

**1 ) MERISE :**

- Méthode de conception et de développement de SIE
- SIE → Ensemble des informations organisées, structurées au sein d'une organisation particulière et visant un but.
- Evolution de MERISE → MERISE II ( Pour gérer la complexité des organisations )

- Méthode systémique

On prend un problème dans sa globalité

Définit des domaines d'applications

Méthode analytique

Un problème à traiter

- Traitement des sous-problèmes
- Restitution dans un ensemble plus complet

**2 ) Modélisation :**

Modèle : Représentation de la réalité à l'aide de symboles

Approche progressive de la complexité d'un problème :

	<b>Communication ( Flux ) ( 1 )</b>	<b>Données ( 2 )</b>	<b>Traitements ( 3 )</b>
<b>Conceptuel ( QUOI )</b>	<b>MCC</b>	<b>MCD</b>	
<b>Organisation ( QUI / QUAND )</b>			
<b>Physique</b>		<b>MPD Modèle Relationnel</b>	

( 1 ) Echange d'informations entre acteurs ( Porteur de données )

( 2 ) Nature, organisation, relation entre données

( 3 ) Les actions réalisées sur les données

### 3 ) Modélisation des flux :

#### 3.1 Objectif :

Représenter les mouvements des données à l'intérieur d'un Système d'Information et l'environnement.

#### 3.2 Concepts :

##### 3.2.1 Domaine d'étude :

Ensemble des traitements homogènes réalisés dans le domaine

*Ex : Traitement des commandes / Traitement des règlements des clients*

##### 3.2.2 Poste de travail : Acteur ( Personnes physiques ou morales )

- Acteur interne : interne au domaine étudié

- Acteur externe : au SIE de l'entreprise

*Ex : Client / Fournisseurs / Etat / Sous-traitants*

- Domaine connexe : Avec lequel le domaine principal interagit.

*Ex : Gestion financière*

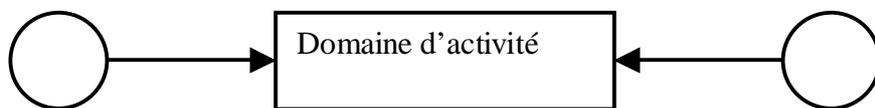
##### 3.2.3 Flux : Elément porteur d'information échangé entre les acteurs

*Ex : Bon de commande*

#### 3.3 Schéma conceptuel de flux :

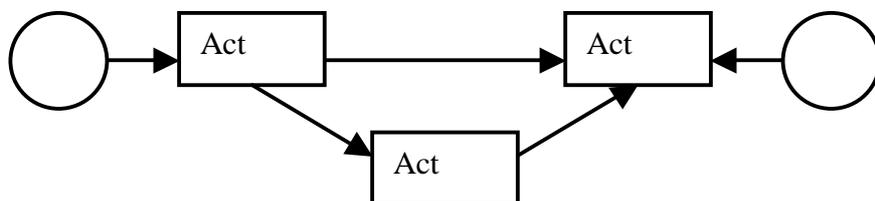
- Modèle de contexte

Il représente le domaine d'étude et l'extérieur



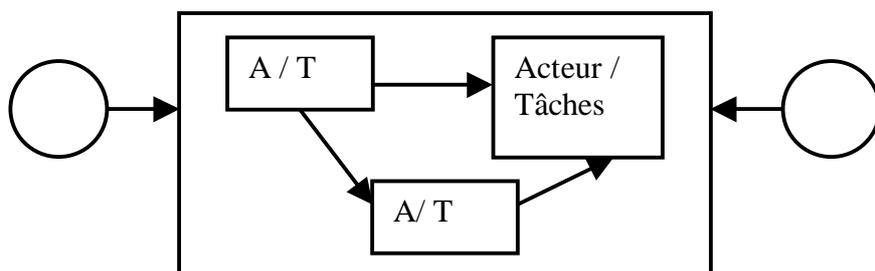
- Diagramme des flux

conceptuel de niveau 1 ( Zoom du domaine d'étude ) traitant des activités et des échanges entre activités



- DFC de niveau 2

On éclate chaque activité en tâche en relation avec l'extérieur

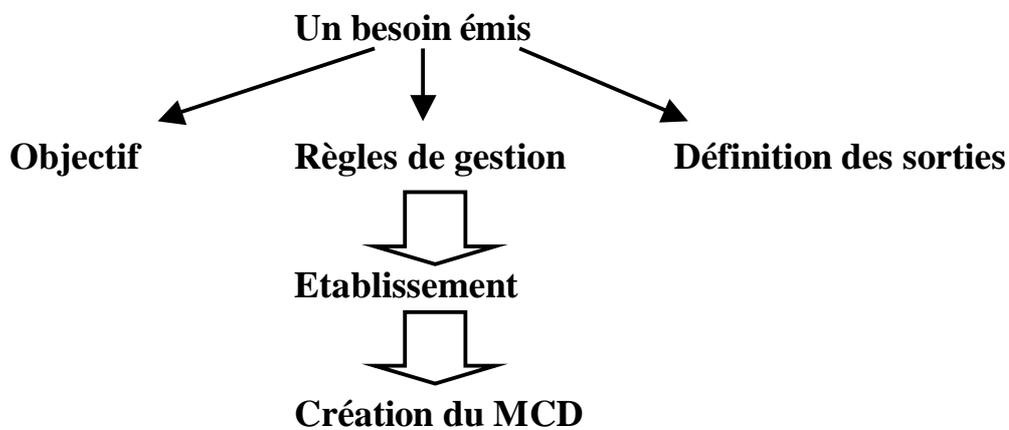


**MCC :**

- 1. Lister les acteurs ( Qui est amené à agir sur le système )**
- 2. Etablir la liste des flux ( Toutes les informations échangées )**
- 3. Réalisation du MMC ( acteur + n° des flux )**

