

## LOW HOMEWORK

### 1-Résumé sur:

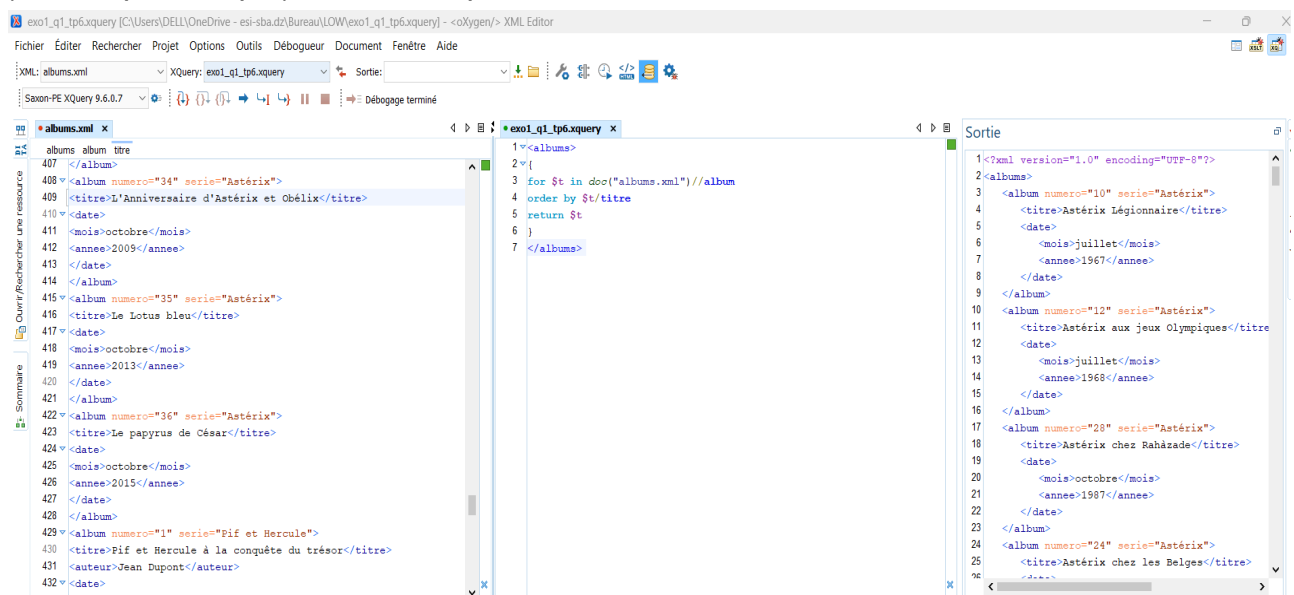
- **template:** Définit un bloc de transformation associé à un nœud ou un motif de correspondance via l'attribut match.
- **apply-template:** Appelle d'autres templates qui sont déclarés dans le même fichier après template, basés sur les nœuds enfants. Permet de parcourir récursivement un document XML. Peut utiliser un attribut select pour cibler certains éléments.
- **call-template:** Appelle un template nommé avec l'attribut nom, elle a les mêmes propriétés qu'une fonction (appel, paramètres, réutilisable ..).

### 2-Sol de TP6:

exo1:

q1:

**explication:** le résultat dans albums, for pour afficher tout les album et order par titre (ordre alphabétique) ascendant par défaut, il retourne toute la balise album.



q2:

**explication:** pour chaque (for) album il va vérifier l'année si elle est > 1997 il retourne la balise album.

The screenshot shows the XQuery IDE with the following components:

- XML Source:** `albums.xml`
- Query:** `exo1_q1_tp6.xquery`
- Sortie:** `exo1_q2_tp6.xquery`
- Query Text:**

```

1 <albums>
2 {
3   for $t in doc("albums.xml")//album[date/annee > "1997"]
4   return $t
5 }
6 </albums>
7

```
- Sortie Text:**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <albums>
3   <album numero="31" serie="Asterix">
4     <titre>Asterix et Latraviata</titre>
5     <date>
6       <mois>mars</mois>
7       <annee>2001</annee>
8     </date>
9   </album>
10   <album numero="33" serie="Asterix">
11     <titre>Le ciel lui tombe sur la tête</titre>
12     <auteur>Jean Dupont</auteur>
13     <date>
14       <mois>octobre</mois>
15       <annee>2005</annee>
16     </date>
17   </album>
18   <album numero="34" serie="Asterix">
19     <titre>L'Anniversaire d'Asterix et Obelix</titre>
20     <date>
21       <mois>octobre</mois>
22       <annee>2009</annee>
23     </date>
24   </album>
25   <album numero="35" serie="Asterix">

```

q3:

**explication:** pour chaque auteur (for) il count le nombre des album avec le même auteur dans une variable \$x si elle est >1 donc il affiche cet auteur et groupe by pour éviter la repetition.

The screenshot shows the XQuery IDE with the following components:

- XML Source:** `albums.xml`
- Query:** `exo1_q2_tp6.xquery`
- Sortie:** `exo1_q3_tp6.xquery`
- Query Text:**

