

# Base de données

---

## Séance 2 - MERISE

1. Modèle Conceptuel des Données (MCD)
2. Du Modèle Conceptuel des Données (MCD) au Modèle Logique des Données (MLD)

# La démarche pour construire le MCD

---

- **Recherche** des propriétés à gérer
  - **Regroupement** des propriétés par entité
  - **Construction** des entités
  - **Recherche** des associations
  - **Recherche** des cardinalités
  - **Vérification et validation** du modèle
-

# Recherche des propriétés

---

- Dans les interviews / documents
    - Les données peuvent être éparpillées dans le discours.
  - Une donnée est caractérisée par :
    - Un nom
    - Une définition (explique à quoi elle sert)
    - Un domaine de définition
    - Un mode de calcul (si donnée calculée)
    - Une décomposition (si donnée non atomique)
-

# Regroupement en entités

---

- ❑ Les données sont explicitement des propriétés d'entités nommées.
  - ❑ Les données sont regroupables dans une entité non explicite, mais sémantiquement identifiables.
  - ❑ Les données sont isolées, il faut construire des entités supplémentaires.
-

# Le dictionnaire de données

---

- Recense toutes les informations utiles au système considéré.
- Pourquoi ?
  - Eviter les doubles définitions
  - Rassembler des définitions très proches, ou concurrentes
- Formalisé par un tableau :

Entité	Propriété	Définition	Domaine	Type	Calculée

---

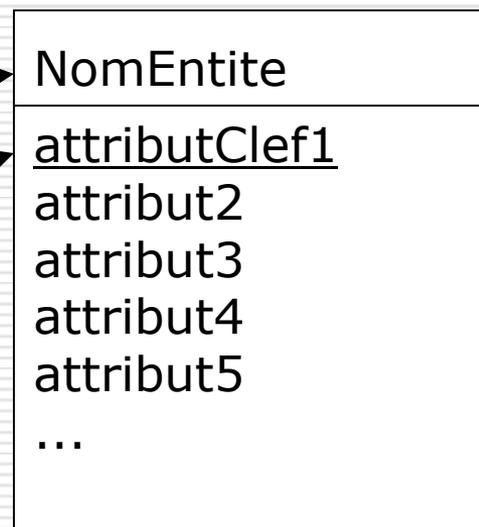
# Construction des entités

---

- Première ébauche du modèle conceptuel des données ne faisant apparaître que :

- Les entités

- Les propriétés



# Clé

---

- Un ensemble d'attributs d'une entité est une clé si et seulement si :
    - deux occurrences de l'entité ont des clés différentes
    - la connaissance des valeurs des attributs clés implique de façon unique les valeurs des attributs non clé
    - les valeurs des attributs clés sont toutes initialisées.
-

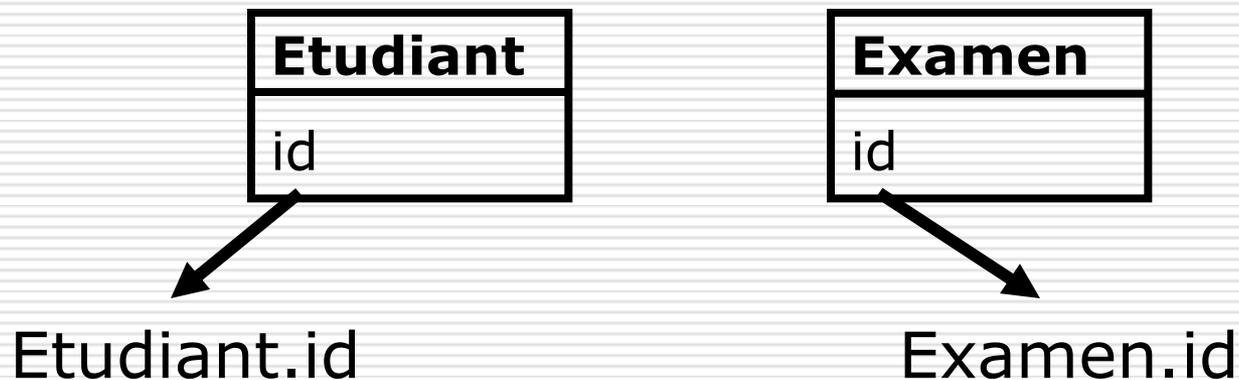
# Noms semblables

---

- Peut-on nommer deux attributs avec le même nom **dans deux entités différentes** ?

**OUI**

- Comment distingue t-on les données dans ce cas ?



On utilise ces formes surtout en cas d'ambigüité.

---

# Recherche des associations

---

- On écrit des phrases « en français » qui mettent en scène les relations entre les entités. On fait apparaître le « QUI dépend de QUOI »
  - Caractéristiques :
    - nom : désigne la nature de la dépendance
    - dimension : désigne le nombre d'entités impliquées
    - qualifiants : des attributs de la liaison
    - cardinalité : donne les limites ordinales d'une association
-

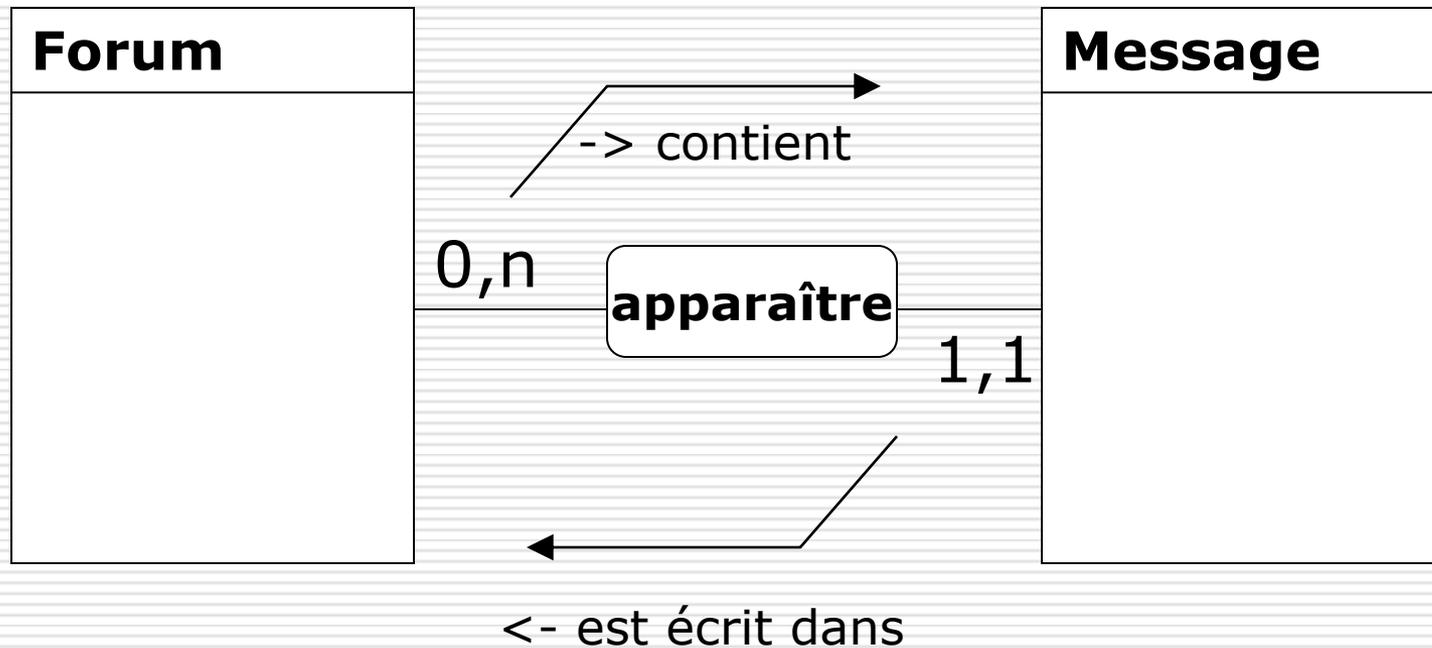
# Identification des cardinalités

---

- Répondre à quatre questions :
    - Une occurrence de A peut être en relation avec une occurrence de B
      - combien de fois au minimum ?
      - combien de fois au maximum?
  
