

# Gestion des notes

(Projet DIU)

Raoul HATTERER

20 février 2020

## Table des matières

<b>1</b>	<b>Installations et paramétrages</b>	<b>3</b>
1.1	Installer Homebrew . . . . .	3
1.2	Installer MariaDB . . . . .	3
1.3	Démarrer MariaDB Server . . . . .	3
1.4	Vérifier que MariaDB a démarré . . . . .	3
1.5	Client-serveur . . . . .	3
1.6	Distinguer les utilisateurs . . . . .	4
1.7	Création d'une nouvelle base de données et d'un gestionnaire pour assurer sa gestion . . . . .	5
<b>2</b>	<b>Afficher les utilisateurs</b>	<b>6</b>
2.1	Afficher tous les utilisateurs (en tant qu'administrateur) . . .	6
2.2	Afficher l'utilisateur connecté . . . . .	6
<b>3</b>	<b>Gestion de la base de données</b>	<b>7</b>
3.1	Se connecter au client <code>mysql</code> en tant que gestionnaire . . . . .	7
3.2	Lister les bases de données (auxquelles le gestionnaire a accès)	7
3.3	Se connecter à une base de données . . . . .	7
3.4	Afficher les tables de la base courante . . . . .	7
3.5	Afficher la structure de la table . . . . .	7
3.6	Créer une nouvelle table . . . . .	7
3.7	Ajouter une colonne à une table . . . . .	7
3.8	Effacer une colonne . . . . .	8
3.9	Effacer une table . . . . .	8
3.10	Afficher le contenu de la table . . . . .	8

<b>4</b>	<b>sql-mode dans emacs</b>	<b>8</b>
4.1	Se connecter au client <code>mysql</code> dans emacs . . . . .	8
4.2	Si emacs ne trouve pas le programme <code>mysql</code> . . . . .	8
4.3	Connexion facilitée . . . . .	9
4.4	SQLUP-MODE . . . . .	9
<b>5</b>	<b>How to run sql script</b>	<b>9</b>
<b>6</b>	<b>Python tkinter table widget for displaying tabular data</b>	<b>9</b>
6.1	installation . . . . .	9
6.2	source . . . . .	10
6.3	paramètres . . . . .	10
6.4	fonctions . . . . .	10

# 1 Installations et paramétrages

**Remarque :** Certaines parties concernent plus spécifiquement mon utilisation personnelle avec [emacs](#) sous [OSX](#).

## 1.1 Installer Homebrew

Pour installer MariaDB, il faut disposer de [Homebrew](#) (The Missing Package Manager for macOS).

## 1.2 Installer MariaDB

MariaDB Server peut ensuite être installé grâce à cette commande :

```
shell  
brew update  
brew install mariadb
```

## 1.3 Démarrer MariaDB Server

Après installation, on peut (mais on ne va pas le faire) démarrer le serveur avec la commande :

```
shell  
mysql.server start
```

Il est plus commode d'activer le démarrage automatique du serveur :

```
shell  
brew services start mariadb
```

## 1.4 Vérifier que MariaDB a démarré

```
shell  
brew services list
```

## 1.5 Client-serveur

Comme la plupart des SGBD ([S]ystème de [G]estion de [B]ase de [D]onnées), MariaDB est basée sur le **modèle client-serveur**. Cela implique que la base de données se trouve sur un **serveur** auquel nous n'avons pas accès directement. Il faut passer par un **client** qui fera la liaison entre nous et le serveur.

Donc, maintenant que MariaDB Server a démarré, on va s'y connecter en utilisant un logiciel client (nommé `mysql`)... mais sous quelle identité et avec quels droits ?

