

Chapitre 1

BDD partie 1

1 Introduction

1.1 BDD et SGBD

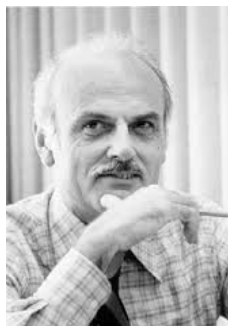
Une *base de données* (BDD) est un ensemble structuré de données ainsi que des relations logiques sur ces données.

Le *système de gestion de bases de données* (SGBD) peut être vu comme le logiciel qui gère :

- la structuration des données;
- le stockage des données;
- la maintenance des données;
- la sécurité des données;
- les accès aux données (lecture et/ou écriture) en temps réel par de multiples intervenants.

1.2 Un peu d'histoire

La majorité de ce que nous allons voir repose sur les travaux d'Edgar F. Codd, c'est lui qui a inventé le modèle relationnel en 1970 alors qu'il travaillait comme informaticien chez IBM.



1.3 Phases de conception d'une BDD

C'est une tâche essentielle pour assurer le bon fonctionnement des applications qui vont l'utiliser.

- *niveau conceptuel* : on représente la BDD à l'aide de schémas indépendamment de toute considération informatique;
- *niveau logique* on adapte le schéma en tables à deux dimensions;
- *niveau physique* on implémente les tables sur un SGBD.

2 Niveau conceptuel : modèle entités-associations

2.1 Entités

On commence par déterminer les types des entités qui interviennent :

- une *entité* est un objet unique avec un nombre fini d'attributs;
- un (ou plusieurs) attribut(s) permet(tent) d'identifier de manière unique l'entité : on parle d'*identifiant(s)* ou de clé.

Type de l'entité
attribut 1 (identifiant)
attribut 2
attribut 3
...

Exemple : entité-type Auteur

Auteur
id_auteur
nom
prenom
date_naissance

2.2 Occurrences

Une entité-type a en général plusieurs occurrences, appelées entités.

id_auteur	nom	prenom	date_naissance
00000001	Dupond	Marie	23/08/1982
12345678	Martin	Luce	13/05/1963
98765432	Leblanc	Jean	18/11/1974

Remarque

Par souci de simplicité, on parlera d'entité à la fois pour désigner l'entité-type et chacune de ses occurrences.

2.3 Entité Pays

Voici une autre entité entrant en jeu dans la BDD :

Pays
nom_pays
population superficie

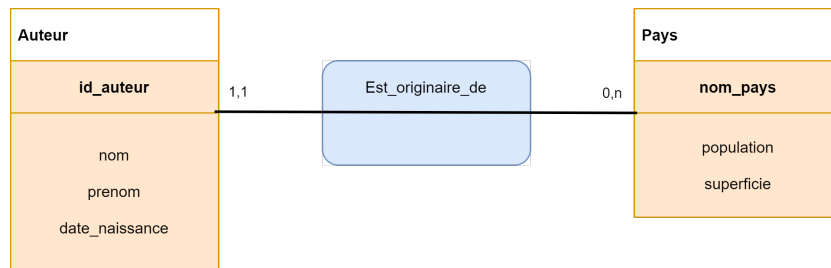
2.4 Associations

Elles définissent des *liens sémantiques* (des liens de sens) que les simples entités ne suffisent pas à définir.

Une association comporte :

- un nom ;
- un lien entre 2 relations ;
- deux cardinalités qui sont représentées sur les extrémités du lien ;
- parfois elle peut comporter un ou des attributs.

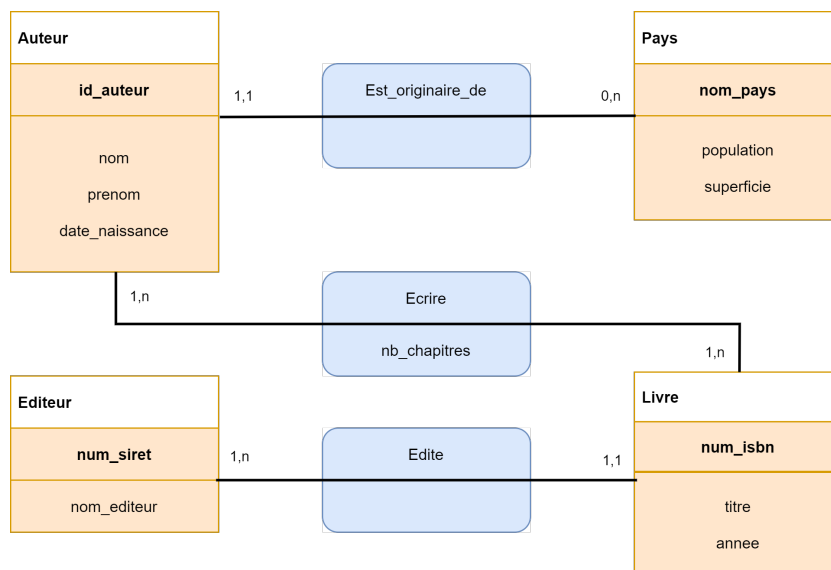
Exemple



Une cardinalité est un couple d'entiers « de type min, max » :

- la cardinalité 1,1 signifie qu'un auteur peut être lié au minimum à 1 pays, et au maximum à 1 pays (donc à un pays et un seul);
- la cardinalité 0,n signifie qu'un pays peut être lié au minimum à aucun auteur et au maximum plusieurs auteurs.

2.5 Un schéma abouti



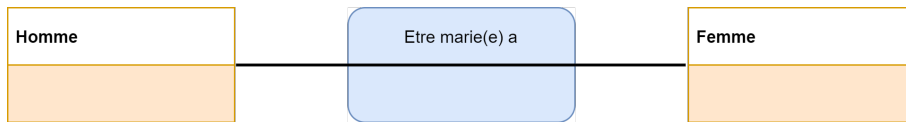
2.6 Non-unicité des schémas

Le schéma précédent s'appelle un modèle conceptuel des données (MCD).

On peut modéliser une situation avec plusieurs MCD, chacun d'entre eux ont leurs avantages et leurs inconvénients.

