

Traitements des accès concurrents dans ORACLE

Bernard ESPINASSE
Professeur à l'Université d'Aix-Marseille
2010

Traitement des accès concurrents dans Oracle

Principe de base :

Dans Oracle, une transaction se termine dans un des cas suivants :

- l'exécution d'un COMMIT ou d'un ROLLBACK
- lors d'une déconnexion, la transaction courante est validée
- lorsqu'un processus utilisateur se termine anormalement, la transaction courante est annulée

Lorsqu'il n'y a aucune transaction en cours, la première instruction SQL exécutable démarre une transaction

Point de sauvegarde intermédiaire (save point) :

- Permet de diviser une transaction en plusieurs petites parties qu'il est possible d'annuler sans annuler toute la transaction
- => Notion très utile dans les transactions composées de beaucoup de commandes de mise à jour

- Syntaxe :

SAVEPOINT point;

l'annulation des opérations comprises entre le point courant et un point de sauvegarde donné se réalise par :

ROLLBACK point;

Comportement des accès concurrents par défaut avec verrous implicites

Par défaut, Oracle :

- autorise que plusieurs transactions concurrentes modifient, ajoutent ou suppriment des lignes dans une même table
- les changements réalisés par une transaction T ne sont pas visibles par une autre transaction concurrente **tant que la transaction T n'est pas validée**
- sont posés automatiquement des **verrous (verrous implicites)** afin d'assurer la cohérence des données selon les trois règles suivantes:
 - une modification bloque une autre modification
 - une lecture ne bloque pas une modification
 - une modification ne bloque pas une lecture

Soit:

- Si une transaction T2 tente de **modifier** un objet sur lequel une transaction T1 possède un accès en écriture, alors T2 est placée **en attente** jusqu'à la fin de T1
- Si une transaction T2 tente de **lire** un objet sur lequel une transaction T1 possède un accès en écriture, alors T2 a accès à une version antérieure confirmée de l'objet
- Si une transaction T2 tente de **modifier** un objet sur lequel une transaction T1 possède un accès en lecture, la transaction T2 obtient l'accès en écriture alors que T1 mémorise l'accès à sa version de l'objet

Traitement des accès concurrents par défaut avec verrous implicites : exemple 1

temps	T1	T2
t1	UPDATE eleves SET nom = 'Tintin' WHERE num_eleve = 1 ;	
t2		SELECT nom (Brisefer) FROM eleves WHERE num_eleve = 1 ;
t3	COMMIT ;	
t4		SELECT nom (Tintin) FROM eleves WHERE num_eleve = 1 ;

Bilan :

- La lecture effectuée sur la table eleves à l'instant t2 par T2 ne voit pas les modifications apportées par T1 => **pas de lecture impropre**
- Par contre, la lecture effectuée à l'instant t4, voit les modifications apportées par T1 car T1 a été validée à l'instant t3 => **lecture non reproductible**

Traitement des accès concurrents par défaut avec verrous implicites : exemple 2

temps	T1	T2
t1	UPDATE resultats SET points=points+1; WHERE num_eleve = 1 ;	
t2		UPDATE resultats SET points=points+3 WHERE num_eleve =1;
t3	COMMIT ;	
t4		COMMIT ;

attente
↓

Bilan : A l'instant t2, T2 est placée en attente jusqu'à l'instant t3 (fin de la transaction T1)

- pas de perte de mise à jour

Limites des verrous implicites

Exemples : Cohérence des lectures au niveau opération :

temps	T1	T2
t1		SELECT nom (Brisefer) FROM eleves WHERE num_eleve = 1 ;
t2	UPDATE eleves SET nom='Tintin' WHERE num_eleve = 1 ; COMMIT ;	
t3		SELECT nom (Tintin) FROM eleves WHERE num eleve = 1 ;

Bilan : Lecture non reproductible !!

=> pas de garantie de la cohérence en lecture au niveau transaction

Limites des verrous implicites

Ainsi avec les options par défaut d'Oracle (uniquement les verrous implicites) il y a **garantie de la cohérence en lecture au niveau opération** :

- pas de lecture impropre
- pas de perte de mise à jour

mais **pas de garantie de la cohérence en lecture au niveau transaction** :

- références fantômes possibles
- lectures pas nécessairement reproductibles

=> Pour réaliser des **lectures cohérentes dans une transaction**, il faut:

- soit démarrer une transaction au moyen de la **commande SET TRANSACTION READ ONLY**
- soit utiliser des **verrous explicites**

Usage de la commande « SET TRANSACTION READ ONLY »

Exemples : Cohérence des lectures au niveau opération de la transaction :

temps	T1	T2
t1		SET TRANSACTION READ ONLY ;
t2		SELECT nom (Brisefer) FROM eleves WHERE num_eleve = 1 ;
t3	UPDATE eleves SET nom='Tintin' WHERE num_eleve = 1 ; COMMIT ;	
t4		SELECT nom (Brisefer) FROM eleves WHERE num eleve = 1 ;

Bilan : Lecture reproductible

Verrouillage explicite dans Oracle

Oracle possède 2 commandes permettant de poser explicitement des verrous :

