

# Créer une base de données

## En ligne de commande

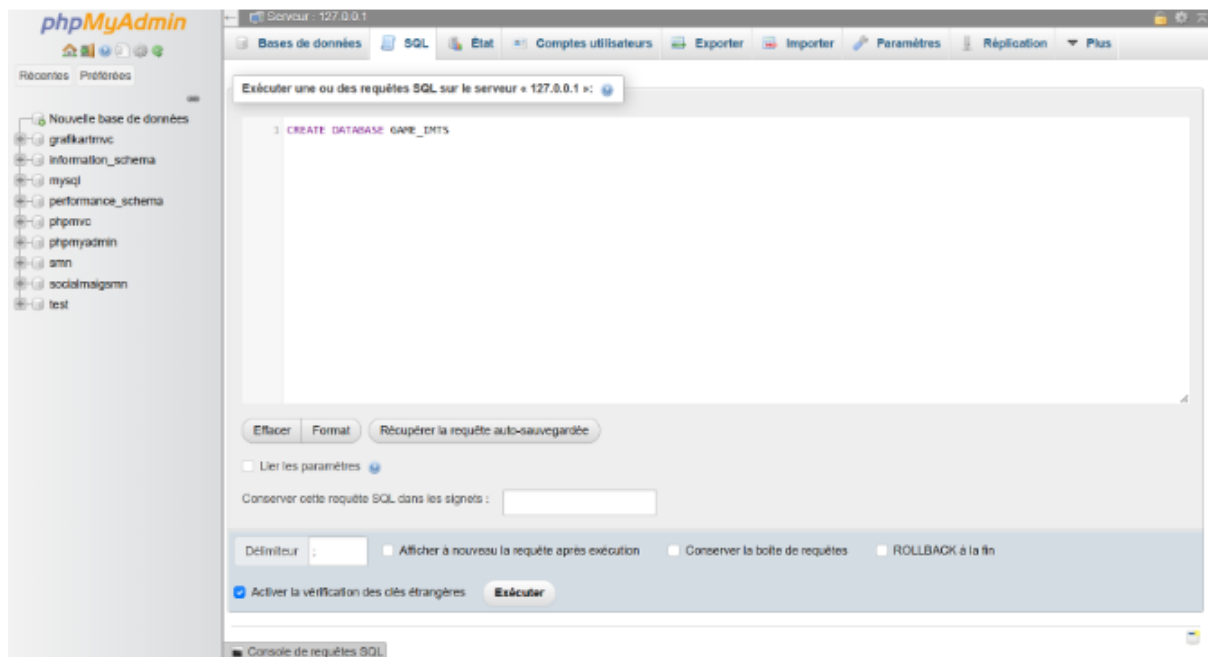
Pour créer une base de données en ligne de commande, vous pouvez utiliser la commande `CREATE DATABASE` dans un environnement de base de données approprié. Par exemple, si vous utilisez MySQL, vous pouvez utiliser la commande suivante pour créer une base de données nommée `GAME_IMTS` :

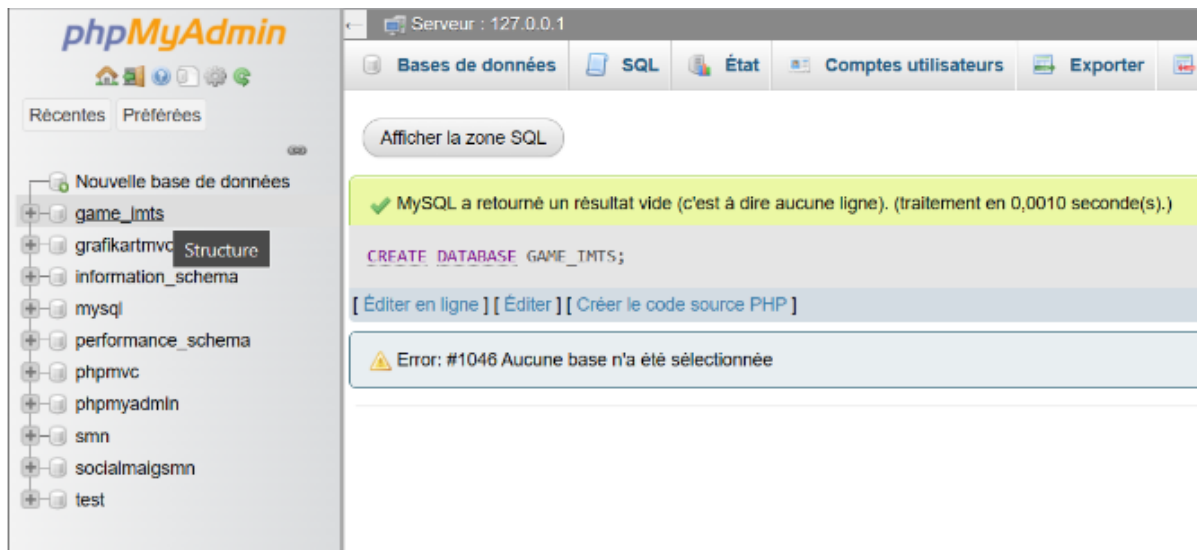
```
mysql -u root -e "CREATE DATABASE GAME_IMTS";
```

Remplacez `root` par le nom d'utilisateur approprié pour votre environnement de base de données. Si votre utilisateur nécessite un mot de passe, vous pouvez ajouter l'option `-p` pour être invité à saisir votre mot de passe.

## En utilisant PhpMyAdmin

```
CREATE DATABASE GAME_IMTS
```





## Créer une table

→ Créer une table users

CREATE TABLE nom\_de\_la\_table

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nom VARCHAR(255),  
  email VARCHAR(255) NOT NULL UNIQUE,  
  mot_de_passe VARCHAR(255)  
);
```

## Les types de champs d'une base de données

Il en existe plusieurs dans une base de données, chacun ayant des caractéristiques et des utilisations spécifiques. Voici une liste de quelques types de champs couramment utilisés dans les bases de données, ainsi qu'un exemple pour chacun :

- **Texte** : Ce type de champ est utilisé pour stocker des données textuelles, telles que des noms, des adresses ou des descriptions. Par exemple, un champ `nom` dans une table `clients` pourrait être de type texte et contenir les noms des clients.
- **Nombre** : Ce type de champ est utilisé pour stocker des données numériques, telles que des quantités, des prix ou des mesures. Par exemple, un champ `prix` dans une table `produits` pourrait être de type nombre et contenir les prix des produits.

- **Date/Heure** : Ce type de champ est utilisé pour stocker des dates et/ou des heures. Par exemple, un champ `date_de_naissance` dans une table `clients` pourrait être de type date/heure et contenir les dates de naissance des clients.
- **Booléen** : Ce type de champ est utilisé pour stocker des valeurs booléennes (vrai/faux). Par exemple, un champ `est_actif` dans une table `clients` pourrait être de type booléen et indiquer si un client est actif ou non.
- **Mémo** : Ce type de champ est utilisé pour stocker de grandes quantités de texte, telles que des notes ou des commentaires. Par exemple, un champ `notes` dans une table `commandes` pourrait être de type mémo et contenir des notes sur les commandes.

