type query expression here

Status	Operation	Date	Connectio...
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✓	Succeed Return All R...	3/23/09 10:...	SAMPLE
✗	Failed Script1.sql	3/23/09 11:...	SAMPLE
✓	Succeed SELECT * ...	3/23/09 11:...	SAMPLE
✓	Succeed SELECT * ...	3/23/09 11:...	SAMPLE

Status	Result1
1	POID STATUS CUSTID ORDERDATE P
1	5000 Unshipped 1002 2006-02-18

Total 1 records shown

EOSDB205 (Connected)



# DB2 z/OS XML Unique Features

- Mature optimized storage infrastructure, compact storage with optional high-ratio compression
  - As little as 0.3 of original document size
- Synergy with z/OS XML system services for highly efficient parsing, achieving unparalleled XML insertion and load performance
- Schema validation exploiting latest XML validating parser (XLXP-C);
- Efficient XML indexes for scalability; highly efficient XPath algorithm for query performance;
- Partitioned table space and data sharing support;
- Redirection of XML processing to the zIIP specialty engine from DRDA workload;
- 100% redirection of XML parsing to the zIIP or zAAP specialty engine in the near future.



# XML and Specialty Engines

- DB2 zIIP support applies to XML processing
  - DRDA zIIP
  - Utility index zIIP
- Non validation parser zAAP support
  - 100% of XMLSS under TCB mode is eligible
  - z/OS 1.9 and 1.8/1.7 with APARs
  - Observed..
    - 13% to 30% using small XML insert
    - Up to 50% using large XML load
- PK50575 for zAAP accounting support
- z/OS 1.10 zIIP support for z/OS XML SS
- z/OS 1.10 Validation parser support



# pureXML is Easy! Developers Love it!

- Easy to get XML data into database
  - Simply store it
  - No need for shredding
- Easy to get XML data out of database
  - Simply retrieve it
  - No need for parsing or re-composition
- Less integration work needed
  - Less code to write and maintain
  - Easier application development
- Hybrid operation for XML and existing relational data
  - Easy to join with relational data



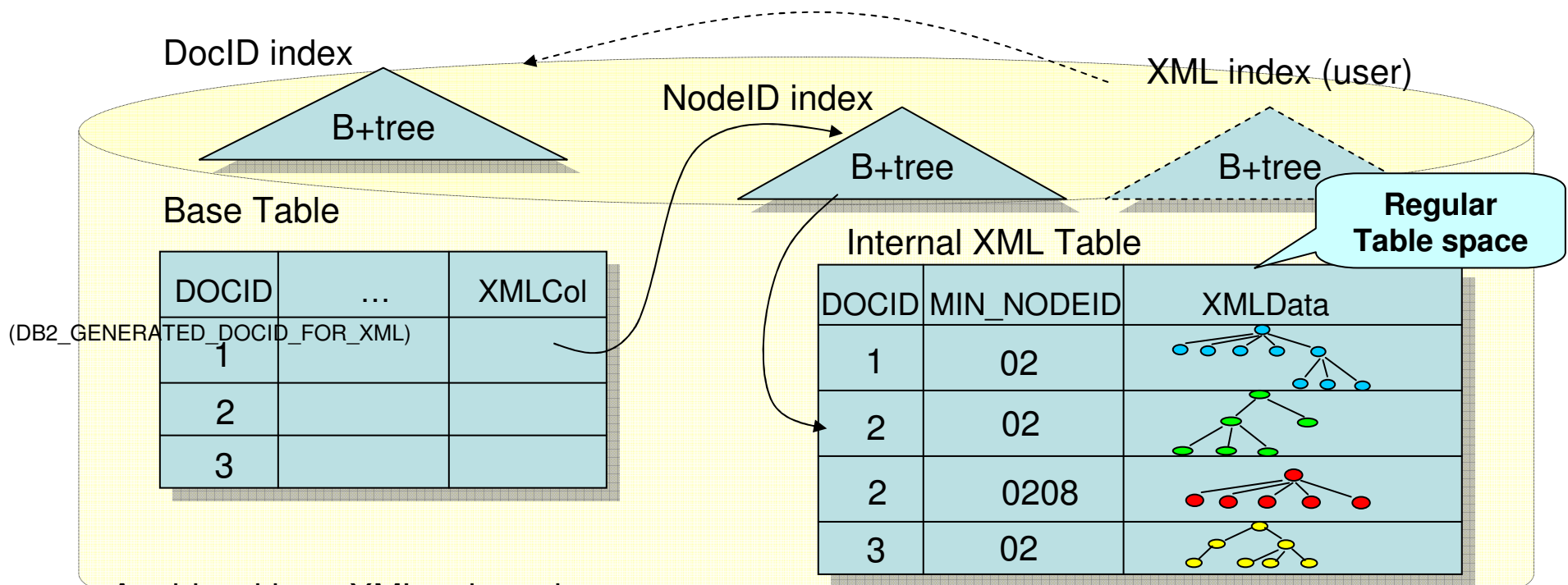
## Impact to DBAs

- Administrating new XML related objects, same infrastructure as existing objects.
- All utilities support XML objects for operations.
- Same performance tools for tuning with same principles.
- Application DBAs need to be able to support XML indexes, new valuable skills





# XML Storage on Mature Infrastructure



A table with an XML column has a DocID column, used to link from the base table to the XML table. A DocID index is used for getting to base table rows from XML indexes.

