Une base de données (BDD) représente un ensemble ordonné de données dont l'organisation est régie par un modèle de données. Ce sont l'apparition des disques durs qui ont permis d'envisager le stockage des données dans les années 50. C'est lors du programme Apollo en 1960 que l'idée de la base de données a été lancée. Le but était de collecter des données afin de permettre d'aller sur la lune avant la fin de la décennie. Mais les bases de données ont pris leur envol avec l'émergence d'internet et l'apparition du big data, gigantesque collecte de données qu'il faut stocker, modifier, traiter. Quasiment toutes les bases de données actuelles sont basées sur les travaux d'Edgar F. Codd (1970).

I) Vocabulaire :

1. SGBD : Système de Gestion de Bases de Données :

Le SGBD peut être vu comme le logiciel qui gère les bases de données. Il permet de :

- Décrire, Modifier la structure de la Base de données,
- Mémoriser
- Interroger
- Traiter et maintenir les données

C'est l'interface entre les données et les utilisateurs.

2. Les principes de la conception du modèle :

Il faut déterminer les objectifs d'utilisation de la base de données :

- À quoi va-t-elle servir ?
- Comment l'utiliser ?
- Qui l'utilisera ?

Pour cela, il faut organiser les informations requises puis identifier les informations élémentaires en les regroupant par thème.

Exemple :

Produit	Fournisseur	Adresse
Ski alpin Lolitta	Rossignol	98 rue Louis Barran 38430 St Jean de Moirans
Ski alpin S1 Pro	Rossignol	98 rue Louis Barran 38430 St Jean de Moirans
Ski alpin S4 Pro JLB	Rosignol	98 rue Louis Barran 38430 St Jean de Moirans
S/Max 10 2020	Salomon	14 che des Croiselets 74370 Epagny Metz-Tessy

Peut être représenté par deux tables :

Produit
Ski alpin Lolitta
Ski alpin S1 Pro
Ski alpin S4 Pro JLB
S/Max 10 2020

Fournisseur	Adresse
Rossignol	98 rue Louis Barran 38430 St Jean de Moirans
Salomon	14 che des Croiselets 74370 Epagny Metz-Tessy

a) Le Champ (attribut):

Le champ est **une information élémentaire** représentant un **intérêt pour le domaine** étudié, **non redondante** et devant avoir un sens.

Exemple : Le nom d'un client, le prénom d'un client, le numéro de téléphone….

Le domaine de valeur d'un champ est un **ensemble de valeurs** pouvant être prises par un champ.

Exemples :

Prix >0, heure entre 0 et 23.

Un champ peut être composé. Exemple : Num INSEE : sexe + année + mois + département + commune + numéro ordre.

b) La table :

Une table est un **ensemble nommé et unique de champs liés entre eux**. Elle comprend une population d'individus homogènes.

Exemples :

Dans une banque, la tables CLIENTS est l'ensemble de personnes physiques ou morales possédant au moins un compte en banque.

Dans un magasin, la table PRODUITS contient la référence du produit, son nom, la quantité en stock et le prix.

PRODUITS
Référence
Nom
Stock
Prix

Une table est **représentée** sous **forme de tableau** ou chaque **ligne** correspond à un **enregistrement** (entité) et dont chaque **colonne** représente **un champ de l'enregistrement** de la table.

PRODUITS			
Référence	Nom	Stock	Prix
554871	Confiture de fraises 250g	5	4.50
554872	Confiture de fraises 500g	10	8.90
557911	Gelée de groseilles 250g	2	3.90
557912	Gelée de groseilles 500g	0	6.20
662458	Beurre doux 250g	5	1.79

Nom

4 Champs

Chaque enregistrement doit être unique.

c) Clé primaire :

Une **clé (identifiant) primaire** ne peut pas prendre deux fois la même valeur dans deux enregistrements différents. Plusieurs « types » de clé primaires existent :

- **Simple** (naturel) : **un seul champ**
- **Composée** : **plusieurs champs**.

La clé doit être **discriminante, stable, minimale** et doit être **unique**. Dans une table, on la surligne et on la met en gras ou on peut simplement rajouter une clé devant.

PRODUITS
Référence
Nom
Stock
Prix

Règles de vérification :

- Une table a une seule clé primaire.
- Une table possède au moins un champ (clé primaire).
- Pour chaque enregistrement d'une table, il ne peut y avoir au plus, qu'une valeur dans un champ.
- Tous les enregistrements d'une table sont homogènes.
- Un champ ne peut appartenir qu'à une seule table. Pour être dans cette table, il doit dépendre de la clé primaire.

d) Relations entre tables :

La relation entre tables, reflète l'interdépendance entre les tables. Elle peut être de différents « types » :

- Un à plusieurs
- Un à un
- Plusieurs à plusieurs

Lien un à plusieurs (cardinalité 1,N):

Exemple : une livraison correspond à un seul fournisseur mais un fournisseur peut effectuer plusieurs livraisons.

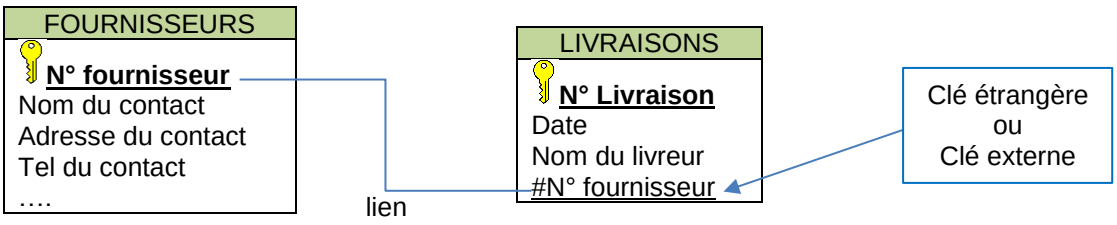


Table maître — Table esclave

Lien un à un (cardinalité 1,1):

Exemple : Un employé peut être responsable d'un service seul. Un service n'est dirigé que par un seul responsable

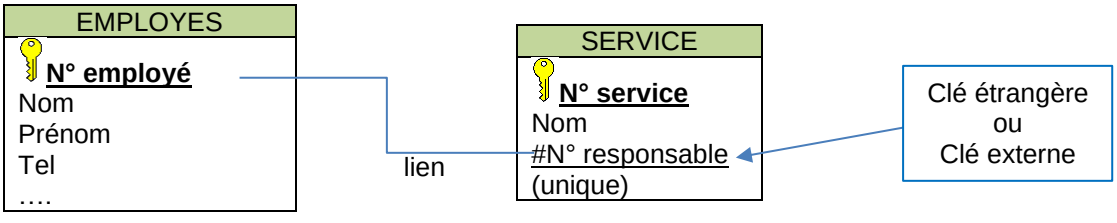
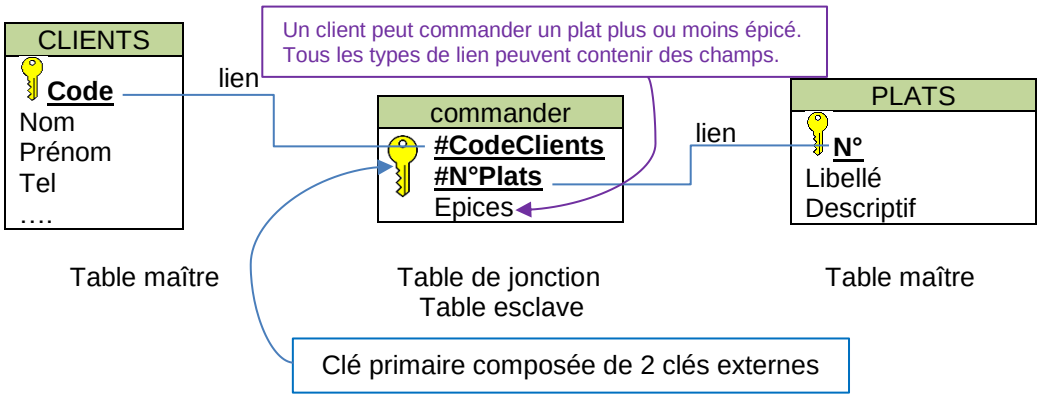


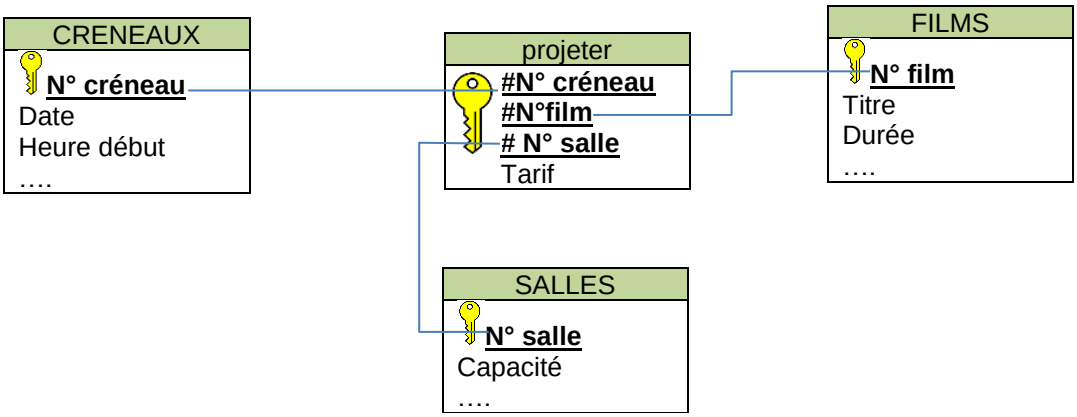
Table maître — Table esclave

Lien plusieurs à plusieurs :

Exemple : Un client peut commander plusieurs plats. Un plat peut être commandé par plusieurs clients. Il faudra créer une table de jonction.



