

BDay-MI

Bases de données avancées

Cristina Sirangelo

IRIF, Université Paris Diderot

[cristina@liafa.univ-paris-diderot.fr](mailto:cristina@liafa.univ-paris-diderot.fr)

# Modélisation conceptuelle : le modèle Entité/Association (E/R)

Sources (quelques slides empruntés et réadaptés) :

- *DB Systems concepts* , A. Silberschatz, Yale U. & H. Korth, Lehigh U. & S.Sudarshan, IIT Bombay

# Modele E/R (*Entity/Relationship* en anglais)

- Un formalisme pour la modélisation conceptuelle des données:
  - ▶ quelles données
  - ▶ comment elles sont reliées
  - ▶ quelle contraintes elle satisfont
- Introduit par Chen :
  - ▶ *Peter P.-S. Chen. The Entity-Relationship Model, Toward a Unified View of Data, ACM Transactions on Database Systems (TODS) 1:(1), 1976*
- La plupart des modèles conceptuels utilisés encore aujourd'hui dans l'industrie s'inspirent du modele E/R
- Plusieurs variantes dans la notation : on fixera une notation, beaucoup de notations alternatives existent !

# Modélisation des données avec E/R

- Les données peuvent être modélisées comme :
  - ▶ un ensemble d'*entités*,
  - ▶ des *associations* entre ces entités

# Entité

- **Entité** : une classe d'objets avec des propriétés communes
  - ▶ Exemples : les étudiants de Paris 7, les Employés d'une entreprise, les spectacles dans Paris,...
- Les entités ont des **attributs** (i.e. leurs propriétés)
  - ▶ Exemple: les étudiants de Paris 7 on un *nom*, un *numero étudiant*, une *adresse*
  - ▶ Les spectacles ont une *description*, un *genre*, un *lieu*, des *horaires*

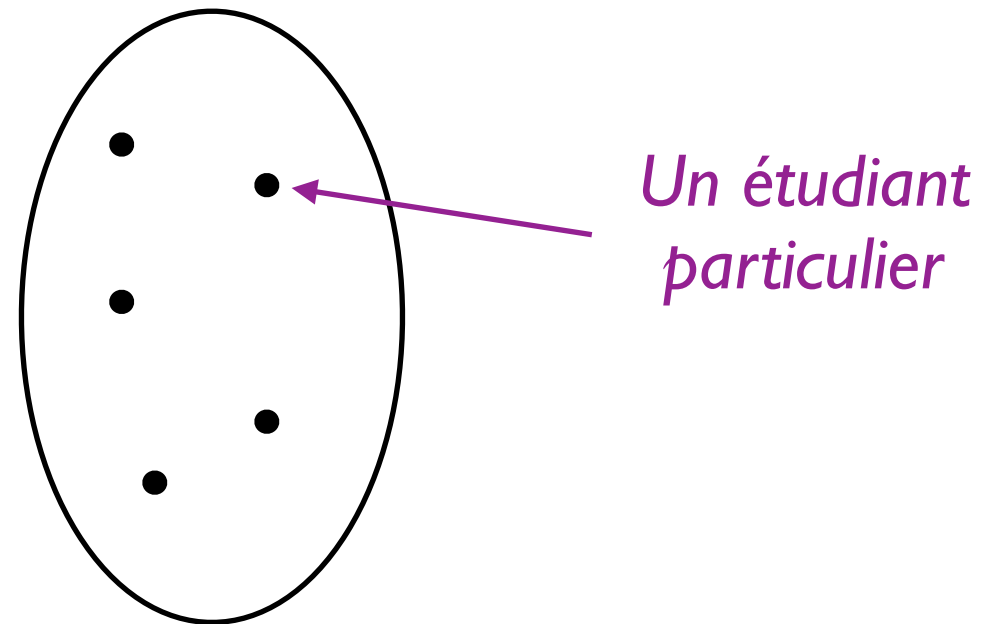
# Entités et attributs

## Syntaxe :

Étudiant
numero étudiant nom adresse

## Sémantique :

- Une **entité** denote un ensemble d'objets, appelés **instances** de l'entité



instances(Étudiant)

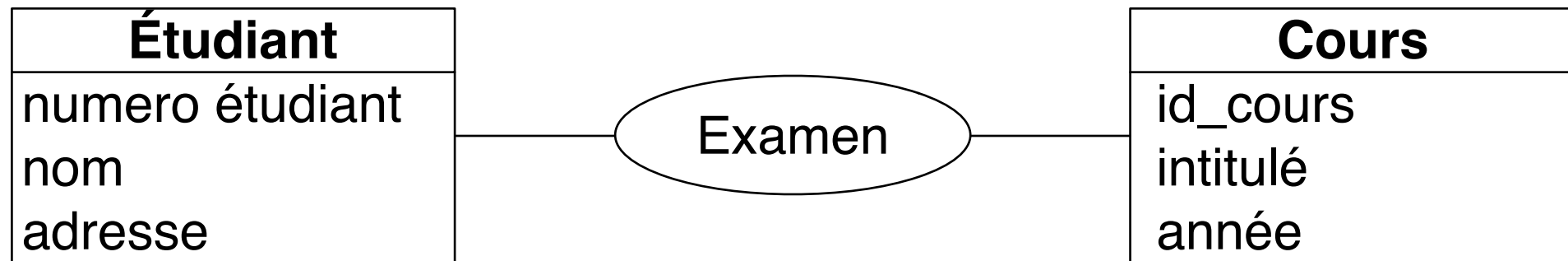
- Un **attribut** denote une **fonction** qui associe à chaque instance de l'entité une valeur dans un domaine
  - Ex.  $\text{nom} : \text{Instances}(\text{Etudiant}) \rightarrow \text{String}$

# Associations

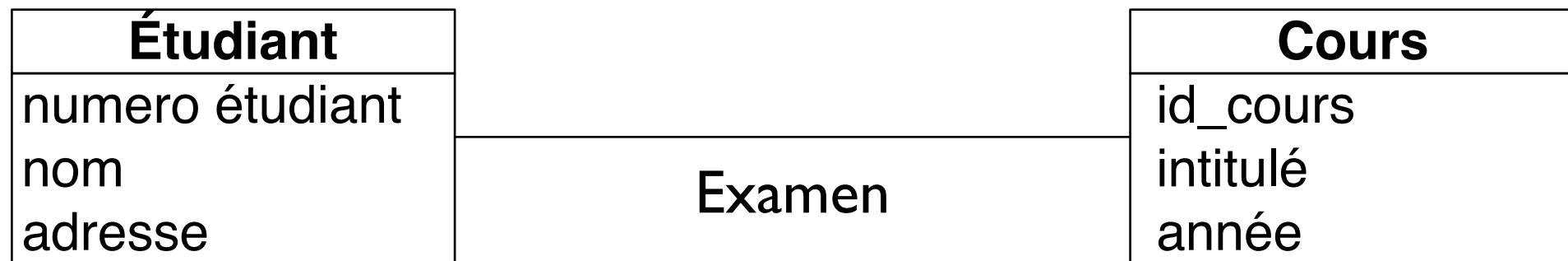
- **Association** : représente un lien entre deux ou plusieurs entités
- Exemples d'associations :
  - ▶ l'association **Examen** relie l'entité **Etudiant** à l'entité **Cours**
  - ▶ l'association **Affectation** relie l'entité **Employé** à l'entité **Département**

# Associations

Syntaxe :



alternative :

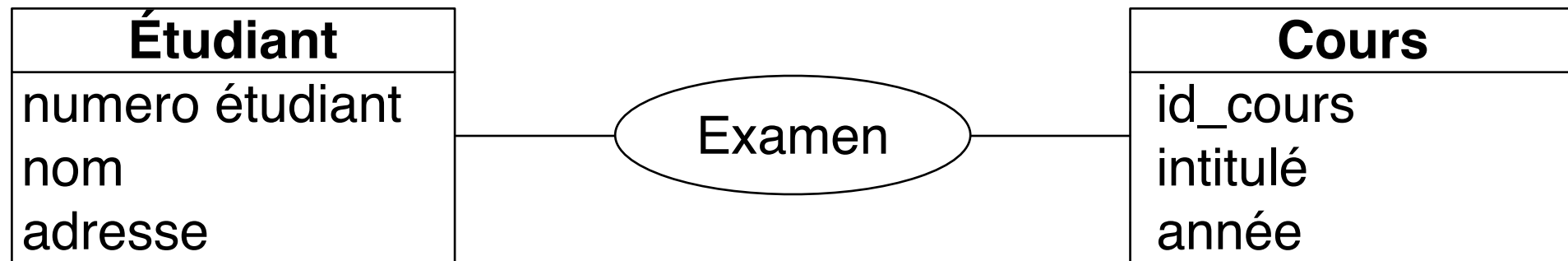


Le nom de l'association peu être omis si il est claire du contexte



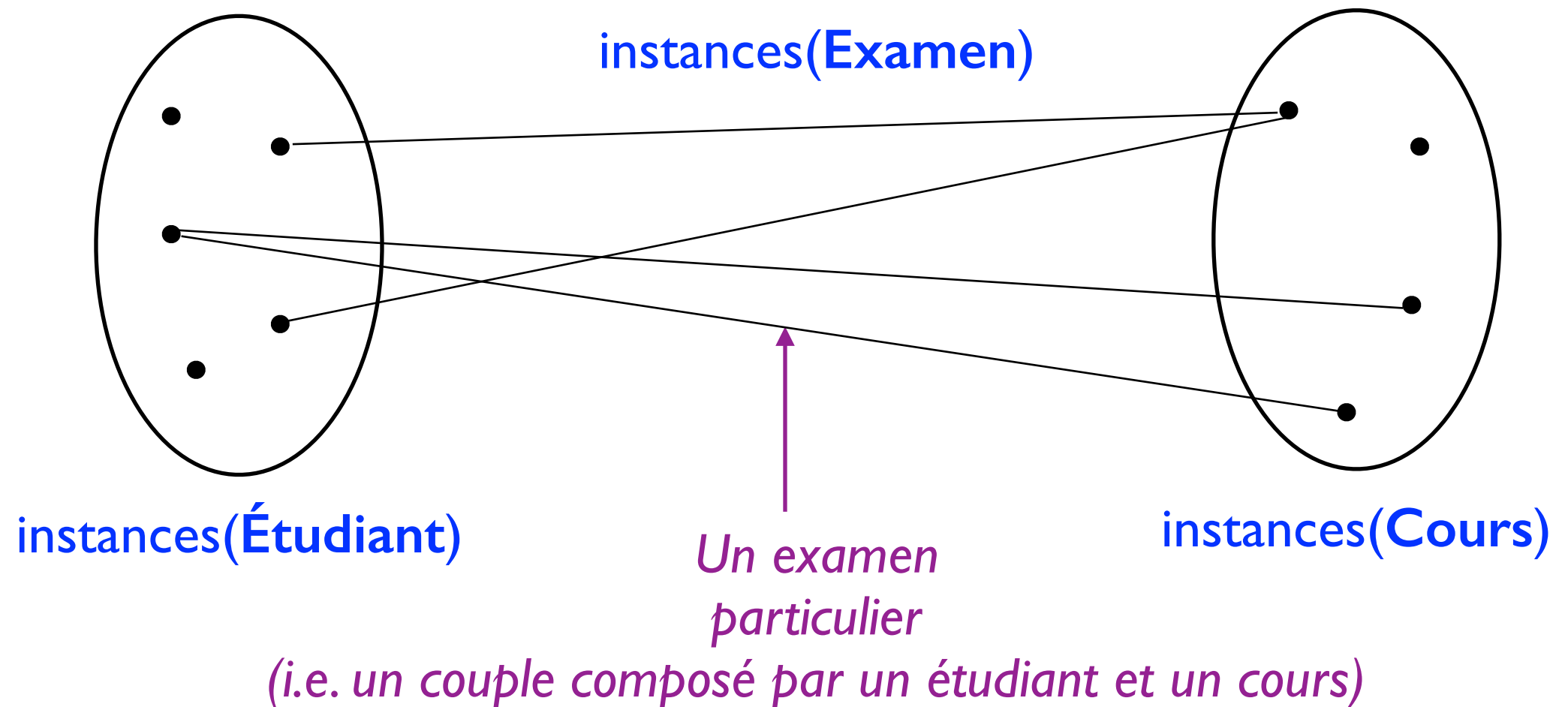
# Associations

Syntaxe :



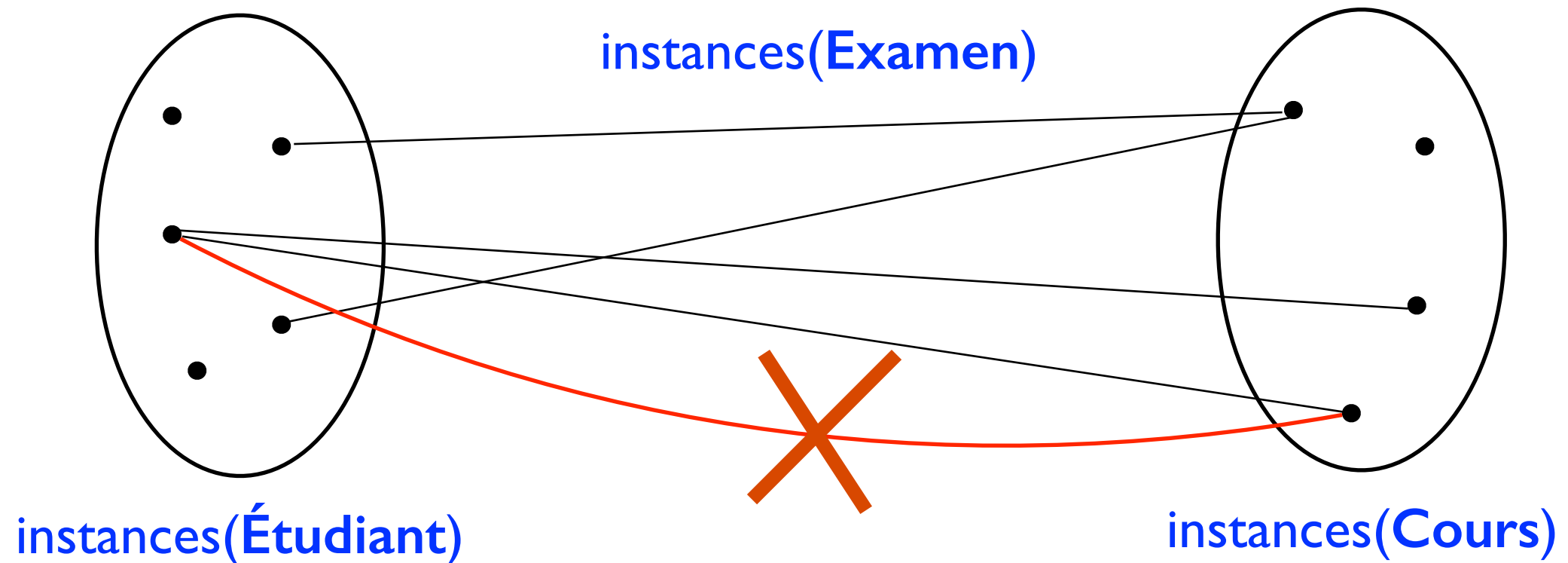
Sémantique :

$\text{instances(Examen)} \subseteq \text{instances(Étudiant)} \times \text{instances(Cours)}$



# Associations

- Remarques sur la sémantique



`instances(Examen)` est un **ensemble** de couples : pas de doublons

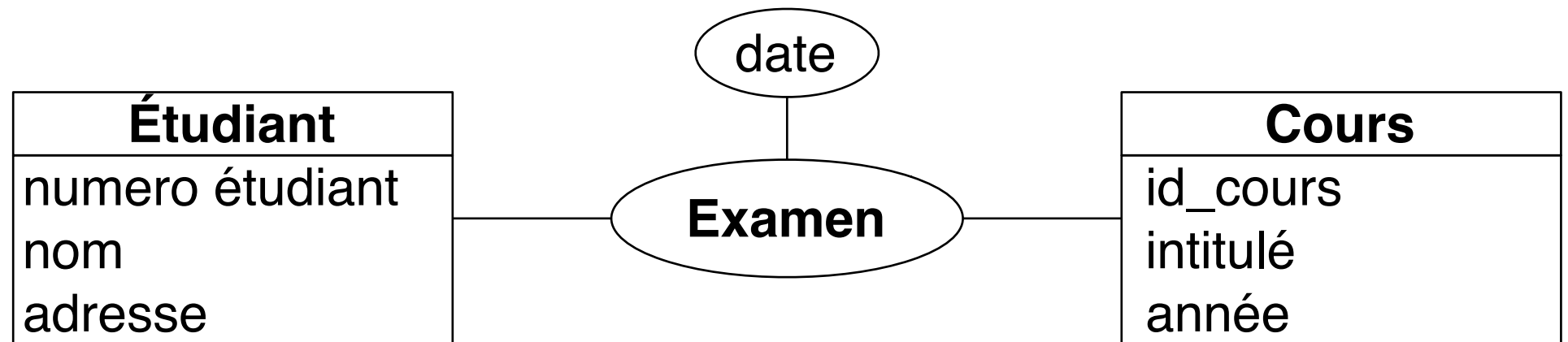
⇒ modéliser Examen comme association introduit automatiquement la contrainte suivante : **un étudiant ne peut pas passer l'examen d'un même cours plusieurs fois**

Si ce n'est pas le cas dans la réalité que nous modélisons (e.g. examens non réussis, à répéter...) il faut changer la modélisation

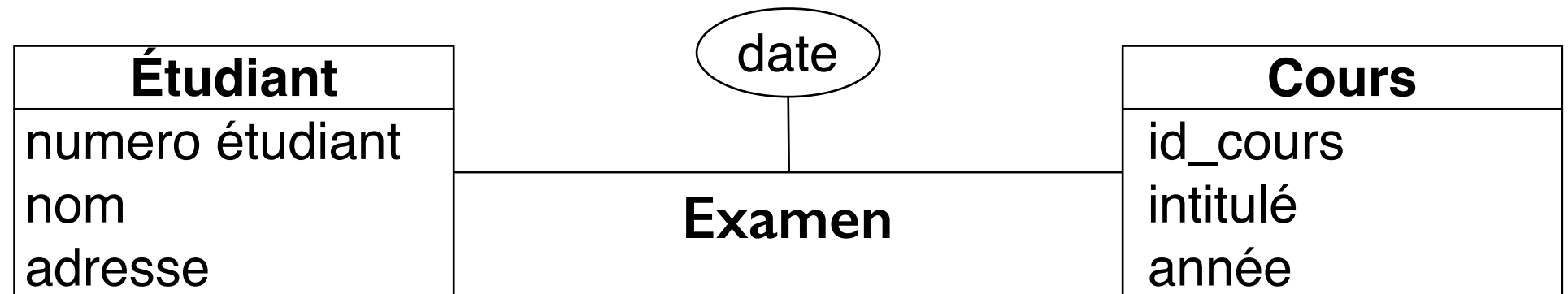
# Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
  - ▶ Exemple : chaque examen a une date

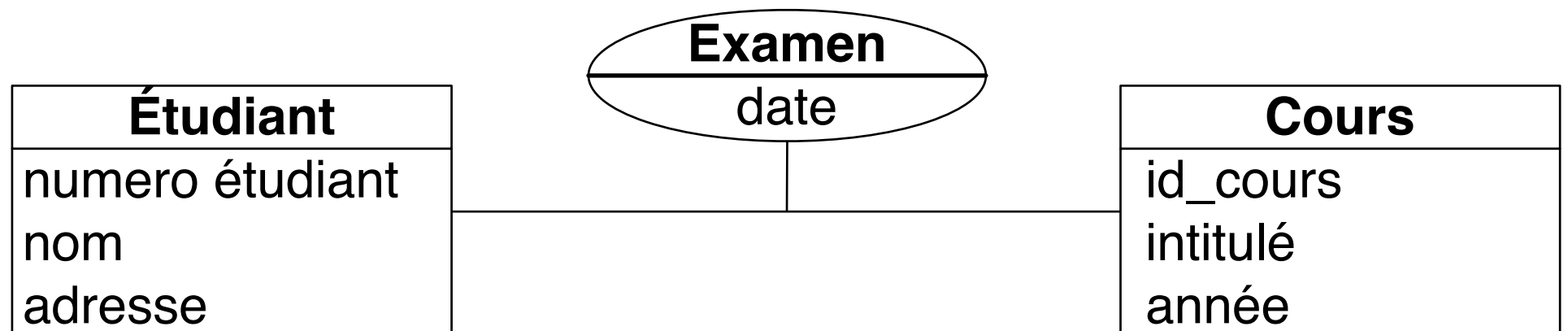
## Syntaxe :



## alternative :



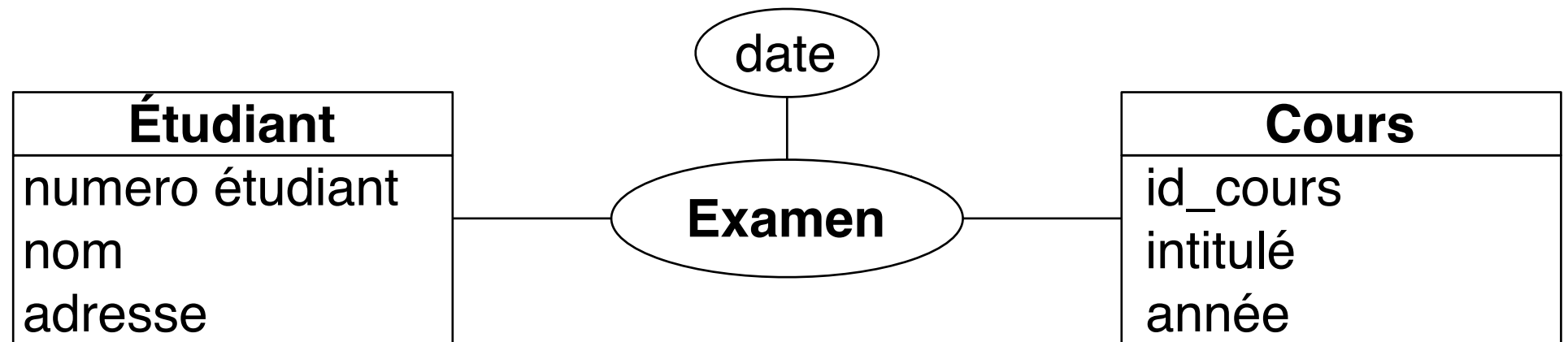
## alternative :



# Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
  - ▶ Exemple : chaque examen a une date

## Syntaxe :



## Sémantique :

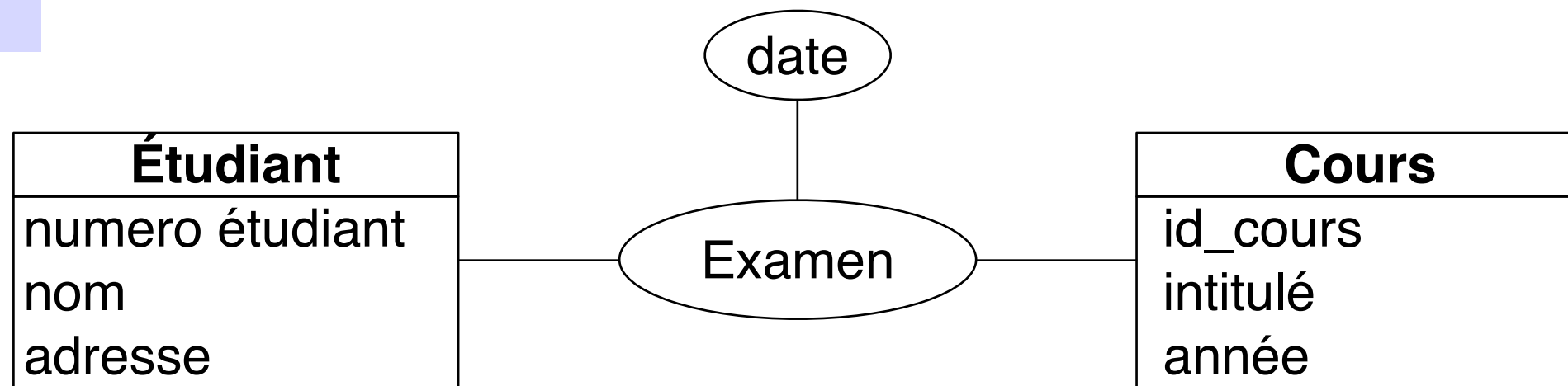
Un **attribut** d'une association denote une **fonction** qui associe à chaque instance de l'association une valeur dans un domaine

- **date : Instances (Examen) → Dates**  
(i.e date est une propriété des couples <étudiant, cours> qui représentent un examen)

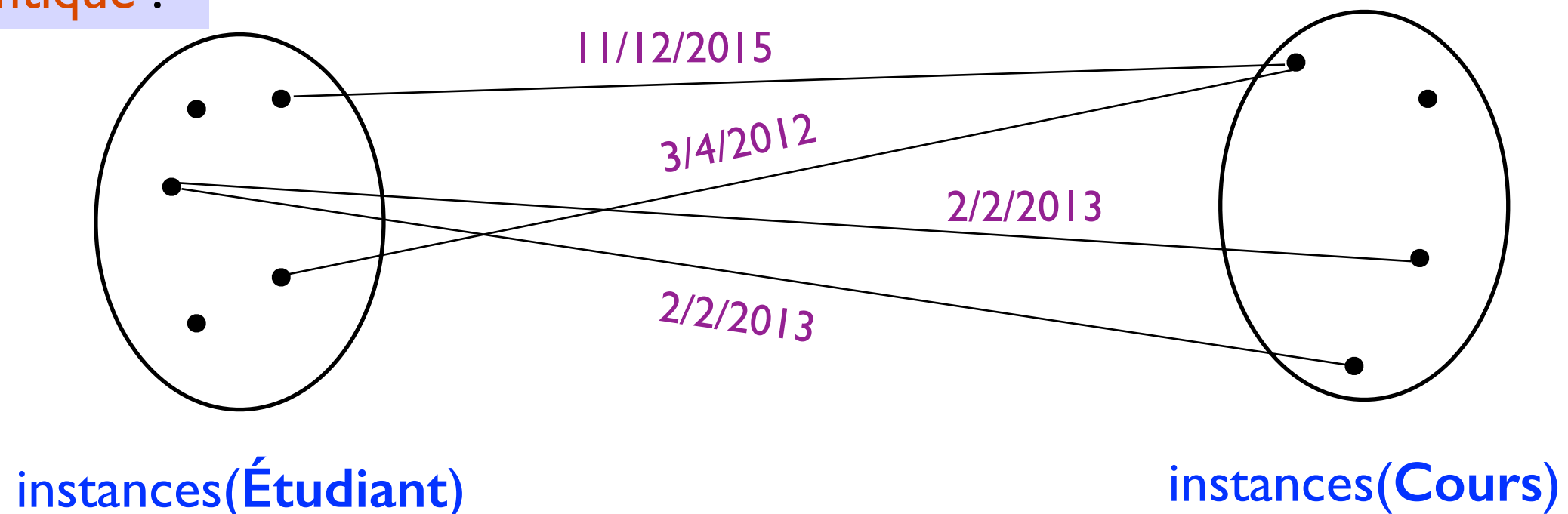
# Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
  - ▶ Exemple : chaque examen a une date

## Syntaxe :

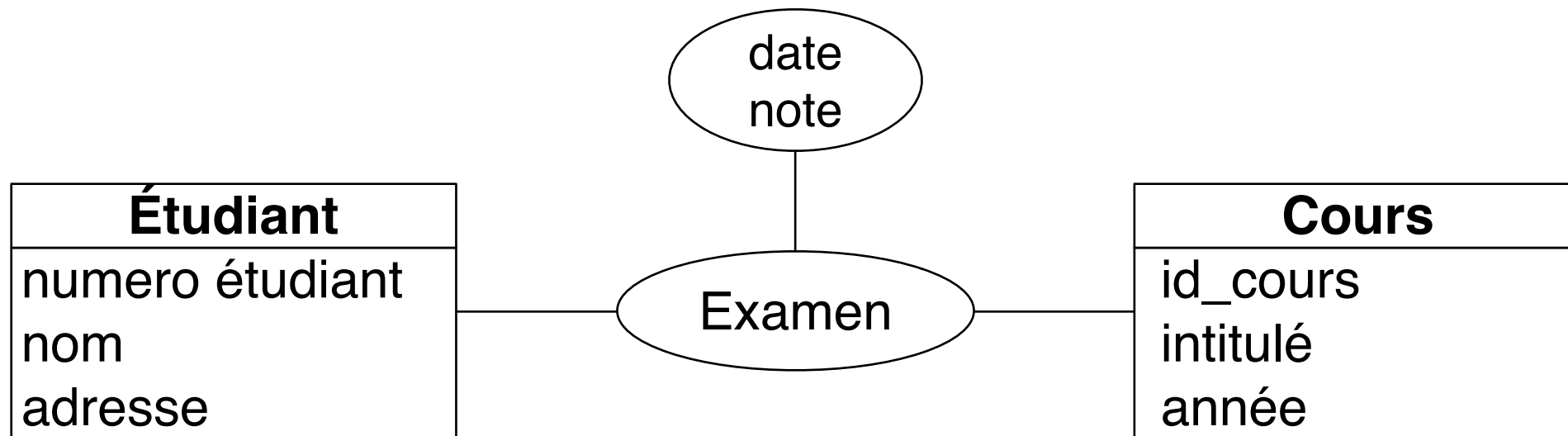


## Sémantique :



# Attributs d'associations

- Plusieurs attributs sont possibles :

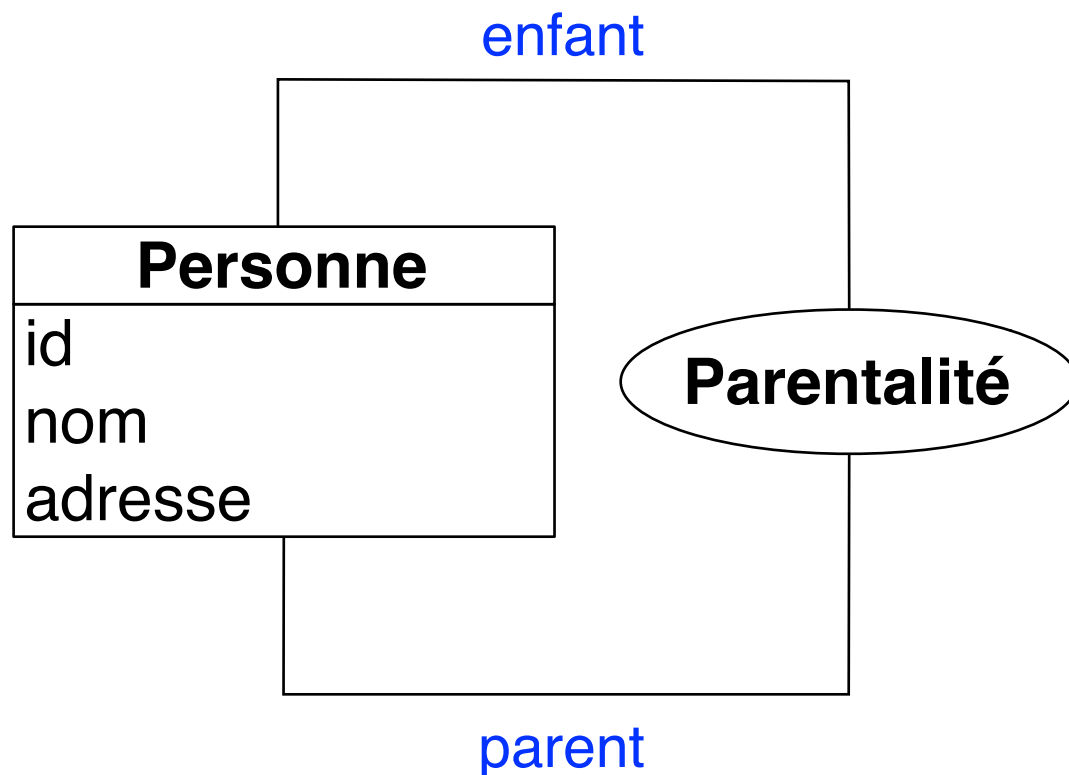


chacun représente une fonction différente

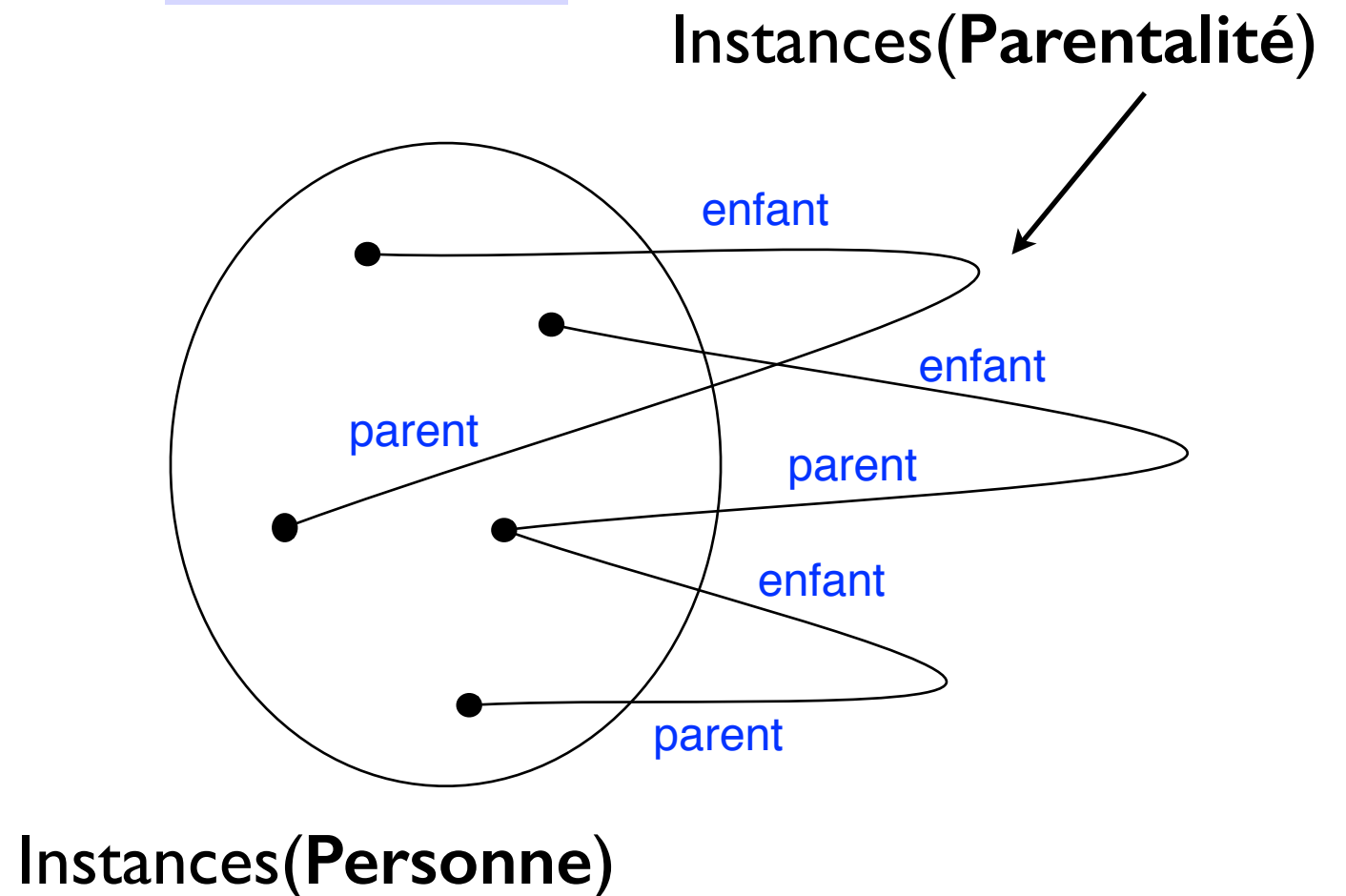
# Associations réflexives

- Une association peut relier un entité à elle même, dans ce cas une étiquette (appelé **rôle**) doit distinguer les deux cotés de l'association :

