TP4 – Administration et Vues - Correction

La commande **show databases** permet d’afficher les bases de données dans le serveur MySQL. En particulier, la base mysql – qui s’installe par défaut lors de l’installation de MySQL- contient les utilisateurs et leurs privilèges.

Utiliser cette base par la commande : use mysql;

1. Créer l’utilisateur **local** qui se connecte à partir de son ordinateur contenant le serveur mysql en utilisant le mot de passe : « **localpwd** »,

create user local@localhost identified by 'localpwd';

flush privileges;

1. Créer l’utilisateur **public** (mot de passe : « **publicpwd** ») qui se connecte depuis n’importe quelle machine en réseau,

create user public identified by 'publicpwd';

flush privileges;

1. Permettre à n’importe quel utilisateur de se connecter en local,

create user **''**@localhost;

flush privileges;

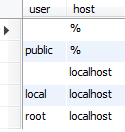
1. Permettre à n’importe quel utilisateur de se connecter de n’importe quelle machine en réseau.

create user **''**;

flush privileges;

1. Lister tous les utilisateurs.

select user, host from user;



1. Renommer l’utilisateur local en private et modifier son mot de passe,

rename user local@localhost to private@localhost;

flush privileges;

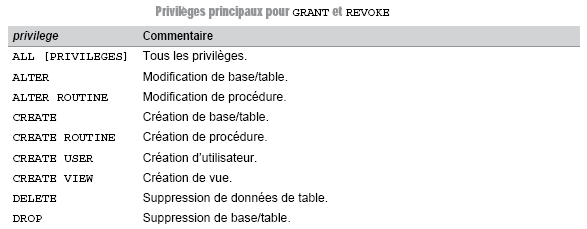
set password for 'private'@'localhost' = 'NewPassWD';

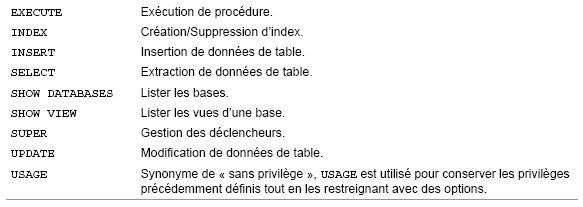
flush privileges;

Les commandes GRANT et REVOKE permettent de donner et de retirer des droits sur différents niveaux :

* Niveau global : droits sur toutes les bases d'un serveur (\*.\*)
* Niveau base de données : droits pour toutes les tables d'une base (base.\*)
* Niveau table : droits sur toutes les colonnes d'une table
* Niveau colonne : droits restreints à certaines colonnes d'une table

Quelques exemples de types de privilèges :





Les droits sont immédiatement opérationnels.

1. En examinant les tables de la base mysql, dire où est ce que sont enregistrés les différents privilèges : niveau global, niveau bases de données, niveau table et niveau colonne.
   * La table user de la base mysql stockant les prérogatives au **niveau global** :

select user, host, Create\_priv, Drop\_priv, Alter\_priv,

max\_questions,max\_updates, max\_connections, max\_user\_connections

from user;

* + La table mysql.db décrit les prérogatives au **niveau *database***.

select user, host, db, create\_priv, drop\_priv, alter\_priv from db;

* + La table mysql.tables\_priv décrit les prérogatives objet au **niveau *table***. Ainsi la colonne Table\_name indique la table concernée, la colonne Grantor précise l’utilisateur ayant donné le droit. La colonne Table\_priv est un SET contenant la liste des droits de l’utilisateur sur la table.

select concat(user, '@', host) 'Compte', concat(db, '.', Table\_name) 'Objet', Grantor, Table\_priv

from tables\_priv;

* + La table mysql.columns\_priv décrit les prérogatives objet au niveau *column*. Ainsi la colonne Table\_name indique la table concernée, la colonne Column\_name précise la colonne concernée par le droit. La colonne Column\_priv est un SET contenant la liste des droits de l’utilisateur sur la colonne de la table.

select concat(user, '@', host) 'Compte', concat(db, '.', Table\_name) 'Objet', column\_name, column\_priv

from columns\_priv;

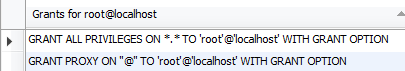
1. A quoi sert la commande show privileges ;

La commande **show privilleges ;** affiche tous les droits d’exécuter une instruction SQL (privilège système), et les droits relatifs aux données des tables situées dans différentes bases (privilège objet).

1. A quoi sert la commande show grants for user ; Vérifier les privilèges de l’utilisateur root.

La commande show grants for user affiche les différentes instruction grants pour un utilisateur particulier ;

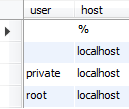
show grants for 'root'@'localhost';



1. Supprimer l’utilisateur public,

drop user public;

select user, host from user;



L’utilisateur public a été supprimé.

