



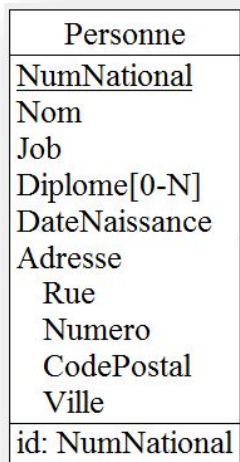
Chapitre 2:

Le modèle Entité–Association

4. Les relations: l'Association

L'Association: définition

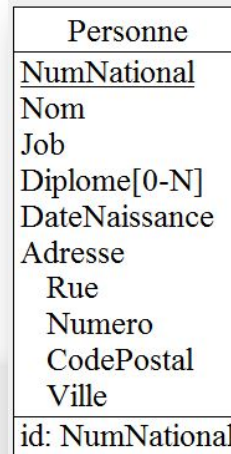
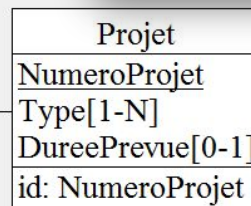
- Une **Association** capture une relation *statique* qu'ont des entités d'une classe avec des entités d'une autre classe
 - Se lie dans les deux sens!



0-N

Participe

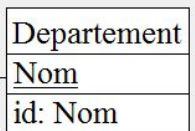
1-N



1-1

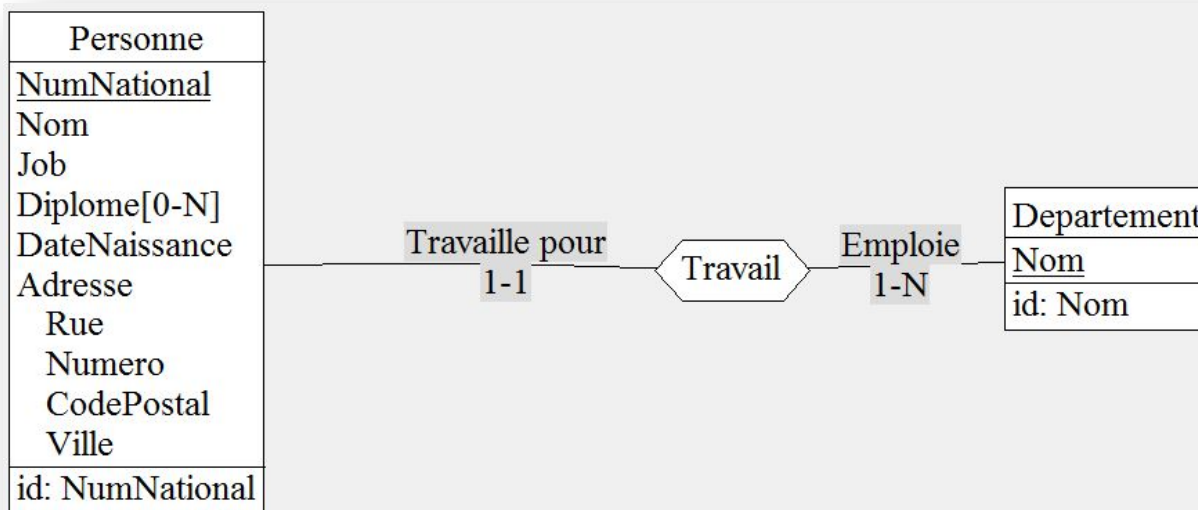
Travaille

1-N



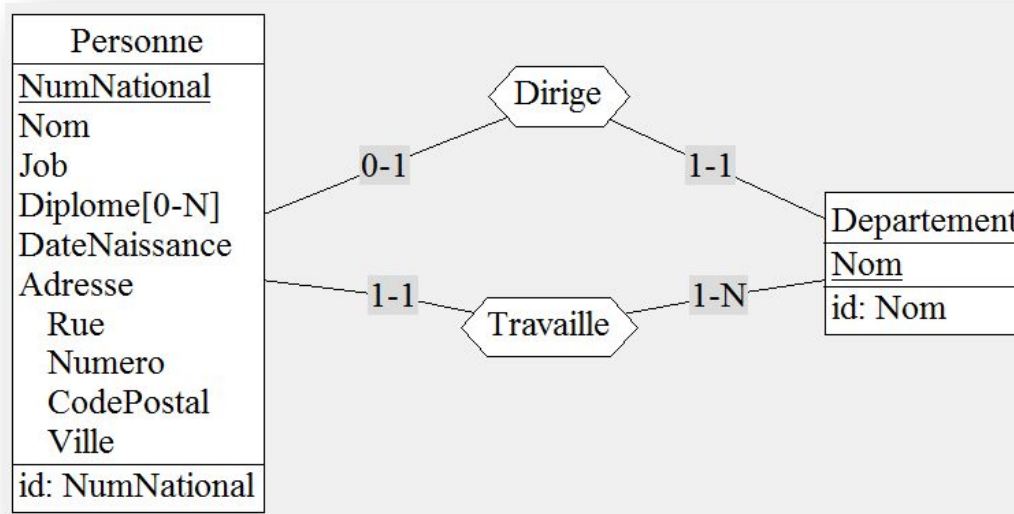
Les rôle d'une Association

- ▶ Un **rôle** permet de donner la fonction de chaque Entité dans l'Association
- ▶ Optionnel (sauf pour les Associations récursives)
- ▶ Utiliser un terme neutre comme dénomination de l'Association lorsque des rôles sont indiqués



