

# Bases de Données II, Mons

Jef Wijsen

1 juin 2011

Cahier fermé. Durée : 3 heures

Nom et prénom
Année

La figure 2 montre une base de données XML concernant des propriétaires de voitures. Chaque voiture (balise `car`) est identifié de façon unique par son modèle (balise `model`, avec valeurs *323i*, *a4*, *q7*,...). La figure 1 montre le DTD.

```
<!-- This file is called cars.dtd -->
<!ELEMENT carsdb (car*, customer*)>
<!ELEMENT car (make, model, category, price)>
<!ELEMENT customer (owns*)>
<!ELEMENT owns (model, color, marketValue)>
<!ELEMENT make (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT marketValue (#PCDATA)>
<!ATTLIST car country CDATA #REQUIRED>
<!ATTLIST customer cid CDATA #REQUIRED>
<!ATTLIST customer cname CDATA #REQUIRED>
<!ATTLIST customer age CDATA #REQUIRED>
```

FIGURE 1 – DTD.

**Question 1** Disons que la *richesse* d'une personne est la somme des valeurs marchandes (`marketValue`) de ses voitures. Par exemple, la richesse de *joyce* est égale à  $25000 + 10000 = 35000$ . Disons qu'une personne est *riche* si sa richesse est supérieure à 45000. Écrivez une expression XPath qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un client riche. Pour le document de la figure 2, *john* et *terry* sont riches.

.../5

**Question 2** Écrivez une expression XPath qui ne fait pas appel à la fonction `count` et qui rend les catégories de voiture qui contiennent au moins deux modèles. Pour le document de la figure 2, les catégories *sport* et *suv* contiennent plus d'un modèle.

.../5

**Question 3** Écrivez une expression XPath qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un client qui possède une voiture de sport. Pour le document de la figure 2, *john*, *jane* et *bob* possèdent une voiture de sport.

.../5

**Question 4** Écrivez une expression XPath qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un client qui ne possède aucune voiture allemande. Pour le document de la figure 2, *ann* et *joyce* ne possèdent pas de voiture allemande.

.../5

```

<?xml version="1.0"?>
<!DOCTYPE carsdb SYSTEM "cars.dtd">
<carsdb>
  <car country="germany"><make>bmw</make><model>323i</model>
    <category>sport</category><price>25000</price></car>
  <car country="germany"><make>audi</make><model>a4</model>
    <category>sport</category><price>30000</price></car>
  <car country="germany"><make>audi</make><model>q7</model>
    <category>suv</category><price>10000</price></car>
  <car country="germany"><make>mercedes</make><model>ml</model>
    <category>suv</category><price>80000</price></car>
  <car country="england"><make>aston martin</make><model>bb9</model>
    <category>sport</category><price>100000</price></car>
  <car country="japan"><make>subaru</make><model>outback</model>
    <category>wagon</category><price>15000</price></car>
  <car country="usa"><make>ford</make><model>f100</model>
    <category>suv</category><price>10000</price></car>
  <customer cid="1" cname="john" age="20">
    <owns><model>a4</model>
      <color>white</color><marketValue>30000</marketValue></owns>
    <owns><model>q7</model>
      <color>black</color><marketValue>40000</marketValue></owns>
    <owns><model>323i</model>
      <color>red</color><marketValue>20000</marketValue></owns></customer>
  <customer cid="2" cname="mary" age="18">
    <owns><model>q7</model>
      <color>pink</color><marketValue>40000</marketValue></owns></customer>
  <customer cid="3" cname="jane" age="28">
    <owns><model>a4</model>
      <color>silver</color><marketValue>30000</marketValue></owns></customer>
  <customer cid="4" cname="ann" age="40" />
  <customer cid="5" cname="joyce" age="33">
    <owns><model>f100</model>
      <color>green</color><marketValue>25000</marketValue></owns>
    <owns><model>outback</model>
      <color>green</color><marketValue>10000</marketValue></owns></customer>
  <customer cid="6" cname="terry" age="25">
    <owns><model>ml</model>
      <color>white</color><marketValue>50000</marketValue></owns></customer>
  <customer cid="7" cname="claire" age="80">
    <owns><model>ml</model>
      <color>brown</color><marketValue>40000</marketValue></owns></customer>
  <customer cid="8" cname="bob" age="60">
    <owns><model>bb9</model>
      <color>yellow</color><marketValue>10000</marketValue></owns>
    <owns><model>ml</model>
      <color>brown</color><marketValue>20000</marketValue></owns>
    <owns><model>outback</model>
      <color>brown</color><marketValue>10000</marketValue></owns></customer>
</carsdb>

```

FIGURE 2 – Fichier XML avec des informations sur des propriétaires de voitures.

