

# **Modèle Logique de Données**

## **MLD**

### **(Model Relationnel)**

# Table of Contents

1. Introduction
2. Règles de transformation du MCD au MLD
  - a) Transformation des entités
  - b) Transformation des relations
3. Niveau Logique : Le modèle relationnel

## 1) Introduction

- Le schéma conceptuel est le point de départ et la partie fondamentale de la conception d'un SI en une base de données.
- Cela nous amène à définir ce qu'est une base de données.
- C'est un ensemble de fichiers (tables) reliés entre eux par un champ (attribut ou propriété).
- Le passage du modèle conceptuel de données au modèle logique de données obéit à des règles selon qu'on est en modèle hiérarchique, réseau ou relationnel ou orienté objet.
- L'organisation logique (composition des fichiers et les liens entre eux) est indépendante de l'organisation physique (disposition des fichiers sur support). Cette indépendance permet de changer les unités de stockage sans modifier les programmes.

## **2) Règles de transformation du MCD au MLD**

### **a) Transformation des entités**

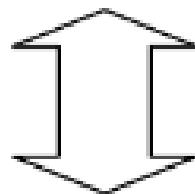
- Toute entité est transformée en une relation (table).
- Les propriétés de l'entité deviennent les attributs de la table (Les colonnes ou les champs).
- L'identifiant de l'entité devient la clé primaire de la table.

## 2) Règles de transformation du MCD au MLD

### a) Transformation des entités

Exemple:

Clients
# Num client
Société
Contact
Fonction
Adresse
Ville

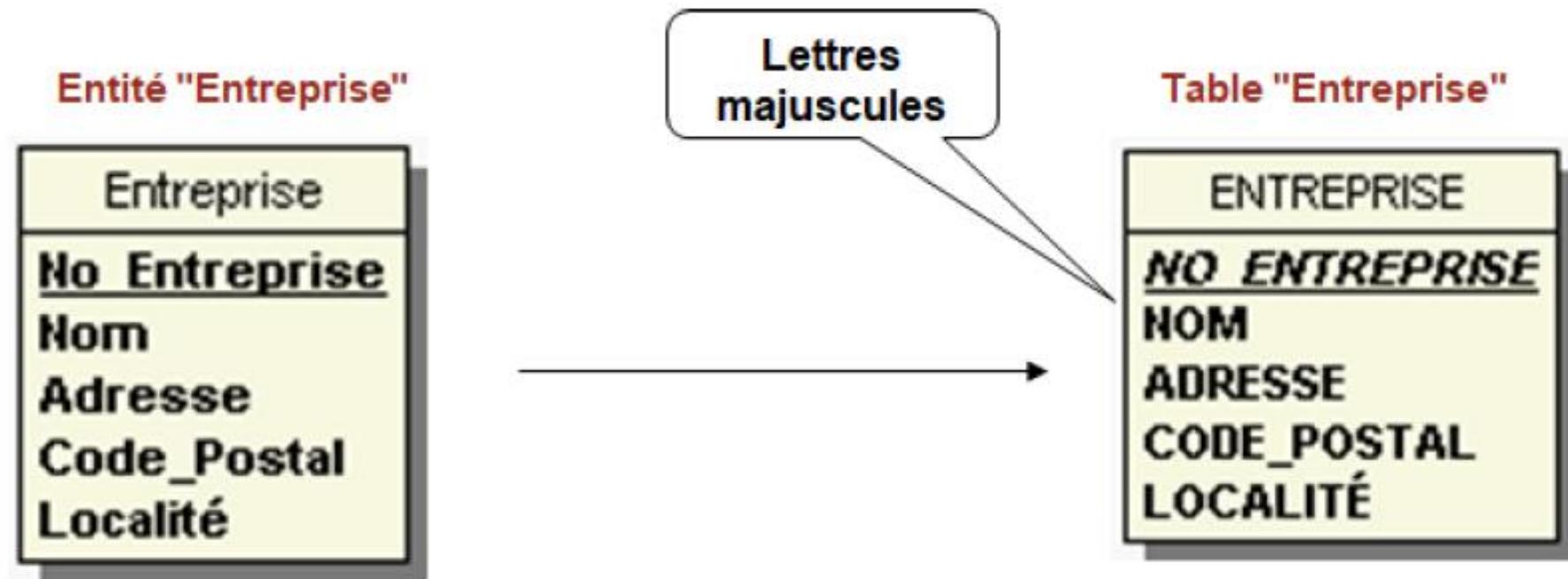


Client (Num client, Société, Contact, Fonction, Adresse, Ville)

## 2) Règles de transformation du MCD au MLD

### a) Transformation des entités

Exemple 1:



**Modèle logique de donnée (MLD)**

Entreprise (NO Entreprise, Nom, Adresse, code\_postale, Localité)

## 2) Règles de transformation du MCD au MLD

### b) Transformation des relations

Le passage du modèle conceptuel au modèle logique au niveau des classes de relation (association) se fait selon les cardinalités des classes d'entité participant à la relation :

#### ❖ Les types d'associations

En fonction des cardinalités :

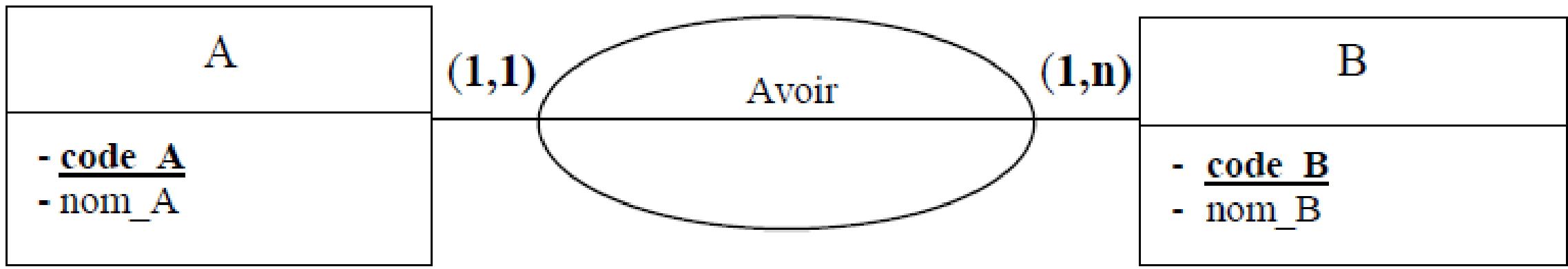
**1:1** si toutes la cardinalités maximales valent **1**  $\Leftrightarrow$  **(0,1/0,1)**

**1:N** s'il existe au moins une cardinalité maximale à **n** et une à **1**  $\Leftrightarrow$   
**(1,1/1,N)** ou **(1,1 / 0,N)**

**N:N** si toutes la cardinalités maximales valent **N**  $\Leftrightarrow$   
**(1, N / 1, N)** ou **(1, N / 0, N)** ou **(0, N / 0, N)**

## b) Transformation des relations

### 1) Cas de relation 1:N (1,1/1,N) ou (1,1 / 0,N)

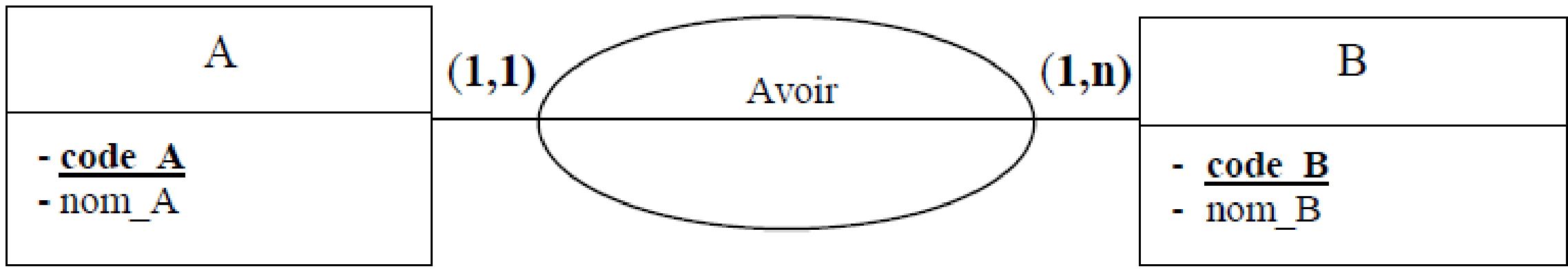


- l'entité B représente le père
- l'entité A représente le fils et
- La relation hiérarchique ou (père-fils)

Dans ce cas: Le **code\_A** détermine d'une manière unique le **code\_B**,  
(dépendance fonctionnelle).

## b) Transformation des relations

### 1) Cas de relation 1:N (1,1 / 1,N) ou (1,1 / 0,N)

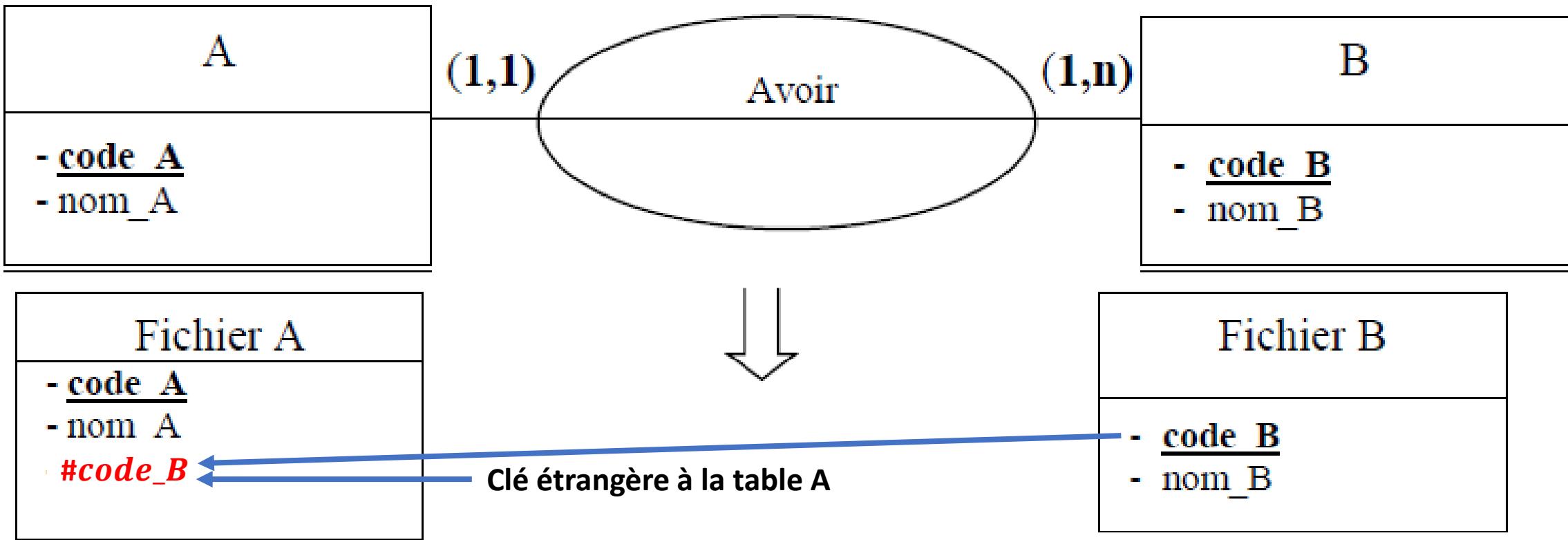


L'opération d'une dépendance fonctionnelle (df) entre entité :

- détruit la relation,
  - copie l'identifiant de l'entité B (père) dans l'entité A (fils)
- Cet identifiant devient **clé étrangère** dans l'entité A.

