mysql : Fiche documentation (3e partie) - gérer les utilisateurs

Gérer les utilisateurs c'est déterminer qui peut se connecter à un serveur, depuis où (le ou les hôtes) et ce qu'il peut faire. Les informations relatives aux utilisateurs sont stockées dans une base relationnelle nommée mysql.

1. La base mysql

Cette base peut être modifiée de deux manières :
- avec des commandes `insert, update, delete` ... mais c'est fortement déconseillé,
- avec les commandes spécifiques : `create user, drop user, rename user, set password, grant, revoke`

La table user de la base mysql contient les informations sur les utilisateurs. Les trois premiers champs (host, user et password) nous indiquent qui peut se connecter depuis quel endroit et avec ou sans mot de passe.

```
mysql> select host,user,password from user;
+-----------+---------+-------------------------------------------+
| host      | user    | password                                  |
+-----------+---------+-------------------------------------------+
| localhost | root    |                                           |
| %         | root    | *BBF81B0057193667E51780C7AF02F7E5CD9274AA |
| %         | lacurie | *9E96B6CE23B0719C2FEB68A679926980C788ACE9 |
| localhost | colling | *1EA525CEE8049E1D04E8768BB4F5C2FB437152BE |
+-----------+---------+-------------------------------------------+
4 rows in set (0.00 sec)
```

Sécurité !!

2. La gestion des utilisateurs

Créer des utilisateurs

commande : create user 'utilisateur'@'hôte' identified by 'mot de passe';

Exemple 1 :

```
mysql> create user david@'192.168.0.55' identified by 'krizan';
Query OK, 0 rows affected (0.01 sec)
mysql> select host,user,password from user where user='david';
+--------------+-------+-------------------------------------------+
| host         | user  | password                                  |
+--------------+-------+-------------------------------------------+
| 192.168.0.55 | david | *9B147CCECA0D376C1E701173441CE8ED551F776B |
+--------------+-------+-------------------------------------------+
1 row in set (0.00 sec)
```

exemple 2 :

```
mysql> create user laurent@'192.168.0.6' identified by 'secret';
Query OK, 0 rows affected (0.37 sec)

mysql> create user laurent@'192.168.0.%' identified by 'arle';
Query OK, 0 rows affected (0.00 sec)
```

ici % remplace le dernier octet de l'IP

```
mysql> create user laurent@'%' identified by 'btsig2';
Query OK, 0 rows affected (0.00 sec)
```

ici % désigne n'importe quel hôte

```
mysql> select host,user,password from user where user='laurent';
+-------------+---------+-------------------------------------------+
| host        | user    | password                                  |
+-------------+---------+-------------------------------------------+
| 192.168.0.6 | laurent | *14E65567ABDB5135D0CFD9A70B3032C179A49EE7 |
| 192.168.0.% | laurent | *7FCBB371AD8C19D4512D68DEAE022AFA06DA14AD |
| %           | laurent | *DD3118DDB1630B9C52EA2C011270AB47B1399218 |
+-------------+---------+-------------------------------------------+
3 rows in set (0.00 sec)
```

Il y a 3 fois le compte laurent avec 3 mots de passe différents ! Que se passe-t-il lorsque laurent se connecte depuis l'hôte 192.168.0.7 ? Deux lignes ci-dessus répondent aux critères (l'avant dernière et la dernière) mais MySQL choisit toujours l'hôte le plus spécifique, c'est donc l'avant dernière ligne qui est choisie : laurent devra se connecter depuis cet hôte avec le mot de passe arle.

Dans sa version courte la commande create user va créer un utilisateur sans mot de passe (attention à la sécurité !)

Exemple :

```
mysql> create user anonymous@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> select host,user,password from user where user='anonymous';
+------+-----------+----------+
| host | user      | password |
+------+-----------+----------+
| %    | anonymous |          |
+------+-----------+----------+
1 row in set (0.00 sec)
```

Renommer des utilisateurs

Commande : rename user 'ancien_nom_utilisateur'@'ancien_hôte' to 'nouveau_nom_utilisateur'@'nouvel_hôte'

Exemple :

```
mysql> rename user colling@'localhost' to philippe@'%';
Query OK, 0 rows affected (1.44 sec)

mysql> select host,user,password from user;
+--------------+-----------+-------------------------------------------+
| host         | user      | password                                  |
+--------------+-----------+-------------------------------------------+
| localhost    | root      | *BBF81B0057193667E51780C7AF02F7E5CD9274AA |
| %            | root      | *BBF81B0057193667E51780C7AF02F7E5CD9274AA |
| %            | lacurie   | *9E96B6CE23B0719C2FEB68A679926980C788ACE9 |
| %            | philippe  | *1EA525CEE8049E1D04E8768BB4F5C2FB437152BE |
...
```

