

UE Informatique

Conception et Exploitation de Bases de Données

Mini Manuel SQLite/Studio

SQLite



- ❑ **SGBD classique** : un SGBD classique s'exécute généralement sur un serveur dédié (ordinateur accessible sur un réseau) auquel se connecteront des clients afin de lui transmettre des requêtes
- ❑ **SQLite** (<http://www.sqlite.org>) est un SGBD embarqué : ie à la fois client et serveur.
- ❑ **Il peut être enveloppé dans une application**
 - ❑ Ex. firefox, thunderbird, antivirus McAfee, Skype...
- ❑ **Inclus dans certains OS (Android notamment)**
- ❑ **C'est un SGBD SQL** qui gère plusieurs tables, les vues, les transactions... La base de données est stockée dans un seul fichier.
- ❑ **La version de SQL** supportée par SQLite est une version simplifiée, largement suffisante pour ce qu'on va faire.



1. Installation / Configuration

Téléchargement SQLite – www.sqlite.org

■ Les logiciels retenus sont dits

- « **no install** » / « **portable** » : signifiant qu'il ne nécessite pas d'installation / droit d'administration car l'installation se limite à une décompression / dézippage.
- **Open Source** (gratuits)

■ Version ligne de commande : **SQLite** (très léger < 1Mo)

Pour les *hard*
codeurs only

- **Téléchargement** : <https://www.sqlite.org/download.html>
- Il faut télécharger les « Precompiled Binaries » correspondant à votre Système (Windows 32/64, MacOS, Linux)

■ Version « **Studio** » : SQLite + environnement graphique



**SQLite
Studio**

- <https://sqlitestudio.pl/>

- La version zip ne nécessite qu'un décompresseur/dezippeur, en général préinstallé avec l'OS. Contient déjà le programme SQLite

Utilisé en TD !

Installation ...

■ Again ... no install ...

1. Télécharger fichier compressé (extension .zip en général)
2. Décompresser le fichier quelque part ☺ (créer un répertoire dans Mes Documents ... whatever ...)
(cf comment décompresser un fichier : <https://www.youtube.com/watch?v=9MMghTwISr4>)
3. Aller dans ce répertoire (explorateur de fichier, finder, whatever)
4. (Double) cliquer sur le programme

SQLiteStudio.exe				
	Qt5Widgets.dll	12/04/2021 00:08	Extension de l'app...	8 460 Ko
	Qt5Xml.dll	12/04/2021 00:08	Extension de l'app...	375 Ko
	sqlite3.dll	12/04/2021 00:08	Extension de l'app...	977 Ko
	SQLiteStudio.exe	12/04/2021 00:08	Application	371 Ko
	sqlitestudiocli.exe	12/04/2021 00:08	Application	321 Ko

Celui là ! (l'extension .exe peut être masquée)



IMPORTANT

Ne pas ouvrir le zip puis exécuter le programme. Cela créera un répertoire provisoire en lecture seule ce qui peut bloquer et faire planter le programme ...

Bref, **don't do it.**

=> 1. Décompresser fichier, 2. aller dans le répertoire créé et 3. exécuter le programme



2. Démarrage, prise en main : création BD, Tables & Relations

Avec SQLite Studio

Histoire de comprendre ce qu'on fait

...

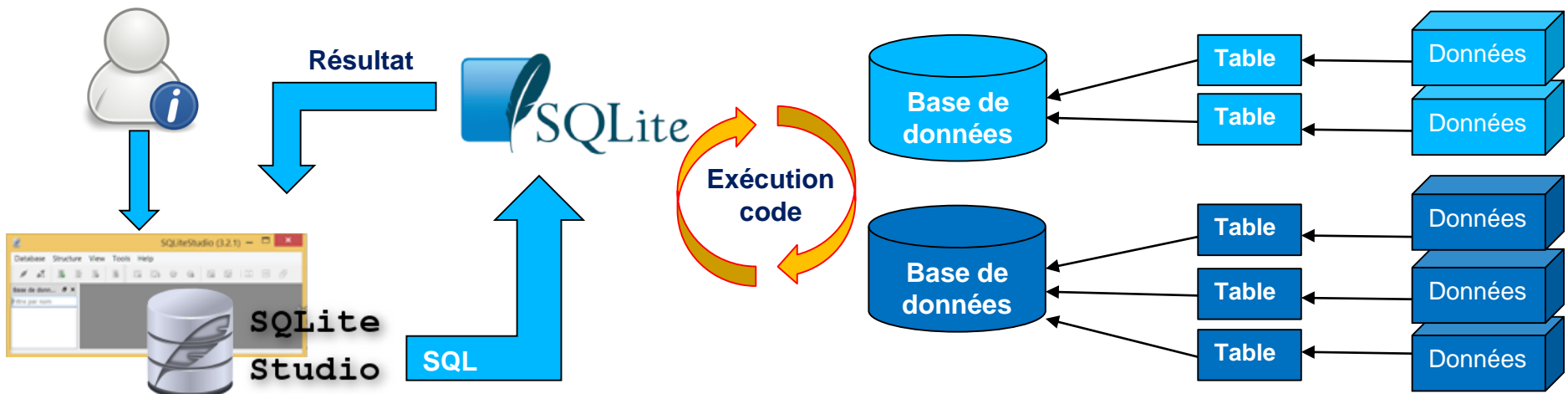
❑ SQLite Studio c'est :

- ❑ Un SGBD (SQLite) capable de comprendre des instructions (requêtes) SQL ...
- ❑ Un « front end », une interface graphique qui permet de dialoguer avec SQLite via des interfaces graphiques ...

