



# Bases de Données Relationnelles

Telecom Nancy 1ère année

Adrien Coulet et Malika Smaïl-Tabbone

[coulet@loria.fr](mailto:coulet@loria.fr)



# Bases de Données Relationnelles

Telecom Nancy 1ère année

Adrien Coulet et Malika Smaïl-Tabbone

coulet@loria.fr

# Plan du cours (1/3)

## Chap. 1 : Introduction

- Limites des systèmes de fichiers
- Notion de BD et de SGBD
- Architecture et fonctions d'un SGBD
- Processus de conception d'une BD

## Chap. 2 : Modèle Entité-Association

- Concepts de base du modèle E/A
- Règles de complétude d'un schéma E/A

# Plan du cours (2/3)

## Chap. 3 : Modèle relationnel de données

- Concepts du modèle relationnel
- Passage du modèle entité-association au relationnel
- Redondance et normalisation
- Langages de manipulation des données (LMD)

## Chap. 4 : Le langage SQL (*SGBD ORACLE*)

- Définition/Mise à jour/Interrogation des données
- Les vues
- Catalogue du système ORACLE (dictionnaire)

# Plan du cours (3/3)

## Chap. 5 : étendre SQL : PL/SQL et les triggers

PL/SQL : extension procédurale de SQL

- variables
- structures de contrôle
- curseurs
- procédures et fonctions

## Chap. 6 : Les transactions

# Mini-bibliographie : BD relationnelles et systèmes d'information

1. « Fundamentals of database systems » R. Elmasri et S. B. Navathe, troisième édition, The Benjamin/Cummings Pub., 2000.
2. « Bases de données et systèmes d'information » N. Boudjlida, Dunod, 1999.
3. « Apprendre et pratiquer MERISE » J. Gabay, Masson, 1989.
4. « Bases de données : objet et relationnel » G. Gardarin, Eyrolles, 1999.
4. « Bases de données : les systèmes et leurs langages » G. Gardarin, Eyrolles, 1984.

# Plan du cours

## 1. Introduction

2. Modèle conceptuel de données Entité-Association

3. Modèle relationnel de données

4. Le langage SQL

5. PL/SQL

6. Transactions

# Chap. 1 : Introduction

## Limites des systèmes de gestion de fichiers

- Chaque application dispose de ses fichiers (données)
  - ⇒ redondance des données
  - ⇒ complexité du maintien de la cohérence
- Programmes dépendants de l'organisation physique des données
- Pas de gestion des accès multiples
  - d'où l'apparition de la notion de **base de données**  
(début des années 60)

# Notion de BD et de SGBD

- **Base de données (BD)** : collection importante de données
  - ✓ cohérentes
  - ✓ structurées indépendamment d'une application
  - ✓ de redondance minimale
  - ✓ accessibles par plusieurs utilisateurs à la fois
- **Système de Gestion de Bases de données (SGBD)** : logiciel assurant structuration, stockage, maintenance, mise à jour et consultation d'une BD

# Quelques applications des BDs

- Gestion (salaires, factures, stock...)
- Transactionnel (comptes bancaires, centrales d'achat...)
- Librairies électroniques (bibliothèques, journaux...)
- Documentation technique (plans, dessins...)
- Génie logiciel (programmes, manuels...)
- ...

# Objectifs d'un SGBD

## Indépendance données-programmes

- **Indépendance physique** : un changement de l'organisation physique des données n'entraîne pas de changement dans les programmes (traitements)
- **Indépendance logique** : un changement de l'organisation logique des données (ex. nouvelle rubrique) n'entraîne pas de changement dans les programmes non concernés.

# Objectifs d'un SGBD

- **Manipulation aisée des données** : par des utilisateurs non informaticiens (interrogation et mise à jour)
- **Administration aisée des données** : un SGBD doit fournir des outils pour définir les données et autoriser leur évolution (tâche de l'*administrateur*)
- **Partage des données** : utilisation simultanée par différentes applications
- **Efficacité des accès aux données** : garantir un bon *débit* (nombre de transactions exécutées par seconde) et un bon *temps de réponse* (temps d'attente moyen par transaction)

# Fonctions d'un SGBD

## a. Définition des données

Un Langage de Définition de Données (LDD) permet de décrire :

- des **objets** (*personnes, véhicules*)
- Des **attributs** sur les objets (*nom, n° d'immatr.*)
- Des **liens** entre objets (*personne possède véhicule*)
- Des **contraintes** sur objets, attributs et liens (*un véhicule n'a qu'un seul propriétaire*)

**Schéma** = description d'une BD à l'aide du LDD

# Fonctions d'un SGBD

## b. Manipulation des données

- Un Langage de Manipulation de Données (LMD) permet de *créer, rechercher, supprimer, modifier* des données
- Un SGBD doit offrir plusieurs interfaces d'accès, correspondant à différents types d'utilisateurs
  - ✓ interface orientée utilisateur final  
(*langage de requêtes comme SQL avec mise en oeuvre graphique, interface de type formulaire, ...*)
  - ✓ interface orientée programmeur d'applications  
(*SQL immergé dans C ou Java*).

# Fonctions d'un SGBD

## c- Persistance des données, gestion de la mémoire et accès aux données

- Durée de vie des données > à celle des programmes qui les ont générées
- Volume des données >> capacité de la MC (mémoire centrale)
- Volume de données transférées (bloc) > capacité de la MC
- Gestion de cache pour un accès efficace aux données sur le disque (index, hachage...)
  - Techniques d'optimisation de requêtes

# Fonctions d'un SGBD

## d. Contrôler l'intégrité des données

Les données enregistrées doivent vérifier certaines propriétés appelées **contraintes d'intégrité (CI)**

- ✓ CI exprimées dans le schéma (définition) de la BD
  - ➔ le SGBD doit les préserver durant la vie de la BD
- ✓ CI plus complexes
  - ➔ à coder dans des programmes d'application (cf. PL/SQL)

# Fonctions d'un SGBD

## e. Assurer la sécurité de fonctionnement

### ■ Mécanisme de **transaction**

- Séquence d'opérations faisant passer une BD d'un état cohérent à un nouvel état cohérent
- Une transaction est exécutée complètement avec succès ou est annulée.

