

Exemple de création d'une BDD

On considère l'exemple d'une bibliothèque. Les spécifications sont les suivantes :

- Un livre est écrit par un auteur (qui a au moins un nom et un prénom), il possède un titre et est écrit dans sa langue d'origine (laquelle est parlée dans un certain nombre de pays).
- Un lecteur est caractérisé par son numéro, son nom et son prénom.
- Les lecteurs empruntent des livres à une date donnée (et les rendent naturellement à une date ultérieure).

Q.1 Créer un schéma entité-association correspondant à cet exemple en ajoutant éventuellement des caractéristiques qui n'apparaissent pas dans la spécification.

Q.2 En déduire le modèle relationnel correspondant (tables de la base).

Q.3 Écrire les requêtes de création de bases et de tables :

- En SQL (hors programme)
- Avec votre SGBD (donc DB Browser ou MySQL).

Solution. Volontairement, la description est ambiguë.

Lecture du sujet Il faut avant toute chose déterminer quel statut (entité, attribut ou association) accorder aux notions évoquées dans la spécification. Informellement on repère les verbes dans la phrase et on associe des entités, associations ou attributs aux sujet, COD, COI etc.

Ainsi « Un livre est écrit par un auteur » suggère :

- soit la création de deux entités **livre**, **auteur** et d'une association **écrit par** ;
- soit la création d'une entité (par exemple **livre**) et d'un attribut pour cette entité (par exemple **auteur**).

Pour choisir entre entité à part entière et attribut, il faut se demander si le futur attribut est « atomique » ou demande des précisions supplémentaires. Par exemple **titre** est une information atomique (que dire de plus à ce propos si on n'est pas littéraire ?) mais un **auteur** a des caractéristiques comme nom, prénom etc. Il vient que :

- **auteur** devrait être une entité,
- **titre** peut n'être qu'un attribut.

Les verbes d'actions (sous forme active ou passive) devraient faire penser à des associations. Ainsi **Emprunté par**, **Écrit par** ou **Écrit dans** devraient être des associations.

Entités On peut créer les entités suivantes :

- Livre : attribut numéro et Titre
- Auteur : attribut numéro, nom, prénom
- Langue : attribut numéro, nom, pays
- Lecteur : attribut nom, prénom numéro

Remarque. Le cas de **pays** est particulier. En effet une langue peut-être parlée dans plusieurs pays. On dit que **pays** est un attribut à valeurs multiples.

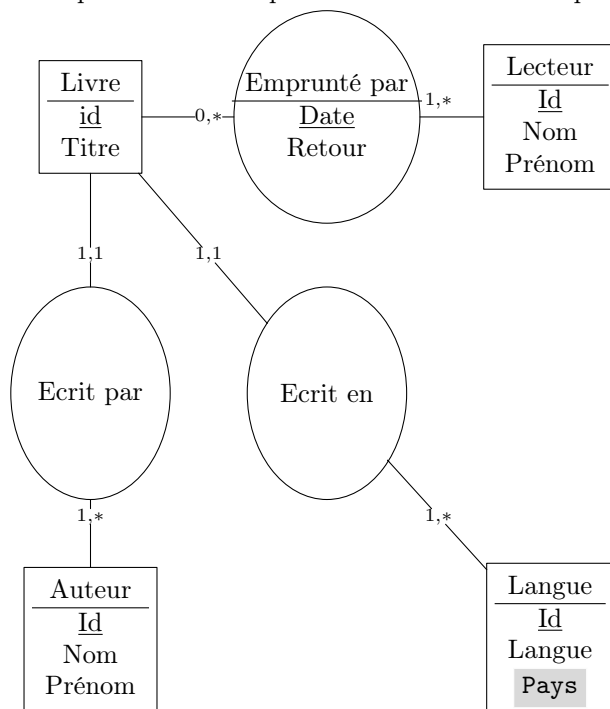
Nous le signalons par un symbole spécial (par exemple en le surlignant).

Associations Un livre est :

- *écrit par* un auteur (certains livres sont écrits par plusieurs auteurs : simplifions en ne traitant pas ce cas là).
- *emprunté* par un lecteur. Attributs d'association : Date et Retour.
- *écrit en* sa langue d'origine.

