

Modèle entité association

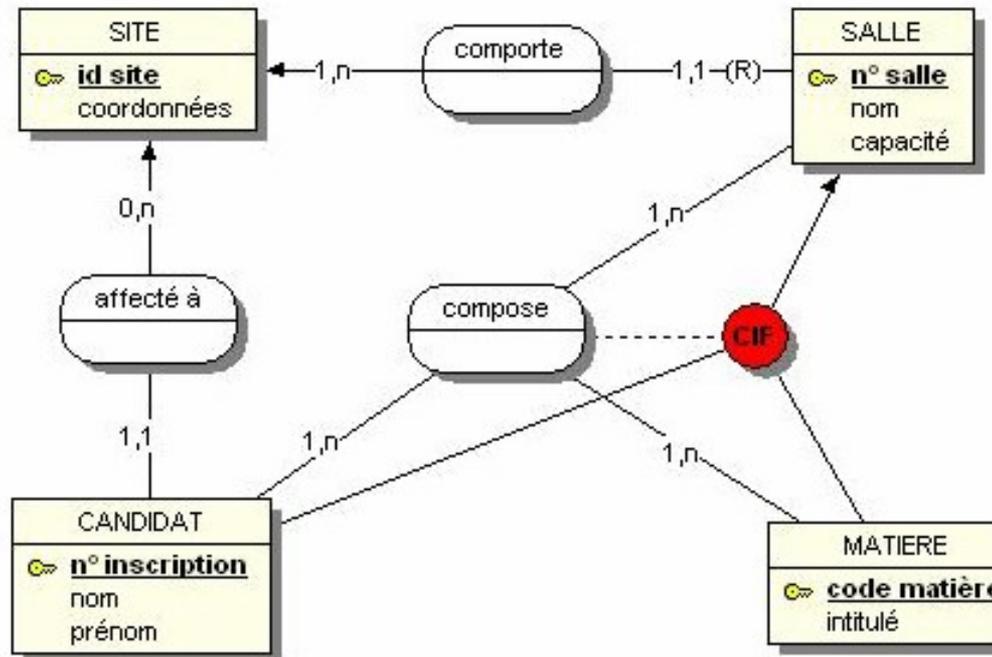
Le modèle Entité /Association

Propose une modélisation abstraite du monde réel sous forme d'un ensemble d'entités reliées par des associations.

Facilite la conception de schéma de bases de données.

Fait abstraction de l'organisation physique des données et des soucis d'efficacité.

Correspond au **Modèle Conceptuel des Données** (MCD) Merise.



Concept de base du modèle *Entité /Association*

Le formalisme *E/S* comporte quatre concepts types de base :

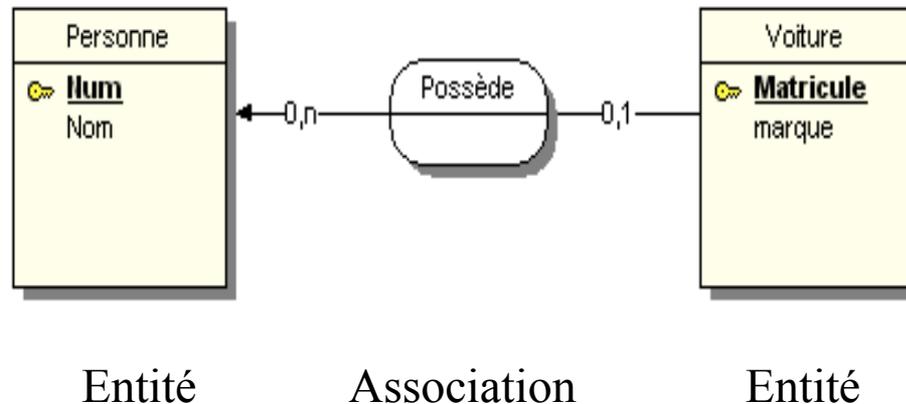
Deux concepts **structuraux** :

le concept *d'entité* pour représenter un ensemble *d'objets*

le concept *d'association* pour représenter un ensemble de **liens** entre les objets

Un troisième concept descriptif : *la propriété*

Un quatrième concept qui qualifie **la liaison** entre entité et association : *la cardinalité*.



L'entité

Un objet du monde réel qui peut être **identifié** et que l'on souhaite **représenter**

La classe d'entité : correspond à une collection d'entités décrites par leur type commun (le format)

L'instance (occurrence) d'entité : correspond à un élément particulier de la classe d'entité

Attention : on dit entité pour les deux ! Comprendre selon le contexte.

Il existe généralement plusieurs entités dans une classe

Exemple :

La classe d'entité PERSONNE comprend les informations : numéro d'identifiant, nom, etc.

Voici deux instances (occurrences) de cette entité :

(445358243,Dupont,...)

(701648143,Martin,...)

Les attributs des entités

Un **attribut** est une donnée élémentaire qui caractérise une entité (ou association).

Dans le modèle E/A de base tout attribut est **simple**

Dans le modèle E/A étendu, les attributs peuvent être :

Simple : non décomposable, de valeur atomique, comme le salaire d'un employé

Composé : comme date de naissance (jour,mois,année) et adresse (rue,ville,code_p).

Obligatoire : une valeur au moins par occurrence, comme le nom d'une personne.

Manquant/optionnel : peut ne pas prendre de valeur (NULL), comme le numéro de téléphone portable.

L'identifiant de l'entité

- Un identifiant aussi appelé clé est un attribut (ou groupe d'attributs) qui permet de retrouver une instance d'entité unique à tout instant parmi celles de la classe

Exemple :

Numéro_client dans l'entité Client.

Votre numéro de carte étudiant dans l'entité Etudiant.

- Plusieurs types d'identifiant:

Identifiant **simple** «naturel» : (nom_pays) ou «artificiel» (num_client)

Identifiant **composé** : numéro de sécurité sociale

Identifiant **relatif** : comprend des propriétés n'appartenant pas à l'entité à identifier

L'identifiant de l'entité (suite)

Un identifiant doit être:

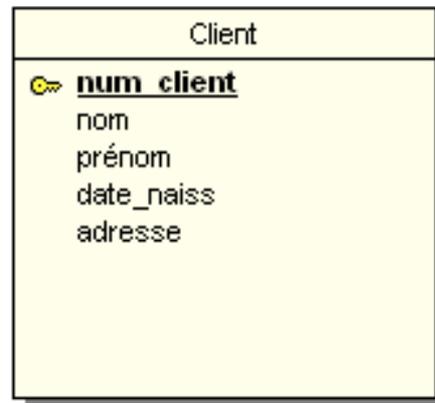
Univalué : une occurrence entité donne une valeur de l'identifiant

Discriminant : une valeur de l'identifiant donne une occurrence de l'entité

Stable : la première valeur attribuée à cette propriété ne peut être ultérieurement modifiée

Minimal : n'inclue pas (au sens strict) un autre identifiant.

Exemple :



Tous les attributs faisant partie de la clé sont **soulignés**.

Choix de l'identifiant

- Eviter les clés obtenues par **concaténation d'attributs** : mieux vaut créer un nouvel attribut.

Exemple: on souhaite enregistrer le nombre d'heures travaillées par un employé chaque mois.

On a donc une entité composée de : no employé, année, mois et heures.

Au lieu de choisir comme clé primaire no employé+année+mois, on crée l'attribut **ID_enregistrement**.

Choix de l'identifiant

- Eviter les clés **composées parlantes**. Par exemple le numéro de sécurité sociale dont la forme parlante est : sexe/année_naiss/mois/...

mieux vaut créer un nouvel attribut sans signification pour les utilisateur, de préférence un entier en auto incrémentation. On parle d'**attributs neutre** (cf. normalisation schéma)

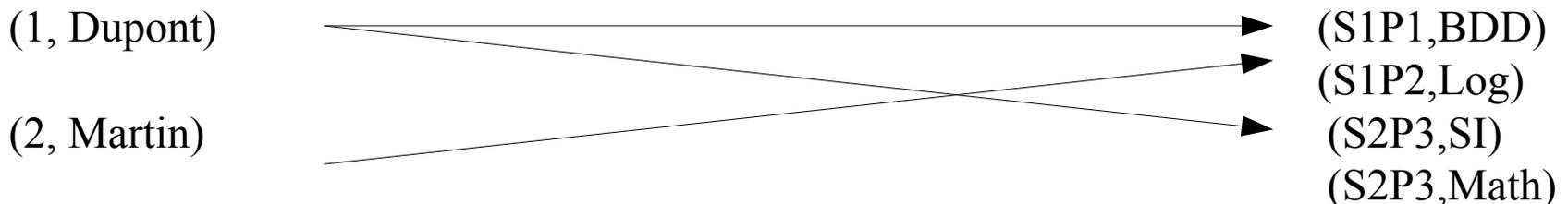
- Dans le même registre on évitera toute clé dont nous n'avons pas le contrôle : qui a été créée et définie par un autre organisme. Risque de modification dans le temps !

L'association

Une **association** représente une possibilité de "lien" entre deux ou plusieurs occurrences d'entités.

Une association n'a **pas d'existence propre** et doit être définie par les occurrences d'entités auxquelles elle est rattachée.

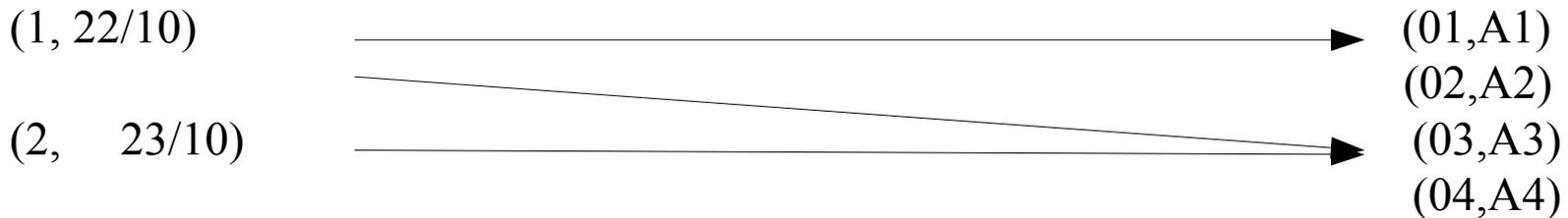
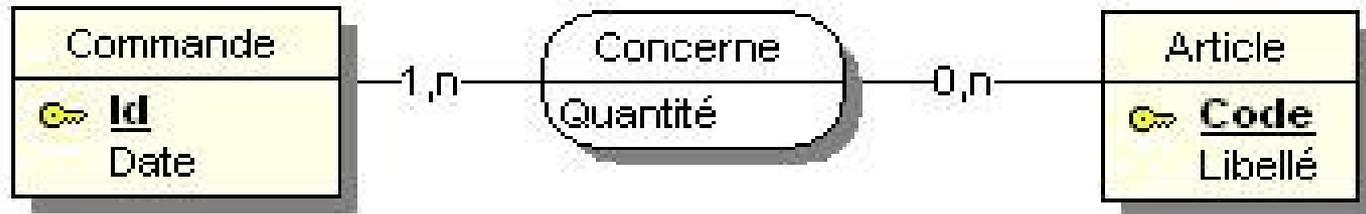
Elle est représentée par une ellipse reliée par des traits aux entités.



Nous avons donc **trois occurrences** de cette association.

Des attributs dans une association

Une **association** peut être porteuse de propriétés



Nous avons deux **occurrences** de cette association

L'attribut **quantité** n'est ni un attribut de Commande, ni un attribut d'Article. Tout comme l'association, il a besoin d'une commande et d'un article pour exister.

Il est recommandé de désigner la relation **par un verbe** qui exprime le rôle d'une entité par rapport à une ou plusieurs autres entités

Une commande concerne des articles – Des articles sont concernés par des commandes

Des attributs dans une association

Très important : l' **identifiant** d'une association = concaténation des identifiants des entités qui lui sont reliées

Donc à une combinaison d'occurrences d'entités composant la collection d'une association il ne peut y avoir au plus qu'une occurrence de cette association.

L'exemple suivant n'est donc pas correcte

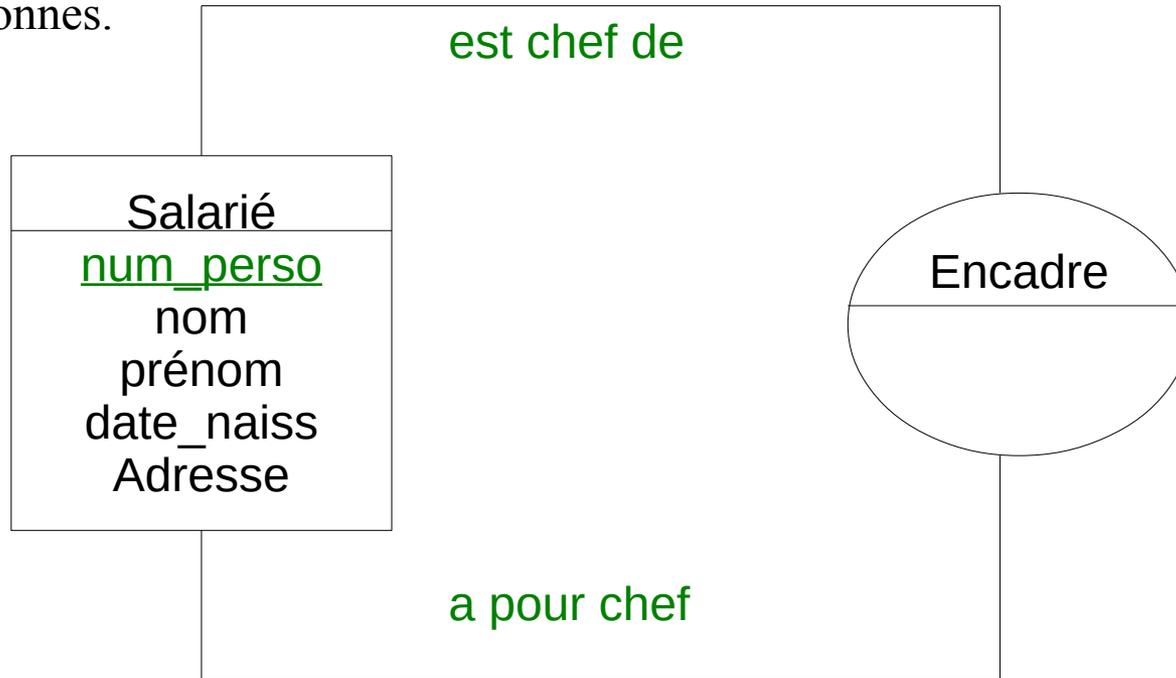


En effet Mr Dupont peut commander 2kg de pommes le 1 janvier et 3kg de pomme le 20 février. Donc pour la combinaison (Id,Code) avec id= identifiant de Dupont et Code = code des pommes on aura deux occurrences possibles de l'association commande (résolution de ce problème dans le prochain exercice)

Association unaire

Une association est dite **unaire** : si elle lie la même entité (réflexive)

Exemple : Chaque salarié a un seul chef **directe** (supérieur hiérarchique **direct**), sauf le patron, et chaque chef a au moins un salarié sous ses ordres. Certains salariés ne sont chefs de personnes.



Dans le cas de relation unaire il est conseillé de nommer chaque patte de l'association on parle alors de rôle : « est chef de » et « a pour chef ». Ceci facilitera grandement le calcul des **cardinalités de chaque pattes**.

Exemple d'association unaire avec attribut

Une pièce entre dans la composition de 0 à plusieurs autres pièces : un pneu entre dans la composition de la pièce « roue ».

Une pièce peut être composée de plusieurs autres pièces : la pièce "roue" est composée d'une pièce "pneu" et d'une pièce "jante".

Une pièce entre dans la composition d'une autre pièce un certain **nombre de fois** : la pièce "voiture" est composée de **4 pièces** "roue" .

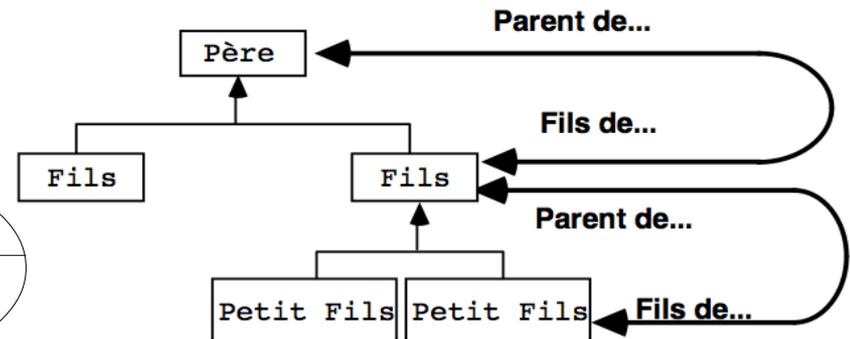
Une pièce entrant dans la composition d'une autre est appelée composant.
Une pièce composée d'autres pièces est appelée composé.



Nomenclature des associations unaires

Deux types de nomenclature, de relation sur la même entité :

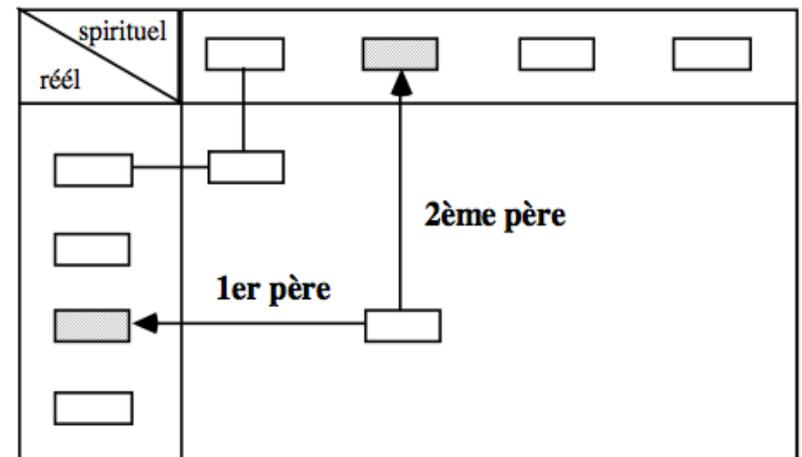
- **Hiérarchique** : un individu n'a qu'un **seul** père



- **Matricielle** : un individu peut avoir **deux** pères :

un réel

un spirituel.



