

**MERISE**  
Modélisation de Systèmes d'Information

**Mohamed Saad Bajjou**  
Sup Management

**Organisation sociale** : circulation d'informations

**Le Système d'Information(S.I.)** d'une organisation c'est l'ensemble des éléments chargés de stocker et de traiter les informations (ordinateurs, postes de travail, règles et méthodes...)

*Exemple d'une organisation sociale :*

Une entreprise

## Base de Données :

Une base de données est une collection d'informations inter reliées et stockées par un ordinateur selon une structure définie avec aussi peu de redondance que possible pour servir une ou plusieurs applications de façon optimale.

**Notation : BD**

## Système de Gestion de Base de données :

Un système de Gestion de Base de Données est un ensemble d'outils logiciels Qui permet la manipulation de **BD**.

**Notation : SGBD**

Exemple de SGBD : Oracle, O2, Posgres, MySQL, Access, DB2.....



## *Position du problème*

- L '**objectif** d'une étude : **améliorer** le traitement de l'information dans une organisation
- Cette amélioration s'exprime en terme de **Collecte, Saisie, Transmission, Stockage, Traitement...** De l'information
- L'étude de fonctionnement d'une organisation peut aboutir à la conclusion que son ***informatisation n'est pas souhaitable***

## *Position du problème*

- **Informatisation réussie** = le SI doit intégrer le schéma global de circulation des informations
- L'étude doit mettre en évidence :
  - les **éventuelles incohérences** du système de traitement de l'information (redondance, inutilité d'une information, ...)
  - les **améliorations** à apporter à l'archivage, à l'accès à une information particulière, ...

# Position du problème

## Le besoin de méthodes

La conception d'un SI n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place.

La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

Ce type de méthode est appelée *analyse*. Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée étant la méthode MERISE.

## Qu'est ce que Merise ?

- Approche ancienne : 1978
- Très répandue en France
- Origine française : développée par :
  - CTI (Centre Technique d'Informatique)
  - CETE (Centre d'Etudes Techniques de l'Équipement)
- Remise à jour : Merise 2
  - à la mode «Objet» : concepts de classes, héritage...

- Merise : Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise
  - Merise = Méthode **d'analyse** : permet d'aboutir **séparément** aux modèles conceptuels :
    - des **données** (MCD) = Modèle **Entité-Relations** (ou Modèle **Entité-Association**)  
vue **statique**
    - des **traitements** (MCT)  
vue **dynamique**
- **MCD** et **MCT** indépendants de toute implémentation

## LES PHASES D'UNE ETUDE MERISE

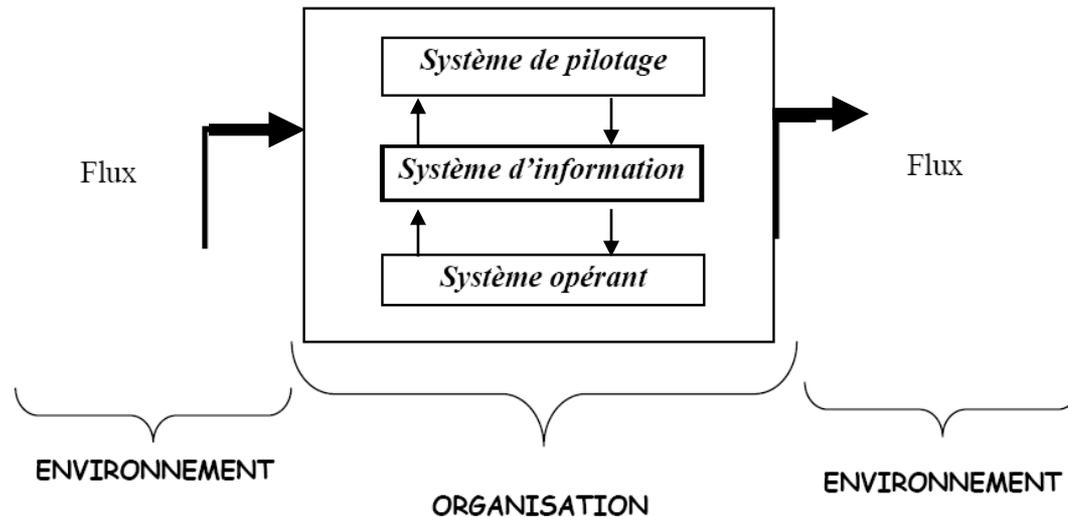
Enchaînement Des Modèles Dans La Démarche Par Etapes

Enchaînement Des Modèles Dans La Démarche Par Niveaux

# Généralités sur les Systèmes d'Informations

° **SYSTEME** : C'est un ensemble d'éléments en interaction visant à atteindre un certain objectif

° **SYSTEME D'INFORMATION (SI)** : C'est un système de couplage entre le système opérant et le système de pilotage



# Généralités sur les Systèmes d'Informations

## ◦ Systeme De Pilotage :

C'est l'endroit où les décisions sont prises en matière :

- de fixations d'objectifs
- d'affectations des ressources allouées au Système Opérant
- de contrôle des résultats obtenus
- de corrections si nécessaire

## ◦ Systeme Opérant :

C'est le système où s'effectuent les transformations physiques, ou Intellectuelles sur les flux qui traversent l'entreprise en vue de produire des sorties valorisées

# Généralités sur les Systèmes d'Informations

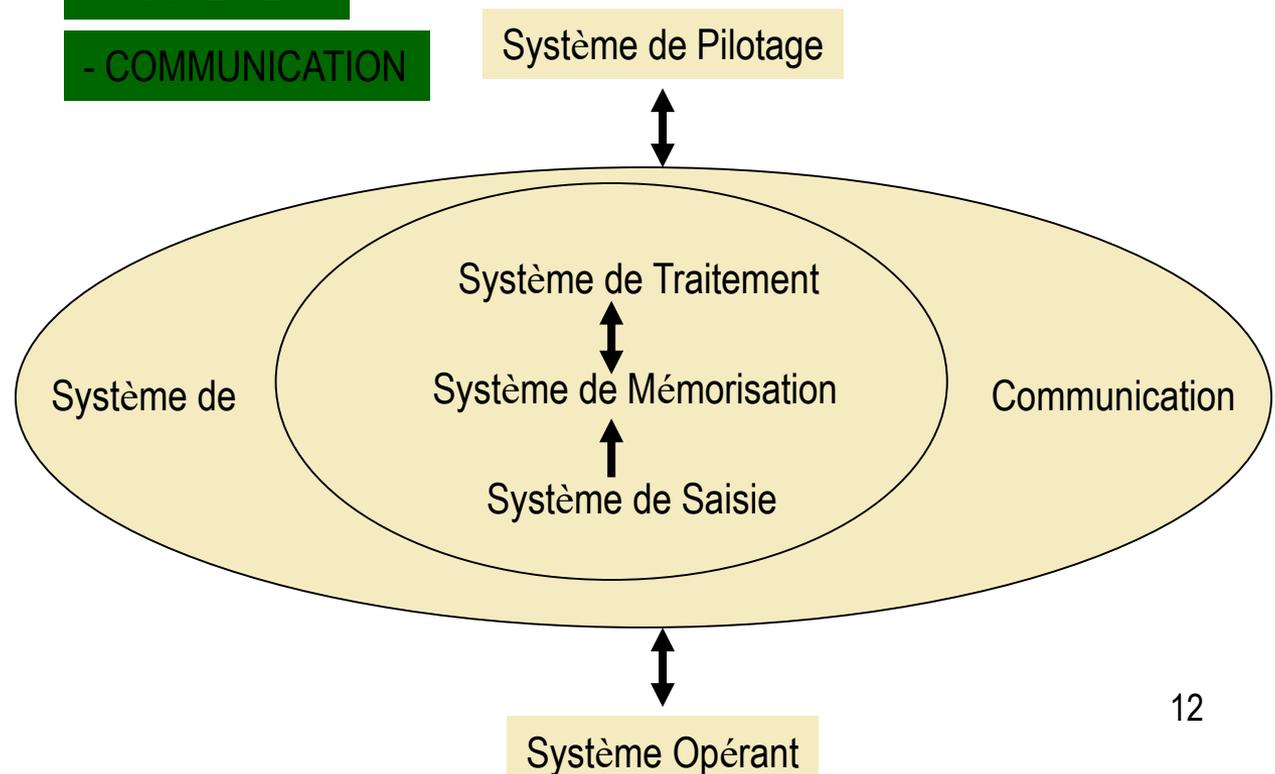
## Fonctions d'un Systèmes d'Informations

- SAISIE

- MEMORISATION

- TRAITEMENT

- COMMUNICATION



# Généralités sur les Systèmes d'Informations

## - SAISIE

- Que saisir ?
- Codification
- Fiabilité, validité, pertinence

## - MEMORISATION

- Nécessité d'un stockage, d'un support
- Obligation d'une structuration

## - TRAITEMENT

- Accès à l'information
- Mise en forme
- Manipulation

## - COMMUNICATION

- Interne
  - système opérant vers pilotage et réciproquement
  - entre divers niveaux verticaux et latéraux
- Externe
  - de et vers l'environnement

# L'approche systémique

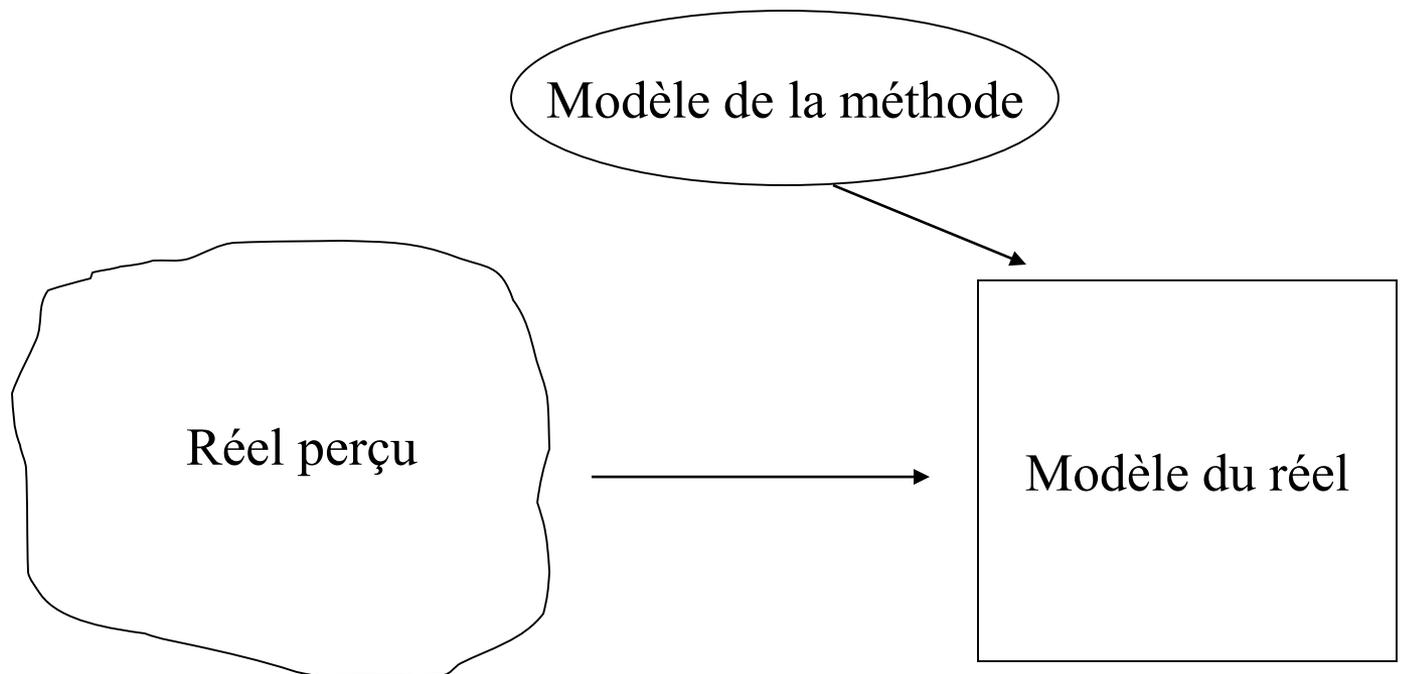
*Merise est une méthode globale de troisième génération dont l'approche est systémique.*

## **Méthodes systémiques**

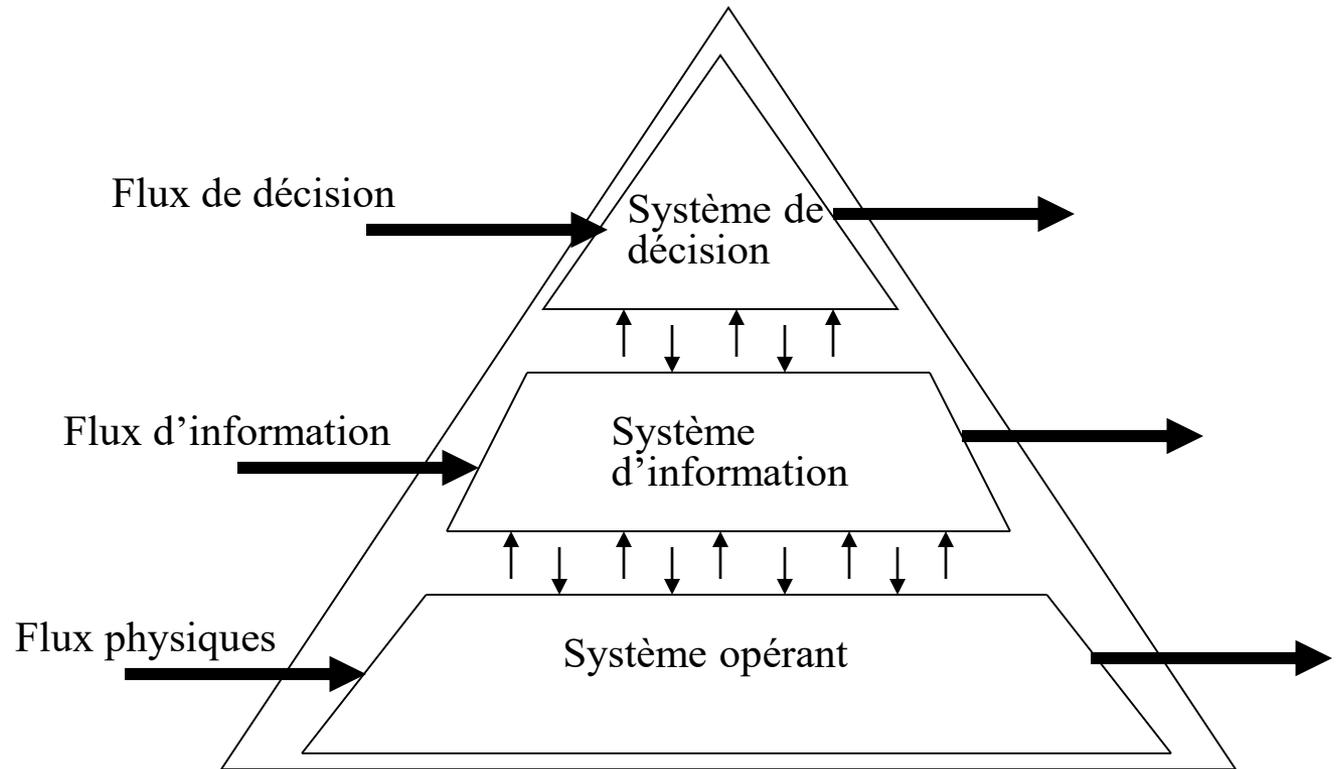
- Passent par la modélisation du système étudié pour mieux le comprendre.
- Prennent en compte les interactions entre les éléments constitutifs et leur environnement extérieur,
- A la décomposition du SI en sous-système

## L'approche modélisatrice

Le SI est considéré comme une représentation abstraite du S.O., qui doit être complète et cohérente.

