

Bases de Données II, 6 juin 2016

NOM + PRÉNOM :

Orientation + Année :

Cet examen contient 13 questions. Durée : 3 heures et 30 minutes.

Les prix et volumes de fruits disponibles auprès de différents magasins sont stockés dans un document XML. Les prix sont exprimés en euro/kg ou eurocent/kg, et les volumes (attribut `stock`) en tonne. Par exemple, vers la fin du fichier, on observe que Colruyt vend la Jonagold au prix de 145 eurocent/kg et dispose d'un stock de 2 tonne de cette variété.

La première partie du fichier liste des variétés de fruits, avec leur mois de cueillette. La Jonagold est une variété de pomme cueillie en octobre.

La DTD est incluse au début du document XML de la figure 1.

Pour les questions 1 à 5, évitez, si possible, l'usage des axes suivants : parent, ancestor, following-sibling, preceding-sibling, following et preceding.

Pour les questions 1 à 5, il n'est pas permis d'utiliser des fonctions d'agrégation, telles que count, max, min...

Question 1 Écrivez une expression XPath (aussi simple que possible) qui rend les variétés de fruits qui sont cueillies en août et que l'on ne trouve dans aucun magasin. Pour le document de la figure 1, la réponse est comme suit :

```
<vnom sorte="prune">Mirabelle de Nancy</vnom>  
<vnom sorte="pomme">James Grieve</vnom>
```

| |
|-------|
| .../5 |
|-------|

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE VenteDeFruits [
<!ELEMENT VenteDeFruits (fruits, ventes)>
<!ELEMENT fruits (variete)*>
<!ELEMENT variete (vnom, cueillette)>
<!ELEMENT ventes (magasin)*>
<!ELEMENT magasin (mnom, gamme)>
<!ELEMENT gamme (fruit)*>
<!ELEMENT vnom (#PCDATA)>
<!ELEMENT mnom (#PCDATA)>
<!ELEMENT cueillette (#PCDATA)>
<!ELEMENT fruit (#PCDATA)>
<!ATTLIST vnom sorte CDATA #REQUIRED>
<!ATTLIST fruit unit (euro|eurocent) #REQUIRED>
<!ATTLIST fruit prix CDATA #REQUIRED>
<!ATTLIST fruit stock CDATA #REQUIRED>
]>
<VenteDeFruits>
<fruits>
  <variete><vnom sorte="poire">Conference</vnom>
    <cueillette>octobre</cueillette> </variete>
  <variete><vnom sorte="poire">Doyenne</vnom>
    <cueillette>octobre</cueillette> </variete>
  <variete><vnom sorte="pomme">Jonathan</vnom>
    <cueillette>octobre</cueillette> </variete>
  <variete><vnom sorte="pomme">Reinette de France</vnom>
    <cueillette>octobre</cueillette> </variete>
  <variete><vnom sorte="prune">Mirabelle de Nancy</vnom>
    <cueillette>aout</cueillette> </variete>
  <variete><vnom sorte="pomme">James Grieve</vnom>
    <cueillette>aout</cueillette> </variete>
  <variete><vnom sorte="prune">Queen Victoria</vnom>
    <cueillette>aout</cueillette> </variete>
  <variete><vnom sorte="cerise">Bigarreau Sunburst</vnom>
    <cueillette>juillet</cueillette> </variete>
</fruits>
<ventes>
  <magasin>
    <mnom>Aldi</mnom>
    <gamme><fruit unit="eurocent" prix="205" stock="11">Conference</fruit>
      <fruit unit="eurocent" prix="145" stock="2">Jonathan</fruit>
      <fruit unit="euro" prix="2.40" stock="2">Reinette de France</fruit> </gamme>
    </magasin>
  <magasin>
    <mnom>Lidl</mnom>
    <gamme><fruit unit="euro" prix="1.90" stock="11">Conference</fruit>
      <fruit unit="euro" prix="2.45" stock="9">Queen Victoria</fruit> </gamme>
    </magasin>
  <magasin>
    <mnom>Colruyt</mnom>
    <gamme><fruit unit="euro" prix="2.10" stock="3">Conference</fruit>
      <fruit unit="euro" prix="2.75" stock="7">Queen Victoria</fruit>
      <fruit unit="eurocent" prix="145" stock="2">Jonathan</fruit> </gamme>
    </magasin>
</ventes>
</VenteDeFruits>

```

FIGURE 1 – Vente de fruits.

Question 2 Écrivez un programme XQuery qui retourne une liste de fruits dont le prix n'est listé qu'en euros (et pas en eurocents). Retournez le nom de la variété dans une balise `variete`, possédant un attribut `minprice` qui indique le prix de vente le plus bas disponible pour cette variété. La réponse devrait être :

