

# Bases de Données Avancées - Transactions

Pierre Parrend  
IUT Lumière Lyon II, 2005-2006  
pierre.parrend@univ-lyon2.fr

- **I. Introduction**
- **II. Transactions et SQL**
- **III. Contrôle de la Concurrence**
- **IV. Récupération de Bases de Données**

- A. Problématique
  - i. Bases de Données Multi-Utilisateurs
    - Nombre important de Manipulations
    - Banques, Agences de voyages
  - Types d'accès
    - Lecture Seule
    - *Ensemble de Lecture* : éléments lus
    - Ecriture
    - *Ensemble d'écriture* : éléments modifiés
  - Granularité d'accès
    - Taille de l'élément auquel on accède
    - 1 champ ... 1 occurrence ... jointure de tables

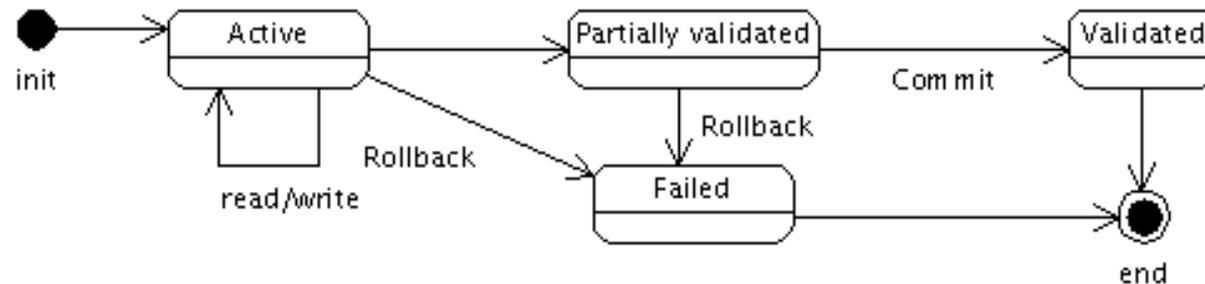
- A. Problématique
  - ii. Accès Simultannés
    - Alice et Eve travaillent chacune dans des agences de voyage
    - Un client leur demande de réserver des billets d'avion Lyon-Paris
    - Si il ne reste qu'une seule place
      - Alice cherche le nombre de places disponibles (1)
      - Eve cherche le nombre de places disponibles (1)
      - Alice réserve une place
      - Que se passe-t-il pour le client de Eve ?
    - Que ce passe-t-il si le trajet comporte
      - une partie en train (10 places restantes)
      - Une partie en avion (1 place restante) ?

- A. Problématique
  - Accès Simultannés
    - Incohérences entre les données lues et le comportement du système
    - Mises à jour perdues
    - Modifications partielles
      - 'Données sales'
      - Lecture de données incohérentes

- A. Problématique
  - iii. Reprise sur erreur
    - Les opérations d'une transaction doivent
      - Soit être toutes validées
      - Soit être toutes annulées
    - Types de pannes
      - Crash système
      - Erreur de transaction ou erreur système
      - Exception détectée par la transaction
      - Arrêt pour cause de transactions concurrentes
      - Erreur disque
      - Problèmes physiques

- B. Concepts
  - i. Transaction = 'Unité atomique d'accès à une base de données'
  - Etapes
    - Début
    - Ensemble d'opérations de lecture/écriture
    - Fin
    - Commit : validation de la transaction
    - Rollback : abandon de la transaction

- B. Concepts
  - ii. Diagramme UML d'Etats-Transitions



Created with Poseidon for UML Community Edition. Not for Commercial Use.

- B. Concepts
  - iii. Journalisation
    - Informations concernant les modifications de la Base de Données
    - Nécessaire aux reprises
    - Stockées dans un fichier distinct
    - Sauvegarde régulière sur bandes magnetiques
  - Point de validation
    - Ensemble d'opérations atomiques clos
    - Journalisation effectuée

- C. Propriétés ACID des Transactions
  - Atomicité
    - Les opérations d'une transaction sont utilisées soit entièrement, soit totalement annulées
  - Cohérence
    - Passage de la base de données d'un état cohérent à un autre
  - Isolation
    - Etat intermédiaire d'une transaction invisibles pour les autres transactions
  - Durabilité
    - Persistence des changements appliqués à une Base de Données

- I. Introduction
- II. **Transactions et SQL**
- III. Contrôle de la Concurrence
- IV. Récupération de Bases de Données

