# DB2 pureXML

| | |
|---|---|
| **Module ID** | 10116 |
| **Length** | 45 minutes |

**For questions about this presentation contact** askdata@ca.ibm.com

January 30, 2015

© 2015 IBM Corporation

# Disclaimer

**© Copyright IBM Corporation 2015. All rights reserved.**

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, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.



© 2015 IBM Corporation

# Module Information

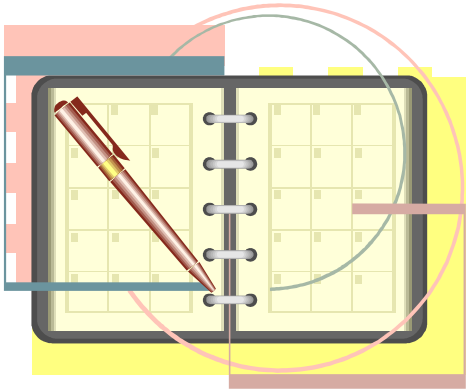
- You should have completed or acquired the necessary knowledge for the following modules in order to complete this module:
  - DB2 Fundamentals

- After completing this module, you should be able to:
  - Explain the concept of:
    - XML storage in DB2
    - XQuery
    - XPath


© 2015 IBM Corporation

# Module Content

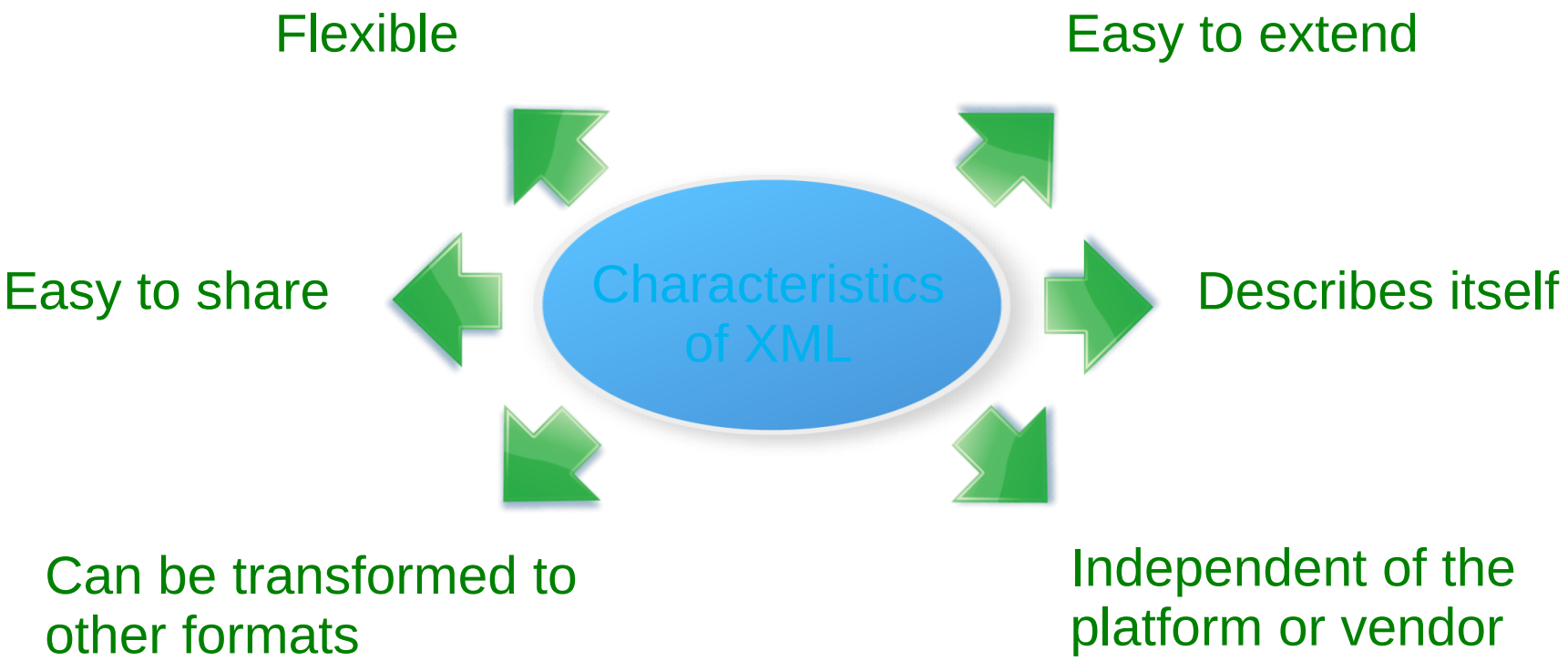
- **pureXML in DB2**
  - pureXML engine
  - XML storage
- **Basic Operations**
- **XQuery and SQL/XML**
- **XML Indexes in DB2**
- **Summary**




© 2015 IBM Corporation

# What is XML?

- eXtensible Markup Language
  - XML is a language designed to describe data
- A hierarchical data model

