

**Connolly et Begg - Chapitres 22 Gestion des transactions**

Marc Philippe Parent Database Systems 6e de Connolly et Begg

## Base de données

**IFT-2004**

**Département d’informatique et de génie logiciel**

## Chapitre 22 - Objectifs

* But du contrôle de simultanéité (concurrence)
* But de la restauration de BD
* Rôle et importance des transactions
* Propriétés d’une transaction
* Signification de la sérialisation et application au contrôle de simultanéité
* Utiliser des verrous pour garantir la sérialisation
* Comment fonctionne le protocole de verrou en deux phases (V2P)
* Signification des verrous indéfinis et comment les résoudre
* Utiliser l’estampillage pour garantir la sérialisation
* Fonctionnement des techniques optimistes de contrôle de simultanéité
* Certaines causes de défaillances des BD
* But du fichier journal de transactions

Connolly et Begg Database systems 6th 2

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Chapitre 22 - Objectifs

* + Rôle des points de contrôle pendant la journalisation des transactions
  + Récupérer un système à la suite d’une défaillance
  + Des modèles alternatifs pour les transactions de longue durée.

Connolly et Begg Database systems 6th 3

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Soutien des transactions

Une Transaction

* + - C’est une ou plusieurs Action(s)

demandée par un seul utilisateur ou programme d’application,

qui lit ou MAJ la BD

C’est une unité logique de travail (C’est un bloc ça forme un tout)

C’est un programme d'application qui :

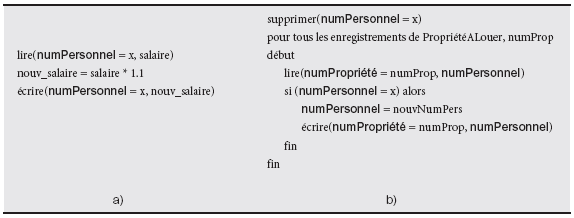
* + - Fait plusieurs transactions avec des traitement qui ne sont pas de type BD entre plusieurs applications
  + Elle transforme la BD d'un état cohérent à un autre état cohérent
    - Cette Cohérence peut être violée pendant la transaction.

Connolly et Begg Database systems 6th 4

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Exemple Transaction

**(pseudo-code)**

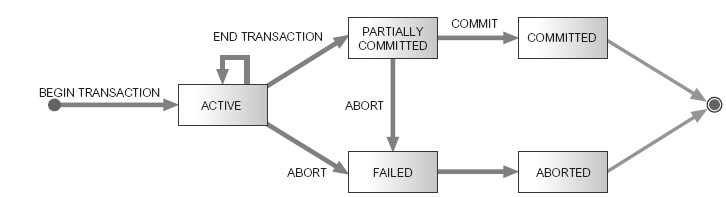
****

Connolly et Begg Database systems 6th 5

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Support aux transactions

## Partiellement valide / validé

****

**Echec Avorté (revenu a l’état initial)**

Connolly et Begg Database systems 6th 6

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Propriétés des transactions

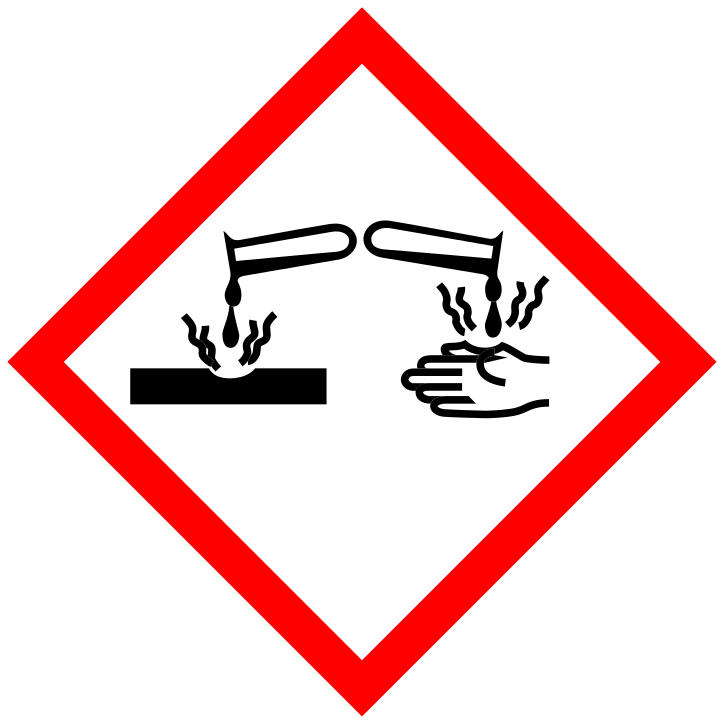
## (ACID) :

## Atomicité,

## Cohérence,

## Isolation (invisible),

## Durabilité (permanent)

****

Connolly et Begg Database systems 6th 7

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Le Contrôle de concurrence :

C’est la Gestion des opérations simultanées de la BD sans interférences

Elle prévient les interférences

Elle s’assure que les Transactions soient correctes

* Parce que l’Interaction peut produire un résultat incorrect

3 exemples problèmes concurrence.

