



# Analyse et Conception des SI Relationnels

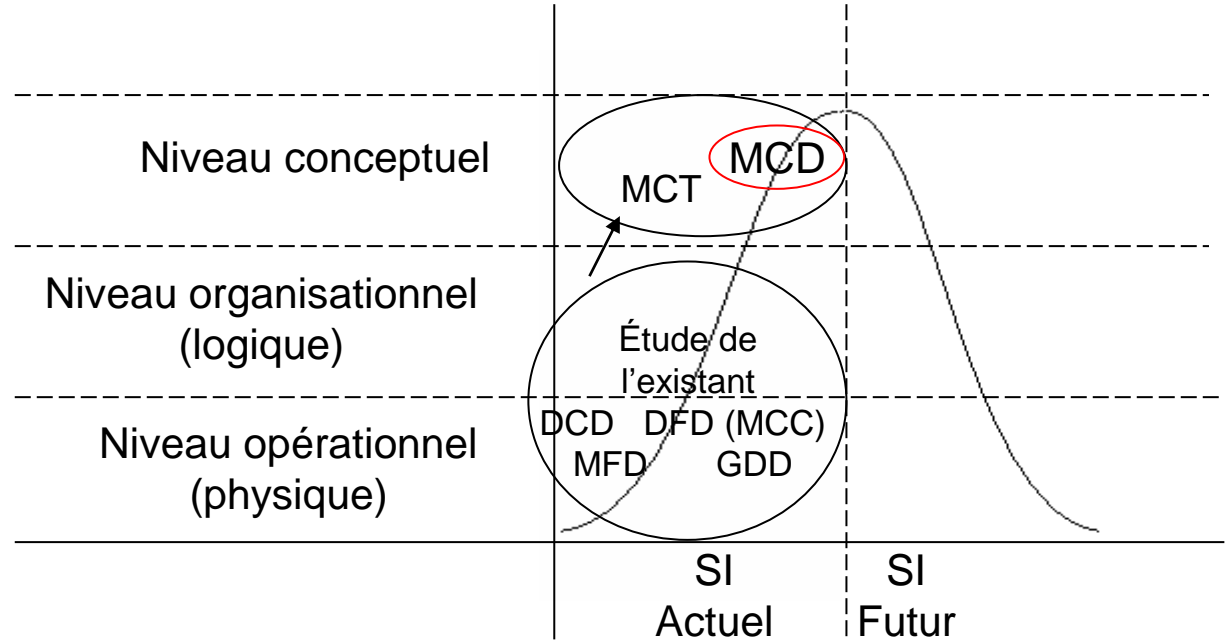
Pr. EL HALOUI

# Rappels

- Etude de cas : MCT du processus facturation
- Gestion des cartes bancaires
- Gestion de réparation des téléphones
- Questions ?

# Modèle Conceptuel des Données (MCD)

- Rappel :
  - Modèles pour chaque niveau
  - Principe de séparation des données (statique) et des traitements (dynamique)
- Nécessité de structurer, classer et stocker les données manipulées
- MCD : Représentation statique du SI (données) au niveau conceptuel.



