

FLWOR

FLWOR (pronunciado flower, flor en inglés) es un lenguaje de consultas que se utiliza en XQuery. Similar a SQL en un sistema de bases de datos relacionales clásico.

El nombre FLWOR proviene de las expresiones **for**, **let**, **where**, **order by**, **return**.

Todas estas expresiones son opcionales excepto **return**, ya que es la que devuelve los resultado de la consulta.

Lo primero, **for** y **let** sirven para crear tuplas con las que trabajará el resto de las cláusulas de la consulta, y pueden usarse tantas veces como se desee en una misma consulta, incluso dentro de otras cláusulas. Sin embargo sólo puede declararse una única cláusula **where**, una única cláusula **order** y una única cláusula **return**.

Entrada de datos

XQuery utiliza las funciones de entrada en las cláusulas **for** o **let** para identificar el origen de los datos. Actualmente XQuery define dos funciones de entrada distintas, **doc(URI)** y **collection(URI)**.

La función **doc(URI)** devuelve el nodo documento, o nodo raíz, del documento referenciado por un identificador universal de recursos (URI). Esta es la función más habitual para acceder a la información almacenada en archivos.

La función **collection(URI)** devuelve una secuencia de nodos referenciados por una URI, sin necesidad de que exista un nodo documento o nodo raíz. Esta es la función más habitual para acceder a la información almacenada en una base de datos que tenga capacidad para crear estructuras de datos XML. En este curso no vamos a utilizarlo.

Ejemplo

Partimos del siguiente ejemplo para explicar las expresiones (informacion.xml):

```

<pelicula fechaEstreno="19/12/2007">
  <titulo>Los Otros </titulo>
  <director>Alejandro Amenabar</director>
</pelicula>
<pelicula fechaEstreno="15/11/1986">
  <titulo>Sangre Facil </titulo>
  <director>Ethan Coen </director>
</pelicula>
<pelicula fechaEstreno="14/06/1990">
  <titulo>No es país para pobres </titulo>
  <director>Ethan Coen </director>
  <director>Joel Coen </director>
</pelicula>
<pelicula fechaEstreno="19/12/2007">
  <titulo> The Lovely Bones </titulo>
  <director> Peter Jackson</director>
</pelicula>

```

FOR y LET

Usando **FOR**, cada nodo recuperado por la expresión, es vinculado a una única tupla. Por ejemplo:

```

FOR $b IN doc('informacion.xml')//película/director
RETURN $b

```

Recuperaremos todos los nodos de todas las películas del documento:

```

<directores>
  <director>Peter Jackson</director>
</directores>
<directores>
  <director>Alejandro Amenabar</director>
</directores>
<directores>
  <director>Ethan Coen</director>
</directores>
<directores>
  <director>Joel Coen</director>
</directores>

```

Usando **LET** todos los nodos recuperados estarán en una única tupla. Por ejemplo:

```
LET $b := doc('informacion.xml')//película/director
RETURN $b
```

Obtendremos lo siguiente:

```
<directores>
  <director>Peter Jackson</director>
  <director>Alejandro Amenabar</director>
  <director>Ethan Coen</director>
  <director>Joel Coen</director>
</directores>
```

WHERE

Filtra las tuplas producidas por las cláusulas LET y FOR, recuperando únicamente aquellas que cumplan una expresión. Dicha expresión es evaluada para cada tupla. Si su evaluación es falsa, esa tupla será descartada.

```
FOR $b IN doc('informacion.xml')//película/director
WHERE $b/../../@fechaEstreno="19/12/2007"
RETURN <directores>$b</directores>
```

Recuperaría únicamente los directores que hubiesen estrenado película el día 19-12-2007 (suponemos que fechaEstreno es un atributo, por eso ponemos @ delante; el ../../ es para subir un nivel en el árbol XML donde está la información)

```
<directores>
  <director>Peter Jackson</director>
  <director>Alejandro Amenabar</director>
</directores>
```

ORDER

Indica, para un conjunto de información, un orden de salida. Por defecto organiza de manera ascendente según el tipo de datos que se esté evaluando.

```
FOR $b IN doc('informacion.xml')//película/director
ORDER BY $b/../../titulo
RETURN <directores>\$b</directores>
```

Recuperaría los directores ordenados por el título de sus películas:

```
<directores>
```

```
        <director>Alejandro Amenabar</director>
</directores>
<directores>
    <director>Ethan Coen</director>
</directores>
<directores>
    <director>Joel Coen</director>
</directores>
<directores>
    <director>Peter Jackson</director>
</directores>
```

RETURN

Formatea la información de salida. Esto nos ofrece muchas posibilidades, ya que podemos convertir la información recuperada a HTML, PDF o cualquier otro tipo. En los anteriores ejemplos hemos estado utilizando la cláusula **return**.