BUT 3
Nouveaux paradigmes de bases de données
2023-2024

Gilles Nachouki

Plan du cours

- Optimisation des requêtes - App. Oracle
- Gestion des transactions - App. Oracle
- Modèle de données Relationnel-Objet - App. Oracle
- Big Data et MapReduce - App. MongoDB
- Réplication et repartition de données - App. MongoDB

Définition

Une transaction est formée d'une suite d'opérations permettant de passer d'un état cohérent vers un autre. Chaque opération peut être :

- Une opération de positionnement : set transaction (début)
- Une opération de lecture : Find: F(R)
- Une opération de mise à jour des données : Update : U(R)
- Une opération de validation : Commit : permet de confirmer les modifications
- Une opération d'annulation : Rollback : permet d'annuler les changements
- La création d'un point de sauvegarde : Savepoint : permet de spécifier UnPoint de retour dans la transaction (Rollback to UnPoint)

Exemple

- DT(T1) : début de la transaction T1
  1. F(R) : Accès aux données
  2. U(R) : Mise à jour des données
  3. (C)ommit ou (R)ollback : Terminaison

Problème

Dans l'hypothèse où il existe plusieurs transactions qui s'exécutent simultanément (par exemple plusieurs utilisateurs).
Problème : comment garder la base de données dans un état cohérent après exécution simultanée de ces transactions?

Propriétés

1. (**A**)tomicité : Une transaction doit effectuer toutes ses mises à jour ou ne rien faire du tout
2. (**C**)ohérence : Une transaction doit faire passer la base de données d'un état cohérent à un autre état cohérent
3. (**I**)solation : Les résultats d'une transaction ne doivent être visibles aux autres transactions qu'une fois validée
4. (**D**)urabilité : Le système doit garantir que les modifications validées par une transaction sont conservées en cas de panne

Contrôle de concurrence - objectifs

- Rendre invisible le partage simultané des données
- Eviter les pertes de mises à jour et/ou l'apparition d'incohérences

Nécessite des protocoles particuliers ...

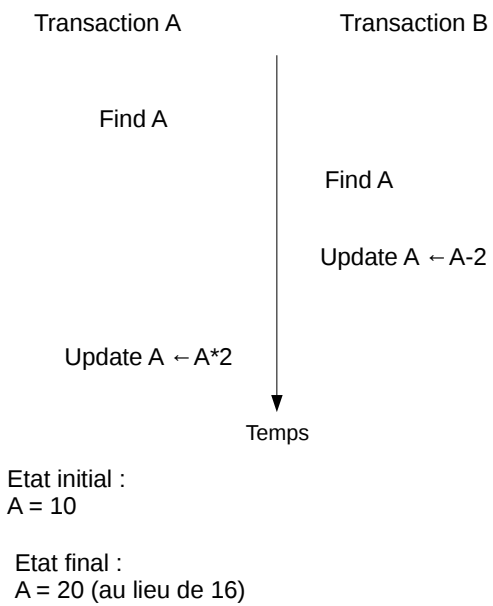
Ordonnancement

Un ordonnancement est composé d'opérations de plusieurs transantions concurrentes de type lecture (F) ou d'écriture (U).

Exemple : DT(T1);F1(x1);DT(T2);F2(x1);U1(x1)...

