

Examen Bases de données avancées et Big Data
M1 MIL-MIV-SII-RSD

Exercice 01 (12 pts)

Soient trois ordonnancements suivants :

O1 : W1(A) R2(A) R2(B) W1(B) CommitT1 CommitT2 R3(A)W3(A) Commit T3

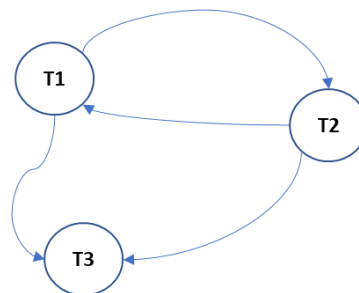
O2 : W1(A) R2(A) R2(B) W1(A) CommitT2 W3(A) R3(A) CommitT1 CommitT3.

O3: W1(A) W2(A) AbortT1 CommitT2 R3(A) R3(B) CommitT3

1. Dessiner le graphe de précédence de O1. O1 est-il sérialisable ?

Sur A : T1 précède T2, T3, T2 précède T3,

Sur B : T2 précède T1



Il existe un Cycle donc non sérialisable

2. En supposant que le réveil des transactions en attente se fait dès la libération des verrous demandés, appliquer le protocole V2P à ces trois ordonnancements.

O1

Action	Demande	Réponse/Attente	Réveil
W1(A)	X1(A)	OK	
R2(A)	S2(A)	NON, T2 attend T1	OK
R2(B)	En attente		OK
W1(B) CommitT1	X1(B)	OK, U(A, B)	
CommitT2			OK, U(A,B)
R3(A)			S3(A), OK
W3(A) Commit T3			X3(A), OK, U(A)

Action	Demande	Réponse/Attente	Réveil
W1(A)	X1(A)	OK	
R2(A)	S2(A)	NON, T2 attend T1	OK
R2(B)	-	Attente	OK
W1(A)	X1(A)	OK	
CommitT2	-	Attente	Ok, U(A,B)
W3(A)	X3(A)	NON T3 attend T1	OK
R3(A)	En attente		S3(A), OK, U(A)
CommitT1.	OK	U(A)	
CommitT3			

O3 : W1(A) W2(A) AbortT1 CommitT2 R3(A) R3(B) CommitT3

Action	Demande	Réponse/Attente	Réveil
W1(A)	X1(A)	OK	
W2(A)	X2(A)	NON, T2 attend T1	OK
AbortT1		OK, U(A)	
CommitT2			OK, U(A)
R3(A)			S3(A), OK
R3(B)			S3(B), OK
CommitT3			OK U(A,B)

3. Vérifier pour chaque ordonnancement s'il y a un deadlock. Donner une solution et montrer l'ordre d'exécution final.

Il n'y a aucun deadlock

Ordre final :

O1 : T1, T2, T3

O2 : T1, T2, T3

O3 : T2, T3 (T1 annulée).

Soient trois granules A, B et C contenant la valeur 0 (zéro) dans la mémoire secondaire.

Soient trois transactions, T1, T2 et T3 qui modifient les granules comme suit :

T1 : A :=42

T2 : B :=20, A :=10

T3 : A :=100, C :=101

4. Compléter la trace d'exécution suivante issue du fichier journal :

1. <T2 Start>
2. <T3 Start>
3. <T2,A, 0, 10>
4. <T2, B, 0, 20>
5. <Commit T2>
6. <Checkpoint 1>
7. <T3, A, 10, 100>
- Panne
8. <T1 Start>
9. <T3, C, 0, 101>
10. <T1, A, 100, 42>
11. <Commit T3>
12. <Checkpoint 2>
13. <Commit T1>

Panne

5. Après une panne après la ligne 7, et une reprise, quelles seraient les états de A et B ?

Au moment de la panne T2, Validée et T3 En cours.

Il faut défaire T3

Remettre A à 10 (rien à faire car la modification de A a été faite après Checkpoint2.