```

1 <albums>
2 {
3   for $t in distinct-values(doc("albums.xml")//album/auteur)
4   let $x := count(doc("albums.xml")//album[auteur = $t])
5   where $x > 1
6   order by $t
7   return
8     <auteur nb-albums="{ $x }">{ $t }</auteur>
9 }
10 </albums>
11

```
- Sortie Text:**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <albums>
3   <auteur nb-albums="5">Goscinny</auteur>
4   <auteur nb-albums="2">Gerard Dubois</auteur>
5   <auteur nb-albums="3">Hergé</auteur>
6   <auteur nb-albums="2">Jean Dupont</auteur>
7   <auteur nb-albums="2">Marie Leferve</auteur>
8   <auteur nb-albums="2">Michel Bernard</auteur>
9   <auteur nb-albums="2">Sophie Marceau</auteur>
10  <auteur nb-albums="4">Uderzo</auteur>
11 </albums>
12

```

q4:

**explication:** pour chaque serie il prend tout albums avec même serie dans \$albums et le max de l'annee dans \$maxYear il parcourt albums et cherche l'album avec recent annee et le retourne.

The screenshot shows the XQuery IDE with the following components:

- XML Source:** `albums.xml`
- Query:** `exo1_q1_tp6.xquery`
- Sortie:** `exo1_q2_tp6.xquery`
- Query Text:**

```

1 <albumrecent>
2 {
3   for $t in distinct-values(doc("albums.xml")//album/$serie)
4   let $albums := doc("albums.xml")//album[$serie = $t]
5   let $maxYear := max($albums/date/annee/x:integer())
6   for $a in $albums
7   where x:integer($a/date/annee) = $maxYear
8   return
9     <album serie="{ $t }">
10       <$a/titre>
11       <annee>{ $maxYear }</annee>
12     </album>
13 }
14 </albumrecent>
15

```
- Sortie Text:**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <albumrecent>
3   <album serie="Tintin">
4     <titre>Tintin et l'Alph-Art</titre>
5     <annee>1964</annee>
6   </album>
7   <album serie="Asterix">
8     <titre>Le papyrus de César</titre>
9     <annee>2015</annee>
10  </album>
11  <album serie="Pif et Hercule">
12    <titre>Pif et Hercule et la quête des éléments</titre>
13    <annee>1964</annee>
14  </album>
15 </albumrecent>
16

```

q5:

explication: \$albums contient tout les albums de la serie \$t donc j'affiche la serie et ces album avec un for et le nombre d'album est dans \$nb

```
1<albums>
2<for $t in distinct-values(doc("albums.xml")//album/@serie)
3  let $albums := doc("albums.xml")//album[@serie = $t]
4  let $nb := count($albums)
5  return <serie nom="{ $t }" nombre-album="{ $nb }" >
6  {
7    for $a in $albums
8    return $a
9  }
10</serie>
11</albums>
```

q6:

explication: j'ai utiliser le resultat de q5 dans un let variable series et compter le nb max après j'ai afficher cette serie du serie avec where nb est max

