

# Modélisation

Introductions aux Bases de Données  
Nathanaël Martel

# ***Trois étapes de modélisation***

- **Conceptuel**

**Organisation des données indépendamment de toute organisation physique.**

- **Logique**

**Restructuration du modèle conceptuel pour une exploitation physique**

- **Physique**

**Faire rentrer le modèle dans le logiciel**

# ***Modélisation conceptuelle***

## **Modélisation conceptuelle**

- **Oublier les logiciels, applications, plateforme, SGBD**
- **Se concentrer sur les données, comment elles sont perçus par les différents personnes utilisant le système**
- **Fournir un diagramme expliquant l'organisation des données ainsi qu'un vocabulaire adapté**

# ***Modélisation conceptuelle***

- **Réunir les entités dont nous disposons**
- **Les décrire**
- **Décrire les relations que les objets ont les uns avec les autres**

# ***Modélisation conceptuelle***

- **Les entités**

**« Un groupes d'objets qui ont les mêmes propriété et qui sont indépendant entre eux »**

**Exemple : les employés, les clients, les produits...**

# ***Modélisation conceptuelle***

## **Représentation des entités sous forme de diagramme :**

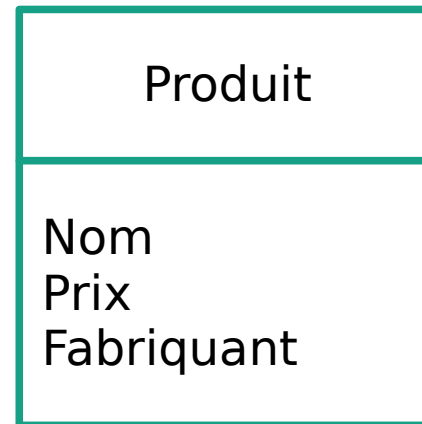
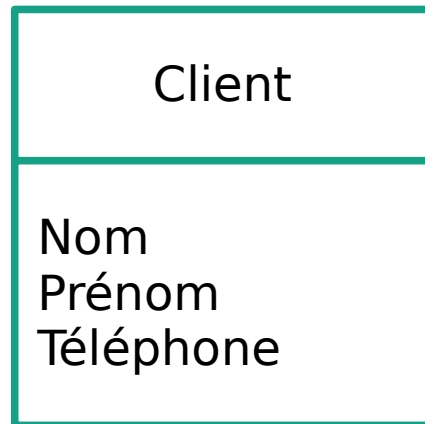
- **Un rectangle avec le nom de l'entité**
- **Le nom de l'entité est un nom commun ou une phrase nominal au singulier**



# Modélisation conceptuelle

## Décrire les entités :

- Préciser les attributs qui compose l'entité



# Modélisation conceptuelle

## Décrire les entités :

- Préciser les types des attributs

Client
Nom <i>texte (max 256)</i> Prénom <i>texte (max 256)</i> Téléphone <i>nombre entier</i> Date de naissance : <i>date</i>

Produit
Nom <i>texte (max 256)</i> Prix <i>nombre décimal</i> Fabriquant <i>texte (max 256)</i>



# Modélisation conceptuelle

## Décrire les entités :

- Clés uniques : identifie de manière unique l'objet
- Clé primaire : identifiant unique principal

Client
Nom Prénom Téléphone <i>attribut unique</i> Date de naissance

Produit
Nom Prix Fabriquant EAN <i>attribut unique</i>

# ***Modélisation conceptuelle***

**Les relations entre les entités :**

- **Les différents liens entre les entités**
- **Nommé par un verbe ou un groupe verbal**
- **Avec une direction qui fait sens**



# ***Modélisation conceptuelle***

## **Les relations entre les entités :**

- **Le nombre d'occurrences minimum et maximum de part et d'autre de la relation**
  - Min = 0, Max = 1 est noté 0..1
  - Le minimum montre la cardinalité
  - Le maximum montre la participation