Les cardinalités d'une association

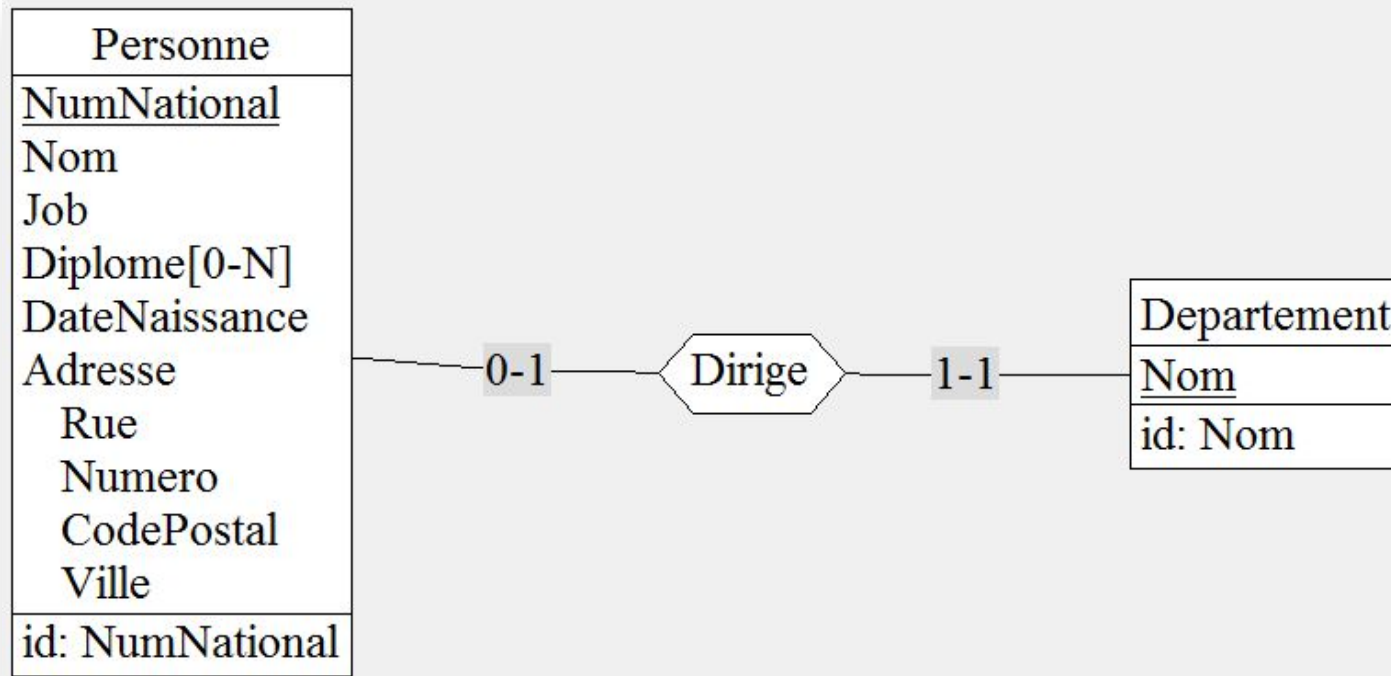
- ▶ Chaque association binaire a 2 cardinalités, une pour chacun des rôles
- ▶ Une cardinalité a une borne supérieure et une borne inférieure
- ▶ Les cardinalités de l'Association '*Travaille*' se lisent:
 - De 'Personne' vers 'Departement': "Une entité de la classe Personne a une et une seule relation Travaille avec les entités de la classe d'entités Departement"
 - De 'Departement' vers 'Personne': "Une entité de la classe Departement a entre 1 et N relation avec les entités de la classe Personne"



Caractéristiques des cardinalités des Associations

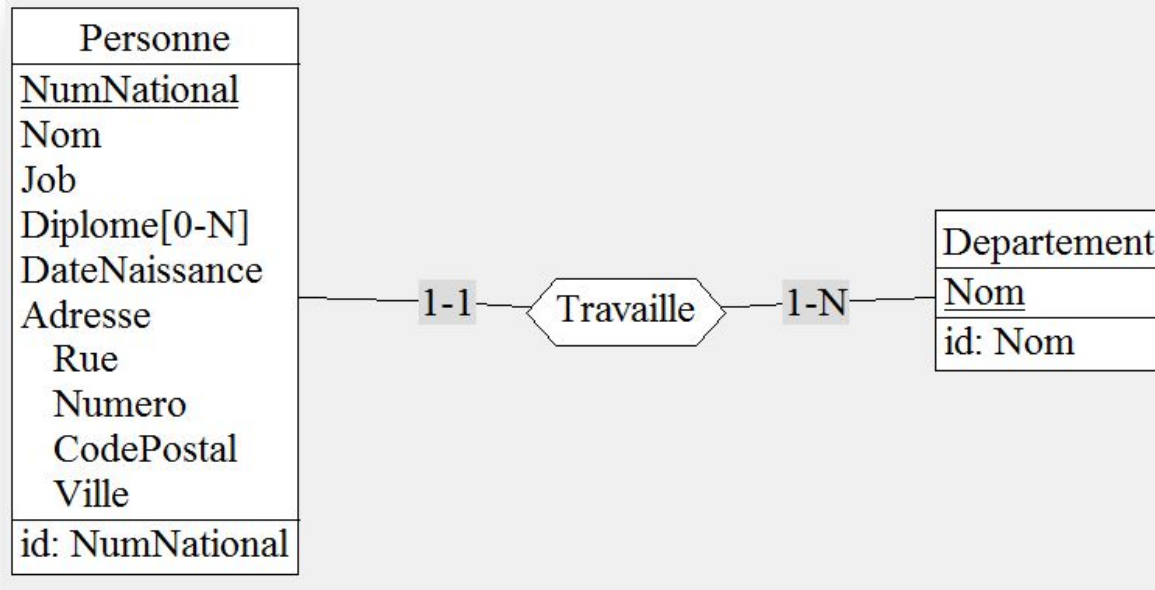
- ▶ Soit la cardinalité d'une Association (x,y)
- ▶ Optionnel VS. Obligatoire
 - **Optionnel:** $x = 0$
Une entité *peut* avoir une relation avec les entités de la classe mise en relation
 - **Obligatoire:** $x \geq 1$
Chaque entité *doit* avoir une relation avec les entités de la classe mise en relation
- ▶ Trois types d'associations:
 - **One-to-One:** les deux bornes supérieures des deux cardinalités d'une association sont égales à 1
 - **One-to-Many:** une des deux bornes supérieures des deux cardinalités d'une association vaut 1, l'autre est strictement supérieur à 1
 - **Many-to-Many:** les deux bornes supérieures des deux cardinalités d'une association sont strictement supérieures à 1

