

Analyse et conception des systèmes d'information

La méthode Merise

Le modèle conceptuel de données (MCD)

Objectif du MCD

- Le modèle conceptuel des données est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique.
 - Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information.
 - Il s'agit donc d'une représentation des données, facilement compréhensible. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».

MCD

1. Le dictionnaire des données

- **Inventaire** exhaustif des données du domaine étudié.

- ***Unicité sémantique*** : à une donnée correspond une **mnémonique**, il faut parvenir à ce que chacun de ces mnémoniques ait une signification unique au sein de l'organisation. Il faut pour cela éviter :

- **Les redondances** : existence d'une donnée en double
- **Les synonymes** : existence de deux mnémoniques décrivant le même objet (difficile à détecter)
 - Libelle Article
 - Nom Produit
 - Il faut trancher en choisissant un des mnémonique
- **Les polysèmes** : mnémonique unique pouvant décrire plusieurs objets différents
 - Date (sous entendu de facture)
 - Date (sous entendu de commande)
 - Pour lever l'ambiguïté il suffit de parler de Date Facture et Date Commande
- **Contraintes d'Intégrité** : (CI) une contrainte d'intégrité est une règle à observer pour que chacune des valeurs que revêt une donnée soit correcte.

DESCRIPTIF DES DONNEES

Domaine :
Processus :

Rédacteur :

Date :

Rubriques	Libellé	Type	Mode	D1	D2	D3	D4
identificateur	libellé	entier réel date chaîne booléen	mémorisée calculée paramètre	x x	x x		x x
						x	

Dictionnaire de données :

Propriété	Désignation
idDotation	Identifiant de la dotation
designation	Désignation de la dotation
dateReception	Date de réception de la dotation
matriculeV	Matricule de véhicule
marqueV	Marque de véhicule
modeleV	Modèle de véhicule
codeLiv	Code de livraison

2. Le modèle conceptuel des données : le modèle entité/association

- a) les concepts de base du modèle E/A,
- b) vérification et normalisation du modèle E/A,
- c) les contraintes d'intégrité dans le modèle E/A.

a) Les concepts de base

ENTITE:

Une entité est un objet **abstrait** ou **concret** de l'univers du discours.

Une entité peut être :

- Une personne (CLIENT)
- Un lieu (DEPOT, BUREAU, ATELIER, ...)
- Un objet documentaire (LIVRE, OUVRAGE, DOSSIER,...)

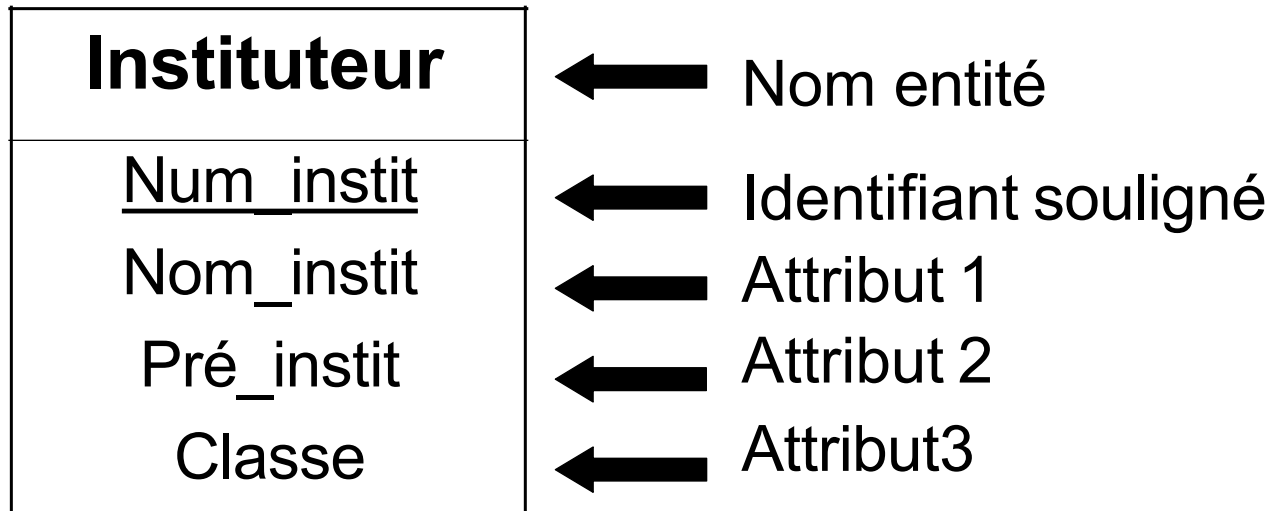
Après avoir réaliser le dictionnaire de données, il faut regrouper ces données par paquet homogène.

Ces paquets représentent les entités.

Une entité est caractérisée par :

- Un **identifiant**
- Une **suite d'information** liée à cet identifiant.

Représentation graphique d'une entité :



REMARQUE:

1. Dans la plupart des études de cas, l'entité « DATE » est présente : c'est une entité formée d'un seul attribut

DATE

(calendrier) - date : JJ/MM/AA

2. De même, on peut créer une entité « HEURE »

IDENTIFIANT:

- C'est une propriété particulière de l'entité; représentation de l'une des occurrences de l'entité ou de l'association.
- Le meilleur moyen pour ne pas risquer d'avoir des synonymes consiste à prendre des numéro de références comme identifiant.
- Un identifiant peut être simple c.à.d. constitué d'une seule propriété élémentaire (d'ordre 1) : NUM_ELEV.
- Un identifiant peut être constitué de plusieurs propriétés élémentaires: d'ordre 2, 3 ou 4

LES OCCURRENCES D'UNE ENTITE:

- Une **occurrence** ou **tuple** est une réalisation particulière de l'entité ou un exemplaire de l'entité.
- L'ensemble des occurrences forme l'entité désignée.

Remarque :

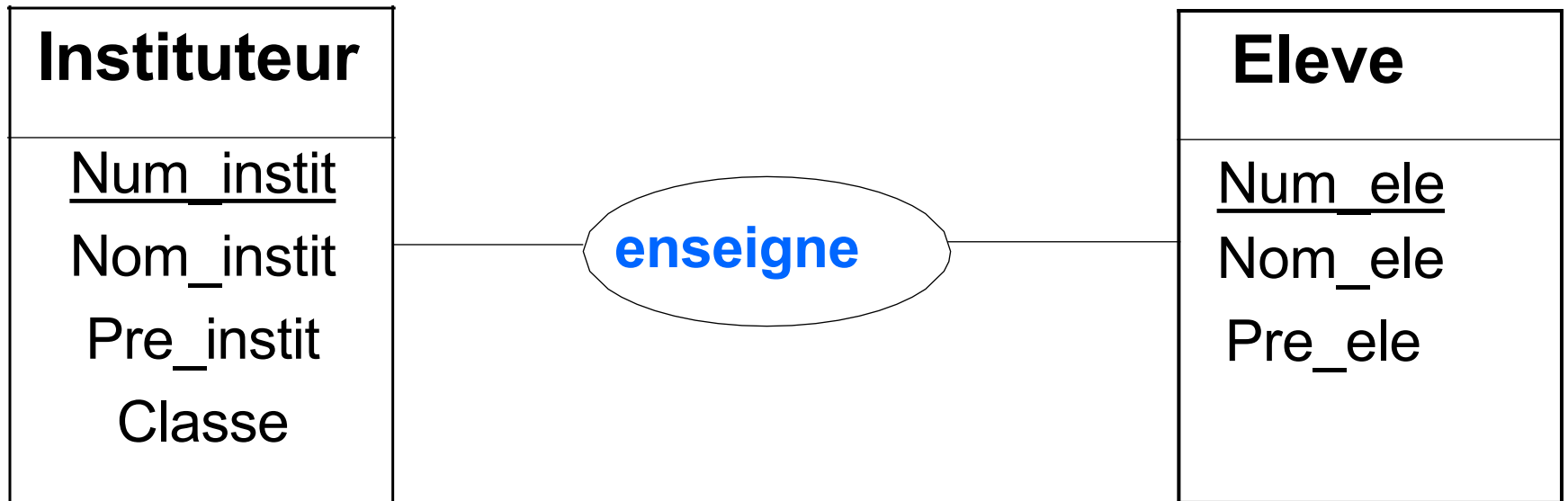
Le but d'une analyse est de pouvoir à partir d'un dictionnaire de donnée aboutir à une collection d'entité sans redondance, et ayant des *liens logiques* entre elles tel que quelque soit la donnée celle-ci sera accessible à volonté.

LES ASSOCIATIONS :

- Une association est un lien sémantique entre plusieurs entités indépendamment de tous traitements.
- Une association est souvent nommé par un verbe qui exprime le sens du lien entre les entités.
- Les liens logiques existant entre deux entités sont appelés **Associations**.

Par exemple, on peut considérer qu'il existe une association **Enseigne** entre l'entité **instituteur** et **élève** dans le cas d'une école

Représentation graphique d'une association:



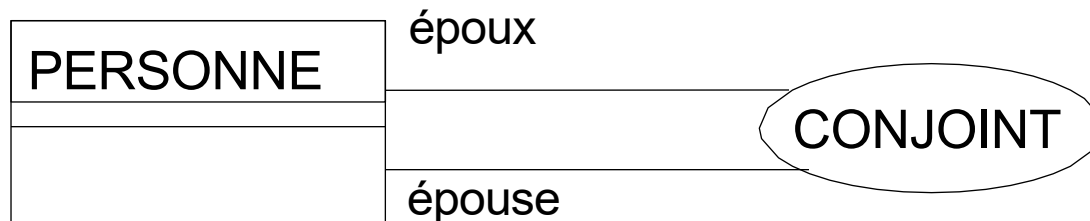
Remarques

- On peut avoir plusieurs associations sur les mêmes entités.

