Nom :	Prénom :	*page 1*

MLBDA – 4I801- Examen 2ème session du 2017

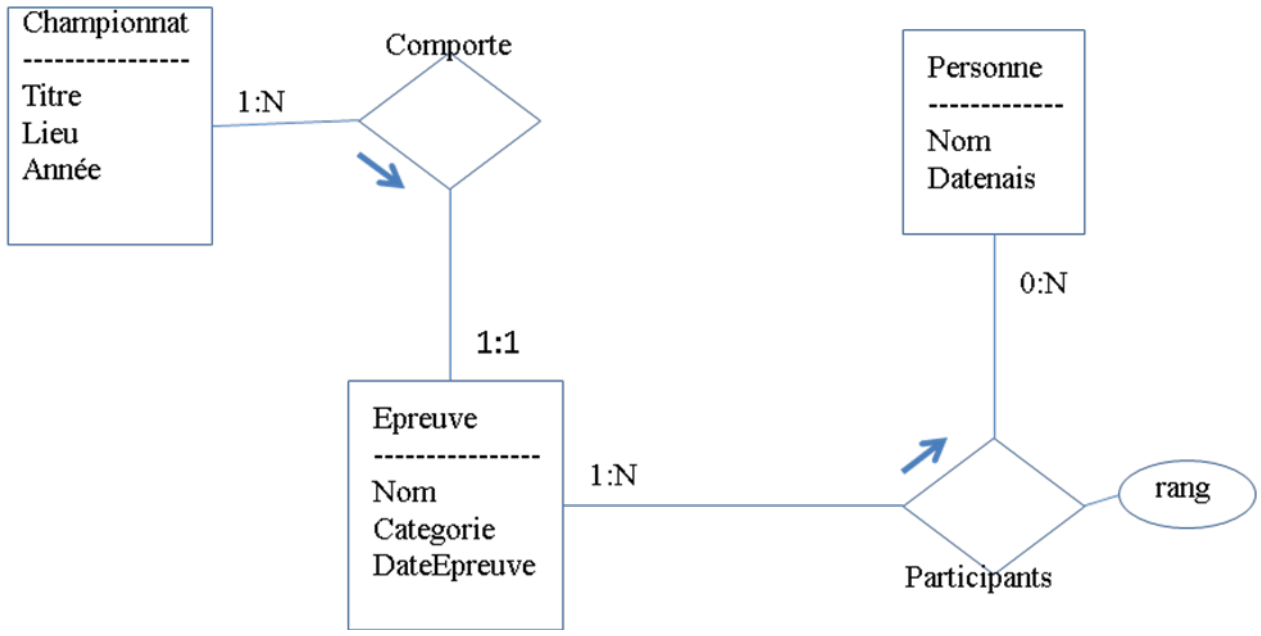
Ex1	Ex2	Ex3	Ex4	Ex5	Total

Seuls les documents de cours et de TD sont autorisés – Durée : 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. Utiliser le dos de la feuille précédente si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet. Eteindre et ranger tout téléphone et autre appareil électronique.

Exercice 1. Modélisation — 7 pts

On considère une base de données décrivant des championnats sportifs, dont le schéma Entité-Association est donné ci-dessous.



Un championnat a un titre (par ex. Championnat de France d'athlétisme), un lieu (par Ex. Angers) et une année. Un championnat comporte plusieurs épreuves, ayant chacune un nom (par ex. '100m'), une catégorie (par ex. 'Homme'), une date et des participants, avec leur rang.

La flèche indique le sens de l'association (un championnat comporte des épreuves).

Question 1. (1pt) On veut modéliser cette base en SQL3. Complétez les déclarations des types Epreuve et Championnat, en définissant de nouveaux types si nécessaire.

```
Create type Personne as object (
Nom varchar2(20)
Datenaiss Date) ;
```

Create type Epreuve as object (

Nom varchar2(20),

…….

Create type Championnat as object (

Titre varchar2(30),

…..

Question 2.(1pt) Donnez la définition des tables LesPersonnes et LesChampionnats.

Question 3. (1 pt) XML et DTD.