- A. Isolation entre Transactions
  - Par défaut, une requête est une transaction
  - Niveaux d'isolation
    - Serializable
      - Transaction représentée dans un fichier en mémoire
    - Repeatable Read
      - Pas d'erreur de lecture
    - Read Committed
      -
    - Read Uncommitted
      - Pas de protection

- B. Erreurs possibles
  - Lecture sale
    - Lecture d'une mise à jour non validée, puis abandonnée
  - Lecture non reproductible
    - Lecture d'une valeur par T1, puis modification par T2
  - Fantômes
    - Lecture d'un ensemble d'occurrences par T2, puis ajout d'occurrences par T2

- C. Erreurs possibles selon l'isolation

	Lecture Sale	Lecture non reproductible	Fantôme
Read Uncommitted	<b>X</b>	<b>X</b>	<b>X</b>
Read Committed	-	<b>X</b>	<b>X</b>
Repeatable Read	-	-	<b>X</b>
Serializable	-	-	-

- I. Introduction
- II. Transactions et SQL
- III. **Contrôle de la Concurrency**
- IV. Récupération de Bases de Données

# III. Contrôle de la Concurrency

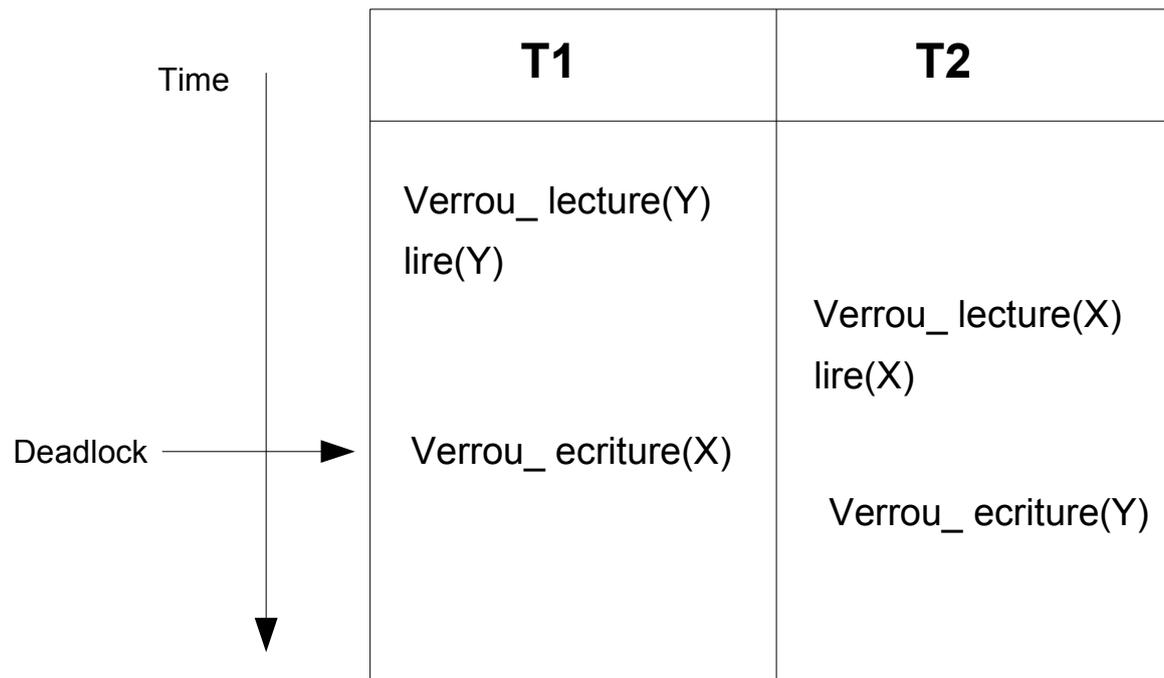
- A. Verrous – Principe
  - Verrous binaires sur chaque donnée
  - 1 = accès bloqué (donnée utilisée)
  - 0 = accès disponible (donnée libre)
  - Verrous partagés (lecture)
  - Verrous restrictifs (écriture)

# III. Contrôle de la Concurrency

- A. Verrous à Deux Phases
  - 2-phase Commit
    - = 2 Phase Locking
  - Verrous multiples pour transactions
  - Phase d'expansion/de croissance
    - Acquisition de verrous
  - Phase de rétrécissement
    - Libération de verrous

# III. Contrôle de la Concurrency

- A. Verrous - Interblocages
  - Verrous mortels
  - Plusieurs transactions attendent la libération d'un verrou



