



IBM Software Group

DB2 Express-C 9 overview course

SQL/XML and Xquery

DB2 Information Management Software

ON DEMAND BUSINESS™

DB2: The big picture

DB2 Commands & SQL/XML, Xquery Statements

SQL/XML, Xquery Statements

create bufferpool
create tablespace
create table
alter bufferpool
alter tablespace
alter table
select
insert
update
delete
...

DB2 System Commands

db2set
db2start
db2stop
db2ilist
db2icrt
db2idrop
...

DB2 CLP Commands

db2 [update dbm cfg
catalog db
list node directory
create database
list applications
list tablespaces
...
<sql statement>
xquery < >]

DB2 Tools

Command Line Tools

Command Editor
Command Line Processor
Command Window

Development Tools

DB2 Developer Workbench
Project Deployment Tool

General Administration Tools

Control Center
Journal
License Center
Replication Center
Task Center

Information

Information Center
Check for DB2 Updates

Monitoring Tools

Event Analyzer
Health Center
Indoubt Transaction Manager
Memory Visualizer
Activity Monitor

Setup Tools

Configuration Assistant
Configure DB2 .Net Data Provider
Default DB2 Selection Wizard
First Steps
Register Visual Studio Add-ins

DB2 Environment

Instance 'myinst'

Database Manager
Configuration
File (dbm cfg)

Database MYDB1

Database
Configuration
File (db cfg)

Port



IBM Software Group

Inserts of XML Data

DB2 Information Management Software

A horizontal decorative bar spanning the width of the slide, composed of various colored squares and rectangles in shades of green, yellow, red, and blue.

ON DEMAND BUSINESS™

Table definitions with XML columns

```
create table items (  
  id          int primary key not null,  
  brandname   varchar(30),  
  itemname    varchar(30),  
  sku         int,  
  srp         decimal(7,2),  
  comments    xml  
);
```

```
create table clients(  
  id          int primary key not null,  
  name        varchar(50),  
  status      varchar(10),  
  contact     xml  
);
```



Contents of D:\Raul directory with XML documents

D:\Raul

Name	Size	Type
Client3227.xml	1 KB	XML Document
Client4309.xml	1 KB	XML Document
Client5681.xml	1 KB	XML Document
Client8877.xml	1 KB	XML Document
Client9077.xml	1 KB	XML Document
Client9177.xml	1 KB	XML Document
ClientInfo.xsd	2 KB	XML Schema
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Document
Comment4023.xml	1 KB	XML Document
Comment4272.xml	1 KB	XML Document
items.del	1 KB	DEL File

Insert & Import of XML data

```
INSERT INTO clients VALUES (77, 'John Smith', 'Gold',  
    '<addr>111 Main St., Dallas, TX, 00112</addr>');
```

```
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul"  
    INSERT INTO ARFCHONG.CLIENTS (ID, NAME, STATUS,  
    CONTACT);
```

```
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul"  
    INSERT INTO ARFCHONG.ITEMS (ID, BRANDNAME,  
    ITEMNAME, SKU, SRP, COMMENTS);
```





IBM Software Group

XPath

DB2 Information Management Software

A horizontal decorative bar spanning the width of the slide, composed of various colored squares and rectangles in shades of green, yellow, red, and blue.