**Syntaxe :**



**Sémantique :**



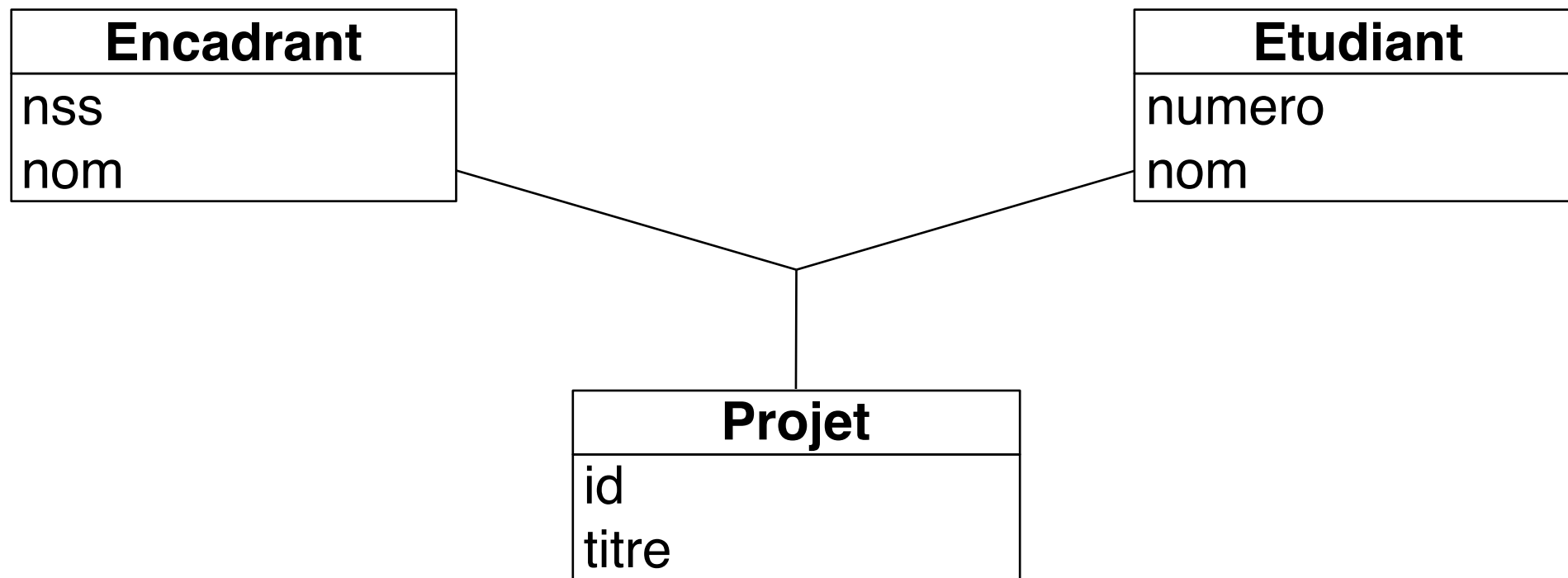
- Sémantique : **Instances(Parentalité)** est un ensemble de couples de personnes étiquettes par les rôles. Dans un chaque couple : une personne a le rôle “parent” et l’autre le rôle “enfant”

## Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Encadrement**

(Les étudiants sont encadrés par des encadrants sur des projets)

**Syntaxe :**



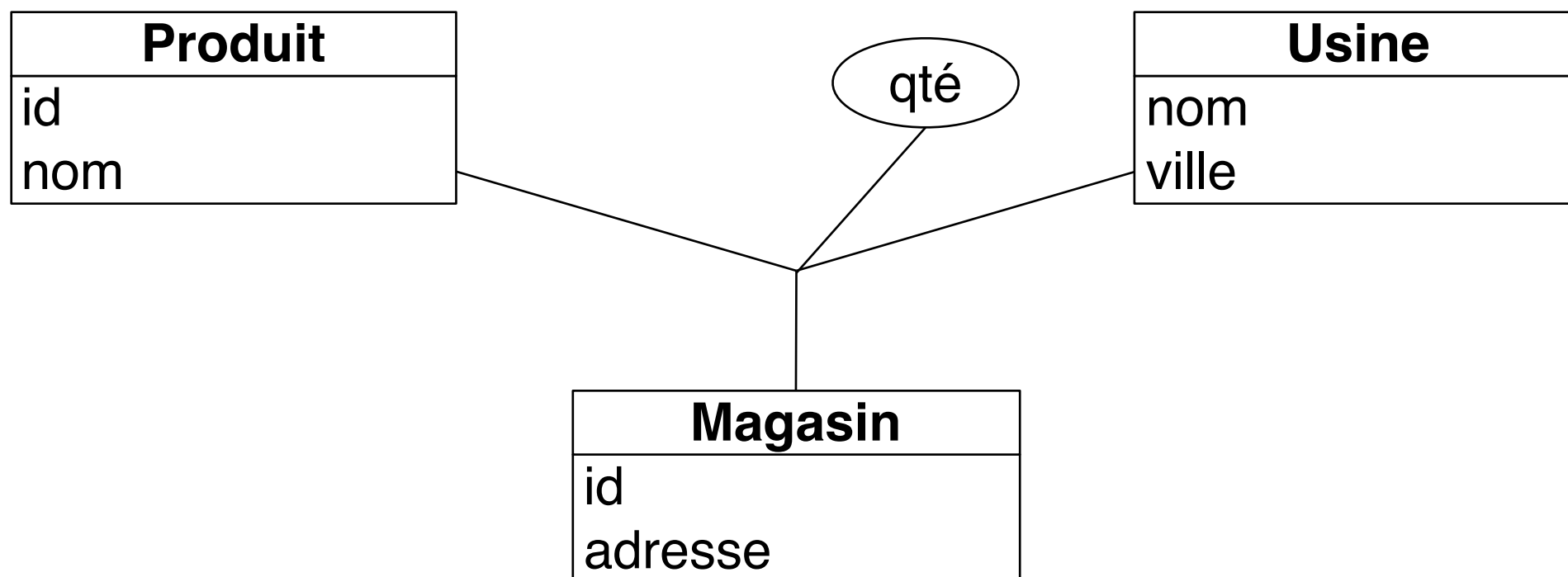


## Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

(Dans chaque magasin il y a une certaine quantité disponible de produits fabriqués dans des usines)

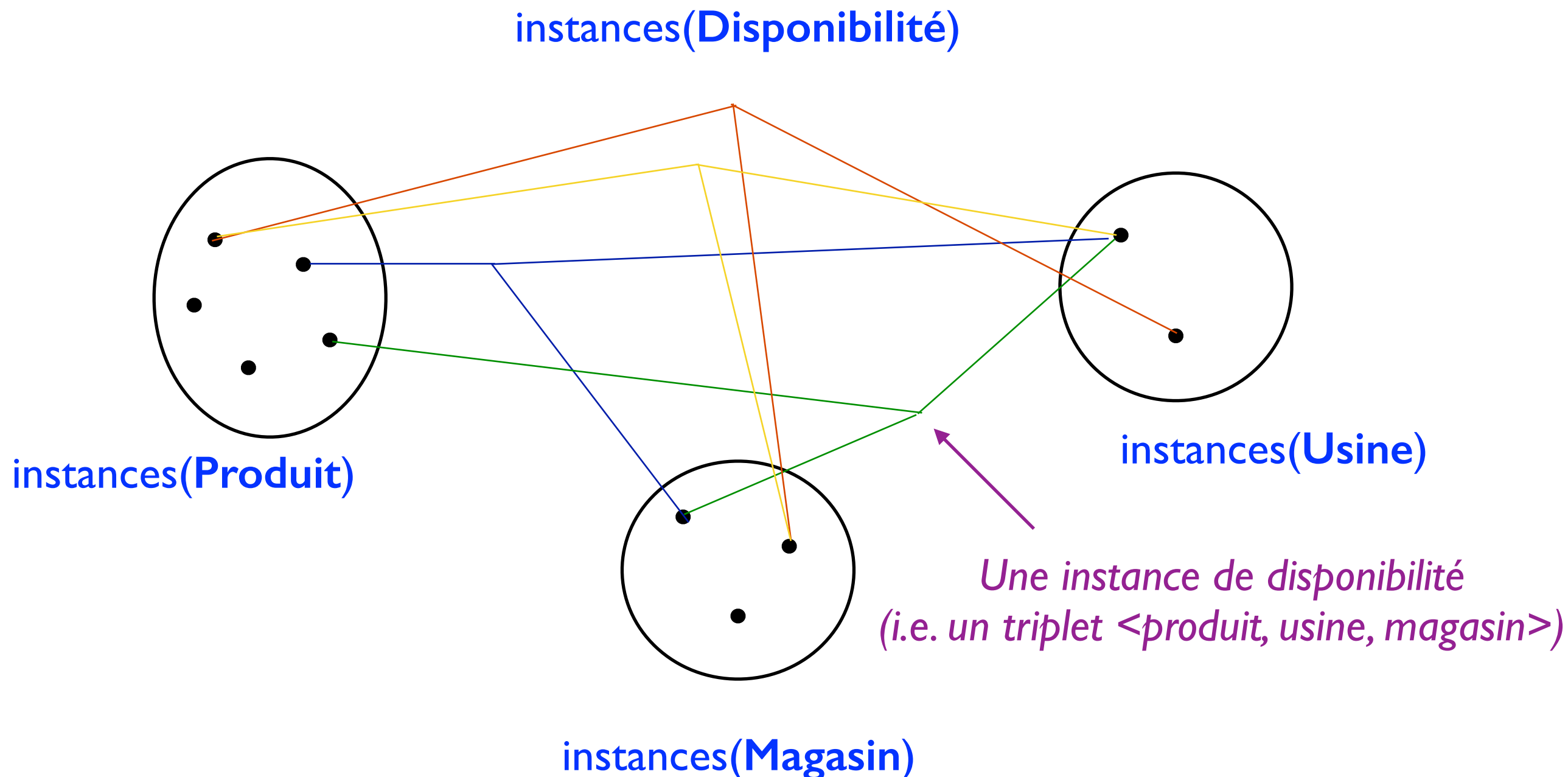
**Syntaxe :**



# Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

**Sémantique :**  $\text{instances(Disponibilité)} \subseteq \text{instances(Produit)} \times \text{instances(Usine)} \times \text{instances(Magasin)}$

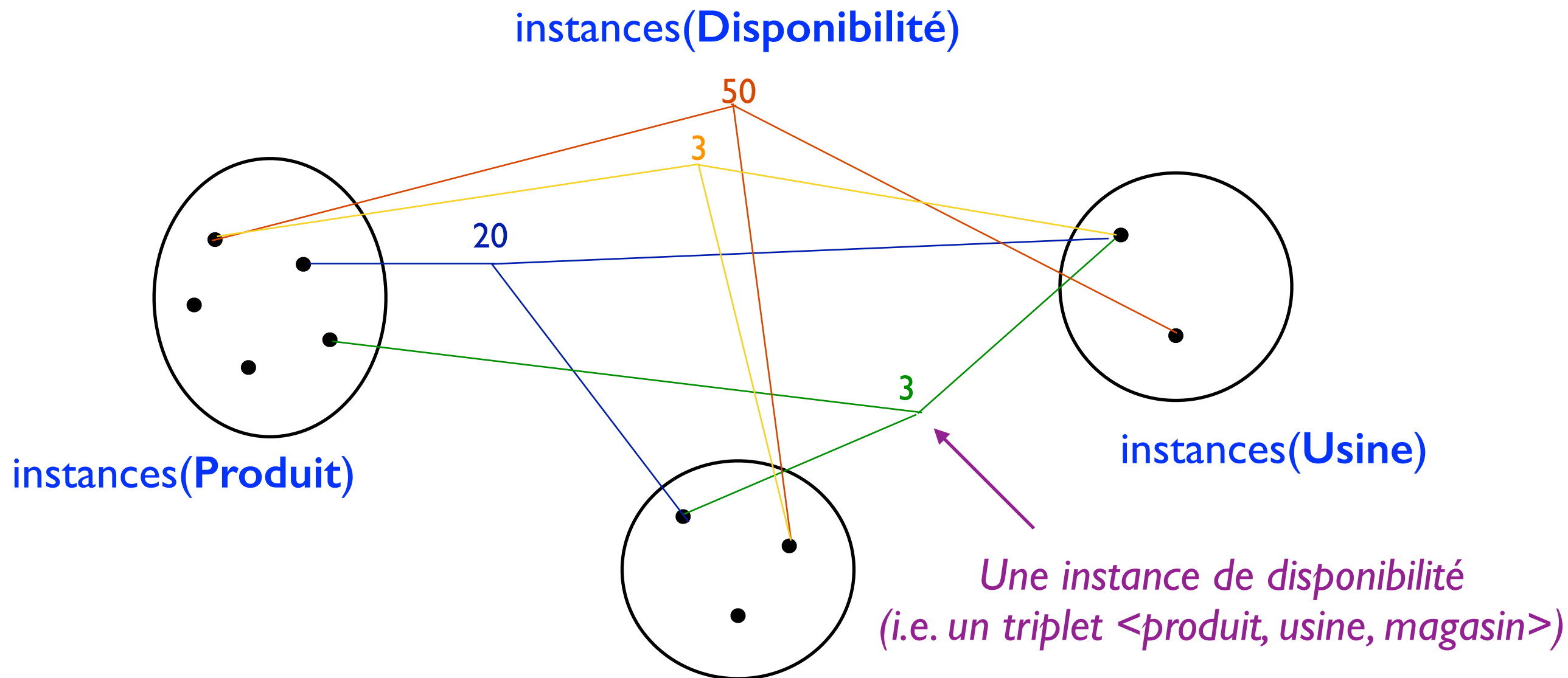


# Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

Sémantique :

$\text{instances(Disponibilité)} \subseteq$   
 $\text{instances(Produit)} \times \text{instances(Usine)} \times \text{instances(Magasin)}$



qté : instances(**Disponibilité**)  $\rightarrow$  Entiers

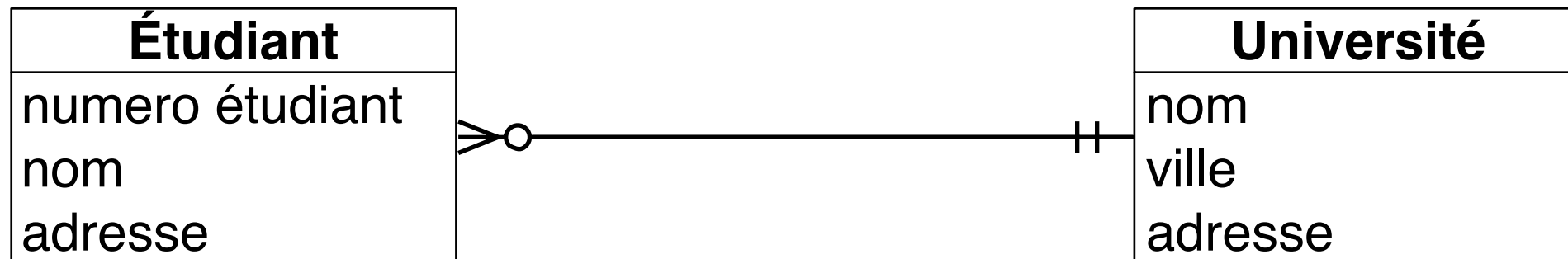
# Contraintes dans le modele E/R

- Le modele E/R peut exprimer certaines contraintes sur les données :
  - ▶ **Contraintes** : conditions additionnelles pour que les instances d'un diagramme E/R soient valides
- Deux types de contraintes exprimables par le modèle E/R:
  - ▶ cardinalité
  - ▶ identification
- D'autres contraintes plus complexes doivent être ajoutées au diagramme E/R en tant que **contraintes externes**

# Contraintes de cardinalité

- Expriment le nombre d'instances des entités qui peuvent être associées via une association

**Syntaxe** (exemple): Contraintes de cardinalité pour l'association **Inscription**

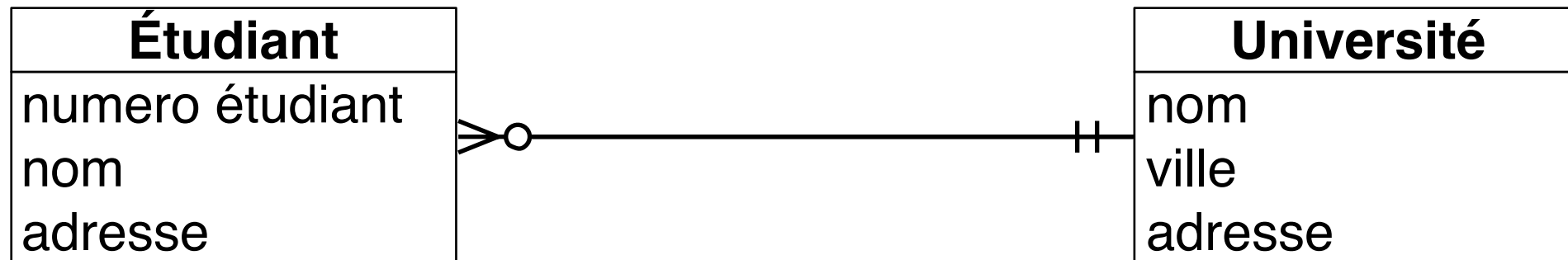


(Notation Crow's Foot)

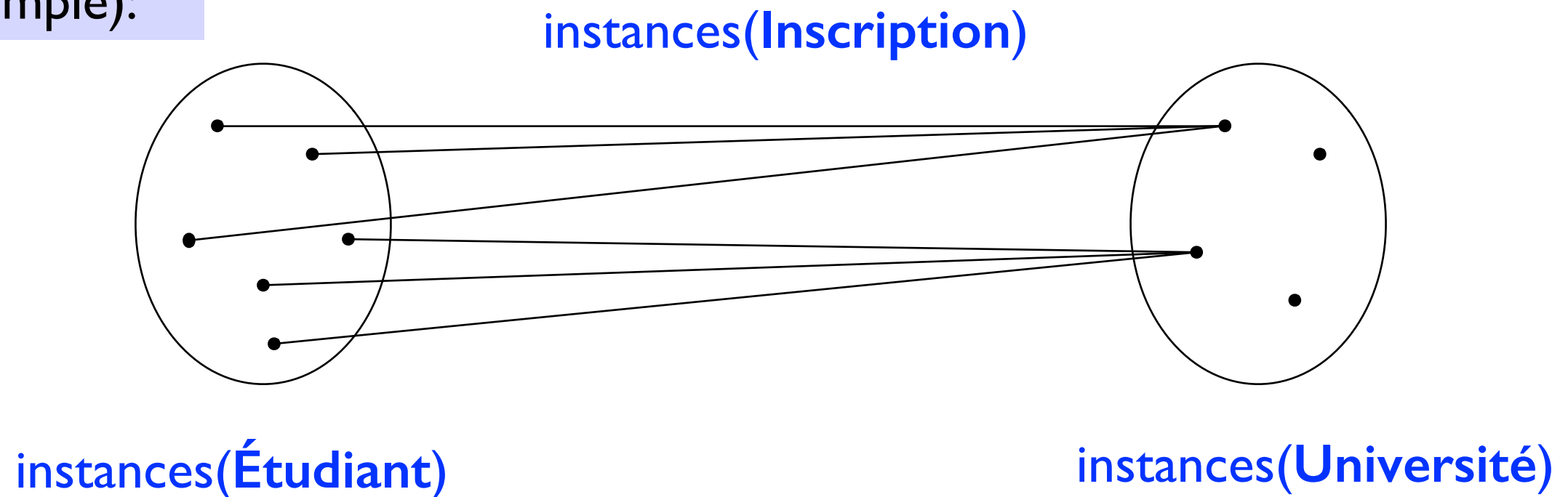
**Sémantique :**

- un même étudiant peut être inscrit à **une et une seule** université (min : 1, max : 1)
- une université peut avoir **entre 0 et plusieurs** inscrits (min : 0, pas de max)

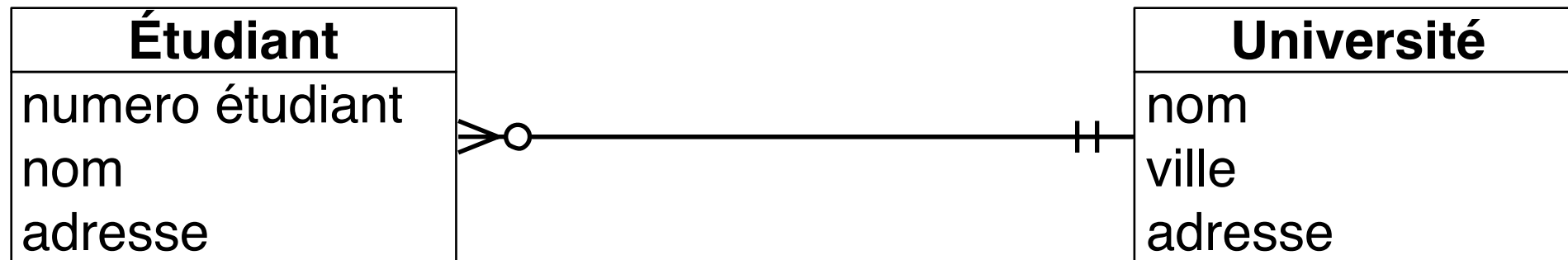
# Contraintes de cardinalité



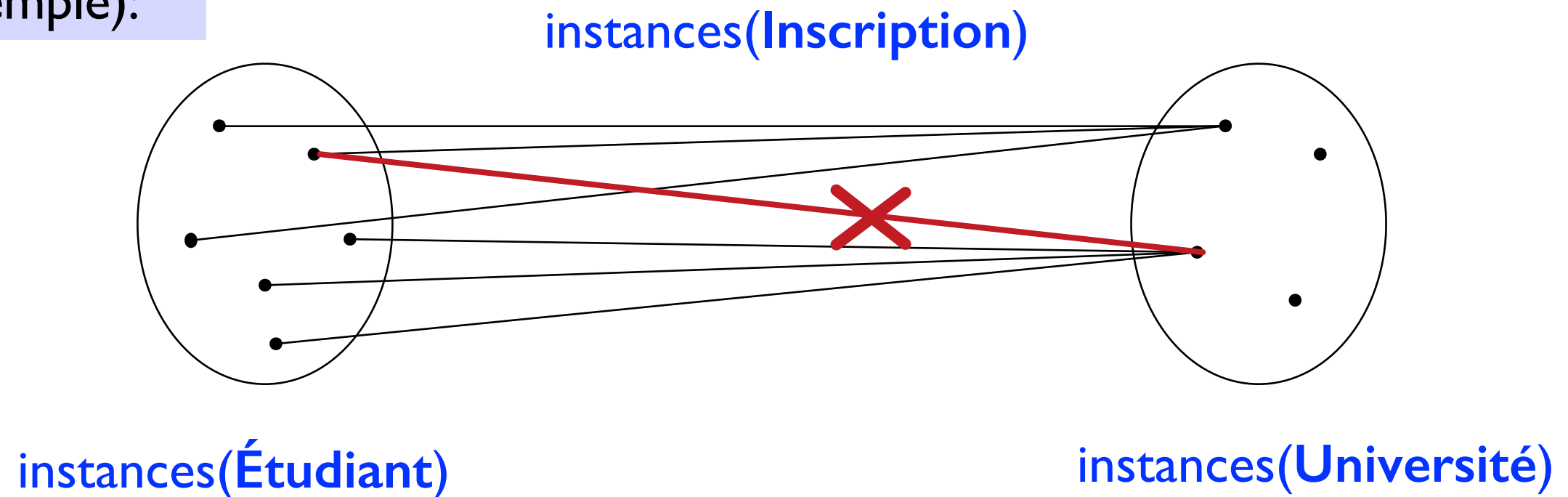
Sémantique  
(exemple):



# Contraintes de cardinalité



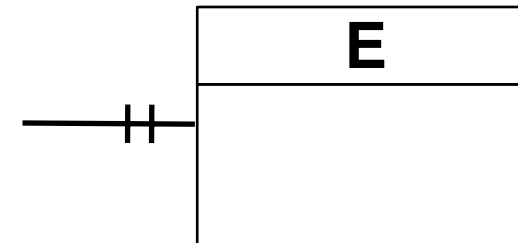
Sémantique  
(exemple):



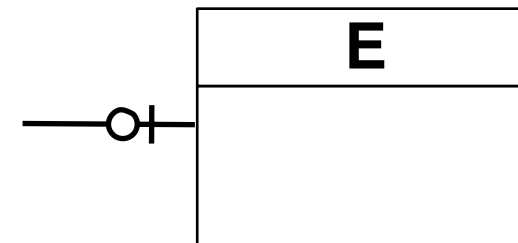
# Contraintes de cardinalité

- Cardinalités possibles :

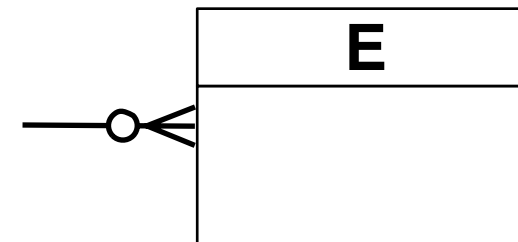
min: 1, max : 1



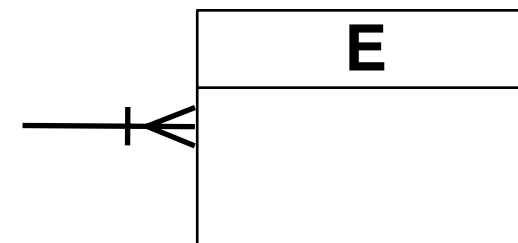
min: 0, max : 1



min: 0, pas de max



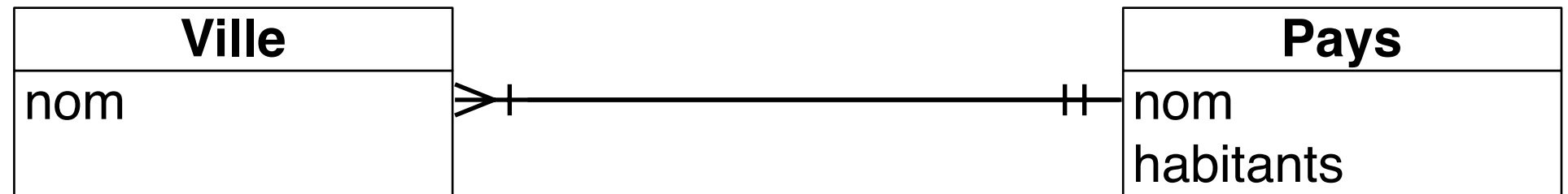
min: 1, pas de max



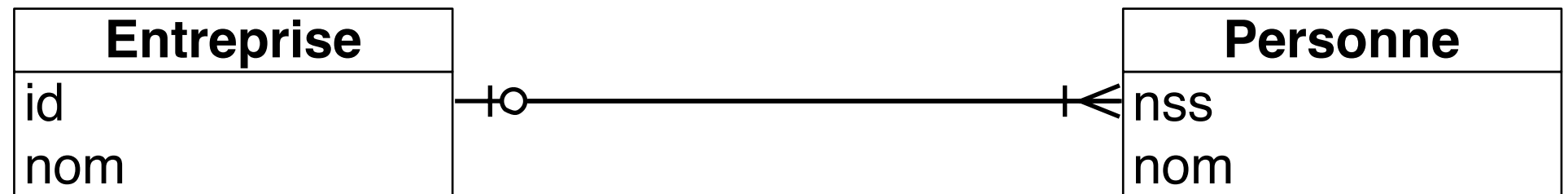


# Contraintes de cardinalité : exemples

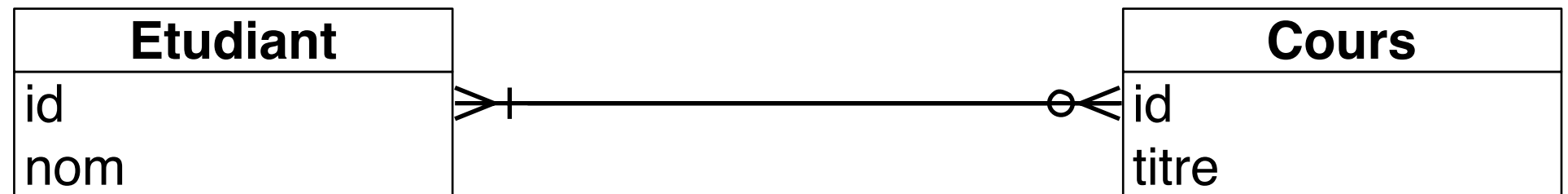
- Localisation :



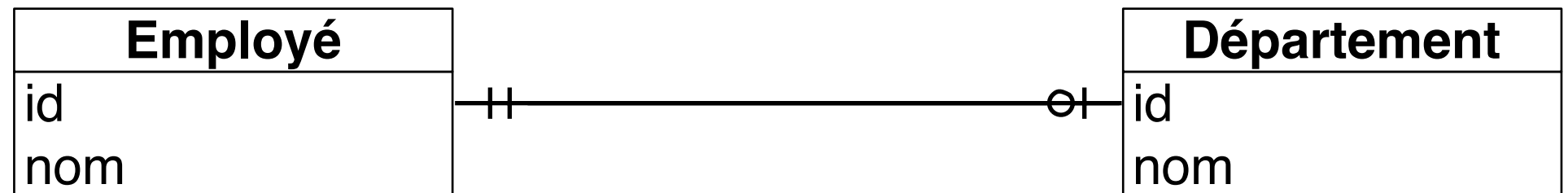
- Emploie :



- Inscription :



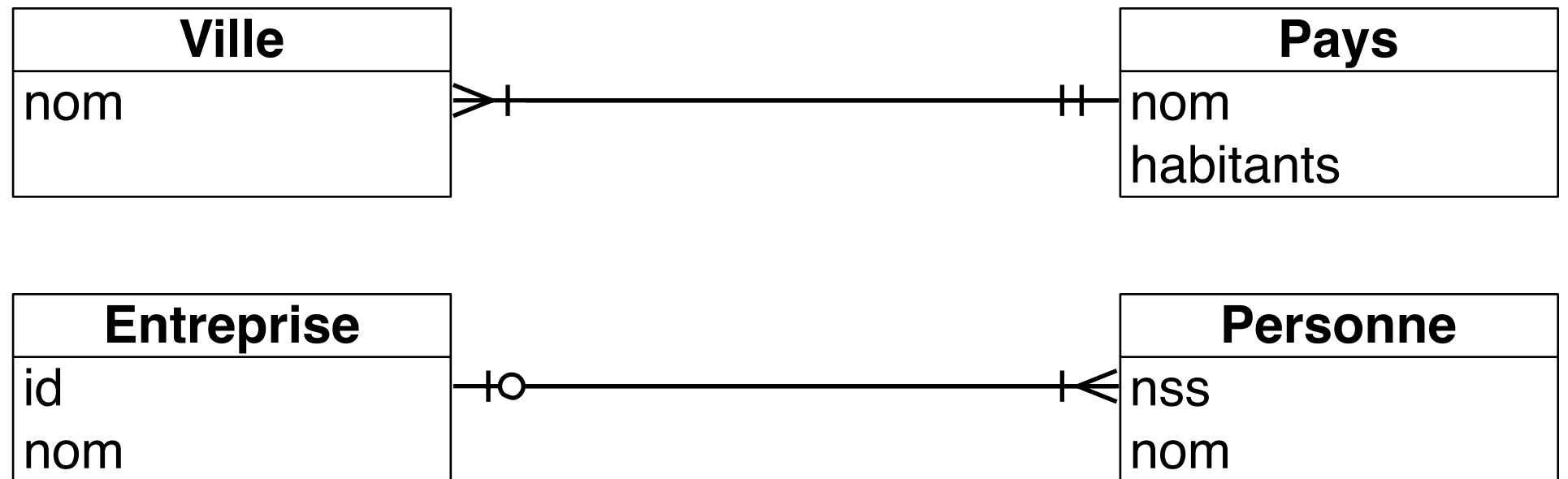
- Directeur :



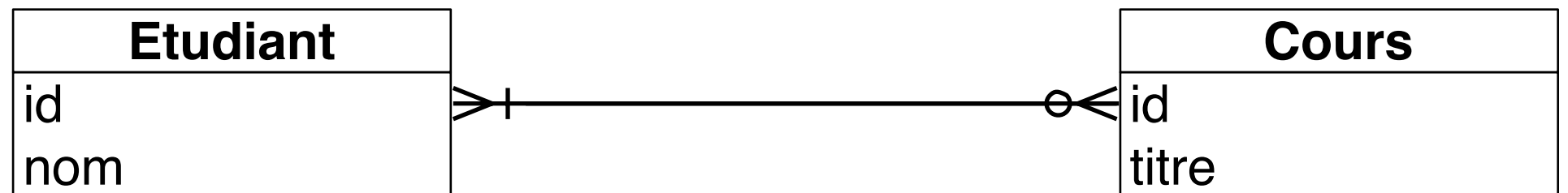
# Multiplicité des associations

- En faisant référence uniquement au **cardinalités maximales** de chaque coté, on distingue entre associations :