Associations binaires

Une association est dite **binaire** si elle lie deux entités

Exemple : modélisez l'énoncé suivant :

Un Magasin vend des produits a des clients.

Les produits possèdent une référence (un code), un libelle et un prix unitaire.

Les clients ont une identité (nom, prénom, adresse...).

Les clients passent des commandes de produits.

On mémorise la date de la commande.

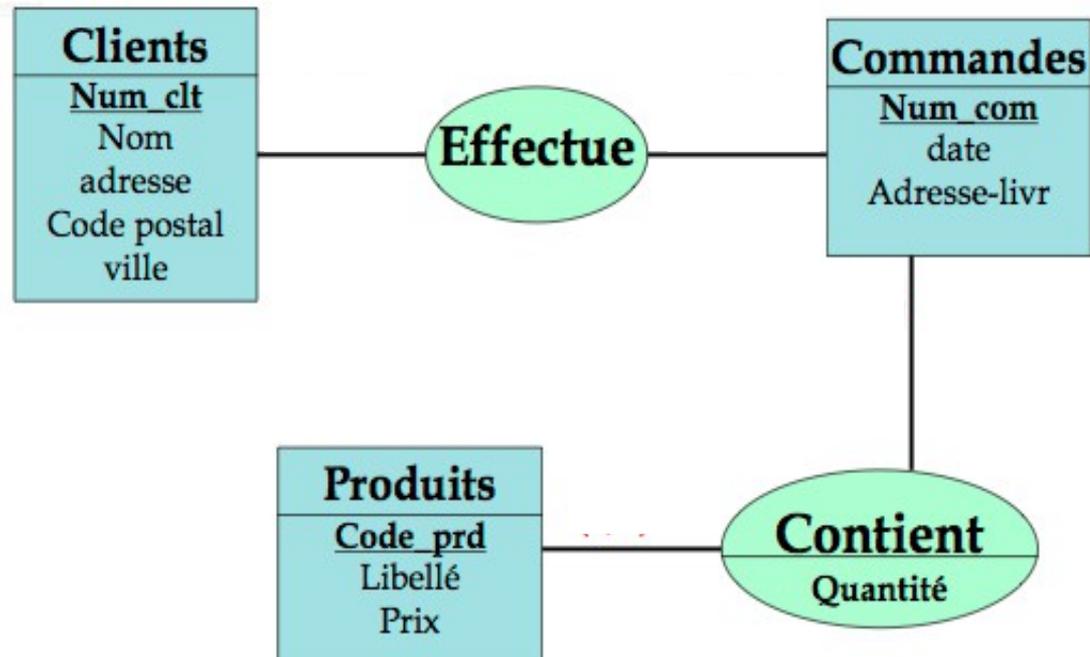
Pour chaque commande, le client précise une adresse de livraison.

La commande concerne un certain nombre de produits, en une quantité spécifiée pour chaque produit.

Associations binaires

- Les entités : Client – Commandes – Produits
- Les association : Effectue entre le client et la commande

Contient entre la commande et le produit



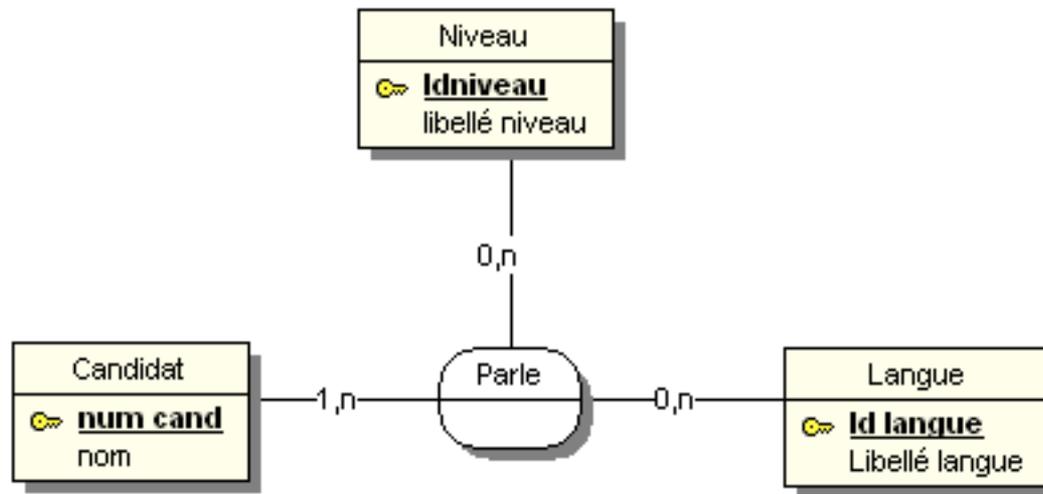
Associations ternaires

Une association est dite **ternaire** si elle lie trois entités. Elle est dite N-aire si elle relie N entités.

Elle peut bien entendu contenir **des attributs** comme pour le cas binaire.

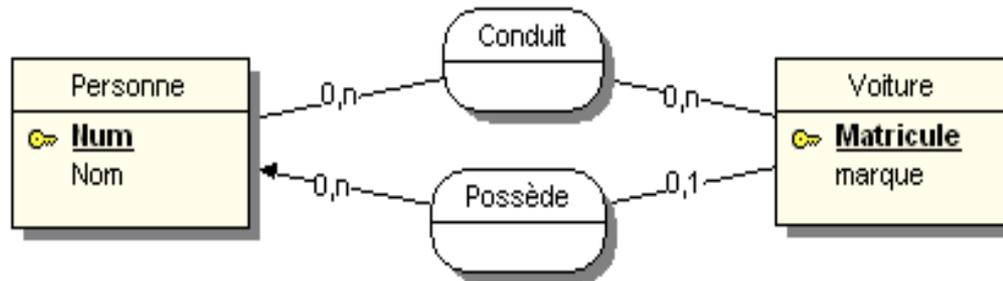
Elle doit cependant obéir à certaines règles de normalisation/décomposition que l'on abordera plus tard.

Exemple : un candidat parle plusieurs langues avec différents niveaux



Remarques

Plusieurs associations peuvent concerner **les mêmes entités**.



Il est temps d'aborder les **cardinalités** et d'une manière générale les **contraintes d'intégrité**.

Les contraintes d'intégrité

Les contraintes sont un complément d'explications, des restrictions indispensables à ajouter sur les modèles de données qui généreront des contrôles ultérieurs.

Un MCD sans contrainte n'est pas complet. Il est nécessaire, de préciser sur un MCD que le propriétaire d'une voiture est **soit** une personne, **soit** une société. De plus, une voiture est la propriété **d'un seul** individu ou de plusieurs.

On appelle règles de gestion, **les contraintes d'intégrité** qui doivent être respectées par le modèle.

Il existe plusieurs types de contraintes :

- **Cardinalité**
- **Intégrité fonctionnelle**
- **Ensembliste**
- **Intégrité syntaxique**
- **Intégrité sémantique**

Les contraintes de cardinalité

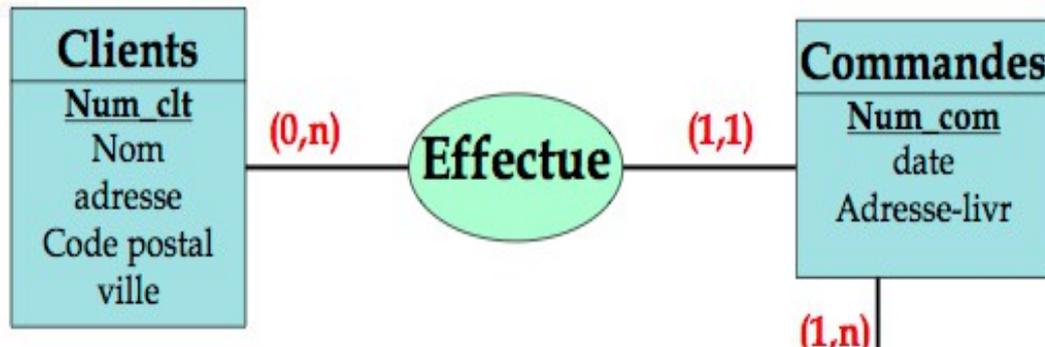
La cardinalité caractérise la participation d'une entité à une association

Elle définit le **nombre d'occurrences** de l'association pour chaque occurrence de l'entité

On distingue :

la cardinalité **minimale** → donne le nombre **minimum** de participation de chacune des occurrences de l'entité à l'association

la cardinalité **maximale** → donne le nombre **maximum** de chacune des occurrences de l'entité à l'association



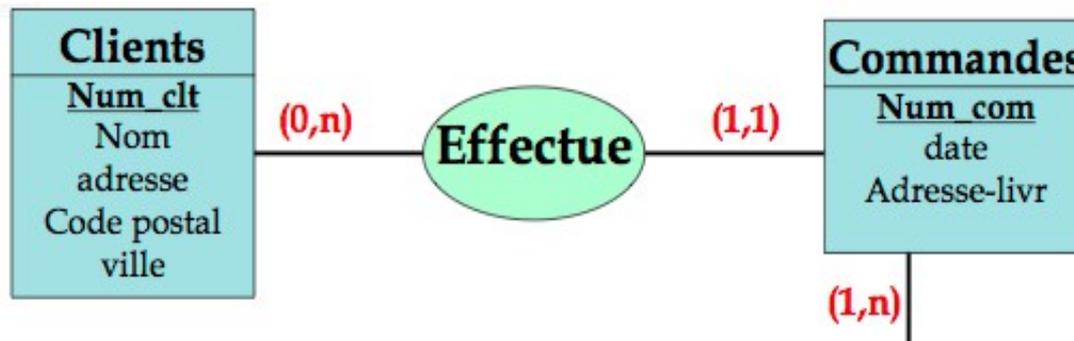
Elle est noté sur chaque patte via un couple (cardinalité minimale, cardinalité maximale)

Comment calculer les cardinalités

Parcourir l'ensemble des occurrences de l'entité et noter pour chaque occurrence le nombre de fois où elle est associée à l'association

S'il existe une occurrence qui n'apparaît pas, la cardinalité min est de 0 sinon elle est de 1

S'il existe une occurrence qui apparaît plusieurs fois, la cardinalité max est n, sinon elle est de 1.



Un client peut effectuer 0 ou plusieurs commandes d'où le $(0,n)$ coté clients.

Une commande est effectuée par un et uniquement un seul client d'où le $(1,1)$ coté commande.

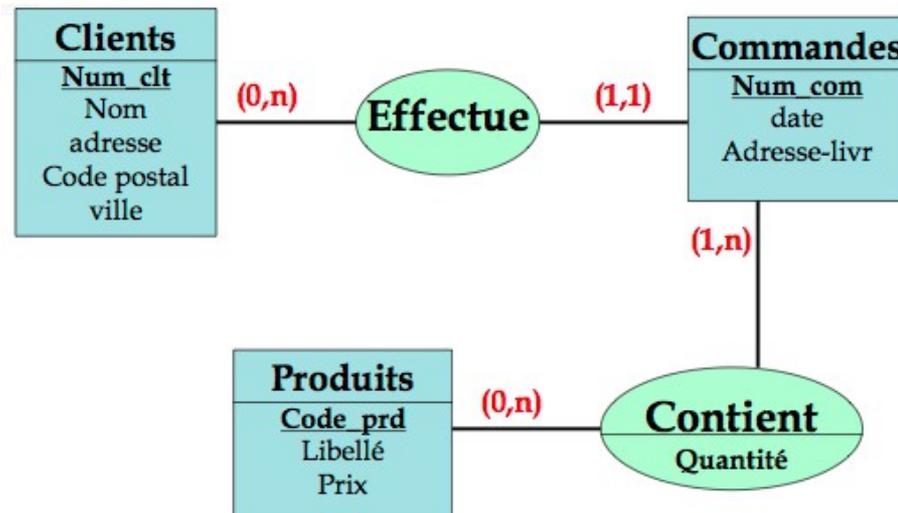
Exemple de cardinalités sur des associations binaires

Un client peut effectuer **0 ou plusieurs commandes** d'où le (0,n) coté clients.

Une commande est effectuée **par un et uniquement un seul** client d'où le (1,1) coté commande.

Une commande contient **au moins un article ou plusieurs** d'où le (1,n) coté commandes

Un produit peut être inclus dans **aucune commande ou dans plusieurs** d'où le (0,n) coté produits.



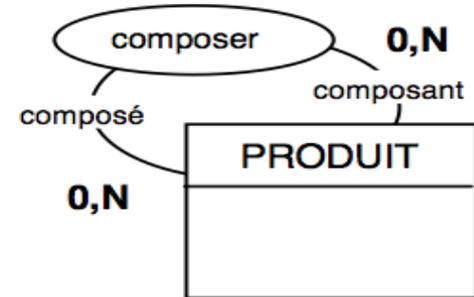
Cardinalités sur des associations unaires

Le fait de nommer les pattes dans les associations unaires permet de plus facilement calculer les cardinalités min et max de chaque patte.

Association matricielle :

Un produit **est composé** de 0 ou de plusieurs autres produits.

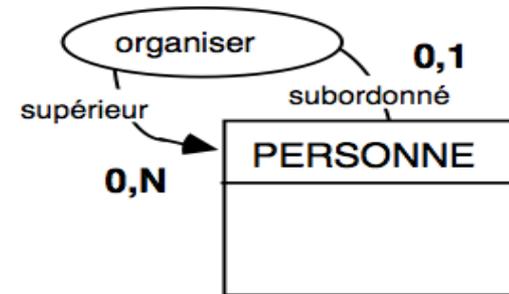
Un produit **est composant** de 0 ou plusieurs autres produits.



Association hiérarchique :

Une personne est le **supérieur** directe de 0 ou de plusieurs autres

Une personne est le sub-ordonné de 0 (le patron) ou d'une personne



Propriété : si la nomenclature est **hiérarchique**, une cardinalité **maximale sera à 1** et la nomenclature peut être représentée par **une flèche (attention au sens de la flèche)**.

Si la nomenclature est non hiérarchique, les cardinalités maximales sont N.

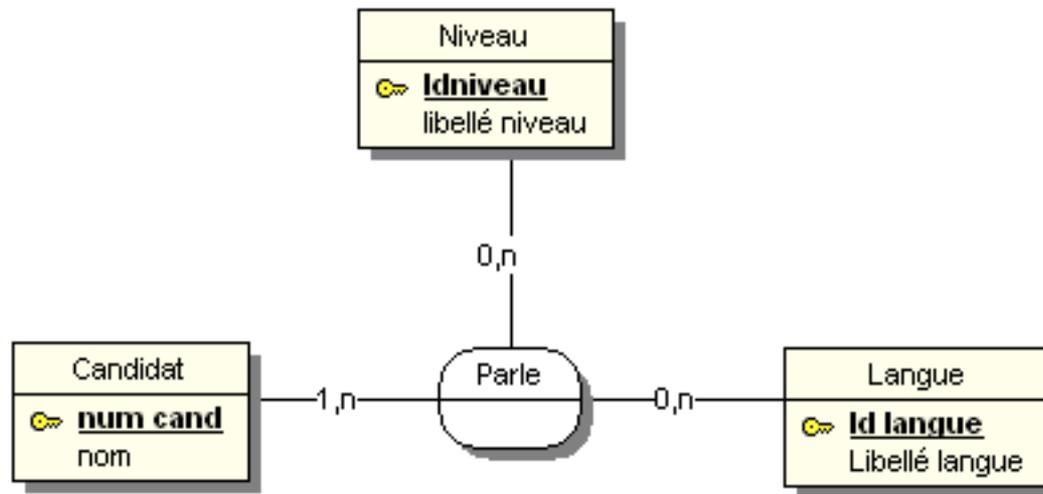
Cardinalités sur des associations N-aires

Il faut calculer les occurrences pour chaque patte.

Pour un candidat X combien de couple $(X, id\ niveaux, id\ langue)$ pouvons nous créer ?

Au moins un \rightarrow le candidat parle uniquement en langue maternelle \rightarrow cardinalité min coté candidat=1

Soit plusieurs \rightarrow le candidat parlent plusieurs langues \rightarrow cardinalité max coté entreprise= n



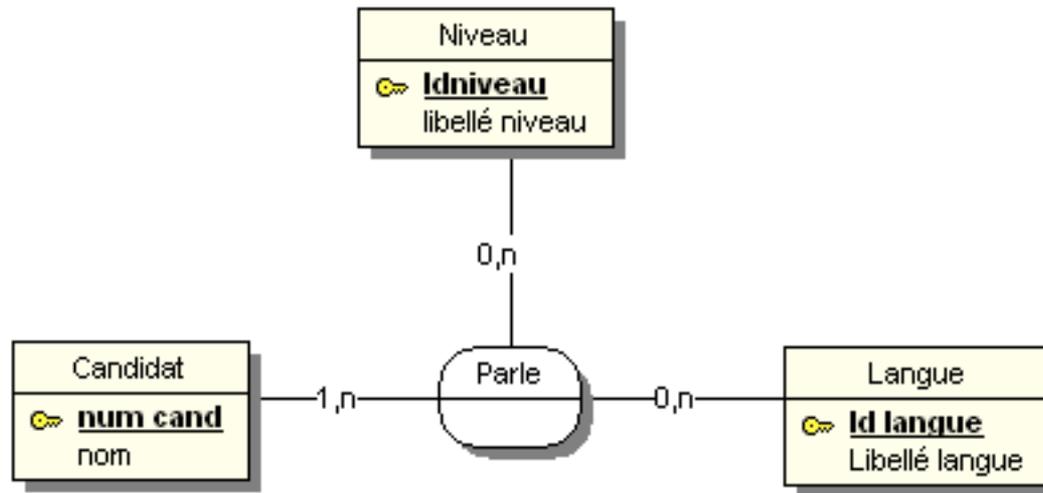
Cardinalités sur des associations N-aires

Il faut calculer les occurrences pour chaque patte.