## → Type des champs de MySQL

MySQL supporte plusieurs types de données dans différentes catégories, notamment les types numériques, les types date et heure, les types chaîne (caractère et octet), les types spatiaux et le type de données JSON1. Voici une liste de quelques types de données couramment utilisés dans MySQL, ainsi qu'un exemple de ligne de commande pour chacun :

- **Types numériques** : Les types numériques incluent `TINYINT`, `SMALLINT`, `MEDIUMINT`, `INT`, `BIGINT`, `DECIMAL`, `FLOAT` et `DOUBLE`. Par exemple, pour créer une colonne `age` de type `INT` dans une table `users`, vous pouvez utiliser la commande suivante :

```
ALTER TABLE users ADD COLUMN age INT;
```

- **Types date et heure** : Les types date et heure incluent `DATE`, `TIME`, `DATETIME`, `TIMESTAMP` et `YEAR`. Par exemple, pour créer une colonne `date_de_naissance` de type `DATE` dans une table `users`, vous pouvez utiliser la commande suivante :

```
ALTER TABLE users ADD COLUMN date_de_naissance DATE;
```

- **Types chaîne** : Les types chaîne incluent `CHAR`, `VARCHAR`, `BINARY`, `VARBINARY`, `BLOB`, `TEXT`, `ENUM` et `SET`. Par exemple, pour créer une colonne `nom` de type `VARCHAR(255)` dans une table `users`, vous pouvez utiliser la commande suivante :

## PRIMARY KEY

En MySQL, une clé primaire (PRIMARY KEY) est une colonne ou un ensemble de colonnes qui identifie de manière unique chaque ligne dans une table. La clé primaire suit ces règles :

- Une clé primaire doit contenir des valeurs uniques. Si elle est composée de plusieurs colonnes, la combinaison des valeurs dans ces colonnes doit être unique.
- Une colonne de clé primaire ne peut pas avoir de valeurs NULL. Toute tentative d'insérer ou de mettre à jour NULL entraînera une erreur.

Une table peut avoir une et une seule clé primaire. Lorsque vous définissez une clé primaire pour une table, MySQL crée automatiquement un index appelé PRIMARY.

## NULL et NOT NULL

En SQL, `NULL` représente une valeur inconnue ou indéfinie dans une colonne d'une table. Par exemple, si une colonne `age` dans une table `users` contient une valeur `NULL`, cela signifie que l'âge de cet utilisateur est inconnu ou non défini.

Lors de la création d'une table ou de l'ajout d'une colonne à une table, vous pouvez spécifier si la colonne peut contenir des valeurs `NULL` ou non en utilisant les options `NULL` ou `NOT NULL`. Si vous spécifiez `NOT NULL`, cela signifie que la colonne ne peut pas contenir de valeurs `NULL`. Si vous spécifiez `NULL`, cela signifie que la colonne peut contenir des valeurs `NULL`.

Il est important de noter que `NULL` est différent de zéro ou d'une chaîne vide. Une colonne contenant une valeur zéro ou une chaîne vide contient toujours une valeur, tandis qu'une colonne contenant une valeur `NULL` ne contient aucune valeur.

## AUTO\_INCREMENT

`AUTO_INCREMENT` est un attribut utilisé dans MySQL pour générer automatiquement une valeur unique pour une colonne lors de l'insertion d'une nouvelle ligne dans une table. Cet attribut est généralement utilisé avec une colonne de clé primaire pour garantir que chaque ligne a une valeur unique.

Par défaut, la valeur de départ pour `AUTO_INCREMENT` est 1, et elle s'incrémente de 1 pour chaque nouvel enregistrement. Vous pouvez modifier la valeur de départ en utilisant la commande `ALTER TABLE` avec l'option `AUTO_INCREMENT`.

Voici un exemple de création d'une table avec une colonne `id` qui utilise l'attribut `AUTO_INCREMENT` :

```
USE GAME_IMTS; CREATE TABLE users (  
  id INT NOT NULL AUTO_INCREMENT,  
  nom VARCHAR(255),  
  email VARCHAR(255) NOT NULL UNIQUE,  
  mot_de_passe VARCHAR(255),  
  PRIMARY KEY (id)  
);
```

Dans cet exemple, la colonne `id` est définie comme clé primaire et utilise l'attribut `AUTO_INCREMENT`. Lorsque vous insérez une nouvelle ligne dans cette table, vous n'avez pas besoin de spécifier une valeur pour la colonne `id`, car elle sera générée automatiquement.

Voici l'explication de chaque ligne de ce code MySQL :

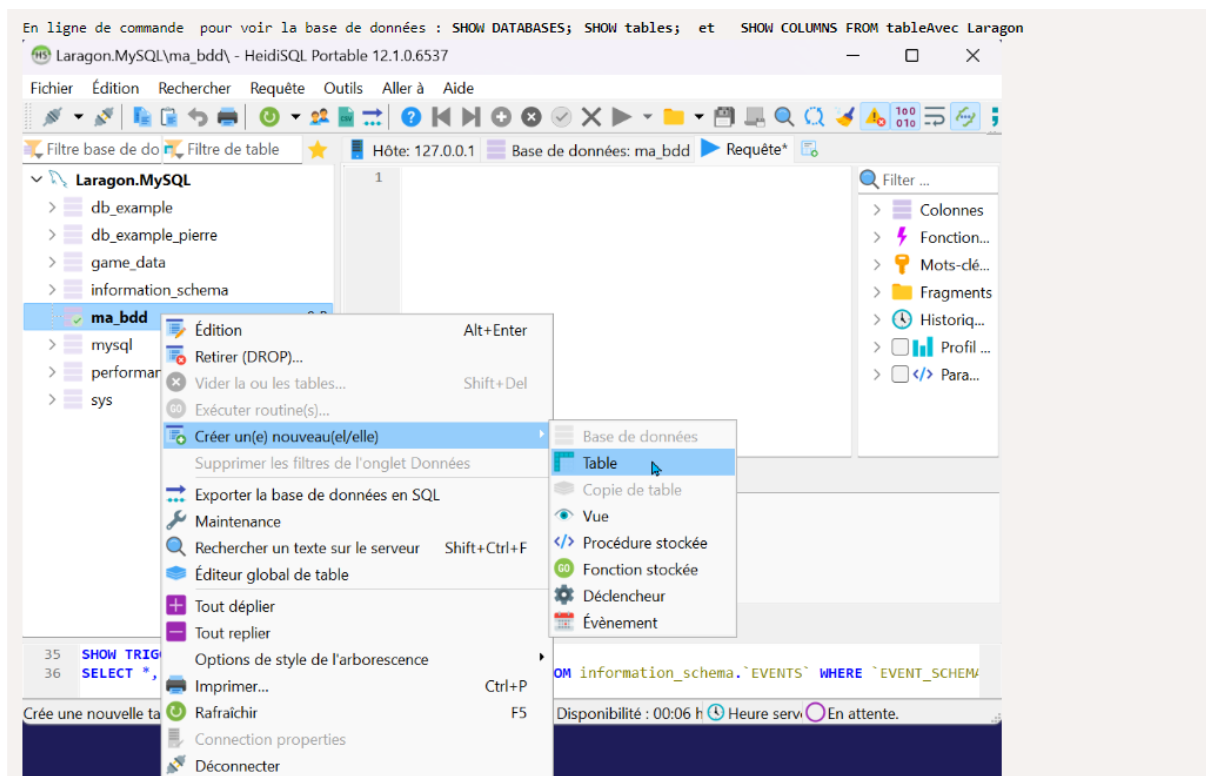
1. `USE GAME_IMTS;` : Cette ligne indique que la base de données à utiliser pour les commandes suivantes est `GAME_IMTS`.
2. `CREATE TABLE users (` : Cette ligne crée une nouvelle table appelée `users` dans la base de données `GAME_IMTS`.
3. `id INT NOT NULL AUTO_INCREMENT,` : Cette ligne crée une colonne appelée `id` dans la table `users`. La colonne est de type `INT` (entier) et ne peut pas être nulle (`NOT NULL`). La colonne est également configurée pour s'incrémenter automatiquement (`AUTO_INCREMENT`) à chaque fois qu'une nouvelle ligne est insérée dans la table.
4. `nom VARCHAR(255),` : Cette ligne crée une colonne appelée `nom` dans la table `users`. La colonne est de type `VARCHAR(255)` (chaîne de caractères variable avec une longueur maximale de 255 caractères).
5. `email VARCHAR(255) NOT NULL UNIQUE,` : Cette ligne crée une colonne appelée `email` dans la table `users`. La colonne est de type `VARCHAR(255)` (chaîne de caractères variable avec une longueur maximale de 255 caractères) et ne peut pas être nulle (`NOT NULL`). La colonne est également configurée pour être unique (`UNIQUE`), ce qui signifie que chaque valeur dans cette colonne doit être unique dans la table.