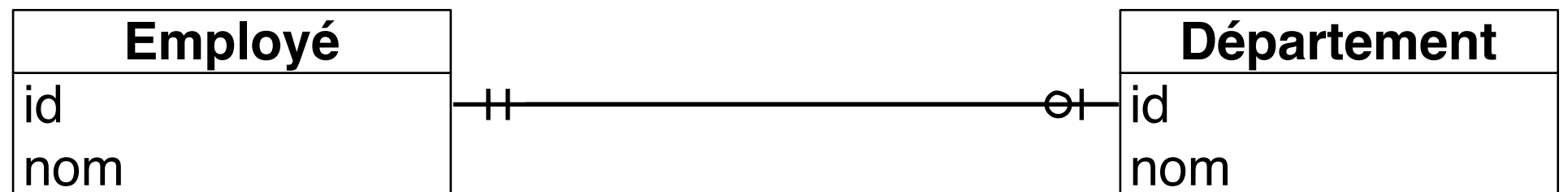
un à plusieurs



plusieurs  
à plusieurs

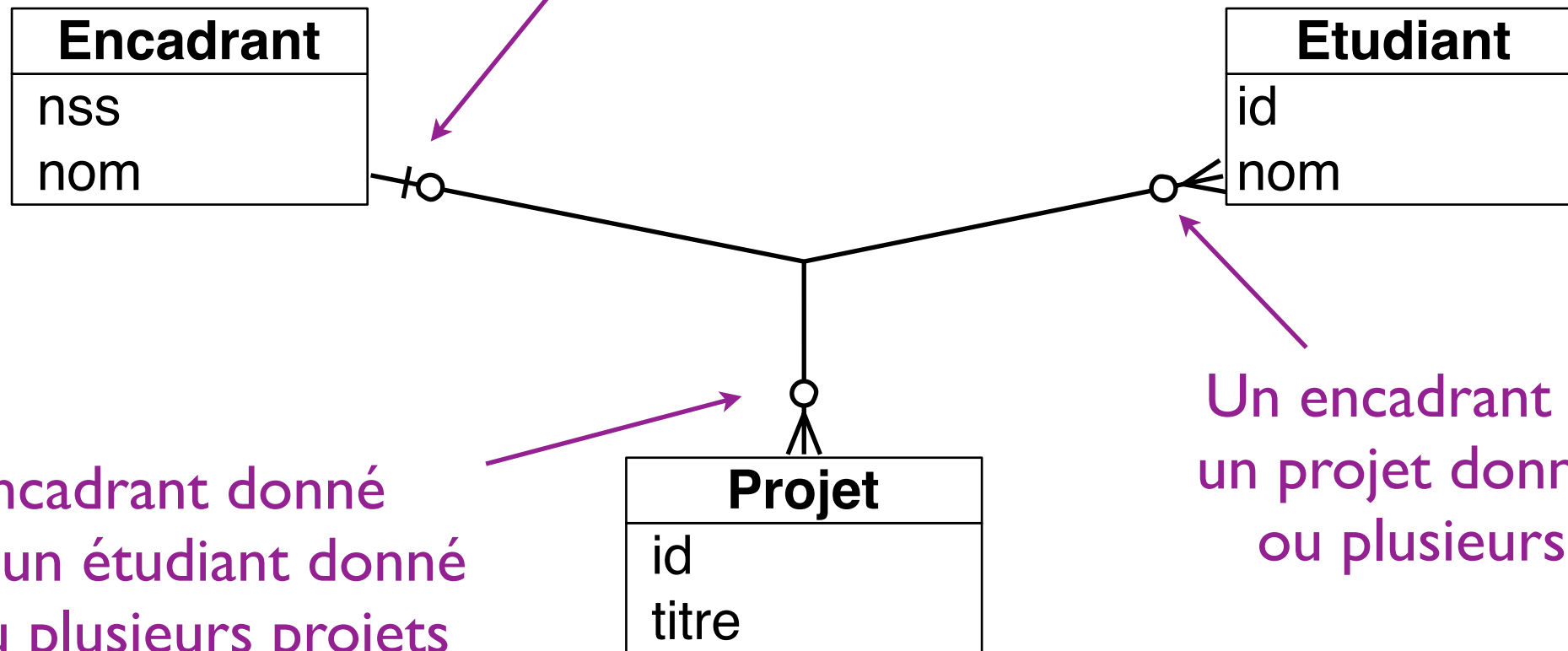


un à un



# Contraintes de cardinalité dans les associations n-aires

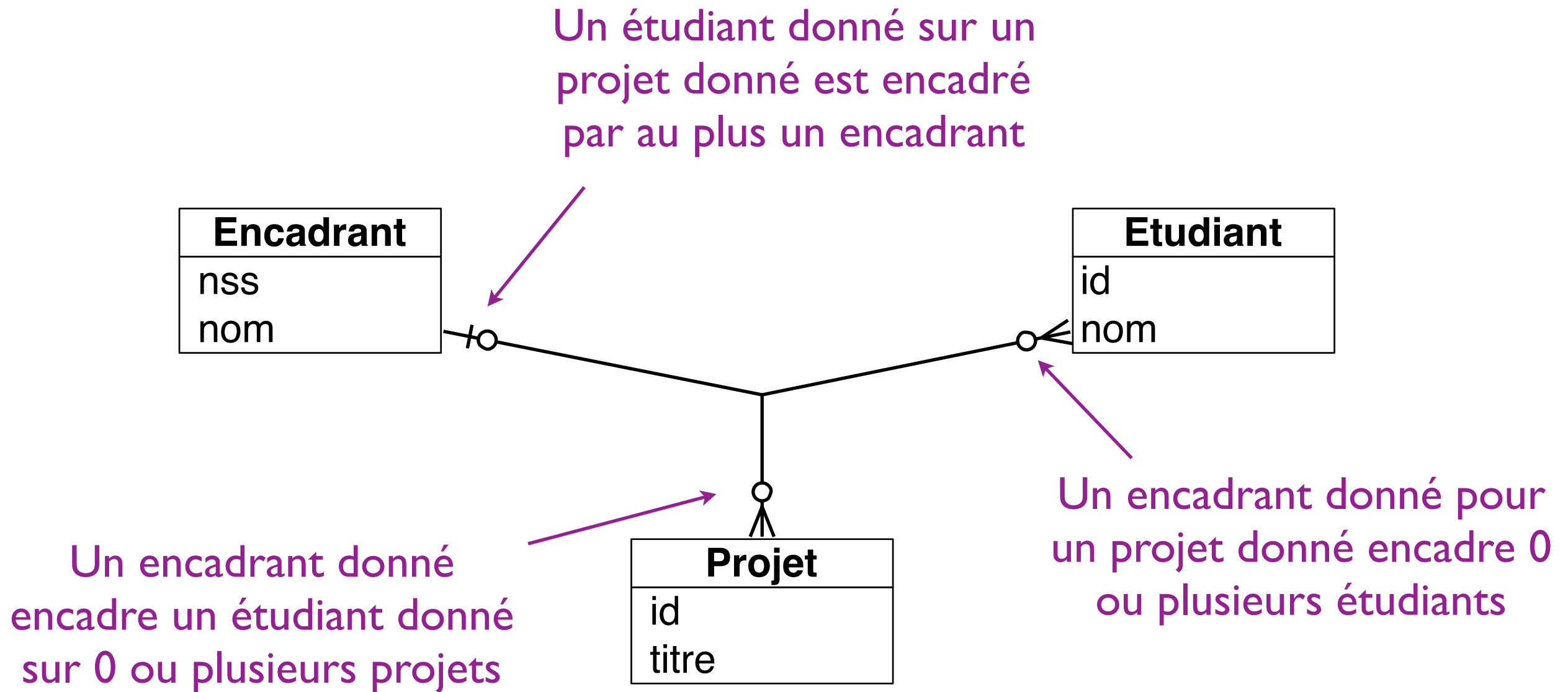
Un étudiant donné sur un projet donné est encadré par au plus un encadrant



Un encadrant donné encadre un étudiant donné sur 0 ou plusieurs projets

Un encadrant donné pour un projet donné encadre 0 ou plusieurs étudiants

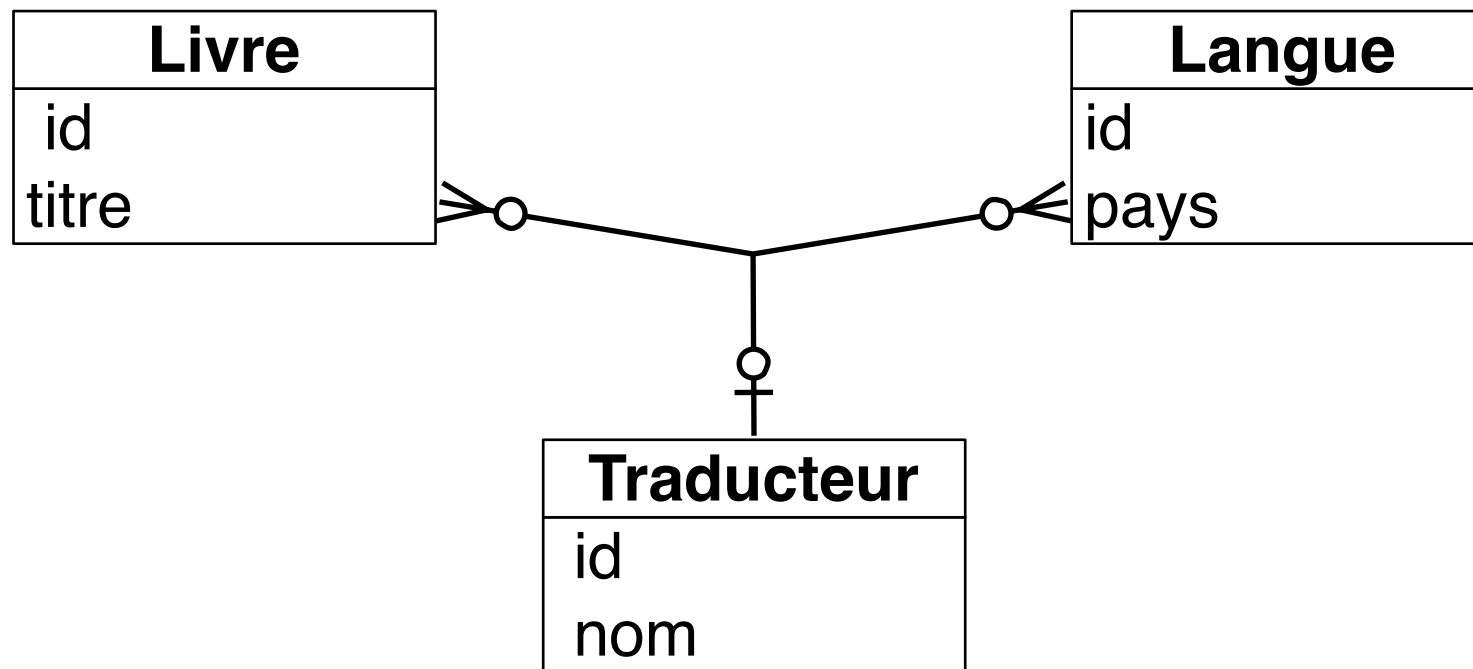
# Contraintes de cardinalité dans les associations n-aires



- **Remarque** : les cardinalités minimales sont moins intéressantes sur les associations ternaires : elles sont en général 0
  - ▶ E.g : si chaque couple <étudiant, projet> était associé à min 1 encadrant, cela voudrait dire que tous les étudiants sont affectés à tous les projets!

# Contraintes de cardinalité dans les associations n-aires

D'autres exemples : l'association **Traduction**



À 1 livre et 1 langue correspond 0 ou 1 traducteur.

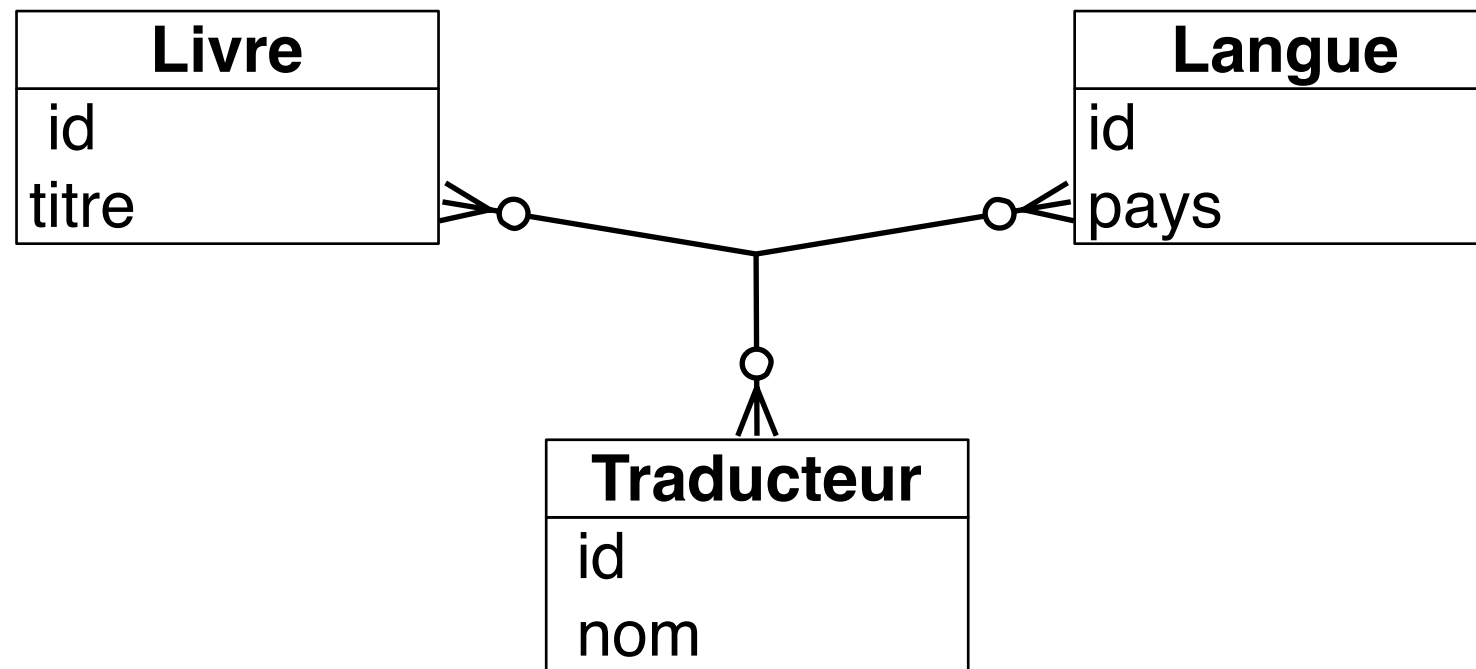
À 1 langue et 1 traducteur correspond 0 ou plusieurs livres.

À 1 livre et 1 traducteur correspond 0 ou plusieurs langues.

# Contraintes de cardinalité dans les associations n-aires

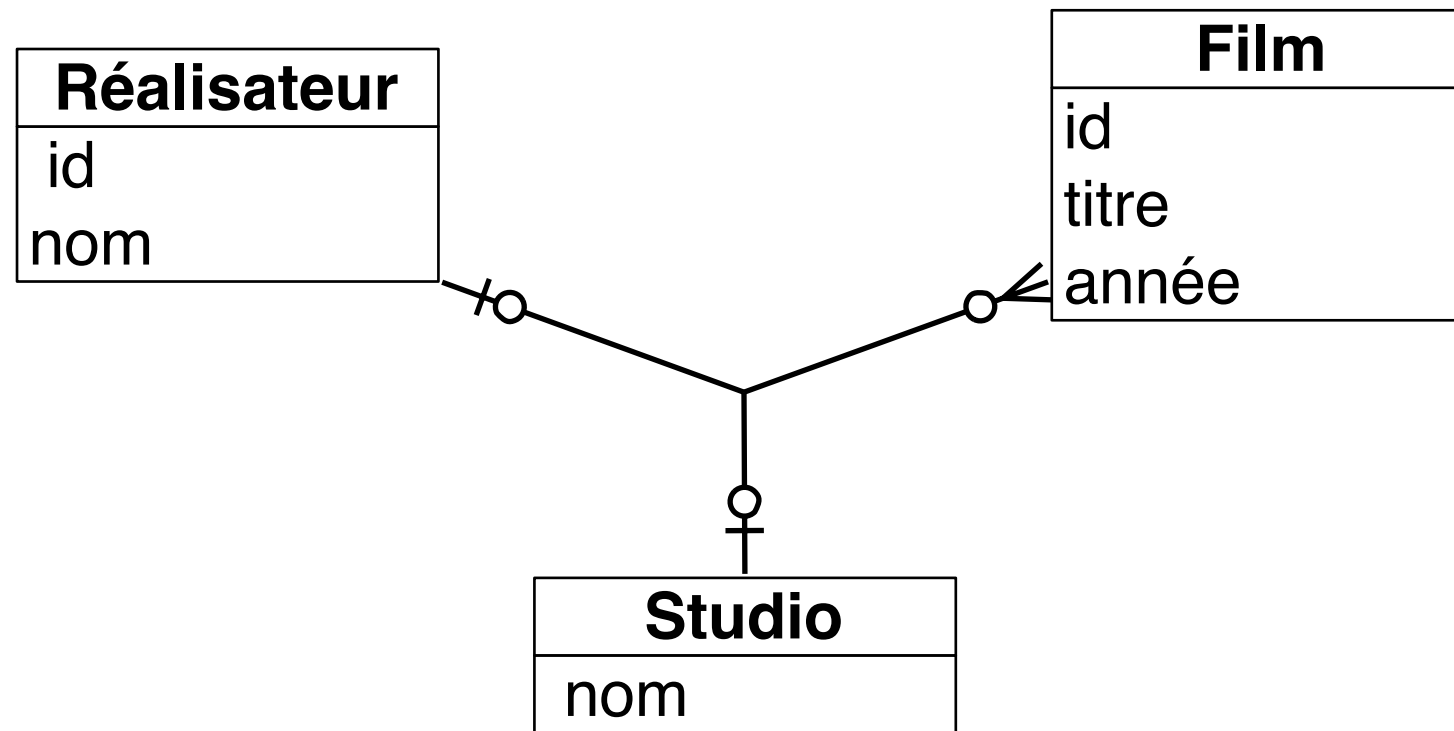
D'autres exemples : l'association **Traduction**

Si en revanche un livre peut être traduit dans la même langue par plusieurs traducteurs :



# Contraintes de cardinalité dans les associations n-aires

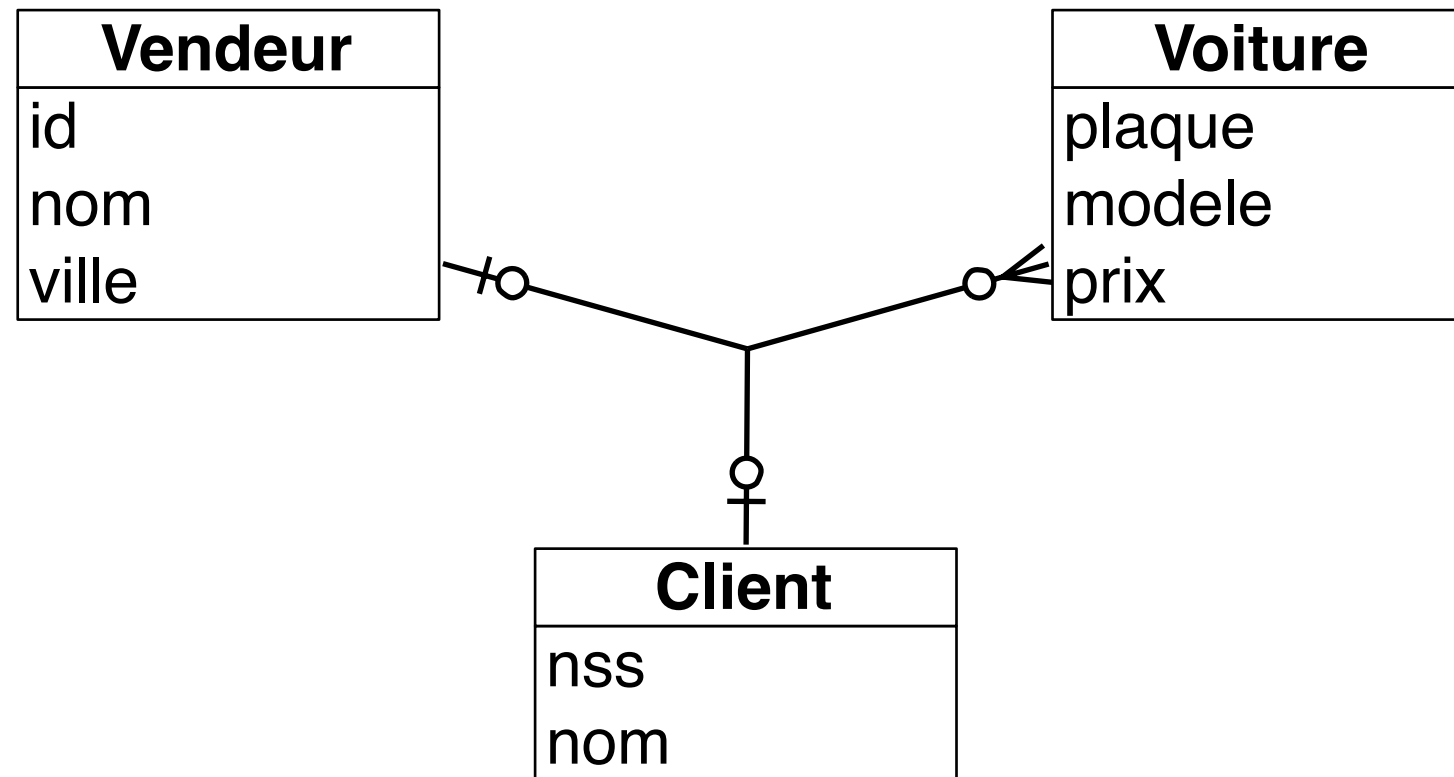
D'autres exemples : l'association **Contrat**



- ▶ Chaque réalisateur pour chaque film fait un contrat avec 0 ou un seul studio (le studio du film)
- ▶ Chaque studio sur chaque film a un contrat avec au plus un réalisateur (le réalisateur du film)
- ▶ Chaque réalisateur avec chaque studio peut faire un contrat pour 0 ou plusieurs films

# Contraintes de cardinalité dans les associations n-aires

D'autres exemples : l'association **Achat** (de voitures neuves)



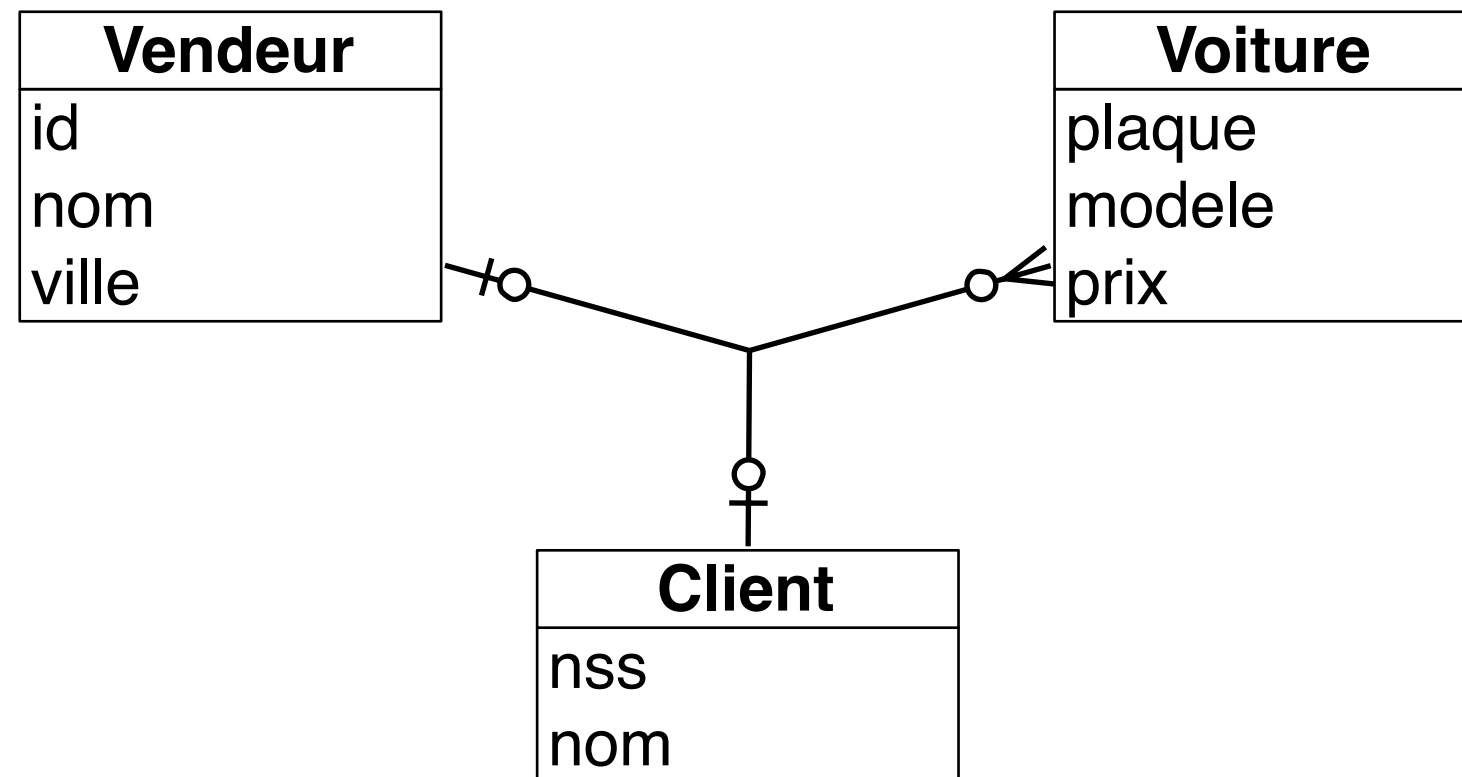
- ▶ Pour chaque client et chaque voiture il y a au plus un vendeur effectuant la vente
- ▶ Pour chaque voiture et chaque vendeur il y a au plus un client acheteur
- ▶ Pour chaque vendeur et chaque client il y a zéro ou plusieurs voitures achetées



# Cardinalité et contraintes externes

- Les contraintes de cardinalité ne peuvent pas toujours exprimer toutes les contraintes imposées par la réalité à modéliser

## Exemple

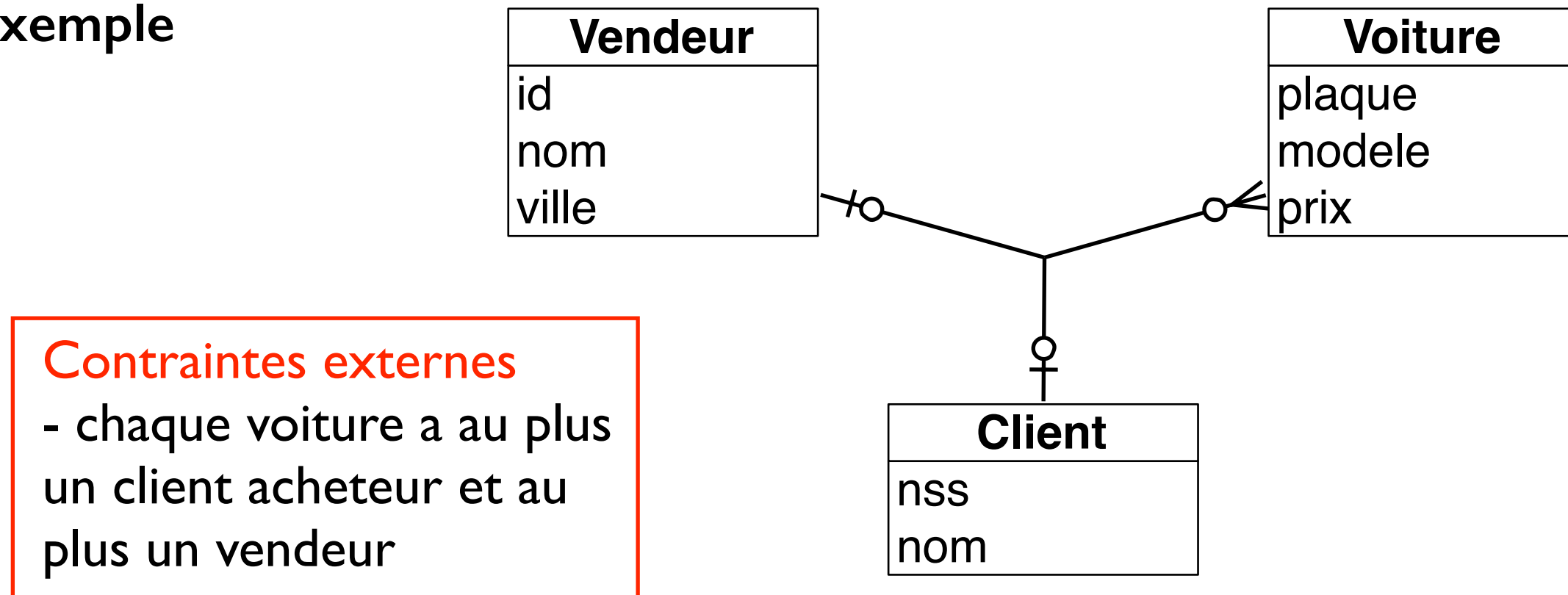


- La “vrai” contrainte sur les données : chaque voiture a au plus un client acheteur et au plus un vendeur
- Pas possible d’exprimer cette contrainte sur la relation ternaire ci-dessus avec notre notation
  - $\langle V, Vend1, C1 \rangle$   $\langle V, Vend2, C2 \rangle$  possibles dans instances(Achat))

# Cardinalité et contraintes externes

- Les contraintes de cardinalité ne peuvent pas toujours exprimer toutes les contraintes imposées par la réalité à modéliser

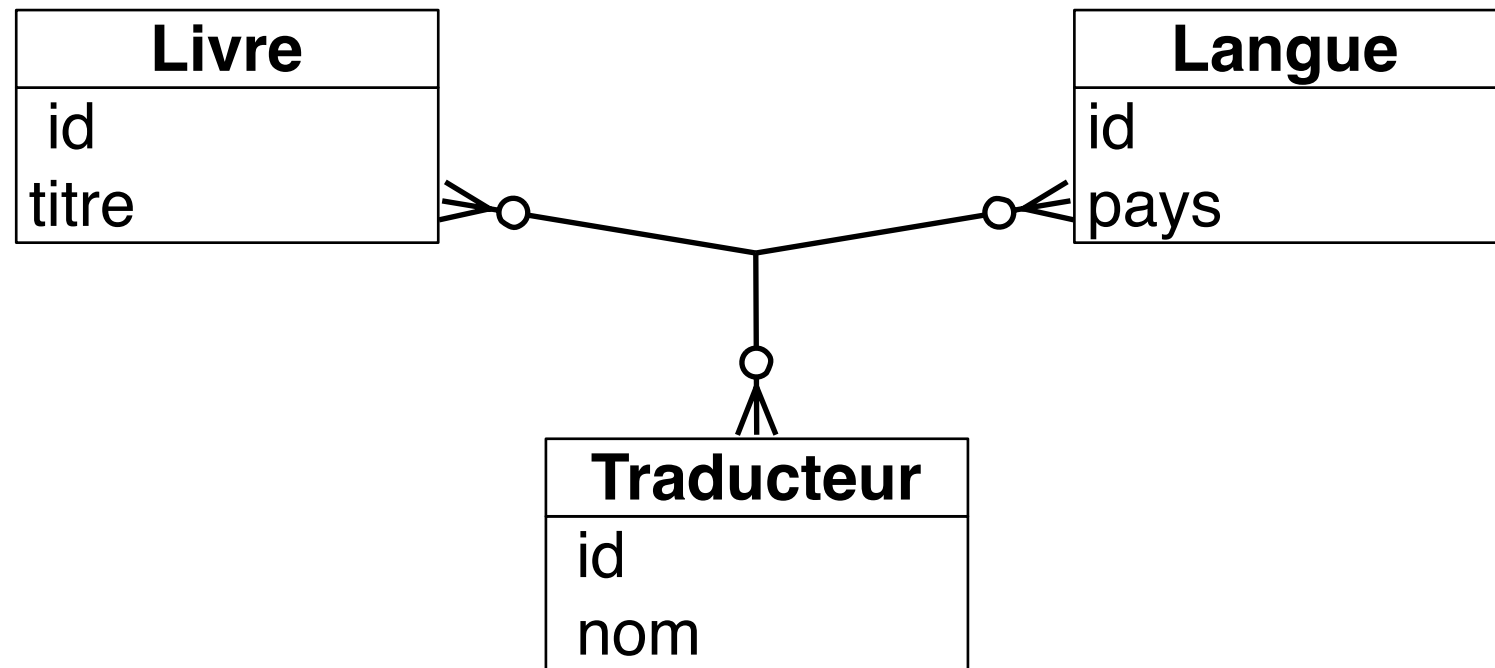
## Exemple



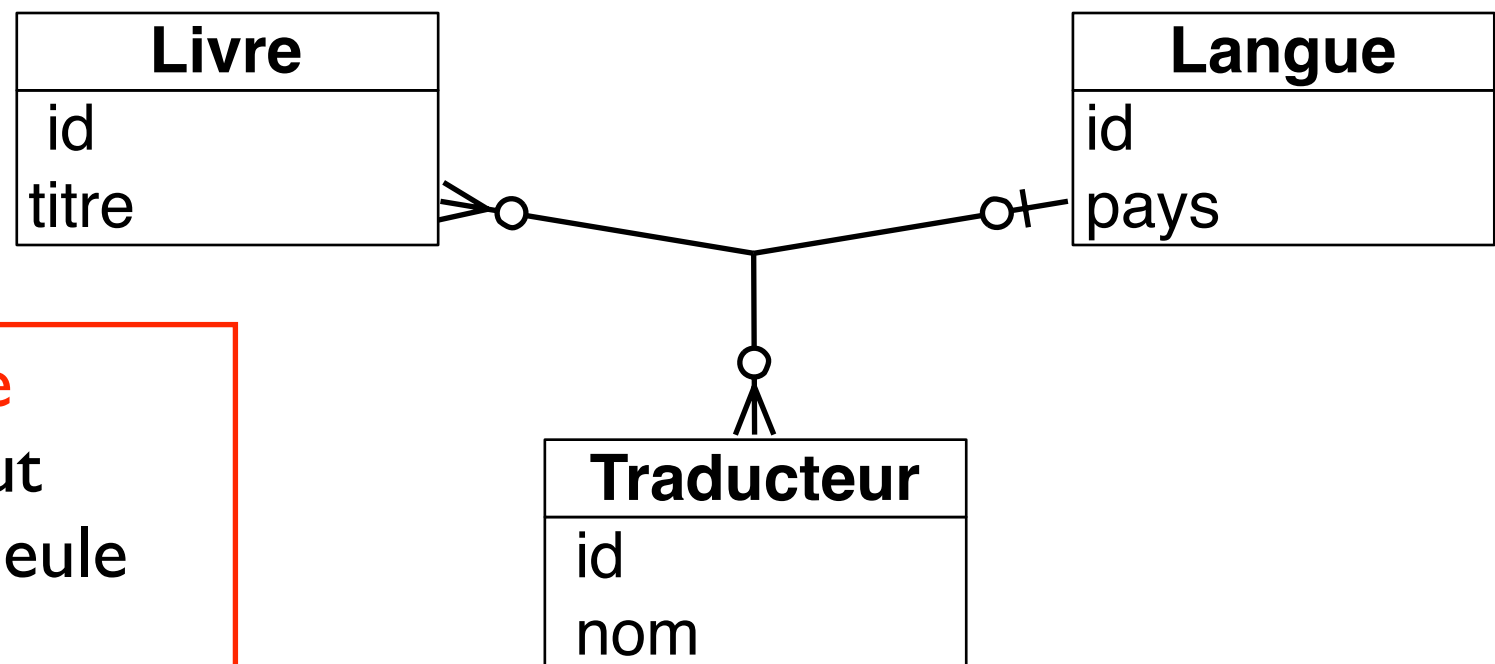
- La “vrai” contrainte sur les données : chaque voiture a au plus un client acheteur et au plus un vendeur
- Pas possible d’exprimer cette contrainte sur la relation ternaire ci-dessus avec notre notation
- Dans ces cas on l’ajoute dans le diagramme comme **contrainte externe**

