

Prise de contact avec XQuery

Cours XML - TP 5

1 Objectif

Le but de ce TP est de vous familiariser avec le langage XQUERY. Vous allez ensuite interroger un ensemble de documents XML avec XQUERY.

2 Schéma de données utilisés

2.1 La base de données épicerie

La structure de la base utilisée est la suivante (la clé primaire de chaque table est en gras) :

```
fournisseur(F:chaîne, Nom:chaîne, Remise:réel, Ville:chaîne)
mafourniture(F:chaîne, P:chaîne, Qte:réel)
produit(P:chaîne, Nom_P:chaîne, Couleur:chaîne, Origine:chaîne)
```

Voici le schéma entité-relation de cette base :

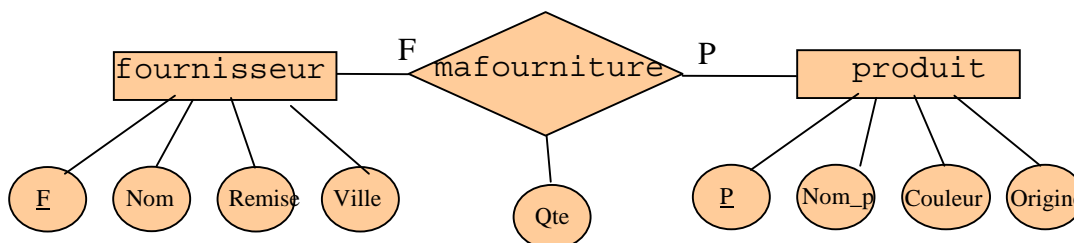


FIGURE 1. schéma entité-relation de la base épicerie

2.2 Ressources pour ce TP

Pour ce tp la base vous est fournie en XML sous deux formes :

1. Un grand document XML unique : `epicerie.xml`
2. Un ensemble de trois documents xml séparés avec une table par document : `fournisseur.xml`, `maFourniture.xml`, `produit.xml`
3. Un fichier de commandes ANT pour lancer les transformations XQUERY à partir d'ECLIPSE.

Voici un extrait du contenu du document unique `epicerie.xml` :

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

```

        <fournisseur>
            <F>f1</F>
            <Nom>Barnibus</Nom>
            <Remise>0,05</Remise>
            <Ville>Paris</Ville>
        </fournisseur>
        <fournisseur>
            <F>f2</F>
        ...
            <Ville>Vannes</Ville>
        </fournisseur>
    </listeFournisseur>
    <listeFourniture>
        <fourniture>
            <F>f1</F>
            <P>p1</P>
            <Qte>1</Qte>
        </fourniture>
        <fourniture>
            <F>f2</F>
        ...
            <Qte>10</Qte>
        </fourniture>
    </listeFourniture>
    <listeProduit>
        <produit>
            <P>p1</P>
            <Nom_p>cassis</Nom_p>
            <Couleur>rouge</Couleur>
            <Origine>Dijon</Origine>
        </produit>
        <produit>
            <P>p2</P>
        ...
            <Origine>Nantes</Origine>
        </produit>
    </listeProduit>
</epicerie>

```

Les éléments racine de chacun des trois fichiers XML séparés sont les trois éléments `listeXxxx` visualisés ci-dessus.

2.3 Utilisation de Eclipse pour ce tp

Les ressources décrites ci-dessus ainsi qu'un fichier de commande ANT `build.xml` vous sont fournies sur le réseau de l'IFSIC sous share à l'endroit habituel pour les ressources de votre formation

2.3.1 Mode de travail

Comme processeur XQUERY vous utiliserez SAXON_X. ($X > 6$). Vous travaillerez sous ECLIPSE qui va vous servir

- d'éditeur de texte pour saisir vos commandes XQUERY.
- de moyen de lancement des exécutions XQUERY à l'aide du programme de commandes ANT qui vous est fourni.
- de moyen de visualisation des résultats produits à l'aide de l'aide de l'éditeur XML fourni par le plugin WTP.
- de moyen de visualisation des messages d'erreur de syntaxe ou d'exécution qui ne manqueront pas d'arriver. Ils apparaîtront dans la vue **console** de votre plan de travail.

Chaque question *I* sera tapée dans un fichier séparé de nom *qI.xqy*. L'exécution de la tâche ANT produira un fichier en sortie *qI.xml*. Il sera créé automatiquement s'il n'existe pas déjà, sinon l'exécution écrasera l'ancienne version. Vous devrez ouvrir ce fichier pour le visualiser. Attention la vue navigation d'ECLIPSE ne se met pas à jour automatiquement. Il faut donc actualiser la vue navigation de votre projet à chaque fois qu'un nouveau fichier est créé par l'exécution de la tâche ANT.