    - Une occurrence de B peut être en relation avec une occurrence de A
      - combien de fois au minimum ?
      - combien de fois au maximum?
-

# Matérialisation des associations et des cardinalités

---



# Vérification et validation du modèle

---

- Les entités et les propriétés doivent vérifier
    - L'intégrité sur les entités
    - L'intégrité référentielle
    - Chaque propriété doit être **élémentaire**
    - Chaque information ne doit apparaître **une seule fois** dans une entité donnée.
    - Chaque propriété doit prendre **une et une seule valeur** pour une occurrence donnée
-

# Normalisation

---

## □ But

- Rendre le modèle le « plus propre possible ».
  - Limiter la redondance de données.
- => Augmenter la fiabilité et diminuer la maintenance.

## □ Les règles de normalisation

- Semi-formelles (intuitives) sur le MCD
  - Formelles sur le MLD (séance prochaine)
-

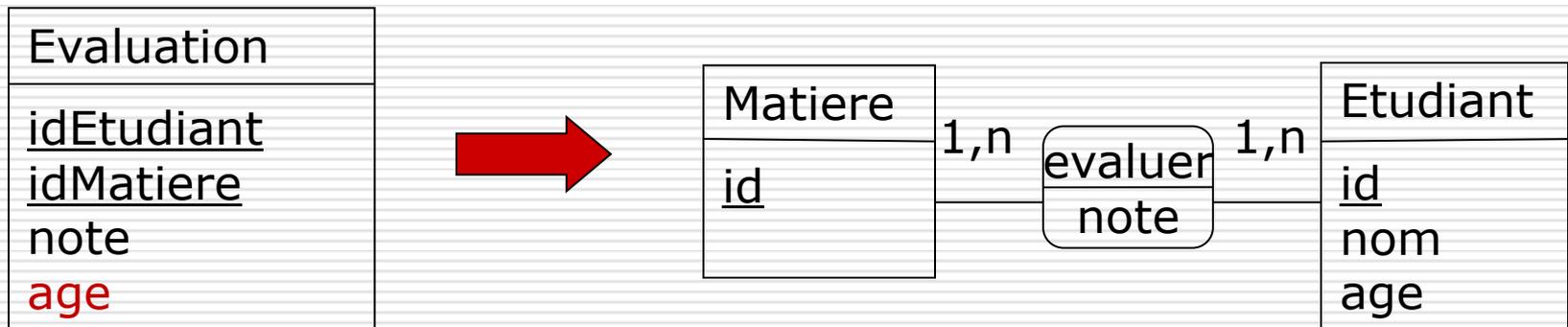
# Propriété élémentaire

---



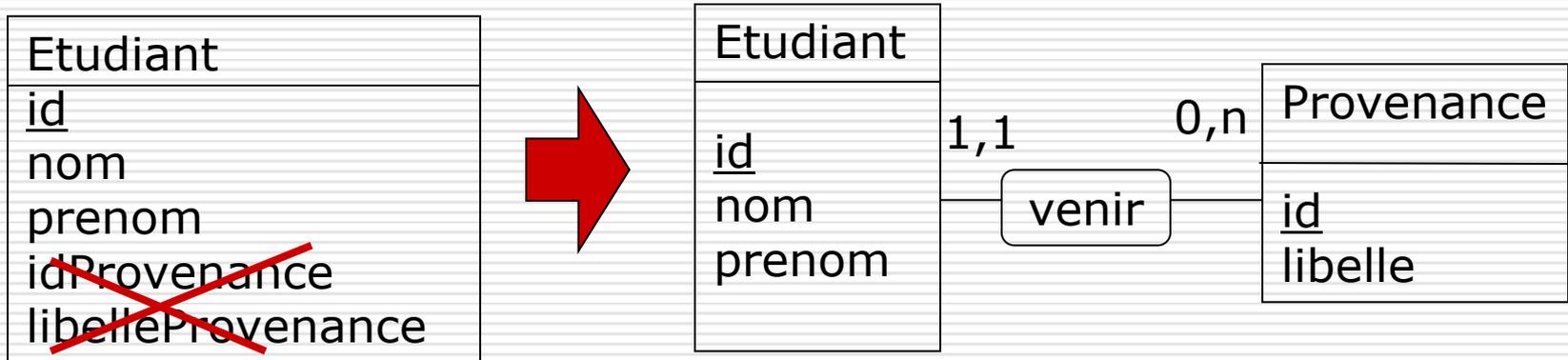
# Intégrité référentielle

---



# Intégrité référentielle

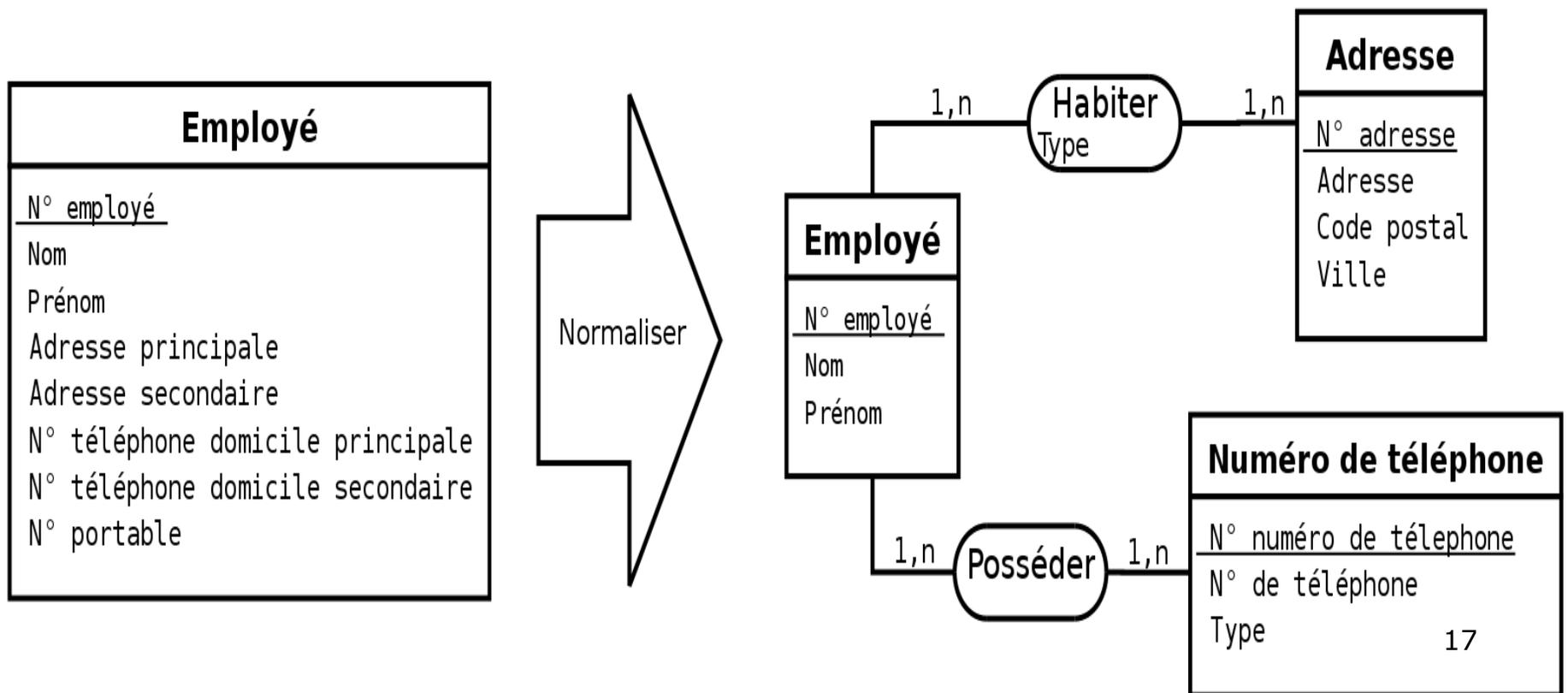
---



libelleProvenance ne dépend pas  
directement de la clé mais d'un autre attribut

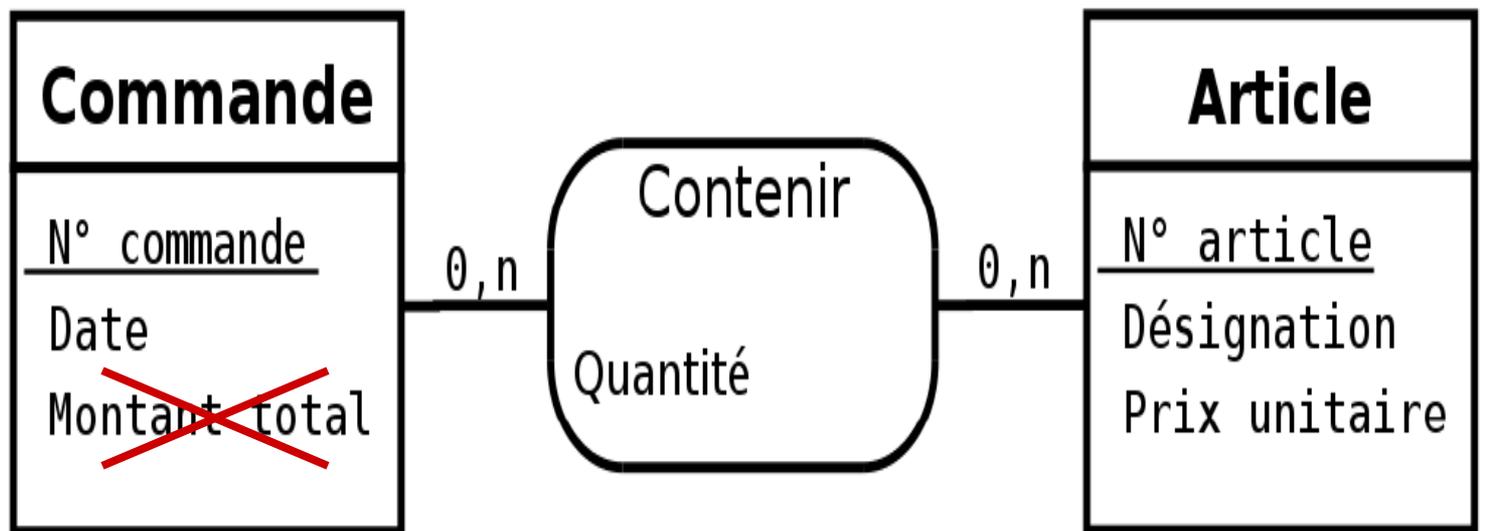
# Attribut multiple

- Remplacer un attribut multiple en association et entité supplémentaire



# Attribut dérivé

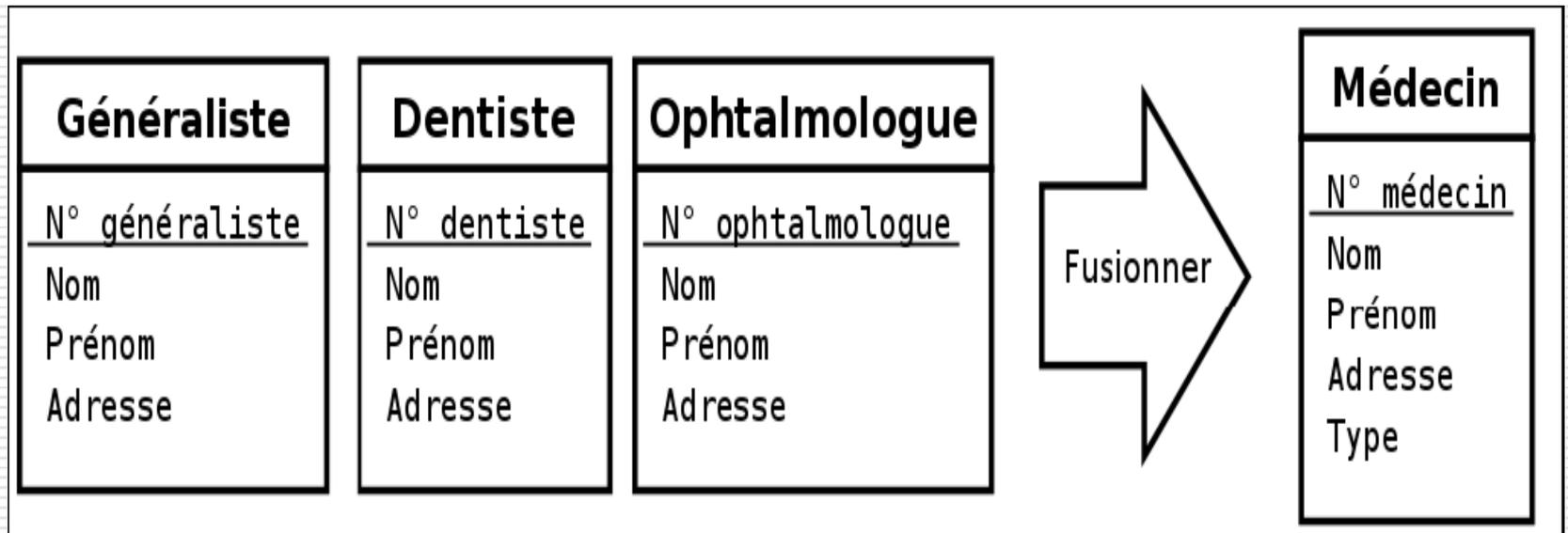
- ❑ Il est déconseillé d'ajouter un attribut dérivé d'autres attributs



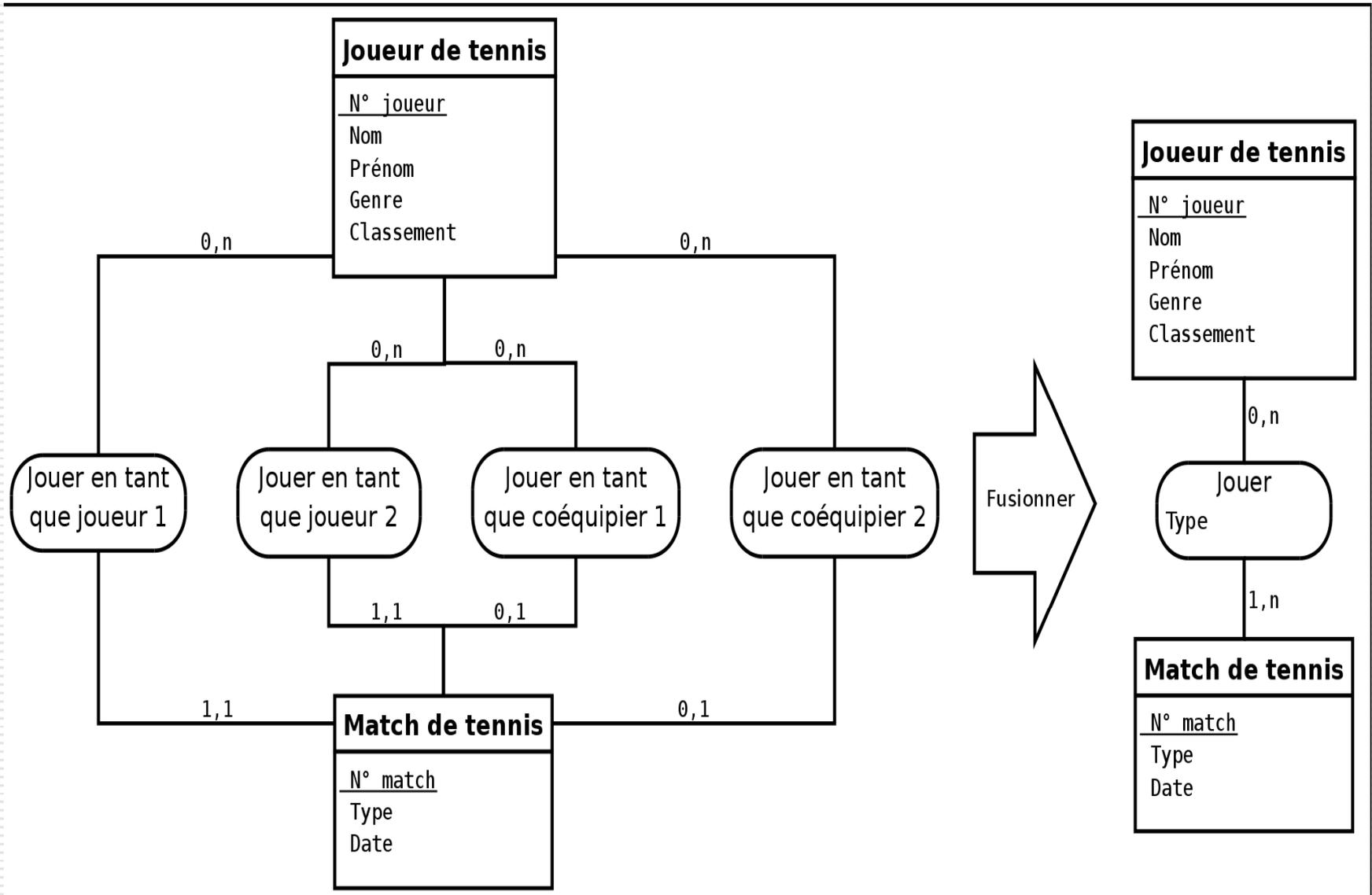
# Fusion/suppression d'entités/associations (1)