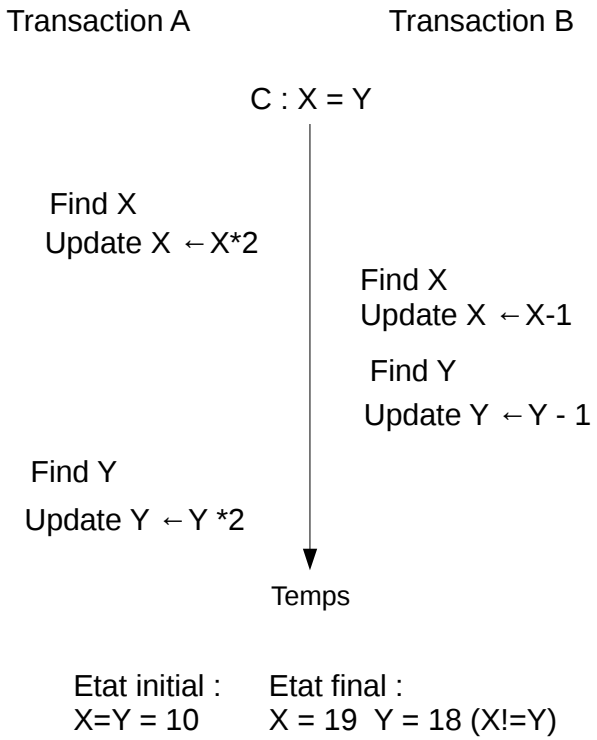
Perte de mise à jour

Une transaction A lit un tuple et le modifie après sa mise à jour par une autre transaction B.



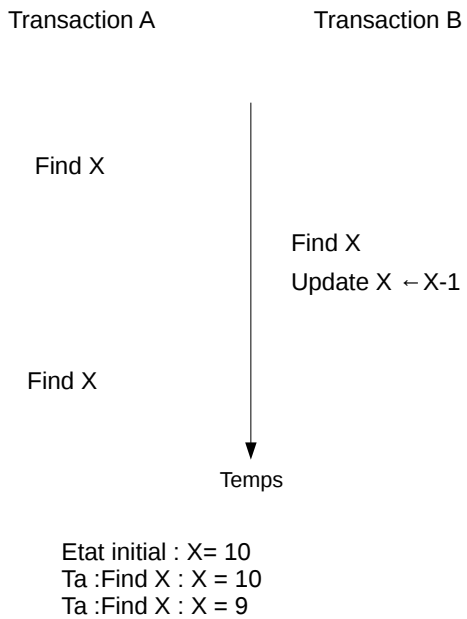
Incohérence de données

Des contraintes d'intégrité ne sont pas respectées dans la base de données



Lectures non-reproductible

Une transaction A accède, en lecture à un tuple qu'elle avait déjà lu auparavant alors que ce tuple a été modifié entre temps par une autre transaction B

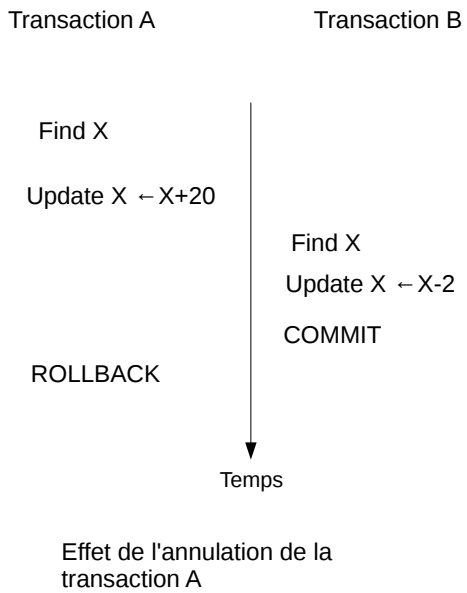


Tuples-fantômes

Une transaction t ajoute un tuple dans une table alors qu'une autre transaction t' a déjà demandé le nombre de tuples de cette table. Problème : t' ne comptabilise pas le tuple qui vient d'être inséré par t.

Lecture sale

une transaction lit un tuple mis à jour par une transaction A, avant que cette dernière ait validé



Définitions

- Granule : Un granule est l'unité de données contrôlée par le SGBD, il s'agit par exemple, d'un tuple, d'une table ou d'une page ou bloc de données.
- Opération : Une opération est l'unité indivisible exécutée par le SGBD sur un granule. Il s'agit généralement d'une lecture (F), d'une écriture (U), commit (C) ou Rollback (R).
- Exécution de transactions : Exécution d'opérations obtenues en intercalant les actions des transactions tout en respectant l'ordre interne de chaque transaction.

Opérations permutables

Oi et Oj sont deux opérations permutables si le résultat de l'exécution de Oi suivi de Oj est le même que celui de Oj suivi de Oi.

