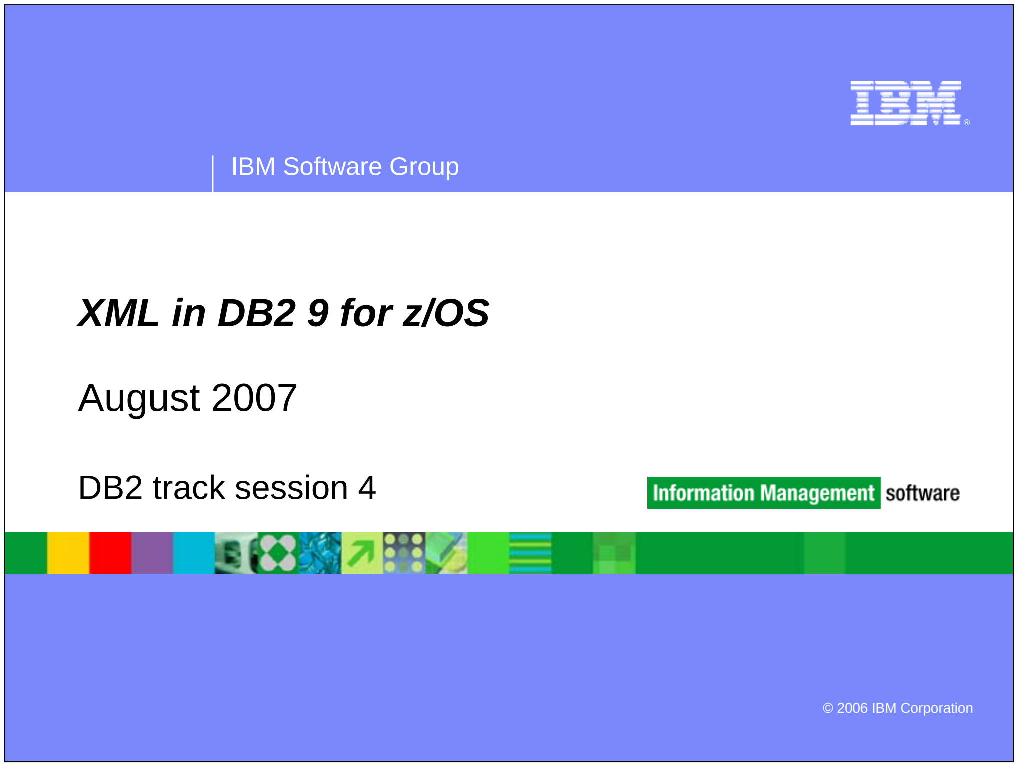


IBM Software Group

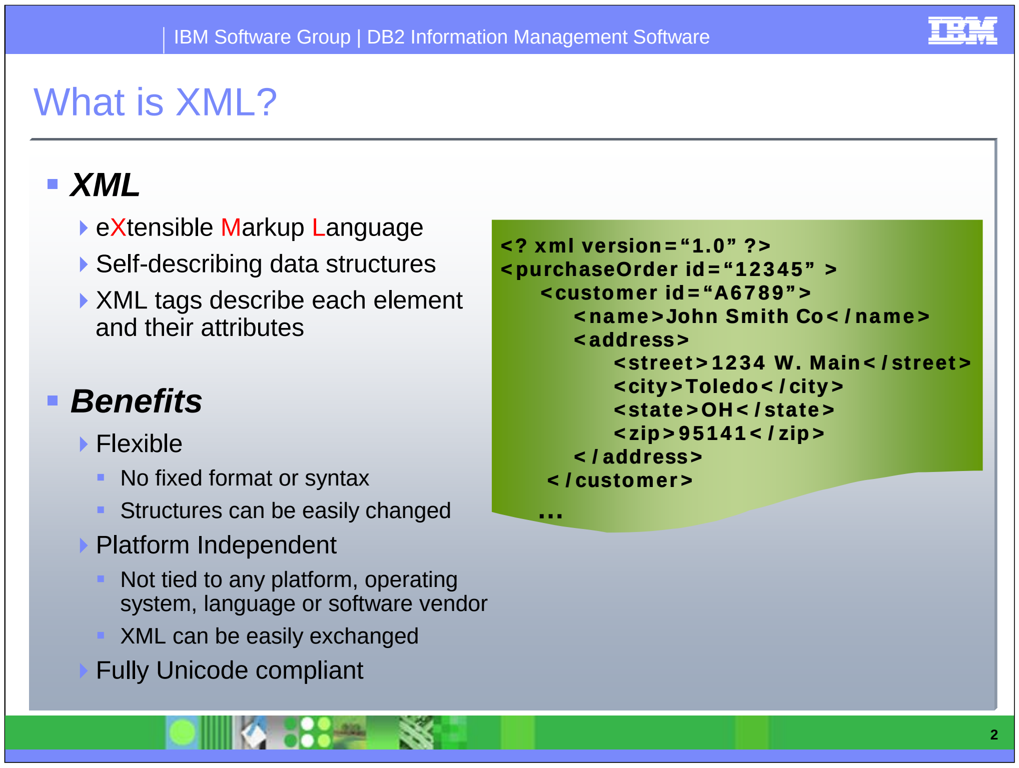
XML in DB2 9 for z/OS

August 2007

DB2 track session 4

Information Management software

DB2 for z/OS provided an XML extender in DB2 V7, then added XML publishing functions in DB2 V8. DB2 9 provides extensive changes for pureXML^{tm}. In this session we explain how to use XML with DB2 for z/OS and provide different usage scenarios including how to leverage XML data in an SOA.

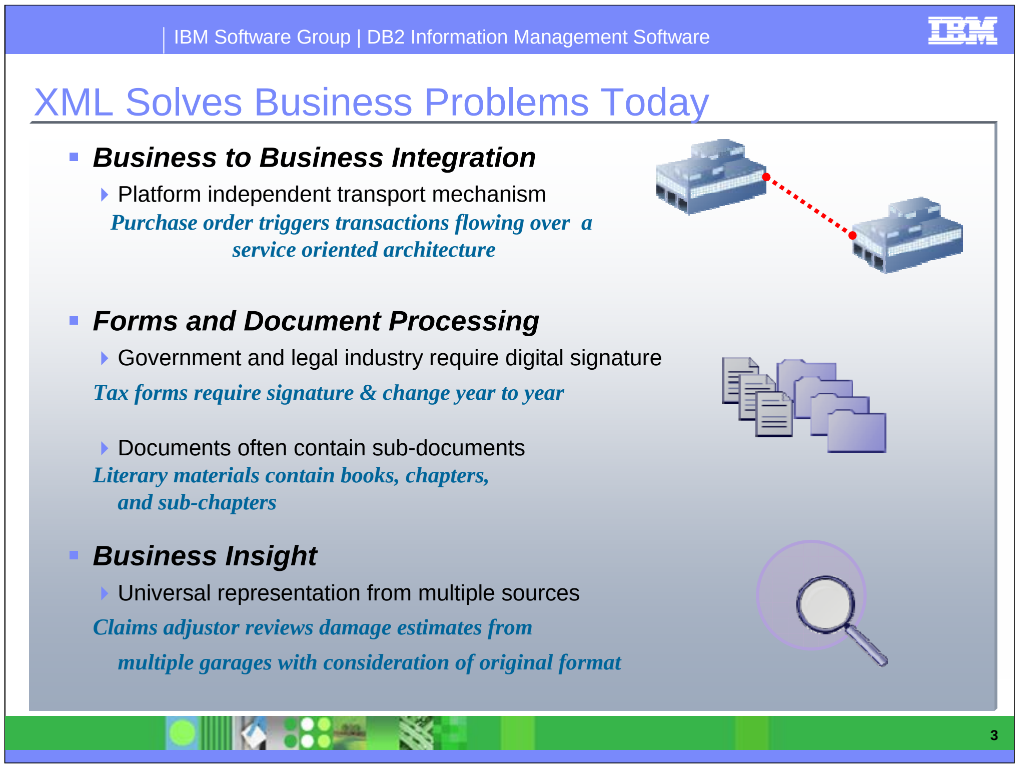


What is XML?

- ***XML***
  - eXtensible Markup Language
  - Self-describing data structures
  - XML tags describe each element and their attributes

```
<? xml version="1.0" ?>
<purchaseOrder id="12345" >
     <customer id="A6789">
          <name>John Smith Co</name>
          <address>
               <street>1234 W. Main</street>
               <city>Toledo</city>
               <state>OH</state>
               <zip>95141</zip>
          </address>
     </customer>
     ...
```

- ***Benefits***
  - Flexible
    - No fixed format or syntax
    - Structures can be easily changed
  - Platform Independent
    - Not tied to any platform, operating system, language or software vendor
    - XML can be easily exchanged
  - Fully Unicode compliant



XML Solves Business Problems Today

- ***Business to Business Integration***
  - Platform independent transport mechanism

    Purchase order triggers transactions flowing over a service oriented architecture
- ***Forms and Document Processing***
  - Government and legal industry require digital signature

    Tax forms require signature & change year to year
  - Documents often contain sub-documents

    Literary materials contain books, chapters, and sub-chapters
- ***Business Insight***
  - Universal representation from multiple sources

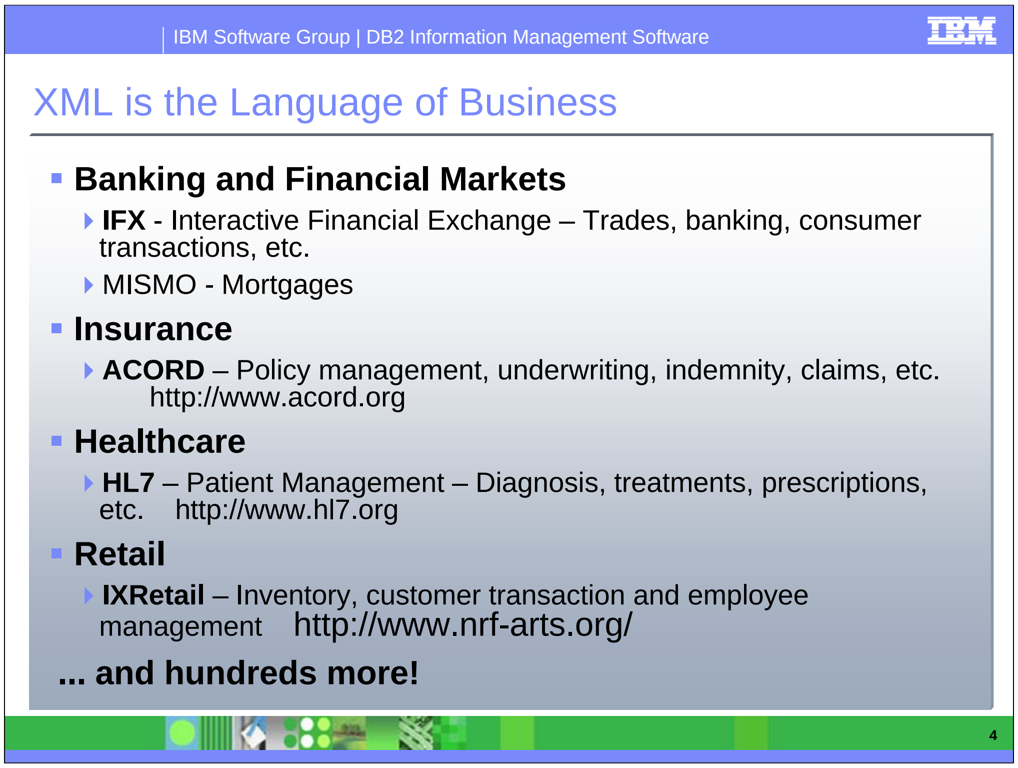
    Claims adjustor reviews damage estimates from multiple garages with consideration of original format

Business to Business – Messaging can be used to transport data from one business process to another. Often 1 message can spawn several transactions down the line. The universal format means all applications can interpret the data and just chooses the information relevant to that application.

Document Management – XML is perfect for documents as it can; maintain hierachies, can be easily changed, and retain any digital signatures. Take for example tax forms, which change from year to year. 2004 documents can be retained and simply appended to for 2005 changes.

Business Intelligence – BI reports can simply choose the data they need form multiple formats. Again, the concept of self-describing elements allow the BI user to simply obtain the information they need by use of the XML tags.

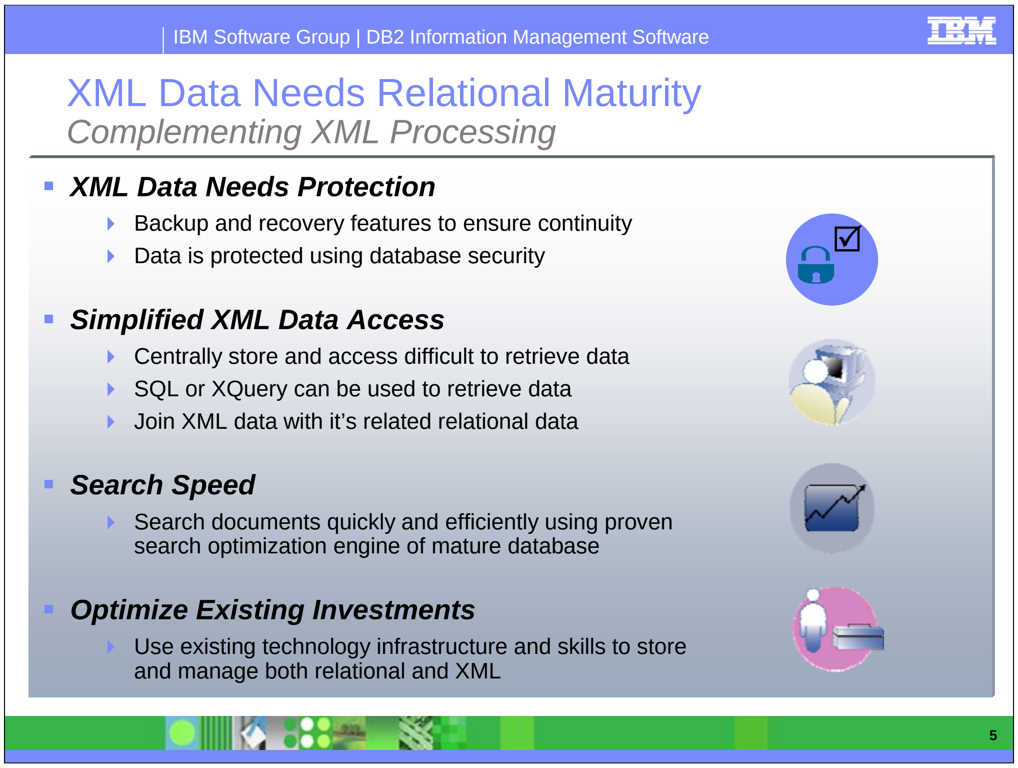
http://www.ibm.com/developerworks/db2/library/techarticle/dm-0705malaika/



XML is the Language of Business

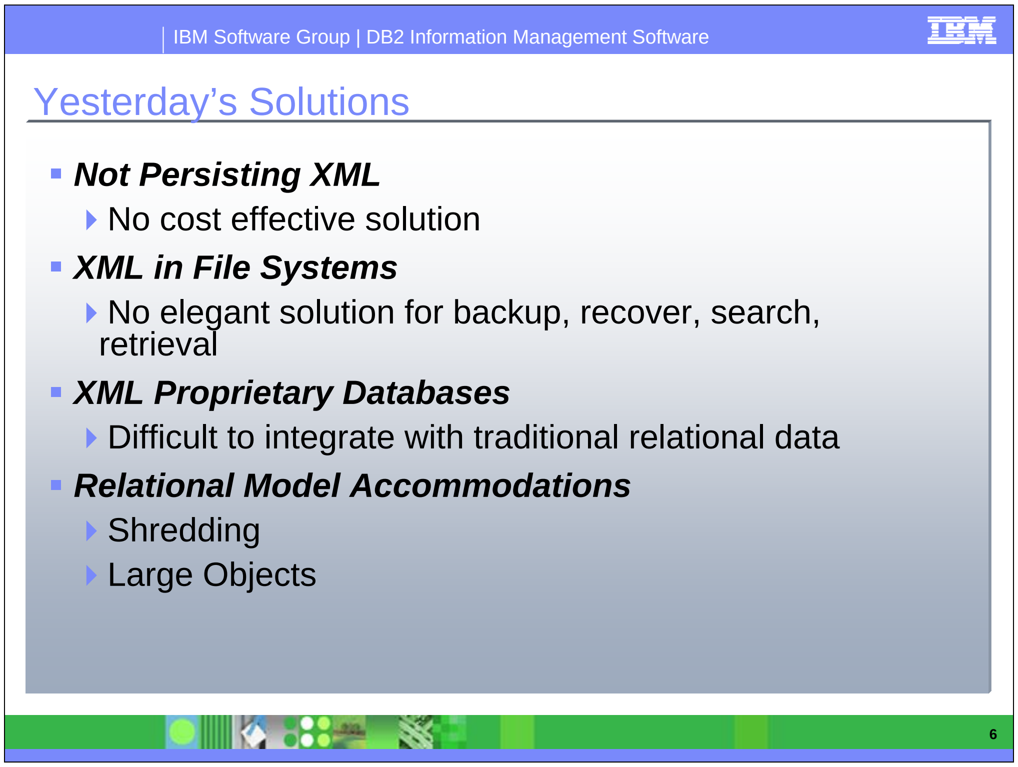
- **Banking and Financial Markets**
  - **IFX** - Interactive Financial Exchange – Trades, banking, consumer transactions, etc.
  - MISMO - Mortgages
- **Insurance**
  - **ACORD** – Policy management, underwriting, indemnity, claims, etc. http://www.acord.org
- **Healthcare**
  - **HL7** – Patient Management – Diagnosis, treatments, prescriptions, etc. http://www.hl7.org
- **Retail**
  - **IXRetail** – Inventory, customer transaction and employee management http://www.nrf-arts.org/

... and hundreds more!