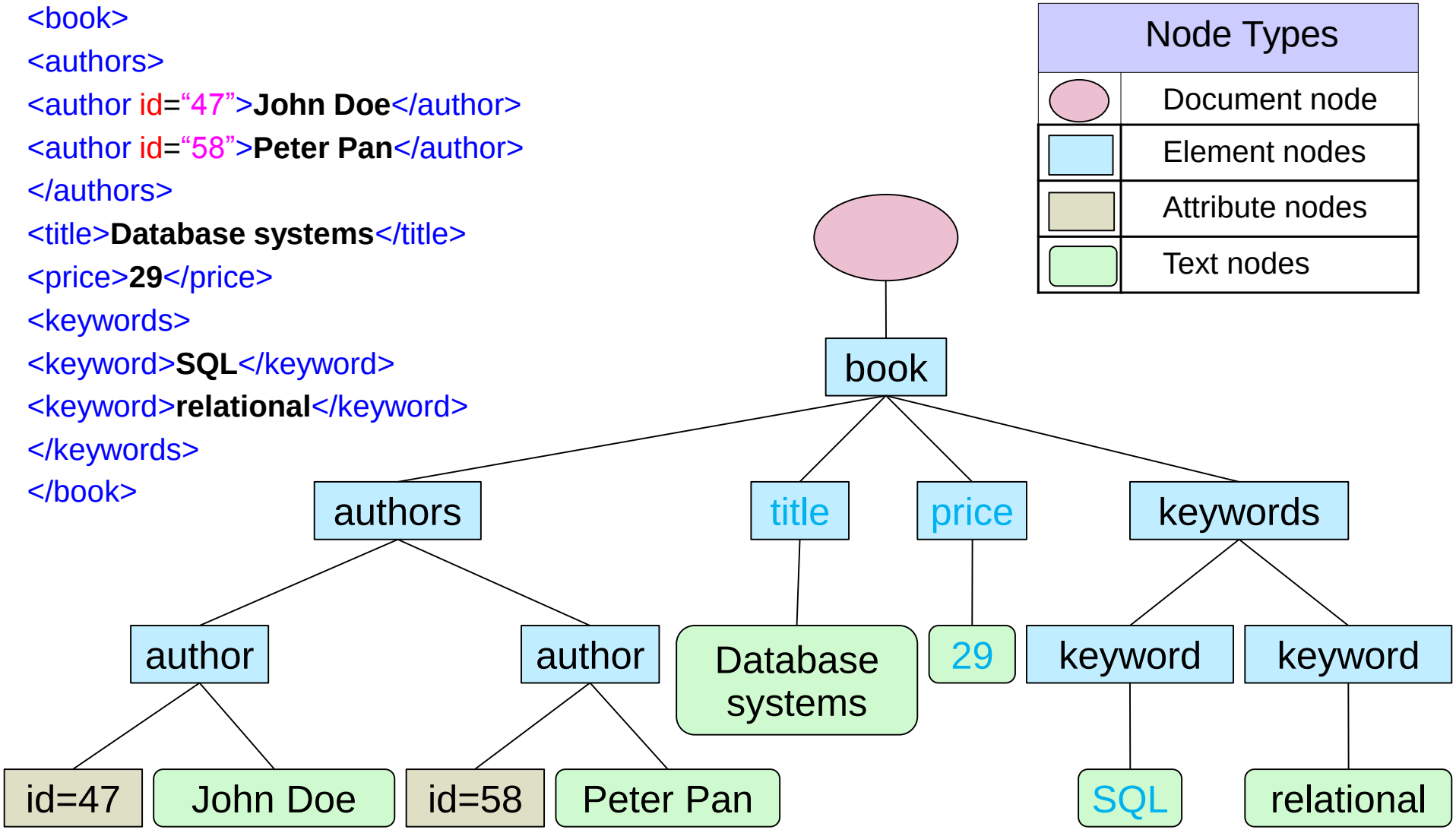




© 2015 IBM Corporation

# The XML Data Model: Node Types

```
<book>
<authors>
<author id="47">John Doe</author>
<author id="58">Peter Pan</author>
</authors>
<title>Database systems</title>
<price>29</price>
<keywords>
<keyword>SQL</keyword>
<keyword>relational</keyword>
</keywords>
</book>
```





© 2015 IBM Corporation

# Relational Versus Hierarchical (XML) Model

| Relational | Hierarchical (XML) |
|---|---|
| Relational data is flat | XML data is nested. |
| Relational model is set oriented. Sets are unordered. | XML retrieves sequences (the order matters) |
| Relational data is structured. | XML data is semi-structured. |
| Relational data has a strong schema, unlikely to change often. | XML data has a flexible schema, appropriate for constant changes. |
| Use NULL for an unknown state. | NULLS don't exist. Don't add any XML element. |
| Based on the ANSI/ISO industry standards. | Based on the W3C industry standards. |

With DB2 you can store and process both.