# Modèle Conceptuel des Données (MCD)

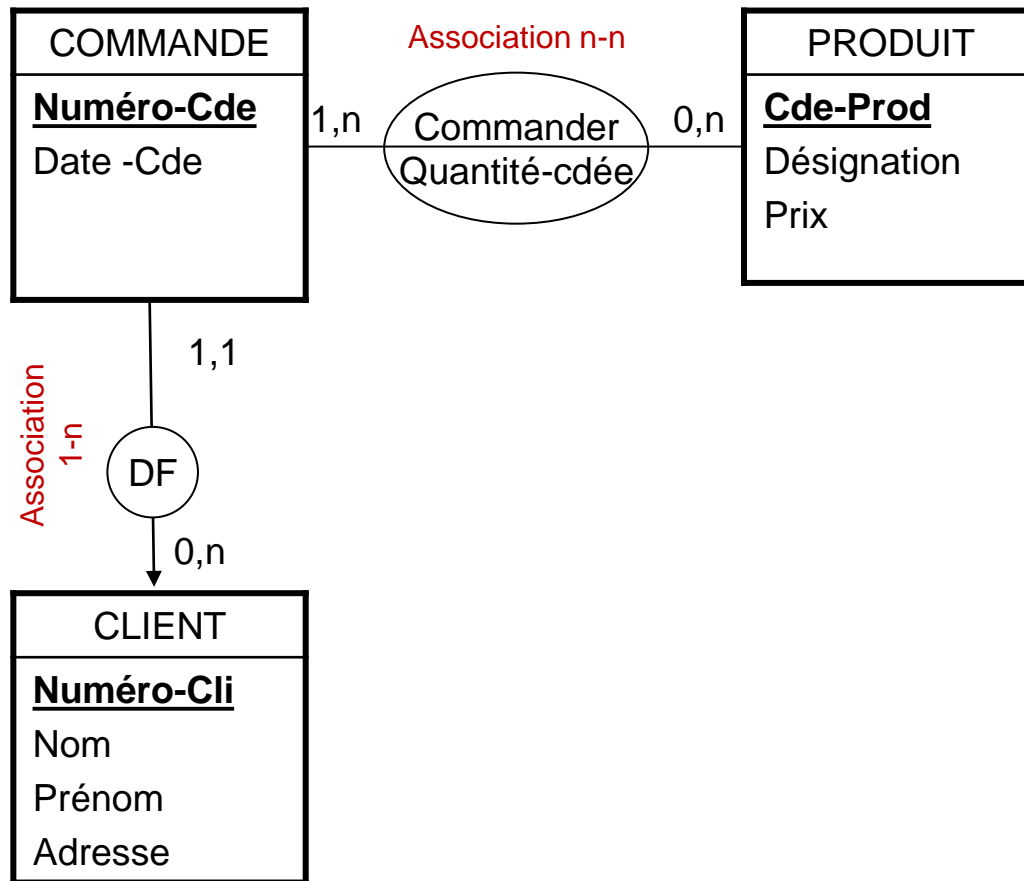
- Le modèle conceptuel de données (MCD) est un modèle qui **représente le système étudié du point de vue données** (niveau conceptuel) ;
- Le MCD est un modèle de données global représentant le **regroupement des données et leurs relations** loin de toute influence organisationnelle ou opérationnelle (organisation des données en document ou tout autre support, traitements, ...) ;
- Le MCD est un modèle appelé également Modèle **entités-relations** ou **entités- associations** ;
- Permet de décrire un ensemble de **données** relatives à un **domaine** défini afin de les intégrer ensuite dans une **Base de Données**.

# MCD : Concepts

- Repose sur un ensemble de concepts :
  - Entité
  - Propriété (ou attribut)
  - Clé
  - Association (ou relation)
  - Cardinalité
  - Dépendance Fonctionnelle (DF)

# MCD : Concepts

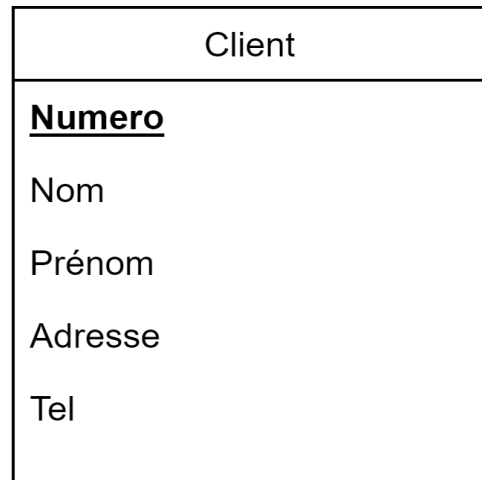
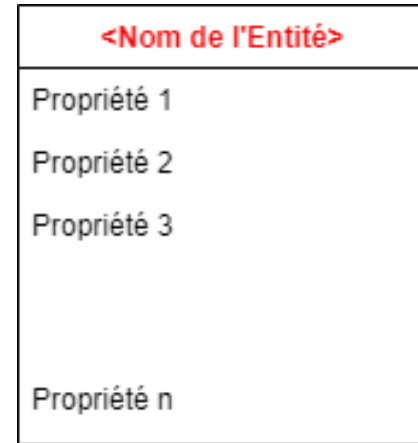
- Exemple de MCD :



