

TME XQuery : séance 1

L'énoncé original de TME se trouve ici

Exercice 1

Les requêtes de XMark

1. Le nom de la personne dont l'identifiant est "person0".

```
//person[@id="person0"]/name
```

Résultat

```
<name>Xiulin Poch</name>
```

2. La valeur initiale des trois premières enchères en cours.

```
for $i in (1 to 3)
return <result> { //open_auctions/auction[$i]/@id } { //open_auctions/auction[$i]/initial } </
```

Résultat

```
<result id="open_auction0">
  <initial>112.22</initial>
</result>
<result id="open_auction1">
  <initial>145.44</initial>
</result>
<result id="open_auction2">
  <initial>93.60</initial>
</result>
```

3. La valeur de la première et de la dernière augmentation effectuée sur les trois premières des enchères en cours, selon l'ordre des enchères défini dans les données xml (ne pas trier les enchères chronologiquement).

```
for $li in (1 to 3)
  let $first := data(//open_auctions/auction[$li]//bidder[1]/increase)
  let $lastIndex := count(//open_auctions/auction[$li]//bidder)
  let $last := data(//open_auctions/auction[$li]//bidder[$lastIndex]/increase)
  return
    <result>
      { //open_auctions/auction[$li]/@id }
      <first>{$first}</first>
      <last>{$last}</last>
</result>
```

Résultat

```

<result id="open_auction0">
  <first>18.00</first>
  <last>3.00</last>
</result>
<result id="open_auction1">
  <first>7.50</first>
  <last>6.00</last>
</result>
<result id="open_auction2">
  <first>4.50</first>
  <last>6.00</last>
</result>

```

4. Le prix des objets vendu à plus de 480.

```
//closed_auctions/auction[price>480]/price
```

Résultat

```

<price>482.90</price>
<price>617.31</price>
<price>594.95</price>
<price>534.37</price>
<price>494.66</price>

```

5. Le nom des objets du continent africain

```
//regions/africa/item/name
```

Résultat

```

<name>duteous nine eighteen</name>
<name>condemn</name>
<name>earnestly subtle spotted attend</name>
<name>poisons</name>
<name>thought inland different</name>
<name>approves</name>
<name>disguise engross hero restraint</name>
<name>renown stained entrails bone</name>
<name>irrevocable holding succeeding</name>
<name>unloose freshness swallowing</name>
<name>parson sure heavy</name>
<name>henceforward decreed</name>
<name>contain spring fate rebellious</name>
<name>sour</name>
<name>canonized piece</name>
<name>truths</name>
<name>forced strangle violence apprehends</name>
<name>sleeps</name>
<name>nev puts</name>

```

```

<name>hie stirr divine niece</name>
<name>fie carrying cassio</name>
<name>bestowed liable congruent everything</name>
<name>foot</name>
<name>bewitched proculeius away quoth</name>
<name>greg house turnips</name>
<name>tripp beef prepar</name>
<name>conquer insupportable assurance</name>
<name>perpetual patient company well</name>
<name>fair use conveyance</name>
<name>balthasar bred breathe</name>
<name>villainy</name>
<name>buried ere burst</name>
<name>flourish harms oratory</name>
<name>esteem observance else</name>
<name>lecherous mortal</name>
<name>stake cassandra shake</name>
<name>relate shame pedant flat</name>
<name>prudence jewels deserves sacred</name>

```

6. Le nom des objets du continent africain avec leur prix de vente

```

for $item in //regions/africa/item, $vente in //closed_auctions/auction
where $vente/itemref[@item=data($item/@id)]
return
<res>
  {$item/name}
  {$vente/price}
</res>

```

Résultat

```

<res>
  <name>condemn</name>
  <price>14.80</price>
</res>
<res>
  <name>thought inland different</name>
  <price>141.43</price>
</res>
<res>
  <name>disguise engross hero restraint</name>
  <price>93.55</price>
</res>
<res>
  <name>irrevocable holding succeeding</name>
  <price>38.59</price>
</res>

```

```

<res>
  <name>henceforward decreed</name>
  <price>174.92</price>
</res>
<res>
  <name>truths</name>
  <price>109.40</price>
</res>
<res>
  <name>nev puts</name>
  <price>68.19</price>
</res>
<res>
  <name>hie stirr divine niece</name>
  <price>43.54</price>
</res>
<res>
  <name>bewitched proculeius away quoth</name>
  <price>35.75</price>
</res>
<res>
  <name>greg house turnips</name>
  <price>80.08</price>
</res>
<res>
  <name>tripp beef prepar</name>
  <price>4.00</price>
</res>
<res>
  <name>conquer insupportable assurance</name>
  <price>249.85</price>
</res>
<res>
  <name>fair use conveyance</name>
  <price>34.71</price>
</res>
<res>
  <name>balthasar bred breathe</name>
  <price>63.02</price>
</res>
<res>
  <name>buried ere burst</name>
  <price>1.90</price>
</res>
<res>
  <name>flourish harms oratory</name>