Opérations permutables - Exemple

Opération A	*Opération B*
F(X)	F(X)
X ← X+4	X ← X+3
U(X)	U(X)

Les deux opérations A et B sont permutables

Opération C	*Opération D*
F(X)	F(X)
X ← X+4	X ← X*3
U(X)	U(X)

Les deux opérations C et D ne sont pas permutables

Succession et exécution sérialisable

Succession de transactions : Une succession est une exécution successive de transactions (sans simultanéité entre transactions).
Exécution sérialisable de transactions : Une exécution sérialisable de transactions est une exécution de transactions donnant le même résultat qu'une succession de transactions.

Exécution sérialisable - Exemple

Transactin A	***Transaction B***	***Transaction C***
Find X	Find Y	Find Y
X ← X+4	Y ← Y*3	Y ← Y-3
UPD X	UPD Y	UPD Y

Exécutions concurrentes A&B, B&C

Ta :Find X
Tb :Find Y
Ta :X ← X+4
Tb :Y ← Y*3
Tb :UPD Y
Ta :UPD X

Tc : Find Y
Tb : Find Y
Tc : Y ← Y-3
Tc : UPD Y
Tb : Y ← Y*3
Tb : UPD Y

Exécution Sérialisable ***Exécution non Sérialisable***

Graphe de précédence et sérialisabilité

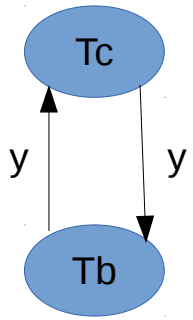
Précédence : On dit qu'une transaction Ti précède Tj si : Ti accomplit une opération Oi avant la transaction Tj qui veut accomplir une opération Oj sur un même granule où Oi et Oj sont en conflits : lecture puis écriture, écriture puis lecture ou écriture puis écriture.

Graphe de précédence : Graphe dont les noeuds sont des transactions et un arc de Ti vers Tj signifie que Ti précède Tj dans l'exécution d'une opération concernant le même granule.

Une condition pour qu'une exécution de transactions soit sérialisable : le graphe de précédence est sans circuit.

Graphe de précédence - Exemple

Tc : Find Y
Tb : Find Y
Tc : Y ← Y-3
Tc : UPD Y
Tb : Y ← Y*3
Tb : UPD Y



Circuit : Exécution non sérialisable

Graphe de précédence

Contrôle de concurrence - Mécanismes de contrôle

Dans un environnement multi-utilisateurs, un mécanisme de contrôle de concurrence est indispensable pour éviter des problèmes tels que la perte des mises à jour, le problème d'incohérence ou des données non reproductible.

Les techniques les plus connues pour éviter ces problèmes sont les **techniques de verrouillage** : une transaction qui souhaite mettre à jour un granule doit tout d'abord obtenir un verrou sur ce granule.

Nous introduisons dans ce chapitre les différents types de verrous et les protocoles associés.

Verrou exclusif - noté X

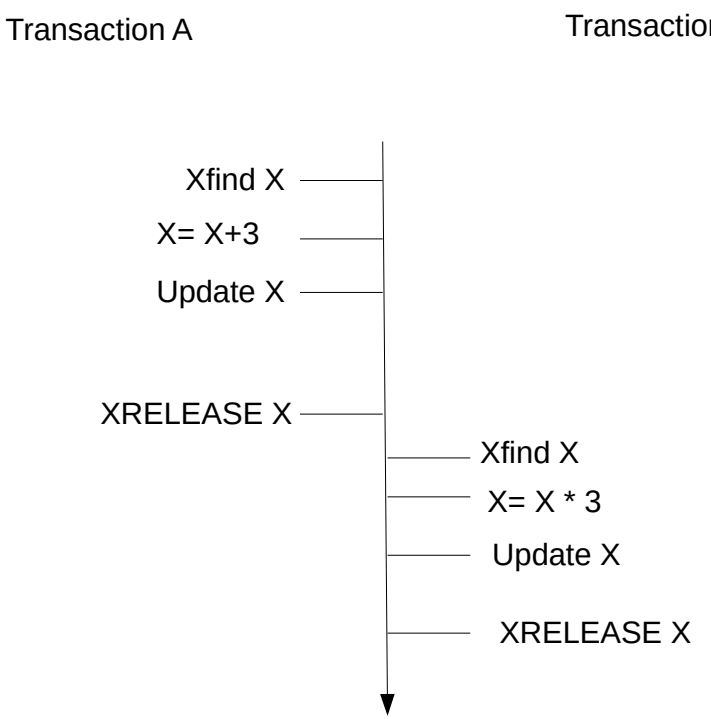
Si une transaction obtient un verrou exclusif sur un granule, alors aucune autre transaction ne pourra acquérir un verrou de quelque type que ce soit sur ce granule tant que la transaction n'a pas libéré le verrou.

