

BD Nativas XML

- Una BD nativa XML almacena documentos XML.
- A las BD nativas XML se les denomina BD centradas en documentos o BD documentales. Sería una BD no SQL ya que no usa SQL como lenguaje principal de consultas.

XQuery

- XQuery es un lenguaje para encontrar y extraer elementos y atributos de documentos XML.
- XQuery es una recomendación del W3C desde enero de 2007.
- XQuery es al XML, lo que SQL es a las bases de datos relacionales.
- La entrada y la salida de una consulta XQuery corresponden a un documento o fragmento de documento XML.

Uso de XPath

- Cuando se va a analizar un documento XML se crea un árbol de nodos.
- La extracción de la información durante el recorrido del árbol consiste en la detección de los nodos a buscar y el procesamiento de la información que se quiere extraer ➔ XPath.

XQuery y XPath

Títulos de los libros de un documento llamado “*books.xml*” cuyo precio se mayor de 30€:

XPath:

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

XQuery:

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

Con la función `doc()` se lee un documento XML que se indique como parámetro y devuelve el nodo raíz o los elementos que se indiquen mediante una expresión XPath.

ELEMENTOS DE SINTAXIS

- XQuery es sensible a mayúsculas.
- Las variables, atributos y elementos en XQuery deben seguir las reglas de nombres válidos en XML:
 - Las cadenas pueden escribirse entre comillas simples o dobles.
- Las variables en Xquery se definen con un \$ seguido por un nombre válido.
 - Ej: \$libro.
- En XQuery los comentarios se delimitan con (: y :).
 - Ej: (: Esto es un comentario :)

FLWOR

- La estructura sintáctica de las expresiones en XQuery se denomina *FLWOR*.
- *FLWOR* es un acrónimo de *For*, *Let*, *Where*, *Order by* y *Return*.
 - *for* indica qué elementos se van a seleccionar.
 - *let* permite declarar variables a las que se les asignan valores.
 - *where* permite introducir condiciones que deben de cumplir los elementos seleccionados.
 - *order by* permite la ordenación de la salida.
 - *return* devuelve los resultados.

FLOWR + HTML

```
<ul>
{
  for $x in /centro/curso/titulo
  order by $x
  return <li>{$x}</li>
}
</ul>
```

FOR

- Es como la cláusula “from” de SQL.
- Proporciona una serie de variables cuyos valores son los resultados de expresiones XPath.

```
for $b in doc ("libros.xml")//libro
where $b/@id=2
return $b/titulo
```


LET

- Cláusula opcional.
- Establece una nueva variable sobre el mismo u otro documento XML. Permite simplificar las expresiones posteriores.

```
for $x in /libros/libro
let $tit:=$x/titulo
where $x/precio>20
order by $x/autor
return <tituloLibro>{$tit}</tituloLibro>
```

- Admite la asignación de valores de rango a variables:

```
let $x := (1 to 5)
return <test>{$x}</test>
```

RESULTADO: <test>1 2 3 4 5</test>

Diferencias entre For y Let

Una tupla por cada título:

```
for $t in /libros/libro/titulo return  
  <titulos>{$t}</titulos>
```

Una única tupla con los títulos:

```
let $t in /libros/libro/titulo return  
  <titulos>{$t}</titulos>
```

Uniones naturales

```
for $a in /banco/cuenta,  
    $c in /banco/cliente,  
    $i in /banco/impositor  
where $a/numero_cuenta=$i/numero_cuenta and  
    $c/nombre_cliente=$i/nombre_cliente  
return <cuenta_cliente>{$c $a}</cuenta_cliente>
```

Consultas anidadas

```
for $c in /banco/cliente,  
return <cliente>  
  {$c/*}  
  {for $i in /banco/impositor[nombre_cliente=$c/nombre_cliente],  
    $a in /banco/cuenta[numero_cuenta=$i/numero_cuenta]  
    return $a}  
</cliente>
```

FUNCIONES

XQuery incorpora más de 100 funciones predefinidas para la manipulación de los datos.

```
doc("books.xml")/bookstore/book[substring(title,1,5)='Harry']
```

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

CONDICIONALES

```
for $x in /bookstore/book
return
if ($x/@category="children") then
  <child>{data($x/title)}</child>
else
  <adult>{data($x/title)}</adult>
```

Nota: Se requiere else, pero puede ser solo else ().