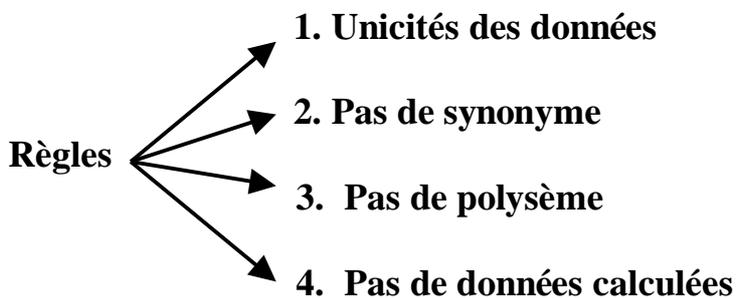
**MCD :**

**Démarche de conception d'une application BDD**



**Dictionnaire des données**

N°	Nom	Désignation	Type de données + Longueu
----	-----	-------------	---------------------------



**MCD :****Exercice 1 : ( COMMANDES / CLIENTS / PRODUITS / TRANSPORTEURS )**

1. Recherche des entités
2. Affecter les propriétés aux entités
  - a. Propriétés identifiant  
Ce qui permet de représenter de manière unique chaque entité  
( Garantie l'unicité d'un élément de l'entité )
  - b. Affecter les autres propriétés en respectant la règle de dépendance fonctionnelle
    - Deux propriétés A et B sont en dépendance fonctionnelle ssi pour toute valeur de A il existe **une unique** valeur de B.

Ex : N° Client ( A ) → Ville Client ( B )

Pour un numéro client donné, j'ai une dépendance fonctionnelle pour la ville du client correspondante.

⇒ On parlera de l'occurrence de l'entité  
Occurrence d'une entité → Un élément de l'entité ( Ex : Client DURANT )

## 3. Relation ( associer des occurrences entre elles )

- Nommer les relations
- Affecter des propriétés aux relations

Ce qui permet d'identifier de manière unique une « quantité commandée » c'est le couple

( N° Commande, N° Produit ) **DF** → Quantité commandée

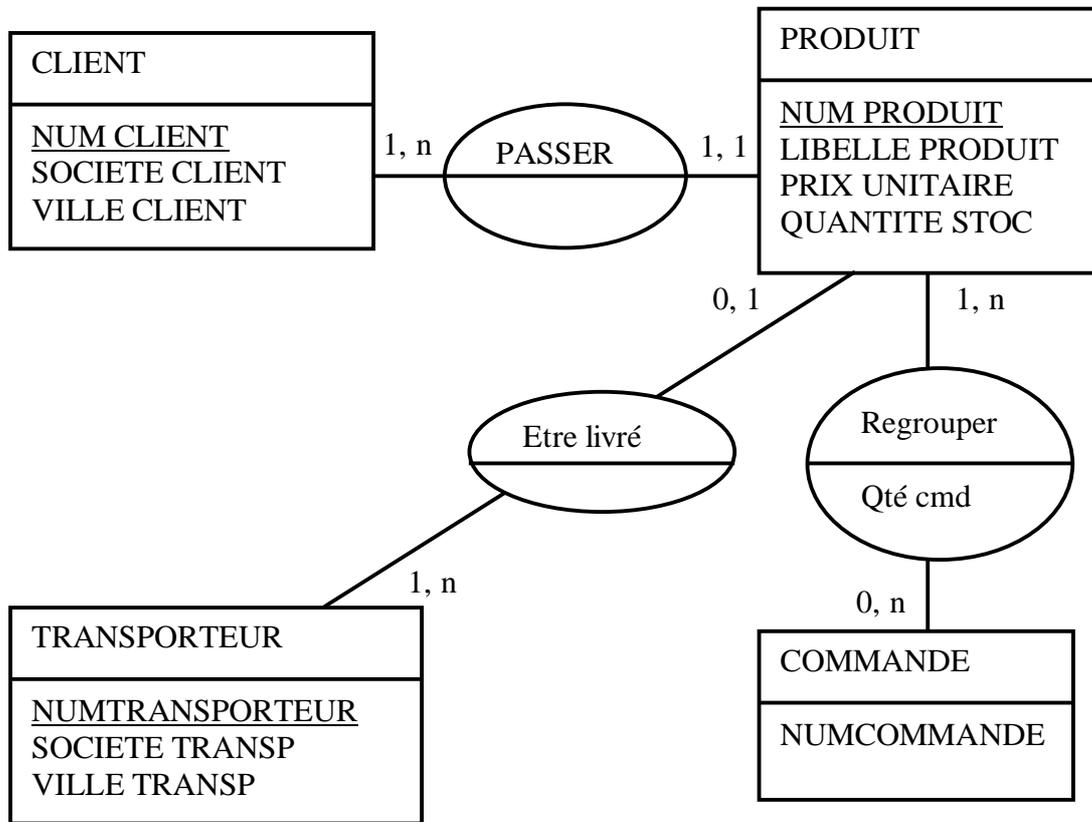
- Définition des cardinalités des relations ( Quantifier les relations dans les deux sens )

	MIN : 1		commandes
Un client passe			
	MAX : n		
		MIN : 1	
Une commande est passée			clients
		MAX : 1	

⇒ Occurrence d'une relation : Association d'occurrence des entités liés

Ex : Relation Passer :  
Toutes les commandes passées avec la mention du client correspondant )

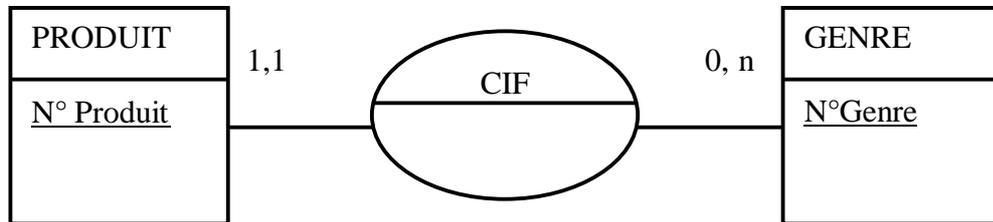
( Association des deux clés des entités )



