

Verrous

BTS SIO SLAM 3 - 2ème année



Exemple

T1 et T2 modifient simultanément A

T_1	T_2	BD
		$A = 10$
read A		
	read A	
$A = A + 10$		
write A		$A = 20$
	$A = A + 50$	
	write A	$A = 60$

Les modifications effectuées par T1 sont perdues

Problèmes

- On suppose le fichier partagé sur le réseau
- Si une seule personne travaille: pas de prb
- Si deux personnes travaillent sur ce fichier -> prb
- Surtout en écriture !
- Le prb est celui des accès concurrents

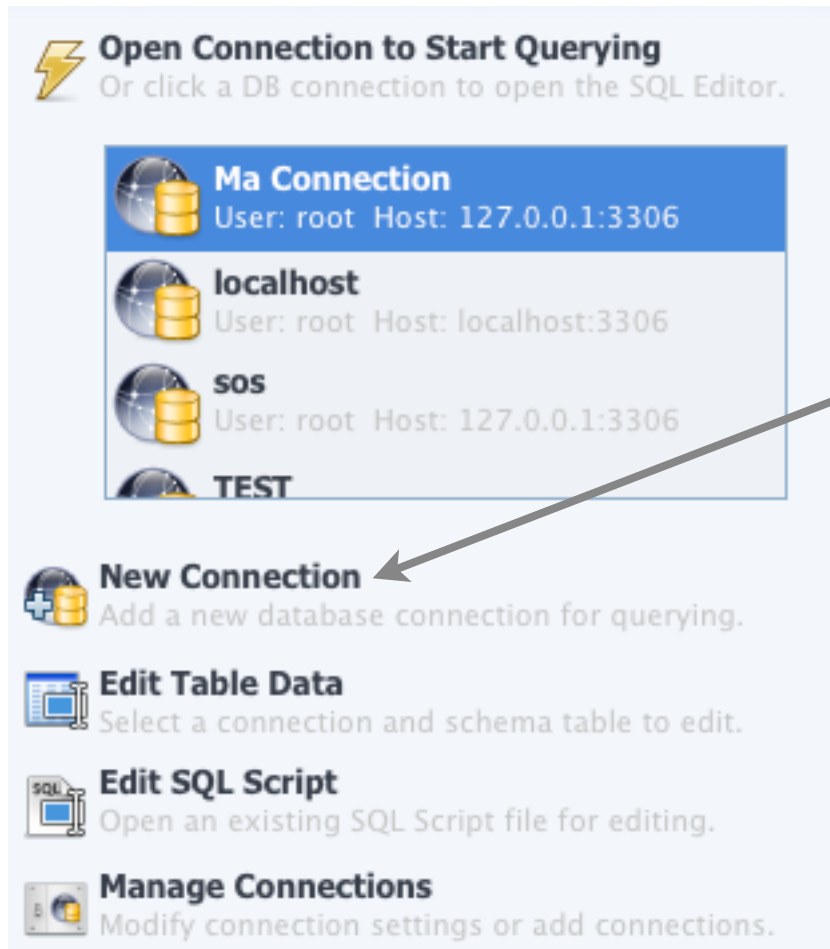
Problème

- Problème de chevauchement
- De cohérence
- Une modification peut ne pas être prise en compte
- Perte de données

