Engenharia de Ontologias (Ontology Engineering)

Universidade Federal de Uberlândia

Faculdade de Computação

Programa de Pós-Graduação em Ciência da Computação

Prof. Fabiano Azevedo Dorça

XQuery

- XQuery
  - XQuery é a linguagem de **consulta** de dados XML
  - XQuery para XML é como **SQL** para **bancos de dados**
  - XQuery é construído em **expressões XPath**
  - O XQuery é **suportado** por todos os principais **bancos de dados**
  - XQuery é uma recomendação do **W3C**
  - Bancos de dados XML

- https://www.w3schools.com/xml/xquery_intro.asp

XQuery

- Estrutura da Linguagem
  - Expressões FLWOR
    - **F**OR
    - **L**ET
    - **W**HERE
    - **O**RDER BY
    - **R**ETURN
  - Expressões XPath
  - Expressões Condicionais (análogo ao IF-THEN-ELSE das linguagens de programação)
  - Construtores de Elementos
  - Quantificador Existencial e Universal
  - Cast de Tipos

XQuery

- XQuery x XPath
  - XPath **não produz resultados** de consultas em uma estrutura **diferente** da existente no documento
  - XPath **não permite** realizar **junções** entre dados de dois documentos XML
- XQuery x XSLT
  - XSLT é mais adequado à tranformação e XQuery à consulta

XQuery

- XQuery é capaz de
  - Gerar respostas com **estrutura diferente** do documento consultado
  - Consultar **vários** documentos
  - Gerar **texto puro** ou fragmentos de documentos XML

XQuery

Exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>

<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>
```

XQuery

- Exemplo:

  doc("books.xml")/bookstore/book/title

- Resultado:

  <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>

XQuery

- Exemplo:

- doc("books.xml")/bookstore/book[price<30]

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

XQuery

- Exemplo:

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

- Resultado:

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

XQuery

- Exemplo:

```
<ul>
{
for $x in doc("books.xml")/bookstore/book/title
order by $x
return <li>{$x}</li>
}
</ul>
```

- Resultado:

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

XQuery

- Exemplo: eliminar o elemento title e mostrar apenas os dados dentro do elemento title:

   ← Construtor de elemento
  { for $x in doc("books.xml")/bookstore/book/title
  order by $x
  return {data($x)}
}

- Resultado:

  Everyday Italian
  Harry Potter
  Learning XML
  XQuery Kick Start

- Exemplo:

```
<html>
<body>

<h1>Bookstore</h1>

<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>
```

- Resultado:

```
<html>
<body>

<h1>Bookstore</h1>

<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>
```

- Exemplo: Adicionar atributos aos elementos HTML

```
<html>
<body>

<h1>Bookstore</h1>

<ul>
{
for $x in doc("books.xml")/bookstore/book
order by $x/title
return <li class="{data($x/@category)}">{data($x/title)}</li>
}
</ul>

</body>
</html>
```

- Resultado:

```
<html>
<body>
<h1>Bookstore</h1>

<ul>
<li class="COOKING">Everyday Italian</li>
<li class="CHILDREN">Harry Potter</li>
<li class="WEB">Learning XML</li>
<li class="WEB">XQuery Kick Start</li>
</ul>

</body>
</html>
```

- Exemplo: A palavra-chave “at” pode ser usada para contar a iteração:

  ```
  for $x at $i in doc("books.xml")/bookstore/book/title
  return <book>{$i}. {data($x)}</book>
  ```

- Resultado:

  ```
  <book>1. Everyday Italian</book>
  <book>2. Harry Potter</book>
  <book>3. XQuery Kick Start</book>
  <book>4. Learning XML</book>
  ```

XQuery

Exemplo: XML de entrada

```
<? xml version=“1.0” ?>
<empregados>
   <empregado cod=“E01” dept=“D01”>
      <nome>João</nome>
      <inicial-meio>S.</inicial-meio>
       <sobrenome>Santos</sobrenome>
   </empregado>
   <empregado cod=“E02” dept=“D01”>
       <nome>Ana</nome>
       <sobrenome>Ferraz</sobrenome>
   </empregado>
</empregados>
```

XQuery

- Exemplo:
  - Construtor de Elemento

<emp-nomes> ← Construtor de elemento
{for $e in doc(“emps.xml”)//empregado
return $e/nome
}
</emp-nomes>

  - Constrói no resultado um elemento empregados, que não existe no documento de origem

XQuery

- Resultado

