Bases de Données II, Mons, 7 juin 2021
Partie « avant la pause » (8H30-10H30)

NOM + PRÉNOM :	
Orientation + Année :	

Cette partie de l'examen contient 6 questions.

Le document XML de la Figure 1 est utilisé pour stocker le palmarès des cyclistes les plus importants de l'histoire du cyclisme. La DTD est incluse au début du document XML. Par exemple, l'élément suivant indique qu'Eddy Merckx est né en 1945 et qu'en 1975, il a gagné (`classement="1"`) la course abrégée comme MSM.

```
<cycliste nom="Eddy Merckx" naissance="1945">
   <participation cid="MSM" annee="1975" classement="1"/>
</cycliste>
```

L'élément suivant indique que MSM est un raccourci pour la course Milan-San Remo :

```
<course cid="MSM">Milan-San Remo</course>
```

Dans chaque course, les coureurs sont classés 1, 2, 3, ... sans ex æquo (i.e., deux coureurs ne peuvent pas avoir le même rang).

Question 1 Certains cyclistes se spécialisent exclusivement en une seule course (par exemple, exclusivement Milan-San Remo ou exclusivement Liège-Bastogne-Liège). Disons qu'un cycliste est un *monocliste* s'il n'a jamais participé à des éditions de deux courses différentes. Pour le document XML de l'exemple, le seul monocliste est Francesco Moser : il a participé deux fois à Milan-San Remo, mais n'a jamais participé à une autre course que Milan-San Remo.

Écrivez une requête en XPath qui renvoie le nom de chaque cycliste qui est un monocliste. **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 l'exemple, la réponse est comme suit :

```
nom="Francesco Moser"
```

.../5

Question 2 Écrivez une requête en XPath qui renvoie le(s) pire(s) résultat(s) d'Eddy Merckx. **Il n'est pas permis d'utiliser des fonctions d'agrégation telles que min et max.** Évitez l'usage des axes *parent* et *ancestor*.

Pour le document XML de l'exemple, la réponse est comme suit :

```
<participation cid="PR" annee="1972" classement="7"/>
```

En effet, aucun élément `participation` d'Eddy Merckx n'a une valeur de `classement` supérieure à 7.

.../5

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE PALMARES [
<!ELEMENT PALMARES (COURSES, CYCLISTES)>
<!ELEMENT COURSES (course)*>
<!ELEMENT CYCLISTES (cycliste)*>
<!ELEMENT cycliste (participation)*>
<!ELEMENT course (#PCDATA)>
<!ELEMENT participation (#PCDATA)>
<!ATTLIST course cid CDATA #REQUIRED>
<!ATTLIST cycliste nom CDATA #REQUIRED>
<!ATTLIST cycliste naissance CDATA #REQUIRED>
<!ATTLIST participation cid CDATA #REQUIRED>
<!ATTLIST participation annee CDATA #REQUIRED>
<!ATTLIST participation classement CDATA #REQUIRED>
]>

<PALMARES>
<COURSES>
   <course cid="LBL">Liège-Bastogne-Liège</course>
   <course cid="MSM">Milan-San Remo</course>
   <course cid="PR">Paris-Roubaix</course>
</COURSES>
<CYCLISTES>
   <cycliste nom="Roger De Vlaeminck" naissance="1947">
      <participation cid="LBL" annee="1970" classement="1"/>
      <participation cid="PR" annee="1972" classement="1"/>
      <participation cid="LBL" annee="1975" classement="8"/>
   </cycliste>
   <cycliste nom="Eddy Merckx" naissance="1945">
      <participation cid="LBL" annee="1970" classement="3"/>
      <participation cid="PR" annee="1972" classement="7"/>
      <participation cid="LBL" annee="1975" classement="1"/>
      <participation cid="MSM" annee="1975" classement="1"/>
   </cycliste>
   <cycliste nom="Francesco Moser" naissance="1951">
      <participation cid="MSM" annee="1975" classement="2"/>
      <participation cid="MSM" annee="1984" classement="1"/>
   </cycliste>
</CYCLISTES>
</PALMARES>
```

FIGURE 1 – Les palmarès des cyclistes.

Question 3 Écrivez un programme en XSLT qui renvoie les gagneurs (i.e., les cyclistes avec `classement="1"`) des différentes éditions des courses. Les résultats doivent être groupés par course, comme suit :