6. `mot_de_passe VARCHAR(255),` : Cette ligne crée une colonne appelée `mot_de_passe` dans la table `users`. La colonne est de type `VARCHAR(255)` (chaîne de caractères variable avec une longueur maximale de 255 caractères).
7. `PRIMARY KEY (id)` : Cette ligne définit la colonne `id` comme clé primaire (`PRIMARY KEY`) de la table `users`. Cela signifie que chaque valeur dans cette colonne doit être unique et ne peut pas être nulle.
8. `);` : Cette ligne termine la commande `CREATE TABLE`.

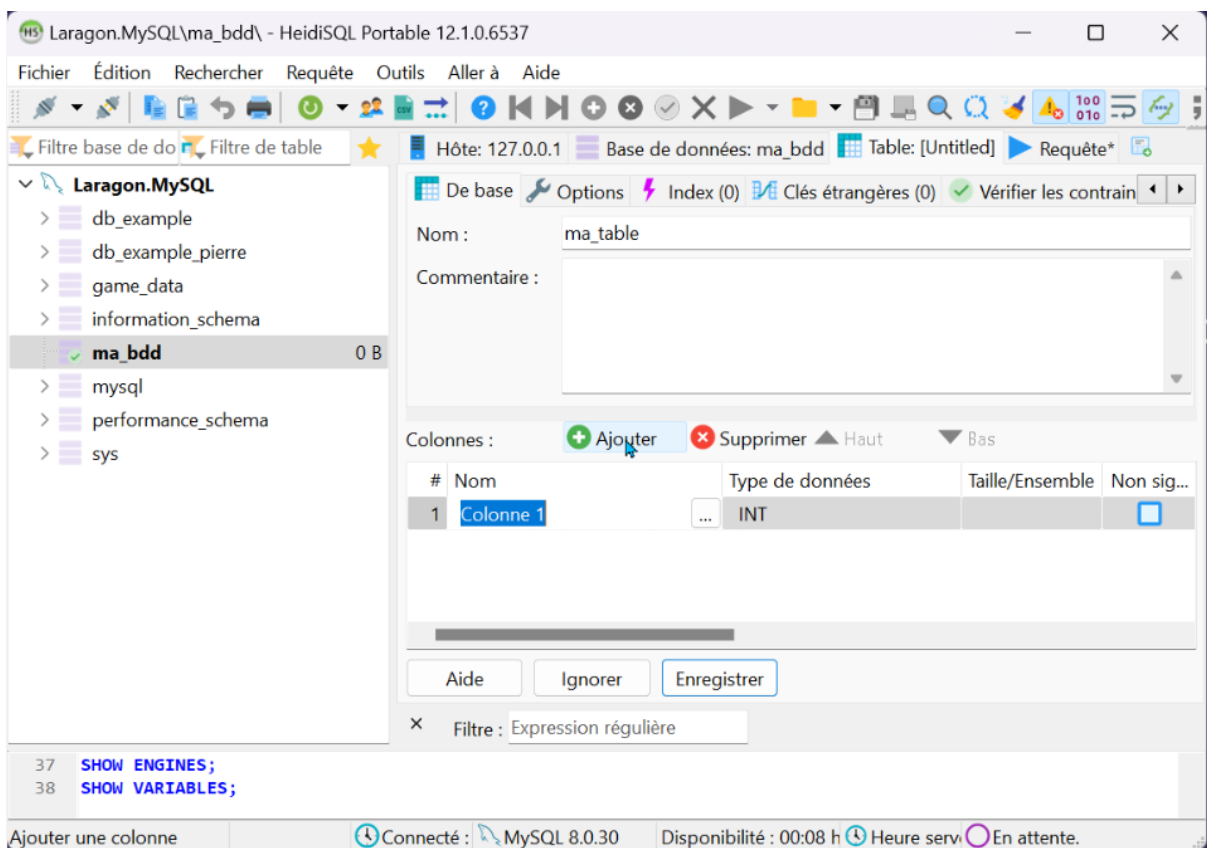
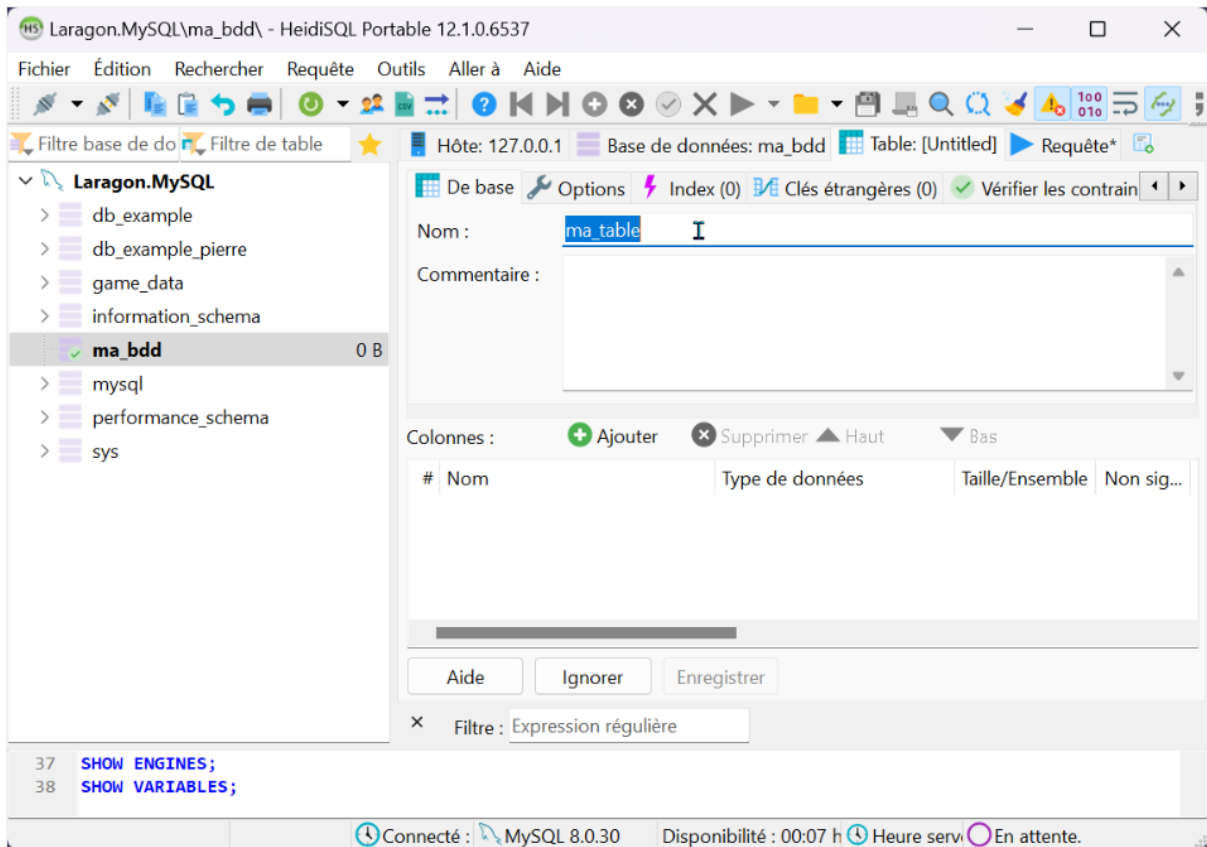
## En ligne de commande

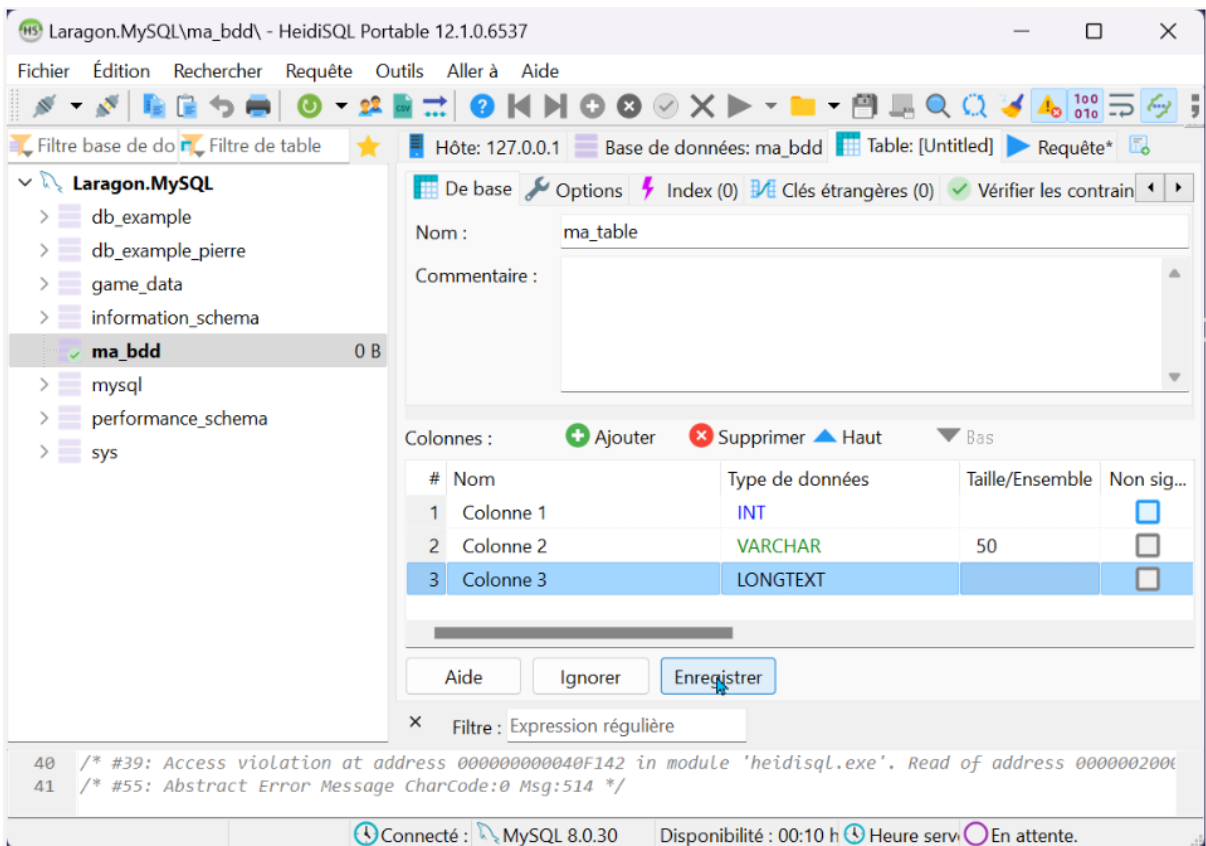
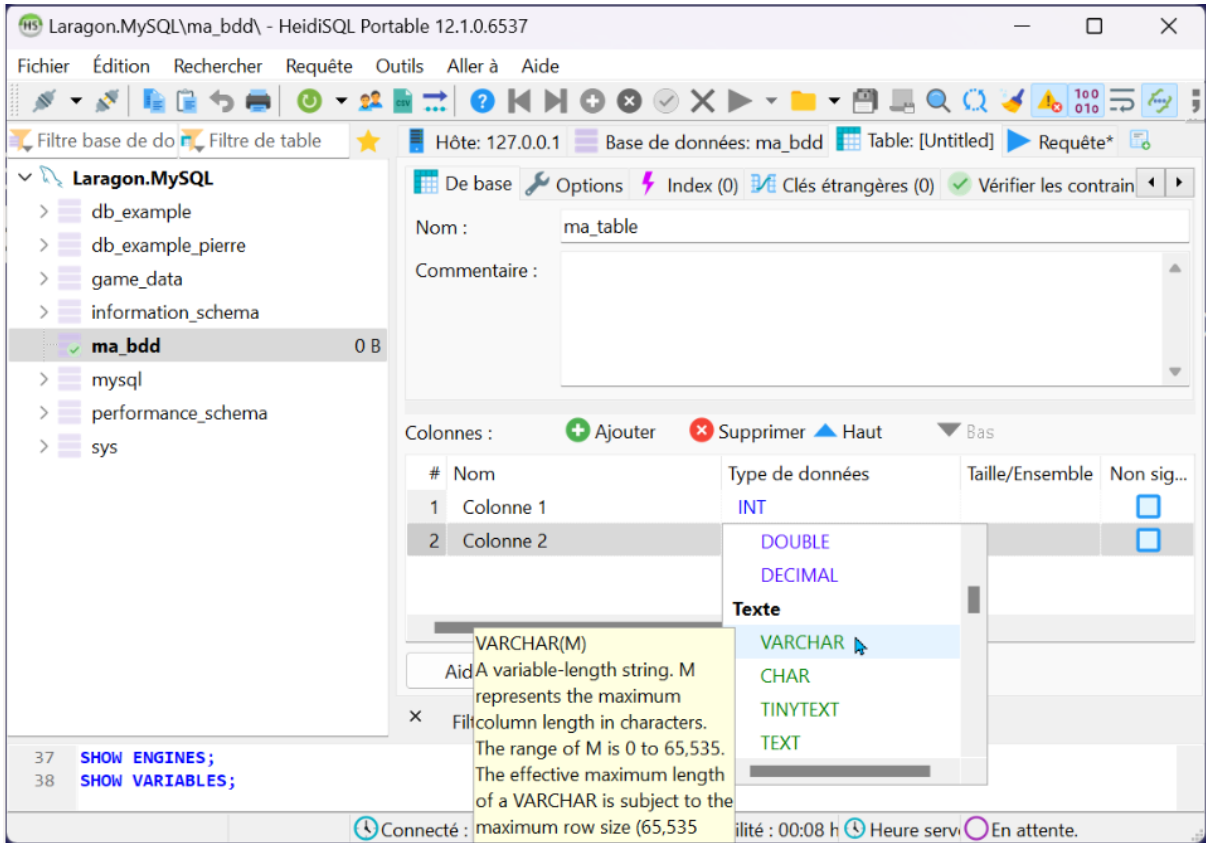
Pour créer une table dans la base de données `GAME_IMTS` en utilisant MySQL en ligne de commande, vous pouvez utiliser les commandes suivantes :