# Modélisation conceptuelle

## Exemples :

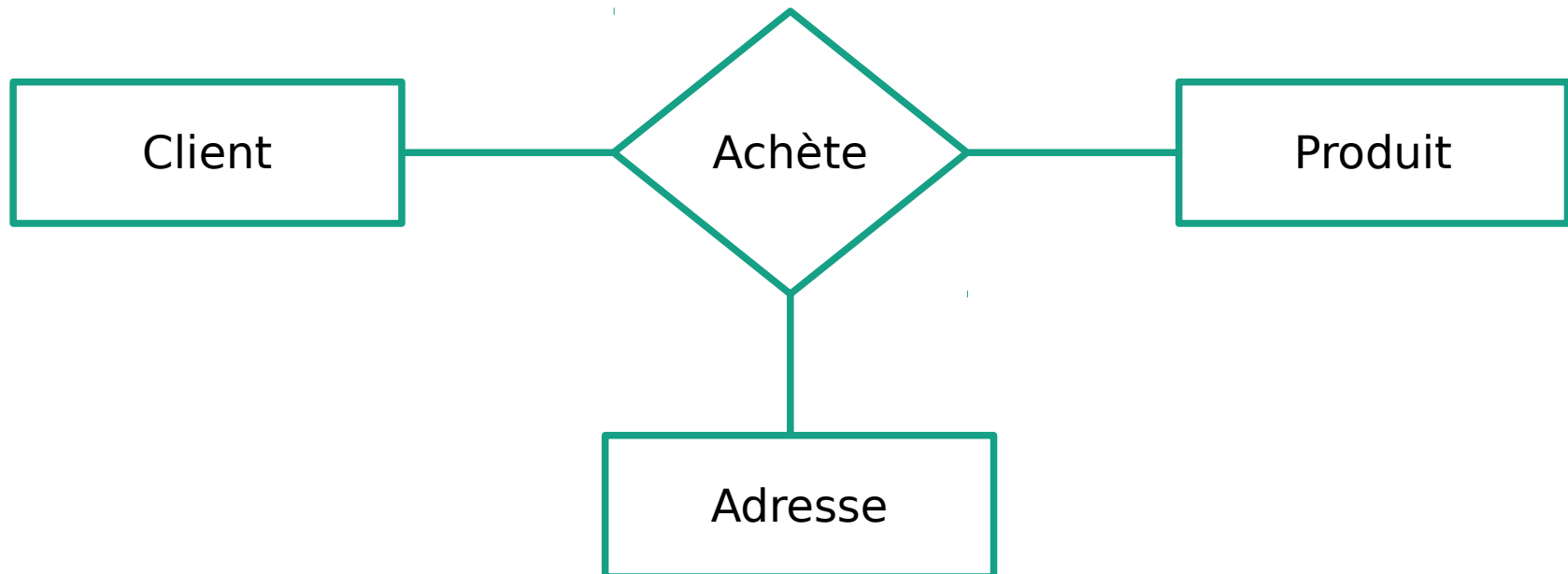
- Relation 0..1 : un ou aucun
- Relation 0..n : plusieurs
- Relation 1..n : au moins un
- Relation 1..1 : un et un seul