On donne la situation suivante :

User : Saleh (localhost)

User : Ali (localhost)

Create

Insert, Select (Pilote, nbvols, compagnie)

Create, Drop

Create

Drop

Alter, Insert, Update, Delete

bdvols

bdbiblio

Livre (IdLivre, Titre, ISBN, Prix, #NumAuteur)

Auteur (NumAuteur, NomPrenom, Adr, Tel)

Emprunteur (NumEmprunteur, NomComplet, Adr)

Emprunt (IdEmprunt, DateEmprunt, DateRetour, #IdLivre, #NumEmprunteur)

bdenig

Pilote (IdPilote, Pilote, nhvols, #Comp)

Compagnie (Compa, compagnie, adresse)

1. En utilisant les commandes create user et grant, essayer de mettre en pratique la figure ci-dessus,

create user Ali@localhost;

create user Saleh@localhost;

flush privileges;

L’utilisaeur root crée les bases de données et attribue les différents droits aux utilisateurs:

create database bdbiblio;

create database bdenig;

create database bdvols;

grant create, drop, alter, insert, update, delete on bdbiblio to 'Ali'@'localhost';

grant create, drop on bdenig to Ali@localhost;

grant create, insert on bdvols to Saleh@localhost;

grant select(Pilote, nbhVol) on bdvols.Pilote to Saleh@localhost;

grant select(compagnie) on bdvols.Compagnie to Saleh@localhost;

1. Saleh ne peut lancer, chaque heure, que 100 select, 10 updates, se connecter 5 fois dont 2 simultanément sur la base bdvols.

Alter user 'Saleh'@'localhost'

with max\_queries\_per\_hour 100

max\_updates\_per\_hour 10

max\_connections\_per\_hour 5

max\_user\_connections 2;

1. Ali peut donner les privilèges create et drop à Saleh.

grant create, drop on bdenig to Ali@localhost **with grant option**;

1. Saleh n’est plus limité en requêtes select et update sur la base bdvols,

Alter user 'Saleh'@'localhost'

with **max\_queries\_per\_hour 0**

**max\_updates\_per\_hour 0**;

On considère la base de données suivante :

Pilote(NumPilote, NomPil, PrenomPil, Adresse, Salaire, Prime)

Avion (NumAvion, NomAv, Capacité, Localisation)

Vol (NumVol, #NumPil, #NumAv, DateVol, HeureDepart, HeureArrivee, VilleDepart, VilleArrivee)

On suppose qu’un vol est effectué par un unique pilote sur un avion.

1. Créer une table virtuelle pour éviter que certains utilisateurs aient accès aux salaires et prime des pilotes,

Create or replace view Restrict\_Pilote

As select NumPilote, NomPil, PrenomPil, Adresse

From Pilote ;

1. Créer une vue (NumPil, Nom, Charge) pour consulter la charge horaire des pilotes,

Create or replace view Charge\_hor (NumPil, Nom, Charge)

As select p.NumPilote, NomPil, sum(HeureArrivee – HeureDepart)

From Pilote p, Vol

Where p.NumPilote = Vol.NumPil

Group by p.NumPil ;

1. Formuler une requête pour afficher les pilotes tunisiens dont la charge excède un seuil de 40 heures,

Select \* from Charge\_hor c, Pilote p

Where c.NumPil = p.NumPilote and Charge > 40 and Adresse like ‘%Tunisie%’ ;

1. Définir une vue permettant de consulter les vols des pilotes habitant Tunis,

Create view VolPil\_Tunis

As select \*

From Vol

Where NumPil in (select NumPilote from Pilote where Adresse like ‘%Tunis%’) ;

Il est possible de mettre à jour la table vol à travers cette vue, mais l’opération de mise à jour peut concerner n’importe quel vol (sans aucune condition)

Par exemple, supposons que le pilote n° 100 habite à Gabes, l’insertion suivante sera réalisée dans la table vol à traves la vue, mais le tuple ne pourra pas être visible en consultant la vue :

Insert into VolPil\_Tunis (NumVol, NumPil, NumAv, VilleDepart)

Values (120, 100, 15, ‘Gabes’) ;

L’ajout de la clause with check option à la création de la vue précédente interdira toute opération de mise à jour sur les vols qui ne sont pas assurés par des pilotes habitant Tunis et l’insertion d’un vol assuré par le pilote 100 échouera.

Create view VolPil\_Tunis

As select \*

From Vol

Where NumPil in (select NumPilote from Pilote where Adresse like ‘%Tunis%’)

With Check Option ;

1. Définir une vue sur Pilote ne permettant l’insertion d’un pilote que si’il a le salaire entre 2000 et 5000 Dinars

Create view PiloteView

As select \* from Pilote

Where Salaire between 2000 and 5000

With check option ;

Ainsi, l’insertion suivante va échouer :

Insert into PiloteView (NumPilote, Salaire) values (100, 1500) ;

1. Définir une vue sur vol permettant de vérifier les contraintes d’intégrité référentielle en insertion et en modification. C'est-à-dire ne permettant d’ajouter ou de modifier que les vols des pilotes et avions existants.

Create view InsertUpdateVol

As select \* from Vol

Where NumPil in (select NumPilote from Pilote)

And NumAv in (select NumAvion from Avion)

With Check Option ;