```
<variete minprice="2.40">Reinette de France</variete>  
<variete minprice="2.45">Queen Victoria</variete>
```

| |
|-------|
| .../5 |
|-------|

NOM + PRÉNOM :

Question 3 Écrivez un programme XQuery qui retourne un document XML listant, pour chaque mois, les variétés, groupées par type de fruit, qui sont cueillies ce mois. Les mois où il n'y a pas de cueillette, ne sont pas présents. Pour chaque variété, listez les magasins qui ne l'ont *pas* en stock. L'output de votre programme devrait être similaire à celui de la figure 2 :

```

<mois nom="octobre">
  <poire>
    <variete nom="Conference"/>
    <variete nom="Doyenne">
      <magasin>Aldi</magasin>
      <magasin>Lidl</magasin>
      <magasin>Colruyt</magasin>
    </variete>
  </poire>
  <pomme>
    <variete nom="Jonathan">
      <magasin>Lidl</magasin>
    </variete>
    <variete nom="Reinette de France">
      <magasin>Lidl</magasin>
      <magasin>Colruyt</magasin>
    </variete>
  </pomme>
</mois>
<mois nom="aout">
  <pomme>
    <variete nom="James Grieve">
      <magasin>Aldi</magasin>
      <magasin>Lidl</magasin>
      <magasin>Colruyt</magasin>
    </variete>
  </pomme>
  <prune>
    <variete nom="Mirabelle de Nancy">
      <magasin>Aldi</magasin>
      <magasin>Lidl</magasin>
      <magasin>Colruyt</magasin>
    </variete>
    <variete nom="Queen Victoria">
      <magasin>Aldi</magasin>
    </variete>
  </prune>
</mois>
<mois nom="juillet">
  <cerise>
    <variete nom="Bigarreau Sunburst">
      <magasin>Aldi</magasin>
      <magasin>Lidl</magasin>
      <magasin>Colruyt</magasin>
    </variete>
  </cerise>
</mois>

```

FIGURE 2 – Output du programme XQuery de la question 3.

Question 4 Écrivez une expression XPath (aussi simple que possible) qui rend les variétés de fruits qui sont disponibles à un prix inférieur à 2 eur/kg (ou 200 eurocents/kg). Pour le document de la figure 1, la réponse est comme suit :

```
<vnom sorte="poire">Conference</vnom>  
<vnom sorte="pomme">Jonathan</vnom>
```

| |
|-------|
| .../5 |
|-------|

Question 5 Écrivez une expression XPath (aussi simple que possible) qui rend le(s) magasin(s) avec le plus grand volume de la variété Conférence en stock (i.e., aucun autre magasin ne dispose d'un stock de conférence plus large). Pour rappel, l'usage de min et max n'est pas permis. Astuce : $\max A = \{x \in A \mid \forall y \in A (x \geq y)\}$. Pour le document de la figure 1, la réponse est comme suit :

```
<mnom>Aldi</mnom>  
<mnom>Lidl</mnom>
```

| |
|-------|
| .../5 |
|-------|

Question 6 Écrivez un programme XSLT qui génère un document XML affichant les prix par variété de fruits, dans le format illustré par la figure 3. Seules les variétés que l'on sait acheter sont affichées. Les variétés sont regroupées par sorte de fruits (poire, pomme...). Ces sortes de fruits ne sont pas connues a priori, mais doivent être calculées à partir du fichier XML. Noter que toutes les sortes de fruits seront affichées, même les sortes pour lesquelles il n'existe pas de vente (comme cerise). Tous les prix sont affichés en eurocent.

La position des blancs et retours à la ligne n'a pas d'importance. Le programme ne peut pas contenir les balises `xsl:for-each`, `xsl:if` ou `xsl:with-param`.