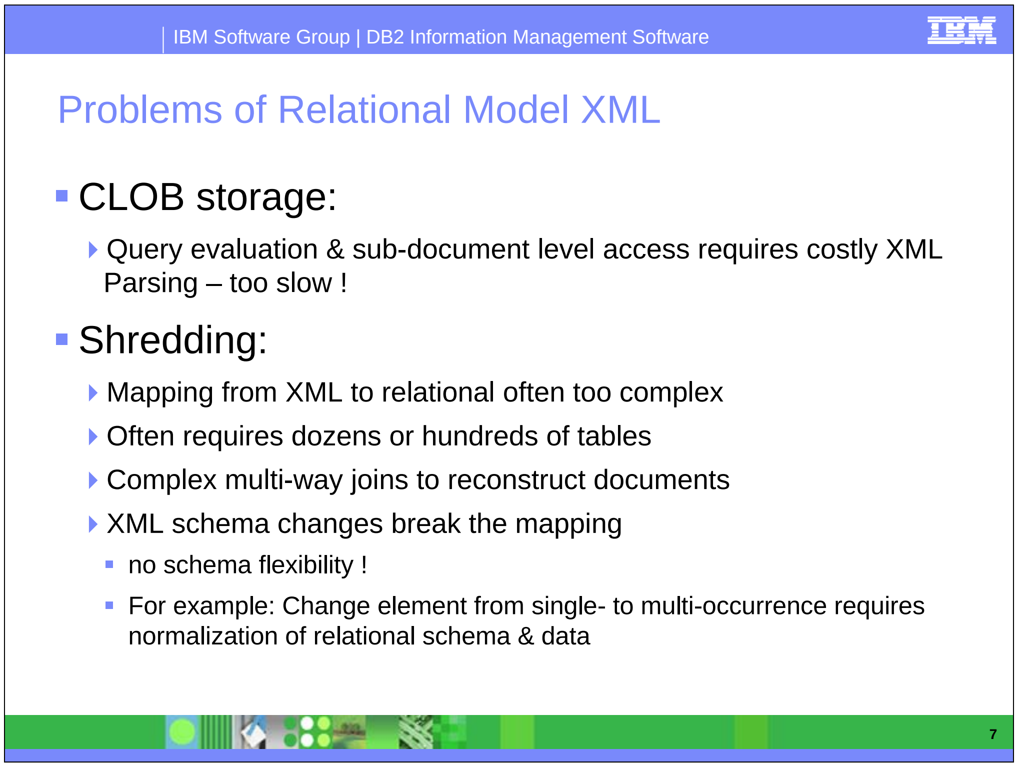
XML Data Needs Relational Maturity
Complementing XML Processing

- ***XML Data Needs Protection***
  - Backup and recovery features to ensure continuity
  - Data is protected using database security
- ***Simplified XML Data Access***
  - Centrally store and access difficult to retrieve data
  - SQL or XQuery can be used to retrieve data
  - Join XML data with it’s related relational data
- ***Search Speed***
  - Search documents quickly and efficiently using proven search optimization engine of mature database
- ***Optimize Existing Investments***
  - Use existing technology infrastructure and skills to store and manage both relational and XML



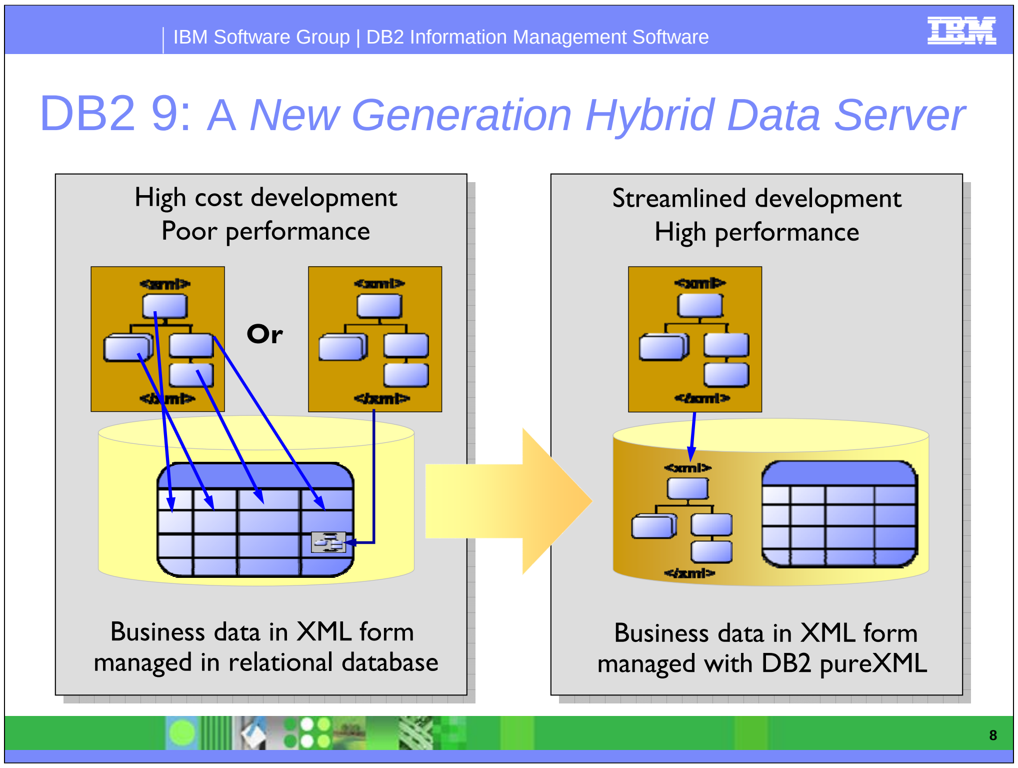
Yesterday's Solutions

- ***Not Persisting XML***
  - No cost effective solution
- ***XML in File Systems***
  - No elegant solution for backup, recover, search, retrieval
- ***XML Proprietary Databases***
  - Difficult to integrate with traditional relational data
- ***Relational Model Accommodations***
  - Shredding
  - Large Objects



Problems of Relational Model XML

- CLOB storage:
  - Query evaluation & sub-document level access requires costly XML Parsing – too slow !
- Shredding:
  - Mapping from XML to relational often too complex
  - Often requires dozens or hundreds of tables
  - Complex multi-way joins to reconstruct documents
  - XML schema changes break the mapping
    - no schema flexibility !
    - For example: Change element from single- to multi-occurrence requires normalization of relational schema & data


IBM Software Group | DB2 Information Management Software

DB2 9: *A New Generation Hybrid Data Server*

High cost development
Poor performance
<xml>
</xml>
Or
Business data in XML form managed in relational database
Streamlined development
High performance
Business data in XML form managed with DB2 pureXML
8

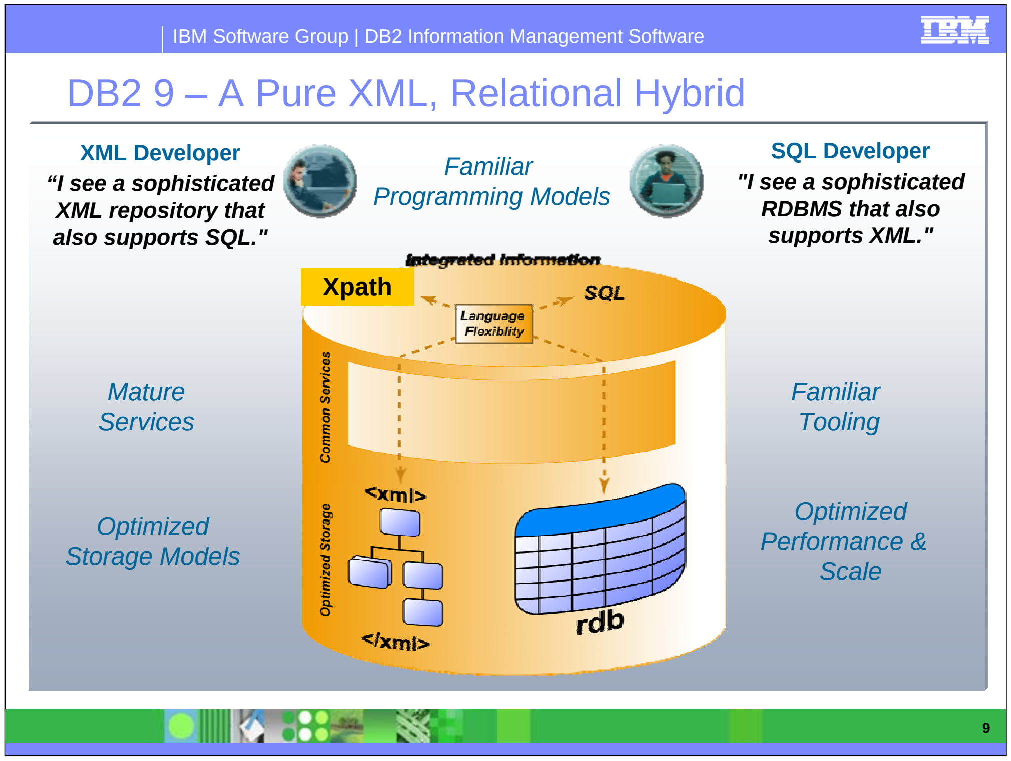
Key point: The amount of business information in XML form is already as great or greater than other forms and growing faster - failure to leverage efficiently as structured data means high cost and/or missed opportunity

DB2 9 provides the best of both worlds, pureXML^{tm} for native storage and integrating XML with object-relational. Performance, integrity, protection, and scale from the proven DB2 infrastructure with the flexibility of XML/XPath and relational/SQL. This overcomes the complexity & limitations of prior models (shred, CLOB, or XML only)

In 2006 IBM introduced a new generation data server with the availability of DB2 9. The explosive growth of XML based data standards in all industries means competitive advantage for those businesses that use it most effectively and efficiently. Client, policy and claims processing in Insurance; supply chain management in Retail; financial transactions and asset management in Banking; patient care in Healthcare; citizen service in Government; implementing Service Oriented Architectures (SOA) in Computing Software and Services - and many other processes across all industries - increasingly rely on information captured and exchanged in XML form. Our clients are increasingly managing XML format text documents in a content management system for proper governance and efficient use in the business process workflow. But few are realizing the full value of all the business data they possess that are in XML format.

Early users of the pureXML feature of DB2 9 are taking advantage of the fact that data in XML format is well structured and can be queried via standard languages such as XPath and XQuery. By doing so they are bringing that data to bear in both transactional and analytic processes - with higher performance and lower development costs than previously possible with a relational database. The difference is that DB2 9 supports both relational (tabular) and XML (hierarchical) structures in the same database so that both can be easily, efficiently and securely managed, analyzed and delivered. Unlike other relational data servers - and previous versions of DB2 - pureXML eliminates the overhead of fitting the “square peg” XML tree structure into the “round hole” row and column relational structure.

Until DB2 9, managing XML data records with a relational data servers meant decomposing the data



DB2 9 – A Pure XML, Relational Hybrid

XML Developer
"I see a sophisticated XML repository that also supports SQL."

Familiar Programming Models

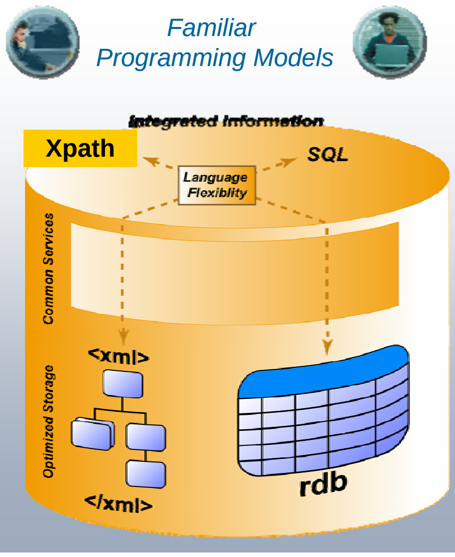
SQL Developer
"I see a sophisticated RDBMS that also supports XML."