---

- Il faut factoriser les entités/associations quand c'est possible

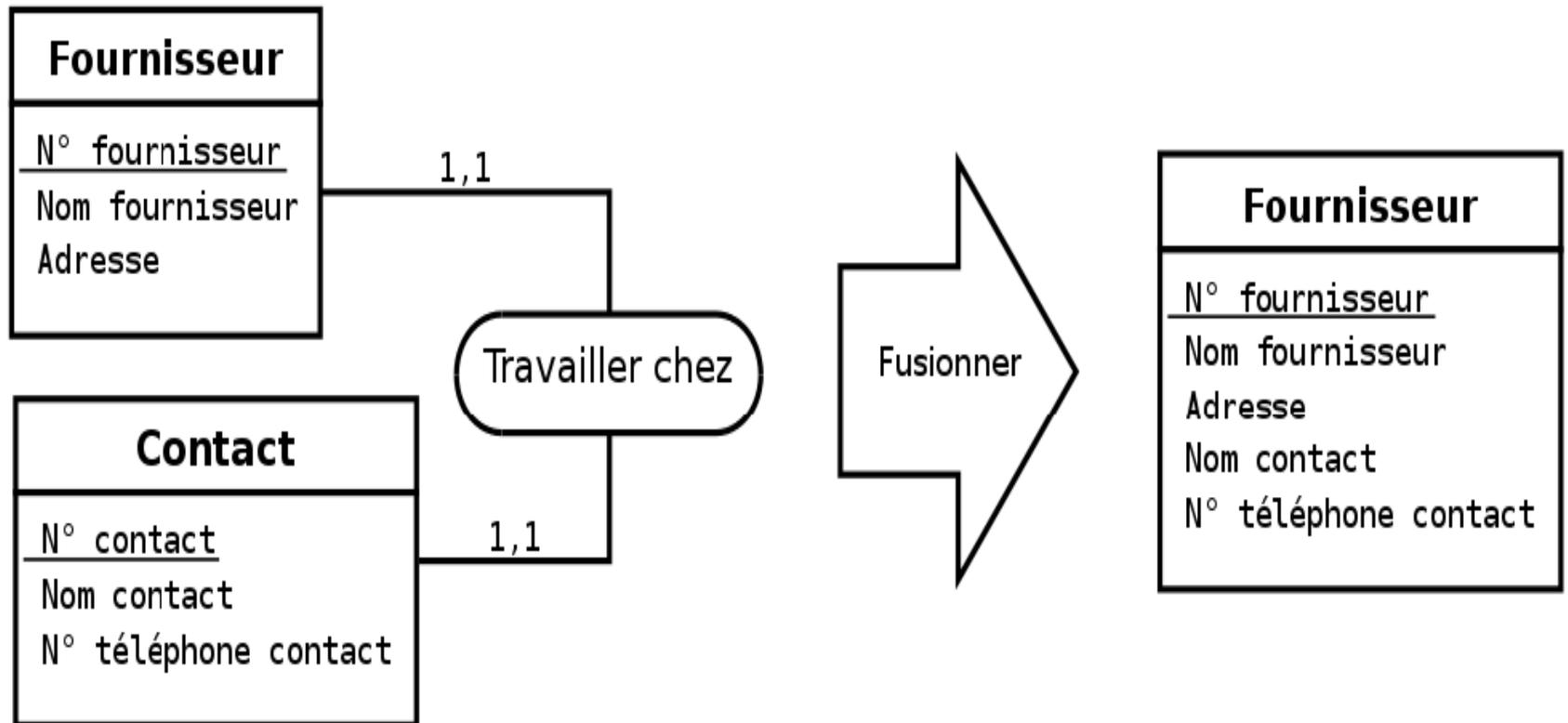


# Fusion/suppression d'entités/associations (2)



# Fusion/suppression d'entités/associations (3)

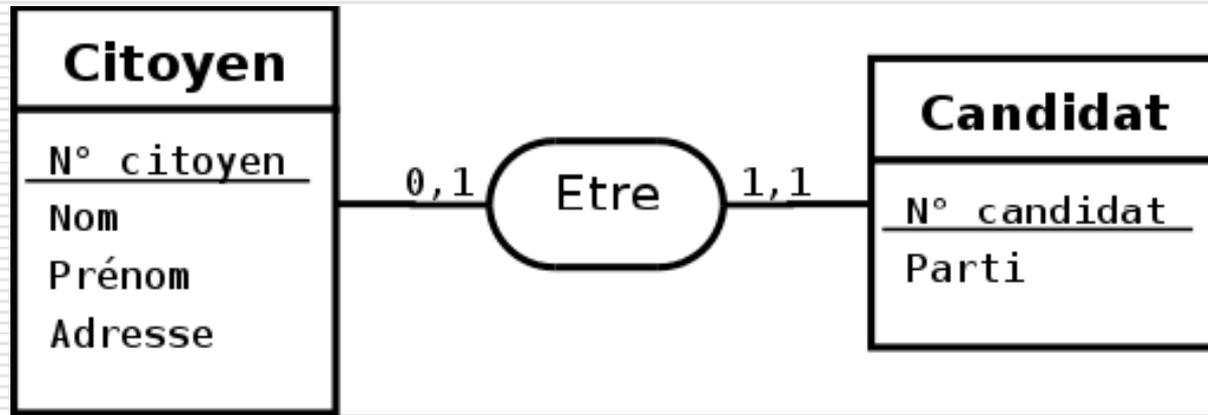
- Lorsque les cardinalités d'une association sont toutes 1,1 c'est que l'association n'a pas lieu d'être