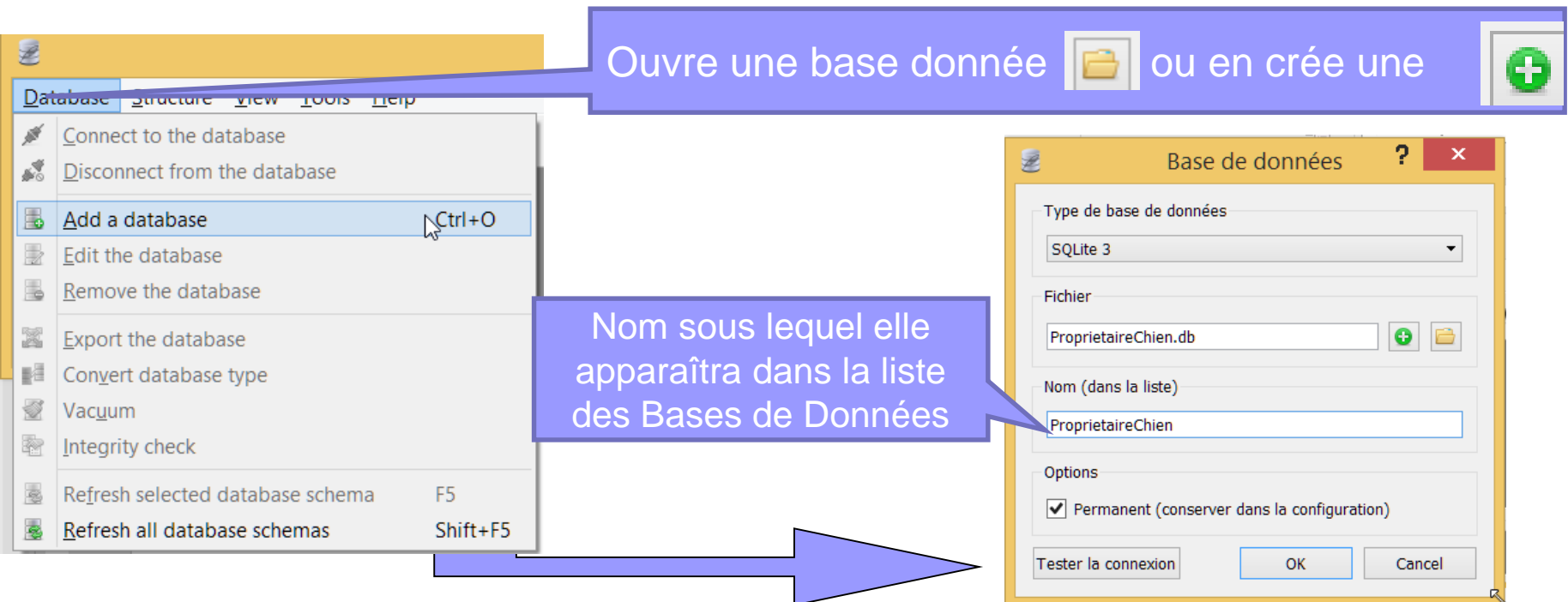
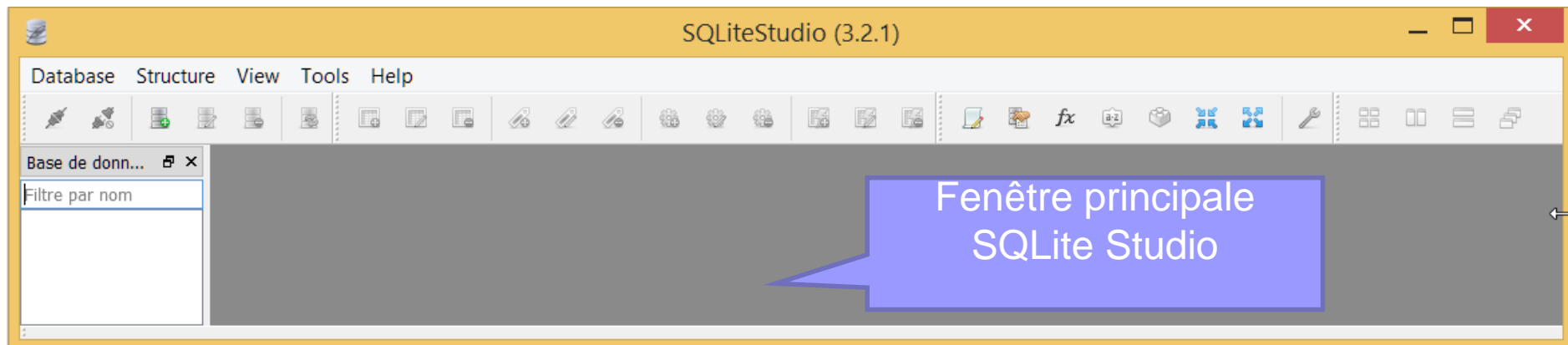
❑ Bref, le côté Studio masque le travail du SGBD...

❑ ... pas très grave s'agissant d'une initiation à la conception et l'exploitation de BD ... mais à garder en tête...