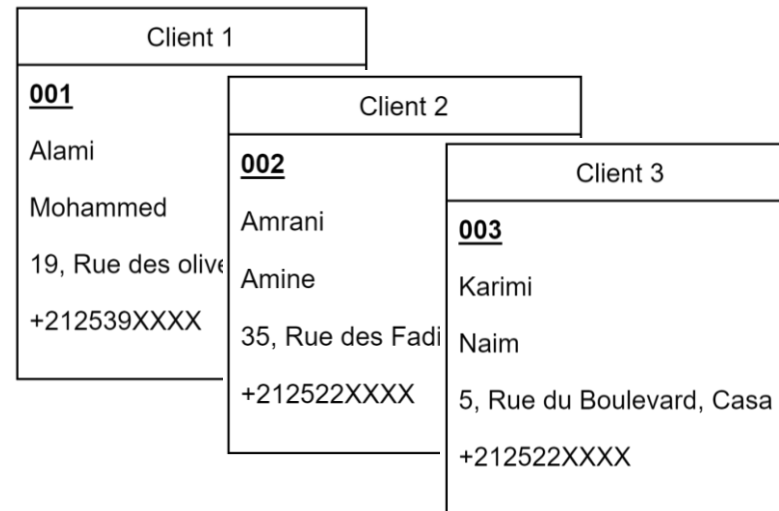
- Entités** : Produit, commande, client
- Association (Relation)** : commander
- Clé** : Numéro-Cde, Cde-Prod, ...
- Propriété (attribut)** : Désignation, prix, Numéro-Cde, quantité-cdée
- Cardinalités** : 1,n // 0,n // 1,1
  - minimale
  - maximale
- Dépendance fonctionnelle** : DF

# MCD : Entité

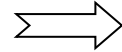
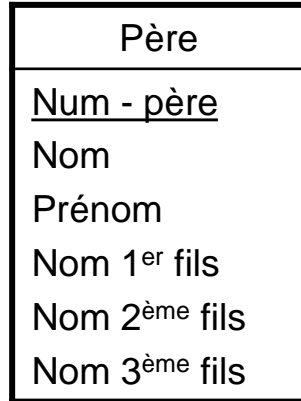
- Une entité : un **groupement d'éléments de données** (propriétés) **désignant un type** (entité) pour lequel il y aura plusieurs exemplaires dans le système étudié (occurrences).
- Une entité est caractérisée par son **nom** et ses **propriétés**
- **Exemple :**
  - L'entité Client peut regrouper les propriétés Numéro, Nom, Prénom, Adresse, Tel.
  - Chacun de ces clients représente **une occurrence** de l'entité Client



Occurrences



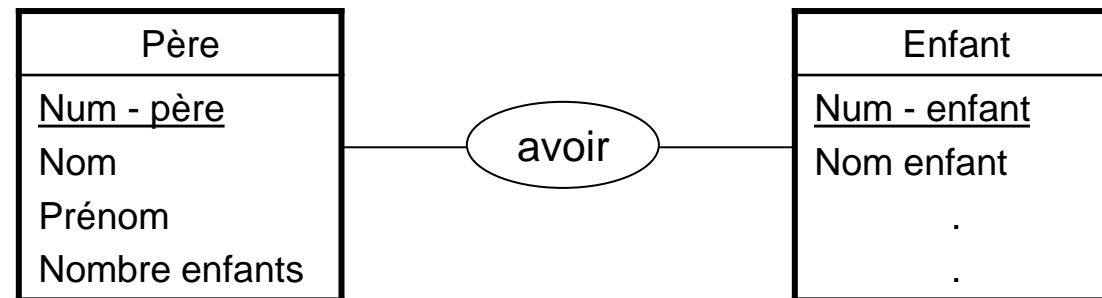
# MCD : Entité



Ce n'est pas une entité (un père a plus de 3 fils n'est pas représenté ).



