

# Rendez-vous Firebird RMML 2007

- Les Transactions

**Pierre Yager**

CRISALID

[pierre@crisalid.com](mailto:pierre@crisalid.com)



# Les Transactions

- Concepts
- Métriques : ACID
- SQL
- Isolement
- Côté Serveur
- Côté Client
- Conseils, Questions Ouvertes



# Concepts

- SGBD : Système de Gestion de Bases de Données...
- SGBD-"R" : ... Relationnelles !
- Client-Serveur : Gestion de la concurrence
- Verrouillage Pessimiste vs Optimiste
- Firebird : MGA
- Tout est transaction !



# ACID

*Theo Haërder & Andreas Reuter (1983)*

## Métriques de l'efficacité d'un système transactionnel

- **A**tomicité : Tout ou Rien
- **C**ohérence : Le système reste stable
- **I**solement : Perméabilité du système aux différentes transactions actives
- **D**urabilité : Persistance des informations



# SQL

- Norme fonctionnelle : pas de proposition ou d'exigences sur l'implémentation
- Décrit le concept de transaction d'après les "phénomènes" qu'elles autorisent
- Pas représentatif de l'Architecture Multi- Générationnelle de Firebird



# SQL // ACID

- Les "phénomènes" dépendent du niveau d'Isolation des Transactions
- L'Isolation est la seule variable d'ajustement manipulable par le développeur



# Isolement // SQL

- SQL décrit 4 niveaux d'isolement
  - READ UNCOMMITTED
  - READ COMMITTED
  - REPEATABLE READ
  - SERIALIZABLE
- Deux niveaux supplémentaires par *Berenson/Bernstein*
  - CURSOR STABILITY
  - SNAPSHOT



# Isolement // Firebird

- Firebird propose 3 niveaux d'isolement des transactions
  - READ COMMITTED
  - SNAPSHOT (CONCURENCY)
  - TABLE STABILITY (CONSISTENCY)
- On peut les affiner à l'aide des blocs de paramétrage des transactions (TPB) pour reproduire ou simuler les niveaux d'isolement normalisés





# Phénomènes

- *P0 "Dirty reads"*

Une transaction T1 modifie un enregistrement. Une autre transaction T2 modifie aussi cet enregistrement avant que T1 n'ait exécuté un COMMIT ou un ROLLBACK. Si T1 ou T2 exécute un ROLLBACK, on ne sait pas quelle devrait être la valeur de l'enregistrement.



# Phénomènes

- *P1 "Dirty writes"*

Une transaction T1 modifie un enregistrement. Une transaction T2 lit le même enregistrement avant que T1 n'ait exécuté un COMMIT ou un ROLLBACK. Si T1 effectue un ROLLBACK, T2 aura lu un enregistrement qui n'a jamais été validé et qui, par conséquent, peut être considéré comme n'ayant jamais existé.



# Phénomènes

- *P2 "Non Repetable Reads"*

Une transaction T1 lit un enregistrement. Une transaction T2 modifie ou supprime cet enregistrement et effectue un COMMIT. Si T1 essaye de relire l'enregistrement, il reçoit l'enregistrement modifié ou découvre que l'enregistrement a été supprimé.



# Phénomènes

- *P3 "Fantômes"*

Une transaction T1 lit un ensemble d'enregistrements N qui satisfont la condition <condition de recherche>. Une transaction T2 exécute des opérations SQL qui vont générer un ou plusieurs enregistrements qui satisfont la même <condition de recherche> utilisée par T1. Si T1 répète la lecture initiale avec la même <condition de recherche>, elle obtient une collection d'enregistrements différente.



# Phénomènes

- *P4 "Lost Updates"*

Une transaction T1 lit un enregistrement. Une Transaction T2 modifie le même enregistrement que T1 a lu (d'après, par exemple, une lecture qu'elle aurait fait avant). T1 modifie l'enregistrement et exécute un COMMIT. A ce moment, même si T2 exécute un COMMIT, la mise à jour effectuée par T2 est perdue.



# Isolement // Phénomènes

	<b>P0</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
<b>READ UNCOMMITTED</b>	NON	OUI	OUI	OUI	OUI
<b>READ COMMITTED</b>	NON	NON	OUI	OUI	OUI
<b>CURSOR STABILITY</b>	NON	NON	O/N	OUI	O/N
<b>REPETABLE READ</b>	NON	NON	NON	OUI	NON
<b>SNAPSHOT</b>	NON	NON	NON	O/N	NON
<b>SERIALIZABLE</b>	NON	NON	NON	NON	NON