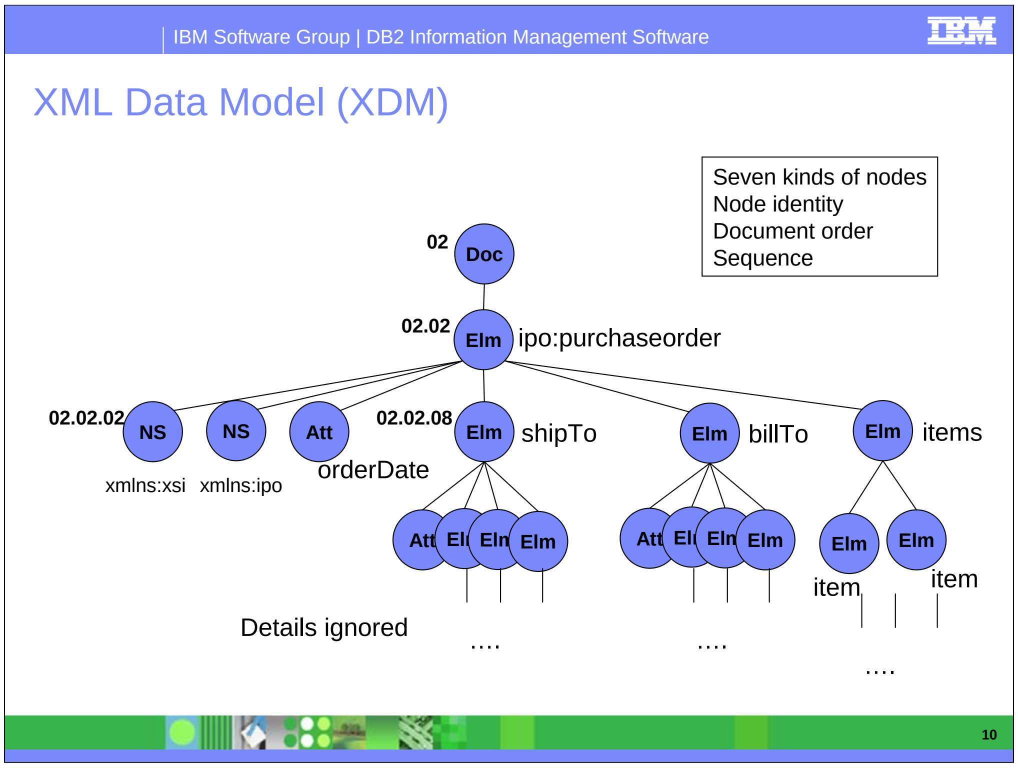
Mature Services

Optimized Storage Models

Familiar Tooling

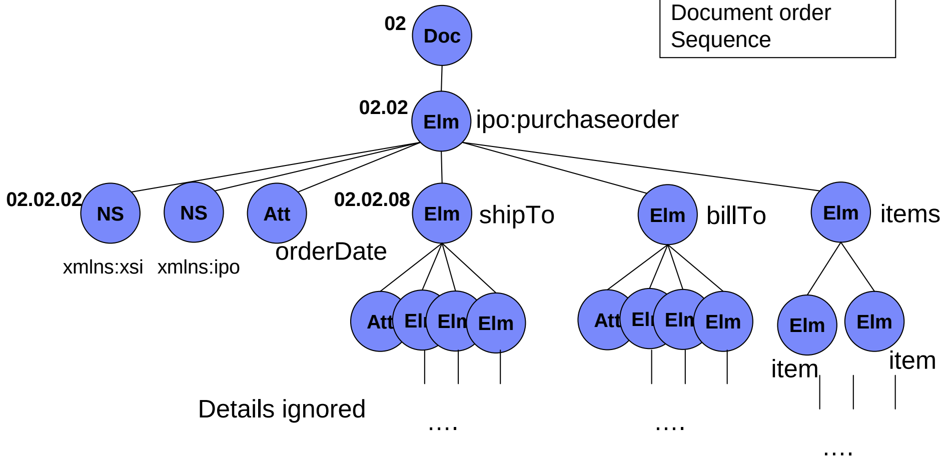
Optimized Performance & Scale





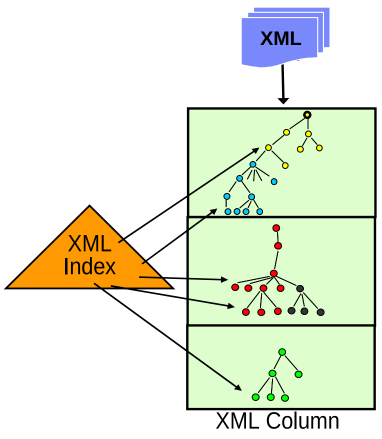
XML Data Model (XDM)

Seven kinds of nodes
Node identity
Document order
Sequence



DB2 pureXML Advantages

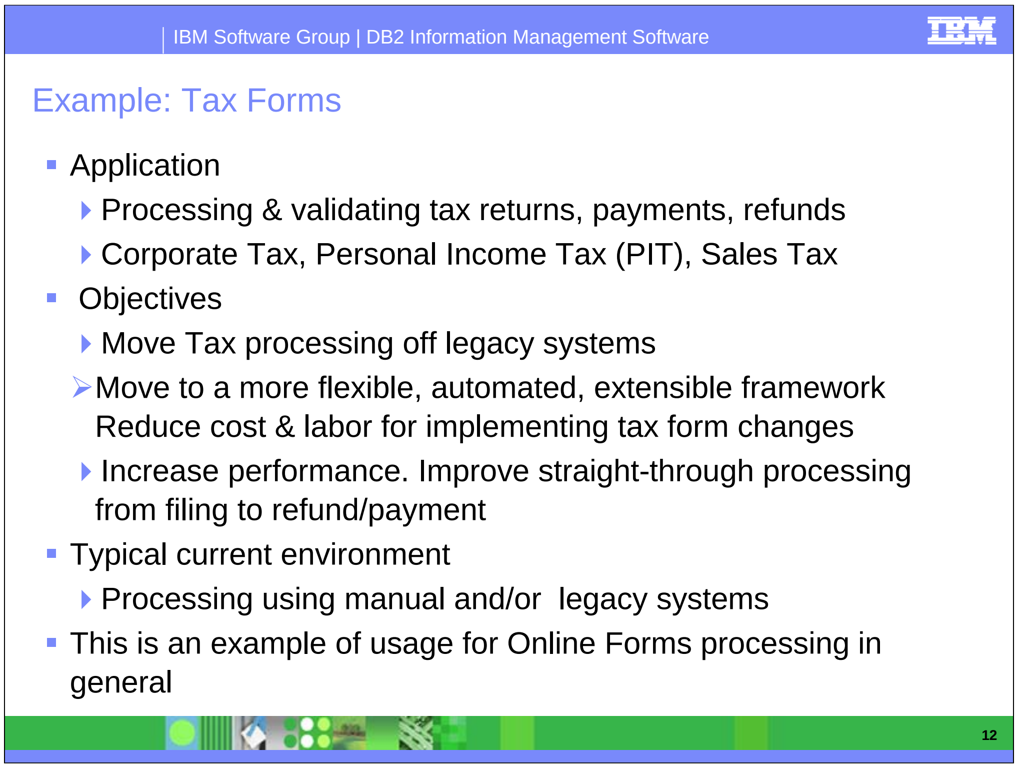
DB2's hierarchical storage:
XML type as XML



- Directly store XML, no decomp/comp, normalize/de-normalize
- Eliminates database schema evolution bottleneck
- Declarative language, reduce complexity, dramatically improve application development productivity
- Native processing, high performance

Up to 10 times

- Unparalleled reliability, availability, scalability



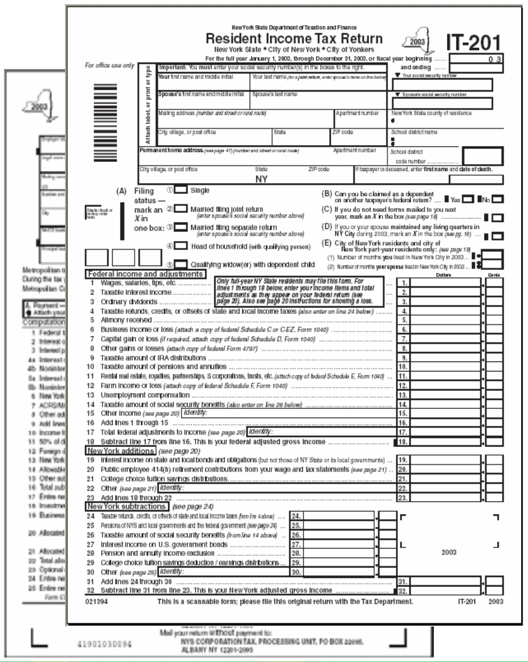
Example: Tax Forms

- Application
  - Processing & validating tax returns, payments, refunds
  - Corporate Tax, Personal Income Tax (PIT), Sales Tax
- Objectives
  - Move Tax processing off legacy systems
  - Move to a more flexible, automated, extensible framework Reduce cost & labor for implementing tax form changes
  - Increase performance. Improve straight-through processing from filing to refund/payment
- Typical current environment
  - Processing using manual and/or legacy systems
- This is an example of usage for Online Forms processing in general

Tax Forms

- Usually hundreds-thousands of different tax forms
  → **Schema Diversity**
- Typically not every field in a form is used
  → **Sparse Data**
- Many forms change every year
  → **Schema Evolution**

➔ A case for XML !



New York State Department of Taxation and Finance
Resident Income Tax Return 2003 **IT-201**
New York State • City of New York • City of Yonkers
For the full year January 1, 2003, through December 31, 2003, or fiscal year beginning 0 3
and ending ....

For office use only

Attach label, or print or type

Important: You **must** enter your social security number(s) in the boxes to the right.

Your first name and middle initial	Your last name *(for a joint return, enter spouse's name on line below)*	Your social security number
Spouse's first name and middle initial	Spouse's last name	Spouse's social security number
Mailing address *(number and street or rural route)*	Apartment number	New York State county of residence
City, village, or post office / State / ZIP code		School district name
Permanent home address *(see page 47) (number and street or rural route)*	Apartment number	School district code number ....
City village, or post office / State NY / ZIP code		If taxpayer is deceased, enter **first name** and **date of death.**

(A) Filing status — mark an *X* in one box:
① Single
② Married filing joint return *(enter spouse's social security number above)*
③ Married filing separate return *(enter spouse's social security number above)*
④ Head of household (with qualifying person)
⑤ Qualifying widow(er) with dependent child

Staple check or money order here.

(B) Can you be claimed as a dependent on another taxpayer's federal return? .... Yes ☐ No ☐
(C) If you do not need forms mailed to you next year, mark an *X* in the box *(see page 18)* .... ☐
(D) If you or your spouse **maintained any living quarters in NY City** during 2003, mark an *X* in the box *(see pg. 19)* .... ☐
(E) City of New York residents and city of New York part-year residents only: *(see page 19)*
(1) Number of months **you** lived in New York City in 2003 ...
(2) Number of months **your spouse** lived in New York City in 2003 ...

Only full-year NY State residents may file this form. For lines 1 through 18 below, enter your income items and total adjustments as they appear on your federal return (see page 20). Also see page 20 instructions for showing a loss.

			Dollars	Cents
Federal income and adjustments				
1	Wages, salaries, tips, etc.	1.		
2	Taxable interest income	2.		
3	Ordinary dividends	3.		
4	Taxable refunds, credits, or offsets of state and local income taxes *(also enter on line 24 below)*	4.		
5	Alimony received	5.		
6	Business income or loss *(attach a copy of federal Schedule C or C-EZ, Form 1040)*	6.		
7	Capital gain or loss *(if required, attach copy of federal Schedule D, Form 1040)*	7.		
8	Other gains or losses *(attach copy of federal Form 4797)*	8.		
9	Taxable amount of IRA distributions	9.		
10	Taxable amount of pensions and annuities	10.		
11	Rental real estate, royalties, partnerships, S corporations, trusts, etc. *(attach copy of federal Schedule E, Form 1040)* ...	11.		
12	Farm income or loss *(attach copy of federal Schedule F, Form 1040)*	12.		
13	Unemployment compensation	13.		
14	Taxable amount of social security benefits *(also enter on line 26 below)*	14.		
15	Other income *(see page 20)* *Identify:*	15.		
16	Add lines 1 through 15	16.		
17	Total federal adjustments to income *(see page 20)* *Identify:*	17.		
18	Subtract line 17 from line 16. This is your federal adjusted gross income	18.		
New York additions *(see page 20)*				
19	Interest income on state and local bonds and obligations (but not those of NY State or its local governments) ...	19.		
20	Public employee 414(h) retirement contributions from your wage and tax statements *(see page 21)* ..	20.		
21	College choice tuition savings distributions............	21.		
22	Other *(see page 21)* *Identify:*	22.		
23	Add lines 18 through 22	23.		
New York subtractions *(see page 24)*				
24	Taxable refunds, credits, or offsets of state and local income taxes *(from line 4 above)*	24.		
25	Pensions of NYS and local governments and the federal government *(see page 24)* ...	25.		
26	Taxable amount of social security benefits *(from line 14 above)* ..	26.		
27	Interest income on U.S. government bonds	27.		
28	Pension and annuity income exclusion	28.		
29	College choice tuition savings deduction / earnings distributions ...	29.		
30	Other *(see page 25)* *Identify:*	30.		
31	Add lines 24 through 30	31.		
32	Subtract line 31 from line 23. This is your New York adjusted gross income	32.		

2003

021394 This is a scannable form; please file this original return with the Tax Department. IT-201 2003

41901030094
Mail your return without payment to:
NYS CORPORATION TAX, PROCESSING UNIT, PO BOX 22095,
ALBANY NY 12201-2095

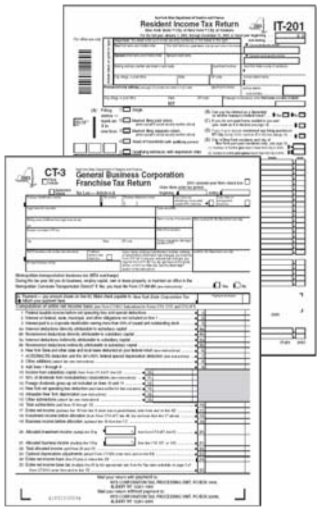
Typical Current Usage: Relational Database

- Solution 1: Each form has a different set of fields (schema)

→ Thousands of Tables … i.e. one per form ?

- Considered not feasible
  - Too many tables to maintain
  - Relational schema would deteriorate over time
  - Not sufficiently flexible and extensible

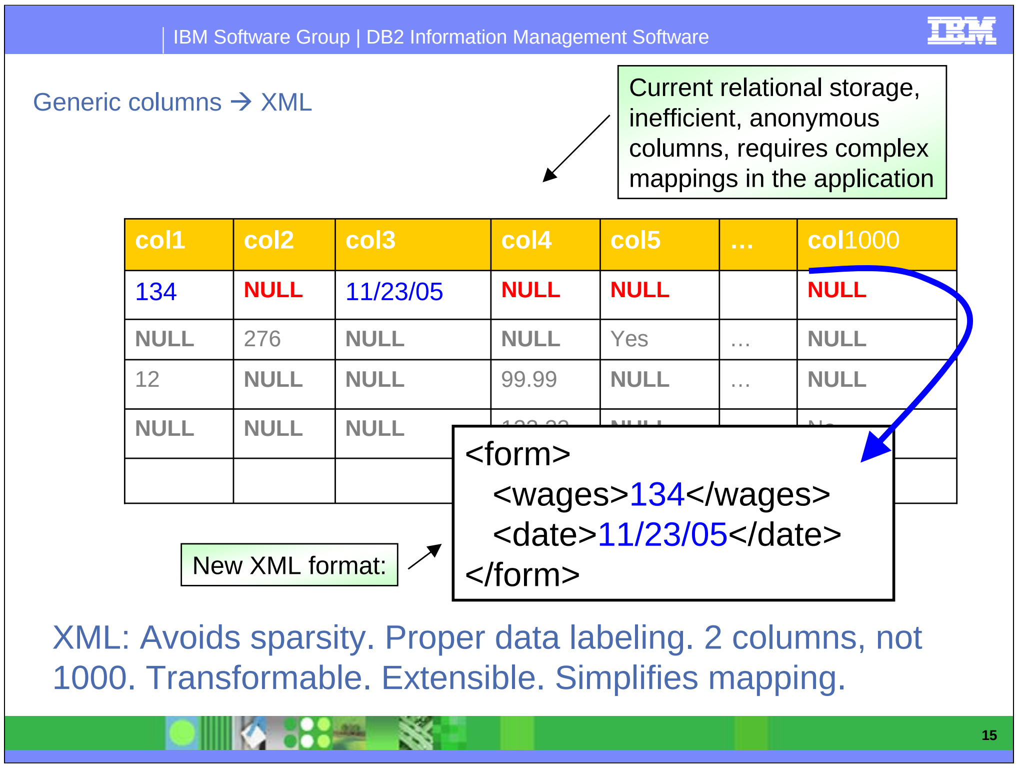
- Solution 2: Single table whose rows can store *any* form
  - 100s of generic columns … Ouch!