On considère la DTD suivante :

```
< !DOCTYPE a[
        < !ELEMENT a (b*,c)>
        < !ELEMENT b EMPTY>
        < !ELEMENT c (#PCDATA)>
        < !ATTLIST b annee CDATA #IMPLIED>
        <!ATTLIST c date CDATA #REQUIRED>
        ]>
```

Pour chacun des documents suivants, indiquez s'il est valide ou non par rapport à cette DTD. Justifiez votre réponse.

1.
```
<a>
<b annee='17' />
<b/>
<c> coucou </c>
</a>
```

Valide : OUI NON (Entourez la bonne réponse)

Justification :

2. <a>
  annee=’2020’
<c date=’mai’>2017</c>

< /a>

Valide : OUI NON (Entourez la bonne réponse)

Justification :

Question 4. (1pt) Les affirmations suivantes sont-elles exactes ? Entourez la bonne réponse.

a) Il peut y avoir plusieurs attributs de même nom dans une DTD.

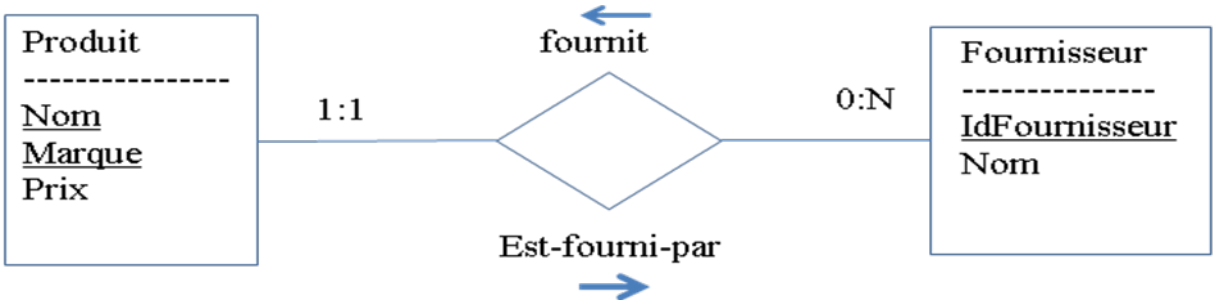
VRAI FAUX

b) Dans un document XML, toutes les valeurs des attributs de type ID, quelque soit l’élément auquel ils appartiennent, sont uniques.

VRAI FAUX

Question 5. XSchema (3 pts)

On considère le schéma entite-association suivant, décrivant des produits et leurs fournisseurs. Les clefs sont soulignées :



On veut décrire le schéma de cette application en XSchema ci-dessous. Les champs de *produit* et *fournisseur* sont représentés par des éléments, sauf IdFournisseur qui est représenté par un attribut.
L’association ‘*un produit* ***est fourni par*** *un fournisseur’* est représentée par un attribut. L’association ‘*un fournisseur* ***fournit*** *plusieurs produits’* est représentée par une séquence d’éléments. Le schéma doit garantir l’intégrité référentielle.
Les définitions de l’élément base et de l’élément produit sont données ci-dessous. .

```
<?xml version=’1.0’ encoding=’ISO-8859-1’?>
<xs:schema xmlns:xs=’http://www.w3.org/2001/XMLSchema’>
<xs:element name=’base’ type=’xs:string’>
        <xs:complexType>
        <xs:choice maxOccurs=’unbounded’>
                <xs:element ref=’produit’/>
                <xs:element ref=’fournisseur’/>
        </xs:choice>
        </xs:complexType>
</xs: element>

<xs:element name=’produit’>
        <xs:complexType>
                <xs:sequence>
                        <xs:element name=’nom’ type=’xs:string’/>
                      <xs:element name=’marque’ type=’xs:string’/>
                        <xs:element name=’prix’ type=’xs:integer’/>
        </xs:sequence>
                <xs:attribute name=’fournit’ type=’xs:string’ use=’optional’/>
    </xs:complexType>
</xs:element>
</xs :schema>
```

Complétez ce schéma en ajoutant les définitions suivantes :

1. Définir l’élément *fournisseur*.

```
<xs:element name=’fournisseur’>
        <xs:complexType>









</xs :element>
```

2. Définir les clefs des éléments *produit* et *fournisseur*, en indiquant à quel endroit il faut insérer ces définitions.