```
<ventes>
  <poire>
    <variete nom="Conference">
      <vente magasin="Aldi" prix="205" />
      <vente magasin="Lidl" prix="190" />
      <vente magasin="Colruyt" prix="210" />
    </variete>
  </poire>
  <pomme>
    <variete nom="Jonathan">
      <vente magasin="Aldi" prix="145" />
      <vente magasin="Colruyt" prix="145" />
    </variete>
    <variete nom="Reinette de France">
      <vente magasin="Aldi" prix="240" />
    </variete>
  </pomme>
  <prune>
    <variete nom="Queen Victoria">
      <vente magasin="Lidl" prix="245" />
      <vente magasin="Colruyt" prix="275" />
    </variete>
  </prune>
  <cerise />
</ventes>
```

FIGURE 3 – Output du programme XSLT.

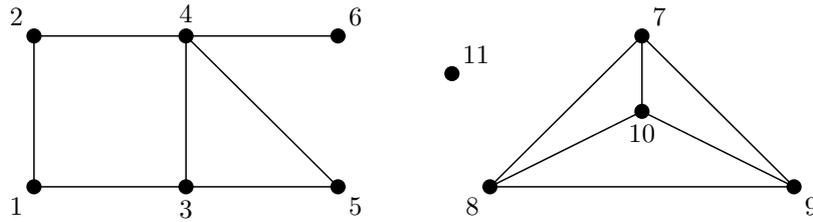


FIGURE 4 – Un graphe non orienté.

Les questions 7–10 font appel aux graphes non orientés, et leur encodage dans le modèle relationnel.

Graphe non orienté Un graphe non orienté G est un couple (V, E) où V est un ensemble fini de *sommets*, et E un ensemble de paires de sommets. Les éléments de E sont appelés *arêtes*. Noter : chaque arête $\{u, v\}$ vérifiera $u \neq v$.

Un chemin dans G est une séquence non vide de sommets telle que si deux sommets u et v sont juxtaposés dans la séquence, alors $\{u, v\}$ est une arête ; on dit que le chemin est *entre* le premier et le dernier élément de la séquence.

Par exemple, soit $G = (V, E)$ avec $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ et $E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}, \{4, 6\}, \{7, 8\}, \{7, 9\}, \{7, 10\}, \{8, 9\}, \{8, 10\}, \{9, 10\}\}$. Un chemin entre 1 et 6 est $\langle 1, 3, 4, 6 \rangle$. La séquence $\langle 8 \rangle$ est un chemin entre 8 et 8. Le graphe est visualisé dans la figure 4.

Encodage d'un graphe non orienté dans le modèle relationnel Dans le modèle relationnel, on utilisera deux relations S et A pour encoder un graphe $G = (V, E)$. La relation S est unaire et contient chaque élément de V . La relation A est binaire et pour chaque arête $\{u, v\}$ dans E , on insère dans A soit $\langle u, v \rangle$, soit $\langle v, u \rangle$ (le choix est non déterministe). Par exemple, le graphe de la figure 4 peut être encodé comme $\{S(1), S(2), S(3), S(4), S(5), S(6), S(7), S(8), S(9), S(10), S(11), A(1, 2), A(3, 1), A(2, 4), A(4, 3), A(3, 5), A(5, 4), A(4, 6), A(10, 7), A(10, 8), A(10, 9), A(7, 8), A(7, 9), A(8, 9)\}$.

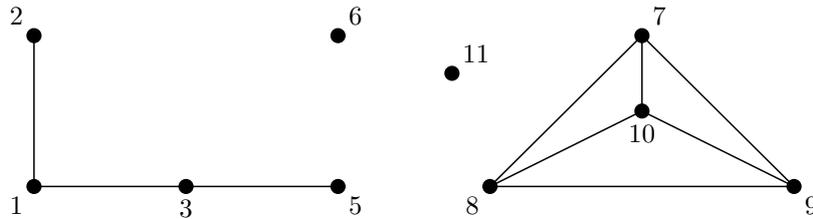
Question 7 Soit $G = (V, E)$ un graphe non orienté. On dit qu'un sommet u de V est *isolé* si u n'appartient à aucune arête. Pour le graphe de la figure 4, le sommet 11 est isolé.

Écrivez un programme en datalog[¬] (i.e., datalog avec négation) pour le prédicat idb $Isole(x)$ qui signifie que x est un sommet isolé. Les prédicats edb sont S et A introduits ci-dessus.

Question 8 Soit $G = (V, E)$ un graphe non orienté. Un ensemble $C \subseteq V$ forme une *composante connexe* si C est un ensemble maximal (par rapport à \subseteq) tel qu'il existe un chemin entre chaque deux sommets de C .

Le graphe de la figure 4 contient trois composantes connexes : $\{1, 2, 3, 4, 5, 6\}$, $\{7, 8, 9, 10\}$, $\{11\}$.

Un sommet est un *point d'articulation* si sa suppression augmente le nombre de composantes connexes. Noter : si l'on supprime un sommet, on supprime aussi toutes les arêtes qui contiennent ce sommet. Le graphe de la figure 4 ne contient qu'un seul point d'articulation : le sommet 4. La suppression du sommet 4 donne le graphe suivant avec quatre composantes connexes :



Écrivez un programme en datalog⁻ pour le prédicat idb $Articulation(x)$ qui signifie que x est un point d'articulation d'un graphe non orienté. Les prédicats edb sont S et A introduits ci-dessus. Le programme peut utiliser le prédicat \neq .

