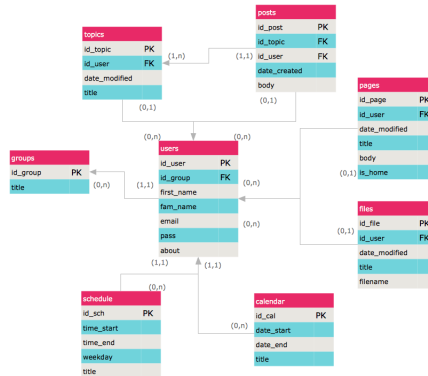


# Bases de données : modèle relationnel



# Historique

Les disques durs ont été inventés en 1956 et a permis d'utiliser les ordinateurs pour collecter, classer et stocker de grandes quantités d'informations.

Le terme database (base de données) est apparu en 1964.

En 1965, Charles Bachman conçoit l'architecture Ansi/Sparc encore utilisée de nos jours. Charles Bachman a reçu le prix Turing en 1973 pour ses «contributions exceptionnelles à la technologie des bases de données».

En 1970, Edgar F. Codd soutient sa thèse qui est à l'origine des bases de données relationnelles. Edgar F. Codd a reçu le prix Turing en 1981.

Le modèle entité-association a été inventé par Peter Chen en 1975 ; il est destiné à clarifier l'organisation des données dans les bases de données relationnelles.

# Quelques exemples

- La collecte de renseignements administratifs : dossiers médicaux, fiscaux.
- Billetterie informatisée : billets de voyage ou de concert.
- Catalogue en ligne comme amazon.com, une des plus grandes bases de données au monde avec plus de 250 millions d'ouvrages catalogués.

# Introduction

La modélisation d'une base de données se réalisent en trois étapes :

# Introduction

La modélisation d'une base de données se réalisent en trois étapes :

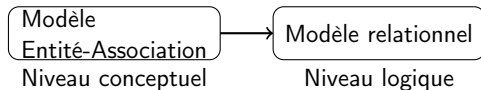
- 1 Niveau conceptuel : représenter la base de données indépendamment de toute considération informatique.

Modèle  
Entité-Association  
Niveau conceptuel

# Introduction

La modélisation d'une base de données se réalisent en trois étapes :

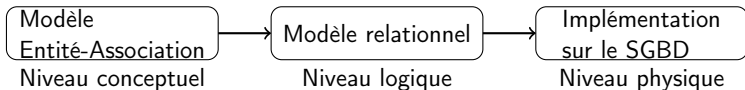
- 1 Niveau conceptuel : représenter la base de données indépendamment de toute considération informatique.
- 2 Niveau logique : adptation du schéma conceptuel en tableaux à deux dimensions.



# Introduction

La modélisation d'une base de données se réalise en trois étapes :

- 1 Niveau conceptuel : représenter la base de données indépendamment de toute considération informatique.
- 2 Niveau logique : adaptation du schéma conceptuel en tableaux à deux dimensions.
- 3 Niveau physique : implémentation informatique sur un SGBD.



# L'emprunt d'un livre au CDI

Chaque élève d'un lycée peut emprunter des livres au CDI, les données concernant ces livres et les emprunts en cours sont stockés dans une base de données.

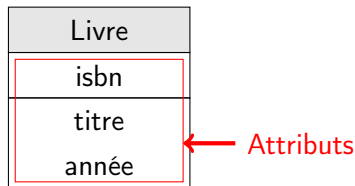
Les **entités** qui constituent cette base sont les *élèves* et les *livres*. Mais on peut en considérer d'autres comme les *auteurs* ou les *éditeurs*.



# L'entité Livre

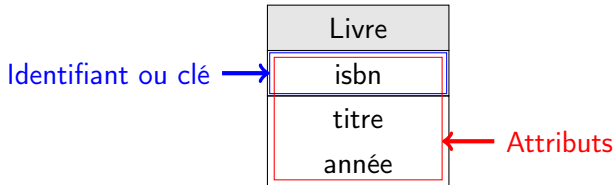
Livre
isbn
titre année

# L'entité Livre



Les attributs sont un ensemble de propriétés qui décrivent l'entité.

