

# Bases de données

## I/ Présentation générale

### 1. Objectifs

Une *base de données* est un ensemble structuré de données :

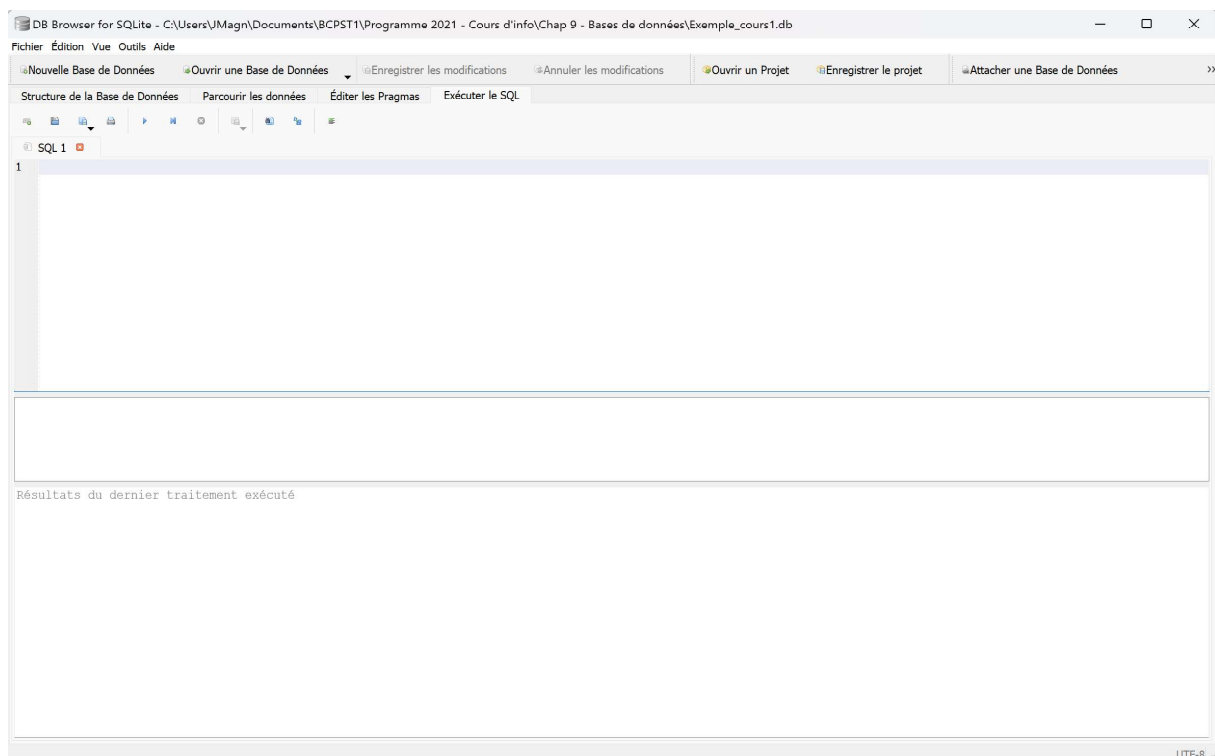
- stockée sur un support accessible à une ou plusieurs machines
- modélisant des informations réelles
- pouvant être interrogée (par des *requêtes*)
- pouvant être mise à jour par une "communauté" d'utilisateurs

Typiquement, une base de données est un bon outil pour gérer un fichier client dans un magasin, pour gérer des commandes à passer auprès de différents fournisseurs dans une grande entreprise, ou même pour analyser les résultats d'une expérience scientifique où beaucoup de données sont collectées.

### 2. Installer SQLite

Si vous disposez d'un ordinateur personnel, il faudra y installer un logiciel permettant de manipuler des bases de données. Le logiciel recommandé est SQLite. Vous pouvez le télécharger à l'adresse : <https://sqlitebrowser.org/>.

En TP nous utiliserons sa version portable, qui ne nécessite pas d'installation.



## II/ Vocabulaire des bases de données

### 1. Structure d'une base de données

Dans une agence de location de voitures, on souhaite assurer le suivi des locations, notamment pour pouvoir facturer les clients et assurer la maintenance des véhicules. On choisit d'organiser ces informations dans une **base de données**.

