

Bases de Données relationnelles

TELECOM Nancy 1ère année

Malika SMAÏL-TABBONE

Maître de conférences à l'UHP

(malika.smail@loria.fr)

Plan du cours (1/3)

Chap. 1 : Introduction au monde des BDs

- 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 conceptuel de données Entité-Association (E/A)

- Concepts de base du modèle E/A
- Règles de complétude d'un modèle 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 : PL/SQL (Programmation sous ORACLE)

PL/SQL : extension procédurale de SQL

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

Gestion de l'**intégrité** des données

- **Triggers**

Chap. 6 : Gestion des transactions et des accès concurrents

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. Le langage PL/SQL

6. Transactions et concurrence d'accès

Limites des systèmes de gestion de fichiers

- ✓ Chaque application dispose de ses fichiers (données)
⇒ redondance des données
- ✓ Chaque application dispose de ses programmes
⇒ 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, 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 décrire les données et autoriser leur évolution (tâche de *l'administrateur*)
- **Partage des données** : utilisation simultanée par différentes applications
- **Efficacité des accès aux données** : garantie d'un bon *débit* (nombre de transactions exécutées par seconde) et d'un bon *temps de réponse* (temps d'attente moyen par transaction)



Les 7 fonctions d'un SGBD

1. Définition des données
2. Manipulation des données
3. Persistance des données, gestion de la mémoire et accès aux données
4. Contrôler l'intégrité des données
5. Assurer la sécurité de fonctionnement
6. Gérer les accès concurrents
7. Assurer la confidentialité des données

1. 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'immatriculation*)
- 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

2. Manipulation des données

- Un Langage de Manipulation de Données (LMD) permet de *créer, rechercher, supprimer, modifier* des données (SQL)
- 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*).

3. Persistance des données, gestion de la mémoire et accès aux données

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

4. 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 (triggers)

5. 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.

Il faut 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

6. Gérer les accès concurrents aux données

Il faut permettre des accès simultanés par plusieurs utilisateurs

Mécanismes de gestion de conflits d'accès

- **autorisation** d'accès multiples en consultation (lecture de la BD)
- **verrouillage** en cas d'accès en modification (écriture dans la BD)

