

Université Paris 7

Master 1 Informatique, Bases de données avancées.

29 mai janvier 2015

Durée : 2h30 Documents manuscrits, notes de cours, notes de TD/TP autorisés. Livres, ordinateurs, téléphones portables interdits.

Le sujet comporte 5 pages.

Le barème est donné à titre indicatif.

On considère une relation $R(A, B, C, D, E)$ qui satisfait les dépendances fonctionnelles suivantes :

$A, B \rightarrow C$
 $C, D \rightarrow E$
 $E \rightarrow A$

On considère une relation R satisfaisant les dépendances fonctionnelles suivantes :

Donner toutes les clés candidates de la relation R . Justifier. (Justifier ne veut pas dire systématiquement toutes les clôtures possibles.)

Question 1: Donner toutes les clés candidates de la relation R . Justifier. (Justifier ne veut pas dire qu'il faut calculer toutes les clôtures possibles.)

Question 2: Est-ce que cette relation est BCNF ? Justifier la réponse.

Question 3: Est-ce que cette relation est 3NF ? Justifier la réponse.

Exercice 2 - Triggers et fonctions - 10 points

Une banque gère les comptes des clients. La base de données de la banque est composée de plusieurs tables :

```
create table client(  
    id_client int primary key check(id_client > 0),  
    nom varchar(30) not null,  
    prenom varchar(30)  
);  
  
create table compte(  
    id_compte int primary key check(id_compte >= 0),  
    type_compte char(1) not null check(type_compte in ('C', 'T', 'O')),  
    date_fin date,  
    solde decimal(12,2) not null,  
    plafond decimal(6,2) not null  
);  
  
create table compte_client(  
    id_client int references client,  
    id_compte int references compte,  
    primary key(id_client, id_compte)
```

```

create table operation(
  id_compte int references compte,
  date_op timestamp not null,
  valeur decimal(8,2) not null,
  type_oper char(1) check(type_oper in ('c','b','v'))
);

```

client La table client regroupe les clients de la banque.

compte La table compte contient tous les comptes ouverts. L'attribut **solde** contient le solde du compte. L'attribut **plafond** contient le plafond, c'est le solde maximal du compte. Le plafond peut être aussi bien supérieur ou inférieur à zéro. Si le plafond est inférieur à zéro cela veut dire que la banque autorise le découvert, par exemple le plafond de -300 signifie que le compte peut être à découvert jusqu'à 300 euros. Le plafond peut être aussi positif, par exemple le plafond de 1000 euros veut dire que le client doit garder au moins 1000 euros sur le compte.

L'attribut **type_compte** indique le type du compte.

C - compte courant (ne rapporte pas d'intérêts).

C - compte rémunéré avec l'argent disponible à tout moment.

T - compte rémunéré par capitalisation, par conséquent l'argent ne sort pas du compte avant la date donnée par l'attribut **date_fin**. Tout retrait avant la date **date_fin** donne lieu aux pénalités.

compte_client Cette table établit le lien entre les comptes et les propriétaires des comptes.

Notons que chaque compte peut avoir plusieurs propriétaires (par exemple compte commun des époux).

operation La table operation enregistre des opérations sur les comptes (débit ou crédit). L'attribut **date_op** indique le moment de l'opération. L'attribut **valeur** donne la valeur de l'opération, les valeurs négatives correspondent aux opérations de débit (retrait de l'argent, paiement par carte ou par chèque), les valeurs positives correspondent aux opérations de crédit (versement de l'argent sur le compte).

L'attribut **type_oper** indique comment l'opération est effectuée :

- c - par chèque,
- b - paiement par carte,
- e - versement en espèce.

Question 1: Écrire un trigger qui sera déclenché par une opération d'insertion dans la table operation (il faut écrire aussi bien une fonction PL/pgSQL qui implémente le trigger que la commande **create trigger**.)

On considère deux situations :

- (1) valeur de l'opération est positive. Dans ce cas l'opération est toujours effectuée et le trigger ajoute valeur de l'opération au solde du compte dans la table compte.
- (2) Si valeur est négative alors le trigger :
 - effectue l'opération si le solde après l'opération reste supérieur égal au plafond. Le trigger doit modifier le solde du compte dans la table compte en ajoutant valeur au solde (rappelons que valeur est négative ici).

~~si le solde après l'opération deviendrait inférieur au plafond l'opération doit être annulée. L'opération annulée ne sera pas enregistrée dans la table operation. L'insertion de l'opération sera refusée et sera annulée.~~

De plus si le `type_oper` est c alors notre chèque était sans provision et il faut appliquer la pénalité de 30 euros que le trigger doit déduire du solde du compte. La pénalité s'applique toujours même si la déduction de la pénalité donne le solde inférieur au plafond.

Par contre l'annulation des opération b et v se fait sans pénalités.

Question 2: Certains clients sont traités de façon privilégiée, les opérations de débit qui pouvait conduire à l'annulation de l'opération (faute de solde suffisant) sont étalées sur plusieurs comptes du client.

Supposons par exemple que le client possède trois comptes avec les `id_compte` 1, 2 et 3. Supposons que l'opération de débit (valeur de l'opération -100) doit être effectuée sur le compte 1 et que cette opération est impossible à réaliser (solde deviendrait inférieur au plafond). Par contre il est possible de débiter le compte 1 de 50 euros, le compte 2 de 20 euros et le compte 3 de 30 euros. Donc au total il est possible de débiter 100 euros en étalant l'opération sur plusieurs comptes du client.