**Typologie de relations ( Standard ) Structure de données d'un MCD**

**1. Relation CIF ( Contrainte d'Intégrité Fonctionnelle )**

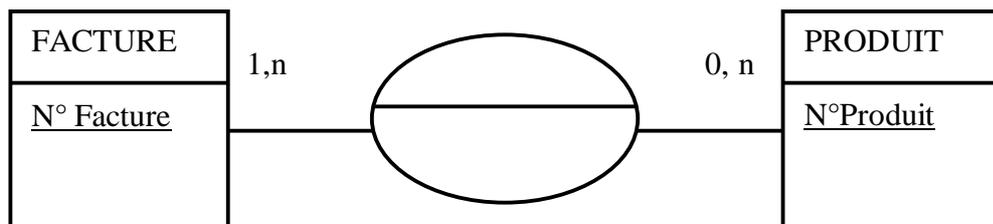
→ 1, 1 sur une des pattes de relations



**Garantie d'existence**

**2. Relation CIM ( Contrainte d'Intégrité Multiple )**

→ 0,  $\underbrace{\quad - 1, \quad}$  n au maximum



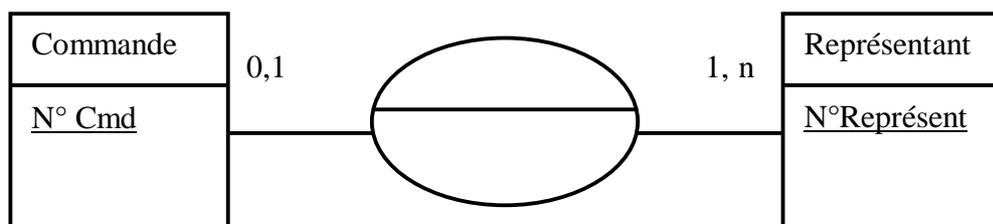
Seule une relation CIM peut comporter des propriétés

C'est le couple ( N° Facture, N° Produit ) → Quantité  
DF

Exemple de propriétés : % de remise sur un produit car il concerne une ligne de facture

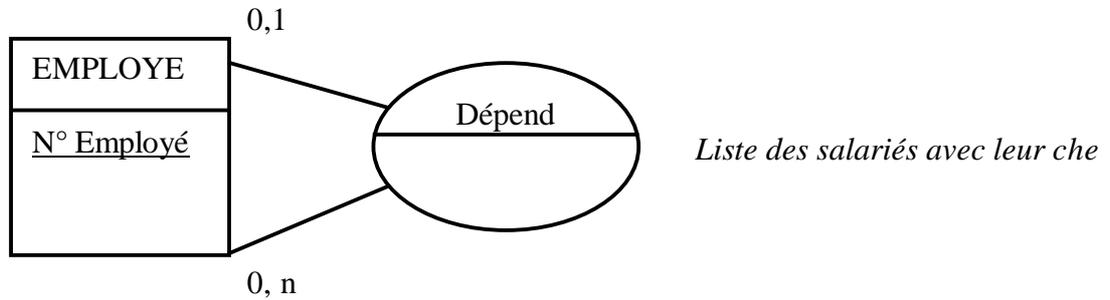
**3. Relation ALT ( ALTernative )**

→ 0, 1 - 0,  $\underbrace{\quad}$  0 au minimum



**4. Réflexivité**

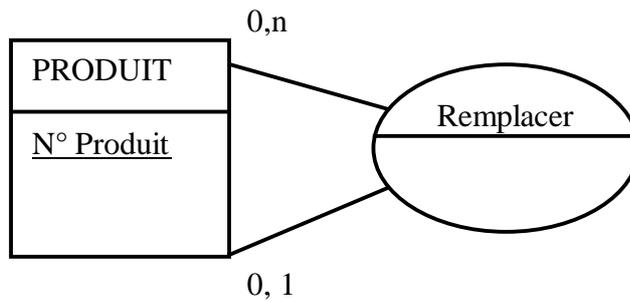
Un employé dépend de personne ou d'un responsable



Un salarié peut avoir soit personne soit plusieurs subordonnés

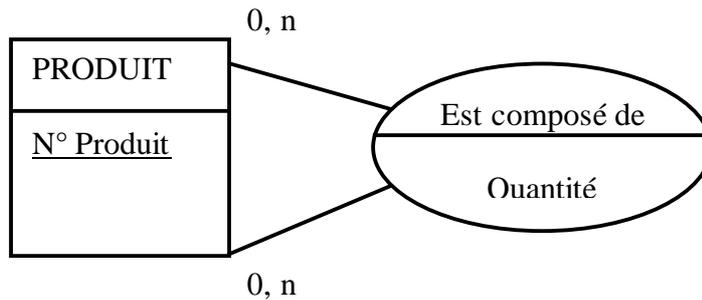
Exemple :

Un produit peut remplacer aucun ou plusieurs



Un produit est remplacé par 0 ou 1 principa

Exemple :