Exemple de lien type plusieurs à plusieurs entre plusieurs tables :



3. Exercices :

On souhaite modéliser un annuaire téléphonique simple dans lequel chaque personne est identifiée par son nom, son prénom et son numéro de téléphone.
Proposer une table pour cet annuaire et précisez la clé principale.

On donne les enregistrements suivants, proposer une table associée avec la clé primaire :

Num_securite_sociale	Nom	Prénom	Date_naissance
182111425164224	DUPOND	Titi	23/11/1982
274037511510792	DURAND	Toto	17/03/1974
175087775621598	LEFRANC	Pierre	03/08/1975

Donner une modélisation relationnelle d'un bulletin scolaire. Cette dernière doit permettre de retrouver :

- Des élèves identifiés par un numéro étudiant unique, un nom et un prénom
- Un ensemble de matière identifie par le numéro de matière et l'intitulé de la matière
- Les notes identifiées par la valeur de la note, la matière et l'élève

Donner une modélisation relationnelle d'un réseau de bus. Cette dernière doit permettre de retrouver :

- Le réseau de bus constitué d'arrêts de bus identifiés par ces coordonnées GPS (longitude et latitude) ainsi que le nom et un identifiant unique.
- Il est aussi constitué de lignes de bus représentées par un identifiant et un nom
- Les horaires de passage sont eux identifiés par ligne de bus et arrêt dont on stockera les horaires de passage ainsi que les jours de validité de l'horaire (semaine, samedi ou le dimanche/jour férié)

On donne la base de données suivantes :

Table « Ticket » :

NumTicket	Date	Heure	NumCarteFidelite	ModeRéglement
2132	19/10/2019	14 :32	900000001	CB
3143	20/10/2019	09 :22		ESPECE
6192	21/10/2019	11 :15	900000023	CHEQUE
7193	22/10/2019	18 :15	900000142	CB

Table « CatégorieProduit » :

CodeCategorie	Libellé
1	Alimentaire
2	Hors Alimentaire

Table « Client » :

NumCarteFidelite	Nom	Prénom	Adresse	CodePostal	Ville	DateDeNaissance
	CLIENT CAISSE					
900000001	CIEL	Leïa	1 grande rue	39100	DOLE	28/04/2003
900000023	Marcheur	Luc	15 rue des granges	25000	BESANCON	13/10/2000
900000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

Table « Produit » :

CodeProduit	NomProduit	PrixtTTC	CodeCategorie	EnStock(O/N)
31	1l Jus Pom. Brt	1.78	1	O
34	2kg orange jus	3,49	1	O
35	1,5kg orange esp	2,25	1	N
37	Baguette Rust	0,9	1	O
39	Bsc Ptt Lycéen	2,45	1	O
40	Bsc Dino	2,89	1	N
44	Dent. TpWhite	1.09	2	O
47	Galette rois	7,5	1	O
51	Huile 5W30 3l	15,95	2	O
50	Lave glace E 5l	2	1	N
54	Lave glace H 5l	2,5	1	O
59	Mgrt Canard	7,52	1	O
61	N&N'S 250g	3,49	1	O
70	Pain épice miel	2.12	1	O
71	Semoule Kebab	2,94	1	O
83	Buche citron	9,42	1	O

Table « AcheterProduit » :

NumTicket	CodeProduit	Quantité
2132	39	1
2132	61	1
2132	83	2
2132	71	1
2132	44	1
3143	70	3
3143	37	1
3143	59	1
3143	34	1
6192	47	1
6192	37	1
6192	31	2
6192	51	1
6192	54	1
7193	61	2
7193	70	1
7193	47	4

Donner le modèle relationnel entre les différentes tables ainsi que les clés primaires.

II) Structuration des tables :

1. Stockage des données :

Pour des raisons de gestion de l'espace disque et d'organisation des enregistrements, chaque propriété est d'un type précis.

Les types de propriétés les plus fréquents sont :

- Numérique : Entier, Réel
- Texte (taille illimitée), Chaîne de n caractères,
- Date, heure
- Booléen (uniquement 2 valeurs possibles : vrai/oui ou faux/non)

Il existe aussi des types liés à la géolocalisation (coordonnées d'un point, coordonnées du contour d'une forme géographique, …) ……

Pour le supermarché, associer à chacun des champs précédents, le type de variable :

Table	Propriété	Type
Produit	CodeProduit	*Numérique entier*
	NomProduit	*Texte 20 caractères*
	PrixtTTC	
	CodeCategorie	
	EnStock(O/N)	
CatégorieProduit	CodeCategorie	
	Libellé	
Client	NumCarteFidelite	
	Nom	
	Prénom	
	Adresse	
	CodePostal	
	Ville	
	DateDeNaissance	
Ticket	NumTicket	
	Date	
	Heure	
	NumCarteFidelite	
	ModeRéglement	
AcheterProduit	NumTicket	
	CodeProduit	
	Quantité	

2. La base de données :

Pour présenter le langage SQL nous nous appuierons sur SQLite à télécharger ici.

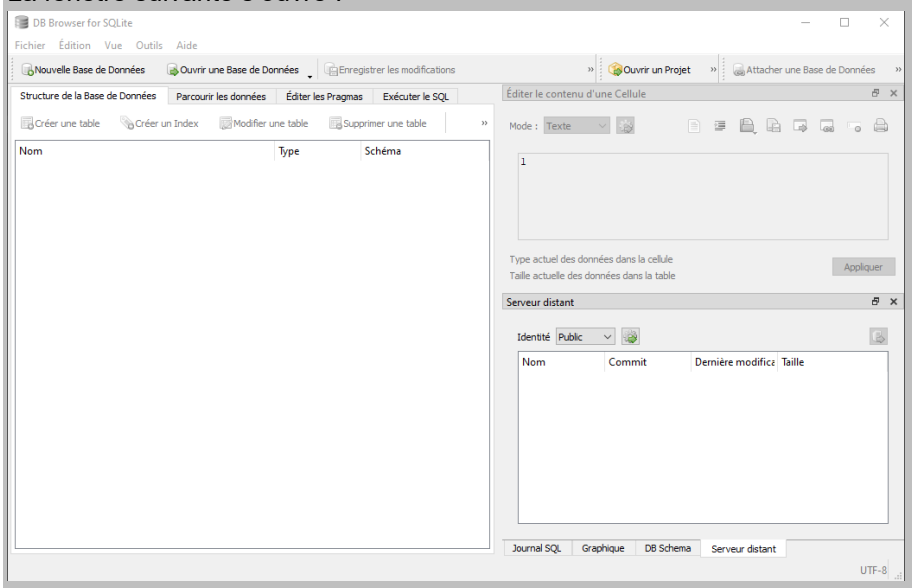
Remarque : SQLite implémente un nombre restreint de types de données. En particulier il n'existe pas de chaîne limitée à un nombre prédéfini de caractères ni de type année. Il reste cependant possible d'utiliser ces types dans la définition du schéma relationnel, mais il faut garder à l'esprit que ceux-ci se verront convertis automatiquement en texte ou en entier.

Nous allons créer la petite base de données tickets de caisse donnée ci-dessous.

Pour cela :

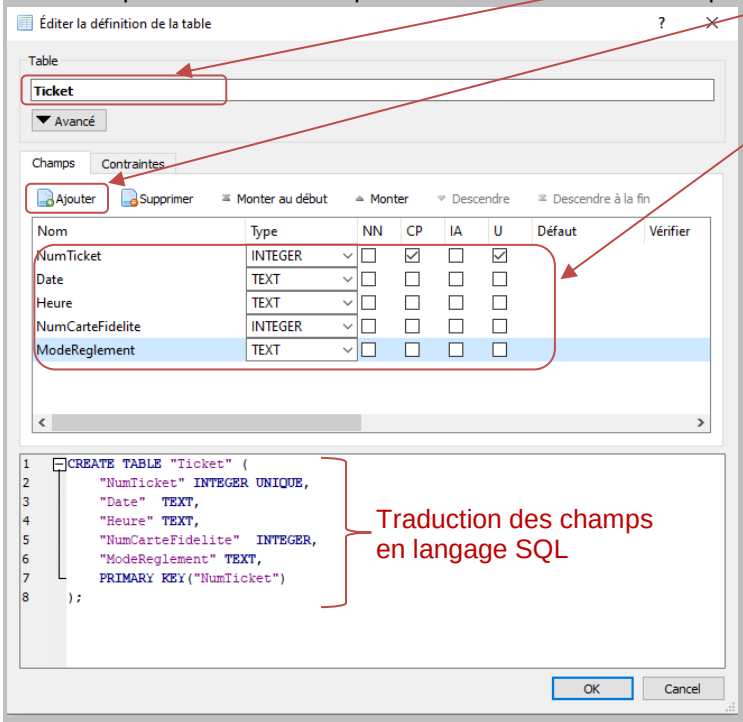
Exécuter SQLiteDatabaseBrowserPortable.exe après l'avoir extrait sur votre disque.

