

BDA

mardi 21 juin 2011

durée 3h

support de cours autorisé

Examen – session 2

Exercice 1 :

On désire écrire une DTD pour des documents décrivant des familles. Une famille porte un nom et est constituée d'une ou plusieurs personnes. Pour chaque personne de la famille, on a le prénom, l'âge, le poids en kilos(kg) ou le poids en livres (lb), et éventuellement la taille. Les liens de parenté (père et mère) sont gérés grâce à des attributs de type ID et IDREF.

Question 1 : Écrire la DTD correspondante `famille.dtd`. Celle-ci devra accepter par exemple le document suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE famille SYSTEM "famille.dtd">
<famille>
  <nom>Martin</nom>
  <personne pnumber="a1">
    <prenom>Juliette</prenom>
    <age>30</age>
    <poids-kg>58</poids-kg>
  </personne>
  <personne pnumber="a2">
    <prenom>Romeo</prenom>
    <age>31</age>
    <poids-lb>97</poids-lb>
  </personne>
  <personne pnumber="a3" mere="a1" pere="a2">
    <prenom>Max</prenom>
    <age>4</age>
    <poids-kg>12</poids-kg>
    <taille>1.25</taille>
  </personne>
  <personne pnumber="a4" mere="a1" pere="a2">
    <prenom>Marie</prenom>
    <age>3</age>
    <poids-lb>18</poids-lb>
    <taille>1.10</taille>
  </personne>
</famille>
```

L'attribut `pnumber` est obligatoire.

Question 2 : Écrire un XML-Schema correspondant.

Exercice 2 : On considère le fichier XML, de nom `liste.xml`, ci-dessous :

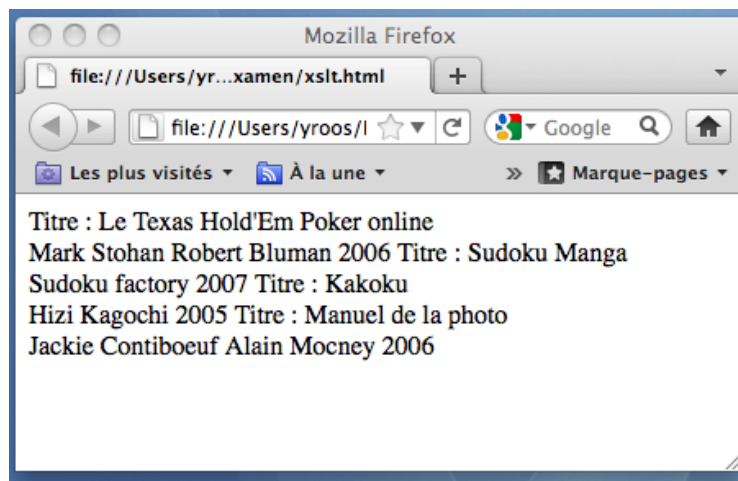
```
<?xml version="1.0" encoding="UTF-8"?>
<liste>
  <livre>
    <titre genre="jeu">Le Texas Hold'Em Poker online</titre>
    <auteur>Mark Stohan</auteur>
    <auteur>Robert Bluman</auteur>
    <parution>2006</parution>
  </livre>
  <livre>
    <titre genre="jeu">Sudoku Manga</titre>
    <auteur>Sudoku factory</auteur>
    <parution>2007</parution>
  </livre>
  <livre>
    <titre genre="jeu">Kakoku</titre>
    <auteur>Hizi Kagochi</auteur>
    <parution>2005</parution>
  </livre>
  <livre>
    <titre genre="photo">Manuel de la photo</titre>
    <auteur>Jackie Contiboeuf</auteur>
    <auteur>Alain Mocney</auteur>
    <parution>2006</parution>
  </livre>
</liste>
```

On considère également la feuille XSLT suivante :

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

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="titre">
    Titre : <xsl:value-of select="."/>
    <br/>
  </xsl:template>
</xsl:stylesheet>
```

Question 1 : La visualisation du fichier HTML produit par cette feuille de style correspond-elle à la capture d'écran ci-dessous ? Expliquer pourquoi.



Question 2 : Donner des requêtes XPATH permettant de :

1. sélectionner tous les nœuds éléments **titre** présents dans le document ;
2. sélectionner tous les nœuds attributs **genre** ;
3. sélectionner tous les nœuds **titre** correspondant à des livres du genre **jeu** ;
4. sélectionner le nœud **titre** du premier livre qui est du genre **photo** ;
5. sélectionner tous les nœuds descendants du deuxième livre ;
6. sélectionner tous les nœuds **titre** des frères suivant le premier livre.
7. retourner le nombre d'auteurs qui ont écrit le quatrième livre ;
8. sélectionner le dernier nœud **livre** du genre **jeu** ;
9. sélectionner le nœud **titre** du deuxième livre paru en 2006.

Exercice 3 :

On considère les trois fichiers **xml** suivants, donnés en annexe

plant-catalog.xml est un catalogue de plantes ;

plant-families.xml qui indique à quelle famille appartiennent certaines plantes ;

plant-order.xml est une commande de plantes.