### ■ Remettre rapidement une BD dans un état opérationnel après un incident hardware/software

1. **Journalisation** des opérations réalisées sur la BD  
(dans le *journal* ou *LOG*)
2. **Ré-exécution** automatique en cas d'incident

# Fonctions d'un SGBD

## f. Gérer les accès concurrents

- Permettre des accès simultanés par plusieurs utilisateurs
- Mécanismes de gestion de conflits d'accès
  - **autorisation** d'accès multiples en consultation  
(lecture)
  - **verrouillage** en cas d'accès en modification  
(écriture)

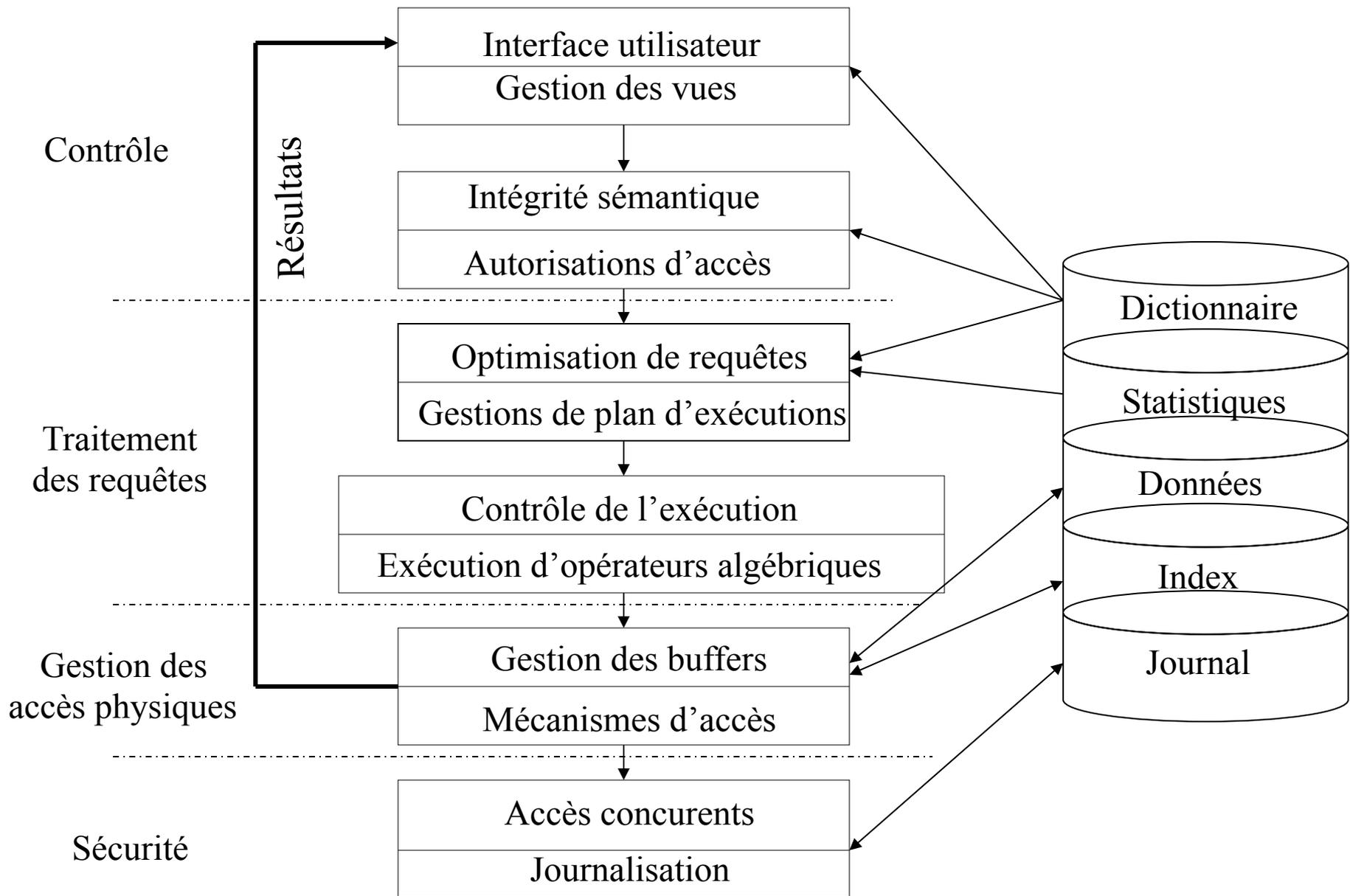
# Fonctions d'un SGBD

## **g. Assurer la confidentialité**

Cacher certaines informations à certains utilisateurs

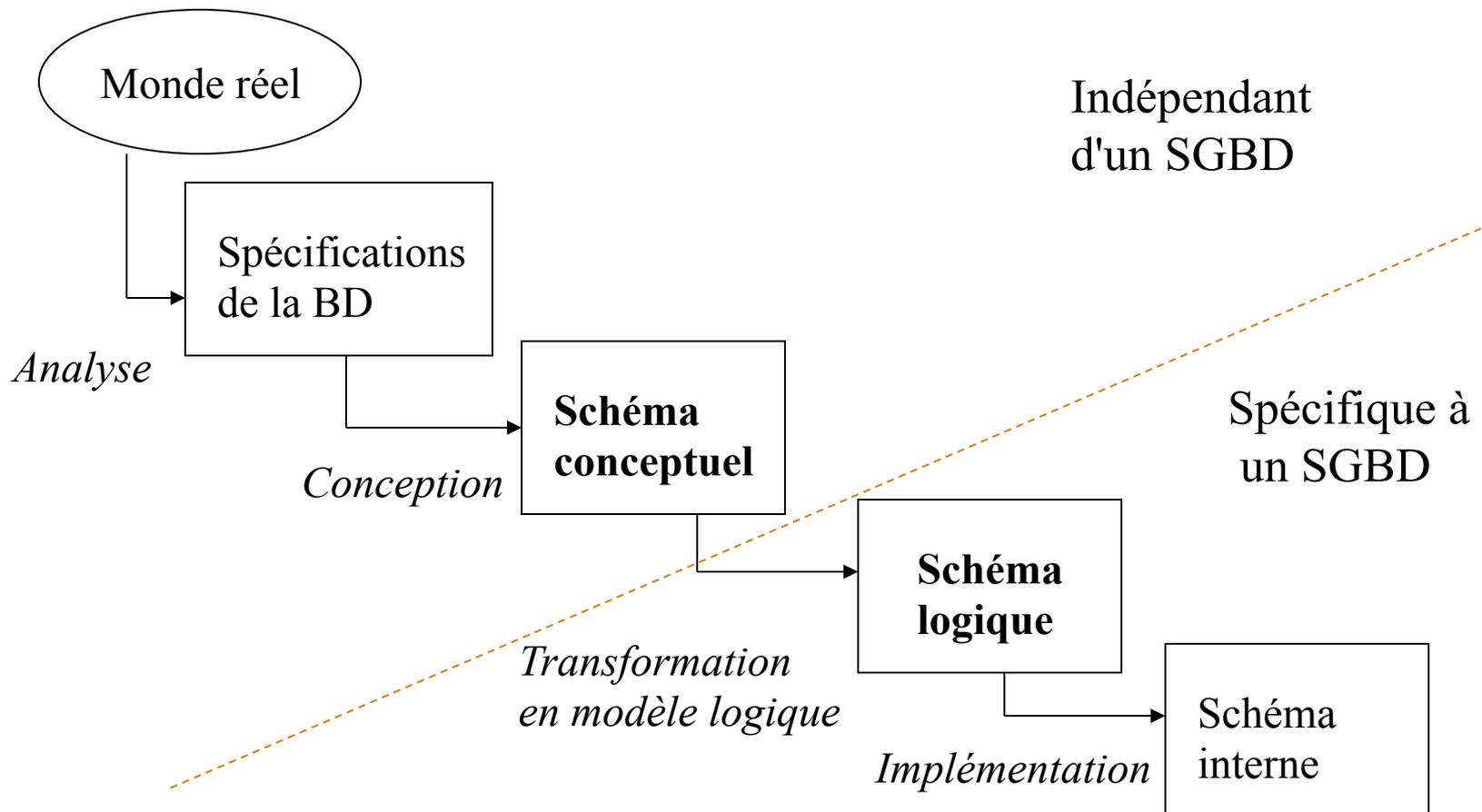
Mise en œuvre :

