

## EXAMEN, BASES DE DONNÉES AVANCÉES, JANVIER 2015

*Les documents sont autorisés.*

*Les exercices sont indépendants.*

*Le barème est sur plus de 20 mais il suffit d'avoir 20 points pour avoir 20.*

### Exercice 1 : Questions de Cours - 8 Points

Pour chacune des affirmations suivantes dites si elle est vraie ou fausse et justifier.

1. Pour toute requête conjonctive, on peut écrire une requête équivalente en algèbre relationnelle.
2. Pour toute requête de l'algèbre relationnelle, on peut écrire une requête conjonctive équivalente.
3. On connaît un algorithme polynomial qui teste l'inclusion des requêtes conjonctives.
4. Il existe un algorithme qui permet de tester si deux requêtes de l'algèbre relationnelle sont équivalentes.
5. Les deux requêtes conjonctives suivantes sont équivalentes ( $u, v, x, y, z$  sont des variables et  $a, b, c$  des constantes) :

$$\begin{aligned}q_1(x, y) &\leftarrow R(a, x), Q(u, v), S(z, y, y), T(b, u), R(z, x), Q(x, c), T(b, x), S(a, y, y) \\q_2(z, x) &\leftarrow Q(z, c), R(a, z), Q(z, y), T(b, z), S(a, x, x), S(v, u, u)\end{aligned}$$

6. En algèbre relationnelle la division est une opération monotone.
7. Il existe des requêtes  $datalog^{\neg}$  qui ne sont pas stratifiables.
8. Il existe des requêtes  $datalog^{\neg}$  pour lesquelles la sémantique bien fondée n'est pas définie.

### Exercice 2 : Requêtes Datalog - 6 Points

On considère le schéma de base de données **FAMILLE** contenant les trois relations  $\{\text{PARENT}, \text{HOMME}, \text{FEMME}\}$  telles que :

- $sort(\text{PARENT}) = \{\text{Nom}, \text{Nom}\}$
- $sort(\text{HOMME}) = \{\text{Nom}\}$
- $sort(\text{FEMME}) = \{\text{Nom}\}$

PARENT contient les paires  $(\text{Nom}1, \text{Nom}2)$  telles que  $\text{Nom}1$  est l'un des deux parents de  $\text{Nom}2$ . HOMME (resp. FEMME) contient les noms des personnes qui sont des hommes (resp. des femmes). On demande d'écrire des programmes  $datalog$  ou  $datalog^{\neg}$  (si nécessaire) qui calculent les relations suivantes :

1. PÈRE qui contient les paires  $(Nom1, Nom2)$  telles que  $Nom1$  est le père de  $Nom2$ .
2. FRÈRE-SŒUR qui contient les paires  $(Nom1, Nom2)$  qui sont frères et sœurs (c'est-à-dire qui ont au moins un parent en commun).
3. DEMI-FRÈRE-SŒUR qui contient les paires  $(Nom1, Nom2)$  qui sont frères et sœurs mais qui n'ont qu'un seul parent en commun.
4. COUSINS-DISTANTS qui contient les paires  $(Nom1, Nom2)$  qui ont un ancêtre commun.
5. COUSINS-DISTANTS-MÊME-GÉNÉRATION qui contient les paires  $(Nom1, Nom2)$  qui ont un ancêtre commun et qui sont exactement de la même génération par rapport à cet ancêtre.
6. COUSINS-DISTANTS-MATERNEL qui contient les paires  $(Nom1, Nom2)$  qui ont un ancêtre commun qui n'est trouvable qu'en suivant les lignées maternelles de  $Nom1$  et  $Nom2$ .
7. RELIÉS qui contient les paires  $(Nom1, Nom2)$  qui sont reliés dans l'arbre généalogique (au sens large, c'est-à-dire qu'on a le droit de monter et descendre plusieurs fois dans l'arbre pour faire le lien).

### Exercice 3 : Stratifications - 3 Points

Pour chacun des programmes *datalog* suivants dites si il est stratifiable. Justifiez votre réponse et si celle-ci est positive, donnez une stratification.

1.

$$\begin{aligned} R(x) &\leftarrow \neg Q(x, a) \\ Q(x, y) &\leftarrow S(y), T(y, b) \\ T(x, y) &\leftarrow R(y), Q(c, x) \end{aligned}$$

2.

$$\begin{aligned} Q(x, z) &\leftarrow T(z, y, v), T(x, y, a), R(x, y) \\ R(x, y) &\leftarrow Q(x, y), T(b, v, x), \neg S(x, y) \\ T(x, y, u) &\leftarrow R(x, u), T(x, y, c) \\ S(x, z) &\leftarrow R(z, b), \neg T(b, v, x) \end{aligned}$$

3.

$$\begin{aligned} Q(x, z) &\leftarrow T(z, y, v), T(x, y, a), \neg R(x, y) \\ R(x, y) &\leftarrow R(y, x), T(b, v, x) \\ T(x, y, u) &\leftarrow R(x, u), T(x, y, c) \\ S(x, z) &\leftarrow R(z, b), \neg T(b, v, x) \\ P(y, x) &\leftarrow P(a, x), \neg S(x, y), Q(x, y) \end{aligned}$$

## Exercice 4 : Mots - 5 Points

On considère un alphabet  $A = \{a, b, c\}$  et le schéma de base de données **MOTS** contenant la relation binaire " $<$ " et les relations unaires  $P_a, P_b$  et  $P_c$ . On code tout mot  $a_1 \cdots a_n$  sur l'alphabet  $A$  comme une instance du schéma **MOTS** de la façon suivante :

- Le domaine est l'ensemble  $\mathbf{dom} = \{1, \dots, n\}$ .
- La relation " $<$ " contient tous les couples  $(i, j) \in \mathbf{dom}^2$  tels que  $i < j$ .
- La relation  $P_a$  contient tous les  $i \in \mathbf{dom}$  tels que  $a_i = a$ .
- La relation  $P_b$  contient tous les  $i \in \mathbf{dom}$  tels que  $a_i = b$ .
- ...

Par exemple le mot  $abac$  est codé par l'instance ayant pour domaine  $\{1, 2, 3, 4\}$  et telle que " $<$ " =  $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$ ,  $P_a = \{1, 3\}$ ,  $P_b = \{2\}$  et  $P_c = \{4\}$ .

Une requête du calcul relationnel sans variables définit donc un langage de mots sur l'alphabet  $A$  : le langage de tous les mots qui satisfont la requête. Par exemple, la requête  $\exists x P_a(x)$ , définit le langage de tous les mots qui contiennent un " $a$ " ( $A^*aA^*$ ). On demande d'écrire des requêtes du calcul relationnel qui définissent les langages suivants :

1. Les mots qui contiennent un  $a$  puis un  $b$ , puis un  $c$  :  $A^*aA^*bA^*cA^*$ .
2. Les mots qui ne contiennent que des " $a$ " et des " $c$ ".
3. Les mots qui finissent par un  $a$  :  $A^*a$ .
4. Les mots qui contiennent le facteur  $ab$  :  $A^*abA^*$ .
5. les mots qui contiennent au plus deux " $c$ ".
6. Les mots qui ne contiennent pas le facteur  $ac$  entre deux  $b$ .
7. Les mots de la forme  $ababababababab \cdots ab$  :  $(ab)^*$ .