

3. La conception: Modèle relationnel

Le Modèle Conceptuel de Données (MCD)

- ❑ **Le niveau conceptuel** correspond à une **formalisation** du SI indépendante de toute contrainte d'organisation
- ❑ La formalisation des données, au niveau conceptuel, constitue **le MODELE CONCEPTUEL DES DONNEES (M.C.D)**

Objectifs:

- Rassembler les données (regrouper les données ayant des traits communs)
- Structurer les données qui seront utilisées par le S.I
- Décrire les données avec des outils (concepts, schémas)
- Modéliser
- Schématiser

Le Modèle Conceptuel de Données (MCD)

- ❑ Le M.C.D décrit la **sémantique** c'est à dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être faite.

- ❑ Le **M.C.D** est connu sous différentes dénominations, diagramme:
 - **entité-association**,
 - entité-relation (dénomination de l'ISO),
 - objet-relation.
 - individu-relation

Principes fondamentaux

- ❑ Le **MCD** a pour but de **décrire** le système d'information à l'aide **d'entités** et de **relations**.
- ❑ Il est à la base de tous les **SGBD dits relationnels** (Access, Oracle, DB2...) qui sont les plus utilisés actuellement dans les entreprises.
- ❑ Il permet donc une représentation du “ réel perçu ” sous la forme de :

ENTITES, **ASOOCIATIONS** et **PROPRIETES**.

Entité : définition

- Est **la représentation formelle** de la mémorisation d'une **information complexe et cohérente**.
- Sert à décrire plusieurs **occurrences** de la même information complexe.
 - Exemple :
 - une *Ford Kuga*, une *Renault Kadger* et une *Peugeot 3008* sont **trois occurrences** de la même **entité** qu'on peut appeler **voiture**.
- Les éléments de l'information (complexe) modélisée par l'entité sont les **propriétés**.
 - Exemple :
 - *Ford*, *Renaud* et *Peugeot* sont des « **marques** » ;
 - *Kuga*, *Kadger* et *3008* sont des « **modèles** ».
 - Ainsi, *Marque* et *Modèle* peuvent être des **propriétés** de l'entité **voiture**.
- Remarque : on l'appel aussi **Entité-Type**.

Entité : propriété

Une propriété est **un élément** d'une entité, et **d'une seule** :

- décrit la mémorisation **d'une information élémentaire**,
- a un nom **unique**,
- permet de mémoriser **une valeur**,
- doit avoir **un sens** (donc une valeur) pour chacune des **occurrences** de la composante → *Domaine de valeurs*.

Domaine= ensemble de valeurs atomiques que peut prendre un attribut.

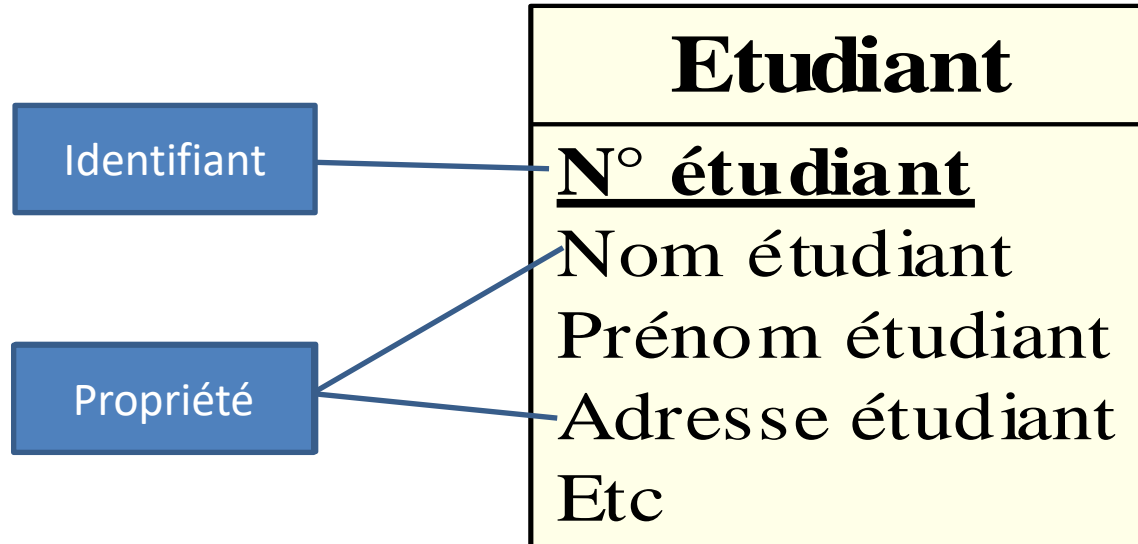
- Exemples de domaines:
 - nom : chaînes de caractères de longueur maximale 30
 - num : entiers compris entre 0 et 99999
 - couleur : {"bleu", "vert", "jaune"}
 - âge : entiers compris entre 16 et 65

Une propriété doit-elle être toujours renseignée ?

Entité : identifiant

- **L'identifiant** de l'entité est une **propriété qui ne peut pas prendre deux fois la même valeur** dans deux occurrences de l'entité.
- **C'est l'identifiant qui fait l'entité.**

Entité : représentation schématique



Entité : occurrences

- Pour une valeur de l'identifiant, on a une valeur de chacune des propriétés.
- Deux occurrences de l'entité ne peuvent avoir la même valeur d'identifiant.
- Les domaines de valeurs des propriétés ne sont pas disjoints.

Etudiant
<u>E0215632</u> Alami Soufiane Fès

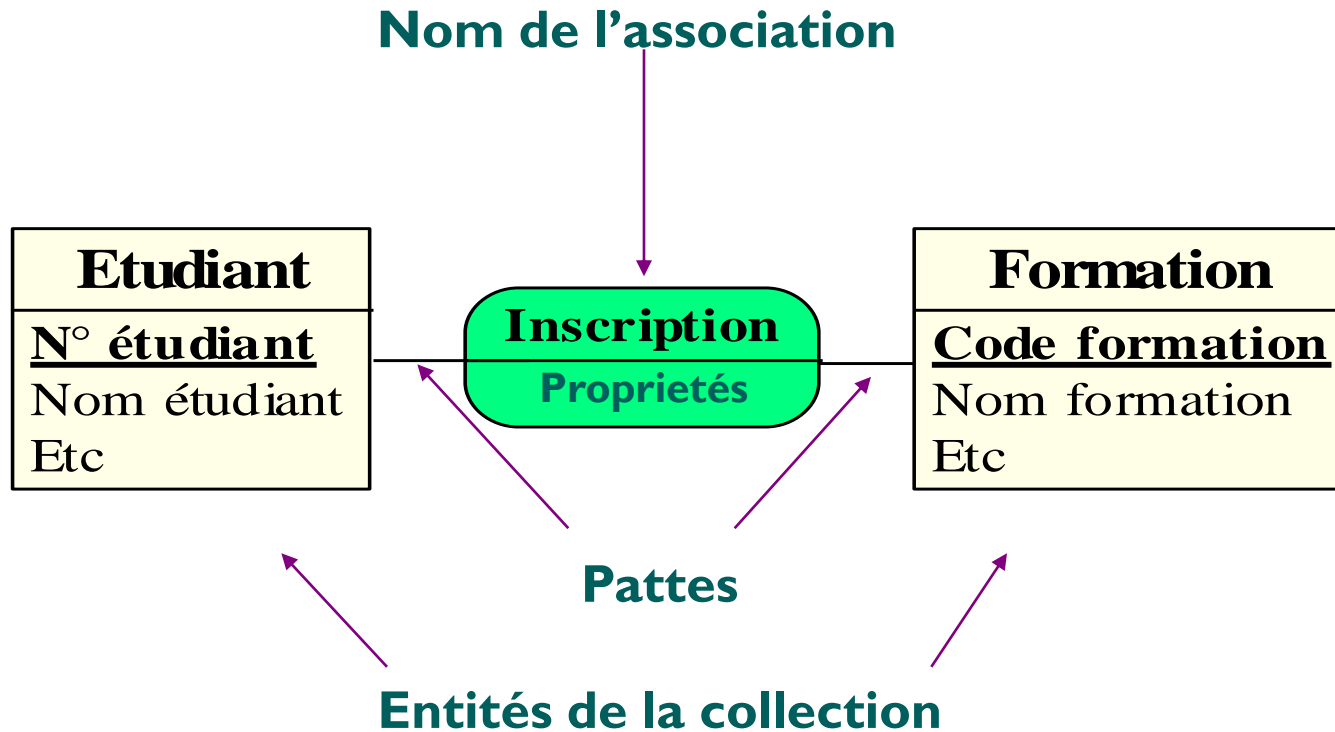
Etudiant
<u>M017895</u> Idrissi Majda Kénitra

Etudiant
<u>Z026793</u> Idrissi Anas Kénitra

Association : introduction

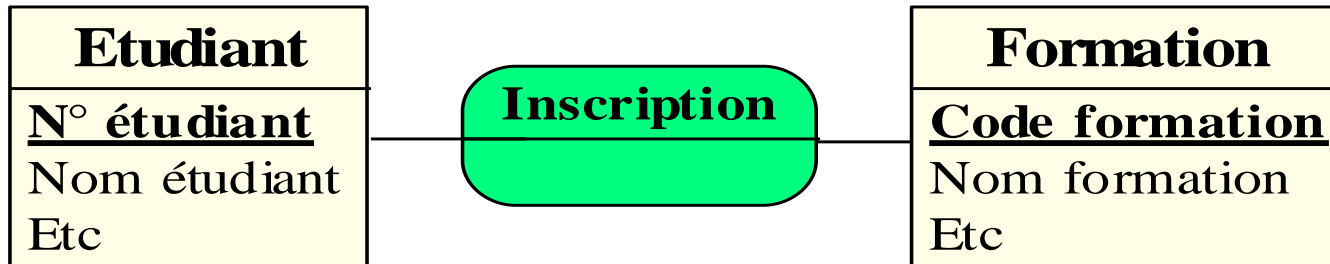
- C'est la représentation abstraite de la mémorisation d'un **lien entre des informations complexes** (représentées par des entités).
- On appelle **collection de l'association** l'ensemble des entités qu'elle relie.
- **Une occurrence de l'association** représente un lien sémantique qui concerne une occurrence de chacune des entités de la collection.

Association : représentation schématique



Association : identifiant

- Il est **implicite** !
- C'est **un n-uplet** composé des identifiants des entités-types concernées.
 - Exemple : l'identifiant de **Inscription** est le couple (N° étudiant, Code formation).



Association : les contraintes de cardinalité

- Une cardinalité est une **précision apportée sur une patte d'une association**.
- Elle indique combien de fois l'entité peut intervenir dans l'association.
- Pour un **MCD**, il est utile de savoir :
 - **la cardinalité minimale** ➔ Si une occurrence de l'entité est obligatoirement concernée par une (au moins) occurrence de l'association ;
 - **la cardinalité maximale** ➔ Si une occurrence de l'entité peut (éventuellement) être concernée par plusieurs occurrences de l'association.