❑ En résumé :



Ouverture SQLiteStudio



Création d'une table

The image shows a database management tool interface. At the top, a window titled 'Base de données' displays a tree view with 'PropriétaireChien (SQLite 3)' expanded, showing 'Tables' and 'Vues'. A context menu is open over the 'Tables' folder, with 'Create a table' selected. A large blue arrow points from this menu to the table creation interface below.

Base de données
Filtre par nom
PropriétaireChien (SQLite 3)
Tables
Vues

Context menu options:
Create a table
Base de données
Copier
Coller
Supprimer les objets sélectionnés
Tout sélectionner
Refresh all database schemas

Structure | Données | Contraintes | Index | Déclencheurs | DDL

Icons: Refresh, Checkmark, Close, Add Column, Add Index, Add Constraint, Add Trigger, Add View, Add Table, Undo

Nom de la table : Ajouter une colonne (Ins) ☐ WITHOUT ROWID

Annotations:

- Double – cliquer sur la BD ou Cliquer sur le triangle pour développer la vue
- Ajouter une colonne i.e propriété à la table
- Nom de la table

Création d'une table : ajout propriétés

Nom & type de données de la propriété

Colonne ? x

Nom et type

Nom de colonne : Type de données : Taille : ,

Contraintes

- ☒ Clef primaire
- ☐ Clef étrangère
- ☐ Unique
- ☐ Vérifier la condition
- ☐ Non NULL
- ☐ Collation
- ☐ Défaut

☐ Mode avancé

AutoIncrement : le SGBD gère directement les valeurs de clefs primaires en ajoutant +1 à chaque nouvel ajout d'enregistrement

Modifier contrainte ? x

Clef primaire

- ☒ Auto-incrémentation
- ☐ Ordre de tri :
- ☐ Contrainte nommée :
- ☐ En cas de conflit :

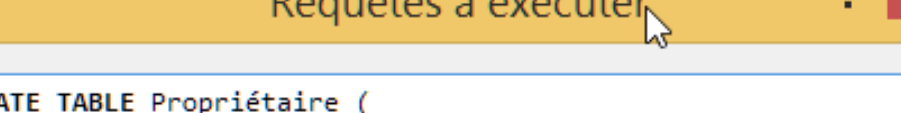
Création d'une table : fin



Cliquer sur sauvegarder provoque l'affichage de l'équivalent SQL de ce que l'on vient de faire (CREATE TABLE)

[illegible]

Chaque modification et sauvegarde fera de même (avec un ALTER TABLE, la commande SQL permettant de modifier la définition d'une table)



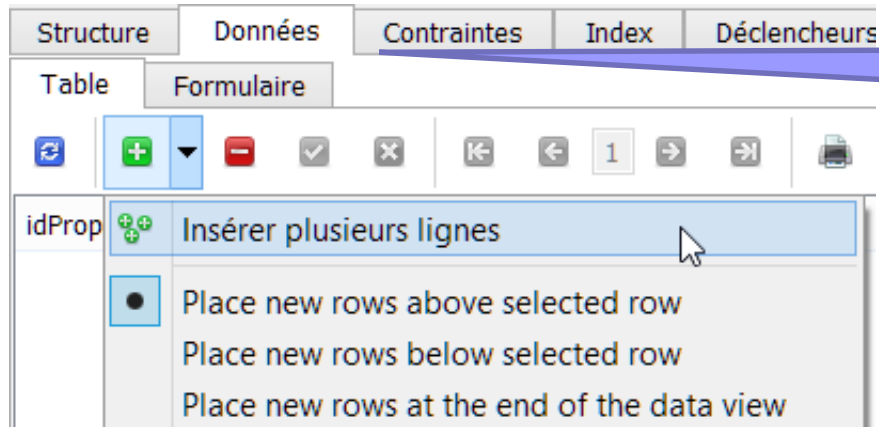
Requêtes à exécuter

```
CREATE TABLE Propriétaire (  
    idProp      INTEGER      PRIMARY KEY AUTOINCREMENT,  
    nomProp     STRING (20)  NOT NULL,  
    prenomProp  
);
```

☐ Ne plus afficher

OK Cancel

Populer (sic) une table



Onglet Données
Saisie d'un enregistrement ou +sieurs

A noter, valeur de clef
primaire laissée vide
(NULL)

1		LAMBERT	Gérard
2	NULL	DHI	Lydie
3	NULL	FALL	Skye
4	NULL	OUHALI	Kamel
5	NULL	FONFEC	Sophie



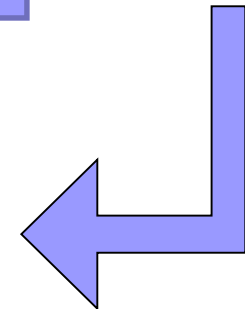
Valider (enregistrer) ou Ctrl+Entrée
sinon ajout non effectué (un
warning s'affiche)



Raffraichir la table
pour voir la table
mise à jour

Valeur de clef primaire
saisie automatiquement
Par SGBD

	idProp	nomProp	prenomProp
1	1	LAMBERT	Gérard
2	2	DHI	Lydie
3	3	FALL	Skye
4	4	OUHALI	Kamel
5	5	FONFEC	Sophie