- **mots de passe** et privilèges d'accès
- utilisation de sous-schémas



**Architecture fonctionnelle d'un SGBD [Boudjlida 03]**

# Processus de conception d'une base de données



# Quelques mots sur la conception de BD

- Processus de conception : quasi **impossible** à formaliser  
(nécessite pratique, expérience et ...flair)
- Il existe de nombreuses méthodes de conception  
(fonctionnelles, orientées objet, orientées système)
- Un SI comporte deux aspects **interdépendants** :
  - ✓ Partie statique : objets structurés et leur relations  
(données, cf. SQL)
  - ✓ Partie dynamique : activité du système  
(traitements opérant sur les données, cf. PL/SQL)

# Plan du cours

1. Introduction
- 2. Modèle conceptuel de données Entité-Association**
3. Modèle relationnel de données
4. Le langage SQL
5. PL/SQL
6. Transactions

# Chap. 2 : Modèle Entité-association (E/A)

- Proposé par Chen en 1976
- Modèle **conceptuel** abstrait très utilisé pour décrire les données d'un SI
- Dans ce cours :
  - ↔ description abstraite d'un domaine à travers les données mises en jeu
  - ↔ mais pas de modélisation des traitements qui affecteront ces données (se sera pour la 2A : MOCI)
- *Entité-Association* (E/A) traduction de *Entity/Relationship* (E/R)
- Une représentation graphique rigoureuse est associée au modèle E/A

## 2.1. Concepts de base du modèle E/A

### a. Entités et attributs

- **Entité** : objet concret ou abstrait à propos duquel on souhaite conserver des informations.

ex. *client, journal, département, inscription*

- **Attribut** : propriété attachée à une entité.

ex. *nom (d'un client), prix (d'un journal)*

- ✓ Attribut **atomique** / **composé** (subdivisé en attributs)

ex. *nom* versus *adresse* subdivisé en *rue, ville, CP*

- ✓ Attribut **dérivé** : calculé à partir d'autre(s) attribut(s)

ex. *âge* calculé à partir de *date\_de\_naissance*

- ✓ Attribut **monovalué** / **multivalué** (plusieurs valeurs par occurrence d'entité)

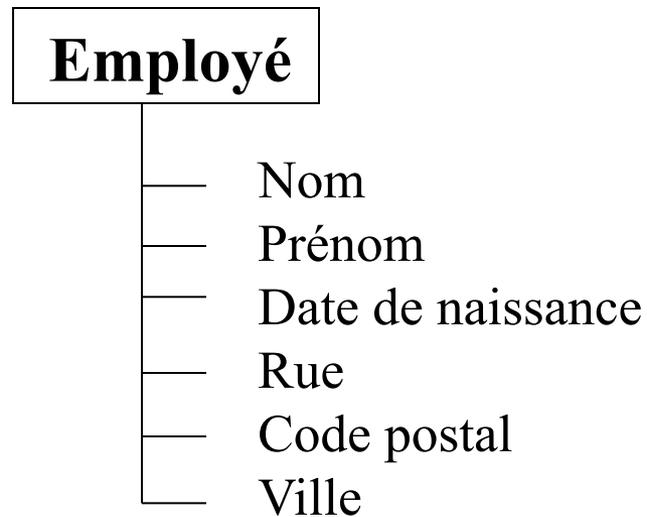
ex. *nom* versus *films préférés*

## a. Entités et attributs

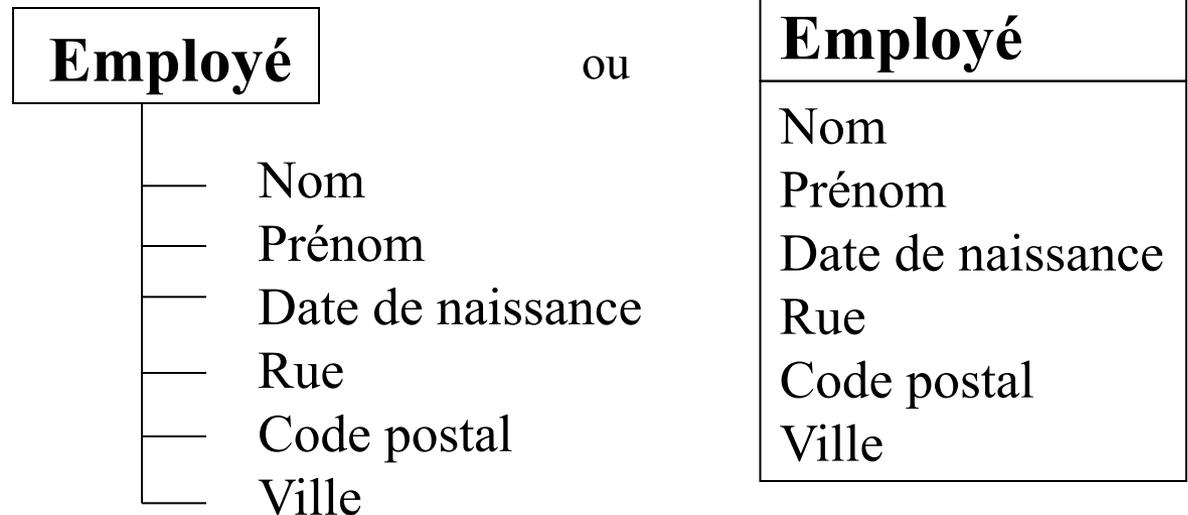
- **Domaine d'un attribut** : ensemble des valeurs que peut prendre un attribut.

ex.    prix (d'un journal) : *nombre réel positif*  
      nom (d'un client) : *chaîne de caractères*  
      *alphabétiques*  
      couleur : {*rouge, vert, bleu, jaune*}

# Représentation graphique d'une entité avec ses attributs



# Représentation graphique d'une entité avec ses attributs



## b. Type et occurrences d'entité

- **Type d'entité** : classe de toutes les entités de même nature et jouant le même rôle.

ex. *Employé*

- **Occurrence d'un type d'entité** : tout individu particulier faisant partie du type d'entité.

ex. Occurrences du type d'entité *Employé* : *les employés*

Albus Dumbeldor

Marie Dupont

Philippe Poutou

...

## b. Type et occurrences d'attribut

- **Type d'attribut** : nom d'un attribut et domaine des valeurs possibles

ex.

