



TRAINING & CONSULTING

---

## XML Support in DB2 V9

Dutch DB2 User Group (DDUG), 09/10/2007

**Ludo Van den dries**

ABIS Training & Consulting  
[www.abis.be](http://www.abis.be)  
[training@abis.be](mailto:training@abis.be)

2007

Document number: DB2\_XML\_01.fm  
10 October 2007

Address comments concerning the contents of this publication to:  
ABIS Training & Consulting, P.O. Box 220, B-3000 Leuven, Belgium  
Tel.: (+32)-16-245610, Fax: (+32)-16-245639

© Copyright ABIS N.V.



# XML Support in DB2 V9

---

- the wonders of XML
- the wonders in DB2 V9

- **XML = Extensible Markup Language**
- **a universal way of structuring text documents**
- **XML 1.0 (1998/2000)**
- **from World Wide Web Consortium (W3C)**
- **markup language, cf. SGML, HTML, ...**
- **objectives:**  
**independent, interoperable, simple, human readable, ...**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XML document

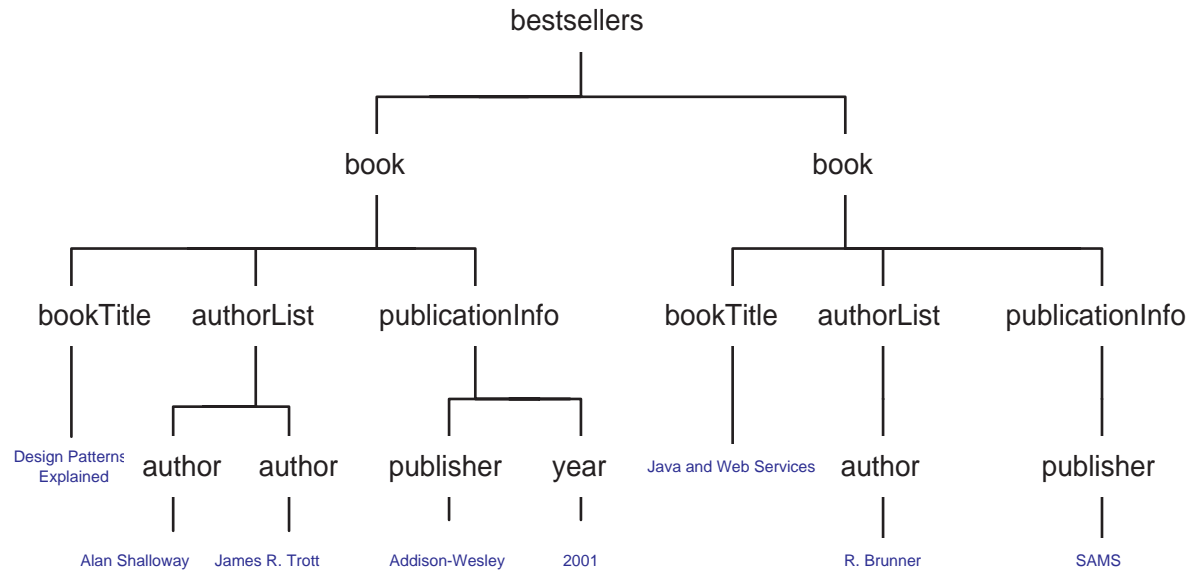
---

```
<?xml version="1.0" encoding="UTF-8"?>
<bestsellers>
  <book>
    <bookTitle>Design Patterns Explained</bookTitle>
    <authorList>
      <author>Alan Shalloway</author> <author>James R. Trott</author>
    </authorList>
    <publicationInfo>
      <publisher>Addison-Wesley</publisher> <year>2001</year>
    </publicationInfo>
    <ISBN>0-201-71594-5</ISBN>
  </book>
  <book>
    <bookTitle>Java and Web Services</bookTitle>
    <authorList>
      <author>R. Brunner</author>
    </authorList>
    <publicationInfo> <publisher>SAMS</publisher> </publicationInfo>
    <ISBN>0-672-32363-X</ISBN>
  </book>
</bestsellers>
```

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XML document (..)



