

# Examen de Bases de Données Relationnelles

Décembre 2007

SI 4 et MAM 4

## 1 Requêtes SQL

On utilise le schéma relationnel suivant pour stocker des informations concernant des droits d'accès à des tables d'une base de données

- Utilisateur (id, nom, prenom)
  - id : clé de Utilisateur, nombre entier
  - nom de l'utilisateur, chaîne de caractères
  - prenom de l'utilisateur, chaîne de caractères
- Table (nomTable, idCreateur)
  - nomTable: clé de Table, chaîne de caractères
  - idCreateur: identifiant du créateur de la table
- Privilège (id, nomTable, idUtilisateur, droit)
  - id: clé de privilège, nombre entier
  - nomTable: identifiant de table
  - idUtilisateur : identifiant d'utilisateur
  - droit  $\in \{SELECT, INSERT, UPDATE, DELETE, ALL\}$

1. Décrire en SQL la création de la table Privilège, en exprimant toutes les contraintes d'intégrité mentionnées ci-dessus..

---

```
CREATE TABLE Privilège(  
    id INTEGER PRIMARY KEY ,  
    nomTable VARCHAR (30),  
    idUtilisateur INTEGER,  
    droit VARCHAR (10) CHECK (droit IN('SELECT','INSERT','DELETE','UPDATE','ALL')),  
    FOREIGN KEY (nomTable) REFERENCES Table,  
    FOREIGN KEY (idUtilisateur) REFERENCES Utilisateur);
```

---

2. Ecrire une requête SQL permettant de lister les noms et prénoms des utilisateurs ayant le droit de détruire la table 'Marques'

---

```
SELECT nom, prenom  
FROM Utilisateur U, Privilège P  
WHERE nomTable='Marques' AND (droit='DELETE' OR droit='ALL') AND U.id = P.idUtilisateur
```

---

3. Quel est le nom et le prénom de l'utilisateur qui a créé le plus de tables ? (Si plusieurs utilisateurs détiennent le même record, les lister tous)

---

```
CREATE VIEW GrandsCreateurs as (SELECT idCreateur , count(*) FROM Table
  GROUP BY idCreateur
  HAVING count(*) >= ALL
    (SELECT count(*) from Table
     GROUP BY idCreateur
    )
  ) ;

SELECT nom, prenom
  FROM Utilisateur U, GrandsCreateurs G
 WHERE U.id =G.idCreateur;
```

---

4. Donnez les noms des tables pour lesquelles le créateur de la table n'a pas le droit 'DELETE'.

---

```
SELECT nomTable
FROM Table T
WHERE idCreateur NOT IN
  (SELECT idUtilisateur
   FROM Privilege P
   WHERE (droit='DELETE' OR  droit='ALL') AND P.nomTable=T.nomTable)
```

---

5. Quelles sont les noms des tables sur lesquelles au moins 4 utilisateurs ont le droit 'ALL'?

---

```
SELECT nomTable FROM Privilege WHERE droit='ALL' GROUP BY nomTable
  HAVING COUNT(*) >= 4
```

---

6. Quels sont les utilisateurs (donner leur nom et leur prénom) ayant le droit 'ALL' sur toutes les tables?

---

```
SELECT idUtilisateur , count(*)
  FROM Privilege p
 WHERE droit='ALL'
 GROUP BY idUtilisateur
 having count(*)>= all select count(*) from table
```

---

## 2 Algèbre relationnelle

On considère les mêmes relations que pour l'exercice précédent. Exprimer en algèbre relationnelle les requêtes suivantes:

1. Noms et prénoms des utilisateurs n'ayant le droit 'all' sur aucune table

---


$$\Pi_{nom, prnom} Utilisateur \bowtie (\Pi_{id} Utilisateur \setminus \Pi_{id} Utilisateur \sigma_{droit=ALL} Privilege)$$


---

2. Utilisateurs ayant le droit 'all' sur toutes les tables qu'ils ont créés

---


$$\Pi_{id} Utilisateur \setminus \Pi_{id} Utilisateur (\delta_{idCreateur \leftarrow idUtilisateur} \bowtie \sigma_{droit \neq ALL} Privilege)$$