Association : cardinalité minimale

Elle peut prendre deux valeurs :

- ❑ **0**, signifie que l'entité peut **ne pas** intervenir dans l'association.
- ❑ **1**, signifie au contraire qu'elle intervient **obligatoirement une fois**.

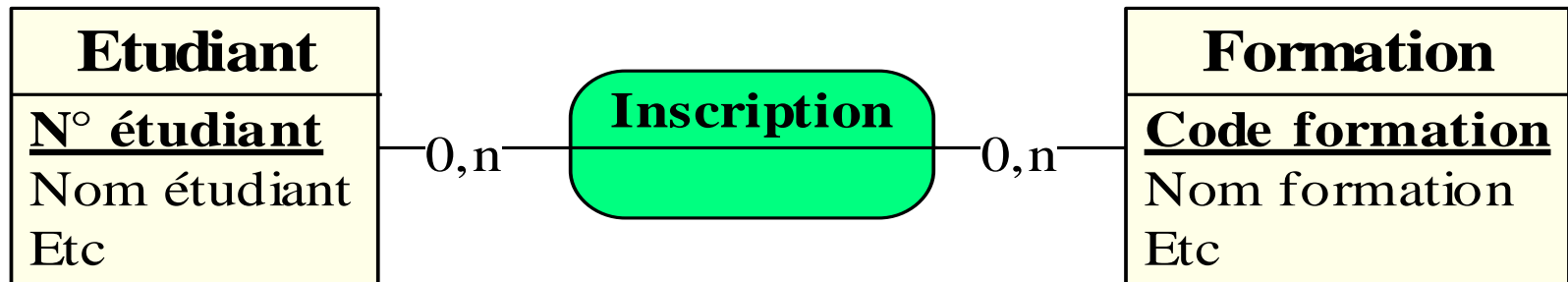
Association : cardinalité maximale

Elle peut prendre deux valeurs :

- ❑ **1**, signifie que l'entité ne peut intervenir **plus d'une seule fois** dans l'association-type ;
- ❑ **n**, signifie au contraire qu'elle peut intervenir **plusieurs fois** dans l'association.

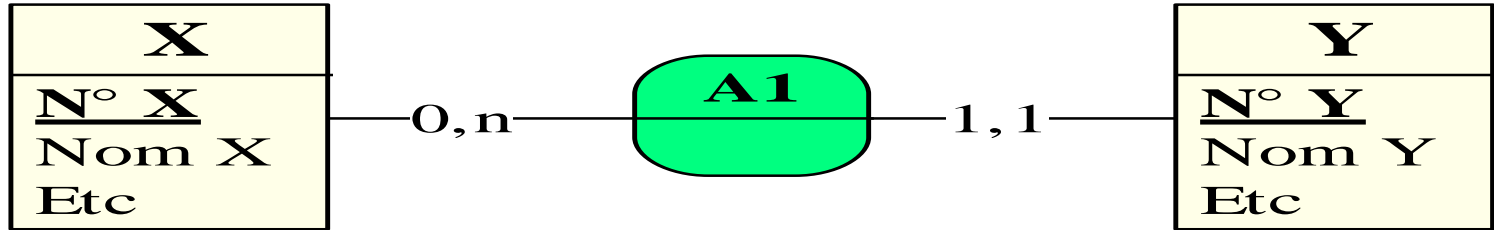
Association : c'est le 1 qui représente la contrainte

Une patte sans contrainte aura
pour cardinalités : (0, n)



Association : double contrainte sur une patte

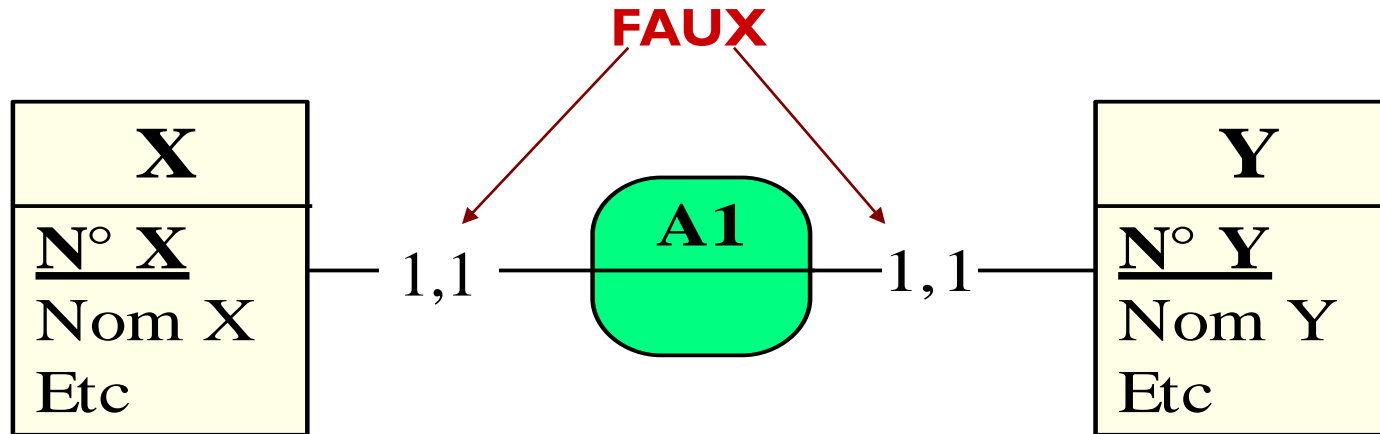
- La cardinalité minimale indique ce qui est **obligatoire**, dans le cas **le plus restrictif**, le plus souvent lors de sa création.



- Sur la patte **X**, le **0** signifie que **X** peut ne pas être reliée à **Y** lors de sa création.
- Le **1** en minimum de la patte **Y** signifie qu'en aucun cas on ne peut créer une occurrence de **Y** sans la relier en même temps à une occurrence de **X**... Cette dernière doit donc avoir été créée avant !

Association : combinaisons de cardinalités possibles

Interdit : (1, 1) / (1, 1)

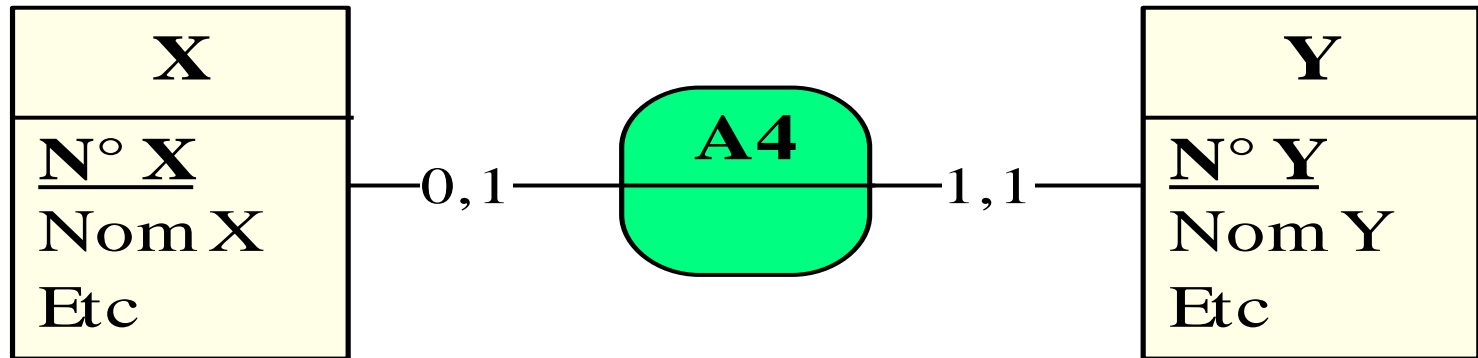


Pourquoi ?

Il faut réunir les propriétés des deux entités dans une seule.

Association : combinaisons de cardinalités possibles

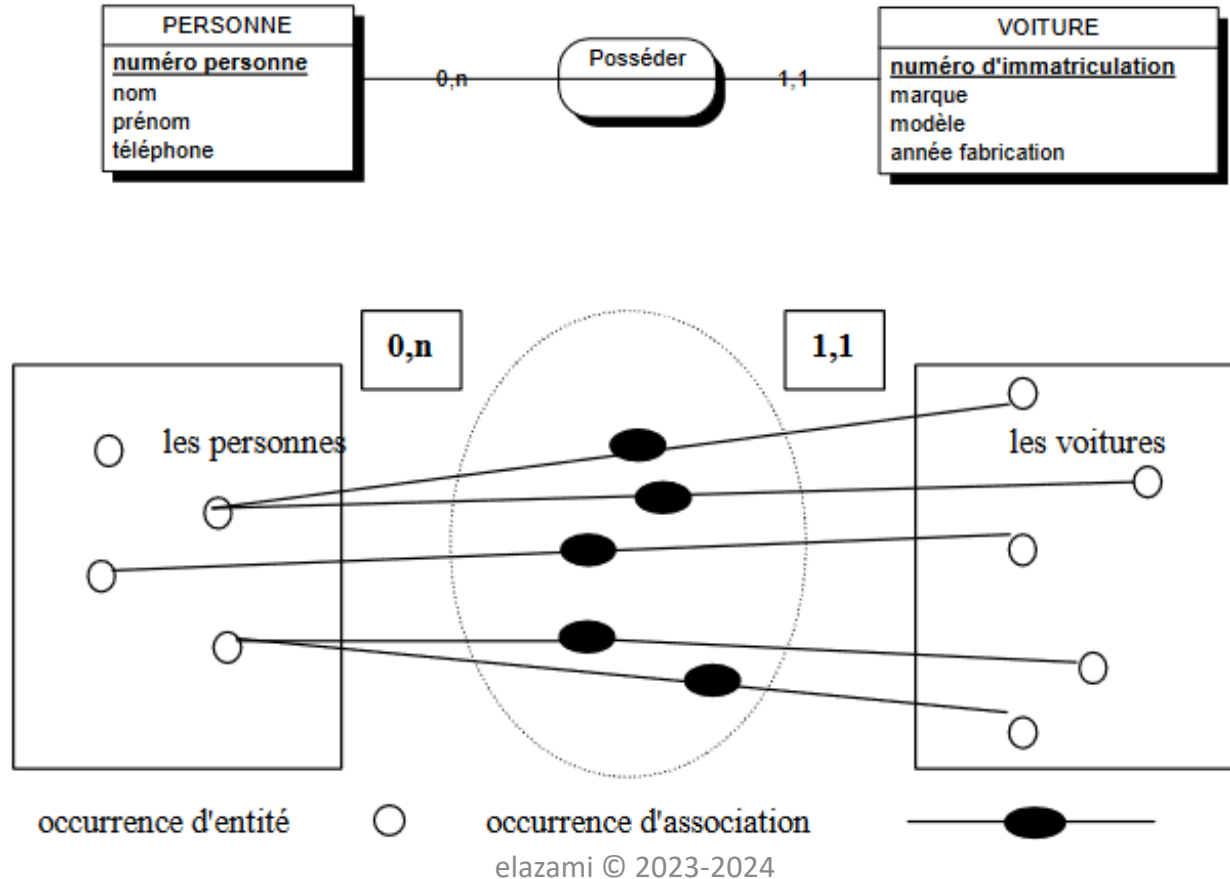
Rare et délicat : (0, 1) / (1, 1)



Significations possibles ?

