

# Bases de Données II, Mons

Jef Wijsen

29 mai 2012

Cahier fermé. Durée : 2 heures

Nom et prénom
Année

La figure 2 montre une base de données XML pour stocker les CD possédés par mes amis. Chaque CD (balise CD) est identifié de façon unique par un identifiant (balise ID, avec valeurs *bd85*, *bt88*, *dp82*,...). La base de données stocke, par exemple, que mon ami *Jean* possède un exemplaire de *dh73* et un exemplaire de *rs90*. La figure 1 montre le DTD.

```
<!-- This file is called cd.dtd -->
<!ELEMENT COLLECTIONS (CATALOG, FRIENDS)>
<!ELEMENT CATALOG (CD*)>
<!ELEMENT FRIENDS (FRIEND*)>
<!ELEMENT CD (TITLE, ARTIST, COUNTRY, COMPANY, YEAR)>
<!ELEMENT FRIEND (DISK*)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT ARTIST (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT COMPANY (#PCDATA)>
<!ELEMENT YEAR (#PCDATA)>
<!ELEMENT DISK (#PCDATA)>
<!ATTLIST CD ID CDATA #REQUIRED>
<!ATTLIST FRIEND NAME CDATA #REQUIRED>
<!ATTLIST DISK ID CDATA #REQUIRED>
```

FIGURE 1 – DTD.

**Question 1** Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `texte` dont la valeur est le nom d'un artiste qui a sorti un album avant 1975. Pour le document de la figure 2, la réponse contient `Dr.Hook`.

.../1

**Question 2** Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un ami qui ne possède pas de CD. Pour le document de la figure 2, *An* ne possède pas de CD.

.../1

**Question 3** Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un ami qui possède un CD de Joe Cocker. Pour le document de la figure 2, *Jean* et *Ed* possèdent un CD de Joe Cocker.

.../3

**Question 4** Écrivez une expression XPath (aussi simple que possible) qui rend chaque nœud de type `attribute` dont la valeur est le nom d'un ami qui possède un CD qui se trouve aussi dans la collection de Jean. Pour le document de la figure 2, *Ed* possède un CD que Jean possède aussi (notamment *dh73*).

.../3

```

<?xml version="1.0"?>
<!DOCTYPE COLLECTIONS SYSTEM "cd.dtd">

<COLLECTIONS>
<CATALOG>
  <CD ID="bd85"><TITLE>Empire Burlesque</TITLE><ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY><COMPANY>Columbia</COMPANY><YEAR>1985</YEAR></CD>
  <CD ID="bt88"><TITLE>Hide your heart</TITLE><ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY><COMPANY>CBS Records</COMPANY><YEAR>1988</YEAR></CD>
  <CD ID="dp82"><TITLE>Greatest Hits</TITLE><ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY><COMPANY>RCA</COMPANY><YEAR>1982</YEAR></CD>
  <CD ID="gm90"><TITLE>Still got the blues</TITLE><ARTIST>Gary Moore</ARTIST>
    <COUNTRY>UK</COUNTRY><COMPANY>Virgin records</COMPANY><YEAR>1990</YEAR></CD>
  <CD ID="er97"><TITLE>Eros</TITLE><ARTIST>Eros Ramazzotti</ARTIST>
    <COUNTRY>EU</COUNTRY><COMPANY>BMG</COMPANY><YEAR>1997</YEAR></CD>
  <CD ID="bg88"><TITLE>One night only</TITLE><ARTIST>Bee Gees</ARTIST>
    <COUNTRY>UK</COUNTRY><COMPANY>Polydor</COMPANY><YEAR>1998</YEAR></CD>
  <CD ID="dh73"><TITLE>Sylvias Mother</TITLE><ARTIST>Dr.Hook</ARTIST>
    <COUNTRY>UK</COUNTRY><COMPANY>CBS</COMPANY><YEAR>1973</YEAR></CD>
  <CD ID="rs90"><TITLE>Maggie May</TITLE><ARTIST>Rod Stewart</ARTIST>
    <COUNTRY>UK</COUNTRY><COMPANY>Pickwick</COMPANY><YEAR>1990</YEAR></CD>
  <CD ID="jc87"><TITLE>Unchain my heart</TITLE><ARTIST>Joe Cocker</ARTIST>
    <COUNTRY>USA</COUNTRY><COMPANY>EMI</COMPANY><YEAR>1987</YEAR></CD>
  <CD ID="jc96"><TITLE>Organic</TITLE><ARTIST>Joe Cocker</ARTIST>
    <COUNTRY>USA</COUNTRY><COMPANY>EMI</COMPANY><YEAR>1996</YEAR></CD>
</CATALOG>
<FRIENDS>
  <FRIEND NAME="Jean">
    <DISK ID="dh73"/><DISK ID="rs90"/><DISK ID="jc87"/><DISK ID="jc96"/>
  </FRIEND>

  <FRIEND NAME="Ed">
    <DISK ID="gm90"/><DISK ID="dh73"/><DISK ID="jc87"/>
  </FRIEND>

  <FRIEND NAME="Pierre">
    <DISK ID="gm90"/>
  </FRIEND>

  <FRIEND NAME="An"/>
</FRIENDS>
</COLLECTIONS>

```

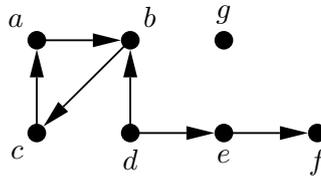
FIGURE 2 – Fichier XML avec des informations sur des propriétaires de CD.

