

BDA

samedi 18 mai 2013

durée 3h

supports de cours et TD autorisés

Examen – session 1

Tous les fichiers mentionnés dans le sujet sont consultables dans l'annexe située à la fin. Il peut être commode de détacher les trois feuilles qui composent cette annexe afin d'avoir sous les yeux le contenu des fichiers concernés par chaque question.

1 XML-Schema

On considère des données XML décrivant des personnes (nom, prénom, sexe, date de naissance et éventuellement date de décès). Un tel exemple de donnée est fourni dans le fichier `personnes.xml` de la figure 1. Toutes ces données satisfont un schéma, décrit dans un fichier de nom `personnes.xsd` et dont le contenu (incomplet) est donné figure 2.

Question 1 : Donner la définition manquante de l'élément `personne` du schéma `personnes.xsd`.

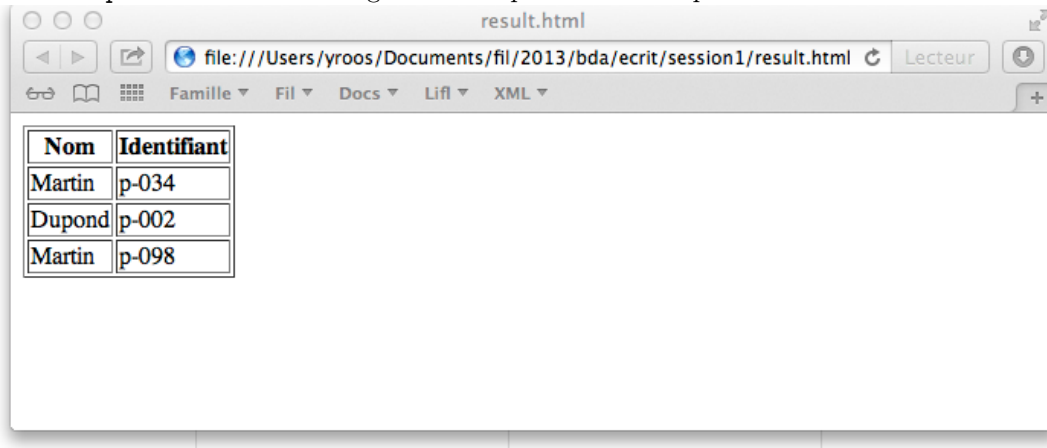
Question 2 : Donner la définition manquante de l'élément `personnes` du schéma `personnes.xsd`.

2 XSLT

Question 3 : On considère le programme XSLT suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:bda="http://www.fil.univ-lille1.fr/bda">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="/">
    <html>
      <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
      <body>
        <table border="1">
          <tr><th>Nom</th><th>Identifiant</th></tr>
          <xsl:apply-templates/>
        </table>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="personne">
    <tr>
      <td><xsl:value-of select="nom"/></td>
      <td><xsl:value-of select="@id"/></td>
    </tr>
  </xsl:template>
</xsl:stylesheet>
```

La visualisation dans un navigateur du fichier HTML produit par ce programme quand on l’applique sur le fichier `personnes.xml` de la figure 1 correspond-elle à la capture d’écran ci-dessous ? Justifiez.



Question 4 : On souhaite définir un programme XSLT produisant un fichier XML regroupant les identifiants de personnes par ville de naissance. Le résultat attendu pour le document d’entrée `personnes.xml` de la figure 1 est donné ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<naissances xmlns:bda="http://www.fil.univ-lille1.fr/bda">
  <ville>
    <nom>Angers</nom>
    <personne>p-034</personne>
    <personne>p-002</personne>
  </ville>
  <ville>
    <nom>Paris</nom>
    <personne>p-098</personne>
  </ville>
</naissances>
```

Complétez¹ le programme XSLT suivant afin qu’il réponde au cahier des charges en ne contenant qu’une seule règle (`template`).

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="2.0"
  xpath-default-namespace="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/">
    <naissances>
      <xsl:for-each select="...">
        A COMPLETER
      </xsl:for-each>
    </naissances>
  </xsl:template>
</xsl:stylesheet>
```

1. N’écoutez que ce qui manque.

Question 5 : Donnez un programme XSLT, équivalent à celui de la question précédente mais sans aucune boucle `xslt:for-each`, en complétant le programme ci-dessous :

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:bda="http://www.fil.univ-lille1.fr/bda">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/">
    <naissances>
      A COMPLETER
    </naissances>
  </xsl:template>

  A COMPLETER
</xsl:stylesheet>
```

3 XPath-XQuery

On dispose, en plus du document `personnes.xml`, du document `parentés.xml` (visible dans l'annexe, figure 3) qui décrit les liens de parentés entre les personnes du fichier `personnes.xml`. Un lien de parenté est défini par un élément de nom `parenté` comme :

```
<parenté parent="p-034" enfant="p-098"/>
```

qui indique que la personne d'*id* `p-034` est directement parent² de la personne d'*id* `p-098`. Les fichiers de liens de parentés satisfont le schéma décrit dans le fichier `parentés.xsd` de la figure 4.