- **Verrouillage explicite des lignes répondant à la condition de clause_where**

```
SELECT clause_select
FROM clause_from
WHERE clause_where
FOR UPDATE
```

- **Verrouillage explicite défini par la requête LOCK TABLE**

```
LOCK TABLE liste_table
IN mode_verrou MODE;
```

- **5 modes de verrouillage explicite possibles (mode_verrou) :**

- **ROWSHARE**
- **ROW EXCLUSIVE**
- **SHARE**
- **SHARE ROW EXCLUSIVE**
- **EXCLUSIVE**

Modes de verrouillage explicite (1)

1 - Lignes partagées (RS : ROW SHARE) :

- Verrouillage sélectif des lignes d'une table en vue de leur modification
- Toutes les opérations et tous les autres modes de verrouillage sont autorisés, à l'exception du verrouillage exclusif de table
- Activation du mode:

```
LOCK TABLE liste_table IN ROW SHARE MODE;
```

ou

```
SELECT clause_select
FROM clause_from
WHERE clause_where
FOR UPDATE
```

2 - Lignes exclusives (RX : ROW EXCLUSIVE) :

- Verrouillage sélectif des lignes d'une table en vue de leur modification
- Lignes non partagées par d'autres transactions
- Toutes les opérations et tous les autres modes de verrouillage sont autorisés, à l'exception des modes suivants: table partagée, lignes partagées et table exclusive
- Activation du mode :

```
LOCK TABLE liste_table IN ROW EXCLUSIVE MODE; ou
```

```
INSERT INTO nom_table; ou
```

```
UPDATE nom_table; ou DELETE FROM nom_table;
```

Modes de verrouillage explicite (2)

3 - Tables partagées (S : SHARE) :

- Permet aux autres transactions d'effectuer sur cette table des sélections, des verrouillages en mode lignes partagées ou en mode table partagée => Possibilité d'avoir plusieurs transactions qui effectuent ce mode de verrouillage simultanément sur la même table
- Toutes les opérations modifiant le contenu de la table (insertion, suppression et mise à jour) ne peuvent pas être exécutées
- Impossible de verrouiller cette table en mode partage exclusif de lignes, ni en mode table exclusive
- Activation du mode : **LOCK TABLE liste_table IN SHARE MODE;**

4 - Partage exclusif de lignes (SRX : SHARE ROW EXCLUSIVE) :

- Verrouillage d'une table entière
- Permet aux autres transactions d'effectuer sur cette table des sélections, des verrouillages en mode lignes partagées
- Une seule transaction peut effectuer ce mode de verrouillage sur une table
- Toutes les opérations modifiant le contenu de la table (insertion, suppression et mise à jour) ne peuvent pas être exécutées
- Impossible de verrouiller cette table en mode partage exclusif de lignes, ni en mode table partagée, ni en mode table exclusive
- Activation du mode : **LOCK TABLE liste_table IN SHARE EXCLUSIVE MODE;**

Modes de verrouillage explicite

5 - Table exclusive (X : EXCLUSIVE) :

- Verrouillage d'une table entière
- Permet aux autres transactions d'effectuer uniquement des opérations de sélection sur cette table
- Une seule transaction peut effectuer ce mode de verrouillage sur une table
- Toutes les opérations modifiant le contenu de la table (insertion, suppression et mise à jour) ne peuvent pas être exécutées
- Impossible d'effectuer un mode de verrouillage
- Activation du mode : **LOCK TABLE liste_table IN EXCLUSIVE MODE;**

Compatibilité des modes :

Modes de verrouillage		Compatibilité avec les autres modes				
		RS	RX	S	SRX	X
1-Lignes partagées	RS	oui	oui	oui	oui	non
2-Lignes exclusives	RX	oui	oui	non	non	non
3-Table partagée	S	oui	non	oui	non	non
4-Partage exclusif de lignes	SRX	oui	non	non	non	non
5-Table exclusive	X	non	non	non	non	non

Du mode le moins restrictif au mode le plus restrictif (1 à 5)

Journaux de reprise ou de transactions

- **Fichiers dans lesquels Oracle enregistre toutes les mises à jour et transactions**
- **Utilisés pour la récupération de données suite à des défaillances inattendues (reprise après panne)**
- **Chaque base de données Oracle dispose au minimum de 2 journaux de reprise**
- **Fonctionnement des journaux de reprise :**
 - les journaux de reprise sont utilisés en boucles
 - soient 2 journaux de reprise attachés à une base de données:
 - les transactions sont enregistrées dans le journal A .
 - lorsque le journal A arrive à saturation, les nouvelles transactions sont enregistrées dans le journal B
 - lorsque le journal B est plein à son tour, le journal A est de nouveau utilisé
- **2 modes de fonctionnement des journaux :**
 - **mode ARCHIVELOG** : conservation de tous les journaux de reprise (protection contre tous les types de défaillances, d'une erreur d'utilisateur à un plantage de disque)
 - **mode NOARCHIVELOG**: pas de conservation des anciens journaux de reprise (protection limitée: pannes de courant)