Ex : PROPRIETAIRE(PERSONNE, VOITURE)
et CONDUIRE(PERSONNE, VOITURE)

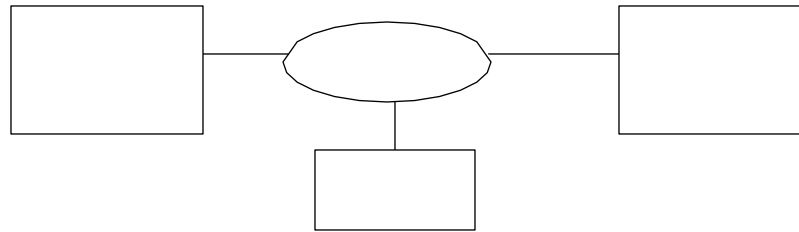
- On peut avoir une association sur une seule entités (on parle d'association 'réflexive'). On ajoute souvent dans ce cas des noms de **rôles** pour distinguer les deux occurrences.

Ex : CONJOINT(PERSONNE, PERSONNE)



- On peut avoir une association définie sur n entités (association n-aire ou d'arité n ou de dimension n ou à « n pattes »).

Ex: COURS(MATIERE, CLASSE, PROF)



Associations

- **Association** : liaison existant entre des entités.
ex. les clients *commandent* des produits

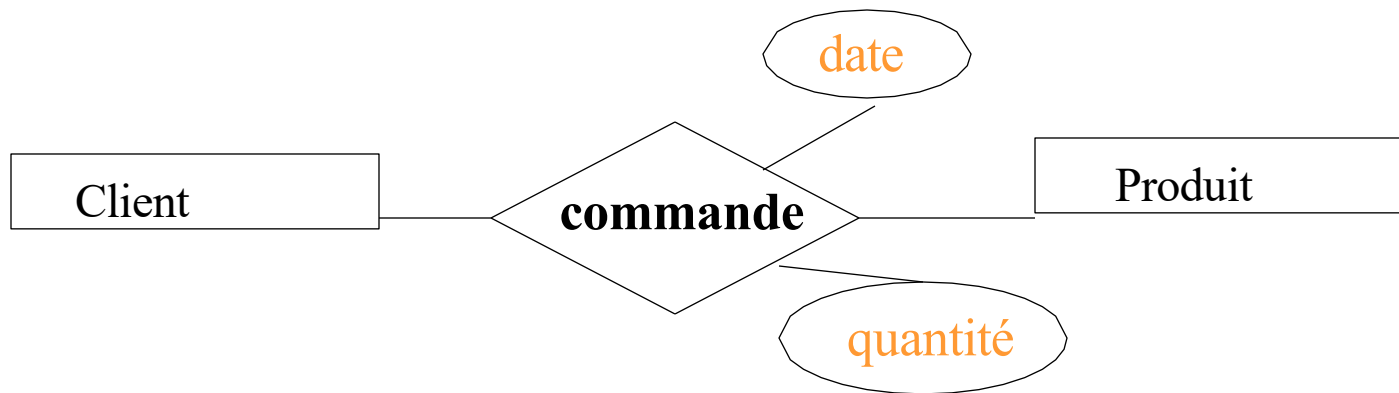


- Chaque entité joue un **rôle** dans l'association
 - Les rôles devront être précisés si l'association relie une entité à elle-même.

Attributs d'associations

- Une association peut être caractérisée par des **attributs**.

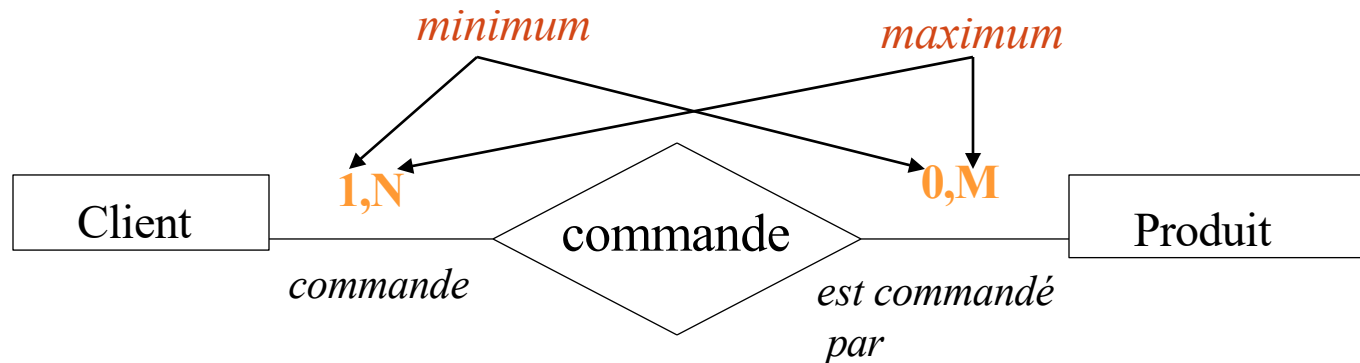
ex. *date* de la commande et *quantité* de produits commandés.



Cardinalités d'associations

- **Cardinalité d'une assoc.** : nombres **minimum** et **maximum** de participations de chaque occurrence d'entité à l'association.

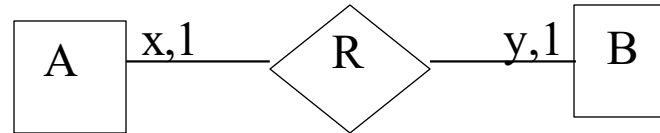
ex. un client doit commander au moins un produit ; un produit peut être commandé par zéro ou un nombre quelconque de clients.



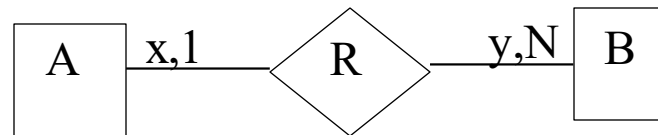
En fonction des cardinalités **maximales**, une association binaire (degré = 2) peut être de type **1-1**, **1-N** ou **N-M**

Cardinalités d'associations

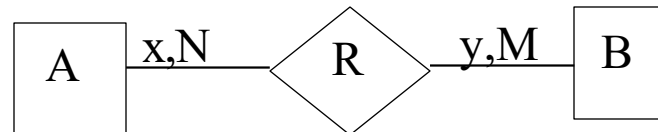
Typologie des associations binaires



Association de type 1-1 (*one-to-one*)



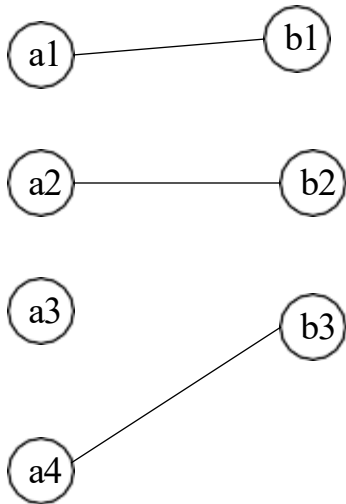
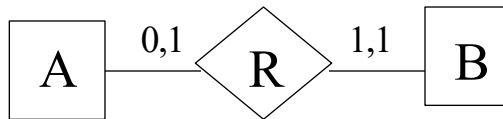
Association de type 1-N (*one-to-many*)



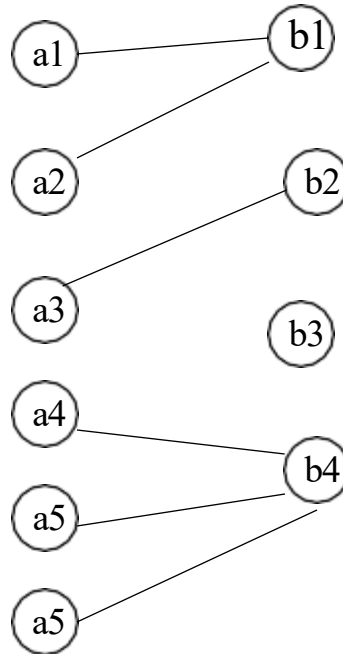
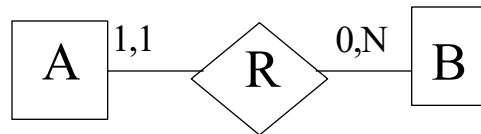
Association de type N-M (*many-to-many*)

Cardinalités d'associations

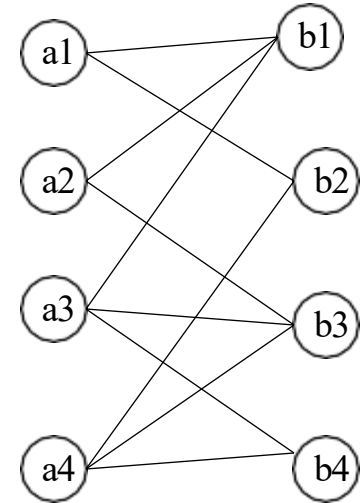
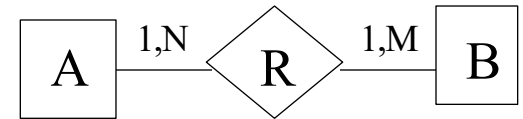
Typologie des associations binaires



