Université
Internationale
de Casablanca

Système d'information

Pr DAOUDI Imane
im.daoudi@yahoo.fr

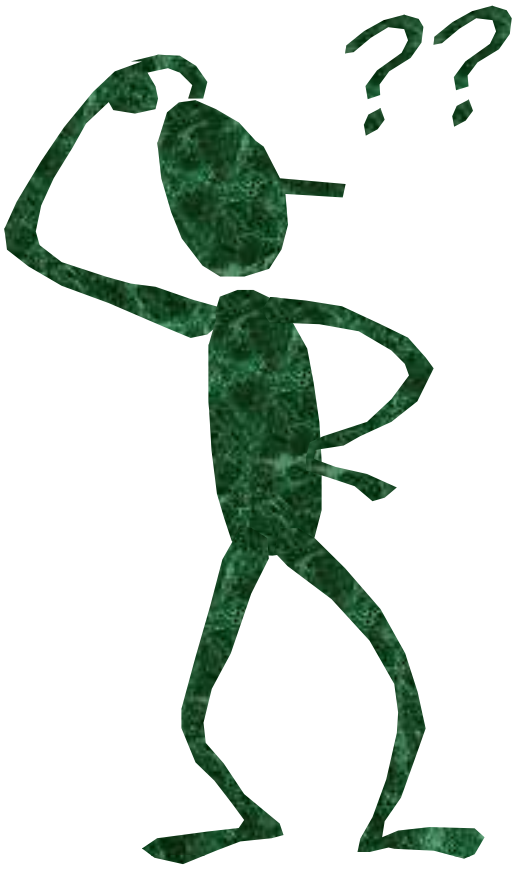
Plan du cours

I. Système de Gestion de Base de Données (SGBD)

- Introduction Générale
- Modélisation des données
  - Merise
  - UML
- Modèle Logique de Données (MLD)

Introduction générale

- Qu’est qu’une base de données??



Introduction générale

- Une base de données
  - Ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique
  - Ensemble de données interrogeable par le contenu
    - Exemple :

      les produits de prix >100 DH
  - Ensemble de données interrogeable à partir des relations entre données
    - Exemple:

      Les produits commandés par un client habitant Casablanca

Introduction générale

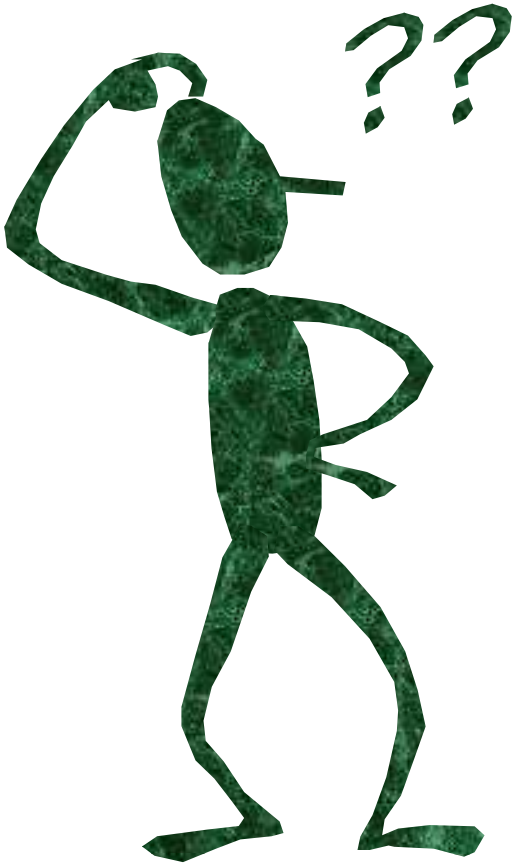
- Une base de données (autre définition)

un ensemble structuré de données (1) enregistrées sur des supports accessibles par l'ordinateur (2) pour satisfaire simultanément plusieurs utilisateurs (3) de manière sélective (4) en un temps opportun (5).

- (1) : Organisation et description de données
- (2) : Stockage sur disque
- (3) : Partage des données
- (4) : Confidentialité
- (5) : Performance

Introduction générale

- Qu’est qu’un SGBD?

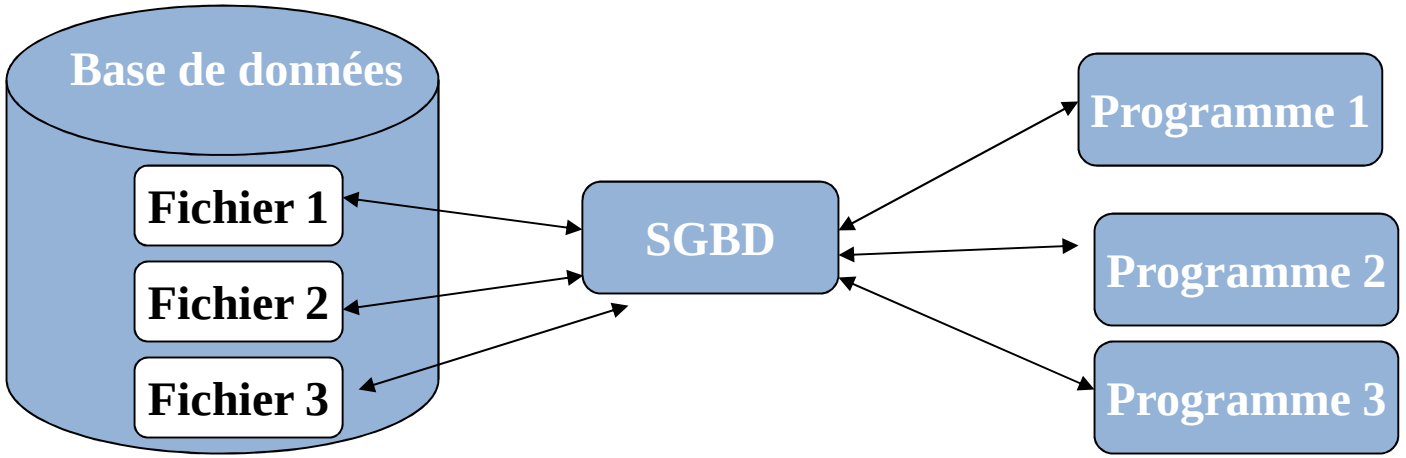


Introduction générale

- SGBD= Système de Gestion de Base de données

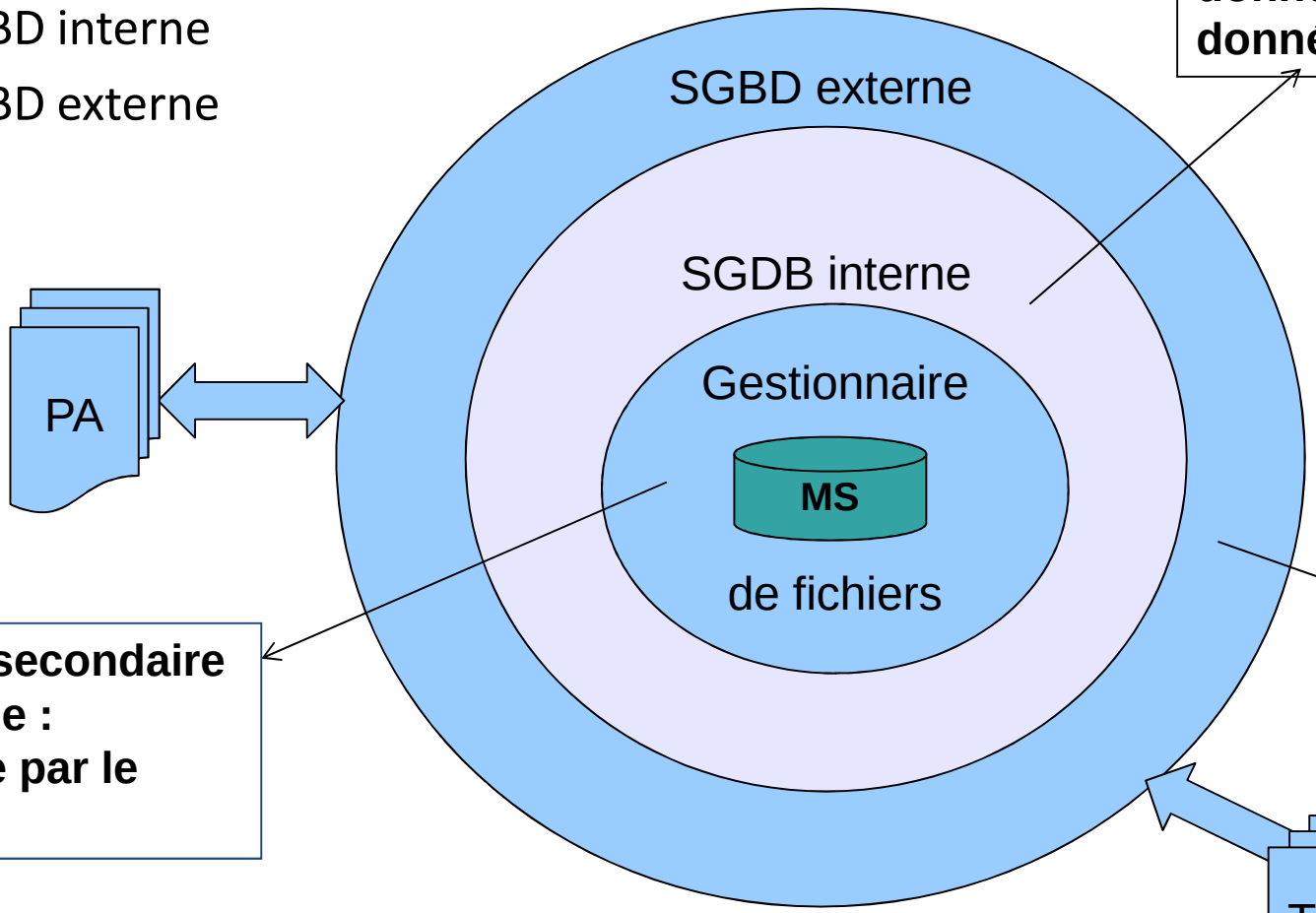
  DATA BASE MANAGEMENT SYSTEM (DBMS)

- Ensemble de logiciels systèmes permettant de gérer (d'insérer, modifier et de rechercher des données) une BD partagée par plusieurs utilisateurs simultanément



Introduction générale

- Un SGBD est composé de :
  - Gestionnaire de fichier
  - SGBD interne
  - SGBD externe



Gestion de données stockées dans les fichiers: gestion des liens entre données, placement des données dans des fichiers

Mise en forme, présentation des données aux programmes d'applications, et aux utilisateurs

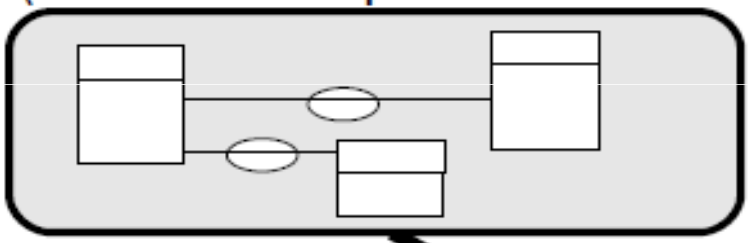
Mémoire secondaire adressable : recherche par le contenu

P.A=Programmes d'application
M.S=Mémoires Secondaires

Modélisation des données: Merise

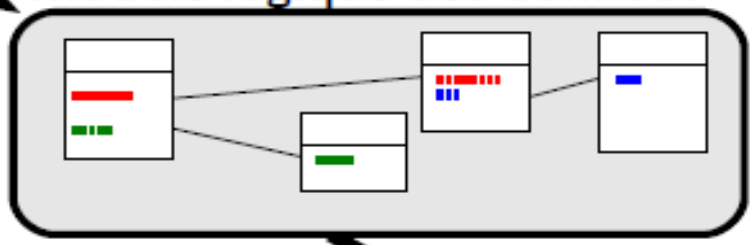
Cahier des charges

Modèle entité-association
(modèle conceptuel des données)

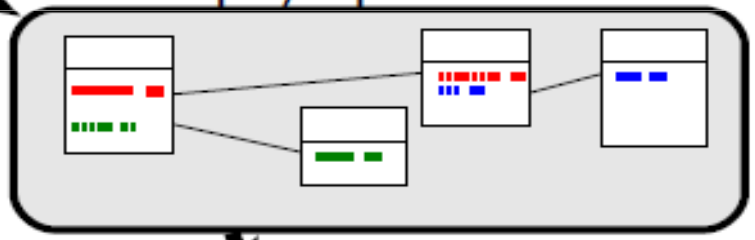


Méthode MERISE :
méthode d'analyse, de conception et de réalisation de systèmes d'informations.

Modèle logique des données



Modèle physique des données



Langage SQL

Méthode MERISE **pas seulement pour les bases de données** :

- Exprimer le besoin
- Créer les modèles conceptuels
- Créer les modèles logiques
- Créer les modèles physiques

Modélisation des données: Merise

Objectif du MCD

Ecrire de façon formelle les données d'une base de données. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire la base de données à l'aide d'entités.

- Il est à la base de tous les SGBD dits relationnels (Access, Oracle, DB2…) qui sont les plus utilisés actuellement dans les entreprises.

- Il est généralement représenté à l'aide du formalisme «entités-associations » sous la forme de :

ENTITES, ASSOCIATIONS et ATTRIBUTS.

Modélisation des données: Merise

Entité

Concept concret ou abstrait (un fait, un moment…) identifié du monde réel caractérisé par un nom et une liste de propriétés.

- Une entité concrète possède une existence physique : client, équipement, et produit
- Une entité abstraite a une existence conceptuelle : une transaction, un tarif, l'annulation d'un vol d'avion

❑ Exemples

- Le client Jean Dupond est une entité concrète
- La commande COM0001 est une entité abstraite
- L'entité Personne(nom, prénom), et l'entité Voiture(nom , puissance fiscale) ne peuvent pas être groupés en une même entité car ils ne partagent pas leurs propriétés

Client

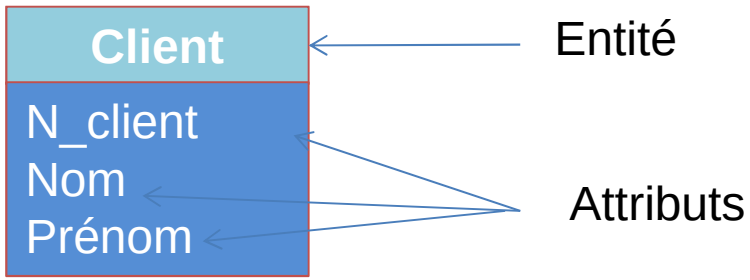
L' entité se représente par un cadre contenant le nom de l'entité

Modélisation des données: Merise

Attribut (Entité)

Propriété d'une entité ou d'une association caractérisée par un nom et un type élémentaire.

- Est un élément d'une entité :
  - a un nom unique,
  - permet de mémoriser une valeur,
  - doit avoir un sens (donc une valeur) pour chacune des occurrences de l'entité.
- Exemple



Représentation graphique d'une entité comportant trois attributs

Modélisation des données: Merise

Règles concernant les attributs

Règle 1

Un attribut ne peut en aucun cas être partagé par plusieurs entités/associations.

Règle 2

Un attribut est une donnée élémentaire, ce qui exclut des données calculées ou dérivées.

Règle 3

Une entité et ses attributs doivent être cohérents entre eux (i.e. ne traitent qu'un seul sujet).

Modélisation des données: Merise

Occurrence: entité

Elément particulier d'une entité, identifiable de façon unique (instance)

- Deux occurrences de l'entité ne peuvent avoir la même valeur d'identifiant.
- Exemple

L'entité client1 dont le N° est 06464M est une occurrence de l'entité client

Client 1	Client 2
064646M Dupont Frank 23 BD zola	012646M Revaud jerome 2 BD alpha

Modélisation des données: Merise

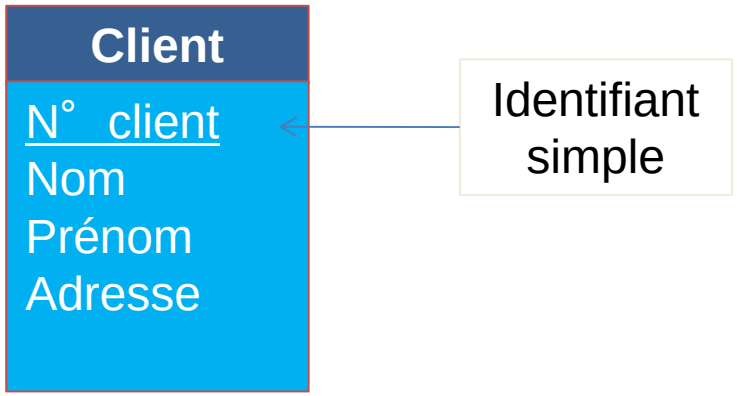
Identifiant: entité

Attribut ou groupe d'attributs permettant d'identifier chaque occurrence d'une entité.