```

```

    <price>54.38</price>
  </res>
  <res>
    <name>lecherous mortal</name>
    <price>133.49</price>
  </res>
  <res>
    <name>prudence jewels deserves sacred</name>
    <price>66.68</price>
  </res>

```

7. Le nombre de personnes qui n'ont pas de page web (i.e. les personnes sans élément homepage)

```
count(//people/person)-count(//people/person/homepage)
```

Résultat

874

Exercice 2

1. Donner l'année et le titre des livres publiés par Addison Wesley après 1991.

```

for $book in //bib/book[@year>1991]
where data($book/publisher) = 'Addison-Wesley'
return
  <book>
    {$book/@year}
    {$book/title}
  </book>

```

Résultat

```

<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
  </book>
</bib>

```

2. Donner la liste de tous les couples (titre, auteur). Un couple est formé par un élément <result> contenant un titre et un auteur.

```

<results>
{
  for $book in //bib/book

```

```

    where $book/author
    return
    <result>
      {$book/title}
      {$book/author}
    </result>
  }
</results>

```

Résultat

```

<results>
  <result>
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </result>
  <result>
    <title>Advanced Programming in the Unix environment</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </result>
  <result>
    <title>Data on the Web</title>
    <author>
      <last>Abiteboul</last>
      <first>Serge</first>
    </author>
  </result>
  <result>
    <title>Data on the Web</title>
    <author>
      <last>Buneman</last>
      <first>Peter</first>
    </author>
  </result>
  <result>
    <title>Data on the Web</title>
    <author>
      <last>Suciu</last>
      <first>Dan</first>
    </author>
  </result>

```

```
</results>
```

3. Pour chaque livre, donner le titre et les auteurs regroupés dans un élément `<results>`.

```
<results>
{
  for $livre in //bib/book
  return
    <result>
      {$livre/title}
      {$livre/author}
    </result>
}
</results>
```

Résultat

```
<results>
  <result>
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </result>
  <result>
    <title>Advanced Programming in the Unix environment</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </result>
  <result>
    <title>Data on the Web</title>
    <author>
      <last>Abiteboul</last>
      <first>Serge</first>
    </author>
    <author>
      <last>Buneman</last>
      <first>Peter</first>
    </author>
    <author>
      <last>Suciu</last>
      <first>Dan</first>
    </author>
  </result>
```

```

    <result>
      <title>The Economics of Technology and Content for Digital TV</title>
    </result>
  </results>

```

4. Pour chaque auteur, donner la liste des titres de ses livres. Un élément du résultat contient un auteur avec tous les titres qu'il a écrit.

```

<results>
{
  for $authorLast in distinct-values(//author/last), $authorFirst in distinct-values(//author/first)
  where //author[first=$authorFirst][last=$authorLast]
  let $author := //author[first=$authorFirst][last=$authorLast]
  return
    <result>
    { $author[1] }
    { //book[author=$author]/title }
    </result>
}
</results>

```

Résultat

```

<results>
  <result>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
    <title>TCP/IP Illustrated</title>
    <title>Advanced Programming in the Unix environment</title>
  </result>
  <result>
    <author>
      <last>Abiteboul</last>
      <first>Serge</first>
    </author>
    <title>Data on the Web</title>
  </result>
  <result>
    <author>
      <last>Buneman</last>
      <first>Peter</first>
    </author>
    <title>Data on the Web</title>
  </result>
  <result>
    <author>

```



```

        <last>Suciu</last>
        <first>Dan</first>
      </author>
      <title>Data on the Web</title>
    </result>
  </results>

```

5. Pour chaque livre de la bibliographie Bib qui a une critique dans Review, donner son titre et tous ses prix.

```

<books-with-prices>
{
  let $reviews := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/r
  for $book in //book
  where $reviews//entry[title=$book/title]
  return
    <book-with-prices>
      { $book/title }
      <price-review>
        {data($reviews//entry[title=$book/title]/price)}
      </price-review>
      <price-bib>
        {data($book/price)}
      </price-bib>
    </book-with-prices>
}
</books-with-prices>

```

Résultat

```

<books-with-prices>
  <book-with-prices>
    <title>TCP/IP Illustrated</title>
    <price-review>65.95</price-review>
    <price-bib>65.95</price-bib>
  </book-with-prices>
  <book-with-prices>
    <title>Advanced Programming in the Unix environment</title>
    <price-review>65.95</price-review>
    <price-bib>65.95</price-bib>
  </book-with-prices>
  <book-with-prices>
    <title>Data on the Web</title>
    <price-review>34.95</price-review>
    <price-bib>39.95</price-bib>
  </book-with-prices>
</books-with-prices>