## b) Transformation des relations

### 1) Cas de relation 1:N (1,1 / 1,N) ou (1,1 / 0,N)



On obtient donc 2 tables représentées ci-dessous :

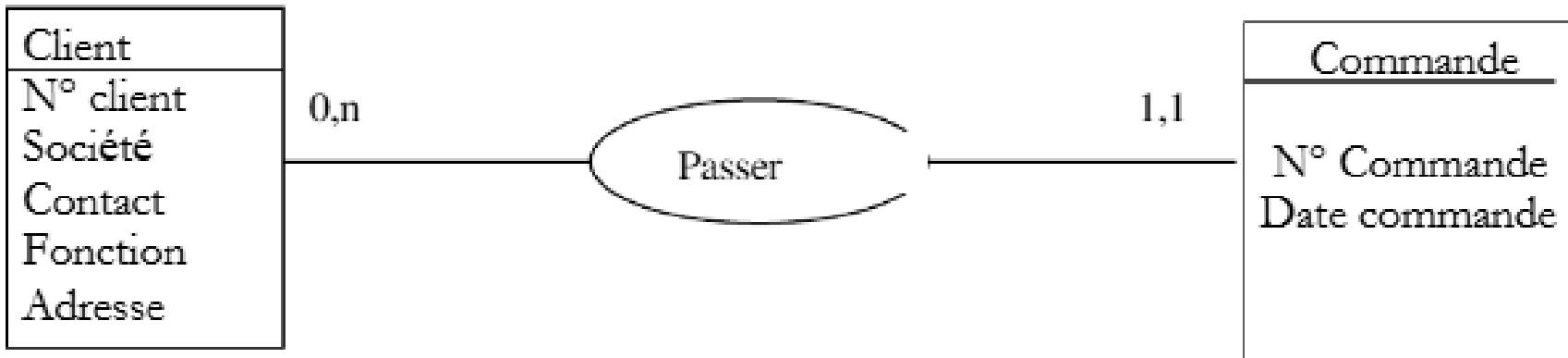
**A** (code\_A, nom\_A, #code\_B)

**B** (code\_B, nom\_B)

## 1) Cas de relation 1:N (1,1 / 1,N) ou (1,1 / 0,N)

### Exemple 2 : cardinalité (1,1) - (0,n)

- Modèle conceptuel de donnée (MCD)

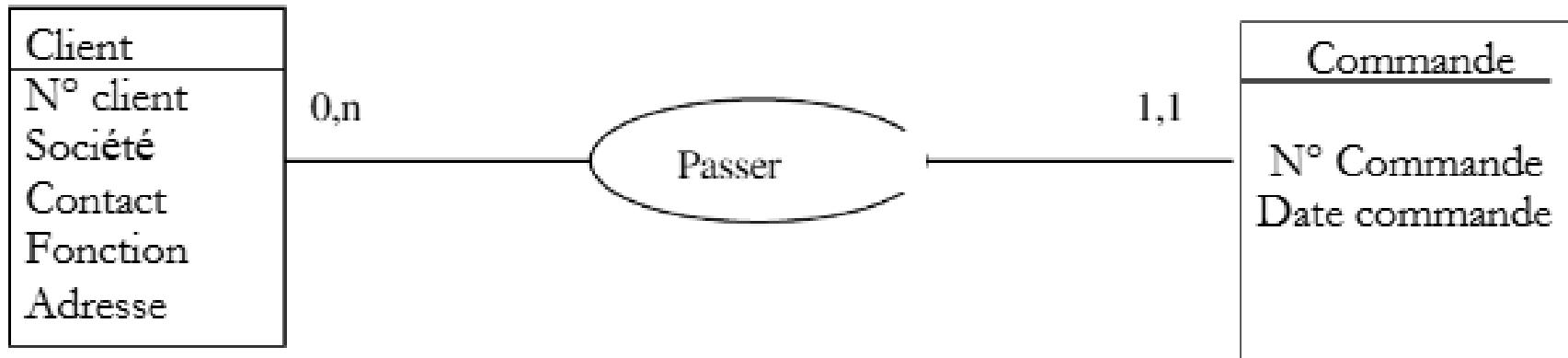


- Un client peut passer 0 ou n commandes et
- une commande ne peut être passée que par un et un seul client

## 1) Cas de relation 1:N (1,1 / 1,N) ou (1,1 / 0,N)

### Exemple 2 : cardinalité (1,1) - (0,n)

- **Modèle conceptuel de donnée (MCD)**



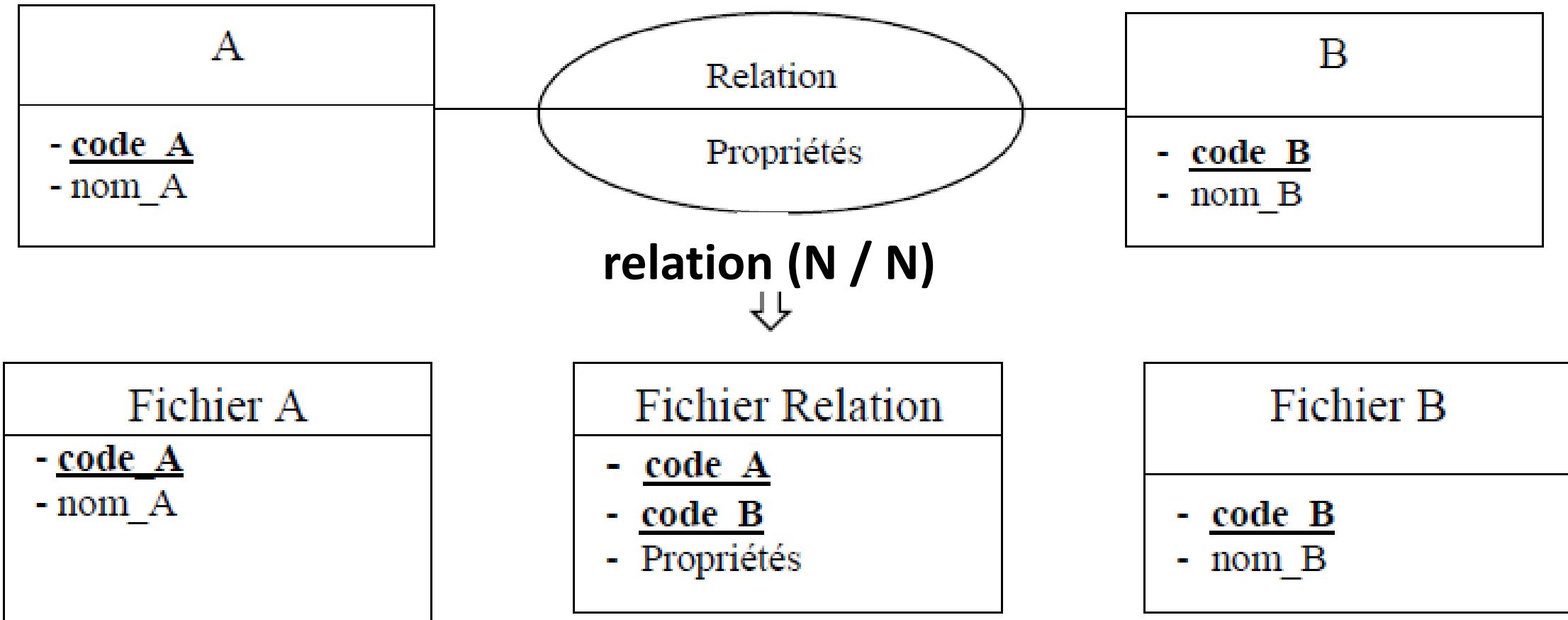
- **Modèle logique de donnée (MLD)**

Client (Num client, Société, Contact, Fonction, Adresse, Ville)

Commande (N° Commande, Date commande, **#N°client**)

- La **Clé Primaire** de la table à la **cardinalité (0,n)** devient une **Clé Etrangère** dans la table à la **cardinalité (1,1)**

## b) Cas de relation (N / N) ; (1,N/1,N), (1,N/0,N), (0,N/0,N)



On obtient donc 3 tables représentées ci-dessous :