Mise-a-jour Perdu

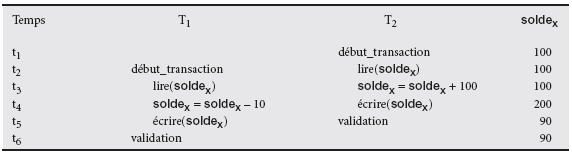
Problème de dépendance non validée

Problème d’analyse non-cohérente

Connolly et Begg Database systems 6th 8

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Problème de: MAJ perdue

****



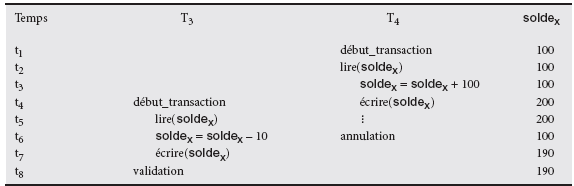
* + - Perte de la MAJ de T2 évitable qu’en interdisant T1

de lire le solde, tant que MAJ T2 pas achevée.

Connolly et Begg Database systems 6th 9

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Problème de : dépendance non validée

****



* Problème évité en empêchant T3 lire solde avant

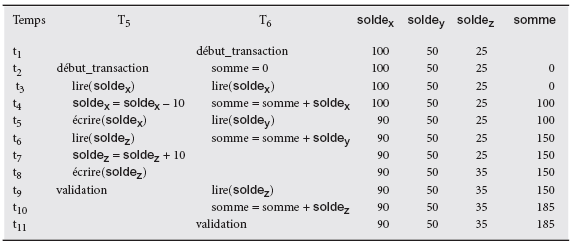
que T4

approuve ou annule.

Connolly et Begg Database systems 6th 10

Département d’informatique et de génie logiciel – Marc Philippe Parent

# Problème de: l'analyse incohérente

****



* + Problème évité en empêchant T6 lire soldes de x

et z avant que T5 ait complété la MAJ.

Connolly et Begg Database systems 6th 11

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Capacité de sérialisation

* + L’Objectif du protocole de contrôle de concurrence
    - Est de faire une planification des transactions sans interférence
  + D’Exécuter les transactions en serie
  + La Sérialisation:
    - Identifie les exécutions de transactions en assurant la cohérence
  + Une Planification c’est une
    - Séquence de lectures/écritures de transactions concurrentes préservant l’ordre des opérations de chacune

La Planification sérielle consiste à mettre les transactions une après l’autre

La Planification non sérielle : c’est d’essayer de faire les requêtes toutes en même temps.

Connolly et Begg Database systems 6th 12

Département d’informatique et de génie logiciel – Marc Philippe Parent

## La Capacité de sérialisation

* + Les actions lectures ou écritures peuvent amener des problèmes de conccurence qu’elles sont les cas? :

1. Si 2 transactions font l’action de lire:

Il n’y a pas de probleme de conccurence

1. Si 2 transactions lisent ou écrivent des données différentes,

Il n’y a pas de problème de concurrence

1. Si des transactions écrit et autre lit ou écrit la même donnée.

Il y a un problème de concurence.

C’est la que la serialization est importante.

Connolly et Begg Database systems 6th 13

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Techniques de contrôle de concurrence

* + 2 techniques de base pour controler la conccurence:
    - Le Verrouillage ou "Locking"
    - Estampillage ou "Timestamping" :

il va regarder quel transaction a commencé en premier

* + - Approches conservatrices (ou pessimistes)

Ex : Approche pessimiste :

Retarder les transactions pour éviter tout problèmes. approche pessismiste

Ex : Technique optimiste :

Laisser les requete se faire et si il y a un problème annuler la transaction.

Connolly et Begg Database systems 6th 14

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Verrouillage (technique conservatrice)

Le verrous consiste à :

* + - * Refuser les accès aux autres transactions
      * Pour Prévenir les MAJ incorrectes
    - C’est L’Approche la plus suivie dans les SGBD
    - La Transaction demande le verrou avant l’opération
      * Le Verrou partagé (lecture) empeche l’écriture
      * Le Verrou Exclusif (écriture) empeche la lecture et l’écriture.

Connolly et Begg Database systems 6th 15

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Verrouillage – Règles de base

* + Si la transaction possède unverrou partagé
    - Elle Peut lire mais pas modifier
  + Si une transaction possède un verrou exclusif
    - Elle Peut lire et modifier

Deux lectures ne sont jamais en conflit

* + - Plusieurs transactions peut avoir :
      * un verrous partagés sur la même donnée
  + Certains systèmes peuvent:

Élever le verrou de lecture en verrou exclusif

* + - Réduire le verrou exclusif en verrou partagé.

Connolly et Begg Database systems 6th 16

Département d’informatique et de génie logiciel – Marc Philippe Pare

**Exemple 22.5 – planification de verrouillage incorrect**

Planification valide, mais incorrecte

**Exemple 22.5 – planification de verrouillage incorrect**

P = {verrou\_ecriture(T9, soldex), lecture(T9, soldex), écriture(T9, soldex), déverrouillage(T9, soldex),

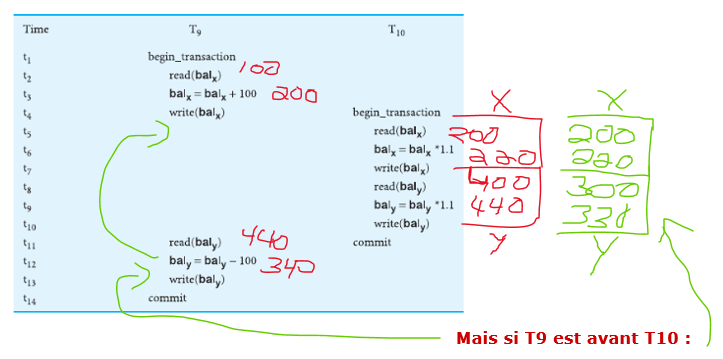
verrou\_ecriture(T10, soldex), lecture(T10, soldex), écriture(T10, soldex), déverrouillage(T10, soldex),

verrou\_ecriture(T10, soldey), lecture(T10, soldey), écriture(T10, soldey), déverrouillage(T10, soldey),

validation(T10),

verrou\_ecriture(T9, soldey), lecture(T9, soldey), écriture(T9, soldey), déverrouillage(T9, soldey),

validation(T9) }.



