

Bases de Données II, 30 mai 2014

NOM + PRÉNOM :

Orientation + Année :

Cet examen contient 13 questions. Durée : 3 heures.

Question 1 La figure 1 montre un schéma XML pour un répertoire téléphonique. Concevez une DTD qui accepte exactement les mêmes documents XML que ceux qui sont acceptés par ce schéma XML.

.../7

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="nomcomplet">
    <xs:complexType>
      <xs:all>
        <xs:element name="prenom" type="xs:string" minOccurs="0"/>
        <xs:element name="nom" type="xs:string"/>
      </xs:all>
    </xs:complexType>
  </xs:element>

  <xs:element name="pseudonyme" type="xs:string"/>
  <xs:element name="telephone" type="xs:string"/>

  <xs:group name="identite">
    <xs:choice>
      <xs:element ref="pseudonyme"/>
      <xs:element ref="nomcomplet"/>
    </xs:choice>
  </xs:group>

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

  <xs:element name="personne">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="identite"/>
        <xs:element ref="sexe" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="telephone" minOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="favoris" type="xs:boolean"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="repertoire">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="groupe" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nomgroupe" type="xs:string"/>
              <xs:element ref="personne" minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

FIGURE 1 – XML Schema

Bases de Données II, 30 mai 2014

NOM + PRÉNOM :

Orientation + Année :

La Ligue de diamant de l'IAAF est une compétition d'athlétisme organisée annuellement par l'Association internationale des fédérations d'athlétisme (IAAF). Elle est composée de 14 meetings dans 14 villes différentes. Toutes les épreuves sont individuelles. Les résultats sont stockés dans un document XML comme illustré par la figure 2. On apprend, par exemple, qu'en 2012, Kevin Borlée a gagné les 400m hommes à Bruxelles, alors qu'il était troisième à Lausanne. En 2011, Kevin était deuxième dans cette discipline à Bruxelles derrière son frère. L'attribut `result` donne le temps en secondes; l'attribut `rank` stocke le classement.

La DTD est incluse au début du document XML de la figure 2.

Pour les questions 2 à 5 évitez l'usage des axes suivants : `parent`, `ancestor`, `following-sibling`, `preceding-sibling`, `following` et `preceding`.

Question 2 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom d'un athlète masculin ayant déjà gagné les 400m. Pour le document de la figure 2, la réponse consiste en `Kevin Borlee`, `Jonathan Borlee` et `Kirani James`.

.../3

Question 3 On se doute que lors de l'encodage, il y ait eu une confusion de sexes. Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom d'un(e) athlète ayant participé à une course pour l'autre sexe. Pour le document de la figure 2, la réponse consiste en `Myriam Soumare` et `Usain Bolt`.

.../3

Question 4 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom d'un(e) athlète qui en 2012 était présent(e) à la fois à Bruxelles et à Lausanne. Pour le document de la figure 2, la réponse consiste en `Usain Bolt` et `Kevin Borlee`.

.../3

Question 5 Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom d'un athlète qui a été dans une même course que Kevin Borlée. Pour le document de la figure 2, la réponse consiste en `Kevin Borlee`, `Jonathan Borlee` et `Kirani James`. **Ajoutez une explication en français.**

