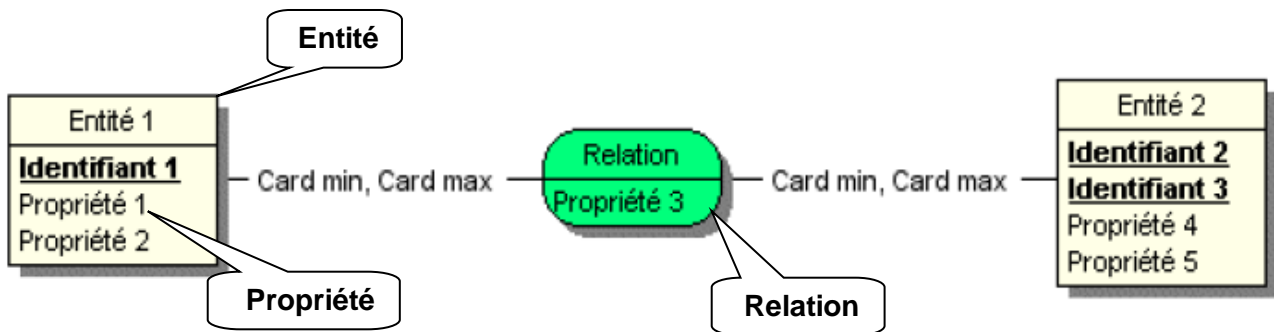
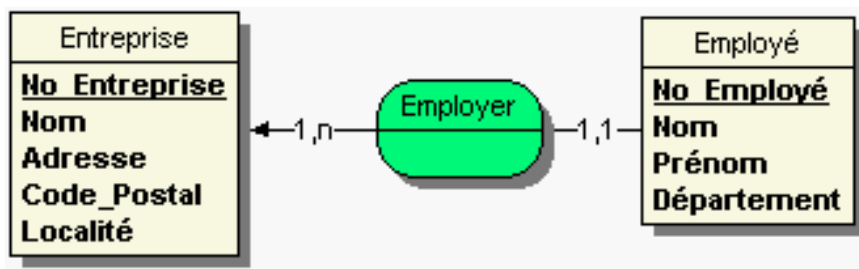


### 4.1 Définition

En se basant sur un document d'analyse, le modèle conceptuel des données (MCD) fait référence à tous les éléments du système d'information et aux relations entre ces éléments. Le formalisme utilisé dans ce modèle est connu sous le nom de "Schéma Entité-Relation". Ce formalisme se base sur trois concepts principaux, les **entités**, les **relations** et les **propriétés**.



Voici par exemple un MCD qui représente une entreprise avec ses employés.

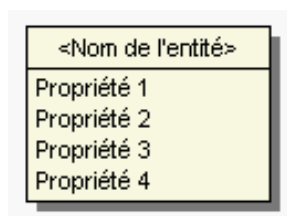


### 4.2 Concepts de base

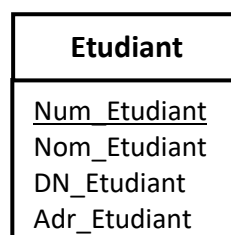
**a. Entité (individu ou objet).** Une entité permet de modéliser un ensemble d'objets concrets ou abstraits de même nature. Une entité est caractérisée par son nom et ses propriétés.

Exemple : Employé, entreprise, Etudiant, Cours, etc.

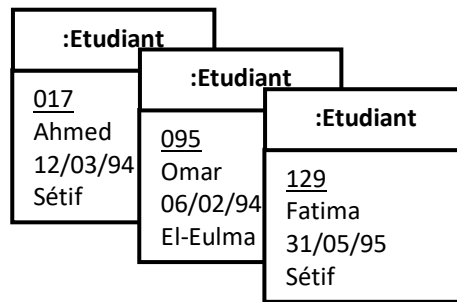
Représentation graphique :



Voici un exemple de l'entité *Etudiant* :



Chacun des étudiants suivants représente une *occurrence* de l'entité *Etudiant*.