La fenêtre suivante s'ouvre :

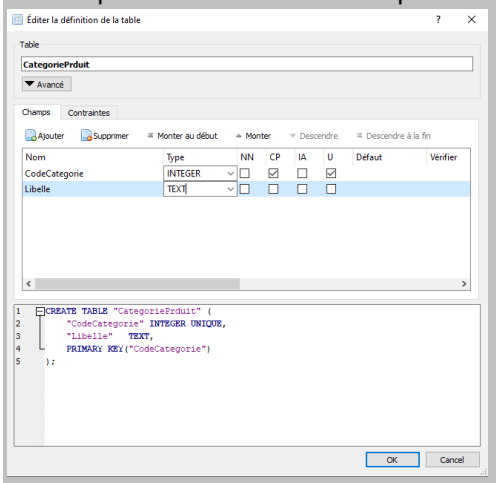


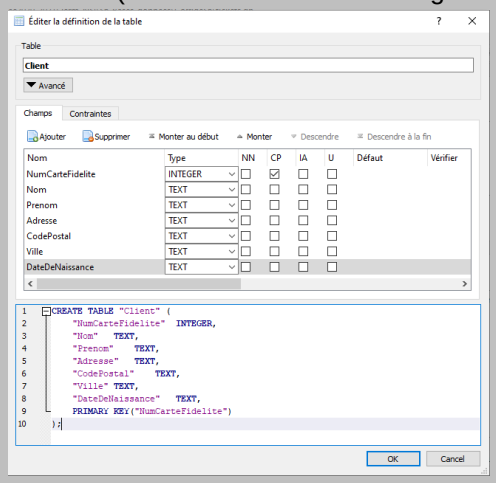
Cliquer sur Nouvelle Base de données et sauver-sous tickets.db.

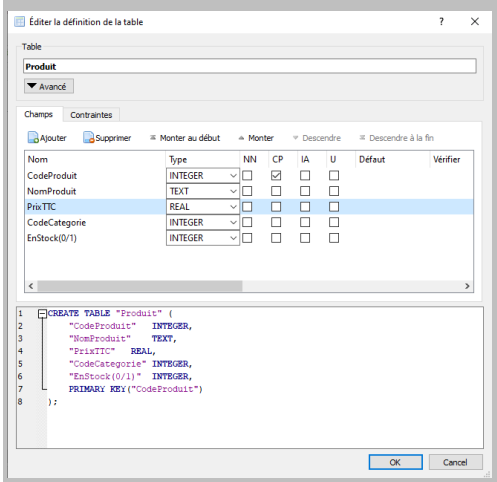
Commencer par créer la table Ticket, étant donné qu'il n'y a pas de type date et heure, nous mettrons texte à la place. Commencer par donner le nom de la table puis ajouter les champs.

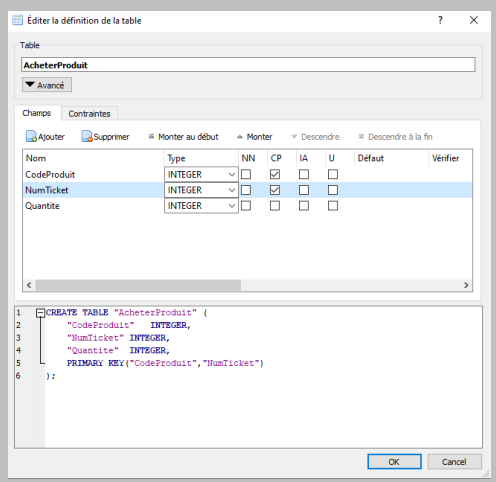


Valider par OK et faire de même pour les 4 autres tables (le booléens seront de integer à 0 ou 1)

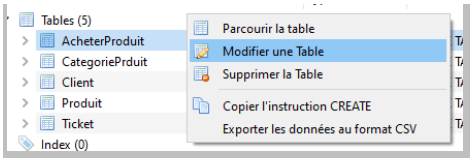




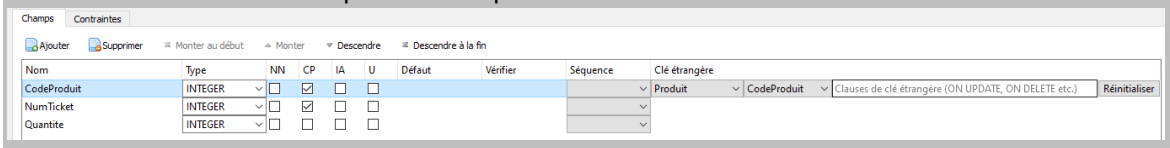




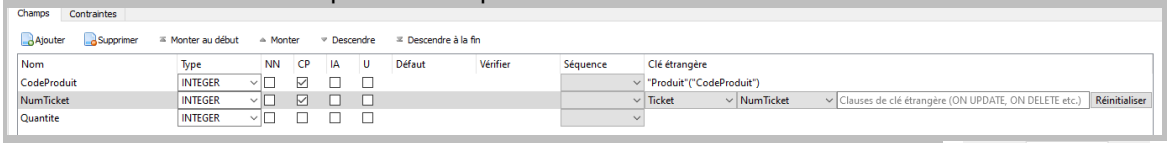
Reste à créer les clés externes qui traduisent les liens.
Pour cela il faut aller dans la table concernée et faire modifier la table.



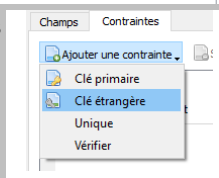
Sélectionner le champ CodeProduit puis dans la colonne Clé étrangère double cliquer pour sélectionner la table Produit puis le champ CodeProduit.

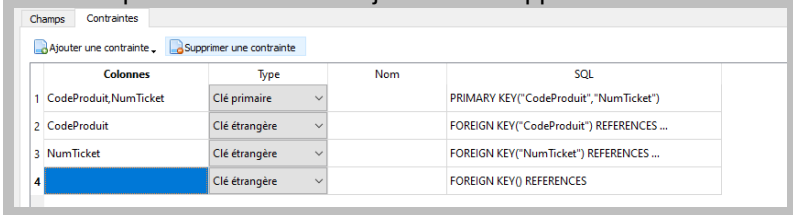


Sélectionner le champ NumTicket puis dans la colonne Clé étrangère double cliquer pour sélectionner la table Ticket puis le champ NumTicket

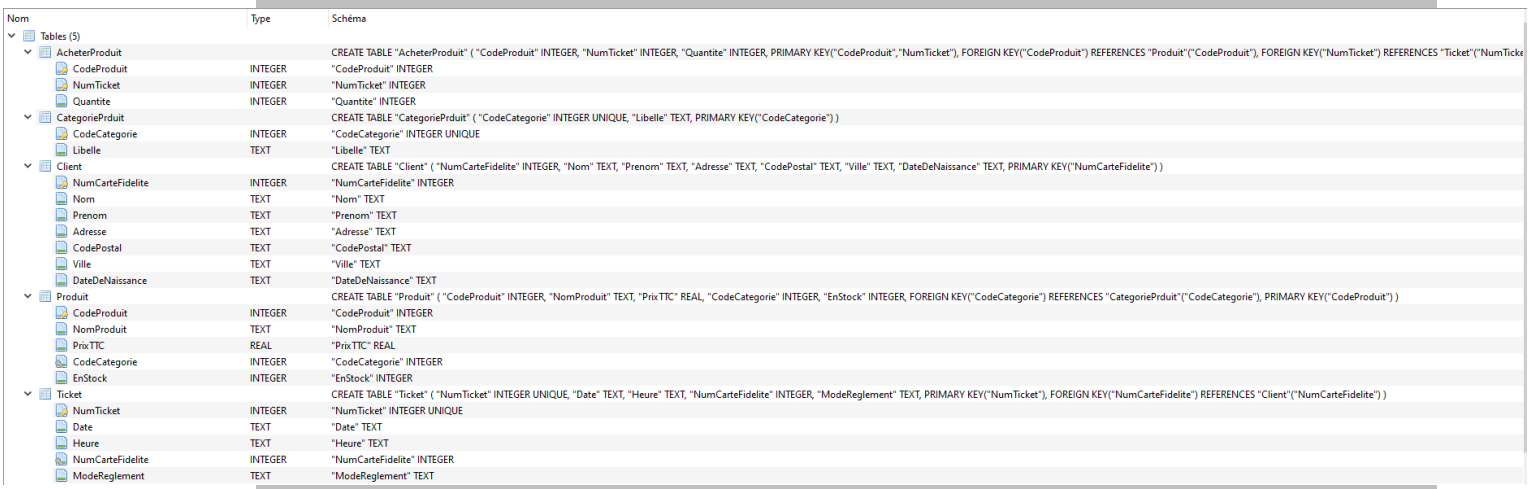


Maintenant reste à valider ces contraintes en allant dans l'onglet contraintes puis Ajouter une contrainte/Clé étrangère.
Les deux précédentes seront ajoutées et supprimer le troisième.