Resident Income Tax Return

IT-201

CT-3

General Business Corporation Franchise Tax Return



Generic columns → XML

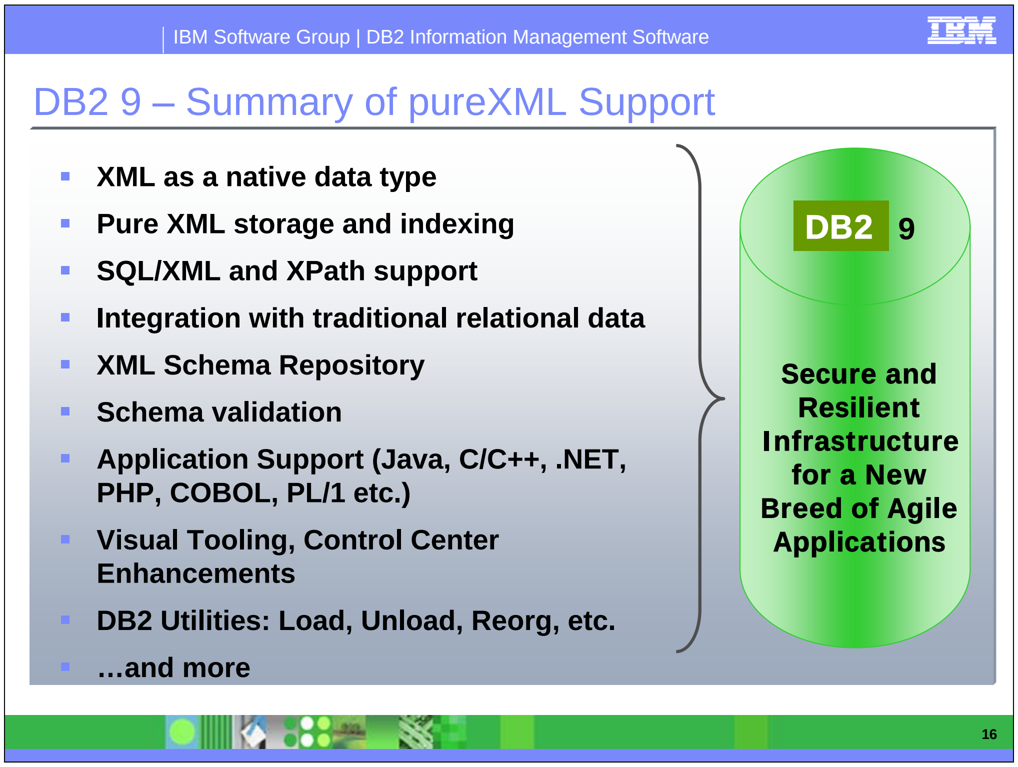
Current relational storage, inefficient, anonymous columns, requires complex mappings in the application

col1	col2	col3	col4	col5	…	col1000
134	NULL	11/23/05	NULL	NULL		NULL
NULL	276	NULL	NULL	Yes	…	NULL
12	NULL	NULL	99.99	NULL	…	NULL
NULL	NULL	NULL	123.23	NULL		No

New XML format:

```
<form>
  <wages>134</wages>
  <date>11/23/05</date>
</form>
```

XML: Avoids sparsity. Proper data labeling. 2 columns, not 1000. Transformable. Extensible. Simplifies mapping.



DB2 9 – Summary of pureXML Support

- XML as a native data type
- Pure XML storage and indexing
- SQL/XML and XPath support
- Integration with traditional relational data
- XML Schema Repository
- Schema validation
- Application Support (Java, C/C++, .NET, PHP, COBOL, PL/1 etc.)
- Visual Tooling, Control Center Enhancements
- DB2 Utilities: Load, Unload, Reorg, etc.
- …and more

DB2 9

Secure and Resilient Infrastructure for a New Breed of Agile Applications

Interface Overview

- **Data Definition**

  create table dept(deptID int, deptdoc xml);
- **Insert**

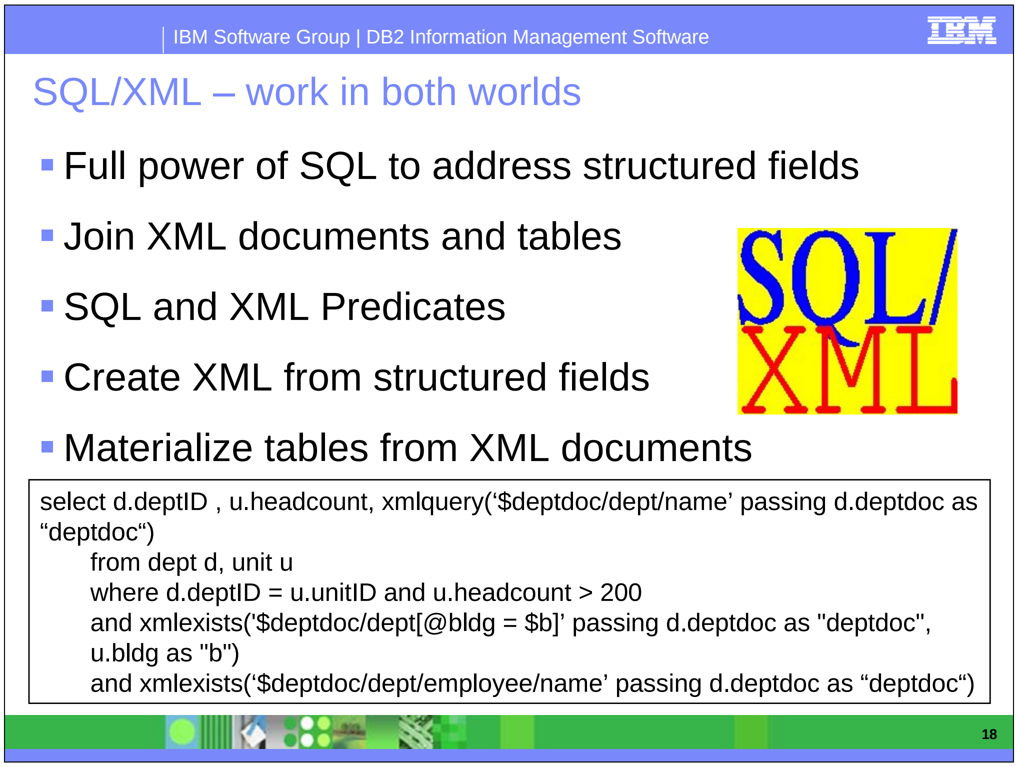
  insert into dept(deptID, deptdoc) values (?,?)
- **Index**

  create index xmlindex1 on dept(deptdoc)

  generate key using xmlpattern '/dept/name' as sql varchar(30);
- **Retrieve**

  select deptdoc from dept where deptID = ?
- **Query**

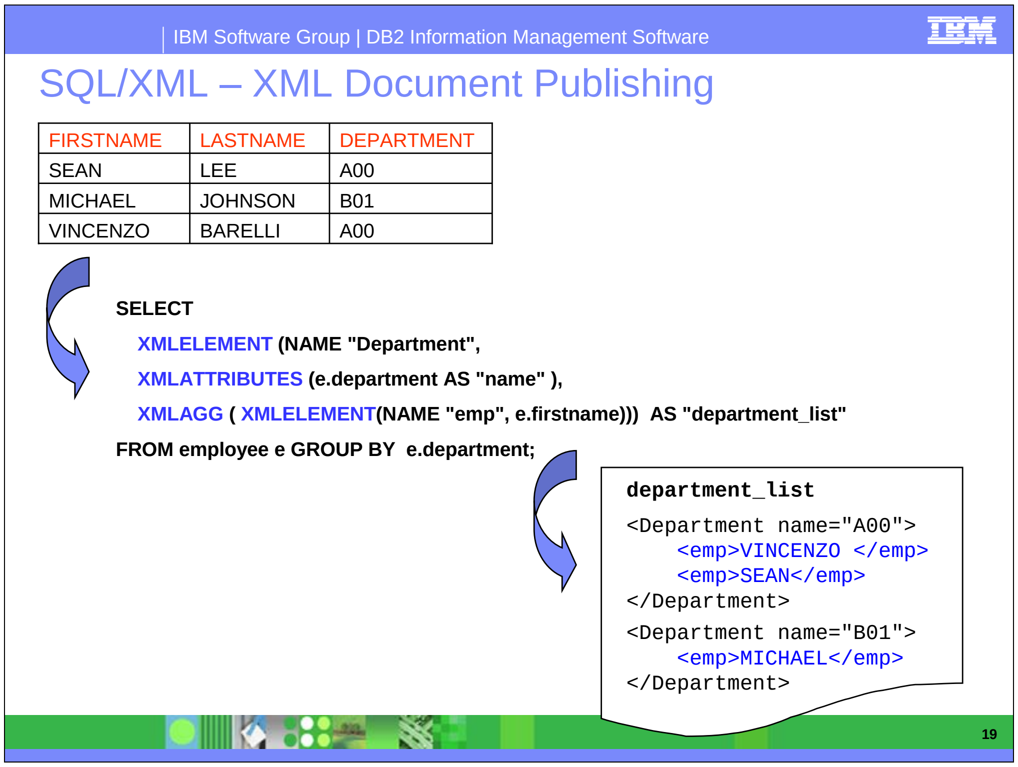
  select deptID, xmlquery('/dept/name' passing deptdoc)
    from dept where deptID <> "PR27";



SQL/XML – work in both worlds

- Full power of SQL to address structured fields
- Join XML documents and tables
- SQL and XML Predicates
- Create XML from structured fields
- Materialize tables from XML documents

```
select d.deptID , u.headcount, xmlquery(‘$deptdoc/dept/name’ passing d.deptdoc as
“deptdoc“)
        from dept d, unit u
        where d.deptID = u.unitID and u.headcount > 200
        and xmlexists('$deptdoc/dept[@bldg = $b]’ passing d.deptdoc as "deptdoc",
        u.bldg as "b")
        and xmlexists(‘$deptdoc/dept/employee/name’ passing d.deptdoc as “deptdoc“)
```

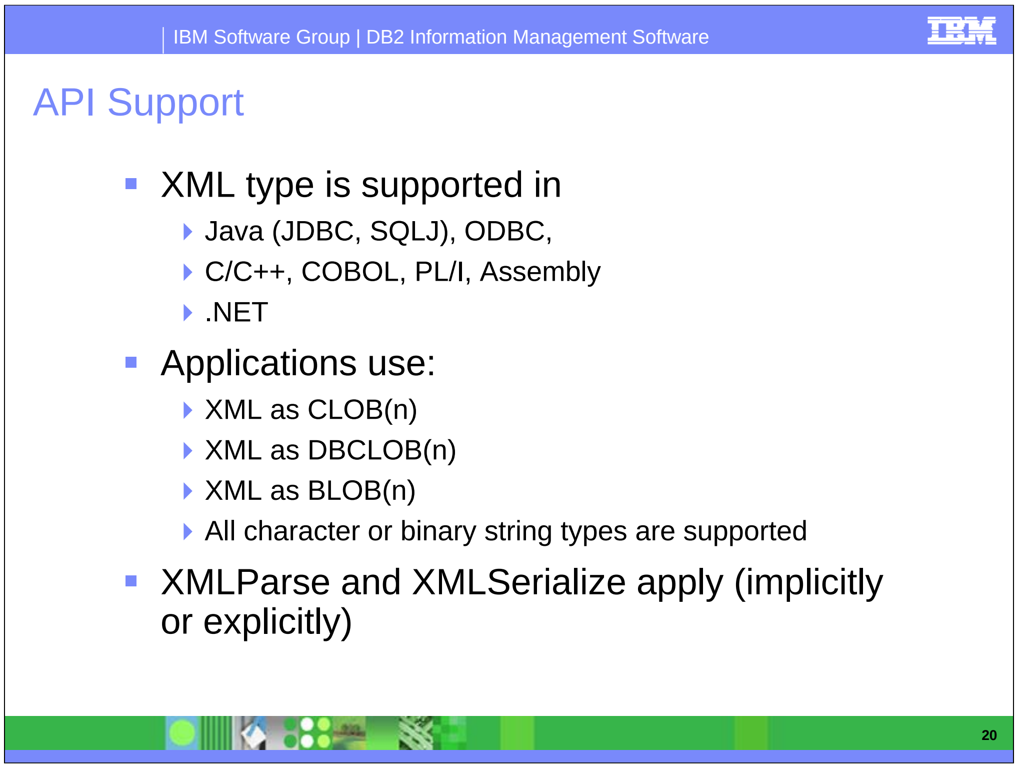


SQL/XML – XML Document Publishing

FIRSTNAME	LASTNAME	DEPARTMENT
SEAN	LEE	A00
MICHAEL	JOHNSON	B01
VINCENZO	BARELLI	A00

```
SELECT
   XMLELEMENT (NAME "Department",
   XMLATTRIBUTES (e.department AS "name" ),
   XMLAGG ( XMLELEMENT(NAME "emp", e.firstname)))  AS "department_list"
FROM employee e GROUP BY  e.department;
```

```
department_list
<Department name="A00">
    <emp>VINCENZO </emp>
    <emp>SEAN</emp>
</Department>
<Department name="B01">
    <emp>MICHAEL</emp>
</Department>
```

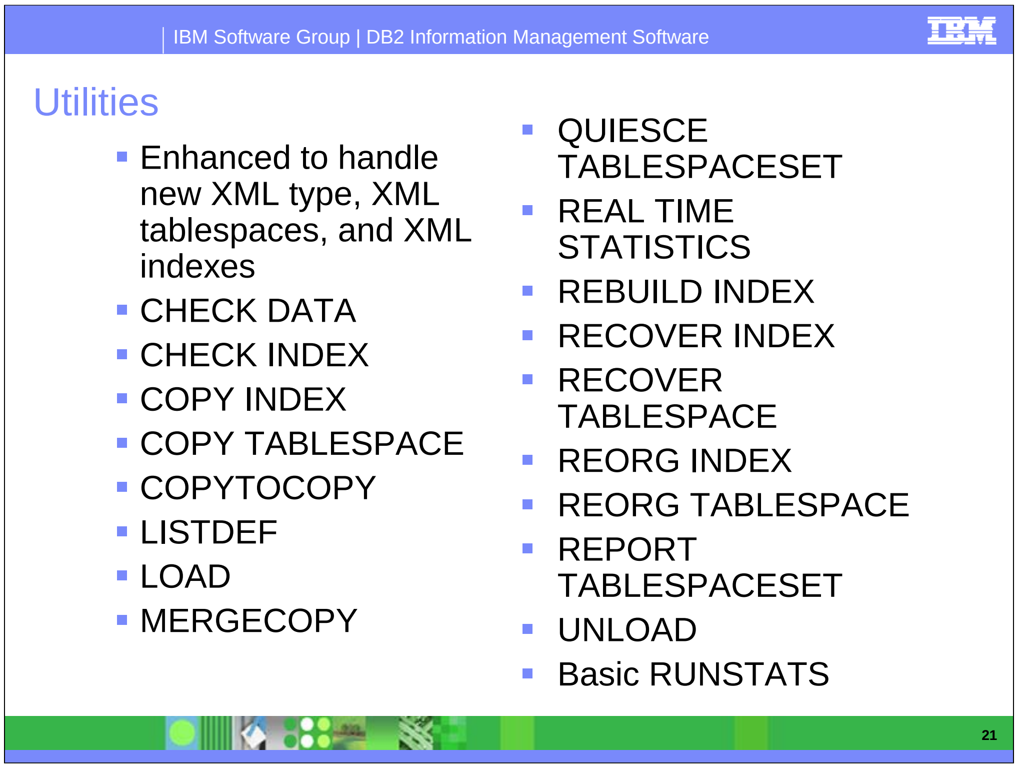

IBM Software Group | DB2 Information Management Software

API Support

- XML type is supported in
  - Java (JDBC, SQLJ), ODBC,
  - C/C++, COBOL, PL/I, Assembly
  - .NET
- Applications use:
  - XML as CLOB(n)
  - XML as DBCLOB(n)
  - XML as BLOB(n)
  - All character or binary string types are supported
- XMLParse and XMLSerialize apply (implicitly or explicitly)

20

DRDA supports internally encoded or externally encoded XML type (in serialized string format).