Pour un niveau X combien de couple (n° candidat, X , id langue) pouvons nous créer ?

Soit aucune \rightarrow pas de candidat parlant une langue avec un niveau $X \rightarrow$ cardinalité min coté Niveau = 0

Soit plusieurs \rightarrow plusieurs candidats parlent une langue avec un niveau X ou un candidat parlent deux langues de niveau X etc \rightarrow cardinalité max coté Niveau = n

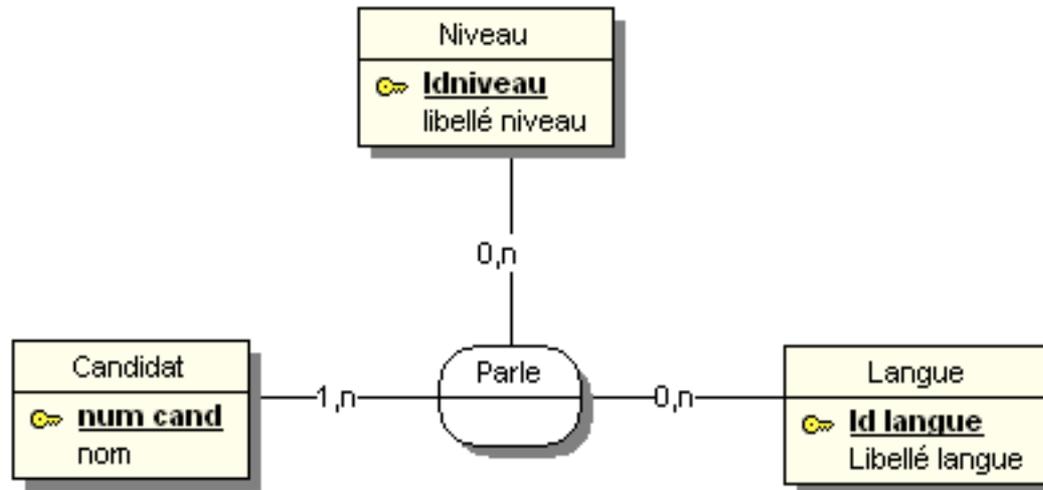


Cardinalités sur des associations N-aires

Pour une langue X combien de couple (n° candidat, id niveau, X) pouvons nous créer ?

Soit aucune → pas de candidat parlant la langue X → cardinalité min coté Langue = 0

Soit plusieurs → plusieurs candidats parlent la langue X → cardinalité max coté Langue = n



Remarque : pour une ternaire, toutes les cardinalités max doivent être égale à N autrement il y a une décomposition possible et/ou une incohérence de modélisation (nous y reviendrons)

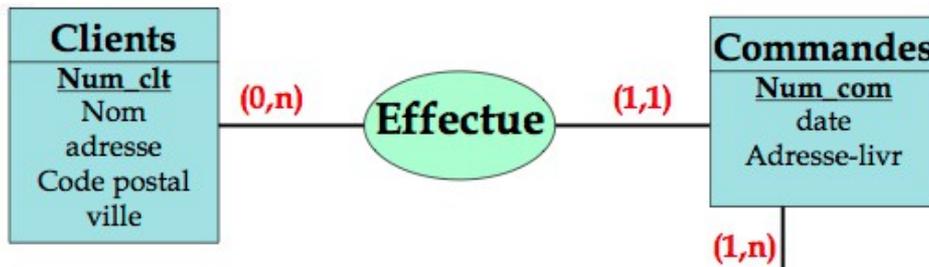
Contrainte d'intégrité fonctionnelle via association binaire

Il y a **dépendance fonctionnelle** (DF) inter entités via **une association binaire** si une des cardinalités max est égale à 1.

On parle de **dépendance forte** pour une cardinalité (1,1) et faible pour (0,1).

Une dépendance forte **n'est pas porteuse de propriétés** → si un tel cas se présente alors tous ces propriétés doivent être déplacées sur l'objet source de la DF

Une commande est liée à un et un seul client → dépendance forte.
Si on connaît la commande alors on connaît le client



Contrainte d'intégrité fonctionnelle via association binaire

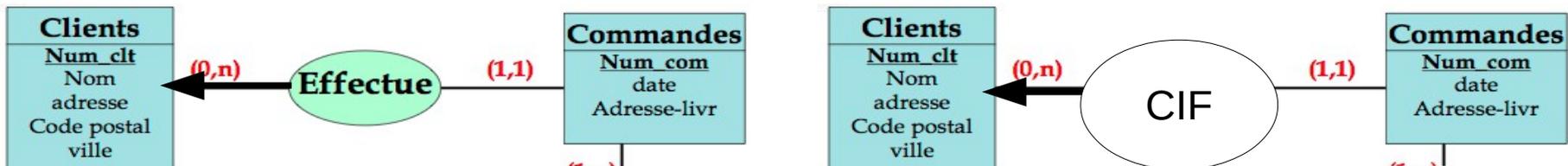
Si le lien est **stable** alors la DF est appelée : **contrainte d'intégrité fonctionnelle**.

Si l'association n'est pas porteuse de propriétés, il arrive que l'on remplace l'association par un **cercle étiquette CIF (ou DF)** en omettant le nom de l'association.

Il est d'usage d'ajouter une **flèche sur l'une des pattes de l'association** : attention au sens de la flèche !

Une commande est liée à un et un seul client → dépendance forte.

Si on connaît la commande alors on connaît le client



Contrainte d'intégrité fonctionnelle via association n-aire ($n > 2$)

Dans une association de dimension > 2 , la notion de DF peut impliquer **tout ou une partie** des entités de la relation et **n'est pas systématiquement liée aux cardinalités**.

Elle est représentée par **un rond noté DF (ou CIF)**, relié à l'association porteuse (**trait pointillé**) et aux entités concernées (**trait plein**) dont l'un est porteur d'une flèche (**la cible de la DF ou CIF**).

Exemple : examen multi sites

- un Candidat est affecté à un **Site**,
- un Site comporte des **Salles**,
- l'examen comporte des **Matières**,
- un Candidat compose une Matière dans une Salle,
- **pour une Matière, un Candidat ne compose que dans une seule Salle (CIF)**

une Matière peut être composée par **plusieurs Candidats** et dans **plusieurs Salles**

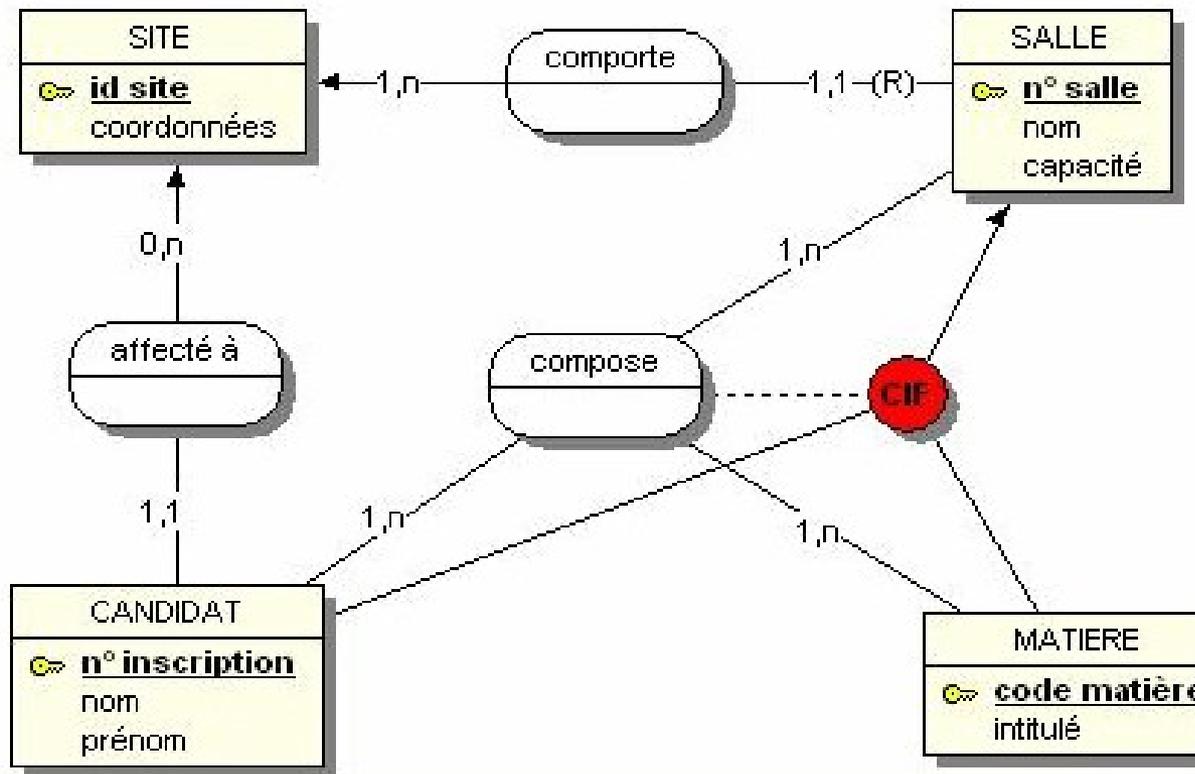
une Salle peut être le lieu de composition **de plusieurs Candidats** et **plusieurs Matières**

un Candidat peut composer dans **plusieurs Matières** et **plusieurs Salles**

Contrainte d'intégrité fonctionnelle via association n-aire (n>2)

Signification de la CIF : pour chaque couple (Matière,Candidat) on associe **une seule** salle d'examen. En d'autre terme la connaissance de la matière et du candidat **implique** la connaissance de la salle.

Le couple (Matière, Candidat) est la source de la CIF. La salle est la cible ou destination.



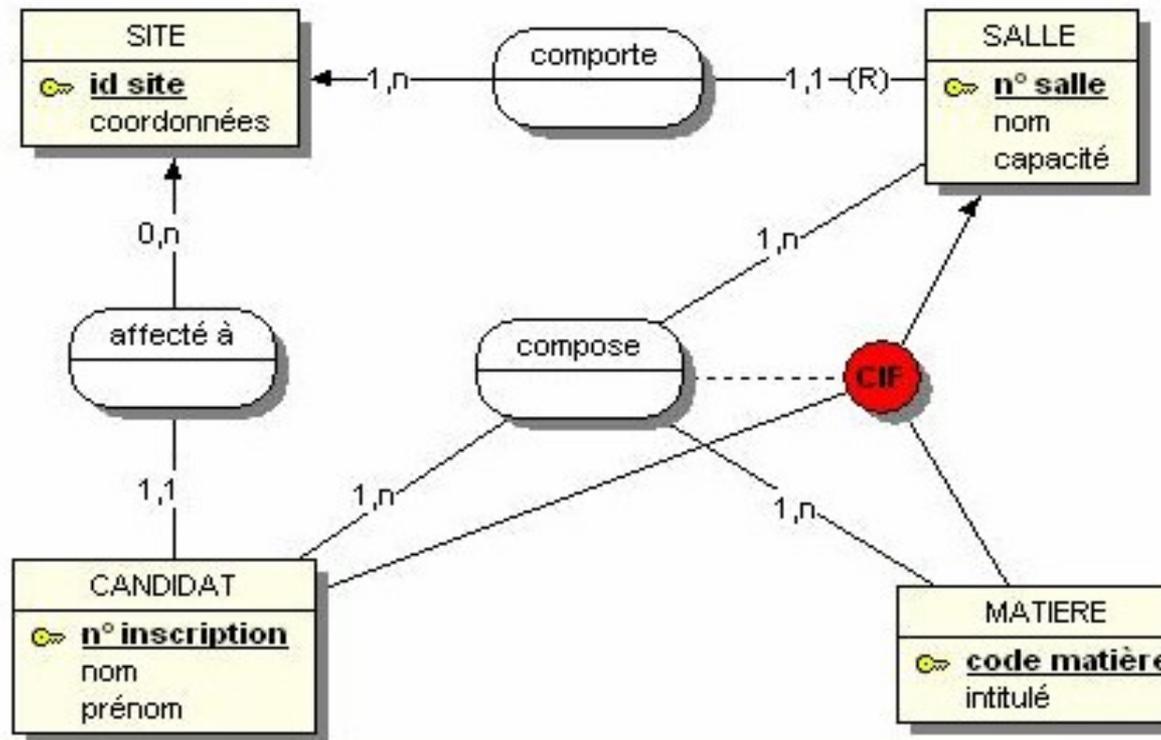
Pourquoi les cardinalités max sont à « n » alors qu'il y a **unicité de la salle** pour le couple (matière,candidat) ?

Contrainte d'intégrité fonctionnelle via association n-aire ($n > 2$)

Pour un candidat X combien de triplets (X , n°salle, code matière) pouvons nous construire ?
(X , E01, BD) , (X , E02 , Logique),... → plusieurs triplets → cardinalité max= n sur candidat

Pour une matière X combien de triplets (n°inscription, n°salle, X) pouvons nous construire ?
(E01, E01, X) , (E02, E01, X),... → plusieurs triplets → cardinalité max= n sur matière

Pour une salle X combien de triplets (n°inscription, X , code matière) pouvons nous construire ?
(E01, X , BD) , (E02, X , BD),... → plusieurs triplets → cardinalité max= n sur salle



Normalisation d'un modèle E/A

Il existe des règles de normalisation de modèle E/A en utilisant entre autres les cardinalités et les CIF. Le but étant de rendre le modèle robuste et d'éviter un maximum de redondance.

Normalisation des noms : l'emploi de noms (entité, association, attribut) **significatifs et uniques**.

Normalisation de l'identifiant :

Éviter les identifiants **composés** de plusieurs attributs

Préférer un **identifiant court** pour rendre la recherche **la plus rapide possible** (éviter notamment les chaînes de caractères comme un numéro de plaque d'immatriculation, un numéro de sécurité sécu)

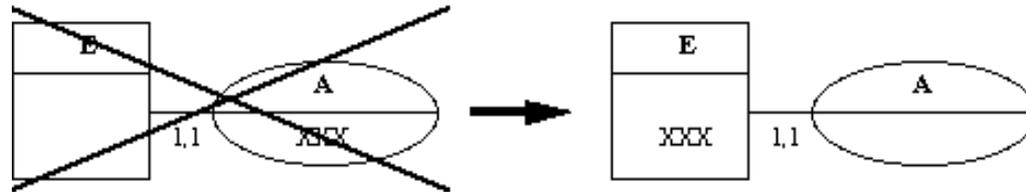
Éviter les identifiants susceptibles **de changer** au cours du temps (comme les plaques d'immatriculation ou les numéros provisoires).

L'identifiant sur un schéma entités-associations normalisée **doit être un entier** , de préférence incrémenté **automatiquement**.

Normalisation des attributs des associations

Les attributs d'une association doivent **dépendre directement** des identifiants de **toutes les entités** en association.

Si ce n'est pas le cas, **déplacer les attributs** dans les entités ou **créer une nouvelle entité**

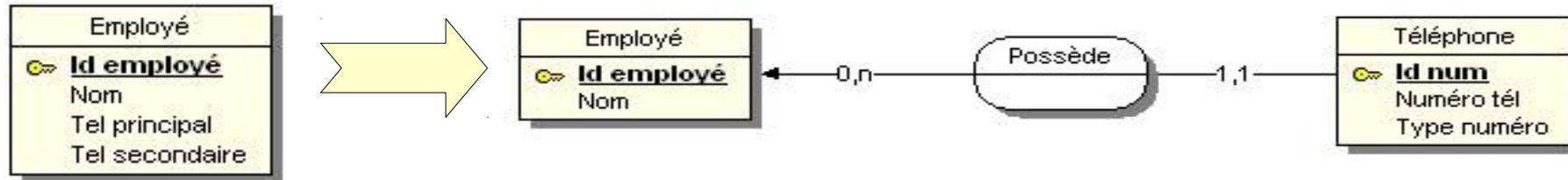


On retrouve ci-dessus le fait que toute **dépendance forte** ne peut être porteuse de propriétés (cf. CIF)

Normalisation des attributs : en plusieurs exemplaires / calculables

- Remplacer les attributs en **plusieurs exemplaires** en une association supplémentaire de cardinalités maximales n.

En effet, les attributs en plusieurs exemplaires posent des problèmes **d'évolutivité** du modèle → sur l'exemple gauche comment faire si un employé a **deux** téléphones secondaires ?