**Question 5** Écrivez un programme XSLT qui affiche pour chaque catégorie les noms des personnes qui possèdent une voiture de cette catégorie. Le résultat doit être formaté comme suit :

```
<?xml version="1.0"?>
<answer><sport>
  <name>john</name><name>jane</name><name>bob</name>
</sport>
<wagon>
  <name>joyce</name><name>bob</name>
</wagon>
<suv>
  <name>john</name><name>mary</name><name>joyce</name>
  <name>terry</name><name>claire</name><name>bob</name>
</suv>
</answer>
```

.../10

**Question 6** Rappelez-vous de vos cours de math qu'une relation binaire  $R$  est appelée *symétrique* si elle vérifie  $\forall x \forall y (R(x, y) \rightarrow R(y, x))$ .

Soit  $q$  une requête conjonctive de la forme  $Answer(x, y) \leftarrow B$ . On dit que la requête  $q$  est *symétrique* si pour toute base de données  $I$ , pour toutes constantes  $a$  et  $b$ ,

si  $Answer(a, b) \in q(I)$ , alors  $Answer(b, a) \in q(I)$ .

C'est-à-dire,  $q$  est symétrique si pour n'importe quelle base de données, l'ensemble des réponses à  $q$  constitue une relation binaire symétrique.

Donnez une requête conjonctive symétrique. Vous pouvez vous-même choisir les prédicats *edb*.

.../5

**Question 7** Donnez les détails d'une méthode pour vérifier si une requête conjonctive de la forme  $Answer(x, y) \leftarrow B$  est symétrique. Illustrez cette méthode à l'aide de votre réponse à la question 6.

.../5

**Question 8**  $A, B, C, \dots$  sont des entreprises d'autobus (le nombre d'entreprises n'est pas connu a priori). La table *Trajet* donne les trajets assurés par chacune des entreprises. On dit que l'entreprise  $X$  est *omnipotente* si pour chaque atome  $Trajet(ville1,ville2,Y)$  dans la base de données, il est possible d'aller de ville1 à ville2 **en faisant appel seulement à l'entreprise  $X$** . Par exemple,

<i>Trajet</i>	Départ	Arrivée	Entreprise
	Mons	Charleroi	A
	Charleroi	Bruxelles	A
	Bruxelles	Mons	A
	Mons	Ath	A
	Ath	Mons	A
	Mons	Charleroi	B
	Charleroi	Mons	B
	Charleroi	Bruxelles	B
	Ath	Bruxelles	C
	Bruxelles	Ath	C

Dans cet exemple,  $A$  est omnipotente. Notez, par exemple, que l'entreprise  $A$  offre un chemin pour le trajet  $Trajet(Ath,Bruxelles,C)$  :  $Trajet(Ath,Mons,A)$ ,  $Trajet(Mons,Charleroi,A)$ ,  $Trajet(Charleroi,Bruxelles,A)$ .

Donnez un programme datalog stratifié qui calcule un prédicat unaire *Omnipotent*, tel que  $Omnipotent(X)$  est vrai si  $X$  est une entreprise omnipotente. **Expliquez en français l'idée derrière votre programme.**

.../5

**Question 9** Cf. le scénario de la question 8. On dit qu'un trajet  $Trajet(u, v, X)$  est *résistant* si on peut aller de  $u$  à  $v$  (à travers un ou plusieurs trajets) sans faire appel à l'entreprise  $X$ . Par exemple,  $Trajet(\text{Mons}, \text{Ath}, \text{A})$  est résistant parce que l'on peut aller de Mons à Ath sans faire appel à A :  $Trajet(\text{Mons}, \text{Charleroi}, \text{B})$ ,  $Trajet(\text{Charleroi}, \text{Bruxelles}, \text{B})$ ,  $Trajet(\text{Bruxelles}, \text{Ath}, \text{C})$ .

Donnez un programme datalog stratifié qui calcule un prédicat binaire *CheminResistant*, tel que  $CheminResistant(u, v)$  est vrai s'il existe un chemin de  $u$  à  $v$  qui ne traverse que des trajets résistants. Notez : si  $CheminResistant(u, v)$  est vrai, il est possible de voyager de  $u$  à  $v$  sans risque d'être bloqué dans une ville quand une des entreprises se met en grève. **Expliquez en français l'idée derrière votre programme.**

.../5

**Question 10** Soit  $q$  la requête conjonctive suivante :

$$q : Ans(x, y) \leftarrow E(x, u, r), E(u, v, s), E(v, y, s)$$

Soit  $P$  le programme datalog suivant :

$$P : \begin{cases} Ans(x, y) \leftarrow E(x, u, r), E(u, y, r) \\ Ans(x, y) \leftarrow E(x, u, s), Ans(u, y) \end{cases}$$

Indiquez si les expressions suivantes sont vraies ou fausses (cochez deux cases).