Each XMLData column is a VARBINARY, containing a subtree or a sequence of subtrees, with context path. Rows in XML table are freely movable, linked with a NodeID index.

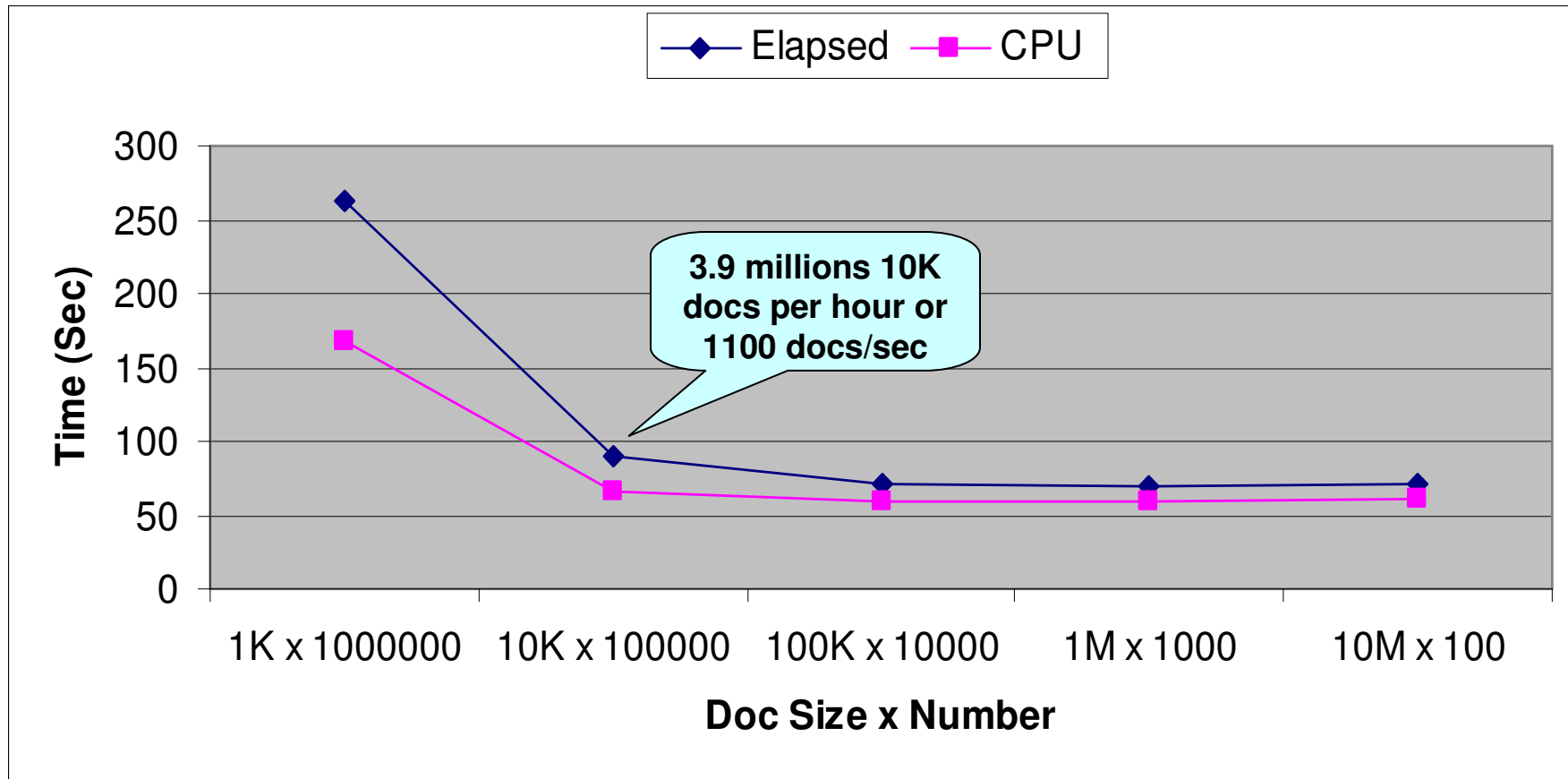


# Overview of utilities for XML

- No new special utilities for XML.
- DB2 utilities support the XML data type and the related database objects
  - The XML data type in LOAD and UNLOAD, with file reference support
  - Support for the XML table spaces and tables used to store XML column values
  - Support for auxiliary relationships used to connect base tables to XML tables
  - Support for the base table space DocID index
  - Support for the XML table space NodeID index and XML indexes
- Some restrictions remain for XML objects.