# Cardinalité et contraintes externes

Un autre exemple :



Si on ajoute la contrainte que un traducteur traduit toujours dans une seule langue, le schema devient :

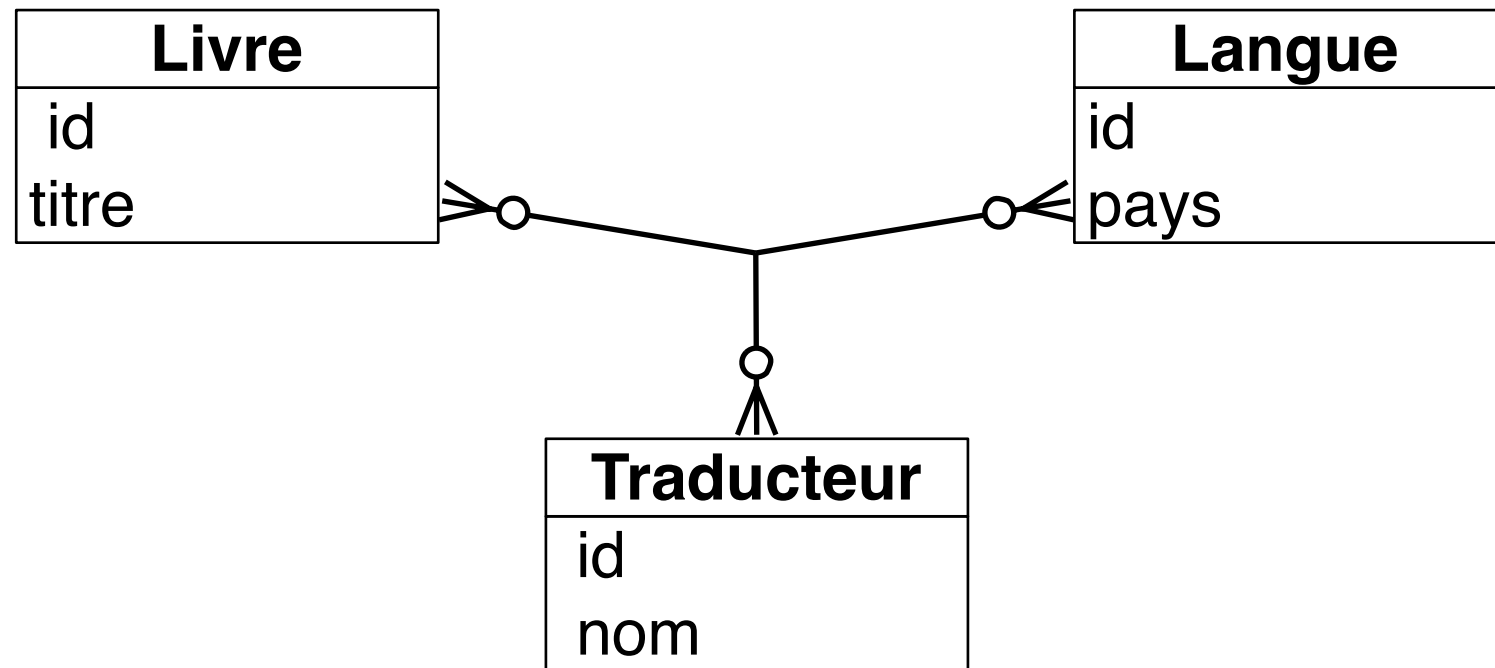


## Contrainte externe

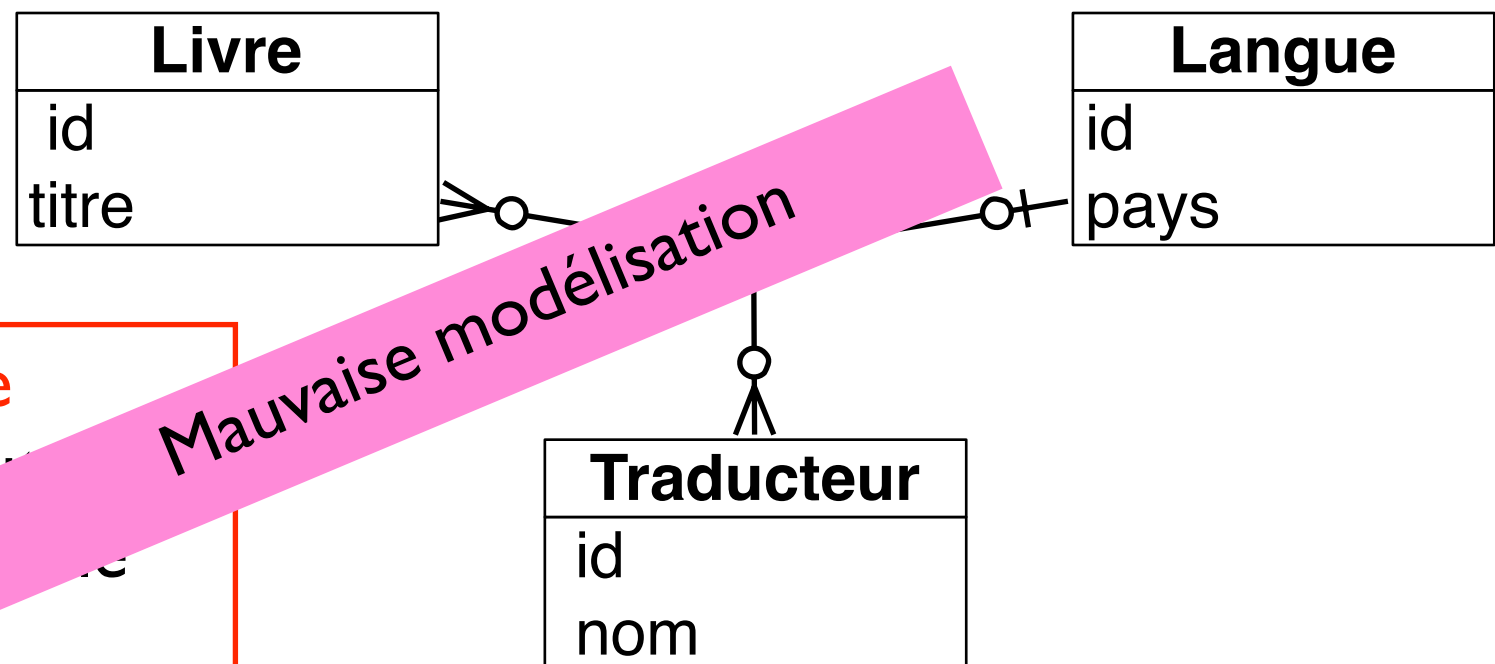
- un traducteur peut traduire dans une seule langue

# Cardinalité et contraintes externes

Un autre exemple :



Si on ajoute la contrainte que un traducteur traduit toujours dans une seule langue, le schema devient :

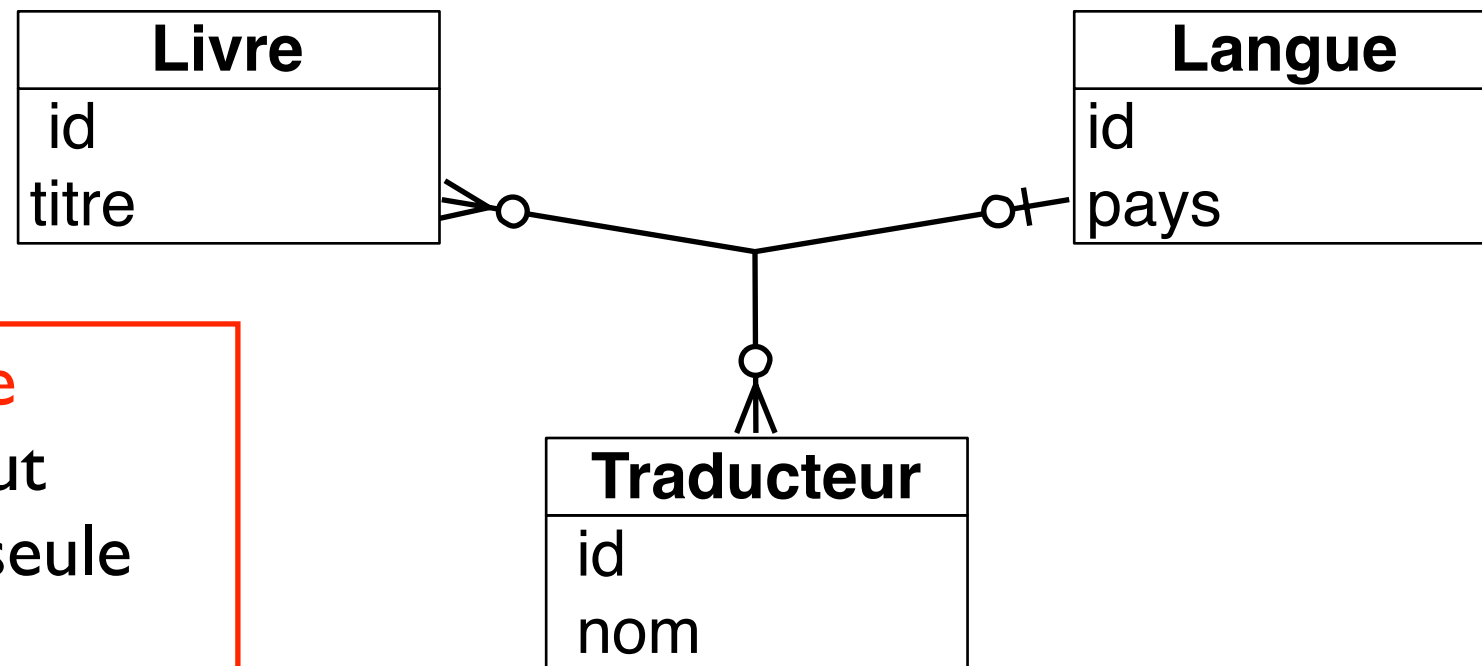


**Contrainte externe**

- un traducteur peut traduire dans une seule langue

# Cardinalité et contraintes externes

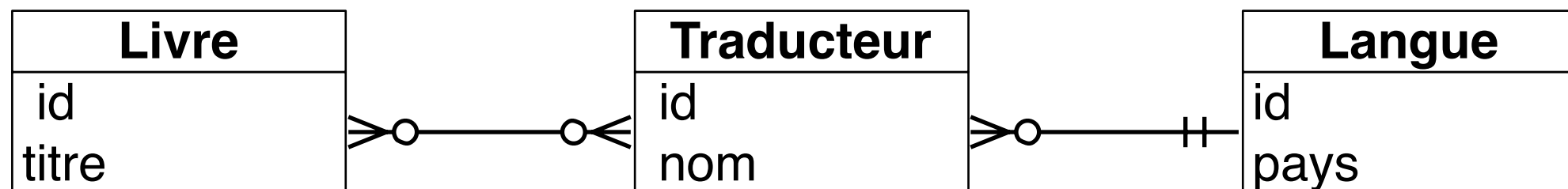
- Pourquoi celle-là est une mauvaise modélisation?



## Contrainte externe

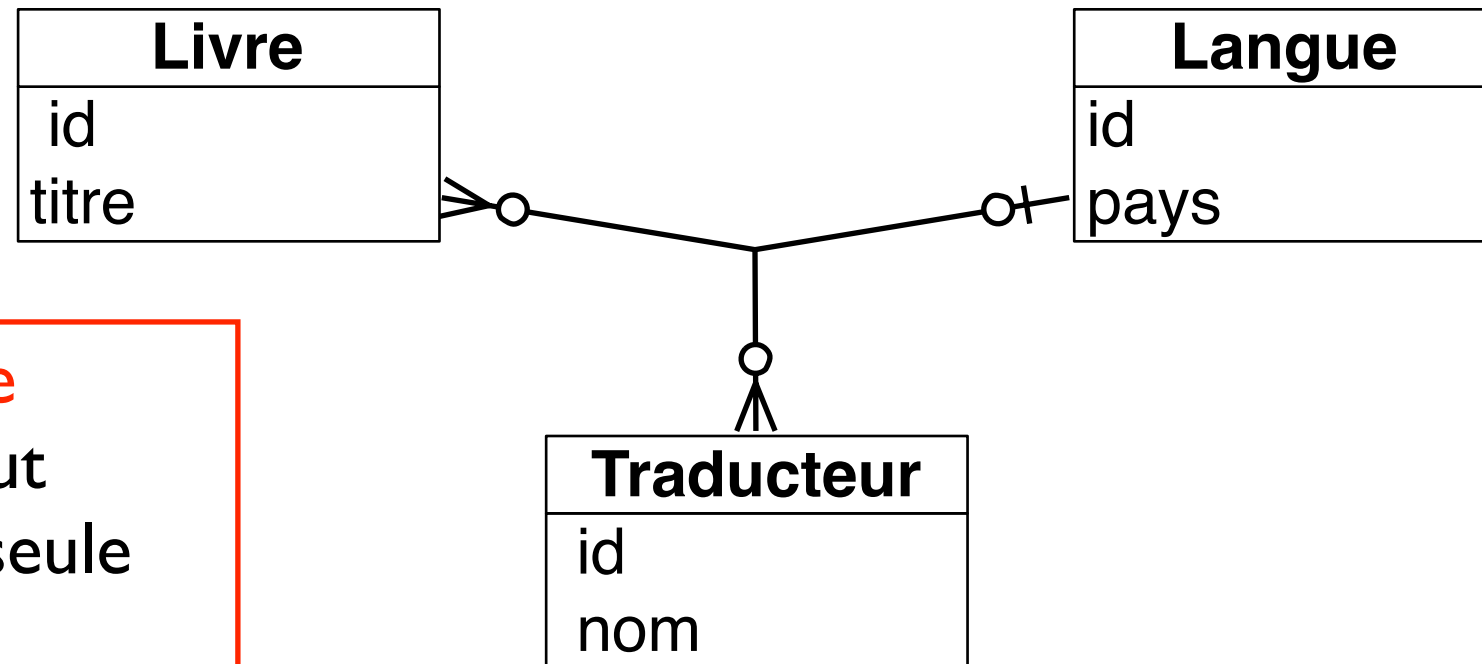
- un traducteur peut traduire dans une seule langue

- On l'expliquera formellement plus tard avec la **théorie de la normalisation**
- Intuitivement puisque un traducteur peut traduire dans une seule langue, une modélisation plus appropriée est :



# Cardinalité et contraintes externes

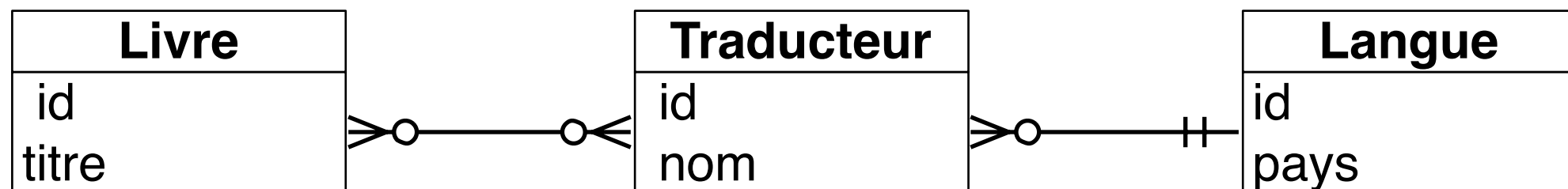
- Pourquoi celle-là est une mauve modélisation?



## Contrainte externe

- un traducteur peut traduire dans une seule langue

**Conclusion** : attention à bien exprimer les contraintes externes sur les associations n-aires : elles peuvent révéler que le choix d'association n-aire est mauvais



## Rappel : Contraintes dans le modèle E/R

- Le modèle E/R peut exprimer certaines contraintes sur les données :
  - ▶ **Contraintes** : conditions additionnelles pour que les instances d'un diagramme E/R soient valides
- Deux types de contraintes :
  - ▶ cardinalité
  - ▶ **identification (clefs)**

# Clefs

- Une **superclef** d'une entité est un ensemble de un ou plusieurs attributs tel que: il n'existe pas deux instances de l'entité avec la même valeur de tous ces attributs
  - ▶ Exemples :
    - **nss** est une superclef pour l'entité Personne
    - **(titre, réalisateur)** est une superclef pour l'entité Film
    - **(nss, nom)** est une superclef pour l'entité Personne
    - **(ville, rue, numéro)** est une superclef pour l'entité Bâtiment
- Une **clef** (ou **clef candidate**) d'une entité est une superclef minimale
  - **nss** est une clef pour l'entité Personne
  - **(titre, réalisateur)** est une clef pour l'entité Film
  - **(ville, rue, numéro)** est une clef pour l'entité Bâtiment
- Plusieurs clefs candidates peuvent exister pour une entité, mais pour chaque entité une seule clef est choisie comme **clef primaire**.



# Clefs primaires

- Dans un diagramme E/R une clef primaire est spécifiée pour chaque entité

Syntaxe :

Bâtiment
<u>numero</u>
<u>rue</u>
<u>ville</u>
nombre-étages
année-construction

Sémantique :

Il n'existe pas deux bâtiments différents  $b1$  et  $b2$  dans  $Instances(Bâtiment)$  tels que :  
 $numero(b1) = numero(b2)$  et  $rue(b1) = rue(b2)$  et  $ville(b1) = ville(b2)$

- **Remarque** : il peut exister deux bâtiments avec le même numéro et le même nom de rue (dans des villes différentes), mais pas avec le même numéro, la même rue et la même ville.

# Clefs primaires

- Un autre exemple de clef primaire :

Syntaxe :

Film
<u>titre</u>
<u>réalisateur</u>
année

Sémantique :

Il n'existe pas deux films différents  $f_1$  et  $f_2$  dans  $\text{Instances}(\text{Film})$  tels que :  $\text{titre}(b_1) = \text{titre}(b_2)$  et  $\text{réalisateur}(b_1) = \text{réalisateur}(b_2)$

# Clefs primaires

- Un dernier exemple :

Syntaxe :

Personne
<u>nss</u>
nom
prenom
date-naissance

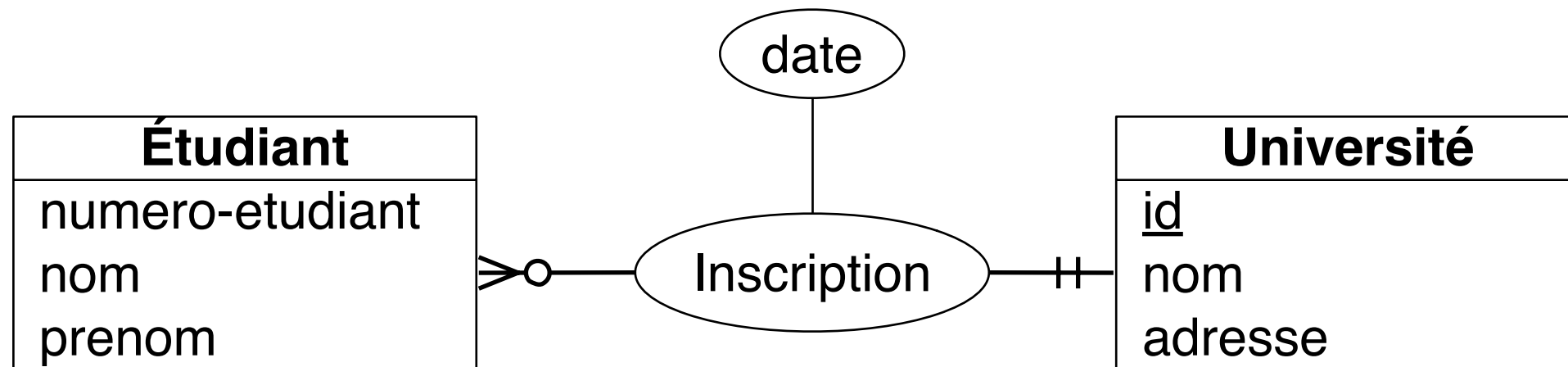
Sémantique :

Il n'existe pas deux personnes différents  $p1$  et  $p2$  dans  $Instances(Personne)$  tels que:  
 $nss(p1) = nss(p2)$

## Entités faibles

- Pour certaines entités, aucun ensemble d'attributs forme un clef

Exemple :

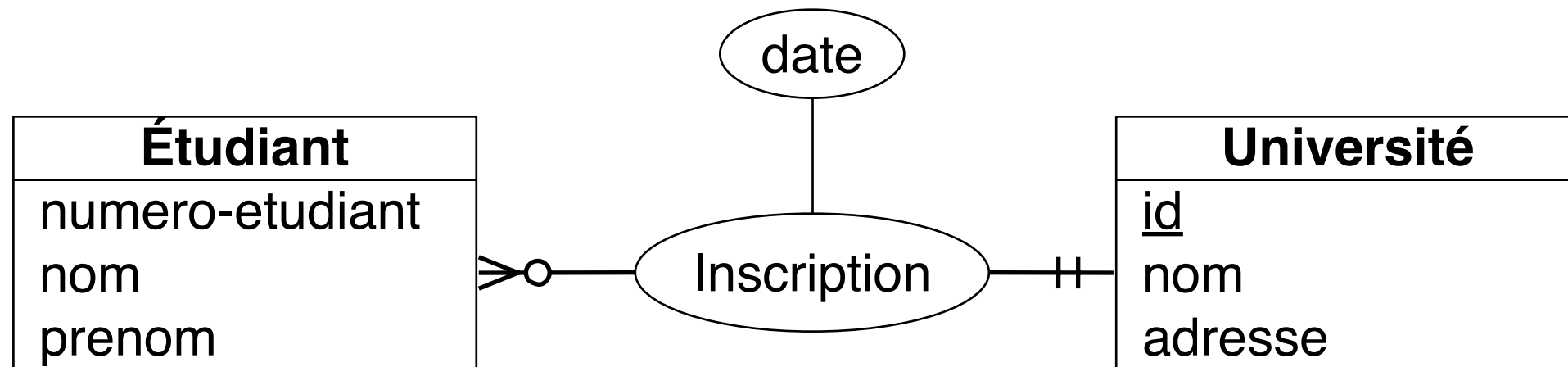


Quel ensemble d'attributs est une clef pour l'entité Étudiant ?

## Entités faibles

- Pour certaines entités, aucun ensemble d'attributs forme un clef

Exemple :



**Un étudiant est identifié par son numéro étudiant au sein de son université.**

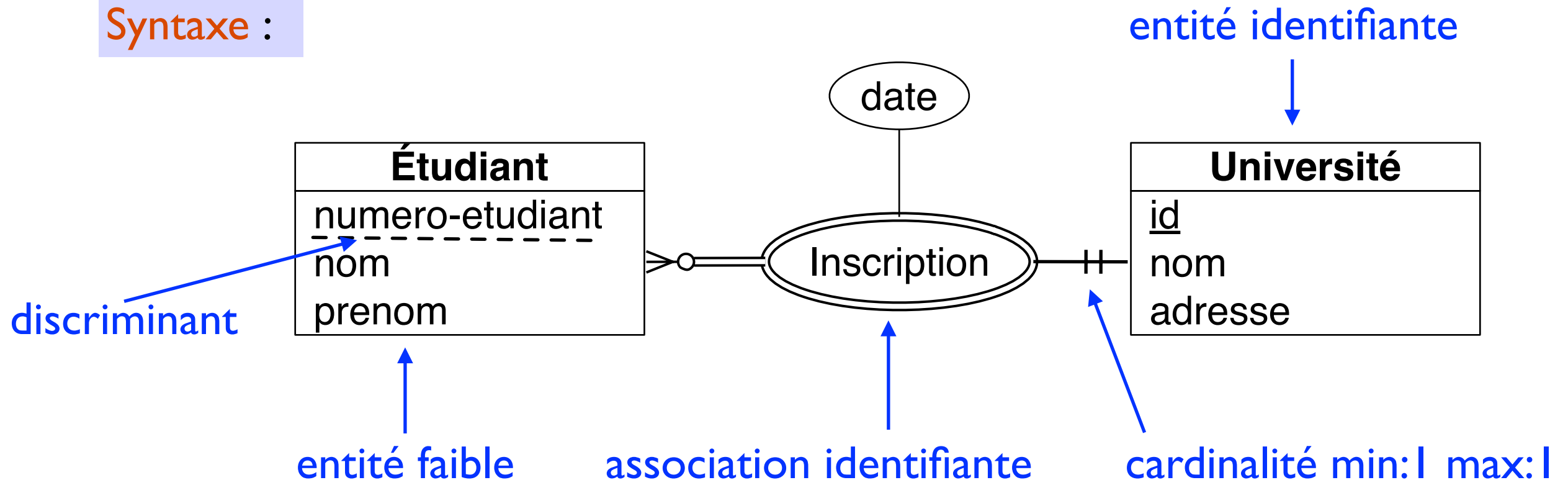
En d'autres termes une clef pour Étudiant est donnée par le couple

- numéro étudiant
- université d'inscription

# Entités faibles

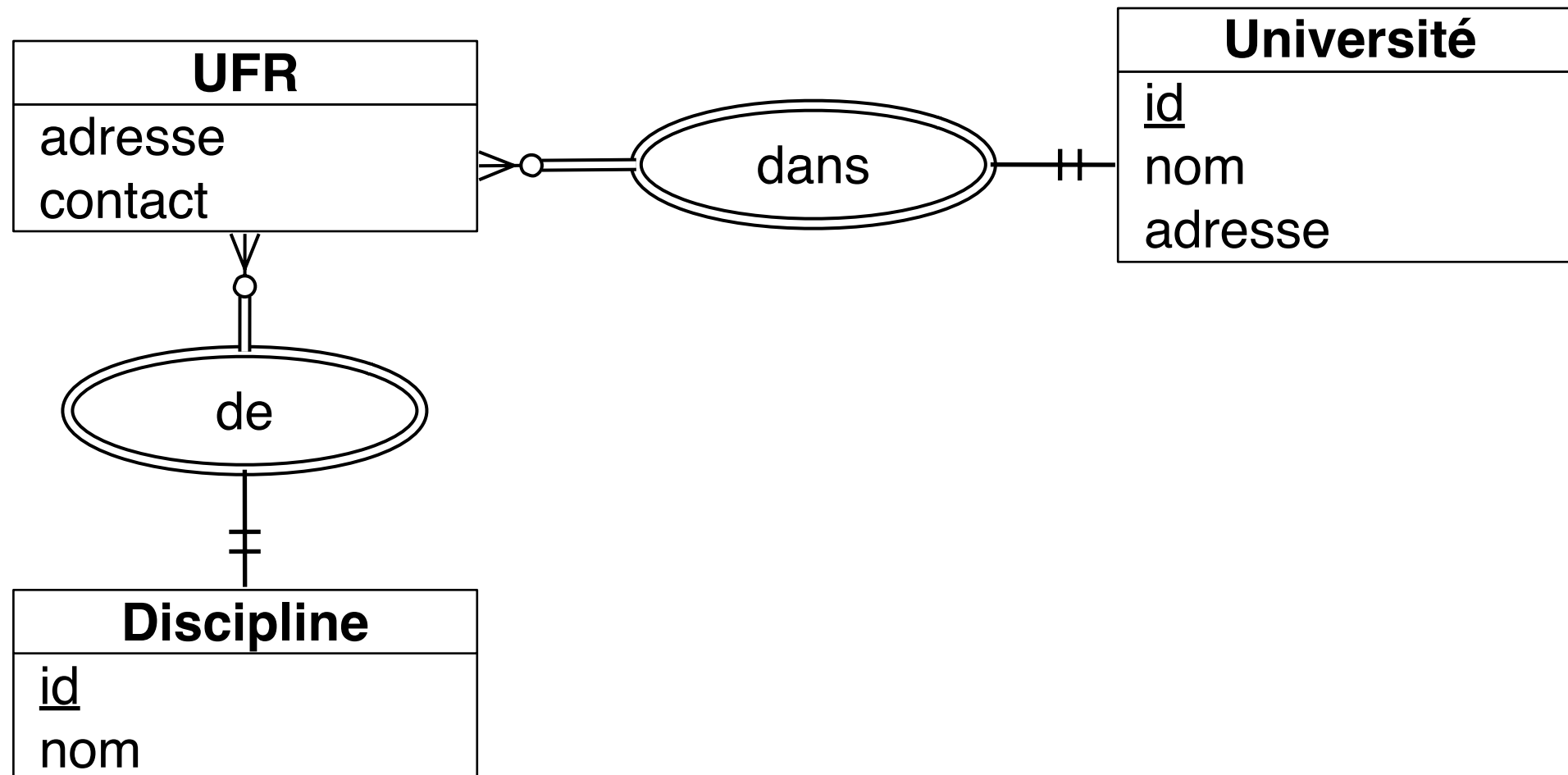
- Une entité faible est une entité identifiée par
  - ▶ un ensemble d'attributs internes appelés *discriminant* et
  - ▶ un ensemble d'entités appelés *entités identifiantes*
- Chaque entité identifiante doit être reliée à l'entité faible par une *association binaire* appelée *association identifiante*
- l'entité faible doit participer à l'association identifiante avec une *cardinalité min:1 max:1*

Syntaxe :



## Entités faibles

- Un exemple avec plusieurs entités identifiantes :

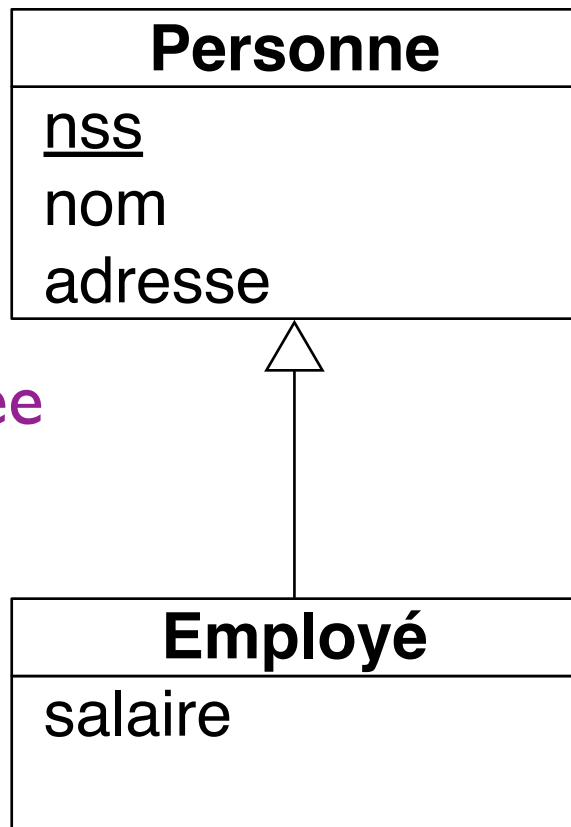


**Remarque :** dans cet exemple il n'y a pas d'attributs discriminants :  
la clef de l'entité faible est totalement externe : Université et Discipline

# Spécialisation

- Dans un diagramme E/R les instances de toutes les entités et toutes les associations sont disjointes entre elles par défaut
- La **spécialisation** (aussi appelé **héritage** ou **généralisation**) est un mécanisme pour spécifier des relations de sous-ensemble entre les instances de différentes entités

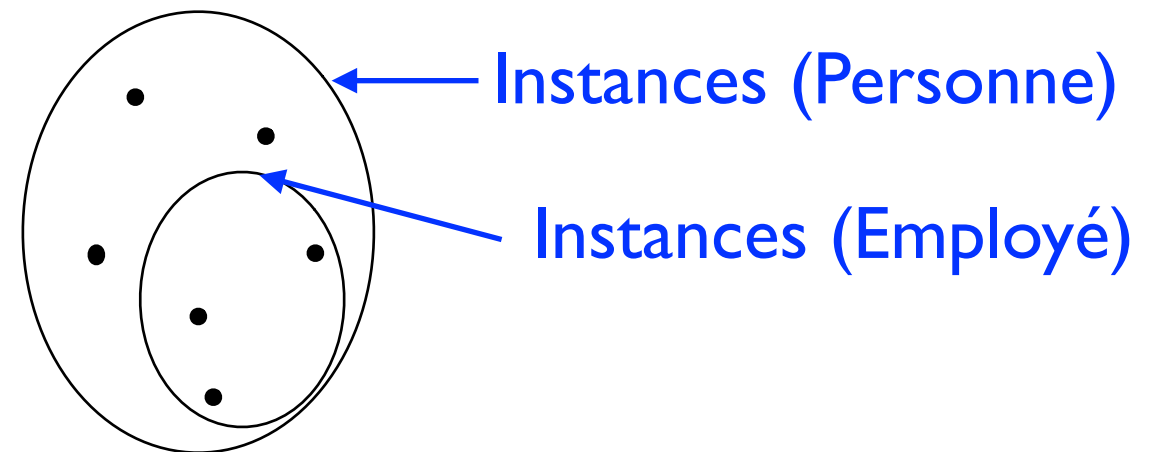
**Syntaxe :**



aussi appelée  
ISA

**Sémantique :**

Instances (Employé)  $\subseteq$  Instances (Personne)

