

Nom :	Prénom :	page 1
-------	----------	--------

MLBDA – 4I801- Examen 2^{ème} session du 10 juin 2015

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. SQL3

5 pts

On considère le schéma SQL3 suivant décrivant des jeux de tirage (loto, euro million, ...). Lire la 1ère colonne puis la 2ème.

create type Cases as table of Number(2); / create type Ticket as object (jeu Varchar2(30), combinaison Cases, jour Date, // la date du ticket ex. '13-6-2014' prix Number // le prix de vente); / create type Tickets as table of Ticket; /	create type Joueur as object (prenom Varchar2(30), // le prénom est unique achat Tickets); / // Club sportif create type Club as object (nom Varchar2(30), ville Varchar2(30), president ref Joueur) not final; / create table LesJoueurs of Joueur ...; create table LesClubs of Club;
--	--

1) On suppose que la table **LesJoueurs** contient déjà le joueur Alan qui a joué au loto le vendredi 13-12-2013 la combinaison (1,3,5,7,9) pour 2 euros, et le vendredi 13-6-2014 la combinaison (2,4,6,8,10) pour 12 euros.

a) Insérer le club AP6 de Paris présidé par Alan.

```
INSERT INTO _____
_____
_____
```

b) On apprend que le prix du ticket de loto joué par Alan le 13-12-2013 ne valait pas 2 euros mais **20** euros. Modifier la base en conséquence. Ecrire la modification en SQL3.

```
UPDATE _____ ( _____
_____
_____ )
```

SET _ _ _ _ _

WHERE _ _ _ _ _

_ _ _ _ _

2) On suppose que la base contient plusieurs joueurs. Qu'affiche cette requête ? Répondre en une phrase brève.

```
select value(j)
from LesJoueurs j
where (      select count(value(a))
            from table(j.achat) a) = ( select max(( select count(value(a2))
                                                    from table(j2.achat) a2))
            from LesJoueurs j2)
;
```

Elle affiche _ _ _ _ _

_ _ _

3) Ecrire la requête : pour un président ayant acheté au moins un ticket le 27-10-2014, donner son prénom et le nom du club qu'il préside. Trier le résultat par nom de club.

Select _ _ _ _ _

From _ _ _ _ _

Where _ _ _ _ _

_ _ _ _ _

4) Pour chaque président de club, afficher le président (i.e. afficher entièrement l'objet de type Joueur) et le nombre de clubs qu'il préside.

Select _ _ _ _ _

From _ _ _ _ _

_ _ _ _ _

5) On complète le type Joueur avec la méthode numFavorisAuLoto() retournant les numéros que le joueur a déjà joués au moins 5 fois au loto. Compléter le corps de cette méthode

```
member function numFavorisAuLoto return Cases is
resultat _ _ _ _ _
begin
```

```

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

```

```

    return resultat ;
end ;

```

6) (bonus) Afficher le prénom des joueurs qui ont 13 parmi leurs numéros favoris au loto. Répondre en invoquant une méthode.

```

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

```

Exercice 2. XML-DTD-XSchema

4 pts

On souhaite écrire une DTD pour le fichier *tweets.xml* (annexe 2) qui décrit les messages envoyés sur le site web de Twitter.

Question 1 (1 pt). Complétez la DTD ci-dessous :

```

< !ELEMENT tweets .....

< !ELEMENT tweet .....

< !ELEMENT retweetDe.....

<!ELEMENT date (#PCDATA)>
<!ELEMENT auteur EMPTY>
<!ATTLIST auteur id ID #REQUIRED>
<!ELEMENT hashtags (hashtag)+>
< !ELEMENT hashtag (#PCDATA)>

```

Question 2 (1,5 pt). On souhaite définir un schema XSchema pour modéliser les informations du fichier *tweets.xml*. Complétez la définition de l'élément `tweet` ci-dessous, afin qu'il soit conforme au fichier *tweets.xml*.

```

<xs:element name="tweets">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tweet"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="tweet">
  <xs:complexType>
    .....
    .....

    <xs:element name="date" type="xs:string"/>
    <xs:element name="auteur" .....

    .....
    .....
    .....
    .....

    </xs:element>
    <xs:element name="texte" type="xs:string"/>
    <xs:element name="hashtags" .....

    .....
    .....
    .....
    .....
    .....

    </xs:element>
    <xs:element name="retweetDe" .....

    .....
    .....
    .....
    .....

    </xs:element>

```

```

.....
.....
.....
    <xs:attribute name="id" type="xs:string" .....
    <xs:attribute name="langue" type="xs:string" .....
    <xs:attribute name="type" type="xs:string" .....
.....
.....
.....
</xs:complexType>
</xs:element>
</xs:schema>

```

Question 3 (1,5 pts).

Définir les contraintes suivantes en indiquant où les modifications doivent être effectuées :

- a) L'attribut `id` de l'élément `tweet` est une clef pour cet élément.

