

TP Administration : Base de données (MySQL/MariaDB)

© 2019 tv <tvaira@free.fr> - v.1.0

Base de données	1
MySQL/MariaDB	2
Présentation	2
Travail demandé	3
Installation	3
Configuration	3
Tests	4
PHP	4

TP Administration

L'objectif de cette activité est d'installer un serveur de base de données (MySQL/MariaDB).

Base de données

Une base de données (*database*) est un « conteneur » permettant de stocker et de retrouver l'intégralité de données brutes ou d'informations. Dans la très grande majorité des cas, ces informations sont très structurées, et la base est localisée dans un même lieu et sur un même support.

Le dispositif comporte un système de gestion de base de données (SGBD) : un logiciel moteur qui manipule la base de données et dirige l'accès à son contenu. De tels dispositifs comportent également des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations.

Une base de données relationnelle est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés des relations ou tables. Les lignes de ces relations sont appelées des *nuplets* (tuples) ou enregistrements. Les noms des colonnes (ou champs) sont appelées des attributs.

Les logiciels qui permettent de créer, utiliser et maintenir des bases de données relationnelles sont des système de gestion de base de données relationnels (SGBDR).

Pratiquement tous les systèmes relationnels utilisent le langage SQL (*Structured Query Language*) pour interroger les bases de données. Ce langage permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

MySQL/MariaDB

Présentation

MySQL est un système de gestion de bases de données relationnelles (SGBDR).

Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

MySQL est un serveur de bases de données relationnelles SQL. Il est multi-thread et multi-utilisateur. MySQL fonctionne sur de nombreux systèmes d'exploitation différents, incluant Linux, Mac OS X et Windows. Les bases de données sont accessibles en utilisant les langages de programmation C, C++, VB, VB .NET, C#, Delphi/Kylix, Eiffel, Java, Perl, PHP, Python, Windev, Ruby et Tcl. Une API spécifique est disponible pour chacun d'entre eux.

MySQL fait partie du quatuor LAMP : Linux, Apache, MySQL, PHP. Il appartient également à ses variantes WAMP (Windows) et MAMP (Mac OS).

MariaDB est un système de gestion de base de données édité sous licence GPL. Il s'agit d'un *fork* communautaire de MySQL.



En 2009, à la suite du rachat de MySQL par Sun Microsystems et des annonces du rachat de Sun Microsystems par Oracle Corporation, Michael Widenius, fondateur de MySQL, quitte cette société pour lancer le projet MariaDB, dans une démarche visant à remplacer MySQL tout en assurant l'interopérabilité. Le nom vient de la 2e fille de Michael Widenius, Maria (la première s'appelant My). MariaDB a été choisi par défaut sur les distributions « Debian ».

Documentations :

- <https://doc.ubuntu-fr.org/mysql>
- <https://doc.ubuntu-fr.org/mariadb>



L'ensemble du comportement du serveur MariaDB reste très similaire à celui de MySQL. La commande `mysql` reste valable, ainsi que les commandes `systemctl ... mysql`. Pour les différences, il faut se référer aux documentations respectives.

Travail demandé

Installation

Question 1. Installez MySQL et vérifiez son état. Indiquez la version installée.

```
$ sudo apt install mysql-server
```

...

Question 2. Indiquez la version installée.

Question 3. Lancer la console MySQL et lister les base de données installées.

Question 4. Lister les utilisateurs.

Question 5. Créer la base de données `meteo` à partir du fichier SQL ci-dessous :

```
$ vim meteo.sql
```

```
CREATE DATABASE IF NOT EXISTS 'meteo';
```

```
USE 'meteo';
```

```
CREATE TABLE 'mesures' (  
  'id' int(10) NOT NULL AUTO_INCREMENT,  
  'date' date NOT NULL,  
  'heure' time NOT NULL,  
  'temperature' float NOT NULL,  
  PRIMARY KEY ('id')  
);
```

```
INSERT INTO 'mesures' ('id', 'date', 'heure', 'temperature') VALUES  
(1, '2019-09-08', '08:00:00', 35.23),  
(2, '2019-09-08', '08:01:00', 35.1),  
(3, '2019-09-08', '08:02:00', 34.45),  
(4, '2019-09-08', '08:03:00', 35.02),  
(5, '2019-09-08', '08:04:00', 35.53),  
(6, '2019-09-09', '08:00:00', 35.23),  
(7, '2019-09-09', '08:01:00', 35.1),  
(8, '2019-09-09', '08:02:00', 34.45),  
(9, '2019-09-09', '08:03:00', 35.02),  
(10, '2019-09-09', '08:04:00', 35.53),  
(11, '2019-09-09', '08:05:00', 35.12);
```

Question 6. Dans la console MySQL, afficher les tables de la base de données `meteo` ainsi que les mesures.

Configuration

Question 7. Créer l'utilisateur *meteorologue* pour lequel vous autorisez l'accès distant pour tous les hôtes spécifiquement à la base de données `meteo`.

Question 8. Autoriser l'accès distant pour le serveur de base de données. Quel est le port d'écoute par défaut du serveur MySQL ?

Tests

Question 9. Tester un accès à partir d'un poste client vers la base de données **meteo** avec le compte *meteorologue*.

Question 10. À partir du poste client, afficher le nombre de mesures.

PHP

Question 11. Pour que PHP dialogue avec le serveur MySQL/MariaDB, il faut installer le module php-mysql.

```
$ sudo apt-get install php-mysql
```

Question 12. Créer le script `meteo.php`.

```
$ vim meteo.php
```

```
<?php
$mysqli = new mysqli("localhost", "meteorologue", "password", "meteo");

if ($mysqli->connect_error)
{
    die('Echec de connexion au serveur de base de données : ' . $mysqli->connect_error . '('
        . $mysqli->connect_errno . ') ');
}

echo "Affichage des mesures\n";

$requete = "SELECT date, heure, temperature FROM mesures";
//var_dump($requete);

if ($result = $mysqli->query($requete))
{
    printf("Classe mysqli : la requête a retourné %d enregistrement(s).\n", $result->num_rows
    );
    while($row = $result->fetch_array())
    {
        //var_dump($row);
        echo $row["date"]." à ".$row["heure"]." -> ".$row["temperature"]." °C\n";
    }

    /* Libération des résultats */
    $result->free();
}

$mysqli->close();
?>
```

Question 13. Tester le script en mode CLI.

```
$ php meteo.php
Affichage des mesures
Classe mysqli : la requête a retourné 11 enregistrement(s).
2019-09-08 à 08:00:00 -> 35.23 °C
2019-09-08 à 08:01:00 -> 35.1 °C
2019-09-08 à 08:02:00 -> 34.45 °C
2019-09-08 à 08:03:00 -> 35.02 °C
2019-09-08 à 08:04:00 -> 35.53 °C
2019-09-09 à 08:00:00 -> 35.23 °C
2019-09-09 à 08:01:00 -> 35.1 °C
2019-09-09 à 08:02:00 -> 34.45 °C
2019-09-09 à 08:03:00 -> 35.02 °C
2019-09-09 à 08:04:00 -> 35.53 °C
2019-09-09 à 08:05:00 -> 35.12 °C
```

Question 14. Si un serveur HTTP (Apache) est installé, tester le script à partir d'un navigateur. Que faut-il modifier dans le script pour obtenir un affichage adapté au navigateur ?

```
$ systemctl status apache2

// Si le module PHP a été installé, il faut recharger le serveur Apache
$ sudo systemctl reload apache2
```