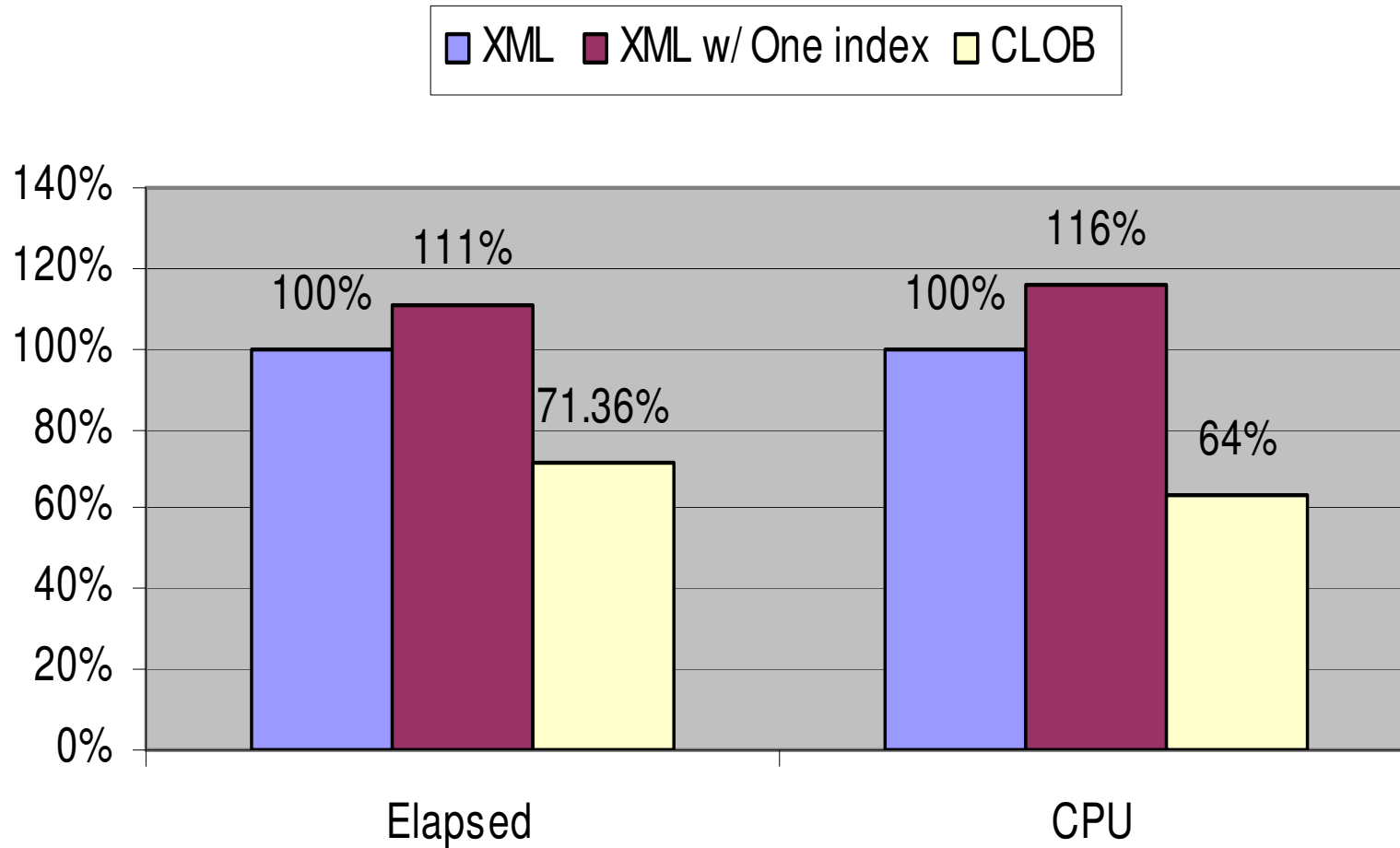
# Insert Performance (Batch)



