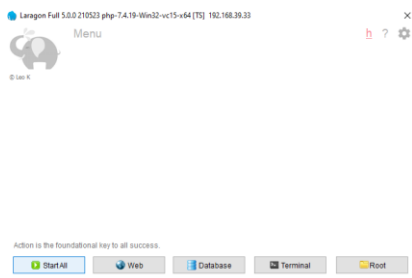


Logiciel de base de données

Nous allons utiliser MySQL. Il fait partie des principaux logiciels de base de données du marché. Cf par exemple <https://survey.stackoverflow.co/2023/#section-admired-and-desired-databases> ¹

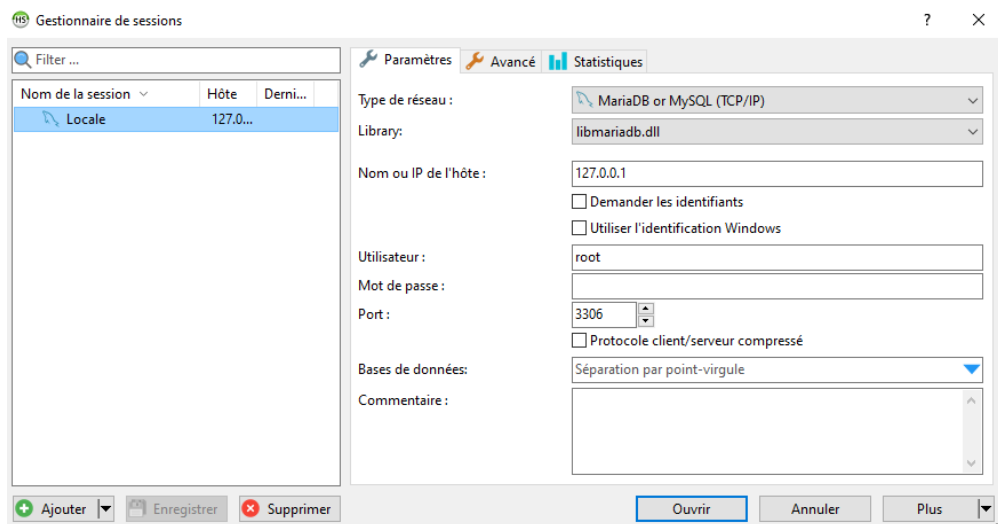
Lancement du logiciel de base de données

1. Démarrer le logiciel Laragon



2. Cliquer sur Database

Ajouter une session. Normalement les paramètres par défaut sont Ok.



Travail à faire

Démarrer le serveur de base de données et créer une nouvelle session, à laquelle vous donnerez votre acronyme, par exemple *TRobert*

¹ Attention, il s'agit ici d'une enquête d'usage, les résultats sont très différents de la répartition par parts de marchés en valeur, car des bases de données comme MySQL ou PostgreSQL sont essentiellement gratuites.

Terminologie

1. Interface ou commande

En utilisant un logiciel de base de données, vous avez toujours le choix entre 3 possibilités :

- a. Utiliser les commandes dans une fenêtre dédiée à l'exécution du code. Cette maîtrise est obligatoire pour les évaluations car elle implique de retenir les instructions de base du langage SQL étudié
- b. Utiliser les menus et fenêtres de l'interface. C'est une possibilité pour découvrir et apprendre au démarrage, mais elle ne doit pas se faire au détriment de la maîtrise des instructions du langage
- c. Exécuter un script déjà préparé et sauvegardé sous forme de fichier texte avec une extension .sql (au lieu de .txt). Cette méthode est adaptée à des logiciels et programmes en production ou alors pour charger des données que je vous prépare

Dans la suite du TP, je vous indique à chaque fois les possibilités a et b.

2. Majuscule / minuscule

Le SQL n'est pas Case Sensitive, c'est-à-dire que les *instructions* SQL ne sont pas sensibles à la casse autrement dit peu important qu'elles soient écrites en minuscule ou majuscule.