Association One-to-One



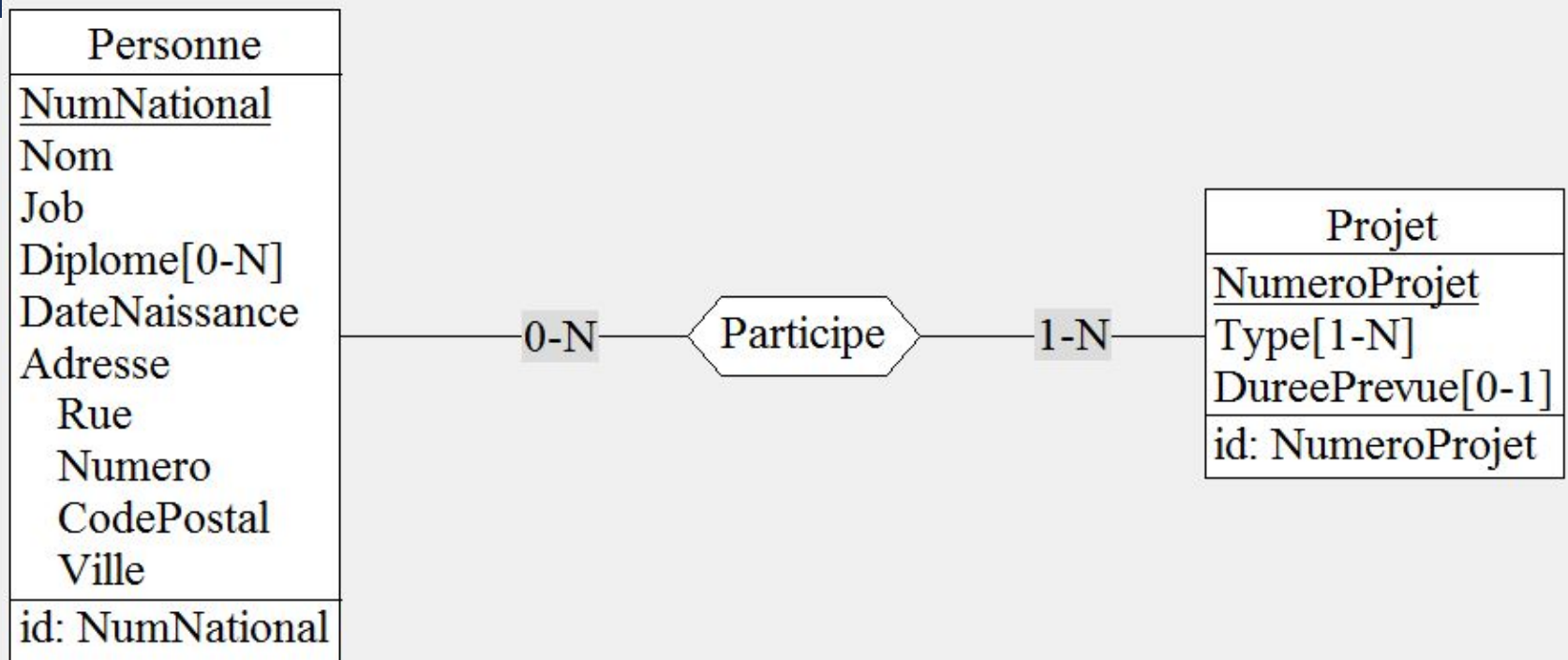
- Une 'Personne' peut diriger un seul 'Departement'
- Un 'Departement' est dirigé par une et une seule 'Personne'

Association One-to-Many



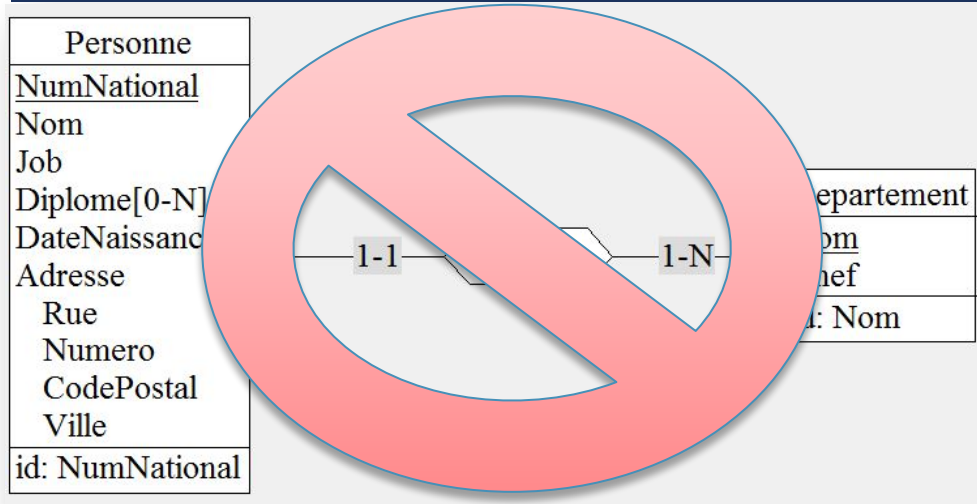
- ▶ Chaque 'Personne' travaille pour un seul 'Departement'
- ▶ Chaque 'Departement' emploie de une à n 'Personnes'
- ▶ Une association One-to-Many est similaire à une association Many-to-One (lecture des associations dans les deux sens!)

Association Many-to-Many

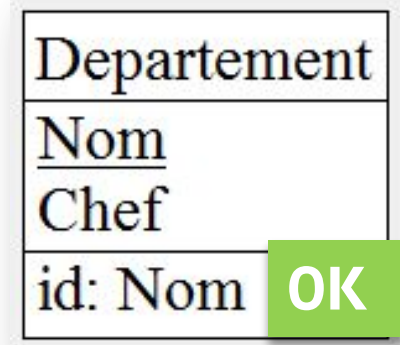


- ▶ Chaque entité 'Personne' peut travailler pour plusieurs 'projets'
- ▶ Chaque entité 'Projet' utilise au moins une 'personne', jusqu'à n 'personne'

