

# Base de données 1

# Objectif

L'objectif de ce module est l'acquisition des connaissances fondamentales relatives aux bases de données, la conception du schéma des données, la construction du modèle relationnel et la manipulation des données en utilisant le langage SQL.

**Apprendre les  
concepts et les notions  
de Base de Données**

**Apprendre les techniques  
de conception de Base de  
Données relationnelles**

**Apprendre le  
langage SQL**

# Plan :

- I.** Introduction aux bases de données.
- II.** Modélisation conceptuelle.
- III.** Base de données relationnelle.
- IV.** Algèbre relationnel.
- V.** Le langage SQL.



# **Chapitre I: Introduction aux bases de données**



## Les informations sont dispersées !!!

- ✓ Risque de perdre des informations (l'oubli, le vol ...)
- ✓ Probabilité d'erreur
- ✓ Perte de temps
- ✓ La redondance des informations
- ✓ ....

Mauvaises

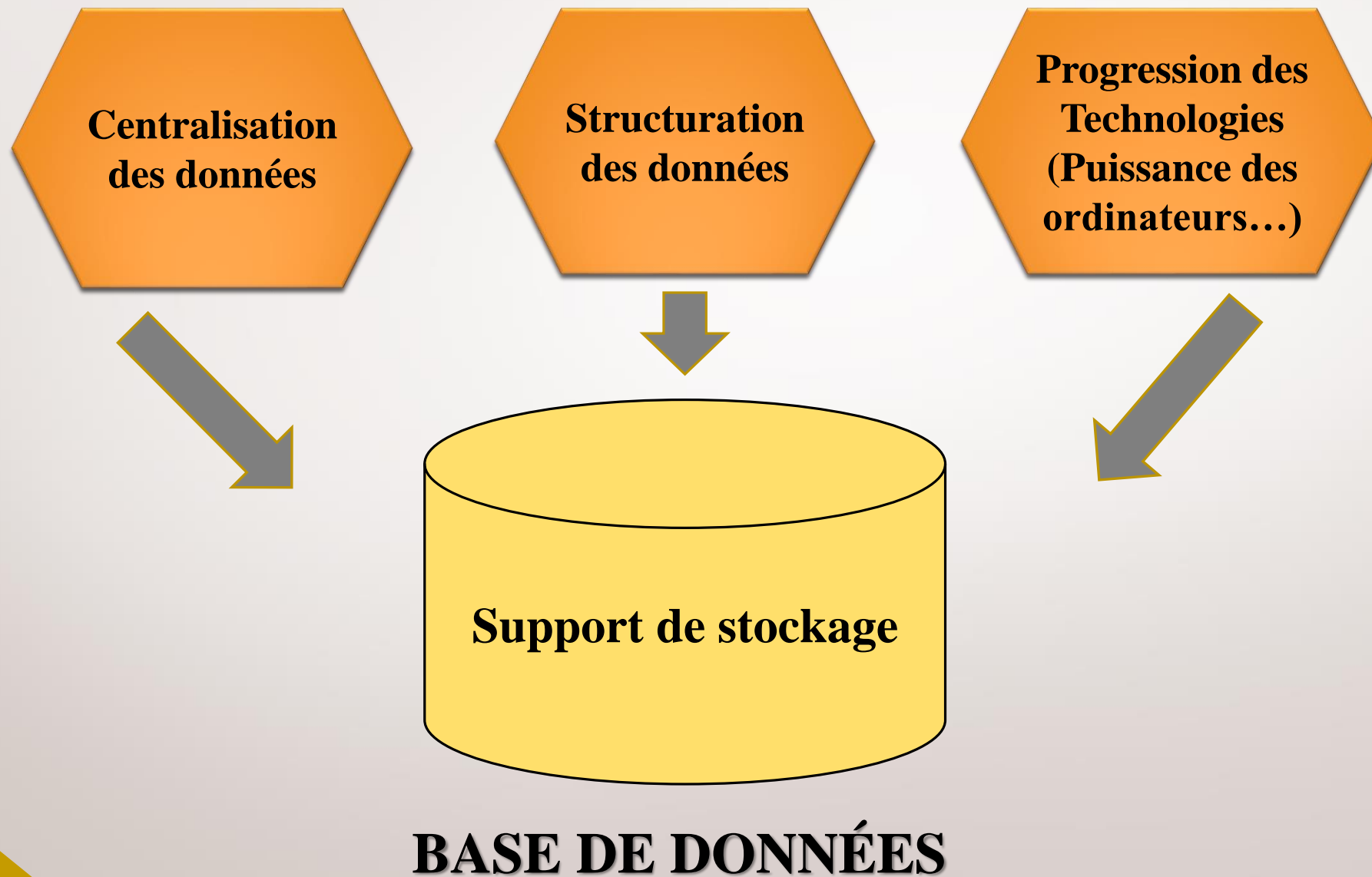
Besoin

Prendre



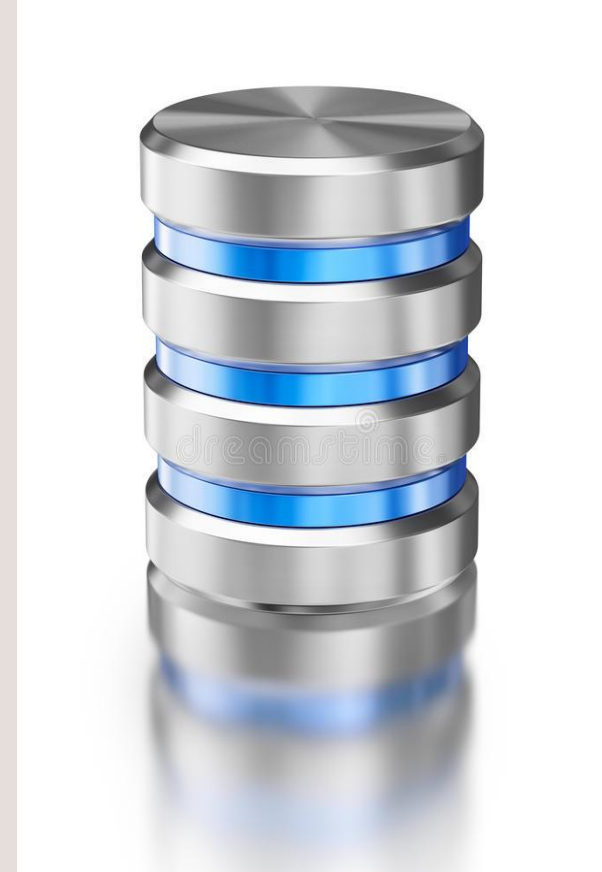
DECISIONS

## LA SOLUTION A ENVISAGER



# 1. Qu'est-ce qu'une base de données ?

- Une base de données (BD, en anglais DB: database) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible.
- Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers.
- La possibilité d'accéder à la base de données par plusieurs utilisateurs simultanément.
- Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau.



## 2. Les caractéristiques d'une base de données.

- **Données structurées** : les informations contenues dans une base de données sont réparties en enregistrements , chaque enregistrement ayant une structure bien définie.
- **Données non redondantes** : Une même information ne sera pas répétée plusieurs fois dans la base de données.
- **Données cohérentes** : Il ne doit pas être permis d'enregistrer dans une base des informations incohérentes entre elles.
- **Données accessibles** : obtention de résultats aux interrogations .
- **Indépendance des programmes et des données** : La base de données doit être indépendante des programmes qui y ont accès.
- **Sécurité des données stockées** : la base de données doit permettre un système de sécurité permettant de gérer les droits d'accès aux informations par les utilisateurs.



### 3. Les avantages d'une base de données

#### **La centralisation**

Les données sont saisies une seule fois ce qui évite la redondance et peuvent être utilisées par plusieurs programmes et plusieurs utilisateurs.

#### **Intégrité des données**

Une base de données est soumise à des règles fondamentales (unicité, référence, valeur) permettant d'assurer sa cohérence.

#### **Indépendance des données et programmes**

Les données sont décrites indépendamment des programmes.

#### **Partage des données**

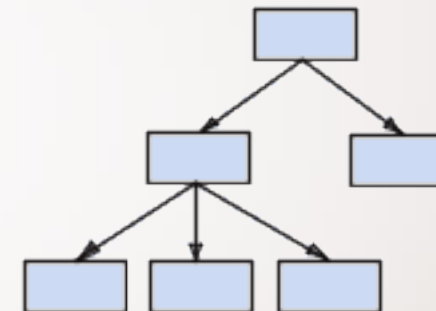
Plusieurs utilisateurs accèdent au même temps à la base sans perturber son fonctionnement.

## 4. Les modèles des bases de données

On distingue plusieurs modèles de bases de données ; la différence entre ces modèles est la représentation des liens entre les données de la base.

### ❑ Le modèle hiérarchique :

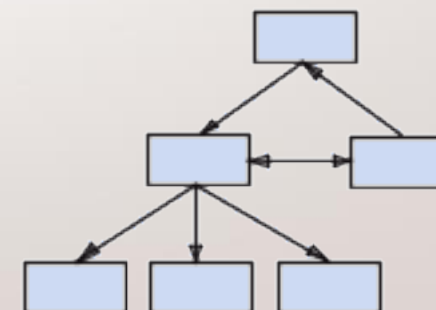
Le modèle hiérarchique organise les données dans une structure arborescente, où chaque enregistrement dispose d'un seul parent (racine). Les données sont classées hiérarchiquement, selon une arborescence descendante. Il s'agit du premier modèle de SGBD.



### ❑ Le modèle réseau :

Le modèle réseau est une extension du modèle hiérarchique qui autorise des relations plusieurs-à-plusieurs entre des enregistrements liés, ce qui implique plusieurs enregistrements parents.

Contrairement au modèle hiérarchique, la structure n'est plus forcément arborescente dans le sens descendant.

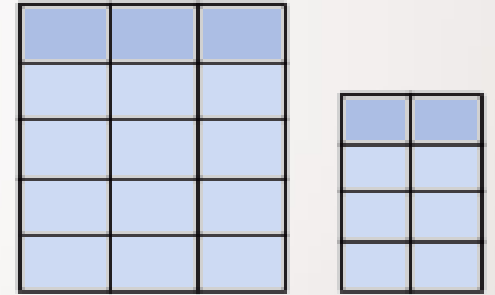


# 4. Les modèles des bases de données

## ❑ Le modèle relationnel

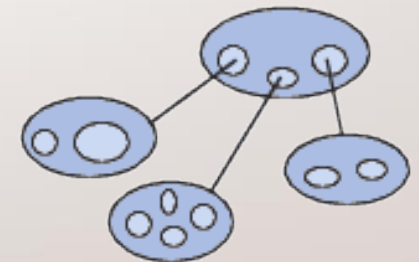
Les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations.

Les bases de données relationnelles sont généralement écrites en langage SQL (Structured Query Language).



## ❑ Le modèle objet

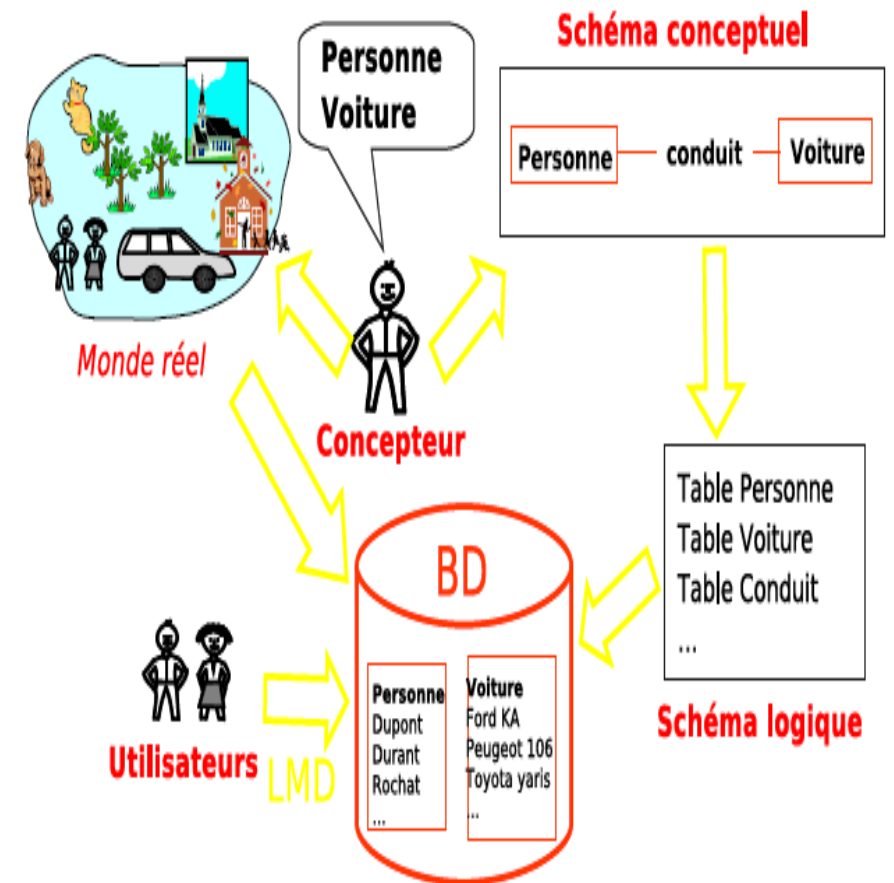
Ce modèle définit une base de données comme une collection d'objets, ou d'éléments logiciels réutilisables, associés à des caractéristiques et des méthodes. Les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées *classes* présentant des données membres.



# 5. Les niveaux de développement d'une base de données

On distingue 4 niveaux de représentation des données :

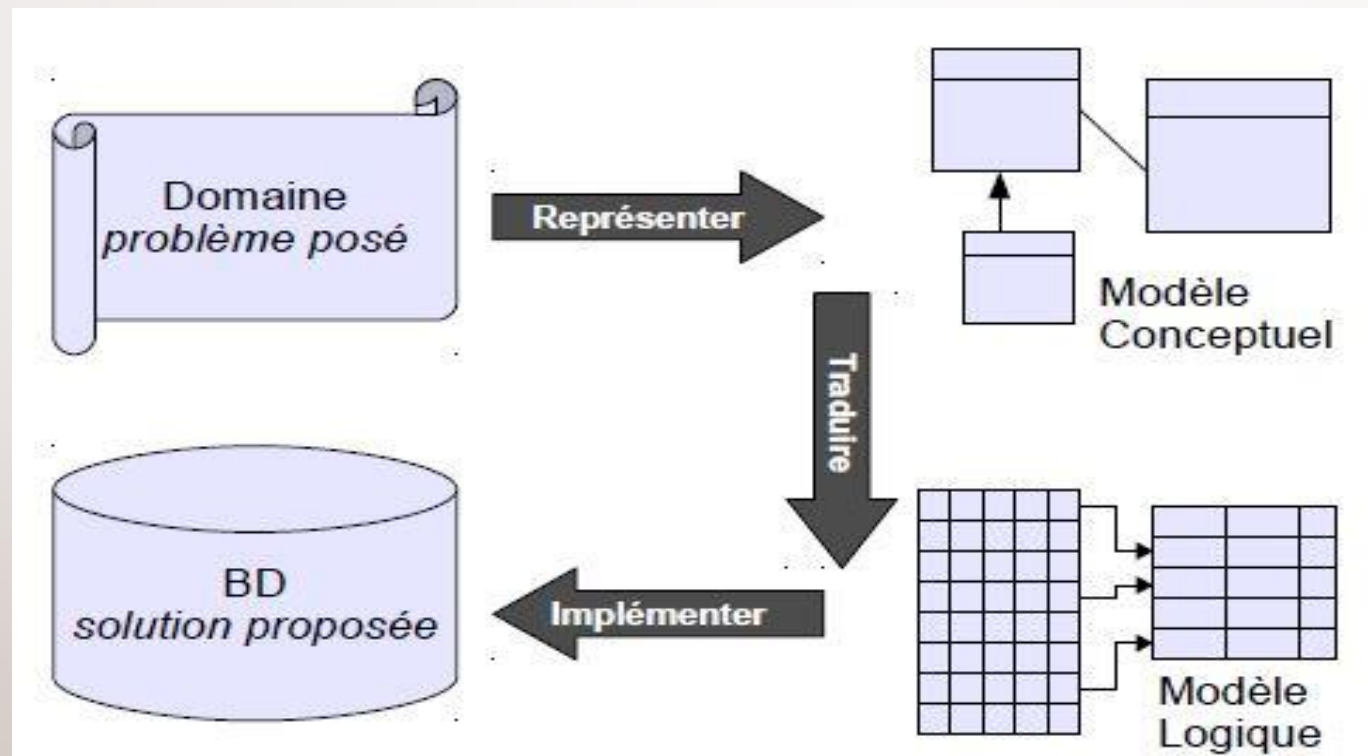
- **Le niveau externe** concerne la manipulation de la BD par des applications et définit les vues des utilisateurs.
- **Le niveau conceptuel** est indépendant de toute technologie (indépendant du SGBD) ; il concerne la définition des données. → MCD (modèle conceptuel des données)
- **Le niveau logique** qui structure les données selon le modèle relationnel, Il définit l'arrangement des informations au sein de la base de données. → MLD (modèle logique des données).
- **Le niveau interne** (physique) : il définit la façon selon laquelle sont stockées les données et les méthodes pour y accéder.



