

# TP3.1 XQuery

Ce TP sur XQuery est principalement basé sur l'utilisation d' **ExchangerXML** . Il n'existe pas encore beaucoup de solutions disponibles en logiciels libres pour utiliser un moteur d'interprétation XQuery. Toutefois si vous souhaitez en utiliser une autre, libre à vous (et, le cas échéant, merci de faire remonter les bonnes trouvailles à votre enseignant !?).

## 1. Prise en main de XQuery dans **Exchanger XML**

Pour vous familiariser avec XQuery, **Exchanger XML** propose un exemple permettant de mettre en oeuvre le moteur d'interprétation. Les fichiers composant l'exemple sont accessibles dans l'onglet "Projects" (habituellement affiché en zone gauche de votre écran), dossier XQuery.

**Attention** : Vous pouvez aussi aller chercher les fichiers dans le répertoire Projets dans "Program Files/Exchanger...".

**Au cas ou** : L'archive avec l'exemple XQuery.

### A faire :

1. Régénérez le résultat de la requête XQuery **books.xq** .

```
<books>
<book>
  <author> <AUTHOR>Stephen R. Davis</AUTHOR> </author>
  <title> <TITLE>Learn Java Now</TITLE> </title>
</book><book>
  <author> <AUTHOR>Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides</AUTHOR> </author>
  <title> <TITLE>Design Patterns</TITLE> </title>
</book><book>
  <author> <AUTHOR>Wardlaw, Lee</AUTHOR> </author>
  <title> <TITLE>Tales of Grandpa Cat</TITLE> </title>
</book><book>
  <author> <AUTHOR>Milne, A. A.</AUTHOR> </author>
  <title> <TITLE>When We Were Very Young</TITLE> </title>
</book><book>
  <author> <AUTHOR>Danzig</AUTHOR> </author>
  <title> <TITLE>Number, the Language of Science</TITLE> </title>
</book><book>
  <author> <AUTHOR>Danzig</AUTHOR> </author>
  <title> <TITLE>Language & the Science of Number</TITLE> </title>
</book><book>
  <author> <AUTHOR>Bonner</AUTHOR> </author>
  <title> <TITLE>Patterns of Crime in Animal Culture</TITLE> </title>
</book><book>
  <author> <AUTHOR>Bonner</AUTHOR> </author>
  <title> <TITLE>Evolution of Complexity in Animal Culture</TITLE> </title>
```

```
</book>
</books>
```

2. Ensuite définissez les requêtes permettant de :

1. Sélectionner les livres présents en stock par ordre de quantité croissante. Pour chaque livre, on affichera uniquement la *valeur* des éléments TITRE et QUANTITY dans des balises xhtml, le premier dans une police de couleur rouge, le second vert.

```
xquery version "1.0";
<books>
{
  for $book in $booklist//BOOKS/ITEM
  where xs:integer($book/QUANTITY) > 0
  order by xs:integer($book/QUANTITY)
  return
  <book>
    <p style="color: red">{$book/TITLE}</p>
    <p style="color: green">{$book/QUANTITY}</p>
  </book>
}
</books>
```

```
<books>
<book>
  <p style="color: red">When We Were Very Young</p>
  <p style="color: green">1</p>
</book><book>
  <p style="color: red">Evolution of Complexity in Animal Culture</p>
  <p style="color: green">2</p>
</book><book>
  <p style="color: red">Design Patterns</p>
  <p style="color: green">2</p>
</book><book>
  <p style="color: red">Number, the Language of Science</p>
  <p style="color: green">3</p>
</book><book>
  <p style="color: red">Tales of Grandpa Cat</p>
  <p style="color: green">5</p>
</book><book>
  <p style="color: red">Language & the Science of Number</p>
  <p style="color: green">5</p>
</book><book>
  <p style="color: red">Learn Java Now</p>
  <p style="color: green">12</p>
</book>
</books>
```