- Supprimer tout attribut **calculable** (ou déductibles) à partir des autres → risque d'incohérence entre les valeurs des attributs de base et celles des attributs calculés



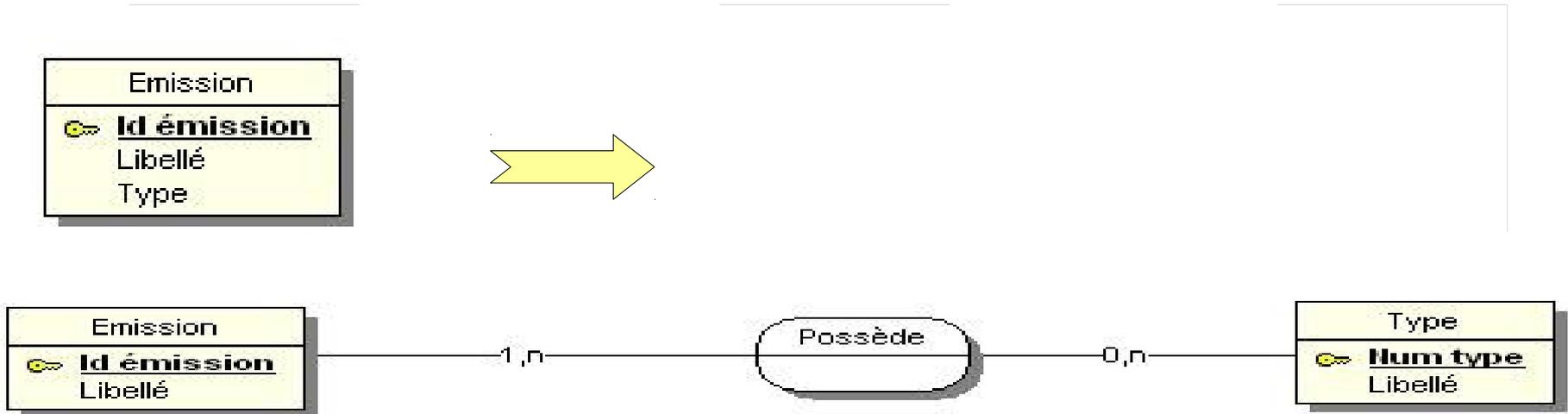
L'attribut « montant » est supprimé du fait que **Montant=Quantité * Prix unitaire**

Normalisation des attributs : type énumérée

- Un attribut correspondant à un **type énuméré** est généralement avantageusement remplacé par une nouvelle entité liée l'entité dans laquelle figurait cet attribut.

Exemple : l'attribut « **Type** » caractérise le type d'une émission et peut prendre des valeurs comme : actualité, culturelle, reportage, divertissement, etc.

Remplacer cet attribut par une entité permet, d'une part, d'augmenter **la cohérence** (en s'affranchissant, par exemple, des variations du genre culturelle, culture, Culture, ...) et d'autre part, si les règles de gestion le permettent, de pouvoir affecter **plusieurs types** à une même entité (par exemple : actualité et culturelle)

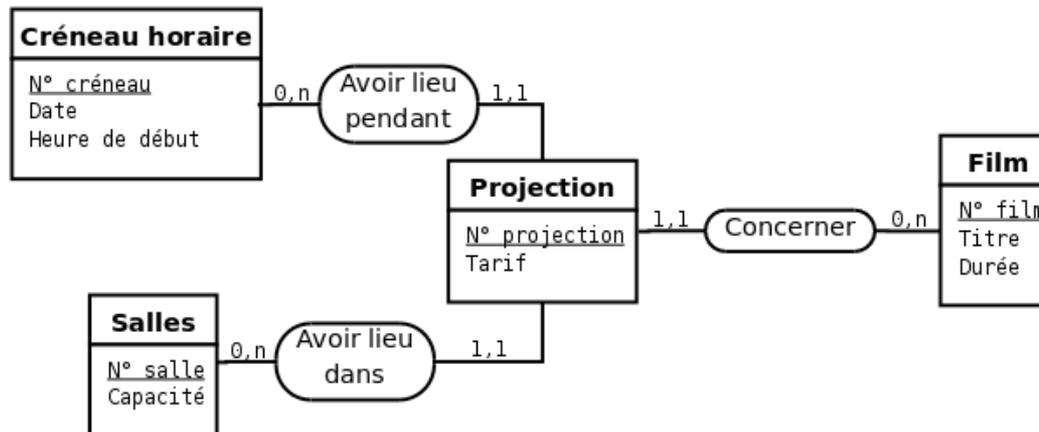


Normalisation des entités : entité remplaçable par une association

Toutes les entités qui sont **remplaçables par une association** doivent être remplacées :

Lorsqu'autour d'une entité, toutes les associations ont **pour cardinalités maximales 1 au centre et n à l'extérieur**, cette entité est candidate pour être remplacée par une association branchée à toutes les entités voisines avec des cardinalités identiques 0,n.

MAIS il faut au préalable toujours vérifier la satisfiabilité de la propriété d'unicité entre les identifiants des entités de cardinalité n et l'identifiant de l'entité à remplacer.



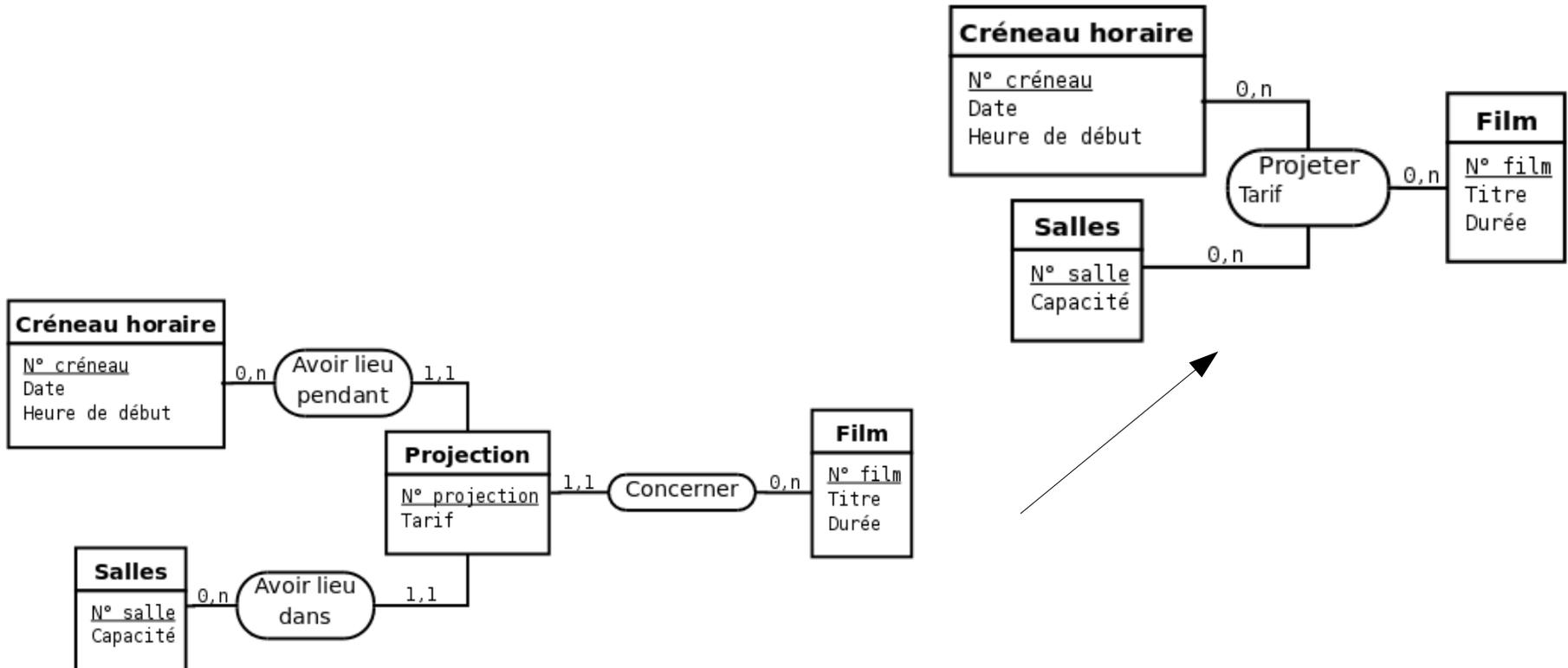
Normalisation des entités : entité remplaçable par une association

L'entité « Projection » est candidate à être remplacée

Sur l'exemple du fait que

(N° créneau, N°salle, N°film) détermine un et un seule N° projection

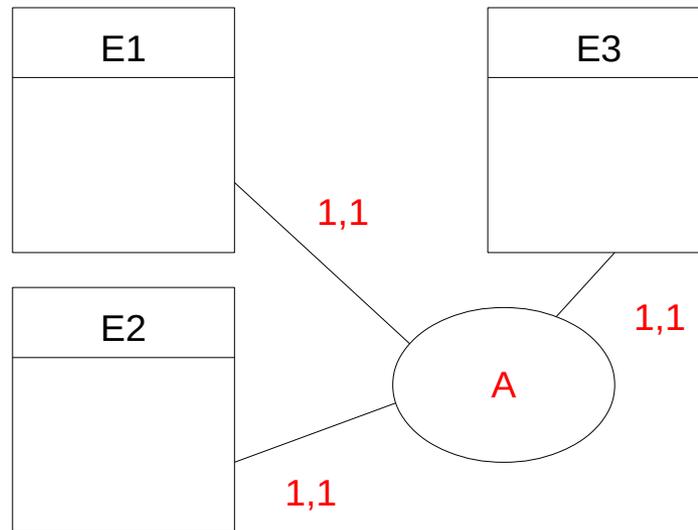
alors on peut appliquer la normalisation et remplacer « projection » par une association.



Normalisation des associations : association fantôme

Supprimer les associations fantômes : Une association fantôme est une association dans laquelle toutes les entités appartenant à sa collection sont impliquées une et une seule fois.

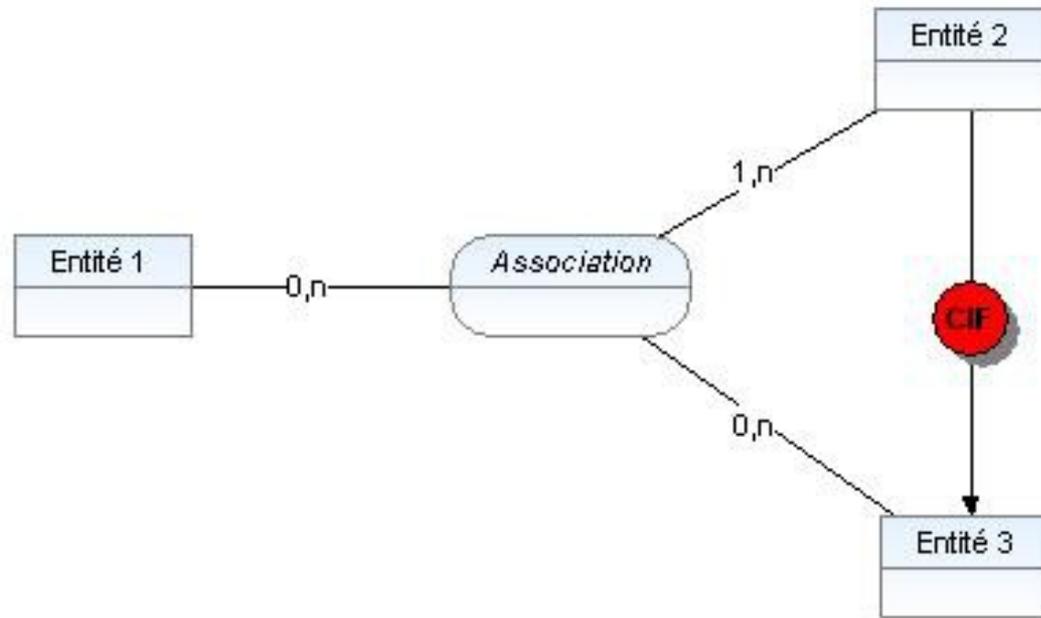
Dans l'exemple ci-dessous, une occurrence de E1 n'est liée qu'à une et une seule occurrence de « A » ; même contrainte pour les occurrences de E2 et de E3. Il y a, en fait, sur-structuration : E1, E2 et E3 **sont une seule et même entité** et « A » **n'a pas lieu d'être**.



Normalisation des associations : décomposition (suite)

La seconde intervention a lieu en présence de **CIF**.

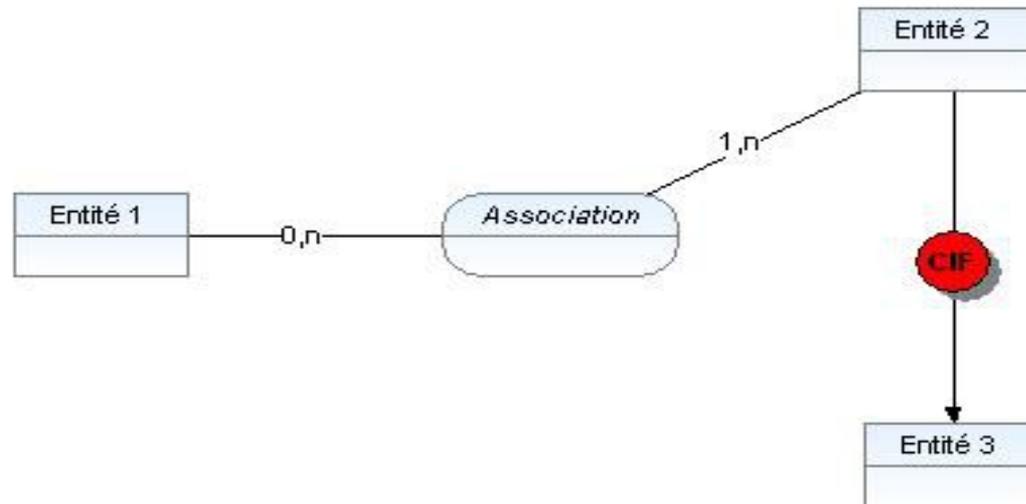
Si la **dimension** de la CIF (nombre d'entités impliquées) est **inférieure** à celle de **l'association** porteuse, alors on doit envisager une **décomposition** de la relation en deux relations : la première représente la CIF et la seconde l'ancienne association amputée de sa CIF.



Normalisation des associations : décomposition (suite)

On crée une association binaire représentant la CIF : c'est l'association « **Association2** »

On maintient l'ancienne association « **Association 1** » mais on lui supprime la destination de la CIF, c'est-à-dire l'entité « Entité 1 »



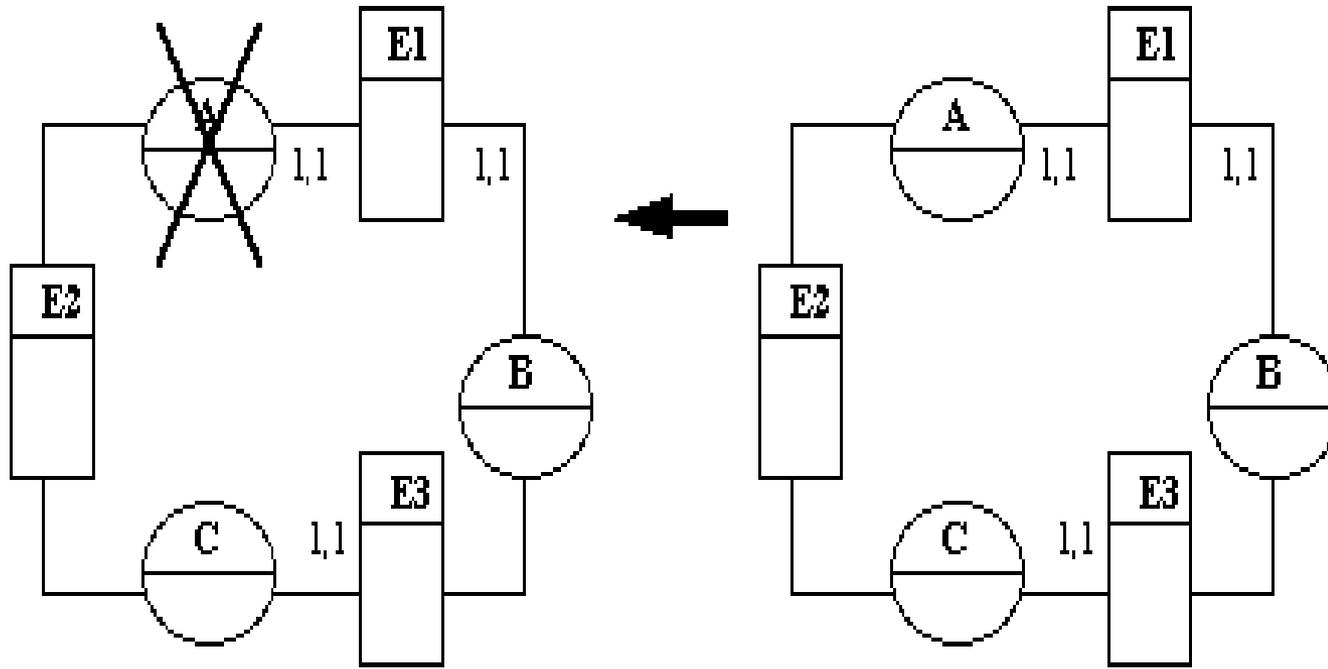
Cette règle de décomposition se généralise pour toute association n-aire avec $n > 2$

Normalisation des associations : association redondante

Supprimer les associations **redondantes**

Si pour chaque occurrence de l'entité de départ on arrive à **la même occurrence** de l'entité d'arrivée.

On va supprimer le plus court chemin (le plus « pauvre »).