```

6. Pour chaque livre ayant au moins un auteur, donner le titre et le nom d'au plus deux auteurs. Le nom des auteurs suivants est remplacé par l'élément <et-al> (i.e., abréviation latine signifiant et les autres).

```
for $book in //book
where $book/author
return
  if (count($book/author) >= 1) then (
    if (count($book/author) >= 2) then (
      if (count($book/author) >= 3) then (
        <book>
          {$book/title}
          {$book/author[1]}
          {$book/author[2]}
          <et-al/>
        </book>
      )
    else (
      <book>
        {$book/title}
        {$book/author}
      </book>
    )
  )
  else (
    <book>
      {$book/title}
      {$book/author}
    </book>
  )
  else
  ()
```

Résultat

```
<bib>
  <book>
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </book>
  <book>
    <title>Advanced Programming in the Unix environment</title>
    <author>
```

```

        <last>Stevens</last>
        <first>W.</first>
    </author>
</book>
<book>
    <title>Data on the Web</title>
    <author>
        <last>Abiteboul</last>
        <first>Serge</first>
    </author>
    <author>
        <last>Buneman</last>
        <first>Peter</first>
    </author>
    <et-al/>
</book>
</bib>

```

7. Donner, dans l'ordre alphabétique, le titre et l'année des livres publiés par Addison-Wesley après 1991.

```

for $book in //book[publisher = "Addison-Wesley"][@year >1991]
order by $book/title
return
    <book>
        {$book/@year}
        {$book/title}
    </book>

```

Résultat

```

<bib>
    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
    </book>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
    </book>
</bib>

```

8. Quels sont les livres qui ont un élément dont le nom se termine par «or» et dont un sous élément contient la chaîne «Suciu».

```

for $book in //book , $b in $book/*
where ends-with(name($b),"or") and contains($book,"Suciu")
return $book

```

Résultat

```

<book>

```

```

<title>Data on the Web</title>
<author>
  <last>Suciu</last>
  <first>Dan</first>
</author>
</book>

```

9. Dans le livre book.xml, quels sont tous les titres de chapitre ou de section qui contiennent le mot « XML » ?

```

let $books := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/book.xml")
for $title in $books//title
where contains($title,"XML")
return $title/..

```

Résultat

```

<section>
  <title>XML</title>
</section>
<section>
  <title>Basic Syntax</title>
</section>
<section>
  <title>XML and Semistructured Data</title>
</section>
</section>
<section>
  <title>XML and Semistructured Data</title>
</section>

```

10. Dans le document price.xml, donner le prix le moins cher de chaque livre. Le résultat est une liste d'éléments <minprice>. Le titre est un attribut de <minprice>, le prix est un élément de <minprice>.

```

let $prices := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/price.xml")
for $title in distinct-values($prices//title)
let $mp := min($prices//book[title = $title]/price)
return
  <minprice title = '{$title}'>
    <price>
      {$mp}
    </price>
  </minprice>

```

Résultat

```

<results>
  <minprice title="Advanced Programming in the Unix environment">
    <price>65.95</price>
  </minprice>

```

```

</minprice>
<minprice title="TCP/IP Illustrated">
  <price>65.95</price>
</minprice>
<minprice title="Data on the Web">
  <price>34.95</price>
</minprice>
</results>

```

11. Pour chaque livre avec au moins un auteur, donner le titre et les auteurs, pour chaque livre avec un éditeur, donner une référence contenant le titre du livre et l'affiliation de l'éditeur.

```

<bib>
{
  for $ba in //book
  where $ba/author
  return
    <book>
      {$ba/title}
      {$ba/author}
    </book>
}
{
  for $be in //book
  where $be/editor
  return
    <reference>
      {$be/title}
      {$be/editor/affiliation}
    </reference>
}
</bib>

```

Résultat

```

<bib>
  <book>
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
  </book>
  <book>
    <title>Advanced Programming in the Unix environment</title>
    <author>
      <last>Stevens</last>

```

```

        <first>W.</first>
    </author>
</book>
<book>
    <title>Data on the Web</title>
    <author>
        <last>Abiteboul</last>
        <first>Serge</first>
    </author>
    <author>
        <last>Buneman</last>
        <first>Peter</first>
    </author>
    <author>
        <last>Suciu</last>
        <first>Dan</first>
    </author>
</book>
<reference>
    <title>The Economics of Technology and Content for Digital TV</title>
    <affiliation>CITI</affiliation>
</reference>
</bib>

```

12. Trouver les paires de livres qui ont les mêmes auteurs et des titres différents.
Ne pas tenir compte de l'ordre des auteurs

```

<bib>
{
    for $book1 in //book , $book2 in //book
    where $book1/author = $book2/author and index-of(//book,$book1) < index-of(//book,$book2)
    return
        <book-pair>
        { $book1/title }
        { $book2/title }
        </book-pair>
}
</bib>