# Firebird / ISQL

```
SET TRANSACTION [NAME <Transaction Name> ]  
  [ READ WRITE ] | [ READ ONLY ]  
  [ WAIT [ LOCK TIMEOUT n ] ] | [ NO WAIT ]  
  [ NO AUTO UNDO ]  
  [ ISOLATION LEVEL ] { SNAPSHOT [ TABLE STABILITY ]  
    | READ COMMITTED [ [ NO ] RECORD VERSION ] }  
  [ RESERVING <reserving-clause> | USING <db-handle> [, db-  
handle ... ] ];
```

```
<reserving-clause ::= <table> [, <table> ... ]  
  [ FOR [ SHARED | PROTECTED ] { READ | WRITE } ]  
  [, <reserving-clause> [, <reserving-clause> ... ] ]
```



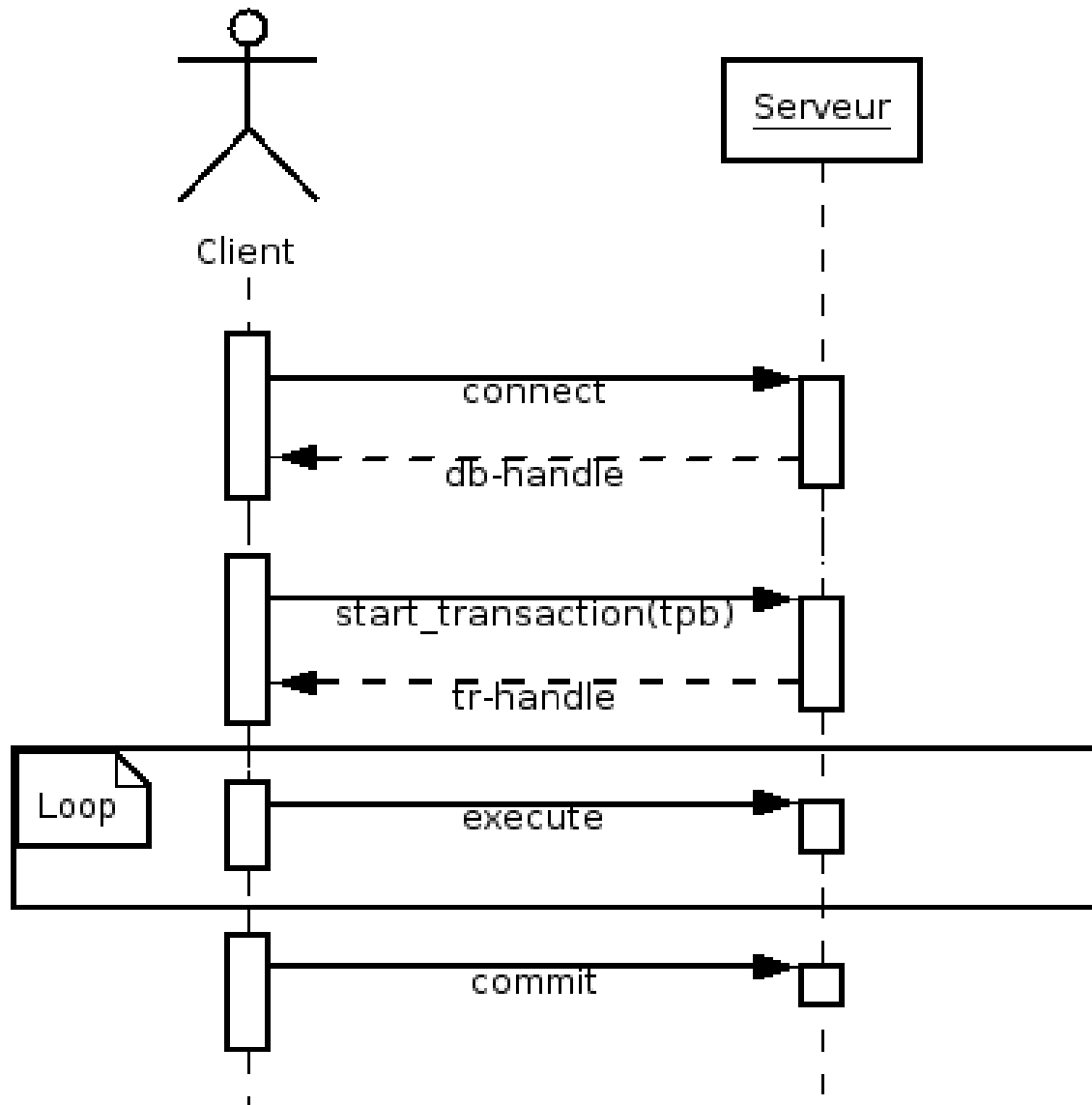
# Firebird // TPB

- Mode d'accès
  - isc\_tpb\_read
  - isc\_tpb\_write
- Résolution des blocages
  - isc\_tpb\_wait
  - isc\_tpb\_nowait
  - isc\_tpb\_lock\_timeout
- Accès aux enregistrements
  - isc\_tpb\_rec\_version
  - isc\_tpb\_no\_rec\_version
- Niveau d'Isolement
  - isc\_tpb\_read\_committed
  - isc\_tpb\_concurrency
  - isc\_tpb\_consistency
- Réserveation explicite
  - isc\_tpb\_shared
  - isc\_tpb\_protected
  - isc\_tpb\_lock\_read
  - isc\_tpb\_lock\_write
- Options
  - isc\_tpb\_no\_auto\_undo