Measurement in March 2007, z9 DS8300, Single thread, Docs in EBCDIC



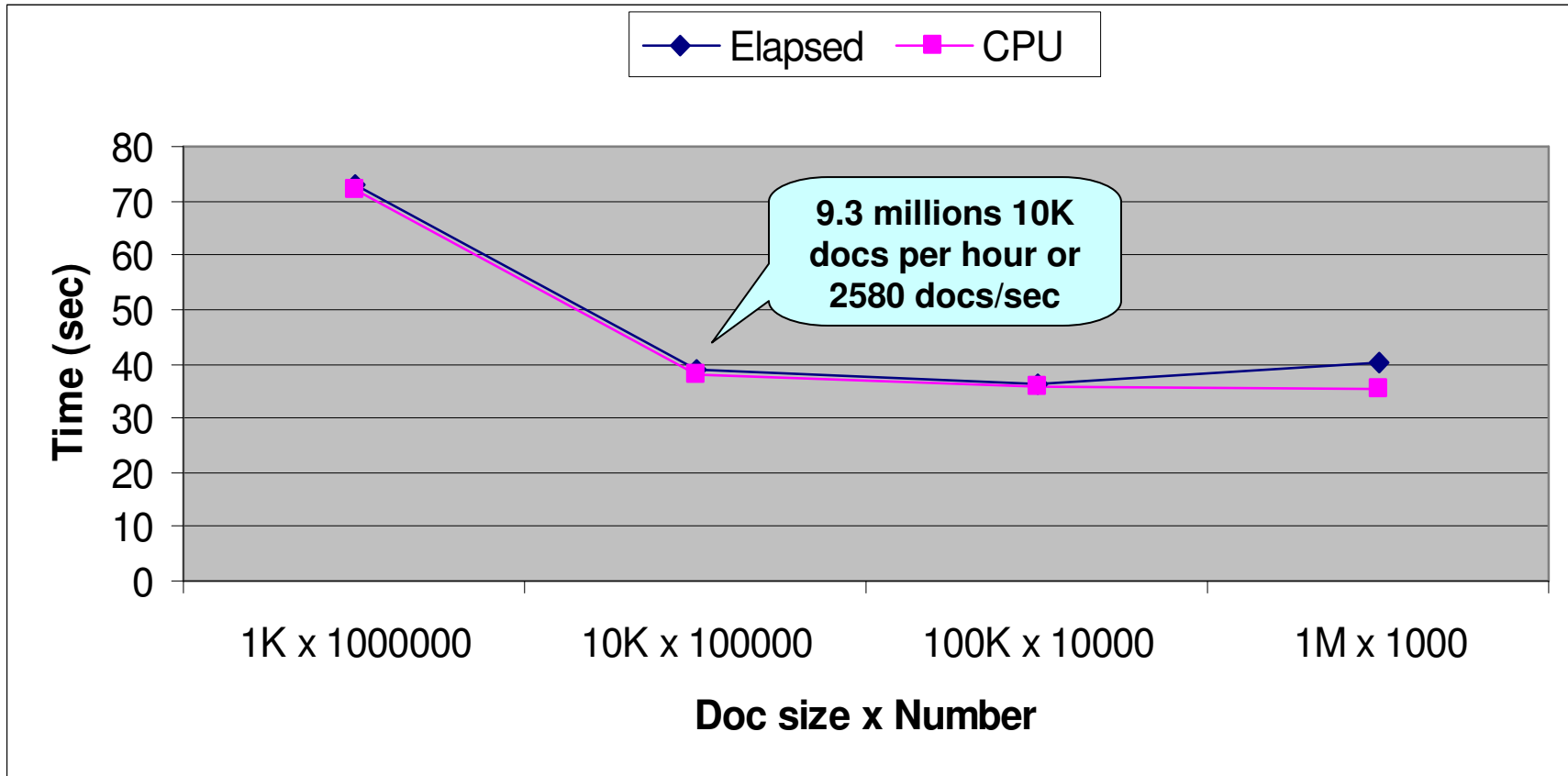
# Insert Performance – compare w/ CLOB



(average of 1K to 10M document insert performance)