# L'entité Livre



Les attributs sont un ensemble de propriétés qui décrivent l'entité. L'identifiant (ou la clé) est un (ou plusieurs) attribut(s) qui permet d'identifier de manière unique l'entité.

# Occurences de l'entité Livre

L'ensemble des livres de la bibliothèque peut être représenté par des *n-upets* (ou tuples).

isbn	titre	année
978-2-07-046614-6	La fête de l'insignifiance	2013
978-2-264-02881-5	Ubik	1969
978-2-13-054387-9	Histoire des sciences	2018

# Les entités Elève et Auteur

Elève
num_etudiant
nom
prenom
classe

# Les entités Elève et Auteur

Elève
num_etudiant
nom
prenom
classe

Auteur
identifiant
nom
prenom

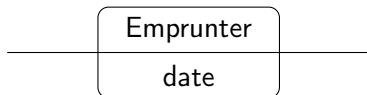
## Occurrences des entités Elève et Auteur

num_etudiant	nom	prénom
1592051067r	Bolognaise	Thomas
2561485589o	Paquito	Enzo
4975322589k	Royal	Henri

identifiant	nom	prénom
1	Kundera	Milan
2	K. Dick	Philip
3	Gingras	Yves

# L'association Emprunter

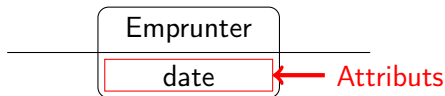
Une **association** est un lien entre des entités.





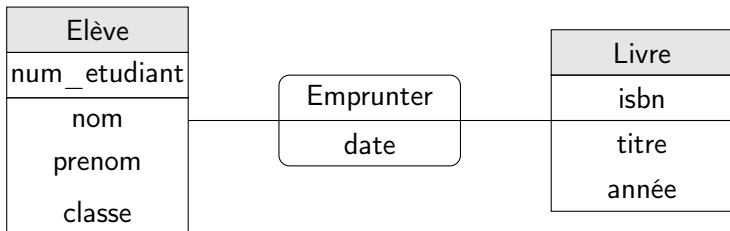
# L'association Emprunter

Une **association** est un lien entre des entités.



# L'association Emprunter

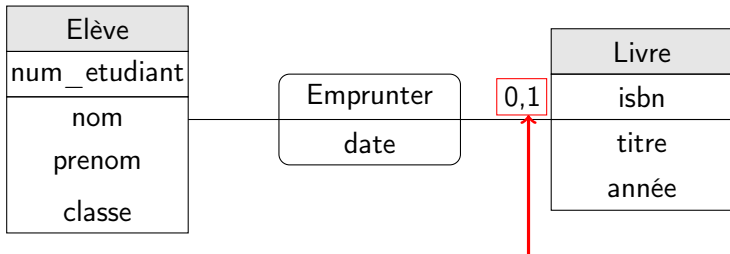
Une **association** est un lien entre des entités.



# L'association Emprunter

Une **association** est un lien entre des entités.

Les cardinalités sont des couples de valeurs (min,max) qui traduisent deux règles de gestion.

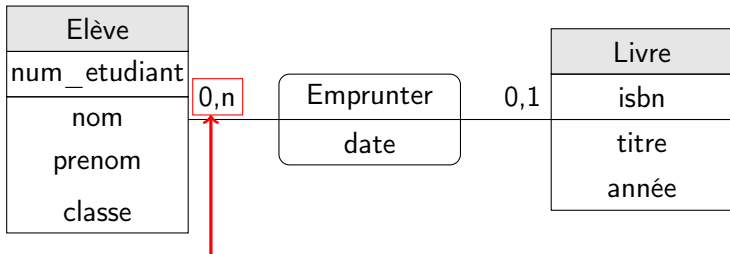


Un livre est emprunté  
par 0 ou 1 personne

# L'association Emprunter

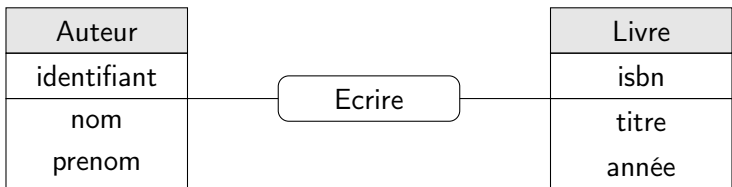
Une **association** est un lien entre des entités.

