

TP 06 - SQL

LCD – Gestion des utilisateurs et des privilèges

CREATE USER, GRANT, REVOKE

05-12-21 v1b

Module TC2-1 : Outils de base du développement

B2.3- Gérer les données

- Exploitation de données à l'aide d'un langage de requêtes

Savoirs technologiques

- Langage de définition des données, des contraintes et de contrôle des données.
- Langage et outils de manipulation et d'interrogation d'une base de données

Objectifs

- Savoir créer une base de données et un utilisateur ayant des privilèges sur celle-ci
- Réaliser des requêtes de contrôle de données (LCD) :
ajouter, supprimer des privilèges (GRANT, REVOKE)

Type de séance : Travaux Pratiques

Période : semestre 1

Durée : 2h00

A rendre : un compte-rendu rédigé.

Table des matières

1 - Visualisation des bases de données et utilisateurs existants.....	2
2 - Création d'une base de données, d'un utilisateur et des privilèges sur celle-ci.....	3
2.1 - Création de la base de données utilisateur 1.....	3
2.2 - Création du compte utilisateur 1.....	3
2.3 - Attribution des privilèges pour l'utilisateur 1.....	3
2.4 - Création de la base de donnée pour l'utilisateur 2, de son compte et de ses privilèges en utilisant phpmyadmin.....	4
2.5 - Création d'un utilisateur 3, administrateur des comptes et bdd utilisateurs.....	5
3 - Test des différents privilèges donnés aux utilisateurs.....	5
4 - ANNEXE.....	6
4.1 - SHOW DATABASES Statement.....	6
4.2 - SHOW TABLES Statement.....	6
4.3 - SHOW COLUMNS Statement.....	6
4.4 - SHOW GRANTS Statement.....	6
4.5 - CREATE DATABASE Statement.....	6
4.6 - CREATE USER Statement.....	7
4.7 - GRANT Statement.....	8

1 - Visualisation des bases de données et utilisateurs existants

Nous souhaitons visualiser les différentes bases de données et utilisateurs existant avec les privilèges qui leur ont été attribués. Pour cela, nous utiliserons un terminal et accèderons à ces informations à l'aide du client MariaDB en ligne de commandes (CLI).

- Ouvrez un terminal et connectez-vous au serveur MariaDB avec le compte root

```
[tux@ordihd-fd30 ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Figure 1: Connexion au serveur MariaDB à l'aide d'un client MariaDB lancé dans un terminal

- Donnez la commande SQL qui permet de visualiser les bases de données existantes (voir ANNEXE...)
- Visualisez les tables présentes dans la bdd **mysql** (voir ANNEXE...)

Les utilisateurs disponibles sont enregistrés dans la table **user** de la base de données **mysql**.

- Visualisez les noms de colonnes de la table **user** de la bdd **mysql**

Les login, hôte, mot de passe et mot de passe expiré sont donnés respectivement par les noms de colonnes suivants : **User**, **Host**, **Password**, **password_expired**

- Visualisez les informations suivantes sur les utilisateurs disponibles : login, hôte, mot de passe
- Visualisez les privilèges des utilisateurs actuels (voir ANNEXE)

2 - Création d'une base de données, d'un utilisateur et des privilèges sur celle-ci

Aujourd'hui, le manuel de référence de MySQL recommande d'utiliser le jeu de caractères **utf8mb4** et le classement **utf8mb4_0900_ai_ci** (ou **utf8mb4_unicode_ci** si vous n'utilisez pas la dernière version).

2.1 - Création de la base de données utilisateur 1

- Créez une base de données pour l'utilisateur Bob MORANE ayant pour nom son login **bmorane** et suivant les recommandations de MySQL concernant le jeu de caractères et de classement. Voir ANNEXE
- Vérifiez que la base de données a bien été créée avec les bons jeu de caractères et classement

2.2 - Création du compte utilisateur 1

- Créez un compte utilisateur pour Bob Morane ayant pour nom son login **bmorane**, pour mot de passe **P@ssw0rd** (création par défaut) et pouvant se connecter depuis n'importe quel hôte.

MariaDB et MySQL utilisent leur propre algorithme de hachage pour stocker les mots de passe (fonction `password()`) et donc empêcher qu'ils soient stockés en clair dans la base de données.

Pour calculer l'empreinte obtenue à partir d'un mot de passe, il suffit d'écrire la requête :