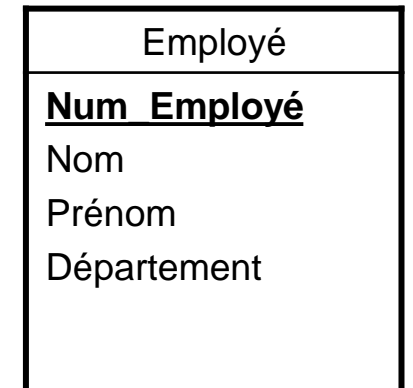
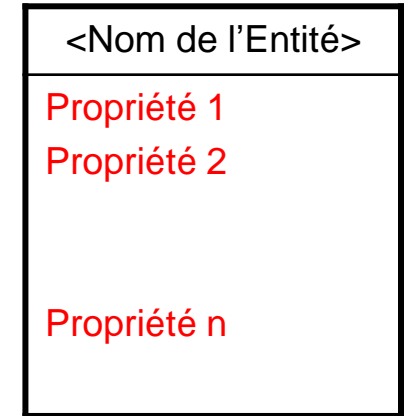
Une solution





# MCD : Propriété

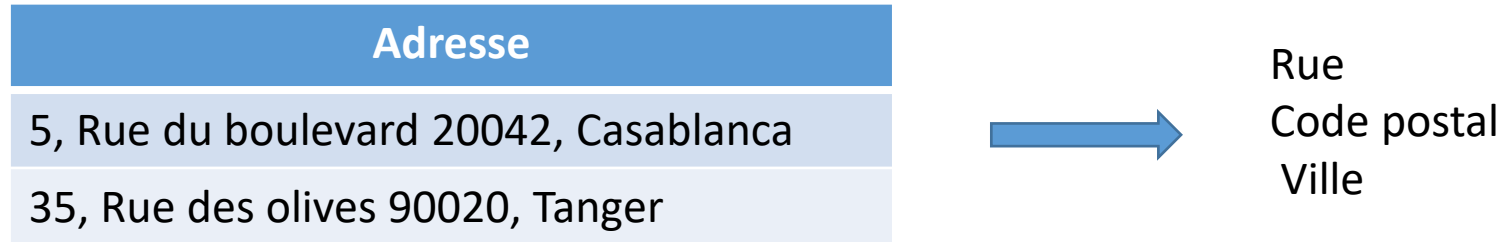
- Une propriété est une **donnée élémentaire** d'une entité ;
- La propriété est indiquée à l'intérieur du rectangle qui représente l'entité correspondante. **Nom de la propriété.**
- Les valeurs prises par une propriété sont appelées **occurrences**.
- Une propriété peut être une valeur **numérique**, une valeur sous forme de **texte (Alphanumérique)** ou encore une **date**.
- A l'intérieur de chaque occurrence, chaque propriété ne prend qu'une **seule valeur** au maximum.



# MCD : Propriété

- Une propriété est dite **simple** (ou **atomique**) si chacune des valeurs qu'elle regroupe n'est pas décomposable.

- **Exemple :**



- NB : La décomposition d'une propriété ne doit pas être systématique :
  - Si l'exploitation est toujours globale, l'atomicité n'est pas nécessaire (sauf pour le cas de Normalisation → à voir plus tard au niveau de ce cours).
  - Les SGBD actuels prévoient des types dans ce sens , c'est pourquoi ces types sont parfois considérés atomiques.