ON DEMAND BUSINESS™

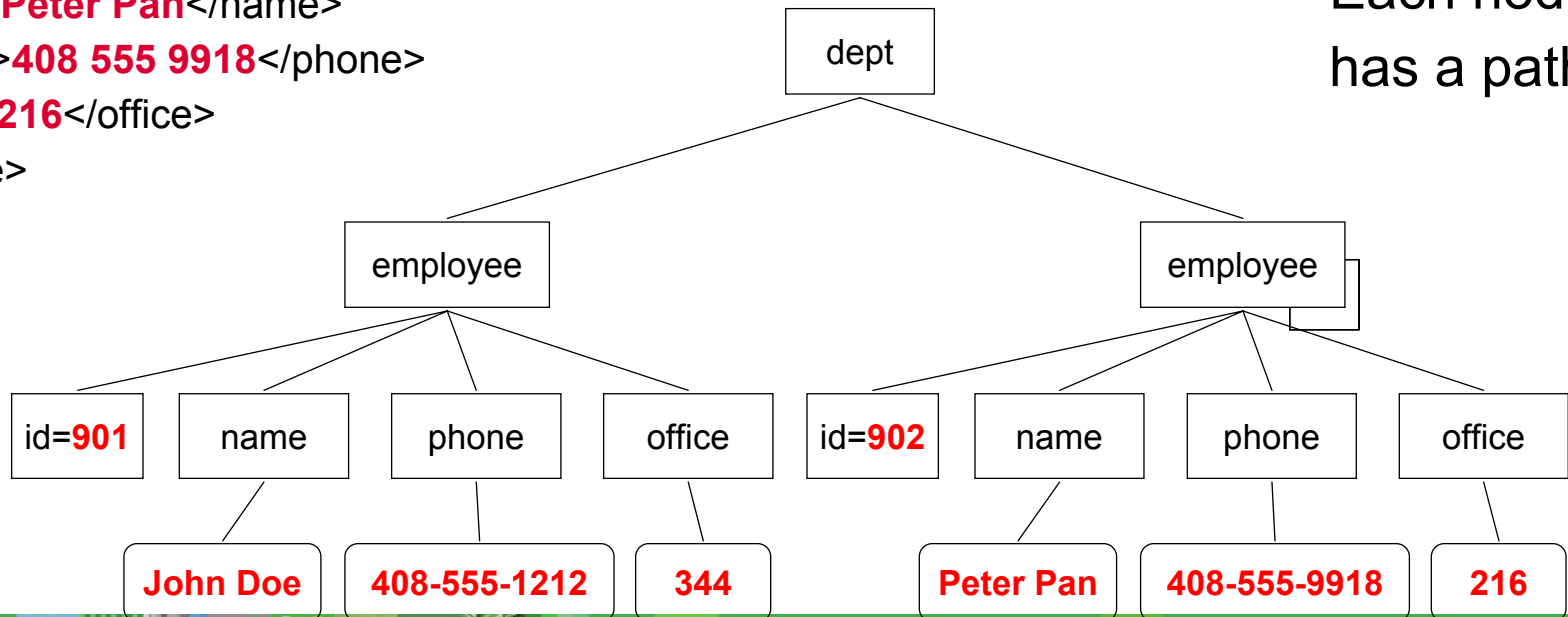
XPath

- ❑ XML Query Language
- ❑ Subset of XQuery & SQL/XML

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

/
/dept
/dept/employee
/dept/employee/@id
/dept/employee/name
/dept/employee/phone
/dept/employee/office
(...)

Each node
has a path





What is XQuery?

XML Schema

[www.w3.org/
XML/Schema](http://www.w3.org/XML/Schema)

XQuery

Expressions

www.w3.org/TR/XQuery & [SQL/XML](http://www.w3.org/TR/SQL/XML)

