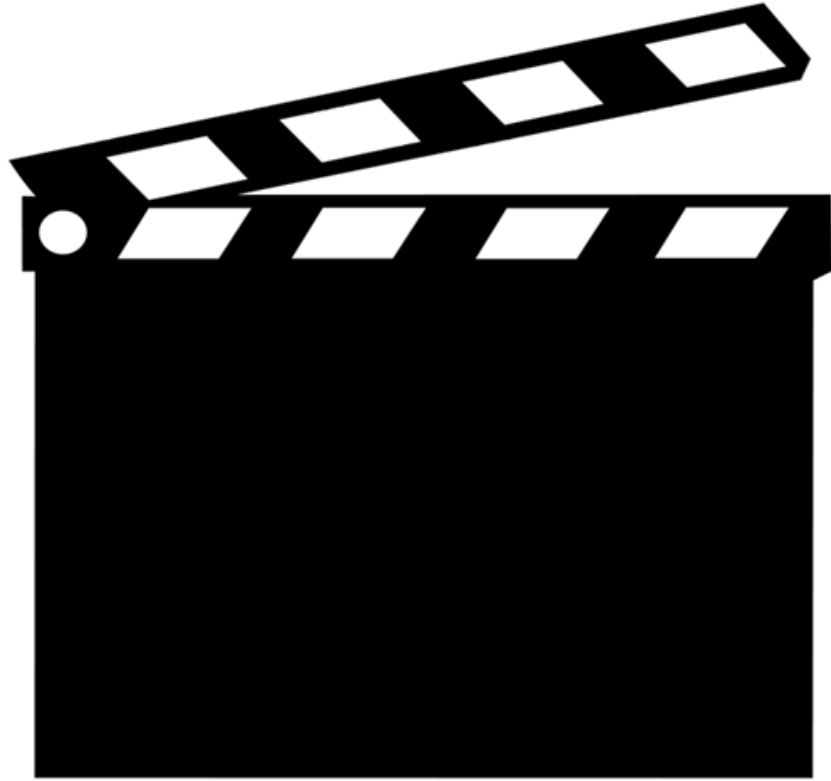


C018SA-W1-S7



SEMAINE 1 : Transactions et concurrence

1. Introduction : les transactions
2. Les problèmes
3. Sériabilité
4. Estampillage
5. Verrouillage à 2 phases
6. Degrés d'isolation dans les SGBD
7. **Verrouillage hiérarchique**

Verrouiller : oui mais quoi ?

- Verrouiller coût cher : $O(n)$ où n est le nombre de nuplets de la table
- **Idée** : verrouiller juste la table $\rightarrow O(1)$

- **Avantage** : Adapter la taille du verrou au nombre de transactions
 - Granule gros \rightarrow peu de verrous, temps d'attente long
 - Granule fin \rightarrow beaucoup de verrous, peu de temps d'attente

 - Le bon compromis dépend du nombre de transactions en cours
 - Deux niveaux de verrous : la table et le nuplet

Se généralise en **verrouillage hiérarchique**

Verrouiller : oui mais quoi ?

- Verrouiller coût cher : $O(n)$ où n est le nombre de nuplets de la table
- **Idée** : verrouiller juste la table $\rightarrow O(1)$

- **Avantage** : Adapter la taille du verrou au nombre de transactions
 - Granule gros \rightarrow peu de verrous, temps d'attente long
 - Granule fin \rightarrow beaucoup de verrous, peu de temps d'attente

 - Le bon compromis dépend du nombre de transactions en cours
 - Deux niveaux de verrous : la table et le nuplet

Se généralise en **verrouillage hiérarchique**

Verrou table vs verrou nuplet

Même comportement
des verrous :

	R(A)	W(A)
R(A)	NON	OUI
W(A)	OUI	OUI

- 1) verrous ligne
- 2) verrou table

EMP	Verrou : X		
NE	NOM	SAL	Verrou ligne
0	Charlie	2000	X
1	Diana	2200	X
2	Eric	1700	X

UPDATE EMP SET SAL = SAL + 100

Verrou table vs verrou nuplet

Même comportement
des verrous :

	R(A)	W(A)
R(A)	NON	OUI
W(A)	OUI	OUI

- 1) verrous ligne
- 2) verrou table

EMP	Verrou : X		
NE	NOM	SAL	Verrou ligne
0	Charlie	2000	X
1	Diana	2200	X
2	Eric	1600	

T₁: UPDATE EMP SET SAL = SAL + 100
WHERE NE = 0

T₂: UPDATE EMP SET SAL = SAL + 100
WHERE NE = 1 ← **EN ATTENTE !!**

Comment fluidifier les transactions ?