Règle 4

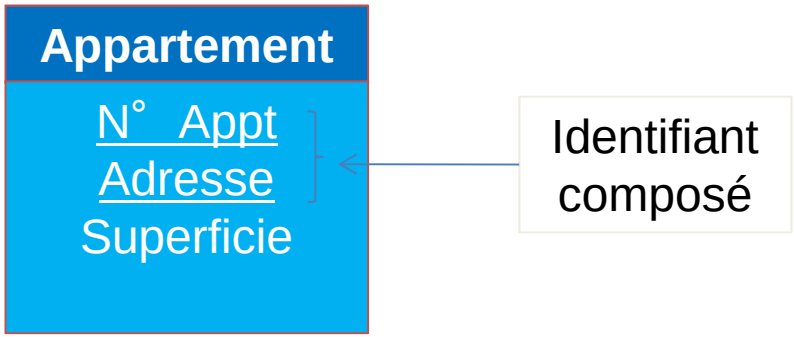
Chaque entité possède au moins un identifiant, éventuellement formé de plusieurs attributs.

- Exemple



Modélisation des données: Merise

- identifiant : entité (suite)
- Il existe 2 types d'identifiants: simple et composé
  - Un identifiant est simple s'il est formé d'un seul attribut
  - Un identifiant est composé s'il est formé de plusieurs attributs,
- Exemple:
  - Entité avec identifiant composé



Modélisation des données: Merise

Association

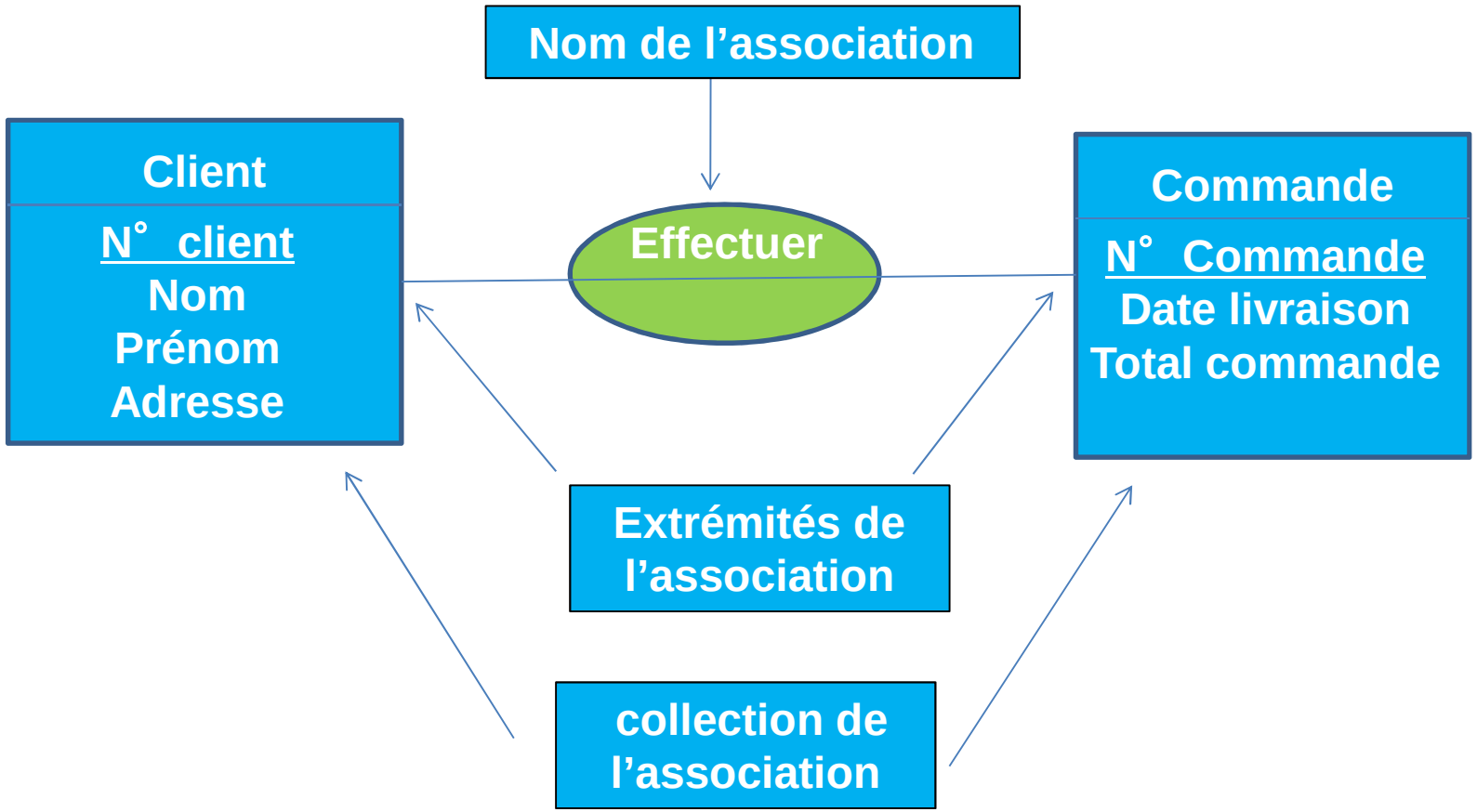
Lien logique entre entités dont le type est défini par un verbe et une liste éventuelle de propriétés

- On appelle **collection de l'association** l'ensemble des entités qu'elle relie.

Règle 5

Un attribut peut être placé dans une association uniquement lorsqu'il dépend de toutes les entités liées par l'association.

- Association : exemple



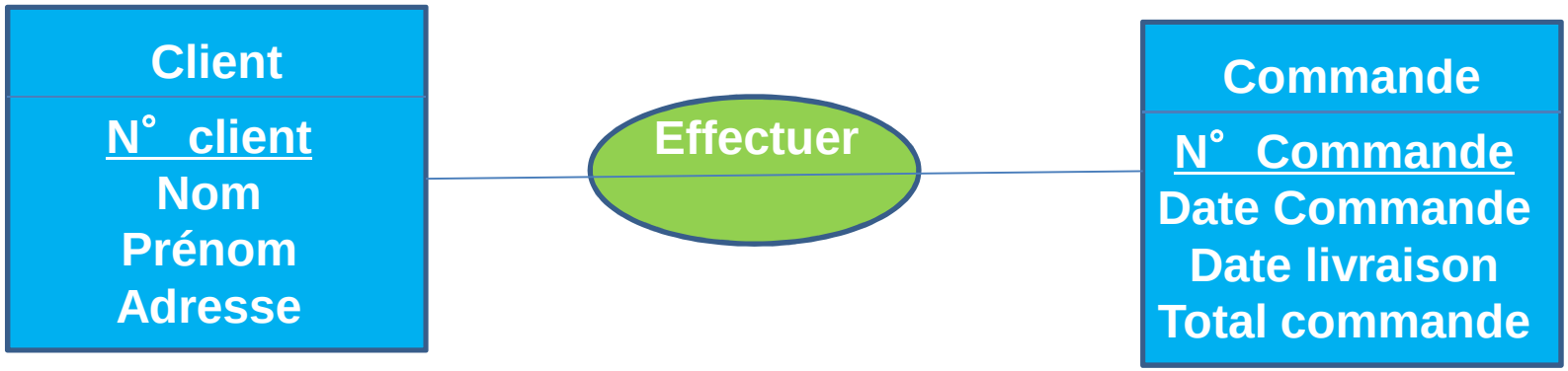
Modélisation des données: Merise

- Association : identifiant
- Il est implicite !
  - C'est un ensemble composé des identifiants de la collection de l'association.

Règle 6

La concaténation des identifiants des entités liés à une association constitue l'identifiant de cette association (cet identifiant n'est pas mentionné sur le modèle (il est implicite).

- Exemple:
  - l'identifiant de l'association « effectuer » est le couple (N° client, N° commande)

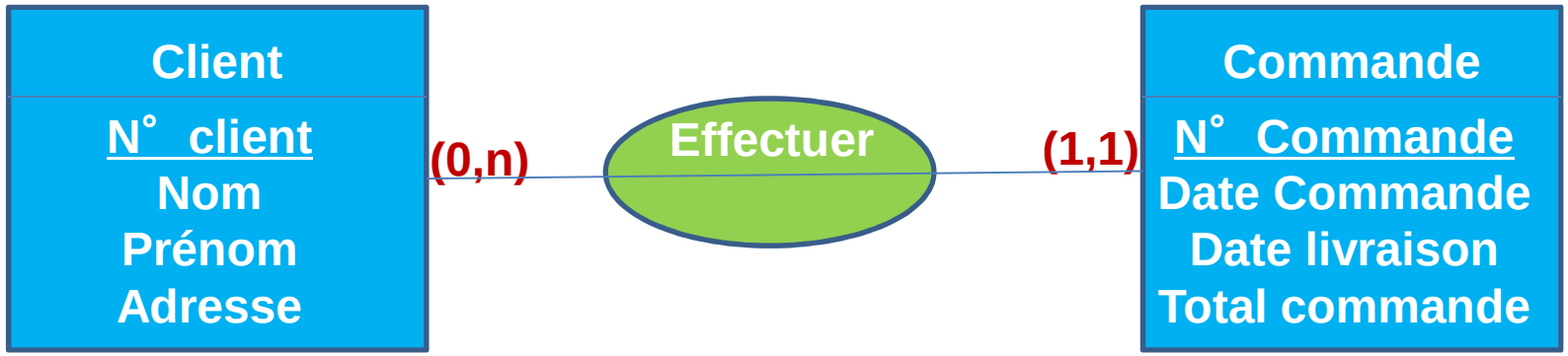


Modélisation des données: Merise

- Association : cardinalités (1)

Contrainte inscrite à chaque extrémité d'une association comportant un couple de valeurs (min-max) qui établit, pour chaque entité de l'association, le nombre minimum et maximum d'occurrences d'une association auxquelles elle peut participer

- Exemple
  - Un client peut effectuer de 0 à n commande, mais une commande ne peut être effectuer que par un seul client



Modélisation des données: Merise

- Association : cardinalités (2)

Règle 7&8

Règle 7: L'expression de la cardinalité est obligatoire pour chaque patte d'une association

Règle 8: Une cardinalité minimal est toujours 0 ou 1, et une cardinalité maximale est toujours 1 ou n

- Remarques
  - Une cardinalité maximale de 0 n'a pas de sens
  - Si une cardinalité maximale est connu et vaut 2, 3 ou plus, alors nous considérons qu'elle est indéterminée et vaut n
  - Les cardinalités minimales qui valent plus de 1 sont modélisées par 1

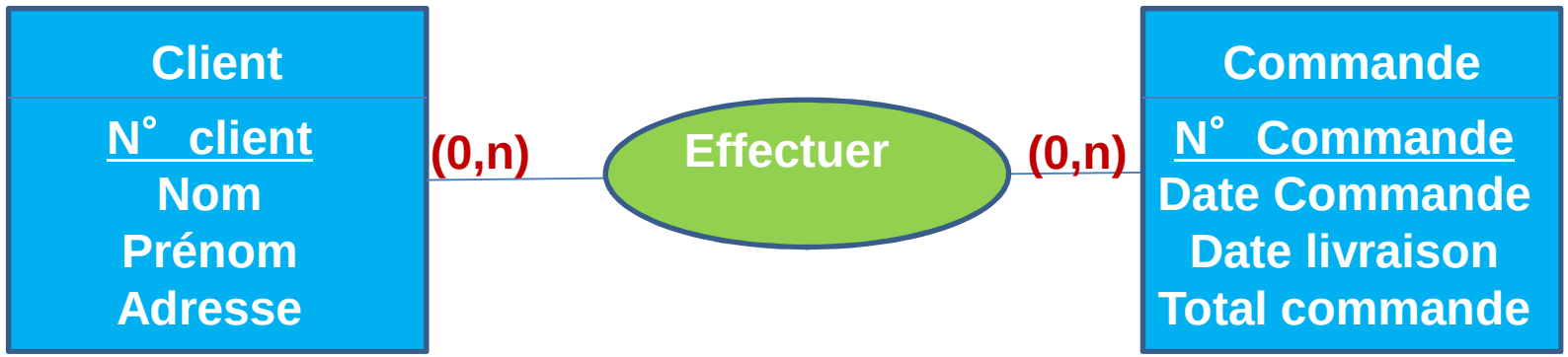
- Les seules cardinalités admises sont:

cardinalités	signification
0,1	Au plus un
1,1 (ou 1)	Un seul
0,n (ou *)	Un nombre indéterminé
1,n	Au moins un

Modélisation des données: Merise

- Association : cardinalités (2)

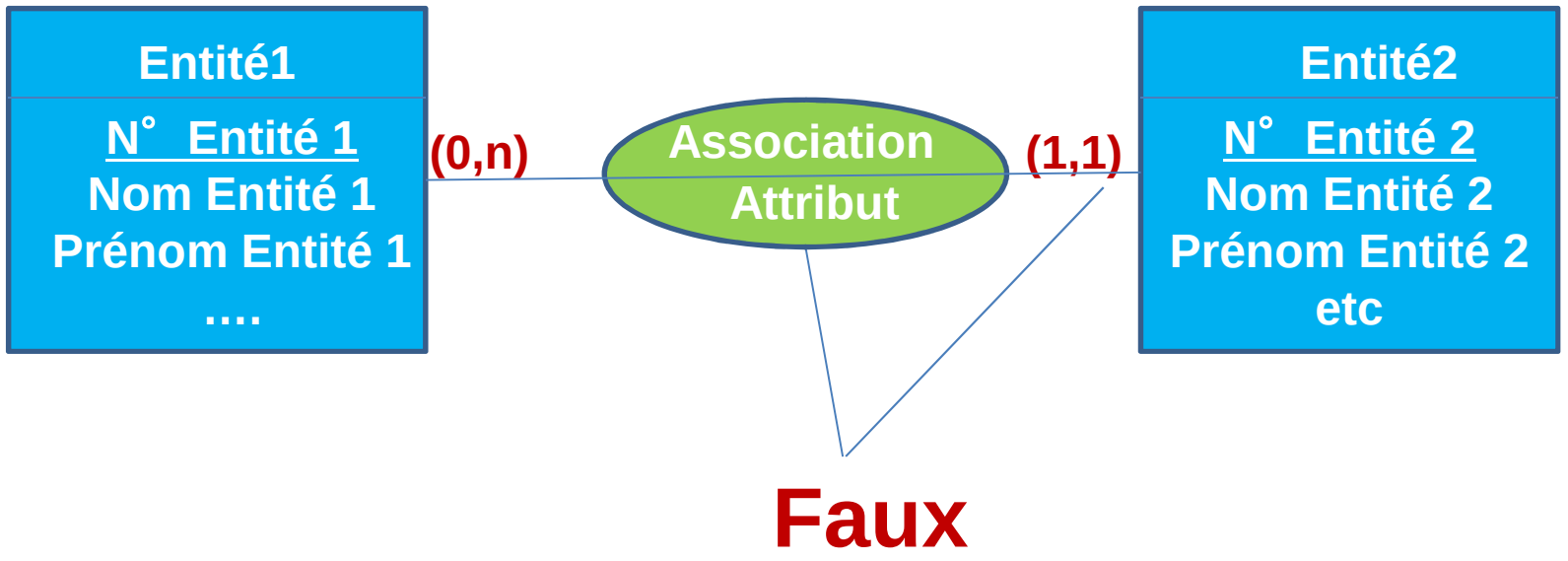
Une extrémité sans contrainte aura pour cardinalité (0,n)



Modélisation des données: Merise

- Règles absolues!! (1)

Une association binaire de cardinalité minimale et maximale égale à un ne peut en aucun cas porter de propriétés !

