



Concours d'accès au Doctorat 3 ième Cycle Informatique 2017 – 2018

Le 29/10/2017

**Matière 2 :** Bases de données, Génie Logiciel,  
Coefficient 3, durée 2 Heures.  
(Spécialité : SIGL)

**Partie 1 :** Bases de données

**Exercice 1 :**

Soit le schéma BD suivant d'une société de fabrication de composants pour appareils électroniques :

**COMPOSANT** (CODE-COMPOSANT, NOM-COMPOSANT, MODELE, QUANTITE)  
**VENTE** (CODE-COMPOSANT, LOT, NOMCLIENT, NOMVENDEUR, MONTANT, DATE)  
**CLIENT** (NOMCLIENT, RUECLIENT, VILLECLIENT, PAYSCLIENT)  
**VENDEUR** (NOMVENDEUR, RUEVENDEUR, VILLEVENDEUR, PAYSVENDEUR)

La quantité en stock d'un composant ne peut être en dessous de 100 unités.

1. Donner la commande qui permet de satisfaire ce besoin

**Alter table COMPOSANT ADD CONSTRAINT ch\_qte CHECK(QAUNTITE>100).**

2. Donner la commande permettant de créer toutes les contraintes d'intégrité sur la table Vente.

**Alter table VENTE ADD constraint pk\_vente Primaray Key(CODE-COMPOSANT, LOT) ;  
Alter table VENTE ADD constraint fk\_vente\_COMPOSANT Foreign Key (CODE-COMPOSANT)  
REFERENCES COMPOSANT(CODE-COMPOSANT) ;  
Alter table VENTE ADD constraint fk\_vente\_CLIENT Foreign Key (NOMCLIENT)  
REFERENCES CLIENT(NOMCLIENT) ;  
Alter table VENTE ADD constraint fk\_vente\_VENDEUR Foreign Key (NOMVENDEUR)  
REFERENCES VENDEUR(NOMVENDEUR) ;**

L'administrateur veut créer deux tablespaces sur « c:\MesTBS » : un tablespace principal « TBS-P » d'une taille de 200 Mo auto extensible, avec le fichier de données « tbs-p.dat » et un tablespace temporaire « TBS-T » d'une taille de 100 Mo auto extensible, avec le fichier de données « tbs-t.dat ».

3. Donner les requêtes permettant de satisfaire ce besoin

**CREATE TABLESPACE TBS-P  
DATAFILE 'c:\MesTBS\tbs-p.dat'  
SIZE 200M  
AUTOEXTEND;**

**CREATE TEMPORARY TABLESPACE TBS-T  
TEMPFILE 'c:\MesTBS\tbs-t.dat'  
SIZE 100M  
AUTOEXTEND;**

L'administrateur veut créer un utilisateur USER-PROD et lui attribue le tablespace TBS-P avec un quota de 50 Mo et le tablespace TBS-T avec un quota illimité. Le mot de passe expire et devra être changé à la première connexion. Le compte de USER-PROD sera bloqué après 2 tentatives infructueuses d'introduction du mot de passe (FAILED\_LOGIN\_ATTEMPTS)

4. Donner les requêtes permettant de satisfaire ce besoin

```
CREATE USER USER-PROD Identified by 'rty6,566'  
DEFAULT TABLE SPACE TBS-P QUOTA 50M  
TEMPORARY TABLESPACE TBS-T QUOTA UNLIMITED  
PASSWORD EXPIRE
```

```
CREATE PROFILE P1  
LIMIT  
FAILED_LOGIN_ATTEMPTS 2
```

```
Alter USER USER-PROD PROFILE P1;
```

USER-PROD peut créer, supprimer et modifier le schéma des tables dans son schéma et dans n'importe quel schéma. Il peut également transmettre ces privilèges.

5. Donner les requêtes permettant de satisfaire ce besoin

```
GRANT  
CREATE ANY TABLE,  
ALTER ANY TABLE,  
DROP ANY TABLE  
TO USER-PROD  
WITH ADMIN OPTION
```

USER-PROD souhaite connaître les noms de toutes les tables dont il a des droits et qui appartiennent à l'utilisateur « USER-TIC ».