*A (code\_A, nom\_A)*

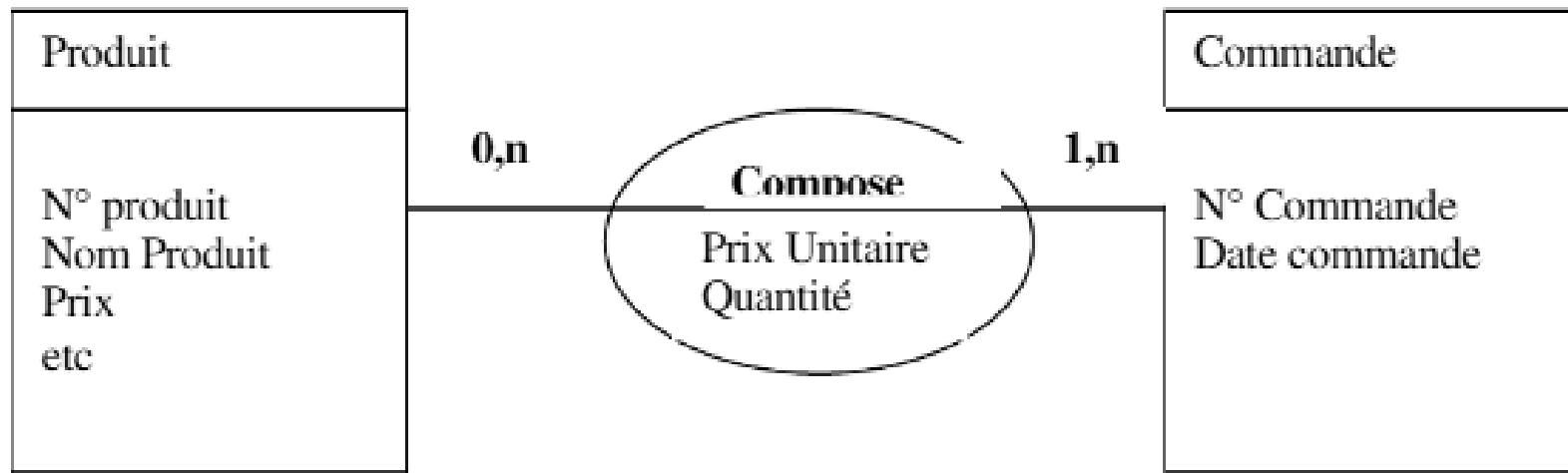
*B (code\_B, nom\_B)*

*Relation (code\_A, code\_B, propriétés)*

## b) Cas de relation (N / N); (1,N/1,N), (1,N/0,N), (0,N/0,N)

Exemple 3: cardinalité (0,n) - (1,n)

Modèle conceptuel de donnée (MCD)

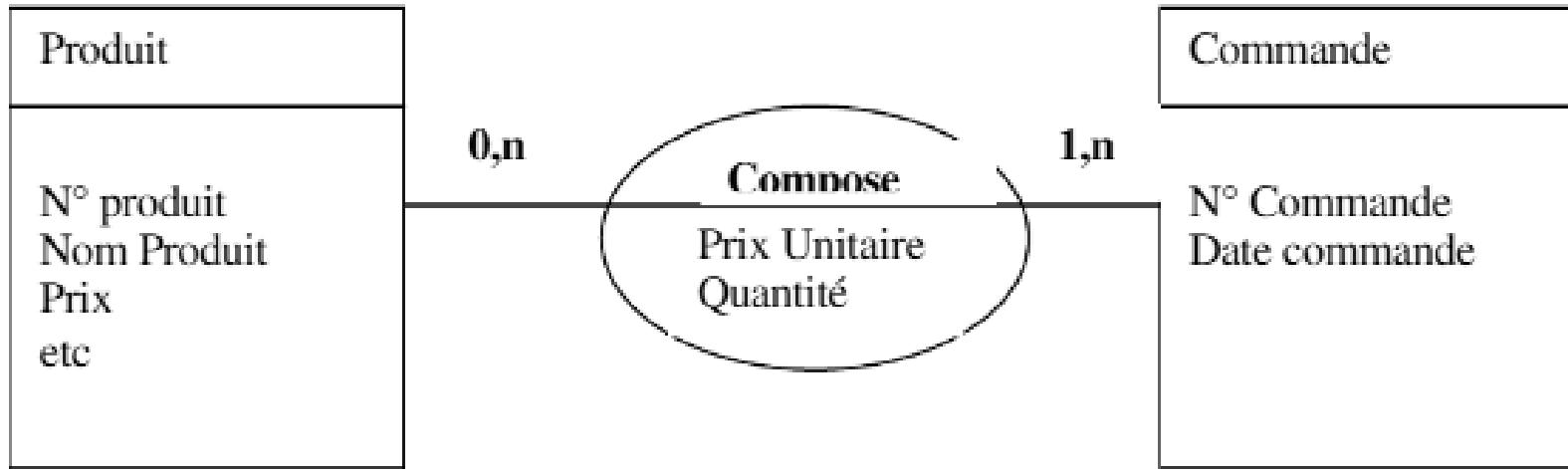


- Une commande est composée de 1 ou n produits distincts en certaine quantité.
- Un produit est présent dans 0 ou n commandes en certaine quantité.

## b) Cas de relation (N / N) ; (1,N/1,N), (1,N/0,N), (0,N/0,N)

### Exemple 3: cardinalité (0,n) - (1,n)

- Modèle conceptuel de donnée (MCD)



- Modèle logique de donnée (MLD)
  - COMMANDE (N° Commande, Date commande)**
  - PRODUIT (N° Produit, libelle)**
  - COMPOSE (N° Commande, N° Produit, quantité)**

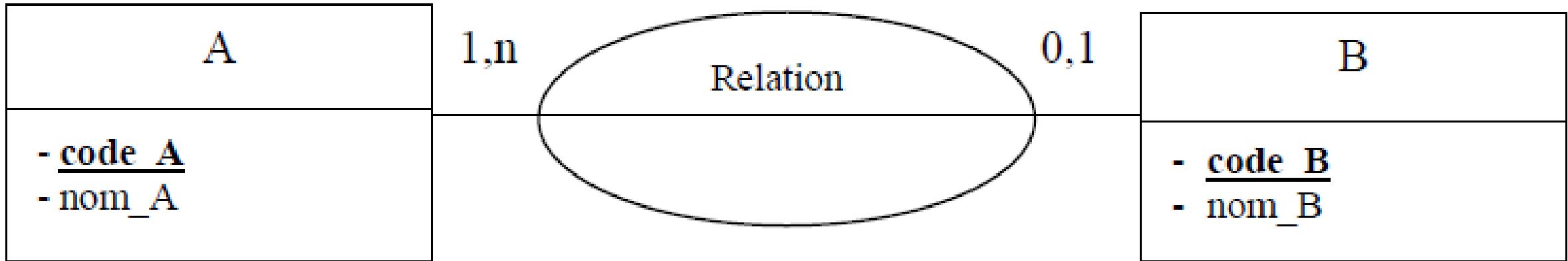
### c) Cas de relations (0,1/1,N), (0,1/0,N), (0,1/0,1)

On a deux solutions possibles : soit

- 1) On peut se ramener au **cas 1 : Cas de relation 1:N** , en considérant qu'il s'agit d'une dépendance fonctionnelle (df) particulière
  - 2) ou bien considérer que c'est un cas dérivé du cas 2: **Cas de relation (N / N)**
- La solution est laissée au concepteur.

### c) Cas de relations (0,1/1,N), (0,1/0,N), (0,1/0,1)

Exemple :



Solution 1:

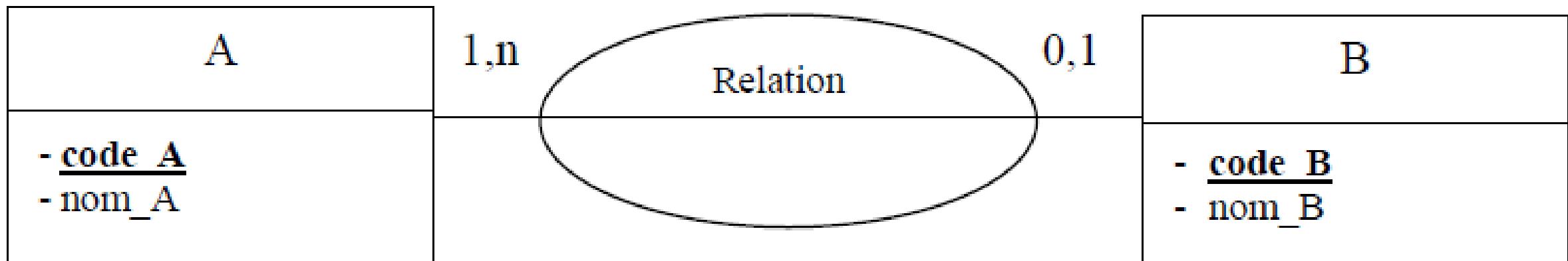
D'après le cas 1: on obtient :

$A (\underline{code}_A, nom_A, \#code_B),$

$B (\underline{code\_B}, nom\_B)$

### c) Cas de relations (0,1/1,N), (0,1/0,N), (0,1/0,1)

Exemple :



Solution 2:

D'après le cas 2: on obtient :

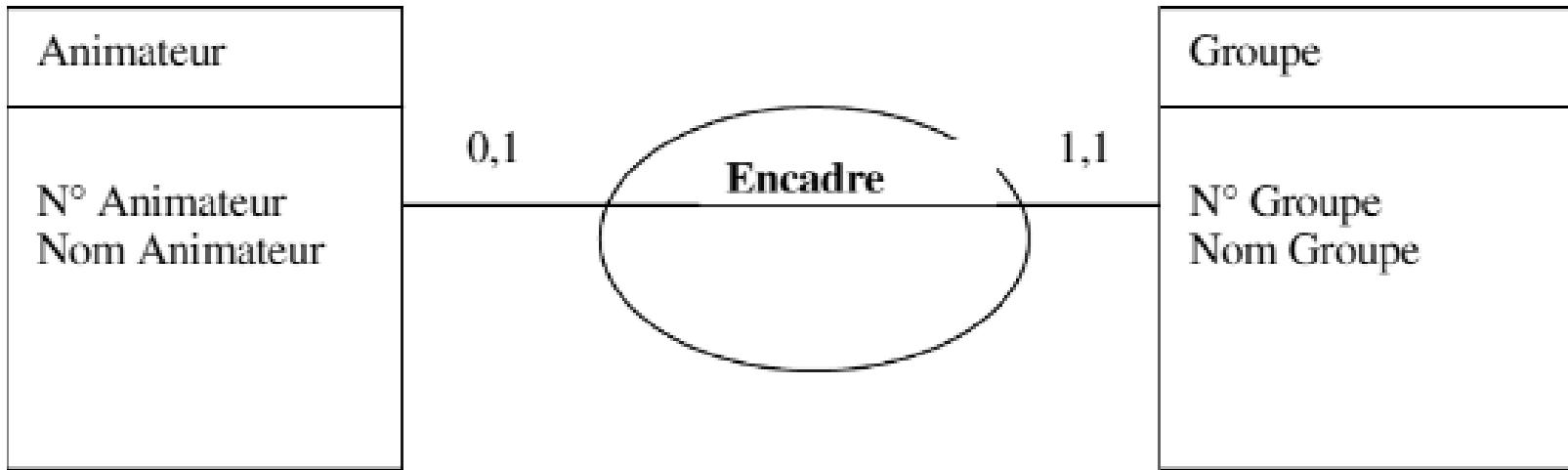
*A (code<sub>A</sub>, nom<sub>A</sub>),*