## 6. Méthodologie de conception d'une base de données

La réalisation d'une base de données est une tâche complexe, le découpage en étapes permet de gérer cette complexité.

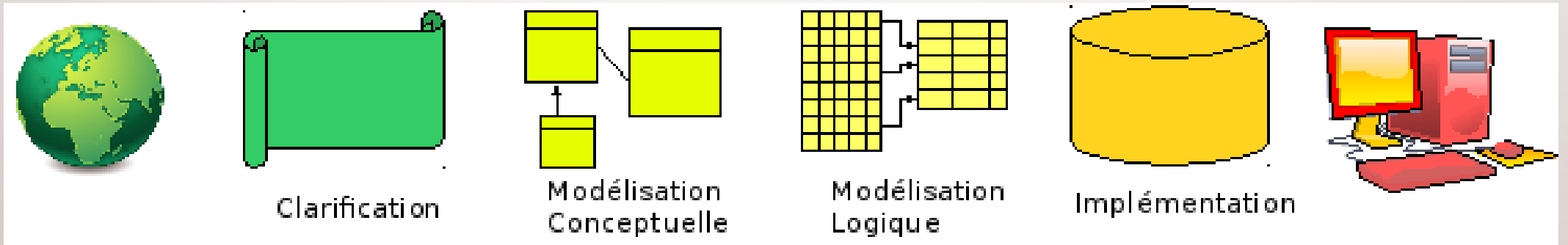
Le processus de conception d'une base de données est le suivant :



# 6. Méthodologie de conception d'une base de données

Pour concevoir une base de données relationnelle, on distingue 4 étapes :

1. L'analyse (Clarification)
2. La modélisation conceptuelle
3. La modélisation logique
4. L'implémentation



# 6. Méthodologie générale de conception d'une base de données

## 6.1. Analyse

L'analyse de la situation existante et des besoins (clarification).

Elle consiste à étudier le problème et à consigner dans un document, les besoins, les choix, les contraintes.

La phase d'analyse de l'existant et des besoins est une phase essentielle et complexe. Elle doit aboutir à des spécifications générales qui décrivent en langage naturel les données manipulées, et les traitements à effectuer sur ces données.

Voici quelques moyens de recueillir des informations avant de créer la base de données :

- ❖ Interroger les personnes qui vont l'utiliser
- ❖ Analyser des formulaires d'entreprise, tels que des factures, des feuilles de présence, des enquêtes
- ❖ Passer en revue tout système de données existant (en incluant tous les fichiers physiques et numériques)



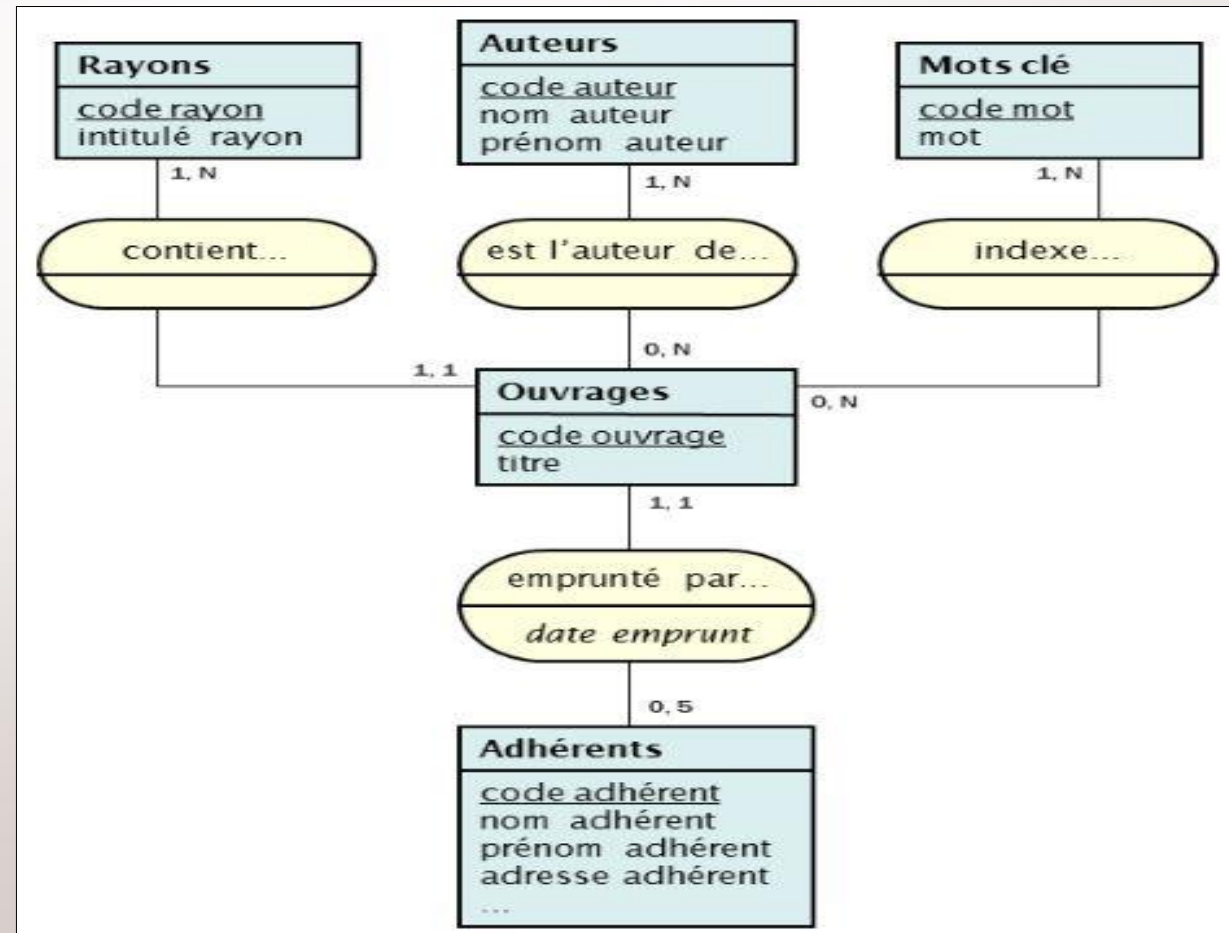
# 6. Méthodologie de conception d'une base de données

## 6.2. Modélisation conceptuelle

L'objectif d'une modélisation conceptuelle est de représenter le problème à l'aide de **représentations graphiques et partiellement formelles, indépendamment** de toute contrainte **technologique**.

Un modèle conceptuel de données permet d'avoir une **lecture accessible à tous** et donc un bon **outil de communication** entre les acteurs techniques et non techniques.

La modélisation conceptuelle en bases de données relationnelle était à l'origine faite avec le formalisme E-A (Entité-Association) de la méthode MERISE.





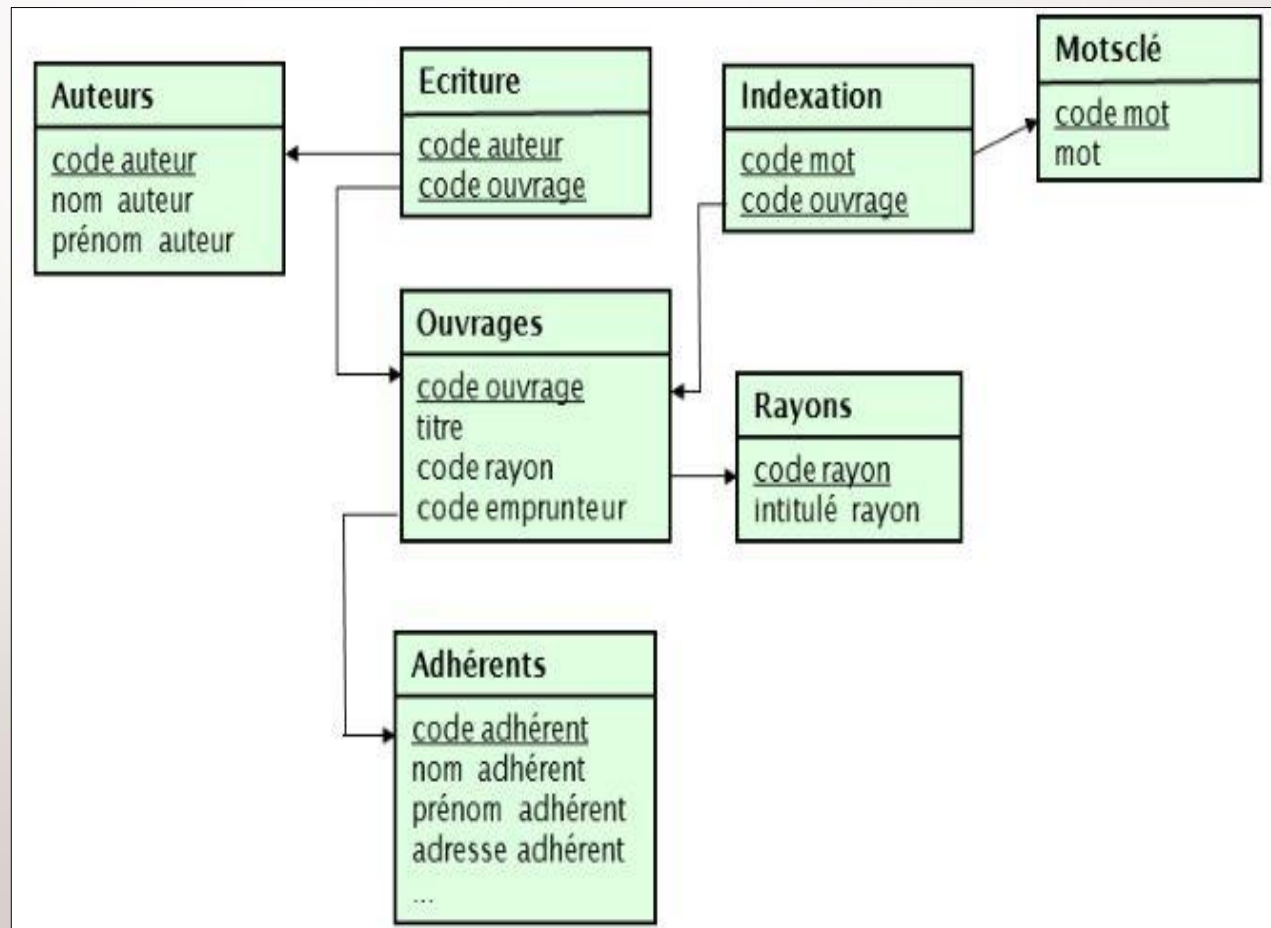
## 6. Méthodologie générale de conception d'une base de données

### 3. Modélisation logique.

Un modèle logique de données est une description, au moyen d'un langage formel, d'un ensemble de données.

C'est un schéma permet de décrire la structure d'une base de données, en décrivant l'ensemble des types de données de la base. Une instance de base de données est constituée d'un ensemble de données qui respectent le schéma de la base.

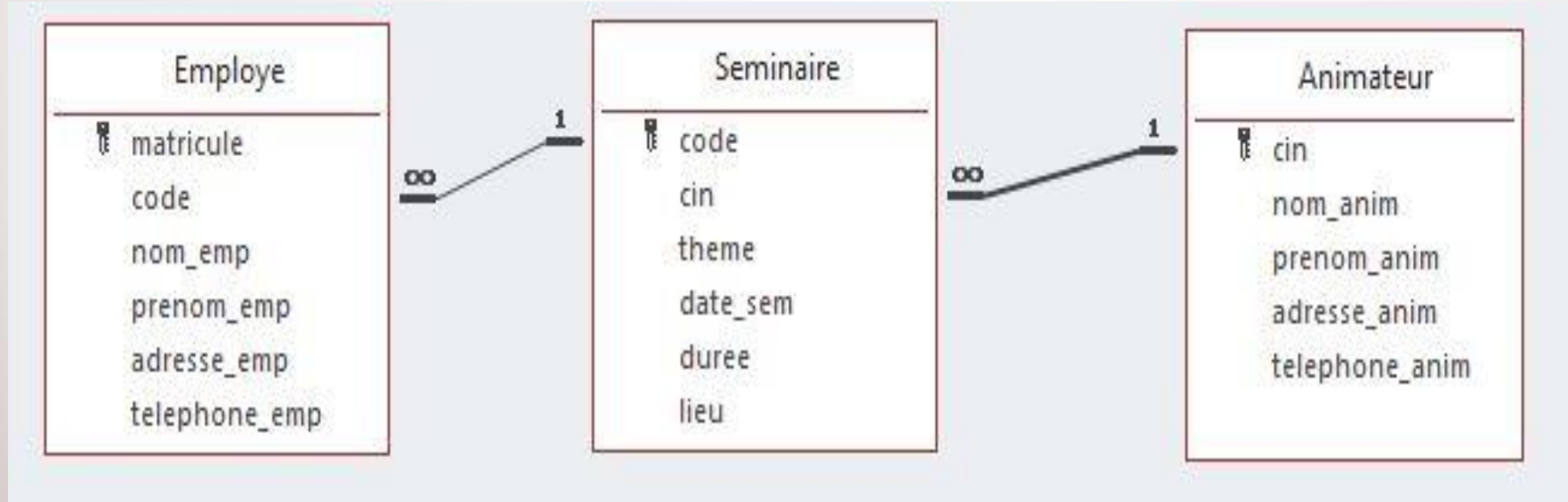
Le niveau logique est le lien entre le niveau conceptuel et l'implémentation effective de l'application. Cette phase permet de modéliser la structure selon laquelle les données seront stockées dans la base de données.



## 6. Méthodologie générale de conception d'une base de données

### 6.4. Implémentation

Elle correspond aux choix techniques, en terme du système de gestion de bases de données « **SGBD** » choisi et à leur mise en œuvre. (Access, MySql, Sql Server, Oracle ...)