Création de table : Type de données

SQL as understood by SQLite

■ Types de donnée réduits par simplification

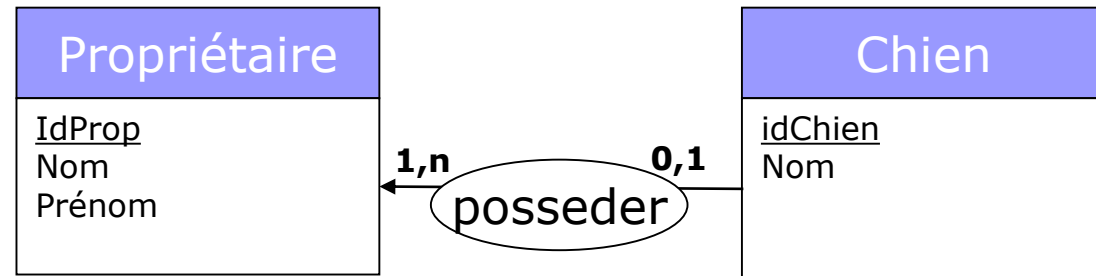
- **NULL** : pas de valeur (ex. NULL comme valeur de note signifie qu'il n'y a pas de note saisie, alors que 0 est une valeur ...)
- **INTEGER** : SQLite gère l'espace mémoire en fonction de la valeur (1,2.. 8 octets selon besoins 1o = 0..255 , 8=2^64)
- **REAL** : valeur réelle (stockée sur 8o)
- **TEXT** : ... no comment
- **BLOB** : forget about it

■ Le reste des types habituels de données est converti selon besoin :

- **Boolean** : stocké comme INTEGER avec valeur 0 ou 1
- **Heure** : stockée comme
 - un texte : "YYYY-MM-DD HH:MM:SS.SSS«
 - un nombre (REAL) de jours depuis le 24 novembre -4714 (calendrier Julien #histoire).
 - un entier (Unix Time) : nb de secondes depuis 1^{er} janvier 1970

Exemple de création de tables dans Base de LibreOffice

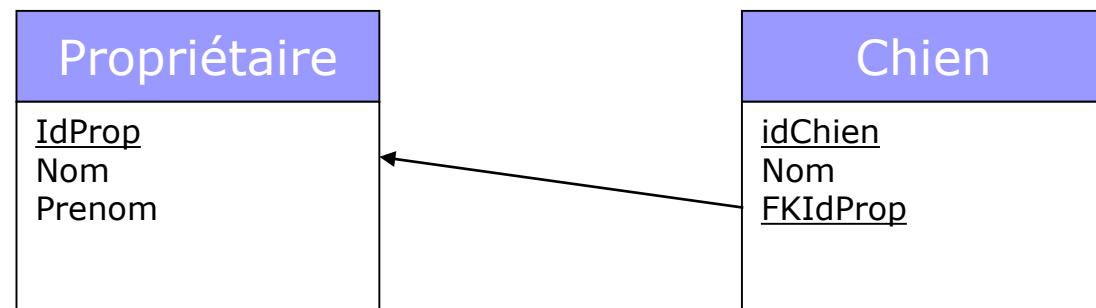
MCD



Un chien peut ne pas avoir de propriétaire

Rappel

MLD



Donc FKIdProp
pourra être NULL

Création d'une table avec clef étrangère

Colonne

Nom et type

Nom de colonne : FkIdProp Type de données : INTEGER Taille : ,

Contraintes

- ☐ Clef primaire
- ☒ Clef étrangère
- ☐ Unique
- ☐ Vérifier la condition
- ☐ Non NULL
- ☐ Collation
- ☐ Défaut

Mode avancé

OK Cancel

Modifier la contrainte

Clef étrangère

Table étrangère : Propriétaire

Colonne étrangère : idProp

Réactions

- ☐ ON UPDATE
- ☐ ON DELETE
- ☐ MATCH

Clef étrangère refusée

Contrainte nommée

Appliquer Cancel

	Nom	Type de données	Primary Key	Foreign Key	Unique	Contrôle	Not NULL
1	idChien	INTEGER					
2	nomChien	STRING (15)					
3	dateNaissance	DATE					
4	FkIdProp	INTEGER					

Vision globale des données saisies

Si on essaye d'ajouter l'identifiant d'un propriétaire **inexistant**, le SGBD (SQLite) refuse l'ajout et lève une erreur : violation de contrainte d'intégrité (clef étrangère)

5	Medor	01/01/2017	10
---	-------	------------	----

[22:49:53] An error occurred while committing the data: FOREIGN KEY constraint failed

Effectivement, pas de propriétaire dans la table Propriétaire ayant comme valeur de clef primaire « 10 »

« Contenu » de la
Table Proprietaire

idProp	nomProp	prenomProp
1	LAMBERT	Gérard
2	DHI	Lydie
3	FALL	Skye
4	OUHALI	Kamel
5	FONFEC	Sophie

10?