Protocole exclusif - noté PX

Toute transaction qui doit mettre à jour un granule doit d'abord exécuter une opération *XFIND* pour acquérir un verrou exclusif sur le granule.
Si l'acquisition du granule ne peut être réalisée alors la transaction se met en attente. La transaction reprend son exécution quand le granule est libéré.
Le protocole PX libère le verrou juste après la mise à jour du granule par la transaction en exécutant l'opération *XRELEASE* (souvent implicite et lié à la réécriture du tuple mis à jour).

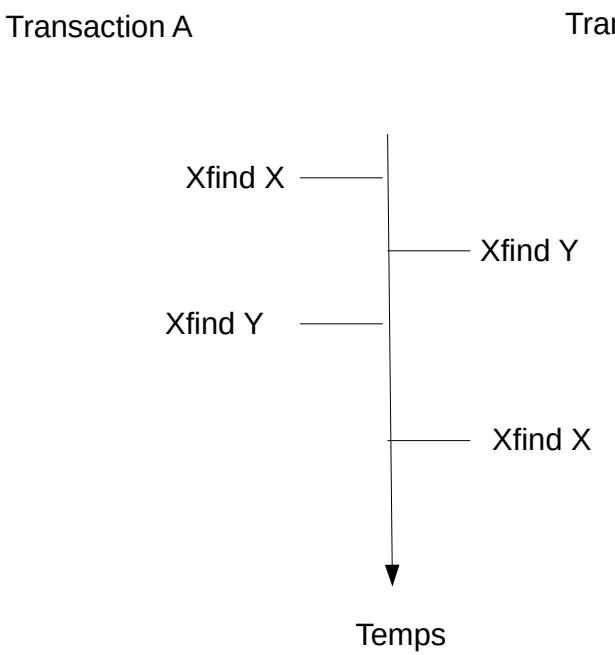
Verrous exclusifs - Exemple 1

Avec le protocole PX, on résout le problème de "pertes de mises à jour".



Protocole exclusif - noté PX

Le protocole PX peut conduire au problème d'interblocage.



Situation d'interblocage :
A et B sont mutuellement bloquées

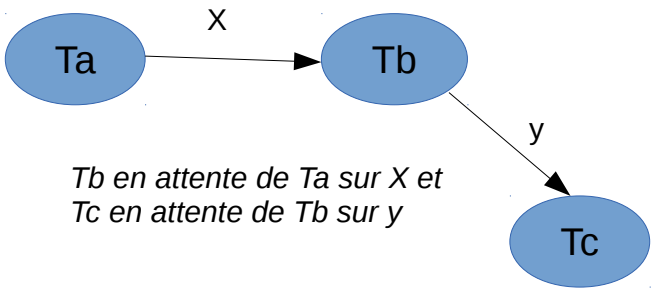
Graphe d'attente

Le système doit être capable de détecter les interblocages entre transactions. Un graphe d'attente permet d'observer le phénomène **d'interblocage** entre transactions.
Un graphe d'attente est un graphe dont les noeuds correspondent aux transactions et les arcs représentent les attentes entre transactions.