# Modélisation conceptuelle

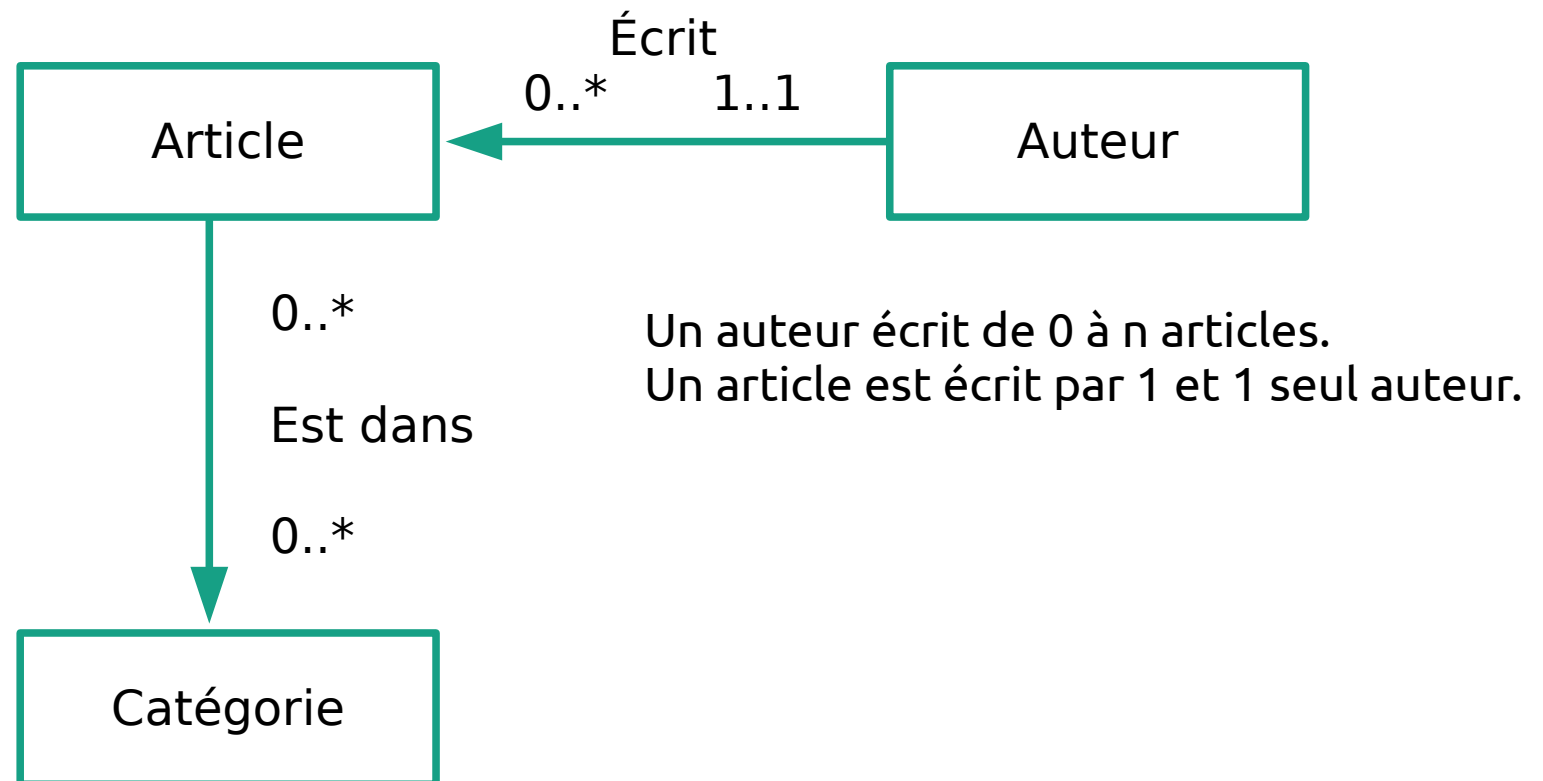
Les relations entre 3 entités :

« Un client achète des produits livrés à une adresse »