P n’est pas une planification sérialisable.

Ce qui veut dire que :

Cela ne donne pas le même résultat si la transaction aurait été mis en série.

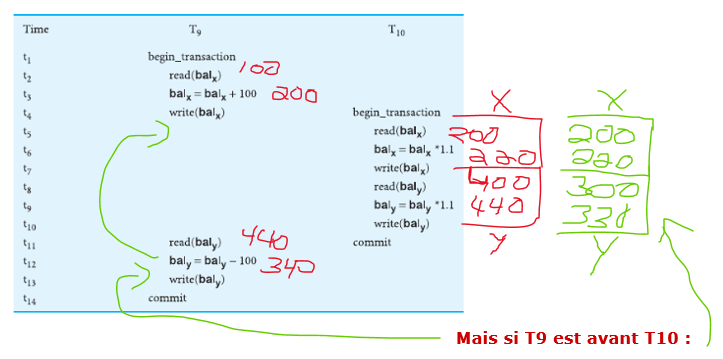
Connolly et Begg Database systems 6th 18

Département d’informatique et de génie logiciel – Marc Philippe Parent

**La planification de verrouillage incorrect amène un**

* + Problème:
    - La Planification libère les verrous trop tôt
    - Cela cause une perte d’isolation et d'atomicité (ACID)
  + La Solution: Faire un Verrouillage en 2 Phase (2PL)

Le problème de verrou : en considérant que x = 100 et y = 200



Connolly et Begg Database systems 6th 19

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Verrouillage en 2 phases (2PL)

* + Le Protocole 2PL (V2P) Consiste à faire que
    - Toutes les opérations de verrouillage précèdent la première opération de déverrouillage :
      * Ce qui veut dire :

qu’une transaction qui libère un verrou ne peux plus demander d’autres verrouillages.

* + Il y a 2 phases

La première phase s’appel Croissance:

* + - * Acquière tous les verrous n'en libère aucun

La deuxième phase s’appel : La Résorption :

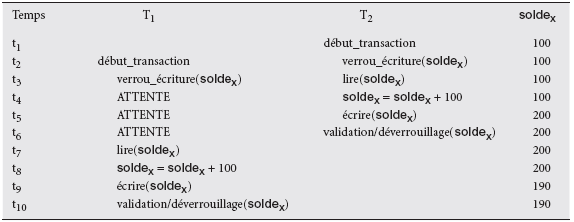
* + - * Relâche verrous, ne peut plus en obtenir d’autres.

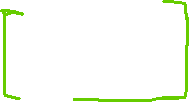
Connolly et Begg Database systems 6th 20

Département d’informatique et de génie logiciel – Marc Philippe Parent

**- Dans ce cas le protocole 2PL permet d’éviter un problème de :**

**Mise à jour perdue.**

****

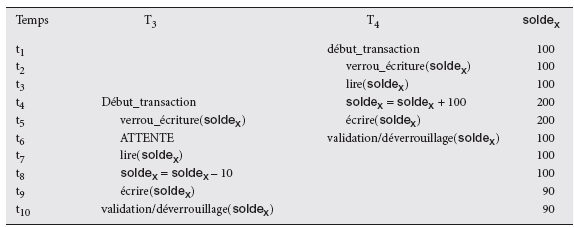


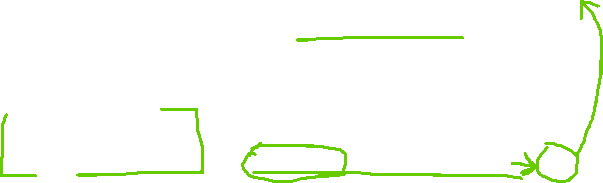
Connolly et Begg Database systems 6th 21

Département d’informatique et de génie logiciel – Marc Philippe Parent

**Dans ce cas le protocole 2PL permet d’éviter un problème de :**

**Dépendance non validée**

****



annulation

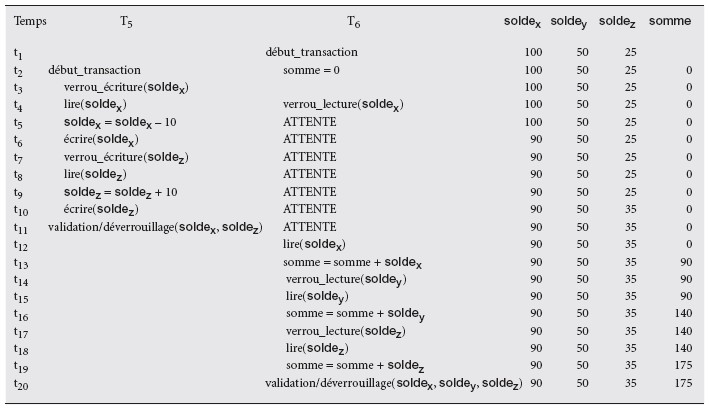


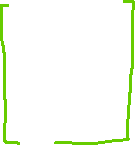
Connolly et Begg Database systems 6th 22

