

LES BASES DE DONNEES

1. INTRODUCTION

En classes de seconde et de première, nous avons travaillé avec des données structurées en les stockant dans des fichiers au format :

- CSV : Comma Separated Values (séparateur est « , », en France « ; »),

Fichier au format .csv	Représentation tabulaire		
Sexe,Prénom,Année de naissance M,Alphonse,1932 F,Béatrice,1964 F,Charlotte,1988	Sexe	Prénom	Année de naissance
	M	Alphonse	1932
	F	Béatrice	1964
	F	Charlotte	1988

- JSON : JavaScript Object Notation (séparateurs sont « { } »),

```
[
  {
    "Sexe": "M ",
    "Prénom": "Alphonse ",
    "Année de naissance": 1932
  },
  {
    "Sexe": "F ",
    "Prénom": "Béatrice ",
    "Année de naissance": 1964
  },
  {
    "Sexe": "F ",
    "Prénom": "Charlotte ",
    "Année de naissance": 1988
  }
]
```

- XML : eXtensible Markup Language (séparateurs sont les balises « <> » et « </> »).

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <eleve>
    <Sexe>M </Sexe>
    <Prénom>Alphonse </Prénom>
    <Année de naissance>1932</Année de naissance>
  </eleve>
  <eleve>
    <Sexe>F </Sexe>
    <Prénom>Béatrice </Prénom>
    <Année de naissance>1964</Année de naissance>
  </eleve>
  <eleve>
    <Sexe>F </Sexe>
    <Prénom>Charlotte </Prénom>
    <Année de naissance>1988</Année de naissance>
  </eleve>
</root>
```

Même si cette méthode de stockage de l'information peut s'avérer pratique dans certains cas, il est souvent souhaitable d'utiliser une base de données (BD) pour stocker des données.

1.1. UN PEU D'HISTOIRE

Voici une vidéo d'introduction :

<https://www.youtube.com/watch?v=iu8z5QtDQhY&t=120s>

L'invention du disque dur en 1956, mémoire de masse de grande capacité, a permis d'utiliser les ordinateurs pour collecter, classer et stocker de grandes quantités d'informations de façon plus souple et plus performante que le support antérieur : la bande magnétique.

Le terme **database** (base de données (BD)) est apparu en 1964 pour désigner une collection d'informations partagées par différents utilisateurs d'un système d'informations militaire.

Avec l'apparition du Web, la quantité de données à stocker a littéralement explosé.

Actuellement, la plupart des sites Internet (du petit site personnel au grand site d'e-commerce) utilisent au moins une base de données. Les bases de données (BD) jouent un rôle fondamental dans notre monde devenu numérique où il est extrêmement facile de dupliquer l'information.

1.2. GESTION DES BASES DE DONNEES

Pour gérer une base de données, il faut utiliser une suite de programmes appelés **système de gestion de bases de données (SGBD)**. Des systèmes de gestion de bases de données (SGBD) de très grande taille (de l'ordre du pétaoctet 10^{15} octet) sont au centre de nombreux dispositifs de collecte, de stockage et de production d'informations. Les SGBD permettent :

- de gérer la lecture, l'écriture ou la modification des informations de la base,
- de gérer les autorisations d'accès à la base (un utilisateur peut être autorisé à lire et écrire dans la base de données alors qu'un autre utilisateur peut n'avoir que la possibilité d'y lire les informations),
- de gérer les accès concurrents à la base (plusieurs personnes peuvent avoir besoin d'accéder en même temps aux informations contenues dans une base données, ce qui peut parfois poser problème, notamment lorsqu'il s'agit de modifier la même donnée au même moment),
- d'assurer la maintenance de la base (les fichiers d'une base de données sont stockés sur des disques durs dans des ordinateurs. Pour palier aux éventuelles pannes matérielles, une même base de données est dupliquée sur plusieurs ordinateurs. Toutes les copies de la base doivent être parfaitement synchronisées et ceci le plus rapidement possible).

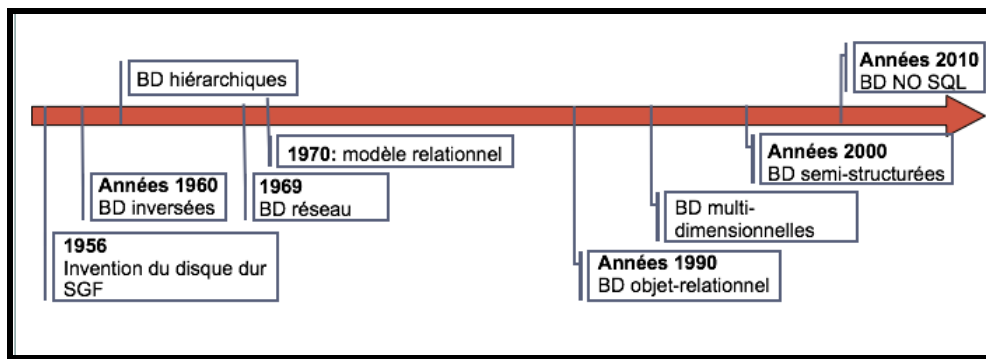
yotta [Y]	10^{24}	= 1 000 000 000 000 000 000 000 000
zetta [Z]	10^{21}	= 1 000 000 000 000 000 000 000 000
exa [E]	10^{18}	= 1 000 000 000 000 000 000 000 000
peta [P]	10^{15}	= 1 000 000 000 000 000 000 000 000
tera [T]	10^{12}	= 1 000 000 000 000 000 000 000 000
giga [G]	10^9	= 1 000 000 000 000 000 000 000 000
mega [M]	10^6	= 1 000 000 000 000 000 000 000 000
kilo [k]	10^3	= 1 000 000 000 000 000 000 000 000
hecto [h]	10^2	= 100 000 000 000 000 000 000 000
deca [da]	10^1	= 10 000 000 000 000 000 000 000

1.3. DIFFERENTES FAMILLES DE BASES DE DONNEES

Il existe différentes familles de bases de données :

- les bases de données **hiérarchiques** (années 60) : les enregistrements de la base sont liés selon une structure arborescente peu adapté pour décrire des structures complexes, répondre aux besoins réels et suivre l'évolution des systèmes d'information (exemple : base de registre de Microsoft Windows).
- les bases de données **relationnelles** (1970) : l'information est organisée dans des tableaux à deux dimensions reliés entre eux par des **clefs**. Ce sont les plus couramment utilisées.
- les bases de données **objet** (années 90) : les informations sont groupées sous formes de collections d'objets (conteneur logique qui englobe des informations et des traitements)

- les bases de données **NoSQL** (Not only SQL, 2009, SQL : Structured Query Language) langage informatique utilisé pour exploiter des bases de données) : enregistrement en structures flexibles, selon un paradigme non relationnel, favorisant l'interrogation des données adapté aux grosses infrastructures matérielles constituées de nombreux serveurs organisés en clusters (grappes), essentiellement pour des applications Web (cloud, réseaux sociaux, ...) (exemples : HBase et BigTable (Google), Cassandra (Facebook, Twitter, Cisco, eBay, Netflix, ...), NoSQL Database (Oracle), DynamoDB (Amazon), ...
- ...



1.4. LANGAGES DES BASES DE DONNEES

Les langues de base de données permettent d'effectuer de multiples tâches :

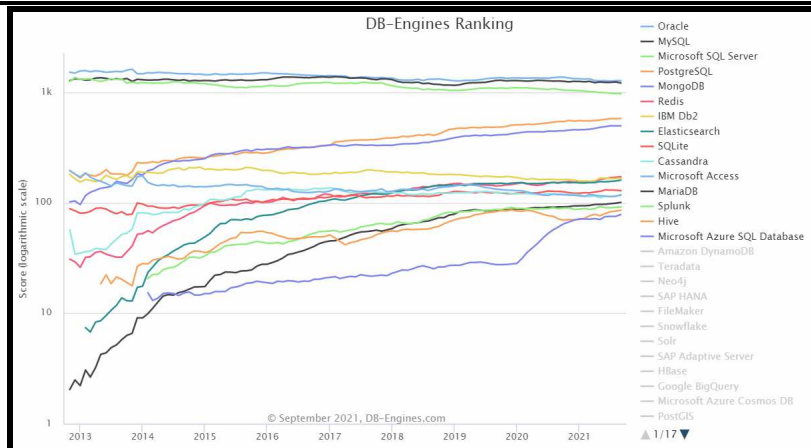
- contrôle de l'accès aux données ;
- définition des types de données (création, modification ou suppression de tables et les relations entre elles) ;
- insertion, mise à jour ou suppression d'occurrences de données ;
- recherche d'informations ;
- ...

Les langages de base de données sont spécifiques à un modèle de données particulier.

Par exemple, l'accès aux données d'une **base de données relationnelle** s'effectue grâce à des **requêtes** d'interrogation et de mise à jour qui peuvent par exemple être rédigées dans le **langage SQL** (Structured Query Language).

Les traitements peuvent associer l'utilisation du langage SQL et d'un langage de programmation.

1.5. PRINCIPAUX SYSTEMES DE GESTION DE BASES DE DONNEES (SGBD)



https://db-engines.com/en/ranking_trend

2. BASES DE DONNEES RELATIONNELLES

Les **bases de données relationnelles** ont été mises au point en 1970 par Edgar Franck Codd, informaticien britannique (1923-2003). Ces bases de données sont basées sur la théorie mathématique des ensembles.

La notion de relation est au coeur des bases de données relationnelles. Une relation peut être considérée comme un tableau ou une table à 2 dimensions contenant :

- un ensemble de n-uplets ou p-uplets ou de tuples (les lignes ou les **enregistrements** de la table) dont les **entrées** (des valeurs qui peuvent être de type nombre, texte, booléen ou NULL) appartiennent à un **domaine**.
- des **attributs** (les colonnes de la table), dont les noms peuvent servir de **clefs**.

Fruits				← Relation
Références	Noms	En_stock	Prix_en_€_au_kg	← Attributs
551 234	Banane	True	1,35	} Enregistrements
551 235	Fraise	True	6,25	
551 236	Cerise	True	5,50	
551 237	Pomme	True	1,70	
551 238	Poire	False	2,40	
551 239	Raisin	True	3,10	
...	

↑ Entrées de l'attribut « Références » ↑ Entrées de l'attribut « Noms » ↑ Entrées de l'attribut « En stock » ↑ Entrées de l'attribut « Prix en € au kg » Entrées

L'**ordre** (ou **degré**) d'une relation est le nombre d'attributs de cette relation.

Donnez l'ordre de la relation « Fruits » ci-dessus : _____

Dans une même base de données, différentes relations peuvent avoir des attributs de même nom. Pour désigner l'attribut « Attribut_1 » de la relation « Relation_1 » et le distinguer des autres attributs de même noms, on le notera : « Relation_1.Attribut_1 ».

En tenant compte du paragraphe précédent, nommez l'attribut « Noms » de la relation « Fruits » ci-dessus : _____

2.1. DOMAINES DES BASES DE DONNEES RELATIONNELLES

Le **domaine** d'un attribut est un ensemble, fini ou infini, des valeurs admissibles.

Le domaine de l'attribut « Références » de la relation « Fruits » ci-dessus est :
551 234 ; 551 235 ; 551 236 ; 551 237 ; 551 238 ; 551 239

Donnez le domaine de l'attribut « Noms » de la relation « Fruits » ci-dessus :

Donnez le domaine de l'attribut « En_stock » de la relation « Fruits » ci-dessus :

Le domaine d'un attribut est renseigné au moment de la création de la relation. Le SGBD (système de gestion de base de données) s'assure par la suite qu'un enregistrement ajouté à une

relation respecte bien le domaine de l'attribut correspondant. Par exemple, dans l'attribut « En_stock » de la relation « Fruits » ci-dessus, le domaine sera défini comme pouvant accueillir un booléen, son domaine sera donc { False ; True }, le SGBD signalera une erreur et ne tolèrera donc pas l'écriture d'un nouvel enregistrements contenant « Vrai » ou « Faux » dans l'attribut « En_stock ».

2.2. RELATIONS ET CLES DES BASES DE DONNEES RELATIONNELLES

2.2.1. CLES PRIMAIRES

Une relation ne peut pas contenir enregistrements identiques. La situation ci-dessous n'est pas autorisée par le SGBD (système de gestion de bases de données) :

Identifiants	Titres	Auteurs	Années_de_publication
1	1984	Orwell	1949
2	Dune	Herbert	1965
2	Dune	Herbert	1965
3	Fondation	Asimov	1951

Afin d'être sûr de respecter cette contrainte des enregistrements uniques, on définit la notion de « clé primaire ». Une clé primaire est un attribut dont la valeur permet d'identifier de manière unique un enregistrement de la relation. Si un attribut est considéré comme clé primaire, on ne doit pas trouver dans toute la relation deux fois la même valeur pour cet attribut.

Soit la relation « Livres » ci-dessous :

Identifiants	Titres	Auteurs	Années_de_publication
1	1984	Orwell	1949
2	Dune	Herbert	1965
3	Fondation	Asimov	1951
4	Le meilleur des mondes	Huxley	1931
5	Fahrenheit 451	Bradbury	1953
6	Ubik	K.Dick	1969
7	Chroniques martiennes	Bradbury	1950
8	La nuit des temps	Barjavel	1968
9	Blade Runner	K.Dick	1968
10	Les Robots	Asimov	1950
11	La Planète des singes	Boulle	1963
12	Ravage	Barjavel	1943
13	Le Maître du Haut Château	K.Dick	1962
14	Le monde des Â	Van Vogt	1945
15	La Fin de l'éternité	Asimov	1955
16	De la Terre à la Lune	Verne	1865

L'attribut « années_de_publication » peut-il jouer le rôle de clé primaire ?

L'attribut « Auteurs » peut-il jouer le rôle de clé primaire ?

L'attribut « Titres » peut-il jouer le rôle de clé primaire ?

En 2013, l'Américaine Jill McCorkle et l'Anglaise Kate Atkison publiaient avec seulement six jours d'écart un livre intitulé « Life After Life ».

Pour l'attribut « Identifiants » : à chaque fois qu'un roman est ajouté à la relation, son attribut « Identifiants » correspond à l'incrémentation de un de celui du roman précédemment. Il est donc impossible d'avoir deux romans avec le même attribut « Identifiants ». Ajouter un attribut « Identifiants » afin qu'il puisse jouer le rôle de clé primaire est une pratique courante (mais non obligatoire) dans les bases de données relationnelles. Il aurait été possible de ne pas utiliser d'attribut « Identifiants », car chaque livre édité possède un numéro qui lui est propre : ISBN (International Standard Book Number) qui aurait pu jouer le rôle de clé primaire.

À noter qu'une clé primaire peut être constituée de plusieurs attributs, par exemple le couple « Auteurs » et « Titres » pourrait jouer le rôle de clé primaire.



2.2.2. CLES ETRANGERES ET MODELE RELATIONEL

Soit la relation « Livres » enrichies (trois attributs ajoutés « Prenoms auteurs », « Années naissance auteurs » et « Langues auteurs ») ci-dessous :

Identifiants	Titres	Auteurs	Prénoms_auteurs	Années_naissance_auteurs	Langues_auteurs	Années_de_publication
1	1984	Orwell	George	1903	anglais	1949
2	Dune	Herbert	Frank	1920	anglais	1965
3	Fondation	Asimov	Isaac	1920	anglais	1951
4	Le meilleur des mondes	Huxley	Aldous	1894	anglais	1931
5	Fahrenheit 451	Bradbury	Ray	1920	anglais	1953
6	Ubik	K.Dick	Philip	1928	anglais	1969
7	Chroniques martiennes	Bradbury	Ray	1920	anglais	1950
8	La nuit des temps	Barjavel	René	1911	français	1968
9	Blade Runner	K.Dick	Philip	1928	anglais	1968
10	Les Robots	Asimov	Isaac	1920	anglais	1950
11	La Planète des singes	Boulle	Pierre	1912	français	1963
12	Ravage	Barjavel	René	1911	français	1943
13	Le Maître du Haut Château	K.Dick	Philip	1928	anglais	1962
14	Le monde des Â	Van Vogt	Alfred Elton	1912	anglais	1945
15	La Fin de l'éternité	Asimov	Isaac	1920	anglais	1955
16	De la Terre à la Lune	Verne	Jules	1828	français	1865

Il y a des informations dupliquées, on retrouve trois fois « K.Dick, Philip, 1928, anglais », trois fois « Asimov, Isaac, 1920, anglais », deux fois « Bradbury, Ray, 1920, anglais » et deux fois « Barjavel, René, 1911, français ». Dans une base de données, il faut éviter autant que possible de dupliquer l'information (sauf à des fins de sauvegarde). S'il y a duplication inutile de données c'est que la structure n'est pas correcte et qu'il faut la modifier.

Les **relations** d'une base de données sont reliées (ou « connectées ») par certaines des valeurs qu'elles contiennent : chaque **enregistrement** d'une table contient un groupe d'informations relatives à un sujet et les différents sujets sont connexes.

Fruits			
Références	Noms	En_stock	Prix_en_€_au_kg
551 234	Banane	True	1,35
551 235	Fraise	True	6,25
551 236	Cerise	True	5,50
551 237	Pomme	True	1,70
551 238	Poire	False	2,40
551 239	Raisin	True	3,10

↑
Clé primaire
de la
relation
« Fruits »

Enregistrements en relation

Ventes			
Identifiants	Références_fruits	Dates	Quantités_en_kg
1	551 234	21/09/21	4
2	551 237	21/09/21	6