**b. Propriété.** Une propriété est une donnée élémentaire d'une entité. Elle est unique dans un MCD; et ne peut pas être rattachée à plusieurs entités différentes. Une propriété est caractérisée par :

- son type (Numérique, Alphabétique, etc.),
- sa longueur (taille en caractères), et
- sa nature (élémentaire, calculée ou concaténée).

Exemple : Nom de l'étudiant, Adresse de l'étudiant, etc.

A l'intérieur de chaque occurrence, chaque propriété prend une valeur, qui est dans la plupart des cas une **valeur numérique**, une valeur sous forme de **texte** ou encore une **date**.

Exemple : La propriété **Nom\_Etud** de l'entité *Etudiant* prend par exemple les valeurs "Ahmed", "Omar" et "Fatima" dans les 3 occurrences de l'exemple précédent.

A l'intérieur de chaque occurrence, chaque propriété ne prend qu'une seule valeur au maximum.

Exemple : L'étudiant 095 ne peut pas avoir 2 adresses.

### c. Identifiant

Afin de pouvoir distinguer les différentes occurrences d'une même entité, l'entité doit être dotée d'un identifiant. Chaque occurrence de l'entité doit avoir une valeur différente pour l'identifiant. Le choix de l'identifiant repose sur 3 possibilités:

1. Une propriété **naturelle**

Exemple: Le nom d'un pays pour une entité *Pays*

2. Une propriété **artificielle** qui est inventée par le développeur du MCD

Exemple: Le numéro d'un étudiant (Num\_etud)

3. Une propriété **concaténée** (composée d'autres propriétés naturelles)

Exemple: Le nom et la localité pour une entité *Entreprise* (*NomEntr\_NomVille*)

Dans le formalisme utilisé, l'identifiant d'une entité est une propriété soulignée en tête de l'ensemble des propriétés.

**d. Relation.** Une relation décrit un lien entre deux ou plusieurs entités (*collection* de la relation). Elle possède un nom, généralement un verbe à l'infinitif. L'existence de la relation est conditionnée par celle des entités qui participent à cette relation.

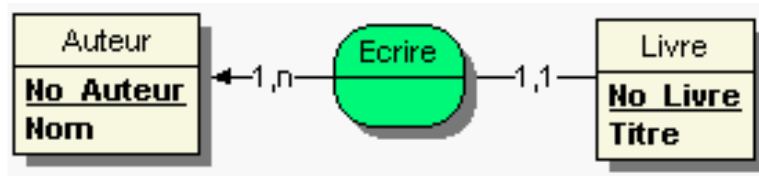
La relation n'a pas d'identifiant propre, mais elle est implicitement identifiée par la concaténation des identifiants des entités qui participent à cette relation.

1. *Dimension d'une relation* : c'est le nombre de pattes de la relation. Le nombre de pattes n'est pas nécessairement identique au nombre d'entités. Il représente plutôt le nombre d'occurrences d'entités qui doivent participer à une occurrence de la relation.

Nous distinguons des relations :

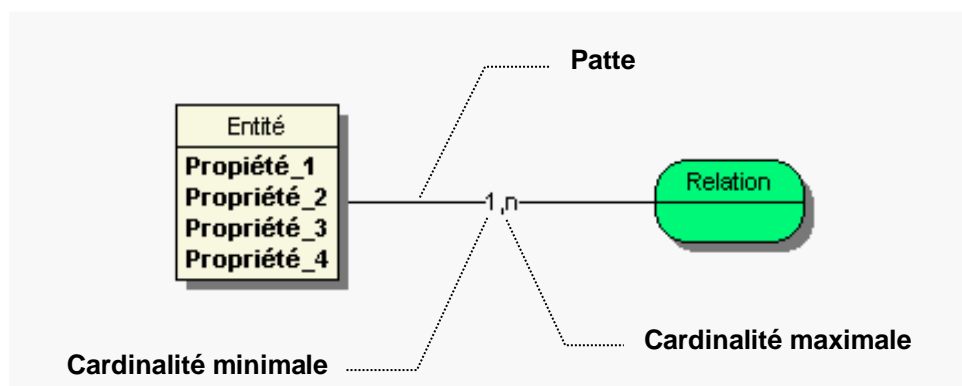
- binaires, qui possèdent 2 pattes;
- ternaires, qui possèdent 3 pattes;
- n-aires, qui possèdent n pattes.