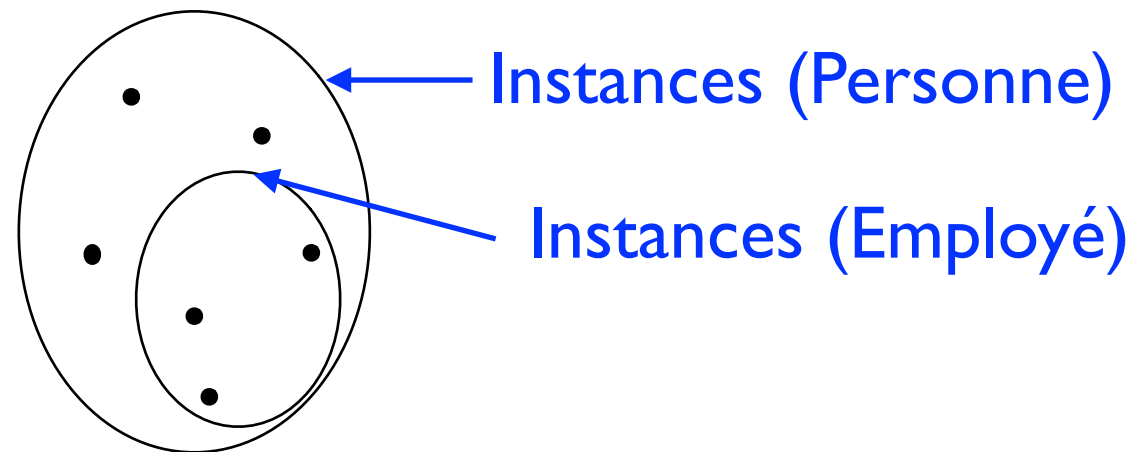
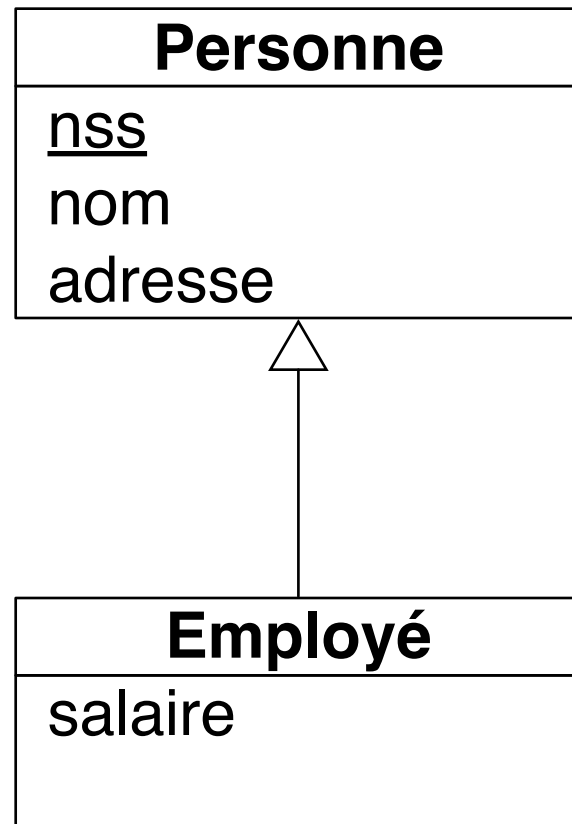


- Utilité : identifier un sous-ensemble des instances d'une entité avec des **propriétés (attributs) spécifiques** (i.e qui ne concernent pas les autres instances de la même entité)
  - ▶ Un employé est une personne qui a en plus un salaire



# Spécialisation : héritage des attributs

- La spécialisation introduit un mécanisme de classification des instances des entités, similaire à l'héritage des classes d'objets dans les types de données
  - ▶ spécialisation  $\leftrightarrow$  relation classe / sous-classe
- De façon similaire aux classes, la spécialisation implique l'héritage des attributs :



Pour chaque attribut  $A$  de **Personne**

$A: \text{Instances}(\text{Personne}) \rightarrow D$

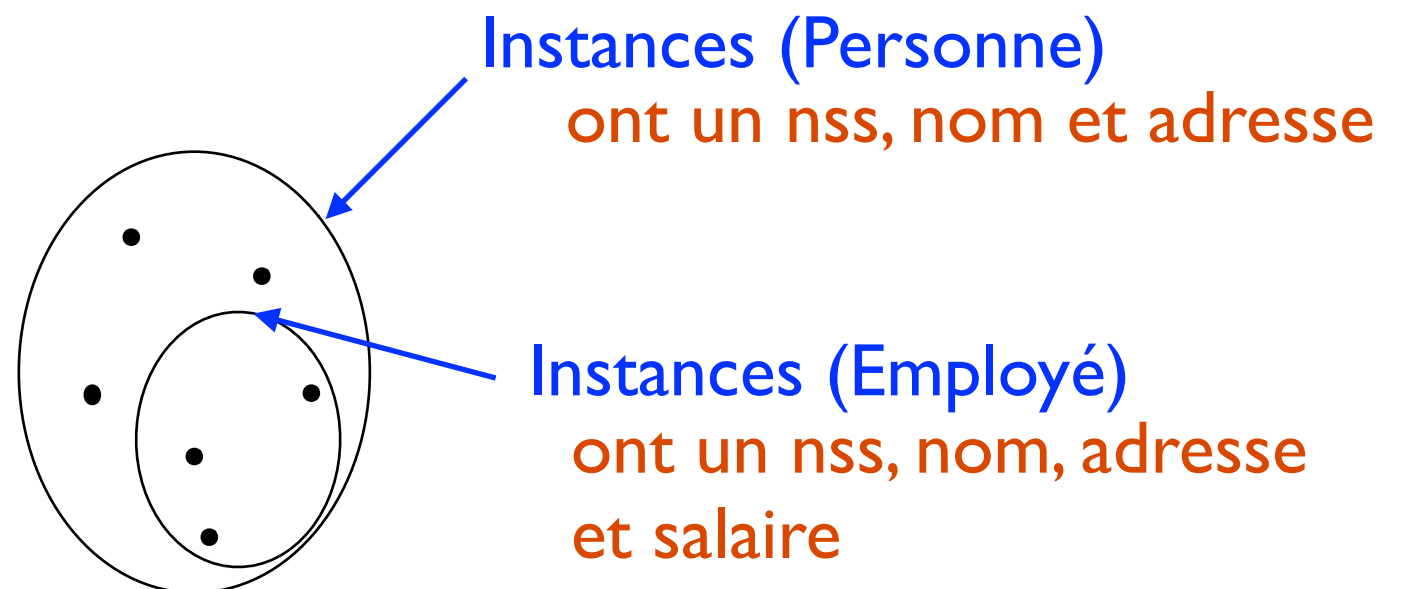
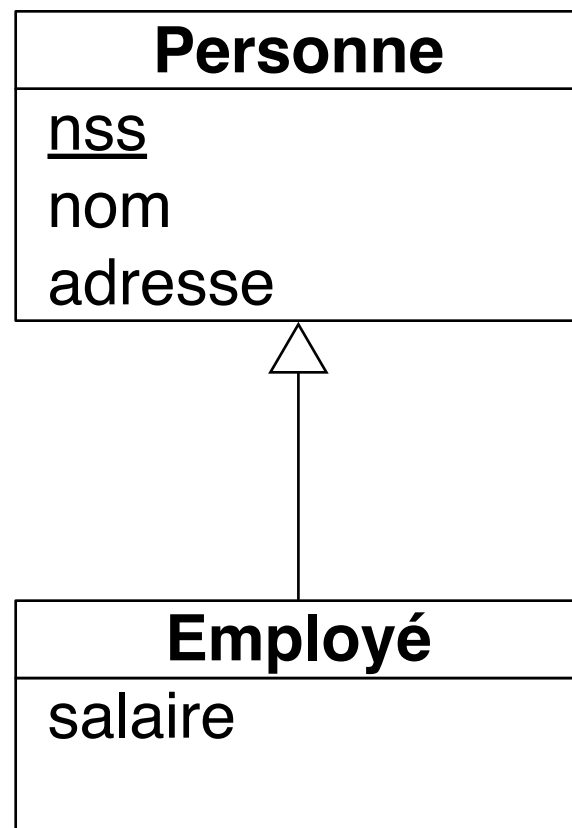
$\Rightarrow A$  est défini également sur les employés

$\Rightarrow A$  est un attribut de l'entité **Employé**

en revanche  $\text{salaire} : \text{Instances}(\text{Employé}) \rightarrow R$

# Spécialisation : héritage des attributs

- En présence de spécialisation **tous les attributs de l'entité mère sont aussi attributs de l'entité fille** (on dit qu'ils sont **hérités** par l'entité fille)
  - ▶ i.e. un employé est une personne donc il a un nss, nom et adresse
- En plus l'entité fille peut avoir d'autres attributs spécifiques

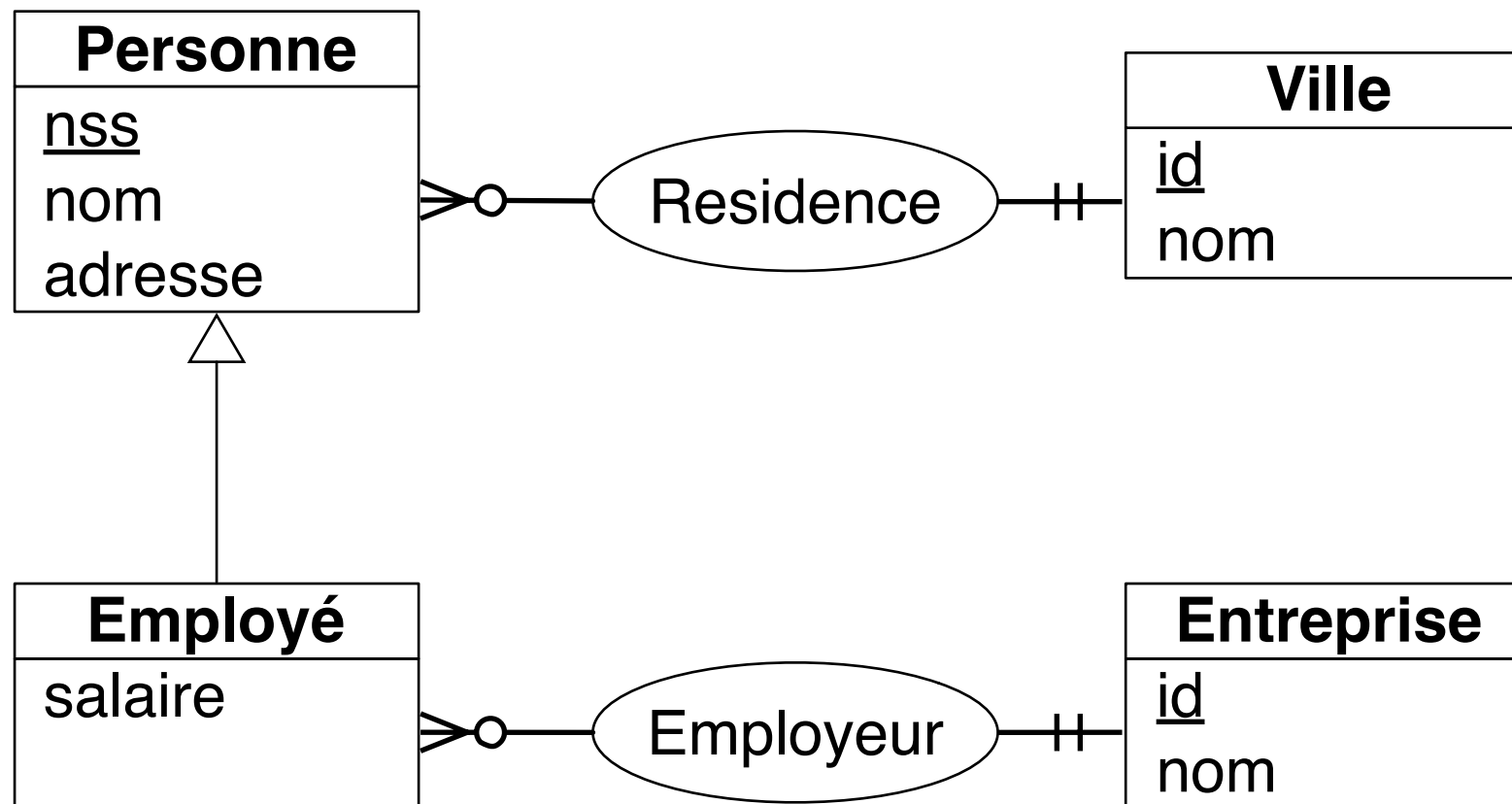


- Conséquence : l'entité fille **hérite la clef primaire** de l'entité mère (on ne spécifie donc pas de clef primaire pour les entités filles)

## Spécialisation : héritage des associations

- Si l'entité mère participe à des associations, ses entité filles y participent automatiquement :

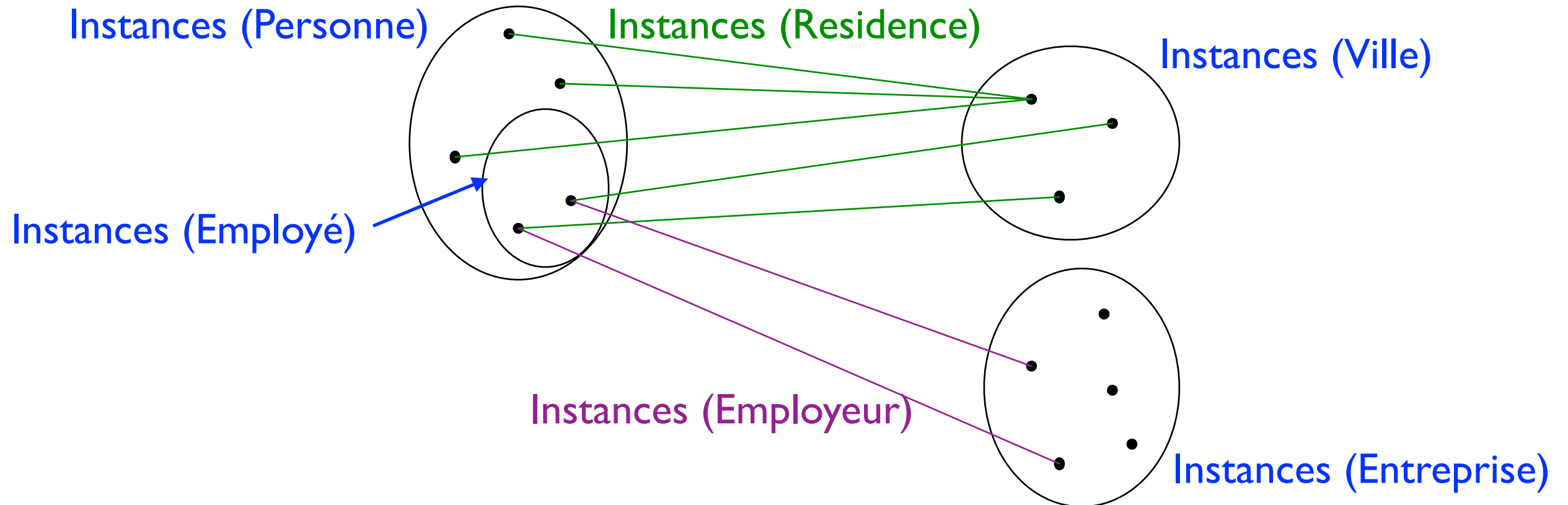
### Exemple



- ▶ Chaque personne réside dans une ville
- ▶ un employé est une personne, donc chaque employé réside dans une ville
- ▶ en plus chaque employé travaille dans une entreprise

# Spécialisation : héritage des associations

- Du point de vue des instances :

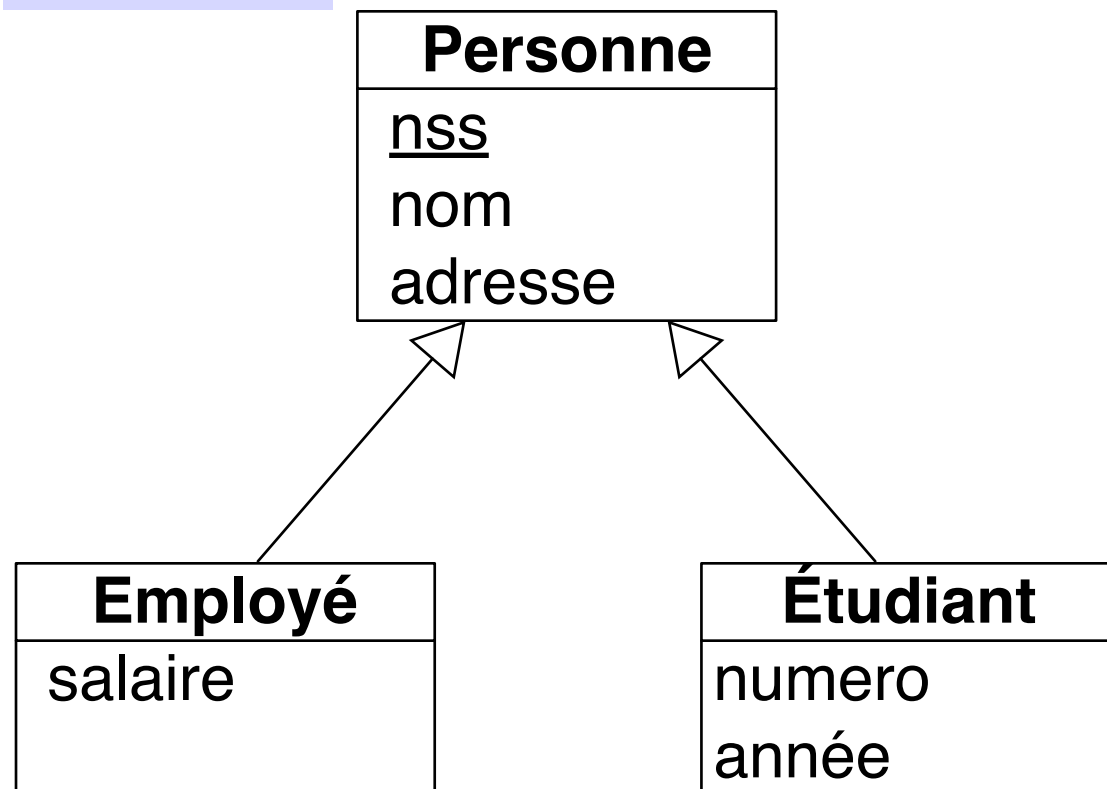


- ▶ Chaque personne réside dans une ville
- ▶ un employé est une personne, donc chaque employé réside dans une ville
- ▶ en plus chaque employé travaille dans une entreprise

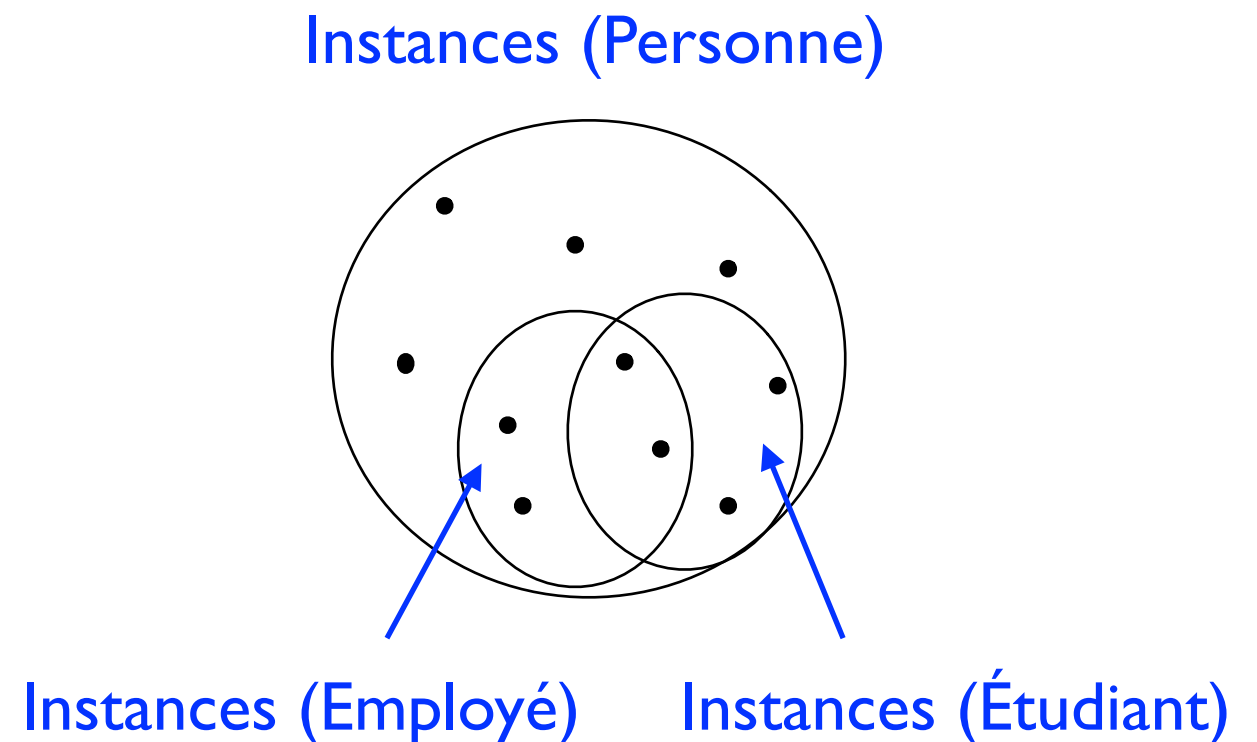
# Spécialisation

- Un même entité peut avoir plusieurs spécialisations (pas de limite sur le nombre)

## Syntaxe :



## Sémantique :

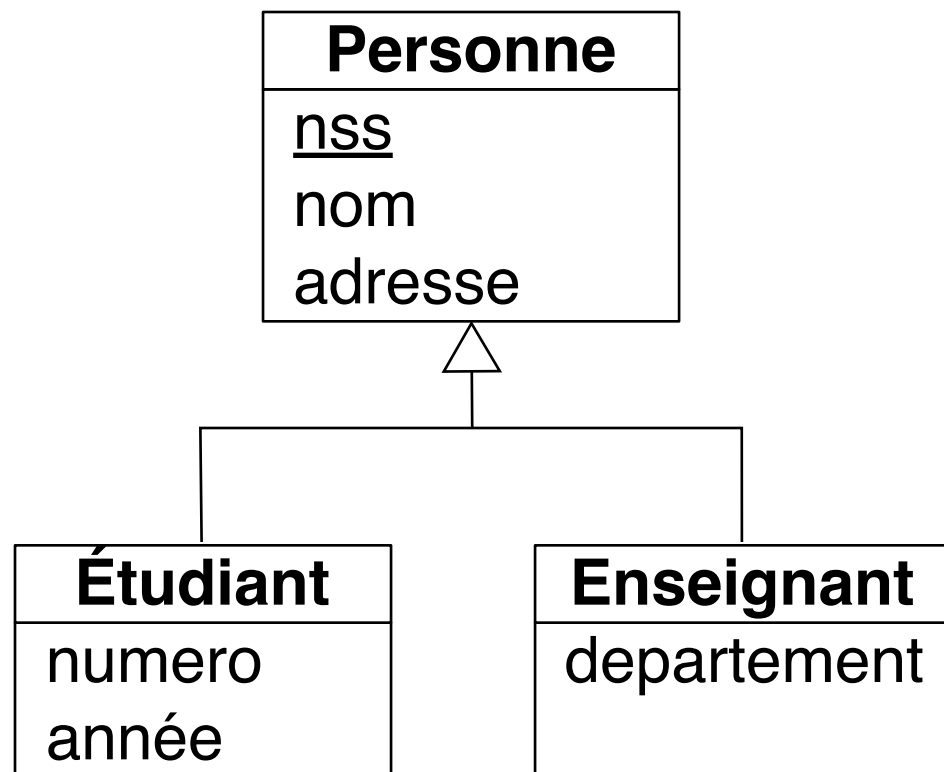


- Remarque : aucune contrainte sur les sous-ensembles :
  - ▶ ils peuvent avoir une intersection (les étudiants qui travaillent) ou pas
  - ▶ Ils peuvent être des sous-ensembles propres (il existe des personnes qui ne sont ni étudiants ni employés) ou pas

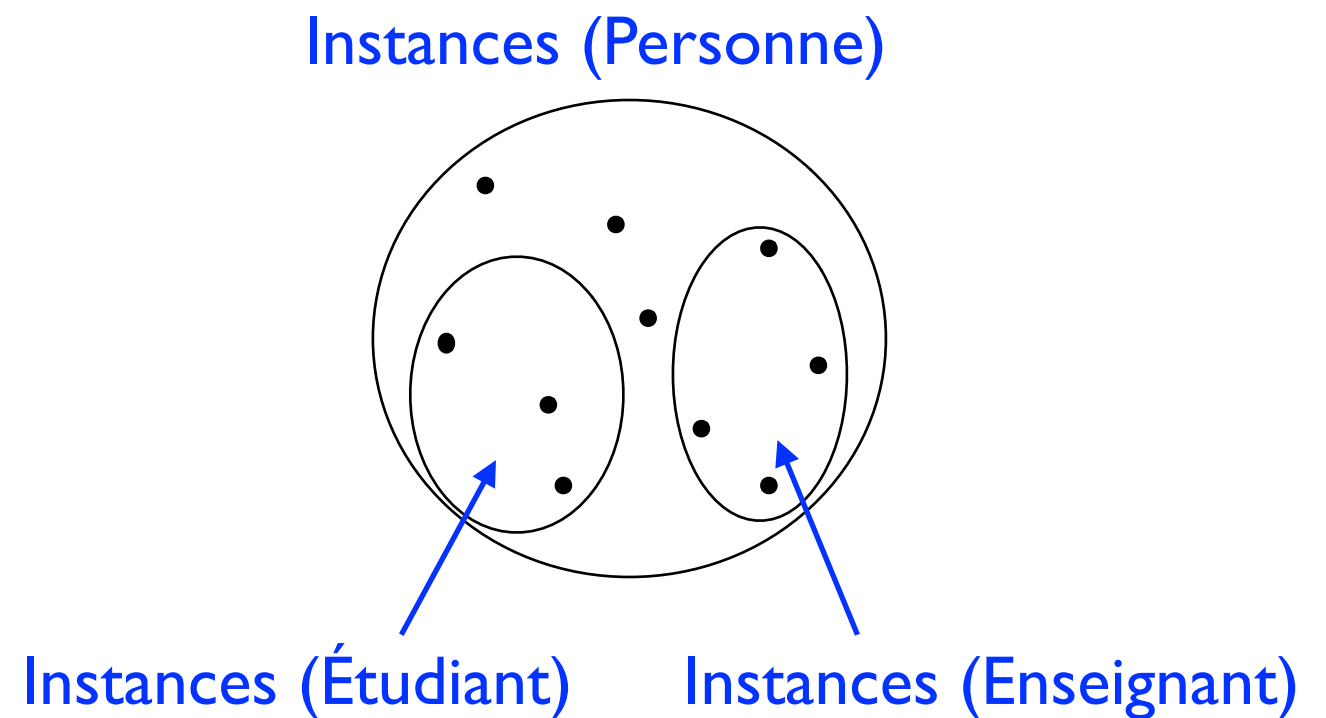
# Spécialisation avec contraintes

- Des formes particulières de spécialisation permettent d'ajouter des contraintes
  - Spécialisation disjointe

## Syntaxe :



## Sémantique :

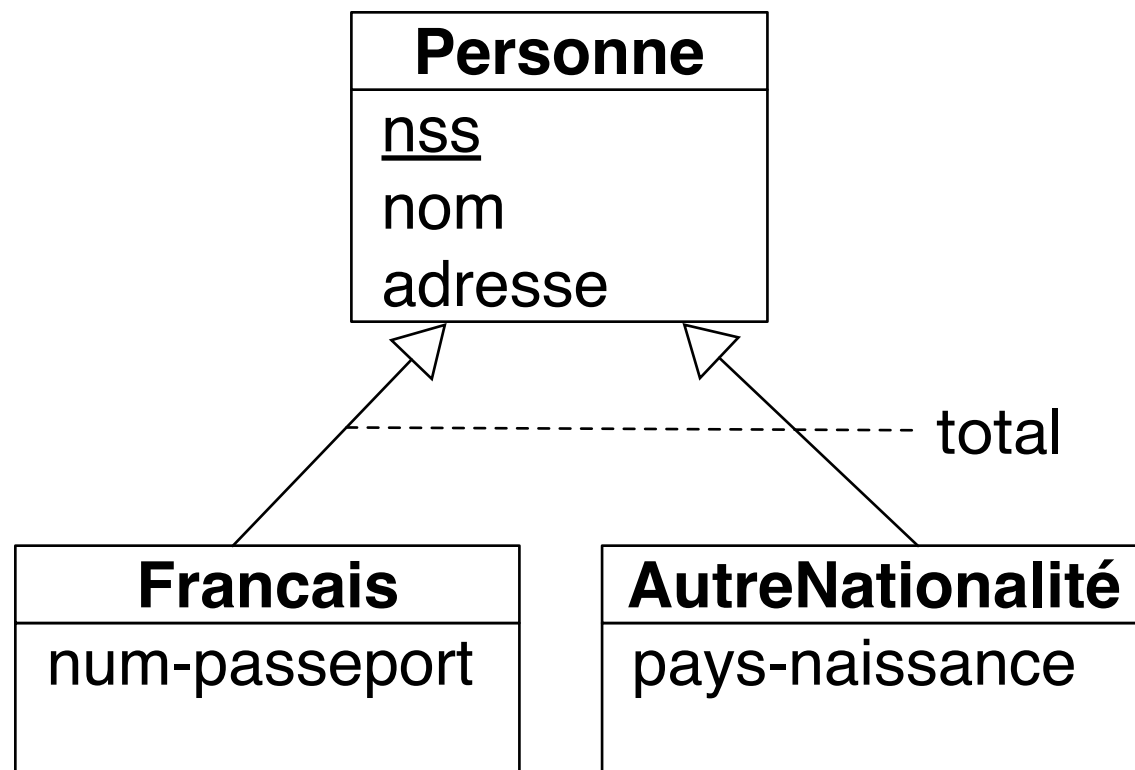


$\text{Instances (Étudiant)} \subseteq \text{Instances (Personne)}$   
 $\text{Instances (Enseignant)} \subseteq \text{Instances (Personne)}$   
 $\text{Instances (Étudiant)} \cap \text{Instances (Enseignant)} = \emptyset$

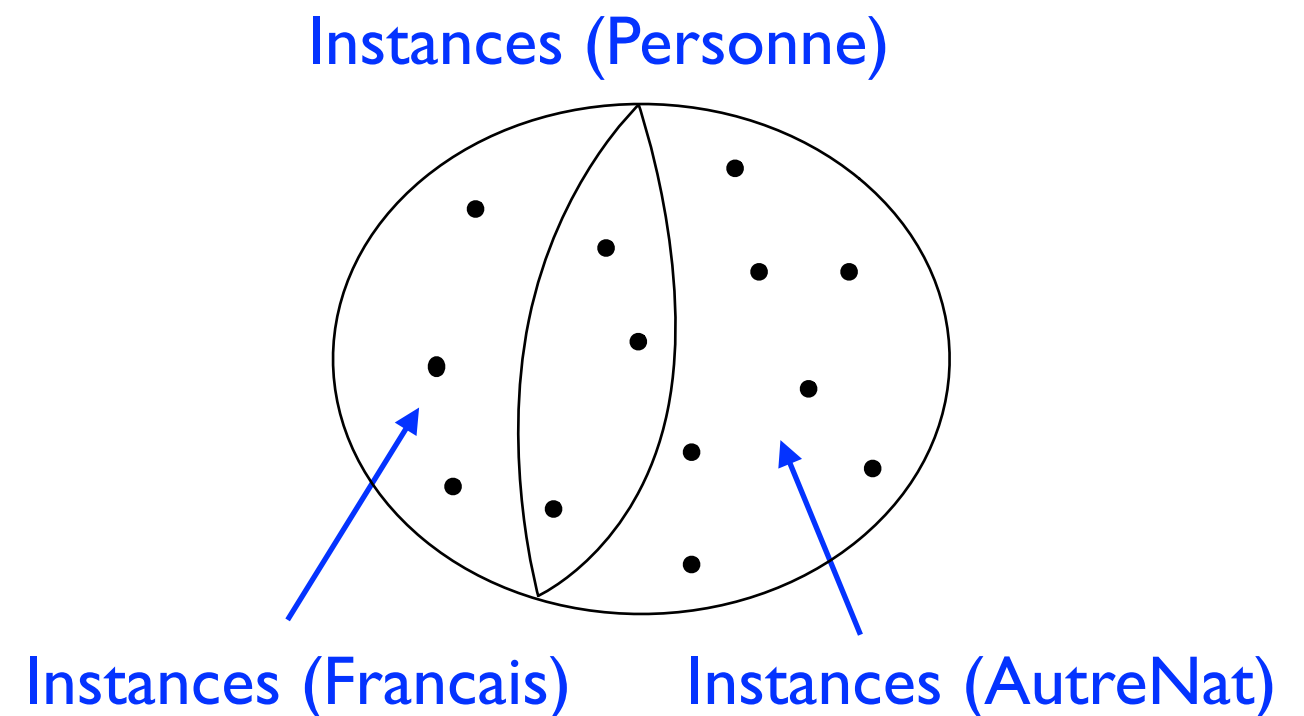
# Spécialisation avec contraintes

- Des formes particulières de spécialisation permettent d'ajouter des contraintes
  - Spécialisation totale