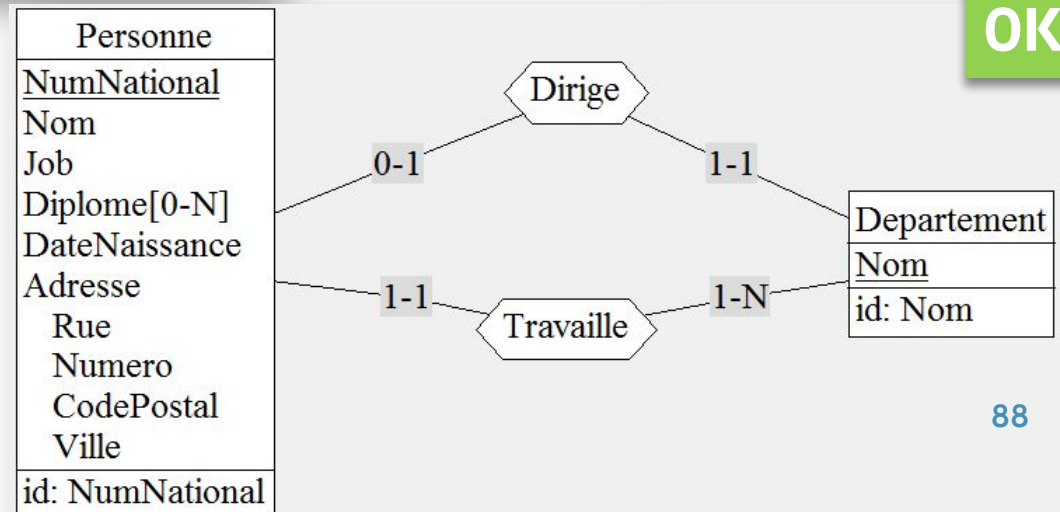
Utiliser adéquatement les Associations



- Si la classe 'Personne' n'est pas dans le schéma



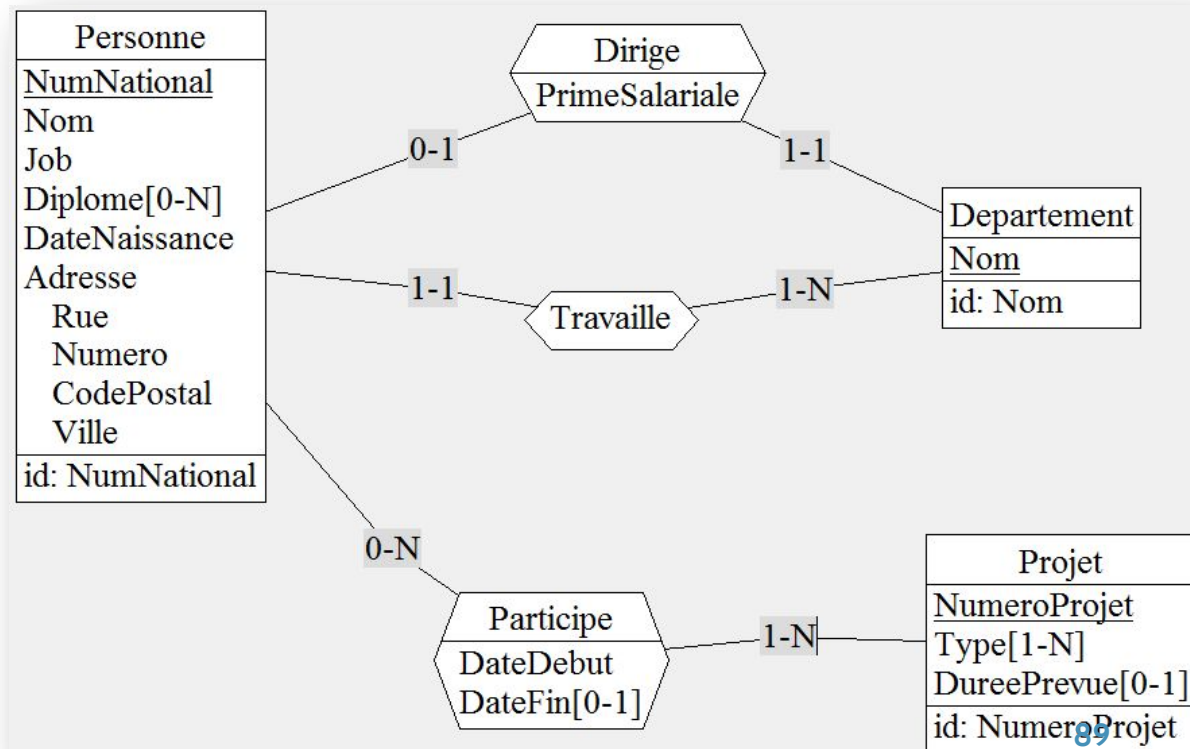
- Utiliser des relations pour capturer les liens du monde réel lorsque une classe de votre schéma capture la même donnée!



