



**SQL**

PARTIE 1

CRÉER | MODIFIER | SUPPRIMER

- 
- Introduction
  - Préparation
  - Les tables
  - Créer une table
  - Clé primaire
  - Auto\_increment
  - Les types de données
  - Insérer des données
  - Modifier des données
  - Supprimer des données

Le langage SQL (Structured Query Language) est un langage informatique standard utilisé pour communiquer avec des bases de données relationnelles. SQL est un langage de programmation très puissant qui permet de manipuler, d'interroger et de gérer des données stockées dans une base de données.

Le langage SQL a été développé dans les années 1970 et a depuis été utilisé dans de nombreuses applications et systèmes de gestion de bases de données. Il est largement utilisé dans les applications web pour stocker et récupérer des données telles que des informations utilisateur, des articles de blog, des produits de commerce électronique et bien plus encore.

SQL permet aux utilisateurs de définir des schémas de base de données, de créer, de modifier et de supprimer des données dans une base de données, ainsi que de rechercher des informations spécifiques. Les requêtes SQL permettent d'extraire des données en fonction de critères spécifiques, tel que la recherche d'un utilisateur par son nom ou la récupération de toutes les ventes réalisées entre deux dates.

Le langage SQL est facile à apprendre et à utiliser, même pour les débutants en programmation. La syntaxe est simple et les concepts fondamentaux sont facilement compréhensibles. SQL est un outil essentiel pour tout développeur ou analyste travaillant avec des bases de données relationnelles.

En conclusion, SQL est un langage de programmation populaire et largement utilisé dans le monde de la technologie pour interagir avec les bases de données relationnelles. Il permet aux utilisateurs de stocker et de récupérer des données de manière efficace et offre de nombreuses fonctionnalités pour manipuler et interroger les données stockées dans une base de données.

Pour apprendre à utiliser mySql nous allons installer Xampp.

XAMPP est un package logiciel open-source qui est utilisé pour créer et gérer des environnements de développement web locaux. Il contient une variété de logiciels open-source, tels que Apache, MySQL, PHP, Perl et d'autres, qui sont nécessaires pour créer et exécuter des applications web. Dans cet article, je vais vous expliquer comment installer XAMPP sur votre ordinateur.

## Étape 1: Téléchargement de XAMPP

Tout d'abord, téléchargez la dernière version de XAMPP à partir du site officiel : <https://www.apachefriends.org/index.html>.

## XAMPP Apache + MariaDB + PHP + Perl

**What is XAMPP?**

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

**Download**  
Click here for other versions

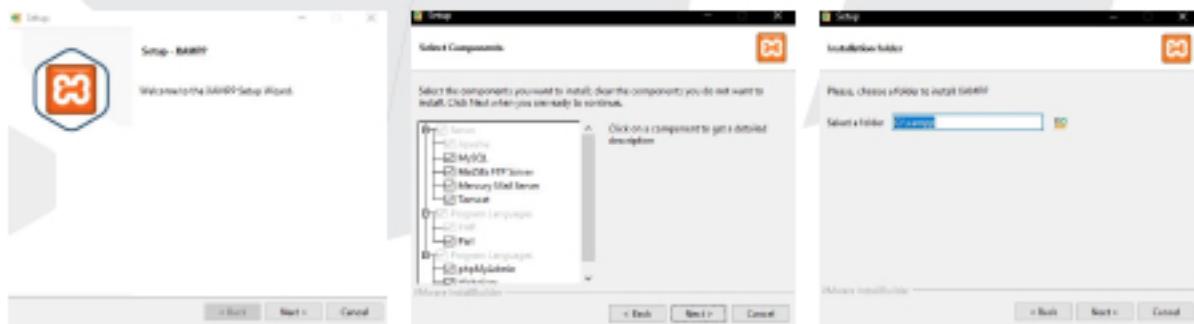
 XAMPP for Windows  
8.2.0 (PHP 8.2.0)