# Modélisation conceptuelle

## Exemple de diagramme conceptuel pour un blog



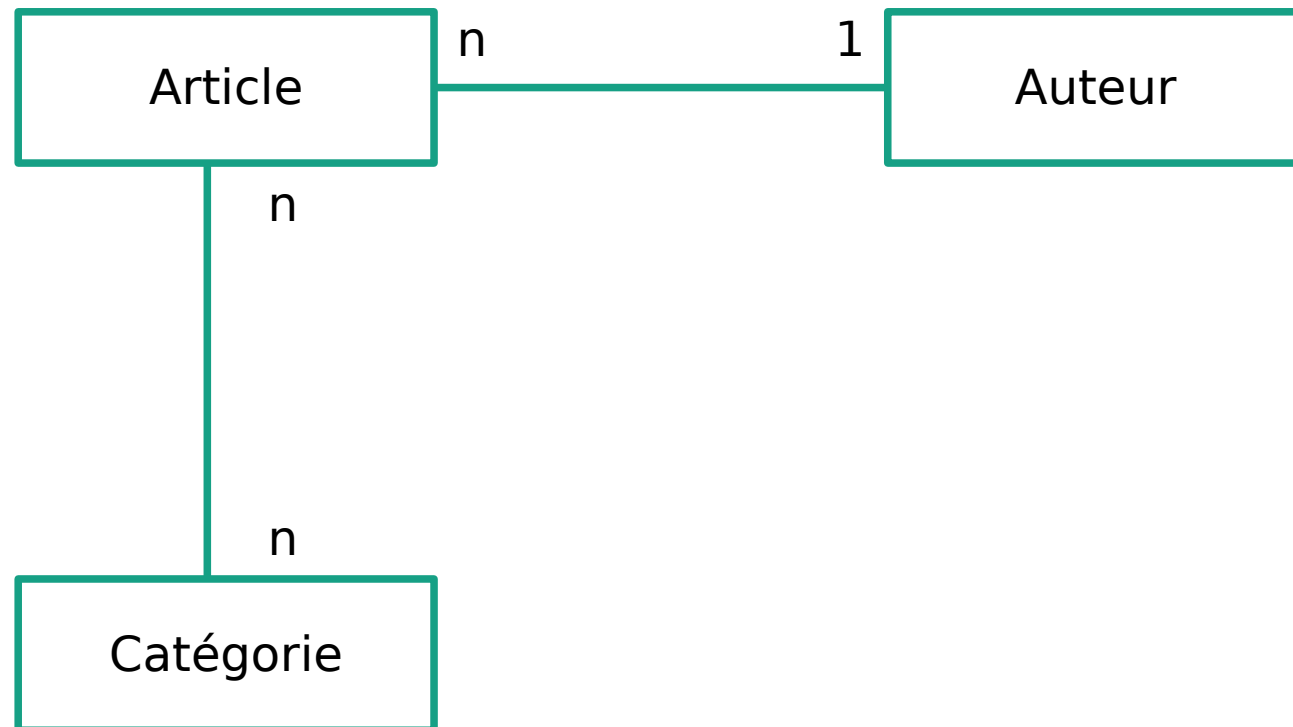
# ***Modélisation conceptuelle***

## **Trois types de relations**

- **One to many «1:\*»**
- **Many to many «\*:»**
- **One to one «1:1»**

# Modélisation conceptuelle

## Exemple de diagramme conceptuel pour un blog





# ***Modélisation conceptuelle***

**Exemple, pour un article donné je peux :**

- **Choisir un et un seul auteur**
  - Liste déroulante `<select>`
  - Boutons radios `<input type="radio" />`
- **Sélectionner plusieurs catégories (ou aucune)**
  - Cases à cocher `<input type="checkbox" />`

# Modélisation conceptuelle

## Relations «One to many»

- 1 auteur peut écrire n articles
- 1 article est écrit par un et un seul auteur



# ***Modélisation conceptuelle***

## **Les étapes de la modélisation conceptuelle**

- **Définir les entités**
- **Définir les attributs des entités**
- **Définir les types des attributs**
- **Identifier les possibles clés unique et primaires**
- **Définir les relations avec leur cardinalité**

# ***Modélisation logique***

## **Modélisation logique**

- **À partir de la modélisation conceptuelle nous déduisons un modèle physique**

# Modélisation logique

## Clé primaire PK

- Pour chaque entité, il faut définir une clé primaire, le plus souvent un identifiant nommé «id» entier qui s'auto-incrémente

### Client

Id PK AI nombre entier  
Nom *texte (max 256)*  
Prénom *texte (max 256)*  
Téléphone *nombre entier*  
Date de naissance : *date*