One-to-one



One-to-many



Many-to-many

Quelques 'critères' de choix :

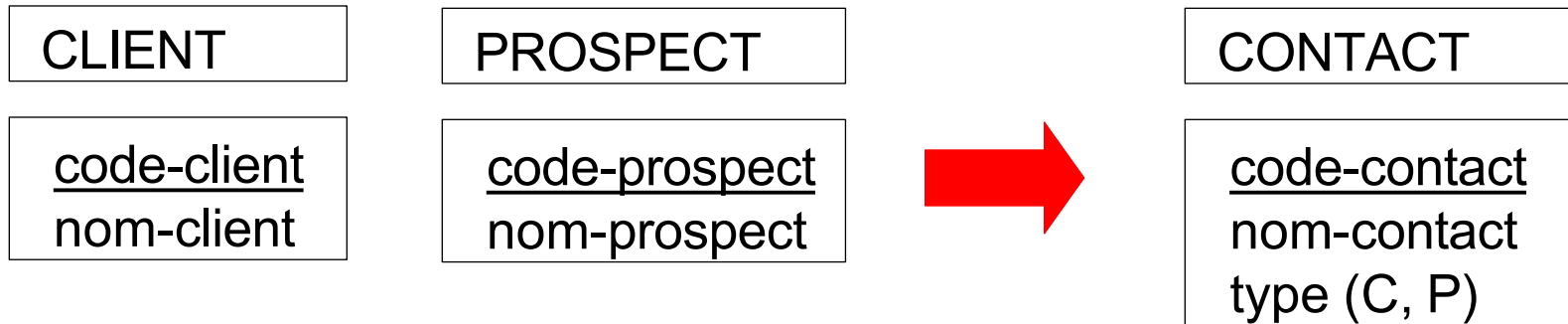
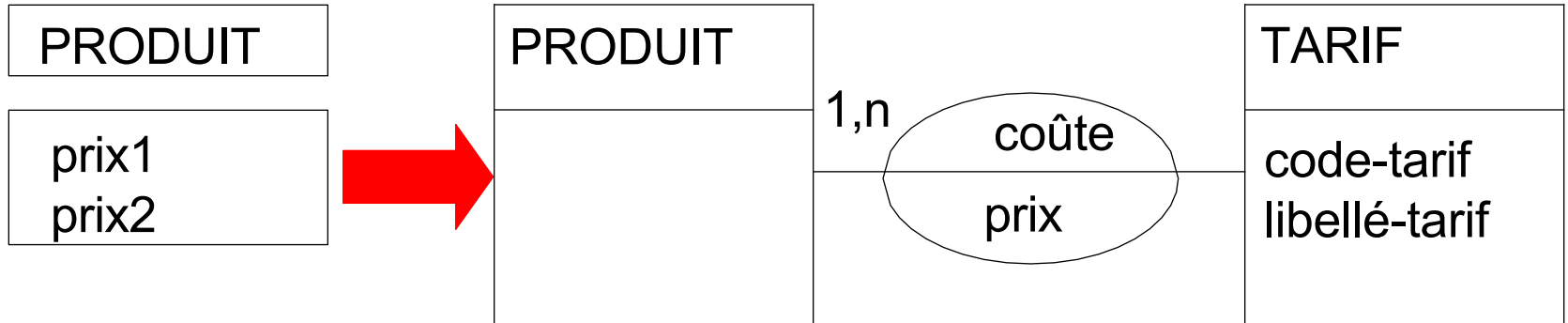
- Une entité a une **existence propre** et un **identifiant**.
- Une association n'existe que si ses extrémités existent et n'a pas d'identifiant explicite.
- Une entité peut être associée à d'autres entités, une association non.



Pour une situation donnée, **il n'existe pas une «solution» unique.**

Un modèle **exprime un point de vue** et reflète **des besoins en information.**

Le bon modèle est celui qui est accepté par les personnes concernées par le projet.



Pas d'héritage dans le modèle E/A de base !

- Toutes les propriétés identifiées doivent apparaître dans le modèle.

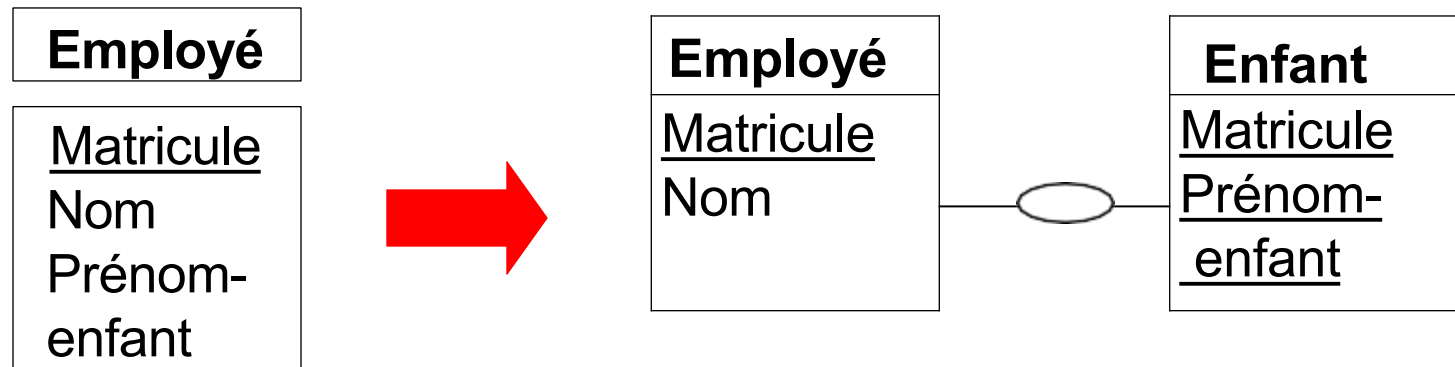
2. Règles sur les entités

a. Règle de l'identifiant

Toutes les entités ont un identifiant.

b. Règle de vérification des entités

Pour une occurrence d'une entité, chaque propriété ne prend qu'une seule valeur (cf. la 1FN du modèle relationnel); MONO-VALUEE