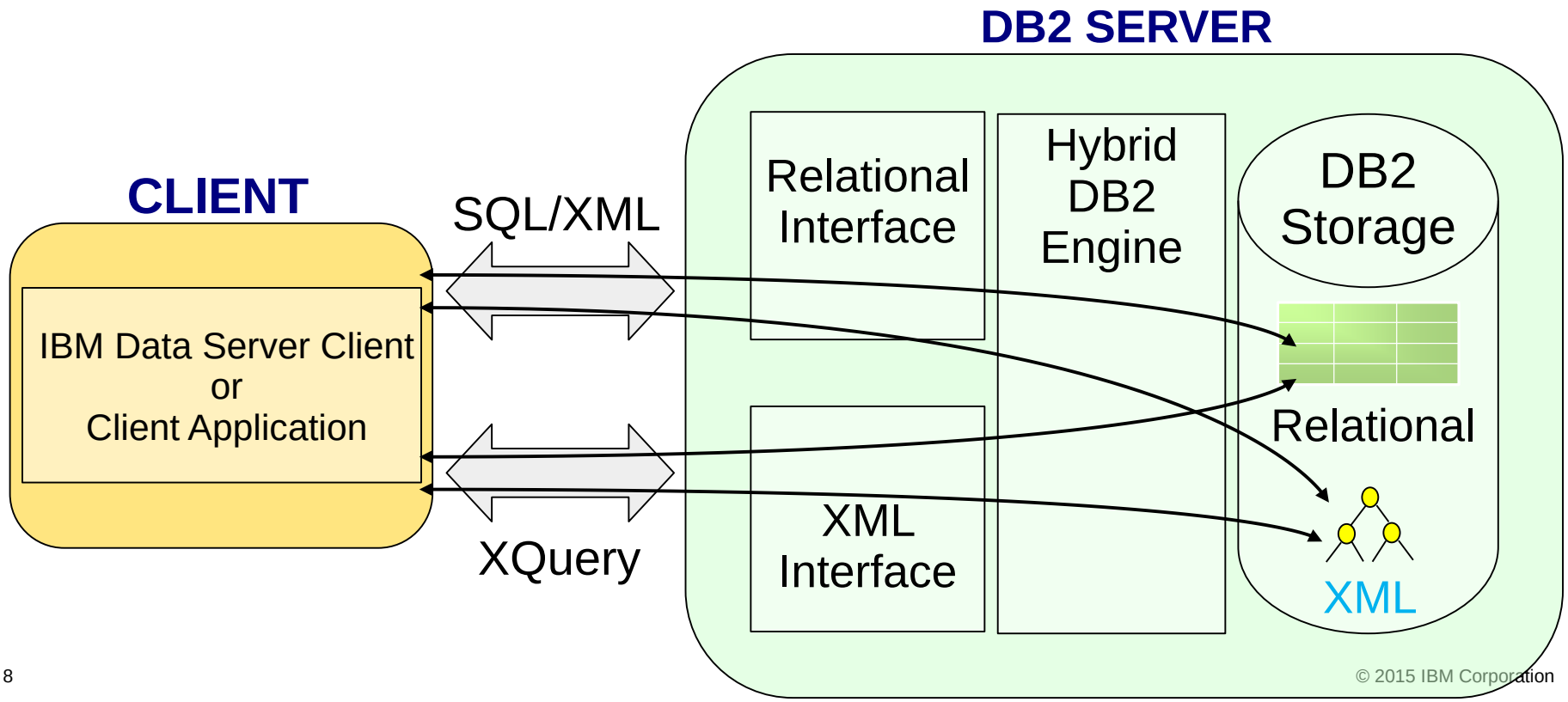


© 2015 IBM Corporation

# The pureXML Engine

- Relational and XML data are stored differently, but closely linked
- XML Capabilities in all DB2 components
- Combine XML and relational data
- Binary XML format is supported for faster data transmission between Java application and DB2 server