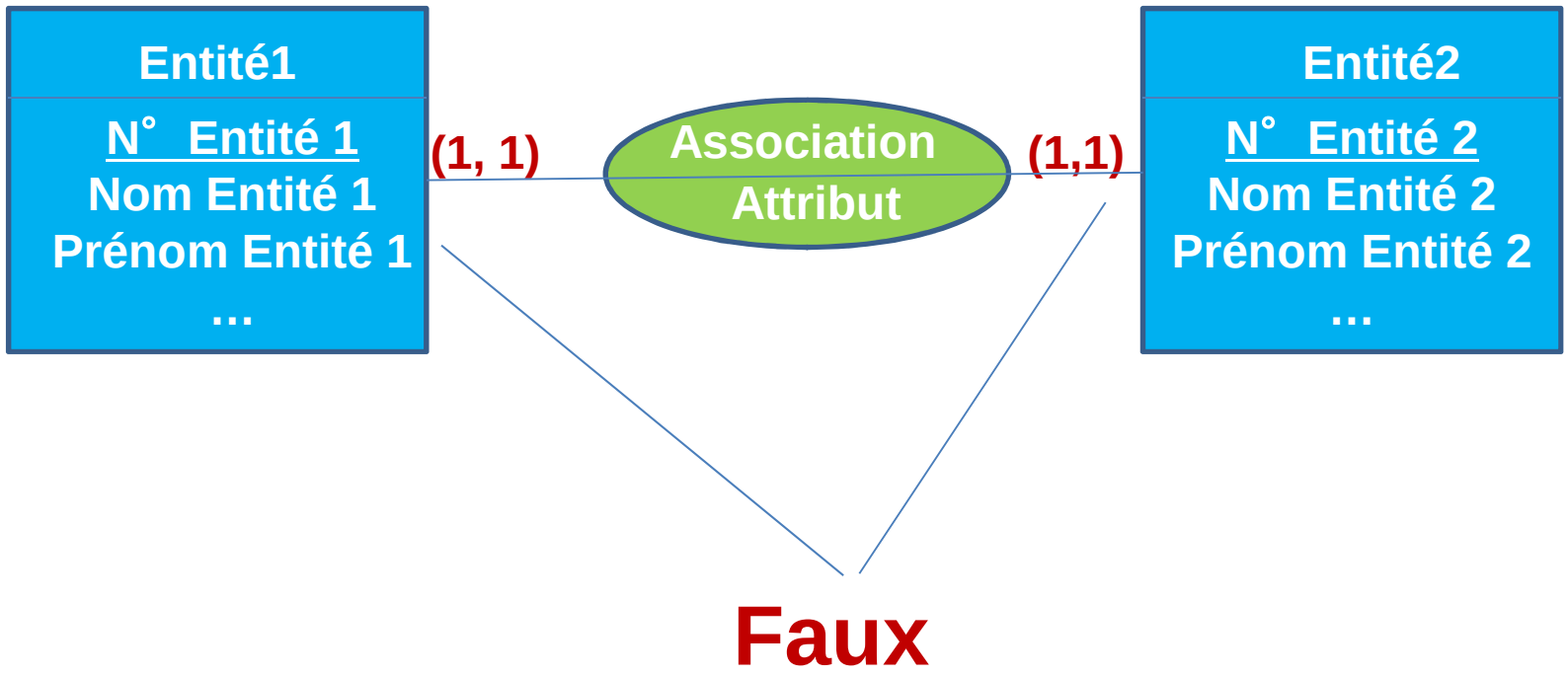


Faux

Modélisation des données: Merise

- Règles absolues!! (2)

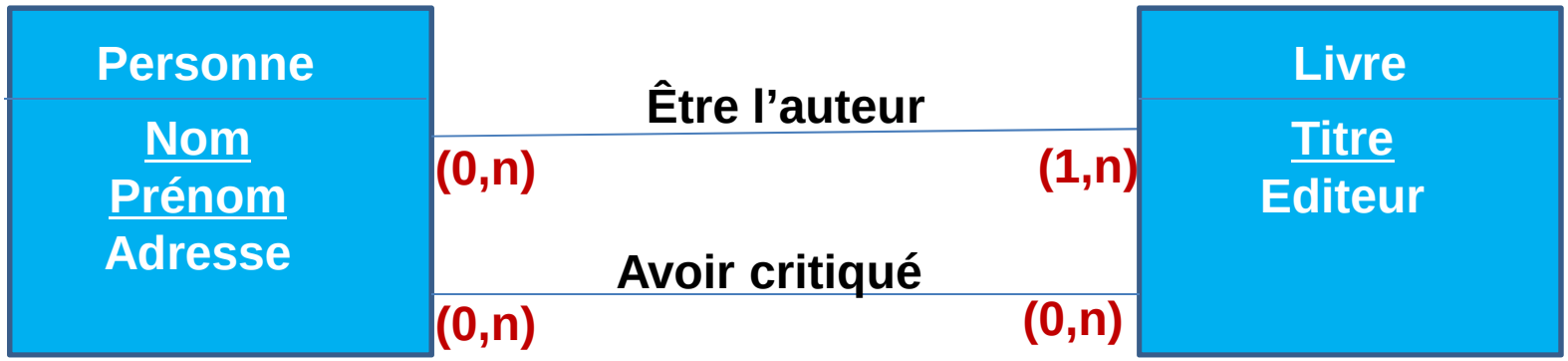
Une association binaire ne peut en aucun cas porter des cardinalités 1,1 des deux extrémités !



Les différents types d'associations

- Les associations plurielles

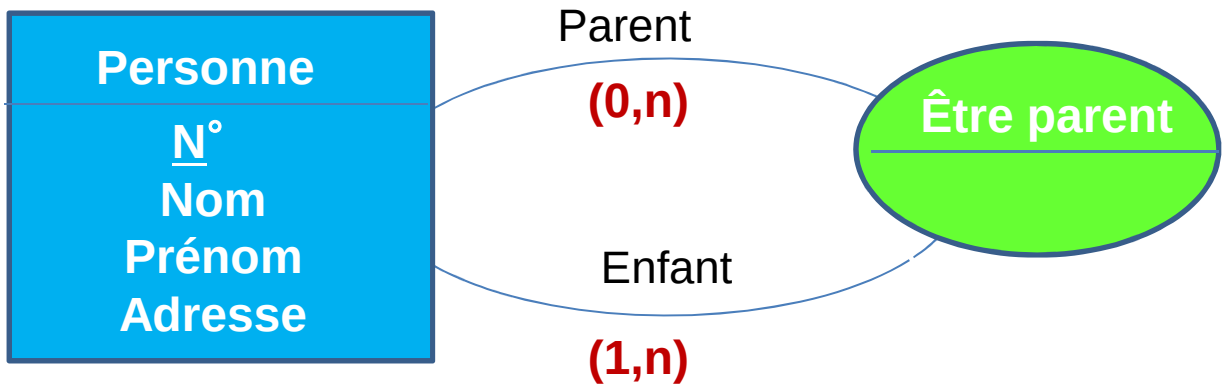
Deux mêmes entités peuvent être plusieurs fois en associations



Les différents types d'associations

- Les associations réflexives

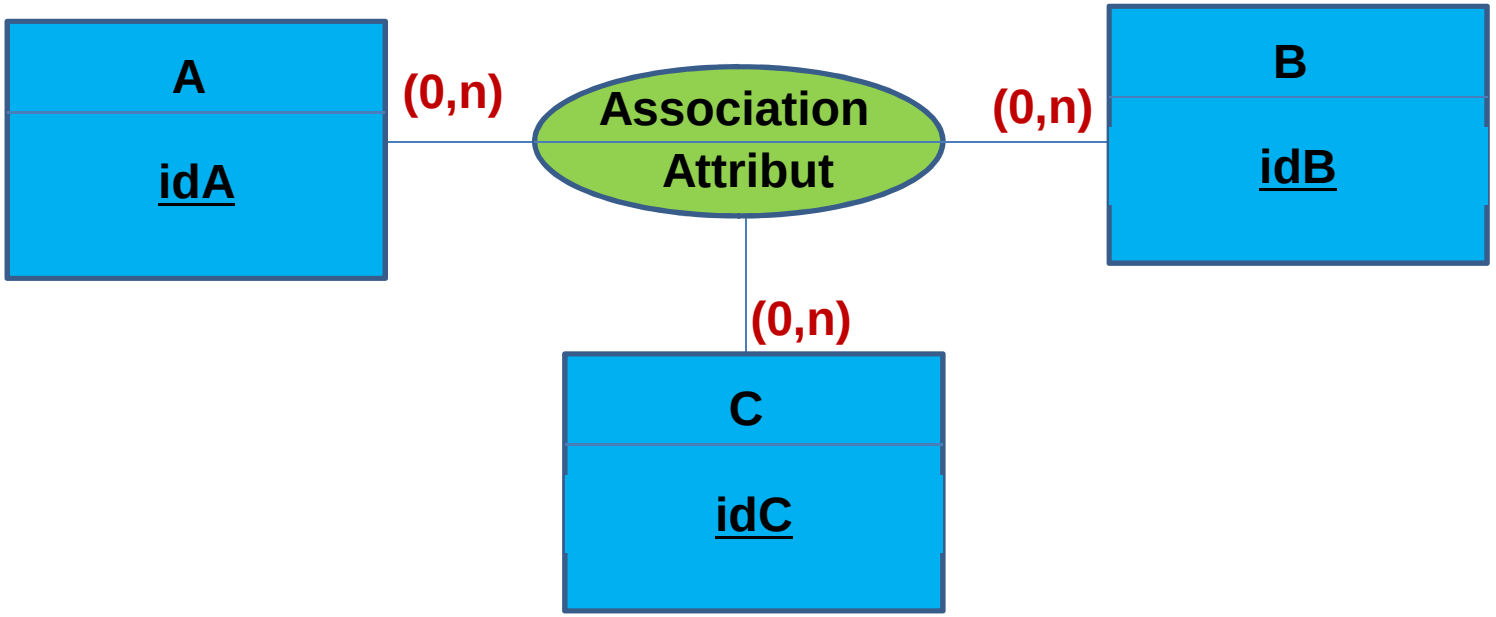
Une association réflexive est une association reliant des occurrences de la même entité



Les différents types d'associations

- Les associations ternaires

Une association ternaire est une association qui décrit un lien sémantique entre trois entités

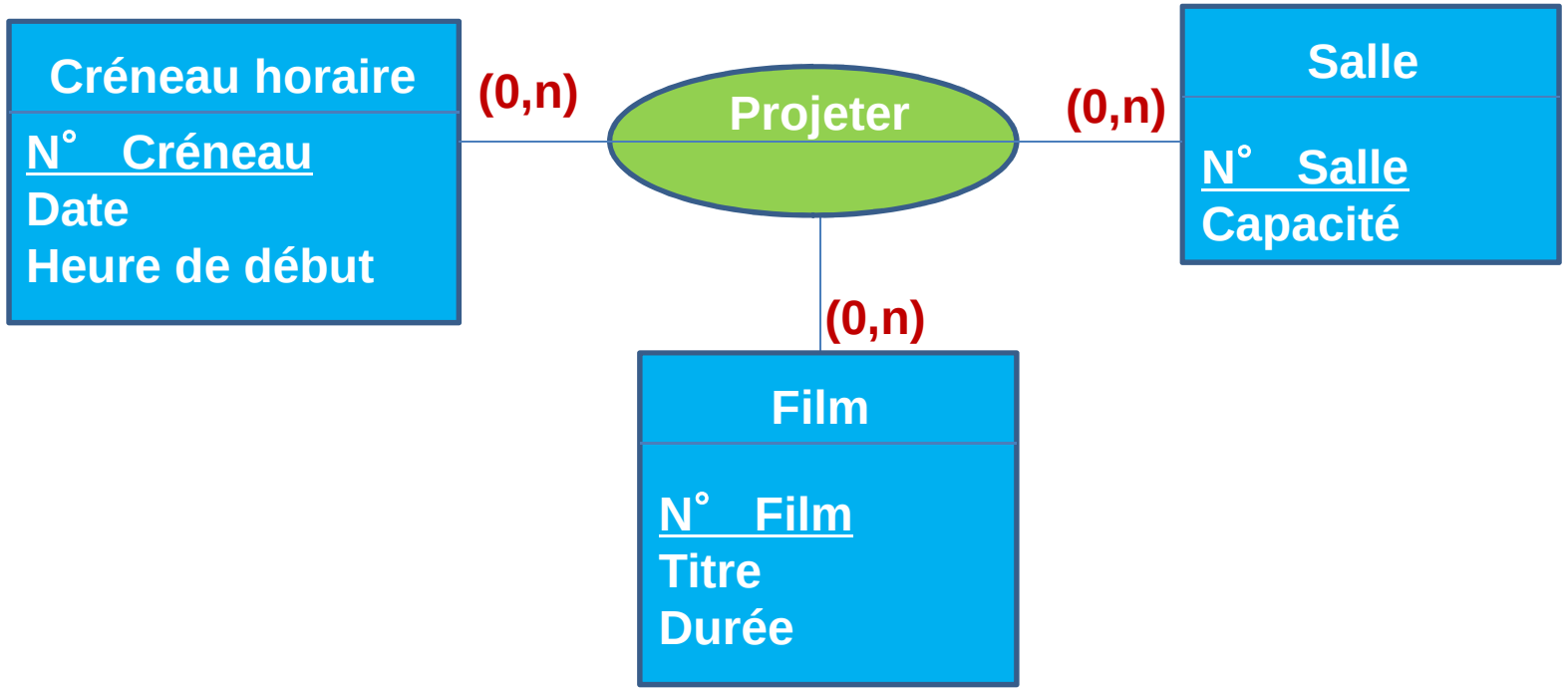


Difficile à gérer en pratique !!!

Il faut essayer d'en avoir le moins possible

Les différents types d'associations

- Les associations ternaires

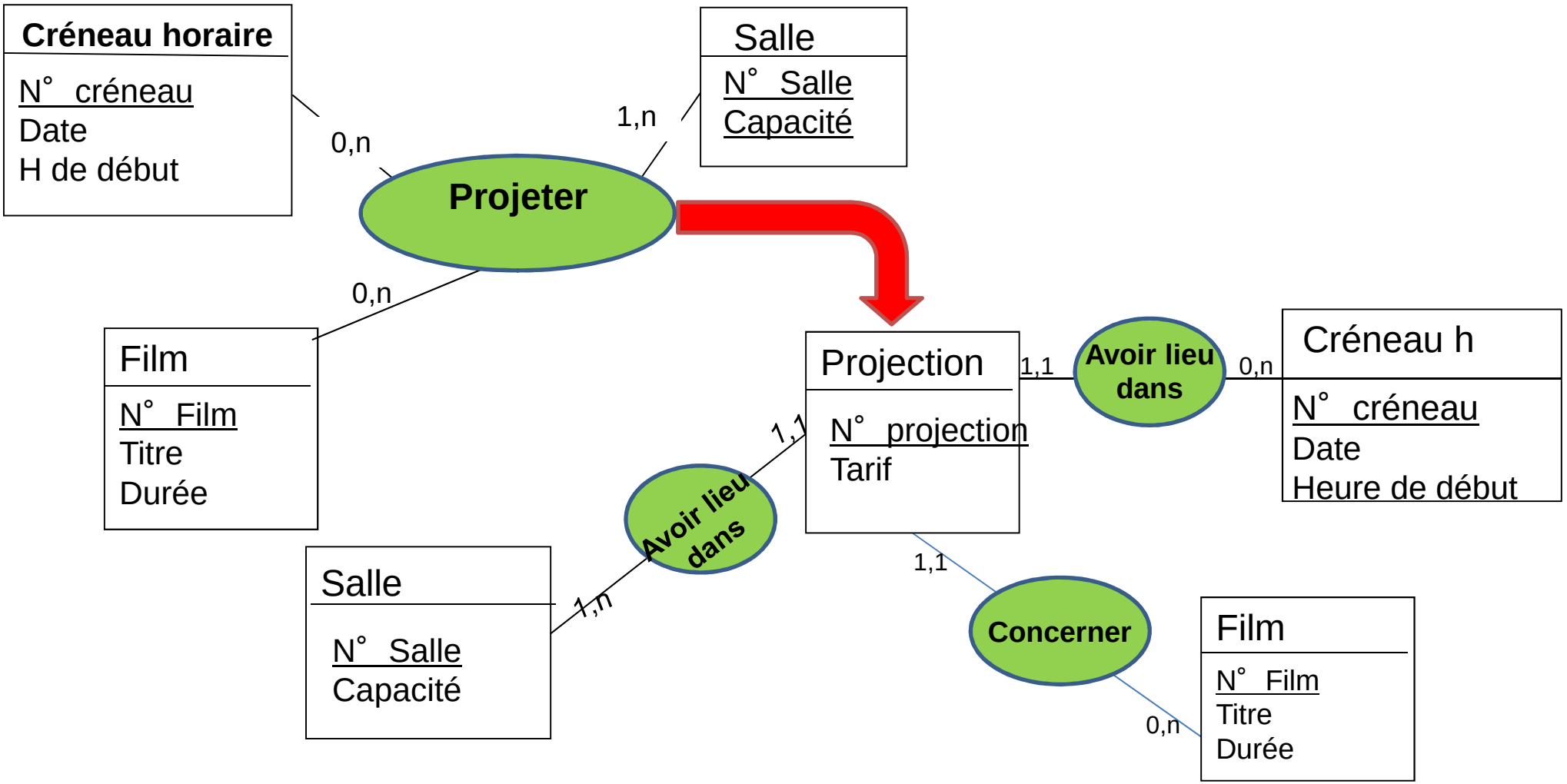


Les différents types d'associations

- Les associations ternaires: décomposition
- On remplace l'association ternaire (ou n-aire) par une entité et on lui attribut un identifiant
- On crée des associations binaires entre la nouvelle entité et toutes les autres entités de la collection de l'ancienne association
- La cardinalité de chacune des associations binaires crées est 1,1 du côté des entités créé et 0,n ou 1,n du côté des entités de la collection de l'ancienne association