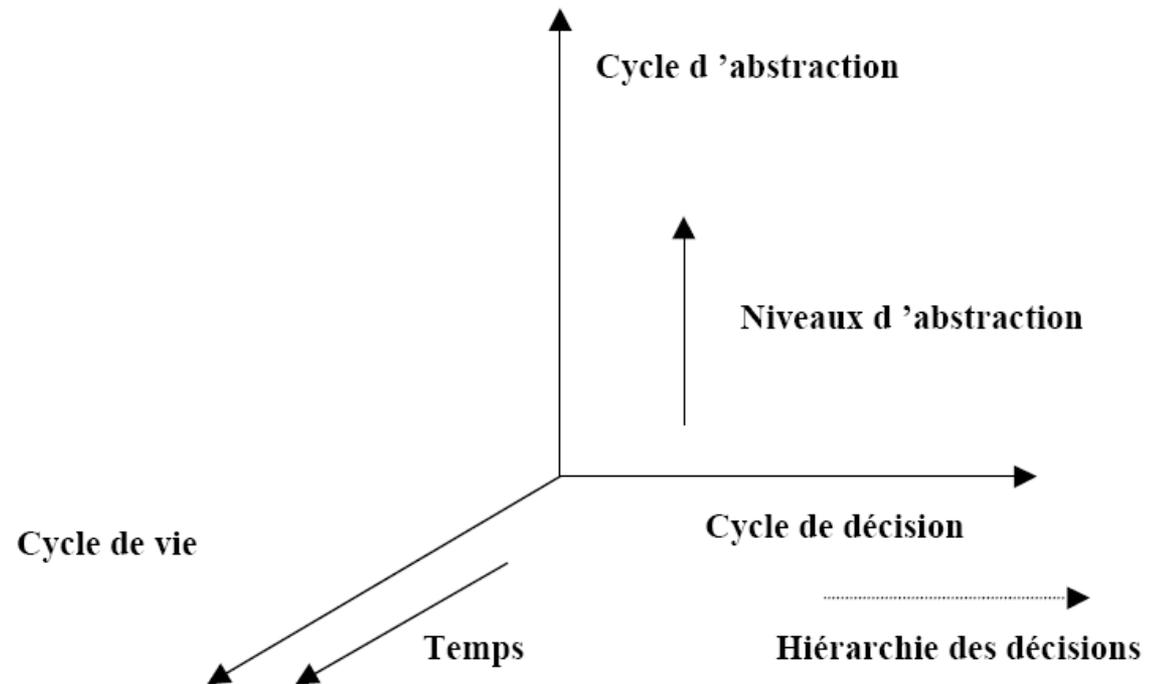


# L'entreprise et ses systèmes

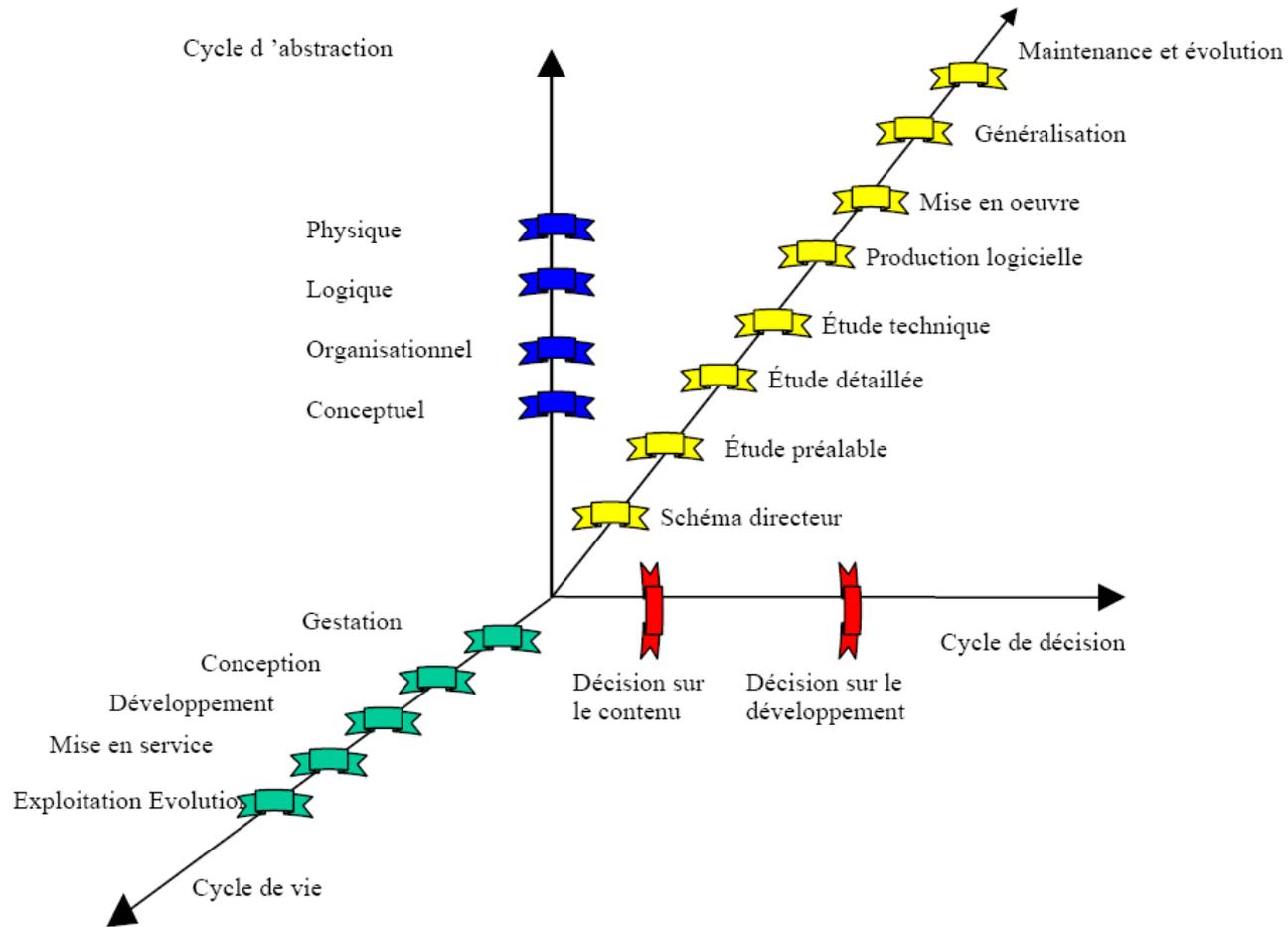


## *Cycles de Merise*

Trois cycles concourent à l'étude d'un système d'information et permettent d'en situer les étapes



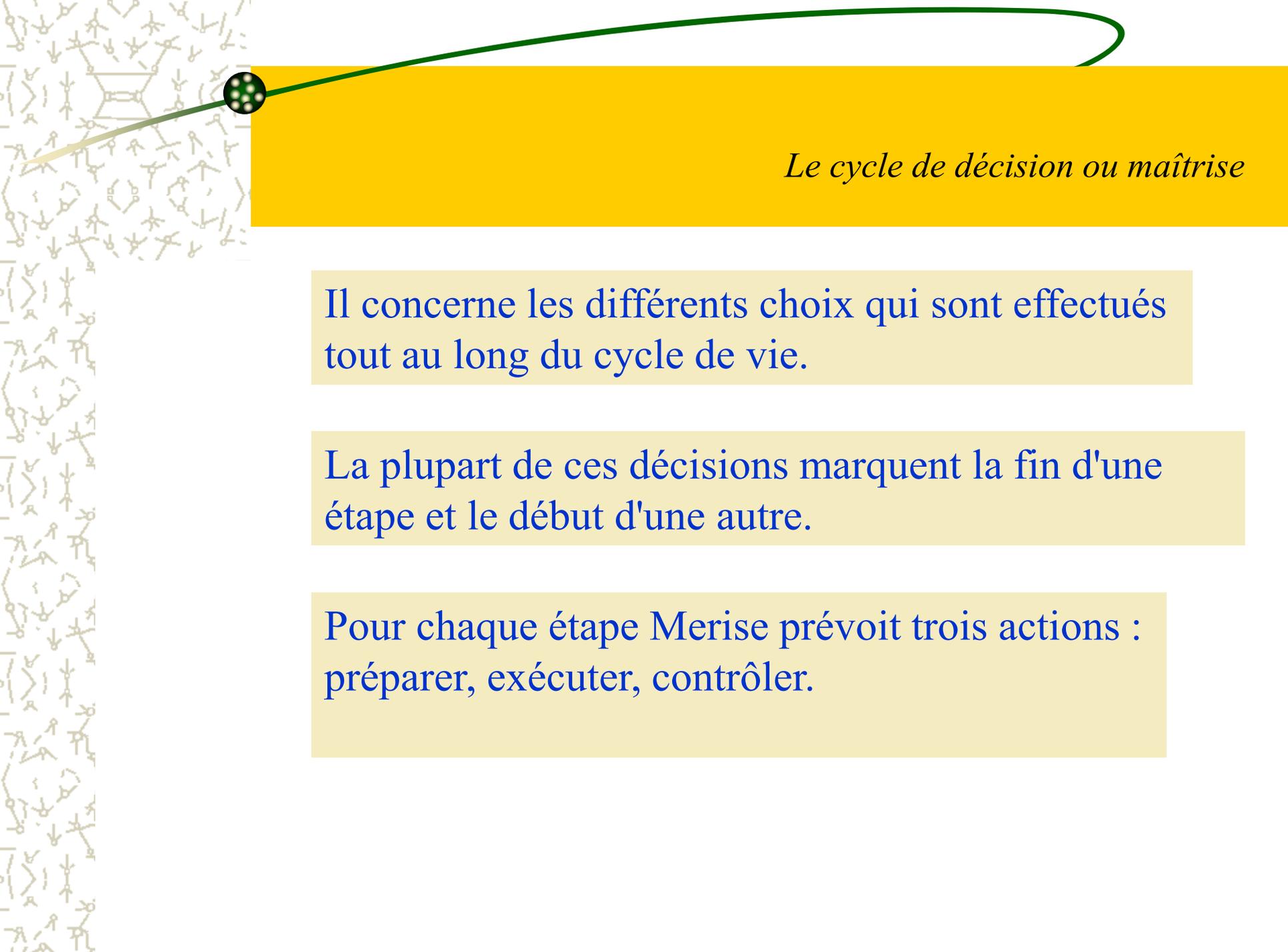
# Cycles de Merise



## *Le cycle de vie ou démarche*

**Il permet de décrire la vie du système d'information.  
MERISE distingue trois périodes :**

- **la conception** du système d'information qui aboutit à la conception détaillée des spécifications fonctionnelles et techniques, (Gestation + Conception)
- **la réalisation** qui consiste à produire des programmes et des consignes d'utilisation correspondant aux spécifications détaillées, (Développement, Mise en oeuvre)
- **la maintenance** du système d'information qui a pour objectif l'adaptation aux évolutions de son environnement. (Exploitation + Evolution)



## *Le cycle de décision ou maîtrise*

Il concerne les différents choix qui sont effectués tout au long du cycle de vie.

La plupart de ces décisions marquent la fin d'une étape et le début d'une autre.

Pour chaque étape Merise prévoit trois actions : préparer, exécuter, contrôler.

## *Le cycle d'abstraction ou raisonnement*

Il offre les concepts nécessaires à la description du monde réel dans le système d'information.

On y trouve les quatre (ou trois) niveaux d'abstraction de MERISE (conceptuel, organisationnel, logique et physique).

Le niveau conceptuel		Système d'information
Le niveau organisationnel	Système d'information organisationnel	
Le niveau logique	Système d'information informatisé	
Le niveau physique ou opérationnel		

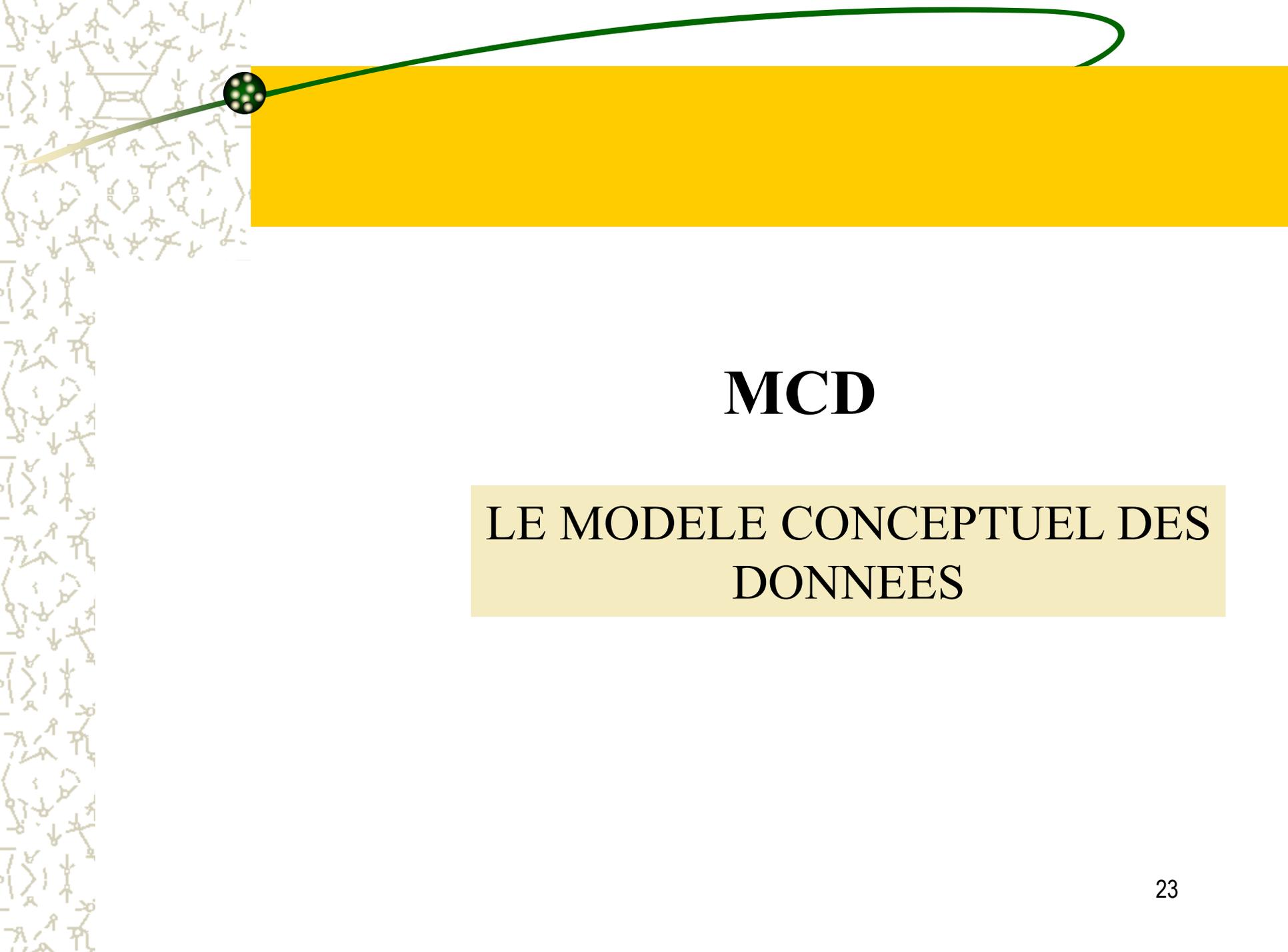
- Le niveau **conceptuel** exprime les choix fondamentaux de gestion indépendamment des moyens à mettre en oeuvre (Quoi?)