*B (code<sub>B</sub>, nom<sub>B</sub>),*

*Relation (code<sub>A</sub> , code<sub>B</sub>)*

## Exemple 4 cardinalité (0,1) - (1,1)

### Modèle Conceptuel de Donnée (MCD):



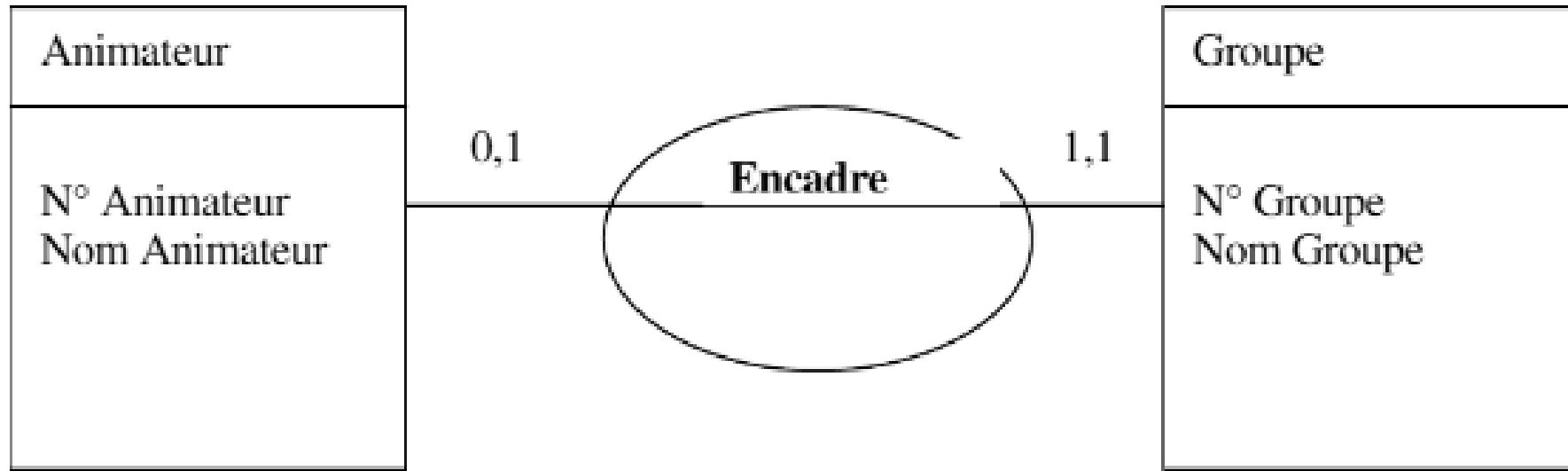
- Dans ce centre de vacances, Chaque animateur encadre 0 ou 1 groupe, chaque groupe étant encadré par un et un seul animateur.

### Modèle Logique de Donnée Relationnel (MLD):

- La Clé Primaire de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1):

## Exemple 4 cardinalité (0,1) - (1,1)

Modèle Conceptuel de Donnée (MCD):



Modèle Logique de Donnée Relationnel (MLD):

**ANIMATEUR (id\_Animateur, Nom\_Animateur)**

**GROUPE (id\_Groupe, Nom\_Groupe, #id\_animateur)**

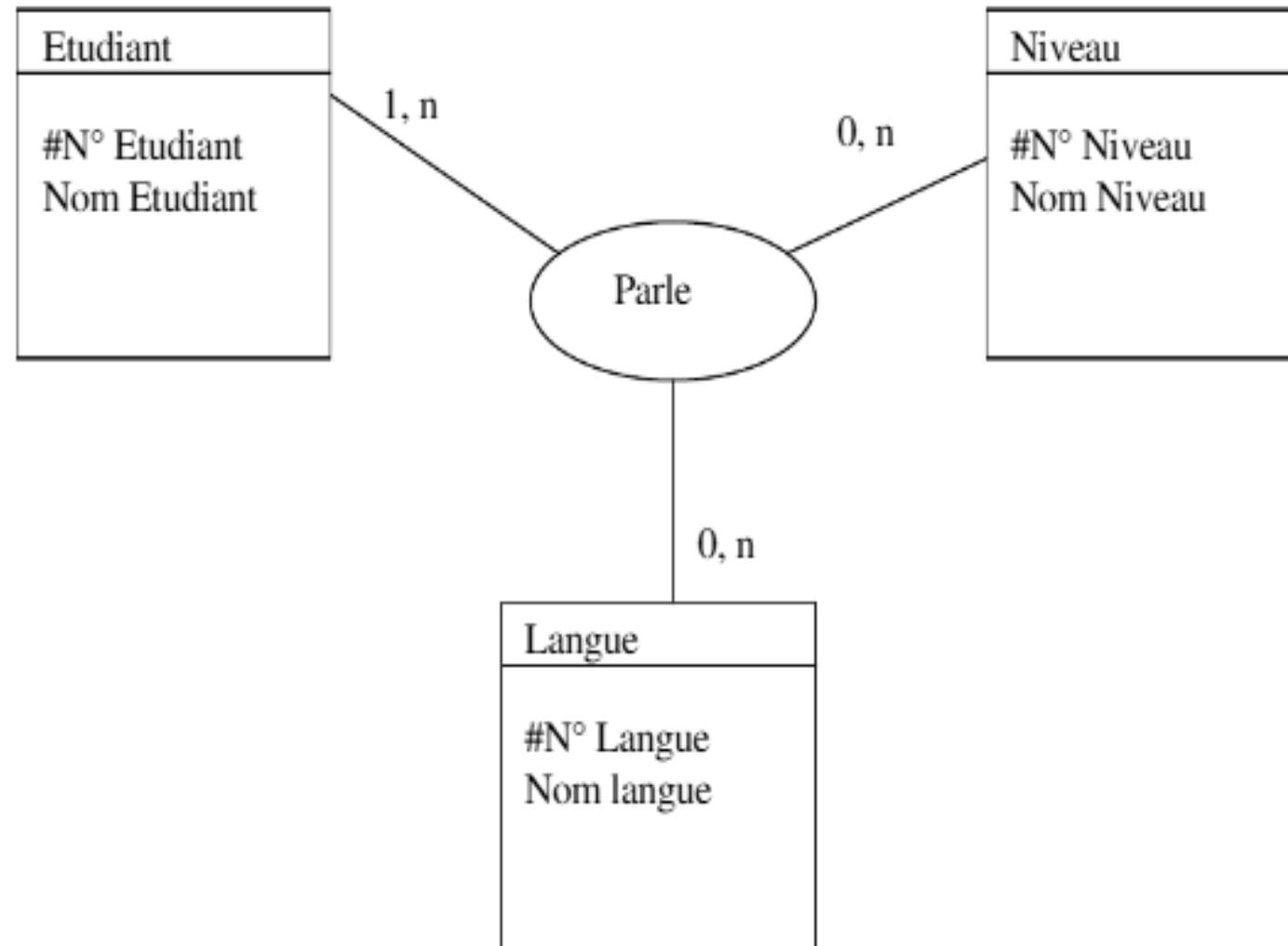
#### d) Relation n-aire (quelles que soient les cardinalités).

- Il y a création d'une table supplémentaire ayant comme Clé Primaire la concaténation des identifiants des entités participant à la relation.
- Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

## d) Relation n-aire (quelles que soient les cardinalités).

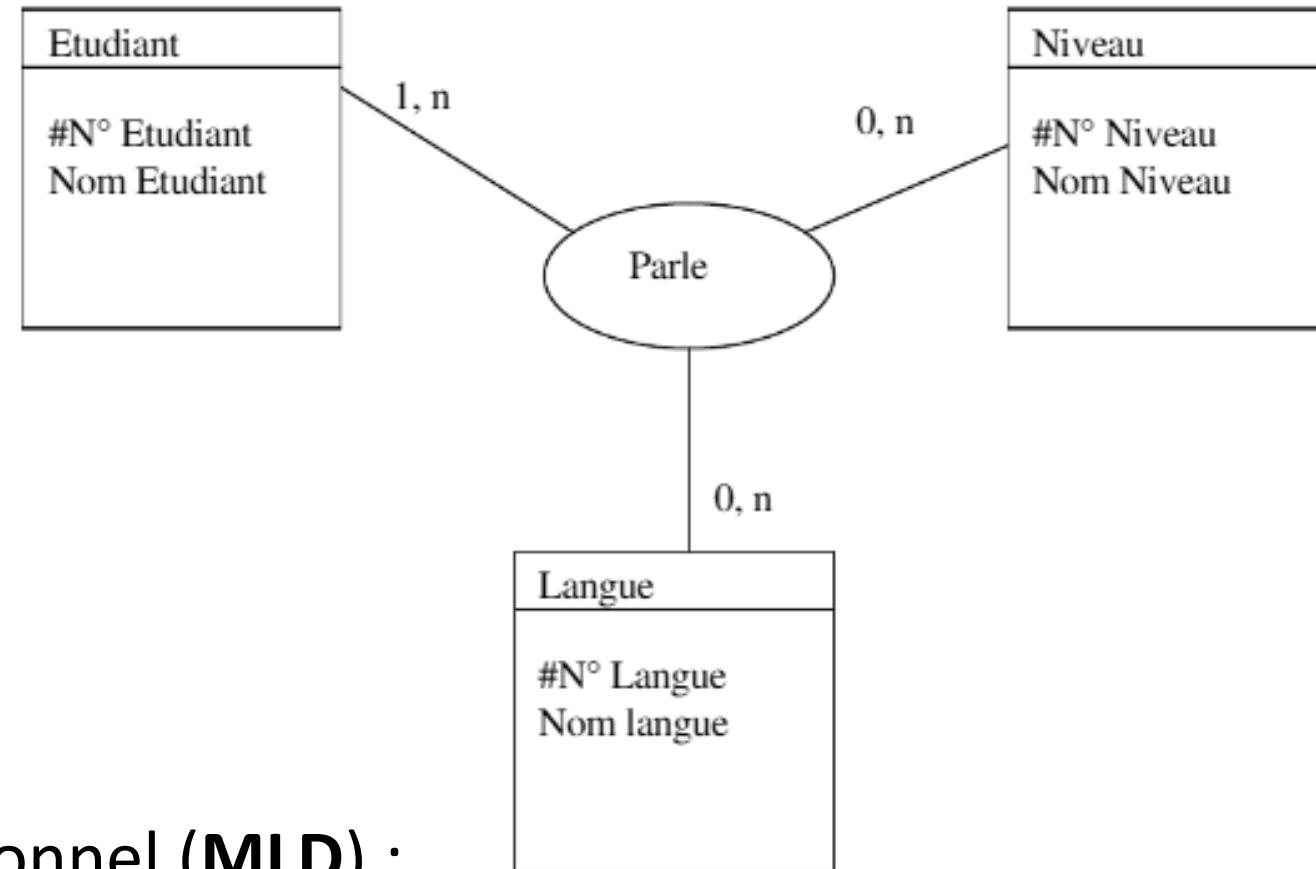
### Exemple 5:

- Un étudiant parle une ou plusieurs langues avec un niveau.
- Chaque langue est donc parlée par 0 ou n étudiants avec un niveau.
- Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.



