

TD et TME : XQuery

L'objectif de ce TD/TME est de connaître XQuery le langage d'interrogation de données XML.

Préparation : Lire la documentation en ligne (lien TME Xquery).

Préparation :

Lire la documentation en ligne (lien TME XQuery depuis la page d'accueil).

Exercice 1 :

1) Requêtes

On utilise le langage d'interrogation XQuery pour consulter un guide touristique au format XML :

Le document **guide.xml** est :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<base>
  <restaurant nom="la tour d'argent" etoile="3" ville="Paris">
    <fermeture>dimanche et lundi</fermeture>
    <menu nom="buffet" prix="200" />
    <menu nom="gourmet" prix="300"/>
  </restaurant>

  <ville nom="Paris" departement="75">
    <plusBeauMonument nom="tour Eiffel" tarif="30"/>
  </ville>

  <restaurant nom="chez Bocuse" etoile="2" ville="Lyon">
    <fermeture>dimanche</fermeture>
    <menu nom="specialites lyonnaises" prix="200"/>
    <menu nom="vegetarien" prix="299"/>
  </restaurant>

  <restaurant nom="MacDo" etoile="0" ville="Avignon">
    <menu nom="standard" prix="35"/>
    <menu nom="enfant" prix="20"/>
    <menu nom="big" prix="49"/>
  </restaurant>

  <ville nom="Lyon" departement="69">
  </ville>

  <ville nom="Avignon" departement="84">
    <plusBeauMonument nom="le pont" tarif="0"/>
  </ville>

  <restaurant nom="Les 4 saisons" etoile="1" ville="Paris">
    <fermeture>mardi</fermeture>
    <menu nom="assiette printaniere" prix="50" />
    <menu nom="salade d'ete" prix="30"/>
    <menu nom="menu d'automne" prix="150"/>
    <menu nom="menu d'hiver" prix="99"/>
  </restaurant>

  <ville nom="Brest" departement="29">
    <plusBeauMonument nom="oceanopolis" tarif="95"/>
  </ville>
</base>
```

Définir en XQuery les requêtes suivantes (une seule requête par fichier).

1.1) req1.xql : Donner le nom de tous les menus des restaurants

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>
```

1.2) req2.xql : Donner le nom et le prix de tous les menus dont le prix est inférieur à 100.

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>
```

1.3) req3.xql : Donner le nom des restaurants 2 étoiles avec leur nom de menu

Résultat:

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

1.4) req4.xql : Donner le nom de chaque restaurant avec son numéro de département

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>
```

1.5) req5.xql : 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.

Résultat:

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

1.6) Pour chaque requête, donner la DTD pour valider le résultat (fichier reqN.dtd).

2) *Présentation de données*

2.1) 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.

2.2) 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).

Exercice 2

Les données manipulées dans cet exercice sont représentées en XML. Il y a 4 types de données : les références bibliographiques (bib), les critiques (review) et les prix comparatifs (price) et les livres (book). Les données sont définies comme suit :

Les références bibliographiques :

Le document **bib.xml** est :

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price> 65.95</price>
  </book>

  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>

  <book year="2000">
    <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>
    <publisher>Morgan Kaufmann Publishers</publisher>
    <price> 39.95</price>
  </book>

  <book year="1999">
    <title>The Economics of Technology and Content for Digital TV</title>
    <editor>
      <last>Gerbarg</last><first>Darcy</first>
      <affiliation>CITI</affiliation>
    </editor>
    <publisher>Kluwer Academic Publishers</publisher>
    <price>129.95</price>
  </book>
</bib>
```

Les critiques de livres

Le document **review.xml** est :

```
<reviews>
  <entry>
    <title>Data on the Web</title>
    <price>34.95</price>
```

```

<review>
  A very good discussion of semi-structured database
  systems and XML.
</review>
</entry>
<entry>
  <title>Advanced Programming in the Unix environment</title>
  <price>65.95</price>
  <review>
    A clear and detailed discussion of UNIX programming.
  </review>
</entry>
<entry>
  <title>TCP/IP Illustrated</title>
  <price>65.95</price>
  <review>
    One of the best books on TCP/IP.
  </review>
</entry>
</reviews>

```

Les prix comparatifs :

La DTD **price.dtd** est :

```

<!ELEMENT prices (book*)>
  <!ELEMENT book (title, source, price)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT source (#PCDATA)>
  <!ELEMENT price (#PCDATA)>

```

Le document **price.xml** est :

```

<prices>
  <book>
    <title>Advanced Programming in the Unix environment</title>
    <source>www.amazon.com</source>
    <price>65.95</price>
  </book>
  <book>
    <title>Advanced Programming in the Unix environment </title>
    <source>www.bn.com</source>
    <price>65.95</price>
  </book>
  <book>
    <title>TCP/IP Illustrated</title>
    <source>www.amazon.com</source>
    <price>65.95</price>
  </book>
  <book>
    <title>TCP/IP Illustrated</title>
    <source>www.bn.com</source>
    <price>65.95</price>
  </book>
  <book>
    <title>Data on the Web</title>
    <source>www.amazon.com</source>
    <price>34.95</price>
  </book>
  <book>
    <title>Data on the Web</title>
    <source>www.bn.com</source>
    <price>39.95</price>
  </book>
</prices>

```

Les livres :

La DTD **book.dtd** est :

