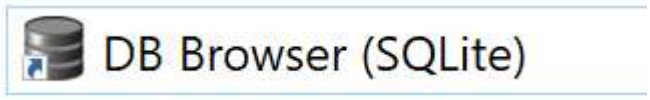
0. Objectifs

- Prise en main d'un SG 

- Etablissement des requêtes SQL de base pour créer une BDD et ses tables suivant un cahier des charges.

Le langage SQL utilisé est SQLite une version « Lite » allégée / transportable du langage SQL . Ce qui induit des différences dans les appellations des types des attributs entre autres mais pas dans l'organisation des requêtes.

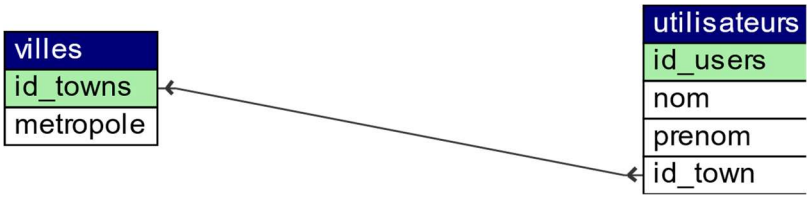
D'une manière générale chaque SGDB poosède quelques particularités qui lui sont propre et différentes du SQL « standard » **SQL:2011** ou **ISO/CEI 9075:2011. Il faut donc ne pas hésiter à lire la documentation.**

La documentation se trouve à la page https://www.sqlitetutorial.net/

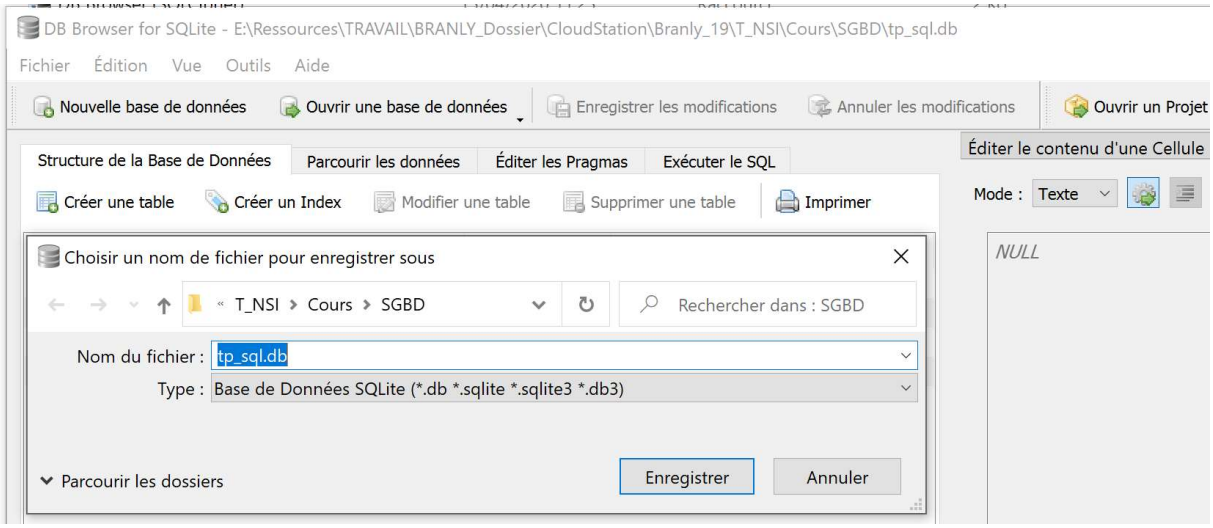
DB Brownser permet de créer graphiquement une base de données dans le TP on vous demande expressément d'utiliser des requêtes SQL (vous pouvez quand même vous aider des facilités du logiciel)

1. Création de la base de données « tp_sql »

Un fournisseur internet a besoin de connaitre la localisation des ses clients. Il établit une base de données « tp_sql »