3	551 235	22/09/21	2
4	551 234	22/09/21	8
5	551 239	23/09/21	10

↑
Clé primaire
de la relation
« Ventes »

↑
Clé
étrangère qui
fait référence
à la clé
primaire de la
relation
« Fruits »

Par exemple : les relations « **Fruits** » et « **Ventes** » sont connectées puisque l'objectif de la base de données est de gérer les ventes de fruits.

Ces liens existants entre les informations sont stockés dans les champs (attributs) des enregistrements sous forme de **clés**.

Pour une relation donnée, une **clé** est un attribut ou un groupe d'attributs permettant d'identifier un unique enregistrement de la relation.

Pour qu'un attribut ou un groupe d'attributs forme une **clé**, il faut être sûr que deux enregistrements n'auront jamais des valeurs identiques pour cet attribut ou ces attributs : on parle d'**unicité** de la clé.

On parle de **clé candidate** si le groupe d'attributs de la clé est **minimal**, c'est-à-dire que si on retire un seul des attributs de ce groupe, l'unicité n'est plus vérifiée.

Toute relation doit avoir au moins une clé candidate et peut en avoir plusieurs.

La **clé primaire** est l'une des clés candidates de la relation, choisie pour être utilisée comme clé étrangère dans une autre relation.

La **clé étrangère** est l'attribut d'une relation dont les valeurs sont des références à une clé primaire d'une autre relation.

Pour préserver l'intégrité d'une base de données, le SGBD (système de gestion de bases de données) se charge de vérifier que toutes les valeurs d'une clé étrangère d'une relation correspondent bien à des valeurs présentes dans la clé primaire de l'autre relation.

Pour éviter les duplications, à partir de la relation « Livres » enrichies vue précédemment, nous allons créer les deux relations suivantes et un modèle relationnel :

Relation « Auteurs » :

Identifiants _auteurs	Noms	Prénoms	Années_naissance _auteurs	Langues_auteurs
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Asimov	Isaac	1920	anglais
4	Huxley	Aldous	1894	anglais
5	Bradbury	Ray	1920	anglais
6	K.Dick	Philip	1928	anglais
7	Barjavel	René	1911	français
8	Boulle	Pierre	1912	français
9	Van Vogt	Alfred Elton	1912	anglais
10	Verne	Jules	1828	français

Relation « Livres » :

Identifiants _livres	Titres	Identifications _auteurs	Années_de _publication
1	1984	1	1949
2	Dune	2	1965
3	Fondation	3	1951
4	Le meilleur des mondes	4	1931
5	Fahrenheit 451	5	1953
6	Ubik	6	1969
7	Chroniques martiennes	5	1950
8	La nuit des temps	7	1968
9	Blade Runner	6	1968
10	Les Robots	3	1950
11	La Planète des singes	8	1963
12	Ravage	7	1943
13	Le Maître du Haut Château	6	1962
14	Le monde des Â	9	1945
15	La Fin de l'éternité	3	1955
16	De la Terre à la Lune	10	1865

Nous avons une relation « Auteurs » et une relation « Livres » dans laquelle nous avons remplacé les attributs « Auteurs », « Prénoms_auteurs », « Années_naissance_auteurs » et « Langues_auteurs » par un seul attribut « Identifications_auteurs ». L'attribut « Identifications_auteurs » de la relation « Livres » permet de créer un lien avec l'attribut « Identifiants_auteurs » de la relation « Auteurs ». Ceci permet d'éviter la redondance d'informations dans les relations.

L'attribut « Identifiants_auteurs » de la relation « Auteurs » est la clé primaire de la relation « Auteurs ».

L'attribut « Identifiants_livres » de la relation « Livres » est la clé primaire de la relation « Livres ».

L'attribut « Identifications_auteurs » de la relation « Livres » est la clé étrangère de la relation « Livres ». L'attribut « Identifications_auteurs » de la relation « Livres » permet d'établir un lien entre la relation « Livres » et la relation « Auteurs ». L'attribut « Identifications_auteurs » de la relation « Livres » correspond à la clé primaire de la relation « Auteurs » donc c'est une clé étrangère.

2.2.3. SCHEMA RELATIONNEL

Un schéma relationnel est l'ensemble des relations présentes dans une base de données. Il contient les informations suivantes :

- Les noms des différentes relations
- pour chaque relation, la liste des attributs avec leur domaine respectif
- pour chaque relation, la clé primaire et éventuellement les clés étrangères

Les attributs soulignés sont des clés primaires, le « # » indique la présence d'un clé étrangère.

Exemple pour les relations « Auteurs » et « Livres » ci-dessus :

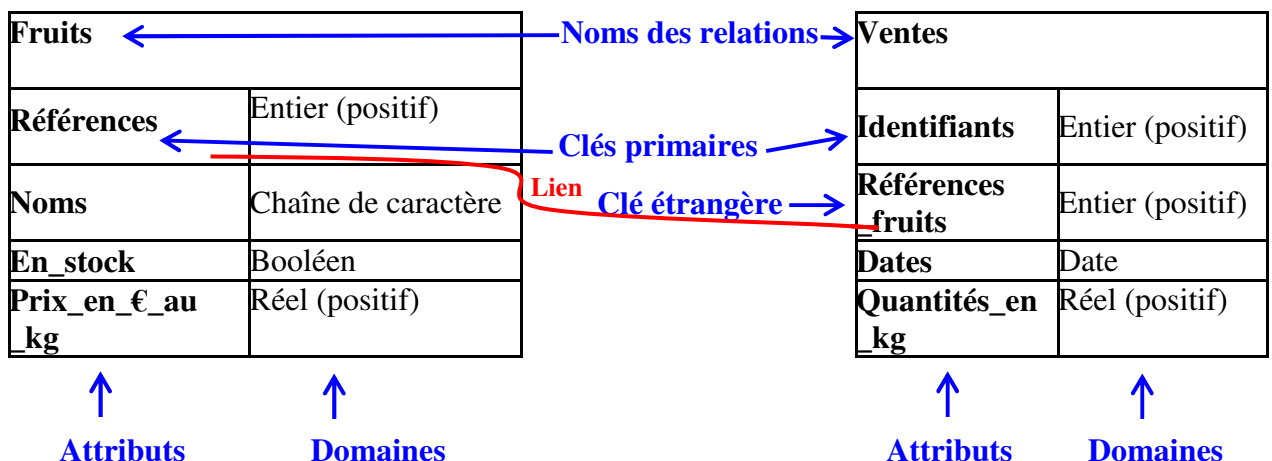
- Auteurs (Identifiants_auteurs, Noms, Prénoms, Années_naissance_auteurs, Langues_auteurs)
- Livres (Identifiants_livres, Titres, #Identifications_auteurs, Années_de_publication)

On peut également préciser les domaines des différents attributs :

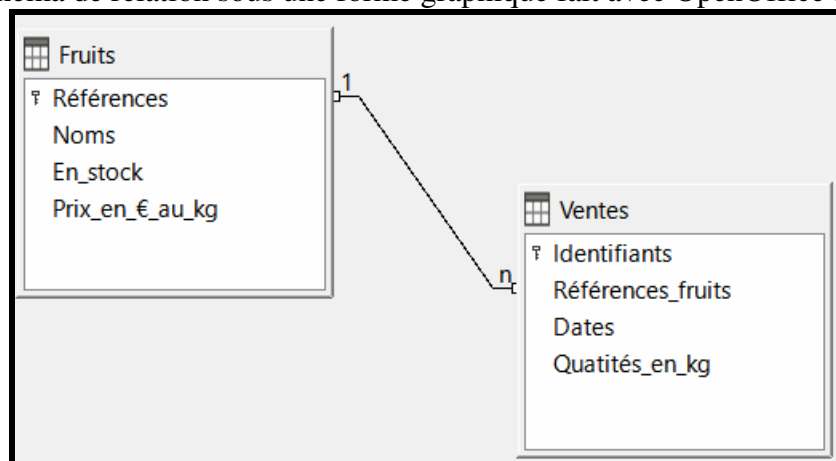
- Auteurs (Identifiants_auteurs : ENTIER, Noms : TEXTE, Prénoms : TEXTE, Années_naissance_auteurs : ENTIER, Langues_auteurs : TEXTE)
- Livres (Identifiants_livres : ENTIER, Titres : TEXTE, #Identifications_auteurs : ENTIER, Années_de_publication : ENTIER)

Un **schéma de relation** précise le nom de la relation ainsi que la liste des attributs avec leurs domaines.

On peut représenter un schéma de relation sous une forme graphique, par exemple à l'aide d'un diagramme :



Le même schéma de relation sous une forme graphique fait avec OpenOffice :



2.2.3.1.SCHEMA RELATIONEL EXEMPLE 1

En précisant les domaines des différents attributs, donnez le schéma relationnel des relations « Fruits » et « Ventes » ci-dessus page 7 :

2.2.3.2.SCHEMA RELATIONEL EXEMPLE 2

Soit une base de données dont le schéma est le suivant :

EMPLOYE (NumEmp, Nom, Prénom, Adresse, Téléphone, Qualification)

SERVICE (NomService, Responsable, Téléphone)

PROJET (NomProjet, DateDeb, DateFin, #NumEmp)

Représentez le schéma de relation sous une forme graphique.

Un employé peut il avoir plusieurs qualifications ?

Un employé peut il faire plusieurs projets en même temps ?

Une personne peut elle être responsable de plusieurs services ?

Un service peut il avoir plusieurs responsables ?

2.2.3.3. SCHEMA RELATIONEL EXEMPLE 3

Soit une base de dont le schéma est le suivant :

Pilote (PLNUM, PLNOM, PLPRENOM, VILLE, SALAIRE)

Avion (AVNUM, AVNOM, CAPACITE, LOCALISATION)

Vol (VOLNUM, #PLNUM, #AVNUM, VILLEDEP, VILLEARR, HEUREDEP, HEUREARR)

Représentez le schéma de relation sous une forme graphique.

2.2.3.4. SCHEMA RELATIONEL EXEMPLE 4

En partant de la relation **Films** ci-dessous, créez une relation **Réalisateurs** (id, nom, prenom et annee_naissance).

Modifiez la relation **Films** afin d'établir un lien avec la relation **Réalisateurs**. Précisez l'attribut qui jouera le rôle de clef étrangère.

Films :

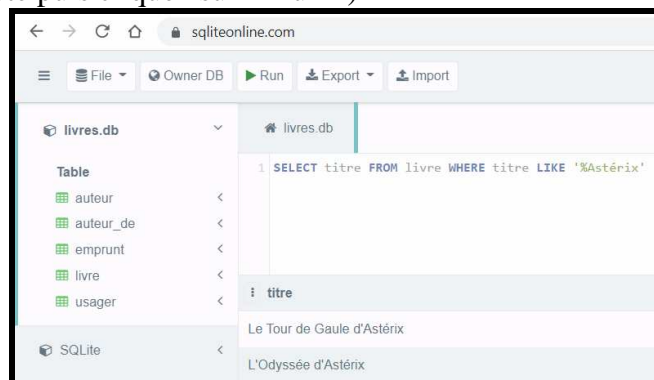
id	titre	realisateur	annee_sortie
1	Alien, le huitième passager	Scott	1979
2	Dune	Lynch	1985
3	2001 : l'odyssée de l'espace	Kubrick	1968
4	Blade Runner	Scott	1982

3. INTERROGER LES BASES DE DONNEES

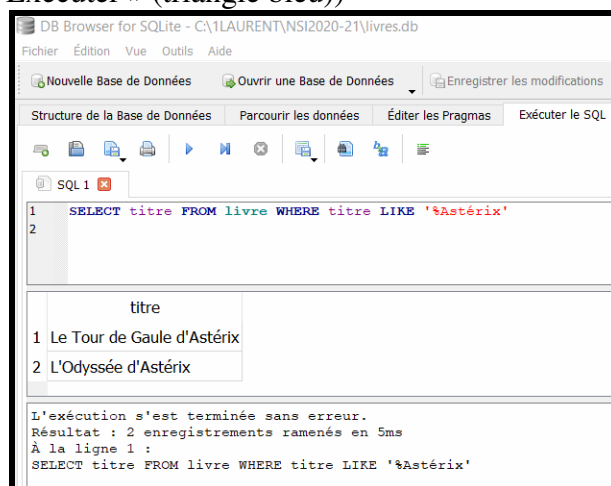
3.1. METHODES POUR INTERROGER LES BASES DE DONNEES

Il existe plusieurs méthodes pour interroger les bases de données :

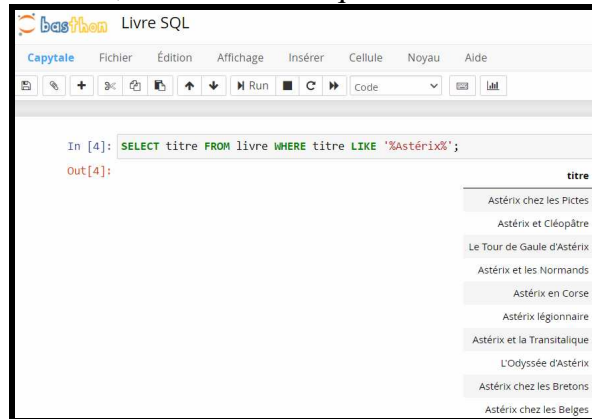
- En ligne : <https://sqliteonline.com/> (« File », « OpenDB », sélectionnez la base de données souhaitée (fichier avec extension .db précédemment téléchargé), écrivez votre requête puis cliquez sur « Run »)



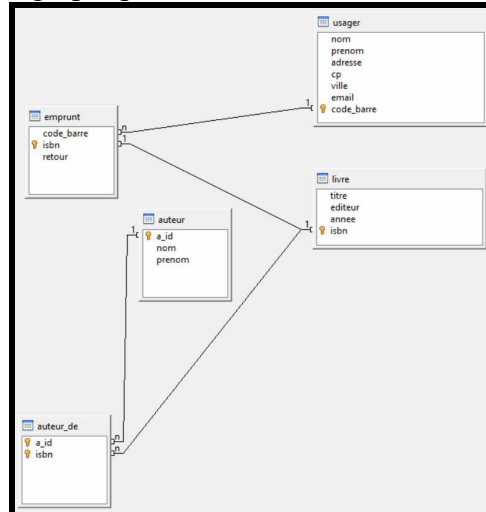
- Avec un logiciel : DB Browser for SQLite (Installez DB Browser for SQLite, téléchargeable à l'adresse <https://sqlitebrowser.org/>, (« Fichier », « Ouvrir une base de données », sélectionnez la base de données souhaitée (fichier avec extension .db précédemment téléchargé), « Exécuter le SQL », écrivez votre requête puis cliquez sur « Exécuter » (triangle bleu))



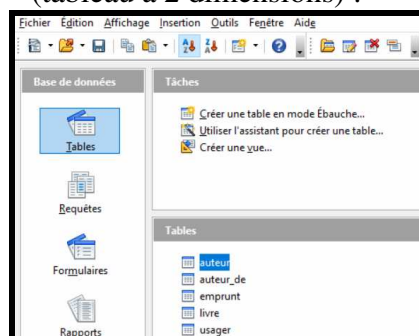
- Avec « Capytale » sur l'ENT : « Capytale », « Mes activités », « Gérer mes services », sélectionnez sur « activé » en face de « Activité SQL », « Enregistrer », « Mes activités », « SQL créer une activité », saisir un « Titre », cocher « Joindre des fichiers annexes au notebook », sélectionnez la base de données souhaitée (fichier avec extension .db précédemment téléchargé), « Enregistrer », afin de vérifier les fichiers annexes cliquez sur le logo contenant un trombone « liste des fichiers annexes », « attachés à l'activité » permet de vérifier que la base de données souhaitée est présente pour l'activité, écrivez votre requête dans une cellule puis cliquez sur « Run ».



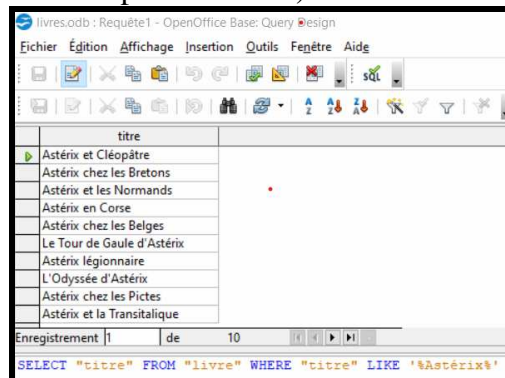
- Avec les logiciels : LibreOffice base ou OpenOffice base : « OpenOffice », « Base de données », « Ouvrir un fichier de base de données existant », « Ouvrir », sélectionnez la base de données souhaitée (fichier avec extension .odb précédemment téléchargé).
 - ✓ « Outils », « Relations » permet d'afficher le schéma relationnel sous forme graphique :



- ✓ « Tables » permet de créer, visualiser ou / et modifier les différentes relations (tableau à 2 dimensions) :

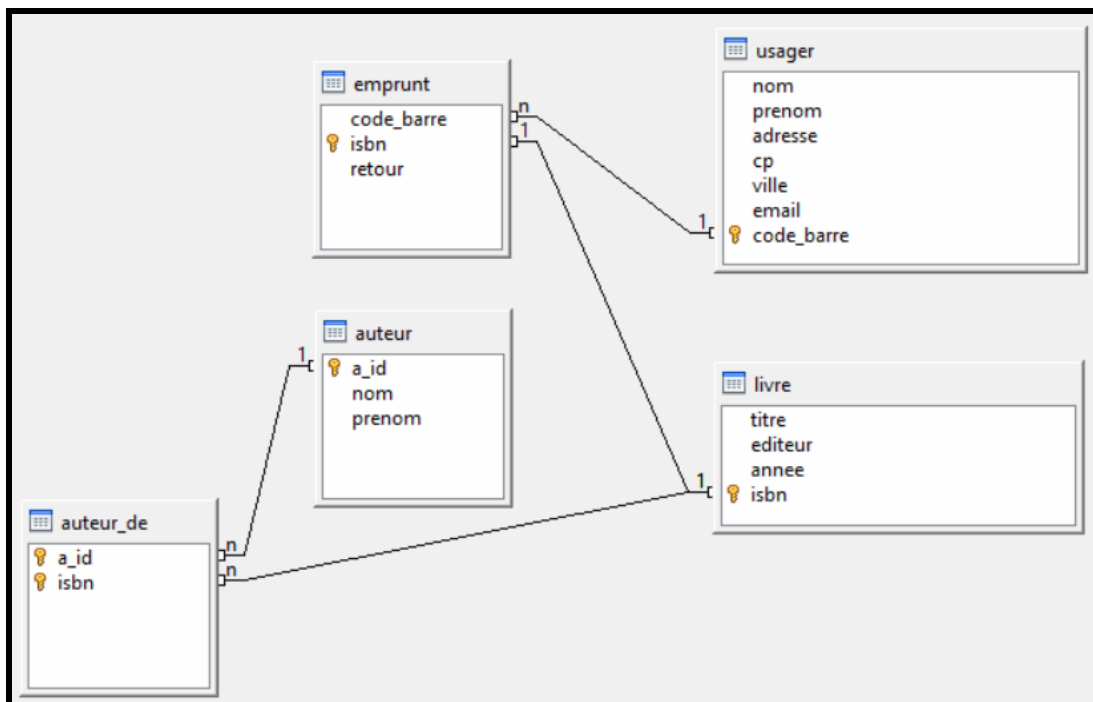


- ✓ « Requêtes » permet de créer (« Insertion », « Requête mode SQL ») , visualiser et / ou modifier et / ou exécuter les différentes requêtes (clic droit sur la requête, puis « Editer en mode SQL », puis « Edition » + « Exécuter la requête » ou « F5 » pour exécuter) :



3.2. INTERROGER LES BASES DE DONNEES

Les manipulations sont à faire en interrogeant la base de données « livres.db » pour l'une des 3 premières méthodes et la base de données « livres.odt » en utilisant LibreOffice base ou OpenOffice base. Ces bases de données sont identiques (uniquement différenciées par l'extension .db et .odt) et contiennent les relations (tables) selon le schéma relationnel sous forme graphique suivant :

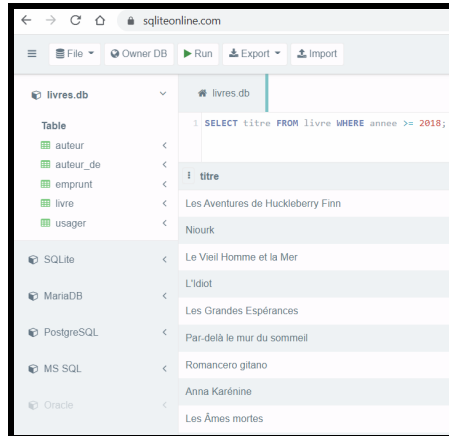


Relations	Clés primaires	Clés étrangères
usager	code_barre	
livre	isbn	
auteur	a_id	
emprunt	isbn	code_barre de la relation usager isbn de la relation livre
auteur_de	Couple a_id et isbn	a_id de la relation auteur isbn de la relation livre

3.2.1. SELECTION DE DONNEES

3.2.1.1. REQUETE BASIQUE AVEC « SELECT », « FROM » ET « WHERE »

- **Commande :**
SELECT titre FROM livre WHERE annee >= 2018;
- **Signification :**
Permet d'obtenir les titres de la relation (table) « livre » qui sont parus en ou après 2018.
- **Résultat :**

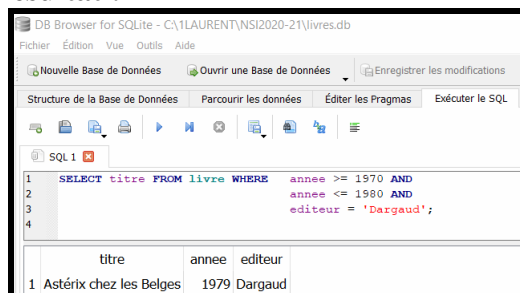


titre
Les Aventures de Huckleberry Finn
Niourk
Le Vieil Homme et la Mer
L'Idiot
Les Grandes Espérances
Par-delà le mur du sommeil
Romancero gitano
Anna Karénine
Les Âmes mortes

- **Exercice :**
Obtenir les titres de la relation (table) « livre » qui sont parus avant 1980.

3.2.1.2. REQUETE BASIQUE AVEC « AND »

- **Commande :**
SELECT titre FROM livre WHERE annee >= 1970 AND annee <= 1980 AND editeur = 'Dargaud';
- **Signification :**
Permet d'obtenir les titres de la relation (table) « livre » qui sont parus entre 1970 et 1980 chez l'éditeur Dargaud.
- **Résultat :**



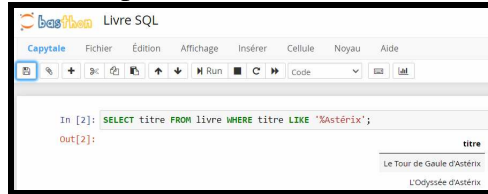
titre	annee	editeur
1 Astérix chez les Belges	1979	Dargaud

- **Exercice :**
Obtenir les titres et les années de parution de la relation (table) « livre » qui sont parus entre 1900 et 2021 chez l'éditeur Editions Gallimard.

3.2.1.3. REQUETE APPROCHEE AVEC « LIKE »

- **Commande :**
SELECT titre FROM livre WHERE titre LIKE '%Astérix%';
- **Signification :**
Permet d'obtenir les titres de la relation (table) « livre » dont le titre contient la chaîne de caractères « Astérix ». Le symbole « % » peut remplacer n'importe quelle chaîne de caractères avant ou après le mot recherché.

- **Résultat** (pour `SELECT titre FROM livre WHERE titre LIKE '%Astérix';`) :



- **Exercice :**
Obtenir les titres, les ISBN et les années de parution de la relation (table) « livre » dont le titre contient la chaîne de caractères « roi ».

3.2.1.4. REQUETE AVEC RENVOI DE PLUSIEURS COLONNES

- **Commande :**
`SELECT titre, annee, isbn FROM livre WHERE annee >= 2018;`
- **Signification :**
Permet d'obtenir les titres et les isbn de la relation (table) « livre » qui sont parus en ou après 2018.
- **Résultat :**

titre	annee	isbn
L'idiot	2019	978-1911572909
Le Vieil Homme et la Mer	2018	978-2072762093
Les Aventures de Huckleberry Finn	2020	978-2081509511
Les Grandes Espérances	2019	978-2322185801
Les Âmes mortes	2018	978-2371240001
Anna Karénine	2018	978-2371240087
Par-delà le mur du sommeil	2018	978-2824904269
Romancero gitano	2020	978-8832957402
Niourk	2018	978-1025100639

3.2.1.5. REQUETE AVEC RENVOI DE TOUTES LES COLONNES

- **Commande :**
`SELECT * FROM livre WHERE annee >= 2018;`
- **Signification :**
Permet d'obtenir toutes les colonnes de la relation (table) « livre » qui sont parus en ou après 2018 (* astérisque = joker wildcard).
- **Résultat :**

titre	editeur	annee	isbn
Les Aventures de Huckleberry Finn	Flammarion	2020	978-2081509511
Niourk	French Pulp éditions	2018	978-1025100639
Le Vieil Homme et la Mer	Editions Gallimard	2018	978-2072762093
L'idiot	Les Editions de Londres	2019	978-1911572909
Les Grandes Espérances	BoD - Books on Demand	2019	978-2322185801
Par-delà le mur du sommeil	République des Lettres	2018	978-2824904269
Romancero gitano	Greenbooks editore	2020	978-8832957402
Anna Karénine	Bibliothèque russe et slave	2018	978-2371240087
Les Âmes mortes	Bibliothèque russe et slave	2018	978-2371240001

- **Exercice :**
Obtenir toutes les colonnes de la relation (table) « livre » dont le titre contient la chaîne de caractères « roi ».

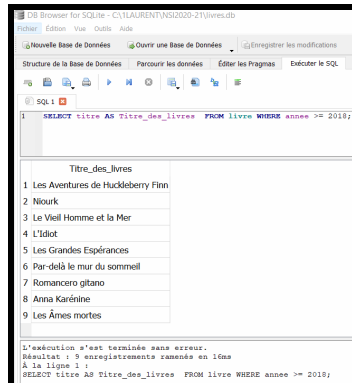
3.2.1.6. REQUETE EN RENOMMANT LES COLONNES (ALIAS) AVEC AS

- **Commande :**
`SELECT titre AS Titre_des_livres FROM livre WHERE annee >= 2018;`

- **Signification :**

Permet d'obtenir les titres de la relation (table) « livre » qui sont parus en ou après 2018. Lors de l'affichage la colonne titre est renommée en Titre_des_livres.

- **Résultat :**



- **Exercice :**

Obtenir toutes les colonnes de la relation (table) « livre » dont le titre contient la chaîne de caractères « roi » en renommant titre par Titre, isbn par ISBN, éditeur par Editeur et année par Année.

3.2.2. OPERATIONS SUR LES DONNEES : FONCTIONS D'AGREGATION

3.2.2.1. COMPTAGE DES DONNEES AVEC COUNT

- **Commande :**

