Dernière mise-à-jour : 2020/02/21 07:40

LRF135 - Gestion du Serveur MySQL sous CentOS 6

Présentation, Installation et Configuration

Présentation de MySQL

MySQL comprend les outils suivants :

- **Un Serveur SQL**
  - Un moteur qui permet d'accéder aux bases de données.
- **Les programmes clients pour accéder aux serveurs**
  - Un programme interactif permet de saisir directement les requêtes et d'afficher les résultats
  - Plusieurs utilitaires et outils d'administration vous permettent de gérer votre site.
- **Une bibliothèque client pour écrire vos propres programmes**
  - Les programmes clients peuvent être écrits en C, car la bibliothèque est elle-même écrite en C.
  - Elle intègre également toutes les bases nécessaires pour supporter d'autres langages.

Développement

Le développement de MySQL depuis son rachat par la société Oracle en 2009 produit des versions différentes selon le cycle suivant :

- Création de une ou plusieurs versions successives dites **DMR** (*Development Milestone Release*) destinée(s) à être testée(s) par des personnes intéressées par des nouvelles fonctionnalités,
- La sortie d'une ou de plusieurs versions successives dites **RC** (*Release Candidate*) qui sont des versions qui deviendront stables,
- La sortie d'une version dite **GA** (*Generally Available*) qui correspond à la version stable actuelle.

A noter que MySQL existe en version 32 bits et 64 bits.

MySQL existe en deux versions, la version communautaire qui est disponible sous la licence **GNU GPL v2** et la version **Enterprise** qui :

- bénéficie du support d'Oracle,
- contient des outils tels Enterprise Monitor, Query Analyzer et Enterprise Backup.

A noter qu'à un instant t, la version communautaire et la version Entreprise sont indentique.

Ce cours se concentre sur la version communautaire.

Protocoles de Communication

MySQL propose 4 protocoles de communication :

Protocole	Connexion	OS	Commentaire
TCP	Local et distant	Unix, Windows	Seul protocole de connexion à distance
Socket Unix	Local	Unix	Protocole par défaut pour les connexions locales
Shared Memory	Local	Windows	Zone de mémoire partagée entre le serveur et le client
Named Pipes	Local	Windows	Fichier permettant deux processus sans lien de parenté de communiquer

Le port TCP par défaut de MySQL est le **3306**. Il est possible de changer le port de communication de MySQL en éditant le fichier my.cnf.
Pour se connecter sur un serveur distant il convient d'utiliser la commande suivante **mysql -u root -p -h nom_serveur –protocol=tcp**.

Architecture

L'architecture est la suivante :

- Le serveur reçoit une requête du client,
- Le serveur regarde dans le cache des requêtes, si activé,
- Si la requête s'y trouve, le serveur renvoie le résultat du requête stocké dans le cache,
- Si la requête ne s'y trouve pas, le serveur analyse et optimise la requête avant de l'exécuter en utilisant le **moteur d'exécution des requêtes**, le met dans le cache et retourne le résultat au client.

Utilisation du Disque

Les bases de données de MySQL, aussi appelées **Schémas** sont représentées sur disque par un répertoire du même nom que la base de données.

Ce répertoire existe dans un répertoire de données appelé le **datadir**.

Pour connaître l'emplacement du datadir, il convient d'utiliser la commande suivante sous **Red Hat** et **CentOS** :

```
mysql> SHOW VARIABLES LIKE 'datadir';
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| datadir       | /var/lib/mysql/ |
+---------------+-----------------+
1 row in set (0.00 sec)

mysql>
```

Sous **Windows™** :

```
mysql> SHOW VARIABLES LIKE 'datadir';
+---------------+-------------------------------------------------+
```

```
| Variable_name | Value                                          |
+---------------+------------------------------------------------+
| datadir       | C:\ProgramData\MySQL\MySQL Server 5.6\Data\ |
+---------------+------------------------------------------------+
1 row in set (0.08 sec)

mysql>
```

L'examen de ce répertoire montre la présence des répertoires des schémas.

Sous **Red Hat** et **CentOS** :

```
[root@centos6 ~]# ls -l /var/lib/mysql
total 28684
-rw-rw----. 1 mysql mysql 18874368 22 janv. 15:28 ibdata1
-rw-rw----. 1 mysql mysql  5242880 22 janv. 15:28 ib_logfile0
-rw-rw----. 1 mysql mysql  5242880 22 janv. 15:28 ib_logfile1
drwx------. 2 mysql mysql     4096 22 janv. 15:28 mysql
srwxrwxrwx. 1 mysql mysql        0 22 janv. 15:28 mysql.sock
drwx------. 2 mysql mysql     4096 22 janv. 15:28 performance_schema
drwx------. 2 mysql mysql     4096 22 janv. 15:28 test
```

Sous **Windows™** :

```
C:\Users\Administrateur>dir "c:\ProgramData\MySQL\MySQL Server 5.6\Data"
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 7C10-520B

 Répertoire de c:\ProgramData\MySQL\MySQL Server 5.6\Data

22/02/2016  11:09    <REP>          .
22/02/2016  11:09    <REP>          ..
21/02/2016  15:52                56 auto.cnf
22/02/2016  11:09        12 582 912 ibdata1
```

```
22/02/2016  11:09        50 331 648 ib_logfile0
21/02/2016  15:52        50 331 648 ib_logfile1
21/02/2016  15:51    <REP>          mysql
21/02/2016  15:51    <REP>          performance_schema
21/02/2016  15:53               201 SERVER-slow.log
22/02/2016  11:09             7 608 SERVER.err
22/02/2016  11:09                 5 server.pid
21/02/2016  15:50    <REP>          test
               7 fichier(s)      113 254 078 octets
               5 Rép(s)  15 815 516 160 octets libres

C:\Users\Administrateur>
```

Il est possible de configurer MySQL pour utiliser plusieurs supports différents pour stocker les journaux, fichiers temporaires, binaires etc.

Utilisation de la Mémoire

L'allocation de la mémoire est de deux types différentes :

Allocation par Instance

C'est la mémoire allouée par le serveur au démarrage de celui-ci. Elle est partagée par le serveur **mysqld** et les **threads** (connexions). Elle comprend, entre autre :

- le *key_buffer_size* : le cache d'index des tables MyISAM,
- le *innodb_buffer_pool_size* : le cache de données InnoDB,
- le *table_definition_cache* et le *table_open_cache* : le cache des tables,
- le *query_cache* : le cache des requêtes.

Allocation par Threads

C'est la mémoire allouée d'une manière dynamique en fonction des besoins de chaque client. Elle comprend, entre autre :

- le *sort_buffer_size* : le buffer de tri,
- le *read_buffer_size* : le buffer de lecture,
- le *tmp_table_size* : l'espace mémoire des tables temporaires,
- le *max_heap_table_size* : l'espace mémoire des tables Memory.

Installation

Red Hat et CentOS

Sous **Red Hat** et **CentOS**, pour installer mysql, utilisez yum :

```
[root@centos6 ~]# yum install mysql-server
Loaded plugins: fastestmirror, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: mir01.syntis.net
 * extras: mir01.syntis.net
 * updates: mir01.syntis.net
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package mysql-server.i686 0:5.1.61-4.el6 will be installed
--> Processing Dependency: mysql = 5.1.61-4.el6 for package: mysql-server-5.1.61-4.el6.i686
--> Processing Dependency: perl-DBI for package: mysql-server-5.1.61-4.el6.i686
--> Processing Dependency: perl-DBD-MySQL for package: mysql-server-5.1.61-4.el6.i686
--> Processing Dependency: perl(DBI) for package: mysql-server-5.1.61-4.el6.i686
--> Running transaction check
---> Package mysql.i686 0:5.1.61-4.el6 will be installed
---> Package perl-DBD-MySQL.i686 0:4.013-3.el6 will be installed
```

```
---> Package perl-DBI.i686 0:1.609-4.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch         Version                 Repository    Size
================================================================================
Installing:
 mysql-server           i686         5.1.61-4.el6            base         8.8 M
Installing for dependencies:
 mysql                  i686         5.1.61-4.el6            base         892 k
 perl-DBD-MySQL         i686         4.013-3.el6             base         134 k
 perl-DBI               i686         1.609-4.el6             base         705 k

Transaction Summary
================================================================================
Install       4 Package(s)

Total download size: 10 M
Installed size: 29 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): mysql-5.1.61-4.el6.i686.rpm                       | 892 kB     00:00
(2/4): mysql-server-5.1.61-4.el6.i686.rpm                | 8.8 MB     00:04
(3/4): perl-DBD-MySQL-4.013-3.el6.i686.rpm               | 134 kB     00:00
(4/4): perl-DBI-1.609-4.el6.i686.rpm                     | 705 kB     00:00
--------------------------------------------------------------------------------
Total                                           961 kB/s |  10 MB     00:11
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : perl-DBI-1.609-4.el6.i686                                   1/4
```

```
  Installing : perl-DBD-MySQL-4.013-3.el6.i686                              2/4
  Installing : mysql-5.1.61-4.el6.i686                                      3/4
  Installing : mysql-server-5.1.61-4.el6.i686                               4/4
  Verifying  : mysql-server-5.1.61-4.el6.i686                               1/4
  Verifying  : perl-DBD-MySQL-4.013-3.el6.i686                              2/4
  Verifying  : mysql-5.1.61-4.el6.i686                                      3/4
  Verifying  : perl-DBI-1.609-4.el6.i686                                    4/4

Installed:
  mysql-server.i686 0:5.1.61-4.el6

Dependency Installed:
  mysql.i686 0:5.1.61-4.el6           perl-DBD-MySQL.i686 0:4.013-3.el6
  perl-DBI.i686 0:1.609-4.el6

Complete!
```

Windows

Sous **Windows™**, téléchargez le fichier **mysql-installer-community-5.6.28.0.msi** du site Internet d'Oracle puis exécutez-le.

> Notez que pour les trois systèmes d'exploitation vous pouvez également installer MySQL à partir des sources disponibles sur le site **http://www.mysql.com**.

Démarrage du Serveur

Le serveur MySQL peut être démarré par l'utilisation d'une de deux méthodes différentes.

Sous Red Hat et CentOS

Le Script mysql.server

Ce script est renommé en **mysql** lors de l'installation des binaires en utilisant rpm :

```
[root@centos6 ~]# service mysqld start
Initialisation de la base de données MySQL :  Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h centos password 'new-password'

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &
```

```
You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

                                                           [  OK  ]
Démarrage de mysqld :                                      [  OK  ]
```

Configurez ensuite le service **mysqld** de démarrer lors du démarrage du système :

```
[root@centos6 ~]# chkconfig mysqld on
[root@centos6 ~]# chkconfig --list mysqld
mysqld          0:arrêt    1:arrêt    2:marche    3:marche    4:marche    5:marche    6:arrêt
```

Si vous installez MySQL à partir des sources, le script mysql.server se trouve dans le répertoire **support-files**

Le script **mysql.server** appel un autre script appelé **mysqld_safe** qui lance le serveur et crée un journal d'erreur. Il relance le serveur en cas d'arrêt intempestif.

Invocation Directe

Il est aussi possible d'invoquer directement le binaire **mysqld** en spécifiant manuellement le fichier de configuration de MySQL, le fichier d'erreurs ainsi que le nom de l'utilisateur. Par exemple :

```
# /chemin/mysqld --defaults-file=/chemin/my.cnf --log-error=/chemin/nom_log --user=mysql &
```

Sous Windows

Le Service MySQL56

Si le service n'est pas déjà installé, il convient de saisir la commande suivante :

```
C:\Users\Administrateur>C:\Program Files\MySQL\MySQL Server 5.6\bin\mysqld.exe" --install MySQL56 --defaults-file="C:\ProgramData\MySQL\MySQL Server 5.6\my.ini"
```

Si le service n'est pas déjà démarré, il convient de saisir la commande suivante :

```
C:\Users\Administrateur>start MySQL56
```

Invocation Directe

Il est aussi possible d'invoquer directement le binaire **mysqld** en spécifiant manuellement le fichier de configuration de MySQL :

```
C:\Users\Administrateur>C:\Users\Administrateur>C:\Program Files\MySQL\MySQL Server 5.6\bin\mysqld.exe" --defaults-file="C:\ProgramData\MySQL\MySQL Server 5.6\my.ini"
```

Arrêt du Serveur

Sous Red Hat et CentOS

Le Script mysql.server

Ce script, qu'il soit nommé mysql.server ou simplement mysql, accepte l'argument **stop**.

La Commande mysqladmin

La comande **mysqladmin** peut aussi être utilisée pour arrêter le serveur à condition que l'utilisateur qui l'invoque possède le privilège **shutdown**. Par exemple :

```
# /chemin/mysqladmin -uroot -p shutdown
```

Sous Windows

Le Service MySQL56

Ce service accepte l'argument **stop**.

La Commande mysqladmin

La commande **mysqladmin** peut aussi être utilisée pour arrêter le serveur à condition que l'utilisateur qui l'invoque possède le privilège **shutdown**. Par exemple :

```
C:\Users\Administrateur>C:\Program Files\MySQL\MySQL Server 5.6\bin\mysqladmin.exe -uroot -p shutdown
```

> Le nom d'utilisateur peut être accoler ou non à l'option **-u**. Par exemple **-uroot** et **-u root** sont tous les deux correctes. Par contre, si vous souhaitez spécifier le mot de passe de l'utilisateur dans la ligne de commande, celui-ci **doit** être accoler à l'option **-p**.

Configuration

Votre première prise en mains de MySQL doit débuter par la commande **mysql** :

Le Client MySQL

Utilisation

MySQL dispose d'un outil client permettant de se connecter et d'envoyer des commandes SQL au serveur.

Pour démarrer une connexion au serveur MySQL nous pouvons utiliser un C.L.I. sous Linux.

Les paramètres les plus courants sont :

```
$ mysql –u root –p Databasename [Entrée]
```

Où :

- **-u**
  - donne le nom de l'utilisateur. Si vous renoncez à l'option –u , le nom de login sera utilisé sous Unix/Linux et le nom ODBC sous Windows.
- **-p**
  - permet de saisir le mot de passe de MySQL, cette option est obligatoire lorsque les utilisateurs MySQL sont sécurisés, c'est-à-dire qu'ils ont un mot de passe.
- **-h**
  - permet de préciser le nom de l'hôte qui héberge MySQL. Par défaut le port de communication est 3306 il est possible de changer ce port soit en configurant l'instance MySQL soit en configurant le fichier **My.ini**. (Attention: vérifiez qu'un par feu ne bloque pas ce port). Enfin vous devez vous assurer que le serveur MySQL est configuré de sorte à autoriser les accès depuis votre ordinateur local ce qui n'est pas le cas par défaut.
- **- -protocol = name**
  - cette option permet de spécifier le protocole de communication à utiliser. Il est rarement nécessaire d'indiquer cette option, puisque MySQL choisit le bon protocole par défaut. Lorsque le client MySQL et le serveur MySQL ne tournent pas sur le même ordinateur, le seul protocole réseau possible est le **TCP**, dans ce cas n'oubliez pas de préciser le paramètre –h.
- **-p n**
  - permet de préciser le port de communication utilisée pour se connecter à MySQL. Cette option est effective que lorsque la communication

passe par TCP/IP.

- **- -default-character-set = nom**
  - cette option indique le jeu de caractères utilisés dans le cadre de communication entre MySQL et le serveur MySQL. En théorie, il s'agit du même jeu de caractères que celui utilisé par défaut dans la vie de commandes sous Windows ou la console sous Linux. Les jeux de caractères pris en charge par MySQL sont notamment Latin 1 (ISO-8559-1), latin 2 (ISO-8559-2), UTF-8 (unicode) et cp850 (le jeu de caractères DOS pour l'Europe occidentale).

- Databasename
  - ce dernier paramètre indique à MySQL le nom d'une base de données, ce qui permet de l'utiliser directement dès l'ouverture de MySQL. Si vous désirez changer ici de base de données après sa connexion vous pouvez utiliser la commande SQL USE name.

Exemple :

```
$ mysql -u root -p -h server --protocol=tcp [Entrée]
$ Password : ********** [Entrée]
```

Options

Dans la console MySQL nous avons diverses options.

Abréviation	Commande	Signification
\c	clear	Annule une commande en cours de saisie
\h	help	Affiche la liste des commandes
\q	exit ou quit	Ferme la connexion à MySQL. Sous Unix/Linux, il est possible utilisé le raccourci ctrl+D
\s	status	Affiche les informations de statut du serveur MySQL
\T[f]	tee[filename]	Enregistre tous ce qui apparaît dans la fenêtre de commandes dans le fichier indiqué
L	notee	Ferme tee. Le protocole doit être repris à tout instant avec tee ou \T. Il n'est pas nécessaire de saisir de nouveau le nom de fichier.
\u db	use database	La base de données saisie devient la base de données par défaut
\. Fn	source file name	exécute les commandes SQL contenues dans le fichier spécifié. Les commandes doivent à séparer par des pointsvirgules.

NB : la fonction \c n'a aucun effet dans les chaînes de caractères(“” ou ' ').

Notez aussi que MySQL se souvient des dernières commandes grâce aux touches **flèche vers le haut** et **flèche vers le bas**.

LAB #1 - Configuration de Base

Sous **Red Hat** et **CentOS**

Saisissez la commande **mysql** :

```
[root@centos6 ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Pour visualiser la liste des bases de données par défaut, utilisez la commande suivante :

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
```

```
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.02 sec)

mysql>
```

Ensuite changez de base de données avec la commande **USE**.

```
mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Afin de consulter les tables présentes dans la base, utilisez la commande **SHOW**.

```
mysql> SHOW TABLES;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
```

```
| plugin                    |
| proc                      |
| procs_priv                |
| servers                   |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
23 rows in set (0.00 sec)

mysql>
```

Pour consulter une table spécifique, utilisez la commande **DESCRIBE**.

```
mysql> DESCRIBE user;
+-----------------------+-----------------------------------+------+-----+---------+-------+
| Field                 | Type                              | Null | Key | Default | Extra |
+-----------------------+-----------------------------------+------+-----+---------+-------+
| Host                  | char(60)                          | NO   | PRI |         |       |
| User                  | char(16)                          | NO   | PRI |         |       |
| Password              | char(41)                          | NO   |     |         |       |
| Select_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Insert_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Update_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Delete_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Create_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Drop_priv             | enum('N','Y')                     | NO   |     | N       |       |
| Reload_priv           | enum('N','Y')                     | NO   |     | N       |       |
| Shutdown_priv         | enum('N','Y')                     | NO   |     | N       |       |
```

```
| Process_priv          | enum('N','Y')                       | NO   |     | N       |        |
| File_priv             | enum('N','Y')                       | NO   |     | N       |        |
| Grant_priv            | enum('N','Y')                       | NO   |     | N       |        |
| References_priv       | enum('N','Y')                       | NO   |     | N       |        |
| Index_priv            | enum('N','Y')                       | NO   |     | N       |        |
| Alter_priv            | enum('N','Y')                       | NO   |     | N       |        |
| Show_db_priv          | enum('N','Y')                       | NO   |     | N       |        |
| Super_priv            | enum('N','Y')                       | NO   |     | N       |        |
| Create_tmp_table_priv | enum('N','Y')                       | NO   |     | N       |        |
| Lock_tables_priv      | enum('N','Y')                       | NO   |     | N       |        |
| Execute_priv          | enum('N','Y')                       | NO   |     | N       |        |
| Repl_slave_priv       | enum('N','Y')                       | NO   |     | N       |        |
| Repl_client_priv      | enum('N','Y')                       | NO   |     | N       |        |
| Create_view_priv      | enum('N','Y')                       | NO   |     | N       |        |
| Show_view_priv        | enum('N','Y')                       | NO   |     | N       |        |
| Create_routine_priv   | enum('N','Y')                       | NO   |     | N       |        |
| Alter_routine_priv    | enum('N','Y')                       | NO   |     | N       |        |
| Create_user_priv      | enum('N','Y')                       | NO   |     | N       |        |
| Event_priv            | enum('N','Y')                       | NO   |     | N       |        |
| Trigger_priv          | enum('N','Y')                       | NO   |     | N       |        |
| ssl_type              | enum('','ANY','X509','SPECIFIED')   | NO   |     |         |        |
| ssl_cipher            | blob                                | NO   |     | NULL    |        |
| x509_issuer           | blob                                | NO   |     | NULL    |        |
| x509_subject          | blob                                | NO   |     | NULL    |        |
| max_questions         | int(11) unsigned                    | NO   |     | 0       |        |
| max_updates           | int(11) unsigned                    | NO   |     | 0       |        |
| max_connections       | int(11) unsigned                    | NO   |     | 0       |        |
| max_user_connections  | int(11) unsigned                    | NO   |     | 0       |        |
+-----------------------+-------------------------------------+------+-----+---------+--------+
39 rows in set (0.00 sec)

mysql>
```

Pour visualiser la liste des utilisateurs autorisés pour MySQL, utilisez la commande **SELECT**.

```
mysql> SELECT host, user, password FROM user;
+-----------+------+----------+
| host      | user | password |
+-----------+------+----------+
| localhost | root |          |
| centos    | root |          |
| 127.0.0.1 | root |          |
| localhost |      |          |
| centos    |      |          |
+-----------+------+----------+
5 rows in set (0.00 sec)

mysql>
```

Vous noterez que l'utilisateur root, l'administrateur de MySQL, n'a pas de mot de passe !

Il faut par conséquence en définir un en urgence !

Pour sortir de l'administration de MySQL, utilisez la commande **exit** :

```
mysql> exit
Bye
[root@centos6 ~]#
```

Pour définir le mot de passe **fenestros** pour root, il convient de saisir la commande suivante :

```
[root@centos6 ~]# mysqladmin -u root password fenestros
```

Lors de la prochaine tentative de connexion en tant que root, vous obtiendrez un message d'erreur car le mot de passe est maintenant non-null :

```
[root@centos6 ~]# mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

Pour vous connecter en tant que l'administrateur de la base il faut maintenant utiliser la commande suivante :

```
[root@centos6 ~]# mysql -u root -p mysql
Enter password: fenestros
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Notez l'utilisation de l'option **-p** qui indique à **mysql** que vous souhaitez saisir un mot de passe. Le mot de passe n'est **pas** mysql mais **fenestros**, comme démontre l'exemple ci-dessus. Bien entendu le mot de passe **fenestros** n'apparaît pas réellement en clair. L'utilisation de l'option **mysql** dans la ligne de commande indique simplement que nous souhaitons se connecter à la base **mysql** dès la connexion.

Saisissez la commande suivante pour vérifier la table des utilisateurs :

```
mysql> SELECT host, user, password FROM user;
+-----------+------+-------------------------------------------+
| host      | user | password                                  |
+-----------+------+-------------------------------------------+
| localhost | root | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| centos    | root |                                           |
```

```
| 127.0.0.1 | root |                                                |
| localhost |      |                                                |
| centos    |      |                                                |
+-----------+------+------------------------------------------------+
5 rows in set (0.00 sec)

mysql>
```

Notez que non seulement le mot de passe de root est présent mais qu'il a été crypté. Vous avez aussi la possibilité de sécuriser votre installation de MySQL en utilisant le script **/usr/bin/mysql_secure_installation**.

Sous Windows

Cliquez sur **Tous les programmes > MySQL > MySQL Server 5.6 > MySQL 5.6 Command Line Client** :

```
Enter password: *********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.28-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Pour visualiser la liste des bases de données par défaut, utilisez la commande suivante :

```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.01 sec)

mysql>
```

Ensuite changez de base de données avec la commande **USE**.

```
mysql> USE mysql;
Database changed
mysql>
```

Afin de consulter les tables présentes dans la base, utilisez la commande **SHOW**.

```
mysql> SHOW TABLES;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
```

```
| help_relation             |
| help_topic                |
| innodb_index_stats        |
| innodb_table_stats        |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| servers                   |
| slave_master_info         |
| slave_relay_log_info      |
| slave_worker_info         |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
28 rows in set (0.03 sec)

mysql>
```

Pour consulter une table spécifique, utilisez la commande **DESCRIBE**.

```
mysql> DESCRIBE user;
+------------------------+-----------------------------------+------+-----+-----------------------+-------+
| Field                  | Type                              | Null | Key | Default               | Extra |
+------------------------+-----------------------------------+------+-----+-----------------------+-------+
| Host                   | char(60)                          | NO   | PRI |                       |       |
| User                   | char(16)                          | NO   | PRI |                       |       |
```

```
| Password                | char(41)                                | NO   |     |                       |       |
| Select_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Insert_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Update_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Delete_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Create_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Drop_priv               | enum('N','Y')                           | NO   |     | N                     |       |
| Reload_priv             | enum('N','Y')                           | NO   |     | N                     |       |
| Shutdown_priv           | enum('N','Y')                           | NO   |     | N                     |       |
| Process_priv            | enum('N','Y')                           | NO   |     | N                     |       |
| File_priv               | enum('N','Y')                           | NO   |     | N                     |       |
| Grant_priv              | enum('N','Y')                           | NO   |     | N                     |       |
| References_priv         | enum('N','Y')                           | NO   |     | N                     |       |
| Index_priv              | enum('N','Y')                           | NO   |     | N                     |       |
| Alter_priv              | enum('N','Y')                           | NO   |     | N                     |       |
| Show_db_priv            | enum('N','Y')                           | NO   |     | N                     |       |
| Super_priv              | enum('N','Y')                           | NO   |     | N                     |       |
| Create_tmp_table_priv   | enum('N','Y')                           | NO   |     | N                     |       |
| Lock_tables_priv        | enum('N','Y')                           | NO   |     | N                     |       |
| Execute_priv            | enum('N','Y')                           | NO   |     | N                     |       |
| Repl_slave_priv         | enum('N','Y')                           | NO   |     | N                     |       |
| Repl_client_priv        | enum('N','Y')                           | NO   |     | N                     |       |
| Create_view_priv        | enum('N','Y')                           | NO   |     | N                     |       |
| Show_view_priv          | enum('N','Y')                           | NO   |     | N                     |       |
| Create_routine_priv     | enum('N','Y')                           | NO   |     | N                     |       |
| Alter_routine_priv      | enum('N','Y')                           | NO   |     | N                     |       |
| Create_user_priv        | enum('N','Y')                           | NO   |     | N                     |       |
| Event_priv              | enum('N','Y')                           | NO   |     | N                     |       |
| Trigger_priv            | enum('N','Y')                           | NO   |     | N                     |       |
| Create_tablespace_priv  | enum('N','Y')                           | NO   |     | N                     |       |
| ssl_type                | enum('','ANY','X509','SPECIFIED')       | NO   |     |                       |       |
| ssl_cipher              | blob                                    | NO   |     | NULL                  |       |
| x509_issuer             | blob                                    | NO   |     | NULL                  |       |
| x509_subject            | blob                                    | NO   |     | NULL                  |       |
```

```
| max_questions           | int(11) unsigned                  | NO   |     | 0                     |       |
| max_updates             | int(11) unsigned                  | NO   |     | 0                     |       |
| max_connections         | int(11) unsigned                  | NO   |     | 0                     |       |
| max_user_connections    | int(11) unsigned                  | NO   |     | 0                     |       |
| plugin                  | char(64)                          | YES  |     | mysql_native_password |       |
| authentication_string   | text                              | YES  |     | NULL                  |       |
| password_expired        | enum('N','Y')                     | NO   |     | N                     |       |
+-------------------------+-----------------------------------+------+-----+-----------------------+-------+
43 rows in set (0.02 sec)

mysql>
```

Pour visualiser la liste des utilisateurs autorisés pour MySQL, utilisez la commande **SELECT**.

```
mysql> SELECT host, user, password FROM user;
+-----------+------+-------------------------------------------+
| host      | user | password                                  |
+-----------+------+-------------------------------------------+
| localhost | root | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| 127.0.0.1 | root | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| ::1       | root | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
+-----------+------+-------------------------------------------+
3 rows in set (0.02 sec)

mysql>
```

Notez que les mots de passe de root sont présents mais qu'ils ont été cryptés. Vous avez aussi la possibilité de sécuriser votre installation de MySQL en utilisant le script **C:\Program Files\MySQL\MySQL Server 5.6\bin\mysql_secure_installation.pl**.

LAB #2 - Configuration Avancée

Sous Red Hat et CentOS

La configuration avancée du serveur MySQL se fait :

- soit en éditant le fichier **/etc/my.cnf**,
- soit en passant des paramètres à l'exécutable mysqld,
- soit en paramétrant le serveur dynamiquement.

Le fichier my.cnf

my.cnf

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

# Settings user and group are ignored when systemd is used (fedora >= 15).
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld according to the
# instructions in http://fedoraproject.org/wiki/Systemd
user=mysql

# Semisynchronous Replication
# http://dev.mysql.com/doc/refman/5.5/en/replication-semisync.html
# uncomment next line on MASTER
;plugin-load=rpl_semi_sync_master=semisync_master.so
# uncomment next line on SLAVE
;plugin-load=rpl_semi_sync_slave=semisync_slave.so
```

```
# Others options for Semisynchronous Replication
;rpl_semi_sync_master_enabled=1
;rpl_semi_sync_master_timeout=10
;rpl_semi_sync_slave_enabled=1

# http://dev.mysql.com/doc/refman/5.5/en/performance-schema.html
;performance_schema


[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Ce fichier n'est pas cependant le seul endroit où est configuré le serveur. En effet, le serveur lit des directives des fichiers /etc/my.cnf /etc/mysql/my.cnf /usr/etc/my.cnf ~/.my.cnf dans l'ordre, comme démontre la sortie de la commande **/usr/libexec/mysqld –help –verbose** :

```
[root@centos6 ~]# /usr/libexec/mysqld --help --verbose | grep my.cnf
140125 14:59:00 [Note] libgovernor.so not found
140125 14:59:00 [Warning] Although a path was specified for the --log-slow-queries option, log tables are used.
To enable logging to files use the --log-output=file option.
140125 14:59:00 [Note] Plugin 'FEDERATED' is disabled.
/etc/my.cnf /etc/mysql/my.cnf /usr/etc/my.cnf ~/.my.cnf
                      my.cnf, $MYSQL_TCP_PORT, /etc/services, built-in default
```

> En cas de duplicité de directives, c'est la dernière lue qui l'emporte. Si vous avez un autre fichier de configuration qui n'est pas un de ceux mentionné, vous pouvez l'utiliser ainsi : /usr/libexec/mysqld –defaults-file=/chemin/votre_fichier.

Le fichier my.cnf est organisé en sections, aussi appelées des groupes. Ces sections font référence au programme concerné par la configuration. Par exemple la section **[mysqld]** fait référence au serveur, tandis qu'une section [mysql] ferait référence au client en mode texte. Il est aussi possible de

trouver les sections suivantes :

- [mysqldump],
- [mysqladmin],
- [mysqlhotcopy],
- [myisamchk],
- [client],
- etc.

> [client] est une section qui configure **tous** les clients. [mysqld] peut être remplacé par [server]. Toute ligne commençant par le caractère **#** ou **;** est un **commentaire**.

Le fichier de configuration peut aussi contenir des lignes **!include** qui référencent des fichiers tiers de configuration ainsi que des lignes **!includedir** qui référencent des répertoires contenant plusieurs fichiers de configuration. Attention, dans ce dernier cas, l'administrateur n'a aucun contrôle sur l'ordre de lecture des fichiers par le serveur.

Chaque section contient des directives au format **option=valeur** ou au format binaire, par exemple **enable-federated**.

Avec presque **300** options possibles et avec autant d'emplacements de fichiers de configuration, deux problèmes se posent.

- Comment savoir bien configurer le serveur,
- Comment connaître avec exactitude la configuration actuelle.

Comment Savoir Bien Configurer le Serveur ?

La réponse au premier problème se trouve dans les fichiers exemples fournis par Oracle. Ces fichiers se trouvent dans le répertoire **support-files**. Oracle propose des fichiers de configuration en fonction de la mémoire du serveur :

```
[root@centos6 ~]# ls -l /usr/share/mysql | grep cnf
-rw-r--r--. 1 root root   4697  2 août  23:07 my-huge.cnf
-rw-r--r--. 1 root root  19779  2 août  23:07 my-innodb-heavy-4G.cnf
-rw-r--r--. 1 root root   4671  2 août  23:07 my-large.cnf
```

```
-rw-r--r--. 1 root root   4682  2 août  23:07 my-medium.cnf
-rw-r--r--. 1 root root   2846  2 août  23:07 my-small.cnf
```

Par exemple :

```
[root@centos6 ~]# cat /usr/share/mysql/my-huge.cnf
# Example MySQL config file for very large systems.
#
# This is for a large system with memory of 1G-2G where the system runs mainly
# MySQL.
#
# MySQL programs look for option files in a set of
# locations which depend on the deployment platform.
# You can copy this option file to one of those
# locations. For information about these locations, see:
# http://dev.mysql.com/doc/mysql/en/option-files.html
#
# In this file, you can use all long options that a program supports.
# If you want to know which options a program supports, run the program
# with the "--help" option.

# The following options will be passed to all MySQL clients
[client]
#password   = your_password
port        = 3306
socket      = /var/lib/mysql/mysql.sock

# Here follows entries for some specific programs

# The MySQL server
[mysqld]
port        = 3306
socket      = /var/lib/mysql/mysql.sock
skip-external-locking
```

```
key_buffer_size = 384M
max_allowed_packet = 1M
table_open_cache = 512
sort_buffer_size = 2M
read_buffer_size = 2M
read_rnd_buffer_size = 8M
myisam_sort_buffer_size = 64M
thread_cache_size = 8
query_cache_size = 32M
# Try number of CPU's*2 for thread_concurrency
thread_concurrency = 8

# Don't listen on a TCP/IP port at all. This can be a security enhancement,
# if all processes that need to connect to mysqld run on the same host.
# All interaction with mysqld must be made via Unix sockets or named pipes.
# Note that using this option without enabling named pipes on Windows
# (via the "enable-named-pipe" option) will render mysqld useless!
#
#skip-networking

# Replication Master Server (default)
# binary logging is required for replication
log-bin=mysql-bin

# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id   = 1

# Replication Slave (comment out master section to use this)
#
# To configure this host as a replication slave, you can choose between
# two methods :
#
```

```
# 1) Use the CHANGE MASTER TO command (fully described in our manual) -
#    the syntax is:
#
#    CHANGE MASTER TO MASTER_HOST=<host>, MASTER_PORT=<port>,
#    MASTER_USER=<user>, MASTER_PASSWORD=<password> ;
#
#    where you replace <host>, <user>, <password> by quoted strings and
#    <port> by the master's port number (3306 by default).
#
#    Example:
#
#    CHANGE MASTER TO MASTER_HOST='125.564.12.1', MASTER_PORT=3306,
#    MASTER_USER='joe', MASTER_PASSWORD='secret';
#
# OR
#
# 2) Set the variables below. However, in case you choose this method, then
#    start replication for the first time (even unsuccessfully, for example
#    if you mistyped the password in master-password and the slave fails to
#    connect), the slave will create a master.info file, and any later
#    change in this file to the variables' values below will be ignored and
#    overridden by the content of the master.info file, unless you shutdown
#    the slave server, delete master.info and restart the slaver server.
#    For that reason, you may want to leave the lines below untouched
#    (commented) and instead use CHANGE MASTER TO (see above)
#
# required unique id between 2 and 2^32 - 1
# (and different from the master)
# defaults to 2 if master-host is set
# but will not function as a slave if omitted
#server-id       = 2
#
# The replication master for this slave - required
#master-host     =   <hostname>
```

```
#
# The username the slave will use for authentication when connecting
# to the master - required
#master-user     =   <username>
#
# The password the slave will authenticate with when connecting to
# the master - required
#master-password =   <password>
#
# The port the master is listening on.
# optional - defaults to 3306
#master-port     =  <port>
#
# binary logging - not required for slaves, but recommended
#log-bin=mysql-bin
#
# binary logging format - mixed recommended
#binlog_format=mixed

# Uncomment the following if you are using InnoDB tables
#innodb_data_home_dir = /var/lib/mysql
#innodb_data_file_path = ibdata1:2000M;ibdata2:10M:autoextend
#innodb_log_group_home_dir = /var/lib/mysql
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
#innodb_buffer_pool_size = 384M
#innodb_additional_mem_pool_size = 20M
# Set .._log_file_size to 25 % of buffer pool size
#innodb_log_file_size = 100M
#innodb_log_buffer_size = 8M
#innodb_flush_log_at_trx_commit = 1
#innodb_lock_wait_timeout = 50

[mysqldump]
```

```
quick
max_allowed_packet = 16M

[mysql]
no-auto-rehash
# Remove the next comment character if you are not familiar with SQL
#safe-updates

[myisamchk]
key_buffer_size = 256M
sort_buffer_size = 256M
read_buffer = 2M
write_buffer = 2M

[mysqlhotcopy]
interactive-timeout
```

Pour connaître toutes les options valides pour le serveur, il convient de saisir la commande suivante :

```
[root@centos6 ~]# /usr/libexec/mysqld --help --verbose | more
140125 15:32:58 [Note] libgovernor.so not found
140125 15:32:58 [Warning] Although a path was specified for the --log-slow-queries option, log tables are used.
To enable logging to files use the --log-output=file option.
140125 15:32:58 [Note] Plugin 'FEDERATED' is disabled.
/usr/libexec/mysqld  Ver 5.5.33-cll-lve for Linux on i686 (MySQL Community Server (GPL) by Atomicorp)
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Starts the MySQL database server.

Usage: /usr/libexec/mysqld [OPTIONS]
```

```
Default options are read from the following files in the given order:
/etc/my.cnf /etc/mysql/my.cnf /usr/etc/my.cnf ~/.my.cnf
The following groups are read: mysqld server mysqld-5.5
The following options may be given as the first argument:
--print-defaults        Print the program argument list and exit.
--no-defaults           Don't read default options from any option file.
--defaults-file=#       Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.

  --abort-slave-event-count=#
                      Option used by mysql-test for debugging and testing of
                      replication.
  --allow-suspicious-udfs
                      Allows use of UDFs consisting of only one symbol xxx()
                      without corresponding xxx_init() or xxx_deinit(). That
                      also means that one can load any function from any
                      library, for example exit() from libc.so
  -a, --ansi          Use ANSI SQL syntax instead of MySQL syntax. This mode
                      will also set transaction isolation level 'serializable'.
  --archive[=name]    Enable or disable ARCHIVE plugin. Possible values are ON,
                      OFF, FORCE (don't start if the plugin fails to load).
--Plus--
```

Pour connaître les options valides pour le client mysql, l'option de la ligne de commande **–verbose** n'est pas requise :

```
[root@centos6 ~]# mysql --help | more
mysql  Ver 14.14 Distrib 5.5.33, for Linux (i686) using readline 5.1
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
```

```
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to use
                      'rehash' to get table and field completion, but startup
                      and reconnecting may take a longer time. Disable with
                      --disable-auto-rehash.
                      (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash
                      No automatic rehashing. One has to use 'rehash' to get
                      table and field completion. This gives a quicker start of
                      mysql and disables rehashing on reconnect.
  --auto-vertical-output
                      Automatically switch to vertical output mode if the
                      result is wider than the terminal width.
  -B, --batch         Don't use history file. Disable interactive behavior.
                      (Enables --silent.)
  --character-sets-dir=name
                      Directory for character set files.
  --column-type-info  Display column type information.
  -c, --comments      Preserve comments. Send comments to the server. The
                      default is --skip-comments (discard comments), enable
                      with --comments.
  -C, --compress      Use compression in server/client protocol.
  -#, --debug[=#]     This is a non-debug version. Catch this and exit.
--Plus--
```

Comment Connaître avec Exactitude la Configuration Actuelle ?

La réponse à la deuxième question est obtenue en utilisant la commande **mysqladmin** :

```
[root@centos6 ~]# mysqladmin variables | more
+-------------------------------------------------+-----------------------------------------------------------
-------------------------------------------------------
-----+
```

```
| Variable_name                                    | Value
     |
+--------------------------------------------------+-------------------------------------------------------
-------------------------------------------------------
-----+
| auto_increment_increment                         | 1
     |
| auto_increment_offset                            | 1
     |
| autocommit                                       | ON
     |
| automatic_sp_privileges                          | ON
     |
| back_log                                         | 50
     |
| basedir                                          | /usr
     |
| big_tables                                       | OFF
     |
| binlog_cache_size                                | 32768
     |
| binlog_direct_non_transactional_updates          | OFF
     |
| binlog_format                                    | STATEMENT
     |
| binlog_stmt_cache_size                           | 32768
     |
| bulk_insert_buffer_size                          | 8388608
     |
| character_set_client                             | latin1
     |
--Plus--
```

Passer des Paramètres à l'Exécutable mysqld

Le serveur mysqld peut être paramétrer en passant des options à l'exécutable lors de son lancement. Dans ce cas, les options sont les mêmes que celles dans le fichier my.ini, précédées par deux tirés.

Paramétrer le Serveur Dynamiquement

Pour paramétrer le serveur à chaud, il convient d'utiliser la commande **SET**. Pour utiliser la commande **SET**, il faut posséder le privilège **SUPER**.

La portée des options peut être SESSION, c'est-à-dire pour la session en cours, ou GLOBAL pour toutes le sessions.

SESSION

Prenant l'example de l'option **tmp_table_size** :

```
mysql> SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
+----------------+----------+
| tmp_table_size | 16777216 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
+----------------+----------+
| tmp_table_size | 16777216 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SET SESSION tmp_table_size=8388608;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
```

```
+----------------+----------+
| tmp_table_size | 16777216 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+---------+
| Variable_name  | Value   |
+----------------+---------+
| tmp_table_size | 8388608 |
+----------------+---------+
1 row in set (0.00 sec)

mysql>
```

Notez que la modification est immédiate.

GLOBAL

En utilisant la même option :

```
mysql> SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
+----------------+----------+
| tmp_table_size | 16777216 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+---------+
```

```
| Variable_name  | Value   |
+----------------+---------+
| tmp_table_size | 8388608 |
+----------------+---------+
1 row in set (0.00 sec)

mysql> SET GLOBAL tmp_table_size=16777216;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+---------+
| Variable_name  | Value   |
+----------------+---------+
| tmp_table_size | 8388608 |
+----------------+---------+
1 row in set (0.01 sec)

mysql>
```

Notez que pour une option ayant à la fois une portée globale et une portée session, la modification de la valeur globale n'est pas prise en compte dans la session active.

Prenons maintenant le cas d'une option qui n'a **qu'une** portée globale :

```
mysql> SHOW VARIABLES LIKE 'query_cache_size';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| query_cache_size | 0     |
+------------------+-------+
1 row in set (0.00 sec)
```

```
mysql> SET SESSION query_cache_size=122880;
ERROR 1229 (HY000): Variable 'query_cache_size' is a GLOBAL variable and should be set with SET GLOBAL
mysql> SET GLOBAL query_cache_size=122880;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW VARIABLES LIKE 'query_cache_size';
+------------------+--------+
| Variable_name    | Value  |
+------------------+--------+
| query_cache_size | 122880 |
+------------------+--------+
1 row in set (0.01 sec)

mysql>
```

Notez que pour une option ayant uniquement une portée globale, la modification de la valeur globale est prise en compte dans la session active.

Les modifications à chaud ne sont pas persistants.

Sous Windows

La configuration avancée du serveur MySQL se fait :

- soit en éditant le fichier **C:\ProgramData\MySQL\MySQL Server 5.6\my.ini**,
- soit en passant des paramètres à l'exécutable mysqld,
- soit en paramétrant le serveur dynamiquement.

Le fichier C:\ProgramData\MySQL\MySQL Server 5.6\my.ini

my.ini

```
C:\ProgramData>more "c:\ProgramData\MySQL\MySQL Server 5.6\my.ini"
# Other default tuning values
# MySQL Server Instance Configuration File
# ----------------------------------------------------------------------
# Generated by the MySQL Server Instance Configuration Wizard
#
#
# Installation Instructions
# ----------------------------------------------------------------------
#
# On Linux you can copy this file to /etc/my.cnf to set global options,
# mysql-data-dir/my.cnf to set server-specific options
# (@localstatedir@ for this installation) or to
# ~/.my.cnf to set user-specific options.
#
# On Windows you should keep this file in the installation directory
# of your server (e.g. C:\Program Files\MySQL\MySQL Server X.Y). To
# make sure the server reads the config file use the startup option
# "--defaults-file".
#
# To run run the server from the command line, execute this in a
# command line shell, e.g.
# mysqld --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"
#
# To install the server as a Windows service manually, execute this in a
# command line shell, e.g.
# mysqld --install MySQLXY --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"
#
# And then execute this in a command line shell to start the server, e.g.
# net start MySQLXY
```

```
#
#
# Guidlines for editing this file
# ------------------------------------------------------------------------
#
# In this file, you can use all long options that the program supports.
# If you want to know the options a program supports, start the program
# with the "--help" option.
#
# More detailed information about the individual options can also be
# found in the manual.
#
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html
#
#
# CLIENT SECTION
# ------------------------------------------------------------------------
#
# The following options will be read by MySQL client applications.
# Note that only client applications shipped by MySQL are guaranteed
-- Suite (14%) --
```

Ce fichier n'est pas cependant le seul endroit où est configuré le serveur. En effet, le serveur lit des directives des fichiers C:\Windows\my.ini, C:\Windows\my.cnf, C:\my.ini, C:\my.cnf, C:\Program Files\MySQL\MySQL Server 5.6\my.ini et C:\Program Files\MySQL\MySQL Server 5.6\my.cnf dans l'ordre, comme démontre la sortie de la commande **mysqld –help –verbose | findstr my.ini** :

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqld --help --verbose | findstr my.ini
2016-02-22 14:42:03 0 [Note] mysqld (mysqld 5.6.28) starting as process 3636 ...
2016-02-22 14:42:03 3636 [Note] Plugin 'FEDERATED' is disabled.
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf C:\Program Files\MySQL\MySQL Server 5.6\my.ini C:\Program
Files\MySQL\MySQL Server 5.6\my.cnf
2016-02-22 14:42:03 3636 [Note] Binlog end
```

```
2016-02-22 14:42:03 3636 [Note] Shutting down plugin 'MyISAM'
2016-02-22 14:42:03 3636 [Note] Shutting down plugin 'CSV'

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

> En cas de duplicité de directives, c'est la dernière lue qui l'emporte. Si vous avez un autre fichier de configuration qui n'est pas un de ceux mentionné, vous pouvez l'utiliser ainsi : mysqld –defaults-file=/chemin/votre_fichier.

