

TP2 Gestion des transactions

Considérons le schéma de base de données suivant :

Compte-cheque(num-cc, num-client, solde)

Compte-epargne(num-ce, num-client, solde)

Decision-crédit(num-dossier, num-client, décision)

1. Implémenter cette base de données sous le SGBD Oracle et remplissez les tables **Compte-cheque** et **Compte-epargne** avec des données exemple.
2. Ecrire les trois procédures PL/SQL suivantes :
 - a. **transfert(num-client, from-num-cpt, to-num-cpt, m)** qui permet de transférer un montant **m** du compte **from-num-cpt** vers le compte **to-num-cpt**.
 - b. **Virement-cc(num-client, num-cc, m)** qui permet de réaliser un virement d'un montant **m** vers le compte chèques **num-cc**.
 - c. **Virement-ce(num-client, num-ce, m)** qui permet de réaliser un virement d'un montant **m** vers le compte épargne **num-ce**.
 - d. **Traitement-crédit(num-client, m)** qui permet de vérifier une demande de crédit du client **num-client**. Si le montant de crédit demandé **m** est 3 fois supérieur à la somme des soldes de ces comptes chèques et épargne, le crédit est accepté (insertion d'une ligne dans la table Decision-crédit avec une décision **OK**). Sinon, la demande de crédit est rejetée (insertion d'une ligne dans la table Decision-crédit avec une décision **KO**).
3. Simuler des problèmes de concurrence
 - a. Ajouter à vos procédures un délai artificiel (utiliser la fonction **dbms_lock.sleep(nb_seconds IN NUMBER)**). Ce délai artificiel permet de simuler un délai qui serait lié à des communications réseau, une attente de confirmation d'un utilisateur, etc.
 - b. Exécuter vos procédures à partir de sessions différentes pour simuler au moins trois problèmes de concurrence.
4. Modifier vos procédures en utilisant les verrous pour éviter les problèmes précédents.
5. Comment peut-on éviter les problèmes de concurrence que vous avez rencontrés sans utiliser les verrous ?