Valider par OK
Faire de même pour le champ CodeCategorie dans la table Produit et NumCarteFidelite dans la table Ticket.



Créons les données des tables.

Table « Ticket » :

NumTicket	Date	Heure	NumCarteFidelite	ModeRéglement
2132	19/10/2019	14 :32	900000001	CB
3143	20/10/2019	09 :22		ESPECE
6192	21/10/2019	11 :15	900000023	CHEQUE
7193	22/10/2019	18 :15	900000142	CB

Table « CatégorieProduit » :

CodeCategorie	Libellé
1	Alimentaire
2	Hors Alimentaire

Table « Client » :

NumCarteFidelite	Nom	Prénom	Adresse	CodePostal	Ville	DateDeNaissance
	CLIENT CAISSE					
900000001	CIEL	Leïa	1 grande rue	39100	DOLE	28/04/2003
900000023	Marcheur	Luc	15 rue des granges	25000	BESANCON	13/10/2000
900000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

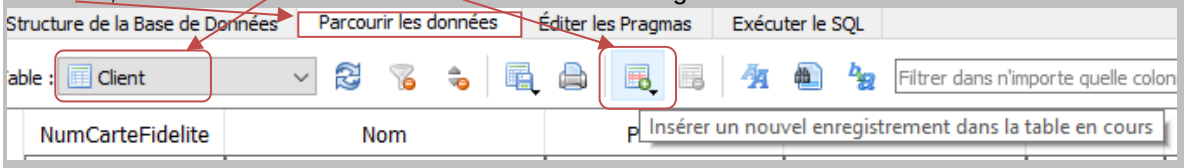
Table « Produit » :

CodeProduit	NomProduit	PrixtTTC	CodeCategorie	EnStock(O/N)
31	1l Jus Pom. Brt	1.78	1	O
34	2kg orange jus	3.49	1	O
35	1,5kg orange esp	2.25	1	N
37	Baguette Rust	0.9	1	O
39	Bsc Ptt Lycéen	2.45	1	O
40	Bsc Dino	2.89	1	N
44	Dent. TpWhite	1.09	2	O
47	Galette rois	7.5	1	O
51	Huile 5W30 3l	15.95	2	O
50	Lave glace E 5l	2	1	N
54	Lave glace H 5l	2.5	1	O
59	Mgrt Canard	7.52	1	O
61	N&N'S 250g	3.49	1	O
70	Pain épice miel	2.12	1	O
71	Semoule Kebab	2.94	1	O
83	Buche citron	9.42	1	O

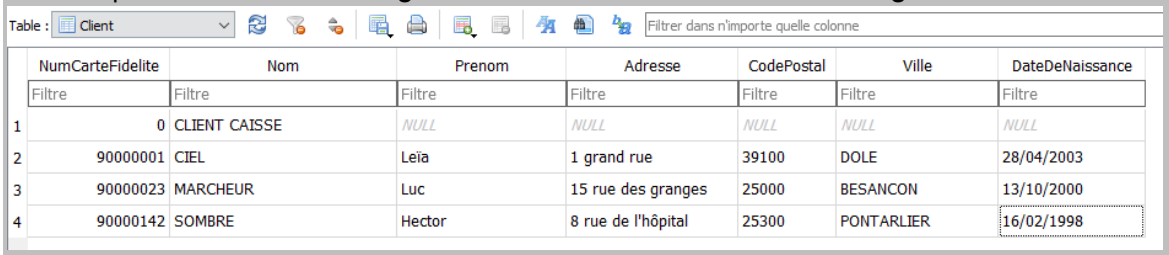
Table « AcheterProduit » :

NumTicket	CodeProduit	Quantité
2132	39	1
2132	61	1
2132	83	2
2132	71	1
2132	44	1
3143	70	3
3143	37	1
3143	59	1
3143	34	1
6192	47	1
6192	37	1
6192	31	2
6192	51	1
6192	54	1
7193	61	2
7193	70	1
7193	47	4

Commençons par la table Client qui n'a pas de clé externe pour cela aller dans l'onglet Parcourir les données, choisir la table Client et insérer un nouvel enregistrement dans la table en cours.

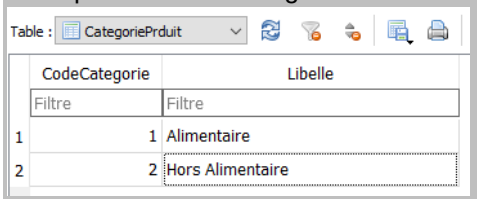


Puis remplir la table et faire **Enregistrer les modifications avant de changer de table.**



	NumCarteFidelite	Nom	Prenom	Adresse	CodePostal	Ville	DateDeNaissance
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	CLIENT CAISSE	NULL	NULL	NULL	NULL	NULL
2	90000001	CIEL	Leïa	1 grand rue	39100	DOLE	28/04/2003
3	90000023	MARCHEUR	Luc	15 rue des granges	25000	BESANCON	13/10/2000
4	90000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

Idem pour la table CategorieProduit **Enregistrer les modifications avant de changer de table.**



	CodeCategorie	Libelle
	Filtre	Filtre
1	1	Alimentaire
2	2	Hors Alimentaire

Puis Ticket : on remarquera que les NumCarteFidelite sont proposés car il y a une clé externe.

Structure de la Base de Données | Parcourir les données | Éditer les Pragmas | Exécuter le SQL

Table : Ticket — Filtrer dans n'importe quelle colonne

	IumTicket	Date	Heure	NumCarteFidelite	ModeReglement
	Filtre	Filtre	Filtre	Filtre	Filtre
1	2132	19/10/2019	14:32	90000001	CB
2	3143	20/10/2019	09:22	NULL	ESPECE
3	6192	21/10/2019	11:15	90000023	CHEQUE
4	7193	22/10/2019	18:15	90000142	CB

Puis Produit (une fois tous les enregistrements créés (16 au total), vous pouvez faire un copier/coller des colonnes CodeProduit, NomProduit, PrixTTC et CodeCategorie)

Structure de la Base de Données | Parcourir les données | Éditer les Pragmas | Exécuter l

Table : Produit

	CodeProduit	NomProduit	PrixTTC	CodeCategorie	EnStock
	Filtre	Filtre	Filtre	Filtre	Filtre
1	31	1l Jus Pom. Brt	1.78	1	1
2	34	2kg orange jus	3,49	1	1
3	35	1,5kg orange esp	2,25	1	0
4	37	Baguette Rust	0,9	1	1
5	39	Bsc Ptt Lycéen	2,45	1	1
6	40	Bsc Dino	2,89	1	0
7	44	Dent. TpWhite	1.09	2	1
8	47	Galette rois	7,5	1	1
9	50	Huile 5W30 3l	15,95	2	1
10	51	Lave glace E 5l	2.0	1	0
11	54	Lave glace H 5l	2,5	1	1
12	59	Mgrt Canard	7,52	1	1
13	61	N&N'S 250g	3,49	1	1
14	70	Pain épice miel	2.12	1	1
15	71	Semoule Kebab	2,94	1	1
16	83	Buche citron	9,42	1	1

Structure de la Base de Données | Parcourir les

Table : AcheterProduit

	CodeProduit	NumTicket	Quantite
	Filtre	Filtre	Filtre
1	39	2132	1
2	61	2132	1
3	83	2132	2
4	71	2132	1
5	44	2132	1
6	70	3143	3
7	37	3143	1
8	59	3143	1
9	34	3143	1
10	47	6192	1
11	37	6192	1
12	31	6192	2
13	51	6192	1
14	54	6192	1
15	61	7193	2
16	70	7193	1
17	47	7193	4

Procéder de même pour la table AcheterProduit.
La base de données est terminée.

3. Les requêtes SQL :

L'instruction de base pour l'interrogation d'une base de données en SQL est constituée du mot-clé **SELECT** suivi du mot-clé **FROM**.

- **SELECT** permet de sélectionner les attributs dont il faut afficher les valeurs. Le caractère * permet d'afficher les valeurs de tous les attributs.
- **FROM** permet de sélectionner la relation à explorer.

Exemple : on souhaite afficher le nom de tous les clients : on exécutera l'instruction SELECT Nom FROM Client dans l'onglet Exécuter le SQL.