Nom et Prénom : *chaînes de caractères*

Date de naissance : *date*

Âge : *entier positif*

- **Occurrence d'attribut** : valeur particulière

ex1. *Philippe* : occurrence de l'attribut Prénom,

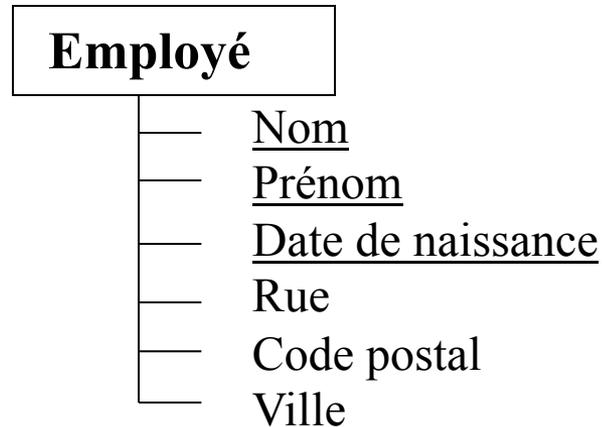
*Pontou* : occurrence de l'attribut Nom

ex2. *bleu, rouge, jaune* : occurrences de l'attribut couleur

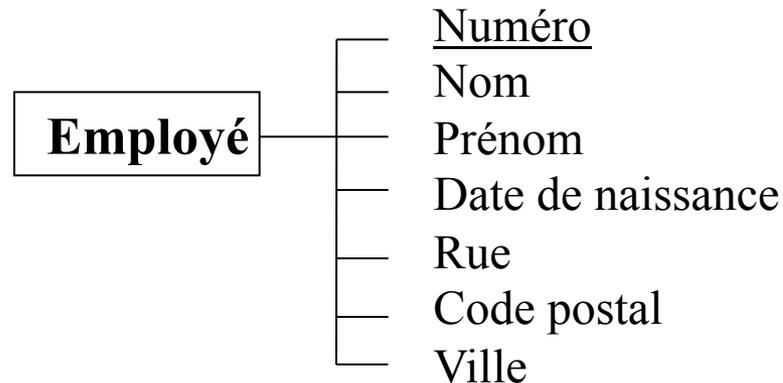
## c. Identifiant d'entité

- **Identifiant d'entité** : ensemble minimal d'attributs caractérisant de façon **unique** chaque occurrences d'un type d'entité.
- Représentation graphique : attribut(s) identifiant(s) souligné(s)

ex. Un identifiant du type d'entité Employé : {Nom, Prénom, Date de naiss.}



Un autre identifiant du type d'entité Employé : un nouvel attribut (artificiel) **Numéro**



## e. Associations

- **Association** : liaison existant entre des entités.  
ex. les clients *commandent* des produits

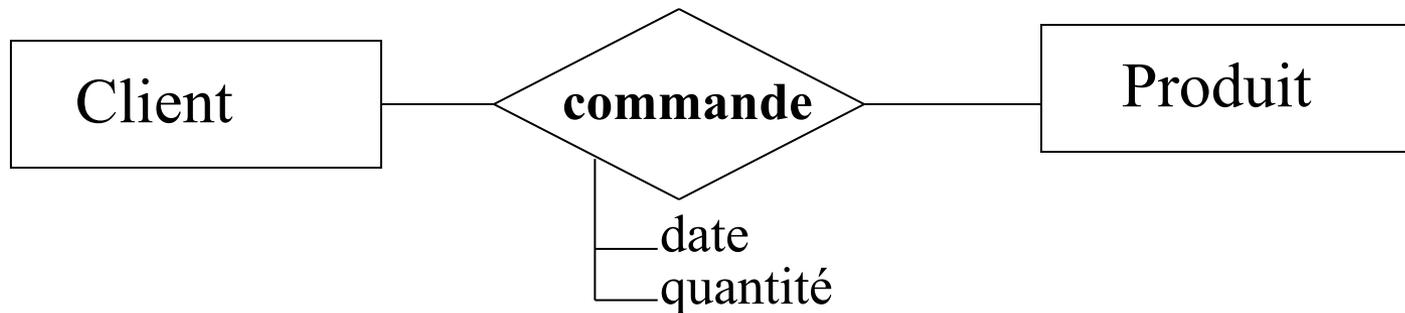


- Chaque entité joue un **rôle** dans l'association
  - Les rôles devront être précisés si l'association relie une entité à elle-même.
- Une association peut concerner plus de deux entités
  - degré d'une association (association binaire, ternaire, n-aire)

## e. Attributs d'associations

- Une association peut être caractérisée par des **attributs**.

ex. *date* de la commande et *quantité* de produits commandés.

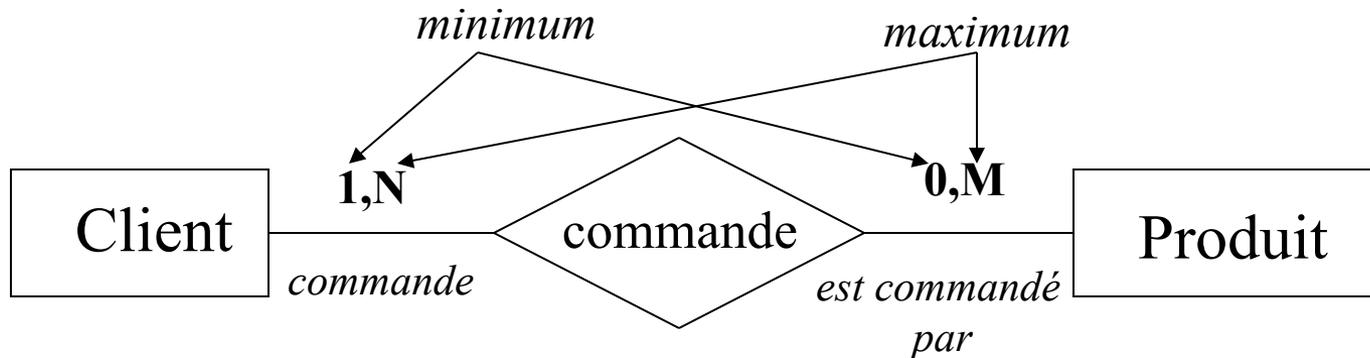


- **Occurrence d'un type d'association** : occurrence de chaque type d'entité + occurrence de chaque attribut de l'association

## e. Cardinalités d'associations

- **Cardinalité d'une assoc.** : nombres minimum et maximum de participations de chaque occurrence d'entité à l'association.