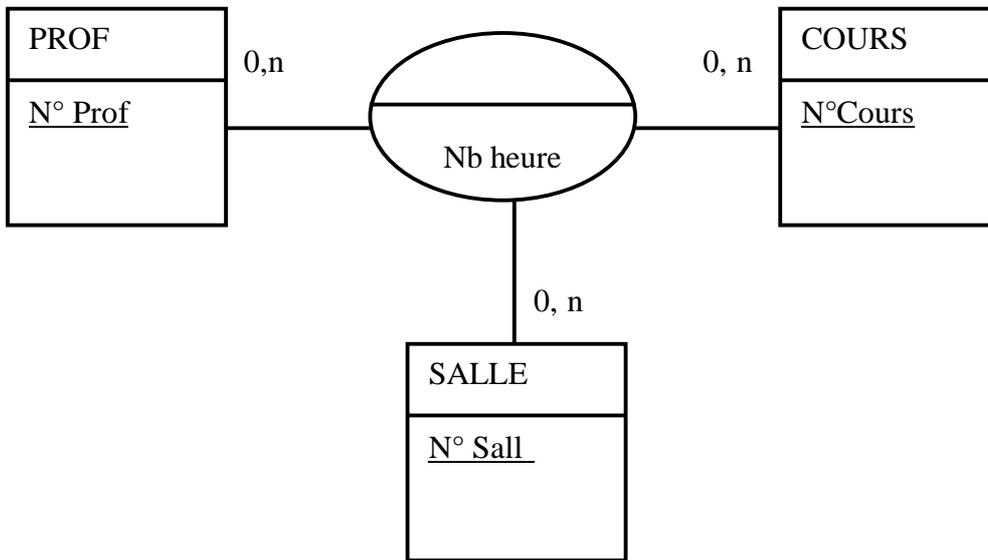


Comme c'est une CIM, on peut mettre une propriét

	Sous-produit			
Produit	P1	P2	P3	
P1			5	2
P2				
P3				

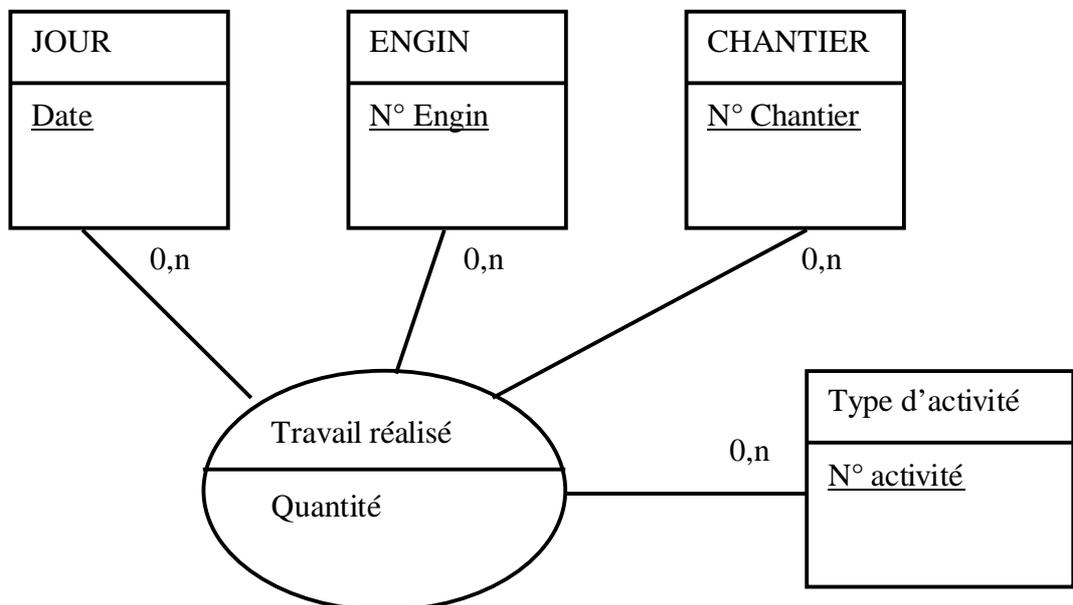
**5. Relation n-aires**

Lorsque l'on met en œuvre dans une relation plus de deux entités



N° Prof ,N° Cours ,N° Sall → Nb Heure  
DF

Exemple : Gestion de chantier



Exemple d'occurrence de la relation

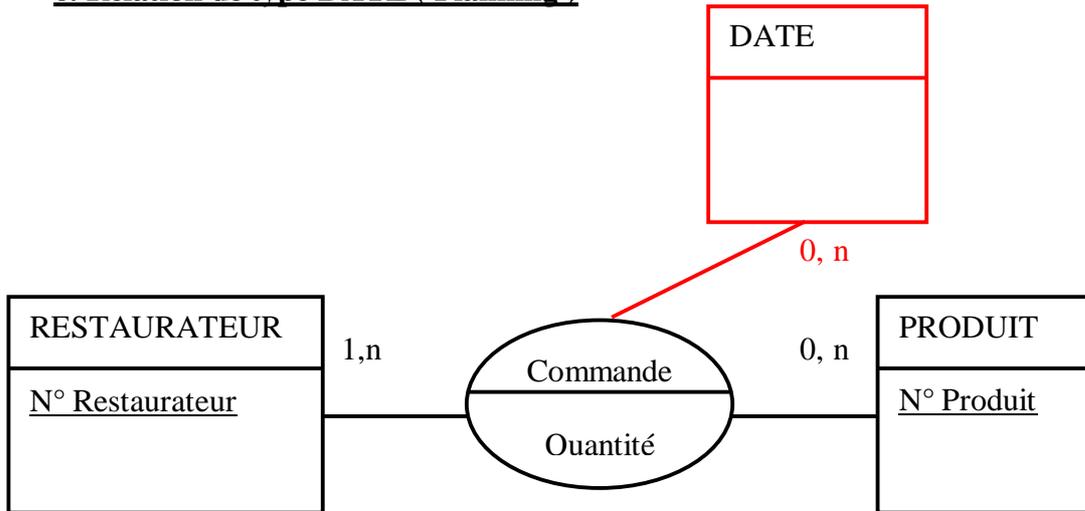
Le **15/01/2001**, le bulldozer **F 324** a réalisé sur un chantier **C034**, u **déblaiement** de **300 m<sup>3</sup>**.

DF entre les quatre éléments et la quantité

Date, N°Engin, N° Chantier, N°Activité → Quantité  
DF



**6. Relation de type DATE ( Planning )**



→ Date d'acha ?

- En propriété de la relation ?