Exemple d'une relation binaire:

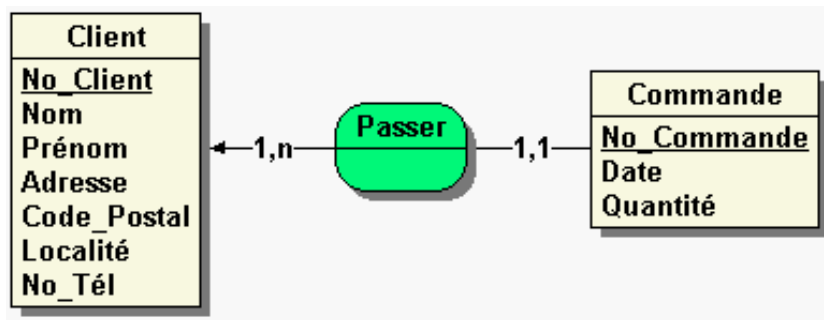


## 2. Cardinalités d'une relation

Une relation est liée à chacune de ses entités par une ou plusieurs pattes. Sur la patte, on indique les cardinalités qui précisent le nombre de participations de chaque occurrence de l'entité à la relation. Le premier nombre indique la cardinalité minimale, le deuxième la cardinalité maximale.



Exemple :



Dans cet exemple, entre l'entité *Client* et la relation *Passer*, nous avons les cardinalités suivantes:

- Cardinalité minimale = 1, *ce qui signifie que chaque client passe au moins une commande.*
- Cardinalité maximale = n, *ce qui signifie que chaque client peut passer plusieurs (n) commandes.*

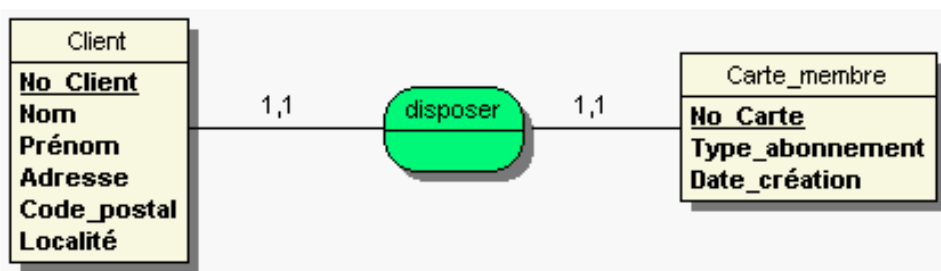
Entre l'entité *Commande* et la relation *Passer*, nous retrouvons les cardinalités suivantes:

- Cardinalité minimale = 1, *chaque commande est passée par au moins un client.*
- Cardinalité maximale = 1, *chaque commande est passée au maximum par un seul client.*

De façon générale, quatre types sont distingués :

- 0,1 : chaque occurrence participe au plus une seule fois à la relation.
- 1,1 : chaque occurrence participe une et une seule fois à la relation.
- 1,n : chaque occurrence participe au moins une fois à la relation.
- 0,n ; aucune précision n'est donnée quant au nombre de participations.

Comme cas particulier, la modélisation suivante par exemple n'est pas correcte:

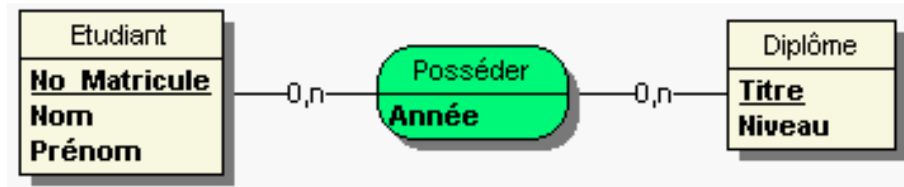


Dans ce cas, il faut réunir les propriétés des deux entités dans une seule.

