

## Bases de Données II, Mons, 3 juin 2022

NOM + PRÉNOM :

Orientation + Année :

Cet examen contient 11 questions. Durée : exactement 3 heures.

Un fleuriste livre des bouquets à la maison. Chaque espèce de fleur (tulipe, rose...) a un prix exprimé en centimes d'euro. Par exemple, le prix d'une rose est de 150 centimes d'euro, indépendamment de sa couleur ; voir la ligne

```
<fleur fnom="rose" prix="150"/>
```

Un bouquet rassemble des fleurs de différentes espèces et couleurs. Par exemple, le bouquet *Exotique* est composé d'un seul tournesol et trois iris bleus. Voir les lignes :

```
<bouquet bnom="Exotique">
  <fleur fnom="tournesol" nombre="1"/>
  <fleur fnom="iris" couleur="bleu" nombre="3"/>
</bouquet>
```

La couleur n'est pas renseignée si la fleur n'existe qu'en une seule couleur (par exemple, tous les tournesols sont de même couleur). La figure 1 montre le DTD et la figure 2 le document XML.

**Question 1** Écrivez une requête en **XPath** qui renvoie le nom de chaque bouquet contenant au moins deux espèces de fleur distinctes. **Il n'est pas permis d'utiliser des fonctions d'agrégation telles que count, min et max.** Évitez l'usage des axes *parent* et *ancestor*.

Pour le document XML de la figure 2, la réponse est comme suit :

```
bnom="Exotique"
bnom="Printemps"
```

Noter que le bouquet Belge ne contient que des tulipes et n'est donc pas dans la réponse.

---

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

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE fleuriste [
<!ELEMENT fleuriste (fleurs, bouquets)>
<!ELEMENT fleurs (fleur)*>
<!ELEMENT bouquets (bouquet)*>
<!ELEMENT bouquet (fleur)*>
<!ELEMENT fleur (#PCDATA)>
<!ATTLIST fleur fnom CDATA #REQUIRED>
<!ATTLIST fleur prix CDATA #IMPLIED>
<!ATTLIST fleur couleur CDATA #IMPLIED>
<!ATTLIST fleur nombre CDATA #IMPLIED>
<!ATTLIST bouquet bnom CDATA #REQUIRED>
]>
]>

```

FIGURE 1 – DTD

```

<?xml version="1.0" encoding="utf-8"?>
<fleuriste>
  <fleurs>
    <!-- Les prix sont en centimes d'euro -->
    <fleur fnom="tulipe" prix="100"/>
    <fleur fnom="rose" prix="150"/>
    <fleur fnom="iris" prix="250"/>
    <fleur fnom="tournesol" prix="300"/>
  </fleurs>
  <bouquets>
    <bouquet bnom="Valentin">
      <fleur fnom="rose" couleur="rouge" nombre="10"/>
    </bouquet>
    <bouquet bnom="Belge">
      <fleur fnom="tulipe" couleur="noir" nombre="3"/>
      <fleur fnom="tulipe" couleur="jaune" nombre="4"/>
      <fleur fnom="tulipe" couleur="rouge" nombre="6"/>
    </bouquet>
    <bouquet bnom="Exotique">
      <fleur fnom="tournesol" nombre="1"/>
      <fleur fnom="iris" couleur="bleu" nombre="3"/>
    </bouquet>
    <bouquet bnom="Printemps">
      <fleur fnom="rose" couleur="jaune" nombre="10"/>
      <fleur fnom="tulipe" couleur="jaune" nombre="4"/>
    </bouquet>
  </bouquets>
</fleuriste>

```

FIGURE 2 – Document XML

**Question 2** Écrivez une requête en **XPath** qui renvoie le nom de chaque bouquet n'ayant aucune espèce de fleur en commun avec un autre bouquet. **Il n'est pas permis d'utiliser des fonctions d'agrégation telles que count, min et max.** Évitez l'usage des axes *parent* et *ancestor*.

Pour le document XML de la figure 2, la réponse est comme suit :

```
bnom="Exotique"
```

Noter que les bouquets Valentin et Printemps ne sont pas dans la réponse car ils ont une rose en commun ; le bouquet Belge n'est pas dans la réponse car il a une tulipe en commun avec le bouquet Printemps.

---

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

---

**Question 3** Écrivez une requête en **XQuery** qui renvoie le nom du bouquet avec le plus grand nombre de fleurs.

Pour le document XML de la figure 2, la réponse est :