Le fichier my.ini est organisé en sections, aussi appelées des groupes. Ces sections font référence au programme concerné par la configuration. Par exemple la section **[mysqld]** fait référence au serveur, tandis qu'une section [mysql] ferait référence au client en mode texte. Il est aussi possible de trouver les sections suivantes :

- [mysqldump],
- [mysqladmin],
- [mysqlhotcopy],
- [myisamchk],
- [client],
- etc.

> [client] est une section qui configure **tous** les clients. [mysqld] peut être remplacé par [server]. Toute ligne commençant par le caractère **#** ou **;** est un **commentaire**.

Le fichier de configuration peut aussi contenir des lignes **!include** qui référencent des fichiers tiers de configuration ainsi que des lignes **!includedir** qui référencent des répertoires contenant plusieurs fichiers de configuration. Attention, dans ce dernier cas, l'administrateur n'a aucun contrôle sur l'ordre de lecture des fichiers par le serveur.

Chaque section contient des directives au format **option=valeur** ou au format binaire, par exemple **enable-federated**.

Avec presque **300** options possibles et avec autant d'emplacements de fichiers de configuration, deux problèmes se posent.

- Comment savoir bien configurer le serveur,
- Comment connaître avec exactitude la configuration actuelle.

Comment Savoir Bien Configurer le Serveur ?

La réponse au premier problème se trouve dans les fichiers exemples fournis par Oracle. Ces fichiers se trouvent dans le répertoire **support-files**. Oracle propose des fichiers de configuration en fonction de la version du serveur :

```
c:\ProgramData\MySQL\MySQL Installer for Windows\Manifest\Templates>dir
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 7C10-520B

 Répertoire de c:\ProgramData\MySQL\MySQL Installer for Windows\Manifest\Templates

21/02/2016  15:49    <REP>          .
21/02/2016  15:49    <REP>          ..
21/02/2016  15:49            15 263 my-template-5.1.ini
21/02/2016  15:49            15 263 my-template-5.5.ini
21/02/2016  15:49            20 663 my-template-5.6.ini
21/02/2016  15:49            20 001 my-template-5.7.ini
21/02/2016  15:49            15 127 my-template.ini
               5 fichier(s)          86 317 octets
               2 Rép(s)  15 690 223 616 octets libres

c:\ProgramData\MySQL\MySQL Installer for Windows\Manifest\Templates>
```

Par exemple :

```
c:\ProgramData\MySQL\MySQL Installer for Windows\Manifest\Templates>more my-template-5.6.ini
## MySQL Server Instance Configuration File Template
## ------------------------------------------------------------------------
## Version 1.0.10
##
## <-- Indicates Template comment.  These lines will not be in the output
```

```
##
## Replaceable things must be like:
##
## # [VARIABLE_NAME]="Formula"
## parameter=default value
##
##  For example:
##
##  # [PORT]="port"
##  port=3306
##
## Note - In the example, the formula consists of a variable named 'port' which must be defined before
processing.
##
## In addition to the standard max operators (+, -, /, *), the "formula" field supports the following functions:
##
##  rnd(x, y) = Round x to the nearest y
##  max(x, y) = Max value from x, y
##  min(x, y) = Min value from x, y
##
## and named variables.
##
##  For example:
##
##  # [MAX_CONNECTIONS]="max_connections:rnd(max(100,max_connections),1000)"
##  max_connections=
##
## ( Note - Uninitialized variables have a value of 0. )
##
## Finally, there is a special directive named [STATE_CHANGE] that allows for a function to be exectuted at that
## point during template processing.
##
##  For example:
## # [STATE_CHANGE]="new_variable : 1"
```

```
##
## The following variables must be defined before the formulas are evaluated (otherwise, you get many values set
to 0):
##
## memory             - Server Type
##                      Dedicated Server (90% of System Memory), Server (50% of System Memory), All others(
rnd(max( 1/12 System Memory, 40*1024*1024
4))
## myiasm_percentage  - Table Type
##                      If main InnoDB, set to 5.  Allow userdef.
## active_connections - # Connections.
##                      DSS = 20, OLTP = 500, else user_defined.
## cpus               - Number of CPUS on the machine.
##
-- Suite (8%) --
```

Pour connaître toutes les options valides pour le serveur, il convient de saisir la commande suivante :

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqld --verbose --help | more
2016-02-22 15:19:30 0 [Note] mysqld (mysqld 5.6.28) starting as process 2336 ...
2016-02-22 15:19:30 2336 [Note] Plugin 'FEDERATED' is disabled.
mysqld  Ver 5.6.28 for Win32 on AMD64 (MySQL Community Server (GPL))
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Starts the MySQL database server.

Usage: mysqld [OPTIONS]
NT and Win32 specific options:
  --install                     Install the default service (NT).
  --install-manual              Install the default service started manually (NT).
```

```
  --install service_name        Install an optional service (NT).
  --install-manual service_name Install an optional service started manually (NT).
  --remove                      Remove the default service from the service list (NT).
  --remove service_name         Remove the service_name from the service list (NT).
  --enable-named-pipe           Only to be used for the default server (NT).
  --standalone                  Dummy option to start as a standalone server (NT).


Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf C:\Program Files\MySQL\MySQL Server 5.6\my.ini C:\Program
Files\MySQL\MySQL Server 5.6\m
The following groups are read: mysqld server mysqld-5.6
The following options may be given as the first argument:
--print-defaults        Print the program argument list and exit.
--no-defaults           Don't read default options from any option file,
                        except for login file.
--defaults-file=#       Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix=#
                        Also read groups with concat(group, suffix)
--login-path=#          Read this path from the login file.

  --abort-slave-event-count=#
                      Option used by mysql-test for debugging and testing of
                      replication.
  --allow-suspicious-udfs
                      Allows use of UDFs consisting of only one symbol xxx()
                      without corresponding xxx_init() or xxx_deinit(). That
                      also means that one can load any function from any
                      library, for example exit() from libc.so
  -a, --ansi          Use ANSI SQL syntax instead of MySQL syntax. This mode
                      will also set transaction isolation level 'serializable'.
  --archive[=name]    Enable or disable ARCHIVE plugin. Possible values are ON,
                      OFF, FORCE (don't start if the plugin fails to load).
```

```
  --auto-increment-increment[=#]
                      Auto-increment columns are incremented by this
  --auto-increment-offset[=#]
                      Offset added to Auto-increment columns. Used when
-- Suite  --
```

Pour connaître les options valides pour le client mysql, l'option de la ligne de commande **–verbose** n'est pas requise :

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysql --help | more
mysql  Ver 14.14 Distrib 5.6.28, for Win32 (AMD64)
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to use
                      'rehash' to get table and field completion, but startup
                      and reconnecting may take a longer time. Disable with
                      --disable-auto-rehash.
                      (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash
                      No automatic rehashing. One has to use 'rehash' to get
                      table and field completion. This gives a quicker start of
                      mysql and disables rehashing on reconnect.
  --auto-vertical-output
                      Automatically switch to vertical output mode if the
                      result is wider than the terminal width.
  -B, --batch         Don't use history file. Disable interactive behavior.
                      (Enables --silent.)
  --bind-address=name IP address to bind to.
```

```
  --character-sets-dir=name
                      Directory for character set files.
  --column-type-info  Display column type information.
  -c, --comments      Preserve comments. Send comments to the server. The
                      default is --skip-comments (discard comments), enable
                      with --comments.
  -C, --compress      Use compression in server/client protocol.
  -#, --debug[=#]     This is a non-debug version. Catch this and exit.
  --debug-check       Check memory and open file usage at exit.
  -T, --debug-info    Print some debug info at exit.
  -D, --database=name Database to use.
  --default-character-set=name
                      Set the default character set.
  --delimiter=name    Delimiter to be used.
  --enable-cleartext-plugin
                      Enable/disable the clear text authentication plugin.
  -e, --execute=name  Execute command and quit. (Disables --force and history
                      file.)
  -E, --vertical      Print the output of a query (rows) vertically.
  -f, --force         Continue even if we get an SQL error.
  -G, --named-commands
                      Enable named commands. Named commands mean this program's
                      internal commands; see mysql> help . When enabled, the
                      named commands can be used from any line of the query,
                      otherwise only from the first line, before an enter.
-- Suite  --
```

Comment Connaître avec Exactitude la Configuration Actuelle ?

La réponse à la deuxième question est obtenue en utilisant la commande **mysqladmin** :

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqladmin -u root -pfenestros extended-status | more
Warning: Using a password on the command line interface can be insecure.
+-------------------------------------------------+-------------+
```

```
| Variable_name                                 | Value       |
+-----------------------------------------------+-------------+
| Aborted_clients                               | 0           |
| Aborted_connects                              | 3           |
| Binlog_cache_disk_use                         | 0           |
| Binlog_cache_use                              | 0           |
| Binlog_stmt_cache_disk_use                    | 0           |
| Binlog_stmt_cache_use                         | 0           |
| Bytes_received                                | 2209        |
| Bytes_sent                                    | 16452       |
| Com_admin_commands                            | 0           |
| Com_assign_to_keycache                        | 0           |
| Com_alter_db                                  | 0           |
| Com_alter_db_upgrade                          | 0           |
| Com_alter_event                               | 0           |
| Com_alter_function                            | 0           |
| Com_alter_procedure                           | 0           |
| Com_alter_server                              | 0           |
| Com_alter_table                               | 0           |
| Com_alter_tablespace                          | 0           |
| Com_alter_user                                | 0           |
| Com_analyze                                   | 0           |
| Com_begin                                     | 0           |
| Com_binlog                                    | 0           |
| Com_call_procedure                            | 0           |
| Com_change_db                                 | 2           |
| Com_change_master                             | 0           |
| Com_check                                     | 0           |
| Com_checksum                                  | 0           |
| Com_commit                                    | 0           |
| Com_create_db                                 | 0           |
| Com_create_event                              | 0           |
| Com_create_function                           | 0           |
| Com_create_index                              | 0           |
```

```
| Com_create_procedure                     | 0          |
| Com_create_server                        | 0          |
| Com_create_table                         | 0          |
| Com_create_trigger                       | 0          |
| Com_create_udf                           | 0          |
| Com_create_user                          | 0          |
| Com_create_view                          | 0          |
| Com_dealloc_sql                          | 0          |
| Com_delete                               | 0          |
| Com_delete_multi                         | 0          |
| Com_do                                   | 0          |
| Com_drop_db                              | 0          |
| Com_drop_event                           | 0          |
| Com_drop_function                        | 0          |
| Com_drop_index                           | 0          |
-- Suite  --
```

Passer des Paramètres à l'Exécutable mysqld

Le serveur mysqld peut être paramétrer en passant des options à l'exécutable lors de son lancement. Dans ce cas, les options sont les mêmes que celles dans le fichier my.ini, précédées par deux tirés.

Paramétrer le Serveur Dynamiquement

Pour paramétrer le serveur à chaud, il convient d'utiliser la commande **SET**. Pour utiliser la commande **SET**, il faut posséder le privilège **SUPER**.

La portée des options peut être SESSION, c'est-à-dire pour la session en cours, ou GLOBAL pour toutes le sessions.

SESSION

Prenant l'example de l'option **tmp_table_size**.

```
mysql> SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
```

```
+----------------+----------+
| tmp_table_size | 63963136 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
+----------------+----------+
| tmp_table_size | 63963136 |
+----------------+----------+
1 row in set (0.01 sec)

mysql> SET SESSION tmp_table_size=8388608;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
+----------------+----------+
| Variable_name  | Value    |
+----------------+----------+
| tmp_table_size | 63963136 |
+----------------+----------+
1 row in set (0.00 sec)

mysql> SHOW SESSION VARIABLES LIKE 'tmp_table_size';
+----------------+---------+
| Variable_name  | Value   |
+----------------+---------+
| tmp_table_size | 8388608 |
+----------------+---------+
1 row in set (0.00 sec)

mysql>
```

Notez que la modification est immédiate.

GLOBAL

```
mysql> SHOW VARIABLES LIKE 'query_cache_size';
+------------------+---------+
| Variable_name    | Value   |
+------------------+---------+
| query_cache_size | 1048576 |
+------------------+---------+
1 row in set (0.01 sec)

mysql> SET SESSION query_cache_size=122880;
ERROR 1229 (HY000): Variable 'query_cache_size' is a GLOBAL variable and should be set with SET GLOBAL
mysql> SET GLOBAL query_cache_size=122880;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'query_cache_size';
+------------------+--------+
| Variable_name    | Value  |
+------------------+--------+
| query_cache_size | 122880 |
+------------------+--------+
1 row in set (0.00 sec)

mysql>
```

Notez que pour une option ayant uniquement une portée globale, la modification de la valeur globale est prise en compte dans la session active.

Les modifications à chaud ne sont pas persistants.

LAB #3 - Le Mode SQL

Sous Red Hat et CentOS

Le mode SQL est utilisé principalement pour empêcher l'insertion de données invalides. La valeur par défaut du mode SQL peut être visualisée avec la commande suivante :

```
mysql> SHOW SESSION VARIABLES LIKE 'sql_mode';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| sql_mode      |       |
+---------------+-------+
1 row in set (0.01 sec)

mysql>
```

Par exemple, l'option **NO_ENGINE_SUBSTITUTION** empêche le serveur d'utiliser le moteur par défaut lors des commandes CREATE TABLE et ALTER TABLE dans le cas où le moteur demandé n'est pas disponible :

```
mysql> SHOW SESSION VARIABLES LIKE 'sql_mode';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| sql_mode      |       |
+---------------+-------+
1 row in set (0.01 sec)
```

```
mysql> USE test;
Database changed
mysql> CREATE TABLE t_federated(i int)ENGINE=FEDERATED;
Query OK, 0 rows affected, 2 warnings (0.06 sec)

mysql> SHOW WARNINGS;
+---------+------+-----------------------------------------------------+
| Level   | Code | Message                                             |
+---------+------+-----------------------------------------------------+
| Warning | 1286 | Unknown storage engine 'FEDERATED'                  |
| Warning | 1266 | Using storage engine InnoDB for table 't_federated' |
+---------+------+-----------------------------------------------------+
2 rows in set (0.00 sec)

mysql> SET SESSION sql_mode='NO_ENGINE_SUBSTITUTION';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE t_federated1(i int)ENGINE=FEDERATED;
ERROR 1286 (42000): Unknown storage engine 'FEDERATED'
mysql>
```

D'autres options existent :

- STRICT_ALL_TABLES,
- STRICT_TRANS_TABLES,
- NO_ZERO_DATE,
- NO_ZERO_IN_DATE,
- ERROR_FOR_DIVISION_BY_ZERO,
- NO_AUTO_CREATE_USER.

Renseignez-vous sur la signification de chaque valeur de l'option **sql_mode**.

Les valeurs de l'option sql_mode peuvent être combinées. Pour aider l'administrateur, MySQL propose un mode sql appelé **TRADITIONAL** :

```
mysql> SET SESSION sql_mode='TRADITIONAL';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW SESSION VARIABLES LIKE 'sql_mode';
+---------------+-----------------------------------------------------------------------------------------------------
-------------------------------------------------------+
| Variable_name | Value
|
+---------------+-----------------------------------------------------------------------------------------------------
-------------------------------------------------------+
| sql_mode      |
STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,TRADITIONAL,NO_AUTO
_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+---------------+-----------------------------------------------------------------------------------------------------
-------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Sous Windows

La valeur par défaut du mode SQL peut être visualisée avec la commande suivante :

```
mysql> SHOW SESSION VARIABLES LIKE 'sql_mode';
+---------------+-----------------------------------------------------------------+
| Variable_name | Value                                                           |
+---------------+-----------------------------------------------------------------+
| sql_mode      | STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+---------------+-----------------------------------------------------------------+
1 row in set (0.01 sec)
```

```
mysql>
```

Par exemple, l'option **NO_ENGINE_SUBSTITUTION** empêche le serveur d'utiliser le moteur par défaut lors des commandes CREATE TABLE et ALTER TABLE dans le cas où le moteur demandé n'est pas disponible :

```
mysql> USE test;
Database changed
mysql> CREATE TABLE t_federated(i int)ENGINE=FEDERATED;
ERROR 1286 (42000): Unknown storage engine 'FEDERATED'
mysql>
```

D'autres options existent :

- STRICT_ALL_TABLES,
- STRICT_TRANS_TABLES,
- NO_ZERO_DATE,
- NO_ZERO_IN_DATE,
- ERROR_FOR_DIVISION_BY_ZERO,
- NO_AUTO_CREATE_USER.

Renseignez-vous sur la signification de chaque valeur de l'option **sql_mode**.

Les valeurs de l'option sql_mode peuvent être combinées. Pour aider l'administrateur, MySQL propose un mode sql appelé **TRADITIONAL** :

```
mysql> SET SESSION sql_mode='TRADITIONAL';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW SESSION VARIABLES LIKE 'sql_mode';
+---------------+-----------------------------------------------------------------------------------------------
-------------------------------------------------+
| Variable_name | Value
```

```
|
+---------------+-----------------------------------------------------------------------------------------------
-------------------------------------------------+
| sql_mode      |
STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,TRADITINAL,NO_AUTO_
CREATE_USER,NO_ENGINE_SUBSTITUTION |
+---------------+-----------------------------------------------------------------------------------------------
-------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

SQL, Champs, Moteurs et Jointures

SQL

Chaînes de caractères

Les chaînes de caractères doivent être entourées de ' ou de **“**.

Exemples

'Linux est incroyable'

“Windows est .....”

“Je lui dit : ”“Je t'aime”””

Nombres

Il existe 3 types de nombres :

Nombres Entiers

Une séquence de chiffres **sans espaces**

Exemple

999256

0

Nombres Décimaux

Utilisent le **point** comme séparateur.

Exemple

120.54

5566.8956e+12

Nombres Négatifs

Sont précédés par le signe **-**.

Exemple

-458

-147.36

Valeurs NULL

Une chaîne **sans données**.

> N'est pas la même chose qu'une chaîne **vide** ou un **0** dans le cas d'un nombre.

Noms de Fichiers

Les noms de bases, tables et colonnes :

- peuvent contenir jusqu'à 64 caractères
- peuvent commencer par un chiffre
- ne peuvent pas contenir **que** de chiffres
- ne peuvent pas contenir un **/**, une **** ou un **point**

Les alias :

- peuvent contenir jusqu'à 255 caractères
- peuvent contenir un **/**, une **** ou un **point**

Variables Utilisateurs

Les variables n'ont pas besoin d'être **initialisées**. Elles sont automatiquement crées avec la valeur **NULL**.

Commentaires

Les commentaires **d'une ligne** sont précédés par le caractère **#** ou **- -**.

Les commentaires sous plusieurs lignes sont entourés comme suit :

```
/*
Ceci est
un commentaire
sur
plusieurs lignes
*/
```

Commandes

SELECT

Obtenir un ensemble de données à partir d'une ou de plusieurs tables.

Syntaxe

```
SELECT [table.][colonne]|expression [AS nom][,[table.[colonne]
FROM nom-table [, nom_table ...]
[WHERE condition [AND|OR condition ...]
[GROUP BY [table.][colonne],[table].[colonne] ...]
[ORDER BY [table.][colonne][Description],[table].[colonne] ...]
```

Exemples

```
SELECT nom, prenom FROM familles ORDER BY nom
```

```
SELECT nom, prenom FROM familles GROUP BY ville
```

UPDATE

Mettre à jour des données dans une table.

Syntaxe

```
UPDATE [LOW_PRIORITY][IGNORE] nom_table
SET colonne = expression
WHERE expression
ORDER BY expression
LIMIT valeur
```

Directive	Description
LOW_PRIORITY	L'opération se fera quand la table n'est pas utilisée par un client
IGNORE	La mise à jour continue malgré des problèmes éventuels rencontrés. Les enregistrements posant problème ne seront PAS mis à jour

Exemples

```
UPDATE familles SET Adresse2='*******', nb_enfants=2;
```

```
UPDATE familles SET nb_enfants=8 ORDER BY Nom LIMIT 3;
```

DELETE FROM

Supprimer des enregistrements d'une table.

Syntaxe

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM nom_table
       [WHERE where_definition]
       [ORDER BY ...]
       [LIMIT row_count]
```

Exemple

```
DELETE FROM familles WHERE Nom = Durant;
```

DROP TABLE

Syntaxe

```
DROP DATABASE [IF EXISTS] db_name
```

Supprimer une table d'une base de données.

Exemple

```
DROP TABLE Test;
```

INSERT

Syntaxe

```
INSERT [LOW_PRIORITY] | [DELAYED] [IGNORE]
[INTO] nom_table [(nom_colonne ...)]
VALUES ((expression | DEFAULT) ...) , (...)
```

ou

```
INSERT [LOW_PRIORITY] | [DELAYED] [IGNORE]
[INTO] nom_table [(nom_colonne ...)]
SELECT ...
```

Directive	Description
LOW_PRIORITY	L'opération se fera quand la table n'est pas utilisée par un client
DELAYED	Permet à un client d'utiliser la table sans attendre la fin de l'opération
IGNORE	La mise à jour continue malgré des problèmes éventuels rencontrés. Les enregistrements posant problème ne seront PAS mis à jour

INSERT VALUES

Insérer des valeurs dans une table.

Exemple

```
INSERT INTO familles (Nom, Prenom) VALUES ('Smith', 'John');
```

INSERT SELECT

Insérer des valeurs en provenance d'une autre table dans la table cible.

Exemple

```
INSERT INTO familles (Prenom) SELECT enfants.prenom FROM enfants;
```

ALTER

Ajouter, supprimer ou modifier une colonne, index ou nom de table.

ALTER ADD

Commande	Description
ADD [COLUMN]	Ajouter un champ
ADD INDEX	Ajouter un index
ADD PRIMARY KEY	Ajouter une clef primaire
ADD UNIQUE	Ajouter un index unique

Commande	Description
ADD FULLTEXT	Ajouter une recherche texte entier

Exemples

```
ALTER TABLE familles ADD Sports VARCHAR(50) NOT NULL;
```

```
ALTER TABLE familles ADD INDEX (Sports);
```

ALTER CHANGE

Permet de modifier le nom ou le type d'une colonne.

Exemple

```
ALTER TABLE familles CHANGE PrenomPere Prenom_Pere VARCHAR(45) NOT NULL;
```

ALTER DROP

Commande	Description
DROP [COLUMN]	Supprimer un champ
DROP INDEX	Supprimer un index
DROP PRIMARY KEY	Supprimer une clef primaire

Exemple

```
ALTER TABLE familles DROP Sports;
```

ALTER KEYS

Permet de désactiver la clef primaire momentanément.

Exemples

```
ALTER TABLE familles DISABLE KEYS;
```

```
ALTER TABLE familles ENABLE KEYS;
```

ALTER RENAME

Permet de renommer une table.

Exemple

```
ALTER TABLE enfants RENAME familles_enfants;
```

ALTER ORDER BY

Permet de reclasser physiquement une table.

Exemple

```
ALTER TABLE familles ORDER BY Prenom_Pere;
```

MATCH

Permet la recherche d'un mot, d'une phrase ou d'une expression sur un **texte entier**

Exemple

```
SELECT * FROM familles WHERE MATCH (Commentaire) AGAINST ('fleuve');
```

Opérateurs

Mathémathiques

Nom	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division
()	Le calcul entre parenthèses est effectué en priorité

Logiques

Nom	Description
!	NON
||	OU
&&	ET
XOR	XOR exclusif

Comparaison

Nom	Description
=	Egal
!= ou <>	Non égal
<	Inférieur
< =	Inférieur ou égal
>	Supérieur
> =	Supérieur ou égal
< = >	Egal gérant la nullité
IS NULL	Nullité
IS NOT NULL	Non nullité

Fonctions

Mathémathiques

Nom	Description
ABS(nbr)	Valeur absolue de nbr
SIGN(nbr)	-1 0 ou 1 en fonctionne du signe de nbr
MOD(nbr1, nbr2)	Indique le reste de la division de nbr1 par nbr2
FLOOR(nbr)	Indique le plus grande nombre entier inférieur à nbr
CEILING(nbr)	Indique la plus petite valeur entière supérieure à nbr
ROUND(nbr)	Indique l'arrondi entier le plus proche de nbr
ROUND(nbr,D)	Indique l'arrondi à D décimales plus plus proche de nbr
EXP(nbr)	Indique l'exponentielle de nbr
POW(nbr,nbr2)	indique nbr à la puissance nbr2
SQRT(nbr)	Indique la racine carrée de nbr
PI()	Indique la valeur de PI
COS(nbr)	Indique le cosinus de nbr
ACOS(nbr)	Indique l'arc cosinus de nbr
SIN(nbr)	Indique le sinus de nbr
ASIN(nbr)	Indique l'arc sinus de nbr
TAN(nbr)	Indique le tangent de nbr
ATAN(nbr)	Indique l'arc tangent de nbr
COT(nbr)	Indique la cotangente de nbr
RAND()	Indique une valeur en virgule flottante entre 0 et 1.0
RAND(nbr)	Indique une valeur en virgule flottante entre 0 et 1.0. La valeur de nbr indique la limite haute
LEAST(nbr1, nbr2 ...)	Indique le plus petit des nombres nbr1 à nbrX
GREATEST(nbr1, nbr2 ...)	Indique le plus grand des nombres nbr1 à nbrX
DEGREES(nbr)	Convertit nbr de gradiants vers degrés
RADIANS(nbr)	Convertit nbr de degrés vers gradiants
TRUNCATE(nbr,D)	Indique nbr tronqué à D décimales

Chaînes

Nom	Description
LIKE	Effectue une comparaison en fonction d'un motif
NOT LIKE	L'inverse de LIKE
_	Remplace un caractère dans le motif d'une chaîne
%	Remplace un ou plusieurs caractère(s) dans le motif d'une chaîne
\	Caractère d'échappement
BINARY	Rend la comparaison avec un motif sensible à la casse
STRCMP(chaîne1, chaîne2)	Compare deux chaînes. Retourne -1 si chaîne 1 < chaîne 2. Retourne 0 si chaîne 1 = chaîne 2. Retourne 1 si chaîne 1 > chaîne 2.
MATCH	Utilisé pour des recherches de texte intégral
UPPER('chaîne')	Transforme les minuscules en majuscules
LOWER('chaîne')	Transforme les majuscules en majuscules

Dates

Nom	Description
NOW()	Retourne la date au format 'AAAA-MM-JJ HH:MM:SS'
DATE_FORMAT(date,format)	Retourne la date selon le format spécifié
DAYOFWEEK(date)	Retourne un chiffre qui représente le jour de la semaine (1 pour dimanche, 7 pour samedi)
WEEKDAY(date)	Retourne un chiffre qui représente le jour de la semaine (7 pour dimanche, 0 pour lundi)
DAYOFMONTH(date)	Retourne un chiffre de 1 à 31
DAYOFYEAR(date)	Retourne un chiffre de 1 à 366
MONTH(date)	Retourne un chiffre de 1 à 12
DAYNAME(date)	Retourne le nom du jour (lundi, mardi ...)
MONTHNAME(date)	Retourne le nom du mois
QUARTER(date)	Retourne un chiffre de 1 à 4 (1 = premier trimestre de l'année)
WEEK(date [,depart]	Retourne une valeur de 1 à 52. La valeur de *depart* peut être 0 ou 1. Dans le cas de 0, le dimanche est considéré comme le premier jour de la semaine. Dans le cas de 1, c'est le lundi.
YEAR(date)	Retourne un chiffre de 1000 à 9999
HOUR(date)	Retourne l'heure
MINUTE(date)	Retourne les minutes
SECOND(date)	Retourne les secondes

Nom	Description
TO_DAYS(date)	Retourne le nombre de jours écoulés depuis le début de l'an 0
FROM_DAYS(date)	Retourne la date à partir d'un nombre de jours écoulés depuis le début de l'an 0
CURDATE()	Retourne la date courante au format AAAA-MM-JJ
CURRENT_DATE()	Retourne la date courante au format AAAA-MM-JJ
CURTIME()	Retourne la date courante au format HH:MM:SS
CURRENT_TIME()	Retourne la date courante au format HH:MM:SS
UNIX_TIMESTAMP([date])	Retourne le nombre de secondes depuis la date 1970-01-01 00:00:00
FROM_UNIXTIME(unix_timestamp[,format])	Retourne la date en fonction d'un nombre de secondes depuis la date 1970-01-01 00:00:00

Motifs

Nom	Description	Exemple
%M	Le mois	janvier
%W	Le jour de la semaine	lundi
%D	La date du mois	1st
%Y	L'année	2009
%y	L'année	09
%a	L'abréviation du jour	lun
%d	Le jour du mois	01
%m	Le numéro du mois	01
%b	L'abréviation du mois	lun
%j	Le jour de l'année	031
%H	L'heure	00
%h	L'heure	12
%i	Les minutes	05
%r	L'heure au format américain	11:01:00 PM
%T	L'heure au format 24 heures	23:01:00
%S	Les secondes	06
%p	AM ou PM	PM
%w	Le numéro du jour de la semaine	0 *(=dimanche)*

Nom	Description	Exemple
%U	Le numéro de la semaine avec le début de la semaine étant un dimanche	03
%u	Le numéro de la semaine avec le début de la semaine étant un lundi	02

Par exemple sous Red Hat et CentOS :

```
mysql> select DATE_FORMAT(now(), '%W %d %M %Y');
+-----------------------------------+
| DATE_FORMAT(now(), '%W %d %M %Y') |
+-----------------------------------+
| Friday 19 October 2012            |
+-----------------------------------+
1 row in set (0.00 sec)

mysql>
```

et sous Windows™ :

```
mysql> select DATE_FORMAT(now(), '%W %d %M %Y');
+-----------------------------------+
| DATE_FORMAT(now(), '%W %d %M %Y') |
+-----------------------------------+
| Tuesday 23 February 2016          |
+-----------------------------------+
1 row in set (0.04 sec)

mysql>
```

Contrôle

Nom	Description
IF(exp1, exp2, exp3)	Si exp1 est vrai, exp2 est retournée sinon exp3 est retournée
IFNULL(exp1,exp2)	Si exp1 est NULL, exp2 est retournée sinon exp1 est retournée

Nom	Description
NULLIF(exp1,exp2)	Si exp1 = exp2, NULL est retournée sinon exp1 est retournée
CASE value WHEN comp1 THEN res1 [WHEN comp2 THEN res2][ELSE elseres] END	La fonction compare *value* à comp1 (comp2 ...). Si une égalité est trouvée, res1 (res2 ...) est retournée. Si aucune égalité n'est trouvée, elseres est retournée

Agrégation

Nom	Description
AVG(colonne)	Moyenne de la colonne
COUNT(items)	Le nombre de valeurs non nulles de la colonne
MIN(colonne)	Valeur minimum de la colonne
MAX(colonne)	Valeur maximum de la colonne
STD(colonne)	Écart type des valeurs de la colonne
SUM(colonne)	Somme des valeurs de la colonne
BIT_OR(colonne)	Ou logique effectué sur les valeurs de la colonne
BIT_AND(colonne)	ET logique effectué sur les valeurs de la colonne

Autres

Nom	Description
CAST(expression) CONVERT(expression)	Convertit l'expression vers le type demandé - *BINARY, DATE, DATETIME, SIGNED, TIME, UNSIGNED*
LAST_INSERT_ID()	Retourne la valeur d'une colonne *AUTO_INCREMENT* lors du dernier *insertion*
VERSION()	La version du serveur MySQL
CONNECTION_ID()	L'identifiant du thread de la connexion courante
DATABASE()	La base de données courante
USER()	L'utilisateur courant
PASSWORD(chaîne)	Encrypte un mot de passe
ENCRYPT(chaîne, [,force])	Encrypte un mot de passe avec la fonction *crypt* d'Unix
ENCODE(chaîne,mdp)	Encode la chaîne avec le mot de passe *mdp*
DECODE(chaîne,mdp)	Décode la chaîne avec le mot de passe *mdp*
MD5(chaîne)	Retourne une chaîne encodée à la norme MD5
SHA1()	Retourne une chaîne encodée à la norme SHA1

Types de Champs

Nombres entiers

Type	Intervalle	Taille en octets	Description
TINYINT[(M)]	-127 à 128	1	Entiers très courts
TINYINT[(M)] UNSIGNED	0 à 255	1	Entiers très courts
SMALLINT[(M)]	-32 768 à 32 767	2	Entiers courts
SMALLINT[(M)] UNSIGNED	0 à 65 535	2	Entiers courts
MEDIUMINT[(M)]	- 8 388 608 à 8 388 607	3	Entiers de taille moyenne
MEDIUMINT[(M)] UNSIGNED	0 16 777 215	3	Entiers de taille moyenne
INT[(M)]	-2^{31} à $2^{31} - 1$	4	Entiers
INT[(M)] UNSIGNED	0 à $2^{32} - 1$	4	Entiers
INTEGER[(M)]	-	-	Synonyme de INT
BIGINT[(M)]	-2^{63} à $2^{63} - 1$	8	Entiers larges
BIGINT[(M)] UNSIGNED	0 à $2^{64} - 1$	8	Entiers larges

Nombres à virgule flottante

Type	Intervalle	Taille en octets	Description
FLOAT(precision)	Varie selon *precision*	Varie	< =24 pour un nombre en simple précision ou >24 et < =53 pour un nombre ne double précision
FLOAT[(M,D)]	+ ou - 1.175494351E-38 à + ou - 3.402823466E+38	4	Simple précision
DOUBLE[(M,D)]	+ ou - 1.7976931348623157E+308 à + ou - 2.2250738585072014E-308	8	Double précision
DOUBLEPRECISION[(M,D)]	-	-	Synonyme de DOUBLE[(M,D)]
REAL[(M,D)]	-	-	Synonyme de DOUBLE[(M,D)]
DECIMAL[(M[,D])]	Varie	M +2	Nombre à virgule flottante
NUMERIC	-	-	Synonyme de DECIMAL

Dates et Heures

Type	Intervalle
DATE	1000-01-01 à 9999-12-31
TIME	-838:59:59 à 838:59:59
DATETIME	1000-01-01 00:00:00 à 9999-12-31 23:59:59
TIMESTAMP[(M)]	Voir tableau ci-dessous
YEAR[(2)]	70 à 69 (1970 à 2069)
YEAR[(4)]	1901 à 2155

Types de données TIMESTAMP

Type	Affichage
TIMESTAMP	YYYYMMDDHHMMSS
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

Chaînes

Type	Longeur	Description
CHAR(M)	1 à 255	Chaîne de longueur fixe où M varie entre 1 et 255
VARCHAR(M)	1 à 255	Chaîne de longueur variable où M varie entre 1 et 255

TEXT et BLOB

- **TEXT**,
  - TEXT respecte la casse des données,

- **BLOB** (Binary Large OBject),
  - BLOB permet de stocker toute information binaire telle une image ou un son.

Type	Longueur Maximale en caractères	Description
TINYBLOB	255	Objet binaire court
TINYTEXT	255	Texte court
BLOB	65 535	Objet binaire de taille normale
TEXT	65 535	Texte de taille normale
MEDIUM BLOB	16 777 215	Objet binaire de taille moyenne
MEDIUM TEXT	16 777 215	Texte de taille moyenne
LONG BLOB	4 294 967 295	Objet binaire de grande taille
LONG TEXT	4 294 967 295	Texte de grande taille

ENUM et SET

- **ENUM**,
  - permet de prendre une valeur ou NULL parmi une liste de valeurs prédéfinies à la création de la colonne,
- **SET**
  - permet de prendre un maximum de 64 valeurs ou NULL parmi une liste de valeurs prédéfinies à la création de la colonne,

Type	Maximum des valeurs dans l'ensemble	Description
ENUM('valeur1','valeur2',...)	65 535	Les valeurs de ce type ne peuvent contenir qu'une seule des valeurs de la liste ou NULL
SET('valeur1','valeur2',...)	64	Les valeurs de ce type ne peuvent contenir un ensemble des valeurs de la liste ou NULL

Types de Moteurs de Stockage

Le type de moteur est spécifié lors de la création de la table. Le moteur d'une table existant peut être modifié avec la commande **ALTER TABLE**. Les différents types **principaux** sont résumés ci-après :

Type de Moteur	Description
InnoDB	Le format par défaut. Supportent les transactions sécurisées avec **COMMIT** et **ROLLBACK**. Permet de verrouiller des enregistrements un-à-un au lieu de la table entière.

Type de Moteur	Description
MyISAM	Données stockées dans un fichier .MYD. Index stockés dans un fichier .MYI. Structure da la table stockée dans un fichier .frm
MEMORY	Anciennement connues sous le nom HEAP. Les données sont stockées en mémoire tandis que la structure de la table est stockée sur disque dans un fichier .frm. Ne supportent pas les champs TEXT, BLOB. Ne supportent pas l'attribut AUTO_INCREMENT. Utilisé pour augmenter la vitesse de traitement d'une requête.
ARCHIVE	Ne supportent que des requêtes de lecture et d'insertion. Utilisé pour stocker des données des journaux d'application. Structure stockée dans un fichier .frm. Données stockées dans un fichier .ARZ. Métadonnées sont stockées dans un fichier .ARM
CSV	Données sont stockées dans un fichier texte au format CSV. Utilisé pour échanger des données avec d'autres applications. Structure stockée dans un fichier .frm. Données stockées dans un fichier .CSV.
FEDERATED	Ne stocke pas de données. Extrait des données de tables en provenance de bases sur des serveurs distants.

Bien que possible techniquement, il n'est pas préférable d'utiliser des types de moteurs différents dans la même base de données car ceci complique la tâche d'optimisation, augmente le nombre de buffers et de caches et complexifie le travail de l'administrateur.

Le choix d'un moteur de stockage se fait en fonction de l'adéquation entre les besoins et les caractéristiques suivants :

Moteur	Verrous	Transactionnel de type ACID	MVCC	Clef Etrangère	Données sur Disque	Sauvegarde à Chaud	COMMIT	ROLLBACK	Crash Recovery
InnoDB	Enregistrement	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
MyISAM	Table	Non	Non	Non	Oui	Non	Non	Non	Non
Memory	Table	Non	Non	Non	Non	Non	Non	Non	Non
Archive	Enregistrement	Non	Non	Non	Oui	Non	Non	Non	Non
CSV	Table	Non	Non	Non	Oui	Non	Non	Non	Non
Federated	Table	Non	Non	Non	Oui	Non	Non	Non	Non
NDB Cluster	Enregistrement	Oui	Non	Oui	Oui	Oui	Oui	Oui	Oui

Caractéristiques des Moteurs

InnoDB

Le moteur InnoDB est le moteur par défaut de MySQL depuis la version 5.5. Les principaux caractéristiques du moteur InnoDB sont :

- Il est transactionnel de type **ACID** (Atomicité, Cohérence, Isolation, Durabilité),
- Il implémente le **MVCC** (Multi Version Concurrency Control) qui permet d'avoir des lectures qui ne bloquent pas des écritures et inversement,
- Il implémente un verrou au niveau enregistrement,
- Il implémente un mécanisme de restauration automatique en utilisant des journaux de transactions **ib_logfile*** ou * est **0**, **1**, **2** etc,
- La structure de la table est stockée dans un fichier .frm,
- Il supporte l'utilisation de clefs étrangères,
- Il peut être sauvegardé à chaud,
- Il est recommandé pour des applications avec beaucoup d'écritures.

Mécanisme Interne

Les transactions en cours sont stockées en mémoire dans le **buffer des journaux de transactions** (*log_buffer*). La taille du buffer est fixée par la valeur de l'option **innodb_log_buffer_size**. Chaque seconde les informations dans le buffer sont écrites sur disque dans l'un des deux journaux des transactions **ib_logfile0** ou **ib_logfile1**. Le nombre de journaux des transactions est fixé par la valeur de l'option **innodb_log_files_in_group**. La taille des logs ib_logfile*, aussi appelés le **Redo log**, est fixée par l'option **innodb_log_file_size** mais ne peut pas dépasser 512 Go. Le buffer est aussi écrit sur disque lors de chaque **commit** (validation) sauf si ce mécanisme a été modifié par la valeur de l'option **innodb_flush_log_at_trx_commit**.

Une partie des données ainsi que les index sont stockés en mémoire dans le **buffer pool**. La taille du buffer est fixée par la valeur de l'option **innodb_buffer_pool_size**. Il faut éviter que la taille de cette mémoire tampon dépasse les 80 % de la RAM du serveur. Chaque seconde, les informations dans le buffer sont écrites sur disque dans le tablespace. L'emplacement du tablespace est fixé par la valeur de l'option **innodb_data_file_path**.

Transactions

Le moteur Innodb est transactionnel. Ceci implique que le moteur collectionne des requêtes jusqu'à la commande **COMMIT**. A ce moment le moteur applique les modifications à toutes les tables concernées **en même temps**. Soit toutes les modifications réussissent, soit il y a un **ROLLBACK**. Par défaut, MySQL utilise un autocommit avec le moteur Innodb. Ceci implique qu'à la fin de chaque requête SQL, le serveur voit un commit virtuel.

Il est possible de modifier ce comportement automatique en utilisant la commande **BEGIN**. Par exemple :

```
mysql> BEGIN;
mysql> INSERT INTO table (colonne) VALUES (value);
mysql> COMMIT;
```

Si la dernière commande est remplacée par la commande ROLLBACK la modification apportée par la requête INSERT INTO sera effacée :

```
mysql> BEGIN;
mysql> INSERT INTO table (colonne) VALUES (value);
mysql> ROLLBACK;
```

Tablespace

Depuis la version 5.6.6 de MySQL, il existe l'option **innodb_file_per_table**. Si cette option est **0**, Innodb se comporte comme les versions précédentes, à savoir les données et les index de toutes les tables sont centralisées dans la **Tablespace**, c'est-à-dire, dans le fichier **ibdata** par défaut. Avec une valeur de **1**, les données et index de chaque table sont stockées dans un fichier individuel ayant une extension **.ibd** se trouvant dans le répertoire du schéma à côté du fichier de définition de la structure, le **.frm**.

Cette modification permet le stockage des fichiers .ibd sur un autre espace de stockage physique. De cette façon, il est possible d'utiliser des espaces de stockage très rapides.

Malgré l'utilisation des fichiers .ibd, il est toujours necéssaire d'avoir le fichier **ibdata** car ce dernier stocke le **dictionnaire des données** qui est une copie de tous les fichiers .frm ainsi que les Redo logs. Les Redo logs journalisent toute requête de modification des objets.

> Malgré l'utilisation des fichiers **.ibd**, il n'est pas possible de manipuler les fichiers dans un gestionnaire de fichiers. Ne les déplacez pas, ne les copiez pas sous peine de perdre des données.

> La quantité de mémoire libre dans un fichier *.ibd ou dans la Tablespace est indiquée par la sortie de la commande **SHOW TABLE STATUS**. Cette valeur est en pages et chaque page vaut 16 Ko. Bien que vous puissiez libérer de l'espace mémoire utilisée dans la

Tablespace en passant la valeur de l'option **innodb_file_per_table** de **0** à **1** et en saisissant la commande **ALTER TABLE table ENGINE=InnoDB** pour chaque table, la taille du fichier ibdata ne diminuera pas. La seule façon de réduire la taille de ce fichier est d'exporter la base de données avec mysqldump puis de recréer la base de données à partir du dump.

Multiversion Concurrency Control

Chaque table Innodb contient deux colonnes cachées :

- le numéro de transaction - **txn#**,
- un pointeur vers la version précédente de la ligne qui se trouve dans le Undo log. Ce pointeur s'appelle le **Pointeur ROLLBACK**.

Quand une ligne est modifiée, l'ancienne version de la ligne y compris son numéro de transaction est copiée dans le Undo log. La nouvelle version de la ligne y compris son numéro de transaction est copiée dans la Tablespace ou dans le fichier .ibd. Le pointeur de la nouvelle ligne identifie l'ancienne ligne dans le Undo log. Ce processus s'appelle le **MVCC** (*Multiversion Concurrency Control*).

Transaction Isolation Levels

Créez la base de données **nombres** :

```
mysql> CREATE DATABASE `Nombres` ;
Query OK, 1 row affected (0.01 sec)

mysql> USE Nombres;
Database changed

mysql> CREATE TABLE english (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Number VARCHAR(10));
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO english (Number) VALUES ('One');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO english (Number) VALUES ('Two');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO english (Number) VALUES ('Three');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | One    |
|  2 | Two    |
|  3 | Three  |
+----+--------+
3 rows in set (0.00 sec)

mysql>
```

Read Uncomitted

Mettez à jour l'enregistrement 1 sans faire de COMMIT ou de ROLLBACK :

```
mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE english SET Number = "Un" WHERE id = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | Un     |
```

```
|  2 | Two    |
|  3 | Three  |
+----+--------+
3 rows in set (0.00 sec)

mysql>
```

Ouvrez une autre terminal, connectez-vous à mysql et visualisez la modification :

```
[root@node01 ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE Nombres;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | One    |
|  2 | Two    |
```

```
|  3 | Three  |
+----+--------+
3 rows in set (0.01 sec)

mysql>
```

Notez que le deuxième client voit l'ancienne valeur de l'enregistrement 1. Ceci est parce que la base de données a suivi le Pointeur ROLLBACK vers le Undo log pour afficher la valeur de la colonne. Notez que **MVCC** ne bloque pas la lecture malgré le fait qu'il y a une transaction en cours !

Pour voir la valeur dans la Tablespace, il faut modifier la valeur de l'option TRANSACTION ISOLATION LEVEL :

```
mysql> SET TRANSACTION ISOLATION LEVEL read uncommitted;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | Un     |
|  2 | Two    |
|  3 | Three  |
+----+--------+
3 rows in set (0.01 sec)

mysql>
```

Dans ce mode, le client voit une donnée fictive car il n'y a pas encore eu de COMMIT !