# MCD : Propriété

- Le nom de la propriété doit être le plus **explicite** possible à ce que représente la propriété et présente un **intérêt** pour le domaine étudié.
- Exclure les **synonymes** (des noms différents pour une même propriété : Produit ou Article) et les **polysémies** (propriétés utilisant les mêmes orthographes mais décrivant des réalités différentes : « Date » pour date d'arrivée et date de réservation).
- Le principe de la non-redondance impose que chaque propriété, correctement identifiée, n'apparaisse qu'une seule fois dans le modèle. Une propriété est **unique** et ne peut pas être rattachée à plusieurs entités différentes ;

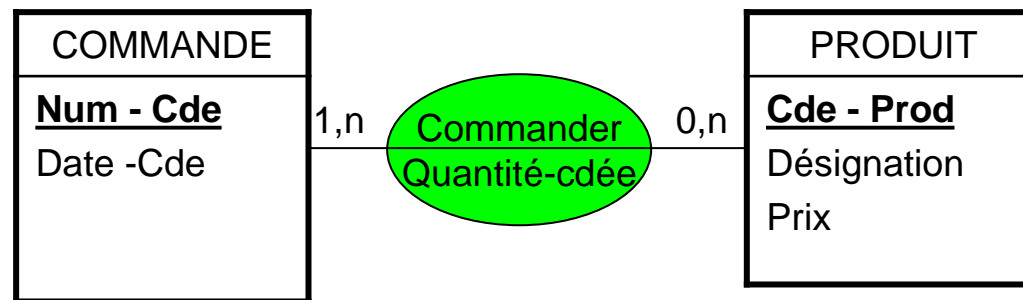
# MCD : Clé ou Identifiant d'une Entité

- Une clé est une propriété (ou un ensemble de plusieurs propriétés) qui **permet de distinguer les différentes occurrences**.
- Au niveau de chaque entité, il faut identifier une propriété « la clé » qui joue le rôle de clé (primaire).
- Pour chaque occurrence, l'attribut clé prend une **valeur différente** des autres.
- Exemple : La propriété « Num\_Etudiant » dans cette entité « ETUDIANT » présente une clé primaire.
- Une clé peut être :
  - Une **propriété naturelle**. *Exemple* : Le nom d'un pays pour une entité Pays
  - Une **propriété artificielle** qui est inventée par le créateur du MCD. *Exemple* : Le numéro d'un client pour une entité Client
  - Une **propriété composée** d'autres propriétés naturelles. *Exemple* : Le nom et la localité pour une entité Entreprise.

