

XQuery

Lino Alves

XQuery

- XQuery é para XML o que SQL é para tabelas de banco de dados.
- XQuery foi projetado para operar sobre dados em XML.
- Dados XML = arquivos + databases
- XQuery é recomendado pelo W3C

XQuery

- XQuery é uma linguagem para busca e extração de elementos e atributos de dados XML.
- XQuery pode ser usado para:
 - Extrair informação de XML
 - Gerar relatórios
 - Transformar dados XML em XHTML
 - Procurar informação relevante em documentos web.

XQuery

- XML Exemplo: book.xml

```
<bookstore>
```

```
<book category="WEB">
```

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

```
<author>James McGovern</author>
```

```
<author>Per Bothner</author>
```

```
<year>2003</year>
```

```
<price>49.99</price>
```

```
</book>
```

```
</bookstore>
```

XQuery

- Expressões de caminho

```
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

- Predicados

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

```
doc("books.xml")/bookstore/book[price>30]/title
```

XQuery

- Expressões FLWOR
- For Let Where Order by Return

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

XQuery

- Apresentando resultados em HTML

```
<ul>
```

```
{
```

```
for $x in doc("books.xml")/bookstore/book/title
```

```
order by $x
```

```
return <li>{$x}</li>
```

```
}
```

```
</ul>
```


XQuery

- 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>
```

```
</ul>
```

XQuery

- Pegando só o conteúdo:

```
<ul>
```

```
{
```

```
for $x in doc("books.xml")/bookstore/book/title
```

```
order by $x
```

```
return <li>{data($x)} </li>
```

```
}
```

```
</ul>
```

XQuery

- Resultado:

```
<ul>
```

```
<li>Everyday Italian</li>
```

```
<li>Harry Potter</li>
```

```
<li>Learning XML</li>
```

```
<li>XQuery Kick Start</li>
```

```
</ul>
```

XQuery

```
<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>
```

XQuery

- Resultado:

```
<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>
```

XQuery

- Sintaxe:
- case-sensitive
- Strings podem estar entre aspas simples ou duplas.
- Variáveis são definidas com \$ seguido de nome, e.g. \$x.
- Comentários são delimitados por (: :), e.g. (: comentário :).

XQuery

- If then else

```
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>
```

XQuery

- For:

```
for $x in (1 to 5)
```

```
return <test>{$x}</test>
```

```
for $x at $i in
```

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

```
return <book>{$i}. {data($x)}</book>
```


XQuery

- Resultado:

<book>1. Everyday Italian</book>

<book>2. Harry Potter</book>

<book>3. XQuery Kick Start</book>

<book>4. Learning XML</book>

XQuery

for \$x in (10,20), \$y in (100,200)

return <test>x={\$x} and y={\$y}</test>

<test>x=10 and y=100</test>

<test>x=10 and y=200</test>

<test>x=20 and y=100</test>

<test>x=20 and y=200</test>

XQuery

- Let

```
let $x := (1 to 5)
```

```
return <test>{$x}</test>
```

- Resultado:

```
<test>1 2 3 4 5</test>
```

XQuery

- Comparações:
- Comparação em geral
= , != , < , <= , > , >=
- Comparação de um valor
eq , ne , lt , gt , ge

XQuery

- Funções:

```
<name>{uppercase($booktitle)}</name>
```

```
let $name := (substring($booktitle,1,4))
```

- Declaração:

```
declare function prefix:function_name($parameter AS  
datatype)
```

```
AS returnDatatype
```

```
{
```

```
...function code here...
```

```
}
```

XQuery

```
declare function local:minPrice($p as xs:decimal?,$d as xs:decimal?)
AS xs:decimal?
{
let $disc := ($p * $d) div 100
return ($p - $disc)
}
```

- Chamando a função:

```
<minPrice>{local:minPrice($book/price,$book/discount)}</minPrice>
```

XQuery

- XQilla

<http://xqilla.sourceforge.net/>

- W3schools.com

<http://www.w3schools.com/xquery/>