Démarrage Lancer ECLIPSE. Créer un projet **général**. Importer les ressources dans ce projet. Ouvrez le fichier `build.xml`. Modifiez si nécessaire la propriété `saxon` de ce fichier. Elle doit refléter l'installation actuelle de SaxonXXX.

```
<property name="saxon" value="../../../service-xml/saxonXXX"/>
```

Dans ce fichier une autre ligne vous concerne. Il s'agit de la ligne qui définit le préfixe courant de fichier correspondant à la question sur laquelle vous travaillez couramment :

```
<property name="prefixe" value="q1"/>
```

Vous devrez modifier cette propriété et sauvegarder la modification de ce fichier de commande à chaque fois que vous changerez de question, avant de lancer une exécution.

Laissez le fichier `build.xml` ouvert pour sa modification ultérieure.

Créez un nouveau fichier `q1.xqy` dans votre projet (**menu contextuel du projet>nouveau>fichier**). Vous êtes prêt à travailler, il vous suffit de taper le texte de votre commande dans la fenêtre d'édition de ce fichier, de sauvegarder ce texte et de lancer une exécution (voir ci-après).

Préparation de l'exécution. Ouvrir une vue ANT par le menu **fenêtre>afficher la vue>Ant>ant**. Dans la vue ANT, afficher le fichier de commande Ant. Pour cela utiliser le menu **afficher les fichiers de compilation à la recherche** (la petite lampe torche). Une fenêtre d'aide s'ouvre, entrez le nom `build-xquery.xml`, rechercher. Ouvrir le fichier. Le fichier `build.xml` apparaît dans la vue ANT. Déployer son Icône, la tâche `saxonXquery` apparaît. comme ci-dessous :

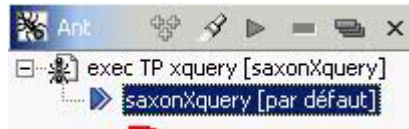


FIGURE 2. La tâche SaxonXquery dans la vue Ant

Pour exécuter cette tâche il vous suffit de double cliquer dessus.

Personnalisation de votre plan de travail Pour ce tp nous vous suggérons la personnalisation du plan de travail suivante :

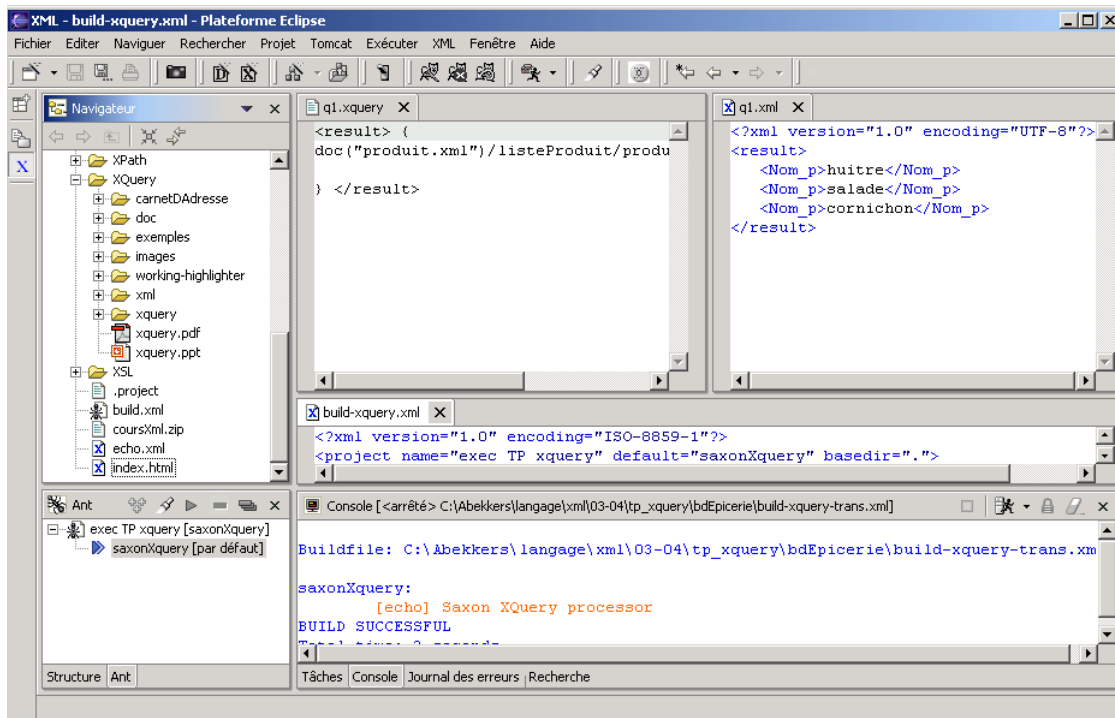


FIGURE 3. Plan de travail personnalisé

En haut, de gauche à droite, on trouve la vue navigation, l'édition du fichier `q1.xquery`, la visualisation du fichier `q1.xml` produit. Au centre gauche, on trouve l'édition du fichier de commande `build-xquery.xml`. En bas, de gauche à droite, on trouve la vue **Ant** et la vue **console** pour les erreurs. Vous pouvez mémoriser cette personnalisation et la retrouver ultérieurement : **menu fenêtre>sauver la perspective sous ...**, donnez un nom, `monXQuery` par exemple, sauver. Vous retrouverez cette perspective par le menu **fenêtre>ouvrir la perspective>monXQuery**.