# III. Contrôle de la Concurrency

- B. Tri des estampilles
  - = TimeStamp (TS)
  - Marque le début de la transaction
  - En cas de conflit
    - annulation des transactions T avec le TS le plus grand
    - celles créées en dernier
    - Annulation des différentes requêtes de T
    - Risque d'annulation en cascade
      - Cascading rollback

# III. Contrôle de la Concurrency

- C. Niveaux de Granularité
  - Valeur d'un champ d'une occurrence
  - Occurrence
  - Bloc du disque
  - Totalité d'un fichier
  - Totalité d'une Base de Données

# III. Contrôle de la Concurrency

- C. Niveaux de Granularité et Transactions
  - Granularité Forte
    - Blocage d'une grande partie des données à chaque accès
    - Peu d'opérations de verrouillage
  - Granularité Faible
    - Blocage uniquement des données nécessaires (champs, occurrence)
    - Nombreuses opérations de verrouillage
    - Perte de performance

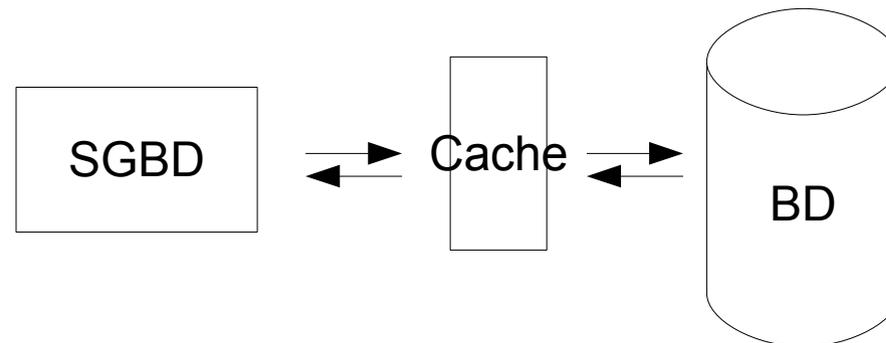
- I. Introduction
- II. Transactions et SQL
- III. Contrôle de la Concurrence
- IV. **Récupération de Bases de Données**

# IV. Récupération de Bases de Données

- A. Approches
  - Panne catastrophique
    - Reprise à partir du dernier enregistrement global de la base
  - Panne de faible ampleur
    - Annulation des dernières modifications, à partir du journal système
  - Types de mise à jour
    - Différée
      - Enregistrement dans le journal
      - Validation
      - PUIS dans la Base de Données
    - Immédiate
      - Enregistrement dans le journal et la Base de Données
      - Validation

# IV. Récupération de Bases de Données

- B. Mise en cache
  - Principe de cache
  - Exploitation du cache pour la reprise



# IV. Récupération de Bases de Données

- C. Journalisation
  - = logging
  - Enregistrement des modifications dans un fichier autre que la base de données
  - Conservation de l'évolution de la Base de Données
  - Pas seulement l'état de la Base de Données

# IV. Récupération de Bases de Données

- D. Points de Reprise
  - Checkpoints
  - Point de validation des transactions en cours
    - Suspension de l'exécution des transactions
    - Ecriture des tampons (caches) vers le disque
    - Ecriture d'un enregistrement 'checkpoint' dans le journal
    - Ecriture du journal sur le disque
    - Reprise de l'exécution des transactions
  - En cas d'erreur grave, Reprise au point de validation

# IV. Récupération de Bases de Données

- E. Annulation de Transactions
  - Echech après la mise à jour de la Base de Données
  - Rollback
  - Annulation d'une transaction T
    - Implique l'annulation des transactions ayant lu des données modifiées T
    - Cascade Rollback

# IV. Récupération de Bases de Données

- F. Reprises dans les Systèmes Multi-Bases
  - Gestionnaire globale de reprise
  - = coordinateur
  - 2 Phase Commit
    - Préparation de la validation pour toutes la bases concernées
    - Si OK, signal de validation aux Bases de Données

# IV. Récupération de Bases de Données

- G. Reprise en cas de Catastrophe
  - Sauvegardes à long terme
    - Réseau
    - Sur Bandes magnétiques
    - La Base de Données
    - Le Journal
      - Sauvegardes plus régulières
      - Moins volumineux

- Introduction
- Transactions et SQL
- Contrôle de la Concurrence
- Récupération de Bases de Données