**Un SGBD c'est quoi !!**

# 7. Le système de gestion de bases de données (SGBD).

## 7.1. Définition

Afin de pouvoir manipuler et contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir.

La gestion de la base de données se fait grâce à un système appelé **SGBD** (système de gestion de bases de données) ou en anglais DBMS (Database management system).

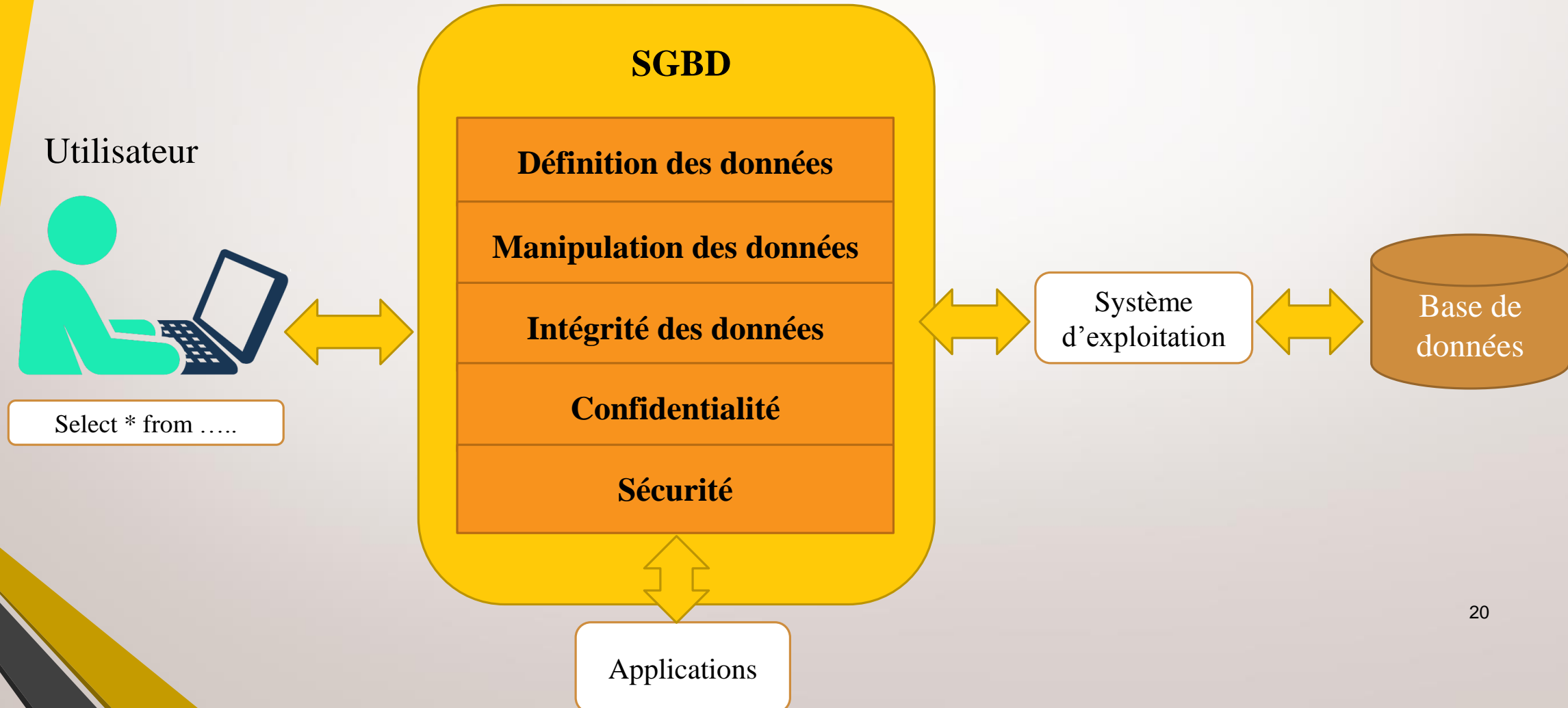
Le SGBD est un ensemble de services (applications logicielles) permettant de gérer les bases de données, c'est-à-dire :

- permettre l'accès aux données de façon simple
- autoriser un accès aux informations à de multiples utilisateurs
- manipuler les données présentes dans la base de données (insertion, suppression, modification).

Un Système de Gestion de Base de Données peut être défini comme un ensemble de logiciels prenant en charge la structuration, le stockage, la mise à jour et la maintenance des données. Autrement dit, il permet de décrire, modifier, interroger et administrer les données. C'est, en fait, l'interface entre la base de données et les utilisateurs.

# 7. Le système de gestion de bases de données (SGBD).

## 7.2. Les fonctions d'un SGBD



# 7. Le système de gestion de bases de données (SGBD).

## 7.2. Les fonctions d'un SGBD

### a. La définition des données :

Le SGBD offre un **L**angage de **D**escription de **D**onnées (**LDD**), qui permet de décrire :

- La liste des entités (objets).
- La liste des attributs de chaque objet.
- La liste des liens entre les objets.
- La liste des contraintes (une contrainte est une règle obligatoire qui peut s'appliquer sur un objet, un attribut ou un lien).

### **Exemple :**

- Liste des objets : client, location, voiture
- Liste des attributs : Nom\_Client, Adresse\_Client ...
- Contrainte : un client ne peut pas louer plus qu'une voiture à la fois.

# 7. Le système de gestion de bases de données (SGBD).

## 7.2. Les fonctions d'un SGBD

### b. La manipulation des données

Le SGBD offre un **L**angage de **M**anipulation de **D**onnées (**LMD**), qui permet, en utilisant des **requêtes**, de réaliser :

- Des recherches de données,
- Des créations de données,
- Des modifications de données,
- Des suppressions de données.

#### **Exemples :**

- ✓ Insertion/ajout de nouvelles voitures.
- ✓ Modification de l'adresse d'un client.
- ✓ Suppression d'une voiture.
- ✓ Recherche des clients fidèles (qui louent souvent des voitures)

# 7. Le système de gestion de bases de données (SGBD).

## 7.2. Les fonctions d'un SGBD

### c. L'intégrité des données

L'intégrité permet d'assurer la cohérence et la fiabilité des données en respectant certaines règles appelées **contraintes d'intégrité**.

#### Exemples :

- ✓ Dans la base de données d'un lycée, l'attribut note doit contenir des valeurs comprises entre 0 et 20.
- ✓ Un matricule d'une voiture est unique.

### d. La gestion des accès concurrents

Le SGBD offre des mécanismes de gestion de conflits (problèmes) des accès simultanés. Parmi ces mécanismes : permettre un accès multiple lors de la consultation des données ou verrouiller les données en cas de modification pour certains utilisateurs.

**Exemple** : Interdire la modification des données pour un groupe d'utilisateurs et l'autoriser pour d'autres



# 7. Le système de gestion de bases de données (SGBD).

## 7.2. Les fonctions d'un SGBD

### e. La confidentialité

Le SGBD assure la confidentialité des données par le biais de mots de passe et de privilèges d'accès.

**Exemple :** seul le directeur peut modifier l'affectation d'un étudiant.

### f. La sécurité de fonctionnement

Le SGBD offre des mécanismes permettant de remettre rapidement la BD dans un état opérationnel en cas d'incident matériel ou logiciel. Elles sont basées sur la journalisation (historique) de la BD et leur réexécution automatique.

**Exemple :** Sauvegarde de la BD une fois par semaine ; Restauration de la BD en cas de panne.



# A retenir :

**Une base de données : c'est quoi ?  
Caractéristiques ? Avantages ?  
Modèles? Niveaux ? Méthodologie de  
conception ?**

**SGBD :  
Définition et  
fonctions**



# **Chapitre II: Modélisation Conceptuelle (Modèle E-A)**

# Introduction

Il est difficile de modéliser un domaine sous une forme directement utilisable par un SGBD. Une ou plusieurs modélisations intermédiaires sont donc utiles, le modèle entités-associations constitue l'une des premières et des plus courantes. Ce modèle permet une description naturelle du monde réel à partir des concepts d'entité et d'association.

Ce modèle, utilisé pour la phase de conception, s'inscrit notamment dans le cadre d'une méthode plus générale et très répandue : **Merise**.

MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) est certainement le langage de spécification le plus répandu dans la communauté de l'informatique des systèmes d'information, et plus particulièrement dans le domaine des bases de données.

Une représentation Merise permet de valider des choix par rapport aux objectifs, de quantifier les solutions retenues, de mettre en œuvre des techniques d'optimisation et enfin de guider jusqu'à l'implémentation.

# 1. Présentation de la méthode Merise

## 1.1. Définition

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements. La séparation des données et des traitements assure une longévité du **SI**.

La démarche de développement d'un système d'information est conduite suivant 3 axes appelés cycles.

- **Le cycle de vie** : Se situe sur une échelle de temps du point de départ à l'exploitation du système.
- **Le cycle de décision** : représente l'ensemble des choix qui doivent être fait durant le déroulement du cycle de vie.

- **Le cycle d'abstraction** : A pour but de découper le **SI** en niveaux :

Le niveau conceptuel, le niveau logique et le niveau technique.

A chaque niveau correspondent une préoccupation et un ensemble de **modèles** pour la<sup>28</sup> représentation des données et de traitements.

# 1. Présentation de la méthode Merise

## 1.2 Approche par étapes :

Suivant les 3 cycles on définit les étapes de la réalisation de projet informatique.

### 1. Schéma directeur

le système d'information est étudié dans sa globalité et de manière très générale.  
Il est ensuite découpé en DOMAINES.

### 2. Etude préalable

Chaque domaine fait l'objet d'une étude préalable. On commence par analyser l'existant pour construire les modèles et proposer une solution

### 3. Etude détaillée

la solution choisie est étudiée en détail.  
Spécifications complètes du futur SI. Point de vue de l'utilisateur (externe).

### 4. Etude Technique

Spécifications complètes du futur SI. Point de vue du réalisateur (interne).

### 5. Réalisation

Elle comporte pour chaque application : la programmation, les tests et le lancement progressif du nouveau système.

### 6. Maintenance

Mise à jour des programmes suite à des erreurs constatées ou suite à des modification qui se sont opérées dans l'organisation.

# 1. Présentation de la méthode Merise

## 1.3. Niveaux d'abstraction

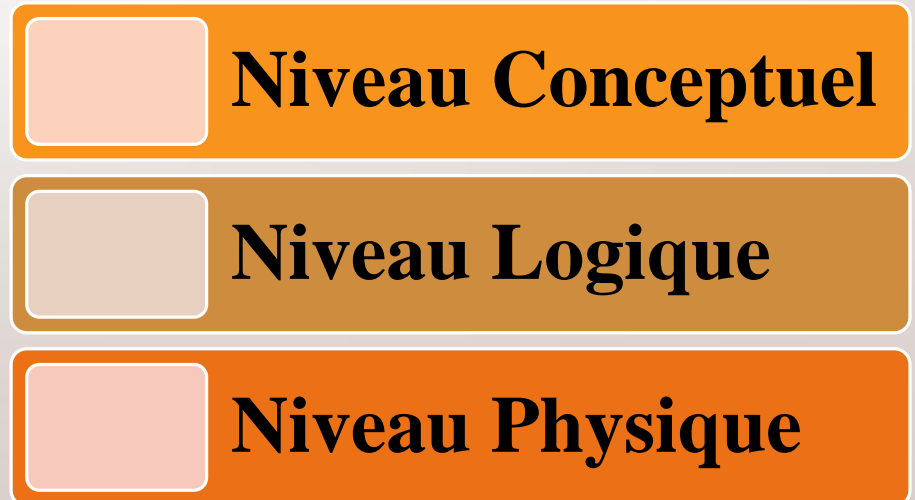
La méthode MERISE est basée sur la séparation des données et des traitements.

MERISE utilise une démarche de **modélisation à trois niveaux**. A chaque niveau correspondent à un modèle pour la représentation des données et à un modèle pour la représentation de traitements.

Un formalisme de représentation est associé à chaque modèle.

L'ensemble de ces trois niveaux constitue le **cycle d'abstraction** qui met en évidence les règles qui régissent le SI.

la méthode merise distingue trois niveaux d'abstraction



# 1. Présentation de la méthode Merise

## 1.3. Niveaux d'abstraction

- Au **niveau conceptuel**, on s'interroge sur l'essence même du système d'information. Il s'agit de répondre aux questions : quoi faire? Que veut-on faire? Avec quelles données? On développe à ce niveau le modèle conceptuel des données (MCD) et le modèle conceptuel des traitements (MCT).
- Le **niveau logique/organisationnel** permet de tenir en compte les problèmes organisationnels: à partir des modèles conceptuels de données et de traitements et en faisant intervenir les contraintes d'organisation, on élabore le modèle logique de données (MLD) et le modèle organisationnel des traitements (MOT).
- Le **niveau physique** est caractérisé par la prise en compte des contraintes technologiques : matériel, logiciel, humain ... etc. A ce niveau, le modèle physique des données et le modèle opérationnel des traitements.

NIVEAU	DONNEES	TRAITEMENTS
Conceptuel	MCD : modèle conceptuel des données	MCT : modèle conceptuel de traitement
Logique/organisationnel	MLD : modèle logique des données	MOT : modèle organisationnel des traitement
Physique	MPD : modèle physique des données	MOPT : modèle opérationnel des traitement

## 2. Modélisation conceptuelle

### 2.1 Définition

Le modèle conceptuel des données (MCD) décrit les entités du monde réel, en terme d'objets, de propriétés et de relations, indépendamment de toute technique d'organisation et d'implantation des données. Ce modèle se concrétise par un schéma entités-associations représentant la structure du système d'information, du point de vue des données.

L'objectif poursuivi est la définition et l'élaboration de la structure globale des données de manière indépendante de toute contrainte organisationnelle ou technologique.

La structure est appelé **modèle conceptuel des données**.

MCD



## 2. Modélisation conceptuelle

### 2.2. Concepts de base constituant modèle E-A :

La représentation du modèle entités-associations s'appuie sur trois concepts de base :

- l'objet ou entité,
- l'association,
- la propriété.

#### a. Entité

Une ENTITE est un objet concret ou abstrait qui a une existence propre, qu'on peut cerner et définir indépendamment de tout autre objet.

##### Exemples d'entités concrètes :

- Le client KADIRI Mohammed,
- Le fournisseur DRISSI Omar,
- L'exemplaire du livre « COMPRENDRE MERISE »
- Une table, une machine ... etc.

##### Exemples d'entités abstraites :

- Le service achat d'une société,
- La société IRMA Maroc,
- La classe d'étudiants G11,
- La matière « Chimie ».

## 2. Modélisation conceptuelle

### 2.2. Concepts de base constituant modèle E-A :

#### b. Association

Une association est une relation qui existe ou qui met en liaison deux ou plusieurs entités.

##### Exemples

- ❖ « Mohamed est marié à Fatima » l'association mariage lie les deux entités Mohamed et Fatima.
- ❖ « L'étudiant Amine Semlali appartient à la classe GI1 » est l'expression d'une relation d'appartenance.

#### c. Propriété

Les propriétés décrivent l'entité ou l'association. Elles apportent l'information utile et nécessaire au système d'information.