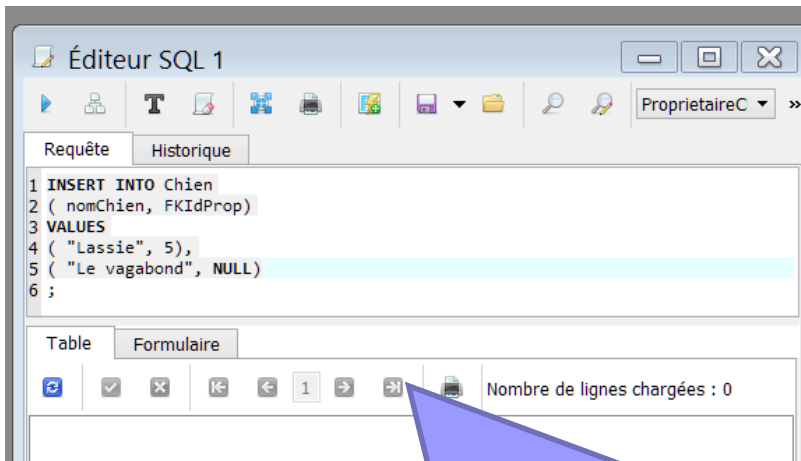
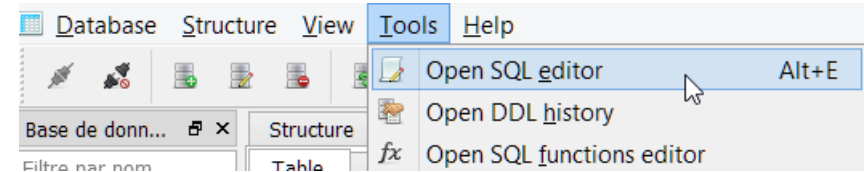
« Contenu » de la
Table Chien

idChien	nomChien	dateNaissance	FkIdProp
1	le chien	NULL	NULL
2	Zeus	01/01/1980	1
3	Appolon	01/01/1982	1
4	Laika	3/11/1954	2
5	Medor	01/01/2017	3

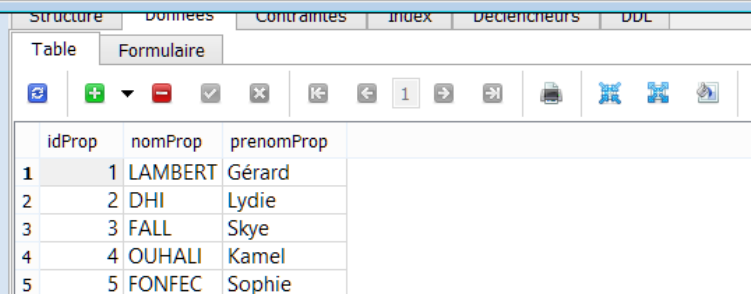
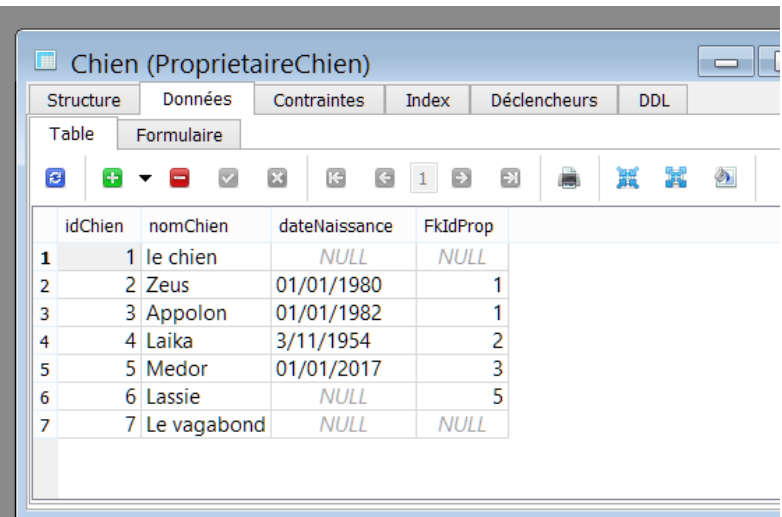
Mais 3 oui...

Saisie et « exécution » de code SQL

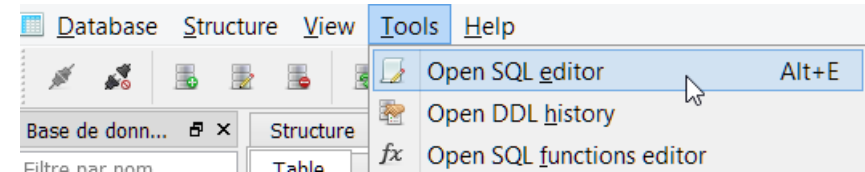
Toutes les opérations précédentes (création table, ajout de données) peuvent également être réalisées directement en SQL



Le multi-fenêtrage peut être utile pour se rappeler de la structure d'une table ou des données disponibles.
Par exemple orthographe exacte (minuscule/majuscule) du nom d'une colonne / table etc.



Saisie et « exécution » de code SQL



Éditeur SQL 1

Requête Historique

```

1 INSERT INTO Chien
2 ( nomChien, FkIdProp)
3 VALUES
4 ( "Lassie", 5),
5 ( "Le vagabond", NULL)
6 ;

```

Table Formulaire

Nombre de lignes chargées

Chien (ProprietaireChien)