```

Résultat

```

<bib>
    <book-pair>
        <title>TCP/IP Illustrated</title>
        <title>Advanced Programming in the Unix environment</title>
    </book-pair>
</bib>

```

TME XQuery : séance 2

L'énoncé original de TME se trouve ici

Exercice 1

- Afficher tous les tournois (lieu,année) triés par année, puis lieu. Utiliser la fonction `distinct-values(...)` pour obtenir un ensemble sans doubles. Le résultat doit être `res0.txt`

On charge d'abord la base avec le fichier `rencontre.xml`. Et on lance les commandes suivantes.

```
for $rannee in distinct-values(//rencontre/annee) , $rlieu in distinct-values(//rencontre/lieu)
where //rencontre[annee=$rannee][lieutournoi=$rlieu]
  let $r := //rencontre[annee=$rannee][lieutournoi=$rlieu]
  let $a := $r/annee
  let $l := $r/lieutournoi
  return
    <tournoi lieu = "{$l}" annee="{$a}">
  </tournoi>
```

Résultat

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tournois>
  <tournoi lieu="Flushing Meadow" annee="1989"></tournoi>
  <tournoi lieu="Wimbledon" annee="1989"></tournoi>
  <tournoi lieu="Roland Garros" annee="1990"></tournoi>
  <tournoi lieu="Flushing Meadow" annee="1991"></tournoi>
  <tournoi lieu="Roland Garros" annee="1992"></tournoi>
  <tournoi lieu="Wimbledon" annee="1992"></tournoi>
  <tournoi lieu="Wimbledon" annee="1993"></tournoi>
  <tournoi lieu="Roland Garros" annee="1994"></tournoi>
</tournois>
```

- pour chaque tournoi (lieu, année) donner les nom et prénom des participants qui sont dans le fichier `gain.xml`. Le résultat doit être `res1.txt`

Pour cette question on charge la base avec le fichier `gain.xml` et on lance les commandes suivantes.

```
let $joueurs := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/joueur.xml")
let $rencontres := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/rencontre.xml")
for $gain in //gain
  let $glieu := $gain/lieutournoi
  let $gannee := $gain/annee
  let $rtournoi := $rencontres//rencontre[lieutournoi=$glieu][annee=$gannee]
```

```

let $gperdants := distinct-values($rtournoi//nuperdant)
let $ggagnants := distinct-values($rtournoi/nugagnant)
let $tnbjoueurs := distinct-values(($gperdants,$ggagnants))
return
  <tournoi lieu='{$glieu}' annee='{$gannee}'>
  {
    for $j in $tnbjoueurs
    let $joueur := $joueurs[nujoueur=$j]
    return
    <participant nom="{ $joueur/nom}" prenom="{ $joueur/prenom}">
    </participant>
  }
</tournoi>

```

Résultat

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<tournois>
  <tournoi lieu="Flushing Meadow" annee="1989">
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="CONNORS" prenom="Jimmy"></participant>
  </tournoi>
  <tournoi lieu="Wimbledon" annee="1989">
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
    <participant nom="LECONTE" prenom="Henri"></participant>
    <participant nom="WILANDER" prenom="Mats"></participant>
    <participant nom="CONNORS" prenom="Jimmy"></participant>
    <participant nom="McENROE" prenom="John"></participant>
  </tournoi>
  <tournoi lieu="Roland Garros" annee="1990">
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="WILANDER" prenom="Mats"></participant>
    <participant nom="CONNORS" prenom="Jimmy"></participant>
    <participant nom="McENROE" prenom="John"></participant>
  </tournoi>
  <tournoi lieu="Flushing Meadow" annee="1991">
    <participant nom="LARSSON" prenom="Magnus"></participant>
    <participant nom="CONNORS" prenom="Jimmy"></participant>
  </tournoi>
  <tournoi lieu="Roland Garros" annee="1992">
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
  </tournoi>

