

### M04 - UF2 - NF3 (Bases de dades XML natives)

### Continguts:

1. XQuery - Introducció
2. Accedir a un XML i als seus nodes
3. FLWOR
4. Treballar contra BD XML generada a partir d'un fitxer XML
5. Expressions condicionals
6. Funcions definides per l'usuari

## 1. XQuery - Introducció

XQuery és un llenguatge específic per a consultar dades en documents XML, bases de dades natives XML i altres estructures similars. Té una certa similitud amb el llenguatge SQL ja que Xquery és un llenguatge de consulta de col·leccions de dades, però expressades en XML.

La base per a la consulta de dades en documents XML són les expressions Xpath i les FLOWR que tractarem al llarg d'aquest document.

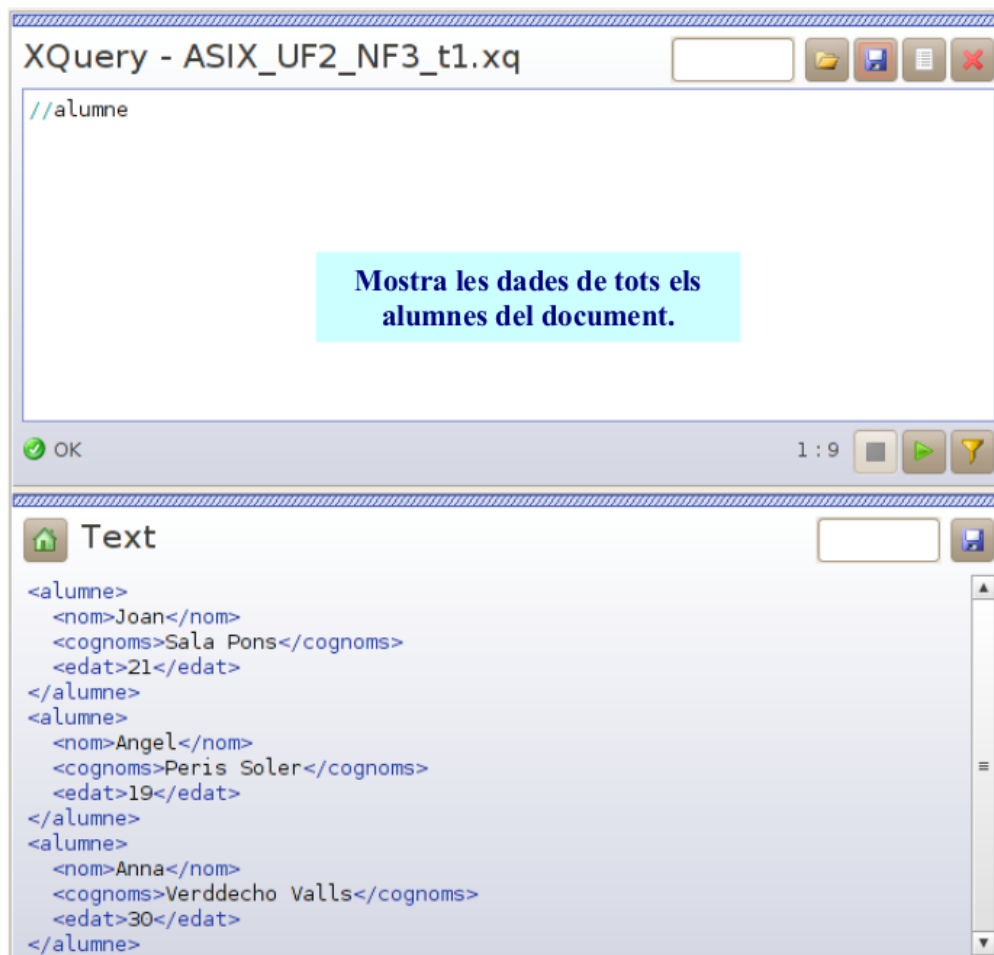
Per poder realitzar les consultes, necessitem un processador de consultes. Com sempre, podem trobar solucions de pagament i solucions lliures, i en el nostre cas ens hem decantat per una eina de codi obert, amb una interfície molt pràctica i senzilla. Aquesta eina s'anomena BaseX i la podem trobar en tres versions, per a Windows, per a MacOS i multiplataforma (Java).

Una mostra de la interfície d'aquesta eina la podeu veure a continuació:

[illegible]

En BaseX podem utilitzar expressions Xpath per cercar dins d'un document XML. Si tenim aquest document:

```
<?xml version="1.0" encoding="UTF-8" ?>
<institut>
  <alumnes>
    <alumne>
      <nom>Joan</nom>
      <cognoms>Sala Pons</cognoms>
      <edat>21</edat>
    </alumne>
    <alumne>
      <nom>Angel</nom>
      <cognoms>Peris Soler</cognoms>
      <edat>19</edat>
    </alumne>
    <alumne>
      <nom>Anna</nom>
      <cognoms>Verddecho Valls</cognoms>
      <edat>30</edat>
    </alumne>
  </alumnes>
</institut>
```