## Syntaxe :



## Sémantique :



$\text{Instances (Francais)} \subseteq \text{Instances (Personne)}$

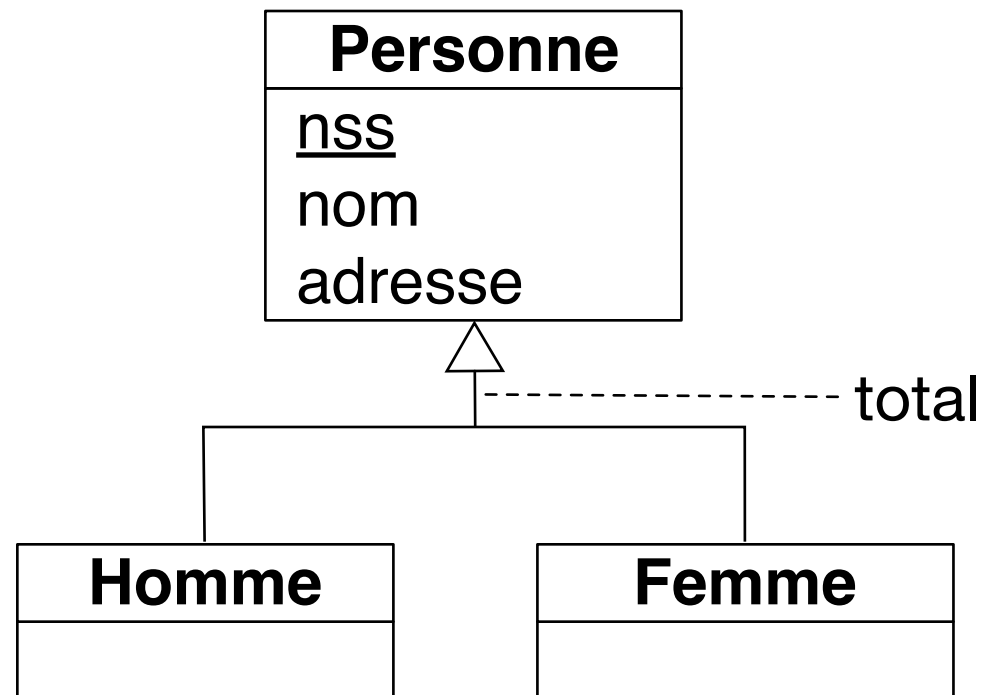
$\text{Instances (AutreNat)} \subseteq \text{Instances (Personne)}$

$\text{Instances (Francais)} \cup \text{Instances (AutreNat)} = \text{Instances (Personne)}$

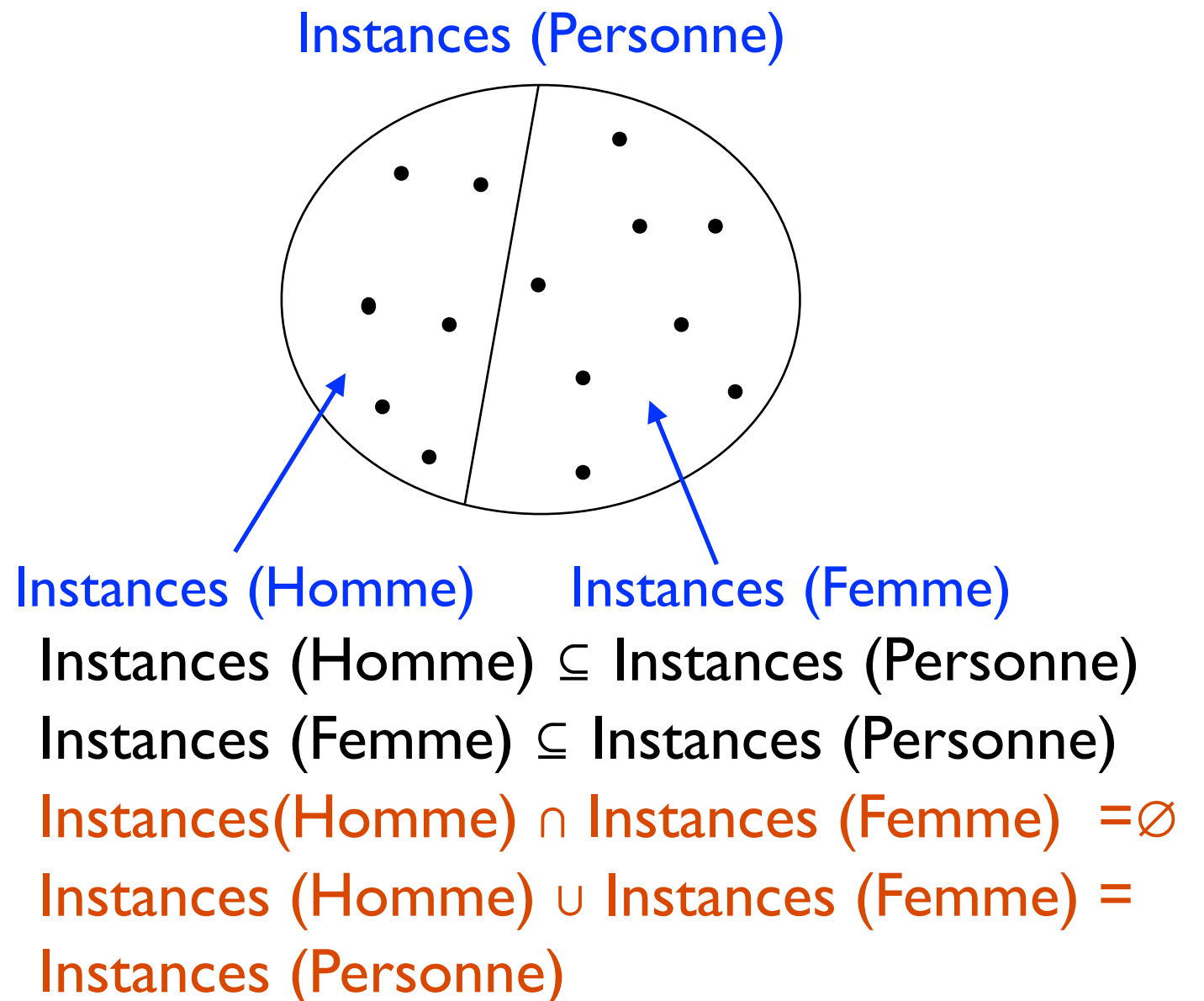
# Spécialisation avec contraintes

- Des formes particulières de spécialisation permettent d'ajouter des contraintes
  - ▶ Spécialisation disjointe totale

## Syntaxe :



## Sémantique :

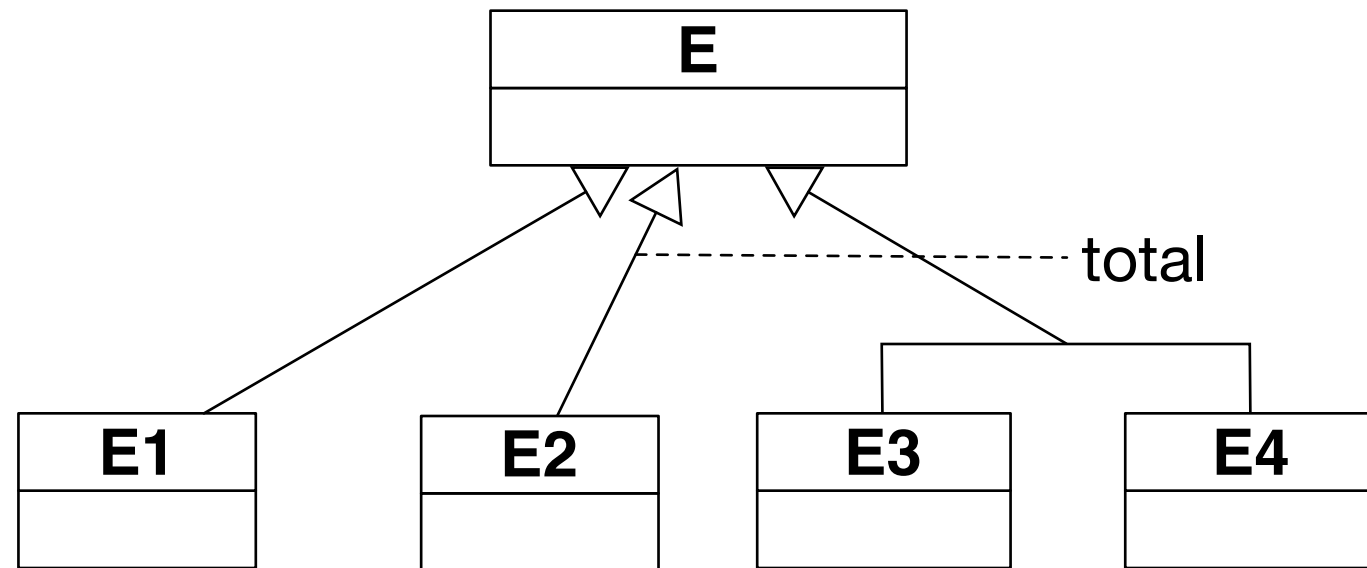




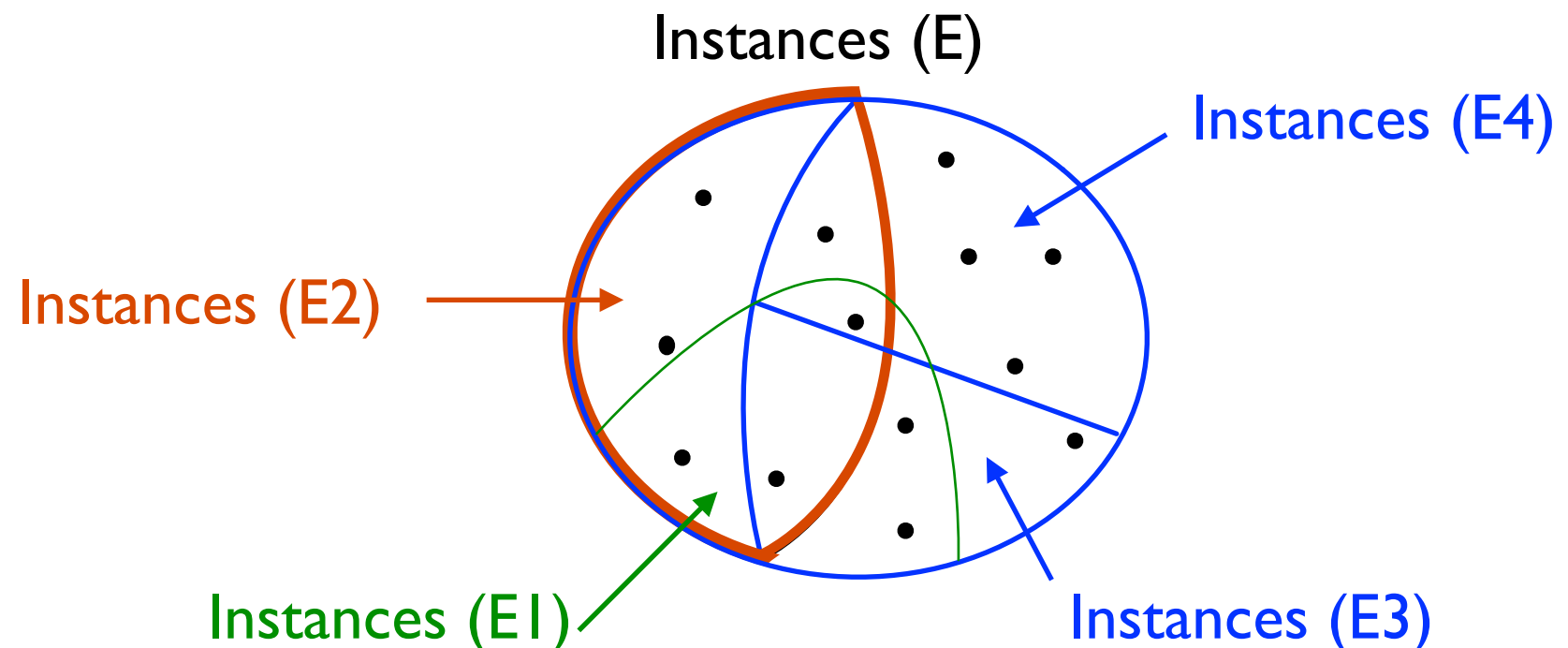
# Spécialisation avec contraintes

- Des formes particulières de spécialisation permettent d'ajouter des contraintes
  - la **contrainte de totalité** peut s'appliquer à un nombre arbitraire de spécialisations de la même entité

**Syntaxe** (exemple):

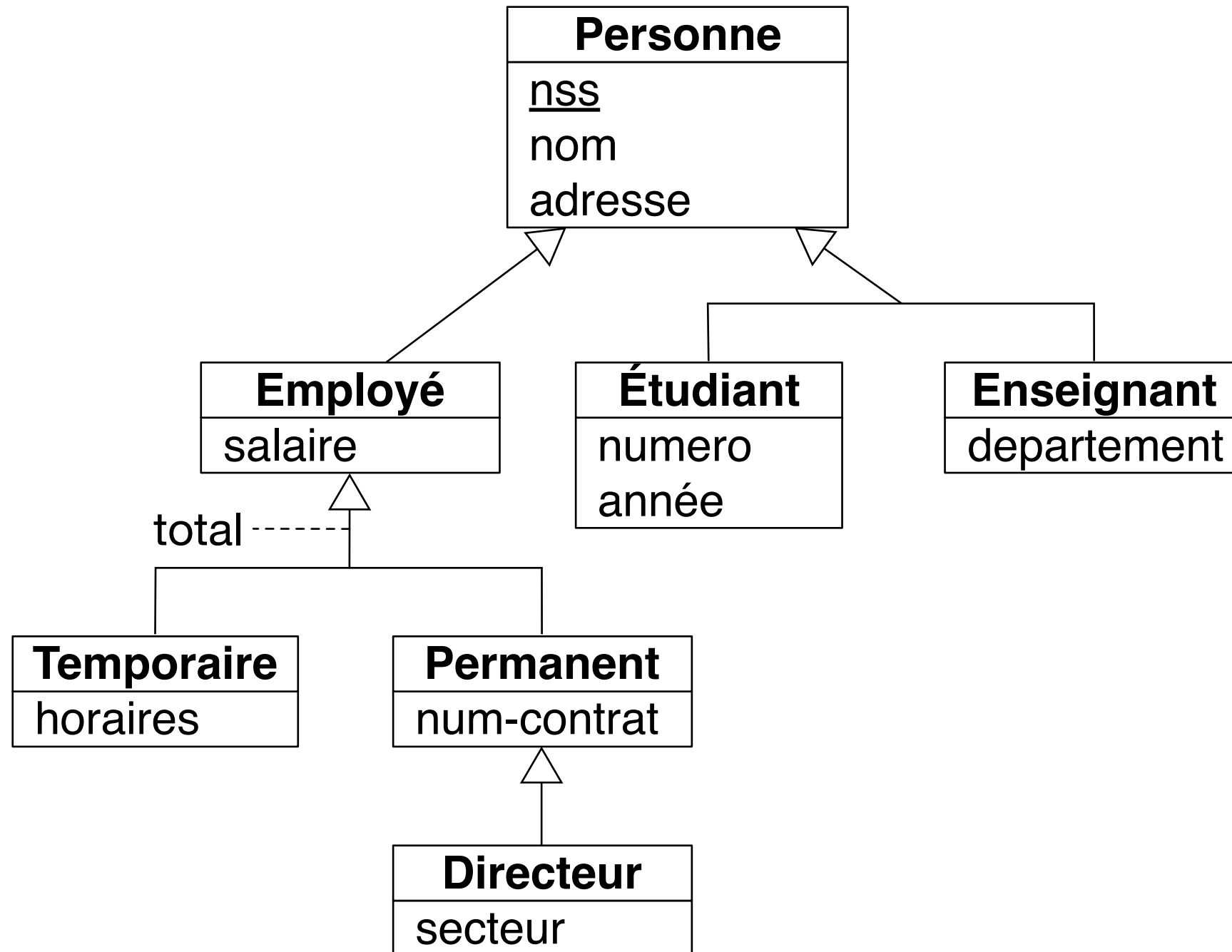


**Sémantique :**



# Hiérarchies de spécialisations

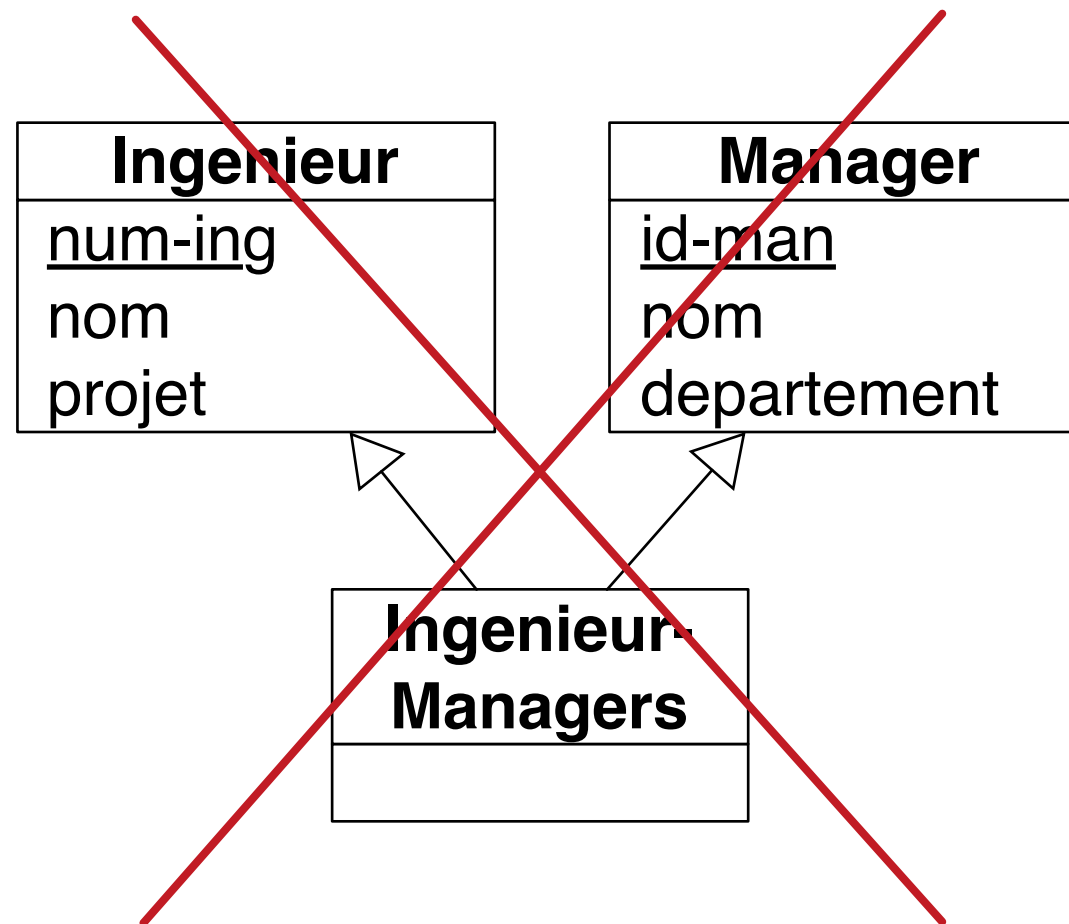
- Les spécialisations peuvent former une hiérarchie



- L'héritage s'applique tout au long de la hiérarchie. E.g. chaque directeur a un nss, nom, adresse, salaire, numéro-contrat et secteur

# Hiérarchies de spécialisations

- Dans la hiérarchie l'**héritage multiple est interdit** : chaque entité peut avoir au plus une entité mère



# Un exemple complet de modélisation E/R

## Un exemple complet de modélisation E/R

Représenter les informations suivantes concernant les vols d'une compagnie aérienne.

La compagnie propose des vols, chacun avec un numéro qui l'identifie (par exemple AF326), une date, un horaire de départ (8:00) et un horaire d'arrivée (12:00), un aéroport de départ (e.g. Paris CDG) et un aéroport d'arrivée (e.g. New York JFK).

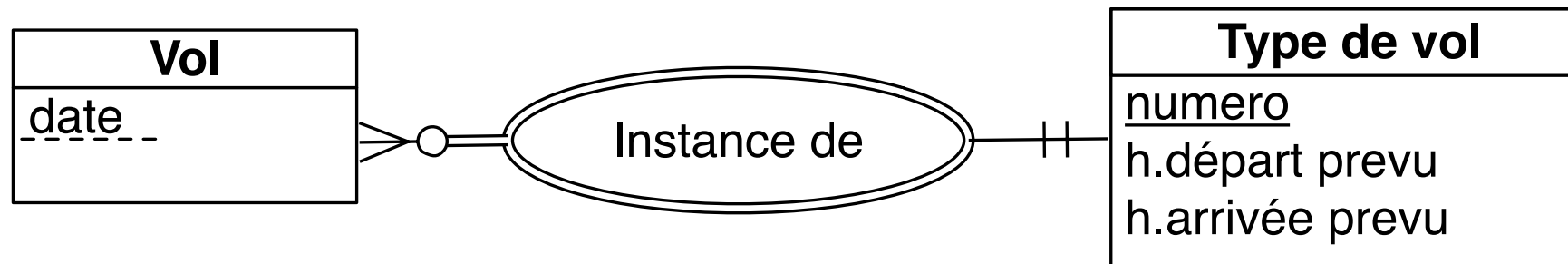
Il y a des vols nationaux et internationaux. Les vols internationaux peuvent avoir un ou plusieurs arrêts.

Pour les vols qui ont déjà eu lieu on représente également les horaires effectifs de départ et d'arrivée (il peut y avoir des retards/anticipations par rapport aux horaires prévus, e.g. pour le vol ci-dessus 8:05 et 12:07). Pour les vols futurs on représente le nombre de places disponibles.

# Un exemple complet de modélisation E/R

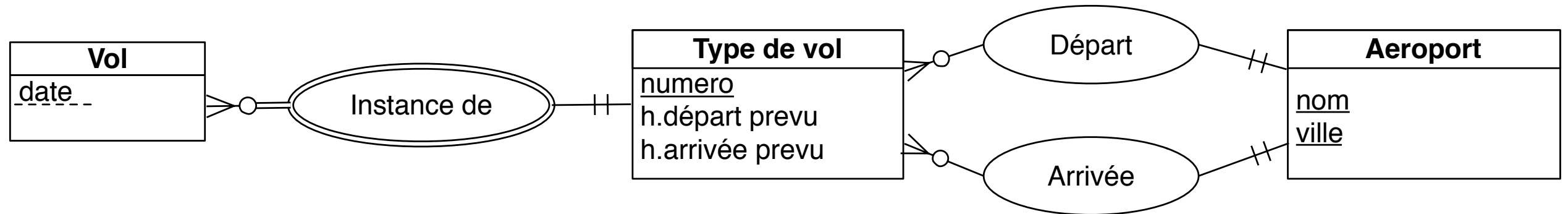
Remarque. Deux concepts distincts de vol :

- **type de vol** : avec un numéro, des horaires prévus, des aéroports de départ, d'arrivée et possiblement d'arrêt;
- **vol effectif** : une instance d'un certain type de vol, qui a lieu à une date, avec des horaires effectifs (s'il a déjà eu lieu) ou des places disponibles (s'il n'a pas encore eu lieu).



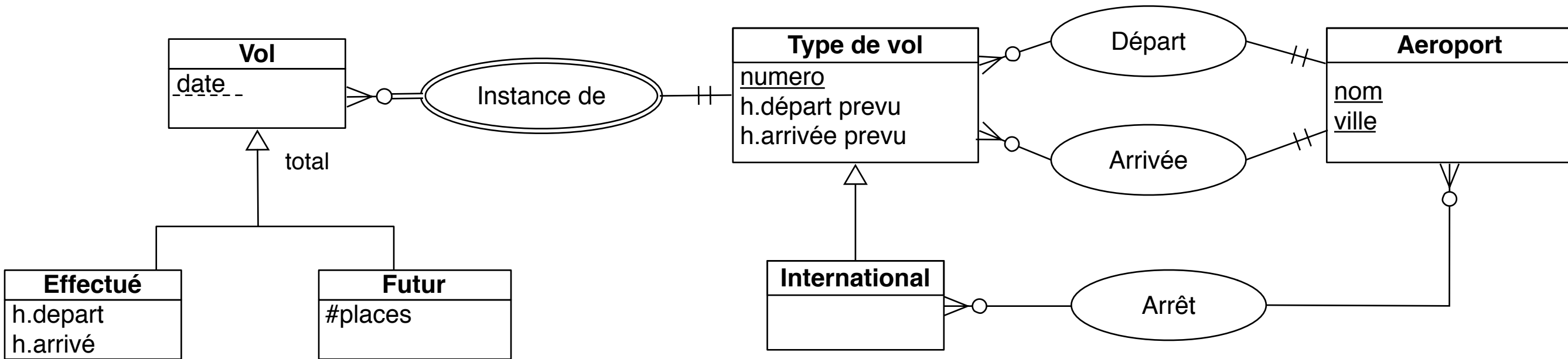
# Un exemple complet de modélisation E/R

- Aéroport de départ et d'arrivée sont des propriétés du type de vol :



# Un exemple complet de modélisation E/R

- La distinction “vol national / international” s’applique au type de vol;
- La distinction “effectué / futur” s’applique au vol effectif



+ **contrainte externe** : la date de chaque vol effectué est  $\leq$  à la date de chaque vol futur

- Remarque : en l’absence de propriétés et contraintes concernant les vols nationaux, on ne les représente pas



## Passage au modèle relationnel

- Un schéma relationnel des données peut être obtenu du diagramme E/R
- Deux phases :
  - ▶ Restructuration du diagramme E/R
  - ▶ Traduction dans un schéma relationnel

# Restructuration du diagramme E/R

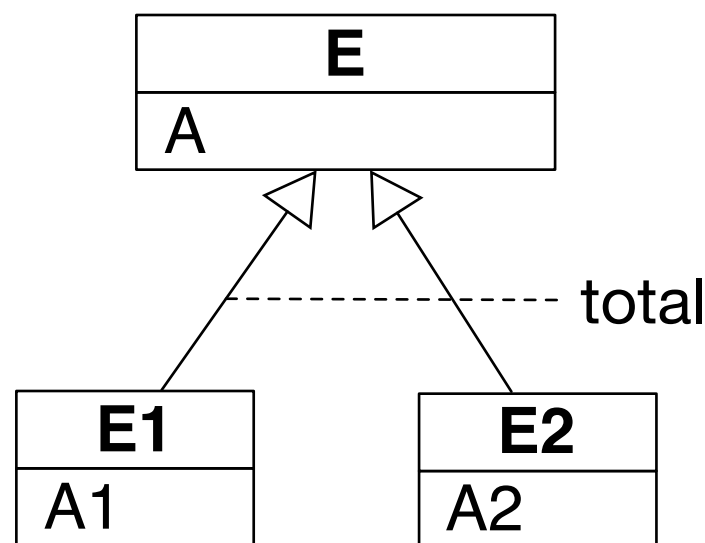
- Les spécialisations ne peuvent pas être traduites directement dans le modèle relationnel
- Le diagramme E/R doit donc être restructuré avant la traduction en relationnel
- D'autres restructurations, motivées par une estimation du coût d'accès aux données peuvent être effectuées dans cette phase :
  - ▶ Exemples
    - introduction d'attributs redondants
    - partition d'une entité (ou association) en deux
    - fusion de plusieurs entités (ou associations)

## Restructuration des spécialisations

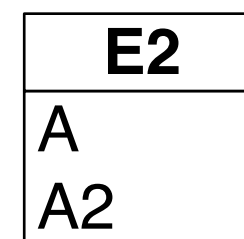
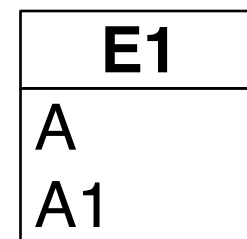
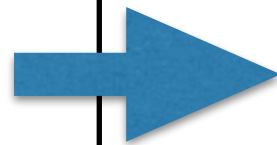
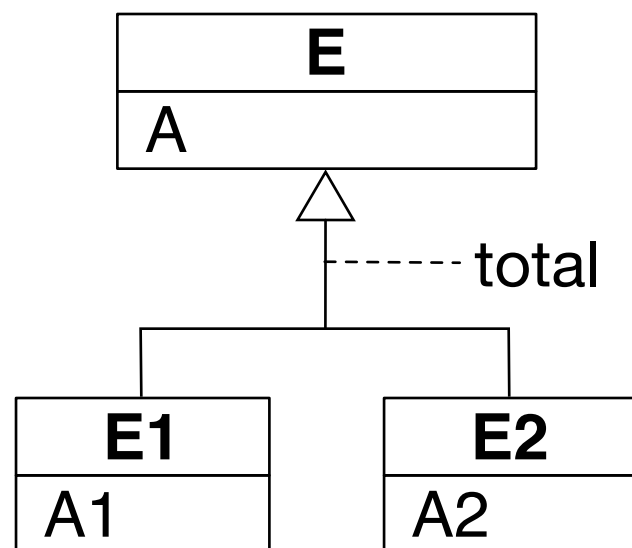
- Chaque spécialisation doit être éliminée du diagramme et remplacée par une reformulation en terme d'entités et associations sans héritage
- Certaines des contraintes exprimées par la spécialisation seront perdues dans cette phase
- Il y a plusieurs possibilités de restructuration et il faut choisir la plus appropriée.
- Trois restructuration typiques

# Restructuration des spécialisations

I) Eliminer l'entité mère (possible **uniquement si la spécialisation est totale**)



ou



À préférer si l'entité mère

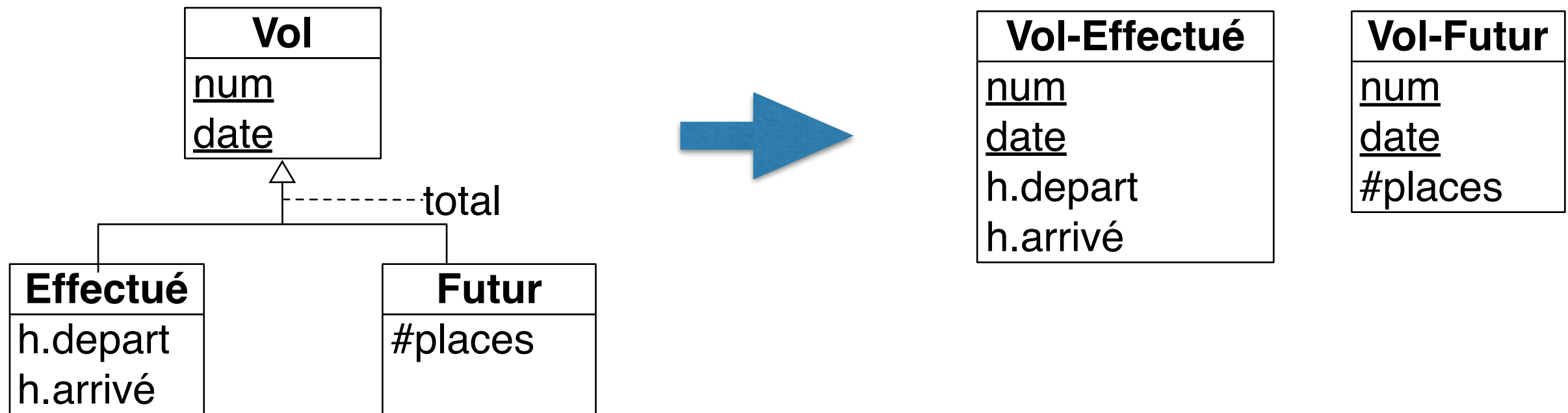
- a peu d'attributs
- participe à peu d'associations

et si, dans l'utilisation prévue des données, les opérations qui accèdent uniquement à **E** sont peu fréquentes

# Restructuration des spécialisations

I) Eliminer l'entité mère (possible **uniquement si la spécialisation est totale**)

## Exemple

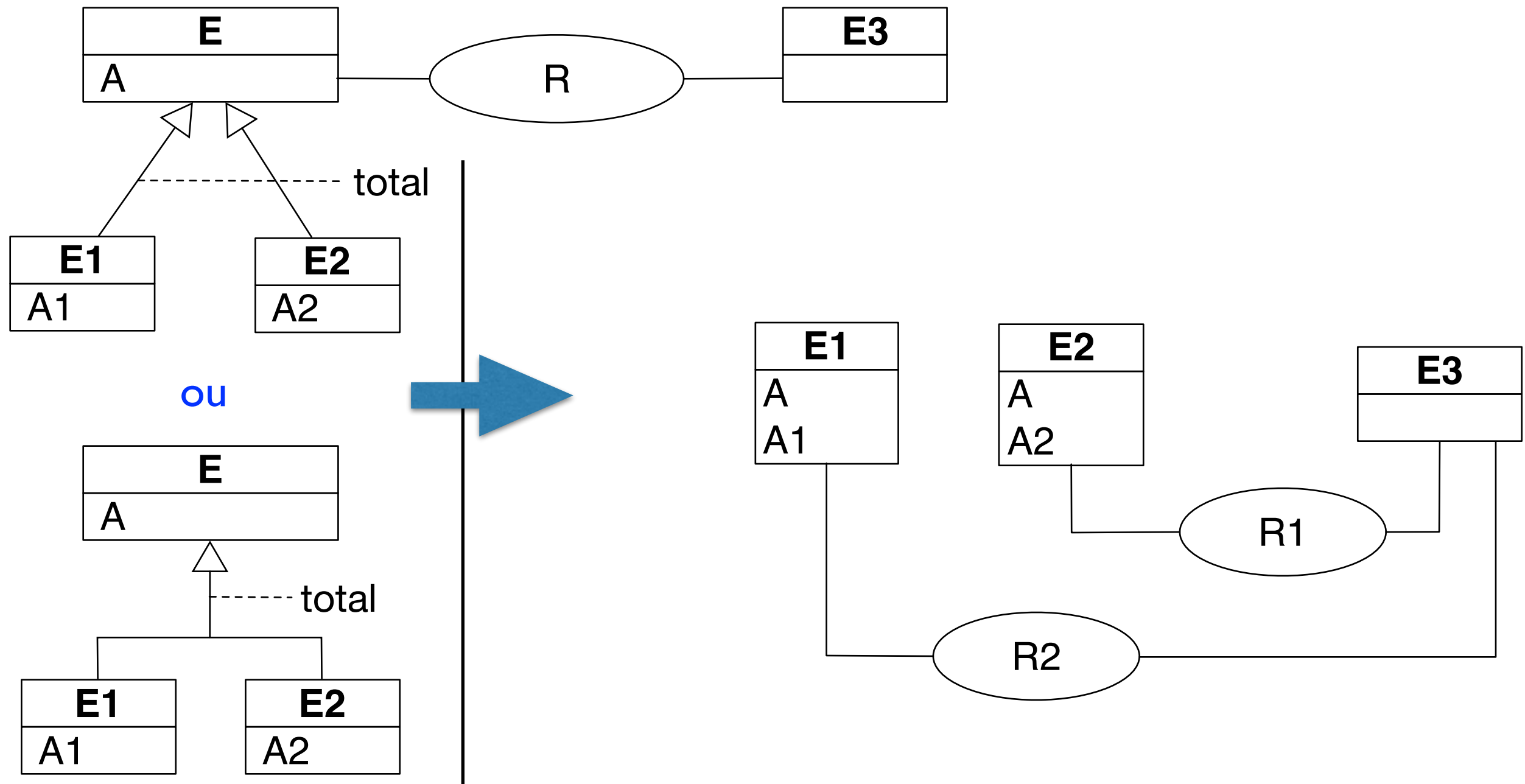


- On peut raisonnablement supposer que les opérations accèdent soit aux vols effectués (e.g. pour connaître leur horaires effectifs) soit aux vols futurs (e.g. pour vérifier la disponibilité de places)