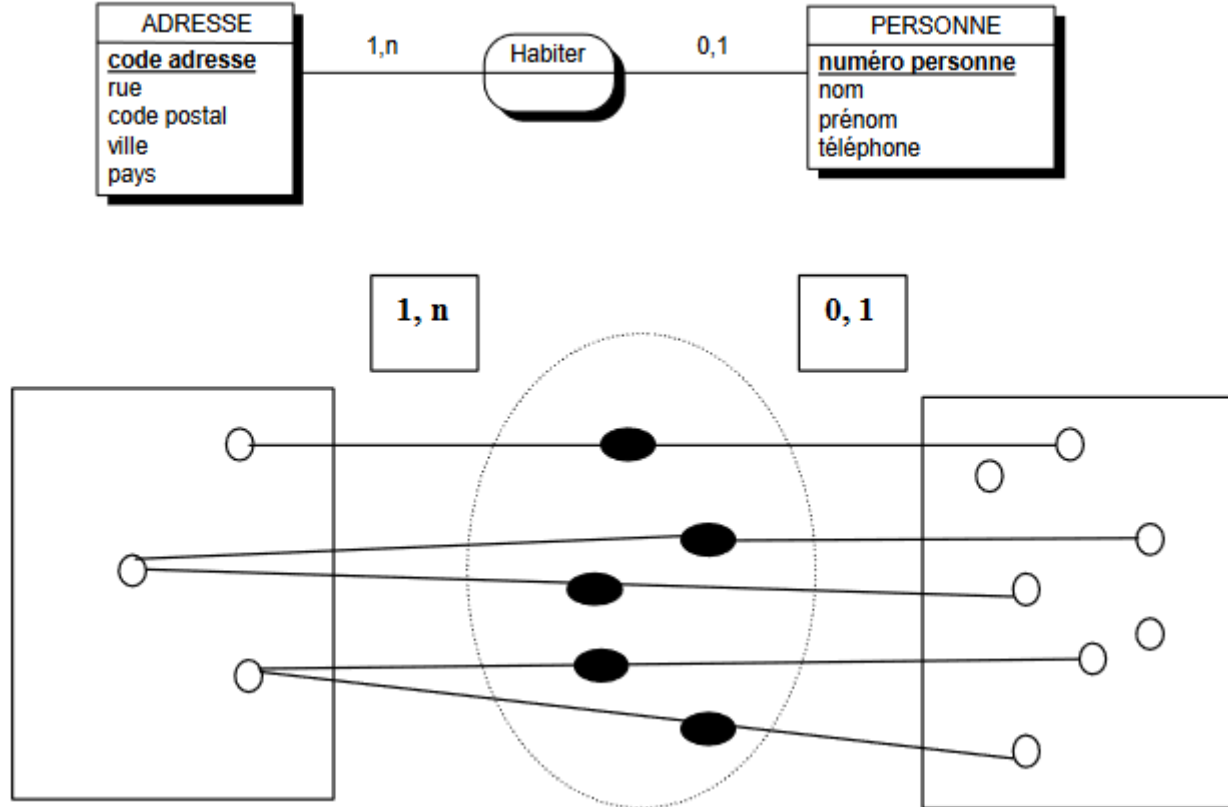
Association: Représentation des cardinalités

Une voiture est possédée par une seule personne. Une personne peut posséder de 0 à plusieurs voitures.



Association: Représentation des cardinalités

Une personne a une adresse ou est sans domicile. A une adresse, il y a une ou plusieurs personnes qui y habitent.



Premières règles...

Entités

Règle 1 : Existence d'un identifiant pour chaque entité

Règle 2 : Pour chaque occurrence d'une entité, chaque propriété ne peut prendre qu'une valeur

Règle 3 : Toutes les propriétés doivent être élémentaires (non décomposables)

Règle 4 : Toutes les propriétés autres que l'identifiant doivent dépendre pleinement et directement de l'identifiant

Association

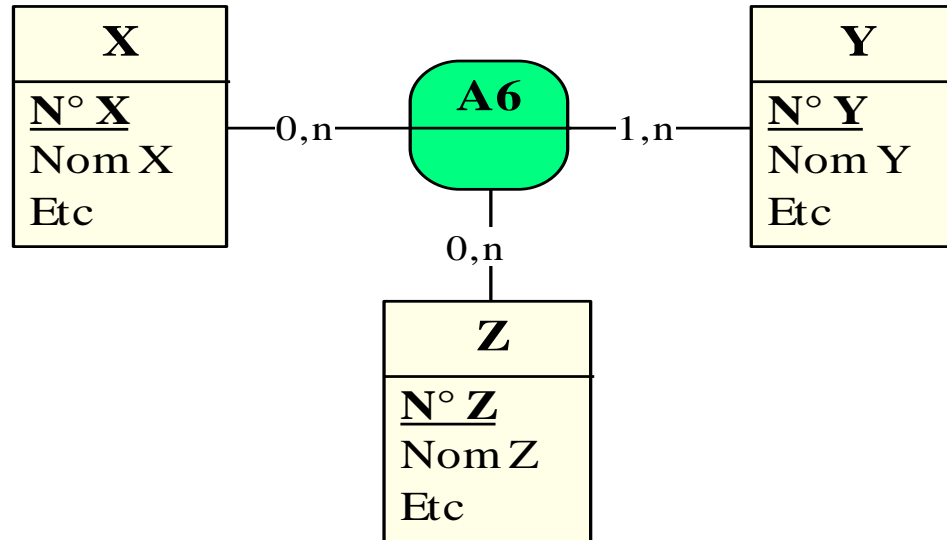
Règle 5 : A chaque occurrence d'une association correspond une et une seule occurrence de chaque entité qui participe à la relation

Règle 6 : Pour chaque occurrence de l'association, il ne peut exister qu'une et une seule valeur pour chaque propriété de l'association

Règle 7 : Toutes les propriétés d'une association doivent dépendre pleinement de l'identifiant de la relation.

Associations ternaire

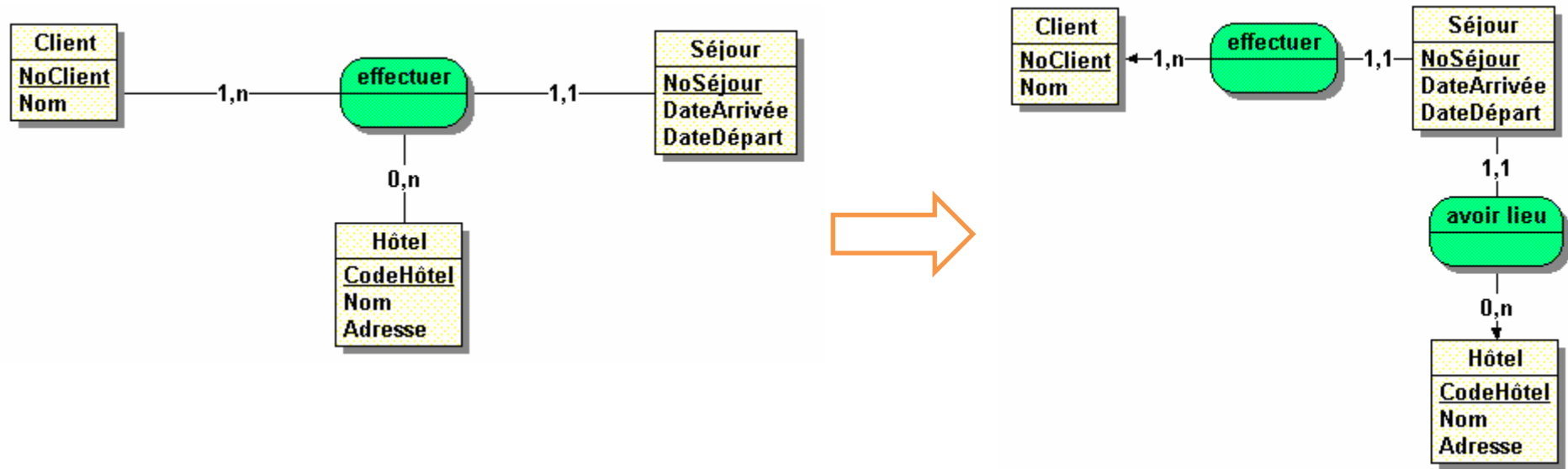
- L'association A6 décrit un lien sémantique entre les entités X, Y et Z.



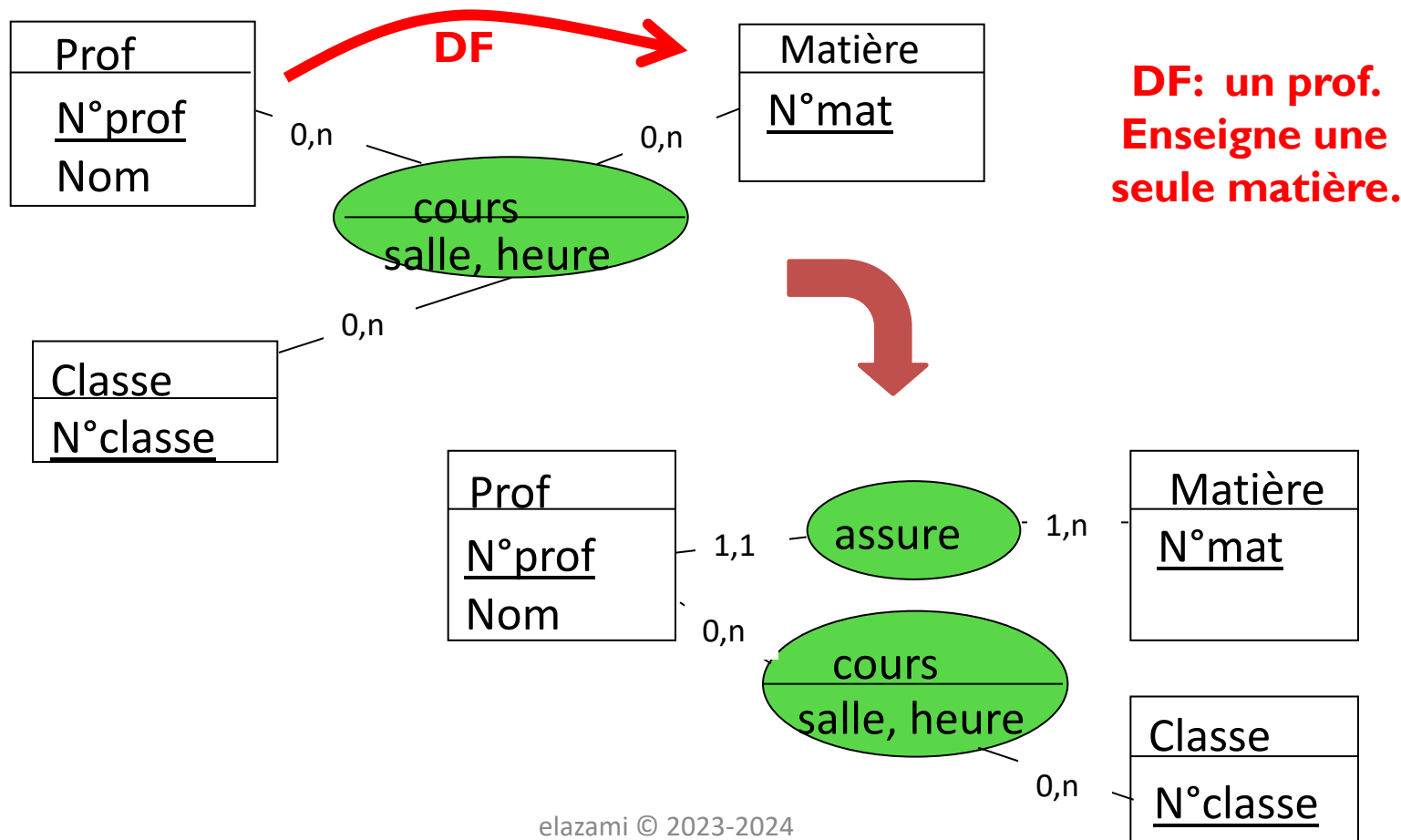
- Difficile à gérer en pratique !!! Il faut essayer d'en avoir le moins possible.

