

*Systèmes d'Information et Base de Données*  
*(SIDB)*  
*DEVELOPPEMENT DIGITAL*  
*2021/2022*

# METHODOLOGIE MERISE

# Introduction

- La méthode MERISE, basée sur le modèle entité-association, est un outil simple et efficace, très répandue chez les développeurs français.
- La plupart des bases de données micro pour PC (dBase, Paradox, Foxpro, Access...) sont imprégnées de cette technique pour montrer les relations entre les tables au sein d'une base.

- **MERISE** = **M**éthode pour **R**assembler les **I**dées **S**ans **E**ffort
- **MERISE** = **M**éthode d'**E**tude et de **R**éalisation **I**nformatique pour les **S**ystèmes d'**E**ntreprise

Cette méthode est actuellement enseignée aux étudiants se dirigeant vers des études informatiques, mais aussi aux élèves ingénieurs de la plupart des écoles.

Nous retrouvons là le besoin qui avait poussé les industriels à investir dans cette méthode. En effet, dans les petites et moyennes entreprises qui n'ont souvent pas de service informatique c'est le comptable qui est l'interlocuteur privilégié entre l'entreprise et le prestataire des services informatiques.

# 1 Présentation de la méthode MERISE

## 1.1 Petite histoire de la méthode MERISE

Le **modèle entité-association** est un outil (et une technique) d'analyse permettant de construire des **schémas théoriques** de raisonnement sur des applications tournant avec les **bases de données relationnelles**.

# 1 Petite histoire de la méthode MERISE

Pour mettre en œuvre cette technique, en connaître les possibilités et les limites, il n'est pas nécessaire de posséder une base de donnée.

Il existe des logiciels permettant de construire des schémas **entités-associations** et d'en analyser les conséquences logiques, puis de construire les **tables associées** aux modèles de manière entièrement automatique.

Ces logiciels sont appelés **AGL** (atelier de génie logiciel). Les logiciels TRAMIS, AMC\*Designor, SELECT... en sont des exemples.

# 1 Petite histoire de la méthode MERISE

Le modèle **entité-association** est apparu dans les travaux des chercheurs, entre 1972 et 1975 lors des travaux du français MOULIN puis de TARDIEU, TEBOUL... etc.

Il a été rendu célèbre dans le monde entier par l'américain Peter CHEN, à la suite d'une publication intitulée "The Entity-Relationship Model" (ACM, Transaction on Database Systems, 1976).

A ce jour tous les spécialistes français et/ou latins du domaine de l'analyse orientée base de données se servent de ce modèle comme outil de communication des applications **SGBDR**.

# **1 Petite histoire de la méthode MERISE**

**Il est présent de manière transparente ou plus visible, dans la plupart des logiciels de construction d'applications de bases de données comme ACCESS, PARADOX, ORACLE, SQL Server, Informix, Ingres, Sybase...**

**Il n'est en revanche pas adapté aux bases de données purement orientés objet comme O2 de Ardent Software...**



# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

La méthode Merise se caractérise par :

- Une **approche systémique** en ayant une vue de l'entreprise en terme de systèmes ;
- Une séparation des données (**le côté statique**) et des traitements (**le côté dynamique**) ;
- Une approche **par niveaux**.

# **1 Présentation de la méthode MERISE**

## **1.1 Caractéristique de la méthode MERISE**

### **1.1.1 La systémique**

La définition du Larousse semble plus explicite : « Combinaison de parties qui se coordonnent pour concourir à un résultat, de manière à former un ensemble ».

En anatomie, un système désigne un ensemble de parties similaires qui participent à une activité commune (système cardiaque, système digestif, système respiratoire, etc.).

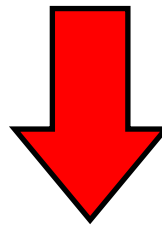
# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.1 La systémique

La représentation schématique des systèmes de l'entreprise:

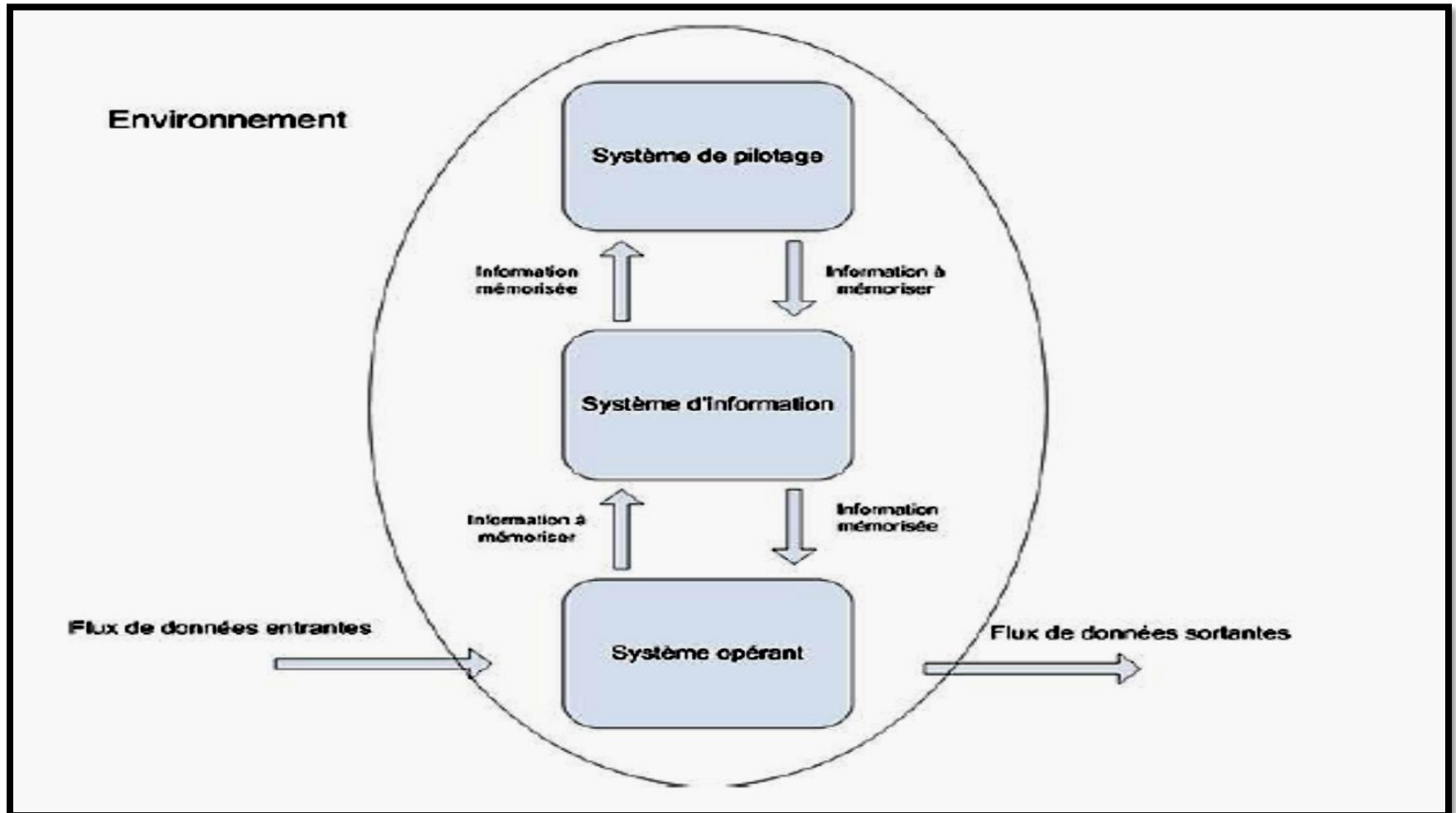
- Si nous reprenons l'analogie anatomique, et si nous comparons **l'entreprise à un corps humain**, nous pouvons réduire l'entreprise à un **cerveau qui pilote**, un **muscle qui opère** et des **nerfs qui font transiter les informations**.



# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.1 La systémique



# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.1 La systémique

#### Le système de pilotage:

- définit les missions et les objectifs,
- organise l'emploi des moyens,
- contrôle l'exécution des travaux.
- assigne des objectifs à l'organisation,
- analyse l'environnement et le fonctionnement interne à l'organisation,
- contrôle le système opérant.

**NB:** Il est aussi dénommé système de management ou de gestion ou de commande ou encore de décision.

Il est relié aux autres systèmes par des flux d'informations internes.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.1 La systémique

**Le système d'information** (ensemble des ressources humaines, techniques et financières) qui:

- traite et distribue l'information de l'organisation.
- alimente l'organisation en informations d'origines diverses (internes ou externes).
- est la passerelle obligatoire pour toutes les informations de l'entreprise.

**NB:** Il apporte les informations opérationnelles nécessaires aux activités et les informations indispensable à la gestion (contrôle, ordres ... )

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.1 La systémique

**Le système opérant** (ensemble des moyens humains, matériels, organisationnels) qui :

- exécutent les ordres du système de pilotage.
- assure le fonctionnement du système global,

C'est le niveau de la production, des opérations, de chaque service, chaque unité ou chaque entreprise considérés.

**NB:** son activité est contrôlée par le système de pilotage.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.2 La séparation des données et des traitements

Les données (ou informations):

- **l'interview:** un entretien avec les différents acteurs de l'organisation. Cet entretien permet de définir le périmètre de l'applicatif futur.
- **l'étude des documents internes:** Les documents internes (factures, bons de livraison, ordres de fabrication) recèlent des informations qui sont souvent omises lors des entretiens.
- **l'étude des documents externes:** L'étude des documents externes (factures des fournisseurs, bons de livraison fournisseurs...) tout comme l'étude des documents internes permet de découvrir des informations oubliées lors des interviews et de découvrir aussi quelques règles de gestion.



# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.2 La séparation des données et des traitements

#### Exemple d'une entreprise commerciale :

- Que se passer depuis la réponse d'un appel d'offre jusqu'à la facturation du produit.

#### Etapes :

- ❖ Envoi des tarifs
- ❖ Acceptation de la proposition concrétisée par un bon de commande.
- ❖ Validation du bon de commande si solvabilité du client
- ❖ Déclenchement de la livraison, émission d'un bon de livraison

<i>FLUX</i>	<i>INFORMATIONS ATTACHEES</i>	<i>EMETTEUR</i>	<i>RECEPTEUR</i>
<b>- Envoi des tarifs</b>	1.Référence des produits 2.Prix 3.Caractéristiques des produits 4.Conditions de vente	MARKETING	CLIENT
<b>- Bon de commande</b>	1.Quantité 2.Montant 3.Date 4.Référence des produits 5.N° de commande	CLIENT	MARKETING
<b>- Acceptation du bon de commande</b>	1.Idem « Bon de commande » 2.Solvabilité du client	MARKETING	Sce Financier Sce Compta. Sce Livraison
<b>- Bon de livraison</b>	1.Référence du client 2.Quantité 3.Référence du produit 4.Montant 5.Date 6.N° de commande	SERVICE LIVRAISON	CLIENT Sce Compta.
<b>- Facture</b>	1.N° Facture 2.Référence du client 3.Quantité et prix du produit 4.Remises éventuelles 5.Montant 6.Modalité de règlement	SERVICE COMPTABILIT E	CLIENT

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

Pour la **conception d'un SI**, il est nécessaire de considérer **quatre niveaux d'étude** :

1. Le niveau conceptuel.
2. Le niveau organisationnel.
3. Le niveau logique.
4. Le niveau physique.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

#### 1. Le niveau conceptuel:

- Le niveau conceptuel répond à la question **Quoi ?** (quoi faire, avec quelles données).

Le formalisme Merise employé sera :

- Le Modèle Conceptuel des Données (MCD).
- Le Modèle Conceptuel des Traitements (MCT): décrivant les règles et les contraintes à prendre en compte.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

## 2. Le niveau organisationnel :

- Le niveau organisationnel fera préciser les notions de temporalité, de chronologie des opérations, d'unité de lieu, définira les postes de travail, l'accès aux bases de données...

Les questions posées, au niveau des traitements, sont : Qui ? Où ? Quand ?

Le formalisme Merise employé sera :

- Le Modèle Organisationnel des Données (MOD): L'étape organisationnelle s'intéresse à l'utilisation de la base de données.
- Le Modèle Organisationnel des Traitements (MOT): consiste à décrire les contraintes dues à l'environnement (organisationnel, spatial et temporel).

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

#### 3. Le niveau logique :

- Le niveau logique est indépendant du matériel informatique, des langages de programmation ou de gestion des données.
- C'est la réponse à la question Avec quoi ?

Le formalisme sera :

- Le Modèle Logique des Données (MLD): représente un choix logiciel pour le système d'information.
- Le Modèle Logique des Traitements (MLT): reflète un choix matériel pour le système d'information.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

#### 4. Le niveau physique :

Le niveau physique permet de définir l'organisation réelle (physique) des données. Il apporte les solutions techniques, par exemple sur les méthodes de stockage et d'accès à l'information.

C'est la réponse au Comment ?

Le formalisme employé sera :

- Le Modèle Physique des Données (MPD): consiste à implémenter le modèle dans le SGBD.
- Le Modèle Opérationnel et physique des Traitements (MOpT): est une architecture d'organisation des menus devant permettre une distribution répartie des divers services transactionnels.

# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

NIVEAU	QUESTION	CONTENU	DOMAINE
<b>Conceptuel</b>	Que faire ?	<ul style="list-style-type: none"><li>- données manipulées</li><li>- règles de gestion</li><li>- enchaînement des traitements</li></ul>	Gestion
<b>Organisationnel</b>	Qui fait quoi ? Quand ? Où ?	<ul style="list-style-type: none"><li>- partage des tâches</li><li>- mode de traitement</li><li>- répartition géographique des traitements</li><li>- organisation des données</li></ul>	Organisation
<b>Physique</b>	Comment ?	<ul style="list-style-type: none"><li>- programmes</li><li>- logiciels</li><li>- matériels</li></ul>	Technique



# 1 Présentation de la méthode MERISE

## 1.1 Caractéristique de la méthode MERISE

### 1.1.3 Une approche par niveaux

NIVEAU	DONNEES	TRAITEMENT
<b>Conceptuel</b>	<b>MCD</b> Modèle Conceptuel des Données	<b>MCT</b> Modèle Conceptuel des Traitements
<b>Organisationnel</b>	<b>MLD</b> Modèle logique des données	<b>MOT</b> Modèle Organisationnel des Traitements
<b>Physique</b>	<b>MPD</b> Modèle Physique des Données	<b>MOPT</b> Modèle Opérationnel des Traitements

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

**Le dictionnaire des données** est un document qui permet:  
de recenser, de classer et de trier toutes les informations (les données)  
collectées lors des entretiens ou de l'étude des documents.

### Exemple :

Nom de la donnée	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			Élémentaire	Calculé			

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

### Nom de la donnée:

- Cette cellule recevra une donnée par exemple : Nom client.

### Format:

- Ici sera indiqué le format de la donnée, par exemple : alphabétique.

### Longueur:

- La longueur approximative ou exacte de la donnée sera indiquée, par exemple : 30.

### Type:

- Une croix sera inscrite dans la colonne pour indiquer si la donnée est élémentaire ou calculée.

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

### Règle de calcul:

- Ici sera indiquée de manière claire la formule ou le calcul nécessaire à appliquer pour obtenir la donnée.

### Règle de gestion:

- Dans cette zone sera indiquée, si nécessaire, la règle de gestion inhérente à la donnée.

### Document:

- La rubrique document permet de saisir le document dans lequel a été trouvée la donnée.

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

Exemple:

Nom de la donnée	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			E	C			
Nom client	Alphabétique	30	X				Facture

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

### Exemple d'application:

*Etablir le dictionnaire des données de la gestion des adhérents.  
Voici une représentation d'une fiche d'adhérent :*

**Association des Palanges**

**Fiche Adhérent**

Numéro	66
Nom :	BAPTISTE
Prénom :	Jean-Luc
Adresse :	Rue de la forêt
Code Postal :	12000
Ville :	Rodez
Téléphone :	05-65-42-00-00
Mail :	jeanluc.baptiste@btsig.org
Date d'adhésion :	20 décembre 2007

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

### Exemple d'application:

À la lecture de la fiche, nous pouvons déterminer la présence de neuf informations différentes :

- ❖ Le numéro de l'adhérent.
- ❖ Le nom.
- ❖ Le prénom.
- ❖ L'adresse.
- ❖ Le code postal.
- ❖ La ville.
- ❖ Le téléphone.
- ❖ Le mail.
- ❖ La date d'adhésion.