6. Donner les requêtes permettant de satisfaire ce besoin

```
SELECT TABLE_NAME  
FROM ALL_TABLES  
WHERE OWNER='USER-TIC'
```

L'administrateur souhaite sauvegarder dans la BD, le chiffre d'affaires réalisé par chaque vendeur sans avoir besoin de le calculer à chaque fois.

7. Proposer une solution à ce problème.

- Créer une colonne « Chiffre-Aff » dans la table Vendeur
- Créer un trigger qui met à jour cette colonne après chaque vente.

8. Donner les requêtes permettant d'implémenter cette solution

```
ALTER TABLE VENDEUR ADD CHIFFRE-AFF NUMBER (10,2)
```

```

Create or Replace Trigger Tr_Chiff-Aff
After insert on Vente
For each row
Begin
Update      Vendeur      set      Chiffre-Aff=Chiffre-Aff      +      :New.Montant      where
NomVendeur= :New.NomVendeur ;
End

```

Pour un bon suivi des ventes, l'administrateur veut sauvegarder (dans une table MAJ-VENTE), pour chaque modification de l'attribut « Montant » dans la table Vente les informations suivantes : Date de la modification, ancienne valeur, nouvelle valeur, utilisateur qui a effectué la modification.

9. Proposer un schéma logique pour la table MAJ-VENTE

```
MAJ-VENTE (Date-Modif, OldValue, NewValue, User) ;
```

10. Donner la commande permettant de remplir automatiquement cette table.

```

Create or Replace Trigger tr_maj_vente
After update(montant) on Vente
For each row
Declare
User varchar(50);
Begin
Select user_name into user from USER_USERS
Insert into MAJ-VENTE Values (Sysdate(), :Old.Montant, :New.Montant, user);
End.

```

## Exercice 2

Soient les transactions suivantes :

Début Transaction T1
Lire (A, a);
a:=(a*2)-5;
Écrire (a, A);
Lire (B, b);
b:=(b*2)-10;
Écrire (b, B);
Fin Transaction

Début Transaction T2
Lire (A, a);
Lire (B, b);
Afficher (a, b);
Fin transaction

Début Transaction T3
Lire (A, a)
Lire (B, b)
Lire (C, c)
a := (a+20)
Ecrire (a, A)
b := (b+50)
Ecrire (b, B)
c := (a + b)
Ecrire (c, C)
Fin transaction

Début Transaction T4
Lire (A, a);
Lire (B, b);
tmp:= a;
a:= b;
b:= tmp;
Écrire (A, a);
Écrire (B, b);
Fin transaction

1. Soit l'ordonnancement d'exécution suivant :

R1(A)R4(A)R4(B)R3(A)R3(B)R2(A)R2(B)R3(C)W1(A)R1(B)W1(B)W4(A)W4(B)W3(A)W3(B)W3(C)

Construire le graphe de précédence : est-ce que cet ordonnancement est sérialisable pourquoi ?

2. On remplace maintenant la transaction T2 par la suivante et on rajoute dans le plan d'exécution la transaction T5 :

Début Transaction T2
Lire (A, a);
a := (a+100)
Ecrire (a, A)
Lire (B, b)
b:= b*2
Ecrire (b, B)
Lire(C, c)
c := (a + b)
Ecrire (c, C)
Fin transaction

Début Transaction T5
Lire (A, a);
Lire (C, c);
c := a
Écrire (A, a);
Écrire (C, a);
Fin transaction

Soit l'ordonnancement suivant :

R1(A)R4(A)R4(B)R3(A)R3(B)R2(A)W2(A)R3(C)R5(A)W1(A)R1(B)W1(B)R5(C)R2(B)W2(B)W4(A)W4(B)W5(A)W5(C)W3(A)W3(B)R2(C)W3(C)W2(C)

- Donnez le scénario d'exécution en appliquant le protocole de verrouillage à 2 phases (V2P)
- Construire le graphe d'attente de cet ordonnancement
- Citer le ou les éventuels deadlock (verrouillage mortel) existants ? si oui comment peut-on les résoudre ?