

MLBDA – 41801- Examen : 20 juin 2018

Ex₁ :	Ex₂ :	Ex₃ :	Ex₄ :	
-------------------------	-------------------------	-------------------------	-------------------------	--

Les documents de cours, TD et TME sont autorisés – durée 2h

Tous les appareils électroniques doivent être éteints et rangés. Répondre en justifiant vos réponses sur la feuille du sujet dans les cadres appropriés. Le barème est indicatif. La clarté de la rédaction compte. Ecrire à l'encre bleue ou noire.

Exercice 1 : SQL3

5 pts

On considère le schéma SQL3 suivant décrivant des analyses et l'enchaînement des traitements à effectuer sur des fichiers (exemple d'analyse: extraire des mots clés à partir d'un ensemble de documents, puis transformer les documents et les mots clé pour obtenir un ensemble de domaines). Une analyse contient plusieurs lectures d e fichiers. Chaque lecture est suivie de plusieurs opérations et ainsi de suite.

<pre> create type Transformation; / create type ListeTransformations as table of ref Transformation; / create type Op as object (numOp Number, suivieDe ListeTransformations, member function OpType return Varchar2) not final; / create type Lecture under Op (fichier Varchar2(30), taille Number -- (taille en Go)); / create type ListeLectures as table of ref Lecture; / </pre>	<pre> create type Transformation under Op (traitement Varchar2(30), resultat Varchar2(30), duree Number(3) -- (durée en heures)); / create type ListeNoms as table of Varchar2(30); / create type Analyse as object (numA Number, auteur Varchar2(30), sources ListeLectures, member function listeFichiers return ListeNoms); / </pre>
---	---

Stockage :

```

create table LesAnalyses of Analyse nested table sources store as t1;
create table LesSources of Lecture nested table suivieDe store as t2;
create table LesTransformations of Transformation nested table suivieDe store as t3;

```

Répondre en SQL3, en suivant le modèle du cadre réponse.

Question 1. Formuler les requêtes suivantes.

- 1) Les analyses créées par Bob et contenant une lecture du fichier doc1 suivie d'une transformation qui dure 2 heures. Un élément du résultat est un objet Analyse.

```

Select _ _ _ _ _
From _ _ _ _ _
_ _ _ _ _
Where _ _ _ _ _

```


- 2) Pour chaque auteur qui a créé plus de 5 analyses, afficher l'auteur et la taille totale des fichiers lus dans les analyses de cet auteur. Un élément du résultat est un couple de type (varchar2, number).

```
Select -----
From -----
-----
... -----
-----
-----
```

- 3) Les couples de références d'analyses qui partagent un même objet Lecture.

```
Select -----
From -----
-----
... -----
-----
-----
```

Question 2 : Ecrire la méthode *listeFichiers* qui retourne la liste des noms de fichiers d'une analyse

```
member function ..... return ..... is
-----;
begin
-----
-----
-----
-----
-----
-----
-----
end;
```

Question 3 : Ecrire la requête utilisant la méthode précédente et qui affiche les analyses avec leur nombre de fichiers. Un élément du résultat est une paire (reference d'une analyse, nombre de fichiers).

Select _ _ _ _ _

From _ _ _ _ _

_ _ _ _ _

... _ _ _ _ _

_ _ _ _ _

Question 4 (bonus): On ajoute dans le type Transformation la méthode **réursive**:

member function toutesLesSuivantes return ListeTransformations

Elle retourne les noms de toutes les transformations reliées directement ou indirectement à une opération de transformation. On suppose que le graphe des opérations contenues dans une analyse est acyclique. Expliquer brièvement l'algorithme de cette méthode.

Exercice 2 : XML-DTD-XSchema

5pts

Question 1 (1 pt). Indiquez pour chacune des assertions suivantes, si elles sont vraies ou fausses :

- a) Un attribut de type ID est toujours de mode obligatoire (required)

Entourez la bonne réponse :

VRAI

FAUX

- b) Un document XML peut contenir plusieurs éléments ayant le même nom mais des types différents

Entourez la bonne réponse :

VRAI

FAUX

- c) Dans Xschema, il est possible de définir plusieurs clefs pour un même élément.