# Fusion/suppression d'entités/associations (4)

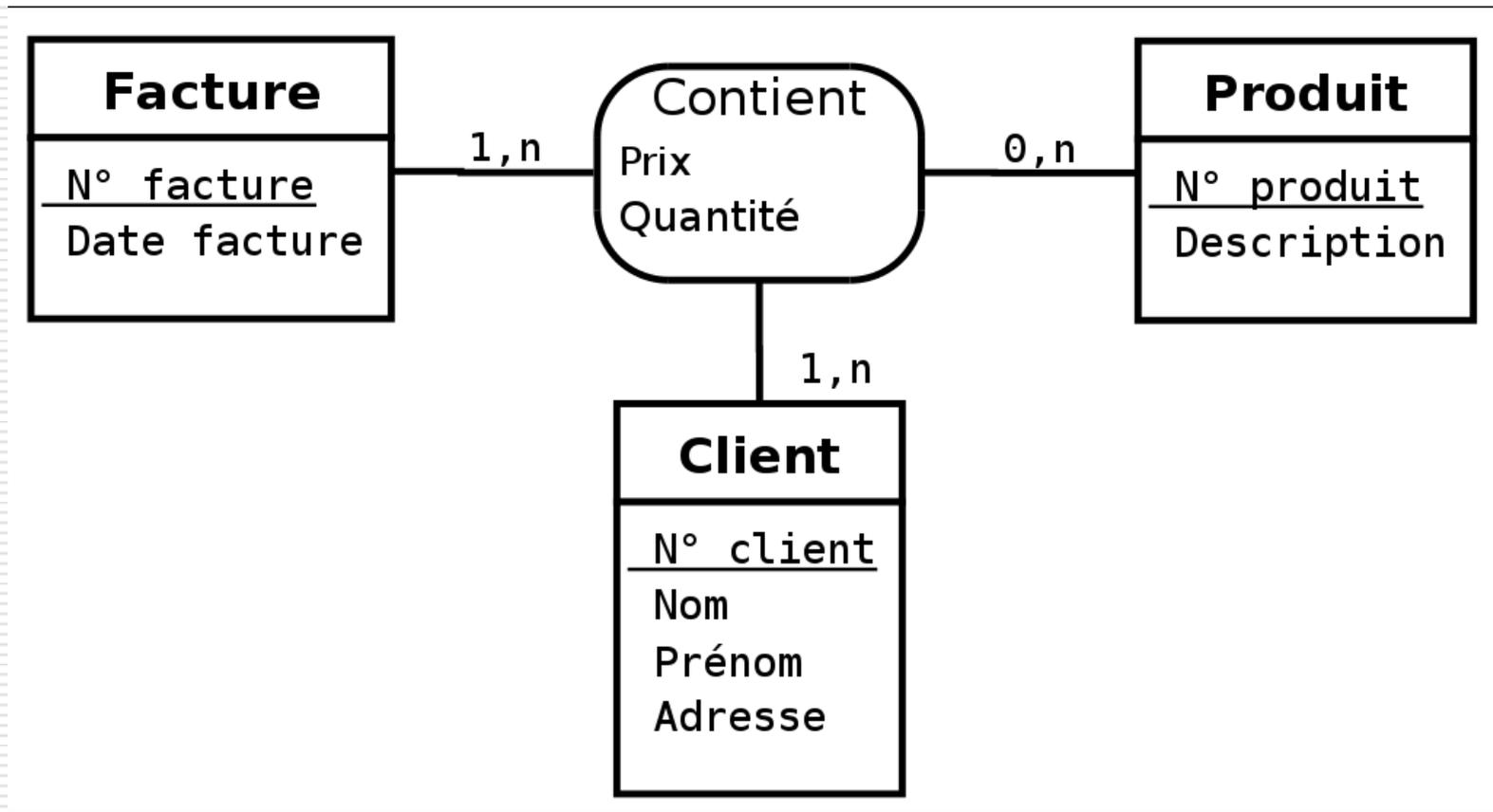
---

- Mais pas cela ...