- structured text content
- document = tree, varying composition
- user defined tags (> extensible)
- elements, root element, whitespace
- simple construction (> well-formedness)

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# Unicode

## One single character set (see [www.unicode.org](http://www.unicode.org))

	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	040A	040B	040C	040D	040E	040F
0	È	А	Р	а	р	è	Ѡ	Ѳ	Ѵ	Г	К	У	І	Ă	З	Û
1	Ë	Б	С	б	с	ë	ѡ	ѳ	ѵ	Г	к	у	Ж	ă	з	ü
2	Ђ	В	Т	в	т	ђ	Ѣ	Ѧ	Ѵ	Ѧ	Ѧ	Ѧ	Ѧ	Ă	Й	Ў
3	Ѓ	Г	У	г	у	ѓ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ă	й	ў
4	Є	Д	Ф	д	ф	є	Ї	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Æ	Й	Ї
5	Є	Е	Х	е	х	є	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	æ	й	ï
6	І	Ж	Ц	ж	ц	і	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ѧ	Ë	Ö	

### XML Support in DB2 V9

- >>> XML basics
- Defining/imposing document structure
- Namespaces
- XML-based standards
- Parsing
- Transforming
- Searching
- Storing
- Ubiquitous XML
- >>> XML in DB2 V9 <<<
- INSERTing XML
- SHREDDING
- UPDATEing
- >> Four ways to query XML in DB2
- Plain SQL
- SQL/XML
- XQuery
- XQuery with embedded SQL
- >> Optimization
- Epilogue: When to store XML in the database

## Unicode (..)

---

### Several encodings:

- UTF-16 (2 / 4 bytes)
- UTF-8 (from 1 to 4 bytes)
- ISO-8859-x (1-byte)
- etc...

### Encoding indicated in xml declaration:

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

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database



## More nodes & trees

---

```
<?xml version="1.0" encoding="UTF-8"?>
<?myProg blablaba ?>
<Enrolments>
  <!-- here is some comment -->
  <Enrolment nr="17" source="web">
    <Course>Java and XML</Course>
    <Student>
      <FirstName>Mary</FirstName>
      <LastName>Jones</LastName>
      <Company>Nepsis</Company>
    </Student>
    <InvoiceDetails>
      <Price currency="USD">350</Price>
      <Enroller> ....
      ...
    </Enrolment>
  <Enrolment>...
```

attributes, processing instruction, comment,

node tree, root node (/)

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XML Document (..)

---

```
<PLAY>
  <TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE>
  <PERSONAE>
    <TITLE>Dramatis Personae</TITLE>
    <PERSONA>CLAUDIUS, king of Denmark.</PERSONA>
    <PERSONA>HAMLET, son to the late, and nephew to the present king.</PERSONA>
    <PERSONA>POLONIUS, lord chamberlain.</PERSONA>
    <PGROUP>
      <PERSONA>VOLTIMAND</PERSONA>
      <PERSONA>CORNELIUS</PERSONA>
      <PERSONA>ROSENCRANTZ</PERSONA>
      <GRPDESCR>courtiers.</GRPDESCR>
    </PGROUP>
  ...
  <SPEECH>
    <SPEAKER>LORD POLONIUS</SPEAKER>
    <LINE>I hear him coming: let's withdraw, my lord.</LINE>
  </SPEECH>
  <STAGEDIR>Exeunt KING CLAUDIUS and POLONIUS</STAGEDIR>
  <STAGEDIR>Enter <PERSONA>HAMLET</PERSONA>, heavily in doubt. </STAGEDIR>
  <SPEECH>
    <SPEAKER>HAMLET</SPEAKER>
    <LINE>To be, or not to be: that is the question:</LINE>
    <LINE>Whether 'tis nobler in the mind to suffer</LINE>
    <LINE>The slings and arrows of outrageous fortune,</LINE>
    <LINE>Or to take arms against a sea of troubles,</LINE>
  ...
  ...
</PLAY>
```

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XML Document (..)

---

**<document>**

**<body>**

**<heading style="head20\_1">Objectives </heading>**

**<paragraph style="simple">This course consists of 2 parts.</paragraph>**

**<paragraph style="simple">In the first part, participants will become familiar with the Enterprise Java (J2EE) platform, the basis for the WebSphere architecture. The integration of Java with existing (enterprise) systems is also covered.**

**...**

**</document>**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

### Two standards:

- **DTD (Document Type Definition)**
- **XML Schema**

**A set of rules defining what specific documents should contain on which place (mandatory and optional things).**

### Can be used to

- **validate a document: does it follow the specific rules?**
- **guide you while composing/editing a document**
- ...

**Note: well-formedness vs. validity**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## DTD example

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bestsellers SYSTEM "E:\XML\Books.dtd">
<bestsellers>
  <book>
    <bookTitle>Design Patterns Explained</bookTitle>
    <authorList>
      <author>Alan Shalloway</author>
      ...
    </authorList>
  </book>
</bestsellers>
```

---

```
<!ELEMENT bestsellers (book+)>
<!ELEMENT book (bookTitle, authorList, publicationInfo, ISBN)>
<!ELEMENT bookTitle (#PCDATA)>
<!ELEMENT authorList (author+)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publicationInfo (publisher, year?)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
```

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Schema Example

---

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bestsellers">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="book" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="bookTitle" type="xs:string"/>
              <xs:element name="authorList" >
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="author" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="publicationInfo" >
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="publisher" type="xs:string"/>
                    <xs:element name="year" type="xs:string" minOccurs=...
```

etc ...

### XML Support in DB2 V9

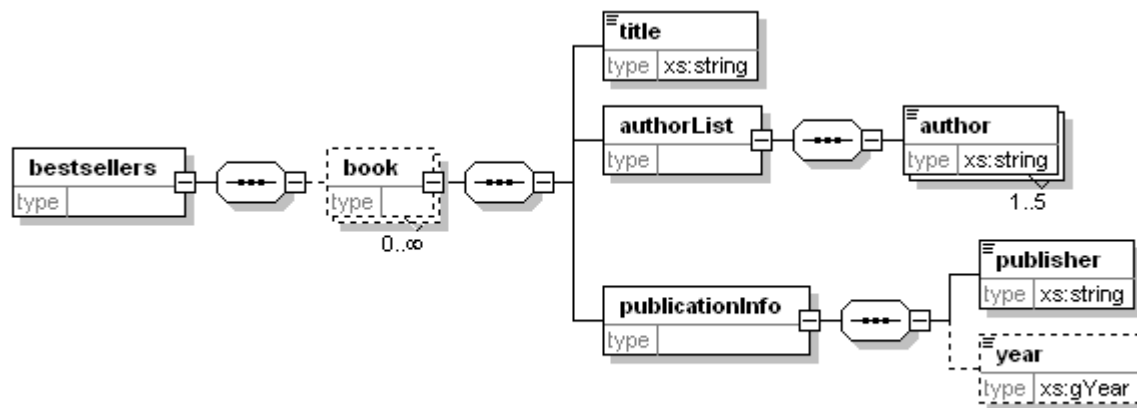
1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XML Schema (..)

## XML by itself!

### Powerful definition of

- complex structures
- data types



Generated by XmlSpy

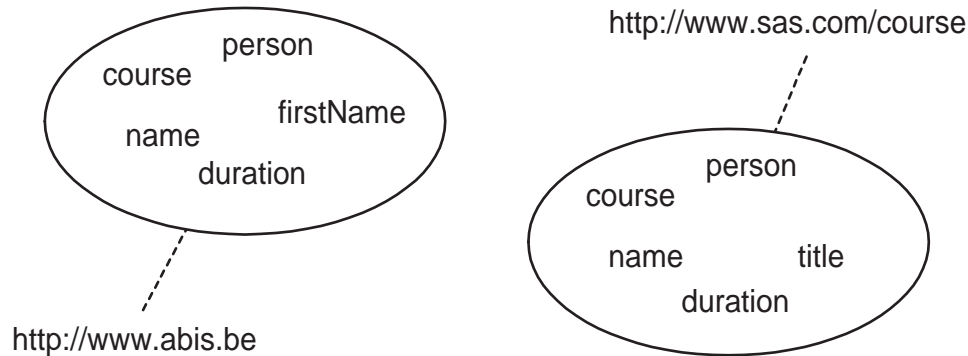
[www.altova.com](http://www.altova.com)