-le niveau **organisationnel** exprime les choix d'organisation des ressources humaines, matérielles et financières. (Qui? Où? Quand?)

- le niveau **logique** exprime les choix de moyens et ressources informatiques sans se soucier de leurs caractéristiques techniques précises alors que le niveau **physique** prend en compte ces spécificités. (Comment?)

## Les modèles Merise

NIVEAU	Données	Traitements	Questions	Contenu	
CONCEPTUEL	MCD	MCT	Quoi	<ul style="list-style-type: none"> <li>- Informations manipulées</li> <li>- Règles de gestion</li> <li>- Enchaînement des activités</li> </ul>	S I
ORGANISATIONNEL	MOD ↓	MOT	Qui Quand Où	<ul style="list-style-type: none"> <li>- Partage des tâches homme/machine</li> <li>- Temps réel ou différé</li> <li>- Traitement unitaire ou par lots</li> <li>- Répartition géographique des traitements</li> <li>- Organisation des données retenues</li> <li>- postes de travail</li> </ul>	S I O
LOGIQUE	MLD	MLT	Comment	<ul style="list-style-type: none"> <li>- Classe de mise en oeuvre des données</li> <li>- Découpage en modules et transactions</li> </ul>	S I I
PHYSIQUE ou OPERATIONNEL	MPD	↓ MOPT		<ul style="list-style-type: none"> <li>- Enregistrements</li> <li>- Ecrans, états</li> <li>- Programmes ou unités de traitement</li> <li>- Matériels</li> <li>- Réseau</li> <li>- Logiciels</li> </ul>	



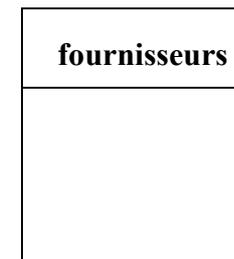
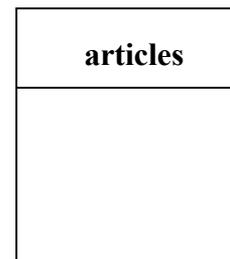
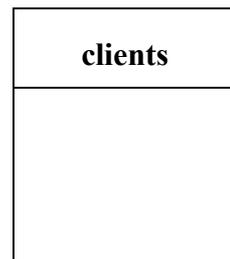
# MCD

## LE MODELE CONCEPTUEL DES DONNEES

## Elaboration de schéma très répondu : Schéma **Entités-Associations**

**Entité** : c'est une population d'individus homogène.

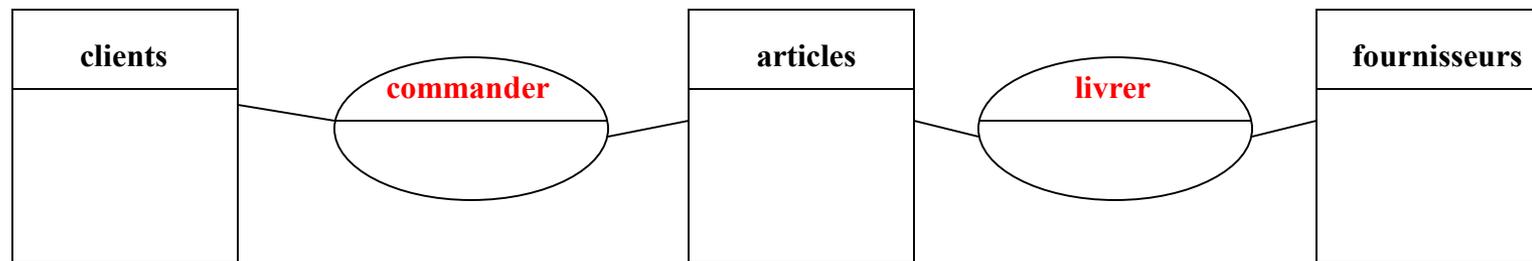
Exemple : les produits ou les articles vendus par une entreprise peuvent être regroupés dans une même entité articles, car d'un Article à autre, les informations ne changent pas de nature (à chaque fois, il s'agit de la désignation, du prix unitaire, etc.).



Les articles et les clients ne peuvent pas être groupés : leurs Informations ne sont pas homogènes.

## Modèle Conceptuel de données

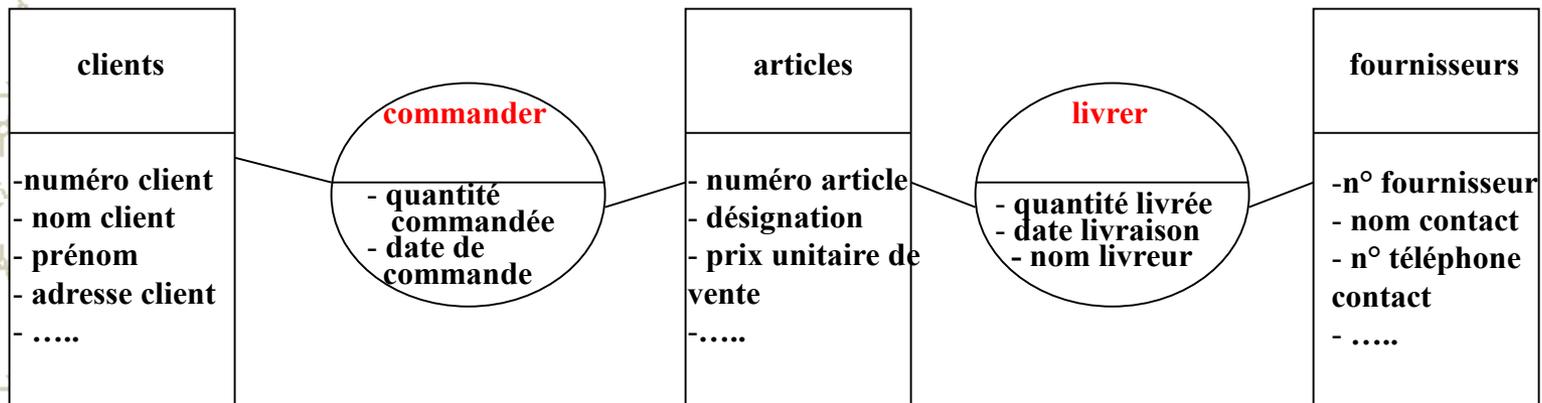
**Association** : c'est une liaison qui a une signification précise entre plusieurs entités.



**Remarque** : les entités **clients** et **fournisseurs** sont liés indirectement, via l'entité **articles**

## Modèle Conceptuel de données

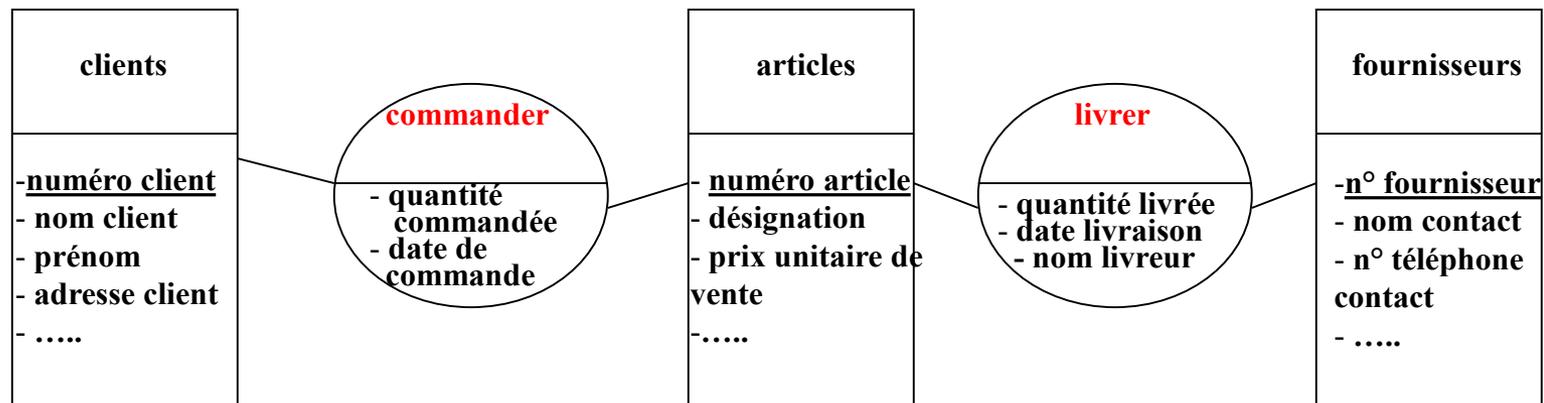
**Attribut** : c'est une propriété d'une entité ou d'une association.



## Modèle Conceptuel de données

**Identifiant** : c'est un attribut sans doublon que toutes les entités doivent posséder. Chaque individu d'une entité doit être identifiable de manière unique.

Par convention, l'identifiant est souligné sur le schéma.

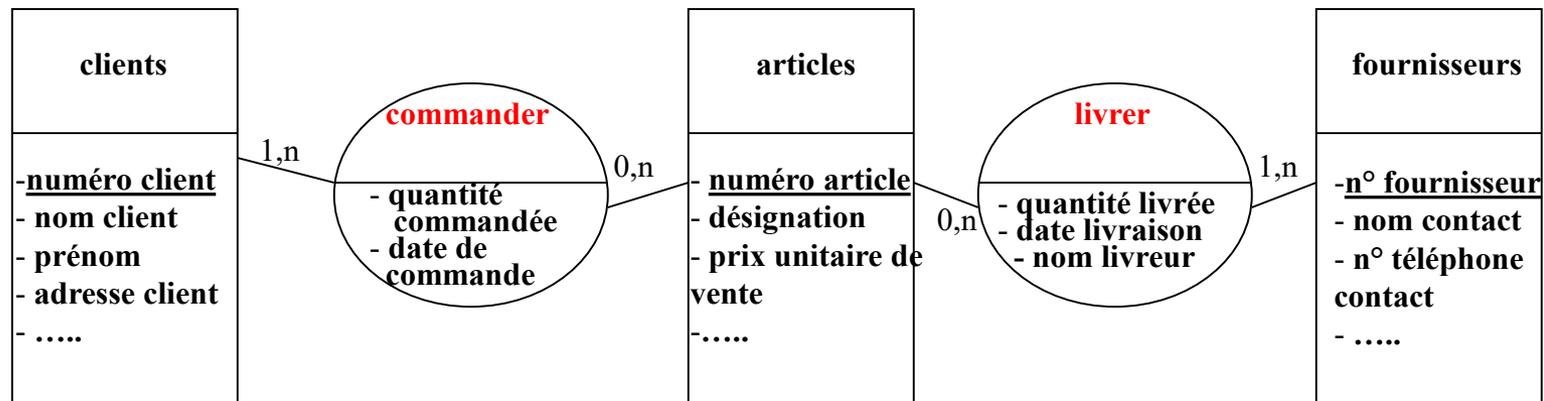


Remarques : - une entité possède au moins un attribut (**Son identifiant**)  
- au contraire, une association peut être dépourvue d'attribut

## Modèle Conceptuel de données

**Cardinalités** : précise le minimum et le maximum de fois q'un individu de l'entité peut être concerné par l'association.

Exemple :



Un client a au moins commandé un article et peut commander **n** articles (**n** étant indéterminé)

Tandis qu'un article peut avoir été commandé entre 0 et **n** fois.

**Remarque** : le schéma comporte des erreurs de conception qui seront corrigées Plus loin.

# Modèle Conceptuel de données

## Discussion sur les cardinalités :

- Card. Mini. de 1 doit se justifier par le fait que les individus de l'entité en question ont besoin de l'association pour exister.

- la discussion autour d'une Card. Mini. 0 ou 1 n'est vraiment intéressante que lorsque la Card. Max. est 1.

- sur l'exemple : un article peut être commandé par plusieurs clients. Cela provient du fait que tous les crayons rouges ne sont pas numérotés individuellement, mais portent un numéro d'article collectif. Plus rigoureux, notre entité articles aurait s'appeler types d'articles.

Pour établir correctement les cardinalités, il faut poser la question dans le bon sens

Autour de l'association commander, par exemple :

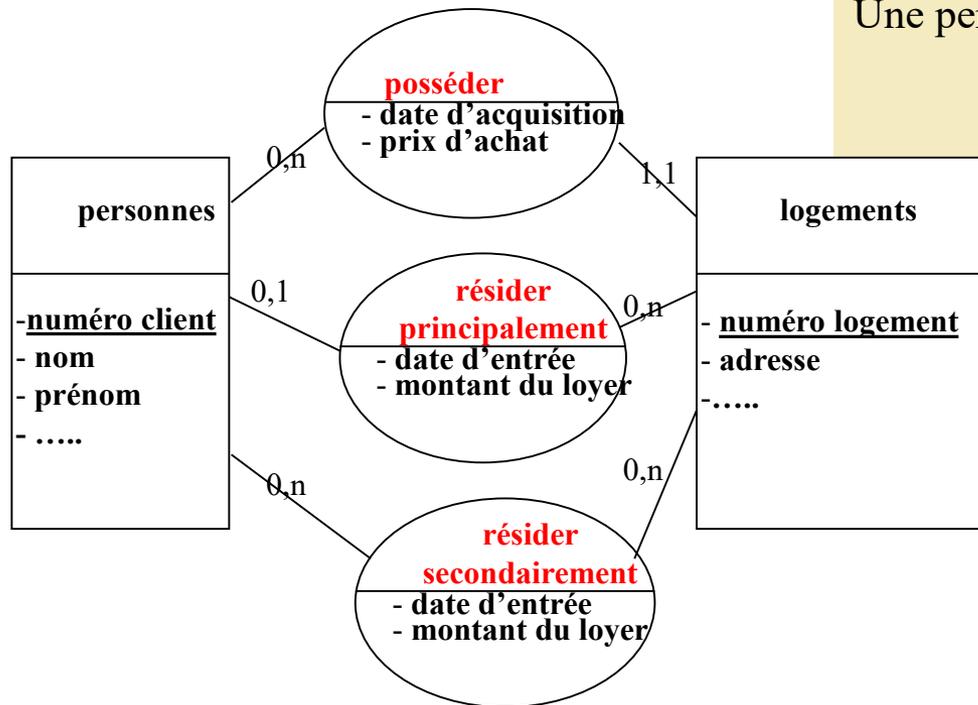
- côté clients, la question est « un client peut commander combien d'articles? » et la réponse est « entre 1 et plusieurs ».