| |
|-------|
| .../7 |
|-------|

Question 9 Est-ce que le programme de la question 8 peut être écrit en datalog sans négation ? Expliquez votre réponse.

| |
|-------|
| .../2 |
|-------|

Question 10 Soit $G = (V, E)$ un graphe non orienté. Si $\{u, v\}$ est une arête, on dit que v est un voisin de u (et vice versa, que u est un voisin de v). Un sous ensemble $C \subseteq V$ forme une *clique* si pour tous sommets distincts u, v de C , on a que $\{u, v\}$ est une arête du graphe (c'est-à-dire, les sommets d'une clique sont deux-à-deux des voisins).

Un sommet est *simplicial* (pluriel : simpliciaux) si l'ensemble de ses voisins forme une clique.

Par exemple, pour le graphe de la figure 4 :

- le sommet 7 est simplicial car (i) les voisins de 7 sont 8, 9 et 10, et (ii) l'ensemble $\{8, 9, 10\}$ forme une clique ;
- le sommet 6 est simplicial car l'ensemble $\{4\}$ est une clique, et 4 est le seul voisin de 6 ;
- le sommet 3 n'est pas simplicial, car il n'y a pas d'arête entre ses voisins 1 et 4.

Écrivez un programme en datalog⁷ pour le prédicat idb *Simplicial*(x) qui signifie que x est un sommet simplicial. Les prédicats edb sont S et A introduits ci-dessus.

| |
|-------|
| .../5 |
|-------|

Question 11 Voici une requête conjonctive. Simplifiez cette requête ou expliquez pourquoi aucune simplification n'est possible.

$$\text{Answer}(x, y, z) \leftarrow R(x, u, w), R(x, y, v_1), R(v_2, u, w), R(x, v_3, z)$$

| |
|-------|
| .../5 |
|-------|

Question 12 Soit \mathbf{S} le schéma $\mathbf{S} = \{R[A, F], S[B, D, E], T[A, C, D], U[B, C, D]\}$. Supposons que l'on souhaite calculer la jointure

$$R \bowtie S \bowtie T \bowtie U. \quad (1)$$

Noter : puisque l'opérateur \bowtie est commutatif et associatif, on peut négliger les parenthèses dans l'expression (1). Pour une base de données avec relations R, S, T et U , on dira qu'un tuple r de la relation R est *dangling* par rapport à la jointure (1) si $r \notin \pi_{AF}(R \bowtie S \bowtie T \bowtie U)$. Pareillement, un tuple s de la relation S est *dangling* par rapport à (1) si $s \notin \pi_{BDE}(R \bowtie S \bowtie T \bowtie U)$. Pareillement pour T et U . Évidemment, si un tuple r de R est *dangling*, alors

$$R \bowtie S \bowtie T \bowtie U = (R \setminus \{r\}) \bowtie S \bowtie T \bowtie U.$$

En général, les tuples qui sont *dangling* peuvent être négligés dans le calcul de la jointure (1). Pour faciliter le calcul de la jointure (1), on peut donc commencer par la suppression des tuples qui sont *dangling*— en supposant, bien sûr, que cette suppression soit plus facile que le calcul de (1). Pour la jointure (1), détaillez une procédure “facile” pour supprimer tous les tuples qui sont *dangling*.

| |
|-------|
| .../5 |
|-------|

Question 13 (*Suite de la question 12.*) Illustrez à l'aide d'une base de données **concrète** que d'un point de vue pratique, l'ordre dans lequel on joint les relations R , S , T et U n'est pas anodin, même si **aucun tuple de la base n'est dangling**. Puis détaillez un ordre qui sera toujours "bon".

| |
|-------|
| .../5 |
|-------|

Construire votre base de données concrète en ajoutant des tuples aux relations ci-après. Noter : aucun tuple ne peut être dangling.

| | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R | A | F | S | B | D | E | T | A | C | D | U | B | C | D |
| | a | f | | b | d | e | | a | c | d | | b | c | d |

Illustrer à l'aide de cet exemple que même si l'ordre de jointure n'a aucun impact sur le résultat final, il affecte les performances.

Donner un ordre qui est susceptible d'être choisi par un optimiseur de SGBD. Expliquer pourquoi cet ordre est "bon".