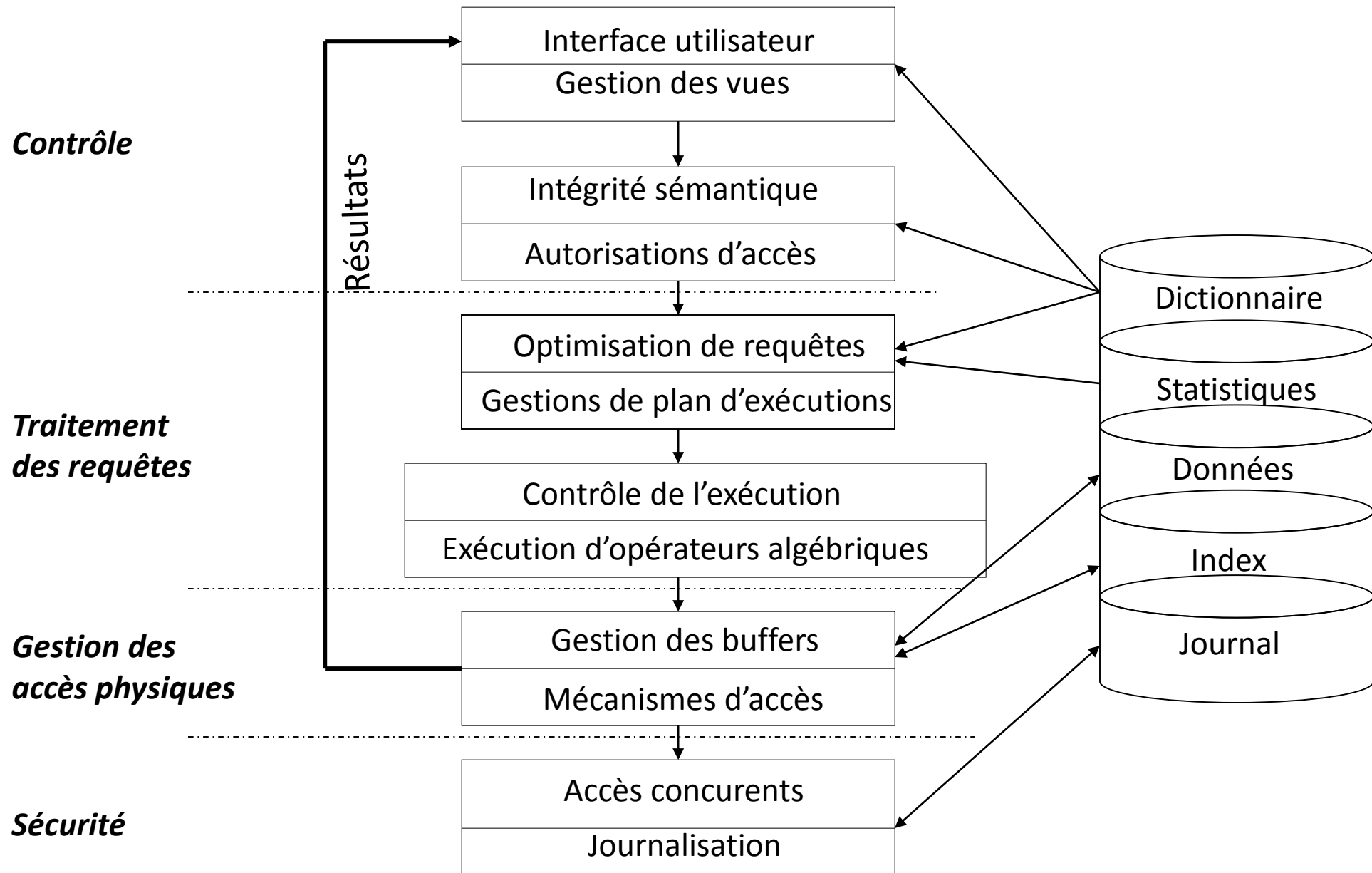


7. Assurer la confidentialité des données

Cacher certaines informations à certains utilisateurs

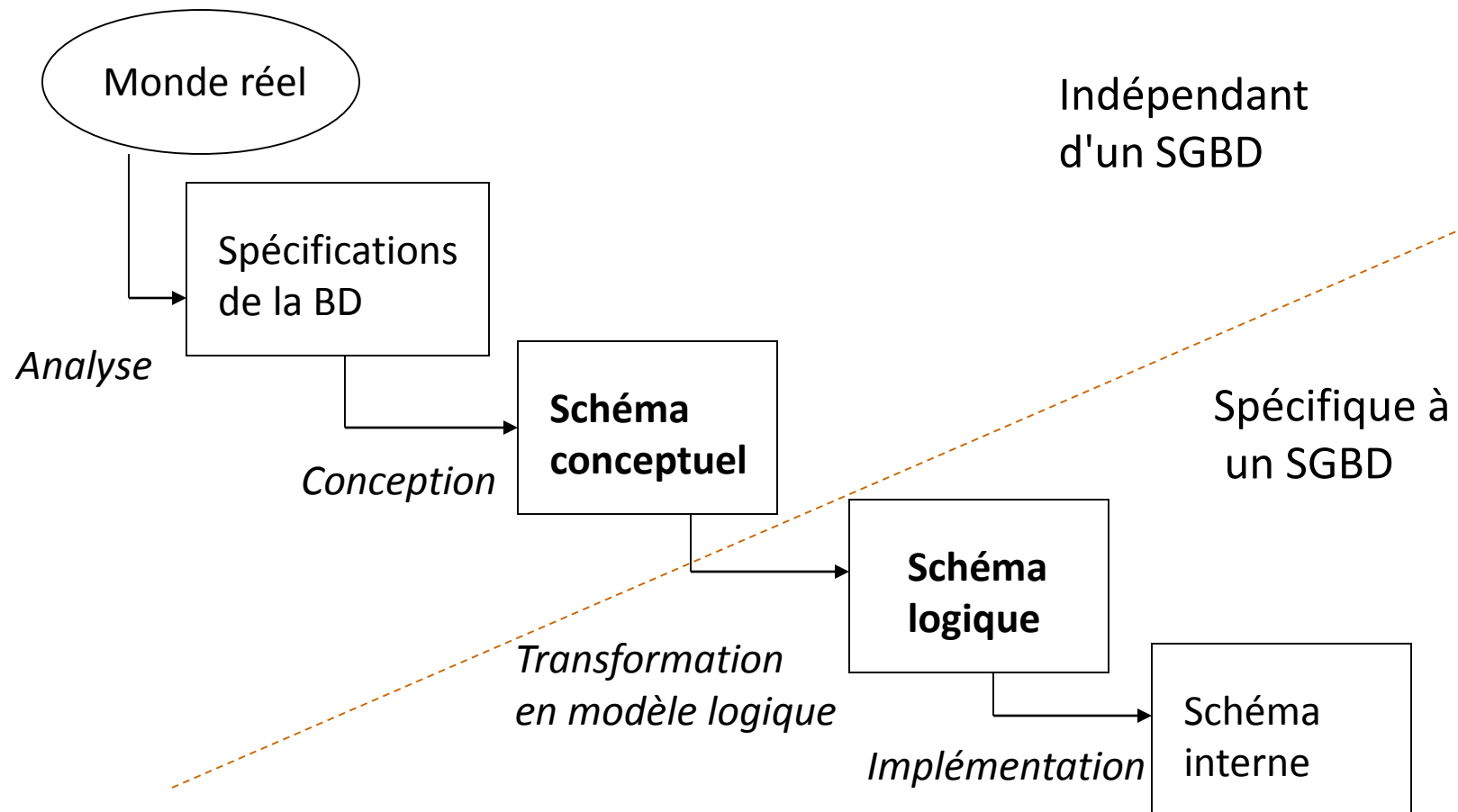
Mise en œuvre :

- comptes utilisateurs, **mots de passe**, 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 : **difficile** 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)
 - ✓ Partie dynamique : activité du système
(traitements opérant sur les données)

Plan du cours

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

Modèle Entité-Association (étendu)

Concepts du modèle E/A étendu

- (Type d') **entité** : représentation d'un ensemble d'objets abstraits ou concrets, caractérisée par une liste d'attributs. Un ou plusieurs attributs jouent le rôle d'*identifiant*
- (Type d') **association** : permet de décrire les liens "sémantiques" entre des entités, peut être caractérisé par des attributs
 - Association *unaire*, association *binaire*, association *n-aire*
 - Spécialisation/Généralisation
- (Type d') **attribut** : décrit une propriété attachée soit à une entité, soit à une association.

Entités et attributs (1/2)

- **Entité** : objet concret ou abstrait à propos duquel on souhaite mémoriser des données

ex. client, journal, département

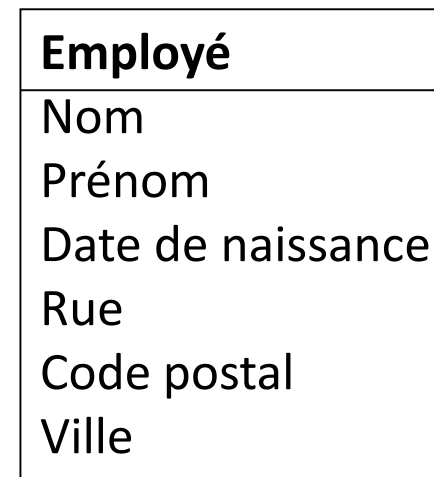
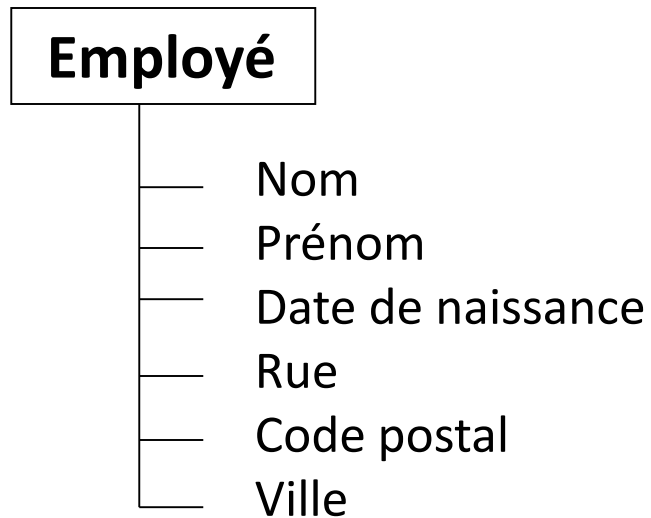
- **Attribut** : propriété attachée à une entité
 - ✓ Attribut **atomique / composé** (subdivisé en attributs)
ex. nom versus adresse composé de rue, ville, CP
 - ✓ Attribut **monovalué / multivalué** (plusieurs valeurs par occurrence d'entité)
ex. nom versus prénoms_des_enfants
 - ✓ Attribut **dérivé** : calculé à partir d'autre(s) attribut(s)
ex. âge calculé à partir de date_de_naissance

Entités et attributs (2/2)

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

ex. Prix des produits : *nombre réel positif*
 nom : *chaîne de caractères alphabétiques*
 couleur : *{rouge, vert, bleu, jaune}*

Deux représentations graphiques possibles d'une entité avec ses attributs



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
Richard Albus
Marie Dupont
Bernard Schmidt ...

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*

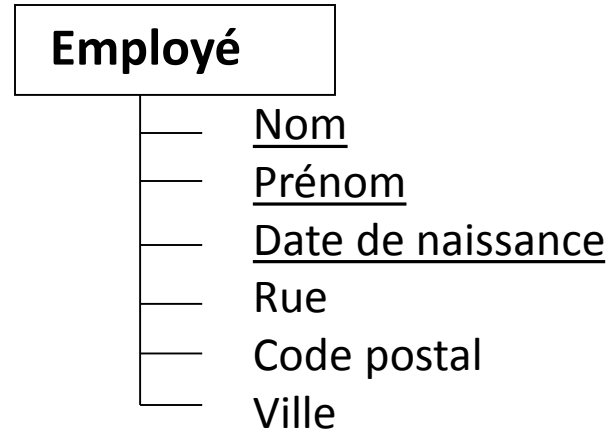
- **Occurrence d'attribut** : valeur particulière