# 1 Présentation de la méthode MERISE

## 1.2 Dictionnaire de données:

Voici le dictionnaire des données :

Nom	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			E	C			
Numéro	Numérique		X				Fiche
Nom	Alphabétique	30	X				//
Prénom	Alphabétique	30	X				//
Adresse	Alphabétique	50	X				//
Code Postal	Alphanumérique	10	X				//
Ville	Alphabétique	50	X				//
Téléphone	Alphanumérique	15	X				//
Mail	Alphanumérique	50	X				//
Date d'adhésion	Date		X				//



# **1 Présentation de la méthode MERISE**

## **1.3 Règles de gestion:**

- **Les règles de gestion sont des phrases (simples, en général, 1 verbe) écrites en « français ».**
- **Les règles de gestion décrivent les « actions » qui sont à modéliser dans le système.**
- **Les règles de gestion décrivent un « processus » utilisé par l'entreprise.**

# **1 Présentation de la méthode MERISE**

## **1.3 Règles de gestion: Exemples**

- ❖ « Un inventaire des stocks doit être dressé chaque mois ».
- ❖ « Une commande non livrable sera mise en attente ».
- ❖ « Les responsables de secteur peuvent changer »

## 2 Modèle Conceptuel des Données (modèle entités-associations)

- Le modèle entités-associations est constitué de deux éléments de base :
  - Les **propriétés**, sont les informations de base du système d'information.
  - Les **entités**, qui sont des regroupements d'informations, et possèdent des attributs (caractéristiques)
  - Les **associations** qui sont les liens logiques entre les entités (et sont quantifiées par des cardinalités)

## **2 Modèle Conceptuel des Données**

### **2.1 Les propriétés (attributs):**

Un client possède un numéro de client, un nom, un prénom, habite à une adresse précise, etc.

Ces informations élémentaires essentielles sont des propriétés.

Les propriétés disposent d'un type. Elles peuvent être numériques, représenter une date, leur longueur peut être aussi définie.

**Par exemple :** le nom est une propriété de type alphabétique et de longueur 50.

## 2 Modèle Conceptuel des Données

### 2.1 Les propriétés (attributs):

*Les attributs sont les caractéristiques décrivant les entités et doivent être représentés comme une **liste de mots**, la plus simple possible, dans le cadre de l'entité correspondante. On devra préciser le **type des données** attendues pour chaque attribut.*

## 2 Modèle Conceptuel des Données

### 2.1 Les propriétés (attributs):

Exemple:

ETRE HUMAIN		ADRESSE	
Nom	A32	Ligne adresse 1	A32
Prénom	A25	Ligne adresse 2	A32
Date de naissance	D	Ligne adresse 3	A32
Lieu de naissance	A32	Code postal	A7
Sexe	BL	Ville	A32
		Pays	A32

## 2 Modèle Conceptuel des Données

### 2.1 Les propriétés (attributs):

Exemple:

- Les types associés aux attributs sont les suivants :

D	Date
A20	Caractères de longueur 20
BL	Booléen (vrai / faux)
T	Temps
DT	Date Temps
N	Nombre
S	(Smallint) entier court
I	(Integer) entier

## **2 Modèle Conceptuel des Données**

### **2.2 Les propriétés (attributs):**

#### **Identifiant (Clef):**

Une de ces propriétés a un rôle bien précis, c'est l'identifiant nommé aussi la clé.

- L'identifiant permet de connaître de façon sûre et unique l'ensemble des propriétés qui participent à l'entité.

#### **Par exemple:**

le fait de connaître la ville d'un client permet-il de connaître son nom ?

La réponse est non.

La connaissance du nom du client permet-elle de connaître sa ville ? La réponse est toujours non.



## **2 Modèle Conceptuel des Données**

### **2.2 Les propriétés (attributs):**

#### **Identifiant (Clef):**

**Une propriété qui lorsque sa valeur est connue permet la connaissance de l'ensemble des valeurs qui s'y rattachent de façon formelle.**

**Le numéro du client est connu, son nom, son prénom et toutes les valeurs des autres propriétés qui s'y rattachent sont connues de façon sûre et unique.**

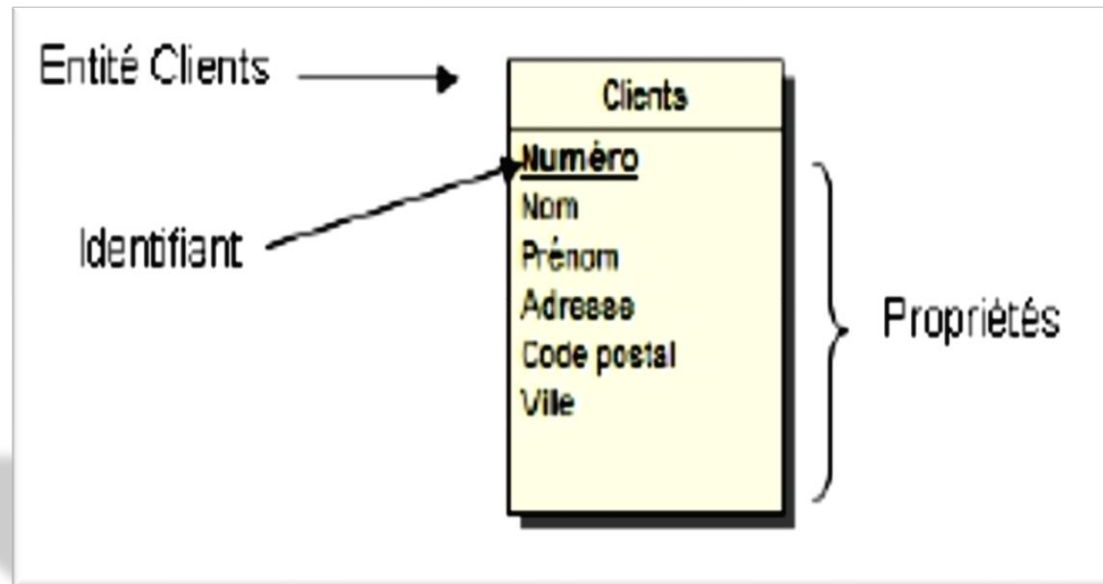
## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Identifiant (Clef):

*Au niveau du formalisme, cette propriété se souligne.*

*Voici le schéma modifié de l'entité **Clients**.*



## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Identifiant (Clef):

##### Définition:

C'est un attribut (ou un ensemble d'attributs) qui permet de distinguer un élément de l'entité de manière unique et sans aucune ambiguïté par rapport à l'ensemble des autres éléments, et à l'univers de tous les éléments qui peuvent entrer un jour ou l'autre dans cette entité.

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Identifiant (Clef):

##### Exemple:

la clef de l'entité "être humain" pourrait être **le nom et le prénom**.  
Cependant il n'est toujours pas impossible d'avoir deux personnes dont le nom et le prénom soient identiques...

##### Notation:

On note qu'un attribut est une clef en le soulignant dans le schéma entité association.

Si c'est une clef composée, alors plusieurs entités seront soulignées.

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Identifiant (Clef):

Remarque 1: On voit ici que dans le cas de l'entité "appartement" tous les attributs sont utilisés pour composer la clef.

Cette clef naturelle n'étant pas pratique, il est plus judicieux de créer un nouvel attribut qui servira expressément de clef à l'association.

Remarque 2: Pour l'entité "être humain", on pourrait se servir du **numéro de sécurité sociale** (plus exactement du numéro INSEE), comme clef de l'entité.

En revanche, pour ce qui est de l'entité "**appartement**" il est conseillé de créer un nouvel attribut clef qui serait, par exemple, un **numéro**.

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Identifiant (Clef):

ETRE HUMAIN	
<u>Numero secu</u>	A13
Nom	A32
Prénom	A25
Date de naissance	D
Lieu de naissance	A32
Sexe	BL

APPARTEMENT	
<u>Numero appartement</u>	I
Ligne adresse 1	A32
Ligne adresse 2	A32
Ligne adresse 3	A32
Code postal	A7
Ville	A32
Pays	A32
Bâtiment	A8
Escalier	A8
Etage	A3
Porte	A4

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Discussion de la qualité d'une clef

De manière générale il convient de **limiter les clefs composées**.

Chaque fois que l'on aura le **choix entre la création d'une clef numérique, et une clef naturelle mais composée**, il sera préférable de créer une **clef numérique**.

En effet, les **SGBDR** sont plus « à l'aise » lorsqu'ils ont à manipuler des clefs purement numérique. De plus une clef est un concept purement informatique.

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Clef évolutive

Pour une entité de type « **Voiture** » il pourrait être fait usage de **l'immatriculation** du véhicule comme **clef de l'entité**.

L'immatriculation d'un véhicule est une mauvaise clef :

en effet, du fait de la fiscalité sur les véhicules à moteur (et en particulier les vignettes), les sociétés n'hésitent pas à faire immatriculer leur parc de véhicules dans le département où les taxes sont les moins élevées (le 51).

Cette immatriculation peut donc être amenée à changer.



## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Clef évolutive

Or toute **clef évolutive** est un **danger** pour le système informatique :  
si la valeur de la clef change, nous verrons qu'il faut la modifier dans tous les fichiers dans laquelle elle est référencée.

#### **Remarque:**

**On veillera donc à prendre une clef totalement indépendante des attributs ordinaires de l'entité**

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Clef informatique:

Le plus simple consiste donc à introduire dans le descriptif de l'entité une clef strictement « informatique » qui se résumera en général à **un numéro (entier long)** que l'on pourra incrémenter automatiquement.

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Technique de la double clef:

Une technique éprouvée consiste à introduire une **double clef** dans toutes les tables : la clef « **informatique** » et une clef « **utilisateur** ».

## **2 Modèle Conceptuel des Données**

### **2.2 Les propriétés (attributs):**

#### **Clef informatique :**

- **La clef informatique est l'index primaire de la table et doit posséder les caractéristiques suivantes :**
  - **purement numérique (par exemple un entier long)**
  - **unique bien entendu**
  - **obligatoire**
  - **sans mise à jour en cascade**
  - **générée automatiquement**
  - **invisible pour l'utilisateur.**

## 2 Modèle Conceptuel des Données

### 2.2 Les propriétés (attributs):

#### Clef utilisateur :

- La **clef utilisateur** doit être assez « souple », c'est à dire posséder dans la mesure du possible, les caractéristiques suivantes :
  - Index unique
  - Obligatoire
  - Utiliser un jeu de caractère réduit s'il s'agit d'un format alpha (par exemple les 26 lettres majuscules de l'alphabet et les chiffres de 0 à 9)
  - Limité à une faible taille (16/32 octets - 16 caractères maximum)

## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

*Ce sont des regroupements d'informations.*

Les informations contenues dans les entités (informations que l'on appelle "attributs") doivent être des informations variables, mais communes à une même classe d'objets.

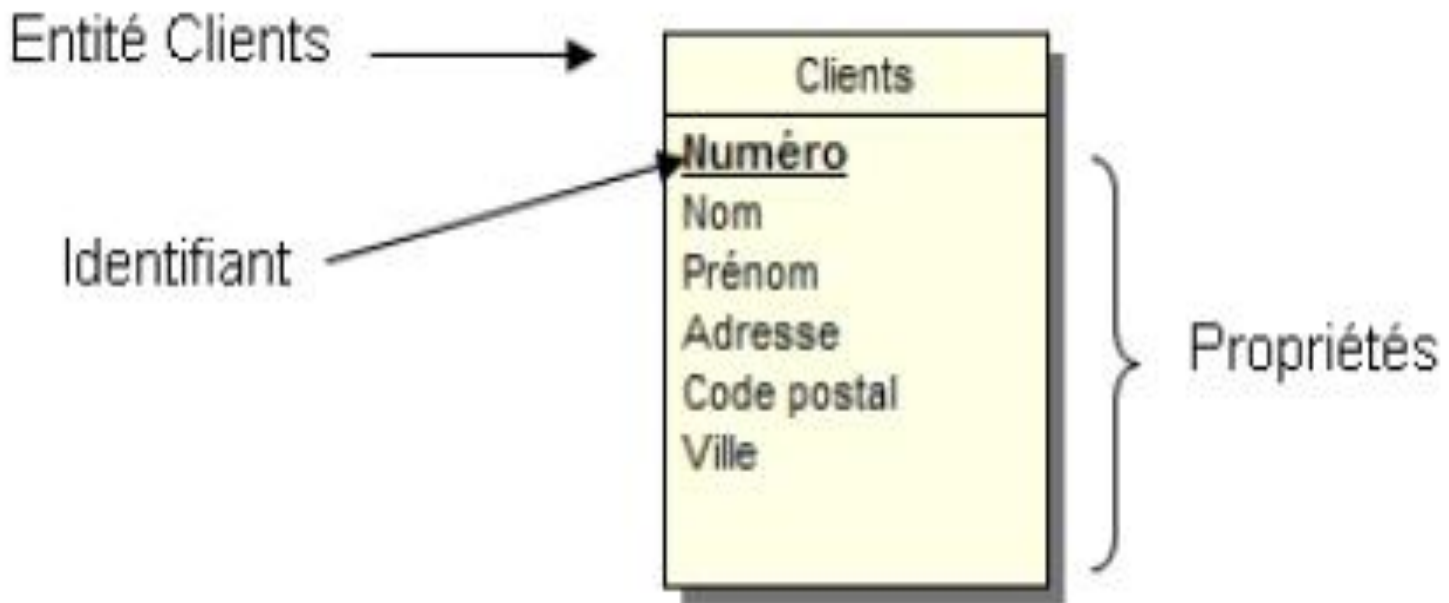
#### **Exemple 1:**

les clients sont définis par certaines propriétés (numéro, nom, prénom...).  
Le fait de les regrouper amène naturellement à créer une entité Clients.

## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

Le symbolisme retenu est le suivant :



## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

#### Exemple2:

Si l'on considère l'entité **"être humain"** les informations communes aux être humains peuvent être :

- le nom,
- le prénom,
- la date de naissance,
- le lieu de naissance,
- le sexe,
- l'adresse,
- etc...



## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

Par exemple, un être humain donné peut habiter au même endroit qu'un autre (si deux personnes vivent sous le même toit parce qu'ils sont mariés). Dans ce cas, **l'adresse constitue une sous-classe** de l'entité "être humain", **c'est à dire une nouvelle entité à part entière.**

- D'un autre côté, il arrive souvent que plusieurs personnes résident au même endroit, sans même se connaître (cas d'un immeuble collectif par exemple).

## 2 Modèle Conceptuel des Données

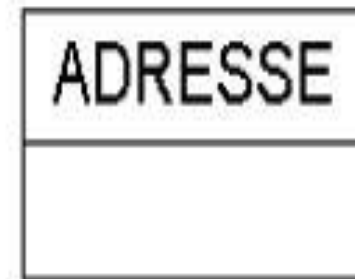
### 2.2 Les entités:

- Dans ce cas on peut considérer l'**adresse**, comme une **entité** et la décrire de la manière suivante :
  - Pays
  - Région
  - Département
  - Rue etc...

## 2 Modèle Conceptuel des Données

### 2.2 Les entités: Schématisation

On schématise une entité par un rectangle:



## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

#### Entité temporelle:

On appelle alors cela une **entité temporelle**. Une entité temporelle possède souvent un seul attribut, mais dans le cas où elle possède plusieurs attributs (année, mois, jour, heure, minute, seconde...), l'ensemble de ces attributs constitue alors la clef de l'entité.

