

Bases de Données Relationnelles

Examen de Décembre 2006

SI 2 et MAM 2

1 Exercice 1 (8 points)

Soit le schéma relationnel suivant :

```
Produit(code_produit, nom, prix, quantite_en_stock)
Client(code_client, nom, prenom, adresse)
Commande(code_commande, date_reception, code_client)
Ligne_commande(code_commande, code_produit, quantite )
```

1. Écrire en SQL les requêtes qui affichent les données suivantes :

- (a) Toutes les lignes (nom_produit, quantité_commande, prix_unitaire, prix_total) correspondant à la commande de code_commande=1.

```
SELECT P.nom as nom_produit , L.quantite as quantite_commande, P.prix as prix_unitaire,
       P.prix*L.quantite as prix_total
FROM ligne_commande as L, produit as P
WHERE P.code_produit = L.code_produit AND L.code_commande=1;
```

- (b) Le nom et l'adresse de toutes les personnes ayant commandé au moins un produit de valeur unitaire supérieure ou égale à 100 euros depuis le 15/12/2006.

```
SELECT distinct Cl.nom, Cl.adresse
FROM client Cl, ligne_commande L , commande Co, produit P
WHERE Cl.code_client=Co.code_client AND
      Co.code_commande=L.code_command AND
      P.code_produit = L.code_produit AND
      P.prix >= 100 AND
      Co.date_reception > 2006-12-15 ;
```

- (c) Le code_client, nom, et prénom du client ayant effectué la commande dont le montant total est le plus grand.

```
CREATE VIEW commande_avec_total AS
SELECT L.code_commande AS code_commande, Co.code_client AS code_client,
SUM (P.prix*L.quantite) AS prix_total, Co.date_reception AS date
FROM ligne_commande AS L, produit AS P, commande AS Co
WHERE P.code_produit = L.code_produit and L.code_commande =Co.code_commande
GROUP BY L.code_commande , Co.code_client, Co.date_reception;
```

```
SELECT Cat.code_client, Cl.nom, Cl.prenom
FROM Client Cl, commande_avec_total Cat
WHERE Cl.code_client = Cat.code_client AND
Cat.prix_total IN (SELECT max(Cat.prix_total) FROM commande_avec_total Cat);
```

-
- (d) Le montant total de la dernière commande du client Pierre Durand. Que se passe-t-il si plusieurs clients portent ce nom?
-

```
SELECT Cat.prix_total FROM Client Cl, commande_avec_total Cat
WHERE Cl.code_client = Cat.code_client AND
Cl.nom='durand' and Cl.prenom= 'pierre' AND
cat.Date IN
(SELECT Max(Cat.date)
FROM commande_avec_total Cat , client Cl
WHERE Cl.code_client = Cat.code_client AND
Cl.nom='durand' AND Cl.prenom= 'pierre');
```

-
- (e) Les clients n'ayant jamais rien commandé.
-

```
SELECT Cl.code_client FROM client Cl
EXCEPT
SELECT Co.code_client FROM commande Co;
```

2. Exprimer en algèbre relationnelle les requêtes 1, 2 et 5.
-

Requête 1:

$\delta_{nom_produit \leftarrow nom, quantite_commande \leftarrow quantite, prix_unitaire \leftarrow prix, prix_total \leftarrow quantite * prix}$
 $\Pi_{nom, quantite, prix, quantite * prix} [\sigma_{code_commande=1}(\overline{Ligne_commande}) \bowtie Produit]$

Requête 2:

$\Pi_{nom, adresse} [(\Pi_{code_produit}(\sigma_{prix \geq 100}(Produit)) \bowtie \Pi_{code_commande, code_produit}(Ligne_commande))$
 \bowtie
 $(\Pi_{code_commande}(\sigma_{date \geq 15/12/06}(Commande)) \bowtie Client)]$

Requête 5:

$\Pi_{code_client}(Client) \setminus \Pi_{code_client}(Commande)$

2 Exercice 2 (3 points)

On considère le schéma relationnel R(A,B,C,D,E,F,G,H,I,J) et l'ensemble des dépendances fonctionnelles suivantes :

DF = { $A \rightarrow B$; $AB \rightarrow CG$; $AB \rightarrow DE$; $DE \rightarrow F$; $H \rightarrow I$; $H \rightarrow J$ }

1. Trouver l'unique clé de R et démontrer son unicité

A et H n'apparaissant dans aucune partie droite de dépendance fonctionnelle doivent appartenir à toutes les clés de la relation. Comme $\{A, H\}^+ = \{A, B, C, D, E, F, G, H, I, J\}$, $\{A, H\}$ est l'unique clé de R

2. Quelle est la forme normale de R ?

Si R n'est pas en 3NF proposer une décomposition en 3NF.