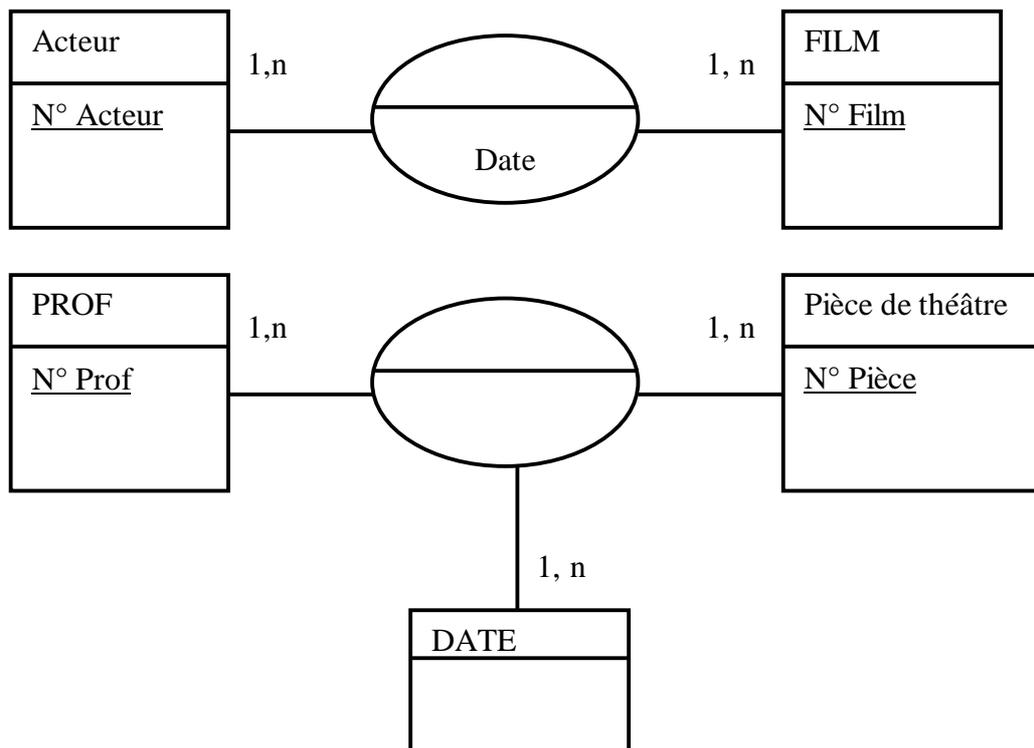
N° Restaurateur, N° Produit → Date  
DF

⇒ Un restaurateur achète un produit à une date.

Pour représenter la liste des achats d'un même produit par le restaurateur à différentes dates ( Historique ) On est obligé d'ajouter une **entité date**.

N° Restaurateur, N° Produit, Date → Quantité  
DF

Exemple :



## 7. Héritage

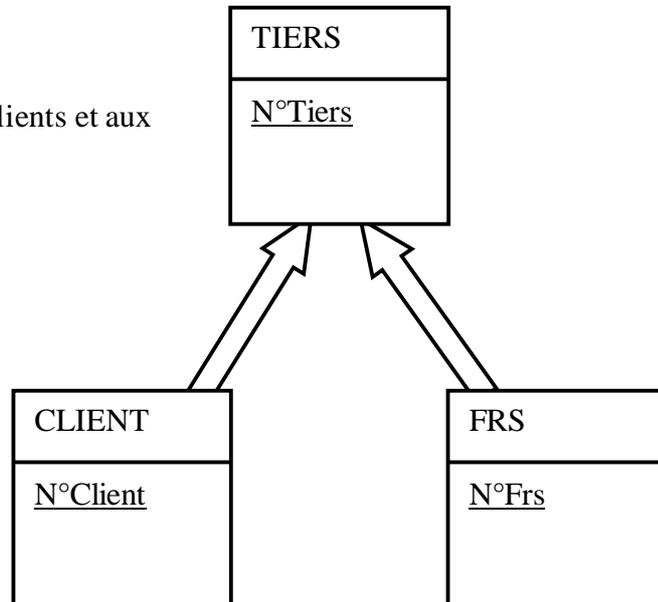
Société qui travaille avec des tiers ( Clients , fournisseurs )

### **SUR TYPE**

( Caractéristique commune aux clients et aux fournisseurs )

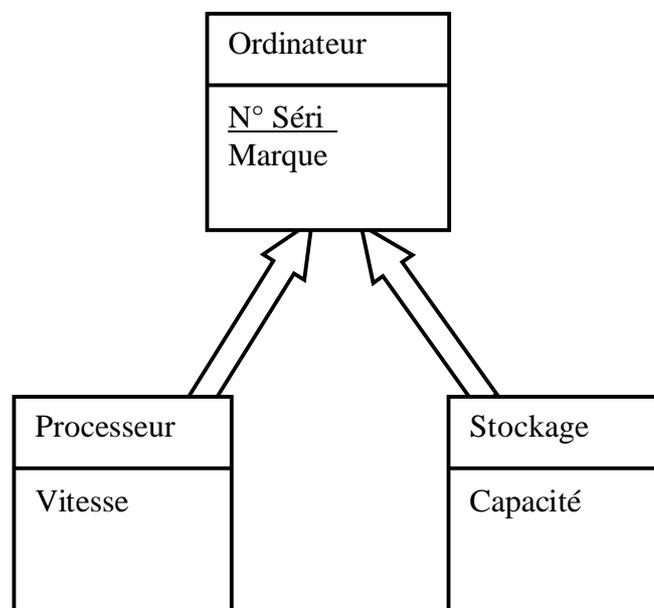
### **SOUS TYPE**

( Caractéristiques spécifiques à chaque objet )

