

Version : **2022.01**

Dernière mise-à-jour : 2022/11/10 07:19

SER805 - Administration et Maintenance

Contenu du Module

- **SER805 - Administration et Maintenance**
 - Contenu du Module
 - Administration
 - Mysqladmin
 - Méta-données
 - La commande DESCRIBE
 - La commande EXPLAIN
 - La commande BENCHMARK
 - La Commande SHOW
 - La Commande SET
 - La Commande KILL
 - La Commande FLUSH
 - La base INFORMATION_SCHEMA
 - Outils d'Administration Graphiques
 - Maintenance
 - Instructions SQL
 - CHECK TABLE
 - REPAIR TABLE
 - OPTIMIZE TABLE
 - ANALYZE TABLE
 - myisamchk
 - mysqlcheck

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

Cette commande crée une nouvelle base de données :

```
[root@centos8 ~]# mysqladmin -u root -p create newdb  
Enter password: fenestros
```

drop

Cette commande supprime une base de données:

```
[root@centos8 ~]# 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
```

password

Cette commande modifie le mot de passe de l'utilisateur donné par -u :

```
[root@centos8 ~]# mysqladmin -u root -p password "fenestros1"
Enter password: fenestros
```

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

```
[root@centos8 ~]# 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 MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.3.28-MariaDB-log MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [mysql]>
```



Important - 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 MariaDB 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 MariaDB
- 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**:

```
MariaDB [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
```

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

```
MariaDB [ligue1]> 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 (0.001 sec)

MariaDB [ligue1]>

La commande EXPLAIN

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

MariaDB [ligue1]> EXPLAIN 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 (0.000 sec)

MariaDB [ligue1]>

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

```
MariaDB [ligue1]> 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 (0.000 sec)
```

```
MariaDB [ligue1]> 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 (0.000 sec)
```

```
MariaDB [ligue1]>
```

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:

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

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

```
MariaDB [ligue1]> select BENCHMARK(1000000000, "select * FROM equipe");
+-----+
```

```
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----+
|                                0 |
+-----+
1 row in set (5.484 sec)

MariaDB [ligue1]> select BENCHMARK(1000000000, "select * FROM equipe");
+-----+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----+
|                                0 |
+-----+
1 row in set (5.473 sec)

MariaDB [ligue1]> select BENCHMARK(1000000000, "select * FROM equipe");
+-----+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----+
|                                0 |
+-----+
1 row in set (5.362 sec)

MariaDB [ligue1]>
```

La Commande SHOW

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

SHOW DATABASES

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW databases;
```

```
+-----+
| Database          |
+-----+
| Nombres           |
| information_schema |
| ligue1            |
| mysql             |
| performance_schema |
+-----+
5 rows in set (0.000 sec)
```

```
MariaDB [ligue1]>
```

SHOW TABLES

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW tables;
```

```
+-----+
| Tables_in_ligue1 |
+-----+
| V_EQUIPE         |
| equipe           |
```

```
| rencontre |
+-----+
3 rows in set (0.000 sec)
```

MariaDB [ligue1]>

SHOW OPEN TABLES

Cette commande prend la forme suivante :

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

Par exemple :

MariaDB [ligue1]> SHOW OPEN TABLES;

Database	Table	In_use	Name_locked
ligue1	equipe	0	0
mysql	servers	0	0
mysql	procs_priv	0	0
mysql	table_stats	0	0
mysql	index_stats	0	0
mysql	plugin	0	0
mysql	time_zone_transition	0	0
mysql	time_zone_leap_second	0	0
mysql	time_zone	0	0
mysql	help_category	0	0
mysql	host	0	0
mysql	general_log	0	0
mysql	user	0	0
ligue1	V_EQUIPE	0	0

mysql	roles_mapping	0	0
mysql	time_zone_transition_type	0	0
mysql	proxies_priv	0	0
mysql	event	0	0
mysql	columns_priv	0	0
mysql	proc	0	0
mysql	help_keyword	0	0
mysql	func	0	0
mysql	time_zone_name	0	0
mysql	db	0	0
mysql	slow_log	0	0
mysql	column_stats	0	0
mysql	help_topic	0	0
mysql	help_relation	0	0
mysql	tables_priv	0	0
mysql	gtid_slave_pos	0	0
mysql	innodb_table_stats	0	0
mysql	innodb_index_stats	0	0
ligue1	rencontre	0	0
mysql	transaction_registry	0	0

+-----+-----+-----+-----+

34 rows in set (0.000 sec)

MariaDB [ligue1]>



Important : 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 l'information, voir



<https://dev.mysql.com/doc/refman/5.6/en/show-open-tables.html>.

SHOW COLUMNS

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> 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	
but	int(11)	NO		0	
entraîneur	varchar(100)	YES		inconnu	
tel_entraîneur	varchar(20)	YES		inconnu	

8 rows in set (0.001 sec)