ETUDIANT
<u>Num_Etudiant</u>
Nom
Prénom
Date _ naissance

# MCD : Association

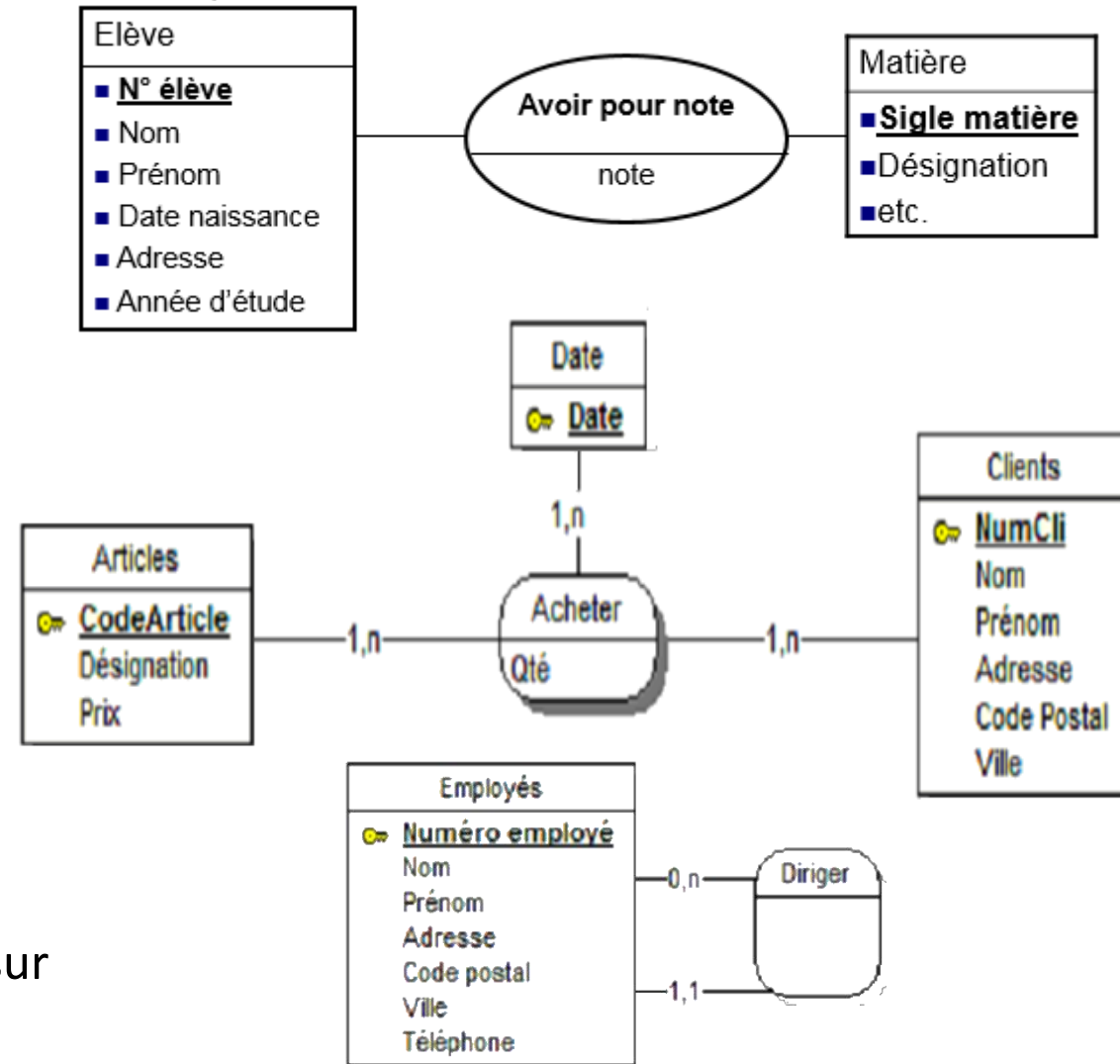
- L'association est une **relation** qui lie des entités pour représenter une liaison donnée.
- Chaque relation possède un **nom**, en général un verbe à l'infinitif.
- Une association peut avoir des propriétés.
- La clé d'une association est la combinaison des clés des entités participantes à la relation.
- Exemple : L'association « COMMANDER » lie les entités « COMMANDE » à « PRODUIT », elle a une propriété : Quantité –cdée . Sa clé est : 'Num –Cde, Code –Prod'



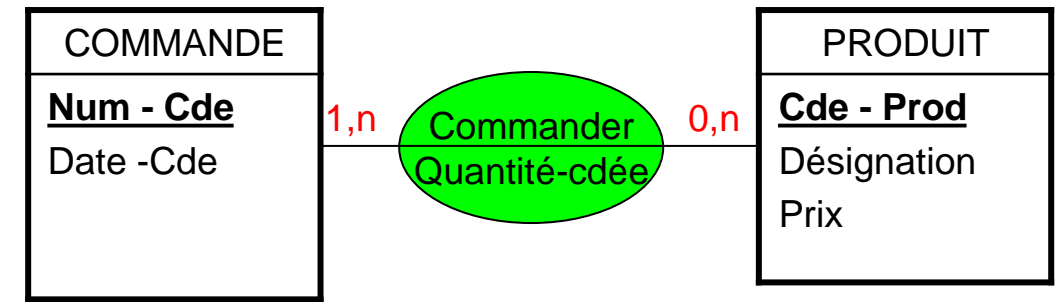
# MCD : Association

On distingue trois types d'associations (relations):

- Les associations **binaires**, qui sont liées à 2 entités
- Les associations **ternaires**, qui sont liées à 3 entités
- Les associations **réflexives**, une association d'une entité sur elle-même