**New in DB2 10**



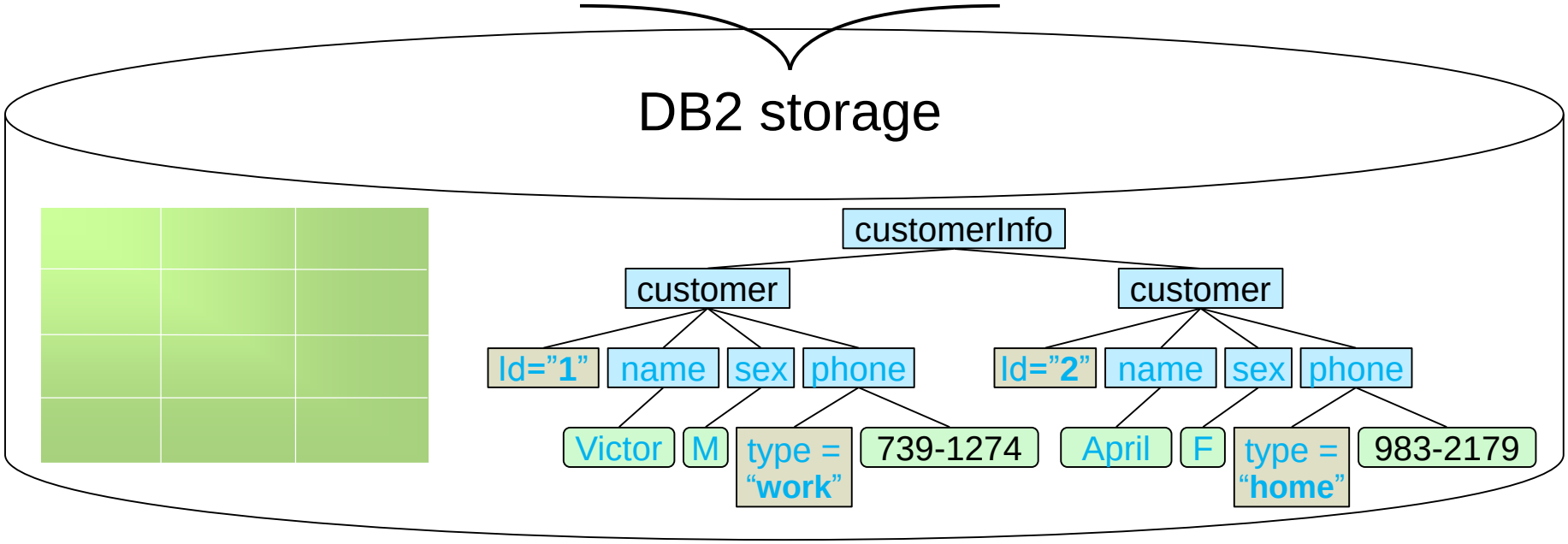


© 2015 IBM Corporation

# pureXML Storage in DB2: XML Data Type

```
CREATE TABLE dept (deptID VARCHAR(30), ..., custDoc XML)
```

| deptID | ... | custDoc |
|---|---|---|
| A001 | .. | |
| A002 | .. | |

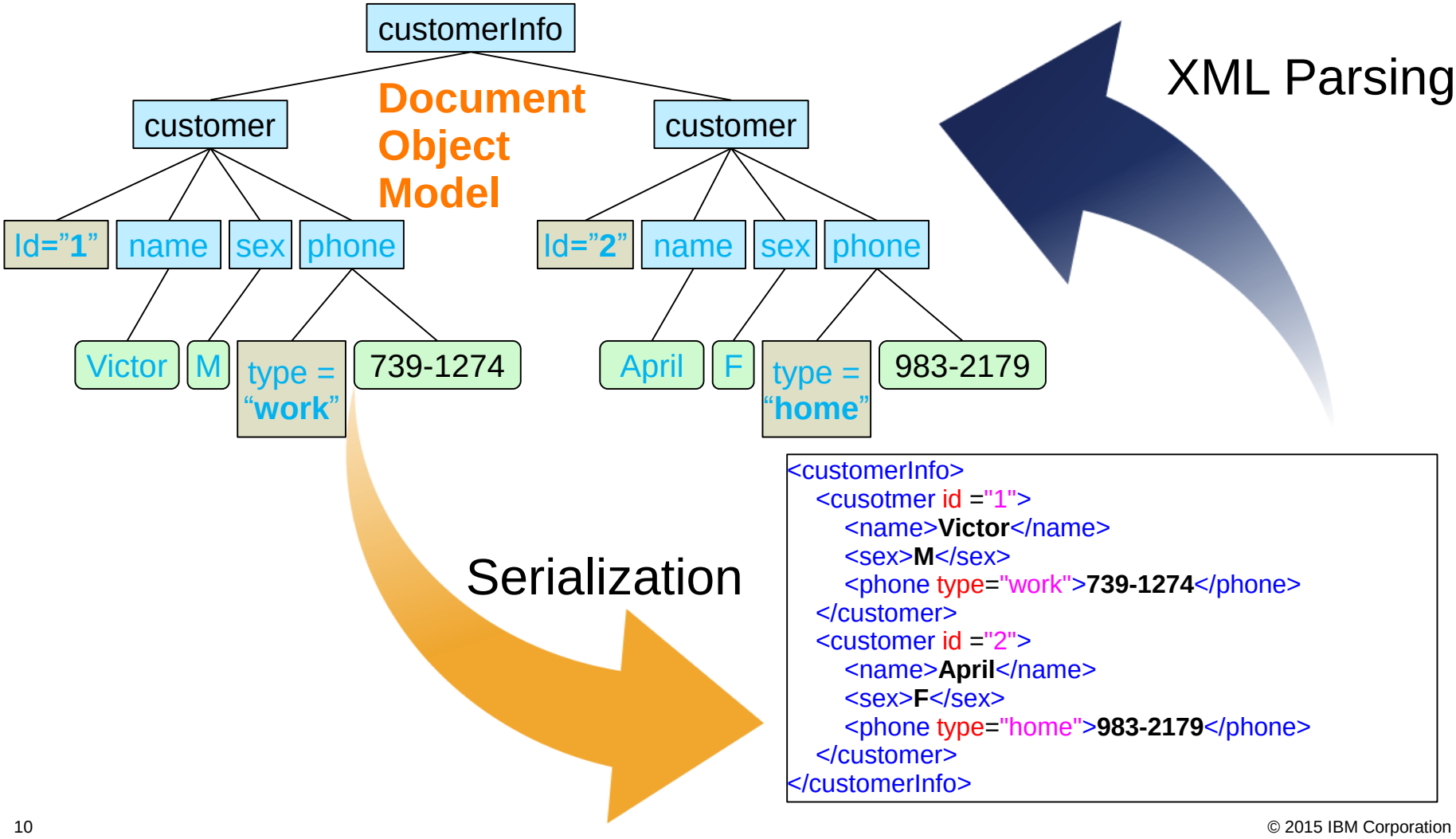




© 2015 IBM Corporation

# Storage as XML Document Tree

- Documents are stored in parsed representation

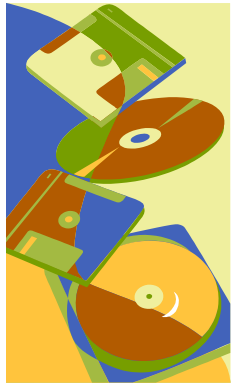




© 2015 IBM Corporation

# Storing XML: Native XML Storage

- Relational columns are stored in relational format (tables)
- Values of XML columns are stored natively (XML documents in parsed hierarchical format), encoded in UTF-8
- XML values are stored:
  - In the pureXML storage area (by default). A descriptor points from row to XML storage
  - Or inlined in row, if enabled for XML document smaller than 32KB.
- No XML parsing for query evaluation!
- XML data can be compressed



**New in DB2 10**



© 2015 IBM Corporation

# How to get Data In?

- Implicit XML parsing:
  - Inserting data of XML data type info a column

```
 INSERT INTO dept VALUES
('PR27', …, '<dept>…<emp>…</emp>…</dept>')
```