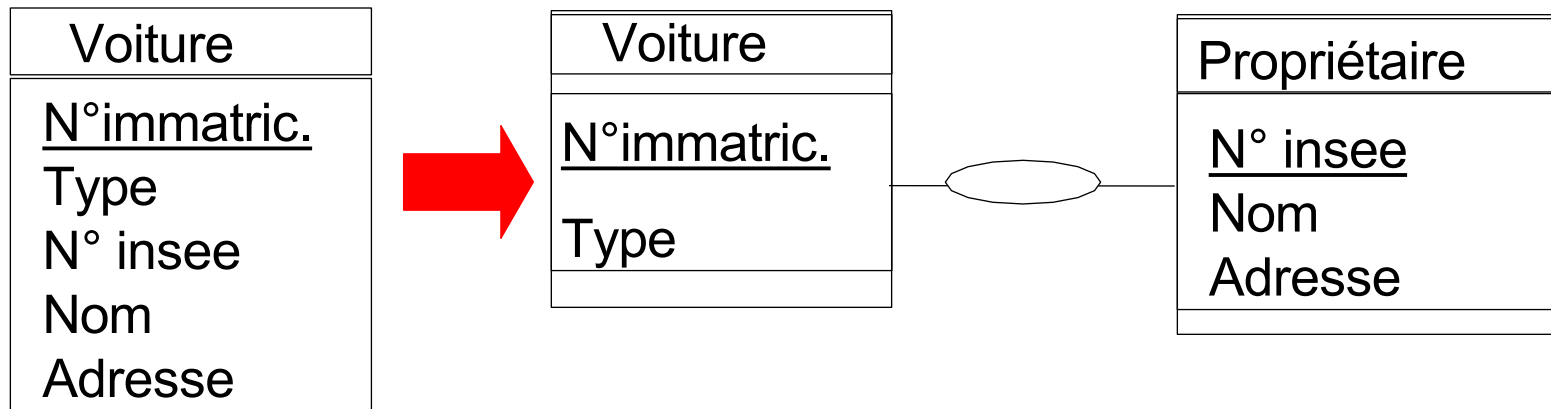


On décompose l'entité Employé en deux entités : Employé, et Enfant

2.c Règles de normalisation des entités

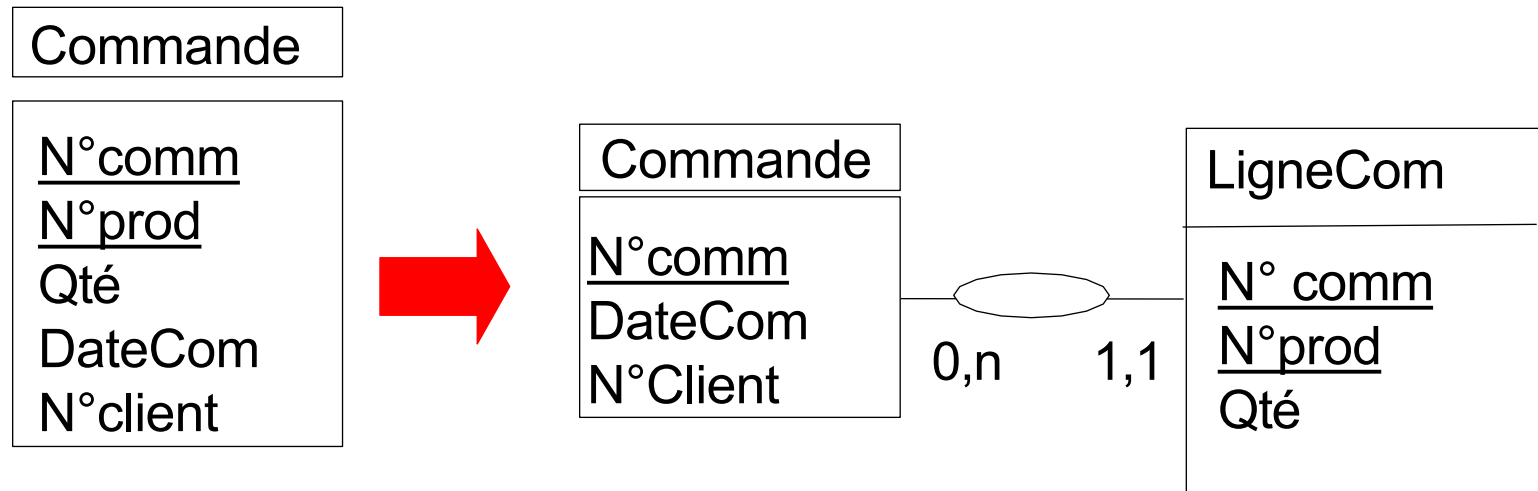
a) Les dépendances fonctionnelles (DF) entre les propriétés d'une entité doivent vérifier la règle suivante : **toutes** les propriétés de l'entité dépendent fonctionnellement de l'identifiant **et uniquement** de l'identifiant.

Rappel : \exists une DF $X \rightarrow Y$ si à une valeur de X correspond une et une seule valeur de Y (réciproque pas vraie).



La DF: $N^{\circ}insee \rightarrow Nom, Adresse$ contredit la règle.

b) Une partie de l'identifiant ne peut pas déterminer certaines propriétés.



La DF $n^\circ\text{-comm} \rightarrow \text{date-comm}, n^\circ\text{-client}$ contredit la règle. On décompose l'entité Commande en deux entités.

Ces règles correspondent aux 2FN et 3FN du modèle Relationnel (dépendance pleine et directe des clés).

Normaliser le MCD

Première forme normale (1NF): Chaque cellule d'une table doit contenir une valeur unique et cette valeur doit être élémentaire.

Deuxième forme normale (2NF): Chaque attribut de la table doit dépendre de la clé primaire au complet et non d'une partie de la clé.

Troisième forme normale (3NF): Aucun attribut non-clé doit dépendre d'un autre attribut non-clé.

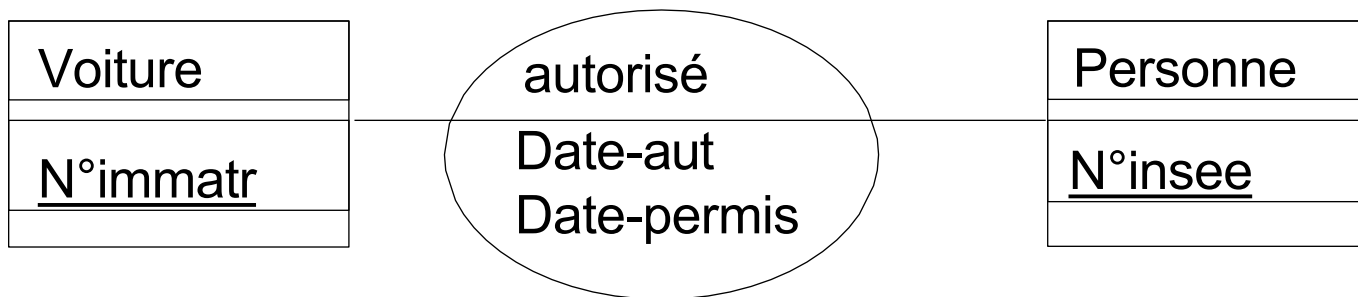
3. Règles sur les associations

a. Règle de vérification des associations

Pour une occurrence d'association, chaque propriété ne prend qu'une seule valeur.

b. Règle de normalisation sur les propriétés des associations

Toutes les propriétés de l'association doivent dépendre fonctionnellement de tous les identifiants des entités portant l'association, et uniquement d'eux.

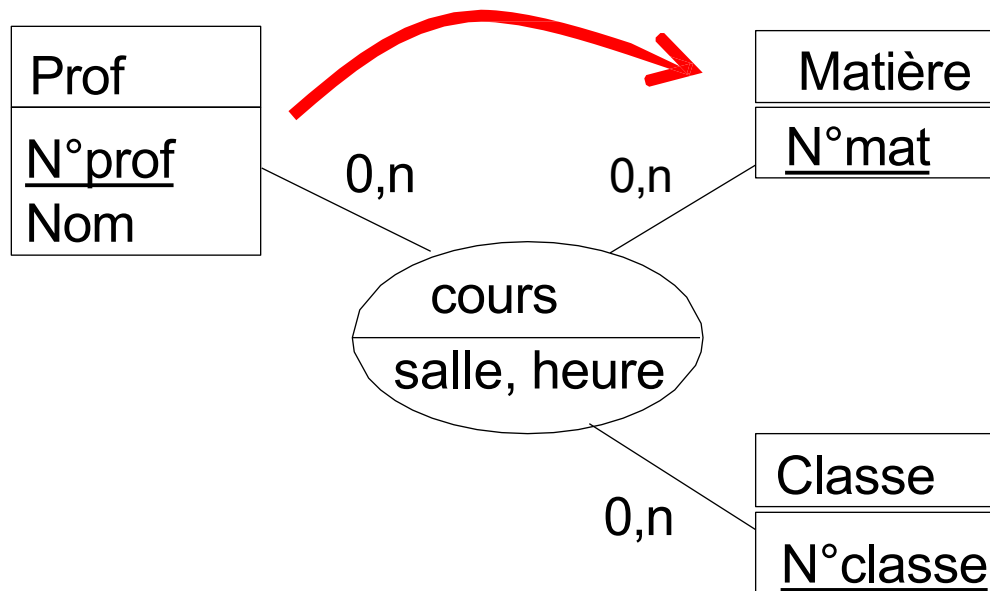


N°-insee → *Date-permis* pose problème (donc déplacer *Date-permis* vers *Personne*)

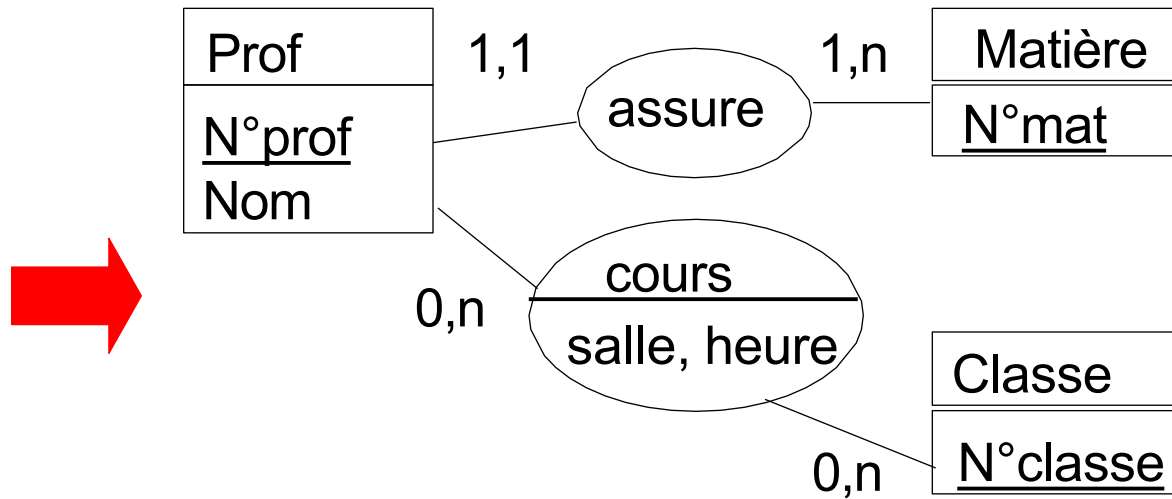
3.c La décomposition des associations n-aires

Il faut garder un minimum d'associations d'arité > 2 .

Si on observe une DF entre un sous-ensemble des entités d'une association, on peut la décomposer en deux associations (on parle aussi de 'contrainte d'intégrité fonctionnelle' ou CIF).

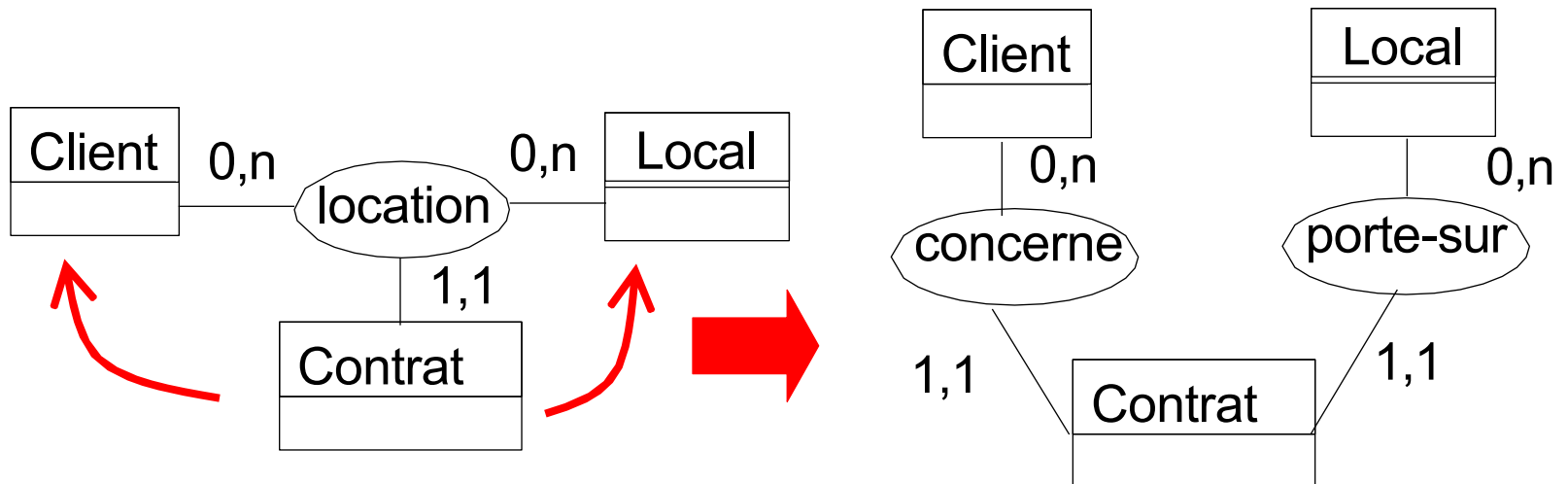


Une éventuelle DF $N°prof \rightarrow N°mat$ donne la décomposition :



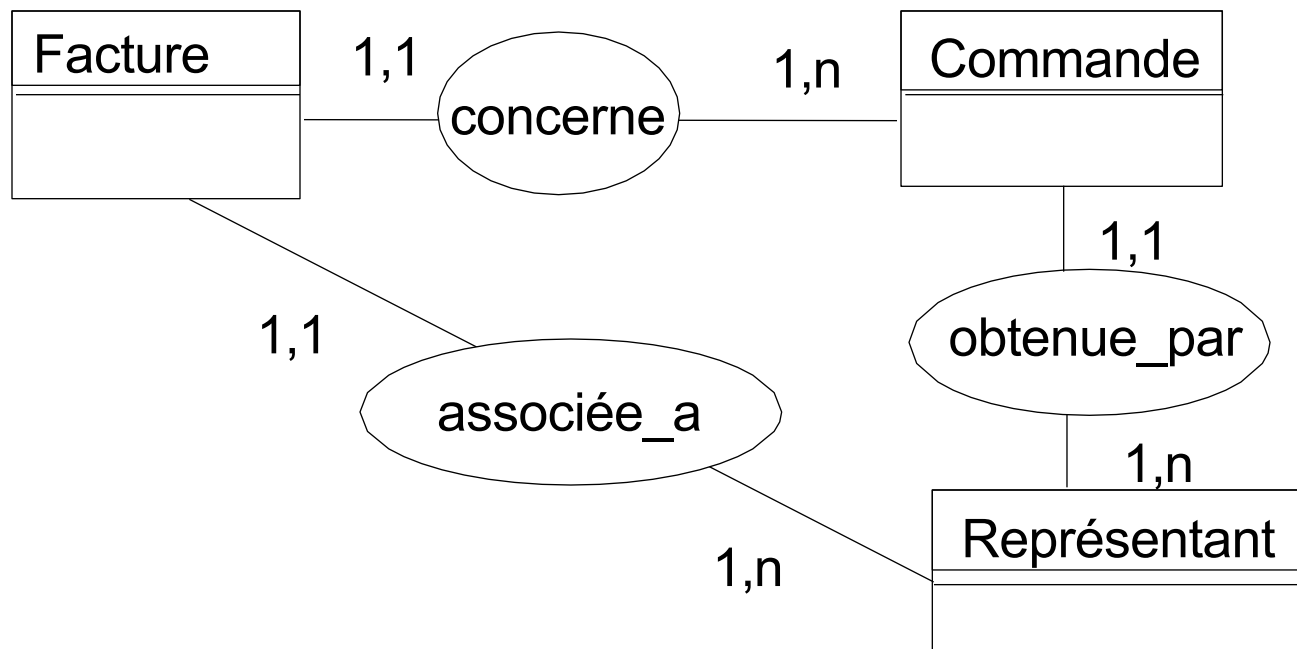
C'est le cas, quand une patte a une cardinalité 1,1.

Par exemple à 1 contrat est associé un client et un local :



3.d La suppression des associations transitives

Toute association pouvant être obtenue par transitivité de n autres associations peut être supprimée.



On supprime l'association *associée_a*, car elle peut être obtenue par transitivité sur les associations *concerne* et *obtenue_par*

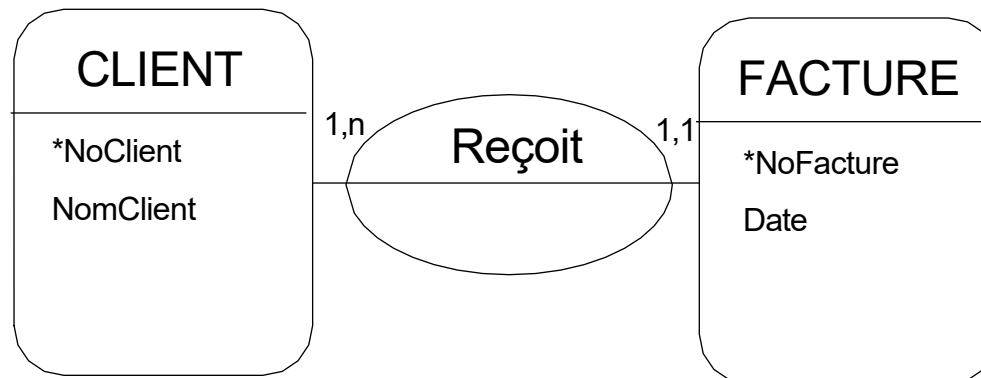
Règle 1

- Deux entités qui doivent être reliées entre elles le seront par le biais d'une relation

