

DBA MySQL

Tutoriels et Cours

SOMMAIRE

1. INTRODUCTION
 - SCHEMAS
 - TABLES
 - USERS
 - CONNECTIONS
 - MYSQLDUMP
2. INSTALLATION MYSQL (WINDOWS|LINUX)
3. MOTEURS MYSQL
 - MYISAM
 - INNODB
 - MEMORY
 - MERGE
 - AUTRES
4. OUTILS MYSQL
 - ROUTINES
 - PROCEDURES STOCKEES
 - FONCTIONS
 - VUES
 - TRIGGERS – DECLENCHEURS
 - EVENT – SCHEDULER
5. PARTITIONNEMENT DES TABLES
6. SURVEILLER MYSQL - TUNING
 - VARIABLES DE CONF ET DE STATUS
7. OPTIMISATION DES REQUETES ET DES INDEX
8. MULTI-INSTANCES DE MYSQL
9. REPLICATION
 - SIMPLE
 - CROISEE
 - EN MODE MULTI-INSTANCES

Tutoriels et Cours



LES DATAS DÉFINITIONS

Nous allons voir les définitions de conception et de modélisation du langage SQL.

Le langage SQL (Structured Query Language) regroupe différentes parties de conception, de manipulation de données et de gestion des droits des utilisateurs .

1ère Partie : Le Data Definition Language : DDL

C'est la partie du langage qui permet de créer des bases de données, les tables, les index Elle regroupe les instructions **CREATE**, **ALTER**, **DROP** (créer, modifier, supprimer un élément de la base).

2ne Partie : Le Data Manipulation Language : DML

C'est la partie du langage qui traite les données.Elle regroupe les instructions **INSERT**, **UPDATE**, **DELETE**, **SELECT** (insertion, mise à jour, suppression et extraction de données).

3ème Partie : Le Data Control Language : DCL

C'est la partie du SQL qui gère les droits d'accès aux tables. Elle regroupe les instructions **GRANT**, **REVOKE** (attribution et suppression des droits).

Tutoriels et Cours

SCHEMAS MYSQL

Dans MySQL un schéma est le synonyme de **database**, on peut donc employer l'un ou l'autre terme. Par contre dans certains SGBR, le schéma ne peut contenir que certaines parties de la database.

```
mysql>
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| test |
| world |
+-----+
6 rows in set (0.00 sec)

mysql> show schemas;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| test |
| world |
+-----+
6 rows in set (0.00 sec)

mysql>
```

Tutoriels et Cours

CONNEXION ET UTILISATION

⇒ Comment se connecter au serveur MySQL en ligne de commande :

`mysql -u root -p` --Le mot de passe est ensuite demandé ...

`use centrale;` -- pour utiliser la base centrale

`mysql -u root -p -D centrale` -- plus directe

⇒ Comment créer une base de données :

Voici un exemple qui permet de créer la base de données nommée centrale avec un jeu d'encodage de caractères en UTF-8 (**default character set**) et un jeu de règles pour comparer et rechercher les données dans le jeu d'enregistrements (**collate**) basé sur du Case insensitive dans notre cas :

```
create database centrale DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

⇒ Comment créer une table en précisant un moteur de stockage :

```
CREATE TABLE formation ( pk_cours int(11) NOT NULL AUTO_INCREMENT,  
cours varchar(100) NOT NULL,  
durée int(7) NOT NULL,  
prix decimal(10,0) NOT NULL, catégorie varchar(150) NOT NULL,  
PRIMARY KEY (pk_cours)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ;
```

NB : on peut tout écrire en minuscules – Si on ne précise pas la taille des entiers MySQL donne une valeur de int(11)

SHOW CREATE TABLE formation

+ Options

Table	Create Table
formation	CREATE TABLE `formation` (`pk_cours` int(11) NOT NULL AUTO_INCREMENT, `cours` varchar(100) NOT NULL, `durée` int(7) NOT NULL, `prix` decimal(10,0) NOT NULL, `catégorie` varchar(150) NOT NULL, PRIMARY KEY (`pk_cours`)) ENGINE=MyISAM DEFAULT CHARSET=utf8

Tutoriels et Cours



SCHEMA MYSQL.USER

Le schéma système **mysql** contient les tables qui permettent de gérer totalement les droits des utilisateurs MySQL . Ces tables sont :

- **user** : contient les privilèges de chaque utilisateur
- **db** : contient les droits spécifiques à une base (schéma)
- **tables_priv** : contient les droits spécifiques à une table ou à une vue
- **columns_priv** : contient les droits spécifiques à une colonne (champ)
- **procs_priv** : contient les droits des routines (procédures ou fonctions)

```
1  -- 1-Affichage des comptes root
2  • SELECT user,host,password FROM mysql.user where user='root';
3
4  -- 2-Suppression des comptes root à distance
5  • DROP USER 'root'@'192.168.10.1';
6  • DROP USER 'root'@'%';
7  • DROP USER 'root'@'127.0.0.1';
8
9  -- 3-Renommer le compte root
10 • RENAME USER 'root'@'localhost' TO 'chef'@'localhost';
11 • SELECT user,host,password FROM mysql.user where user='chef';
12
13 • SELECT user,host,password FROM mysql.user;
14
```

Tutoriels et Cours



GESTION DES USERS

⇒ Voici un exemple qui permet de créer un utilisateur nommé michel associé au mot de passe tux (clause optionnelle) et qui aura accès au serveur local.