## d) Relation n-aire (quelles que soient les cardinalités)

### Exemple 5:



Modèle conceptuel de donnée relationnel (**MLD**) :

**ETUDIANT (N° Etudiant, Nom\_Etudiant)**

**NIVEAU (N° Niveau, Nom\_Niveau)**

**LANGUE (N° Langue, Nom\_Langue)**

**PARLE (N° Etudiant, N° Niveau, N° Langue)**

## e) Association Réflexive

Nous appliquons les règles générales avec la seule différence que la relation est deux fois reliée à la même entité :

### 1) cardinalité (X,1) - (X,n), avec X=0 ou X=1.

La **Clé Primaire** de l'entité se dédouble et devient **une Clé Etrangère** dans la table.

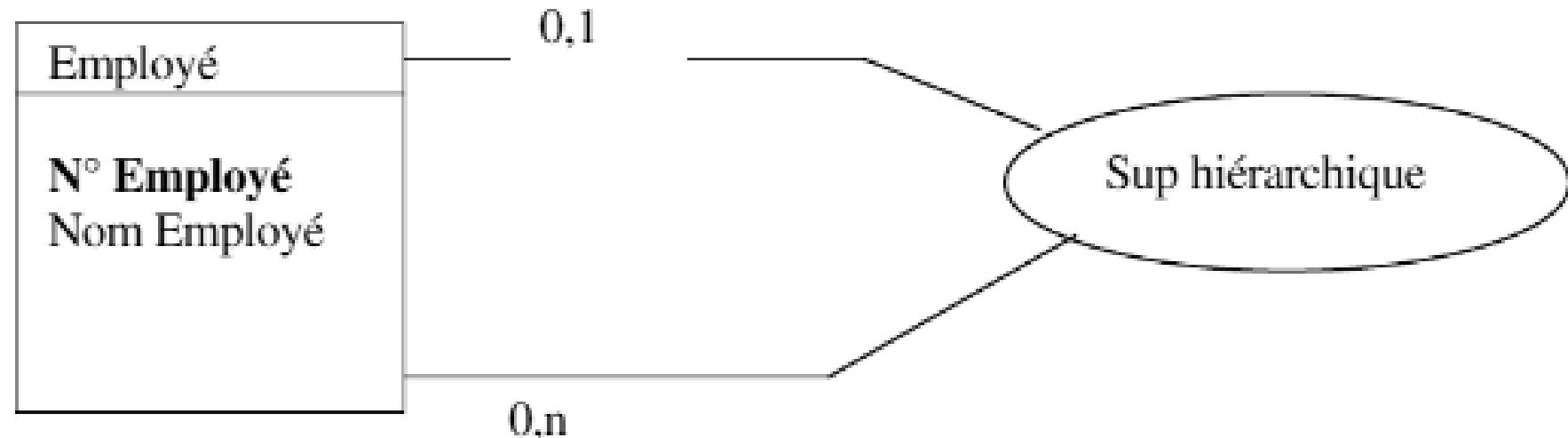
## e) Association Réflexive

### Exemple cardinalité (0,1) - (0,n)

Prenons l'exemple d'une société organisée de manière pyramidale :

- chaque employé a 0 ou 1 supérieur hiérarchique direct.
- Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.

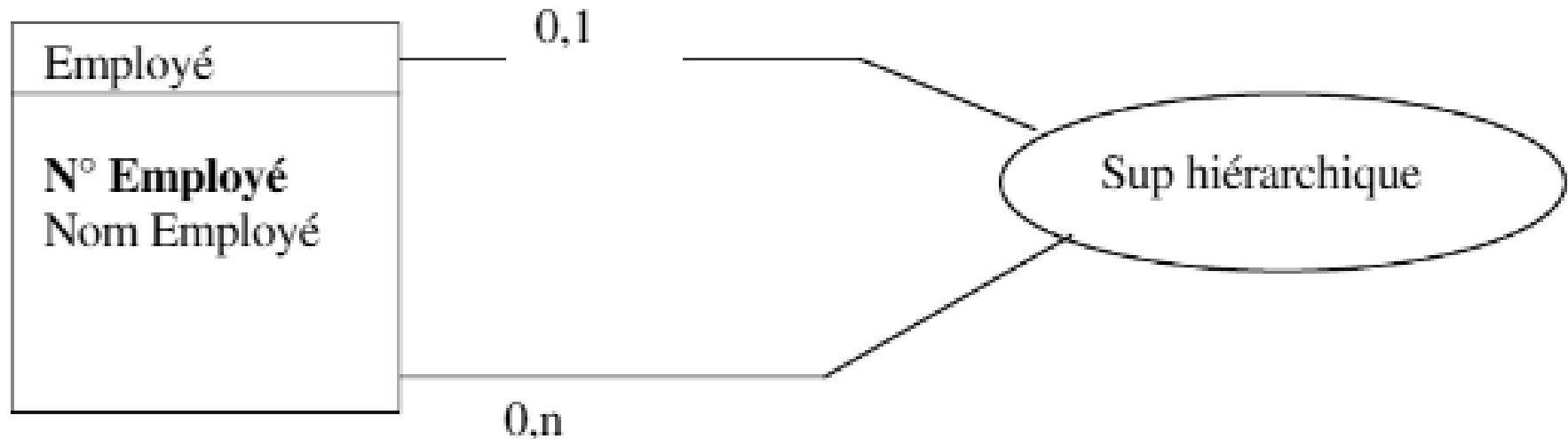
Modèle conceptuel de donnée (MCD):



## e) Association Réflexive

Exemple cardinalité (0,1) - (0,n)

Modèle conceptuel de donnée (**MCD**):



Modèle Logique de donnée relationnel (**MLD**):

**EMPLOYEE (id\_Employe, Nom\_Employe, #id\_Sup\_Hierarchique)**

**#id\_Sup\_Hierarchique** est l'identifiant (id\_Employe) du supérieur hiérarchique direct de l'employé considéré.

## e) Association Réflexive

### 2) cardinalité $(X,n)$ - $(X,n)$ , avec $X=0$ ou $X=1$

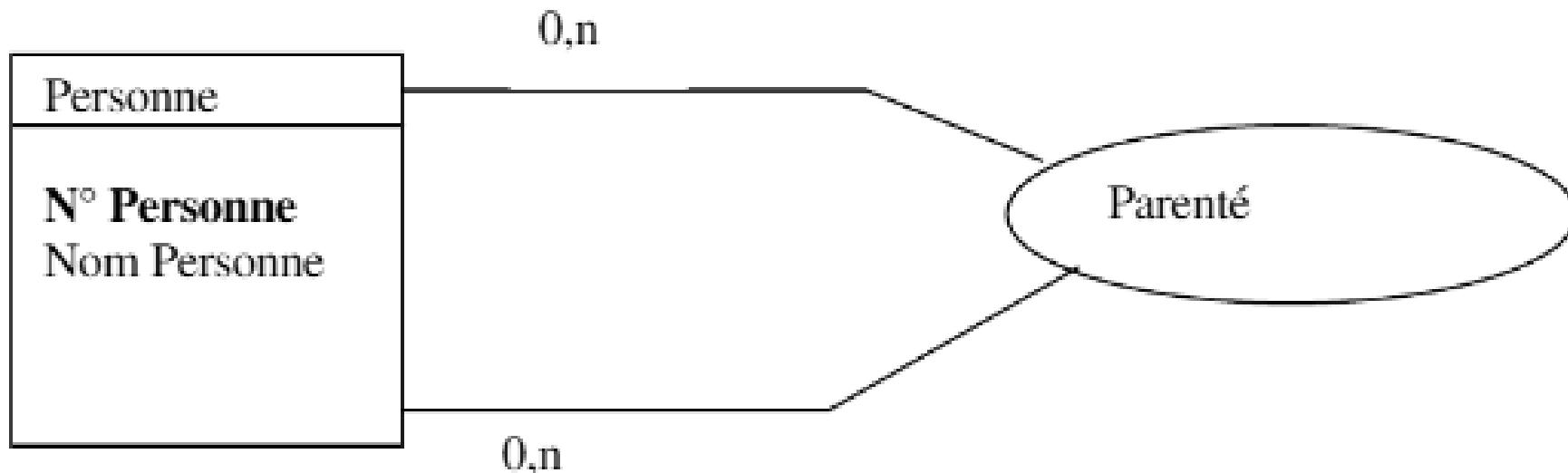
De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire  $(X,n)$  -  $(X,n)$  (règle 3). Il y a donc création d'une nouvelle table.

## e) Association Réflexive

### Exemple cardinalité (0,n) - (0,n)

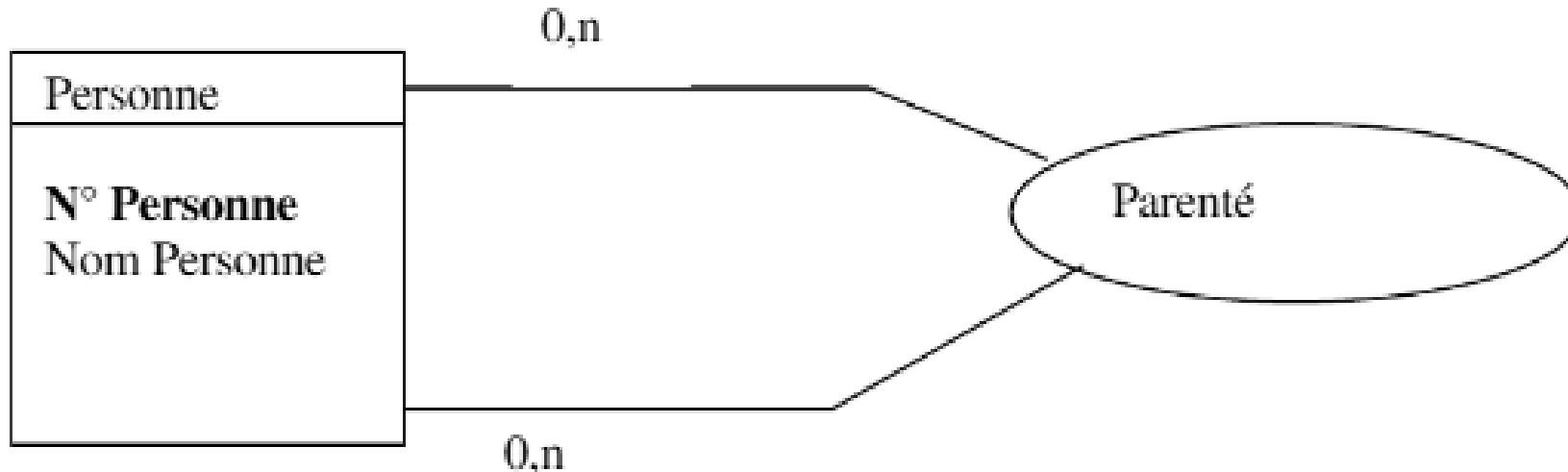
Prenons cette fois l'exemple d'une organisation de type familiale :

- chaque personne a 0 ou n descendants directs (enfants), et a aussi 0 ou n descendants indirects (enfants).



