

# Le modèle conceptuel de données : MCD – Notion de dépendances fonctionnelles (DF)

- Dans un MCD la dépendance fonctionnelle signifie qu'un élément B dépend fonctionnellement de A si la connaissance de A implique la connaissance de B et on note  $A \twoheadrightarrow B$ .
- La DF s'applique dans les cas suivants :
  1. **Entre les attributs d'une même entité:** il s'agit d'une DF entre l'identifiant d'une entité et les autres attributs de l'entité.

On dit que 2 propriétés a et b sont reliées par une dépendance fonctionnelle ( $a \twoheadrightarrow b$ ) si la connaissance de la valeur de a détermine une et une seule valeur de b

La réciproque est fausse

**Exemple:**  $cd\_client \twoheadrightarrow nom\_client$

# Le modèle conceptuel de données : MCD – Notion de dépendances fonctionnelles (DF)

**2. entre entités:** On dit qu'il existe une dépendance fonctionnelle entre deux entités A et B et on note  $(A \twoheadrightarrow B)$  si toute occurrence de A détermine une et une seule occurrence de B.

- La cardinalité maximum 1 correspond toujours à une DF

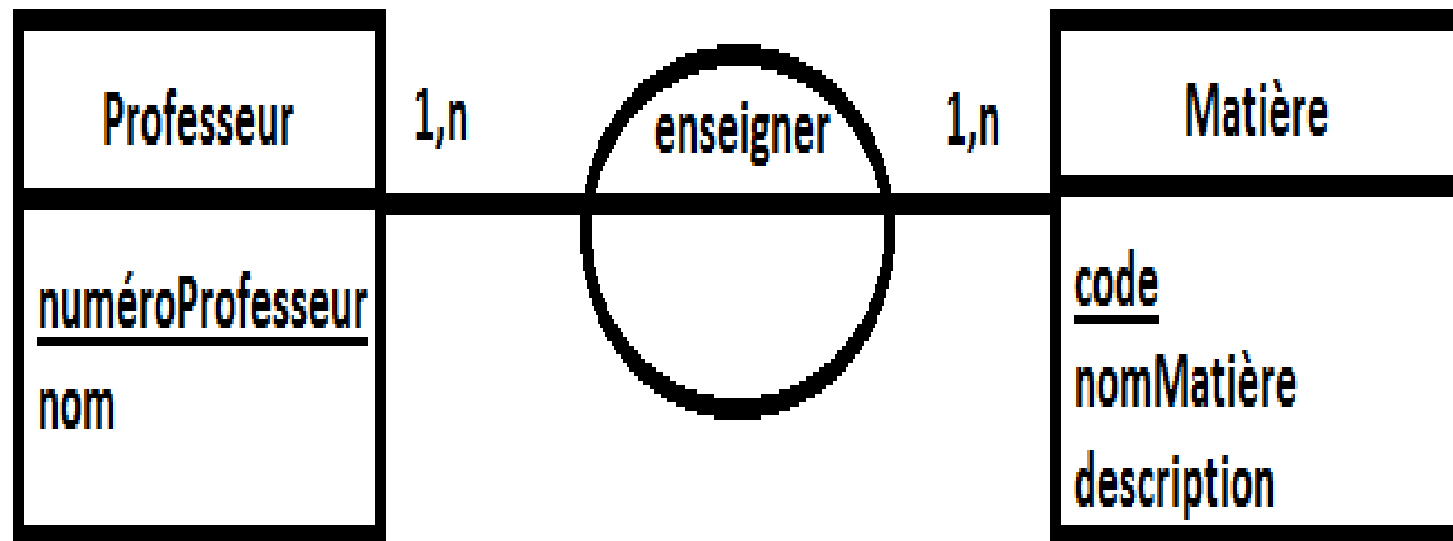
**Exemple :**

Commande ---- > Client

Est assimilable à :