- côté articles, la question est « un article peut être commandé par combien de clients? » et cette fois-ci la réponse est « entre 0 et plusieurs ».

# Modèle Conceptuel de données

Associations plurielles : Deux mêmes entités peuvent être plusieurs fois en association

Exemple d'une agence immobilière gérant des logements:



Une personne peut être :

- propriétaire
- résider principalement,
- résider secondairement.

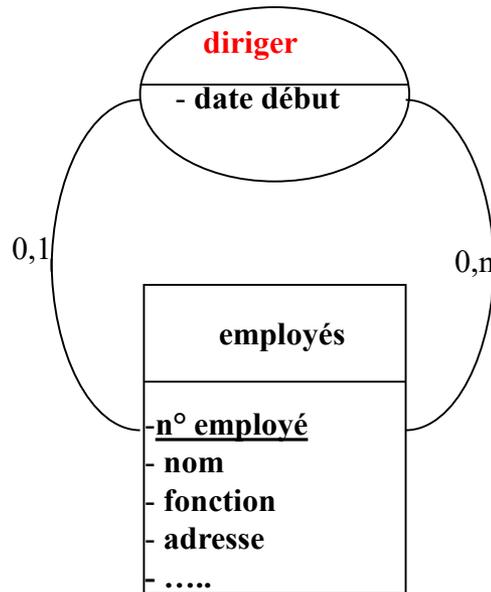
Un logement non géré ne figure pas dans l'entité ce qui explique les Card. 0.

On suppose qu'un logement est détenu par une seule personne qui figure obligatoirement dans l'entité des personnes

# Modèle Conceptuel de données

Association réflexive : Etre branché plusieurs fois à la même entité est permis pour une association.

## Exemple d'une association réflexive binaire



- Tout employé est dirigé par un autre employé (sauf le directeur général).

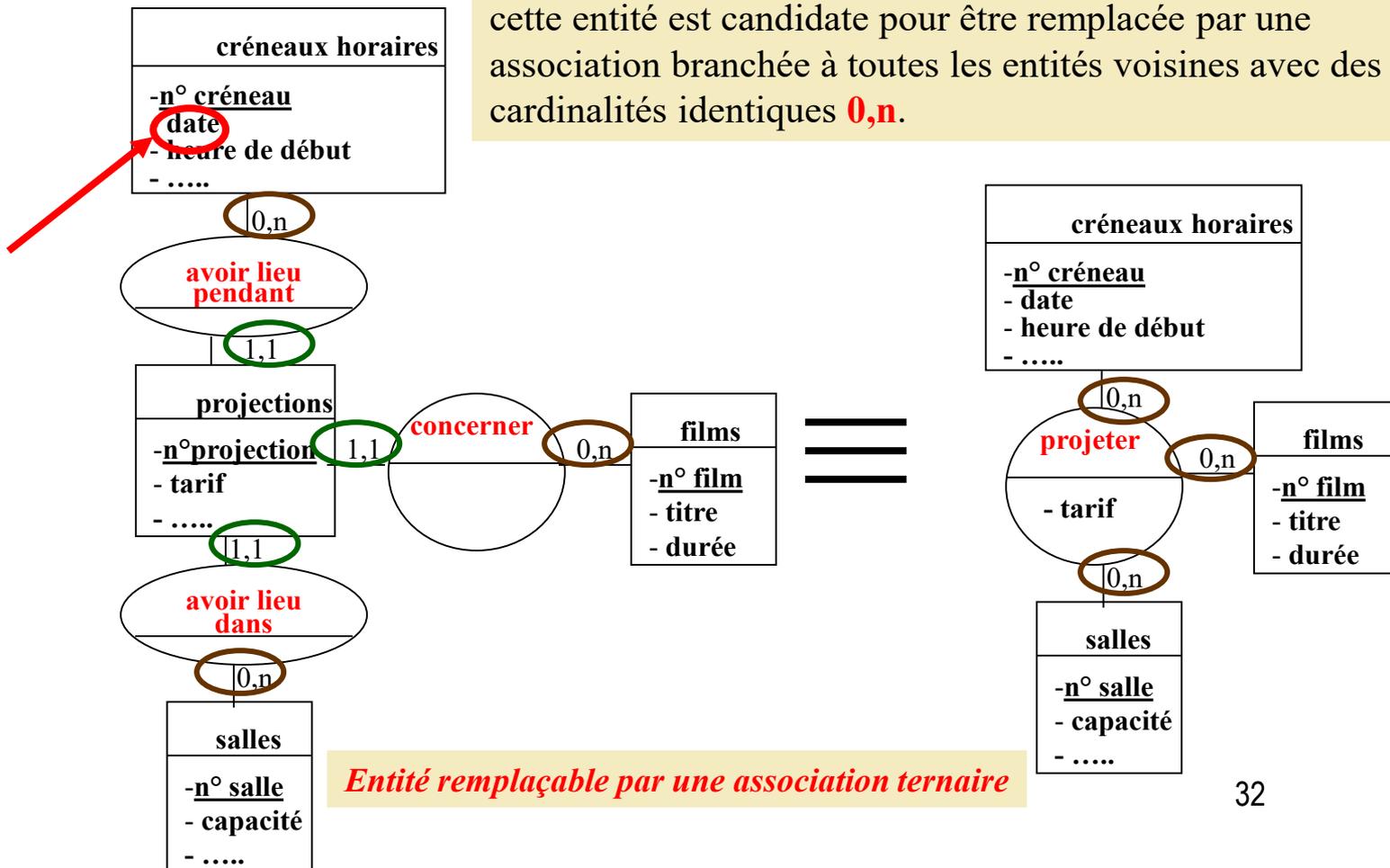
- Un employé peut diriger plusieurs autres employés

# Modèle Conceptuel de données

## Associations non binaires

### Exemple issu d'un cinéma

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**.

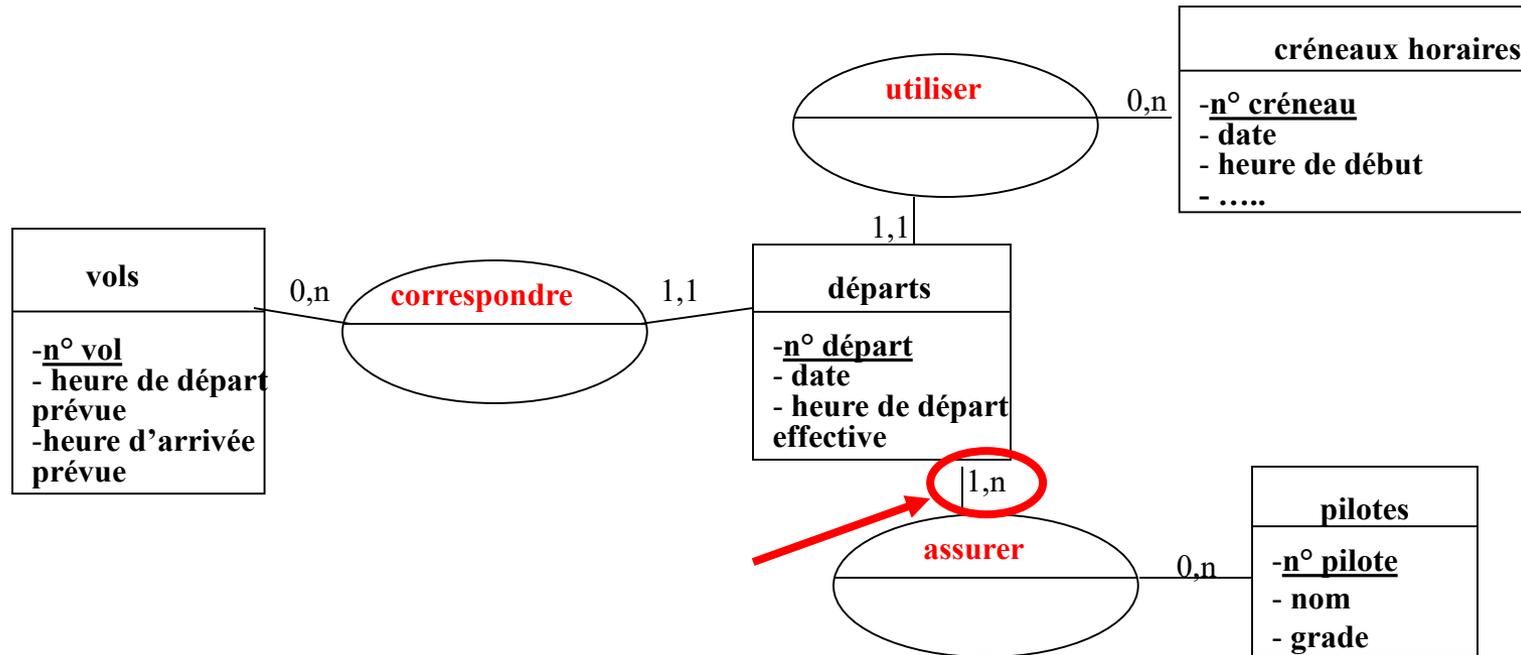


# Modèle Conceptuel de données

## Remarque

Il est conseillé d'en passer par un schéma entités-associations dans lequel on ne trouve que des associations binaires, puis de repérer les entités remplaçables par des associations, comme dans l'exemple précédent.

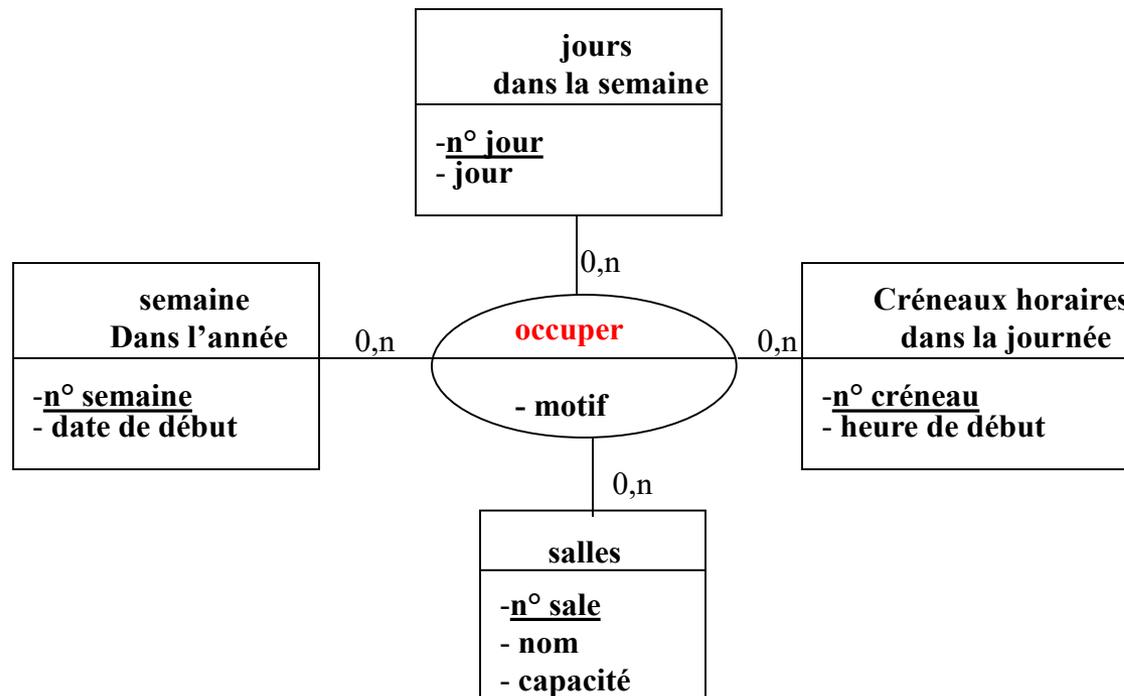
La difficulté de concevoir une association ternaire (ou plus) directement est d'établir les bonnes cardinalités.



## Modèle Conceptuel de données

Une association peut être branchée à plus de trois entités (**4-aire par exemple**).

Il est conseillé de passer par un schéma intermédiaire c.à.d une **entité occupations** et **4 associations** binaires en vérifiant les **cardinalités**.



## REGLES DE NORMALISATION

Un schéma entités-associations doit répondre à 9 règles de normalisation :

### 1- Normalisation des entités :

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

Exemple issu d'un cinéma

### 2- Normalisation des noms :

Le nom d'une entité, d'une association ou d'un attribut doit être unique.

#### **Conseils :**

- pour les entités, utiliser un nom commun au pluriel (par exemple : **clients**);

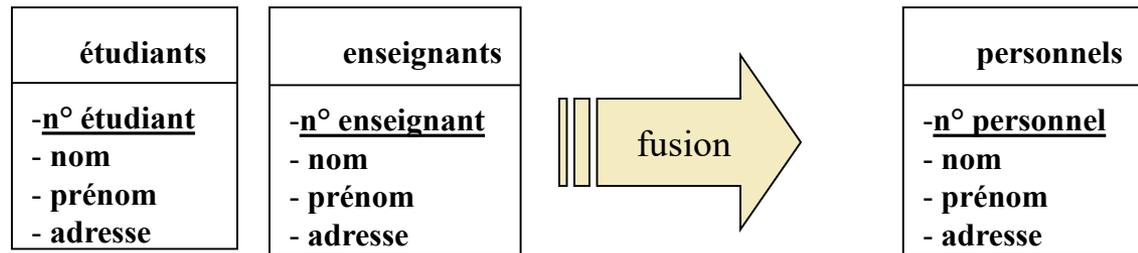
-pour les associations, utiliser un verbe à l'infinitif (par exemple : **effectuer**, **concerner**), une forme passive (par exemple : **être commandé**) et accompagné d'un adverbe (**avoir lieu dans**, **pendant**, **à**);

-pour les attributs, utiliser un nom commun (par exemple : **nom**, **numéro**, **libellé**, **description**) éventuellement accompagné du nom de l'entité ou de l'association dans laquelle il se trouve (par exemple : **nom de client**, **numéro d'article**)

# Modèle Conceptuel de données

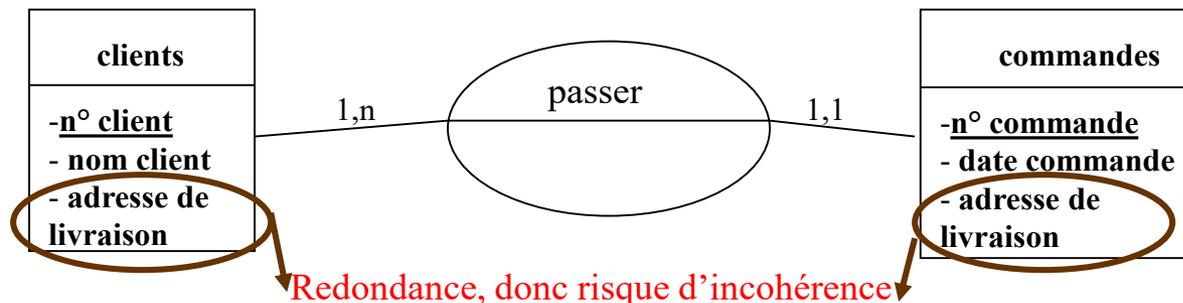
**remarque :** Lorsqu'il reste plusieurs fois le même nom, c'est parfois symptomatique d'une modélisation qui :

n'est pas terminée



**OU**

Le signe d'une redondance



## Modèle Conceptuel de données

3- Normalisation des identifiants : chaque entité doit posséder un identifiant.