Précisez à quel endroit il faut insérer ces définitions :

3. Définir l'intégrité référentielle à l'aide de *xs :keyref*. Indiquez à quel endroit il faut insérer ces définitions.

Précisez à quel endroit il faut insérer ces définitions :

Exercice 2 : SQL3 — 3 pts

On considère le schéma SQL3 suivant décrivant des fournisseurs et des produits fournis.

```
create type EnsCouleurs as table of varchar2(30) ;
/
create type Produit as object (
  nom        varchar2(30),
  marque     varchar2(30))
  coloris EnsCouleurs;
/
create type Fourniture as object(
  produit ref Produit,
  prix Number(6, 2)) ;
/
```

```
create type EnsF as table of Fourniture;
/
create type EnsP as table of ref Produit;
/
create type Fournisseur as object(
  nom        varchar2(30),
  fournit    EnsF,
  member function produitsPrixMax(m varchar2)
return EnsP // le paramètre m est une marque
);
/
```

Stockage :

Les objets *Fournisseur* sont stockés dans la table **LesF** et les objets *Produit* dans **LesP**.

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

1) Quels sont les objets Fournisseur fournissant des toupies de couleur rouge et de marque ProSpiner à moins de 5 euros? Le résultat affiche des **objets**. Le résultat ne contient **pas** de doubles.

```
Select _ _ _ _ _ _ _ _

From _ _ _ _ _ _ _ _

Where _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _


```

2) On suppose qu'un produit a au moins un fournisseur. Afficher pour chaque marque, le nombre de fournisseurs **distincts**. Le résultat est un ensemble de couples (marque, nombre de fournisseurs). Par exemple, si le fournisseur 'F1' est le seul à fournir exclusivement les produits de la marque 'M1', alors le résultat contient ('M1', 1).

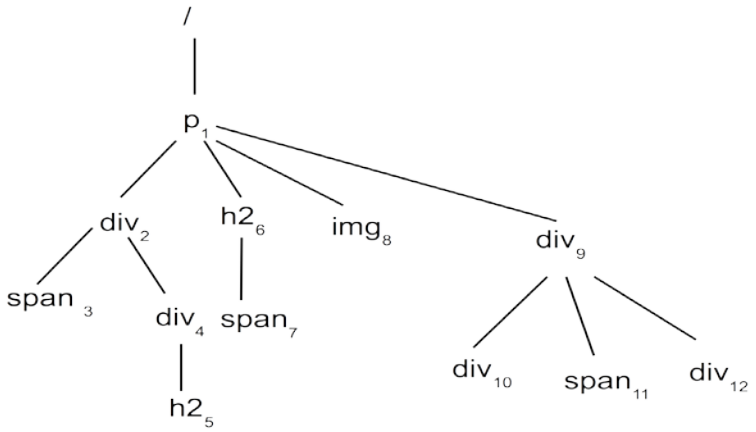
```
Select _ _ _ _ _ _ _ _

From _ _ _ _ _ _ _ _    _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Where _ _ _ _ _ _ _

_ _ _ _ _ _ _


```

3).Ecrire la méthode produitsPrixMax(m) du type Fournisseur qui retourne un ensemble de références à des produits fournis de la marque *m* dont le prix est le plus grand parmi les produits fournis de la marque *m*. Remarque, on considère seulement les produits fournis par le fournisseur sur lequel la méthode est invoquée.

```
member function produitsPrixMax(m varchar2) return EnsP is
    res _ _ _ _ _ _ _ _;
  begin
    _ _ _ _ _ _ _ _

    _ _ _ _ _ _ _ _

    _ _ _ _ _ _ _ _

    _ _ _ _ _ _ _ _

    _ _ _ _ _ _ _ _

    return res;
  end;
end;
```

Exercice 3. XPath 3 pts

On considère l'arbre XML suivant qui modélise un extrait d'un document HTML. Dans cet arbre chaque noeud est identifié par un attribut @id (l'identifiant de la racine est égal à 0, son élément fils de type p a l'identifiant 1, etc):



Le résultat d'une expression XPath est une liste de noeuds DOM triés dans l'ordre du document, sans doublons. Par exemple, l'expression `/descendant::div/following-sibling::*/@id` retourne les identifiants `6,8,9,11,12`.