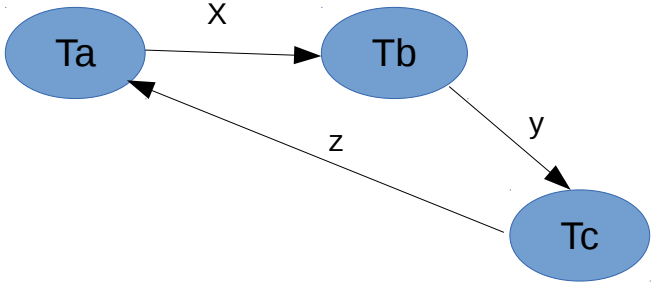
Definition

Transaction en attente : Une transaction B *attend* une autre transaction A si B demande un verrou sur un granule qui est déjà occupé par A.

Exemple : Graphe d'attente



Graphe d'attente sans circuit



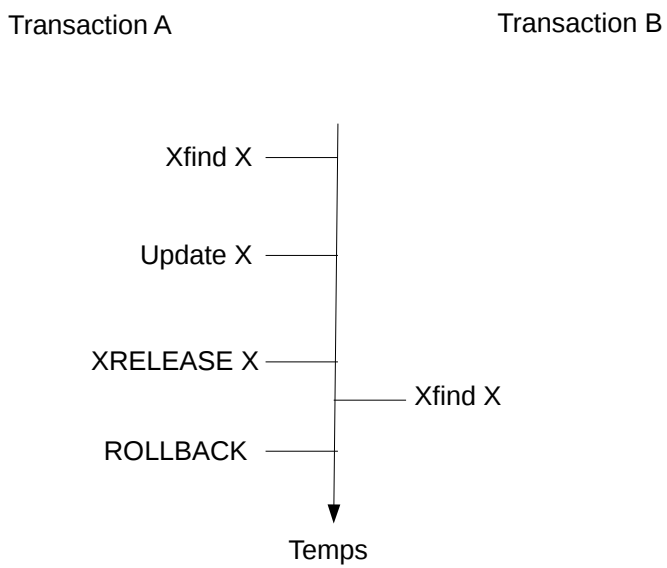
Graphe d'attente avec circuit - interblocage

Interblocage - résolution

On choisit une transaction **victime** pour laquelle on doit annuler les opérations effectuées par cette transaction (ROLLBACK). Le choix de cette transaction peut se faire selon différents critères : la plus récente, ou celle qui mobilise le plus petit nombre de verrous etc. Concernant la transaction annulée, plusieurs solutions : on peut décider de la redémarrer immédiatement (RETRY) après avoir fait disparaître le circuit du graphe ou après avoir laissé s'écouler un certain laps de temps.

Protocole exclusif - noté PX

Le protocole PX ne permet pas de résoudre le problème d'incohérence (ou lecture sale) lorsqu'une transaction décide à la fin d'annuler toutes ses opérations.