2. Faire la facture pour l'achat de deux livres particuliers (à choisir). Votre requête

devra définir au minimum les variables permettant d'accéder aux livres, afficher leur titre et le prix total (les prix doivent être récupérés dans `books.xml`). **Attention** : les variables dans XQuery ne sont pas modifiables (propriétés d'un *langage fonctionnel*). Il faut donc traiter les 2 livres en *une seule expression* FLOWER (par exemple avec 2 variables), et réaliser la somme des prix avec un let interne unique (pas d'accumulateur).

```
xquery version "1.0";
<invoice>
{
  let $book1 := //BOOKS/ITEM[TITLE = 'Learn Java Now']
  let $book2 := //BOOKS/ITEM[TITLE = 'Design Patterns']
  let $totalPrice := xs:decimal($book1/PRICE) + xs:decimal($book2/PRICE)
  return
  <purchase>
    <book>
      <author>{data($book1/AUTHOR)}</author>
      <title>{data($book1/TITLE)}</title>
      <price>{data($book1/PRICE)}</price>
    </book>
    <book>
      <author>{data($book2/AUTHOR)}</author>
      <title>{data($book2/TITLE)}</title>
      <price>{data($book2/PRICE)}</price>
    </book>
    <totalPrice>{$totalPrice}</totalPrice>
  </purchase>
}
</invoice>
```

```
<invoice>
<purchase>
  <book>
    <author>Stephen R. Davis</author>
    <title>Learn Java Now</title>
    <price>9.95</price>
  </book>
  <book>
    <author>Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides</author>
    <title>Design Patterns</title>
    <price>49.95</price>
  </book>
  <totalPrice>59.9</totalPrice>
</purchase>
</invoice>
```

3. Relever pour chaque catégorie de livre la somme des prix des livres appartenant à cette catégorie. La catégorie devra être indiquée explicitement.

```
xquery version "1.0";
<category_prices>
{
  for $category in distinct-values(//BOOKS/ITEM/@CAT)
  let $books := //BOOKS/ITEM[@CAT = $category]
  let $totalPrice := sum($books/PRICE)
  return
  <category>
    <name>{data(//CATEGORIES/CATEGORY[@CODE = $category]/@DESC)}</name>
    <totalPrice>{data($totalPrice)}</totalPrice>
  </category>
}
</category_prices>
```

```
<category_prices>
<category>
  <name>Science</name>
  <totalPrice>20.849999999999998</totalPrice>
</category><category>
  <name>Fiction</name>
  <totalPrice>19.08</totalPrice>
</category><category>
  <name>Crime</name>
  <totalPrice>15.95</totalPrice>
</category><category>
  <name>Computing</name>
  <totalPrice>9.95</totalPrice>
</category><category>
  <name/>
  <totalPrice>49.95</totalPrice>
</category>
</category_prices>
```

## 2. Les livres et les auteurs

L'objectif est d'interroger les documents [livres.xml](#) et [auteurs.xml](#) avec XQuery. Pour faire des requêtes XQuery nous utiliserons **Exchanger XML**. Vous remarquerez que certaines questions ont déjà été posées dans l'exercice 1 du TP sur XPath, vous pouvez ainsi comparer les résultats obtenus entre les requêtes XPath et XQuery.

### A faire :

Avec **Exchanger XML**, faites les recherches suivantes sur les fichiers [livres.xml](#) et [auteurs.xml](#) :

#### 1. Pour s'échauffer, sans les fichiers : 5+7 ?

xQuery:

5+7

Résultat:

12

## 2. Quelle est le genre du livre « *Astérix et Cléopatre* » ?

xQuery:

```
xquery version "1.0";  
//LIVRE[@titre = "Astérix et Cléopatre"]/GENRE
```

Résultat:

BD

## 3. Quels sont les titres (*affichés en attribut d'une balise <livre>*) et les auteurs (*affichés en élément de <livre>*) des romans en anglais ?

xQuery:

```
xquery version "1.0";  
for $livre in //LIVRE[LANG = 'EN' and GENRE = 'Roman']  
return  
  <result>  
    <titre>{$livre/@titre}</titre>  
    <auteur>{data($livre/AUTEUR)}</auteur>  
  </result>
```

Résultat:

```
<result>  
  <titre titre="The Alchemist"/>  
  <auteur>Coelho</auteur>  
</result>
```

## 4. Quels sont les titres de toutes les BD ? Les titres ne devront plus apparaître sous forme d'attribut mais en valeur textuelle d'une balise <bédé> directement.

xQuery:

```
xquery version "1.0";
<BDs>
{
  for $bd in //LIVRE[GENRE = 'BD']
  return
    <bédé>{data($bd/@titre)}</bédé>
}
</BDs>
```

Résultat:

```
<BDs>
  <bédé>Tintin au Congo</bédé>
  <bédé>Astérix et Cléopatre</bédé>
  <bédé>Simpsons Comics</bédé>
</BDs>
```

## 5. Quel est le livre qui précède immédiatement « *La première nuit* » dans le document ?

xQuery:

```
xquery version "1.0";
let $targetTitle := "La première nuit"
let $precedingBook := //LIVRE[@titre = $targetTitle]/preceding-sibling::LIVRE[1]
return data($precedingBook/@titre)
```