```
<WINNERS>
   <RACE name="Liège-Bastogne-Liège">
      <EDITION year="1970">Roger De Vlaeminck</EDITION>
      <EDITION year="1975">Eddy Merckx</EDITION>
   </RACE>
   <RACE name="Milan-San Remo">
      <EDITION year="1975">Eddy Merckx</EDITION>
      <EDITION year="1984">Francesco Moser</EDITION>
   </RACE>
   <RACE name="Paris-Roubaix">
      <EDITION year="1972">Roger De Vlaeminck</EDITION>
   </RACE>
</WINNERS>
```

Notez que les balises sont en anglais.

.../10

Question 4 Écrivez une requête en XQuery qui renvoie les courses où Roger De Vlaeminck a battu Eddy Merckx.

Pour le document XML de l'exemple, la réponse devrait être formatée comme suit :

```
<RogerAvantEddy>
   <Course edition="1970">Liège-Bastogne-Liège</Course>
   <Course edition="1972">Paris-Roubaix</Course>
</RogerAvantEddy>
```

Notez, par exemple, qu'Eddy Merckx était deux positions derrière Roger De Vlaeminck dans l'édition 1970 de Liège-Bastogne-Liège.

.../10

Question 5 Écrivez une requête en XQuery qui renvoie les noms des coureurs qui ont gagné le plus grand nombre de courses.

Pour le document XML de l'exemple, la réponse devrait être formatée comme suit :

```
<Big-Winners>
   <Winner nom="Roger De Vlaeminck"/>
   <Winner nom="Eddy Merckx"/>
</Big-Winners>
```

Notez que dans le document XML qui sert comme exemple, Roger de Vlaeminck et Eddy Merckx ont gagné deux courses, et aucun coureur n'a gagné trois courses.

.../10

Question 6 Simplifiez la requête (UCQ) suivante ou expliquez pourquoi aucune simplification n'est possible :

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

Détaillez les calculs.

.../10

Bases de Données II, Mons, 7 juin 2021
Partie « après la pause » (11H30-13H30)

NOM + PRÉNOM :

Orientation + Année :

Cette partie de l'examen contient les questions 7 à 12.

Question 7 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 colorié 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, $R(a, c, \mathsf{red})$ signifie qu'il existe un arc dirigé de a à b avec la couleur rouge.

On définit un *chemin dirigé de* u_1 *à* u_n comme une séquence $\langle u_1, u_2, \ldots, u_n \rangle$ de sommets telle que $n \geq 2$ et pour chaque $i \in \{1, 2, \ldots, n-1\}$, (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.

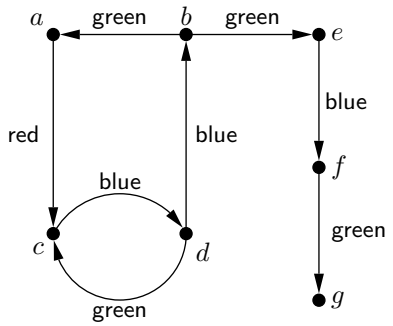

FIGURE 2 – Graphe dirigé avec des arcs colorés.

Par exemple, dans le graphe de la Figure 2, $\langle a, c, d, b \rangle$ est un chemin de a à b qui utilise la couleur red (et qui utilise aussi la couleur blue).

Écrivez un programme en Datalog avec $\neq$ et négation stratifiée pour le prédicat IDB binaire P tel que $P(x, y)$ est vrai s'il existe un chemin dirigé de x à y qui utilise **exactement** deux couleurs. Par exemple, dans le graphe de la Figure 2, $P(c, g)$ est vrai, mais $P(a, e)$ est faux. En effet, $\langle c, d, b, e, f, g \rangle$ est un chemin de c à g qui utilise exactement deux couleurs, à savoir blue et green. Par contre, il est impossible d'aller de a à e en utilisant exactement deux couleurs. Noter qu'un chemin peut traverser un même sommet plusieurs fois. Par exemple, $P(c, b)$ est vrai à cause du chemin $\langle c, d, c, d, b \rangle$.

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 atteint un sommet v et la cardinalité de cet ensemble est de 2, alors on ajoute (u, v) 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 ! On peut bien sûr obtenir toutes les couleurs avec une règle :