```
MariaDB [ligue1]>
```

ou

```
MariaDB [ligue1]> 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 (0.001 sec)

MariaDB [ligue1]>

```

SHOW INDEX

Cette commande prend la forme suivante :

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



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+
| V_EQUIPE | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
NULL | NULL | VIEW | NULL | NULL | NULL | NULL | NULL | NULL |
| equipe | MyISAM | 10 | Dynamic | 4 | 80 | 320 | 281474976710655 |
2048 | 0 | 5 | 2022-11-08 12:02:12 | 2022-11-08 12:02:54 | NULL | latin1_swedish_ci |
NULL | | 1125899906841600 | N | |
| rencontre | MyISAM | 10 | Dynamic | 6 | 32 | 192 | 281474976710655 |
2048 | 0 | NULL | 2022-11-08 04:59:52 | 2022-11-08 12:01:12 | NULL | latin1_swedish_ci |
NULL | | 288230376151710720 | N | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [ligue1]>

```

SHOW STATUS

Cette commande prend la forme suivante :

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

Par exemple:

```

MariaDB [ligue1]> SHOW STATUS \G
...
***** 531. row *****
Variable_name: wsrep_provider_name
Value:

```

```
***** 532. row *****
Variable_name: wsrep_provider_vendor
Value:
***** 533. row *****
Variable_name: wsrep_provider_version
Value:
***** 534. row *****
Variable_name: wsrep_ready
Value: OFF
***** 535. row *****
Variable_name: wsrep_rollbacker_thread_count
Value: 0
***** 536. row *****
Variable_name: wsrep_thread_count
Value: 0
536 rows in set (0.002 sec)

MariaDB [ligue1]>
```

SHOW VARIABLES

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW VARIABLES \G
...
***** 638. row *****
Variable_name: wsrep_sst_donor
Value:
```

```
***** 639. row *****
Variable_name: wsrep_sst_donor_rejects_queries
Value: OFF
***** 640. row *****
Variable_name: wsrep_sst_method
Value: rsync
***** 641. row *****
Variable_name: wsrep_sst_receive_address
Value: AUTO
***** 642. row *****
Variable_name: wsrep_start_position
Value: 00000000-0000-0000-0000-000000000000:-1
***** 643. row *****
Variable_name: wsrep_sync_wait
Value: 0
643 rows in set (0.001 sec)

MariaDB [ligue1]>
```

SHOW PROCESSLIST

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW PROCESSLIST;
```

```
+---+-----+-----+-----+-----+-----+-----+-----+
---+
| Id | User          | Host          | db          | Command | Time  | State          | Info          |
Progress |
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
---+
| 2 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |
0.000 |
| 1 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |
0.000 |
| 3 | system user |      | NULL | Daemon | NULL | InnoDB purge coordinator | NULL |
0.000 |
| 4 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |
0.000 |
| 5 | system user |      | NULL | Daemon | NULL | InnoDB shutdown handler | NULL |
0.000 |
| 10 | user2      | localhost | NULL | Sleep | 12108 |      | NULL |
0.000 |
| 14 | root      | localhost | ligue1 | Query | 0 | Init | SHOW PROCESSLIST |
0.000 |

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
---+
7 rows in set (0.000 sec)

```

MariaDB [ligue1]> SHOW FULL PROCESSLIST;

```

+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| Id | User      | Host      | db      | Command | Time | State | Info |
Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| 2 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |
0.000 |
| 1 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |
0.000 |
| 3 | system user |      | NULL | Daemon | NULL | InnoDB purge coordinator | NULL |
0.000 |
| 4 | system user |      | NULL | Daemon | NULL | InnoDB purge worker | NULL |

```

```

0.000 |
| 5 | system user |          | NULL | Daemon | NULL | InnoDB shutdown handler | NULL          |
0.000 |
| 10 | user2          | localhost | NULL | Sleep  | 12151 |          | NULL          |
0.000 |
| 14 | root           | localhost | ligue1 | Query  |      0 | Init          | SHOW FULL PROCESSLIST |
0.000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
7 rows in set (0.000 sec)

MariaDB [ligue1]>

```

SHOW GRANTS

Cette commande prend la forme suivante :

```
SHOW GRANTS FOR utilisateur;
```

Par exemple:

```

MariaDB [ligue1]> SHOW GRANTS;
+-----+
-----+
| Grants for root@localhost
|
+-----+
-----+
| GRANT ALL PRIVILEGES ON *.* TO `root`@`localhost` IDENTIFIED BY PASSWORD
'|F3E048E28ED63F281CF8A403F96F5D283C8700E6' WITH GRANT OPTION |
| GRANT PROXY ON ''@%' TO 'root'@'localhost' WITH GRANT OPTION
|

```

```
+-----+
-----+
2 rows in set (0.000 sec)

MariaDB [ligue1]>
```

SHOW CREATE TABLE

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> 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,
```

```
`entraîneur` varchar(100) DEFAULT 'inconnu',  
`tel_entraîneur` varchar(20) DEFAULT 'inconnu',  
PRIMARY KEY (`id_equipe`)  
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1 |
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.000 sec)
```

MariaDB [ligue1]>

ou

```
MariaDB [ligue1]> 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,  
  `entraîneur` varchar(100) DEFAULT 'inconnu',  
  `tel_entraîneur` varchar(20) DEFAULT 'inconnu',  
  PRIMARY KEY (`id_equipe`)  
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1  
1 row in set (0.000 sec)
```

MariaDB [ligue1]>





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

La Commande SET

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

Par exemple la saisie de la commande suivante démontre que la valeur de max_connections est de **151**:

```
MariaDB [ligue1]> SHOW VARIABLES LIKE 'max_connections';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 151   |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [ligue1]>
```

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

```
MariaDB [ligue1]> SET GLOBAL max_connections = 200;
Query OK, 0 rows affected (0.000 sec)

MariaDB [ligue1]> SHOW VARIABLES LIKE 'max_connections';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 200   |
+-----+-----+
1 row in set (0.000 sec)
```

```
MariaDB [ligue1]>
```



Important - 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.

```
> KILL idThread; [Entrée]
```

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

```
MariaDB [ligue1]> GRANT USAGE ON * TO fenestros@localhost IDENTIFIED BY 'fenestros';  
Query OK, 0 rows affected (0.000 sec)
```

```
MariaDB [ligue1]>
```

Accordez les privilèges de connexion à l'utilisateur fenestros:

```
MariaDB [ligue1]> SHOW GRANTS for fenestros@localhost;  
+-----+  
--+  
| Grants for fenestros@localhost  
|  
+-----+  
--+  
| GRANT USAGE ON *.* TO `fenestros`@`localhost` IDENTIFIED BY PASSWORD  
`*00269BA49BEC800F9CCF34C20C1FD83E0236B89A` |
```

```
+-----+  
--+  
1 row in set (0.000 sec)
```

MariaDB [ligue1]>

Ouvrez une autre session sur la VM et connectez-vous à votre serveur MariaDB avec le compte de **fenestros** sans vous déconnecter en tant que root et saisissez la commande pour vérifiez que tout va bien:

```
[root@centos8 ~]# mysql -u fenestros -p  
Enter password: fenestros  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 15  
Server version: 10.3.28-MariaDB-log MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database          |  
+-----+  
| information_schema |  
+-----+  
1 row in set (0.000 sec)  
  
MariaDB [(none)]>
```



Important - Notez que le mot de passe ne sera **pas** visible.

Revenez à la première connexion à votre VM et saisissez la commande suivante:

```
MariaDB [(none)]> SHOW PROCESSLIST;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
| Id | User          | Host          | db  | Command | Time | State                | Info                | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
| 2 | system user |              | NULL | Daemon  | NULL | InnoDB purge worker | NULL                | 0.000   |
| 1 | system user |              | NULL | Daemon  | NULL | InnoDB purge worker | NULL                | 0.000   |
| 3 | system user |              | NULL | Daemon  | NULL | InnoDB purge coordinator | NULL                | 0.000   |
| 4 | system user |              | NULL | Daemon  | NULL | InnoDB purge worker  | NULL                | 0.000   |
| 5 | system user |              | NULL | Daemon  | NULL | InnoDB shutdown handler | NULL                | 0.000   |
| 10 | user2      | localhost    | NULL | Sleep   | 12771 |                      | NULL                | 0.000   |
| 15 | fenestros  | localhost    | NULL | Sleep   | 164  |                      | NULL                | 0.000   |
| 16 | root       | localhost    | NULL | Query   | 0    | Init                 | SHOW PROCESSLIST   | 0.000   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
8 rows in set (0.000 sec)
```

```
MariaDB [(none)]>
```



Important - 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 :

```
MariaDB [(none)]> KILL 15;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> SHOW PROCESSLIST;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
+ Id | User      | Host      | db  | Command | Time  | State              | Info              | Progress
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
+ 2 | system user |          | NULL | Daemon  | NULL  | InnoDB purge worker | NULL              | 0.000
+
+ 1 | system user |          | NULL | Daemon  | NULL  | InnoDB purge worker | NULL              | 0.000
+
+ 3 | system user |          | NULL | Daemon  | NULL  | InnoDB purge coordinator | NULL              | 0.000
+
+ 4 | system user |          | NULL | Daemon  | NULL  | InnoDB purge worker  | NULL              | 0.000
+
+ 5 | system user |          | NULL | Daemon  | NULL  | InnoDB shutdown handler | NULL              | 0.000
+
+ 10 | user2      | localhost | NULL | Sleep   | 12811 |                    | NULL              | 0.000
+
+ 16 | root       | localhost | NULL | Query   | 0     | Init                | SHOW PROCESSLIST | 0.000
+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
7 rows in set (0.000 sec)

MariaDB [(none)]>
```



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

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

```
MariaDB [(none)]> SHOW DATABASES;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 17
Current database: *** NONE ***
```

```
+-----+
| Database |
+-----+
| information_schema |
+-----+
1 row in set (0.004 sec)
```

```
MariaDB [(none)]>
```



Important - Notez que le client se reconnecte au serveur!

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

```
MariaDB [(none)]> SHOW PROCESSLIST;
```

```
+---+-----+-----+-----+-----+-----+-----+-----+-----+
+
| Id | User      | Host      | db  | Command | Time | State              | Info              | Progress |
+---+-----+-----+-----+-----+-----+-----+-----+-----+
+
| 2  | system user |          | NULL | Daemon  | NULL | InnoDB purge worker | NULL              | 0.000    |
|
| 1  | system user |          | NULL | Daemon  | NULL | InnoDB purge worker | NULL              | 0.000    |
```

```

|
| 3 | system user |          | NULL | Daemon | NULL | InnoDB purge coordinator | NULL          | 0.000
|
| 4 | system user |          | NULL | Daemon | NULL | InnoDB purge worker      | NULL          | 0.000
|
| 5 | system user |          | NULL | Daemon | NULL | InnoDB shutdown handler  | NULL          | 0.000
|
| 10 | user2      | localhost | NULL | Sleep  | 12912 |                          | NULL          | 0.000
|
| 16 | root       | localhost | NULL | Query  | 0     | Init                      | SHOW PROCESSLIST | 0.000
|
| 17 | fenestros  | localhost | NULL | Sleep  | 28    |                          | NULL          | 0.000
|
+-----+-----+-----+-----+-----+-----+-----+-----+
+
8 rows in set (0.000 sec)

MariaDB [(none)]>

```

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

```

MariaDB [(none)]> SHOW PROCESSLIST;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User      | Host      | db  | Command | Time | State | Info          | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 17 | fenestros | localhost | NULL | Query   | 0    | Init  | SHOW PROCESSLIST | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]>

```



Important - 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 MariaDB.

FLUSH HOSTS

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

```
MariaDB [(none)]> FLUSH HOSTS; [Entrée]
```

FLUSH LOGS

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

```
MariaDB [(none)]> FLUSH LOGS; [Entrée]
```

FLUSH PRIVILEGES

Cette commande recharge la table de privileges:

```
MariaDB [(none)]> FLUSH PRIVILEGES; [Entrée]
```

FLUSH TABLES

Cette commande vide les caches des tables et les ferme:

```
MariaDB [(none)]> FLUSH TABLES; [Entrée]
```

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

```
MariaDB [(none)]> FLUSH [TABLE|TABLES] nomtable [,nomtable]; [Entrée]
```

La base INFORMATION_SCHEMA

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

```
MariaDB [(none)]> 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

```
MariaDB [information_schema]> SHOW TABLES;
```

```
+-----+
| Tables_in_information_schema |
+-----+
| ALL_PLUGINS                  |
| APPLICABLE_ROLES             |
| CHARACTER_SETS               |
| CHECK_CONSTRAINTS            |
| COLLATIONS                   |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                     |
| COLUMN_PRIVILEGES             |
| ENABLED_ROLES                 |
| ENGINES                       |
| EVENTS                       |
| FILES                         |
| GLOBAL_STATUS                 |
| GLOBAL_VARIABLES              |
| KEY_CACHES                    |
```

KEY_COLUMN_USAGE	
PARAMETERS	
PARTITIONS	
PLUGINS	
PROCESSLIST	
PROFILING	
REFERENTIAL_CONSTRAINTS	
ROUTINES	
SCHEMATA	
SCHEMA_PRIVILEGES	
SESSION_STATUS	
SESSION_VARIABLES	
STATISTICS	
SYSTEM_VARIABLES	
TABLES	
TABLESPACES	
TABLE_CONSTRAINTS	
TABLE_PRIVILEGES	
TRIGGERS	
USER_PRIVILEGES	
VIEWS	
GEOMETRY_COLUMNS	
SPATIAL_REF_SYS	
CLIENT_STATISTICS	
INDEX_STATISTICS	
INNODB_SYS_DATAFILES	
USER_STATISTICS	
INNODB_SYS_TABLESTATS	
INNODB_LOCKS	
INNODB_MUTEXES	
INNODB_CMPMEM	
INNODB_CMP_PER_INDEX	
INNODB_CMP	
INNODB_FT_DELETED	

```
| INNODB_CMP_RESET |
| INNODB_LOCK_WAITS |
| TABLE_STATISTICS |
| INNODB_TABLESPACES_ENCRYPTION |
| INNODB_BUFFER_PAGE_LRU |
| INNODB_SYS_FIELDS |
| INNODB_CMPMEM_RESET |
| INNODB_SYS_COLUMNS |
| INNODB_FT_INDEX_TABLE |
| INNODB_CMP_PER_INDEX_RESET |
| user_variables |
| INNODB_FT_INDEX_CACHE |
| INNODB_SYS_FOREIGN_COLS |
| INNODB_FT_BEING_DELETED |
| INNODB_BUFFER_POOL_STATS |
| INNODB_TRX |
| INNODB_SYS_FOREIGN |
| INNODB_SYS_TABLES |
| INNODB_FT_DEFAULT_STOPWORD |
| INNODB_FT_CONFIG |
| INNODB_BUFFER_PAGE |
| INNODB_SYS_TABLESPACES |
| INNODB_METRICS |
| INNODB_SYS_INDEXES |
| INNODB_SYS_VIRTUAL |
| INNODB_TABLESPACES_SCRUBBING |
| INNODB_SYS_SEMAPHORE_WAITS |
+-----+
76 rows in set (0.001 sec)
```

```
MariaDB [information_schema]>exit
Bye
```

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.

Outils d'Administration Graphiques

Téléchargez **mysql-workbench-community-8.0.20-1.el8.x86_64.rpm** :

```
[root@centos8 ~]# wget  
https://downloads.mysql.com/archives/get/p/8/file/mysql-workbench-community-8.0.20-1.el8.x86_64.rpm
```

Installez maintenant le paquet :

```
[root@centos8 ~]# yum localinstall mysql-workbench-community-8.0.20-1.el8.x86_64.rpm --nogpgcheck
```



A faire - Connectez-vous à votre VM via Guacamole. Testez MySQL Workbench.

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

Vérifiez les tables de votre base ligue1 ;

```
[root@centos8 ~]# mysql -u root -p
Enter password: fenestros1
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.3.28-MariaDB-log MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> 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 (0.000 sec)
```

MariaDB [(none)]>

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

```
MariaDB [(none)]> 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 (0.000 sec)
```

MariaDB [(none)]>

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 :

```
> 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

Option	Description
QUICK	Ne répare que l'index
EXTENDED	Gère 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 :

```
MariaDB [(none)]> 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 (0.001 sec)

MariaDB [(none)]>
```

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.

OPTIMIZE TABLE

La commande OPTIMIZE TABLE permet de défragmenter une table ou plusieurs tables. La commande OPTIMIZE TABLE fonctionne avec le moteur **MyISAM** et le moteur **InnoDB**.

Saisissez la commande suivante :

```
MariaDB [(none)]> 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 (0.001 sec)
```

```
MariaDB [(none)]>
```

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 :

```
MariaDB [(none)]> 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 (0.000 sec)
```

```
MariaDB [(none)]>
```





A faire - Pour plus d'information, voir
<http://dev.mysql.com/doc/refman/5.6/en/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 **arrêté**.

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

```
MariaDB [(none)]> exit
Bye
[root@centos8 ~]# systemctl stop mariadb
```

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

```
[root@centos8 ~]# 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

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

Saisissez la commande suivante :

```
[root@centos8 ~]# myisamchk -ri /var/lib/mysql/liguel/rencontre /var/lib/mysql/liguel/equipe
- recovering (with sort) MyISAM-table '/var/lib/mysql/liguel/rencontre'
Data records: 6
- Fixing index 1
-----

- recovering (with sort) MyISAM-table '/var/lib/mysql/liguel/equipe'
Data records: 4
- Fixing index 1
User time 0.00, System time 0.00
Maximum resident set size 4176, Integral resident set size 0
Non-physical pagefaults 257, Physical pagefaults 0, Swaps 0
Blocks in 0 out 32, Messages in 0 out 0, Signals 0
Voluntary context switches 0, Involuntary context switches 0
```

Saisissez la commande suivante :

```
[root@centos8 ~]# myisamchk -d /var/lib/mysql/liguel/rencontre /var/lib/mysql/liguel/equipe
```

```
MyISAM file:      /var/lib/mysql/liguel/rencontre
Record format:    Packed
Character set:     latin1_swedish_ci (8)
Data records:      6 Deleted blocks:      0
Recordlength:     98
```

table description:

Key	Start	Len	Index	Type
1	2	4	unique	long
	6	4		long
	10	5		binary

```
MyISAM file:      /var/lib/mysql/liguel/equipe
Record format:    Packed
Character set:     latin1_swedish_ci (8)
Data records:      4 Deleted blocks:      0
Recordlength:     269
```

table description:

Key	Start	Len	Index	Type
1	2	4	unique	long

Total of all 2 MyISAM-files:

```
Data records:      10 Deleted blocks:      0
```

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** :

```
[root@centos8 tmp]# systemctl start mariadb
```

Saisissez la commande suivante :

```
[root@centos8 ~]# mysqlcheck -u root -p --host=127.0.0.1 ligue1
Enter password: fenestros1
ligue1.equipe                OK
ligue1.rencontre             OK
```

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. C'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