## 1.6 Distinguer les utilisateurs

### 1.6.1 Types d'utilisateurs

Il faut distinguer trois types d'utilisateur d'un SGBD ayant chacun des rôles bien distincts :

1. L'**administrateur** du SGBD et donc des Bases de Données. Il s'appelle **root**. Son rôle est d'installer et gérer le SGBD dans sa globalité. Chaque fois que l'administrateur crée une base de données, il délègue sa gestion à un autre utilisateur à qui il donne les droits nécessaires.  
**Remarque :** Il ne faut pas travailler sur une bases de données en tant que **root** ; ce n'est pas son rôle (*c'est mal*).
2. Le **gestionnaire** de Bases de Données (dans un SGBD existant) qui doit connaître **sql** (Structured Query Language) : création (CREATE TABLE), modification (ALTER TABLE), suppression (DROP TABLE) de tables...
3. L'**utilisateur** de bases de données (il va interagir avec la base de donnée à travers une application se voulant conviviale ; il n'a pas à connaître **sql**).

### 1.6.2 À la création

**À la création, la base de donnée est initialisée avec 2 comptes :**

- le premier correspond au nom d'utilisateur actuel (checké avec la commande **whoami**) ; il ne nécessite pas de mot de passe ;
- tandis que le second est le compte administrateur **root** ; il est inaccessible tant qu'on n'a pas défini de mot de passe.

On va donc faire ce qu'il faut pour activer l'accès au compte administrateur (**root**) puis s'y connecter pour créer une base de données (**bd\_gestion\_des\_notes**) et un compte gestionnaire.

### 1.6.3 Activer l'accès au compte root

Pour cela, d'abord se connecter au client **mysql** avec son nom d'utilisateur :

```
mysql -u $(whoami)
```

Puis, grâce au commandes du client **mysql**, définir le mot de passe de **root** :

```
sql
SHOW databases;
USE mysql;
SET password FOR 'root'@'localhost' =
password('YOUR_ROOT_PASSWORD_HERE');
FLUSH PRIVILEGES;
quit
```

où YOUR\_ROOT\_PASSWORD\_HERE est à remplacer par le mot de passe souhaité.

## 1.7 Création d'une nouvelle base de données et d'un gestionnaire pour assurer sa gestion

### 1.7.1 Login en tant que root

Maintenant, nous pouvons nous connecter en tant que root :

```
shell
mysql -u root -p
```

Il faudra fournir le mot de passe renseigné à l'étape précédente.

### 1.7.2 Commandes de base à la disposition de l'administrateur

— Afficher le statut du serveur :

```
sql
SHOW STATUS;
```

— Créer une nouvelle base de données :

```
sql
CREATE DATABASE nom_database;
```

— Lister les bases de données :

```
sql
SHOW DATABASES;
```

— Effacer une base de donnée :

```
sql
DROP DATABASE nom_database;
```

### 1.7.3 Création de la nouvelle base de données `bd_gestion_des_notes` :

```
sql  
CREATE DATABASE bd_gestion_des_notes;  
SHOW DATABASES;
```

### 1.7.4 Création d'un utilisateur `gestionnaire`

Utilisateur à qui `root` va donner les droits nécessaires pour créer ou modifier des tables dans la base de données `bd_gestion_des_notes` ou pour inscrire des utilisateurs

```
sql  
CREATE USER user_gestionnaire@localhost  
IDENTIFIED BY 'gestionnaire';  
GRANT ALL PRIVILEGES ON bd_gestion_des_notes.*  
TO user_gestionnaire@localhost;  
quit
```

### 1.7.5 Effacer l'utilisateur `gestionnaire`

```
sql  
DROP USER user_gestionnaire@localhost;
```

## 2 Afficher les utilisateurs

### 2.1 Afficher tous les utilisateurs (en tant qu'administrateur)

Connecté en `root` saisir la commande :

```
sql  
select host, user from mysql.user;
```

### 2.2 Afficher l'utilisateur connecté

Deux commandes possibles pour cela :

```
sql  
SELECT current_user;
```

ou

```
sql  
SELECT user();
```

## 3 Gestion de la base de données

### 3.1 Se connecter au client `mysql` en tant que gestionnaire

```
mysql -u user_gestionnaire -p
```

Saisir le mot de passe précédemment défini.