Question 1 (2pts). Donnez pour chaque expression XPath la liste des identifiants des noeuds qui sont retournés.

a). */descendant::span[2]/@id*

Réponse:

b) */descendant::*[not(child::div[child::span])]/ancestor::div/@id*

Réponse:

c) */descendant::span/preceding-sibling::*/@id*

Réponse:

d) //[not(parent::p) and not(child::*)]/@id*

Réponse:

Question 2 (1 pt). Donnez pour chaque séquence d'identifiants de noeuds ci-dessous une expression XPath (syntaxe étendue ou abrégée) qui la calcule (***Attention**: ne **pas** utiliser les identifiants des noeuds du résultat dans l'expression Xpath, par exemple pour le résultat a) la réponse **//div[@id=2 or @id=9]/@id** n'est pas acceptée*).

a) *9, 12*

Réponse:

b) 3, 11

Réponse:

Exercice 4. XQuery 2 pts

Considérons le document *projets.xml* suivant, qui contient trois projets avec leurs employés, leur localisation et leurs budget.

```
<projects>
    <project name='Accounting'>
        <employee><ln> Jenkins</ln><fn> David</fn></employee>
        <employee><ln>Williams </ln><fn> Jessica</fn> </employee>
        <employee><ln>Smith </ln><fn>Alex </fn> </employee>
        <budget> 400 </budget>
        <location>New York </location>
            <duration>24 months</duration>
    </project>
    <project name='Quality'>
        <employee><ln>Crosby </ln><fn>Williams </fn></employee>
        <employee><ln> Williams</ln><fn>Jessica </fn> </employee>
        <budget> 200 </budget>
        <location> San Francisco</location>
    </project>
    <project name='Design'>
        <employee><ln>Crosby </ln><fn>Williams </fn></employee>
        <employee><ln> Thomas</ln><fn>Martin </fn> </employee>
        <location>San Francisco</location>
            <budget>100</budget>
            <duration>12 months </duration>
    </project>
</projects>
```

Question 3 (1 pt). Exprimez en français la requête suivante et donnez son résultat:

```
<résultats>
    {for $l in distinct-values(document("projets.xml")// employee),
     let $n := count(//projet[employee=$l] and budget < 250])
       where $n > 0 return
            <employee nom="{$l/nom/text()}" prenom="{$l/fn/text()}" nb={$n}/>

    }
</résultats>
```

Description en français:

Résultat:

Question 4 (1 pt). Exprimez en XQuery la requête qui retourne un élément <liste_projets> qui contient le nom et le budget des projets localisés à San Francisco, **triés par le nom du projet**. Le résultat attendu est le suivant:

```
<liste_projets>
<projet><nom='Design'/> <budget=100></projet>
<projet><nom='Quality'/> <budget=200></projet>
</liste_project>
```

Réponse:

Exercice 5. SPARQL — 5 pts

Considérons les triplets du document compositeurs.*ttl* donnés sous forme factorisée (cf. annexe)
Exprimer les requêtes **SPARQL** qui retournent les informations suivantes.

Question 1 (0,5 pt). Nombre d'élèves appartenant à trois écoles différentes.

Le résultat de la requête est :

?res
1

Q1

Question 2 (0,5 pt). Nombre d'élèves qui n'ont composé aucune œuvre.

Le résultat de la requête est :

?res
2

Q2

Question 3 (1 pt). Les paires encadrants avec leur(s) élève(s) qui ont composé au moins deux œuvres distinctes en commun.

Le résultat de la requête est :

?encadrant	?eleve
:jalman	:ylee

Q3

Question 4 (1.5 pt). Les compositeurs qui encadrent au moins deux élèves avec lesquels ils ont éventuellement composé une même œuvre ensemble. Le résultat de la requête ne doit pas comporter de doublons. Il doit être comme indiqué ci-dessous**.**

?en	?el
:jalman	:ylee
:fafti	
:dascu	

Q4

Question 5 (1 pt). Les œuvres composées uniquement par des encadrants. Le résultat doit être trié par ordre alphabétique **inverse.**

Le résultat de la requête est :

?o
:sonata
:concerto

Q5

Question 6 (.5 pt). Décrire ce que calcule cette requête.

ask {?c1 :compose ?o. ?c2 :compose ?o filter(str(?c1)<str(?c2))**}**