##### Exemples

- L'entité **client** est décrite par les propriétés : **Code client, nom, téléphone ...**
- L'entité **commande** contient : **N° commande, date commande.**
- L'entité **Produit** est décrite par : **N° Produit, Libellé du produit , Prix**
- L'entité **Facture** : **N° Facture, date facture, montant facture**

## 2. Modélisation conceptuelle

### 2.2. Concepts de base constituant modèle E-A :

#### d. Identifiant Entité

Pour chaque entité, il est impératif de trouver une propriété particulière qu'on appelle clé ou identifiant et qui permet de désigner chaque occurrence de manière unique.

Par exemple le numéro de la carte d'identité nationale (N° CIN) permet de distinguer sans ambiguïté chaque citoyen Marocain. Autrement dit il ne peut pas y avoir deux individus ayant le même N° CIN.

#### Exemples

- Le **numéro d'inscription** est un identifiant pour l'entité « **Etudiant** »
- Le **référence** est un identifiant de l'entité « **Livre** »
- Le **code article** est un identifiant de l'entité « **Article** »

## 2. Modélisation conceptuelle

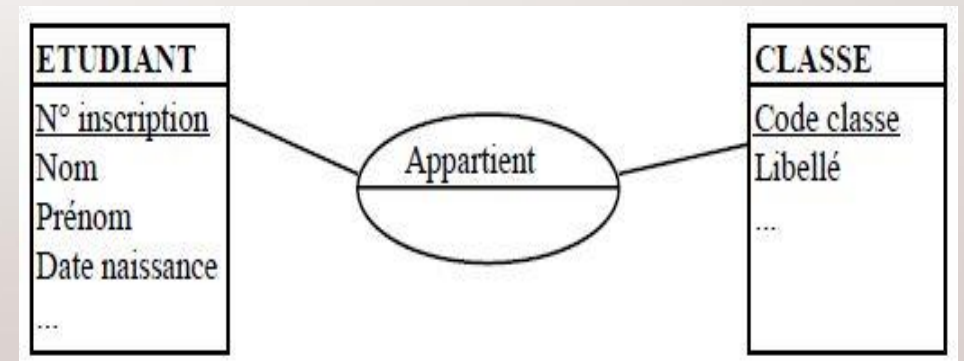
### 2.3 Représentation schématique:

#### a. Schéma Entité - Association

Une entité est représentée par un rectangle barré à l'intérieur duquel on inscrit, dans sa partie supérieure le nom de l'entité et dans l'autre la liste des propriétés en prenant soin de souligner l'identifiant.

L'association est schématisée par un ovale barré. Dans la partie supérieure, on inscrit le nom de l'association et dans l'autre, la liste des propriétés.

Le schéma suivant représente deux entités ETUDIANT et CLASSE liées par une association d'appartenance. On peut facilement lire qu'un étudiant appartient à une classe.



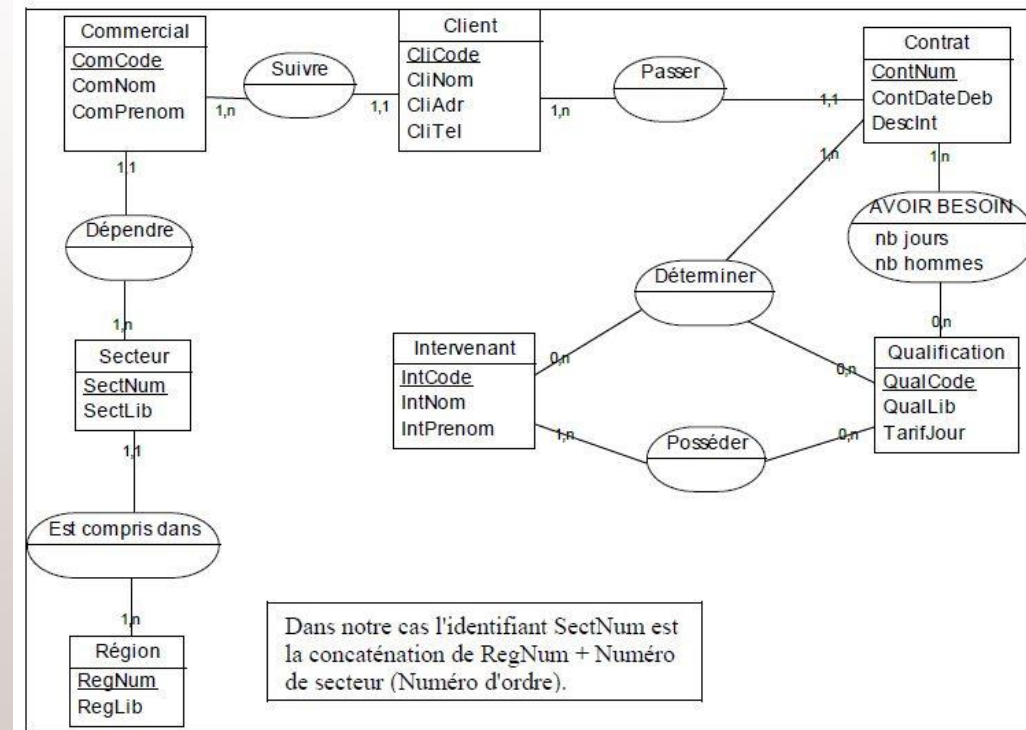
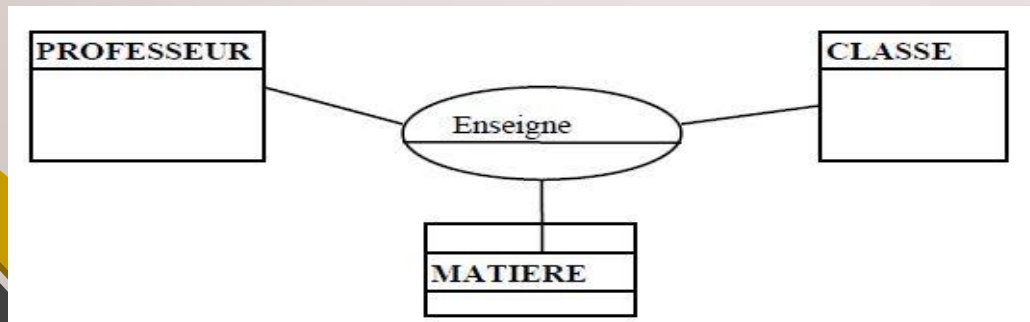
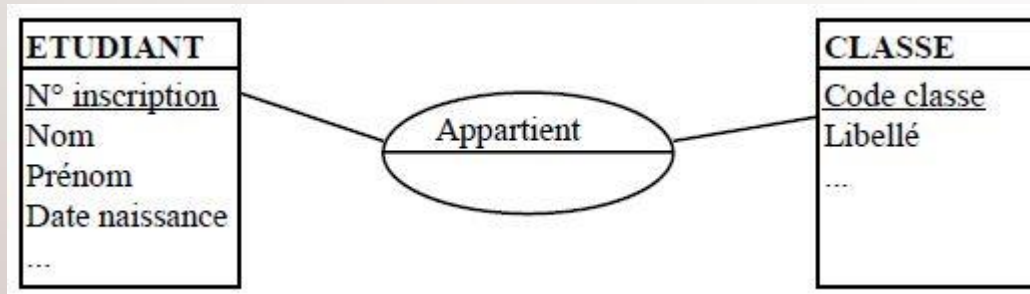
# 2. Modélisation conceptuelle

## 2.3 Représentation schématique:

### b. Dimension d'une association

Définition: C'est le nombre d'objet participant à la relation (de 1 à n).

- Une relation entre deux objets est appelée : relation **binaire**.
- Une relation entre trois objets est appelée : relation **ternaire**.
- Une relation entre n objets est appelée : relation **n-aire**.



## 2. Modélisation conceptuelle

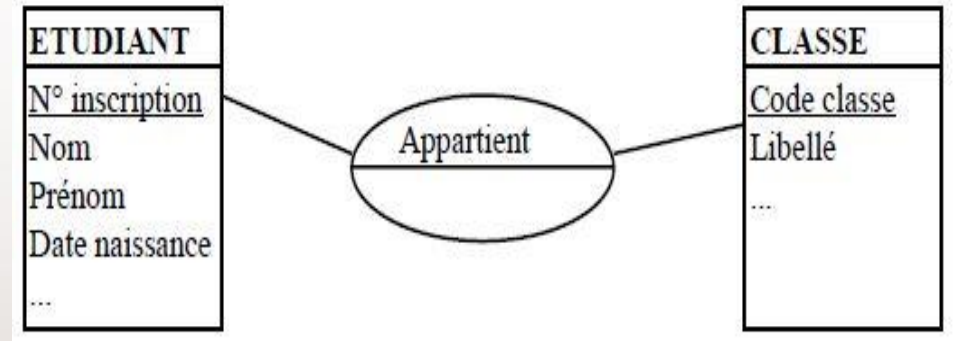
### 2.4 Les cardinalités:

Le modèle conceptuel de données doit être complété par les cardinalités qui apportent des informations importantes sur les associations.

En effet, revenons au MCD (Etudiant-Classe). On ne peut pas savoir à partir du schéma si un étudiant peut appartenir à une ou plusieurs classes. On ne peut pas non plus savoir si une classe contient un ou plusieurs étudiants.

Les cardinalités permettent d'ajouter ces précisions quantitatives.

De chaque côté de l'association et donc pour chaque entité intervenant dans l'association, on définit des cardinalités minimum et des cardinalités maximum.



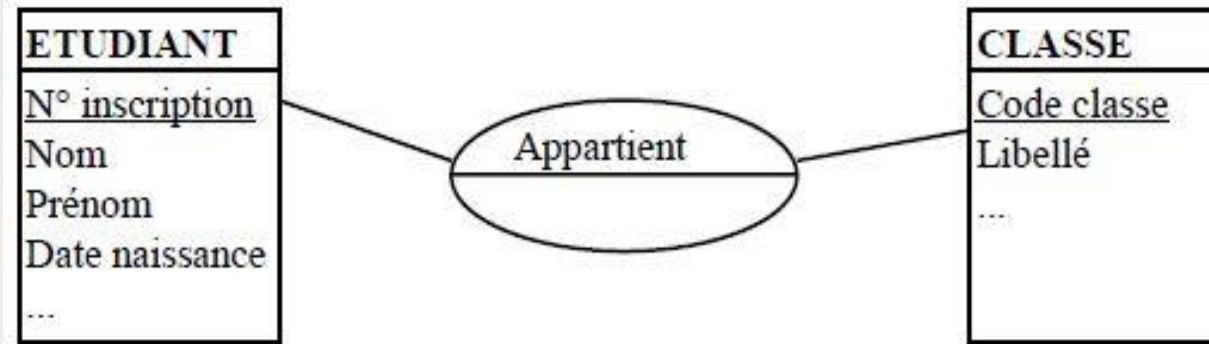
- **Cardinalité minimum** : indique le nombre minimum d'intervention d'une entité dans une association.
- **Cardinalité maximum** : indique le nombre maximum d'intervention d'une entité dans une association.



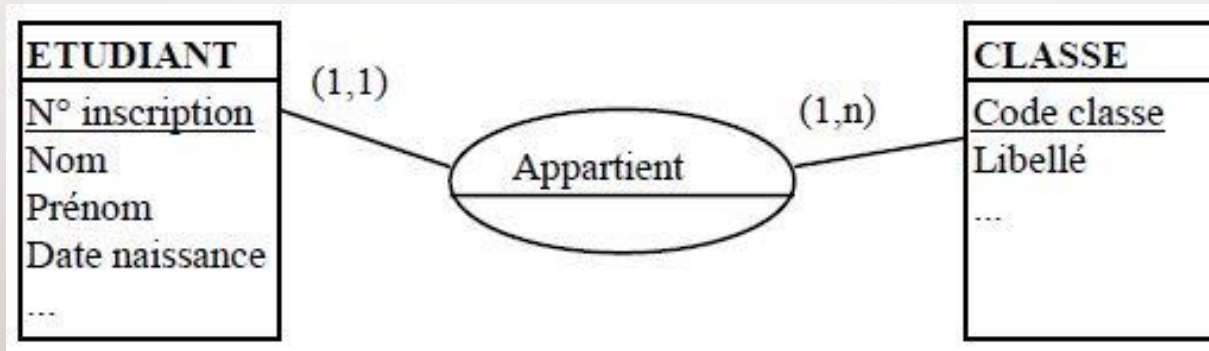
## 2. Modélisation conceptuelle

### 2.4 Les cardinalités:

Reprenons l'exemple précédent



*On sait qu'un étudiant ne peut appartenir qu'à une et une seule classe.  
Une classe peut contenir plusieurs étudiants.*



Du côté de l'entité « ETUDIANT » on a inscrit les cardinalités (1,1) parce que un étudiant appartient à une et une seule classe.

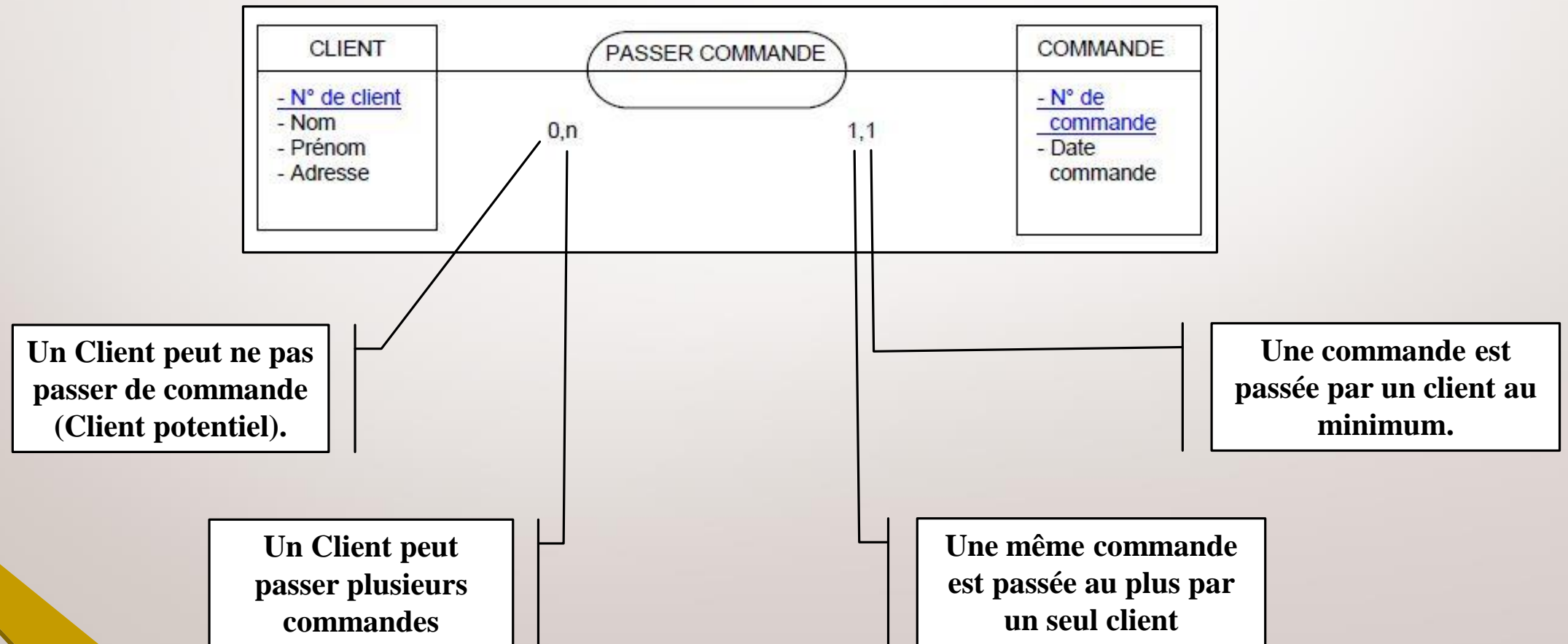
Du côté de l'entité « CLASSE » on a inscrit les cardinalités (1,n) car une classe contient au moins un étudiant et au plus n étudiants.