R est 1NF, tous ses attributs étant unitaires. Elle n'est pas en 2NF : par exemple $H \rightarrow J$ est une violation de la deuxième forme normale, H étant une sous clé.

Il faut donc commencer par trouver une décomposition de R en 2NF.

De la sous clé H dépendent I et J, pour supprimer cette violation de la 2NF, on peut décomposer R en deux tables $R_1 = \{H, I, J\}$ et $R_2 = \{A, B, C, D, E, F, G, H\}$. R_1 a pour unique clé H et la seule dépendance fonctionnelle est $H \rightarrow I, J$, R_1 est donc en 3NF.

Les dépendances fonctionnelles de R_2 sont $DF_2 = \{A \rightarrow BCDEFG; DE \rightarrow F\}$, l'unique clé est AH. R_2 ne vérifie pas la deuxième forme normale.

On décompose à nouveau R_2 en deux tables $R_3 = \{A, H\}$ et $R_4 = \{A, B, C, D, E, F, G\}$.

R_3 a pour unique clé AH est en 3NF.

Les dépendances de R_4 sont $DF_4 = \{A \rightarrow BCDEFG; DE \rightarrow F\}$, et son unique clé est A. R_4 est bien en 2NF mais pas en 3NF à cause de la dépendance $DE \rightarrow F$.

On décompose R_4 en deux tables $R_5 = \{D, E, F\}$ et $R_6 = \{A, B, C, D, E, G\}$.

R_5 a pour unique clé DE, et sa seule dépendance fonctionnelle est $DE \rightarrow F$. R_5 est donc en 3NF.

R_6 a pour unique clé A, et sa seule dépendance fonctionnelle est $A \rightarrow BCDEG$. R_6 est donc en 3NF.

Une décomposition possible en 3NF pour R est donc $R_1 = \{\underline{H}, I, J\}$, $R_3 = \{\underline{A}, H\}$, $R_5 = \{\underline{D}, \underline{E}, F\}$ et $R_6 = \{\underline{A}, B, C, D, E, G\}$.

3 Exercice 3 (2 points)

Soit R(a,b) et S(b,c) deux relations. Prouver la correction ou donner un contre exemple pour les formules suivantes:

- $R = \Pi_{\{a,b\}}(R \bowtie S)$
- $R = \Pi_a(R) \bowtie \Pi_b(R)$

Soit, $R : \{(a_1, b_1), (a_2, b_2)\}$, $S : \{(b_1, c_1), (b_1, c_2)\}$

alors on a:

$\Pi_{\{a,b\}}(R \bowtie S) = \{(a_1, b_1)\}$ et $\Pi_a(R) \bowtie \Pi_b(R) = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$

4 Gestion des abonnés d'un fournisseur d'accès (9 points)

Le fournisseur d'accès à Internet "SI-World" utilise PostgreSQL pour gérer les informations concernant les internautes abonnés. Voici le schéma relationnel utilisé.

- Abonnés(Login, Nom, Prénom, NumTél, Ville)

Login est un identificateur unique attribué par le fournisseur d'accès à chaque utilisateur. Les autres attributs correspondent au nom, prénom, numéro de téléphone et à la ville de résidence de l'abonné.

- Abonnements(Login, Type, EspaceMax, AdressesMax, EspaceU, AdressesU) avec:

- Type est le type d'abonnement (Gratos, WeekEnd, DeLuxe) choisi par le client.
- EspaceMax et AdressesMax correspondent à l'espace maximale (en MO) et au nombre maximal d'adresses email pour chaque type d'abonnement (e.g., pour Type = 'Gratos', EspaceMax = 5 et AdressesMax = 10).

- EspaceU et AdressesU correspondent à l'espace disque (en MO) et le nombre d'adresses email utilisées par le client.
- Connexions(Login, Date, Heure, Durée). Cette table contient des informations sur les connexions pour chaque utilisateur : ils correspondent à la date (format 'jj/mm/aaaa'), l'heure (format 'hh:mm') et la durée (en minutes) de chaque connexion effectuée par l'abonné.