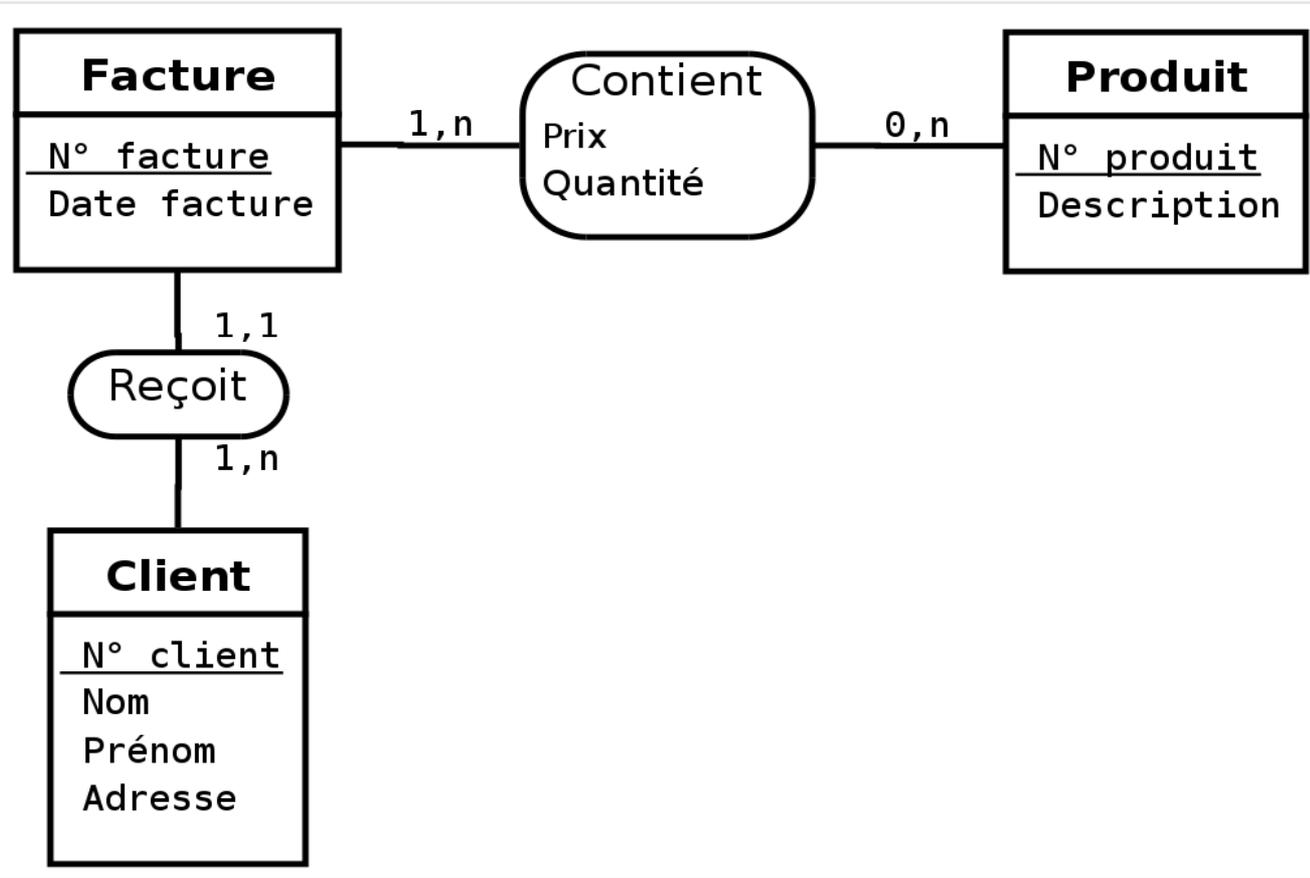


# Association n-aire inappropriée

---

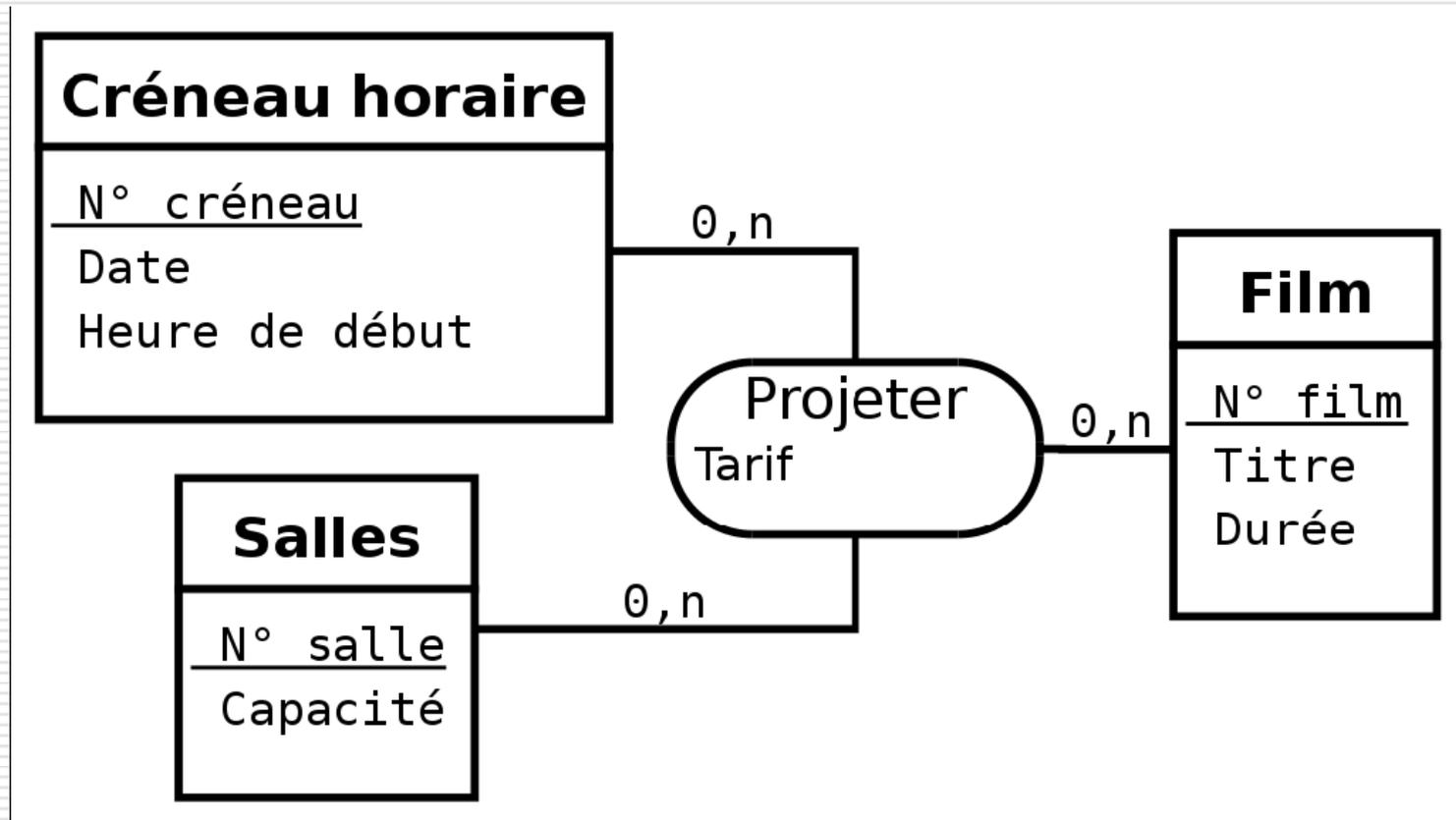


# Corrigée en 2 associations binaires

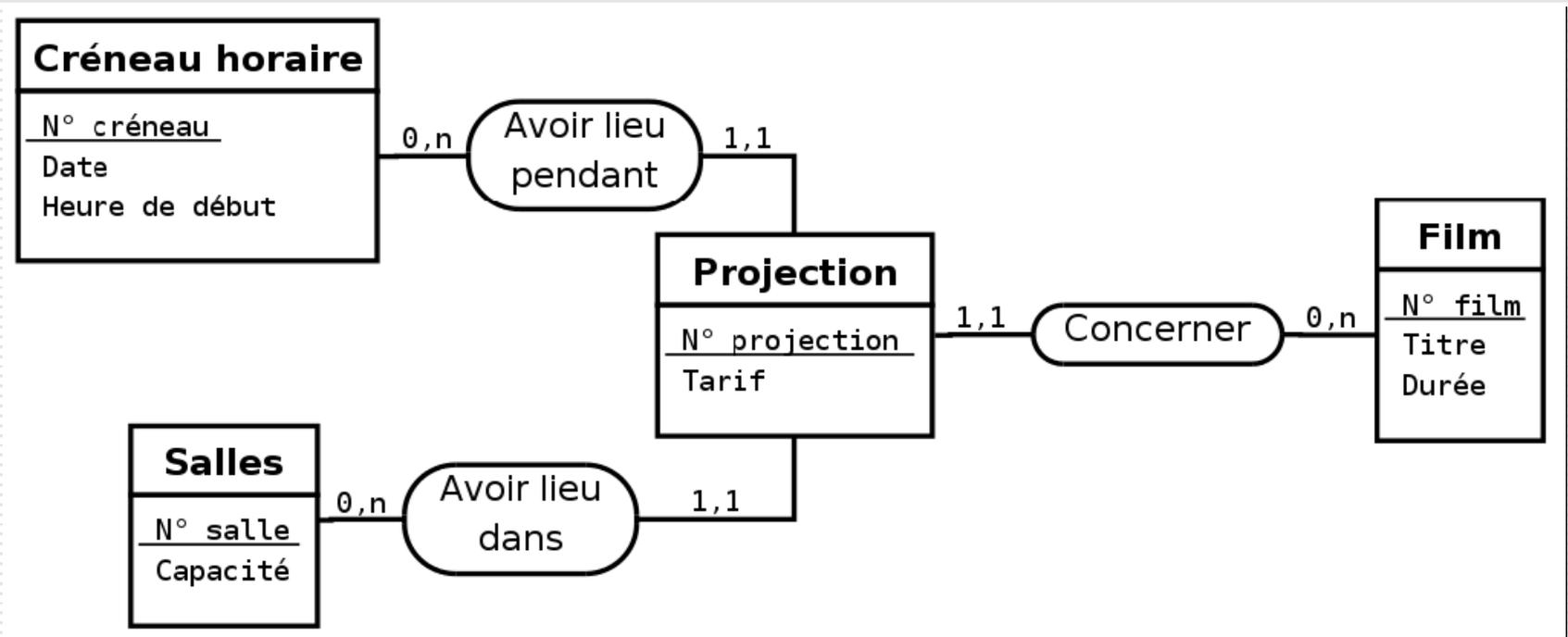


# Décomposition d'une association n-aire

---



# Transformation en 1 entité et 3 associations binaires



# Extension du formalisme entités

## – associations

---

- But : de nouveaux opérateurs pour enrichir la modélisation et symboliser des situations plus complexes
  
  - Introduction des concepts :
    - généralisation/spécialisation
      - représentation plus proche du monde réel
    - de nouvelles contraintes
      - inclusion, exclusion, ou exclusif, ...
-

# Du MCD à la base de données

---

Modèle conceptuel  
de données  
(MCD)

Diagramme  
de classes

On peut appliquer  
des règles de  
conversion

Modèle logique de données  
(MLD)

Transcription en langage SQL

Script de création des tables  
(BDD)

---

# Modèle logique de données (MLD)

---

- Reprend le contenu du MCD mais précise la structure et l'organisation des données telle qu'elles pourront être implémentées :
    - fichiers simples
    - **modèle relationnel**, ...
  
  - Modèle relationnel :
    - Schéma de relation :  
Entite(id, propriete1, propriete2, #aid)
    - Relation : clé primaire, attributs, #clé étrangère
-

# Définitions

---

## □ Clé candidate

- Une clé candidate d'une relation est un ensemble minimal des attributs de la relation dont les valeurs identifient à coup sûr une occurrence.

## □ Clé primaire

- La clé primaire d'une relation est une de ses clés candidates.
- Pour signaler la clé primaire, ses attributs sont généralement soulignés.

## □ Clé étrangère

- Une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui constituent une clé primaire dans une autre relation.
-

# Règles de passage de MCD à MLD

---

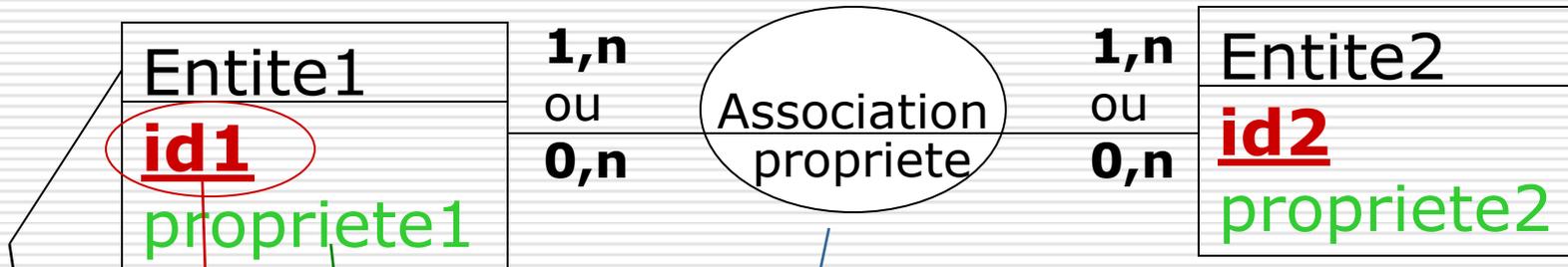
- Règle 1 : Toute entité est représentée par une relation. Chaque attribut de l'entité devient un attribut de la relation. L'identifiant est conservé en tant que clé de la relation.
  - Règle 2 : Toute association qui associe plus de deux entités (ternaire et au-delà) est représentée par une relation.
-

# Règles de passage de MCD à MLD

---

- Règle 3 : Toute association binaire dont les cardinalités maximales sont  $n$  de chaque côté est une relation (relation dont les attributs sont les attributs clefs des entités qu'elle relie ainsi que les éventuels attributs propres à l'association).
  - Règle 4 : Une association de type père - fils, cardinalité maximum à  $n$  d'un côté et à  $1$  de l'autre, n'est pas représentée par une relation. On indique les attributs clefs de l'entité père (côté  $(.,n)$ ) dans le fils (côté  $(.,1)$ ).
-

# Relation matricielle ou n..n



- Chaque entité devient une relation
- Les **identifiants de chaque entité** sont les **clés primaires**
- Les **attributs de chaque relation** sont les **propriétés des entités**
- L'association se traduit en créant une relation Association dont la clé primaire est formée des identifiants de chaque entité participant à l'association- elles sont des clés étrangères