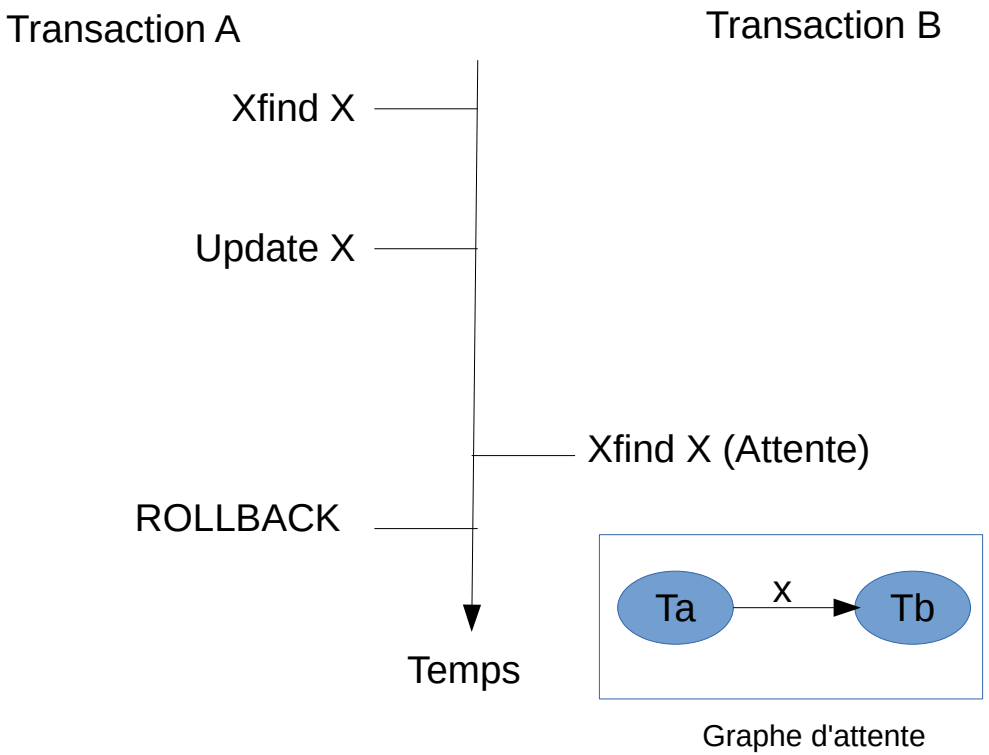


B lit une valeur annulée ensuite par la Transaction A

Protocole eXclusif Etendu - noté PXE

Pour pallier aux problèmes de pertes de mise à jour provoquées par l'annulation d'une transaction, le protocole PXE étend PX dans le sens où les verrous sont maintenus jusqu'à la fin de la transaction.

PXE - Exemple



Graphe d'attente

Verrou partagé - noté S

Si une transaction possède un verrou partagé sur un granule alors une autre transaction peut acquérir un verrou partagé sur ce granule mais aucune transaction ne peut acquérir un verrou exclusif sur le même granule jusqu'à ce que les verrous partagés soient libérés. L'acquisition d'un verrou partagé est réalisée par la commande *SFIND*.

Protocole partagé - noté PS

Toute tentative de mise à jour d'un granule partagé par une transaction transforme le verrou partagé en verrou exclusif et provoque sa mise à jour.
Les systèmes supportant le protocole PS montrent qu'il se produit de nombreux interblocages.

Protocole partagé Etendu - noté PSE

Le protocole PSE étend le protocole PS dans le sens où les verrous sont libérés à la fin de la transaction (idem que PXE pour PX).

Matrice de Compatibilité entre verrous

	X	S
X	N	N
S	N	O

X : Verrou exclusif
S : Verrou partagé
N : Demande non satisfaite
O : Demande accordée

Verrou de mise à jour - noté U

Un verrou de mise à jour donne une indication sur l'intention de mise à jour d'un granule par une transaction.
Ce type de verrou est incompatible avec des verrous de mise à jour et des verrous exclusifs mais reste compatible avec des verrous partagés.
Ce type de verrou permet d'éviter des situations d'interblocage.

Protocole de mise à jour - noté PU

Toute transaction qui souhaite mettre à jour un granule doit exécuter la commande UFIND pour obtenir un verrou. Toute mise à jour du granule transforme le verrou en verrou exclusif.

Protocole de mise à jour Etendue - noté PUE

Le protocole PUE étend PU comme c'était le cas pour les protocoles précédents en conservant les verrous jusqu'à la fin de la transaction.

Matrice de Compatibilité entre verrous

	X	U	S
X	N	N	N
U	N	N	O
S	N	O	O

X : Verrou exclusif S : Verrou partagé U : Mise à jour

N : Demande non satisfaite

O : Demande accordée

Prévention des interblocages

La prévention des interblocages consiste à éviter la création d'un circuit dans le graphe d'attente en respectant un des protocoles qui rend impossible l'apparition de tels circuits :

- Wait-Die
- Wound-Wait

Protocole Wait-Die - PWD

Il garantit que les transactions en attente attendent des transactions plus jeunes :
Lorsqu'une transaction A demande un verrou déjà verrouillé par une autre transaction B alors A attend si elle est plus vieille que B, autrement A est annulée et redémarrée

Protocole Wound-Wait - P2W

Il garantit que les transactions en attente attendent des transactions plus vieilles :
Lorsqu'une transaction A demande un verrou déjà verrouillé par une autre transaction B alors A attend si elle est plus jeune que B, autrement B est annulée et redémarrée

Quatre niveaux

1: READ UNCOMMITED
2: READ COMMITTED
3: REPEATABLE READ
4: SERIALIZABLE

READ UNCOMMITED

Ce mode d'isolation correspond à l'absence du contrôle de concurrence entre transactions. Dans ce mode, il est possible de rencontrer des lectures sales, des lectures non répétables ou des tuples fantômes.