4.1 Requêtes

1. Écrire en SQL les requêtes qui affichent les données suivantes :

- (a) Le nom des abonnés qui se sont connectés le mois de mai 2000 et qui utilisent plus d'espace disque que celui autorisé par le type de leur abonnement.

```
SELECT A.Nom FROM Abonnés A, Abonnements B, Connexions C
      WHERE A.Login=B.Login AND B.Login=C.Login AND C.Date >= '2000-05-01'
      AND C.Date <= '2000-05-31' AND B.EspaceU > B.EspaceMax;

ou

SELECT A.Nom FROM Abonnés A WHERE A.Login IN
      (SELECT C.Login FROM Connexions C
      WHERE C.Date >= '2000-05-01' AND C.Date <= '2000-05-31' )
      AND A.Login IN
      (SELECT B.Login FROM Abonnements B WHERE B.EspaceU > B.EspaceMax);
```

- (b) Le login des abonnés qui ne se sont pas connectés depuis le 1er mai 2000 .

```
SELECT A.Login FROM Abonnés A WHERE NOT EXISTS
      (SELECT * FROM Connexions C WHERE C.Date >= '2000-05-01' AND C.Login =
      A.Login);

ou

SELECT A.Login FROM Abonnés A WHERE A.Login NOT IN
      (SELECT C.Login FROM Connexions C WHERE C.Date >= '2000-05-01');
```

- (c) Le login et la durée de la plus longue connexion depuis le 1er janvier 2000.

```
SELECT C.Login, C.Durée FROM Connexions C WHERE C.Date >= '2000-01-01' AND
      C.Durée >= ALL (SELECT D.Durée FROM Connexions D WHERE C.Date >='2000-01-01');

ou

SELECT C.Login, C.Durée FROM Connexions C WHERE C.Date >= '2000-01-01' AND
      C.Durée = (SELECT MAX(D.Durée) FROM Connexions D WHERE C.Date >='2000-01-01');
```

- (d) Pour chaque abonné, le login et la somme des durées de ses connexions pendant le mois de décembre 2000, si cette somme est supérieure à 10h.

```
SELECT C.Login, SUM(C.Durée) FROM Connexions C
      WHERE C.Date >= '2000-12-01' AND C.Date <= '2000-12-31'
      GROUP BY C.Login
      HAVING SUM(C.Durée) > 600;
```

2. Écrire en calcul relationnel des tuples (CRT) les requêtes 1 et 3.

Requêtes 1:

$\Pi_{Nom} [Abonnes \bowtie \sigma_{'2000-05-01' \leq Date \leq 2000-05-01} Abonnements \bowtie \sigma_{EspaceU > EspaceMax} Connexions]$

Requêtes 2:

$A = \Pi_{duree} (Connexions) \quad B = \delta_{d' \leftarrow duree} A$

$\Pi_{login, duree} [(A \setminus \Pi_{duree}(\sigma_{d' < duree}(A \bowtie B))) \bowtie Connexions]$

4.2 Dépendances Fonctionnelles

Pour chacune des relations ci-dessus, on a défini un ensemble de dépendances fonctionnelles (on a remplacé le nom des attributs par leur initiale) :

Abonnés(L, N, P, NT, V)

$DF_{Abonns} = \{L \rightarrow (N, P, NT, V); NT \rightarrow L\}$

Abonnements(L, T, EM, AM, EU, AU)

$DF_{Abonnements} = \{T \rightarrow (EM, AM); L \rightarrow (T, EU, AU)\}$

Connexions(L, DA, H, D)

$DF_{Connexions} = \{L, DA, H \rightarrow D\}$

1. Quelles sont la (les) clé(s) des relations ?

(a) NT et L sont les clés de Abonnés car :

- $\{NT\}^+ = \{L\}^+ = \{L, N, P, NT, V\}$

- Il s'agit d'ensembles qui ne contiennent qu'un seul élément et qui ne sont donc pas décomposables

(b) L est l'unique clé de Abonnements: $\{L\}^+ = \{L, T, EM, AM, EU, AU\}$

(c) $\{L, DA, H\}$ est l'unique clé de Connexions: $\{L, DA, H\}^+ = \{L, DA, H, D\}$

2. Donner la forme normale de la relation Abonnements. Faites une décomposition si nécessaire.

La relation Abonnements est sous 1ère et 2ème forme normale. Elle n'est pas sous 3ème forme normale car EM et AM dépendent de T qui n'est pas un attribut d'une clé. Une décomposition en 3ème forme normale est donnée par :

R1: {L,T,EU,AU} qui pour clé L

R2: {T,EM,AM} qui pour clé T