## Exemple cardinalité (0,n) - (0,n)

### Modèle conceptuel de donnée (MCD):



### Modèle Logique de donnée relationnel (MLD):

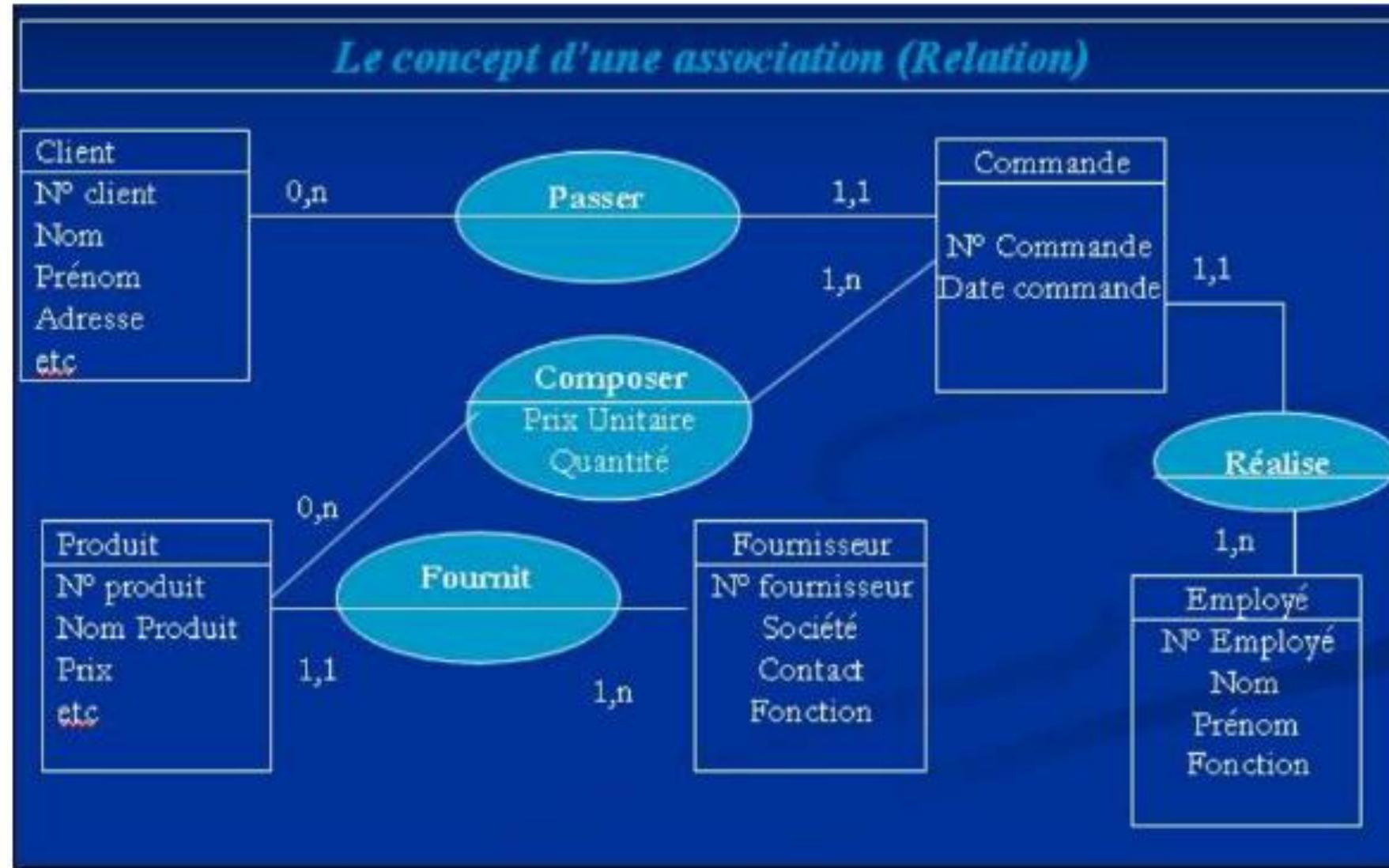
PERSONNE (N° Personne, Nom\_Personne)

PARENTE (N° Parent, #id\_Enfant)

**#N°\_Parent** est l'identifiant (N° Personne) d'un ascendant direct de la personne.  
**#N°\_Enfant** est l'identifiant (N° Personne) d'un descendant direct de la personne.  
La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présent dans cette famille

## Exemple récapitulatif

Considérant le cas de la gestion des ventes dans un magasin. Le MCD correspondant est :



## Exemple récapitulatif

Le modèle logique de donnée Relationnel (MLD) devient donc

**Client (N° client, Nom, Prénom, Adresse)**

**Commande (N° Commande, Date commande, # N° client, # N° Employé)**

**Employé (N° Employé, Nom, Prénom, Fonction)**

**Fournisseur (N° fournisseur, société, contact, Fonction)**

**Composer(N° Commande, N° produit, Prix Unitaire, Quantité)**

**Produit (N° produit, Nom Produit, Prix, # N° fournisseur)**

## Modèle relationnel

Le modèle relationnel a été défini par E.F. Codd en 1970 à IBM San José

Aspects fondamentaux du modèle relationnel:

- une algèbre permettant de manipuler des tables et des relations
- une démarche de conception permettant de définir une collection de relations
- Les concepts du modèle relationnel découlent de la théorie des ensembles

# Modèle relationnel

A ce type de modèle sont associées les notions suivantes:

- domaine
- table relationnelle
- attribut
- tuple (ou n-uplet)

# Domaine

Ensemble de valeurs caractérisées par un nom.

## Exemple:

- Le domaine des noms, le domaine des âges,...
- Le domaine des entiers  $E = \{\dots - 2, -1, 0, +1, +2, \dots\}$
- Le domaine des jours de la semaine ={samedi, dimanche, lundi, mardi, mercredi, jeudi, vendredi},

## **Modèle relationnel:**

Ce modèle s'inspire du concept mathématique de relation.

### **Produit Cartésien**

Soient  $A_1, A_2, A_3, \dots, A_n$ ;  $n$  ensembles quelconques.

On appelle produit cartésien de  $A_1, A_2, A_3, \dots, A_n$

l'ensemble  $A_1 \times A_2 \times A_3 \times \dots \times A_n$  défini par :

$$A_1 \times A_2 \times A_3 \times \dots \times A_n = \{(x_1, x_2, x_3, \dots, x_n) | x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n\}$$

## Produit Cartésien

C'est à dire c'est la totalité des combinaisons possibles que l'on peut tirer de ces domaines.

### Exemple

Le produit cartésien des domaines: Pays={Algérie, France, Espagne} et Capitale={Alger, Paris, Madrid}:

Algérie	Alger
Algérie	Paris
Algérie	Madrid
France	Alger
France	Paris
France	Madrid
Espagne	Alger
Espagne	Paris
Espagne	Madrid

## Notion de relation

sous-ensemble du produit cartésien d'une liste de domaines

Une relation est généralement caractérisée par un nom.

Exemple: A partir des domaines :

$D_1 = \{Algérie, France, Espagne\}$  et  $D_2 = \{Alger, Paris, Madrid\}$

On peut composer une relation qui nous intéresse, c'est à dire celui qui a un sens par rapport à la réalité :

Pays capital	D1	D2
Algérie	Algérie	Alger
France	France	Paris
Espagne	Espagne	Madrid

## Table relationnelle

Une relation peut être vue comme un tableau à deux dimensions dont les colonnes correspondent aux domaines et les lignes contiennent les tuples.

**Notion de tuple (n-uplet)** : est une ligne d'une relation

Par exemple: (Algérie, Alger)

**Notion d'attribut:** C'est une colonne d'une relation caractérisée par un nom

The diagram shows a relational table with three columns: **Pays\_capital**, **D1**, and **D2**. The first row contains the column headers. The second, third, and fourth rows contain data tuples: (Algérie, Alger), (France, Paris), and (Espagne, Madrid) respectively. An orange arrow points from the text "le nom de la relation" to the header cell **Pays\_capital**. A blue arrow points from the text "les attributs" to the header cells **D1** and **D2**. A green arrow points from the text "tuple, n-uplet ou n-uple" to the data cells in the second row.

<b>Pays_capital</b>	<b>D1</b>	<b>D2</b>
Algérie	Alger	
France	Paris	
Espagne	Madrid	

## Exemple

Considérons les domaines:

*Couleur* = {*Rouge*, *Vert*, *Bleu*} et

*Constructeur* = {*Fiat*, *Renault*, *Volvo*, *Opel*}.

Considérons la relation **Voiture** définie par:

*Voiture* = {(*Rouge*, *Fiat*), (*Rouge*, *Renault*), (*Vert*, *Opel*), (*Bleu*, *Opel*)}.

Cette relation peut s'écrire, encore, sous la forme

<b>Voiture</b>	<b>Couleur_Voiture</b>	<b>Marque</b>
	Rouge	Fiat
	Rouge	Renault
	Vert	Opel
	Bleu	Opel

## Exemple

le nom de la relation

The diagram shows a relational database table with three columns: Voiture, Couleur\_Voiture, and Marque. The table has four rows of data. An orange arrow points from the text "le nom de la relation" to the column header "Voiture". A blue arrow points from the text "les attributs" to the column headers "Couleur\_Voiture" and "Marque". A green arrow points from the text "tuple, n-uplet ou n-uple" to the first row of the table, specifically pointing to the "Couleur\_Voiture" value "Rouge".