XQuery - ASIX\_UF2\_NF3\_t1B.xq

/institut/alumnes/alumne[edat>20]

**Mostra les dades de tots els alumnes majors de 20 anys.**

OK 1 : 25

**Text**

```
<alumne>
  <nom>Joan</nom>
  <cognoms>Sala Pons</cognoms>
  <edat>21</edat>
</alumne>
<alumne>
  <nom>Anna</nom>
  <cognoms>Verddecho Valls</cognoms>
  <edat>30</edat>
</alumne>
```

XQuery - ASIX\_UF2\_NF3\_t2.xq

/institut/alumnes/alumne/edat

**Mostra totes les edats.**

OK 1 : 30

**Text**

```
<edat>21</edat>
<edat>19</edat>
<edat>30</edat>
```

## 2. Accedir a un XML i als seus nodes

La funció **doc()** serveix per accedir a un XML.

```
doc("alumnes_del_institut.xml")
```

Per a navegar a través dels elements del XML seria:

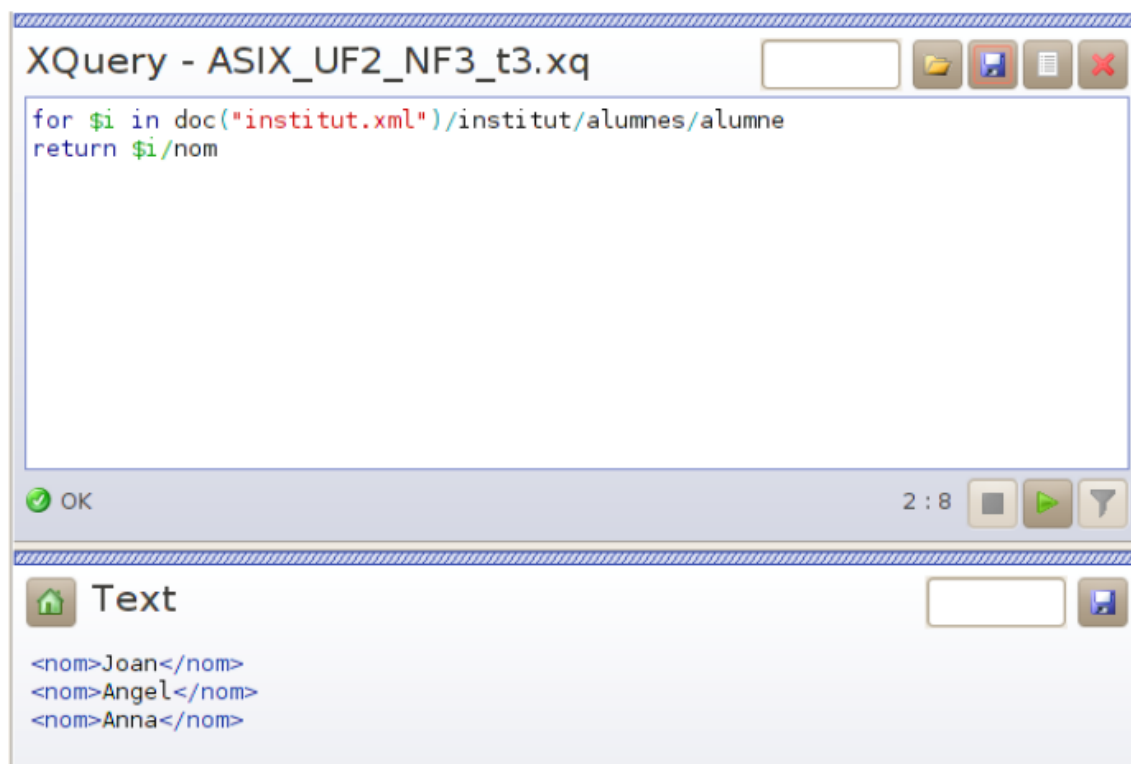
```
doc("alumnes_del_institut .xml")/institut/alumnes
```