Associations ternaire

- Généralement, si une ou plusieurs des entités liées à une **relation ternaire** possèdent une **cardinalité maximale de 1**, la modélisation n'est pas optimisée
- Il faudrait mieux **décomposer** la relation ternaire, c.à.d. la représenter par 2 relations binaires.



Association ternaire : dépendance fonctionnelle



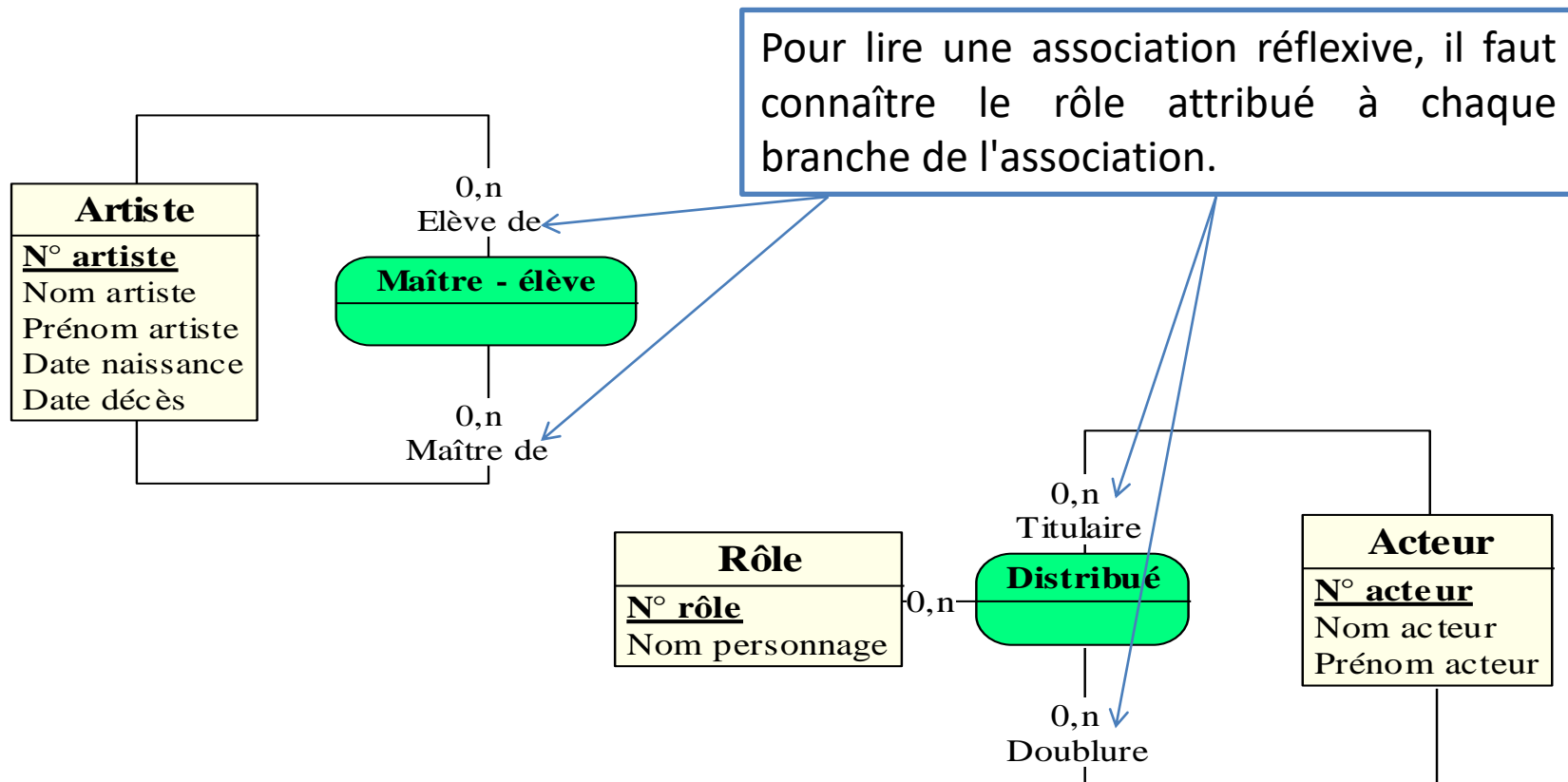
Association ternaire : règle absolue

Pour garder une association ternaire (et au-delà)...

**Toutes les cardinalités maximum
d'une
association ternaire
(et au-delà) doivent être égales à n et
pas de dépendance fonctionnelle**

Association réflexive:

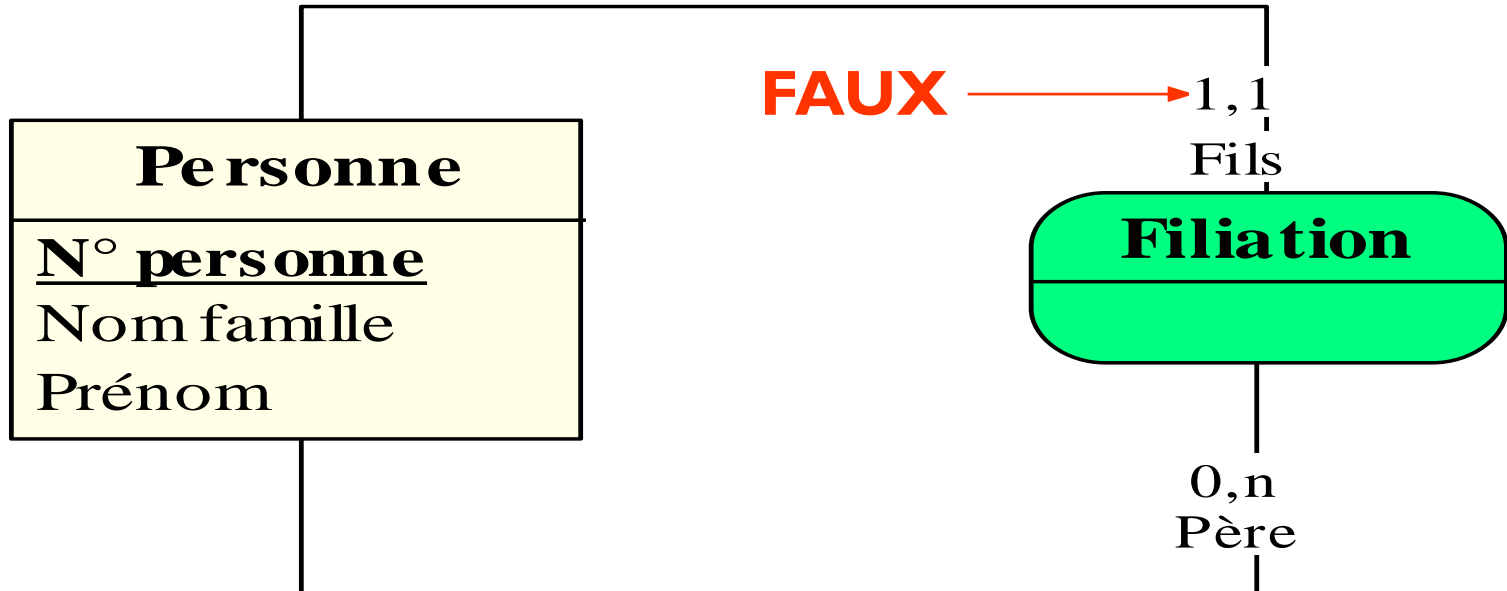
Association entre occurrences d'une même entité (1/2)



Association réflexive:

Association entre occurrences d'une même entité (2/2)

Toutes les cardinalités minimums doivent être égales à zéro !



Règles de gestion

- Elles expriment les **CONTRAINTES D'INTÉGRITÉ** du modèle et représentent les lois de l'univers réel modélisé dans le S.I.
- **Contraintes statiques:** portent sur:
 - une propriété (liste de valeurs possibles ...)
 - plusieurs propriétés d'une **même** relation ou entité
commande(no,date-cmde,date-livr) avec *date-cmde < dte-livr*
 - des propriété d'entités/relation **différentes**
 - les cardinalité
 - les dépendances fonctionnelles
- **Contraintes dynamiques :** règles d'évolution
ex: un salaire ne doit pas baisser

Règles de gestion

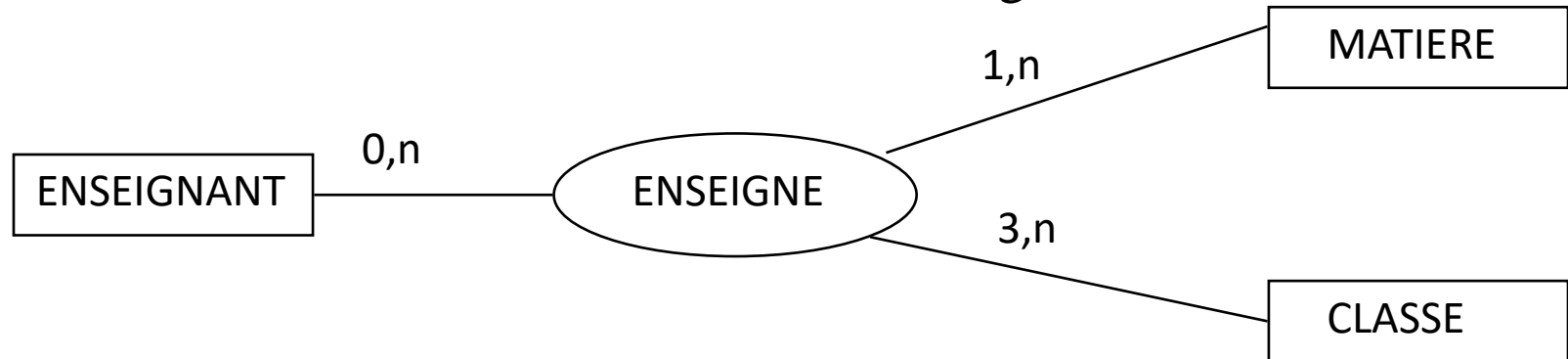
Exemple:

Dans le MCD d'une école les règles de gestion peuvent être les suivants :

RG1 : tout professeur enseigne en principe au moins une matière, mais certains d'entre eux peuvent être dispensés d'enseignement en raison de leurs travaux de recherche

RG2 : toute matière est enseignée dans au moins une classe

RG3 : toute classe a au moins trois enseignants



Dépendance fonctionnelle (DF)

Définition

- 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 de la propriété **b**.
- On note: **a** \rightarrow **b**

Exemple :

La dépendance fonctionnelle **CIN** \rightarrow **NOM** signifie qu'à un numéro CIN est associé un nom seulement.