**Note: document < > schema**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

Namespace = vocabulary as defined by 1 or more schemas



```
<courseList xmlns:abis="http://www.abis.be" xmlns:sas="www.sas.com/course">
```

```
<abis:course>
```

```
<abis:name>XML basics</abis:name>
```

```
<abis:duration>3</abis:duration>
```

```
</abis:course>
```

```
<sas:course>
```

```
<sas:title>XML and SAS</sas:title>
```

```
</sas:course>
```

```
<abis:course>
```

```
etc .....
```

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database



### Many auxiliary/derived/applied standards

- **Basic (W3C):**
  - **XSLT & Xpath**
  - **XQuery**
  - **XLink & XPointer**
- **Horizontal application:**
  - **SVG (Scalable Vector Graphics)**
  - **SMIL (Synchronized Multimedia Integration Language)**
  - **RSS / Atom**
- **Vertical application (cf. OASIS, xml.org) :**
  - **XBRL (Extensible Business Reporting Language)**
  - **HR-XML (Human Resources)**
  - **GML (Geography Markup Language)**

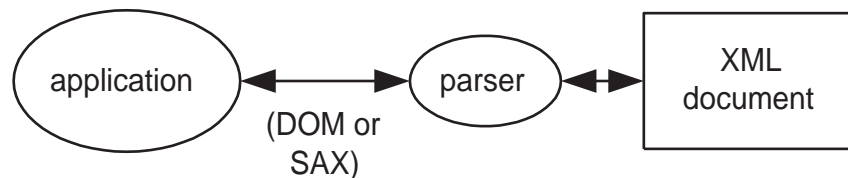
### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

**Parsing = getting logical tree data out/in physical XML text**

**Applications leave this task to specialized ‘parsers’**

**Most popular standard APIs for XML parsers: DOM and SAX**



**E.g.**  
**Microsoft MSXML**  
**Java Xerces**

**XML Support in DB2 V9**

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XSL = eXtensible Stylesheet Language

### Aspects:

- transformation recipe (XSLT)
- formatting for rendition (XSL-FO)
- accessing the underlying XML structure (XPath)

### XML Support in DB2 V9

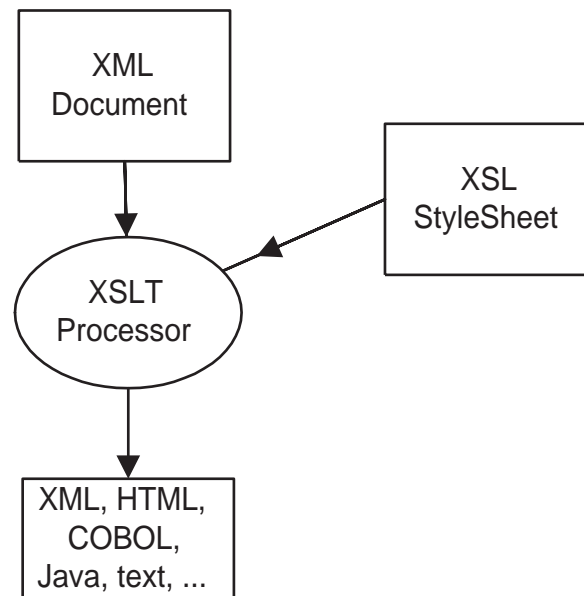
1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XSLT

---

## XSL stylesheet:

- patterns to search in input document
- templates to produce the output document
- is XML itself (of course...)



## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XPath:

- locating information inside a document
- for query, pointing, linking, transformation, schema, ...

## XMLQuery (aka XQuery):

- powerful querying in document & entire XML databases
- emerging standard, many predecessors...
- query only

## Versions:

- XPath 1.0 + XSLT 1.0 (1999)
- XPath 2.0 + XSLT 2.0 + XQuery 1.0 (2007)