Lorsque j'indique une commande en MAJUSCULE c'est qu'elle est obligatoire (mais elle marchera aussi bien écrite en minuscule). Quand j'indique en italique c'est qu'il s'agit d'un *paramètre* c'est-à-dire quelque chose d'obligatoire mais au choix de l'utilisateur.

Création d'une base de données

Une base de données est le réceptacle de l'ensemble des données nécessaires soit :

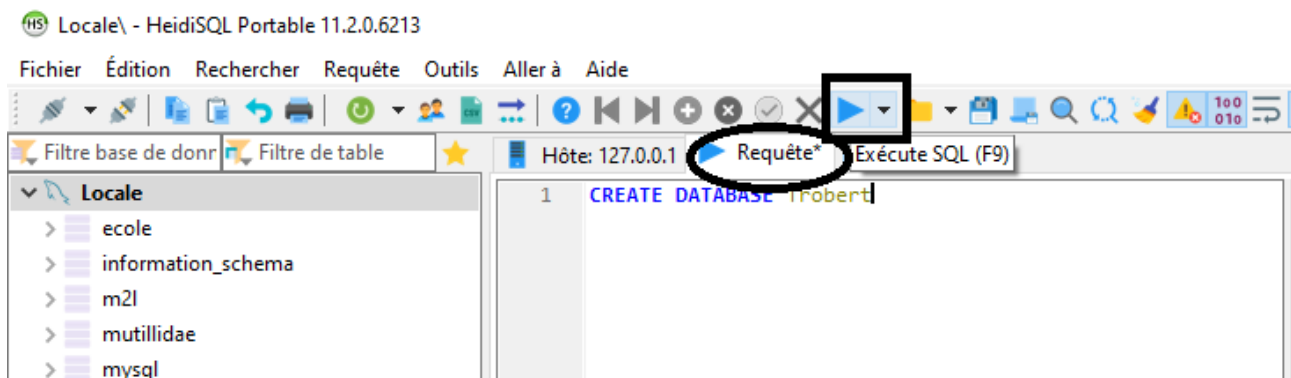
- à un projet de développement informatique,
- un progiciel de gestion comme SAP
- un logiciel tel que Pronote

1. Avec des instructions SQL

L'instruction de base pour créer une base de données est :

CREATE DATABASE *nom_de_la_base*

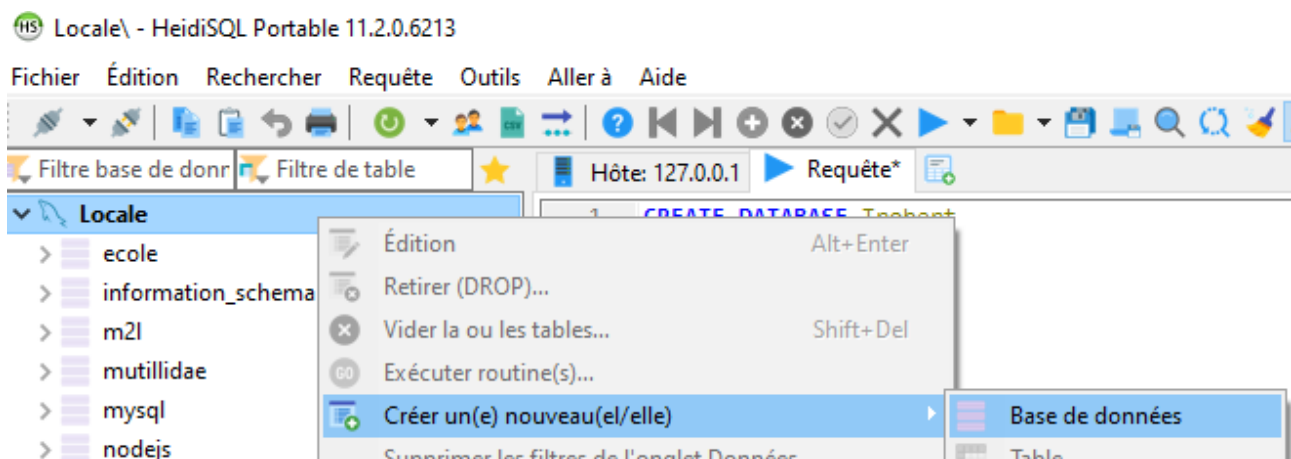
Ici *nom_de_la_base* est obligatoire mais le rédacteur du programme mets « ce qu'il veut » (le nom de sa grand-mère, quelque chose de censé ou il respecte une convention de nommage utilisée par son entreprise, son responsable etc).



