

# Chapitre 03 : Bases de données

## Partie 2

---

NSI2

29 août 2023

Niveau logique  
Modèle relationnel

---

On adapte un MCD en tables à deux dimensions.

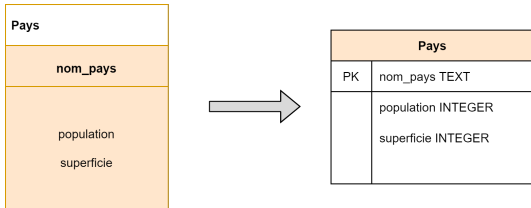
On décide du type des attributs.

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;

# Transformer une entité en relation

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



On indique les types de chaque attribut de la relation.

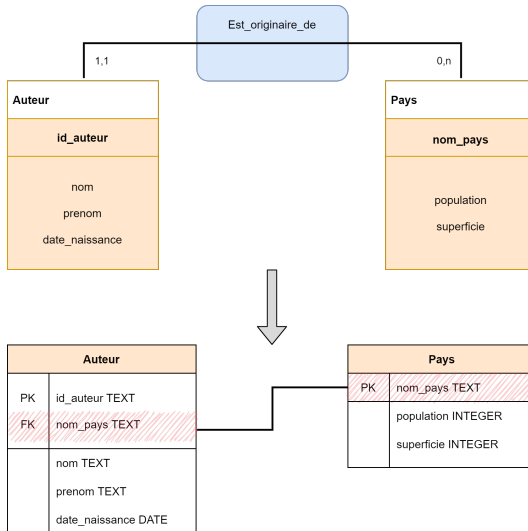
Le ou les identifiants de l'entité sont appelés des **clés primaires** pour la relation : « PK » est l'abréviation de **PRIMARY KEY**.

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)



## Transformer une association en relation : cas (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**.

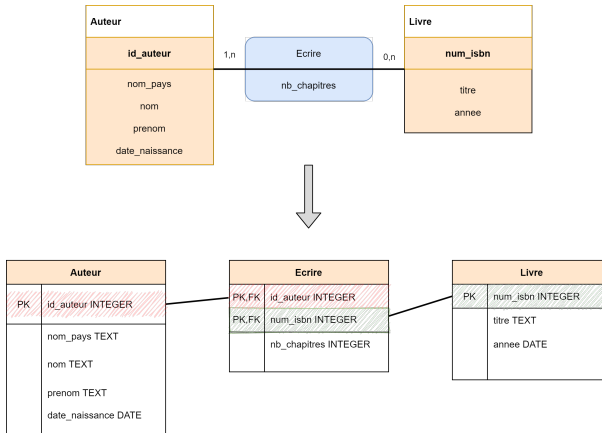
La clé étrangère est soulignée en traits discontinus.

**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)



# Transformer une association en relation : autre cas

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 celaans 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.



**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 TEXTE, date\_naissance DATE)

**Ecrire**(id\_auteur INTEGER, num\_isbn INTEGER, nb\_chapitres INTEGER)

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.

Lors qu'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.

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).

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.