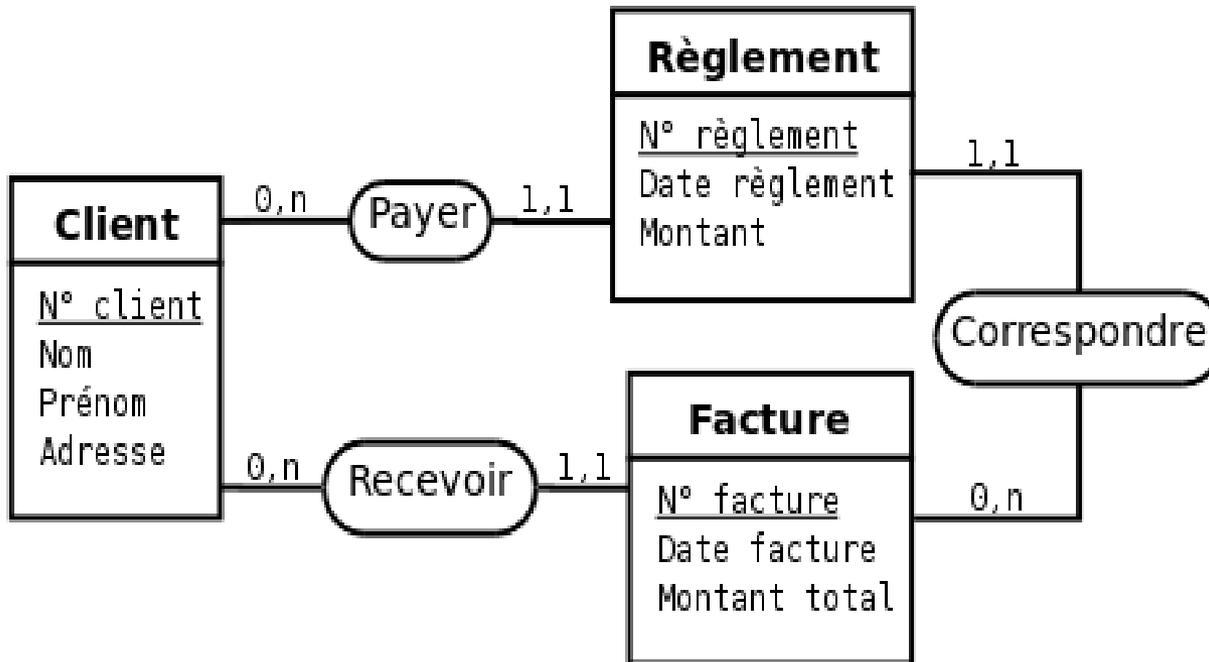
Les deux chemins doivent avoir la même durée de vie.

Normalisation des associations : association redondante (suite)

Dans notre exemple on **supprime l'association « payer »** : on peut retrouver le client qui a payé le règlement en passant par la facture qui correspond.

En effet : pour un règlement X il existe **une seule** facture Y et du fait du 1,1 coté facture il existera qu'**un seul client**.

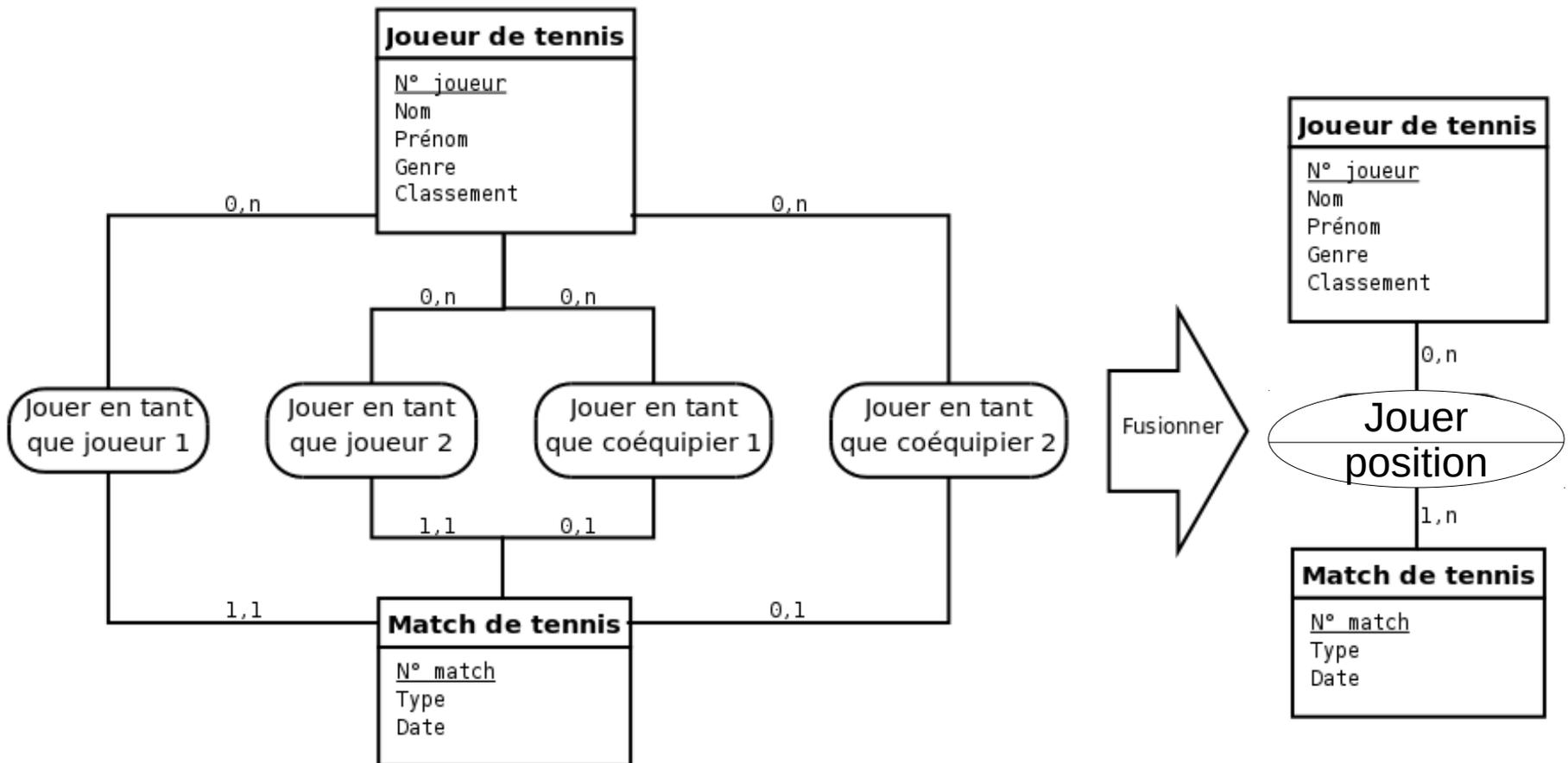
Le passage du règlement vers le client à travers « payer » est donc bien **redondant**.



Normalisation des associations : associations similaires

Fusionner les associations **similaires** → introduction d'un **attribut supplémentaire** dont l'ensemble des valeurs possibles est l'ensemble des noms des type-associations factorisés.

Par exemple : modélisation de matchs d'un tournoi de tennis. Au lieu de maintenir 4 association de structures similaires on **les fusionne** en une seule.



Normalisation des cardinalités

Il est d'usage de ne retenir que trois valeurs pour les cardinalités : 0, 1 et N.

La raison de cette restriction est à rechercher dans la volonté de construire un système d'information **qui dure le plus longtemps possible**.

Fixer à 37 le nombre maximum de commandes prises par un client donné c'est courir le risque de devoir **modifier le logiciel** le jour où, exceptionnellement, on rencontre un client qui passe 38 commandes.

Les entorses à cette norme seront l'exception. Elles devront être justifiées et approuvées par **l'utilisateur** du système d'information

On les programmera alors sous forme de contrôles via des **déclencheurs** en SQL par exemple.

Construction d'un MCD

On distingue deux démarches de construction du MCD :

(A) une démarche **déductive** qui s'appuie sur l'analyse des documents manipulés dans le domaine (utilisé en MERISE).

(B) une démarche **inductive** qui cherche à mettre en évidence rapidement les différents concepts évoqués dans le discours, puis à les décrire par des informations

Les étapes :

1. identifier les entités en présence.
2. lister les attributs de chaque entité.
3. exhiber des identifiants.
4. établir les associations entre entités.
5. normaliser.

Application : garage automobile

Lors de l'entrée d'un **véhicule** au garage, un **numéro d'intervention** est affecté au véhicule. Chaque **intervention** est formée d'une suite **d'opérations**. Chaque opération, identifiée par un code spécifique, correspond à un travail spécifique caractérisé par une dénomination (réglage parallélisme, vidange, etc.) et un descriptif.

Identifiée par son numéro, une **intervention** est définie par un libellé (révision des 30000 km, etc.), la date d'entrée du véhicule pour intervention et le kilométrage du véhicule concerné.

Les caractéristiques retenues pour un **véhicule** sont son numéro d'immatriculation et son année de mise en circulation. Le **type de véhicule** comprend un code, la marque du constructeur et l'appellation commerciale du modèle de véhicule (Mégane, 607, New Beetle, etc.). Les caractéristiques d'un **client** sont un numéro qui lui est affecté, son nom et prénom, son adresse (rue, code postal, ville) et un numéro de téléphone.

Dans le cadre d'une **intervention** sur un véhicule donné, une **opération** a **une durée variable** et peut occasionner le remplacement de **pièces**, d'une **certaine quantité**, compatibles avec le véhicule concerné.

Les caractéristiques d'une **pièce** sont son code, un libellé et le prix unitaire de la pièce. Un **coût horaire** de main d'œuvre dépendant du code opération et du type de véhicule est défini par le garage. Les caractéristiques d'une **facture** sont son numéro et sa date d'émission.

Application : garage automobile

Les entités :

Intervention

Véhicule

Client

Opération

Pièce de rechange

Facture

Type véhicule

Application : garage automobile

Les entités : Intervention – Véhicule – Client – Opération - Pièce de rechange -Facture

Type véhicule

Les associations évidentes :

intervention ↔ véhicule (CIF)

véhicule ↔ client (CIF)

véhicule ↔ type véhicule (CIF)

Facture ↔ intervention (CIF)

pièce de rechange ↔ type véhicule

Il manque 3 associations.

Problème 1 : ou mettre l'attribut « main d'oeuvre » ?

Application : garage automobile

Problème : ou mettre l'attribut « main d'oeuvre »

Solution : on crée une **association** entre « **type véhicule** » et « **opération** » en mettant l'attribut **main d'oeuvre** en attribut d'association.

Attention à la **durée d'une opération**, elle est variable selon l'intervention et on n'est pas un attribut de l'opération. Par exemple pour la 206 de Mr Dupont l'opération vidange a duré 30minutes alors qu'elle a duré 1heure sur la 206 de Mr Martin.

Problème : ou mettre alors cet attribut ?

Application : garage automobile

Problème : ou mettre l'attribut « durée » de l'opération ?

Solution : on crée l'association entre « intervention » et « opération » en mettant « durée » en attribut d'association. Par la CIF entre intervention et client on retrouve le client à partir d'une intervention. La durée est donc saisie à chaque fois qu'il y a une intervention et une opération. C'est donc un attribut d'association entre ces deux entités.

Attention à ne pas ajouter « client » dans l'association autrement la règle de décomposition pourrait s'appliquer vu la présence de la CIF avec « intervention ».

Problème : ou mettre la quantité d'une pièce X utilisée pour une opération Y ?

Indice : que faut-il connaître pour déterminer le nombre de pièces utilisées ?

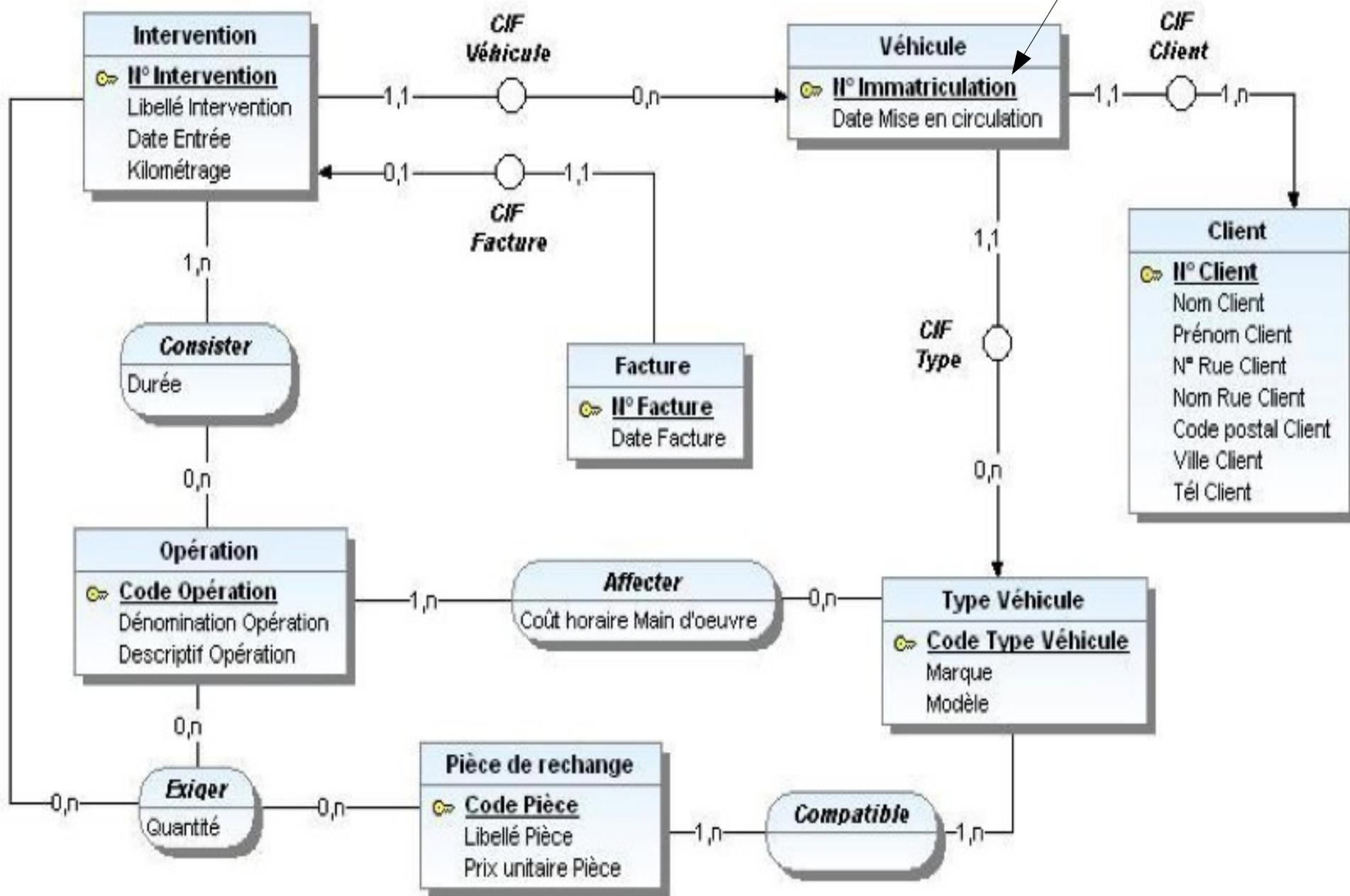
Application : garage automobile

Problème : ou mettre la quantité d'une pièce X utilisée pour une opération Y ?

Solution : on crée une association ternaire entre « intervention », « pièce de rechange » et « opération ». L'attribut nb pièce est alors attribut de cet association.

Attention à ne pas ajouter « client » dans l'association autrement la règle de décomposition pourrait s'appliquer vu la présence de la CIF avec « intervention ».

À éviter



Application : gestion des stages

Un stage d'une durée de quatre (resp. six) mois environ est prévu en fin de licence (resp. master). Ces stages sont obligatoires et commencent début avril.

Une convention de stage règle les rapports entre l'entreprise et l'université d'Orléans. La secrétaire dispose d'une liste d'entreprises ayant déjà pris des étudiants en stage.

Suite à des acceptations de candidatures d'étudiant en stage, les conventions de stage sont signées.

En cours de stage, chaque stagiaire se voit désigner un parrain, enseignant du département Informatique, qui doit assurer le suivi de stage de l'étudiant en lui rendant visite dans l'entreprise d'accueil et qui doit veiller au bon déroulement du stage.

Afin de mieux répartir les enseignants pour les suivis de stage, le responsable des stages doit connaître les entreprises que les enseignants ont déjà visitées lors des stages des années précédentes.

Application : gestion des stages

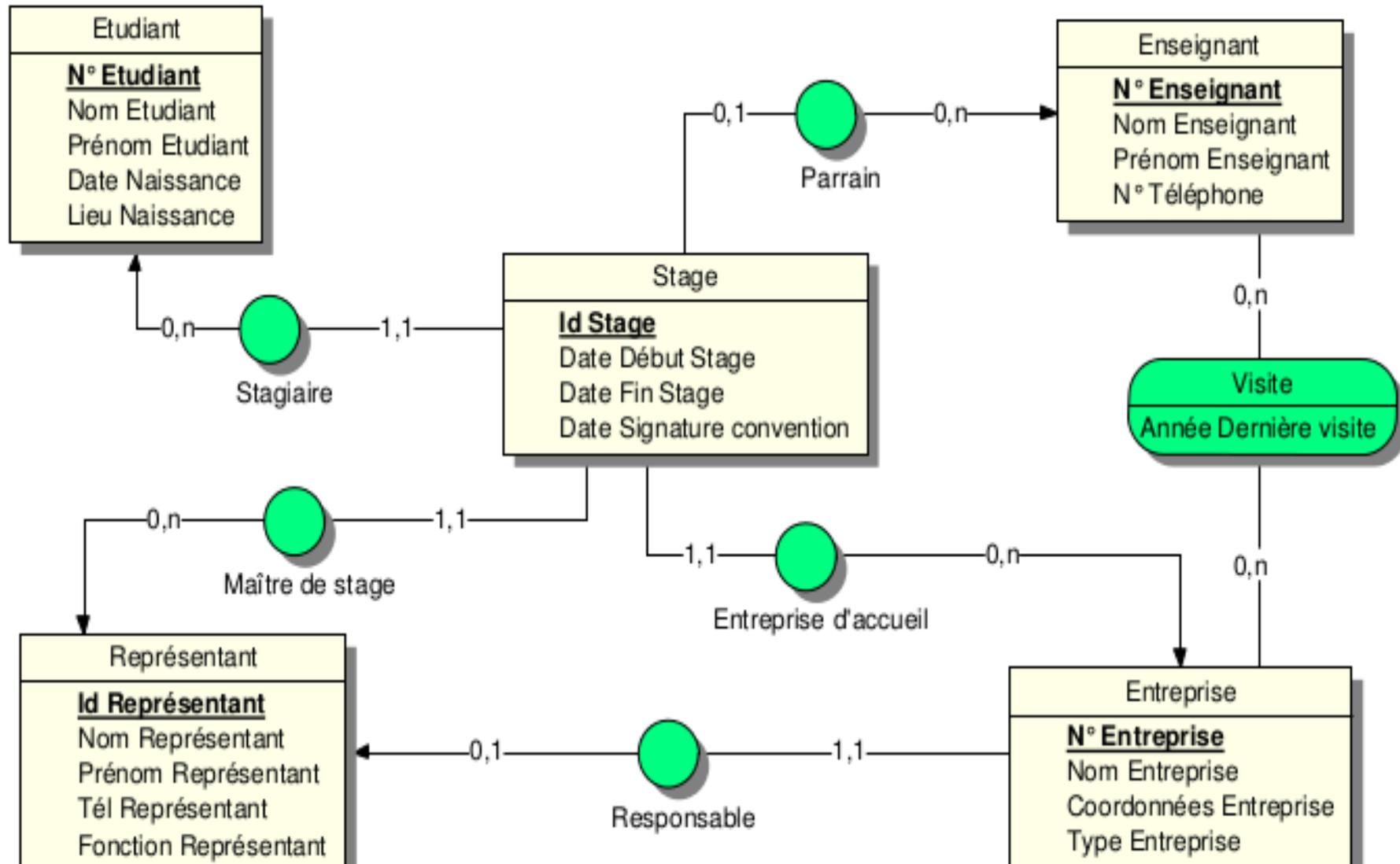
pour chaque entreprise : raison sociale, coordonnées, type (administration, SSII, etc.) et des informations relatives à un responsable légal de l'entreprise (nom, prénom, fonction, téléphone) signataire des conventions de stage ;

pour chaque étudiant : numéro, nom, prénom, date et lieu de naissance et, concernant chaque stage, le nom de son entreprise d'accueil, des informations relatives au maître de stage (nom, prénom, fonction, téléphone), la période de stage et la date de la signature de la convention ;

pour chaque enseignant : nom, prénom, numéro de téléphone, les entreprises qu'il a déjà visitées, avec, pour chacune, l'année de sa dernière visite.