Transposition à MySQL

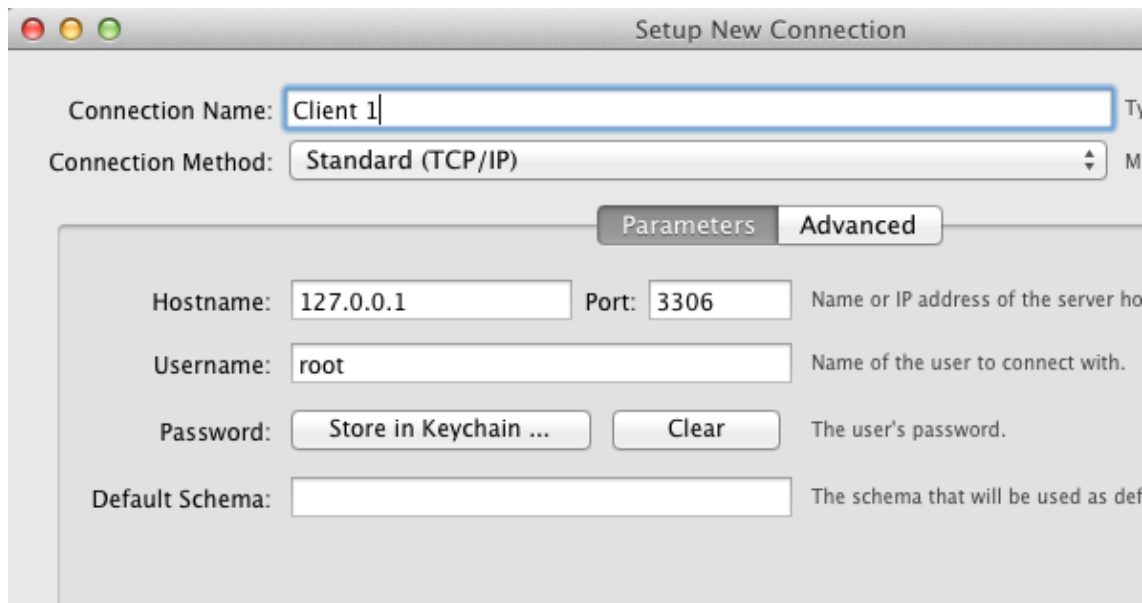
- MySQL est multithreadé
- Il peut répondre à plusieurs clients simultanément
- Il peut donc y avoir des accès concurrents
- Besoin de stratégies pour éviter les corruptions de données

Plusieurs Clients



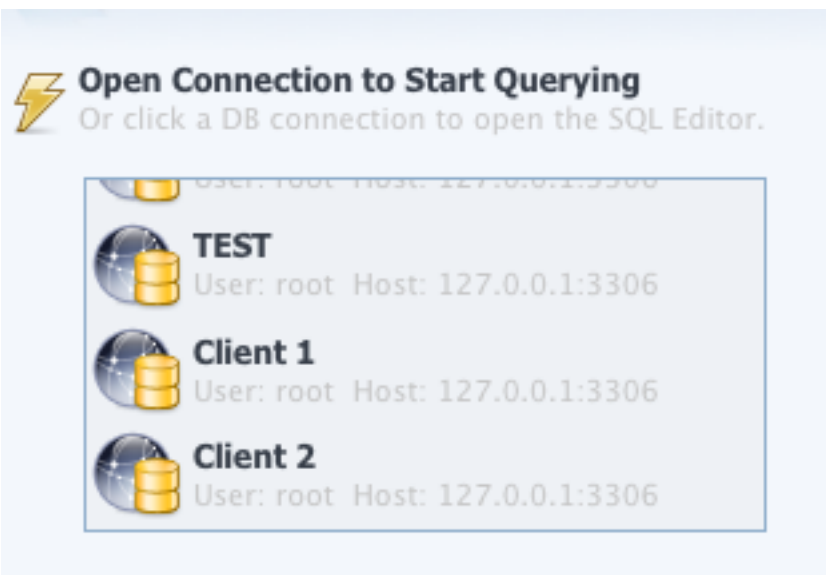
On va créer deux connexions afin de simuler deux clients et tester les accès concurrents

Plusieurs Clients



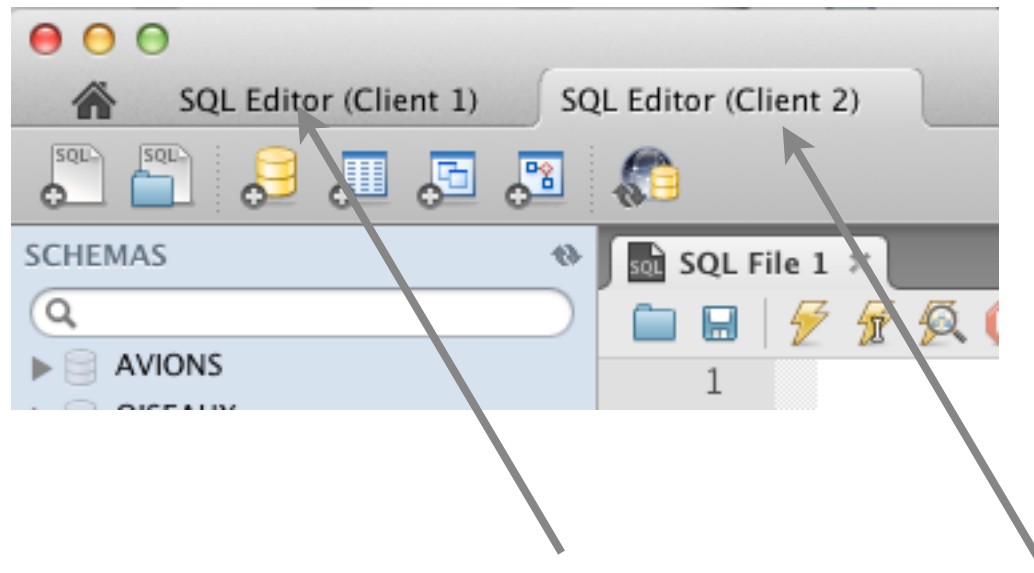
The screenshot shows a 'Setup New Connection' dialog box. The 'Connection Name' field is set to 'Client 1'. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing fields for 'Hostname' (127.0.0.1), 'Port' (3306), 'Username' (root), 'Password' (with 'Store in Keychain ...' and 'Clear' buttons), and 'Default Schema'. Descriptive text for each field is provided on the right.