Structure de la Base de Données | Parcourir les données | Éditer les Pragmas | Exécuter le SQL

Exécuter Tout ou seulement le SQL sélectionné [, F5, Ctrl+R]

SQL 1

```
SELECT Nom FROM Client
```

	Nom
1	CLIENT CAISSE
2	CIEL
3	MARCHEUR
4	SOMBRE

Résultat de la requête

On peut afficher plusieurs colonnes, exemple si l'on veut le nom et prénom, il suffit de séparer d'une virgule.

SQL 1

```
SELECT Nom,Prenom FROM Client
```

	Nom	Prenom
1	CLIENT CAISSE	NULL
2	CIEL	Leïa
3	MARCHEUR	Luc
4	SOMBRE	Hector

On peut lui faire afficher toute la table avec *.

```
SELECT * FROM Client
```

	NumCarteFidelite	Nom	Prenom	Adresse	CodePostal	Ville	DateDeNaissance
1	0	CLIENT CAISSE	NULL	NULL	NULL	NULL	NULL
2	90000001	CIEL	Leïa	1 grand rue	39100	DOLE	28/04/2003
3	90000023	MARCHEUR	Luc	15 rue des granges	25000	BESANCON	13/10/2000
4	90000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

Donner l'instruction pour afficher les numéros de ticket à partir de la table AcheterProduit.

On constate qu'il y a plusieurs fois la même valeur. Il est possible de ne pas afficher les doublons à l'aide du mot clé **DISTINCT**.

Tester l'instruction **SELECT DISTINCT** NumTicket **FROM** AcheterProduit pour afficher les numéros de ticket à partir de la table AcheterProduit.

```
SELECT DISTINCT NumTicket FROM AcheterProduit
```

	NumTicket
1	6192
2	3143
3	2132
4	7193

On peut aussi trier ces valeurs à l'aide du mot clé **ORDER BY** (pour l'ordre décroissant, il faut rajouter **DESC**)

Tester l'instruction **SELECT DISTINCT** NumTicket **FROM** AcheterProduit **ORDER BY** NumTicket pour afficher les numéros de ticket à partir de la table AcheterProduit du plus petit au plus grand.

```
SELECT DISTINCT NumTicket FROM AcheterProduit ORDER BY NumTicket
```

	NumTicket
1	2132
2	3143
3	6192
4	7193

Télécharger la base de données suivantes : Communes.db

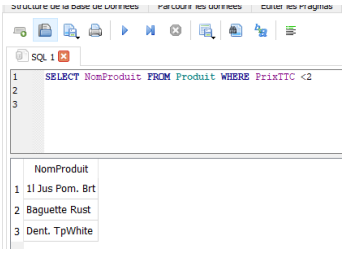
	idVille	nomVille	population	nbElusMunicipaux	nbInscritsListesElectorales	exprimesPremierTour	exprimesDeuxiemeTour
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1001	L'Abergement-Clémenciat	776	15	598	341	NULL
2	1002	L'Abergement-de-Varey	248	11	212	188	NULL
3	1004	Ambérieu-en-Bugey	14035	33	8101	4585	4368
4	1005	Ambérieux-en-Bombes	1689	19	1186	479	NULL
5	1006	Ambléon	111	11	99	66	NULL
6	1007	Ambronay	2726	23	1883	1191	1304
7	1008	Ambutrix	752	15	551	337	NULL
8	1009	Andert-et-Condon	330	11	269	176	NULL
9	1010	Anglefort	1115	15	733	520	NULL

1. Quelle requête permet d'afficher tous les noms de commune ?
2. Quelle requête permet d'afficher tous les noms de commune et leur population ?
3. Quelle requête permet d'afficher tous les noms de commune et leur population, classé par ordre croissant de population ?
4. Quelle requête permet d'afficher tous les noms de commune et leur population, classé par ordre décroissant d'élus municipaux ?

Faire une capture pour chaque question.

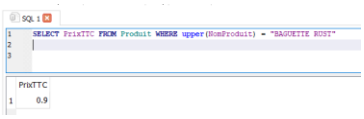
Une restriction est une sélection de lignes d'une relation, sur la base d'une condition à respecter, définie à la suite du terme **WHERE**. Cette condition peut être une combinaison de comparaisons (=, <>, >, >=, <, <=) à l'aide de **AND**, de **OR** et de **NOT** (attention donc aux parenthèses dans ce cas).

Reprenons notre base de données ticket.db et affichons tous les articles dont le prix est inférieur à 2€ : **SELECT** NomProduit **FROM** Produit **WHERE** PrixTTC<2.



Attention : Pour les comparaisons de chaînes de caractères, il est important de faire attention à la casse. Par définition, un "a" est donc différent d'un "A". Pour remédier à ce problème, il existe les fonction upper() et lower() pour transformer une chaîne en respectivement majuscule et minuscule.

SELECT PrixTTC **FROM** Produit **WHERE upper(NomProduit)** = "BAGUETTE RUST"



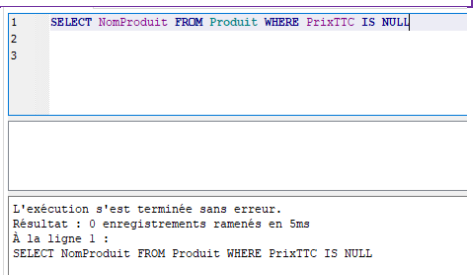
Proposer une requête pour afficher les produits dont le prix est supérieur à 10€.
Faire une capture du résultat.

Proposer une requête pour afficher les produits dont le prix est supérieur à 5€ et en stock.
Faire une capture du résultat.

A l'aide de la base de données Communes.db précédente :
1. Quelle requête permet d'afficher le nom des communes dont le nombre d'élus municipaux est strictement inférieur à 10 ?
2. Quelle requête permet d'afficher le nom des communes dont le nombre d'élus municipaux est strictement supérieur à 10 ET dont la population est inférieure ou égale à 150 ?
3. Quelle requête permet d'afficher le nombre d'élus municipaux et la population de la ville de «anglefort » ? (attention à la casse)

Une **donnée manquante** en SQL est repérée par un **NULL**. Il y a plusieurs raisons, bonnes ou mauvaises, pour avoir des données manquantes, et il est parfois utile de tester leur présence. Pour cela, nous allons utiliser le terme **IS NULL** comme condition.
Au contraire, si l'on veut uniquement les données pour lesquels l'information est présente, nous devrons utiliser la négation avec **IS NOT NULL**.

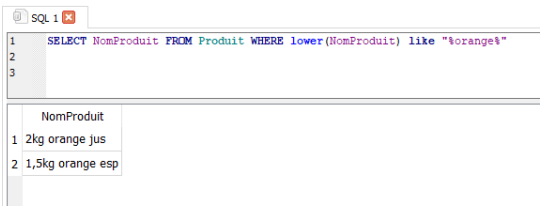
Dans notre base de données ticket.db, vérifions s'il y a bien un prix pour chaque article de la table NomProduit.
On constate que c'est bien le cas puisqu'il n'y a aucun enregistrement qui correspond à la requête.



A l'aide de la base de données Communes.db précédente :
1. Quelle requête permet d'afficher le nom des communes dans lesquelles il y a eu un deuxième tour ?
2. Quelle requête permet d'afficher le nom des communes dont la population est supérieure à 5 000 habitants et dans lesquelles il y a eu un deuxième tour ?

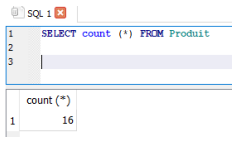
L'opérateur **like** permet de rechercher les **valeurs contenant une partie seulement de la chaîne de caractères**. Le caractère % représente une suite de caractères, éventuellement nulle.

Dans notre base de données ticket.db, on souhaite afficher tous les produits à base d'oranges, on va donc rechercher le mot orange. (Attention à la casse)



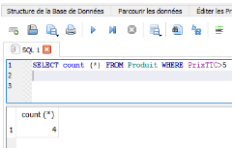
A l'aide de la base de données Communes.db précédente :
1. Quelle requête permet d'afficher le nom des communes dont le nom commence par Roncherolles ?
2. Quelle requête permet d'afficher le nom des communes dont le nom contient le mot « -en- » ?

- La fonction d'agrégation **count()** permet de compter le nombre d'enregistrement dans une table.



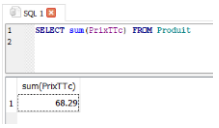
Pour compter le nombre de produits dans la table NomProduit : **SELECT count** (*) **FROM** Produit

Pour compter le nombre de produits dont le prix est supérieur à 5 € : **SELECT count** (*) **FROM** Produit **WHERE** PrixTTC>5



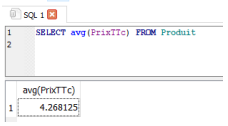
- La fonction **sum**(attribut) permet de faire la somme des valeurs non nulles de l'attribut passé en paramètre.

SELECT sum(PrixTTc) **FROM** Produit



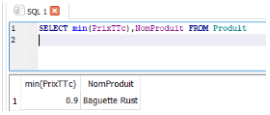
- La fonction **avg**(attribut) permet de faire la moyenne des valeurs de l'attribut passé en paramètre.

SELECT avg(PrixTTc) **FROM** Produit



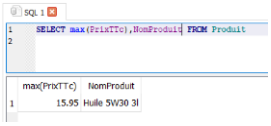
- La fonction **min**(attribut) permet d'afficher la valeur minimale de l'attribut passé en paramètre.

SELECT min(PrixTTc),NomProduit **FROM** Produit



- La fonction **MAX**(attribut) permet d'afficher la valeur maximale de l'attribut passé en paramètre.

SELECT max(PrixTTc),NomProduit **FROM** Produit



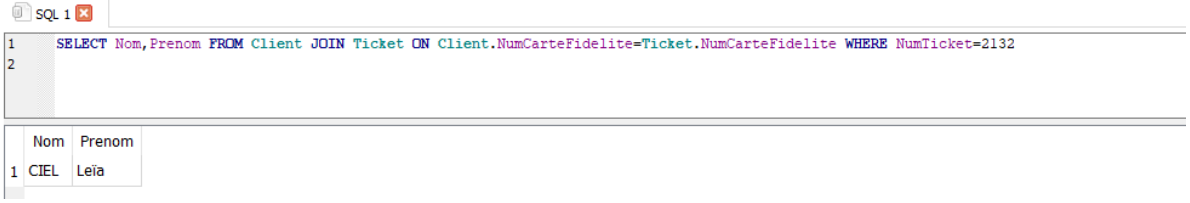
A l'aide de la base de données Communes.db précédente :

1. Quelle requête permet d'afficher le nombre total d'élus de toutes les communes ?
2. Quelle requête permet d'afficher la moyenne du nombre d'habitants dans les communes ?
3. Quelle requête permet d'afficher le nom de la ville avec la population la plus grande pour laquelle il n'y a pas de deuxième tour ?

Dans un modèle relationnel, une clé étrangère référence une clé primaire venant d'une autre table. Cela s'appelle une jointure. Grâce à celle-ci, on peut récupérer des données provenant de plusieurs tables.

La syntaxe est : table1 **JOIN** table2 **ON** table1.champ1 = table2.champ2
La jointure se positionne après les clauses **SELECT** et **FROM**.

Exemple : on souhaite récupérer le nom et prénom de la personne qui a fait le ticket 2132. Cela est possible en faisant une jointure entre la table Ticket (clé étrangère NumCarteFidelite) et Client.
SELECT Nom, Prenom **FROM** Client **JOIN** Ticket **ON** Ticket.NumCarteFidelite = Client.NumCarteFidelite **WHERE** NumTicket=2132



Reprenons notre base de données ticket.db :

1. Quelle requête permet d'obtenir le nom de tous les articles achetés ainsi que leurs quantités par le ticket 3143.
2. Quelle requête permet d'obtenir le nom de tous les articles achetés ainsi que leurs quantités par Luc Marcheur.

III) EXERCICES :

Exo1 :

Soit la base de données d'un festival de musique : dans une représentation peut participer un ou plusieurs musiciens. Un musicien ne peut participer qu'à une seule représentation.

- Representation (numRep , titreRep , lieu)
- Musicien (numMus , nom , #numRep)
- Programmer (date , #numRep, tarif)

Indiquez les requêtes qui permettent d'obtenir : (vous pourrez tester vos réponses avec cette base)

1. La liste des titres des représentations.
2. La liste des titres des représentations ayant lieu à Khatu.
3. La liste des noms des musiciens et les titres des représentations auxquelles ils participent.
4. La liste des titres des représentations, les lieux et les tarifs du 14/02/2020.
5. Le nom des musiciens qui participent à la représentations n°20.
6. Les représentations et leurs dates dont le tarif ne dépasse pas 20 euros.

Exo2 :

Soit la Base de données hôtel qui contient 3 tables :

- Chambre (numChambre, prix, nbrLit, nbrPers, confort, equipement)
- Client (numClient, nom, prenom, adresse)
- Reservation (# numClient, # numChambre, dateArr, dateDep)

Table : Chambre

numChambre	prix	nbrLit	nbrPers	confort	equipement
10	80	1	2	WC	NON
20	100	2	2	Douche	NON
25	180	3	3	Bain	TV

Table : Client

numClient	nom	prenom	adresse
1000	DURAND	Pierre	STRASBOURG
1001	NOUA	Paul	MULHOUSE

Table Reservation

numClient	numChambre	dateArr	dateDep
1000	10	05/02/2019	07/02/2019
1001	20	02/11/2020	

Indiquez les requêtes qui permettent d'obtenir : (vous pourrez tester vos réponses avec cette base)

1. Les numéros de chambres avec TV.
2. Les numéros de chambres et leurs capacités.
3. La capacité théorique d'accueil de l'hôtel.
4. Le prix par personne des chambres avec TV.
5. Les numéros des chambres et le numéro des clients ayant réservé des chambres pour le 05/02/2019.
6. Les numéros des chambres coûtant au maximum 80 Euro ou ayant un bain et volant au maximum 120 Euro.
7. Les Noms, Prénoms et adresses des clients dont le nom commence par « D ».
8. Le nombre de chambres dont le prix est entre 85 et 120 Euro.
9. Les noms des clients n'ayant pas fixé la date de départ.

IV) Manipuler les données :

1. Base de données Nobel :

Télécharger la base de données nobel.db et l'ouvrir avec SQlite pour répondre aux questions suivantes:

1. Combien de tables possède la base de données ?
2. Combien de champs possède la table Nobel ?
3. Quel est le type du champ annee ?
4. Combien d'enregistrement possède la table nobel ?

5. Dans quelle discipline Paul Krugman est-il devenu Prix Nobel ?
6. En quelle année Albert Fert a-t-il eu le prix Nobel ?

En utilisant l'onglet Exécuter le SQL, indiquez le code SQL permettant de répondre aux questions suivantes (faire une capture de la requête et du résultat) :

7. Comment afficher le nom de tous les lauréats en évitant les doublons ?
8. Comment afficher le nom de toutes les disciplines en évitant les doublons ?
9. Quelle est la discipline de Wilhelm Conrad Röntgen ?
10. Dans quelle discipline Paul Krugman est-il devenu Prix Nobel ?
11. En quelle année Albert Fert a-t-il eu le prix Nobel ?
12. Quelle est l'année de distinction de Pierre Curie ?
13. Quelle est l'année de distinction et la matière de Bertha von Suttner ?
14. Quels sont les lauréats distingués au XXI e siècle ?
15. Quels sont les lauréats du prix Nobel de la Paix durant la deuxième guerre mondiale ?
16. Quels sont les lauréats distingués en Médecine en 1901 et en 2001 ?
17. Quels sont les lauréats des prix nobel de Physique et de Médecine en 2008 ?
18. Combien d'enregistrements au total comporte la table ?
19. Combien de personnes ont reçu le prix Nobel de la paix ?
20. Combien de personnes ont reçu le prix Nobel de littérature ?
21. Combien de personnes ont reçu le prix Nobel de mathématiques ?
22. Combien de personnes ont reçu un prix Nobel en 1901 ?
23. Combien de personnes ont reçu un prix Nobel de chimie en 1939 ?
24. En quelle année a été décerné le premier prix Nobel d'économie ?
25. Combien de prix Nobel a reçu Marie Curie ?
26. Quels sont les prix lauréats, leur discipline et l'année de distinction de tous les prix Nobel contenant cohen dans leur nom (on ne fera pas de distinction de casse) ?
27. Combien y'a t'il eut de lauréats en Physique et en Chimie ?
28. Combien y'a t'il eut de lauréats de Médecine et de littérature en 2000 ?
29. Nombre de lauréats différents parmi les prix nobels de la paix ?

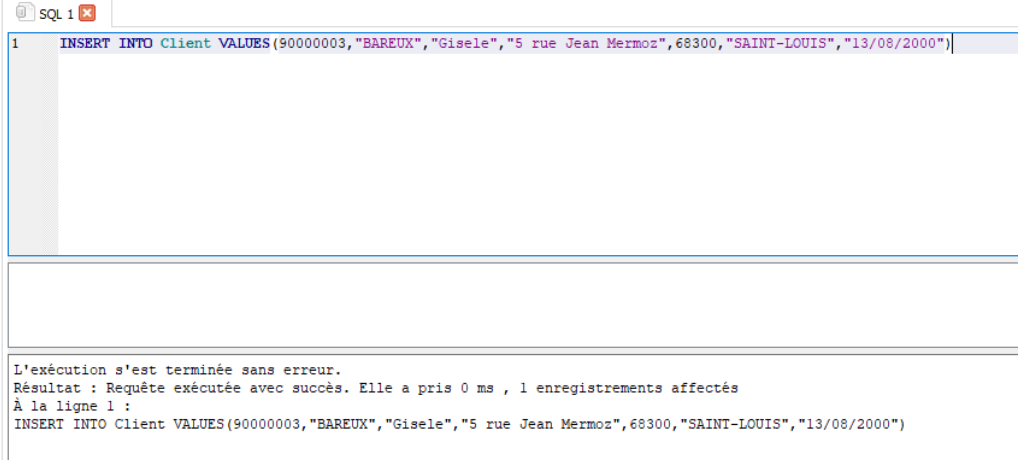
On peut aussi modifier la base de données grâce à des requêtes SQL :

Insertion de données :

L'instruction de base pour l'insertion de données dans une base est constituée du mot-clé **INSERT INTO** suivi du mot-clé **VALUES**.