Les instructions sont saisies dans la partie Requête (Ovale) et exécutées via le triangle (Rectangle).

2. Avec l'interface

Clic droit sur la session à gauche puis → Crée une nouveau → Base de données



Travail à faire

Créer une nouvelle base de données qui a pour nom votre acronyme, par exemple *TRobert*

Suppression d'une base de données existante

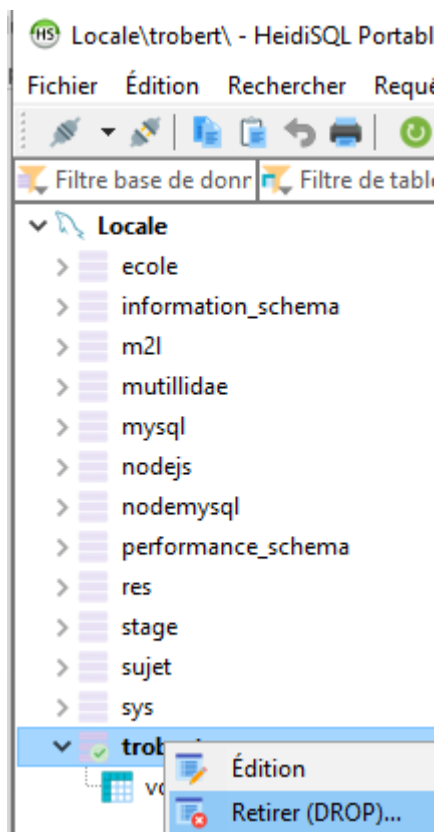
Attention : la suppression d'une base de données supprime tous les objets, c'est-à-dire tables et vues, contenus dans la base de données.

1. Via une instruction SQL

`DROP DATABASE nom_de_la_base`

2. Via l'interface

Clic droit sur le base de données qu'on souhaite supprimer → Retirer (DROP)



Création de table – Structure – Détail du CREATE TABLE

Introduction

L'instruction CREATE TABLE est la commande SQL principale pour créer des structures de table.

L'ajout de données se fait ensuite via des instructions INSERT pour insérer des données dans ces tables.

Nous allons détailler la structure des tables et les instructions permettant de définir cette structure.

Les principaux types MySQL

Dans MySQL, les principaux types sont :

- Numérique
- Texte et Chaîne caractère
- Date et Heure

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

A faire

Cherchez les différents types de données numériques sous MySQL et répartissez les entre les nombres entiers et les nombres décimaux.

Nombres entier
Nombres décimaux

A faire

Recherchez et indiquez quelle est la différence entre le type CHAR et le type VARCHAR ?

A faire

Cherchez et expliquez la différence entre les types DATE, DATETIME et TIME

Création d'une table

Introduction

La table est la structure logique et physique de base qui va recevoir et stocker des données utilisées par une application sur un poste, client-serveur, un logiciel ou une application Web.

Rappel du cours : on utilise plusieurs tables pour optimiser les performances, le stockage, la mémoire et la qualité de données tout en répondant à des impératifs de sécurité (cf cybersécurité).

Dans une table Excel, on accède à une cellule comme croisement d'une colonne (lettre) et d'une ligne (numéro).

Dans une table de base de données, on définit la structure de la table, c'est-à-dire les **colonnes** (ou champs ou attributs) en leur donnant des noms qu'on choisit et en précisant leur **type** et longueur. Ensuite on insère des données dans la table en respectant cette structure.

Enfin on récupère des données en sélectionnant des colonnes et filtrant les lignes.

1. Avec des instructions SQL

La syntaxe complète d'une instruction CREATE TABLE en MySQL est disponible sur le site suivant: <https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

Remarque :

Les crochets [] désignent une instruction optionnelle.