# Restructuration des spécialisations

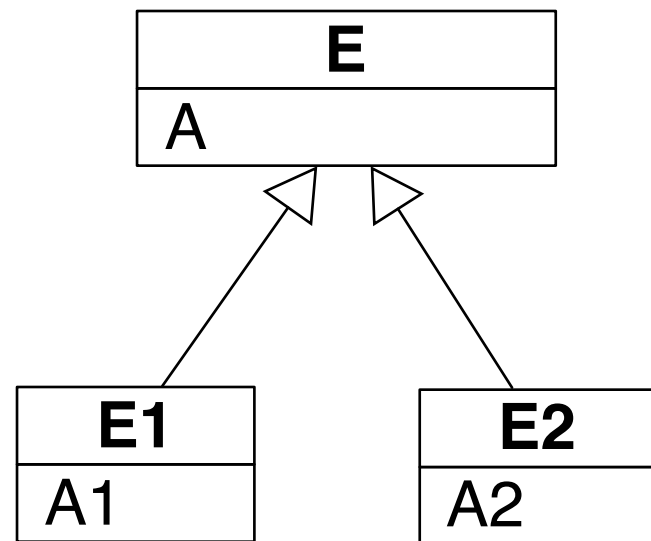
I) Eliminer l'entité mère (possible **uniquement si la spécialisation est totale**)

attention! les associations de E souvent être dédoublées

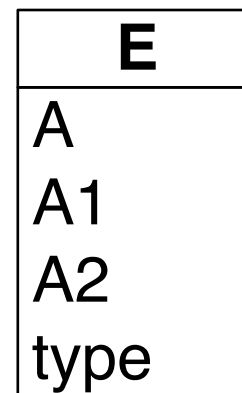
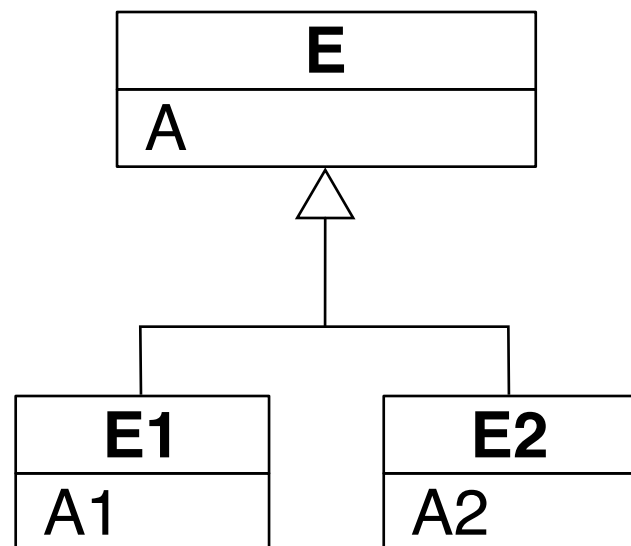


# Restructuration des spécialisations

## 2) Eliminer les entités filles



ou



**type** a valeurs :

- $\{E1, E2, E12, E\}$   
si la spécialisation est non-disjointe
- $\{E1, E2, E\}$   
si la spécialisation est disjointe

**E** n'est pas parmi les valeurs en cas de spécialisation totale

À préférer si les entités filles

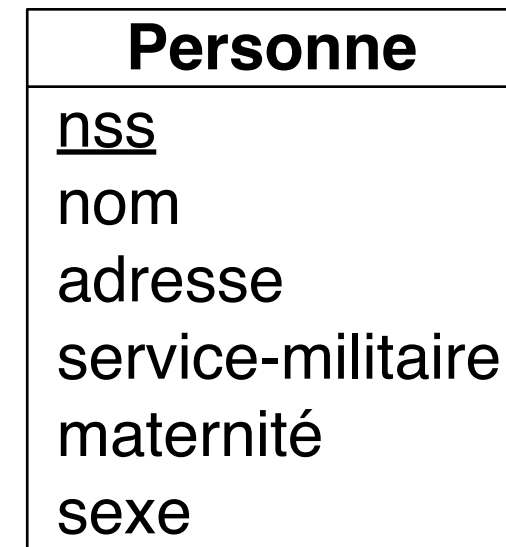
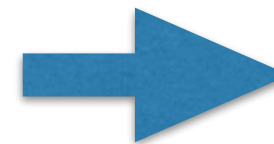
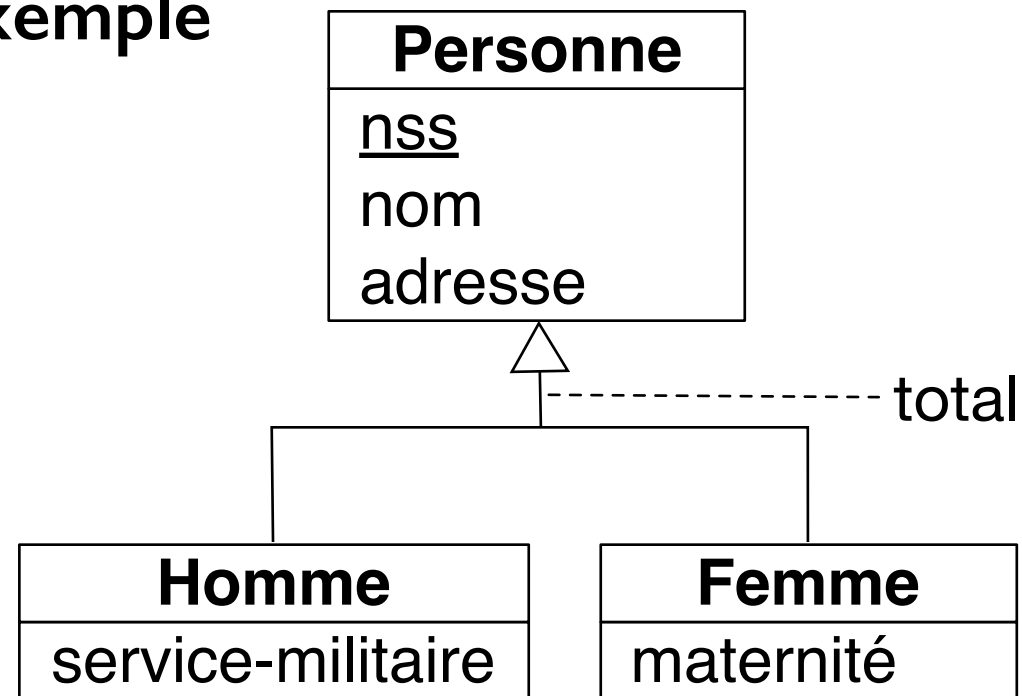
- ont peu d'attributs
- participent à peu d'associations

et si, dans l'utilisation prévue des données, les opérations qui accèdent uniquement à **E1** ou uniquement à **E2** sont peu fréquentes

# Restructuration des spécialisations

## 2) Eliminer les entités filles

### Exemple



sexe a valeurs {Homme, Femme}

Assumption : la plus part du temps les opérations accèdent aux personnes indépendamment du fait qu'ils soient homme ou femme

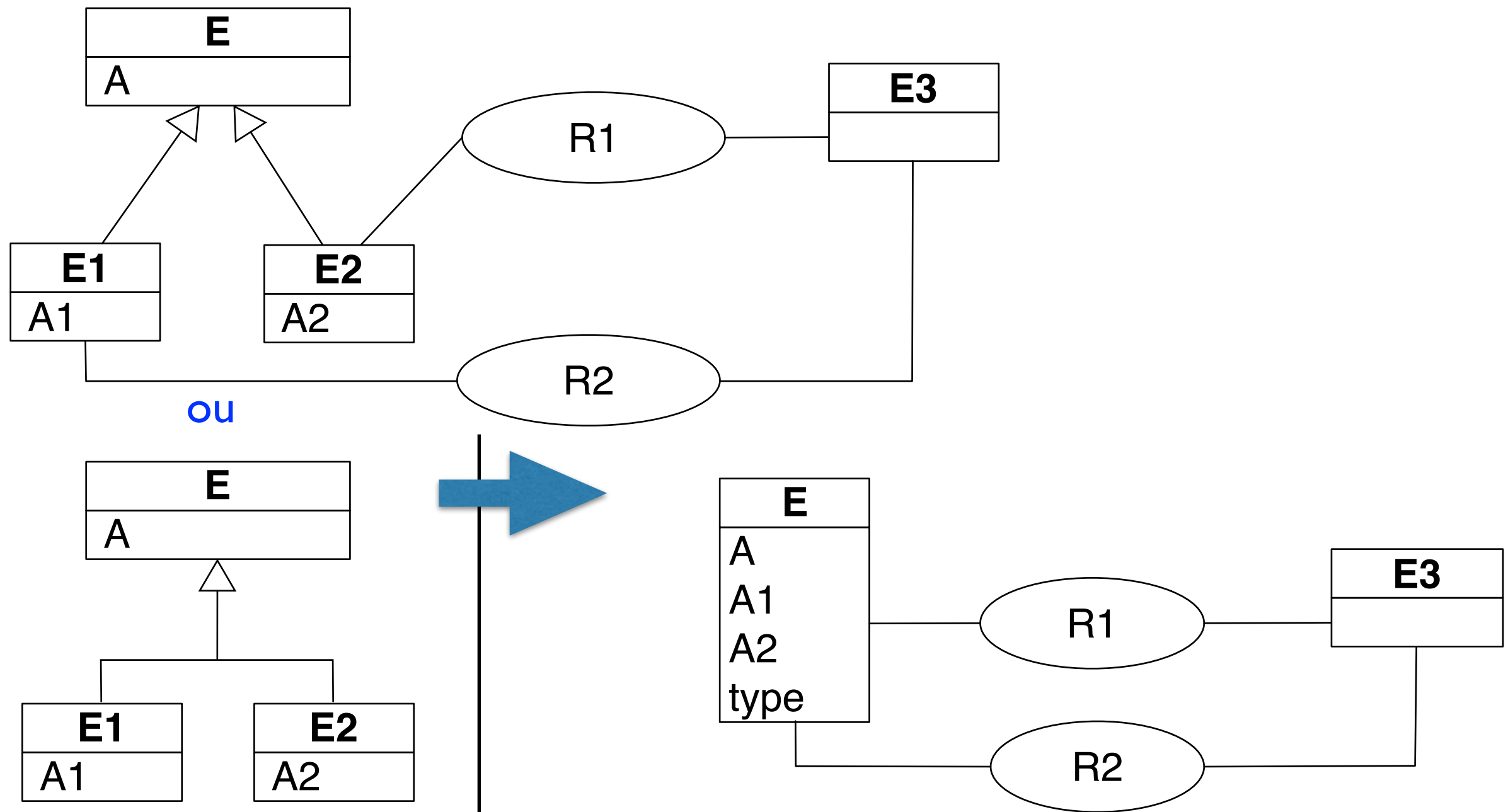
Inconvénient de l'alternative 2):  
les attributs spécifiques des entités filles (service militaire, maternité) ont des valeurs non significatifs (NULL une fois traduits dans le models relationnel) pour une partie des instances



# Restructuration des spécialisations

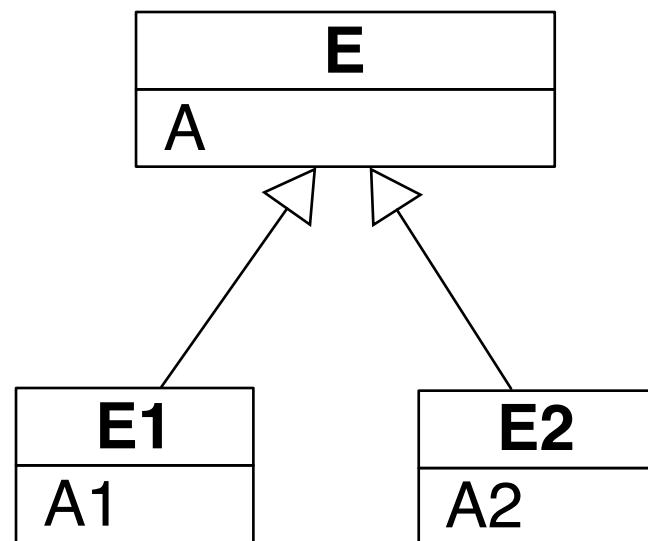
## 2) Eliminer les entités filles.

attention! les associations de E1 et E2 doivent être reportées sur E

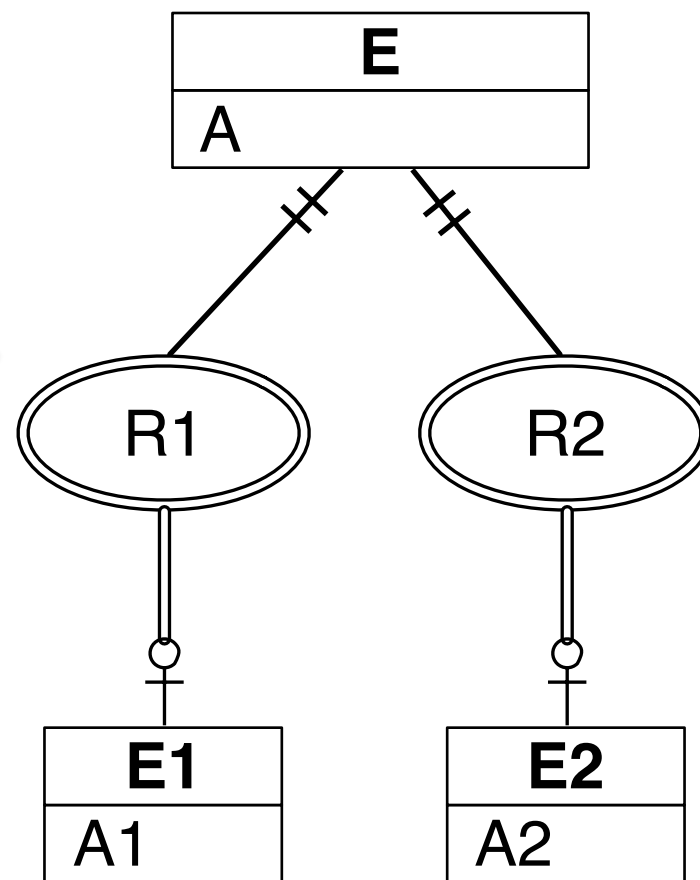
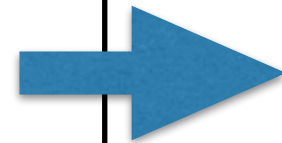
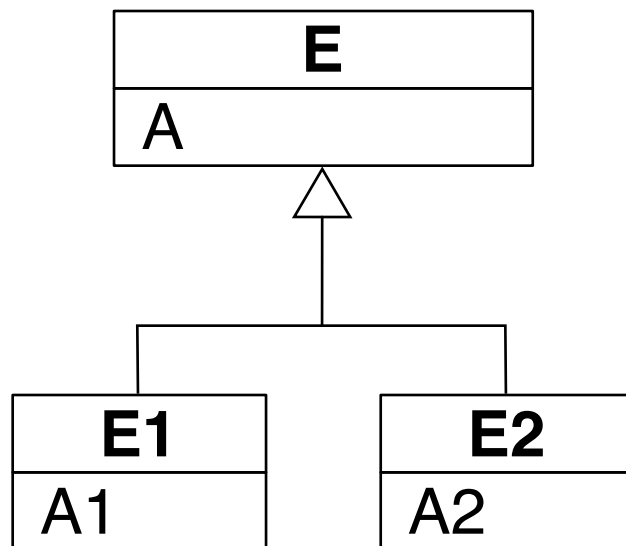


# Restructuration des spécialisations

3) Maintenir toutes les entités, simuler la spécialisation avec des associations



ou



**contrainte externe en cas de spécialisation**

**totale:**

chaque instance de **E** participe à **R1** ou à **R2**

**contrainte externe en cas de spécialisation**

**disjointe :**

aucune instance de **E** participe à la fois à **R1** et à **R2**

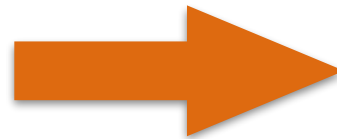
Option la plus générale : à utiliser quand les conditions d'applicabilité de 1) ou 2) ne sont pas satisfaites

# Traduction : diagramme E/R → schéma relationnel

- Règle générale
  - ▶ chaque entité est traduite en un schéma de relation
  - ▶ chaque association est traduite en
    - une schéma de relation
    - des contraintes de clefs étrangère
  - ▶ d'autres contraintes du schéma relationnel (inclusion, NOT NULL, ..) peuvent être déduites des contraintes de cardinalité du diagramme E/R
  - ▶ les contraintes externes plus complexes du diagramme E/R sont traduites par des assertions ou conditions de CHECK (cf. plus tard)

# Entités

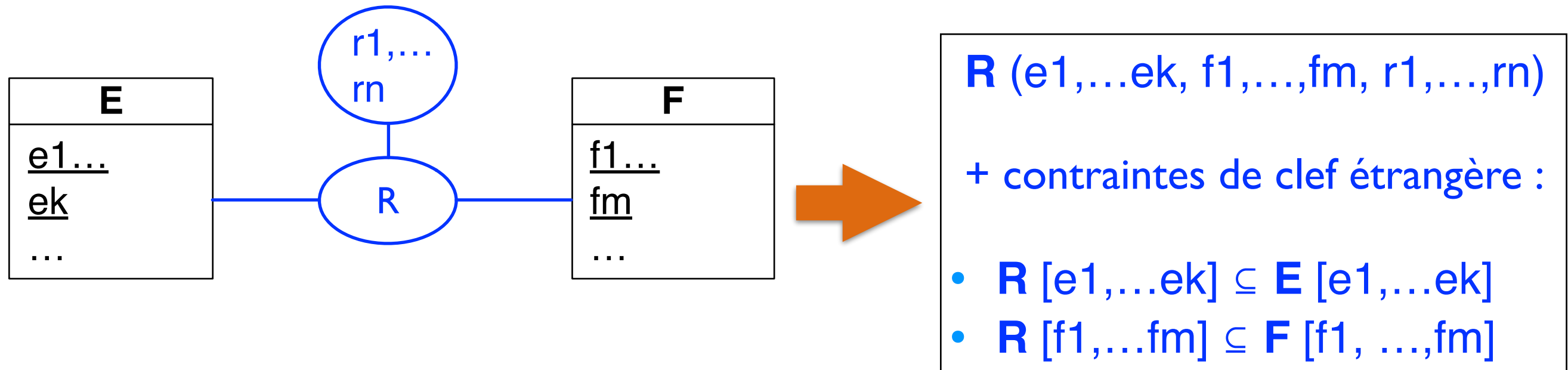
E
<u>c1...</u> <u>ck</u>
a1... an



**E** (c1, ...ck, a1,...,an)

- Cette traduction **peut être en suite modifiée** au fur et à mesure que les associations auxquels E participe sont traduites, cf. plus tard
  - ▶ cela est vrai **en particulier si E est une entité faible** et c1..ck son discriminant

## Associations dont aucun coté a cardinalité 1

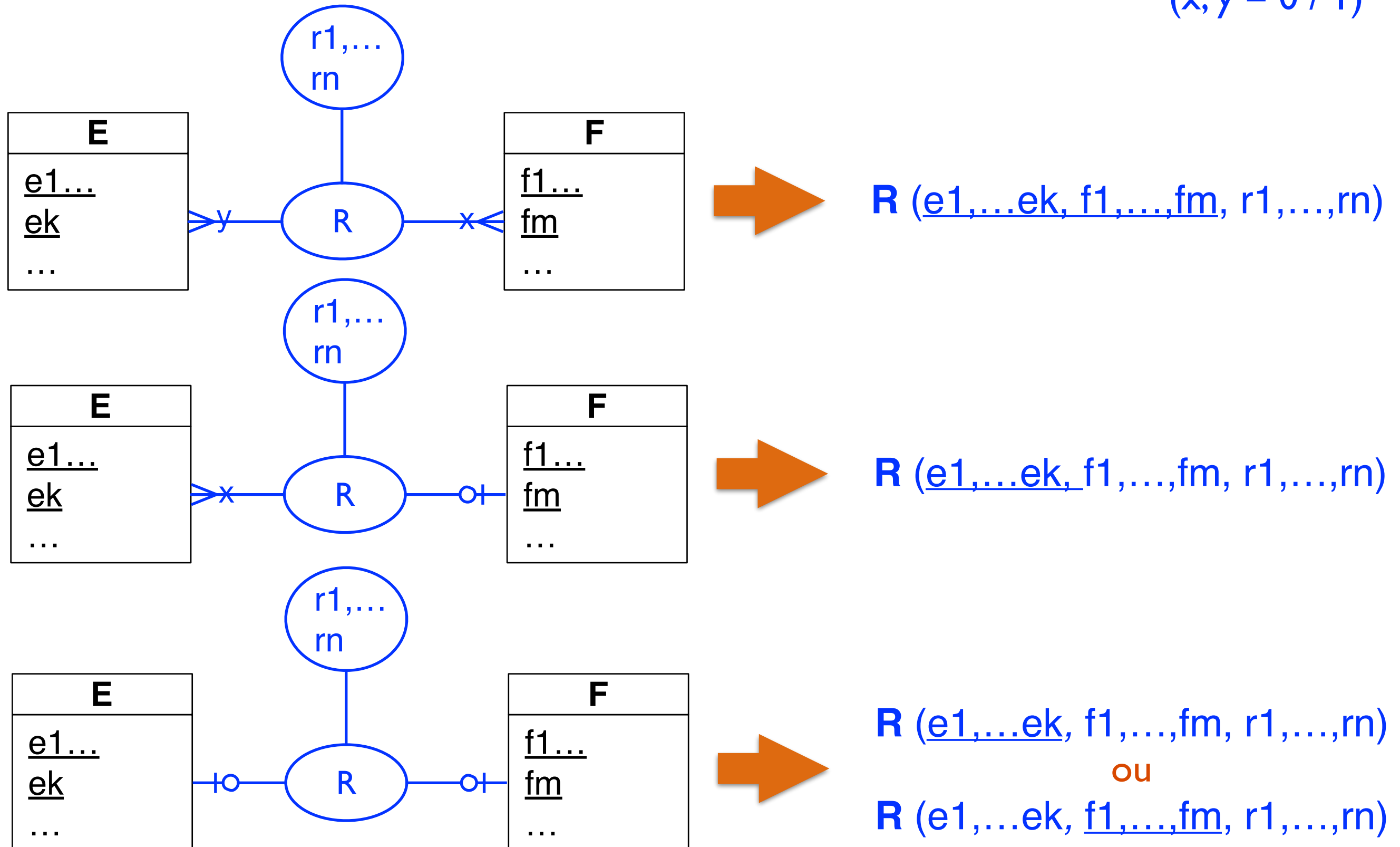


- Remarque. Celle ci-dessus est une traduction de  $R$  uniquement ( $E$  et  $F$  doivent être traduites indépendamment)
- La clef primaire de la relation  $R$  dépend des contraintes de cardinalité maximales (cf. prochain transparent)
- les contraintes de cardinalité minimales impliquent des contraintes d'inclusion additionnelles sur le schéma relationnel (cf. dans deux transparents)

# Associations dont aucun coté a cardinalité 1 - suite

- Choix de la clef de R

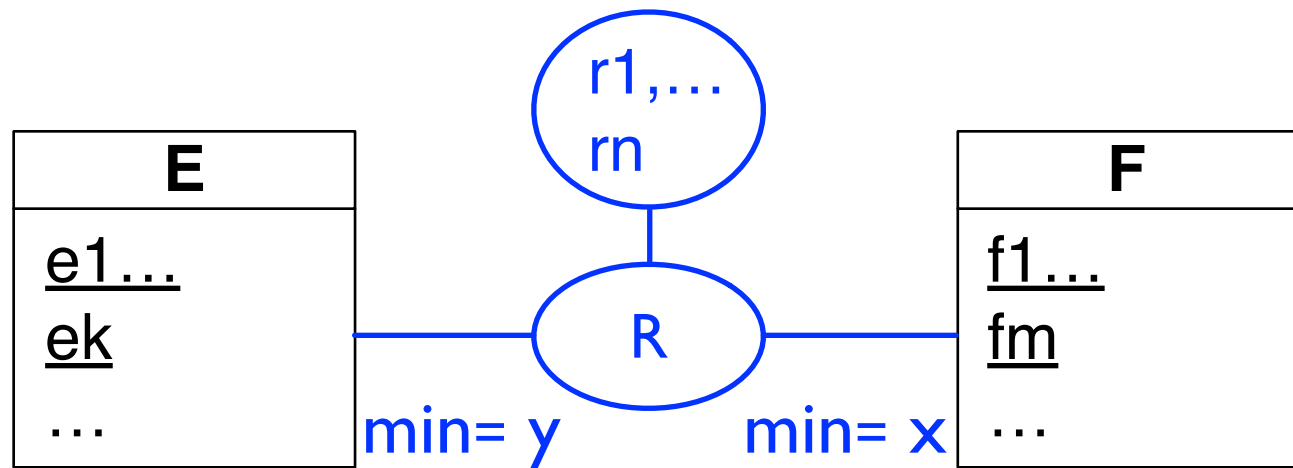
(x, y = 0 / 1)



# Associations dont aucun coté a cardinalité 1 - suite

- Contraintes additionnelles

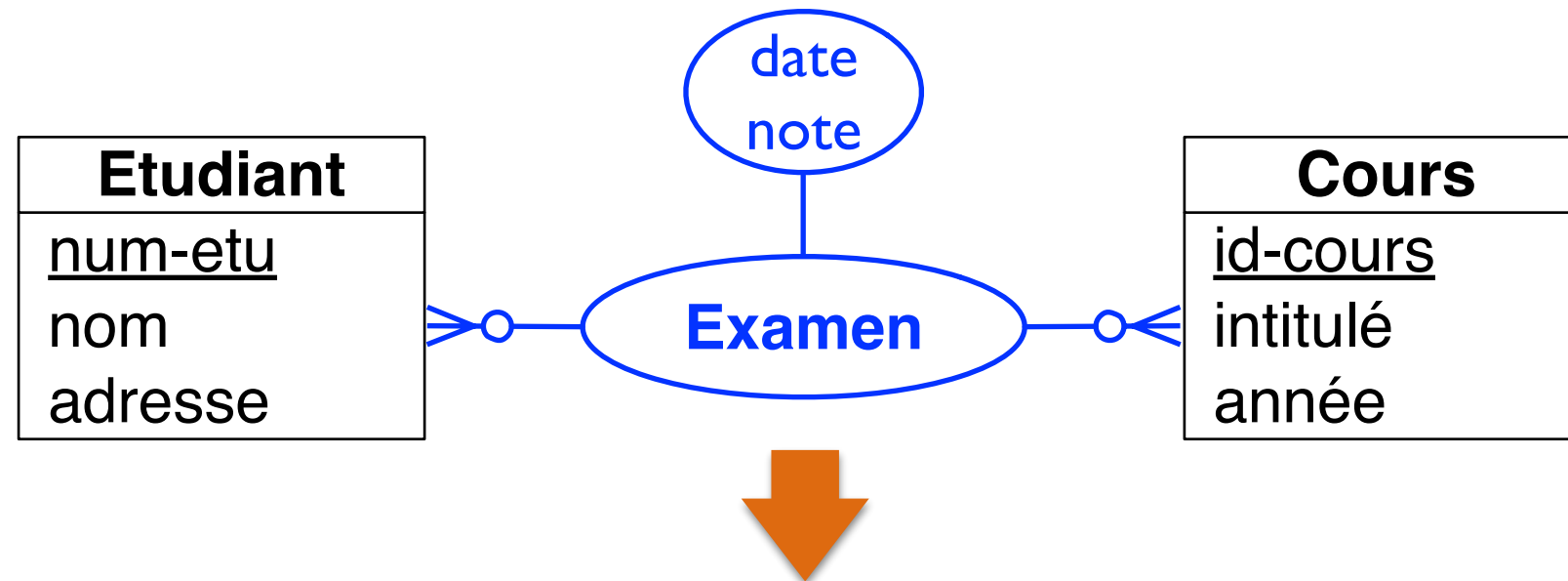
- ▶ D'autres contraintes d'inclusion sur le schéma relationnel dérivent des cardinalité minimales



- Si  $x$  est 1:  
 $\mathbf{E} [e1, \dots, ek] \subseteq \mathbf{R} [e1, \dots, ek]$
- Si  $y$  est 1:  
 $\mathbf{F} [f1, \dots, fm] \subseteq \mathbf{R} [f1, \dots, fm]$

# Associations dont aucun coté a cardinalité 11 - exemples

- Exemple : association plusieurs à plusieurs



**Examen** (num-etu, id-cours, date, note)

+ contraintes :

**Examen** [num-etu]  $\subseteq$  **Etudiant** [num-etu],

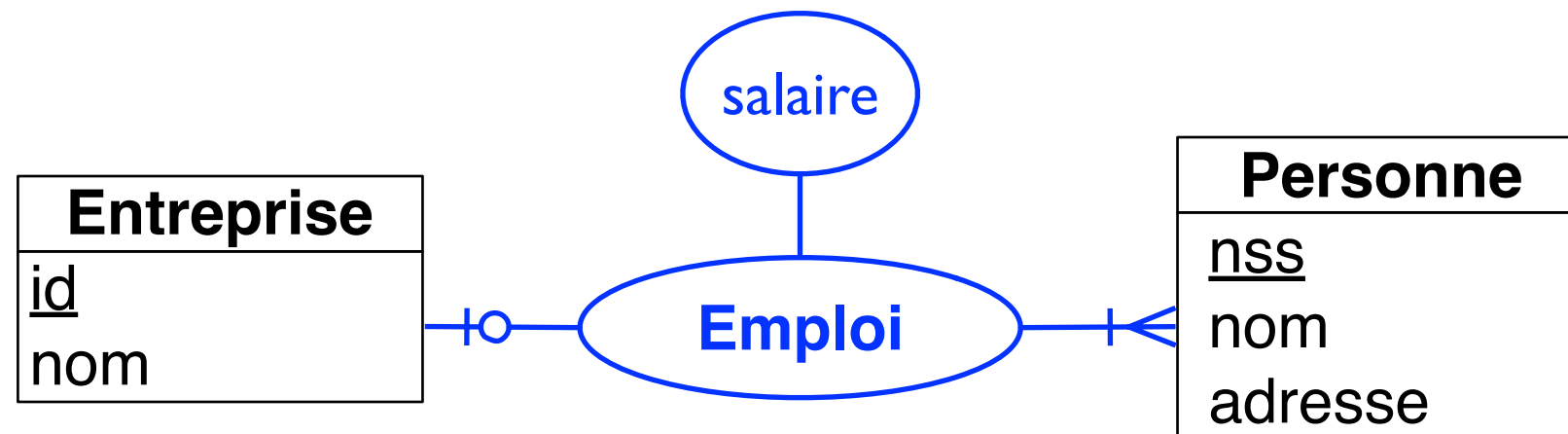
**Examen**[id-cours]  $\subseteq$  **Cours**[id-cours]

- Remarque sur la clef de la relation produite.** La relation Examen représente les instances de l'association Examen, i.e. des couples <étudiant, cours>
  - un tel couple est identifié par <clef de l'étudiant, clef du cours>
  - cet identifiant est minimal pour Examen parce que un étudiant peut être associé à plusieurs cours et un cours à plusieurs étudiants



# Associations dont aucun coté a cardinalité 11 - exemples

- Exemple : Association un à plusieurs



**Emploi** (nss, id-entreprise, salaire)

+ contraintes :

$\text{Emploi}[\text{nss}] \subseteq \text{Personne}[\text{nss}]$

$\text{Emploi}[\text{id-entreprise}] \subseteq \text{Entreprise}[\text{id}]$

$\text{Entreprise}[\text{id}] \subseteq \text{Emploi}[\text{id-entreprise}]$

# Associations dont aucun coté a cardinalité 11 - exemples

- Exemple : Association un à un



**Occupation** (num-chambre, id-client)

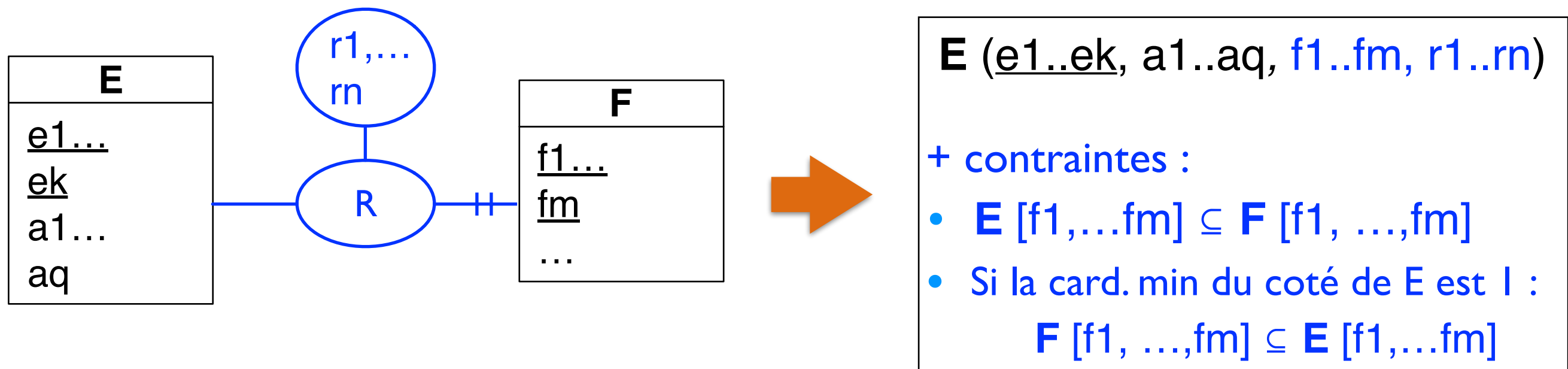
+ contraintes :