Utilities

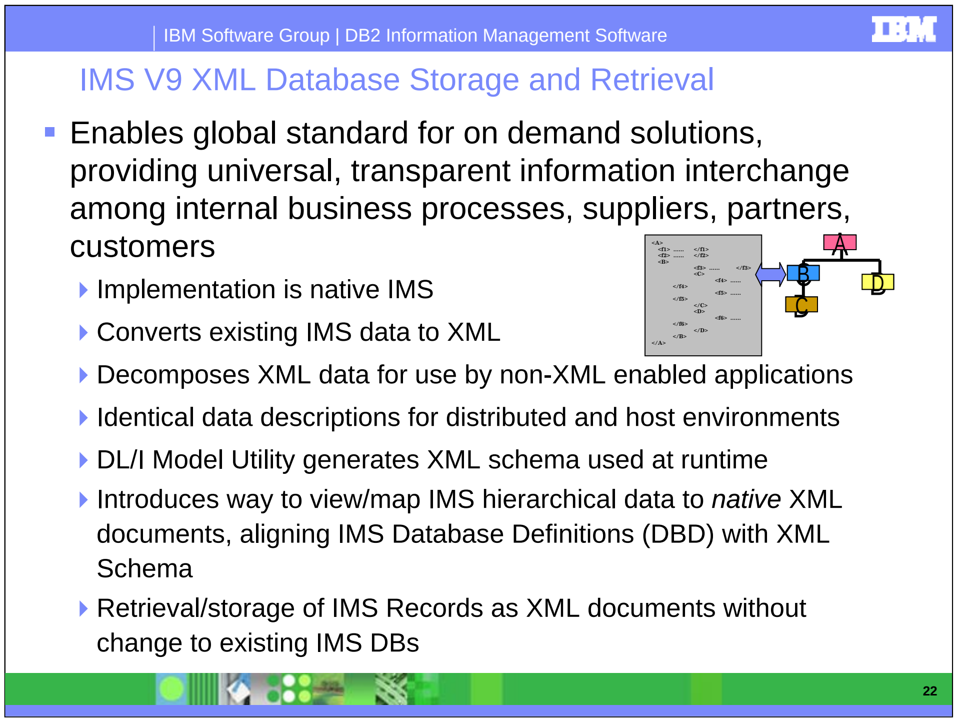
- Enhanced to handle new XML type, XML tablespaces, and XML indexes
- CHECK DATA
- CHECK INDEX
- COPY INDEX
- COPY TABLESPACE
- COPYTOCOPY
- LISTDEF
- LOAD
- MERGECOPY
- QUIESCE TABLESPACESET
- REAL TIME STATISTICS
- REBUILD INDEX
- RECOVER INDEX
- RECOVER TABLESPACE
- REORG INDEX
- REORG TABLESPACE
- REPORT TABLESPACESET
- UNLOAD
- Basic RUNSTATS

And relationship between base table and internal XML tables.

Some of the utilities do not do lot of work, but to tolerate and recognize XML data.

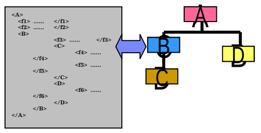
LOAD file references for large XML documents.

Believe or not ONLINE REORG is supported for XML tablespaces.



IMS V9 XML Database Storage and Retrieval

- Enables global standard for on demand solutions, providing universal, transparent information interchange among internal business processes, suppliers, partners, customers
  - Implementation is native IMS
  - Converts existing IMS data to XML
  - Decomposes XML data for use by non-XML enabled applications
  - Identical data descriptions for distributed and host environments
  - DL/I Model Utility generates XML schema used at runtime
  - Introduces way to view/map IMS hierarchical data to *native* XML documents, aligning IMS Database Definitions (DBD) with XML Schema
  - Retrieval/storage of IMS Records as XML documents without change to existing IMS DBs



Key Message: A key element is XML Database support for storage and retrieval of XML data in IMS databases

•The implementation is native IMS, not merely a mapping as must be done with other technologies, providing efficient utilization of resources and top overall performance. Existing IMS data can be easily converted to XML to facilitate integration with business processes, improve programmer productivity and reduce development lead times.

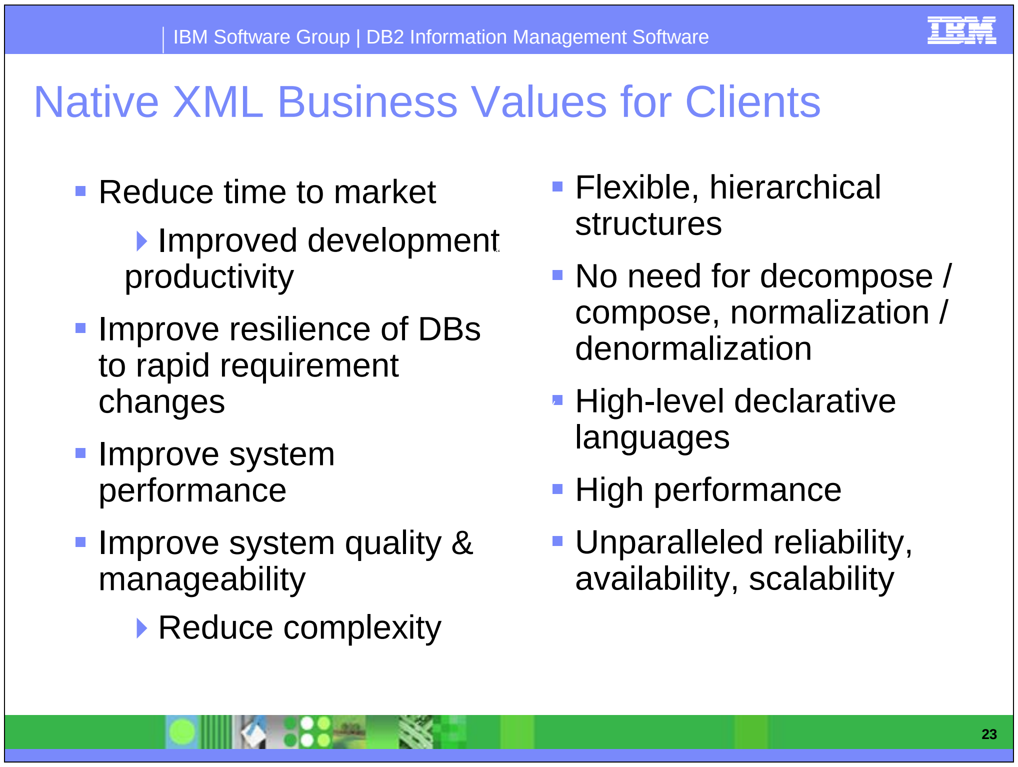
•XML data can be decomposed for use by non-XML enabled applications, thereby preserving and extending past investment and enhancing programmer productivity.

•Data descriptions can be the same for distributed and host environments to reduce overhead and improve data consistency and integrity.

•DL/I Model Utility Enhancements generate XML schema from existing IMS Database Definitions (DBDs) and Program Status Blocks (PSBs) for XML storage and retrieval at runtime. This improves application development time, reduces errors and makes it possible to consolidate skills by allowing programmers to code in an industry standard interface.

•IMS V9 introduced a way to view your IMS data (from existing or new IMS databases) as collections of XML documents -- aligning the stored hierarchical IMS records with the hierarchically structured XML documents for retrieval and automated conversion between

•IMS V9 XML-DB allows you to retrieve and automatically convert an IMS record into an XML document. Similarly, it allows you to store an XML document into IMS by automatically converting the XML document into an IMS record.

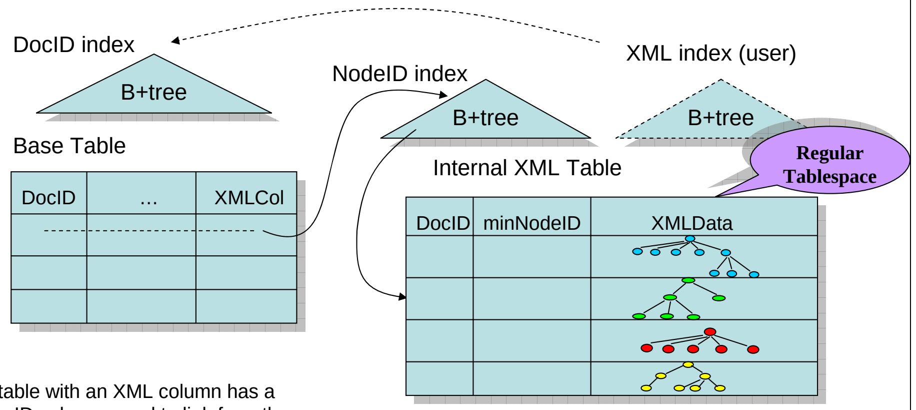


Native XML Business Values for Clients

- Reduce time to market
  - Improved development productivity
- Improve resilience of DBs to rapid requirement changes
- Improve system performance
- Improve system quality & manageability
  - Reduce complexity

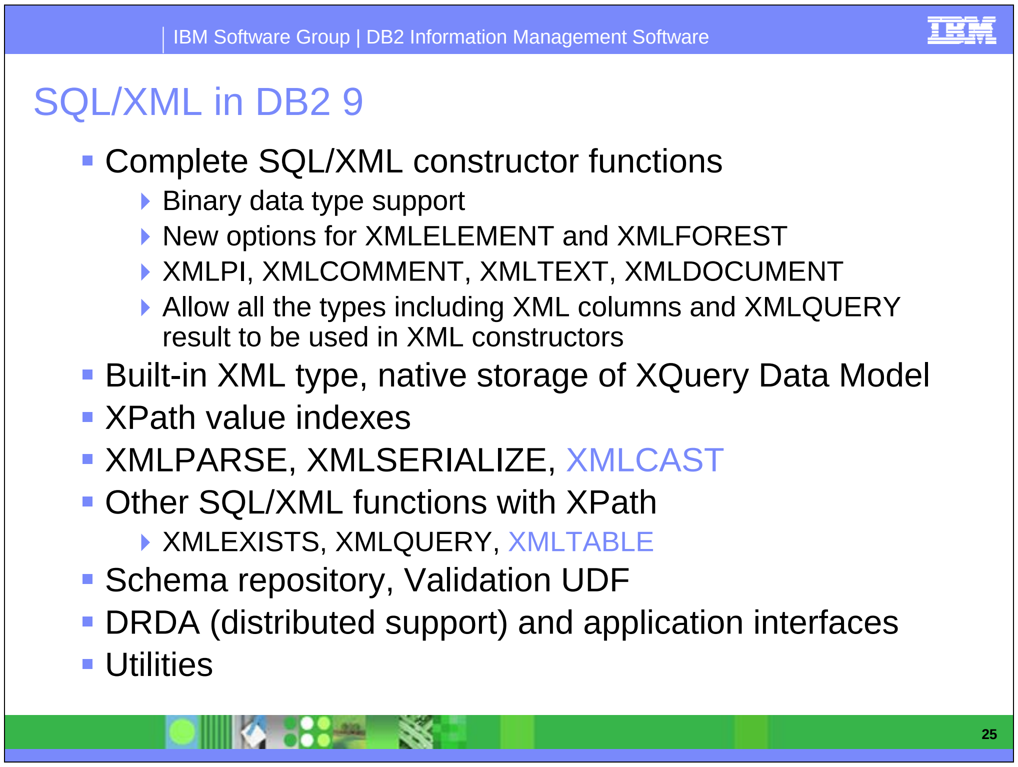
- Flexible, hierarchical structures
- No need for decompose / compose, normalization / denormalization
- High-level declarative languages
- High performance
- Unparalleled reliability, availability, scalability

XML Storage



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.

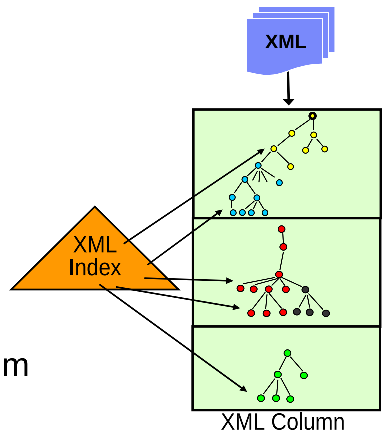


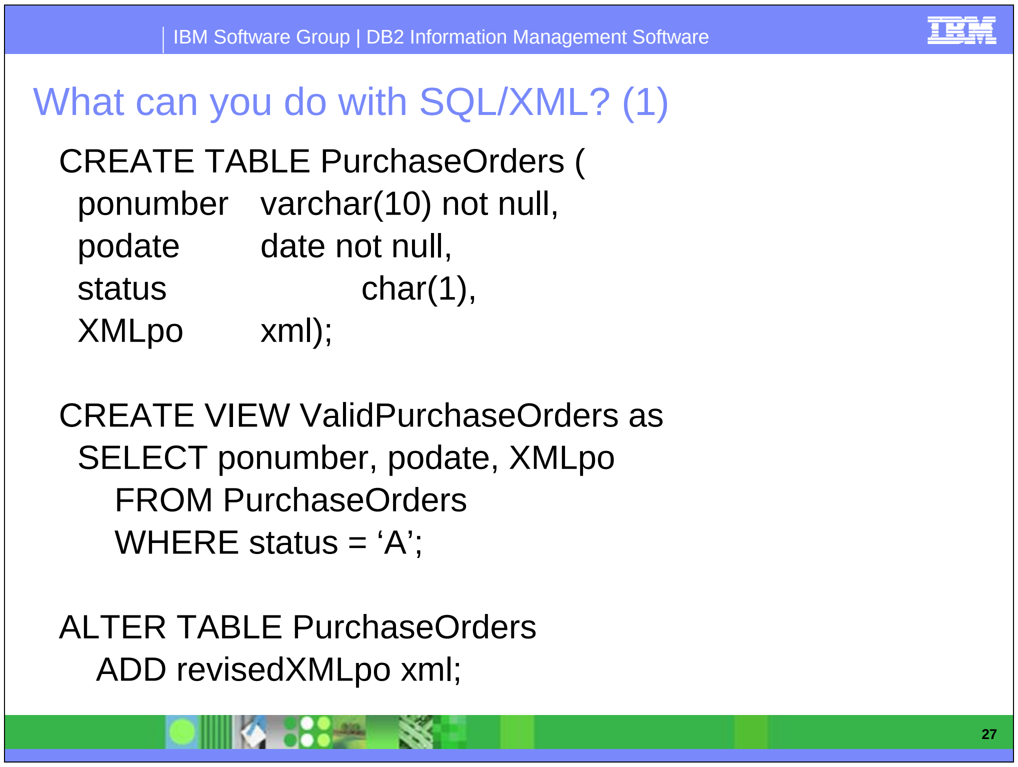
SQL/XML in DB2 9

- Complete SQL/XML constructor functions
  - Binary data type support
  - New options for XMLELEMENT and XMLFOREST
  - XMLPI, XMLCOMMENT, XMLTEXT, XMLDOCUMENT
  - Allow all the types including XML columns and XMLQUERY result to be used in XML constructors
- Built-in XML type, native storage of XQuery Data Model
- XPath value indexes
- XMLPARSE, XMLSERIALIZE, XMLCAST
- Other SQL/XML functions with XPath
  - XMLEXISTS, XMLQUERY, XMLTABLE
- Schema repository, Validation UDF
- DRDA (distributed support) and application interfaces
- Utilities

