SQL : Langage de contrôle d'accès aux données :

Cas de MySQL

Dr N. BAME

Plan

- Gestion des utilisateurs
  - Création, modification, suppression, …
- Gestion des privilèges
  - Attribution, révocation, …

Introduction

- Comme dans tout système **multi-utilisateur**, l'usager d'un SGBD **doit être identifié** avant de pouvoir **utiliser des ressources**.
- Les **accès** aux informations et à la base de données **doivent être contrôlés** à des fins de **sécurité et de cohérence**.

- **Objectifs** :
  - la **gestion des utilisateurs** qui **manipuleront** des **bases de données** dans lesquelles **se trouvent des objets** tels que des tables, index, séquences, vues, procédures, etc.;
  - la **gestion des privilèges** qui permettent de **donner/retirer des droits sur la base** de données (**privilèges système**) et sur les données de la base (**privilèges objet**) ;

Gestion des utilisateurs

- Un **utilisateur** (user) est **identifié** par MySQL par **son nom et celui de la machine** à partir de laquelle **il se connecte**.

- Il pourra **accéder** **à différents** **objets** (tables, vues, séquences, index, procédures, etc.) d'une ou de plusieurs bases **sous réserve d'avoir reçu** un certain nombre de **privilèges**

Les types d'utilisateurs

Les types d'utilisateurs, leurs fonctions et leur nombre **peuvent varier** d'une base à une autre :

Généralement, on peut classifier les utilisateurs de la manière suivante :

- **Le DBA (DataBase Administrator)**
  - Il en existe au moins un.
  - Une base importante peut en regrouper plusieurs qui se partagent les tâches suivantes :
    - **installation et mises à jour** de la base et des outils éventuels ;
    - gestion de **l'espace disque** et des **espaces pour les données** ;
    - **gestion des utilisateurs** et de leurs **objets**
    - **optimisation** des performances ;
    - **sauvegardes**, **restaurations** et archivages ;
- **L'administrateur réseau** (qui peut être le DBA)
  - se charge de la **configuration des couches** client pour les **accès distants**.

Les types d'utilisateurs

- **Les développeurs**
  - qui **conçoivent** et **mettent à jour** la base.
  - Ils peuvent aussi agir sur leurs objets (création et modification des tables, index, etc.).
  - Ils transmettent au DBA leurs demandes spécifiques (stockage, optimisation, sécurité).
- **Les administrateurs d'application**
  - qui gèrent les **données manipulées par la ou les applications**.
  - Pour les petites et les moyennes bases, le DBA joue ce rôle.
- **Les utilisateurs**
  - qui se connectent et interagissent avec la base à travers les applications ou à l'aide d'outils (interrogations pour la génération de rapports, ajouts, modifications ou suppressions d'enregistrements).

Tous seront des utilisateurs (au sens MySQL) avec des privilèges différents.

Création d'un utilisateur : CREATE USER

- Pour pouvoir créer un utilisateur, **vous devez posséder le privilège CREATE USER** ou **INSERT** sur la **base système mysql**
  - car c'est la table **mysql.user** qui stockera l'existence de ce nouvel arrivant

Syntaxe

```
CREATE USER  nomCompletUtilisateur
[IDENTIFIED BY  'motdePasse'];
```

- **IDENTIFIED BY motdePasse** permet d'affecter un mot de passe (16 caractères maximum,sensibles à la casse) à un utilisateur (16 caractères maximum, sensibles aussi à la casse).
- **nomCompletUtilisateur** : **'username'@'hote'**
  - si on met le joker **%** dans **hôte**, alors l'utilisateur peut se connecter de **plusieurs hôtes**
  - si **on ne précise pas** la hôte, c'est **%** **par défaut**.

Création d'utilisateur : exemple

```
CREATE USER  'bouba'@'localhost'
[IDENTIFIED BY 'Passer5423' ];
```

- bouba est déclaré « utilisateur à **accès local** » (localhost),
- il devra se connecter à l'aide de son mot de passe (Passer5423).

Remarques

1. Il est possible de créer un utilisateur **sans mot de passe**
   - **Il ne faut jamais le faire**

   ```
   CREATE USER  'ndiaga'@'%';
   ```

2. **Par défaut**, un utilisateur, une **fois créé, n'a aucun droit** sur aucune base de données
   - à part en **lecture écriture** sur la base **test** et en **lecture** seule sur la base **information_schema**.