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

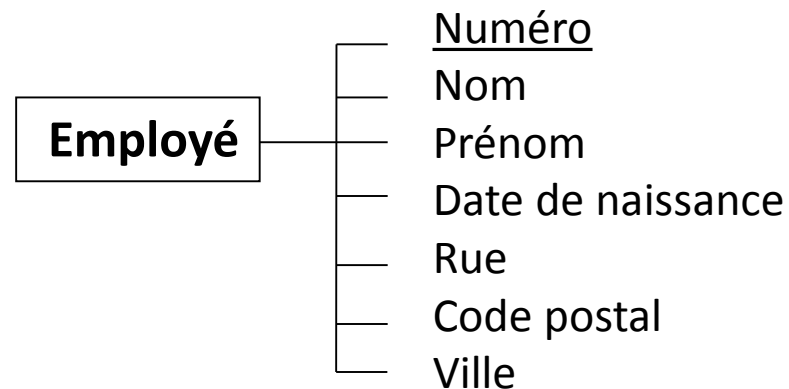
Identifiant d'une entité

- **Identifiant d'une entité** : ensemble **minimal** d'attributs caractérisant de façon **unique** chaque occurrence 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 naissance}



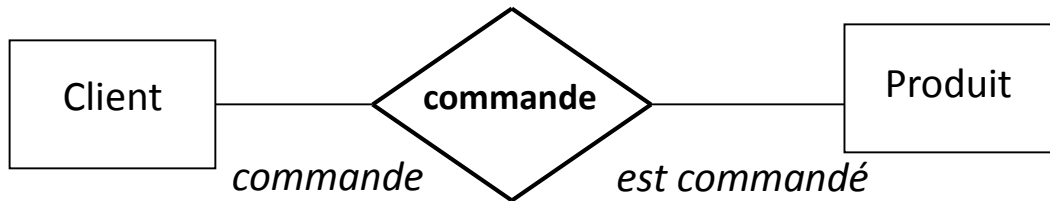
Un autre identifiant du type d'entité Employé : un nouvel attribut (artificiel) **Numéro**



Associations

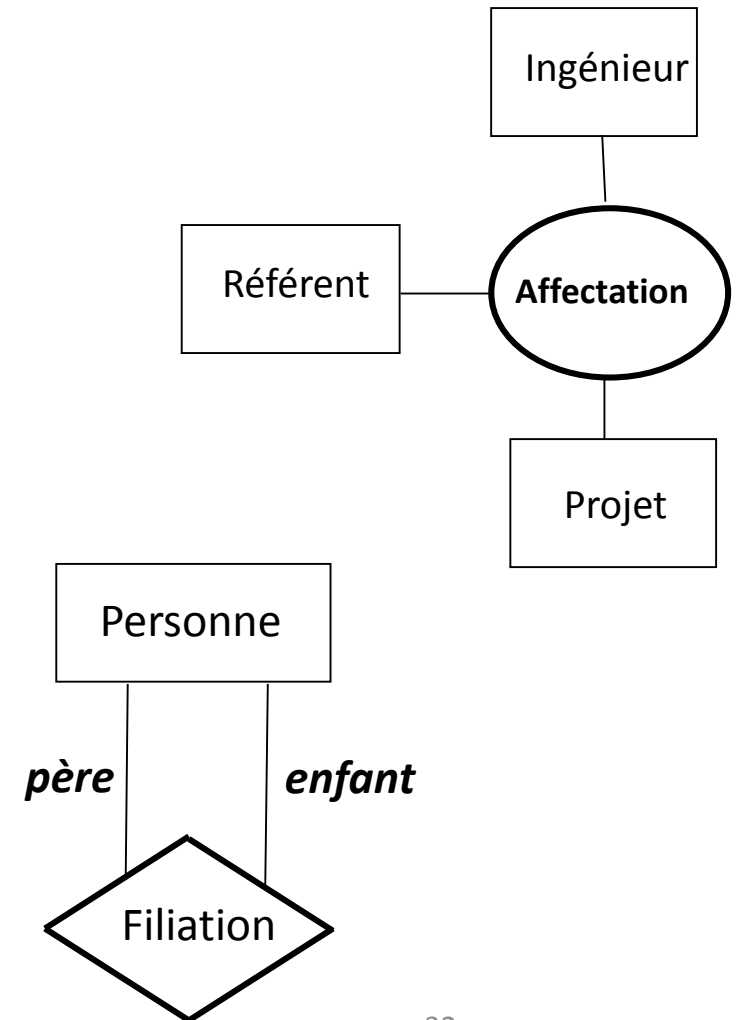
- **Association** : lien existant entre des entités

ex. les clients **commandent** des produits; les ingénieurs sont **affectés** à des projets sous la responsabilité de référents; chaque personne à un père (lien de **filiation**)



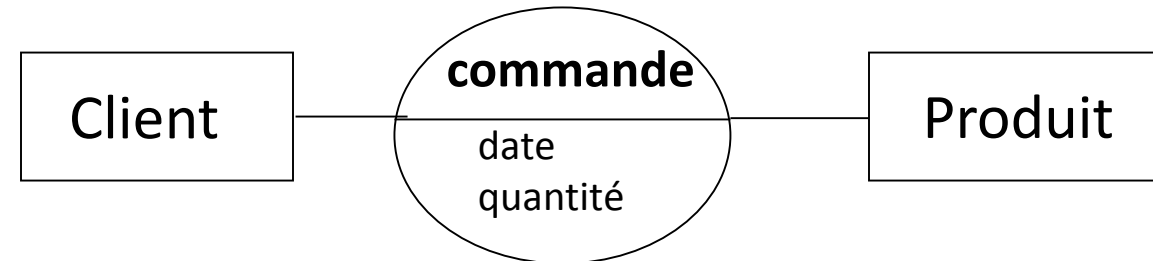
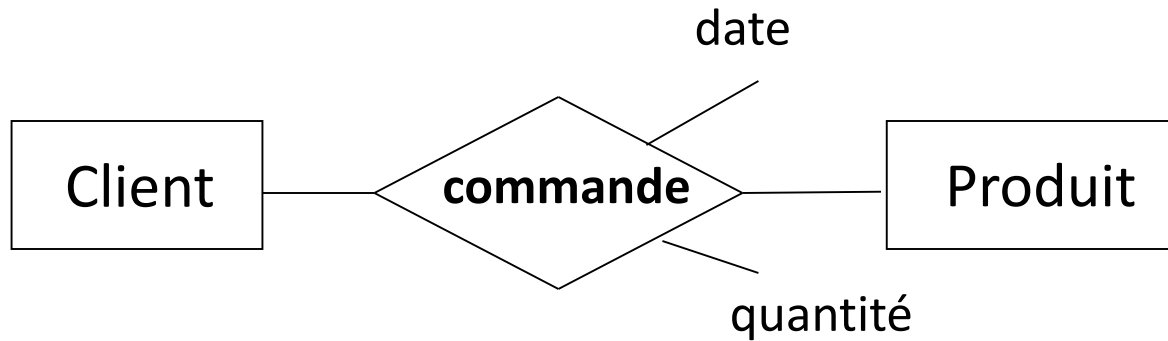
- Chaque entité joue un **rôle** dans l'association
- **Arité** d'une association : nombre d'entités reliées

Association **binaire** (arité=2), **unaire** (arité=1), **n-aire** (arité=n)



Attributs d'associations (2 représentations graphiques)

- Une association peut être caractérisée par des **attributs**.
ex. *date* de la commande et *quantité* de produits commandés.