Structure Données Contraintes Index Déclencheurs DDL

Table Formulaire

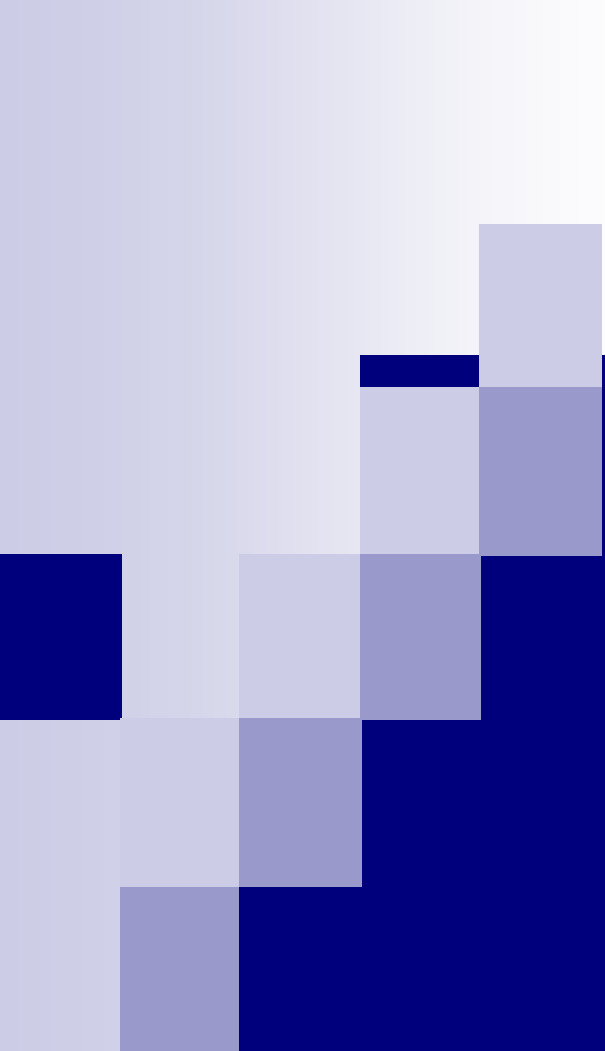
	idChien	nomChien	dateNaissance	FkIdProp
1	1	le chien	NULL	NULL
2	2	Zeus	01/01/1980	1
3	3	Appolon	01/01/1982	1
4	4	Laika	3/11/1954	2
5	5	Medor	01/01/2017	3
6	6	Lassie	NULL	5
7	7	Le vagabond	NULL	NULL

Menu déroulant sélectionnant la base de données courante... Si vous en avez plusieurs bien sélectionner la BD qui correspond à la requête en cours d'édition...

Pour lancer la requête cliquer sur



Menu déroulant sélectionnant la base de données courante... Si vous en avez plusieurs bien sélectionner la BD qui correspond à la requête en cours d'édition...



4. Requêtes SQL (SELECT)

Création d'une requête mono table

The screenshot shows an SQL Editor window titled "Éditeur SQL 1". The query editor contains the following SQL query:

```
1 SELECT *  
2 FROM Chien  
3 ;
```

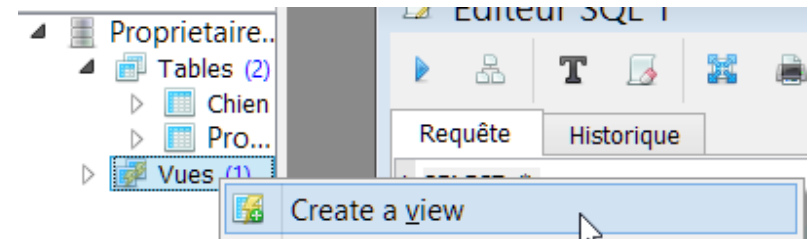
The query is executed, and the results are displayed in a table view. The table has four columns: **idChien**, **nomChien**, **dateNaissance**, and **FkIdProp**. The results are as follows:

	idChien	nomChien	dateNaissance	FkIdProp
1	1	le chien	NULL	NULL
2	2	Zeus	01/01/1980	1
3	3	Appolon	01/01/1982	1
4	4	Laika	3/11/1954	2
5	5	Medor	01/01/2017	3
6	6	Lassie	NULL	5
7	7	Le vagabond	NULL	NULL

The interface also includes a toolbar with various icons for query execution, formatting, and navigation. The "Table" tab is selected, and the "Formulaire" tab is also visible. The status bar indicates "Nombre de lignes chargées : 7".

Création d'une vue

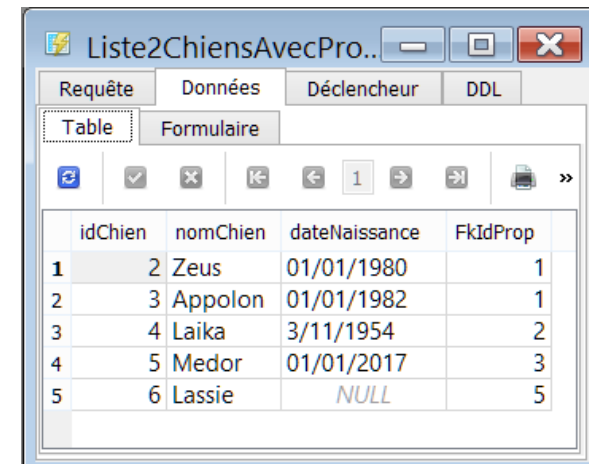
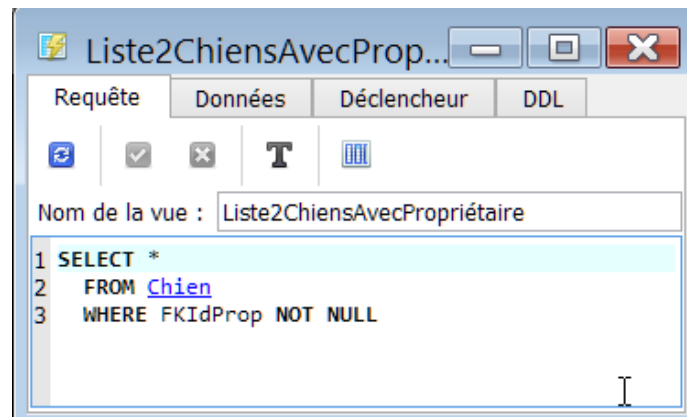
- ❑ Une vue permet de créer une table virtuelle (en lecture seule) correspondant à une requête **Select** (~sorte de sauvegarde réutilisable de requête)



- ❑ **Intérêt** : simplifie l'imbrication de requête, masque les informations sur la structure des tables impliquées dans la requête
- ❑ Vous pouvez ainsi **sauvegarder vos requêtes** (ex R12 pour la requête correspondant à la question 12). **SQLite Studio sauvegarde tout dans 1 seul fichier (la BD).**



ProprietaireChien.db



Jointure Interne ...