Supprimer des utilisateurs

Commande : drop user 'utilisateur'@'hôte'

Exemple :

```
mysql> drop user laurent@'192.168.0.%';
```

Problème d'identité

Dans certaines circonstances (essais, ...) il peut être utile de connaître son identité, ceci ce fait grâce à la fonction user(), elle s'utilise comme dans l'exemple ci-après :

```
mysql> select user();
+----------------+
| user()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

3. Les mots de passe

Modifier ou affecter un mot de passe

Commande : set password for 'utilisateur'@'hôte' = password('nouveau_mot_de_passe');

Exemple :

```
mysql> set password for david@'192.168.0.55' = password('christophe');
Query OK, 0 rows affected (0.00 sec)

mysql> select host,user,password from user where user='david';
+--------------+-------+-------------------------------------------+
| host         | user  | password                                  |
+--------------+-------+-------------------------------------------+
| 192.168.0.55 | david | *F581D39D8CDB00E97C3FC9DCD6E69C62D98A0CDE |
+--------------+-------+-------------------------------------------+
1 row in set (0.00 sec)
```

Un utilisateur connecté peut changer son propre mot de passe.

Commande : set password = password('nouveau_mot_de_passe');

Exemple :

```
mysql> set password = password('christophe');
Query OK, 0 rows affected (0.00 sec)
```

Supprimer un mot de passe (dans ce cas il faut bien savoir ce que l'on fait !)

Commandes : set password for 'utilisateur'@'hôte' = '';
set password ='"; (si on veut supprimer son propre mot de passe)

Exemple :

```
mysql> set password for david@'192.168.0.55' ='';
Query OK, 0 rows affected (0.01 sec)

mysql> select host,user,password from user where user='david';
+--------------+-------+-------------------------------------------+
| host         | user  | password                                  |
+--------------+-------+-------------------------------------------+
| localhost    | david | *F581D39D8CDB00E97C3FC9DCD6E69C62D98A0CDE |
| 192.168.0.55 | david |                                           |
```

Le chiffrage des mots de passe

MySQL ne stocke pas les mots de passe en clair mais sous forme chiffrée. Le même mot de passe en clair donnera toujours le même mot de passe chiffré (déterminisme) mais il est « impossible » de retrouver le mot de passe en clair à partir du mot de passe chiffré (irréversibilité). MySQL ne connaît pas les mots de passe en clair. Lorsqu'un utilisateur se connecte MySQL chiffre son mot de passe et le compare avec celui stocké dans la table user. Les mots de passe chiffrés sont constitués d'un astérisque suivi de 40 chiffres hexadécimaux.

4. Les privilèges

En base de données on parle de privilèges et non de droits. Lorsqu'un utilisateur est créé il n'a pratiquement aucun privilège.

Exemple :

```
mysql -u david -p
Enter password: **********
...
mysql> use biblio
ERROR 1044 (42000): Access denied for user 'david'@'localhost' to database 'biblio'
```

Accorder des privilèges

Il faut distinguer deux types de privilèges : les privilèges administrateur qui ne sont associés à l'utilisateur sans considération de bases, tables ... et les privilèges liés à des objets (bases, tables, colonnes ...).

Le tableau ci-dessous, obtenu par la commande show privileges, liste les privilèges administrateur (Server Admin ou File ... dans la colonne context) et les privilèges liés à des objets (indiqués dans la colonne context).

Privilege	Context	Comment
01. Alter	Tables	To alter the table
02. Alter routine	Functions,Procedures	To alter or drop stored functions/procedures
03. Create	Databases,Tables,Indexes	To create new databases and tables
04. Create routine	Functions,Procedures	To use CREATE FUNCTION/PROCEDURE
05. Create temporary tables	Databases	To use CREATE TEMPORARY TABLE
06. Create view	Tables	To create new views
07. Create user	Server Admin	To create new users
08. Delete	Tables	To delete existing rows
09. Drop	Databases,Tables	To drop databases, tables, and views
10. Execute	Functions,Procedures	To execute stored routines
11. File	File access on server	To read and write files on the server
12. Grant option	Databases,Tables,Functions,Procedures	To give to other users those privileges you possess
13. Index	Tables	To create or drop indexes
14. Insert	Tables	To insert data into tables
15. Lock tables	Databases	To use LOCKTABLES (together with SELECT privilege)
16. Process	Server Admin	To view the plain text of currently executing queries
17. References	Databases,Tables	To have references on tables
18. Reload	Server Admin	To reload or refresh tables, logs and privileges
19. Replication client	Server Admin	To ask where the slave or master servers are
20. Replication slave	Server Admin	To read binary log events from the master
21. Select	Tables	To retrieve rows from table
22. Show databases	Server Admin	To see all databases with SHOW DATABASES
23. Show view	Tables	To see views with SHOW CREATE VIEW
24. Shutdown	Server Admin	To shut down the server
25. Super	Server Admin	To use KILL thread, SET GLOBAL, CHANGE MASTER, etc.
26. Update	Tables	To update existing rows
27. Usage	Server Admin	No privileges - allow connect only

La commandes qui permet d'accorder des privilèges est : grant privilège on base.table to 'utilisateur'@'hôte'.

Exemple de privilège accordé au niveau global :

1ère étape : root accorde à david le privilège de créer des utilisateurs.

