

« La structure de toute “chose”, qu'il s'agisse d'un langage, d'une maison, d'une machine etc., se résume à des relations. » — Alfred Korzybski

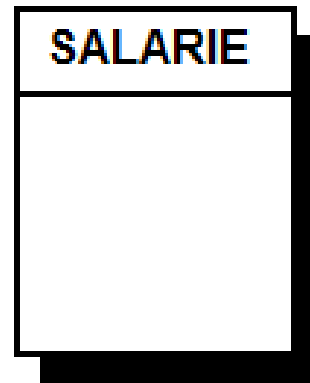
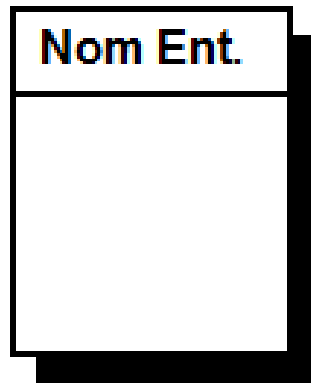
[http://fr.wikipedia.org/wiki/Alfred\\_Korzybski](http://fr.wikipedia.org/wiki/Alfred_Korzybski)

Labor omnia vincit improbus

Un travail opiniâtre vient à bout de  
tout

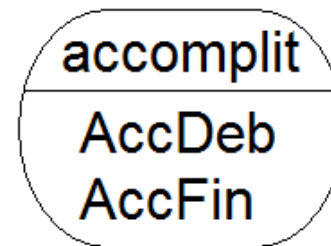
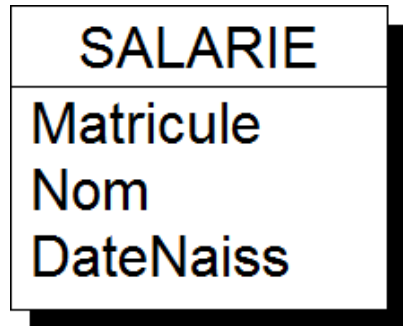
# DEFINITION

- ENTITE : une entité est la représentation dans le SI d'un objet matériel ou immatériel ayant une existence propre et conforme aux choix de gestion de l'entreprise.



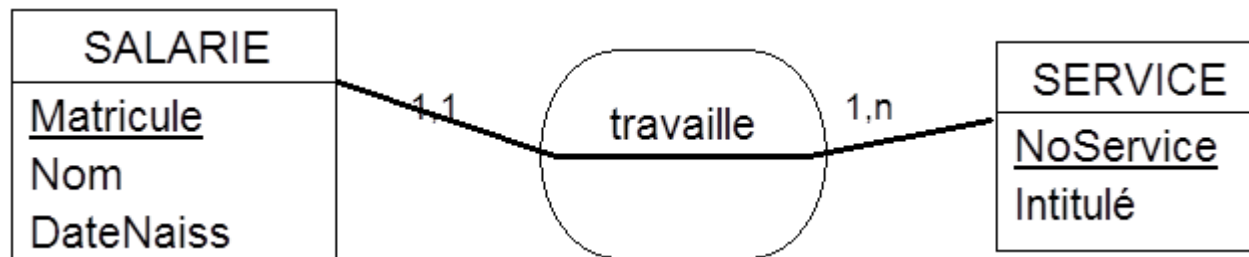
# DEFINITION II

- **PROPRIETE** : une propriété (ou attribut) est une donnée élémentaire que l'on peut rattacher à une entité ou à une association. Le nom de chaque propriété peut être inscrit à l'intérieur de chaque entité ou association lorsque celle ci est porteuse d'attributs.



# DEFINITION III

- **ASSOCIATION** : une association est la prise en charge par le SI d'une relation (un lien) entre des entités.
- Le lien entre une entité et une association est porteur de CARDINALITES :
- **CARDINALITÉ MINI** : c'est le nombre de fois minimal qu'une occurrence d'une entité participe aux occurrences de l'association. Elle ne peut être qu'égale à 0 ou 1 ( sauf exception ). Cette cardinalité est également appelée cardinalité de contrainte.
- **CARDINALITÉ MAXI** : c'est le nombre de fois maximal qu'une occurrence d'une entité participe aux occurrences de l'association. Elle ne peut être qu'égale à 1 ou n ( sauf exception ).



# IDENTIFIANT d'une entité

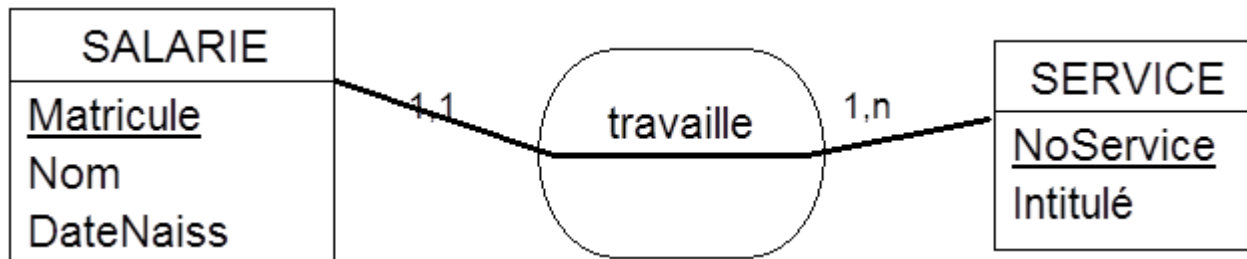
- Règle 1 : toute entité doit posséder un identifiant
- Règle 2 : toutes les propriétés d'une entité ou d'une association doivent être élémentaires, c'est à dire non décomposables
- Règle 3 : pour chaque occurrence d'entité ou d'association chaque propriété ne peut prendre qu'une seule valeur. Autrement dit, on ne peut avoir de valeurs répétitives (multiples) pour une même propriété
- Règle 4 : toutes les propriétés autres que l'identifiant doivent dépendre pleinement de l'identifiant et non d'une partie de celui-ci
- Règle 5 : chaque propriété doit dépendre directement de l'identifiant et non par l'intermédiaire d'une ou plusieurs autres propriétés.

