WIKIPEDIA

XQuery

XQuery (**XML Query**) is a query and functional programming language that queries and transforms collections of structured and unstructured data, usually in the form of XML, text and with vendor-specific extensions for other data formats (JSON, binary, etc.). The language is developed by the XML Query working group of the W3C. The work is closely coordinated with the development of XSLT by the XSL Working Group; the two groups share responsibility for XPath, which is a subset of XQuery.

XQuery 1.0 became a W3C Recommendation on January 23, 2007.[4]

XQuery 3.0 became a W3C Recommendation on April 8, 2014.[5]

XQuery 3.1 became a W3C Recommendation on March 21, 2017.[6]

> "The mission of the XML Query project is to provide flexible query facilities to extract data from real and virtual documents on the World Wide Web, therefore finally providing the needed interaction between the Web world and the database world. Ultimately, collections of XML files will be accessed like databases."[7]

XQuery

Paradigm	declarative, functional, modular
Designed by	W3C
First appeared	2007
Stable release	3.1 (http://www.w3.org/TR/xquery-31/) / March 21, 2017[1]
Typing discipline	dynamic or static,[2][3] strong
OS	Cross-platform
Filename extensions	.xq, .xql, .xqm, .xqy, .xquery
Website	www.w3.org/XML/Query/ (http://www.w3.org/XML/Query/)
Major implementations	
Many (http://www.w3.org/XML/Query#implementations)	
Influenced by	
XPath, SQL, XSLT	
XQuery at Wikibooks	

Contents

Features

XQuery is a functional, side effect-free, expression-oriented programming language with a simple type system, summed up by Kilpeläinen:[8]

> All XQuery expressions operate on sequences, and evaluate to sequences. *Sequences* are ordered lists of items. *Items* can be either *nodes*, which represent components of XML documents, or *atomic values*, which are instances of XML Schema base types like `xs:integer` or `xs:string`. Sequences can also be empty, or consist of a single item only. No distinction is made between a single item and a singleton sequence. (...) XQuery/XPath sequences differ from lists in languages like Lisp and Prolog by excluding nested sequences. Designers of XQuery may have considered nested sequences unnecessary for the manipulation of document contents. Nesting, or hierarchy of document structures is instead represented by nodes and their child-parent relationships

XQuery provides the means to extract and manipulate data from XML documents or any data source that can be viewed as XML, such as relational databases[9] or office documents.

XQuery contains a superset of XPath expression syntax to address specific parts of an XML document. It supplements this with a SQL-like "FLWOR expression" for performing joins. A FLWOR expression is constructed from the five clauses after which it is named: FOR, LET, WHERE, ORDER BY, RETURN.

The language also provides syntax allowing new XML documents to be constructed. Where the element and attribute names are known in advance, an XML-like syntax can be used; in other cases, expressions referred to as dynamic node constructors are available. All these constructs are defined as expressions within the language, and can be arbitrarily nested.

The language is based on the XQuery and XPath Data Model (XDM) which uses a tree-structured model of the information content of an XML document, containing seven kinds of nodes: document nodes, elements, attributes, text nodes, comments, processing instructions, and namespaces.

XDM also models all values as sequences (a singleton value is considered to be a sequence of length one). The items in a sequence can either be XML nodes or atomic values. Atomic values may be integers, strings, booleans, and so on: the full list of types is based on the primitive types defined in XML Schema.

Features for updating XML documents or databases, and full text search capability, are not part of the core language, but are defined in add-on extension standards: XQuery Update Facility 1.0 supports update feature and XQuery and XPath Full Text 1.0 supports full text search in XML documents.

XQuery 3.0 adds support for full functional programming, in that functions are values that can be manipulated (stored in variables, passed to higher-order functions, and dynamically called).

Examples

The sample XQuery code below lists the unique speakers in each act of Shakespeare's play Hamlet, encoded in hamlet.xml (http://www.ibiblio.org/xml/examples/shakespeare/hamlet.xml)

```
<html><body>
{
  for $act in doc("hamlet.xml")//ACT
  let $speakers := distinct-values($act//SPEAKER)
  return
    <div>
      <h1>{ string($act/TITLE) }</h1>
      <ul>
      {
        for $speaker in $speakers
        return <li>{ $speaker }</li>
      }
      </ul>
    </div>
}
</body></html>
```

All XQuery constructs for performing computations are expressions. There are no statements, even though some of the keywords appear to suggest statement-like behaviors. To execute a function, the expression within the body is evaluated and its value is returned. Thus to write a function to double an input value, one simply writes:

```
declare function local:doubler($x) { $x * 2 }
```

To write a full query saying 'Hello World', one writes the expression:

```
"Hello World"
```

This style is common in functional programming languages.