- **INSERT INTO** permet de sélectionner la relation dans laquelle on insère les données.
- **VALUES** indique les valeurs qui doivent être insérées. Elles sont indiquées entre deux parenthèses. Elles doivent respecter l'ordre dans lequel les champs sont présents dans la table.

Exemple pour rajouter un client dans la table client :
INSERT INTO Client **VALUES**(90000003,"BAREUX","Gisele","5 rue Jean Mermoz",68300,"SAINT-LOUIS","13/08/2000")



On retrouve bien ce client dans la base :

	NumCarteFidelite	Nom	Prenom	Adresse	CodePostal	Ville	DateDeNaissance
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	CLIENT CAISSE	NULL	NULL	NULL	NULL	NULL
2	90000001	CIEL	Leïa	1 grand rue	39100	DOLE	28/04/2003
3	90000003	BAREUX	Gisele	5 rue Jean Mermoz	68300	SAINT-LOUIS	13/08/2000
4	90000023	MARCHEUR	Luc	15 rue des granges	25000	BESANCON	13/10/2000
5	90000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

Modification de données :

L'instruction de base pour la modification de données dans une base est constituée du mot-clé **UPDATE**, suivi du mot-clé **SET**.

- **UPDATE** permet de sélectionner la relation dans laquelle on insère les données.
- **SET** indique les attributs qui doivent être modifiés, et les valeurs correspondantes.

La syntaxe est : **UPDATE** table **SET** champ = valeur **WHERE** condition

Exemple pour modifier le prénom du client BAREUX dans la table client :
UPDATE Client **SET** Prenom="Toto" **WHERE** upper(Nom)="BAREUX"

SQL 1

```
1   UPDATE Client SET Prenom="Toto" WHERE upper(Nom)="BAREUX"
```

```
L'exécution s'est terminée sans erreur.
Résultat : Requête exécutée avec succès. Elle a pris 1 ms , 1 enregistrements affectés
À la ligne 1 :
UPDATE Client SET Prenom="Toto" WHERE upper(Nom)="BAREUX"
```

On retrouve bien le prénom modifié.

	NumCarteFidelite	Nom	Prenom	Adresse	CodePostal	Ville	DateDeNaissance
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	CLIENT CAISSE	NULL	NULL	NULL	NULL	NULL
2	90000001	CIEL	Leïa	1 grand rue	39100	DOLE	28/04/2003
3	90000003	BAREUX	Toto	5 rue Jean Mermoz	68300	SAINT-LOUIS	13/08/2000
4	90000023	MARCHEUR	Luc	15 rue des granges	25000	BESANCON	13/10/2000
5	90000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

Suppression de données :

L'instruction de base pour l'insertion de données dans une base est constituée du mot-clé **DELETE FROM.**

Syntaxe **: DELETE FROM** table **WHERE** condition

Exemple pour supprimons le client BAREUX dans la table client :
DELETE FROM Client **WHERE** upper(Nom)="BAREUX"

SQL 1

```
1   DELETE FROM Client WHERE upper(Nom)="BAREUX"
```

```
L'exécution s'est terminée sans erreur.
Résultat : Requête exécutée avec succès. Elle a pris 0 ms , 1 enregistrements affectés
À la ligne 1 :
DELETE FROM Client WHERE upper(Nom)="BAREUX"
```

Le client Bareux a bien été supprimé

Structure de la Base de Données | Parcourir les données | Éditer les Pragmas | Exécuter le SQL

Table : Client — Filtrer dans n'importe quelle colon

	NumCarteFidelite	Nom	Prenom	Adresse	CodePostal	Ville	DateDeNaissance
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	CLIENT CAISSE	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*
2	90000001	CIEL	Leïa	1 grand rue	39100	DOLE	28/04/2003
3	90000023	MARCHEUR	Luc	15 rue des granges	25000	BESANCON	13/10/2000
4	90000142	SOMBRE	Hector	8 rue de l'hôpital	25300	PONTARLIER	16/02/1998

En utilisant l'onglet Exécuter le SQL, indiquez le code SQL permettant de répondre aux questions suivantes :

30. En 2019, Esther Duflo a reçu le prix Nobel d'économie. Écrivez la requête permettant d'insérer cet enregistrement.
31. Quelle requête permet de modifier l'enregistrement précédent pour accoler le nom d'époux (Banerjee) après celui de Duflo ?
32. De nombreuses pétitions circulent pour retirer le prix Nobel à Aung San Suu Kyi. Quelle requête permettrait cela ?

2. Base de données cinema:

Télécharger la base de données cinema.db et l'ouvrir avec SQlite pour répondre aux questions suivantes :

1. Combien de tables possède la base de données ?
2. Combien de champs possède la table Artiste ?
3. Quelle est sa clé primaire ?
4. Combien de champs possède la table Film ?
5. Quelle est sa clé primaire ?
6. Quelle est sa clé étrangère ?
7. Effectuez un clic-droit sur la table Film, puis choisissez Modifier la table. En inspectant le code, indiquez quelle est la référence de la clé étrangère.
8. En procédant de la même façon pour toutes les tables, représentez le schéma relationnel sous forme graphique, en faisant bien figurer les relations entre les clés primaires et clés étrangères.

En utilisant l'ongle Exécuter le SQL, indiquez le code SQL permettant d'afficher (faire une capture de la requête et du résultat) :

9. Les titres des films triés par ordre alphabétique.
10. Les prénoms, noms et année de naissance des artistes nés avant 1950.
11. Les cinémas qui sont situés dans le 12e arrondissement.
12. Les artistes dont le nom commence par la lettre H (on utilisera la commande LIKE).
13. Les artistes dont on ignore la date de naissance (cela signifie que la valeur n'existe pas).
14. Le nombre de fois où Bruce Willis a joué le rôle de McLane (on utilisera la commande UPPER()) pour s'affranchir de la casse).

On rappelle qu'une jointure consiste à rapprocher les clé primaire et étrangère de deux relations.

15. Quel est le nom et le prénom de l'acteur qui a joué Tarzan (pensez à la commande UPPER().
16. Quelle est l'année de naissance du réalisateur de Reservoir Dogs ?
17. Quels sont les titres des films dans lesquels a joué Woody Allen. Donnez aussi le rôle joué.
18. Quels films peut-on voir au cinéma Rex ? (Attention aux doublons)
19. Quels films peut-on voir à 15h dans un cinéma parisien ?
20. Quels sont les cinémas (nom, adresse et arrondissement) qui diffusent des films le matin.
21. Quels films peut-on voir dans un cinéma du 12 e arrondissement ? On donnera le titre du film, le cinéma dans lequel il est joué et son adresse.
22. Quels sont les nom et prénom des acteurs qui ont joué dans le film Vertigo.
23. Quel réalisateur a tourné dans ses propres films ? Donnez le nom, le rôle et le titre des films.

24.Où peut-on voir le film Pulp Fiction ? On donnera le nom, l'adresse du cinéma et numéro de la séance.
25.Où peut-on voir un film avec Clint Eastwood ? On donnera le titre du film, le nom et l'adresse du cinéma, ainsi que l'heure de début du film.
26.Quels sont les cinémas (nom, adresse et arrondissement) ayant une séance le matin d'un film avec Bruce Willis ?
27.On souhaite insérer un nouvel artiste. Il s'agit de Ridley Scott, né en 1937. Indiquez la requête SQL permettant de réaliser cette opération.
28.Ridely Scott est le réalisateur du film Blade Runner, sorti en 1982. Indiquez la requête SQL permettant de saisir ce nouvel enregistrement.
29.Les acteurs principaux sont Harisson Ford (né en 1942) dans le rôle de Rick Deckard et Rutger Hauer (né en 1944) dans le rôle de Roy Batty. Indiquez la requête SQL permettant de saisir ces nouveaux enregistrements. (pour chaque enregistrement il faut finir la ligne par un ;)
30. Mark Hamill (né en 1951), Harisson Ford (né en 1942) et Carrie Fisher (née en 1956) sont les acteurs principaux du film L'empire contre attaque réalisé en 1980 par Irvin Kershner (né en 1923). Indiquez les requêtes SQL permettant de saisir ces enregistrements. Indiquez l'ordre dans lequel il faut remplir les relations.