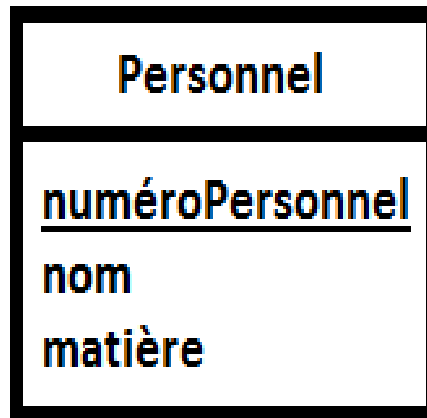
N°cde --- > Code\_Client

# Le modèle conceptuel de données : MCD – Les règles d'élaboration



# Le modèle conceptuel de données : MCD – Les règles d'élaboration

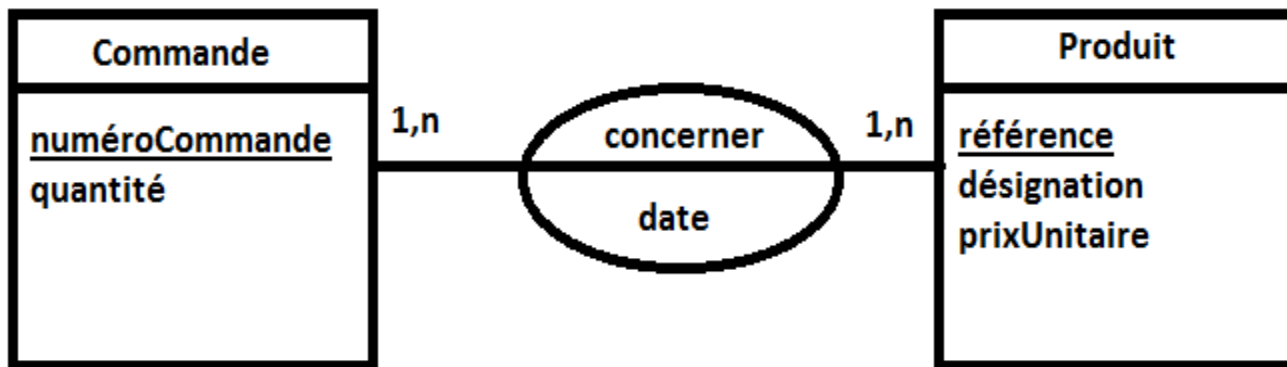
- Cas 2 : Attributs sans signification



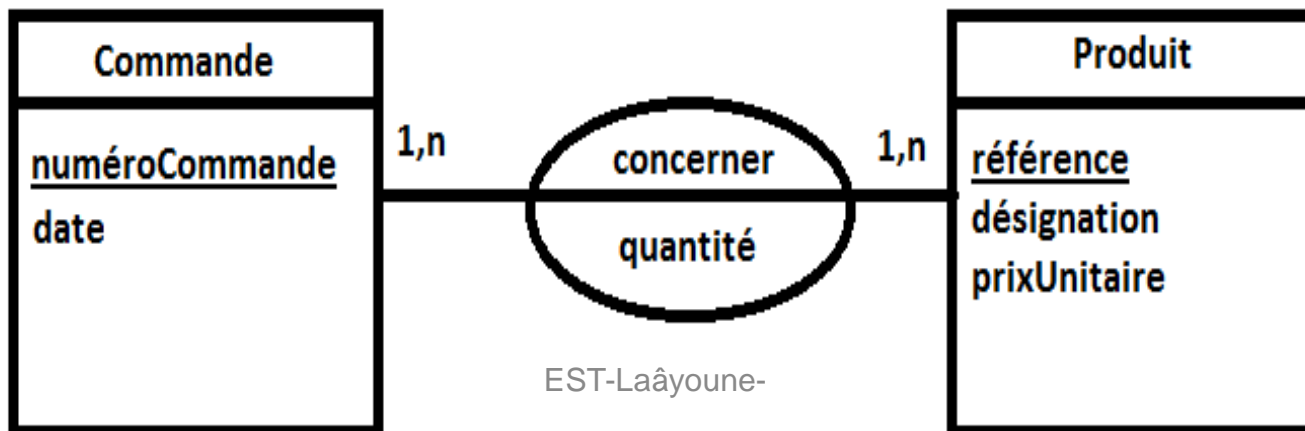
- L'attribut "Matière" ne prend pas de valeur pour une secrétaire ou un surveillant.