## 2. Modélisation conceptuelle

### 2.4 Les cardinalités:

En générale, la borne inférieure est 0 ou 1, la borne supérieure est 1 ou n.

**Exemple :**



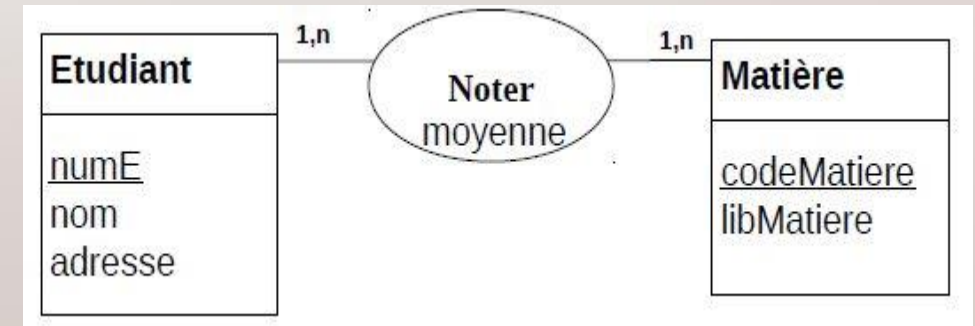
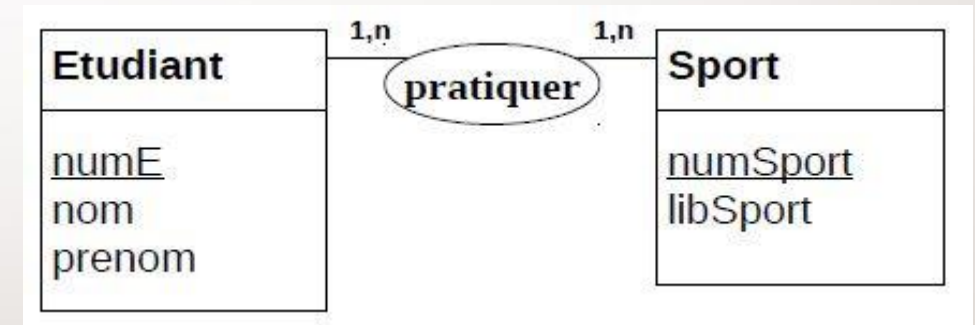
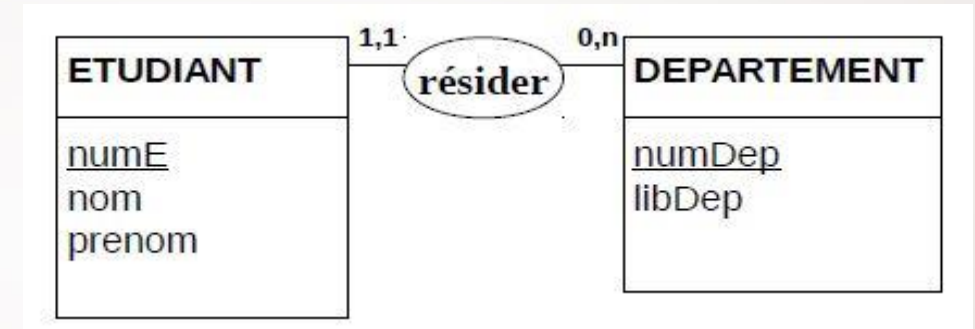


## 2. Modélisation conceptuelle

### 2.4 Les cardinalités:

#### Exemples :

- Association de type [1,N]
- Association de type [N,N]
- Association de type [n,n] avec propriété



## 2. Modélisation conceptuelle

### 2.4 Les cardinalités:

#### Exercice d'application 1:

On souhaite gérer un parc d'animaux, on cite les entités intervenantes dans ce système.

**Animal, Espèce (classe d'animaux), Personne, Aliment.**

#### Exemples :

- **Animal** : chat, dauphin, serpent, cheval ...
- **Espèce** : mammifère, reptile, poisson ...
- **Personne** : Rachid, Jacque, Sarah ...
- **Aliment** : caviar, plancton, fromage ...

En prenant en considération les règles suivantes :

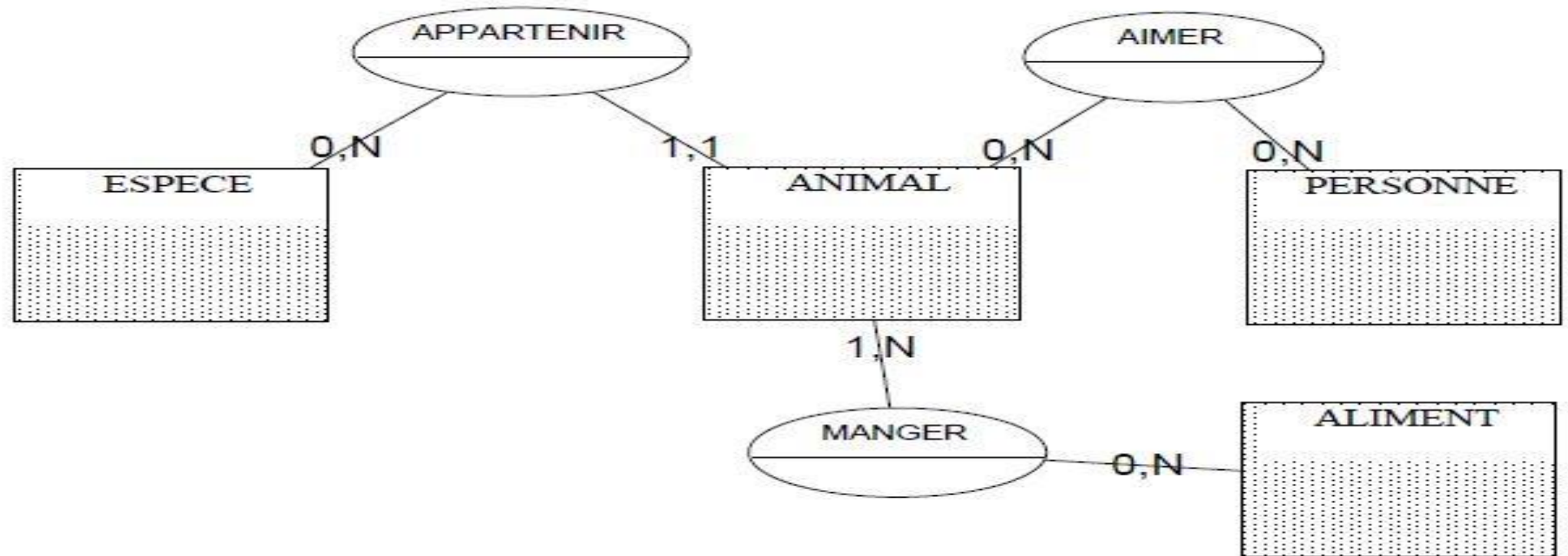
- ✓ *Un animal appartient à une espèce et une seule.*
- ✓ *Un animal peut ne pas être aimé par aucune personne*
- ✓ *Un animal mange au minimum un aliment.*
- ✓ *Un aliment mangé par personne n'est pas un aliment.*

#### Travail à faire :

- 1- Etablir un modèle permettant de relier les entités ci-dessus par des associations convenables.
- 2- Inscire les cardinalités sur le modèle.

## Exercice d'application 1 :

Corrigé :



## 2. Modélisation conceptuelle

### 2.4 Les cardinalités:

#### **Exercice d'application 2 :**

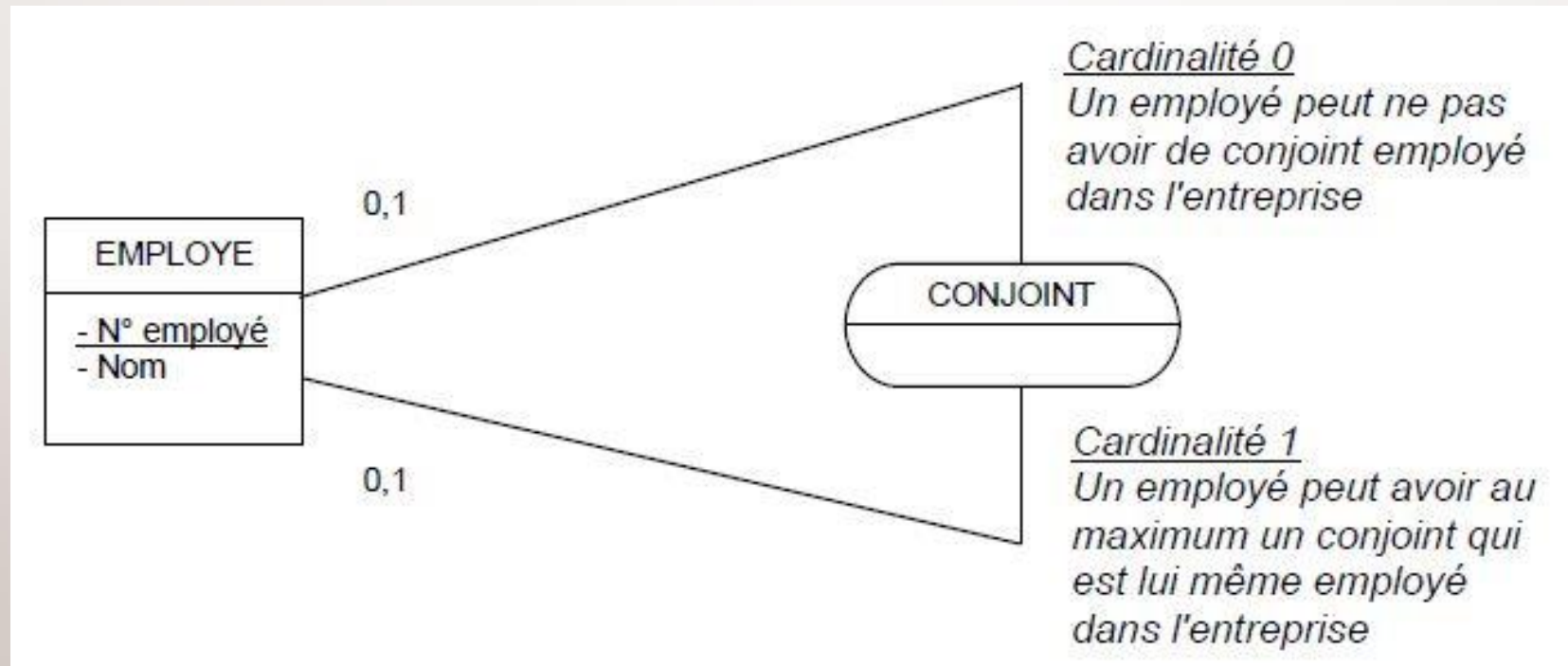
(Voir exercice 1 TD1)

## 2. Modélisation conceptuelle

### 2.5 Associations spéciales

**Association réflexive** : C'est l'association d'un objet sur lui-même.

**Exemple** : Considérons l'entité **EMPLOYE**, d'une entreprise, et l'association **CONJOINT**.

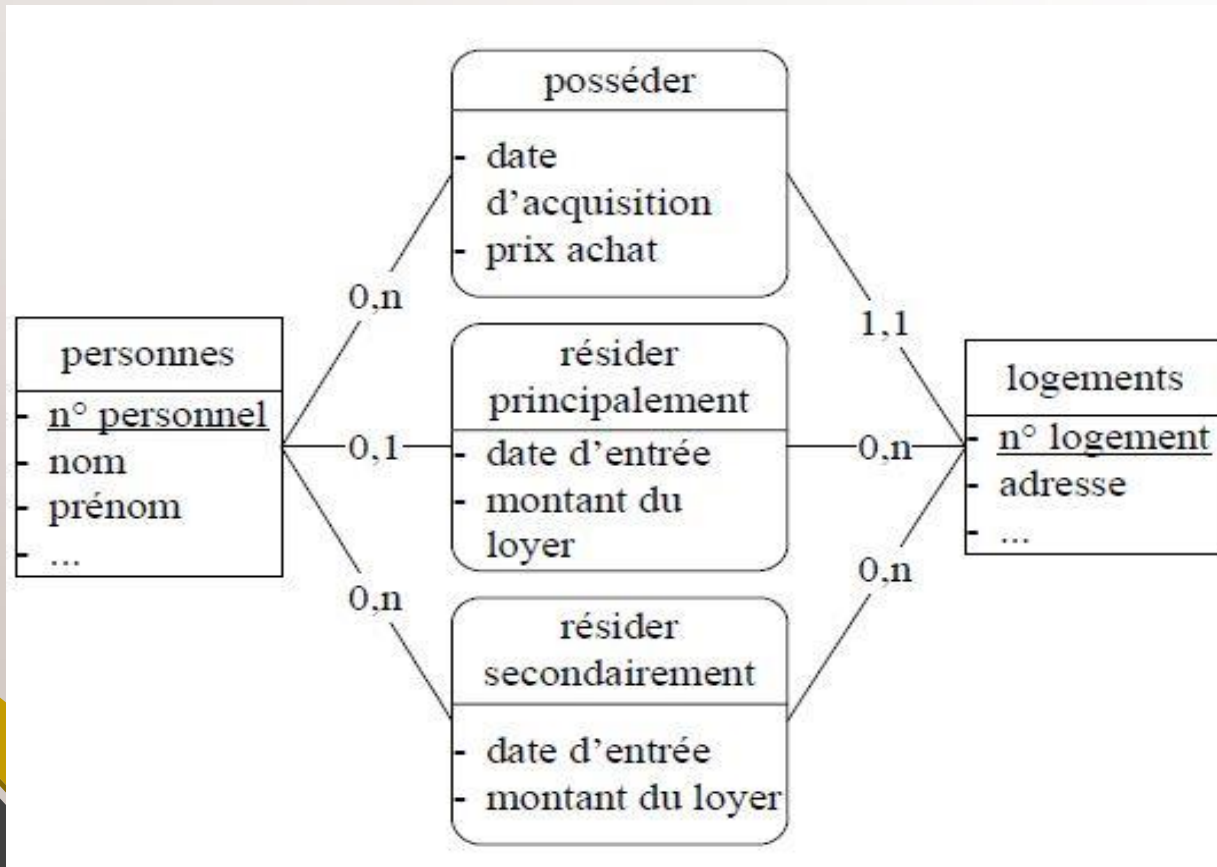


## 2. Modélisation conceptuelle

### 2.5 Associations spéciales

**Association multiple :** Deux entités distinctes peuvent avoir plusieurs lien de dépendance. Dans ce cas elles vont être reliées par plusieurs associations. C'est ce qu'on appelle association multiple.

#### Exemple



Dans cet exemple issu d'une agence immobilière, une personne peut être propriétaire, résider principalement ou résider secondairement dans un logement géré par l'agence.

Les logements qui ne sont pas gérés par l'agence ne figurent pas dans l'entité des logements, ce qui explique certaines cardinalités 0 du schéma.

Nous supposons également qu'un logement n'est détenu que par une seule personne et que ce propriétaire figure obligatoirement dans l'entité des personnes.

