

MySQL

- Objet : le système de gestion de bases de données relationnelles MySQL
- Niveau requis : [avisé](#)
- Commentaires : *installer, configurer et utiliser le SGBDR MySQL.*
- Suivi :
 - Créé par  [smolski](#) 24/07/2010
 - Testé par  [paskal](#) 18/10/2014
- Commentaires sur le forum : [c'est ici](#) ¹⁾

Introduction

MySQL est un serveur de base de données très souvent utilisé avec le langage de création de pages web dynamiques : PHP.

MySQL dérive directement de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel.

Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MySQL (sélections imbriquées, clés étrangères...).

On peut installer le serveur MySQL seul pour l'utiliser sans réseau.

Sinon, il est nécessaire d'y ajouter une application comme **Apache** (voir [Installation d'un serveur LAMP](#)).

Installation de mysql

```
apt-get install mysql-server
```

Sécurisation de mysql

En tant que super-utilisateur, tapez :

```
mysql_secure_installation
```

Et utilisez le mot de passe que vous avez déjà indiqué à l'installation.

```
Set root password? [Y/n]
```

tapez n (NON) pour le conserver.

Confirmez ensuite par simple validation :

```
Remove anonymous users? [Y/n]
```

Ôter l'utilisateur lambda créé automatiquement.

```
Disallow root login remotely? [Y/n]
```

Ne pas autoriser la gestion de mysql par le réseau, mais seulement en local.

```
Remove test database and access to it? [Y/n]
```

Supprimez la base de données test.

```
Reload privilege tables now? [Y/n]
```

Validez les modifications.

Utilisation de mysql

Accéder à mysql

```
mysql -u root -p
```

```
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 29  
Server version: 5.5.38-0+wheezy1 (Debian)
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```



Renseignez :

Enter password:
avec le *passwd mysql* et non le *passwd root* du système.

Lister tous les utilisateurs de mysql

```
use mysql; select * from user;
```



Les utilisateurs de MySQL/MariaDB sont stockés dans la table "user" de la base "mysql".

Tableau des commandes mysql

Les commandes mysql depuis l'appel dans un terminal ou en console.

Synopsis :

```
mysql> <commandes1 objet_A> <commandes2 objet_B>... ;
```

Dans la gestion de mysql :

1. Chaque séparation de colonne se termine par une virgule (,).
2. Chaque commande se détermine entre 2 parenthèses et se finalisent par un point virgule (;) afin d'être exécutées.

Tableau :

COMMANDES	SIGNIFICATION	COMMENTAIRE
-D	Accéder à la base de donnée spécifiée	-D <i>nom_de_la_base</i>
-u	Spécifier la session <i>utilisateur</i> à ouvrir dans mysql	-u <user> en général root pour l'administration de la base. Les users et le root de mysql ainsi que leurs passwd respectifs sont différents de ceux du système ouvert.
-p	permettra d'indiquer le passwd mysql à utiliser pour ouvrir la base	Les sécurités de mysql sont à renouveler pour chaque user mysql créé.
quit	Quitter mysql	
use <nom_de_la_base>	Accéder à une base de données	
create <objet>	Créer un <objet>	
create database	Créer une base de données	
create table	Créer une table	Les tables contiennent elle-même des colonnes qui les spécifies.
drop <objet>	Supprime l'objet spécifié.	
drop <nom_table> <nom_colonne>	Supprime une colonne dans la table spécifiée.	
drop <nom_table>	Supprime une table entière.	

COMMANDES	SIGNIFICATION	COMMENTAIRE
drop <nom_base>	Supprime une base spécifiée entièrement.	

Créer une base de donnée

Nota :

Pour l'exemple, nous allons créer une base de données nommée : agenda_net concernant les liens vers des sites qui nous sont les plus utiles.

```
mysql> create database agenda_net;  
Query OK, 1 row affected (0.00 sec)
```

Souvent, les simples utilisateurs ne peuvent pas créer une base des données, donc **root** doit créer la base et donner les privilèges à un simple utilisateur.

Pour donner les privilèges à un simple utilisateur, se connecter sous l'utilisateur **root de mysql**, puis :

```
GRANT ALL PRIVILEGES ON agenda_net.* TO 'user00'@'localhost' IDENTIFIED BY  
'pwd_user00';  
FLUSH PRIVILEGES;
```

Créer un simple utilisateur

Doit-être executer sous l'utilisateur root **de mysql**

```
CREATE USER 'login'@'hote' IDENTIFIED BY 'mot_de_passe';
```

Accéder à une base de donnée

Directement depuis le terminal :

```
mysql -D agenda_net -u root -p
```

```
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 77  
Server version: 5.5.38-0+wheezy1 (Debian)  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

Depuis mysql :

```
mysql> use agenda_net
Database changed
```

Importer/exporter une base de données

Depuis un terminal :

```
mysqldump -u user00 -p agenda_net > agenda.sql
```

Après avoir entré le mot de passe demandé, le fichier `agenda.sql` sera créé dans le répertoire `/home/user00`.