Read Comitted

Dans ce mode de fonctionnement InnoDB suit les Pointeurs ROLLBACK jusqu'à il trouve la dernière version ayant subi un COMMIT :

```
mysql> SET TRANSACTION ISOLATION LEVEL read committed;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | One    |
|  2 | Two    |
|  3 | Three  |
+----+--------+
3 rows in set (0.01 sec)

mysql>
```

Retournez au premier terminal :

```
mysql> COMMIT;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Retournez au deuxième terminal :

```
<mysql> SELECT * FROM english;
+----+--------+
| id | Number |
+----+--------+
|  1 | Un     |
|  2 | Two    |
|  3 | Three  |
+----+--------+
```

```
3 rows in set (0.00 sec)

mysql>
```

MyISAM

L'architecture du moteur MyISAM est plus simple que le moteur InnoDB. Les principaux caractéristiques du moteur MyISAM sont :

- Il n'est pas transactionnel,
- Il implémente un verrou au niveau table,
- Il n'y a pas de recouvrement automatique des données lors d'un crash,
- La structure de la table est stockée dans un fichier .frm,
- Les données sont stockées dans un fichier .MYD tandis que les index sont stockés dans un fichier .MYI,
- Il ne supporte pas de clefs étrangères,
- Il ne peut pas être sauvegardé à chaud,
- Il peut être compressé,
- Il supporte les index plain text (fulltext),
- Il est recommandé pour des applications avec beaucoup de lectures.

Il existe trois types de moteurs MyISAM : FIXED , DYNAMIC et COMPRESSED :

MyISAM FIXED

Une table est FIXED si elle ne contient aucun champ de type VARCHAR, VARBINARY, TEXT ou BLOB ou si la valeur de l'option **row_format** est FIXED (row_format=FIXED). Tous les enregistrements sont de la même taille. De ce fait, les tables sont plus rapides et plus robustes que les tables dynamiques mais elles prennent plus d'espace disque. Les données de la table ne peuvent pas être fragmentées. Une table devient FIXED quand la valeur de l'option **row_format** est FIXED (row_format=FIXED) :

```
mysql> USE mysql;
Database changed
mysql> SHOW TABLE STATUS LIKE 'db' \G;
*************************** 1. row ***************************
```

```
           Name: db
         Engine: MyISAM
        Version: 10
     Row_format: Fixed
           Rows: 2
 Avg_row_length: 440
    Data_length: 880
Max_data_length: 123848989752688639
   Index_length: 5120
      Data_free: 0
 Auto_increment: NULL
    Create_time: 2014-01-22 15:28:03
    Update_time: 2014-01-22 15:28:04
     Check_time: 2014-01-22 15:28:04
      Collation: utf8_bin
       Checksum: NULL
 Create_options:
        Comment: Database privileges
1 row in set (0.01 sec)

ERROR:
No query specified

mysql>
```

MyISAM DYNAMIC

Une table est DYNAMIC si elle contient au moins un champ de type VARCHAR, VARBINARY, TEXT ou BLOB ou la valeur de l'option **row_format** est DYNAMIC (row_format=DYNAMIC). Les données de la table peuvent être fragmentées. Dans ce cas, l'utilisation de la commande OPTIMIZE TABLE ou **myisamchk** est nécessaire :

```
mysql> SHOW TABLE STATUS LIKE 'proc' \G;
*************************** 1. row ***************************
```

```
           Name: proc
         Engine: MyISAM
        Version: 10
     Row_format: Dynamic
           Rows: 0
 Avg_row_length: 0
    Data_length: 0
Max_data_length: 281474976710655
   Index_length: 2048
      Data_free: 0
 Auto_increment: NULL
    Create_time: 2014-01-22 15:28:04
    Update_time: 2014-01-22 15:28:04
     Check_time: NULL
      Collation: utf8_general_ci
       Checksum: NULL
 Create_options:
        Comment: Stored Procedures
1 row in set (0.01 sec)

ERROR:
No query specified

mysql>
```

MyISAM COMPRESSED

Après avoir arrêté le serveur, la compression est obtenue en utilisant la commande **myisampack** suivi de la commande **myisamchk -rq**. Le décompression est obtenue en utilisant la commande **myisamchk –unpack**. La compression peut atteindre les 70% maximum. Une fois compressée les requêtes UPDATE, DELETE et INSERT ne peuvent pas être utilisées :

```
[root@centos6 ~]# myisampack /var/lib/mysql/mysql/help_keyword.MYI
Compressing /var/lib/mysql/mysql/help_keyword.MYD: (467 records)
```

```
- Calculating statistics
- Compressing file
95.21%
Remember to run myisamchk -rq on compressed tables

[root@centos6 ~]# myisamchk -rq /var/lib/mysql/mysql/help_keyword.MYI
- check record delete-chain
- recovering (with sort) MyISAM-table '/var/lib/mysql/mysql/help_keyword.MYI'
Data records: 467
- Fixing index 1
- Fixing index 2

[root@centos6 ~]# myisamchk --unpack /var/lib/mysql/mysql/help_keyword.MYI
- recovering (with sort) MyISAM-table '/var/lib/mysql/mysql/help_keyword.MYI'
Data records: 467
- Fixing index 1
- Fixing index 2
```

Particularités

- Le moteur MyISAM est particulièrement portable car il suffit de copier les fichiers *.frm, *.MYD et *.MYI pour obtenir une sauvegarde physique,
- Le nombre d'enregistrements est stocké dans la table,
- Les index non-uniques peuvent être désactivés momentanément,
- Il est possible d'avoir des INSERT et SELECT simultanés en utilisant l'option **concurrent_insert**.

Memory

Les principaux caractéristiques du moteur Memory sont :

- Il n'est pas transactionnel,
- Il implémente un verrou au niveau table,
- Il n'y a pas de recouvrement automatique des données lors d'un crash,

- La structure de la table est stockée dans un fichier .frm,
- Les données sont stockées dans la mémoire,
- Il ne supporte pas de clefs étrangères,
- Il ne peut pas être sauvegardé à chaud,
- Il est recommandé pour des applications ayant besoin de la vitesse.

Particularités

- La consommation de la mémoire est fixée par les options **MAX_ROWS** et **max_heap_table_size**,
- Le moteur peut implémenter soit l'algorithme **HASH** pour les index pour augmenter la performance pour les recherches d'égalité soit l'algorithme **BTREE** pour les index pour augmenter la performance pour les recherches d'inégalité,
- Au démarrage du serveur les tables sont évidement vides. Il est possible de les pré-remplir grâce à des instructions SQL contenues dans un fichier référencé par l'option **init-file**.

Archive

Les principaux caractéristiques du moteur Archive sont :

- Il n'est pas transactionnel,
- Il implémente un verrou au niveau enregistrement,
- Il n'y a pas de recouvrement automatique des données lors d'un crash,
- Les données sont stockées en les compressant,
- Il ne supporte pas de clefs étrangères,
- Il ne peut pas être sauvegardé à chaud,
- Il ne supporte pas d'Index,
- Il est recommandé pour réduire l'espace disque utilisé par les tables.

Particularités

- Le moteur ne permet que des requêtes INSERT et SELECT.

CSV

Les principaux caractéristiques du moteur CSV sont :

- Il n'est pas transactionnel,
- Il implémente un verrou au niveau table,
- Il n'y a pas de recouvrement automatique des données lors d'un crash,
- La structure de la table est stockée dans un fichier .frm,
- Les données sont séparées par des **;** et stockées dans un fichier .CSV tandis que les informations concernant l'état de la table ainsi que le nombre d'enregistrements sont stockées dans un fichier .CSM,
- Il ne supporte pas de clefs étrangères,
- Il ne peut pas être sauvegardé à chaud,
- Il ne supporte pas d'Index,
- Il est recommandé pour des tables ayant besoin d'être manipulées par des logiciels tiers.

FEDERATED

Les principaux caractéristiques du moteur FEDERATED sont :

- Il n'est pas transactionnel,
- Il n'y a pas de recouvrement automatique des données lors d'un crash,
- Il ne supporte pas de clefs étrangères,
- Il ne peut pas être sauvegardé à chaud,
- Il ne supporte pas d'Index,
- Il supporte des requêtes SELECT, INSERT, UPDATE, DELETE, TRUNCATE et DROP TABLE,
- Il n'utilise pas le cache de requêtes,
- Il est utilisé pour accéder à des données localisées sur un serveur **MySQL** distant.

NDB Cluster

Le moteur NDB Cluster est le moteur de MySQL Cluster. Les principaux caractéristiques du moteur NDB Cluster sont :

- Il est transactionnel de type **ACID** (Atomicité, Cohérence, Isolation, Durabilité),
- Il implémente le **MVCC** (Multi Version Concurrency Control) qui permet d'avoir des lectures qui ne bloquent pas des écritures et inversement,
- Il implémente un verrou au niveau enregistrement,
- Il implémente la réplication synchrone,
- Il implémente le basculement automatique sur un autre nœud en cas de panne et la synchronisation automatique du nœud à son démarrage,
- Il peut être sauvegardé à chaud,
- Il permet le dimensionnement des requêtes de lecture et d'écriture,
- Il utilise l'architecture **Shared Nothing** où les nœuds ne partagent pas un disque de données,
- Il implémente l'absence du **SPOF** (*Single Point Of Failure*).

Autres Moteurs Non Standards

XtraDB

XtraDB est le moteur de la société **Percona**. Il est basé sur InnoDB et a pour but d'être plus performant. Les principaux caractéristiques du moteur XtraDB sont :

- Il est transactionnel de type **ACID** (Atomicité, Cohérence, Isolation, Durabilité),
- Il implémente le **MVCC** (Multi Version Concurrency Control) qui permet d'avoir des lectures qui ne bloquent pas des écritures et inversement,
- Il implémente un verrou au niveau enregistrement,
- Il implémente un mécanisme de restauration automatique,
- Il supporte l'utilisation de clefs étrangères,
- Il peut être sauvegarder à chaud,
- Le moteur peut implémenter soit l'algorithme **HASH** pour les index pour augmenter la performance pour les recherches d'égalité soit l'algorithme **B+TREE** pour les index pour augmenter la performance pour les recherches d'inégalité.

Aria

Aria est le moteur de **Michael Widenius**. Il a été créé pour être un alternatif au moteur MyISAM. Les principaux caractéristiques du moteur Aria sont :

- En version 2.0, il est transactionnel de type **ACID** (Atomicité, Cohérence, Isolation, Durabilité),

- Il implémente un verrou au niveau enregistrement,
- Il implémente un mécanisme de restauration automatique.

Jointures

Les jointures permettent de créer un lien entre deux tables. Le champ utilisé pour la jointure dans la première table est souvent la **clef primaire**. Le champs utilisé dans la deuxième table est appelé le **clef étrangère**. La clef primaire et la clef étrangère doivent être du même **type**.

Dans le cas où on utilise un champs autre que la clef primaire pour assurer la jointure, cet autre champs doit être **indexé**.

FULL JOIN

Ce type de jointure permet de renvoyer uniquement les enregistrements des **deux** tables 1 et 2 ayant une correspondance :

```
mysql> SELECT table1.champ1, table2.champ1 FROM table1, table2 WHERE table1.clefprimaire = table2.clefetrangere;
```

ou

```
mysql> SELECT table1.champ1, table2.champ1 FROM table1 JOIN table2 WHERE table1.clefprimaire =
table2.clefetrangere;
```

LEFT JOIN

Ce type de jointure permet de renvoyer les enregistrements de la première table qui ont une correspondance dans la deuxième table. La syntaxe est la suivante :

```
mysql> SELECT table1.champ1, table2.champ1 FROM table1 LEFT JOIN table2 ON table1.clefprimaire =
table2.clefetrangere;
```

Le résultat de cette requête est l'affichage de **tous** les enregistrements de la **table 1** avec ou sans les enregistrements correspondants de la **table 2**.

Dans le cas où la table 2 ne contient pas d'enregistrement correspondant un un enregistrement de la table 1, la valeur retournée est **NULL**. Cette information nous permet de rechercher uniquement les enregistrements dans la table1 n'ayant **pas** d'enregistrements dans la table 2 :

```
mysql> SELECT table1.champ1, table2.champ1 FROM table1 LEFT JOIN table2 ON table1.clefprimaire =
table2.clefetrangere WHERE table2.clefetrangere IS NULL;
```

RIGHT JOIN

Ce type de jointure est l'inverse d'un LEFT JOIN.

LAB #4 - Le Langage SQL

Créez maintenant la base de données **CarnetAdresses** :

```
mysql> CREATE DATABASE `CarnetAdresses` ;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Créez ensuite deux tables **familles** et **enfants** dans la base **CarnetAdresses** :

```
mysql> USE CarnetAdresses ;
Database changed
mysql>
```

```
mysql> CREATE TABLE familles (CodeFamille BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY, Nom VARCHAR(40),
    -> PrenomPere VARCHAR(40) , PrenomMere VARCHAR(40) , Adresse1 VARCHAR(40) , Adresse2 VARCHAR(40),
    -> CodePostal VARCHAR(5), Ville VARCHAR(40), ProfPere VARCHAR(40), ProfMere VARCHAR(40));
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Pour plus de facilité, copiez simplement la requête et collez-la dans votre terminal :

familles

```
CREATE TABLE familles (CodeFamille BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY, Nom VARCHAR(40),\\
PrenomPere VARCHAR(40) , PrenomMere VARCHAR(40) , Adresse1 VARCHAR(40) , Adresse2 VARCHAR(40),\\
CodePostal VARCHAR(5), Ville VARCHAR(40), ProfPere VARCHAR(40), ProfMere VARCHAR(40));
```

```
mysql> CREATE TABLE Enfants ( CodeFamille BIGINT NOT NULL, Prenom VARCHAR(40) NOT NULL,
    -> Sexe VARCHAR(1) NOT NULL, DateNaissance DATE NOT NULL, PRIMARY KEY (CodeFamille));
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Pour plus de facilité, copiez simplement la requête et collez-la dans votre terminal :

enfants

```
CREATE TABLE Enfants ( CodeFamille BIGINT NOT NULL, Prenom VARCHAR(40) NOT NULL,\\
Sexe VARCHAR(1) NOT NULL, DateNaissance DATE NOT NULL, PRIMARY KEY (CodeFamille));
```

Utilisez ensuite la commande SHOW pour visualiser le résultat de chaque instruction, par exemple :

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CarnetAdresses     |
| mysql              |
| test               |
```

```
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> SHOW TABLES FROM CarnetAdresses;
+--------------------------+
| Tables_in_CarnetAdresses |
+--------------------------+
| Enfants                  |
| familles                 |
+--------------------------+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> SHOW COLUMNS FROM familles FROM CarnetAdresses;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| CodeFamille | bigint(20)  | NO   | PRI | NULL    | auto_increment |
| Nom         | varchar(40) | YES  |     | NULL    |                |
| PrenomPere  | varchar(40) | YES  |     | NULL    |                |
| PrenomMere  | varchar(40) | YES  |     | NULL    |                |
| Adresse1    | varchar(40) | YES  |     | NULL    |                |
| Adresse2    | varchar(40) | YES  |     | NULL    |                |
| CodePostal  | varchar(5)  | YES  |     | NULL    |                |
| Ville       | varchar(40) | YES  |     | NULL    |                |
| ProfPere    | varchar(40) | YES  |     | NULL    |                |
| ProfMere    | varchar(40) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> SHOW INDEX FROM familles;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed |
Null | Index_type | Comment |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+
| familles |          0 | PRIMARY  |            1 | CodeFamille | A         |           0 |     NULL | NULL   |
| BTREE      |         |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+--
----+------------+---------+
1 row in set (0.00 sec)

mysql>
```

```
mysql> SHOW TABLE STATUS FROM CarnetAdresses;
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
-+------------+----------------+---------------------+-------------+------------+-------------------+----------+--
--------------+---------+
| Name     | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length
| Data_free | Auto_increment | Create_time         | Update_time | Check_time | Collation         | Checksum |
Create_options | Comment |
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
-+------------+----------------+---------------------+-------------+------------+-------------------+----------+--
--------------+---------+
| Enfants  | InnoDB |      10 | Compact    |    0 |              0 |       16384 |               0 |           0
|   9437184 |           NULL | 2014-01-27 10:46:10 | NULL        | NULL       | latin1_swedish_ci |     NULL |
|         |
| familles | InnoDB |      10 | Compact    |    0 |              0 |       16384 |               0 |           0
|   9437184 |              1 | 2014-01-27 10:45:39 | NULL        | NULL       | latin1_swedish_ci |     NULL |
|         |
+----------+--------+---------+------------+------+----------------+-------------+-----------------+------------
```

```
-+-----------+----------------+---------------------+-------------+-----------+-------------------+----------+--
-------------+---------+
2 rows in set (0.00 sec)

mysql>
```

Créez les enregistrements dans la table **familles** :

data

```
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Durant', 'Jacques', 'Jane', '23 rue Dutor','','92200', 'Neuilly', 'Plombier',''); INSERT
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Tortua', 'Benoît', 'Carole', '7 rue Verget','', '75005', 'Paris', 'Cadre', 'Cadre');
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Portier', 'Pierre', 'Elisabeth', '5 rue Toulet','', '92200', 'Neuilly', 'Dentiste','');
INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Renault', 'Damien', 'Anne', '2 rue Ragon','', '75007', 'Paris', 'Comptable',
'Enseignante'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal ,
Ville , ProfPere , ProfMere) VALUES ('Darduet', 'Jean','','','','','','',''); INSERT INTO familles (Nom ,
PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Rodier',
'Gérard', 'Aurélie', '6 rue Agien', '77000', 'Fontainebleau', 'Professeur',''); INSERT INTO familles (Nom ,
PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES ('Tarte',
'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier', 'Cuisinière'); INSERT INTO familles
(Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere , ProfMere) VALUES
('Cohen', 'David', 'Sarah', '7 Av d\'Eylau','', '75016', 'Paris', 'PDG', 'Assistante Direction'); INSERT
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Dupont2', 'Bruno', 'Odile', '12 rue Sébastien','', '75008', 'PARIS', 'Electricien',
'Comptable'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville
, ProfPere , ProfMere) VALUES ('Durand2', 'Jacques', 'Jane', '23 rue Dutor','', '92200', 'Neuilly',
'Plombier',''); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal ,
Ville , ProfPere , ProfMere) VALUES ('Darmon2', 'Bruno', 'Béatrice', '2 rue Nicolo','', '13008',
'Marseille', 'Cadre', 'Peintre'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2
, CodePostal , Ville , ProfPere , ProfMere) VALUES ('Darduet2', 'Jean', '','' ,'' ,'' ,'','','',''); INSERT
```

```
INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal , Ville , ProfPere ,
ProfMere) VALUES ('Tarte2', 'Alla', 'Crème', '1 allée Durond','', '92200', 'Neuilly', 'Cuisinier',
'Cuisinière'); INSERT INTO familles (Nom , PrenomPere , PrenomMere , Adresse1 , Adresse2 , CodePostal ,
Ville , ProfPere , ProfMere) VALUES ('Cohen2', 'David', 'Sarah', '7 Av d\'Eylau','', '75016', 'Paris',
'PDG', 'Assistante Direction');
```

Saisissez maintenant chacune des commandes suivantes et expliquez le résultat obtenu. Les instructions sont numérotées pour faciliter vos prises de notes.

1	ALTER TABLE Enfants ADD INDEX (Prenom);
2	ALTER TABLE Enfants DROP INDEX Prenom;
3	ALTER TABLE Enfants ADD TelPort VARCHAR(50) NOT NULL;
4	ALTER TABLE Enfants CHANGE TelPort TelPort VARCHAR(22) NOT NULL;
5	ALTER TABLE Enfants ADD Test VARCHAR(40) NOT NULL;
6	ALTER TABLE Enfants DROP Test;
7	SELECT * FROM familles;
8	SELECT * FROM familles LIMIT 8,12;

LIMIT - Avec un argument, la valeur spécifie le nombre de lignes à retourner depuis le début du jeu de résultat. Si deux arguments sont donnés, le premier indique le décalage du premier enregistrement à retourner, le second donne le nombre maximum d'enregistrement à retourner. Le décalage du premier enregistrement est 0 (pas 1).

9	UPDATE familles SET Adresse1 = '120 rue de Vaugirard', CodePostal = '75015', Ville = 'PARIS' WHERE Nom = 'DARDUET' AND PrenomPere = 'Jean' LIMIT 1;
10	DELETE FROM familles WHERE Nom = 'DURANT' AND PrenomPere = 'Jacques';
11	DELETE FROM familles WHERE Ville = 'Neuilly';
12	SELECT * FROM familles LIMIT 8,12;
13	SELECT nom, PrenomPere FROM familles;
14	SELECT Nom AS 'Nom de famille', PrenomPere AS 'Prenom du Père' FROM familles;
15	INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Georges', 'M', '14');

```
16 SELECT familles.nom, familles.PrenomPere FROM familles, Enfants;
```

Créez la colonne **familles.nb_enfants**.

```
17 UPDATE familles SET nb_enfants = '2' WHERE CodeFamille = '2';
18 UPDATE familles SET nb_enfants = '1' WHERE CodeFamille = '4';
19 UPDATE familles SET nb_enfants = '3' WHERE CodeFamille = '5';
20 UPDATE familles SET nb_enfants = '2' WHERE CodeFamille = '6';
21 UPDATE familles SET nb_enfants = '4' WHERE CodeFamille = '8';
22 UPDATE familles SET nb_enfants = '5' WHERE CodeFamille = '9';
23 UPDATE familles SET nb_enfants = '3' WHERE CodeFamille = '11';
24 UPDATE familles SET nb_enfants = '1' WHERE CodeFamille = '12';
25 UPDATE familles SET nb_enfants = '5' WHERE CodeFamille = '14';
26 SELECT SUM(nb_enfants) FROM familles;
27 SELECT MIN(nb_enfants) AS 'Nb enfants minimum', MAX(nb_enfants) AS 'Nb enfants maximum', AVG(nb_enfants) AS 'Nb enfants moyen' FROM familles;
28 SELECT nom FROM familles WHERE ville = 'Paris';
29 SELECT nom FROM familles WHERE ville = 'Paris' OR ville = 'Neuilly';
30 ALTER TABLE Enfants DROP PRIMARY KEY;
31 INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Alex', 'F', '12');
32 INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Mila', 'F', '2');
33 INSERT INTO Enfants (Prenom, Sexe, CodeFamille) VALUES ('Amandine', 'F', '5');
34 SELECT familles.Nom , Enfants.Prenom FROM familles, Enfants WHERE ( Enfants.CodeFamille = familles.CodeFamille );
35 SELECT familles.nom, familles.PrenomPere FROM familles, Enfants;
36 SELECT ville, min(nb_enfants) as 'Nb enfants minimum', max(nb_enfants) as ' Nb enfants maximum', avg(nb_enfants) as 'Nb enfants moyen' from familles GROUP BY Ville;
37 SELECT * FROM familles ORDER BY nom;
38 SELECT * FROM familles GROUP BY ville, Nom;
39 UPDATE familles SET Adresse2 = '-';
40 UPDATE familles SET Adresse2 = '___',nb_enfants=2;
```

41	UPDATE familles SET nb_enfants=4 WHERE Ville = 'Paris';
42	UPDATE familles SET nb_enfants=7 LIMIT 4;
43	UPDATE familles SET nb_enfants=8 ORDER BY Nom LIMIT 3;
44	DELETE FROM familles WHERE Nom = 'Dupont';
45	CREATE TABLE loisirs(Nom VARCHAR(30) NOT NULL);
46	DROP TABLE loisirs;
47	INSERT INTO familles (Nom, PrenomPere) VALUES ('Alouet','Jean'),('Rahtmi','Patrick');
48	INSERT INTO familles (PrenomPere) SELECT Enfants.prenom FROM Enfants;
49	ALTER TABLE familles ADD Sports VARCHAR(50) NOT NULL;
50	ALTER TABLE familles ADD INDEX (Sports);
51	ALTER TABLE familles ADD INDEX (PrenomPere), ADD EmailPere VARCHAR(50) NOT NULL;
52	ALTER TABLE familles CHANGE PrenomPere Prenom_Pere VARCHAR(45) NOT NULL;
53	ALTER TABLE familles DROP Sports;
54	ALTER TABLE familles DROP INDEX PrenomPere;
55	ALTER TABLE familles DISABLE KEYS;
56	ALTER TABLE Enfants RENAME familles_enfants;
57	ALTER TABLE familles ORDER BY Prenom_Pere;
58	ALTER TABLE familles ADD Commentaire LONGTEXT NOT NULL ;
59	ALTER TABLE familles ADD FULLTEXT (Commentaire);
60	UPDATE familles SET Commentaire = 'La vie est un long fleuve tranquille' WHERE CodeFamille = '11';
61	UPDATE familles SET Commentaire = 'Paris, capitale de la France est traversée par un fleuve' WHERE CodeFamille ='4';
62	UPDATE familles SET Commentaire = 'Le ruisseau se jette dans la rivière qui se jette dans le FLEUVE' WHERE CodeFamille = '6';
63	SELECT * FROM familles WHERE MATCH (Commentaire) AGAINST ('fleuve');
64	SELECT Nom, Commentaire FROM familles WHERE MATCH (Commentaire) AGAINST ('-capitale+fleuve'IN BOOLEAN MODE);

Procédures, Fonctions, Déclencheurs, Vues et le Planificateur d'Evénements

Routines Stockées

Procédures stockées

Les procédures stockées et les fonctions sont créées avec les commandes **CREATE PROCEDURE** et **CREATE FUNCTION**. Une procédure est appelée avec la commande **CALL** et peut appeler une autre **routine stockée**. Une routine est une procédure stockée ou une fonction.

Une routine hérite de la base de données par défaut de l'utilisateur appelant, et donc, il est recommandé de commencer les routines avec la commande USE dbname, ou de spécifier explicitement les tables et les bases de données qui sont utilisées : i.e. base.table.

Pour illustrer les procédures stockées, vous allez travailler sur une base de données conçue pour suivre des rencontres et résultats d'un ligue d'équipes de football. Commencez par créer la base **ligue1** et les deux tables **equipe** et **rencontre**:

```
mysql> CREATE DATABASE ligue1;
Query OK, 1 row affected (0.00 sec)
```

```
  CREATE TABLE ligue1.equipe (
  id_equipe int primary key auto_increment,
  nom varchar(50) not null,
  stade varchar(50) not null,
  ville varchar(30) not null,
  points int not null default 0,
  buts int not null default 0
  ) ENGINE=MyISAM;
```

```
  CREATE TABLE ligue1.rencontre (
  id_domicile int,
  id_visiteurs int,
  date_match DATETIME,
  score_domicile tinyint,
  score_visiteurs tinyint,
  arbitre varchar(80),
  primary key (id_domicile, id_visiteurs, date_match)
  ) ENGINE=MyISAM;
```

Consultez les bases:

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CarnetAdresses     |
| ligue1             |
| mysql              |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

Sélectionnez la base **ligue1**:

```
mysql> USE ligue1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Ensuite créez la première procédure dans la base ligue1:

```
  DELIMITER //
  CREATE PROCEDURE ligue1.INIT_EQUIPE (nom_eq varchar(50), stade_eq varchar(50), ville_eq varchar(30))
  BEGIN
  INSERT INTO equipe (nom, stade, ville) VALUES (nom_eq, stade_eq, ville_eq);
  END//
  DELIMITER ;
```

Cette procédure a pour but l'insertion d'une nouvelle équipe, le stade et la ville. Vous obtiendrez un résultat similaire à celui-ci:

```
mysql>   DELIMITER //
mysql>   CREATE PROCEDURE ligue1.INIT_EQUIPE (nom_eq varchar(50), stade_eq varchar(50), ville_eq varchar(30))
    ->   BEGIN
    ->   INSERT INTO equipe (nom, stade, ville) VALUES (nom_eq, stade_eq, ville_eq);
    ->   END//
Query OK, 0 rows affected (0.00 sec)

mysql>   DELIMITER ;
mysql>
```

Appelez la procédure pour saisir trois équipes:

```
mysql> CALL ligue1.INIT_EQUIPE('FC Mandriva', 'Parc des Princes', 'Paris');
Query OK, 1 row affected (0.00 sec)

mysql> CALL ligue1.INIT_EQUIPE('Debian AC', 'Yankee Stadium', 'New York');
Query OK, 1 row affected (0.00 sec)

mysql> CALL ligue1.INIT_EQUIPE('Vista FC', 'Qwest Field', 'Redmond');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Visualisez les entrées de la table **equipe**:

```
mysql> SELECT * FROM ligue1.equipe;
+-----------+-------------+------------------+----------+--------+------+
| id_equipe | nom         | stade            | ville    | points | buts |
+-----------+-------------+------------------+----------+--------+------+
|         1 | FC Mandriva | Parc des Princes | Paris    |      0 |    0 |
|         2 | Debian AC   | Yankee Stadium   | New York |      0 |    0 |
|         3 | Vista FC    | Qwest Field      | Redmond  |      0 |    0 |
```

```
+----------+-------------+------------------+----------+--------+------+
3 rows in set (0.00 sec)

mysql>
```

Créez une deuxième procédure pour initialiser les rencontres:

```
  DELIMITER //
  CREATE PROCEDURE ligue1.SAISIR_RENCONTRE (id_dom INTEGER, id_vis INTEGER, date_m DATETIME, arbitre_m
VARCHAR(80))
  BEGIN
  INSERT INTO rencontre (id_domicile, id_visiteurs, date_match, arbitre)
  VALUES (id_dom, id_vis, date_m, arbitre_m);
  END//
  DELIMITER ;
```

Cette procédure crée un rencontre entre deux équipes à une date précise et avec un arbitre précis. Vous obtiendrez un résultat similaire à celui-ci:

```
mysql>   DELIMITER //
mysql>   CREATE PROCEDURE ligue1.SAISIR_RENCONTRE (id_dom INTEGER, id_vis INTEGER, date_m DATETIME, arbitre_m
VARCHAR(80))
    ->   BEGIN
    ->   INSERT INTO rencontre (id_domicile, id_visiteurs, date_match, arbitre)
    ->   VALUES (id_dom, id_vis, date_m, arbitre_m);
    ->   END//
Query OK, 0 rows affected (0.00 sec)

mysql>   DELIMITER ;
mysql>
```

Appelez la procédure pour créer trois rencontres:

```
mysql> CALL ligue1.SAISIR_RENCONTRE(1, 2, '2006-12-25', 'Paul Unix');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> CALL ligue1.SAISIR_RENCONTRE(2, 3, '2006-12-26', 'Joseph Bash');
Query OK, 1 row affected (0.00 sec)

mysql> CALL ligue1.SAISIR_RENCONTRE(3, 4, '2006-12-27', 'Jean Fenêtre');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Visualisez les entrées de la table **rencontre**:

```
mysql> SELECT * FROM ligue1.rencontre;
+-------------+--------------+---------------------+----------------+-----------------+---------------+
| id_domicile | id_visiteurs | date_match          | score_domicile | score_visiteurs | arbitre       |
+-------------+--------------+---------------------+----------------+-----------------+---------------+
|           1 |            2 | 2006-12-25 00:00:00 |           NULL |            NULL | Paul Unix     |
|           2 |            3 | 2006-12-26 00:00:00 |           NULL |            NULL | Joseph Bash   |
|           3 |            4 | 2006-12-27 00:00:00 |           NULL |            NULL | Jean Fenêtre |
+-------------+--------------+---------------------+----------------+-----------------+---------------+
3 rows in set (0.00 sec)

mysql>
```

Les matchs ayant eu lieu, il faut maintenant mettre à jour cette table. Créez une procédure pour la mise à jour des résultats:

```
DELIMITER //
CREATE PROCEDURE `ligue1`.`SAISIR_RESULTAT`(id_dom INTEGER, id_vis INTEGER, score_dom TINYINT(4), score_vis
TINYINT(4))
BEGIN
UPDATE ligue1.rencontre
SET score_domicile = score_dom, score_visiteurs = score_vis
WHERE id_domicile=id_dom AND id_visiteurs=id_vis;
IF score_dom = score_vis THEN
UPDATE equipe SET points = points+1
WHERE id_equipe=id_dom OR id_equipe=id_vis;
```

```
ELSEIF score_dom > score_vis THEN
UPDATE equipe SET points = points+3 WHERE id_equipe = id_dom;
ELSE UPDATE equipe SET points = points+3 WHERE id_equipe = id_vis;
END IF;
END//
DELIMITER ;
```

Vous obtiendrez un résultat similaire à celui-ci:

```
mysql> CREATE PROCEDURE `ligue1`.`SAISIR_RESULTAT`(id_dom INTEGER, id_vis INTEGER, score_dom TINYINT(4),
score_vis TINYINT(4))
    -> BEGIN
    -> UPDATE ligue1.rencontre
    -> SET score_domicile = score_dom, score_visiteurs = score_vis
    -> WHERE id_domicile=id_dom AND id_visiteurs=id_vis;
    -> IF score_dom = score_vis THEN
    -> UPDATE equipe SET points = points+1
    -> WHERE id_equipe=id_dom OR id_equipe=id_vis;
    -> ELSEIF score_dom > score_vis THEN
    -> UPDATE equipe SET points = points+3 WHERE id_equipe = id_dom;
    -> ELSE UPDATE equipe SET points = points+3 WHERE id_equipe = id_vis;
    -> END IF;
    -> END//
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

Appelez la procédure pour les trois matchs:

```
mysql> CALL ligue1.SAISIR_RESULTAT(1, 2, 2, 1);
Query OK, 1 row affected (0.01 sec)

mysql> CALL ligue1.SAISIR_RESULTAT(2, 3, 1, 3);
```

```
Query OK, 1 row affected (0.00 sec)

mysql> CALL ligue1.SAISIR_RESULTAT(3, 4, 4, 2);
Query OK, 1 row affected (0.00 sec)

mysql>
```

Visualisez les entrées de la table **rencontre**:

```
mysql> SELECT * FROM ligue1.rencontre;
+-------------+--------------+---------------------+----------------+------------------+---------------+
| id_domicile | id_visiteurs | date_match          | score_domicile | score_visiteurs | arbitre       |
+-------------+--------------+---------------------+----------------+------------------+---------------+
|           1 |            2 | 2006-12-25 00:00:00 |              2 |                1 | Paul Unix     |
|           2 |            3 | 2006-12-26 00:00:00 |              1 |                3 | Joseph Bash   |
|           3 |            4 | 2006-12-27 00:00:00 |              4 |                2 | Jean Fenêtre |
+-------------+--------------+---------------------+----------------+------------------+---------------+
3 rows in set (0.01 sec)

mysql>
```

Visualisez la table ligue1.equipe et vérifiez que les buts et les points ont été mis à jour :

```
mysql> SELECT * FROM equipe ;
+-----------+-------------+------------------+----------+--------+------+
| id_equipe | nom         | stade            | ville    | points | buts |
+-----------+-------------+------------------+----------+--------+------+
|         1 | FC Mandriva | Parc des Princes | Paris    |      3 |    0 |
|         2 | Debian AC   | Yankee Stadium   | New York |      0 |    0 |
|         3 | Vista FC    | Qwest Field      | Redmond  |      6 |    0 |
+-----------+-------------+------------------+----------+--------+------+
3 rows in set (0.00 sec)
```

```
mysql>
```

Expliquez pourquoi les buts n'ont pas été mis à jour. Proposez une solution pour que les buts soient mis à jour.

La nouvelle saison arrivée, il serait utile d'avoir une procédure qui vous crée les rencontres automatiquement. Créez donc une procédure pour initier les rencontres:

```
DELIMITER //
CREATE PROCEDURE `ligue1`.`INIT_RENCONTRES`()
BEGIN
DECLARE fini INT default 0;
DECLARE domicile, visiteur INT;
DECLARE cur_domicile CURSOR FOR SELECT id_equipe FROM equipe;
DECLARE cur_visiteur CURSOR FOR SELECT id_equipe FROM equipe;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET fini = 1;
OPEN cur_domicile;
WHILE fini <> 1 DO
FETCH cur_domicile INTO domicile;
IF fini=0 THEN
OPEN cur_visiteur;
WHILE fini <> 1 DO
FETCH cur_visiteur INTO visiteur;
IF domicile <> visiteur AND fini <> 1 THEN
INSERT INTO rencontre (id_domicile, id_visiteurs) VALUES (domicile, visiteur);
END IF;
END WHILE;
CLOSE cur_visiteur;
SET fini=0;
END IF;
END WHILE;
CLOSE cur_domicile;
```

```
END//
DELIMITER ;
```

Vous obtiendrez un résultat similaire à celui-ci:

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE `ligue1`.`INIT_RENCONTRES`()
    -> BEGIN
    -> DECLARE fini INT default 0;
    -> DECLARE domicile, visiteur INT;
    -> DECLARE cur_domicile CURSOR FOR SELECT id_equipe FROM equipe;
    -> DECLARE cur_visiteur CURSOR FOR SELECT id_equipe FROM equipe;
    -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET fini = 1;
    -> OPEN cur_domicile;
    -> WHILE fini <> 1 DO
    -> FETCH cur_domicile INTO domicile;
    -> IF fini=0 THEN
    -> OPEN cur_visiteur;
    -> WHILE fini <> 1 DO
    -> FETCH cur_visiteur INTO visiteur;
    -> IF domicile <> visiteur AND fini <> 1 THEN
    -> INSERT INTO rencontre (id_domicile, id_visiteurs) VALUES (domicile, visiteur);
    -> END IF;
    -> END WHILE;
    -> CLOSE cur_visiteur;
    -> SET fini=0;
    -> END IF;
    -> END WHILE;
    -> CLOSE cur_domicile;
    -> END//
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
```

```
mysql>
```

Un **curseur** est un objet permettant de parcourir d'une manière séquentielle le jeu de résultats retourné par une requête SQL. Dans notre cas, nous avons besoin de créer deux curseurs **cur_domicile** et **cur_visiteur** portant sur la même requête **SELECT id_equipe FROM equipe**. Le parcours du deuxième curseur est placé à l'intérieur du premier pour passer en revue toutes les combinaisons de couples (equipe1, equipe2). De cette façon, pour chaque équipe domicile on parcourt toutes les équipes visiteuses qui vont jouer contre elle (c'est à dire toutes sauf elle-même) et on insère un nouveau match à chaque fois.

Le curseur est ouvert via l'ordre **OPEN** et fermé avec **CLOSE**. La syntaxe d'itération sur un curseur repose sur une simple boucle WHILE dans laquelle on fait avancer le curseur avec **FETCH**, et sur un **HANDLER**.

Un **HANDLER** sert à déterminer ce qui se passe lorsque le **SQLSTATE NOT FOUND** est atteint, autrement dit quand le curseur est au bout du jeu de résultats. En général on sort de la boucle WHILE.

Supprimez maintenant les enregistrements de la saison précédente se trouvant dans la table ligue1.rencontres :

```
mysql> DELETE FROM ligue1.rencontre WHERE id_domicile IS NOT NULL;
Query OK, 3 rows affected (0.00 sec)

mysql>
```

Appelez la procédure:

```
mysql> CALL ligue1.INIT_RENCONTRES();
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
```

Visualisez les entrées de la table **rencontre**:

```
mysql> SELECT * FROM ligue1.rencontre;
```

```
+-------------+--------------+---------------------+----------------+-----------------+---------+
| id_domicile | id_visiteurs | date_match          | score_domicile | score_visiteurs | arbitre |
+-------------+--------------+---------------------+----------------+-----------------+---------+
|           2 |            3 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
|           2 |            1 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
|           1 |            3 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
|           1 |            2 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
|           3 |            1 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
|           3 |            2 | 0000-00-00 00:00:00 |           NULL |            NULL | NULL    |
+-------------+--------------+---------------------+----------------+-----------------+---------+
6 rows in set (0.00 sec)

mysql>
```

Réinitialisez la table ligue1.equipe :

```
mysql> UPDATE ligue1.equipe SET points=0, buts=0;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 3  Changed: 2  Warnings: 0

mysql>
```

Visualisez les entrées de la table **equipe** :

```
mysql> SELECT * FROM ligue1.equipe;
+-----------+-------------+------------------+----------+--------+------+
| id_equipe | nom         | stade            | ville    | points | buts |
+-----------+-------------+------------------+----------+--------+------+
|         1 | FC Mandriva | Parc des Princes | Paris    |      0 |    0 |
|         2 | Debian AC   | Yankee Stadium   | New York |      0 |    0 |
|         3 | Vista FC    | Qwest Field      | Redmond  |      0 |    0 |
+-----------+-------------+------------------+----------+--------+------+
3 rows in set (0.00 sec)
```

```
mysql>
```

Le code d'une procédure est stocké dans la table interne **INFORMATION_SCHEMA.ROUTINES**.

Pour visualiser les procédures créées, il convient d'utiliser la commande suivante :

```
mysql> SELECT * FROM information_schema.routines WHERE ROUTINE_TYPE='PROCEDURE'\G
*************************** 1. row ***************************
           SPECIFIC_NAME: INIT_EQUIPE
         ROUTINE_CATALOG: def
          ROUTINE_SCHEMA: ligue1
            ROUTINE_NAME: INIT_EQUIPE
            ROUTINE_TYPE: PROCEDURE
               DATA_TYPE:
CHARACTER_MAXIMUM_LENGTH: NULL
  CHARACTER_OCTET_LENGTH: NULL
       NUMERIC_PRECISION: NULL
           NUMERIC_SCALE: NULL
      CHARACTER_SET_NAME: NULL
          COLLATION_NAME: NULL
          DTD_IDENTIFIER: NULL
            ROUTINE_BODY: SQL
      ROUTINE_DEFINITION: BEGIN
  INSERT INTO equipe (nom, stade, ville) VALUES (nom_eq, stade_eq, ville_eq);
  END
           EXTERNAL_NAME: NULL
       EXTERNAL_LANGUAGE: NULL
         PARAMETER_STYLE: SQL
        IS_DETERMINISTIC: NO
         SQL_DATA_ACCESS: CONTAINS SQL
                SQL_PATH: NULL
```

```
           SECURITY_TYPE: DEFINER
                 CREATED: 2014-01-27 15:41:31
            LAST_ALTERED: 2014-01-27 15:41:31
                SQL_MODE:
         ROUTINE_COMMENT:
                 DEFINER: root@localhost
    CHARACTER_SET_CLIENT: utf8
    COLLATION_CONNECTION: utf8_general_ci
      DATABASE_COLLATION: latin1_swedish_ci
*************************** 2. row ***************************
           SPECIFIC_NAME: INIT_RENCONTRES
         ROUTINE_CATALOG: def
          ROUTINE_SCHEMA: ligue1
            ROUTINE_NAME: INIT_RENCONTRES
            ROUTINE_TYPE: PROCEDURE
               DATA_TYPE:
CHARACTER_MAXIMUM_LENGTH: NULL
  CHARACTER_OCTET_LENGTH: NULL
       NUMERIC_PRECISION: NULL
           NUMERIC_SCALE: NULL
      CHARACTER_SET_NAME: NULL
          COLLATION_NAME: NULL
          DTD_IDENTIFIER: NULL
            ROUTINE_BODY: SQL
      ROUTINE_DEFINITION: BEGIN
DECLARE fini INT default 0;
DECLARE domicile, visiteur INT;
DECLARE cur_domicile CURSOR FOR SELECT id_equipe FROM equipe;
DECLARE cur_visiteur CURSOR FOR SELECT id_equipe FROM equipe;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET fini = 1;
OPEN cur_domicile;
WHILE fini <> 1 DO
FETCH cur_domicile INTO domicile;
IF fini=0 THEN
```

```
OPEN cur_visiteur;
WHILE fini <> 1 DO
FETCH cur_visiteur INTO visiteur;
IF domicile <> visiteur AND fini <> 1 THEN
INSERT INTO rencontre (id_domicile, id_visiteurs) VALUES (domicile, visiteur);
END IF;
END WHILE;
CLOSE cur_visiteur;
SET fini=0;
END IF;
END WHILE;
CLOSE cur_domicile;
END
           EXTERNAL_NAME: NULL
       EXTERNAL_LANGUAGE: NULL
         PARAMETER_STYLE: SQL
        IS_DETERMINISTIC: NO
         SQL_DATA_ACCESS: CONTAINS SQL
                SQL_PATH: NULL
           SECURITY_TYPE: DEFINER
                 CREATED: 2014-01-27 15:42:19
            LAST_ALTERED: 2014-01-27 15:42:19
                SQL_MODE:
         ROUTINE_COMMENT:
                 DEFINER: root@localhost
    CHARACTER_SET_CLIENT: utf8
    COLLATION_CONNECTION: utf8_general_ci
      DATABASE_COLLATION: latin1_swedish_ci
*************************** 3. row ***************************
           SPECIFIC_NAME: SAISIR_RENCONTRE
         ROUTINE_CATALOG: def
          ROUTINE_SCHEMA: ligue1
            ROUTINE_NAME: SAISIR_RENCONTRE
            ROUTINE_TYPE: PROCEDURE
```

```
               DATA_TYPE:
CHARACTER_MAXIMUM_LENGTH: NULL
  CHARACTER_OCTET_LENGTH: NULL
       NUMERIC_PRECISION: NULL
           NUMERIC_SCALE: NULL
      CHARACTER_SET_NAME: NULL
          COLLATION_NAME: NULL
          DTD_IDENTIFIER: NULL
            ROUTINE_BODY: SQL
      ROUTINE_DEFINITION: BEGIN
  INSERT INTO rencontre (id_domicile, id_visiteurs, date_match, arbitre)
  VALUES (id_dom, id_vis, date_m, arbitre_m);
  END
           EXTERNAL_NAME: NULL
       EXTERNAL_LANGUAGE: NULL
         PARAMETER_STYLE: SQL
        IS_DETERMINISTIC: NO
         SQL_DATA_ACCESS: CONTAINS SQL
                SQL_PATH: NULL
           SECURITY_TYPE: DEFINER
                 CREATED: 2014-01-27 15:41:44
            LAST_ALTERED: 2014-01-27 15:41:44
                SQL_MODE:
         ROUTINE_COMMENT:
                 DEFINER: root@localhost
    CHARACTER_SET_CLIENT: utf8
    COLLATION_CONNECTION: utf8_general_ci
      DATABASE_COLLATION: latin1_swedish_ci
*************************** 4. row ***************************
           SPECIFIC_NAME: SAISIR_RESULTAT
         ROUTINE_CATALOG: def
          ROUTINE_SCHEMA: ligue1
            ROUTINE_NAME: SAISIR_RESULTAT
            ROUTINE_TYPE: PROCEDURE
```

```
               DATA_TYPE:
CHARACTER_MAXIMUM_LENGTH: NULL
  CHARACTER_OCTET_LENGTH: NULL
       NUMERIC_PRECISION: NULL
           NUMERIC_SCALE: NULL
      CHARACTER_SET_NAME: NULL
          COLLATION_NAME: NULL
          DTD_IDENTIFIER: NULL
            ROUTINE_BODY: SQL
      ROUTINE_DEFINITION: BEGIN
UPDATE ligue1.rencontre
SET score_domicile = score_dom, score_visiteurs = score_vis
WHERE id_domicile=id_dom AND id_visiteurs=id_vis;
IF score_dom = score_vis THEN
UPDATE equipe SET points = points+1
WHERE id_equipe=id_dom OR id_equipe=id_vis;
ELSEIF score_dom > score_vis THEN
UPDATE equipe SET points = points+3 WHERE id_equipe = id_dom;
ELSE UPDATE equipe SET points = points+3 WHERE id_equipe = id_vis;
END IF;
END
          EXTERNAL_NAME: NULL
       EXTERNAL_LANGUAGE: NULL
         PARAMETER_STYLE: SQL
        IS_DETERMINISTIC: NO
         SQL_DATA_ACCESS: CONTAINS SQL
                SQL_PATH: NULL
           SECURITY_TYPE: DEFINER
                 CREATED: 2014-01-27 15:41:57
            LAST_ALTERED: 2014-01-27 15:41:57
                SQL_MODE:
         ROUTINE_COMMENT:
                 DEFINER: root@localhost
    CHARACTER_SET_CLIENT: utf8
```

```
    COLLATION_CONNECTION: utf8_general_ci
      DATABASE_COLLATION: latin1_swedish_ci
4 rows in set (0.01 sec)
```

Fonctions Stockées

Vous allez maintenant créer une **Fonction Stockée** que vous utiliserez ultérieurement. La différence entre une procédure stockée et une fonction stockée est que cette dernière n'a pas besoin d'être appelée en utilisant la commande CALL. La fonction est appelée à partir d'une commande SQL. Cette fonction va vous identifier les équipes relégables en fin de saison.

