

Travaux Pratiques Transactions

Objectifs :

Dans ce TP, vous allez manipuler des transactions en utilisant **COMMIT**, **ROLLBACK**, et **SAVEPOINT** pour assurer la cohérence des données dans des scénarios de mise à jour multi-étapes.

Partie 1 :

1. Insérez trois ou quatre lignes dans la table **PRODUIT**, puis visualisez les données insérées.
2. Modifiez une ligne, supprimez-en une autre, puis annulez les mises à jour en utilisant **ROLLBACK**.
3. Observez-vous les modifications ? Expliquez pourquoi.
4. Insérez à nouveau trois ou quatre lignes, modifiez-les et supprimez partiellement certaines lignes, puis validez les mises à jour avec **COMMIT**.
5. Essayez maintenant de faire un **ROLLBACK**. Que s'est-il passé ?
6. Supprimez toutes les produits 9 de la table et validez.
7. Insérez deux ou trois lignes dans la table, puis ajoutez une nouvelle colonne **DIMENSIONS** à la table et tentez d'annuler les dernières insertions.
8. Les insertions ont-elles été annulées ? La colonne ajoutée est-elle toujours présente ? Pourquoi ?
9. Videz la table en utilisant un **DELETE**, puis validez avec **COMMIT**.
10. Insérez trois lignes dans la table **CLIENT**, puis créez un **SAVEPOINT** après chaque insertion. Affichez les données.
11. Supprimez la dernière insertion en utilisant **ROLLBACK TO SAVEPOINT**.
12. La ligne insérée pour a-t-elle été supprimée ? Expliquez.
13. Insérez une nouvelle ligne, puis effectuez plusieurs modifications avec des **SAVEPOINTS** et annulez certaines d'entre elles.
 - Insérez un nouveau client.
 - Modifiez deux clients en plaçant un **SAVEPOINT** après chaque modification.
 - Revenez à un **SAVEPOINT** pour annuler une modification spécifique.

14. Effectuez une transaction avec plusieurs SAVEPOINTS et annulez seulement une partie de la transaction.
 - Insérez deux produits dans la table PRODUIT.
 - Modifiez le prix de chaque produit en plaçant un **SAVEPOINT** après chaque modification.
 - Annulez seulement le dernier changement de prix.
15. Validez la transaction avec COMMIT et effectuez un ROLLBACK pour observer l'effet après validation.
 - Validez les changements avec **COMMIT**.
 - Effectuez un **ROLLBACK** après **COMMIT** et vérifiez si les changements sont annulés.
16. Supprimez toutes les données de la table PRODUIT, placez un SAVEPOINT, puis essayez de revenir en arrière.
 - Supprimez les lignes de la table.
 - Placez un **SAVEPOINT**.
 - Tentez d'ajouter une nouvelle ligne, puis annulez cette dernière insertion seulement.
17. La table PRODUIT est-elle vide ? Qu'avez-vous appris sur l'utilisation de **SAVEPOINT** ?

Partie 2 :

1. Transférons une quantité de stock d'un produit à un autre produit. On vérifie manuellement les résultats pour voir si la transaction a été annulée ou validée.
 - Débiter 50 unités du produit 1.
 - Créditez 50 unités au produit 2.
 - Validez la transaction avec **COMMIT** ou annulez-la avec **ROLLBACK**.
2. Transférons une quantité de stock d'un produit à un autre produit de la même couleur, puis validons la transaction.
 - Débiter 50 unités du produit numéro 1.
 - Créditez 50 unités au produit numéro 2.
 - Validez la transaction avec **COMMIT**.

3. Transférons une quantité de stock d'un produit à un autre, mais annulons la transaction si le stock du produit d'origine devient insuffisant.
 - Débitier 300 unités du produit numéro 3.
 - Créditez 300 unités au produit numéro 4.
 - Si le stock du produit 3 devient négatif, annulez la transaction.

4. Création de Commande avec Vérification de Stock

Instructions :

- Lorsqu'un client passe une commande, vérifier que le stock est suffisant.
 - Débitier la quantité commandée du stock du produit dans la table PRODUIT.
 - En cas de stock insuffisant, annuler la transaction.
5. Imaginons une commande avec plusieurs produits. Utilisez SAVEPOINT pour enregistrer les étapes intermédiaires. Si le stock de l'un des produits est insuffisant, annulez seulement la mise à jour de ce produit sans affecter les autres.
 - Insérez une nouvelle commande dans la table COMMANDE.
 - Débitez les quantités de plusieurs produits dans la table PRODUIT.
 - Créez un **SAVEPOINT** après chaque produit mis à jour.
 - Si une quantité est insuffisante pour un produit, revenez au **SAVEPOINT** correspondant.
 6. Imaginons que l'on souhaite ajuster les quantités commandées pour des produits dans une commande existante. Utilisez des SAVEPOINTS pour effectuer des ajustements tout en ayant la possibilité de revenir à l'état initial si un problème est détecté.
 - Débitez des quantités supplémentaires de produits pour ajuster une commande.
 - Créez un **SAVEPOINT** avant chaque mise à jour de produit.
 - Si une mise à jour entraîne un stock négatif, effectuez un **ROLLBACK TO** pour revenir au dernier état valide.