- **Problème :**

Poser juste des verrous X et S risque de bloquer trop de transactions

- **Idée :**

Créer des nouveaux types de verrous *au niveau table* mais annonçant des changements sur les lignes : RS et RX

ROW SHARE
ROW EXCLUSIVE

Verrous de type ROW-S et ROW-X

- Un verrou de type RS ou RX pose un verrou au niveau table ET au niveau ligne (de type S ou X)
- La décision **NON** est prise au niveau table → rapide
- La décision **OUI** indique que la décision doit être validée au niveau ligne → pareil qu'avant

Verrou demandé sur table \ Verrou détenu sur table	RS	RX	S	X
RS	oui	oui	oui	non
RX	oui	oui	non	non
S	oui	non	oui	non
X	non	non	non	non

Verrou table RS / RX

1) verrous ligne

2) verrou table RX

EMP	Verrou : RX₁ RX₂		
NE	NOM	SAL	Verrou ligne
0	Charlie	2000	X
1	Diana	2200	X
2	Eric	1600	

T₁: UPDATE EMP SET SAL = SAL + 100
WHERE NE = 0

T₂: UPDATE EMP SET SAL = SAL + 100
WHERE NE = 1

ROW SHARE MODE (RS)

Explicite : **LOCK TABLE** *table* **IN ROW SHARE MODE;**

- **Permet** : toutes les lectures de la table, et des modifications sur ce qu'on ne compte pas lire dans la table. (tout sauf X)
- **Utilité** : on annonce qu'on va lire certaines données de la table. On ne bloque rien à part des personnes qui souhaiteraient également modifier ces mêmes données.

ROW EXCLUSIVE MODE (RX)

Explicite : `LOCK TABLE table IN ROW EXCLUSIVE MODE;`

Implicite :

`INSERT INTO table ... ;`

`UPDATE table ... ;`

`DELETE FROM table ... ;`

`SELECT ... FROM table ... FOR UPDATE ;`

- **Permet** : RS, RX
- **Utilité** : permet d'annoncer qu'on a apporté une modification à la table → empêche les verrous S et X sur les nuplets en question

SHARE MODE (S)

Explicite : `LOCK TABLE table IN SHARE MODE;`

- **Permet** : RS, S (toutes les lectures)
- **Utilité** : on veut que personne ne modifie la table sur laquelle on lit. Par contre on ne souhaite pas modifier la table.
On ne bloque aucune lecture.

EXCLUSIVE MODE (X)

Explicite : `LOCK TABLE table IN EXCLUSIVE MODE;`

- **Permet** : rien !
- **Utilité** : bloque toutes les autres opérations sur la table.
Utile par exemple pour une mise à jour globale de la table.

SHARE ROW EXCLUSIVE MODE (SRX)

Explicite : `LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE;`

- **Permet** : RS (uniquement une lecture sur ce qu'on n'a pas modifié)
- **Utilité** : permet de faire S et RX en même temps : on bloque les modifications des autres sur toute la table *ET* on annonce une modification sur certaines lignes.

Moins permissif que S

Conclusion sur le verrouillage hiérarchique

Pour obtenir	Il faut avoir sur tous les ancêtres
RS ou S	RS ou RX (avec S on l'a déjà)
RX, SRX ou X	RX ou SRX (avec X on l'a déjà)

Hierarchie « BD »

base



table



tuple

Verrou demandé \ Verrou détenu	RS	RX	S	SRX	X
RS	oui	oui	oui	oui	non
RX	oui	oui	non	non	non
S	oui	non	oui	non	non
SRX	oui	non	non	non	non
X	non	non	non	non	non

CONCLUSION SUR LE VERROUILLAGE EN GENERAL

Approche pessimiste de la concurrence

- Prévient les conflits
- Assez coûteuse et assez complexe

Avantages du verrouillage

- Abandon des seules transactions rendant l'exécution non sérialisable
- Les performances sont bonnes : isolation paramétrable, taille variable du granule