What You Can Do with pureXML

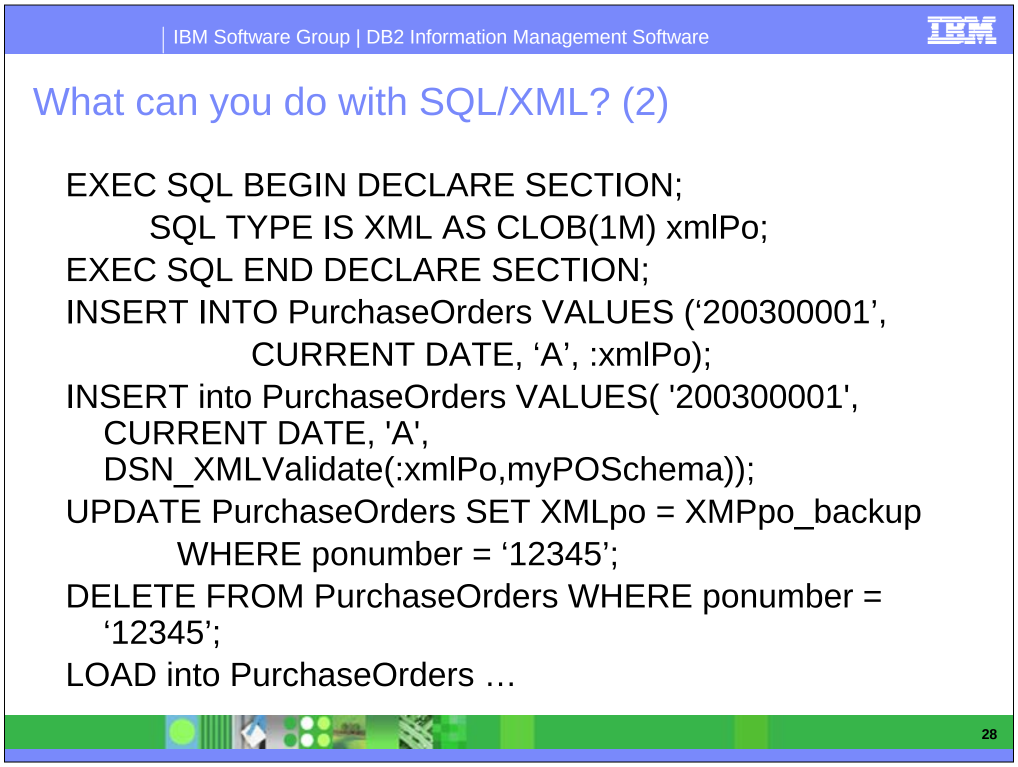
- Create tables with XML columns
- Insert XML data, optionally validated against schemas
- Create indexes on XML data
- Efficiently search XML data
- Extract XML data
- Decompose XML data into relational data
- Construct XML documents from relational and XML data
- All the utilities and tools support for XML





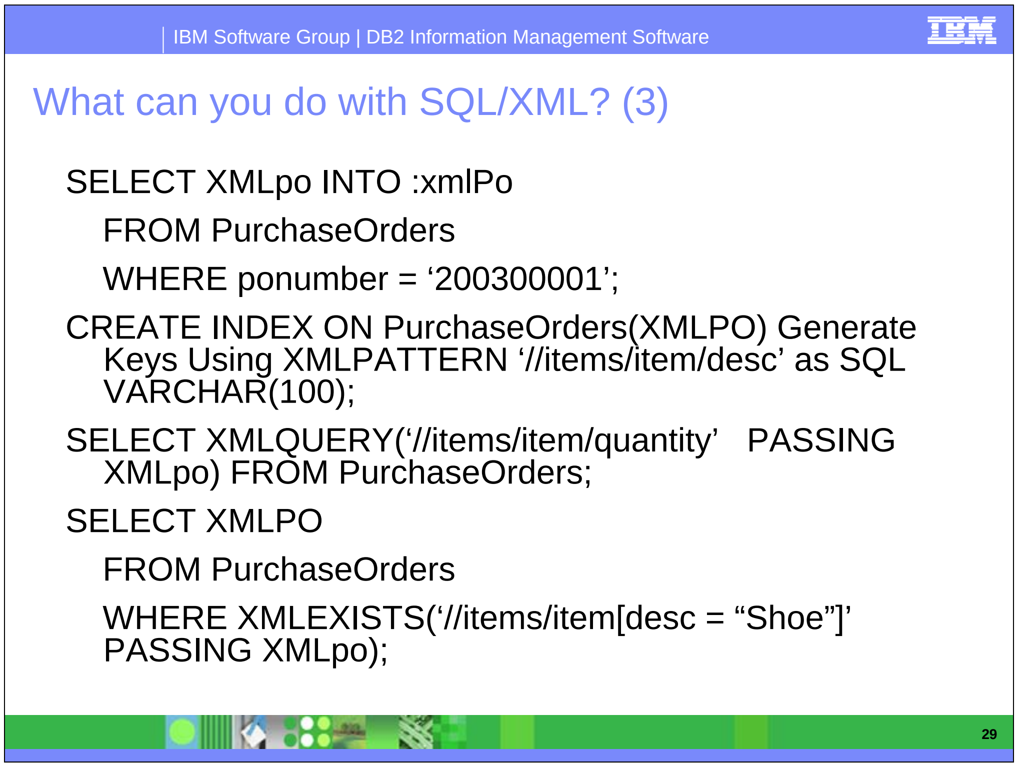
What can you do with SQL/XML? (1)

```
CREATE TABLE PurchaseOrders (
  ponumber   varchar(10) not null,
  podate        date not null,
  status                  char(1),
  XMLpo        xml);

CREATE VIEW ValidPurchaseOrders as
  SELECT ponumber, podate, XMLpo
      FROM PurchaseOrders
      WHERE status = ‘A’;

ALTER TABLE PurchaseOrders
    ADD revisedXMLpo xml;
```



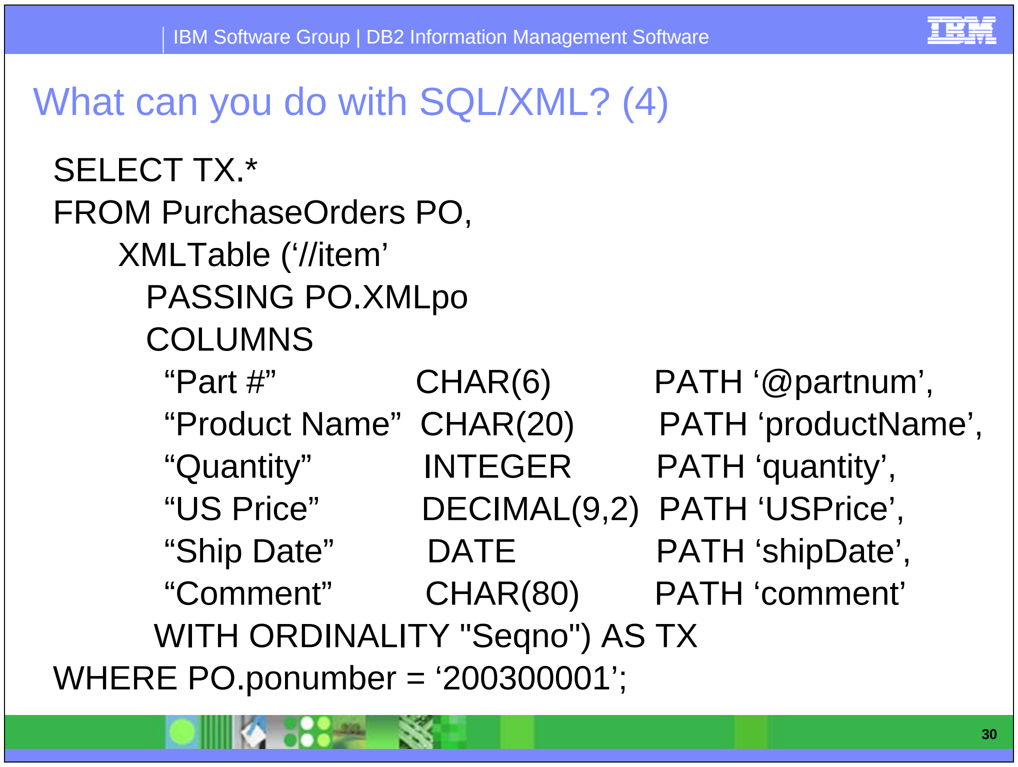
What can you do with SQL/XML? (2)

```
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( '200300001',
   CURRENT DATE, 'A',
   DSN_XMLValidate(:xmlPo,myPOSchema));
UPDATE PurchaseOrders SET XMLpo = XMPpo_backup
          WHERE ponumber = ‘12345’;
DELETE FROM PurchaseOrders WHERE ponumber =
   ‘12345’;
LOAD into PurchaseOrders …
```



What can you do with SQL/XML? (3)

```
SELECT XMLpo INTO :xmlPo
   FROM PurchaseOrders
   WHERE ponumber = ‘200300001’;
CREATE INDEX ON PurchaseOrders(XMLPO) Generate
   Keys Using XMLPATTERN ‘//items/item/desc’ as SQL
   VARCHAR(100);
SELECT XMLQUERY(‘//items/item/quantity’   PASSING
   XMLpo) FROM PurchaseOrders;
SELECT XMLPO
   FROM PurchaseOrders
   WHERE XMLEXISTS(‘//items/item[desc = “Shoe”]’
   PASSING XMLpo);
```



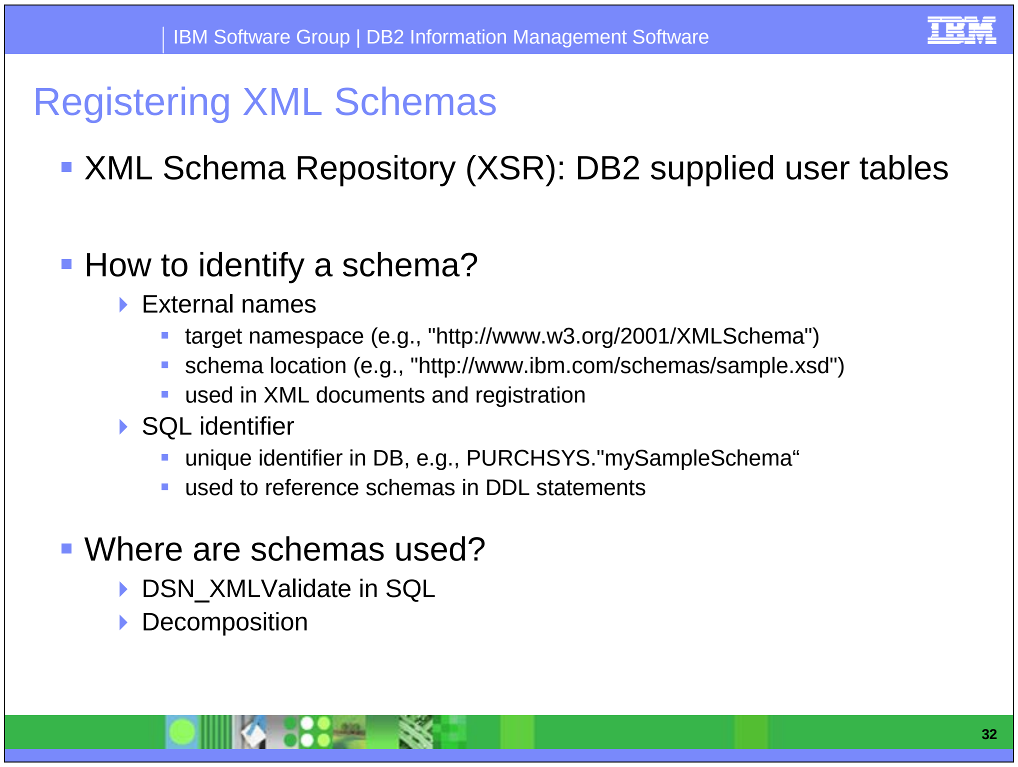
What can you do with SQL/XML? (4)

```
SELECT TX.*
FROM PurchaseOrders PO,
      XMLTable (‘//item’
         PASSING PO.XMLpo
         COLUMNS
           “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”        CHAR(80)       PATH ‘comment’
          WITH ORDINALITY "Seqno") AS TX
WHERE PO.ponumber = ‘200300001’;
```

What can you do with SQL/XML? (5)

```
<?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">
     <productName>Lawnmower</productName>
     <quantity>1</quantity>
     <USPrice>148.95</USPrice>
     <comment>Confirm this is electric</comment>
    </item>
    <item partNum="926-AA">
     <productName>Baby Monitor</productName>
     <quantity>1</quantity>
     <USPrice>39.98</USPrice>
     <shipDate>2003-05-21</shipDate>
    </item>
   </items>
</purchaseOrder>
```

Seqno	Part#	Product Name	Quantity	US Price	Ship Date	Comment
1	872-AA	Lawnmower	1	148.95	null	Confirm this is electric
2	926-AA	Baby Monitor	1	39.98	‘2003-05-21’	null

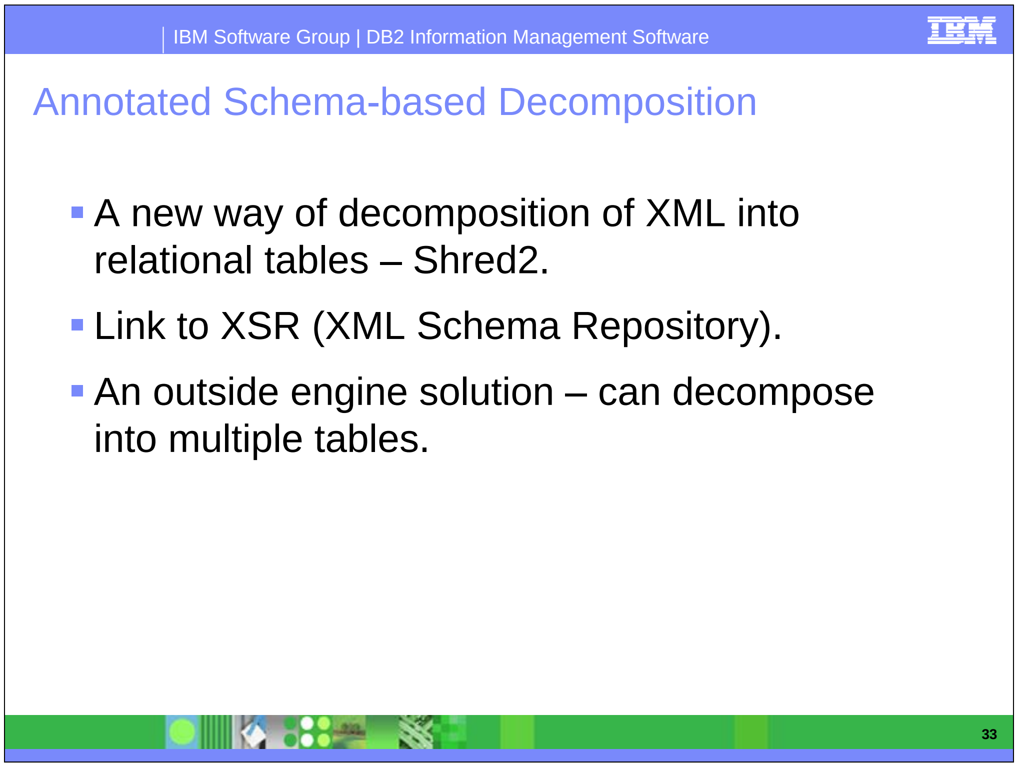


Registering XML Schemas

- XML Schema Repository (XSR): DB2 supplied user tables

- How to identify a schema?
  - External names
    - target namespace (e.g., "http://www.w3.org/2001/XMLSchema")
    - schema location (e.g., "http://www.ibm.com/schemas/sample.xsd")
    - used in XML documents and registration
  - SQL identifier
    - unique identifier in DB, e.g., PURCHSYS."mySampleSchema“
    - used to reference schemas in DDL statements