# Séquence



# Et à la fin ?

- COMMIT (Hard-Commit) rend les modifications persistantes. Au retour de COMMIT le client a la garantie que les propriétés ACID sont respectées.
- ROLLBACK annule toutes les modifications.  
N'échoue JAMAIS !

*COMMIT et ROLLBACK libèrent toutes les ressources liées à la transaction*

- COMMIT RETAINING (Soft-Commit) rend les modifications persistantes et garde le contexte de la transaction actif.



# MGA

- Conserve toutes les versions *intéressantes* des enregistrements.
- Transactions READ COMMITTED accèdent au TSP (Transaction State Bitmap) global.
- SNAPSHOT [TABLE STABILITY] disposent d'une copie du TSP jusqu'à ce qu'elles soient terminées.
- Le Garbage Collector rend disponible l'espace utilisé par les versions des enregistrements qui ne sont plus intéressantes.
- Le Sweep fait le ménage dans les versions issues de transactions annulées et permet à GC de faire son travail.



# Interesting ?

- Les versions des enregistrements et les transactions intéressantes sont toutes celles qui n'ont pas été Hard-Committed (COMMIT) :
  - Actives
  - Limbo
  - Dead
  - Annulées (ROLLBACK)
  - Soft-Commitées (COMMIT RETAINING)
- Pas de Garbage Collector



# Cas du ROLLBACK

- Si le journal d'annulation automatique (Auto Undo Log) est actif, les modifications sont "annulées" et la transaction est COMMITED !
- Sinon la transaction est marquée ROLLED-BACK et il faut attendre que le SWEEP décide quelles versions des enregistrements sont (ou pas) intéressante pour que le Garbage Collector fasse le ménage



# En cas de problème

- En cas de violation des contraintes d'intégrité référentielles et structurelles (domaine de définition des types de données, check, clés étrangères, index uniques...) une exception est levée mais le serveur ne ROLLBACK rien par lui-même, c'est au développeur de décider quoi faire. Parfois l'erreur est récupérable et la transaction peut-être COMMITED



# OIT, OAT, NT et le "GAP"

- **O**ldest **I**nteresting **T**ransaction est la plus ancienne transaction qui n'a pas été hard-committed
- **O**ldest **A**ctive **T**ransaction est la plus ancienne transaction active (pas hard-committed, pas rolled-back, pas limbo)
- **N**ext **T**ransaction est le numéro de la prochaine transaction
- *Gap* : différence entre OIT et OAT et entre OAT et NT



# Lecture Seule

- Les transactions au niveau d'isolement **READ COMMITTED** en lecture seule : **READ ONLY** ne "collent pas" l'OIT et l'OAT et n'interfèrent pas dans les mécanismes d'auto-nettoyage du système
- Ne pas confondre avec une transaction **READ WRITE**, même *READ COMMITTED* qui ne ferait qu'un "Simple SELECT" !





# Comparaisons

	Firebird	MySQL	PostgreSQL	Oracle	SQL Server	SQLite
Transactions	Oui	InnoDB	Oui	Oui	Oui	Non
ROLLBACK partiels	Oui	InnoDB	Oui	Oui	Oui	Non
Réservations	Oui	InnoDB	Oui	Oui	Oui	Non
Détection et gestion des verrous	Oui	Oui	Oui	Oui	Oui	Non
Choix du niveau d'isolement	Oui	InnoDB	Oui	Oui	Oui	Non
Transactions Distribuées	Oui	Non	8.1	Oui	Oui	Non



# Côté Serveur

- PSQL (Procédures Stockées, Triggers) permet d'écrire du code exécuté au sein du serveur.
- Pas de gestion explicite des transactions (BEGIN/COMMIT/ROLLBACK)
- Les SAVEPOINTS internes et les EXCEPTIONS permettent de gérer les problèmes.



