

Examen Administration des SGBD
M1 : IL-SII

Etude de cas :

Soit la base de données suivante qui gère les produits et brevets de la recherche scientifique dans les laboratoires de recherche des établissements de l'enseignement supérieur et des centres de recherche.

LABORATOIRE-RECH (CODELAB, INTITULE-LAB, CODEORG, CODEDOM, DIRECTEUR, URL)

PRODUIT-RECH (CODEPROD, CODELAB*, INTITULE-PROD, DESCRIPTIF-PROD, CODETHEM*, CONTACT-PROD, TYPE-PROD)

BREVET-INVENTION (CODEBRV, CODEORG*, CODETHEM*, DATE-BRV)

ORGANISME (CODEORG, CODEDOM*, TYPEORG, DENOMINATION)

DOMAINE-RECH (CODEDOM, DESCRIPTIF-DOM)

THEMATIQUE-RECH (CODETHEM, CODEDOM*, DESCRIPTIF-THEM)

Où: CODEORG=CODE ORGANISME, CODETHEM= CODE THEMATIQUE, CODEDOM = CODE DOMAINE,

TYPEORG peut prendre les valeurs « Université » ou « Centre de recherche ».

TYPE-PROD peut prendre « communication » ou « publication ».

Les clés primaires sont soulignées et * représente une clé étrangère.

Partie I : Fonctions générales de SGBD et Optimisation

1. Donner les scripts de création des tables PRODUIT-RECH et ORGANISME.

Create Table Produit-Rech

(CodeProd varchar(10), CODELAB varchar(10), INTITULE-PROD Varcher (50), Descriptif-prod varchar (200), CodeTHEM varchar(10), Contacf-Prod Varchar (50), Type-Prod varchar (15),

Constraint pk-Prod Primary Key (CodeProd),

Constraint fk_prod_lab Foreign key (Codelab) references laboratoire-rech(codelab),

Constraint fk_prod_them Foreign key (CodeThem) references Thematique-Rech(codethem),

Constraint ch_prod Check (Type-Prod in ('communication', 'publication'))

)

Create Table Organisme

(CodeProd varchar(10), CODEDOM varchar(10), TypeOrg varchar (20), Denomination varchar (50),

Constraint pk-Prod Primary Key (CodeProd),

Constraint fk_prod_lab Foreign key (Codelab) references laboratoire-rech(codelab),

Constraint fk_prod_them Foreign key (CodeThem) references Thematique-Rech(codethem),

Constraint ch_org Check (TypeOrg in ('universit','centre de recherche'))

)

2. L'administrateur crée un utilisateur GEST-ORG qui gère les organismes dans la BD. Il lui attribue un quota de 70 % du tablespace « ORG-BTS » dont la taille globale est de 1 Go auto extensible. Il lui attribue 50 Mo d'espace sur le tablespace temporaire système TEMP.

- a. Donner le script permettant de créer les tablespaces

```
Create tablespace Gest-Org-TBS  
Datafile 'gest-org.dat' size 1000 M  
Autoextended
```

Le tablespace TEMP existe déjà dès l'installation d'Oracle.

- b. Donner le script permettant de créer l'utilisateur.

```
Create user Gest-org identified by xxxx  
Default table space Gest-org-TBS Quota 700 M  
Temporary Tablespace TEMP quota 50 M
```

3. GEST-ORG veut supprimer l'organisme dont le code est « Univ10 ».

- a. Donner la requête qu'il utilise.

```
Delete from Organisme where codeorg='Univ10'
```

- b. Donner la requête permettant de lui donner ce droit.

```
Grant delete on Organisme to Gest-Org
```

- c. Quelles seraient les conséquences de cette requête si

- i. Toutes les contraintes de clés étrangères ont été créées sans l'option « ON DELETE CASCADE »

Si au moins un laboratoire ou un brevet référence cet organisme comme clé étrangère (Codeorg='univ10') alors la suppression n'est pas acceptée par le SGBD pour violation de contrainte de clé étrangère.