---

3. nom des tables qui ne peuvent pas être détruites par leur créateur

---


$$\Pi_{nomTable}(x) \setminus \Pi_{nomTable} (\Pi_{nomTable, idUtilisateur} (\sigma_{droit=delete \vee droit=all} Privilege) \bowtie \sigma_{idCreateur \leftarrow idUtilisateur} Table)$$


---

## 3 Conception de schéma

Une société de location de véhicules veut gérer ses agences, véhicules (voitures, motos, vélos) et locations avec une base de données relationnelle. Voici la proposition d'un schéma relationnel avec sept tables :

- Agence(idAg, ville, region, nomResp)
- Voiture(idVeh, idAg, idTar, marque, modele, numImmat)
- Moto(idVeh, idAg, idTar, marque, modele, numImmat)
- Velo(idVeh, idAg, idTar, type)
- Client(idCl, nom, ville, region)
- Location(idCl, idVeh, debut, fin)
- Tarif(idTar, prix)

1. L'une de ces tables ne peut pas avoir une clé atomique, laquelle et pourquoi?
2. Quel est l'intérêt de la table Tarif, pourquoi ne pas remplacer idTar par un prix dans les tables Moto, Velo, et Voiture ?

- 
- Agence(idAg, ville, region, nomResp)
  - Voiture(idVeh, idAg, idTar, marque, modele, numImmat)
  - Moto(idVeh, idAg, idTar, marque, modele, numImmat)
  - Velo(idVeh, idAg, idTar, type)
  - Client(idCl, nom, ville, region)
  - Location(idCl, idVeh, debut, fin)
  - Tarif(idTar, prix)
-

## 4 Formes normales

1. On donne la relation :  $R(A, B, C, D, E, F)$  et l'ensemble de dépendances fonctionnelles :

$$DF_1 = (ABC \rightarrow DEF; BC \rightarrow E; E \rightarrow F)$$

- (a) Quelle est la forme normale de R?

---

Première forme: E dépend d'une sous clé (donc pas en deuxième)

---

- (b) Comment peut-on transformer R pour la mettre sous forme normale juste supérieure ?

---

Deux tables: (A,B,C,D) et (B,C,E,F)

---

- (c) Même questions avec l'ensemble de dépendances :

$$DF_2 = (AB \rightarrow CDEF; F \rightarrow C; D \rightarrow F)$$

---

Deuxième forme normale.

En 3NF, on a 3 tables: (A,B,D,E), (D,F) et (F,C)

---

2. Soit la relation  $R(A,B,C,D,E,F,G,H,I,J,K)$  et l'ensemble des dépendances :

$$AB \rightarrow CDH; C \rightarrow E; D \rightarrow FG; H \rightarrow IJ; J \rightarrow K$$

Décomposer cette relation de façon à obtenir un schéma en troisième forme normale sans perte de dépendances.

---

En 3NF, on a 5 tables:

(A,B, C,D,H); (C, E) ; D, F,G) ; H, I,J) ; J, K)

---

## 5 TP1, question 10

1. Pour calculer “ dans chacune des classes, le/les propriétaires ayant le plus grand nombre de marque dans cette classe”, nous avons proposé dans la correction du TP1, la requête suivante:

```
select classe, S.nom, S.pays, count( * )
  from marque as M1, societe AS S
 where M1.prop = S.id
 group by classe, S.nom, S.pays
 having count( * ) >= all
(select count ( * ) from marque AS M2
 where M2.classe = M1.classe
   group by prop);
```

Comment peut on modifier très simplement cette requête pour recherche “ dans chacune des classes, le/les éventuels propriétaires qui ont un monopole marque dans cette classe, c'est à dire qui possède toutes les marques de la classe”

---

Il suffit d'enlever le dernier “group by” !

---

2. Donner une autre solution à cette dernière requête basée sur la division:

- (a) Donner la formulation en algèbre relationnelle pour chaque marque

---

L'algèbre relationnelle ne permet d'écrire une solution générique. Pour chaque marque  $x$ , l'ensemble recherché est défini par:

$$\Pi_{prop, id} marque \div \Pi_{id}(\sigma_{classe=x} marque)$$

---

- (b) Donner une implémentation en SQL basée sur la négation

---

```
select distinct M1.classe, S.nom, S.pays
  from marque as M1, societe AS S
 where M1.prop = S.id
 AND NOT EXISTS
      (select *   from marque AS M2
       where M2.classe = M1.classe
       and M1.prop <> M2.prop);
```

---