```
1<albums>
2<let $series:=for $t in distinct-values(doc("albums.xml")//album/@serie)
3  let $nb := count($albums)
4  return <serie nom="{ $t }" nombre-album="{ $nb }" >
5  {
6    for $a in $albums
7    return $a
8  }
9</series>
10let $maxalbums := max($series/@nombre-album xs:integer())
11for $serie in $series
12where xs:integer($serie/@nombre-album) = $maxalbums
13return $serie
14</albums>
```

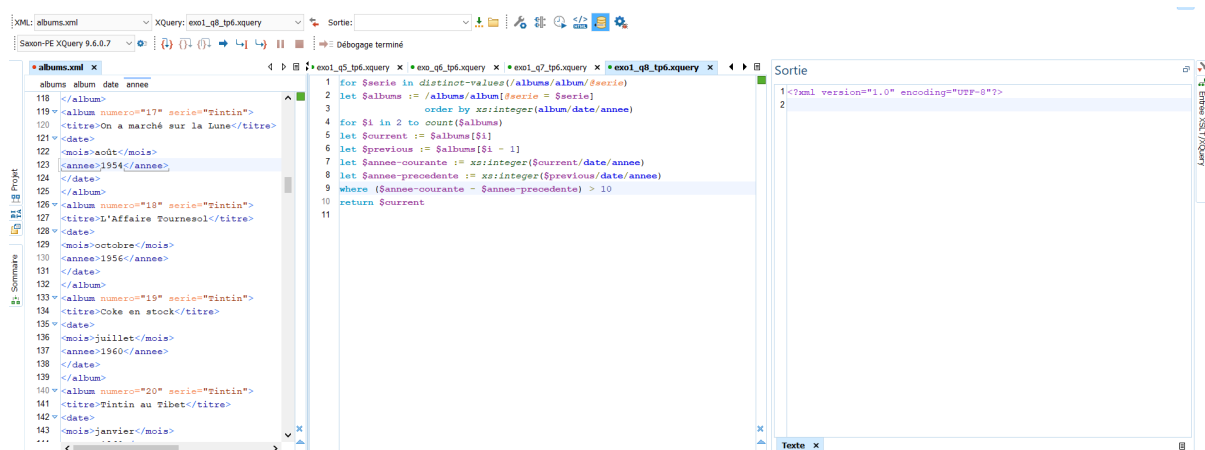
q7:

explication: j'ai grouper dans albums par annee apres prendre l'annee avec le nombre d'album tout ça dans la variable \$annees apres j'ai qualculer le max de nb album et afficher l'annee qui correspond à ce max.

```
1<albums>
2<let $annees:=for $t in distinct-values(doc("albums.xml")//album/@date/annee)
3  let $albums := doc("albums.xml")//album[ date=$t ]
4  let $nb := count($albums)
5  return <annee nombre-album="{ $nb }" >
6  { $t }
7</annees>
8let $maxalbums := max($annees/@nombre-album xs:integer())
9for $annee in $annees
10where xs:integer($annee/@nombre-album) = $maxalbums
11return $annee
12</albums>
```

q8:

**explication:** j'ai trier dans serie par annee pour comparer dans chaque serie album i avec i-1 debutant de 2 (2-1=1) et apres prendre l'annee de i- l'annee de i-1 si elle est plus de 10 j'affiche le current album. la sortie est vide car il y a pas des album en écart plus de 10.



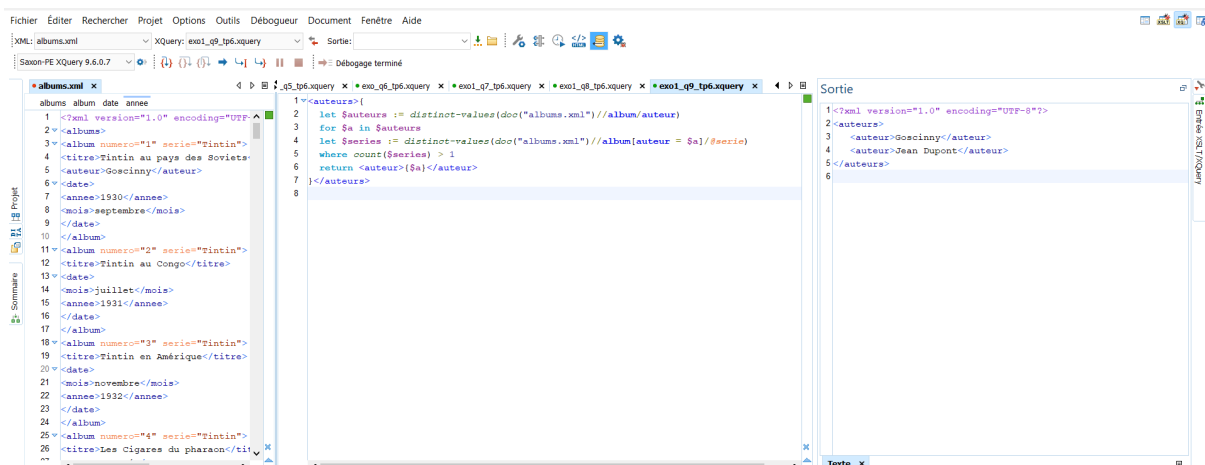
The screenshot shows the XQuery IDE with a file named 'albums.xml' and a query named 'exo1\_q8\_tp6.xquery'. The XML file contains a list of albums with attributes 'numero', 'serie', 'date', and 'annee'. The query is as follows:

```
1 for $serie in distinct-values(/albums/album/$serie)
2 let $albums := /albums/album[$serie = $serie]
3 order by xs:integer(album/date/annee)
4 for $i in 2 to count($albums)
5 let $current := $albums[$i]
6 let $previous := $albums[$i - 1]
7 let $annee-courante := xs:integer($current/date/annee)
8 let $annee-precedente := xs:integer($previous/date/annee)
9 where ($annee-courante - $annee-precedente) > 10
10 return $current
11
```

The output pane, titled 'Sortie', is empty, showing only the XML declaration: `1<?xml version="1.0" encoding="UTF-8"?>`.

q9:

**explication:** pour chaque auteur je compte le nombre de serie si il est >1 je retourne l'auteur



The screenshot shows the XQuery IDE with a file named 'albums.xml' and a query named 'exo1\_q9\_tp6.xquery'. The XML file contains a list of albums with attributes 'numero', 'serie', 'date', 'annee', and 'auteur'. The query is as follows:

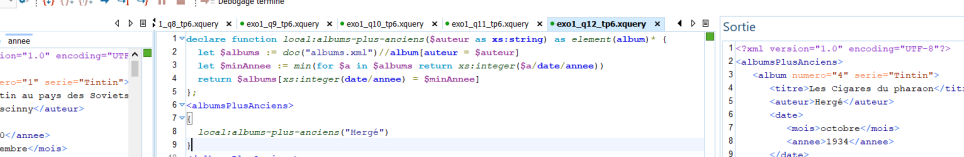
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <auteurs>
3   for $a in $auteurs
4   let $series := distinct-values(doc("albums.xml")//album[auteur = $a]/$serie)
5   where count($series) > 1
6   return <auteur>{$a}</auteur>
7 </auteurs>
8
```

The output pane, titled 'Sortie', shows the following XML output:

```
1<?xml version="1.0" encoding="UTF-8"?>
2<auteurs>
3  <auteur>Goscinny</auteur>
4  <auteur>Jean Dupont</auteur>
5</auteurs>
6
```

q10:

**explication:** récupérer les auteurs avec répétition est count qui répète le max



The screenshot shows the XQuery IDE with the 'albums.xml' file open. The file contains an XQuery script that declares a local function 'albumsPlusAnciens' and uses it to process the 'albums.xml' file. The 'Sortie' window shows the output of the XQuery, which is a list of albums with their titles, authors, and dates.

**albums.xml**

```

1 albums album date annee
2
3 <?xml version="1.0" encoding="UTF-8"?>
4 <albums>
5   <album numero="1" serie="Tintin">
6     <titre>Tintin au pays des Soviets
7     <auteur>Goscinnny</auteur>
8     <date>
9       <annee>1930</annee>
10      <mois>septembre</mois>
11    </date>
12  </album>
13  <album numero="2" serie="Tintin">
14    <titre>Tintin au Congo</titre>
15    <date>
16      <mois>juillet</mois>
17      <annee>1931</annee>
18    </date>
19  </album>
20  <album numero="3" serie="Tintin">
21    <titre>Tintin en Amérique</titre>
22    <date>
23      <mois>novembre</mois>
24      <annee>1932</annee>
25    </date>
26  </album>
27  <album numero="4" serie="Tintin">
28    <titre>Les Cigares du pharaon</ti

```

**Sortie**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <albumsPlusAnciens>
3   <album numero="4" serie="Tintin">
4     <titre>Les Cigares du pharaon</titre>
5     <auteur>Bergé</auteur>
6     <date>
7       <mois>octobre</mois>
8       <annee>1934</annee>
9     </date>
10  </album>
11 </albumsPlusAnciens>

```

### q13:

**explication:** sélectionner album num 1 serie tintin est creer element auteur avec element.

The screenshot shows the XQuery IDE with the following query in the editor:

```
1 let $premier := doc("albums.xml")//album[@numero = "1" and $serie = "Tintin"]
2 return
3 <album numero="1" serie="Tintin">
4   ($premier/titre)
5   ($premier/auteur)
6   ($premier/date)
7   <element (name($premier/*[2])) { "Hergé" }
8 </album>
```

The result pane shows the output XML:

```
1<?xml version="1.0" encoding="UTF-8"?>
2<album numero="1" serie="Tintin">
3  <titre>Tintin au pays des Soviets</titre>
4  <auteur>Goscinny</auteur>
5  <date>
6    <mois>1930</mois>
7    <mois>septembre</mois>
8  </date>
9  <auteur>Hergé</auteur>
10 </album>
```

### q14:

**explication:** sélectionner album num 3 serie Astérix et creer attribut editeur avec attribut.

The screenshot shows the XQuery IDE with the following query in the editor:

```
1 let $premier := doc("albums.xml")//album[@numero = "3" and $serie = "Astérix"]
2 return
3 <album numero="3" serie="Astérix">
4   <attribut { "editeur" } { "La plume" }
5   ($premier/titre)
6   ($premier/auteur)
7   ($premier/date)
8 </album>
```

The result pane shows the output XML:

```
1<?xml version="1.0" encoding="UTF-8"?>
2<album numero="3" serie="Astérix" editeur="La plume">
3  <titre>Astérix et les Goths</titre>
4  <date>
5    <mois>juillet</mois>
6    <annee>1963</annee>
7  </date>
8 </album>
```

### q15:

**explication:** sélectionner albums serie tintin sans auteur ou qui contient pas Hergé et creer element auteur Hergé.

The screenshot shows the XQuery IDE with the following query in the editor:

```
1 for $premier in doc("albums.xml")//album[$serie = "Tintin" and (not($auteur) or normalize-space($auteur) != "Hergé")]
2 return
3 <album numero="{ $premier/@numero }" serie="Tintin">
4   ($premier/titre)
5   ($premier/auteur)
6   ($premier/date)
7   <element (name($premier/*[2])) { "Hergé" }
8 </album>
```

The result pane shows the output XML:

```
4 <auteur>Goscinny</auteur>
5 <date>
6   <annee>1930</annee>
7   <mois>septembre</mois>
8 </date>
9 <auteur>Hergé</auteur>
10 </album>
11 <album numero="2" serie="Tintin">
12   <titre>Tintin au Congo</titre>
13   <date>
14     <mois>juillet</mois>
15     <annee>1931</annee>
16   </date>
17   <date>Hergé</date>
18 </album>
19 <album numero="3" serie="Tintin">
20   <titre>Tintin en Amérique</titre>
21   <date>
22     <mois>novembre</mois>
23     <annee>1932</annee>
24   </date>
25   <date>Hergé</date>
26 </album>
27 <album numero="5" serie="Tintin">
28   <titre>Le Lotus bleu</titre>
```

q16:

explication: sélectionner albums serie Astérix et remplacer par Astérix et Obélix.

The screenshot shows the XQuery editor with the following query:

```
1 for $premier in doc("albums.xml")//album[$serie = "Astérix"]
2 return
3 <album numero="{ $premier/numero }">
4   [attribute ("serie") { "Astérix et Obélix" }]
5   ($premier/titre)
6   ($premier/auteur)
7   ($premier/date)
8 </album>
```

The result pane shows the transformed XML output, where the series name has been updated to "Astérix et Obélix" for the first album.

q17:

explication: retourne supprime serie Tintin publier avant 1950

The screenshot shows the XQuery editor with the following query:

```
1 for $album in doc("albums.xml")//album[($serie = "Tintin" and date/annee > 1950) or $serie="Tintin"]
2 return
3 $premier
```

The result pane shows the XML output, which includes albums from the series "Tintin" published after 1950, as the query filters out those published before 1950.

q18:

explication: augmenter par 1 l'année de publication de tous les albums de la série "Astérix" après 1980.

The screenshot shows the XQuery editor with the following query:

```
1 for $album in doc("albums.xml")//album[$serie = "Astérix" and xs:integer(date/annee) > 1980]
2 return
3 <album numero="{ $album/numero }" serie="{ $album/serie }">
4   ($album/titre)
5   ($album/auteur)
6   <date>
7     { $album/date/annee + 1 }
8   </date>
9 </album>
```

The result pane shows the transformed XML output, where the publication year has been incremented by 1 for all albums in the series "Astérix" published after 1980.

q19:

explication: la balise album de l'album numéro 1 de chaque série par "Premier\_album".

```
Fichier Éditer Rechercher Projet Options Outils Débogueur Document Fenêtre Aide
XML: albums.xml XQuery: exo1_q19.tp6.xquery Sortie:
Saison-PE XQuery 9.6.0.7
exo1_q15.tp6.xquery x exo1_q16.tp6.xquery x exo1_q17.tp6.xquery x exo1_q18.tp6.xquery x exo1_q19.tp6.xquery x exo1_q20.tp6.xquery x
albums.xml x
1 for $premier in doc("albums.xml")//album[@numero = "1"]
2 return
3 <album numero="Premier_album" serie="{ $premier/@serie }">
4 { $premier/titre }
5 { $premier/auteur }
6 { $premier/date }
7 </album>
8
Sortie
1 <?xml version="1.0" encoding="UTF-8"?>
2 <album numero="Premier_album" serie="Tintin">
3 <titre>Tintin au pays des Soviets</titre>
4 <auteur>Goscinny</auteur>
5 <date>
6 <mois>1930</mois>
7 <mois>septembre</mois>
8 </date>
9 </album>
10 <album numero="Premier_album" serie="Astérix">
11 <titre>Astérix le Gaulois</titre>
12 <auteur>Goscinny</auteur>
13 <auteur>Uderzo</auteur>
14 <date>
15 <mois>octobre</mois>
16 <annee>1961</annee>
17 </date>
18 </album>
19 <album numero="Premier_album" serie="Pif et Hercule">
20 <titre>Pif et Hercule à la conquête du trésor</titre>
21 <auteur>Jean Dupont</auteur>
22 <date>
23 <mois>janvier</mois>
24 <annee>1968</annee>
25 </date>
26 </album>
27 </albums>
28
Texte x
```

q20:

explication: mon fichier on serie tintin + nouveau album + autre serie .

