Using XML data with XQuery

Class Goals

- Show what XQuery is and what it does
- Get class to write a simple XQuery script
- Give class a starting point for later exploration

What is XQuery? How is it used?

- W3C standard
- Designed for users without formal programming background
- Designed to extract, transform, and manipulate XML data
- mySQL for XML data

XQuery Processors

- Saxon
- Zorba (for PHP and Python)
- eXist XML Database (REST interface)
- Proprietary XML databases (MarkLogic)
- BaseX
  - Java
  - GUI

Prolog and Body

```
xquery version “3.0”;
declare variable $input := doc(“myfile.xml”);

for $data in $input/element/info
return $data
```

No XML, Odd Punctuation

```
xquery version “3.0”;

for $data in doc("myfile.xml")/folder/info
let $x := lower-case($data)
where $x >= 733
order by $data@type
return $x
```

Variables

- Can be any text you like

```
$data
$xml
$info
$my_info
```

not `$my info`

FLOWR expressions

For

Let

Order by

Where

Return

FLOWR expressions

```
xquery version “3.0”;

for $data in doc("myfile.xml")/folder/info
let $x := lower-case($data)
where $x >= 733
order by $data@type
return $x
```

XPath in XQuery

```
xquery version “3.0”;

for $data in doc("myfile.xml")/folder/info
return $data
```

XPath in XQuery

```
xquery version “3.0”;

for $data in doc("myfile.xml")//info
return $data
```

XPath in XQuery

```
xquery version “3.0”;

for $data in doc(“myf.xml")//info/../sibling
return $data/text
```

XPath in XQuery

```
xquery version “3.0”;

for $data in doc(“myf.xml")//info@attribute
return $data
```

Operators

- Math symbols:

  + - = * div > < >= <=

```
where $x + 733 = 1000
```

Integers and Strings

Integers are: `1  535  2345.343`

Strings are:

```
"my string"   'string of  text'
"anything /+&= goes"   ‘234’
```

Strings have indexes that start with 0:

```
 "my string"
```

`m` is `0`

`s` is `3`

IF expressions

```
xquery version "3.0";

for $data in doc("myfile.xml")/folder/info
return
     if ($data = "match")
          then ("data matches!")
     else ("data does not match")
```

IF expressions

```
xquery version "3.0";

for $data in doc("myfile.xml")/folder/info
return
     if ($data = "match")
          then ("data matches!")
     else if ($data = "no match")
          then ("data does not match")
     else ("ERROR")
```

Functions

- Magic Words

```
sum()       count()      string-join()
substring()              contains()
starts-with()            index-of()
```

```
for $data in doc("myfile.xml")//info
let $x := lower-case($data)
return $x
```

Formatting results in XML or HTML

```
for $x in doc("myfile.xml")/folder/info
return <element>{$x}</element>
```

```
for $x in doc("myfile.xml")/folder/info
return
     <root>
          <element>{data($x)}</element>
          <element>{$x@attrb}</element>
     </root>
```

Formatting results in XML or HTML

```
<root>
     {
     for $data in doc("myfile.xml")//info
     return
          <element>
               <tag>{data($data)}</tag>
               <tag>{$data@attrb}</tag>
          </element>
     }
</root>
```

XQuery can teach you about XML

- XML is very flexible
- Hard to predict how data will be used until you use it
- Breaks document-centric thinking
- Query and manipulate not reformat
- Further separate data storage and display

Example of Better Encoding

```
<physdesc label="Extent">
      <extent type="shelf">28.25 cubic feet</extent>
</physdesc>
```

```
<physdescstructured physdescstructuredtype="spaceoccupied">
        <quantity approximate="no">28.25</quantity>
        <unittype>cubic feet</unittype>
</physdescstructured>
```

Example of Better Encoding

```
<langusage>
        This finding aid is written in <language langcode="eng">English</language>
        with some materials in <language langcode="esp">Spanish<language>,
        and one document in <language langcode="fre">French</language>.
</langusage>


<langmaterial>
        <languageset>
                <language langcode="eng">English</language>
                <language langcode="esp">Spanish</language>
                <language langcode="fre">French</language>
        </languageset>
        <descriptivenote>
                This finding aid is written in English, with some materials in
                Spanish, and one document in French.
        </descriptivenote>
</langmaterial>
```

In-Class Exercise

- **Easier:** from the baseball collection, return a basic XML file that lists the name, team, and RBIs of each player that had over 90 RBIs
- **Medium:** From the baseball collection, return a HTML table listing player name, team, hits, RBIs, and WAR, sorted by hits
- **Hardest:** use the EAD files in the EAD folder to make a HTML table of collections, listing the collection title, unitdate, extent, and author