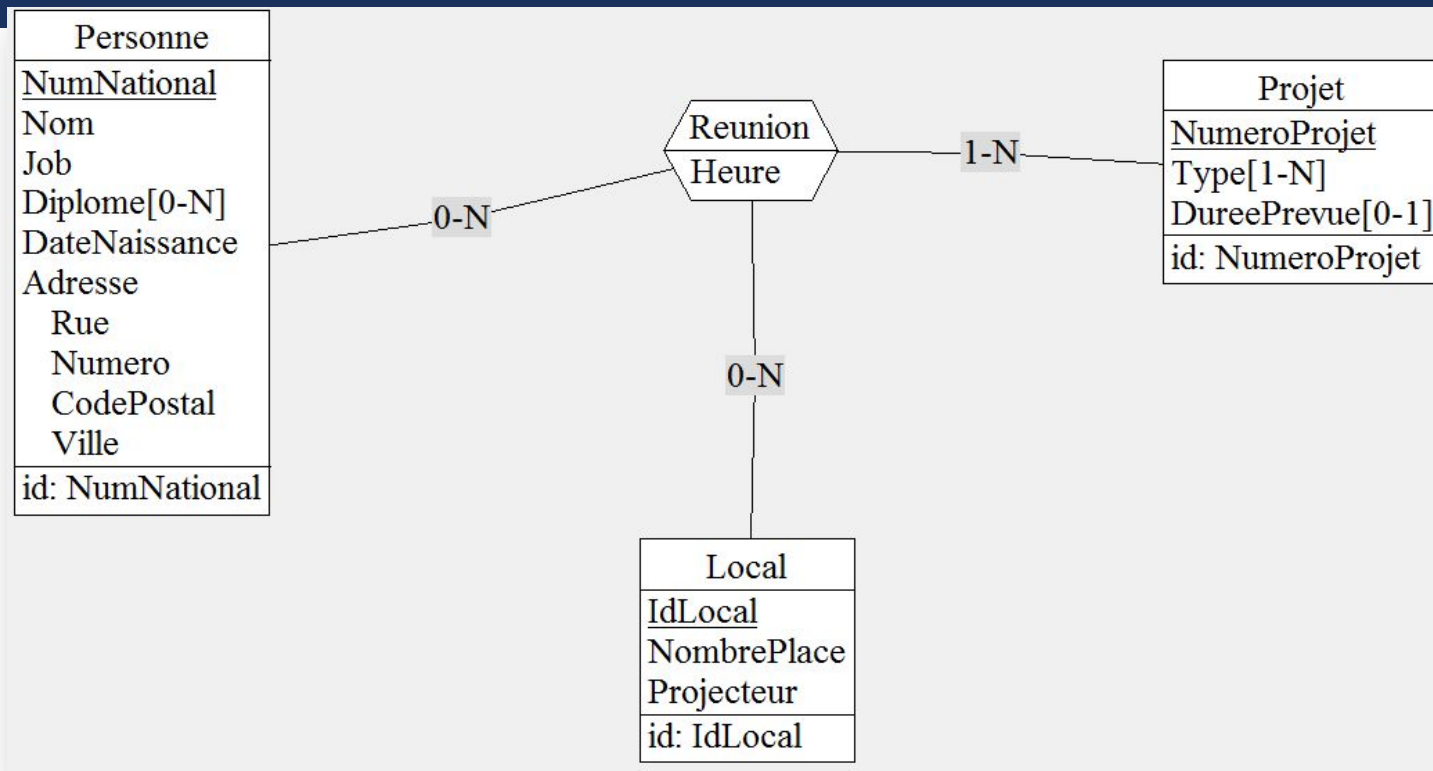
Les Attributs des Associations

- Un Attribut est ajouté à une Association ssi cet Attribut dépend des *deux* Classes d'Entités que l'Association lie

- Tout comme les Attributs de Classes, les Attributs d'Associations ont un domaine et une cardinalité, et peuvent être composite ou identifiant

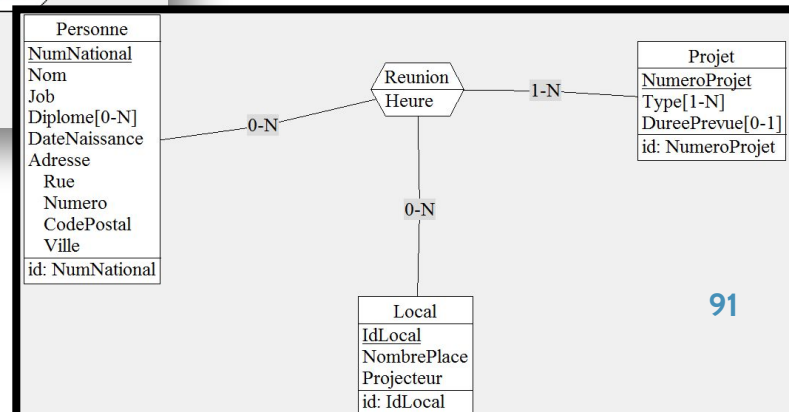
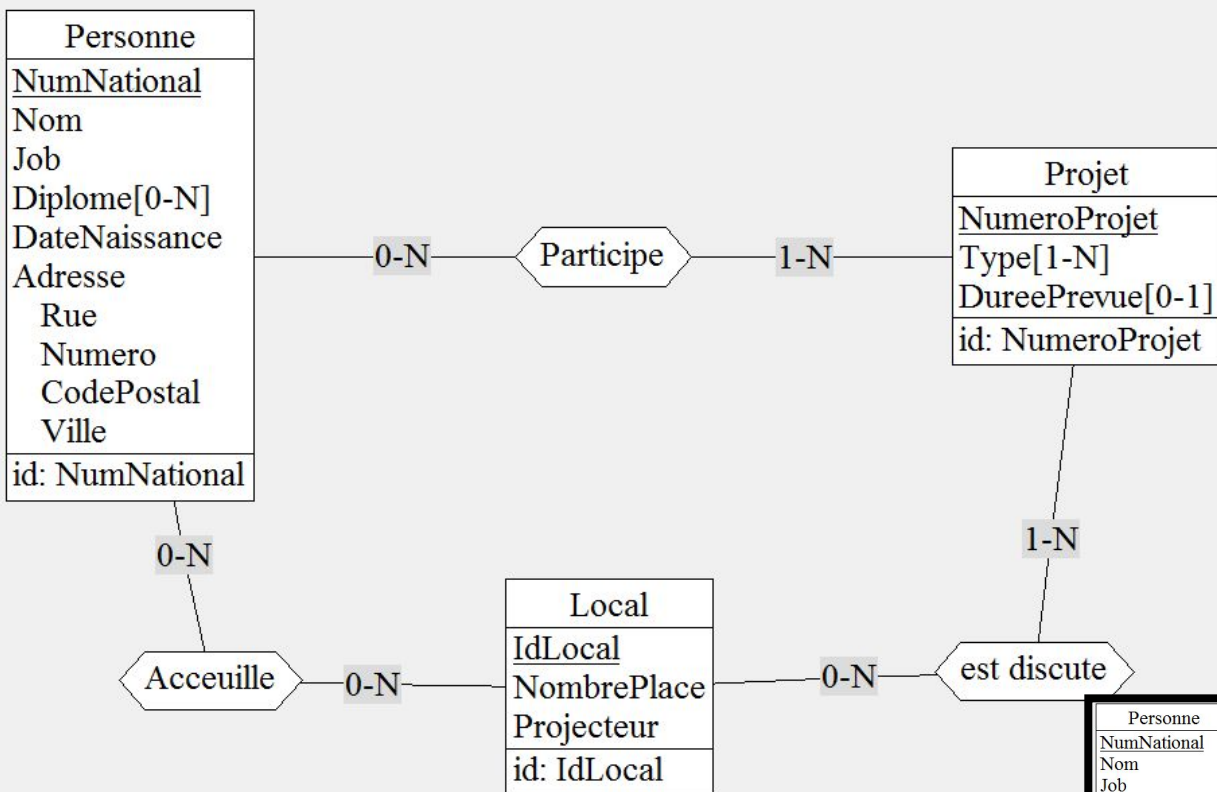