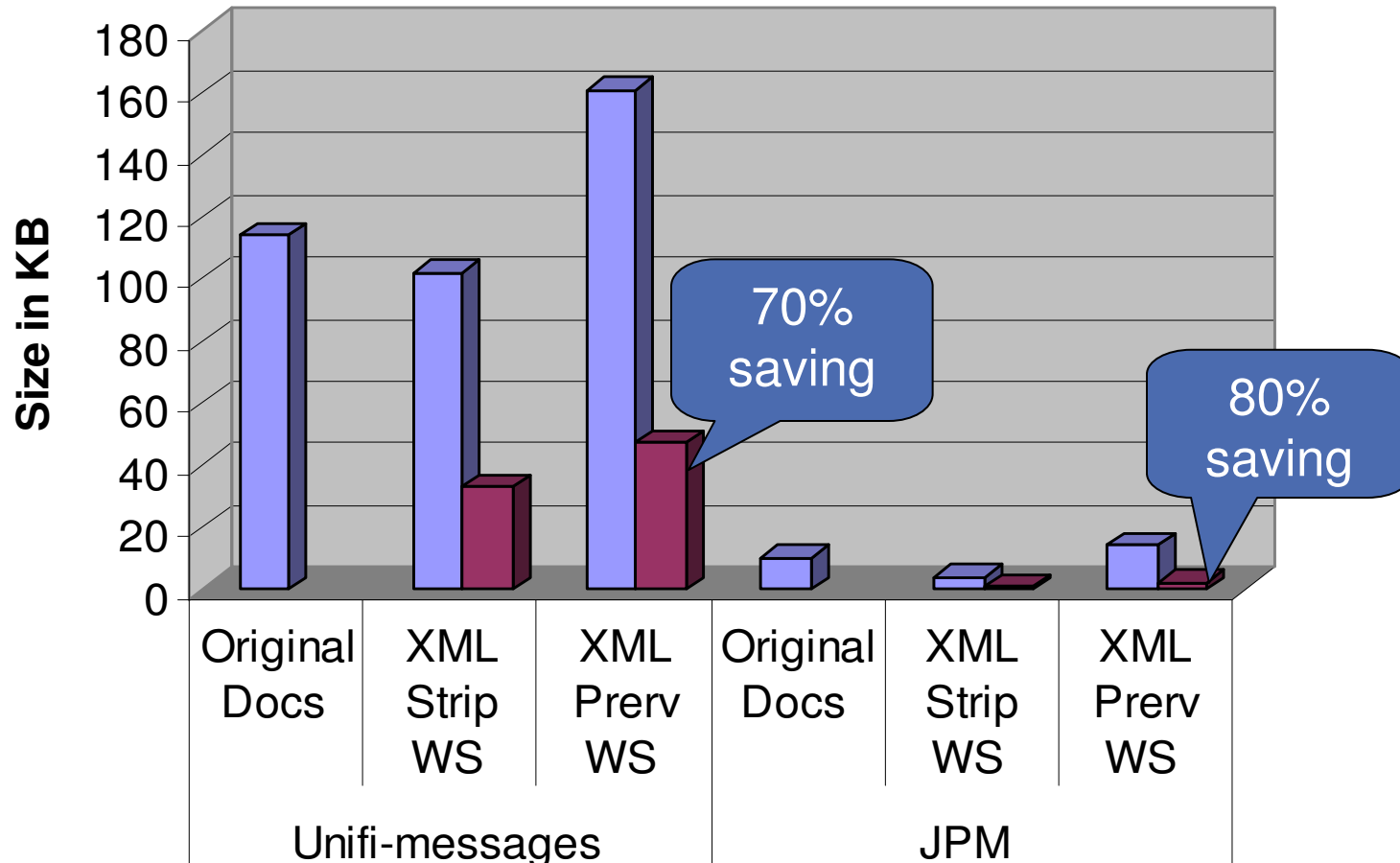
# Fetch Performance (Batch)



