IIC3432 - Tópicos Avanzados en Bases de Datos

Extracción de Información en XML: XPath con atributos y XQuery

DTD: Biblioteca

```
<!DOCTYPE bib [
  <!ELEMENT bib (book*)>
  <!ELEMENT book (title, author+)>
    <!ATTLIST book
              isbn CDATA #REQUIRED
              year CDATA #REQUIRED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
]>
```

Documento XML: Biblioteca

```
<bib>
  <book isbn="0-471-34506-7" year="2000">
    <title>Theory of Computational Complexity</title>
    <author>Du</author>
    <author>Ko</author>
  </book>
  <book isbn="1-55860-622-X" year="2000">
    <title>Data on the Web</title>
    <author>Abiteboul</author>
    <author>Buneman</author>
    <author>Suciu</author>
  </book>
</bib>
```

XPath: Extracción de valores

Caminos básicos:

$$\texttt{paso} \quad ::= \quad \texttt{child} \mid \texttt{parent} \mid \texttt{right} \mid \texttt{left}$$

Expresiones para caminos:

$$\begin{aligned} \texttt{camino} \quad ::= \quad & \texttt{paso} \mid \texttt{paso}^* \mid \texttt{camino/camino} \mid \\ & \texttt{camino} \cup \texttt{camino} \mid ?\texttt{test} \end{aligned}$$

Consulta XPath:

$$\texttt{consulta} \quad ::= \quad \texttt{camino} \mid \texttt{camino/string()} \mid \texttt{camino/@}\mathit{atributo}$$

XPath: Extracción de valores

Filtros:

$$\texttt{test} \quad ::= \quad \texttt{termino op termino} \mid \textit{etiqueta} \mid \langle\texttt{camino}\rangle \mid \neg\texttt{test} \mid \texttt{test} \wedge \texttt{test}$$

Términos:

$$\texttt{termino} \quad ::= \quad \textit{constante} \mid \texttt{string}() \mid @\textit{atributo} \mid \texttt{position}() \mid \texttt{last}()$$

Operadores:

$$\texttt{op} \quad ::= \quad = \mid \neq \mid < \mid \leq \mid > \mid \geq$$

XPath: Ejercicios

1. Lista de títulos de libros.
2. Lista de números isbn en la biblioteca.
3. Lista de títulos de libros publicados el año 2000.
4. Lista de primeros autores de libros.
5. Lista de últimos autores de libros.
6. Lista de títulos de libros publicados por Abiteboul.
7. Lista de títulos de libros publicados por Abiteboul después de 1995.
8. Lista de títulos de libros publicados por Abiteboul en los que él no es el primer autor.

XPath: Respuestas a ejercicios

1. Lista de títulos de libros:

   $\texttt{child}^*/?book/\texttt{child}/?title/\texttt{string}()$

2. Lista de números isbn en la biblioteca.

   $\texttt{child}^*/?book/@isbn$

3. Lista de títulos de libros publicados el año 2000:

   $\texttt{child}^*/?(book \wedge @year = 2000)/\texttt{child}/?title/\texttt{string}()$

4. Lista de primeros autores de libros:

   $\texttt{child}^*/?book/\texttt{child}/?(author \wedge position() = 2)/\texttt{string}()$

5. Lista de últimos autores de libros:

   $\texttt{child}^*/?book/\texttt{child}/?(author \wedge position() = last())/\texttt{string}()$

XPath: Respuestas a ejercicios

6. Lista de títulos de libros publicados por Abiteboul:

$$\texttt{child}^*/?(book \wedge \langle child/?(author \wedge string() = Abiteboul)\rangle)/\texttt{child}/?title/\texttt{string}()$$

7. Lista de títulos de libros publicados por Abiteboul después de 1995:

$$\texttt{child}^*/?(book \wedge @year > 1995 \wedge \langle child/?(author \wedge string() = Abiteboul)\rangle)/\texttt{child}/?title/\texttt{string}()$$

8. Lista de títulos de libros publicados por Abiteboul en los que él no es el primer autor:

$$\texttt{child}^*/?(book \wedge \langle child/?(author \wedge position() > 2 \wedge string() = Abiteboul)\rangle)/\texttt{child}/?title/\texttt{string}()$$

XPath: Otro ejercicio

Suponga que cambiamos:

```
<!ELEMENT book (title, author+)>
```

por:

```
<!ELEMENT book (title, author+, section*)>
```

¿Qué consulta retorna la lista de últimos autores de libros?