*ex. un client doit commander au moins un produit ; un produit peut être commandé par zéro ou un nombre quelconque de clients.*

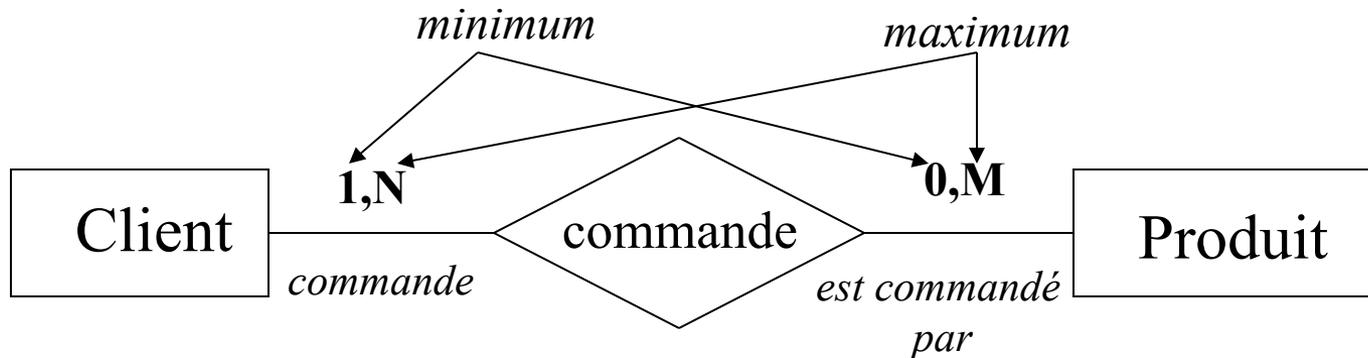


En fonction des cardinalités **maximales**, une association binaire (degré = 2) peut être de type 1-1, 1-N ou N-M

## e. Cardinalités d'associations

- **Cardinalité d'une assoc.** : nombres minimum et maximum de participations de chaque occurrence d'entité à l'association.

*ex. un client doit commander au moins un produit ; un produit peut être commandé par zéro ou un nombre quelconque de clients.*



En fonction des cardinalités **maximales**, une association binaire (degré = 2) peut être de type 1-1, 1-N ou N-M

## e. Cardinalités d'associations

### Association 1-1 :

Une association A entre deux entités E et F est de cardinalité 1-1 si une occurrence de E ne peut être associée **au maximum** qu'à une occurrence de F et réciproquement.



*Un client est titulaire d'au maximum un compte; un compte appartient à un client et un seul.*

## e. Cardinalités d'associations

### Association 1-N :

Une association  $A$  entre deux entités  $E$  et  $F$  est de cardinalité 1-N si une occurrence de  $F$  peut être associée à plusieurs occurrences de  $E$  et qu'une occ. de  $E$  ne peut être associée (au max.) qu'à une occ. de  $F$ .  
(ou l'inverse !)



*Un produit est fourni par un fournisseur et un seul;  
un fournisseur peut fournir un nombre qq de produits.*

## e. Cardinalités d'associations

### Association N-M

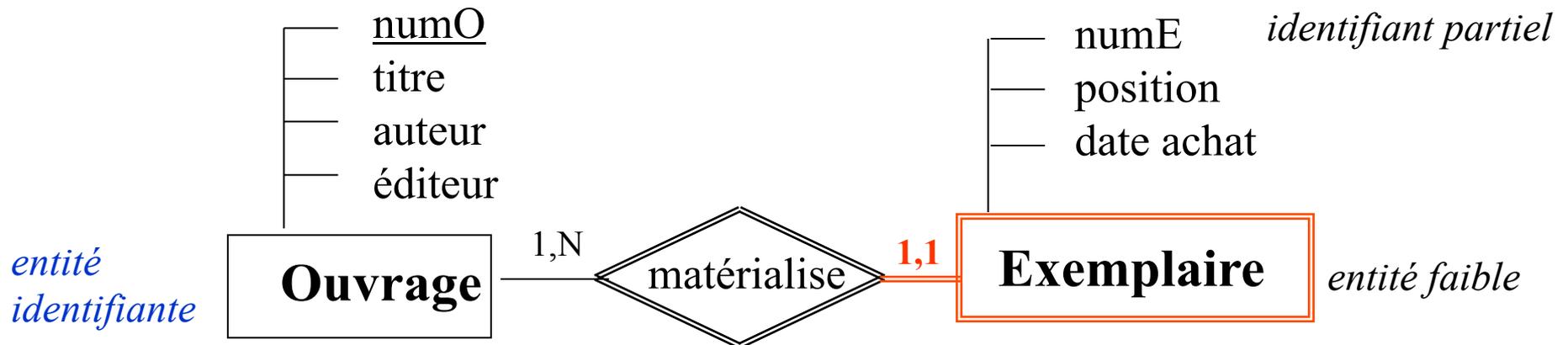
Une association  $A$  entre deux entités  $E$  et  $F$  est de cardinalité N-M si une occurrence de  $E$  peut être associée à plusieurs occurrences de  $F$  et réciproquement.



*Un client peut commander plusieurs produits;  
un produit peut être commandé par plusieurs clients.*

## d. Entité faible

- Entité faible = entité sans identifiant propre
- Une entité faible n'existe qu'en référence à une autre entité dite **entité identifiante**
  - l'association qui les unit est dite **association identifiante**
  - l'entité faible a une contrainte de cardinalité 1,1 sur son association identifiante



# f. Contraintes d'intégrité

**But** : Spécifier des contraintes de la réalité perçue qui ne sont pas exprimables dans le modèle E/A.

■ **Contrainte d'intégrité (CI)** : propriété que doivent satisfaire les données.

– **CI statique** : doit être vérifiée à tout moment.

ex1 : *date-de-mariage* > *date-de-naissance*

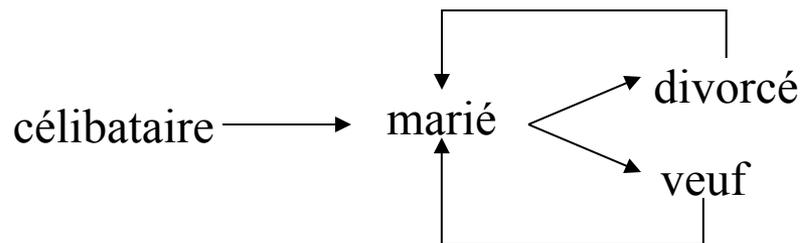
ex2 : *nom-de-jeune-fille*, *numero-portable* : attributs facultatifs

ex3 : *nom*, *no-sécurité-sociale* : attributs obligatoires

– **CI dynamique** : tout changement d'état de la BD doit la respecter.

ex1 : *l'attribut salaire ne peut que croître.*

ex2 : *le changement d'état civil doit respecter le graphe suivant :*



## 2.2. Règles de complétude d'un schéma E/A

Pour qu'un schéma E/A soit complet, chaque classe d'objets (entité, association, attribut, CI) doit posséder toutes les propriétés requises par le modèle E/A :

**Pour une entité :**

- ✓ un nom
- ✓ une définition
- ✓ une liste d'attributs
- ✓ au moins un identifiant

*Exemple*

*Auteur*

*"Un auteur a écrit au moins un livre édité par un éditeur qui nous intéresse."*

*{ nom-aut, prénom-aut, ville-aut }*

*{ nom-aut, prénom-aut }*

## 2.2. Règles de complétude d'un schéma E/A

Pour une association :