3 Exercices

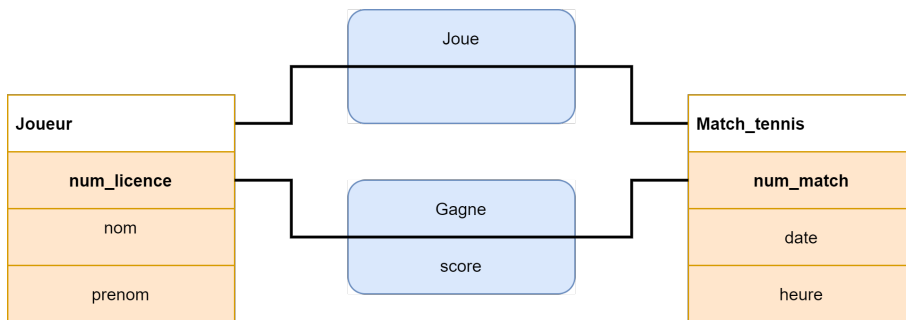
Exercice 1



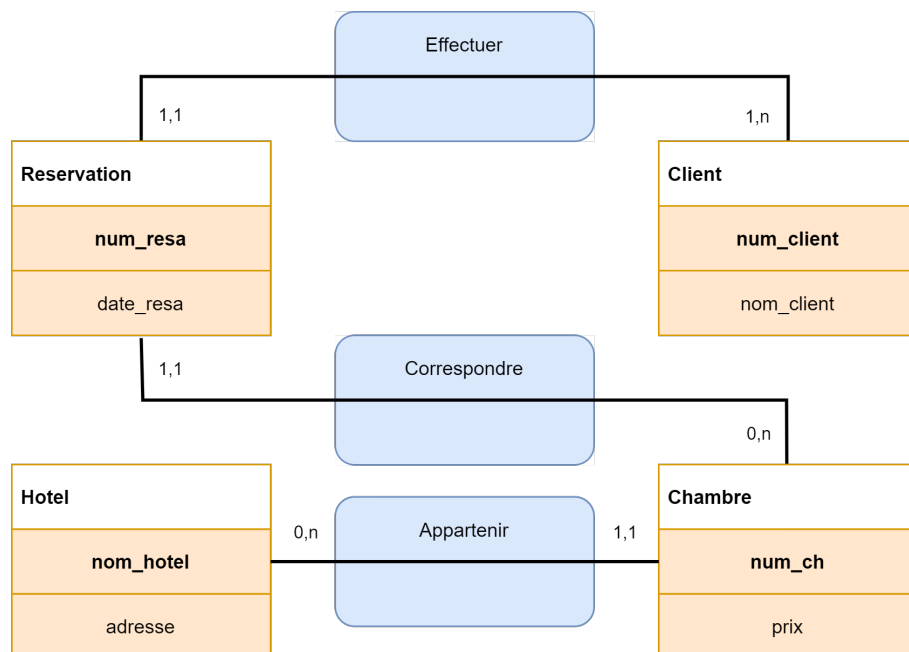
Préciser les cardinalités de l'association *Etre marie(e) a* :

- dans une société monogame ;
- dans une société dans laquelle les femmes ont le droit d'être polygames.

Exercice 2



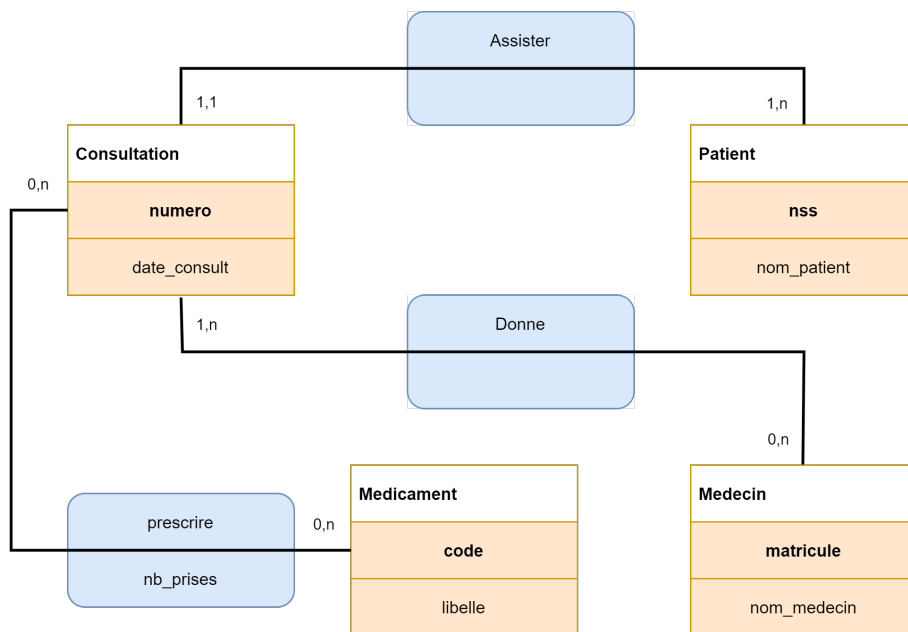
Préciser les cardinalités des associations.

Exercice 3

Dans ce modèle

1. Peut-on avoir des clients homonymes ?
2. Un client peut-il réserver plusieurs chambres à une même date ?
3. Est-il possible de réserver une chambre plusieurs jours d'affilée ?
4. Peut-on savoir si une chambre est libre à une date donnée ?
5. Peut-on réserver la même chambre plusieurs fois à la même date ?

Exercice 4



Dans ce modèle

1. Un patient peut-il assister à plusieurs consultations ?
2. Deux patients peuvent-ils assister à la même consultation ?
3. Deux consultations peuvent-elles avoir lieu le même jour ?
4. Un médecin peut-il recevoir plusieurs patients lors d'une même consultation ?
5. Plusieurs médecins peuvent-ils assister à la même consultation ?
6. Une consultation entraîne-t-elle toujours une prescription ?
7. Peut-on prescrire plusieurs médicaments lors d'une même consultation ?
8. Deux médecins différents peuvent-ils prescrire le même médicament ?
9. Étant donné un médicament prescrit, peut-on toujours connaître le médecin qui l'a prescrit ?