## XSLT versus XQuery: competing standards?

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XPath

---

## Locating things in a tree

**Result = node-set, boolean, string, number**

### One step:

- **axis::node-test[predicate][predicate][...]**

**child::Person[position()=3]**

**descendant::Book[attribute::number="55"]**

### Complete location path:

- **step1 / step2 / step3 / ...**

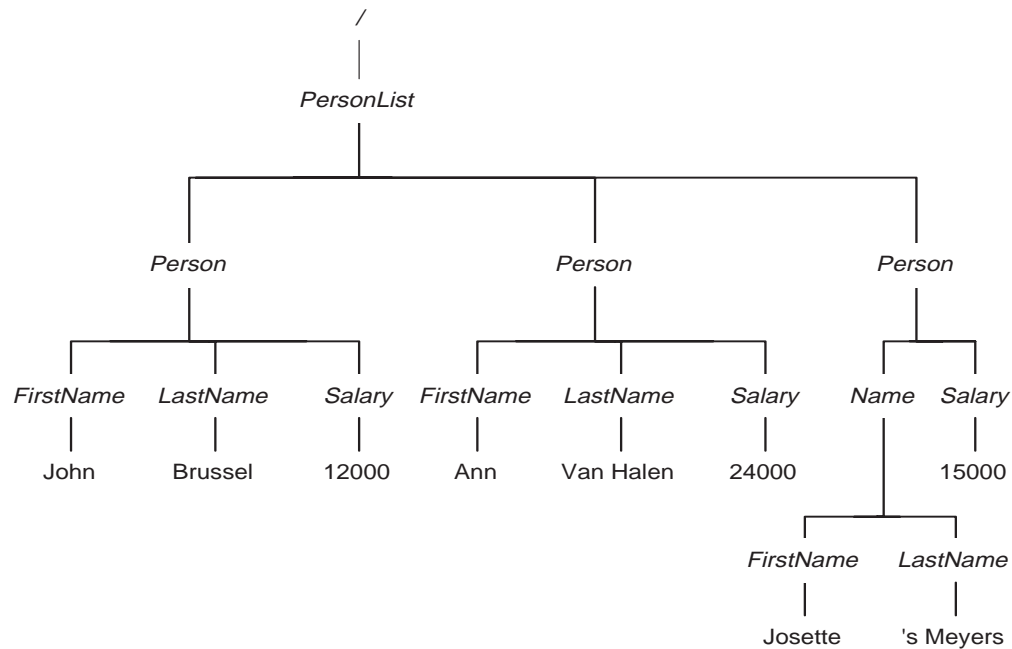
**PersonList/Person[@nr="55"]/FirstName**

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XPath (..)