*Exemple*

- ✓ un nom
- ✓ une définition
- ✓ la liste des entités participantes
- ✓ éventuellement, les rôles et leur cardinalité
- ✓ la liste des attributs

*Écriture*

*"L'association écriture associe un auteur aux livres qu'il a écrits"*

*Auteur, Livre*

*Auteur écrit (1, N) Livre*

*Livre est écrit par (1,1) Auteur*

## 2.2. Règles de complétude d'un schéma E/A

Pour un attribut :

*Exemple*

✓ un nom

*ville-aut*

✓ une définition

*"Nom de la ville dans laquelle  
réside un auteur."*

✓ une structure

(atomique/composé,  
mono/mutivalué)

*atomique, monovalué*

✓ un domaine de valeurs

*chaîne de caractères alphabétiques*

✓ Attribut obligatoire

ou facultatif

*facultatif*

## 2.2. Règles de complétude d'un schéma E/A

Pour une contrainte d'intégrité : *Exemple*

- ✓ un nom *CI d'existence d'un mariage*
- ✓ les éléments de la CI *association Mariage, attributs âge et ~~sexe~~ de l'entité Personne.*
- ✓ expression de la CI *"Une occurrence de l'association Mariage entre P1 et P2 n'est valide que si  $\text{âge}(P1) > 18$  et  $\text{âge}(P2) > 18$  (et avant 2013:  $\text{sexe}(P1) \neq \text{sexe}(P2)$ )"*

## 2.3. Exemple VPC\* complet

### Démarche pratique pour produire un schéma E/A

- 1) Identifier les (types d') entités
- 2) Identifier les (types d') associations entre les entités
- 3) Identifier les attributs de chaque entité et de chaque association
- 4) Évaluer les cardinalités des associations.
- 5) Exprimer les CI éventuelles.

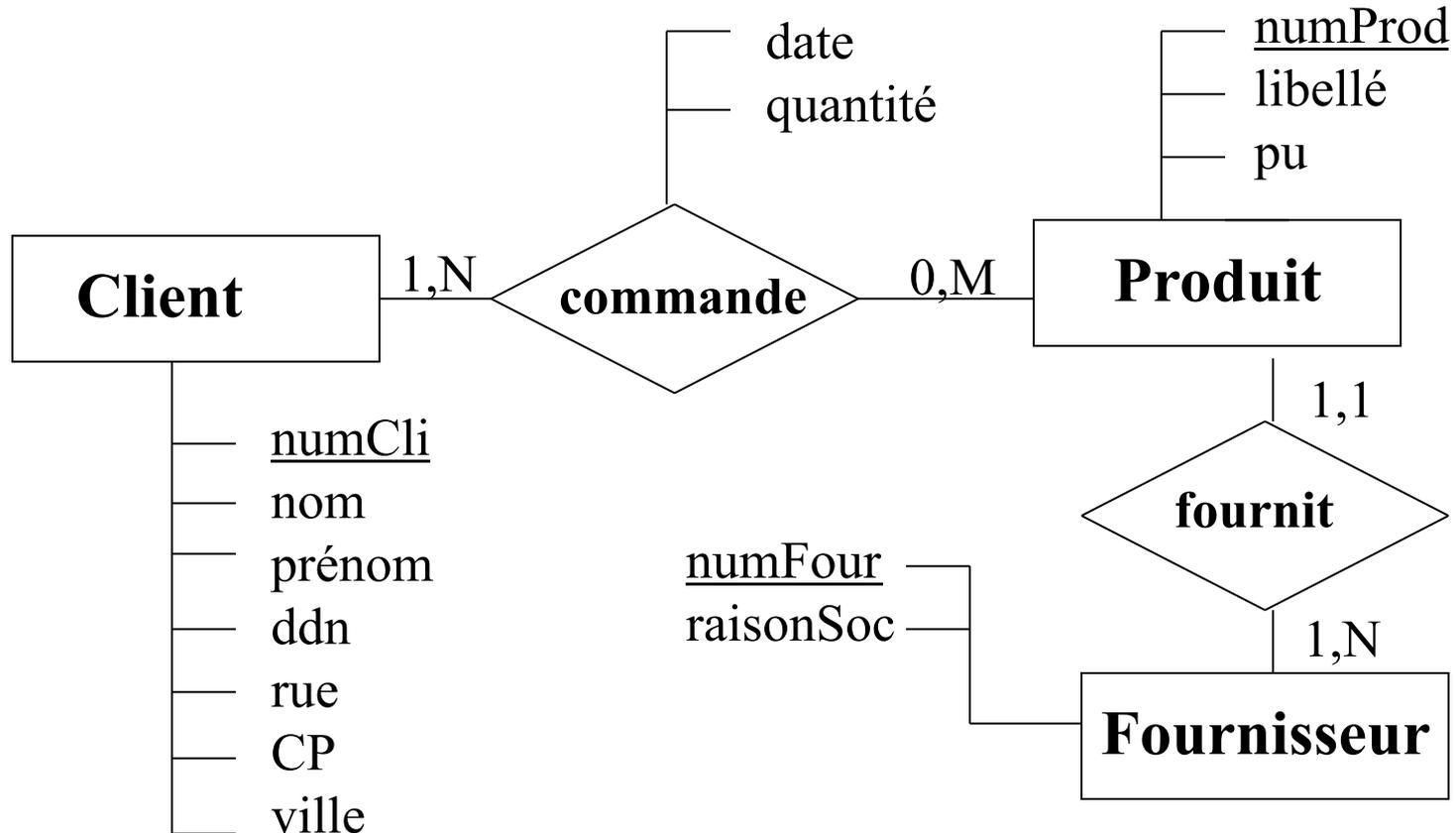
**Documenter correctement tous les éléments du modèle !**