À ne pas
faire



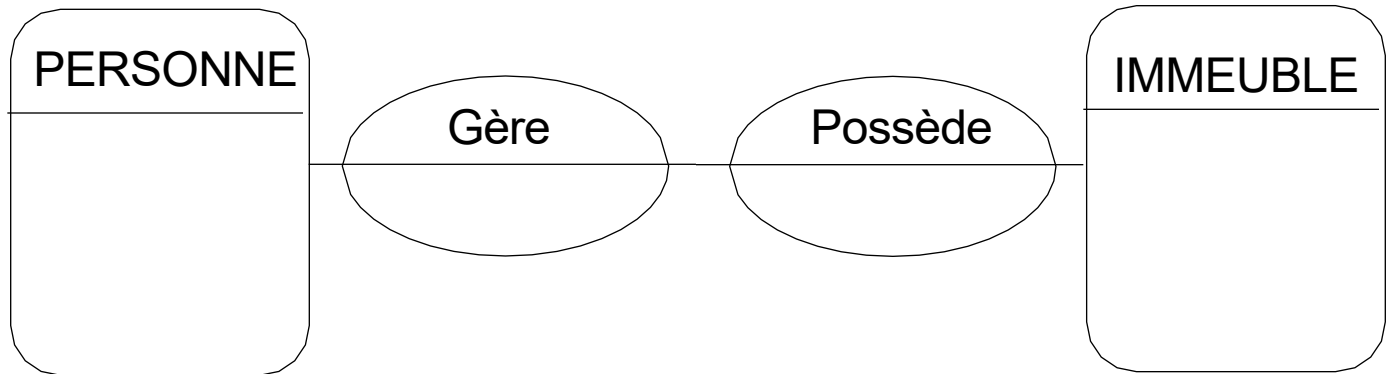
OK



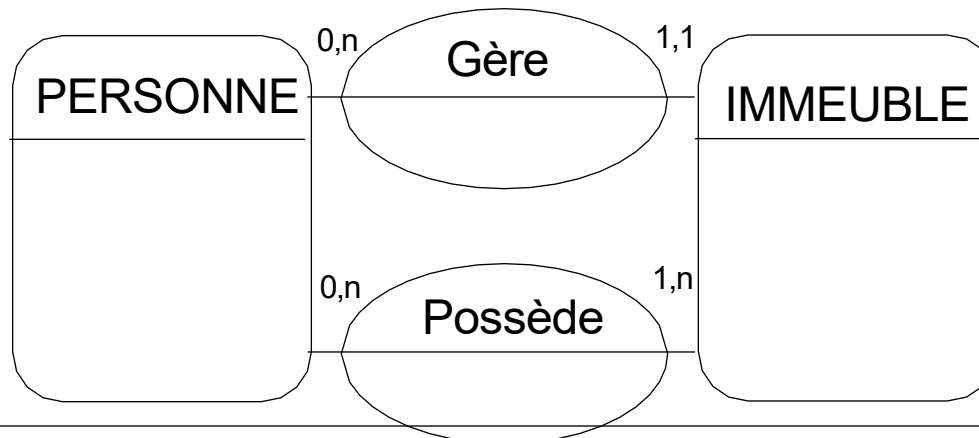
Règle 2

- Deux relations ne peuvent jamais être directement reliées entre elles

À ne pas faire



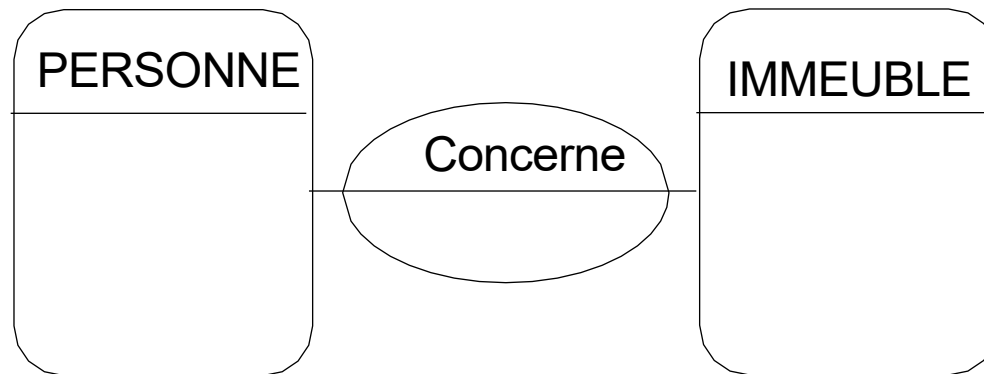
OK



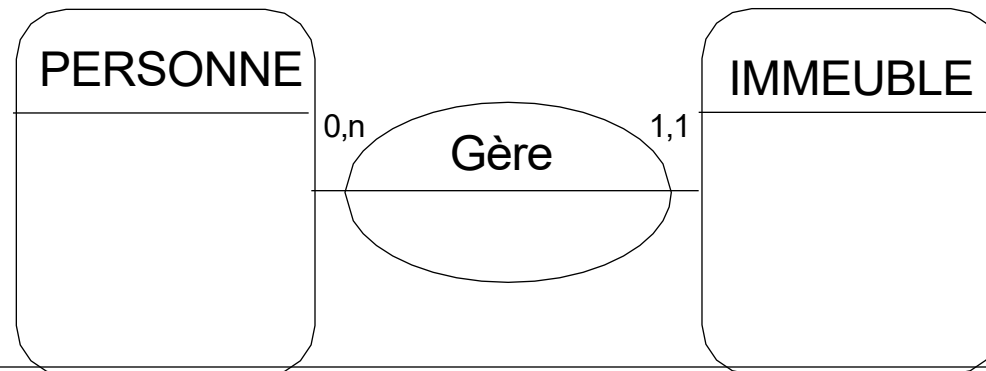
Règle 3

- Le nom de la relation doit représenter d'une manière concrète et significative l'information que l'on veut obtenir

À ne pas
faire



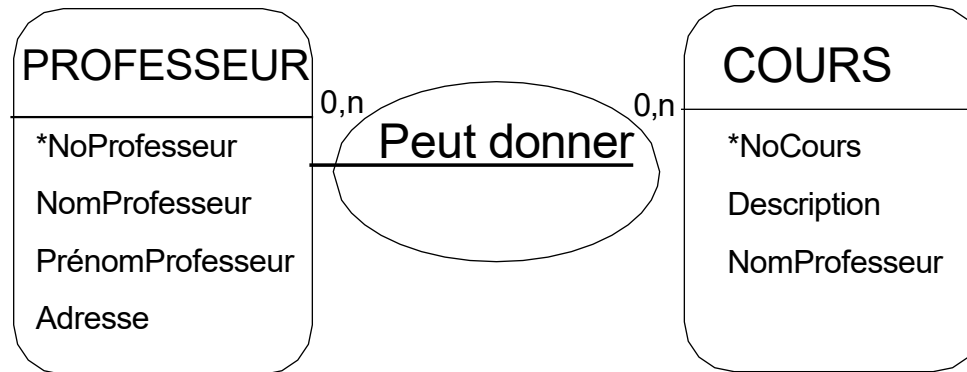
OK



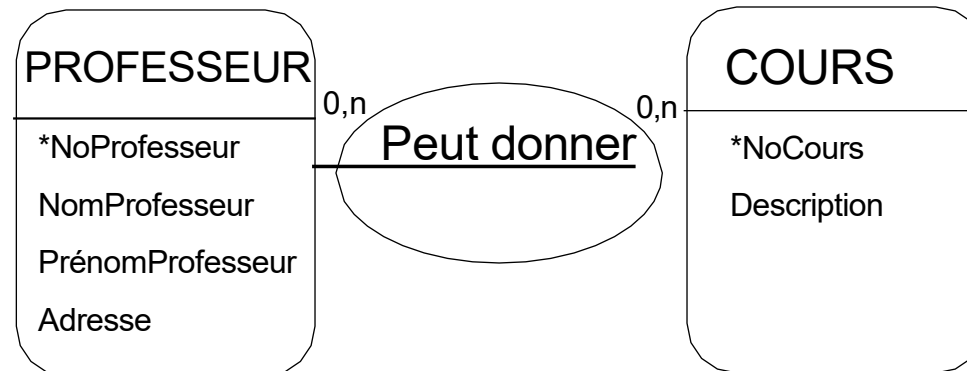
Règle 4

- Un attribut est unique à une entité ou à une relation

À ne pas
faire



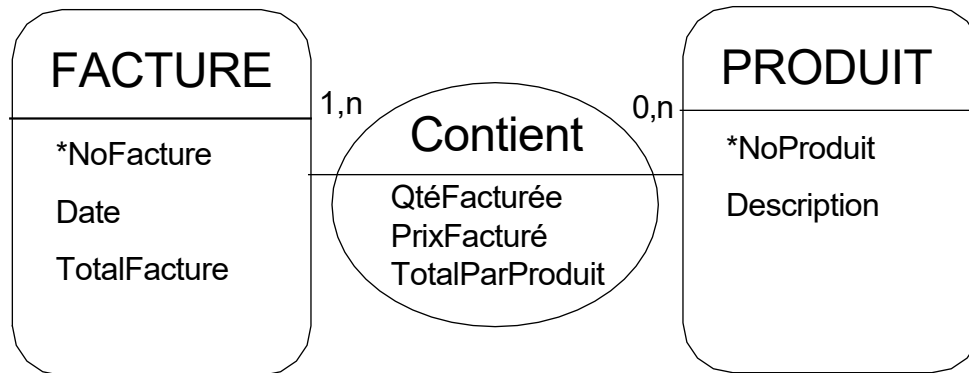
OK



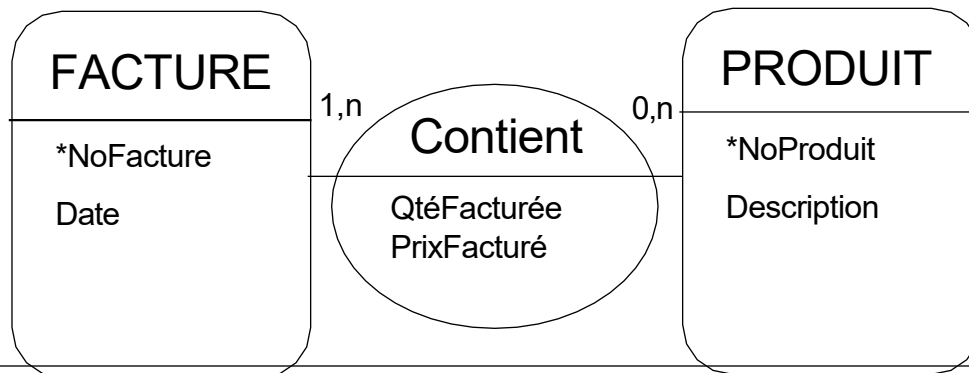
Règle 5

- Les entités et les relations ne doivent contenir que des données élémentaires, donc ne pas contenir des résultats de calcul/traitement

À ne pas
faire



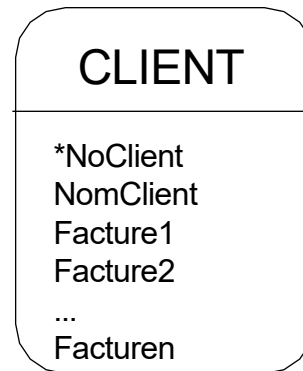
OK



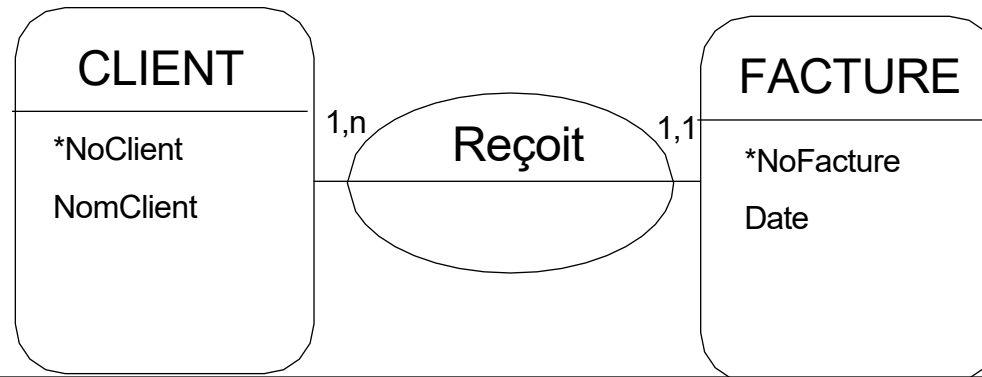
Règle 6

- Pour une occurrence donnée, une seule valeur doit être attribuée à chaque attribut de l'entité ou de la relation