Applications

Below are a few examples of how XQuery can be used:

1. Extracting information from a database for use in a web service.
2. Generating summary reports on data stored in an XML database.
3. Searching textual documents on the Web for relevant information and compiling the results.
4. Selecting and transforming XML data to XHTML to be published on the Web.
5. Pulling data from databases to be used for the application integration.
6. Splitting up an XML document that represents multiple transactions into multiple XML documents.

XQuery and XSLT compared

Scope

Although XQuery was initially conceived as a query language for large collections of XML documents, it is also capable of transforming individual documents. As such, its capabilities overlap with XSLT, which was designed expressly to allow input XML documents to be transformed into HTML or other formats.

The XSLT 2.0 and XQuery standards were developed by separate working groups within W3C, working together to ensure a common approach where appropriate. They share the same data model (XDM), type system, and function library, and both include XPath 2.0 as a sublanguage.

Origin

The two languages, however, are rooted in different traditions and serve the needs of different communities. XSLT was primarily conceived as a stylesheet language whose primary goal was to render XML for the human reader on screen, on the web (as web template language), or on paper. XQuery was primarily conceived as a database query language in the tradition of SQL.

Because the two languages originate in different communities, XSLT is stronger in its handling of narrative documents with more flexible structure, while XQuery is stronger in its data handling (for example, when performing relational joins).

Versions

XSLT 1.0 appeared as a Recommendation in 1999, whereas XQuery 1.0 only became a Recommendation in early 2007; as a result, XSLT is still much more widely used. Both languages have similar expressive power, though XSLT 2.0 has many features that are missing from XQuery 1.0, such as grouping, number and date formatting, and greater control over XML namespaces.[10][11][12] Many of these features were planned for XQuery 3.0.[13]

Any comparison must take into account the fact that XSLT 1.0 and XSLT 2.0 are very different languages. XSLT 2.0, in particular, has been heavily influenced by XQuery in its move to strong typing and schema-awareness.

Strengths and weaknesses

Usability studies have shown that XQuery is easier to learn than XSLT, especially for users with previous experience of database languages such as SQL.[14] This can be attributed to the fact that XQuery is a smaller language with fewer concepts to learn, and to the fact that programs are more concise. It is also true that XQuery is more orthogonal, in that any expression can be used in any syntactic context. By contrast, XSLT is a two-language system in which XPath expressions can be nested in XSLT instructions but not vice versa.

XSLT is currently stronger than XQuery for applications that involve making small changes to a document (for example, deleting all the NOTE elements). Such applications are generally handled in XSLT by use of a coding pattern that involves an identity template that copies all nodes unchanged, modified by specific templates that modify selected nodes. XQuery has no equivalent to this coding pattern, though in future versions it will be possible to tackle such problems using the update facilities in the language that are under development.[15]

XQuery 1.0 lacked any kind of mechanism for dynamic binding or polymorphism; this has been remedied with the introduction of functions as first-class values in XQuery 3.0. The absence of this capability starts to become noticeable when writing large applications, or when writing code that is designed to be reusable in different environments. XSLT offers two complementary mechanisms in this area: the dynamic matching of template rules, and the ability to override rules using `xsl:import`, that make it possible to write applications with multiple customization layers.

The absence of these facilities from XQuery 1.0 was a deliberate design decision: it has the consequence that XQuery is very amenable to static analysis, which is essential to achieve the level of optimization needed in database query languages. This also makes it easier to detect errors in XQuery code at compile time.

The fact that XSLT 2.0 uses XML syntax makes it rather verbose in comparison to XQuery 1.0. However, many large applications take advantage of this capability by using XSLT to read, write, or modify stylesheets dynamically as part of a processing pipeline. The use of XML syntax also enables the use of XML-based tools for managing XSLT code. By contrast, XQuery syntax is more suitable for embedding in traditional programming languages such as Java (see XQuery API for Java) or C#. If necessary, XQuery code can also be expressed in an XML syntax called XQueryX. The XQueryX representation of XQuery code is rather verbose and not convenient for humans, but can easily be processed with XML tools, for example transformed with XSLT stylesheets.[16][17]

Extensions and future work

W3C extensions

Two major extensions to the XQuery were developed by the W3C:

- XQuery 1.0 and XPath 2.0 Full-Text[18]
- XQuery Update Facility

Both reached Recommendation status as extensions to XQuery 1.0, but work on taking them forward to work with XQuery 3.0 was abandoned for lack of resources.

