

Module Bases de Données et Web

Examen du 20 décembre 2007

Version CORRIGÉE

Les documents sont autorisés – Durée: 2h.

Répondre aux questions sur la feuille du sujet
attendue. Utiliser le dos de la feuille précédente
de la rédaction sera prise en compte. Ecrire à l'encre

dans les cadres appropriés. La taille des cadres sur
la réponse doit être égale à celle du cadre. Le barème est donné
à titre indicatif. La qualité de la réponse n'est pas à dégrader.

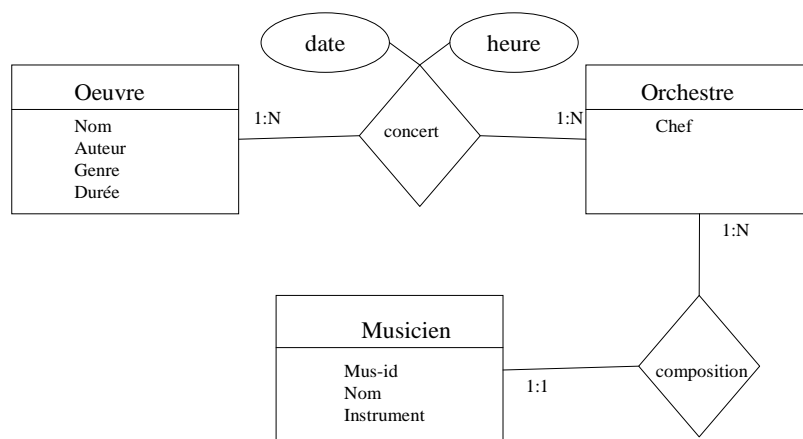
La qualité de la réponse n'est pas à dégrader.

Exercice 1: XML et DTD

4pts

Une association musicale souhaite exporter vers un
les schéma Entité-Association (E/A) est donné ci-dessous.

fichier XML la gestion des salles de concerts, dont
sous.



Question 1 .Ecrire une DTD décrivant cette structure pour le fichier XML. L'élément racine est l'élément *Programme*, et contient des *Concerts*. Les entités E/A (musicien, orchestre, œuvre) seront représentées par des éléments. Les attributs E/A sont représentés par des attributs.

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

(2) <!ELEMENT Programme(

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

(17)

(18)

(19)

(20)

(21)

(22)

(23)

(24)

```

<?xmlversion="1.0"encoding="ISO-8859-1"?>
<!ELEMENTSalles((concert)*)>
<!ELEMENTconcert(orchestre,(œuvre)*)>
<!ATTLISTconcertdateCDATA#REQUIRED
                heureCDATA#REQUIRED>
<!ELEMENTorchestre((musicien)*)>
<!ATTLISTorchestrechefID#REQUIRED>
<!ELEMENTœuvreEMPTY>
<!ATTLISTœuvretitreCDATA#REQUIRED
                auteurCDATA#REQUIRED
                genreCDATA#REQUIRED
                dureeCDATA#IMPLIED>
<!ELEMENTmusicienEMPTY>
<!ATTLISTmusicienid-musID#REQUIRED
                nomCDATA#REQUIRED
                instrumentCDATA#REQUIRED
                joue-dansIDREF#REQUIRED>

```

Question 2. Exprimez (lorsque c'est possible) les contraintes suivantes sur la DTD. (Si la contrainte est déjà exprimée dans votre DTD, indiquez le numéro de ligne qui la définit.

1. Un concert comprend au moins 2 et au plus 5 œuvres

```

<!ELEMENTconcert(orchestre,(œuvre,œuvre,œuvr      e+,œuvre+,œuvre+),)>

```

2. Un concert est identifié de façon unique par la date et l'heure.

impossible

3. Le genre d'une œuvre est 'concerto', 'symphonie' ou 'opéra'.

<!ATTLIST œuvre type("concerto"|"symphonie"|"opéra") "concerto">

4. Un musicien joue dans un et un seul orchestre.

<!ATTLIST musicien joue-dans IDREF#REQUIRED>

Exercice 2: XSchema

6pts

On souhaite maintenant modéliser l'application de l'exercice précédent à l'aide de XSchema. L'élément racine est l'élément *Programme*, qui contient un ensemble de *Concerts*.

Question 1 .

