SQL – Privilèges

Les privilèges correspondent aux droits, aux autorisations dont dispose un utilisateur donné.

Il existe de nombreux privilèges dont voici quelques exemples courants (liste exhaustive disponible sur le site de la documentation officielle MySQL) :

- `CREATE` : création de DB, tables ou index
- `DROP` : suppression de DB, tables ou index
- `ALTER` : modification de tables
- `DELETE` : suppression de tables
- `SELECT` : sélection de tables ou de colonnes
- `INSERT` : insertion de tables ou de colonnes
- `UPDATE` : mise à jour de tables ou de colonnes

SQL – Privilèges

Pour accorder un privilège particulier, il faut préciser le domaine d'application de ce privilège :

- **`nom_table`** : à la table mentionnée de la DB courante
- **`nom_db.nom_table`** : à la table mentionnée de la DB mentionnée
- **`nom_db`** : à la DB mentionnée
- **`nom_db.*`** : à tous les éléments de la DB mentionnée
- **`*.*`** : à tous les éléments de toutes les DB
- **`*`** : à tous les éléments de la DB courante si elle a été spécifiée, sinon, équivalent à **`*.*`**

Rôles

Les rôles sont des groupes de privilèges.

Attention, ils ne sont disponibles qu'à partir de la version 8 de MySQL.

Tous les utilisateurs associés à un rôle héritent de ses privilèges. Ce mécanisme facilite les modifications.

Rôle – Création

La requête de création d'un rôle est la suivante :

```
CREATE ROLE [IF NOT EXISTS] role1 [, role2];
```

Par exemple :

```
CREATE ROLE IF NOT EXISTS 'secretaire'@'localhost',
'magasinier'; -- @'%' par défaut
```

Rôle – Suppression

La requête de suppression d'un rôle est la suivante :

```
DROP ROLE [IF EXISTS] role1 [, role2];
```

Par exemple :

```
DROP ROLE IF EXISTS 'magasinier';
```

Privilèges – Affichage

La commande qui permet d'afficher les privilèges est la suivante (rôle si disponible dans la version de MySQL) :

```
SHOW GRANTS [FOR nom_login_ou_role];
```

SQL – Privilèges

Il existe deux commandes pour gérer les privilèges :

- **`GRANT`** : pour accorder un privilège
- **`REVOKE`** : pour retirer un privilège

Privilèges – Ajout

La commande générique d'ajout d'un privilège est la suivante :

```
GRANT privilège [(col1, col2,…)]
ON [type_élément] type_domaine
TO nom_login [IDENTIFIED BY mot_de_passe];
```

Si le compte utilisateur existe, on lui accorde les privilèges.

S'il n'existe pas, on commence par créer celui-ci avant de lui accorder les privilèges.

Privilèges – Ajout

En version ≥ 8 :

```
GRANT privilège [(col1, col2,…)]
ON [type_élément] type_domaine
TO nom_login_ou_role;
```

Privilèges – Ajout

La commande générique d'ajout de plusieurs privilèges est la suivante :

```
GRANT privilège1 [(col1, col2,…)]
, privilège2 [(col1, col2,…)], …]
,…
ON [type_élément] type_domaine
…
```

Privilèges – Ajout

En version de MySQL <8, la procédure est d'attribuer des privilèges à chaque utilisateur.

En version de MySQL ≥8, la procédure est d'attribuer des privilèges à des rôles et ensuite de donner ces rôles aux utilisateurs.

```
CREATE USER [IF NOT EXISTS] nom_login
DEFAULT ROLE role1 [, role1] …
```

Et si l'utilisateur existe :

```
ALTER USER [IF EXISTS] nom_login
DEFAULT ROLE {NONE | ALL | role1 [, role2] …};
```

Privilèges – Ajout

Exemple

Nous voulons accorder les droits de sélection et d'insertion sur toutes les colonnes de la table `client` à l'utilisateur `secretaire1` qui n'existe pas :

```
GRANT SELECT, INSERT
ON TABLE clicom.client
TO 'secretaire1'@'localhost'
IDENTIFIED BY 'pw_secretaire1';
```

Vérifions ensuite ce qui a été ajouté dans la table **`information_schema.table_privileges`** :

```
SELECT *
FROM information_schema.table_privileges;
```

Privilèges – Ajout

Exemple

Nous voulons accorder les droits de mise à jour sur la colonne `qstock` de la table `produit` à l'utilisateur `magasinier1` :

```
GRANT UPDATE(qstock)
ON clicom.produit
TO 'magasinier1'@'localhost'
IDENTIFIED BY 'pw_mag1';
```

Vérifions ensuite ce qui a été ajouté dans la table **`information_schema.column_privileges`** :

```
SELECT *
FROM information_schema.column_privileges;
```

Privilèges – Ajout

En résumé, pour visualiser les privilèges, nous pouvons utiliser les tables suivantes de la DB `information_schema` :

- `schema_privileges` : tous les privilèges sur l'entièreté d'une DB
- `table_privileges` : tous les privilèges sur l'entièreté d'une table
- `column_privileges` : tous les privilèges sur une colonne d'une table
- `user_privileges` : tous les privilèges d'un utilisateur à donner ses privilèges (ou `USAGE`)

Privilèges – Ajout

Il est également possible de donner tous les privilèges (sauf le droit de donner ses privilèges) sur un élément grâce au mot-clé **`ALL`** :

```
GRANT ALL [PRIVILEGES]
ON [type_élément] type_domaine
TO nom_login [IDENTIFIED BY mot_de_passe];
```

Privilèges – Ajout

Il est également possible de modifier un mot de passe sans modifier les privilèges à l'aide de **`USAGE`** :

```
GRANT USAGE
ON *.* -- toujours sur tous les éléments
TO nom_login IDENTIFIED BY mot_de_passe;
```

Privilèges – Ajout

Il est également possible de donner le droit de donner ses privilèges grâce à l'option **`GRANT OPTION`** :

```
GRANT GRANT OPTION
ON [type_élément] type_domaine
TO nom_login [IDENTIFIED BY mot_de_passe];
```

```
GRANT …
ON [type_élément] type_domaine
TO nom_login [IDENTIFIED BY mot_de_passe]
WITH GRANT OPTION;
```

Privilèges – Suppression

La commande générique de suppression d'un privilège est la suivante :

```
REVOKE privilège
ON [type_élément] type_domaine
FROM nom_login;
```

Privilèges – Suppression

Exemple

Retirer le privilège d'insertion dans la table client à l'utilisateur `secretaire1` :

```
REVOKE INSERT
ON clicom.client
FROM 'secretaire1'@'localhost';
```

Privilèges – Exercices

Exercice 1

Vous devez :

- Créer les utilisateurs suivants (en `localhost`) :
  - `secretaire1`
  - `directeur1`
  - `directeur2`
  - `comptable1`
  - `magasinier1`
  - `magasinier2`
- Leur donner des privilèges pertinents sur la DB `clicom`
- Ensuite enlever tous les privilèges de l'utilisateur `directeur2`
- Supprimer l'utilisateur `magasinier2`

Privilèges – Exercices

Exercice 2

Vous devez :

- Créer une nouvelle DB avec quelques tables
- Insérer quelques données dans chaque table
- Créer un utilisateur pour chaque fonctionnalité suivante (en `localhost`) :
  - administration sur cette DB
  - création des tables dans cette DB
  - insertion et suppression de données dans toutes les tables
  - mise à jour de données dans toutes les tables
  - sélection de données dans toutes les tables
- Tester ce que chaque utilisateur peut réellement faire et consigner cela dans un tableau