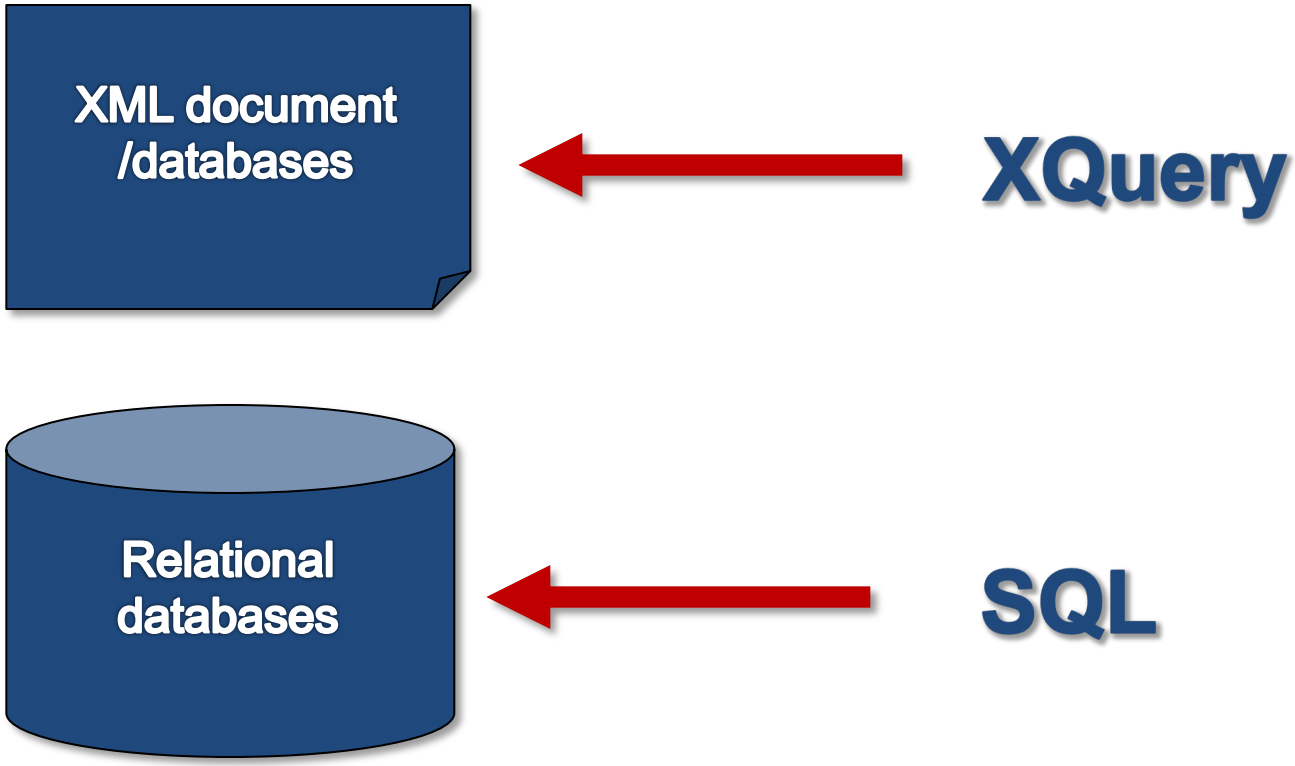
XQuery

XQuery



What is XQuery

- Designed to meet the requirements identified by the W3C XML Query Working Group
  - "XML Query 1.0 Requirements"
  - "XML Query Use Cases".
- Designed to be a small, easily implementable language.
- Flexible enough to query a broad spectrum of XML sources (both databases and documents).
- Defines a human-readable syntax for that language.
- A query in XQuery is an expression that:
  - Reads a number of XML documents or fragments
  - Returns a sequence of well-formed XML fragments

The Principal Forms of XQuery

- Path
  - Locates nodes within a tree, and returns a sequence of distinct nodes in document order.
- Sequence
  - An ordered collection of zero or more items, where an item may be an atomic value or a node.
- Arithmetic
  - Arithmetic operators for addition, subtraction, multiplication, division, and modulus.
- Comparison
  - Four kinds of comparisons: value, general, node, and order comparisons.
- Logical
  - A logical expression is either an AND-expression or an OR-expression.
  - The value of a logical expression is always a Boolean value.