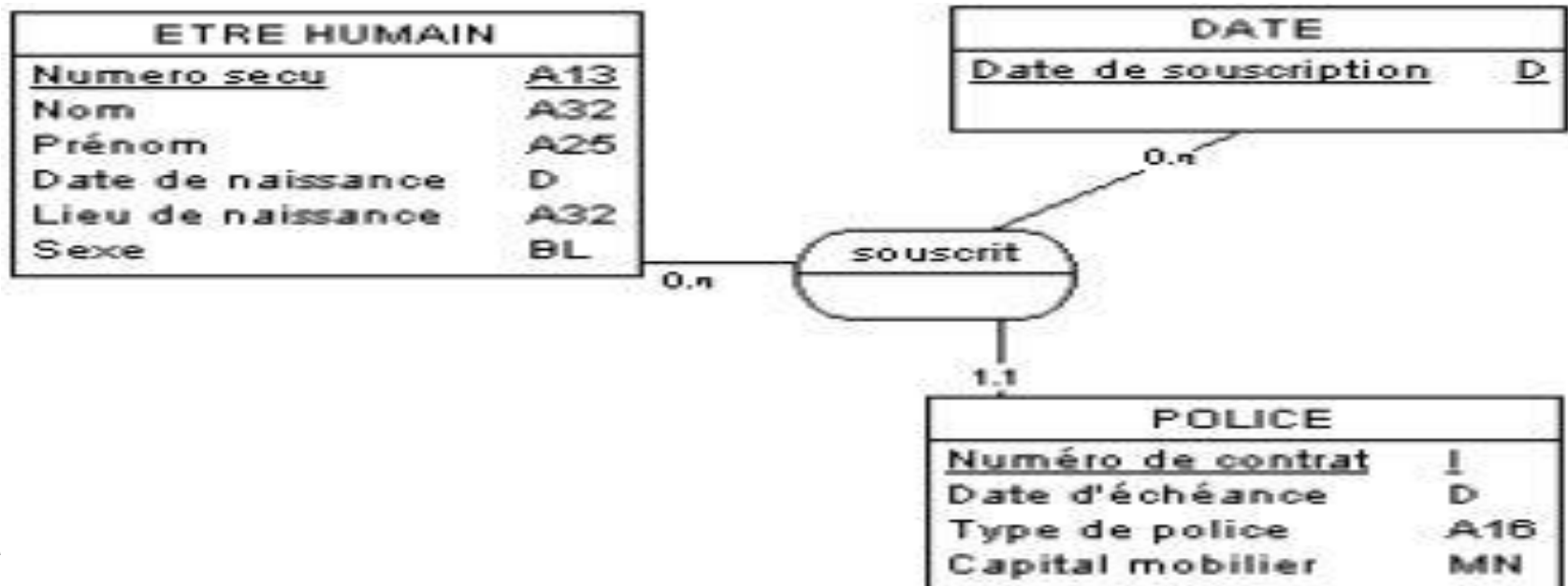
## 2 Modèle Conceptuel des Données

### 2.2 Les entités:

#### Entité temporelle:

Il arrive dans certains cas que l'attribut "**date**" soit d'une importance capitale, notamment dans les applications SGBDR portant sur la signature de contrats à échéance ou dans la durée (assurance par exemple).

Il n'est pas rare alors que le seul attribut "**date**" constitue à lui seul une entité.



## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

*Ce sont des liaisons logiques entre les entités.*

Elles peuvent être de nature factuelle, ou de nature dynamique. Par exemple, une personne peut acheter un objet (action d'acheter), mais si l'on considère qu'une personne est propriétaire d'un objet, alors l'association entre l'objet et cette personne est purement factuelle.

## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Exemple 1:

Un client peut commander des articles.

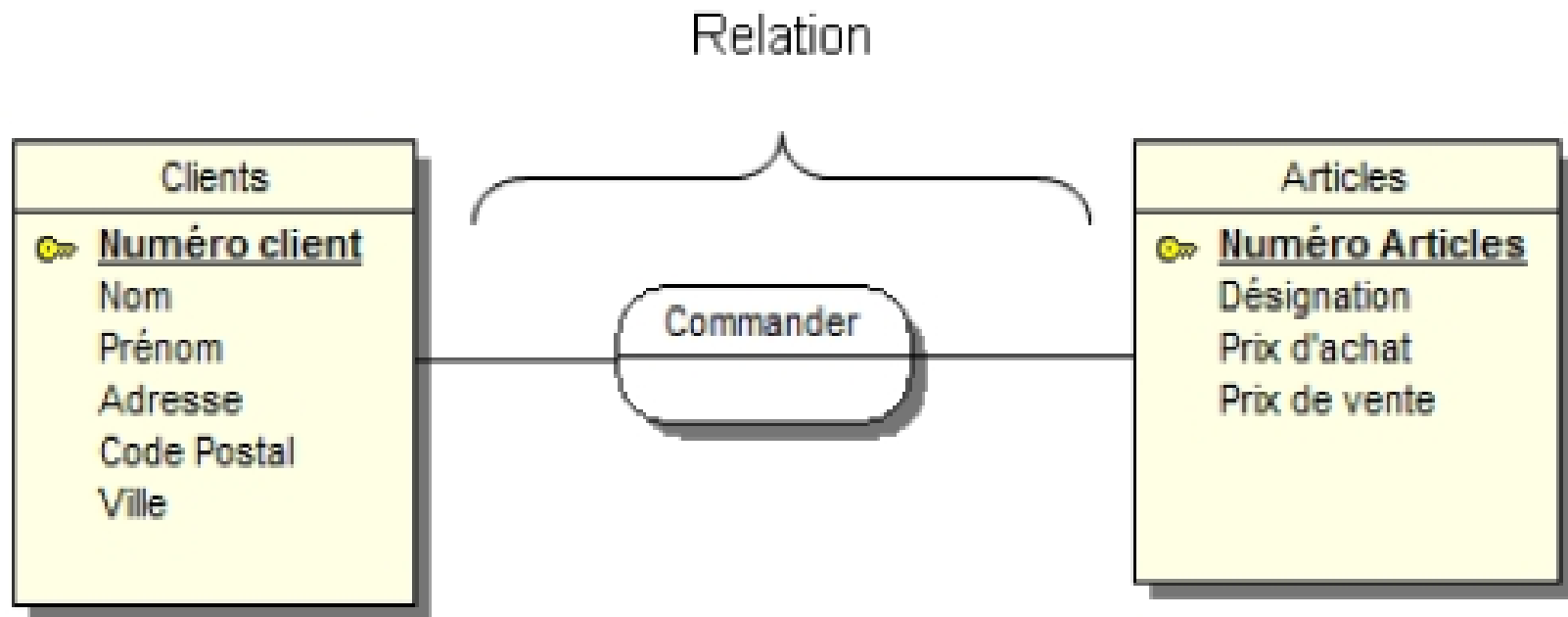
Si nous analysons cette phrase, on distingue deux entités (**clients** et **articles**) et un verbe (**commander**) qui indique un lien entre clients et articles.

Formalisons cette phrase avec Merise.

## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Exemple 1:

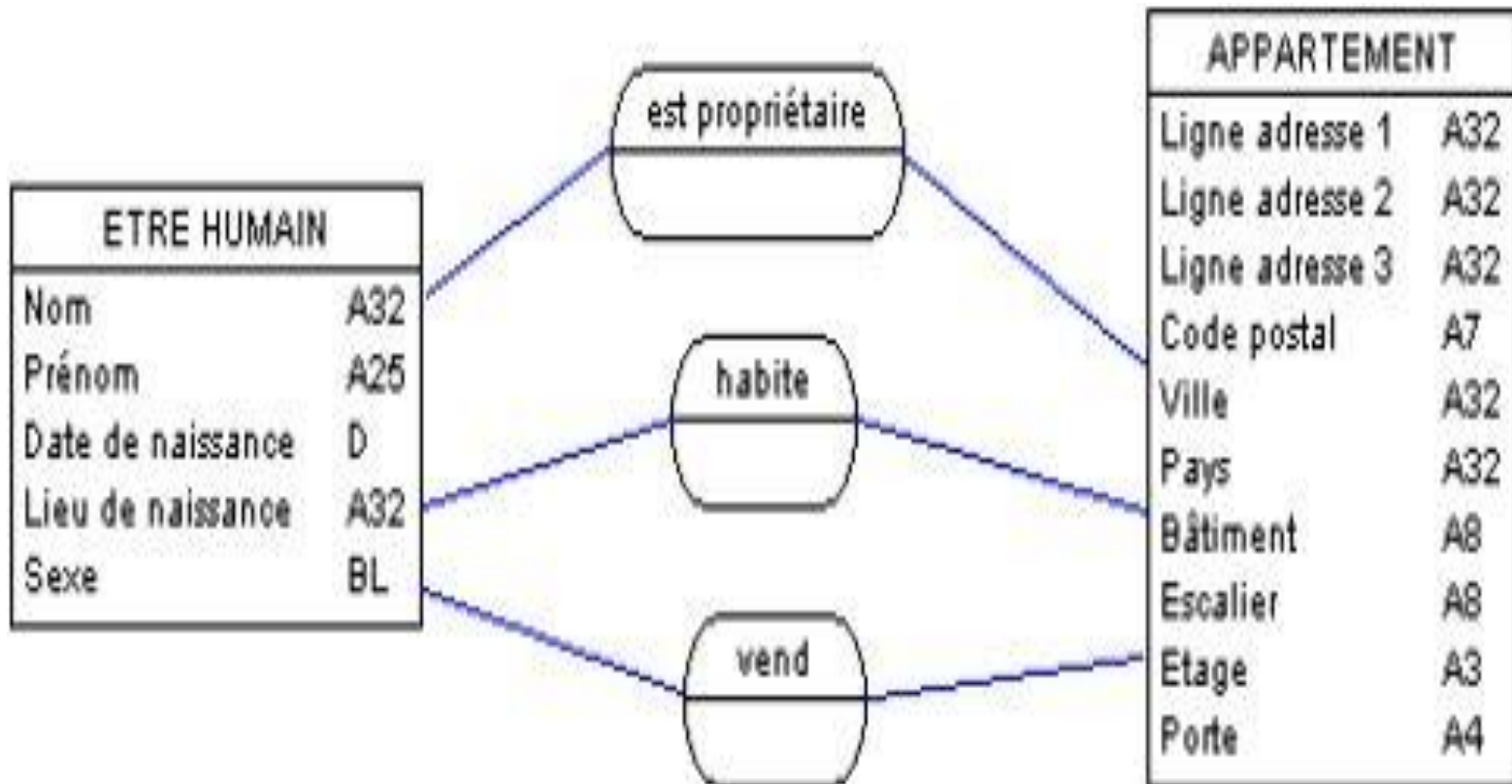




## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Exemple2:



## **2 Modèle Conceptuel des Données**

### **2.2 Les associations (relations):**

#### **Cardinalités:**

**Elles expriment le nombre de fois ou l'occurrence d'une entité participe aux occurrences de la relation.**

**Dans notre exemple on peut se poser les questions suivantes :**

- Combien de fois au minimum un client peut-il commander un article ?**
- Combien de fois au maximum un client peut-il commander un article ?**

## 2 Modèle Conceptuel des Données

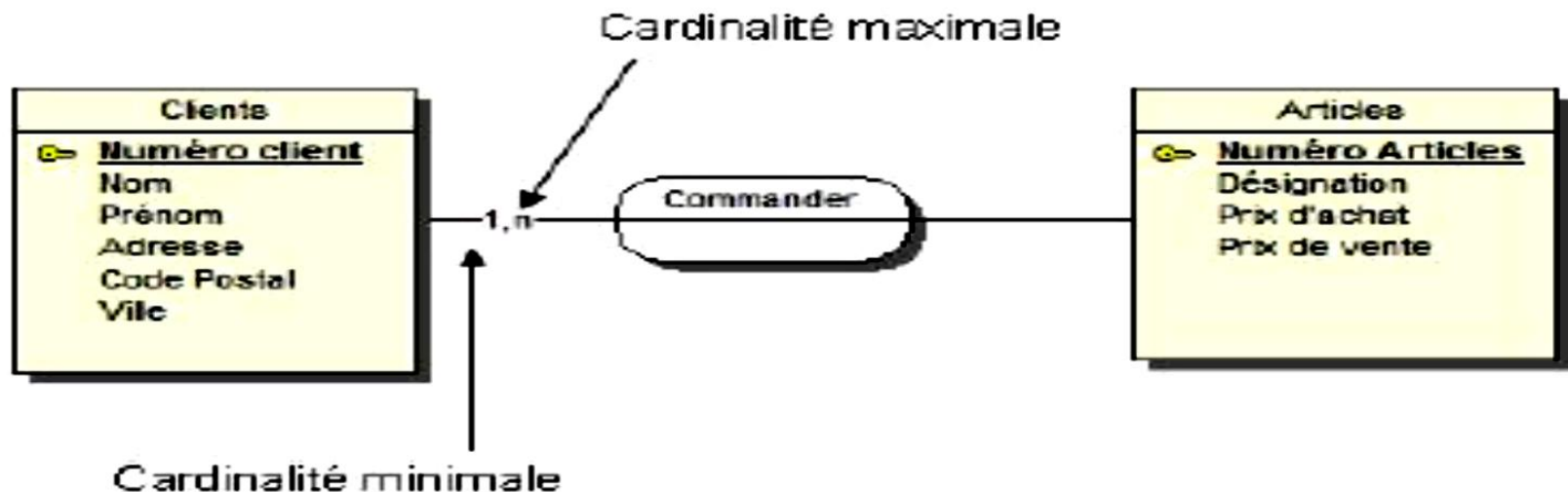
### 2.2 Les associations (relations):

#### Cardinalités:

À la première question, nous pouvons répondre qu'un client, pour être client, doit commander au moins un article.

À la deuxième question, nous pouvons répondre qu'un client peut commander plusieurs articles.

Voici comment symboliser cet état :



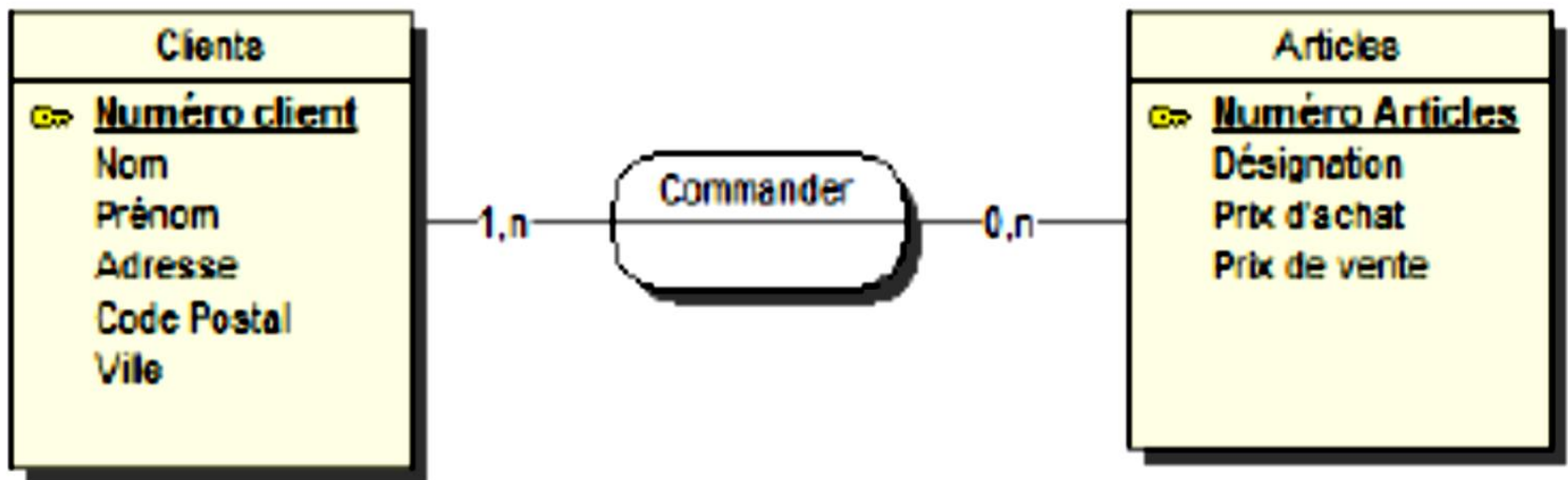
## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Cardinalités:

Le n représente la notion de « plusieurs » ; ici nous avons représenté le fait qu'un client peut commander un ou plusieurs articles. Il faut que nous nous posions les mêmes questions pour l'article :

- Combien de fois au minimum un article peut-il être commandé par un client ?
- Combien de fois au maximum un article peut-il être commandé par un client ?



## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Cardinalités:

Les cardinalités, au sens arithmétique du terme, permettent de dénombrer les éléments de l'entité d'arrivée en relation avec un élément de l'entité de départ, et vice versa.

#### Définitions:

La cardinalité minimale (**0 ou 1**) exprime le nombre de fois minimum qu'une occurrence d'une entité **participe aux occurrences** d'une relation.

La cardinalité maximale (**1 ou n**) exprime le nombre de fois maximal qu'une occurrence d'une entité **participe aux occurrences** de la relation.

## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Cardinalités: Exemple

Considérons le cas de l'association "habite" et les deux entités "être humain" et "appartement" du schéma précédent :

- les cardinalités minimales et maximales sont les suivantes : (résider)
  - sens "être humain" vers "appartement" : 1 (minimum) et 1 (maximum)
  - sens "appartement" vers "être humain" : 0 (minimum) et n (maximum)

Ce qui signifie que dans cette modélisation un être humain réside dans un appartement et un seul à la fois, mais qu'un appartement peut se trouver vide ou être pourvus de plusieurs résidents.

## **2 Modèle Conceptuel des Données**

### **2.2 Les associations (relations):**

#### **Cardinalités: Exemple**

Elles expriment le nombre de fois ou l'occurrence d'une entité participe aux occurrences de la relation.

- Dans notre exemple on peut se poser les questions suivantes :
- Combien de fois au minimum un client peut-il commander un article ?
- Combien de fois au maximum un client peut-il commander un article ?

## 2 Modèle Conceptuel des Données

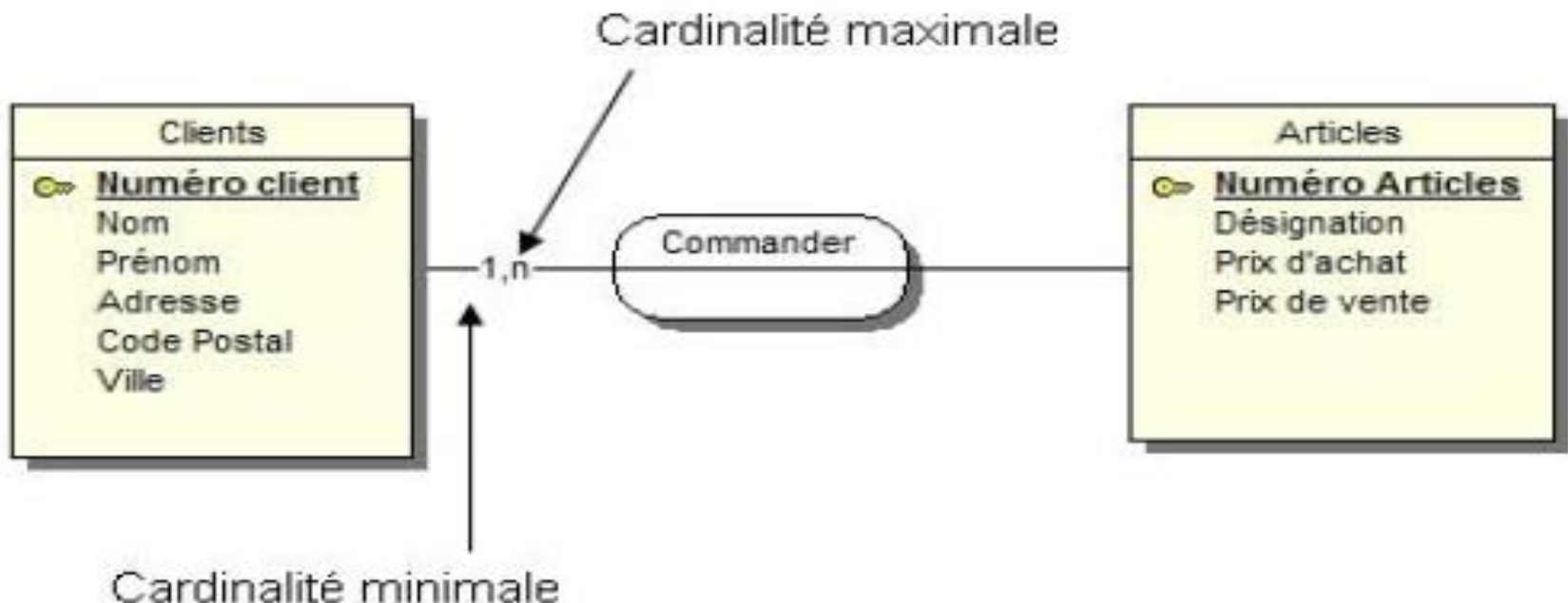
### 2.2 Les associations (relations):

#### Cardinalités: Exemple

À la première question, nous pouvons répondre qu'un client, pour être client, doit commander au moins un article.

À la deuxième question, nous pouvons répondre qu'un client peut commander plusieurs articles.

Voici comment symboliser cet état :



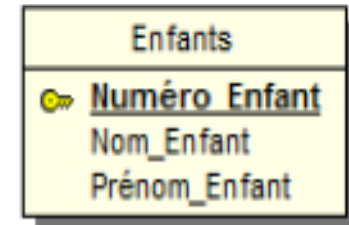
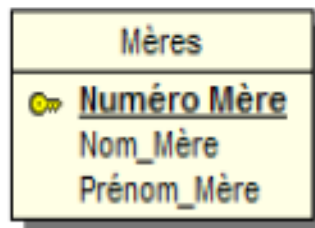


## 2 Modèle Conceptuel des Données

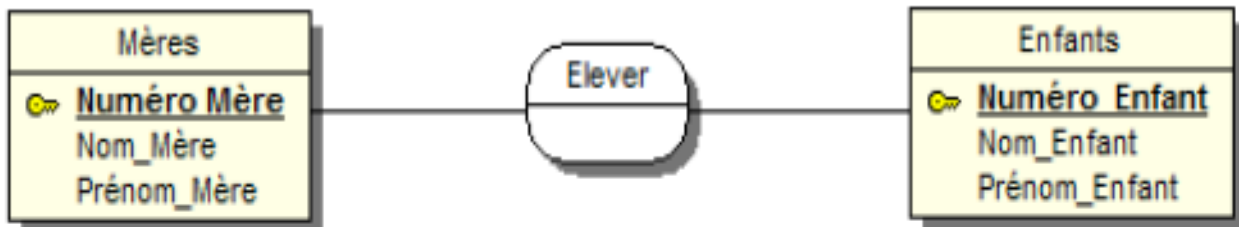
### 2.2 Les associations (relations):

#### Cardinalités: Autre exemple

- Modélisons le fait qu'une mère élève des enfants.
- Nous avons deux entités Mères et Enfants :



Une relation Elever .



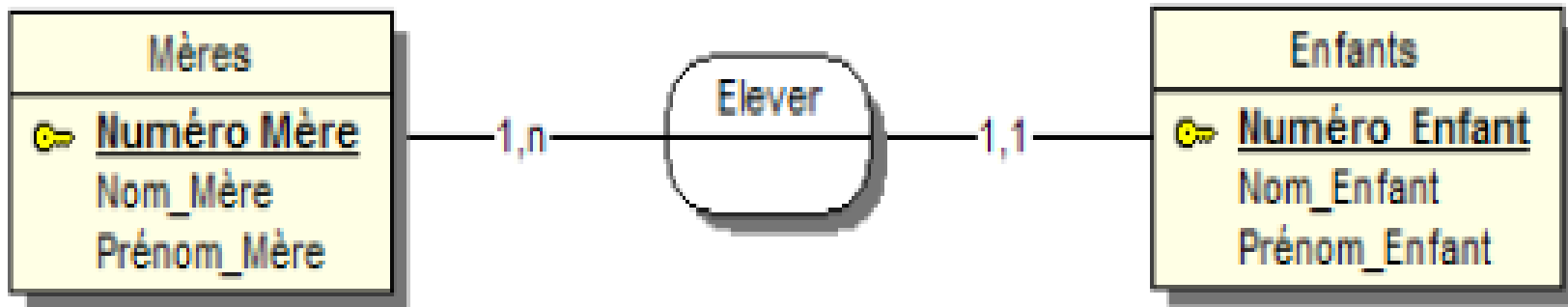
## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Cardinalités: Autre exemple

Des cardinalités :

- Une mère peut élever un ou plusieurs enfants.
- Un enfant peut être élevé par une et une seule mère.



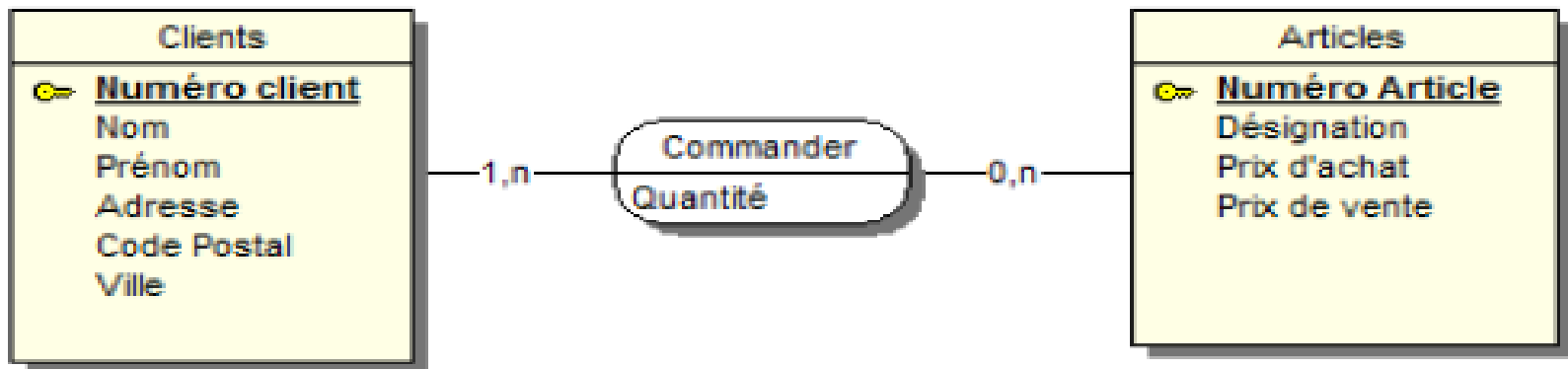
## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Relation porteuse de données

##### Définition:

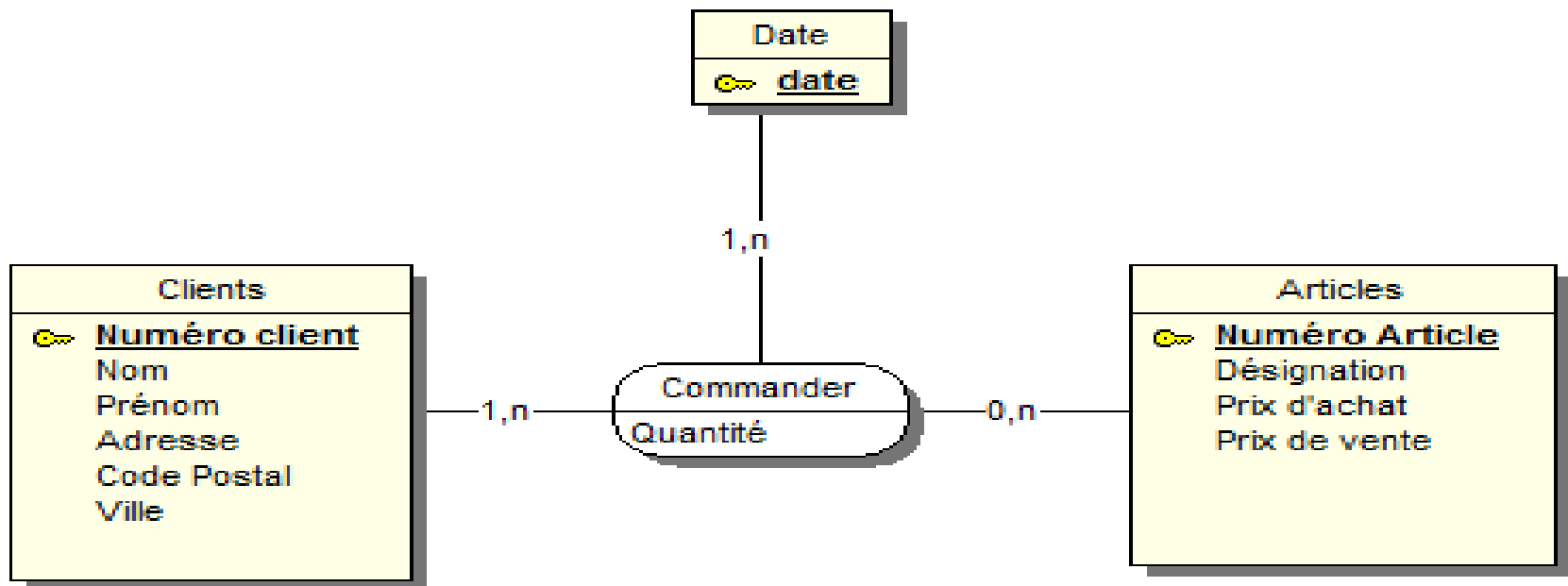
Une relation est dite porteuse lorsqu'elle contient des propriétés.



## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Relation ternaire (ou Tri-pattes):



## 2 Modèle Conceptuel des Données

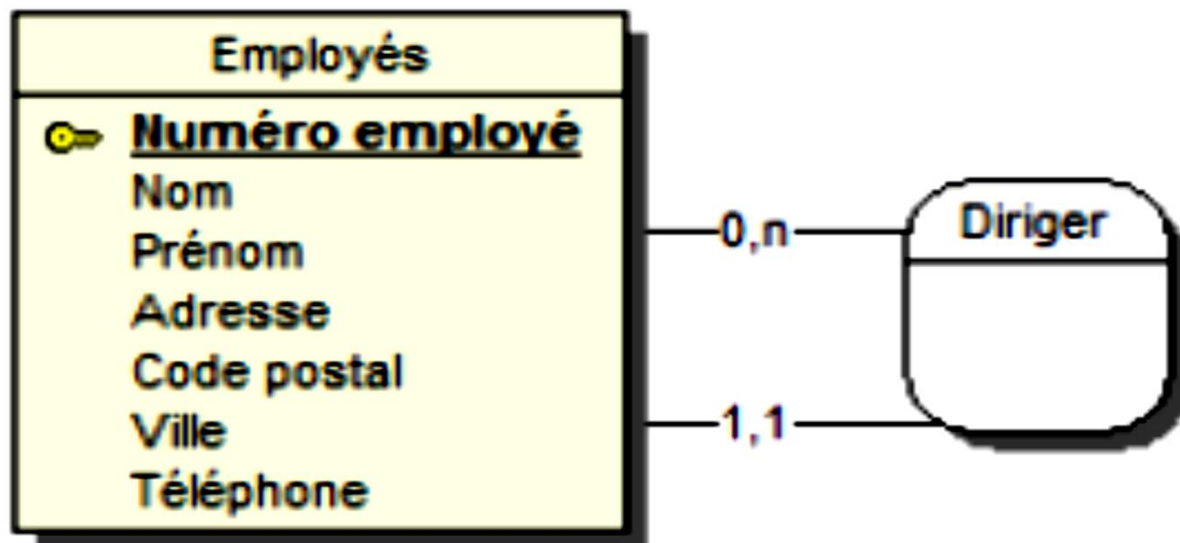
### 2.2 Les associations (relations):

#### Relation réflexive:

##### Définition:

Une relation réflexive est une relation d'une entité sur elle même.