Les différents types d'associations

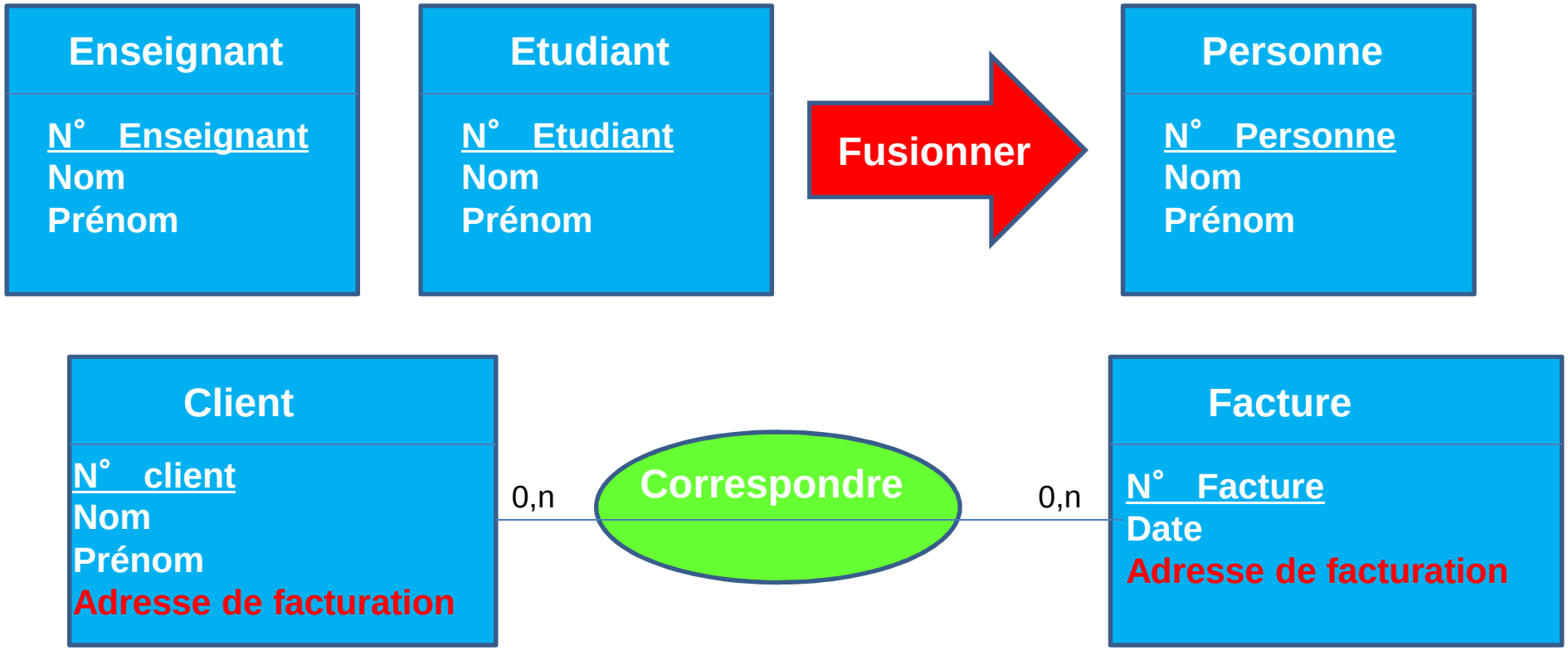
- Les associations ternaires



Règles de bonne formation d'un MCD

- Règles portant sur les noms

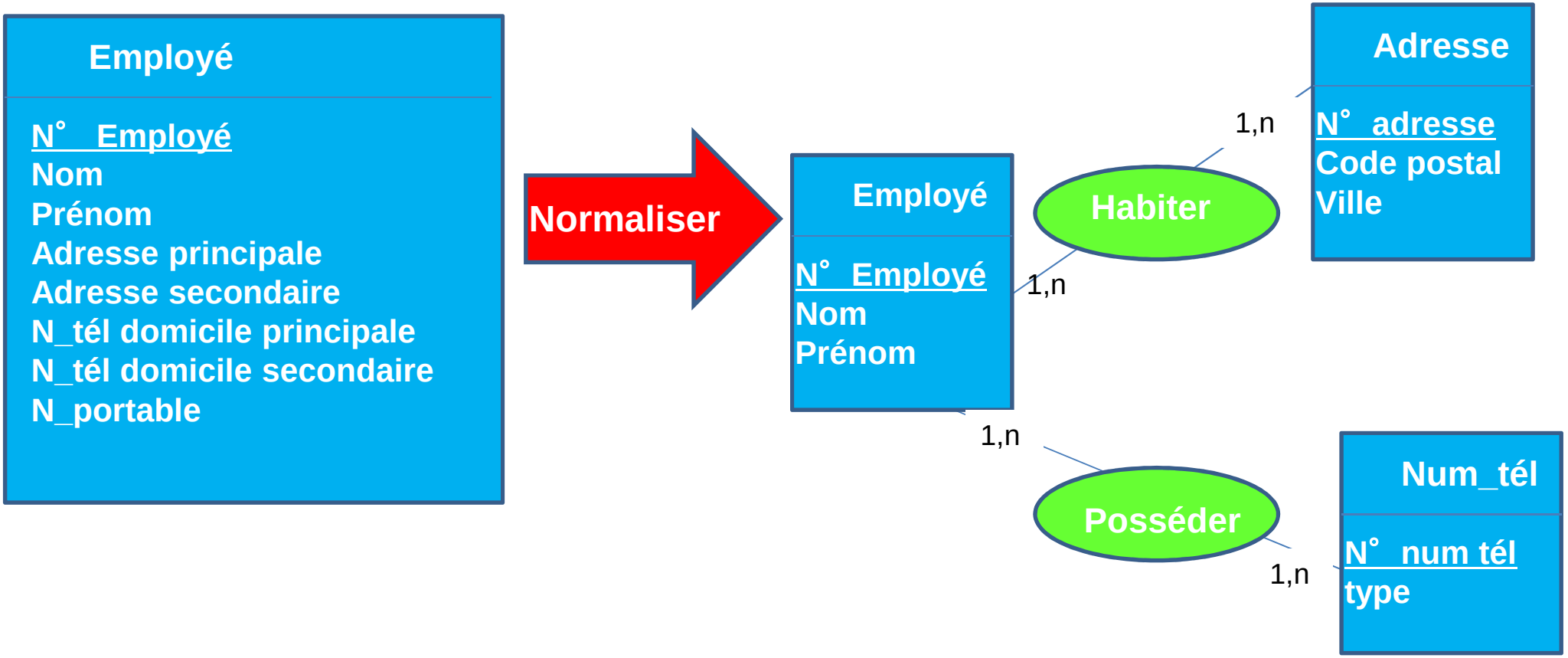
Dans un modèle entités-associations, le nom d'une entité, d'une association, ou d'un attribut doit être unique



Règles de bonne formation d'un MCD

- Règles de normalisation des attributs

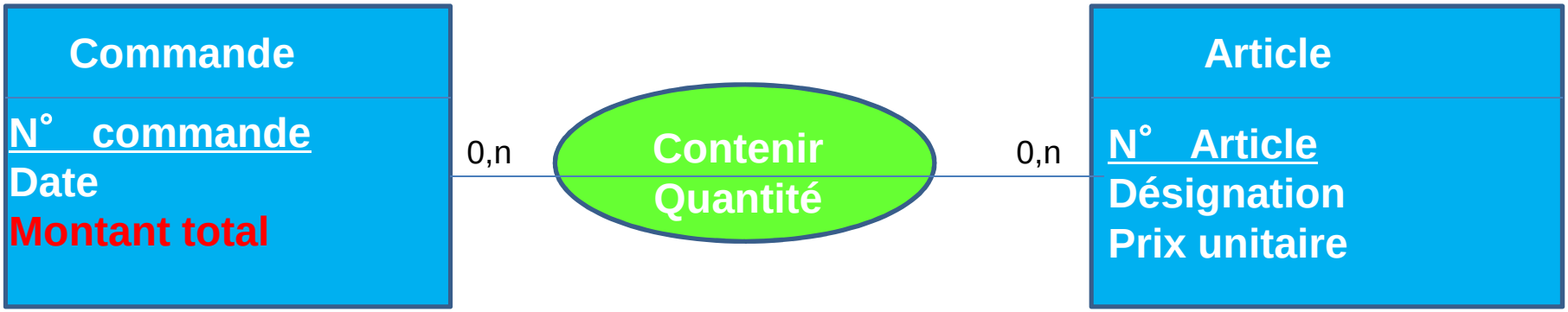
un attribut multiple doit être remplacé par une association et une entité supplémentaires



Règles de bonne formation d'un MCD

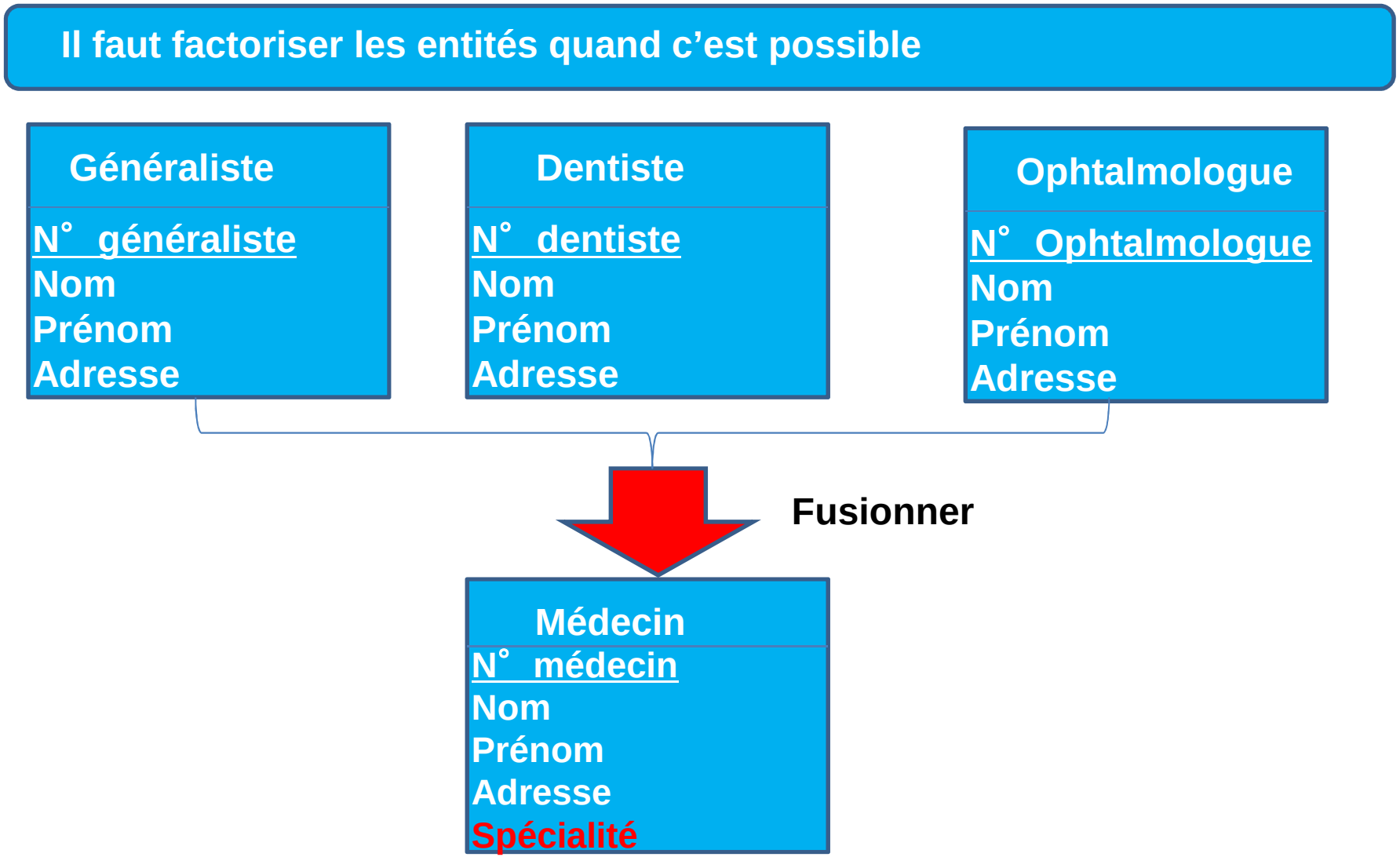
- Règles de normalisation des attributs

Un attribut est une donnée élémentaire, ce qui exclut des données calculées ou dérivées



Règles de bonne formation d'un MCD

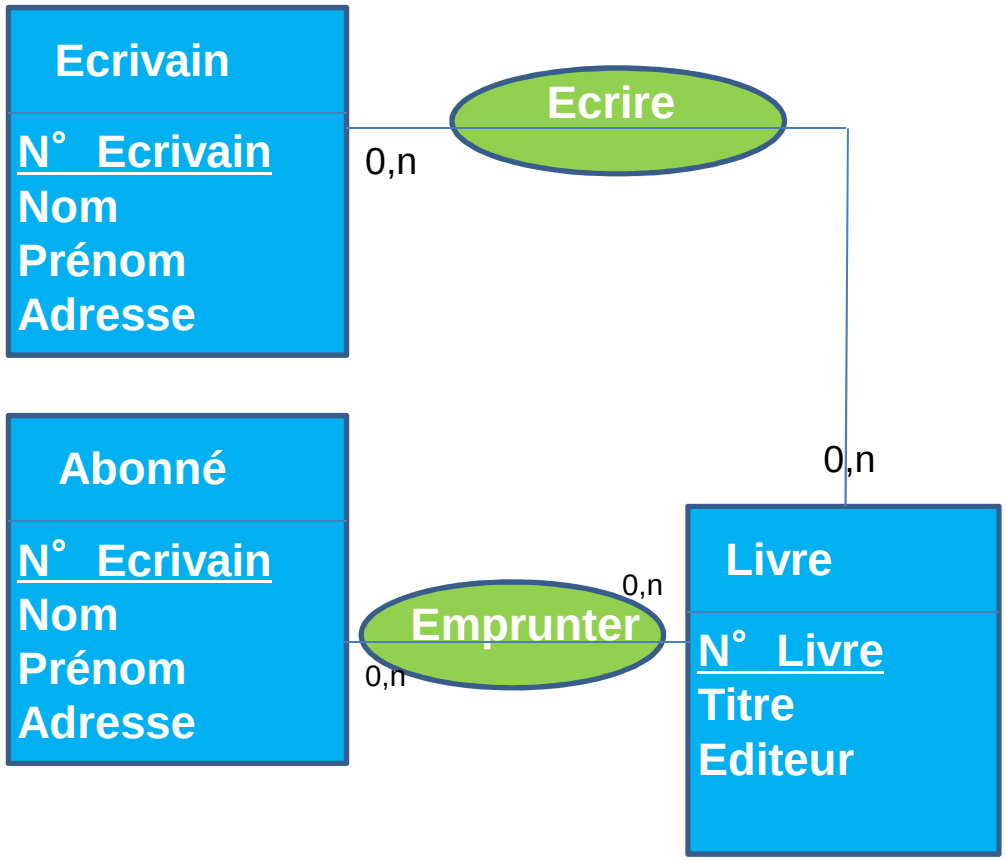
- Règles de fusion/ suppression entités/associations

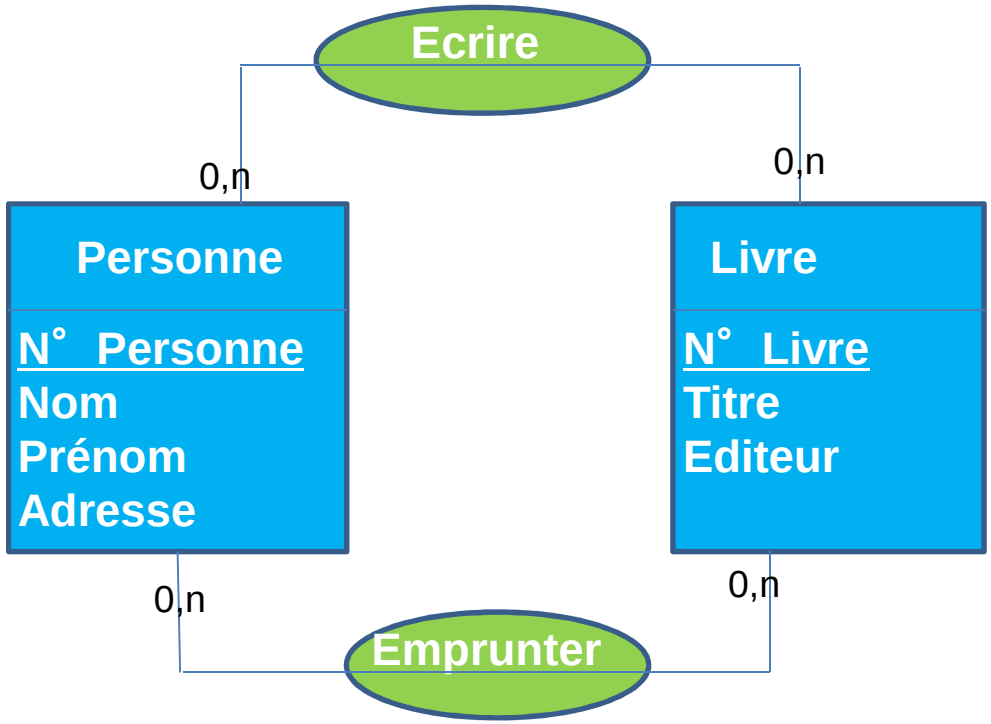


Règles de bonne formation d'un MCD

- Règles de fusion/ suppression entités/associations

Il faut factoriser les entités quand c'est possible, mais l'introduction d'un attribut supplémentaire n'est pas toujours nécessaire