Measurement in March 2007, z9 DS8300, Single thread, Docs in EBCDIC



# Whitespace option and Compression

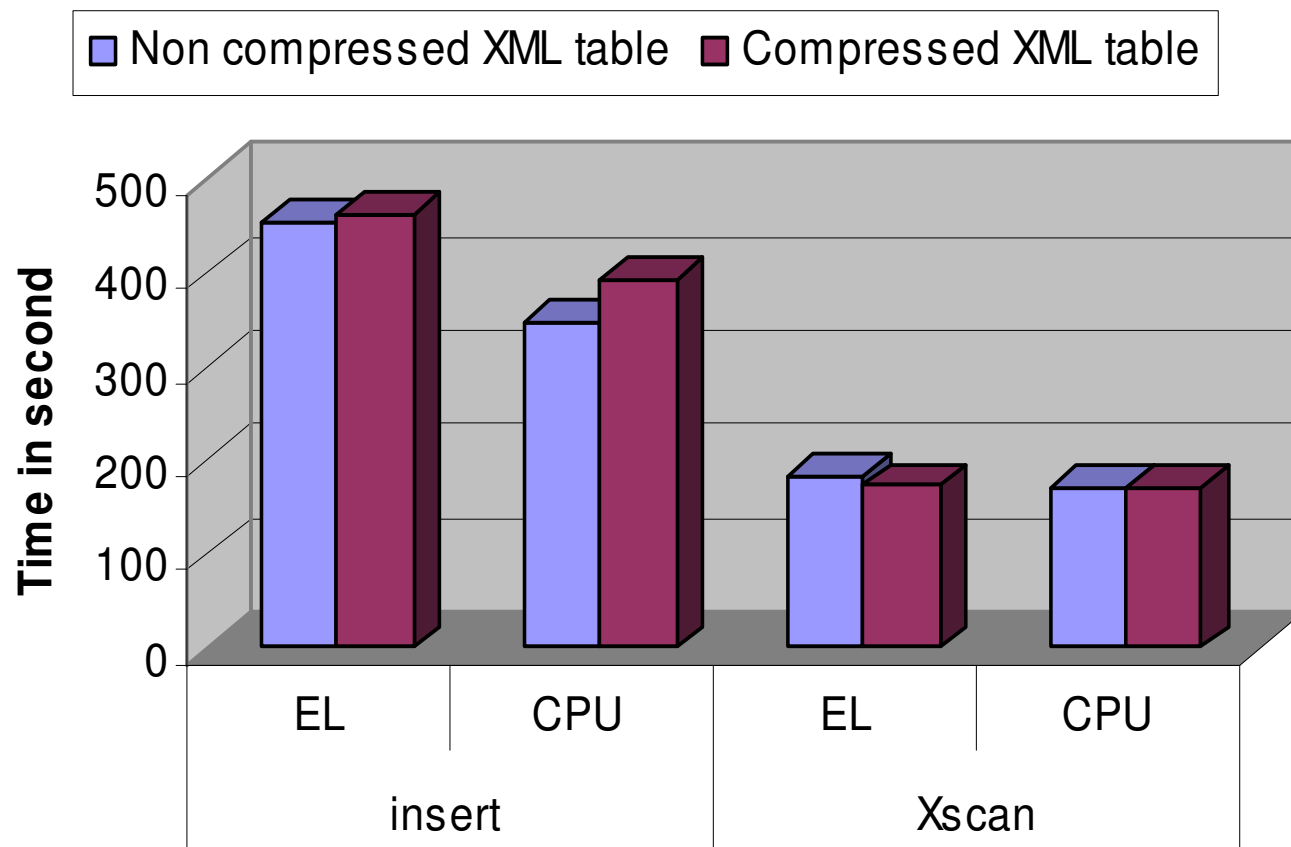


Note : UNIFI (International Standard ISO 20022 – UNiversal Financial Industry message scheme )

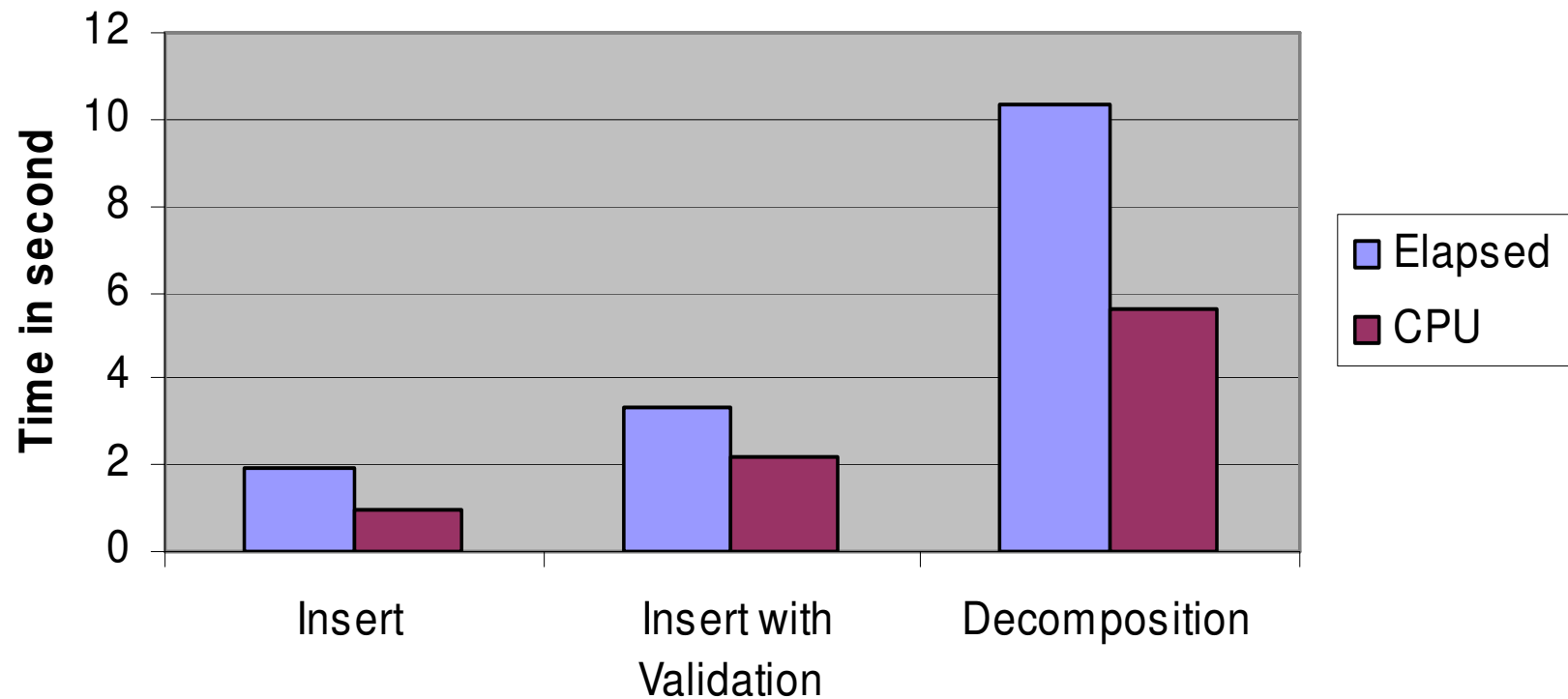
■ Non compressed ■ Compressed



# Table Compression Performance Impact



# XML Insert v.s. Validation v.s. Decomposition



4K doc Single thread 3000 repeat





# TPoX Mixed Transaction Test (1/2)

Processor	IBM System z9 Enterprise Class (z9 EC)
LPAR configuration:	3 General Purpose CPs, dedicated (no zAAP or zIIP)
Memory:	24GB memory
Storage	IBM DS8300
Operating system	z/OS Version 1.9
DB2	DB2 9 Feb 2008 PTF level
Threads	35