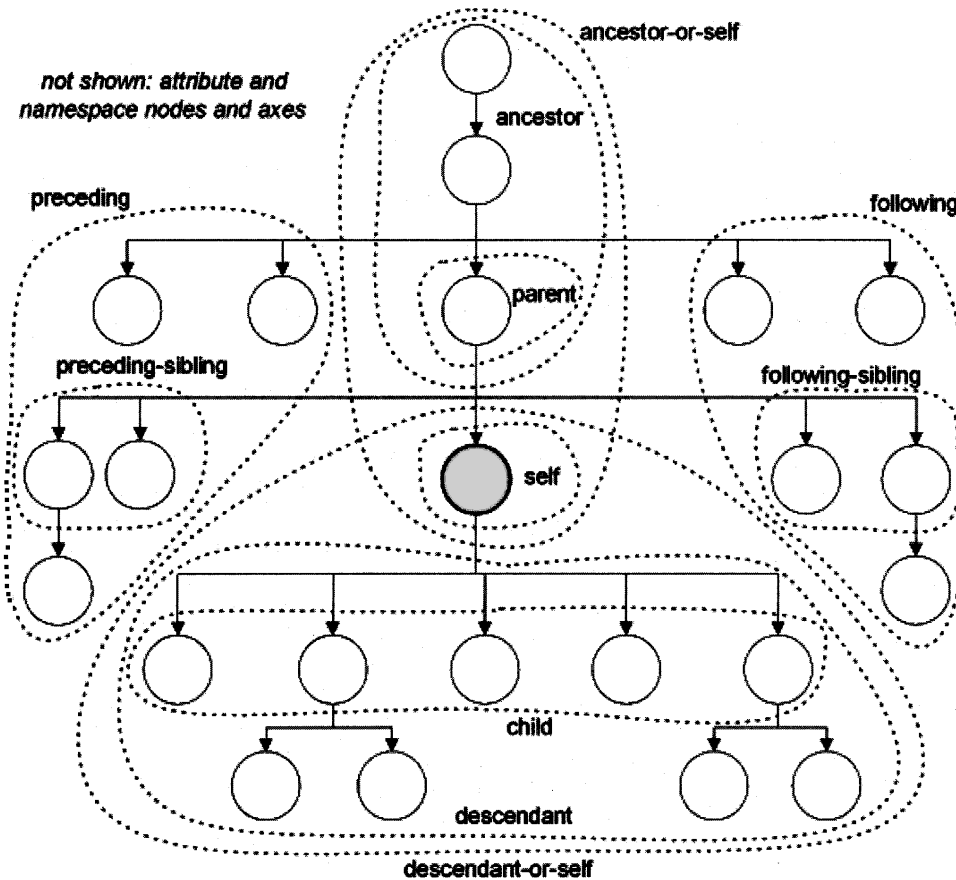
## Example



### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# XPath axis



## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database



## XSLT vs. XQuery (..)

---

### XSLT

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="bestsellers">
    <authorList>
      <xsl:apply-templates select="//book[@category='fiction']"/>
    </authorList>
  </xsl:template>
  <xsl:template match="book">
    <xsl:copy-of select="descendant::author"/>
  </xsl:template>
</xsl:stylesheet>
```

### XQuery: FLWOR (for, let, where, order by, return)

```
<authorList>
{ for $b in document('bestsellers.xml')/bestsellers/book
  where $b/@category='fiction'
  return $b//author }
</authorList>
```

#### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Main options:

- **simply as files (\*.XML)**
- **in native XML database:**
  - **added support e.g. for querying, transforming, linking**
  - **Tamino, X-Hive, XIS ...**
- **in XML-enabled 'conventional' RDBMS**
  - **DB2, Oracle, SQL Server, ...**
  - **diverse techniques:**
    - stuffing into CLOBs & BLOBs
    - shredding into columns & tables
  - **XML as 'second-class' data**

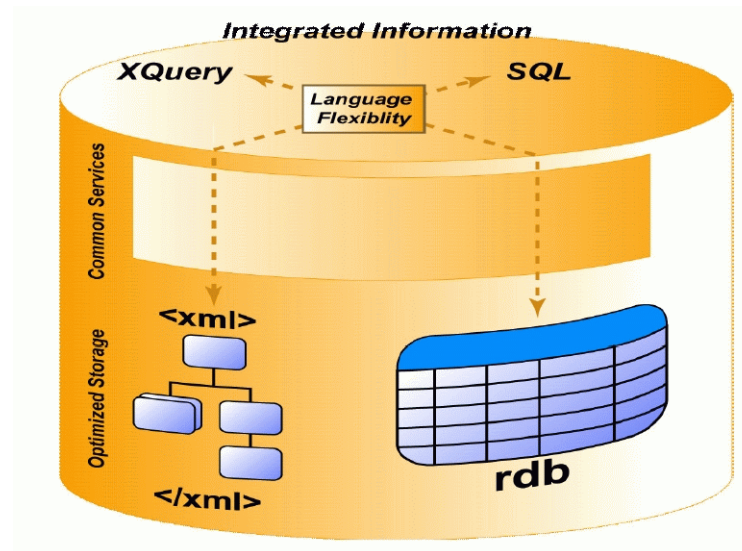
## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

# Storing XML in DB2 V9

## pureXML (Viper)

## DB2 as the hybrid database



## XML: first-class data & first-class queries

## hybrid storage model -- transparent

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## pureXML (..)

---

```
CREATE DATABASE test USING CODESET utf-8 TERRITORY US
```

```
CREATE TABLE orders (oid INT, otype CHAR(2), ocustomer XML, oinfo XML)
```

```
INSERT INTO orders (oid, otype, ocustomer, oinfo) VALUES (5, 'EX',  
'<customer><firstname>John</firstname><lastname>Williams</lastname></customer>',  
'<order>  
  <orderdate>2006-07-07</orderdate>  
  <shipnote>Fragile Contents</shipnote>  
</order>'  
);
```

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XML is used/usable for many purposes:

- document oriented and 'data' oriented data
- storage and import/export/transport
- content management
- transformation, rendering, publishing
- user interfaces (Web, HTML, WML, ...)
- configuration files
- descriptive programming
- Web Services ( > SOA)
- RSS, Atom, ...

## Tools:

dedicated (XMLSpy) <> integrated (WSAD, RAD, XML Developer Workbench, ..., Visual Studio .NET, ...)

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Following topics will be illustrated:

- inserting & updateing XML
- ways to query XML
- optimization

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

- the TRADITIONAL way!
  - inserting a document
  - copying a document form an existing table
  - constructing a document using appropriate functions