```
mysql> select user();
+----------------+
| user()         |
+----------------+
| root@localhost |
+----------------+
```

```
mysql> grant create user on *.* to david@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

. désigne toute les tables de toutes les BDD

2e étape : david crée un utilisateur

```
mysql> select user();
+-----------------+
| user()          |
+-----------------+
| david@localhost |
+-----------------+
```

```
mysql> create user loic@'localhost' identified by 'binome';
Query OK, 0 rows affected (0.00 sec)
```

3e étape : root peut vérifier que l'utilisateur a bien été créé et que david a le privilège de créer les utilisateurs.

```
mysql> select host,user,password,create_user_priv from user;
+-------------+-----------+-------------------------------------------+------------------+
| host        | user      | password                                  | create_user_priv |
+-------------+-----------+-------------------------------------------+------------------+
| localhost   | root      | *BBF81B0057193667E51780C7AF02F7E5CD9274AA | Y                |
| localhost   | loic      | *3BBCC85C70B24970C3A5814C993E46F464FD5B33 | N                |
...
| localhost   | david     | *F581D39D8CDB00E97C3FC9DCD6E69C62D98A0CDE | Y                |
```

MySQL distingue 4 niveaux de privilèges et utilise pour stocker ces informations 4 tables de la base mysql : global (table user), base (table db), table (table tables_priv) ou routine (procs_priv) et colonne (table columns_priv).

Exemples :

```
mysql> grant select on *.* to laurent@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on biblio.* to loic@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> grant update on biblio.auteurs to loic@'localhost';
Query OK, 0 rows affected (0.37 sec)

mysql> grant select (nom,prenom) on biblio.auteurs to david@'192.168.0.55';
Query OK, 0 rows affected (0.00 sec)
```

Les privilèges peuvent entrer en conflit, dans ce cas MySQL donne toujours la préférence à l'autorisation.

Le mot ALL permet d'accorder tous les privilèges.

Exemple :

```
mysql> grant all on *.* to lacurie@'%';
Query OK, 0 rows affected (0.00 sec)
```

Connaître les privilèges

Exemple : un utilisateur peut connaître ses privilèges

```
mysql> show grants;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| Grants for david@localhost                                                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| GRANT SHOW DATABASES, CREATE USER ON *.* TO 'david'@'localhost' IDENTIFIED BY PASSWORD '*F581D39D8CDB00E97C3FC9DCD6E69C62D98A0CDE'|
| GRANT SELECT (prenom, nom) ON `biblio`.`auteurs` TO 'david'@'localhost'                                                                                                                       |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

root ou tout utilisateur qui a les privilèges pourra connaître les privilèges d'autres utilisateurs.

```
mysql> show grants for laurent@'%';
+-----------------------------------------------------------------------------------------------------------------+
|Grants for laurent@%                                                                                             |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT SELECT ON *.* TO 'laurent'@'%' IDENTIFIED BY PASSWORD '*DD3118DDB1630B9C52EA2C011270AB47B1399218' |
+-----------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

Supprimer des privilèges.

La commande revoke sert à enlever des privilèges à un utilisateur sa syntaxe est proche de la commande grant :

revoke privilège on base.tables from 'utilisateur'@'hôte'

Exemple :

```
mysql> revoke select (nom) on biblio.auteurs from david@'192.168.0.55';
Query OK, 0 rows affected (0.00 sec)

mysql> revoke all privileges on *.* from lacurie@'%';
Query OK, 0 rows affected (0.00 sec)
```