- Explicit XMLPARSE
  - Transform XML value from serialized (text) form into internal representation.
  - Tell system how to treat whitespaces (strip/preserve)
    - Default is 'Strip WHITESPACE'

```
 INSERT INTO dept VALUES ('PR27', xmlparse(document
'<a>...</a>'));
 INSERT INTO dept VALUES ('PR27',
    xmlparse(document '<a>...</a>' preserve whitespace));
```



© 2015 IBM Corporation

# Deleting XML Data

- DELETE
  - Will delete every XML document for a row

```
DELETE FROM dept WHERE deptID='A001'
```

  - You can also delete based on the XML content

```
DELETE FROM dept WHERE
    XMLEXISTS ('$d//phone[type="Home"]'
                passing INFO as "d")
```

- Note: Setting an XML column to NULL deletes the XML document

```
UPDATE dept SET custDoc = NULL WHERE deptID='A001'
```


© 2015 IBM Corporation

# Import

DEL file to import

```
import from /data/dept.del of del
XML from /data/xmlfiles
insert into dept
```

**/data/dept.del**

```
1000,"<XDS FIL='deptdoc.001.xml' />"
1001,"<XDS FIL='deptdoc.002.xml' />"
1002,"<XDS FIL='deptdoc.003.xml' />"
1003,"<XDS FIL='deptdoc.004.xml' />"
1004,"<XDS FIL='deptdoc.005.xml' />"
```

**/data/xmlfiles**

```
/data/xmlfiles/deptdoc.001.xml
/data/xmlfiles/deptdoc.002.xml
/data/xmlfiles/deptdoc.003.xml
/data/xmlfiles/deptdoc.004.xml
/data/xmlfiles/deptdoc.005.xml
```

Directory that includes the XML files that are referenced in the DEL file

**dept**

| | |
|---|---|
| 1000 | <dept><employee><name>John Doe</name><br><address><street>555 Bailey Ave</street><city>…</city><zip>95141</zip><br></address>…</employee></dept> |
| 1001 | <dept><employee><name>Kathy Smith</name> … |
| 1002 | <dept><employee><name>Jim Noodle …. |



© 2015 IBM Corporation

# Export

```
EXPORT TO /data/dept.del of DEL
XML TO /data/xmlfiles
XMLFILE deptdoc
MODIFIED BY XMLINSEPFILES
SELECT * FROM dept
```

- DEL file to output
- Directory to place XML files
- Base name for exported XML files
- Store each XML document in separate file (Optionally: Concatenate all XML documents in one large file.)
- What to export

**dept**

| | |
|---|---|
| 1000 | <dept><employee><name>John Doe</name><br><address><street>555 Bailey Ave</street><city>…</city><zip>95141</zip><br></address>…</employee></dept> |
| 1001 | <dept><employee><name>Kathy Smith</name> … |
| 1002 | <dept><employee><name>Jim Noodle …. |

**/data/dept.del**

```
1000,"<XDS FIL='deptdoc.001.xml' />"
1001,"<XDS FIL='deptdoc.002.xml' />"
1002,"<XDS FIL='deptdoc.003.xml' />"
1003,"<XDS FIL='deptdoc.004.xml' />"
1004,"<XDS FIL='deptdoc.005.xml' />"
```

**/data/xmlfiles**

```
/data/xmlfiles/deptdoc.001.xml
/data/xmlfiles/deptdoc.002.xml
/data/xmlfiles/deptdoc.003.xml
/data/xmlfiles/deptdoc.004.xml
/data/xmlfiles/deptdoc.001.xml
```