### Conseils :

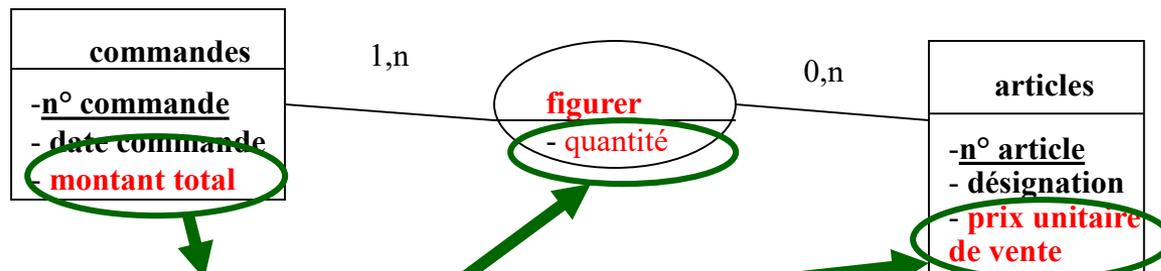
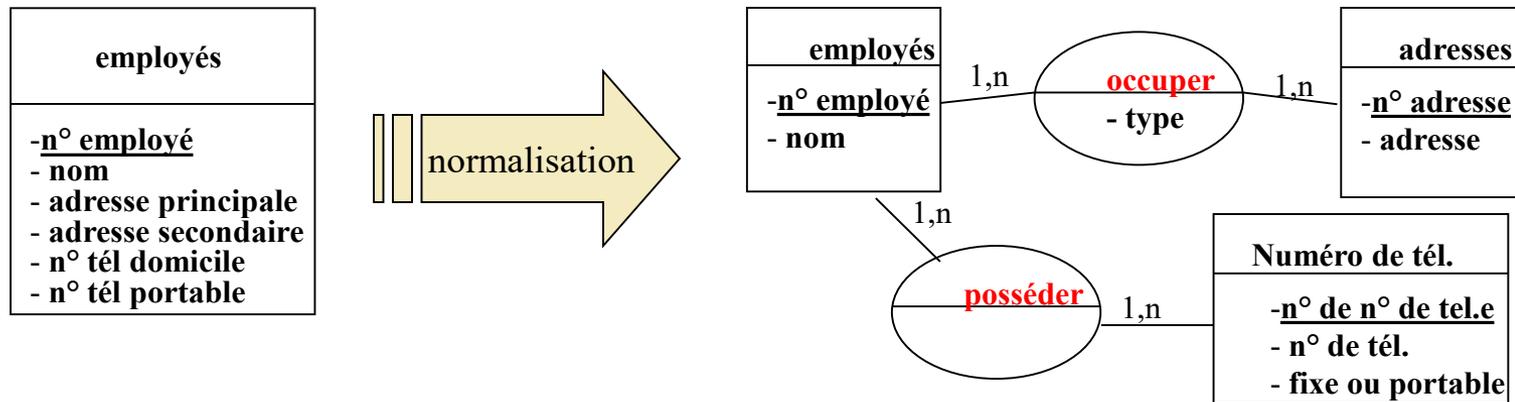
-éviter les identifiants composés de plusieurs attributs  
( exemple : identifiant formé par nom et prénom)

-Préférer un identifiant court pour rendre la recherche la plus rapide possible  
c.à.d par exemple éviter : n° plaque d'immatriculation  
n° de SS  
code postal

# Modèle Conceptuel de données

## 4- Normalisation des attributs :

- Remplacer les attributs en plusieurs exemplaires en une association supplémentaire de cardinalités maximales **n**.
- Ne pas ajouter d'attribut calculable à partir d'autres attributs.

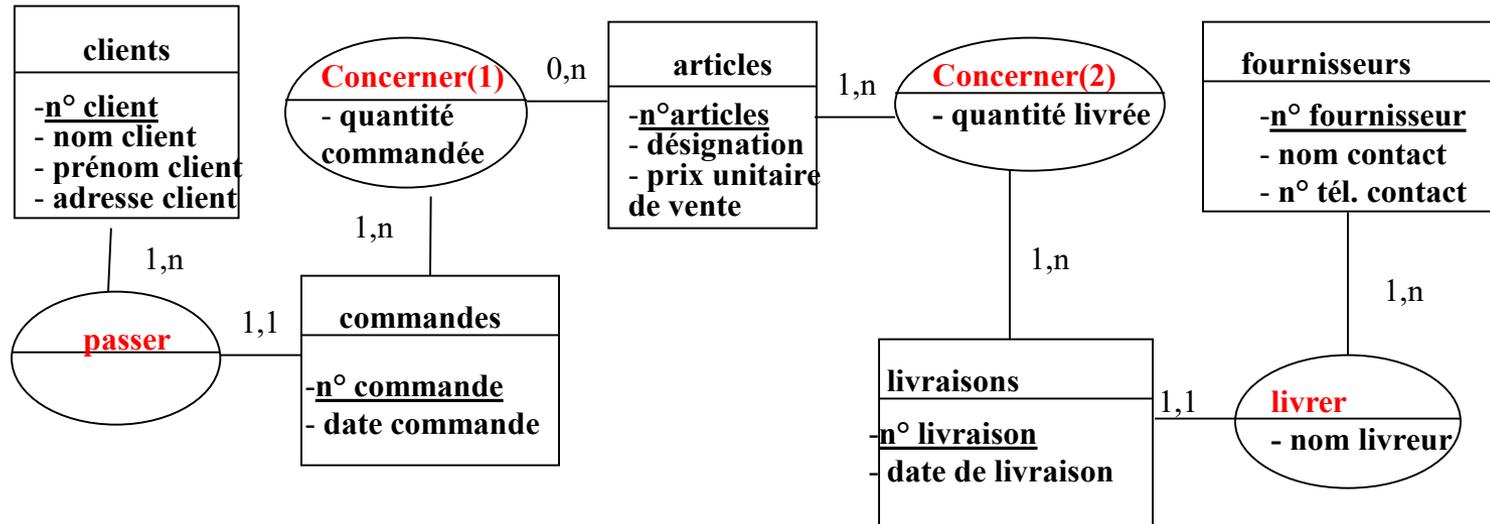
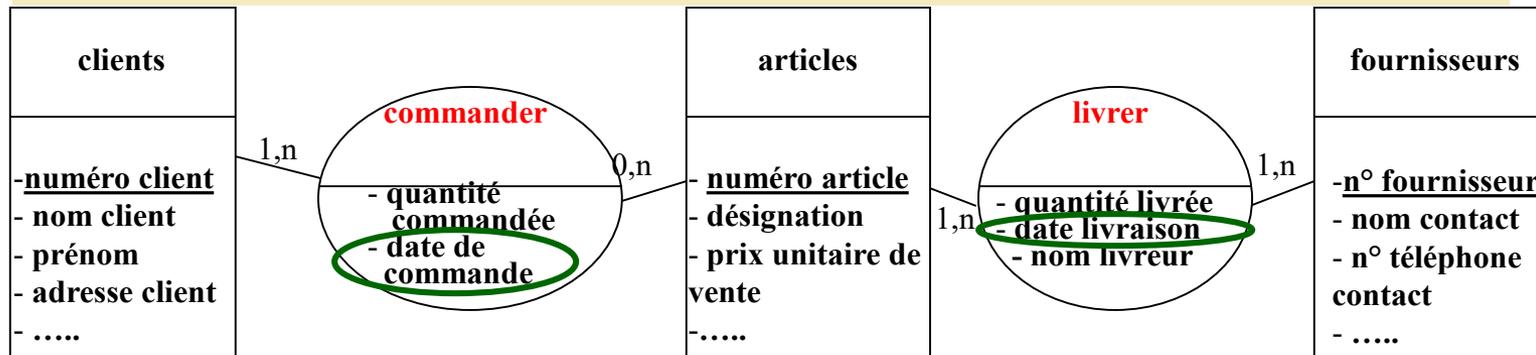


**calculable à partir de  
qu'il faut retirer du schéma**

# Modèle Conceptuel de données

## 5- Normalisation des attributs des associations :

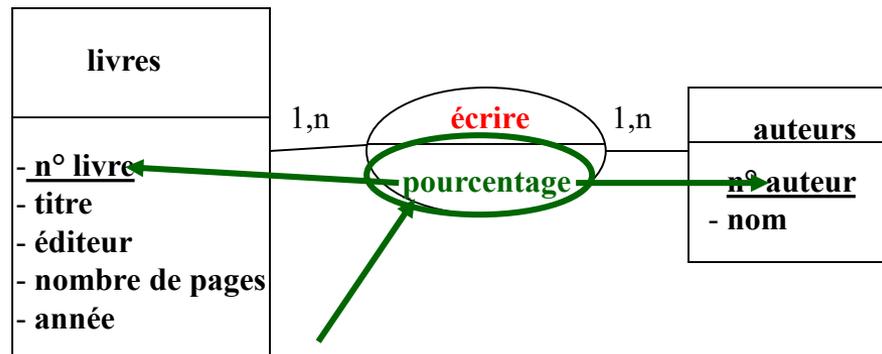
- Les attributs d'une association doivent dépendre directement des identifiants de **toutes** les entités en association.



## Modèle Conceptuel de données

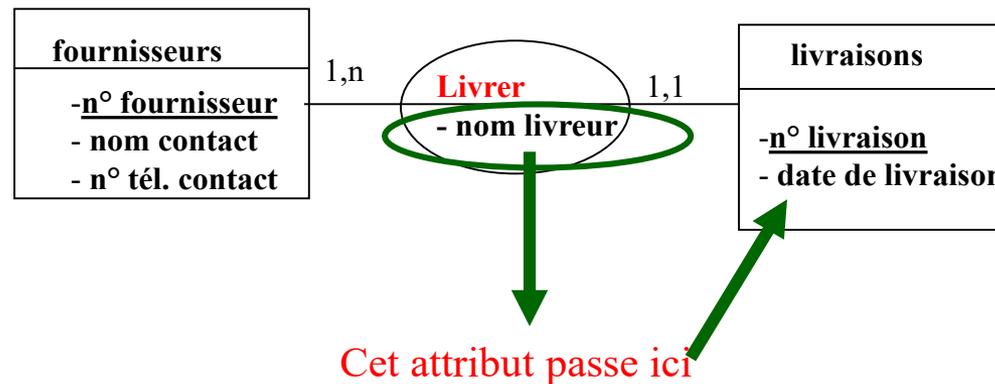
**Inconvénient :** Difficile à appliquer pour les associations ne possédant pas d'attribut

Pour vérifier malgré tout qu'une association sans attribut est bien normalisée, on peut donner temporairement à cette association un attribut imaginaire (mais pertinent) qui permet de vérifier la règle.



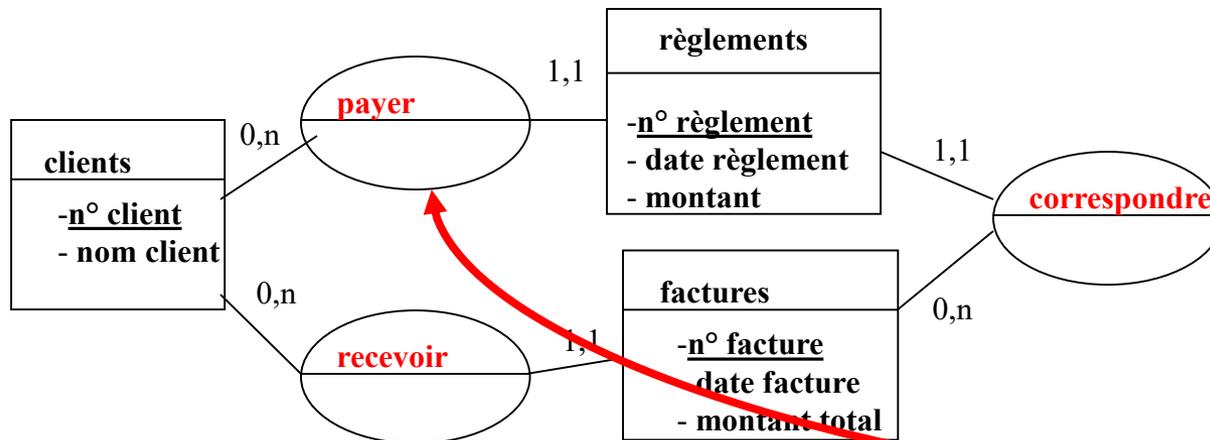
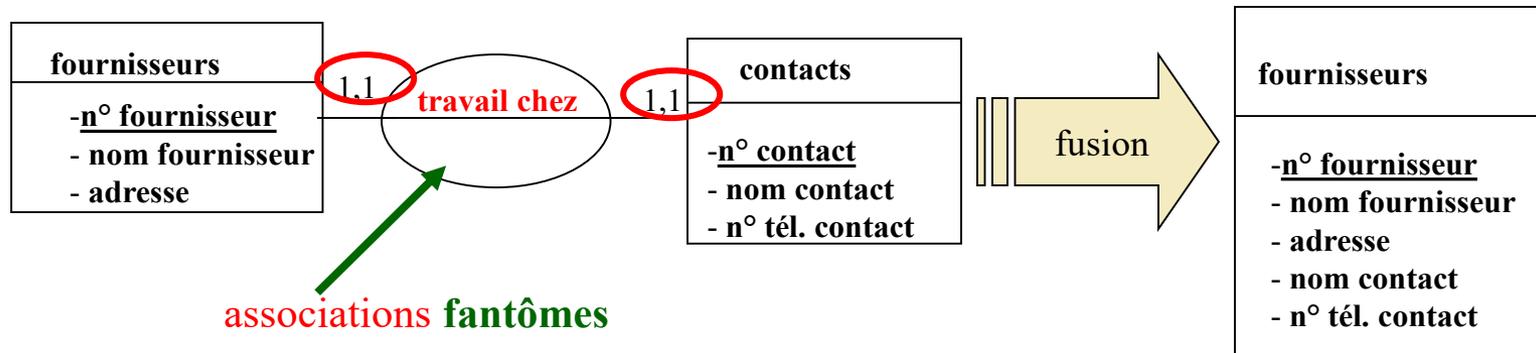
## Autre conséquence :