1. Définir le type *TypeOeuvre* pour représenter les Œuvres en utilisant uniquement des attributs. L'attribut *genre* prend les valeurs "concerto", "symphonie" ou "opéra".

```
<xs:complexType name="TypeOeuvre">
  <xs:attribute name="titre" type="xs:string"/>
  <xs:attribute name="auteur" type="xs:string"/>
  <xs:attribute name="genre" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
      <xs:enumeration value="concerto"/>
      <xs:enumeration value="symphonie"/>
      <xs:enumeration value="opera"/>
    </xs:simpleType>
  <xs:attribute name="duree" type="xs:integer"/>
</xs:complexType>
```

2. À partir du type *TypeOeuvre*, définir un type *TypeOeuvreAvecInstruments* contenant la liste des noms des instruments pour lesquels l'œuvre a été écrite. Par exemple, une sonate en trio est écrite pour un violon, un violoncelle et une flûte.

```
<xs:complexType name="TypeOeuvreAvecInstruments">
  <complexContent>
    <xs:extension base="TypeOeuvre">
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="instrument" type="xs:string">
        </xs:sequence>
      <xs:attribute name="titre" type="xs:string" use="required"/>
      <xs:attribute name="auteur" type="xs:string"/>
      <xs:attribute name="genre">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
          <xs:enumeration value="concerto"/>
          <xs:enumeration value="symphonie"/>
          <xs:enumeration value="opera"/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="duree" type="xs:integer" use="required"/>
    </xs:extension>
  </complexContent>
</xs:complexType>
```

3. À partir du type *TypeOeuvreAvecInstrument*, définir le type *Musique de Chambre* qui caractérise les œuvres n'ayant pas plus de 5 instruments.

```

<xs:complexType name="Musique de Chambre">
  <xs:complexContent>
    <xs:restriction base="Type Oeuvre Avec Instrument">
      <xs:sequence maxOccurs="5">
        <xs:element name="instrument" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="titre" type="xs:string" use="required"/>
      <xs:attribute name="auteur" type="xs:string" use="required"/>
      <xs:attribute name="genre" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="concerto"/>
            <xs:enumeration value="symphonie"/>
            <xs:enumeration value="opera"/>
          </xs:simpleType>
        </xs:attribute>
      <xs:attribute name="duree" type="xs:integer" use="required"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

Question 2. Définir l'élément *concert*, en représentant *date* et *heure* par des attributs. On utilisera le type *TypeOrchestre* défini ci-dessous pour l'élément *orchestre*.

```

<xs:complexType name="TypeOrchestre">
  <xs:attribute name="chef" type="xs:string" use="required"/>
</xs:complexType>

```

```

<xs:element name="concert">
  <complexType>
    <xs:sequence>
      <xs:element name="orchestre" type="TypeOrchestre" />
      <xs:element name="œuvre" type="TypeOeuvre" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="date" type="xs:string" use="required" />
    <xs:attribute name="heure" type="xs:integer" use="required" />
  </xs:complexType>
</xs:element>

```

Question 3 . On veut exprimer la contrainte qu'un concert est identifié de façon unique par la date et l'heure.

- a) dans quel élément doit-on définir cette contrainte ?

elle doit être définie à la fin de l'élément Programme

- b) écrivez cette contrainte en Xschema

```
<xs:unique name="cleconcert">
  <xs:selector xpath="/concert"/>
  <xs:field xpath="@date"/>
  <xs:field xpath="@heure"/>
</xs:unique>
```

Question 4 . Définir le type *TypeInstrument* qui est une chaîne de caractère de longueur 50

```
<simpleType name="TypeInstrument">
  <xs:restriction base="xs:string">
    <xs:length value="50"/>
  </xs:restriction>
</xs:simpleType>
```

Question 5 .

Définir l'élément *musicien*, en exprimant l'association *joue-dans* à l'aide d'un élément.


```

<xs:element name="musicien">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="joue-dans" type="xs:string"/>
    </sequence>
    <xs:attribute name="mus-id" type="xs:integer" use="required"/>
    <xs:attribute name="nom" type="xs:string" use="required"/>
    <xs:attribute name="instrument" type="TypeInstrument" use="required"/>
  </xs:complexType>
</xs:element>

```

Question 6. Exprimez lorsque c'est possible les contraintes suivantes:

1. Un concert comprend au moins 2 et au plus 5 œuvres

```

<xs:element name="œuvre" type="TypeOeuvre" minOccurs="2" maxOccurs="5"/>

```

2. Les pianistes jouent dans un et un seul orchestre. (On suppose que l'élément *Orchestre* a une clef nommée *clé orchestre* dont la valeur est le nom du chef)

```

<xs:sequence>
  par défaut minOccurs="1" maxOccurs="1"
  <xs:element name="joue-dans" type="xs:string"/>

```