```



```

    <participant nom="EDBERG" prenom="Stephan"></participant>
    <participant nom="LARSSON" prenom="Magnus"></participant>
    <participant nom="LECONTE" prenom="Henri"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="WILANDER" prenom="Mats"></participant>
    <participant nom="CONNORS" prenom="Jimmy"></participant>
    <participant nom="McENROE" prenom="John"></participant>
    <participant nom="SAMPRAS" prenom="Pete"></participant>
  </tournoi>
  <tournoi lieu="Wimbledon" annee="1992">
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
    <participant nom="HALARD" prenom="Julie"></participant>
    <participant nom="PIERCE" prenom="Mary"></participant>
    <participant nom="EDBERG" prenom="Stephan"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="McENROE" prenom="John"></participant>
    <participant nom="SAMPRAS" prenom="Pete"></participant>
  </tournoi>
  <tournoi lieu="Wimbledon" annee="1993">
    <participant nom="MARTINEZ" prenom="Conchita"></participant>
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="HALARD" prenom="Julie"></participant>
    <participant nom="PIERCE" prenom="Mary"></participant>
    <participant nom="LARSSON" prenom="Magnus"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="FLEURIAN" prenom="Jean-Philippe"></participant>
    <participant nom="SAMPRAS" prenom="Pete"></participant>
  </tournoi>
  <tournoi lieu="Roland Garros" annee="1994">
    <participant nom="MARTINEZ" prenom="Conchita"></participant>
    <participant nom="NAVRATILOVA" prenom="Martina"></participant>
    <participant nom="GRAF" prenom="Steffi"></participant>
    <participant nom="HALARD" prenom="Julie"></participant>
    <participant nom="LECONTE" prenom="Henri"></participant>
    <participant nom="FORGET" prenom="Guy"></participant>
    <participant nom="FLEURIAN" prenom="Jean-Philippe"></participant>
    <participant nom="SAMPRAS" prenom="Pete"></participant>
  </tournoi>
</tournois>

```

- pour chaque année, donner le nombre tournois auxquels chaque joueur a participé. Le résultat doit être res2.txt

On charge la base à partir de fichier rencontre.xml qui contient les rencontres entre les joueurs.

```

let $joueurs := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquery/joueur
for $ranne in distinct-values(//rencontre/annee)
return
  <annee valeur="{ $ranne}">
  {
    for $joueur in $joueurs
    where //rencontre[annee=$ranne] [nuperdant=$joueur/@num] or //rencontre[annee=$ranne]
    let $rperdant := //rencontre[annee=$ranne] [nuperdant=$joueur/@num]
    let $rgagnant := //rencontre[annee=$ranne] [nugagnant=$joueur/@num]
    let $jr := distinct-values(($rperdant,$rgagnant))
    return <joueur nom="{ $joueur/nom}" tournois="{count($jr)}"></joueur>
  }
</annee>

```

Résultat

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<reponse>
<annee valeur="1989">
  <joueur nom="NAVRATILOVA" tournois="2"></joueur>
  <joueur nom="GRAF" tournois="2"></joueur>
  <joueur nom="LECONTE" tournois="1"></joueur>
  <joueur nom="FORGET" tournois="1"></joueur>
  <joueur nom="WILANDER" tournois="1"></joueur>
  <joueur nom="CONNORS" tournois="2"></joueur>
  <joueur nom="McENROE" tournois="1"></joueur>
</annee>
<annee valeur="1990">
  <joueur nom="NAVRATILOVA" tournois="1"></joueur>
  <joueur nom="GRAF" tournois="1"></joueur>
  <joueur nom="FORGET" tournois="1"></joueur>
  <joueur nom="WILANDER" tournois="1"></joueur>
  <joueur nom="CONNORS" tournois="1"></joueur>
  <joueur nom="McENROE" tournois="1"></joueur>
</annee>
<annee valeur="1991">
  <joueur nom="LARSSON" tournois="1"></joueur>
  <joueur nom="CONNORS" tournois="1"></joueur>
</annee>
<annee valeur="1992">
  <joueur nom="NAVRATILOVA" tournois="2"></joueur>
  <joueur nom="GRAF" tournois="2"></joueur>
  <joueur nom="HALARD" tournois="1"></joueur>
  <joueur nom="PIERCE" tournois="1"></joueur>
  <joueur nom="EDBERG" tournois="2"></joueur>
  <joueur nom="LARSSON" tournois="1"></joueur>
  <joueur nom="LECONTE" tournois="1"></joueur>

```

```

    <joueur nom="FORGET" tournois="2"></joueur>
    <joueur nom="WILANDER" tournois="1"></joueur>
    <joueur nom="CONNORS" tournois="1"></joueur>
    <joueur nom="McENROE" tournois="2"></joueur>
    <joueur nom="SAMPRAS" tournois="2"></joueur>
  </annee>
  <annee valeur="1993">
    <joueur nom="MARTINEZ" tournois="1"></joueur>
    <joueur nom="NAVRATILOVA" tournois="1"></joueur>
    <joueur nom="HALARD" tournois="1"></joueur>
    <joueur nom="PIERCE" tournois="1"></joueur>
    <joueur nom="LARSSON" tournois="1"></joueur>
    <joueur nom="FORGET" tournois="1"></joueur>
    <joueur nom="FLEURIAN" tournois="1"></joueur>
    <joueur nom="SAMPRAS" tournois="1"></joueur>
  </annee>
  <annee valeur="1994">
    <joueur nom="MARTINEZ" tournois="1"></joueur>
    <joueur nom="NAVRATILOVA" tournois="1"></joueur>
    <joueur nom="GRAF" tournois="1"></joueur>
    <joueur nom="HALARD" tournois="1"></joueur>
    <joueur nom="LECONTE" tournois="1"></joueur>
    <joueur nom="FORGET" tournois="1"></joueur>
    <joueur nom="FLEURIAN" tournois="1"></joueur>
    <joueur nom="SAMPRAS" tournois="1"></joueur>
  </annee>
</reponse>

```

Exercice 2

- L'exercice 2 de TD a été fait lors de la séance 1. Il reste les exercices 1 et 3 à faire.

Exercice 1

1) Requêtes

On charge la base à partir de fichier guide.xml

- Donner le nom de tous les menus des restaurants