The screenshot shows an SQL Editor window titled "Éditeur SQL 1". The query editor contains the following SQL code:

```
1 SELECT nomChien, nomProp, prenomProp
2 FROM Chien INNER JOIN Proprietaire ON FkIdProp = idProp
3 ;
```

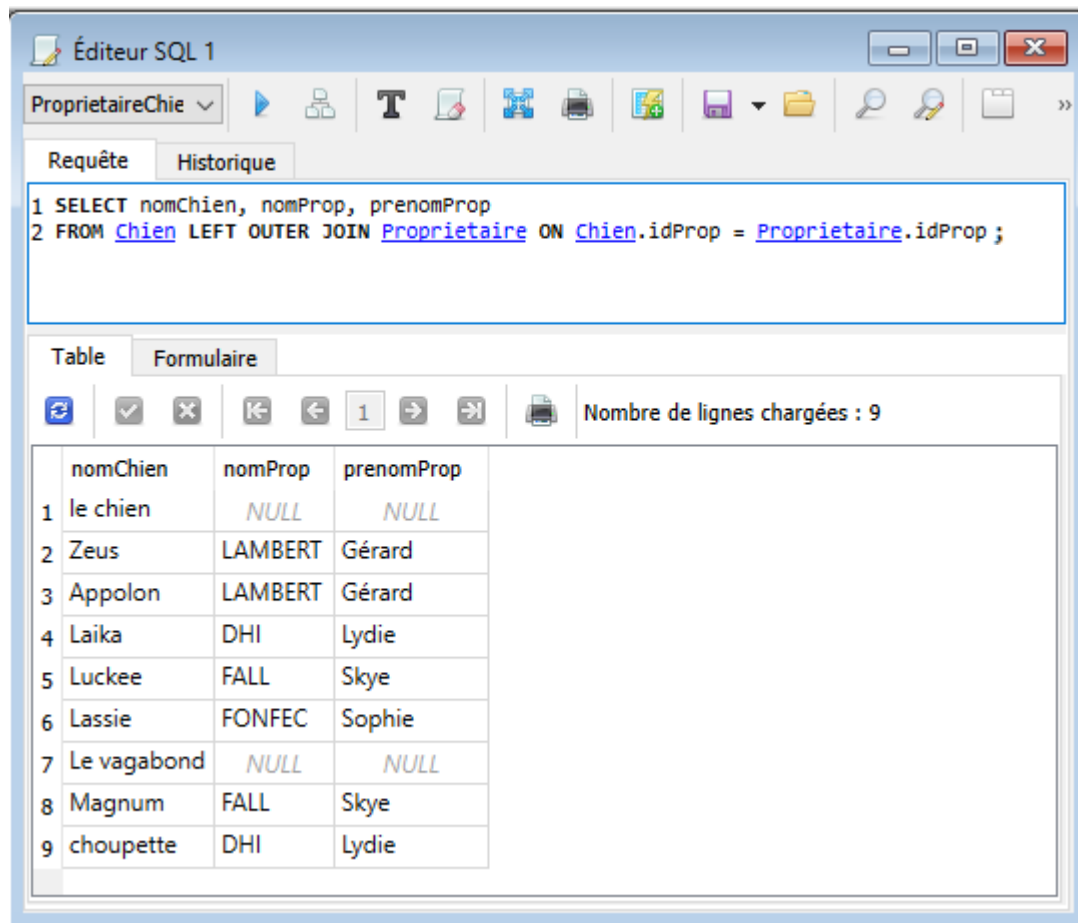
Below the query editor, the "Table" tab is active, displaying the results of the query in a table format. The table has three columns: nomChien, nomProp, and prenomProp. The results are as follows:

	nomChien	nomProp	prenomProp
1	Zeus	LAMBERT	Gérard
2	Appolon	LAMBERT	Gérard
3	Laika	DHI	Lydie
4	Medor	FALL	Skye
5	Lassie	FONFEC	Sophie

The interface also includes a toolbar with various icons for editing and viewing, and a status bar indicating "Nombre de lignes chargées : 5".

Jointure Externe ...

Liste les chiens avec leur propriétaire quand ils en ont



The screenshot shows a SQL editor window titled "Éditeur SQL 1". The query editor contains the following SQL statement:

```
1 SELECT nomChien, nomProp, prenomProp
2 FROM Chien LEFT OUTER JOIN Proprietaire ON Chien.idProp = Proprietaire.idProp ;
```

Below the query editor, the "Table" tab is selected, displaying the results of the query in a table. The table has three columns: nomChien, nomProp, and prenomProp. The results are as follows:

	nomChien	nomProp	prenomProp
1	le chien	NULL	NULL
2	Zeus	LAMBERT	Gérard
3	Appolon	LAMBERT	Gérard
4	Laika	DHI	Lydie
5	Luckee	FALL	Skye
6	Lassie	FONFEC	Sophie
7	Le vagabond	NULL	NULL
8	Magnum	FALL	Skye
9	chouquette	DHI	Lydie

The interface also includes a toolbar with various icons for editing and viewing the query, and a status bar indicating "Nombre de lignes chargées : 9".

