

INSIA - ING 1

Bases de données

Piscine MySQL – Cours et TP n° 09

Utilisateur – Droits – Vue – Sauvegarde

LA BIBLE : MySQL 5.0 Reference Manual

<http://dev.mysql.com/doc/refman/5.0/fr/index.html>

Site officiel MySql : <http://www-fr.mysql.com/>

Documentation MySQL : <http://mysql.org/>

La Base de Données Open Source la plus Populaire au Monde

Bertrand LIAUDET

SOMMAIRE

SOMMAIRE	<u>1</u>
UTILISATEURS - DROIT – VUE - SAUVEGARDE	<u>2</u>
1. Les utilisateurs et leurs droits	2
2. La notion de vue	5
3. Sauvegarde	7
4. Organisation physique des données	8

Première édition : décembre 2007

UTILISATEURS - DROIT – VUE - SAUVEGARDE

PRINCIPALES NOTIONS

Create user
Grant
Vue
Sauvegarde
Show create table

Drop user
Revoke

mysqldump

1. Les utilisateurs et leurs droits

Présentation d'un utilisateur

Un utilisateur est caractérisé par son nom et par son hôte : **host, user**.

```
mysql> select user();
```

Création d'un utilisateur

Par la commande CREATE USER

La commande CREATE USER crée un utilisateur qui n'a aucun droit. Il ne peut donc rien faire avec la BD. Il faudra ensuite donner des droits avec un GRANT.

```
CREATE USER nomUtilisateur[@nomHote] [IDENTIFIED BY motDePasse]
```

Pour créer un utilisateur, il faut être un utilisateur qui a le droit de créer des utilisateurs (with grant option).

Quand on ne précise pas le nom de l'utilisateur, la commande vaut pour tout hôte. Tous les hôtes : '%' (entre quotes simples)

Par la commande GRANT

La commande GRANT appliquée à un utilisateur n'existant pas crée cet utilisateur.

➤ *Exemple de création d'un utilisateur sans droits*

```
mysql> GRANT USAGE ON *.* TO toto IDENTIFIED BY 'mdpToto';
```

L'utilisateur toto peut se connecter sur n'importe quelle machine et n'a aucun droits. Il accède uniquement à la BD information_schema.

Suppression d'un utilisateur

Par la commande DROP USER

```
DROP USER nomUtilisateur[@nomHote]
```

Modification des mots de passe

Par la commande SET PASSWORD

```
mysql> SET PASSWORD FOR nomUtilisateur = PASSWORD('motDePasse')
```

La fonction PASSWORD() permet de crypter le mot de passe.

Consultation de la liste des utilisateurs

Consultation directe de la table des USER

```
mysql> select host, user, password from mysql.user;
```

Liste de privilèges

Les principaux privilèges sont les suivants :

Nom	Classe	Niveau	Droit
USAGE	Serveur	User	Pour créer un utilisateur sans droits.
ALL [PRIVILEGES]	Serveur	User	Tous les privilèges, sauf WITH GRANT OPTION
WITH GRANT OPTION	Serveur	User	Autorise l'utilisation de GRANT

Nom	Classe	Niveau	Droit
CREATE	Serveur	BD, Table	Création (toutes les créations possible)
DROP	Serveur	BD, Table	Destruction de bases ou de tables
ALTER	Serveur	Table	Autorise l'utilisation de ALTER.

Nom	Classe	Niveau	Droit
SHOW DATABASES	Client	User	Autorise l'utilisation de SHOW DATABASES.
SELECT	Client	Tuples	Autorise l'utilisation de SELECT.
INSERT	Client	Tuples	Autorise l'utilisation de INSERT.
UPDATE	Client	Tuples	Autorise l'utilisation de UPDATE.
DELETE	Client	Tuples	Autorise l'utilisation de DELETE.

Donner des privilèges : commande GRANT

Syntaxe simplifiée

```
GRANT privilege [,privilege] ON composant TO nomUtilisateur  
[IDENTIFIED BY motDePasse] [WITH GRANT OPTION]
```

Exemples

➤ Exemple de création d'un super-utilisateur

```
mysql> GRANT ALL PRIVILEGES ON *.* TO admin2@localhost IDENTIFIED  
BY 'mdpAdmin' WITH GRANT OPTION;
```

- *Donner le droit à l'utilisateur 1 de consulter la table emp :*

```
GRANT SELECT ON emp TO utilisateur1;
```

- *Consultation et insertion :*

```
GRANT SELECT, INSERT ON emp TO utilisateur1;
```

- *Tous les droits pour deux utilisateurs :*

```
GRANT ALL PRIVILEGES ON emp TO utilisateur1, utilisateur2;
```

- *Limiter les droits à certains attributs*

```
GRANT SELECT (ename, job, deptno) ON emp TO utilisateur1;
```

Retirer des privilèges : commande REVOKE

Syntaxe simplifiée

```
REVOKE privilege [,privilege] ON composant FROM nomUtilisateur  
[IDENTIFIED BY motDePasse] [WITH GRANT OPTION]
```

Exemples

- *Suppression de tous les droits sauf le grant option:*

```
mysql> revoke all privileges on *.* from admin2@localhost identified by  
      'mdpAdmin';
```

équivalent à :

```
mysql> revoke all on *.* from admin2@localhost identified by 'mdpAdmin';
```

- *Suppression du droit de grant :*

```
mysql> revoke grant option on *.* from admin2@localhost identified by  
      'mdpAdmin';
```

- *Suppression du droit d'insertion de l'utilisateur 1 :*

```
REVOKE INSERT ON emp TO utilisateur1;
```

Consulter les privilèges : commande SHOW

Pour voir la commande de création des droits :

```
mysql> SHOW GRANTS FOR user ;
```

2. La notion de vue

Présentation

Une commande **select** peut être conservée dans une variable appelée "vue".