Département d’informatique et de génie logiciel – Marc Philippe Parent

**Dans ce cas le protocole 2PL permet d’éviter un problème de :**

**D’analyse incohérente**

****



Connolly et Begg Database systems 6th 23

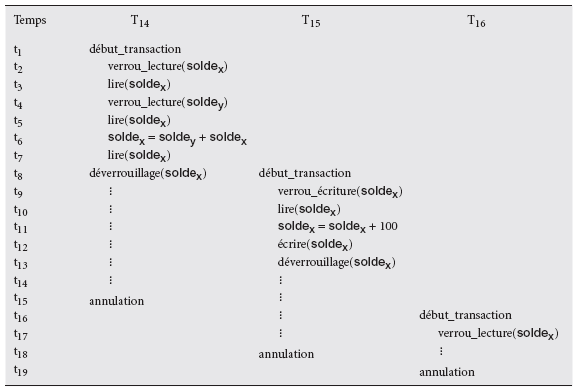
Département d’informatique et de génie logiciel – Marc Philippe Parent

## Il reste un problème présent en 2PL celui-ci est :

## L’Annulation en cascade

**Cela se produit lorsqu’il y a annulation d’une transaction qui a libéré les verrous, et une autre transaction demande le verrou avant cette annulation.**

Transactions conformes 2PL

* + T14 -> échouée -> annulée
  + T15 dépend de T14

T15 aussi annulée

* + T16 dépend de T15
    - aussi annulée

Annulation en cascade

La solution à ce problème est de:

Ne faire aucuns relâchements

de verrous avant la fin de

la transaction.

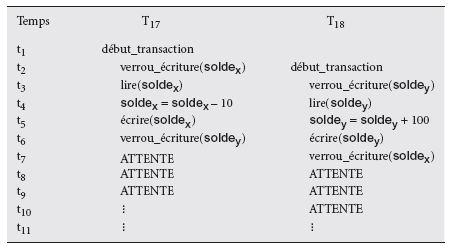


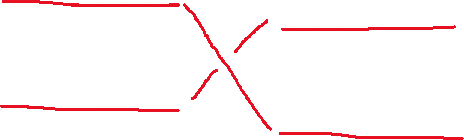
Connolly et Begg Database systems 6th 24



Département d’informatique et de génie logiciel – Marc Philippe Parent

## Problème de: Blocage indéfini (Deadlock)

****



Connolly et Begg Database systems 6th 25

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Blocage indéfini

* + Le seul moyen de briser un Blocage indéfinie est

D’abandonner 1 ou plusieurs transactions

* + Cela doit être fait de manière transparent (invisible) à l'utilisateur

Le SGBD doit annuler et répartir ces transaction(s)

* + Il y a 3 techniques générales de gestion de blocage :
    - Le délai imparti (fait à l’aveugle)
    - La prévention (les prévient)
    - La détection et la restauration. (Les détectes)

## Le délai imparti,

## La prevention

## et La detection et la restauration

## sont des techniques de gestions de blocages (DEADLOCK)

## Délai imparti (timeout)

* + Quand une transaction demande un verrou:
    - Elle a un compte à rebours :

Le serveur attendra que pour un temps maximal alloué.

* + Si le verrou n’est pas obtenu dans ce délai:
    - La Demande est annulée
  + Le SGBD suppose que la transaction est en blocage indéfini
    - Cela Annule transaction
    - Et la transaction est relancée automatiquement.

Un délai imparti court:

Peut souvent se tromper et annuler des transactions qui ne sont pas en deadlock.

Un délai imparti long:

augmente les chances de trouver un vrai deadlock mais crée un temps d’attente trop long.

Connolly et Begg Database systems 6th 27

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Pour faire la Prévention des blocages indéfinis :

Le SGBD regarde en avant pour vérifier si un blocage va se produire,

Puis l’empêche en :

Ordonnant les transactions avec des estampilles :

L’estampille (Attendre - Mourir) ou (Wait, Die) consiste à :

* + - * Faire attendre une ancienne transaction après une transaction plus récente

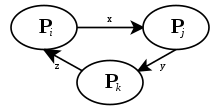
Tandis que l’estampille : (Blessé- Attend) ou (Wound-Wait)

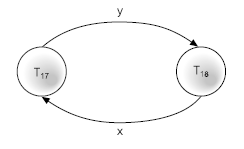
* + - * Fait attendre une transaction plus récente après une plus ancienne.

Connolly et Begg Database systems 6th 28

Département d’informatique et de génie logiciel – Marc Philippe Parent

## La Détection des blocages indéfinis consiste à

* + Laisser le SGBD permettre les blocages indéfinis
    - Celui-ci est capable de reconnaitre les deadlocks et de les briser.
  + Cela est géré par des graphes d’attentes (wait-for graph)
  + 
    - Un nœud est créé pour chaque transactions
    - Une flèche Pi -> Pj, est créé si Pi attend un verrou de Pj
  + Il y a un Blocage indéfini si un cycle est formé (une boucle)
  + Le SGBD va créer un graphe des attentes à intervalle réguliers.



**Comment le SGBD va faire la Récupération à la suite de la détection d'un blocage indéfini :**

Il y a plusieurs solutions :

* Faire le choix de la victime par moindre coût :

Cela veut dire : de choisir la transaction qui n’a pas modifié beaucoup de données ou n’a pas fait un long temps d’exécution.