```
SELECT COUNT(*) AS  
total_livre_avec_Astérix_dans_le_titre FROM livre WHERE  
titre LIKE "%Astérix%";
```

- **Signification :**

Permet de compter le nombre d'enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre. Le résultat sera le seul élément d'une colonne nommée « total_livre_avec_Astérix_dans_le_titre ».

- **Résultat :**



- **Exercice :**

Compter le nombre d'enregistrements de la relation (table) « livre » comportant le mot "nuit" dans le titre. Le résultat sera le seul élément d'une colonne nommée « Total_livre_avec_nuit_dans_le_titre ».

3.2.2.2. SOMME DES VALEURS NUMERIQUES DES DONNEES AVEC SUM

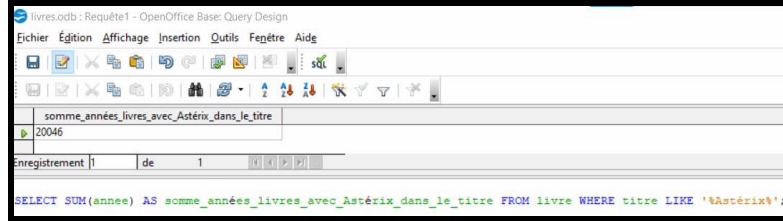
- **Commande :**