.../5

```

<?xml version="1.0"?>
<!DOCTYPE DIAMONDLEAGUE [
  <!ELEMENT DIAMONDLEAGUE (ATHLETES,MEETINGS)>
  <!ELEMENT ATHLETES (ATHLETE*)>
  <!ELEMENT MEETINGS (YEAR*)>
  <!ELEMENT YEAR (MEETING*)>
  <!ELEMENT MEETING (DISCIPLINE*)>
  <!ELEMENT DISCIPLINE (ATHLETE*)>
  <!ELEMENT ATHLETE (#PCDATA)>
  <!ATTLIST ATHLETE id CDATA #REQUIRED>
  <!ATTLIST ATHLETE sex CDATA #IMPLIED>
  <!ATTLIST ATHLETE nat CDATA #IMPLIED>
  <!ATTLIST ATHLETE result CDATA #IMPLIED>
  <!ATTLIST ATHLETE rank CDATA #IMPLIED>
  <!ATTLIST DISCIPLINE name CDATA #REQUIRED>
  <!ATTLIST DISCIPLINE sex CDATA #REQUIRED>
  <!ATTLIST YEAR yr CDATA #REQUIRED>
  <!ATTLIST MEETING city CDATA #REQUIRED>
  <!ATTLIST MEETING date CDATA #REQUIRED> ]>
<DIAMONDLEAGUE>
<ATHLETES>
  <ATHLETE id="ms" sex="F" nat="FRA">Myriam Soumare</ATHLETE>
  <ATHLETE id="yb" sex="M" nat="JAM">Yohann Blake</ATHLETE>
  <ATHLETE id="ub" sex="M" nat="JAM">Usain Bolt</ATHLETE>
  <ATHLETE id="nc" sex="M" nat="JAM">Nesta Carter</ATHLETE>
  <ATHLETE id="kb" sex="M" nat="BEL">Kevin Borlee</ATHLETE>
  <ATHLETE id="jb" sex="M" nat="BEL">Jonathan Borlee</ATHLETE>
  <ATHLETE id="cm" sex="M" nat="NED">Churandy Martina</ATHLETE>
  <ATHLETE id="kj" sex="M" nat="GRD">Kirani James</ATHLETE>
  <ATHLETE id="cl" sex="M" nat="FRA">Christophe Lemaitre</ATHLETE>
</ATHLETES>
<MEETINGS>
  <YEAR yr="2012"><MEETING city="Bruxelles" date="Sep 7">
    <DISCIPLINE name="200m" sex="F">
      <ATHLETE id="ms" result="22.63" rank="1"/>
      <ATHLETE id="ub" result="19.58" rank="43"/></DISCIPLINE>
    <DISCIPLINE name="200m" sex="M">
      <ATHLETE id="yb" result="19.54" rank="1"/>
      <ATHLETE id="cl" result="20.17" rank="3"/>
      <ATHLETE id="ms" result="22.63" rank="42"/></DISCIPLINE>
    <DISCIPLINE name="100m" sex="M">
      <ATHLETE id="ub" result="9.86" rank="1"/>
      <ATHLETE id="nc" result="9.96" rank="2"/></DISCIPLINE>
    <DISCIPLINE name="400m" sex="M">
      <ATHLETE id="kb" result="44.75" rank="1"/>
      <ATHLETE id="jb" result="45.02" rank="2"/></DISCIPLINE></MEETING>
    <MEETING city="Lausanne" date="Aug 23">
      <DISCIPLINE name="200m" sex="M">
        <ATHLETE id="ub" result="19.58" rank="1"/>
        <ATHLETE id="cm" result="19.85" rank="2"/></DISCIPLINE>
      <DISCIPLINE name="400m" sex="M">
        <ATHLETE id="kj" result="44.37" rank="1"/>
        <ATHLETE id="kb" result="45.27" rank="3"/></DISCIPLINE></MEETING></YEAR>
  <YEAR yr="2011"><MEETING city="Bruxelles" date="Sep 16">
    <DISCIPLINE name="400m" sex="M">
      <ATHLETE id="jb" result="44.78" rank="1"/>
      <ATHLETE id="kb" result="44.97" rank="2"/></DISCIPLINE></MEETING></YEAR>
</MEETINGS>
</DIAMONDLEAGUE>