Une entreprise peut accueillir plusieurs stagiaires simultanément. Dans ce cas, tous les stagiaires de l'entreprise n'ont pas nécessairement le même maître de stage, ni le même parrain.

Application : gestion des stages



Commentaires :

Lors de la normalisation on pourrait transformer l'entité « stage » en une association vu que les cardinalités sont à 1 au centre et n en dehors mais la règle d'unicité ne sera pas respecté car un étudiant peut passer son stage de L3 en 2011 puis le même stage avec les mêmes acteurs en M2.

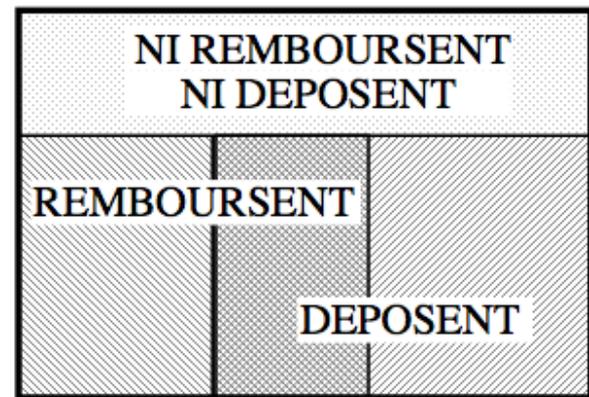
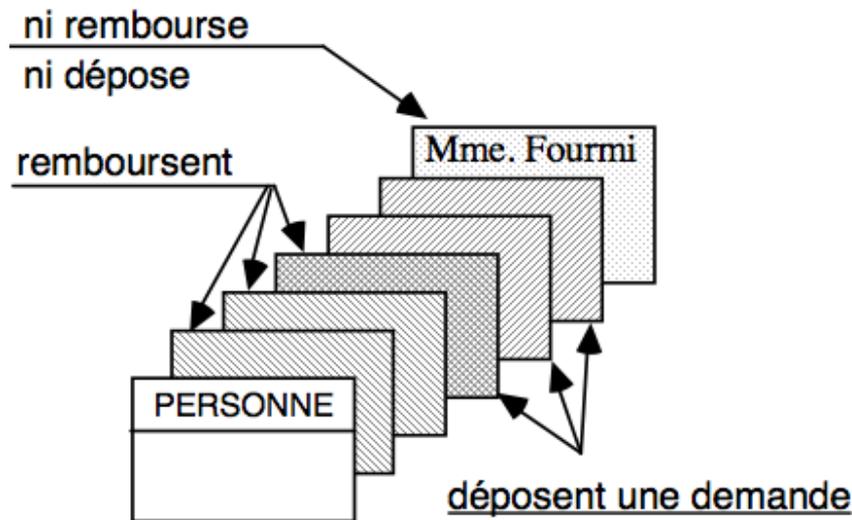
MCD étendus

Contraintes ensemblistes

Considérons les personnes ayant déposé des **demandes de prêts** et celles qui remboursent des prêts. Elles forment **deux sous-ensembles** des personnes

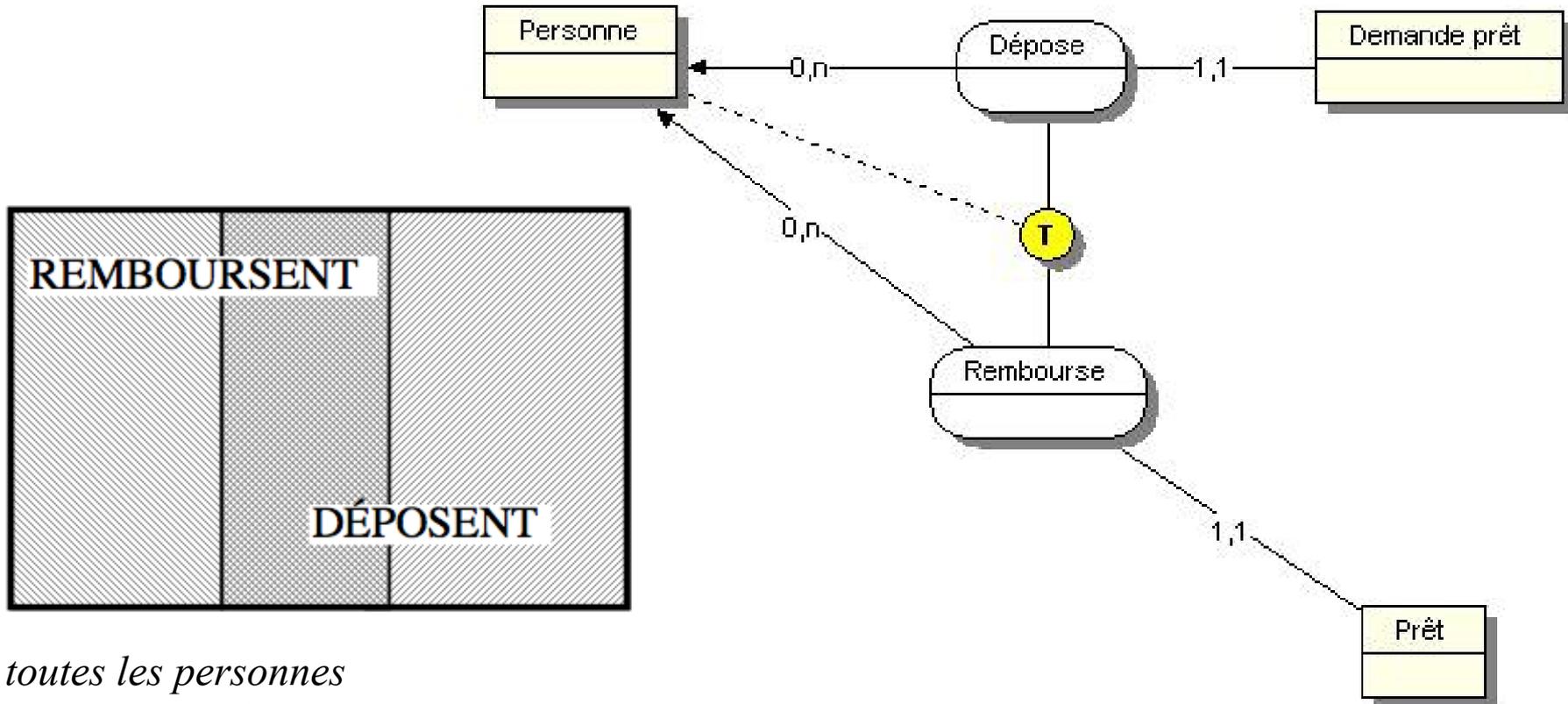
Ces deux sous-ensembles ont une **intersection**, ceux qui empruntent et qui remboursent et une réunion ceux qui empruntent ou qui remboursent.

Il y a aussi un ensemble de personnes qui ni emprunte ni rembourse.



Le ou inclusif « T »

toute occurrence du pivot *apparaît dans au moins une* occurrence des associations.
Notée **T** pour total.

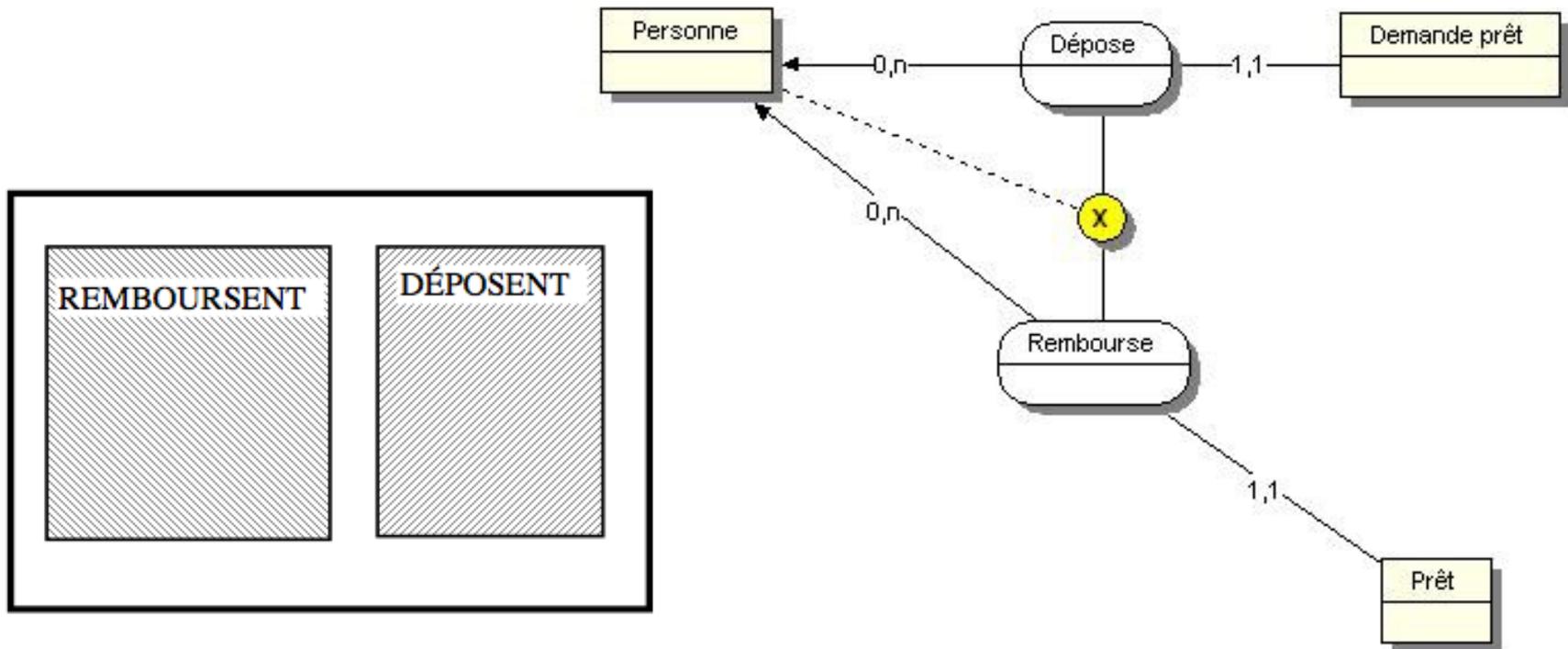


toutes les personnes
dépose *et* rembourse.

On met un *trait en pointillé* pour spécifier l'ensemble contraint. Ici c'est l'entité « personne »

Le ou exclusif « X »

l'exclusion (X) où toute occurrence du pivot appartient à au plus un des ensembles d'occurrences participant aux relations considérées. L'intersection est vide.

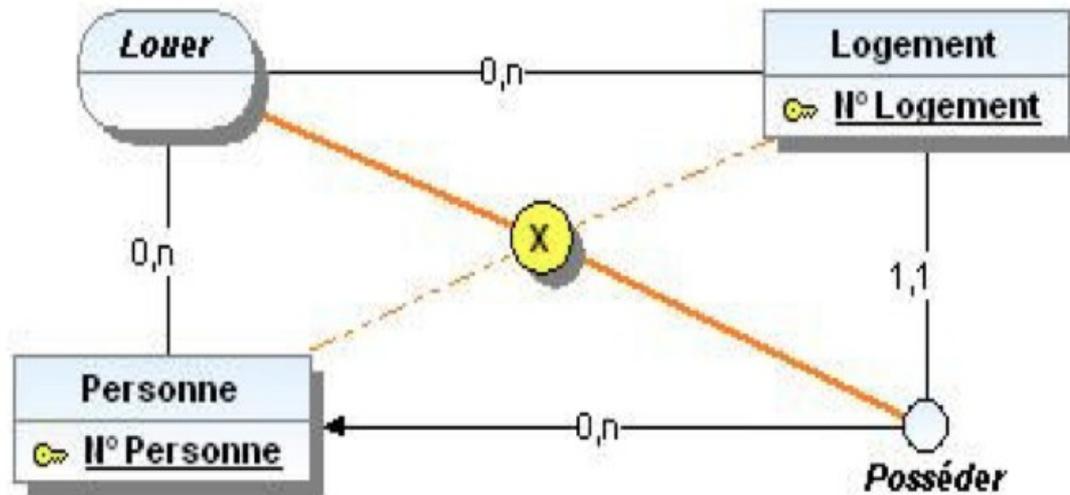


Une personne ne peut demander un prêt et rembourser un autre.

On met un *trait en pointillé* pour spécifier l'ensemble contraint. Ici c'est l'entité « *personne* »

Le ou exclusif « X »

Possibilité d'avoir des contraintes d'exclusion d'une association sur d'autres associations.

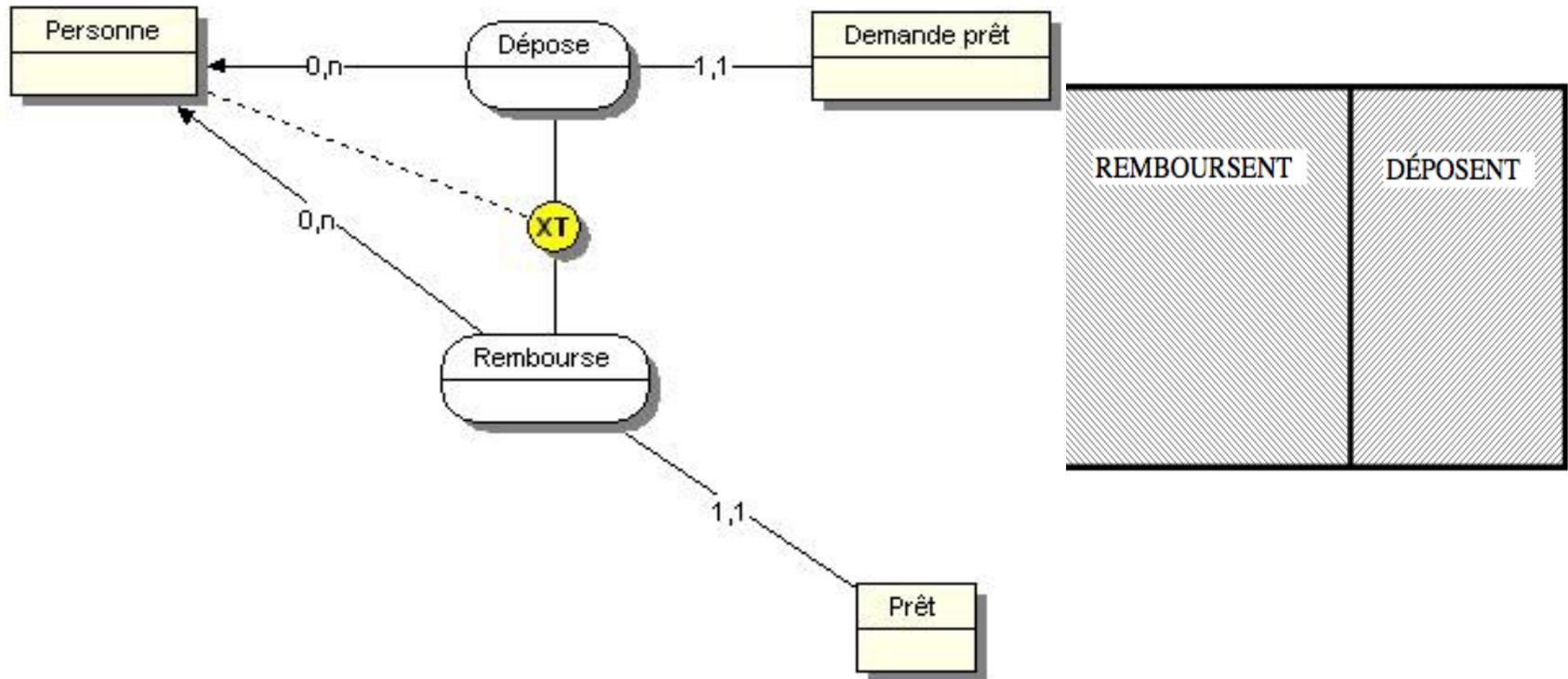


Une occurrence (personne,logement) ne peut être en même temps occurrence de l'association Posséder et occurrence de l'association Louer.