```
SELECT algorithme('mot de passe') avec algorithme = password, md5, sha1, ...
```

- Visualisez les informations enregistrées pour ce nouveau compte et précisez l'algorithme de hachage utilisé pour stocker le mot de passe.
- Visualisez les privilèges actuels de Bob MORANE. Quels sont-ils ?

2.3 - Attribution des privilèges pour l'utilisateur 1

- Donnez à Bob MORANE les privilèges suivant, uniquement sur sa base de données : interroger la base de données, insérer et modifier des enregistrements, créer et modifier des tables et colonnes,
- Vérifiez que les nouveaux privilèges ont bien été pris en compte

2.4 - Création de la base de donnée pour l'utilisateur 2, de son compte et de ses privilèges en utilisant phpmyadmin

- Créez une 2ème base de données pour l'utilisateur Lucie ROCCO nommée comme login, **lrocco**, en suivant les recommandations de MySQL concernant le jeu de caractères et de classement.



- Créez le compte utilisateur **lrocco** avec comme mot de passe **P@ssw0rd** (création par défaut) et ne pouvant se connecter qu'en local. Lucie ROCCO n'a aucun privilège global.

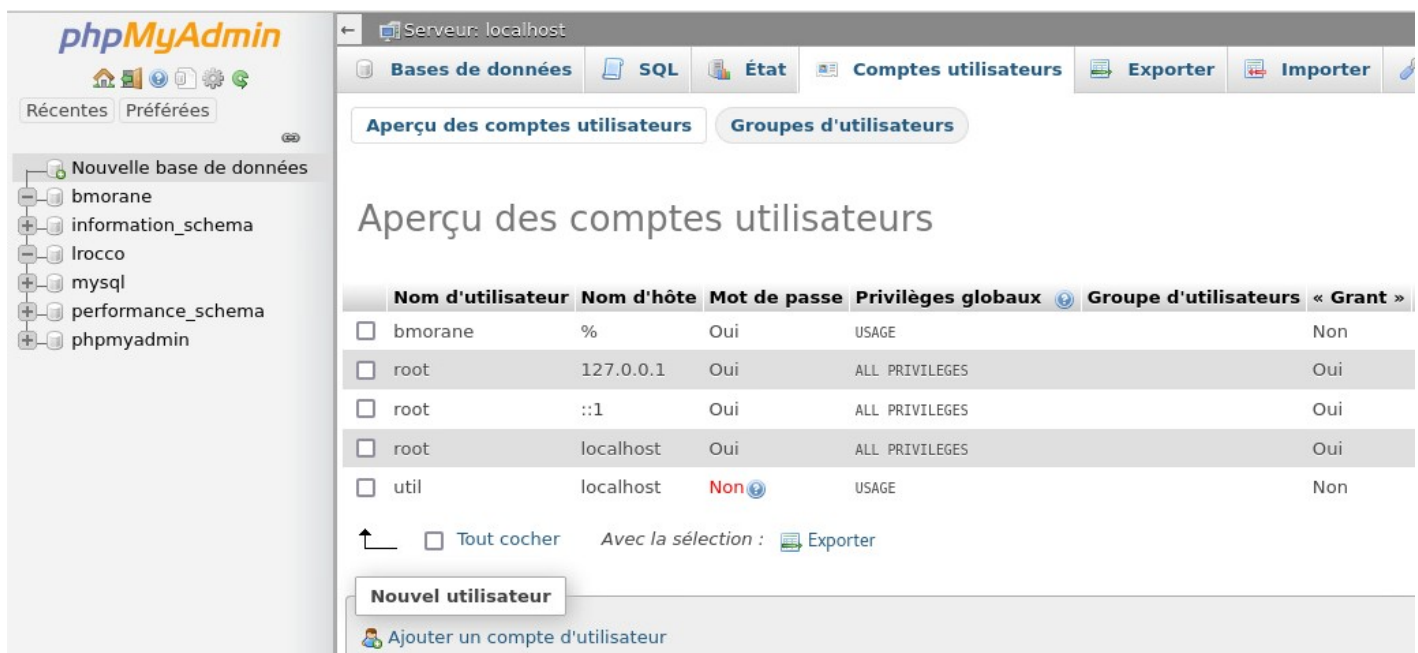


Figure 2: Pour créer un nouvel utilisateur, cliquer en bas sur Ajouter un compte utilisateur

✓ L'utilisateur a ajouté un utilisateur.

