

Introduction à Laragon

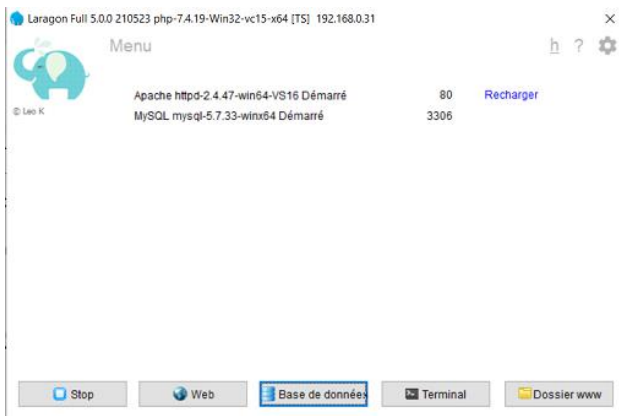
Laragon est un environnement de développement puissant pour la programmation Web. Il permet d'allier le langage PHP et les bases de données sous un seul logiciel. Gratuit, performant, il gère l'installation du **serveur Apache** (le plus utilisé au monde et de grande qualité) et de **MySQL**, base de données largement utilisée notamment par les développeurs indépendants.

On se consacrera dans un premier temps à l'utilisation d'une base de données via Laragon.

I/ L'environnement Laragon (Bases de données)

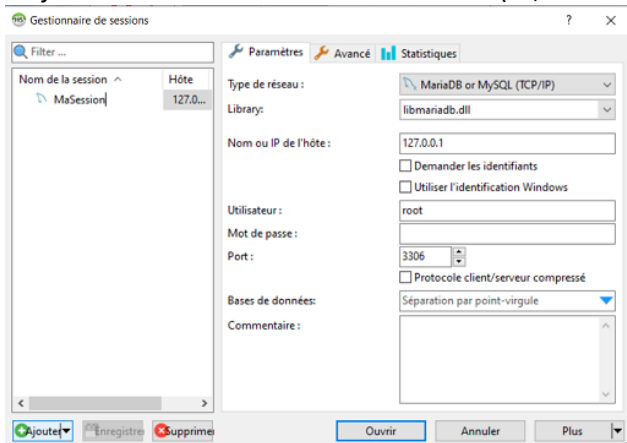
Manipulations :

1/ Laragon est installé sur chaque ordinateur. Il suffit de cliquer sur l'icône et après chargement, la fenêtre suivante s'affiche :



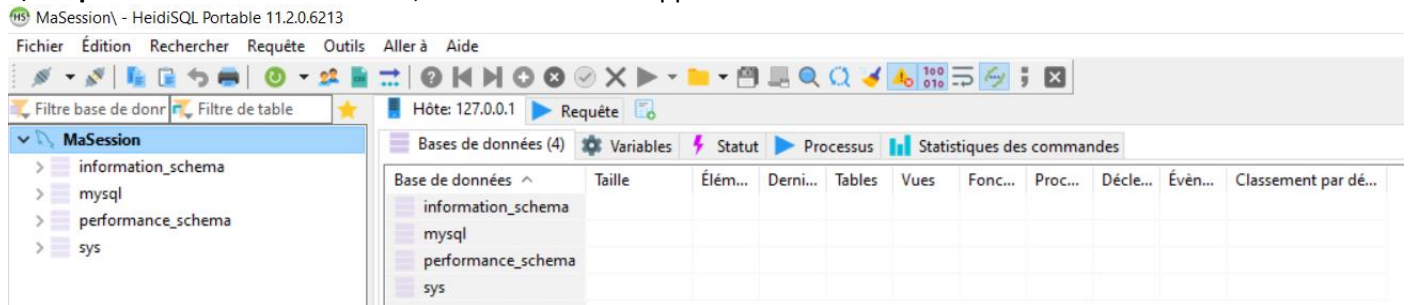
Remarque : la version est peut-être différente mais cela n'est pas important.

2/ **Cliquer** ensuite sur le bouton « *Base de données* ». Il faut créer une **nouvelle session** en cliquant sur le bouton « *Ajouter* ». La nommer comme souhaité (ici, '*MaSession*').

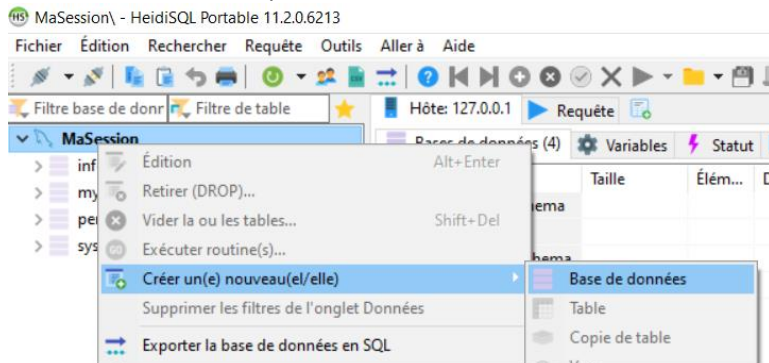


Remarque : on peut cliquer sur la *croix* en haut à droite de la fenêtre de Laragon pour la minimiser (cela n'arrête pas le logiciel).