V) Les bases de données et python :

Pour interagir avec une base de données au format SQLite3 depuis Python, il est nécessaire d'utiliser le module sqlite3.

```
import sqlite3
```

Pour se connecter à la base de données (qui doit être dans le même répertoire que le script python), il faut utiliser l'instruction suivante :

```
connexion = sqlite3.connect('Ludotheque.db')#connexion à la base
```

A la fin du script, il faudra se déconnecter de la base à l'aide de l'instruction :

```
connexion.close()#deconnexion de la base
```

Les bases de données sont des systèmes transactionnels. Cela signifie qu'un programme qui interagit avec une base de données fait ses modifications en mémoire, jusqu'à ce que celles-ci soient validées. Lors de cette validation, si les modifications sont toujours possibles, elles sont toutes transférées dans la base de données sur le disque. Sinon, aucune n'est effectuée. C'est cette validation par paquet qui constitue une transaction.
En Python, les transactions sont validées en appelant la méthode **commit()** sur l'objet connexion.
Chaque commit termine la transaction précédente, puis en commence une autre. La première transaction est initiée lors de la connexion à la base de données.
Bien sûr, si aucune modification n'a été apportée à la base, la phase de validation n'est pas nécessaire.
Pour valider les modifications sur une base de données, il faut utiliser l'instruction :

```
connexion.commit()#Validation de la modification de la base
```

Pour exécuter une requête, il faut la passer en paramètre de la méthode **execute()** de l'objet connexion.
Il faut au préalable avoir activé la vérification des clés étrangères en exécutant la requête **PRAGMA foreign_keys = ON**.

```
connexion.execute('PRAGMA foreign_keys = ON')#Activation des clés étrangères
```

Création de la table Editeur dans la base de données Ludothèque :

```
#Création de la table editeur
connexion.execute('CREATE TABLE Editeur (idEditeur int primary key not null,nomEditeur text not null,nationaliteEditeur text)')
```

Valider celle-ci avec l'instruction :

```
connexion.commit()#Validation de la modification de la base
```

Si vous ouvrez le fichier créer avec SQlite on constate bien que la table a été créée.

Nom	Type	Schéma
Tables (1)		
Editeur		CREATE TABLE Editeur (idEditeur int primary key not null,nomEditeur text not null,nationaliteEditeur text)
idEditeur	int	"idEditeur" int NOT NULL
nomEditeur	text	"nomEditeur" text NOT NULL
nationaliteEditeur	text	"nationaliteEditeur" text
Index (0)		
Vues (0)		
Déclencheurs (0)		

Ajout d'un enregistrement : ATTENTION Pour exécuter à nouveau le script, il faut mettre en commentaire la création de la table car celle-ci a été créée lors de la dernière exécution.

```
#Création de la table editeur
#connexion.execute('CREATE TABLE Editeur (idEditeur int primary key not null,n
#Enregistrement d'un éditeur PUISSANCE4 de nationalié Française
connexion.execute('INSERT INTO Editeur VALUES(1,"PUISSANCE4", "Française")')
```

Table : Editeur

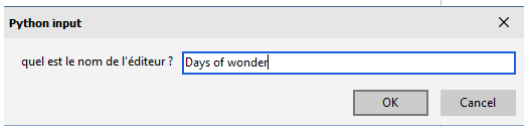
	idEditeur	nomEditeur	nationaliteEditeur
	Filtre	Filtre	Filtre
1	1	PUISSANCE4	Française

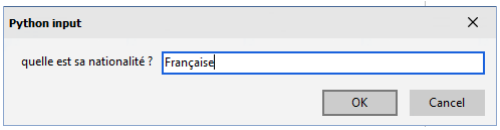
On pourrait aussi rajouter un enregistrement grâce à des instructions input à mettre avant l'ouverture de la base de données.

```
# Créé par gbareux, le 24/10/2020 en Python 3.7
import sqlite3
nom_jeu = input("quel est le nom de l'éditeur ? ")
nationalite_editeur = input("quelle est sa nationalité ? ")
connexion = sqlite3.connect('Ludotheque.db')#connexion à la base
connexion.execute('PRAGMA foreign_keys = ON')#Activation des clés étrangères
#Création de la table editeur
#connexion.execute('CREATE TABLE Editeur (idEditeur int primary key not null,nomEditeur text
#Enregistrement d'un éditeur PUISSANCE4 de nationalié Française
#connexion.execute('INSERT INTO Editeur VALUES(1,"PUISSANCE4", "Française")')

#Ajout des valeurs saisies par l'utilisateur
connexion.execute('INSERT INTO Editeur VALUES (?, ?, ?)', (2, nom_jeu, nationalite_editeur))
```

A l'exécution :






```
*** Console de processus distant Réinitialisée ***
quel est le nom de l'éditeur ? Days of Wonder
quelle est sa nationalité ? Française
```

	idEditeur	nomEditeur	nationaliteEditeur
	Filtre	Filtre	Filtre
1	1	PUISSANCE4	Française
2	2	Days of Wonder	Française

La méthode **execute()** présentée précédemment renvoie un objet **cursor** qui est itérable. On peut donc aisément parcourir les résultats d'une requête dans un boucle for, en itérant sur l'objet **cursor**.

```
#connexion.execute('INSERT INTO Editeur VALUES (?, ?, ?)', (2, nom_jeu, nationalite_editeur))
#affichage du contenu
curseur = connexion.execute("SELECT * FROM Editeur")
for tuple in curseur:
    print(tuple)
```

Pour récupérer les valeurs pour les différents champs de chaque tuple, il est préférable de les convertir en liste.

```
curseur = connexion.execute("SELECT * FROM Editeur")
#for tuple in curseur:
#    print(tuple)
for tuple in curseur:
    donnee = list(tuple)
    print("L'éditeur est :", donnee[1],"et sa nationalité est :",donnee[2])
```

Exercice :

Télécharger la base de données SpeBac.db et l'ouvrir avec SQlite pour répondre aux questions suivantes :

1. Combien de tables possède la base de données ?
2. Combien de champs possède la table Eleve ?
3. Quelle est sa clé primaire ?
4. Combien de champs possède la table Professeur ?
5. Quelle est sa clé primaire ?
6. En procédant de la même façon pour toutes les tables, représentez le schéma relationnel sous forme graphique, en faisant bien figurer les relations entre les clés primaires et clés étrangères.

En utilisant python et à l'aide des requêtes SQL sur la base SpeBac.db :

7. Afficher le nom, prenom de chaque élève suivi de son identifiant et de son sexe.

On attend :

```
*** Console de processus distant Réinitialisée ***
Maurice McVittie a pour identifiant idEleve : 1 et pour sexe : M
Anthony Cromly a pour identifiant idEleve : 2 et pour sexe : M
Maelys Cortese a pour identifiant idEleve : 3 et pour sexe : F
Marine Willoughey a pour identifiant idEleve : 4 et pour sexe : F
Didier Driscoll a pour identifiant idEleve : 5 et pour sexe : M
Lian Souza a pour identifiant idEleve : 6 et pour sexe : M
Yahn Basford a pour identifiant idEleve : 7 et pour sexe : M
Norbert Tuley a pour identifiant idEleve : 8 et pour sexe : M
Marine Berth a pour identifiant idEleve : 9 et pour sexe : F
Cecile De Blasio a pour identifiant idEleve : 10 et pour sexe : F
Aline Muller a pour identifiant idEleve : 11 et pour sexe : F
Jerome Dupont a pour identifiant idEleve : 12 et pour sexe : M
>>>
```

8. Afficher le nom, prenom de chaque prof suivi de son identifiant

On attend :

```
Steve Mizen a pour identifiant idProf : 1
Christophe Twining a pour identifiant idProf : 2
Milene Denyukhin a pour identifiant idProf : 3
Mathis Dable a pour identifiant idProf : 4
Charles Dupont a pour identifiant idProf : 5
Pierre Durand a pour identifiant idProf : 6
```

9. Mise à jour de la table Eleve : un élève (3) change de spécialité il passe de la spécialité math à physique.

On attend :

```
Spe1=Math, Spé2=Physique,Spé3=NSI et Spé4=SI
idEleve : 3 a choisi la spé: 1
idEleve : 3 a choisi la spé: 4
idEleve : 3 a choisi la spé: 3
Après changement
idEleve : 3 a choisi la spé: 2
idEleve : 3 a choisi la spé: 4
idEleve : 3 a choisi la spé: 3
```

10. Compter le nombre d'élèves ayant choisi la spécialité NSI.

On attend :

```
**********************************************************
Il y a  9  élèves qui ont choisi l'option NSI
```

11.Afficher Nom et prénom des filles ayant choisi la spécialité NSI, il faudra utiliser une jointure.

On attend :

```
**********************************************************
Cortese Maelys a choisi la spé NSI
Willoughey Marine a choisi la spé NSI
De Blasio Cecile a choisi la spé NSI
Muller Aline a choisi la spé NSI
```

12.Afficher le nombre de filles ayant choisi la spécialité NSI.

On attend :

```
**********************************************************
Il y a  4 filles qui ont choisi la spé NSI
```

VI) Bibliographie

https://fxjollois.github.io/cours-sql/

https://developpement-informatique.com/article/45/exercices-corriges-de-langage-sql

https://adrien.poupa.fr/efrei/Bases%20de%20donn%C3%A9es/TP1_%20SQL%20Cinema.pdf