```

<results>
  {
    for $menu in //menu
      return <menu>
  }

```

```

        {$menu/@nom}
      </menu>
    }
  </results>

```

Résultat

```

<results>
  <menu nom="buffet"/>
  <menu nom="gourmet"/>
  <menu nom="specialites lyonnaises"/>
  <menu nom="vegetarien"/>
  <menu nom="standard"/>
  <menu nom="enfant"/>
  <menu nom="big"/>
  <menu nom="assiette printaniere"/>
  <menu nom="salade d'ete"/>
  <menu nom="menu d'automne"/>
  <menu nom="menu d'hiver"/>
</results>

```

- Donner le nom et le prix de tous les menus dont le prix est inférieur à 100

```

<resultats>
  { //menu[@prix<100] }
</resultats>

```

Résultat

```

<results>
  <menu nom="standard" prix="35"/>
  <menu nom="enfant" prix="20"/>
  <menu nom="big" prix="49"/>
  <menu nom="assiette printaniere" prix="50"/>
  <menu nom="salade d'ete" prix="30"/>
  <menu nom="menu d'hiver" prix="99"/>
</results>

```

- Donner le nom des restaurants 2 étoiles avec leur nom de menu

```

<results>
{
  for $resto in//restaurant[@etoile=2]
  return
    <restaurant nom="{ $resto/@nom}">
    {
      for $menu in $resto/menu
      return
        <menu nom="{ $menu/@nom}">
        </menu>
    }
}

```

```

    }
  </restaurant>
}
</results>

```

Résultat

```

<results>
  <restaurant nom="chez Bocuse">
    <menu nom="specialites lyonnaises"/>
    <menu nom="vegetarien"/>
  </restaurant>
</results>

```

- Donner le nom de chaque restaurant avec son numéro de département

```

<results>
  {
    for $resto in //restaurant
    let $v := //ville[@nom=$resto/@ville]
    return
    <restaurant nom="{ $resto/@nom}" departement="{ $v[@nom=$resto/@ville]/@departement}" />
  }
</results>

```

Résultat

```

<results>
  <restaurant nom="la tour d'argent" departement="75"/>
  <restaurant nom="chez Bocuse" departement="69"/>
  <restaurant nom="MacDo" departement="84"/>
  <restaurant nom="Les 4 saisons" departement="75"/>
</results>

```

- Quels sont les restaurants ayant (au moins) un menu dont le prix est égal au tarif de visite du plus beau monument de la ville ? Donner le nom du restaurant et le tarif.

```

<results>
  {
    for $pbm in //ville[plusBeauMonument], $restaurant in //restaurant
    where $restaurant[@ville = $pbm/@nom]/menu[@prix=$pbm/plusBeauMonument/@tarif]
    let $m := $pbm/plusBeauMonument
    return
    <result>
      <restaurant nom="{ $restaurant/@nom}" />
      <tarif_monument prix="{ $m/@tarif}" />
    </result>
  }

```

```
</results>
```

Résultat

```
<results>
  <result>
    <restaurant nom="Les 4 saisons"/>
    <tarif_monument prix="30"/>
  </result>
</results>
```

2) Présentation de données

Pour cette exercice on reste toujours dans la base qu'on avait rechargé à partir de guide.xml

- Définir la requête tableau.xql qui formate tous les restaurants dans un tableau XHTML. Une ligne du tableau contient le nom, la ville et le nombre d'étoiles du restaurant. Les restaurants sont triés par nom dans l'ordre alphabétique.

```
<table>
<tr>
<th>Nom</th>
<th>Ville</th>
<th>Nombre d'étoiles</th>
</tr>
{
  for $restaurant in //restaurant
  order by $restaurant/@nom
  return
    <tr>
      <td>{data($restaurant/@nom)}</td>
      <td>{data($restaurant/@ville)}</td>
      <td>{data($restaurant/@etoile)}</td>
    </tr>
}
</table>
```

Résultat

```
<table>
<tr>
  <th>Nom</th>
  <th>Ville</th>
  <th>Nombre d'étoiles</th>
</tr>
<tr>
```

```

        <td>Les 4 saisons</td>
        <td>Paris</td>
        <td>1</td>
    </tr>
    <tr>
        <td>MacDo</td>
        <td>Avignon</td>
        <td>0</td>
    </tr>
    <tr>
        <td>chez Bocuse</td>
        <td>Lyon</td>
        <td>2</td>
    </tr>
    <tr>
        <td>la tour d'argent</td>
        <td>Paris</td>
        <td>3</td>
    </tr>
</table>

```

- Définir la requête liste.xql qui présente la structure arborescente des éléments d'un document XML quelconque, sous la forme de listes XHTML imbriquées avec les éléments (liste), et (élément de liste).

A COMPLETER!!

Exercice 3

- **Structure aplatie et désimbriquée** Lister les joueurs avec leur classement et ordonnés par nationalité.

Pour cette question on recharge la base à partir de fichier **joueurTd.xml** et on lance le requête suivant.