```

<!ELEMENT chapter (title, section*)>
<!ELEMENT section (title, section*)>
<!ELEMENT title (#PCDATA)>

```

Le document **book.xml** est :

```

<chapter>
  <title>Data Model</title>
  <section>
    <title>Syntax For Data Model</title>

```

	<pre> </section> <section> <title>XML</title> <section> <title>Basic Syntax</title> </section> <section> <title>XML and Semistructured Data</title> </section> </section> </chapter> </pre>
--	---

Ecrire les requêtes suivantes en XQuery

R1 : Sélection

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

Résultat:

```

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

```

R2 : Structure plate et désimbriquée

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

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>
```

R3 : Structure hiérarchique

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

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>
```

R4 : Auto jointure d'un document

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.

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>

```

R5 : Jointure entre deux documents

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

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>

```

R6 : Expression conditionnelle

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*).

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>

```

R7 : Tri

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

Résultat:

```

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

```

R8 : Comparaison de chaînes de caractères

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

Résultat:

```

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

```

R9 : Recherche plein texte

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

La DTD du livre *book.xml* est

```

<!ELEMENT chapter (title, section*)>
  <!ELEMENT section (title, section*)>
  <!ELEMENT title (#PCDATA)>

```

R10 :

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>.

Résultat:

```

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

```



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

R11 :

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.

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>
```

R12 :

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

Résultat:

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

Exercice 3. Extrait de l'examen de Juin2003

Les données manipulées dans cet exercice sont représentées en XML. Il y a deux types de données : des joueurs de tennis et des résultats de rencontres. Les données sont définies comme suit :

Les joueurs de tennis

Le document **joueur.xml** est :

```
<joueurs>

<joueur>
  <identite>
    <nom>Hewitt</nom>
    <prenom>Lleyton</prenom>
    <nationalite>AUS</nationalite>
  </identite>
  <classement>1</classement>
</joueur>

<joueur>
  <identite>
    <nom>Arthurs</nom>
    <prenom>Wayne</prenom>
    <nationalite>AUS</nationalite>
  </identite>
  <classement>55</classement>
</joueur>

<joueur>
  <identite>
    <nom>Mathieu</nom>
    <prenom>Paul-Henry</prenom>
    <nationalite>FR</nationalite>
  </identite>
  <classement>44</classement>
</joueur>

<joueur>
  <identite>
    <nom>Clement</nom>
    <prenom>Arnaud</prenom>
    <nationalite>FR</nationalite>
  </identite>
  <classement>37</classement>
</joueur>

<joueur>
  <identite>
    <nom>Costa</nom>
    <prenom>Albert</prenom>
    <nationalite>ESP</nationalite>
  </identite>
  <classement>8</classement>
</joueur>

<joueur>
  <identite>
    <nom>Gasquet</nom>
    <prenom>Richard</prenom>
    <nationalite>FR</nationalite>
  </identite>
  <classement>124</classement>
</joueur>

</joueurs>
```

Les rencontres

Le document **rencontre.xml** est :

```
<rencontres>

<rencontre>
  <joueur1>
    <identite>
      <nom>Hewitt</nom>
      <prenom>Lleyton</prenom>
      <nationalite>AUS</nationalite>
    </identite>
  </joueur1>
  <joueur2>
    <identite>
      <nom>Arthurs</nom>
      <prenom>Wayne</prenom>
      <nationalite>AUS</nationalite>
    </identite>
  </joueur2>
  <gagnant>1</gagnant>
</rencontre>

<rencontre>
  <joueur1>
    <identite>
      <nom>Arthurs</nom>
      <prenom>Wayne</prenom>
      <nationalite>AUS</nationalite>
    </identite>
  </joueur1>
  <joueur2>
    <identite>
      <nom>Hewitt</nom>
      <prenom>Lleyton</prenom>
      <nationalite>AUS</nationalite>
    </identite>
  </joueur2>
  <gagnant>1</gagnant>
</rencontre>

<rencontre>
  <joueur1>
    <identite>
      <nom>Hewitt</nom>
      <prenom>Lleyton</prenom>
      <nationalite>AUS</nationalite>
    </identite>
  </joueur1>
  <joueur2>
    <identite>
      <nom>Mathieu</nom>
      <prenom>Paul-Henry</prenom>
      <nationalite>FR</nationalite>
    </identite>
  </joueur2>
  <gagnant>2</gagnant>
</rencontre>

<rencontre>
  <joueur1>
    <identite>
      <nom>Clement</nom>
      <prenom>Arnaud</prenom>
      <nationalite>FR</nationalite>
    </identite>
  </joueur1>
  <joueur2>
    <identite>
      <nom>Mathieu</nom>
      <prenom>Paul-Henry</prenom>
```

```
<nationalite>FR</nationalite>
</identite>
</joueur2>
<gagnant>2</gagnant>
</rencontre>

</rencontres>
```

Ecrire les requêtes suivantes en XQuery :

R1 : Structure aplatie et désimbriquée

Lister les joueurs avec leur classement et ordonnés par nationalité.

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>
```

R2 : Rebalisage

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

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>
```

R3 : Agrégation

Donner le nombre de joueurs par nationalité.

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>
```

R4 : Jointure

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

Résultat :

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