- ii. Avec l'option « ON DELETE CASCADE »

- Tous les laboratoires dont (Codeorg='univ10') seront supprimés
- Tous les brevets dont (Codeorg='univ10') seront supprimés
- Tous les produits de recherche dont codelab correspond à un des laboratoires supprimés seront tous supprimés.

- iii. Que conclure concernant l'utilisation de cette option.

Cette option est très pratique dans certains cas où la suppression du père implique automatiquement la suppression des fils
Cette option est très dangereuse puisque une suppression accidentelle d'un tuple peut engendrer plusieurs suppressions successifs dans plusieurs tables.

4. Les laboratoires changent de sites Web. L'administrateur veut sauvegarder toutes les URL des sites web dans le temps. Lorsqu'un laboratoire change d'adresse, l'administrateur sauvegarde l'ancienne URL dans une autre table « Hist_URL_Lab ». Cette table contient quatre attributs : date_modification, Codelab, URL et Utilisateur qui a modifié l'URL. Où date_modification désigne la date système lors de la modification, CodeLab : le code du laboratoire et URL contient l'ancienne URL.

- a. Proposer un schéma logique pour la table Hist_URL_Lab.

```
Hist_URL_Lab(Date_modification, CodeLab, URL)
```

- b. Donner la script de création de la table et l'effet sur les différents catalogues.

```
Create Table Hist_URL_Lab (Date_modification Date Primary Key, CodeLab varchar(10), URL
varchar(200), User varchar(50))
Effet sur les catalogues
+ 1 ligne dans le catalogue TABLE
+ 4 lignes dans le catalogue COLONNE
+ 1 ligne dans le catalogue CONTRAINTE
+1 ligne dans le catalogue INDEX (après la création automatique de l'index sur la clé primaire)
```

c. Donner le script permettant de remplir automatiquement cette table.

```
Create or Replace Trigger insert_hist_URL_Lab
After update(URL) on Laboratoire-Rech
For each row
Declare
Utilisateur user_users.username%type;
Begin
Select username into utilisateur from user_users;
Insert into Hist_URL_Lab values (Sys_date(), :old.CodeLab, :old.URL, utilisateur);
End
```

5. L'administrateur veut sauvegarder dans la BD, pour chaque organisme, le nombre de réalisations (brevet, produit de recherche) sans à le recalculer à chaque fois.

a. Proposer une solution

Ajouter un attribut (nb_real) dans organisme contenant le nombre de réalisations
Créer deux triggers l'un sur produit-rech et l'autre sur brevet et qui incrémentent ce nombre à chaque fois d'un produit/brevet est inséré.

b. Donner les requêtes permettant d'implémenter cette solution.

```
Alter table Organisme Add Nb_Real Number ;
-----

Create or Replace Trigger MAJ-NB_REAL_prod
After insert or delete on Produit-Rech
For each row
Declare
Org Laboratoire-Rech.Codeorg%type;
Begin
If inserting
Begin
Select Codeorg into Org from Laboratoire-rech where CodeLab=:new.codelab;
Update organisme set NB_Real=Nb_Real+1 where codeorg=org;
End
If deleting
Begin
Select Codeorg into Org from Laboratoire-rech where CodeLab=:old.codelab;
Update organisme set NB_Real=Nb_Real-1 where codeorg=org;
End
End
-----

Create or Replace Trigger MAJ-NB_REAL_Brv
```

After insert or delete on Brevet-Invention

For each row

Begin

If inserting Update organisme set NB_Real=Nb_Real+1 where codeorg=:new.codeorg;

If deleting Update organisme set NB_Real=Nb_Real-1 where codeorg=:old.codeorg;