**Predicats:** per a limitar l'extracció de dades:

```
doc("alumnes_del_institut .xml")/institut/alumnes[edat<18]
```

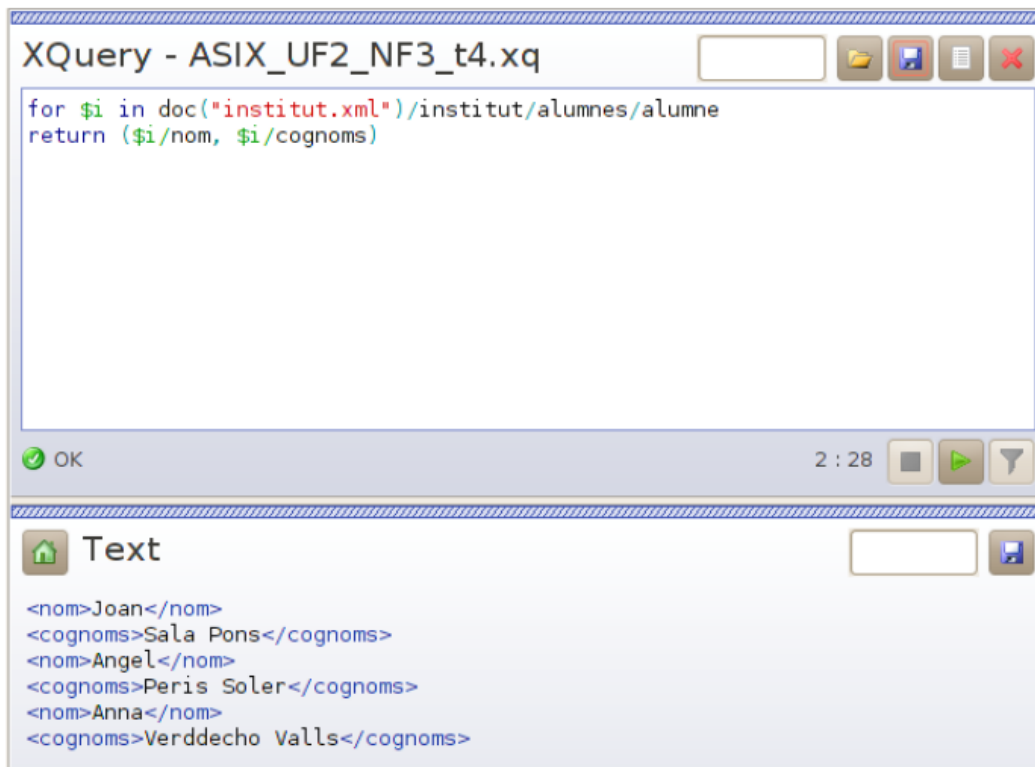
## 3. FLWOR

La denominació FLWOR és un acrònim de les funcions “**for**”, “**let**”, “**where**”, “**order by**” i “**return**” que utilitzem a XQuery. Aquestes expressions ens permeten crear consultes molt més elaborades com anirem veient.



Com veiem a l'exemple de dalt, la funció “**for**” assigna a la variable “**\$i**” les dades de l'alumne mentre hi hagin alumnes. Retornem el nom de cada alumne.

Al següent exemple, es demana el nom i cognoms per a cada alumne:



The screenshot shows the XQuery editor window titled "XQuery - ASIX\_UF2\_NF3\_t4.xq". The query code is as follows:

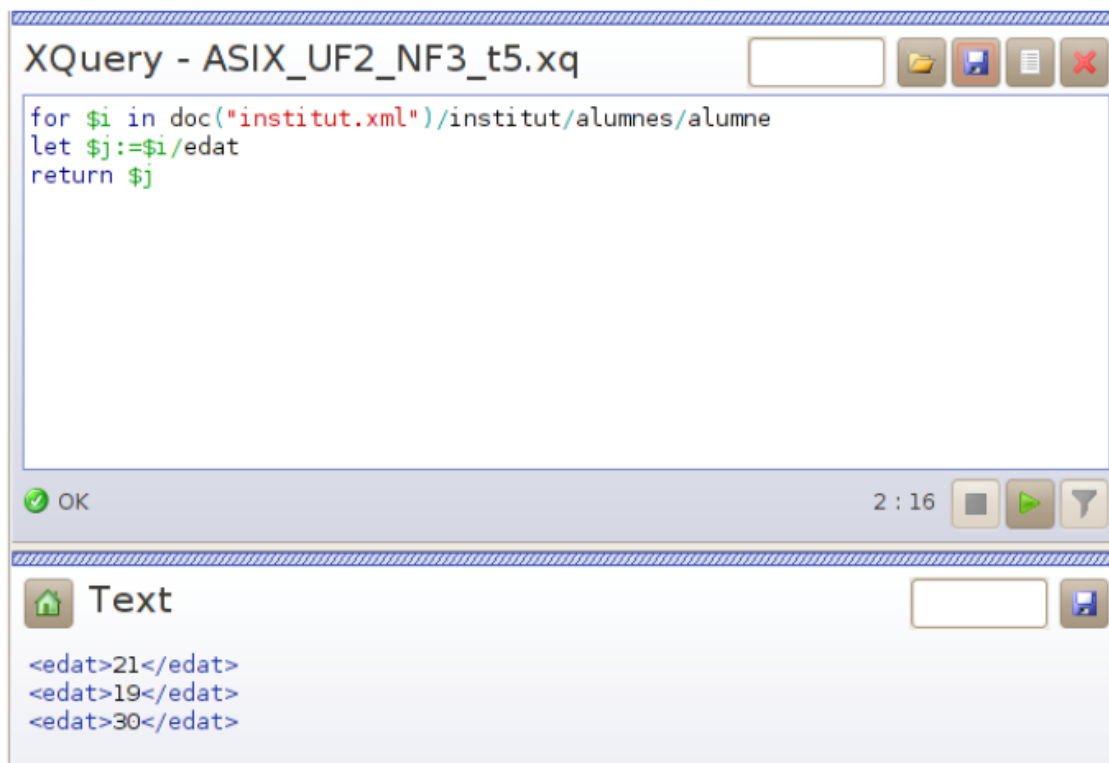
```
for $i in doc("institut.xml")/institut/alumnes/alumne
return ($i/nom, $i/cognoms)
```