L'Association Ternaire



- **L'association ternaire** capture un lien entre trois types d'objets identifiés dans le domaine d'application

Projection d'une Association ternaire



Association ternaire VS. Trois Associations binaires

► Un exemple avec des instances *Reunion*

Personne	Projet	Local
Georges	Modélisation n°1	Salle 17
Lucas	Modélisation n°1	Salle 18
Georges	Modélisation n°2	Salle 18

Participe

Personne	Projet
Georges	Modélisation n°1
Lucas	Modélisation n°1
Georges	Modélisation n°2

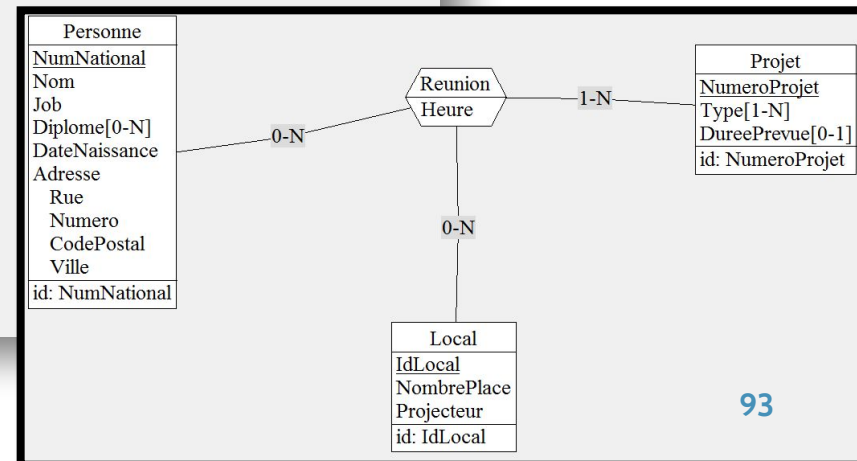
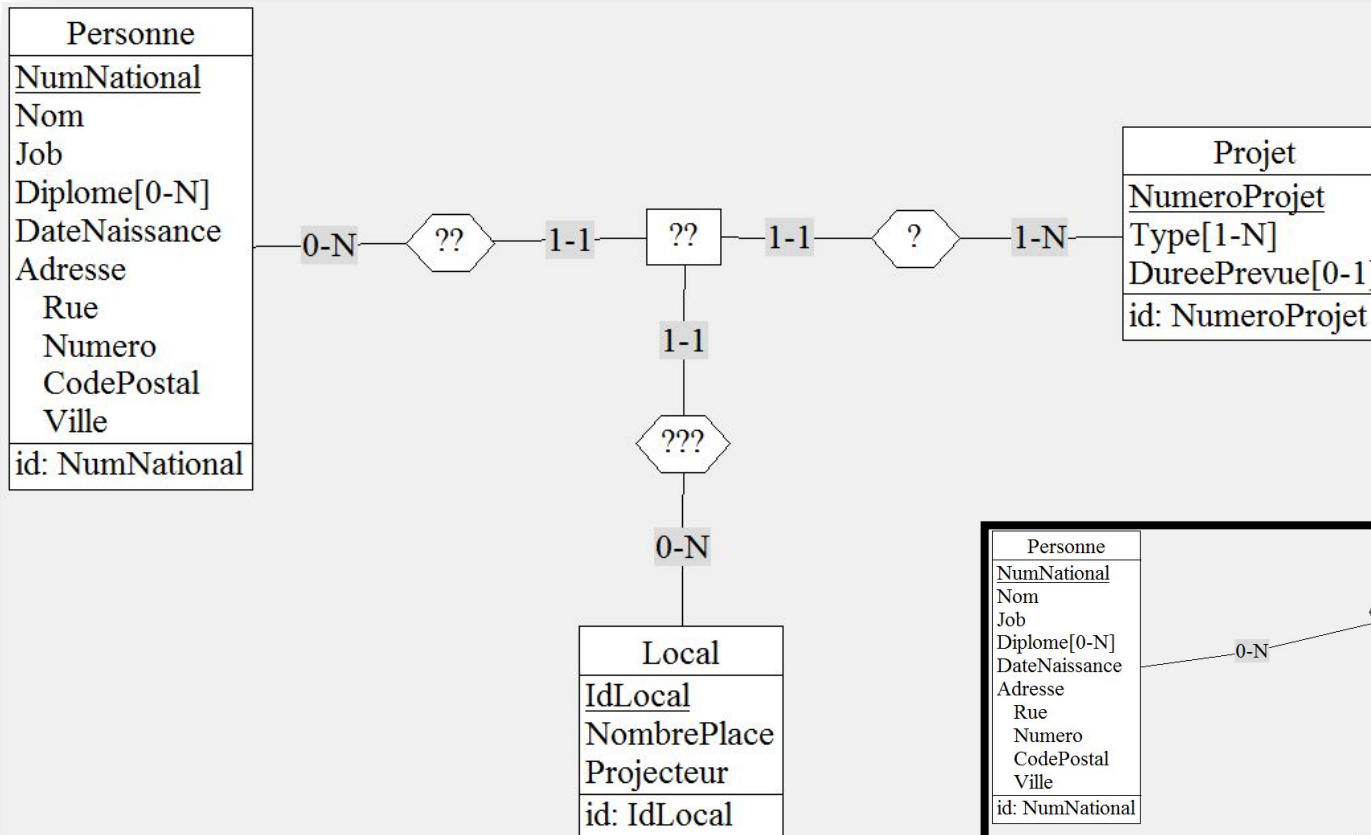
Accueille