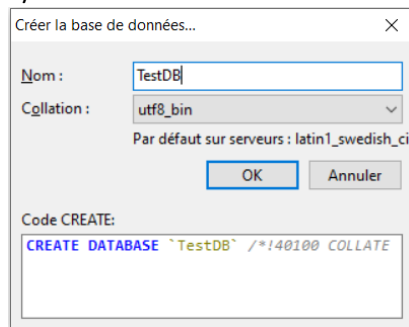
3/ Cliquer sur le bouton « ouvrir », cette fenêtre doit apparaître une fois la session créée :



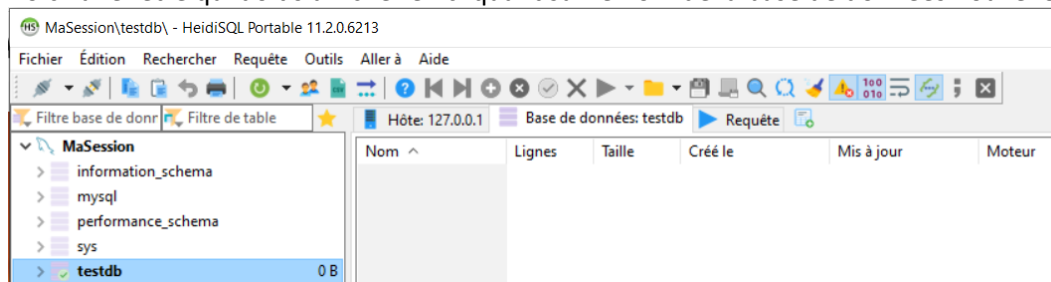
4/ On va maintenant créer une **base de données**. Pour se faire, faire un clic droit sur la session créée, et sélectionner « Créer un nouveau » puis « Base de données », la fenêtre suivante doit s'ouvrir :



5/ Donner un nom à la base de données (ici 'TestDB') et choisir **utf_8** en *collation*.



Voici la fenêtre qui doit s'afficher en cliquant sur le nom de la base de données nouvellement créée :



La base de données est évidemment vide, on va donc créer des tables pour la construire. L'onglet « Requêtes » permet d'écrire les requêtes nécessaires.

II/ Création d'une base de données sous Laragon

Voici un extrait d'une base de données gérant des agences proposant des voitures de location :

Table voitures

id_voiture	id_agence	constructeur	annee	carburant
0	0	Ford	2015	diesel
1	5	Renault	2018	essence
2	0	Ford	2020	essence
4	3	Peugeot	2017	diesel
6	6	Renault	2016	diesel
7	0	Toyota	2021	essence
12	3	Audi	2020	essence
9	6	BMW	2020	hybride
10	7	Ford	2019	essence
3	5	Chrysler	2020	diesel
14	0	Toyota	2018	hybride

Table agences

id_agence	ville	dirigeant	flotte
0	Paris_I	Dupont	55
5	Lyon	Ferrière	82
3	Paris_XII	Dupont	103
6	Bordeaux	Durand	41
7	Tours	Bertrand	39

Table agences :

id_agence : clé primaire, de type INT, en auto-incrémentation,
ville : type VARCHAR (chaîne de caractères de taille variable),
dirigeant : type VARCHAR (chaîne de caractères de taille variable),
flotte : type INT.

Table voitures :

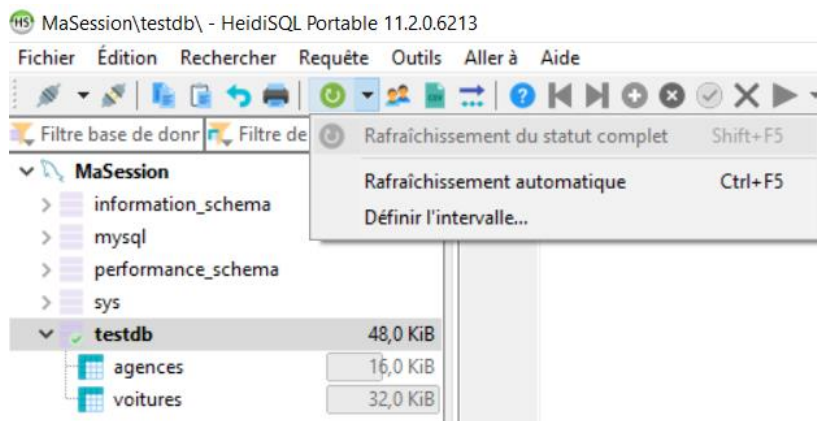
id_voiture : clé primaire, de type INT, en auto-incrémentation,
#id_agence : clé étrangère, de type INT, pointant sur la clé primaire de la table Agences.
constructeur : type VARCHAR (chaîne de caractères de taille variable),
annee : type INT,
carburant : type VARCHAR (chaîne de caractères de taille variable).