The status bar at the bottom indicates "OK" and "2 : 28". Below the editor, the "Text" output pane displays the results in XML format:

```
<nom>Joan</nom>
<cognoms>Sala Pons</cognoms>
<nom>Angel</nom>
<cognoms>Peris Soler</cognoms>
<nom>Anna</nom>
<cognoms>Verddecho Valls</cognoms>
```

Adoneu-vos-en que per a mostrar les dues dades, ho fem separant-les amb una coma. Fixat també que per a mostrar cada dada, sempre utilitzem el precedent **\$i**.

El següent exemple, mostra les edats dels alumnes amb expressions FLWOR:



The screenshot shows the XQuery editor window titled "XQuery - ASIX\_UF2\_NF3\_t5.xq". The query code is as follows:

```
for $i in doc("institut.xml")/institut/alumnes/alumne
let $j:=$i/edat
return $j
```

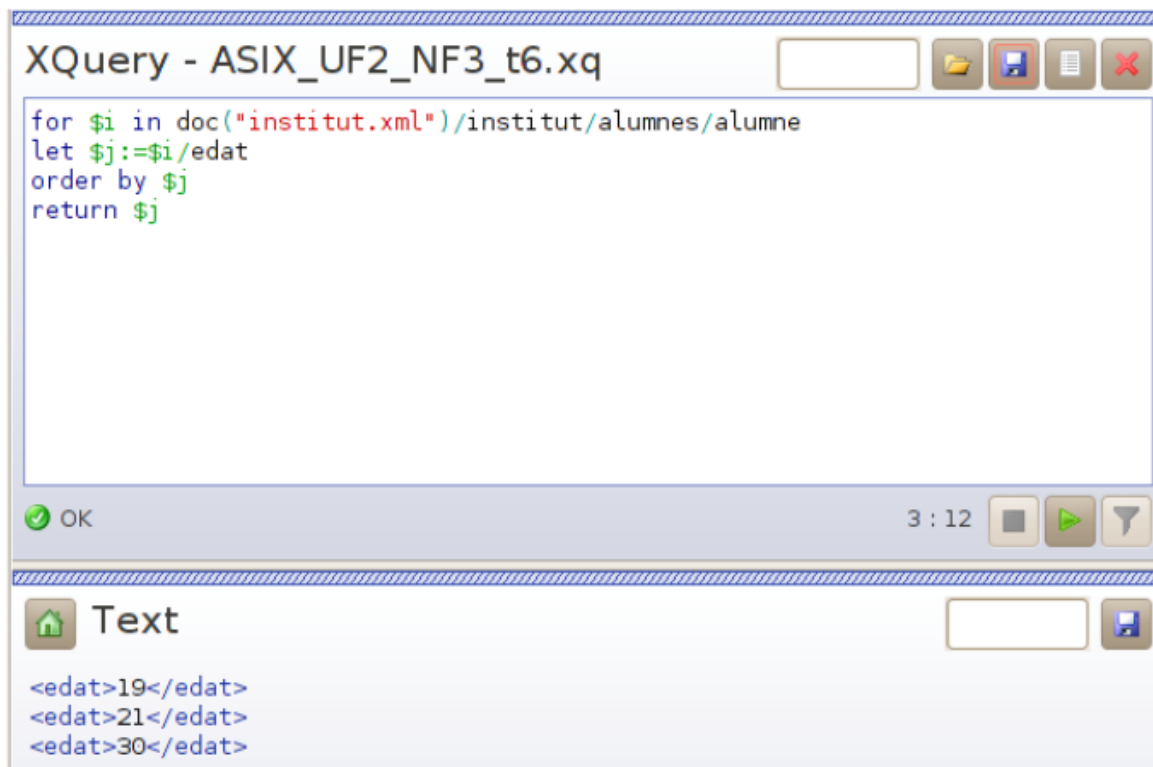
The status bar at the bottom indicates "OK" and "2 : 16". Below the editor, the "Text" output pane displays the results in XML format:

```
<edat>21</edat>
<edat>19</edat>
<edat>30</edat>
```

L'expressió **"let"** permet assignar l'edat de cada alumne (les dades de cada alumne les tenim en **"\$i"**) a la variable **"\$j"**.