# On distingue plusieurs catégories d'association :

- 1 - 1 : à chaque occurrence de X correspond une seule occurrence de Y
- 1 - n (1 à plusieurs) A chaque occurrence de X correspond une ou plusieurs occurrences de Y et à chaque occurrence de Y correspond une seule occurrence de X.
- n - m (plusieurs à plusieurs) A chaque occurrence de X correspond une ou plusieurs occurrences de Y et réciproquement.

# RELATION 1,N

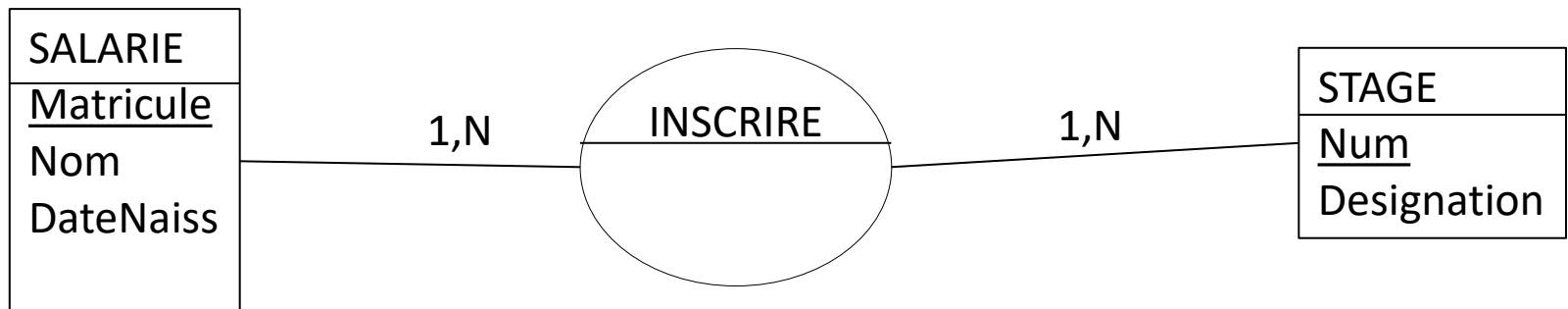
- 1 - n (1 à plusieurs) A chaque occurrence de X correspond une ou plusieurs occurrences de Y et à chaque occurrence de Y correspond une seule occurrence de X.
  - Un salarié travaille dans un ou plusieurs services
  - Dans un service travail au moins un salarié





# RELATION N,N

- n - m (plusieurs à plusieurs) A chaque occurrence de X correspond une ou plusieurs occurrences de Y et réciproquement.
  - Un salarié est inscrit dans au moins un stage
  - Dans un stage est inscrit au moins un salarié



# Modèle relationnel

- Dans ce modèle, les données sont représentées par des tables, sans préjuger de la façon dont les informations sont stockées dans la machine.

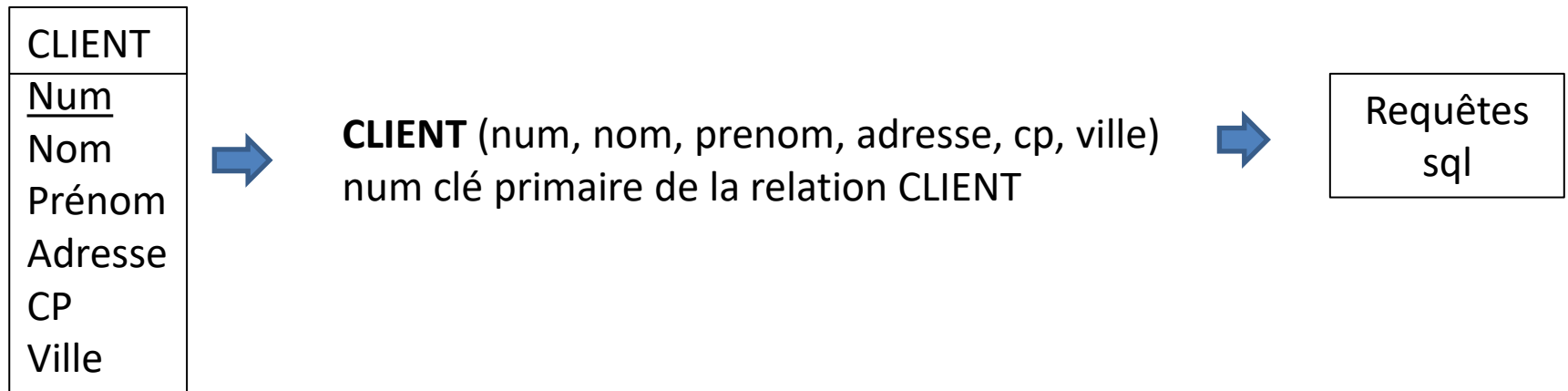
# **Passage du MCD au MRD**

MCD modèle conceptuel de données

MRD modèle relationnel de données

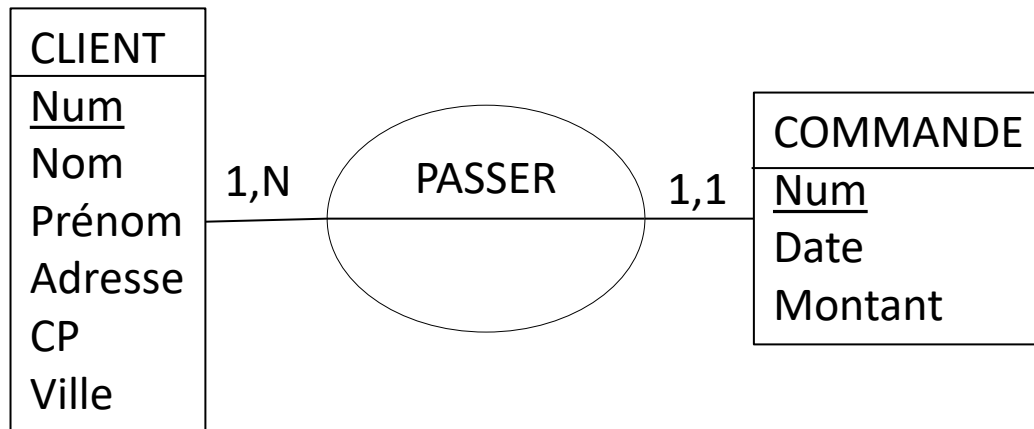
# Règle 1

- **Toute entité devient une relation** ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.