Exercice 5

Une entreprise est identifiée par son nom.

Dans cette entreprise, un département est identifié par un nom et caractérisé par une localisation.

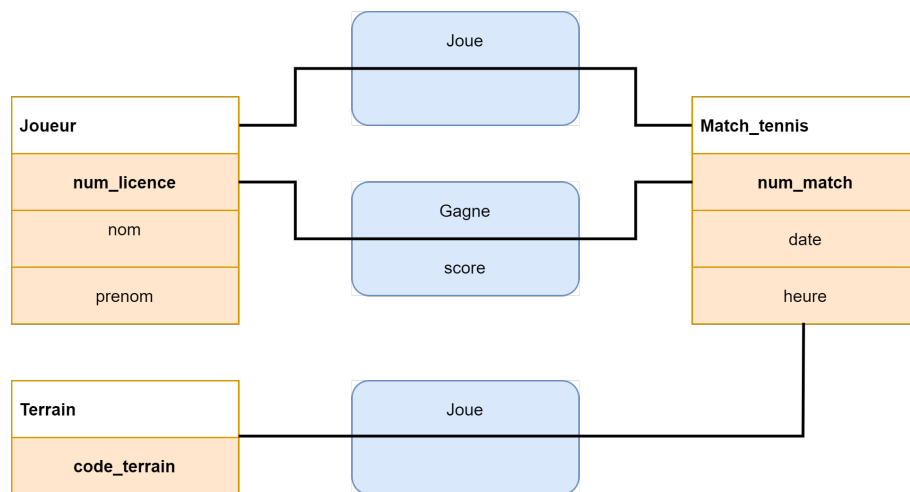
Un employé est caractérisé par un numéro, son nom, son grade et le département dans lequel il travaille.

Le numéro d'un employé est unique dans un département mais pas dans l'entreprise.

Donner le MCD, en précisant les attributs.

Indication : « caractérisé » fait référence à un attribut, « identifié » à un identifiant.

Exercice 6



1. Reprendre les cardinalités du précédent MCD sur le tennis et préciser celle de *Joue*.
2. Selon ce modèle peut-on jouer des matchs de double ?
3. Un joueur peut-il gagner un match sans y avoir participé ?
4. Peut-il y avoir 2 matchs sur le même terrain à la même heure ?
5. Connaissant un joueur, peut-on savoir sur quel(s) terrain(s) il a joué ?

Exercice 7

On considère une médiathèque contenant des ouvrages pouvant être empruntés.

Un ouvrage est caractérisé par un numéro unique, un titre, un auteur et un éditeur. En outre, on décrit un ouvrage par un certain nombre de mots-clés qui indiquent les sujets qui y sont traités.

La médiathèque dispose d'un ou plusieurs exemplaires de chaque ouvrage, L'exemplaire est identifié par un numéro et caractérisé par sa position dans les rayonnages et sa date d'achat. Un exemplaire peut être emprunté par un emprunteur. Ces derniers sont identifiés par un numéro d'emprunteur et possèdent un nom et une adresse.



Donner le MCD.

Chapitre 2

BDD partie 2

1 Niveau logique : modèle relationnel

1.1 Principe

On adapte un MCD en tables à deux dimensions.

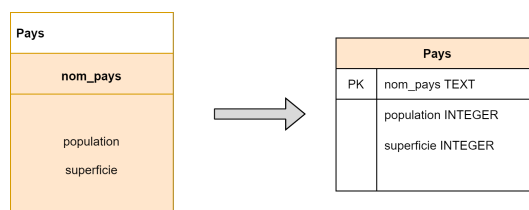
On décide du type des attributs.

Pour l'instant, on peut utiliser des types génériques, qui sont susceptibles de varier légèrement d'un SGBD à un autre :

- **INTEGER** pour les entiers;
- **FLOAT** ou **REAL** pour les nombres en virgule flottante;
- **VARCHAR(taille)** ou **TEXT** pour les chaînes de caractères de taille fixe ou illimitée;
- **BIT** pour les booléens;
- **DATE** et **TIME** pour les heures et les dates;

1.2 Transformer une entité en relation

On va transformer chaque entité du MCD en *relation* :



On indique les types de chaque attribut de la relation.

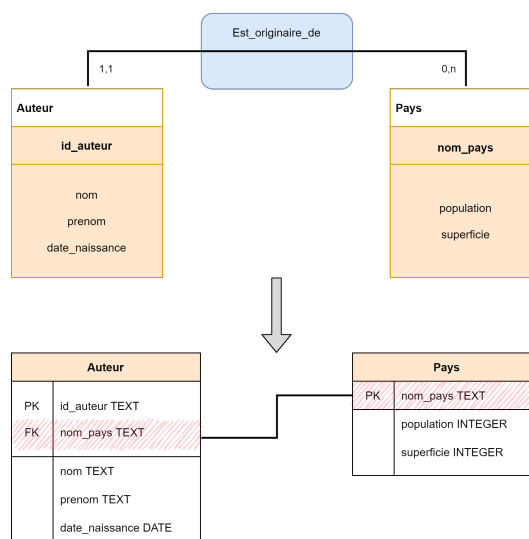
Le ou les identifiants de l'entité sont appelés des *clés primaires* pour la relation : « PK » est l'abréviation de **PRIMARY KEY**.

Le nom de la relation est noté en gras, la clé primaire soulignée.

Pays(nom_pays TEXT, population INTEGER, superficie INTEGER)

Transformer une association en relation : cas (0,1) ou (1,1)

Quand la relation possède une cardinalité valant (0,1) ou (1,1)



Puisqu'un auteur vient d'un pays et un seul, on ajoute un attribut nom_pays à la relation **Auteur**.

On précise que cet attribut est *nécessairement* l'un des attributs nom de la relation **Pays** en ajoutant « FK » dans le tableau, qui est l'abréviation de **FOREIGN KEY**.

On dit que nom_pays est une *clé étrangère*, qui *fait référence* à l'attribut nom de la relation **Pays**.