* Faire le choix selon la profondeur d’annulation de transaction

La profondeur veut dire combien d’autre transaction cela va annuler en cascade. À quel point la transaction est avancée.

* Faire le choix d’éviter la famine :

Veut dire de ne pas toujours choisir la même transaction à annuler.

Connolly et Begg Database systems 6th 30

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Exercices

* + Les 4 propriétés des transactions?

A-Atrocité, Calculabilité, Indépendance, Développement B-Atomicité, Cohérence, Isolation, Développement

C-Atrocité, Calculabilité, Isolation, Durabilité D-Atomicité, Cohérence, Isolation, Durabilité

E-Atomicité, Calculabilité, Indépendance, Développement

* + Pour résoudre les blocages indéfinis, le serveur peut

A-Utiliser le délai imparti

B-Les prévenir en regardant en avant

C-Les détecter à l'aide d'un graphe des dépendances D-Aucune de ces réponses

E-Toutes ces réponses

Connolly et Begg Database systems 6th 31

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Les Méthodes d'estampillage (méthode pessimiste)

## consiste à:

* + Ordonner globalement les transactions
  + Les conflits sont résolus en annulant et en redémarrant une transaction récente
  + Il n’y pas de verrous dans cette méthode et donc:
    - * Il n’y a pas de verrou indéfini
  + Utilisation des Estampille – timestamp:
    - C’est un Identificateur unique créé par le SGBD pour chaque transaction.
    - Elle Indique l’heure relative du démarrage de chaque transaction
      * Cela est base sur l’horloge système (date,heure,min etc)
      * ou un Compteur logique (1..2..3..4..etc).

Connolly et Begg Database systems 6th 32

Département d’informatique et de génie logiciel – Marc Philippe Parent

## L’Estampillage consiste

* + À autoriser une Lecture ou écriture seulement si
    - * + La dernière MAJ apportée par une transaction est plus ancienne

Sinon

* + - La Transaction qui demande la lecture/ou l’écriture sera redémarrée avec une nouvelle estampille.
  + L’Estampille de donnée, c’est une identification qui détermine qu’elle est la dernière transaction qui a lu ou écrit la donnée.
    - L’Estampille-lecture : Indique la dernière transaction ayant lu la donnée

L’estampille-Écriture : Indique la dernière transaction ayant écrit la donnée

Connolly et Begg Database systems 6th 33

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Une transaction fait une demande de lecture(x) avec la méthode d’estampillage comment cela fonctionne?

**Quand on veut lire : on vérifie l’estampille d’écriture.**

La Transaction T possède l'estampille e(T)

* + - Si e(T) < estampille\_ecriture(x)
      * La donnée x a déjà été mis à jour (écrite) par une transaction plus récente.
      * La transaction est annulée et redémarrée avec une nouvelle estampille
    - Sinon la lecture est permise
      * estampille\_lecture(x) = max(e(T), estampille\_lecture(x)).

Explication pour la dernière formule, l’estampille de lecture, devient la valeur la plus haute dans ces choix : (Soit l’estampille de cette transaction courante, soit elle a déjà été lu par une estampille de lecture plus récente encore.)

Connolly et Begg Database systems 6th 34

Département d’informatique et de génie logiciel – Marc Philippe Paren

## Une transaction fait une demande d’écriture(x) avec la méthode d’estampillage comment cela fonctionne?

**Quand on veut écrire, on vérifie l’estampille de lecture.**

* + Si e(T) < estampille\_lecture(x)
    - La donnée x a déjà été lu par une transaction plus jeune
    - T est annulée et redémarrée avec une nouvelle estampille
  + Sinon si e(T) < estampille\_écriture (x)
    - La donnée x a déjà été écrite par une transaction plus jeune qui n’a pas fait de lecture.
    - L’écriture est ignorée car elle n’a pas besoin d’être lue.
  + Sinon (la donnée, a été lu et modifiée par une plus ancienne) :

L’opération est acceptée et executée

estampille\_écriture(x) = e(T).

Connolly et Begg Database systems 6th 35

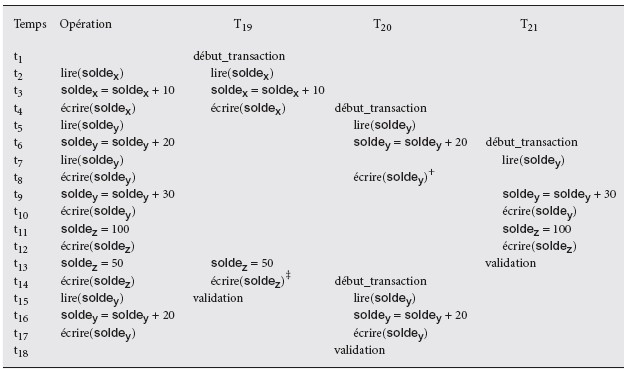
Département d’informatique et de génie logiciel – Marc Philippe Par

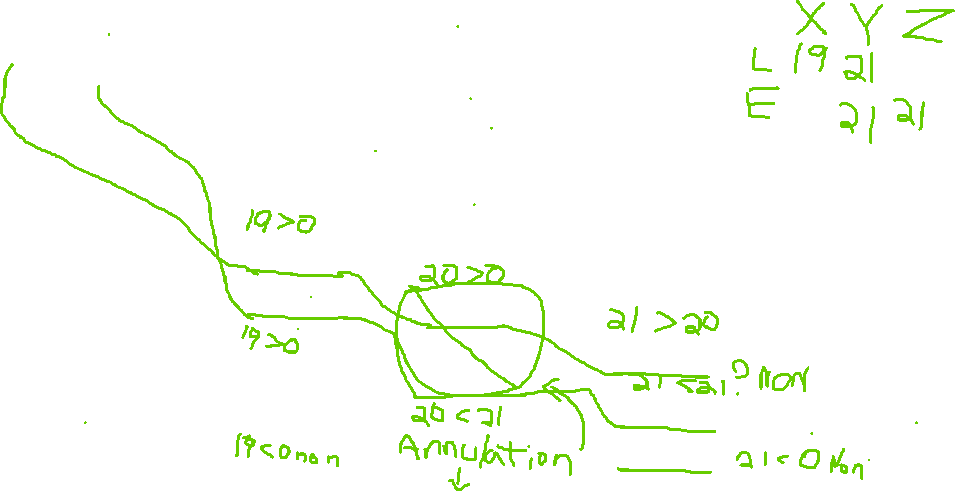
**Exemple 22.10 – Classement élémentaire par ordre d'estampille**

**Est-ce que l’estampille de T < que l’estampille d’écriture? (oui?) annulation**

**Est-ce que l’estampille de T < que l’estampille de lecture? (oui?) annulation**

**Est-ce que l’estampille de T < que l’estampille d’écriture? (oui?) ignorer**

****



## Le Tri des estampilles multiversions

* + Fait la gestion de plusieurs versions de données
    - Afin d’améliorer la concurrence
  + Dans l’estampillage de base
    - Il y a une seule version de la donnée
      * Une seule transaction peut y accéder
  + Si plusieurs transactions lisent/écrivent différentes version de la même donnée
  + Chaque transaction voit son ensemble cohérent
  + Une transaction qui fait une écriture  Va créer une nouvelle version de la donnée
  + Lire  Sélectionne une version qui va permettre la sérialisation = OK
  + Les versions seront effacées après pour conserver une seule version.

Connolly et Begg Database systems 6th 37

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Techniques optimistes

* + Repose sur l’Hypothèse

Que les conflits sont rares :

* + - Il est peut-être plus efficace de laisser transactions s’exécuter sans délai
  + À la validation :
    - Il y a une vérification pour déterminer s’il y a eu conflit
    - S’il y a eu conflit :
      * La transaction est annulée et redémarrée
      * Il y a potentiellement une plus grande concurrence car il n’y jamais annulation avant la fin de la transaction.
      * Il n’y a pas de verrou dans cette technique.

Les techniques optimistes se passent en 3 Phases :

1 : Lecture

## Techniques optimistes

## Phase 1 – Lecture

## La phase commence au début et se termine immédiatement avant la validation de la transaction

## La transaction lit les valeurs BD et les stocke dans des variables locales. (à l’extérieur de la BD)

Phase 2 – Validation

* + - S’il y a validation des transactions qui ne font que de la lecture :

Il y aura vérification si les données ont toujours les valeurs courantes :

Autrement dit s’il n’y a pas eu écriture durant la transaction.

* + - S’il y a validation des transactions qui font des mises à jour :

Il y aura vérification si la BD est toujours dans un état cohérent

* + - S’il n’y a pas eu d'interférence  La transaction est validée
    - Sinon  Elle est Annulée et redémarrée
* Phase 3 – écritures - Transactions MAJ
  + Elle suit la phase de validation lorsque terminée avec succès
  + Tout ce qui est MAJ localement est appliqué (enregistré) à la BD.

## La Récupération d'une BD

## Ou restauration de BD :

## Les SGBD ont besoin d’un contrôle de récupération

## Il y a 3 types de stockage de données

* + - Volatile (mémoire principale) (RAM)
      * Quand le serveur s’exécute (en fonction) il utilise ce stockage.
    - Non-volatile (disque dur)
      * C’est là où les données sont stockées.
    - Stockage hors ligne.
      * C’est là où se trouve les sauvegardes pour la récupération durant les pannes.

Connolly et Begg Database systems 6th 40

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Types de pannes

* Plantage système : La mémoire volatile (RAM) est perdue.
* Défaillances média (un disque dur peut lâcher)
* Erreurs logicielles
* Catastrophes naturelles
* Destruction données/équipements
* Sabotage

Connolly et Begg Database systems 6th 41

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Transactions et récupération

Les transactions deviennent :

L’unité de récupération

* Le Gestionnaire de récupération est responsable
  + De l’atomicité des transactions.
    - Si une transaction pas validée :

On doit défaire ses effets après le redémarrage.

* + De la Durabilité :
    - S’il y a eu une panne entre la validation et l’écriture sur le stockage secondaire (disque) :
      * Refaire la mise-à-jour (transaction)

Connolly et Begg Database systems 6th 42

Département d’informatique et de génie logiciel – Marc Philippe Parent

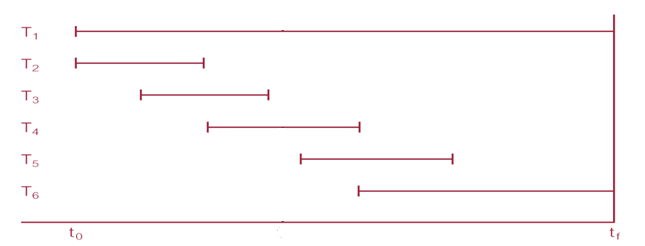
## Transactions et récupération

**Durant le temps T0 les transaction T2, T3, T4, T5, ont été commited. Puis il y a une panne tf.**

**Au redémarrage le SGBD devra s’assurer que**

**les transactions T2-T3-T4-T5 soient : inscrit sur le disque.**

**et que les transactions T1 et T6 soient : Annulées et ensuite redémarrées**

****

Connolly et Begg Database systems 6th 43

Département d’informatique et de génie logiciel – Marc Philippe Paren

## Gestion des tampons

* Le Gestionnaire tampons va :
  + Administrer les transferts entre les pages mémoire et le stockage secondaire. (Transfert données entre Disque et RAM)
* Il va Lire les pages disque vers tampons jusqu’à saturation du tampon
* Il ne va pas lire une page sur le disque qui est déjà dans un tampon