Remarquons qu'une dépendance fonctionnelle n'est généralement pas symétrique, c'est-à-dire que CIN \rightarrow NOM n'interdit pas que deux personnes distinctes (correspondant à deux CIN différents) portent le même nom.

Dépendance fonctionnelle (DF)

Dépendance fonctionnelle élémentaire:

on parle de **DFE** entre les propriétés **a** et **b** ($a \rightarrow b$) si :

- $a \text{ --- df --- } > b$
- et si aucune partie de **a** ne détermine **b**

Exemples :

- $\text{Code-client} + \text{Nom-Client} \text{ ----- df ----- } > \text{Adresse-Client}$
n'est pas élémentaire, puisque la connaissance de **Code-Client** (partie de $\text{Code-Client} + \text{Nom-Client}$) suffit à déterminer l'adresse
- $\text{Code-client} \text{ ----- df ----- } > \text{Adresse-Client}$
est élémentaire et on peut écrire: $\text{Code-Client} \rightarrow \text{Adresse-Client}$

Dépendance fonctionnelle (DF)

Dépendance fonctionnelle élémentaire directe:

on dit que la propriété **b** dépend fonctionnellement de **a** par une dépendance fonctionnelle élémentaire directe si cette dépendance est élémentaire $a \rightarrow b$ et s'il n'existe pas de propriété **c** telle que :

$a \twoheadrightarrow c$ et $c \rightarrow b$ (c-à-d on élimine toute **transitivité**)

Exemple :

- $N^{\circ}Prof \twoheadrightarrow Code-matière$
- $Code-matière \twoheadrightarrow Nom-matière$
- $N^{\circ}Prof \twoheadrightarrow Nom-matière$

Les 2 premières dépendances fonctionnelles sont directes, mais la troisième ne l'est pas en raison de la transitivité :

$N^{\circ}Prof \rightarrow Code-matière \rightarrow Nom-matière.$

Dépendance fonctionnelle (DF)

Notion de clé

Une clé est une propriété (ou une concaténation de propriétés) d'une entité telle que **toutes les autres propriétés de l'entité dépendent d'elle fonctionnellement** et telle que ce ne soit plus vrai pour aucune de ses parties.

Dépendance fonctionnelle entre entités

- notée $A \rightarrow B$
- si toute occurrence de A est déterminée par **une et une seule** occurrence de B
- NB: les cardinalités 1-1 correspondent toujours à une DF

Propriétés de DF

Réflexivité

$$a \xrightarrow{\text{df}} a$$

Projection

$$a \xrightarrow{\text{df}} b+c \implies a \xrightarrow{\text{df}} b \text{ et } a \xrightarrow{\text{df}} c$$

Augmentation

$$a \xrightarrow{\text{df}} b \implies \forall c : a+c \xrightarrow{\text{df}} b$$

Additivité

$$a \xrightarrow{\text{df}} b \text{ et } a \xrightarrow{\text{df}} c \implies a \xrightarrow{\text{df}} b+c$$

Transitivité

$$a \xrightarrow{\text{df}} b \text{ et } b \xrightarrow{\text{df}} c \implies a \xrightarrow{\text{df}} c$$

Pseudo-transitivité

$$a \xrightarrow{\text{df}} b \text{ et } b+c \xrightarrow{\text{df}} d \implies a+c \xrightarrow{\text{df}} d$$

Propriétés de DF: Exemples

Réflexivité

$$\text{REF} \xrightarrow{\text{df}} \text{REF}$$

Projection

$$\text{REF} \xrightarrow{\text{df}} \text{DESIGN} + \text{PU} \implies \text{REF} \xrightarrow{\text{df}} \text{DESIGN}$$

Augmentation

$$\text{REF} \xrightarrow{\text{df}} \text{PU}$$

$$\text{REF} \xrightarrow{\text{df}} \text{PU} \implies \text{REF} + \text{DESIGN} \xrightarrow{\text{df}} \text{PU}$$

Additivité

$$\text{REF} \xrightarrow{\text{df}} \text{DESIGN}$$

$$\text{et } \text{REF} \xrightarrow{\text{df}} \text{PU} \implies \text{REF} \xrightarrow{\text{df}} \text{DESIGN} + \text{PU}$$

Transitivité

$$\text{REF} \xrightarrow{\text{df}} \text{CODETVA}$$

$$\text{et } \text{CODETVA} \xrightarrow{\text{df}} \text{TXTVA} \implies \text{REF} \xrightarrow{\text{df}} \text{TXTVA}$$

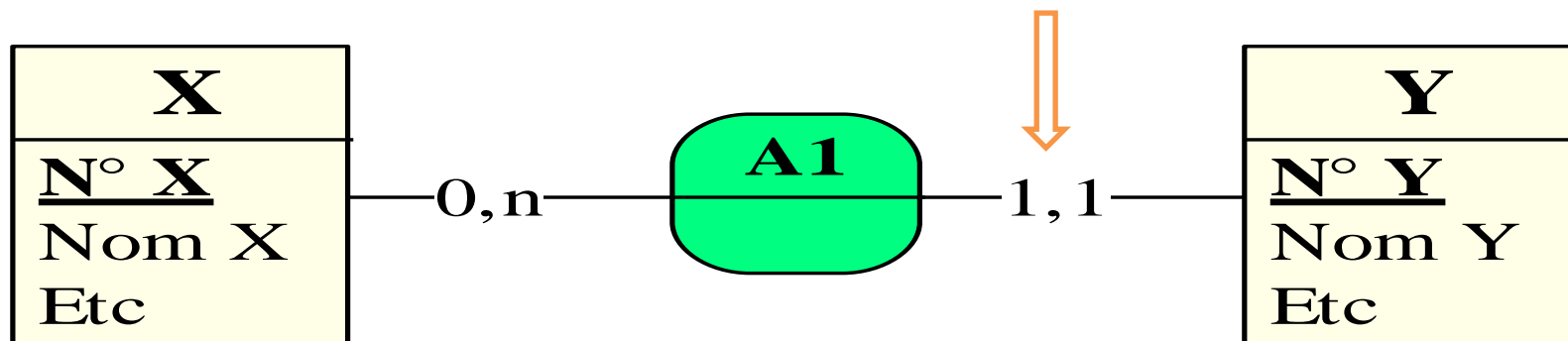
Pseudo-transitivité

$$\text{REF} \xrightarrow{\text{df}} \text{CODETVA}$$

$$\text{CODETVA} + \text{PU} \xrightarrow{\text{df}} \text{TXTVA} \implies \text{REF} + \text{PU} \xrightarrow{\text{df}} \text{TXTVA}$$

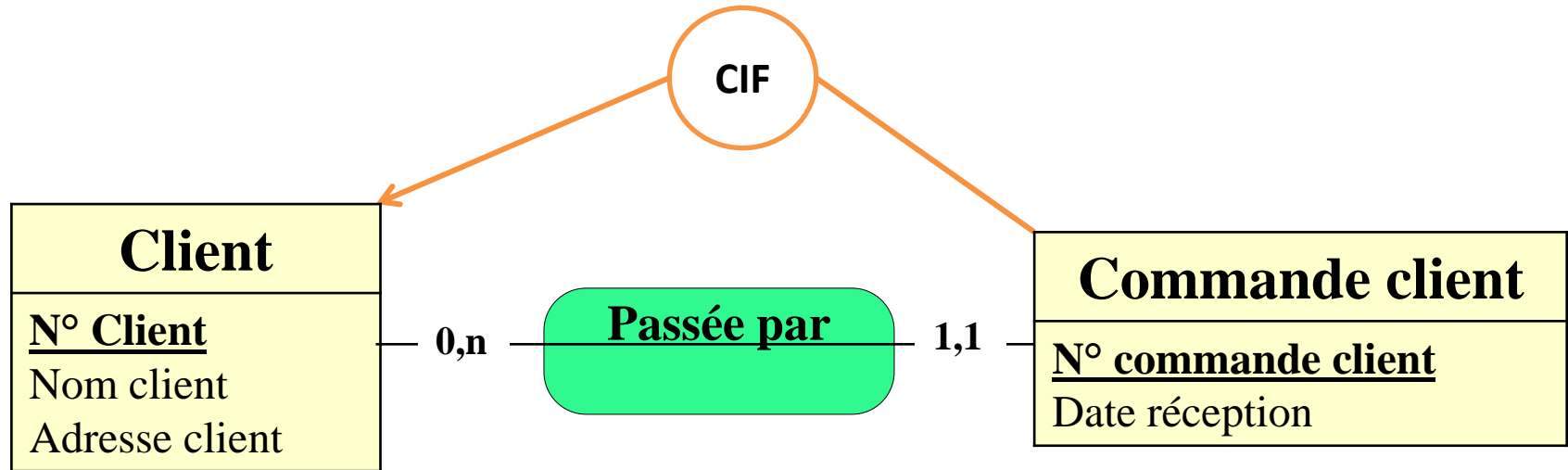
Association binaire fonctionnelle

Lorsqu'une association-type binaire a une patte dont la **cardinalité maximale** est **égale à 1**, alors elle est dite **fonctionnelle**.