```
SELECT SUM(annee) AS  
somme_années_livres_avec_Astérix_dans_le_titre FROM  
livre WHERE titre LIKE "%Astérix%";
```

- **Signification :**

Permet de faire la somme des années des enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre. Le résultat sera le seul élément d'une colonne nommée « somme_années_livres_avec_Astérix_dans_le_titre ».

- **Résultat** (avec LibreOffice base ou OpenOffice base remplacer « " » par « ' ») :



- **Exercice :**

Faire la somme des années des enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre. Le résultat sera le seul élément d'une colonne nommée « somme_années_livres_avec_nuit_dans_le_titre ».

3.2.2.3. MOYENNE DES VALEURS NUMERIQUES DES DONNEES AVEC AVG

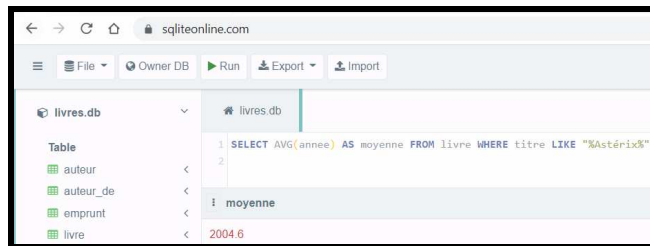
- **Commande :**