# MCD : Cardinalité

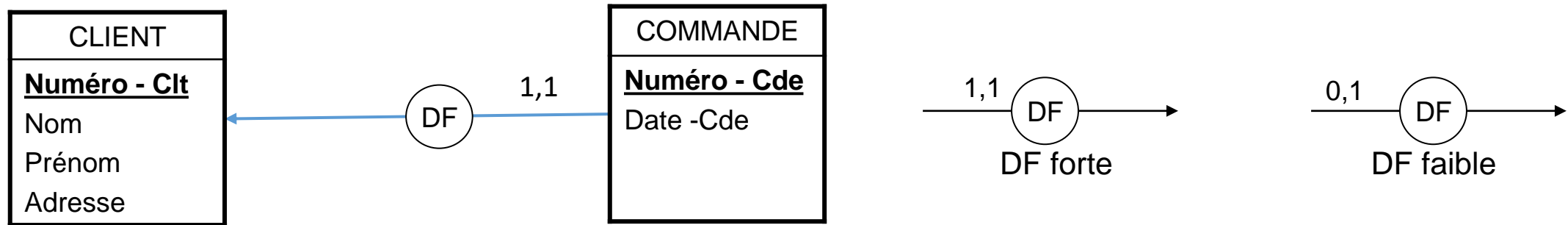


- L'objectif des cardinalités est de montrer le nombre des occurrences ENTITE impliquées dans une association.
- On se limite aux deux cardinalités ; Minimale et Maximale (min, max) :
  - **Cardinalité minimale** : Le nombre d'occurrences ENTITE minimales pouvant participer dans une association.  
Les valeurs d'une cardinalité minimale sont « 0 » ou « 1 ». (graphiquement c'est ce qui est écrit à gauche).  
*Exemple:* Est-ce qu'on peut ne pas avoir de commande dans l'association "commander" ?
    - si oui → 0
    - si non → 1
  - **Cardinalité maximale** : Le nombre d'occurrences ENTITE maximales pouvant participer dans une association.  
Les valeurs d'une cardinalité maximale sont « 1 » ou « n ». (graphiquement c'est ce qui est écrit à droite).  
*Exemple :* Est-ce qu'on peut avoir une commande dans l'association "commander" plusieurs fois?
    - si oui → n
    - si non → 1

# MCD : Dépendance Fonctionnelle (DF)

Deux types :

- **1) DF entre deux entités** : Une DF est une association qui a **1 comme cardinalité maximale** (c'est une liaison forte indiquant que l'entité d'origine dépend fonctionnellement de l'entité cible) :
  - *Exemple* : Une commande dépend fonctionnellement du client qui l'a fait (càd que toute commande est liée à un seul client). Avec la cardinalité maximale qui est égale à 1 on va dire que toute commande est **liée à un et un seul client**.





# MCD : DÉPENDANCE FONCTIONNELLE (DF)

- **2) DF entre attributs** : On dit que B est en dépendance fonctionnelle (DF) de A si à une valeur quelconque de la propriété A, on ne peut faire correspondre qu'une seule valeur au plus pour la propriété B.
- On note :  
$$\begin{array}{ccc} A & \longrightarrow & B \\ \text{(source)} & & \text{(cible)} \end{array}$$
- Si on connaît la valeur de A, on peut en déduire une seule valeur de B.
- *Exemple* : Num-Client  $\longrightarrow$  Nom-Client (c'est une DF)
- La réciproque n'est pas vraie (connaissant B, on ne peut pas déduire A).
- *Exemple* : Nom-Client  $\not\longrightarrow$  Num-Client (ce n'est pas une DF)