Pour cela le gestionnaire va utiliser différentes stratégies :

* La Stratégie de vol : avant la validation, il prend une des page et l’écrit en mémoire.
* La Stratégie d’antivol : s’il n’y a pas eu de commit, elle empêche l’écriture de la page.
* Stratégie de forçage : Il y a écriture immédiatement au moment du commit dans le disque dur.
* Plus simple en pratique.

Connolly et Begg Database systems 6th 44

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Utilitaires de récupérations

* Le SGBD fournit outils pour récupération
  + Un Mécanisme de sauvegarde (backup)
  + La journalisation (Log) : Il conserve des traces sur les transactions)
  + Les point de contrôles : point pour controle ce qui se passait a ce moment là.
  + Les Gestionnaires de récupération : Il va s’executer lorsqu’on redémarre le serveur.

Connolly et Begg Database systems 6th 45

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Le Fichier journal

**Son but est de**

* Conserver des Informations sur les MAJ de BD

Aussi pour autres fonctions ( )

* Peut être double ou
* Parfois séparé en deux fichiers à accès
* Goulot d'étranglement potentiel.

Connolly et Begg Database systems 6th 46

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Fichier journal

* Enregistrements de transaction

de transaction

d'enregistrement

Image

Image

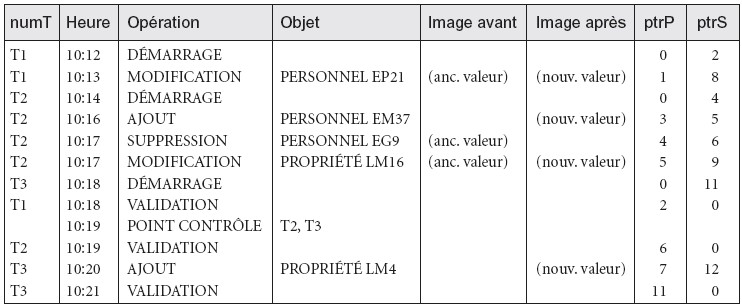
* + IInformation de

du journal.

Connolly et Begg Database systems 6th 47

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Segment de fichier journal

****

Connolly et Begg Database systems 6th 48

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Points de contrôle

* Point de

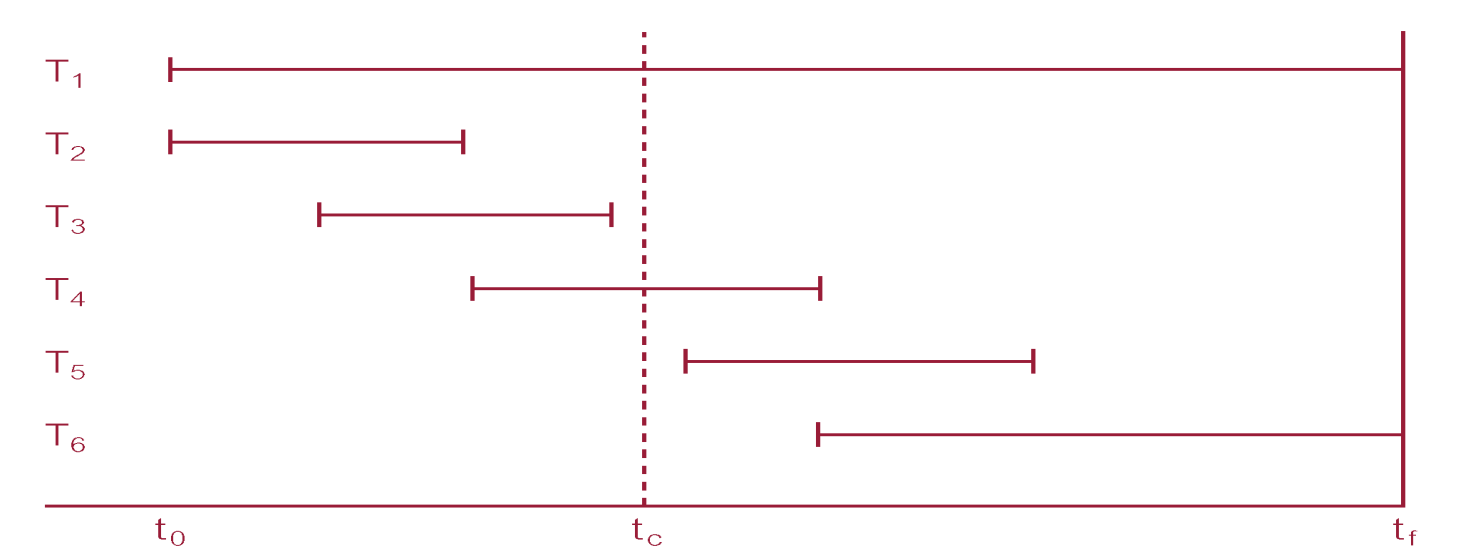
entre BD et journal

* Écriture forcée des tampons sur stockage secondaire
* Enregistrement point de contrôle créé
* Panne.