```
bnom="Printemps"
```

En effet, le bouquet Printemps contient 14 fleurs et les autres bouquets contiennent au plus 13 fleurs.

---

.../10
--------

**Question 4** Écrivez un programme en XSLT qui renvoie, pour chaque couleur (sans doublons), les noms des bouquets contenant cette couleur. Les résultats doivent être groupés par couleur, comme suit :

```
<ParCouleur>
  <noir><Bouquet>Belge</Bouquet></noir>
  <rouge><Bouquet>Valentin</Bouquet><Bouquet>Belge</Bouquet></rouge>
  <bleu><Bouquet>Exotique</Bouquet></bleu>
  <j jaune><Bouquet>Belge</Bouquet><Bouquet>Printemps</Bouquet></jaune>
</ParCouleur>
```

L'ordre des couleurs et les retours à la ligne n'ont pas d'importance.

---

.../10
--------

**Question 5** Simplifiez la requête (UCQ) suivante ou expliquez pourquoi aucune simplification n'est possible :

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

Détaillez les calculs.

---

.../10
--------

**Question 6** Un *graphe dirigé simple* est un couple  $G = (V, E)$  avec  $V$  un ensemble fini de *sommets* et  $E$  un ensemble d'*arcs*  $(u, v)$  avec  $u, v \in V$ ,  $u \neq v$ . On traite seulement des graphes sans sommets isolés, i.e., chaque sommet a au moins un arc sortant ou entrant.

Chaque arc du graphe est coloré en une et une seule couleur.

Dans une base de données, un prédicat EDB  $R$  d'arité 3 est utilisé pour stocker les arcs du graphe et leur couleur. Par exemple,  $R(a, c, \text{red})$  signifie qu'il existe un arc dirigé de  $a$  à  $c$  avec la couleur rouge.

On définit un *chemin dirigé* de  $u_1$  à  $u_n$  comme une séquence  $\langle u_1, u_2, \dots, u_n \rangle$  de sommets telle que  $n \geq 2$  et pour chaque  $i \in \{1, 2, \dots, n-1\}$ ,  $(u_i, u_{i+1})$  est un arc du graphe. La *longueur* de ce chemin est de  $n-1$ , i.e., la longueur est le nombre d'arcs parcourus. On dira que ce chemin utilise la couleur  $c$  si au moins un arc du chemin a la couleur  $c$ . Noter que la condition  $n \geq 2$  dans la définition précédente implique que la longueur d'un chemin n'est jamais zéro.

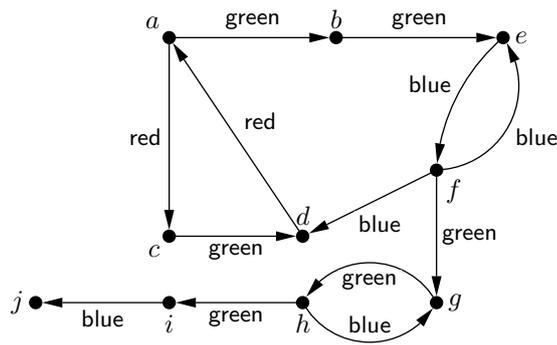


FIGURE 3 – Graphe dirigé avec des arcs colorés.

Par exemple, dans le graphe de la Figure 3,  $\langle a, b, e, f, g \rangle$  est un chemin de  $a$  à  $g$  qui utilise la couleur **green** (et qui utilise aussi la couleur **blue**). Noter qu'un sommet peut se répéter dans un chemin. Par exemple,  $\langle a, c, d, a, b \rangle$  est un chemin de  $a$  à  $b$  qui utilise deux couleurs.

Écrivez un programme en Datalog avec  $\neq$  et négation stratifiée pour le prédicat IDB unaire  $P$  tel que  $P(x)$  est vrai s'il existe un chemin dirigé de  $x$  à  $x$  qui utilise **au plus** deux couleurs (c'est-à-dire, qui utilise une ou deux couleurs, mais pas trois). Par exemple, pour le graphe de la Figure 3, le programme doit renvoyer, entre autres,  $P(a)$  et  $P(e)$ . En effet,  $\langle a, c, d, a \rangle$  est un chemin de  $a$  à  $a$  qui utilise seulement deux couleurs, à savoir **red** et **green**.  $P(e)$  est vrai à cause du chemin  $\langle e, f, e \rangle$  qui n'utilise que la couleur **blue**. Par contre, il est impossible d'aller de  $b$  à  $b$  en n'utilisant que deux couleurs.

Avant de donner votre programme, expliquez en français la stratégie suivie par votre programme. Avant chaque règle (ou groupe de règles), expliquez en français ce que cette règle (ou ces règles) vont calculer.

À titre d'exemple, un étudiant pourrait organiser sa réponse comme suit :

*Stratégie globale* : Pour chaque sommet  $u$ , le programme va parcourir tous les chemins à partir de  $u$ . Lors du parcours d'un tel chemin, on maintient l'ensemble des couleurs déjà utilisées sur ce chemin. Si on revient au sommet  $u$  et la cardinalité de cet ensemble est de 1 ou de 2, alors on ajoute  $u$  à la réponse.

*Les règles, avec explication* : Les deux règles suivantes calculent les sommets sans arc sortant :

