

# Developpement Web

## HeidiSQL

### Introduction :

L'objectif principal de ce projet est de maîtriser l'utilisation d'HeidiSQL pour gérer et manipuler des bases de données, un outil essentiel pour le développement web. Le but est d'appliquer des compétences acquises en SQL tout en réalisant des actions manuelles et via des requêtes dans HeidiSQL. Ce travail permettra également d'approfondir la compréhension des concepts de base de données tels que les clés primaires, les relations entre tables, et les colonnes/attributs.

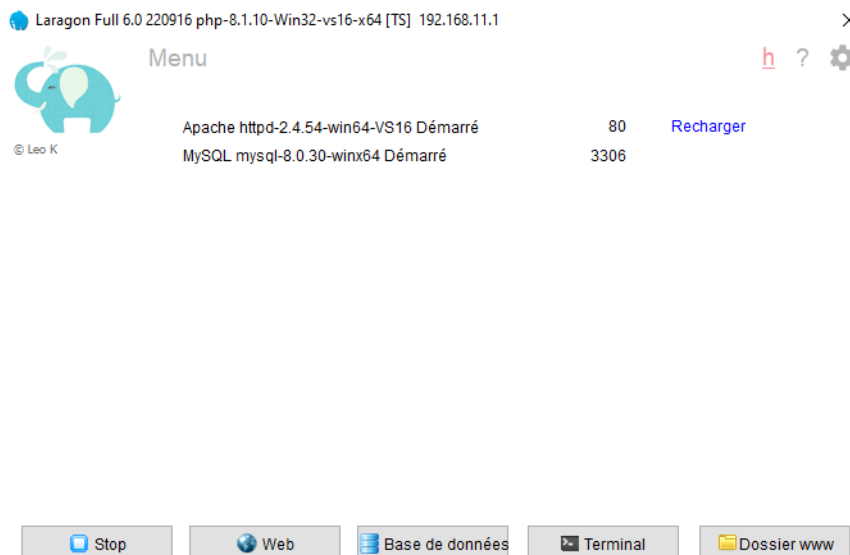
### Questions et réponses :

1. **Quels sont les services qui ont été démarrés ?**
  - Réponse : *Le serveur web Apache et le serveur MySQL.*
  
2. **Justifiez l'IP de l'hôte et le port renseignés.**
  - Réponse : *Nous sommes en localhost avec le port 3306.*
  
3. **À quoi correspond une colonne dans une table ?**
  - Réponse : *Une colonne représente un attribut. Chaque colonne contient un type de données spécifique (par exemple, un entier, un texte, etc.).*
  
4. **À quoi correspond une ligne dans une table ?**
  - Réponse : *Une ligne contient les valeurs des attributs de chaque colonne pour un enregistrement donné.*
  
5. **Dans l'exemple affiché ci-dessus, à quoi correspond la clé devant la première colonne?**
  - Réponse : *La clé correspond à une clé primaire. Une clé primaire est un champ ou un ensemble de champs dans une table qui contient des valeurs uniques.*

## Objectifs et travail réalisé:

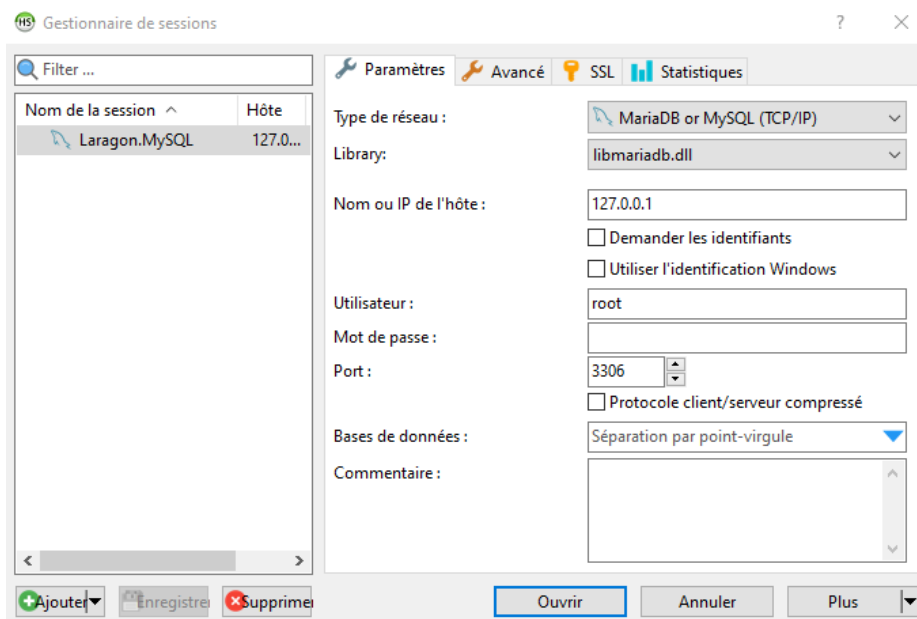
### Configuration et démarrage des services nécessaires