→ une personne ne peut pas être propriétaire d'un appartement X et locataire de l'appartement X.

La partition (XT)

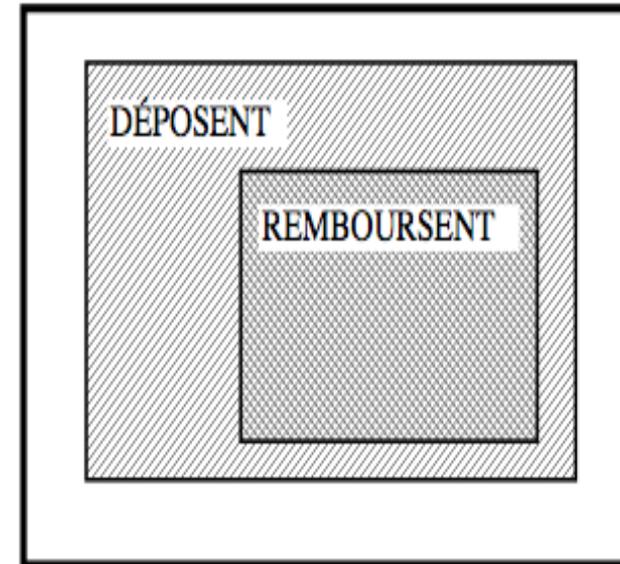
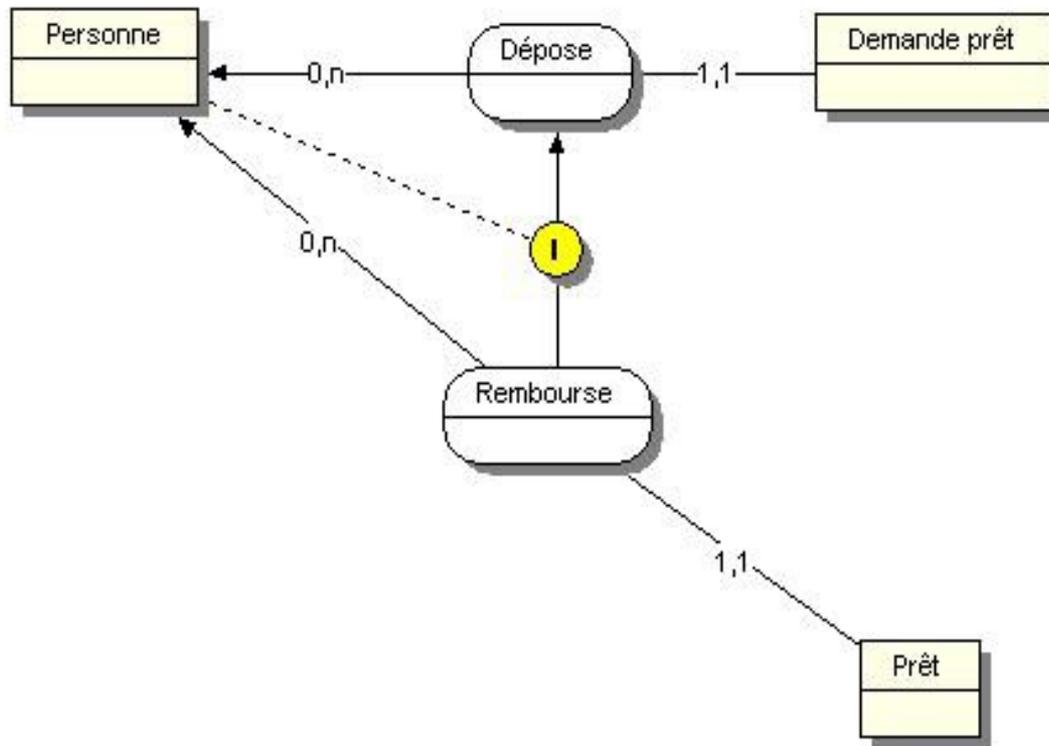
La partition où toute occurrence du pivot appartient à **un et un seul ensemble** c'est la



On met un **trait en pointillé** pour spécifier l'**ensemble constraint**. Ici c'est l'entité « *personne* »

L'inclusion (I)

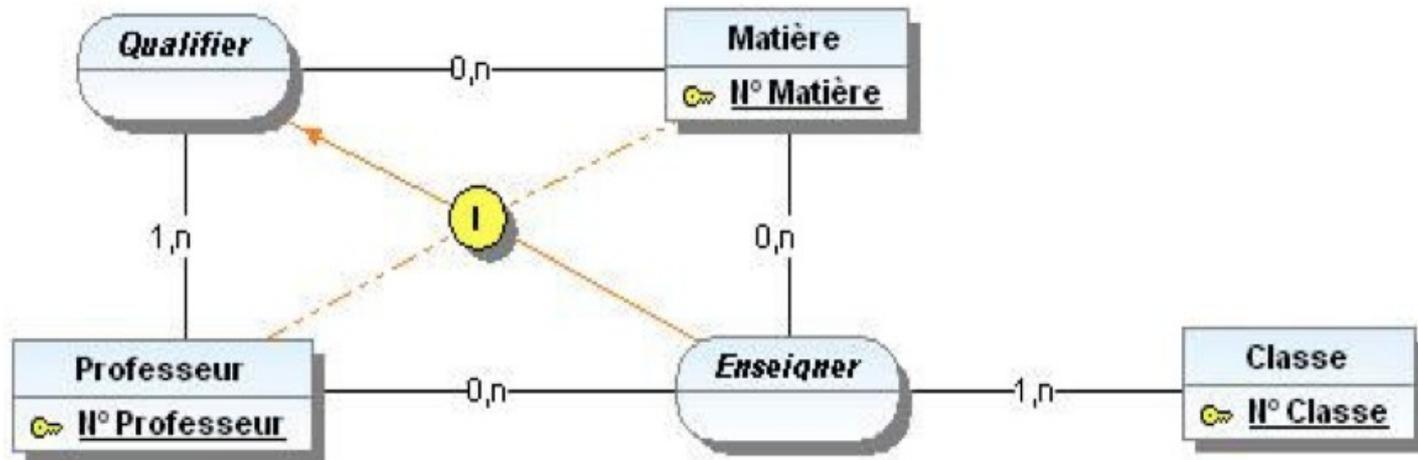
Toute occurrence de l'ensemble des individus participant à une relation (rembourser) participe à une autre relation (déposer). Les personnes remboursant un emprunt doivent avoir déposé une demande de prêt.



On met un *trait en pointillé* pour spécifier l'ensemble contraint. Ici c'est l'entité « *personne* »

L'inclusion (I)

Possibilité d'avoir des contraintes d'inclusion d'une association sur d'autres associations.



Tout occurrence (professeur,matière) de l'association Enseigner est incluse dans l'ensemble des occurrence (professeur, matière) de l'association Qualifier.

→ Un enseignant enseigne une matière que s'il est qualifié pour cette matière

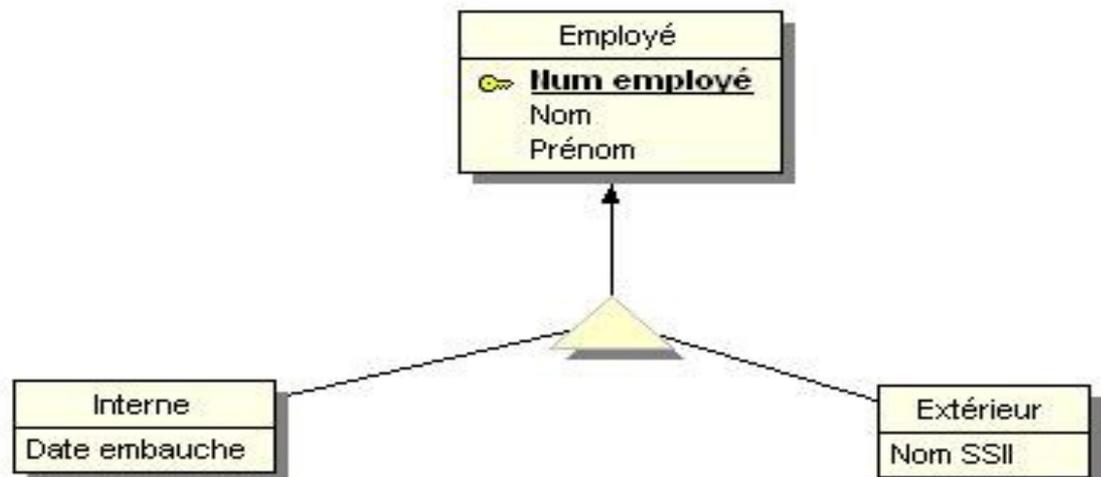
Spécialisation - Généralisation

Lors de la démarche de conception :

soit, on part d'une entité générale que l'on décline selon les cas particuliers : c'est le mécanisme de **spécialisation**.

soit, on part d'entités spécialisées et on factorise les propriétés communes dans une entité chapeau : c'est le mécanisme de **généralisation**. La notation est similaire à une spécialisation sauf que les sous types ont déjà chacun leur propre identifiant.

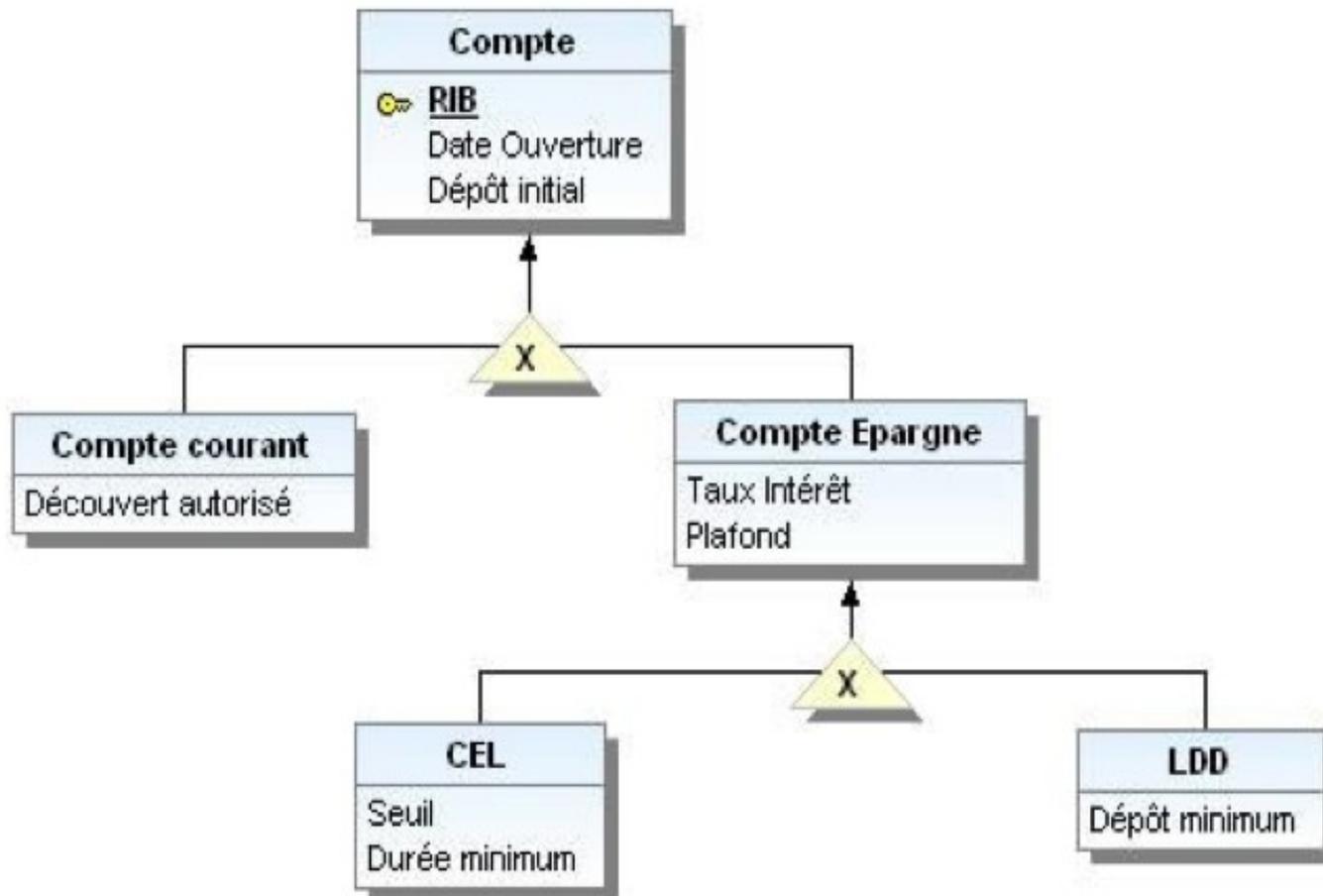
Exemple d'une spécialisation



Spécialisation exclusive (X)

(X) : Une occurrence sur-type est spécialisée en **au plus** une occurrence sous-type.

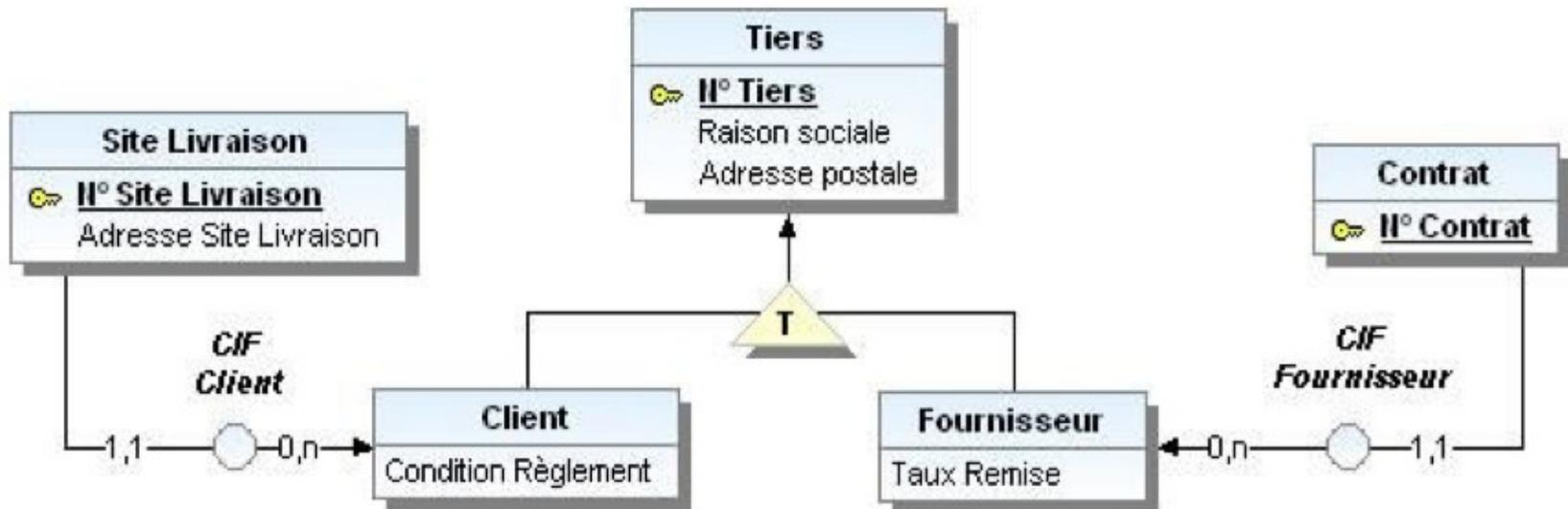
Le schéma ci dessous exprime le fait qu'un compte ne peut pas être à la fois « courant » et « d'épargne » mais on peut trouver un compte non classifié (ni courant, ni épargne).



Spécialisation totale (T)

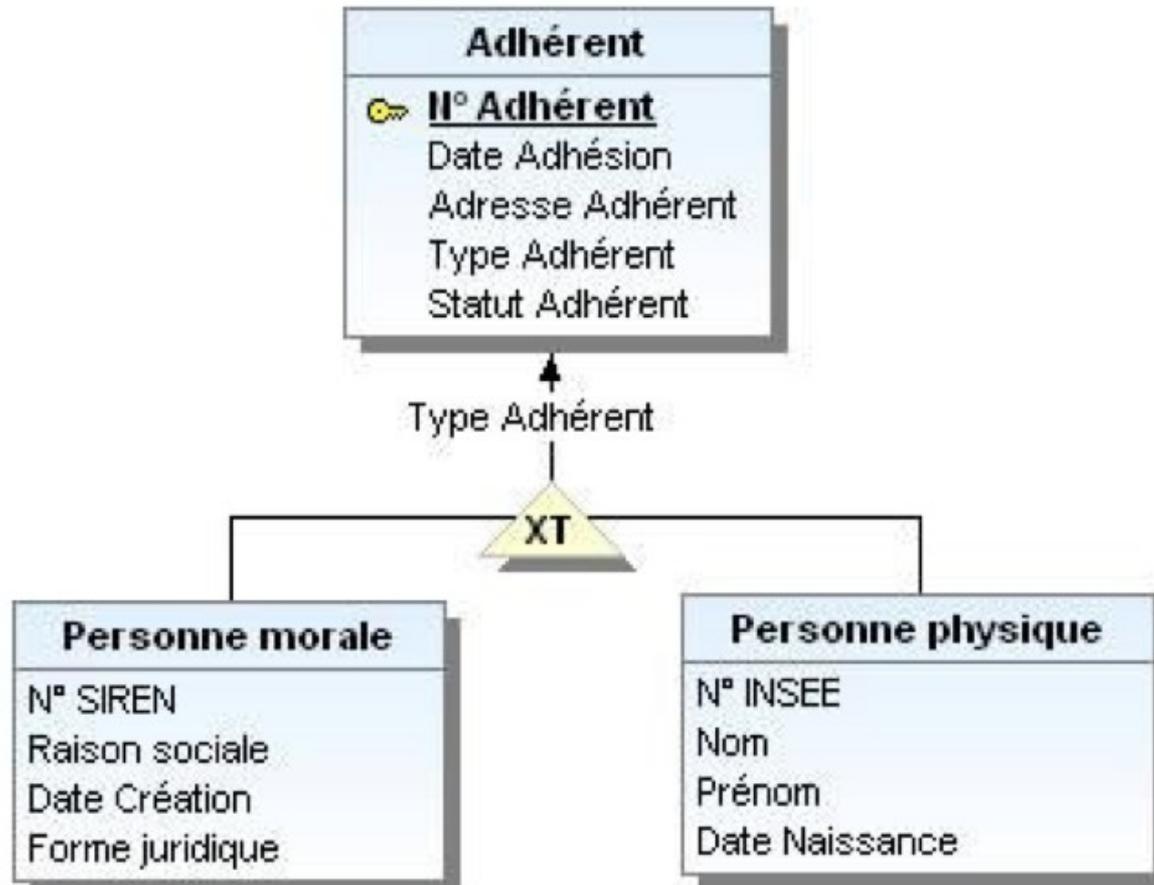
(T) : Une occurrence sur-type est spécialisée en au moins une occurrence sous-type.