# Le modèle conceptuel de données : MCD – Les règles d'élaboration

- Cas 3 : Dépendance incomplète



L'attribut "Date" ne dépend pas du produit et l'attribut « quantité » peut prendre plusieurs valeurs.



# règles relatives au MCD – La normalisation

La normalisation **élimine les redondances**, ce qui permet:

- une diminution de la **taille** de la base de donnée sur le disque
- une diminution des **risques d'incohérence**
- d'éviter une **mise à jour** multiple des mêmes données

## Première forme normale (1FN)

Une entité est normalisée en ***première forme normale*** si :

1. elle possède une clé qui identifie formellement chaque occurrence
2. chaque attribut dépend fonctionnellement de la clé
3. chaque attribut ne peut avoir qu'une seule valeur par enregistrement
4. aucun attribut n'est décomposable en plusieurs attributs

# règles relatives au MCD – La normalisation

## *Exemple :*

CLIENT (nom-client, adresse), Cette entité n'est pas en 1FN

Car il n'y a pas de clé (plusieurs clients peuvent avoir le même nom).  
D'autre part adresse-client est une concaténation de rue et ville et ne constitue alors pas une propriété élémentaire.

Il doit être :

CLIENT (code\_client, nom, rue, ville)

# règles relatives au MCD – La normalisation

## Deuxième forme normale (2FN):

Une entité est en **deuxième forme normale** si et seulement si :

1. elle est en 1FN
2. toute propriété de l'entité doit dépendre de la clé par **une dépendance fonctionnelle élémentaire**. Autrement dit toute propriété de l'entité doit dépendre de tout l'identifiant.

la **DF**(Code-client + Nom-Client----> Adresse-Client) n'est pas élémentaire puisque la connaissance de Code-Client (partie de Code-Client + Nom-Client) suffit à déterminer l'adresse

Code-client ----- > Adresse-Client est élémentaire

**Conséquence:** toutes les entités qui n'ont qu'un **seul attribut** clé, sont en 2FN si elles sont en 1FN.



# règles relatives au MCD – La normalisation

## Exemple :

LIGNE (num commande.**num article**, **description\_article**,  
quantité\_commandée)

- description\_article **n'est pas élémentaire**, puisque « description\_article » ne dépend que d'une partie de la clé « num\_article »

### *Elle doit être :*

LIGNE (num commande)

Liée par la relation « Concerne (quantité) » à l'entité

ARTICLE (num article, description\_article)

# règles relatives au MCD – La normalisation

## Troisième forme normale (3FN)

- Une entité est en **3<sup>o</sup> forme normale** si et seulement si :
  - elle est en 2<sup>o</sup> forme normale
  - toute propriété de l'entité doit dépendre de l'identifiant par une dépendance fonctionnelle élémentaire directe, c'est-à-dire qu'aucun attribut ne doit dépendre de la clé par transitivité.

### Exemple :

CLIENT (code\_client, nom\_client, code\_categ, nom\_categ)

Cette relation n'est pas en 3FN car la dépendance fonctionnelle  $\text{code\_client} \twoheadrightarrow \text{nom\_categ}$  n'est pas directe du fait de la transitivité  $\text{code\_client} \twoheadrightarrow \text{code\_categ} \twoheadrightarrow \text{nom\_categ}$

### Ça doit être :

CLIENT (code\_client, nom\_client)