 XAMPP for Linux  
8.2.0 (PHP 8.2.0)

 XAMPP for OS X  
8.2.0 (PHP 8.2.0)

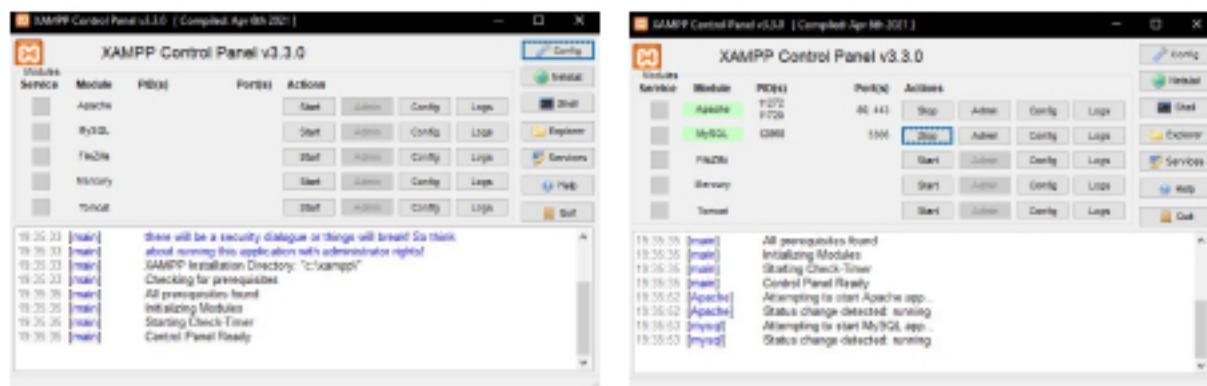
## Étape 2: Installer XAMPP

1. Une fois le téléchargement terminé, double-cliquez sur le fichier d'installation pour lancer l'assistant d'installation
2. Dans l'assistant d'installation, cliquez sur le bouton "Suivant" pour continuer.
3. Choisissez les composants que vous souhaitez installer en cochant les cases correspondantes. Par défaut, tous les composants sont sélectionnés, mais vous pouvez en désélectionner certains si vous le souhaitez.
4. Choisissez le répertoire d'installation pour XAMPP. Par défaut, il est installé dans le répertoire C:\xampp sur Windows.
5. Cliquez sur le bouton "Installer" pour lancer l'installation.
6. Attendez que l'installation soit terminée. Cela peut prendre plusieurs minutes en fonction de la vitesse de votre ordinateur.
7. Une fois l'installation terminée, vous pouvez lancer XAMPP en double-cliquant sur l'icône de raccourci XAMPP sur votre bureau.