© 2015 IBM Corporation

# SQL/XML and XQuery

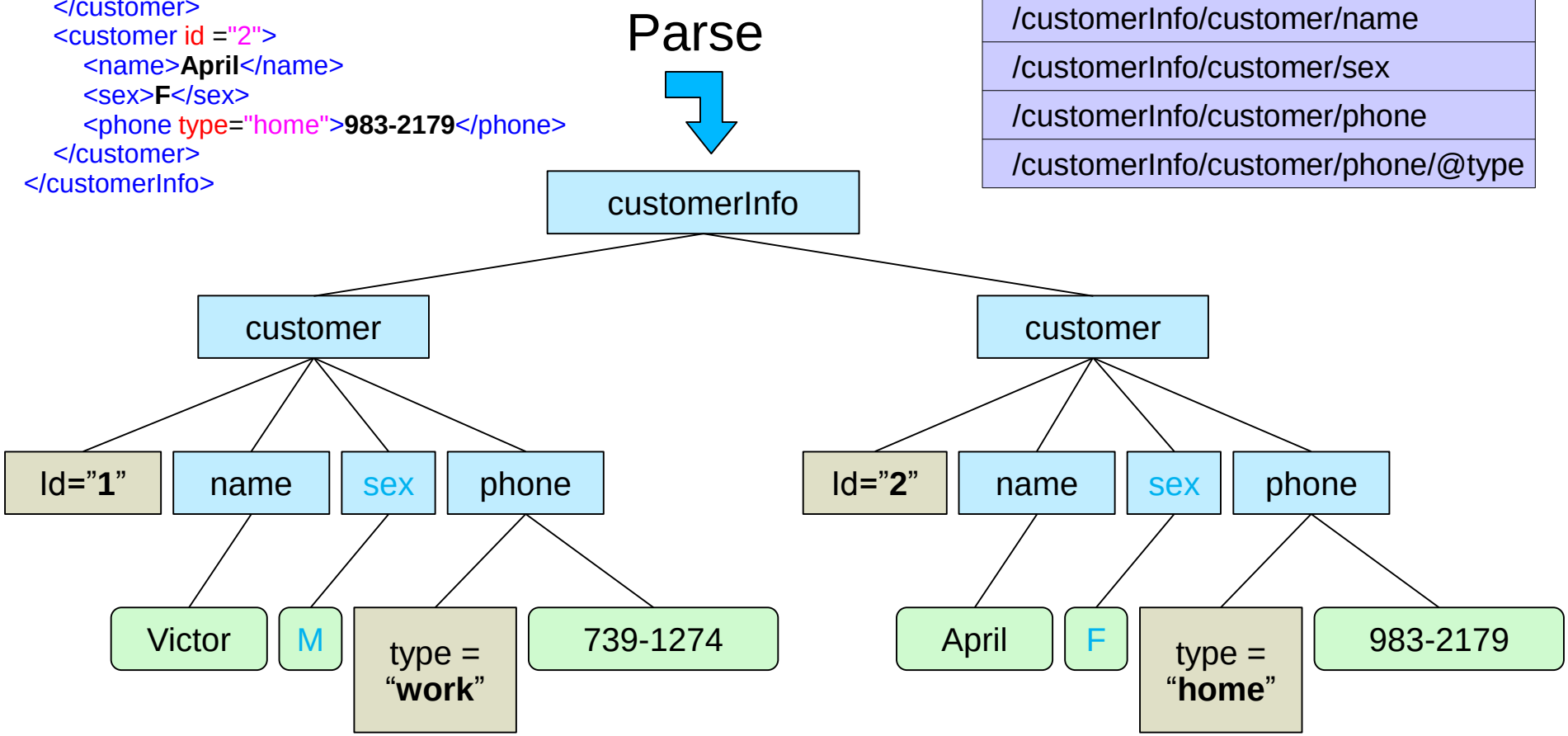
- XPath
  - Cornerstone for both XQuery and SQL/XML standard
  - Provides ability to navigate within XML documents
- XQuery
  - Query language for XML data
  - Can be embedded into SQL via SQL/XML standard
  - DB2 LUW : Can be used via XQuery interface
- SQL/XML
  - Provides functions to work with both XML and relational data at the same time.
  - Embed XPath and XQuery expressions into SQL to query XML data
  - Turn XML data into relational (table) form
  - Produce XML out of relational data
  - Enhanced XML casting in DB2 10
  - Improved XML query performance in DB2 10


© 2015 IBM Corporation

# XPath

```
<customerInfo>
   <customer id ="1">
      <name>Victor</name>
      <sex>M</sex>
      <phone type="work">739-1274</phone>
   </customer>
   <customer id ="2">
      <name>April</name>
      <sex>F</sex>
      <phone type="home">983-2179</phone>
   </customer>
</customerInfo>
```

| Path Table |
|---|
| / |
| /customerInfo |
| /customerInfo/customer/@id |
| /customerInfo/customer/name |
| /customerInfo/customer/sex |
| /customerInfo/customer/phone |
| /customerInfo/customer/phone/@type |





© 2015 IBM Corporation

# Some Common XPath Expressions

```
<customerInfo>
   <customer id ="1">
      <name>Victor</name>
      <sex>M</sex>
      <phone type="work">739-1274</phone>
   </customer>
   <customer id ="2">
      <name>April</name>
      <sex>F</sex>
      <phone type="home">983-2179</phone>
   </customer>
</customerInfo>
```

| | |
|---|---|
| **/** | Selects from the root node. |
| **//** | Selects nodes in the document from the current node that match the select. |
| **text()** | Specifies the text node under an element. |
| @ | Specifies an attribute. |
| ***** | Matches any element node. |
| @* | Matches any attribute node. |
| **[ … ]** | Predicates |

| XPath Expression | Result Description | Result |
|---|---|---|
| `/customerInfo/*/phone/text()` | Selects the text node under the **phone** element of **customerInfo** | **739-1274**<br>**983-2179** |
| `/customerInfo//phone/@type` | Selects the `type` attribute under the **phone** element of **customerInfo** | **work**<br>**home** |
| `/customerInfo/customer[1]/phone/text()` | Selects the **phone** element text node under the first **customer** of **customerInfo** | **739-1274** |
| `/customerInfo//phone[@type='home']` | Selects all **phone** elements under **cusomterInfo** which has an attribute named type with a value of 'home' | **<phone type = "home”>**<br>**983-2179**<br>**</phone>** |

# SQL/XML Functions

- XQuery can be invoked from SQL
  - **`XMLQUERY()`**
  - **`XMLTABLE()`**
  - **`XMLEXISTS()`**
- By executing XQuery expressions from within the SQL context, you can:
  - Operate on parts of stored XML documents instead of entire XML documents
  - Enable XML data to participate in SQL queries
  - Operate on both relational and XML data
  - Apply further SQL processing to the returned XML values


© 2015 IBM Corporation

# Query and Extract XML data

- XMLQUERY
  - Scalar function, applied once to each qualifying document
  - Evaluates an XPath (or XQuery) expression
  - Input arguments can be passed into the XQuery (e.g. column names, constants, parameter markers)
  - Returns a sequence of 0, 1 or multiple items from each document