B reste à 20

6. Après une panne après la ligne 13 et une reprise, quelles seraient les états de A, B et C?

Après la ligne 13 on a :

T2 et T3 validées

T1 partiellement validée

Checkpoint 2, toutes les mises à jour ont été effectués

A=42, C=101, B=20

7. Répondre à la question 5 Si la ligne 11 était « Abort T3 » au lieu de « Commit T3 »

Aucune modification par rapport à la question 5.

Si la panne arrive après la ligne 13

A=10, B=20, C=0.

Exercice 2 :

La banque C.N.E.P gère un ensemble de terrains à construire pour le développement de logements. Deux relations ont été dégagées pour le sous-système de gestion de terrains à bâtir.

TERRAIN (Nom-Terrain, Adresse, Superficie, Numpermis_construire, date_deb-Trav, date_fin-Trav, structure_rattachement, NB-Visites)

VISITE-TERRAIN (Id-visite, Nom-visiteur, Date-visite, Rapport)

1. L'administrateur veut créer un utilisateur TECH qui peut modifier le schéma de la table TERRAIN et peut créer des tables et transférer ces deux privilèges à d'autres utilisateurs. Donner les requêtes SQL permettant de :

1.a. Créer l'utilisateur

```
Create User TECH identified by xxxxx
```

1.b. Donner les privilèges cités plus haut.

```
Grant Alter On TERRAIN To TECH with grant option  
Grant Create Table TO TECH with Admin option
```

2. La table TERRAIN a été créée sans clé primaire. Deux terrains peuvent avoir le même nom, alors l'administrateur veut qu'un nouvel attribut Num-Terrain soit la clé primaire dans TERRAIN. Un terrain peut avoir plusieurs visites. Une visite concerne un seul terrain. La suppression d'un terrain de la BD entraîne automatiquement la suppression de toutes ses visites.

2.a. Donner les commandes SQL permettant de modifier la BD afin de répondre à ces besoins.

```
Alter Table TERRAIN ADD NumTerrain Varchar (20) Primary Key;  
Alter Table Visite-Terrain ADD NumTerrain varchar(20) Foreign Key references TERRAIN.NumTerrain On delete  
Cascade;
```

2.b. Supposons maintenant que la table TERRAIN contient déjà des données et que la table VISITE-TERRAIN est vide. Quel est le problème rencontré en exécutant les commandes précédentes. Donner la solution.

Problème rencontré : la clé NumTerrain est vide, tous les enregistrements auront la valeur NULL.
Solution : Ajouter la colonne NumTerrain, remplir la colonne pour les tuples existants par des valeurs différentes ensuite créer la contrainte de clé primaire.
Alter Table TERRAIN ADD NumTerrain Varchar (20); // ajout de l'attribut
Update TERRAIN set Num-Terrain=Rownum; // remplissage de la colonne terrain
Alter Table TERRAIN ADD CONSTRAINT pk_Terrain Primary key (Num-Terrain); // définition de la clé primaire.

3. Les travaux sur les terrains peuvent avoir des retards. Dans ces cas, des responsables du suivi modifient la date de fin de travaux. La nouvelle date doit impérativement être plus récente que celle inscrite sur la BD. L'administrateur

veut garder dans une table séparée, la trace des différents reports : utilisateur qui a effectué la modification, la date de la modification, la nouvelle date.

3.a. Proposer une solution et son implémentation

- Créer une table Trace-modif ayant le schéma suivant

Trace-Modif (NumModif, User, Date-Modif, Nvle-Date) ;

Créer un trigger qui vérifie que la nouvelle date est plus récente

```
Create or Replace Trigger Tr-Verif-Date
Before update on Terrain(Date-Fin-trav)
For each row
Begin
If :new.Date-Fin-trav<:old. Date-Fin-trav then Raise_Application_Error (-21001, 'ERREUR, la nouvelle date doit être
supérieur à' || :old.Date-Fin-trav) ;
End.
```

- Créer un trigger qui insère une ligne dans Trace-modif après chaque modification de la date.

```
Create or replace Trigger Tr-Trace
After update on Terrain(Date-Fin-trav)
For each row
Util User_Users.user_name%type;
Begin
Select user_name into util from User_Users;
Insert into Trace-Modif VALUES (util,sysdate(),:new. Date-Fin-trav) ;
End.
```

Bon courage