Personne	Local
Georges	Salle 17
Lucas	Salle 18
Georges	Salle 18

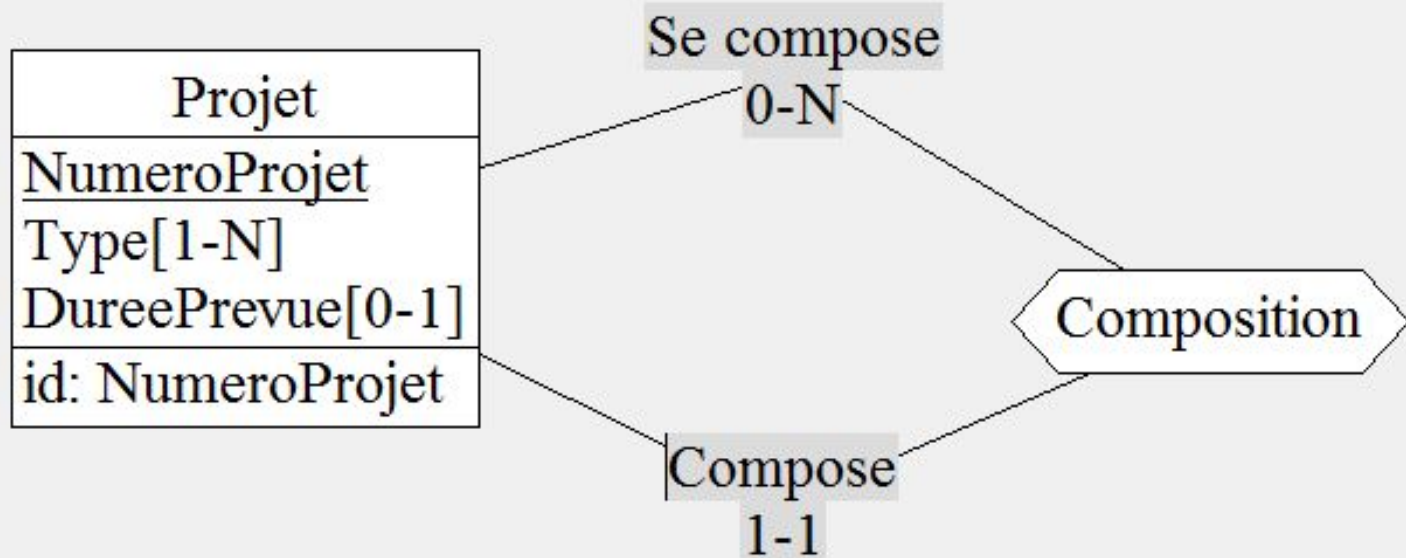
Est discute

Local	Projet
Salle 17	Modélisation n°1
Salle 18	Modélisation n°1
Salle 18	Modélisation n°2

Modélisation correcte d'une relation ternaire avec des associations binaires



L'Association Récursive (ou Association cyclique)



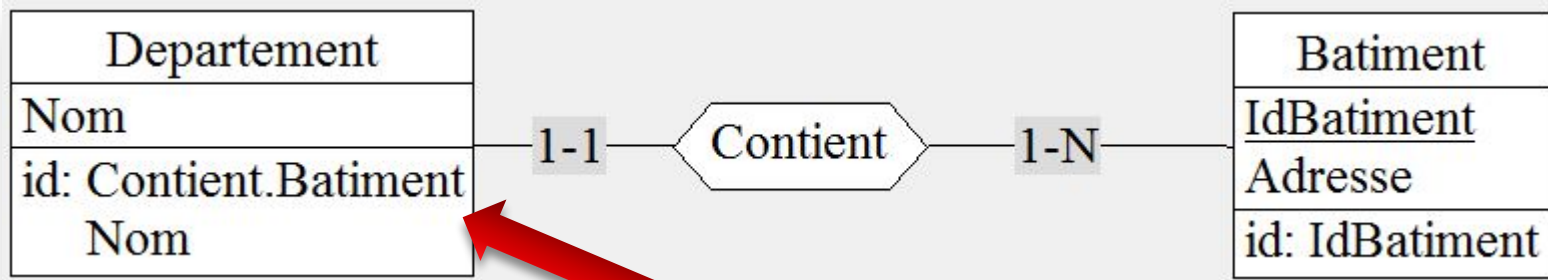
- ▶ **L'Association Récursive** modélise un lien que possède une entité d'une classe avec les autres entités de cette même classe
- ▶ Les rôles sont obligatoires!
- ▶ Il faut surveiller les **cycles/boucles** et éventuellement les interdire!

Un exercice d'illustration

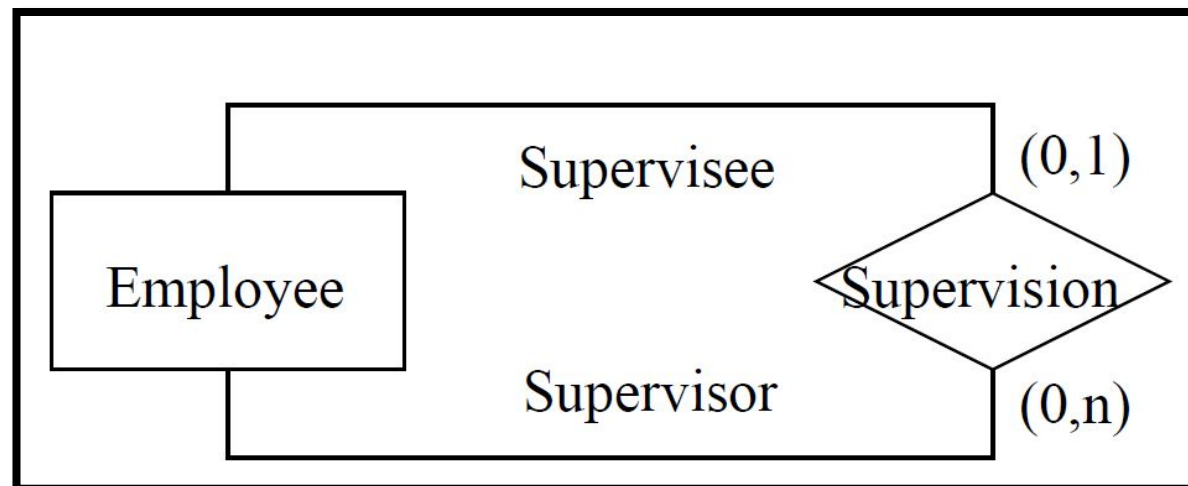
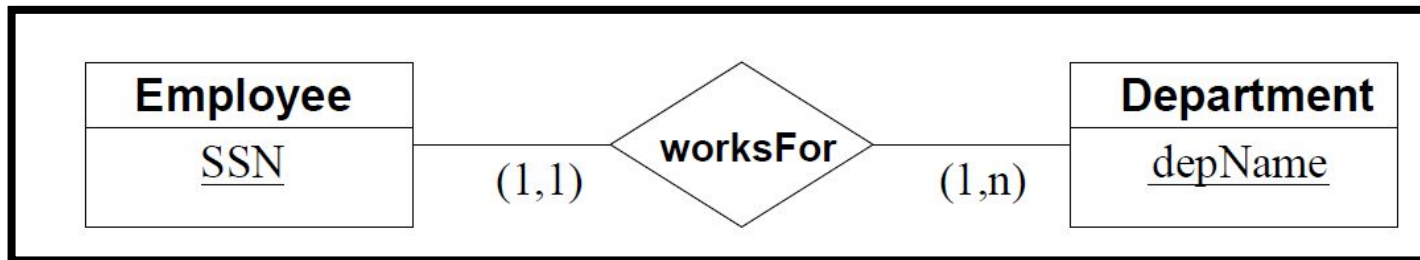
- ▶ Modélisez la relation récursive suivante: on identifie des employés dont certains sont supervisés par d'autres. Chaque employé ne peut être supervisé que par maximum un autre employé
- ▶ *Quel est le gros risque de cette modélisation?*