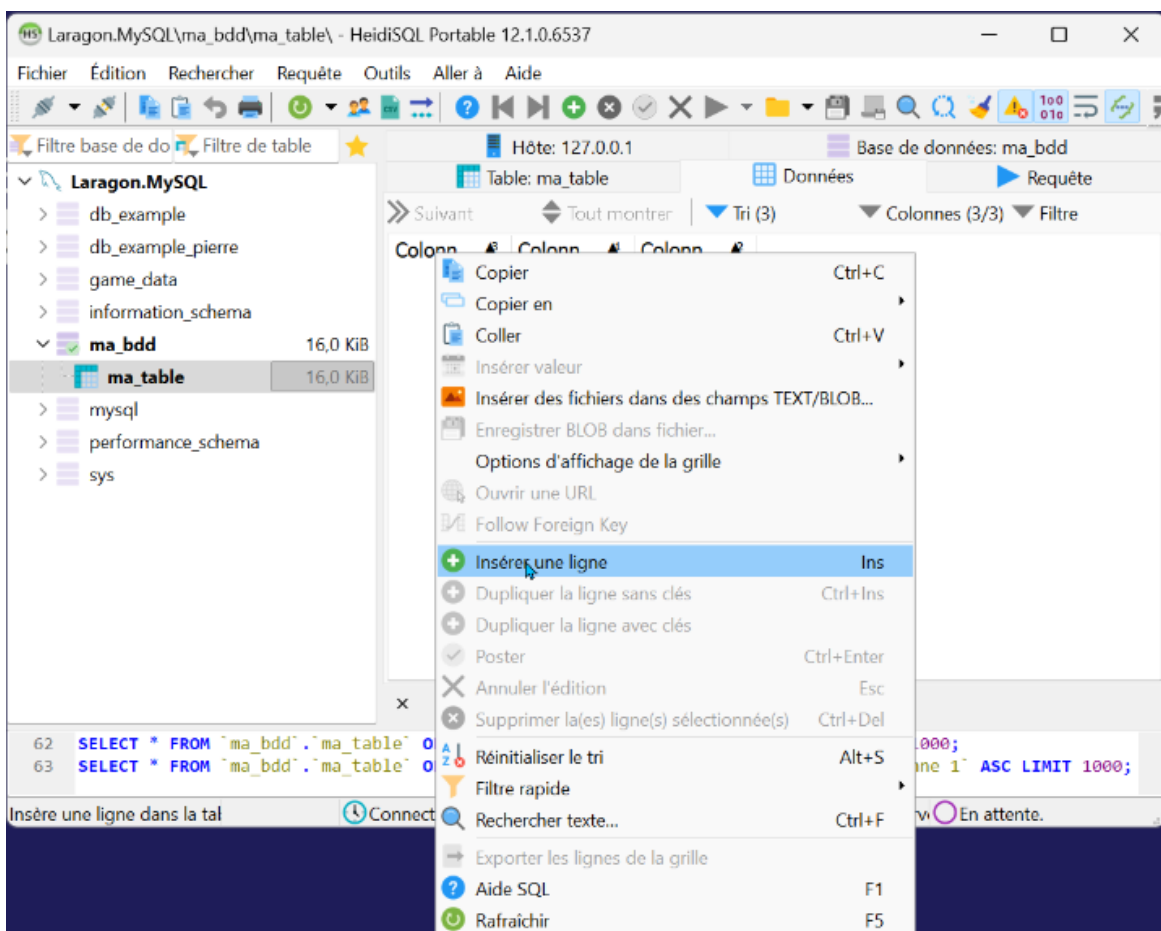
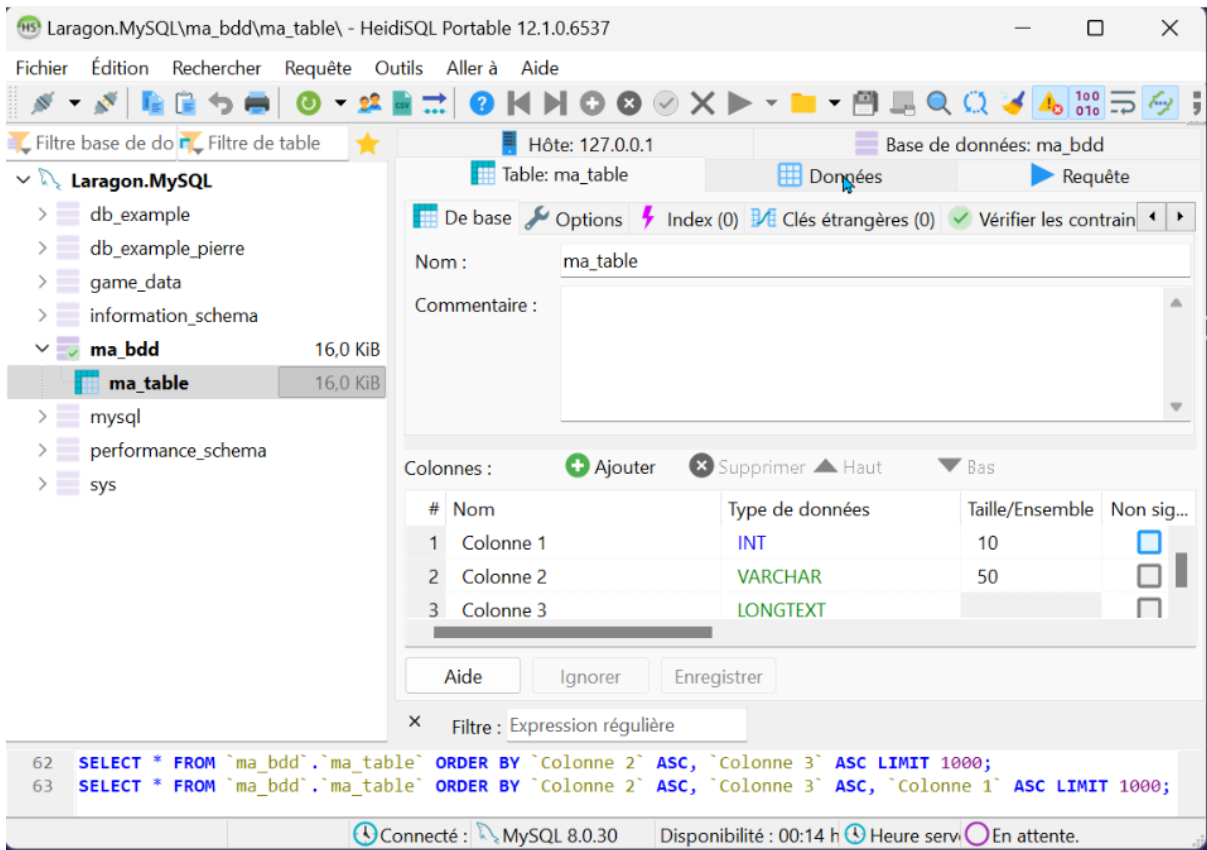
1. Ouvrez un terminal ou une invite de commande.
2. Connectez-vous à votre serveur MySQL en utilisant la commande `mysql -u nom_utilisateur -p`, en remplaçant `nom_utilisateur` par votre nom d'utilisateur MySQL. Vous serez invité à saisir votre mot de passe.