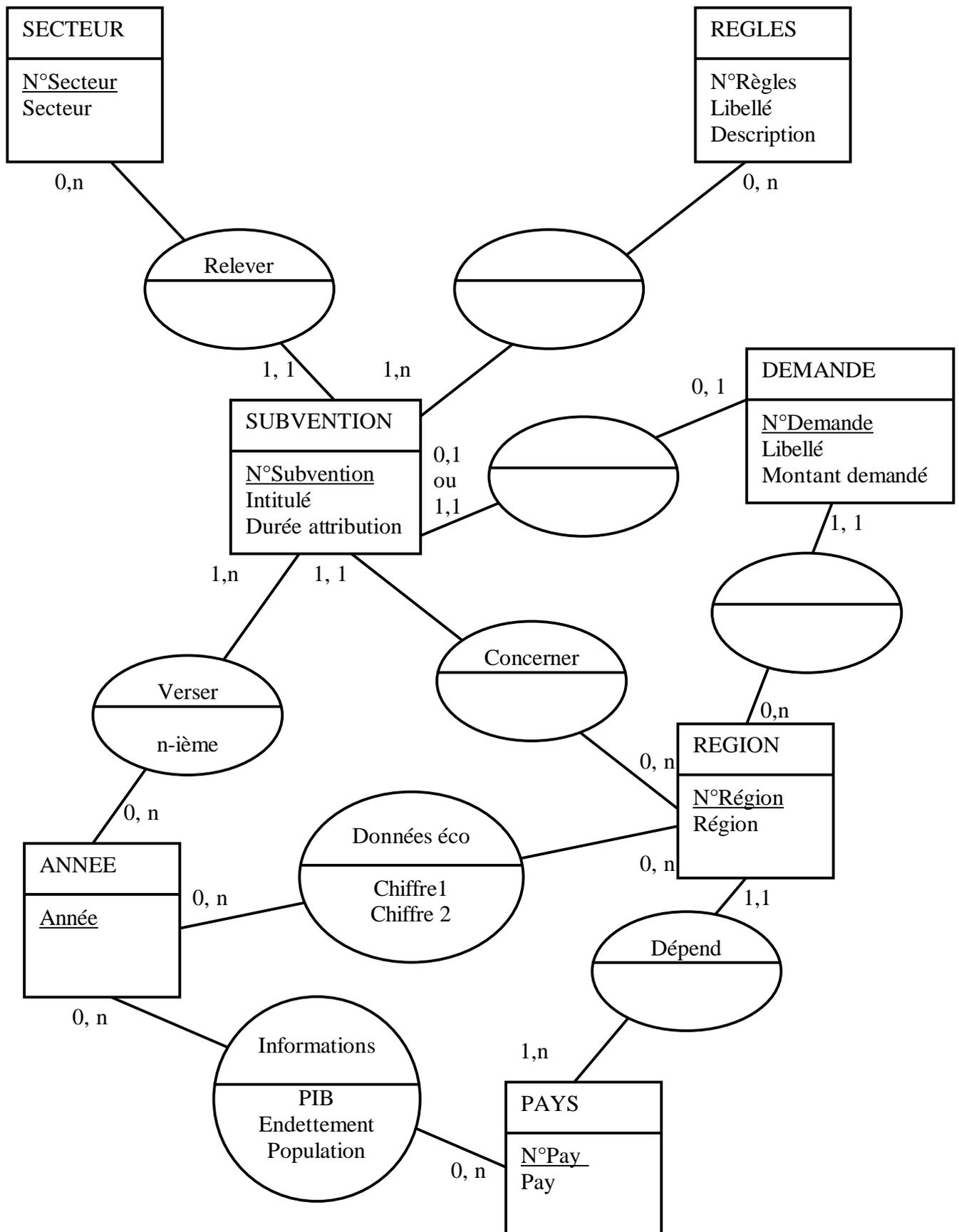


Les sous-types héritent des propriétés du sur-type.

Exemple :



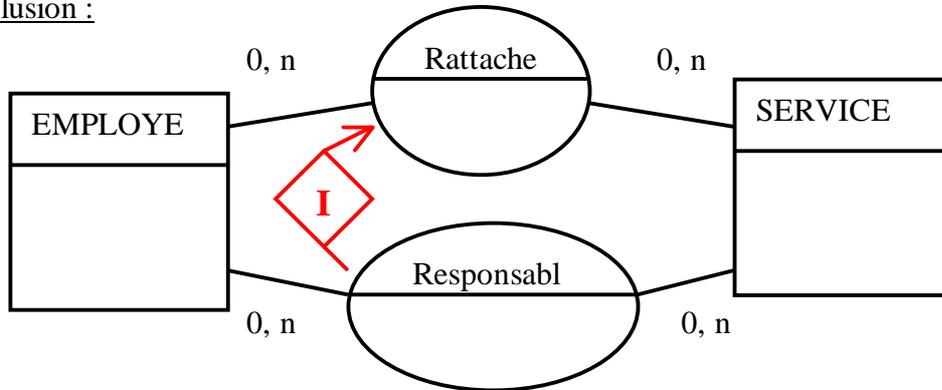
**Correction Contrôle n°1 :**



**Extension du MCD ( MCD analytique )**

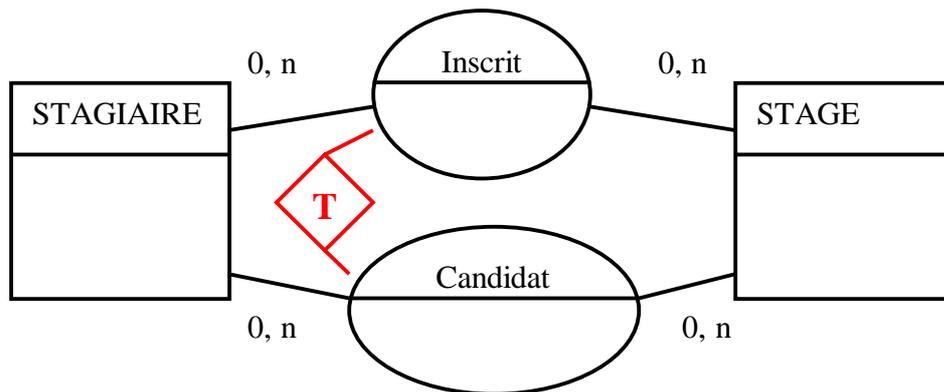
1. Héritage
2. Contraintes ensemblistes

a. Inclusion :



Un employé ne peut pas être responsable d'un service s'il n'est pas rattaché à ce service.

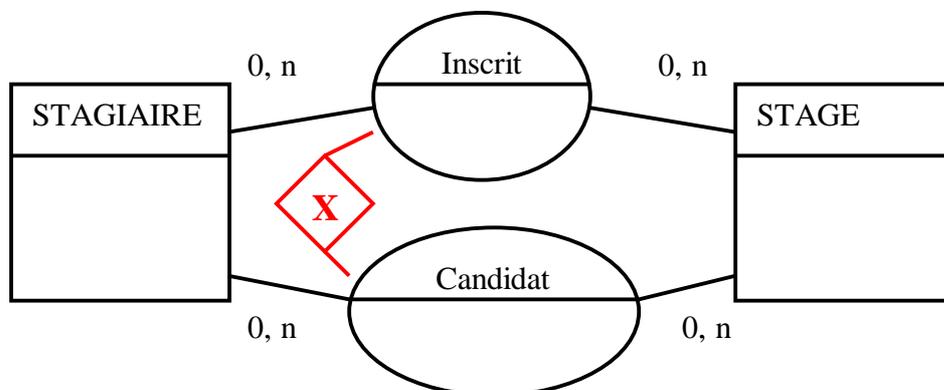
b. Totalité ( OU inclusif )



Un stagiaire es obligatoirement :