Jointure Externe à droite ? ...

SQLite ne gère pas les jointures à **droite**... qu'importe ☺ , suffit de permuter l'ordre des tables.

Liste des « propriétaires » et de leur chien lorsqu'ils en ont.

Éditeur SQL 1

ProprietaireChie

Requête Historique

```
1 SELECT nomProp, prenomProp, nomChien
2 FROM Proprietaire LEFT OUTER JOIN Chien
3     ON Chien.idProp = Proprietaire.idProp;
```

Table Formulaire

	nomProp	prenomProp	nomChien
1	LAMBERT	Gérard	Appolon
2	LAMBERT	Gérard	Zeus
3	DHI	Lydie	Laika
4	DHI	Lydie	choupette
5	FALL	Skye	Luckee
6	FALL	Skye	Magnum
7	OUHALI	Kamel	NULL
8	FONFEC	Sophie	Lassie

SELECT nomChien, nomProp, prenomProp
FROM Chien LEFT OUTER JOIN Proprietaire ON Chien.idProp = Proprietaire.idProp

SELECT nomProp, prenomProp, nomChien
FROM Proprietaire LEFT OUTER JOIN Chien

« propriétaire »
sans chien

Malin !!!

Mais alors comment faire une jointure « complète » (FULL OUTER JOIN)

SQLite ne supporte pas cette « fonction » ☹ ...

... mais en combinant (UNION) les résultats d'une jointure externe à gauche .. et à droite ... soit 2 jointures à gauche (les 2 précédentes), on obtient le même résultat ☺

Éditeur SQL 1

ProprietaireChien

Requête Historique

```

1 SELECT nomChien, nomProp, prenomProp
2 FROM Chien LEFT OUTER JOIN Proprietaire
3     ON Chien.idProp = Proprietaire.idProp
4
5 UNION
6
7 SELECT nomChien, nomProp, prenomProp
8 FROM Proprietaire LEFT OUTER JOIN Chien
9     ON Chien.idProp = Proprietaire.idProp

```

Table Formulaire

	nomChien	nomProp	prenomProp
1	NULL	OUHALI	Kamel
2	Appolon	LAMBERT	Gérard
3	Laika	DHI	Lydie
4	Lassie	FONFEC	Sophie
5	Le vagabond	NULL	NULL
6	Luckee	FALL	Skye
7	Magnum	FALL	Skye

Suppose de demander les mêmes informations (colonnes) ... dans le même ordre !

Propriétaire sans chien

Propriétaire avec SON chien

Chien sans propriétaire

Propriétaire avec SES chiens

Etonnant, non ?



Pour celles et ceux qui veulent s'aventurer en mode
ligne de commande

SQLITE ... TOUT COURT

Démarrage rapide – Sqlite en ligne de commande

Créer une base de donnée qui contiendra nos tables :

1. Ouvrir une console en ligne de commande
2. Se positionner dans le répertoire de SQLite (commande CD – Change Directory)
3. Lancer le programme en indiquant le nom de la base de données (ex. sous Windows, sqlite sur une clef USB identifié par la lettre D:)

```
C:> D:  
D:> CD \ici\etla\sqlite3\  
D:> sqlite3.exe MaBD.db  
sqlite>
```

L'usage est de donner l'extension **.db** (database) aux fichiers Bases de données.

Indique qu'on est dans l'interface en mode texte de SQLite. Doivent suivre ensuite des commandes SQLite ou des requêtes SQL.
Nous avons ouvert la base de données MaBD

Démarrage rapide – Sqlite en ligne de commande

Une fois « inside » SQLite

- ❑ Exécuter des commandes SQLite (précédent d'un point) ou des requêtes SQL.

Quelques commandes SQLite : non lié à SQL

<code>.help</code>	<i>affiche les commandes usuelles</i>
<code>.output stdout</code>	<i>résultat affiché dans la console</i>
<code>.output FILE</code>	<i>résultat renvoyé dans un fichier</i>
<code>.tables ?Pattern?</code>	<i>Liste les tables correspondant au pattern (vide liste toutes les tables)</i>
<code>.read FILENAME</code>	<i>exécute les commandes SQL contenues dans FILENAME</i>
<code>.schema Table</code>	<i>affiche la structure de la table Table (ie code SQL - CREATE TABLE)</i>
<code>.mode column</code>	<i>les résultats des requêtes SELECT sont tabulées (so much nicer)</i>
<code>.headers on</code>	<i>ajoute en haut du résultat de la requête le nom des colonnes</i>
<code>.dump</code>	<i>“balance” le code SQL permettant de recréer la base de données (CREATE TABLE + INSERT INTO)</i>
<code>.exit</code>	<i>quitte SQLite</i>

```
sqlite> .output saveSQL.txt
sqlite> .dump
sqlite> .output stdout
```

Redirige la sortie vers un fichier texte
Sauve la base de données en équivalent SQL dans le fichier SaveSQL.txt
Rétabli la sortie vers l'écran (sinon on ne verrait plus le résultat de nos requêtes)