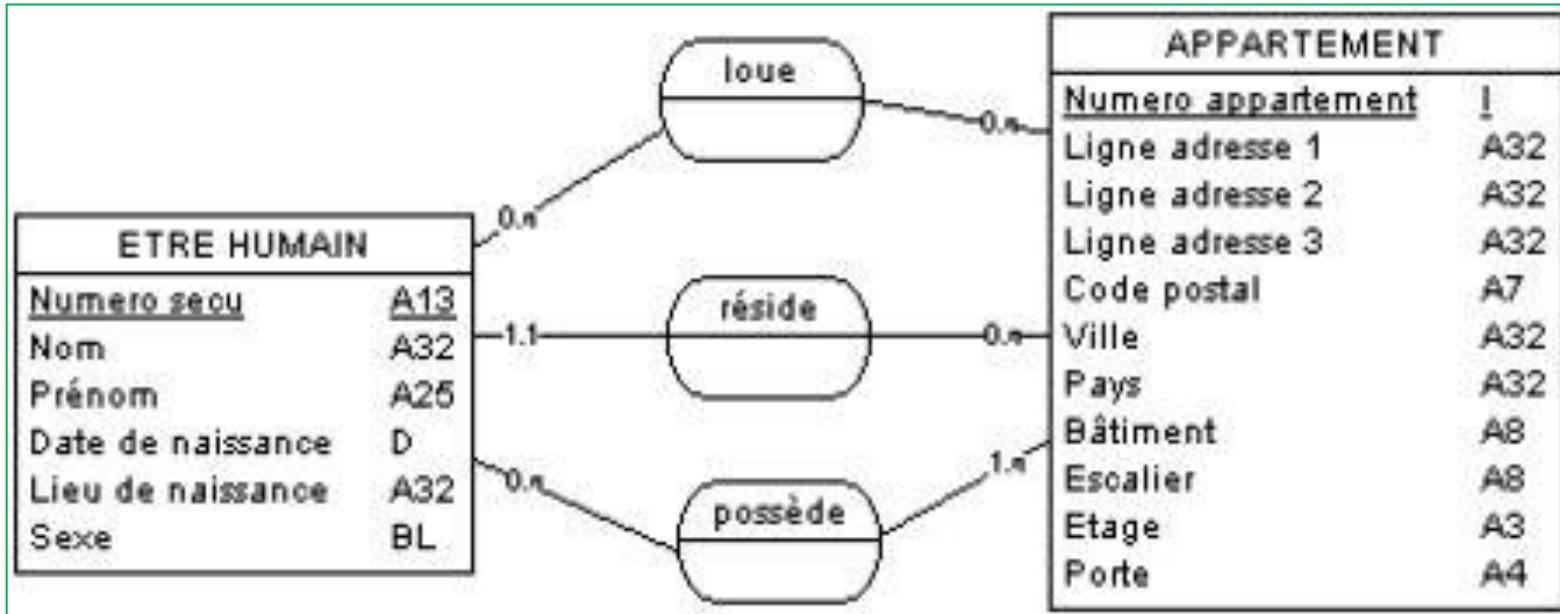
Par exemple, on désire modéliser le fait qu'un employé peut diriger d'autres employés.



## 2 Modèle Conceptuel des Données

### 2.2 Les associations (relations):

#### Types des relations:



La relation « **loue** » est de type n :m

La relation « **réside** » est de type 1 :n

La relation « **possède** » est de type n :m

### 3. Topologie des associations

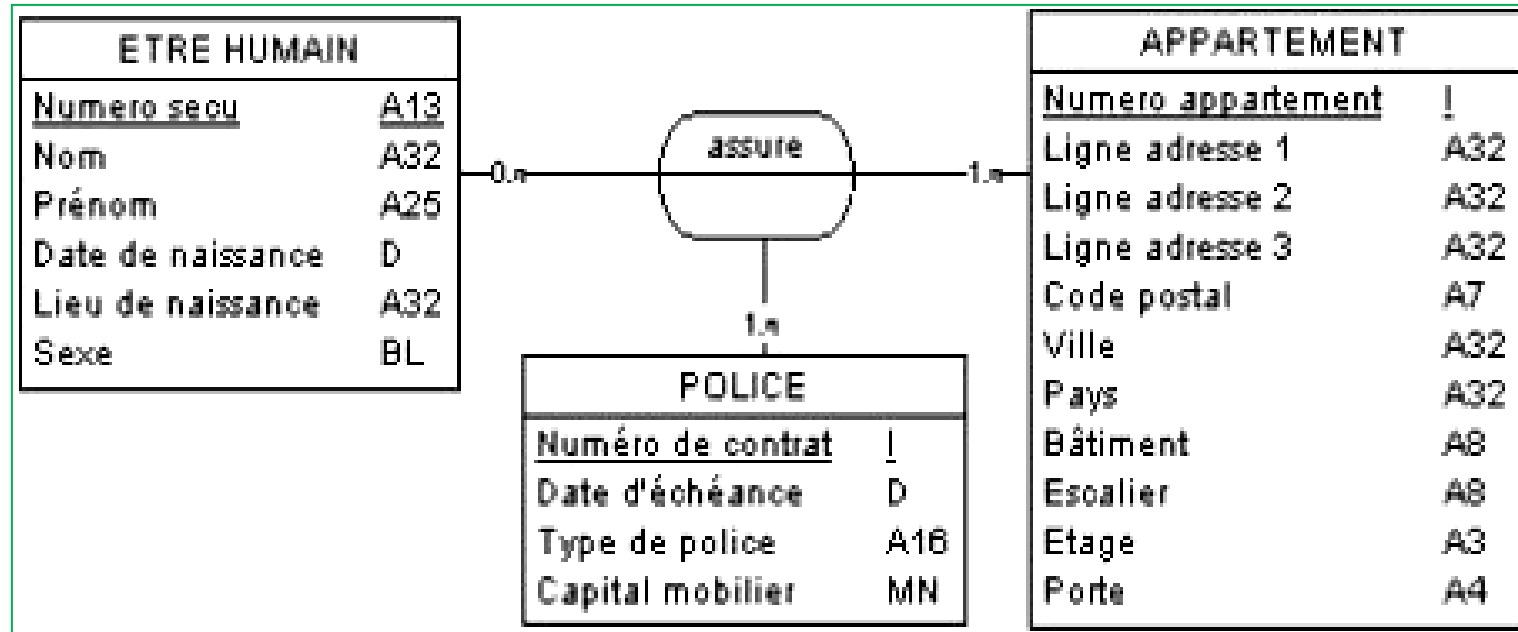
La plupart des associations sont de nature **binaire**, c'est à dire composées de deux entités mise en relation par une ou plusieurs associations.

C'est le cas par exemple de l'association "est propriétaire" mettant en relation "être humain" et "appartement".

Cependant il arrive qu'une **association concerne plus de deux entités** (on dit alors qu'il s'agit d'association "**n-aires**").

### 3. Topologie des associations

#### Exemple:

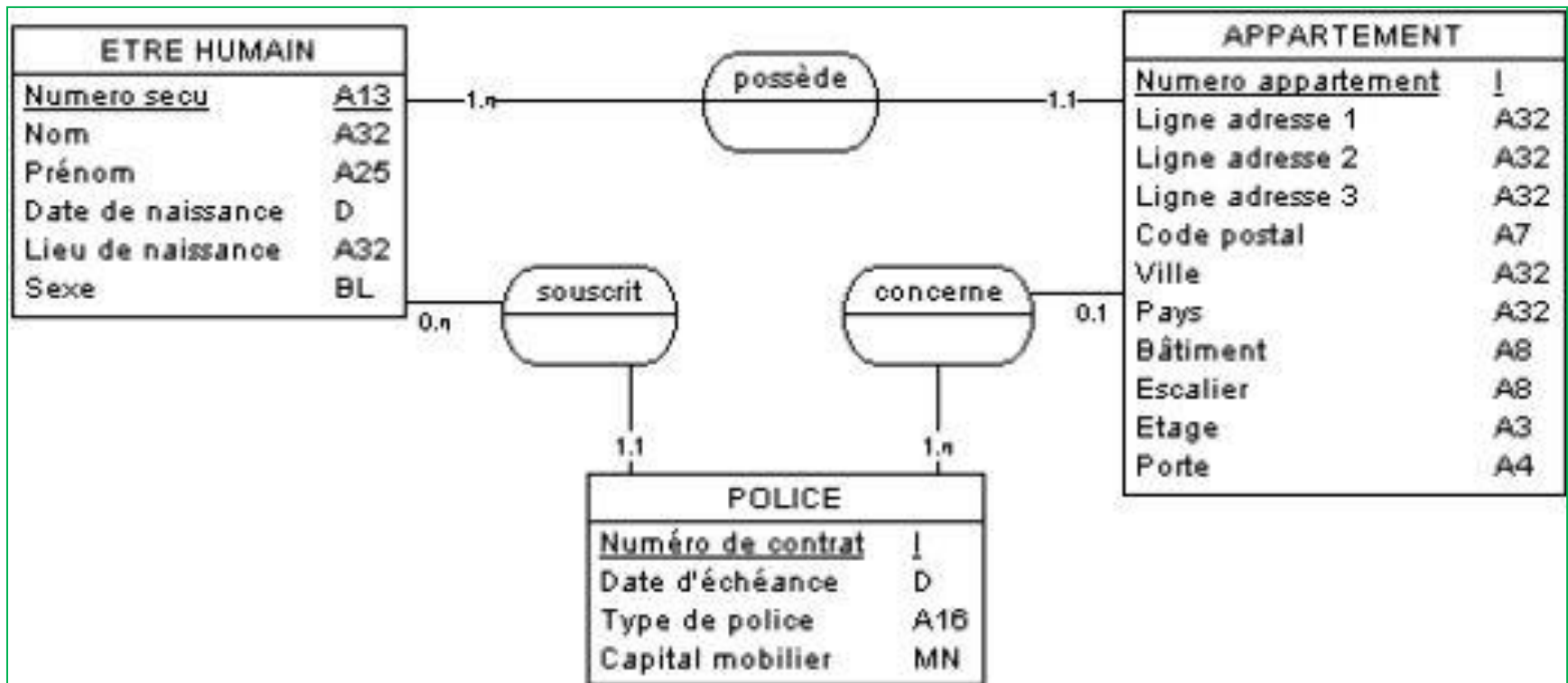


- Mais dans ce cas il y a de **grandes difficultés** pour exprimer les cardinalités.
- On aura tout intérêt à essayer de transformer le schéma de manière à n'obtenir que des associations binaires.



### 3. Topologie des associations

#### Exemple: Solution



### 3. Topologie des associations

#### Attributs d'association:

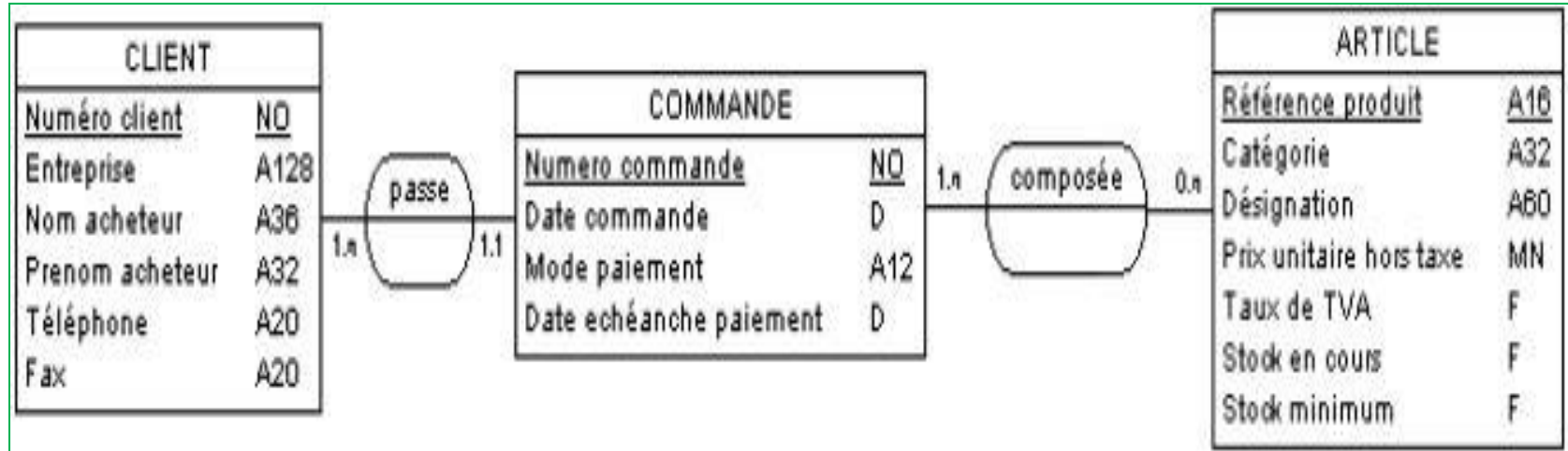
Il arrive parfois que l'on soit obligé de **munir d'attributs des associations**.

#### Exemple:

Considérons que nous voulons **modéliser** les **relations** existant entre les entités "**client**", "commande" et "**article**" :

### 3. Topologie des associations

#### Attributs d'association:



Mais comment dans ce schéma introduire l'attribut "**quantité**" et plus encore l'attribut "**réduction**" dont on voudrait qu'il puisse s'appliquer à chacun des articles d'une commande de manière différente ?

### 3. Topologie des associations

#### Attributs d'association:

En effet si l'on introduit l'attribut quantité à l'entité **COMMANDE**, chaque ligne de la commande se verra dotée de la même quantité...

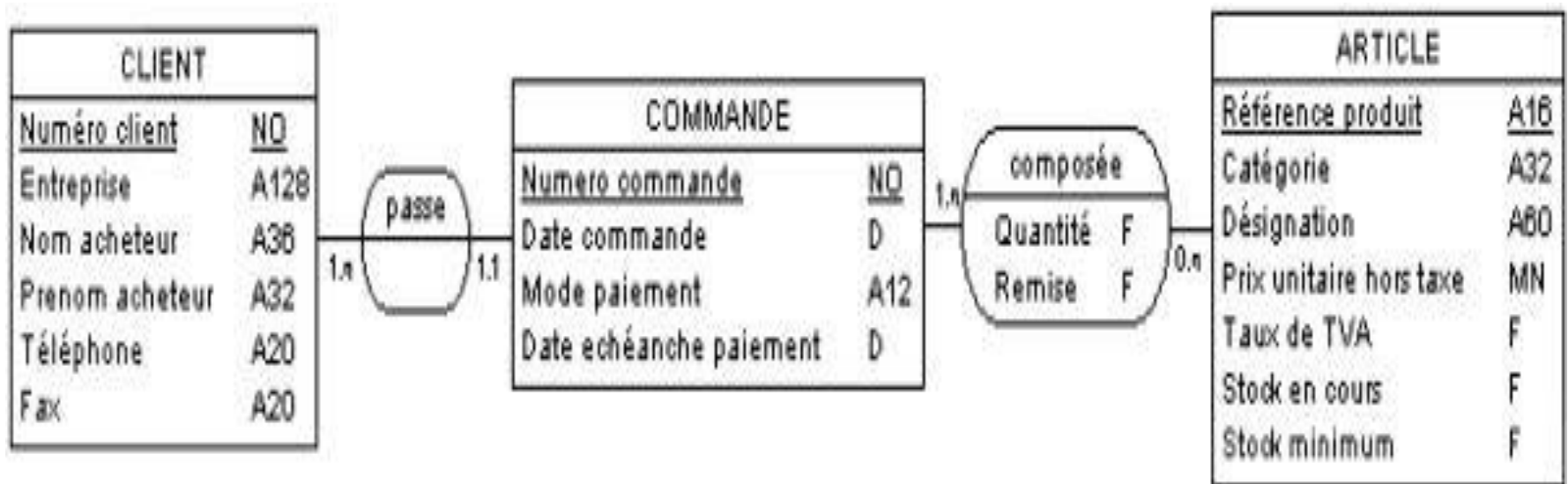
D'autre part si l'on introduit l'attribut quantité à l'entité **ARTICLE** alors chacun des article se verra doté de la même quantité quelque soit la commande...