Entourez la bonne réponse :

VRAI

FAUX

- d) Dans un élément *xs:schema*, il est impossible d'associer deux préfixes différents à un même URI

Entourez la bonne réponse :

VRAI

FAUX

Question 2 (1 pt). On considère les deux DTD suivantes :

DTD 1 :

```
<!ELEMENT A ((B,C*) | (C+, B))>
<!ELEMENT B (#PCDATA)>
<!ELEMENT C (#PCDATA)>
```

DTD 2:

```
<!ELEMENT A (B,C) +>
<!ELEMENT B (#PCDATA)>
<!ELEMENT C (#PCDATA)>
```

- a) Donnez un document XML qui soit valide pour la DTD 1 et qui ne soit pas valide pour la DTD 2.

- b) Donnez un document XML qui soit valide pour la DTD 2 et qui ne soit pas valide pour la DTD 1.

Question 3 (3 pts). On considère le schéma *analyse.xsd* suivant :

```
1. <xs:schema>
2.   <xs:element name='analyse'>
3.     <xs:complexType>
4.       <xs:sequence maxOccurs='unbounded' >
5.         <xs:element ref='lecture' />
6.         <xs:element ref='transformation' />
7.       </xs:sequence>
8.       <xs:attribute name='num' type='xs:integer' use='required' />
9.       <xs:attribute name='auteur' type='xs:string' use='required' />
10.    </xs:complexType>
11.  </xs:element>

12. <xs:element name='lecture'>
13.   <xs:complexType>
14.     <xs:sequence maxOccurs='unbounded'>
15.       <xs:element name='suivieDe' type='xs:integer' />
16.     </xs:sequence>
17.     <xs:attribute name='num' type='xs:integer' use='required' />
18.     <xs:attribute name='fichier' type='xs:string' use='required' />
19.     <xs:attribute name='taille' type='xs:integer' use='required' />
20.   </xs:complexType>
21. </xs:element>

22. <xs:element name='transformation'>
23.   <xs:complexType>
24.     <xs:sequence maxOccurs='unbounded'>
25.       <xs:element name='suivieDe' type='xs:integer' />
26.     </xs:sequence>
27.     <xs:attribute name='num' type='xs:integer' use='required' />
28.     <xs:attribute name='traitement' type='xs:string' use='required' />
29.     <xs:attribute name='resultat' type='xs:string' use='required' />
30.     <xs:attribute name='duree' type='xs:integer' use='required' />
31.   </xs:complexType>
32. </xs:element>
33. </xs:schema>
```

1. Complétez le schéma pour exprimer les contraintes suivantes, en indiquant avec précision où se situent les modifications ou ajouts dans le fichier :

- a. un traitement ne doit pas dépasser 30 caractères.

- b. Les lectures et transformations sont suivies uniquement de transformations.

2. On considère le type nommé *T_operation* suivant :

```
<xs:complexType name='T_operation'>
  <xs:sequence maxOccurs='unbounded'>
    <xs:element name='suivieDe' type='xs:integer' />
  </xs:sequence>
  <xs:attribute name='num' type='xs:integer' use='required' />
</xs:complexType>
```

Définir les types nommés *T_lecture* et *T_transformation* pour les éléments *lecture* et *transformation* respectivement, en fonction du type *T_operation*.

```
<xs:complexType name='T_lecture'>
```

```
<xs:complexType name='T_transformation'>
```

Exercice 3. XPath et XQuery

5 pts

On considère la DTD ci-dessous qui décrit les articles d'un journal en ligne. Pour chaque article on connaît l'organisation ou les personnes qui l'ont signé (élément signature). On connaît aussi les termes qui servent de mots-clés pour la recherche et les médias (photo, vidéo, sons) qui y sont attachés. On connaît aussi l'année de parution de l'article et pour certains articles, le résumé (abstract).

```
<!DOCTYPE Journal [
<!ELEMENT Journal (article)+>
<!ELEMENT article (signature, terme+, media+, annee, abstract ?)>
<!ATTLIST article ident ID #REQUIRED>
<!ELEMENT signature (organisation | personne+)>
<!ELEMENT terme (#PCDATA)>
<!ELEMENT media (type, url)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT abstract (#PCDATA)>
<!ELEMENT organisation EMPTY>
<!ATTLIST organisation nom CDATA #REQUIRED>
<!ELEMENT personne EMPTY>
<!ATTLIST personne pid ID #REQUIRED nom CDATA #REQUIRED>
<!ELEMENT type (#PCDATA)>
<!ELEMENT url (#PCDATA)>
]>
```