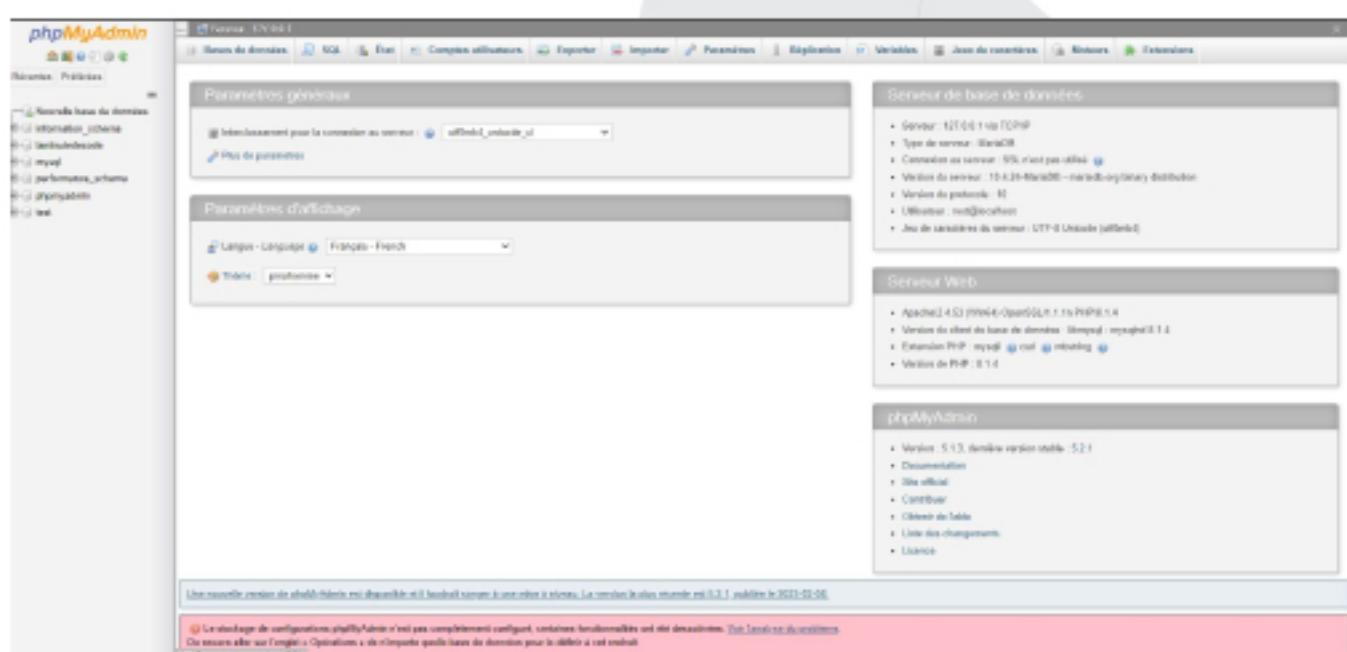


## Étape 2: lancer mySql depuis notre serveur local

1. Une fois XAMPP installé sur votre ordinateur, vous pouvez facilement lancer MySQL en suivant les étapes suivantes :
2. Ouvrez XAMPP en double-cliquant sur l'icône de raccourci XAMPP sur votre bureau.
3. Dans la fenêtre de contrôle XAMPP, cliquez sur le bouton "Start" à côté de "MySQL". Si MySQL n'est pas démarré, le bouton "Start" sera vert. Si MySQL est déjà en cours d'exécution, le bouton sera grisé et indiquera "Running".
4. Attendez que MySQL soit lancé. Cela peut prendre quelques secondes pour que MySQL se connecte à la base de données.
5. Une fois que MySQL est lancé, vous pouvez accéder à la base de données en utilisant un client MySQL tel que phpMyAdmin ou en utilisant une commande MySQL dans un terminal ou une invite de commande.
6. Pour accéder à phpMyAdmin, suivez les étapes suivantes :
7. Ouvrez votre navigateur Web et accédez à l'URL <http://localhost/phpmyadmin/>.
8. Dans la page d'accueil de phpMyAdmin, vous pouvez gérer vos bases de données MySQL, y compris la création de nouvelles bases de données, la modification des tables existantes et l'exécution de requêtes SQL.
9. Pour vous connecter à votre base de données MySQL, vous devrez fournir votre nom d'utilisateur et votre mot de passe MySQL, que vous pouvez trouver dans le fichier de configuration de XAMPP (par défaut, il se trouve dans le répertoire C:\xampp\mysql\bin\my.ini sur Windows).