```
SELECT AVG(annee) AS moyenne FROM livre WHERE titre LIKE "%Astérix%";
```

- **Signification :**

Permet de faire la moyenne (average) des années des enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre. Le résultat sera le seul élément d'une colonne nommée « moyenne ».

- **Résultat :**



- **Exercice :**

Faire la moyenne (average) des années des enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre. Le résultat sera le seul élément d'une colonne nommée « moyenne ».

3.2.2.4. MINIMUM DES VALEURS NUMERIQUES DES DONNEES AVEC MIN

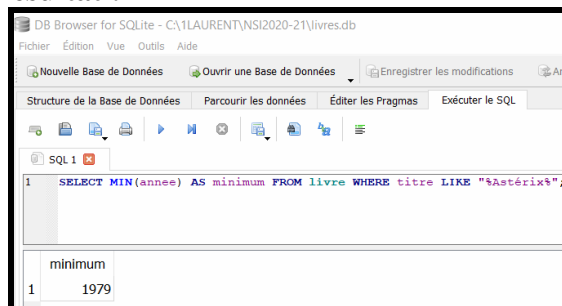
- **Commande :**

```
SELECT MIN(annee) AS minimum FROM livre WHERE titre LIKE "%Astérix%";
```

- **Signification :**

Permet de trouver la plus petite valeur des années des enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre. Le résultat sera le seul élément d'une colonne nommée « minimum ».

- **Résultat :**



- **Exercice :**

Trouver la plus petite valeur des années des enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre. Le résultat sera le seul élément d'une colonne nommée « minimum ».

3.2.2.5. MAXIMUM DES VALEURS NUMERIQUES DES DONNEES AVEC MAX

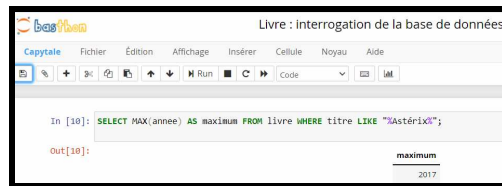
- **Commande :**