# Règle 2

- Toute association hiérarchique (de type  $[1, n]$ ) se traduit par une clé étrangère.
- La clé primaire correspondant à l'entité père (côté  $n$ ) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1).

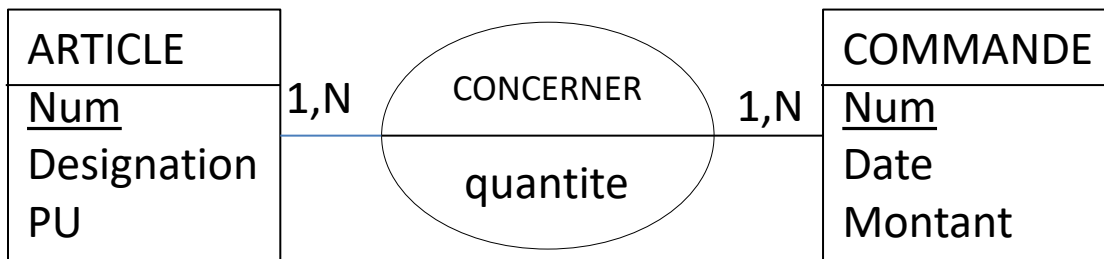


**COMMANDE**(Num, date, montant, Numclient)

- Num cle primaire de la relation COMMANDE
- Numclient clé étrangère de la relation COMMANDE en référence à Num de la relation CLIENT

# Règle 3

- **Toute association non hiérarchique** (de type  $[n, n]$  ou de dimension  $> 2$ ) **devient une relation**.
- La clé primaire est formée par la concaténation (juxtaposition) de l'ensemble des identifiants des entités reliées.
- Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.



**CONCERNER**(NumA, NumC, quantite)

- NumA, NumC clés primaires de la relation CONCERNER
- NumA clé étrangère de la relation CONCERNER en référence à Num de la relation ARTICLE
- NumC clé étrangère de la relation CONCERNER en référence à Num de la relation COMMANDE

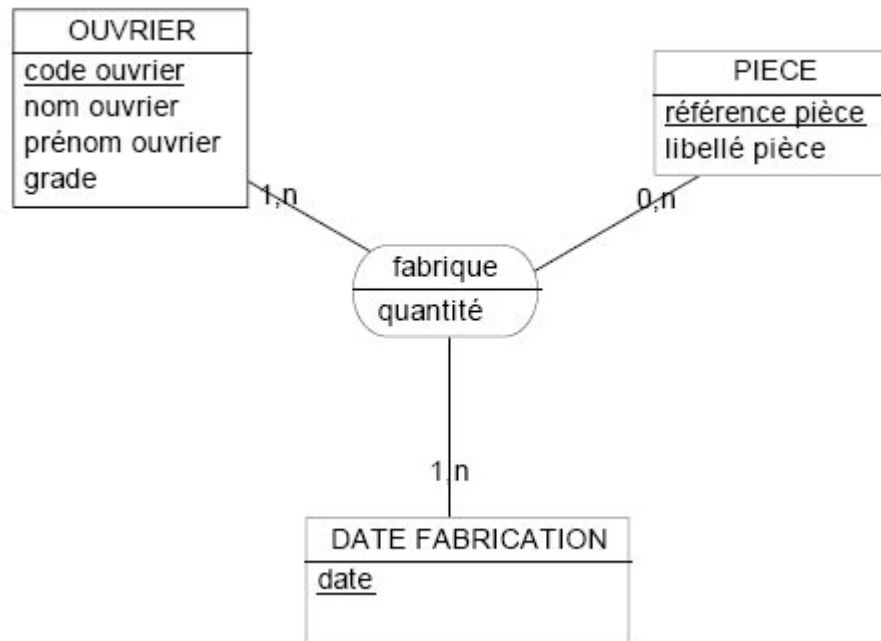
# Règles de passage - Résumé

- Chaque type-entité donne naissance à une relation.
  - Chaque attribut de ce type-entité devient un attribut de la relation.
  - L'identifiant est conservé en tant que clé de la relation.
- Chaque type-association (n,n) dont aucune patte n'a pour cardinalité maximale 1 donne naissance à une relation.
  - Chaque attribut de ce type-association devient un attribut de la relation.
  - L'identifiant, s'il est précisé, est conservé en tant que clé de la relation,
  - sinon cette clé est formée par la concaténation des identifiants des type-entités qui interviennent dans le type-association.
- Un type-association (1,n) dont au moins une patte a une cardinalité maximale à 1 (ce type-association devrait être binaire et n'a généralement pas d'attribut) ne devient pas une relation.
  - La relation correspondant au type-entité dont la patte vers le type-association a une cardinalité maximale valant 1, se voit simplement ajouter comme attribut (et donc comme clé étrangère) l'identifiant de l'autre type-entité.

# **LES CAS PARTICULIERS**



# I. La ternaire



Avec ce modèle, on mémorise chaque jour pour chaque ouvrier les pièces qu'il a fabriqué et en quelle quantité.

# MLD

- **FABRIQUE** (code\_ouvrier, référence\_pièce, date, quantité)
  - code\_ouvrier, référence\_pièce, date clés primaires de la relation FABRIQUE
  - code\_ouvrier clé étrangère de la relation FABRIQUE en référence à code\_ouvrier de la relation OUVRIER
  - référence\_pièce clé étrangère de la relation FABRIQUE en référence à référence\_pièce de la relation PIECE
  - date clé étrangère de la relation FABRIQUE en référence à date de la relation DATE FABRICATION

# Occurrences

- Une occurrence de l'association fabrique implique une seule occurrence de chaque entité.