- Where are schemas used?
  - DSN_XMLValidate in SQL
  - Decomposition

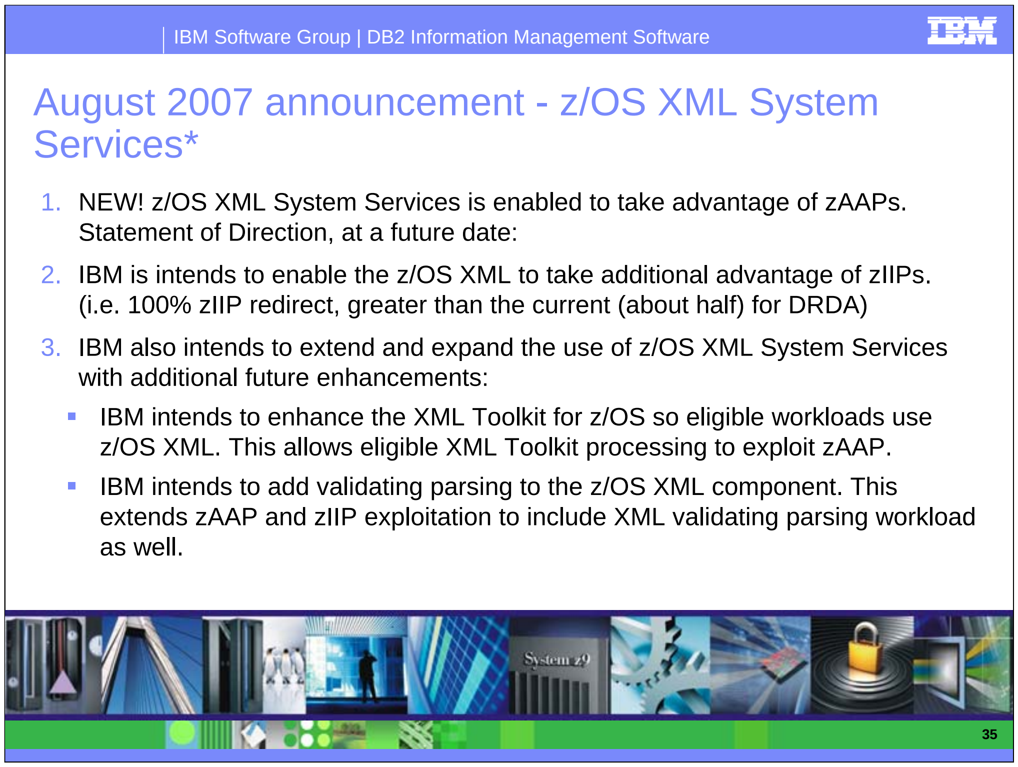


Annotated Schema-based Decomposition

- A new way of decomposition of XML into relational tables – Shred2.
- Link to XSR (XML Schema Repository).
- An outside engine solution – can decompose into multiple tables.

Efficient XPath Support

- Used in XMLEXISTS, XMLQUERY, XMLTABLE
- XPath 1.0 + -
  - XPath 1.0 constructs in XPath 2.0 semantics
  - + more data types: xs:boolean, xs:integer, xs:decimal, xs:double, xs:string
  - + namespace declaration from XQuery prolog
  - - Axes: only 5 forward axes & parent axis are supported.
- All stored XML data are untyped in V9.
  - Explicit type casting may be needed in some cases



August 2007 announcement - z/OS XML System Services*

1. NEW! z/OS XML System Services is enabled to take advantage of zAAPs. Statement of Direction, at a future date:
2. IBM is intends to enable the z/OS XML to take additional advantage of zIIPs. (i.e. 100% zIIP redirect, greater than the current (about half) for DRDA)
3. IBM also intends to extend and expand the use of z/OS XML System Services with additional future enhancements:
   - IBM intends to enhance the XML Toolkit for z/OS so eligible workloads use z/OS XML. This allows eligible XML Toolkit processing to exploit zAAP.
   - IBM intends to add validating parsing to the z/OS XML component. This extends zAAP and zIIP exploitation to include XML validating parsing workload as well.

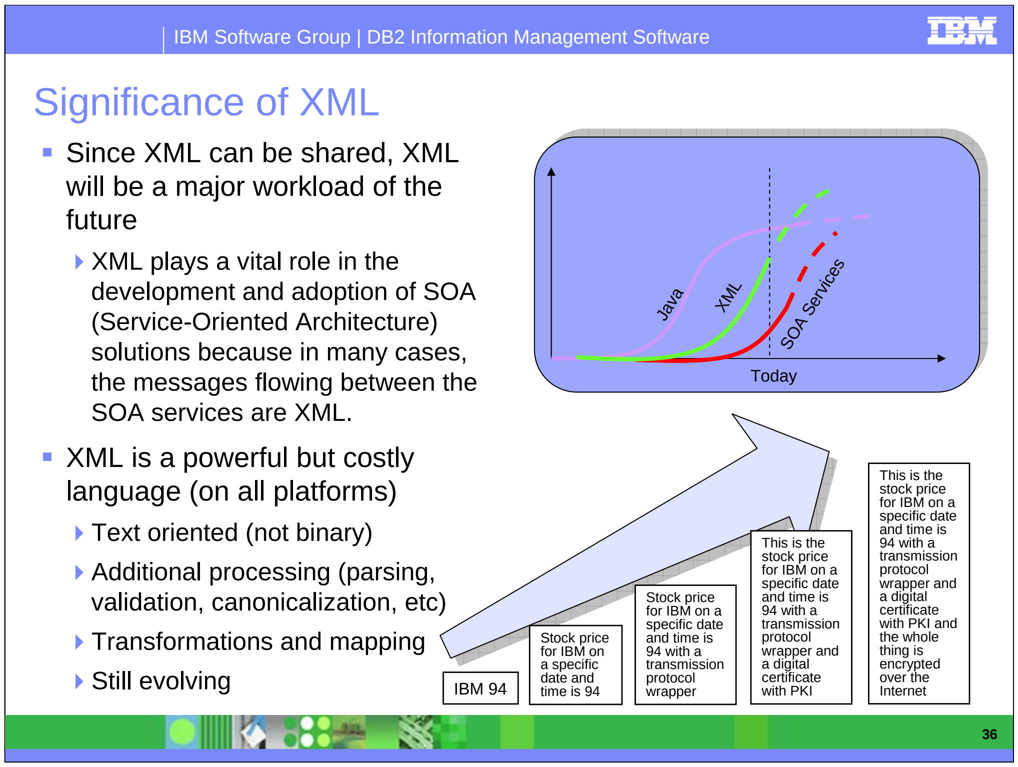
STATEMENT OF DIRECTION

z/OS XML enabled for both zAAP and zIIP specialty processors

This SOD is broken down into 3 parts so that it is easier to explain and communicate.

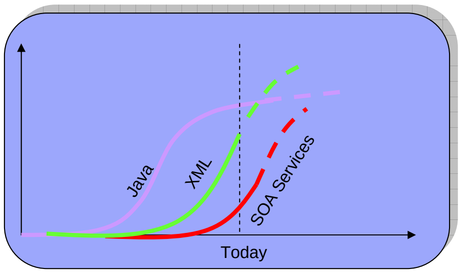
In z/OS V1.8, IBM introduced a new system component of z/OS, z/OS XML System Services (z/OS XML), a system-level XML parser integrated with the base z/OS operating system and designed to deliver an optimized set of services for parsing XML documents (z/OS XML has also been made available on z/OS V1.7). The initial beneficiaries of this system component are middleware and applications requiring high performance non-validating XML parsing. z/OS XML may currently be accessed by an Assembler programming interface and one of the first exploiters, DB2 9 for z/OS, uses this Assembler interface for XML native support. IBM plans to add C/C++ support for z/OS XML with z/OS V1.9 (satisfying the Statement of Direction in Software Announcement 206-039, dated February 28, 2006).

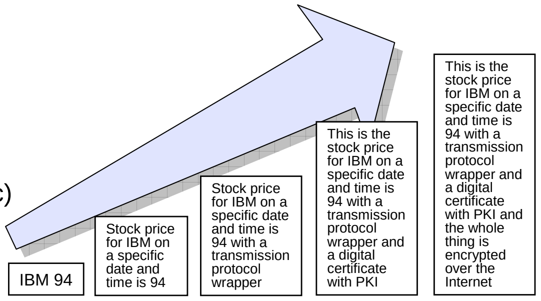
1) Today, IBM is announcing its intent to enable the z/OS XML component to take advantage of zAAPs. This future enhancement means that middleware and applications requesting z/OS XML System Services (for example DB2 processing via local connection) will have z/OS XML System Services processing execute on the zAAP. Specifically, all z/OS XML System Services parsing executing in TCB mode will be redirected to the zAAP.



Significance of XML

- Since XML can be shared, XML will be a major workload of the future
  - XML plays a vital role in the development and adoption of SOA (Service-Oriented Architecture) solutions because in many cases, the messages flowing between the SOA services are XML.
- XML is a powerful but costly language (on all platforms)
  - Text oriented (not binary)
  - Additional processing (parsing, validation, canonicalization, etc)
  - Transformations and mapping
  - Still evolving




IBM 94
Stock price for IBM on a specific date and time is 94
Stock price for IBM on a specific date and time is 94 with a transmission protocol wrapper
This is the stock price for IBM on a specific date and time is 94 with a transmission protocol wrapper and a digital certificate with PKI
This is the stock price for IBM on a specific date and time is 94 with a transmission protocol wrapper and a digital certificate with PKI and the whole thing is encrypted over the Internet

XML plays a vital role in the development and adoption of SOA (Service-Oriented Architecture) solutions because in many cases, the messages flowing between the SOA services are XML.

Some features of XML that make it well-suited for data transfer are:

•its simultaneously human- and machine-readable format;

•it has support for Unicode, allowing almost any information in any written human language to be communicated;

•the ability to represent the most general computer science data structures: records, lists and trees;

•the self-documenting format that describes structure and field names as well as specific values;

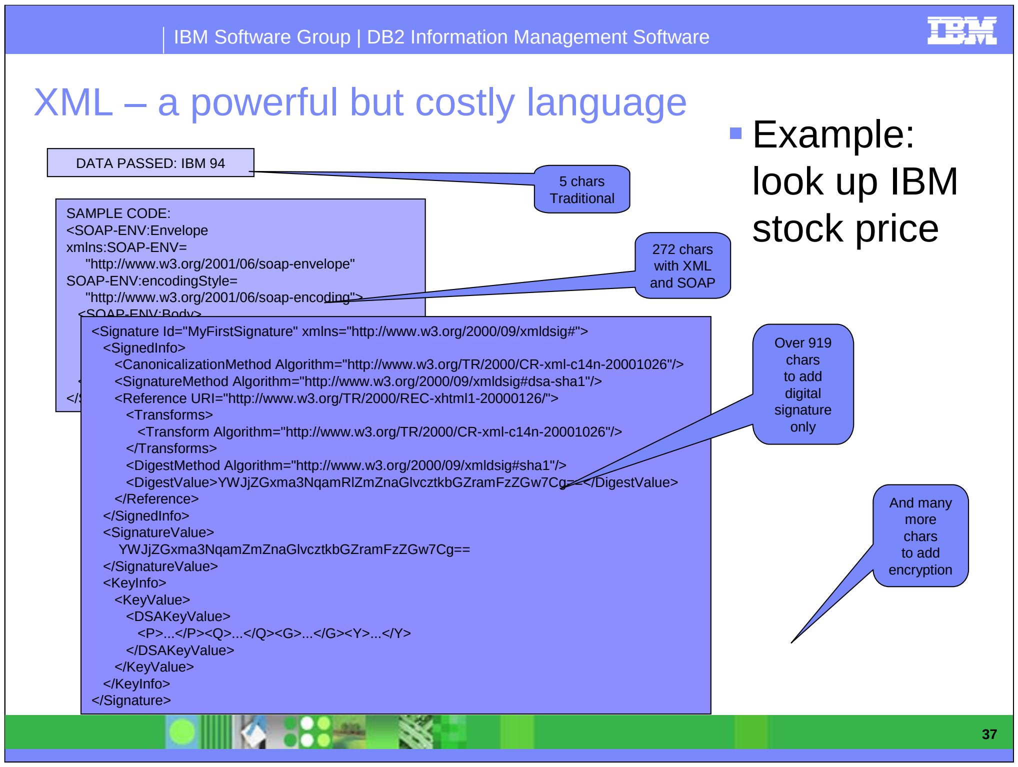
•the strict syntax and parsing requirements that allow the necessary parsing algorithms to remain simple, efficient, and consistent.

•it is platform-independent, thus relatively immune to changes in technology;

For certain applications, XML also has the following weaknesses:

1) Its syntax is verbose and redundant. This can hurt human readability and application efficiency, and yields higher storage costs. It can also make XML difficult to apply in cases where bandwidth is limited (though compression can reduce the problem in some cases).

2) Parsers are designed to recurse arbitrarily nested data structures and perform additional checks to detect improperly formatted or differently ordered syntax or data. This causes a significant overhead for most basic uses of XML. Furthermore



IBM Software Group | DB2 Information Management Software

XML – a powerful but costly language

- Example: look up IBM stock price

DATA PASSED: IBM 94

5 chars Traditional

```
SAMPLE CODE:
<SOAP-ENV:Envelope
xmlns:SOAP-ENV=
    "http://www.w3.org/2001/06/soap-envelope"
SOAP-ENV:encodingStyle=
    "http://www.w3.org/2001/06/soap-encoding">
  <SOAP-ENV:Body>
```