```

<resultats>
{
for $joueur in //joueur
order by $joueur/identite/nationalite
return
    <resultat>
        { $joueur/identite/nom}
        { $joueur/identite/prenom}
        { $joueur/identite/nationalite}
        { $joueur/classement }
    </resultat>
}

```

```
</resultats>
```

Résultat

```
<resultats>
  <resultat>
    <nom>Hewitt</nom>
    <prenom>Lleyton</prenom>
    <classement>1</classement>
    <nationalite>AUS</nationalite>
  </resultat>
  <resultat>
    <nom>Arthurs</nom>
    <prenom>Wayne</prenom>
    <classement>55</classement>
    <nationalite>AUS</nationalite>
  </resultat>
  <resultat>
    <nom>Costa</nom>
    <prenom>Albert</prenom>
    <classement>8</classement>
    <nationalite>ESP</nationalite>
  </resultat>
  <resultat>
    <nom>Mathieu</nom>
    <prenom>Paul-Henry</prenom>
    <classement>44</classement>
    <nationalite>FR</nationalite>
  </resultat>
  <resultat>
    <nom>Clement</nom>
    <prenom>Arnaud</prenom>
    <classement>37</classement>
    <nationalite>FR</nationalite>
  </resultat>
  <resultat>
    <nom>Gasquet</nom>
    <prenom>Richard</prenom>
    <classement>124</classement>
    <nationalite>FR</nationalite>
  </resultat>
</resultats>
```

- **Rebalisage** Lister les joueurs en donnant leur nom, prénom et nationalité séparés par des virgules.

Pour cette question on reste toujours sur la base qu'on a chargé depuis le fichier **joueurTd.xml** et on lance le requête.


```

<resultats>
{
  for $joueur in //joueur
  order by $joueur/identite/nationalite
  return
  <resultat>
  { data($joueur/identite/nom)},{ data($joueur/identite/prenom) },{data($joueur/iden
  </resultat>
}
</resultats>

```

Résultat

```

<resultats>
<resultat>Hewitt,Lleyton,AUS</resultat>
<resultat>Arthurs,Wayne,AUS</resultat>
<resultat>Mathieu,Paul-Henry,FR</resultat>
<resultat>Clement,Arnaud,FR</resultat>
<resultat>Costa,Albert,ESP</resultat>
<resultat>Gasquet,Richard,FR</resultat>
</resultats>

```

- **Agrégation** Donner le nombre de joueurs par nationalité.

```

<resultats>
{
  for $nationalite in //joueur/identite/nationalite
  return
  <resultat>
  <nationalite>
  { $nationalite }
  </nationalite>
  <nb_joueurs>
  { count(//joueur/identite[nationalite=$nationalite]) }
  </nb_joueurs>
  </resultat>
}
</resultats>

```

Résultat

```

<resultats>
<resultat>
<nationalite>
AUS
</nationalite>
<nb_joueurs>
2
</nb_joueurs>

```

```

</resultat>
<resultat>
  <nationalite>
    FR
  </nationalite>
  <nb_joueurs>
    3
  </nb_joueurs>
</resultat>
<resultat>
  <nationalite>
    ESP
  </nationalite>
  <nb_joueurs>
    1
  </nb_joueurs>
</resultat>
</resultats>

```

- **Jointure** Pour chacune des rencontres, donner le classement des deux joueurs. Seules les rencontres entre joueurs de classement inférieur à 50 sont conservées.

Pour cette question on reste toujours sur la base qu'on a rechargé à partir de fichier **joueurTd.xml** et on va utiliser le fichier **rencontreTd.xml** pour les rencontres. On pouvait très bien faire l'invers (de partir sur la base à partir de fichier **rencontreTd.xml** et utiliser le fichier **joueurTd.xml** pour les classements)

```

<resultats>
{
  let $rencontres := doc("/Users/ky/Google Drive/Kisisel/UPMC/MLBDA/TD:TME/XQuery/xquer
    for $rencontre in $rencontres
      let $rji1 := $rencontre/joueur1/identite
      let $rji2 := $rencontre/joueur2/identite
      let $joueurs := //joueur[classement<=50]
      where $joueurs[identite=$rji1] and $joueurs[identite=$rji2]
      return
        <resultat>
          { $joueurs[identite=$rji1]/classement}
          { $joueurs[identite=$rji2]/classement}
        </resultat>
    }
</resultats>

```

Résultat