Une vue est donc une table virtuelle :

- elle n'a pas d'existence physique ;
- elle est recalculée à chaque utilisation ;
- elle est équivalente à une requête.

Création d'une vue

La syntaxe de la création d'une vue est la suivante :

```
CREATE VIEW nom_vue AS  
select ...
```

ou

```
CREATE or REPLACE VIEW nom_vue AS  
select ...
```

Utilisation d'une vue

Une vue s'utilise comme une table.

Consultation du code d'une vue

```
mysql> SHOW CREATE VIEW nom_vue;
```

Suppression d'une vue

Pour supprimer une vue, on écrira :

```
DROP VIEW nom_vue ;
```

Exemple

Tous les employés travaillant dans un département qui contient au moins un 'ANALYST' (c'est un métier) :

```
CREATE or REPLACE VIEW vue1 as  
Select distinct nd from emp  
where job = 'ANALYST'  
order by nd;
```

```
Select ne, nom from emp  
where nd in (select nd from vue1 );
```

ou

```
Select distinct e.ne, e.nom from emp e, vue1 v  
where e.nd =v.nd;
```

Les deux usages des vues

- Pour décomposer les requêtes en sous-requêtes
- Pour mieux gérer la protection des données en limitant la visibilité des données par les utilisateurs.

3. Sauvegarde

```
shell > mysql dump
```

<http://dev.mysql.com/doc/refman/5.0/fr/mysqldump.html>

Mysqldump est une commande qui permet de produire le code SQL permettant de recréer entièrement la BD.

Trois usages de mysqldump

➤ *Sauvegarder des BD*

```
shell> mysqldump [options] --databases DB1 [DB2 DB3...]
```

➤ *Sauvegarder toutes les BD*

```
shell> mysqldump [options] --all-databases
```

Affichage à l'écran ou redirection dans un fichier

```
shell> mysqldump -uroot -p nomBD
```

La commande affiche le code SQL de la BD nomBD.

```
shell> mysqldump -uroot -p nomBD > nomFichier
```

La commande écrit le code SQL de la BD nomBD dans le fichier nomFichier.

Usage courant de MYSQLDUMP pour sauvegarder et recharger une BD

Commande usuelle de sauvegarde d'une BD :

```
shell> mysqldump --opt nomBD > backupFile.sql
```

L'option --opt inclut l'option --quick et l'option --lock-tables, entre autres.

Pour recharger le fichier de sauvegarde :

```
shell> mysql nomBD < backupFile.sql
```

Usage courant de MYSQLDUMP pour sauvegarder toutes les BDs

Commande usuelle de sauvegarde de toutes les BD :

```
shell> mysqldump --opt --all-databases > allBD.sql
```

Pour recharger le fichier de sauvegarde de toutes les BD :

```
shell> mysql < allBD.sql
```

mysql > SELECT ... INTO OUTFILE

La commande :

```
SELECT attributs INTO OUTFILE 'nomFichier'  
FROM etc.
```

Permet de sauvegarder le résultat d'un select dans un fichier texte.
Cela permet de réaliser des dumps rapides des tables
C'est le complément de LOAD DATA INFILE.

```
mysql > LOAD DATA INFILE nomFichier INTO TABLE nomTableOUTFILE
```

<http://dev.mysql.com/doc/refman/5.0/fr/load-data.html>

La commande :

```
LOAD DATA INFILE 'nomFichier' INTO TABLE nomTableOUTFILE
```

Permet charger les données d'un fichier texte dans une table.

Avec les options REPLACE ou IGNORE, quand une donnée rentre avec une clé primaire existant déjà, elle remplacera l'ancienne ou elle sera ignorée. Sinon, l'opération s'arrête en erreur.

```
LOAD DATA INFILE 'nomFichier' REPLACE INTO TABLE  
nomTableOUTFILE
```

4. Organisation physique des données

Le répertoire DATADIR

```
SHOW VARIABLES like '%data%'
```

ou

```
SHOW VARIABLES like 'datadir'
```

Permet de voir le répertoire où se trouvent les données de la BD.

Organisation des données

Chaque BD est rangée dans le DATADIR dans un répertoire du nom de la BD.

Chaque table d'une BD MyISAM est stockée sur disque dans trois fichiers.

Les fichiers portent le nom de la table, et ont une extension qui spécifie le type de fichier.

Le fichier .frm stocke la définition de la table.

Le fichier des données possède l'extension .MYD (MYData).

Le fichier d'index possède l'extension .MYI (MYIndex),

Pour spécifier explicitement que vous souhaitez une table MyISAM, indiquez le avec l'option ENGINE ou TYPE lors de la création de la table.

MySQL crée toujours un fichier .frm pour stocker le type de la table et les informations de définition. Les données et les index de la table peuvent être stockés ailleurs, en fonction du type de tables.