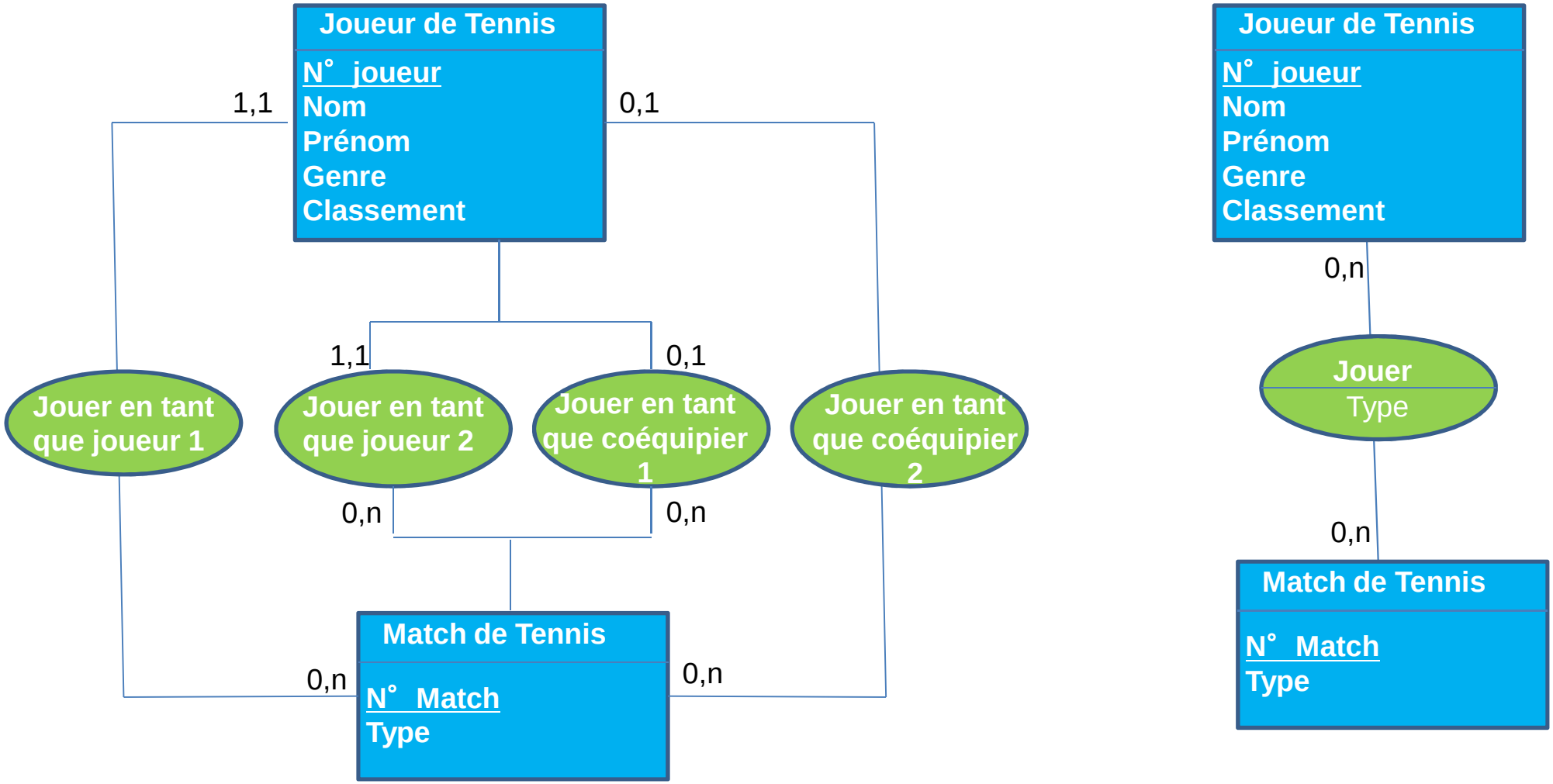




Règles de bonne formation d'un MCD

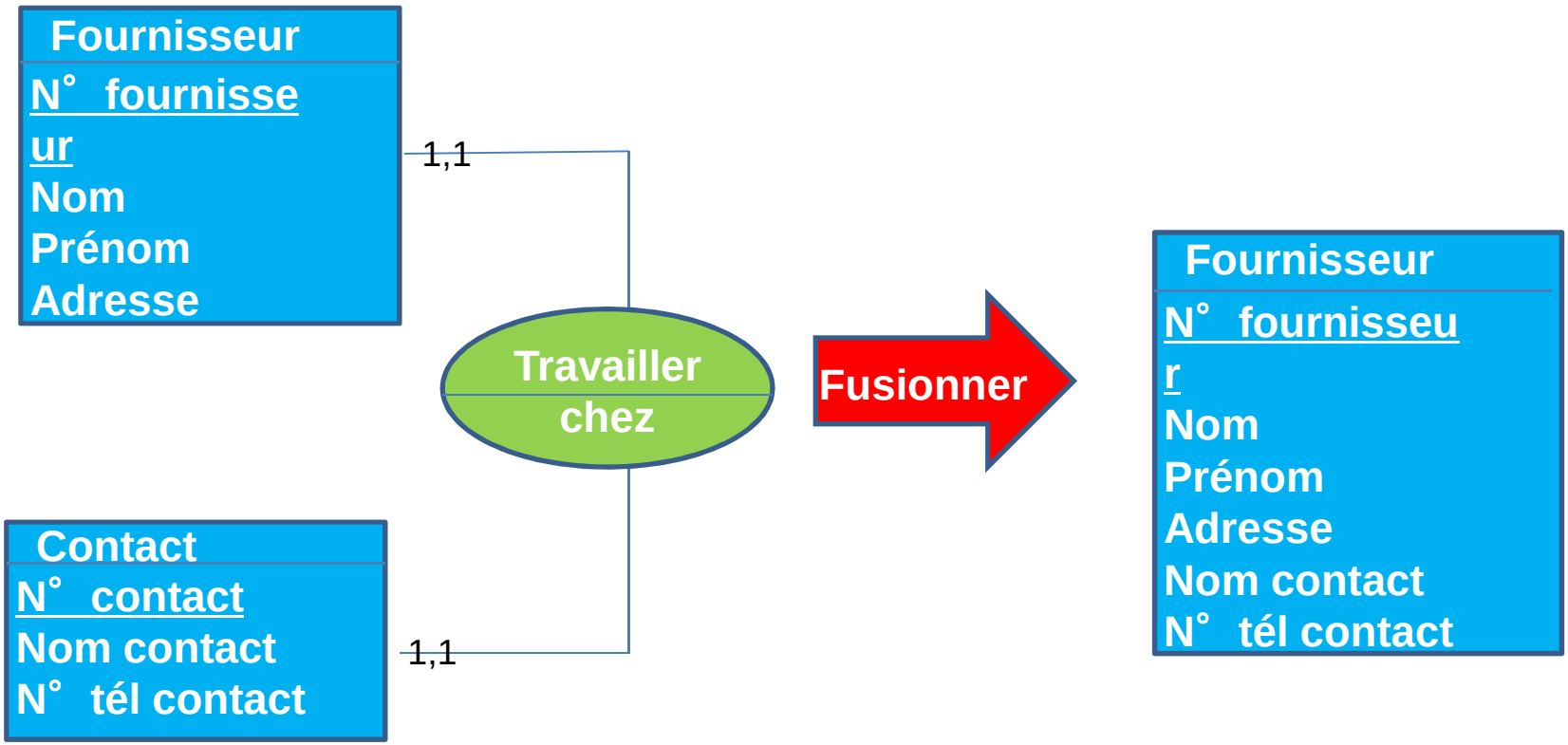
- Règles de fusion/ suppression entités/associations

Il faut factoriser les associations quand c'est possible



Règles de bonne formation d'un MCD

Il faut aussi se poser la question de l'intérêt de l'association quand les cardinalités maximale sont toutes de 1



A retenir….

Entités

Règle 1 Toute entité présente dans un MCD doit obligatoirement comporter un identifiant

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

Règle 3 Un attribut ne peut en aucun cas être partagé par plusieurs entités/associations

Règle 4 Un attribut est une donnée élémentaire, ce qui exclut des données calculées ou dérivées

Règle 5 Deux occurrences de l'entité ne pourraient avoir la même valeur pour leur identifiant

A retenir….

Associations

Règle 6 Un attribut peut être placé dans une association uniquement lorsqu'il dépend de toutes les entités liées par l'association

Règle 7 La concaténation des identifiants des entités liés à une association constitue l'identifiant de cette association (cet identifiant n'est pas mentionné sur le modèle (il est implicite).

Règle 8 L'expression de la cardinalité est obligatoire pour chaque patte d'une association

Règle 9 Une cardinalité minimal est toujours 0 ou 1 est une cardinalité maximale est toujours 1 ou n

Règle 10 Une association binaire de cardinalité (1,1) ne peut en aucun cas porter de propriétés !

Règle 11 Une association binaire ne peut en aucun cas porter des cardinalités 1,1 des deux extrémités !

Exercice de modélisation

- Un éditeur souhaite installer une base de données pour mémoriser les informations suivantes:

- les livres sont identifiés par leur numéro ISBN. Un livre possède un titre et un prix de vente. Il est écrit par un ou plusieurs auteurs.
- Chaque livre est tiré en une ou plusieurs éditions, datées et identifiées par leur ordre (première édition, seconde édition, etc.). Chaque édition comporte un certain nombre d'exemplaires. Le prix de vente peut changer d'une édition à l'autre. Chaque occurrence d'Edition décrit une édition d'un livre.
- les auteurs sont identifiés par leur nom et prénoms et peuvent avoir un pseudonyme. Pour chaque livre, un auteur perçoit des droits d'auteur annuels, calculés comme un pourcentage des ventes (il est aussi fonction du nombre d'auteurs).
- les libraires (identifiés par leur nom et adresse complète) commandent des livres en précisant l'édition et le nombre d'exemplaires désiré et la quantité commandée.

- **Question**: Composer le schéma conceptuel correspondant à cette base de données

Modélisation des données : UML

- UML

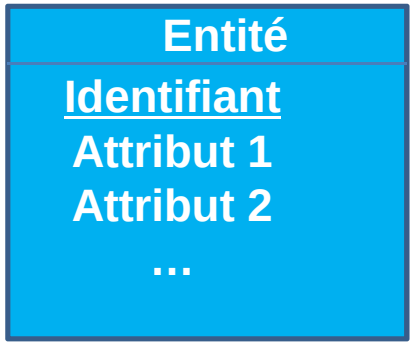
UML (Unified Modeling Language) :
- autre langage de modélisation
- langage dédié à l'objet
- plusieurs types de diagramme, dont un utile en bases de données :

le **diagramme de classes**

Modélisation des données : UML

- Lien / traduction entre UML et Merise

Merise

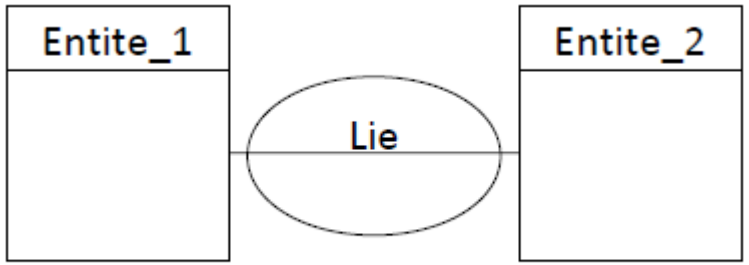


UML

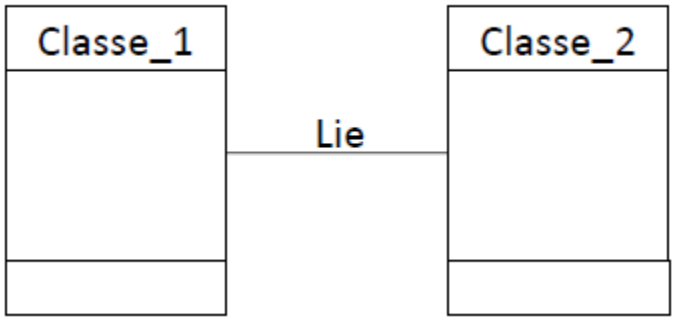
Modélisation des données : UML

- Lien / traduction entre UML et Merise: association

Merise



UML



Modélisation des données : UML

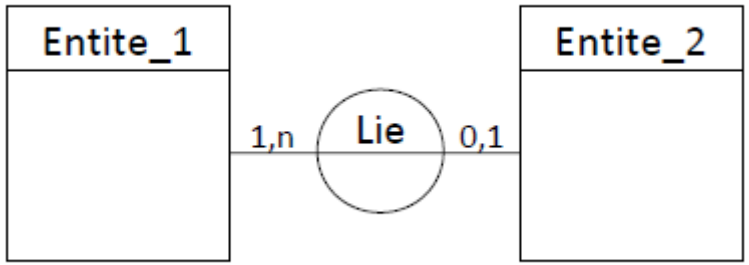
- Lien / traduction entre UML et Merise: association

Merise	**UML**
Lien vers 0 ou 1 : **0,1**	Lien vers 0 ou 1 : **0..1**
Lien vers 1 : **1,1**	Lien vers 1 : **1**
Lien vers 0 ou plusieurs : **0,n**	Lien vers 0 ou plusieurs : *****
Lien vers 1 ou plusieurs : **1,n**	Lien vers 1 ou plusieurs : **1..***