Work on XQuery 3.0 was published as a Recommendation on 8 April 2014,[19] and XQuery 3.1 is a Recommendation as at February 2017.

A scripting (procedural) extension for XQuery was designed, but never completed.[20][21]

The EXPath Community Group[22] develops extensions to XQuery and other related standards (XPath, XSLT, XProc, and XForms). The following extensions are currently available:

- Packaging System[23]
- File Module[24]
- Binary Module[25]
- Web Applications[26]

Third-party extensions

JSONiq is an extension of XQuery that adds support to extract and transform data from JSON documents. JSONiq is a superset of XQuery 3.0. It is published under the Creative Commons Attribution-ShareAlike 3.0 license.

The EXQuery[27] project develops standards around creating portable XQuery applications. The following standards are currently available:

- RESTXQ[28]

Further reading

- Querying XML: XQuery, XPath, and SQL/XML in context. Jim Melton and Stephen Buxton. Morgan Kaufmann, 2006. ISBN 1-55860-711-0.
- Walmsley, Priscilla (2007). *XQuery, 1st Edition*. O'Reilly Media. ISBN 978-0-596-00634-1.
- Walmsley, Priscilla (2015). *XQuery, 2nd Edition*. O'Reilly Media. ISBN 978-1-4919-1510-3.
- XQuery: The XML Query Language. Michael Brundage. Addison-Wesley Professional, 2004. ISBN 0-321-16581-0.
- XQuery from the Experts: A Guide to the W3C XML Query Language. Howard Katz (ed). Addison-Wesley, 2004. ISBN 0-321-18060-7.
- An Introduction to the XQuery FLWOR (http://www.stylusstudio.com/xquery_flwor.html) Expression. Dr. Michael Kay (W3C XQuery Committee), 2005.

Implementations

Overview of popular XQuery implementations

Name	License	Language	XQuery 3.1	XQuery 3.0	XQuery 1.0	XQuery Update 1.0	XQuery Full Text 1.0
BaseX	BSD license	Java	Yes	Yes	Yes	Yes	Yes
eXist	LGPL	Java	Partial	Partial	Yes	No	No
MarkLogic	Proprietary	C++	No	Partial	Yes	No	No
Saxon HE	Mozilla Public License	Java	Partial	Partial	Yes	Yes	No
Saxon EE	Proprietary	Java	Yes	Yes	Yes	Yes	No
Zorba	Apache License	C++	No	Yes	Yes	Yes	Yes

Other implementations include:

- Oracle XQuery for Hadoop: "Using Oracle XQuery for Hadoop" (https://web.archive.org/web/20160304211700/http://docs.oracle.com/cd/E63064_01/doc.42/e63063/oxh.htm). Archived from the original (http://docs.oracle.com/cd/E63064_01/doc.42/e63063/oxh.htm) on 2016-03-04.
- Oracle (http://docs.oracle.com/database/121/ADXDK/adx_j_xqj.htm): XQuery Processor for Java
- RaptorXML Server (http://www.altova.com/raptorxml.html): Altova's XSLT and XQuery engine
- Sirix (https://github.com/sirixdb/sirix): XQuery with versioning extensions
- XQilla (http://xqilla.sourceforge.net/HomePage): XQilla XQuery implementation
- SPARQL2XQuery (http://www.dblab.ntua.gr/~bikakis/SPARQL2XQuery.html): SPARQL to XQuery translator
- Galax (http://galax.sourceforge.net/about.html) – an open source XQuery implementation.
- Db2 (https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.xml.doc/doc/xqrbasics.html) – XQuery implementation in IBM Db2.
- Xidel (http://www.videlibri.de/xidel.html) – open-source XQuery 3.0 implementation

Surveys

- XQuery Test Suite Results (https://dev.w3.org/2006/xquery-test-suite/PublicPagesStagingArea/XQTSReport_XQTS_1_0_2.html): Compliance with XQuery 1.0 features
- EXPath (http://www.w3.org/community/expath/wiki/Engines): XPath/XQuery engines, including a feature matrix
- W3C (http://www.w3.org/XML/Query#implementations): XQuery implementations