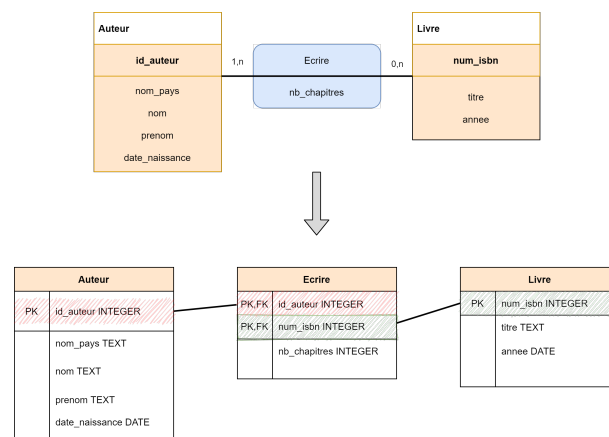
La clé étrangère est soulignée en traits discontinus.

Pays(nom_pays TEXT, population INTEGER, superficie INTEGER)

Auteur(id_auteur INTEGER, nom_pays TEXT, nom TEXT, prenom TEXTE, date_naissance DATE)

Transformer une association en relation : autre cas

Quand la relation ne possède pas de cardinalité valant (0,1) ou (1,1)



Dans ce cas on fabrique une nouvelle relation :

- on considère les clés primaires des relations issues des entités concernées par l'association;
- on fabrique une *nouvelle relation* avec comme clé primaire ce couple de de clés primaires;
- ces clés primaires sont également des *clés étrangères*;
- on ajoute si besoin est d'autres attributs spécifiques à l'association.

On va noter cela

Ecrire(id_auteur INTEGER, num_isbn INTEGER, nb_chapitres INTEGER)

1.3 Modèle complet

Pays(nom_pays TEXT, population INTEGER, superficie INTEGER)

Livre(num_isbn INTEGER, titre TEXT, annee DATE)

Auteur(id_auteur INTEGER, nom_pays TEXT, nom TEXT, prenom TEXT, date_naissance DATE)

Ecrire(id_auteur INTEGER, num_isbn INTEGER, nb_chapitres INTEGER)

Remarque

Lorsqu'on modélise une BDD, on n'a pas toujours besoin de passer par le MCD pour établir le modèle relationnel : on peut parfois le faire directement.

Bilan

Lorsqu'on établit un modèle relationnel (à partir d'un MCD ou directement) on définit des relations qui symbolisent des entités ou des associations.

On définit aussi les *contraintes* de la BDD :

- *Contraintes de domaines* : c'est essentiellement définir le type des attributs des relations;
- *Contraintes d'entité* : c'est déterminer des clés primaires pour garantir l'unicité de chaque élément d'une relation;
- *Contraintes de référence* : c'est déterminer les clés étrangères dans les relations;
- *Contraintes utilisateur* : ce sont des contraintes sur les valeurs des attributs qui garantissent leur cohérence.

1.4 Contraintes

Ces contraintes vont garantir la cohérence logique de la future base de données

- à tout instant;
- dans le cas d'une mise à jour des données (insertion ou suppression d'éléments de la relation).

Exemple de contraintes utilisateur

Dans la relation **Pays**(nom_pays TEXT, population INTEGER, superficie INTEGER)

On peut rajouter les contraintes utilisateurs suivantes :

- population > 0;
- superficie > 0.

De même dans **Auteur** et **Livre** on peut décider que les dates doivent être postérieures à une date donnée.

2 Exercices

Exercice 8

Reprendre le MCD de l'exercice 3 du chapitre « BBD partie 1 » (**Hotel Reservation Client Chambre**) et donner le modèle relationnel

- sous la forme d'un schéma;
- sous forme écrite

Exercice 9

Reprendre le MCD de l'exercice 4 du chapitre « BBD partie 1 » (**Consultation Patient Medicament Medecin**) et donner le modèle relationnel

- sous la forme d'un schéma;
- sous forme écrite

Exercice 10

Donner la modélisation relationnelle d'un bulletin scolaire. Elle doit permettre de représenter

1. des élèves possédants un numéro d'identifiant alphanumérique unique;
2. des matières, qui grâce à la dernière réforme du lycée, varient d'un élève à l'autre;
3. au plus une note sur 20 par matière.

Exercice 11

On modélise un annuaire téléphonique de la manière suivante :

Annuaire(nom TEXT, prenom TEXT, tel TEXT)

Dire si les ensembles suivants sont valides pour cette modélisation.

1. {}
2. {'titi', 'toto', '0123456789'}
3. {'titi', 'toto', '0123456789'}, {'tata', 'tutu', '0123456789'}
4. {'titi', 'toto', '0123456789'}, {'titi', 'toto',

```
'9876543210')}]}
```

```
5. {('titi', 'toto', '0123456789'), ('tata', 'tutu')}
```

```
6. {('titi', 'toto', 0123456789)}
```

Exercice 12

1. Proposer une modélisation des départements français. On veut pouvoir stocker le nom, le code, le chef-lieu et la liste de tous les départements voisins.
Attention les codes de départements ne sont pas tous des nombres : 2A et 2B pour la Corse et les départements d’Outre-Mer ont un code à 3 chiffres.
2. Proposer une contrainte utilisateur supplémentaire, non indiquée dans le schéma, pour éviter la redondance d’informations dans la liste des voisins.

Exercice 13

Proposer une modélisation du réseau de bus d’une agglomération. Elle doit permettre de générer la liste des horaires passage de chaque bus de chaque ligne pour chaque jour de la semaine arrêt par arrêt.

Chapitre 3

BDD partie 3

1 Niveau physique Langage SQL

1.1 Le SGBD

Il garantit entre autres

- *l'indépendance physique* de la BDD : l'utilisateur n'a pas à se soucier des aspects matériels;
- *l'indépendance logique* : les programmes qui utilisent la BDD sont indépendants de sa structure logique;
- *l'accès aux données* : il se fait grâce à un *langage de manipulation des données* (LMD) optimisé pour la rapidité et l'accès simultané multiple en lecture/écriture;
- *la centralisation des données pour administration*;
- *la non-redondance des données*;
- *la sécurité des données vis-à-vis du piratage* mais aussi des pannes physiques.