```
SELECT MAX(annee) AS maximum FROM livre WHERE titre LIKE "%Asterix%";
```

- **Signification :**

Permet de trouver la plus grande valeur des années des enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre. Le résultat sera le seul élément d'une colonne nommée « maximum ».

- **Résultat :**



- **Exercice :**

Trouver la plus grande valeur des années des enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre. Le résultat sera le seul élément d'une colonne nommée « maximum ».

3.2.2.6. TRI DES VALEURS NUMERIQUES DES DONNEES AVEC ORDER BY

3.2.2.6.1. TRI DES VALEURS NUMERIQUES DES DONNEES AVEC DESC

- **Commande :**

```
SELECT titre, annee FROM livre WHERE titre LIKE "%Asterix%" ORDER BY annee DESC;
```

- **Signification :**

Permet d'afficher tous les enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre et dont l'année de parution est classée par ordre décroissant.

- **Résultat** (avec LibreOffice base ou OpenOffice base remplacer « " » par « ' ») :

titre	annee
Asterix et la Transalique	2017
Asterix chez les Pictes	2013
Asterix légionnaire	2011
L'Odyssée d'Asterix	2008
Le Tour de Gaule d'Asterix	2007
Asterix et les Normands	2005
Asterix en Corse	2005
Asterix chez les Bretons	2002
Asterix et Cleopâtre	1999
Asterix chez les Belges	1979

- **Exercice :**

Afficher tous les enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre et dont l'année de parution est classée par ordre décroissant.

3.2.2.6.2. TRI DES VALEURS NUMERIQUES DES DONNEES AVEC ASC

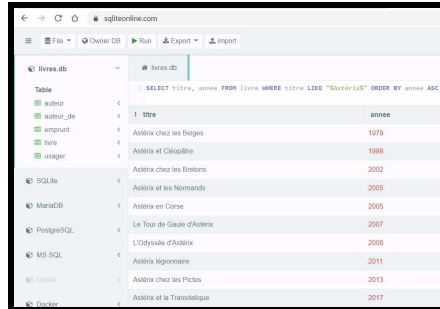
- **Commande :**

```
SELECT titre, annee FROM livre WHERE titre LIKE "%Asterix%" ORDER BY annee ASC;
```

- **Signification :**

Permet d'afficher tous les enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre et dont l'année de parution est classée par ordre croissant.

- **Résultat :**



titre	annee
Astérix chez les Belges	1979
Astérix et Cléopâtre	1999
Astérix chez les Bretons	2002
Astérix et les Normands	2005
Astérix en Corse	2005
Le Tour de Gaule d'Astérix	2007
L'Odyssée d'Astérix	2008
Astérix légionnaire	2011
Astérix chez les Pictes	2013
Astérix et la Transalpinique	2017

- **Exercice :**

Afficher tous les enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre et dont l'année de parution est classée par ordre croissant.

3.2.2.6.3. TRI DES NOMS DES DONNEES PAR ORDRE ALPHABETIQUE INVERSE AVEC DESC

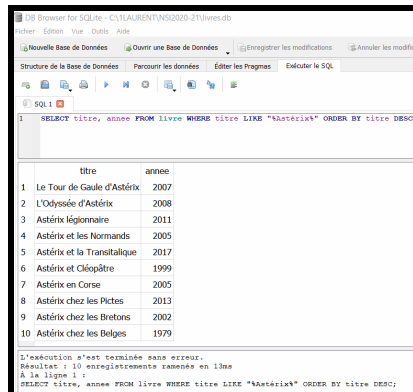
- **Commande :**

```
SELECT titre, annee FROM livre WHERE titre LIKE "%Astérix%" ORDER BY titre DESC;
```

- **Signification :**

Permet d'afficher tous les enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre et dont les titres sont classés par ordre alphabétique inversé.

- **Résultat :**



titre	annee
Le Tour de Gaule d'Astérix	2007
L'Odyssée d'Astérix	2008
Astérix légionnaire	2011
Astérix et les Normands	2005
Astérix et la Transalpinique	2017
Astérix et Cléopâtre	1999
Astérix en Corse	2005
Astérix chez les Pictes	2013
Astérix chez les Bretons	2002
Astérix chez les Belges	1979

- **Exercice :**

Afficher tous les enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre et dont les titres sont classés par ordre alphabétique inversé.

3.2.2.6.4. TRI DES NOMS DES DONNEES PAR ORDRE ALPHABETIQUE AVEC ASC

- **Commande :**

```
SELECT titre, annee FROM livre WHERE titre LIKE "%Astérix%" ORDER BY titre ASC;
```

- **Signification :**

Permet d'afficher tous les enregistrements de la relation (table) « livre » comportant le mot « Astérix » dans le titre et dont les titres sont classés par ordre alphabétique.

- **Résultat :**

titre	annee
Astérix chez les Belges	1979
Astérix chez les Bretons	2002
Astérix chez les Picotes	2013
Astérix en Corse	2005
Astérix et Cléopâtre	1999
Astérix et la Transalpinique	2017
Astérix et les Normands	2005
Astérix légendaire	2011
L'Odyssée d'Astérix	2008
Le Tour de Gaule d'Astérix	2007

- **Exercice :**

Afficher tous les enregistrements de la relation (table) « livre » comportant le mot « nuit » dans le titre et dont les titres sont classés par ordre alphabétique.

3.2.2.7. SUPPRESSION DES DOUBLONS AVEC DISTINCT

- **Commande :**

SELECT DISTINCT editeur FROM livre;

- **Signification :**

Permet d'afficher tous les éditeurs de la relation (table) « livre » sans doublons.

- **Résultat :**

editeur
Addison-Wesley Professional
adsaa
Archipoche
Arles [France]: Actes sud
Belles Lettres
Bibebok
Bibliothèque russe et slave
BnF collection ebooks
BoD - Books on Demand
Books LLC, Wiki Series

- **Exercices :**

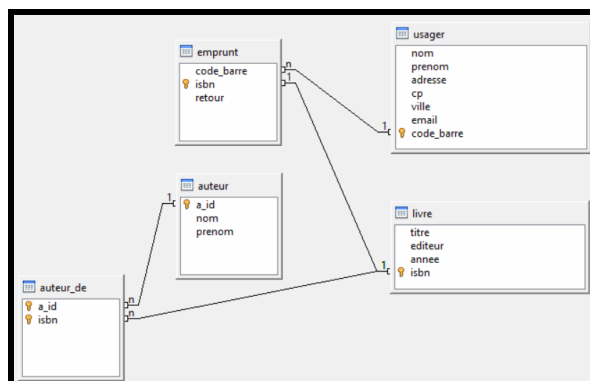
Afficher toutes les années de parutions de la relation (table) « livre » sans doublons.

Afficher toutes les années de parutions de la relation (table) « livre », sans doublons, classées par ordre croissant.

Afficher toutes les années de parutions de la relation (table) « livre », sans doublons, classées par ordre décroissant.

3.2.3. RECHERCHES CROISEES SUR LES TABLES : LES JOINTURES

En reprenant les relations (tables) selon le schéma relationnel sous forme graphique suivant :



Le contenu de la relation (table) « emprunt » est peu lisible car il contient uniquement les attributs (colonnes) « code_barre », « isbn » et « retour ». On l'obtient avec la commande :

SELECT * FROM emprunt;

Qui donne le résultat suivant :

code_barre	isbn	retour
421921003090881	978-2081358881	2020-04-28
421921003090881	978-2207249123	2020-04-28
421921003090881	978-2824709420	2020-04-28
137332830764072	978-2352879183	2020-02-20
137332830764072	978-2355006586	2020-02-20
137332830764072	978-2013230827	2020-02-20
533299198788609	978-2253174561	2020-02-28
533299198788609	978-2251013039	2020-02-28
917547585216771	978-2260105504	2020-04-07
654834075188732	978-2864973270	2020-02-17
654834075188732	978-2070406340	2020-02-17
654834075188732	978-2806221087	2020-02-17

Pour que la relation (table) « emprunt » soit lisible par un humain, il faudrait que l'on affiche à la place de l'ISBN (peu parlant pour l'humain) le titre de l'ouvrage. Or il n'est disponible que dans la relation (table) « livres ». On va donc faire une **jointure** de ces deux tables.

3.2.3.1. JOINTURE DE DEUX RELATIONS AVEC JOIN

- **Commande :**

**SELECT livre.titre, emprunt.code_barre, emprunt.retour
FROM emprunt JOIN livre ON emprunt.isbn = livre.isbn;**

- **Signification :**

Permet d'afficher la relation « emprunt » dans laquelle on remplace l'ISBN par le titre du livre. Comme plusieurs relations sont appelées, chaque attribut (colonne) est préfixé avec le nom de la relation (table).

Permet donc d'afficher le titre du livre, le code barre de l'emprunteur et la date de retour de l'emprunt en effectuant une jointure des relations « emprunt » et « livre » en mettant en lien les ISBN de ces deux relations.

- **Résultat :**

titre	code_barre	retour
Mrs Dalloway	421921003090881	2020-04-28
Fondation et Empire	421921003090881	2020-04-28
Le Journal d'un fou	421921003090881	2020-04-28
Guerre et Paix	137332830764072	2020-02-20
Les Voyages de Gulliver	137332830764072	2020-02-20
Gargantua et Pantagruel	137332830764072	2020-02-20
Les Hauts de Hurlevent	533299198788609	2020-02-28
Enéide	533299198788609	2020-02-28
Jack Barron et l'Éternité	917547585216771	2020-04-07
Astérix et la Transitalique	654834075188732	2020-02-17
Les Contes de Canterbury	654834075188732	2020-02-17
Le Père Goriot	654834075188732	2020-02-17
La Planète des singes	934701281931582	2020-01-01
Anna Karénine	934701281931582	2020-01-01
Les Robots	035184062854281	2020-02-18
Le Vieil Homme et la Mer	035184062854281	2020-02-18
Les Fils de la Médina	035184062854281	2020-02-18

Ceci a permis d'améliorer la lisibilité de la relation « emprunt », mais l'attribut (colonne) « code_barre » est encore peu lisible pour l'humain. Nous pouvons la remplacer par le titre du livre, en faisant une nouvelle jointure, en invitant maintenant les deux tables «livre » et « usager ».

3.2.3.2. JOINTURE DE TROIS RELATIONS AVEC JOIN

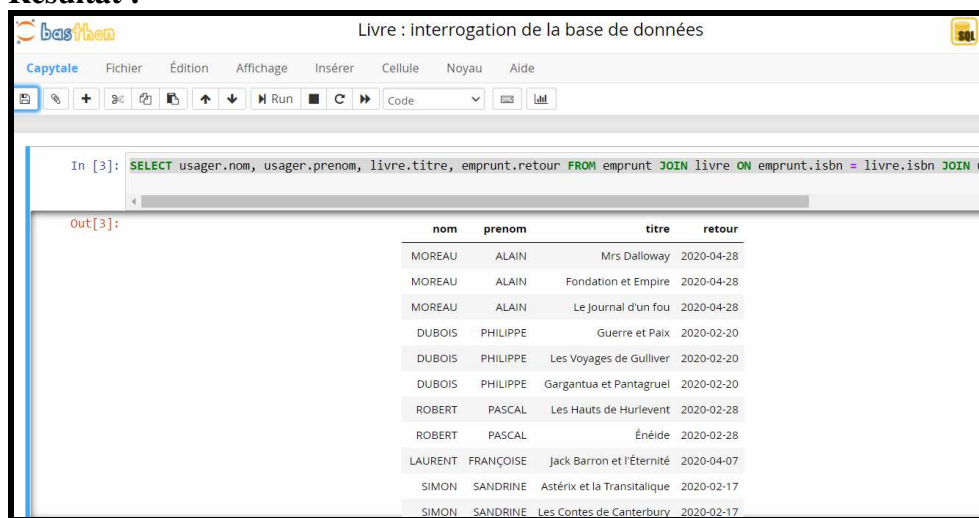
- **Commande :**

