

XQuery

RA6-GESTIONA LA INFORMACIÓ EN FORMATS
D'INTERCANVI DE DADES ANALITZANT I UTILITZANT
TECNOLOGIES D'EMMAGATZEMATGE I LLENGUATGES
DE CONSULTA.

Contingut

- Definició.
- Per a què usar XQuery
- Tipus de dades suportades
- Funcions XQuery
- FLWOR
- Document libros.xml
- Creació de nodes
- BaseX
- Exemples
- Consultes
- Diferències entre for i let
- Expressions condicionals
- Qualificadors existencials
- Operadors i funcions
- Qüestions
- Enllaços d'interès

Definició

- Llenguatge que permet de forma ràpida fer consultes o recorreguts sobre col·leccions de dades en documents XML. El resultat obtingut són els nodes que compleixen certes condicions.
- És un llenguatge independent de la font de dades: poden ser documents XML o bases de dades relacionals.
- Tot i que es pot comparar amb SQL, en XQuery és important la jerarquia i l'ordre de les dades.
- Es basa en XPath per a seleccionar la informació.
- Una consulta XQuery és una expressió que llegeix una seqüència de dades i retorna una altra.

Per a què usar XQuery

- Extracció de dades de XML databases.
- Transformació de jerarquies: transformar unes estructures de dades XML en altres estructures que organitzen la informació de forma diferent.
- Transformació de dades XML en diferents formats (per exemple HTML, JSON).
- Construcció de serveis web que consumeixen i produeixen dades XML.
- Creació d'informes i visualitzacions complexes a partir de dades XML.

Tipus de dades suportades en XQuery

Suporta diversos tipus de dades, entre ells:

Tipus atòmics: tipus primitius per a representar valors individuals com:

- xs:string: Text data.
- xs:integer: Integer values.
- xs:decimal: Decimal numbers.
- xs:boolean: True or false values.
- xs:date: Date values.
- xs:time: Time values.

Tipus Seqüència: representen col·leccions ordenades de valors atòmics o nodes.

Tipus Node: representen elements, atributs, documents i altres parts d'un document XML.

Funcions XQuery

Les funcions estan predefinides, permeten fer operacions amb les dades i fan tasques comunes. Per exemple:

fn:concat(): Concatena strings.

fn:substring(): Extracts una porció d'una cadena.

fn:count(): Compta el número d'ítems en una seqüència.

fn:sum(): Calcula la suma de valors numèrics.

fn:min() i fn:max(): Troben el valor mínim i màxim en una seqüència.

Podeu trobar les funcions aquí:

<https://www.datypic.com/xq/>

FLWOR

Les consultes segueixen la norma FLWOR, acrònim de (For, Let, Where, Order, Return). Són les 5 clàusules que s'utilitzen en expressions XQuery.

- *For*: vincula una o més variables a expressions XPath. Itera sobre una seqüència de ítems
- *Let*: assigna un valor a una variable.
- *Where*: filtra els valors segons unes condicions.
- *Order by*: ordena els valors segons un criteri
- *Return*: especifica el resultat de l'expressió.

Combinant aquestes clàusules es formen expressions per a manipular les dades.

Document libros.xml

Imagineu un document que conté la informació d'una sèrie de llibres com aquest:

```
<?xml version="1.0" encoding="UTF-8"?>
<bib>
  <libro año="1994">
    <titulo>TCP/IP Illustrated</titulo>
    <autor>
      <apellido>Stevens</apellido>
      <nombre>W.</nombre>
    </autor>
    <editorial>Addison-Wesley</editorial>
    <precio> 65.95</precio>
  </libro>
  <libro año="1992">
    <titulo>Advan Programming for Unix environment</titulo>
    <autor>
      <apellido>Stevens</apellido>
      <nombre>W.</nombre>
    </autor>
    <editorial>Addison-Wesley</editorial>
    <precio>65.95</precio>
  </libro>
  ...
</bib>
```


Creació de nodes

- Per a indicar una expressió XQuery i que sigui avaluada s'ha d'escriure entre claus:

{ XQuery expression }

- Amb XQuery:
 - es poden crear nodes
 - es poden ordenar valors

BaseX

Per a treballar amb XQuery descarrega l'aplicació BaseX:

<https://basex.org/download/>

Escull el fitxer .zip i descomprimeix.

La documentació de l'aplicació es pot consultar a:

https://docs.basex.org/main/Getting_Started

Crea una carpeta de nom *Llibres*

Descarrega el document XML del classroom.

Realitza les consultes que trobaràs a continuació

Exemples de consultes

Consulta la wiki

https://en.wikibooks.org/wiki/XQuery#Beginning_Examples

- [HelloWorld](#) - A simple test to see if XQuery is installed correctly.
- [FLWOR Expression](#) - A basic example of how XQuery FLWOR statements work.
- [Sequences](#) - Working with sequences is central to XQuery.
- [XPath examples](#) - Sample XPath samples for people new to XML and XPath
- [Regular Expressions](#) - Regular expressions make it easy to parse text.
- [Displaying Lists](#) - How to take a list of values in an XML structure and return a comma separated list.
- [Displaying data in HTML Tables](#) - How to display XML data in an HTML table.
- [Filtering Words](#) - How to test to see if a word is on a list.

