Bloc1 Données	**Chapitre 4** **Sécuriser les accès à une base de données**

Objectifs :

Créer un compte utilisateur

Supprimer un compte utilisateur

Donner un privilège à un compte utilisateur

Enlever un privilège à un compte utilisateur

Identifier les privilèges d'un compte

Sommaire

Fiche SQL utile pour ce chapitre :

SQL3-Gestion-des-utilisateurs.docx

I. POURQUOI SÉCURISER LES ACCÈS À UNE BASE DE DONNÉES

Pour demander à un SGDB de délivrer de l'information ou de gérer de l'information, il faut s'identifier.

Les accès aux informations d'une base de données doivent être contrôlés pour des raisons de **sécurité** et de **cohérence des données**.

Il est donc nécessaire de :

- Gérer les comptes des utilisateurs (identification, authentification)
- Gérer les privilèges de ces comptes (ce qu'ils ont le droit de faire dans la base de données)

1) Les utilisateurs

On peut avoir différents utilisateurs :

- L'administrateur de la base de données qui va gérer la structure de la base de données, les utilisateurs de la base de données et l'octroi ou la révocation de leurs privilèges, l'optimisation des performances, les sauvegardes et les restaurations en cas de problème.
- Les développeurs qui peuvent agir sur les objets de la base de données (création d'enregistrements, modification d'enregistrements …). Attention dans certaines entreprises les développeurs doivent s'adresser au DBA pour agir sur les bases de données !!
- Les utilisateurs qui peuvent accéder directement à la base de données pour réaliser des requêtes en lecture
- Les responsables de l'application qui peuvent en cas de problème intervenir sur la base de données
- Les applications qui accèdent à la base de données

2) Les privilèges

Après avoir créé un compte, vous devez indiquer ce qu'il pourra faire sur les bases de données, sinon la personne n'aura aucun droit, à part en lecture écriture sur la base de test et en lecture sur la base information_schema.

Voici les possibilités d'accès :

- le compte peut accéder à toutes les bases ou à certaines bases nommées explicitement
- le compte peut ou non intervenir sur la structure de la base (créer des tables, supprimer des tables, etc.)
- le compte peut ou non gérer les valeurs enregistrées dans les tables (ajouter des enregistrements, en modifier, en supprimer)
- le compte peut ou non lire les valeurs enregistrées dans les tables (voir les valeurs de toutes les tables ou de certaines tables, voir les valeurs de tous les attributs ou de certains attributs)

Remarques très importantes :

- **les comptes utilisateurs créés et les privilèges qui leur sont donnés dépendent de la politique de sécurité de l'entreprise qui gère la base de données : il est donc nécessaire de vous renseigner avant de créer des comptes et leurs privilèges.**

- **les privilèges peuvent différer d'un SGBD à l'autre**.

II. GESTION DES COMPTES AVEC LE SGBD MARIADB

1) Pour créer un compte

```
CREATE USER 'nomcompte'@'machine'  IDENTIFIED by 'motdepasse' ;
```

On crée un compte en spécifiant la machine à partir de laquelle le compte a le droit de se connecter.

Exemples :

CREATE USER justinof@172.15.5.8 …..
CREATE USER justinof @localhost …..
CREATE USER appliex@'%' …..

Même si le mot de passe est facultatif, il est fortement conseillé de donner un mot de passe solide (au moins 12 caractères et 4 types différents : des minuscules, des majuscules, des chiffres et des caractères spéciaux).

Consulter le lien suivant de la CNIL pour définir un bon mot de passe :
https://www.cnil.fr/fr/les-conseils-de-la-cnil-pour-un-bon-mot-de-passe

2) Pour changer le mot de passe d'un compte

```
ALTER USER 'nomcompte'@'machine'  IDENTIFIED by 'motdepasse' ;
```

Attention : si un utilisateur dispose d'un même nom de compte mais qu'il peut se connecter à partir de machines différentes, il faudra réaliser un ordre « ALTER USER » pour chacune des machines pour changer le mot de passe du compte.

3) Supprimer un compte

```
DROP USER 'nomcompte'@'machine' ;
```

Attention : si un utilisateur dispose d'un même nom de compte mais qu'il peut se connecter à partir de machines différentes, il faudra réaliser un ordre « DROP USER » pour chacune des machines.

4) Consulter les comptes créés

Les comptes sont créés dans une table de la base système nommée mysql.