## 2. Modélisation Conceptuelle

### 2.6 Construction du MCD

La construction du MCD passe par les étapes suivantes :

**1. Recueil des informations**

**2. Construction Dictionnaire de données**

**3. Epuration du dictionnaire**

**4. Dépendances fonctionnelles**

**5. Elaboration MCD**



# 2. Modélisation Conceptuelle

## 2.6 Construction du MCD

### a. Recueil des informations

Le recueil d'information est la première phase à l'informatisation d'un SI. Toutes les informations concernant le sujet doivent être rassemblées.

Cette phase de recueil est effectuée en plusieurs niveaux :

#### Les techniques et les outils :

Le premier problème à résoudre est de trouver l'information. On recherchera dans :

- Les documents
- Les règlements
- Les normes, les procédures,
- Les fichiers
- ...

**Localisation  
des  
informations**

Pour recenser les informations, on utilise essentiellement :

- L'étude de documents
- Les entretiens
- Parfois les questionnaires

**Techniques  
de recueil**



# 2. Modélisation Conceptuelle

## 2.6. Construction du MCD

### a. Recueil des informations

#### Classification des données (Nature):

On distingue 3 classes majeures:

- Les données *élémentaires* : Par exemple, un nom, un email, ... Ces données doivent être recensées de manière exhaustive.
- Les données *calculées* ou déduites : elles sont obtenues par l'application d'un traitement mathématique ou logique. Ces données sont associées à des règles de calcul (règles de gestion).
- Les données *composées* : certaines données sont regroupées en une même entité sémantique (par exemple une adresse).

#### Types de données

Les types de données ont un sens plus restrictif que la catégorie (Nature). Alors que la catégorie s'applique à la nature d'une donnée, le type est une contrainte physique liée à la manière dont sera stockée la donnée dans le système d'information.

Les principaux types à retenir sont :

- *Alphanumérique* (AN)
- *Numérique* (on peut préciser entier, réel, monétaire...)
- *Date* (Date/Heure, Date, Heure )
- *Logique ou booléen* (L ou B)

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### b. Dictionnaire des données

Le dictionnaire des données est en fait le résultat de la phase de collecte des données. C'est un document permettant de recenser, de classer et de trier toutes les données collectées lors de la phase de recueil des informations.

Ce dictionnaire est un outil important car il constitue la référence de toutes les études effectuées ensuite.

#### **Formalisme**

Les données sont présentées dans un tableau sous l'entête suivante :

Champ	Description	Nature	Type	Taille	Remarque
-------	-------------	--------	------	--------	----------

*Avec :*

- Champ : Nom de la propriété
- Description : Signification de la propriété
- Nature : E (Elémentaire) , CA ( Calculée) , CO (Composée)
- Type : AN, N, Date ou Booléen
- Taille : longueur maximale prise par les valeurs de ce champ.
- Remarque : Observation, contrainte ...

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### b. Dictionnaire des données

**Exemple :**

Champ	Description	Nature	Type	Taille	Remarque
CodeEmp	Code employé	E	AN	10	
NomEmp	Nom employé	E	AN	25	
DateNaissEmp	Date naissance	E	Date	-	inférieure à la date courante
AdresseEmp	Adresse employé	CO	AN	30	
SalaireEmp	Salaire employé	E	N	10	

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### c. Epuration du dictionnaire

Les données recueillies fait généralement apparaître des incohérences qu'il convient d'éliminer. Ces incohérences sont de deux types:

##### a) De synonymes

Des noms différents désignent la même réalité

**Exemples** : Numéro de commande et référence commande, agent et employé,  
Marchandise et produit, TVA et Taxe ...

**On dit aussi qu'existent alors deux signifiants pour un même signifié.**

##### b) Des polysèmes

Un même nom désigne deux réalités distinctes.

**Exemples** : Café désigne l'établissement et la boisson,  
Prix désigne le prix d'achat et le prix de stockage,...

**On dit que l'on a un signifiant pour deux signifiés.**

L'épuration du dictionnaire consiste à détecter et à éliminer les synonymes et les polysèmes.

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### c. Epuration du dictionnaire

Les données calculées doivent être examinées avec soin.

La règle est la suivante :

Si une donnée calculée peut être obtenue par l'application d'un traitement à partir de données élémentaires valides, on peut la supprimer du dictionnaire.

#### Exemple :

Examinons le cas d'une entité **Employé** ayant comme propriétés : *N°*, *Nom*, *Prénom*, *DateNaissance*, *Téléphone*, *email*, *SalaireHoraire*, *NbHeures*, *SalaireBrut*.

On remarque que la propriété *SalaireBrut* est un champ calculé obtenu par l'application du calcul :

$$\textit{SalaireBrut} = \textit{NbHeures} * \textit{SalaireHoraire}$$

Les données élémentaires *NbHeures* et *SalaireHoraire* qui participent à son calcul sont présentes. On peut donc éliminer cette donnée (*SalaireBrut*) pour la phase de modélisation conceptuelle.

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### Exercice d'application 1 :

Une société X de construction de bâtiments organise des séminaires de formation pour ses employés. Des animateurs, sont recrutés et payés pour l'animation de ces séminaires. Ils sont identifiés par un nom, prénom, numéro de téléphone et par un numéro de carte d'identité nationale.

Les séminaires sont organisés par thème, ils ont un code, une date, une durée et un lieu. Un séminaire est pris en charge par un seul animateur mais un animateur peut animer plusieurs séminaires

Un employé qui désire suivre un séminaire doit remplir une fiche indiquant son nom, son prénom, son adresse, son matricule et son numéro de téléphone, et ne peut participer qu'un seul séminaire à la fois.

#### Travail à faire :

1. Recueillir toutes les données circulant dans le SI de la société X.
2. Dresser le dictionnaire de donnée associé.
3. Citer les entités intervenantes dans le SI de la société.

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

**Exercice d'application 2 :**

(Voir exercice 2 TD1)

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

Une dépendance fonctionnelle est une interrelation, un lien entre deux données ou deux groupes de données.

On distingue une source et une cible.

**Définition :** Pour une valeur source (Partie gauche), on peut déterminer une et une seule valeur cible (Partie droite).

On dit qu'une propriété B dépend fonctionnellement d'une autre propriété A et on note :

**A** —————> **B**

Autrement dit à une valeur de A correspond toujours une et une seule valeur de B. La réciproque n'est pas vraie.

**Exemple :**

**N°CIN** —————> **NOM**



## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

Les dépendances fonctionnelles ont les propriétés suivantes :

- **Union :**

Si on a deux DF ayant la même source, on peut les rassembler en une seule, en séparant les cibles par une virgule.

Si  $A \rightarrow B$  et  $A \rightarrow C$  alors on peut écrire que  $A \rightarrow B, C$

**Exemple** : Référence  $\rightarrow$  Désignation et Référence  $\rightarrow$  Prix de vente unitaire

Alors par union on a : Référence  $\rightarrow$  Désignation, Prix de vente unitaire

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

- **Transitivité :**

Si  $A \rightarrow B$  et  $B \rightarrow C$  alors on a  $A \rightarrow C$

**Exemple** : Num Médecin  $\rightarrow$  Code Service et Code Service  $\rightarrow$  Num Hopital  
Alors on a Num Médecin  $\rightarrow$  Num Hopital

Les DF qui peuvent être déduites par transitivité de deux autres DF doivent être éliminées car elles sont alors redondantes.

**Une DF doit être élémentaire et directe.**

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

**Elémentaire** : C'est l'intégralité de la source qui doit déterminer la cible d'une DF.  
Si  $P1 \rightarrow P3$  alors  $P1, P2 \rightarrow P3$  n'est pas élémentaire.

**Exemple :**       $N^{\circ} \text{ CIN, NOM} \longrightarrow \text{Prénom}$

Cette dépendance n'est pas élémentaire puisque le  $N^{\circ} \text{ CIN}$  suffit pour déterminer le prénom.

La source d'une dépendance fonctionnelle peut se composer d'une concaténation de deux ou plusieurs propriétés.

$N^{\circ} \text{ Etudiant} + \text{Code Matière} \longrightarrow \text{Note}$

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

**Directe** : La DF ne doit pas être obtenue par transitivité. Par exemple, si  $P1 \rightarrow P2$  et  $P2 \rightarrow P3$  alors  $P1 \rightarrow P3$  a été obtenue par transitivité et n'est donc pas directe.

$A \rightarrow B$  est directe s'il n'existe pas de propriété  $C$  tel que  $A \rightarrow C$  et  $C \rightarrow B$

Autrement dit la dépendance fonctionnelle n'est pas le résultat d'une transitivité.

DF	Directe (Oui / Non)
<b>Exemple :</b> N° Professeur $\rightarrow$ code matière	Oui
Code matière $\rightarrow$ nom matière	Oui
N° Professeur $\rightarrow$ nom matière	Non

Les deux premières dépendances sont directes, mais la troisième ne l'est pas car elle résulte de l'application de la transitivité

$N^{\circ} \text{ Professeur} \rightarrow \text{Code matière} \rightarrow \text{Nom matière}$

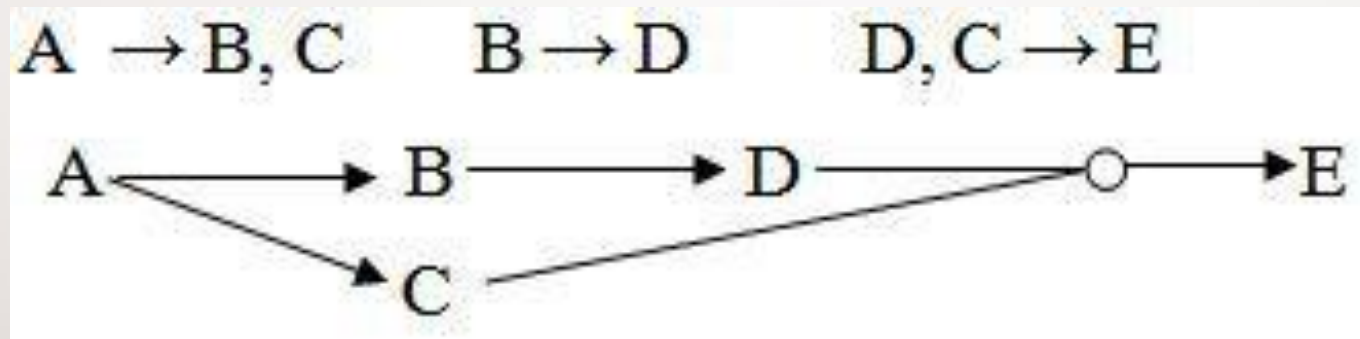
## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

**Grphe de dépendance fonctionnelles (GDF)** : Le GDF fait apparaître les dépendances fonctionnelles entre les données.

**Exemple :**



La schématisation de l'ensemble des dépendances fonctionnelles sous forme de GDF intervient à l'élaboration du modèle conceptuel de données (MCD), en respectant certaines règles.

# 2. Modélisation Conceptuelle

## 2.6. Construction du MCD

### d. Les dépendances fonctionnelles

#### Exercice d'application 1 :

Soit une liste des données recensées dans un établissement scolaire

- Nom de l'élève
- Prénom de l'élève
- email de l'élève
- Libellé matière
- Nombre d'heures
- Code filière
- Libellé filière
- Note
- Numéro de l'élève
- Numéro de la matière
- Langue d'enseignement

Les règles de gestion appliquées dans cet établissement :

**RG1** : Chaque élève est attribué à une et une seule filière.

**RG2** : Une matière est enseignée pour différentes filières avec des nombres d'heures différents.

**RG3** : Pour tout élève, chaque matière est évaluée par une note.

#### Travail à faire :

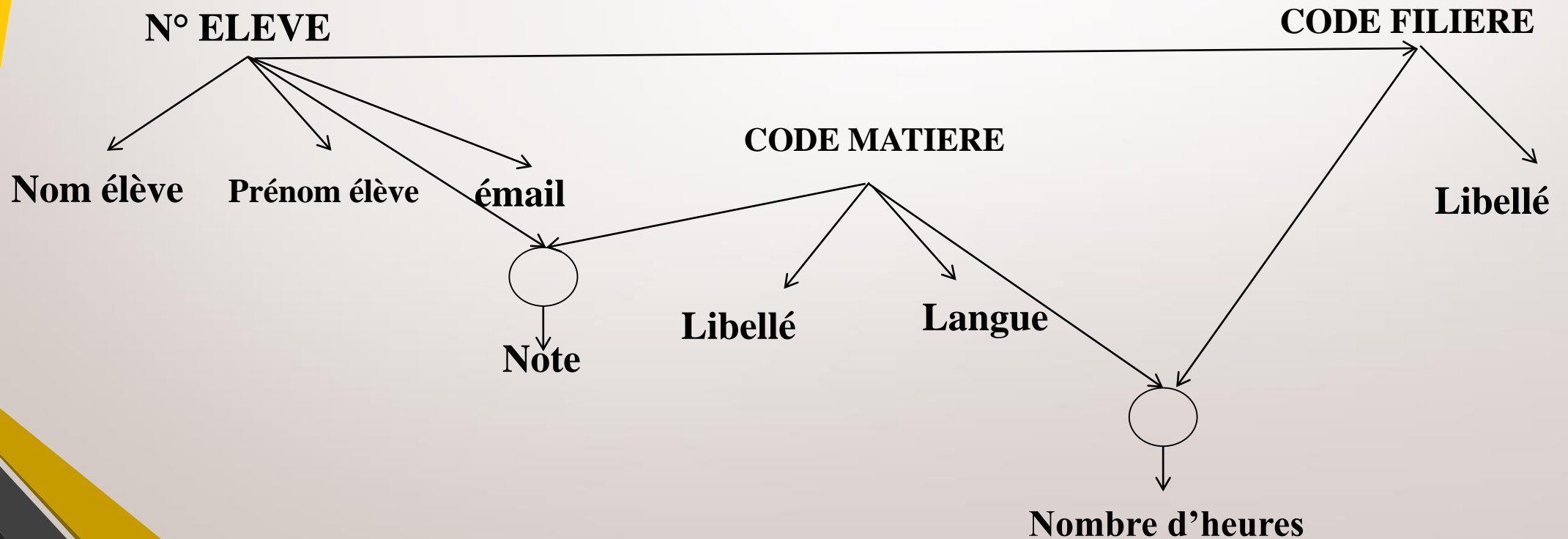
1. Citer les différentes dépendances fonctionnelles.
2. Déduire le GDF associé.