Question 1 : Donner une transformation **XSLT** qui produit à partir des fichiers **plant-catalog.xml** et **plant-families.xml** un document **XML** en ajoutant dans chaque élément **PLANT** apparaissant dans **plant-catalog.xml** un élément **FAMILY** qui donne le nom de la famille à laquelle appartient la plante comme dans l'exemple ci-dessous :

```
<PLANT>
  <COMMON>Bloodroot</COMMON>
  <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
  <ZONE>4</ZONE>
  <LIGHT>Mostly Shady</LIGHT>
  <PRICE>$2.44</PRICE>
  <AVAILABILITY>031599</AVAILABILITY>
  <FAMILY>Papaveraceae</FAMILY>
</PLANT>
```

Question 2 : Donner une transformation XSLT qui classe et regroupe les éléments PLANT du fichier `plant-catalog.xml` en fonction du contenu de leur élément LIGHT comme dans l'exemple ci-dessous :

```
<CATALOG>
  <LIGHT>
    <EXPOSURE>Mostly Shady</EXPOSURE>
    <PLANT>
      <COMMON>Bloodroot</COMMON>
      <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
      <ZONE>4</ZONE>
      <PRICE>$2.44</PRICE>
      <AVAILABILITY>031599</AVAILABILITY>
    </PLANT>
    <PLANT>
      <COMMON>Columbine</COMMON>
      <BOTANICAL>Aquilegia canadensis</BOTANICAL>
      <ZONE>3</ZONE>
      <PRICE>$9.37</PRICE>
      <AVAILABILITY>030699</AVAILABILITY>
    </PLANT>
    (...)
    <EXPOSURE>Mostly Sunny</EXPOSURE>
    <PLANT>
      <COMMON>Marsh Marigold</COMMON>
      <BOTANICAL>Caltha palustris</BOTANICAL>
      <ZONE>4</ZONE>
      <PRICE>$6.81</PRICE>
      <AVAILABILITY>051799</AVAILABILITY>
    </PLANT>
  </LIGHT>
</CATALOG>
```

Question 3 : Donner une transformation XSLT qui calcule le montant total de la commande décrite dans `plant-order.xml` en donnant le résultat dans un élément PRICE comme dans l'exemple : `<PRICE>663.2</PRICE>`

Exercice 4 :

Voici de nouveau le fichier de description de logements déjà vu en TP :

```
<?xml version="1.0" encoding="UTF-8"?>
<maisons>
  <maison id="1">
    <RDC>
      <cuisine surface-m2="12">Evier Inox. Mobilier encastre</cuisine>
      <WC>Lavabo.</WC>
      <sejour>Cheminee en pierre. Baie vitree</sejour>
      <bureau surface-m2="14">Bibliotheque</bureau>
      <garage/>
    </RDC>
    <etage>
      <terrasse/>
      <chambre surface-m2="28" >alcove surface-m2="8"/></chambre>
```

```

    <chambre surface-m2="18"/>
    <sallededeBain surface-m2="15">
      Douche, baignoire, lavabo
    </sallededeBain>
  </etage>
</maison>
<maison id="2">
  <RDC>
    <cuisine surface-m2="12">en ruine</cuisine>
    <garage/>
  </RDC>
  <etage>
    <mirador surface-m2="1">
      Vue sur la mer
    </mirador>
    <sallededeBain surface-m2="15">Douche</sallededeBain>
  </etage>
</maison>
<maison id="3">
  <RDC>
    <sejour surface-m2="40"/>
  </RDC>
  <etage>
    <chambre surface-m2="17.5">Exposition plein sud</chambre>
  </etage>
</maison>
</maisons>

```

Question 1 : Écrire un programme XQuery qui, à partir de ce fichier XML, calcule, pour chaque maison, sa superficie totale. La sortie du programme sera un fichier HTML dont la visualisation correspond à la capture écran suivante :

Maisons	Surfaces (m2)
Maison 1	87
Maison 2	28
Maison 3	57.5

Exercice 5 : On considère le fichier XML ci-dessous qui représente le stock d’un maraîcher :

```
<?xml version="1.0" encoding="UTF-8"?>
<produits>
  <fruit type="clementine" prix="290" calibre="1">
    <producteur>Production Bastia</producteur>
    <origine region="Corse">France</origine>
    <qty>15</qty>
    <note>Sans pepins , avec feuilles</note>
    <bio/>
  </fruit>
  <fruit type="clementine" prix="290" calibre="1">
    <producteur>Production Bastia</producteur>
    <origine region="Corse">France</origine>
    <qty>15</qty>
    <note>Sans pepins , avec feuilles</note>
  </fruit>
  <legume type="courgette" prix="300" calibre="2">
    <producteur>Madrid Hortelano</producteur>
    <origine>Espagne</origine>
    <qty>100</qty>
    <bio/>
  </legume>
  <legume type="chou fleur" prix="090" calibre="2">
    <producteur>Pontivy et Cie</producteur>
    <origine region="Bretagne">France</origine>
    <qty>100</qty>
  </legume>
  <legume type="salade" prix="075" calibre="3">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qty>35</qty>
    <note>Batavia</note>
  </legume>
  <fruit type="melon" prix="150" calibre="1">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qty>50</qty>
    <note>Melon brode</note>
    <bio/>
  </fruit>
</produits>
```

Question 1 : Donner trois requêtes XQuery Update Facility pour¹

1. supprimer tous les producteurs de fruits.
2. transformer tous les attributs **type** en sous-éléments **type**.
3. supprimer l’attribut **region** des origines des clémentines de calibre 1 issues de l’agriculture biologique.

Question 2 : On souhaite réorganiser le fichier XML initial pour placer d’abord les légumes, puis les fruits. L’ensemble des articles est encore placé dans un élément **produits**. Proposer une requête XQuery Update Facility réalisant cette mise à jour.

1. ces requêtes doivent bien sûr fonctionner sur tout document de même nature que celui de l’exemple.