- Exemple :

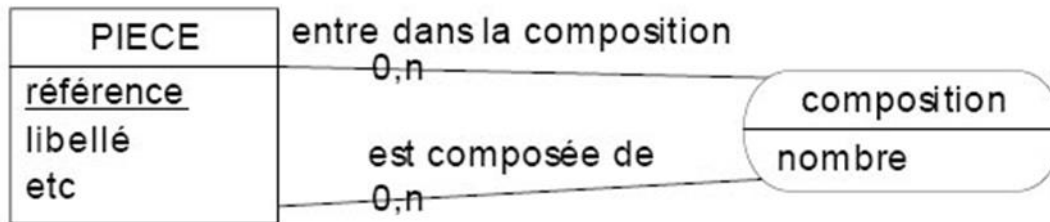
	Ouvrier	Pièce	Date	quantité
•	Dupont	1	17/02	50
•	Dubois	2	17/02	100
•	Martin	1	17/02	40
•	Dupont	3	17/02	55
•	Dubois	3	17/02	20
•	Dupont	1	16/02	40
•	Martin	1	16/02	40

- Quelle relation peut-on définir entre ouvrier et pièces ?
  - Un ouvrier fabrique au moins une pièce
- Quelle relation peut-on définir entre pièces et ouvrier ?
  - Une pièce est fabriquée par au moins un ouvrier
- A quelles conditions un ouvrier peut il fabriquer la même pièce ?
  - A partir du moment où elle est fabriquée à une période différente

## II. Les associations réflexives

- Une association réflexive est une association reliant des occurrences de la même entité. Ces associations sont quasiment toujours binaire (2 branches).
- Pour lire une association réflexive, il est faut connaître le rôle attribué à chaque branche de l'association, c'est à dire le rôle de chaque occurrence dans le lien.

# Réflexive non hiérarchique



## Lecture de l'association

Une pièce entre dans la composition de 0 à plusieurs autres pièces.

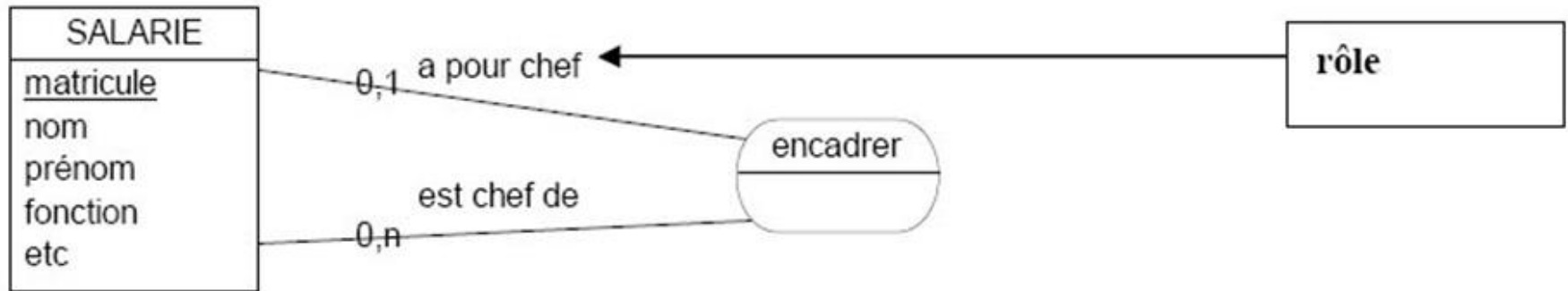
Une pièce peut être composée de plusieurs autres pièces.

Une pièce entre dans la composition d'une autre un certain nombre de fois.

# MLD

- **COMPOSITION** (référence composition, référence composée, nombre)
  - référence composition, référence composée clés primaires de la relation COMPOSITION
  - référence composition clé étrangère de la relation COMPOSITION en référence à référence de la relation COMPOSITION

# Réflexive hiérarchique



## Lecture de l'association

Un salarié a pour chef aucun ou un seul autre salarié.

Un salarié est chef d'aucun à plusieurs autre(s) salarié.

# MLD

- **SALARIE** (matricule, nom, prénom, fonction,... , matricule\_chef)
  - matricule clés primaires de la relation SALARIE
  - matricule\_chef clé étrangère de la relation SALARIE en référence à matricule de la relation SALARIE



# III. Les entités faibles

- Une entité faible est une entité dont les occurrences ne peuvent exister que si les occurrences d'une autre entité existent (contrainte d'identification).

# Occurrences

N°Chantier	N°Facture
001	1
001	2
001	3
002	1
003	1

Le numéro de  
facture  
recommence à 1  
à chaque  
changement de  
chantier.

# Solution Merise 2



Les parenthèses entourant les cardinalités 1,1 indiquent que :

N° Facture est un identifiant relatif.

L'identifiant complet de l'entité FACTURE est la concaténation de :

N°Chantier et N°Facture.