```
CREATE USER 'lrocco'@'localhost' IDENTIFIED VIA mysql_native_password USING '****';GRANT USAGE ON *.* TO 'lrocco'@'localhost'
REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

- Donnez à Lucie ROCCO les privilèges suivant, uniquement sur sa base de données : interroger la base de données, créer des tables et colonnes, insérer des enregistrements.
 - Cliquer sur **Éditer les privilèges** de l'utilisateur concerné
 - Cliquer sur **Base de données** (par défaut vous êtes dans les privilèges globaux)
 - Sélectionner la bdd de Lucie puis **Exécuter**
 - Cocher les privilèges à donner...

2.5 - Création d'un utilisateur 3, administrateur des comptes et bdd utilisateurs

- Créez le compte utilisateur **adminloc** avec comme mot de passe **P@ssw0rd** (création par défaut) et pouvant se connecter sur n'importe quel hôte. Ce compte aura les droits suivants sur toutes les bases de données utilisateurs (celles existantes et les futures) : créer, modifier, supprimer et attribuer des droits à un utilisateur, créer et supprimer une base de données, voir toutes les bases de données, interroger n'importe quelle base de données.

3 - Test des différents privilèges donnés aux utilisateurs

- Testez les différentes opérations indiquées dans le tableau et complétez-le en fonction des résultats obtenus.

Indiquez la procédure réalisée pour tester l'opération et faites une capture d'écran du résultat obtenu.

Remarque : il faut se connecter au serveur MariaDB avec le compte de l'utilisateur que l'on souhaite tester. Les tests peuvent être réalisés en ligne de commandes ou avec phpmyadmin.

Mettre une croix lorsque l'opération est autorisée

	Bob MORANE	Lucie ROCCO	Adminloc
Créer une base de donnée			X
Supprimer une base de donnée			
Créer un utilisateur			
Supprimer un utilisateur			
Se connecter à la bdd d'un autre utilisateur			
Créer une table dans sa bdd			
Supprimer une table dans sa bdd			
Créer une colonne			
Modifier une colonne			
Supprimer une colonne			
Insérer un enregistrement			
Modifier un enregistrement			
Supprimer un enregistrement			
Interroger sa bdd			
Donner des privilèges à un utilisateur			
Supprimer des privilèges à un utilisateur			

4 - ANNEXE

Sources : <https://mariadb.com/kb/en/sql-statements/> <https://dev.mysql.com/doc/refman/8.0/en/>

4.1 - SHOW DATABASES Statement

```
SHOW {DATABASES | SCHEMAS}
    [LIKE 'pattern' | WHERE expr]
```

4.2 - SHOW TABLES Statement

```
SHOW [EXTENDED] [FULL] TABLES
    [{FROM | IN} db_name]
    [LIKE 'pattern' | WHERE expr]
```

4.3 - SHOW COLUMNS Statement

```
SHOW [EXTENDED] [FULL] {COLUMNS | FIELDS}
    {FROM | IN} tbl_name
    [{FROM | IN} db_name]
    [LIKE 'pattern' | WHERE expr]
```