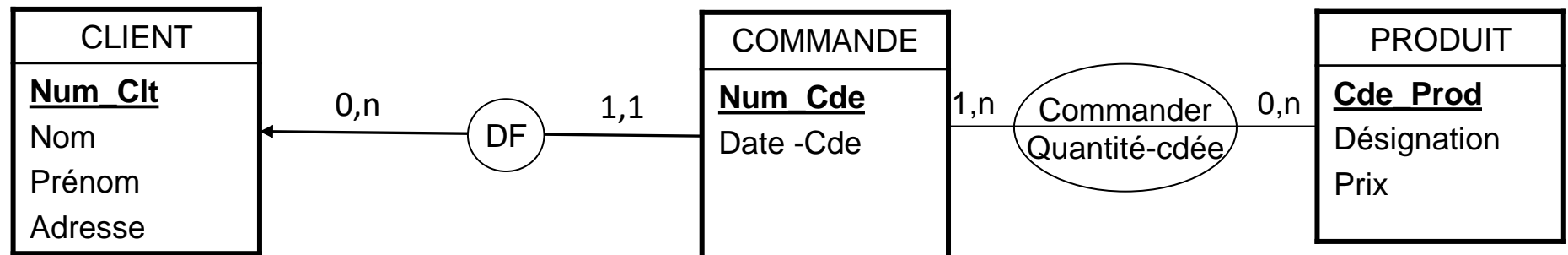
# MCD : DÉPENDANCE FONCTIONNELLE (DF)

- Une DF entre attributs concaténés : Une DF entre des attributs de départ  $(a, \dots, a)$  vers des attributs d'arrivée  $(b, \dots, b)$  est une relation notée  $a, \dots, a \rightarrow b, \dots, b$  exprimant les propos suivants :
  - Les attributs  $a, \dots, a$  déterminent les attributs  $b, \dots, b$
  - ou Les valeurs de  $a, \dots, a$  déterminent de façon unique les valeurs de  $b, \dots, b$
  - ou à chaque ensemble de valeurs prises par les attributs  $a, \dots, a$  il y a un et un seul ensemble de valeurs prises par les attributs  $b, \dots, b$

# MCD : DÉPENDANCE FONCTIONNELLE (DF)

- **Exemple :**

- Num\_Cde  $\longrightarrow$  Date\_Com
  - Num\_Cde  $\longrightarrow$  Num\_Clt
  - Num\_Cde, Code\_Prod  $\longrightarrow$  Qté \_ Cdée
  - Num \_ Cde, code \_ prod, Num \_ Clt  $\longrightarrow$  Qté \_ Cdée
  - Num \_ Cde  $\not\longrightarrow$  code \_ prod
  - Date \_ Cde  $\not\longrightarrow$  Num \_ Cde
  - Qté \_ Cdée  $\not\longrightarrow$  Num \_ Cde, code \_ prod
  - Num \_ Cde, code \_ prod  $\xrightarrow[\text{totale}]{\text{DF}}$  Qté \_ Cdée
  - Num \_ Cde, code \_ prod, num \_ Clt  $\xrightarrow[\text{partielle}]{\text{DF}}$  Qté \_ Cdée
- en plus
- sont des DF
- ne sont pas des DF



# MCD : DÉPENDANCE FONCTIONNELLE (DF)

## Définitions et Remarques :

- Une DF est dite **totale (ou élémentaire)** si tous les attributs de départ sont nécessaires pour assurer la DF (Dès qu'on supprime un attribut de départ la DF n'est plus valable).
- Une DF est dite **partielle** s'il y a des attributs de départ qui ne sont pas nécessaires pour la DF (en les supprimant la DF est toujours valable).
- Une DF entre entités n'est autre qu'une **DF entre les clés** de ces deux entités
- La **clé** d'une entité est l'attribut qui détermine par **DF les autres attributs** de la même entité.

# Exercice : Gestion hôtelière

Un groupe hôtelier vous demande de réaliser une application de gestion hôtelière. Cette application doit permettre la gestion de 8 hôtels, chaque hôtel possède 100 chambres maximum. Pour chaque hôtel il y a au maximum 9 catégories de chambres différentes(capacité, degré de confort). Cette application doit gérer aussi, les clients ainsi que leurs réservations.

## **Travail à faire :**

1. Trouver les entités en proposant leurs propriétés
2. Trouver les associations et les cardinalités
3. Elaborer le Modèle Conceptuel de Données