```
INSERT INTO customer (id, name)
VALUES (102,
       XMLDOCUMENT(XMLELEMENT(name "CustomerName", 'De Greef')));
```

```
INSERT INTO customer (id, name)
VALUES (101, (SELECT name FROM old_customers
             WHERE id = 1));
```

- PARSING is implicit!
  - invoked automatically to check document is 'well-formed'
  - to be invoked manually if whitespace in the XML document has to be preserved
    - default behaviour: do not preserve!

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## INSERTing XML (ii)

---

- **XML document validation**

- **requires an EXISTING schema, i.e. schema needs to be REGISTERED [XSD can be dropped, not updated]**

stored in a XML schema repository (XSR) - DB2 catalog tables

- **procedure:**

- register the primary schema, using:
  - > fully qualified name for the schema file
  - > SQL identifier

**register xmlschema http://customer from D:\xsd\customer.xsd as customer.xsd**

- register 'dependant' XSDs, ie. XSDs included/imported into the primary one registered above
- complete the registration: checks for content of XSD, well-formedness, ...

**complete xmlschema customer.xsd**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database





## XML data can be SHRED into relational tables

XML is a 'received' message

Its content has to be stored in relational tables

- **use annotated XML schema doc to describe SHREDding rules, ie. relate XML to relational structures, e.g.**
  - tables to be used
  - datatypes of columns in the tables
  - ...
- **register the schema (cfr.above), indicating the schema will be used for decomposition**
- **invoke the decompose command**

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Two ways:

- **SQL UPDATE: the full XML column!**
- **supplied stored procedure (DB2XMLFUNCTIONS.XMLUPDATE)**

## XMLUPDATE:

- **partial updates to the document in a XML column**
- **using the stored procedure, supply:**
  - the full XML column to be updated
  - the XML elements/attributes (in the doc in the column) to be inserted/updated/deleted

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

### ‘Pure’:

- plain SQL
- plain XQuery

### ‘Blended’:

- SQL/XML (= SQL with XPath/XQuery inside)
- XQuery with SQL inside

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

```
select ocustomer  
from orders  
where oid = 5;
```

```
select oinfo  
from orders  
where otype in ('XE', 'AB');
```

- **use when:**
  - **full document retrieval - no fragments required**
  - **search criteria on non-XML columns**
  - **insert, update, delete of full documents**
- **why not use CLOB, VARCHAR?**
- **DB2 Net search extender available in plain SQL, combined with e.g. Xpath expressions**

relational predicates - insert, update, delete - parameter markers - full text search function calls

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

**SQL language standard:**

- defines the XML datatype
- defines functions for manipulating XML data:
  - v8 [rel data => XML] :  
xmlelement, xmlattribute, xmlforest, ...
  - v9 [use Xquery, Xpath in SQL] :  
xmlquery, xmltable, xmlexists

```
INSERT INTO nss (id, doc) VALUES (999, XMLDOCUMENT(
XMLELEMENT (name "Name",
XMLATTRIBUTES ('MALE' as "gender"),
XMLFOREST ('Steve' as "FirstName",
'Ferrington' as "LastName"))));
```

```
select oinfo
from orders o
where xmlexists('$o//order[orderdate = "2006-07-07"]'
passing o.oinfo as "o");
```

**XML Support in DB2 V9**

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XMLTable

---

Function that 'converts' a XML document into a table structure, ie:

- input: an XQuery expression
- output: a table containing columns of any SQL type

**Conclusion: generates a relational view into XML documents!**

(cf. shredding < > publishing)

**Views!**