Rien à faire de particulier
On nomme Client 1 le
premier et Client 2 le second



On a maintenant deux accès à la
base de données. Il suffit de les
activer en cliquant

Plusieurs Clients



Solution simple

- Solution simple : verrouiller le fichier dès qu'un utilisateur l'ouvre (plus d'accès simultané)
- Problème : le temps d'attente pour les autres...

Solution améliorée

- On va verrouiller mais en tenant compte des actions
- Il y a deux actions : lecture, écriture
- Il n'est pas gênant d'avoir plusieurs personnes qui lisent simultanément.
- Par contre en cas de modification il faut s'assurer que personne n'utilise ce fichier en écriture...

Notion de verrou

- Un verrou bloque l'accès à une table ou à des lignes d'une table.
- Un verrou se pose et se retire.
- Il n'est donc actif que durant cette période

Verrou en lecture

(verrou partagé)

- Plusieurs clients peuvent accéder simultanément en lecture.
- Aucun en écriture (même celui qui a posé le verrou)

Verrou en écriture

(verrou exclusif)

- Seul le client ayant posé le verrou peut accéder à la ressource
- En lecture et en écriture !
- **Exception** : InnoDB permet d'accéder en lecture à des lignes protégées par un verrou exclusif.

Verrou suite...

- Si le verrou ne peut être mis immédiatement -> file d'attente (Le client doit attendre)
- Seul le client ayant posé le verrou a la possibilité de le lever
- En cas de déconnexion d'un client, le serveur lève automatiquement tous les verrous posés par ce client.
- En cas d'arrêt du serveur tous les verrous sont relâchés.

Verrou gestion

- C'est le SGBDR qui gère implicitement et de manière interne la pose et la levée des verrous
- Mais en cas de besoin, l'utilisateur pourra poser explicitement des verrous.

Granularité des verrous

- On a deux granularités :
 - sur la table
 - sur quelques lignes de la table.
- Verrou sur la table :
 - Plus simple à mettre en place
 - Mais moins performants
- Verrou sur les lignes
 - Plus de travail pour le serveur
 - Accès concurrents plus performants

Verrou et Moteur

- MyISAM et MEMORY verrouillage sur les tables
- InnoDB verrouillage sur les lignes

Verrous implicites

- SELECT, INSERT, UPDATE (granularité dépend du moteur)
- REPAIR TABLE, OPTIMIZE TABLE, ALTER TABLE -> sur la table

Poser/enlever un verrou

- LOCK TABLES
- READ pose un verrou partagé
- WRITE pose un verrou exclusif
- Exemple LOCK TABLE t1 READ, t2 READ, t3 WRITE;

Poser/enlever un verrou

- UNLOCK TABLES : déverrouille l'ensemble des tables
- LOCK TABLES déverrouille toutes les tables précédemment verrouillé (il faut donc tout verrouiller en une seule fois)

Verrouillage explicite InnoDB

- Pour indiquer au moteur qu'il faut verrouiller les lignes lues :
`SELECT ... LOCK IN SHARE MODE`
- `SELECT ... FOR UPDATE` pose un verrou exclusif, empêchant toutes les lectures faites avec un `SELECT ... LOCK IN SHARE MODE`
- Ces modificateurs n'ont de sens qu'à l'intérieur d'une transaction (`START TRANSACTION`)

Intérêt du verrouillage explicite

- Si vous avez un moteur non transactionnel
- Avec MyISAM mise à jour des index en une fois grâce au verrouillage

Connaître les verrous

- show open tables from VERROU;

Filter:		Export: 	
Database	Table	In_use	Name_locked
▶ VERROU	Vente	1	0
VERROU	Personne	0	0

- Dans In_use vous avez le nombre de verrou