Les cardinalités sont des couples de valeurs (min,max) qui traduisent deux règles de gestion.



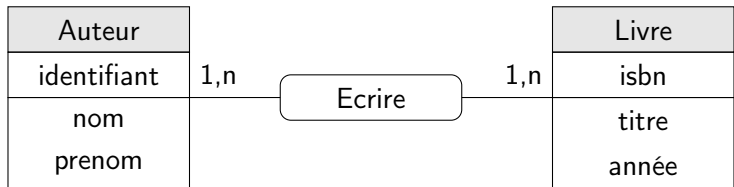
Un élève peut emprunter  
entre 0 et plusieurs livres

# L'association Ecrire



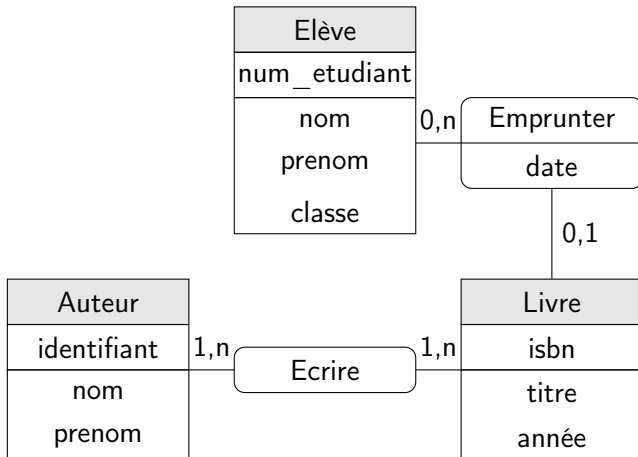
Compléter les cardinalités de l'association *Ecrire*

# L'association Ecrire



- Un auteur écrit entre 1 et plusieurs livres.
- Un livre est écrit par 1 ou plusieurs auteurs.

# Schéma conceptuel



# Exercice

Compléter ce schéma conceptuel avec une entité *Editeur* qui comporte un attribut *siret*, qui l'identifie de manière unique, et un *nom* ainsi qu'une relation *Editer* qui la relie à l'entité *Livre*.



# Transformation d'une entité en relation

Dans le modèle relationnel, les entités et les associations sont transformées en tableau à deux dimensions appelés **relations**.

# Transformation d'une entité en relation

Dans le modèle relationnel, les entités et les associations sont transformées en tableau à deux dimensions appelés **relations**.

Livre
isbn
titre
année

Livre
<u>isbn</u> : TEXT
titre : TEXT
année : INTEGER

← Domaine des attributs

- L'ensemble des valeurs possibles d'un attribut définit un **domaine**.

# Transformation d'une entité en relation

Dans le modèle relationnel, les entités et les associations sont transformées en tableau à deux dimensions appelés **relations**.

Livre
isbn
titre
année

Livre
<u>isbn</u> : TEXT
titre : TEXT
année : INTEGER

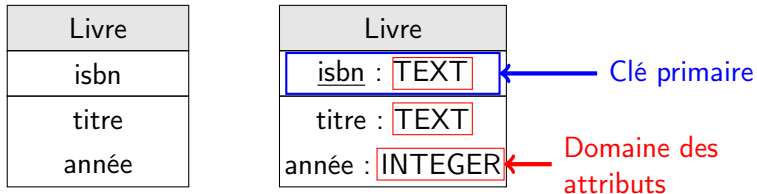
← Clé primaire

← Domaine des attributs

- L'ensemble des valeurs possibles d'un attribut définit un **domaine**.
- L'identifiant de l'entité est la **clé primaire** de la relation.

# Transformation d'une entité en relation

Dans le modèle relationnel, les entités et les associations sont transformées en tableau à deux dimensions appelés **relations**.