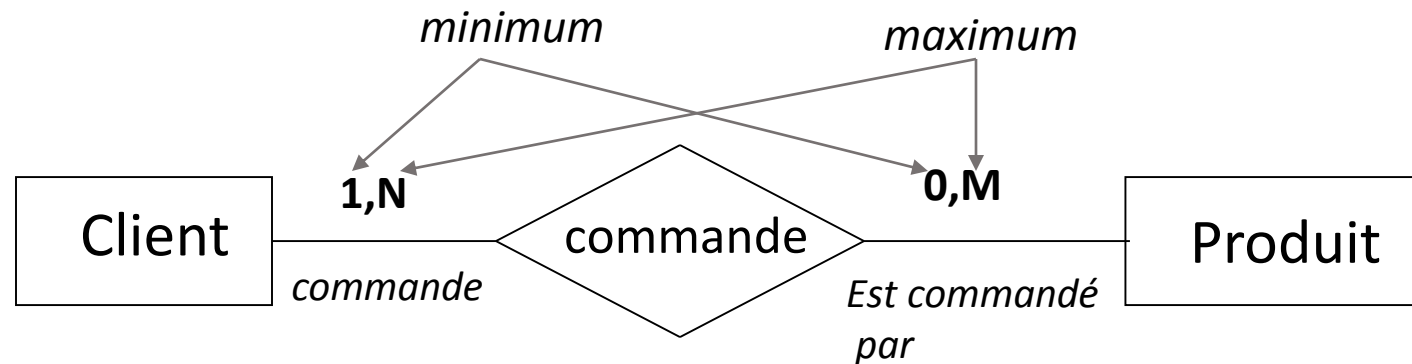


- **Occurrence d'un type d'association** : occurrence de chaque type d'entité + occurrence de chaque type d'attribut

Cardinalités d'associations

- **Cardinalité d'une association** : 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 un nombre quelconque de clients.

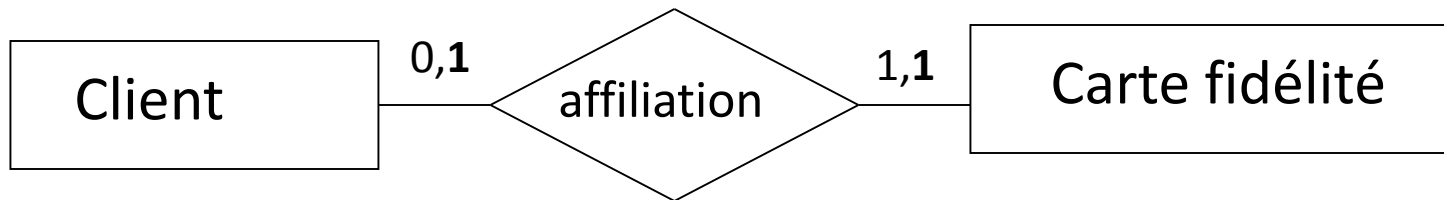


En fonction des cardinalités **maximales**, une association binaire peut être de **type** 1-1, 1-N ou N-M

Association binaire 1-1

Association 1-1 (un à un, *one-to-one*) :

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



*Un client est titulaire d'au maximum une carte de fidélité ;
une carte appartient à un client et un seul.*

Association binaire 1-N

Association 1-N (un à plusieurs, *one-to-many*) :

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 qu'à une occ. de F



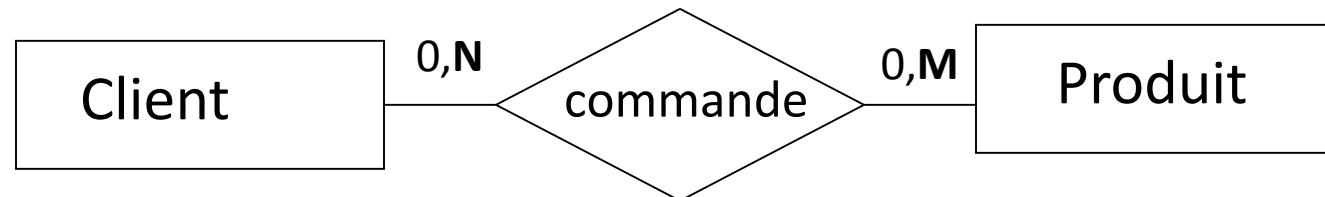
Un étudiant s'inscrit à une formation et une seule ;

Une formation peut accueillir plusieurs étudiants

Associations binaires N-M

Association N-M (plusieurs à plusieurs, *many-to-many*) :

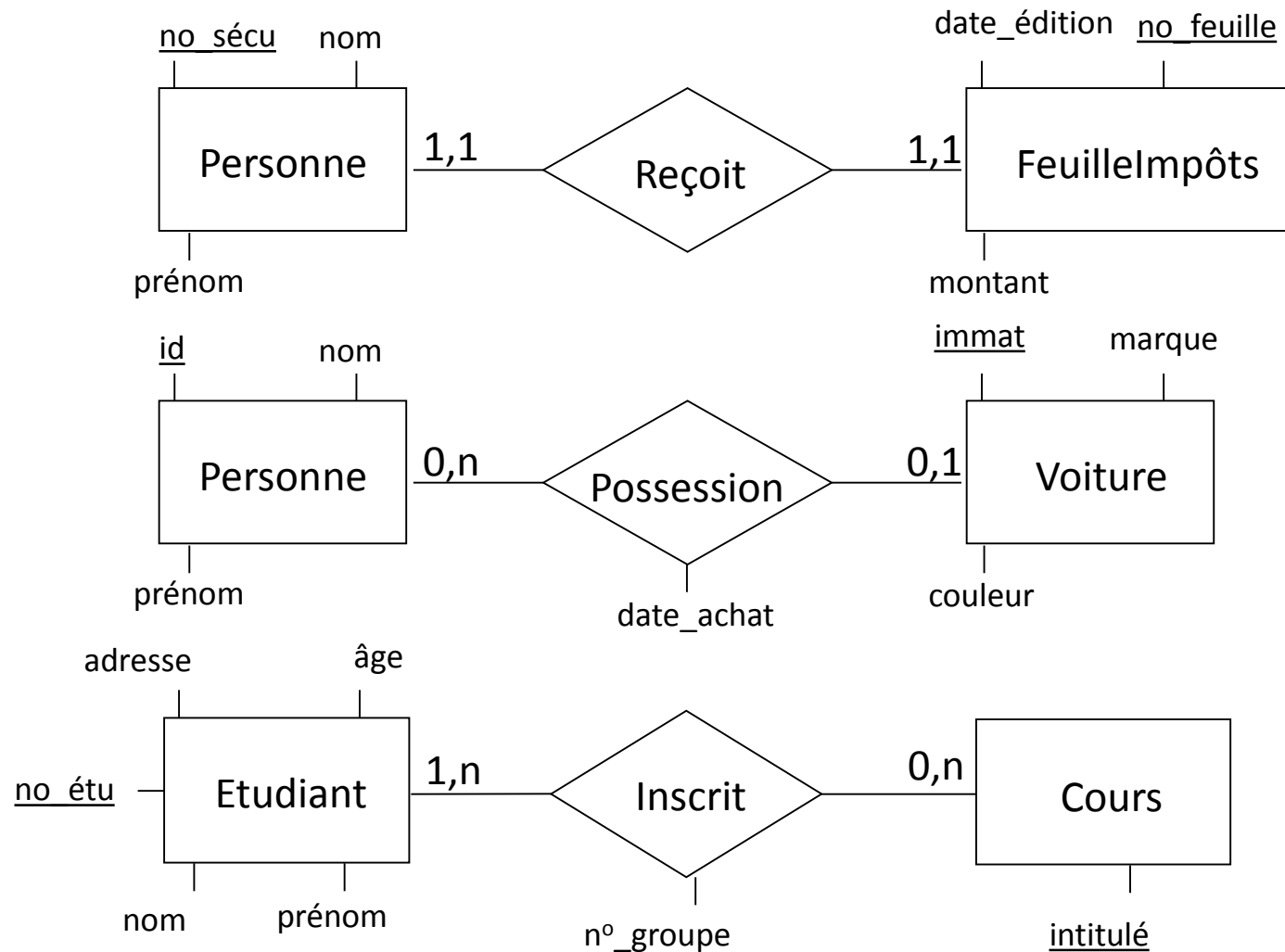
Une association A entre deux entités E et F est de cardinalité N-M si une occurrence de E peut être associée à plusieurs occurrences de F et réciproquement



Un client peut commander plusieurs produits;

un produit peut être commandé par plusieurs clients.