# MLD

**CHANTIER (N°Chantier,AdrChantier, NomChef)**

N°Chantier clé primaire de la relation CHANTIER

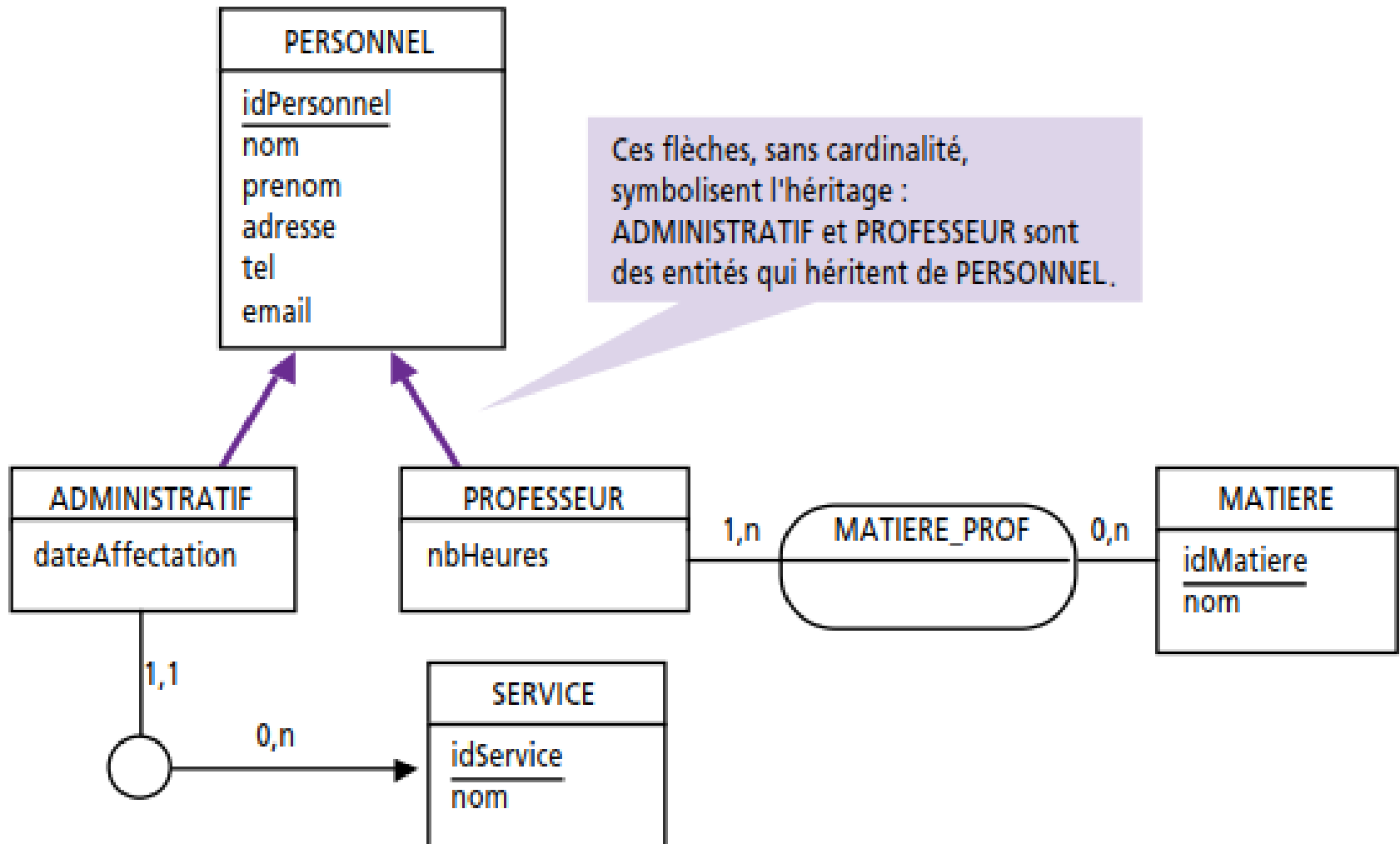
**FACTURE (N° Chantier, N°Facture, DateFacture)**

N°Facture : clé primaire de la relation FACTURE

N°Chantier clé primaire de la relation FACTURE

N°Chantier clé étrangère de la relation FACTURE en référence à  
N°Chantier de la relation CHANTIER

# IV. Heritage



# MLD

PERSONNEL (idPersonnel, nom, prenom, adresse, tel, email)

idPersonnel : clé primaire

ADMINISTRATIF (idPersonnel, dateAffectation, idService)

idPersonnel : clé primaire

idPersonnel : clé étrangère en réf. à idPersonnel de PERSONNEL

idService : clé étrangère en réf. à idService de SERVICE

PROFESSEUR (idPersonnel, nbHeures)

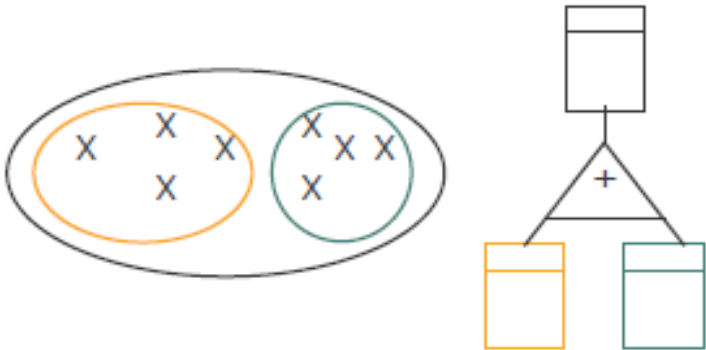
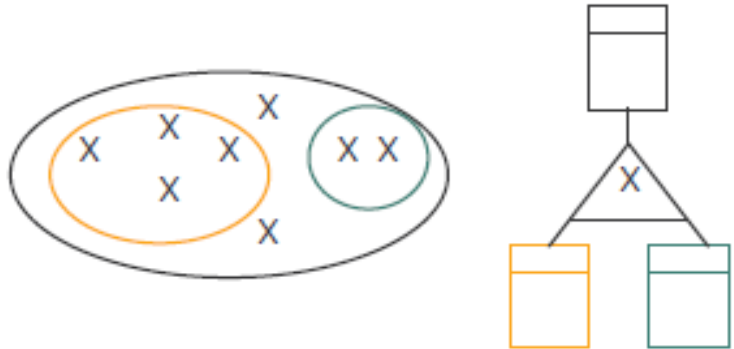
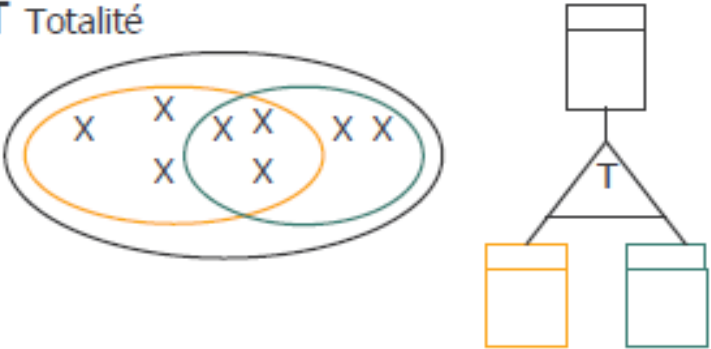
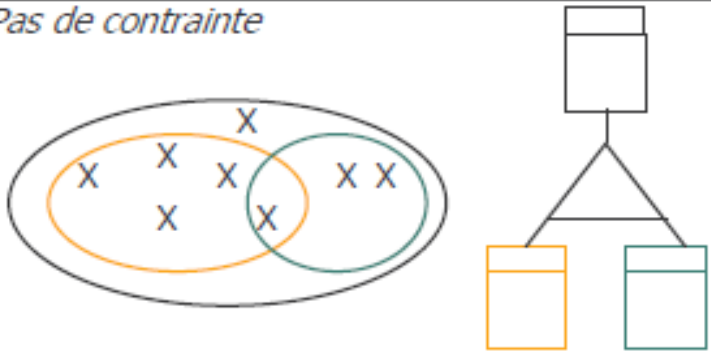
idPersonnel : clé primaire