1.2 Principaux SGBD en 2020





1.3 Le SQL

- *Structured Query Langage* (langage de requêtes structuré).
- Créé en 1974, normalisé en 1986, dernière version parue en 2011.
- Utilisé par la plupart des SGBD avec de petites différences.

1.4 Exemple de requête

SQL

```
SELECT DISTINCT nom, prenom
FROM Auteur
      JOIN Ecrire ON Ecrire.id_auteur = Auteur.id_auteur
      JOIN Livre  ON Livre.num_isbn = Ecrire.num_isbn
WHERE Livre.titre LIKE '%s%';
```

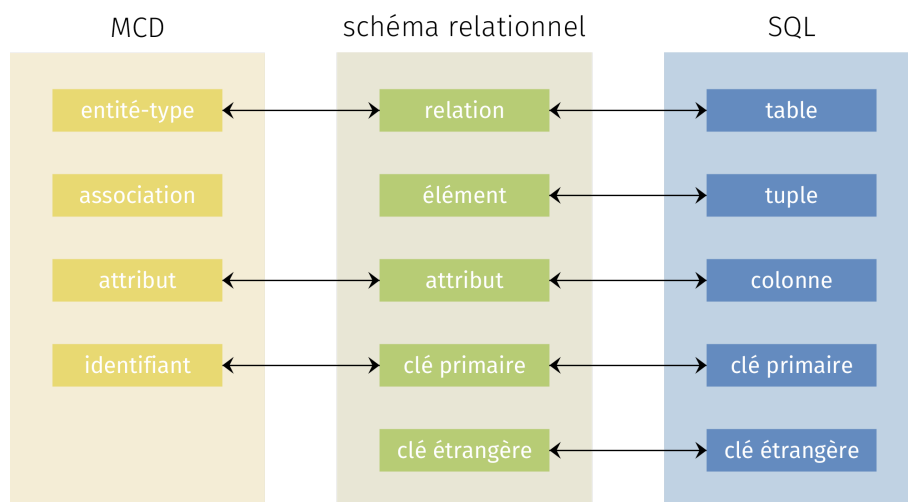
Voici comment obtenir la liste des noms et prénoms des auteurs ayant écrit un livre dont le titre comporte la lettre «s». Nous expliquerons comment produire de telles requêtes plus tard.

1.5 Vocabulaire

En SQL, les relations s'appellent des *tables*.

Les éléments des tables s'appellent des *tuples*.

1.6 Bilan des termes utilisés



1.7 Conventions

On écrira les mots-clés SQL en majuscules.

On ne met pas d'accents ou d'espaces dans les noms des tables ou des attributs.

Les espaces et tabulations n'ont qu'un rôle esthétique.

Les requêtes peuvent prendre plusieurs lignes mais doivent se terminer par un point-virgule.

On utilisera SQLite car on peut s'en servir avec DB Browser sans installation compliquée.

2 Création de la BDD

2.1 Créer une BDD

SQL

```
CREATE DATABASE Bibliotheque;  
USE Bibliotheque;
```

On n'utilisera pas (ou peu) cette commande dans les exercices.

2.2 Supprimer une BDD

SQL

```
DROP DATABASE Bibliotheque;
```

On n'utilisera pas cette commande non plus.

2.3 Créer une table

Voici comment créer la table **Pays** :

SQL

```
DROP TABLE IF EXISTS Pays; -- recréer la table de zéro
CREATE TABLE Pays
(
    nom_pays    TEXT,
    population  INTEGER,
    superficie   INTEGER,
    PRIMARY KEY (nom_pays), -- clé primaire
    CHECK (population > 0), -- contraintes utilisateur
    CHECK (superficie > 0)
);
```

2.4 Table Livre

SQL

```
DROP TABLE IF EXISTS Livre;
CREATE TABLE Livre
(
    num_isbn  INTEGER,
    titre     TEXT,
    annee     TEXT,
    PRIMARY KEY (num_isbn),
    CHECK (date(annee) BETWEEN '1900' AND '2100')
);
```

`date(annee) BETWEEN '1900' AND '2100'` est l'équivalent SQL de `'1900' <= date(annee) <= '2100'` en Python.

Attention

SQLite ne connaît pas le type **DATE**, il faut créer des attributs de type **TEXT** et utiliser la fonction **date**.

2.5 Table Livre**SQL**

```
DROP TABLE IF EXISTS Auteur;
CREATE TABLE Auteur
(
    id_auteur      INTEGER,
    nom_pays       TEXT,
    nom            TEXT,
    prenom         TEXT,
    date_naissance TEXT,
    PRIMARY KEY (id_auteur),
    UNIQUE (nom, prenom), -- contrainte d'unicité
    FOREIGN KEY (nom_pays) REFERENCES Pays (nom_pays)
    /*nom_pays est une clé étrangère*/
    ON DELETE CASCADE
    /*si on supprime des tuples dans Pays, automatiquement
    (en cascade) on supprimera les tuples qui y font
    référence dans Auteur*/
    ON UPDATE CASCADE
    /*si on met à jour les attributs nom_pays dans Pays,
    alors le SGBD les mettra à jour aussi dans Auteur*/
);
```

2.6 Ordre des créations des tables

On ne peut pas créer **Auteur** avant d'avoir créé **Pays** car **Auteur** possède une clé étrangère liée à **Pays**.