$\text{Occupation}[\text{num-chambre}] \subseteq \text{Chambre}[\text{num}]$

$\text{Occupation}[\text{id-client}] \subseteq \text{Client}[\text{id}]$

# Associations dont un coté a cardinalité 1 1

- Premier cas : R n'est pas identifiante pour l'entité avec participation 1 1 (E)

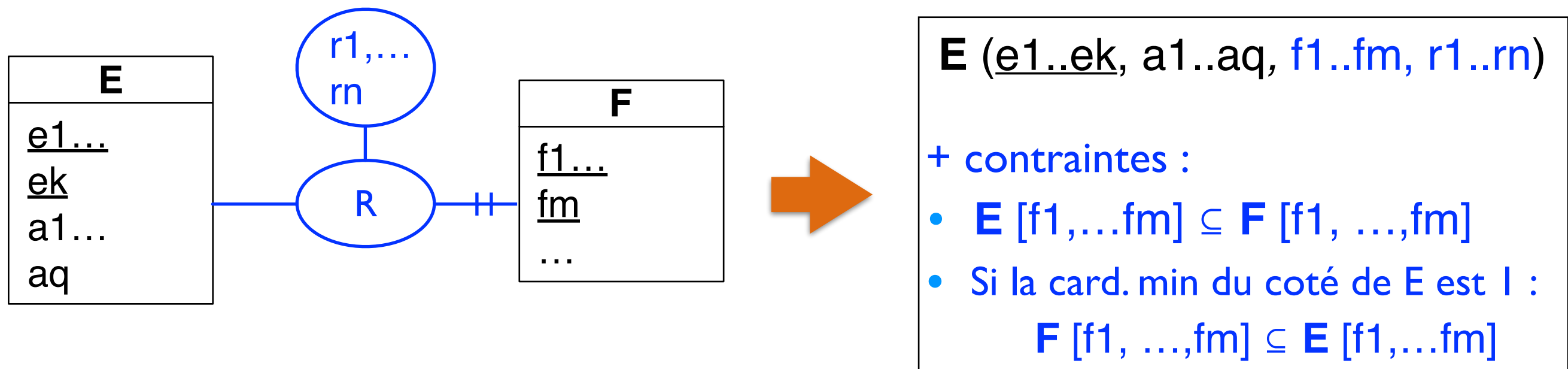


## Remarques :

- on modifie la traduction de E en ajoutant à ses attributs l'identifiant de F, et les attributs de R
- cela est correct puisque chaque instance de E est associée par R à exactement une instance de F.
- on évite cette traduction dans le cas de cardinalité 0 1 (au lieu de 1 1), puisque cela pourrait introduire des NULL dans les valeurs de  $f1 \dots fm, r1, \dots, rn$

# Associations dont un coté a cardinalité 11

- Premier cas : R n'est pas identifiante pour l'entité avec participation 11 (E)

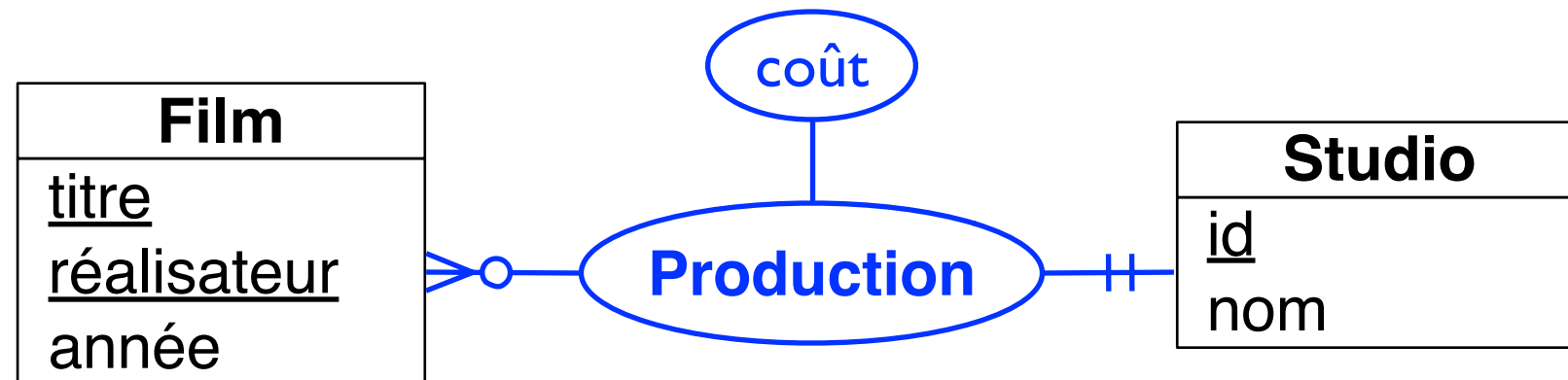


## Encore des remarques :

- ▶ la traduction de R est la partie en bleu (E et F doivent être traduites indépendamment)
- ▶ l'effet de la traduction de plusieurs associations de ce type peut se cumuler en E
- ▶ si la cardinalité est 11 des deux cotés, on modifie E ou F, pas les deux

# Associations dont un coté a cardinalité 1 1

- Exemple



**Film** (titre, réalisateur, année, id-studio, coût-production)

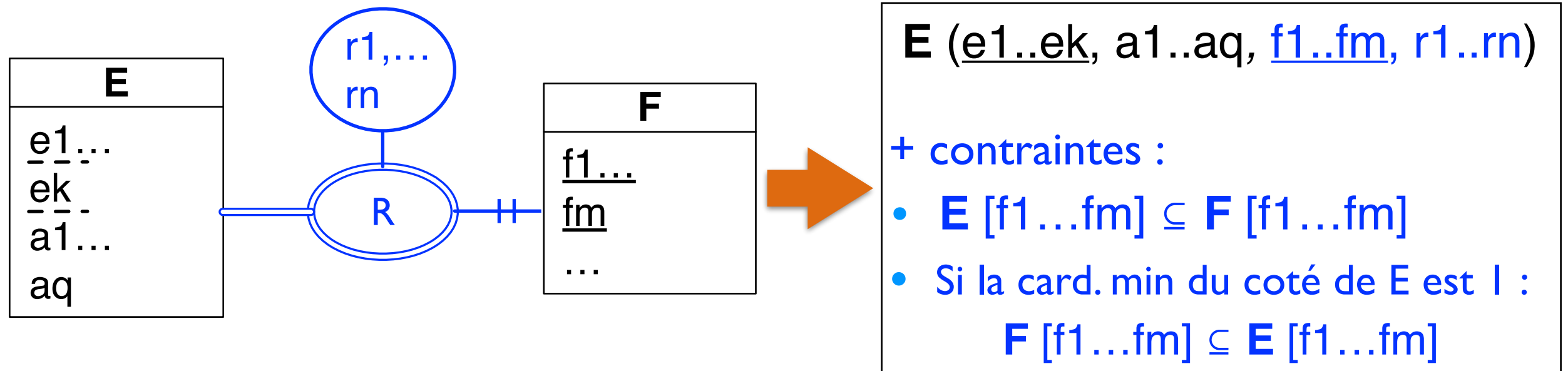
+ contrainte de clef étrangère :

**Film** [id-studio]  $\subseteq$  **Studio** [id]

- Remarque.** On peut renommer les concepts (entités, associations, attributs) en passant au schéma relationnel

## Associations dont un coté a cardinalité 1 1

- Deuxième cas : R est identifiante pour l'entité avec participation 1 1 (E)

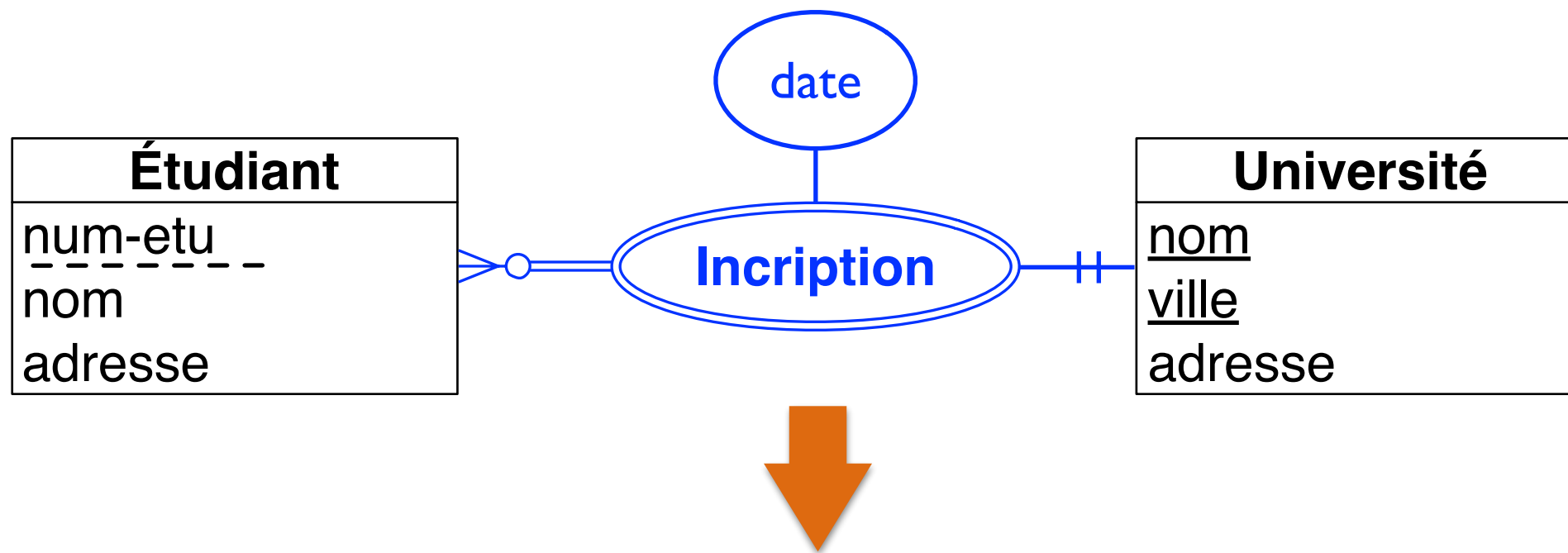


### Remarques :

- ▶ même traduction que dans le premier cas, mais la clef de  $E$  est étendue avec celle de  $F$
- ▶ l'effet de la traduction de plusieurs associations de ce type (et du premier type) peut se cumuler en  $E$
- ▶ si la cardinalité est 1 1 des deux cotés,  $R$  doit être traduite dans  $E$  (dont  $R$  est identifiante), pas dans  $F$

# Associations dont un coté a cardinalité 1 1

- Exemple

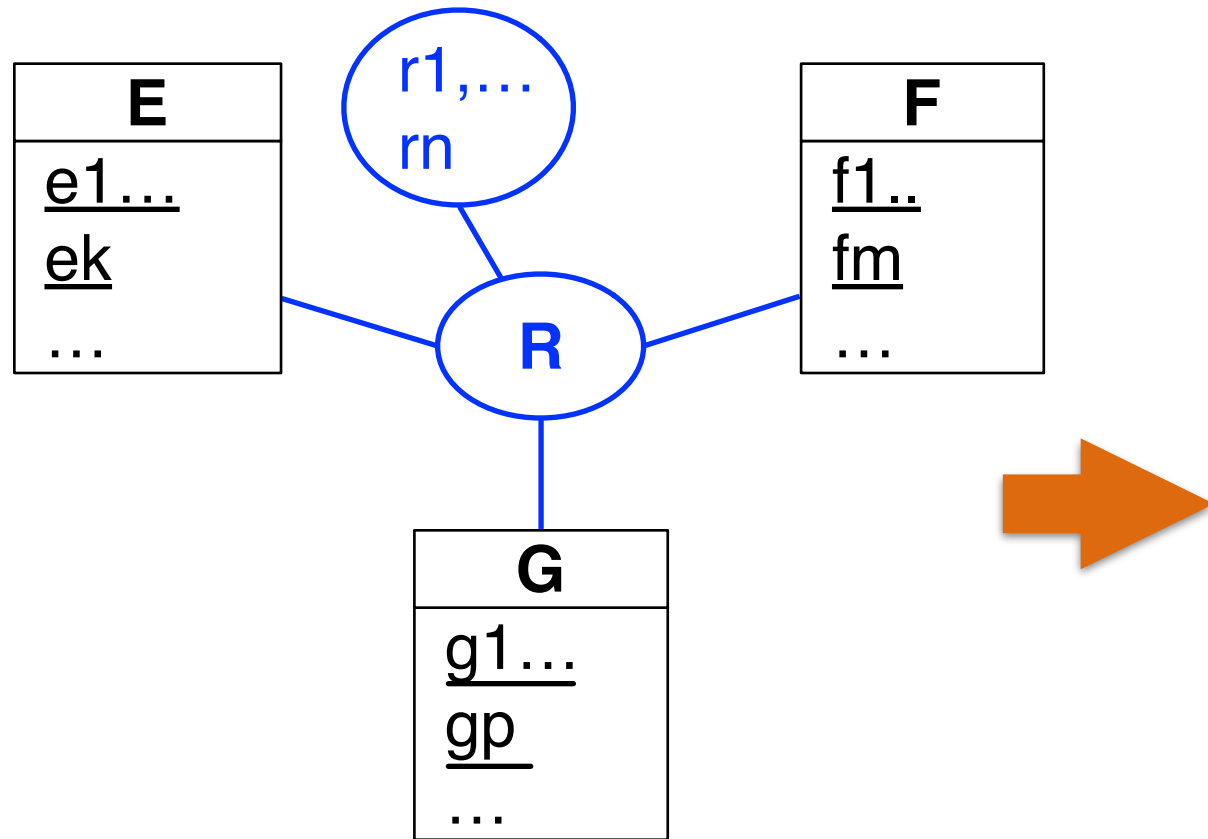


Étudiant (num-etu, nom-univ, ville-univ, nom, adresse, date-inscription)

+ Contrainte de clef étrangère :

Étudiant [nom-univ, ville-univ]  $\subseteq$  Université[nom, ville]

## Traduction des associations n-aires



$R(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p, r_1, \dots, r_n)$

+ contraintes :

- $R[e_1, \dots, e_k] \subseteq E[e_1, \dots, e_k]$
- $R[f_1, \dots, f_m] \subseteq F[f_1, \dots, f_m]$
- $R[g_1, \dots, g_p] \subseteq G[g_1, \dots, g_p]$

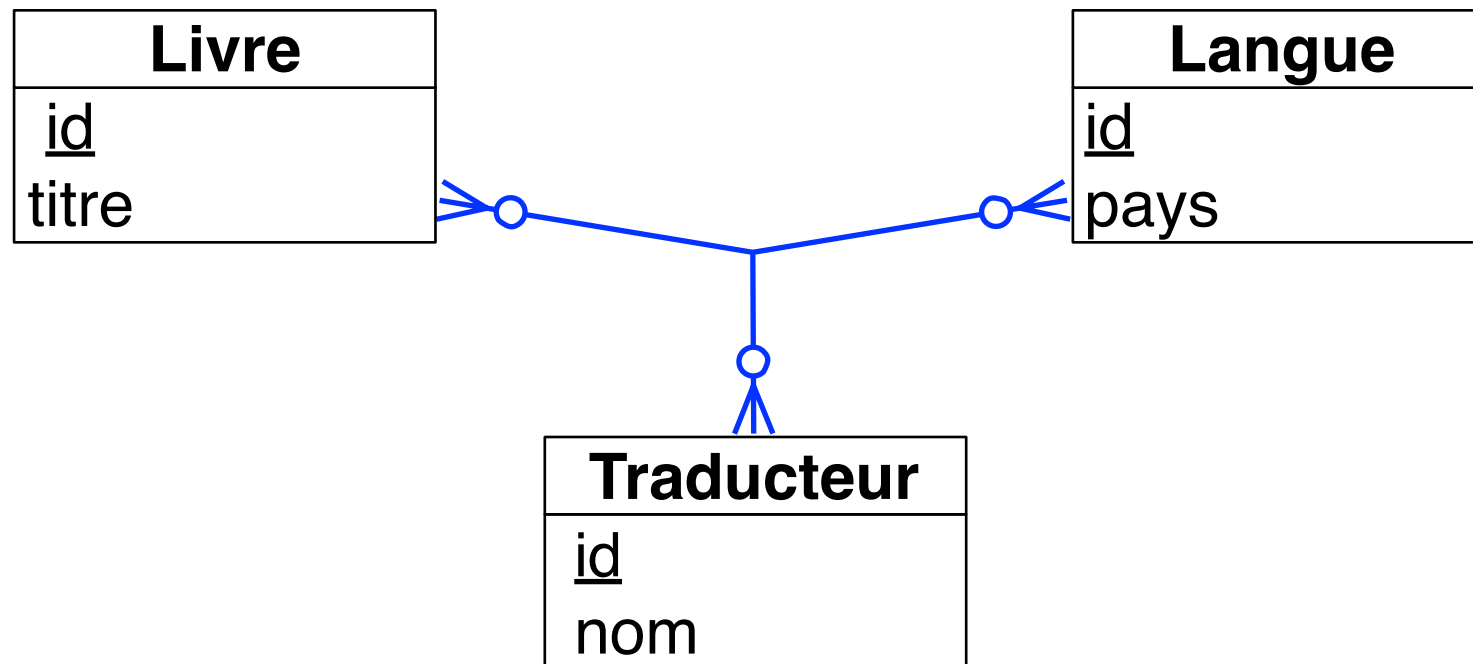
- **Clef de R :**

- ▶  $(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p)$  est une superclef de  $R$ , mais pas nécessairement minimale
- ▶ superclef minimale : dépend des contraintes de cardinalité mais aussi de possibles contraintes externes; à voir le cas par cas.
- ▶ Si la cardinalité maximale du côté de  $G$  est 1 alors  $(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p)$  n'est pas minimale puisque  $(e_1, \dots, e_k, f_1, \dots, f_m)$  est aussi une superclef
  - mais  $(e_1, \dots, e_k, f_1, \dots, f_m)$  pourrait aussi ne pas être minimale



# Traduction des associations n-aires

- Exemple I



Traduction (id-livre, id-langue, id-traducteur)

+ contrantes :

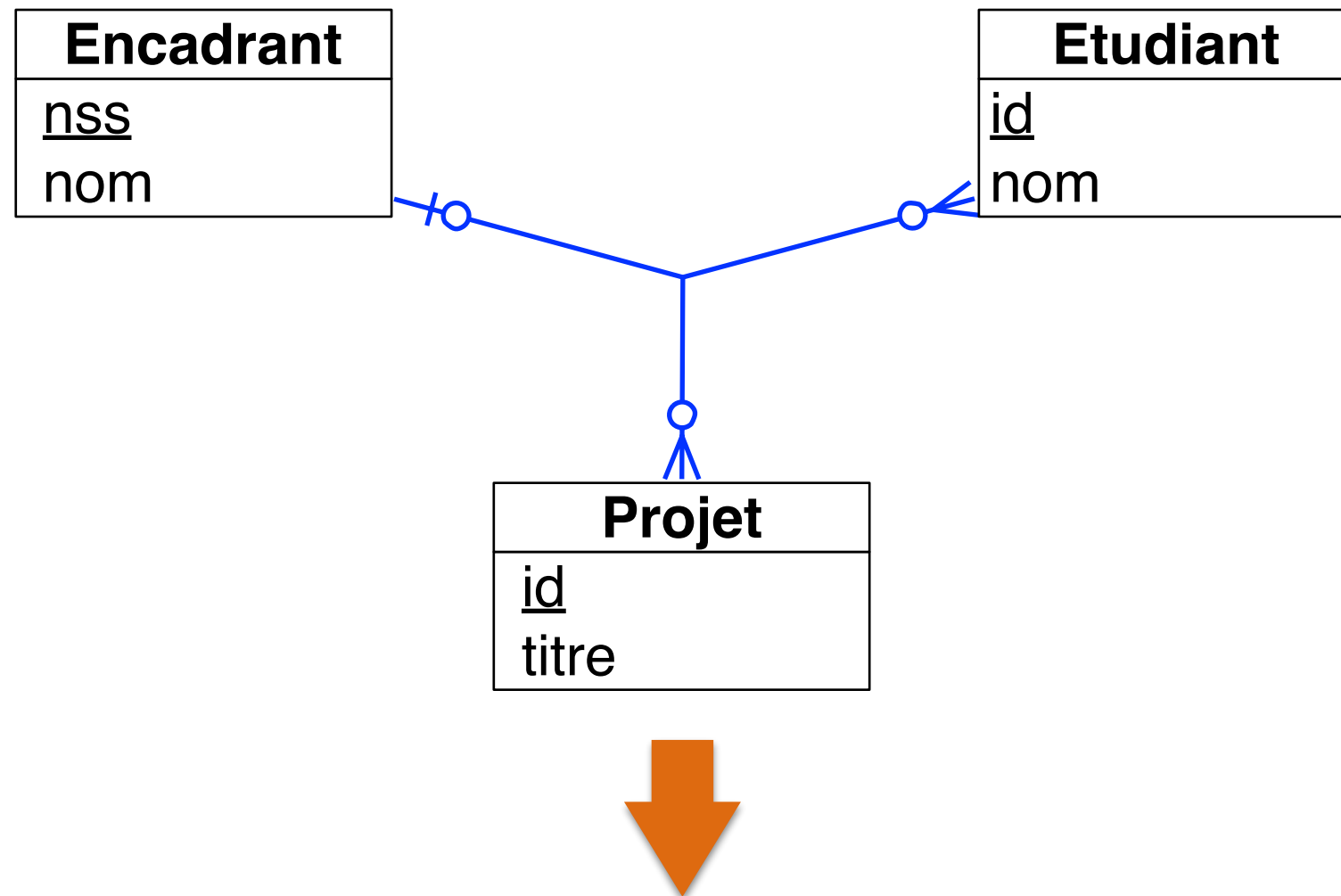
Traduction[id-livre]  $\subseteq$  Livre[id]

Traduction[id-langue]  $\subseteq$  Langue[id]

Traduction[id-traducteur]  $\subseteq$  Traducteur[id]

# Traduction des associations n-aires

- Exemple 2



**Encadrement** (id-et, id-projet, nss-enc)

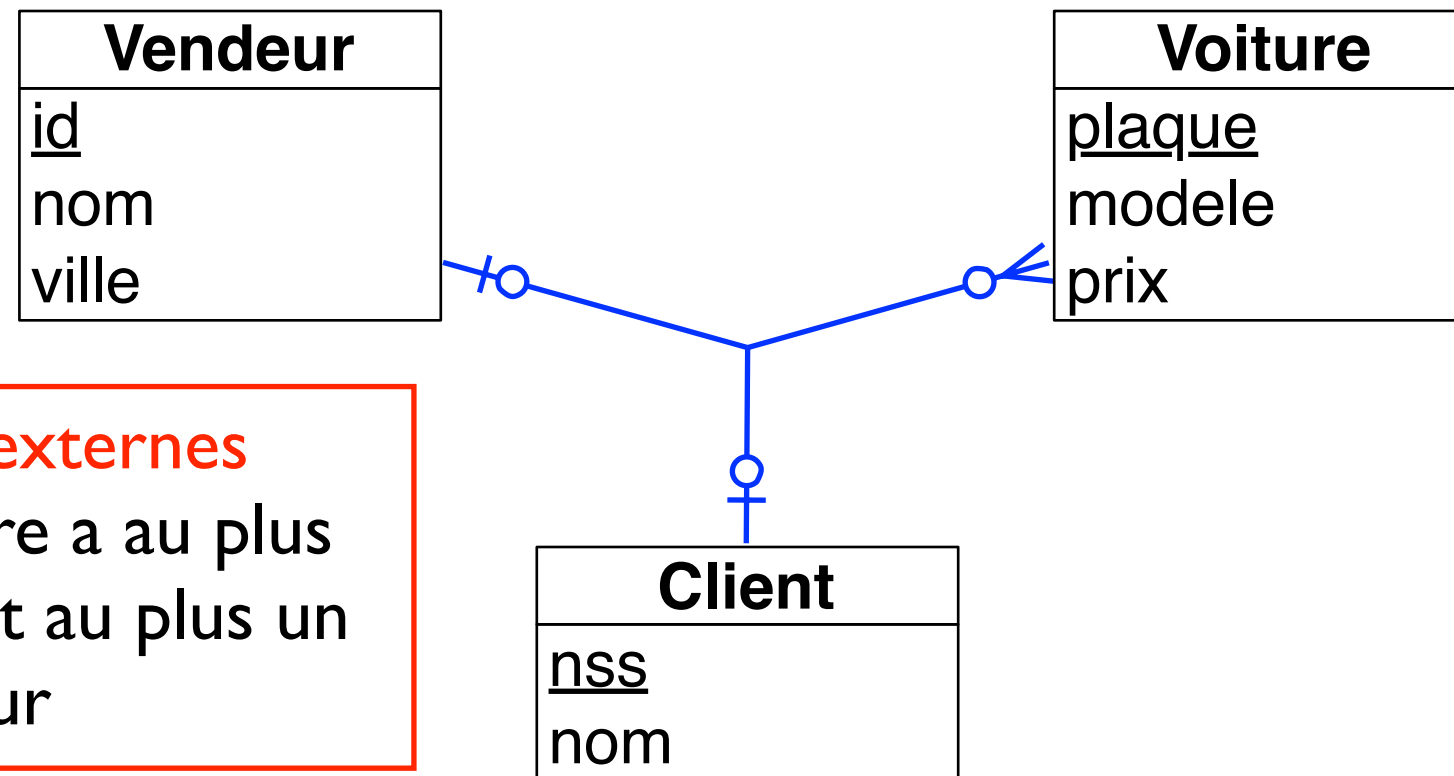
Encadrement [id-et]  $\subseteq$  Etudiant[id]

Encadrement [id-projet]  $\subseteq$  Projet[id]

Encadrement [nss-enc]  $\subseteq$  Encadrant [id]

# Traduction des associations n-aires

- Exemple 3



## Contraintes externes

chaque voiture a au plus un vendeur et au plus un client acheteur



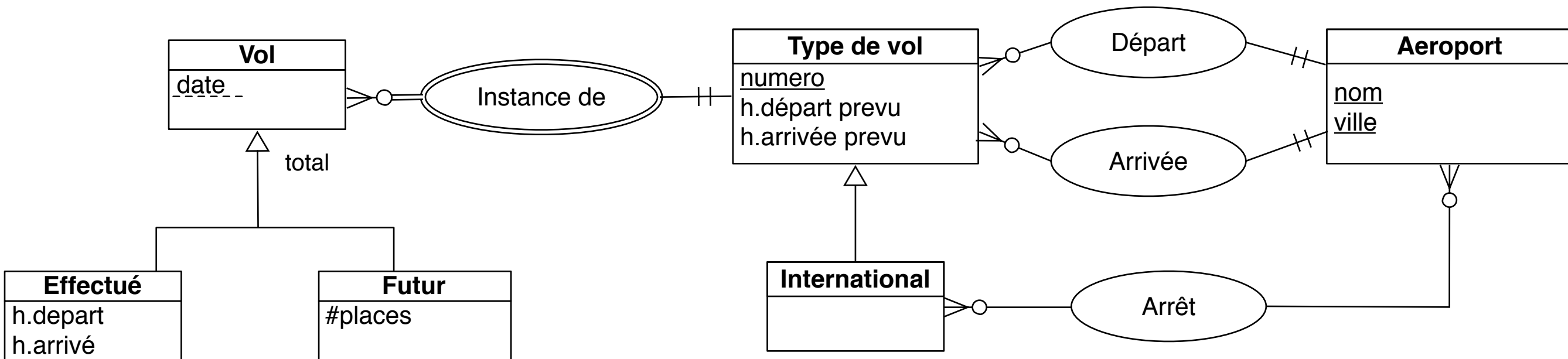
**Achat** (plaque-voiture, id-vendeur, nss-client)

**Achat** [plaque-voiture]  $\subseteq$  **Voiture** [plaque]

**Achat** [nss-client]  $\subseteq$  **Client** [nss]

**Achat** [id-vendeur]  $\subseteq$  **Vendeur** [id]

# Un exemple complet de traduction

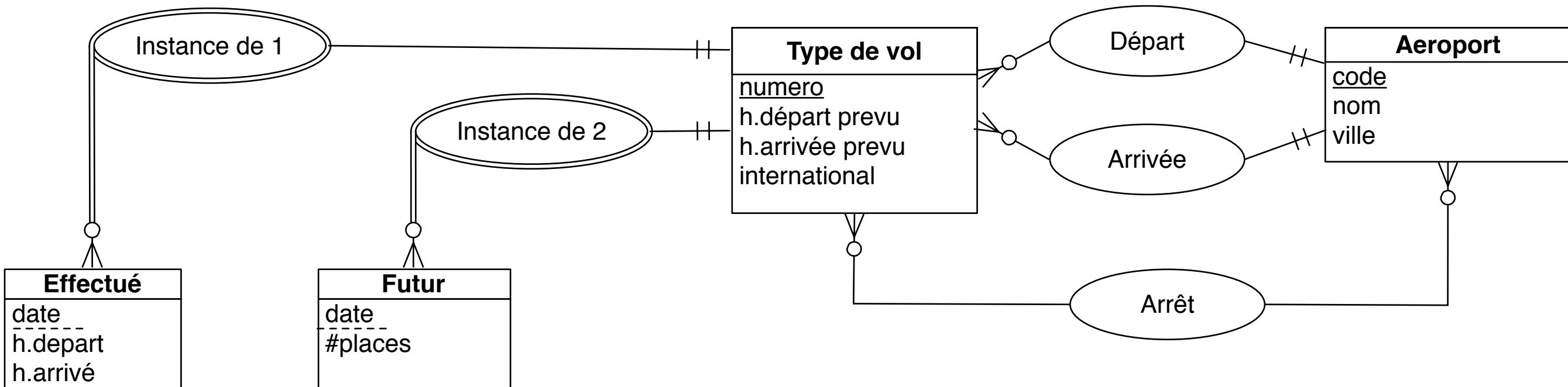


+ contrainte externe : la date de chaque vol effectué est  $\leq$  à la date de chaque vol futur

- **Première étape : restructuration**

- ▶ **Vol** : élimination de l'entité mère
- ▶ **Type de Vol** : élimination de l'entité fille avec ajout d'attribut
- ▶ introduction d'un nouvel identifiant pour **Aéroport**

# Un exemple complet de traduction



## + contraintes externes :

- ▶ la date de chaque vol effectué est  $\leq$  à la date de chaque vol futur
- ▶ (nom, ville) est une clef candidate pour Aéroport

## Diagramme restructuré

# Un exemple complet de traduction

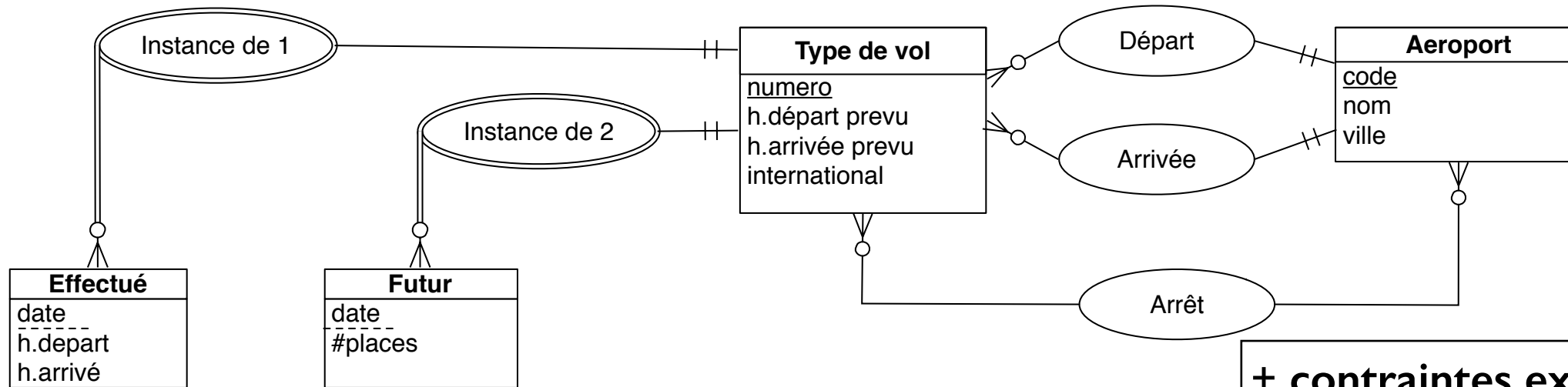


Diagramme E/R restructuré

+ **contraintes externes** :

- \* la date de chaque vol effectué est  $\leq$  à la date de chaque vol futur
- \* (nom, ville) clef candidate pour Aéroport

## Schéma relationnel

Aéroport (code, nom, ville)

Arret (num\_vol, code\_aero)

Vol\_Effectue (num\_vol, date, h\_dep, h\_arr)

Vol\_Futur (num\_vol, date, #places)

Type\_Vol (numero, aero\_dep, aero\_arr, hp\_dep, hp\_arr, international)

## Contraintes de clefs étrangère :

- R [num\_vol]  $\subseteq$  Type\_Vol [numero]  
(pour R = Vol\_Effectué, Vol\_Futur, Arret)
- Arret [code\_aero]  $\subseteq$  Aéroport [code]
- Type\_Vol [aero\_dep]  $\subseteq$  Aéroport [code]
- Type\_Vol [aero\_arr]  $\subseteq$  Aéroport [code]

**Contraintes d'unicité** : Aéroport (nom, ville)

**Autres contraintes** : Check ou assertions

# Outils gratuits pour dessiner des diagrammes E/R

- Dia
- Calligra Flow
- Beaucoup d'autres outils permettent également la transformation automatique ER → Relationnel (MySQL Workbench, ER2SQL, ...)
  - ▶ Attention : ils ont leur propre syntaxe et sémantique pour ER
  - ▶ on obtient souvent un meilleur modèle quand on traduit “à la main”