À ne pas
faire



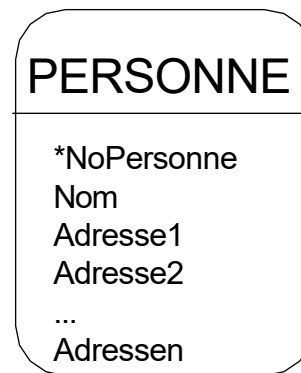
OK



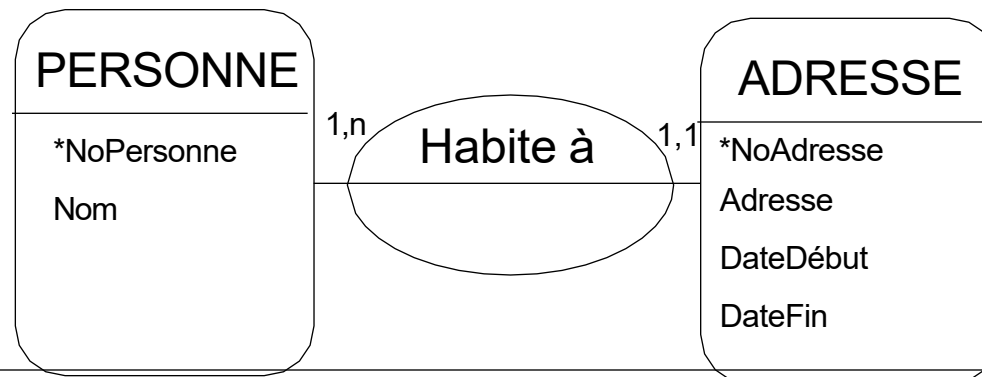
Règle 7

- Pour conserver l'historique d'une donnée d'une entité, on forme une nouvelle entité avec cette donnée et on ajoute une période d'application

À ne pas
faire



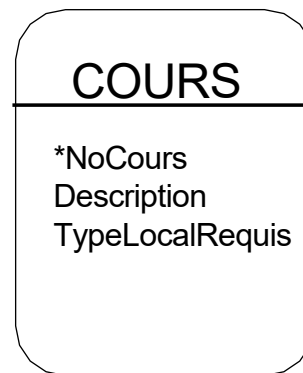
OK



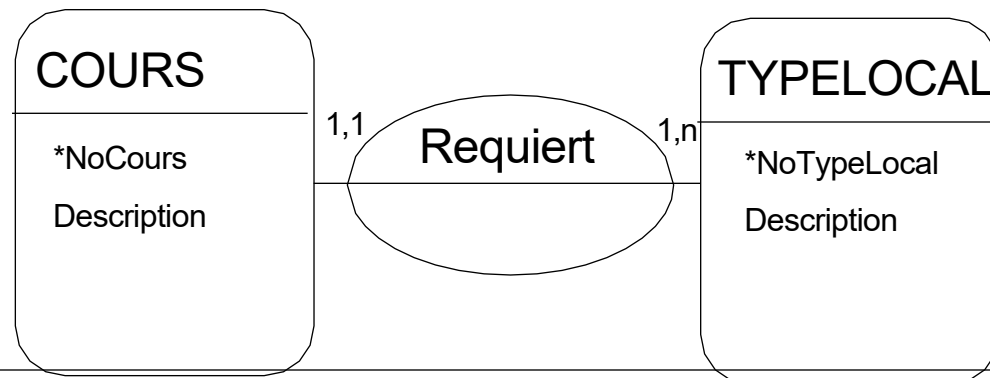
Règle 8

- Chaque fois qu'un attribut est un code ou un type, on forme une nouvelle entité avec ce code/type et sa description