2.7 Table Ecrire

SQL

```
DROP TABLE IF EXISTS Ecrire;
CREATE TABLE Ecrire
(
    id_auteur    INTEGER,
    num_isbn     INTEGER,
    nb_chapitres INTEGER,
    PRIMARY KEY (id_auteur, num_isbn),
    FOREIGN KEY (id_auteur) REFERENCES Auteur (id_auteur)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (num_isbn) REFERENCES Livre (num_isbn)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

3 Variantes syntaxiques et autres

3.1 Écriture plus compacte

On peut signifier qu'un attribut est une clé étrangère dans sa définition même :

SQL

```
DROP TABLE IF EXISTS Ecrire;
CREATE TABLE Ecrire
(
    id_auteur    INTEGER REFERENCES Auteur (id_auteur)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    num_isbn     INTEGER REFERENCES Livre (num_isbn)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    nb_chapitres INTEGER,
    PRIMARY KEY (id_auteur, num_isbn)
);
```

3.2 Écriture plus compacte (bis)

On peut signifier qu'un attribut est une clé primaire dans sa définition même :

SQL

```
DROP TABLE IF EXISTS Livre;  
CREATE TABLE Livre  
(  
    num_isbn INTEGER PRIMARY KEY,  
    titre    TEXT,  
    annee    TEXT,  
    CHECK (date(annee) BETWEEN '1900' AND '2100')  
);
```

4 Insertion des données dans la BDD

4.1 Données de Pays

SQL

```
INSERT INTO Pays  
VALUES ('France', 672051, 67064000),  
       ('Italie', 301336, 66436000),  
       ('Royaume-Uni', 242900, 60317000);
```

Les attributs des tuples sont dans le même ordre que lors de la création.

4.2 Et cætera

De même que lors de la création, on ne peut pas insérer de tuples dans **Auteur** avant d'avoir peuplé **Pays** : en effet dans un tuple de **Auteur** tel que

```
(1, 'France', 'Hugo', 'Victor', '1802-02-26')
```

Les contraintes de référence font qu'un tuple « France » doit d'abord exister dans **Pays**.

5 Exercices

Exercice 14

Ouvrir le fichier `create_Auteurs.sql` qui a servi à créer la BDD du cours (avec le bloc-notes ou autre).

1. Trouver une écriture compacte telle que vue dans le cours.
2. Trouver une écriture compacte qui n'avait pas été vue.
3. En fouillant « à vue d'œil » dans les données
 - a. Combien y a-t-il de livres ?
 - b. Combien y a-t-il d'auteurs ?
 - c. Où voit-on clairement qu'un auteur a écrit 2 livres ?

Exercice 15

1. Voici le modèle relationnel de l'exercice **Hotel Reservation Client Chambre** du chapitre précédent :

Hotel(nom_hotel TEXT, adresse TEXT)

Chambre(num_ch INTEGER, prix INTEGER, nom_hotel TEXT)

Client(num_client INTEGER, nom_client TEXT)

Reservation(num_resa INTEGER, date_resa DATE, num_client INTEGER, num_chambre INTEGER)

Écrire le script SQL qui crée cette BDD.

2. Insérer 3 valeurs de votre choix dans chaque table.

Exercice 16

On considère une base de données qui ne contient aucune table.

Dire quelles erreurs produisent les séquences SQL suivantes (tout au moins ce qui n'est pas acceptable du point de vue du cours).

1. `DROP TABLE Client;`


```
CREATE TABLE Client
(
    cid          INTEGER,
    nom          TEXT,
    prenom       TEXT,
    points_fidelite INTEGER NOT NULL,
    PRIMARY KEY (cid),
    CHECK (points_fidelite >= 0)
);
```

2. CREATE TABLE Client

```
(
    cid          INTEGER PRIMARY KEY,
    nom          TEXT,
    prenom       TEXT,
    points_fidelite INTEGER NOT NULL,
    CHECK (points_fidelite >= 0)
);
```

CREATE TABLE Commande

```
(
    cid INTEGER PRIMARY KEY, -- variante plus rapide et
    ↪ valide
    pid INTEGER,
    date TEXT NOT NULL,
    FOREIGN KEY (cid) REFERENCES Client (cid),
    FOREIGN KEY (pid) REFERENCES Client (pid)
);
```

CREATE TABLE Produit

```
(
    pid INTEGER PRIMARY KEY,
    nom TEXT,
    prix REAL(10, 2) -- 10 chiffres max avant la virgule, 2
    ↪ après
);
```

3. CREATE TABLE Client

```
(
    cid          INTEGER PRIMARY KEY,
    nom          TEXT,
    prenom       TEXT,
```

```

        points_fidelite INTEGER NOT NULL,
        CHECK (points_fidelite >= 0)
    );
CREATE TABLE Produit
(
    pid INTEGER PRIMARY KEY,
    nom TEXT,
    prix REAL(10, 2)
);
CREATE TABLE Commande
(
    cid TEXT REFERENCES Client (cid),
    nomp INTEGER REFERENCES Produit (nom),
    date TEXT NOT NULL,
    PRIMARY KEY (cid, pid)
);

```

```

4. CREATE TABLE Client
(
    cid INTEGER PRIMARY KEY,
    nom TEXT,
    prenom TEXT,
    points_fidelite INTEGER NOT NULL,
    CHECK (points_fidelite >= 0)
);
CREATE TABLE Produit
(
    pid INTEGER PRIMARY KEY,
    nom TEXT,
    prix REAL(10, 2)
);
CREATE TABLE Commande
(
    cid INTEGER,
    pid INTEGER,
    date TEXT NOT NULL,
    PRIMARY KEY (cid, nomp),
    FOREIGN KEY (cid) REFERENCES Client (cid),
    FOREIGN KEY (pid) REFERENCES Produit (pid)
);

```