3 Interrogation XQuery

Répondez aux questions suivantes :

Question q1

Liste des produits verts

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <Nom_p>huitre</Nom_p>
  <Nom_p>salade</Nom_p>
  <Nom_p>cornichon</Nom_p>
</result>
```

Question q2

Nombre de produits verts

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>3</result>
```

Question q3

Ecrire la fonction `produitParCouleur(couleur)` qui prend un paramètre de type qui est une couleur, qui rend les éléments `<produit>` de la couleur considérée en les extrayant du fichier "produit.xml". Le résultat de l'évaluation de l'expression suivante :

```
<result> {
  local:produitParCouleur('vert')/Nom_p
} </result>
```

donne en résultat la liste des noms des produits verts.

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <Nom_p>huitre</Nom_p>
  <Nom_p>salade</Nom_p>
  <Nom_p>cornichon</Nom_p>
</result>
```

Question q4

Nombre de produits verts de Riec (avec ou sans utiliser la fonction ci-dessus)

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>1</result>
```

Question q5

Code des fournisseurs de produits verts sans doublons

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <F>f4</F>
  <F>f5</F>
  <F>f1</F>
</result>
```

Question q6

Noms des produits verts fournis par Barnibus

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <produit>salade</produit>
  <produit>cornichon</produit>
</result>
```

Question q7

Numéro des fournisseurs fournissant au moins les produits de code p2 et p4

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <F>f2</F>
  <F>f3</F>
</result>
```

Question q8

Ecrire la fonction fournisseurParProduit(codep) qui prend un paramètre de type string qui est un code de produit et qui rend en résultat un ensemble d'éléments qui sont les fournisseurs qui fournissent le produit considéré. Le résultat de l'évaluation de l'expression suivante :

```
<result> {
  local:fournisseurParProduit('p4')/Nom
} </result>
```

donne en résultat la liste des noms des fournisseurs de moutarde (codep='p4').

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <Nom>Barnibus</Nom>
  <Nom>Mercier</Nom>
  <Nom>Colbert</Nom>
  <Nom>Bossuet</Nom>
</result>
```

Question q9

Liste de tous les produits avec chacun leur liste de fournisseurs

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <produit nom="cassis">
    <Nom>Barnibus</Nom>
  </produit>
  <produit nom="champagne">
    <Nom>Mercier</Nom>
    <Nom>Colbert</Nom>
  </produit>
  <produit nom="huitre">
    <Nom>Bossuet</Nom>
    <Nom>Tanguy</Nom>
  </produit>
  <produit nom="moutarde">
    <Nom>Barnibus</Nom>
    <Nom>Mercier</Nom>
    <Nom>Colbert</Nom>
    <Nom>Bossuet</Nom>
  </produit>
  <produit nom="salade">
    <Nom>Barnibus</Nom>
    <Nom>Bossuet</Nom>
  </produit>
  <produit nom="cornichon">
    <Nom>Barnibus</Nom>
    <Nom>Bossuet</Nom>
  </produit>
  <produit nom="muscadet"/>
</result>
```

Question q10

Pour chaque origine de produit donnez la liste des produits fournis avec la liste des noms de fournisseur

Résultat attendu

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <origine ville="Dijon">
    <produit>
      <nom>cassis</nom>
      <fournisseur>Barnibus</fournisseur>
    </produit>
    <produit>
      <nom>moutarde</nom>
      <fournisseur>Barnibus</fournisseur>
      <fournisseur>Mercier</fournisseur>
      <fournisseur>Colbert</fournisseur>
      <fournisseur>Bossuet</fournisseur>
    </produit>
    <produit>
      <nom>cornichon</nom>
      <fournisseur>Barnibus</fournisseur>
      <fournisseur>Bossuet</fournisseur>
    </produit>
  </origine>
  <origine ville="Reims">
    <produit>
      <nom>champagne</nom>
      <fournisseur>Mercier</fournisseur>
      <fournisseur>Colbert</fournisseur>
    </produit>
  </origine>
  <origine ville="Riec">
    <produit>
      <nom>huitre</nom>
      <fournisseur>Bossuet</fournisseur>
      <fournisseur>Tanguy</fournisseur>
    </produit>
  </origine>
  <origine ville="Nice">
    <produit>
      <nom>salade</nom>
      <fournisseur>Barnibus</fournisseur>
      <fournisseur>Bossuet</fournisseur>
    </produit>
  </origine>
  <origine ville="Nantes">
    <produit>
      <nom>muscadet</nom>
    </produit>
  </origine>
</result>
```