4.4 - SHOW GRANTS Statement

```
SHOW GRANTS
    [FOR user_or_role
      [USING role [, role] ...]]
```

```
user_or_role: {
    user (see Section 6.2.4, "Specifying Account Names")
  | role (see Section 6.2.5, "Specifying Role Names".
}
```

4.5 - CREATE DATABASE Statement

```
CREATE [OR REPLACE] {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification] ...
```

```
create_specification:
    [DEFAULT] CHARACTER SET [=] charset_name
  | [DEFAULT] COLLATE [=] collation_name
  | COMMENT [=] 'comment'
```

To see the default character set and collation for a given database, use these statements:

```
USE db_name;
SELECT @@character_set_database, @@collation_database;
```

Alternatively, to display the values without changing the default database:

```
SELECT DEFAULT_CHARACTER_SET_NAME, DEFAULT_COLLATION_NAME
FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME = 'db_name';
```

4.6 - CREATE USER Statement

<https://mariadb.com/kb/en/create-user/>

```
CREATE [OR REPLACE] USER [IF NOT EXISTS]
  user_specification [,user_specification ...]
  [REQUIRE {NONE | tls_option [[AND] tls_option ...] }]
  [WITH resource_option [resource_option ...] ]
  [lock_option] [password_option]

user_specification:
  username [authentication_option]

authentication_option:
  IDENTIFIED BY 'password'
  | IDENTIFIED BY PASSWORD 'password_hash'
  | IDENTIFIED {VIA|WITH} authentication_rule [OR authentication_rule ...]

authentication_rule:
  authentication_plugin
  | authentication_plugin {USING|AS} 'authentication_string'
  | authentication_plugin {USING|AS} PASSWORD('password')

tls_option:
  SSL
  | X509
  | CIPHER 'cipher'
  | ISSUER 'issuer'
  | SUBJECT 'subject'

resource_option:
  MAX_QUERIES_PER_HOUR count
  | MAX_UPDATES_PER_HOUR count
  | MAX_CONNECTIONS_PER_HOUR count
  | MAX_USER_CONNECTIONS count
  | MAX_STATEMENT_TIME time

password_option:
  PASSWORD EXPIRE
  | PASSWORD EXPIRE DEFAULT
  | PASSWORD EXPIRE NEVER
  | PASSWORD EXPIRE INTERVAL N DAY

lock_option:
  ACCOUNT LOCK
  | ACCOUNT UNLOCK
}
```

4.7 - GRANT Statement

GRANT

```
priv_type [(column_list)]
[, priv_type [(column_list)]] ...
ON [object_type] priv_level
TO user_specification [ user_options ...]
```

user_specification:

```
username [authentication_option]
```

authentication_option:

```
IDENTIFIED BY 'password'
| IDENTIFIED BY PASSWORD 'password_hash'
| IDENTIFIED {VIA|WITH} authentication_rule [OR authentication_rule ...]
```

authentication_rule:

```
authentication_plugin
| authentication_plugin {USING|AS} 'authentication_string'
| authentication_plugin {USING|AS} PASSWORD('password')
```

GRANT PROXY ON username

```
TO user_specification [, user_specification ...]
[WITH GRANT OPTION]
```

GRANT rolename TO grantee [, grantee ...]

```
[WITH ADMIN OPTION]
```

grantee:

```
rolename
username [authentication_option]
```

user_options:

```
[REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
[WITH with_option [with_option] ...]
```

object_type:

```
TABLE
| FUNCTION
| PROCEDURE
```

priv_level:

```
*
| *.*
| db_name.*
| db_name.tbl_name
| tbl_name
| db_name.routine_name
```

with_option:

```
GRANT OPTION
| resource_option
```

resource_option:

```
MAX_QUERIES_PER_HOUR count
| MAX_UPDATES_PER_HOUR count
| MAX_CONNECTIONS_PER_HOUR count
| MAX_USER_CONNECTIONS count
| MAX_STATEMENT_TIME time
```

tls_option:

```
SSL
| X509
| CIPHER 'cipher'
| ISSUER 'issuer'
| SUBJECT 'subject'
```