1/ Voici les **requêtes MySQL** permettant de créer les deux tables ci-dessus :

```
CREATE TABLE IF NOT EXISTS agences (  
    id_agence INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ville VARCHAR(30) COLLATE UTF8mb4_UNICODE_CI,  
    dirigeant VARCHAR(30) COLLATE UTF8mb4_UNICODE_CI,  
    flotte INT  
)  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS voitures(  
    id_enseignant INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    id_agence INT NOT NULL,  
    constructeur VARCHAR(30) COLLATE UTF8mb4_UNICODE_CI,  
    annee INT,  
    FOREIGN KEY(id_agence) REFERENCES Agences(id_agence)  
)  
ENGINE = InnoDB;
```

Les **recopier** dans **Notepad++** (en extension .sql) puis copier-coller dans l'onglet « Requêtes » et les exécuter.
Le fichier pourra être ainsi sauvegardé.

2/ Après un rafraîchissement (voir ci-dessous), les deux tables nouvellement créées doivent apparaître :

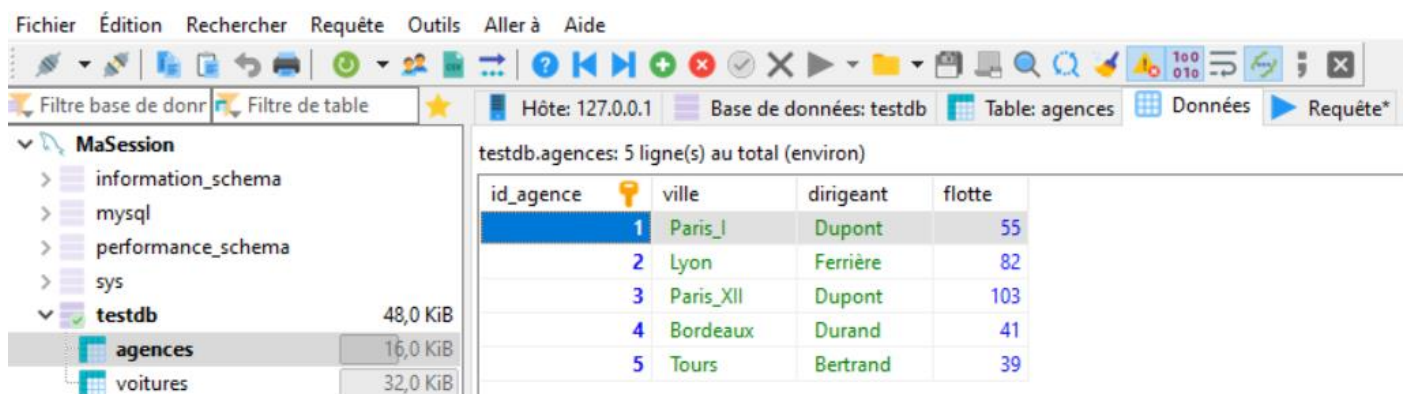


Remarque : il faudra rafraîchir la fenêtre assez régulièrement.

3/ Voici la requête permettant d'insérer les valeurs dans la table *agences* :

```
INSERT INTO agences(ville,dirigeant,flotte) VALUES
("Paris_I","Dupont",55),
("Lyon","Ferrière",82),
("Paris_XII","Dupont",103),
("Bordeaux","Durand",41),
("Tours","Bertrand",39);
```

La **recopier**. Après rafraîchissement, on peut accéder aux nouvelles données entrées dans la table *agences*.



Remarque : les valeurs de *id_agence* sont différentes de celle de la table présentée : c'est normal, il s'agit d'une auto-incrémentation.

4/ Dans un **autre onglet** « Requetes », écrire la requête permettant d'insérer les valeurs de la table *voitures*.

Attention à bien adapter les valeurs de la clé étrangère.

Penser à **rafraîchir** ☺ et **vérifier** que les lignes sont bien enregistrées dans la table.

Appeler le professeur (si besoin)

III/ Requêtes dans une base de données sous MySQL

La base de données sous MySQL suit bien entendu la structure du langage SQL avec quelques points spécifiques qui simplifient les requêtes (jointures notamment).

Quelques requêtes :

```
# Afficher la table agences
SELECT * FROM agences;

# Afficher les villes où Dupont est dirigeant d'une agence
SELECT ville FROM agences WHERE dirigeant = "Dupont";

# Compter le nombre d'agences dirigées par Dupont
SELECT COUNT(id_agence) FROM agences
WHERE dirigeant = "Dupont";

# L'agence 1 a désormais 58 voitures
UPDATE agences
SET flotte = 58
WHERE id_agence = 1;

# Pour vérifier la requête précédente
SELECT flotte FROM agences
WHERE id_agence = 1;

# Affiche les constructeurs proposés par Dupont
SELECT DISTINCT constructeur FROM voitures
JOIN agences USING(id_agence)
WHERE agences.dirigeant = "Dupont";
```

Requêtes à écrire :

- Afficher la liste des voitures.
- Afficher les années de construction des voitures « Ford ».
- Compter le nombre de Renault.
- L'agence de Bordeaux compte désormais 46 véhicules. Mettre la table à jour et vérifier le résultat.
- Afficher les agences proposant des voitures « Renault ».
- Afficher les agences ayant des véhicules récents (2019 au moins).

Appeler le professeur