- L'ensemble des valeurs possibles d'un attribut définit un **domaine**.
- L'identifiant de l'entité est la **clé primaire** de la relation.

**Notation textuelle :**

Livre(isbn TEXT, titre TEXT, année INTEGER)

# Relation Elève et Auteur

Elève
<u>num etudiant</u> : TEXT
nom : TEXT prénom : TEXT classe : TEXT

Elève(num etudiant TEXT, nom TEXT, prénom TEXT, classe TEXT)

# Relation Elève et Auteur

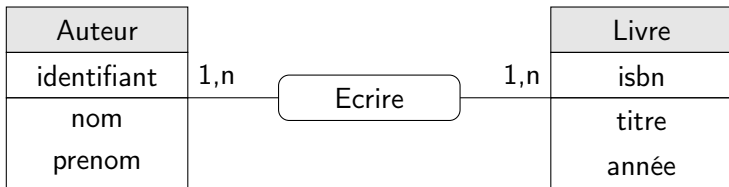
Elève
<u>num etudiant</u> : TEXT
nom : TEXT prénom : TEXT classe : TEXT

Auteur
<u>identifiant</u> : INTEGER
nom : TEXT prenom : TEXT

Elève(num etudiant TEXT, nom TEXT, prénom TEXT, classe TEXT)

Auteur(identifiant INTEGER, nom TEXT, prénom TEXT)

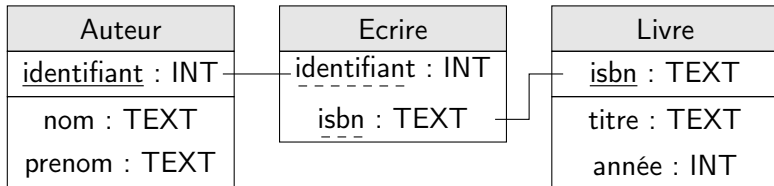
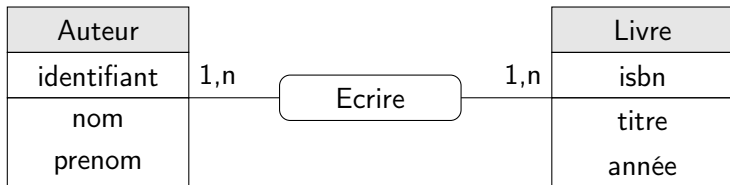
# Transformation d'une association en relation



Auteur
<u>identifiant</u> : INT
nom : TEXT
prenom : TEXT

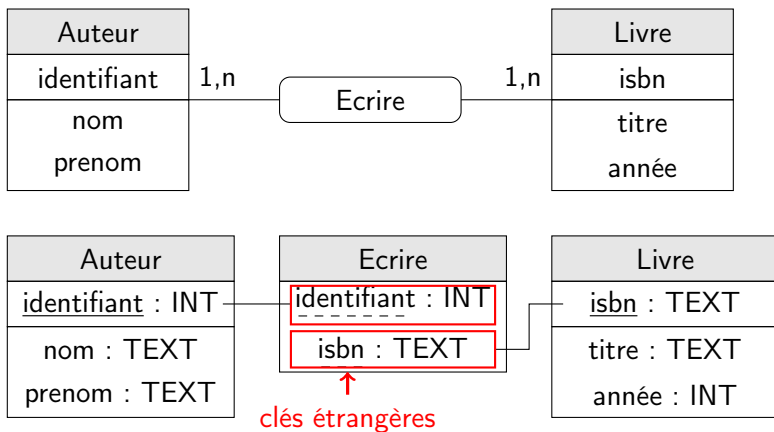
Livre
<u>isbn</u> : TEXT
titre : TEXT
année : INT

