

Exercice A (*SQL*)

Le schéma relationnel suivant représente la gestion simplifiée de librairies de bande dessinées (BD).

BD(NumBD, NomAlbum, NBPages, Editeur NumDessinateur, NumScénariste)
Auteur(NumAuteur, Prénom, Nom, Nationalité)
Magasin(NumMag, NomMag, Adresse, Téléphone, NomResponsable)
BDDansMagasin(NumBD, NumMagasin, NbBDDispo, PrixVente)

(A.i) Donnez les noms des auteurs de bande dessinées (en tant que scénariste ou dessinateur) qui sont responsables d'un magasin.

```
select Auteur.Nom
from Auteur, Magasin
where Auteur.Nom = Magasin.NomResponsable
```

(A.ii) Donnez les noms des magasins dont le responsable est aussi un des auteurs d'une des BD disponibles dans ce même magasin.

```
select Magasin.Nom
from Magasin, BD, Auteur, BDDansMagasin
where Magasin.NomResponsable=Auteur.Nom and
(BD.NumDessinateur=Auteur.NumAuteur
or
BD.NumScenariste=Auteur.NumAuteur) and
BDDansMagasin.NumBD= BD.NumBD and
BDDansMagasin.NbBDDispo>0
```

(A.iii) Donnez la liste des titres d'albums de BD disponibles dans les magasins, regroupés par magasin.

```
select BD.NomAlbum
from BD, BDDansMagasin
where BD.NumBD = BDDansMagasin.NumBD and BDDansMagasin.NbBDDispo>0
group by NumMag
```

(A.iv) * Donnez la liste des numéros des magasins ayant le plus grand nombre d'albums de BD disponibles. [On expliquera sa réponse, et le sens de chaque sous-requête s'il y en a.]

```

select T.NumMag
from T as (select NumMag, sum(NBBDDispo) as Total
from BDDansMagasin
groupby NumMag)
where T.Total >= all (select sum(NBBDDispo)
from BDDansMagasin
groupby NumMag)

```

La sous requête:

```

(select NumMag, sum(NBBDDispo) as Total
from BDDansMagasin
groupby NumMag)

```

fabrique les couples N,T où T est le nombre d'albums disponibles dans le magasin numéro NumMag. Ensuite on cherche les numéros de magasins dont le nombres d'albums disponibles est supérieur ou égal à tous ceux des autres magasins: on a donc la même sous-requête que précédemment dont on ne garde que le nombre d'albums disponibles.

(A.v) ★ Donnez la liste des noms de magasins ayant le plus grand nombre de titres d'albums de BD disponibles. [On expliquera sa réponse, et le sens de chaque sous-requête s'il y en a.]

C'est un peu la même chose, mais on ne compte que le nombre de titres disponibles, c'est-à-dire de NumBD différents dans BDDansMagasin avec NbBDDispo>0, et on doit retrouver les noms des magasins — à la place du "select" le plus externe, on peut aussi utiliser "natural join".

```

select NomMag
from Magasin, U as
(select T.NumMag
from T as (select NumMag, count(distinct NbBDDispo) as NTitre
from BDDansMagasin
where NbBDDispo>0
groupby NumMag)
where T.NTitre>= all (select count(distinct NBBDDispo)
from BDDansMagasin
where NbBDDispo>0
groupby NumMag)
)
where Magasin.NumMag=U.NumMag

```

Exercice B (Algèbre relationnelle)

Donner des principes d'optimisation des requêtes comportant les opérateurs σ , π , \bowtie . A quoi sert cette optimisation des requêtes? Quand est-elle utilisée par le système de gestion de bases de données? Expliquer par un exemple sa pertinence. [Sur environ une demi page.]

Pour plus de détails, voir le cours.

Une table ayant en général bien plus de ligne que de colonnes, il convient de faire en premier les sélections, puis les projections, qui suppriment des colonnes, et c'est en dernier, après que la taille des tables a été réduite autant que possible, que sont effectués les jointures et les produits — ces dernières opérations multiplie le nombre de lignes des deux tables.

Cette phase d'optimisation intervient après la compilation des requêtes SQL en algèbre relationnelle et avant qu'elles ne soient exécutées.

