

Advanced Query Languages



- W3C's Activity on XQuery

Shirish Kucheria

- **Introduction**
 - Need for advanced Query Languages
 - XML Query Working Group
 - Goals
 - XPath
- **XQuery**
 - Introduction
 - How it Works?
 - Syntax
 - Adding Element and Attributes to Result
 - XQuery Functions
 - Type Specification
- **Implementation in Digital Infrastructure**
 - CLASSIX, XIRQL, DataDirect XQuery
- **Conclusion**
- **References**

Need For Advanced Query Language

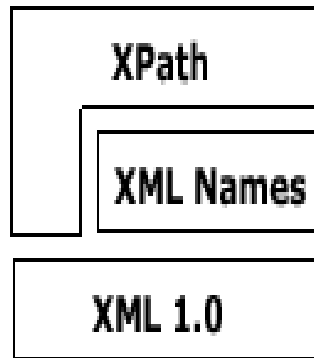


- XML – Fast becoming method of choice for representing all kind of documents like product catalog, digital libraries, scientific data repositories, etc
- XML – Catalyst in constructing Semantic Web
- There is a need for easy to use yet powerful and efficient search language
- Mechanism for interaction between Web World and Database world



- XML Query Working Group started development on XQuery after QL'98 event
- Works on devising specifications for development of XML Query Language
- Also develops number of specifications which are shared with XSL Working Group
- These include XML selection and an expression language XPath

- Ambitious task of XML Query (XQuery) Working Group is to develop first, world standard for querying Web documents
- Proposed Query language should provide flexible query facilities to extract data from real and virtual document available locally or on World Wide Web
- Interaction between Web World and Database World



- XPath is a language for finding information in an XML document.
- XPath is used to navigate through elements and attributes in an XML document.
- XPath is a major element in the W3C's XSLT standard.
- XQuery and XPointer are both built on XPath expressions.

What is XQuery?

- XQuery is the language for querying XML data
- XQuery is to XML what SQL is to databases
- XQuery is built on XPath expressions
- XQuery is defined by W3C
- XQuery is supported by all the major database engines like IBM, Oracle, Microsoft, etc

- XQuery can answer to the following kind of question
 - “Select all CD records with a price less than \$10 from the CD collection stored in XML document called cd_catalog.xml”
- XQuery 1.0 and XPath 2.0 share the same data model and support the same functions and operators.
- XQuery can be used to
 - Extract information to use in Web Service
 - Transform XML data to XHTML
 - Search Web documents for relative information.
- XQuery has been widely implemented and recently has achieved **candidate recommendation**.

XML Example – books.xml



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year> <price>30.00</price> </book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year> <price>29.99</price> </book>
<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author> <author>Per Bothner</author> <author>Kurt
Cagle</author> <author>James Linn</author> <author>Vaidyanathan Nagarajan</author>
  <year>2003</year> <price>49.99</price> </book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year> <price>39.95</price> </book>
</bookstore>
```

Osnabrück, 2005-11-24

How it works?



- XQuery uses functions to extract data from XML documents
 - Doc() function is used to open the XML file
 - Ex doc(“books.xml”)
- Path Expressions
 - XQuery uses Path expressions to navigate through elements in XML document.
 - Ex doc(“books.xml”)/bookstore/book/title
 - XQuery will extract the following

```
<title lang="en">Everyday Italian</title>  
<title lang="en">Harry Potter</title>  
<title lang="en">XQuery Kick Start</title>  
<title lang="en">Learning XML</title>
```

How it works? ... Contd



- Predicates – limit the extracted data from XML documents
- Ex doc(“books.xml”)/bookstore/book[price<30]
- XQuery will extract the following