```
SELECT usager.nom, usager.prenom, livre.titre,
emprunt.retour FROM emprunt JOIN livre ON emprunt.isbn =
livre.isbn JOIN usager ON emprunt.code_barre =
usager.code_barre;
```

- **Signification :**

Permet d'afficher la relation « emprunt » dans laquelle on remplace l'ISBN par le titre du livre et le code barre de l'emprunteur par son nom et son prénom. Permet donc d'afficher le titre du livre, le nom et le prénom de l'emprunteur et la date de retour de l'emprunt en effectuant une première jointure des relations « emprunt » et « livre » en mettant en lien les ISBN de ces deux relations et une seconde jointure des relations « emprunt » et « usager » en mettant en lien les codes barre de ces deux relations.

- **Résultat :**



nom	prenom	titre	retour
MOREAU	ALAIN	Mrs Dalloway	2020-04-28
MOREAU	ALAIN	Fondation et Empire	2020-04-28
MOREAU	ALAIN	Le Journal d'un fou	2020-04-28
DUBOIS	PHILIPPE	Guerre et Paix	2020-02-20
DUBOIS	PHILIPPE	Les Voyages de Gulliver	2020-02-20
DUBOIS	PHILIPPE	Gargantua et Pantagruel	2020-02-20
ROBERT	PASCAL	Les Hauts de Hurlevent	2020-02-28
ROBERT	PASCAL	Énéide	2020-02-28
LAURENT	FRANÇOISE	Jack Barron et l'Éternité	2020-04-07
SIMON	SANDRINE	Astérix et la Transitalique	2020-02-17
SIMON	SANDRINE	Les Contes de Canterbury	2020-02-17

- **Exercices :**

Afficher le titre du livre, le nom, le prénom, l'adresse, le code postal et l'email de l'emprunteur ainsi que la date de retour de l'emprunt.

Afficher le nom de l'auteur, le prénom de l'auteur, le titre du livre, l'éditeur du livre et l'année de parution du livre.

3.2.3.3. JOINTURE DE TROIS RELATIONS AVEC JOIN ET ALIAS AVEC AS

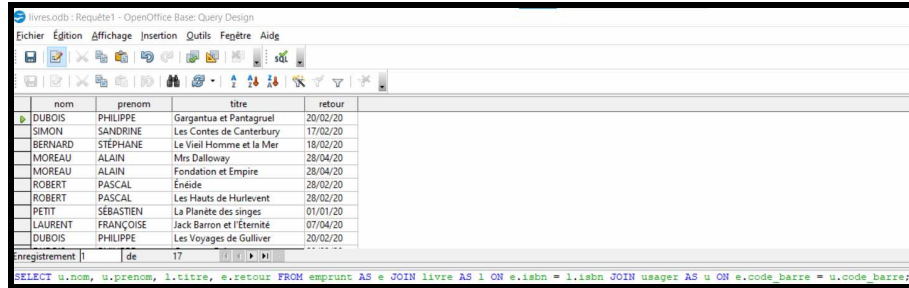
- **Commande** (équivalente à la précédente mais en utilisant des alias u, l et e) :

```
SELECT u.nom, u.prenom, l.titre, e.retour FROM emprunt
AS e JOIN livre AS l ON e.isbn = l.isbn JOIN usager AS u
ON e.code_barre = u.code_barre;
```

- **Signification :**

Permet d'afficher la relation « emprunt » dans laquelle on remplace l'ISBN par le titre du livre et le code barre de l'emprunteur par son nom et son prénom. Permet donc d'afficher le titre du livre, le nom et le prénom de l'emprunteur et la date de retour de l'emprunt en effectuant une première jointure des relations « emprunt » et « livre » en mettant en lien les ISBN de ces deux relations et une seconde jointure des relations « emprunt » et « usager » en mettant en lien les codes barre de ces deux relations. Des alias sont donnés aux tables à l'aide de AS afin de faciliter l'écriture de la requête SQL.

- **Résultat :**



	nom	prenom	titre	retour
	DUBOIS	PHILIPPE	Gargantua et Pantagruel	20/02/20
	SIMON	SANDRINE	Les Contes de Canterbury	17/02/20
	BERNARD	STEPHANE	Le Vieil Homme et la Mer	18/02/20
	MOREAU	ALAIN	Mrs Dalloway	28/04/20
	MOREAU	ALAIN	Fondation et Empire	28/04/20
	ROBERT	PASCAL	Énéide	28/02/20
	ROBERT	PASCAL	Les Hauts de Hurlevent	28/02/20
	PETIT	SÉBASTIEN	La Planète des singes	01/01/20
	LAURENT	FRANÇOISE	Jack Baran et l'Éternité	07/04/20
	DUBOIS	PHILIPPE	Les Voyages de Gulliver	20/02/20

- **Exercices :**

Afficher (en utilisant les alias « u » pour « usager », « e » pour « emprunt » et « l » pour « livre ») le titre du livre, le nom, le prénom, l'adresse, le code postal et l'email de l'emprunteur ainsi que la date de retour de l'emprunt.

Afficher (en utilisant les alias « a » pour « auteur », « ad » pour « auteur_de » et « l » pour « livre ») le nom de l'auteur, le prénom de l'auteur, le titre du livre, l'éditeur du livre et l'année de parution du livre.

3.2.4. EXERCEZ VOUS ENCORE ET ENCORE AVEC CET EXEMPLE

3.2.5. EXERCEZ VOUS AVEC « THE SQL MURDER MYSTERY »

Uniquement pour ceux qui sont très en avance.

Cet exercice en ligne est proposé le Knight Lab de l'université américaine Northwestern University.

Le point de départ de l'histoire : un meurtre a été commis dans la ville de SQL City le 15 janvier 2018. En utilisant une base de données, il s'agit de trouver le meurtrier. Le lien : <https://mystery.knightlab.com/walkthrough.html>

4. CREER ET MODIFIER LES BASES DE DONNEES

4.1. CREER DES BASES DE DONNEES AVEC CREATE TABLE

- L'objectif est de créer la relation suivante :

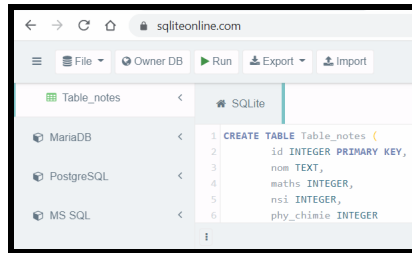
id	nom	maths	nsi	phy_chimie
1	Alice	16	11	17
2	Bob	12	15	10
3	Charles	9	10	16

- La méthode avec CREATE TABLE :
Pour DB Browser, il faut avoir au préalable créé une nouvelle base de données.

- **Commande :**