### 3. Topologie des associations

#### Attributs d'association:

#### Solution:

est de pourvoir l'association "composée" des attributs "quantité" et "réduction" :



## 4 Passage du schéma entité-association (**MCD**) à la construction des tables : **Modèle Logique des Données (MLD)**

Ce que nous venons de voir concerne **l'analyse conceptuelle des données**, c'est à dire un niveau d'analyse qui s'affranchi de toutes les contraintes de la base de données sur lequel va reposer l'application.

Une fois décrit sous forme graphique, ce modèle est couramment appelé **MCD** pour "**Modèle Conceptuel des Données**".

Dès lors, tout MCD peut être transformé en un **MLD** ("Modèle Logique des Données") c'est à dire un modèle directement **exploitable par la base de données** que vous voulez utiliser...

## **4 Passage du schéma entité-association (MCD) à la construction des tables : Modèle Logique des Données (MLD)**

**But du MLD:**

Tout l'intérêt de cet outil d'analyse est de permettre de modéliser plus aisément les relations existant entre les entités et d'automatiser le passage du schéma muni d'attributs aux tables de la base de données pourvues de leurs champs.

Voici maintenant les règles de base nécessaire à une bonne **automatisation du passage du MCD au MLD** :

## 4.1 Transformation des entités

**Règle n°1 :** toute entité doit être représentée par une table.

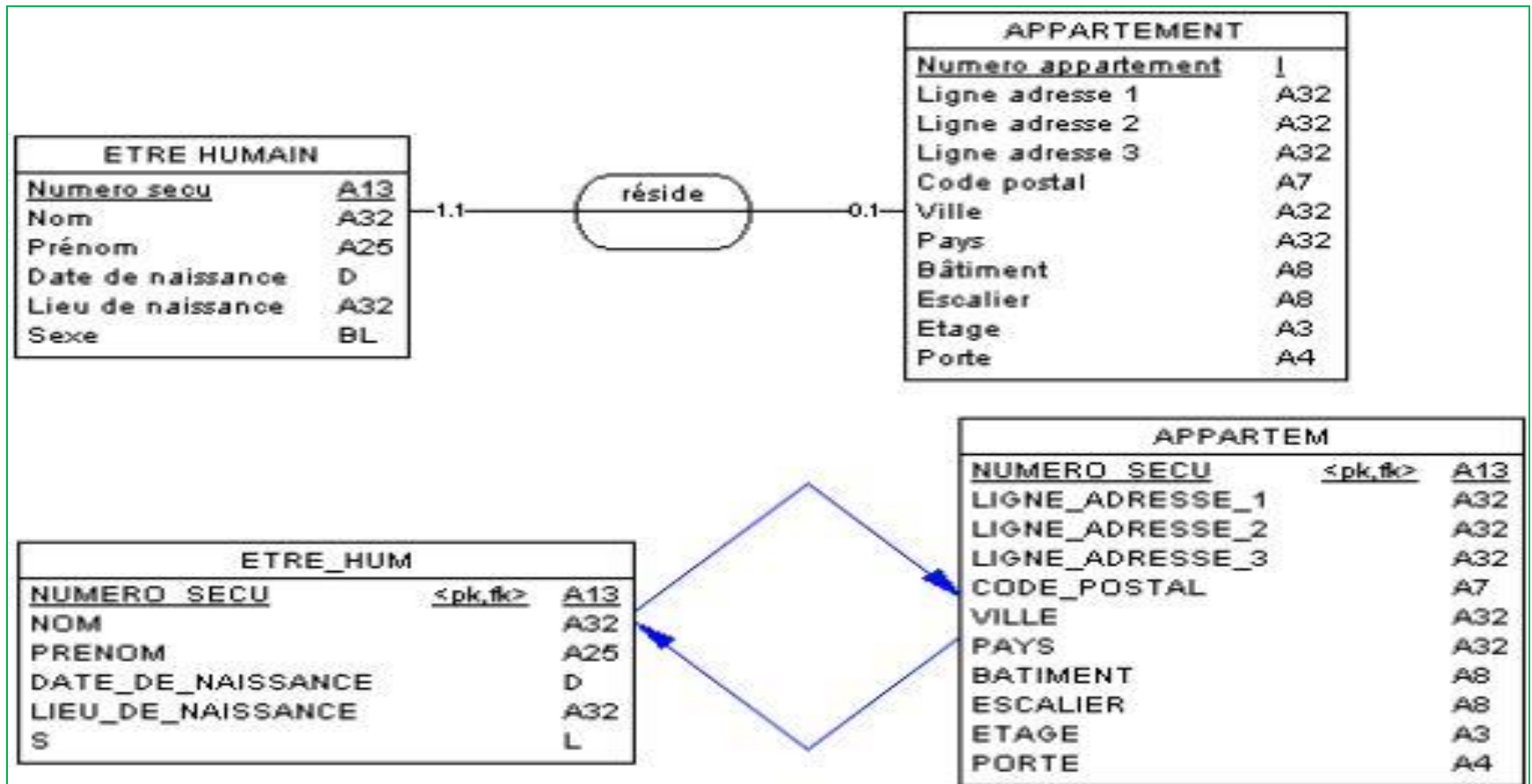
### 4.1.1 Relations de type 1:1

**Règle n°2 :** Dans le cas d'entités reliées par des associations de type 1:1, les tables doivent avoir la même clef.



# 4.1 Transformation des entités

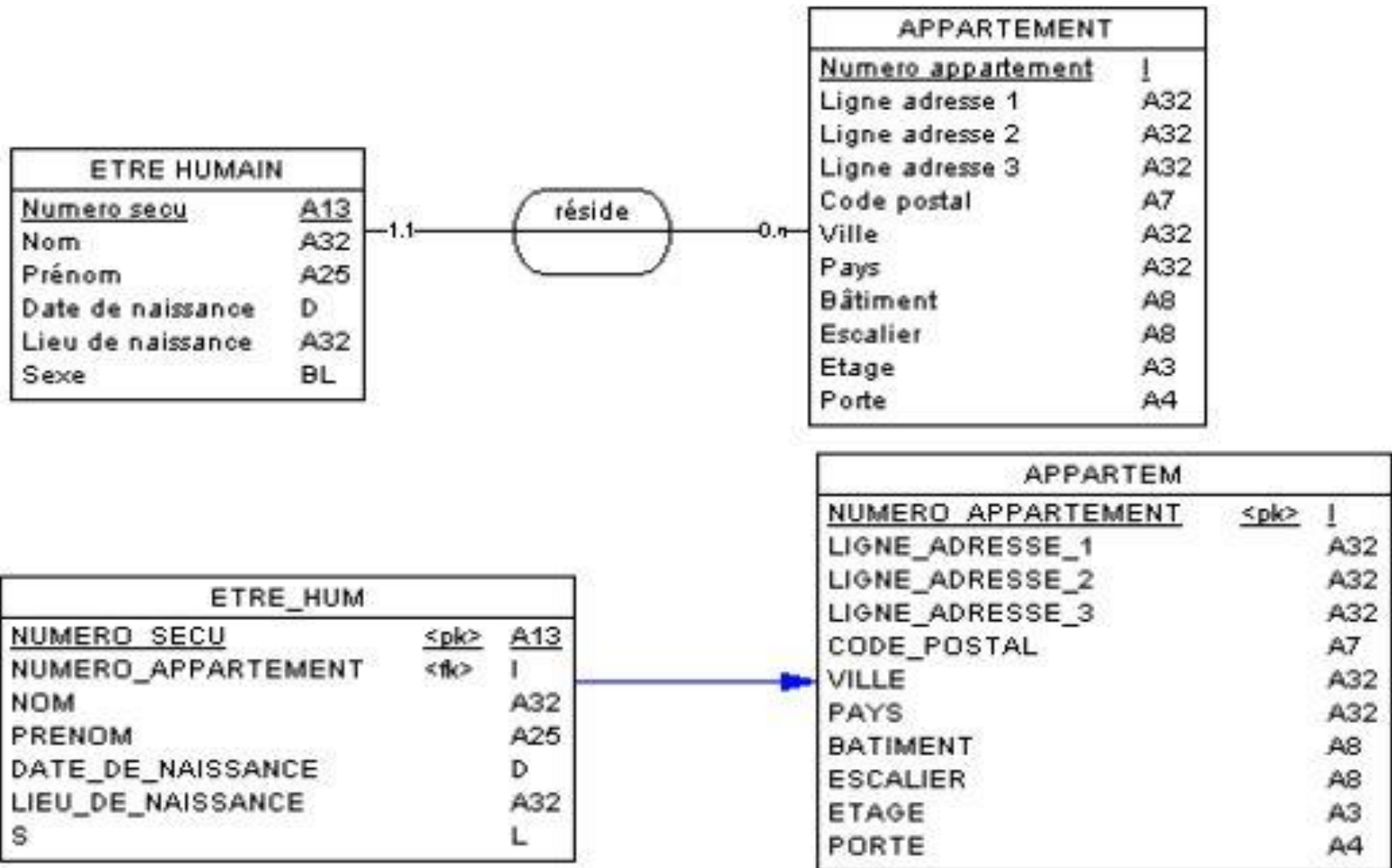
## Exemple:



## 4.1.2 Relations de type 1:n

**Règle n°3 :** Dans le cas d'entités reliées par des associations de type 1:n, chaque table possède sa propre clef, mais la clef de l'entité côté 0,n (ou 1,n) migre vers la table côté 0,1 (ou 1,1) et devient une clef étrangère (index secondaire).

# Exemple:

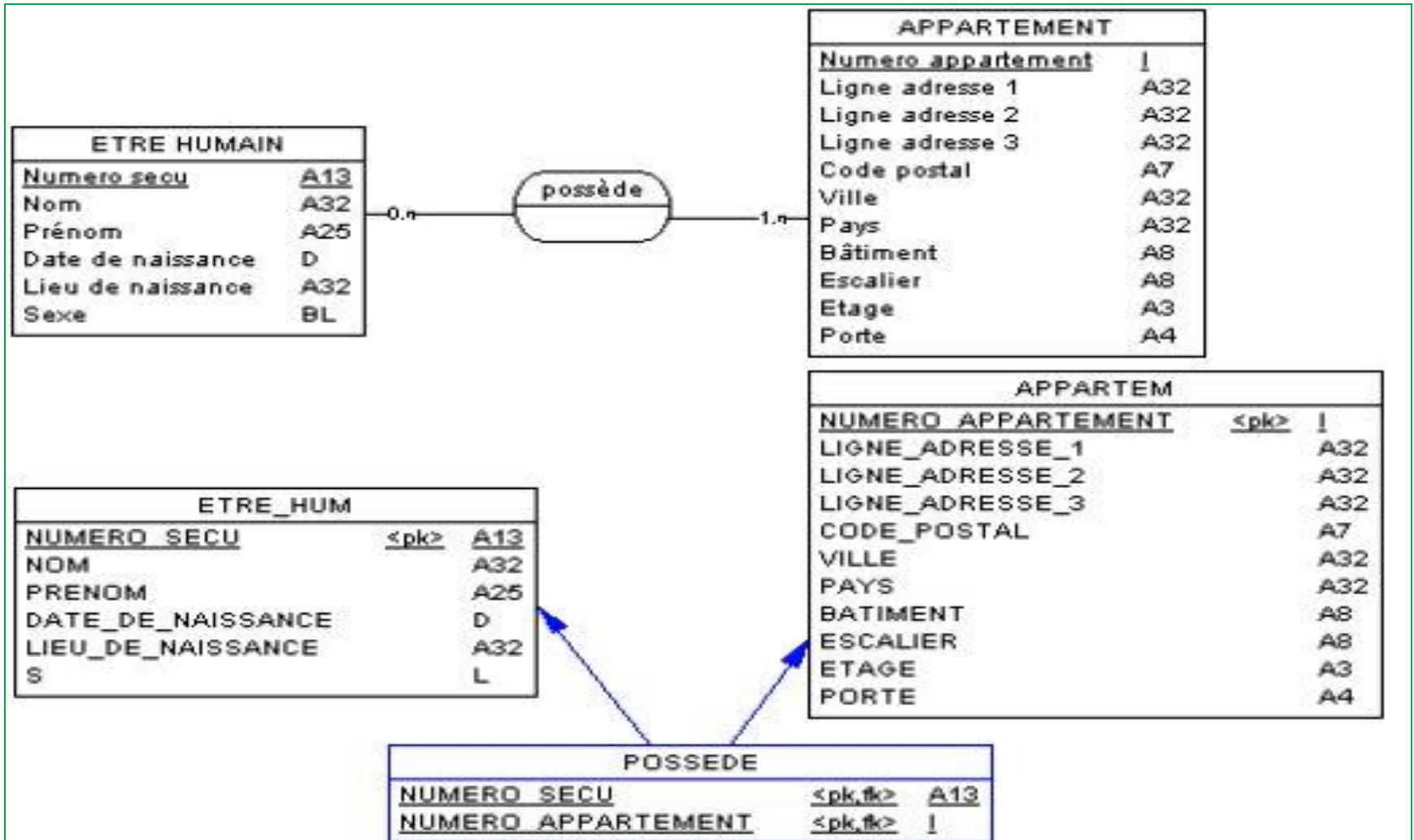


## 4.1.3 Relations de type n:m

### Règle n°4 :

Dans le cas d'entités reliées par des associations de type n:m, une table intermédiaire dite table de jointure, doit être créée, et doit posséder comme clef primaire une conjonction des clefs primaires des deux tables pour lesquelles elle sert de jointure.