idPersonnel : clé étrangère en réf. à idPersonnel de PERSONNEL

# Autres solutions

- Migration vers l'entité mère
  - Si les entités filles contiennent peu d'informations et ne possèdent aucun lien (associations reliées à elles), alors il est préférable de les supprimer et de migrer leurs attributs vers l'entité mère. Certains attributs resteront parfois vides mais il n'y aura qu'une table à gérer
- Migration vers les entités filles
  - Si au contraire l'entité mère possède peu d'attributs et aucun lien (associations reliées à elle) et si les entités filles ont une activité plus importante (des attributs spécifiques et/ou des liens), alors il est préférable de migrer les attributs de l'entité mère vers les entités filles. Attention, cette solution ne peut être choisie que dans le cas où il n'est pas nécessaire de réaliser de traitements qui regroupent les entités filles.

# En d'autres termes :

	Couverture	Non couverture
Disjonction	<p><b>+</b> Partition (=exclusion+Totalité)</p> 	<p><b>X</b> Exclusion</p> 
Non disjonction	<p><b>T</b> Totalité</p> 	<p><i>Pas de contrainte</i></p>  <p>Il existe des éléments qui n'appartiennent à aucune sous entité et des éléments qui appartiennent à plusieurs sous entités.</p>

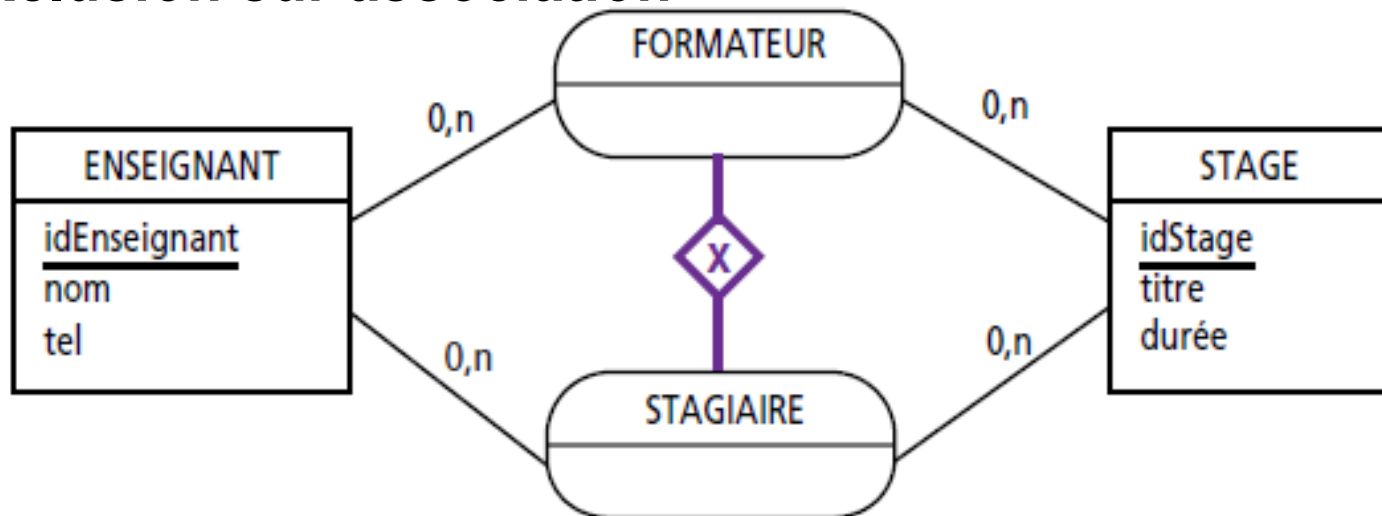


# Les contraintes sur les associations

- Pour vous donnez une idée de leur utilité, pensez aux informations qu'elles fournissent en cas d'insertion, ou de mise à jour

# Les contraintes sur associations(1/3)

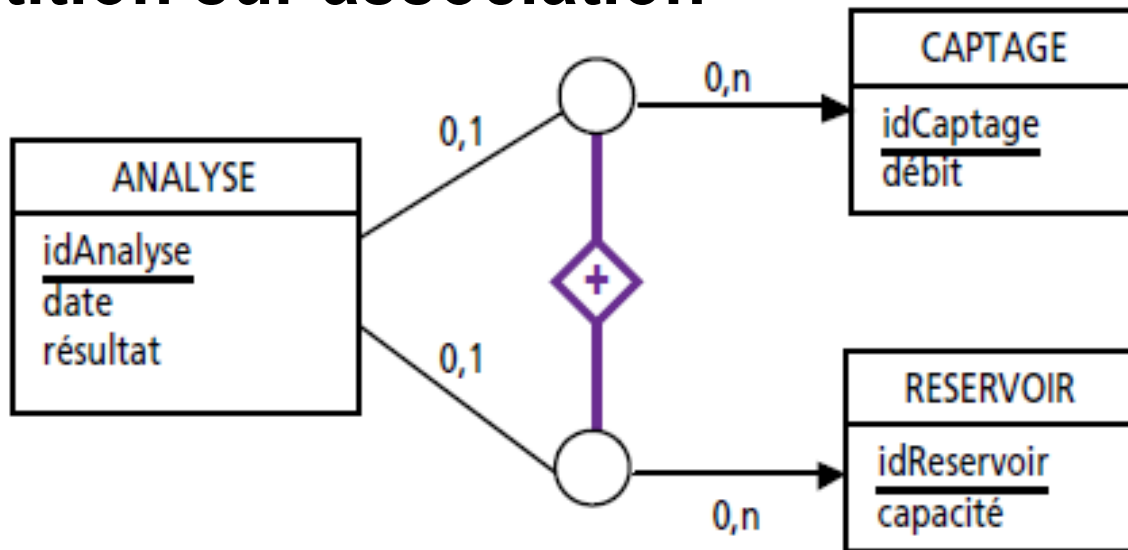
- **exclusion** = non(couverture) + disjonction
- **Exclusion sur association**



- Un enseignant peut participer à des stages. Il peut aussi être formateur de certains stages. Mais pour un même stage, il ne peut pas être à la fois stagiaire et formateur. Certains enseignants ne sont ni stagiaire, ni formateur pour certains stages.