***XMLCUSTOMER***

| CID | INFO |
|---|---|
| 1001 | |
| 1002 | |
| 1003 | |

```
SELECT
  XMLQUERY('$i/customerinfo/name'
  PASSING INFO AS "i")
FROM
  CUSTOMER
```

| 1 |
|---|
| <name>...</name> |
| <name>...</name> |
| ... |



© 2015 IBM Corporation

# Query and Extract XML Data

- XMLTABLE
  - Creates a temporary SQL table using XML data

***XMLCUSTOMER***

| CID | INFO |
|---|---|
| 1001 | |
| 1002 | |
| 1003 | |

```
SELECT T.*
FROM XMLTABLE(
    'db2-fn:xmlcolumn("XMLCUSTOMER.INFO")/customerinfo'
     COLUMNS "NAME" VARCHAR (20) PATH 'name',
             "STREET" VARCHAR (20) PATH 'addr/street',
             "CITY" VARCHAR (20) PATH 'addr/city'
) AS T
```

```
<customerinfo>
  <name>John Smith</name>
  <addr country="Canada">
    <street>Fourth</street>
    <city>Calgary</city>
    <prov-state>Alberta</prov-
state>
    <pcode-zip>M1T 2A9</pcode-zip>
  </addr>
  <phone type="work">
    963-289-4136
  </phone>
</customerinfo>
```

| NAME | STREET | CITY |
|---|---|---|
| Amir Malik | Young | Toronto |
| John Smith | Fourth | Calgary |
| … | … | … |



© 2015 IBM Corporation

# SQL/XQuery XML Data for SQL Developers

- XMLEXISTS
  - Predicate that tests if an XQuery expression returns a sequence

***XMLCUSTOMER***

| CID | INFO |
|---|---|
| 1001 | |
| 1002 | |
| 1003 | |

```
SELECT CID, INFO
FROM XMLCUSTOMER WHERE
XMLEXISTS(
    '$d/customerinfo[name = "John Smith"]'
    passing INFO as "d")
```

```
<customerinfo>
  <name>John Smith</name>
  <addr country="Canada">
    <street>Fourth</street>
    <city>Calgary</city>
    <prov-state>Alberta</prov-
state>
    <pcode-zip>M1T 2A9</pcode-zip>
  </addr>
  <phone type="work">
    963-289-4136
  </phone>
</customerinfo>
```

| CID | INFO |
|---|---|
| 1003 | |



© 2015 IBM Corporation

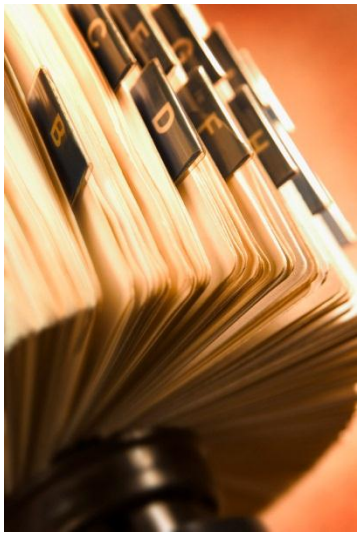
# Indexing

- Several data types supported
  - **VARCHAR** and **VARCHAR HASHED**
  - **DATE** and **TIMESTAMP**
  - **DOUBLE**, **DECIMAL**, **INTEGER**

**New in DB2 10**

- Nodes identified using XPath-like pattern
  - Navigation within documents similar to query
  - Supports functional XML indexes
    - **fn:exists()**
    - **fn:upper-case()**

**New in DB2 10**



```
CREATE INDEX custname_IX ON XMLCUSTOMER(INFO)
GENERATE KEY USING XMLPATTERN '/customerinfo/name'
AS SQL VARCHAR(60);

CREATE INDEX custname_IX ON XMLCUSTOMER(INFO)
GENERATE KEY USING XMLPATTERN
'/customerinfo/name/fn:upper-case(.)'
AS SQL VARCHAR(60);
```