**Entite1(id1, propriete1)**

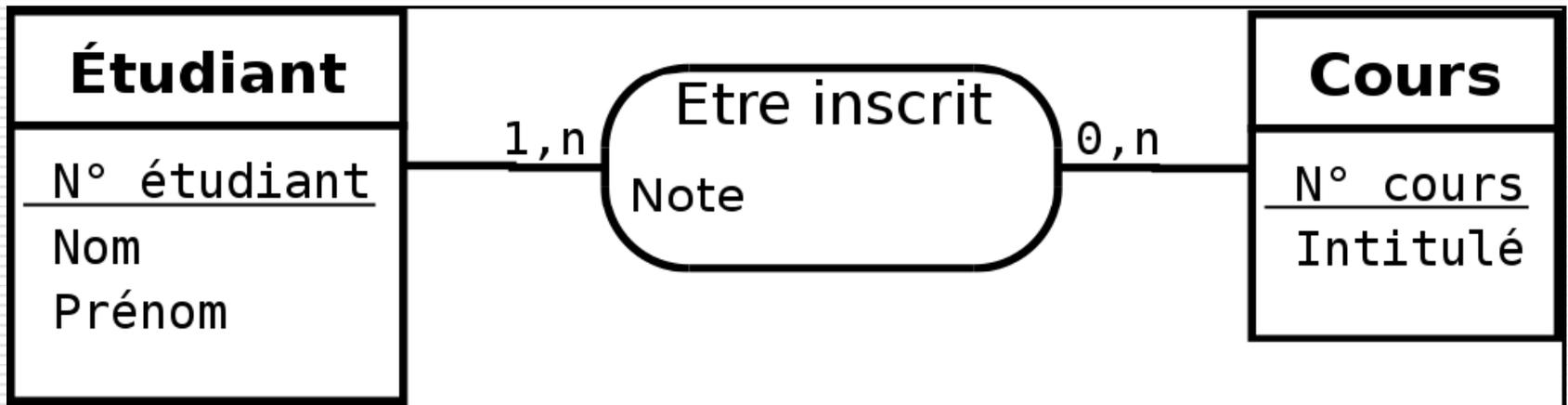
**Entite2(id2, propriete2)**

**Association(#id1, #id2, propriete)**

# Exemple 1

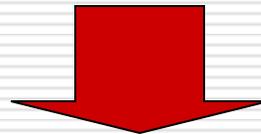
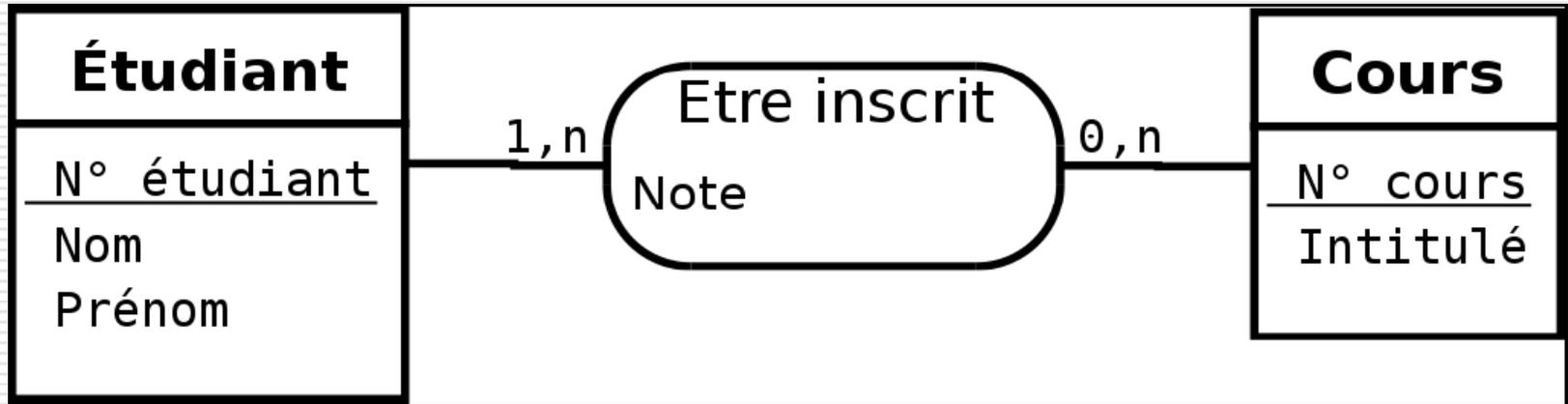
---

- ❑ Un étudiant s'est inscrit en 1 ou plusieurs cours
- ❑ Un cours peut être inscrit par zéro ou plusieurs étudiants
- ❑ Pour chaque cours, un étudiant est évalué par une note



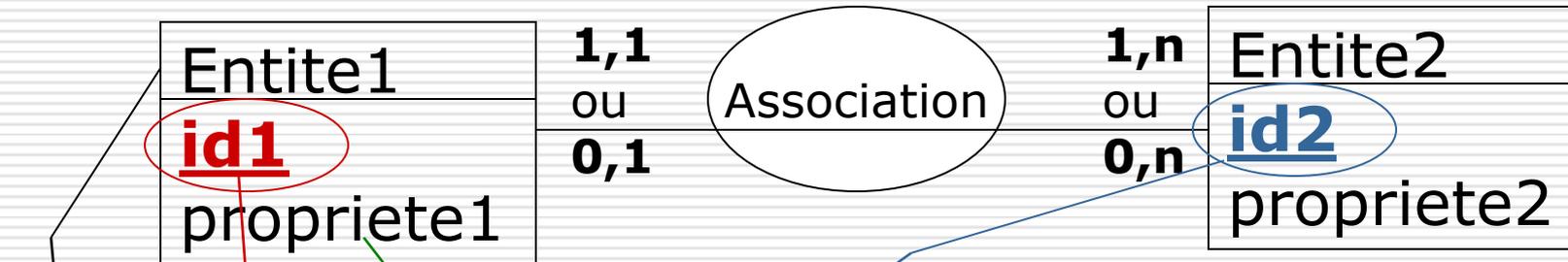
## Exemple 1 : Traduction du MCD vers MLD

---



- ❑ Etudiant(numEtudiant, nom, prenom)
  - ❑ Cours(numCours, intitule)
  - ❑ Inscription(#numEtudiant, #numCours, note)
-

# Relation père-fils



Entite1(id1, #id2, propriete1)

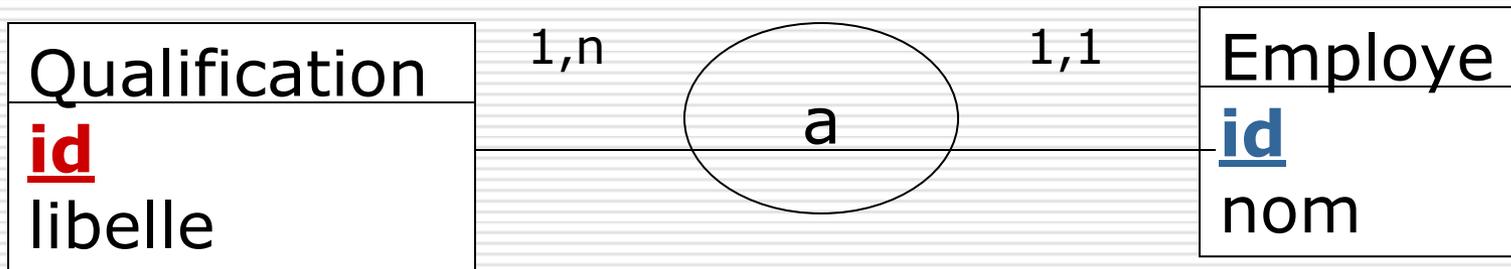
Entite2(id2, propriete2)

- Chaque entité devient une relation
- Les identifiants de chaque entité sont les clés primaires
- Les attributs de chaque relation sont les propriétés des entités
- L'association se traduit en rajoutant dans la relation Entite1 l'attribut correspondant à l'identifiant de Entite2, id2 devient donc clé étrangère