```
INSERT INTO Commande VALUES(0, 0, '2020-03-02')
```

Exercice 17

Liste les différents attributs de cette relation. Donne le domaine de chaque attribut. Pour chaque attribut dire s'il peut jouer ou non le rôle de clef primaire et pourquoi.

Film

id	titre	realisateur	ann_sortie	note_sur_10
1	Alien, le huitième passager	Scott	1979	10
2	Dune	Lynch	1985	5
3	2001 : l'Odyssée de l'Espace	Kubrick	1968	9
4	Blade Runner	Scott	1982	10

Exercice 18

Indique les attributs qui peuvent servir de lien entre ces deux relations.

Auteur

id	nom	prenom	ann_naiss	langue_ecriture
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Asimov	Isaac	1920	anglais
4	Barjavel	René	1911	français
5	Verne	Jules	1828	français
...

Livre

id	titre	id_auteur	ann_publi	note
...
34	La nuit des temps	4	1968	7
35	De la Terre à la Lune	5	1865	10
36	Les Robots	6	1950	9
...

Exercice 19

1. En partant de la relation **Film** ci-dessus, crée une relation **Realisateur** (attributs de la relation : **id**, **nom**, **prenom** et **ann_naissance**, tu trouveras toutes les informations nécessaires sur le Web). Modifie ensuite la relation **Film** afin d'établir un lien entre les relations **Film** et **Realisateur**. Tu préciseras l'attribut qui jouera le rôle de clef étrangère.
2. Écris le code SQL permettant de générer les 2 tables.
3. Écris le code SQL pour insérer les films et les réalisateurs correspondants.

Chapitre 4

BDD partie 4

1 Requêtes SQL

1.1 Requête et résultat

Une requête est une commande SQL et renvoie une table.
On se replace dans le contexte du chapitre précédent.

1.2 Sélection d'attributs

```
SELECT nom, prenom  
FROM Auteur;
```

nom	prenom
Ammaniti	Niccolo
Avallone	Silvia
Camus	Albert
Hamilton	Peter
Hugo	Victor
Murgia	Michela
Rhode james	Montague
Tolkien	John

1.3 Sélection de tous les attributs

```
SELECT *  
FROM Auteur;
```

id_auteur	nom_pays	nom	prenom	date naissance
1	France	Hugo	Victor	1802-02-26
2	France	Camus	Albert	1913-11-07
4	Italie	Avallone	Silvia	1948-04-13
5	Italie	Ammaniti	Niccolo	1966-09-25
6	Italie	Murgia	Michela	1972-06-03
7	Royaume-Uni	Hamilton	Peter	1960-03-02
8	Royaume-Uni	Tolkien	John	1892-01-03
9	Royaume-Uni	Rhode James	Montague	1862-08-01

1.4 Sélection avec condition

```
SELECT nom, date_naissance
FROM Auteur
WHERE date(date_naissance) < '1900';
```

nom	date_naissance
Hugo	1802-02-16
Tolkien	1892-01-03
Rhode James	1862-08-01

1.5 Sélection avec conditions multiples

```
SELECT nom, date_naissance
FROM Auteur
WHERE date(date_naissance) < '1900'
AND nom_pays = 'France';
```

nom	date_naissance
Hugo	1802-02-16

1.6 Renommer les colonnes

```
SELECT titre AS Titre_ouvrage, num_isbn AS Reference_ISBN
FROM Livre
WHERE date(annee) > '2015';
```

Titre_ouvrage	Reference_ISBN
les misérables	9782072730672
et je t'emmène	9782221133651
d'acier	9782867465987
salvation	9791093835334

1.7 Fonction COUNT

```
SELECT COUNT(titre) AS Nb_Livres_avant_2015
FROM Livre
WHERE annee < 2015;
```

Nb_Livres_avant_2015

4

1.8 Autres fonctions similaires

Fonctions MIN, MAX, SUM et AVG (moyenne).

1.9 Éliminer les doublons

Sans élimination :

```
SELECT id_auteur
FROM Ecrire;
```

id_auteur

1

1

2

4

5

6

7

8

9

Avec élimination :

```
SELECT DISTINCT id_auteur  
FROM Ecrire;
```

id_auteur
1
2
4
5
6
7
8
9

1.10 Ordonner les tuples

Ordonner les noms dans l'ordre croissant :

```
SELECT nom,prenom FROM Auteur  
ORDER BY nom ASC;
```

nom	prenom
Ammaniti	Niccolo
Avallone	Silvia
Camus	Albert
Hamilton	Peter
Hugo	Victor
Murgia	Michela
Rhode james	Montague
Tolkien	John

Pour l'ordre décroissant on utilise **DESC**.

2 Jointures

2.1 Principe

Considérons 2 tables T1 et T2 et supposons que c est une clé étrangère qui fait référence à b.

a	b	c	d
0	0	0	10
0	1	0	30
1	1	1	12
2	1	2	100
3	2	2	200
4	5		

Voici table qui est la *jointure* T1 et T2 selon la condition $b=c$:

a	b	c	d
0	0	0	10
0	0	0	30
0	1	1	12
1	1	1	12
2	1	1	12
3	2	2	100
3	2	2	200

C'est la table obtenue en faisant correspondre chaque tuple de T1 avec chaque autre tuple de T2 tel que b et c soient égaux.

2.2 Applications

Produire la table des noms des auteurs venant de pays de plus de 61 millions d'habitants :

```
SELECT nom
FROM Auteur
      JOIN Pays ON Auteur.nom_pays = Pays.nom_pays
WHERE population > 62000000;
```

nom
Hugo
Camus
Avallone
Ammaniti
Murgia

Produire la table des noms et prénoms des auteurs ayant écrit un livre dont le titre comporte « la » :

```
SELECT DISTINCT nom, prenom
FROM Auteur
      JOIN Ecrire ON Ecrire.id_auteur = Auteur.id_auteur
      JOIN Livre ON Livre.num_isbn = Ecrire.num_isbn
WHERE Livre.titre LIKE '%la%';
```

nom	prenom
Camus	Albert
Rodhe James	Montague

3 Mises à jour

3.1 INSERT INTO

Insérer un nouveau tuple dans la table **Auteur** :

```
INSERT INTO Auteur VALUES
(128, 'France', 'Leleu', 'Frédéric', '1974-05-16');
```

Les colonnes doivent être dans le même ordre qu'à la création, sinon utiliser

```
INSERT INTO Auteur VALUES (nom, id_auteur)
('Leleu', 128);
```

Les colonnes non renseignées prendront par défaut la valeur **NULL** ce qui peut poser problème.

3.2 DELETE

Supprimer les tuples de **Ecrire** dont l'auteur a l'id_auteur 1 :

```
DELETE FROM Ecrire WHERE id_auteur = 1;
```

Penser aux contraintes de références (clé étrangères) : si on supprime un tuple et qu'un tuple d'une autre table fait référence à celui qu'on supprime, cela provoquera une erreur.

3.3 UPDATE

Mettre à jour l'id du tuple de **Auteur** dont le nom est Hugo