```
CREATE USER 'michel'@'localhost' IDENTIFIED BY 'tux';
```

IDENTIFIED encrypte le mdp avec password par défaut

Ou **CREATE USER 'michel'@'localhost';** -- sans mot de passe

⇒ Le mot de passe est ici saisi en clair, il vaudrait mieux utiliser l'empreinte SHA1 ou md5 que l'on peut générer avec la commande `select sha1('tux')` , ce qui donnerait :

```
CREATE USER 'michel'@'localhost' IDENTIFIED BY  
'*AC5CC6D66967DC0DC9222066A7D80A36208DF22D';
```

Pour rappel md5 et sha1 sont des algorithmes de hashage, ils créent une empreinte du mot de passe et il est donc plus compliqué de récupérer l'original.

Malgré tout les attaques dites de force brute basées sur de puissants dictionnaires peuvent venir à bout d'un hash d'où la nécessité d'utiliser des mots de passe complexes.

⇒ Un autre exemple qui donne les droits classiques uniquement sur la database base_formation et la table table_planning ...

```
GRANT SELECT , INSERT ,UPDATE ,DELETE ,CREATE ,DROP ,ALTER ON base_formation.table_planning TO 'bob'@'localhost';
```

ALL PRIVILEGES donne tous les privilèges ...

SUPER permet de modifier à chaud les variables systèmes ...

USAGE ne donne aucun droits ...

```
GRANT USAGE ON * . * TO 'michel'@'localhost';
```

```
REVOKE ALL PRIVILEGES ON * . * FROM 'michel'@'localhost';
```

 supprime tous les droits

⇒ Les commandes **GRANT REVOKE** et **SET PASSWORD** sont prises immédiatement en compte par MySQL.

Si on modifie les privilèges d'un utilisateur avec les insert ou update directement dans la table `mysql.user`, on doit exécuter la commande **FLUSH PRIVILEGES** pour prise d'effet.

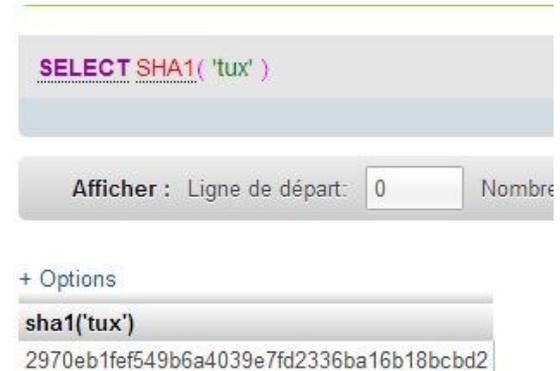
```
SET PASSWORD FOR 'bob'@'localhost' = PASSWORD(NouveauMDP');
```

-- est l'équivalent de :

```
UPDATE mysql.user SET Password=PASSWORD(NouveauMDP') WHERE User='bob' AND Host='localhost';
```

```
FLUSH PRIVILEGES;
```

Reproduction Interdite



Tutoriels et Cours



MYSQLDUMP

MySqlDump est un outil de sauvegarde qui peut exporter et restaurer une ou plusieurs bases de données en incluant les routines, les données et la structure.

Exemple 1: exporte toutes les bases vers svg-25-12-2012.sql
(le user concerné est root avec un mdp debian)

⇒ `mysqldump -u root -p debian --all-databases --opt > svg-25-12-2012.sql`

Exemple 2 : export d'une base de données nommée centrale

⇒ `mysqldump -u root -p debian -B centrale > /tmp/centrale.sql`

Exemple : Importe la sauvegarde au format sql centrale.sql

⇒ `mysql -uroot -pdebian centrale < /tmp/centrale.sql`

Tutoriels et Cours

TP - USER

- **Créer un schéma nommé world en utf-8**
- **Créer une table ville :**
 - pk_ville (integer auto_increment primary key)
 - Ville varchar(150)
 - Cp varchar(5)
 - Dept varchar(150)
 - Habitants integer
 - Moteur MyISAM
- **Insérer 3 villes dans la table :**
 - Nice 06200 alpes maritimes 343000
 - Sophia Antipolis 06600 Alpes Maritimes 9102
 - Antibes 06600 Alpes Maritimes 75820
- **Importer avec la commande mysql les datas world.sql**
- **Créer 2 utilisateurs : (mdp identique au login)**
 - Bob droits select sur table ville
 - Tux droits select + **execute** sur une base (execute concerne avant tout les bases !!)

```
mysql> grant select, execute on world.* to 'bob'@'localhost';
```

- Admin tous les droits sur table ville
- Mac : droits de base (dml) sans mot de passe

Tutoriels et Cours