Exercice C (*Algèbre relationnelle*)

On dispose de deux relations $R(A,B,C)$ et $S(C,D,E)$.

Pour chacune des requêtes suivantes, donner une requête équivalente plus efficace,

(C.i) $\pi_E(R \bowtie S)$

$\pi_E(\pi_C(R) \bowtie \pi_{CE}(S))$

(C.ii) $\pi_{BDE}(R \bowtie S)$.

$\pi_{BDE}(\pi_{BC}R \bowtie S)$

(C.iii) $\sigma_{[D=3]}(R \bowtie S)$

$R \bowtie \sigma_{[D=3]}(S)$

(C.iv) $\sigma_{[C=2]}(R \bowtie S)$

$\sigma_{[C=2]}(R) \bowtie \sigma_{[C=2]}(S)$

(C.v) $\sigma_{[B=1 \text{ AND } C=2]}\pi_{B,C,D}(R \bowtie S)$

$\pi_{AC}\sigma_{[B=1 \text{ AND } C=2]}(R) \bowtie \pi_{CE}\sigma_{[C=2]}(S)$
--

Exercice D (*Algèbre relationnelle*)

On rappelle qu'étant données deux relations, $R(A_1, \dots, A_p, A_{p+1}, \dots, A_n)$ (n attributs) et $S(A_{p+1}, \dots, A_n)$ (les $n-p$ derniers attributs de R) la division de R par S est une relation dont les attributs sont A_1, \dots, A_p et qui est définie par:

$$R \div S = \{(a_1, \dots, a_p) \mid \forall (a_{p+1}, \dots, a_n) \in S \quad (a_1, \dots, a_p, a_{p+1}, \dots, a_n) \in R\}$$

Dans les deux premières questions les relations sont des ensembles de n -uplets, et les opérations utilisées sont celles de l'algèbre relationnelle ensembliste.

(D.i) Que vaut $(T \times S) \div S$?

T

(D.ii) ★ Exprimer la division à l'aide du produit, de la projection et de la différence, en justifiant la formule obtenue.

<p>Un classique! Soit $R1p = \pi_{A_1, \dots, A_p}(R)$ que vaut $R1p \times S$? Ce sont les n-uplets qui sont obtenus en prolongeant n'importe quel p-uplet constitué des p premiers éléments d'un n-uplet de R par tous les $(n-p)$-uplets de S — si ces p-premiers éléments étaient dans $R \div S$, tous les n-uplets ainsi obtenus seraient dans R. $W = \pi_{A_1, \dots, A_p}((R1p \times S) - R)$ contient donc tous les p-uplets qui sont dans $R1p$ mais pas dans $R \div S$. Donc $R \div S = R1p - W$</p>

(D.iii) * Cette définition est-elle encore valable lorsqu'on fonctionne, comme SQL, avec des multiensembles et les opérations sur les multiensembles? Pourquoi?

Et non, à cause de la double différence! Il faut utiliser un opérateur δ qui élimine les doublons.

Exercice E (*Datalog*)

Le tramway fait des siennes. Une relation $Acces(x,y,n,E)$, mise à jour régulièrement, indique que la station y suit la station x sur la ligne n , et que le tramway est dans l'état E de x à y : si E vaut 1 alors la ligne fonctionne de x à y , et sinon la ligne ne fonctionne pas de x à y .

[On répondra en **datalog**. On vérifiera la correction des clauses en particulier celle des clauses récursives, et on expliquera la signification des relations auxiliaires utilisés.]

(E.i) Quels sont les triplets (x,y,n) tels que la station y suit la station x sur la ligne n .

Suit(x,y,n) :- Acces(x,y,n,_).

(E.ii) Quels sont les triplets (x,y,n) tels que la station y suit la station x sur la ligne n , et que le tramway fonctionne de x à y .

Possible(x,y,n) :- Acces(x,y,n,1).

(E.iii) Quels sont les couples de stations consécutives qui sont reliées dans un sens et dans l'autre par le tramway?

R1(x,y) :- Possible(x,y,n) AND Possible(y,x,n).

(E.iv) Quels sont les couples (x,y) de stations tels que l'on peut aller de x à y en tramway en suivant la même ligne?