Functions & Operators

www.w3.org/TR/XQuery-operators/

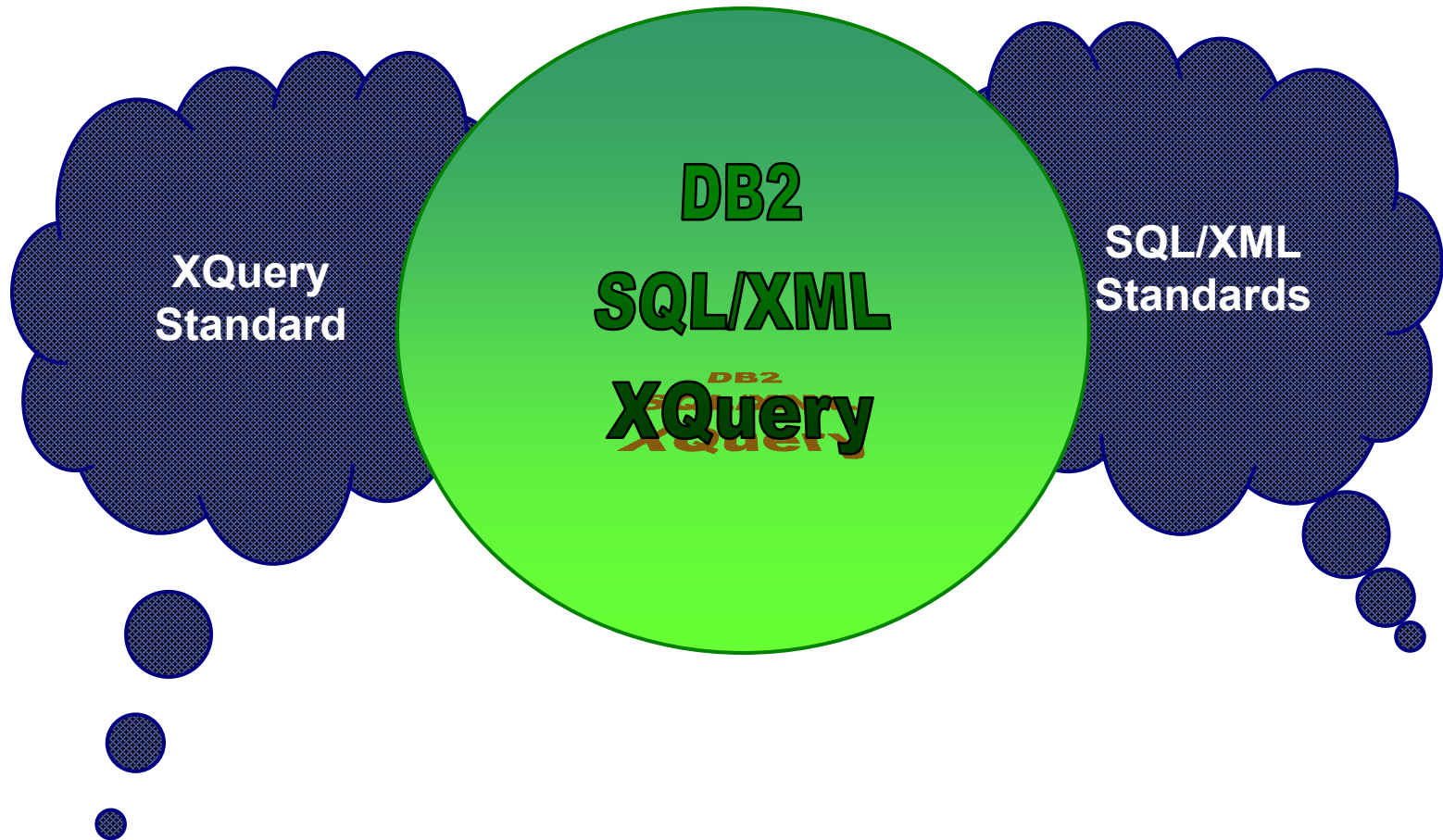
XPath 2.0

[www.w3.org/
TR/xpath20/](http://www.w3.org/TR/xpath20/)

XQuery 1.0 & XPath 2.0 Data Model

www.w3.org/TR/query-datamodel/

Two Worlds





IBM Software Group

Query DB2 XML Data with SQL

DB2 Information Management Software



ON DEMAND BUSINESS™

New SQL/XML Functions in SQL 2006

- XMLPARSE
 - ▶ parses character/BLOB data, produces XML value
- XMLSERIALIZE
 - ▶ converts an XML value into character/BLOB data
- XMLVALIDATE
 - ▶ validates XML value against an XML schema and type-annotates the XML value
- **XMLEXISTS**
 - ▶ determines if an XQuery returns a results (i.e. a sequence of one or more items)
- **XMLQUERY**
 - ▶ executes an XQuery and returns the result sequence
- **XMLTABLE**
 - ▶ executes an XQuery, returns the result sequence as a relational table (if possible)
- XMLCAST
 - ▶ cast to or from an XML type



xmlexists function

- Restricts results based on an XML element value

```
select name from clients  
where xmlexists('$c/Client/Address[zip="95116"]'  
passing clients.contact as "c")
```