```
Fichier Éditer Rechercher Projet Options Outils Débogueur Document Fenêtre Aide
XML: albums.xml XQuery: exo1_q20.tp6.xquery Sortie:
Saison-PE XQuery 9.6.0.7
exo1_q15.tp6.xquery x exo1_q16.tp6.xquery x exo1_q17.tp6.xquery x exo1_q18.tp6.xquery x exo1_q19.tp6.xquery x exo1_q20.tp6.xquery x
albums.xml x
1 let $albums := doc("albums.xml")//album[@serie="Tintin"]
2 let $nouveau := doc("albums.xml")//album[@serie="Tintin"]
3
4 <album numero="25" serie="Tintin">
5 <titre>Le Nouveau Mystère</titre>
6 <auteur>Hergé</auteur>
7 <date>
8 <mois>mars</mois>
9 <annee>2025</annee>
10 </date>
11 </album>
12 return
13 <albums>
14 {
15 for $a in $albums
16 return $a
17 }
18 { $nouveau }
19 </albums>
20
21 for $a in $albums2
22 return $a
23 </albums>
24
Sortie
166 </date>
167 </album>
168 <album numero="24" serie="Tintin">
169 <titre>Tintin et l'Alph-Art</titre>
170 <date>
171 <mois>octobre</mois>
172 <annee>1986</annee>
173 </date>
174 </album>
175 <album numero="25" serie="Tintin">
176 <titre>Le Nouveau Mystère</titre>
177 <auteur>Hergé</auteur>
178 <date>
179 <mois>mars</mois>
180 <annee>2025</annee>
181 </date>
182 </album>
183 <album numero="1" serie="Astérix">
184 <titre>Astérix le Gaulois</titre>
185 <auteur>Goscinny</auteur>
186 <auteur>Uderzo</auteur>
187 <date>
188 <mois>octobre</mois>
189 <annee>1961</annee>
190 </date>
191 </album>
192 </albums>
193
Texte x
```

-SOL EXO 2:

q1:

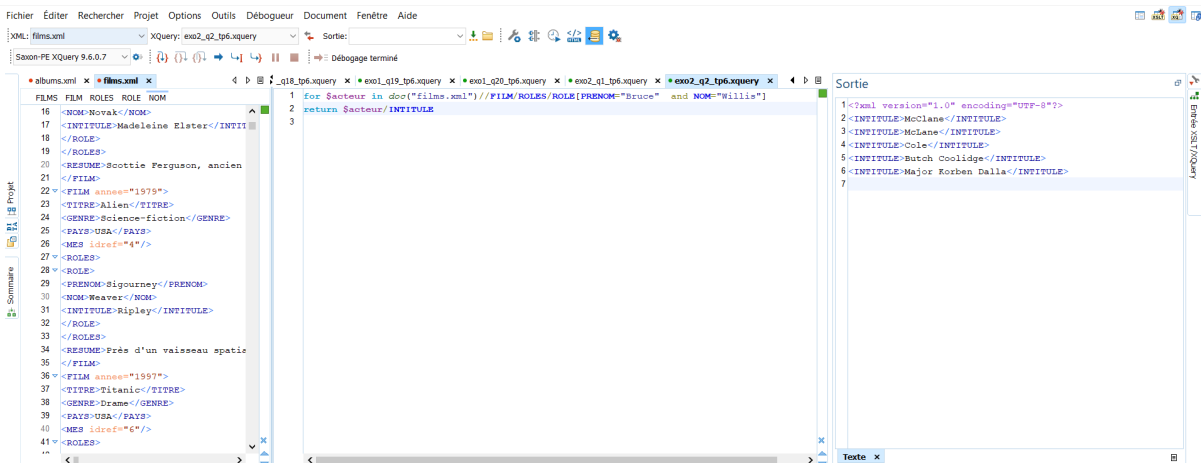
explication: selectionner les films annee <1970 et afficher titre, genre,pays .