S2(x,y,n) on peut aller de x à y en suivant la ligne n.
S2(x,y,n) :- Possible(x,y,n).
S2(x,y,n) :- S2(x,z,n) AND Possible(z,y,n).

La réponse:

R2(x,y) :- S2(x,y,_).

Attention, dans S2 le numéro de ligne est constant.

(E.v) Quels sont les couples (x,y) de stations tels que l'on peut aller de x à y en tramway, en changeant de ligne si besoin est?

$R3(x,y) : \neg \text{Possible}(x,y, _)$.
 $R3(x,y) : \neg R3(x,z) \text{ AND Possible}(z,y, _)$

(E.vi) Quels sont les couples de stations (x,y) tels que l'on puisse aller en tramway de x à y mais pas revenir en tramway de y à x ?

$S4(x,y) : \neg R3(x,y) \text{ AND NOT } R3(y,x)$.

Toute variable apparaît dans un atome positif. Il y a une négation mais pas de récurrence.

(E.vii) Un voyageur situé à la station u veut bien faire en tout au plus une station de tramway à pieds. A quelles stations peut-il accéder?

$R5(u)$.
 $R5(x) : \neg R3(u,x)$.
 $R5(x) : \neg R3(u,z) \text{ AND Suit}(z,w, _) \text{ AND } R3(w,x)$.
 $R5(x) : \neg \text{Suit}(z,w, _) \text{ AND } R3(w,x)$.
 $R5(x) : \neg R3(u,z) \text{ AND Suit}(z,w, _)$.
 $R5(x) : \neg \text{Suit}(u,x, _)$.

Exercice F (Conception de schémas relationnels)

Donner un exemple de schéma relationnel qui soit en 3NF mais pas en BCNF. On expliquera pourquoi il est en 3NF et pas en BCNF, et en quoi cela peut poser problème. [Sur environ une demi page.]

Un exemple classique est Ville, Code postal, Rue avec les dépendances $V,R \rightarrow C$ $C \rightarrow V$ (la dépendance $C \rightarrow V$ fonctionne avec les villes comme Paris, Lyon, Marseille où le code postal détermine la commune).
 Les clefs sont VR et RC . Le schéma est en 3NF: tous ces attributs sont premiers (appartiennent à une clef minimale) et donc chaque partie droite d'une dépendance est un attribut premiers. Par contre il n'est pas BCNF puisque $C \rightarrow V$ sans que C contienne une clef.
 Le problème qu'on rencontre est une redondance de l'information: si un n-uplet est $(v,r1,c)$ et un autre $(?,r2,c)$ alors, à cause de $C \rightarrow V$ on sait que "?" vaut v , c'est donc une information redondante qu'on ne peut pas factoriser.

Exercice G (Conception de schémas relationnels)

On considère le schéma relationnel suivant, Cours, Prof, Heure, Salle, Etudiant, Note dont les dépendances fonctionnelles sont

Cours	→	Prof	Heure Salle	→	Cours
Heure Prof	→	Salle	Cours Etudiant	→	Note
Heure Etudiant	→	Salle			

(G.i) ★ Mettre ce schéma en BCNF — on expliquera l'algorithme utilisé.

Corrigé disponible à part sur le web.

(G.ii) La décomposition obtenue préserve-t-elle les dépendances fonctionnelles? Pourquoi?

Idem.

(G.iii) La décomposition obtenue est-elle sans perte par jointure? Pourquoi? [Rappel: Soit I un ensemble fini d'attributs, soit J_k des sous-ensembles de I pour $1 \leq k \leq K$. Soit R un schéma relationnel sur I . La décomposition de $R(I)$ en $(R_k)_{1 \leq k \leq K}$ est dite *sans perte par jointure* si et seulement si $\bowtie_{k=1}^{k=K} \pi_{J_k}(R) = R$.]

Oui, ce genre de décomposition est toujours sans perte par jointure. A chaque fois que l'on fractionne un schéma T en XA et $T - A$ on a $X \rightarrow A$, il n'y a donc pas de perte par jointure (théorème de Heath ou vérification par l'algorithme matriciel).