Vous devriez arriver sur cette page:



The screenshot shows the phpMyAdmin dashboard with the following sections visible:

- Bases de données:** Information\_schema, mysql, performance\_schema, sys, sysbench, test.
- Préférences:** Réinitialiser les préférences.
- Paramètres généraux:** Sélectionnez pour la connexion au serveur: « localhost\_unicode » et « Plus de paramètres ».
- Paramètres d'affichage:** Langue - Langage: Français - French; Thème: sombre.
- Serveur de base de données:**
  - Général: 127.0.0.1 via TCP/IP
  - Type de serveur: MariaDB
  - Connexion au serveur: MySQL n'est pas utilisé.
  - Méthode de reprise: 10.4.11-MariaDB - mariaDB-binary distribution
  - Version du protocole: 10
  - Utilisateur: root@localhost
  - Jeu de caractères du serveur: UTF-8 Unicode (affichage)
- Serveur Web:**
  - Apache2.4.53 (Ubuntu)/OpenSSL/1.1.1n PHP8.1.4
  - Version du client de base de données: MariaDB - mariadb-10.1.4
  - Extension PHP: mysql (mod) mbstring (mod)
  - Version de PHP: 8.1.4
- phpMyAdmin:**
  - Mise à jour: 5.1.3, dernière version stable: 5.2.1
  - Documentation
  - Site officiel
  - Contribuer
  - Obtenir des Table
  - Liste des changements
  - Licence

Messages en bas de page:

- Une nouvelle version de phpMyAdmin est disponible et il faudrait mettre à jour votre système. La version la plus récente est 5.2.1 publiée le 2023-02-06.
- Le téléchargement de configurations phpMyAdmin n'est pas complètement configuré, certaines fonctionnalités ont été désactivées. Voir TUTO sur la sécurité. Cela nécessite une migration à Operations & de configurer quelques bases de données pour le délier à cet endroit.
- Cliquez sur Consultez les résultats SQL.

En MySQL, une table est un objet de base de données qui permet de stocker des données sous forme de lignes et de colonnes. Chaque table est composée d'un ensemble de colonnes qui représentent les différents types de données que l'on souhaite stocker, ainsi que d'un ensemble de lignes qui représentent les enregistrements individuels.

Par exemple, si l'on souhaite stocker des informations sur des clients dans une base de données MySQL, on peut créer une table "clients" avec des colonnes telles que "nom", "prénom", "adresse", "ville", "code postal", etc. Chaque enregistrement dans la table "clients" correspondrait alors à un client individuel, avec ses propres valeurs pour chaque colonne.

Pour créer une table dans MySQL, on utilise la commande SQL "CREATE TABLE". On spécifie le nom de la table, ainsi que les noms et les types de données des colonnes, en utilisant la syntaxe SQL appropriée. On peut également ajouter des contraintes de clé primaire, de clé étrangère, de valeur par défaut, etc. pour garantir l'intégrité des données et la cohérence entre les tables.

Une fois créée, une table peut être manipulée en utilisant des commandes SQL telles que "INSERT" pour ajouter des enregistrements, "SELECT" pour récupérer des données, "UPDATE" pour modifier des données existantes, ou "DELETE" pour supprimer des enregistrements.

Une clé primaire (ou "primary key" en anglais) dans MySQL est une colonne ou un ensemble de colonnes qui identifient de manière unique chaque enregistrement dans une table. La clé primaire est utilisée pour garantir l'intégrité des données, car elle assure que chaque enregistrement est unique et que les données ne sont pas dupliquées dans la table.

Lors de la création d'une table dans MySQL, on peut spécifier une ou plusieurs colonnes comme clé primaire en utilisant la clause "PRIMARY KEY". MySQL permet également d'utiliser une clé primaire composée de plusieurs colonnes en spécifiant plusieurs colonnes dans la clause "PRIMARY KEY".

La clé primaire est souvent utilisée comme référence dans les relations entre tables, afin de garantir l'intégrité référentielle des données. Par exemple, si une table "commandes" contient une colonne "client\_id" faisant référence à la clé primaire de la table "clients", MySQL garantit que les commandes ne peuvent pas être associées à un client qui n'existe pas dans la table "clients".