READ COMMITED

Ce mode d'isolation est adopté par défaut dans ORACLE. Il empêche la lecture des tuples qui ne sont pas encore validés. Ce mode n'empêche pas des lectures non répétables ou des tuples fantômes.

REPEATABLE READ

Ce mode d'isolation est adopté par défaut dans MYSQL.
Il garantit que l'exécution successive d'une requête dans une transaction retourne le même résultat qu'au début de la transaction.
Il empêche la lecture des tuples qui ne sont pas encore validés.
Il n'empêche pas la création de tuples fantômes.

SERIALIZABLE

Ce mode d'isolation est total. Il retourne un résultat équivalent à une exécution séquentielle des transactions. Cette garantie se fait au prix d'un blocage élevé entre les transactions.

Accès concurrents avec Oracle

- Vérrouillage au niveau d'un n-uplet ou d'une table
- Seules les données validées par des transactions sont accessibles en consultation
- Oracle ne prend pas compte des lectures non reproductibles. Pour pallier au problème de lectures non reproductibles il existe des transactions à lecture seulement (Read Only).

Accès concurrents avec Oracle

Pour le choix du niveau d'isolation, il est possible de lancer la commande SQL suivante :

- SET TRANSACTION ISOLATION LEVEL {SERALIZABLE | READ COMMITTED}
- Par défaut le niveau d'isolation est READ COMMITTED

Accès concurrents avec Oracle

Oracle effectue principalement deux opérations lors de pose d'un verrou sur un n-uplet :

- Un verrou DML (Data Manipulation Language) est posé. Ce verrou évite que d'autres transactions verrouillent le granule. Il sera relâché à la fin de la transaction;
- Un verrou DDL (Data Dictionnary Language) est posé sur la table afin d'éviter les modifications structurelles de la table. Il sera relâché à la fin de la transaction.

Accès concurrents avec Oracle

Le verrouillage peut correspondre à un des modes suivants :

- ROW SHARE
- ROW EXCLUSIVE
- SHARE
- SHARE ROW EXCLUSIVE
- EXCLUSIVE

Verrous

- ROW SHARE (RS) : accès simultané à la table en vue de modifier ses lignes et en interdisant de verrouiller la table en exclusif (X)
- ROW EXCLUSIVE (RX) : similaire à RS mais en interdisant tout accès partagé (S) ou exclusif (X ou SRX) sur la table. Ce mode est positionné automatiquement dans les cas suivants : updating, inserting, or deleting sur la table

Verrous

- SHARE (S) : accès interdit en mode exclusive (RX, SRX, X) à la table. Il autorise des verrous partagés (RS ou S) en interdisant aux transactions des opérations de mise à jour
- SHARE ROW EXCLUSIVE (SRX) : il pose un verrou exclusif sur la table en partageant ses lignes avec les autres transactions.
- EXCLUSIVE (X) : il pose un verrou exclusif sur la table en interdisant aux autres transactions toutes les opérations de mise à jour des lignes de la table

Verrous

Compatibilité	RS	RX	S	SRX	X
RS	O	O	O	O	N
RX	O	O	N	N	N
S	O	N	O	N	N
SRX	O	N	N	N	N
X	N	N	N	N	N

Quelques commandes SQL

SET TRANSACTION ISOLATION LEVEL <option>
SET TRANSACTION READ ONLY
Deux possibilités pour placer des verrous explicites sous Oracle avec:

- l'instruction select avec l'option For Update : select from employe where nuempl=20 for update
- l'instruction lock : LOCK TABLE nom-table IN mode-verrou