272 chars with XML and SOAP

```
<Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>YWJjZGxma3NqamRlZmZnaGlvcztkbGZramFzZGw7Cg==</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
    YWJjZGxma3NqamZmZnaGlvcztkbGZramFzZGw7Cg==
  </SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Over 919 chars to add digital signature only

And many more chars to add encryption

37

What follows here is a SHORT example of the structure of XML.

This is not script for the customer, rather a very high understanding of the structure of XML.

An Example XML Document

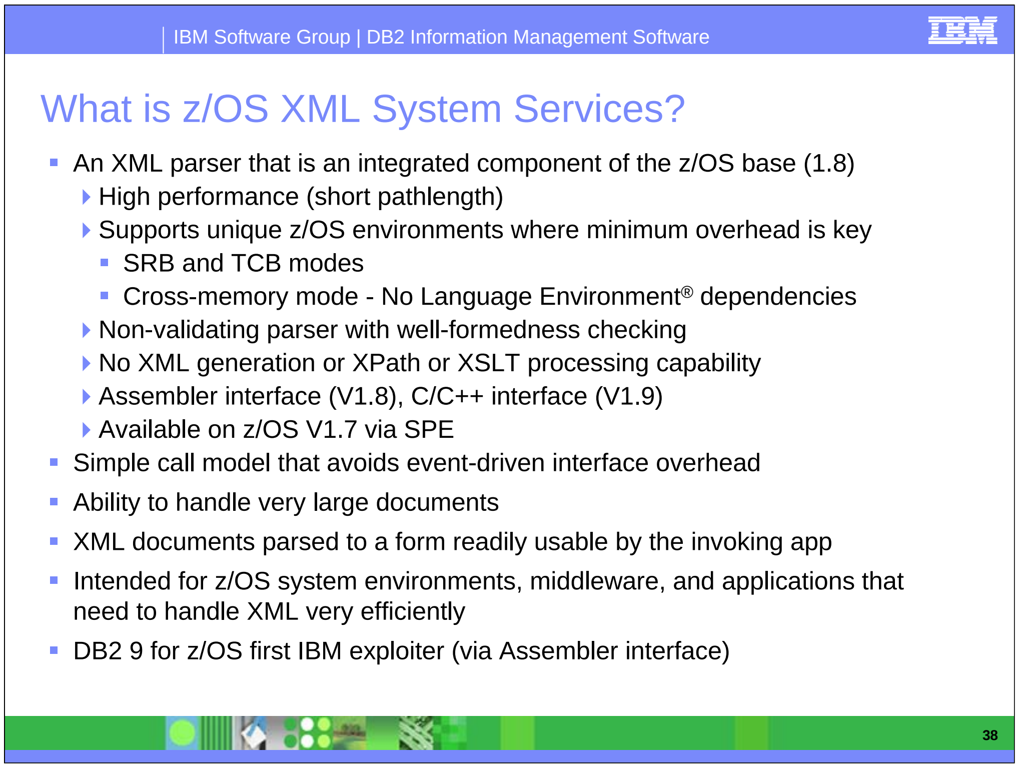
XML documents use a self-describing and simple syntax.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The first line in the document - the XML declaration - defines the XML version and the character encoding used in the document. In this case the document conforms to the 1.0 specification of XML and uses the ISO-8859-1 (Latin-1/West European) character set.

The next line describes the root element of the document (like it was saying: "this document is a note"):

<note>


IBM Software Group | DB2 Information Management Software

What is z/OS XML System Services?

- An XML parser that is an integrated component of the z/OS base (1.8)
  - High performance (short pathlength)
  - Supports unique z/OS environments where minimum overhead is key
    - SRB and TCB modes
    - Cross-memory mode - No Language Environment® dependencies
  - Non-validating parser with well-formedness checking
  - No XML generation or XPath or XSLT processing capability
  - Assembler interface (V1.8), C/C++ interface (V1.9)
  - Available on z/OS V1.7 via SPE
- Simple call model that avoids event-driven interface overhead
- Ability to handle very large documents
- XML documents parsed to a form readily usable by the invoking app
- Intended for z/OS system environments, middleware, and applications that need to handle XML very efficiently
- DB2 9 for z/OS first IBM exploiter (via Assembler interface)

38

In z/OS V1.8, IBM introduced a new system component of z/OS, z/OS XML System Services (z/OS XML), a system-level XML parser integrated with the base z/OS operating system and designed to deliver an optimized set of services for parsing XML documents (z/OS XML has also been made available on z/OS V1.7). The initial beneficiaries of this system component are middleware and applications requiring high performance non-validating XML parsing. z/OS XML may currently be accessed by an Assembler programming interface and one of the first exploiters, DB2 9 for z/OS, uses this Assembler interface for XML native support. IBM plans to add C/C++ support for z/OS XML with z/OS V1.9 (satisfying the Statement of Direction in Software Announcement 206-039, dated February 28, 2006).

Simple call model that avoids event driven interface overhead

z/OS XML System Services is a lower level interface – z/OS XML parser looks at the whole document and parses everything it can thus avoiding interactive overhead.

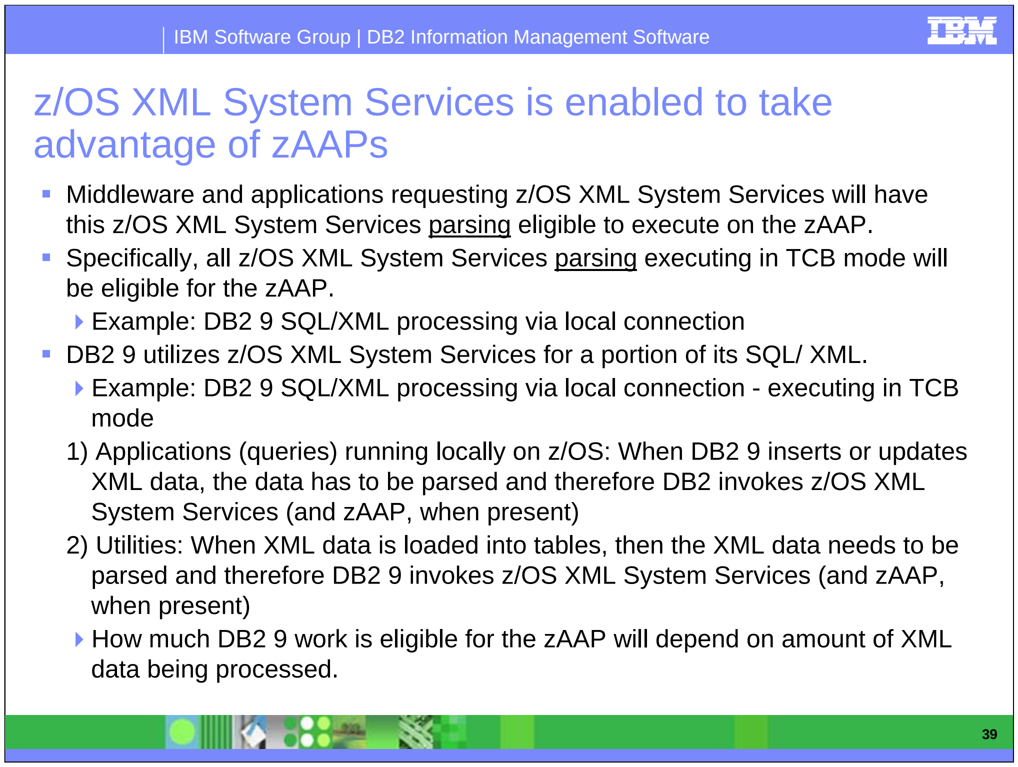
Other parsers on parse interactively thus causing some performance overhead.

Ability to handle very large documents

a lot of parses have difficulty handling large documents – they want to bring in the whole doc but cannot... z/OS XML has large input and output buffers and can process more/ larger docs... in the event the buffer fills z/OS XML can request more memory from the application

XML documents parsed to an output buffer in a form readily usable by the invoking application

The binary form is sharable within z/OS systems and provides possible performance improvement by avoiding relocation overhead. (ie you move the XML parsing from one address space to another, no need


IBM Software Group | DB2 Information Management Software

z/OS XML System Services is enabled to take advantage of zAAPs

- Middleware and applications requesting z/OS XML System Services will have this z/OS XML System Services parsing eligible to execute on the zAAP.
- Specifically, all z/OS XML System Services parsing executing in TCB mode will be eligible for the zAAP.
  - Example: DB2 9 SQL/XML processing via local connection
- DB2 9 utilizes z/OS XML System Services for a portion of its SQL/ XML.
  - Example: DB2 9 SQL/XML processing via local connection - executing in TCB mode

  1) Applications (queries) running locally on z/OS: When DB2 9 inserts or updates XML data, the data has to be parsed and therefore DB2 invokes z/OS XML System Services (and zAAP, when present)

  2) Utilities: When XML data is loaded into tables, then the XML data needs to be parsed and therefore DB2 9 invokes z/OS XML System Services (and zAAP, when present)
  - How much DB2 9 work is eligible for the zAAP will depend on amount of XML data being processed.

39

Middleware and applications requesting z/OS XML System Services will have this z/OS XML System Services parsing eligible to execute on the zAAP.   Only the XML parsing being performed by z/OS XML System Services (a base element of z/OS) is eligible for zAAP.

•Please note, there is a Java-based XML parser in the IBM SDK 1.3 (and above) – the technology is called XML4J. This Java-based XML parser is already fully eligible for zAAP because it is Java workload. Other C++, COBOL, PL/I and roll-your own parsers are not eligible for zAAP.

Specifically, all z/OS XML System Services parsing executing in TCB mode will be eligible for the zAAP.

•Example: DB2 9 SQL/XML processing via local connection

•Currently DB2 9 is the only IBM exploiter of z/OS XML System Services (via an Assembler interface)

DB2 9 utilizes z/OS XML System Services for a portion of its SQL/ XML.

Please note, not all DB2 9 XML processing is done using z/OS XML System Services. XML Validation is not eligible.

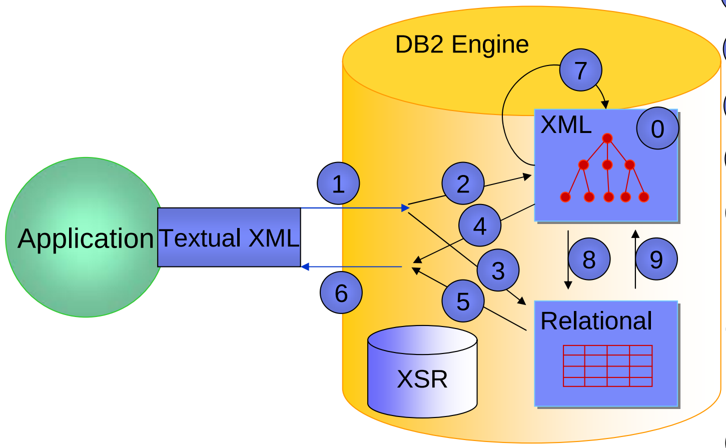
•Example: DB2 9 SQL/XML processing via local connection. When executing locally, DB2 9 is already executing in TCB mode

•Types of DB2 9 XML parsing workloads that are eligible for zAAP would be

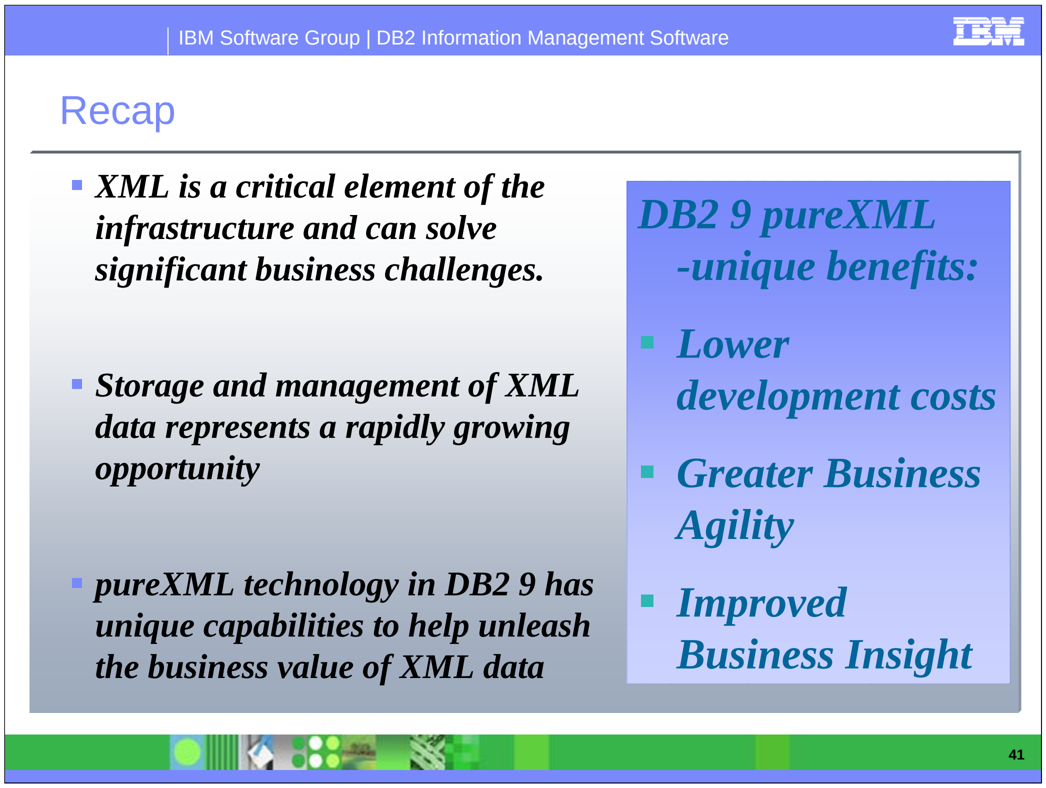
1) Applications (queries) running locally on z/OS

When DB2 9 inserts or updates XML data, the data has to be parsed and therefore DB2 invokes z/OS XML System Services (and zAAP, when

Summary of SQL/XML Features



0	XML Storage
1	Bind in XML
2	Store as XML
3	Shred into Rel
4	Retrieve XML
5	Publish XML
6	Bind out XML
7	XML to XML
8	XML to Rel
9	Rel to XML



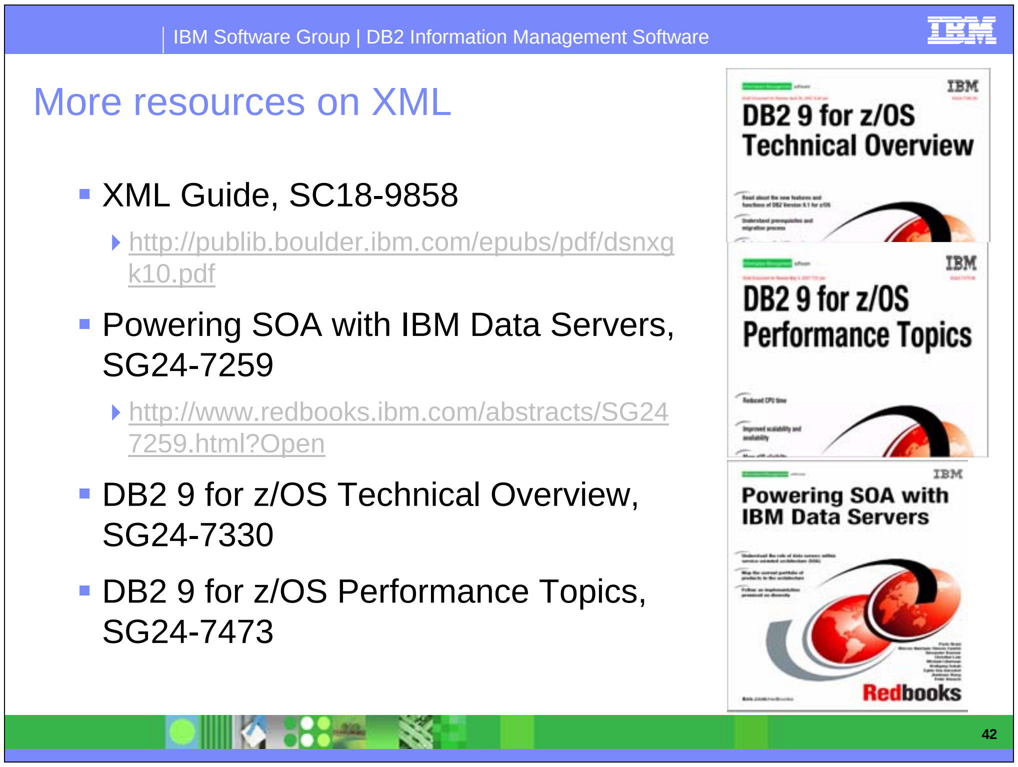
Recap

- *XML is a critical element of the infrastructure and can solve significant business challenges.*
- *Storage and management of XML data represents a rapidly growing opportunity*
- *pureXML technology in DB2 9 has unique capabilities to help unleash the business value of XML data*

DB2 9 pureXML -unique benefits:

- *Lower development costs*
- *Greater Business Agility*
- *Improved Business Insight*

DB2 9 is a major step in XML support, providing a hybrid database which has substantial support for relational and XML. As Kevin Campbell stated “This is XML without compromise.”



IBM Software Group | DB2 Information Management Software

More resources on XML

- XML Guide, SC18-9858
  - http://publib.boulder.ibm.com/epubs/pdf/dsnxgk10.pdf
- Powering SOA with IBM Data Servers, SG24-7259
  - http://www.redbooks.ibm.com/abstracts/SG247259.html?Open
- DB2 9 for z/OS Technical Overview, SG24-7330
- DB2 9 for z/OS Performance Topics, SG24-7473

DB2 9 for z/OS Technical Overview
DB2 9 for z/OS Performance Topics
Powering SOA with IBM Data Servers
Redbooks

42

There are many other resources on XML. See DeveloperWorks.

http://www.ibm.com/developerworks/db2/products/db2zos/

http://www.ibm.com/developerworks/db2/library/techarticle/dm-0705malaika/