3. Une fois connecté, sélectionnez la base de données `GAME_IMTS` en utilisant la commande `USE GAME_IMTS;`.
4. Créez la table en utilisant la commande `CREATE TABLE` comme décrit précédemment. Par exemple, pour créer une table `users` avec les colonnes `id`, `nom`, `email` et `mot_de_passe`, vous pouvez utiliser la commande suivante :

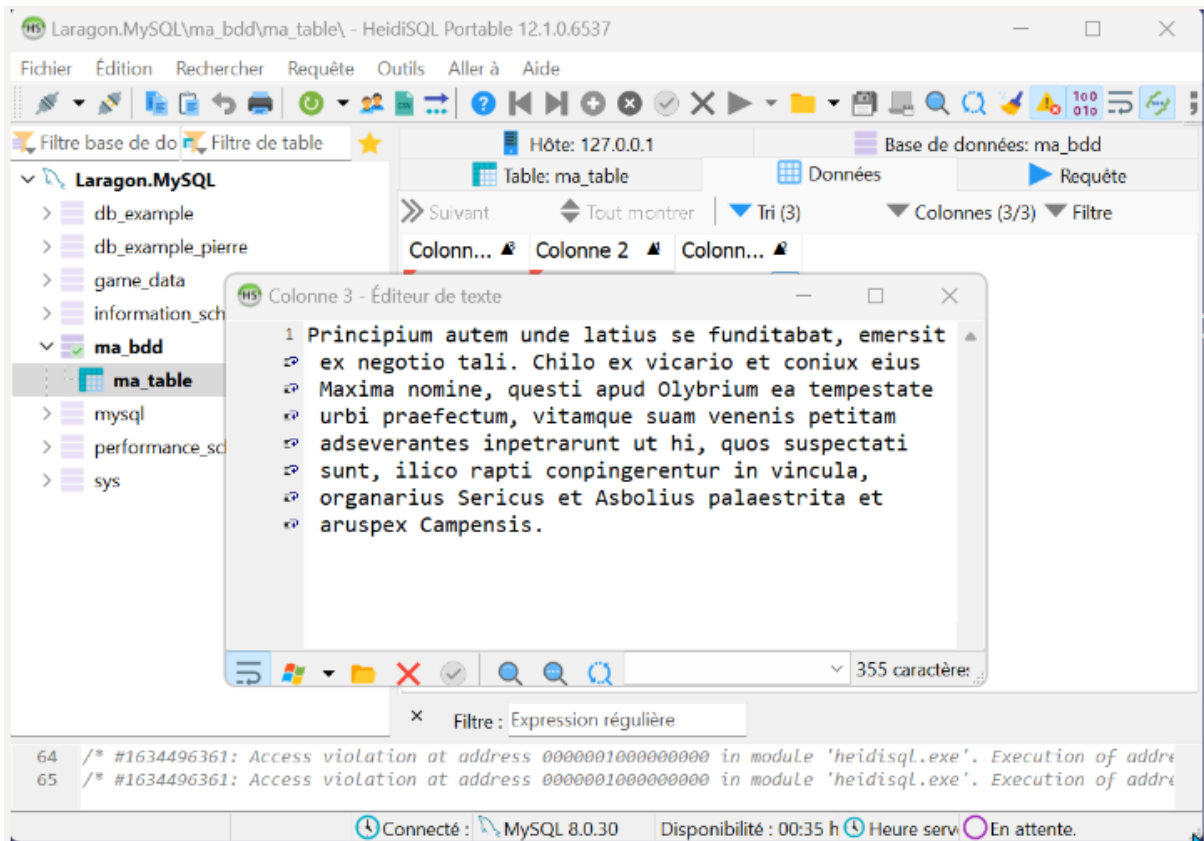
```
CREATE TABLE users (  
  id INT NOT NULL AUTO_INCREMENT,  
  nom VARCHAR(255),  
  email VARCHAR(255) NOT NULL UNIQUE,  
  mot_de_passe VARCHAR(255),  
  PRIMARY KEY (id)  
);
```











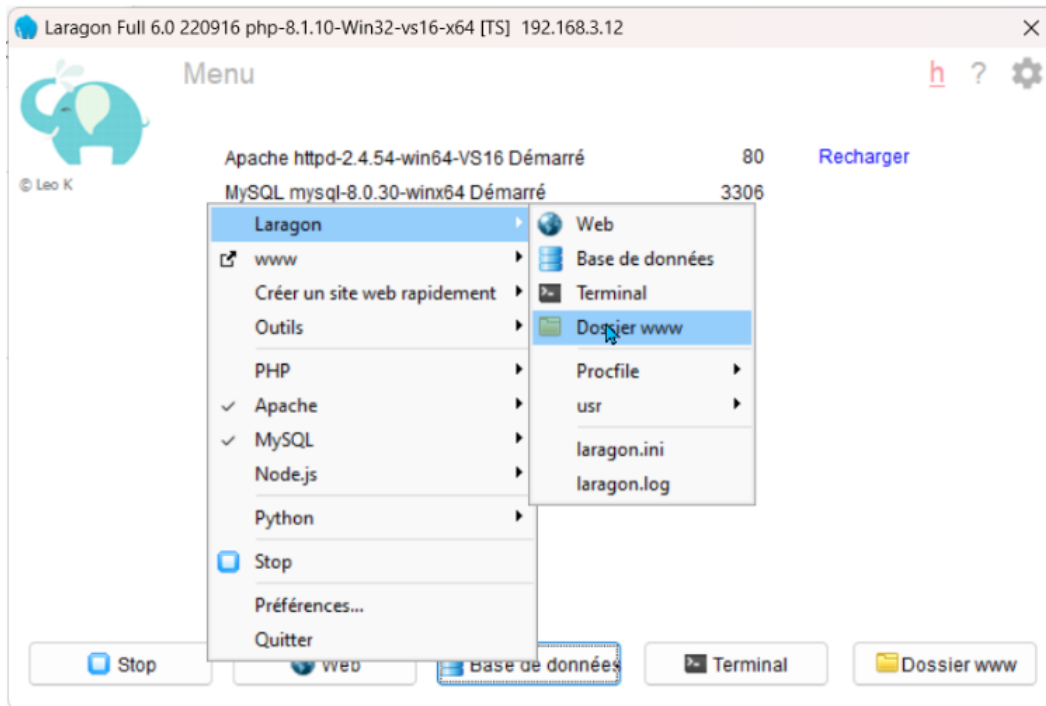
## Résumé:

- Pour démarrer MySQL, ouvrez un terminal et entrez la commande `mysql -u root -p`.