Un tiers doit être : soit client, soit fournisseur, soit client et fournisseur.



Spécialisation exclusive totale (XT)

(XT) : Une occurrence sur-type est spécialisée en **une et une seule** occurrence sous-type.

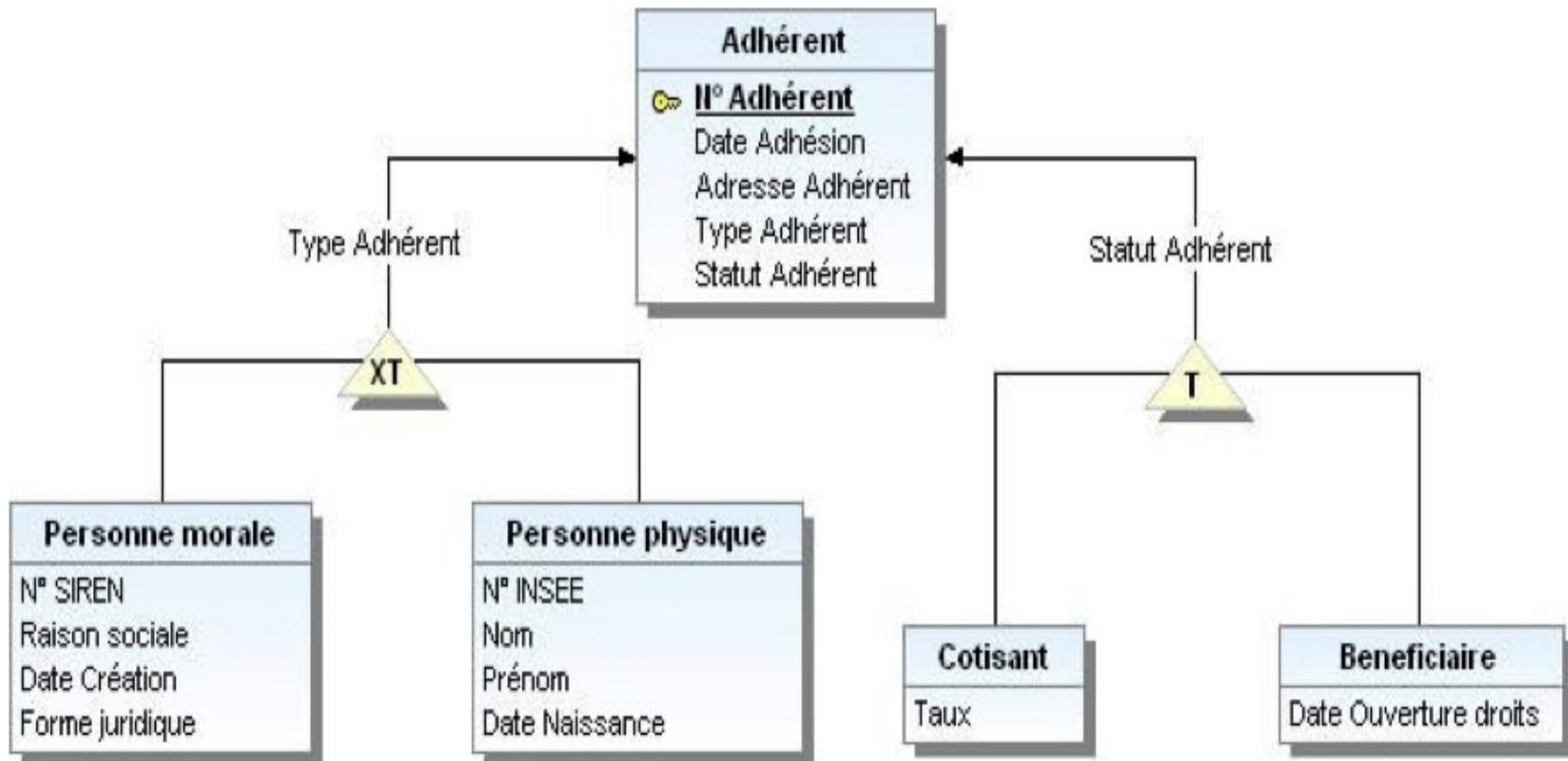


Spécialisation multiple

Le découpage d'une population en **sous populations** peut s'effectuer selon plusieurs critères, chaque critère produisant une **spécialisation** en différents sous-types.

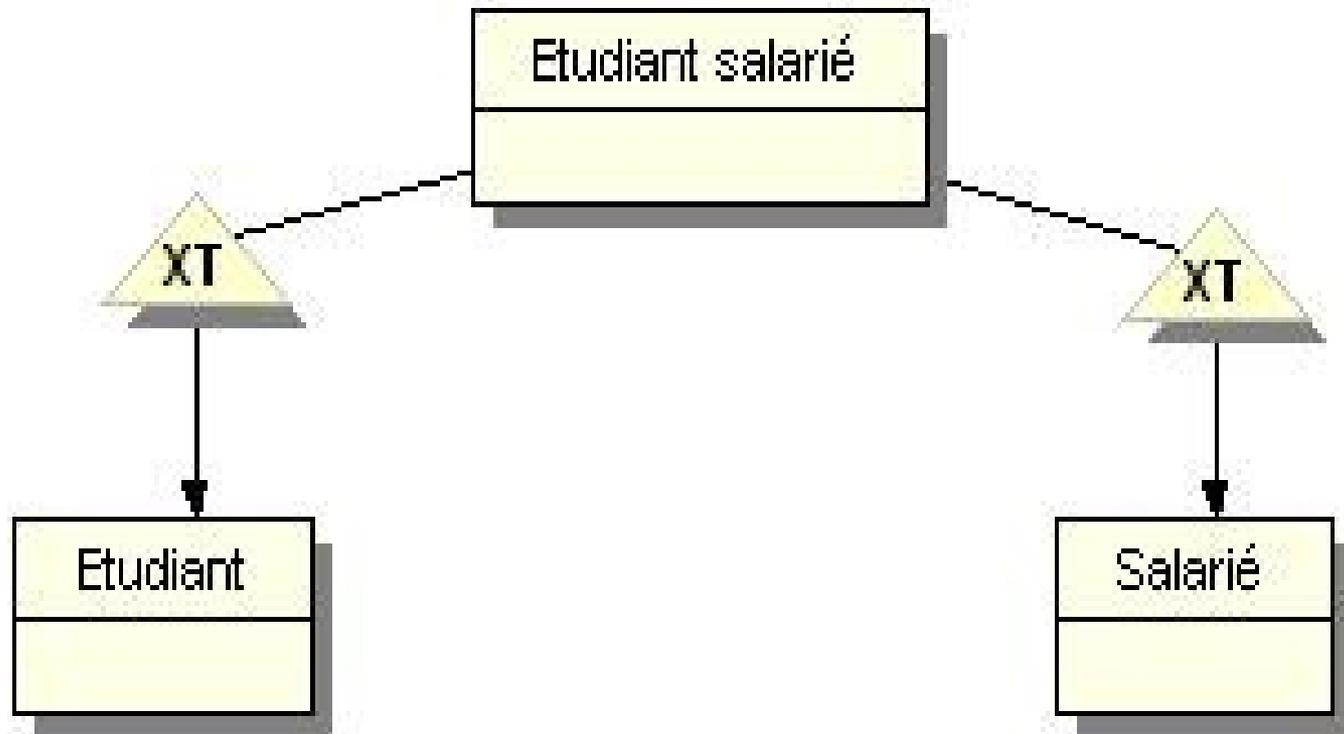
Adhérent est découpé en

- (1) personne morale+ personne physique (XT) : tout adhérent est soit morale, soit physique.
- (2) cotisant + bénéficiaire (T) : un adhérent est soit cotisant, soit bénéficiaire soit les deux.



Héritage multiple (à éviter) : définir une entité **sous-type** qui représente une sous population **commune** à des populations **différentes**. L'entité sous-type **hérite** de toutes les propriétés des différentes entités sur-types.

E



Stabilité

Une propriété est dite **stable** si, étant donné une occurrence de l'entité ou de l'association décrite par cette propriété, **la première valeur significative** attribuée à cette propriété ne peut être ultérieurement **modifiée**.

Elle se modélise par un **(S)**

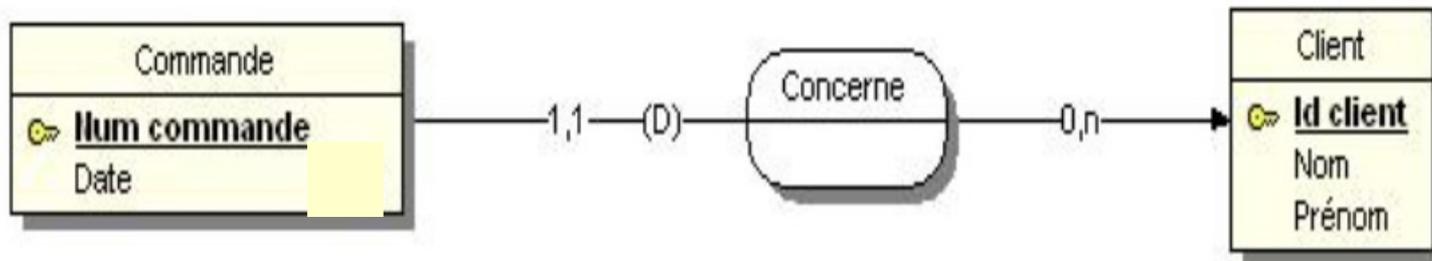


Par exemple : pour un contrat donné la date de signature ne peut changer dans le temps.

Patte définitive

Une fois l'**occurrence de l'entité** impliquée dans une occurrence d'une relation, elle ne peut plus se **détacher** de cette occurrence de relation pour se **rattacher** à une autre occurrence.

Sur les diagrammes, la patte définitive est notée **(D)**. Dans l'exemple, une commande n'est rattachée qu'à un et un seul client, et cela définitivement; une commande ne peut pas changer de client ! Elle est cependant **modifiable en mise à jour**



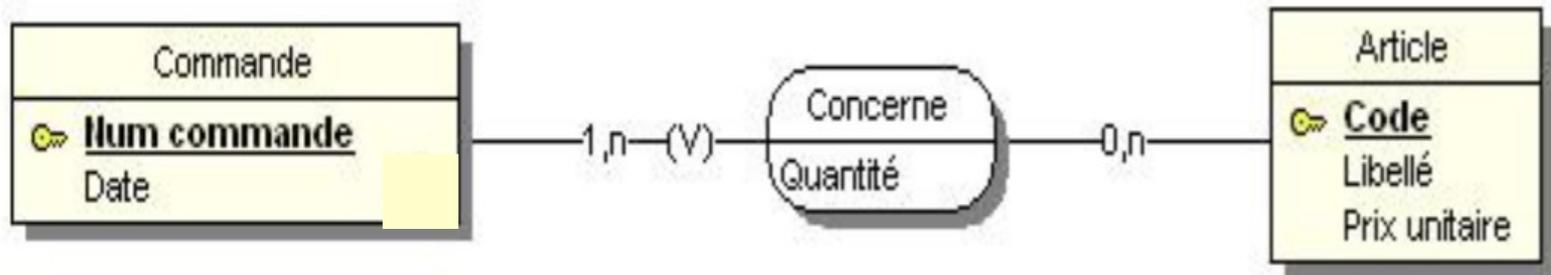
Patte verrouillée

Cette notion ne concerne que les pattes de relation à **cardinalité maxi n**.

Elle signifie que les n occurrences de la relations rattachées à l'occurrence d'une entité ont été créées **en même temps**.

On ne peut, dans le temps, **ni modifier**, **ni rajouter** (n+1) **ni supprimer** (n-1) d'occurrences de la relation à l'occurrence de l'entité; cette patte est **verrouillée**.

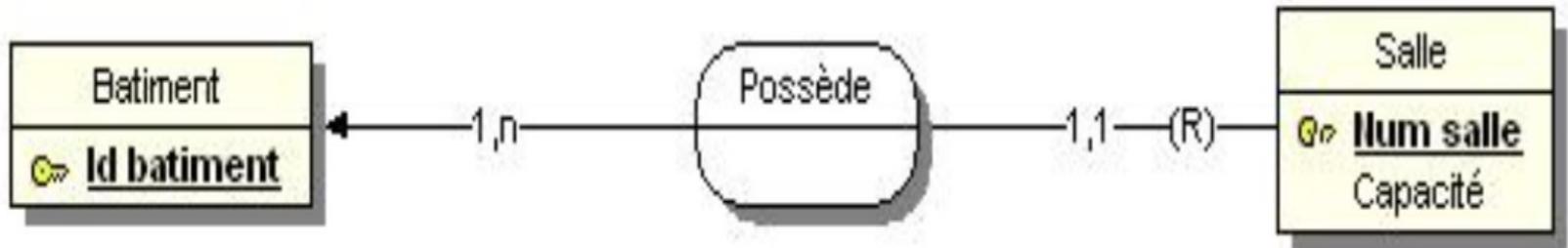
Elle est notée (V) et implique **S+D** (mais **pas équivalent**)



Dans cet exemple, on **ne peut plus modifier** une commande : ni ajouter, ni supprimer, ni modifier.

L'identifiant relatif

Il est constitué de propriétés de l'entité et/ou au moins de l'identifiant d'une autre entité reliée par une association 1-n ou 1-1.

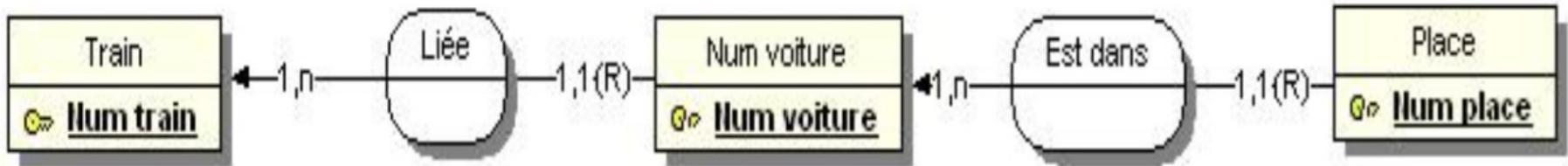


Bien qu'un numéro de salle (par exemple salle 1) peut se retrouver dans **plusieurs bâtiment** on note une cardinalité **1,1** suivie obligatoirement par **le symbole (R)** coté salle.

Pour connaître une salle dans des bâtiments on doit connaître le numéro de bâtiment pour localiser la salle. Parler juste de la salle 1 ou 2 n'a pas de sens.

L'individu Salle est représenté dans le MCD **sans l'attribut clé de bâtiment** mais un (R) est ajouté sur la patte pour préciser que Salle à une **clé relative**.

L'identification relative peut se propager à plusieurs niveaux à travers des associations.



On parle ainsi de la troisième place de la deuxième voiture du train 1024.

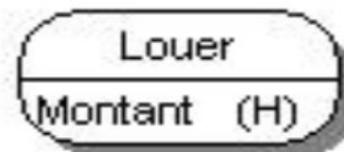
Historisation (H)

On désire parfois conserver, en cas de modification, les valeurs antérieures prises par des propriétés. Au lieu de créer une propriété de temps il est plus préférable d'indiquer explicitement le caractère **historisable** d'une propriété, d'une entité ou d'une association.

historiser **une entité**



historiser **une relation**



historiser **un attribut**



Gestion d'une conférence

Les chercheurs peuvent adresser des articles en indiquant les thèmes de rattachement des articles. Pour chaque article, on connaît le titre. Le texte de l'article n'est pas stocké dans la base. Chaque chercheur peut adresser plusieurs articles.

Pour chaque chercheur, on connaît son nom, son prénom, son adresse professionnelle, son email, le nom de son université de tutelle ou (exclusif) de son laboratoire de recherche. Un laboratoire peut être rattaché à une université. Les articles peuvent être co-signés. Dans ce cas, un des auteurs est désigné comme auteur principal.

On dispose d'une liste de relecteurs dont le rôle consiste à évaluer les articles et à juger s'ils sont de qualité suffisante pour être acceptés à la conférence. Les relecteurs sont des chercheurs. Ils peuvent eux-mêmes soumettre des articles. Pour chaque relecteur, on connaît la liste des thèmes relevant de sa compétence. On affecte un ou plusieurs articles à chaque relecteur. Plusieurs relecteurs sont désignés par article. Un relecteur doit rendre un avis détaillé sur chacun des articles qui lui sont affectés.

Un avis final (accepté / rejeté) correspondant à l'avis global des relecteurs est donné à chaque article soumis. Le texte final des articles acceptés tient compte des remarques détaillées des relecteurs.