```

<resultats>

```

```

<resultat>
  <classement>
    1
  </classement>
  <classement>
    44
  </classement>
</resultat>
<resultat>
  <classement>
    37
  </classement>
  ><classement>
    44
  </classement>
</resultat>
</resultats>

```

- Dans la suite de l'exercice 2 de la séance 2 on récupère le fichier site.xml et sur la base construit à partir de ce fichier on réponds aux questions suivantes.
- **Question 6** Les appareils photo qui ont été utilisés pour faire plusieurs photos. Chaque appareil doit apparaître *une seule* fois dans le résultat. Le résultat doit satisfaire la DTD suivante:

```

< !ELEMENT résultats (appareil)*>
< !ELEMENT appareil EMPTY>
< !ATTLIST appareil nom CDATA>

<resultats>
{
  for $appareil in distinct-values(//photo/appareil)
  where count(//photo[appareil=$appareil])>1
  return
    <appareil nom="{ $appareil}"/>
}
</resultats>

```

Résultat

```

<resultats>
  <appareil nom="Samsung T1"/>
  <appareil nom="Nikon F1"/>
  <appareil nom="Canon D50"/>
</resultats>

```

- **Question 7** Pour chaque format de photo, le nombre total de photos de ce format qui ont été publiées dans *chaque collection*. Le nombre

total sera affiché dans un élément `<nb>`. S'il n'y a pas de photo avec un format donné dans une certaine collection, on affiche seulement le nom de la collection, *l'élément `<nb>` ne sera pas affiché*. Chaque format doit apparaître une seule fois dans le résultat. Le résultat doit satisfaire la DTD suivante :

```
< !ELEMENT résultats (format)*>
< !ELEMENT format (collection)*>
< !ATTLIST format nom CDATA>
< !ELEMENT collection (#PCDATA|nb) *>
< !ELEMENT nb (#PCDATA)>
```

Un exemple de fragment du document xml résultat correspondant au format 'jpeg' est le suivant:

```
<résultats>
  <format nom='jpeg'>
    <collection> Pour les copains <nb>2</nb>
  </collection>
  <collection> Collection publique </collection>
  <collection>Photos préférées </collection>
</format>
....
</résultats>
```

```
for $format in distinct-values(//photo/format)
let $photos := //photo[format=$format]
return
  <format nom="{ $format }">
    {
      for $collection in //collection
      let $publications := $collection//publication
      let $intersection :=
      for $publication in $publications , $photo in $photos
      where $publication/@idP = $photo/@idP
      return <nb>{count($photo)}</nb>
      return
      <collection>
        { data($collection/nom) }
        {
          if (count($intersection)>0)
          then (
            <nb>{count($intersection)}</nb>
          )
          else()
        }
      </collection>
    }
}
```

```
</format>
```

Résultat

```
<format nom="jpeg">
  <collection>Pour les copains<nb>2</nb>
</collection>
  <collection>Collection publique</collection>
  <collection>Photos preferees</collection>
</format>
<format nom="bmp">
  <collection>Pour les copains</collection>
  <collection>Collection publique<nb>1</nb>
</collection>
  <collection>Photos preferees</collection>
</format>
<format nom="gif">
  <collection>Pour les copains</collection>
  <collection>Collection publique</collection>
  <collection>Photos preferees<nb>3</nb>
</collection>
</format>
```

- **Question 8** On considère la requête XQuery **R** suivante:

```
<resultats>
{
  for $u1 in document("site.xml")//utilisateur,
  $u2 in document("site.xml")//utilisateur[@idU != $u1/@idU]
  let $a := for $v in
    distinct(document("site.xml")//photo[//collection[@idU=$u1/@idU]/publication/@idP=
      /appareil)
  where
    document("site.xml")//photo[//collection[@idU=$u2/@idU]/publication/@idP=@idP]/ap
  return $v
  where exists($a)
  return <utilisateurs id1="{ $u1/@idU}" id2="{ $u2/@idU}">
    { $a}
  </utilisateurs>
}
</resultats>
```

On lance ce requête en supprimant les expressions document("site.xml") dans la base qu'on avait chargé depuis le fichier site.xml. Le résultat de ce requête est le suivant

```
<resultats>
<utilisateurs id1="m23" id2="m56">Samsung T1</utilisateurs>
<utilisateurs id1="m23" id2="m62">Nikon F1</utilisateurs>
```

```
<utilisateurs id1="m56" id2="m23">Samsung T1</utilisateurs>  
<utilisateurs id1="m62" id2="m23">Nikon F1</utilisateurs>  
</resultats>
```

Réponse

Les couples d'utilisateurs qui ont pris des photos utilisant les mêmes appareils.