## 2.3. Exemple VPC\* complet

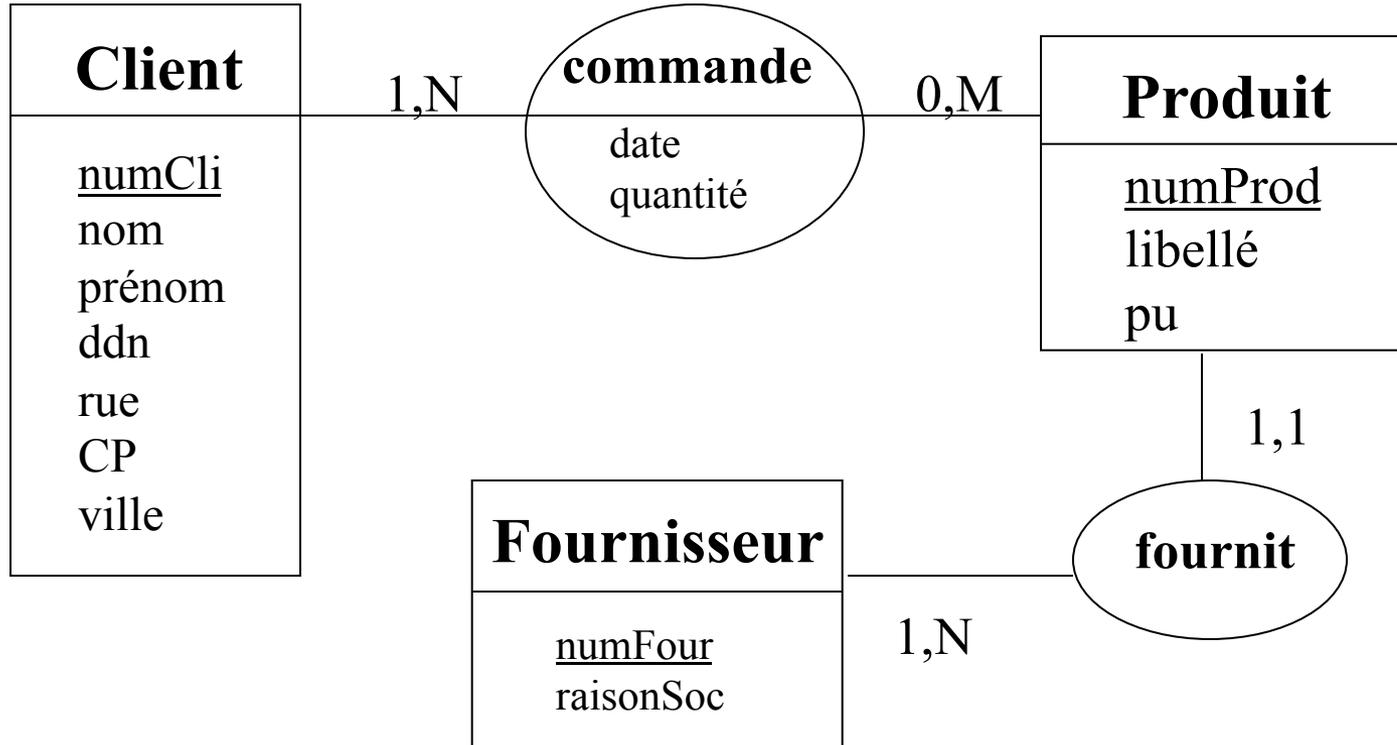
### Spécifications du système d'informations

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.
- Les clients commandent des produits à une date donnée et dans une quantité donnée.
- Les produits sont décrits par un numéro de produit, libellé et prix unitaire.
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

# Schéma E/A de l'exemple VPC



# Autre représentation graphique utilisée



# A vous : Construire un schéma E/A

## Gestion d'une mini-bibliothèque

- Un *ouvrage* est caractérisé par un numéro identifiant, un titre, un auteur et un éditeur.
- Un ouvrage est aussi décrit par une liste de *mots-clés* numérotés précisant les sujets qui y sont traités.
- La biblio. dispose d'un ou plusieurs *exemplaires* de chaque ouvrage. L'exemplaire est identifié par un numéro et est caractérisé par sa position dans les rayons et sa date d'achat.
- Un exemplaire peut être emprunté par un *lecteur* pouvant en emprunter plusieurs.
- Un lecteur est décrit par un numéro, son nom et son adresse.