Voiture	Couleur_Voiture	Marque
	Rouge	Fiat
	Rouge	Renault
	Vert	Opel
	Bleu	Opel

L'attribut **Couleur\_Voiture** est un sous-ensemble du **domaine Couleur**  
et l'attribut **Marque** est un sous-ensemble du **domaine Constructeur**.

## **Notion de schéma de relation**

C'est le nom d'une relation suivi de la liste des attributs qui la constituent.

Une relation est représentée par son nom suivi de la liste de ses attributs.

**Par exemple** la relation Voiture s'écrit : **Voiture (Couleur\_Voiture, Marque)**.

<b>Voiture</b>	<b>Couleur_Voiture</b>	<b>Marque</b>
	Rouge	Fiat
	Rouge	Renault
	Vert	Opel
	Bleu	Opel

## Schéma relationnel

On peut représenter les tables d'une base de données relationnelle par un schéma relationnel dans lequel les tables sont appelées relations et les liens entre les clés étrangères et leur clé primaire est symbolisé par un connecteur :

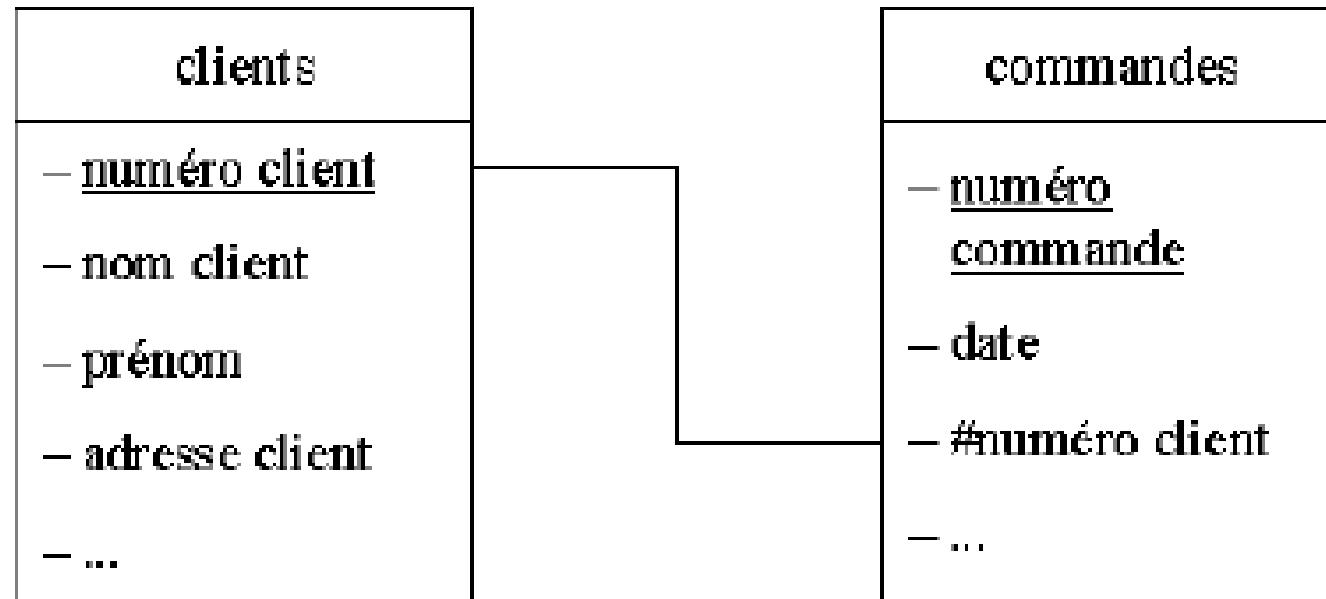


Figure 1.1 Schéma relationnel

# Notion de clé et trois premières formes normales

## Notion de clé d'une relation

C'est un sous-ensemble  $X$  des attributs d'une relation  $R(A_1, A_2, \dots, A_n)$  tel que

- 1)  $X \rightarrow A_1, A_2, \dots, A_n$
- 2) Il n'existe pas de sous-ensemble  $Y \subset X$  tel que  $Y \rightarrow A_1, A_2, \dots, A_n$

En clair, une clé est un ensemble minimum d'attributs qui détermine tous les autres. Par exemple, N°V est une clé de la relation voiture alors que (N°V, type) n'est pas une clé.

Il peut y avoir plusieurs clés pour une relation, on les appelle clés candidates, parmi elles, l'une sera choisie pour être la clé primaire, les autres clés candidates sont des clés secondaires.

## Exemples :

Soit la relation : Etudiant (NEtud, Nom, Prénom, Matière, Note)

<b>NEtud</b>	<b>Nom</b>	<b>Prénom</b>	<b>Matière</b>	<b>Note</b>
101	Nasri	Ahmed	Bases de données	10
101	Nasri	Ahmed	Réseaux	13
101	Nasri	Ahmed	Génie Logiciel	08
102	Haddad	Omar	Bases de données	09
102	Haddad	Omar	Réseaux	11
102	Haddad	Omar	Génie Logiciel	07

On remarque que le numéro de l'étudiant, son nom et son prénom, sont répétés autant de fois que le nombre des matières  $\Rightarrow$  redondance des données.

## Exemples :

Soit la relation : Livraison (N°fourn, adrF, N°prod, prixP, qté)

N°fourn	adrF	N°prod	prixP	qté
3	Annaba	52	65	100
22	Alger	10	15	50
22	Alger	33	10	120
3	Annaba	33	10	50
3	Constantine	10	15	200

- Si un fournisseur change d'adresse et qu'un seul tuple est mis à jour, ou si un nouveau tuple est inséré pour un fournisseur connu avec une adresse différente ⇒ incohérence.
- Si un fournisseur n'a pas de livraison en cours ⇒ perte d'adresse.

## **Normalisation**

La normalisation est un processus qui permet d'éviter la redondance, la perte et l'incohérence de données dans le modèle relationnel.

## **Normalisation**

Une mauvaise conception des entités et associations représentant le monde réel modélisé conduit à des relations problématiques

Une redondance des données conduit à des risques d'incohérences lors des mise à jour

Il s'agit d'éliminer toute anomalie afin de faciliter la manipulation des relations:

### **Normalisation des relations**

= Eclatement d'une relation donnée en plusieurs relations normalisées

## **Normalisation et Décomposition:**

- La décomposition d'un schéma de relation intervient dans le cas où la phase de modélisation conceptuelle n'a pas été réalisée ou lors d'une validation du modèle relationnel .
- Il est alors indispensable de décomposer les relations de telle manière à minimiser la redondance d'informations et les anomalies de mises à jour des tuples qui s'en suivent.

## Normalisation et Décomposition:

il faut décomposer la relation R en plusieurs relations. La décomposition se fait en plaçant les attributs qui violent la FN dans une nouvelle relation avec la clé primaire ou la clé partielle.

### Exemple :

Relation **Fournisseur** (numFour, nom, adr, tel, refProd, prix)

Décomposable en :

**Fournisseur** (numFour, nom, adr, tel)

**FournisseurProduit** (numFour, refProd, prix)

## Modèle relationnel et formes normales

Notons qu'en arrivant au stade du MCD validé dans la méthode de merise, ce dernier est en 2<sup>ème</sup> forme normale d'après les règles de la normalisation au niveau du MCD brut.

Notons aussi que si une table relationnelle doit être en 3<sup>ème</sup> forme normale, il faudrait qu'elle soit en 1<sup>ère</sup> forme normale et 2<sup>ème</sup> forme normale.

# Modèle relationnel et formes normales

## Première Forme Normale (1FN)

Une relation est dite en première forme normale (**1FN**) si :

- Elle possède une clé,
- Tous ses attributs ont des valeurs simples (atomiques, non multiples, non composées, non multivalués)

Exemple:

*Table ECRIRE*

**ECRIRE**

<u>numLivre</u>	<u>Auteurs</u>
1	Alain Collongues, Jean Hugues, Bernard Laroche
2	Irène Charon
3	Christophe Dabancourt
4	Philippe Chaleat, Danier Charnay, Jean-rené Rouet

# **Modèle relationnel et formes normales**

## **Première Forme Normale (1FN)**

La table ECRIRE n'est pas en première forme normale : l'attribut Auteurs contient plusieurs valeurs.

# Modèle relationnel et formes normales

## Première Forme Normale (1FN)

Autres exemples de relations qui ne sont pas en 1FN :

Personne (no\_secu, nom, prénom, **prenom\_enfants**, **age\_enfants**)

Etudiant (no\_étudiant, nom, prénom, **diplômes**)

Salarie (matr, nom, prénom, **adresse**)

- L'adresse peut être décomposée

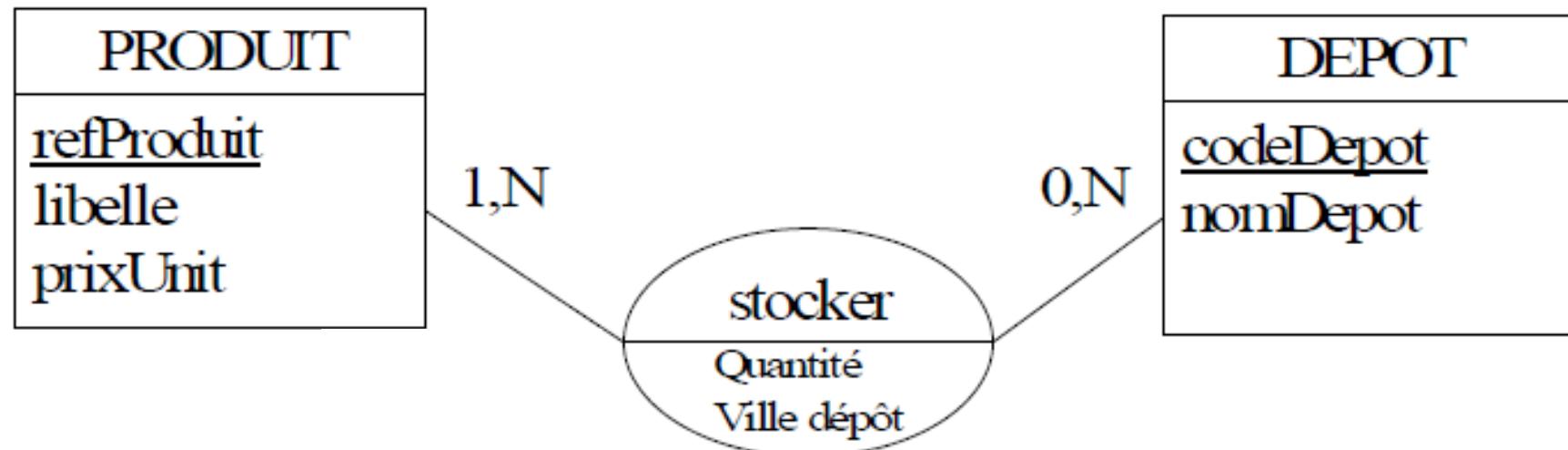
# Modèle relationnel et formes normales

## Deuxième Forme normale 2NF – dépendance pleine de la clef

Une Relation est en deuxième forme normale si :