## Transformation d'une association en relation

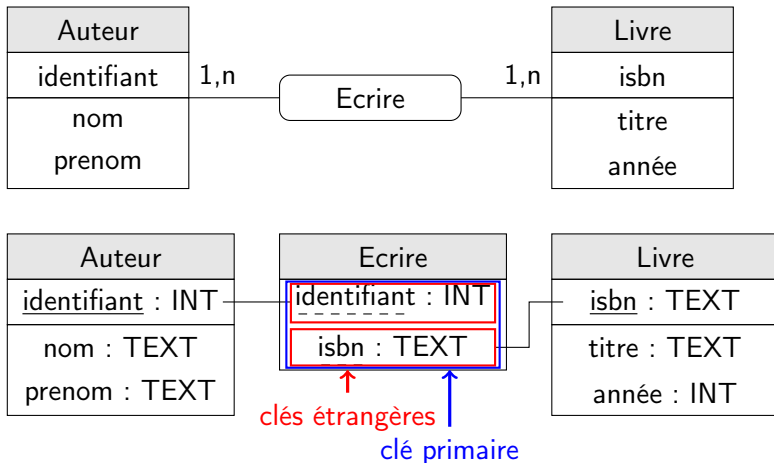




## Transformation d'une association en relation

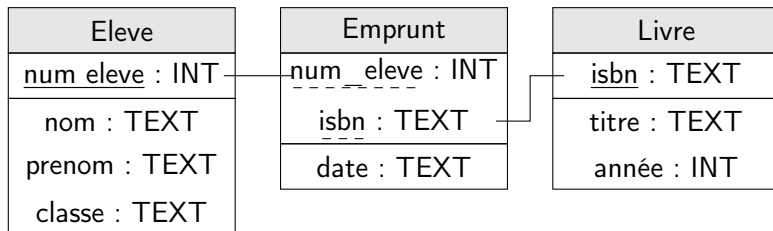


## Transformation d'une association en relation

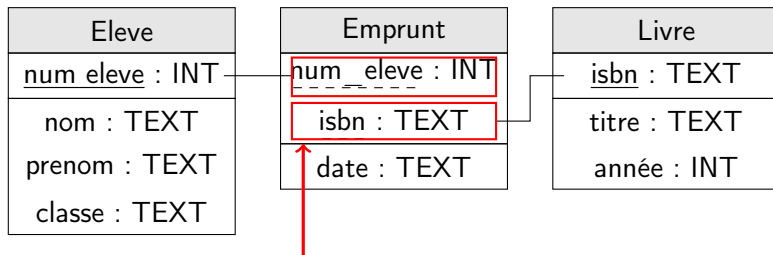


Ecrire(identifiant INT, isbn TEXT)