Il suffit ensuite de télécharger ce fichier de dump par [FTP](#) ou mieux encore par [SFTP](#).

À l'inverse, après avoir téléversé un fichier dump sur un serveur, la commande :

```
mysqldump -u user01 -p adresse_net < adresses.sql
```

remplira la base de données `adresse_net` à partir des données contenues dans le fichier `adresses.sql`.

Quitter mysql

```
mysql> quit
```

Tableau des codes mysql

Codes des chaînes de caractères et les textes :

CODE	SIGNIFICATION	COMMENTAIRE
unsigned	Après un nombre entier interdit un nombre négatif	
not null	La ligne spécifiée doit y être associée à une valeur	Seule une ligne vide permet à ce champ d'être vide (null).
null	Permet de laisser vide cette attribution	C'est le comportement par défaut de mysql.
auto_increment	Incrémentation automatique	
primary key	Clé d'identification de la table	Les entrées de cette colonne doivent être uniques. L'indexation y est automatique.
char(50)	Chaîne de caractères limités à 50	Pas d'espace, si ils sont nécessaires, les remplacer par un "_" ainsi : chaîne1_chaîne2
tinytext	Petit texte	
text	Texte de longueur normal	

CODE	SIGNIFICATION	COMMENTAIRE
mediumtext	Texte de longueur plus étendu	
longtext	Texte de longueur plus étendu encore	
tinyblob	Un petit blob	Un blob est un champ text qui tiendra compte des majuscules et minuscules
blob	Taille normale	
mediumblob	<i>Devinez ! 😊</i>	
longblob		

Codes des chiffres et comptages :

CODE	SIGNIFICATION	COMMENTAIRE
int (ou : integer)	Chiffres entiers classiques	
tinyint (ou : bool, ou bit)	Chiffres entiers très courts	
bigint	Chiffres entiers très longs	
smallint	Chiffres entiers courts	
mediumint	Chiffres entiers moyens	
float (avec une spécification. (Ex : 4,2 = 4 chiffres + 2 après la virgule)	Chiffres à virgules flottantes	
unsigned	Interdit les valeurs négatives	Après un nombre entier (int par exemple) interdit de prendre une valeur négative

Codes particuliers de limitations :

CODE	SIGNIFICATION	COMMENTAIRE
enum (valeur1,valeur2,...)	Limite uniquement à une des valeurs listées ou null	
set (valeur1,valeur2,...)	Limite uniquement à une ou des ensembles de ces valeurs listées ou null	

Créer des tables



Nous devons auparavant nous situer dans la base de donnée créée pour y créer des tables bien sûr !



Nous ne pouvons pas créer de table vide.

```
mysql> create table adresse_net
( adresse_id int unsigned not null auto_increment primary key,
  adresse_nom char(50) not null,
  adresse_adresse text not null,
  adresse_comment mediumtext null,
  adresse_niveau tinyint unsigned null
);
```

où :

adresse_id	<i>est la colonne d'identification du site référencé</i>
adresse_nom	<i>est le nom du site référencé</i>
adresse_adresse	<i>l'URL du site recensé</i>
adresse_comment	<i>permet de commenter le site référencé</i>
adresse_niveau	<i>donne un niveau de compétence pour ce site référencé</i>

Insertion de données

Synopsis d'insertion de données :

Déjà logés dans la base de données :

```
mysql> insert into <table> value (données_colonne1, données_colonne2);
```

Ce qui donne pour l'exemple de la base agenda_net :

```
mysql> insert into adresse_net value
(NULL, 'debian-facile',
'http://debian-facile.org/forum/index.php',
'Site Debian, la ou Debian est le plus simple',
'1');
```

Vérification d'une table

C'est pour voir le code qui a servit a créer la table00:

```
show create table00\G
```

Lecture dans la base de données.

Avec :

```
mysql> select adresse_nom
from adresse_net;
```

On obtient ce tableau :

```
+-----+
| adresse_nom |
+-----+
| debian-facile |
+-----+
1 row in set (0.00 sec)
```

Pour avoir tout les résultats possibles, on peut utiliser :

```
mysql> select * from adresse_net;
```

```
select * from adresse_net;
+-----+-----+-----+-----+
| adresse_id | adresse_nom | adresse_adresse |
adresse_comment | adresse_niveau |
+-----+-----+-----+-----+
| 1 | debian-facile | http://debian-facile.org/forum/index.php |
Site Debian, la ou Debian est le plus simple | 1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Et voilà ! 😊



Pour la suite du tuto, créez plusieurs adresses différentes,

Par exemple :