- Vous pouvez créer une nouvelle base de données en utilisant la commande `CREATE DATABASE;`.
- Pour sélectionner une base de données spécifique à utiliser, utilisez la commande `USE mabasededonnee;`.
- Pour afficher les bases de données disponibles, utilisez la commande `SHOW DATABASES;`.
- Une table est un espace dans votre base de données qui stocke des objets de même type.
- Vous pouvez créer une table en utilisant la commande `CREATE TABLE;`. Lorsque vous créez une table, vous devez spécifier le nom, le type et les options de chaque champ.
- Pour vérifier la création des tables, utilisez les commandes `SHOW tables;` et `SHOW COLUMNS FROM table;`.

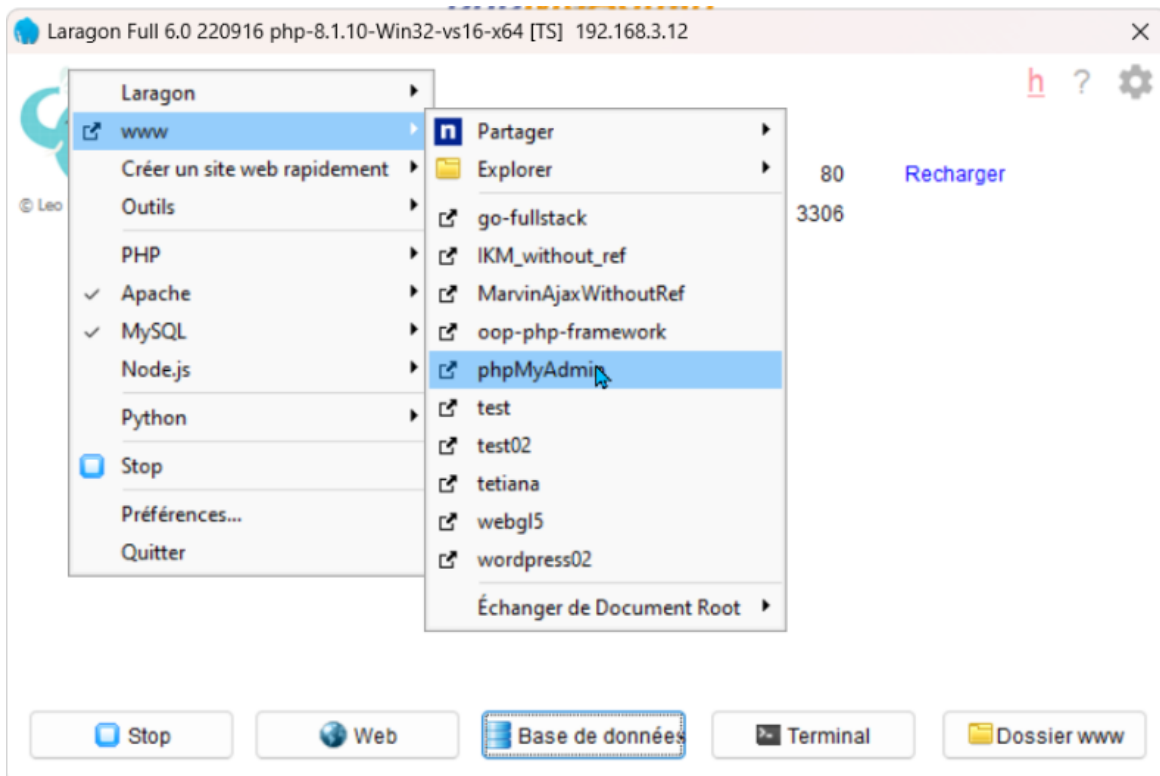
# Installer PHPMyAdmin

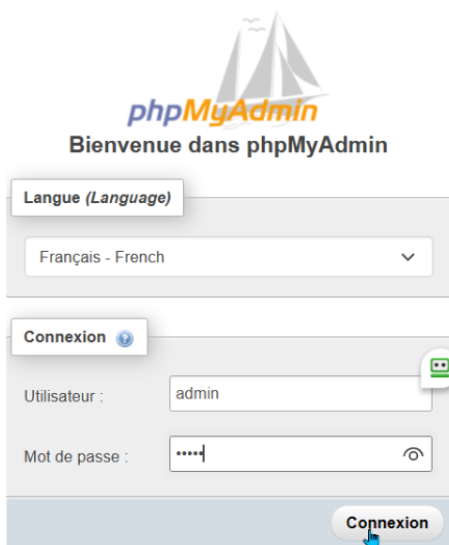