- elle est en première forme normale
- tout attribut non-clef est en dépendance fonctionnelle de toute la clef

Exemple 2:



## Deuxième Forme normale 2NF – dépendance pleine de la clef

### Exemple 2:

L'application des règles de transformation donne la table STOCKER :

STOCKER (#refProduit, #codeDepot, quantite, ville)

*Table STOCKER*

**STOCKER**

<u>#refProduit</u>	<u>#codeDepot</u>	quantite	ville
100	A	12	ARRAS
100	B	48	LILLE
350	A	50	ARRAS
420	A	76	ARRAS

## Deuxième Forme normale 2NF – dépendance pleine de la clef

Exemple 2:

**STOCKER**

*Table STOCKER*

<u>#refProduit</u>	<u>#codeDepot</u>	<u>quantite</u>	<u>ville</u>
100	A	12	ARRAS
100	B	48	LILLE
350	A	50	ARRAS
420	A	76	ARRAS

La table STOCKER n'est pas en deuxième forme normale : en effet l'attribut « ville » n'est pas en dépendance fonctionnelle de toute la clef, mais seulement de « codeDepot ».

## Deuxième Forme normale 2NF – dépendance pleine de la clef

Exemple 3: Soit le schéma de relations :

CATALOGUE(ref\_prod, num\_fourn, libelle\_prod, nom\_fourn, adr\_fourn, prix\_achat)

Avec les dépendances :

Num\_fourn → nom\_fourn, adr\_fourn

Num\_prod → libelle\_prod

ref\_prod, num\_fourn → prix\_achat

CATALOGUE n'est pas en 2FN car **une partie de la clé** (Num\_fourn ) détermine un **attribut non clé** (nom\_fourn ) et aussi **attribut non clé** (adr\_fourn )

Et aussi pour **une partie de la clé** (Num\_prod) détermine un **attribut non clé** (libelle\_prod)

## Deuxième Forme normale 2NF – dépendance pleine de la clef

Exemple 3: Cette relation n'est pas en FN2

Soit le schéma de relations :

CATALOGUE(ref\_prod, num\_fourn, libelle\_prod, nom\_fourn, adr\_fourn, prix\_achat)

Avec les dépendances :

Num\_fourn → nom\_fourn, adr\_fourn

Num\_prod → libelle\_prod

ref\_prod, num\_fourn → prix\_achat

la décomposition nous donne :

**FOURNISSEUR** (num\_fourn, nom\_fourn, adr\_fourn)

**CATALOGUE** (ref\_prod, libelle\_prod,)

**FOURNIR** (ref\_prod, num\_fourn, prix\_achat)

# Modèle relationnel et formes normales

## Deuxième Forme normale 2NF – dépendance pleine de la clef

**Exemple 2 :**

Soit : R(Vol, AéroportDépart, Appareil, AéroportArrivée).

Avec : Vol, AéroportDépart → AéroportArrivée et

Vol →Appareil.

R n'est pas en 2FN car **une partie de la clé (Vol ) détermine un attribut non clé (Appareil)**.

# Modèle relationnel et formes normales

## Deuxième Forme normale 2NF – dépendance pleine de la clef

La décomposition de R donne :

**R1(Vol , Appareil) et**

**R2(Vol, AéroportDépart, AéroportArrivée).**

# Modèle relationnel et formes normales

## Troisième Forme normale 3FN – pas de dépendances transitives

La troisième forme normale permet d'assurer l'élimination des redondances dues aux dépendances transitives.

# Modèle relationnel et formes normales

## Troisième Forme normale 3FN – pas de dépendances transitives

Une Relation est en troisième forme normale si :

- elle est en deuxième forme normale
- tout attribut non-clef n'est en dépendance (fonctionnelle directe que de la clef et pas d'un attribut qui ne fait pas partie de la clef.)

## Troisième Forme normale 3FN – pas de dépendances transitives

Exemple 4:

COMMERCIAL
<u>numCommercial</u>
nom
codeAgence
pays

L'application des règles de transformation à produit la table relationnelle suivante:

COMMERCIAL (numCommercial, nom, codeAgence, pays)

## Troisième Forme normale 3FN – pas de dépendances transitives

Exemple 4:

**COMMERCIAL**

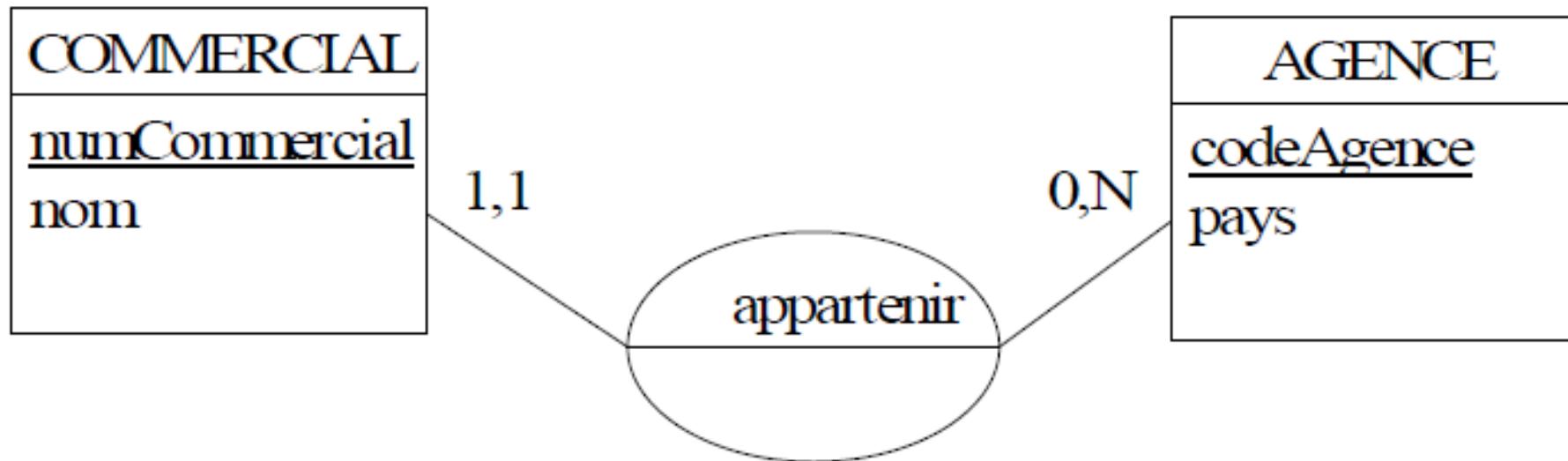
<b><u>numCommercial</u></b>	<b>nom</b>	<b>codeAgence</b>	<b>pays</b>
1	Dupont	Paris	FR
2	Durant	Londres	GB
3	Lebon	Copenhague	DK
4	Martin	Stockholm	S

La relation COMMERCIAL n'est pas en troisième forme normale : en effet l'attribut « pays » est en dépendance fonctionnelle de « codeAgence ».

## Troisième Forme normale 3FN – pas de dépendances transitives

### Exemple 4:

La modélisation correcte aurait dû être la suivante :



et ainsi les tables produites :

COMMERCIAL (numCommercial, nom, #codeAgence)

AGENCE (codeAgence, pays)

## Troisième Forme normale 3FN – pas de dépendances transitives

### Exemple 4:

PROFESSEUR (num\_prof, num\_etablissement)

ETUDIANT( num\_etu, num\_etablissement, num\_prof)

Avec les dépendances fonctionnelles :

Num\_prof → num\_etablissement

Num\_etud → num\_etablissement, num\_prof

## Troisième Forme normale 3FN – pas de dépendances transitives

Cette relation n'est pas en FN3:

**PROFESSEUR** (num\_prof, num\_etablissement)

**ETUDIANT**( num etu, num\_etablissement, num\_prof)

Le schéma de relation en FN3 est :

**PROFESSEUR** (num\_prof, num\_etablissement)

**ETUDIANT**( num etu, num\_prof)

## Troisième Forme normale 3FN – pas de dépendances transitives

### Exemple 5:

Soit : **Enseignant** (NumEns, Non, Prénom, Grade, ChargeHoraire).

Avec : Grade → ChargeHoraire.

La relation n'est pas en 3FN car un **attribut non clé** (Grade) détermine un **attribut non clé** (ChargeHoraire).

## Troisième Forme normale 3FN – pas de dépendances transitives

### Exemple 5:

Cette relation doit être décomposée en deux relations :

- **Enseignant (NumEns, Non, Prénom, Grade)**
- **Charge (Grade, ChargeHoraire).**

## Exercice

On considère la relation COMMANDE dont le schéma est donné par :

**COMMANDE(N°Commande, date, N°Client, nom, N°Article, désignation, quantité)**

**COMMANDE**

**N°commande**

date

N°client

nom

N°article

désignation

qté

} groupe répétitif



**COMMANDE**

**N°commande**

date

N°client

nom

**ARTICLE\_COMMANDE**

**N°commande**

**N°article**

désignation

qté

## ARTICLE\_COMMANDE

N° commande \_\_\_\_\_  
N° article \_\_\_\_\_  
désignation ← \_\_\_\_\_  
qté ← \_\_\_\_\_



## LIGNE\_COMMANDE

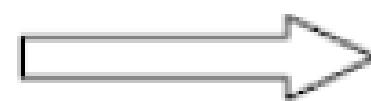
N° commande \_\_\_\_\_  
N° article \_\_\_\_\_  
qté ← \_\_\_\_\_

## ARTICLE

N° article \_\_\_\_\_  
désignation ← \_\_\_\_\_

## *COMMANDE*

N°commande \_\_\_\_\_  
date  
N°client ←  
nom ←



## *COMMANDE*

N°commande \_\_\_\_\_  
date  
N°client ←

## *CLIENT*

N°client \_\_\_\_\_  
nom ←