```
Fichier Éditer Rechercher Projet Options Outils Débogueur Document Fenêtre Aide
XML: films.xml XQuery: exo2_q1.tp6.xquery Sortie:
Saison-PE XQuery 9.6.0.7
albums.xml x films.xml x
1 for $film in doc("films.xml")//FILM[@annee<1970]
2 return
3 <?xml version="1.0" encoding="UTF-8"?>
4 { $film/TITRE }
5 { $film/GENRE }
6 { $film/PAYS }
7 </FILM>
8
Sortie
1 <?xml version="1.0" encoding="UTF-8"?>
2 <FILM annee="1958">
3 <TITRE>Vertigo</TITRE>
4 <GENRE>Drame</GENRE>
5 <PAYS>USA</PAYS>
6 </FILM>
7 <FILM annee="1957">
8 <TITRE>12 Hommes en colère</TITRE>
9 <GENRE>Drame</GENRE>
10 <PAYS>USA</PAYS>
11 </FILM>
12 <FILM annee="1960">
13 <TITRE>Psychose</TITRE>
14 <GENRE>
15 <PAYS>USA</PAYS>
16 </FILM>
17 <FILM annee="1963">
18 <TITRE>Les oiseaux</TITRE>
19 <GENRE>Horreur</GENRE>
20 <PAYS>USA</PAYS>
21 </FILM>
22 <FILM annee="1964">
23 <TITRE>Pas de printemps pour Marnie</TITRE>
24 <GENRE>
25 <PAYS>USA</PAYS>
26 </FILM>
27 </films>
28
Texte x
```



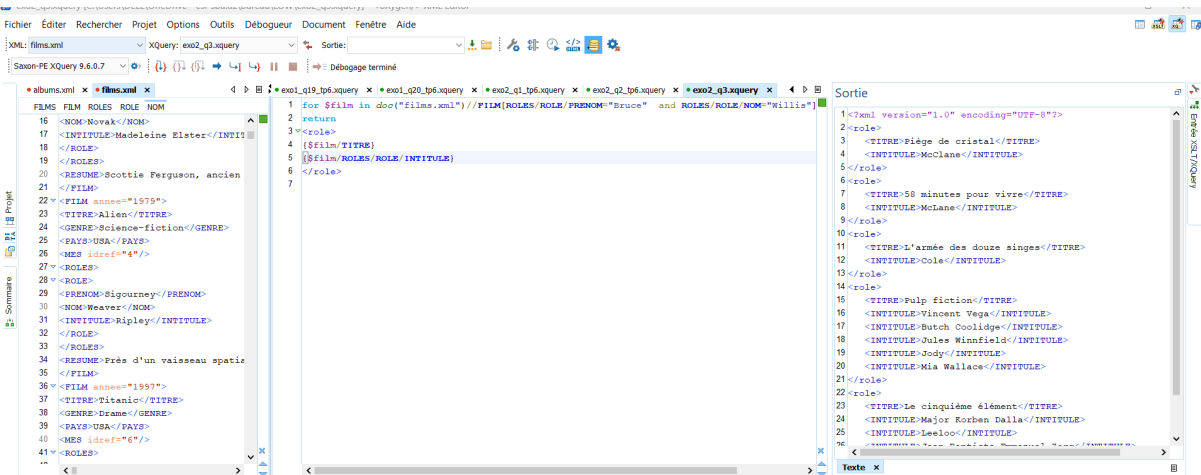
q2:

explication: selectionner les role de bruce willis et donne intitule.



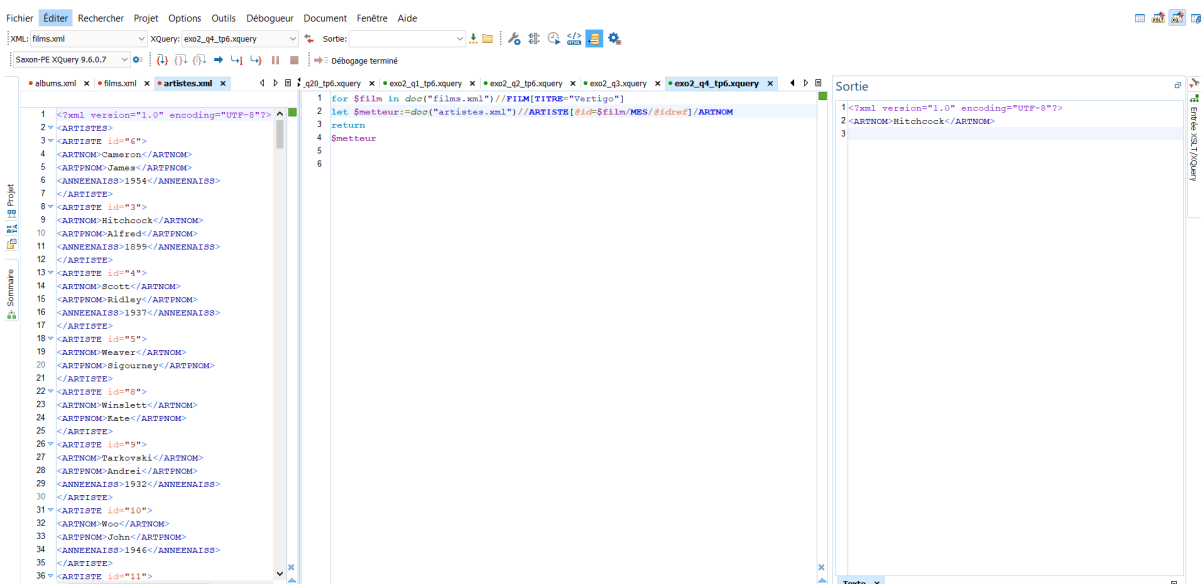
q3:

explication: selectionner les film de bruce willis et donne titre + intitule.



q4:

explication: selectionner les film vertigo et metteur de meme idref et l'afficher.



q5:

explication: selectionner les artiste et afficher tout les films avec idref f=de id de cet artiste.