Cardinalités entités-associations

- Un livre est emprunté par 0 ou 1 lecteur (**0-1**) ;
- Un livre est écrit dans une langue (cardinalité **1-1** ou, plus simplement, **1**).
- Un lecteur peut emprunter 0 ou plusieurs livres (**0-***) ;
- Une langue peut avoir été utilisée pour écrire 0 ou plusieurs livres (**0-***)
- Un auteur peut avoir écrit un ou plusieurs livres **1-***. On peut aussi se dire qu'un auteur n'est pas forcément représenté dans la bibliothèque. Dans ce cas, ce sera **0-***.



Modèle relationnel Créer une table par entité (4 tables).

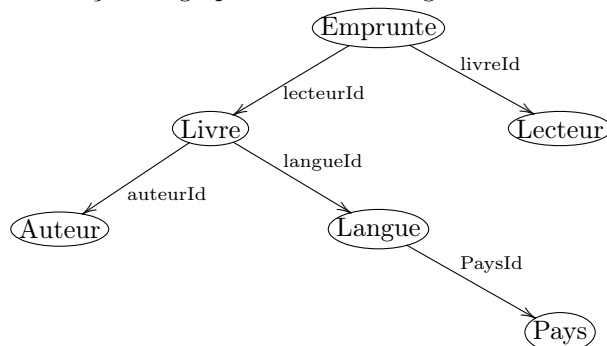
1-* il ya une clé étrangère **Langue** dans **Livre**, et une autre **Auteur** qui référencent les tables de mêmes noms.

- L'association *Emprunté par* devient une table avec deux clés étrangères **Livre,Lecteur** qui référencent les tables de mêmes noms. Sa clé primaire est le triplet **Date,Lecteur,Livre**

Créer une table par attribut multiple et une clé étrangère depuis la table d'origine vers la table créée.

- Livre(LivreId,titre,auteurid#,langueid#)
- Lecteur(LecteurId,Nom,Prénom)
- Emprunt(date,LecteurId#,Livreid#,retour)
- Auteur(AuteurId,nom,prénom)
- Langue(id,langue,PaysId#)
- Pays(id,pays)

Traçons le graphe des clés étrangères



On crée donc d'abord les feuilles (**Auteur,Pays,Lecteur**) puis leurs pères (ici juste **Langue** puis **Livre**) et enfin la racine **Emprunte**.

```

1  -- code pour sqlite
2
3
4  DROP TABLE IF EXISTS Emprunt;
5  DROP TABLE IF EXISTS Livre;
6  DROP TABLE IF EXISTS Langue;
7  DROP TABLE IF EXISTS Pays;
8  DROP TABLE IF EXISTS Auteur;
9  DROP TABLE IF EXISTS Lecteur;
10
11
12  CREATE TABLE Lecteur(LecteurID INT PRIMARY KEY NOT NULL, Nom TEXT,
13  Prénom TEXT);
14
15  CREATE TABLE Auteur(AuteurID INT PRIMARY KEY NOT NULL, Nom TEXT,
16  Prénom TEXT);
17
  
```

```
18
19 CREATE TABLE Pays(PaysID INT PRIMARY KEY NOT NULL, nom TEXT);
20
21 CREATE TABLE Langue(LangueID INT PRIMARY KEY NOT NULL,
22 langue TEXT, PaysID INT, FOREIGN KEY(PaysID) REFERENCES Pays(PaysID) );
23
24 CREATE TABLE Livre(LivreID INT PRIMARY KEY NOT NULL, Titre TEXT,
25 AuteurID INT, LangueID INT, FOREIGN KEY (LangueID) REFERENCES
26 Langue(LangueID), FOREIGN KEY (AuteurID) REFERENCES Auteur(AuteurID));
27
28
29 CREATE TABLE Emprunt(Date TEXT, LecteurID INT, LivreID INT, Retour
30 TEXT, FOREIGN KEY (LivreID) REFERENCES Livre(LivreID), FOREIGN KEY
31 (LecteurID) REFERENCES Lecteur(LecteurID), PRIMARY
32 KEY(Date ,LecteurID ,LivreID));
```

□