Voici un exemple de commande SQL pour créer une table nommée "shop" dans MySQL, avec deux colonnes "id" et "ville", et la clé primaire sur la colonne "id" :

```
CREATE TABLE shops (
    shop_id VARCHAR(30) PRIMARY KEY,
    shop_ville VARCHAR(30)
);
```

Dans cet exemple, la colonne "nom\_magasin" a été définie comme clé primaire en utilisant la clause "PRIMARY KEY". Le type de données utilisé pour cette colonne est "VARCHAR(50)", qui permet de stocker des chaînes de caractères jusqu'à 50 caractères de long. La colonne "ville" est également définie comme une colonne de type "VARCHAR(30)".

Notez que vous pouvez personnaliser les noms des colonnes et les types de données en fonction de vos besoins. De plus, vous pouvez ajouter d'autres contraintes et options à la table, telles que des contraintes de clé étrangère ou des index, en utilisant la syntaxe SQL appropriée.

VARCHAR est un type de données dans MySQL qui permet de stocker des chaînes de caractères de longueur variable.

L'avantage de VARCHAR est qu'il utilise la quantité minimale d'espace de stockage nécessaire pour stocker une chaîne de caractères donnée. Par exemple, si vous avez une colonne qui stocke des noms d'utilisateur, la longueur des noms peut varier considérablement.

En utilisant VARCHAR, vous pouvez définir la longueur maximale attendue de la chaîne (par exemple VARCHAR(30) pour une longueur maximale de 30 caractères) et MySQL n'allouera que l'espace nécessaire pour chaque valeur stockée.

Il est important de noter que la longueur maximale de la chaîne définie pour une colonne VARCHAR ne doit pas dépasser 65 535 caractères dans MySQL. Si vous avez besoin de stocker des chaînes plus longues, vous pouvez utiliser le type de données TEXT, qui permet de stocker des chaînes de caractères plus grandes.

Lorsque vous créez une table dans MySQL, vous devez choisir le type de données approprié pour chaque colonne en fonction du type de données que vous souhaitez stocker. Si vous prévoyez de stocker des chaînes de longueur variable, VARCHAR est généralement le type de données le plus approprié.

Pour ajouter un nouveau champ dans une table SQL, vous devez exécuter une instruction ALTER TABLE.

L'instruction ALTER TABLE vous permet de modifier la structure d'une table existante en ajoutant, supprimant ou modifiant des colonnes.

Voici la syntaxe de base pour ajouter une colonne à une table :

```
ALTER TABLE nom_de_la_table  
ADD nom_du_champ type_de_donnees;
```

Explications :

- nom\_de\_la\_table : le nom de la table à laquelle vous souhaitez ajouter un nouveau champ.
- nom\_du\_champ : le nom que vous souhaitez donner au nouveau champ.
- type\_de\_donnees : le type de données que vous souhaitez stocker dans le nouveau champ (par exemple, VARCHAR, INTEGER, DATE, etc.).

Voici une liste des types de données les plus courants dans MySQL, avec une brève description pour chacun :