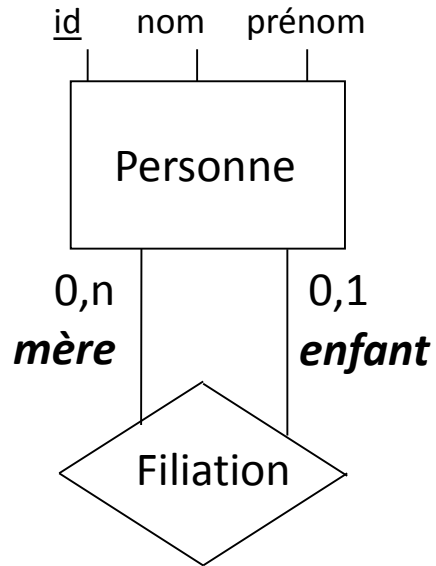
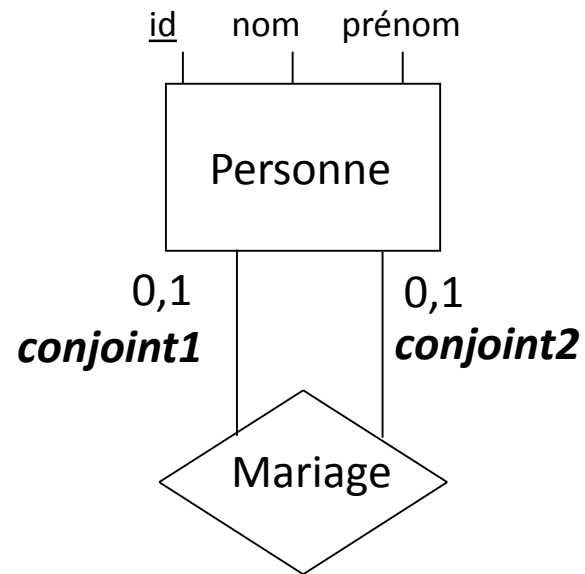
Autres exemples d'associations binaires



Association unaire (réflexive)

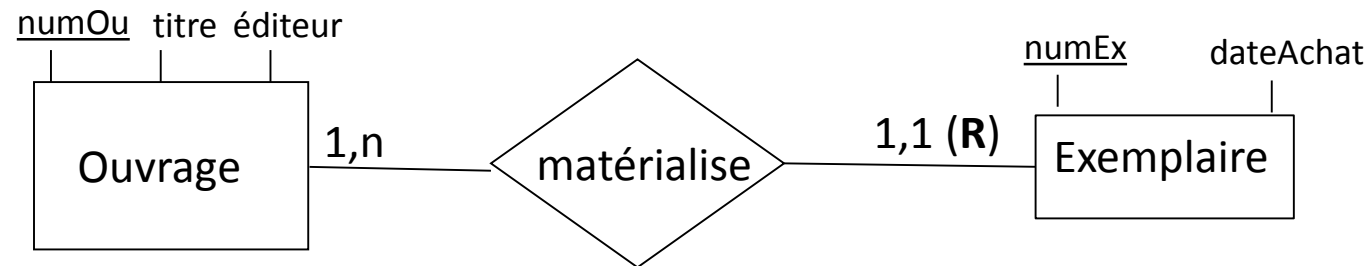
Association reliant une entité à elle-même

- Nécessité de nommer les rôles pour ces associations



Entités faibles

- Entité faible est une entité sans identifiant propre
 - **Dépendance d'existence** : une occurrence d'une entité faible n'existe qu'en référence à une occurrence d'une autre entité dite *entité identifiante*
 - L'association qui les unit est dite association *identifiante*
 - L'entité faible a une cardinalité (1,1) sur son association identifiante et possède un **identifiant relatif (R)**



Exemple est une entité faible

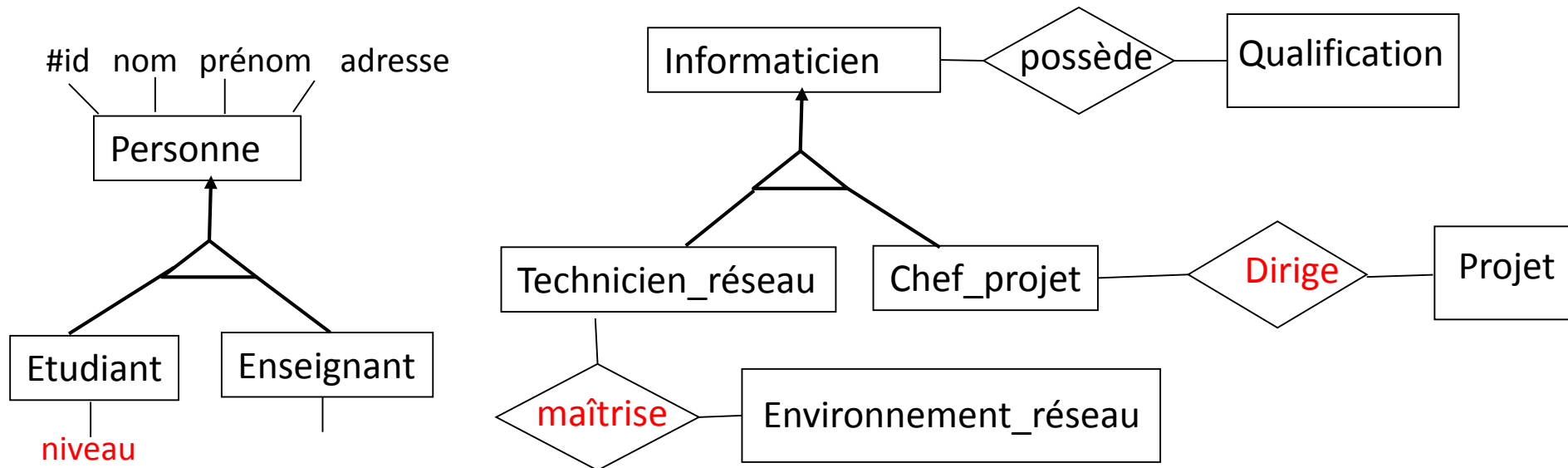
numEx permet d'identifier un exemple relativement à un Ouvrage

Spécialisation d'entité (sous-type)