The Principal Forms of XQuery

- Constructor
  - Constructors can create XML structures within a query.
  - There are constructors for elements, attributes, CDATA sections, processing instructions, and comments.
- FLWR
  - Expression for iteration and for binding variables to intermediate results.
  - Useful for computing joins between two or more documents and for restructuring data.
  - Pronounced "flower", stands for the keywords FOR, LET, WHERE, and RETURN, the four clauses found in a FLWR expression.

The Principal Forms of XQuery

- Sorting expressions
  - Provides a way to control the order of items in a sequence.
- Conditional expressions
  - Based on the keywords IF, THEN, and ELSE.
- Quantified expressions
  - support existential and universal quantification.
  - The value of a quantified expression is always true or false.

Example XML Document

```
<bib>
  <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
 </book>
 <book year=”1995">
        <title>XML</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>45</price>
 </book>
</bib>
```

XQuery Example 1

Find all books with a price of $39.95

XQuery:

```
document("bib.xml")/bib/book[price = 39.95]
```

Result:

```
<book year="2000">
  <title>Data on the Web</title>
  <author><last>Abiteboul</last><first>Serge</first></author>
  <author><last>Buneman</last><first>Peter</first></author>
  <author><last>Suciu</last><first>Dan</first></author>
  <publisher>Morgan Kaufmann Publishers</publisher>
  <price> 39.95</price>

</book>
```

XQuery Example 2

Find the title of all books published before 2017

XQuery:

```
document("bib.xml")/bib/book[@year < 1995]/title
```

Result:

```
<title>XML</title>
<title>Data on the Web</title>
```

XQuery Example 3 (For Loop)

List books published by Addison-Wesley after 1991, including their year and title.

XQuery:

```
<bib>
 {
  for $b in document("bib.xml")/bib/book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
  return
    <book year="{ $b/@year }">
     { $b/title }
    </book>
 }
</bib>
```

XQuery Example 3 (For Loop)

- List books published by Addison-Wesley after 1991, including their year and title…

Result:

```
<bib>
 <book year="1995">
     <title>XML</title>
 </book>
 <book year=”2000">
     <title>Data on the Web</title>
 </book>
</bib>
```

XQuery Example 4 (Join)

For each book found at both bn.com and amazon.com, list the title of the book and its price from each source.

XQuery:

```
<books-with-prices>
  {
    for $b in document("bib.xml")//book,
        $a in document("reviews.xml")//entry
    where $b/title = $a/title
    return
        <book-with-prices>
            { $b/title }
            <price-amazon>{ $a/price }</price-amazon>
            <price-bn>{ $b/price }</price-bn>
        </book-with-prices>
  }
</books-with-prices>
```

XQuery Example 4 (Join)

For each book found at both bn.com and amazon.com, list the title of the book and its price from each source.

Result:

```
<books-with-prices>
  <book-with-prices>
            <title>XML</title>
            <price-amazon><price>65.95</price></price-amazon>
            <price-bn><price> 65.95</price></price-bn>
  </book-with-prices><book-with-prices>
            <title>Data on the Web</title>
            <price-amazon><price>34.95</price></price-amazon>
            <price-bn><price> 39.95</price></price-bn>
        </book-with-prices>
</books-with-prices>
```

XQuery Support on RDBMSs

- Oracle XQuery Engine
  - http://www.oracle.com/technology/tech/xml/xquery/index.html
- Introduction to XQuery in SQL Server 2005
  - http://msdn.microsoft.com/en-us/library/ms345122(SQL.90).aspx
- Query DB2 XML data with XQuery
  - http://www.ibm.com/developerworks/data/library/techarticle/dm-0604saracco/
- DataDirect: Data Integration Suite – MySQL Database Support
  - http://www.datadirect.com/products/data-integration/datasources/databases/mysql/index.ssp