# Identification des entités

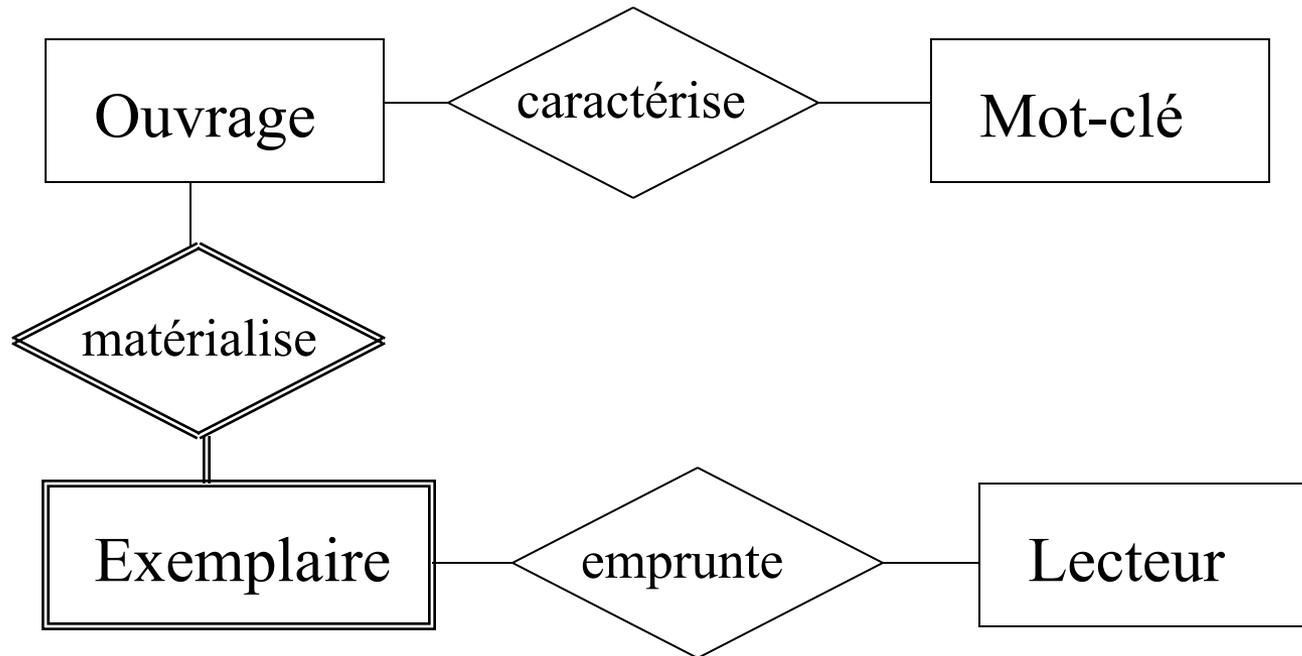
Ouvrage

Mot-clé

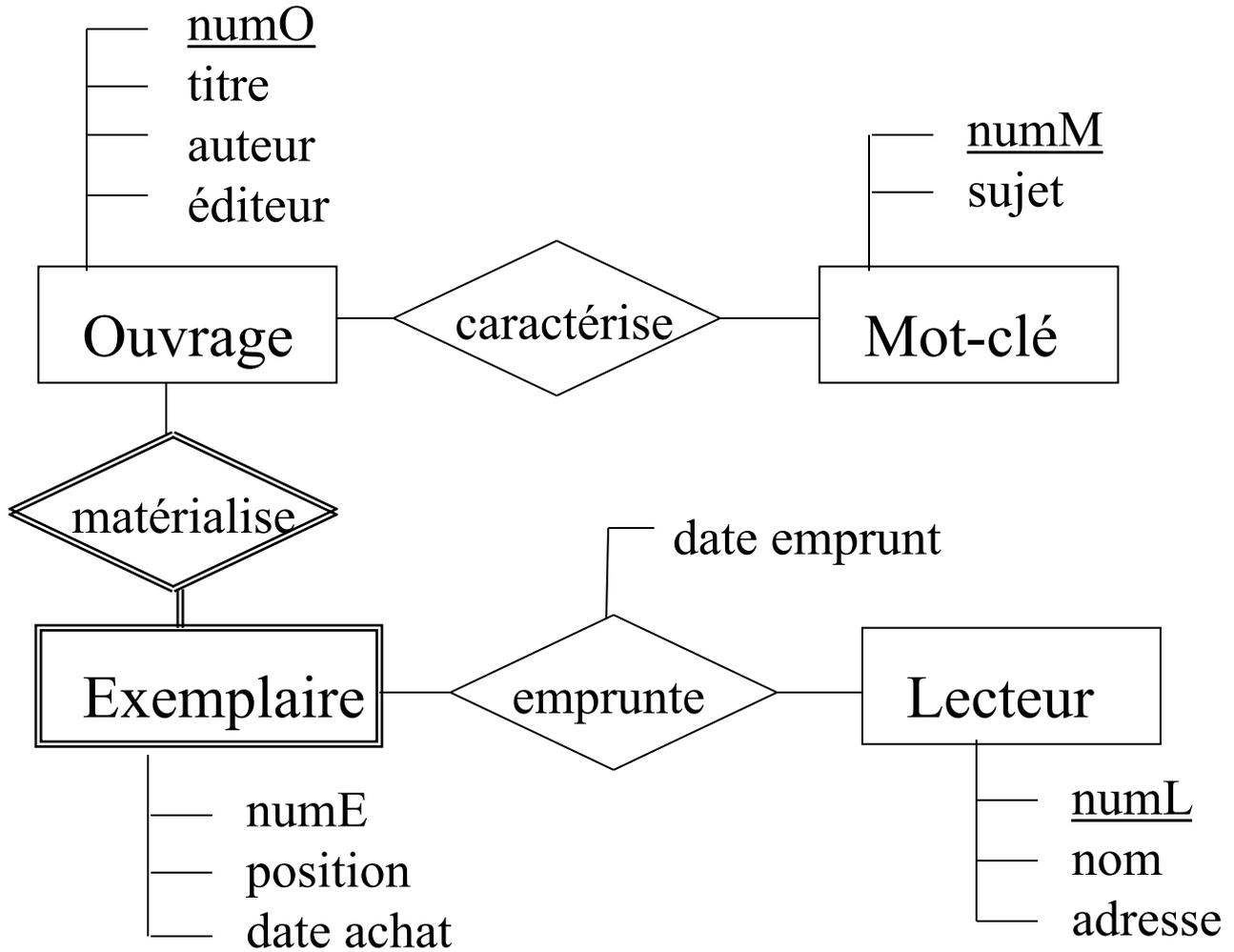
Exemplaire

Lecteur

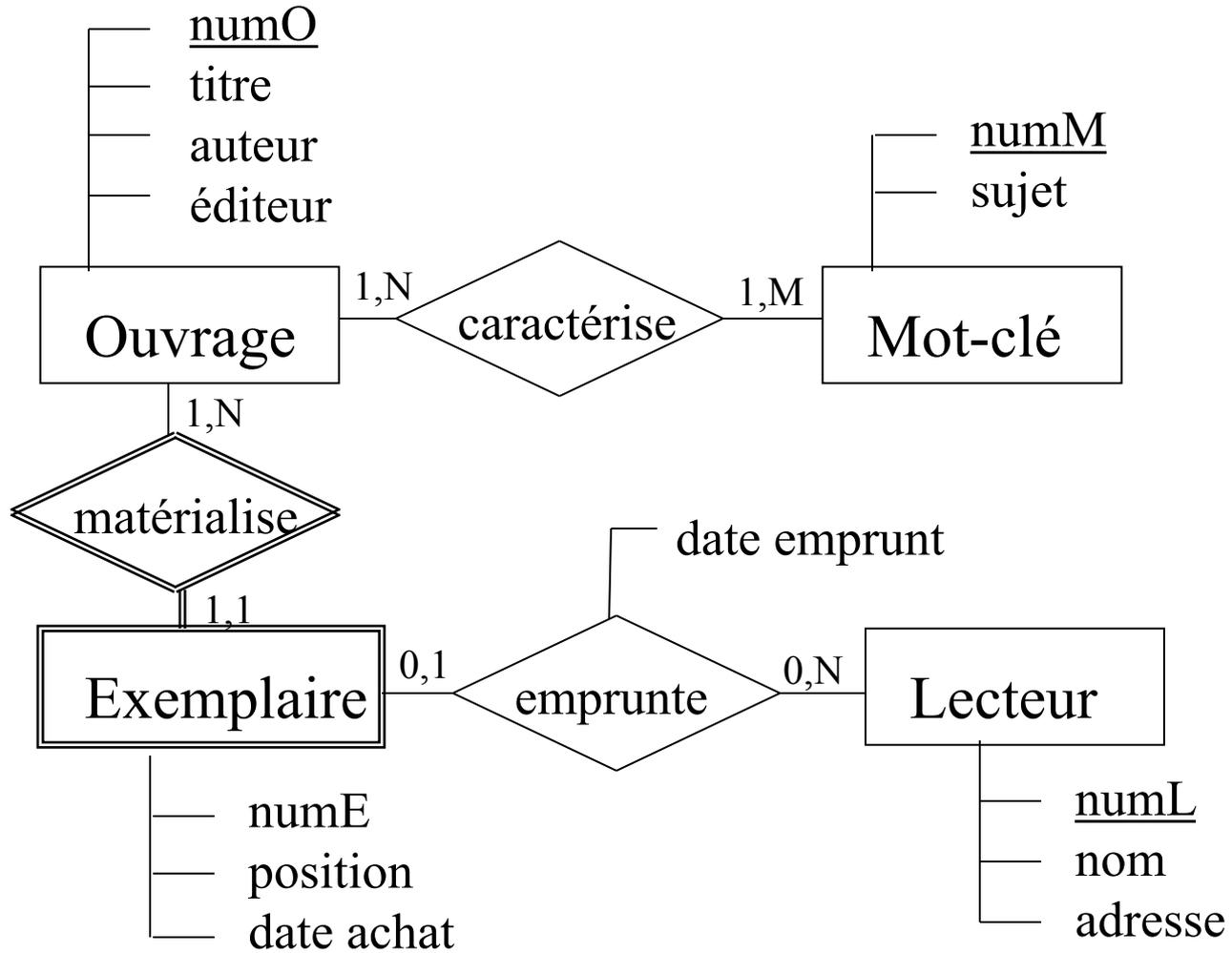
# Identification des associations



# Affectation des attributs



# Affectation des attributs



# Plan du cours

1. Introduction
2. Modèle conceptuel de données Entité-Association
- 3. Modèle relationnel de données**
4. Le langage SQL
5. PL/SQL
6. Transactions