Un utilisateur bien connu : root

- **Lors de l'installation**, vous avez dû noter la présence de l'utilisateur **root** (mot de passe saisi à l'installation).
- Cet utilisateur **est le DBA** que MySQL vous offre.
- Il vous permettra d'effectuer vos **tâches administratives** en ligne de commande ou par une interface graphique

Liste des utilisateurs

- Les informations sur les utilisateurs se retrouvent dans la table **user** de la base **mysql** (**mysql.user**).
- L'extraction des colonnes **user** et **host** restitue la **liste des utilisateurs** **connus du serveur**.

```
SELECT user, host
FROM user;
```

Création d'utilisateur

- Il apparaît **quatre** **accès potentiels**.
- L'utilisateur **vide** **''** correspond à une **connexion anonyme**.
- La **machine** désignée par « **%** » indique que la **connexion est autorisée à partir de tout site**
  - en supposant qu'un client MySQL est installé et qu'il est relié au serveur par TCP-IP
- La **machine** désignée par « **localhost** » spécifie que la **connexion est autorisée en local**.

Modification d'un utilisateur

Changer le mot de passe

- Le **mot de passe** d'un utilisateur **peut être modifié** sans parler de privilèges.
- Pour changer un mot de passe, il faut donc modifier la table user par la seule **commande SQL** : **UPDATE**.
- **Exemple**

```
UPDATE mysql.user
SET Password = PASSWORD('pwd321')
WHERE User = 'bouba'
AND   Host = 'localhost';
FLUSH PRIVILEGES;
```

  - Notez l'utilisation de la fonction **PASSWORD()** qui **code** le mot de passe à affecter à la colonne Password de la table user.
  - Il est plus prudent d'utiliser ensuite **FLUSH PRIVILEGES** qui **recharge les tables système** de manière à rendre la manipulation effective sur l'instant (un peu comme un COMMIT sur des données).

Ou

```
SET PASSWORD FOR bouba '@ 'localhost ' = PASSWORD('pwd321');
FLUSH PRIVILEGES;
```

Renommer un utilisateur (RENAME USER)

- Pour pouvoir renommer un utilisateur, vous devez posséder le **privilège CREATE USER** (ou le **privilège UPDATE** sur la base de données **mysq**l).

Syntaxe :

```
RENAME USER nomUtilisateur TO nouveauNom ;
```

Exemple

```
RENAME USER 'bouba'@'localhost' TO 'modou'@'localhost' ;
```

Suppression d'un utilisateur : DROP USER

- Pour pouvoir supprimer un utilisateur, vous devez posséder le privilège **CREATE USER** (ou le privilège **DELETE** sur la base de données **mysql**).

Syntaxe

```
DROP USER 'nom_utilisateur'@'hote' ;
```

- Il faut spécifier **l'accès à éliminer** (**user@machine**).
- **Tous les privilèges** relatifs à cet **accès sont détruits**.
- Si l'utilisateur est **connecté** dans le même temps, sa **suppression ne sera effective qu'à** la fin de sa (dernière) session.

Remarque

- Aucune donnée d'aucune table que l'utilisateur aura mis à jour durant toutes ses connexions ne sera supprimée.
- Il n'y a pas de notion d'appartenance d'objets (tables, index, procédure, etc.) à un utilisateur.
  - Tout ceci est relatif à la base de données (database).

Exercice

- Afficher tous les utilisateurs du serveur : bd mysql
- Créer les utilisateurs suivants avec des mots de passe pour certains et sans mdp pour d'autres :
  - modou, ndiaga, demba (en accès local) ,
  - samba, anna et coumba (en accès réseau).
- Réafficher tous les utilisateurs du serveur
- Connectez-vous au serveur avec l'utilisateur ndiaga ;
  - Afficher les BDs; que remarquez-vous ?
  - Tenter de se connecter à la BD championnat ; que se passe-t-il? pourquoi ?
  - Modifier le mdp de ndiaga (soi-même).
  - Déconnexion/connexion avec ndiaga.

- Se connecter avec root et modifier le mdp de demba
- Renommer l'utilisateur demba en mademba
  - Réafficher tous les utilisateurs du serveur
- Supprimer l'utilisateur samba

Privilèges

Un **privilège** est :

- un **droit d'exécuter une certaine instruction** SQL
  - on parle de **privilège système**,
- ou un **droit relatif aux données des tables** situées dans différentes bases
  - on parle de **privilège objet**.