```
SELECT id, firstname, lastname, zipcode
FROM xps, XMLTABLE(
'for $cust in $d/Customer
return $cust' passing xps.doc as "d"
COLUMNS firstname VARCHAR(20) path 'Name/FirstName/text()',
lastname VARCHAR(20) path 'Name/LastName/text()',
zipcode VARCHAR(10) path 'Address/Zip/text()')
as nameszip
WHERE zipcode = '95030'
```

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## SQL/XML (iii)

---

- **advantages**
  - **integration/interaction between SQL and XML based criteria**
    - join based on XML and column criteria
    - 'group by' only implemented using SQL/XML
  - **full-text search condition supported**
  - **use of parameter markers**
- **disadvantages**
  - **difficult for transformation purposes, yet ...**  
xslttransform (-> XSLT stylesheet based transformation)

XML predicates - relational predicates - combining predicates - join (XML & relational, XML & XML) - insert, update, delete - parameter markers - XML aggregation, grouping - full text search function calls

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database



## All DB2 APIs support XQuery:

- CLI/ODBC (including therefore db2clp)
- embedded SQL
- JDBC
- .Net

### xquery

```
for $order in db2-fn:xmlcolumn("ORDERS.OINFO")/order
where $order/orderdate="2006-07-07"
return $order/shipnote;
```

### xquery

```
<ul> {
for $y in db2-fn:xmlcolumn('CLIENTS.CONTACTINFO')/Client/Address
order by $y/zip
return <li>{$y}</li>
} </ul>
```

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XQuery (ii)

---

- **advantages**
  - **XML only apps; no SQL, relational constructs required**
  - **easy to embed in XML documents to be created**
  - **join, union XML documents**
- **disadvantages**
  - **no DB2 Net search extender support**
  - **no support for parameter markers - yet**
  - **no support for UDFs**

XML predicates - join (XML & XML) - transforming XML

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

### Purpose: embed SQL in XML (aka XQuery)

#### XQUERY

```
for $order in db2-fn:sqlquery("select order from orders where otype='EX'")/id
where $order/orderdate="2006-07-07"
return $order/shipnote;
```

#### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## XQuery with embedded SQL (ii)

---

- **advantages**
  - **based on a valid SQL query**
  - **DB2 Net search extender supported**
  - **integration/interaction between SQL and XML based criteria**
  - **parameter marker support**
  - **UDF support**

XML predicates - relational predicates - combining predicates - join (XML & relational, XML & XML) - XML aggregation, grouping - full text search function calls

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

### New in DB2 v9: 2 language parsers

- XQuery
- SQL

**Result of parse: a common, language-neutral, internal representation of queries.**

**Traditional, cost based query optimization is performed [based on traditional stats]!**

- **cost-based access plan generation**
- **query rewrite**
- **operator selection**
- **index selection**
- ...

### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Optimization (ii)

---

### New - XML INDEXes

```
CREATE INDEX odindex ON orders(orderinfo)
GENERATE KEY USING XMLPATTERN '/order/orderdate' as SQL DATE;
```

- specify table name
- specify column name
- specify XML pattern to be used for identifying target to be indexed, using XPath notation
- traditional behaviour

Observation: internal indexes are created and maintained by DB2 when using XML data type columns:

*regions index* (data retrieval - one per table)

*XML column path index* (data (subset) retrieval, document indexing - one per XML column)

#### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

## Optimization (iii)

---

### **RUNSTATS has been enhanced to support XML type columns!**

- **collected info**
  - XML document level statistics
  - XML node level statistics
- **collected by means of sampling**
- **info gathered:**
  - Path distribution data [# rows with XPath # occurrences of XPath is encountered within each XML]
  - Path-Value distribution data [value of XPath expression, # rows that contain that value, # of value occurrences within each document]
- **traditional use**
- **when requesting runstats on XML columns only, statistics describing other column data is maintained!**

#### XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database

**Hierarchical data**

**Semi-structured data**

**Document/narrative data**

**Many different schemas**

**Large schemas (with sparsely populated attributes/elements)**

**Quickly evolving schema**

**Forms-based applications**

**Data with nulls & multiple values**

**Existing industry standard schemas for XML**

## XML Support in DB2 V9

1. >>> XML basics
2. Defining/imposing document structure
3. Namespaces
4. XML-based standards
5. Parsing
6. Transforming
7. Searching
8. Storing
9. Ubiquitous XML
10. >>> XML in DB2 V9 <<<
11. INSERTing XML
12. SHREDDING
13. UPDATEing
14. >> Four ways to query XML in DB2
15. Plain SQL
16. SQL/XML
17. XQuery
18. XQuery with embedded SQL
19. >> Optimization
20. Epilogue: When to store XML in the database