# Exemple 2

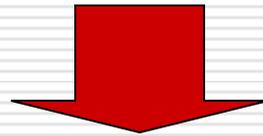
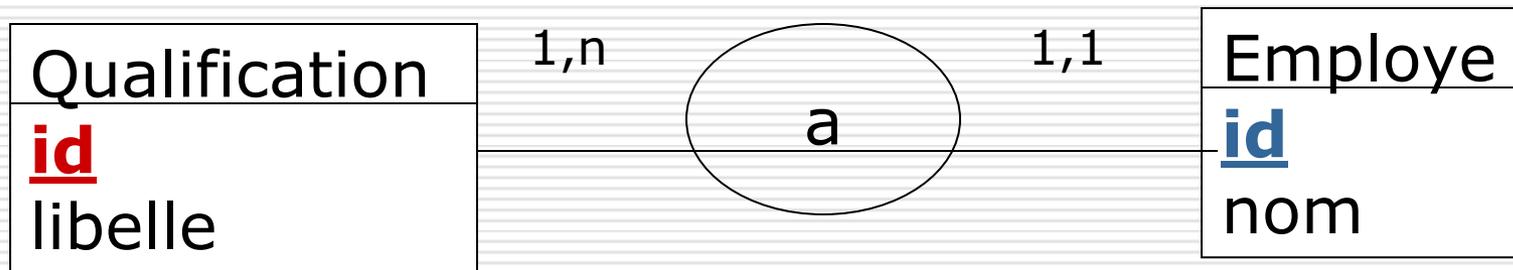
---

- Un employé a une qualification
- Une qualification peut correspondre à plusieurs employés



## Exemple 2 : Traduction du MCD vers MLD

---

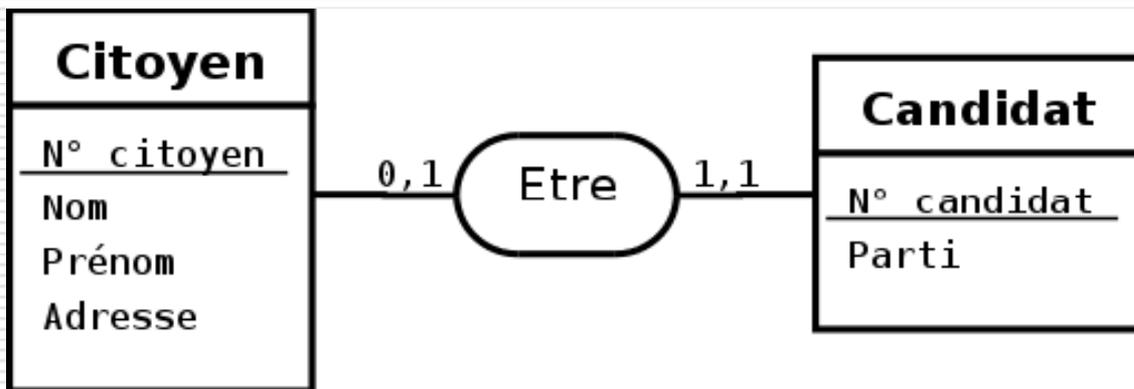


- `Employe(id, nom, #id_qualif)`
  - `Qualification(id, libelle)`
-

# Cas particuliers (1)

---

- Association (0,1) – (1,1) : on déclare une clé étrangère du côté (1,1) pour éviter des champs NULL



Citoyen(numCitoyen, nom, prenom, adresse)  
Candidat(numCandidat, parti, #numCitoyen)

- Association (0,1) – (0,1) : on a le choix entre les deux relations pour placer la clé étrangère
-

## Cas particuliers (2)

---

- Association réflexive : mêmes règles qu'une association binaire (avec renommage de la clé étrangère dans le cas d'une association père-fils)
  - Exceptions : on pourra créer une relation supplémentaire pour une association (0,1)-(1,n) suivant la proportion d'occurrences de celle-ci, pour éviter une quantité trop importante de champs NULL
-

# Problème classique : *étudiants, cours, enseignants, salles ...*

---

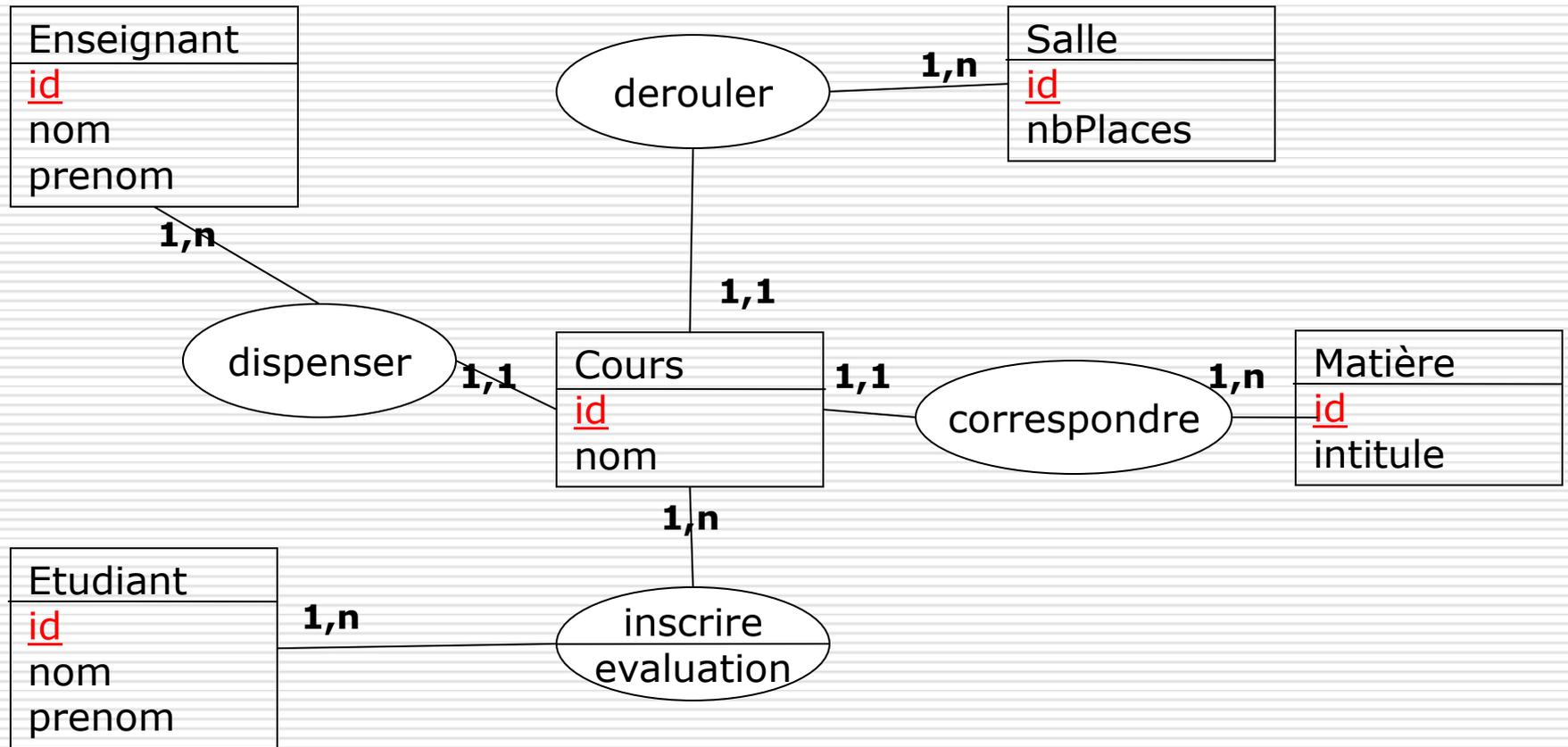
1. Il existe plusieurs matières (mathématiques, français, philosophie, ...).
  2. Plusieurs cours sont offerts et il peut y avoir plusieurs cours de la même matière.
  3. Un cours est dispensé par un, et un seul, enseignant et correspond à une matière.
  4. Un enseignant peut dispenser plusieurs cours dans la même matière ou dans des matières différentes.
  5. Un étudiant peut s'inscrire à plusieurs cours.
  6. Un cours est toujours dispensé dans une même salle, mais une salle peut recevoir plusieurs cours (successivement).
  7. Chaque cours ne comporte qu'une seule évaluation.
-

# Exercice !

---

- Proposez un MCD permettant de modéliser la situation décrite ci-dessus !
-

# MCD !



# Exercice !

---

- Proposez un MLD à partir du MCD précédent !
-