Écrire la fonction

```
decimal(10,2) maxdebit(nc compte.id_compte%TYPE)
```

qui retourne la somme maximale qu'on peut retirer du compte nc sans dépassement du plafond.

Question 3: Écrire la fonction

```
boolean debiter(nc compte.id_compte%TYPE,  
               somme DECIMAL(10,2),  
               t operation.type_oper%TYPE)
```

qui implémente l'opération de débit d'un compte nc de la somme somme. Si nécessaire l'opération sera étalée sur plusieurs comptes dont le propriétaire est le même que le propriétaire du compte nc.

~~On utilisera l'algorithme le plus simple possible :~~

~~1. si la somme peut être débitée du compte nc alors la fonction debiter insère l'opération correspondante dans la table operation et termine exécutée.~~

2. Sinon trouver id_client du propriétaire du compte nc. Trouver tous les comptes appartenant à cette personne et calculer la somme totale qui peut être débitée de tous ces comptes. Si cette somme est inférieure à la valeur du paramètre `somme` de la fonction `debiter` alors terminer en retournant FALSE.
3. Dans le cas contraire la fonction `debiter` insère dans la table operation les opérations de débit sur les différents comptes du client jusqu'à ce que la valeur totale de débits atteigne somme. Le premier compte à débiter sera le compte nc. Les autres comptes peuvent débiter dans n'importe quel ordre mais le solde de chaque compte doit rester supérieur ou égal au plafond.

7 - 6.5 points

document books.xml. Pour comprendre la structure
 tit extrait :

```
<?xml version="1.0" standalone="no" ?>
<?xml-stylesheet type="text/css" href="books.css" ?>
```

```
<title>
<author>
</publisher>
```

```
</title>
</author>
</publisher>
```

```
<commentaires>
```

```
<com idlivre="1">
```

```
<note>5</note>
```

```
<blurb>
```

This is the best book I know about basic algorithms
 and data structures.

I refer to it frequently.

```
</blurb>
```

```
</com>
```

```
<com idlivre="1">
```

```
<note>5</note>
```

```
<blurb>
```

j'ai rien compris de ce livre, c'est nul.

```
</blurb>
```

```
</com>
```

```
<com idlivre="2">
```

```
<note>19</note>
```

```
<blurb>
```

A fascinating look at why C++ is what it is.

```
</blurb>
```

```
</com>
```

```
<com idlivre="2">
```

```
<note>4</note>
```

```
<blurb>
```

Impossible to learn C++ from this book. Useless.

```
</blurb>
```

```
</com>
```

```
</commentaires>
```

```
</document>
```

Exercice 3 - XPath et XQuery

Les questions ci-dessous sont basées sur le doc
 de fichier books.xml nous donnons ici un pe

```
<?xml version="1.0" standalone="no" ?>
<?xml-stylesheet type="text/css" href="books.css" ?>
<document>
<books>
```

```
<livre idlivre="1">
```

```
<title>Algorithms in C++</title>
```

```
<author>Robert Sedgewick</author>
```

```
<publisher>Addison-Wesley</publisher>
```

```
</livre>
```

```
<livre idlivre="2">
```

```
<title>The Design and Evolution of C++</title>
```

```
<author>Bjarne Stroustrup</author>
```

```
<publisher>Addison-Wesley</publisher>
```

```
</livre>
```

Le fichier complet contient beaucoup plus de livres et beaucoup plus de commentaires et de notes donnés par des lecteurs.

est un catalogue de livres et de commentaires sur

qui donne le nombre de notes supérieures à 10
tous les commentaires. (La fonction `count(pathexpr)`
est équivalente à `count(pathexpr)`.)

ci-dessus.

Comme vous pouvez constater `books.xml` est
des livres.

Question 1: Écrire une expression XPath
(peu importe le livre) en prenant en compte
retourne le nombre d'éléments qui correspon-

Donner une requête XQuery qui affiche tous les commentaires avec la note
supérieure à 16. L'affichage doit prendre la forme d'une table HTML à trois colonnes :

```
<tr><td> titre_1 </td><td> note_1</td><td> commentaire_1 </td></tr>
.....
<tr><td> titre_n </td><td> note_n</td><td> commentaire_n </td></tr>
```

Exercice 4 - transactions (0.5 point)

Question 1: Nous avons trois transactions T_1, T_2, T_3 exécutée simultanément.
Est-ce que l'histoire suivante est serializable :

$$H_1 : r_1(X), r_2(Z), r_1(Z), r_3(X), r_3(Y), w_1(X), w_3(Y), r_2(Y), w_2(Z), w_2(Y)$$

Justifier la réponse.

Exercice 5 - indexes (1 point)

Question 1: On exécute la requête :

```
select *
from T
where A <> 5 and B = 4;
```

où A, B les attributs de la table T. Est-ce que l'index de type B-arbre sur l'attribut A sera
utile pour accélérer l'exécution de cette requête ?

Question 2: Est-ce que le même index peut accélérer l'exécution de la requête :

```
select *
from T
where A = 5 and B <> 4;
```

Justifier (très brièvement) vos réponses.