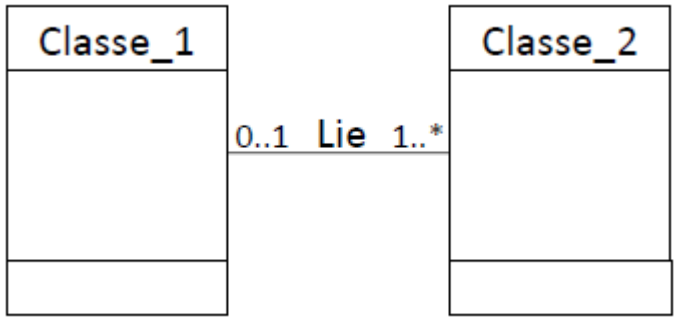
Modélisation des données : UML

- Lien / traduction entre UML et Merise: association et cardinalité

Merise



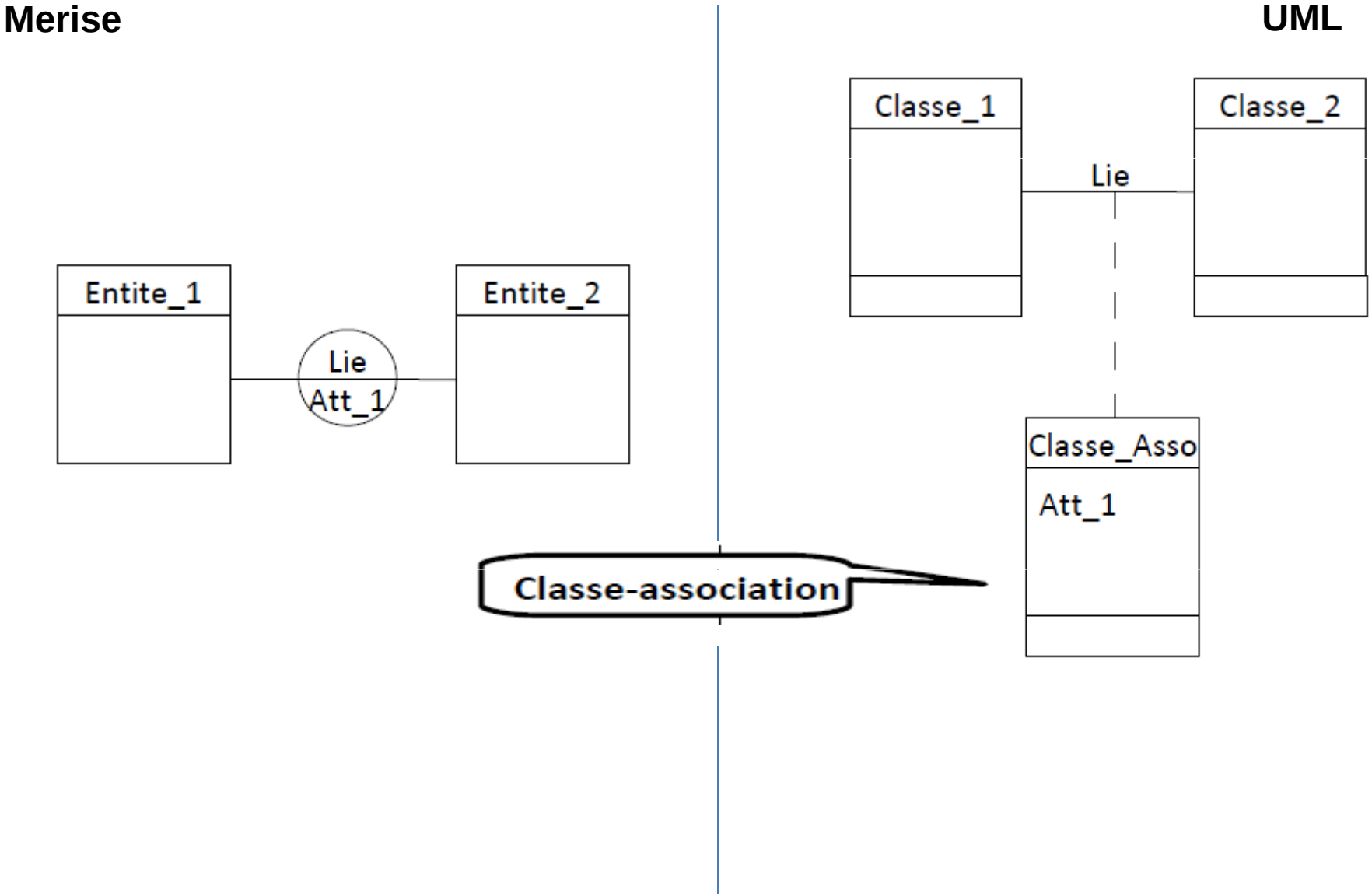
UML



Inversion du sens des cardinalités

Modélisation des données : UML

- Lien / traduction entre UML et Merise: association et cardinalité

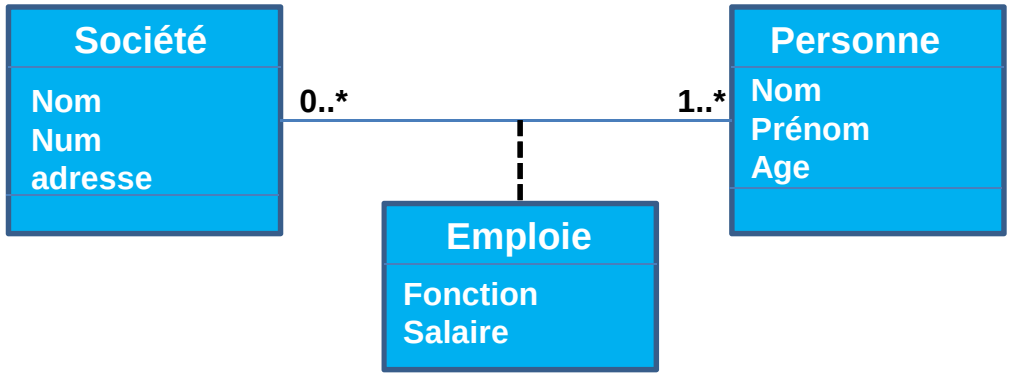


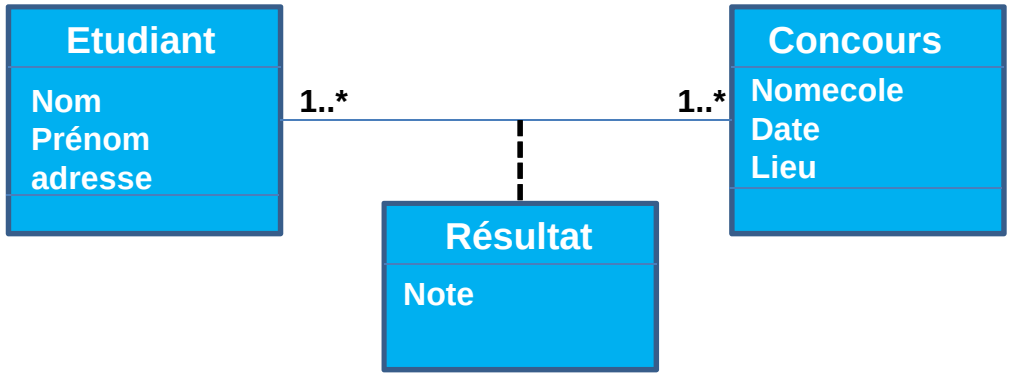
Modélisation des données : UML

- Lien / traduction entre UML et Merise:

- Classe-association :

- Exemples:





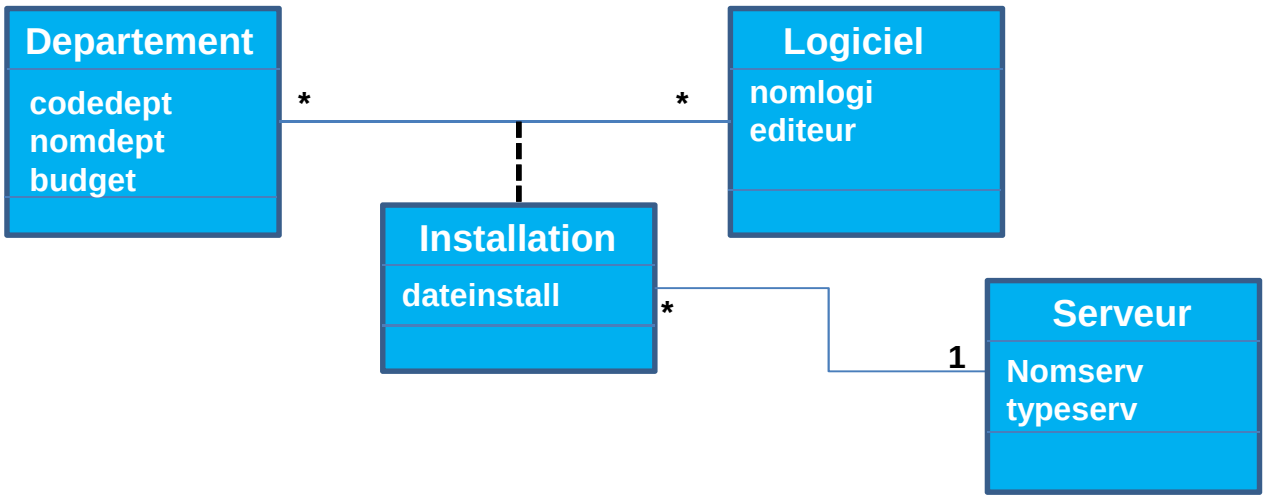
Modélisation des données : UML

❑ Lien / traduction entre UML et Merise:

❑ Utilisation d'une classe-association :

✓ Une classe association peut participer à d'autres relations:

- L'exemple met en œuvre la classe-association **Installation** en liaison avec la classe **Serveur**. Il met en évidence qu'une installation (constituée d'un logiciel et d'un département à une date donnée) est associée à un seul serveur (multiplicité 1 du côté Serveur).
- Contrainte d'unicité: un logiciel d'un département n'est installé que sur un seul serveur.



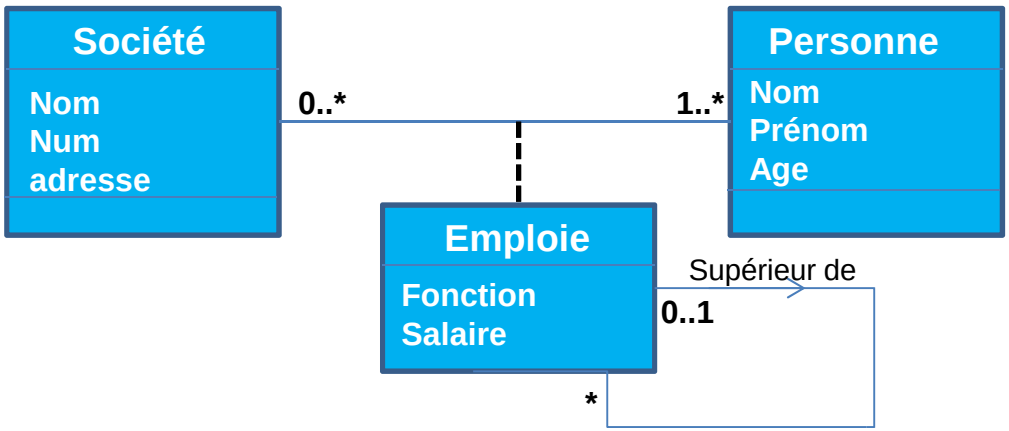
Modélisation des données : UML

❑ Lien / traduction entre UML et Merise:

❑ Auto-association sur classe-association :

✓ Exemple:

- pour préciser qu'une personne est le supérieur d'une autre personne. On ne peut pas ajouter une association réflexive sur la classe *Personne*. En effet, une personne n'est pas le supérieur d'une autre dans l'absolu. Une personne est, en tant qu'employé d'une entreprise donnée, le supérieur d'une autre personne dans le cadre de son emploi pour une entreprise donnée. Il s'agit donc d'une association réflexive, non pas sur la classe *Personne,* mais sur la classe-association *Emploie*

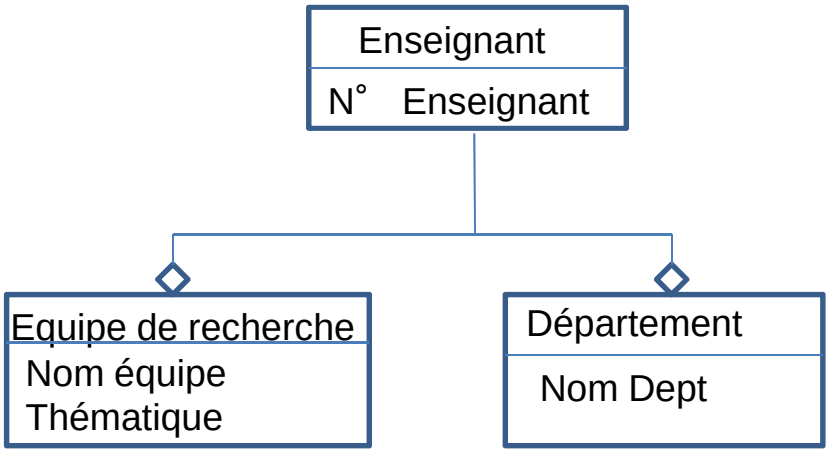


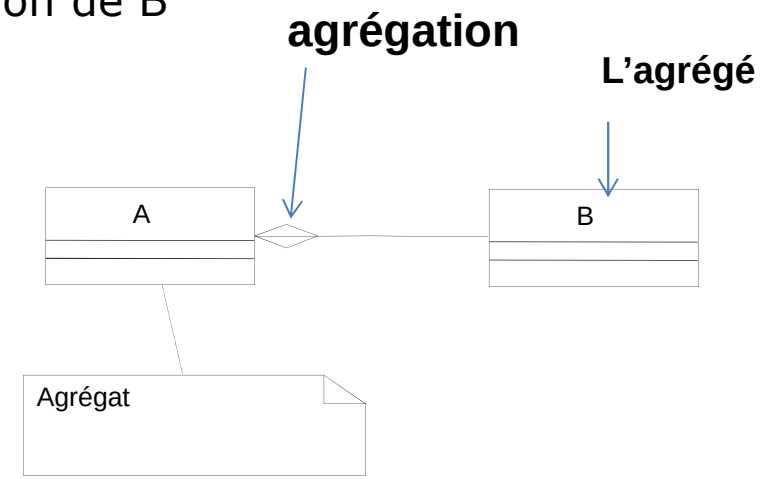
Modélisation des données : UML

- Les plus d'UML: agrégation

Association particulière dans laquelle l'une des entités décrit un tout alors que l'entité associée décrit des parties. L'entité qui représente les tout est appelée composite, l'entité qui représente une partie du tout est appelée composant.

- Propriétés de l'agrégation
  - A « contient » des instances de B
  - La suppression de A n'implique pas la suppression de B
  - L'élément agrégé peut être partagé
- Exemple:





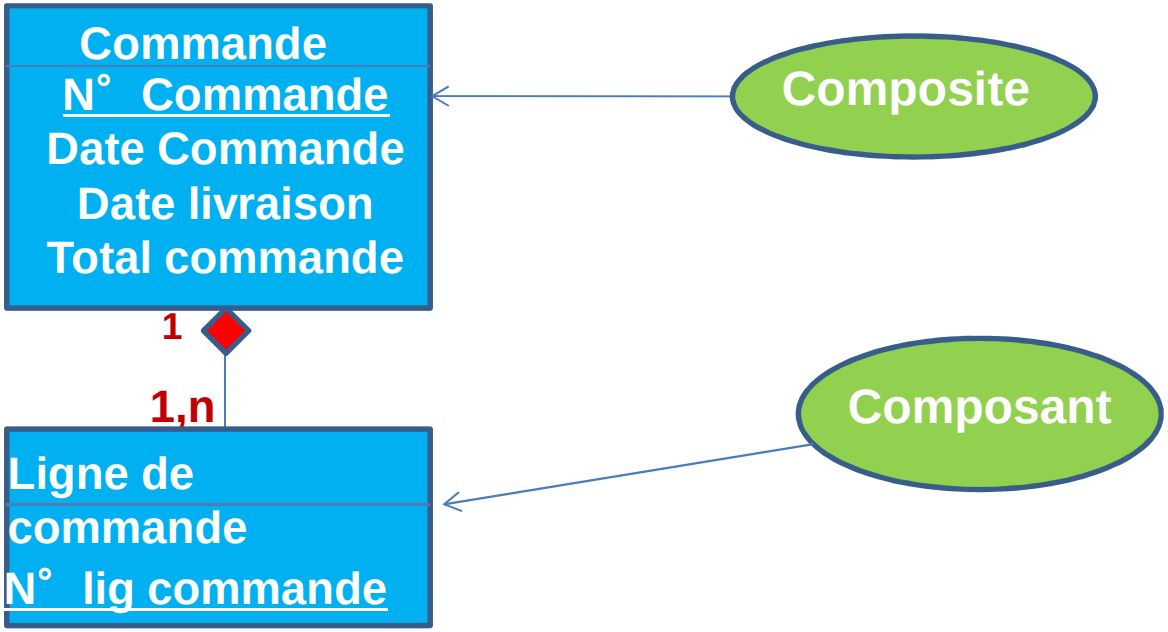
- **L'enseignant est un composant d'une (ou plusieurs) équipe de recherche**
- **La disparition d'une équipe de recherche n'entraine pas la disparition d'un enseignant**

Modélisation des données : UML

- Les plus d'UML: composition

Une forme forte d'agrégation avec le composé qui à chaque moment a une possession exclusive des parties. Le temps de vie des parties coïncide avec celui du composé

- Remarque 1:Dans une composition, la multiplicité du côté du composite est toujours à 1, car un composant doit appartenir à un seul et un seul composite
- Remarque 2: la création d'une occurrence d'un composant exige la présence d'un composite pour s'y associer.
- Remarque 3 : la fin de vie d'une occurrence de composite entraîne en cascade la fin de vie de toutes les occurrences des composants associés.



Modélisation des données : UML

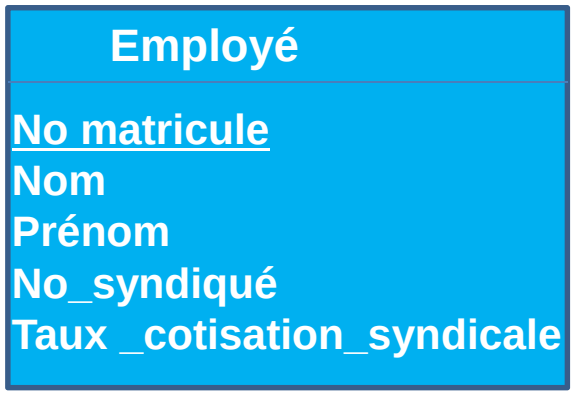
❑ Associations : Héritage

- L'association *d'héritage* est employée d'abord pour assurer le respect d'une règle de modélisation appelée la *règle* ***d'homogénéité*** .

Tous les attributs d'une entité sont pertinents à cette entité et éventuellement tous doivent posséder une valeur.