Dans cette base de données, les six tables suivantes permettent à MariaDb d'enregistrer les utilisateurs et leurs privilèges : user, db, host, mysql.tables_priv,

columns_priv, procs_priv Pour visualiser les comptes créés, faites une requête sur la

```
SELECT host, user, password FROM mysql.user;
```

III. GESTION DES PRIVILÈGES AVEC LE SGBD MARIADB

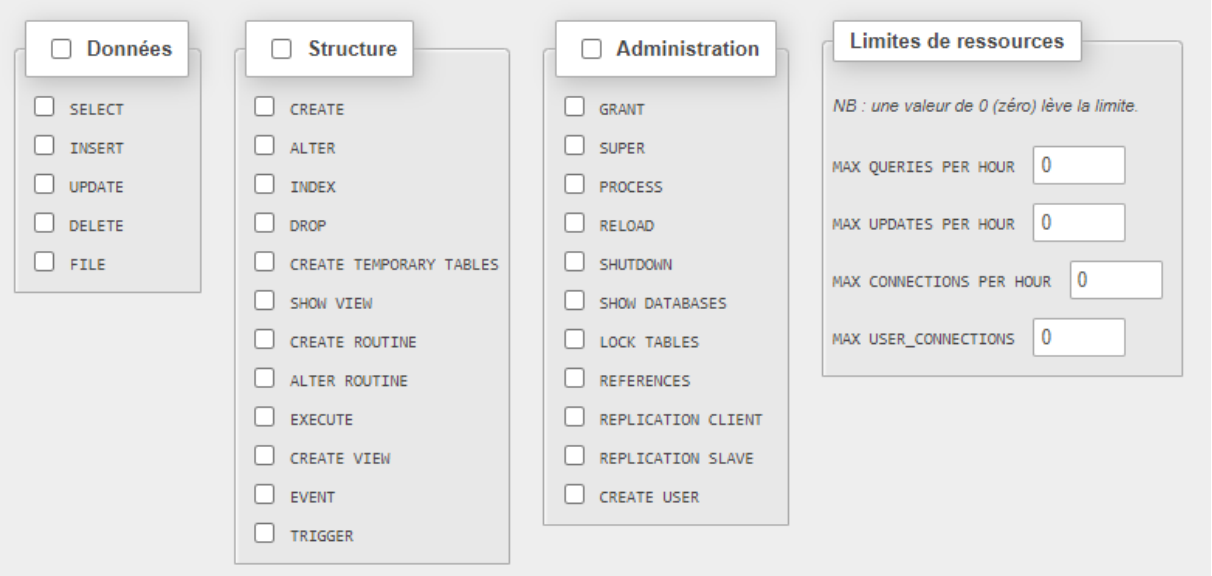
1) Donner des privilèges

Tout ce qu'un compte utilisateur a le droit de faire doit être explicitement indiqué par la commande GRANT.

Attention : **ce qui n'est pas autorisé avec la commande GRANT est interdit**

```
GRANT privilège[(col1[, col2…])]  [ , privilège2… ]
ON [ { TABLE | FUNCTION | PROCEDURE } ]
    {nomTable | * | *.* | nombase.*}
TO nomDeCompte
[ WITH [GRANT OPTION]
[ MAX_QUERIES_PER_HOUR nb1]
[ MAX_UPDATES_PER_HOUR nb2]
[ MAX_CONNECTIONS_PER_HOUR nb3]
[ MAX_USER_CONNECTIONS nb4]
```

- les [ ] signifient que ce qui est entre les [ ] est facultatif
- { val1 | val2 | val3 } signifie qu'il faut indiquer val1 ou val2 ou val3
- privilèges possibles pouvant être donné à l'utilisateur :



- col précise la ou les colonnes sur lesquelles portent les privilèges select,insert ou update. Par exemple update(nomClient) n'autorisera que la modification du nom du client

- **WITH GRANT OPTION : permet à l'utilisateur de transmettre les privilèges qu'il a reçu à une autre personne**

Les privilèges(droits) peuvent être donnés à plusieurs niveaux ;

Niveau global : privilège s'appliquant à toutes les bases du serveur
Grant create on *.* to util1@localhost

Niveau base de données : privilège pour tous les objets d'une base de données.
Grant select on **BdExemp.*** to util2@localhost

Niveau table : privilège pour une table d'une base de données
Grant select on **BdExemp.Client** to util3@localhost

Niveau colonne : privilège sur une colonne d'une table d'une base de données
Grant select**(nomClient)** on **BdExemp.Client** to util4@localhost

Niveau procédure : privilège sur les procédures cataloguées (create, execute …) d'une base de données
Grant execute on procedure **BdExemp.procedure01** to util5@localhost

Attention :

- Tout ce qu'un utilisateur a le droit de faire doit être explicitement indiqué par la commande GRANT.
- **ce qui n'est pas autorisé avec la commande GRANT est donc interdit**

2) Consulter les privilèges d'un compte

SHOW GRANTS FOR *nomUtilisateur*

3) Enlever les privilèges à un utilisateur

```
REVOKE CREATE
ON BdRh.*
FROM util11@localhost

REVOKE ALTER, INSERT, UPDATE
ON BdRh.Salarié
FROM util12@localhost
```

Vous pouvez utiliser une instruction qui révoque tous les droits en une seule fois :