# Les contraintes sur associations(2/3)

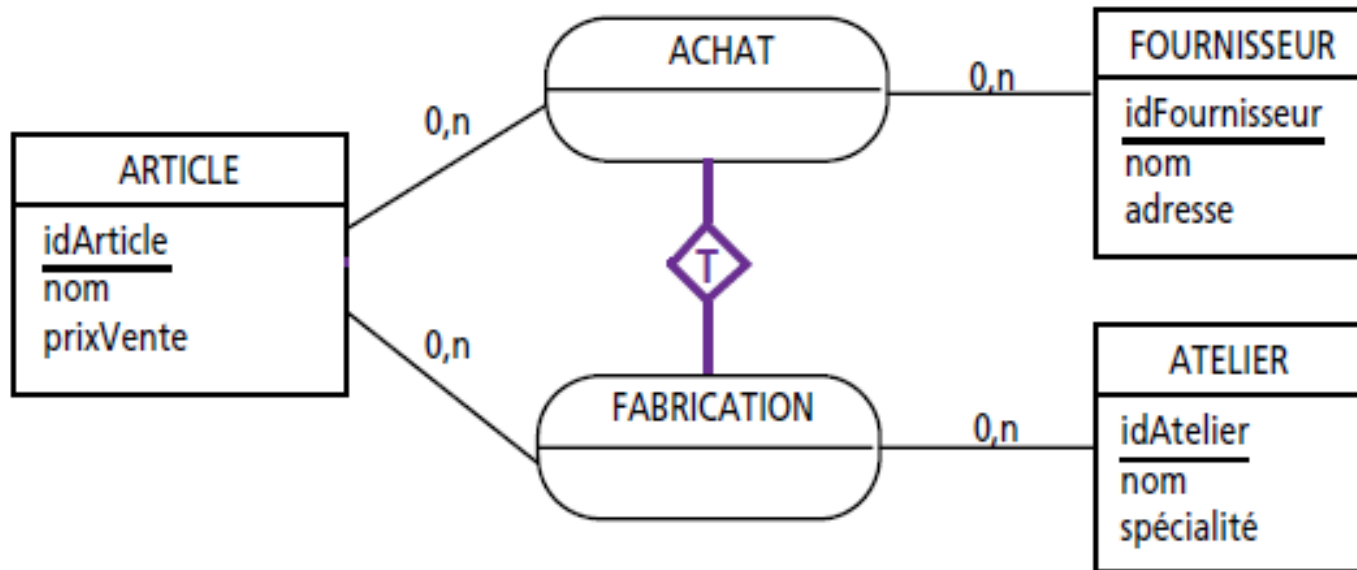
- partition = couverture + disjonction
- **Partition sur association**



- **idAnalyse** ne peut pas être à la fois présent dans l'une et l'autre CIF. Il est forcément présent dans l'une des 2."

# Les contraintes sur associations(3/3)

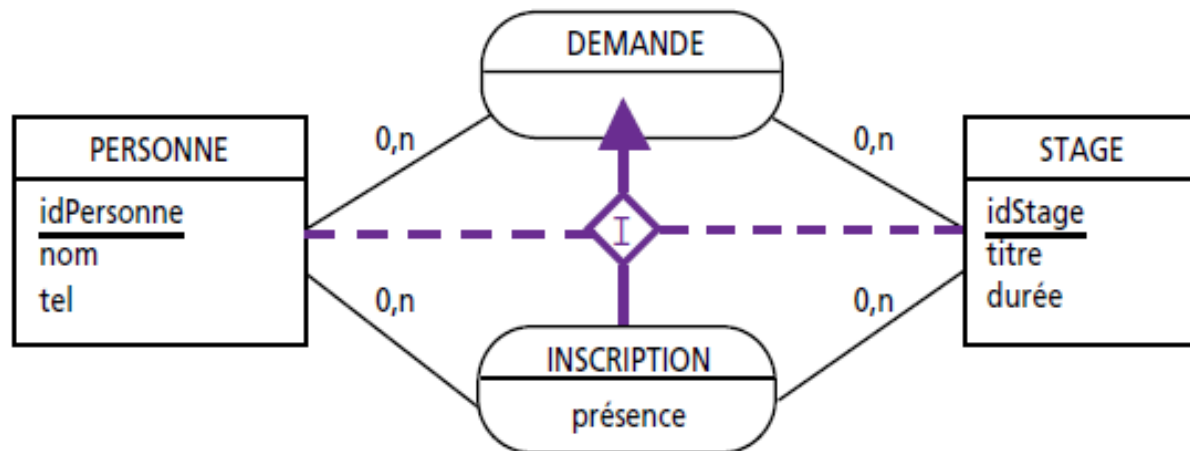
- totalité = couverture + non(disjonction)
- **Totalité sur association**



- *Un article est acheté auprès de fournisseurs ou fabriqué en ateliers. Il peut aussi être acheté auprès de fournisseurs puis être transformé en ateliers. Il n'y a pas d'autres possibilités.*

# L'inclusion : Exemple

- Une personne inscrite à un stage a dû en faire la demande préalable. Toutes les demandes de stage ne sont pas forcément satisfaites.