Une entité avec un cardinalité de 1,1 ou 0,1 aspire les attributs de l'association



# Modèle Conceptuel de données

6- Normalisation des associations : il faut éliminer les **associations fantômes**, redondantes ou en plusieurs exemplaires



Couper le chemin le plus courte

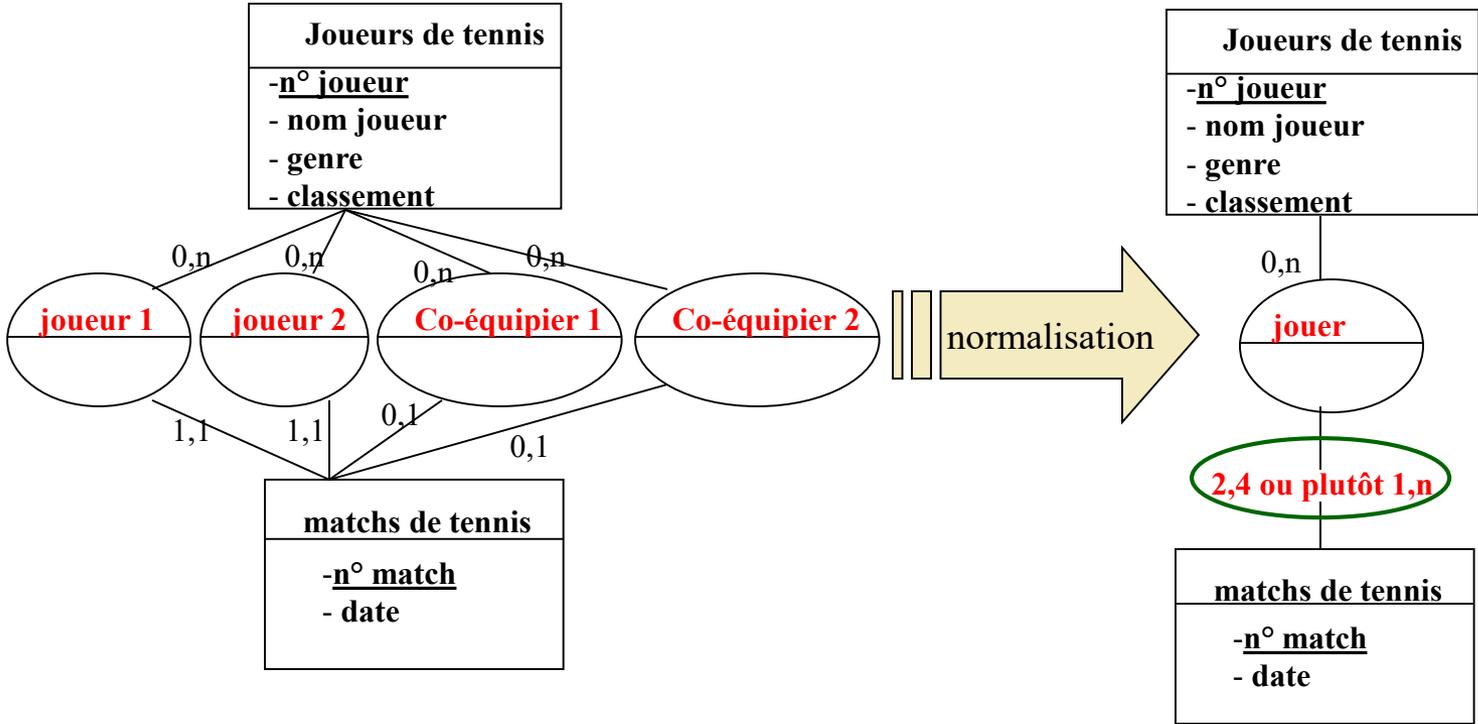
Ou

Retirer l'entité règlements et d'ajouter une association règle avec les mêmes attributs (sauf l'identifiant) entre clients et factures

Si un client ne peut pas régler la facture d'un autre client, alors l'association **payer** est inutile et doit être supprimée (maintenue dans le cas contraire)

# Modèle Conceptuel de données

Une association suffit pour remplacer les 4 associations



## Modèle Conceptuel de données

7- Normalisation des cardinalités : une cardinalité minimale est toujours 0 ou 1 ( et pas 2, 3 ou n) et une cardinalité maximale est toujours 1 ou n ( et pas 2, 3 ...)

Cela se justifie par le fait que même si nous connaissons n au moment de la conception, il se peut que cette valeur évolue au cours du temps. Il vaut mieux considérer n comme une inconnue dès le départ.

Même chose pour une cardinalité minimale supérieur à 1.

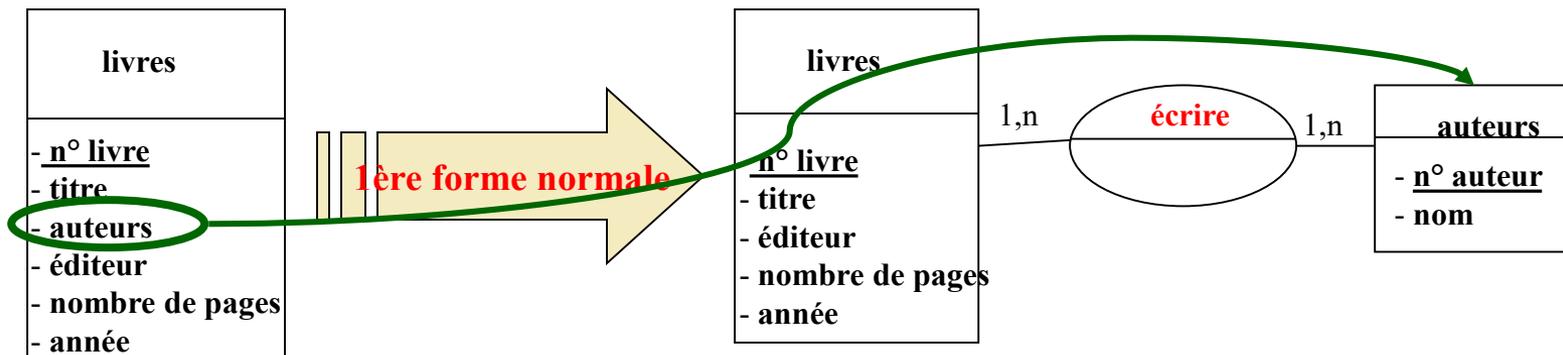
## LES FORMES NORMALES

A ces **7 règles de normalisation**, il convient d'ajouter les **3 premières formes normales**.

### 1- Première forme normale :

À un instant donné dans une entité, pour un individu, un attribut ne peut prendre qu'une valeur et non pas, un ensemble ou une liste de valeurs.

Si un attribut prend plusieurs valeurs, alors ces valeurs doivent faire l'objet d'une Entité supplémentaire, en association avec la première .



# Modèle Conceptuel de données

## 2- Deuxième forme normale :

L'identifiant peut être composé de plusieurs attributs mais les autres attributs de l'entité doivent dépendre de l'identifiant en entier (et non pas une partie de cet identifiant).

## 3- Troisième forme normale de Boyce-Codd :

Tous les attributs d'une entité doivent dépendre directement de son identifiant et d'aucun autre attribut.

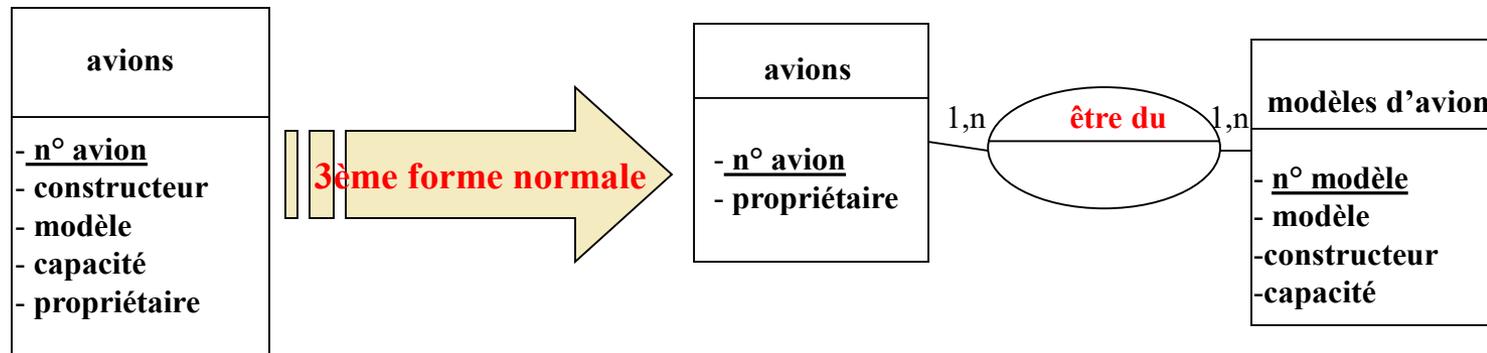
Si ce n'est pas le cas, il faut placer l'attribut pathologique dans une entité séparée, mais en association avec la première.

n° avion	constructeur	modèle	capacité	propriétaire
1	Airbus	A380	180	Air France
2	Boeing	B747	314	British Airways
3	Airbus	A380	180	KLM

Risque d'incohérence car Il y a redondance dans les colonnes *constructeur* et *capacité*

# Modèle Conceptuel de données

n° avion	constructeur	modèle	capacité	propriétaire
1	Airbus	A380	180	Air France
2	Boeing	B747	314	British Airways
3	Airbus	A380	180	KLM



L'entité avions dont les valeurs sont données dans le tableau, n'est pas en 3ème forme normale de Boyce-Bodd, car la **capacité** et **constructeur** d'un avion ne dépend pas du **n° d'avion** mais de son **modèle**.

La solution normalisée

## DEPENDANCES FONCTIONNELLES (DF)

### 1 - Définitions et propriétés :

Un attribut Y **dépend fonctionnellement** d'un attribut X si et seulement si une Valeur de X induit une unique valeur de Y. On note  $X \longrightarrow Y$ .

#### exemple

Si X est le **numéro** de client et Y le nom de **client**, alors on a bien  $X \longrightarrow Y$ . par contre on a pas  $Y \longrightarrow X$ , car plusieurs clients de **numéros** différents peuvent porter le même **nom**.

#### Transitivité :

$X \longrightarrow Y$  et  $Y \longrightarrow Z$  alors  $X \longrightarrow Z$

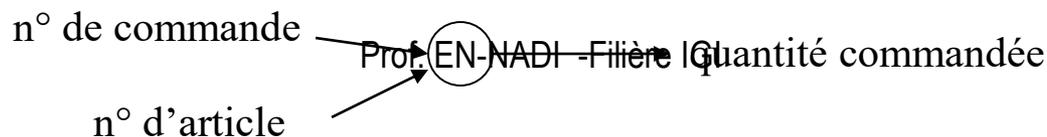
Dite directe  
qui nous intéresse

Dite transitive

Un attribut peut avoir une DF qui repose sur la jonction  
De plusieurs attributs, auquel la dépendance est dite **non élémentaire**.

#### Notation et exemple

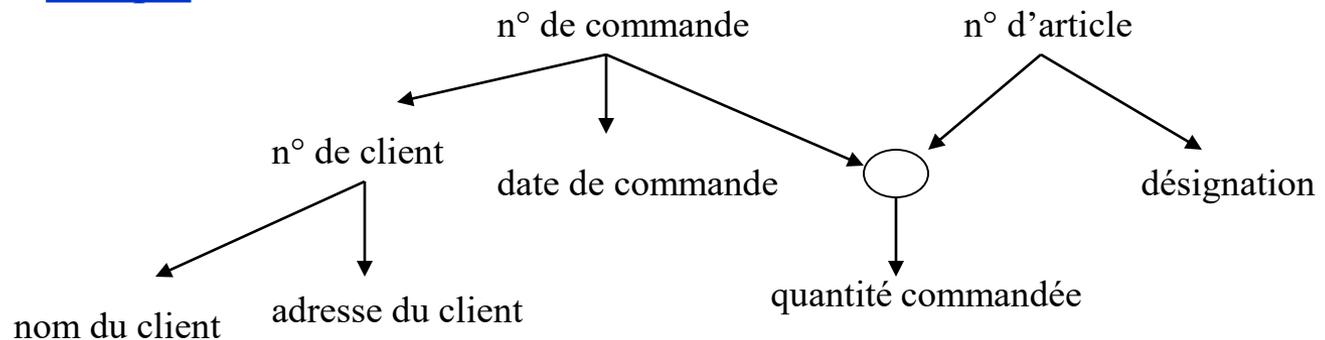
n° de commande + n° d'article  $\longrightarrow$  quantité commandée



## 2 – Graphe de couverture minimale :

En représentant tous les attributs et toutes les DF directes entre eux, nous obtenons un réseau appelé graphe de couverture minimale.

### exemple



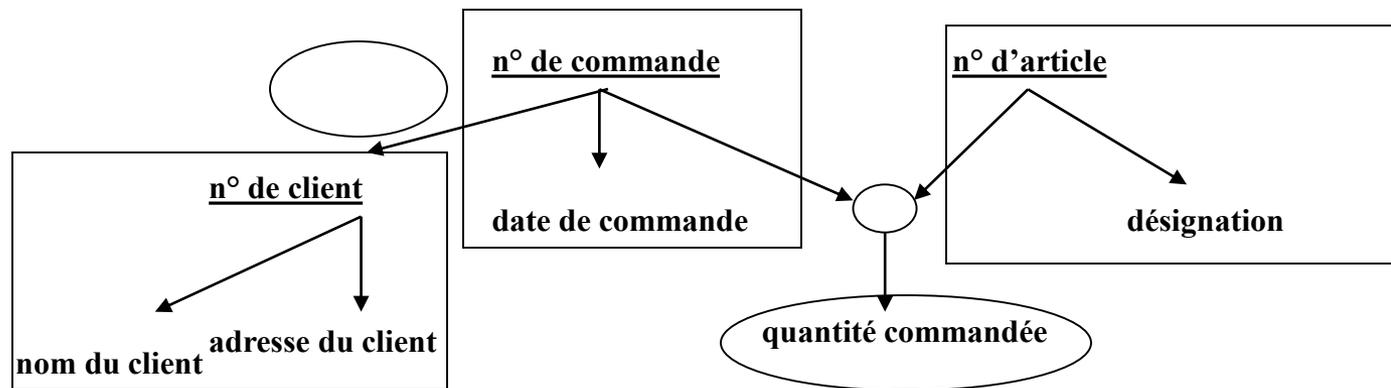
Toutes les DF du graphe doivent partir d'un identifiant.

Si ce n'est pas le cas, c'est qu'un identifiant **a été omis**.

## 3 – Traduction vers un schéma entités-associations :

Identification des entités et des associations sur un graphe de couverture minimale.

Reprenons l'exemple précédent :



## Schéma entités-associations normalisé apparaît naturellement en suivant quelques étapes simples :

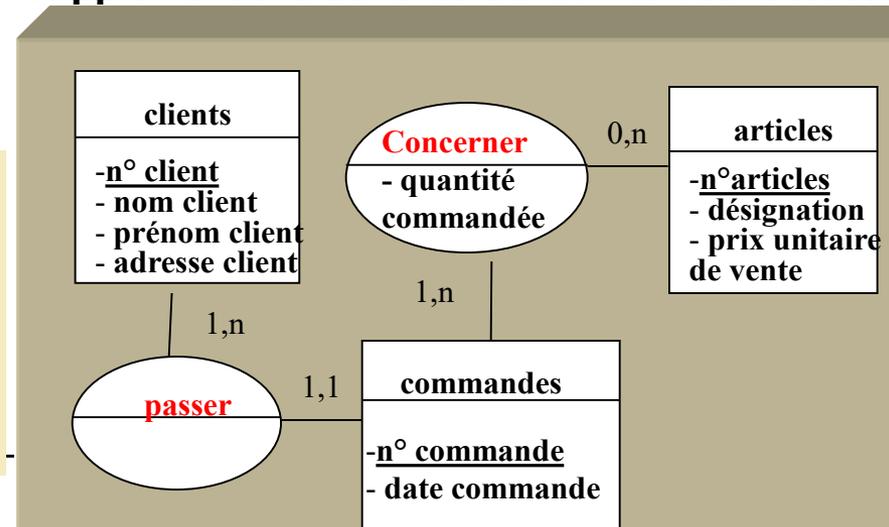
**étape 1** : il faut repérer et souligner les identifiants.

**étape 2** : tous les attributs non identifiant qui dépendent directement d'un identifiant et d'un seul, forment une entité (avec l'identifiant).

**étape 3** : les dépendances élémentaires entre les identifiants forment des associations binaires dont les card. Max. sont **1** au départ de la **DF** et **n** à l'arrivée.

**étape 4** : les attributs (non identifiant) qui dépendent de plusieurs identifiants sont les attributs d'une association supplémentaire dont les **Card. Max.** sont toutes **n**.

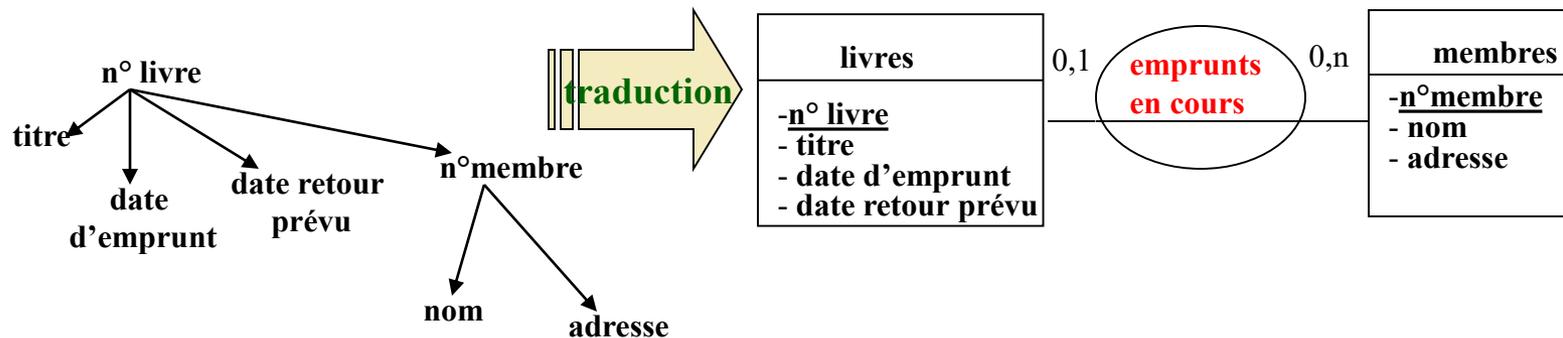
*Traduction vers un schéma entités-associations normalisé de notre exemple dont nous avons ajouter un nom aux entités et aux associations chose qui n'existe pas dans le graphe.*



## 3 – Gestion des dates et du caractère historique :

Cas d'une bibliothèque : stocker les **emprunts en cours** et/ou **historiques**.

**Pour les emprunts en cours, la date de retour prévu est un attribut de l'entité livres car un livre ne peut faire l'objet que d'un seul emprunt en cours.**



**dans ce cas, l'établissement du graphe de couverture minimale ne pose aucun problème**

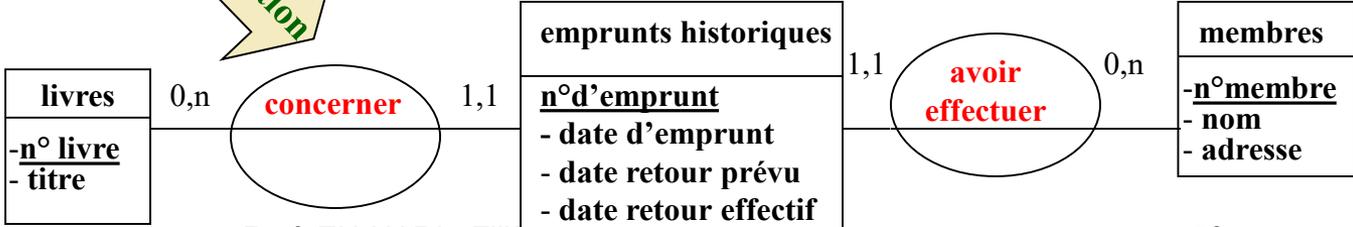
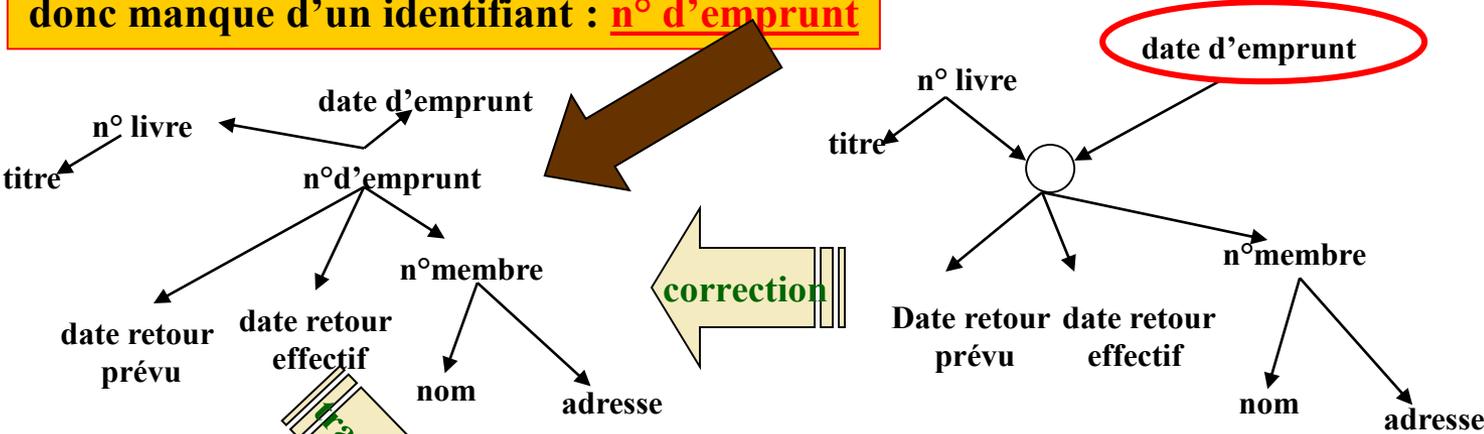
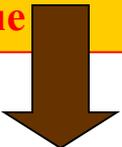
# Modèle Conceptuel de données

Par contre un livre peut faire l'objet de plusieurs emprunts historiques



**Date d'emprunt est déterminante pour connaître la date de retour prévue**

Date n'est pas un identifiant et une **DF** ne peut partir que d'un ou plusieurs identifiants donc manque d'un identifiant : n° d'emprunt

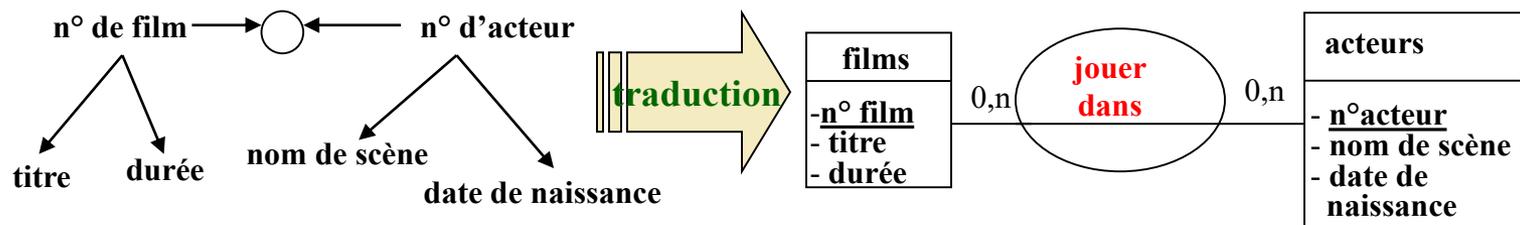


## 4 – Association sans attributs :

Les associations dont les Card. Maxi. Sont toutes n et qui sont attributs ne figurent pas sur le graphe de couverture minimale. Il faut alors :

- Soit leur inventer temporairement un attribut
- Soit introduire une notation spéciale (par exemple, une dépendance élémentaire qui ne débouche pas sur aucun attribut)

**Exemple :**



## **METHODOLOGIE DE BASE**

**Face à une situation bien définie, nous pouvons procéder sans établir le graphe de couverture minimale :**

- **Identifier les entités en présence;**
- **lister leurs attributs;**
- **ajouter les identifiants (numéro arbitraire et auto-incrémenté);**
- **établir les associations binaires entre les entités;**
- **lister leurs attributs;**
- **calculer les cardinalités;**
- **vérifier les règles de normalisation et en particulier, la normalisation des entités ( c'est à ce stade qu'apparaissent les associations binaires), des associations et de leurs attributs ainsi que la troisième forme normale de Boyce-Codd.**
- **effectuer les corrections nécessaires.**

## Modèle Conceptuel de données

**Mais, il est parfois plus intuitif d'en passer par l'étude des DF directes :**

- Identifier les entités en présence et leur donner un identifiant (numéro arbitraire et auto-incrémenté);
- ajouter l'ensemble des attributs et leurs DF directes avec les identifiants (en commençant par les dépendances élémentaires);
- traduire le graphe de couverture minimale obtenu en un schéma entités-associations;
- ajouter les cardinalités minimales et maximales;
  
- à ce stade, la majorité des règles de normalisation devraient être vérifiées, il reste tout de même la normalisation des noms, la présence d'attributs en plusieurs exemplaires et d'associations redondantes ou en plusieurs exemplaires, à corriger.

Garder à l'esprit que le modèle doit être **exhaustif**.

Eviter toute **redondance**.

Eliminer les **synonymes**. Exemple : nom, patronyme, appellation

Eliminer les **polysèmes**. Exemple : qualité, statut



**MLD**

**LE MODELE LOGIQUE  
DE DONNEES**

## Modèle logique de données (MLD)

Une fois le MCD est établi

TRADUCTION

Aux différents systèmes logiques

systèmes logiques :

- Avant apparition des SGBD, les données étaient stockées dans des fichiers binaires et gérées par des programmes exécutables.

Les programmes développés par exemple en : Basic, Cobol, Dbase ....

Proposition d'une traduction d'un MPD (Modèle physique de données) en MLD

Problématiques :

Maintenance de programmes notamment la modification de la structure des données.

- - Apparition des SGBD hiérarchiques, les données sont organisées en arbre

Exemple : IMS-DL1 d'IBM

- Puis les SGBD réseaux → organisation sur un graphe plus général  
Exemple : IDS2 de Bull MPD → MLD Codasyl

Les SGBD hiérarchiques et réseaux sont dits navigationnels

c.à.d connaître l'information → connaître le chemin d'accès

Aujourd'hui sont remplacés pas des SGBD relationnels (SGBDR)

(SQL : Structured Query Langage)

Information obtenue par une requête formulée dans un langage naturel

**Exemple** : Oracle, Access, SQL Server, DB2.....

Récemment , apparition du **Modèle logique** orienté objet et **même des SGBD** orientés objets

Les **SGBDR** restent pourtant majoritaires

Concentrons-nous désormais sur le **MLDR**

## 1 – Tables, lignes et colonnes :

Lorsque les données ont la même structure on peut les organiser en **TABLE**

- **Les colonnes** décrivent les champs en commun

- **Les lignes** contiennent les valeurs de ces champs pour chaque enregistrement

Exemple : *renseignements relatifs aux clients*

<b>numéro client</b>	<b>nom</b>	<b>prénom</b>	<b>adresse</b>
1	Alaoui	Mohamed	127, rue....
2	Akioud	Driss	314, boulevard....
3	El bougrini	Lahcen	330, avenue....
5	Fari	Ahmed	2, impasse....
.....	.....	.....	.....

## 2 – Clés primaires et clés étrangères :

Les lignes d'une Table doivent être uniques



Une colonne au moins doit servir à les identifier :  
Il s'agit de la **clé primaire** de la table

La valeur vide (NULL) est interdite dans une colonne qui sert de clé primaire

La valeur de la clé primaire d'une ligne ne devrait pas changer au cours du temps (en principe)

Il se peut qu'une colonne **colonne 1** d'une table ne doive contenir que des valeurs prises par la **colonne 2** d'une autre table.

**Exemple** : le numéro du client sur une commande doit correspondre à un vrai numéro de client

la **colonne 2** doit être sans doublons. On dit alors que la **colonne 1** est la clé étrangère et qu'elle référence la **colonne 2**.

**Remarque** : la **colonne 2** s'agit bien souvent d'une clé primaire

**Convention** : on souligne les clés primaires et on fait précéder les clés étrangères d'un dièse # dans la description des colonnes d'une table

**Exemple** :

clients (numéro client, nom client, prénom client, adresse client)

commandes (numéro commande, date de commande, #numéro client (non vide))

**Remarque** :

- une même table peut avoir plusieurs clés étrangères mais une seule clé primaire
- **une colonne clé étrangère peut aussi être primaire (dans la même table)**
- **une clé étrangère peut être composée ( si la clé primaire référencée est composée par exemple)**
- implicitement, chaque colonne qui compose une clé primaire ne peut recevoir la valeur vide (NULL interdit)
- par contre, si une colonne clé étrangère ne doit pas recevoir la valeur vide, alors il faut le préciser dans la description des colonnes.

## Schéma relationnel :

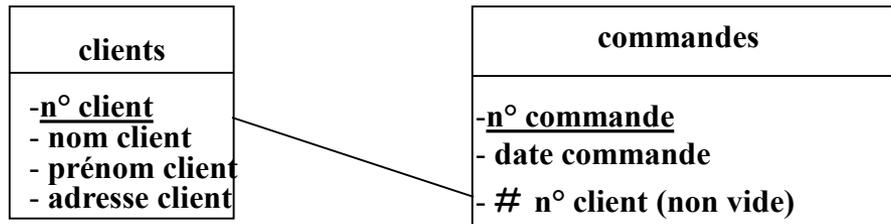
Tables d'une base des données relationnelle



Relations dans un schéma relationnel

- Les liaisons entre les clés étrangères et leur clé primaire est symbolisé par un **connecteur**

*Exemple : schéma relationnel simple entre deux tables.*



*Remarque :* Certains éditeurs inscrivent sur le connecteur un **symbole 1** côté Clé primaire et un **symbole ∞** côté clé étrangère (à condition que Celle-ci ne soit pas déjà clé primaire).

*Attention :* Il faut prendre garde avec cette convention, car le symbole ∞ se trouve du côté **opposé** à la Card. Max. **n** correspondante.

## Traduction d'un MCD en un MLDR :

Pour traduire un MCD en un MLDR, il suffit d'appliquer cinq règles.

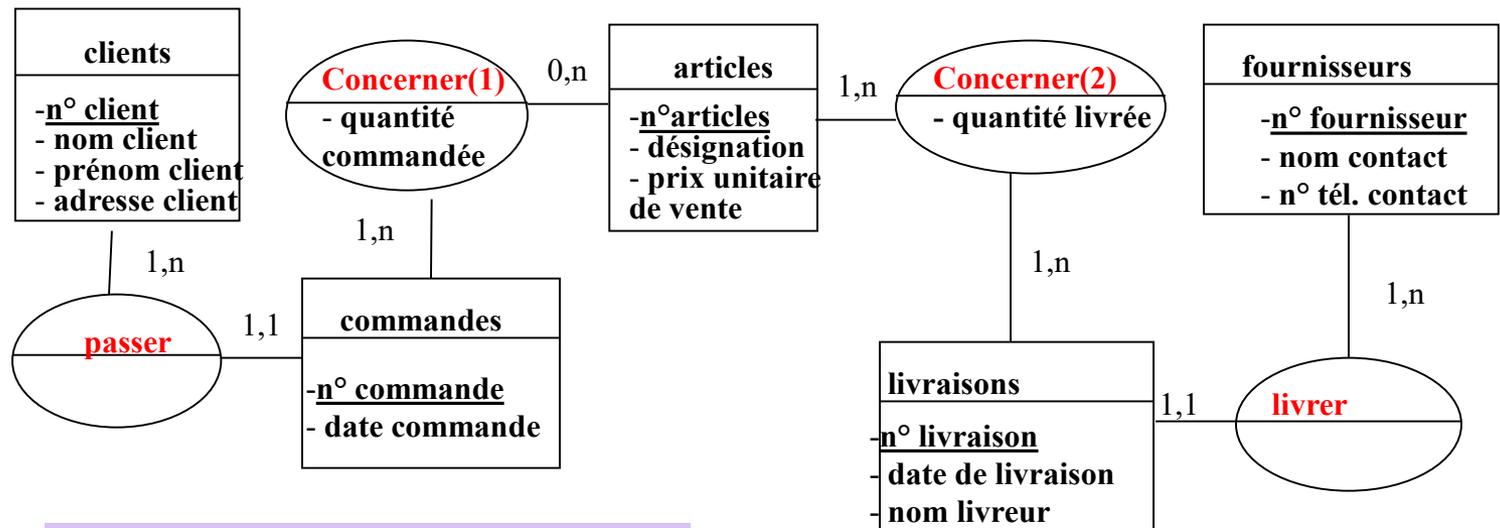
### Notation :

On dit qu'une relation binaire ( entre deux entités ou réflexive) est de type :

- 1 : 1 (un à un) si aucune des deux **card. Max.** n'est **n**;
- 1 : n (un à plusieurs) si une des deux **card. Max.** est **n**;
- n : m (plusieurs à plusieurs) si les deux **card. Max.** sont **n**.

- Un schéma relationnel ne peut faire la différence entre **0,n** et **1,n**.
- Par contre, il peut la faire entre **0,1** et **1,1** (règles 2 et 4)

**Règle 1 : toute entité devient une table dans laquelle les attributs deviennent les colonnes. L'identifiant de l'entité constitue alors la clé primaire de la table.**

Exemple

L'entité articles devient la table :

articles (n° articles, désignation, prix unitaire de vente)

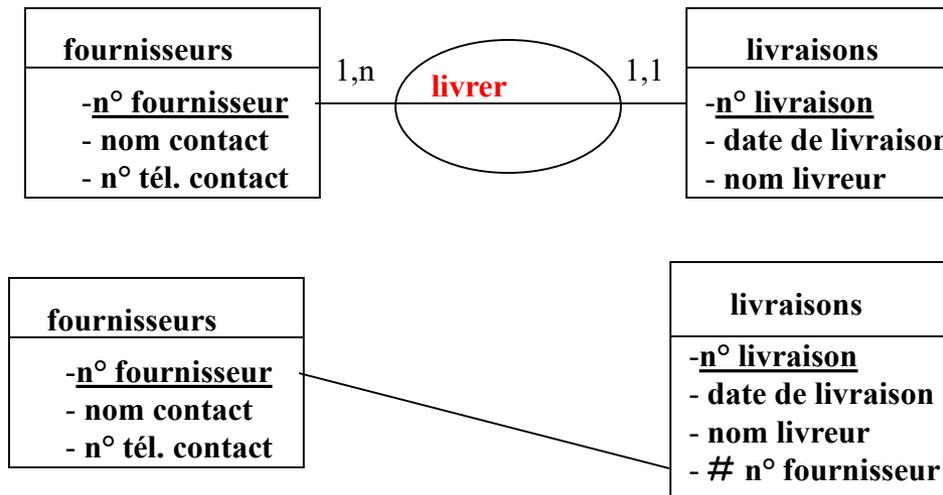
**Règle 2** : une association binaire de type  $1 : n$  disparaît, au profit d'une clé étrangère dans la table côté  $0,1$  ou  $1,1$  qui référence la clé primaire de l'autre table. Cette clé étrangère ne peut recevoir la valeur vide si la cardinalité est  $1,1$ .

**Exemple**

L'association livrer est traduit par :

fournisseurs (n° fournisseur, nom contact, n° téléphone contact)

livraisons (n° livraison, date de livraison, nom livreur, # n° fournisseur (non vide))



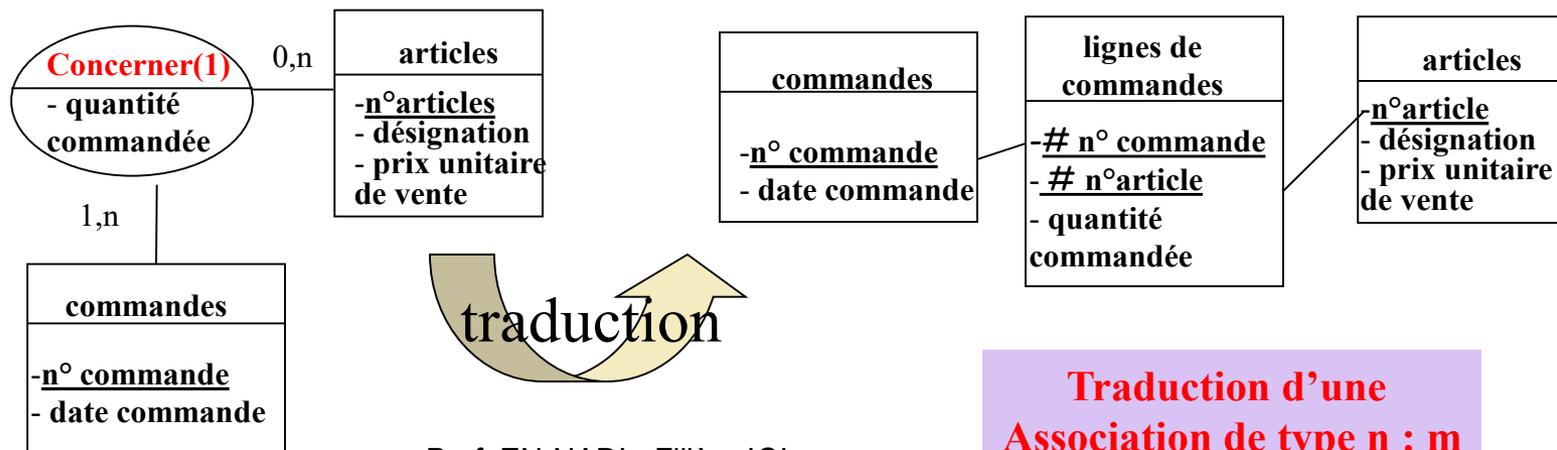
traduction

Il ne devrait pas y avoir d'attribut dans une association de type  $1 : n$ .

**Règle 3** : une association binaire de type  $n : m$  devient une table supplémentaire (parfois appelée table de jonction, table de jointure ou table d'association) dont la clé primaire est composée de deux clés étrangères ( qui référencent les deux clés primaires des deux tables en association). Les attributs de l'association deviennent des colonnes de cette nouvelle table.

**Exemple** L'association **concerner (1)** est traduite par la table supplémentaire **lignes de commande** :

lignes de commande(# n° commande, # n° article, quantité commandée)



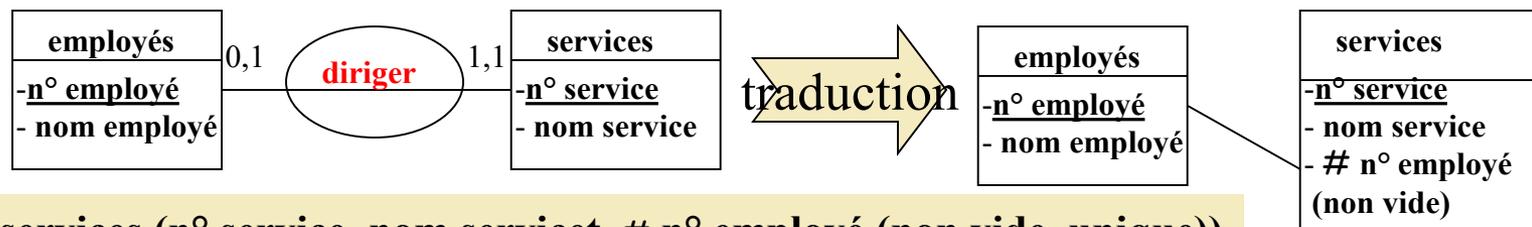
**Traduction d'une Association de type  $n : m$**

***Règle 4 : Une association binaire de type 1 : 1 est traduit comme une association binaire de type 1 : n sauf que la clé étrangère se voit imposer une contrainte d'unicité en plus d'une éventuelle contrainte de non vacuité ( cette contrainte d'unicité impose à la colonne correspondante ne peut prendre que des valeurs distinctes).***

***Si les associations fantômes ont été éliminées, il devrait y avoir au moins un côté de cardinalité 0,1. C'est alors dans la table du côté opposé que doit aller la clé étrangère. Si les deux côtés sont de cardinalité 0,1 alors la clé étrangère peut être placée indifféremment dans l'une des deux tables.***

### ***Exemple***

L'association **diriger** est traduite par :



services (n° service, nom service, # n° employé (non vide, unique))

employés (n° employé, nom employé)

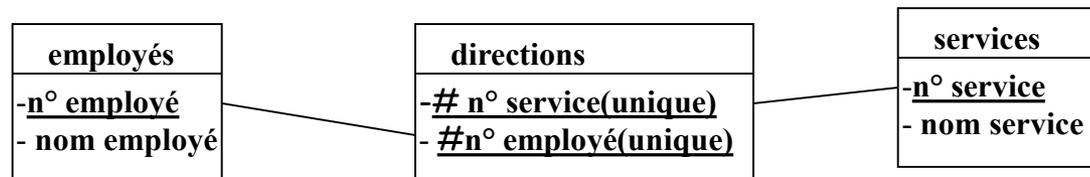
### Alternative de la règle 4 :

*En réalité, la règle 4 proposée ici considère qu'une association binaire de type 1 : 1  
Correspond à une association binaire de type 1 : n particulière. Une alternative  
Consiste à voir une association binaire de type 1 : 1 comme une association binaire  
De type n : m particulière. Il suffit pour cela d'ajouter une contrainte d'unicité sur  
Chacune des clés étrangères de la table de jonction supplémentaire.*

services (n° service, nom service)

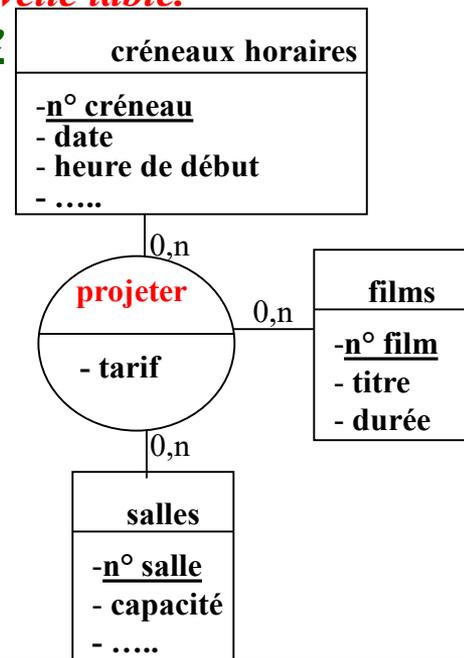
directions ( # n° service (unique), # n° employé (unique))

employés (n° employé, nom employé)

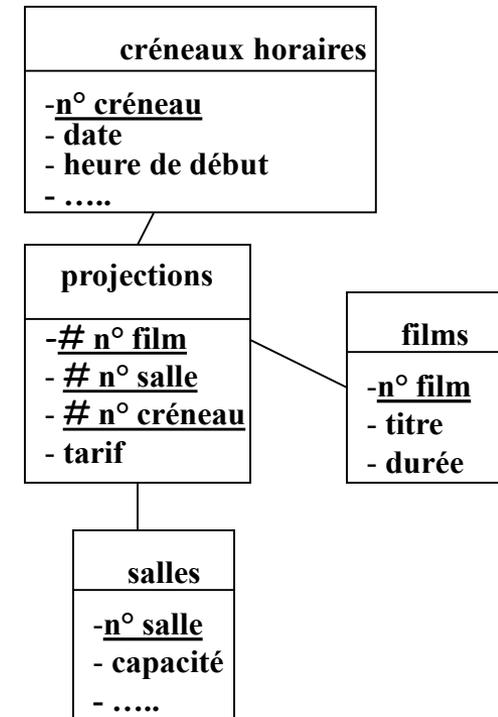


**Règle 5** : Une association non binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entités en association. Les attributs de l'association deviennent des colonnes de cette nouvelle table.

**Exemple**



traduction



L'association **projeter** devient la table :

**projections ( # n° film, # n° salle, # n° créneau, tarif)**