

Consultas FLWOR con XBase 2

Tratamiento con atributos

Para acceder al atributo de un elemento se utiliza la @. Por ejemplo:

```
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/@propietario = 1
return $suministra/@identificador
```

En la anterior consulta se está haciendo un filtrado por el propietario de de suministra (que es un atributo) y se está recuperando con el return el identificador que de suministra (que es un atributo).

Return con etiqueta raíz

Para extraer información en una consulta XQuery y que quede marcada por una etiqueta raíz, al igual que hacemos con las etiquetas en el return, meteremos toda la consulta entre las etiquetas y llaves:

```
<etiquetaraiz>
{
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/cantidad > 450
return $suministra
}
</etiquetaraiz>
```

Extracción de varios elementos

Para extraer varios elementos de una tupla en la cláusula del return:

```
<etiquetaraiz>
{
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/cantidad > 450
return <suministra> { $suministra/proyecto, $suministra/cantidad } </suministra>
}
</etiquetaraiz>
```

**Cuando devolvemos varios elementos, es obligatorio que estos estén enmarcados entre etiquetas.*

Ordenación en XQuery

Si se desea ordenar un conjunto de datos puede usarse la cláusula `order by` poniendo después uno o varios elementos o atributos y usando `ascending` o `descending`, de manera similar a SQL.

Si tenemos la consulta:

```
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/cantidad > 450
return $suministra
```

Para ordenar la consulta anterior por cantidad usamos esto:

```
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/cantidad > 450
order by $suministra/cantidad descending
return $suministra
```

Igual que en SQL se pueden combinar varios campos. Si por ejemplo quisiéramos ordenar por proveedor ascendente y luego por parte descendente haríamos esto.

```
for $suministra in doc("datos.xml")/datos/suministros/suministra
where $suministra/cantidad > 450
order by $suministra/proveedor ascending, $suministra/parte descending
return $suministra
```

Operaciones matemáticas

Se permiten hacer operaciones matemáticas en cualquier cláusula, por ejemplo si queremos calcular una multiplicación y guardar el resultado en una variable en una variable, para después recuperar su resultado:

```
for $suministra in doc("datos.xml")/datos/suministros/suministra
let $doblecantidad := $suministra/cantidad * 2
return <doble> { $doblecantidad } </doble>
```

Funciones en XQuery

XQuery permite aplicar procesamiento extra a los nodos de un XML. A continuación se nombran algunas muy usadas:

- `concat($fila, ' ')`: concatena dos elementos, en este caso pone un espacio tras los datos de fila.

- `string-length($elemento)`: devuelve la longitud de una cadena.
- Funciones de agregación:
 - `sum($elemento)`
 - `count($elemento)`
 - `avg($elemento)`
 - `max($elemento)`
 - `min($elemento)`

Se pueden aplicar directamente a un conjunto sin necesidad de hacer un bucle `for`.