```
UPDATE Auteur
SET id_auteur = 1024
WHERE nom = 'Hugo';
```

Penser aux contraintes de références (clé étrangères) lors de la mise à jour.

4 Exercices

Exercice 20 : Prix Nobel



Premier contact

Avec un éditeur de texte ouvrir le fichier `create_nobel.sql`.

En explorant la structure de la base de données, répondez aux questions suivantes :

1. Combien de tables possède la base de données ?
2. Combien d'attributs possède la table Nobel ?
3. Quel est le type de l'attribut année ?

La table Nobel

Importer ce fichier dans DB Browser pour créer la BDD `nobel.db`.

En explorant les données de la table Nobel, répondez aux questions suivantes :

4. Combien d'enregistrements 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 ?

Requêtes d'interrogation

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

7. Comment afficher le nom de tous les lauréats en évitant les doublons ? (809 enregistrements)
8. Comment afficher le nom de toutes les disciplines en évitant les doublons ? (6 enregistrements)
9. Quelle est la discipline de Wilhelm Conrad Röntgen ? (1 enregistrement)
10. Dans quelle discipline Paul Krugman est-il devenu Prix Nobel ? (1 enregistrement)
11. En quelle année Albert Fert a-t-il eu le prix Nobel ? (1 enregistrement)
12. Quelle est l'année de distinction de Pierre Curie ? (1 enregistrement)
13. Quelle est l'année de distinction et la matière de Bertha von Suttner ? (1 enregistrement)
14. Quels sont les lauréats distingués au XXI e siècle ? (97 enregistrements)
15. Quels sont les lauréats du prix Nobel de la Paix durant la deuxième guerre mondiale ? (2 enregistrements)
16. Quels sont les lauréats distingués en Médecine en 1901 et en 2001 ? (4 enregistrements)
17. Quels sont les lauréats des prix Nobel de Physique et de Médecine en 2008 ? (3 enregistrements)

Requêtes d'agrégation

18. Combien d'enregistrements au total comporte la table ? (816 enregistrements)
19. Combien de personnes ont reçu le prix Nobel de la paix ? (119 enregistrements)
20. Combien de personnes ont reçu le prix Nobel de littérature ? (105 enregistrements)
21. Combien de personnes ont reçu le prix Nobel de mathématiques ? (0 enregistrements)
22. Combien de personnes ont reçu un prix Nobel en 1901 ? (6 enregistrements)
23. Combien de personnes ont reçu un prix Nobel de chimie en 1939 ? (2 enregistrements)
24. En quelle année a été décerné le premier prix Nobel d'économie ? (Réponse : 1969)
25. Combien de prix Nobel a reçu Marie Curie ? (Réponse : 2)

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)? (2 enregistrements)
27. Combien y a-t-il eu de lauréats en Physique et en Chimie? (335 enregistrements)
28. Combien y a-t-il eu de lauréats de Médecine et de littérature en 2000? (4 enregistrements)
29. Nombre de lauréats différents parmi les prix Nobel de la paix? (116 enregistrements)

Requêtes de mise à jour

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

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

Exercice 21 : JO

Nous allons travailler sur une base de données liée aux Jeux Olympiques de Londres qui ont eu lieu en 2012.



Partie 1 : Étude du schéma relationnel

Avec un éditeur de texte tout simple, ouvrir le fichier `create_J0.sql`, regarder les lignes qui définissent les différentes tables de la BDD et donner sous forme écrite son schéma relationnel en soulignant clés primaires (en trait plein) et clés étrangère (en pointillés).

Partie 2 : Requêtes SQL

Avant toute chose, ouvrir DB Browser, importer le fichier `create_J0.sql` pour créer la BDD `J0.db`.

Ensuite exécuter les bonnes requêtes SQL pour obtenir les données suivantes.

Requêtes sans jointures

1. Afficher le nom et prénom des sportifs. Combien y en a-t-il ?
2. Afficher les codes des pays dont viennent les sportifs par ordre alphabétique en éliminant les doublons.
3. Afficher la liste des sportifs français (utiliser `cio = 'France'`).
4. Afficher la liste des 301 disciplines triées par l'identifiant du sport auxquelles elles se rapportent.
5. Afficher les noms des 86 pays situés après la France et avant la Russie (Russia) par ordre alphabétique.
Utiliser les opérateurs inférieur et supérieur. Remarquer que l'opérateur **BETWEEN** ne produit pas le résultat attendu (88 pays).
6. Afficher les 98 identifiants de discipline dont au moins une épreuve a eu lieu entre le 27 et le 31 juillet 2012 inclus.
7. Afficher les noms des 61 sportifs qui sont soit français (FRA) soit britanniques (GBR).
8. Afficher les intitulés des 131 disciplines contenant la chaîne de caractères « WOMEN ».
9. Donner les 3 pays (CIO, nom) dont on ne connaît pas le code ISO2 ou ISO3 (utiliser le critère **IS NULL**).
10. Donner les noms et prénoms des 2 sportifs dont le sexe est mentionné dans la BDD.
11. À l'aide de la fonction **COUNT**, donner le nombre de sports (pas la liste).
12. Donner le nombre de discipline(s) du sport d'identifiant 1 (pas la liste).
13. Combien de noms de familles différents sont portés par les sportifs ?
14. Donner le nombre de pays n'ont pas d'ISO2.

15. Donner le nombre de médailles d'or attribués lors de ces JO.
16. Afficher en une table le premier et le dernier évènement sportif de ces JO.

Requêtes avec jointures

17. Afficher la listes des noms et prénoms des sportifs européens.
18. Afficher la liste des disciplines dépendant de l'athlétisme.
19. Afficher toutes jours pendant lesquels un évènement lié à l'athlétisme eu lieu.
20. Afficher les noms, prénoms et médailles gagnées par des sportifs dont le sexe figure dans la BDD.
21. Afficher la liste des Français médaillés d'or.
22. Afficher les noms, prénoms, sports et disciplines des sportifs ayant obtenu une médaille d'or.