### Produit

Id PK AI nombre entier  
Nom *texte (max 256)*  
Prix *nombre décimal*  
Fabriquant *texte (max 256)*  
EAN UK

# Modélisation logique

## Relations binaire

- Les relations avec 1 ou 0 indiquent simplement qu'un objet fait (ou peut faire) référence à un autre
- La clé primaire du côté de 1 va être utilisé comme clé étrangère de l'autre côté de la relation



# Modélisation logique

## Relations binaire



# Modélisation logique

## Relations «many to many»

- Une catégorie peut contenir plusieurs articles
- Un articles peut être dans plusieurs catégories





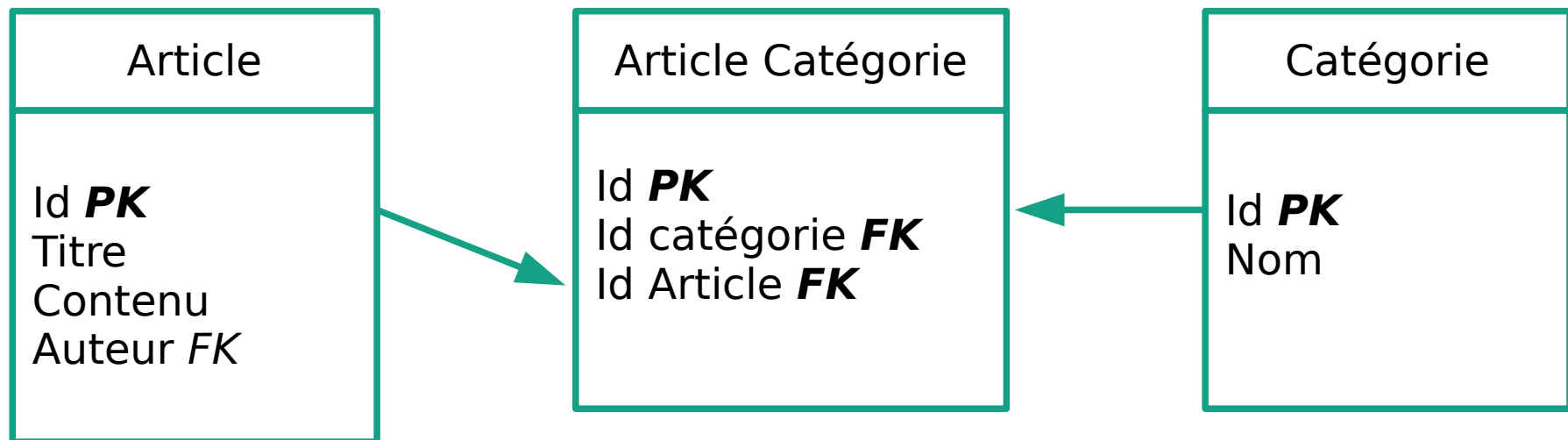
# ***Modélisation logique***

## **Relations multiple**

- **Quand il a deux « n »**
- **Il faut créer une table intermédiaire reprenant les clés primaire des deux tables**
- **La relation peut avoir d'autres attributs.**