# Exemple :

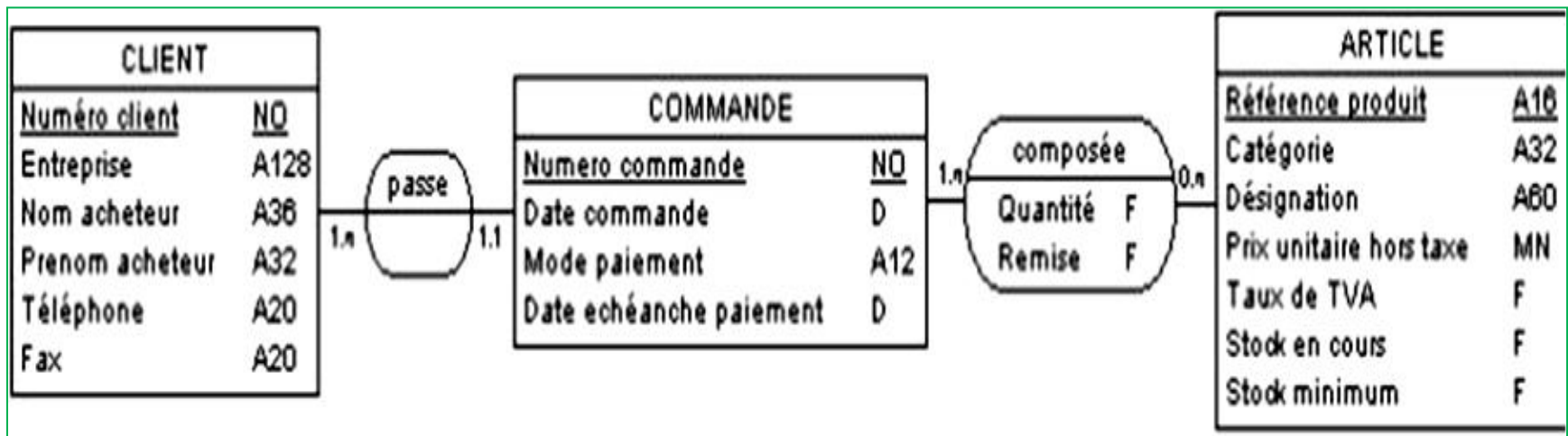


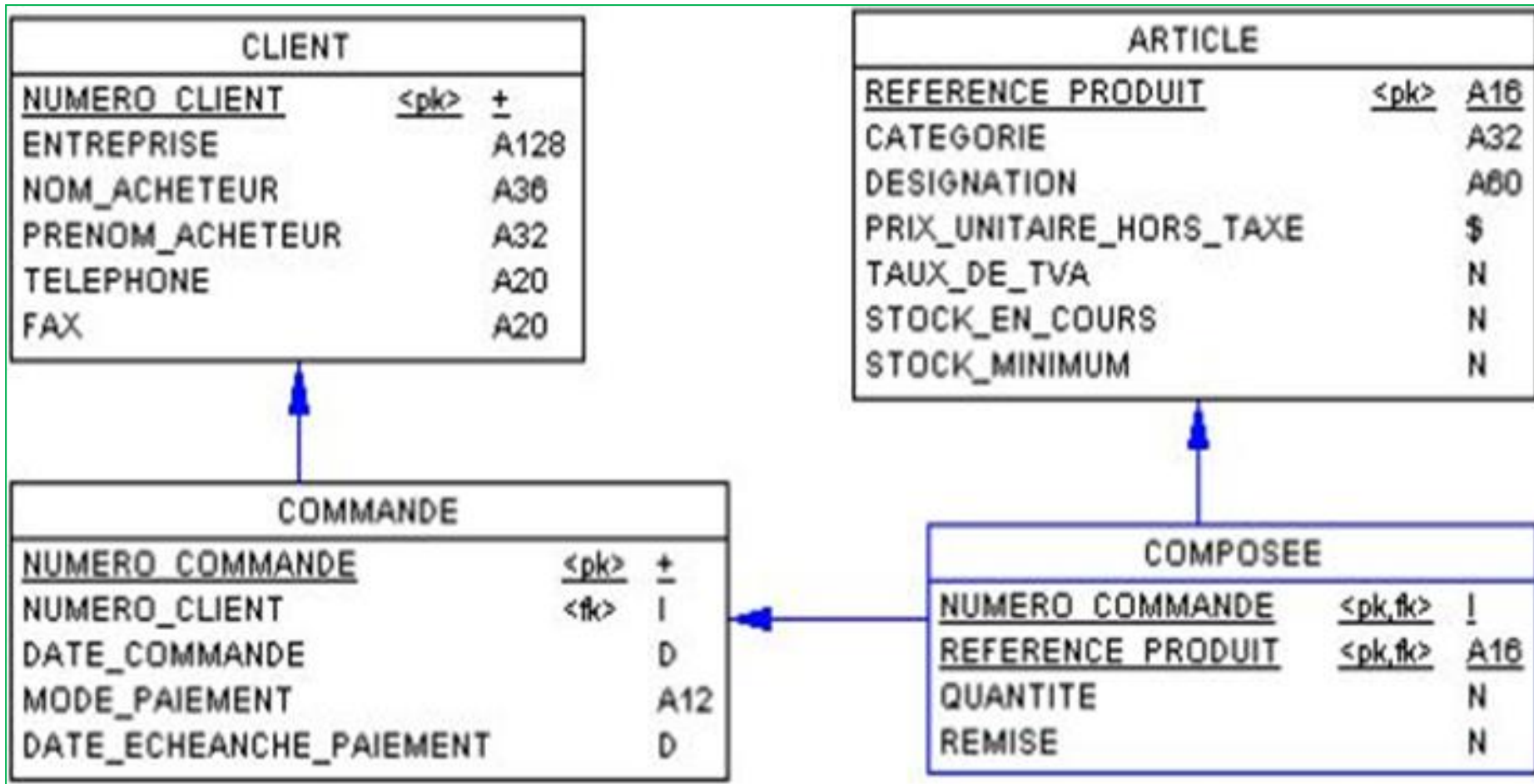
## 4.2 Ou placer les attributs d'association ?

- **Règle n°5 : Cas des associations pourvues d'au moins un attribut :**
  - si le type de relation est **n:m**, alors les attributs de l'association deviennent des attributs de la table de jointure.
  - si le type de relation est **1:n**, il convient de faire glisser les attributs vers l'entités pourvue des cardinalités **1:1**.
  - si le type de relation est **1:1**, il convient de faire glisser les attributs vers l'une ou l'autre des entités.

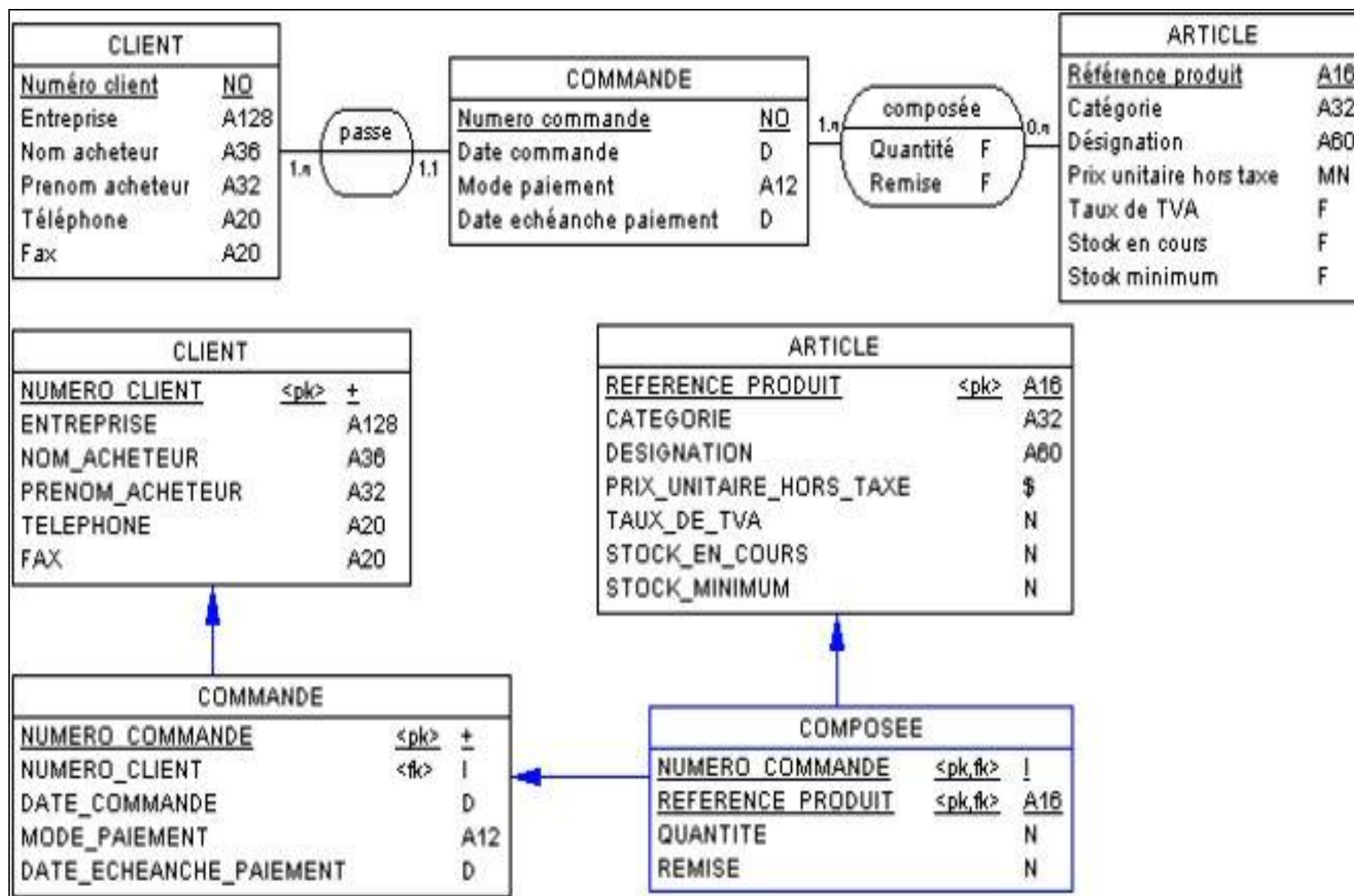
# Exemple :

Pour synthétiser toutes ces règles, voici un exemple de modélisation d'une application. En l'occurrence il s'agit d'un **service commercial** désirant **modéliser les commandes de ses clients**.









# **MPD (Modèle Physique de Données)**

Etablir le MPD (Modèle Physique de Données) correspondant?

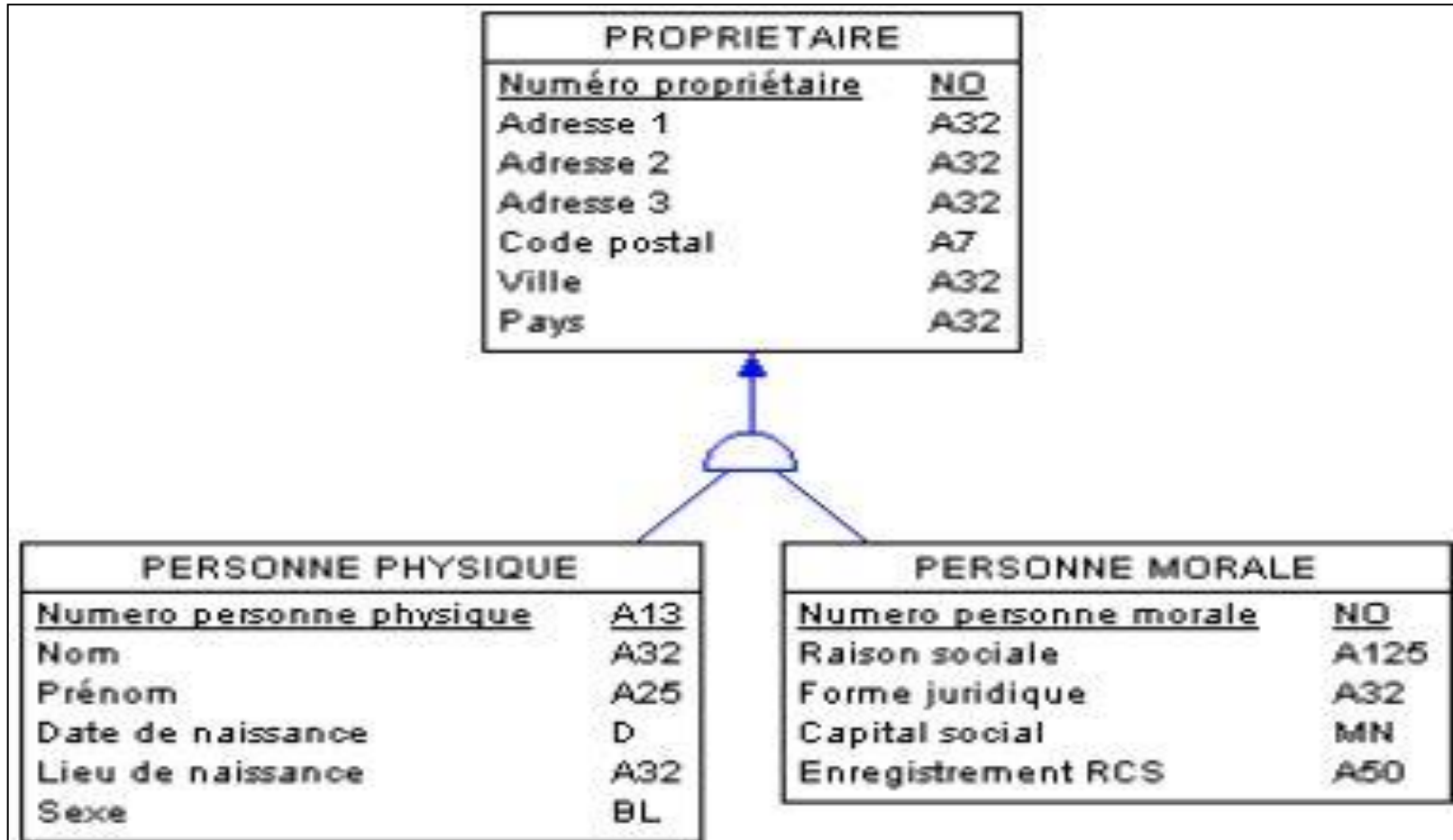
## 5. Conseils divers

- 5.1 Généralisation (héritage)

- Dans le schéma ci-après, les entités "Personne physique" (des êtres humains) et "Personne morales" (des sociétés, associations, collectivités, organisations...) sont généralisées dans l'entité "Propriétaires".

On dit aussi que l'entité "**Propriétaire**" est une **entité mère** et que les entités "**Personne morale**" et "**Personne physique**" sont des **entités filles**, car il y a une notion d'héritage...

# Exemple :



## Exemple

Une entité "**Etre humain**" est une généralisation pour toute entité faisant appel à une personne, comme les entités:

**"Etudiant", "Client", "Artiste", "Patient" ...**

On les appelle aussi "**entités-génériques**".

## Remarque

Certains ateliers de modélisation représentant les données sous la forme d'entités « encapsulés ».

# Exemple :

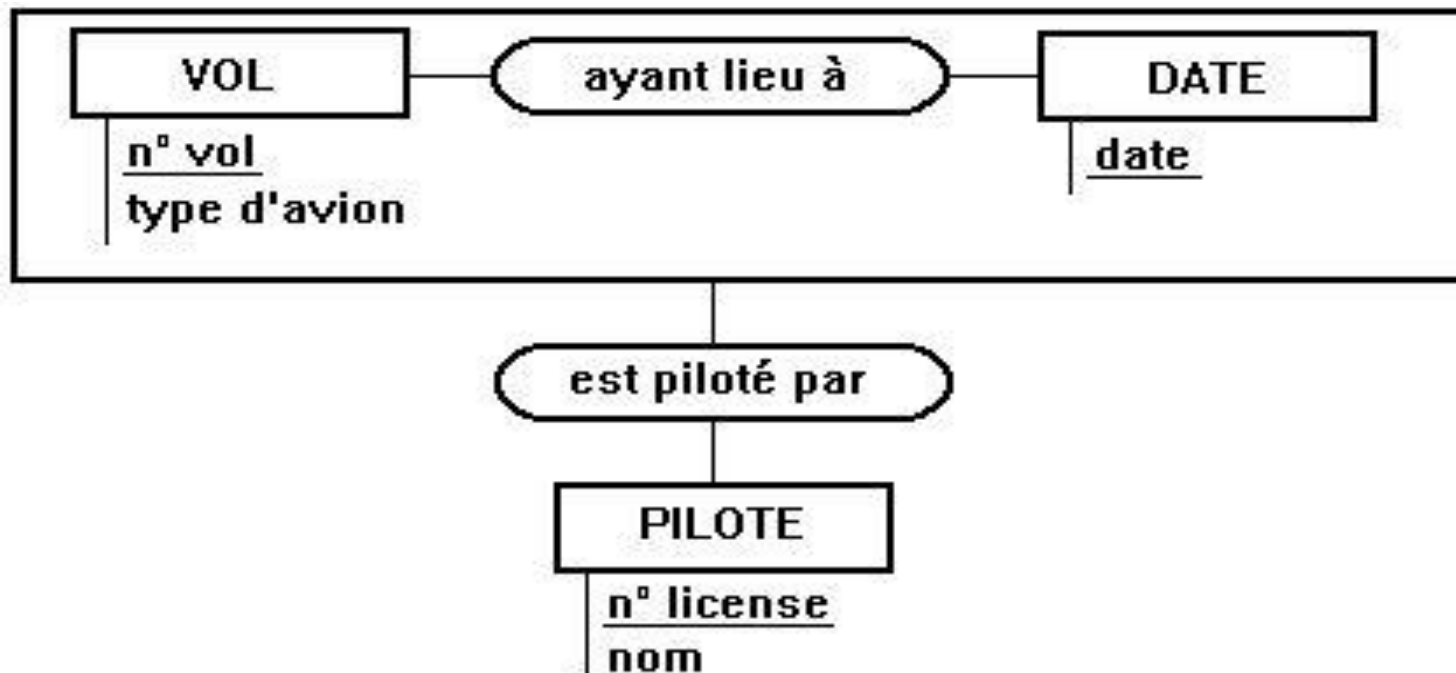
VEHICULE									
<table border="1"><thead><tr><th>VOITURE</th></tr></thead><tbody><tr><td><u>No voit.</u></td></tr><tr><td>couleur</td></tr><tr><td>nb. passagers</td></tr></tbody></table>	VOITURE	<u>No voit.</u>	couleur	nb. passagers	<table border="1"><thead><tr><th>CAMION</th></tr></thead><tbody><tr><td><u>No cam.</u></td></tr><tr><td>type atelage</td></tr><tr><td>tonnage</td></tr></tbody></table>	CAMION	<u>No cam.</u>	type atelage	tonnage
VOITURE									
<u>No voit.</u>									
couleur									
nb. passagers									
CAMION									
<u>No cam.</u>									
type atelage									
tonnage									
<table border="1"><tbody><tr><td><u>No véhicule</u></td></tr><tr><td>Immatriculation</td></tr><tr><td>Marque</td></tr><tr><td>Type</td></tr><tr><td>Cylindrée</td></tr><tr><td>Puissance</td></tr><tr><td>Carburant</td></tr></tbody></table>		<u>No véhicule</u>	Immatriculation	Marque	Type	Cylindrée	Puissance	Carburant	
<u>No véhicule</u>									
Immatriculation									
Marque									
Type									
Cylindrée									
Puissance									
Carburant									

## 5.2 Personnalisation

**Une personnalisation est un regroupement dans une super entité de plusieurs entités munies d'une ou de plusieurs associations.**

# Exemple:

Une compagnie d'aviation proposant des vols peut modéliser le planning des pilotes par le schéma suivant :





## 5.3 Regroupement d'entités

Comme toute technique, le schéma entité-association possède des limites et des contraintes que seuls l'expérience et le bon sens peuvent permettre d'éliminer.