❑ Contre exemple

- Remarque 1: Transgression de la règle d'homogénéité
- Remarque 2: Certains employés ne sont pas syndiqués, notamment les employés cadres
- Remarque 3: les occurrences de l'entité représentant des employés non syndiqués ne pourront avoir de valeur pour **No_syndiqué** et **Taux _cotisation_syndicale**

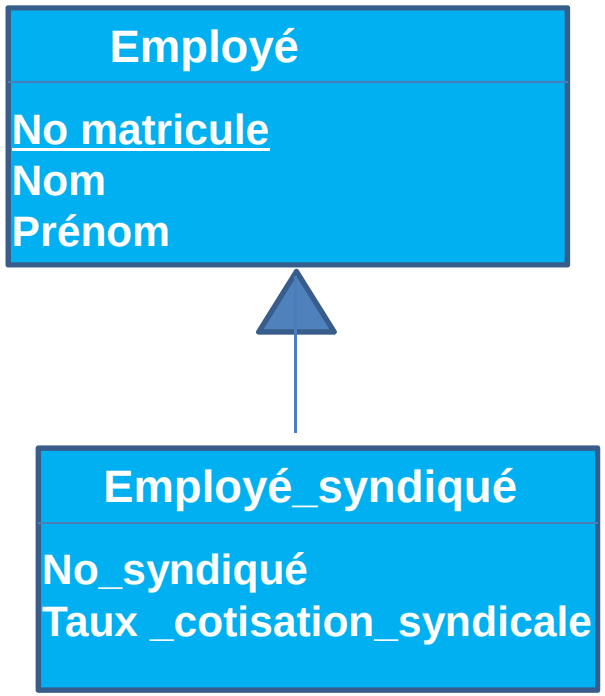


Modélisation des données : UML

- Association : Héritage

- Exemple : Héritage et respect de la règle d'homogénéité

- Remarque 1: L'entité **Employé_syndiqué** est une sorte d'entité Employé et qui hérite de tous les attributs de Employé
- Remarque 2: Toute occurrence de **Employé** possède une valeur pour chacun de ses 3 attributs.
- Remarque 3: Dans le cas de **Employé_syndiqué** chaque Occurrence possède une valeur pour les 5 attributs



Modélisation des données : UML

- Association : Héritage

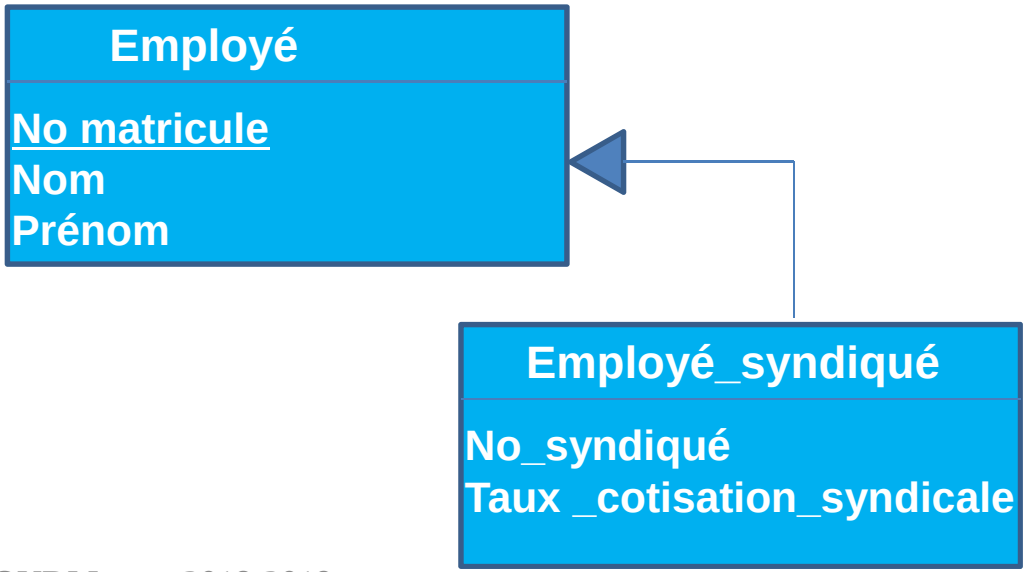
Type d'association qui définit la structure d'une entité en fonction d'une autre. Une entité appelée le **supertype** identifie les attributs communs, une autre précise les attributs spécifiques à un **sous-type** de la première ; le **sous-type** hérite à la fois des attributs, dont l'identifiant, et des associations de son **supertype**

- Exemple : L'héritage porte sur les attributs et les associations

— Remarque 1: L'entité Employé_syndiqué hérite des attributs de Employé, elle hérite aussi de l'association avec Poste

— Remarque 2: Un employé_syndiqué est une sorte d'employé et tout employé occupe un poste.

— Remarque 3: **Employé syndiqué** est un sous-type de l'entité **Employé**



EXERCICES DE MODÉLISATION

Plan du cours

I. Système de Gestion de Base de Données (SGBD)

- Introduction Générale
- Modélisation des données
  - Merise
  - UML
- Modèle Logique de Données (MLD)

Modèle logique de données (MLD): Terminologie

- Le Modèle Logique des Données (MLD) est une étape intermédiaire pour passer du modèle E/A, qui est un modèle sémantique, vers une représentation physique des données : fichiers, SGBD hiérarchique, SGBD réseau, SGBD relationnel.

- Nous nous limitons au seul MLD relationnel, qui prépare le passage aux SGBD relationnels.

Modèle logique de données (MLD): Terminologie

- Comment est constitué un MLD (relationnel) et comment l'établir?
- Tables, lignes, colonnes, attribut et domaine
- clefs primaires et clefs étrangères
- Schéma relationnel

Modèle logique de données (MLD): Terminologie

- Tables

✓ Lorsque les données ont la même structure (par ex. renseignement relatifs à un client), on peut alors les organiser en tables dans lesquelles:

  - Les colonnes décrivent les champs en commun
  - Les lignes contiennent les valeurs de ces champs pour chaque enregistrement

- Exemple

N° Client	Nom	Prénom	Adresse
1	Durand	Marie	2, rue la Paix
2	Motte	Pierre	7, rue Cler
…	…	…	…

Modèle logique de données (MLD): Terminologie

- Tables, lignes, colonnes
  - Une table est une relation comportant des lignes (tuples) et des colonnes

Relation

Considérant N ensembles E1, E2,…En. Tout sous-ensemble du produit cartésien des N ensembles, noté E1 X E2 X…X En, constitue une relation

- Exemple:
- $E_1 = \{a, b\}$ et $E_2 = \{c, d\}$, sont deux ensembles
- le produit cartésien est: $E_1 \times E_2 = \{(a,c),(a,d),(b,c),(b,d)\}$
- l'ensemble $R_1 = \{(a,c),(a,d)\}$ est un sous ensemble de $E_1 \times E_2$ il constitue une relation.
- Les éléments de la relation sont appelés des tuples.

E1	E2
a	c
a	d
b	c
b	d

R1

Modèle logique de données (MLD): Terminologie

- Attribut, et domaine

Attribut

Colonne d'une relation caractérisée par un nom

Domaine

Ensemble des valeurs admises pour un attribut. Il établit les valeurs acceptables dans une colonne

- Exemple :
  - Domaine= {Entier, réel, booléen, caractère}

Modèle logique de données (MLD): Terminologie

- clef primaire

- Les lignes d'une table sont unique→il existe au moins une colonne qui sert à identifier les lignes: il s'agit de la clef primaire de la table

- Les propriétés et conventions requises
  - La valeur vide (NULL) est interdite
  - La valeur de la clef primaire d'une ligne ne devrait pas changer au cours du temps
  - On souligne les clefs primaire dans un MLD

Modèle logique de données (MLD): Terminologie

- clef primaire

clef primaire

Ensemble minimal de colonnes qui permet d'identifier de manière unique chaque tuple dans une table (primary key)

- une clef primaire est simple si elle est composée d'une seule colonne
- Une clef primaire composée peut être constituée de deux colonnes ou plus
- Exemple
  - Client (N°, Nom, prénom, adresse)
  - *Appartement(N°, adresse*, superficie)

Modèle logique de données (MLD): Terminologie

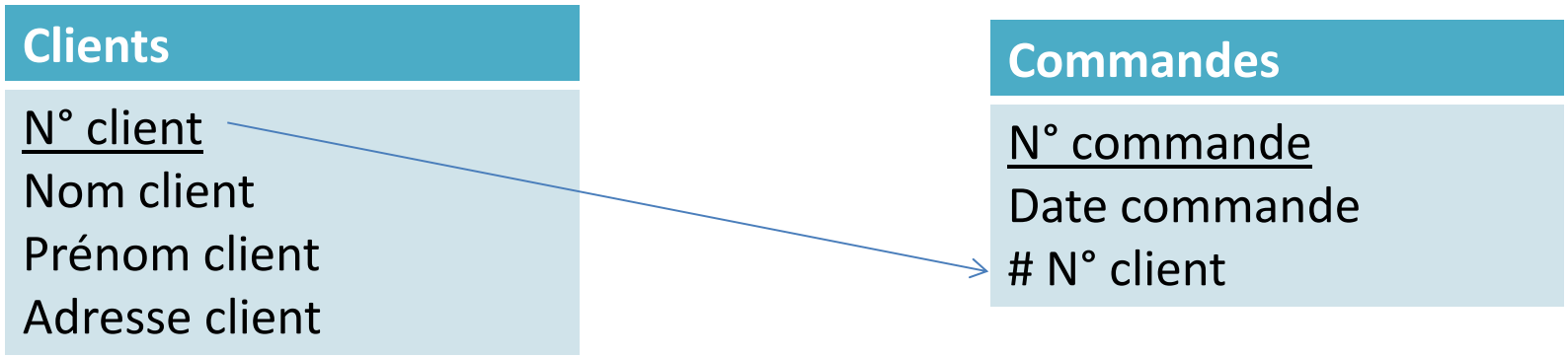
- clef étrangère

clef étrangère

une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables. La clef primaire de la première table est dupliquer dans la deuxième. On l'appelle aussi clef externe (Foreign key)

- Convention:
  - On fait précéder par # la clef étrangère

- Exemple



Modèle logique de données (MLD): Terminologie

- Tables, lignes, colonnes, attribut, et domaine

Modèle relationnel	Modèle conceptuel
Table/relation	Entité
Ligne/tuple	Occurrence d'entité
Nom de colonne	Attribut d'entité
clef primaire/ étrangère	identifiant

Modèle logique de données (MLD): Terminologie

Remarques

Rq1: Une même table peut avoir plusieurs clefs étrangères mais une seule clef primaire(éventuellement composée de plusieurs colonnes)

Rq2: Une clef étrangère peut aussi être primaire (dans la même table)

Rq3: Une clef étrangère peut être composée (c'est le cas si la clef primaire référencée est composée)

Rq4: Implicitement chaque colonne qui compose une clef primaire ne peut pas recevoir la valeur NULL

Rq5: Si une clef étrangère ne doit pas recevoir la valeur NULL, alors il faut le préciser dans la description des colonnes

Modèle logique de données (MLD): Schéma relationnel d'une BD

- Schéma relationnel
  - Les tables sont appelées relations
  - Les liens entre les clefs étrangères et leur clefs primaires sont symbolisés par un connecteur

Modèle logique de données (MLD): Schéma relationnel d'une BD

- Entité

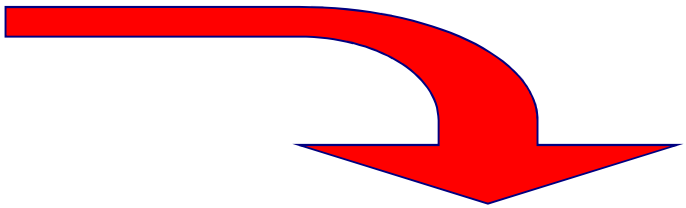
Règle

Toute entité (dans un MCD) devient une table (dans un MRD) dans laquelle les attributs deviennent les colonnes et l'identifiant de l'entité constitue la clef primaire de la table

Client
Codcli
Nomcli
Adrcli

Entité

se traduit par



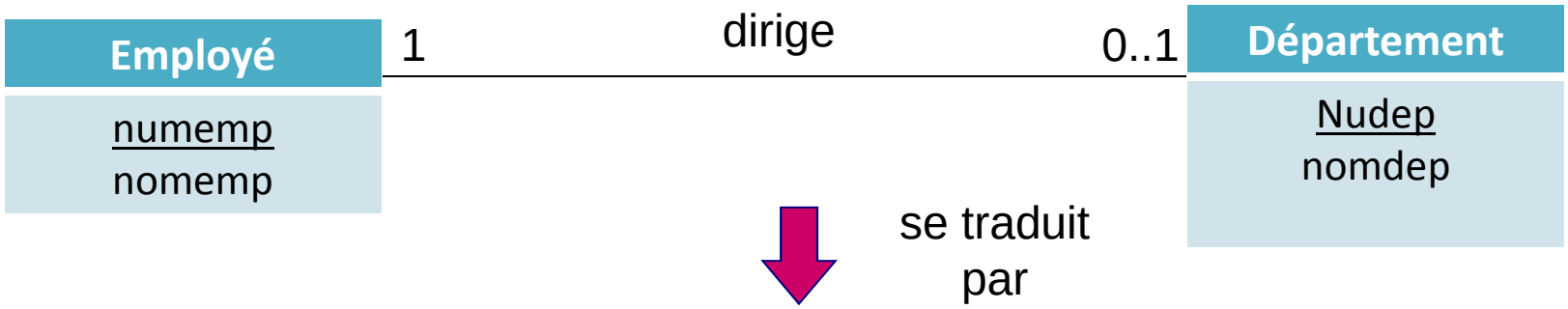
Client (**codcli**, nomcli, adrcli)

Table

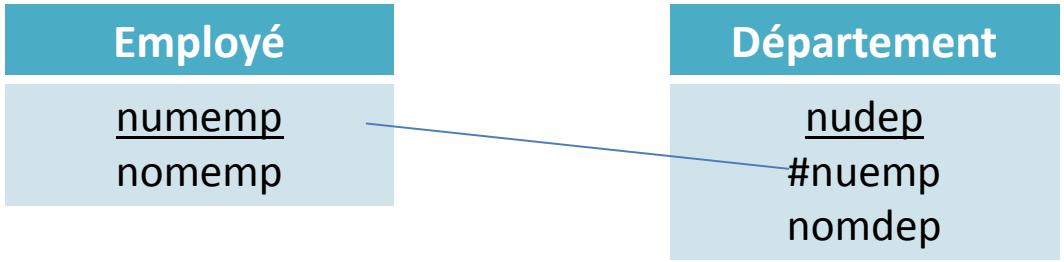
Modèle logique de données (MLD): Schéma relationnel d'une BD

- Association binaire 1/1- 0/1

  se traduit en ajoutant une clef étrangère (identifiant de l'entité de cardinalité (0,1)) à la table provenant de l'entité dont la cardinalité est (1,1).