Demo 1



xmlquery function

- Retrieve one or more element values from our XML document

```
select xmlquery('$c/Client/email'  
passing contact as "c")  
from clients  
where status = 'Gold'
```



xmltable function: From XML to Relational

```
select t.comment#,i.itemname,t.customerID,Message  
from items i,  
xmltable('$c/Comments/Comment' passing i.comments as "c"  
         columns Comment# integer path 'CommentID',  
                  CustomerID integer path 'CustomerID',  
                  Message varchar(100) path 'Message') as t
```



xmlelement function: From Relational to XML

```
select
  xmlelement (name "item",itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand",brandname),
  xmlelement (name "sku",sku)
from items
where srp < 100
```

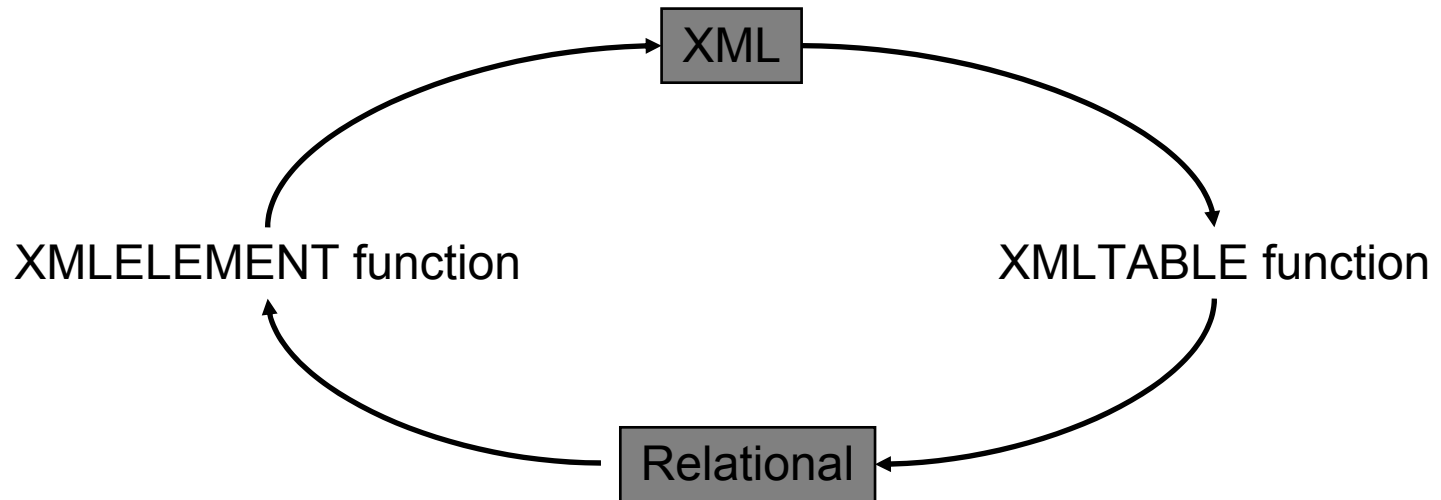


Sample Output

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```



Review: XMLTABLE vs XMLELEMENT





IBM Software Group

Query DB2 XML Data with XQuery

DB2 Information Management Software



ON DEMAND BUSINESS™

XQuery: The FLWOR Expression

- **FOR**: iterates through a sequence, bind variable to items
- **LET**: binds a variable to a sequence
- **WHERE**: eliminates items of the iteration
- **ORDER**: reorders items of the iteration
- **RETURN**: constructs query results



```
create table dept(deptID char(8),deptdoc xml);
```

xquery

```
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
  <EmpList>
    {$d/@bldg, $emp}
  </EmpList>
```