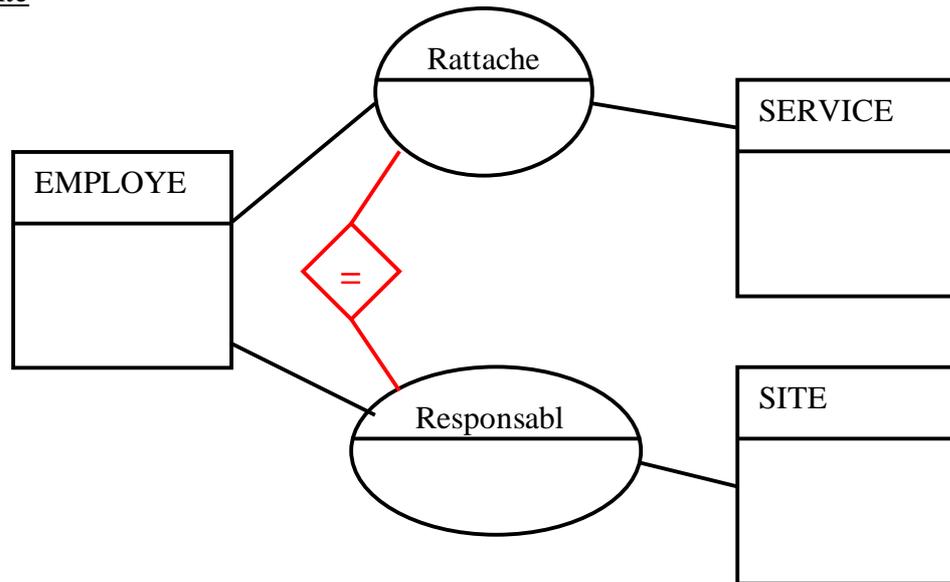
- soit inscrit
- soit candida
- ou les deux

c. Exclusion



Un stagiaire es soit inscrit, soit candidat mais jamais les deux

d. Egalité

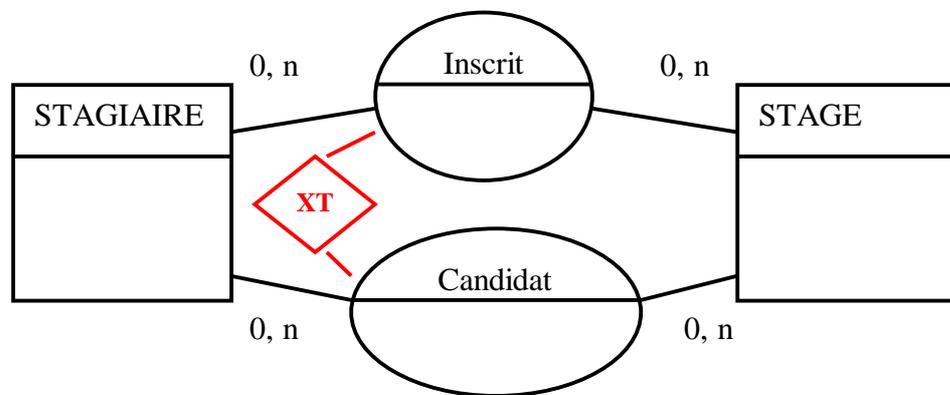


- Un employé rattaché à un service dépend d'un site
- Un employé dépendant d'un site est rattaché à un service

e. Partition (OU Excl sif)

Totalité + Exclusion

1 des 2 ou les 2 l'un ou l'autre



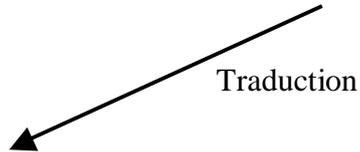
( On note XT ou + dans le losange )

Un stagiaire es obligatoirement soit inscrit, soit candidat, jamais deux et jamais aucun.

## 1 Cadre de Travail :

Problème

Représentation du réel perçu ⇒ Création du MCD ( Objet et Relation )

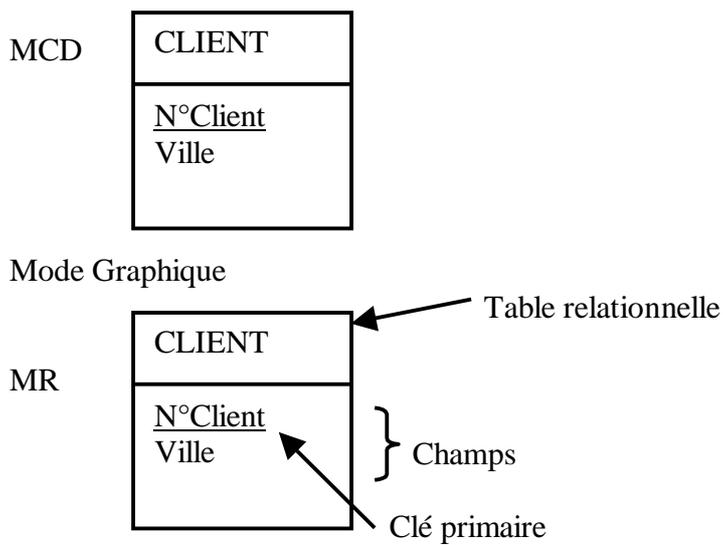


Modèle Relationnelle ( Modèle Logique de Données )  
( Mise en œuvre dans un Système Informatique )

## 2 Règles de Traduction :

Support : Feuille Gestion de Commande

### Règle 1 : Entité

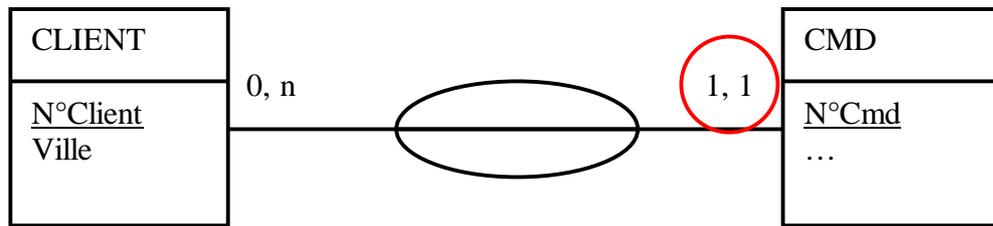


Mode Textuel

CLIENT {N°Client, Ville }

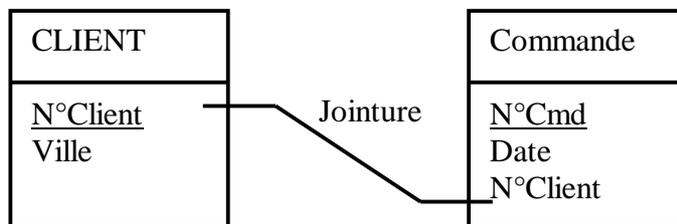
**Règle 2 : Traduction d'une CIF( 1,1 - \*-\*)**