```
</sequence>nesuffit pas car il n'edit que le string qui est un orchestre.
```

On peut aussi utiliser keyref (à définir dans l'élément racine):

```
<xs:keyref name="un seul orchestre" keyref="cleorchestre">
<xs:selector xpath="//musicien[instrument='piano'] /joue-dans"/>
<xs:field xpath="."/>
</xs:keyref>
```

Un orchestre a un et un seul pianiste

Exercice 3: XPath

5pts

Soit la DTD d'un catalogue de disques de musique ci-dessous:

```
<!ELEMENT listeCD ((CD)+)>
<!ELEMENT CD (titre, compositeur, (morceau)+, editeur, (durée)?)>
<!ELEMENT morceau (titre, (soliste)*, (orchestre, chef d'orchestre)?)>
<!ELEMENT soliste (Nom, Prenom, age, instrument)>
<!ELEMENT compositeur (Nom, Prenom)>
<!ELEMENT chef d'orchestre (Nom, Prenom, age)>
<!ELEMENT editeur (#PCDATA)>
<!ELEMENT durée (#PCDATA)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT orchestre (#PCDATA)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
<!ELEMENT Instrument (#PCDATA)>
```

Question 1. Exprimez en XPath les requêtes suivantes:

1. Tous les titres de morceaux interprétés avec un et un seul soliste et comportant le mot "symphonie".

1 point

//morceau[soliste][not(soliste[2])]/titre[contains(., "symphonie")]

2. Prénom des chefs d'orchestre ayant dirigé le pianiste soliste "Raymond Charles" (le prénom est Raymond)

1 point

//morceau[soliste[instrument="piano"andprenom="Raymond"andnom="Charles"]]/chefdorchestre/prenom

3. Prénom des solistes ayant joué avec l'orchestre "OMPF" (Orchestre de Mélodicas de la Place des Fêtes) sur un disque édité par Syno.

1 point

//CD[editeur="Syno"]/morceau[orchestre="OMPF"]/soliste/Prenom

4. Titre des CD où le pianiste Raymond Charles interprète plus de morceaux que l'OMPF.

1 point

//CD[count(morceau[soliste[nom="Raymond"andprenom="Charles"andinstrument="piano"]])

>count(morceau[orchestre="OMPF"])]/titre

4.

5. Titre des CD comportant au moins un morceau dirigé par un chef d'orchestre plus jeune que la moyenne des solistes du même morceau.

1 point

//CD[morceau[avg(soliste/age)>chefdorchestre/age]]/titre

Exercice 4: XQuery**5pts**

Soit le fichier XML book.xml suivant. Ce fichier contient un seul livre qui lui-même contient des sections de niveau 1 qui contiennent des sections de niveau 2.

book.xml	
<pre> <?xml version="1.0"?> <book> <title>Data on the Web</title> <author>Serge Abiteboul</author> <author>Peter Buneman</author> <author>Dan Suciu</author> <sectionid="intro" difficulty="easy"> <title>Introduction</title> <p>Text...</p> </section> <section> <title>Audience</title> <p>Text...</p> </section> <section> <title>Web Data and the Two Cultures</title> <p>Text...</p> <figure height="400" width="400"> <title>Traditional client/server architecture</title> <image source="csarch.gif"/> </figure> <p>Text...</p> </section> </book> </pre>	<pre> <sectionid="syntax" difficulty="medium"> <title>A Syntax For Data</title> <p>Text...</p> <figure height="200" width="500"> <title>Graph representation of structures</title> <image source="graphs.gif"/> </figure> <p>Text...</p> <section> <title>Base Types</title> <p>Text...</p> </section> <section> <title>Representing Relational Databases</title> <p>Text...</p> <figure height="250" width="400"> <title>Examples of Relations</title> <image source="relations.gif"/> </figure> </section> <section> <title>Representing Object Databases</title> <p>Text...</p> </section> </section> </book> </pre>

Ecrire en XQuery les requêtes suivantes:

1. Donner la table des matières au niveau 1 pour chaque livre. Le résultat doit être:

```

<resultat>
  <livre titre="Data on the Web">
    <sectionid="intro" difficulty="easy">

```

```
<title>Introduction</title>
</section>
<sectionid="syntax"difficulty="medium">
<title>ASyntaxForData</title>
</section>
</livre>
</resultat>
```

1 point

```
<resultat>{
  for $b in doc("book.xml")/book
  return
  <livre titre="{ $b/title/text()}">
  {
```