Consultes

Exemple de for:

```
for $b in doc("libros.xml")//bib/libro
```

- La variable `$b` agafa com a valor tots els nodes *libro* dintre de l'arxiu “libros.xml”

NOTA: No es necessari que aparegui cap de les cinc clàusules FLWOR en una consulta.

Consultes

- Consulta dels títols dels llibres de l'any 2000

```
for $b in doc("libros.xml")//libro
where $b/@año = "2000"
order by $b/titulo
return $b/titulo
```

- La clàusula `for` assigna a la variable `$b` cada un dels nodes `<libro> <\libro>` que apareixen en qualsevol lloc del document XML.
- La clàusula `where` fa un filtre dels valors de la variable `$b` comprovant per a cada valor de `$b` si l'atribut `año` és igual a 2000. Si ho és, s'inclou en la solució
- La clàusula `order by` ordena els nodes `título` seleccionats, per ordre alfabètic de contingut
- La clàusula `return` torna como els nodes seleccionats

Diferències entre les clàusules for i let

- La clàusula `for` vincula una variable amb cada node que troba en la col·lecció de dades

Exemple:

```
for $b in doc("libros.xml")/bib/libro/titulo
where $b/@año = "2000"
return <titulos>{ $b }</titulos>
```

`for` assigna a `$b` cada un dels llibres de l'any 2000 que apareixen en tot el document XML. En el resultat es repeteix 2 vegades l'etiqueta `<titulos>`

- `let` vincula una variable amb tot el resultat d'una expressió

Substituint `for` per `let` en la consulta anterior:

```
let $b in doc("libros.xml")/bib/libro/titulo
where $b/@año = "2000"
return <titulos>{ $b }</titulos>
```

La variable `$b` es vincula una sola vegada a tots els títols de tots els llibres de l'any 2000. L'etiqueta `<titulos>` només apareix un cop.

Diferències entre les clàusules for i let

- Si una clàusula let apareix en una consulta que ja té una o més clàusules for, els valors de la variable vinculada per la clàusula let s'afegeix a cada una de les tuples generades per la clàusula for.
- Si en la consulta apareix més d'una clàusula for (o més d'una variable en una clàusula for), el resultat és el producte cartesià de les variables.

Expressions condicionals

- XQuery suporta expressions condicionals del tipus if-then-else (com en els llenguatges de programació més comuns).
- Una expressió condicional permet crear una o una altra estructura de nodes en el resultat depenent dels valors de las tuples filtrades.

Exemple:

```
for $b in doc("libros.xml")//libro
return
<libro>
{ $b/titulo }
{
for $a at $i in $b/autor
where $i <= 2
return <autor>{string($a/apellido), ", ", string($a/nombre)}</aut>
{
if (count($b/autor) > 2) then <autor>et al.</autor>
else ()
}
}
</libro>
```


Qualificadors existencials (some, every)

- Aquests quantificadors permeten definir consultes que retornin algun element que satisfaci la condició some o consultes que retornin els elements en els que tots els nodes satisfacin la condició every.

Exemple:

```
for $b in doc("libros.xml")//libro
where some $a in $b/autor
satisfies ($a/apellido="Stevens" and $a/nombre="W.")
return $b/titulo
```

Operadors i funcions

- **matemàtiques**: +, -, *, div, idiv, mod, count(), min(), max(), avg(), sum()
- **de cadenes**: concat(), string-length(), startwith(), ends-with(), substring(), upper-case(), lower-case(), string()
- per al tractament d'expressions regulars
- **comparació**: =; !=; <; >; <=; >; not()
- **manipulació de seqüències**: union (|), intersect, except
- manipulació de nodes XML
- comprovació i conversió de tipus

Qüestions

Xquery vs XSLT

Tant XSLT com XQuery es poden utilitzar per transformar dades XML. XSLT és una transformació basada en documents (plantilles), mentre que XQuery és una transformació basada en consultes (estructura semblant a SQL). Per tant, XQuery acostuma a ser més fàcil d'aprendre. XQuery acostuma a ser el favorit per a treballs més grans i pot suportar bases de dades relacionals.

Puc fer tot en JavaScript que puc fer a XQuery? I viceversa?

Sí, finalment. JavaScript del costat del servidor es basa en la mateixa base de C++ que utilitza el temps d'execució Xquery.

Enllaços d'interès

- www.w3schools.com/xquery/default.asp
- www.w3.org/XML/Query/
- <http://es.scribd.com/doc/30081210/XQuery-1>
- <https://www.youtube.com/watch?v=zJpr6eeugXo>
- Funcions Xquery: <https://www.datypic.com/xq/>