Créez donc la fonction intégrée :

```
DELIMITER //
CREATE FUNCTION EST_RELEGABLE(idequipe INTEGER) RETURNS TINYINT
BEGIN
DECLARE relegable TINYINT default 0;
DECLARE id_courante INTEGER;
DECLARE fini TINYINT DEFAULT 0;
DECLARE cur1 CURSOR FOR SELECT id_equipe FROM equipe ORDER BY points ASC LIMIT 3;
DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET fini = 1;
OPEN cur1;
BOUCLE: REPEAT
FETCH cur1 INTO id_courante;
IF idequipe=id_courante THEN
SET relegable=1;
LEAVE BOUCLE;
END IF;
UNTIL fini END REPEAT BOUCLE;
CLOSE cur1;
RETURN relegable;
END//
DELIMITER ;
```

Vous obtiendrez un résultat similaire à celui-ci:

```
mysql> DELIMITER //
mysql> CREATE FUNCTION EST_RELEGABLE(idequipe INTEGER) RETURNS TINYINT
    -> BEGIN
    -> DECLARE relegable TINYINT default 0;
    -> DECLARE id_courante INTEGER;
    -> DECLARE fini TINYINT DEFAULT 0;
    -> DECLARE cur1 CURSOR FOR SELECT id_equipe FROM equipe ORDER BY points ASC LIMIT 3;
    -> DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET fini = 1;
    -> OPEN cur1;
    -> BOUCLE: REPEAT
    -> FETCH cur1 INTO id_courante;
    -> IF idequipe=id_courante THEN
    -> SET relegable=1;
    -> LEAVE BOUCLE;
    -> END IF;
    -> UNTIL fini END REPEAT BOUCLE;
    -> CLOSE cur1;
    -> RETURN relegable;
    -> END//
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

A la fin de la saison, il conviendrait de saisir la commande suivante pour identifier les équipes relégables :

```
mysql> SELECT * FROM equipe WHERE EST_RELEGABLE(id_equipe)=1; [Entrée]
```

Ensuite il conviendrait de les supprimer :

```
mysql> DELETE FROM equipe WHERE EST_RELEGABLE(id_equipe)=1; [Entrée]
```

Le code d'une fonction est stocké dans la table interne **INFORMATION_SCHEMA.ROUTINES**.

Pour visualiser les fonctions stockées, utilisez la commande suivante :

```
mysql> SELECT * FROM information_schema.routines WHERE ROUTINE_TYPE='FUNCTION'\G
*************************** 1. row ***************************
           SPECIFIC_NAME: EST_RELEGABLE
         ROUTINE_CATALOG: def
          ROUTINE_SCHEMA: ligue1
            ROUTINE_NAME: EST_RELEGABLE
            ROUTINE_TYPE: FUNCTION
               DATA_TYPE: tinyint
CHARACTER_MAXIMUM_LENGTH: NULL
  CHARACTER_OCTET_LENGTH: NULL
       NUMERIC_PRECISION: 3
           NUMERIC_SCALE: 0
      CHARACTER_SET_NAME: NULL
          COLLATION_NAME: NULL
          DTD_IDENTIFIER: tinyint(4)
            ROUTINE_BODY: SQL
      ROUTINE_DEFINITION: BEGIN
DECLARE relegable TINYINT default 0;
DECLARE id_courante INTEGER;
DECLARE fini TINYINT DEFAULT 0;
DECLARE cur1 CURSOR FOR SELECT id_equipe FROM equipe ORDER BY points ASC LIMIT 3;
DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET fini = 1;
OPEN cur1;
BOUCLE: REPEAT
FETCH cur1 INTO id_courante;
IF idequipe=id_courante THEN
SET relegable=1;
```

```
LEAVE BOUCLE;
END IF;
UNTIL fini END REPEAT BOUCLE;
CLOSE cur1;
RETURN relegable;
END
           EXTERNAL_NAME: NULL
       EXTERNAL_LANGUAGE: NULL
         PARAMETER_STYLE: SQL
        IS_DETERMINISTIC: NO
         SQL_DATA_ACCESS: CONTAINS SQL
                SQL_PATH: NULL
           SECURITY_TYPE: DEFINER
                 CREATED: 2014-01-27 16:01:23
            LAST_ALTERED: 2014-01-27 16:01:23
                SQL_MODE:
         ROUTINE_COMMENT:
                 DEFINER: root@localhost
    CHARACTER_SET_CLIENT: utf8
    COLLATION_CONNECTION: utf8_general_ci
      DATABASE_COLLATION: latin1_swedish_ci
1 row in set (0.01 sec)
```

Vous ne pouvez pas utiliser les commandes suivantes dans des routines stockées : LOCK TABLES, UNLOCK TABLES, ALTER VIEW, LOAD DATA/TABLE, PREPARE, EXECUTE, DEALLOCATE PREPARE.

Déclencheurs

Le support des déclencheurs (*triggers*) est inclus dans les versions de MySQL à partir de la version 5.0.2. Un déclencheur est un objet de base de données nommé, qui est associé à une table et qui s'active lorsqu'une **ACTION** de type **INSERT**, **DELETE** ou **UPDATE** sont effectuées sur une table.

Un déclencheur peut être appelé avec une **POSITION**. La valeur de la POSITION est **BEFORE** ou **AFTER** l'**ACTION**.

Créez un trigger pour convertir la ville en majuscules lors de son insertion:

```
DELIMITER //
CREATE TRIGGER TGR_BI_EQUIPE BEFORE INSERT ON equipe FOR EACH ROW
BEGIN
SET new.ville = UPPER(new.ville);
END//
DELIMITER ;
```

Insérez une nouvel enregistrement:

```
mysql> INSERT INTO equipe (nom, stade, ville) VALUES ('Racing Club Strasbourg', 'La Meinau', 'strasbourg');
Query OK, 1 row affected (0.00 sec)

mysql>
```

```
mysql> SELECT nom, stade, ville FROM equipe;
+------------------------+------------------+------------+
| nom                    | stade            | ville      |
+------------------------+------------------+------------+
| FC Mandriva            | Parc des Princes | Paris      |
| Debian AC              | Yankee Stadium   | New York   |
| Vista FC               | Qwest Field      | Redmond    |
| Racing Club Strasbourg | La Meinau        | STRASBOURG |
+------------------------+------------------+------------+
4 rows in set (0.00 sec)

mysql>
```

Créez un trigger pour faire le total des buts:

```
DELIMITER //
CREATE TRIGGER TGR_BU_RENCONTRE BEFORE UPDATE ON rencontre FOR EACH ROW
BEGIN
IF old.score_domicile IS NULL and new.score_domicile IS NOT NULL THEN
UPDATE equipe SET buts = buts + new.score_domicile WHERE id_equipe=new.id_domicile;
END IF;
IF  old.score_visiteurs IS NULL and new.score_visiteurs IS NOT NULL THEN
UPDATE equipe SET buts = buts + new.score_visiteurs WHERE id_equipe=new.id_visiteurs;
END IF;
END//
DELIMITER ;
```

Appelez la procédure pour inscrire un résultat:

```
mysql> CALL SAISIR_RESULTAT(1, 2, 2, 1);
Query OK, 1 row affected (0.01 sec)

mysql>
```

Appelez la procédure une deuxième fois pour inscrire un autre résultat:

```
mysql> CALL SAISIR_RESULTAT(2, 1, 2, 1);
Query OK, 1 row affected (0.00 sec)

mysql>
```

Examinez les résultats obtenus :

```
mysql> SELECT nom, stade, points, buts FROM equipe;
+-----------------------+------------------+--------+------+
| nom                   | stade            | points | buts |
+-----------------------+------------------+--------+------+
```

```
| FC Mandriva            | Parc des Princes |      3 |    3 |
| Debian AC              | Yankee Stadium   |      3 |    3 |
| Vista FC               | Qwest Field      |      0 |    0 |
| Racing Club Strasbourg | La Meinau        |      0 |    0 |
+------------------------+------------------+--------+------+
4 rows in set (0.00 sec)

mysql>
```

Les limites des triggers:

- Limitation à un seul trigger par table, par ACTION et par POSITION.
- Uniquement triggers FOR EACH ROW (déclencheurs niveau ligne).
- Pas de déclencheurs sur des événements système autres que des requêtes SQL,
- Les déclencheurs ne peuvent pas être mis sur les tables de la base mysql,
- Les déclencheurs ne sont pas déclenchés par les ACTIONS de clés étrangères.

Vues

Les vues, y compris les vues modifiables, sont disponibles dans les versions binaires depuis la version 5.0.1.

- Création ou modification de vues avec les commandes CREATE VIEW ou ALTER VIEW
- Destruction de vues avec DROP VIEW
- Affichage des méta-données de vues avec SHOW CREATE VIEW

Ajoutez d'abord des champs à notre table **equipe** pour stocker le nom des entraîneurs ainsi que leurs numéros de téléphone :

```
ALTER TABLE equipe
ADD entraineur varchar(100) default 'inconnu',
ADD tel_entraineur varchar(20) default 'inconnu';
```

Mettez à jour la table:

```
mysql> UPDATE equipe SET entraineur='Ricardo GOMES', tel_entraineur='06-56-56-56-56' WHERE id_equipe=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE equipe SET entraineur='Gérard Houllier', tel_entraineur='06-57-57-57-57' WHERE id_equipe=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

Créez une vue qui **ne montre pas** le numéro de téléphone:

```
mysql> CREATE VIEW V_EQUIPE AS SELECT id_equipe, nom, stade, ville, points, buts, entraineur FROM equipe;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Le code d'un view est stocké dans la table interne **INFORMATION_SCHEMA.VIEWS**.

Donnez les droits sur cette vue à user1:

```
mysql> GRANT SELECT, INSERT, DELETE, UPDATE ON V_EQUIPE TO user1@localhost IDENTIFIED BY 'user1';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Déconnectez-vous et reconnectez-vous en tant qu'user1:

```
mysql> exit
```

```
Bye
[root@centos6 ~]# mysql -u user1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Changez de base de données:

```
mysql> use ligue1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Ensuite, visualisez la vue V_EQUIPE:

```
mysql> SELECT * FROM V_EQUIPE;
+-----------+------------------------+------------------+------------+--------+------+------------------+
| id_equipe | nom                    | stade            | ville      | points | buts | entraineur       |
+-----------+------------------------+------------------+------------+--------+------+------------------+
|         1 | FC Mandriva            | Parc des Princes | Paris      |      3 |    3 | Ricardo GOMES    |
|         2 | Debian AC              | Yankee Stadium   | New York   |      3 |    3 | Gérard Houllier  |
```

```
|         3 | Vista FC               | Qwest Field      | Redmond    |      0 |    0 | inconnu          |
|         4 | Racing Club Strasbourg | La Meinau        | STRASBOURG |      0 |    0 | inconnu          |
+-----------+------------------------+------------------+------------+--------+------+------------------+
4 rows in set (0.00 sec)

mysql>
```

Déconnectez-vous et reconnectez-vous en tant que root.

Planificateur d'Evénements

```
mysql> SHOW VARIABLES LIKE 'event_scheduler';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| event_scheduler | OFF   |
+-----------------+-------+
1 row in set (0.01 sec)

mysql> SET GLOBAL event_scheduler = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'event_scheduler';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| event_scheduler | ON    |
+-----------------+-------+
1 row in set (0.01 sec)
```

```
mysql>
```

```
mysql> SHOW PROCESSLIST;
+----+-----------------+-----------+------+---------+------+-----------------------+------------------+
| Id | User            | Host      | db   | Command | Time | State                 | Info             |
+----+-----------------+-----------+------+---------+------+-----------------------+------------------+
| 12 | root            | localhost | NULL | Query   |    0 | NULL                  | SHOW PROCESSLIST |
| 13 | event_scheduler | localhost | NULL | Daemon  |  193 | Waiting on empty queue | NULL             |
+----+-----------------+-----------+------+---------+------+-----------------------+------------------+
2 rows in set (0.01 sec)

mysql>
```

Administration et Maintenance

Administration

Mysqladmin

Mysqladmin est utile pour diverses tâches d'administration, comme la création et la suppression de base de données ou le changement démographe. La syntaxe est simple :

```
 # mysqladmin [option] admincommand
```

où

- **admincommand** définit la tâche effectuée par mysqladmin.

Nous présentons ici que les trois commandes principales.

create crée une nouvelle base de données :

```
[root@centos6 ~]# mysqladmin -u root -p create newdb
Enter password: fenestros
```

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqladmin -u root -p create newdb
Enter password: *********

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

drop supprime une base de données:

```
[root@centos6 ~]# mysqladmin -u root -p drop newdb
Enter password: fenestros
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'newdb' database [y/N] y
Database "newdb" dropped
```

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqladmin -u root -p drop newdb
Enter password: *********
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'newdb' database [y/N] y
Database "newdb" dropped

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

password modifier le mot de passe de l'utilisateur donné par –u :

```
[root@centos6 ~]# mysqladmin -u root -p password "fenestros1"
Enter password: fenestros
```

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqladmin -u root -p password "fenestros1"
Enter password: *********
Warning: Using a password on the command line interface can be insecure.

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

Connectez-vous à MySQL en tant que root en utilisant le nouveau mot de passe :

```
[root@centos6 ~]# mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysql -u root -p mysql
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.6.28-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Notez que les mots de passe ne sont pas visibles. Ils le sont dans les exemples précédents uniquement pour rendre le cours plus clair.

Méta-données

Avec MySQL on parle souvent de **Méta-données**. Ces données donnent non seulement des renseignements sur le contenu d'une table, mais aussi des précisions sur ses propriétés. Par exemple:

- Les bases de données gérées par MySQL
- Les tables contenus dans ces bases de données
- Les propriétés dont disposent les tables

Plusieurs commandes peuvent être utilisées pour visualiser les **Méta-données**:

La commande DESCRIBE

Sélectionnez la base de données **ligue1**:

```
mysql> USE ligue1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql>
```

Décrivez la table **equipe** de la base **ligue1**:

```
mysql> DESCRIBE equipe;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id_equipe      | int(11)      | NO   | PRI | NULL    | auto_increment |
| nom            | varchar(50)  | NO   |     | NULL    |                |
| stade          | varchar(50)  | NO   |     | NULL    |                |
| ville          | varchar(30)  | NO   |     | NULL    |                |
| points         | int(11)      | NO   |     | 0       |                |
| buts           | int(11)      | NO   |     | 0       |                |
| entraineur     | varchar(100) | YES  |     | inconnu |                |
| tel_entraineur | varchar(20)  | YES  |     | inconnu |                |
+----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql>
```

La commande EXPLAIN

La commande EXPLAIN produit le même résultat que la commande DESCRIBE :

```
mysql> EXPLAIN equipe;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id_equipe      | int(11)      | NO   | PRI | NULL    | auto_increment |
| nom            | varchar(50)  | NO   |     |         |                |
```

```
| stade          | varchar(50)  | NO   |     |         |                |
| ville          | varchar(30)  | NO   |     |         |                |
| points         | int(11)      | NO   |     | 0       |                |
| buts           | int(11)      | NO   |     | 0       |                |
| entraineur     | varchar(100) | YES  |     | inconnu |                |
| tel_entraineur | varchar(20)  | YES  |     | inconnu |                |
+----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql>
```

Par contre la commande EXPLAIN peut aussi être utilisée pour contrôler le processus suivi par MySQL lors d'une commande donnée. Pour comprendre ceci, saisissez les deux commandes suivantes:

```
mysql> SELECT nom, stade, ville FROM equipe;
+------------------------+------------------+------------+
| nom                    | stade            | ville      |
+------------------------+------------------+------------+
| FC Mandriva            | Parc des Princes | Paris      |
| Debian AC              | Yankee Stadium   | New York   |
| Vista FC               | Qwest Field      | Redmond    |
| Racing Club Strasbourg | La Meinau        | STRASBOURG |
+------------------------+------------------+------------+
4 rows in set (0.00 sec)

mysql> EXPLAIN SELECT nom, stade, ville FROM equipe;
+----+-------------+--------+------+---------------+------+---------+------+------+-------+
| id | select_type | table  | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+--------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | equipe | ALL  | NULL          | NULL | NULL    | NULL |    4 |       |
+----+-------------+--------+------+---------------+------+---------+------+------+-------+
1 row in set (0.00 sec)
```

```
mysql>
```

La commande BENCHMARK

Cette commande retourne toujours la valeur 0. Par contre son utilité est de contrôler le temps d'exécution du client MySQL. La commande prend la forme suivante:

```
mysql> BENCHMARK(nbfois, expression); [Entrée]
```

Par exemple, la commande suivante, exécutée trois fois de suite:

```
mysql> select BENCHMARK(1000000000, "select * FROM equipe");
+-----------------------------------------------+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----------------------------------------------+
|                                             0 |
+-----------------------------------------------+
1 row in set (8.18 sec)

mysql> select BENCHMARK(1000000000, "select * FROM equipe");
+-----------------------------------------------+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----------------------------------------------+
|                                             0 |
+-----------------------------------------------+
1 row in set (8.14 sec)

mysql> select BENCHMARK(1000000000, "select * FROM equipe");
+-----------------------------------------------+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----------------------------------------------+
|                                             0 |
```

```
+-----------------------------------------------+
1 row in set (8.14 sec)

mysql>
```

La Commande SHOW

La commande SHOW permet d'obtenir de différentes informations sur les bases MySQL:

SHOW DATABASES

Cette commande prend la forme suivante :

```
SHOW DATABASES [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CarnetAdresses     |
| ligue1             |
| mysql              |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

SHOW TABLES

Cette commande prend la forme suivante :

```
SHOW TABLES [FROM nomdb] [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW tables ;
+------------------------+
| Tables_in_ligue1       |
+------------------------+
| V_EQUIPE               |
| equipe                 |
| rencontre              |
+------------------------+
3 rows in set (0.00 sec)

mysql>
```

SHOW OPEN TABLES

Cette commande prend la forme suivante :

```
SHOW OPEN TABLES [FROM nomdb] [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW OPEN TABLES;
+----------------+----------------------------+--------+-------------+
| Database       | Table                      | In_use | Name_locked |
+----------------+----------------------------+--------+-------------+
| ligue1         | equipe                     |      0 |           0 |
| mysql          | event                      |      0 |           0 |
| mysql          | time_zone                  |      0 |           0 |
| mysql          | procs_priv                 |      0 |           0 |
```

```
| mysql          | time_zone_transition      |      0 |            0 |
| mysql          | db                        |      0 |            0 |
| mysql          | user                      |      0 |            0 |
| mysql          | tables_priv               |      0 |            0 |
| CarnetAdresses | familles                  |      0 |            0 |
| mysql          | help_keyword              |      0 |            0 |
| mysql          | help_category             |      0 |            0 |
| mysql          | help_relation             |      0 |            0 |
| mysql          | func                      |      0 |            0 |
| mysql          | slow_log                  |      0 |            0 |
| mysql          | columns_priv              |      0 |            0 |
| ligue1         | rencontre                 |      0 |            0 |
| mysql          | host                      |      0 |            0 |
| mysql          | proc                      |      0 |            0 |
| mysql          | servers                   |      0 |            0 |
| mysql          | help_topic                |      0 |            0 |
| mysql          | time_zone_leap_second     |      0 |            0 |
| mysql          | ndb_binlog_index          |      0 |            0 |
| mysql          | time_zone_transition_type |      0 |            0 |
| mysql          | time_zone_name            |      0 |            0 |
| mysql          | plugin                    |      0 |            0 |
| CarnetAdresses | Enfants                   |      0 |            0 |
| mysql          | general_log               |      0 |            0 |
+----------------+---------------------------+--------+--------------+
27 rows in set (0.00 sec)

mysql>
```

La valeur de la colonne **In_use** étant 0, ceci indique que la table est ouverte mais qu'il n'y a pas d'accès actuel. Si un client cause un verrou d'être posé sur une table, la valeur passe à **1**. Quand un autre client exécute une requête nécessitant un verrou sur la table, la valeur passe à 2. Mais dans ce cas, la deuxième requête n'est pas exécutée avant la libération du verrou par le premier client. La valeur de la colonne **Name_locked** indique le nombre de verrous posés sur le nom de la table. Un verrou est posé sur le nom d'une table lors d'une opération de renommage de la table ou de la suppression de la table. Pour plus 'information, voir

http://dev.mysql.com/doc/refman/5.5/en/show-open-tables.html.

SHOW COLUMNS

Cette commande prend la forme suivante :

```
SHOW [FULL] COLUMNS FROM nomtable [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW COLUMNS FROM equipe;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id_equipe      | int(11)      | NO   | PRI | NULL    | auto_increment |
| nom            | varchar(50)  | NO   |     | NULL    |                |
| stade          | varchar(50)  | NO   |     | NULL    |                |
| ville          | varchar(30)  | NO   |     | NULL    |                |
| points         | int(11)      | NO   |     | 0       |                |
| buts           | int(11)      | NO   |     | 0       |                |
| entraineur     | varchar(100) | YES  |     | inconnu |                |
| tel_entraineur | varchar(20)  | YES  |     | inconnu |                |
+----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql>
```

ou

```
mysql> SHOW FULL COLUMNS FROM equipe;
+----------------+--------------+-------------------+------+-----+---------+----------------+-------------------
-------------+---------+
| Field          | Type         | Collation         | Null | Key | Default | Extra          | Privileges
```

```
| Comment |
+---------------+-------------+-------------------+------+-----+---------+----------------+-------------------
------------+---------+
| id_equipe      | int(11)      | NULL              | NO   | PRI | NULL    | auto_increment |
select,insert,update,references |         |
| nom            | varchar(50)  | latin1_swedish_ci | NO   |     | NULL    |                |
select,insert,update,references |         |
| stade          | varchar(50)  | latin1_swedish_ci | NO   |     | NULL    |                |
select,insert,update,references |         |
| ville          | varchar(30)  | latin1_swedish_ci | NO   |     | NULL    |                |
select,insert,update,references |         |
| points         | int(11)      | NULL              | NO   |     | 0       |                |
select,insert,update,references |         |
| buts           | int(11)      | NULL              | NO   |     | 0       |                |
select,insert,update,references |         |
| entraineur     | varchar(100) | latin1_swedish_ci | YES  |     | inconnu |                |
select,insert,update,references |         |
| tel_entraineur | varchar(20)  | latin1_swedish_ci | YES  |     | inconnu |                |
select,insert,update,references |         |
+---------------+-------------+-------------------+------+-----+---------+----------------+-------------------
------------+---------+
8 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-columns.html.

SHOW INDEX

Cette commande prend la forme suivante :

```
SHOW INDEX FROM nomtable [FROM nomdb];
```

Par exemple;

```
mysql> SHOW INDEX FROM equipe;
+--------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+----
--+------------+---------+
| Table  | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed |
Null | Index_type | Comment |
+--------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+----
--+------------+---------+
| equipe |          0 | PRIMARY  |            1 | id_equipe   | A         |           4 |     NULL | NULL   |
| BTREE      |         |
+--------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+----
--+------------+---------+
1 row in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-index.html.

SHOW TABLE STATUS

Cette commande prend la forme suivante :

```
SHOW TABLE STATUS [FROM nomdb] [LIKE '%motif%'];
```

Par exemple;

```
mysql> SHOW TABLE STATUS;
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+-----------
```

```
--+-----------+----------------+---------------------+---------------------+------------+-------------------+----
------+----------------+---------+
| Name      | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length |
Index_length | Data_free | Auto_increment | Create_time         | Update_time         | Check_time | Collation
| Checksum | Create_options | Comment |
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+-----------
--+-----------+----------------+---------------------+---------------------+------------+-------------------+----
------+----------------+---------+
| V_EQUIPE  | NULL   |    NULL | NULL       | NULL |           NULL |        NULL |            NULL |
NULL |      NULL |           NULL | NULL                | NULL                | NULL       | NULL              |
NULL | NULL           | VIEW    |
| equipe    | MyISAM |      10 | Dynamic    |    4 |             80 |         320 | 281474976710655 |
2048 |         0 |              5 | 2012-10-23 09:57:32 | 2012-10-23 09:58:11 | NULL       | latin1_swedish_ci |
NULL |                |         |
| rencontre | MyISAM |      10 | Dynamic    |    6 |             30 |         184 | 281474976710655 |
2048 |         0 |           NULL | 2012-10-22 14:38:00 | 2012-10-23 09:44:53 | NULL       | latin1_swedish_ci |
NULL |                |         |
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+-----------
--+-----------+----------------+---------------------+---------------------+------------+-------------------+----
------+----------------+---------+
3 rows in set (0.01 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-table-status.html.

SHOW STATUS

Cette commande prend la forme suivante :

```
SHOW STATUS [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW STATUS;
+-----------------------------------+-----------+
| Variable_name                     | Value     |
+-----------------------------------+-----------+
| Aborted_clients                   | 0         |
| Aborted_connects                  | 1         |
| Binlog_cache_disk_use             | 0         |
| Binlog_cache_use                  | 0         |
| Bytes_received                    | 1112      |
| Bytes_sent                        | 24194     |
| Com_admin_commands                | 0         |
| Com_assign_to_keycache            | 0         |
| Com_alter_db                      | 0         |
| Com_alter_db_upgrade              | 0         |
| Com_alter_event                   | 0         |
| Com_alter_function                | 0         |
| Com_alter_procedure               | 0         |
| Com_alter_server                  | 0         |
| Com_alter_table                   | 0         |
| Com_alter_tablespace              | 0         |
| Com_analyze                       | 0         |
| Com_backup_table                  | 0         |
| Com_begin                         | 0         |
| Com_binlog                        | 0         |
| Com_call_procedure                | 0         |
| Com_change_db                     | 1         |
| Com_change_master                 | 0         |
| Com_check                         | 0         |
| Com_checksum                      | 0         |
| Com_commit                        | 0         |
| Com_create_db                     | 0         |
| Com_create_event                  | 0         |
```

```
| Com_create_function                | 0          |
| Com_create_index                   | 0          |
| Com_create_procedure               | 0          |
| Com_create_server                  | 0          |
| Com_create_table                   | 0          |
| Com_create_trigger                 | 0          |
| Com_create_udf                     | 0          |
| Com_create_user                    | 0          |
| Com_create_view                    | 0          |
| Com_dealloc_sql                    | 0          |
| Com_delete                         | 0          |
| Com_delete_multi                   | 0          |
| Com_do                             | 0          |
| Com_drop_db                        | 0          |
| Com_drop_event                     | 0          |
| Com_drop_function                  | 0          |
| Com_drop_index                     | 0          |
| Com_drop_procedure                 | 0          |
| Com_drop_server                    | 0          |
| Com_drop_table                     | 0          |
| Com_drop_trigger                   | 0          |
| Com_drop_user                      | 0          |
| Com_drop_view                      | 0          |
| Com_empty_query                    | 0          |
| Com_execute_sql                    | 0          |
| Com_flush                          | 0          |
| Com_grant                          | 0          |
| Com_ha_close                       | 0          |
| Com_ha_open                        | 0          |
| Com_ha_read                        | 0          |
| Com_help                           | 0          |
| Com_insert                         | 0          |
| Com_insert_select                  | 0          |
| Com_install_plugin                 | 0          |
```

```
| Com_kill                          | 0         |
| Com_load                          | 0         |
| Com_load_master_data              | 0         |
| Com_load_master_table             | 0         |
| Com_lock_tables                   | 0         |
| Com_optimize                      | 0         |
| Com_preload_keys                  | 0         |
| Com_prepare_sql                   | 0         |
| Com_purge                         | 0         |
| Com_purge_before_date             | 0         |
| Com_release_savepoint             | 0         |
| Com_rename_table                  | 0         |
| Com_rename_user                   | 0         |
| Com_repair                        | 0         |
| Com_replace                       | 0         |
| Com_replace_select                | 0         |
| Com_reset                         | 0         |
| Com_restore_table                 | 0         |
| Com_revoke                        | 0         |
| Com_revoke_all                    | 0         |
| Com_rollback                      | 0         |
| Com_rollback_to_savepoint         | 0         |
| Com_savepoint                     | 0         |
| Com_select                        | 7         |
| Com_set_option                    | 0         |
| Com_show_authors                  | 0         |
| Com_show_binlog_events            | 0         |
| Com_show_binlogs                  | 0         |
| Com_show_charsets                 | 0         |
| Com_show_collations               | 0         |
| Com_show_column_types             | 0         |
| Com_show_contributors             | 0         |
| Com_show_create_db                | 0         |
| Com_show_create_event             | 0         |
```

```
| Com_show_create_func               | 0         |
| Com_show_create_proc               | 0         |
| Com_show_create_table              | 0         |
| Com_show_create_trigger            | 0         |
| Com_show_databases                 | 3         |
| Com_show_engine_logs               | 0         |
| Com_show_engine_mutex              | 0         |
| Com_show_engine_status             | 0         |
| Com_show_events                    | 0         |
| Com_show_errors                    | 0         |
| Com_show_fields                    | 30        |
| Com_show_function_status           | 0         |
| Com_show_grants                    | 0         |
| Com_show_keys                      | 1         |
| Com_show_master_status             | 0         |
| Com_show_new_master                | 0         |
| Com_show_open_tables               | 1         |
| Com_show_plugins                   | 0         |
| Com_show_privileges                | 0         |
| Com_show_procedure_status          | 0         |
| Com_show_processlist               | 0         |
| Com_show_profile                   | 0         |
| Com_show_profiles                  | 0         |
| Com_show_slave_hosts               | 0         |
| Com_show_slave_status              | 0         |
| Com_show_status                    | 1         |
| Com_show_storage_engines           | 0         |
| Com_show_table_status              | 1         |
| Com_show_tables                    | 3         |
| Com_show_triggers                  | 0         |
| Com_show_variables                 | 0         |
| Com_show_warnings                  | 0         |
| Com_slave_start                    | 0         |
| Com_slave_stop                     | 0         |
```

```
| Com_stmt_close                | 0         |
| Com_stmt_execute              | 0         |
| Com_stmt_fetch                | 0         |
| Com_stmt_prepare              | 0         |
| Com_stmt_reprepare            | 0         |
| Com_stmt_reset                | 0         |
| Com_stmt_send_long_data       | 0         |
| Com_truncate                  | 0         |
| Com_uninstall_plugin          | 0         |
| Com_unlock_tables             | 0         |
| Com_update                    | 0         |
| Com_update_multi              | 0         |
| Com_xa_commit                 | 0         |
| Com_xa_end                    | 0         |
| Com_xa_prepare                | 0         |
| Com_xa_recover                | 0         |
| Com_xa_rollback               | 0         |
| Com_xa_start                  | 0         |
| Compression                   | OFF       |
| Connections                   | 12        |
| Created_tmp_disk_tables       | 4         |
| Created_tmp_files             | 5         |
| Created_tmp_tables            | 15        |
| Delayed_errors                | 0         |
| Delayed_insert_threads        | 0         |
| Delayed_writes                | 0         |
| Flush_commands                | 1         |
| Handler_commit                | 0         |
| Handler_delete                | 0         |
| Handler_discover              | 0         |
| Handler_prepare               | 0         |
| Handler_read_first            | 0         |
| Handler_read_key              | 0         |
| Handler_read_next             | 0         |
```

```
| Handler_read_prev                 | 0         |
| Handler_read_rnd                  | 0         |
| Handler_read_rnd_next             | 125       |
| Handler_rollback                  | 0         |
| Handler_savepoint                 | 0         |
| Handler_savepoint_rollback        | 0         |
| Handler_update                    | 0         |
| Handler_write                     | 107       |
| Innodb_buffer_pool_pages_data     | 178       |
| Innodb_buffer_pool_pages_dirty    | 0         |
| Innodb_buffer_pool_pages_flushed  | 189       |
| Innodb_buffer_pool_pages_free     | 333       |
| Innodb_buffer_pool_pages_misc     | 1         |
| Innodb_buffer_pool_pages_total    | 512       |
| Innodb_buffer_pool_read_ahead_rnd | 0         |
| Innodb_buffer_pool_read_ahead_seq | 0         |
| Innodb_buffer_pool_read_requests  | 1403      |
| Innodb_buffer_pool_reads          | 0         |
| Innodb_buffer_pool_wait_free      | 0         |
| Innodb_buffer_pool_write_requests | 1174      |
| Innodb_data_fsyncs                | 16        |
| Innodb_data_pending_fsyncs        | 0         |
| Innodb_data_pending_reads         | 0         |
| Innodb_data_pending_writes        | 0         |
| Innodb_data_read                  | 0         |
| Innodb_data_reads                 | 0         |
| Innodb_data_writes                | 38        |
| Innodb_data_written               | 3400192   |
| Innodb_dblwr_pages_written        | 16        |
| Innodb_dblwr_writes               | 1         |
| Innodb_log_waits                  | 0         |
| Innodb_log_write_requests         | 77        |
| Innodb_log_writes                 | 4         |
| Innodb_os_log_fsyncs              | 10        |
```

```
| Innodb_os_log_pending_fsyncs      | 0         |
| Innodb_os_log_pending_writes      | 0         |
| Innodb_os_log_written             | 37888     |
| Innodb_page_size                  | 16384     |
| Innodb_pages_created              | 178       |
| Innodb_pages_read                 | 0         |
| Innodb_pages_written              | 189       |
| Innodb_row_lock_current_waits     | 0         |
| Innodb_row_lock_time              | 0         |
| Innodb_row_lock_time_avg          | 0         |
| Innodb_row_lock_time_max          | 0         |
| Innodb_row_lock_waits             | 0         |
| Innodb_rows_deleted               | 0         |
| Innodb_rows_inserted              | 0         |
| Innodb_rows_read                  | 0         |
| Innodb_rows_updated               | 0         |
| Key_blocks_not_flushed            | 0         |
| Key_blocks_unused                 | 7235      |
| Key_blocks_used                   | 10        |
| Key_read_requests                 | 91        |
| Key_reads                         | 4         |
| Key_write_requests                | 52        |
| Key_writes                        | 44        |
| Last_query_cost                   | 10.499000 |
| Max_used_connections              | 1         |
| Not_flushed_delayed_rows          | 0         |
| Open_files                        | 54        |
| Open_streams                      | 0         |
| Open_table_definitions            | 28        |
| Open_tables                       | 27        |
| Opened_files                      | 316       |
| Opened_table_definitions          | 0         |
| Opened_tables                     | 0         |
| Prepared_stmt_count               | 0         |
```

```
| Qcache_free_blocks                | 0         |
| Qcache_free_memory                | 0         |
| Qcache_hits                       | 0         |
| Qcache_inserts                    | 0         |
| Qcache_lowmem_prunes              | 0         |
| Qcache_not_cached                 | 0         |
| Qcache_queries_in_cache           | 0         |
| Qcache_total_blocks               | 0         |
| Queries                           | 321       |
| Questions                         | 48        |
| Rpl_status                        | NULL      |
| Select_full_join                  | 0         |
| Select_full_range_join            | 0         |
| Select_range                      | 0         |
| Select_range_check                | 0         |
| Select_scan                       | 14        |
| Slave_open_temp_tables            | 0         |
| Slave_retried_transactions        | 0         |
| Slave_running                     | OFF       |
| Slow_launch_threads               | 0         |
| Slow_queries                      | 0         |
| Sort_merge_passes                 | 0         |
| Sort_range                        | 0         |
| Sort_rows                         | 0         |
| Sort_scan                         | 0         |
| Ssl_accept_renegotiates           | 0         |
| Ssl_accepts                       | 0         |
| Ssl_callback_cache_hits           | 0         |
| Ssl_cipher                        |           |
| Ssl_cipher_list                   |           |
| Ssl_client_connects               | 0         |
| Ssl_connect_renegotiates          | 0         |
| Ssl_ctx_verify_depth              | 0         |
| Ssl_ctx_verify_mode               | 0         |
```

```
| Ssl_default_timeout               | 0         |
| Ssl_finished_accepts              | 0         |
| Ssl_finished_connects             | 0         |
| Ssl_session_cache_hits            | 0         |
| Ssl_session_cache_misses          | 0         |
| Ssl_session_cache_mode            | NONE      |
| Ssl_session_cache_overflows       | 0         |
| Ssl_session_cache_size            | 0         |
| Ssl_session_cache_timeouts        | 0         |
| Ssl_sessions_reused               | 0         |
| Ssl_used_session_cache_entries    | 0         |
| Ssl_verify_depth                  | 0         |
| Ssl_verify_mode                   | 0         |
| Ssl_version                       |           |
| Table_locks_immediate             | 97        |
| Table_locks_waited                | 0         |
| Tc_log_max_pages_used             | 0         |
| Tc_log_page_size                  | 0         |
| Tc_log_page_waits                 | 0         |
| Threads_cached                    | 0         |
| Threads_connected                 | 1         |
| Threads_created                   | 11        |
| Threads_running                   | 1         |
| Uptime                            | 329158    |
| Uptime_since_flush_status         | 329158    |
+-----------------------------------+-----------+
291 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-status.html.

SHOW VARIABLES

Cette commande prend la forme suivante :

```
SHOW [GLOBAL|SESSION] VARIABLES [LIKE '%motif%'];
```

Par exemple:

```
mysql> SHOW VARIABLES;
+---------------------------------------+-------------------------------------------------------------------
--------------------+
| Variable_name                         | Value
|
+---------------------------------------+-------------------------------------------------------------------
--------------------+
| auto_increment_increment               | 1
|
| auto_increment_offset                 | 1
|
| autocommit                            | ON
|
| automatic_sp_privileges               | ON
|
| back_log                              | 50
|
| basedir                               | /usr/
|
| big_tables                            | OFF
|
| binlog_cache_size                     | 32768
|
| binlog_direct_non_transactional_updates | OFF
|
| binlog_format                         | STATEMENT
```

```
|
| bulk_insert_buffer_size                 | 8388608
|
| character_set_client                    | latin1
|
| character_set_connection                | latin1
|
| character_set_database                  | latin1
|
| character_set_filesystem                | binary
|
| character_set_results                   | latin1
|
| character_set_server                    | latin1
|
| character_set_system                    | utf8
|
| character_sets_dir                      | /usr/share/mysql/charsets/
|
| collation_connection                    | latin1_swedish_ci
|
| collation_database                      | latin1_swedish_ci
|
| collation_server                        | latin1_swedish_ci
|
| completion_type                         | 0
|
| concurrent_insert                       | 1
|
| connect_timeout                         | 10
|
| datadir                                 | /var/lib/mysql/
|
| date_format                             | %Y-%m-%d
```

```
|
| datetime_format                         | %Y-%m-%d %H:%i:%s
|
| default_week_format                     | 0
|
| delay_key_write                         | ON
|
| delayed_insert_limit                    | 100
|
| delayed_insert_timeout                  | 300
|
| delayed_queue_size                      | 1000
|
| div_precision_increment                 | 4
|
| engine_condition_pushdown               | ON
|
| error_count                             | 0
|
| event_scheduler                         | OFF
|
| expire_logs_days                        | 0
|
| flush                                   | OFF
|
| flush_time                              | 0
|
| foreign_key_checks                      | ON
|
| ft_boolean_syntax                       | + -><()~*:""&|
|
| ft_max_word_len                         | 84
|
| ft_min_word_len                         | 4
```

```
|
| ft_query_expansion_limit                | 20
|
| ft_stopword_file                        | (built-in)
|
| general_log                             | OFF
|
| general_log_file                        | /var/run/mysqld/mysqld.log
|
| group_concat_max_len                    | 1024
|
| have_community_features                 | YES
|
| have_compress                           | YES
|
| have_crypt                              | YES
|
| have_csv                                | YES
|
| have_dynamic_loading                    | YES
|
| have_geometry                           | YES
|
| have_innodb                             | YES
|
| have_ndbcluster                         | NO
|
| have_openssl                            | DISABLED
|
| have_partitioning                       | YES
|
| have_query_cache                        | YES
|
| have_rtree_keys                         | YES
```

```
|
| have_ssl                                | DISABLED
|
| have_symlink                            | DISABLED
|
| hostname                                | centos
|
| identity                                | 0
|
| ignore_builtin_innodb                   | OFF
|
| init_connect                            |
|
| init_file                               |
|
| init_slave                              |
|
| innodb_adaptive_hash_index              | ON
|
| innodb_additional_mem_pool_size         | 1048576
|
| innodb_autoextend_increment             | 8
|
| innodb_autoinc_lock_mode                | 1
|
| innodb_buffer_pool_size                 | 8388608
|
| innodb_checksums                        | ON
|
| innodb_commit_concurrency               | 0
|
| innodb_concurrency_tickets              | 500
|
| innodb_data_file_path                   | ibdata1:10M:autoextend
```

```
|
| innodb_data_home_dir                    |
|
| innodb_doublewrite                      | ON
|
| innodb_fast_shutdown                    | 1
|
| innodb_file_io_threads                  | 4
|
| innodb_file_per_table                   | OFF
|
| innodb_flush_log_at_trx_commit          | 1
|
| innodb_flush_method                     |
|
| innodb_force_recovery                   | 0
|
| innodb_lock_wait_timeout                | 50
|
| innodb_locks_unsafe_for_binlog          | OFF
|
| innodb_log_buffer_size                  | 1048576
|
| innodb_log_file_size                    | 5242880
|
| innodb_log_files_in_group               | 2
|
| innodb_log_group_home_dir               | ./
|
| innodb_max_dirty_pages_pct              | 90
|
| innodb_max_purge_lag                    | 0
|
| innodb_mirrored_log_groups              | 1
```

```
|
| innodb_open_files                          | 300
|
| innodb_rollback_on_timeout                 | OFF
|
| innodb_stats_method                        | nulls_equal
|
| innodb_stats_on_metadata                   | ON
|
| innodb_support_xa                          | ON
|
| innodb_sync_spin_loops                     | 20
|
| innodb_table_locks                         | ON
|
| innodb_thread_concurrency                  | 8
|
| innodb_thread_sleep_delay                  | 10000
|
| innodb_use_legacy_cardinality_algorithm | ON
|
| insert_id                                  | 0
|
| interactive_timeout                        | 28800
|
| join_buffer_size                           | 131072
|
| keep_files_on_create                       | OFF
|
| key_buffer_size                            | 8384512
|
| key_cache_age_threshold                    | 300
|
| key_cache_block_size                       | 1024
```

```
|
| key_cache_division_limit                | 100
|
| language                                | /usr/share/mysql/english/
|
| large_files_support                     | ON
|
| large_page_size                         | 0
|
| large_pages                             | OFF
|
| last_insert_id                          | 0
|
| lc_time_names                           | en_US
|
| license                                 | GPL
|
| local_infile                            | ON
|
| locked_in_memory                        | OFF
|
| log                                     | OFF
|
| log_bin                                 | OFF
|
| log_bin_trust_function_creators         | OFF
|
| log_bin_trust_routine_creators          | OFF
|
| log_error                               | /var/log/mysqld.log
|
| log_output                              | FILE
|
| log_queries_not_using_indexes           | OFF
```

```
|
| log_slave_updates                       | OFF
|
| log_slow_queries                        | OFF
|
| log_warnings                            | 1
|
| long_query_time                         | 10.000000
|
| low_priority_updates                    | OFF
|
| lower_case_file_system                  | OFF
|
| lower_case_table_names                  | 0
|
| max_allowed_packet                      | 1048576
|
| max_binlog_cache_size                   | 4294963200
|
| max_binlog_size                         | 1073741824
|
| max_connect_errors                      | 10
|
| max_connections                         | 151
|
| max_delayed_threads                     | 20
|
| max_error_count                         | 64
|
| max_heap_table_size                     | 16777216
|
| max_insert_delayed_threads              | 20
|
| max_join_size                           | 18446744073709551615
```

```
|
| max_length_for_sort_data                | 1024
|
| max_long_data_size                      | 1048576
|
| max_prepared_stmt_count                 | 16382
|
| max_relay_log_size                      | 0
|
| max_seeks_for_key                       | 4294967295
|
| max_sort_length                         | 1024
|
| max_sp_recursion_depth                  | 0
|
| max_tmp_tables                          | 32
|
| max_user_connections                    | 0
|
| max_write_lock_count                    | 4294967295
|
| min_examined_row_limit                  | 0
|
| multi_range_count                       | 256
|
| myisam_data_pointer_size                | 6
|
| myisam_max_sort_file_size               | 2146435072
|
| myisam_mmap_size                        | 4294967295
|
| myisam_recover_options                  | OFF
|
| myisam_repair_threads                   | 1
```

```
|
| myisam_sort_buffer_size                 | 8388608
|
| myisam_stats_method                     | nulls_unequal
|
| myisam_use_mmap                         | OFF
|
| net_buffer_length                       | 16384
|
| net_read_timeout                        | 30
|
| net_retry_count                         | 10
|
| net_write_timeout                       | 60
|
| new                                     | OFF
|
| old                                     | OFF
|
| old_alter_table                         | OFF
|
| old_passwords                           | OFF
|
| open_files_limit                        | 1024
|
| optimizer_prune_level                   | 1
|
| optimizer_search_depth                  | 62
|
| optimizer_switch                        |
index_merge=on,index_merge_union=on,index_merge_sort_union=on,index_merge_intersection=on |
| pid_file                                | /var/run/mysqld/mysqld.pid
|
| plugin_dir                              | /usr/lib/mysql/plugin
```

```
|
| port                                | 3306
|
| preload_buffer_size                 | 32768
|
| profiling                           | OFF
|
| profiling_history_size              | 15
|
| protocol_version                    | 10
|
| pseudo_thread_id                    | 11
|
| query_alloc_block_size              | 8192
|
| query_cache_limit                   | 1048576
|
| query_cache_min_res_unit            | 4096
|
| query_cache_size                    | 0
|
| query_cache_type                    | ON
|
| query_cache_wlock_invalidate        | OFF
|
| query_prealloc_size                 | 8192
|
| rand_seed1                          |
|
| rand_seed2                          |
|
| range_alloc_block_size              | 4096
|
| read_buffer_size                    | 131072
```

```
|
| read_only                                 | OFF
|
| read_rnd_buffer_size                      | 262144
|
| relay_log                                 |
|
| relay_log_index                           |
|
| relay_log_info_file                       | relay-log.info
|
| relay_log_purge                           | ON
|
| relay_log_space_limit                     | 0
|
| report_host                               |
|
| report_password                           |
|
| report_port                               | 3306
|
| report_user                               |
|
| rpl_recovery_rank                         | 0
|
| secure_auth                               | OFF
|
| secure_file_priv                          |
|
| server_id                                 | 0
|
| skip_external_locking                     | ON
|
| skip_name_resolve                         | OFF
```

```
|
| skip_networking                         | OFF
|
| skip_show_database                      | OFF
|
| slave_compressed_protocol               | OFF
|
| slave_exec_mode                         | STRICT
|
| slave_load_tmpdir                       | /tmp
|
| slave_net_timeout                       | 3600
|
| slave_skip_errors                       | OFF
|
| slave_transaction_retries               | 10
|
| slow_launch_time                        | 2
|
| slow_query_log                          | OFF
|
| slow_query_log_file                     | /var/run/mysqld/mysqld-slow.log
|
| socket                                  | /var/lib/mysql/mysql.sock
|
| sort_buffer_size                        | 2097144
|
| sql_auto_is_null                        | ON
|
| sql_big_selects                         | ON
|
| sql_big_tables                          | OFF
|
| sql_buffer_result                       | OFF
```

```
|
| sql_log_bin                          | ON
|
| sql_log_off                          | OFF
|
| sql_log_update                       | ON
|
| sql_low_priority_updates             | OFF
|
| sql_max_join_size                    | 18446744073709551615
|
| sql_mode                             |
|
| sql_notes                            | ON
|
| sql_quote_show_create                | ON
|
| sql_safe_updates                     | OFF
|
| sql_select_limit                     | 18446744073709551615
|
| sql_slave_skip_counter               |
|
| sql_warnings                         | OFF
|
| ssl_ca                               |
|
| ssl_capath                           |
|
| ssl_cert                             |
|
| ssl_cipher                           |
|
| ssl_key                              |
```

```
|
| storage_engine                          | MyISAM
|
| sync_binlog                             | 0
|
| sync_frm                                | ON
|
| system_time_zone                        | CEST
|
| table_definition_cache                  | 256
|
| table_lock_wait_timeout                 | 50
|
| table_open_cache                        | 64
|
| table_type                              | MyISAM
|
| thread_cache_size                       | 0
|
| thread_handling                         | one-thread-per-connection
|
| thread_stack                            | 196608
|
| time_format                             | %H:%i:%s
|
| time_zone                               | SYSTEM
|
| timed_mutexes                           | OFF
|
| timestamp                               | 1350980852
|
| tmp_table_size                          | 16777216
|
| tmpdir                                  | /tmp
```

```
|
| transaction_alloc_block_size            | 8192
|
| transaction_prealloc_size               | 4096
|
| tx_isolation                            | REPEATABLE-READ
|
| unique_checks                           | ON
|
| updatable_views_with_limit              | YES
|
| version                                 | 5.1.61
|
| version_comment                         | Source distribution
|
| version_compile_machine                 | i386
|
| version_compile_os                      | redhat-linux-gnu
|
| wait_timeout                            | 28800
|
| warning_count                           | 0
|
+-----------------------------------------+-----------------------------------------------------------------------
----------------------+
276 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-variables.html.

SHOW PROCESSLIST

Cette commande prend la forme suivante :

```
SHOW [FULL] PROCESSLIST;
```

Par exemple:

```
mysql> SHOW  PROCESSLIST;
+----+------+-----------+--------+---------+------+-------+------------------+
| Id | User | Host      | db     | Command | Time | State | Info             |
+----+------+-----------+--------+---------+------+-------+------------------+
| 11 | root | localhost | ligue1 | Query   |    0 | NULL  | SHOW  PROCESSLIST |
+----+------+-----------+--------+---------+------+-------+------------------+
1 row in set (0.00 sec)

mysql> SHOW FULL PROCESSLIST;
+----+------+-----------+--------+---------+------+-------+----------------------+
| Id | User | Host      | db     | Command | Time | State | Info                 |
+----+------+-----------+--------+---------+------+-------+----------------------+
| 11 | root | localhost | ligue1 | Query   |    0 | NULL  | SHOW FULL PROCESSLIST |
+----+------+-----------+--------+---------+------+-------+----------------------+
1 row in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-processlist.html.

SHOW GRANTS

Cette commande prend la forme suivante :

```
SHOW GRANTS FOR utilisateur;
```

Par exemple:

```
mysql> SHOW GRANTS;
+-----------------------------------------------------------------------------------------------------------------
------------------------+
| Grants for root@localhost
|
+-----------------------------------------------------------------------------------------------------------------
------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY PASSWORD
'*F3E048E28ED63F281CF8A403F96F5D283C8700E6' WITH GRANT OPTION |
+-----------------------------------------------------------------------------------------------------------------
------------------------+
1 row in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-grants.html.

SHOW CREATE TABLE

Cette commande prend la forme suivante :

```
SHOW CREATE TABLE nomtable[\G];
```

Par exemple:

```
mysql> SHOW CREATE TABLE equipe;
+--------+-----------------------------------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
| Table  | Create Table
|
+--------+----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
| equipe | CREATE TABLE `equipe` (
  `id_equipe` int(11) NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) NOT NULL,
  `stade` varchar(50) NOT NULL,
  `ville` varchar(30) NOT NULL,
  `points` int(11) NOT NULL DEFAULT '0',
  `buts` int(11) NOT NULL DEFAULT '0',
  `entraineur` varchar(100) DEFAULT 'inconnu',
  `tel_entraineur` varchar(20) DEFAULT 'inconnu',
  PRIMARY KEY (`id_equipe`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1 |
+--------+----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

ou

```
mysql> SHOW CREATE TABLE equipe\G;
*************************** 1. row ***************************
       Table: equipe
Create Table: CREATE TABLE `equipe` (
```

```
  `id_equipe` int(11) NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) NOT NULL,
  `stade` varchar(50) NOT NULL,
  `ville` varchar(30) NOT NULL,
  `points` int(11) NOT NULL DEFAULT '0',
  `buts` int(11) NOT NULL DEFAULT '0',
  `entraineur` varchar(100) DEFAULT 'inconnu',
  `tel_entraineur` varchar(20) DEFAULT 'inconnu',
  PRIMARY KEY (`id_equipe`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

ERROR:
No query specified

mysql>
```

Notez l'utilisation de **\G** pour rendre la sortie plus lisible.

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/show-create-table.html.

La Commande SET

La commande SET permet de spécifier les valeurs des options du serveur ou des programmes clients.

Par example la saisie de la commande suivante démontre que la valeur de max_connections est de **100**:

```
mysql> SHOW VARIABLES LIKE 'max_connections';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| max_connections | 151   |
+-----------------+-------+
1 row in set (0.00 sec)

mysql>
```

Pour modifier cette valeur à **200**, saisissez la commande suivante:

```
mysql> SET GLOBAL max_connections = 200;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'max_connections';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| max_connections | 200   |
+-----------------+-------+
1 row in set (0.00 sec)

mysql>
```

Notez que dans ce cas il s'agit d'une variable globale. Dans le cas d'une variable de **SESSION**, il convient de modifier la commande à **SET SESSION**.

La Commande KILL

La commande **KILL** permet d'interrompre un thread en cours. Elle prend la forme suivante.

```
mysql> KILL idThread; [Entrée]
```

Pour comprendre son utilisation, créez d'abord un utilisateur :

```
mysql> GRANT USAGE ON * TO fenestros@localhost IDENTIFIED BY 'fenestros';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Accordez les privilèges de connexion à l'utilisateur fenestros:

```
mysql> GRANT USAGE ON * TO fenestros@localhost IDENTIFIED BY 'fenestros';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS for fenestros@localhost;
+-----------------------------------------------------------------------------------------------------------
--+
| Grants for fenestros@localhost
|
+-----------------------------------------------------------------------------------------------------------
--+
| GRANT USAGE ON *.* TO 'fenestros'@'localhost' IDENTIFIED BY PASSWORD
'*00269BA49BEC800F9CCF34C20C1FD83E0236B89A' |
+-----------------------------------------------------------------------------------------------------------
--+
1 row in set (0.00 sec)

mysql>
```

Connectez-vous à votre serveur MySQL avec le compte de **fenestros sans vous déconnecter en tant que root** et saisissez une commande pour vérifiez que tout va bien:

```
[trainee@centos6 ~]$ mysql -u fenestros -p
Enter password: fenestros
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
+--------------------+
2 rows in set (0.01 sec)

mysql>
```

Notez que le mote de passe ne sera **pas** visible.

En tant que root, saisissez la commande suivante:

```
mysql> SHOW PROCESSLIST;
+----+-----------+-----------+--------+---------+------+-------+------------------+
| Id | User      | Host      | db     | Command | Time | State | Info             |
+----+-----------+-----------+--------+---------+------+-------+------------------+
| 11 | root      | localhost | ligue1 | Query   |    0 | NULL  | SHOW PROCESSLIST |
| 12 | fenestros | localhost | NULL   | Sleep   |   65 |       | NULL             |
+----+-----------+-----------+--------+---------+------+-------+------------------+
2 rows in set (0.00 sec)

mysql>
```

Notez bien la présence de l'utilisateur **fenestros**.

Pour interrompre le thread de **fenestros**, saisissez la commande KILL suivi de la valeur de l'**Id** correspondante :

```
mysql> KILL 12;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW PROCESSLIST;
+----+------+-----------+--------+---------+------+-------+------------------+
| Id | User | Host      | db     | Command | Time | State | Info             |
+----+------+-----------+--------+---------+------+-------+------------------+
| 11 | root | localhost | ligue1 | Query   |    0 | NULL  | SHOW PROCESSLIST |
+----+------+-----------+--------+---------+------+-------+------------------+
1 row in set (0.00 sec)

mysql>
```

Notez que le thread de l'utilisateur **fenestros** a été terminée.

Consultez maintenant la connexion de fenestros au serveur MySQL et lancez la commande suivante:

```
mysql> SHOW DATABASES;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:    13
Current database: *** NONE ***

+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
+--------------------+
2 rows in set (0.01 sec)

mysql>
```

Notez que le client se reconnecte au serveur!

Contrôlez maintenant la PROCESSLIST à partir de la connexion de **root** :

```
mysql> SHOW PROCESSLIST;
+----+-----------+-----------+--------+---------+------+-------+------------------+
| Id | User      | Host      | db     | Command | Time | State | Info             |
+----+-----------+-----------+--------+---------+------+-------+------------------+
| 11 | root      | localhost | ligue1 | Query   |    0 | NULL  | SHOW PROCESSLIST |
| 13 | fenestros | localhost | NULL   | Sleep   |   32 |       | NULL             |
+----+-----------+-----------+--------+---------+------+-------+------------------+
2 rows in set (0.00 sec)
```

```
mysql>
```

Voici la même commande à partir de la connexion de **fenestros** :

```
mysql> SHOW PROCESSLIST;
+----+-----------+-----------+------+---------+------+-------+------------------+
| Id | User      | Host      | db   | Command | Time | State | Info             |
+----+-----------+-----------+------+---------+------+-------+------------------+
| 13 | fenestros | localhost | NULL | Query   |    0 | NULL  | SHOW PROCESSLIST |
+----+-----------+-----------+------+---------+------+-------+------------------+
1 row in set (0.00 sec)

mysql>
```

Notez qu'un autre thread pour la connexion de fenestros a été démarré.

La Commande FLUSH

La commande FLUSH permet de vider les caches du serveur MySQL.

FLUSH HOSTS

Cette commande ré-initialise le cache des adresse IP des clients:

```
mysql> FLUSH HOSTS; [Entrée]
```

FLUSH LOGS

Cette commande ferme les fichiers de logs et crée de nouveaux:

```
mysql> FLUSH LOGS; [Entrée]
```

FLUSH PRIVILEGES

Cette commande recharge la table de privileges:

```
mysql> FLUSH PRIVILEGES; [Entrée]
```

FLUSH TABLES

Cette commande vide les caches des tables et les ferme:

```
mysql> FLUSH TABLES; [Entrée]
```

En cas de tables multiples, il convient d'utiliser la commande suivante:

```
mysql> FLUSH [TABLE|TABLES] nomtable [,nomtable]; [Entrée]
```

La base INFORMATION_SCHEMA

Les commandes SHOW ne correspondent pas aux standards SQL2003. **MySQL 5.x** fournit une base spécifique appelée **INFORMATION_SCHEMA** qui contient les **Méta-données** :

```
mysql> USE information_schema
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+---------------------------------------+
| Tables_in_information_schema          |
+---------------------------------------+
```

```
| CHARACTER_SETS                          |
| COLLATIONS                              |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                                 |
| COLUMN_PRIVILEGES                       |
| ENGINES                                 |
| EVENTS                                  |
| FILES                                   |
| GLOBAL_STATUS                           |
| GLOBAL_VARIABLES                        |
| KEY_COLUMN_USAGE                        |
| PARTITIONS                              |
| PLUGINS                                 |
| PROCESSLIST                             |
| PROFILING                               |
| REFERENTIAL_CONSTRAINTS                 |
| ROUTINES                                |
| SCHEMATA                                |
| SCHEMA_PRIVILEGES                       |
| SESSION_STATUS                          |
| SESSION_VARIABLES                       |
| STATISTICS                              |
| TABLES                                  |
| TABLE_CONSTRAINTS                       |
| TABLE_PRIVILEGES                        |
| TRIGGERS                                |
| USER_PRIVILEGES                         |
| VIEWS                                   |
+-----------------------------------------+
28 rows in set (0.00 sec)

mysql>
```

Chaque table de cette base fournit des informations spécifiques, par exemple :

TABLE	Contenu
CHARACTER_SETS	Informations sur jeux de caractères.
COLLATIONS	Informations sur interclassements.
COLLATION_CHARACTER_SET_APPLICABILITY	Correspondances interclassements / charsets.
COLUMNS	Informations sur leurs colonnes.
COLUMN_PRIVILEGES	Informations sur les privilèges sur les colonnes.
KEY_COLUMN_USAGE	clés primaires et étrangères.
ROUTINES	Informations sur les procédures et fonctions stockées.
SCHEMATA	Informations sur les bases (ou schémas).
SCHEMA_PRIVILEGES	Informations sur privilèges sur les bases.
STATISTICS	Informations sur les indexes.
TABLES	Informations sur les tables.
TABLE_CONSTRAINTS	Informations sur les contraintes des tables.
TABLE_PRIVILEGES	Informations sur les privilèges sur les tables.
TRIGGERS	Informations sur les déclencheurs.
USER_PRIVILEGES	Informations sur les privilèges globaux.
VIEWS	Informations sur les vues.

Maintenance

Instructions SQL

CHECK TABLE

Cette commande fonctionne avec les moteurs de type **MyISAM**, Archive, CSV et **InnoDB**. La commande peut prendre 5 **options** :

Option	Description
QUICK	Ne recherche pas d'enregistrements orphelins
FAST	Ne vérifie que les tables qui n'ont pas été fermées correctement
CHANGED	Ne vérifie que les tables ayant subi une modification depuis la dernière vérification

Option	Description
MEDIUM	Vérifie les enregistrements et génère une cléf d'intégrité (Checksum). Si aucune option n'est spécifiée, MEDIUM est l'option par défaut.
EXTENDED	Vérifie les enregistrements et génère une cléf d'intégrité (Checksum) **par** enregistrement

Vérifiez les tables de votre base ligue1 ;

```
mysql> CHECK TABLE ligue1.equipe, ligue1.rencontre;
+------------------+-------+----------+----------+
| Table            | Op    | Msg_type | Msg_text |
+------------------+-------+----------+----------+
| ligue1.equipe    | check | status   | OK       |
| ligue1.rencontre | check | status   | OK       |
+------------------+-------+----------+----------+
2 rows in set (0.00 sec)

mysql>
```

Vérifiez les tables de votre base ligue1 avec l'option **FAST**;

```
mysql> CHECK TABLE ligue1.equipe, ligue1.rencontre FAST;
+------------------+-------+----------+----------------------------+
| Table            | Op    | Msg_type | Msg_text                   |
+------------------+-------+----------+----------------------------+
| ligue1.equipe    | check | status   | Table is already up to date |
| ligue1.rencontre | check | status   | Table is already up to date |
+------------------+-------+----------+----------------------------+
2 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/check-table.html.

REPAIR TABLE

Dans certains cas, la commande CHECK TABLE relève des erreurs. Dans le cas précis d'une erreur du type **warning** due à la non-fermeture d'une table, il convient de fermer toutes les tables en utilisant la commande :

```
mysql> FLUSH TABLES; [Entrée]
```

Dans d'autres cas plus sérieux, on fait appelle à la commande **REPAIR TABLE**. La commande REPAIR TABLE ne fonctionne qu'avec des moteurs **MyISAM**, Archive et CSV. La commande peut prendre 4 **options** :

Option	Description
NO_WRITE_TO_BINLOG	N'écrit pas la commande dans le journal binaire
QUICK	Ne répare que l'index
EXTENDED	Gére des index de chaînes longues
USE_FRM	Permet de recréer un fichier d'index (.MYI) à partir des informations dans le fichier .frm. Attention : La valeur auto-incrément actuel est perdu et les tables compressées sont supprimées

Saisissez la commande suivante :

```
mysql> REPAIR TABLE ligue1.equipe, ligue1.rencontre;
+------------------+--------+----------+----------+
| Table            | Op     | Msg_type | Msg_text |
+------------------+--------+----------+----------+
| ligue1.equipe    | repair | status   | OK       |
| ligue1.rencontre | repair | status   | OK       |
+------------------+--------+----------+----------+
2 rows in set (0.00 sec)

mysql>
```

Les actions de maintenance peuvent être automatisées par l'addition de la directive **myisam-recover = OPTION** au fichier my.cnf. Cette directive force le serveur d'exécuter un CHECK TABLE dans le cas où une table est marquée **crashed** ou si elle a été mal fermée. Dans le

cas de problèmes le serveur exécutera un REPAIR TABLE. La directive peut prendre 4 options :

- BACKUP : génère une sauvegarde au format nom_table-dateheure.BAK du fichier des données dans le cas où la réparation nécessite la modification d'enregistrements,
- FORCE : force la réparation même si des enregistrements doivent être supprimés,
- QUICK : les tables sans trous sont ignorés,
- DEFAULT : tente de restaurer la table.

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/repair-table.html.

OPTIMIZE TABLE

La commande OPTIMIZE TABLE permet de défragmenter une table ou plusieurs tables. La commande OPTIMIZE TABLE ne fonctionne qu'avec le moteur **MyISAM**.

Saisissez la commande suivante :

```
mysql> OPTIMIZE TABLE ligue1.equipe, ligue1.rencontre;
+------------------+----------+----------+----------+
| Table            | Op       | Msg_type | Msg_text |
+------------------+----------+----------+----------+
| ligue1.equipe    | optimize | status   | OK       |
| ligue1.rencontre | optimize | status   | OK       |
+------------------+----------+----------+----------+
2 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/optimize-table.html.

ANALYZE TABLE

La commande ANALYZE TABLE permet d'analyser et d'enregistrer des statistiques sur une table ou plusieurs tables. La commande ANALYZE TABLE ne fonctionne qu'avec des tables au format **MyISAM** et **BDB**.

Saisissez la commande suivante :

```
mysql> ANALYZE TABLE ligue1.equipe, ligue1.rencontre;
+------------------+---------+----------+----------------------------+
| Table            | Op      | Msg_type | Msg_text                   |
+------------------+---------+----------+----------------------------+
| ligue1.equipe    | analyze | status   | Table is already up to date |
| ligue1.rencontre | analyze | status   | Table is already up to date |
+------------------+---------+----------+----------------------------+
2 rows in set (0.00 sec)

mysql>
```

Pour plus d'information, voir http://dev.mysql.com/doc/refman/5.0/fr/analyze-table.html.

myisamchk

Le programme **myisamchk** permet d'exécuter en ligne de commande à partir du shell les opérations de maintenance. Comme son nom indique, il ne

peut être utilisé qu'avec des tables au format **MyISAM**. Le programme myisamchk doit être utilisé avec le serveur mysqld **arreté**.

Sortez de mysql et **arretez le serveur mysqld** :

```
mysql> exit
Bye
[root@centos6 ~]# service mysqld stop
Arrêt de MySQL :                                           [  OK  ]
[root@centos6 ~]#
```

Saisissez ensuite la commande suivante pour vérifier tous les index de la base **ligue1** :

```
[root@centos6 ~]# myisamchk /var/lib/mysql/ligue1/*MYI
Checking MyISAM file: /var/lib/mysql/ligue1/equipe.MYI
Data records:       4   Deleted blocks:       0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check record links
---------

Checking MyISAM file: /var/lib/mysql/ligue1/rencontre.MYI
Data records:       6   Deleted blocks:       0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check record links
```

```
C:\ProgramData\MySQL\MySQL Server 5.6\data>"C:\Program Files\MySQL\MySQL Server 5.6\bin\myisamchk.exe"
"C:\ProgramData\MySQL\MySQL Server 5.6\data\ligue1\
```

```
MYI"
Checking MyISAM file: C:\ProgramData\MySQL\MySQL Server 5.6\data\ligue1\equipe.MYI
Data records:       4   Deleted blocks:       0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check record links

---------

Checking MyISAM file: C:\ProgramData\MySQL\MySQL Server 5.6\data\ligue1\rencontre.MYI
Data records:       6   Deleted blocks:       0
- check file-size
- check record delete-chain
- check key delete-chain
- check index reference
- check data record references index: 1
- check record links

C:\ProgramData\MySQL\MySQL Server 5.6\data>
```

La commande myisamchk peut prendre plusieurs options. Nous détaillons ici que les principales :

Option	**Description**
-F ou - -fast	Ne vérifie que les tables fermées
-C ou - -check-only-changed	Ne vérifie que les tables ayant subi une modification depuis la dernière vérification
-i ou - -information	Permet de montrer les statistiques concernant la vérification
-r ou - -recover	Permet de réparer une table
-a ou - -analyze	Permet d'analyser une table
-d ou - -description	Permet de recevoir des informations sur une table
-S ou - -sort-index	Permet de trier les index pour optimiser des requêtes

mysqlcheck

Le programme **mysqlcheck** permet d'effectuer la maintenance des tables pendant que le serveur mysqld est en fonctionnement. Il ne peut être utilisé qu'avec des tables au format **MyISAM**.

Démarrez le serveur **mysqld** et saisissez la commande suivante :

```
[root@redhat ~]# mysqlcheck -u root -p --host=127.0.0.1 ligue1
Enter password: fenestros1
ligue1.equipe                                      OK
ligue1.rencontre                                   OK
```

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqlcheck -u root -p --host=127.0.0.1 ligue1
Enter password: **********
ligue1.equipe                                      OK
ligue1.rencontre                                   OK

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

La commande mysqlcheck peut prendre plusieurs options. Nous détaillons ici que les principales :

Option	**Description**
-F ou - -fast	Ne vérifie que les tables mal fermées
-C ou - -check-only-changed	Ne vérifie que les tables ayant subi une modification depuis la dernière vérification
-r ou - -recover	Permet de réparer une table
-a ou - -analyze	Permet d'analyser une table
-c ou - -check	Permet de vérifier une table. Ce'est l'option par défaut
-o ou - -optimize	Permet d'optimiser une table
-A ou - -all-databases	Permet de traiter toutes les bases de données
-B ou - -databases	Permet de spécifier les bases de données à vérifier

Supervision

MySQLReport

mysqlreport est un script perl qui regroupe des valeurs des principales variables de statut dans un rapport court et lisible. Pour obtenir ce script, utilisez le lien ci-dessous :

```
[root@centos6 ~]# wget hackmysql.com/scripts/mysqlreport
--2014-02-11 13:46:42--  http://hackmysql.com/scripts/mysqlreport
Résolution de hackmysql.com... 64.13.235.8
Connexion vers hackmysql.com|64.13.235.8|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 38873 (38K) [application/x-perl]
Sauvegarde en : «mysqlreport»

100%[=============================================================================================
==================>] 38 873       105K/s   ds 0,4s

2014-02-11 13:46:48 (105 KB/s) - «mysqlreport» sauvegardé [38873/38873]

[root@centos6 ~]# ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog  mysqlreport
[root@centos6 ~]# chmod u+x mysqlreport
[root@centos6 ~]#
```

Le script s'utilise simplement en l’exécutant avec deux arguments :

```
[root@centos6 ~]# ./mysqlreport --user=root --password
Password for database user root:
Use of uninitialized value $is in multiplication (*) at ./mysqlreport line 829, <STDIN> line 1.
Use of uninitialized value in formline at ./mysqlreport line 1227, <STDIN> line 1.
Use of uninitialized value in formline at ./mysqlreport line 1235, <STDIN> line 1.
```

```
MySQL 5.5.33-cll-lve     uptime 0 2:43:53       Tue Feb 11 13:50:33 2014

__ Key _________________________________________________________________
Buffer used     2.00k of   8.00M  %Used:   0.02
  Current     946.00k            %Usage:  11.55
Write hit       0.00%
Read hit      100.00%

__ Questions ___________________________________________________________
Total              38     0.0/s
  DMS           23.57k     2.4/s  %Total: 62028.
  -Unknown      23.56k     2.4/s          62010.
  Com_              21     0.0/s           55.26
  COM_QUIT          10     0.0/s           26.32
Slow 10 s            1     0.0/s            2.63  %DMS:   0.00  Log: OFF
DMS             23.57k     2.4/s          62028.
  INSERT        23.56k     2.4/s          62005.         99.96
  SELECT             9     0.0/s           23.68          0.04
  REPLACE            0       0/s            0.00          0.00
  DELETE             0       0/s            0.00          0.00
  UPDATE             0       0/s            0.00          0.00
Com_                21     0.0/s           55.26
  show_variab        4     0.0/s           10.53
  show_tables        2     0.0/s            5.26
  show_status        2     0.0/s            5.26

__ SELECT and Sort _____________________________________________________
Scan               13     0.0/s %SELECT: 144.44
Range               0       0/s            0.00
Full join           0       0/s            0.00
Range check         0       0/s            0.00
Full rng join       0       0/s            0.00
Sort scan           0       0/s
Sort range          0       0/s
```

```
Sort mrg pass       0       0/s

__ Table Locks _________________________________________________________
Waited              0       0/s  %Total:   0.00
Immediate      23.61k     2.4/s

__ Tables ______________________________________________________________
Open               43 of  400    %Cache:  10.75
Opened             52     0.0/s

__ Connections _________________________________________________________
Max used            2 of  151      %Max:   1.32
Total              12     0.0/s

__ Created Temp ________________________________________________________
Disk table         10     0.0/s
Table              56     0.0/s    Size:  16.0M
File                7     0.0/s

__ Threads _____________________________________________________________
Running             1 of    1
Cached              0 of    0      %Hit:   8.33
Created            11     0.0/s
Slow                0       0/s

__ Aborted _____________________________________________________________
Clients             0       0/s
Connects            2     0.0/s

__ Bytes _______________________________________________________________
Sent           34.05k     3.5/s
Received        2.79k     0.3/s

__ InnoDB Buffer Pool __________________________________________________
```

```
Usage           9.11M of 127.98M  %Used:   7.12
Read hit      100.00%
Pages
  Free          7.61k            %Total:  92.88
  Data            582                     7.11 %Drty:   0.00
  Misc              1                     0.01
  Latched                                 0.00
Reads         228.06k    23.2/s
  From file           0       0/s           0.00
  Ahead Rnd           0       0/s
  Ahead Sql               0/s
Writes        146.43k    14.9/s
Flushes         1.02k     0.1/s
Wait Free           0       0/s

__ InnoDB Lock _____________________________________________________________
Waits               0       0/s
Current             0
Time acquiring
  Total             0 ms
  Average           0 ms
  Max               0 ms

__ InnoDB Data, Pages, Rows ________________________________________________
Data
  Reads             0       0/s
  Writes       24.71k     2.5/s
  fsync        23.68k     2.4/s
  Pending
    Reads           0
    Writes          0
    fsync           0

Pages
```

```
  Created          582     0.1/s
  Read               0       0/s
  Written       1.02k     0.1/s

Rows
  Deleted            0       0/s
  Inserted      47.12k     4.8/s
  Read          47.12k     4.8/s
  Updated            0       0/s
```

Consultez **ce lien** pour comprendre le rapport généré par mysqlreport.

Sauvegardes et Replication

Sauvegardes

mysqlhotcopy

mysqlhotcopy est un script perl qui **ne fonctionne que sous Linux**. Ce script permet de prendre une copie **à chaud** d'une ou plusieurs bases de données.

```
[root@centos6 ~]# which mysqlhotcopy
/usr/bin/mysqlhotcopy
```

Pour exécuter le script en sauvegardant une base vers le répertoire /tmp, saisissez la commande suivante:

```
[root@centos6 ~]# mysqlhotcopy -u root -p fenestros1 ligue1 /tmp
Locked 3 tables in 0 seconds.
```

```
Flushed tables (`ligue1`.`V_EQUIPE`, `ligue1`.`equipe`, `ligue1`.`rencontre`) in 0 seconds.
Copying 12 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 3 tables (12 files) in 0 seconds (0 seconds overall).
```

Consultez le répertoire /tmp:

```
[root@centos6 ~]# ls -l /tmp | grep ligue
drwxr-x---. 2 mysql   mysql     4096 25 oct.  14:22 ligue1
```

En effet **mysqlhotcopy** a créé un répertoire au nom de la base de données. A l'intérieur de ce répertoire, vous constaterez la présence des fichiers de la base de données:

```
[root@centos6 ~]# ls -l /tmp/ligue1/
total 64
-rw-rw----. 1 mysql mysql   65 22 oct.  14:33 db.opt
-rw-rw----. 1 mysql mysql 8818 23 oct.  09:57 equipe.frm
-rw-rw----. 1 mysql mysql  296 25 oct.  14:05 equipe.MYD
-rw-rw----. 1 mysql mysql 2048 25 oct.  14:08 equipe.MYI
-rw-rw----. 1 mysql mysql  301 23 oct.  09:43 equipe.TRG
-rw-rw----. 1 mysql mysql 8800 22 oct.  14:38 rencontre.frm
-rw-rw----. 1 mysql mysql  120 25 oct.  14:05 rencontre.MYD
-rw-rw----. 1 mysql mysql 2048 25 oct.  14:08 rencontre.MYI
-rw-rw----. 1 mysql mysql  607 23 oct.  09:44 rencontre.TRG
-rw-rw----. 1 mysql mysql   38 23 oct.  09:43 TGR_BI_EQUIPE.TRN
-rw-rw----. 1 mysql mysql   41 23 oct.  09:44 TGR_BU_RENCONTRE.TRN
-rw-rw----. 1 mysql mysql  953 23 oct.  09:58 V_EQUIPE.frm
```

mysqldump

MySQL vous permet de traiter des commandes SQL se trouvant dans un fichier. Pour cela lancez MySQL de la manière suivante :

```
# mysql [options] database < file.sql
```

Toutes les commandes SQL contenues dans file.sql seront alors appliquées à la base de données spécifiée. Le fichier ne doit pas nécessairement se terminer par .sql – ce qui compte, c'est le contenu. Il faut absolument que les commandes soient délimitées par des points-virgules. Si vous le souhaitez, vous pouvez configurer un autre séparateur de commandes à l'aide de la commande DELIMITER character. Cela s'avère nécessaire lorsque les fichiers SQL contient des commandes CREATE FUNCTION, PROCEDURE ou TRIGGER, qui contiennent des points-virgules.

Sauvegarder avec mysqldump

Mysqldump crée une sauvegarde d'une base de données. Le fichier de sauvegarde résultant contient des commandes SQL permettant de créer des tables et d'ajouter des données. La syntaxe de mysqldump est la suivante :

```
mysqldump [option] dbname > backupfile.sql
```

> Les **routines** ne sont pas sauvegardées par défaut lors de l'utilisation de la commande mysqldump. Pour les sauvegarder il convient de passer l'option **–routines** à mysqldump.

Restauration avec mysqldump

Dans la pratique, vous rencontrerez souvent les fichiers *.sql comportant une sauvegarde de bases de données créée avec **mysqldump**. Pour lire ces fichiers, vous devez normalement créé la nouvelle base de données avec mysqladmin. Les deux commandes suivantes illustrent la procédure :

```
# mysqladmin -u root -p create dbname [Entrée]
Password : ********** [Entrée]
```

```
# mysql -u root -p dbname < backupfile.sql [Entrée]
Password : ********** [Entrée]
```

Les sauvegardes de base de données créée avec une version récente de mysqldump comporte des commandes SQL pour activer le jeu de caractères utilisé dans le fichier de sauvegarde. Ne vous préoccupez donc pas du jeu de caractères. Cette information manque dans les sauvegardes créées avec d'anciennes versions de mysqldump ou d'autres programmes comme phpMyadmin.

Notez que les sauvegardes réalisées avec les versions récentes de phpMyadmin utilisent par défaut le jeu de caractères UTF-8.

Pour qu'une ancienne sauvegarde puisse être correctement lue, vous devez utiliser l'option –default-character-set= .... Si vous ne connaissez pas le jeu de caractères utilisés, saisissez latin1. Ce jeu de caractères était employé par défaut par les versions de MySQL 4.0 et antérieures.

Par exemple:

```
mysql –u –root –p –default_character_set=latin1 dbname < backup.sql
```

LAB #5 - Sauvegardes

Sauvegardez l'ensemble de vos bases de données. Choisissez une interface graphique (Webmin, mysql-gui-tools ou MySQL Workbench) et recréez toutes les bases, y compris les tables, données, fonctions, procédures, vues et déclencheurs.

Réplication

Présentation

Le réplication a lieu en générale entre une machine maître et une machine esclave. Dans notre cas, nous allons créer ultérieurement un clone de la VM utilisée jusqu'à maintenant pour servir de machine esclave.

Quand la réplication est mise en place, deux threads sont démarrés sur l'esclave :

- IO_THREAD - copie les informations des journaux binaires du maître dans son relay log,

- SQL-THREAD - exécute les modifications sur la ou les bases de données de l'esclave.

Un troisième thread est ouvert sur le serveur maître par l'IO_THREAD pour le transfert des journaux.

Un esclave ne peut avoir qu'un seul maître mais un maître peut avoir plusieurs esclaves.

Dans une réplication Maître - esclave, ceci permet d'être évolutif en lecture mais pas en écriture. Trop de requêtes en écriture vont “inondées” le ou les esclaves.

Dans une réplication Maître-Maître il est possible d'avoir deux modes de fonctionnement :

- actif-passif,
  - un des maîtres est utilisé pour les écritures/lectures tandis que l'autre est un backup en cas de défaillance du premier,
- actif-actif,
  - les deux maîtres sont utilisés pour les écritures/lectures.

Il existe deux types de réplications :

- SBR - Statement Based Replication,
  - les requêtes SQL sont ré-exécutées sur l'esclave. Cette méthode crée des problèmes quand la requête est dynamique (par exemple dans le cas d'une fonction),
- RBR - Row Based Replication,
  - les modifications et les valeurs des enregistrements sont répliqués. Cette méthode est plus lente à cause de la quantité de données à répliquer.

A partir de la version 5.6 de MySQL, il est possible de choisir un mode **mixed**. Le serveur va choisir la méthode approprié pour chaque cas.

Il est possible de fixer la valeur de plusieurs variable pour choisir quelles bases de données vont être répliquées :

- binlog-do-db et binlog-ignore-db sur le maître,
- replicate-do-db et replicate-ignore-db sur l'esclave.

Dans le premier cas cela nécessite le re-démarrage du serveur maître. Il est donc préférable d'utiliser les variables sur l'esclave.

LAB #6 - Mise en Place de la Réplication Maître/Esclave

Création du compte de réplication

Saisissez la commande suivante pour créer l'utilisateur **replicant** et y associer un mot de passe et les permissions adéquates :

```
GRANT REPLICATION SLAVE, REPLICATION CLIENT
    ON *.*
    TO 'replicant'@'%'
    IDENTIFIED BY 'password';
```

Vous obtiendrez un résultat similaire à celui-ci :

```
mysql> GRANT REPLICATION SLAVE, REPLICATION CLIENT
    ->     ON *.*
    ->     TO 'replicant'@'%'
    ->     IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)
```

> Dans le cas de deux machines physiques, cette commande doit être saisie sur le maître **et** l'esclave. Le compte doit avoir les droits de REPLICATION SLAVE (avant MySQL 4.0.2, il devait avoir les droits de FILE). Le mot de passe **password** est à modifier mais doit être identique sur le serveur maître et le serveur esclave.

Vérifiez ensuite la création de votre utilisateur **replicant** :

```
mysql> USE mysql;
Database changed
mysql> SELECT user, password FROM user;
+-----------+-------------------------------------------+
| user      | password                                  |
```

```
+-----------+-------------------------------------------+
| root      | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
| root      |                                           |
| root      |                                           |
|           |                                           |
|           |                                           |
| user1     | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| fenestros | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| replicant | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+-------------------------------------------+
8 rows in set (0.00 sec)

mysql>
```

Création d'un clone de la VM

Eteignez votre VM et créez deux clônes, **maitre** et **esclave** dans VirtualBox et attachez les clônes créés.

Configurer le maître

Configurez l'interface réseau **eth0** en mode **Réseau Interne** (*intnet*) via l'interface VirtualBox de votre système d'exploitation hôte. Configurez une adresse IP fixe pour eth0.

Lancez la VM **maitre**.

Ajoutez ensuite deux lignes dans la section **mysqld** du fichier **/etc/my.cnf** du maître:

my.cnf

```
[mysqld]
datadir=/var/lib/mysql
```

```
socket=/var/lib/mysql/mysql.sock
user=mysql
#AJOUTER LA LIGNE SUIVANTE
server-id = 1
#AJOUTER LA LIGNE SUIVANTE
log-bin = mysql-bin
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Redémarrez le service **mysqld** :

```
[root@centos6 ~]# service mysqld restart
Arrêt de mysqld :                                          [  OK  ]
Démarrage de mysqld :                                      [  OK  ]
```

Afin de permettre une sauvegarde complète des bases de données sur le maître, vous devez vous assurer que seulement root y a accès. Ceci est possible grâce à la directive **max_connections**.

Contrôlez d'abord la valeur actuelle de **max_connections**:

```
[root@centos6 ~]# mysql -u root -p mysql
Enter password: fenestros1
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.61-log Source distribution
```

```
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW VARIABLES LIKE 'max_connections';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| max_connections | 151   |
+-----------------+-------+
1 row in set (0.01 sec)

mysql>
```

Notez la valeur et ensuite définissez-la à 0 pour n'avoir que la connexion de root :

```
mysql> SET GLOBAL max_connections = 1;
```

Sauvegardez ensuite la totalité des bases de données sur le maître:

```
[root@centos6 ~]# mysqldump --user=root --password=fenestros1 --extended-insert --all-databases --master-data --
event  > /tmp/backup.sql
```

Notez que **–master-data** permet à mysqldump de récupérer les données du maître concernant la réplication.

L'examen de votre fichier backup.sql révélera la présence d'une section spécifique à la fonction de réplication - le master-data :

```
...
--
-- Position to start replication or point-in-time recovery from
--

CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000002', MASTER_LOG_POS=106;
...
```

Re-définissez à 151 connexions la valeur de **max-connexions** et ensuite contrôlez la valeur de **max_connections**:

```
[root@centos6 ~]# mysql -u root -p mysqlEnter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.1.61-log Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SET GLOBAL max_connections = 151;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'max_connections';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| max_connections | 151   |
```

```
+-----------------+-------+
1 row in set (0.00 sec)

mysql>
```

Configurer l'esclave

Configurez l'interface réseau en mode **Réseau Interne** (*intnet*) via l'interface VirtualBox de votre système d'exploitation hôte. Configurez une adresse IP fixe pour eth0.

Lancez la VM **esclave**.

Ajoutez ensuite les lignes dans la section **mysqld** du fichier **/etc/my.cnf** de l'esclave:

```
server-id = 2
log-bin = /var/log/mysqld/bin.log
log-bin-index = /var/log/mysqld/log-bin.index
log-error = /var/log/mysqld/error.log
relay-log = /var/log/mysqld/relay.log
relay-log-info-file = /var/log/mysqld/relay-log.info
relay-log-index = /var/log/mysqld/relay-log.index
```

Vous obtiendrez :

my.cnf

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
server-id = 2
log-bin = /var/log/mysqld/bin.log
log-bin-index = /var/log/mysqld/log-bin.index
```

```
log-error = /var/log/mysqld/error.log
relay-log = /var/log/mysqld/relay.log
relay-log-info-file = /var/log/mysqld/relay-log.info
relay-log-index = /var/log/mysqld/relay-log.index
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Dans le cas de deux machines physiques, sur l'esclave, injectez le fichier backup.sql en provenance du maître:

```
# mysql --user=root --password=fenestros < /tmp/backup.sql [Entrée]
```

Créez maintenant le répertoire **/var/log/mysqld** et attribuez-le à l'utilisateur **mysql** du groupe **mysql** :

```
[root@centos6 ~]# mkdir /var/log/mysqld
[root@centos6 ~]# chown mysql:mysql /var/log/mysqld
```

Définir les paramètres de l'esclave :

```
[root@centos6 ~]# mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.61 Source distribution
```

```
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CHANGE MASTER TO MASTER_HOST ='192.168.22.10';
Query OK, 0 rows affected (0.03 sec)

mysql> CHANGE MASTER TO MASTER_PORT =3306;
Query OK, 0 rows affected (0.01 sec)

mysql> CHANGE MASTER TO MASTER_USER ='replicant';
Query OK, 0 rows affected (0.01 sec)

mysql> CHANGE MASTER TO MASTER_PASSWORD ='password';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

N'oubliez pas de modifier l'instruction **CHANGE MASTER TO MASTER_HOST ='192.168.22.10';** en fonction de l'adresse IP de votre maître. Pour plus d'informations concernant CHANGE MASTER TO, consultez **ce lien**. Les valeurs de ces variables sont normalement stockées dans **/var/lib/mysql/master.info**.

Redémarrez le service mysqld :

```
mysql> exit
Bye
[root@centos6 ~]# service mysqld restart
```

```
Arrêt de mysqld :                                          [  OK  ]
Démarrage de mysqld :                                      [  OK  ]
```

Ensuite démarrez l'esclave:

```
[root@centos6 ~]# mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.61-log Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
```

Contrôler la réplication

Arrêtez le pare-feu **iptables** sur le maître et l'esclave :

```
[root@centos6 ~]# service iptables stop
```

```
iptables : Suppression des règles du pare-feu :            [  OK  ]
iptables : Configuration des chaînes sur la politique ACCEP[  OK  ]er
iptables : Déchargement des modules :                      [  OK  ]
```

Pour visualiser le statut de la réplication sur l'esclave, utilisez la commande suivante:

```
mysql> SHOW SLAVE STATUS;
+----------------------------------+--------------+-------------+-------------+---------------+-----------------
-+--------------------+----------------+---------------+----------------------+------------------+-------------
------+-----------------+--------------------+--------------------+----------------------+--------------------
-----+------------------------------+------------+------------+--------------+---------------------+-------------
---+-----------------+----------------+---------------+--------------------+--------------------+----------------
----+-----------------+--------------------+----------------+----------------------+----------------------------
--+---------------+---------------+----------------+----------------+
| Slave_IO_State                   | Master_Host   | Master_User | Master_Port | Connect_Retry | Master_Log_File
| Read_Master_Log_Pos | Relay_Log_File | Relay_Log_Pos | Relay_Master_Log_File | Slave_IO_Running |
Slave_SQL_Running | Replicate_Do_DB | Replicate_Ignore_DB | Replicate_Do_Table | Replicate_Ignore_Table |
Replicate_Wild_Do_Table | Replicate_Wild_Ignore_Table | Last_Errno | Last_Error | Skip_Counter |
Exec_Master_Log_Pos | Relay_Log_Space | Until_Condition | Until_Log_File | Until_Log_Pos | Master_SSL_Allowed |
Master_SSL_CA_File | Master_SSL_CA_Path | Master_SSL_Cert | Master_SSL_Cipher | Master_SSL_Key |
Seconds_Behind_Master | Master_SSL_Verify_Server_Cert | Last_IO_Errno | Last_IO_Error | Last_SQL_Errno |
Last_SQL_Error |
+----------------------------------+--------------+-------------+-------------+---------------+-----------------
-+--------------------+----------------+---------------+----------------------+------------------+-------------
------+-----------------+--------------------+--------------------+----------------------+--------------------
-----+------------------------------+------------+------------+--------------+---------------------+-------------
---+-----------------+----------------+---------------+--------------------+--------------------+----------------
----+-----------------+--------------------+----------------+----------------------+----------------------------
--+---------------+---------------+----------------+----------------+
| Waiting for master to send event | 10.192.168.78 | replicant   |       3306 |           60 | mysql-bin.000003
|                 106 | relay.000006   |          251 | mysql-bin.000003      | Yes              | Yes
```

```
|                 |                      |                    |                          |                            |
|          0 |           |            0 |                 106 |             541 | None            |
|             0 | No                 |                      |                   |                  |
|                |                    0 | No                           |            0 |               |
0 |                |
+---------------------------------+---------------+-------------+-------------+---------------+-----------------
-+---------------------+----------------+---------------+-----------------------+------------------+-------------
------+------------------+---------------------+--------------------+-------------------------+--------------------
-----+-------------------------------+------------+------------+--------------+---------------------+-------------
---+-----------------+----------------+---------------+--------------------+--------------------+---------------
----+-----------------+--------------------+----------------+-----------------------+----------------------------
--+---------------+---------------+----------------+----------------+
1 row in set (0.00 sec)

mysql>
```

Pour visualiser le statut de la réplication sur le **maître**, utilisez la commande suivante:

```
mysql> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000003 |      106 |              |                  |
+------------------+----------+--------------+------------------+
1 row in set (0.00 sec)

mysql>
```

Modifiez ensuite un enregistrement dans la base **ligue1** du Maître :

```
mysql> use ligue1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> show tables;
+------------------+
| Tables_in_ligue1 |
+------------------+
| V_EQUIPE         |
| equipe           |
| rencontre        |
+------------------+
3 rows in set (0.00 sec)

mysql> select * from equipe;
+-----------+------------------------+------------------+------------+--------+------+------------------+--------
--------+
| id_equipe | nom                    | stade            | ville      | points | buts | entraineur       |
tel_entraineur |
+-----------+------------------------+------------------+------------+--------+------+------------------+--------
--------+
|         1 | FC Mandriva            | Parc des Princes | Paris      |      3 |    3 | Ricardo GOMES    |
06-56-56-56-56 |
|         2 | Debian AC              | Yankee Stadium   | New York   |      3 |    3 | Gérard Houllier |
06-57-57-57-57 |
|         3 | Vista FC               | Qwest Field      | Redmond    |      0 |    0 | inconnu          | inconnu
|
|         4 | Racing Club Strasbourg | La Meinau        | STRASBOURG |      0 |    0 | inconnu          | inconnu
|
+-----------+------------------------+------------------+------------+--------+------+------------------+--------
--------+
4 rows in set (0.04 sec)

mysql> SELECT * FROM equipe WHERE id_equipe=1;
+-----------+-------------+------------------+-------+--------+------+---------------+----------------+
| id_equipe | nom         | stade            | ville | points | buts | entraineur    | tel_entraineur |
+-----------+-------------+------------------+-------+--------+------+---------------+----------------+
```

```
|         1 | FC Mandriva | Parc des Princes | Paris |      3 |    3 | Ricardo GOMES | 06-56-56-56-56 |
+-----------+-------------+------------------+-------+--------+------+---------------+----------------+
1 row in set (0.00 sec)

mysql> UPDATE equipe SET stade = 'La réplication fonctionne!!' WHERE id_equipe=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM equipe WHERE id_equipe=1;
+-----------+-------------+------------------------------+-------+--------+------+---------------+--------------
-+
| id_equipe | nom         | stade                        | ville | points | buts | entraineur    | tel_entraineur
|
+-----------+-------------+------------------------------+-------+--------+------+---------------+--------------
-+
|         1 | FC Mandriva | La réplication fonctionne!! | Paris |      3 |    3 | Ricardo GOMES | 06-56-56-56-56
|
+-----------+-------------+------------------------------+-------+--------+------+---------------+--------------
-+
1 row in set (0.00 sec)

mysql>
```

Connectez-vous maintenant à mysql sur la VM **esclave** et contrôlez la réplication :

```
mysql> USE ligue1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM equipe WHERE id_equipe=1;
+-----------+-------------+------------------------------+-------+--------+------+---------------+--------------
-+
| id_equipe | nom         | stade                        | ville | points | buts | entraineur    | tel_entraineur
```

```
|
+----------+-------------+------------------------------+-------+--------+------+--------------+--------------
-+
|         1 | FC Mandriva | La réplication fonctionne!! | Paris |      3 |    3 | Ricardo GOMES | 06-56-56-56-56
|
+----------+-------------+------------------------------+-------+--------+------+--------------+--------------
-+
1 row in set (0.00 sec)

mysql>
```

Journalisation et Sécurité

Fichiers Logs

Les fichiers journaux du serveur MySQL se trouvent dans le répertoire **/var/lib/mysql/** sauf dans le cas où la configuration dans **/etc/my.cnf** stipule autrement. MySQL utilise quatre journaux différents :

- Le journal des erreurs (*Error log*),
- Le journal binaire (*Binary Log*),
- Le journal des requêtes lentes (*Slow Query Log*),
- Le journal général (*General Query log*).

Le Journal des Erreurs

Sous Red Hat et CentOS, par défaut, seul le journal des erreurs est activé par la directive suivante dans le fichier **/etc/my.cnf** :

```
...
  log-error=/var/log/mysqld.log
...
```

Sous Windows™, par défaut, seul le journal des erreurs est activé par la directive suivante dans le fichier **C:\ProgramData\MySQL\MySQL Server 5.6\my.ini** :

```
...
# Error Logging.
log-error="SERVER.err"
```

Le Journal Binaire

Le journal binaire, aussi appelé **binlog**, est chargé de stocker sous format binaire toutes les requêtes qui modifient les objets de la base de données. Il est activé par l'addition de l'option **log-bin** dans le fichier **/etc/my.cnf**. Son fichier d'index est spécifié par l'option **log-bin-index**. Ce fichier contient la liste de tous les fichiers binlogs depuis la dernière purge et permet d'identifier le binlog en cours d'utilisation. Le binlog est l'élément central à la réplication.

Le premier binlog créé est nommé **nom-fichier.000001**. Le passage au binlog suivant, dénommé **nom-fichier.000002** a lieu dans trois cas spécifiques :

- le serveur est redémarré,
- la taille maximale définie par l'option **max_binlog_size** est atteinte,
- la commande **FLUSH LOGS** est exécutée.

Pour purger les anciens binlogs il convient de :

- soit fixer la valeur de l'option **expire_logs_days** dans le fichier **/etc/my.cnf**,
- soit utiliser la commande **PURGE BINARY LOGS BEFORE** suivi par une date,
- soit utiliser la commande **PURGE BINARY LOGS TO** suivi par un numéro de journal,
- soit supprimer tous les fichiers et repartir d'un fichier .000001 avec la commande **RESET MASTER**.

Le Journal des Requêtes Lentes

Le journal des requêtes lentes permet d'identifier les requêtes lentes. Une fois activé graĉe à l'option **slow_query_log**, toutes les requêtes dépassant la valeur en secondes de l'option **long_query_time** seront consignées. Les informations peuvent être stockées soit dans un fichier spécifié par l'option

slow_query_log_file soit dans une table dédiée nommée **slow_log** dans le schéma **mysql**. Le choix est fait en éditant l'option **log_output** :

- FILE : les informations sont stockées dans un fichier,
- TABLE : les informations sont stockées dans la table,
- FILE,TABLE : les informations sont stockées dans un fichier **et** dans la table,
- NONE : pas de journalisation.

Par exemple sous Red Hat et CentOS :

```
[mysqld]
slow_query_log
slow_query_log_file = /var/log/mysql-slow.log
long-query-time = 2
...
```

et sous Windows™ :

```
...
slow-query-log=1
slow_query_log_file="SERVER-slow.log"
...
```

L'interrogation de MySQL démontre la prise en compte des directives :

```
mysql> SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
+---------------------+-------------------------+
| Variable_name       | Value                   |
+---------------------+-------------------------+
| slow_query_log      | ON                      |
| slow_query_log_file | /var/log/mysql-slow.log |
+---------------------+-------------------------+
2 rows in set (0.00 sec)
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
```

```
+---------------------+-----------------+
| Variable_name       | Value           |
+---------------------+-----------------+
| slow_query_log      | ON              |
| slow_query_log_file | SERVER-slow.log |
+---------------------+-----------------+
2 rows in set (0.02 sec)

mysql>
```

Il est aussi possible de désactiver et d'activer le journal à chaud :

```
mysql> SET GLOBAL slow_query_log = 'OFF';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
+---------------------+-------------------------+
| Variable_name       | Value                   |
+---------------------+-------------------------+
| slow_query_log      | OFF                     |
| slow_query_log_file | /var/log/mysql-slow.log |
+---------------------+-------------------------+
2 rows in set (0.01 sec)
```

```
mysql> SET GLOBAL slow_query_log = 'OFF';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
+---------------------+-----------------+
| Variable_name       | Value           |
+---------------------+-----------------+
| slow_query_log      | OFF             |
| slow_query_log_file | SERVER-slow.log |
+---------------------+-----------------+
```

```
2 rows in set (0.01 sec)
mysql>
```

La valeur par défaut de **long-query-time** est **10** secondes !!. La valeur recommandée est 1 ou 2 secondes.

Le Journal Général

Le journal général permet de consigner les requêtes valides et les informations de connexion/déconnexion des clients. Il n'est pas conseillé d'activer ce journal en production car il crée une surcharge importante. De la même manière que le journal des requêtes lentes, le journal général peut être écrit dans un fichier ou dans une table dénommée **mysql.general_log**. Encore une fois c'est la valeur de la directive **log_output** qui détermine la destination finale :

- FILE : les informations sont stockées dans un fichier,
- TABLE : les informations sont stockées dans la table,
- FILE,TABLE : les informations sont stockées dans un fichier **et** dans la table,
- NONE : pas de journalisation.

Red Hat et CentOS

Activez le journal générale :

```
mysql> SET GLOBAL general_log = 'ON';
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'general_log%';
+------------------+----------------------------+
| Variable_name    | Value                      |
+------------------+----------------------------+
| general_log      | ON                         |
```

```
| general_log_file | /var/lib/mysql/centos.log |
+------------------+---------------------------+
2 rows in set (0.01 sec)

mysql> exit
Bye

[root@centos6 log]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.01 sec)

mysql> exit
Bye
```

```
[root@centos6 log]# cat /var/lib/mysql/centos.log
/usr/libexec/mysqld, Version: 5.5.33-cll-lve (MySQL Community Server (GPL) by Atomicorp). started with:
Tcp port: 3306  Unix socket: /var/lib/mysql/mysql.sock
Time                 Id Command    Argument
140123 15:55:06     3 Query    SHOW GLOBAL VARIABLES LIKE 'general_log%'
140123 15:56:21     3 Quit
140123 15:56:25     4 Connect    root@localhost on
            4 Query    select @@version_comment limit 1
140123 15:56:40     4 Query    SHOW DATABASES
140123 15:56:42     4 Quit
```

En ajoutant la directive **log_output = TABLE** au fichier **/etc/my.cnf** et en **redémarrant le serveur**, on note que les traces sont maintenant dirigées vers la table **mysql.general_log** au lieu du fichier **/var/lib/mysql/centos.log** :

```
mysql> exit
Bye

[root@centos6 log]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW GLOBAL VARIABLES LIKE 'general_log%';
+------------------+---------------------------+
| Variable_name    | Value                     |
+------------------+---------------------------+
```

```
| general_log      | OFF                         |
| general_log_file | /var/lib/mysql/centos.log |
+------------------+-----------------------------+
2 rows in set (0.00 sec)

mysql> SET GLOBAL general_log = 'ON';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'general_log%';
+------------------+-----------------------------+
| Variable_name    | Value                       |
+------------------+-----------------------------+
| general_log      | ON                          |
| general_log_file | /var/lib/mysql/centos.log |
+------------------+-----------------------------+
2 rows in set (0.02 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM mysql.general_log;
+----------------------+----------------------------+-----------+-----------+--------------+----------------------
--------------------+
| event_time           | user_host                  | thread_id | server_id | command_type | argument
|
+----------------------+----------------------------+-----------+-----------+--------------+----------------------
```

```
--------------------+
| 2014-01-23 16:03:11 | root[root] @ localhost [] |         4 |         0 | Query        | SHOW GLOBAL VARIABLES
LIKE 'general_log%' |
| 2014-01-23 16:03:19 | root[root] @ localhost [] |         4 |         0 | Query        | SHOW DATABASES
|
| 2014-01-23 16:03:26 | root[root] @ localhost [] |         4 |         0 | Query        | SELECT * FROM
mysql.general_log           |
+---------------------+---------------------------+-----------+-----------+--------------+----------------------
--------------------+
3 rows in set (0.00 sec)

mysql> exit
Bye

[root@centos6 log]# cat /var/lib/mysql/centos.log
/usr/libexec/mysqld, Version: 5.5.33-cll-lve (MySQL Community Server (GPL) by Atomicorp). started with:
Tcp port: 3306  Unix socket: /var/lib/mysql/mysql.sock
Time                 Id Command    Argument
140123 15:55:06     3 Query    SHOW GLOBAL VARIABLES LIKE 'general_log%'
140123 15:56:21     3 Quit
140123 15:56:25     4 Connect    root@localhost on
            4 Query    select @@version_comment limit 1
140123 15:56:40     4 Query    SHOW DATABASES
140123 15:56:42     4 Quit
/usr/libexec/mysqld, Version: 5.5.33-cll-lve (MySQL Community Server (GPL) by Atomicorp). started with:
Tcp port: 3306  Unix socket: /var/lib/mysql/mysql.sock
Time                 Id Command    Argument
```

Notez que la valeur de la directive **general_log** était **OFF** après le redémarrage du serveur.

Windows

Activez le journal générale :

```
mysql> SET GLOBAL general_log = 'ON';
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'general_log%';
+------------------+------------+
| Variable_name    | Value      |
+------------------+------------+
| general_log      | ON         |
| general_log_file | SERVER.log |
+------------------+------------+
2 rows in set (0.01 sec)

mysql>
exit
```

Connectez-vous de nouveau à MySQL :

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| ligue1             |
| mysql              |
| nombres            |
| performance_schema |
| test               |
+--------------------+
6 rows in set (0.01 sec)
```

```
mysql>exit
```

Dans une fenêtre de CMD, lancez la commande suivante :

```
C:\ProgramData\MySQL\MySQL Server 5.6\data>more server.log
C:\Program Files\MySQL\MySQL Server 5.6\bin\mysqld.exe, Version: 5.6.28-log (MySQL Community Server (GPL)).
started with:
TCP Port: 3306, Named Pipe: MySQL
Time                 Id Command    Argument
160224 16:40:57     6 Query     SHOW GLOBAL VARIABLES LIKE 'general_log%'
160224 16:46:18     6 Quit
160224 16:47:07     7 Connect   root@localhost on
                    7 Query     select @@version_comment limit 1
160224 16:47:19     7 Query     SHOW DATABASES
160224 16:47:40     7 Quit

C:ProgramDataMySQLMySQL Server 5.6data>
```

En modifiant la directive **log_output=FILE** à **log_output = TABLE** dans le fichier **C:\ProgramData\MySQL\MySQL Server 5.6\my.ini** et en **redémarrant le serveur**, on note que les traces sont maintenant dirigées vers la table **mysql.general_log** au lieu du fichier **C:\ProgramData\MySQL\MySQL Server 5.6\data\server.log** :

```
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.28-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'general_log%';
+------------------+------------+
| Variable_name    | Value      |
+------------------+------------+
| general_log      | OFF        |
| general_log_file | SERVER.log |
+------------------+------------+
2 rows in set (0.01 sec)

mysql> SET GLOBAL general_log = 'ON';
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| ligue1             |
| mysql              |
| nombres            |
| performance_schema |
| test               |
+--------------------+
6 rows in set (0.03 sec)

mysql> SELECT * FROM mysql.general_log;
+----------------------+------------------------------+-----------+-----------+--------------+-------------------
-------------+
| event_time           | user_host                    | thread_id | server_id | command_type | argument
|
+----------------------+------------------------------+-----------+-----------+--------------+-------------------
-------------+
| 2016-02-25 09:21:04 | root[root] @ localhost [::1] |         1 |         1 | Query        | SHOW DATABASES
|
```

```
| 2016-02-25 09:21:14 | root[root] @ localhost [::1] |         1 |         1 | Query        | SELECT * FROM
mysql.general_log |
+---------------------+------------------------------+-----------+-----------+--------------+------------------
-------------+
2 rows in set (0.00 sec)

mysql>
```

Notez que la valeur de la directive **general_log** était **OFF** après le redémarrage du serveur.

Sécurité

Le système de sécurité sous MySQL se repose sur des **privilèges** qui utilisent trois données:

- Le nom de l'utilisateur
- Le mot de passe de l'utilisateur
- Le nom de l'ordinateur ou l'adresse IP d'où se connecte l'utilisateur

Il est important de noter qu'il n'y a pas de corréspondance entre les noms d'utilisateurs et les mots de passe sous MySQL et ceux du système d'exploitation.

Les privilèges sont stockées dans cinq tables de la base **mysql**:

- **user**,
  - La table **user** stocke les privilèges globaux des utilisateurs,
- **db**,
  - La table **db** stocke quels utilisateurs peuvent se connecter à partir de quels hôtes sur quelles bases de données,
- **host**,
  - Cette table est un complément de la table précédente. Dans le cas où le champ **host** est laissé en blanc dans la table db, MySQL cherchera ces informations dans la table **host**,

- **tables_priv**,
  - Cette table stocke des privilèges spécifiques aux tables,
- **columns_priv**,
  - Cette table stocke des privilèges spécifiques aux colonnes.

Privilèges d'Administration

Droit	Description
CREATE TEMPORARY TABLES	Créer des tables temporaires
CREATE USER	Créer, modifier, supprimer des utilisateurs avec les commandes CREATE, DROP et RENAME
FILE	Lire et écrire dans des fichiers sur le serveur avec les commandes SELECT ... INTO OUTFILE, LOAD DATA
GRANT OPTION	Transmettre ses privilèges à un autre utilisateur
LOCK TABLES	Verrouiller des tables
PROCESS	Voir les threads
PROXY	Activer le mode proxy
RELOAD	Réinitialiser des tables, journaux, statistiques avec les commandes FLUSH et RESET
REPLICATION CLIENT	Superviser la réplication avec les commandes SHOW MASTER STATUS, SHOW SLAVE STATUS
REPLICATION SLAVE	Récupérer les évenements du journal binaire du Maître
SHOW SCHEMAS/DATABASES	Voir la liste des bases de données
SHUTDOWN	Arrêter le serveur avec la commande mysqladmin shutdown
SUPER	Exécuter les commandes CHANGE MASTER TO, KILL, SET GLOBAL etc

Privilèges au Niveau des Schémas

Droit	Description
ALTER	Modifier le schéma et les tables avec les commandes ALTER SCHEMA/DATABASE/TABLE
CREATE	Créer des schémas et des tables avec les commandes CREATE SCHEMA/DATABASE/TABLE
CREATE TEMPORARY TABLE	Créer des tables temporaires
CREATE VIEW	Créer des vues
DELETE	Effacer des enregistrements
DROP	Supprimer des schémas et des tables avec les commandes DROP SCHEMA/DATABASE/TABLE
EVENT	Créer des évènements dans l'ordonnanceur

Droit	Description
GRANT OPTION	Transmettre ses privilèges à un autre utilisateur
INDEX	Créer et supprimer des index avec les commandes CREATE/DROP INDEX
INSERT	Insérer des enregistrements dans une table
LOCK TABLES	Verrouiller des tables
SELECT	Afficher des enregistrements d'une table ainsi que la structure de la table avec les commandes SELECT, DESCRIBE, SHOW CREATE TABLE
SHOW VIEW	Voir le code SQL d'une vue avec la commande SHOW CREATE VIEW
UPDATE	Modifier des enregistrements d'une table

Les privilèges liés aux schémas ne prennent effet que lors de la prochaine connexion à ce premier.

Privilèges au Niveau des Tables

Droit	Description
ALTER	Modifier les tables
CREATE	Créer des tables
DELETE	Effacer des enregistrements
DROP	Supprimer des tables
GRANT OPTION	Transmettre ses privilèges à un autre utilisateur
INDEX	Créer et supprimer des index
INSERT	Insérer des enregistrements dans une table
SELECT	Afficher des enregistrements d'une table ainsi que la structure de la table avec les commandes SELECT, DESCRIBE, SHOW CREATE TABLE
TRIGGER	Créer/supprimer des déclencheurs
UPDATE	Modifier des enregistrements d'une table

Les privilèges liés aux tables prennent effet immédiatement.

Privilèges au Niveau des Colonnes

Droit	Description
INSERT	Insérer des enregistrements dans une table
SELECT	Afficher des enregistrements d'une table ainsi que la structure de la table avec les commandes SELECT, DESCRIBE, SHOW CREATE TABLE
UPDATE	Modifier des enregistrements d'une table

Les privilèges liés aux colonnes prennent effet immédiatement.

Privilèges pour les Routines Stockées

Droit	Description
CREATE ROUTINE	Créer des procédures et fonctions stockées
ALTER ROUTINE	Modifier des procédures et fonctions stockées
EXECUTE	Exécuter des procédures et fonctions stockées
GRANT OPTION	Transmettre ses privilèges à un autre utilisateur

Lors de la création d'une routine stockées ou une vue, le créateur peut choisir si la routine ou la vue sera exécuter par la suite avec :

- INVOKER : les droits de l'utilisateur appelant,
- DEFINIR : les droits du créateur.

Cette possibilité est donner par l'utilisation de la commande **SQL SECURITY**. Par exemple :

```
mysql> CREATE SQL SECURITY INVOKER VIEW V_EQUIPE AS SELECT id_equipe, nom, stade, ville, points, buts, entraineur
FROM equipe;
```

Limitations des Ressources

Droit	Description
MAX_QUERIES_PER_HOUR	Limiter le nombre de requêtes par heure
MAX_UPDATES_PER_HOUR	Limiter le nombre de UPDATE par heure
MAX_CONNECTIONS_PER_HOUR	Limiter le nombre de connexions par heure
MAX_USER_CONNECTIONS	Limiter le nombre de connexions simultanées

L'utilisateur anonyme

Lors de l'installation de MySQL deux utilisateurs sont créés:

- **root**
  - l'administrateur du serveur
- **anonyme**
  - l'utilisateur n'ayant accès qu'à la base **test** et à toutes les bases commençant par **test**

Pour des raisons de sécurité, l'utilisateur anonyme ainsi que les bases test doivent être supprimés.

Procédez donc comme suit pour supprimez l'utilisateur anonyme :

```
mysql> USE mysql;
Database changed
mysql> DELETE FROM user WHERE user ='';
Query OK, 2 rows affected (0.00 sec)

mysql>
```

Vérifiez que l'utilisateur anonyme a bien été supprimé :

```
mysql> SELECT user, host, password FROM user;
+-----------+-----------+-------------------------------------------+
| user      | host      | password                                  |
+-----------+-----------+-------------------------------------------+
| root      | localhost | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
```

```
| root      | centos    |                                           |
| root      | 127.0.0.1 |                                           |
| user1     | localhost | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| fenestros | localhost | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| replicant | %         | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)

mysql>
```

Pour supprimer **l'accès** aux bases de données **test**, saisissez la commande suivante:

```
mysql> DELETE FROM db WHERE db LIKE 'test%';
Query OK, 2 rows affected (0.01 sec)

mysql>
```

Saisissez enfin la commande suivante pour mettre à jour les privilèges:

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

La table user

Visualisez la table user:

```
mysql> DESCRIBE user;
+------------------------+-----------------------------------+------+-----+---------+-------+
| Field                  | Type                              | Null | Key | Default | Extra |
+------------------------+-----------------------------------+------+-----+---------+-------+
```

```
| Host                    | char(60)                                | NO   | PRI |         |       |
| User                    | char(16)                                | NO   | PRI |         |       |
| Password                | char(41)                                | NO   |     |         |       |
| Select_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Insert_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Update_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Delete_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Create_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Drop_priv               | enum('N','Y')                           | NO   |     | N       |       |
| Reload_priv             | enum('N','Y')                           | NO   |     | N       |       |
| Shutdown_priv           | enum('N','Y')                           | NO   |     | N       |       |
| Process_priv            | enum('N','Y')                           | NO   |     | N       |       |
| File_priv               | enum('N','Y')                           | NO   |     | N       |       |
| Grant_priv              | enum('N','Y')                           | NO   |     | N       |       |
| References_priv         | enum('N','Y')                           | NO   |     | N       |       |
| Index_priv              | enum('N','Y')                           | NO   |     | N       |       |
| Alter_priv              | enum('N','Y')                           | NO   |     | N       |       |
| Show_db_priv            | enum('N','Y')                           | NO   |     | N       |       |
| Super_priv              | enum('N','Y')                           | NO   |     | N       |       |
| Create_tmp_table_priv   | enum('N','Y')                           | NO   |     | N       |       |
| Lock_tables_priv        | enum('N','Y')                           | NO   |     | N       |       |
| Execute_priv            | enum('N','Y')                           | NO   |     | N       |       |
| Repl_slave_priv         | enum('N','Y')                           | NO   |     | N       |       |
| Repl_client_priv        | enum('N','Y')                           | NO   |     | N       |       |
| Create_view_priv        | enum('N','Y')                           | NO   |     | N       |       |
| Show_view_priv          | enum('N','Y')                           | NO   |     | N       |       |
| Create_routine_priv     | enum('N','Y')                           | NO   |     | N       |       |
| Alter_routine_priv      | enum('N','Y')                           | NO   |     | N       |       |
| Create_user_priv        | enum('N','Y')                           | NO   |     | N       |       |
| Event_priv              | enum('N','Y')                           | NO   |     | N       |       |
| Trigger_priv            | enum('N','Y')                           | NO   |     | N       |       |
| Create_tablespace_priv  | enum('N','Y')                           | NO   |     | N       |       |
| ssl_type                | enum('','ANY','X509','SPECIFIED')       | NO   |     |         |       |
| ssl_cipher              | blob                                    | NO   |     | NULL    |       |
```

```
| x509_issuer            | blob                                | NO   |     | NULL    |       |
| x509_subject           | blob                                | NO   |     | NULL    |       |
| max_questions          | int(11) unsigned                    | NO   |     | 0       |       |
| max_updates            | int(11) unsigned                    | NO   |     | 0       |       |
| max_connections        | int(11) unsigned                    | NO   |     | 0       |       |
| max_user_connections   | int(11) unsigned                    | NO   |     | 0       |       |
| plugin                 | char(64)                            | YES  |     |         |       |
| authentication_string  | text                                | YES  |     | NULL    |       |
+------------------------+-------------------------------------+------+-----+---------+-------+
42 rows in set (0.01 sec)

mysql>
```

Notez la longueur des champs **User** et **Host** :

Champs	**Longueur**
User	16
Host	60

Mots de Passe

Les mots de passe sous MySQL sont hachés avant d'être stockés. Le type de hachage a été modifié à partir de la version 4.1.1. Cet ancien type de hachage est pourtant toujours disponible avec la fonction **old_password** :

```
mysql> SELECT password('root');
+-------------------------------------------+
| password('root')                          |
+-------------------------------------------+
| *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
+-------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT old_password('root');
```

```
+----------------------+
| old_password('root') |
+----------------------+
| 67457e226a1a15bd     |
+----------------------+
1 row in set (0.00 sec)

mysql>
```

MySQL utilise maintenant des Plug-ins d'authentification. La méthode d'authentification par défaut est implémenté par le plug-in **mysql-native-password** :

```
mysql> SHOW PLUGINS;
+---------------------------+----------+--------------------+---------+---------+
| Name                      | Status   | Type               | Library | License |
+---------------------------+----------+--------------------+---------+---------+
| binlog                    | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| mysql_native_password     | ACTIVE   | AUTHENTICATION     | NULL    | GPL     |
| mysql_old_password        | ACTIVE   | AUTHENTICATION     | NULL    | GPL     |
| MyISAM                    | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| CSV                       | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| MRG_MYISAM                | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| MEMORY                    | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| ARCHIVE                   | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| InnoDB                    | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| INNODB_TRX                | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_LOCKS              | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_LOCK_WAITS         | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_CMP                | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_CMP_RESET          | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_CMPMEM             | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_CMPMEM_RESET       | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_BUFFER_PAGE        | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| INNODB_BUFFER_PAGE_LRU    | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
```

```
| INNODB_BUFFER_POOL_STATS | ACTIVE   | INFORMATION SCHEMA | NULL    | GPL     |
| PERFORMANCE_SCHEMA       | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| FEDERATED                | DISABLED | STORAGE ENGINE     | NULL    | GPL     |
| BLACKHOLE                | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
| partition                | ACTIVE   | STORAGE ENGINE     | NULL    | GPL     |
+--------------------------+----------+--------------------+---------+---------+
23 rows in set (0.02 sec)

mysql>
```

Le plug-in **mysql_old_password** implémente l'ancien type de hachage. Les modifications des mots de passe ne prennent effet que lors de la connexion suivante du l'utilisateur. MySQL propose 5 types de hachage : crc32, MD5, SHA1, PASSWORD et OLD_PASSWORD. Les algorithmes PASSWORD et OLD_PASSWORD de MySQL sont internes à MySQL.

LAB #7 - Perte du Mot de Passe de l'Administrateur

Dans le cas de la perte du mot de passe de l'administrateur de MySQL, une solution consiste en une connexion qui détourne la vérification des droits des utilisateurs.

Arrêtez le serveur MySQL puis relancez le serveur MySQL manuellement :

```
[root@centos6 ~]# /usr/libexec/mysqld --skip-grant-tables --skip-networking --console --
socket=/tmp/depannage.sock &
[1] 3223
[root@centos6 ~]# 140126 13:13:41 [Note] libgovernor.so not found
140126 13:13:41 [Warning] Although a path was specified for the --log-slow-queries option, log tables are used.
To enable logging to files use the --log-output=file option.
140126 13:13:41 [Note] Plugin 'FEDERATED' is disabled.
140126 13:13:41 InnoDB: The InnoDB memory heap is disabled
140126 13:13:41 InnoDB: Mutexes and rw_locks use GCC atomic builtins
```

```
140126 13:13:41 InnoDB: Compressed tables use zlib 1.2.3
140126 13:13:41 InnoDB: Using Linux native AIO
140126 13:13:41 InnoDB: Initializing buffer pool, size = 128.0M
140126 13:13:41 InnoDB: Completed initialization of buffer pool
140126 13:13:41 InnoDB: highest supported file format is Barracuda.
140126 13:13:41  InnoDB: Waiting for the background threads to start
140126 13:13:42 InnoDB: 5.5.33 started; log sequence number 1597308
/usr/libexec/mysqld: File '/var/log/mysql-slow.log' not found (Errcode: 13)
140126 13:13:42 [ERROR] Could not use /var/log/mysql-slow.log for logging (error 13). Turning logging off for the
whole duration of the MySQL server process. To turn it on again: fix the cause, shutdown the MySQL server and
restart it.
140126 13:13:42 [Note] /usr/libexec/mysqld: ready for connections.
Version: '5.5.33-cll-lve'  socket: '/tmp/depannage.sock'  port: 0  MySQL Community Server (GPL) by Atomicorp
```

Appuyez sur la touche ↵ Entrée et connectez-vous au serveur MySQL sur le socket utilisé :

```
[root@centos6 ~]# mysql --socket=/tmp/depannage.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Definissez de nouveau le mot de passe de l'utilisateur root :

```
mysql> FLUSH PRIVILEGES;
```

```
Query OK, 0 rows affected (0.01 sec)

mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('fenestros');
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Sortez de MySQL et testez la connexion avec le nouveau mot de passe :

```
 exit
Bye
[root@centos6 ~]# mysql -u root -p --socket=/tmp/depannage.sock
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Expliquez le concept du socket /tmp/depannage.sock.

Tuez maintenant le processus de mysqld et relancez le serveur normalement :

```
[root@centos6 ~]# ps aux | grep mysqld
mysql     3223  1.1  3.0 335580 31632 pts/0    Sl   13:13   0:10 /usr/libexec/mysqld --skip-grant-tables --skip-
networking --console --socket=/tmp/depannage.sock
root      3293  3.0  0.0   4376   800 pts/0    S+   13:28   0:00 grep mysqld
[root@centos6 ~]# kill -9 3223
[root@centos6 ~]# service mysqld start
Démarrage de mysqld :                                      [  OK  ]
[1]+  Processus arrêté      /usr/libexec/mysqld --skip-grant-tables --skip-networking --console --
socket=/tmp/depannage.sock
```

Veuillez noter que toutes les requêtes saisies sont journalisées dans votre fichier historique de MySQL. Par défaut ce fichier est **~/.mysql_history**. Les mots de passe saisis sont en clair :

```
[root@centos6 ~]# cat .mysql_history
SHOW VARIABLES LIKE 'datadir';
SHOW databases;
use mysql;
SHOW tables;
SHOW TABLE STATUS LIKE 'db';
SHOW TABLE STATUS LIKE 'db' \G;
SHOW TABLE STATUS LIKE 'proc' \G;
USE mysql;
SHOW TABLE STATUS LIKE 'help_keyword'\G
use information_shema
show databases;
use information_schema
show tables;
SHOW MASTER STATUS\G
SHOW VARIABLES LIKE 'binlog_checksum';
SHOW GLOBAL VARIABLES LIKE 'slow-query-log%';
SET GLOBAL slow-query-log = 'ON';
SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
SET GLOBAL slow_query_log = 'OFF';
SHOW GLOBAL VARIABLES LIKE 'slow_query_log%';
```

```
SET GLOBAL general_log = 'ON';
SHOW GLOBAL VARIABLES LIKE 'general_log%';
SHOW DATABASES;
USE mysql;
SELECT * FROM general_log;
SHOW DATABASES;
SELECT * FROM general_log;
SELECT * FROM mysql.general_log;
SHOW GLOBAL VARIABLES LIKE 'general_log%';
SET GLOBAL general_log = 'ON';
SHOW GLOBAL VARIABLES LIKE 'general_log%';
SHOW DATABASES;
SELECT * FROM mysql.general_log;
use mysql;
describe user;
FLUSH PRIVILEGES;
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('fenestros');
```

Il convient donc de protéger ce fichier ! Il est pratique courante de créer un lien symbolique vers /dev/null :

ls -s /dev/null $HOME/.mysql_history

La connexion

Lors de la connexion d'un utilisateur, MySQL utilise les trois champs **User**, **Password** et **Host**. MySQL trie la table ainsi obtenue du privilège le plus restrictif au privilège le moins restrictif.

La table suivante:

```
+-----------+-----------+-------------------------------------------+
```

```
| user      | host      | password                                  |
+-----------+-----------+-------------------------------------------+
| root      | localhost | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
| root      | centos    |                                           |
| root      | 127.0.0.1 |                                           |
| user1     | localhost | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| fenestros | localhost | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| replicant | %         | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+-----------+-------------------------------------------+
```

est vue par MySQL ainsi:

```
+-----------+-----------+-------------------------------------------+
| user      | host      | password                                  |
+-----------+-----------+-------------------------------------------+
| root      | 127.0.0.1 |                                           |
| root      | localhost | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
| user1     | localhost | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| fenestros | localhost | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| root      | centos    |                                           |
| replicant | %         | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+-----------+-------------------------------------------+
```

Le symbole **%** indique tous les hôtes.

Prenons le cas d'une connexion de root à partir du localhost. Dans ce cas, MySQL commence par rechercher l'hôte **localhost**. Trois lignes correspondent:

```
...
| root      | localhost | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
| user1     | localhost | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| fenestros | localhost | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
...
```

MySQL recherche ensuite le nom de l'utilisateur, dans notre cas **root**. Une ligne correspond:

```
...
| root      | localhost | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
...
```

MySQL compare ensuite le mot de passe saisi par root avec le mot de passe crypté dans la table. Dans le cas où les deux mots de passe sont équivalents, l'accès est accordé. Dans le cas contraire, l'accès est refusé.

La commande GRANT

La commande **GRANT** permet de:

- créer un utilisateur,
- accorder des privilèges à cet utilisateur.

La commande prend la forme suivante:

```
mysql> GRANT privileges [columns] ON objet TO utilisateur [IDENTIFIED BY 'password'] [WITH GRANT OPTION];
[Entrée]
```

où:

- **privilèges** est une liste de privilèges séparés par des virgules,
  - les privilèges sont **all, all privileges, alter, create, create temporary tables, delete, drop, execute, file, index, insert, lock tables, process, references, reload, select, show databases, shutdown, super, update, usage, create user, create view, show view, create routine, alter routine**,
  - dans le cas où le privilège est **usage**, l'utilisateur est seulement créé, sans privilèges supplémentaires. Autrement dit ce privilège donne le droit de se connecter à MySQL,
- **[columns]** est optionnel. Cette directive permet de spécifier que les privilèges portent sur une ou plusieurs colonnes, séparées par des virgules, de la table spécifiée par **objet**,
- **objet** est une base ou une table
- **WITH GRANT OPTION** donne le droit à l'utilisateur de donner ses propres privilèges à un autre utilisateur.

Saisissez la commande suivante pour créer un nouvel utilisateur:

```
mysql> GRANT usage ON *.* TO user2@localhost IDENTIFIED BY 'motdepasse'; [Entrée]
```

La **portée des droits *.*** implique tous les objets. La portée des droits peut également être de type **schema.***, autrement dit tous les objets d'une base ou de type **schema.table** pour une table précise.

Vous pouvez aussi utiliser * qui implique tous les objets de la base courante dans le cas où la commande **USE** a été utilisée.

Vérifiez que l'utilisateur a bien été créé :

```
mysql> SELECT host, user, password FROM user;
+-----------+-----------+-------------------------------------------+
| host      | user      | password                                  |
+-----------+-----------+-------------------------------------------+
| localhost | root      | *F3E048E28ED63F281CF8A403F96F5D283C8700E6 |
| centos    | root      |                                           |
| 127.0.0.1 | root      |                                           |
| localhost | user2     | *1F48A8CB9F3BAAE4504A9A4549B0AA290BD4E27B |
| localhost | user1     | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| localhost | fenestros | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| %         | replicant | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+-----------+-------------------------------------------+
7 rows in set (0.00 sec)

mysql>
```

Connectez-vous maintenant à MySQL avec le compte d'user2 :

```
[root@centos6 ~]# mysql -u user2 -p
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.61-log Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.00 sec)

mysql> USE lique1;
ERROR 1044 (42000): Access denied for user 'user2'@'localhost' to database 'lique1'
mysql>
```

Notez que l'utilisateur **user2** n'a pas accès aux bases de données ! En effet, il ne peut que se connecter à mysql.

La commande REVOKE

La commande REVOKE est utilisée pour retirer les privilèges. La commande REVOKE ne permet **pas** de supprimer l'utilisateur. Pour supprimer un utilisateur, il convient d'utiliser la commande SQL **DELETE**.

La commande REVOKE prend la forme suivante:

```
mysql> REVOKE privileges [columns] ON objet FROM utilisateur; [Entrée]
```

Par exemple pour retirer les privilèges de l'utilisateur **user2**, connectez-vous à mysql en tant que **root** et saisissez la commande suivante:

```
mysql> REVOKE all ON *.* FROM user2@localhost; [Entrée]
```

Notez l'utilisation du mot clef **all** pour enlever tous les privilèges.

Modifier le mot de passe d'un utilisateur

Pour modifier le mot de passe d'un utilisateur, trois commandes SQL existent.

En utilisant **SET PASSWORD** :

```
mysql> SET PASSWORD FOR utilisateur@host=PASSWORD('nouveau_mot_de_passe'); [Entrée]
```

Notez l'utilisation de la directive **PASSWORD()** pour encrypter le mot de passe.

En utilisant **GRANT** :

```
mysql> GRANT usage ON *.* TO utilisateur@host IDENTIFIED BY 'nouveau_mot_de_passe'; [Entrée]
```

En utilisant **UPDATE USER** :

```
mysql> UPDATE user SET PASSWORD=PASSWORD('nouveau_mot_de_passe') WHERE user='utilisateur' AND host='host';
[Entrée]
```

N'oubliez pas qu'après chaque modification, vous devez utiliser la commande **FLUSH PRIVILEGES**.

Utilisez **SET PASSWORD** et **UPDATE USER** à tour de rôle pour modifier le mot de passe de l'utilisateur **user2** et connectez-vous après chaque modification pour vérifier que cette dernière a été réussie,

Sécuriser l'échange de données

Pour vérifier si votre instance de MySQL peut fonctionner avec **openssl**, il convient de saisir la commande suivante:

```
mysql> SHOW VARIABLES LIKE 'have_openssl';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
+---------------+-------+
1 row in set (0.00 sec)
```

votre instance de MySQL peut accepter des connexions sécurisées.

Si la variable **have_openssl** indique **DISABLED**, cela signifie que le support est disponible mais non activé :

```
mysql> SHOW VARIABLES LIKE 'have_openssl';
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | DISABLED |
+---------------+----------+
1 row in set (0.00 sec)

mysql>
```

Openssl

Lors de l'installation du paquet **openssl**, une clef privée et un certificat d'exemple ont été générés dans le répertoire **/etc/pki** :

```
[root@centos6 ~]# ls -lR /etc/pki
/etc/pki:
total 24
drwxr-xr-x. 6 root root 4096 23 août  06:55 CA
drwxr-xr-x. 2 root root 4096 25 juin  11:30 java
drwxr-xr-x. 2 root root 4096 19 oct.  14:32 nssdb
drwxr-xr-x. 2 root root 4096 19 oct.  14:28 rpm-gpg
drwx------. 2 root root 4096 22 juin  16:27 rsyslog
drwxr-xr-x. 5 root root 4096 19 oct.  14:31 tls

/etc/pki/CA:
total 16
drwxr-xr-x. 2 root root 4096 23 août  06:55 certs
drwxr-xr-x. 2 root root 4096 23 août  06:55 crl
drwxr-xr-x. 2 root root 4096 23 août  06:55 newcerts
drwx------. 2 root root 4096 23 août  06:55 private

/etc/pki/CA/certs:
total 0

/etc/pki/CA/crl:
total 0

/etc/pki/CA/newcerts:
total 0

/etc/pki/CA/private:
total 0
```

```
/etc/pki/java:
total 128
-rw-r--r--. 1 root root 130601  7 avril  2010 cacerts

/etc/pki/nssdb:
total 128
-rw-r--r--. 1 root root 65536 12 janv.  2010 cert8.db
-rw-r--r--. 1 root root  9216 12 janv.  2010 cert9.db
-rw-r--r--. 1 root root 16384 12 janv.  2010 key3.db
-rw-r--r--. 1 root root 11264 12 janv.  2010 key4.db
-rw-r--r--. 1 root root   451 25 juin  09:43 pkcs11.txt
-rw-r--r--. 1 root root   451 29 févr.  2012 pkcs11.txt.rpmnew
-rw-r--r--. 1 root root 16384 12 janv.  2010 secmod.db

/etc/pki/rpm-gpg:
total 16
-rw-r--r--. 1 root root 1706 26 juin  11:29 RPM-GPG-KEY-CentOS-6
-rw-r--r--. 1 root root 1730 26 juin  11:29 RPM-GPG-KEY-CentOS-Debug-6
-rw-r--r--. 1 root root 1730 26 juin  11:29 RPM-GPG-KEY-CentOS-Security-6
-rw-r--r--. 1 root root 1734 26 juin  11:29 RPM-GPG-KEY-CentOS-Testing-6

/etc/pki/rsyslog:
total 0

/etc/pki/tls:
total 24
lrwxrwxrwx. 1 root root    19 25 juin  11:30 cert.pem -> certs/ca-bundle.crt
drwxr-xr-x. 2 root root  4096 19 oct.  14:31 certs
drwxr-xr-x. 2 root root  4096 19 oct.  14:31 misc
-rw-r--r--. 1 root root 10906 15 août  18:25 openssl.cnf
drwxr-xr-x. 2 root root  4096 23 août  06:54 private

/etc/pki/tls/certs:
total 1204
```

```
-rw-r--r--. 1 root root 571450  7 avril  2010 ca-bundle.crt
-rw-r--r--. 1 root root 651083  7 avril  2010 ca-bundle.trust.crt
-rwxr-xr-x. 1 root root    610 23 août  06:55 make-dummy-cert
-rw-r--r--. 1 root root   2242 23 août  06:55 Makefile

/etc/pki/tls/misc:
total 24
-rwxr-xr-x. 1 root root 5178 23 août  06:54 CA
-rwxr-xr-x. 1 root root  119 23 août  06:54 c_hash
-rwxr-xr-x. 1 root root  152 23 août  06:54 c_info
-rwxr-xr-x. 1 root root  112 23 août  06:54 c_issuer
-rwxr-xr-x. 1 root root  110 23 août  06:54 c_name

/etc/pki/tls/private:
total 0
```

Activer SSL

Pour configurer mysql pour SSL, il convient d'abord d'arrêter le service :

```
[root@centos6 ~]# service mysqld stop
Arrêt de mysqld :                                          [  OK  ]
```

Dans cet exemple, vous allez créer vos propres clefs et certificats. Commencez par créer une cléf :

```
[root@centos6 ~]# mkdir /etc/pki/mysql
[root@centos6 ~]# cd /etc/pki/mysql/
[root@centos6 mysql]# openssl genrsa 2048 > ca-key.pem
Generating RSA private key, 2048 bit long modulus
.................................+++
.............+++
e is 65537 (0x10001)
```

Créez ensuite le fichier ca-cert.pem :

```
[root@centos6 mysql]# openssl req -new -x509 -nodes -days 1000 -key ca-key.pem > ca-cert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:FR
State or Province Name (full name) []:VAR
Locality Name (eg, city) [Default City]:Toulon
Organization Name (eg, company) [Default Company Ltd]:Linux E-Learning
Organizational Unit Name (eg, section) []:Training
Common Name (eg, your name or your server's hostname) []:centos.fenestros.loc
Email Address []:root@localhost
```

Créer ensuite le certificat du serveur :

```
[root@centos6 mysql]# openssl req -newkey rsa:2048 -days 1000 -nodes -keyout server-key.pem > server-req.pem
Generating a 2048 bit RSA private key
......+++
.......................................+++
writing new private key to 'server-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:FR
```

```
State or Province Name (full name) []:VAR
Locality Name (eg, city) [Default City]:Toulon
Organization Name (eg, company) [Default Company Ltd]:Linux E-Learning
Organizational Unit Name (eg, section) []:Training
Common Name (eg, your name or your server's hostname) []:server.fenestros.loc
Email Address []:root@localhost

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

[root@centos6 mysql]# openssl x509 -req -in server-req.pem -days 1000 -CA ca-cert.pem -CAkey ca-key.pem -
set_serial 01 > server-cert.pem
Signature ok
subject=/C=FR/ST=VAR/L=Toulon/O=Linux E-Learning/OU=Training/CN=server.fenestros.loc/emailAddress=root@localhost
Getting CA Private Key
```

Créer maintenant le certificat du client:

```
[root@centos6 mysql]# openssl req -newkey rsa:2048 -days 1000 -nodes -keyout client-key.pem > client-req.pem
Generating a 2048 bit RSA private key
..............................+++
...+++
writing new private key to 'client-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:FR
```

```
State or Province Name (full name) []:VAR
Locality Name (eg, city) [Default City]:Toulon
Organization Name (eg, company) [Default Company Ltd]:Linux E-Learning
Organizational Unit Name (eg, section) []:TRaining
Common Name (eg, your name or your server's hostname) []:client.fenestros.loc
Email Address []:root@localhost

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

[root@centos6 mysql]# openssl x509 -req -in client-req.pem -days 1000 -CA ca-cert.pem -CAkey ca-key.pem -
set_serial 01 > client-cert.pem
Signature ok
subject=/C=FR/ST=VAR/L=Toulon/O=Linux E-Learning/OU=TRaining/CN=client.fenestros.loc/emailAddress=root@localhost
Getting CA Private Key
```

Modifiez votre fichier my.cnf :

my.cnf

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
server-id = 1
log-bin = mysql-bin
log=/var/lib/mysql/requete.log
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

ssl-ca=/etc/pki/mysql/ca-cert.pem
ssl-cert=/etc/pki/mysql/server-cert.pem
```

```
ssl-key=/etc/pki/mysql/server-key.pem

[client]
ssl-ca=/etc/pki/mysql/ca-cert.pem
ssl-cert=/etc/pki/mysql/client-cert.pem
ssl-key=/etc/pki/mysql/client-key.pem

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Démarrez votre serveur mysql :

```
[root@centos6 mysql]# service mysqld start
Démarrage de MySQL :                                       [  OK  ]
```

Vérifiez que MySQL fonctionne en mode SSL :

```
mysql> show status like 'Ssl_cipher';
+---------------+--------------------+
| Variable_name | Value              |
+---------------+--------------------+
| Ssl_cipher    | DHE-RSA-AES256-SHA |
+---------------+--------------------+
1 row in set (0.00 sec)

mysql>
```

Supprimez votre utilisateur user2:

```
mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> DELETE FROM user WHERE User='user2';
Query OK, 1 row affected (0.02 sec)

mysql>
```

Contrôlez que l'utilisateur a été supprimé:

```
mysql> SELECT host, user, password FROM user;
+----------------------+-----------+-------------------------------------------+
| host                 | user      | password                                  |
+----------------------+-----------+-------------------------------------------+
| localhost            | root      | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| centos.fenestros.loc | root      |                                           |
| 127.0.0.1            | root      |                                           |
| ::1                  | root      |                                           |
| localhost            | user1     | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| localhost            | fenestros | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| %                    | replicant | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+----------------------+-----------+-------------------------------------------+
8 rows in set (0.00 sec)

mysql>
```

Créez maintenant votre utilisateur **user2** afin que celui-ci se connecte en utilisant SSL:

```
mysql> GRANT usage on *.* TO 'user2'@'localhost' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Contrôlez que l'utilisateur a été ajouté:

```
mysql> SELECT host, user, password FROM user;
```

```
+----------------------+-----------+-------------------------------------------+
| host                 | user      | password                                  |
+----------------------+-----------+-------------------------------------------+
| localhost            | root      | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| centos.fenestros.loc | root      |                                           |
| 127.0.0.1            | root      |                                           |
| ::1                  | root      |                                           |
| %                    | user2     |                                           |
| localhost            | user1     | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| localhost            | fenestros | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| %                    | replicant | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+----------------------+-----------+-------------------------------------------+
8 rows in set (0.00 sec)

mysql>
```

Donnez à user2 un mot de passe:

```
mysql> SET PASSWORD FOR 'user2'@'localhost'=PASSWORD('toto2');
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Contrôlez que le mot de passe a été ajouté:

```
mysql> SELECT host, user, password FROM user;
+----------------------+-----------+-------------------------------------------+
| host                 | user      | password                                  |
+----------------------+-----------+-------------------------------------------+
| localhost            | root      | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| centos.fenestros.loc | root      |                                           |
| 127.0.0.1            | root      |                                           |
| ::1                  | root      |                                           |
| %                    | user2     | *9296FFD029BFAE29EDDC1E57E53F4A8E555895B8 |
```

```
| localhost            | user1     | *34D3B87A652E7F0D1D371C3DBF28E291705468C4 |
| localhost            | fenestros | *00269BA49BEC800F9CCF34C20C1FD83E0236B89A |
| %                    | replicant | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+----------------------+-----------+-------------------------------------------+
8 rows in set (0.00 sec)

mysql>
```

Saisissez enfin la commande suivante pour mettre à jour les privilèges:

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Connectez-vous maintenant en utilisant le compte d'user2.

```
[root@redhat ~]# mysql -u user2 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.45-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Optimisation

Matériel

Processeur

La vitesse d'un processeur est la caractéristique la plus importante. Il n'est pas possible de paralléliser une requête sur plusieurs processeurs. Pour cette raison, une requête occupe un processeur à la fois. Prenons le cas d'un serveur esclave dans le contexte d'une replication. Toutes les requêtes de replication passeront par un seul processeur, rendant les autres processeurs éventuels inutiles.

Mémoire

La mémoire est un facteur critique pour le serveur MySQL. La donnée la plus importante est la quantité de données consultées. Prenons le cas d'une base de données de 20Go où seules les données de la dernière semaine sont consultées. Si ces données ne représentent que 1Go, un serveur muni de 2Go suffira largement.

Disque Dur

Les performances d'un disque dur se mesurent avec deux paramètres, le temps d'accès et le débit. Avec une application transactionnelle de boutique Internet, le temps d'accès est le facteur le plus important. Si par contre, l'application est faite pour stocker l'ensemble des livres dans une bibliothèque, le débit devient très important lors des requêtes de recherche pour trouver un livre spécifique.

Système d'Exploitation

MySQL est disponible pour Windows™, Solaris™ et Linux. Le choix du Système d'Exploitation est une question de coût, de compétences internes et de politique de l'infrastructure informatique.

Cache de Requêtes

Le cache des requêtes est utilisé par MySQL lors des requêtes de type SELECT. Le cache est utile dans le cas où la majorité des requêtes sont de type lecture. Lors d'un SELECT, MySQL vérifie si la requête est déjà dans le cache. Cette vérification est faite pour **toutes** les requêtes, mêmes celles qui sont **impropres**. Il ne faut donc **pas** activer le cache quand :

- les écritures sont nombreuses,
- les lectures ne sont pas répétées,
- le requêtes sont en grande partie **impropre**.

L'activation du cache se fait en modifiant la valeur de l'option **query_cache_type** :

```
mysql> SHOW VARIABLES LIKE 'query_cache_type';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| query_cache_type | ON    |
+------------------+-------+
1 row in set (0.00 sec)

mysql>
```

Cette option peut prendre trois valeurs :

- **ON** : le serveur essaie de mettre en cache toutes les requêtes SELECT sauf celles comportant la phrase SQL_NO_CACHE,
- **OFF** : aucune requête n'est mise en cache,
- **DEMAND** : le serveur essaie de mettre en cache toutes les requêtes SELECT comportant la phrase SQL_CACHE.

Exclusions

Seules les requêtes dites déterministes peuvent être mises en cache. Par exemple, une requête de type SELECT * FROM table donnera toujours le même résultat tant que la table n'est pas modifiée. Cette requête peut être mise dans le cache. Par contre la requête SELECT CURRENT_TIMESTAMP() est non-déterministe. Par conséquent, la requête est dite **impropre**.

Requêtes

MySQL recherche dans le cache chaque fois qu'il y a une requête SELECT. MySQL est sensible à la case. Pour cette raison, MySQL considère que les trois requêtes suivantes sont différentes :

- SELECT prenom,nom FROM familles,
- select prenom,nom FROM familles,
- SELECT nom,prenom FROM familles.

Invalidations

Le serveur MySQL met à jour automatiquement le contenu du cache. Lors d'une modification d'une table au niveau de sa structure ou bien au niveau de ses données, le serveur MySQL cherche l'efficacité et supprime du cache **toutes** les requêtes impliquant la table concernée.

Effacements

Dans le cas où le cache n'est pas suffisant pour stocker une requête, MySQL procède a un effacement de l'entrée la plus ancienne. Ce processus s’appelle un **effacement**.

Fragmentation

MySQL divise le cache en blocs mémoire d'une taille égale. Quand une requête est mise en cache, le serveur MySQL ne connaît pas à l'avance le nombre d'enregistrements qui vont être retournées. De ce fait, il ne connaît pas non plus l'occupation de la mémoire. Si la requête n'occupe pas un nombre entier de blocs, la mémoire concernée est perdue. Dans ce cas on parle de **fragmentation**.

Paramètres

Les options associées avec le cache sont :

Option	**Descripton**
query_cache_size	La taille de la mémoire dédiée au cache. La mémoire est affectée par multiples de 1 024 octets. Dans la pratique il ne faut pas dépasser les 256 Mo.
query_cache_limit	La taille maximale d'une requête qui peut être mise en cache
query_cache_type	Indique le type de cache

Option	Descripton
query_cache_min_res_unit	La taille des blocs de mémoire du cache
query_cache_wlock_invalidate	Quand cette option est **OFF**, le serveur peut renvoyer le contenu du cache même si la table est verrouillée par un autre utilisateur

Verification du Cache

Pour consulter la configuration du cache, utilisez la requête suivante :

```
mysql> SHOW GLOBAL STATUS LIKE 'Qcache%';
+-------------------------+-------+
| Variable_name           | Value |
+-------------------------+-------+
| Qcache_free_blocks      | 0     |
| Qcache_free_memory      | 0     |
| Qcache_hits             | 0     |
| Qcache_inserts          | 0     |
| Qcache_lowmem_prunes    | 0     |
| Qcache_not_cached       | 0     |
| Qcache_queries_in_cache | 0     |
| Qcache_total_blocks     | 0     |
+-------------------------+-------+
8 rows in set (0.01 sec)
```

Cette configuration s'explique de la façon suivante :

Option	Descripton
Qcache_free_memory	La taille de la mémoire cache libre
Qcache_lowmem_prunes	Le nombre d'entrées supprimées dans le cache par défaut d'assez d'espace disque
Qcache_hits	Le nombre de fois que le serveur a pu servir une requête du cache

Les valeurs ci-dessus doivent être comparées avec le nombre total de requêtes de type SELECT :

```
mysql> SHOW GLOBAL STATUS LIKE 'Com_select';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| Com_select    | 14    |
+---------------+-------+
1 row in set (0.00 sec)
```

L'efficacité du cache peut être calculé en utilisant le formule suivant :

```
100 * Qcache_hits / Com_select
```

Le taux d'insertion dans le cache est le résultat du formule suivant :

```
100 * Qcache_inserts / Com_select
```

La fragmentation peut être evaluée par le formule :

```
Qcache_free_blocks / Qcache_total_blocks
```

Pour défragmenter le cache, il convient d'utiliser la commande **FLUSH QUERY CACHE**. Il est à noter que cette commande ne vide pas le contenu du cache. Pour vider le contenu du cache, il convient d'utiliser la commande **RESET QUERY CACHE**.

Optimisation du Schéma

L'optimisation du schéma est un aspect primordial de la bonne gestion de MySQL. Les règles de base sont les suivantes :

- Plus le type de données est petit et compact, plus il sera léger et performant,
- Évitez des colonnes NULL car cela demande plus de travail au serveur que des colonnes NOT NULL,

- Utilisez des champs adéquats aux besoins réels de la donnée à stocker,
- Ne surdimensionnez pas les champs de texte car MySQL utilise la taille maximale pour effectuer des tris. Privilégiez donc des champs VARCHAR par rapport CHAR car ce dernier a une longueur fixe.

PROCEDURE ANALYSE

Afin de vous aider, MySQL fournit une commande capable d'examiner les données réellement présentes dans une table et de vous conseiller sur le type de champ à adopter :

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CarnetAdresses     |
| ligue1             |
| mysql              |
| performance_schema |
| test               |
+--------------------+
6 rows in set (0.01 sec)

mysql> USE ligue1;
Database changed
mysql> SHOW TABLES;
+------------------+
| Tables_in_ligue1 |
+------------------+
| equipe           |
| rencontre        |
+------------------+
2 rows in set (0.00 sec)
```

```
mysql> SHOW CREATE TABLE equipe\G
*************************** 1. row ***************************
       Table: equipe
Create Table: CREATE TABLE `equipe` (
  `id_equipe` int(11) NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) NOT NULL,
  `stade` varchar(50) NOT NULL,
  `ville` varchar(30) NOT NULL,
  `points` int(11) NOT NULL DEFAULT '0',
  `buts` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_equipe`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

mysql> SELECT * FROM equipe PROCEDURE ANALYSE()\G
*************************** 1. row ***************************
             Field_name: ligue1.equipe.id_equipe
              Min_value: 1
              Max_value: 3
             Min_length: 1
             Max_length: 1
       Empties_or_zeros: 0
                  Nulls: 0
Avg_value_or_avg_length: 2.0000
                    Std: 0.8165
      Optimal_fieldtype: ENUM('1','2','3') NOT NULL
*************************** 2. row ***************************
             Field_name: ligue1.equipe.nom
              Min_value: Debian AC
              Max_value: Vista FC
             Min_length: 8
             Max_length: 11
       Empties_or_zeros: 0
                  Nulls: 0
```

```
Avg_value_or_avg_length: 9.3333
                    Std: NULL
      Optimal_fieldtype: ENUM('Debian AC','FC Mandriva','Vista FC') NOT NULL
*************************** 3. row ***************************
             Field_name: ligue1.equipe.stade
              Min_value: Parc des Princes
              Max_value: Yankee Stadium
             Min_length: 11
             Max_length: 16
       Empties_or_zeros: 0
                  Nulls: 0
Avg_value_or_avg_length: 13.6667
                    Std: NULL
      Optimal_fieldtype: ENUM('Parc des Princes','Qwest Field','Yankee Stadium') NOT NULL
*************************** 4. row ***************************
             Field_name: ligue1.equipe.ville
              Min_value: New York
              Max_value: Redmond
             Min_length: 5
             Max_length: 8
       Empties_or_zeros: 0
                  Nulls: 0
Avg_value_or_avg_length: 6.6667
                    Std: NULL
      Optimal_fieldtype: ENUM('New York','Paris','Redmond') NOT NULL
*************************** 5. row ***************************
             Field_name: ligue1.equipe.points
              Min_value: 0
              Max_value: 0
             Min_length: 1
             Max_length: 1
       Empties_or_zeros: 3
                  Nulls: 0
Avg_value_or_avg_length: 0.0000
```

```
                    Std: 0.0000
      Optimal_fieldtype: ENUM('0') NOT NULL
*************************** 6. row ***************************
             Field_name: ligue1.equipe.buts
              Min_value: 0
              Max_value: 0
             Min_length: 1
             Max_length: 1
       Empties_or_zeros: 3
                  Nulls: 0
Avg_value_or_avg_length: 0.0000
                    Std: 0.0000
      Optimal_fieldtype: ENUM('0') NOT NULL
6 rows in set (0.00 sec)

mysql>
```

Normalisation

La normalisation d'une base de données consiste en la structuration des objets de façon à obtenir un modèle de données performant et sur.

Les trois règles les plus importantes sont :

- Eviter des colonnes qui se répètent,
- Eviter l'incohérence des données,
- Tous les champs doivent dépendre uniquement de la clef primaire.

LAB #8 - Normalisation

Créez la base de données **connaissances** :

```
mysql> CREATE DATABASE connaissances;
```

```
Query OK, 1 row affected (0.00 sec)
mysql> USE connaissances;
Database changed
```

Utilisez la requête suivante pour créer la table **employe** :

```
CREATE TABLE employe (
id int(11) NOT NULL,
nom varchar(30) NOT NULL DEFAULT '',
ville varchar(30) NOT NULL DEFAULT '',
savoir1 varchar(30) NOT NULL DEFAULT '',
niv1 tinyint(4) NOT NULL,
savoir2 varchar(30) NOT NULL DEFAULT '',
niv2 tinyint(4) NOT NULL,
PRIMARY KEY (id)
) ENGINE=InnoDB;
```

```
mysql> CREATE TABLE employe (
    -> id int(11) NOT NULL,
    -> nom varchar(30) NOT NULL DEFAULT '',
    -> ville varchar(30) NOT NULL DEFAULT '',
    -> savoir1 varchar(30) NOT NULL DEFAULT '',
    -> niv1 tinyint(4) NOT NULL,
    -> savoir2 varchar(30) NOT NULL DEFAULT '',
    -> niv2 tinyint(4) NOT NULL,
    -> PRIMARY KEY (id)
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.06 sec)

mysql>
```

Utilisez la requête suivante pour injecter des données :

```
INSERT INTO employe (id , nom , ville , savoir1 , niv1 , savoir2 , niv2) VALUES ('1' , 'Alex', 'Londres',
```

```
'Linux', '10','Solaris', '8');INSERT INTO employe (id , nom , ville , savoir1 , niv1 , savoir2 , niv2) VALUES
('2' , 'Mathieu', 'Paris', 'Apache', '5','MySQL', '7');INSERT INTO employe (id, nom , ville , savoir1 , niv1 ,
savoir2 , niv2) VALUES ('3' , 'Thomas', 'Paris', 'Linux', '5','MySQL', '7');
```

```
mysql> INSERT INTO employe (id , nom , ville , savoir1 , niv1 , savoir2 , niv2) VALUES ('1' , 'Alex', 'Londres',
'Linux', '10','Solaris', '8');INSERT INTO employe (id , nom , ville , savoir1 , niv1 , savoir2 , niv2) VALUES
('2' , 'Mathieu', 'Paris', 'Apache', '5','MySQL', '7');INSERT INTO employe (id, nom , ville , savoir1 , niv1 ,
savoir2 , niv2) VALUES ('3' , 'Thomas', 'Paris', 'Linux', '5','MySQL', '7');
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)
```

Vous devez obtenir le résultat suivant :

```
mysql> SELECT * FROM employe;
+----+---------+---------+---------+------+---------+------+
| id | nom     | ville   | savoir1 | niv1 | savoir2 | niv2 |
+----+---------+---------+---------+------+---------+------+
|  1 | Alex    | Londres | Linux   |   10 | Solaris |    8 |
|  2 | Mathieu | Paris   | Apache  |    5 | MySQL   |    7 |
|  3 | Thomas  | Paris   | Linux   |    5 | MySQL   |    7 |
+----+---------+---------+---------+------+---------+------+
3 rows in set (0.01 sec)

mysql>
```

Normalisez la table connaissances.employe.

Indexes

Un index est une structure de données liée à une table dont le rôle est de permettre le serveur de trouver une référence au plus vite.

Un index peut être créé à la création de la table avec la clause KEY ou INDEX ou bien après sa création avec une des commandes suivantes :

```
CREATE INDEX nom_index ON table (colonne);
```

```
ALTER TABLE table ADD INDEX (colonne);
```

Ces deux commandes on des différences :

- Avec CREATE INDEX le nom de l'index est obligatoire,
- Avec ALTER TABLE si le nom de l'index n'est pas spécifié, MySQL en crée un,
- Avec ALTER TABLE il est possible de créer plusieurs index en même temps en séparant les clauses ADD INDEX par des virgules.

Pour supprimer un index on utilise une des deux commandes suivantes :

```
DROP INDEX nom_index
```

```
ALTER TABLE nom_index
```

Types d'Index

Index Uniques

Un index unique comporte une contrainte. Toutes les valeurs non NULL doivent être unique. Par contre il est possible d'avoir plusieurs valeurs NULL :

```
mysql> use indexes;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed

mysql> INSERT INTO t (id) VALUES (1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO t (id) VALUES (1);
ERROR 1062 (23000): Duplicate entry '1' for key 'id'

mysql> INSERT INTO t (id) VALUES (NULL);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO t (id) VALUES (NULL);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM t;
+------+
| id   |
+------+
| NULL |
| NULL |
|    1 |
+------+
3 rows in set (0.00 sec)

mysql>
```

Clef Primaires

Une Clef Primaire ressemble à un index unique mais possède trois différences :

- Il ne peut y avoir qu'une clef unique par table,
- Une clef unique ne peut pas contenir des valeurs NULL,
- Le nom d'une clef unique est toujours PRIMARY.

```
mysql> CREATE TABLE t1 (id INT);
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE INDEX `PRIMARY` ON t1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near '' at line 1

mysql> ALTER TABLE t1 ADD PRIMARY KEY (ID);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Index sur Plusieurs Colonnes

```
mysql> CREATE TABLE t2 (id INT, col1 VARCHAR(30), col2 VARCHAR(30));
Query OK, 0 rows affected (0.33 sec)

mysql> CREATE INDEX index_col12 ON t2 (col1,col2);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

MySQL est capable d'exploiter un préfixe à gauche de l'index multicolonne. Cela implique que si MySQL juge que seul **col1** est nécessaire pour faire le travail, il est capable de l'isoler de l'index multicolonne.

Index sur un Préfixe de Colonne

```
mysql> CREATE TABLE t3 (col TEXT);
```

```
Query OK, 0 rows affected (0.94 sec)

mysql> CREATE INDEX index_prefixe ON t3 (col);
ERROR 1170 (42000): BLOB/TEXT column 'col' used in key specification without a key length

mysql> CREATE INDEX index_prefixe ON t3 (col(50));
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Pour les index stockant du texte, il est possible de ne créer l'index que sur les N premiers caractères du champ.

Clefs Etrangères

Une clef étrangère indique à MySQL de rejeter tout ajout ou modification dans la table enfant si la valeur de la clef n'a pas de correspondance dans la table parent.

```
mysql> CREATE TABLE t4 (
    -> id INT(11) NOT NULL AUTO_INCREMENT,
    -> nom VARCHAR(20) NOT NULL DEFAULT'',
    -> PRIMARY KEY (id),
    -> KEY index_nom (nom)
    -> ) ENGINE = InnoDB;
Query OK, 0 rows affected (0.32 sec)

mysql> CREATE TABLE t5 ( id INT(11) NOT NULL AUTO_INCREMENT, ref_nom VARCHAR(20) NOT NULL DEFAULT'', PRIMARY KEY
(id), CONSTRAINT foreign_index FOREIGN KEY foreign_ref_nom (ref_nom) REFERENCES t4(nom)) ENGINE = InnoDB;
Query OK, 0 rows affected (0.31 sec)
```

```
mysql>
```

> La clause CONSTRAINT est optionnelle. Le nom de la contrainte doit être unique dans toute la base. Le nom de l'index est optionnel. La clef étrangère peut référencer plusieurs champs en même temps mais tous les champs doivent être dans la même table.

Lors d'une suppression ou d'une mise à jour, le comportement est dicté par l'ajout d'une clause supplémentaire dans la clause FOREIGN KEY :

- **RESTRICT** ou **NO ACTION** - C'est l'action par défaut. Toute tentative de mise à jour ou de suppression dans la table parente provoque une erreur si une ou plusieurs lignes correspondantes se trouvent dans la table enfant,
- **CASCADE** - Quand une ligne est mise à jour ou supprimée de la table parente, la ou les lignes corrspondantes dans la table enfant sont mises à jour ou supprimées,
- **SET NULL** - Quand une ligne est mise à jour ou supprimée de la table parente, la ou les lignes corrspondantes dans la table enfant sont mises à NULL.

Index Cluster avec InnoDB

Le moteur InnoDB stocke les données **avec** la clef primaire. Par conséquent toute requête qui utilise la clef primaire est très efficace. Par contre toute requête qui utilise une clef secondaire est moins efficace car MySQL recherche d'abord dans l'index secondaire la clef primaire associée puis recherche les données à partir de la clef primaire.

> Dans le cas où une clef primaire n'existe pas, MySQL essaie de trouver une clef unique ne contentant aucune valeur NULL. S'il n'en trouve pas, il crée une clef primaire cachée.

Index Couvrant

Dans le cas où toutes les données nécessaires à la résolution d'une requête se trouvent dans un index, on parle d'un index **couvrant**. Dans ce cas,

l'optimiseur de MySQL n'a pas besoin de chercher les données.

Un index peut être créé avec un de deux algorithmes principaux : B-Tree ou Hash. Un index couvrant est forcément de type B-Tree.

Index FULLTEXT

Dans le cas où il est nécessaire de rechercher une chaîne de caractères dans un champs, on doit utiliser un index FULLTEXT :

```
mysql> CREATE TABLE t6(
    -> col1 VARCHAR(50),
    -> col2 VARCHAR(50),
    -> col3 VARCHAR(50),
    -> FULLTEXT index_col1_col2 (col1,col2)
    -> );
ERROR 1214 (HY000): The used table type doesn't support FULLTEXT indexes
```

Le moteur InnoDB supporte les index FULLTEXT à partir de la version 5.6 de MySQL.

```
mysql> CREATE TABLE t6( col1 VARCHAR(50), col2 VARCHAR(50), col3 VARCHAR(50), FULLTEXT index_col1_col2
(col1,col2) ) ENGINE=MyISAM;
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE FULLTEXT INDEX index_col2 ON t6 (col2);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE t6 ADD FULLTEXT index_col3 (col3);
```

```
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Téléchargez la base de donnée exemple sakila :

```
[root@centos6 ~]# wget http://downloads.mysql.com/docs/sakila-db.tar.gz
--2014-02-01 11:46:10--  http://downloads.mysql.com/docs/sakila-db.tar.gz
Résolution de downloads.mysql.com... 137.254.60.14
Connexion vers downloads.mysql.com|137.254.60.14|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 722633 (706K) [application/x-gzip]
Sauvegarde en : «sakila-db.tar.gz»

100%[======================================>] 722 633     20,2K/s   ds 14s

2014-02-01 11:46:30 (50,6 KB/s) - «sakila-db.tar.gz» sauvegardé [722633/722633]
```

Installez sakila :

```
[root@centos6 ~]# tar xvf sakila-db.tar.gz
sakila-db/
sakila-db/sakila-schema.sql
sakila-db/sakila.mwb
sakila-db/sakila-data.sql
[root@centos6 ~]# mysql -uroot -p < sakila-db/sakila-schema.sql
Enter password:
[root@centos6 ~]# mysql -uroot -p < sakila-db/sakila-data.sql
Enter password:
```

Mode Langage Naturel

L'appel à une recherche FULLTEXT nécessite l'utilisation de clauses spécifiques, à savoir **MATCH()** et **AGAINST()** :

```
mysql> USE sakila;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT film_id,
    -> LEFT(description,30),
    -> MATCH(title,description)
    -> AGAINST('Database Administrator') AS pertinence
    -> FROM film_text
    -> WHERE MATCH(title,description)
    -> AGAINST('Database Administrator')
    -> LIMIT 5;
+---------+--------------------------------+-------------------+
| film_id | LEFT(description,30)           | pertinence        |
+---------+--------------------------------+-------------------+
|     113 | A Thrilling Yarn of a Database | 6.662790775299072 |
|     962 | A Fanciful Story of a Database | 6.662790775299072 |
|      27 | A Amazing Reflection of a Data | 6.662790775299072 |
|     363 | A Fast-Paced Display of a Car  | 4.527374267578125 |
|     372 | A Taut Display of a Cat And a  | 4.527374267578125 |
+---------+--------------------------------+-------------------+
5 rows in set (0.00 sec)

mysql>
```

Par défaut une requête FULLTEXT est une recherches en langage naturel. Ceci implique que chaque mot de la clause AGAINST est recherché dans l'index et que la pertinence est calculé en fonction de la fréquences d'apparition. Notez que si un mot est peu fréquent dans l'ensemble des lignes indexées et ce mot fait partie de ceux recherchés, la pertinence de la ligne trouvée sera plus haute que la pertinence de la ligne contenant un mot peu qui est fréquent dans les lignes. Dernièrement et par défaut les mots faisant partie de au moins 50% des enregistrements sont automatiquement ignorés. Il n'est pas possible de modifier ce pourcentage.

Mode Booléen

En utilisant le mode booléen, il vous est possible d'ajouter des **modificateurs** pour préciser quels mots ont une pertinence plus élevée que d'autres :

Modificateur	Description
+	Le résultat doit contenir ce mot
-	Le résultat ne doit pas contenir ce mot
~	La pertinence de ce mot est plus faible que les autres mots
	La pertinence de ce mot est plus haute que les autres mots
*	Placé à la fin d'une chaine, ce modificateur est un joker

Par exemple :

```
mysql> SELECT film_id,title,description FROM film_text WHERE MATCH(title,description) AGAINST('Data*
Administrator +Anaconda' IN BOOLEAN MODE)\G*************************** 1. row ***************************
    film_id: 23
      title: ANACONDA CONFESSIONS
description: A Lacklusture Display of a Dentist And a Dentist who must Fight a Girl in Australia
*************************** 2. row ***************************
    film_id: 315
      title: FINDING ANACONDA
description: A Fateful Tale of a Database Administrator And a Girl who must Battle a Squirrel in New Orleans
2 rows in set (0.01 sec)

mysql>
```

> Notez que seulement deux résultats sont retournés. En effet, il n'y a que deux enregistrements dans la base qui contiennent le mot Anaconda. Notez que les mots faisant partie de au moins 50% des enregistrements **ne** sont **pas** automatiquement ignorés.

Mode Expansion de Requête

Dans ce mode, la recherche est lancée deux fois. La première fois avec les mots de la clause AGAINST, la deuxième avec les mots ainsi que les résultats les plus pertinents.

```
mysql> SELECT film_id,title,description FROM film_text WHERE MATCH(title,description) AGAINST('MySQL');
+---------+-----------------------+----------------------------------------------------------------------------
----------------------------------------+
| film_id | title                 | description
|
+---------+-----------------------+----------------------------------------------------------------------------
----------------------------------------+
|     428 | HOMICIDE PEACH        | A Astounding Documentary of a Hunter And a Boy who must Confront a Boy in A
MySQL Convention                        |
|     577 | MILE MULAN            | A Lacklusture Epistle of a Cat And a Husband who must Confront a Boy in A
MySQL Convention                        |
|     812 | SMOKING BARBARELLA    | A Lacklusture Saga of a Mad Cow And a Mad Scientist who must Sink a Cat in A
MySQL Convention                        |
|     936 | VANISHING ROCKY       | A Brilliant Reflection of a Man And a Woman who must Conquer a Pioneer in A
MySQL Convention                        |
|     494 | KARATE MOON           | A Astounding Yarn of a Womanizer And a Dog who must Reach a Waitress in A
MySQL Convention                        |
|     608 | MURDER ANTITRUST      | A Brilliant Yarn of a Car And a Database Administrator who must Escape a Boy
in A MySQL Convention                   |
|     341 | FROST HEAD            | A Amazing Reflection of a Lumberjack And a Cat who must Discover a Husband
in A MySQL Convention                   |
|     773 | SEABISCUIT PUNK       | A Insightful Saga of a Man And a Forensic Psychologist who must Discover a
Mad Cow in A MySQL Convention           |
|     242 | DOOM DANCING          | A Astounding Panorama of a Car And a Mad Scientist who must Battle a
Lumberjack in A MySQL Convention                |
|     819 | SONG HEDWIG           | A Amazing Documentary of a Man And a Husband who must Confront a Squirrel in
A MySQL Convention                      |
|      15 | ALIEN CENTER          | A Brilliant Drama of a Cat And a Mad Scientist who must Battle a Feminist in
A MySQL Convention                      |
|     137 | CHARADE DUFFEL        | A Action-Packed Display of a Man And a Waitress who must Build a Dog in A
MySQL Convention                        |
```

```
|     844 | STEERS ARMAGEDDON      | A Stunning Character Study of a Car And a Girl who must Succumb a Car in A
MySQL Convention                          |
|     119 | CAPER MOTIONS          | A Fateful Saga of a Moose And a Car who must Pursue a Woman in A MySQL
Convention                                |
|     907 | TRANSLATION SUMMER     | A Touching Reflection of a Man And a Monkey who must Pursue a Womanizer in A
MySQL Convention                          |
|     899 | TOWERS HURRICANE       | A Fateful Display of a Monkey And a Car who must Sink a Husband in A MySQL
Convention                                |
|     918 | TWISTED PIRATES        | A Touching Display of a Frisbee And a Boat who must Kill a Girl in A MySQL
Convention                                |
|     933 | VAMPIRE WHALE          | A Epic Story of a Lumberjack And a Monkey who must Confront a Pioneer in A
MySQL Convention                          |
|     716 | REAP UNFAITHFUL        | A Thrilling Epistle of a Composer And a Sumo Wrestler who must Challenge a
Mad Cow in A MySQL Convention             |
|     727 | RESURRECTION SILVERADO | A Epic Yarn of a Robot And a Explorer who must Challenge a Girl in A MySQL
Convention                                |
|     809 | SLIPPER FIDELITY       | A Taut Reflection of a Secret Agent And a Man who must Redeem a Explorer in
A MySQL Convention                        |
|     567 | MEET CHOCOLATE         | A Boring Documentary of a Dentist And a Butler who must Confront a Monkey in
A MySQL Convention                        |
|     129 | CAUSE DATE             | A Taut Tale of a Explorer And a Pastry Chef who must Conquer a Hunter in A
MySQL Convention                          |
|     261 | DUFFEL APOCALYPSE      | A Emotional Display of a Boat And a Explorer who must Challenge a Madman in
A MySQL Convention                        |
|     213 | DATE SPEED             | A Touching Saga of a Composer And a Moose who must Discover a Dentist in A
MySQL Convention                          |
|      11 | ALAMO VIDEOTAPE        | A Boring Epistle of a Butler And a Cat who must Fight a Pastry Chef in A
MySQL Convention                          |
|     201 | CYCLONE FAMILY         | A Lacklusture Drama of a Student And a Monkey who must Sink a Womanizer in A
MySQL Convention                          |
|     352 | GATHERING CALENDAR     | A Intrepid Tale of a Pioneer And a Moose who must Conquer a Frisbee in A
MySQL Convention                          |
|     398 | HANOVER GALAXY         | A Stunning Reflection of a Girl And a Secret Agent who must Succumb a Boy in
A MySQL Convention                        |
```

```
|     426 | HOME PITY              | A Touching Panorama of a Man And a Secret Agent who must Challenge a Teacher
in A MySQL Convention                    |
|      72 | BILL OTHERS            | A Stunning Saga of a Mad Scientist And a Forensic Psychologist who must
Challenge a Squirrel in A MySQL Convention   |
|     845 | STEPMOM DREAM          | A Touching Epistle of a Crocodile And a Teacher who must Build a Forensic
Psychologist in A MySQL Convention        |
|     870 | SWARM GOLD             | A Insightful Panorama of a Crocodile And a Boat who must Conquer a Sumo
Wrestler in A MySQL Convention              |
|     466 | INTOLERABLE INTENTIONS | A Awe-Inspiring Story of a Monkey And a Pastry Chef who must Succumb a
Womanizer in A MySQL Convention             |
|     974 | WILD APOLLO            | A Beautiful Story of a Monkey And a Sumo Wrestler who must Conquer a A Shark
in A MySQL Convention                    |
|     980 | WIZARD COLDBLOODED     | A Lacklusture Display of a Robot And a Girl who must Defeat a Sumo Wrestler
in A MySQL Convention                     |
|     551 | MAIDEN HOME            | A Lacklusture Saga of a Moose And a Teacher who must Kill a Forensic
Psychologist in A MySQL Convention            |
|     987 | WORDS HUNTER           | A Action-Packed Reflection of a Composer And a Mad Scientist who must Face a
Pioneer in A MySQL Convention           |
|     183 | CONVERSATION DOWNHILL  | A Taut Character Study of a Husband And a Waitress who must Sink a Squirrel
in A MySQL Convention                     |
|     804 | SLEEPING SUSPECTS      | A Stunning Reflection of a Sumo Wrestler And a Explorer who must Sink a
Frisbee in A MySQL Convention                |
|     576 | MIGHTY LUCK            | A Astounding Epistle of a Mad Scientist And a Pioneer who must Escape a
Database Administrator in A MySQL Convention |
|     733 | RIVER OUTLAW           | A Thrilling Character Study of a Squirrel And a Lumberjack who must Face a
Hunter in A MySQL Convention              |
|     303 | FANTASY TROOPERS       | A Touching Saga of a Teacher And a Monkey who must Overcome a Secret Agent
in A MySQL Convention                     |
|     937 | VARSITY TRIP           | A Action-Packed Character Study of a Astronaut And a Explorer who must Reach
a Monkey in A MySQL Convention           |
|     782 | SHAKESPEARE SADDLE     | A Fast-Paced Panorama of a Lumberjack And a Database Administrator who must
Defeat a Madman in A MySQL Convention    |
|     114 | CAMELOT VACATION       | A Touching Character Study of a Woman And a Waitress who must Battle a
Pastry Chef in A MySQL Convention            |
```

```
|     971 | WHALE BIKINI           | A Intrepid Story of a Pastry Chef And a Database Administrator who must Kill
a Feminist in A MySQL Convention        |
|     822 | SOUP WISDOM            | A Fast-Paced Display of a Robot And a Butler who must Defeat a Butler in A
MySQL Convention                          |
|     107 | BUNCH MINDS            | A Emotional Story of a Feminist And a Feminist who must Escape a Pastry Chef
in A MySQL Convention                     |
|     750 | RUN PACIFIC            | A Touching Tale of a Cat And a Pastry Chef who must Conquer a Pastry Chef in
A MySQL Convention                        |
+---------+------------------------+-------------------------------------------------------------------------
----------------------------------------+
50 rows in set (0.01 sec)

mysql>
```

```
mysql> SELECT film_id,title,description FROM film_text WHERE MATCH(title,description) AGAINST('MySQL' WITH QUERY
EXPANSION);
...
|     113 | CALIFORNIA BIRDS           | A Thrilling Yarn of a Database Administrator And a Robot who must
Battle a Database Administrator in Ancient India                |
|     617 | NATURAL STOCK              | A Fast-Paced Story of a Sumo Wrestler And a Girl who must Defeat a Car
in A Baloon Factory                                    |
|     424 | HOLOCAUST HIGHBALL         | A Awe-Inspiring Yarn of a Composer And a Man who must Find a Robot in
Soviet Georgia                                          |
|     589 | MODERN DORADO              | A Awe-Inspiring Story of a Butler And a Sumo Wrestler who must Redeem a
Boy in New Orleans                                     |
|     432 | HOPE TOOTSIE               | A Amazing Documentary of a Student And a Sumo Wrestler who must Outgun
a A Shark in A Shark Tank                               |
|     145 | CHISUM BEHAVIOR            | A Epic Documentary of a Sumo Wrestler And a Butler who must Kill a Car
in Ancient India                                       |
|      41 | ARSENIC INDEPENDENCE       | A Fanciful Documentary of a Mad Cow And a Womanizer who must Find a
Dentist in Berlin                                        |
|     751 | RUNAWAY TENENBAUMS         | A Thoughtful Documentary of a Boat And a Man who must Meet a Boat in An
Abandoned Fun House                                    |
|     962 | WASTELAND DIVINE           | A Fanciful Story of a Database Administrator And a Womanizer who must
```

```
Fight a Database Administrator in Ancient China               |
|     177 | CONNECTICUT TRAMP           | A Unbelieveable Drama of a Crocodile And a Mad Cow who must Reach a
Dentist in A Shark Tank                                        |
|     627 | NORTH TEQUILA               | A Beautiful Character Study of a Mad Cow And a Robot who must Reach a
Womanizer in New Orleans                                       |
|     702 | PULP BEVERLY                | A Unbelieveable Display of a Dog And a Crocodile who must Outrace a Man
in Nigeria                                                     |
|     511 | LAWRENCE LOVE               | A Fanciful Yarn of a Database Administrator And a Mad Cow who must
Pursue a Womanizer in Berlin                                    |
|     376 | GRAPES FURY                 | A Boring Yarn of a Mad Cow And a Sumo Wrestler who must Meet a Robot in
Australia                                                      |
|     430 | HOOK CHARIOTS               | A Insightful Story of a Boy And a Dog who must Redeem a Boy in
Australia                                                       |
|     267 | EAGLES PANKY                | A Thoughtful Story of a Car And a Boy who must Find a A Shark in The
Sahara Desert                                                  |
|     968 | WEREWOLF LOLA               | A Fanciful Story of a Man And a Sumo Wrestler who must Outrace a
Student in A Monastery                                          |
|     692 | POTLUCK MIXED               | A Beautiful Story of a Dog And a Technical Writer who must Outgun a
Student in A Baloon                                            |
|     192 | CROSSING DIVORCE            | A Beautiful Documentary of a Dog And a Robot who must Redeem a
Womanizer in Berlin                                             |
|     752 | RUNNER MADIGAN              | A Thoughtful Documentary of a Crocodile And a Robot who must Outrace a
Womanizer in The Outback                                       |
|     448 | IDAHO LOVE                  | A Fast-Paced Drama of a Student And a Crocodile who must Meet a
Database Administrator in The Outback                           |
|     996 | YOUNG LANGUAGE              | A Unbelieveable Yarn of a Boat And a Database Administrator who must
Meet a Boy in The First Manned Space Station                    |
+---------+-----------------------------+-------------------------------------------------------------------------
-------------------------------------------------------------+
999 rows in set (0.02 sec)

mysql>
```

Configuration

Une recherche FULLTEXT peut être configurée en modifiant les trois variables suivants :

```
mysql> SHOW VARIABLES LIKE 'ft_min_word_len';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| ft_min_word_len | 4     |
+------------------+-------+
1 row in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'ft_max_word_len';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| ft_max_word_len | 84    |
+------------------+-------+
1 row in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'ft_stopword_list';
Empty set (0.00 sec)

mysql>
```

La dernière variable peut contenir une liste de mots qui ne seront pas indexés. Notez que chaque fois que vous modifiez une des ces variables, il est nécessaire de reconstruire l'index FULLTEXT avec la commande REPAIR TABLE table QUICK;

Limitations

Veuillez garder en tête les points suivants :

- Pour effectuer une recherche FULLTEXT, il **faut** utiliser MATCH AGAINST,
- Dès que vous utilisez MATCH AGAINST, le serveur utilisera une recherche FULLTEXT même si l'optimiseur trouve un autre index qui aurait été

meilleur,
- Les résultats en mode langage naturel sont triés par la pertinence. Ceci n'est **pas** le cas pour les deux autres modes,
- Un index FULLTEXT ne peut **pas** être utilisé comme index couvrant.

La Commande EXPLAIN

La commande EXPLAIN est utilisée pour voir le plan d'exécution de la requête choisi par l'optimiseur :

```
mysql> EXPLAIN SELECT film_id,title,description FROM film_text WHERE MATCH(title,description) AGAINST('MySQL')\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: film_text
         type: fulltext
possible_keys: idx_title_description
          key: idx_title_description
      key_len: 0
          ref:
         rows: 1
        Extra: Using where
1 row in set (0.00 sec)

mysql>
```

La sortie de la commande démontre 10 colonnes dont les plus importantes sont **type**, **possible_keys**, **keys**, **key_len**, **rows** et **extra** :

Pour l'explication des autres colonnes, voir **ce lien**.

La Colonne type

Cette colonne montre comment l'optimiseur a prévu d'accéder aux données. Les cas suivants peuvent être trouvés en sachant qu'ils sont dans l'ordre du moins performant vers le plus performant :

ALL

Un parcours complet de la table est nécessaire car il n'y a pas d'index disponible ou satisfaisant :

```
mysql> EXPLAIN SELECT * FROM film_text WHERE title LIKE 'The%'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: film_text
         type: ALL
possible_keys: idx_title_description
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 1000
        Extra: Using where
1 row in set (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM film_text WHERE title LIKE 'The%'\G
*************************** 1. row ***************************
    film_id: 886
      title: THEORY MERMAID
description: A Fateful Yarn of a Composer And a Monkey who must Vanquish a Womanizer in The First Manned Space
Station
1 row in set (0.00 sec)
```

```
mysql>
```

index

Un parcours complet d'un index est necéssaire :

```
mysql> EXPLAIN SELECT category_id FROM category\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: category
         type: index
possible_keys: NULL
          key: PRIMARY
      key_len: 1
          ref: NULL
         rows: 16
        Extra: Using index
1 row in set (0.01 sec)

mysql>
```

```
mysql> SELECT category_id FROM category\G
*************************** 1. row ***************************
category_id: 1
*************************** 2. row ***************************
category_id: 2
*************************** 3. row ***************************
category_id: 3
*************************** 4. row ***************************
category_id: 4
*************************** 5. row ***************************
category_id: 5
```

```
*************************** 6. row ***************************
category_id: 6
*************************** 7. row ***************************
category_id: 7
*************************** 8. row ***************************
category_id: 8
*************************** 9. row ***************************
category_id: 9
*************************** 10. row ***************************
category_id: 10
*************************** 11. row ***************************
category_id: 11
*************************** 12. row ***************************
category_id: 12
*************************** 13. row ***************************
category_id: 13
*************************** 14. row ***************************
category_id: 14
*************************** 15. row ***************************
category_id: 15
*************************** 16. row ***************************
category_id: 16
16 rows in set (0.00 sec)

mysql>
```

range

Un parcours d'une partie d'un index est nécessaire :

```
mysql> EXPLAIN SELECT * FROM category WHERE category_id > 50\G
*************************** 1. row ***************************
           id: 1
```

```
  select_type: SIMPLE
        table: category
         type: range
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 1
          ref: NULL
         rows: 1
        Extra: Using where
1 row in set (0.01 sec)

mysql>
```

```
mysql> SELECT * FROM category WHERE category_id > 50\G
Empty set (0.01 sec)

mysql>
```

index_merge

Plusieurs index sont utilisés simultanément :

```
mysql> EXPLAIN SELECT * FROM rental WHERE rental_id > 10 OR inventory_id < 100\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: index_merge
possible_keys: PRIMARY,idx_fk_inventory_id
          key: idx_fk_inventory_id,PRIMARY
      key_len: 3,4
          ref: NULL
         rows: 8557
```

```
        Extra: Using sort_union(idx_fk_inventory_id,PRIMARY); Using where
1 row in set (0.01 sec)

mysql>
```

```
mysql> SELECT * FROM rental WHERE rental_id > 10 OR inventory_id < 100\G
...
*************************** 13776. row ***************************
   rental_id: 13790
 rental_date: 2005-08-20 12:17:27
inventory_id: 1385
 customer_id: 11
 return_date: 2005-08-25 12:20:27
    staff_id: 1
 last_update: 2006-02-15 21:30:53
*************************** 13777. row ***************************
   rental_id: 13791
 rental_date: 2005-08-20 12:21:05
inventory_id: 1890
 customer_id: 67
 return_date: 2005-08-22 17:58:05
    staff_id: 1
 last_update: 2006-02-15 21:30:53
*************************** 13778. row ***************************
   rental_id: 13792
 rental_date: 2005-08-20 12:21:37
inventory_id: 4157
 customer_id: 78
 return_date: 2005-08-27 14:28:37
    staff_id: 1
 last_update: 2006-02-15 21:30:53
*************************** 13779. row ***************************
   rental_id: 13793
 rental_date: 2005-08-20 12:22:04
```

```
inventory_id: 2598
 customer_id: 424
 return_date: 2005-08-27 09:51:04
    staff_id: 2
 last_update: 2006-02-15 21:30:53
...
```

ref

Un index non unique est examiné. Plusieurs lignes peuvent être retournées :

```
mysql> EXPLAIN SELECT rental_date FROM rental INNER JOIN customer USING (customer_id) WHERE customer_id=1\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: customer
         type: const
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 2
          ref: const
         rows: 1
        Extra: Using index
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: ref
possible_keys: idx_fk_customer_id
          key: idx_fk_customer_id
      key_len: 2
          ref: const
         rows: 32
```

```
        Extra:
2 rows in set (0.01 sec)

mysql>
```

```
mysql> SELECT rental_date FROM rental INNER JOIN customer USING (customer_id) WHERE customer_id=1\G
*************************** 1. row ***************************
rental_date: 2005-05-25 11:30:37
*************************** 2. row ***************************
rental_date: 2005-05-28 10:35:23
*************************** 3. row ***************************
rental_date: 2005-06-15 00:54:12
*************************** 4. row ***************************
rental_date: 2005-06-15 18:02:53
*************************** 5. row ***************************
rental_date: 2005-06-15 21:08:46
*************************** 6. row ***************************
rental_date: 2005-06-16 15:18:57
*************************** 7. row ***************************
rental_date: 2005-06-18 08:41:48
*************************** 8. row ***************************
rental_date: 2005-06-18 13:33:59
*************************** 9. row ***************************
rental_date: 2005-06-21 06:24:45
*************************** 10. row ***************************
rental_date: 2005-07-08 03:17:05
*************************** 11. row ***************************
rental_date: 2005-07-08 07:33:56
*************************** 12. row ***************************
rental_date: 2005-07-09 13:24:07
*************************** 13. row ***************************
rental_date: 2005-07-09 16:38:01
*************************** 14. row ***************************
rental_date: 2005-07-11 10:13:46
```

```
*************************** 15. row ***************************
rental_date: 2005-07-27 11:31:22
*************************** 16. row ***************************
rental_date: 2005-07-28 09:04:45
*************************** 17. row ***************************
rental_date: 2005-07-28 16:18:23
*************************** 18. row ***************************
rental_date: 2005-07-28 17:33:39
*************************** 19. row ***************************
rental_date: 2005-07-28 19:20:07
*************************** 20. row ***************************
rental_date: 2005-07-29 03:58:49
*************************** 21. row ***************************
rental_date: 2005-07-31 02:42:18
*************************** 22. row ***************************
rental_date: 2005-08-01 08:51:04
*************************** 23. row ***************************
rental_date: 2005-08-02 15:36:52
*************************** 24. row ***************************
rental_date: 2005-08-02 18:01:38
*************************** 25. row ***************************
rental_date: 2005-08-17 12:37:54
*************************** 26. row ***************************
rental_date: 2005-08-18 03:57:29
*************************** 27. row ***************************
rental_date: 2005-08-19 09:55:16
*************************** 28. row ***************************
rental_date: 2005-08-19 13:56:54
*************************** 29. row ***************************
rental_date: 2005-08-21 23:33:57
*************************** 30. row ***************************
rental_date: 2005-08-22 01:27:57
*************************** 31. row ***************************
rental_date: 2005-08-22 19:41:37
```

```
*************************** 32. row ***************************
rental_date: 2005-08-22 20:03:46
32 rows in set (0.00 sec)

mysql>
```

Il existe une variante de ref, appelé ref_or_null. Dans ce cas, il y a une deuxième passe pour retrouver les valeurs NULL.

eq_ref

Un index unique ou une clé primaire est examiné :

```
mysql> EXPLAIN SELECT email FROM customer INNER JOIN rental USING (customer_id) WHERE rental_date > '2006-04-10
00:00:00' AND RENTAL_DATE < '2006-05-24 00:00:00'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: range
possible_keys: rental_date,idx_fk_customer_id
          key: rental_date
      key_len: 8
          ref: NULL
         rows: 1
        Extra: Using where; Using index
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: customer
         type: eq_ref
```

```
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 2
          ref: sakila.rental.customer_id
         rows: 1
        Extra:
2 rows in set (0.00 sec)

mysql>
```

Ce type est le meilleur t(ype d'accès possible.

Cas Spécifiques

const

Une seule ligne a besoin d'être lue dans la table concernée :

```
mysql> EXPLAIN SELECT * FROM category WHERE category_id=1\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: category
         type: const
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 1
          ref: const
         rows: 1
        Extra:
```

```
1 row in set (0.00 sec)

mysql> SELECT * FROM category WHERE category_id=1\G
*************************** 1. row ***************************
category_id: 1
       name: Action
last_update: 2006-02-15 04:46:27
1 row in set (0.00 sec)

mysql>
```

System

Identique au type précédent dans le cas où la table est une table à une seule ligne.

NULL

Aucun parcours de table ou d'index n'est nécessaire :

```
mysql> EXPLAIN SELECT COUNT(*) FROM film_text\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: NULL
         type: NULL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: NULL
        Extra: Select tables optimized away
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM film_text\G
```

```
*************************** 1. row ***************************
COUNT(*): 1000
1 row in set (0.01 sec)

mysql>
```

Les Colonnes possible_keys, keys et key_len

La colonne **possible_keys** contient l'ensemble des index que l'optimiseur va considérer pour la requête concernée tandis que la colonne **key** indique l'index qui serait utiliser.

Une valeur NULL dans la colonne **possible_keys** indique qu'il n'existe pas d'index utilisable tandis que la valeur NULL dans la colonne **key** indique que l'optimiseur n'a pas trouvé d'index pertinent.

La colonne **key_len** indique la taille en octets de l'index choisi.

Téléchargez la base de données exemple **world** :

```
[root@centos6 ~]# wget http://downloads.mysql.com/docs/world.sql.gz
```

Installez la base de données exemple **world** :

```
[root@centos6 ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.5.33-cll-lve MySQL Community Server (GPL) by Atomicorp

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE world;
Query OK, 1 row affected (0.00 sec)

mysql> exit
Bye
[root@centos6 ~]# gunzip world.sql.gz
[root@centos6 ~]# mysql -uroot -p world < world.sql
Enter password:
[root@centos6 ~]#
```

Voyons un exemple maintenant avec cette base de données :

```
mysql> USE world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ALTER TABLE City ADD INDEX idx_name_district(Name,District);
Query OK, 4079 rows affected (0.11 sec)
Records: 4079  Duplicates: 0  Warnings: 0

mysql> EXPLAIN SELECT * FROM City WHERE Name LIKE 'San%'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: City
         type: range
possible_keys: idx_name_district
          key: idx_name_district
      key_len: 35
          ref: NULL
         rows: 141
```

```
        Extra: Using where
1 row in set (0.01 sec)

mysql>
```

Notez que la taille de l'index utilisé est de 35 octets.

La Colonne rows

La colonne **rows** contient une estimation du nombre de lignes que le moteur doit parcourir avant de pouvoir retourner un résultat de la requête :

```
mysql> EXPLAIN SELECT * FROM City CROSS JOIN Country\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: Country
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 239
        Extra:
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: City
         type: ALL
possible_keys: NULL
          key: NULL
```

```
      key_len: NULL
          ref: NULL
         rows: 4079
        Extra: Using join buffer
2 rows in set (0.00 sec)

mysql>
```

Dans l'exemple ci-dessus, MySQL estime avoir besoin de parcourir 239*4079 lignes soit 974 881 enregistrements !!

Pour prouver qu'il s'agit bien d'une estimation, saisissez la requête suivante :

```
mysql> EXPLAIN SELECT * FROM City WHERE Name LIKE 'San%'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: City
         type: range
possible_keys: idx_name_district
          key: idx_name_district
      key_len: 35
          ref: NULL
         rows: 141
        Extra: Using where
1 row in set (0.05 sec)

mysql> SELECT COUNT(*) FROM City WHERE Name LIKE 'San%'\G
*************************** 1. row ***************************
COUNT(*): 110
1 row in set (0.02 sec)
```

```
mysql>
```

Si vous vous rendez compte que les estimations sont largement erronées vous pouvez utilisez la commande ANALYZE TABLE pour mettre à jour les statistiques sur la table. D'une manière générale, plus élevée est la valeur de **ref**, plus lente sera la requête.

La Colonne extra

La colonne extra donne des informations supplémentaires concernant la requête :

- **Using where** - visible quand le serveur n'a pas pu résoudre une clause WHERE en utilisant un index,
- **Using index** - le serveur va utiliser un index couvrant,
- **Using temporary** - le serveur va utiliser une table temporaire pour effectuer un tri,
- **Using filesort** - le serveur doit trier les résultats de la requête.

La Commande EXPLAIN EXTENDED

La commande **EXPLAIN EXTENDED** est disponible dans MySQL depuis la version 5.1. EXPLAIN EXTENDED donne une approximation, en pourcentage par rapport à la valeur de **ref**, du nombre de lignes qui seront retournées lors de la requête :

```
mysql> EXPLAIN EXTENDED SELECT ID FROM City WHERE ID < 50\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: City
         type: range
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 4
          ref: NULL
```

```
         rows: 50
     filtered: 100.00
        Extra: Using where; Using index
1 row in set, 1 warning (0.00 sec)

mysql> SELECT COUNT(ID) FROM City WHERE ID < 50\G
*************************** 1. row ***************************
COUNT(ID): 49
1 row in set (0.01 sec)

mysql>
```

Par contre cette valeur n'est qu'une estimation :

```
mysql> EXPLAIN EXTENDED SELECT Name FROM City WHERE CountryCode LIKE 'F%' AND district LIKE 'F%' AND population
>100000\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: City
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 4079
     filtered: 100.00
        Extra: Using where
1 row in set, 1 warning (0.00 sec)

mysql> SELECT Name FROM City WHERE CountryCode LIKE 'F%' AND district LIKE 'F%' AND population >100000\G
*************************** 1. row ***************************
Name: Besançon
1 row in set (0.07 sec)
```

```
mysql>
```

Malgré le peu de fiabilité dans le résultat de la colonne **filtered**, EXPLAIN EXTENDED est une clause bien utile car elle permet de voir la requête réécrite à partir du plan d'exécution :

```
mysql> EXPLAIN EXTENDED SELECT COUNT(*) FROM Country\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: NULL
         type: NULL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: NULL
     filtered: NULL
        Extra: Select tables optimized away
1 row in set, 1 warning (0.00 sec)

mysql> SHOW WARNINGS\G
*************************** 1. row ***************************
  Level: Note
   Code: 1003
Message: select count(0) AS `COUNT(*)` from `world`.`Country`
1 row in set (0.00 sec)

mysql>
```

Optimisation des Requêtes

Isolation des Colonnes

L'encapsulation d'une colonne dans une fonction interdit l'utilisation d'un index :

```
mysql> USE sakila;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> EXPLAIN SELECT * FROM rental WHERE TO_DAYS(CURRENT_DATE()) - TO_DAYS(rental_date) < 10\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 15629
        Extra: Using where
1 row in set (0.00 sec)

mysql> EXPLAIN SELECT * FROM rental WHERE rental_date > CURRENT_DATE() + INTERVAL 10 DAY\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: range
possible_keys: rental_date
          key: rental_date
      key_len: 8
          ref: NULL
```

```
         rows: 1
        Extra: Using where
1 row in set (0.00 sec)

mysql>
```

La réécriture de la requête sans encapsuler une colonne dans une fonction produit une requête plus performante (range > ALL).

Jointures

Quand MySQL effectue des jointures il le fait une à une et successivement. Par exemple, dans le cas de 4 tables, table1, table2, table3, table4, la table1 est jointe à la table2 produisant une table table12. La table12 est ensite jointe à la table3 produisant la table123. La table123 est ensute joite à la table4 pour produire la table1234.

Dans le cas d'un INNER JOIN, MySQL peut décider de ne pas respecter cet ordre :

```
mysql> EXPLAIN SELECT email FROM customer INNER JOIN rental ON customer.customer_id = rental.customer_id WHERE
rental_date > '2006-04-10 00:00:00' AND rental_date < '2006-05-24 00:00:00'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: range
possible_keys: rental_date,idx_fk_customer_id
          key: rental_date
      key_len: 8
          ref: NULL
         rows: 1
        Extra: Using where; Using index
*************************** 2. row ***************************
```

```
           id: 1
  select_type: SIMPLE
        table: customer
         type: eq_ref
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 2
          ref: sakila.rental.customer_id
         rows: 1
        Extra:
2 rows in set (0.01 sec)

mysql>
```

L'ordre de jointure ici est rental vers customer et non pas customer vers rental.

Pourquoi l'optimiseur a choisi d'inverser l'ordre de jointure ? Dans l'exemple précédent il faut parcourir 1 ligne. Forcez maintenant la jointure dans l'autre direction :

```
mysql> EXPLAIN SELECT email FROM customer STRAIGHT_JOIN rental ON customer.customer_id = rental.customer_id WHERE
rental_date > '2006-04-10 00:00:00' AND rental_date < '2006-05-24 00:00:00'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: customer
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 577
```

```
       Extra:
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: range
possible_keys: rental_date,idx_fk_customer_id
          key: rental_date
      key_len: 8
          ref: NULL
         rows: 1
        Extra: Using where; Using index; Using join buffer
2 rows in set (0.00 sec)

mysql>
```

Dans le cas ci-dessus, l'optimiseur a vu qu'il faut parcourir 577*1 lignes !!

Pour les jointure externes, LEFT JOIN et RIGHT JOIN, l'ordre est important.

Indexes

USE INDEX

Il est possible d'indiquer à MySQL d'utiliser un index spécifique ou d'ignorer un index :

```
mysql> EXPLAIN SELECT email FROM customer INNER JOIN rental USE INDEX (idx_fk_customer_id) ON
customer.customer_id = rental.customer_id WHERE rental_date > '2006-04-10 00:00:00' AND rental_date < '2006-05-24
00:00:00'\G
```

```
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: customer
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 577
        Extra:
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: ref
possible_keys: idx_fk_customer_id
          key: idx_fk_customer_id
      key_len: 2
          ref: sakila.customer.customer_id
         rows: 13
        Extra: Using where
2 rows in set (0.00 sec)
```

Dans le cas d'utilisation de la clause USE INDEX, l'auteur de la requête indique à l'optimiseur d'utiliser l'index passé en paramètre. Cependant l'optimiseur peut décider d'utiliser un autre index si celui-si s'avère plus approprié. Dans le cas ci-dessus l'optimiseur a été d'accord avec l'auteur.

FORCE INDEX

```
mysql> EXPLAIN SELECT email FROM customer INNER JOIN rental FORCE INDEX (idx_fk_customer_id) ON
customer.customer_id = rental.customer_id WHERE rental_date > '2006-04-10 00:00:00' AND rental_date < '2006-05-24
00:00:00'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: customer
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 577
        Extra:
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: rental
         type: ref
possible_keys: idx_fk_customer_id
          key: idx_fk_customer_id
      key_len: 2
          ref: sakila.customer.customer_id
         rows: 13
        Extra: Using where
2 rows in set (0.01 sec)
```

Dans le cas d'utilisation de la clause FORCE INDEX, l'auteur de la requête indique à l'optimiseur d'utiliser l'index passé en paramètre. L'index sera utiliser sauf dans le cas où il n'est pas utilisable.

IGNORE INDEX

mysql> EXPLAIN SELECT email FROM customer INNER JOIN rental IGNORE INDEX (idx_fk_customer_id,rental_date) ON customer.customer_id = rental.customer_id WHERE rental_date > '2006-04-10 00:00:00' AND rental_date < '2006-05-24 00:00:00'\G *** 1. row** *

```
         id: 1
select_type: SIMPLE
      table: rental
       type: ALL
```

possible_keys: NULL

```
        key: NULL
    key_len: NULL
        ref: NULL
       rows: 15629
      Extra: Using where
```

*** 2. row** *

```
         id: 1
select_type: SIMPLE
      table: customer
       type: eq_ref
```

possible_keys: PRIMARY

```
        key: PRIMARY
    key_len: 2
        ref: sakila.rental.customer_id
       rows: 1
      Extra:
```

2 rows in set (0.01 sec)

mysql> </code>

> Dans le cas d'utilisation de la clause IGNORE INDEX, l'auteur de la requête indique à l'optimiseur de ne **pas** utiliser le ou les index passés en paramètre. De ce fait notez l'inversion de l'ordre de jointure.

CLAUSES LENTES

Les clauses d'agrégation telles SUM(), MAX() et MIN() provoquent en générale des requêtes lentes car il faut lire l'ensemble de la table pour obtenir un résultat. C'est le même cas pour la clause COUNT() avec le moteur InnoDB, mais pas avec le moteur MyISAM. Le moteur MyISAM stocke dans ces métadonnées le nombre d'enregistrements de la table :

```
mysql> SHOW TABLE STATUS WHERE Name = 'film_text'\G
*************************** 1. row ***************************
           Name: film_text
         Engine: MyISAM
        Version: 10
     Row_format: Dynamic
           Rows: 1000
 Avg_row_length: 119
    Data_length: 119616
Max_data_length: 281474976710655
   Index_length: 205824
      Data_free: 0
 Auto_increment: NULL
    Create_time: 2014-02-01 11:48:00
    Update_time: 2014-02-01 11:48:16
     Check_time: NULL
      Collation: utf8_general_ci
```

```
       Checksum: NULL
 Create_options:
        Comment:
1 row in set (0.00 sec)

mysql> EXPLAIN SELECT COUNT(*) FROM film_text\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: NULL
         type: NULL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: NULL
        Extra: Select tables optimized away
1 row in set (0.00 sec)

mysql>
```

Dans le cas ci-dessus la valeur de la colonne **type** est **NULL** indiquant qu'aucun accès à la table n'est nécessaire.

Sous-requêtes

MySQL exécute parfois très lentement des sous-requêtes. Privilégiez donc les jointures plutôt que des sous-requêtes :

```
mysql> USE world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> SELECT Language FROM CountryLanguage WHERE CountryCode = (SELECT Code FROM Country WHERE Name='France');
+------------+
| Language   |
+------------+
| Arabic     |
| French     |
| Italian    |
| Portuguese |
| Spanish    |
| Turkish    |
+------------+
6 rows in set (0.03 sec)

mysql> SELECT Language FROM CountryLanguage INNER JOIN Country ON Code=CountryCode WHERE Name='France';
+------------+
| Language   |
+------------+
| Arabic     |
| French     |
| Italian    |
| Portuguese |
| Spanish    |
| Turkish    |
+------------+
6 rows in set (0.01 sec)

mysql>
```

Moteurs

MyISAM

Le paramètre le plus important pour le moteur MyISAM est le key_buffer_size :

```
mysql> SHOW GLOBAL VARIABLES LIKE 'key_buffer_size';
+-----------------+---------+
| Variable_name   | Value   |
+-----------------+---------+
| key_buffer_size | 8388608 |
+-----------------+---------+
1 row in set (0.00 sec)

mysql> SHOW GLOBAL STATUS LIKE 'Key_%';
+------------------------+-------+
| Variable_name          | Value |
+------------------------+-------+
| Key_blocks_not_flushed | 0     |
| Key_blocks_unused      | 7239  |
| Key_blocks_used        | 9     |
| Key_read_requests      | 39    |
| Key_reads              | 9     |
| Key_write_requests     | 0     |
| Key_writes             | 0     |
+------------------------+-------+
7 rows in set (0.01 sec)

mysql>
```

La variable **Key_reads** indique le nombre de requêtes de lectures qui n'ont pas pu être satisfaites par le cache. La variable **Key_read_requests** indique le nombre de lectures totale. La formule (1 - Key_reads / Key_read_requests) * 100 donne une indication de l'efficacité du cache. Dans notre cas la valeur est de 76,92%. Une valeur basse indique qu'il faut augmenter la taille de key_buffer_size.

InnoDB

Les paramètres les plus importants pour le moteur **InnoDB** sont résumés dans la table suivante :

Paramètre	Valeur Suggérée
innodb_buffer_size	80% de la mémoire physique de la machine
innodb_flush_at_trx_commit	1 si la protection des données est primordiale, 2 si la performance est optimale

Partitionnement

Le partitionnement est utilisé pour diviser les données d'une table en parties logiques selon une règle pré-établie :

- Par les données contenues dans la table. Dans ce cas on parle de partitionnement **horizontal**. Il existe 4 types possibles :
  - **Par plages**,
  - **Par listes**,
  - **Par hachage**,
  - **Par clef**,
- Par les colonnes de la table. Dans ce cas on parle de partitionnement **vertical**.

> Il est possible de faire un partitionnement vertical et un partitionnement horizontal sur la même table. Il est aussi possible de faire des partitions de partitions.

Le partitionnement apporte trois avantages :

- Créer des tables plus grandes que la taille autorisée par un système de fichiers. Il est aussi possible de stocker des partitions à des endroits différents,
- Supprimer très rapidement des données en détruisant la partition qui les contient. Ceci s'appelle le **Scaling Back**,
- Optimiser certaines requêtes car les données étant organisées dans différentes partitions. Ceci s'appelle le **Partition Pruning**.

Partitionnement Horizontal

LAB #9 - Partitionnement par Plages

Créez une base de données **transactions** ainsi que la table **transac** :

```
CREATE DATABASE transactions;

USE transactions;

CREATE TABLE transac
(
id INT UNSIGNED PRIMARY KEY,
montant INT UNSIGNED NOT NULL,
jour DATE NOT NULL,
codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
);
```

```
mysql> CREATE DATABASE transactions;
Query OK, 1 row affected (0.00 sec)

mysql> USE transactions;
Database changed
mysql>

mysql> CREATE TABLE transac
    -> (
    ->    id INT UNSIGNED PRIMARY KEY,
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql>
```

Imaginons qu'en étudiant les requêtes des utilisateurs sur cette table sur une période probante, nous découvrons que la majorité utilise des plages de date. Dans ce cas il serait utile de partitionner la table par la valeur de la colonne **jour**. Créer donc la table partitionnée appelée **transac_part** :

```
CREATE TABLE transac_part
(
id INT UNSIGNED NOT NULL,
montant INT UNSIGNED NOT NULL,
jour DATE NOT NULL,
codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
) PARTITION BY RANGE(YEAR(jour))
(
PARTITION p1 VALUES LESS THAN(1997),
PARTITION p2 VALUES LESS THAN(1998),
PARTITION p3 VALUES LESS THAN(1999),
PARTITION p4 VALUES LESS THAN(2000),
PARTITION p5 VALUES LESS THAN(2001),
PARTITION p6 VALUES LESS THAN(2002),
PARTITION p7 VALUES LESS THAN(2003),
PARTITION p8 VALUES LESS THAN(2004),
PARTITION p9 VALUES LESS THAN(2005),
PARTITION p10 VALUES LESS THAN(2006),
PARTITION p11 VALUES LESS THAN MAXVALUE
);
```

```
mysql> CREATE TABLE transac_part
    -> (
    ->    id INT UNSIGNED NOT NULL,
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
    -> ) PARTITION BY RANGE(YEAR(jour))
    -> (
```

```
    ->    PARTITION p1 VALUES LESS THAN(1997),
    ->    PARTITION p2 VALUES LESS THAN(1998),
    ->    PARTITION p3 VALUES LESS THAN(1999),
    ->    PARTITION p4 VALUES LESS THAN(2000),
    ->    PARTITION p5 VALUES LESS THAN(2001),
    ->    PARTITION p6 VALUES LESS THAN(2002),
    ->    PARTITION p7 VALUES LESS THAN(2003),
    ->    PARTITION p8 VALUES LESS THAN(2004),
    ->    PARTITION p9 VALUES LESS THAN(2005),
    ->    PARTITION p10 VALUES LESS THAN(2006),
    ->    PARTITION p11 VALUES LESS THAN MAXVALUE
    -> );
Query OK, 0 rows affected (0.13 sec)

mysql>
```

L'ordre de définition des partitions est important. Les informations liées aux partitions sont stockées dans la table **information_schema.partitions**.

Créez ensuite une procédure pour injecter des données dans la table **transac** :

```
DELIMITER //
CREATE PROCEDURE remplir_transaction(nbTransacs INT)
BEGIN
DECLARE i INT DEFAULT 1;
DECLARE nbAlea DOUBLE;
DECLARE _jour DATE;
DECLARE _montant INT UNSIGNED;
DECLARE _codePays TINYINT UNSIGNED;

WHILE i <= nbTransacs DO
```

```
SET nbAlea = RAND();
SET _jour = ADDDATE('1996-01-01', FLOOR(nbAlea * 4015));
SET _montant = FLOOR(1 + (nbAlea * 9999));
SET nbAlea = RAND();
SET _codePays = FLOOR(1 + (nbAlea * 6));
INSERT INTO transac (id, montant, jour, codePays) VALUES (i, _montant, _jour, _codePays);
SET i = i + 1;
END WHILE;
END //

DELIMITER ;
```

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE remplir_transaction(nbTransacs INT)
    -> BEGIN
    ->    DECLARE i INT DEFAULT 1;
    ->    DECLARE nbAlea DOUBLE;
    ->    DECLARE _jour DATE;
    ->    DECLARE _montant INT UNSIGNED;
    ->    DECLARE _codePays TINYINT UNSIGNED;
    ->
    ->    WHILE i <= nbTransacs DO
    ->       SET nbAlea = RAND();
    ->       SET _jour = ADDDATE('1996-01-01', FLOOR(nbAlea * 4015));
    ->       SET _montant = FLOOR(1 + (nbAlea * 9999));
    ->       SET nbAlea = RAND();
    ->       SET _codePays = FLOOR(1 + (nbAlea * 6));
    ->       INSERT INTO transac (id, montant, jour, codePays) VALUES (i, _montant, _jour, _codePays);
    ->       SET i = i + 1;
    ->    END WHILE;
    -> END //
Query OK, 0 rows affected (0.05 sec)

mysql> DELIMITER ;
```

```
mysql>
```

Appelez ensuite la procédure et patientez 10 minutes. A l'issue de la période, contrôlez le nombre d'enregistrements créés en sachant qu'il faut au mois 150 000 pour pouvoir voir l'impact de partitionnement :

```
mysql> CALL remplir_transaction(200000);
^CCtrl-C -- sending "KILL QUERY 2" to server ...
Ctrl-C -- query aborted.
mysql> SELECT COUNT(*) FROM transac;
+----------+
| COUNT(*) |
+----------+
|   191939 |
+----------+
1 row in set (0.09 sec)

mysql>

Insérez maintenant les mêmes données dans la table **transac_part** :

<code>
mysql> INSERT INTO transac_part SELECT * FROM transac;
Query OK, 191939 rows affected (3.63 sec)
Records: 191939  Duplicates: 0  Warnings: 0

mysql> SELECT COUNT(*) FROM transac_part;
+----------+
| COUNT(*) |
+----------+
|   191939 |
+----------+
1 row in set (0.14 sec)
```

```
mysql>
```

Pour voir l'impact du partitionnement sur la performance, saisissez les requêtes suivantes :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE jour BETWEEN '1998-01-01' AND '1998-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     39970960 |    2276.8989 |
+--------------+--------------+
1 row in set (0.17 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE jour BETWEEN '1998-01-01' AND '1998-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     39970960 |    2276.8989 |
+--------------+--------------+
1 row in set (0.03 sec)
```

Notez que la requête sur la table partitionnée retourne un résultat **5,67** fois plus vite que sur la table d'origine. Ceci est du au fait que le serveur exclue les partitions qui ne contiennent pas les données recherchées. Ceci peut être constaté en utilisant la clause EXPLAIN :

```
mysql> EXPLAIN PARTITIONS SELECT SUM(montant), AVG(montant) FROM transac_part WHERE jour BETWEEN '1998-01-01' AND
'1998-12-31'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: transac_part
   partitions: p3
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
```

```
          ref: NULL
         rows: 18035
        Extra: Using where
1 row in set (0.01 sec)

mysql>
```

Notez que la valeur de la colonne **partitions** est **p3**.

Evidemment, plus qu'il y a de partitions à consulter, moins important est le gains en performance :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE jour BETWEEN '1996-01-01' AND '2000-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    198031481 |    2268.9216 |
+--------------+--------------+
1 row in set (0.18 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE jour BETWEEN '1996-01-01' AND '2000-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    198031481 |    2268.9216 |
+--------------+--------------+
1 row in set (0.16 sec)

mysql>
```

Le système de partitionnement peut aussi s'avérer un handicap au niveau de la performance si les requêtes portent sur l'ensemble des données :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    959746672 |    5000.2692 |
+--------------+--------------+
1 row in set (0.16 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    959746672 |    5000.2692 |
+--------------+--------------+
1 row in set (0.20 sec)

mysql>
```

Dernièrement, l'effacement de données en grande quantité est plus rapide quand on drop une partition :

```
mysql> DELETE FROM transac_part WHERE jour BETWEEN '1996-01-01' AND '1996-12-31';
Query OK, 17680 rows affected (0.18 sec)

mysql> ALTER TABLE transac_part DROP PARTITION p1;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

LAB #10 - Partitionnement par Listes

Imaginons maintenant que nous avons découvert que les requêtes des utilisateurs portent souvent sur continent géographique. Dans ce cas, il serait

utile de faire un partitionnement par liste. Le champs crée par l'instruction suivante :

```
codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
```

associe une valeur entière à chaque valeur du champ de type ENUM. Nous allons utiliser ce fait pour créer la table partitionnée.

Commencez par nettoyer ce que vous avez déjà fait :

```
mysql> TRUNCATE TABLE transac;
Query OK, 0 rows affected (0.17 sec)

mysql> DROP TABLE transac_part;
Query OK, 0 rows affected (0.09 sec)

mysql>
```

Créez maintenant la table transac_part :

```
mysql> CREATE TABLE transac_part
    -> (
    ->    id INT UNSIGNED NOT NULL,
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays TINYINT UNSIGNED NOT NULL
    -> ) PARTITION BY LIST(codePays)
    -> (
    ->    PARTITION pEurope VALUES IN (1, 2, 3),
    ->    PARTITION pAmeriqueNord VALUES IN (4, 5),
    ->    PARTITION pAsie VALUES IN (6)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Appelez la procédure remplir_transaction et copiez la valeurs dans transac_part :

```
mysql> CALL remplir_transaction(10000000);
^CCtrl-C -- sending "KILL QUERY 2" to server ...
Ctrl-C -- query aborted.
mysql> INSERT INTO transac_part SELECT * FROM transac;
Query OK, 52193 rows affected (0.57 sec)
Records: 52193  Duplicates: 0  Warnings: 0

mysql>
```

Comparez le gains de performance en utilisant la table partitionnée :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE codePays IN ('FR','BE','UK');
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    130838834 |    5018.9433 |
+--------------+--------------+
1 row in set (0.07 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE codePays IN (1, 2, 3);
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    130838834 |    5018.9433 |
+--------------+--------------+
1 row in set (0.04 sec)

mysql>
```

Notez que la requête sur la table partitionnée retourne un résultat **1,75** fois plus vite que sur la table d'origine.

Comparez le gains de performance en utilisant uniquement la partition numéro 6 :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE codePays = 'JP';
```

```
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     43098698 |    4973.3093 |
+--------------+--------------+
1 row in set (0.05 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE codePays = 6;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     43098698 |    4973.3093 |
+--------------+--------------+
1 row in set (0.01 sec)

mysql>
```

Notez que la requête sur la table partitionnée retourne un résultat **5** fois plus vite que sur la table d'origine. Ceci est du au fait qu'il y a moins d'enregistrements à parcourir.

LAB #11 - Partitionnement par Hash

Dans ce cas la partition à laquelle appartient un enregistrement est déterminée à partir de la valeur de retour d'une fonction définie par l'utilisateur.

Re-créez la table transac_part :

```
mysql> DROP TABLE transac_part;
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> CREATE TABLE transac_part
    -> (
    ->    id INT UNSIGNED NOT NULL,
```

```
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
    -> ) PARTITION BY HASH(YEAR(jour)) PARTITIONS 11;
Query OK, 0 rows affected (0.07 sec)

mysql>
mysql>
mysql> INSERT INTO transac_part SELECT * FROM transac;
Query OK, 52193 rows affected (0.85 sec)
Records: 52193  Duplicates: 0  Warnings: 0

mysql>
```

Notez qu'ici on indique le champs de référence **jour** et le nombre de partitions désirées.

Testez maintenant le gains de performance en utilisant les partitions :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE jour BETWEEN '1998-01-01' AND '1998-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     10867590 |    2271.6534 |
+--------------+--------------+
1 row in set (0.09 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE jour BETWEEN '1998-01-01' AND '1998-12-31';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     10867590 |    2271.6534 |
```

```
+--------------+--------------+
1 row in set (0.08 sec)

mysql>
```

Notez qu'ici le gains de performance est minime.

Le gains de performance est minime parce que toutes les partitions ont été scannées :

```
mysql> EXPLAIN PARTITIONS SELECT SUM(montant), AVG(montant) FROM transac_part WHERE jour BETWEEN '1998-01-01' AND
'1998-12-31'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: transac_part
   partitions: p0,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 52982
        Extra: Using where
1 row in set (0.00 sec)

mysql>
```

Le partitionnement par hachage n'est efficace que dans le cas ou le hash est généré à partir d'une colonne de type entier.

Pour prouver ce point, recréez la table transac_part avec des partitions basées sur la colonne **codePays** :

```
mysql> DROP TABLE transac_part;
Query OK, 0 rows affected (0.22 sec)

mysql>
mysql> CREATE TABLE transac_part
    -> (
    ->    id INT UNSIGNED NOT NULL,
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays TINYINT UNSIGNED NOT NULL
    -> ) PARTITION BY HASH(codePays) PARTITIONS 6;
Query OK, 0 rows affected (0.23 sec)

mysql>
mysql> INSERT INTO transac_part SELECT * FROM transac;
Query OK, 52193 rows affected (0.65 sec)
Records: 52193  Duplicates: 0  Warnings: 0

mysql>
```

Testez de nouveau le gain de performance en utilisant les partitions :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE codePays = 1;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|     43320506 |    5023.2498 |
+--------------+--------------+
1 row in set (0.04 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE codePays = 1;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
```

```
|     43320506 |    5023.2498 |
+--------------+--------------+
1 row in set (0.02 sec)

mysql>
```

Cette fois-ci le gain est plus important.

Partitionnement par Key

Dans ce cas la partition à laquelle appartient un enregistrement est déterminée à partir de la valeur de retour d'une fonction définie par le serveur MySQL.

Par exemple :

```
CREATE TABLE transac_part
(
   id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
   montant INT UNSIGNED NOT NULL,
   jour DATE NOT NULL,
   codePays ENUM('FR', 'BE', 'UK', 'US', 'CA', 'JP') NOT NULL
) PARTITION BY KEY() PARTITIONS 5;
```

Dans ce cas, la clef n'est pas spécifiée. Le hachage est généré à partir de tous les champs de la clef primaire. Dans notre cas c'est le champ **id**. Puisqu'il y a 5 partitions, si le nombre de données est de 100 000, le serveur mettra exactement 20 000 enregistrements dans chaque partition.

LAB #12 - Sous-partitionnement

Avec MySQL il est possible de créer des sous-partitions. Cependant il existe deux limitations :

- La partition de premier niveau doit être de type RANGE ou LIST,
- La partition de deuxième niveau doit être de type HASH ou KEY,
- Le nombre de partitions total ne peut pas dépasser **8192**.

Prenons le cas où les requêtes fréquentes portent sur les dates **et** sur la zone géographique.

Créez la table transac-part :

```
mysql> DROP TABLE transac_part;
Query OK, 0 rows affected (0.18 sec)

mysql>
mysql> CREATE TABLE transac_part
    -> (
    ->    id INT UNSIGNED NOT NULL,
    ->    montant INT UNSIGNED NOT NULL,
    ->    jour DATE NOT NULL,
    ->    codePays INT UNSIGNED NOT NULL
    -> )
    -> PARTITION BY RANGE(YEAR(jour))
    -> SUBPARTITION BY HASH(codePays) SUBPARTITIONS 6
    -> (
    ->    PARTITION p1 VALUES LESS THAN(1997),
    ->    PARTITION p2 VALUES LESS THAN(1998),
    ->    PARTITION p3 VALUES LESS THAN(1999),
    ->    PARTITION p4 VALUES LESS THAN(2000),
    ->    PARTITION p5 VALUES LESS THAN(2001),
    ->    PARTITION p6 VALUES LESS THAN(2002),
    ->    PARTITION p7 VALUES LESS THAN(2003),
    ->    PARTITION p8 VALUES LESS THAN(2004),
```

```
    ->    PARTITION p9 VALUES LESS THAN(2005),
    ->    PARTITION p10 VALUES LESS THAN(2006),
    ->    PARTITION p11 VALUES LESS THAN MAXVALUE
    -> );
Query OK, 0 rows affected (1.37 sec)

mysql>
mysql> INSERT INTO transac_part SELECT * FROM transac;
```

Testez maintenant le gain de performance en utilisant les partitions :

```
mysql> SELECT SUM(montant), AVG(montant) FROM transac WHERE codePays=2 AND jour BETWEEN '2000-01-01' AND
'2000-12-01';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|      2928528 |    4073.0570 |
+--------------+--------------+
1 row in set (0.05 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part WHERE codePays=2 AND jour BETWEEN '2000-01-01' AND
'2000-12-01';
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|      2928528 |    4073.0570 |
+--------------+--------------+
1 row in set (0.01 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    261035779 |    5001.3561 |
```

```
+--------------+--------------+
1 row in set (0.07 sec)

mysql> SELECT SUM(montant), AVG(montant) FROM transac_part;
+--------------+--------------+
| SUM(montant) | AVG(montant) |
+--------------+--------------+
|    261035779 |    5001.3561 |
+--------------+--------------+
1 row in set (0.09 sec)

mysql>
```

Notez la dégradation des performance avec les partitions dans le cas d'une requête sans clause WHERE :

```
mysql> EXPLAIN PARTITIONS SELECT SUM(montant), AVG(montant) FROM transac_part WHERE codePays=2 AND jour BETWEEN
'2000-01-01' AND '2000-12-01'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: transac_part
   partitions: p5_p5sp2
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 893
        Extra: Using where
1 row in set (0.01 sec)

mysql> EXPLAIN PARTITIONS SELECT SUM(montant), AVG(montant) FROM transac_part\G
*************************** 1. row ***************************
           id: 1
```

```
  select_type: SIMPLE
        table: transac_part
   partitions:
p1_p1sp0,p1_p1sp1,p1_p1sp2,p1_p1sp3,p1_p1sp4,p1_p1sp5,p2_p2sp0,p2_p2sp1,p2_p2sp2,p2_p2sp3,p2_p2sp4,p2_p2sp5,p3_p3
sp0,p3_p3sp1,p3_p3sp2,p3_p3sp3,p3_p3sp4,p3_p3sp5,p4_p4sp0,p4_p4sp1,p4_p4sp2,p4_p4sp3,p4_p4sp4,p4_p4sp5,p5_p5sp0,p
5_p5sp1,p5_p5sp2,p5_p5sp3,p5_p5sp4,p5_p5sp5,p6_p6sp0,p6_p6sp1,p6_p6sp2,p6_p6sp3,p6_p6sp4,p6_p6sp5,p7_p7sp0,p7_p7s
p1,p7_p7sp2,p7_p7sp3,p7_p7sp4,p7_p7sp5,p8_p8sp0,p8_p8sp1,p8_p8sp2,p8_p8sp3,p8_p8sp4,p8_p8sp5,p9_p9sp0,p9_p9sp1,p9
_p9sp2,p9_p9sp3,p9_p9sp4,p9_p9sp5,p10_p10sp0,p10_p10sp1,p10_p10sp2,p10_p10sp3,p10_p10sp4,p10_p10sp5,p11_p11sp0,p1
1_p11sp1,p11_p11sp2,p11_p11sp3,p11_p11sp4,p11_p11sp5
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 54365
        Extra:
1 row in set (0.01 sec)

mysql>
```

LAB #13 - Partitionnement Vertical

Dans ce cas les partitions sont des tables différentes. Prenons l'exemple d'une base de données contenant des photos :

```
mysql> CREATE TABLE produit
    -> (
    ->    id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    ->    libelle VARCHAR(50) NOT NULL,
    ->    prix DECIMAL(10,2) NOT NULL,
    ->    photo BLOB NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)
```

Séparez maintenant les photos des autres données :

```
mysql> CREATE TABLE produit2
    -> (
    ->    id INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,
    ->    libelle VARCHAR(50) NOT NULL,
    ->    prix DECIMAL(10,2) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> CREATE TABLE photo_produit
    -> (
    ->    idProd INT UNIQUE NOT NULL,
    ->    photo BLOB NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Utilisez la procédure suivante pour injecter les données :

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE remplir_produit(nbProds INT)
    -> BEGIN
    ->    DECLARE i INT DEFAULT 1;
    ->    DECLARE _prix DECIMAL(10,2);
    ->    DECLARE _photo BLOB DEFAULT REPEAT('A', 20480);
    ->
    ->    WHILE i <= nbProds DO
    ->       SET _prix = CAST((RAND() * 1000) AS DECIMAL(10,2));
    ->       INSERT INTO produit VALUES (i, CONCAT('Produit_', i), _prix, _photo);
    ->       INSERT INTO produit2 VALUES (i, CONCAT('Produit_', i), _prix);
```

```
    ->       INSERT INTO photo_produit VALUES (i, _photo);
    ->       SET i = i + 1;
    ->    END WHILE;
    -> END //
Query OK, 0 rows affected (0.10 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL remplir_produit(10000);
^CCtrl-C -- sending "KILL QUERY 3" to server ...
Ctrl-C -- query aborted.
ERROR 1317 (70100): Query execution was interrupted
mysql> SELECT COUNT(*) FROM produit;
+----------+
| COUNT(*) |
+----------+
|     1664 |
+----------+
1 row in set (0.02 sec)

mysql> SELECT COUNT(*) FROM produit2;
+----------+
| COUNT(*) |
+----------+
|     1664 |
+----------+
1 row in set (0.01 sec)

mysql> SELECT COUNT(*) FROM photo_produit;
+----------+
| COUNT(*) |
+----------+
|     1663 |
+----------+
```

```
1 row in set (0.01 sec)

mysql>
```

Testez maintenant le gain de performance en utilisant les partitions :

```
mysql> SELECT id, libelle, prix FROM produit;
mysql> SELECT id, libelle, prix FROM produit2;
```

<html>

</html>