```

FIGURE 2 – Fichier XML avec des informations sur les podiums des courses cyclistes.

```

<RESULTS>
<Myriam-Soumare>
  <RESULT date="2012 Sep 7" discipline="200m F">22.63</RESULT>
</Myriam-Soumare>
<Yohann-Blake>
  <RESULT date="2012 Sep 7" discipline="200m M">19.54</RESULT>
</Yohann-Blake>
<Usain-Bolt>
  <RESULT date="2012 Sep 7" discipline="100m M">9.86</RESULT>
  <RESULT date="2012 Aug 23" discipline="200m M">19.58</RESULT>
</Usain-Bolt>
<Nesta-Carter>
<RESULT date="2012 Sep 7" discipline="100m M">9.96</RESULT>
</Nesta-Carter>
<Kevin-Borlee>
  <RESULT date="2012 Sep 7" discipline="400m M">44.75</RESULT>
  <RESULT date="2012 Aug 23" discipline="400m M">45.27</RESULT>
  <RESULT date="2011 Sep 16" discipline="400m M">44.97</RESULT>
</Kevin-Borlee>
<Jonathan-Borlee>
  <RESULT date="2012 Sep 7" discipline="400m M">45.02</RESULT>
  <RESULT date="2011 Sep 16" discipline="400m M">44.78</RESULT>
</Jonathan-Borlee>
<Churandy-Martina>
  <RESULT date="2012 Aug 23" discipline="200m M">19.85</RESULT>
</Churandy-Martina>
<Kirani-James>
  <RESULT date="2012 Aug 23" discipline="400m M">44.37</RESULT>
</Kirani-James>
<Christophe-Lemaitre>
  <RESULT date="2012 Sep 7" discipline="200m M">20.17</RESULT>
</Christophe-Lemaitre>
</RESULTS>

```

FIGURE 3 – Output du programme XSLT.

Question 6 Écrivez un programme XSLT qui génère un document XML affichant les résultats de chaque athlète, dans le format illustré par la figure 3. La position des blancs et retours à la ligne n'a pas d'importance. Le programme ne peut pas contenir des `xsl:for-each` ou `xsl:if`.

Noter :

- XSLT permet de concaténer des strings avec la fonction `concat(string1,string2,...,stringn)`.
- Dans le résultat, on n'affiche pas les participations avec un conflit de sexes (voir la question 3).

Question 7 Écrivez une requête XQuery qui rend tout triple (x,y,z) tel que x est le nom d'un athlète qui a participé dans l'année y au meeting dans la ville z . La liste peut contenir des doublons. Pour le document de la figure 2, la réponse contient les triples suivants :

(Myriam Soumare, 2012, Bruxelles)
(Usain Bolt, 2012, Bruxelles)
(Yohann Blake, 2012, Bruxelles)
(Christophe Lemaitre, 2012, Bruxelles)
(Nesta Carter, 2012, Bruxelles)
⋮
(Kevin Borlee, 2011, Bruxelles)

.../5

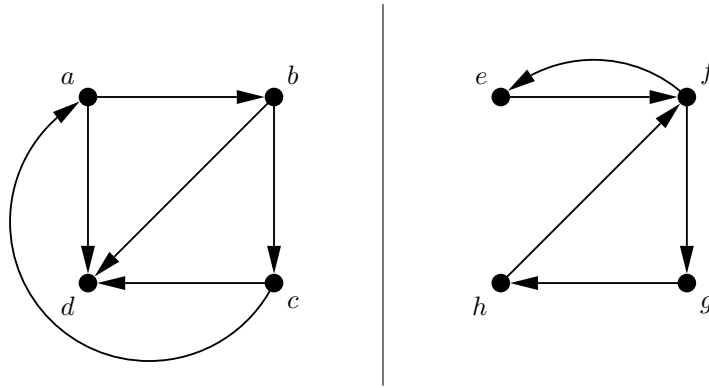


FIGURE 4 – Deux graphes dirigés.

Un graphe dirigé est encodé en utilisant le prédicat E pour les arêtes. On peut supposer que chaque nœud a une arête entrante ou sortante. Soient a, b deux nœuds. Soit $k \geq 1$. Un *chemin dirigé de a à b* est une séquence de nœuds $\langle n_1, n_2, \dots, n_k \rangle$ telle que $a = n_1$, $b = n_k$ et pour tout $i \in \{1, \dots, k-1\}$, $E(n_i, n_{i+1})$ est une arête du graphe. La *longueur* de ce chemin est $k-1$. Un graphe dirigé est *fortement connexe* si pour tous les nœuds a et b , il existe un chemin dirigé de a à b et un chemin dirigé de b à a . Un nœud a d'un graphe dirigé est *un roi* si pour tout nœud b , il existe un chemin dirigé de a à b dont la longueur est strictement inférieure à 3.