- La **connexion**, par exemple, sera considérée comme un **privilège système** bien que n'étant pas une commande SQL.

- Les **privilèges système** **diffèrent sensiblement** **d'un SGBD à un autre**.
- En revanche, on retrouvera les **mêmes privilèges objet**
  - **exemple** : autorisation de modifier la colonne *budget* de la table *Departement*

qui sont attribués ou retirés par les instructions **GRANT** et **REVOKE**.

Niveaux de privilèges

- **Global level** : privilèges **s'appliquant à toutes les bases du serveur.**
  - Ces privilèges sont stockés dans la table **mysql.user**
  - exemple d'attribution d'un privilège global :

    GRANT CREATE ON *.* …

- **Database level** : privilèges **s'appliquant à tous les objets d'une base de données** en particulier.
  - Ces privilèges sont stockés dans les tables **mysql.db** et **mysql.host**
  - exemple d'attribution d'un privilège database :

    GRANT SELECT ON championnat.* …

Niveaux de privilèges

- **Table level :** privilèges **s'appliquant à la globalité** d'une **table d'une base de données en particulier**.
  - Ces privilèges sont stockés dans la table **mysql.tables_priv**

exemple :

  - **GRANT INSERT ON championnat.gain**…

- **Column level** : privilèges **s'appliquant à une des colonnes d'une table** d'une base de données en particulier.
  - Ces privilèges sont stockés dans la table **mysql.columns_priv**

exemple :

GRANT UPDATE (nomSponsor) ON championnat.Sponsorise
…

Niveaux de privilèges

- ***.*** : **toutes les bases de données**
- ***** : **base de données courante**
- **nomBD.*** : **tous les éléments** de la base nomBD
- **nomBD.nomTable** : la **table nomTable** de la base nomBD.
- …

- La base ***mysql*** possède des tables permettant de **stocker les privilèges** des utilisateurs à différents niveaux :
  - ***db*** : privilèges au niveau des **bases de données**.
  - ***tables_priv*** : privilèges au niveau des **tables**.
  - ***columns_priv*** : privilèges au niveau des **colonnes**.

Remarque

- Tout comme on peut **préciser la table à laquelle appartient une colonne** en **préfixant** le nom de la colonne par le nom de la table, ***nom_table.nom_colonne***,
- il est possible de préciser à **quelle base de données appartient une table**, ***nom_bdd.nom_table***,
- ou de préciser **à la fois** la **table** et la **base de données** dans laquelle se trouve **une colonne** : ***nom_bdd.nom_table.nom_colonne***.

Attribution de privilèges

```
GRANT privilège [ (col1 [, col2...])] [,privilège2 ... ]
ON {nomTable | * | *.* | nomBase.*}
TO utilisateur [,utilisateur2 ...]
[ WITH  GRANT OPTION ]   ;
```

Privilège

- SELECT
- INSERT, UPDATE, DELETE
- CREATE { TABLE, VIEW, USER }
- ALTER, DROP

WITH GRANT OPTION
permet à un utilisateur de passer à un autre tout privilège dont il a accès.

Affecter **tous les privilèges** : **ALL [PRIVILEGES]**

```
GRANT  ALL
ON  *
TO  'modou'@'localhost' ;
```

Exemple

```
GRANT ALL PRIVILEGES ON gescom.*
TO 'moussAdmin'@'localhost';
```

```
CREATE USER ‘samba'@'localhost' IDENTIFIED BY ‘bathie';
```

```
GRANT SELECT, UPDATE (nom, quantite, categorie),
DELETE, INSERT ON ventes.produit TO 'samba'@'localhost';
```

Révocation de privilèges à un utilisateur

```
REVOKE privilège [,privilège2 ... ]
ON {nomTable | * | *.* | nomBase.*}
FROM utilisateur [,utilisateur2 ...]
```

Exemple

```
REVOKE DELETE, UPDATE ON ventes.produit
FROM ‘samba'@'localhost' ;
```

Exercice

- Attribuer les privilèges d'interrogation dans la base championnat à ndiaga.
- Se connecter avec ndiaga
  - Sélectionner la base championnat, puis, Interrogation des tables
  - Tenter d'augmenter l'âge des joueurs

- Avec root, attribuer le privilège de mises à jour à ndiaga sur la table joueur
- Avec ndiaga, tenter à nouveau d'augmenter l'âge des joueurs, puis tenter d'insérer un nouvel enregistrement

- …