### 3. Propriétés d'une relation

Une relation peut généralement être porteuse de propriétés.

Exemple:



### f. Règles de gestion

Les règles de gestion précisent les contraintes qui doivent être respectées par le modèle.

Exemples : pour un magasin,

- Une personne habite dans une et une seule maison;
- Plusieurs personnes peuvent habiter dans la même maison;
- Une personne peut être propriétaire de plusieurs maisons;
- Pour une maison, nous avons au moins un propriétaire.

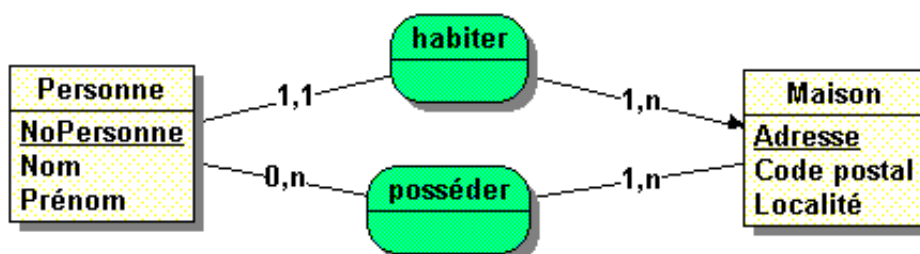
Les règles de gestion expriment les contraintes d'intégrité du modèle (les lois de l'univers réel modélisé dans le SI).

### 4.3 Cas particuliers du MCD

#### a. Plusieurs relations différentes entre deux entités

Il est possible d'avoir plusieurs relations différentes entre deux mêmes entités. Chacune de ces relations possède évidemment une sémantique différente. Le rôle de l'entité diffère alors d'une relation à une autre.

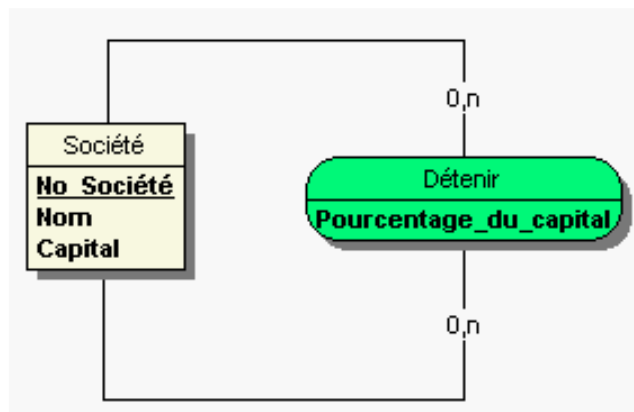
Exemple :



Une personne, peut être l'habitant d'une maison et/ou le propriétaire de la maison. Ici, *habitant* et *propriétaire* sont des rôles joués par l'entité *Personne*.

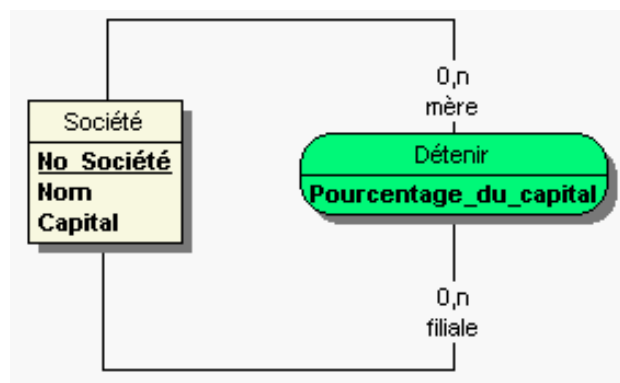
### b. Relation réflexive et rôle d'une patte de relation

Exemple :



**Une relation réflexive, est une relation, dont les deux pattes sont liées à une même entité.** En général, la signification des pattes d'une relation réflexive devrait être clarifiée par l'indication d'un rôle.