End

6. L'administrateur veut savoir, pour chaque organisme, le nombre de produits de recherche de type « publication ».

- a. Donner la requête SQL

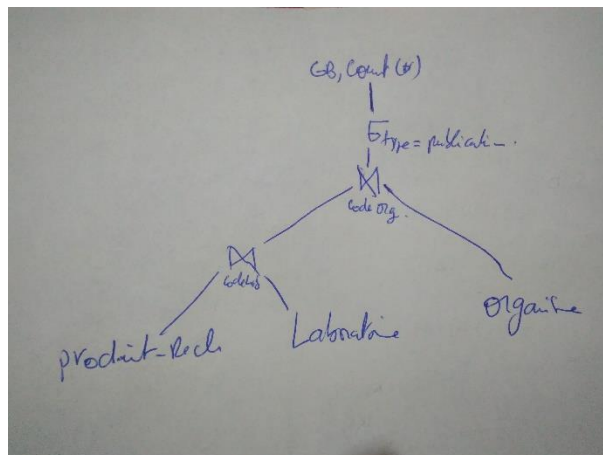
Select CodeOrg, Count(Codeprod)

From Organisme O, Produit-rech PR, laboratoire L

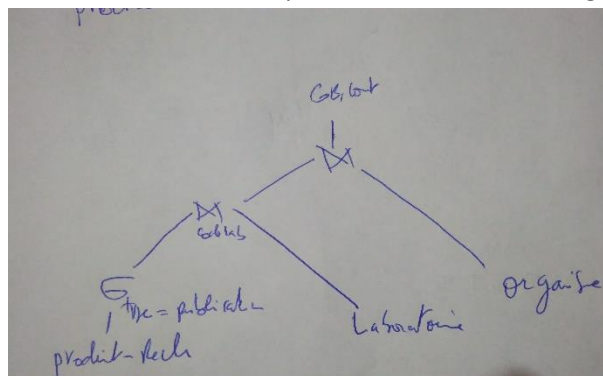
Where PR.Codelab=L.Codelab and O.Codeorg=L.CodeOrg and PR.Type="publication"

Group by CodeOrg

- b. Donner un arbre algébrique non optimisé



- c. Générer un arbre optimisé en utilisant les règles de transformation



- d. Estimer le coût d'exécution de la requête selon l'arbre optimisé sachant que les résultats intermédiaires ne peuvent être stockés dans la RAM.

| Opération | Coût | Taille du résultat |
|--|---|---|
| SIGMA $PR.Type="publication"$ (produit-rech) | Charger toute la table = 50 000/5 = 10 000 pages | $10\% \times 50\,000 = 5000$ tuples. |
| Sauvegarde du résultat en MS | $5000/5 = 1000$ pages | |
| Produit-Rech JOIN Lab = Resultat | $3 (1000 + 5000/5) = 6\,000$ pages (jointure par hachage) | Jointure par clé étrangère → nb-tuples = 5000 tuples de |

| | | |
|---------------------------------|---|--|
| | $1000+1000*(5000/5) = 101000$ pages (boucles imbriquées) | Produit-rech collées avec 5000 tuples de Lab |
| Sauvegarder Resultat en MS | $5000/5 + 5000/5 = 2000$ pages | |
| Resultat JOIN Organismes = Res2 | $3(2000 + 500/5) = 6300$ pages | Jointure par clé étrangère 5000 tuples de Resultat collées avec 5000 tuples de organismes |
| Sauvegarder Res2 | 3000 pages | |
| Group By et Count | Charger 3000 pages | |
| Total | 31300 pages (hachage) | |

e. Proposer un index permettant d'optimiser cette requête. Donner la commande correspondante.

Créer un index sur l'attribut Type-Prod
Create index on Produit-rech. Type-Prod

Données : une page contient 5 tuples, la BD contient 500 organismes, 5000 laboratoires et 50 000 Produits de recherche dont 10% sont des publications.

Partie II :Accès concurrents et Reprise sur panne

L'exécution suivante est reçue par un SGBD utilisé pour des applications de commerce électronique.

O1: R1(X) R2(Y) W1(X) R2(Z) R3(Z) R3(X) W2(Z) R1(Z) W1(X) W3(X)

1. L'une des opérations dans l'application de commerce électronique est la modification du prix d'un produit suite à une remise de prix d'un pourcentage donné (le prix du produit et le taux de remise sont stockés dans la base de données).

– Y a-t-il une ou plusieurs transactions de **O1** qui puissent provenir de l'exécution de cette opération ? Justifiez votre réponse.