Lié par la relation « appartient à » à l'entité

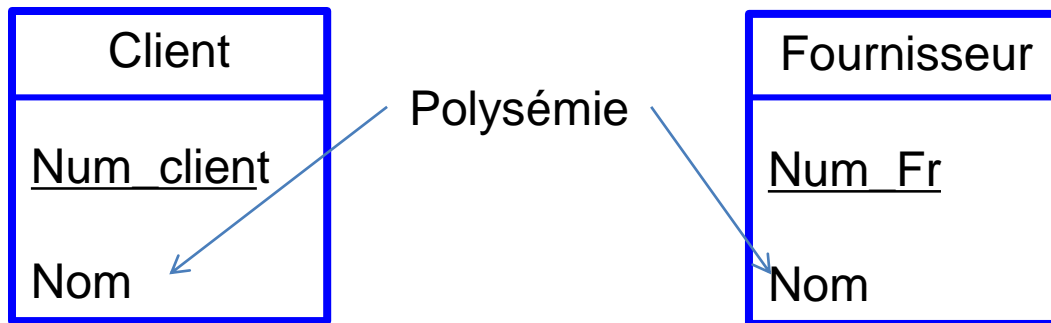
CATEGORIE (code\_categ, nom\_categ)

# règles relatives au MCD – La normalisation

## Quatrième forme normale (3FN)

- Une entité est en **4<sup>°</sup> forme normale** si et seulement si :
  - elle est en 3<sup>°</sup> forme normale
  - Une propriété ne peut qualifier qu'un seul objet ou qu'une seule relation.

Par exemple:



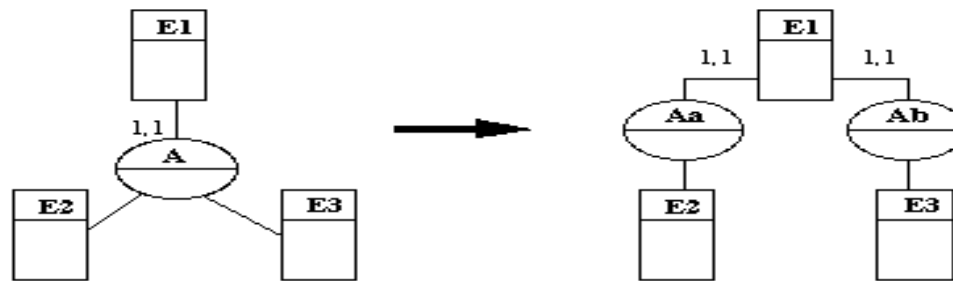
la propriété «Nom» ne peut être présente à la fois dans l'objet «Client» et dans l'objet «Fournisseur»; si tel est le cas, alors il faut appeler l'une « Nom client » et l'autre « Nom fournisseur »

# règles relatives au MCD – La décomposition

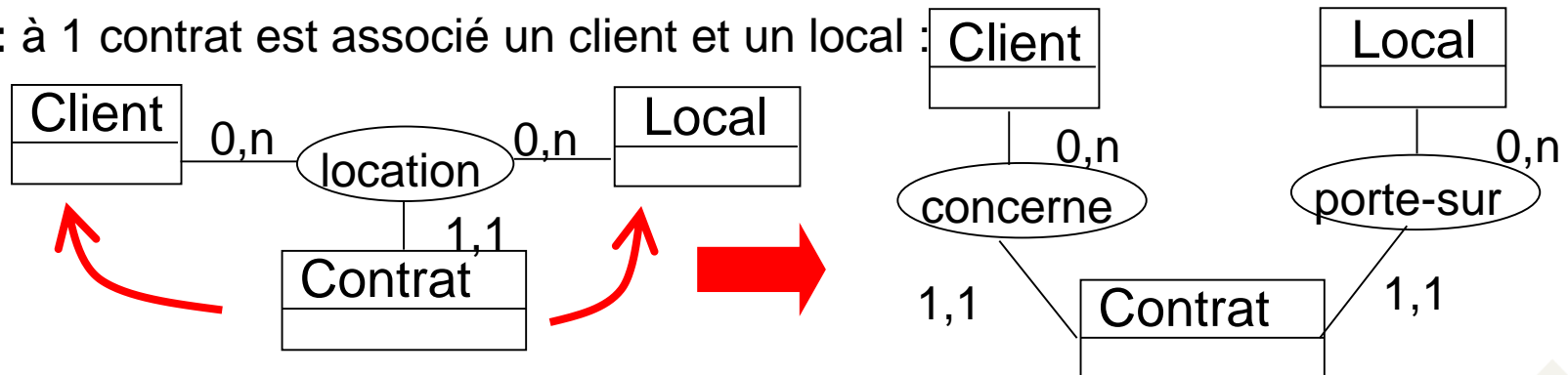
## *La décomposition des associations n-aires*