Les pipe (|) désignent des options alternatives.

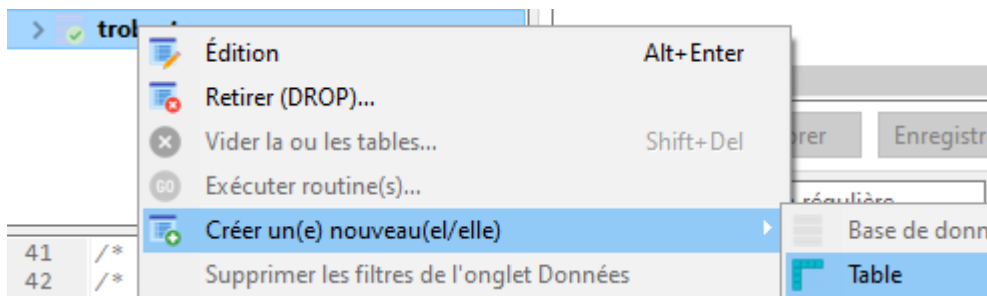
Exemple :

Ci-joint des instructions pour créer une table VOITURE

```
CREATE TABLE VOITURE
(
  ID_VOITURE CHAR(8) ,
  MARQUE VARCHAR(30),
  MODELE VARCHAR(200),
  ENERGIE CHAR(7),
  PUISSANCE INT,
  DATE_MISE_EN_SERVICE DATE
);
```

2. Au moyen de l'interface

Clic droit sur la base de données → Créer une nouvelle → Table



Ensuite : on donne un nom à la table. Par exemple ici Voiture.

De base Options Index (0) Clés étrangères (0) ✓

Nom : Saisir le nom de la table

Commentaire :

Colonnes : + Ajouter x Supprimer ▲ Haut ▼ Bas

#	Nom	Type de données	Taille/Ensem...	No

Ensuite on clique sur le bouton + Ajouter pour ajouter une par une les colonnes (champ de la table).

Colonnes : + Ajouter x Supprimer ▲ Haut ▼ Bas

#	Nom	Type de données	Taille/Ensem...	No
1	ID_VOITURE	CHAR	8	

Travail à faire

Créer une table nommée NOTES_ETUDIANT avec les champs suivant : nom et prénom de l'étudiant, section, date de naissance, libellé matière, type évaluation, date évaluation, coefficient et note

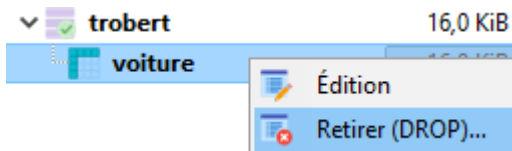
Suppression d'une table existante

1. Au moyen d'une instruction SQL

`DROP TABLE nom_table`

2. Via l'interface

Clic droit sur la table que l'on souhaite supprimer totalement.



Insertion de données (dans une table)

Une insertion de données ne peut se faire que dans une table préexistante (qui doit donc déjà exister au moment où on souhaite ajouter des données dedans).

L'instruction `CREATE TABLE` définit la structure, la liste et le type des champs (colonnes) d'une table.

L'insertion de données permet de peupler la table, c'est-à-dire d'ajouter des lignes.

1. Ajout via l'instruction SQL `INSERT`

La syntaxe de l'instruction `INSERT` est la suivante :

`INSERT INTO nom_table (col1, col2, ... coln) VALUES(...);`

Contraintes:

- Derrière le nom de la table, entre parenthèses, on doit mettre les noms de colonne que l'on souhaite alimenter
- Ces noms de colonne doivent correspondre à des noms de colonne définis lors de la création de la table
- L'ordre des valeurs dans l'instruction `VALUES` doit correspondre à l'ordre des colonnes défini entre parenthèses après le nom de la table
- Le type des valeurs dans l'instruction `VALUES` doit correspondre au type de la colonne tel qu'il a été défini dans l'instruction `CREATE TABLE`