- INT : Un type de données qui permet de stocker des nombres entiers, positifs ou négatifs, de taille fixe. La taille d'un INT peut varier de 1 à 8 octets, selon le nombre maximum que vous souhaitez stocker.
- BIGINT : Un type de données qui permet de stocker des nombres entiers plus grands que ceux pouvant être stockés dans un INT, de taille fixe. La taille d'un BIGINT peut varier de 1 à 8 octets, selon le nombre maximum que vous souhaitez stocker.
- FLOAT : Un type de données qui permet de stocker des nombres à virgule flottante, positifs ou négatifs, avec une précision de 7 décimales. Le FLOAT utilise 4 octets pour le stockage.
- DOUBLE : Un type de données qui permet de stocker des nombres à virgule flottante, positifs ou négatifs, avec une précision de 15 décimales. Le DOUBLE utilise 8 octets pour le stockage.
- DECIMAL : Un type de données qui permet de stocker des nombres à virgule fixe, positifs ou négatifs, avec une précision de jusqu'à 65 décimales. Le DECIMAL utilise un nombre variable d'octets pour le stockage, en fonction de la précision spécifiée.
- DATE : Un type de données qui permet de stocker une date, au format YYYY-MM-DD.
- TIME : Un type de données qui permet de stocker une heure, au format HH:MM:SS.
- DATETIME : Un type de données qui permet de stocker une date et une heure, au format YYYY-MM-DD HH:MM:SS.
- TIMESTAMP : Un type de données qui permet de stocker une date et une heure, au format YYYY-MM-DD HH:MM:SS. Le TIMESTAMP est automatiquement mis à jour avec l'heure actuelle chaque fois qu'une ligne est insérée ou mise à jour.
- ENUM : Un type de données qui permet de stocker une liste de valeurs possibles pour une colonne donnée. Les valeurs doivent être spécifiées lors de la définition de la table.
- SET : Un type de données qui permet de stocker un ensemble de valeurs possibles pour une colonne donnée. Les valeurs doivent être spécifiées lors de la définition de la table.

Pour insérer des données dans une table MySQL, vous pouvez utiliser la commande SQL INSERT INTO. La structure générale de la commande INSERT INTO est la suivante :

```
INSERT INTO nom_de_la_table (colonne1,  
colonne2, colonne3, ...) VALUES (valeur1,  
valeur2, valeur3, ...)
```

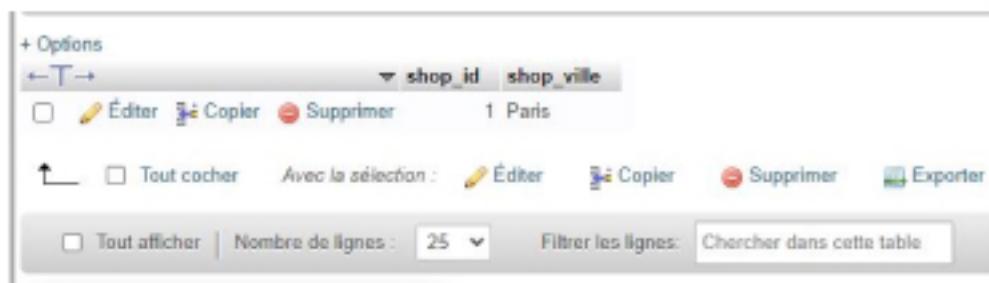
- nom\_de\_la\_table : le nom de la table dans laquelle vous voulez insérer des données.
- colonne1, colonne2, colonne3, ... : les noms des colonnes auxquelles vous voulez affecter des valeurs.
- valeur1, valeur2, valeur3, ... : les valeurs que vous souhaitez insérer dans chaque colonne correspondante.

Exemple :

Supposons que nous avons un a la table "shops" avec les colonnes, nous pourrions utiliser la commande SQL suivante :

```
INSERT INTO shops (shop_ville) VALUES  
( 'Paris' );
```

Ceci ajoutera au niveau de votre table shops de l'interface de phpMyAdmin



shop_id	shop_ville
1	Paris

Pour modifier des données dans MySQL, nous utilisons l'instruction UPDATE. La syntaxe générale de l'instruction UPDATE est la suivante :

```
UPDATE table  
SET colonne1 = valeur1,  
colonne2 = valeur2, ...  
WHERE condition(s);
```

Voici un exemple de modification de données dans une table appelée shops avec des colonnes shop et employes :

```
UPDATE employes SET emp_nom = 'Clement'  
WHERE emp_id = 2;
```

Dans cet exemple, nous modifions la colonne emp\_nom de la ligne correspondant à la table employes pour lui donner une nouvelle valeur 'Clement'. La condition WHERE est utilisée pour s'assurer que la modification est effectuée sur la bonne ligne. Du coup notre employé 'Marie' deviendra 'Clement'.