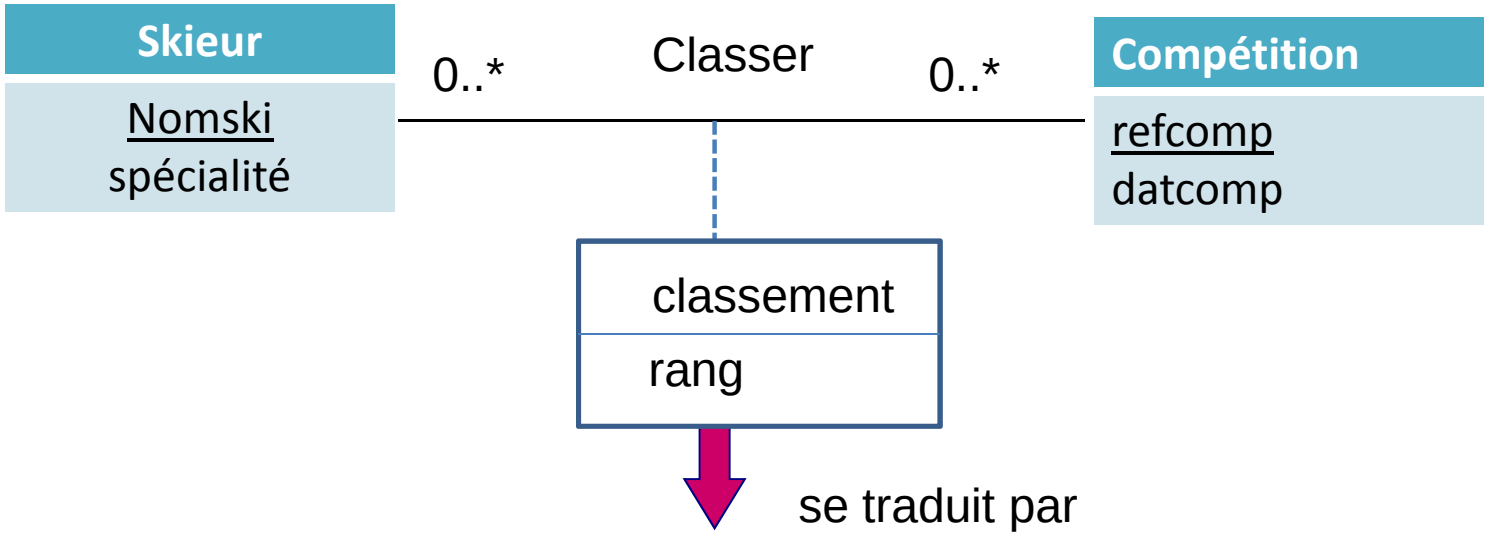
Employé (**<u>nuemp</u>**, nomemp)
Département (**<u>nudep</u>**, nomdep**, #nuemp**)



Modèle logique de données (MLD): Schéma relationnel d'une BD

- Association binaire plusieurs à plusieurs

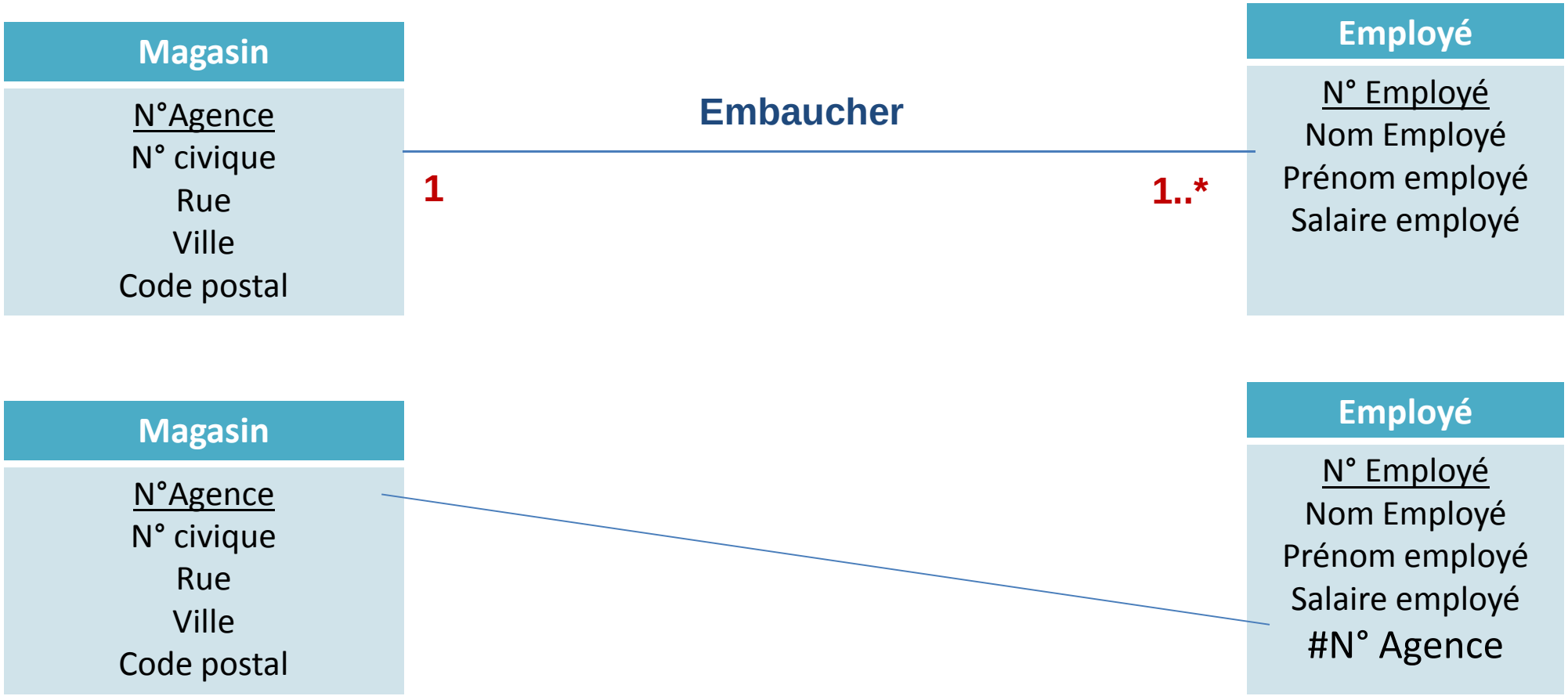
se traduit par une nouvelle table dont la clef primaire est composée des identifiants des deux entités. Les éventuelles propriétés de l'association deviennent les attributs de cette table.



Classer **(#nomski, refcomp**, rang)

Modèle logique de données (MLD): Schéma relationnel d'une BD

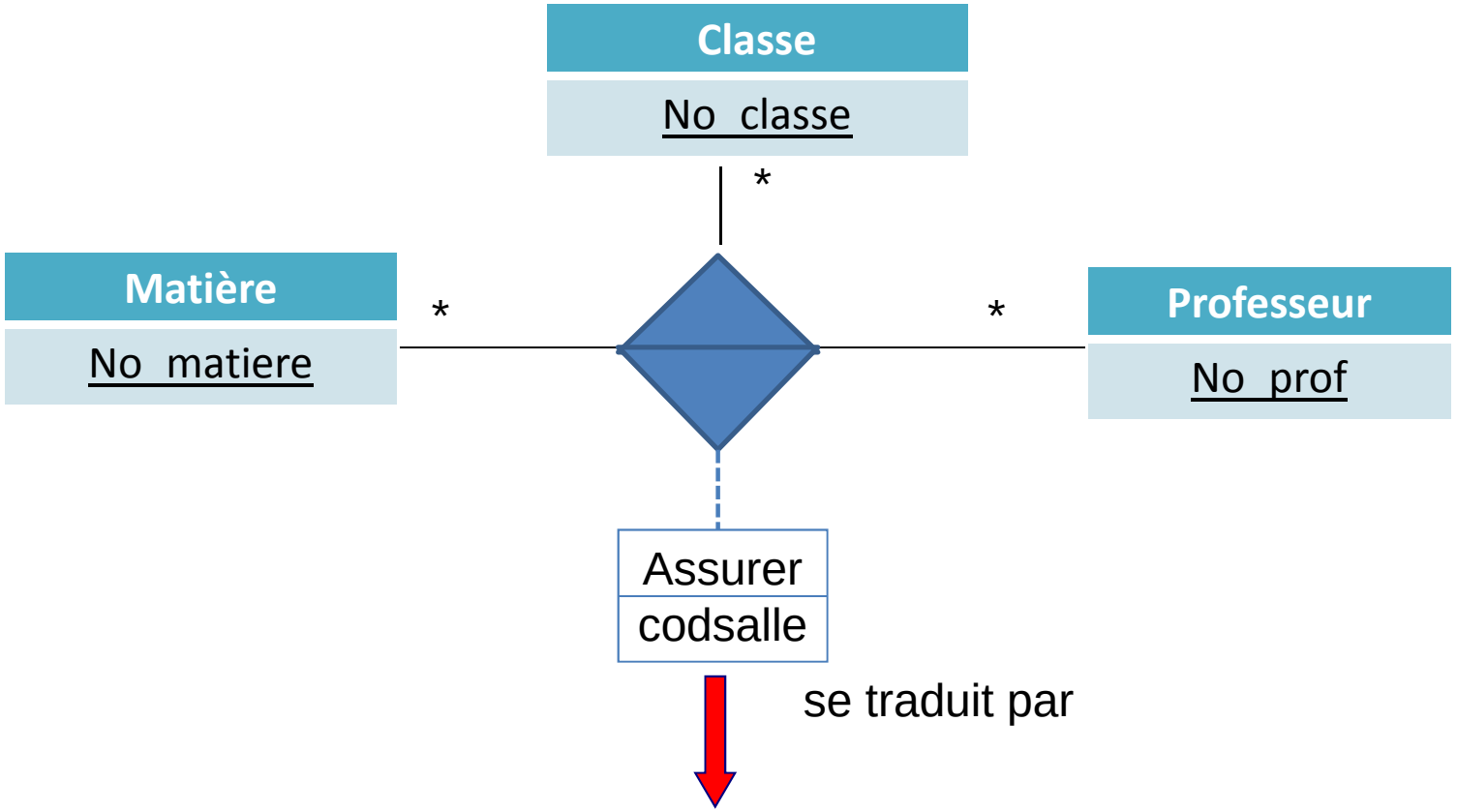
- Association binaire un à plusieurs



Modèle logique de données (MLD): Schéma relationnel d'une BD

- Association n-aire (n>2)

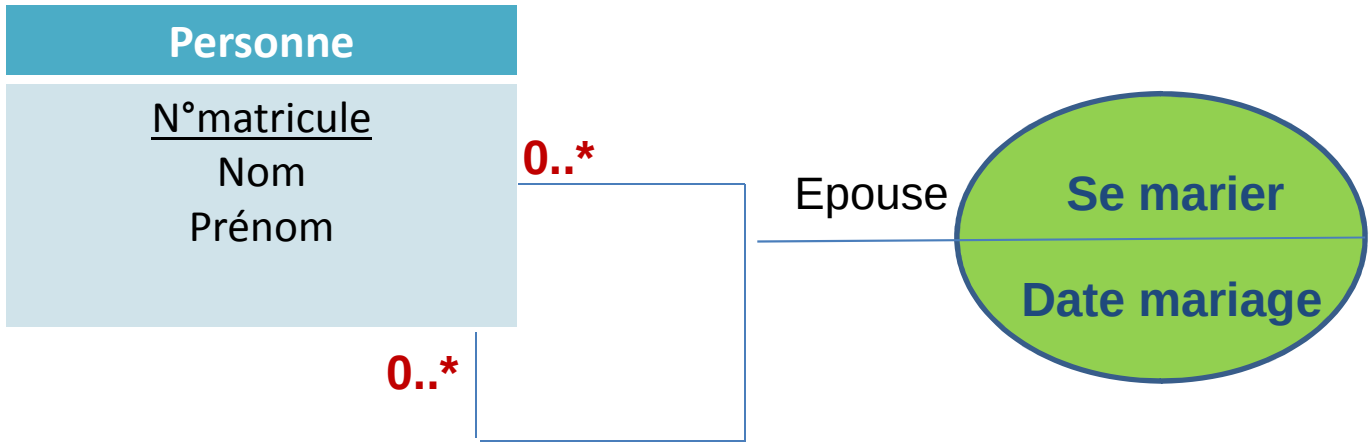
on crée une table ayant pour clef primaire les identifiants des différentes entités de l'association. Les éventuelles propriétés de l'association deviennent les attributs de la table.

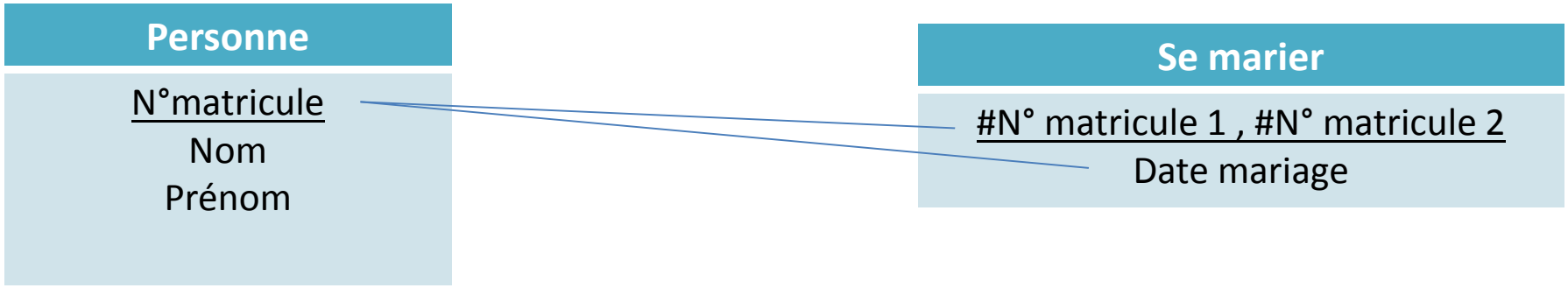


Assurer (#**No_classe, No_matiere, No_prof,** codsalle)

Modèle logique de données (MLD): Schéma relationnel d'une BD

- Association réflexive





Modèle logique de données (MLD): Schéma relationnel d'une BD

- La composition

Un client est identifié par son nom, prénom et adresse

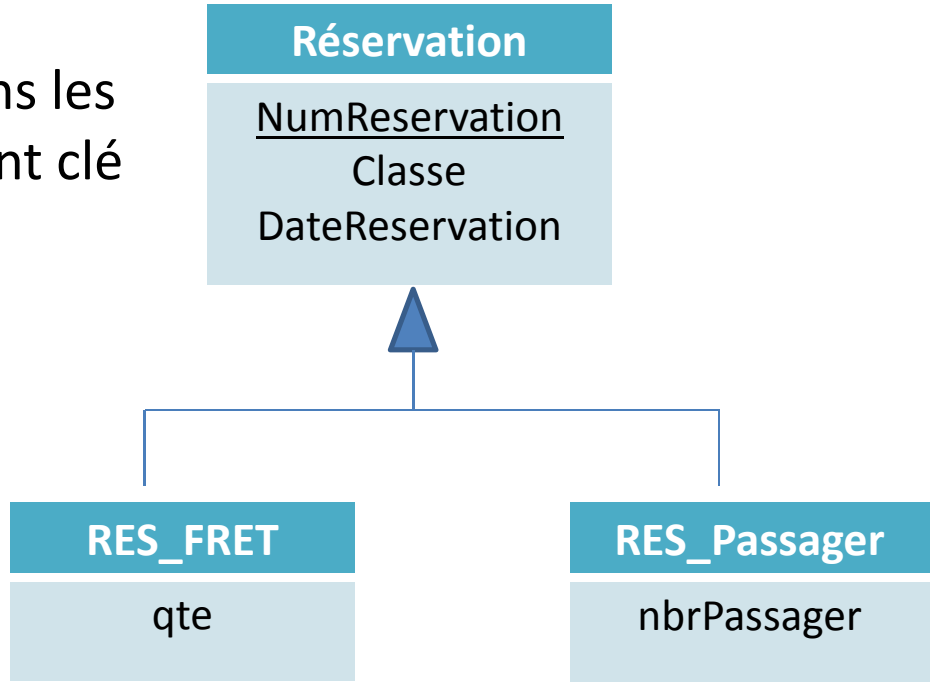
UML

Client (Nom, Prénom, #idAdresse, Tél)
Adresse(idAdresse, NumRue, NomRue, CP, NomVille)

- L'entité faible a pour clé sa propre clé + celle de l'entité forte (qui est clé étrangère)

Modèle logique de données (MLD): Schéma relationnel d'une BD

- ❑ L'héritage
- ✓ Chaque classe fille→ une relation
- ✓ Clé primaire de la classe mère migre dans les relations issues des classes fille et devient clé et clé étrangère



Réservation(NumReservation, Classe, DateReservation)
RES_FRET(#NumReservation, qte)
RES_Passager(#NumReservation, nbrPassager)

Modèle logique de données (MLD): Règles d'intégrité structurelles

- Les règles d'intégrité sont les règles que doivent vérifiés les données contenues dans une base de données. Ces règles sont inhérentes au modèle de données

- On distingue plusieurs règles structurelles correspondant au concepts:
  - Entité
  - Domaine
  - Clef (contrainte référentielle)

Modèle logique de données (MLD): Règles d'intégrité structurelles

Contrainte d'entité

Contrainte imposant que toute relation possède une clef primaire et tout attribut participant à cette clef primaire est non nul

❑Valeur nulle: valeur conventionnelle introduite dans une relation pour représenter une information inconnue ou inapplicable

Contrainte de Domaine

Contrainte imposant que la colonne d'une relation doit comporter des valeurs vérifiant une assertion logique

❑ L' assertion logique est l'appartenance à une plage ou liste de valeurs:

✓Exemple:

- Le domaine des salaires mensuels qui sont des réels compris entre 900 et 3000€
- L'âge d'une personne est compris entre 1 et 150

Modèle logique de données (MLD): Règles d'intégrité structurelles

Contrainte référentielle (clef étrangère)

Contrainte d'intégrité portant sur une relation R1, consistant à imposer que la valeur connue d'un groupe d'attributs apparaisse comme valeur de clef dans une autre relation R2

- Lors d'une insertion d'une valeur dans une relation, la valeur des attributs doit exister dans la relation référencée
- Exemple:
  - Insertion : Commande
  - Contrainte: le client correspondant doit exister
- Lors d'une suppression dans la relation référencée , les tuples référençant ne doivent pas exister
  - Exemple: la suppression d'un client entrainera la vérification qu'aucun client ne référence une commande