```
<emp-nomes>
   <nome>João</nome>
   <nome>Ana</nome>
</emp-nomes>
```

Exemplo WHERE/ORDER BY

```
<emp-nomes>
{
for $e in doc(“emps.xml”)//empregado
    where $e/@dept= “D01”
    order by $e/nome
    return $e/nome
}
</emp-nomes>
```

XQuery

- Todos os empregados são ligados a **$e** (um de cada vez), mas a cláusula **return** só é executada para os que satisfazem a condição **$e/@dept=“D01”**

- Além disso, os resultados são ordenados por $e/nome

Resultado:

```
<emp-nomes>
    <nome>Ana</nome>
    <nome>João</nome>
</emp-nomes>
```

XQuery

- Exemplo CONSULTA ANINHADA

```
<departamentos>
  {for $d in distinct-values(doc("emps.xml")//empregado/@dept)
     return
     <departamento>
     <codigo>{$d}</codigo>
     <empregados>
        {for $e in doc("emps.xml")//empregado
           where $e/@dept=$d
           return
             <empregado>
               {$e/nome}
               {$e/sobrenome}
             </empregado>
        }
      </empregados>
      </departamento>
  }
</departamentos>
```

distinct-values retorna departamentos distintos (sem repetição)

XQuery

- Resultado

```
<departamentos>
    <departamento>
        <codigo>D01</codigo>
         <empregados>
                <empregado>
                    <nome>João</nome>
                    <sobrenome>Santos</sobrenome>
                </empregado>
                <empregado>
                    <nome>Ana</nome>
                    <sobrenome>Ferraz</sobrenome>
                </empregado>
        </empregados>
     </departamento>
</departamentos>
```

XQuery

- XQuery Conditional Expressions

- Exemplo:
  for $x in doc("books.xml")/bookstore/book
  return if ($x/@category="children")
  then <child>{data($x/title)}</child>
  else <adult>{data($x/title)}</adult>

- Resultado:
  <adult>Everyday Italian</adult>
  <child>Harry Potter</child>
  <adult>XQuery Kick Start</adult>
  <adult>Learning XML</adult>

- parênteses em torno da expressão if são obrigatórios. else é obrigatório, mas pode ser apenas else ().

XQuery

- XQuery Functions
- XQuery é construído em expressões XPath.
- XQuery e XPath compartilham o mesmo modelo de dados e suportam as mesmas funções e operadores.
- https://www.w3schools.com/xml/xpath_operators.asp
- https://www.w3schools.com/xml/xsl_functions.asp

XQuery

- Exemplos de chamadas de função
  Uma chamada para uma função pode aparecer onde uma expressão pode aparecer. Veja os exemplos abaixo:

- Exemplo 1: em um elemento
  <name>{upper-case($booktitle)}</name>

- Exemplo 2: no predicado de uma expressão de caminho
  doc("books.xml")/bookstore/book[substring(title,1,5)='Harry']

- Exemplo 3: em uma cláusula let
  let $name := (substring($booktitle,1,4))

XQuery

- Exemplos usando “let”

```
for $prod in doc("catalog.xml")//product
let $prodDept := $prod/@dept
where $prodDept = "ACC" or $prodDept = "WMN"
return $prod/name
```

- And /Or

```
for $prod in doc("catalog.xml")//product
let $prodDept := $prod/@dept
where $prod/number > 100
      and starts-with($prod/name, "F")
      and exists($prod/colorChoices)
      and ($prodDept = "ACC" or $prodDept = "WMN")
return $prod
```

exists
Retorna true se o valor dos argumentos NÃO for vazio, caso contrário, retorna false

XQuery

- Exemplo - Join:

```
for $item in doc("order.xml")//item,
    $prod in doc("catalog.xml")//product[number = $item/@num]
return <item num="{$item/@num}"
            name="{$prod/name}"
            quan="{$item/@quantity}"/>
```

- Resultado:

```
<item num="557" name="Fleece Pullover" quan="1"/>
<item num="563" name="Floppy Sun Hat" quan="1"/>
<item num="443" name="Deluxe Travel Bag" quan="2"/>
<item num="784" name="Cotton Dress Shirt" quan="1"/>
<item num="784" name="Cotton Dress Shirt" quan="1"/>
<item num="557" name="Fleece Pullover" quan="1"/>
```

XQuery

- API XQuery JAVA
  - Package javax.xml.xquery

XQuery

- Exchanger XML
  - Execução de consultas Xpath, XQuery, validação DTD, XML Schema, etc.
  - http://exchanger-xml-lite.software.informer.com/3.2/

XQuery

- Fim.