L'une des premières étapes du TP consiste à configurer l'environnement de travail. Cela inclut le démarrage des services comme Apache pour le serveur web et MySQL pour la base de données. Ces services sont essentiels pour pouvoir interagir avec HeidiSQL, car ce dernier nécessite une base de données en cours d'exécution pour permettre la connexion et l'exécution des requêtes.



### Connexion à la base de données via HeidiSQL

HeidiSQL permet de se connecter à un serveur de base de données local ou distant. Ici, nous utilisons "localhost" avec le port 3306 par défaut pour MySQL. L'objectif est d'établir une connexion stable afin de pouvoir interagir avec la base de données et y exécuter des requêtes.



## Création d'une base de données et des tables

L'un des objectifs majeurs est d'apprendre à créer une base de données manuellement et via des requêtes SQL. Dans ce TP, nous devons créer une base de données et y ajouter deux tables : "Client" et "Commande". Chaque table est composée de colonnes avec des types de données spécifiques, et la relation entre ces tables est établie via une clé étrangère.

</

## Gestion des relations entre tables

Un autre objectif est de comprendre comment gérer les relations entre différentes tables. Ici, la relation est établie entre la table "Client" et la table "Commande". Cela se fait en définissant une clé étrangère (idClient) dans la table "Commande" qui fait référence à l'identifiant du client dans la table "Client". Ce type de relation est essentiel pour structurer des bases de données complexes et maintenir l'intégrité des données.

Nom > Index 1 > FK_commande_client		Type / Taille KEY KEY	Algorithme BTREE BTREE	Commentaire
--	--	-----------------------------	------------------------------	-------------

## Exécution de requêtes SQL

Un autre objectif clé est d'apprendre à interagir avec la base de données via des requêtes SQL. Après avoir créé les tables, nous devons insérer des données dans celles-ci en utilisant des commandes INSERT. Par exemple, nous devons insérer des clients et leurs commandes. Les requêtes permettent également de récupérer les informations, les modifier ou les supprimer.

Création de la table « medecin »

```
CREATE TABLE Medecin (  
    idMedecin INT,  
    nom VARCHAR(50),  
    PRENOM VARCHAR(50),  
    adresse VARCHAR(50)  
);
```

Création de la table « patient »

```
CREATE TABLE Patient (  
    idPatient INT,  
    nomPatient VARCHAR(50),  
    prenom VARCHAR (50),  
    idMedecin INTnewtp  
);
```

Insertion dans la table « medecin » du medecin Yakub

```
1 INSERT INTO medecin  
2 VALUE (1,'ONAL','Yakub','Rive de Gier');|
```

Insertion dans la table « patient » du patient Yves avec le medecin Yakub

```
1 INSERT INTO patient  
2 VALUE (1,'DUPONT','Yves',1);|
```

Insertion dans la table « patient » du patient Yvette avec le medecin Yakub

```
INSERT INTO patient  
VALUE (2,'DUPONT','Yvette',1);
```

## Conclusion :

Ce TP m'a permis de consolider mes connaissances sur la gestion des bases de données et les requêtes SQL. L'utilisation d'HeidiSQL s'avère très pratique pour effectuer des opérations sur la base de données de manière efficace. J'ai appris à gérer des relations entre tables, à manipuler des données via des requêtes SQL et à configurer un environnement de base de données fonctionnel. Les difficultés rencontrées concernaient principalement l'établissement des relations entre les tables et l'utilisation correcte des types de données. Cependant, ces obstacles m'ont permis de renforcer ma compréhension globale des bases de données relationnelles.

Yakub ONAL