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

Corrigé :



## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

**Exercice d'application 2 :**

(Voir exercice 3 TD1)



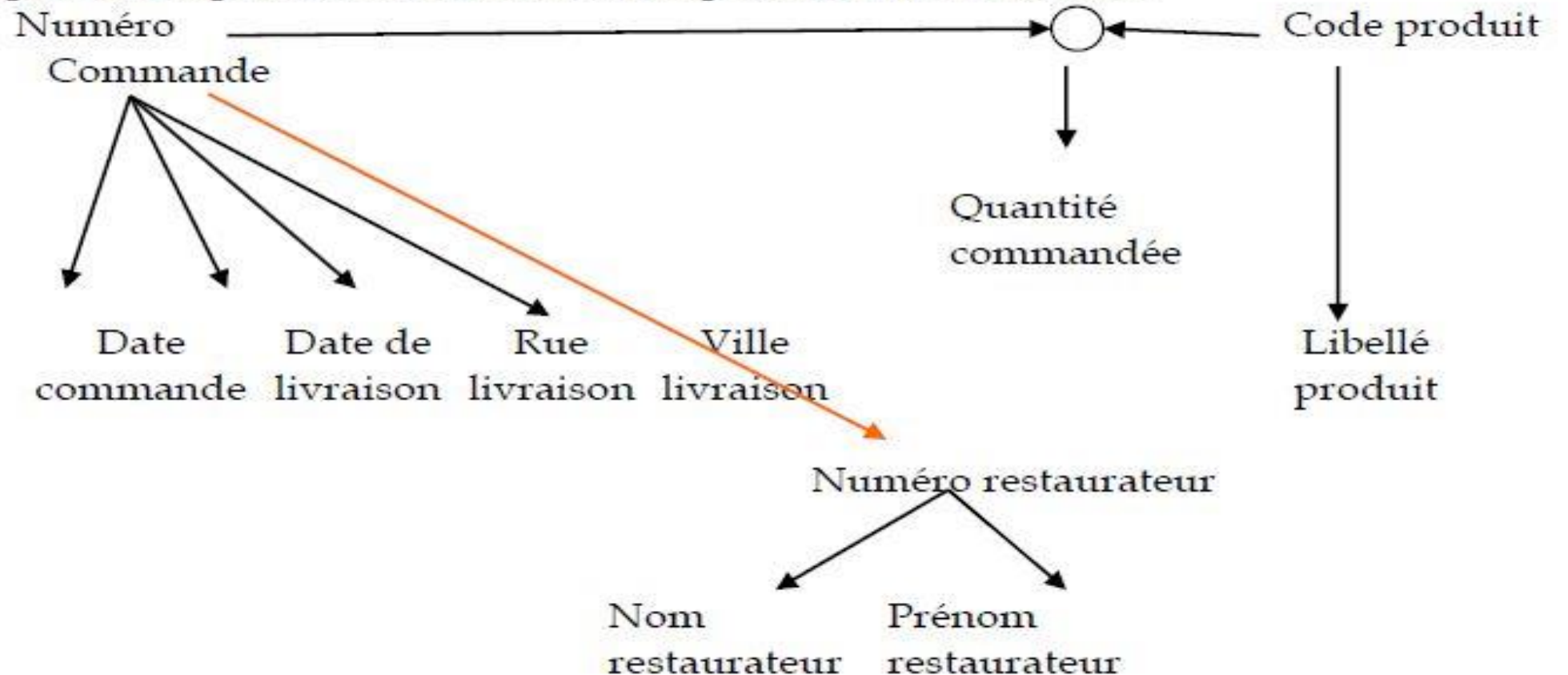
## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### d. Les dépendances fonctionnelles

Corrigé

Le graphe des dépendances fonctionnelles prend l'allure suivante :



# 2. Modélisation Conceptuelle

## 2.6. Construction du MCD

### e. Elaboration MCD

#### Les règles de passage des DF (ou GDF) au MCD :

**R1:** Toute donnée du graphe devient une propriété.

**R2:** Toute donnée source de DF devient identifiant de l'entité.

**R3:** Toute donnée cible de DF et non source (d'une autre DF) devient une propriété de l'entité dont la donnée source de la DF est identifiant.

**R4:** Une DF entre deux données sources de DF révèle une association hiérarchique toujours non porteuse de propriété(s).

**R5:** La présence d'une donnée concaténée (association de plusieurs données élémentaires) source de DF révèle une association non hiérarchique porteuse de propriété(s)

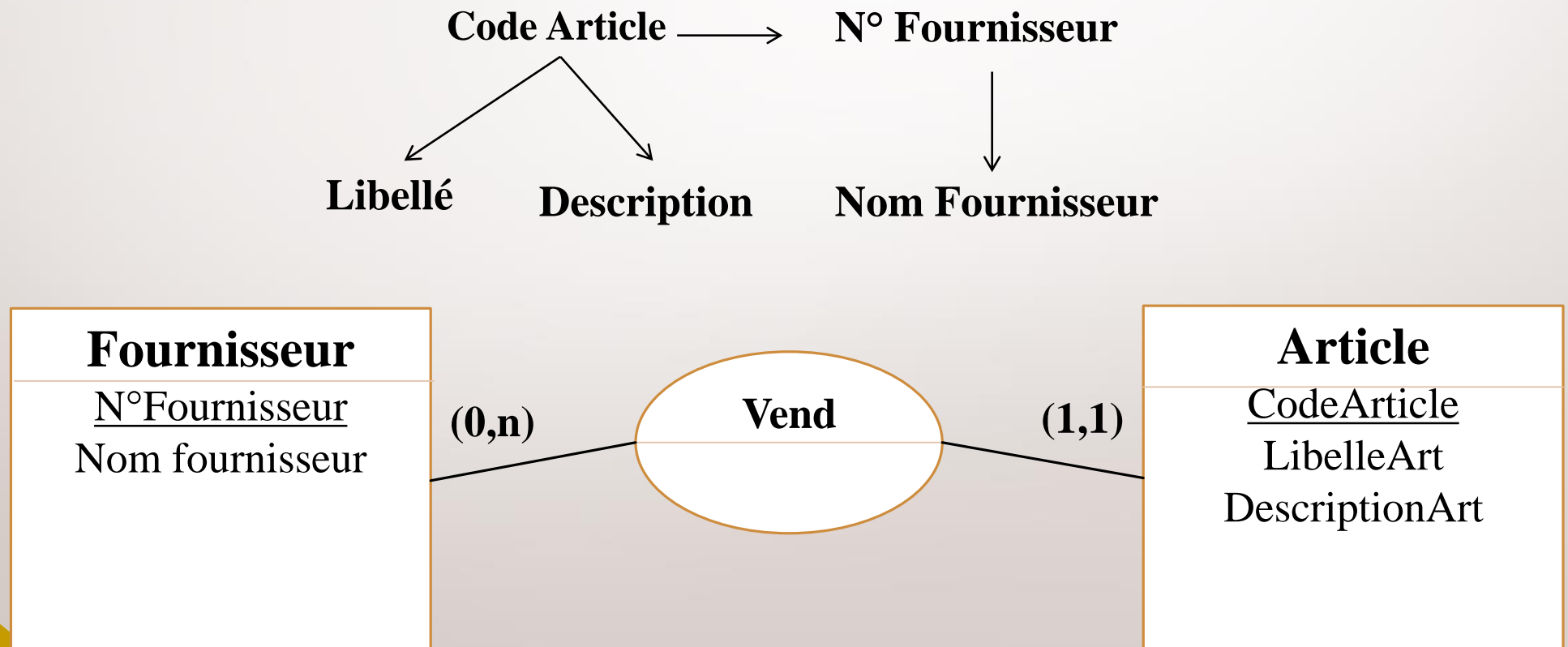
- Une association hiérarchique (1-n) est aussi appelée association fonctionnelle ou **Contrainte d'Intégrité Fonctionnelle (CIF)**.
- Une association non hiérarchique (n-n) est aussi appelée non fonctionnelle ou **Contrainte d'Intégrité Multiple (CIM)**

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### e. Elaboration MCD

**Exemple 1:** Un article identifié par son code, un libellé et une description est acheté chez un seul fournisseur, ce dernier est connu par son numéro et nom

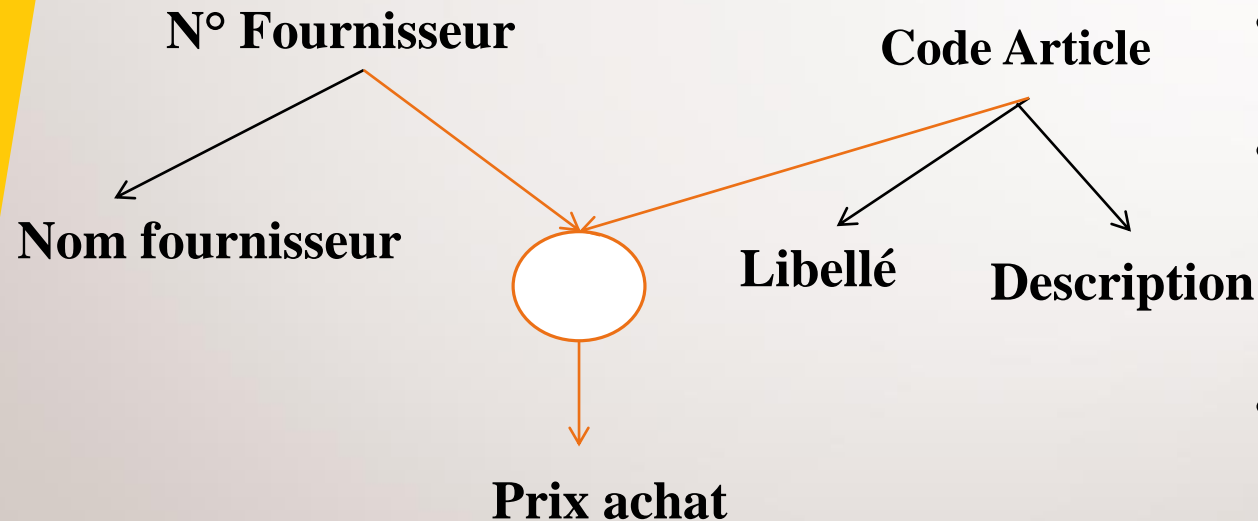


## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### e. Elaboration MCD

**Exemple 2:** Un article identifié par son code, un libellé et une description est acheté chez différents fournisseurs avec des prix différents, un fournisseur peut vendre plusieurs articles.

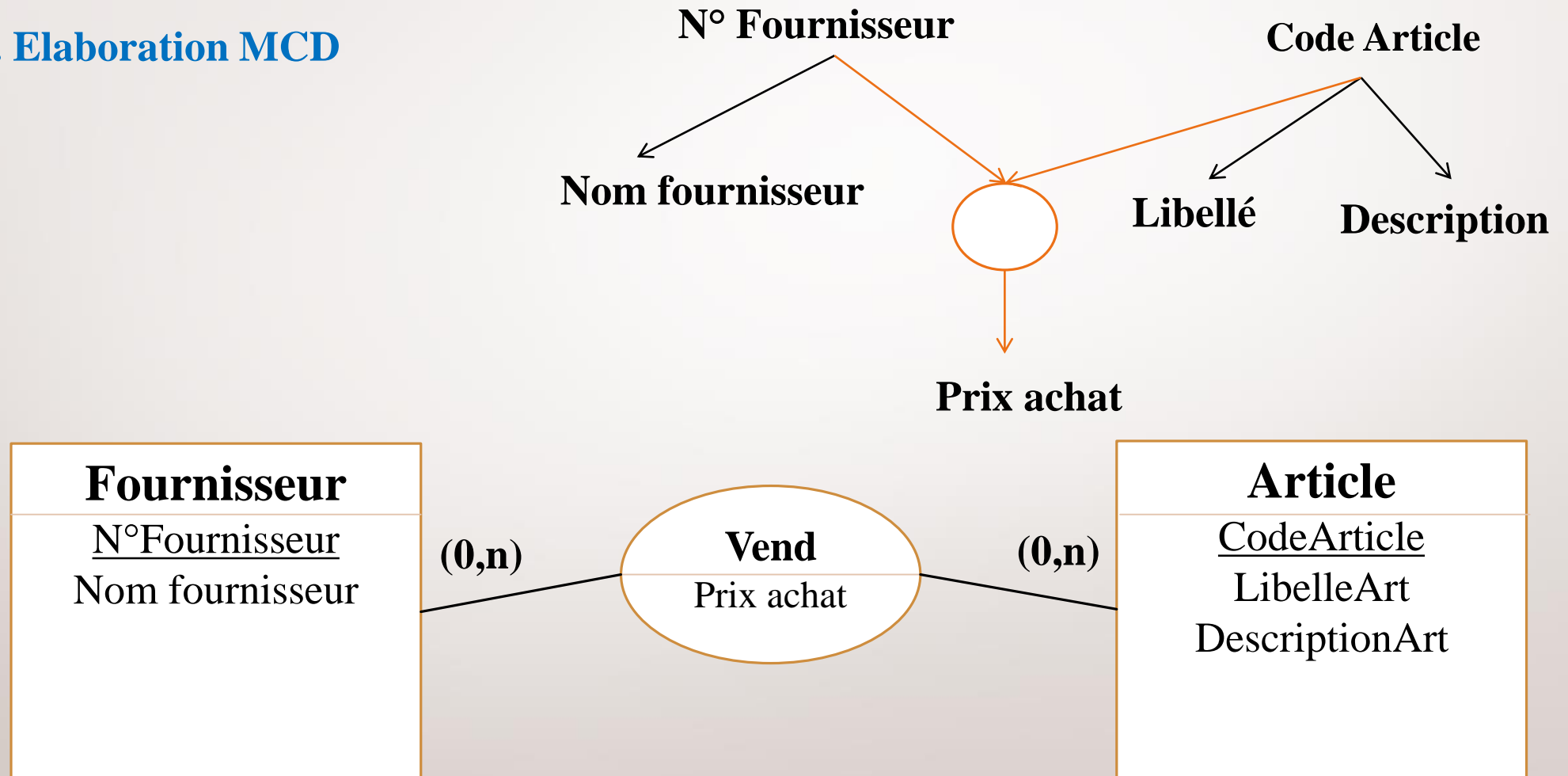


- N° fournisseur et code article deviennent des identifiants d'entités (R1).
- Nom fournisseur, libellé et description deviennent des propriétés de l'entité Fournisseur, libellé et description deviennent des propriétés de l'entité Article (R2).
- Il n'y a pas de DF entre des données sources de DF, donc pas d'association hiérarchiques (R3).
- Prix achat dépend de N° fournisseur et code article, donc il y a une association non hiérarchique porteuse de la propriété Prix achat (R4).

## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### e. Elaboration MCD

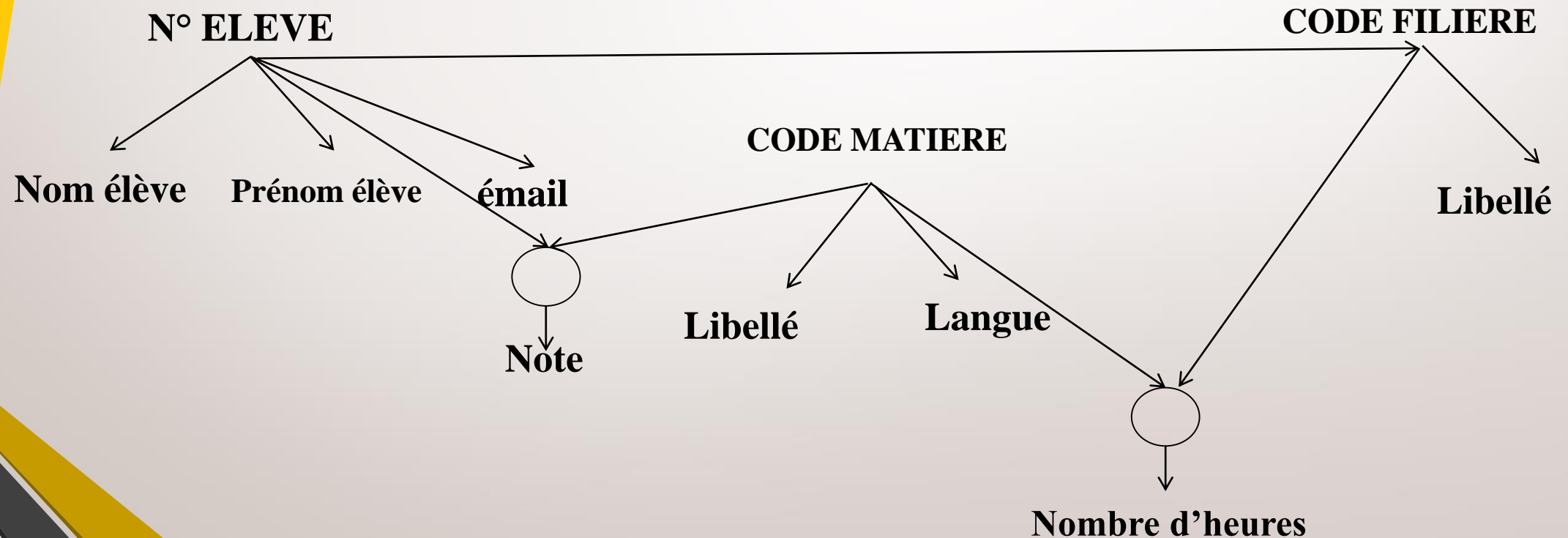


## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### e. Elaboration MCD

**Exercice d'application** : Transformer le GDF suivant en modèle conceptuel de données.

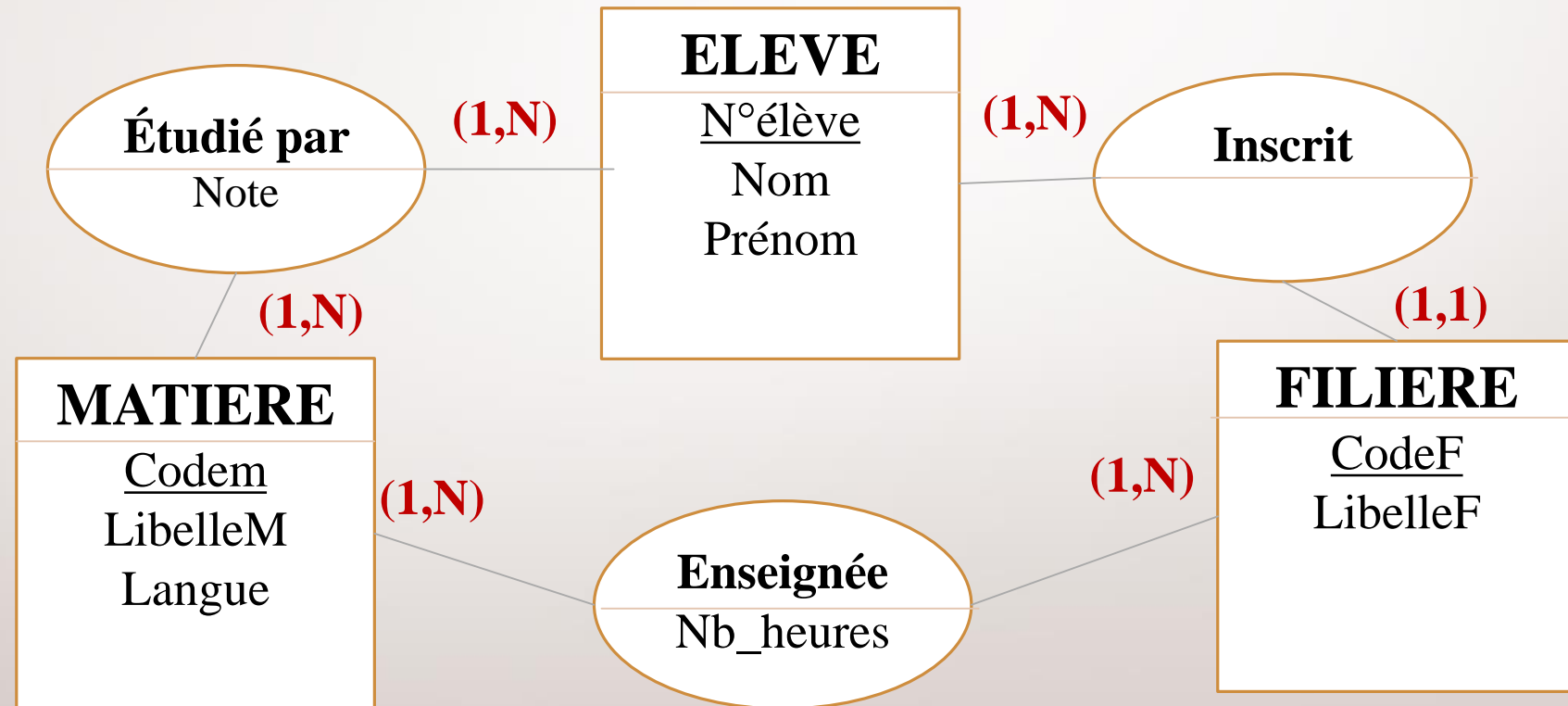


## 2. Modélisation Conceptuelle

### 2.6. Construction du MCD

#### e. Elaboration MCD

Corrigé :



## 2. Modélisation Conceptuelle

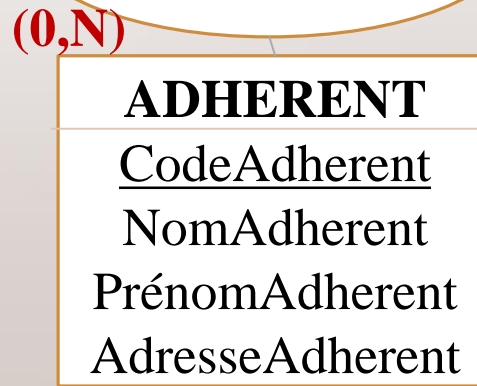
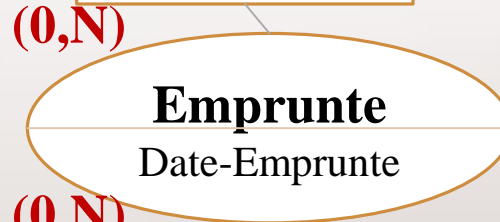
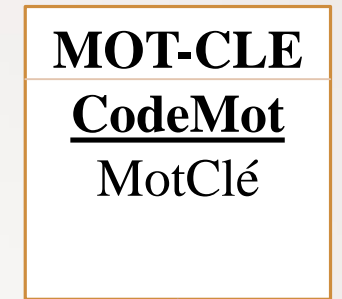
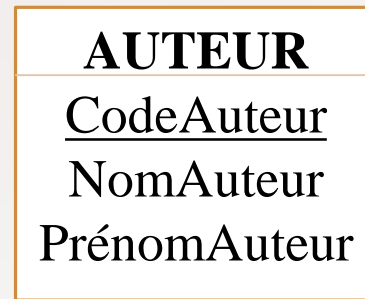
### 2.6. Construction du MCD

#### e. Elaboration MCD

**Exercice d'application 2 :**

(Voir exercice 4 TD1)





(0,N)

(1,N)

(0,N)

(1,1)

(1,N)

(1,N)

(0,N)

(0,N)

## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

L'élaboration d'un M.C.D. se réalise en fait en plusieurs étapes (modèle brut, modèle complet, modèle validé...). Une de ces étapes essentielle est celle qui consiste à vérifier le MCD en appliquant un certain nombre de règles dites de vérification et de normalisation.

**Règle 1 :** Toutes les propriétés doivent être élémentaires, c'est à dire non décomposables.

**Exemple :**

- ↪ Soit une propriété **ADRESSE** constituée du **NUMERO**, **NOM** de la voirie et la ville.
- ↪ Si l'adresse est gérée globalement, cette propriété est élémentaire.
- ↪ Si l'adresse est gérée au niveau du numéro, du nom de la voirie, ou ville il convient d'éclater la propriété.

<b>ADHERENT</b>
NomAdherent
PrénomAdherent
AdresseAdherent



<b>ADHERENT</b>
NomAdherent
PrénomAdherent
NumRue
NomRue
Ville

## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

**Règle 2 :** Chaque objet doit posséder un identifiant et un seul.

**Exemple :**



## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

**Règle 3 :** Les propriétés d'une entité autres que son identifiant doivent être en dépendance fonctionnelle monovaluée de cet identifiant.

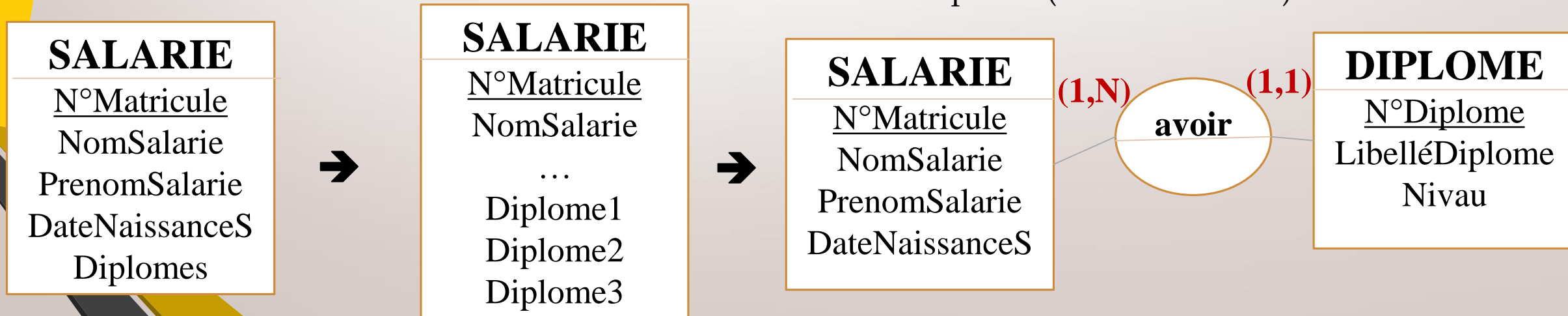
**Exemple :** Soit l'entité ci-dessous « SALARIE »

Cet entité ne peut être défini ainsi que dans le cas bien improbable où chaque salarié de l'entreprise n'aurait au plus qu'un diplôme.

Si le salarié a plusieurs diplômes, l'une des solutions possible est de définir plusieurs propriétés destinées à prendre les différentes valeurs de diplômes détenus par un même salarié «le 1<sup>er</sup> diplôme», « 2<sup>eme</sup> diplôme », etc.

N° matricule → Nom salarié (Détermine)

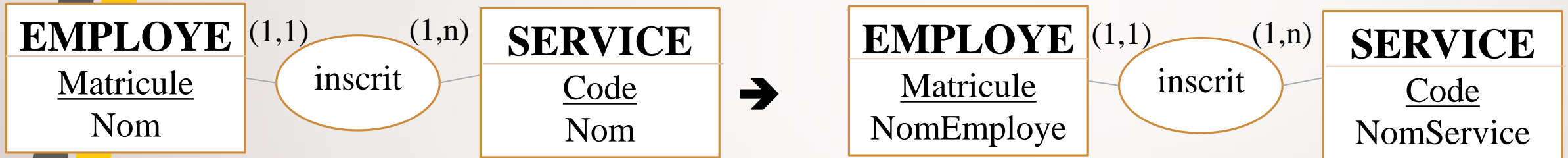
N° matricule → diplôme (Multi détermine)



## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

**Règle 4 :** Une propriété ne peut qualifier qu'une seule entité ou qu'une seule association. Il est nécessaire d'éliminer les cas de redondance et de polysémie.



**Règle 5 :** La dépendance fonctionnelle transitive doit être écartée, pour apporter une plus grande ouverture et une plus grande souplesse au modèle, et pour limiter les redondances.



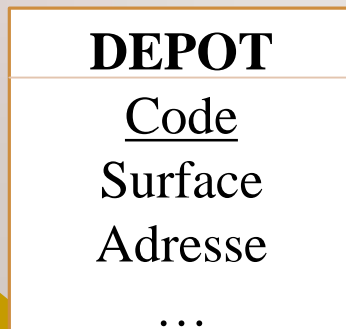
## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

**Règle 6 :** Les propriétés d'une association doivent dépendre de la totalité de l'identifiant de cette association.

Considérons l'association, « STOCKER » entre les entités « FOURNISSEUR », « ARTICLE » et « DEPOT »

Si Certains articles proposés par les fournisseurs n'étant pas forcément présents en stock.

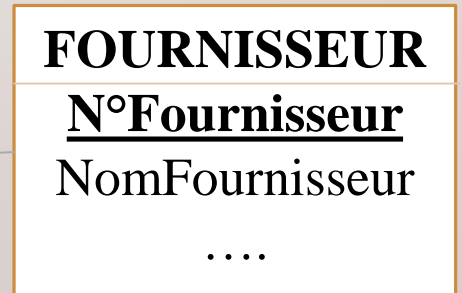


La propriété **quantité** ne dépend pas de la totalité de l'identifiant (certains quantité n'est pas en dépendance fonctionnelle de l'emplacement)

(0,N)

(0,N)

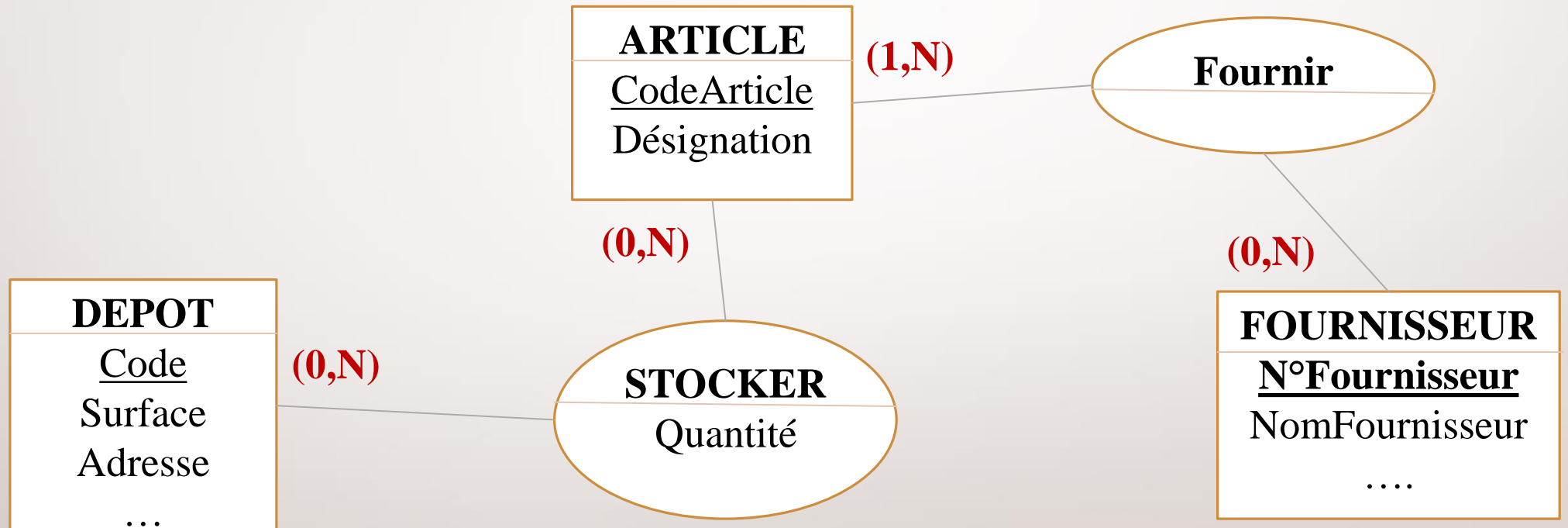
(0,N)



## 2. Modélisation Conceptuelle

### 2.7. Validation MCD

La relation stocker se décompose en fait en deux :



# A retenir :

**Modélisation conceptuelle: Entité, association, propriété, identifiant, cardinalités ? Dictionnaire de données ? Dépendance fonctionnelle ? GDF? MCD ? Validation MCD?**