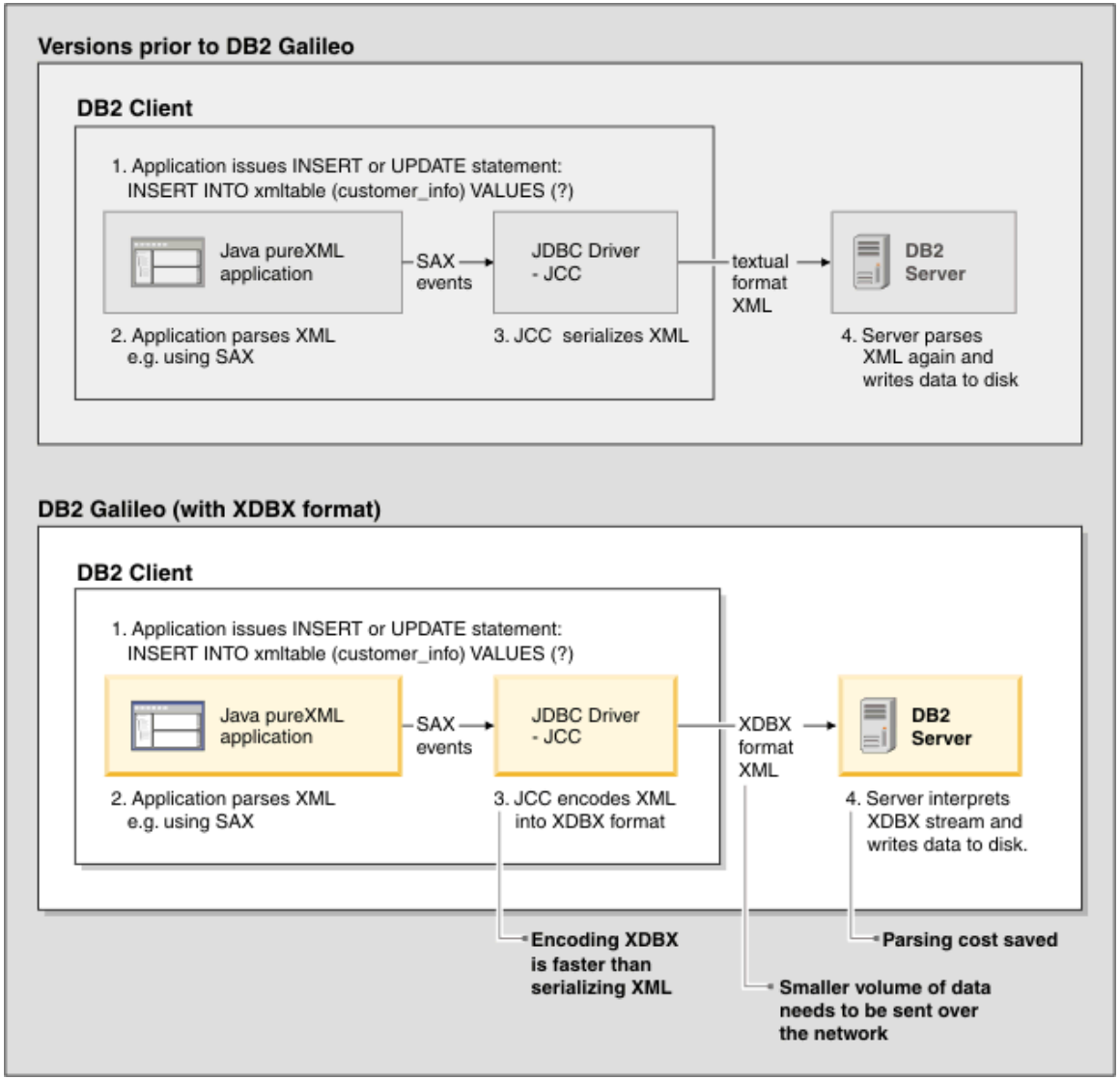


© 2015 IBM Corporation

# Improved Insert/Query Performance with Binary XML Format

**New in DB2 10**

- Improve end to end performance of client/server applications using JDBC or SQLJ interfaces
- Avoid conversion between XML as text and internal format (client / server) by introducing XDBX
  - Already used in DB2 for z/OS
  - Simple to generate, simple to parse
  - Provides improved end to end performance
- Transparent to applications
- Storage and retrieval of binary XML data requires version 4.9 or later of the IBM Data Server Driver for JDBC and SQLJ





© 2015 IBM Corporation

# Summary

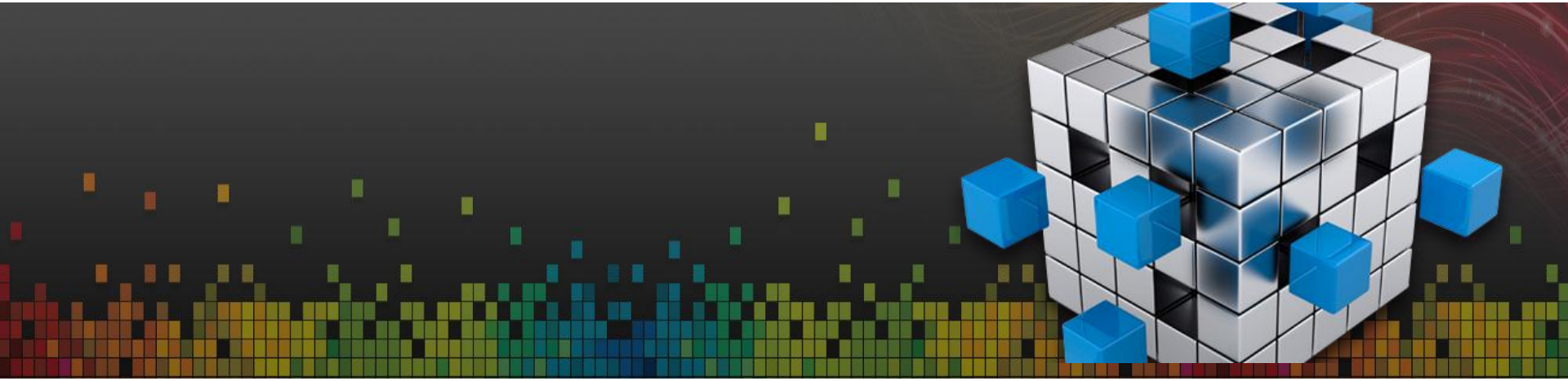
- Native XML hierarchical storage
  - No shredding, no CLOBs, no BLOBs required
  - Optimized for XPath and XQuery processing

- High performance
  - Superior indexing technology
  - No parsing of XML data at query runtime

- Fully integrated XML and relational processing
  - Seamlessly query various types of data at once
  - SQL/XML functions for combining SQL and XQuery
  - No internal translation of XQuery into SQL


© 2015 IBM Corporation

# The next steps…



© 2015 IBM Corporation

# The Next Steps…

- Complete the online quiz for this module
  - Log onto SKI, go to “My Learning” page, and select the “In Progress” tab.
  - Find the module and select the quiz
- Provide feedback on the module
  - Log onto SKI, go to “My Learning” page
  - Find the module and select the “Leave Feedback” button to leave your comments

# Questions?
# askdata@ca.ibm.com



© 2015 IBM Corporation