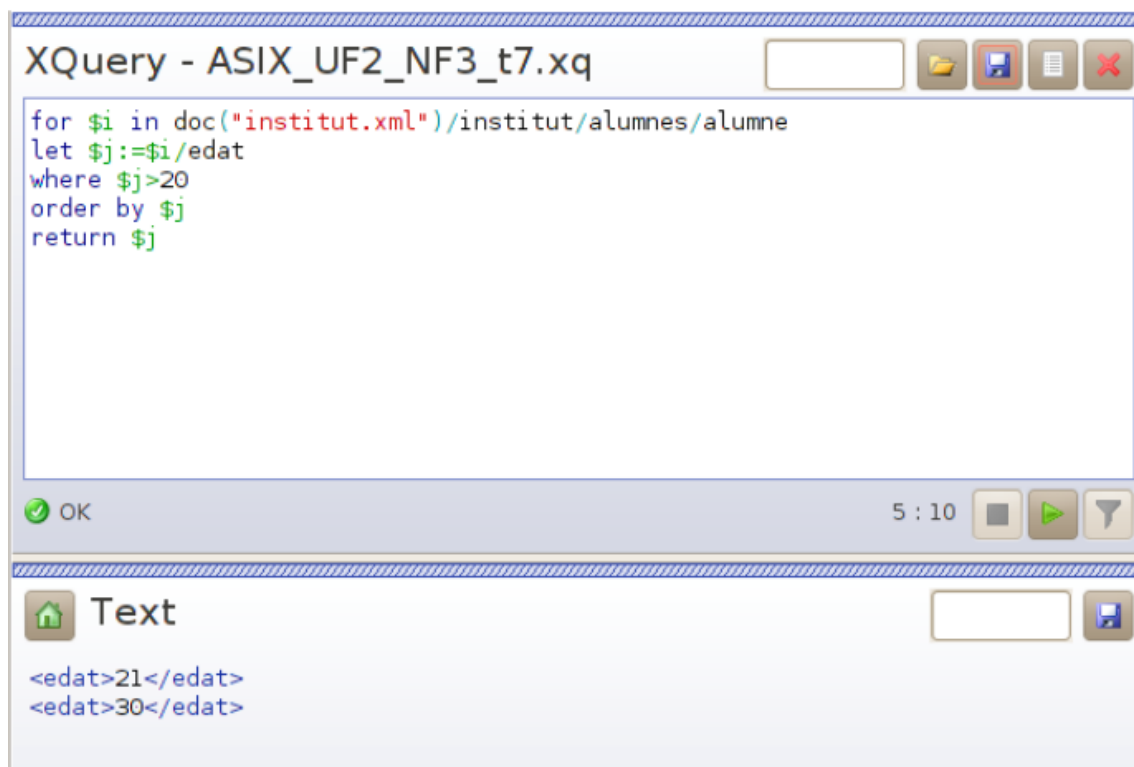
A continuació, mostra la variable **"\$j"** de cada alumne.

La següent imatge mostra les mateixes dades que la precedent, però, ordenades de menor a major edat. Si volguérem ordenar les edats de manera descendent, utilitzaríem l'opció "**descending**".



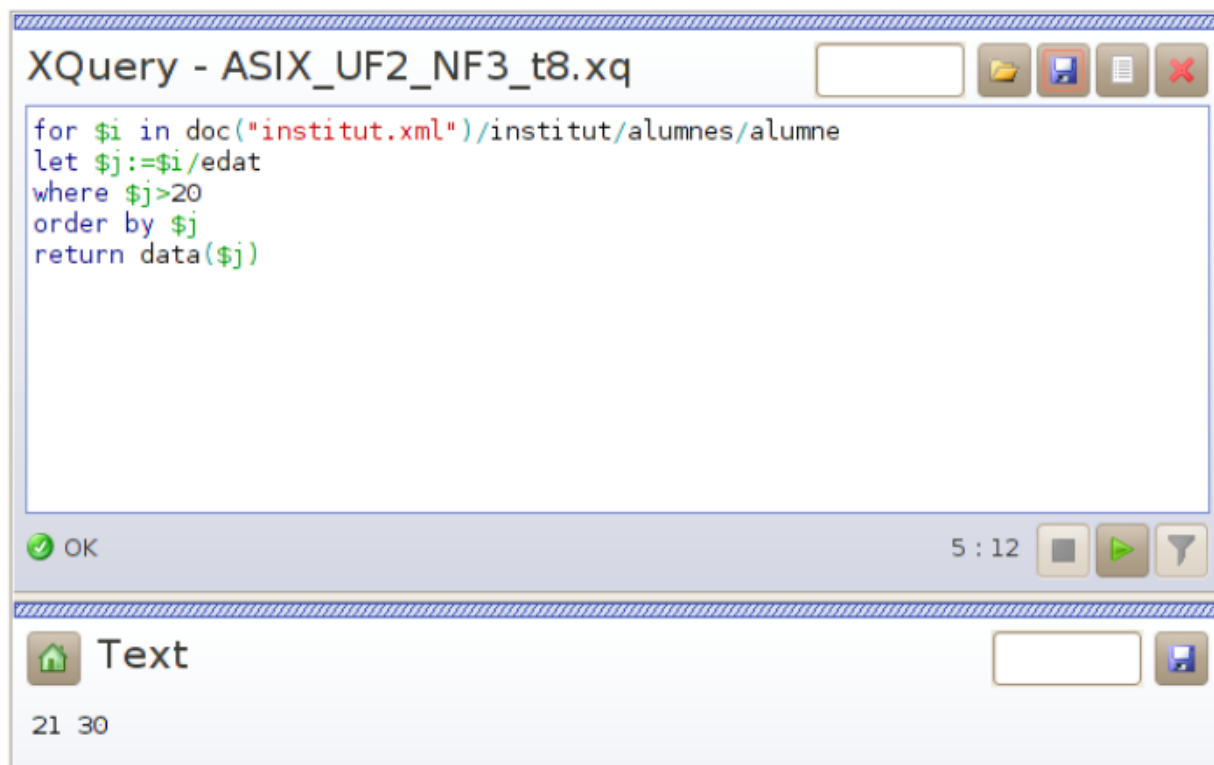
Podem afegir una restricció més, per exemple, mostrar les edats dels alumnes ordenades de manera ascendent d'aquells alumnes que superen els 20 anys, es a dir, de 21 anys o més.