# La relation Emprunt

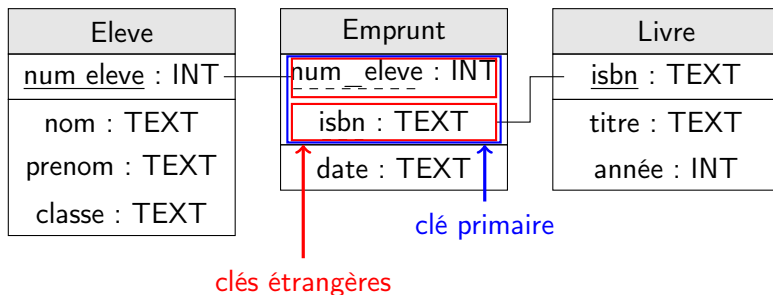


# La relation Emprunt

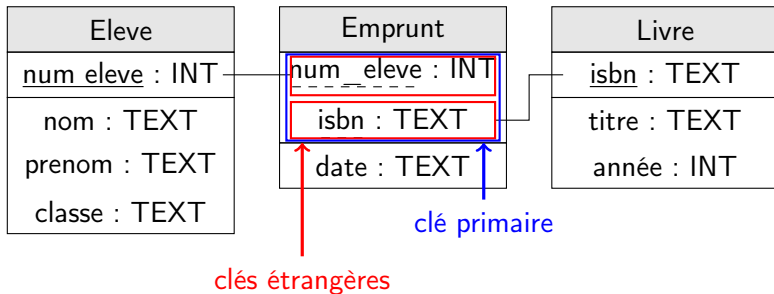


clés étrangères

# La relation Emprunt

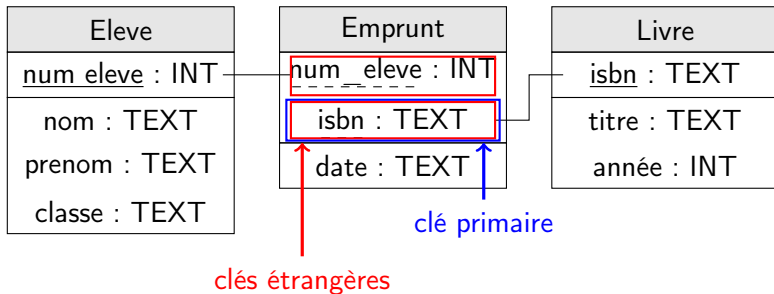


# La relation Emprunt



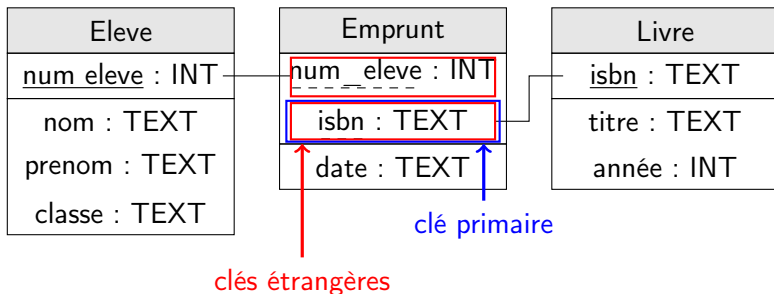
Problème : Un même livre peut être emprunter par deux élèves différents.

# La relation Emprunt



Un même livre ne peut plus apparaître dans deux emprunts différents.

# La relation Emprunt

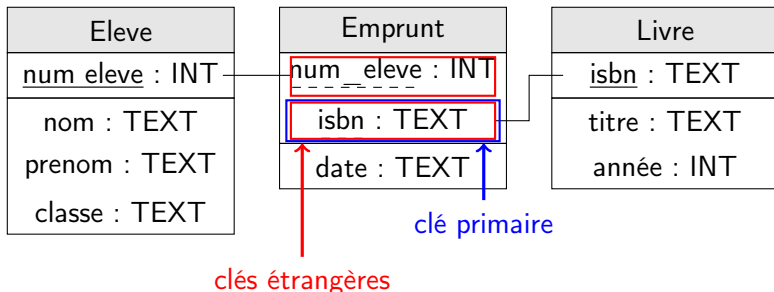


Un même livre ne peut plus apparaître dans deux emprunts différents.

Il faut cependant effacer de la base un emprunt une fois que celui ci est fini.



# La relation Emprunt



Un même livre ne peut plus apparaître dans deux emprunts différents.

Il faut cependant effacer de la base un emprunt une fois que celui ci est fini.

Ecrire(isbn TEXT, num\_eleve TEXT, date TEXT)

# Exercice

Ecrire les relations *Editeur* et *Editer*.

# Contraintes d'intégrité

Une contrainte d'intégrité permet de garantir la cohérence des données lors des mises à jour de la base.

- ❶ **Contrainte d'entité** (de relation) : chaque relation dans le modèle relationnel est identifiée par une clé primaire qui doit être unique et non nulle.

# Contraintes d'intégrité

Une contrainte d'intégrité permet de garantir la cohérence des données lors des mises à jour de la base.

- ❶ **Contrainte d'entité** (de relation) : chaque relation dans le modèle relationnel est identifiée par une clé primaire qui doit être unique et non nulle.
- ❷ **Contrainte de référence** : Une valeur de clé étrangère existe bien en tant que clé primaire dans une autre table.

# Contraintes d'intégrité

Une contrainte d'intégrité permet de garantir la cohérence des données lors des mises à jour de la base.

- ❶ **Contrainte d'entité** (de relation) : chaque relation dans le modèle relationnel est identifiée par une clé primaire qui doit être unique et non nulle.
- ❷ **Contrainte de référence** : Une valeur de clé étrangère existe bien en tant que clé primaire dans une autre table.
- ❸ **Contrainte de domaine** : chaque attribut doit prendre une valeur dans le domaine de valeurs.