Cette base de données est constituée de trois **tables** ou **relations** :

- La table "clients" qui contient les informations relatives aux clients : numéro de client, nom et prénom
- La table "voitures" qui contient les informations relatives aux véhicules : modèle, immatriculation et compteur
- La table "locations" qui contient les informations relatives aux locations : date de début, date de fin, distance parcourue, montant, client et voiture

Ainsi, pour cette agence de location, on a :

clients		
numero	nom	prenom
1	Chatenet	Benjamin
2	Chevalier	Caroline
3	Haïdar	Kamal
4	Joguet	Jennifer
5	Magnoux	Julie
6	Metz	Florence
7	Dugat	Aurianne

voitures		
modele	immatriculation	compteur
Peugeot 2008	AA123AA	65780
Renault Captur	CD456CD	100755
Renault Clio	BA326XW	40129
Fiat 500	FG989IP	12090
Citroën DS3	HJ098BB	90854

locations					
date_debut	date_fin	distance	montant	clients	voitures
20241223	20241227	80	90	Magnoux	BA326XW
20220219	20220226	320	575	Chatenet	HJ098BB
20241228	20250102	1070	680	Chevalier	FG989IP
20220226	20220305	150	450	Metz	BA326XW
20220304	20220306	240	120	Joguet	CD456CD
20220304	20220306	370	200	Magnoux	AA123AA

Détaillons la structure de la table "clients" :

- Elle est constituée de trois **attributs** ; "numero", "nom" et "prenom". Les attributs sont les titres des colonnes.
- Chaque attribut a un **domaine**, qui correspond à la notion de type en Python. Ainsi, l'attribut "numero" a pour domaine les entiers, l'attribut "nom" a pour domaine les chaînes de caractères et l'attribut "prenom" a pour domaine les chaînes de caractères. Un attribut peut aussi avoir pour domaine un nombre réel.
- Le **schéma de la table** est l'ensemble de ses attributs avec leurs domaines.  
Ainsi, le schéma de la table est : clients(numero : int, nom : string, prenom : string).
- La table est constituée de six **lignes** ou **enregistrements**, qui donnent la liste de six clients de l'agence.

Le schéma de la table "voitures" est :

.....

.....

Le schéma de la table "locations" est :

.....

.....

## 2. Visualisation dans SQLite

Dans SQLite, l'onglet "Structure de la Base de Données" permet de voir la liste des tables et le schéma de chaque table :



## 3. Clés

Pour chaque table, on peut également définir une **clé**, c'est-à-dire un sous-ensemble de ses attributs permettant d'identifier de manière unique les enregistrements. Nous nous limiterons au cas où la clé est réduite à un seul attribut ; on parle dans ce cas de **clé primaire**.

Pour la table "clients" :

- L'attribut "numero" peut constituer une clé ; chaque client a un seul numéro de client et réciproquement.
- L'attribut "nom" ne peut pas constituer une clé ; il peut exister des homonymes.

Pour la table "voitures" : .....

Pour la table "locations" : .....

Une **clé étrangère** dans une table T est un attribut de T qui est une clé primaire dans une autre table T'.

Ainsi, dans la table "locations", l'attribut "voitures" est une clé étrangère, puisque cet attribut permet de désigner de façon unique les enregistrements de la table "voitures".

### III/ Requêtes SQL

Le *langage SQL* (pour Structured Query Language) est un langage permettant de manipuler des bases de données. Les commandes sont appelées des *requêtes*. En SQL, une requête se termine par un point-virgule.

#### 1. Sélection

Pour accéder aux données, on utilise la requête **SELECT**. La syntaxe est la suivante :

```
SELECT attribut FROM table ;
```

On peut également ajouter des conditions avec la syntaxe suivante :

```
SELECT attribut FROM table WHERE conditions ;
```

Pour exprimer des conditions, on peut utiliser les opérateurs :

+   -   \*   /   =   <>   <   <=   >   >=   **AND**   **OR**   **NOT**   **IN**

Ainsi, toujours dans l'exemple de la base de données de l'agence de location de voitures :

- Pour connaître uniquement les noms des clients de l'agence :

```
SELECT nom FROM clients ;
```

- Pour connaître les immatriculations des voitures ayant au moins 60 000 km au compteur :

```
SELECT .....  
.....
```

- Pour connaître .....

```
SELECT voitures FROM locations WHERE date_fin < 20220228 ;
```

De façon optionnelle, on peut écrire une requête dans laquelle on élimine les redondances. Par exemple, la requête :

```
SELECT DISTINCT clients FROM locations ;
```

sert à .....

De façon optionnelle, on peut écrire une requête dans laquelle on effectue un tri selon un attribut. Par exemple, la requête :

```
SELECT modele FROM voitures ORDER BY compteur ;
```

sert à .....

## 2. Projection

Par projection, on peut accéder à plusieurs attributs, toujours éventuellement avec des conditions. La syntaxe est la même que précédemment, en séparant les attributs par des virgules.

```
SELECT A1 , A2 , ... , An FROM table WHERE conditions ;
```

Ainsi, toujours dans l'exemple de la base de données de l'agence de location de voitures :

- Pour connaître les noms et prénoms des clients de l'agence :

```
SELECT .....
```

- Pour connaître .....

```
SELECT date_debut , date_fin FROM locations WHERE voitures = 'BA326XW' ;
```

Les mots-clés DISTINCT et ORDER BY peuvent également être utilisés.