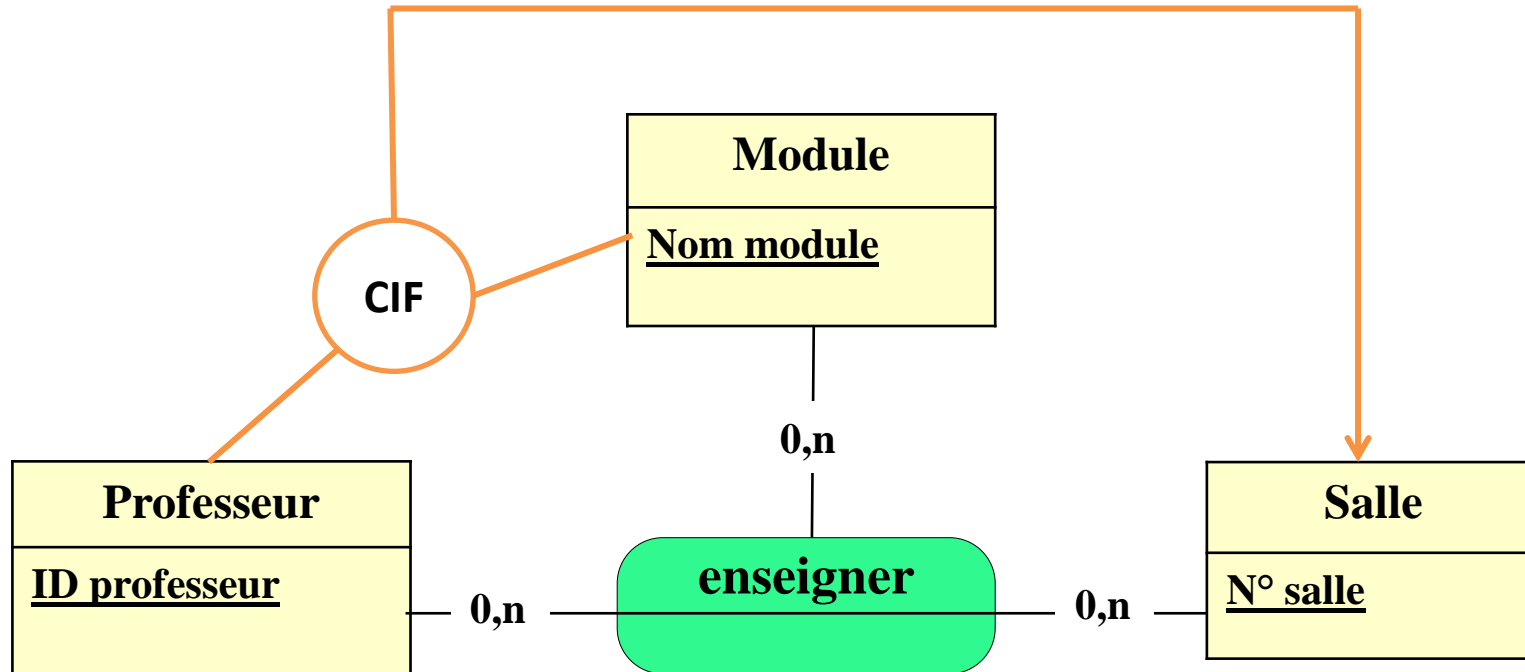


Contraintes d'Intégrité Fonctionnelle (CIF)

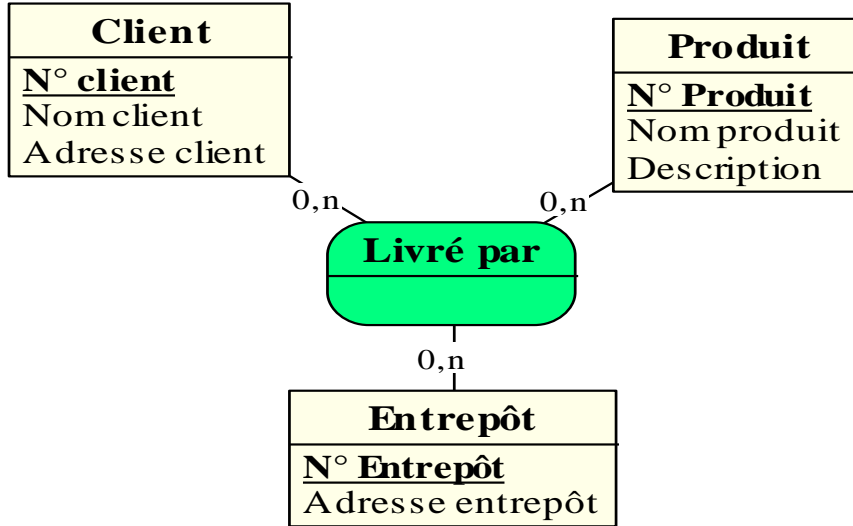
- Sur association-type **binaire** :



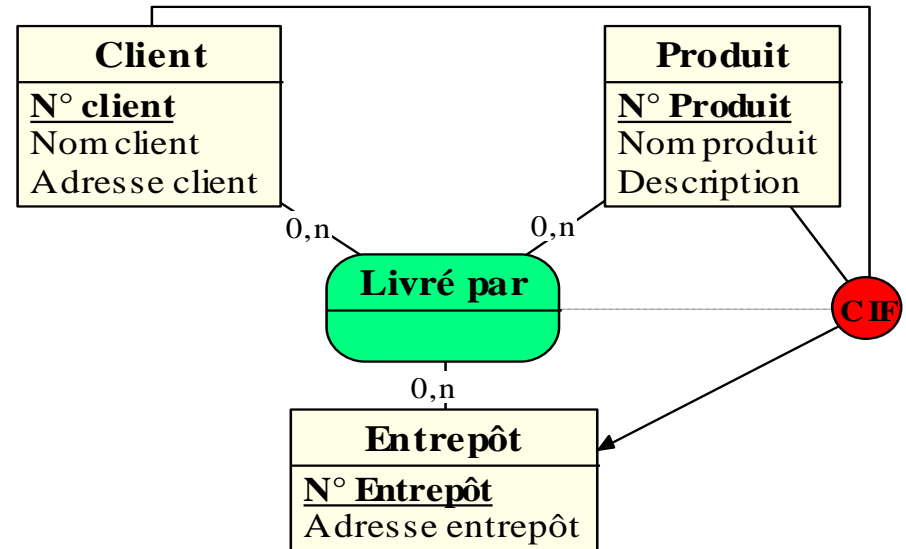
CIF : cas d'une association non binaire



CIF : cas d'une association non binaire (Contre exemple)



Comment garantir l'unicité de l'entrepôt pour un couple (Client,Produit) ?



Exercices:

Exercice 1: Formation

On veut gérer **les étudiants** et **les professeurs** d'un ensemble de **formations** dispensés par une université.

- Un **étudiant** est identifié par son **nom**, son **prénom**, et son **âge**.
- Un étudiant suit une formation.
- Chaque **formation** a un **nom** et une **durée** (nombre d'années).
- Elle est assurées par un ensemble d'enseignants.
- Chaque **enseignant** est connu par son **nom**, son **prénom** et la matière qu'il enseigne.
- On désire savoir le nombre d'heurs qu'un enseignant effectue pour chacune des formations dans lesquelles il enseigne.

Donner le modèle **Entité-Association**.

Exercices:

Exercice 2 : Bibliothèque

- Le système informatique d'une bibliothèque enregistre **le numéro national, le nom, le prénom et l'adresse (rue, numéro, code postal, ville)** de chacun de ses **clients**.
- Le système enregistre pour chaque **livre** disponible son numéro unique **ISBN**, son **nom, le nom de ses auteurs** et sa **date d'achat**.
- On suppose que la bibliothèque possède **au plus** un livre par numéro **ISBN**.
- Le système enregistre également tous **les emprunts** des clients.
- On identifie ces emprunts avec **un numéro unique**, on retient **la date d'emprunt** et **la date de retour** lorsque le livre rentre.

Donner un modèle **entité-association** de ce système et préciser **les contraintes d'intégrité**.

Exercices:

Exercice 3 : Employés

- Un employé est identifié par son **numéro de sécurité sociale (SSN)** et a **un nom** et **un prénom**.
- Chaque employé peut **être supervisé** par **au plus** un autre employé.

Donner un modèle **entité-association** de ce problème et préciser les contraintes d'intégrité.

Normalisation de MCD

Normalisation

- Les règles de normalisation permettent de concevoir un schéma de base de données correct :
 - sans redondance d'information.
 - sans anomalie de mise à jour.
- Elles se basent sur
 - les **dépendances fonctionnelles** (DF) qui traduisent les relations entre les données.
 - les **formes normales** qui définissent les relations bien conçues.

Normalisation

L'intérêt de la normalisation

- voyons les problèmes que peuvent poser l'utilisation d'une représentation non normalisée.
- Soit le schéma entité:

FOURNISSEUR (NomFournisseur, AdresseFournisseur, Produits, Prix)

- On peut distinguer les occurrences suivantes des attributs de cette entité :

NomFournisseur	AdresseFournisseur	Produit	Prix
Ali	10 , Rue des Gras - Fès	Chaise	20
		Table	35
Said	86 , Rue neuve - Rabat	Bureau	60
Taher	26 , Rue des Dômes - Méknes	Lit	50
Said	39, Rue des Buttes - Fès	Lampe	18
		Table de chevet	25

Normalisation

1° problème :

Il n'y a pas de clé primaire : on ne sait pas si les deux **Said** sont différents ou pas (si c'est le même **Said**, il y a une des deux adresses qui est fausse).

2° problème :

L'adresse n'est pas décomposée. Si on veut par exemple rechercher tous les fournisseurs qui habitent la même ville, ça ne va pas être possible

3° problème :

Une relation (table) correspondant à ce schéma pourra éventuellement contenir plusieurs produits pour un même fournisseur.

Dans ce cas, il faudra faire face à un certain nombre de problèmes :

- l'adresse du fournisseur sera dupliquée dans chaque occurrence (redondance),
- si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher et mettre à jour toutes les occurrences correspondant à ce fournisseur,
- si on insère un nouveau produit pour un fournisseur déjà référencé, il faudra vérifier que l'adresse est identique,
- si on veut supprimer un fournisseur, il faudra retrouver et supprimer tous les occurrences correspondantes à ce fournisseur (pour différents produits) dans la table.

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**

Normalisation

- Il existe plusieurs niveaux de normalisation :
 - Première forme normale (1FN)
 - Deuxième forme normale (2FN)
 - Troisième forme normale (3FN)
- ...
- Un modèle relationnel est dit **normalisé** quand toutes ses tables sont en **3FN**.

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

Première forme normale (1FN)

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 sans doute la concaténation de rue et ville et ne constitue alors pas une propriété élémentaire.
- Il doit être : **CLIENT (code_client, nom, rue, ville)**

Deuxième forme normale (2FN)

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

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

Deuxième forme normale (2FN)

Exemple :

- **LIGNE(num_commande+num_article,description_article,qté_commandée)**
- L'identifiant de cette entité est la concaténation de **num_commande** et **num_article**.
- Cette entité n'est pas en 2FN car la dépendance fonctionnelle **num_commande+num article → 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)**

Troisième forme normale (3FN)

Une entité est en *3^o forme normale* si et seulement si :

- elle est en 2^o 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.