# Modélisation logique

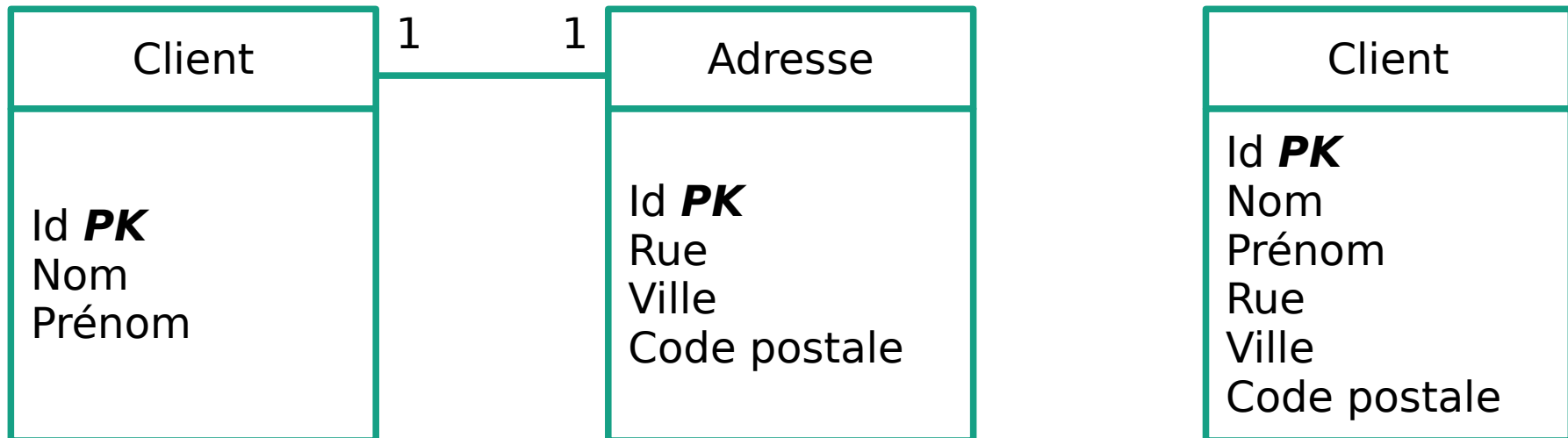
## Relations multiple



# Modélisation logique

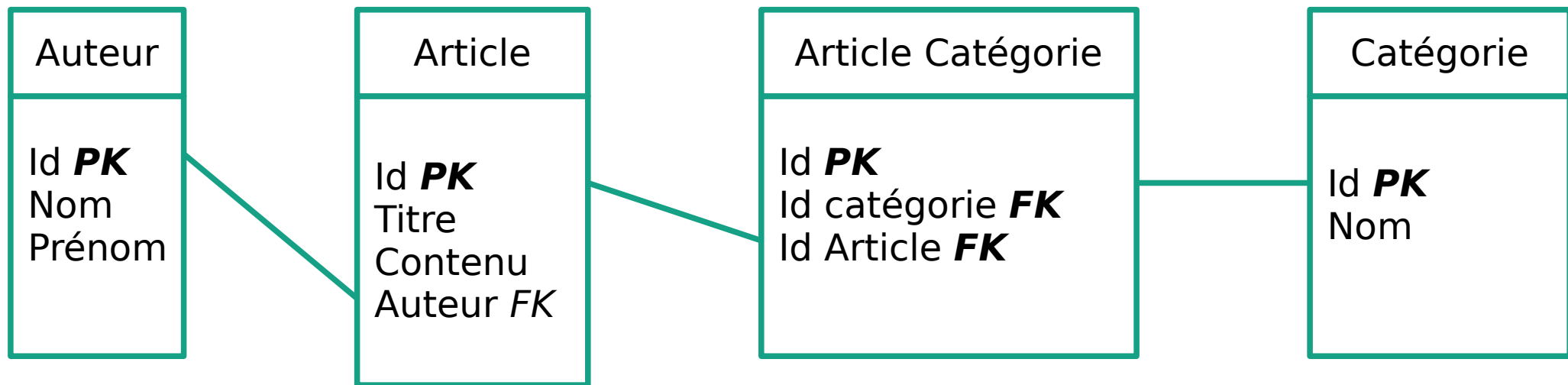
## Relations «one to one»

- Un client a une et une seule adresse
- À part quelque rare cas, les deux entités peuvent fusionner.