```
color(C)  :-  R(X,Y,C).
```

.../10

Question 8 Écrivez un programme en Datalog avec $\neq$ et négation stratifiée pour le prédicat IDB binaire Q tel que $Q(x, y)$ est vrai s'il existe un chemin de x à y et, en plus, tout chemin de x à y utilise toutes les couleurs présentes dans le graphe.

Pour le graphe de la Figure 2, le programme doit renvoyer $Q(a, a)$, $Q(a, e)$, $Q(a, f)$, $Q(a, g)$, $Q(b, b)$, et $Q(b, d)$.

Noter, par exemple, que $Q(c, c)$ n'est pas dans la réponse car le chemin $\langle c, d, c\rangle$ n'utilise pas la couleur red. $Q(g, e)$ n'est pas dans la réponse car il n'existe même pas de chemin de g à e.

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.

.../10

Question 9 Disons qu'un chemin de x à y est *ennuyeux* si les trois conditions suivantes sont toutes satisfaites :

1. la longueur du chemin est paire (i.e., la longueur est dans $\{2, 4, 6, 8, \ldots\}$) ;
2. le chemin utilise exactement deux couleurs ; et
3. chaque deux arcs qui se suivent immédiatement sur le chemin sont de couleurs différentes.

De façon informelle, un chemin ennuyeux commence avec deux arcs de couleurs différentes, disons c_1 et c_2 ; si le chemin continue, il alterne entre les couleurs c_1 et c_2.

Par example, le chemin $\langle d, b, e, f, g\rangle$ est un chemin ennuyeux dans la Figure 2 : ce chemin est de longueur 4, un nombre pair, et les couleurs des arcs alternent entre blue et green.

Écrivez un programme en Datalog avec $\neq$ et négation stratifiée pour le prédicat IDB binaire S tel que $S(x, y)$ est vrai si les deux conditions suivantes sont respectées :

1. il existe un chemin, *de longueur paire* (et non-zéro), de x à y ; et
2. aucun chemin de x à y n'est ennuyeux.

Pour le graphe de la Figure 2, le programme doit renvoyer $S(a, a)$, $S(a, e)$, $S(a, g)$, $S(b, b)$, $S(c, b)$, $S(c, f)$:

— $S(a, a)$ est dans la réponse parce que $\langle a, c, d, b, a\rangle$ est un chemin de longueur 4 (i.e., de longueur paire) de a à a, et il n'existe pas de chemin ennuyeux de a à a.
— $S(a, d)$ n'est pas dans la réponse parce que $\langle a, c, d\rangle$ est un chemin ennuyeux de a à d.
— $S(a, b)$ n'est pas dans la réponse parce qu'aucun chemin de a à b est de longueur paire.

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.

.../10

Question 10 Pour le schéma

$$\{ABCE, ABG, ADEF, DEH\},$$

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

.../5

Question 11 Pour le schéma

$$\{ABCE, ABG, ADEF, DEH\},$$

(i.e., le même schéma que celui de la question 10), donnez un *full reducer*.

.../5

Question 12 Pour la requête

$$\pi_{CD}(ABCE \bowtie ABG \bowtie ADEF \bowtie DEH),$$

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 11, mais il faut indiquer où ce *full reducer* intervient dans votre plan d'exécution.

.../5