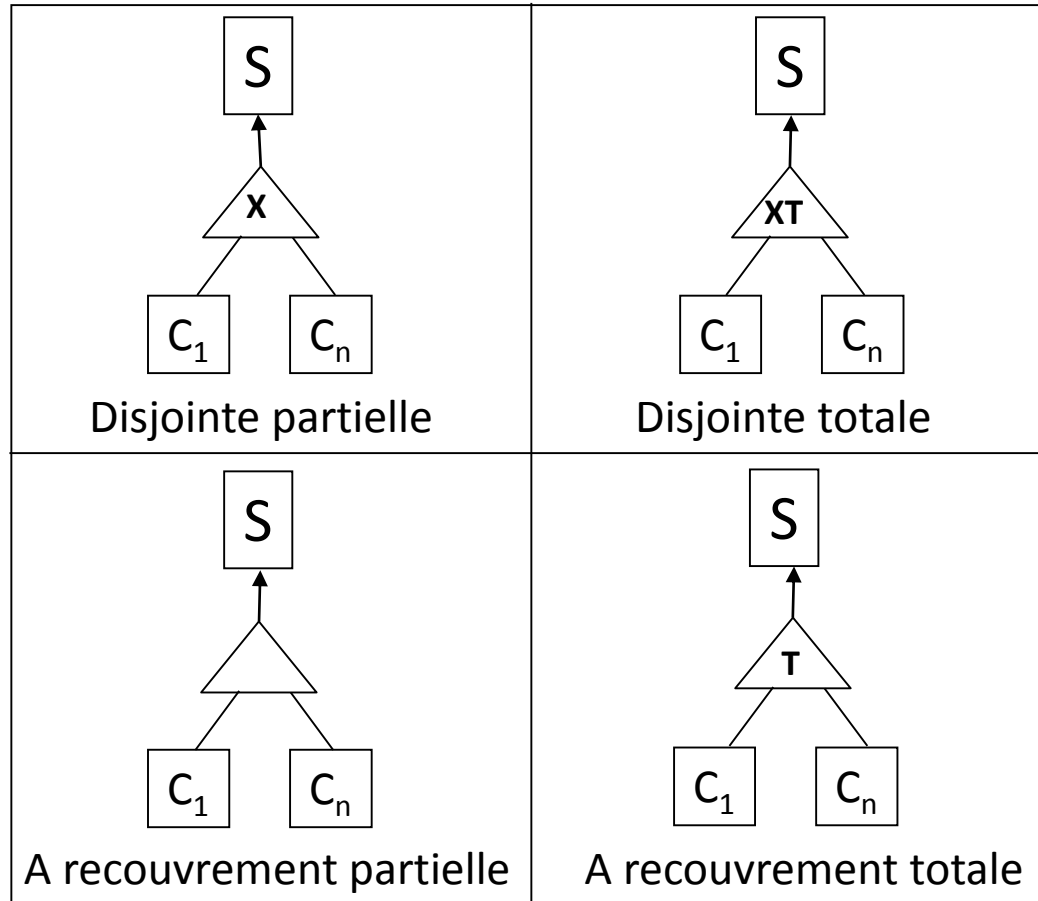
❑ Décomposition d'une entité (générique) en entités spécialisées pour prendre en compte des différences entre certaines occurrences en cas de :

- de **propriétés** spécifiques
- d'**associations** spécifiques

❑ **Héritage** des attributs de l'entité générique par les entités spécialisées



Typologie des spécialisations



- Spécialisation **Disjointe (X)**:

$$\forall i,j, i \neq j, |C_i| \cap |C_j| = \emptyset$$

ou à **recouvrement** (sinon)

- Spécialisation **totale (T)**:

$$|S| = \bigcup |C_i|$$

ou **partielle** :

$$\bigcup |C_i| \subset |S|$$

Contraintes d'intégrité associées à un MCD (CI)

But : Identifier des contraintes 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*

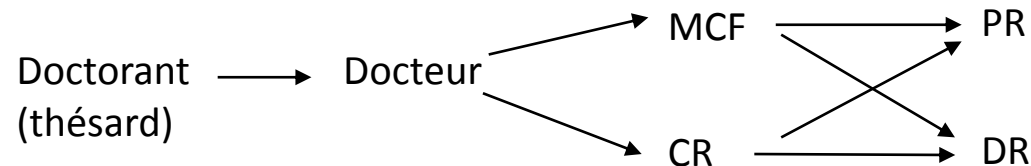
- ex2 : *nom*, *no-sécurité-sociale* : attributs obligatoires

- ex3 : *le volume d'un avion ne dépend que de son modèle* (dépendance fonctionnelle)

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

- ex1 : *l'attribut salaire ne peut pas diminuer*

- ex2 : *changements d'état dans l'enseignement supérieur et la recherche*

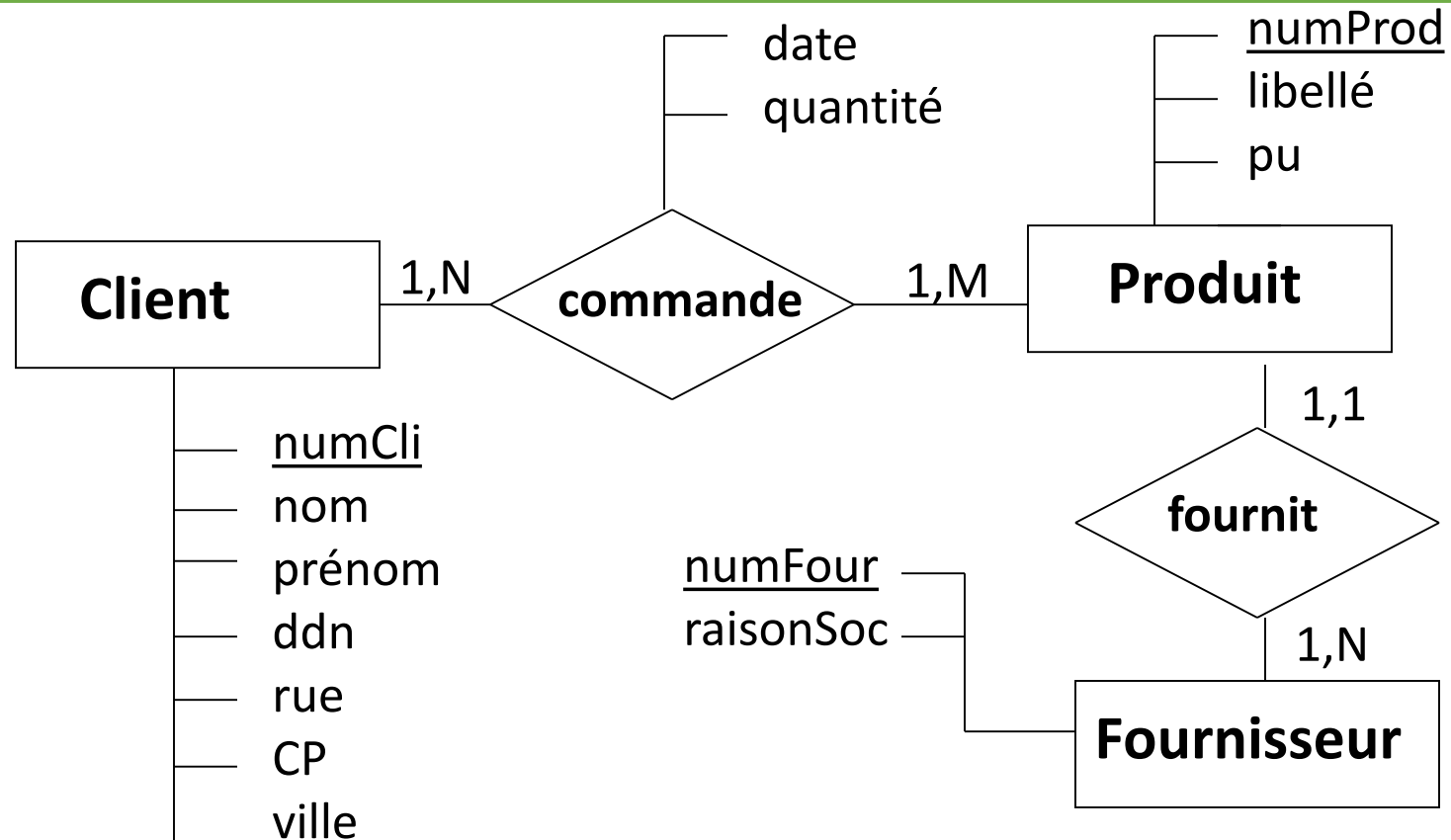


Exemple de MCD "complet" (VPC) (1/2)