Il est important de noter que si la condition WHERE n'est pas utilisée, toutes les lignes de la table seront modifiées. Il est donc essentiel de veiller à ce que la condition soit correctement spécifiée afin de ne pas causer de dommages accidentels à la table.

Avec MySQL, la clause AUTO\_INCREMENT est utilisée pour spécifier qu'une colonne de type numérique, généralement INT, doit être incrémentée automatiquement lorsqu'une nouvelle ligne est insérée dans la table.

Par exemple, si vous avez une table "clients" avec une colonne "id\_client" définie comme AUTO\_INCREMENT, la première ligne insérée dans la table aura une valeur "id\_client" de 1, la deuxième ligne aura une valeur de 2, et ainsi de suite.

La syntaxe pour ajouter la clause AUTO\_INCREMENT lors de la création d'une table est la suivante :

```
CREATE TABLE nom_de_la_table (
    nom_de_la_colonne INT AUTO_INCREMENT PRIMARY KEY,
    ...
);
```

Il est important de noter que la colonne définie comme AUTO\_INCREMENT doit également être définie comme clé primaire de la table en utilisant la clause PRIMARY KEY. Cela garantit que chaque valeur de la colonne "id\_client" est unique et ne peut pas être NULL.

Vous pouvez également ajouter la clause AUTO\_INCREMENT à une colonne existante dans une table en utilisant la commande ALTER TABLE. Voici un exemple :

```
ALTER TABLE nom_de_la_table MODIFY
nom_de_la_colonne INT AUTO_INCREMENT PRIMARY KEY;
```

Cette commande ajoute la clause AUTO\_INCREMENT à la colonne "nom\_de\_la\_colonne" et la définit comme clé primaire.

Pour insérer plusieurs données:

```
INSERT INTO employes (emp_nom,  
emp_prenom, shop_id)  
VALUES ('Marie', 'Dublanc', 1), ('Tommy',  
'Carl', 2), ('John', 'Snow', 1),  
('Peter', 'Harley', 2);
```

Pour supprimer un élément d'une table en SQL, nous utilisons l'instruction DELETE. Voici la syntaxe générale :

```
DELETE FROM table  
WHERE condition;
```

Attention cette commande est irréversible.

Voici un exemple concret suivant notre derniers exemple et nous allons effacer 'Tommy':

```
DELETE FROM employes WHERE emp_id = 3;
```

Si nous mettons aucune condition comme ceci

```
DELETE FROM employes ;
```

cela va effacer la table employes

MySQL est un système de gestion de bases de données relationnelles open-source. Il est très populaire et largement utilisé dans le monde de l'informatique. MySQL permet de stocker et de gérer des données de manière efficace, sécurisée et fiable.

MySQL utilise le langage SQL (Structured Query Language) pour effectuer des opérations de lecture, d'écriture, de mise à jour et de suppression de données dans une base de données. SQL est un langage de programmation standard pour les bases de données relationnelles, ce qui rend MySQL compatible avec de nombreux autres systèmes de bases de données.

MySQL est utilisé dans de nombreuses applications, telles que les sites Web, les applications de commerce électronique, les systèmes de gestion de contenu, les applications mobiles et de bureau, et bien d'autres encore. Il est particulièrement apprécié pour sa rapidité, sa fiabilité et sa facilité d'utilisation.

MySQL est également hautement personnalisable et extensible grâce à une large gamme de plugins et d'outils tiers. Il est compatible avec les principaux systèmes d'exploitation, notamment Windows, Linux et macOS, ce qui le rend facilement accessible pour les développeurs de logiciels dans le monde entier.

En résumé, MySQL est une solution de gestion de bases de données fiable et performante, qui est utilisée dans de nombreux domaines de l'informatique. Grâce à sa compatibilité, sa personnalisation et son extensibilité, MySQL est un choix populaire pour les développeurs de logiciels et les entreprises.



# laminutedecode.io

@LAMINUTEDECODE