Exemple : pour insérer une donnée cohérente avec notre table `voiture` précédemment définie


```
INSERT INTO voiture(ID_VOITURE, DATE_MISE_EN_SERVICE, MODELE, MARQUE, PUISSANCE, ENERGIE) VALUES('CITC5AIR','2020-05-12','C5 AIRCROSS','Citroen',225,'Hybride') ;
```

Table après l'insertion de la donnée :

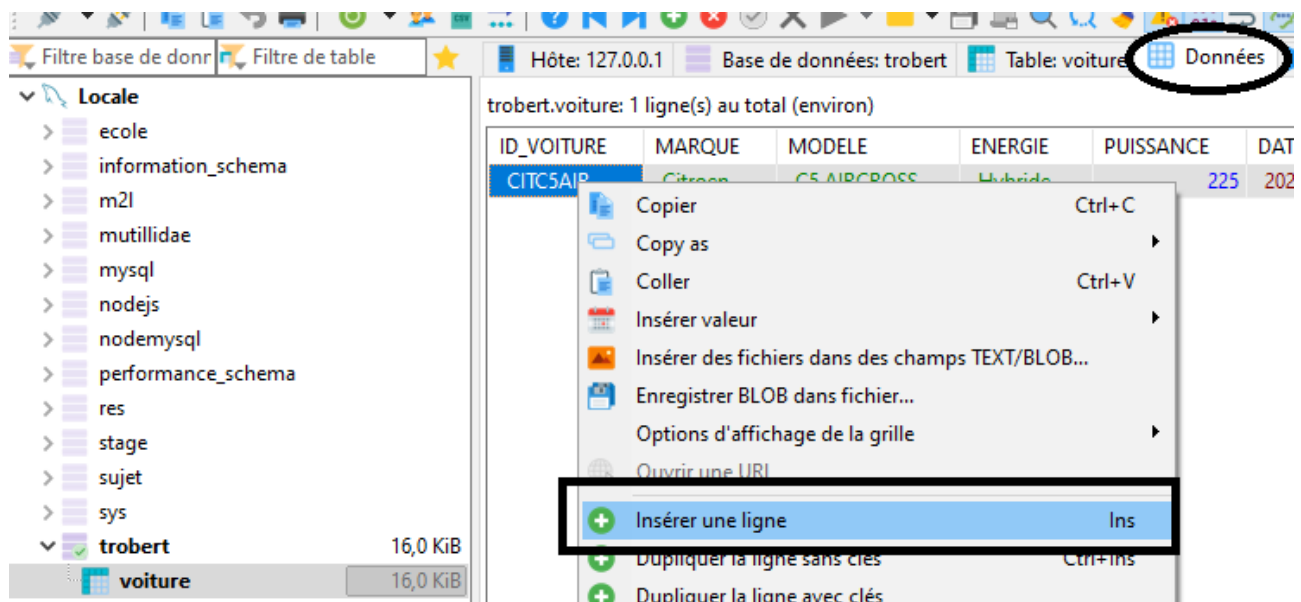
trobert.voiture: 1 ligne(s) au total (environ)

ID_VOITURE	MARQUE	MODELE	ENERGIE	PUISSANCE	DATE_MISE_EN_SERVICE
CITC5AIR	Citroen	C5 AIRCROSS	Hybride	225	2020-05-12

2. Insertion d'une donnée via l'interface

Tout d'abord on se positionne sur l'onglet donnée (losange) de la table dans laquelle on souhaite ajouter des données.

Ensuite on fait un clic droit → Insérer une ligne



Il reste ensuite à saisir les valeurs.

Travail à faire

Ajouter vous 2 notes dans la table NOTES_ETUDIANT

Suppression de données (dans une table)

La suppression de données dans une table enlève une ou plusieurs lignes sans supprimer la table et sa structure.

1. Via une instruction SQL

`DELETE FROM nom_table`

Cette instruction supprime **toutes** les lignes de la table indiquée.

Nous verrons plus tard comment supprimer une ou plusieurs lignes uniquement sans supprimer toutes les lignes

2. Via l'interface

On se positionne sur l'onglet Données de la table dont on souhaite supprimer des lignes puis on clique droit sur la ligne qu'on veut supprimer puis → Supprimer la ligne sélectionnée