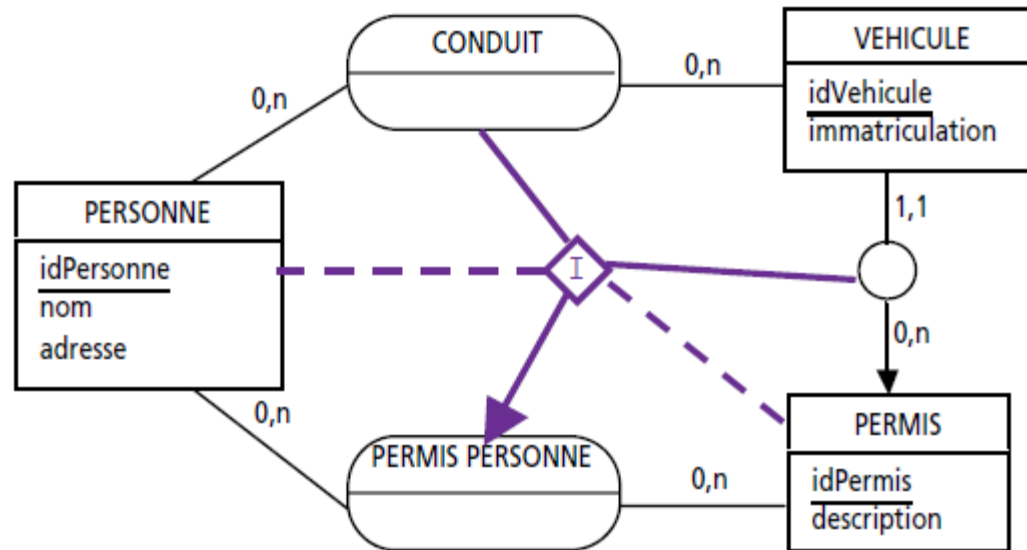


- La contrainte se traduit ainsi :

"Si le couple idPersonne+idStage est présent dans INSCRIPTION, il est forcément présent dans DEMANDE."

# Inclusion à portée multiple

- *Une personne peut conduire un véhicule que si elle possède le bon permis.*



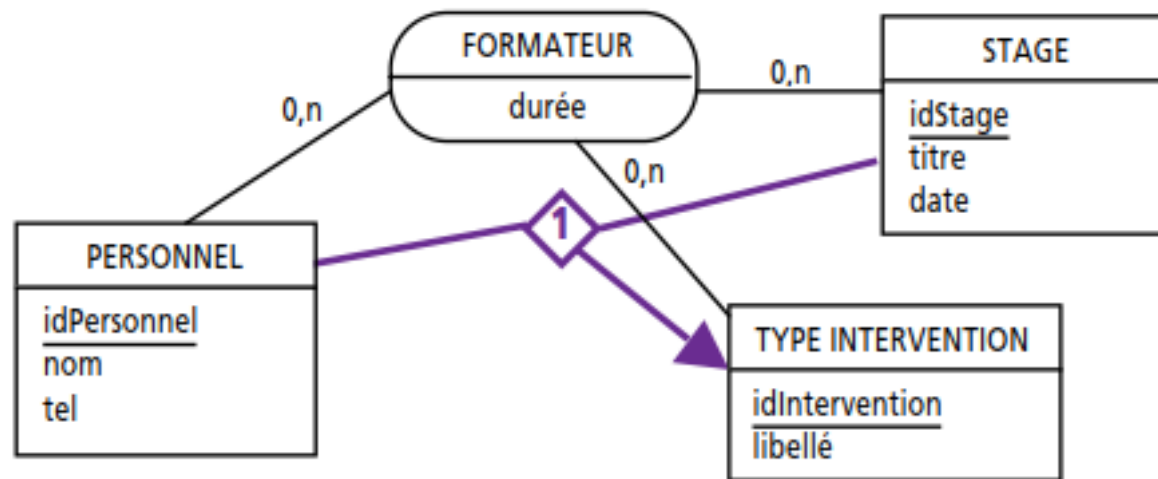
- La contrainte se traduit ainsi :

"Le couple idPersonne+idPermis formé en passant par CONDUIT et l'association entre VEHICULE ET PERMIS, doit être présent dans PERMIS\_PERSONNE."

# Unicité 2<sup>ème</sup> représentation

- Plusieurs personnels peuvent intervenir en qualité de formateur sur un stage. Pour chaque intervention, il faut connaître la durée en nombre d'heures et le type d'intervention

(cou  
prati

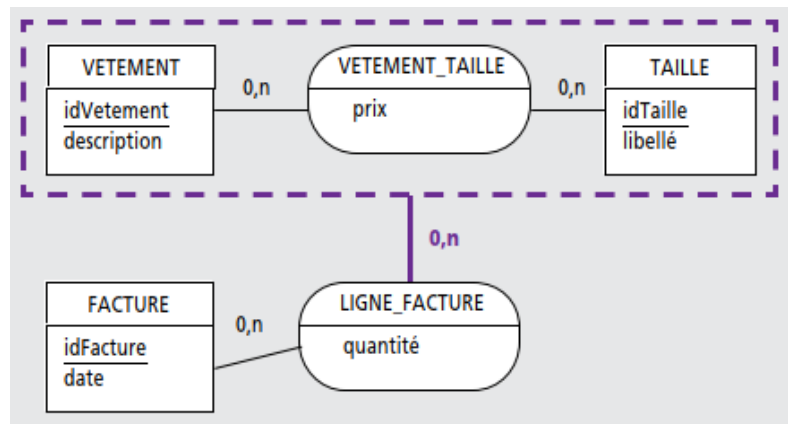


# Unicité

- Attention, au niveau relationnel, FORMATEUR n'est pas une ternaire (association à 3 liens directs) : elle ne contient en réalité que 2 attributs en clé primaire et la contrainte d'unicité permet de récupérer la troisième clé en clé étrangère uniquement.



# Agrégation - Au niveau relationnel, attention à la clé étrangère double :



- LIGNE\_FACTURE(idFacture, idVetement, idTaille, quantité)
- idFacture, idVetement, idTaille : clé primaire
- idFacture : clé étrangère en réf. à idFacture de FACTURE
- idVetement, idTaille : clé étrangère en réf. à (idVetement, idTaille) de VETEMENT\_TAILLE

# Résumé

Clés primaires	Clés étrangère	Merise
1	0	Entité / Héritage 1 entités
1	1	2 Entités (1,n) /Réflexive hiérarchique
2	2	2 Entités (m,n) /Réflexive non hiérarchique
2	0	Héritage 2 entités
2	1	Entité faible
3	2	Héritage 3 entités
3	3	Ternaire
2	3	Contrainte d'unicité