```
mysql> insert into adresse_net value
(NULL, 'chezlefab',
'http://wiki.chezlefab.net/tuto_nix/accueil',
'Site Debian, toujours un peu plus loin',
'1');
```

```
mysql> insert into adresse_net value
(NULL, 'lehobey',
'http://lehobey.net/dokuwiki/doc:backuppc:backuppc',
'backuppc - la sauvegarde en ligne',
'2');
```

```
mysql> insert into adresse_net value
(NULL, 'debian-fr.org',
'http://forum.debian-fr.org/viewtopic.php?f=8&t=23693&p=232939&hilit=pxe#p23
2939',
'PXE - Installer une Debian sur une machine sans CD a partir de son reseau',
'4');
```

```
mysql> insert into adresse_net value
(NULL, 'ixquick.com',
'https://ixquick.com/',
'Moteur de recherche libre.',
'1');
```

```
mysql> insert into adresse_net value
(NULL, 'slackware-fr.org',
'http://www.slackware-fr.org/');
```

```
'slackware le portail du site francais.',  
'2');
```

```
mysql> select adresse_nom  
from adresse_net;  
+-----+  
| adresse_nom |  
+-----+  
| debian-facile |  
| chezlefab |  
| lehobey |  
| debian-fr.org |  
| ixquick.com |  
| slackware-fr.org |  
+-----+  
6 rows in set (0.00 sec)
```

Option alter - Modification des tables

Option add / first / after / drop - Ajouter/Ôter une colonne dans une table :

Ajouter dans la table `adresse_net` la colonne `adresse_OS` :

```
mysql> alter table adresse_net add adresse_OS char(10);  
Query OK, 6 rows affected (0.00 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

Ainsi, la nouvelle colonne se rajoute à la fin des autres.

Pour l'ajouter en première position utiliser l'option `first` à la fin de la commande :

```
mysql> alter table adresse_net add adresse_OS char(10) first;  
Query OK, 6 rows affected (0.00 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

Pour insérer la nouvelle colonne après une colonne spécifique, toujours en fin de ligne de commande, utiliser l'option `after` ainsi :

```
mysql> alter table adresse_net add adresse_OS char(10) after adresse_id;  
Query OK, 6 rows affected (0.00 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

Ceci a placé la colonne `adresse_OS` après la colonne `adresse_id`.

Pour ôter une colonne (et tous ses renseignements contenus), utiliser l'option `drop` ainsi :

```
mysql> alter table adresse_net drop adresse_OS;  
Query OK, 6 rows affected (0.00 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

Vider une table

Il faut que la table à vider ne soit pas reliée à aucune autre table:

```
truncate table00;
```

Ainsi, si la table possède des relations avec d'autres tables alors il faut supprimer cette relation

```
alter table table_a_relation drop foreign key nom_relation;
```

Pour voir le nom_relation:

```
show create table table_a_relation; #ou  
show create table table_a_relation\G
```

Option modify - Modifier l'attribut d'une colonne :

Après cette commande :

```
mysql> alter table adresse_net add adresse_05 char(10) after adresse_id;  
Query OK, 6 rows affected (0.00 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

nous pouvons juger que *10 caractères* ne vont pas être suffisants pour cette colonne.
Pour passer à *20 caractères*, nous utiliserons l'option `modify` ainsi :

```
mysql> alter table adresse_net modify adresse_05 char(20);  
Query OK, 6 rows affected (0.01 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

Vérification d'une base

C'est pour vérifier les relations entre les tables:

```
select * from information_schema.referential_constraints where  
constraint_schema = 'nom_base';
```

Utilisation avancée

Trigger

Trigger: pour executer des commandes sql après certains actions,

appeler aussi un `__déclenheur__`

ex: après l'insertion dans une `table00` on veut une insertion dans `table01`: Ainsi,

- 1) on crée une autre delimiter,
- 2) on crée le trigger,
- 3) on change le trigger par sa valeur par défaut, (';')

Création trigger

```
delimiter | #pour la creation du trigger
create trigger trigger00 after insert on table00 for each row
begin
    #instruction of the trigger
    insert into table01(col00, col01) values('val00', 'val01');
end
delimiter ; #on remet le trigger par sa valeur par défaut
```

Vérification des triggers

c'est pour voir les triggers qui sont dans la base

```
show triggers \G
```

Ainsi, on a les triggers qui sont présents, \et pour voir la définition du trigger00:

```
show create trigger trigger01;
```

Liens

- [Toute la doc](#)
- [Un lien formidable où télécharger un pdf à conserver](#)
- [Sauvegarder automatiquement ses bases de données](#) (sur ubuntu-fr.org)

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:programmation:mysql>



Last update: **25/10/2018 19:03**