```
CREATE TABLE Table_notes (
    id INTEGER PRIMARY KEY,
    nom TEXT,
    maths INTEGER,
    nsi INTEGER,
    phy_chimie INTEGER
);
```

L'utilisateur qui spécifie ou non quel attribut sera la clé primaire.



Pour LibreOffice base ou OpenOffice base, la méthode est différente, il faut :

- « créer une table en mode ébauche ... »,
- nommer les « nom de champ » pour chaque attribut,
- donner les « type de champ » pour chaque attribut,
- indiquer l'attribut étant clé primaire,
- enregistrer la relation sous le nom de table.

Ou

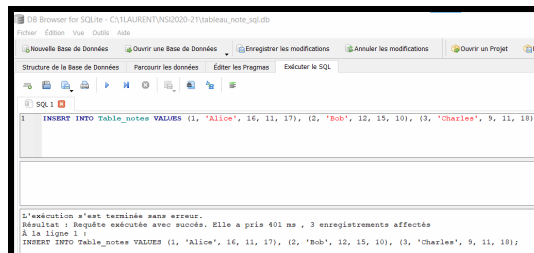
- « outils »,
- « SQL... » ,
- **CREATE TABLE Table_notes(id INTEGER PRIMARY KEY, nom VARCHAR(20), maths INTEGER, nsi INTEGER, phy_chimie INTEGER)**
- « Exécuter »,
- Enregistrer, « Affichage » et enfin « Actualiser les tables ».

4.2. INSERER DES VALEURS DANS LES BASES DE DONNEES AVEC INSERT INTO VALUES

- Commande :

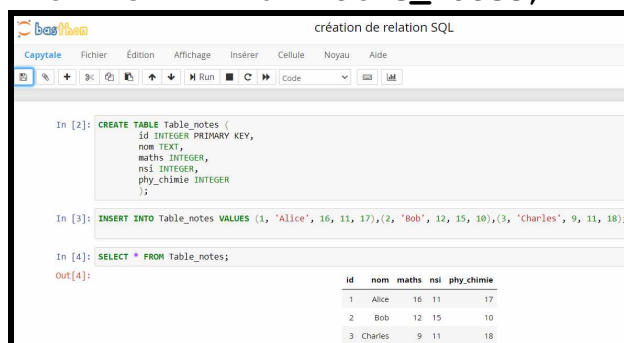
```
INSERT INTO Table_notes VALUES
(1, 'Alice', 16, 11, 17),
(2, 'Bob', 12, 15, 10),
(3, 'Charles', 9, 11, 18);
```

- Résultat :



- Vérification :

```
SELECT * FROM Table_notes;
```



Pour LibreOffice base ou OpenOffice base, la méthode est différente, il faut :

- « outils »,
- « SQL... » ,
- **INSERT INTO Table_notes VALUES (1, 'Alice', 16, 11, 17);**
- « Exécuter »,
- **INSERT INTO Table_notes VALUES (2, 'Bob', 12, 15, 10);**
- Exécuter »,
- **INSERT INTO Table_notes VALUES (3, 'Charles', 9, 11, 18);**
- Exécuter »

▪ **Exercice :**

Créez la relation « Ventes » suivantes :

Identifiants	References_fruits	Dates	Quantites_en_kg
1	551234	21-09-2021	4.2
2	551237	21-09-2021	6.3
3	551235	22-09-2021	2.4

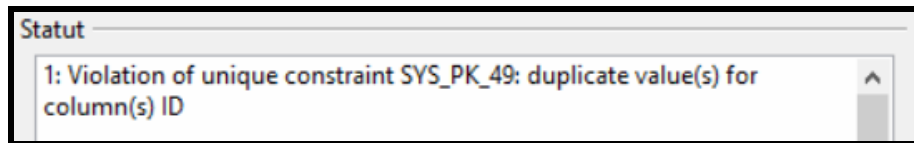
4.3. INTERET DE LA CLE PRIMAIRE DANS LES BASES DE DONNEES

Essayons d'insérer un enregistrement ayant la même clé primaire (id) qu'un autre.

• **Commande :**

INSERT INTO Table_notes VALUES (1, 'David', 11, 12, 13);

• **Résultat :**



La contrainte sur la clé primaire de la relation est violée : le SGBD « protège » la base de données en n'acceptant pas la proposition d'insertion. La base de données n'est pas modifiée.

• **Remarque :**

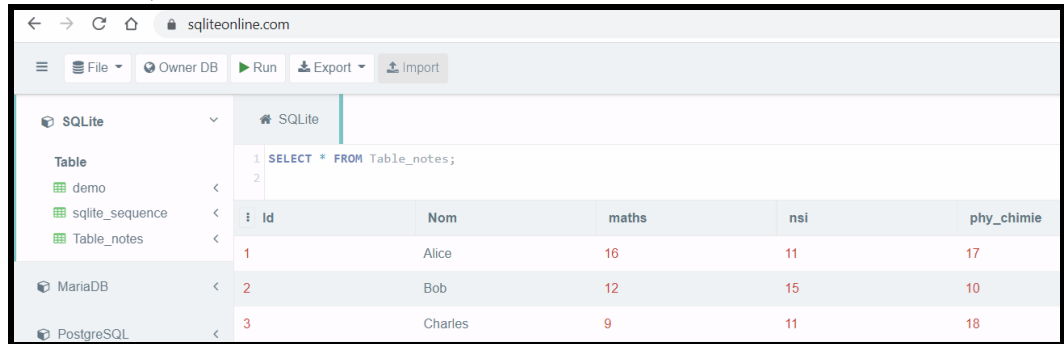
Il est possible de «déléguer» la gestion des valeurs de la clé primaire avec l'instruction **AUTOINCREMENT**. La déclaration de la relation et l'insertion des valeurs seraient :

```
CREATE TABLE Table_notes (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Nom TEXT,
    maths INTEGER,
    nsi INTEGER,
    phy_chimie INTEGER
);
```

Puis

```
INSERT INTO Table_notes (Nom, maths, nsi, phy_chimie)
VALUES
    ('Alice', 16, 11, 17),
    ('Bob', 12, 15, 10),
    ('Charles', 9, 11, 18);
```

Avec l'attribut « **id** » auto incrémenté par le sgbd (système de gestion de bases de données), le résultat serait :



id	Nom	maths	nsi	phy_chimie
1	Alice	16	11	17
2	Bob	12	15	10
3	Charles	9	11	18

▪ **Exercice :**

Créez la relation « Ventes » suivantes avec l'instruction **AUTOINCREMENT** :

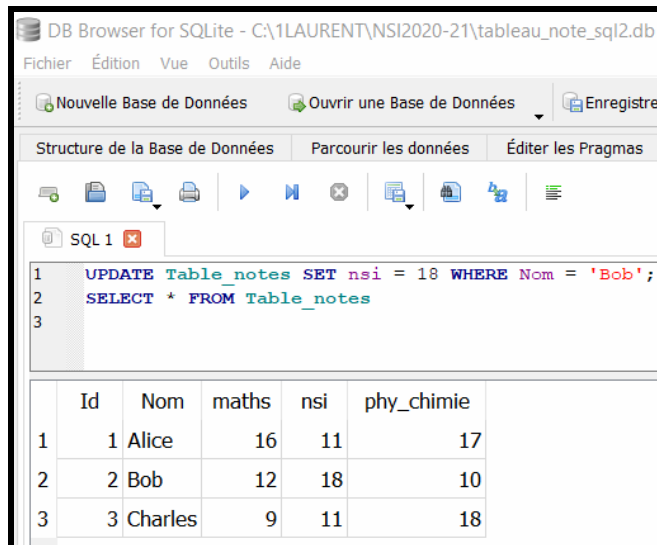
Identifiants	References_fruits	Dates	Quantites_en_kg
1	551234	21-09-2021	4.2
2	551237	21-09-2021	6.3
3	551235	22-09-2021	2.4

4.4. MODIFIER DES VALEURS DANS LES BASES DE DONNEES AVEC UPDATE ET SET

- **Commande** pour modifier la note de NSI de Bob :

UPDATE Table_notes SET nsi = 18 WHERE Nom = 'Bob';

- **Résultat :**



	Id	Nom	maths	nsi	phy_chimie
1	1	Alice	16	11	17
2	2	Bob	12	18	10
3	3	Charles	9	11	18

- **Exercice :**

Dans la relation « Ventes », modifiez la référence correspondant à l'identifiant 2 et la mettre à 551238.

4.5. SUPPRIMER DES ENREGISTREMENTS DANS LES BASES DE DONNEES AVEC DELETE

- **Commande** pour supprimer l'enregistrement de Bob :

Si une autre relation contient un attribut de la première comme clé étrangère et que l'enregistrement que l'on souhaite supprimer fait partie de cette relation, le sgbd refusera de supprimer cette ligne, afin de ne pas violer la contrainte de référence.

DELETE FROM Table_notes WHERE Nom = 'Bob';

- **Résultat :**

The screenshot shows the basethon SQL editor interface. The title bar says "création de relation SQL". The menu bar includes "Capitale", "Fichier", "Édition", "Affichage", "Insérer", "Cellule", "Noyau", and "Aide". The toolbar has icons for file operations, editing, and execution. The main area displays a series of SQL commands and their outputs:

```
In [1]: CREATE TABLE Table_notes (
        Id INTEGER PRIMARY KEY AUTOINCREMENT,
        Nom TEXT,
        maths INTEGER,
        nsi INTEGER,
        phy_chimie INTEGER
        );
```

```
In [2]: INSERT INTO Table_notes (Nom, maths, nsi, phy_chimie) VALUES
        ('Alice', 16, 11, 17),
        ('Bob', 12, 15, 10),
        ('Charles', 9, 11, 18);
```

```
In [3]: SELECT * FROM Table_notes;
```

Out[3]:

Id	Nom	maths	nsi	phy_chimie
1	Alice	16	11	17
2	Bob	12	15	10
3	Charles	9	11	18

```
In [4]: UPDATE Table_notes SET nsi = 18 WHERE Nom = 'Bob';
```

```
In [5]: SELECT * FROM Table_notes;
```

Out[5]:

Id	Nom	maths	nsi	phy_chimie
1	Alice	16	11	17
2	Bob	12	18	10
3	Charles	9	11	18

```
In [6]: DELETE FROM Table_notes WHERE Nom = 'Bob';
```

```
In [7]: SELECT * FROM Table_notes;
```

Out[7]:

Id	Nom	maths	nsi	phy_chimie
1	Alice	16	11	17
3	Charles	9	11	18

- **Exercice :**

Dans la relation « Ventes », supprimez l'enregistrement correspondant à l'identifiant 2.

4.6. SUPPRIMER UNE RELATION DANS LES BASES DE DONNEES AVEC DROP

- **Commande** pour supprimer définitivement une relation :

Si une autre relation est liée par la clé étrangère à la relation que l'on souhaite supprimer, le sgbd refusera de supprimer cette relation.

DROP TABLE Table_notes;

- **Exercice :**

Supprimez la relation « Ventes ».