Transaction name	Type of transaction	Transaction Weight
Get_order (1)	Query	12
Get_security (2)	Query	12
Customer_profile (3)	Query	12
Account_summary (5)	Query	12
Get_security_price (6)	Query	12
Insert_custacc	Insert	20
Insert_order	Insert	20



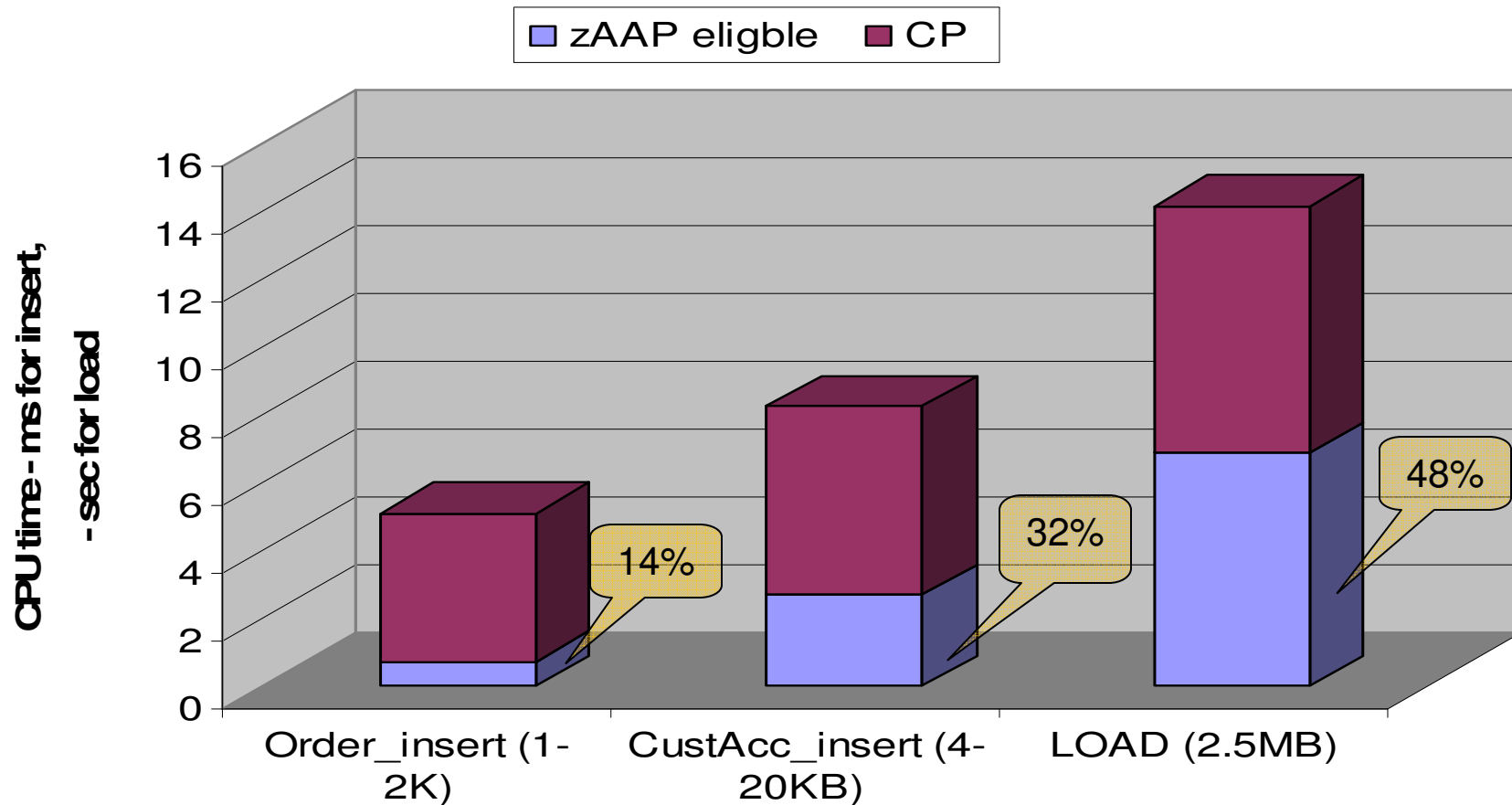
# TPoX Mixed Transaction Test (2/2)

Transaction	Average Transaction Response time
Get_order	0.03 second
Get_security	0.03 second
Customer_profile	0.03 second
Account_summary	0.03 second
Get_security_price	0.02 second
Insert_custacc	0.03 second
Insert_order	0.02 second
Transactions per second	1207 tps
CPU utilization	58.3%
Internal Throughput Rate	2068 tps
z/OS XML System Services CPU consumption	3.5%