➤ **Réponse :** les transactions de l'ordonnancement O1 sont :

- T1 : R1(X) W1(X) R1(Z) W1(X).
- T2 : R2(Y) R2(Z) W2(Z)
- T3 : R3(Z) R3(X) W3(X)

La réduction du prix nécessite la lecture du taux de réduction et ensuite la mise-à-jour (lecture+écriture) du prix. Les transactions T2 et T3 correspondent à cette suite d'actions. Mais les deux ne peuvent pas être en même temps des réductions de prix, car Z ne peut pas être à la fois prix (pour T2) et taux de réduction (pour T3). Donc la réponse correcte est oui, soit T2, soit T3, mais pas les deux ensembles

2. Vérifiez si **O1** est sérialisable [en identifiant les conflits et] en construisant le graphe de précedence.

➤ **Réponse :**

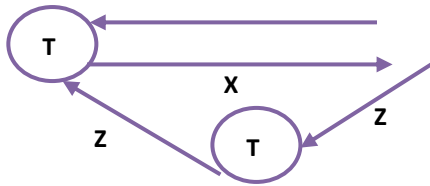
Les conflits :

- Sur X : R1(X)-W3(X), W1(X)-R3(X), W1(X)-W3(X), R3(X)-W1(X)
- Sur Y : pas de conflit
- Sur Z : R3(Z)-W2(Z), W2(Z)-R1(Z)

le graphe de précedence :

X

T



Le graphe contient plusieurs de cycles (par exemple {T1, T2, T1}), l'ordonnancement **O1** est n'est pas sérialisable.

3. Quelle est l'exécution obtenue en appliquant le protocole de verrouillage à deux phases à partir de **O1**?

➤ Réponse :

| Transaction | Action | Demande de verrous | Réponse |
|-------------|--------|--------------------|---|
| T1 | R1(X) | SLOCK(X) | OK |
| T2 | R2(Y) | SLOCK(Y) | OK |
| T1 | W1(X) | XLOCK(X) | OK |
| T2 | R2(Z) | SLOCK(Z) | OK |
| T3 | R3(Z) | SLOCK(Z) | OK |
| T3 | R3(X) | SLOCK(X) | Non, T3 attend la fin de T1 |
| T2 | W2(Z) | XLOCK(Z) | Non, T2 attend la fin de T3 |
| T1 | R1(Z) | SLOCK(Z) | OK |
| T1 | W1(X) | XLOCK(X) | OK, fin de T1, libérer les verrous (UNLOCK(X), UNLOCK(Z)) et réveiller T3 |
| T3 | R3(X) | SLOCK(X) | OK |
| T3 | W3(X) | XLOCK(X) | OK, fin de T3, libérer les verrous (UNLOCK(X), UNLOCK(Z)) et réveiller T2 |
| T2 | W2(Z) | XLOCK(Z) | OK, fin de T2, libérer les verrous (UNLOCK(Y), UNLOCK(Z)) |

4. Notons **O1'** le scénario d'exécution des actions obtenu à la question précédente.

- Peut-on dire, sans construire le graphe de précedence, que **O1'** est sérialisable? Pourquoi ?
- **O1'** peut-elle être obtenue en appliquant le protocole par estampillage à deux estampilles ? Justifiez votre réponse.

➤ Réponse :

- **O1': R1(X) R2(Y) W1(X) R2(Z) R3(Z) R1(Z) W1(X) R3(X) W3(X) W2(Z)**
- Oui, le verrouillage à deux phases garantit la sérialisabilité du résultat.
- O1'ne peut pas être obtenu en appliquant le protocole par estampillage à deux estampilles, voir le tableau

| Transaction | Action | EL(X) | EE(X) | EL(Y) | EE(Y) | EL(Z) | EE(Z) |
|-------------|--------|-------|-------|-------|-------|-------|-------|
| | | 0 | 0 | 0 | 0 | 0 | 0 |
| T1 | R1(X) | 1 | | | | | |
| T2 | R2(Y) | | | 2 | | | |
| T1 | W1(X) | | 1 | | | | |
| T2 | R2(Z) | | | | | 2 | |
| T3 | R3(Z) | | | | | 3 | |
| T1 | R1(Z) | | | | | 3 | |
| T1 | W1(X) | | 1 | | | | |
| T3 | R3(X) | 3 | | | | | |
| T3 | W3(X) | | 3 | | | | |