```

has-outgoing-arc(X) :- R(X,Y,C).
has-no-outgoing-arc(X) :- R(U,X,C), not has-outgoing-arc(X).

```

La règle suivante etc.

**Attention** : Il n'est pas dit que la stratégie décrite ci-dessus est la bonne : ce n'est pas clair si elle est exprimable en Datalog avec  $\neq$  et négation stratifiée.

*Note* : L'ensemble des couleurs possibles n'est pas fixé. Donc, les constantes **red**, **blue**, **green** ne peuvent pas apparaître dans votre programme !



**Question 7** On dira qu'un sommet  $u$  voit la couleur  $c$  s'il existe un chemin dirigé de  $u$  vers  $u$  qui utilise la couleur  $c$ . Par exemple, pour le graphe de la Figure 3, le sommet  $b$  voit les couleurs **green**, **blue** et **red** à cause du chemin  $\langle b, e, f, d, a, b \rangle$ . Le sommet  $g$  voit seulement les couleurs **blue** et **green**. Le sommet  $i$  ne voit aucune couleur, car il n'existe pas de chemin dirigé de  $i$  à  $i$ .

Écrivez un programme en Datalog avec  $\neq$  et négation stratifiée pour le prédicat IDB unaire  $Q$  tel que  $Q(x)$  est vrai si  $x$  est un sommet qui voit **toutes** les couleurs présentes dans le graphe.

Pour le graphe de la Figure 3, le programme doit renvoyer, entre autres,  $Q(b)$ .

Avant de donner votre programme, expliquez en français la stratégie suivie par votre programme. Avant chaque règle (ou groupe de règles), expliquez en français ce que cette règle (ou ces règles) vont calculer.

*Note* : L'ensemble des couleurs possibles n'est pas fixé. Donc, les constantes **red**, **blue**, **green** ne peuvent pas apparaître dans votre programme!

---

.../10
--------

**Question 8** Pour le schéma

$\{AB, ABC, AC, BCDEF, DE, DF, EF\}$ ,

1. déterminez si, oui ou non, ce schéma est  $\alpha$ -acyclique ;
2. si le schéma est  $\alpha$ -acyclique, donnez un arbre de jointure pour ce schéma.

---

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

**Question 9** Pour le schéma

$\{AB, ABC, AC, BCDEF, DE, DF, EF\}$ ,

(i.e., le même schéma que celui de la question 8), donnez un *full reducer*.

---

.../5

**Question 10** Pour la requête

$$\pi_{AF}(AB \bowtie ABC \bowtie AC \bowtie BCDEF \bowtie DE \bowtie DF \bowtie EF),$$

détaillez un plan d'exécution qui garantit que, pour toute base de données, aucun résultat intermédiaire ne contiendra plus de tuples que  $I \times U$ , avec  $I$  le nombre de tuples dans la base de données et  $U$  le nombre de tuples dans la réponse à la requête.

*Note* : Il n'y a pas besoin de recopier le *full reducer* de la question 9, mais il faut indiquer où ce *full reducer* intervient dans votre plan d'exécution.

---

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

**Question 11** Pour le schéma  $ABCDE$ , déterminez si  $\vDash [ABC, ADE]$  est une conséquence logique de l'ensemble  $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ . Détaillez les calculs.

---

.../10