Input

```
<dept bldg=101>
  <employee id=901>
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id=902>
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

FLWOR expression to retrieve client fax data

```
xquery  
FOR $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax  
RETURN $y
```

```
<fax>4081112222</fax>  
<fax>5559998888</fax>
```



Querying DB2 xml data and returning results as HTML

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



Sample HTML

```
<ul>
<li>
<address>
    <street>9407 Los Gatos Blvd.</street>
    <city>Los Gatos</city>
    <state>ca</state>
    <zip>95302</zip>
</address>
</li>
<Address>
<street>4209 El Camino Real</street>
    <city>Mountain View</city>
    <state>CA</state>
    <zip>95302</zip>
</address>
</li>
...
</ul>
```



XQueries with embedded SQL

- db2-fn:sqlquery
 - A function which executes an SQL query and returns only the selected data
 - The SQL Query passed to db2-fn:sqlquery must return XML data
 - This XML data can then be further processed by XQuery



Embedded SQL within XQuery

- db2-fn:sqlquery

- ▶ A function which executes an SQL query and returns only the selected data
- ▶ The SQL Query passed to db2-fn:sqlquery must return XML data
- ▶ This XML data can then be further processed by XQuery

```
xquery
for $y in
db2-fn:sqlquery('select comments from items where srp > 100')/Comments/Comment
where $y/ResponseRequested='Yes'
return (
  <action>
    {$y/ProductID
     $y/CustomerID
     $y/Message}
  </action>
)
```

Demo 5



XML Indexing Examples

create table customer(info XML);

create unique index **idx1** on **customer**(info)

generate key using

xmlpattern '**/customerinfo/@Cid**'

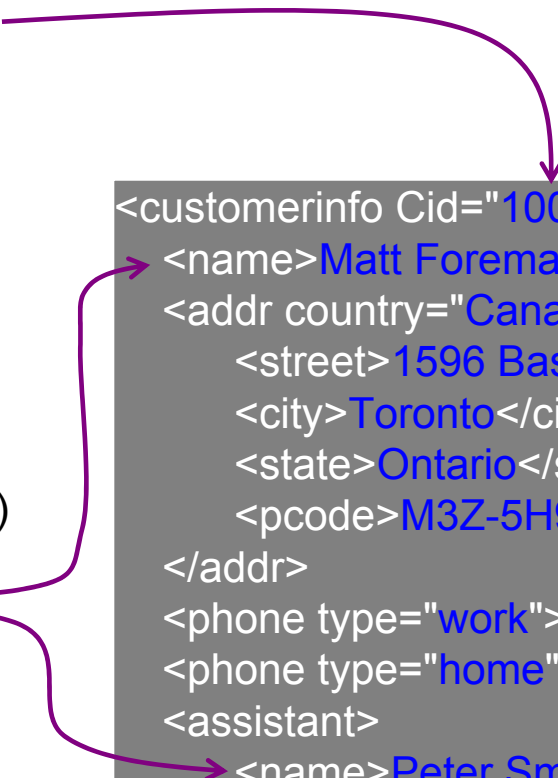
as sql double;

create index **idx2** on **customer**(info)

generate key using

xmlpattern '**//name**'

as sql varchar(40);



```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

XML Indexing Examples



Don't index everything!
Too expensive for
insert, update, delete !



```
create table customer( info XML);
```

create index **idx3** on **customer**(info)
generate key using
xpath **//text()**
as sql varchar(40);

```
<customerinfo Cid="1004">  
  <name>Matt Foreman</name>  
  <addr country="Canada">  
    <street>1596 Baseline</street>  
    <city>Toronto</city>  
    <state>Ontario</state>  
    <pcode>M3Z-5H9</pcode>  
  </addr>  
  <phone type="work">905-555-4789</phone>  
  <phone type="home">416-555-3376</phone>  
  <assistant>  
    <name>Peter Smith</name>  
    <phone type="home">416-555-3426</phone>  
  </assistant>  
</customerinfo>
```