	Vrai	Faux
$q \sqsubseteq P$	<input type="checkbox"/>	<input type="checkbox"/>
$P \sqsubseteq q$	<input type="checkbox"/>	<input type="checkbox"/>

Expliquez en détail pourquoi l'expression  $q \sqsubseteq P$  est Vraie/Fausse.

.../5

Expliquez en détail pourquoi l'expression  $P \sqsubseteq q$  est Vraie/Fausse.

.../5

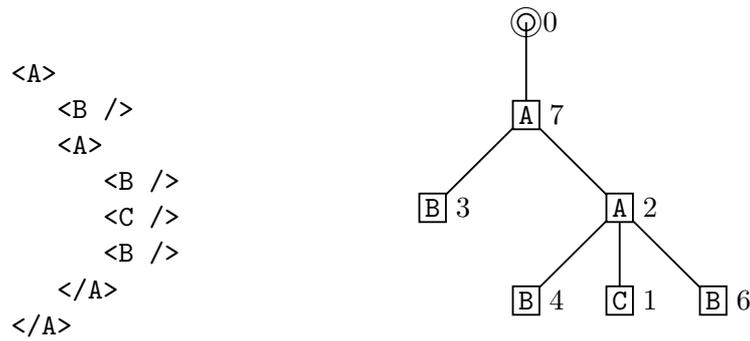


FIGURE 3 – Document et arbre XML.

**Question 11** Dans cet exercice, on encodera en datalog des documents XML et les axes XPath. Pour des raisons de simplicité, on ne considère que deux types de nœud : *the root node* et *element nodes*. Chaque nœud est identifié par un nombre naturel. La numérotation ne suit aucune logique ; la seule contrainte est que deux nœuds différents ne peuvent pas porter le même nombre. L’encodage de l’arbre en datalog utilise quatre prédicats *edb*, à savoir *Root*, *Child*, *NextSibling*, *Label*.

- *Root*( $i$ ) est vrai si le nœud  $i$  est la racine (*the root node*) ;
- *Child*( $i, j$ ) est vrai si le nœud  $j$  est un enfant du nœud  $i$  ;
- *NextSibling*( $i, j$ ) est vrai si  $j$  est un *sibling* de  $i$  qui vient juste après  $i$  ;
- *Label*( $i, X$ ) est vrai si  $i$  est un *element node* de type  $X$ .

La figure 3 montre un document XML (à gauche) et son arbre (à droite) dans lequel les nœuds sont numérotés. Cet arbre sera encodé par la base de données suivante :

$$\{ \textit{Root}(0), \\ \textit{Child}(0, 7), \textit{Child}(7, 3), \textit{Child}(7, 2), \textit{Child}(2, 4), \textit{Child}(2, 1), \textit{Child}(2, 6), \\ \textit{NextSibling}(3, 2), \textit{NextSibling}(4, 1), \textit{NextSibling}(1, 6), \\ \textit{Label}(1, C), \textit{Label}(2, A), \textit{Label}(3, B), \textit{Label}(4, B), \textit{Label}(6, B), \textit{Label}(7, A) \}$$

Écrivez un programme datalog pour le prédicat *idb Answer*, défini comme suit :

$$\textit{Answer}(i, j) \iff \begin{array}{l} \text{si } i \text{ est le nœud de contexte, alors le nœud } j \text{ est} \\ \text{dans } \textit{descendant}::B/\textit{following-sibling}::B, \\ \text{suivant la sémantique d'XPath.} \end{array}$$

Pour l’exemple ci-dessus, on obtient les atomes *Answer*(0,6), *Answer*(7,6) et *Answer*(2,6). **Expliquez en français l’idée derrière votre programme.**

.../10

**Question 12** Soit :

$$\sigma_1 = \forall x \forall y \forall z \forall u \forall w (R(x, y, z), R(x, u, w) \rightarrow R(x, y, w))$$

$$\sigma_2 = \forall x \forall y \forall z \forall u (R(x, y, z), R(x, y, u) \rightarrow u = z)$$

$$\sigma_3 = \forall x \forall y \forall z \forall u \forall w (R(x, y, z), R(x, u, w) \rightarrow z = w)$$

Est-ce que  $\{\sigma_1, \sigma_2\} \models \sigma_3$  ? Détaillez votre réponse.

.../5

	Vrai	Faux
$\{\sigma_1, \sigma_2\} \models \sigma_3$	<input type="checkbox"/>	<input type="checkbox"/>