

Cours : Gestion des utilisateurs SQL (MySQL / MariaDB)

1 Concepts clés

- **Utilisateur SQL** : un compte qui permet de se connecter à la base de données.
- **Hôte** : chaque utilisateur est lié à un hôte (localhost, % pour tout réseau, ou une IP spécifique).
Exemple : 'alk'@'localhost'.
- **Mot de passe** : utilisé pour authentifier l'utilisateur.
- **Privilèges / permissions** : droits sur les bases, tables, colonnes, etc.
- **Plugin d'authentification** : détermine comment MySQL/MariaDB authentifie l'utilisateur (ex: mysql_native_password, unix_socket).

2 Liste des utilisateurs existants

```
SELECT User, Host, plugin FROM mysql.user;
```

- User → nom de l'utilisateur - Host → d'où il peut se connecter - plugin → type d'authentification

3 Créer un nouvel utilisateur

Avec mot de passe

```
CREATE USER 'nom_utilisateur'@'localhost' IDENTIFIED BY 'MotDePasse';
```

Avec mot de passe vide (uniquement dev local)

```
CREATE USER 'alk'@'localhost' IDENTIFIED BY '';
```

4 Modifier un utilisateur

- **Changer le mot de passe :**

```
ALTER USER 'alk'@'localhost' IDENTIFIED BY 'NouveauMotDePasse';
```

- **Changer le plugin d'authentification :**

```
ALTER USER 'alk'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'MotDePasse';
```

- Renommer un utilisateur :

```
RENAME USER 'ancien_nom'@'localhost' TO 'nouveau_nom'@'localhost';
```

5 Supprimer un utilisateur

```
DROP USER 'alk'@'localhost';
```

6 Attribuer des privilèges (permissions)

Exemple de base : droits sur une base entière

```
GRANT ALL PRIVILEGES ON nom_base.* TO 'alk'@'localhost';
```

- Pour un droit spécifique, par ex. lecture seule :

```
GRANT SELECT ON nom_base.* TO 'alk'@'localhost';
```

Appliquer les changements

```
FLUSH PRIVILEGES;
```

7 Révoquer des privilèges

```
REVOKE ALL PRIVILEGES ON nom_base.* FROM 'alk'@'localhost';
```

Ou juste certains droits :

```
REVOKE INSERT, UPDATE ON nom_base.* FROM 'alk'@'localhost';
```

8 Mot de passe et sécurité

- Éviter les mots de passe vides sur un serveur réel.

- Préférer un utilisateur dédié par projet plutôt que `root`.
- `root` local utilise souvent `unix_socket` sur Debian → pas de mot de passe, connexion seulement depuis le serveur.
- Pour PDO et applications PHP, créer un **utilisateur spécifique** avec mot de passe.

9 Vérifier les privilèges

```
SHOW GRANTS FOR 'alk'@'localhost';
```

10 Bonnes pratiques

1. Créer un **utilisateur par projet**.
2. Ne jamais donner `ALL PRIVILEGES` à `root` pour les applications.
3. Limiter l'accès à l'hôte nécessaire (`localhost` pour local, IP spécifique pour distant).
4. Utiliser des mots de passe forts.
5. Révoquer les droits inutiles.