| | | |
|-----------|-------|--|
| T2 | W2(Z) | On a $EL(Z)=3>E(T2)$ d'où l'action W2(Z) est rejetée et T2 est annulée et sera relancée avec une nouvelle estampille 3 |
|-----------|-------|--|

5. Soit maintenant l'ordonnancement O2 :

O2 : R1(X) R2(Y) R2(Z) R3(Z) R3(X) W1(X) W2(Z) R1(Z) W1(X) W3(X)

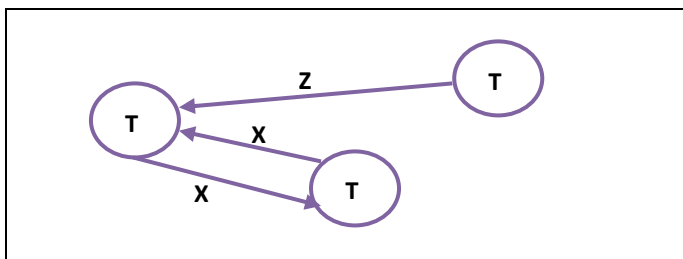
a- Construire le graphe d'attente de cet ordonnancement.

➤ **Réponse :**

V2P :

| Transaction | Action | Demande de verrous | Réponse |
|-------------|--------|--------------------|-----------------------------|
| T1 | R1(X) | SLOCK(X) | OK |
| T2 | R2(Y) | SLOCK(Y) | OK |
| T2 | R2(Z) | SLOCK(Z) | OK |
| T3 | R3(Z) | SLOCK(Z) | OK |
| T3 | R3(X) | SLOCK(X) | OK |
| T1 | W1(X) | XLOCK(X) | Non, T1 attend la fin de T3 |
| T2 | W2(Z) | XLOCK(Z) | Non, T2 attend la fin de T3 |
| T1 | R1(Z) | SLOCK(Z) | T1 est en attente |
| T1 | W1(X) | XLOCK(X) | T1 est en attente |
| T3 | W3(X) | XLOCK(X) | Non, T3 attend la fin de T1 |

Graphe :



b- Existe-t-il un deadlock ?

➤ **Réponse :** De graphe d'attente, on constate qu'on a un cycle T1-T3-T1 d'où on a un inter-blocage.

c- Si oui, proposer une solution à ce problème.

- **Réponse :** Pour lever l'inter blocage on choisit une transaction victime parmi T1 et T3 on l'annule et on la relance à la fin. Pour T1 victime on aura la séquence : T3 T2 T1 ; Ou bien T3 victime : T2 T1 T3 ;

6. Soit l'ordonnancement **O3 : R1(X) R2(Y) W1(X) R2(Z) R3(Z) R1(Z) W1(X) R3(X) W3(X) W2(Z).**

Si une panne intervient après l'exécution de W3(X) dans le cadre de l'ordonnancement **O3** et qu'un **CHEKPOINT** a eu lieu après l'exécution de W1(X) comme suit :

R1(X) R2(Y) W1(X) R2(Z) R3(Z) R1(Z) W1(X) **CHEKPOINT** R3(X) W3(X) **PANNE** W2(Z)

- a. Donner l'état des transactions juste après le Checkpoint ensuite juste avant la panne.
- **Juste après le Checkpoint :**
 - T1 est validée
 - T2 et T3 sont actives
- **Juste avant la panne:**
 - T1 est terminée
 - T3 partiellement validée
 - T2 est active
- b. Donner les actions à entreprendre après la reprise. **0.5 pts**
- T1 est terminée, T3 sera refaite (W3(X)) et T2 sera défaite (aucun effet car elle n'a pas fait de mise à jour).

Bonne Chance