Exemple : COMMANDE et CLIENT



CLIENT { N°Client, Ville }  
 COMMANDE { N°Commande, Date, #N°Client }

Clé étrangère

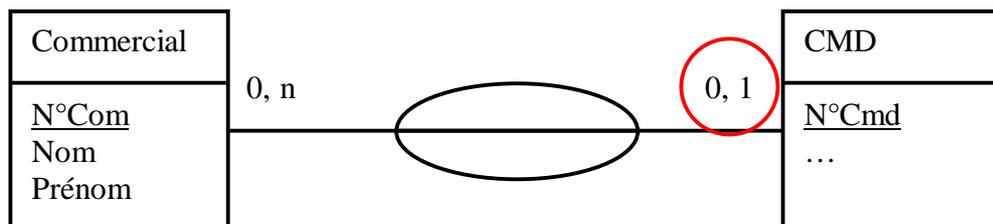


**Remarque :**

**L'intégrité référentielle :**

**on doit associer un client à une commande au niveau physique**

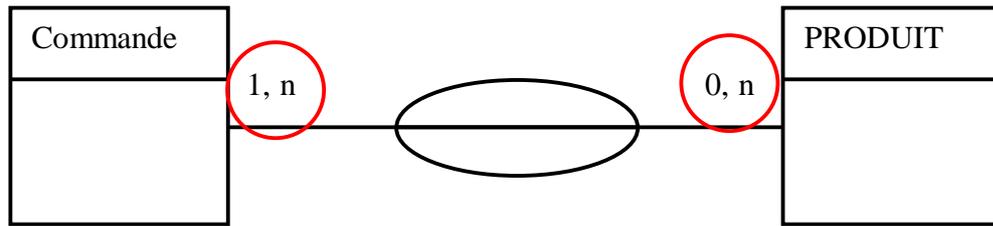
**Règle 3 : Traduction d'une relation alternative ( 0,1 - \*,\*)**



COMMERCIAL { N°Commercial, Nom, Prénom }  
 COMMANDE [ N°Commande, Date, #N°Commercial ]

**Remarque : Il n'y a pas d'intégrité référentielle au niveau physique**

**Règle 4 : Traduction d'une CIM (\*,n - \*,n)**



**4.1 Chaque entité devient une table relationnelle**

COMMANDE { N°Commande, Date, ... }

PRODUIT { N°Produit, Libellé, Prix, ... }

**4.2 L'association devient une table relationnelle**

Ligne\_Commande { #N°Commande, #N°Produit, Quantité }

Les Clés étrangères sont les clés primaires des deux tables liées

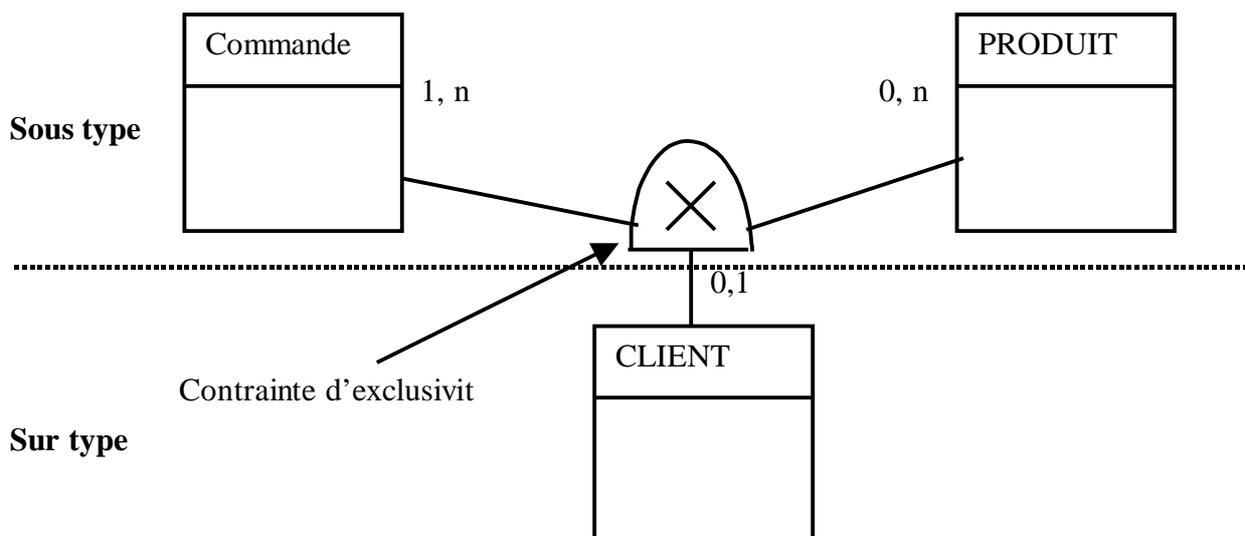
**4.3 La table relationnelle prend, s'il y en a, les propriétés de l'association**

**4.4 La clé primaire de la table est composée des deux clés étrangères rapatriées**

Autre forme pour la table :

Ligne\_Commande { N°Ligne, Quantité, #N°Commande, #N°Produit }

**5 Héritage :**



CLIENT { N°Client, Ville }

PARTICULIER { N°Client, Date Naissance, Type Règle }

ENTREPRISE { N°Client, Société, Remise, Condition }

Les sous-types possèdent la même clé primaire que le sur type.

**MLD Complet :**

PARTICULIER { N°Client, Date, Type Règlement }  
 ENTREPRISE { N°Client, Société, Remise, Condition Règle ent }  
 CLIENT { N°Client, Ville }  
 COMMANDE { N°Commande, Date, #N°Client, #N°Commercial }  
 PRODUIT { N°Produit, Libellé, Prix }  
 LIGNE\_COMMANDE { #N°Commande, # N°Produit, Prix }  
 COMMERCIAL { N°Commercial, Nom, Prénom }

**Cas IMI :**

EMPLOYES { CodeEmployés, Nom employé,..., # NumService }  
 REALISATION { #CodeEmployés, #CodeProjet, NbHeures }  
 PROJET{ CodeProjet, NomProjet,..., #NumService }  
 SERVICE { NumService, NomService }