Troisième forme normale (3FN)

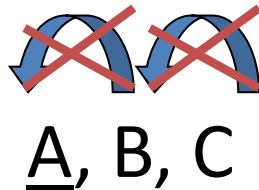
Exemple :

- CLIENT (code_client, nom_client, code_categ, nom_categ)
- Cette relation n'est pas en 3FN car la dépendance fonctionnelle **code_client** \rightarrow **nom_categ** n'est pas directe du fait de la transitivité **code_client** \rightarrow **code_categ** \rightarrow **nom_categ**
- Ça doit être :
 - CLIENT (code_client, nom_client)
 - Lié par la relation « appartient à » à l'entité:
 - CATEGORIE (code_categ, nom_categ)

Forme normale de Boyce-Codd

Définition : Boyce-Codd forme normale:

- Une relation est en **BCFN** si :
 - elle est en **3^{ème} forme normale** ;
 - **aucun attribut ne dépend d'un attribut non clé.**
- C.a.d. toute DF a pour **partie gauche** (origine de la flèche) **une clé candidate** ou *primaire* entière.



Forme normale de Boyce-Codd

Considérons la relation suivante:

Cours (Matiere, Classe, Professeur)

complétée par les règles de gestion suivantes:

**un professeur n'enseigne qu'une seule matière,
une classe n'a qu'un seul enseignant par matière**

→ desquelles on déduit les DF suivantes:

DF 1. Matière, Classe → Professeur

DF 2. Professeur → Matière

- Cette relation **est en 3FN**, néanmoins il est impossible d'enregistrer un professeur sans classe affectée (puisque classe est une partie de la clé primaire de la table Cours),
- et la disparition d'une classe peut entraîner la disparition de professeur.

Forme normale de Boyce-Codd

On pourrait alors décomposer la table Cours en 2 tables:

Spécialité (Professeur, Matière)

Enseignant (Classe, Professeur)

Mais la première DF serait alors perdue...

Normalisation : 2ième exemple FNBC

Soit les DF suivantes sur cette relation :

- $N^{\circ}SS, Pays \rightarrow Nom$
 - $N^{\circ}SS, Pays \rightarrow Région$
 - $Région \rightarrow Pays$
- ❑ Il existe une DFE qui n'est pas issue d'une clé et qui détermine un attribut appartenant à une clé.
- ❑ Cette relation est en 3NF, mais pas en BCNF (car en BCNF toutes les DFE sont issues d'une clé).

Exercices

Exercice :

Critiquer un MCD qui comporterait l'entité suivante :

Stock
<u>Réf-Produit, N°-Magasin</u> Désignation QuantitéStock

Sachant qu'un produit peut être en stock dans plusieurs magasins

Proposer un MCD normalisé

Exercices

- **STOCK** n'est pas en **2FN** du fait de la DF :

Réf-Prduit \longrightarrow Désignation

- Le modèle correct est :



Modèle Logique des Données

(MLD)

Introduction au MLD

- la structure des données représentée sous forme d'un **MCD** ne peut être portée sur un système informatique.
- Il convient donc de traduire le MCD en une structure transportable sur une machine → **Modèle Logique de Données** ou **MLD**.
- Ce **MLD** est en fait le dernier pas vers le **Modèle Physique de donnée (MPD)**
- Le passage du **MCD** au **MLD** est assuré à partir d'un certain nombre de règles.

Règles de passage du MCD au MLD

- Les données sont représentées sous forme de TABLES appelées **Tables Relationnelles**.

Notation :

ETUDIANT (num étudiant, nom, âge)

- *Une entité* devient **une Table Relationnelle**
- *Une propriété* devient **un Attribut**
- *L'identifiant* devient **Clé primaire**

Règles de passage du MCD au MLD

A PROPOS DES CLES

CLE PRIMAIRE

Il s'agit de la clé qui assure l'unicité de l'occurrence (la ligne) pour que la table soit relationnelle. *Cette clé primaire est **obligatoire***

CLE ETRANGERE

Une propriété **clé étrangère** dans une table est une **Clé primaire** dans une autre table. Elle permet de faire la jointure entre les deux tables.

CLE CANDIDATE

Il s'agit d'un attribut qui n'est pas clé primaire de la table, mais dont on désire l'unicité de ses valeurs comme pour la clé primaire (Quand on a plusieurs clés on en choisira une comme identifiant).

Exemple:

PERSONNE (matricule , nom, prénom, *numéro CIN*, adresse...)

Matricule est clé primaire

Numéro CIN est une clé candidate.

CLE SECONDAIRE

Il s'agit de l'indexation d'une colonne sur laquelle il y a de nombreux accès (optimisation des temps de traitement).

Règles de passage du MCD au MLD

❑ Cas des entités

Toute entité devient une table, l'identifiant de l'entité correspond à la clé de la table.

CLIENT
<u>id_client</u>
Nom_Client
Tel_Client



CLIENT (id_client, Nom_Client, Tel_client)

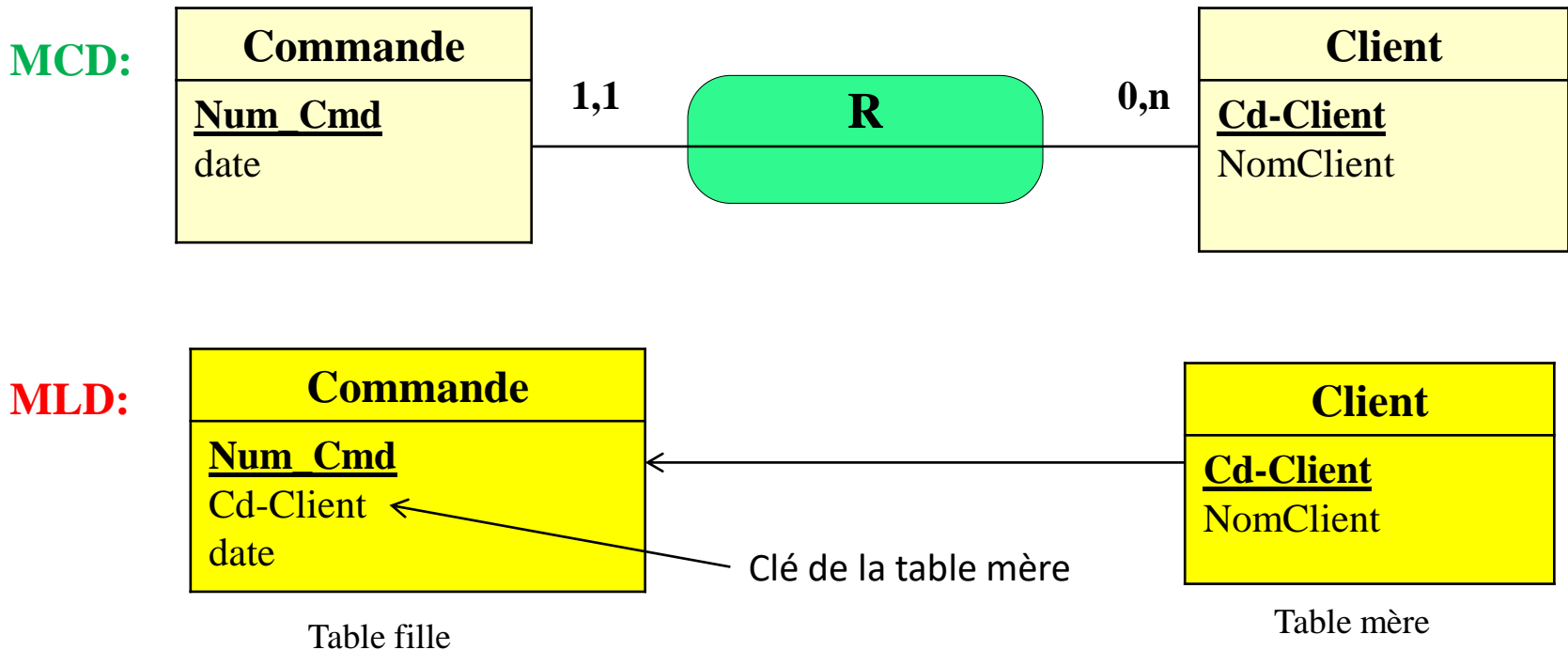
Règles de passage du MCD au MLD

- ❑ **Cas de relations binaires aux cardinalités $(x,1)$ - (x,n) (où $x=0$ ou 1)**
 - Les **entités** concernées deviennent des **tables**.
 - La table qui correspond aux cardinalités **0,n** ou **1,n** devient **la table mère** et celle qui correspond aux cardinalités **0,1** ou **1,1** devient **la table fille**.
 - **La clé de la table mère** est ajoutée aux champs de **la table fille**.
 - Si la **relation** est porteuse de **propriétés**, celles-ci migrent dans **la table fille**

Règles de passage du MCD au MLD

❑ Cas de relations binaires aux cardinalités $(x,1)$ - (x,n) (où $x=0$ ou 1)

Exemple:



Règles de passage du MCD au MLD

❑ Cas de relations binaires aux cardinalités $(x,n) - (x,n)$ (ou $x=0$ ou 1)

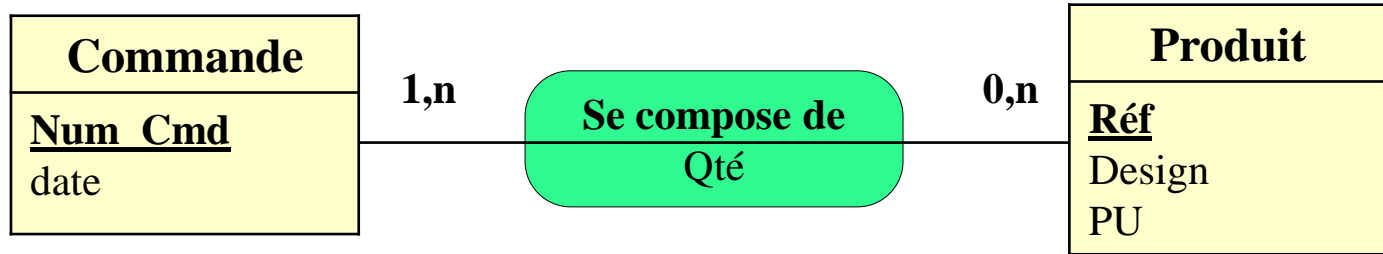
- La relation est transformée en une table supplémentaire.
- Cette table est fille des tables correspondantes aux entités pour lesquelles il y a cardinalités $0,n$ ou $1,n$.
- Les clés de celles-ci migrent dans la table fille et leur concaténation constitue la clé de celle-ci.
- Si la relation n'est pas porteuse de propriétés, la table fille commune (lien) ne contiendra que les clés de ses tables mères