Il arrive parfois que certaines entités apparaissent comme redondantes.

Dans ce cas, et pour **gagner de la place en matière de stockage de l'information**, il convient de **regrouper ces entités dans une seule et même table** du SGBDR en ajoutant un champ supplémentaire à cette table de manière à permettre de distinguer les entités du schéma théorique.

# Exemple

Si l'on désire modéliser une gestion de **compact-disc**, on peut créer une entité "**Compositeur**" et une entité "**Interprète**".

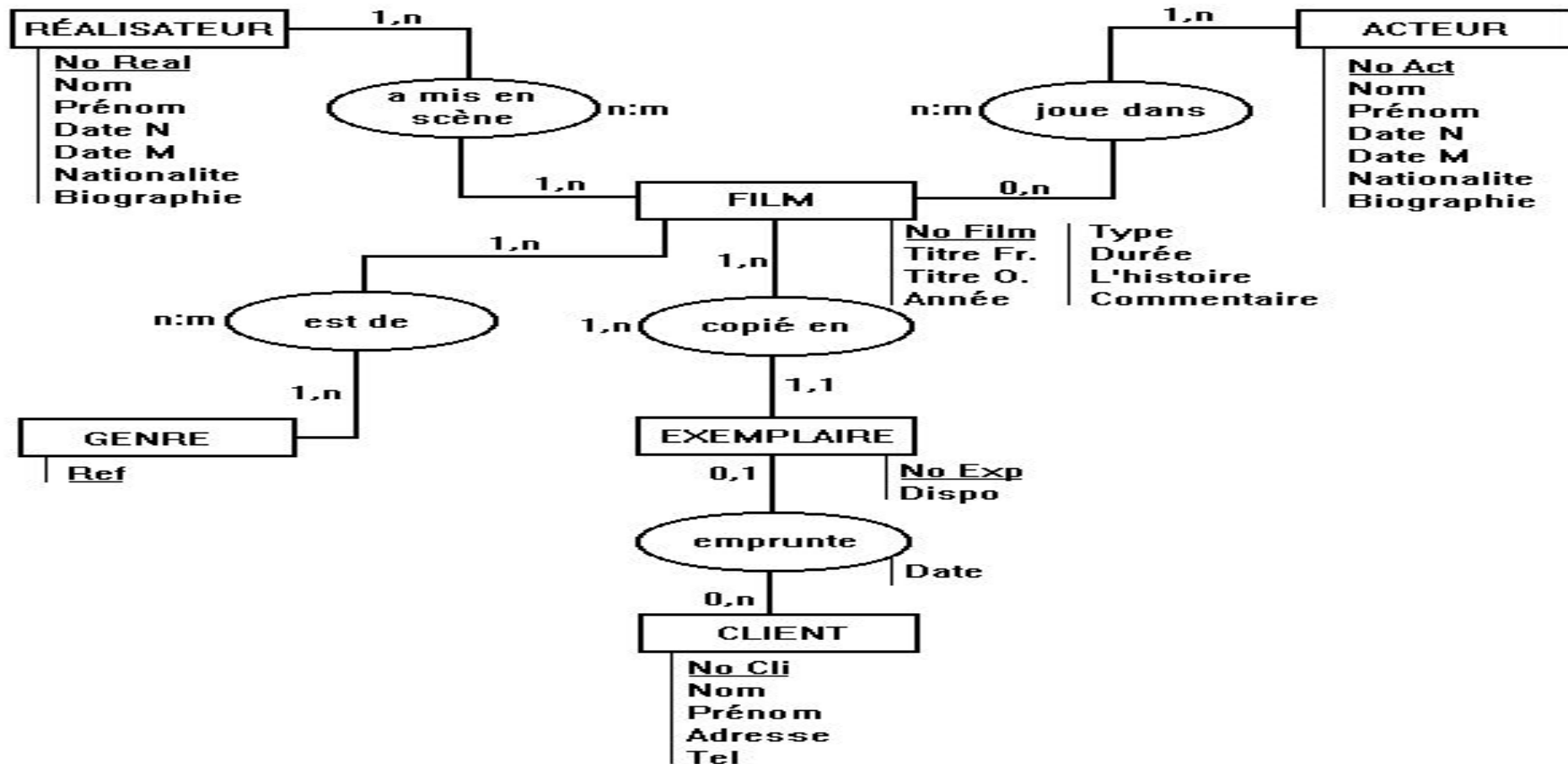
Mais on constate qu'une grande majorité de compositeurs sont leurs propres interprètes, ce qui signifie qu'une même personne peut se trouver présente dans les deux entités.

# Solution

Pour résoudre ce problème il suffit de **construire une seule table pour les deux entités** (par exemple une table "**MUSICIEN**") et d'y ajouter un champ permettant de distinguer le type de "musicien" : compositeur ou interprète ou les deux.

# 6. Exemples de MCD

## 6.1 Agence de location de films vidéo



## Remarques:

**1. Lors de la réalisation de la base de données, les entités RÉALISATEUR et ACTEUR peuvent être regroupées en une seule table car il y a un nombre non négligeable de réalisateurs qui sont acteurs, et vice-versa.**

**Dans ce cas, un champ d'un seul caractère permettra de faire la différence entre un réalisateur pur, un acteur pur et un acteur réalisateur.**

2. Notez aussi les cardinalités entre les entités **ACTEUR** et **FILM**, en effet, un film d'animation ne possède aucun acteur.
3. Dans l'entité **EXEMPLAIRE** figure un attribut "dispo" permettant de savoir si l'exemplaire n°X d'un film est disponible ou en cours d'emprunt.

### Premier exercice :

Un agriculteur, Monsieur Bousquet, fait de la vente directe de ses produits ou animaux qu'il élève. Il vend des lapins, des poules, des dindes, des veaux, des cochons. Selon la saison il vend aussi des légumes (choux, pommes de terre, carottes...) et des fruits (fraises, poires, pommes...). Il ne fait que de la vente directe. Suite à votre discussion, il ressort les informations suivantes.

À l'heure actuelle, les ventes sont inscrites sur trois cahiers distincts :

- Un pour les animaux.
- Un pour les fruits.
- Un pour les légumes.

Tout est vendu au kilo, les animaux sont pesés vivants avant d'être vendus.

Il souhaiterait un logiciel simple pour saisir les ventes journalières et pouvoir éditer un récapitulatif mensuel par type de vente (animaux, légumes et fruits) et par produit (poulets, lapins, poireaux, poires...) pour sa comptabilité.

### Travail à faire:

1. Créer le Modèle Conceptuel des Données.
2. Concevoir le Modèle Logique des Données.
3. Finir par le Modèle Physique des Données.

### Deuxième exercice :

Voici un modèle relationnel décrivant une nomenclature de conception d'un meuble. Le meuble est un ensemble composé de sousensembles et de composants divers. Un sous-ensemble est élaboré grâce à un assemblage de composants.

À partir de ce modèle relationnel, il vous est demandé de procéder à du reverse engineering ou en français de la rétroingénierie. C'est-à-dire de remonter jusqu'au modèle conceptuel en passant par le modèle logique des données.

### Modèle relationnel:

Ensembles(CodeEnsemble, Désignation)

SousEnsembles(CodeSousEnsemble, Désignation, Longueur, Largeur, Hauteur, Prix\_Unitaire)

Composants(CodeComposant, Désignation, Prix\_Unitaire)

LienEnsSE(#CodeEnsemble, #CodeSousEnsemble, Qté)

LienEnsComposant(#CodeEnsemble, #CodeComposant, Qté)

LienSEComposant(#CodeSousEnsemble, #CodeComposant, Qté)



### Troisième exercice :

La nouvelle loi sur l'autoentrepreneuriat vient d'être promulguée et vous vous dites que c'est peut-être le moment de vous mettre à votre compte.

Comme toutes les personnes de votre village font appel à vos services lorsqu'ils ont un problème informatique, vous êtes sûr que votre affaire va fonctionner.

Pour démarrer il vous faut un petit logiciel vous permettant de saisir vos interventions pour faciliter la tenue de votre comptabilité.

Ce logiciel permettra la saisie des coordonnées des clients et le matériel sur lequel vous êtes intervenu.

Vous décidez d'appliquer un prix horaire différent selon le type d'intervention (certaines réparations ou manipulation complexes doivent être facturées plus cher).

Pour certaines pannes vous vendrez le composant neuf. Le logiciel devra donc intégrer la vente de matériel inhérente à la réparation.

### Travail à faire:

1. Concevoir le dictionnaire des données simplifié.
2. Concevoir le Modèle Conceptuel des Données.
3. Concevoir le Modèle Logique des Données.

### Quatrième exercice :

**Vous êtes missionné par un de vos amis qui exerce la profession d'agent immobilier pour lui réaliser un petit programme.**

**Il désire un logiciel dans lequel il peut inscrire son fichier des maisons, des propriétaires et des locataires.**

**Règles de gestion:**

**Une maison appartient à une ou plusieurs personnes.**

**Une personne peut être propriétaire d'une maison et en louer une autre.**

### Travail à faire:

- 1. Créer le Modèle Conceptuel des Données.**
- 2. Concevoir le Modèle Logique des Données.**
- 3. Finir par le Modèle Physique des Données.**

## Exercices

### Cinquième exercice :

Vous êtes missionné par un de vos amis qui exerce la profession d'agent immobilier pour lui réaliser un petit programme.

ASSUR'AUTO, comme son nom l'indique, est une petite société d'assurance spécialisée dans les contrats d'assurance automobile. Malgré son envergure restreinte (elle dispose tout de même de plusieurs agences et plusieurs employés sur le territoire) elle assure aussi bien les véhicules de tourisme que les véhicules utilitaires.

Pour assurer un véhicule, son propriétaire, dont on enregistre le nom, le prénom, l'adresse et les coordonnées (téléphone, fax éventuel, e-mail...), doit fournir au conseiller de l'agence la carte grise du véhicule afin que l'on enregistre son type, sa marque, son numéro d'immatriculation, sa date de mise en circulation et sa puissance fiscale.

S'il s'agit d'un véhicule de tourisme, on enregistre aussi le nombre de portes et de passagers autorisés, tandis que s'il s'agit d'un véhicule utilitaire on enregistre le poids à vide, le poids autorisé en charge, la longueur, la largeur.

Chaque contrat, établi à une certaine date, est référencé par un numéro de contrat et est d'une certaine catégorie : tous risques, au « tiers »...

Le contrat est attaché à la personne, pas au véhicule: lorsqu'il y a changement de véhicule le propriétaire conserve le bonus ou le malus attaché à ce contrat qui est alors reporté sur le nouveau véhicule.

### Travail à faire:

1. Créer le Modèle Conceptuel des Données.
2. Concevoir le Modèle Logique des Données.

## Exercices

### Cinquième exercice :

Vous êtes missionné par un de vos amis qui exerce la profession d'agent immobilier pour lui réaliser un petit programme.

ASSUR'AUTO, comme son nom l'indique, est une petite société d'assurance spécialisée dans les contrats d'assurance automobile. Malgré son envergure restreinte (elle dispose tout de même de plusieurs agences et plusieurs employés sur le territoire) elle assure aussi bien les véhicules de tourisme que les véhicules utilitaires.

Pour assurer un véhicule, son propriétaire, dont on enregistre le nom, le prénom, l'adresse et les coordonnées (téléphone, fax éventuel, e-mail...), doit fournir au conseiller de l'agence la carte grise du véhicule afin que l'on enregistre son type, sa marque, son numéro d'immatriculation, sa date de mise en circulation et sa puissance fiscale.

S'il s'agit d'un véhicule de tourisme, on enregistre aussi le nombre de portes et de passagers autorisés, tandis que s'il s'agit d'un véhicule utilitaire on enregistre le poids à vide, le poids autorisé en charge, la longueur, la largeur.

Chaque contrat, établi à une certaine date, est référencé par un numéro de contrat et est d'une certaine catégorie : tous risques, au « tiers »...

Le contrat est attaché à la personne, pas au véhicule: lorsqu'il y a changement de véhicule le propriétaire conserve le bonus ou le malus attaché à ce contrat qui est alors reporté sur le nouveau véhicule.

### Travail à faire:

1. Créer le Modèle Conceptuel des Données.
2. Concevoir le Modèle Logique des Données.
3. Finir par le Modèle Physique des Données.

## Exercices

### Sizième exercice :

L'entreprise XProd fabrique et commercialise divers produits. Ils sont identifiés par une référence propre à XProd et on enregistre une désignation (libellé court), un descriptif (libellé long) et un prix de vente catalogue unitaire hors taxes.

Dans la base de données elle gère deux types de produits :

l les produits qu'elle fabrique pour lesquels on enregistre le nombre moyen d'heures de main d'œuvre nécessaire à leur fabrication ;

l les produits dits « approvisionnés » parce qu'elle ne les fabrique pas : ils sont achetés à un ou plusieurs fournisseurs à un prix d'achat unitaire moyen.

Pour ne pas dépendre d'un fournisseur, enregistré par ses raison sociale, adresse, etc., pour chaque produit approvisionné l'entreprise a établi une liste de fournisseurs capables de livrer ce produit. Bien entendu pour un même produit chaque fournisseur peut avoir sa propre référence et un prix différent.

Lorsque XProd passe une commande à une certaine date à un fournisseur, elle essaie de grouper plusieurs lignes de commande : une par produit dans une certaine quantité avec sa date de livraison prévue, pour réduire les frais de livraison de la commande et essayer de négocier un prix d'achat unitaire inférieur au prix catalogue du fournisseur.

### Travail à faire:

1. Créer le Modèle Conceptuel des Données.
2. Concevoir le Modèle Logique des Données.

# BIBLIOGRAPHIE

- La méthode MERISE, principes et outils (Tome 1 & 2) - TARDIEU, ROCHFELD, COLLETTI - Les éditions d'organisation 1986
- Conception de bases de données : du schéma conceptuel au schéma physique - GALACSI - Dunod 1989
- Modélisation dans la conception des systèmes d'information - ACSIOME - Masson 1990
- Apprendre et pratiquer MERISE - J. GABAY - Masson 1993
- Maîtriser les bases de données — Georges GARDARIN — Eyrolles 1993
- Concepts fondamentaux de l'informatique — Alfred AHO, Jeffrey ULLMAN — Dunod 1993
- Bases de Données et Modèles de Calcul - Outils et Méthodes pour l'Utilisateur - Jean-Luc HAINAUT - InterEditions 1994
- MERISE, vers une modélisation orientée objet — José MOREJON — Les Editions d'Organisation 1994
- AMC\*Designor, mise en œuvre de MERISE — Gilles GUEDJ — Eyrolles 1996
- Introduction aux bases de données, 6e édition— Chris J. DATE — Thomson International Publishing 1998
- De UML à SQL - Christian SOUTOU - Eyrolles 2002