On souhaite dans un premier temps faire des requêtes permettant de vérifier l'intégrité et la cohérence des données contenues dans un fichier `personnes.xml` et `parentés.xml` (un exemple de données de parentés incohérentes apparaît figure 5) .

Question 6 : Donner des requêtes XPath ou XQuery permettant

1. de lister les identifiants des personnes dont la date de décès précède strictement leur date de naissance.
2. de lister les identifiants de personnes apparaissant dans le fichier `parentés.xml` mais pas dans le fichier `personnes.xml`.
3. de lister les identifiants des personnes qui sont parents (immédiats) d'eux-mêmes.
4. de lister les identifiants des personnes qui ont strictement plus de deux parents immédiats.
5. de lister les liens de parenté redondants. Un lien de parenté est redondant s'il existe deux nœuds différents `parenté` ayant même valeur d'attribut `parent` et même valeur d'attribut `enfant`. Le résultat doit être donné sous la forme d'une liste de nœuds `parenté` elle-même sans redondance.

Question 7 : Pour chacune des expressions FLOWR suivantes, expliquer (en français) ce qu'elle fait et dire si elle est équivalente à une simple expression de chemin (c'est à dire à une expression XPath sans `if`, `for`, `some`, `every`, les prédicats étant bien-sûr eux autorisés). Si la réponse est "oui", donner l'expression XPath correspondante, sinon dire pourquoi ce n'est pas possible :

2. en l'occurrence père puisque la personne d'*id* `p-034` est de sexe masculin.

1.

```
for $x in doc("personnes.xml")//personne
  where exists($x/décès)
return ($x/nom,$x/prénom)
```

2.

```
for $x in doc("personnes.xml")//personne
  where exists($x/décès)
return <décédé>{($x/nom,$x/prénom)}</décédé>
```

3.

```
for $x in doc("personnes.xml")//personne ,
  $y in doc("parentés.xml")//parenté
  where $x/sexe eq "M" and $y/@parent eq $x/@id
return data($y/@enfant)
```

4 XQuery Update Facility

Question 8 : Écrire une requête **XQuery Update Facility** regroupant dans le fichier **personnes.xml** les personnes n'apparaissant dans aucun lien de parenté du fichier **parentés.xml** (que ce soit comme parent ou comme enfant) sous un nouvel élément **sans-parenté**. Les autres personnes doivent être inchangées. Par exemple, cette transformation appliquée sur le fichier **personnes.xml** de la figure 1 avec les informations de parentés du fichier **parentés.xml** de la figure 3 doit donner le fichier de la figure 6.

Question 9 : Écrire une requête **XQuery Update Facility** permettant d'inclure dans le fichier **personnes.xml** les relations de parenté décrites dans le fichier **parentés.xml** en ajoutant autant d'éléments **enfant** que nécessaire aux personnes ayant des enfants. Ces éléments **enfant** doivent apparaître par date de naissance croissante et leur contenu est l'identifiant (l'**id**) de l'enfant. Par exemple, cette transformation appliquée sur le fichier **personnes.xml** de la figure 1 avec les informations de parentés du fichier **parentés.xml** de la figure 3 doit donner le fichier de la figure 7.

5 Fonctions XQuery

Question 10 : Écrire deux versions d'une fonction **XQuery** de nom **descendance**, prenant en argument une valeur d'identifiant d'une personne *p* et retournant la liste des identifiants des personnes descendant (au sens large) de la personne *p*,

1. en supposant que le fichier **parentés.xml** ne contient aucune incohérence ni redondance comme dans la figure 3,
2. en supposant que le fichier **parentés.xml** peut contenir des incohérences ou redondances comme dans la figure 5.

Sur les fichiers donnés en exemple, un appel à **local:descendance("p-034")** doit retourner la liste ("p-034" , "p-098" , "p-123").

```

<?xml version="1.0" encoding="UTF-8"?>
<personnes xmlns="http://www.fil.univ-lille1.fr/bda"
  xsi:schemaLocation="http://www.fil.univ-lille1.fr/bda personnes.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <personne id="p-034">
    <nom>Martin</nom><prénom>Jules</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1918-04-12</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>1973-08-22</date>
      <lieu>Paris</lieu>
    </décès>
  </personne>
  <personne id="p-002">
    <nom>Dupond</nom><prénom>Yvonne</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1922-10-13</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>2000-11-07</date>
      <lieu>Angers</lieu>
    </décès>
  </personne>
  <personne id="p-003">
    <nom>Dupond</nom><prénom>Paul</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1925-01-26</date>
      <lieu>Angers</lieu>
    </naissance>
  </personne>
  <personne id="p-098">
    <nom>Martin</nom><prénom>Henri</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1931-08-14</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
  <personne id="p-123">
    <nom>Martin</nom><prénom>Christine</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1956-09-21</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
</personnes>