À ne pas
faire



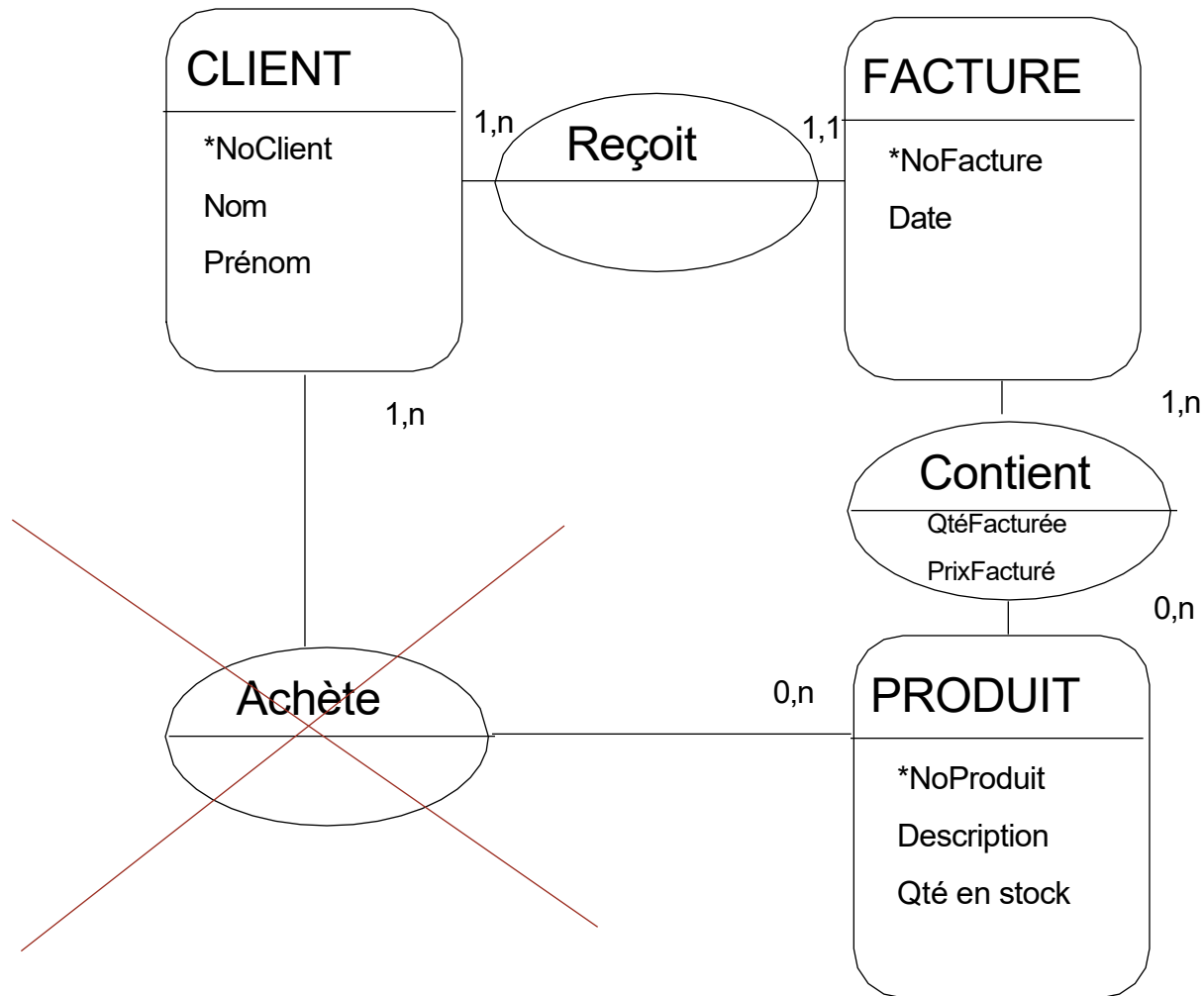
OK



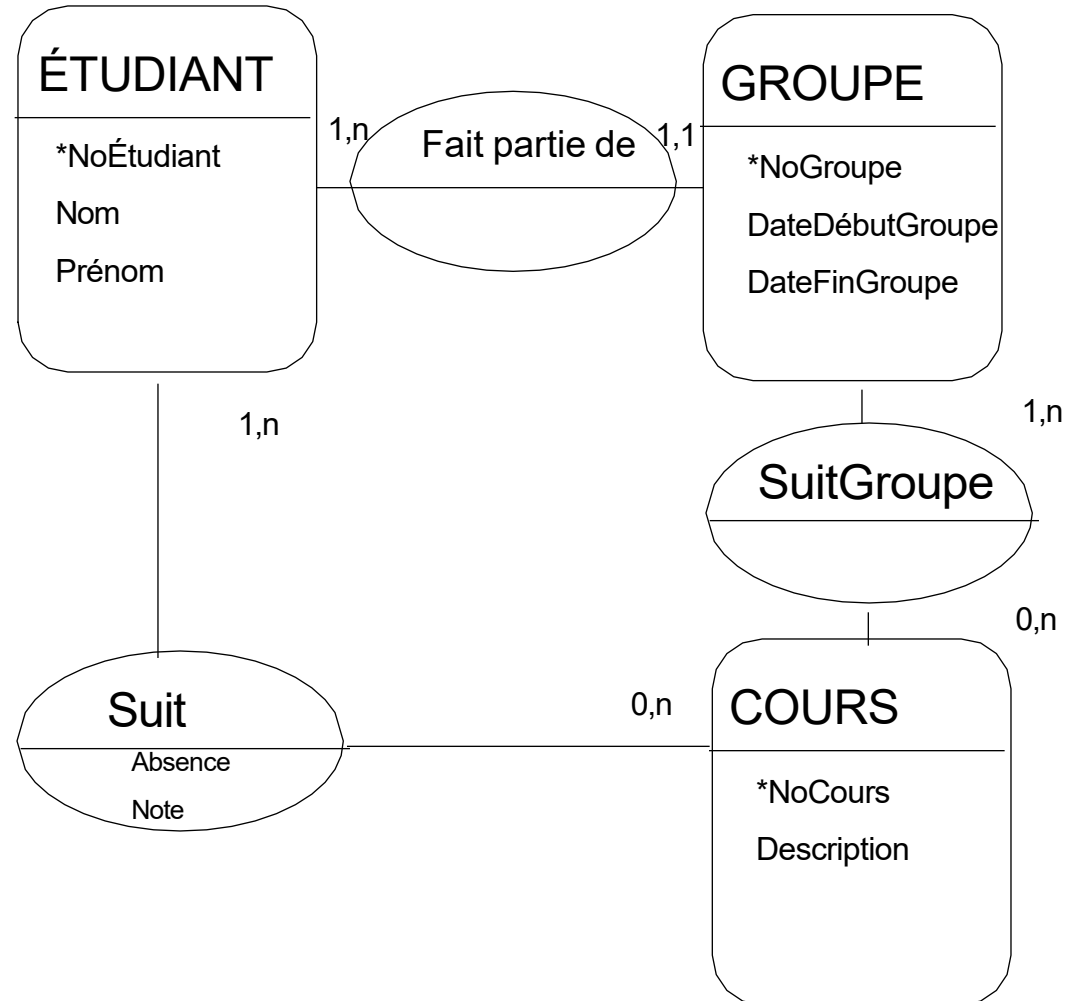
Règle 9

- Lorsqu'une relation peut être déduite des autres relations, elle n'est pas représentée à moins qu'on veuille extraire une information spécifique à cette relation

Règle 9



Règle 9



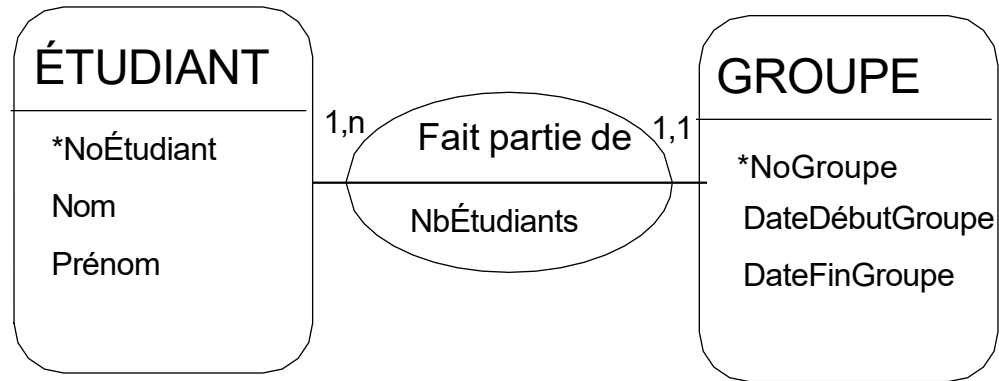
OK

Règle 10

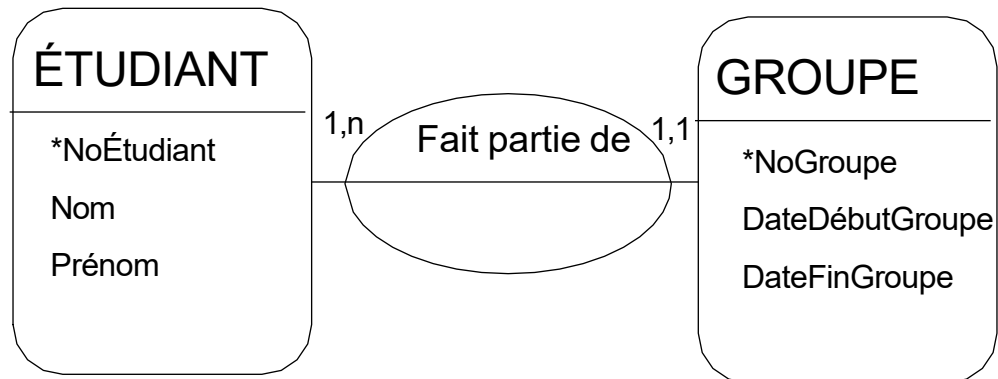
- Seule une association de type plusieurs à plusieurs (N:M) peut avoir des attributs
- Si vous avez des attributs sur une relation 1:N, il y a un problème !
 - L'attribut doit être placée sur une entité
 - L'attribut doit être éliminé (ex. valeur calculée)
- Note : Une relation N:M n'a pas obligatoirement des attributs

Règle 10

À ne pas
faire



OK



Une démarche de construction ?

Certains auteurs suggèrent la démarche suivante :

- 1 Analyser l'existant et constituer le dictionnaire des données
- 2 Épurer les données (éliminer synonymes et polysèmes)
- 3 Dégager les 'entités naturelles' grâce aux identifiants existants déjà dans l'organisation
- 4 Rattacher les propriétés aux entités
- 5 Recenser les associations entre entités et leur rattacher leurs éventuelles propriétés
- 6 Déterminer les cardinalités
- 7 Décomposer si possible les associations n-aires (cf. règles)
- 8 S'assurer de la conformité du modèle aux règles de construction (cf. règles)
- 9 Normaliser le modèle : s'assurer qu'il est en 3FN

Malheureusement, dans le monde réel, il n'y a pas d'énoncé ! L'existant n'est pas complètement connu au départ, ni toutes les données. Imaginer avoir un **dictionnaire exhaustif au départ** n'est pas réaliste dans les cas complexes.

Il n'y a donc pas une suite linéaire d'étapes mais plutôt un ensemble d'itérations :

- **ébaucher un modèle avec les entités et associations qui semblent essentielles,**
- **évaluer si ce qui est modélisé est correct et correspond à ce que les utilisateurs comprennent,**
- **itérer en complétant progressivement jusqu'à ce que le modèle semble raisonnablement complet.**

Exercice 1 : Bibliothèque

Une bibliothèque gère les informations suivantes :

- **Livres** : Chaque livre a un identifiant unique, un titre, un auteur, et une année de publication.
- **Adhérents** : Chaque adhérent a un numéro unique, un nom, un prénom, et une adresse.
- Un adhérent peut emprunter plusieurs livres. Chaque emprunt est identifié par une date d'emprunt et une date de retour prévue.

Etablir le MCD.

Exercice 2 : Gestion de cours

Une école gère les informations suivantes :

- **Étudiants** : Chaque étudiant a un matricule unique, un nom, et une date de naissance.
- **Cours** : Chaque cours a un code unique, un intitulé, et un nombre d'heures.
- Un étudiant peut s'inscrire à plusieurs cours. Pour chaque inscription, on enregistre la date d'inscription et la note finale.

Etablir le MCD.

Exercice 3 : Gestion d'une université

Une université suit les informations suivantes :

- **Étudiants** : Chaque étudiant a un matricule unique, un nom, un prénom, et une date de naissance.
- **Enseignants** : Chaque enseignant a un identifiant unique, un nom, un prénom, et un département d'affectation.
- **Département** : Chaque département a un identifiant unique, un nom, et un responsable (enseignant).
- **Cours** : Chaque cours a un code unique, un titre, et un volume horaire.
- Un enseignant peut enseigner plusieurs cours, mais un cours est enseigné par un seul enseignant.
- Un étudiant peut s'inscrire à plusieurs cours, et chaque inscription est caractérisée par une date d'inscription et une note.
- Un département a plusieurs enseignants, mais un enseignant appartient à un seul département.
- Un département peut proposer plusieurs cours, mais un cours appartient à un seul département.

Etablir le MCD

Exercice 4 : Gestion d'un Magasin

Un magasin gère les informations suivantes :

- **Produits** : Chaque produit a un identifiant unique, un nom, une description, et un prix.
- **Fournisseurs** : Chaque fournisseur a un identifiant unique, un nom, une adresse, et un numéro de téléphone.
- **Commandes** : Chaque commande est passée par un fournisseur et est caractérisée par un numéro unique, une date, et un total.
- Chaque commande peut contenir plusieurs produits, et pour chaque produit, on enregistre la quantité commandée.

Etablir le MCD

Exercice 5:

Un historien souhaite établir des statistiques sur des soldats de la Première Guerre mondiale.

Pour chaque soldat, outre l'état-civil, il souhaite avoir la trace :

- de la date de son décès si celui-ci est survenu suite aux combats
- des blessures reçues (type et date de la blessure, en plus de la bataille où elle a été infligée. Les batailles seront référencées dans une liste comportant le lieu, les dates de début et de fin)
- des grades obtenus (avec les dates)
- de l'unité de rattachement (avec les dates)

Question : Etablir le MCD