Connolly et Begg Database systems 6th 49

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Points de contrôle

****

Connolly et Begg Database systems 6th 50

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Techniques de récupération

* Si BD
  + Restaurer dernière
  + Refaire MAJ transactions validées avec journal
* Si DB
  + changements incohérence
  + Refaire transactions pour que MAJ soient sur stockage secondaire
  + Pas de
    - Restaurer BD avec images avant et après.

Connolly et Begg Database systems 6th 51

Département d’informatique et de génie logiciel – Marc Philippe Parent

## La Pagination à l'arrière-plan

* Consiste a conserver 2 tables de page pendant la transaction

Courante

* + Arrière-plan
* Au Début de la transaction

Les 2 pages identiques

* Pendant la transaction
  + La page d’Arrière-plan
    - N’estJamais modifiée
    - Elle est Utilisée pour restauration BD si panne

La page Courante enregistre toutes les MAJ

* Lorsque que Complétée  La page Courante

devient Arrière-plan.

Connolly et Begg Database systems 6th 52

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Modèles avancés de transaction

## Il y en a 5 a apprendre, seulement 2 sont présenté ici.

Si les Transactions sont de longue durée

* + Et qu’il y a une Défaillance :
    - On Ne peut l’annuler/recommencer
  + On ne peut Verrouiller un grand nombre données
    - La Concurrence sera limitée si les données sont inacessible

longtemps

* + - Verrou indéfinis plus probable
  + La Coopération sera restreinte sur l’utilisation des données
* Nous voyons 2 modèles
  + Modèle de transactions imbriquées
  + Modèle de flux de travaux.

Connolly et Begg Database systems 6th 53

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Modèle de transactions imbriquées

Transaction

hiérarchie

Haut niveau

de sous-transactions

Plusieurs transactions enfant

Enfant

Peut avoir transactions imbriquées

Niveau feuille

* + Exécute opérations de BD
* C’est Validées

Du Bas vers le Haut .

Connolly et Begg Database systems 6th 54

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Modèle de transactions imbriquées

* Si il y a un Annulation à un niveau

Cela ne va pas Affecter les transaction au niveau supérieur

* Parent peut récupérer en

Réessayant de faire la sous-transaction

Ignorant les pannes (de la transaction)

* + - Pour les Sous-transaction non vitale
  + Exécutant une sous-transaction alternative

Tout avorter (en dernier recours)

* Les MAJ des sous-transactions niveaux intermédiaires visibles seulement

dans portée parentsimmédiats.

Connolly et Begg Database systems 6th 55

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Modèle de transactions imbriquées

* Une Validation de sous-transaction

Est Conditionnelle

Au Haut-niveau

* + ACID est respecté
* Les Avantages sont

à la validation/annulation de ses supérieurs

* + C’est Décomposable en sous-transactions
    - Simultanéité et récupération

Granularité plus fine (verrou le plus petit possible)

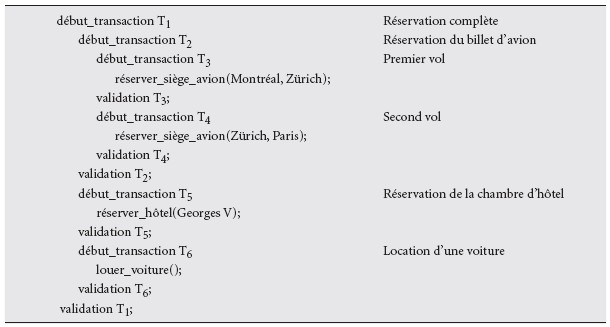
Parallélisme dans transaction

Récupération dans transaction

Connolly et Begg Database systems 6th 56

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Exemple de transactions imbriquées

****

Connolly et Begg Database systems 6th 57

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Modèles de flux d'activité

* Modèle précédent
  + Pas puissants avec certaines activités industrielles et

commerciales

* Flux d'activité
  + Implique exécution coordonée de traitement

de tâches effectuées par entités

* + - Personnes ou systèmes logiciels
* 2 problèmes généraux
  + Spéficiation et exécution
* Compliqués par plusieurs organisations utilisent plusieurs

systèmes indépendants pour automatiser processus.

différentes parties du

Connolly et Begg Database systems 6th 58

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Exercices

* Dans l'estampillage multiversions, il existe

A- plusieurs versions des données

B- plusieurs versions des estampilles

C- plusieurs versions de page (avant-plan et arrière plan) D- plusieurs versions de verrous

E- Aucune de ces réponses

* Laquelle est fausse, dans le modèle imbriqué des transactions

A-Une sous-transaction est valide si supérieurs le sont B-les sous-transactions respectent les propriétés ACID

C’est les haut niveau qui respecte les propriétés acid et non les sous-transactions

C-Il peut y avoir du parallèlisme entre les sous-transactions D-Aucune de ces réponses

E-A et C F-B et C G-A et B H-Toutes ces réponses

Connolly et Begg Database systems 6th 59

Département d’informatique et de génie logiciel – Marc Philippe Parent

## Références

* image ACID https://upload.wikimedia.org/wikipedia/commons/a/a1/GHS-pictogram-acid.svg

Connolly et Begg Database systems 6th 60

Département d’informatique et de génie logiciel – Marc Philippe Parent