References

1. "XQuery 3.1 Recommendation" (http://www.w3.org/TR/xquery-31/). 2017-03-21.
2. "XQuery 3.1: An XML Query Language" (https://www.w3.org/TR/xquery-31/#id-static-typing-feature). 2017-03-21.
3. "XQuery and Static Typing" (https://technet.microsoft.com/en-us/library/ms190938.aspx).
4. "XML and Semantic Web W3C Standards Timeline" (http://www.dblab.ntua.gr/~bikakis/XML%20and%20Semantic%20Web%20W3C%20Standards%20Timeline-History.pdf) (PDF). 2012-02-04.
5. "XQuery 3.0 Recommendation" (http://www.w3.org/TR/xquery-30/). 2014-04-08.
6. "XQuery 3.1 Recommendation" (http://www.w3.org/TR/xquery-31/). 2017-03-21.
7. W3C (2003-10-25). "cited by J.Robie" (http://lists.w3.org/Archives/Public/www-tag/2003Oct/0115.html).
8. Kilpeläinen, Pekka (2012). "Using XQuery for problem solving" (http://www.cs.uef.fi/~kilpelai/RDK11/refs/xqueryProblems.pdf) (PDF). *Software: Practice and Experience*. **42** (12): 1433–1465. doi:10.1002/spe.1140 (https://doi.org/10.1002%2Fspe.1140). S2CID 15561027 (https://api.semanticscholar.org/CorpusID:15561027).
9. "Data retrieval with XQuery" (http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.xml.doc/doc/c0023907.html?lang=en). Retrieved on 18 January 2016.
10. Kay, Michael (May 2005). "Comparing XSLT and XQuery" (https://www.mscs.mu.edu/~praveen/Teaching/fa05/AdvDb/PaperTeams/XSLT2XQTeam/Comparing%20XSLT%20and%20XQuery.htm).
11. Eisenberg, J. David (2005-03-09). "Comparing XSLT and XQuery" (http://www.xml.com/pub/a/2005/03/09/xquery-v-xslt.html).
12. Smith, Michael (2001-02-23). "XQuery, XSLT "overlap" debated" (http://www.xmlhack.com/read.php?item=1080).
13. "XQuery 3.0 requirements" (http://www.w3.org/TR/xquery-30-requirements/).
14. Usability of XML Query Languages. Joris Graaumans. SIKS Dissertation Series No 2005-16, ISBN 90-393-4065-X
15. "XQuery Update Facility" (http://www.w3.org/TR/xqupdate/).
16. "XML Syntax for XQuery (XQueryX)" (http://www.w3.org/TR/xqueryx/).
17. Michael Kay. "Saxon diaries: How not to fold constants" (http://saxonica.blogharbor.com/blog/_archives/2007/1/20/2665644.html).
18. XQuery and XPath Full Text 1.0 (http://www.w3.org/TR/xquery-full-text/)
19. XML Query (XQuery) 3.0 (http://www.w3.org/TR/xquery-30/)
20. XQuery Scripting Extension 1.0 Requirements (http://www.w3.org/TR/xquery-sx-10-requirements/)
21. XQuery 1.0 Scripting Extension (https://www.w3.org/TR/2017/PR-xquery-31-20170117/)
22. EXPath Community Group (http://www.w3.org/community/expath/)
23. Packaging System (http://expath.org/spec/pkg/20120509)

24. File Module (http://expath.org/spec/file/20120614)
25. Binary Module (http://expath.org/spec/binary/20130312)
26. Web Applications (http://expath.org/spec/webapp/20130401)
27. "Standard for portable XQuery applications" (http://www.exquery.org/). Retrieved 12 December 2013.
28. "RESTXQ 1.0: RESTful Annotations for XQuery" (https://exquery.github.io/exquery/exquery-restxq-specification/restxq-1.0-specification.html).

- *Portions borrowed with permission from the books "XML Hacks" (O'Reilly Media) and "XQuery" (O'Reilly Media).*
- *Previous version based on an article at the French language Wikipedia*

External links

- W3C XML Query (XQuery) (http://www.w3.org/XML/Query)
- XQuery tutorial (https://www.w3schools.com/xml/xquery_intro.asp)
- XQuery API for Java (XQJ) Java Specification Request
- hamlet.xml (http://www.ibiblio.org/xml/examples/shakespeare/hamlet.xml) Hamlet in XML Format
- XQuery (http://www.cafeconleche.org/slides/xmlsig/xquery/index.html) (presentation - as HTML slides)
- List of open-source XQuery implementations (http://www.martinbroadhurst.com/open-source-xquery-implementations.html)

Retrieved from "https://en.wikipedia.org/w/index.php?title=XQuery&oldid=1024779822"

This page was last edited on 24 May 2021, at 02:02 (UTC).