- b) L'attribut `tw` de l'élément `retweetDe` doit référencer un `tweet` existant dans la base.

- c) La longueur d'un `tweet` est limitée à 140 caractères.

Exercice 3. XPath**3 pts**

On considère le fichier *"tweets.xml"* (annexe 2) qui décrit les messages (tweets) envoyés sur le site web de microblogage Twitter. Exprimez en XPath les requêtes suivantes :

Question 1 (0,5pt). Le texte des tweets avec le hashtag *"Nepal"* ou *"Katmandou"*.

Question 2 (0,5pt). L'identifiant des tweets sans hashtags.

Question 3 (0,5pt). Le deuxième hashtag des tweets écrits en français, s'il y en a.

Question 4 (0,5pt). Le tweet qui suit dans le document le tweet avec l'identifiant *"t2"*.

Question 5 (1pt). Donnez le résultat des requêtes XPath suivantes :

a) `//tweet[@langue="fr"][3]/@id`

Résultat :

b) `//tweet[3][@langue="fr"]/@id`

Résultat :

Exercice 4. XQuery**4 pts**

On considère les fichiers *"tweets.xml"* (annexe 2) et *"utilisateurs.xml"* (annexe 3) contenant la liste des utilisateurs de Twitter.

Exprimez en XQuery les requêtes suivantes. Pour chaque requête, on donne la DTD du résultat qu'on veut obtenir et un exemple de résultat en XML.

Question 1 (1 pt). Le login et nom des utilisateurs avec au moins 2 tweets :

DTD du résultat :

```
<!ELEMENT resultat (utilisateur)*>
<!ELEMENT utilisateur EMPTY>
<!ATTLIST utilisateur login CDATA>
<!ATTLIST utilisateur nom CDATA>
```

Exemple

```
<resultat>
  <utilisateur login="@afpfr"
    nom="Agence France-Presse"/>
  <utilisateur login="@pauwels"
    nom="Chloé Pauwels"/>
</resultat>
```

```

<resultat> {
  for

  where

  return

} </resultat>

```

Question 2 (1,5 pt). Pour chaque utilisateur, la liste des langues distinctes utilisées dans ses tweets :

DTD du résultat

```

<!ELEMENT resultat (utilisateur)*>
<!ELEMENT utilisateur (langue)*>
<!ATTLIST utilisateur login CDATA>
<!ELEMENT langue CDATA>

```

Exemple

```

<resultat>
  <utilisateur login="@pauwels">
    <langue>fr</langue>
    <langue>en</langue>
  </utilisateur>
  ...
</resultat>

```

```

<resultat> {
  for ...

  return <utilisateur ...

} </resultat>

```

Question 3 (1,5 pt). Pour chaque hashtag, l'identifiant des tweets qui le contiennent. Triez les hashtags par ordre décroissant du nombre de tweets :

DTD du résultat

```

<!ELEMENT resultat (hashtag)*>
<!ELEMENT hashtag (tweet)+>
<!ATTLIST hashtag valeur CDATA>
<!ATTLIST hashtag nbr CDATA>
<!ELEMENT tweet EMPTY>
<!ATTLIST tweet id CDATA>

```

Exemple

```

<resultat>
  <hashtag valeur="Belgique" nbr="2">
    <tweet id="t1" />
    <tweet id="t2" />
  </hashtag>
  ...
</resultat>

```

```

<resultat> {
for ...
...
...
return <hashtag ...

} </resultat>

```

Exercice 5. SPARQL**4 pts**

Considérons les triplets SPARQL donnés sous forme factorisée (annexe 1).

Question 1 (1 pt). Renseigner les cardinalités (nombre de réponses) des requêtes Q1 et Q2 données ci-dessous.

Q1

```

select (count(*) as ?nb_triples)
where { ?s :countryOfBirth ?o }

```

Nombre de réponses :

Q2

```

select (count(distinct ?p) as ?nb_pred)
where { :ryan_gosling ?p ?o }

```

Nombre de réponses :

Question 2 Exprimer les requêtes suivantes en SPARQL.

Remarque : ne pas écrire la déclaration des préfixes dans les requêtes

a)(1 pt). Les acteurs qui ont joué dans un film de budget égal à 12 ainsi que ce film.

Le résultat de la requête est :

actor	movie
:Christina_Hendricks	: drive
:ryan_gosling	: drive

b) (1 pt). Les acteurs qui ont joué dans Drive avec éventuellement leur pays de naissance.

Le résultat de la requête est :

Actor	birthCount
:Christina_Hendricks	
:ryan_gosling	"Canada "

c) (1 pt). Les acteurs qui ont joué dans le même film.

Remarque : le résultat ne doit pas contenir de doublons.

Le résultat de la requête est

actor1	actor2
:Christina_Hendricks	:ryan_gosling