Exprimer en **XPath** les requêtes qui retournent les informations suivantes.

1- **(0.5pt)** Les identifiants des articles de l'année 2018 contenant un média de type photo.

2- **(0.5pt)** Les identifiants des articles dont on connaît l'abstract et qui sont signés par une organisation.

3- **(0.5pt)** Les personnes ayant signé un article pour lequel trois (3) termes sont renseignés.

Exprimer en **XQuery** les requêtes suivantes.

4- **(1pt)** Pour chaque organisation le nombre total d'articles qu'elle a signés depuis l'année 2010. Le résultat devra être de la forme :

```
<organisation nom="..." nbArticles="..." />
```

5- **(1pt)** L'année ou les années où tous les articles ont été signés par plus de trois personnes.

6- **(1pt)** Pour chaque terme qui apparaît dans plus de dix (10) articles signés par des personnes, et uniquement des personnes (ne pas compter les articles signés par une organisation), retourner ce terme ainsi que le nombre de média de type 'photo' de tous les articles où ce terme apparaît. Le résultat devra être de la forme :

```
< termeFreq terme= ... nbPhoto =... / >
```

7- **(0.5pt)** Que retourne cette requête ?

```
for $t1 in distinct-values(//terme), $t2 in distinct-values(//terme) where $t1!=$t2
let $ct := count (//article[terme=$t1 and terme=$t2])
return <mystere t1="{ $t1}" t2="{ $t2}" nb="{ $ct}" />
```

Exercice 4 : RDF-SPARQL**5 pts**

Considérons une base de données contenant des villes et des départements représentés sous forme de triplets conformément au document fourni en annexe. Dans ce document la population des villes est exprimée en milliers d'habitants, la superficie des départements est exprimée en km². Exprimez les requêtes **SPARQL** qui retournent les informations suivantes. Vous ne devez pas préciser les espaces de noms utilisés ni les données interrogées (expressions **SELECT...WHERE**)

Question 1 (1 pt). La liste de tous les noms (de ville ou de département) existants dans la base. Lorsqu'il s'agit d'un nom de ville retourner son nombre d'habitants, lorsqu'il s'agit d'un département retourner sa superficie dans la colonne informations. Le département "Var" ne sera pas retourné car il n'y a pas de ville dans ce département. Il n'y a pas d'ordre particulier pour l'affichage. Le résultat doit être le suivant :

noms	informations
Grenoble	160
Savoie	6028
Vienne	30
Chambéry	50
Isère	7432

Réponse:

Question 2 (1 pt). Les noms et le nombre d'habitants des villes qui ne sont pas des préfectures. Répondre en utilisant **minus**. Le résultat est le suivant :

nom	population
Vienne	30

Réponse:

Question 3 (1 pt). Pour chaque préfecture, le nombre total de villes de son département. Le résultat sera le suivant:

préfecture	total
Grenoble	2
Chambéry	1

Réponse:

Question 4 (1 pt). Pour chaque département retourner la ville de ce département qui a le moins d'habitants. Donnez une solution en utilisant **optional** et une solution en utilisant **not exists**. Le résultat sera le suivant :

dept	ville	population
:d1	Vienne	30
:d2	Chambéry	50

Réponse:

Solution 1:

Solution 2:

Question 5 (1 pt). Les deux premiers départements avec le plus grand nombre d'habitants. Le résultat sera le suivant:

nom	population
Isère	190
Savoie	50

Rappel du n° d'anonymat :			<i>page 10</i>
----------------------------------	--	--	----------------

Réponse:
