

Version: 2022/11/08 14:19

SER405 - 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

Cette commande crée une nouvelle base de données :

```
[root@centos7 ~]# mysqladmin -u root -p create newdb  
Enter password: fenestros
```

drop

Cette commande supprime une base de données:

```
[root@centos7 ~]# 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@centos7 ~]# 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@centos7 ~]# 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 MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, 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  
MariaDB [ligue1]>
```

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	
butts	int(11)	NO		0	
entraîneur	varchar(100)	YES		inconnu	
tel_entraîneur	varchar(20)	YES		inconnu	

```
8 rows in set (0.01 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	
butts	int(11)	NO		0	
entraîneur	varchar(100)	YES		inconnu	

```
| tel_entraineur | varchar(20) | YES | | inconnu | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 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 (0.01 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 (0.00 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: wget https://downloads.mysql.com/archives/get/p/8/file/mysql-workbench-community-6.3.10-1.el7.x86_64.rpm

```
> 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 (16.70 sec)

MariaDB [ligue1]> select BENCHMARK(1000000000, "select * FROM equipe");
+-----+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----+
|                                0 |
+-----+
1 row in set (16.91 sec)

MariaDB [ligue1]> select BENCHMARK(1000000000, "select * FROM equipe");
+-----+
| BENCHMARK(1000000000, "select * FROM equipe") |
+-----+
|                                0 |
+-----+
1 row in set (16.93 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          |
+-----+
| information_schema |
| CarnetAdresses    |
| Nombres           |
| ligue1            |
| mysql             |
| performance_schema |
| test              |
+-----+
7 rows in set (0.01 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 (0.01 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           |
```

```
| performance_schema | events_waits_summary_global_by_event_name | 0      | 0           |
```

performance_schema	setup_timers	0	0
performance_schema	events_waits_history_long	0	0
mysql	time_zone_name	0	0
mysql	time_zone	0	0
performance_schema	events_waits_summary_by_instance	0	0
mysql	db	0	0
performance_schema	setup_consumers	0	0
mysql	help_keyword	0	0
mysql	time_zone_transition_type	0	0
performance_schema	setup_instruments	0	0
mysql	user	0	0
mysql	slow_log	0	0
performance_schema	file_instances	0	0
performance_schema	cond_instances	0	0
mysql	plugin	0	0
mysql	proc	0	0
mysql	proxies_priv	0	0
mysql	help_topic	0	0
performance_schema	rwlock_instances	0	0
mysql	time_zone_transition	0	0
mysql	event	0	0
mysql	columns_priv	0	0
performance_schema	performance_timers	0	0
performance_schema	threads	0	0
mysql	general_log	0	0
performance_schema	events_waits_summary_by_thread_by_event_name	0	0
performance_schema	file_summary_by_event_name	0	0
mysql	time_zone_leap_second	0	0
performance_schema	file_summary_by_instance	0	0
liguel	rencontre	0	0
mysql	servers	0	0
mysql	host	0	0
performance_schema	events_waits_history	0	0
mysql	func	0	0

```
| mysql      | ndb_binlog_index      | 0 | 0 |
| performance_schema | mutex_instances      | 0 | 0 |
| ligue1     | V_EQUIPE              | 0 | 0 |
| mysql      | tables_priv           | 0 | 0 |
| mysql      | procs_priv            | 0 | 0 |
| performance_schema | events_waits_current | 0 | 0 |
| mysql      | help_category         | 0 | 0 |
| mysql      | help_relation         | 0 | 0 |
+-----+-----+-----+-----+
44 rows in set (0.01 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 d'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	
butts	int(11)	NO		0	
entraîneur	varchar(100)	YES		inconnu	
tel_entraîneur	varchar(20)	YES		inconnu	

```
8 rows in set (0.00 sec)
```

```
MariaDB [ligue1]>
```

ou

```
MariaDB [ligue1]> SHOW FULL COLUMNS FROM equipe;
```

Field	Type	Collation	Null	Key	Default	Extra	Privileges
id_equipe	int(11)	NULL	NO	PRI	NULL	auto_increment	
nom	varchar(50)	latin1_swedish_ci	NO		NULL		
stade	varchar(50)	latin1_swedish_ci	NO		NULL		
ville	varchar(30)	latin1_swedish_ci	NO		NULL		

```

| 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 (0.01 sec)

MariaDB [ligue1]>

```



A Faire - Pour plus d'information, voir

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

SHOW INDEX

Cette commande prend la forme suivante :

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

Par exemple;

```

MariaDB [ligue1]> SHOW INDEX FROM equipe;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+
| Table  | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed |
Null | Index_type | Comment | Index_comment |

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+
| equipe |          0 | PRIMARY |          1 | id_equipe | A          |          4 | NULL | NULL |
| BTREE  |          |          |          |           |           |           |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [ligue1]>

```



A faire - Pour plus d'information, voir <https://dev.mysql.com/doc/refman/5.6/en/show-index.html>.

SHOW TABLE STATUS

Cette commande prend la forme suivante :

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

Par exemple;

```
MariaDB [ligue1]> 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
------	--------	---------	------------	------	----------------	-------------	-----------------	--------------	-----------	----------------	-------------	-------------	------------	-----------

```

-----+-----+-----+
| V_EQUIPE | NULL | NULL | 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 | 2017-10-31 11:12:25 | 2017-10-31 11:13:16 | NULL | latin1_swedish_ci |
NULL |
| rencontre | MyISAM | 10 | Dynamic | 6 | 30 | 184 | 281474976710655 |
2048 | 0 | NULL | 2017-10-31 10:20:18 | 2017-10-31 11:08:20 | NULL | latin1_swedish_ci |
NULL |
+-----+-----+-----+
+-----+-----+-----+
-----+
3 rows in set (0.00 sec)

MariaDB [ligue1]>

```



A faire - Pour plus d'information, voir
<https://dev.mysql.com/doc/refman/5.6/en/show-table-status.html>.

SHOW STATUS

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW STATUS \G
...
```

```
***** 394. row *****
Variable_name: Ssl_used_session_cache_entries
Value: 0
***** 395. row *****
Variable_name: Ssl_verify_depth
Value: 18446744073709551615
***** 396. row *****
Variable_name: Ssl_verify_mode
Value: 5
***** 397. row *****
Variable_name: Ssl_version
Value: TLSv1.2
***** 398. row *****
Variable_name: Subquery_cache_hit
Value: 0
***** 399. row *****
Variable_name: Subquery_cache_miss
Value: 0
***** 400. row *****
Variable_name: Syncs
Value: 3
***** 401. row *****
Variable_name: Table_locks_immediate
Value: 60
***** 402. row *****
Variable_name: Table_locks_waited
Value: 0
***** 403. row *****
Variable_name: Tc_log_max_pages_used
Value: 0
***** 404. row *****
Variable_name: Tc_log_page_size
Value: 0
***** 405. row *****
```

```
Variable_name: Tc_log_page_waits
Value: 0
***** 406. row *****
Variable_name: Threadpool_idle_threads
Value: 0
***** 407. row *****
Variable_name: Threadpool_threads
Value: 0
***** 408. row *****
Variable_name: Threads_cached
Value: 0
***** 409. row *****
Variable_name: Threads_connected
Value: 1
***** 410. row *****
Variable_name: Threads_created
Value: 8
***** 411. row *****
Variable_name: Threads_running
Value: 1
***** 412. row *****
Variable_name: Uptime
Value: 11802
***** 413. row *****
Variable_name: Uptime_since_flush_status
Value: 11802
413 rows in set (0.01 sec)

MariaDB [ligue1]>
```



A faire - Pour plus d'information, voir

<http://dev.mysql.com/doc/refman/5.6/en/show-status.html>.

SHOW VARIABLES

Cette commande prend la forme suivante :

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

Par exemple:

```
MariaDB [ligue1]> SHOW VARIABLES \G
...
***** 418. row *****
Variable_name: timed_mutexes
Value: OFF
***** 419. row *****
Variable_name: timestamp
Value: 1509703244.822412
***** 420. row *****
Variable_name: tmp_table_size
Value: 16777216
***** 421. row *****
Variable_name: tmpdir
Value: /var/tmp
***** 422. row *****
Variable_name: transaction_alloc_block_size
Value: 8192
***** 423. row *****
Variable_name: transaction_prealloc_size
Value: 4096
***** 424. row *****
Variable_name: tx_isolation
Value: REPEATABLE-READ
***** 425. row *****
Variable_name: unique_checks
```

```
Value: ON
***** 426. row *****
Variable_name: updatable_views_with_limit
Value: YES
***** 427. row *****
Variable_name: userstat
Value: OFF
***** 428. row *****
Variable_name: version
Value: 5.5.56-MariaDB
***** 429. row *****
Variable_name: version_comment
Value: MariaDB Server
***** 430. row *****
Variable_name: version_compile_machine
Value: x86_64
***** 431. row *****
Variable_name: version_compile_os
Value: Linux
***** 432. row *****
Variable_name: wait_timeout
Value: 28800
***** 433. row *****
Variable_name: warning_count
Value: 0
433 rows in set (0.01 sec)
```

MariaDB [ligue1]>



A faire - Pour plus d'information, voir

<http://dev.mysql.com/doc/refman/5.6/en/show-variables.html>.

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 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8  | root | localhost | ligue1  | Query   | 0    | NULL  | SHOW PROCESSLIST    | 0.000    |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host      | db      | Command | Time | State | Info                | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8  | root | localhost | ligue1  | Query   | 0    | NULL  | SHOW FULL PROCESSLIST | 0.000    |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

```
MariaDB [ligue1]>
```



A faire - Pour plus d'information, voir
<http://dev.mysql.com/doc/refman/5.6/en/show-processlist.html>.

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 (0.00 sec)

MariaDB [ligue1]>
```



A faire - Pour plus d'information, voir
<http://dev.mysql.com/doc/refman/5.6/en/show-grants.html>.

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 (0.00 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 (0.00 sec)

MariaDB [ligue1]>
```



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



A faire - Pour plus d'information, voir
<http://dev.mysql.com/doc/refman/5.6/en/show-create-table.html>.

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 **100**:

```
MariaDB [ligue1]> SHOW VARIABLES LIKE 'max_connections';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 151   |
+-----+-----+
1 row in set (0.00 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 (0.00 sec)

MariaDB [ligue1]> SHOW VARIABLES LIKE 'max_connections';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 200   |
+-----+-----+
1 row in set (0.00 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 (0.00 sec)
```

```
MariaDB [ligue1]>
```

Accordez les privilèges de connexion à l'utilisateur fenestros:

```
MariaDB [ligue1]> GRANT USAGE ON * TO fenestros@localhost IDENTIFIED BY 'fenestros';  
Query OK, 0 rows affected (0.00 sec)
```

```
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 (0.00 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@centos7 ~]# mysql -u fenestros -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, 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 (0.00 sec)

MariaDB [(none)]>
```



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

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

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

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User      | Host      | db      | Command | Time | State | Info              | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8  | root      | localhost | ligue1  | Query   | 0    | NULL  | SHOW PROCESSLIST | 0.000    |
| 9  | fenestros | localhost | NULL    | Sleep   | 75   |       | NULL              | 0.000    |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
MariaDB [ligue1]>
```



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 [ligue1]> KILL 9;
Query OK, 0 rows affected (0.00 sec)
```

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

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host      | db      | Command | Time | State | Info              | Progress |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8  | root | localhost | ligue1  | Query   | 0    | NULL  | SHOW PROCESSLIST | 0.000    |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [ligue1]>
```



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: 10  
Current database: *** NONE ***
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
+-----+  
1 row in set (0.13 sec)
```

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



Important - Notez que le client se reconnecte au serveur!

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

```
MariaDB [ligue1]> SHOW PROCESSLIST;  
+---+---+---+---+---+---+---+---+---+---+---+---+  
| Id | User      | Host      | db      | Command | Time | State | Info              | Progress |  
+---+---+---+---+---+---+---+---+---+---+---+---+  
| 8  | root      | localhost | ligue1  | Query   | 0    | NULL  | SHOW PROCESSLIST | 0.000    |  
| 10 | fenestros | localhost | NULL    | Sleep   | 34   |       | NULL              | 0.000    |  
+---+---+---+---+---+---+---+---+---+---+---+---+  
2 rows in set (0.00 sec)
```

```
MariaDB [ligue1]>
```

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

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

Id	User	Host	db	Command	Time	State	Info	Progress
10	fenestros	localhost	NULL	Query	0	NULL	SHOW PROCESSLIST	0.000

1 row in set (0.00 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:

```
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. **MariaDB 5.x** fournit une base spécifique appelée **INFORMATION_SCHEMA** qui contient les **Méta-données** :

```
MariaDB [ligue1]> 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 |
+-----+
| CHARACTER_SETS               |
| CLIENT_STATISTICS            |
| COLLATIONS                    |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                     |
| COLUMN_PRIVILEGES             |
| ENGINES                      |
| EVENTS                       |
| FILES                        |
| GLOBAL_STATUS                 |
| GLOBAL_VARIABLES              |
| INDEX_STATISTICS              |
| KEY_CACHES                    |
| KEY_COLUMN_USAGE              |
| PARAMETERS                    |
| PARTITIONS                    |
| PLUGINS                      |
| PROCESSLIST                   |
| PROFILING                     |
| REFERENTIAL_CONSTRAINTS       |
| ROUTINES                     |
| SCHEMATA                     |
| SCHEMA_PRIVILEGES             |
| SESSION_STATUS                |
| SESSION_VARIABLES             |
| STATISTICS                    |
| TABLES                      |
| TABLESPACES                 |
| TABLE_CONSTRAINTS           |
| TABLE_PRIVILEGES             |
| TABLE_STATISTICS            |
```

TRIGGERS	
USER_PRIVILEGES	
USER_STATISTICS	
VIEWS	
INNODB_CMPMEM_RESET	
INNODB_RSEG	
INNODB_UNDO_LOGS	
INNODB_CMPMEM	
INNODB_SYS_TABLESTATS	
INNODB_LOCK_WAITS	
INNODB_INDEX_STATS	
INNODB_CMP	
INNODB_CMP_RESET	
INNODB_CHANGED_PAGES	
INNODB_BUFFER_POOL_PAGES	
INNODB_TRX	
INNODB_BUFFER_POOL_PAGES_INDEX	
INNODB_LOCKS	
INNODB_BUFFER_POOL_PAGES_BLOB	
INNODB_SYS_TABLES	
INNODB_SYS_FIELDS	
INNODB_SYS_COLUMNS	
INNODB_SYS_STATS	
INNODB_SYS_FOREIGN	
INNODB_SYS_INDEXES	
XTRADB_ADMIN_COMMAND	
INNODB_TABLE_STATS	
INNODB_SYS_FOREIGN_COLS	
INNODB_BUFFER_PAGE_LRU	
INNODB_BUFFER_POOL_STATS	
INNODB_BUFFER_PAGE	
+-----+-----+	
62 rows in set (0.00 sec)	

```
MariaDB [information_schema]>
```

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

Le programmes MySQL Administrator et MySQL Query Browser sont deux logiciels graphiques MySQL historiquement disponibles pour plusieurs plateformes. Comme son nom l'indique MySQL Administrator vous assiste dans les tâches d'administration. MySQL Query Browser permet, quant à lui, d'écrire des commandes et de les tester.

Téléchargez **mysql-workbench-community-6.3.10-1.el7.x86_64.rpm** :

```
[root@centos7 ~]# wget  
https://downloads.mysql.com/archives/get/p/8/file/mysql-workbench-community-6.3.10-1.el7.x86_64.rpm
```

Installez maintenant le paquet :

```
[root@centos7 ~]# yum localinstall mysql-workbench-community-6.3.10-1.el7.x86_64.rpm --nogpgcheck --skip-broken
```

MySQL Administrator et de MySQL Query Browser ont été remplacés par un nouveau logiciel appelé **MySQL Workbench** de la société **Oracle**.



A faire - Passez votre VM en démarrage graphique et re-bootez. Testez MySQL Administrator et MySQL Query Browser.

Pour passer votre VM en mode graphique, utilisez les commandes suivantes :

```
[root@centos7 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37 Apr 30 2016 /etc/systemd/system/default.target -> /lib/systemd/system/multi-user.target
[root@centos7 ~]# rm -rf /etc/systemd/system/default.target
[root@centos7 ~]# ln -s /lib/systemd/system/graphical.target /etc/systemd/system/default.target
[root@centos7 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 36 Dec 11 05:59 /etc/systemd/system/default.target -> /lib/systemd/system/graphical.target
[root@centos7 ~]# shutdown -h now
```

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@centos7 tmp]# mysql -u root -p
Enter password: fenestros1
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, 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 (0.14 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 (0.00 sec)
MariaDB [(none)]>
```



A faire - Pour plus d'information, voir

<http://dev.mysql.com/doc/refman/5.6/en/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 :

```
> 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	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 (0.42 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.



A faire - Pour plus d'information, voir
<http://dev.mysql.com/doc/refman/5.6/en/repair-table.html>.

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 (0.00 sec)
```

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



A faire - Pour plus d'information, voir

<http://dev.mysql.com/doc/refman/5.6/en/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 :

```
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 (0.00 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 **arretez le serveur mysqld** :

```
MariaDB [(none)]> exit
Bye
[root@centos7 tmp]# systemctl stop mariadb
[root@centos7 tmp]#
```

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

```
[root@centos7 tmp]# 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/liguel/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@centos7 tmp]# 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 0.01
Maximum resident set size 2104, Integral resident set size 0
Non-physical pagefaults 623, Physical pagefaults 0, Swaps 0
Blocks in 0 out 32, Messages in 0 out 0, Signals 0
Voluntary context switches 1, Involuntary context switches 0
```

Saisissez la commande suivante :

```
[root@centos7 tmp]# 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:     101
```

table description:

Key	Start	Len	Index	Type
1	2	4	unique	long
	6	4		long
	10	8		ulonglong

```
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@centos7 tmp]# systemctl start mariadb
```

Saisissez la commande suivante :

```
[root@centos7 tmp]# mysqlcheck -u root -p --host=127.0.0.1 ligue1
Enter password: fenestros
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

Option	Description
-B ou -databases	Permet de spécifier les bases de données à vérifier

<html> <center> Copyright © 2021 Hugh Norris. </center> </html>