La Classe d'Entité faible

- ▶ Une **Classe d'Entité faible** est identifiée par une Association que cette Classe a avec une autre Classe d'Entité dite forte, et *éventuellement* un ou plusieurs de ses attributs
- ▶ L'Association doit avoir une cardinalité (1,1) du côté de la Classe d'Entité faible
- ▶ *Si une entité forte disparaît, toutes les entités faibles y étant liées disparaissent aussi*



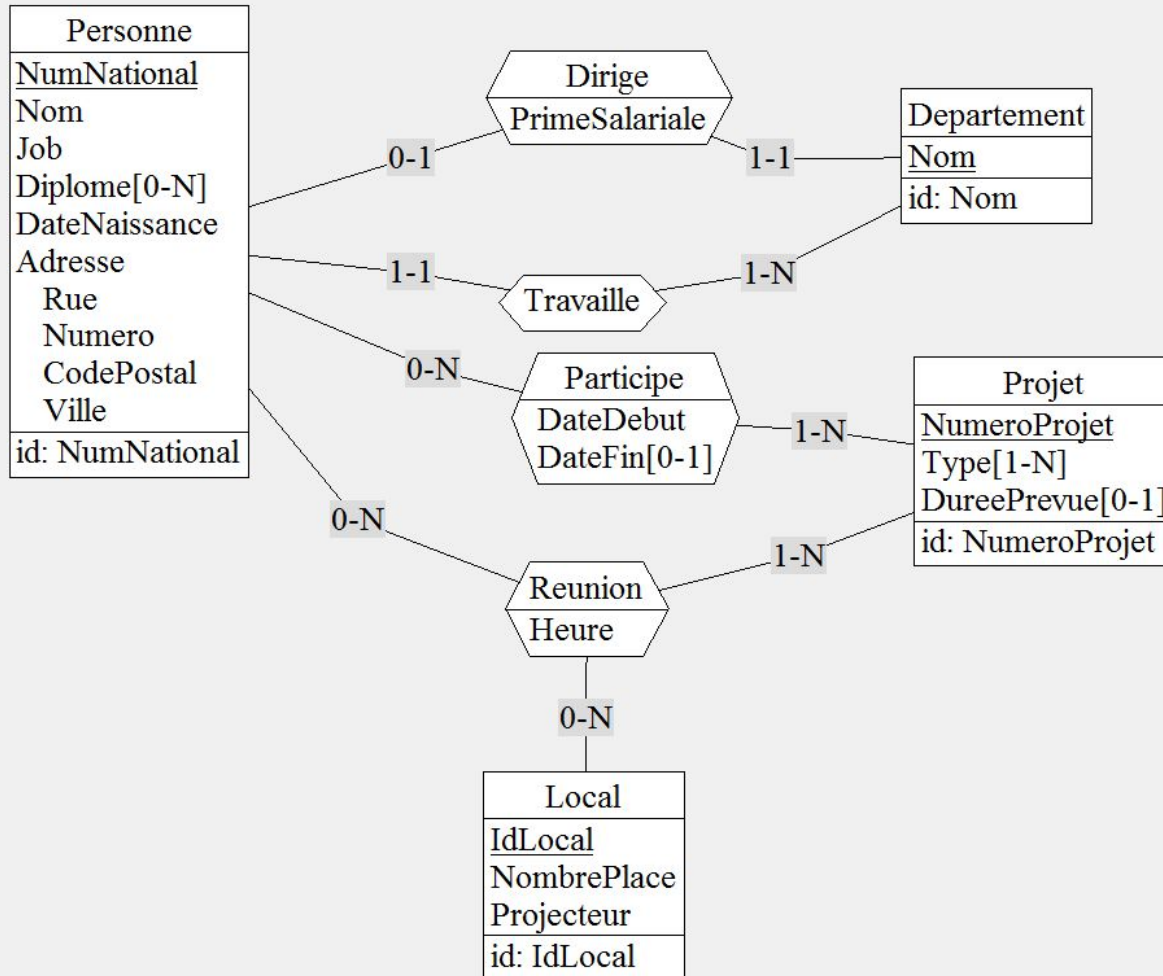
Autres notations pour l'Association



L'importance de la sémantique en modélisation

- ▶ La sémantique externe d'un modèle = **correspondance** entre chaque **construction** du modèle et les objets/**concepts du domaine d'application** qu'il représente
=> permet d'interpréter le schéma en termes accessibles aux utilisateurs
Doit être précis et complet!
- ▶ La sémantique externe intéresse le concepteur, le développeur et l'utilisateur

Un exemple de schéma Entité-Associations



- ▶ **Personne**
=>Employe
- ▶ **Truc & Astuce**: la dénomination à utiliser est souvent donnée dans les phrases concernant le cas à modéliser

Exercices

- ▶ Quelques exercices pour exploiter les bases du modèle Entité–Association