

TP5 (Transactions réparties) Pr Abdelmounaïm ABDALI

1ère partie sur serveur1 : les transactions dans un compte

1. Créer une table ne contenant pas de clé:

Sir2020Etudiant(CNE , Nom, Prénom)

2. Insérer trois lignes dans la table, puis lister le contenu de la table.

3. Utiliser l'instruction ROLLBACK, puis lister à nouveau le contenu de la table.
Que s'est-il passé?

4. Insérer trois lignes dans la table, utiliser l'instruction COMMIT;
Puis l'instruction ROLLBACK, Lister le contenu de la table. Quel est le rôle de l'instruction COMMIT?

5. Insérer à nouveau deux ou trois lignes dans la table et fermer brutalement votre session (par exemple en fermant la fenêtre SQL Plus ou en quittant le navigateur), puis rentrer à nouveau dans votre compte. Les données saisies ont-elles été préservées ?

6. Insérer deux ou trois lignes dans la table, ajouter une nouvelle colonne à la table puis insérer à nouveau deux ou trois lignes et enfin utiliser l' instruction ROLLBACK. Lister le contenu de la table. Que peut-on en conclure ?

7. Pour chacun des changements suivants, refaire les essais du point précédent:

○ Créer une nouvelle table nommé **Sir2020Cours** au lieu d'ajouter simplement une colonne. Cette table contienne les colonnes : IdCours et Libellé

- Lancer ensuite un Rollback
- Lancer la commande DESC cours
- Qu'est ce que vous pouvez conclure
- Supprimer la nouvelle table
- Lancer ensuite un Rollback

Qu'est ce que vous pouvez conclure sur les instructions CREATE TABLE et DROP TABLE ?

2ème partie sur serveur1: transactions concurrentes

1. Vider le contenu de la table **Sir2020Etudiant** et valider l'opération.
2. Se connecter au même compte à partir d'une autre fenêtre et constater aussi que le compte est bien vide.
3. Insérer dans chacune des fenêtres deux ou trois lignes différentes. Que voit-on des lignes insérées à partir de chacun des deux écrans ? Conclusion ?
4. Valider les modifications dans l'une des deux fenêtres puis lister le contenu de la table dans chacune des deux fenêtres. Conclure sur la portée de la validation.
5. Valider les modifications dans chacune des fenêtres. Insérer à nouveau quelques lignes dans chacune des fenêtres puis annuler les modifications dans une des deux fenêtres. Lister le contenu de la table dans les deux comptes et conclure sur la portée de l'annulation.
6. Valider les mise-à-jour dans chaque fenêtre. Ajouter une clé à la table. Ajouter dans chaque fenêtre une ligne ayant la même valeur pour la clé. Que se passe-t-il ? Annuler l'insertion dans la fenêtre encore active. Que devient le blocage?
7. Refaire les opérations du point précédent, mais en validant l'insertion au lieu de l'annuler. Expliquer ce qui c'est passé dans les deux cas (annulation et validation).

3ème partie - Mise à jour en réparti : le protocole de validation à deux phases

Dans cette partie, on va travailler sur les tables *ClientsMarrakech* et *ClientsCasablanca* créées lors du premier TP de **Synonyme-Réplication-Vues--Procédure Stockées**

- 1- Sur la base de **Serveur1**, insérer deux nouveaux clients: un dans la table *ClientsMarrakech* et un dans la table *ClientsCasablanca* distante. Vérifier dans chaque table la présence du nouveau client.
- 2- Sur la base de **Serveur2**, insérer deux autres clients : un dans la table *ClientsCasablanca* et un dans la table *ClientsMarrakech* distante. Vérifier dans chaque table la présence du nouveau client.
- 3- Sur la base de **Serveur1**, faites un **COMMIT**.
- 4- Sur la base de **Serveur2**, faites un **ROLLBACK**. Vérifier dans les deux tables des deux sites et commenter