```
<book category="CHILDREN">  
<title lang="en">Harry Potter</title>  
<author>J K. Rowling</author>  
<year>2005</year>  
<price>29.99</price>  
</book>
```

- **Basic syntax rules**
 - XQuery is case sensitive
 - XQuery elements, variables must be valid XML names
 - An XQuery string value can be in single or double quotes
 - An XQuery variable is defined with a \$ followed by a name ex \$bookstore
 - XQuery comments are delimited by (: and :) ex (: XQuery Comment:)

Adding Elements & Attributes



- XQuery can be embedded in HTML

```
<html> <body>
<ul>
  { for $x in doc("books.xml")/bookstore/book
    order by $x/title
    return <li> {data($x/title)}. Category: {data($x/@category)} </li> }
</ul>
</body> </html>
```

```
<html> <body>
<ul>
  <li>Everyday Italian. Category: COOKING</li>
  <li>Harry Potter. Category: CHILDREN</li>
  <li>Learning XML. Category: WEB</li>
  <li>XQuery Kick Start. Category: WEB</li>
</ul>
</body> </html>
```

- XQuery 1.0, XPath 2.0 and XSLT 2.0 share the same function library
- XQuery includes over 100 built-in functions
- Available functions – String values, numeric values, date and time comparison, node manipulation, sequence manipulation, Boolean values, etc
- Defining new functions is permitted
- **Example 1 – In an Element**
 - `<name>{uppercase($booktitle)}</name>`
- **Example 2 – In the predicate of Path Expression**
 - `doc("books.xml")/bookstore/book[substring(title,1,5='Harry')]`

- XQuery is strongly typed programming language.
- Like Java, C# and other programming language it is mix of static typing and dynamic typing.
- Types in XQuery are different from classes in OOPS.
- Types matches to XQuery's data model

- For amenable and effective information search on XML significant process is needed on following issues
 - Effective search language with ontology backed information retrieval capability
 - Extracting more semantics from existing documents
 - Classifying existing documents into hierarchical structure
- CLASSIX – Joint project of Universities of Dortmund and Saarland in Germany, addressing above 3 issues.
- For more information
 - http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Fuhr_Weikum:02.pdf

- Document Centric View
 - Focuses on structured documents
 - Logic markup of text at both macro and micro level
- Data Centric View
 - Uses XML for exchanging formatted data
 - Important for e-business application (exchanging orders, bills)
- Most approaches for XML query languages including XQuery supports Data centric view
- XIRQL – Xml IR Query Language

- XIRQL extends XPath part of XQuery by following features
 - Weighing and Ranking
 - Relevance oriented search
 - Data types with vague predicates
 - Structural relativism
- Implementation
 - HYREX retrieval engine
 - Accepts XIRQL queries, translates and process to give efficient retrieval
 - Current version supports gigabyte of data

- DataDirect XQuery can query XML and relational databases and produces XML needed for your application
- Implementation of W3C XQuery standard using XQuery API for Java (XQJ)
- Supports DB2, Microsoft SQL Server 2000, Oracle 9i
- Available, to be installed, in beta version

- XQuery is a powerful and convenient tool for analyzing or generating XML
- XQuery is protocol independent, thus can be evaluated on any system with predictable results
- There are no standard implementations but XQuery site lists known implementation
- XQuery is compatible with other W3C standards
 - XML 1.1, Namespaces, XML Schema, XPath
- Nov 3rd – XQuery gets Candidate Recommendation

- “W3C XML Query (XQuery)”, <http://www.w3.org/XML/Query/> , last modified 2005/11/11
- “XQuery 1.0 – XML Query Language”, W3C Working Draft, 04 April 2005, <http://www.w3.org/TR/xquery/>
- “XQuery Tutorial”, W3schools, <http://www.w3schools.com/xquery/default.asp>
- Bothner Per, “What is XQuery”, <http://www.gnu.org/software/qexo/XQuery-Intro.html>, Last modified Jun 2003
- Fuhr, N Weikum, G “Classification and Intelligent Search on Information in XML”, http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Fuhr_Weikum:02.pdf

Thank you!