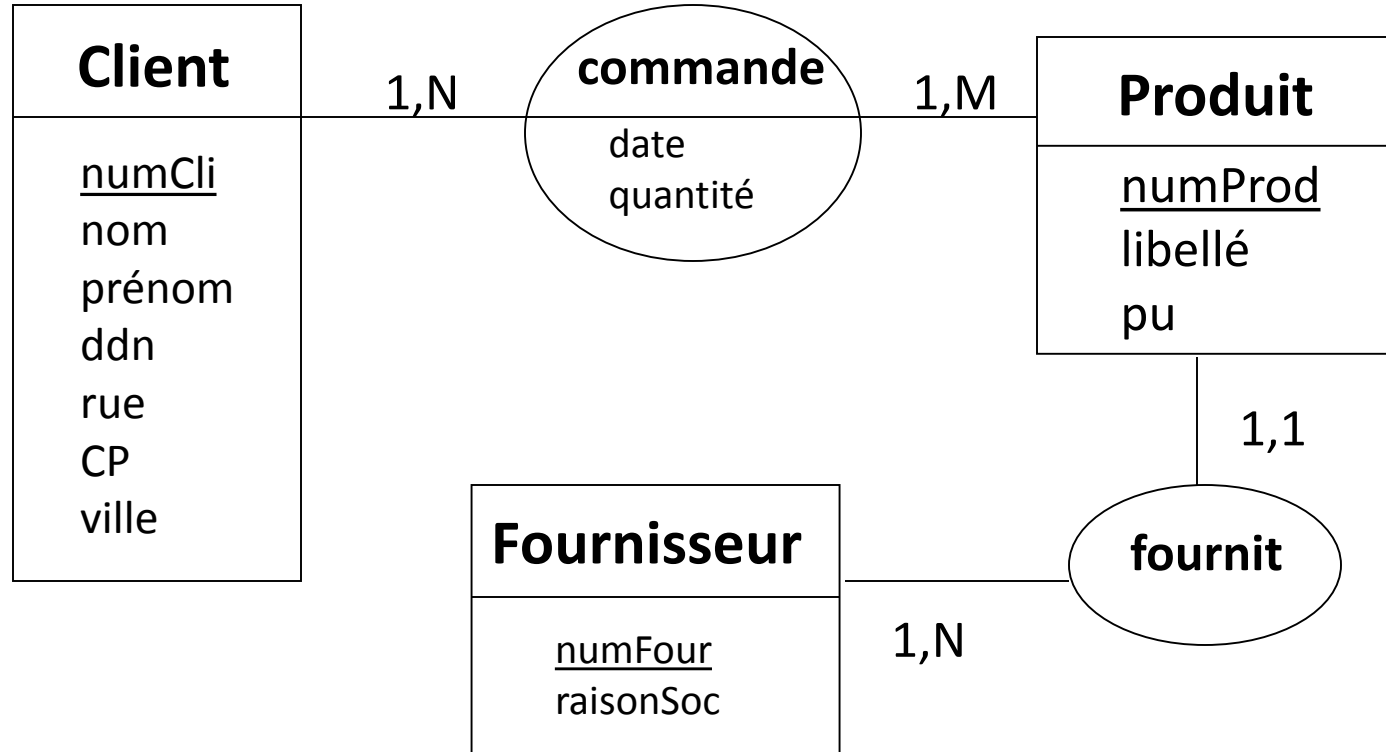
Spécifications du système d'informations VPC

- Les clients sont caractérisés par un numéro (unique) 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 (unique) 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 (unique) de fournisseur et leur raison sociale

Exemple de MCD "complet" (VPC) (2/2)



Autre représentation graphique possible (à la UML)



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.

Quelles sont les entités ?