Règles de passage du MCD au MLD

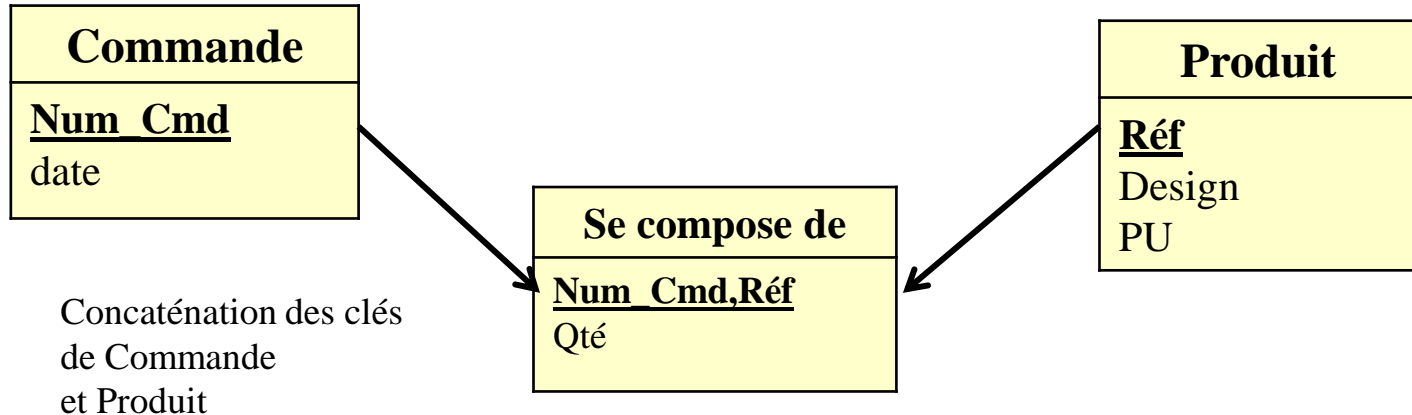
❑ Cas de relations binaires aux cardinalités $(x,n) - (x,n)$ (ou $x=0$ ou 1)

Exemple:

MCD:



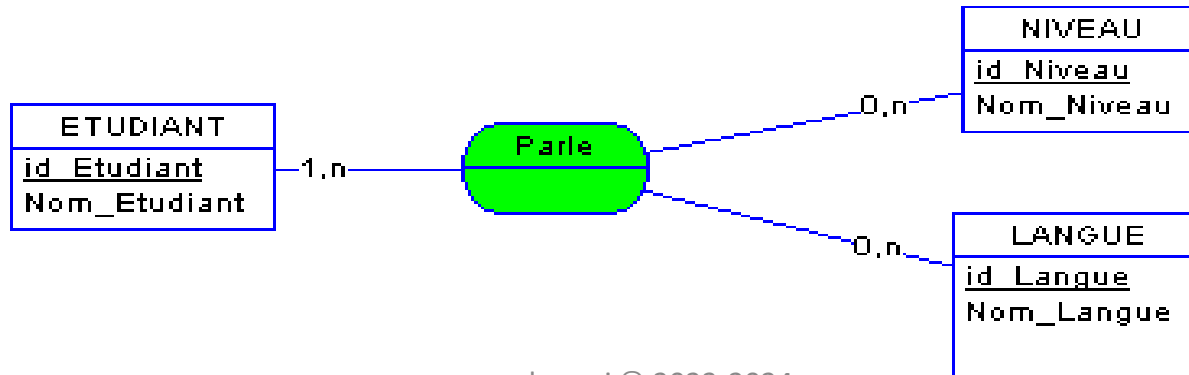
MLD:



Règles de passage du MCD au MLD

❑ Relation n-aire (quelles que soient les cardinalités)

- Il y a création d'une table supplémentaire ayant comme Clé Primaire la concaténation des identifiants des entités participant à la relation.
- Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.
- **Exemple:** Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.



Règles de passage du MCD au MLD

□ Relation n-aire (quelles que soient les cardinalités)

Schémas relationnels:

ETUDIANT (id_Etudiant, Nom_Etudiant)

NIVEAU (id_Niveau, Nom_Niveau)

LANGUE (id_Langue, Nom_Langue)

PARLE (id_Etudiant, id_Niveau, id_Langue)

Règles de passage du MCD au MLD

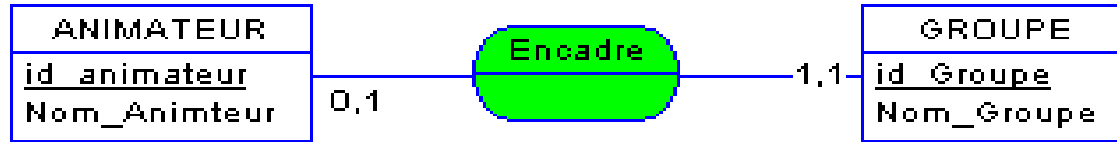
❑ Cas de relations binaires aux cardinalités (0,1) – (1,1)

- La Clé Primaire de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1) :

Exemple:

Dans un centre de vacances, Chaque animateur encadre 0 ou 1 groupe, chaque groupe étant encadré par un et un seul animateur.

MCD



MLDR :

ANIMATEUR (id_Animateur, Nom_Animateur)

GROUPE (id_Groupe, Nom_Groupe, #id_animateur)

Règles de passage du MCD au MLD

❑ Cas particulier des associations réflexives

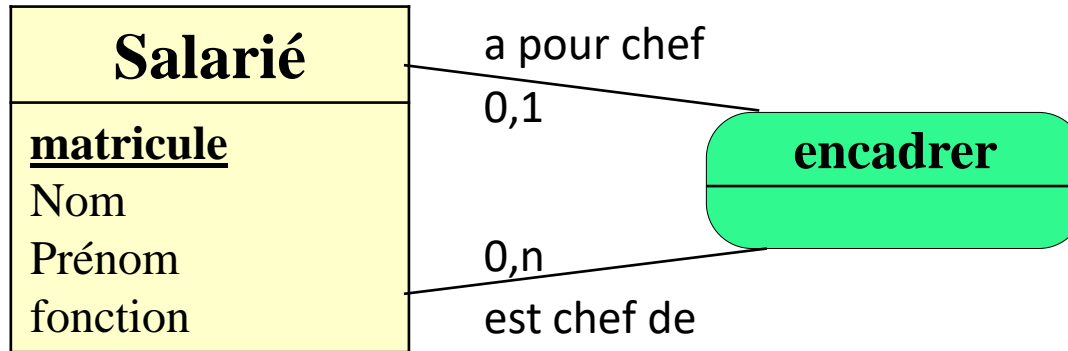
- Les associations réflexives suivent les mêmes règles, mais posent un problème particulier :
 - une même propriété va se retrouver **deux fois** en attribut dans la même relation.
 - Il faut alors donner un **nom différent et significatif** aux deux attributs correspondants.
- Dans les réflexives, il est conseillé de **nommer** les branches par des **rôles** pour pouvoir lire dans le bon sens l'association.
- Les rôles aident à nommer les attributs correspondant à l'association.

Règles de passage du MCD au MLD

❑ Cas particulier des associations réflexives

- Réflexive hiérarchique

Une branche à la cardinalité max à 1 et l'autre à n



Traduction en modèle relationnel :

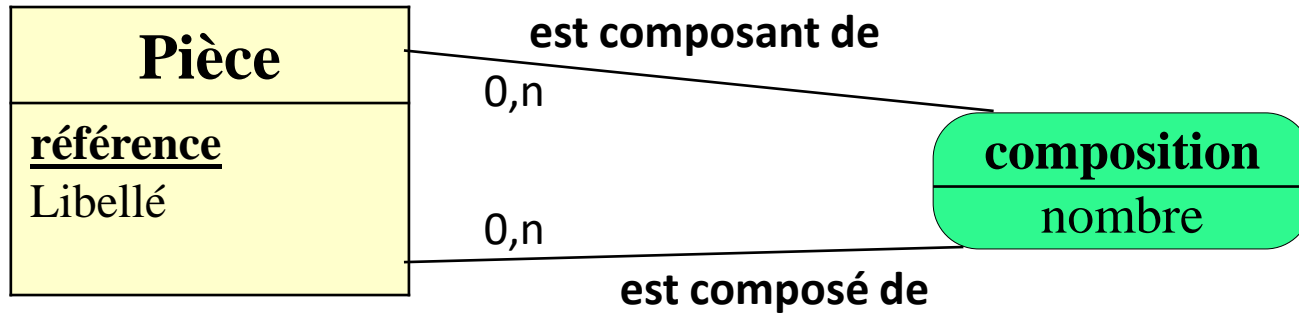
Salarié (matricule, nom, prénom, fonction, #matricule_chef)

Règles de passage du MCD au MLD

❑ Cas particulier des associations réflexives

- Réflexive non hiérarchique

les deux branches ont la cardinalité max à **n**



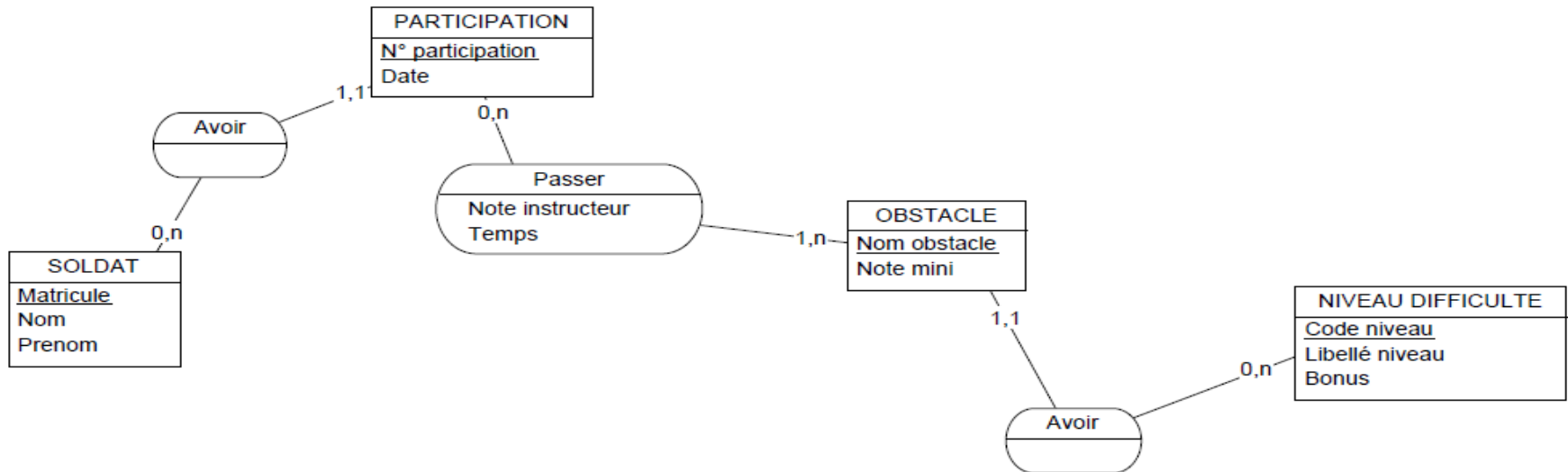
Traduction en modèle relationnel:

Pièce (référence, libellé)

Composition (#référence composé, #référence composant, nombre)

Exercices

Donner le MLD correspondant à l'MCD suivant:



MCD-Gestion du Parcours du Combattant

Exercices

Donner le MLD correspondant à l'MCD suivant:

