

Examen MIF18 - Bases de données avancées - 15 décembre 2011

Durée : 1h30

Documents autorisés

**Numéro de copie :**

*Il faut rendre ces feuilles en les glissant dans votre copie anonyme.* Ne pas l'utiliser comme brouillon. Les réponses sont à donner sur ces feuilles, pas dans la copie. Remplir le champ ci-dessus avec le *numéro de la copie* dans laquelle vous allez la glisser. Remplir la partie d'anonymat de la copie, puis coller le coin.

### **Exercice 1: Datalog**

On considère les trois relations suivantes :

**Employé(x)**  
**A-Comme-Responsable(x,y)**  
**Responsable (y)**

**Question 1:** Ecrire, en Datalog, la requête permettant de retourner tous les employés ayant le même responsable que "Smith".

**Question 2:** Ecrire, en Datalog, la requête permettant de retourner toutes les personnes (les personnes regroupent les employés et les responsables)

**Question 3:** Ecrire, en Datalog, la requête permettant de retourner tous les responsables qui sont des employés avec un niveau de responsabilité supérieur ou égal à deux (c'est-à-dire que chaque responsable retourné est responsable au moins d'un autre responsable).

## Exercice 2: Inclusion de requêtes

Soient deux (2) requêtes  $q_1$  et  $q_2$ , spécifiées sur le même schéma de base de données.

**Question 1:** Quand  $q_1$  est-elle incluse dans  $q_2$  (que l'on écrit  $q_1 \subseteq q_2$ ) ?

**Question 2:** Pour chacun des cas suivants, indiquez (en entourant OUI ou NON) si  $q_1 \subseteq q_2$  ?

$q_1(x) \leftarrow R(x, u), R(u, v), R(v, w)$ $q_2(x) \leftarrow R(x, u), R(u, v)$	<table border="1" style="margin: auto;"> <tr><td>OUI</td></tr> <tr><td>NON</td></tr> </table>	OUI	NON	$q_1(x) \leftarrow R(x, u), R(u, v), R(v, x)$ $q_2(x) \leftarrow R(x, u), R(u, x)$	<table border="1" style="margin: auto;"> <tr><td>OUI</td></tr> <tr><td>NON</td></tr> </table>	OUI	NON
OUI							
NON							
OUI							
NON							
$q_1(x) \leftarrow R(x, u), R(u, u)$ $q_2(x) \leftarrow R(x, u), R(u, v)$	<table border="1" style="margin: auto;"> <tr><td>OUI</td></tr> <tr><td>NON</td></tr> </table>	OUI	NON	$q_1(x) \leftarrow R(x, u), R(u, \text{Smith})$ $q_2(x) \leftarrow R(x, u), R(u, v)$	<table border="1" style="margin: auto;"> <tr><td>OUI</td></tr> <tr><td>NON</td></tr> </table>	OUI	NON
OUI							
NON							
OUI							
NON							

## Exercice 3: Réécriture de requêtes

**Question 1:** On considère les trois relations suivantes :

**Employé(x)**  
**A-Comme-Responsable(x,y)**  
**Responsable (y)**

Soient les deux vues suivantes :

**L(x,y) :- A-Comme-Responsable (x,u), A-Comme-Responsable (u,y)**  
**E(x,y) :- A-Comme-Responsable (x,y), Employé (y)**

Soit la requête suivante :

**Q(x,y) :- A-Comme-Responsable (x,u), A-Comme-Responsable (u,v),**  
**A-Comme-Responsable (v,w), A-Comme-Responsable (w,y),**  
**Employé(y)**

Peut-on répondre (et comment) à la requête Q si nous ne disposons que des vues L et E?

**Question 2:** On considère une base de données avec les relations **Vente**, **Produit**, **Magasin** et **Temps** :

**Ventes**(ID-produit, ID-magasin, ID-temps, Montant, ...)  
**Produit**(ID-produit, Catégorie, ...)  
**Magasin**(ID-magasin, Région, ...)  
**Temps**(ID-temps, Heure, Jours, Année,...)

On considère la vue suivante :

```
CREATE VIEW VentesAnnuelles AS
  SELECT Année, Catégorie, Région, sum(Montant)
  FROM Ventes, Produit, Magasin, Temps
  WHERE Ventes.ID-produit = Produit.ID-produit
        AND Ventes.ID-magasin = Magasin.ID-magasin
        AND Ventes.ID-temps = Temps.ID-temps
  GROUP BY Année, Catégorie, Région
```

Soit la requête Q suivante :

```
SELECT Catégorie, Région, sum(Montant)
FROM Ventes, Produit, Magasin, Temps
WHERE Ventes.ID-produit = Produit.ID-produit
      AND Ventes.ID-magasin = Magasin.ID-magasin
      AND Ventes.ID-temps = Temps.ID-temps
      AND Temps.Année = 2009
GROUP BY Catégorie, Région
```

Peut-on réécrire la requête Q en utilisant la vue **VentesAnnuelles**. Si oui, donnez la requête Q' qui résulte de la réécriture de Q en utilisant la vue **VentesAnnuelles**. Si non, dites pourquoi.

#### Exercice 4: Optimisation sémantique

On considère la base de données déductive suivante, composée d'une partie EDB, d'une partie IDB et de contraintes d'intégrité :

##### EDB

- E1** : Employé(Nom, Salaire, Nom\_département)  
*un employé est identifié par un nom, son salaire et le département où il exerce*
- E2** : Dept(Nom\_département, Fonction, Nom\_responsable)  
*un département est identifié par un nom, une activité (fonction), et le nom du responsable*
- E3** : Emp-Véhicules(Numéro-immatriculation, appartient-à, Couleur)  
*les véhicules personnels sont identifiés par un numéro, le propriétaire (qui est un employé), et une couleur*
- E4** : Entreprise-Véhicules(Propriétaire, Numéro-immatriculation, Dépt, Couleur, Kilométrage)  
*un véhicule d'entreprise est identifié par l'entreprise d'appartenance, un numéro, un département au sein de l'entreprise, une couleur, et le nombre de kilomètres*

##### IDB :

- A1** : Travaille-sous-direction-de(n1, n2)  $\leftarrow$  Employé(n1, s1, d), Dept(d, f1, n2).
- A2** : Propriétaires-Véhicules(o, d, t, c)  $\leftarrow$  Employé(o, s, d), Emp-Véhicules(t, o, c).
- A3** : Propriétaires-Véhicules(o, d, t, c)  $\leftarrow$  Entreprise-Véhicules(o, t, d, c, mi).

IC :

(IC1) Les salaires de tous les employés sont inférieurs à 70000 et supérieurs à 10000.

**Question 1:** Exprimez sous forme de clause(s) cette contrainte d'intégrité.

(IC2) Le nom de l'employé détermine fonctionnellement (dépendance fonctionnelle) le salaire et le nom du département.

**Question 2:** Exprimez sous forme de clause(s) cette contrainte d'intégrité.

(IC3) Tout véhicule dans la relation Emp-Véhicules appartient à un employé.

**Question 3:** Exprimez sous forme de clause(s) cette contrainte d'intégrité.

**Question 4:** Ecrire, sous forme de clause, la requête suivante :

La liste des propriétaires (nom) de voitures rouges qui travaillent dans un département publicité (fonction).

**Question 5:** Donnez les axiomes sémantiquement contraints (élaborés à partir des contraintes d'intégrité) pour la base de données. Il s'agit de réécrire les EDB et les IDB en tenant compte de l'interaction de ceux-ci avec les contraintes d'intégrité.

### Exercice 5: SPARQL

On considère un graphe RDF contenant, entre autres, les prédicats suivants, donnés avec leur signification intuitive.

- `http://voitures.com/#modele` le sujet a pour modèle l'objet. Le sujet peut être un véhicule ou un moteur.
- `http://voitures.com/#nbplaces` le sujet (un véhicule) a pour nombre de places l'objet (ce dernier est une valeur).
- `http://voitures.com/#marque` le sujet (un modèle) a pour marque l'objet (ce dernier est une valeur).
- `http://voitures.com/#nom` le sujet a pour nom l'objet.
- `http://voitures.com/#moteur` le sujet a pour moteur l'objet. Si le sujet est un véhicule, l'objet est un moteur ; si le sujet est un modèle, l'objet est un modèle.

**Question 1:** Écrire une requête SPARQL permettant d'obtenir les véhicules de plus de 4 places de marque "Peugeot".

**Question 2:** Écrire une requête SPARQL permettant d'obtenir les véhicules dont le modèle de moteur ne correspond pas au modèle du modèle.

**Question 3:** On suppose que seuls les prédicats mentionnés ci-dessus apparaissent dans le graphe et on suppose également que le graphe vérifie les contraintes intuitives associées aux prédicats. Donner un schéma relationnel permettant de représenter les informations concernant les véhicules, moteurs, marques et modèles (y compris les noms et les nombres de places). On souhaite que les prédicats ne soient pas représentés par des valeurs mais par des relations (autrement dit on s'interdit le schéma Triplet(Sujet,Prédicat,Objet) ).



## Exercice 6: XML et relationnel

On considère le schéma relationnel suivant :

```
WAGON(num INTEGER PRIMARY KEY, type VARCHAR(20))
MOTRICE(num INTEGER PRIMARY KEY, modele VARCHAR(20))
TRAIN(num INTEGER PRIMARY KEY, motrice INTEGER REFERENCES MOTRICE(num))
COMPOSE(train INTEGER REFERENCES TRAIN(num), place INTEGER,
         wagon INTEGER REFERENCES WAGON(num), PRIMARY KEY (train,place))
```



On considère la DTD suivante :

```
<!ELEMENT trains (train*)>
<!ELEMENT train (motrice,wagon*)>
<!ATTLIST train num CDATA #REQUIRED>
<!ELEMENT motrice (#PCDATA)>
<!ATTLIST motrice num CDATA #REQUIRED>
<!ELEMENT wagon (#PCDATA)>
<!ATTLIST wagon num CDATA #REQUIRED>
```

On suppose que le texte dans les motrices indique leur modèle et que celui des wagons indique leur type.

On suppose donnée une fonction XQuery `view(nom)` qui étant donné un nom de table renvoie un document XML ayant la forme suivante :

```
<nom>
  <ROW><att1>val1</att1><att2>val2</att2>...</ROW>
  <ROW><att1>val3</att1><att2>val4</att2>...</ROW>
  ...
</nom>
```

**Question 1:** Ecrire une requête XQuery qui génère un document XML conforme à la DTD ci-dessus et contenant les informations stockées dans une base conforme au schéma relationnel.

**Question 2:** Si on remplace le type des attributs `num` de la DTD par `ID`, la transformation reste-elle possible dans tous les cas ? Justifier.