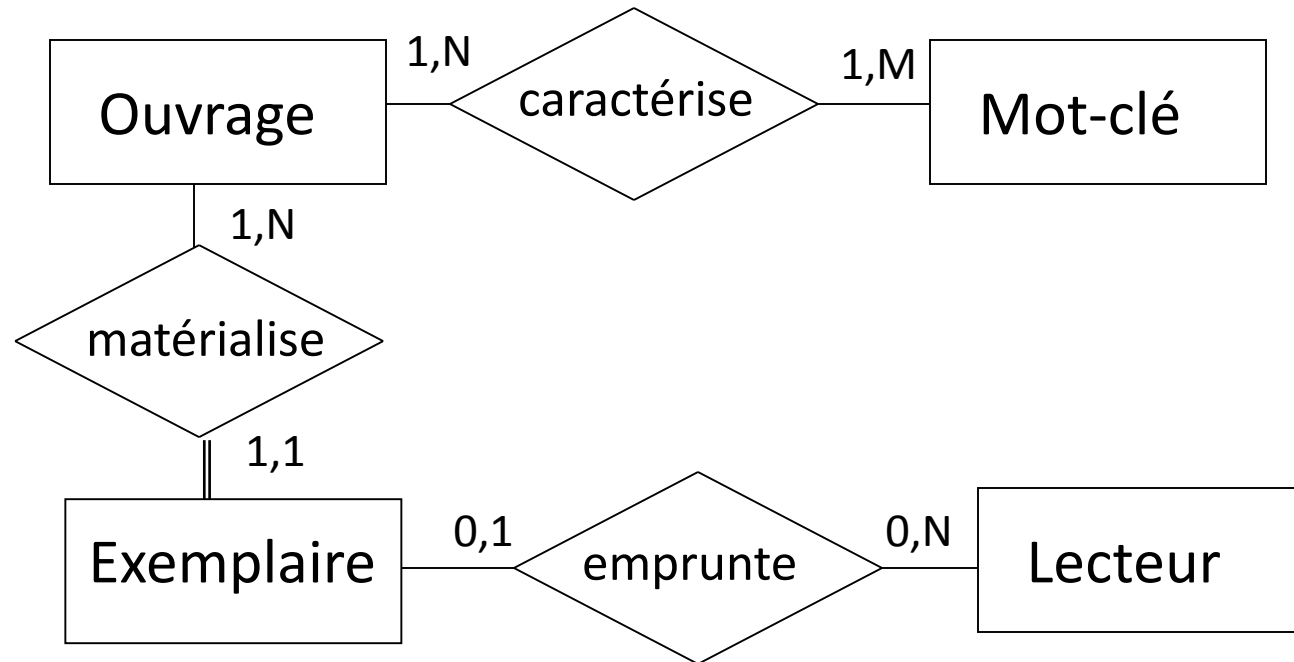
Ouvrage

Mot-clé

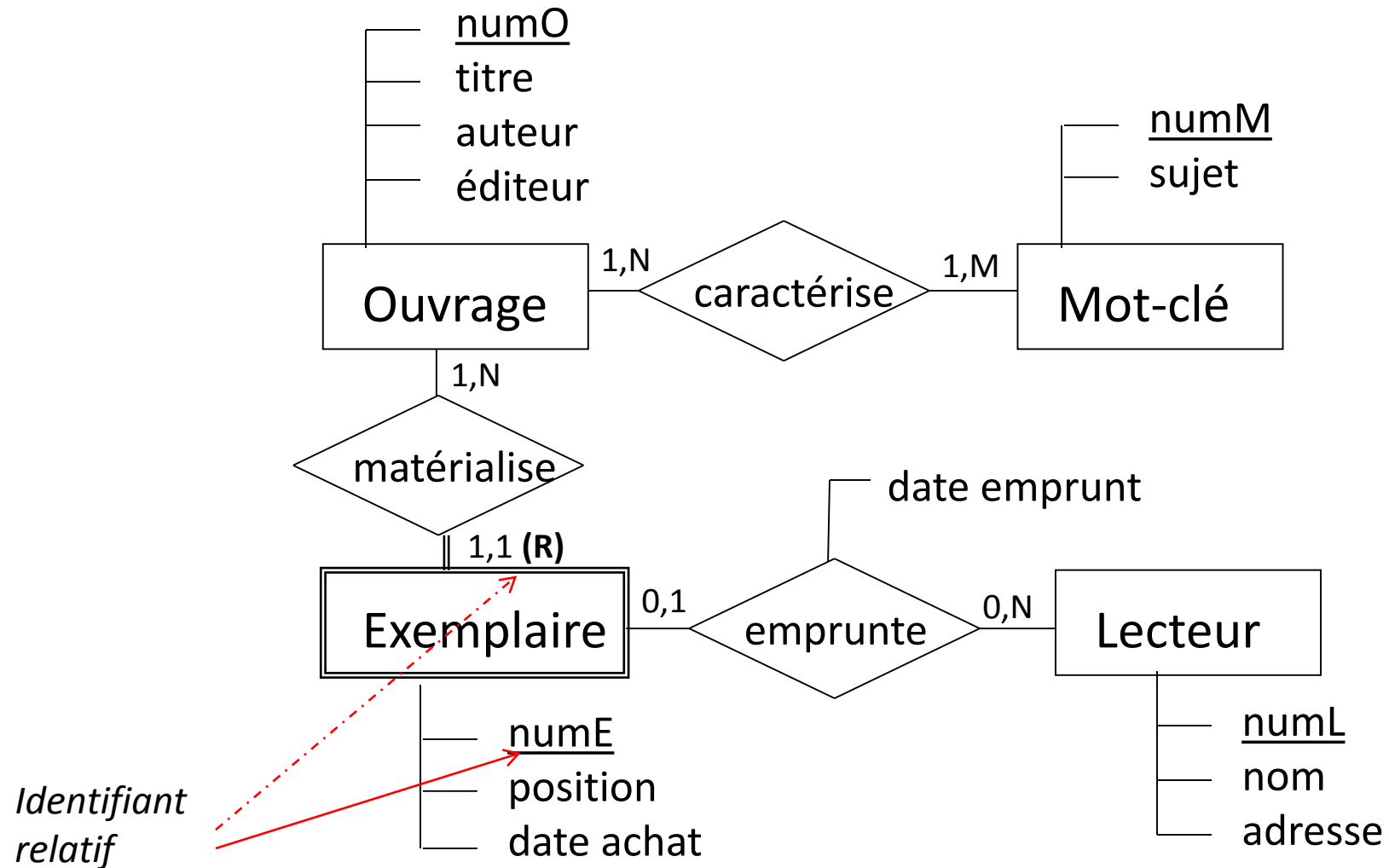
Exemplaire

Lecteur

Quelles sont les associations et leurs cardinalités ?



Quels sont les attributs ? les identifiants (relatifs) ?



Démarche pour construire un modèle E/A complet

- 1) Identifier les (types d') entités
- 2) Identifier les (types d') associations entre les entités et leurs cardinalités
- 3) Identifier les (types d') attributs de chaque entité et de chaque association et choix d'un identifiant par entité
- 4) Exprimer les contraintes d'intégrité
 - règles de gestion
 - dépendances fonctionnelles entre attributs ...

Documenter correctement tous les éléments du modèle !

Modèle E/A complet (1/4)

Pour qu'un schéma E/A soit complet, chaque entité/association/attribut/CI doit être documenté(e)

Pour chaque **entité** :

Exemple

✓ un **nom**

Auteur

✓ une **définition**

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

✓ une liste d'attributs

{nom-aut, prénom-aut, ville-aut}

✓ au moins un identifiant

{nom-aut, prénom-aut}

Modèle E/A complet (2/4)

Pour chaque **association** :

- ✓ un **nom** *auteur-livre*
- ✓ une **définition** *"auteur-livre associe l'auteur aux livres qu'il a écrits"*
- ✓ la liste des entités *Auteur, Livre*
 - les rôles *Auteur écrit (1, N) Livre*
 - et leur cardinalité *Livre est écrit par (1,1) Auteur*
- ✓ la liste des attributs *NA (non applicable)*

Modèle E/A complet (3/4)

Pour chaque contrainte d'intégrité :

- ✓ un **nom** *CI d'existence d'une occurrence de l'association **Mariage***
- ✓ les **éléments de la CI** *association **Mariage**, attribut **âge** de l'entité **Personne**.*
- ✓ **expression de la CI** *"Une occurrence de l'association Mariage entre deux personnes $P1$ et $P2$ n'est valide que si $\text{âge}(P1) > 18$ et $\text{âge}(P2) > 18$ "*

Modèle E/A complet (4/4)

La documentation des attributs d'un modèle conceptuel de données se fait sous forme d'un **dictionnaire des données**

Dictionnaire de données (1/4)

- Un soin particulier pour la documentation des attributs dans le dictionnaire de données
- Il est possible de commencer par la construction du dictionnaire avant même l'esquisse du premier MCD
 - Recenser toutes les **données utiles à mémoriser** (Grille d'analyse) qui vont former les attributs
 - Chaque attribut a une entrée dans le dictionnaire (tableau)

Dictionnaire de données (2/4)

Pour identifier les attributs d'un système d'information, il faut gérer :

- Les données **composées**
éclater en plusieurs données élémentaires
- Les problèmes de synonymie et polysémie
- Les données **calculées** ou **dérivées** à partir d'autres données
ne pas les inclure dans le dictionnaire (sauf cas particulier)
- Les **paramètres** du SI : Propriétés dont la valeur est constante ou prévisible (ex. date du jour, taux TVA)

Dictionnaire de données (3/4)

Chaque entrée du dictionnaire comporte au moins :

- Le nom d'un attribut *ex. ville-aut*
- La **définition** en clair *ex. "Nom de la ville dans laquelle réside un auteur."*
- Le **domaine** de valeurs possibles (type de données) *ex. chaîne de caractères*
- D'autres **propriétés de l'attribut**
 - atomique/composé ? *ex. atomique*
 - monovalué/multivalué ? *ex. monovalué*
 - Obligatoire/facultatif ? *ex. facultatif*
 - Calculé (O/N) ? *ex. N*

Aperçu d'un dictionnaire de données (4/4)

Nom attribut	Définition	Domaine	Atomique ?	Monovalué ?	Obligatoire ?	Calculé ?