XPath: Otro ejercicio

Suponga que cambiamos:

```
<!ELEMENT book (title, author+)>
```

por:

```
<!ELEMENT book (title, author+, section*)>
```

¿Qué consulta retorna la lista de últimos autores de libros?

$$\texttt{child}^*/?book/\texttt{child}/?(author \wedge (position() = last() \vee \langle \texttt{right}/?section \rangle))/\texttt{string}()$$

DTD Recursivo: Biblioteca

```
<!DOCTYPE bib [
  <!ELEMENT bib (book*)>
  <!ELEMENT book (title, author+, section*)>
    <!ATTLIST book
              isbn CDATA #REQUIRED
              year CDATA #REQUIRED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT section (title, section*)>
]>
```

Documento XML: Biblioteca (`bib.xml`)

```
<bib>
  <book isbn="0-471-34506-7" year="2000">
    <title>Theory of Computational Complexity</title>
    <author>Du</author>
    <author>Ko</author>
  </book>
  <book isbn="1-55860-622-X" year="2000">
    <title>Data on the Web</title>
    <author>Abiteboul</author>
    <author>Buneman</author>
    <author>Suciu</author>
  </book>
</bib>
```

DTD: Amazon

```
<!DOCTYPE amazon [
  <!ELEMENT amazon (book*)>
  <!ELEMENT book (title, author+, price, review*)>
    <!ATTLIST book
              isbn CDATA #REQUIRED
              year CDATA #REQUIRED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT price (#PCDATA)>
  <!ELEMENT review (#PCDATA)>
]>
```

Documento XML: Amazon (amazon.xml)

```
<amazon>
  <book isbn="0-471-34506-7" year="2000">
    <title>Theory of Computational Complexity</title>
    <author>Du</author>
    <author>Ko</author>
    <price>114.48</price>
    <review>Overall, I would recommend this book as an
    excellent addition to the literature.</review>
    <review>Promises to become the standard reference
    on computational complexity</review>
  </book>
</amazon>
```

XQuery 1.0: An XML Query Language

- Versión actual (3 de Noviembre de 2005): W3C Candidate Recommendation.

  `http://www.w3.org/TR/xquery/`

- Editores:

Scott Boag	IBM Research
Don Chamberlin	IBM Almaden Research Center
Mary F. Fernández	AT&T Labs
Daniela Florescu	Oracle
Jonathan Robie	DataDirect Technologies
Jérôme Siméon	IBM T.J. Watson Research Center

XQuery 1.0: An XML Query Language

- Implementación completa: Galax
  - No es la implementación más eficiente de XQuery.
  - Ultima versión: 0.6.5, 17 de Mayo de 2006.
    Esta versión está basada en la última versión de XQuery (Noviembre de 2005).

- Buena referencia:

  XQuery from the Experts: A Guide to the W3C XML Query Language. Addison Wesley.

XQuery

XQuery es la suma de:

- XPath + expresiones FLWOR

  FLWOR: For, Let, Where, Order by, Return

- Instrucciones para construir (transformar) documentos XML

- Funciones definidas por el usuario

XQuery: Let

`let:` Permite asociar una variable XQuery al resultado de una consulta XPath.

- Variable XQuery es de la forma `$nombre.`

  Ejemplos: `$book`, `$review`, ...

- Una variable puede asociarse a un documento: `doc("...")`

```
let $documento :=
    doc("http://www.ing.puc.cl/~marenas/bib.xml")
```

- Una variable puede asociarse a una consulta XPath:

```
let $libro := $documento/child*/?book
```

XQuery: For

`for:` Permite iterar sobre los nodos o valores obtenidos de ejecutar una consulta XPath.

- Ejemplo simple: `for $x in $documento/child/?book`

- `for` puede iterar sobre más de una variable:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
let $amazon := doc("http://www.ing.puc.cl/~marenas/amazon.xml")
for $x in $bib/child/?book,
    $y in $amazon/child/?book
```

XQuery: Where

`where:` Permite verificar si un conjunto de nodos satisface una determinada condición.

- `where` puede ser utilizado para mezclar (join) información de distintos documentos:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
let $amazon := doc("http://www.ing.puc.cl/~marenas/amazon.xml")
for $x in $bib/child/?book,
    $y in $amazon/child/?book
where $x/@isbn = $y/@isbn
```

XQuery: Return

`return:` Permite retornar valores.