# Modélisation logique

## Diagramme logique



# ***Modélisation logique***

## **Les étapes de la modélisation logique**

- **Définir les clés primaire, les ajouter si nécessaire**
- **Transformer les relations en ajoutant si besoins des tables**

# ***Modélisation Physique***

## **Modélisation Physique**

- **À partir de la modélisation logique nous implémentons un système de base de données**
- **Certains logiciel sont capable de faire automatiquement le passage de la modélisation physique à la modélisation logique et dans le sens inverse**

# ***Modélisation Physique***

- **Certaines interfaces proposent des « assistants » pour créer les tables**
- **Sinon il est possible de le faire directement avec SQL**

# *Modélisation Physique*

- **Création de la Base de Données**

```
CREATE DATABASE `bibliotheque`;
```



# Modélisation Physique

- Création de tables avec SQL

```
CREATE TABLE `livre` (  
    `id` int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    `titre` varchar(255) NOT NULL  
);
```

# Modélisation Physique

- **Options génériques des champs**
  - **AUTO\_INCREMENT** : à chaque nouvel enregistrement, la valeur du champs va s'incrémenter de 1 par rapport au précédent (même s'il a été supprimé entre temps). C'est idéal pour s'assurer de l'unicité de la valeur d'un champs et c'est donc parfait pour un clé primaire.
  - **PRIMARY KEY** : indique que ce champs est la clé primaire de la table. Une clé primaire n'est pas nécessairement auto incrémenté.

# Modélisation Physique

- **Options génériques des champs**
  - **DEFAULT** ‘ ’ : valeur par défaut que prend le champs si rien n’est précisé à l’enregistrement.
  - **Null / Not Null** : possibilité ou non que la valeur du champ soit à Null. Attention, Null est différent de 0 ou d’une chaîne vide.

# Modélisation Physique

- **Différents types de champs numérique**
  - **Int** : un nombre entier (longueur)
  - **Tinyint** : un «petit» nombre entier, entre -128 et 128
  - **Decimal, float, double** : d'autres types numérique (option sur le nombre de chiffres)

# Modélisation Physique

- **Différents types de champs dates**
  - **Date** : une date dans le format de la SGBD
  - **Datetime** : une date avec l'heure dans le format de la SGBD
  - **Timestamp** : date et heure au format Unix, c'est à dire le nombre entier de seconde depuis le 01/01/1970

# Modélisation Physique

- **Différents types de champs textuels**
  - **Varchar** : une chaîne de caractère (longueur)
  - **Text** : un chaîne de texte de longueur indéfinis (et potentiellement infinis)

# Modélisation Physique

- ... et pleins d'autres types
  - **Enum** : un choix dans une liste de propositions (proposition). En fait, cela revient à faire un tinyint.
  - **Bit**, **Binary**, **Blob** : des données sous forme binaire.
  - ...

# Modélisation Physique

- Modification de tables avec SQL

```
ALTER TABLE `livre`
```

```
ADD `auteur` varchar(255) COLLATE 'latin1_swedish_ci'  
NOT NULL DEFAULT 'Victor Hugo',
```

```
ADD `date_achat` date NULL AFTER `auteur`;
```



# Modélisation Physique

- Autres exemple

```
CREATE TABLE `auteur` (  
  `id` int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `nom` varchar(255) NOT NULL,  
  `prenom` varchar(255) NOT NULL,  
  `date_naissance` date NULL,  
  `date_mort` date NULL  
);
```

# Modélisation Physique

- **Clés étrangères**

- C'est un champs comme un autre, il doit être déclaré comme les autres.
- Il n'est pas nécessaire de signaler à la base de données que c'est une clé étrangère : il faudra toujours préciser sur quoi ce fait la requête dans les jointures
- Il est possible de demander à la base de données de gérer la cohérence des clés étrangère avec une "contrainte"

```
CREATE TABLE `auteur` (  
  `id` int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `nom` varchar(255) NOT NULL,  
  `prenom` varchar(255) NOT NULL,  
  `date_naissance` date NULL,  
  `date_mort` date NULL  
);
```

# Conclusion

La modélisation d'une base de données passe par une phase conceptuelle puis logique.

Ces phases permettent d'obtenir un diagramme qui sert de documentation ainsi qu'une structure de travail.