Allez sur le serveur et copier phpMyAdmin.zip

\\192.168.3.5\public\Softwares ou télécharger sur le site [phpMyAdmin](#). Décompresser le zip dans votre répertoire web local

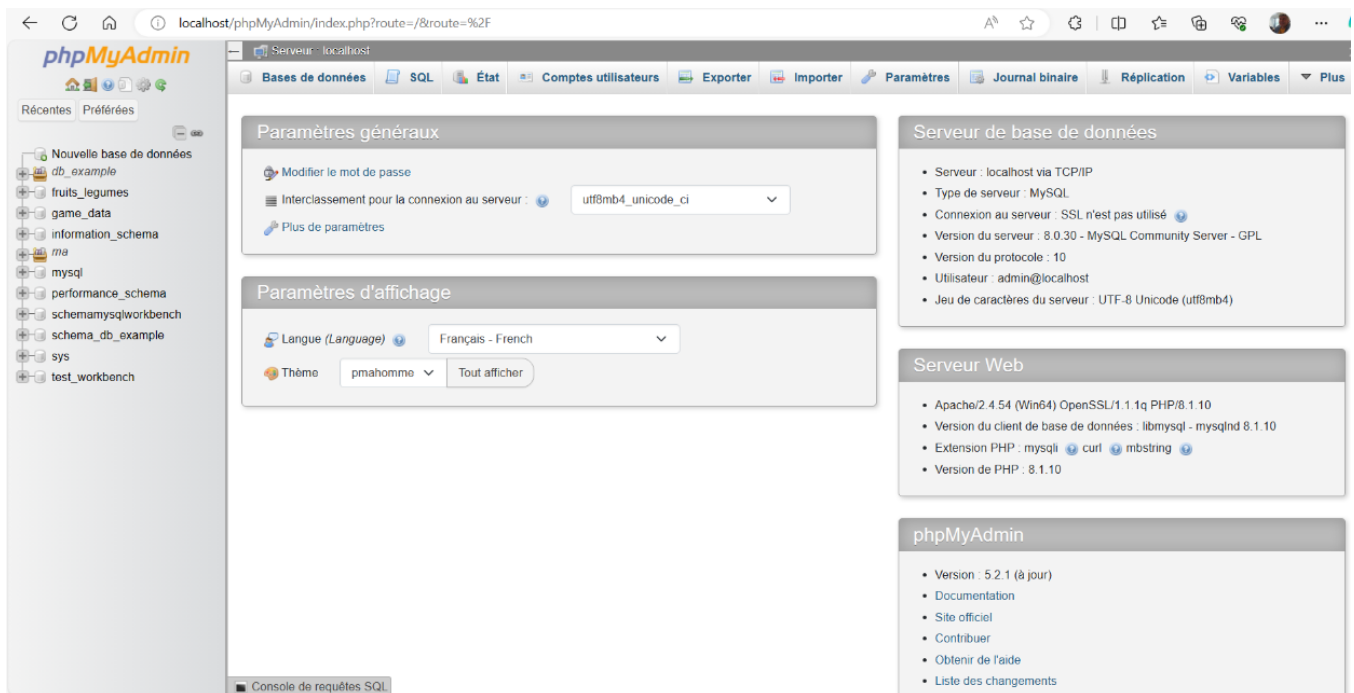


<localhost/phpMyAdmin/>



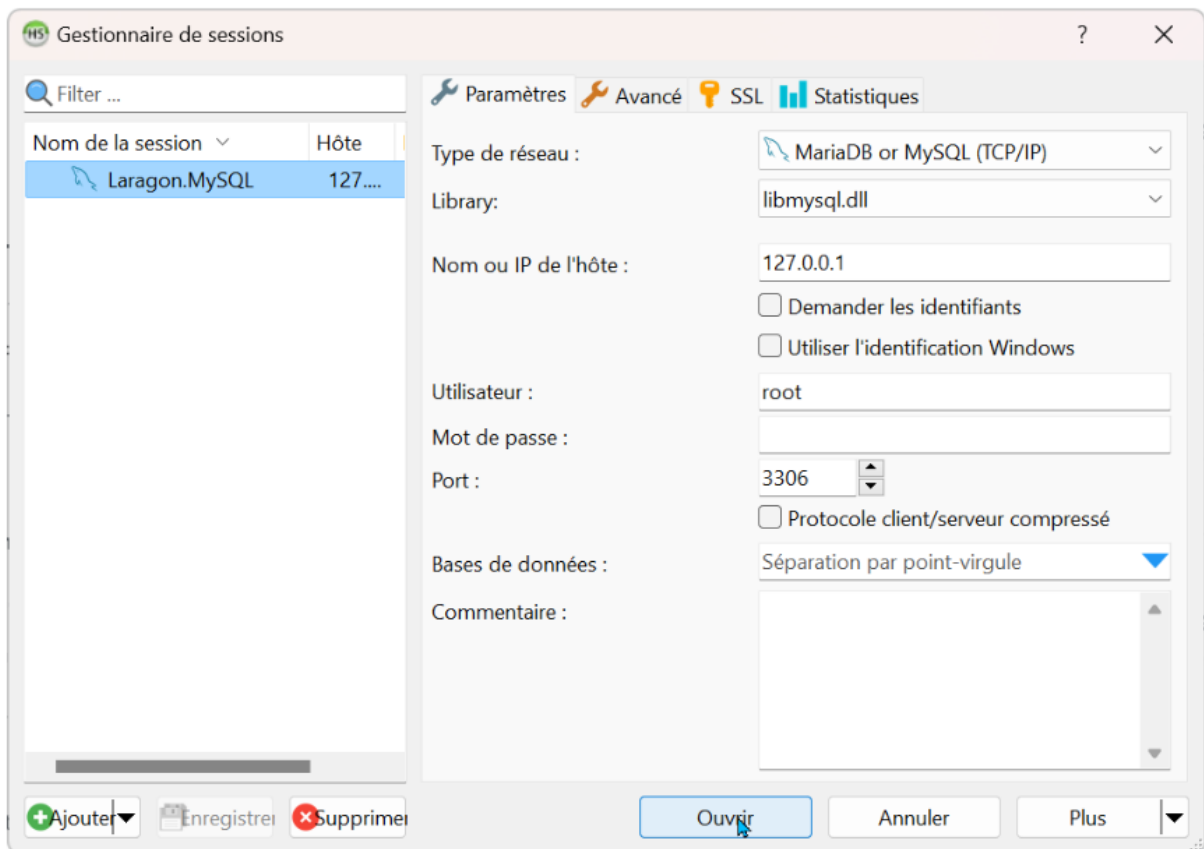
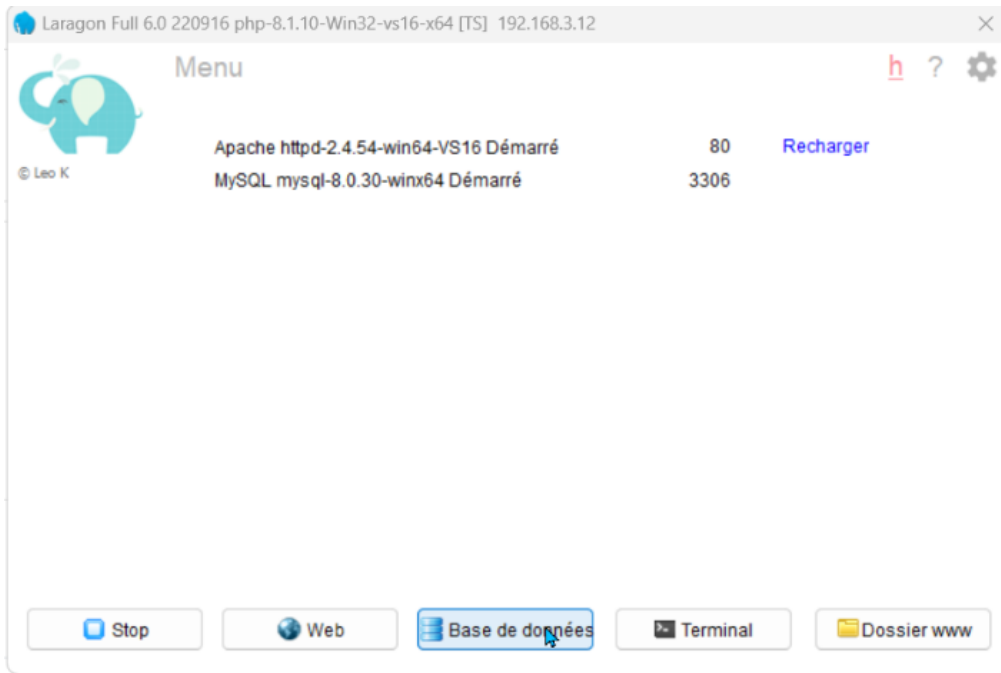


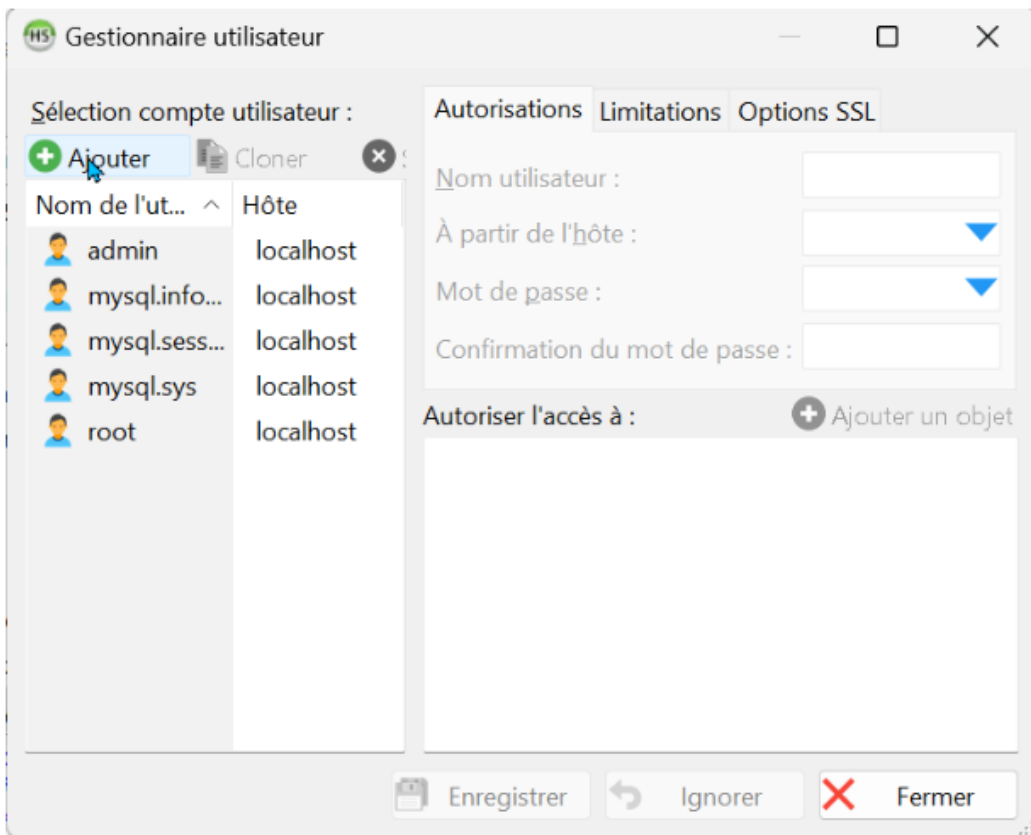
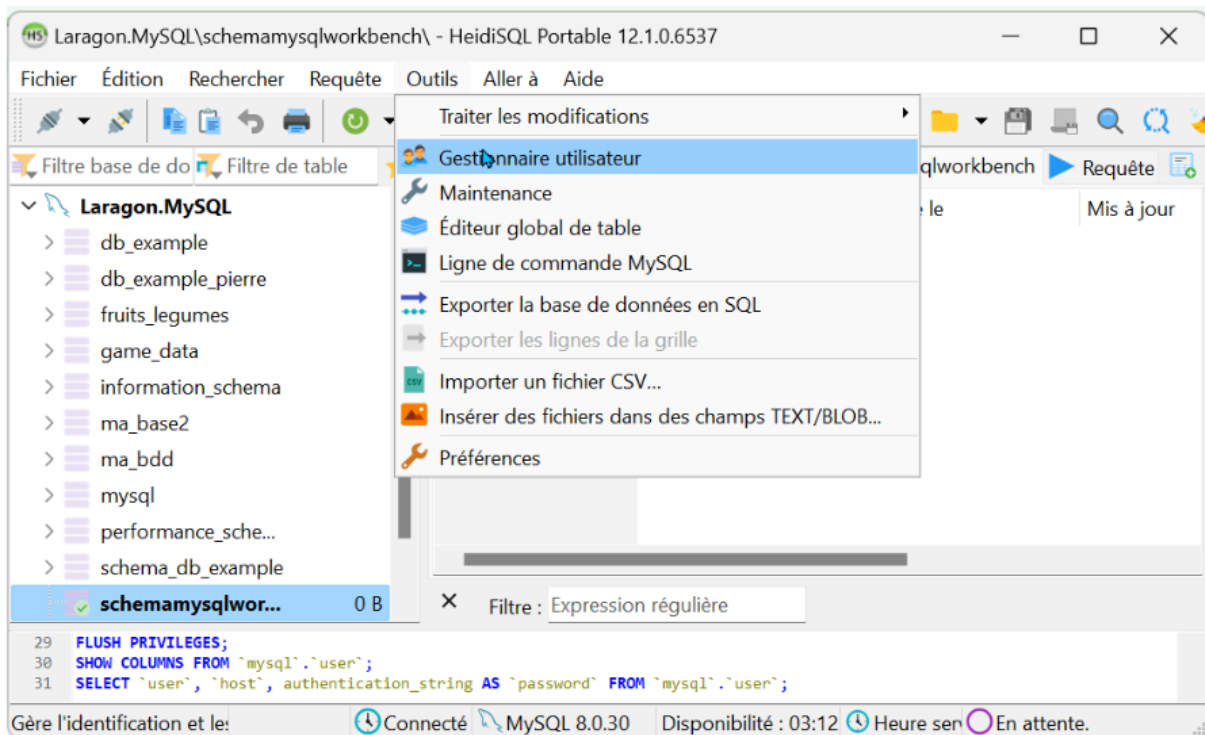
Et voilà !



## **Ajouter un compte + mot de passe avec Laragon**

Pour que PhpMyAdmin fonctionne vous devez ajouter un utilisateur avec mot de passe. Voici comment le faire avec Laragon





Exemple : Ajouter un utilisateur admin avec le mot passe admin.

N'oubliez pas de sélectionner la checkbox Privilèges Globaux