### 3.2 Lister les bases de données (auxquelles le gestionnaire a accès)

```
SHOW DATABASES;
```

### 3.3 Se connecter à une base de données

Pour utiliser la base de donnée `bd_gestion_des_notes` :

```
USE bd_gestion_des_notes;
```

### 3.4 Afficher les tables de la base courante

```
SHOW TABLES;
```

### 3.5 Afficher la structure de la table

```
DESCRIBE nom_table;
```

### 3.6 Créer une nouvelle table

```
CREATE TABLE nom_table (  
  nom_colonne1 int,  
  nom_colonne2 VARCHAR(100)  
);
```

### 3.7 Ajouter une colonne à une table

```
ALTER TABLE nom_table ADD nom_colonne type_colonne;
```

### 3.8 Effacer une colonne

\_\_\_\_\_ sql \_\_\_\_\_  
`ALTER TABLE nom_table DROP COLUMN nom_colonne;`

### 3.9 Effacer une table

\_\_\_\_\_ sql \_\_\_\_\_  
`DROP nom_table;`

### 3.10 Afficher le contenu de la table

\_\_\_\_\_ sql \_\_\_\_\_  
`SELECT * FROM nom_table;`

## 4 sql-mode dans emacs

### 4.1 Se connecter au client mysql dans emacs

Utiliser la commande : M-x sql-mysql

Renseigner :

- User : user\_gestionnaire
- Password : gestionnaire
- Database : (ne rien mettre)
- Server : localhost

### 4.2 Si emacs ne trouve pas le programme mysql

- Déterminer l'emplacement de mysql avec la commande :

\_\_\_\_\_ shell \_\_\_\_\_  
`which mysql`

qui retourne /usr/local/bin/mysql

- modifier .emacs.d/init.el

Emacs recherche les programmes dans les répertoires listés dans la variable `exec-path`. On va ajouter /usr/local/bin à cette variable.

\_\_\_\_\_ elisp \_\_\_\_\_  
`(add-to-list 'exec-path "/usr/local/bin")`



### 4.3 Connexion facilitée

À placer dans `.emacs.d/init.el`  
elisp

```
(setq sql-user "user_gestionnaire")
(setq sql-database "bd_gestion_des_notes")
(setq sql-server "localhost")
```

Connexion acilitée, toujours avec la commande `M-x sql-mysql` et dorénavant il ne reste plus qu'à saisir le mot de passe utilisateur (en l'occurrence : `gestionnaire`); le reste étant renseigné à l'avance.

### 4.4 SQLUP-MODE

Capitalise les mots clés sql.

Les lignes suivantes sont à placer dans le fichier `init.el` :

```
elisp
(use-package sqlup-mode
  :ensure t
  :config
  (add-hook 'sql-mode-hook 'sqlup-mode)
  (add-hook 'sql-interactive-mode-hook 'sqlup-mode)
  (add-hook 'redis-mode-hook 'sqlup-mode)
  )
```

## 5 How to run sql script

If youre at the MySQL command line `mysql>` you have to declare the SQL file as source.

```
mysql> source \path\to\the\script_file.sql;
```

For example :

```
mysql> source \home\user\Desktop\script_file.sql;
```

## 6 Python tkinter table widget for displaying tabular data

### 6.1 installation

```
pip install tksheet
```

## 6.2 source

<https://github.com/ragardner/tksheet>

## 6.3 paramètres

[https://github.com/ragardner/tksheet/blob/master/DOCUMENTATION.  
md](https://github.com/ragardner/tksheet/blob/master/DOCUMENTATION.md)

## 6.4 fonctions

[https://github.com/ragardner/tksheet/blob/master/tksheet/\\_tksheet.  
py](https://github.com/ragardner/tksheet/blob/master/tksheet/_tksheet.py)