Il faut garder un minimum d'associations d'aires.

**1. La première type de décomposition:** lorsqu'une (et une seule) entité appartenant à la collection est impliquée avec des cardinalités (1, 1).

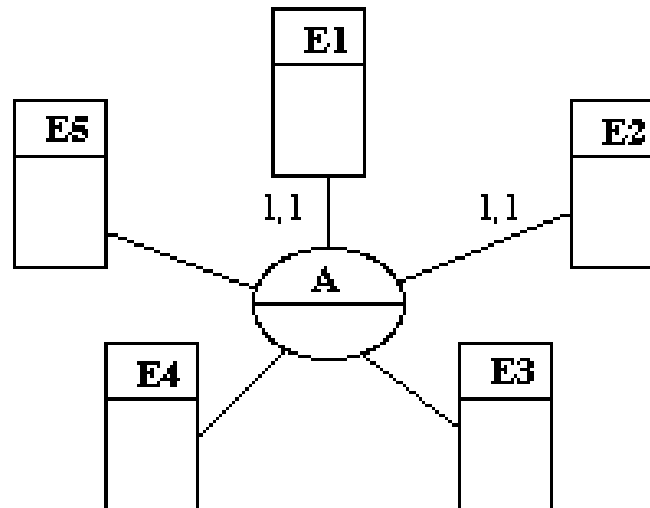


**Exemple:** à 1 contrat est associé un client et un local :



# règles relatives au MCD – La décomposition

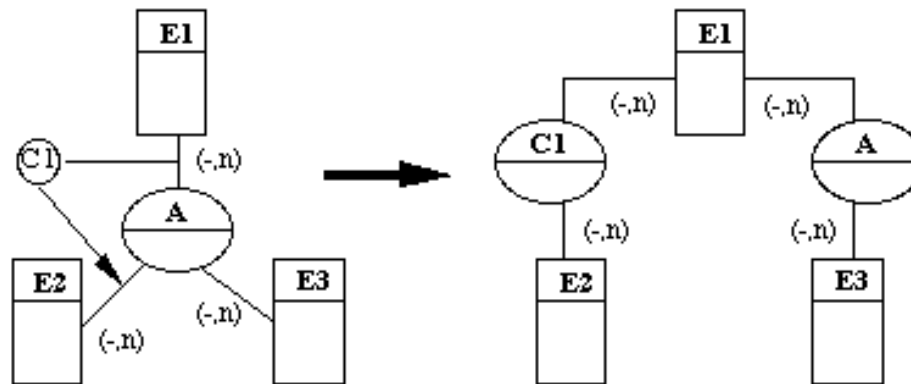
seul le couple (1, 1) provoque une telle décomposition ; ce couple ne peut exister en plusieurs exemplaires dans une association.



Association mal-formée, impossible à décomposer en l'état

# règles relatives au MCD – La décomposition

**2. La deuxième type de décomposition:** lorsque une association n-aire est accompagnée d'une contrainte de type dépendance fonctionnelle impliquant (n-1) entités de la collection



La contrainte C1 permet de dire que lorsque l'on connaît la valeur de l'identifiant de E1 on a la valeur de celui de E2