```
1 for $artiste in doc("artistes.xml")//ARTISTE
2 let $film := doc("films.xml")//FILM[MES/Idref=$artiste/Id]
3 return
4 <artiste>
5 <nom>{$artiste/ARTNOM}</nom>
6 </nom>
7 <FILM annee="1957">
8 <TITRE>Titanic</TITRE>
9 <GENRE>Drame</GENRE>
10 <PAYS>USA</PAYS>
11 <MES idref="6"/>
12 </FILM>
13 </artiste>
```

q6:

explication: selectionner film et son artiste apes calculer annee-date de naissance et l'afficher avec titre.

```
1 for $film in doc("films.xml")//FILM
2 let $artiste := doc("artistes.xml")//ARTISTE[Id=$film/MES/Idref]
3 let $annee-sortie:=$film/ANNEE
4 let $date-naissance:=$artiste/ANNEENAISS
5 let $age := number($annee-sortie) - number($date-naissance)
6 return
7 <film>
8 <titre>{$film/TITRE}</titre>
9 </titre>
10 <age>{$age}</age>
11 </film>
```

q7:

explication: selectionner les film de meme genre et les grouper

The screenshot shows the XQuery editor with a query that groups films by genre. The query is as follows:

```
1 for $genre in distinct-values(doc("films.xml")//FILM/GENRE)
2 let $films := doc("films.xml")//FILM[GENRE=$genre]
3 return
4 <GENRE name="{ $genre }">
5 {
6   [for $film in $films
7    return $film/TITRE]
8 }
9 </GENRE>
```

The output window shows the result of the query, which is a list of genres and their corresponding film titles. The genres are: Drame, Science-fiction, Fantastique, and Comédie dramatique. The film titles listed under each genre are: Vertigo, Titanic, Sacrifice, 12 Hommes en colère, Un homme d'exception, Gladiator, Van Gogh, Rain Man, Le grand bleu, Alien, L'armée des douze singes, Terminator, Le retour du Jedi, Le cinquième élément, Volte/Face, Blade Runner, Piège de cristal, 58 minutes pour vivre, Pulp fiction, Godzilla, Mission: Impossible, Top Gun, and Léon.

q8:

explication: selectionner les film de qu'ils ont l'artiste come un role est aussi mes et les afficher dans artiste.

The screenshot shows the XQuery editor with a query that selects films where an actor is also an artist. The query is as follows:

```
1 <ARTISTES>
2 {
3   for $artiste in doc("artistes.xml")//ARTISTE
4   let $films := doc("films.xml")//FILM[MES/$idref = $artiste/ARTNOM and
5     (for $role in //ROLE where $role/NOM = $artiste/ARTNOM and $role/PRENOM = $artiste/ARTNOM return true())]
6   where $films
7   return
8     <ARTISTE nom="{ $artiste/ARTNOM }" ($artiste/ARTNOM)>
9     {
10       [for $film in $films
11        return
12          <FILM annee="{ $film/$annee }">
13            { $film/TITRE }
14        ]
15     }
16   }
17 </ARTISTES>
```

The output window shows the result of the query, which is a list of artists and their corresponding film titles. The artists listed are: Clint Eastwood, Quentin Tarantino, and Reservoir dogs.