# Côté Client

- Composants, librairies ou connecteurs libres pour Firebird :
  - UIB (Delphi, Lazarus)
  - IBPP (C++)
  - JayBird (Java) \*
  - Firebird .NET Provider (.NET, Mono) \*
  - KinterbasDB (Python)
  - FireRuby (Ruby, Ruby On Rails)
- JayBird et Fb.NET ne sont pas des wrappers de l'API, ils réimplémentent les protocoles au niveau de TCP



# Connecteurs de Haut Niveau

- Borland Database Engine (BDE)
- CodeGear/Borland DBExpress
- ODBC
- PHP (PEAR)
- Ruby (ActiveRecord)
- Ne permettent pas la gestion fine des transactions (pas de début et de fin explicites)
- Ne supportent pas toutes les fonctionnalités de Firebird : Nivellement par le bas (SQLite, MySQL)



# Conseils

- Attention aux transactions longues (Long Running Transactions) : READ COMMITTED READ ONLY + Commit Retaining
- Réservations Explicites souvent inutiles. Peuvent cependant servir pour simuler SERIALIZABLE.
- De manière générale : transactions courtes et TOUJOURS terminées (COMMIT ou ROLLBACK) *explicitement* : ne laissez pas un composant décider pour vous !



# Maintenance

- Utiliser les outils fournis avec Firebird :
  - GBAK : Backup / Restore
  - GFIX : Paramétrage et Maintenance
  - GSTAT / ISQL : Informations sur la base de données
- GSTAT (-h) et ISQL (SHOW DATABASE) permettent de surveiller le "GAP"
- GFIX permet de démarrer SWEEP manuellement et de configurer l'intervalle de SWEEP automatique
- GBAK par un backup quotidien force le serveur à visiter toutes les pages ce qui facilite le travail du Garbage Collector



# Stats

```
$ isql
Use CONNECT or CREATE DATABASE to specify a database
SQL> connect 'myserver:db_alias'
CON> user 'SYSDBA' password 'masterkey';

Database: 'myserver:db_alias', User: SYSDBA

SQL> show database;

Database: db_alias
Owner: SYSDBA
PAGE_SIZE 4096
Number of DB pages allocated = 6551
Sweep interval = 20000
Forced Writes are ON

Transaction - oldest = 624816
Transaction - oldest active = 624817
Transaction - oldest snapshot = 624817
Transaction - Next = 624820

Default Character set: NONE

SQL> exit;
```

```
$ gstat -h db_alias
Database "db_alias"
Database header page information:
  Flags                                0
  Checksum                             12345
  Generation                           624847
  Page size                             4096
  ODS version                           11.0
  Oldest transaction                    624816
  Oldest active                         624817
  Oldest snapshot                       624817
  Next transaction                      624840
  Bumped transaction                    1
  Sequence number                       0
  Next attachment ID                   0
  Implementation ID                    16
  Shadow count                          0
  Page buffers                          2048
  Next header page                      0
  Database dialect                      3
  Creation date                         Dec 1, 2006 14:39:42
  Attributes                            force write

Variable header data:
  Sweep interval:                       20000
*END*
```



# Pas bon !

```
$ isql
```

```
Use CONNECT or CREATE DATABASE to specify a database
```

```
SQL> connect 'myserver:bad_database_alias'
```

```
CON> user 'SYSDBA' password 'zebulon';
```

```
Database: 'CHOASPES:D:\Firebird\Calepin2002\Calepin.fdb', User: SYSDBA
```

```
SQL> show database;
```

```
Database: CHOASPES:D:\Firebird\Calepin2002\Calepin.fdb
```

```
Owner: SYSDBA
```

```
PAGE_SIZE 2048
```

```
Number of DB pages allocated = 8738
```

```
Sweep interval = 20000
```

```
Forced Writes are ON
```

```
Transaction - oldest = 45976018
```

```
Transaction - oldest active = 45976019
```

```
Transaction - oldest snapshot = 45976019
```

```
Transaction - Next = 46013588
```

```
Default Character set: ISO8859_1
```

Gap NT-OAT = 37569 !





# COMMIT !

En un mot : *Merci* ! Et maintenant à vous...

- Questions ?
- Commentaires ?
- Remarques ?

**Pierre Yager**  
CRISALID  
pierre@crisalid.com



# WEBLIOGRAPHIE

- The Firebird Book (Apress, 2004, Helen Borrie & IBPhoenix)
- Firebird 2 Supplement to The Firebird Book (IBPhoenix, 2007, Helen Borrie)
- L'isolement des transactions dans Firebird (Pierre Yager, 2005)  
[http://www.ibphoenix.fr/article.php3?id\\_article=39](http://www.ibphoenix.fr/article.php3?id_article=39)
- Critique des niveaux d'isolement dans la norme SQL ANSI (Berenson/Bernstein, 1995)  
<http://www.cs.duke.edu/~junyang/courses/cps216-2003-spring/pape>
- Inside Savepoints (Dmitry Yemanov, 2005)  
<http://ibdeveloper.com/issues/issue-1-sep-1-2005/inside-savepoints/>