```

FIGURE 1 – le fichier `personnes.xml`.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:bda="http://www.fil.univ-lille1.fr/bda">

  <xs:element name="personnes">
    A COMPLETER
  </xs:element>

  <xs:element name="personne">
    A COMPLETER
  </xs:element>

  <xs:simpleType name="type-sexe">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="M"/>
      <xs:enumeration value="F"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="type-événement">
    <xs:sequence>
      <xs:element name="date" type="xs:date"/>
      <xs:element name="lieu" type="xs:NCName"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

FIGURE 2 – le fichier `personnes.xsd`.

```

<?xml version="1.0" encoding="UTF-8"?>
<parentés xsi:schemaLocation="http://www.fil.univ-lille1.fr/bda parentés. →
  ↪ xsd"
  xmlns="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <parenté parent="p-034" enfant="p-098"/>
  <parenté parent="p-002" enfant="p-098"/>
  <parenté parent="p-098" enfant="p-123"/>
</parentés>

```

FIGURE 3 – Un fichier `parentés.xml` ne comportant ni incohérence ni redondance.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.fil.univ-lille1.fr/bda">
  <xs:element name="parentés">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="parenté"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="parenté">
    <xs:complexType>
      <xs:attribute name="parent" use="required" type="xs:NCName"/>
      <xs:attribute name="enfant" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

FIGURE 4 – le fichier `parentés.xsd`.

```

<?xml version="1.0" encoding="UTF-8"?>
<parentés xsi:schemaLocation="http://www.fil.univ-lille1.fr/bda parentes. →
  ↪ xsd"
  xmlns="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <parenté parent="p-034" enfant="p-098"/>
  <parenté parent="p-034" enfant="p-098"/>
  <parenté parent="p-002" enfant="p-097"/>
  <parenté parent="p-123" enfant="p-123"/>
  <parenté parent="p-123" enfant="p-034"/>
  <parenté parent="p-098" enfant="p-123"/>
</parentés>

```

FIGURE 5 – Un fichier `parentés.xml` comportant des incohérences et des redondances.

```

<personnes xmlns="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.fil.univ-lille1.fr/bda personnes.xsd">
  <personne id="p-034">
    <nom>Martin</nom><prénom>Jules</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1918-04-12</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>1973-08-22</date>
      <lieu>Paris</lieu>
    </décès>
  </personne>
  <personne id="p-002">
    <nom>Dupond</nom><prénom>Yvonne</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1922-10-13</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>2000-11-07</date>
      <lieu>Angers</lieu>
    </décès>
  </personne>
  <personne id="p-098">
    <nom>Martin</nom><prénom>Henri</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1931-08-14</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
  <personne id="p-123">
    <nom>Martin</nom><prénom>Christine</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1956-09-21</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
  <sans-parenté>
    <personne id="p-003">
      <nom>Dupond</nom><prénom>Paul</prénom>
      <sexe>M</sexe>
      <naissance>
        <date>1925-01-26</date>
        <lieu>Angers</lieu>
      </naissance>
    </personne>
  </sans-parenté>
</personnes>

```

FIGURE 6 – La transformation de la question 8 appliquée sur le fichier de la figure 1.


```

<personnes xmlns="http://www.fil.univ-lille1.fr/bda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.fil.univ-lille1.fr/bda personnes.xsd">
  <personne id="p-034">
    <nom>Martin</nom><prénom>Jules</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1918-04-12</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>1973-08-22</date>
      <lieu>Paris</lieu>
    </décès>
    <enfant>p-098</enfant>
  </personne>
  <personne id="p-002">
    <nom>Dupond</nom><prénom>Yvonne</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1922-10-13</date>
      <lieu>Angers</lieu>
    </naissance>
    <décès>
      <date>2000-11-07</date>
      <lieu>Angers</lieu>
    </décès>
    <enfant>p-098</enfant>
  </personne>
  <personne id="p-003">
    <nom>Dupond</nom><prénom>Paul</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1925-01-26</date>
      <lieu>Angers</lieu>
    </naissance>
  </personne>
  <personne id="p-098">
    <nom>Martin</nom><prénom>Henri</prénom>
    <sexe>M</sexe>
    <naissance>
      <date>1931-08-14</date>
      <lieu>Paris</lieu>
    </naissance>
    <enfant>p-123</enfant>
  </personne>
  <personne id="p-123">
    <nom>Martin</nom><prénom>Christine</prénom>
    <sexe>F</sexe>
    <naissance>
      <date>1956-09-21</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
</personnes>

```

FIGURE 7 – La transformation de la question 9 appliquée sur le fichier de la figure 1.