**Question 5** Écrivez un programme XSLT qui affiche, pour chacun de mes ami(e)s, les CD dans sa collection. Le résultat doit être formaté comme suit (la position des blancs et retours à la ligne n'a pas d'importance) :

```
<?xml version="1.0" encoding="UTF-16"?>
<answer>
  <Jean>  <OWNS>Sylvias Mother (Dr.Hook)</OWNS>
          <OWNS>Maggie May (Rod Stewart)</OWNS>
          <OWNS>Unchain my heart (Joe Cocker)</OWNS>
          <OWNS>Organic (Joe Cocker)</OWNS>          </Jean>
  <Ed>    <OWNS>Still got the blues (Gary Moore)</OWNS>
          <OWNS>Sylvias Mother (Dr.Hook)</OWNS>
          <OWNS>Unchain my heart (Joe Cocker)</OWNS>    </Ed>
  <Pierre> <OWNS>Still got the blues (Gary Moore)</OWNS> </Pierre>
  <An></An>
</answer>
```

.../7

**Question 6** Un graphe dirigé est encodé en utilisant le prédicat  $V$  pour les nœuds et le prédicat  $E$  pour les arêtes. Par exemple, le graphe ci-après est encodé par  $\{V(a), V(b), V(c), V(d), V(e), V(f), V(g), E(a,b), E(b,c), E(c,a), E(d,b), E(d,e), E(e,f)\}$ . On exige  $x \neq y$  pour tout arête  $E(x,y)$ .



Un nœud sans arête sortante est appelé *un puit*. Dans l'exemple, les puits sont  $f$  et  $g$ . Écrivez un programme en `datalog¬` (i.e., `datalog` avec négation) qui calcule les puits. Utilisez le prédicat  $Puit$ , i.e.,  $Puit(x)$  est vrai si  $x$  est un puit.

.../3

**Question 7** Expliquez **de façon détaillée** pourquoi le prédicat  $Puit$  de la question 6 ne peut pas être calculé en `datalog` sans négation.

.../3

**Question 8** En faisant appel au prédicat *Puit* de la question 6, écrivez un programme en datalog<sup>7</sup> qui donne chaque nœud  $x$  tel que (i)  $x$  n'est pas un puit et (ii) il existe un chemin de  $x$  vers un puit. Pour le graphe montré ci-dessus, la réponse contient  $d$  et  $e$ . Il ne faut pas répéter les règles définissant *Puit*.

.../3

**Question 9** Pour  $n \in \{2, 3, 4, \dots\}$ , soit  $q_n$  la requête conjonctive suivante :

$$q_n : \text{Answer}(x_1) \leftarrow E(x_1, x_2), E(x_2, x_3), E(x_3, x_4), \dots, E(x_{n-1}, x_n), E(x_n, x_1)$$

Donc,

$$\begin{aligned} q_2 & : \text{Answer}(x_1) \leftarrow E(x_1, x_2), E(x_2, x_1) \\ q_3 & : \text{Answer}(x_1) \leftarrow E(x_1, x_2), E(x_2, x_3), E(x_3, x_1) \\ q_4 & : \text{Answer}(x_1) \leftarrow E(x_1, x_2), E(x_2, x_3), E(x_3, x_4), E(x_4, x_1) \\ & \vdots \end{aligned}$$

Intuitivement,  $q_n$  repère tout nœud qui se trouve dans un cycle de longueur  $n$ . Pour le graphe de la question 6,  $q_2$  et  $q_4$  ne repèrent aucun nœud ;  $q_3$  repère les nœuds  $a$ ,  $b$  et  $c$ .

Déterminez l'ensemble  $A$  des paires  $(i, j)$  telle que  $q_i \sqsubseteq q_j$ . Évidemment,  $A$  contient les paires  $(2, 2)$ ,  $(3, 3)$ ,  $(4, 4)$ , ... Mais quelles sont les autres paires dans  $A$ ? Déterminez votre réponse. En particulier, expliquez les homomorphismes que peuvent exister de  $q_j$  vers  $q_i$ .



**Question 10** Considérez les requêtes  $q_2, q_3, q_4, \dots$  de la question 9. Écrivez un programme  $P$  en datalog tel que pour toute base de données  $I$ , pour tout nœud  $x$ ,

$$Ans(x) \in P(I) \iff \exists j \in \{2, 3, 4, \dots\} : Ans(x) \in q_j(I)$$

Donc, en quelque sorte,

$$P \equiv q_2 \cup q_3 \cup q_4 \cup \dots$$

Notez que l'expression  $q_2 \cup q_3 \cup q_4 \cup \dots$  n'est pas finie et ne correspond donc pas à une "union of conjunctive queries".

.../4