Par exemple, le graphe de gauche dans la figure 4 est encodé par $\{E(a, b), E(b, c), E(b, d), E(c, d), E(a, d), E(c, a)\}$. Dans ce graphe, $\langle a, b, c, d \rangle$ est un chemin dirigé de a à d , de longueur 3. Le graphe n'est pas fortement connexe, car il n'y a pas de chemin de d à a . Les rois sont a , b et c . Le graphe de droite dans la figure 4 est fortement connexe. Les rois sont f et h .

Question 8 Écrivez un programme en datalog⁻ (i.e., datalog avec négation) pour le prédicat idb $Roi(x)$ qui signifie que x est un roi du graphe.

.../5

Question 9 Écrivez un programme en datalog[¬] (i.e., datalog avec négation) pour le prédicat idb $Fc(\text{“oui”})$ qui vérifie si un graphe est fortement connexe.

Ne soyez pas perturbé : un programme datalog[¬] peut avoir “oui” comme réponse. Par exemple,

$$Cycle(\text{“oui”}) \leftarrow E(x, y), E(y, x)$$

retourne $Cycle(\text{“oui”})$ si et seulement si le graphe contient un cycle de taille 2.

.../5

Question 10 Voici une requête qui est l'union de trois requêtes conjonctives. Simplifiez cette requête en s'appuyant sur les théorèmes vus au cours ou expliquez pourquoi aucune simplification n'est possible.

$$\begin{cases} \text{Answer}(x, y) \leftarrow R(x, v, v), R(x, v, w), R(u, u, y), R(v, w, u), R(w, u, y) \\ \text{Answer}(y, x) \leftarrow R(y, u, u), R(u, v, w), R(w, w, x) \\ \text{Answer}(y, x) \leftarrow R(y, u, v), R(v, v, v), R(v, w, x) \end{cases}$$

.../5

Question 11 On considère les trois relations edb suivantes :

- $Frequente(x, y)$ signifie que x est un buveur qui fréquente le bar y . Par exemple, $Frequente(\text{Ed}, \text{Calypso})$.
- $Sert(y, z)$ signifie que y un bar qui sert la bière z . Par exemple, $Sert(\text{Calypso}, \text{Orval})$.
- $Aime(x, z)$ signifie que x est un buveur qui aime la bière z . Par exemple, $Aime(\text{Ed}, \text{Orval})$.

On suppose que chaque buveur aime au moins une bière et fréquente au moins un bar, que chaque bar sert au moins une bière et que chaque bière est servie dans au moins un bar. Exprimer en datalog⁷ le prédicat idb $Attentif(y)$ qui signifie que y est un bar tel que chaque buveur qui fréquente y aime au moins une bière servie par y .

.../5

Question 12 Dessinez le *dependency graph* pour le programme datalog[−] suivant. Donnez le résultat de ce programme pour la base de données $\{E(a, b), E(b, c), E(c, d), F(a, d), F(b, a), F(c, b)\}$. Détaillez les calculs.

$$\left\{ \begin{array}{l} R(x, y) \leftarrow E(x, y) \\ G(x, y) \leftarrow F(x, y), \neg R(x, y) \\ G(x, z) \leftarrow G(x, y), F(y, z) \\ R(x, z) \leftarrow R(x, y), R(y, z) \\ B(x, y) \leftarrow F(x, y), \neg G(x, y) \end{array} \right.$$

.../5

Question 13 Soit \mathbf{S} le schéma $\mathbf{S} = \{R[B, C, E], S[A, B], T[A, D], U[D, F], V[A, B, C, D]\}$. Supposons que l'on souhaite calculer la jointure de ces cinq relations. Puisque l'opérateur \bowtie est commutatif et associatif, le résultat final ne dépend pas de l'ordre dans lequel on effectue les opérations. Cependant, l'ordre peut être important d'un point de vue pratique.

1. Illustrez à l'aide d'une base de données concrète pourquoi l'expression suivante pose un problème pratique, même si aucun tuple de la base n'est *dangling* :

$$(((R \bowtie S) \bowtie T) \bowtie U) \bowtie V).$$

Notez que les parenthèses fixent l'ordre dans lequel on effectue les opérations.

2. Donnez une expression (un ordre) qui évite ce problème. Expliquez comment vous trouvez cette expression.

.../10