Résultat:

```
Simpsons Comics
```

## 6. Afficher les détails des livre écrits par Freud.

xQuery:

```
xquery version "1.0";
for $livre in //LIVRE[AUTEUR = 'Freud']
return
  <livre>
    <titre>{$livre/@titre}</titre>
    <genre>{$livre/GENRE}</genre>
    <lang>{$livre/LANG}</lang>
  </livre>
```

Résultat:

```

<livre>
  <titre titre="Psychologie de la vie quotidienne"/>
  <genre><GENRE>Psychologie</GENRE></genre>
  <lang><LANG>FR</LANG></lang>
</livre>
<livre>
  <titre titre="The Ego and the ID"/>
  <genre><GENRE>Psychologie</GENRE></genre>
  <lang><LANG>EN</LANG></lang>
</livre>

```

**7. Quels sont les genres de livres qui ont plusieurs exemplaires présents dans la base ? Afficher pour chaque genre le nombre de livres correspondant.**

xQuery:

```

xquery version "1.0";
let $genres := distinct-values(//LIVRE/GENRE)
for $genre in $genres
let $nombre := count(//LIVRE[GENRE = $genre])
where $nombre > 1
return
  <genre>
    <nom>{$genre}</nom>
    <nombre>{$nombre}</nombre>
  </genre>

```

Résultat:

```

<genre>
  <nom>BD</nom>
  <nombre>3</nombre>
</genre>
<genre>
  <nom>Roman</nom>
  <nombre>4</nombre>
</genre>
<genre>
  <nom>Psychologie</nom>
  <nombre>3</nombre>
</genre>

```

**8. Combien de livres sont proposés dans chaque langue ?**

xQuery:

```

xquery version "1.0";
for $langue in distinct-values(//LIVRE/LANG)
let $nombre := count(//LIVRE[LANG = $langue])
return

```

```

<language>
  <nom>{$langue}</nom>
  <nombre>{$nombre}</nombre>
</language>

```

Résultat:

```

<language>
  <nom>FR</nom>
  <nombre>9</nombre>
</language>
<language>
  <nom>EN</nom>
  <nombre>6</nombre>
</language>

```

## 9. Afficher la liste des auteurs, avec pour chacun d'eux la liste de leurs livres.

xQuery:

```

xquery version "1.0";
for $auteur in distinct-values(//LIVRE/AUTEUR)
return
  <auteur>
    <nom>{$auteur}</nom>
    <livres>
      {
        for $livre in //LIVRE[AUTEUR = $auteur]
        return
          <livre>{$livre/@titre}</livre>
      }
    </livres>
  </auteur>

```

Résultat:

```

<auteur>
  <nom>Herge</nom>
  <livres>
    <livre titre="Tintin au Congo"/>
  </livres>
</auteur>
<auteur>
  <nom>Dickens</nom>
  <livres>
    <livre titre="A tale of two cities"/>
  </livres>
</auteur>
<auteur>
  <nom>Coelho</nom>
  <livres>

```



```

        <livre titre="The Alchemist"/>
        <livre titre="Veronika décide de mourir"/>
    </livres>
</auteur>
<auteur>
    <nom>Hugo</nom>
    <livres>
        <livre titre="Les Misérables"/>
    </livres>
</auteur>
<auteur>
    <nom>Goscinny</nom>
    <livres>
        <livre titre="Astérix et Cléopâtre"/>
    </livres>
</auteur>
<auteur>
    <nom>Groening</nom>
    <livres>
        <livre titre="Simpsons Comics"/>
    </livres>
</auteur>
<auteur>
    <nom>Levy</nom>
    <livres>
        <livre titre="La première nuit"/>
    </livres>
</auteur>
<auteur>
    <nom>Freud</nom>
    <livres>
        <livre titre="Psychologie de la vie quotidienne"/>
        <livre titre="The Ego and the ID"/>
    </livres>
</auteur>
<auteur>
    <nom>Gregorie</nom>
    <livres>
        <livre titre="Professional C++"/>
    </livres>
</auteur>
<auteur>
    <nom>Brewer</nom>
    <livres>
        <livre titre="le corps humain"/>
    </livres>
</auteur>
<auteur>
    <nom>Viktor</nom>
    <livres>
        <livre titre="Men search for meaning"/>
    </livres>
</auteur>
<auteur>
    <nom>Scmitt</nom>
    <livres>
        <livre titre="La femme au miroir"/>
    </livres>
</auteur>

```

```
<auteur>
  <nom>Christie</nom>
  <livres>
    <livre titre="Mort sur le Nil"/>
  </livres>
</auteur>
```

10. **(facultatif) Pour finir, vous pouvez reprendre les requêtes XPath du TP correspondant et vérifiez que vous obtenez bien les mêmes résultats...**