```

    for $section in $b/section
    return
      <section>
        { $section/@* , $section/title }
      </section>
    }
  </livre>
}
</resultat>

```

2. Liste des figures avec leur titre et leurs attributs, mais avec, pour chaque figure, la section où elle apparaît. Le résultat doit être:

```

<liste_figures>
<figure section="Web Data and the Two Cultures" height="400" width="400">
<title>Traditional client/server architecture</title>
</figure>
<figure section="A Syntax For Data" height="200" width="500">
<title>Graph representations of structures</title>
</figure>
<figure section="Representing Relational Databases" height="250" width="400">
<title>Examples of Relations</title>
</figure>
</liste_figures>

```

1 point

```
<liste_figures>
{
  for $f in doc("book.xml")//figure
  return
    <figure section="{ $f/../../title }">
      { $f/@* }
      { $f/title }
    </figure>
}
</liste_figures>
```

3. Nombre de section pour chaque niveau et total.

Le résultat doit être:

```
<resultat>
<nb_sections_niveau1>2</nb_sections_niveau1>
<nb_sections_niveau2>5</nb_sections_niveau2>
<nb_sections_total>7</nb_sections_total>
</resultat>
```

1 point

```
<resultat>
<nb_sections_niveau1>
{
  count(doc("book.xml")/book/section)
}
</nb_sections_niveau1>
<nb_sections_niveau2>
{
  count(doc("book.xml")/book/section/section)
}
</nb_sections_niveau2>
<nb_sections_total>
```

```
{
  count(doc("book.xml")//section)
}
</nb_sections_total>
</resultat>
```

4. Pour chaque section de niveau 1, donner la hauteur moyenne des figures et la liste des figures dont la hauteur est supérieure ou égale à cette moyenne. La liste des sections doit apparaître dans l'ordre croissant de la hauteur moyenne.

On obtient les résultats suivants:

```
<resultat>
<sectiontitle="ASyntaxForData"hauteurmoy="225">
<figure_hautetitle="ExamplesofRelations"hauteur="250"/>
</section>
<sectiontitle="Introduction"hauteurmoy="400">
<figure_hautetitle="Traditionalclient/serverarchitecture"hauteur="400"/>
</section>
</resultat>
```

1 point

```
resultat>
{
```



```

for $s in doc("book.xml")/book/section
let $hm := avg($s//figure/@height)
order by $hm
return
  <section title="{ $s/title/text() }" hauteurmoy="{ $hm }">
    {for $f in $s//figure[@height>=$hm]
     return
      <figure_haute title="{ $f/title/text() }" hauteur="{ $f/@height }"/>
    }
  </section>
}
</resultat>

```

5. Pour chaque livre, donner la liste des auteurs s
section (niveau 2 imbriquée dans niveau 1), donner l
partir des éléments. La taille d'un élément <p>
taille d'une figure est donnée par ses dimensions.

ous forme d'attribut de type string, et, pour chaqu
a taille (surface) occupée par la section, calculée
à
vaut 3000, tout comme la taille d'un élément <title
>. La

On obtient le résultat suivant:

```

<resultat>
<livre titre="Data on the Web" auteurs="Serge Abiteboul Peter Buneman Dan Suciu">
  <section_niv1 titre="Introduction" taille="184000">
    <section_niv2 titre="Audience" taille="6000"/>
    <section_niv2 titre="Web Data and the Two Cultures" taille="172000"/>
  </section_niv1>
  <section_niv1 titre="A Syntax For Data" taille="233000">
    <section_niv2 titre="Base Types" taille="6000"/>
    <section_niv2 titre="Representing Relational Databases" taille="109000"/>
    <section_niv2 titre="Representing Object Data bases" taille="6000"/>
  </section_niv1>
</livre>
</resultat>

```

1 point

```
<resultat>
{
  for $b in doc("book.xml")/book
  return
  <livre titre="{ $b/title}" auteurs="{ $b/author/text()}">

  {for $s in $b/section
   let $taille := sum($s//figure/(@height * @width))
   let $taille2 := 3000*count($s//p|title))
   return
  <section_niv1 titre="{ $s/title}" taille="{ $taille+$taille2}" >
  {for $ss in $s/section
   let $sstable := sum($ss//figure/(@height * @width))
   let $sstable2 := 3000*count($ss//p|title))
   return
   <section_niv2 titre="{ $ss/title}" taille="{ $sstable+$sstable2}" />
  }
  </section_niv1>
}
</livre>
}
</resultat>
```