# z/OS XML Specialty Engine Support

## XMLSS usage in insert and load



# Business Value of pureXML Summary

- ***Accelerated Application Development***
  - Reduced system and development complexity
  - Improved developer productivity
- ***Greater Business Agility***
  - Easily accommodate changes to data and schemas
  - Update applications rapidly and reduce maintenance costs
- ***Improved Business Insight***
  - Access to information in otherwise unexploited documents
  - Unprecedented application performance
- ***Consolidating converged information on System z***
  - Reduce floor space, power consumption, cooling cost
  - Consolidate information resources and reduce admin cost



# Future Directions in DB2 for z/OS

## For Informational purposes only

- Basic XQuery support in XMLQuery, XMLTable, and XMLExists
  - FLWOR, if-then-else, and constructors
  - More XML built-in types, XQuery functions
  - Support date and time related types
- XML column type modifier with XML schema
- Sub-document update
- XML support in SQL PL stored procedures and UDFs
- Schema validation inside the engine and zIIP/zAAP eligible
- Performance enhancements, including binary XML
- XSLT support
- ...

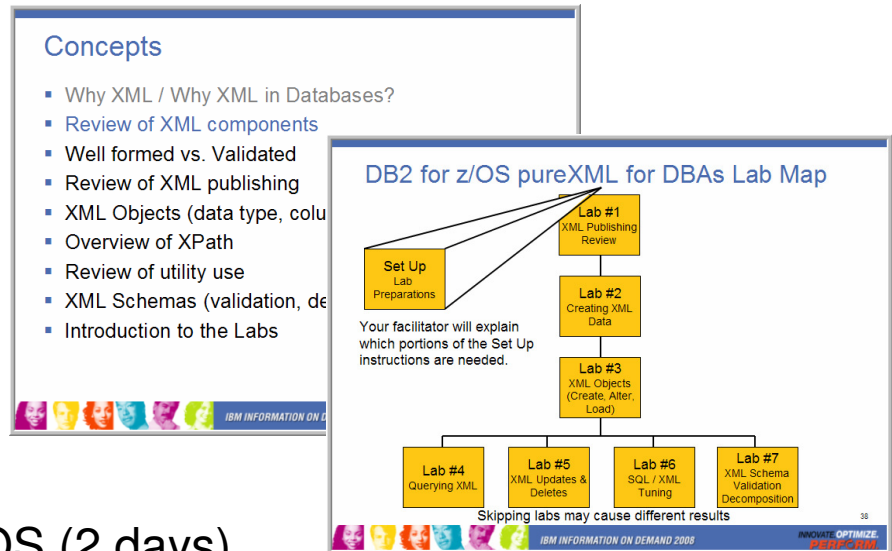


# Proof of Technology and Course Offerings

- PoT: on-site or near-site offering (no charge)
  - Lecture + Lab: 5-6 hours.

- IBM Courses

- CV120:  
Query XML Data with DB2 9 for z/OS (2 days)  
[http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course\\_description&courseCode=CV120](http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course_description&courseCode=CV120)
- 3L12: Query XML Data with DB2 9 for z/OS – Instructor Led Online
- CV260:  
Manage XML Data with DB2 9 for z/OS (3 days)  
[http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course\\_description&courseCode=cv260](http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course_description&courseCode=cv260)
- 3L26: Manage XML Data with DB2 9 for z/OS – Instructor Led Online
- <http://ibm.com/training> – select “Training search - global” or Search



# Other DB2 pureXML Resources

## Team Blog (z/OS heavy)

[www.ibm.com/developerworks/blogs/page/purexml](http://www.ibm.com/developerworks/blogs/page/purexml)

## Wiki

[www.ibm.com/developerworks/wikis/display/db2xml/](http://www.ibm.com/developerworks/wikis/display/db2xml/)

DB2 for z/OS pureXML wiki:

<http://www.ibm.com/developerworks/wikis/display/db2xml/DB2+for+zOS+pureXML>

## Forum

[www.ibm.com/developerworks/forums/forum.jspa?forumID=1423](http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1423)

## Article and whitepaper

DB2 with pureXML delivers faster answers at a lower cost

[http://www.applix.com/newsletter/it/st\\_090330\\_04.html](http://www.applix.com/newsletter/it/st_090330_04.html)

IBM pureXML for SOA: Unlocking the business value of information

[http://www-01.ibm.com/software/sw-library/en\\_US/detail/V157629M94375T43.html](http://www-01.ibm.com/software/sw-library/en_US/detail/V157629M94375T43.html)

## Web site

[www.ibm.com/software/data/db2/xml/](http://www.ibm.com/software/data/db2/xml/)

## ChannelDB2 User Group

[www.channeldb2.com/group/pureXML](http://www.channeldb2.com/group/pureXML)

## DB2 9 for z/OS XML Guide

[http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.xml/db2z\\_xml.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.xml/db2z_xml.htm)



# Important Disclaimer

**THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.**

**WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.**

**IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.**

**IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.**

**NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:**

- **CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR**
- **ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.**