Nous avons donc:



## 4.4 Dépendances fonctionnelles

### 4.4.1 Définition

Les propriétés a et b sont reliées par une dépendance fonctionnelle  $a \text{ --df--> } b$ , si la connaissance de la valeur de a détermine une et une seule valeur de b.

Exemple : Num\_Etud  $\text{--df-->}$  Nom\_Etud.

La dépendance fonctionnelle peut porter sur la concaténation de plusieurs propriétés.

Exemple : NumCommande + RéfProduit  $\text{--df-->}$  QtéCommandée.

### 4.4.2 Dépendance fonctionnelle élémentaire

Il y a dépendance fonctionnelle élémentaire entre les propriétés a et b et on note  $a \rightarrow b$ , si  $a \text{ --df--> } b$  et aucune partie de a ne détermine b.

Exemple :  $\text{Num\_Etud} + \text{Nom\_Etud} \text{ -df} \rightarrow \text{Adr\_Etud}$  n'est pas élémentaire puisque la connaissance de  $\text{Num\_Etud}$  suffit pour déterminer  $\text{Adr\_Etud}$ .

Par contre,  $\text{Num\_Etud} \text{ -df} \rightarrow \text{Adr\_Etud}$  est élémentaire et on note  $\text{Num\_Etud} \rightarrow \text{Adr\_Etud}$ .

#### 4.4.3 Dépendance fonctionnelle élémentaire directe

Les propriétés  $a$  et  $b$  sont reliées par une dépendance fonctionnelle élémentaire directe si  $a \rightarrow b$  et s'il n'existe pas une propriété  $c$  telle que  $a \text{ -df} \rightarrow c$  et  $c \text{ -df} \rightarrow b$ . autrement dit, on élimine toute transitivité.

Exemple :

$\text{NumProf} \rightarrow \text{CodeMatière}$

$\text{CodeMatière} \rightarrow \text{NomMatière}$

$\text{NumProf} \rightarrow \text{NomMatière}$ .

Les deux premières dépendances fonctionnelles sont directes, mais la troisième ne l'est pas : transitivité  $\text{NumProf} \rightarrow \text{CodeMatière} \rightarrow \text{NomMatière}$ .

#### 4.4.4 Clé d'une entité

Toutes les propriétés d'une entité doivent dépendre fonctionnellement de sa *clé* et ceci n'est plus vrai pour aucune des parties de cette clé.

Exemple 1 :

$\text{Num\_Etud} + \text{Nom\_Etud}$  **n'est pas une clé** puisque  **$\text{Num\_Etud}$  détermine  $\text{Adr\_Etud}$** . En revanche,  **$\text{Num\_Etud}$  est une clé** car :

$\text{Num\_Etud} \text{ -df} \rightarrow \text{Nom\_Etud}$

$\text{Num\_Etud} \text{ -df} \rightarrow \text{Adr\_Etud}$

#### 4.4.5 Propriétés des dépendances fonctionnelles

Réflexivité :  $a \text{ -df} \rightarrow a$

Projection :  $a \text{ -df} \rightarrow b+c \rightarrow a \text{ -df} \rightarrow b$  et  $a \text{ -df} \rightarrow c$

Augmentation :  $a \text{ -df} \rightarrow b \rightarrow a+c \text{ -df} \rightarrow b$

Additivité :  $a \text{ -df} \rightarrow b$  et  $a \text{ -df} \rightarrow c \rightarrow a \text{ -df} \rightarrow b+c$

Transitivité :  $a \text{ -df} \rightarrow b$  et  $b \text{ -df} \rightarrow c \rightarrow a \text{ -df} \rightarrow c$

Pseudo-transitivité :  $a \text{ -df} \rightarrow b$  et  $b+c \text{ -df} \rightarrow d \rightarrow a+c \text{ -df} \rightarrow d$

## 4.5 Règles relatives au MCD

Tout MCD doit être *normalisé, vérifié et décomposé*.