- Tags XML pueden aparecer como parte del patrón de retorno.
- {`expresión`}: Indica que se debe reemplazar `expresión` por el resultado de evaluarla.

```
<doc>
   <isbn> $x/@isbn </isbn>
   <year> {$x/@year} </year>
</doc>
```

Resultado:

```
<isbn> $x/@isbn </isbn>
<isbn> 2000 </isbn>
```

XQuery: Return

- Una consulta XQuery puede retornar un documento XML:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
     for $x in $bib/child*/?book
     return <isbn> {$x/@isbn} </isbn>
   }
   </lista>
```

Resultado:

```
<lista>
  <isbn> 0-471-34506-7 </isbn>
  <isbn> 1-55860-622-X </isbn>
</lista>
```

XQuery: Return

- Valores retornados pueden incluir atributos:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in $bib/child*/?book
      return <book isbn="{$x/@isbn}"/>
   }
   </lista>
```

Resultado:

```
<lista>
  <book isbn="0-471-34506-7"/>
  <book isbn="1-55860-622-X"/>
</lista>
```

Ejemplo: Agrupación

Lista de libros de la Biblioteca con sus comentarios de Amazon:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
let $amazon := doc("http://www.ing.puc.cl/~marenas/amazon.xml")
return
```

Ejemplo: Agrupación

```
<lista>
{
   for $x in $bib/child/?book
   return
      <book isbn="{$x/isbn}">
      {
         for $y in $amazon/child/?book
         where $x/@isbn = $y/@isbn
         return
            for $z in $y/child/?review
            return <review> {$z/string()} </review>
      }
      </book>
}
</lista>
```

Ejemplo: Verificación de tipos

Lista de pares de libros distintos con el mismo título:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in $bib/child/?book, $y in $bib/child/?book
      where $x/@isbn != $y/@isbn and
            $x/child/?title/string() = $y/child/?title/string()
      return
         <book title="{$x/child/?title/string()}">
            <isbn> {$x/@isbn} </isbn>
            <isbn> {$y/@isbn} </isbn>
         </book>
   }
   </lista>
```

XQuery: Order by

`order by`: Permite ordenar el resultado de una consulta

Ejemplo:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in $bib/child*/?book
      order by $x/@isbn descending
      return <isbn> {$x/@isbn} </isbn>
   }
   </lista>
```

XQuery: Order by

Resultado:

```
<lista>
  <book isbn="1-55860-622-X"/>
  <book isbn="0-471-34506-7"/>
</lista>
```

En este caso: Archivo es ordenado de manera descendente por el valor de `isbn`.

Ejemplo: Transformación y ordenación

Lista alfabética de autores con sus libros:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in $bib/child/?book/child/?author
      order by $x/string() ascending
      return
```

Ejemplo: Transformación y ordenación

```
        <author>
            <name> {$x/string()} </name>
            {
                for $y in $bib/child/?book/child/?author
                where $x/string() = $y/string()
                return
                    <book>
                        <title> {$y/left*/?title/string()} </isbn>
                        <isbn> {$y/parent/@isbn} </isbn>
                        <year> {$y/parent/@year} </year>
                    </book>
            }
        </author>
}
</lista>
```

Solución alternativa: Transformación y ordenación

Lista alfabética de autores con sus libros:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in $bib/child/?book/child/?author/string()
      order by $x ascending
      return
```

Solución alternativa: Transformación y ordenación

```
        <author>
            <name> {$x} </name>
            {
                for $y in $bib/child/?book/child/?author
                where $x = $y/string()
                return
                    <book>
                        <title> {$y/left*/?title/string()} </isbn>
                        <isbn> {$y/parent/@isbn} </isbn>
                        <year> {$y/parent/@year} </year>
                    </book>
            }
        </author>
}
</lista>
```

Ejemplo: Eliminación de repeticiones

Lista alfabética de autores con sus libros pero sin repeticiones:

```
let $bib := doc("http://www.ing.puc.cl/~marenas/bib.xml")
return
   <lista>
   {
      for $x in distinct-values(
                  $bib/child/?book/child/?author/string())
      order by $x ascending
      return
         ...
```

XQuery: Ejercicios

1. De la lista de libros para los cuales Amazon tiene registrado más de un precio.

2. De la lista de libros para los cuales Amazon no tiene comentarios.

3. De la lista de libros en la Biblioteca que tienen más de un autor.

4. ¿Puede escribir la consulta (Conditional XPath)

   `child/?book/child/?section/(child/?section)`*`/string()`

   en XQuery?