A continuació es mostra la solució:



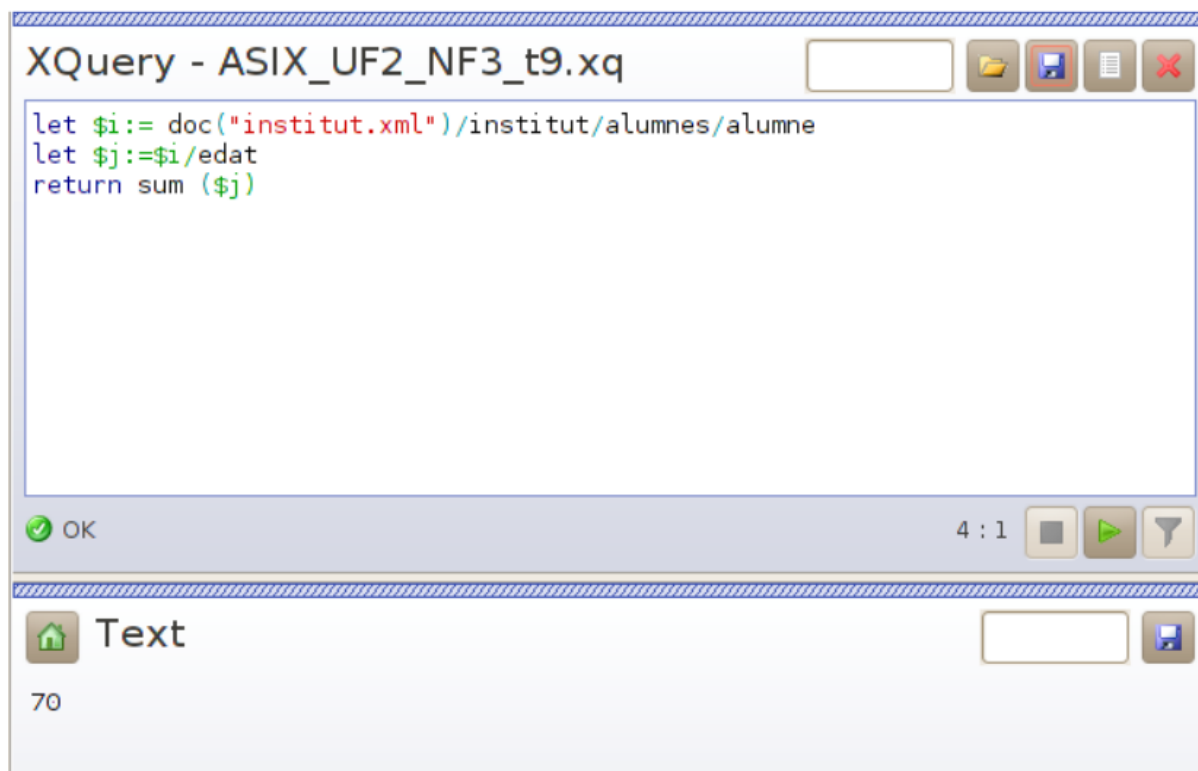
Si t'has fixat, en el resultat sempre apareixen les dades envoltades per l'etiqueta a la que pertanyen. Podem

“amagar” les etiquetes de manera que tans sols es mostren les dades utilitzant l'expressió **data()** com es mostra a continuació:



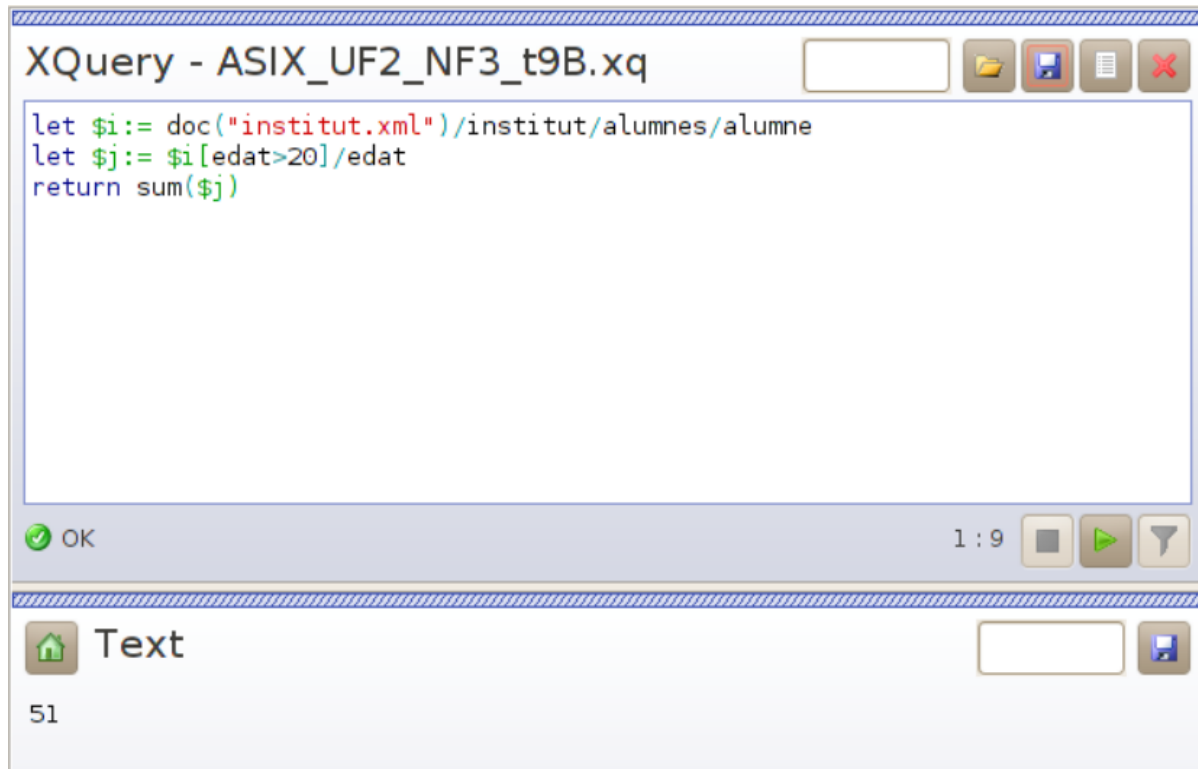
Als exemples següents, podem veure com utilitzar funcions predefinides que ens ajuden a estalviar feina, aquestes expressions, poden ser matemàtiques (**sum**, **count**, **avg**, ...), de manipulació de cadenes, etc, etc. Concretament, veurem de quina manera calculem la suma de totes les edats dels alumnes i també, la suma de les edats dels alumnes majors de 20 anys.

Tingueu en compte que per a poder sumar, no utilitzem el “for” al principi, el canviem per un “let”.



Mostra la suma de totes les edats dels alumnes del document. Fixeu-vos que hem substituït el “for” per un “let” i el “in” per una assignació (:=).

El mateix exemple però amb un filtre sobre el camp edat.



#### 4. Treballar contra BD XML generada a partir d'un fitxer XML

Per a no haver d'escriure el **in doc("ruta/fitxer.xml")** podem crear una BD a partir del fitxer XML.

Sobre el menú Database --> New --> amb el botó Browse seleccionem el XML que volem.

A partir d'ara hem de fer **in** en comptes del **:= doc("ruta/fitxer.xml")**

Exemple:

```
for $i in doc("cotxes.xml")/tenda/vehicles/vehicle
let $preu:=$i/preu
order by number($preu) descending
return data($preu)
```

passem a

```
for $i in tenda/vehicles/vehicle
let $preu:=$i/preu
order by number($preu) descending
return data($preu)
```

