

Sécurisation des accès aux données : les transactions

Nous verrons...

- Transactions
- Gestions des privilèges (droits)
- Vues

Transaction

Une transaction est une séquence atomique d'opérations (générées par des ordres SQL) qui fait passer la base d'un **état cohérent** à un autre état cohérent.

Par atomique, on entend que toutes les opérations de mise à jour incluses dans la transaction doivent être terminées, ou aucune (c'est tout ou rien).

Transaction : exemple classique

Un transfert de n euros d'un compte A à un compte B

- Retirer n de A
- Mettre n sur B

Que se passerait-il si un problème survenait au niveau du SGBD entre les deux instructions et que la première uniquement était exécutée ?

→ n euros sont perdus

Solution : imposer l'exécution de l'ensemble des instructions ou aucune via une transaction ! (Propriété d'atomicité)

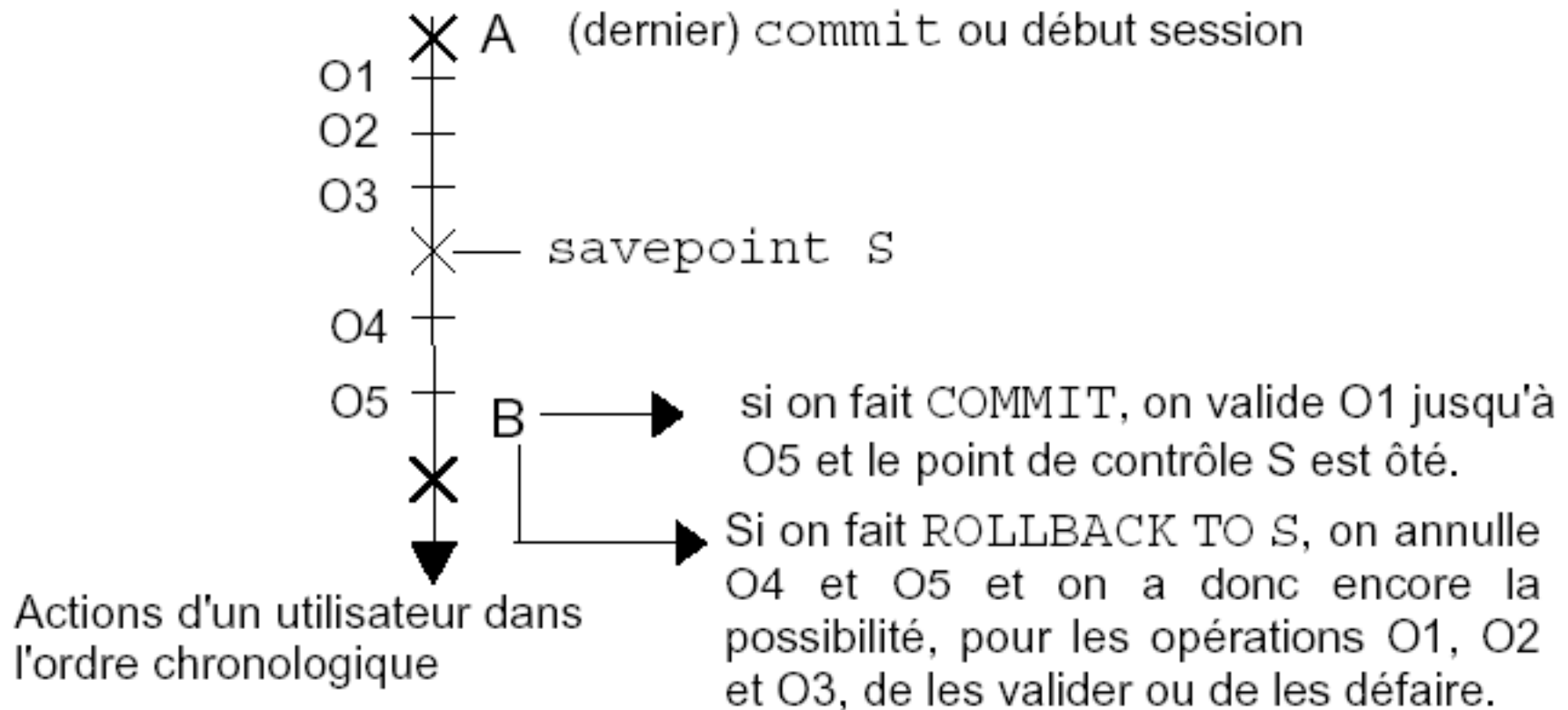
Propriétés ACID d'une transaction

- Atomicité : une transaction doit s'exécuter en tout ou rien
- Cohérence : une transaction doit respecter l'ensemble des contraintes d'intégrité
- Isolation : une transaction ne voit pas les effets des transactions concurrentes
- Durabilité : les effets d'une transaction validée ne sont jamais perdus, quel que soit le type de panne

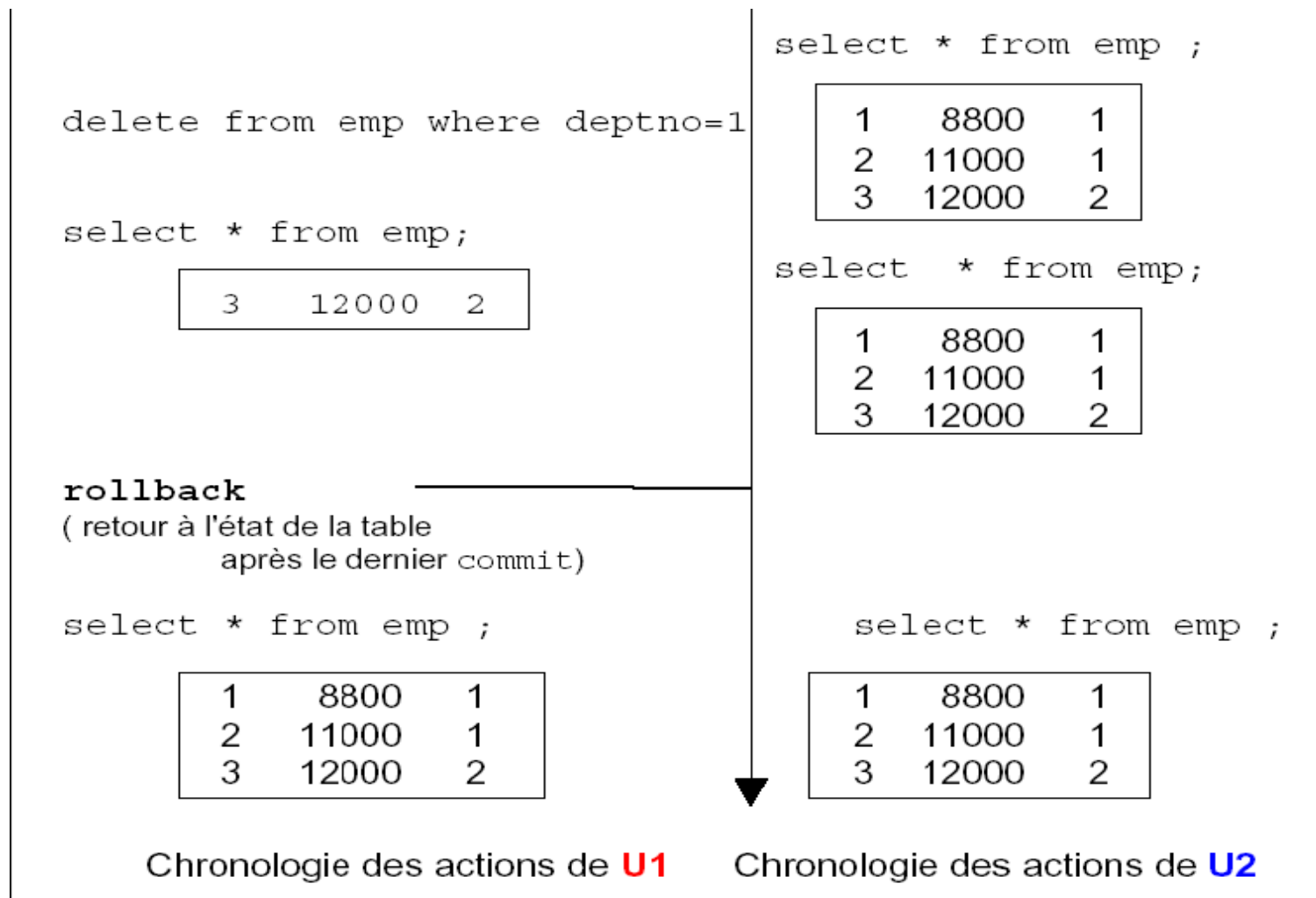
En pratique

- Les transactions sont délimitées par des primitives COMMIT et ROLLBACK ou la fermeture/ouverture de session
- Points de sauvegarde possibles avec SAVEPOINT
- Les commandes CREATE, DROP, ALTER, GRANT... provoquent un COMMIT automatique

Déroulement d'une transaction



Déroulement d'une transaction



Propriété d'isolation : une transaction ne voit pas les effets des transactions concurrentes

Contributeurs

- Agnès Braud
- Julien Schnell