### 4.5.1 Normalisation des entités

Les entités du MCD doivent vérifier les règles suivantes :

**a. 1<sup>ère</sup> forme normale** : dans une entité, toutes les propriétés sont élémentaires et il existe au moins une clé caractérisant chaque occurrence (identifiant).

Exemple : (voir exemple du cours)

**b. 2<sup>ème</sup> forme normale** : toutes les propriétés d'une entité doivent dépendre de la clé par une dépendance fonctionnelle élémentaire.

Exemple : (voir exemple du cours)

**c. 3<sup>ème</sup> forme normale** : dans une entité, les propriétés doivent dépendre de la clé par une dépendance fonctionnelle élémentaire directe.

Exemple : (voir exemple du cours)

**d. forme normale de Boyce-Codd (BCNF)** : si l'entité possède un identifiant concaténé, un des éléments composant cet identifiant ne doit pas dépendre d'une autre propriété.

Exemple : (voir exemple du cours)

L'objectif de la normalisation est d'éliminer la redondance (répéter la désignation du produit commandé à chaque commande du même produit) et les anomalies de mise-à-jour (annuler employé → annuler catégorie)

### 4.5.2 Vérification

Dans toute occurrence d'entité ou de relation, il ne doit y avoir qu'une seule valeur de chaque propriété. Pour les entités ceci résulte de la première forme normale.

Exemple : (voir exemple du cours)

*"Les propriétés d'une relation doivent dépendre fonctionnellement des identifiants des entités concernées par la relation. La concaténation de ces identifiants forme l'identifiant de la relation."*

### 4.5.3 Normalisation des relations

Chaque propriété d'une relation doit dépendre fonctionnellement de *l'ensemble* des identifiants des entités de la collection de cette relation, mais d'aucun sous-ensemble.

Exemple : (voir exemple du cours)



#### 4.5.4 Décomposition des relations

La décomposition consiste à remplacer une relation de dimension n (supérieure à 2) en plusieurs relations de dimensions plus petites en utilisant les dépendances fonctionnelles présentes sur la relation.

La décomposition n'est possible que si les deux conditions sont vérifiées :

1. Cardinalité minimale des entités à gauche de la dépendance fonctionnelle doit être 1 dans la relation à décomposer;
2. Si la dépendance fonctionnelle provient d'une autre relation, il faut qu'elle concerne les mêmes occurrences d'entités que la relation à décomposer.

Exemple : (voir exemple du cours)

#### 4.6 Démarche de construction du MCD

1. Etablir la liste des données recensées : cette liste est appelée *dictionnaire de données*.
2. Epurer le dictionnaire de données : ne garder que les données utiles pour le SI. De ce fait, les synonymes, les polysémies, les données calculées et les données concaténées doivent être supprimées. Le formalisme du DD est le suivant :

Code	Désignation	Type	Taille	Observation
Code_F	Numéro du fournisseur	N	3	
Nom_F	Nom du fournisseur	A	20	
Adr_F	...	..	..	
Date_Cde	Date de la commande	Date	..	
.	...	..	..	
.				
.				
.				

3. Etablir le graphe des dépendances fonctionnelles (structure d'accès théorique) : à partir des règles de gestion.
  - a. Il ne faut garder que les dépendances fonctionnelles élémentaires directes,
  - b. S'il reste des propriétés isolées, proposer des identifiants qui permettent de les déterminer.
4. Construire le MCD :
  - a. Les arcs terminaux d'une même origine → Entité,
  - b. Les arcs restants qui relient deux ou plusieurs entités traduisent des relations entre ces entités,
  - c. Tenir compte des règles de gestion qui ne sont pas encore couvertes (celles qui n'expriment pas de dépendances fonctionnelles),
  - d. Normaliser, vérifier et décomposer le MCD brut

## 5. Quantifier le MCD : description des entités et des relations.

## a. Description des entités :

Entité	Identifiant	Propriétés	Longueur	Nombres occurrences
....	....			

## b. Description des relations :

Relation	Entités	Cardinalités	Propriétés	Longueur	Nombres d'occurrences
....	....				