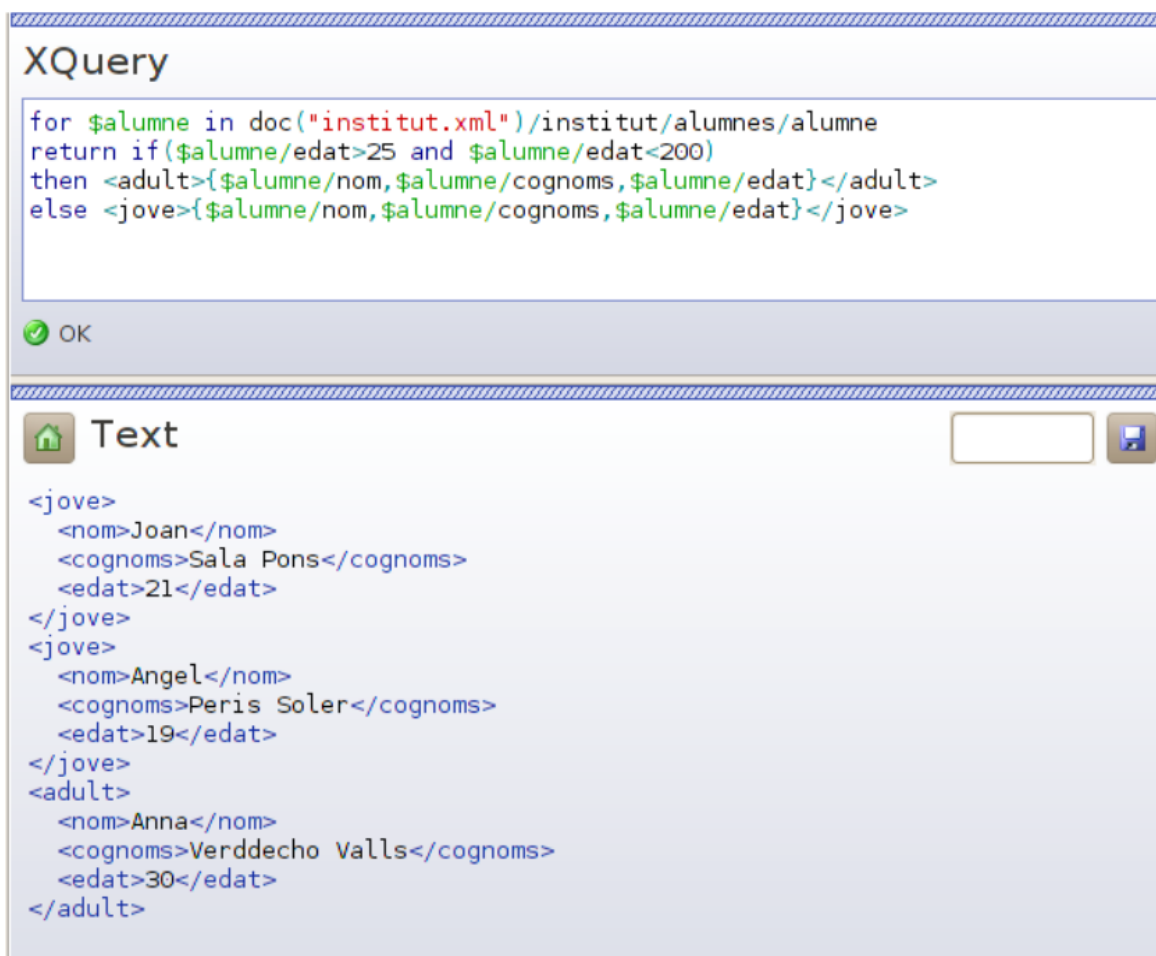


## 5. Expressions condicionals

També amb XQuery podem utilitzar les expressions condicionals per a obtenir els resultats que ens interessin de la base de dades XML (del document XML). La sintaxi és similar a la que s'utilitza en C, Java, etc.

A continuació veurem la utilització de les expressions condicionals a partir d'un exemple que mostra el nom, els cognoms i l'edat del alumnes, fent dos grups. El primer grup el formen els alumnes que tenen entre 26 i 199 anys, i el segon la resta d'alumnes (tenint en compte que no es coneix cap persona major de 199 anys).

Així, les dades del primer grup venen delimitades per les etiquetes <adult></adult> i el segon grup entre les etiquetes <jove></jove>.



S'ha de tenir en compte que per que es mostren correctament les dades de cada alumne, entre les etiquetes de <adult></adult> i <jove></jove> hem de posar { }.

## 5. Funcions definides per l'usuari

A més de les funcions predefinides, tenim la possibilitat de crear les nostres pròpies funcions en cas de que no trobem aquella que ens fa falta. Podeu veure un exemple de funció definida per l'usuari en la següent imatge que calcula el percentatge d'una quantitat, tant el percentatge com la quantitat són paràmetres que

es passen a la funció:

```
declare function local:percentatge($quantitat as xs:integer?, $percent as xs:integer?) as xs:float?
{
  $quantitat*$percent div 100
};

for $x in doc("cotxes.xml")/tenda/vehicles/vehicle
let $y:=$x/preu
where $x/any='2010'
return <cotxe>{$x/marca_model,<iva>{local:percentatge($y,16)}</iva>}</cotxe>
```

OK

Text

```
<cotxe>
  <marca_model>AUDI A3 / Spor. 1.4TFSI Ambiente</marca_model>
  <iva>3440</iva>
</cotxe>
<cotxe>
  <marca_model>VOLVO C30 / 2.0 D</marca_model>
  <iva>2848</iva>
</cotxe>
<cotxe>
  <marca_model>VOLKSWAGEN Golf / 1.6TDI CR Sport 105</marca_model>
  <iva>3024</iva>
</cotxe>
```

vehicle

Declaració d'un funció:

```
declare function local:percentatge($quantitat as xs:integer?, $percent as xs:integer?) as xs:float?
{
  $quantitat*$percent div 100
};
```

on:

- On el nom de la funció va sempre precedit per "local:"
- Els tipus de dades utilitzen el prefixe "XS:" seguit d'interrogant "?"
- La declaració de la funció acaba amb punt i coma

Crida a la funció dintre del codi XQuery:

```
for $x in doc("cotxes.xml")/tenda/vehicles/vehicle
let $y:=$x/preu
where $x/any='2010'
return <cotxe>{$x/marca_model,<iva>{local:percentatge($y,16)}</iva>}</cotxe>
```