1.1. Créer une nouvelle base de données « tp_sql»

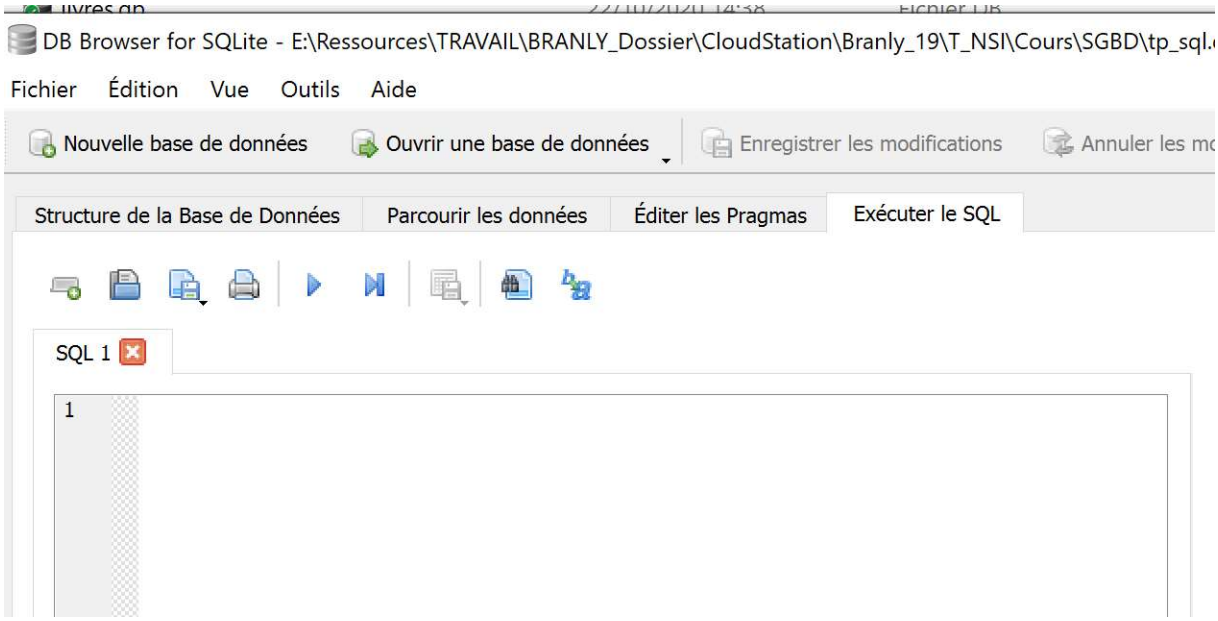


1.2. Insertion des tables dans la base de données

Ajouter deux tables « utilisateurs » et villes en utilisant une requête SQL dans l'ordre

villes(id_town INTEGER , metropole TEXT)

utilisateurs(id_user INTEGER , nom TEXT, prenom TEXT, genre TEXT, age INTEGER,# id_town)



La flèche bleue orientée à droite exécute la requête

Requète1

```
CREATE TABLE villes (
id_towns INTEGER PRIMARY KEY NOT NULL,
metropole TEXT
);
```

Requète2

```
CREATE TABLE utilisateurs (
id_users INTEGER PRIMARY KEY NOT NULL,
nom TEXT ,
prenom TEXT ,
genre TEXT,
age INTEGER ,
id_towns INTEGER,
FOREIGN KEY(id_towns) REFERENCES villes(id_town)
);
```

1.3. Remplissage des tables

Requête pour « villes » pour obtenir :

	id_town	metropole
	Filtre	Filtre
1	1	LYON
2	2	PARIS
3	3	MARSEILLE
4	4	MACON

```
INSERT INTO    villes(id_town,metropole)
VALUES
(1,"LYON"),
(2,"PARIS"),
(3,"MARSEILLE"),
(4,"MACON");
```

Requête pour « utilisateurs » pour obtenir

	id_users	nom	prenom	genre	age	id_towns
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	4	Louison	Bobet	homme	27	3
2	3	Alice	Granjean	femme	45	2
3	2	Louis	Durand	homme	37	1
4	1	Jean	Chouhan	homme	50	1

```
INSERT INTO
      utilisateurs(id_users,prenom,nom,genre,age,id_towns)
VALUES
(1,"Jean","Chouhan","homme",50,1),
(2,"Louis","Durand","homme",37,1),
(3,"Alice","Granjean","femme",45,2),
(4,"Louison","Bobet","homme",27,3);
```

1.4. Modification des tables

- Ajouter un 5ème enregistrement à la table « utilisateur »

```
INSERT INTO
      utilisateurs
VALUES
(5,"Arnaud","Champin","homme",23,1);
```

- Modifier l'âge de Durand par 83

```
UPDATE
utilisateurs
SET
age  =  83
WHERE
nom = "Durand"
```

1.5. Opérations d'agrégation

- Calculer la somme de tous les âges et la ranger dans l'attribut « somme_age »

```
SELECT SUM(age) AS somme_age
FROM
Utilisateurs
```

- Donner le nombre d'utilisateurs

```
SELECT COUNT (*) AS nb
FROM
Utilisateurs
```

Grouper ce résultat par genre

```
SELECT genre, COUNT (*) AS nb
FROM
utilisateurs
GROUP BY genre
```

	genre	nb
1	femme	1
2	homme	3

- Donner la moyenne des âges des utilisateurs et ranger la dans l'attribut « age_moyen »

```
SELECT AVG(age) AS age_moyen
FROM
Utilisateurs
```

	age_moyen
1	51.25

Différencier suivant les genres

```
SELECT genre , AVG(age) AS age_moyen
FROM
utilisateurs
GROUP BY genre
```

	genre	age_moyen
1	femme	45.0
2	homme	53.3333333333333

- Donner l'âge maximum de chacun des genres

```
SELECT genre , MAX(age) AS age_maximum
FROM
utilisateurs
GROUP BY genre
```

	genre	age_maximum
1	femme	45
2	homme	83

1.6. Un truc rigolo quand on a bien rempli toute une table !

Crée à l'aide du logiciel une table « matable » puis supprimer la.

```
DROP TABLE matable ;
```

1.7. Mise ne relation des deux tables

On désire obtenir les noms et prénoms des clients qui habitent LYON écrivez la requête correspondante. Le résultat devra faire apparaitre le prénom avant le nom.

```
SELECT prenom,nom
FROM utilisateurs AS u
INNER JOIN villes ON u.id_towns = villes.id_town
WHERE metropole ="LYON"
```

	prenom	nom
1	Chouhan	Jean
2	Durand	Louis
3	Champin	Arnaud