

Entité–Association et Relationnel

Société CogniTIC
contact@cognitic.be

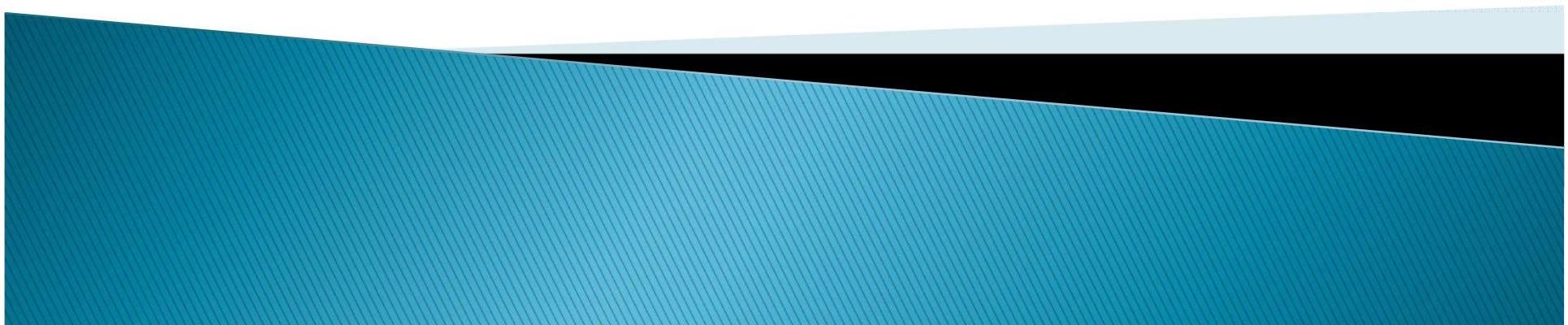


Table des matières



- ▶ Chapitre 1: Introduction aux bases de données
- ▶ Chapitre 2: Le modèle Entité–Association
 1. Méthodologie de modélisation
 2. La notion d'Entité et de Classe d'Entité
 3. Les relations: l'Association
 4. Les Attributs
 5. Les relations: la Spécialisation–Généralisation
 6. Les contraintes d'intégrités
- ▶ Chapitre 3: Le Schéma Relationnel
- ▶ (Chapitre 4: SQL)

Chapitre 2: Le modèle Entité-Association

» 1. Méthodologie de modélisation

Créer une base de données "en live"

- ▶ On ne crée pas une base de données d'une entreprise directement dans le workbench d'un SGBD!
 - Pas de communication, ou difficilement
 - Travail solitaire
 - Pas de modèle conceptuel ni logique
 - Pas de travail itératif de modélisation
 - Pas de documentation de l'implémentation (le plus souvent)
 - Source d'erreurs importantes (Rappel: en informatique, plus une erreur est découverte tôt avant l'implémentation, moins elle a de conséquences financières et temporelles)
 - ...
- ▶ Nécessité d'une approche disciplinée, structurée et basée sur des techniques et modèles rigoureux

Quelques chiffres

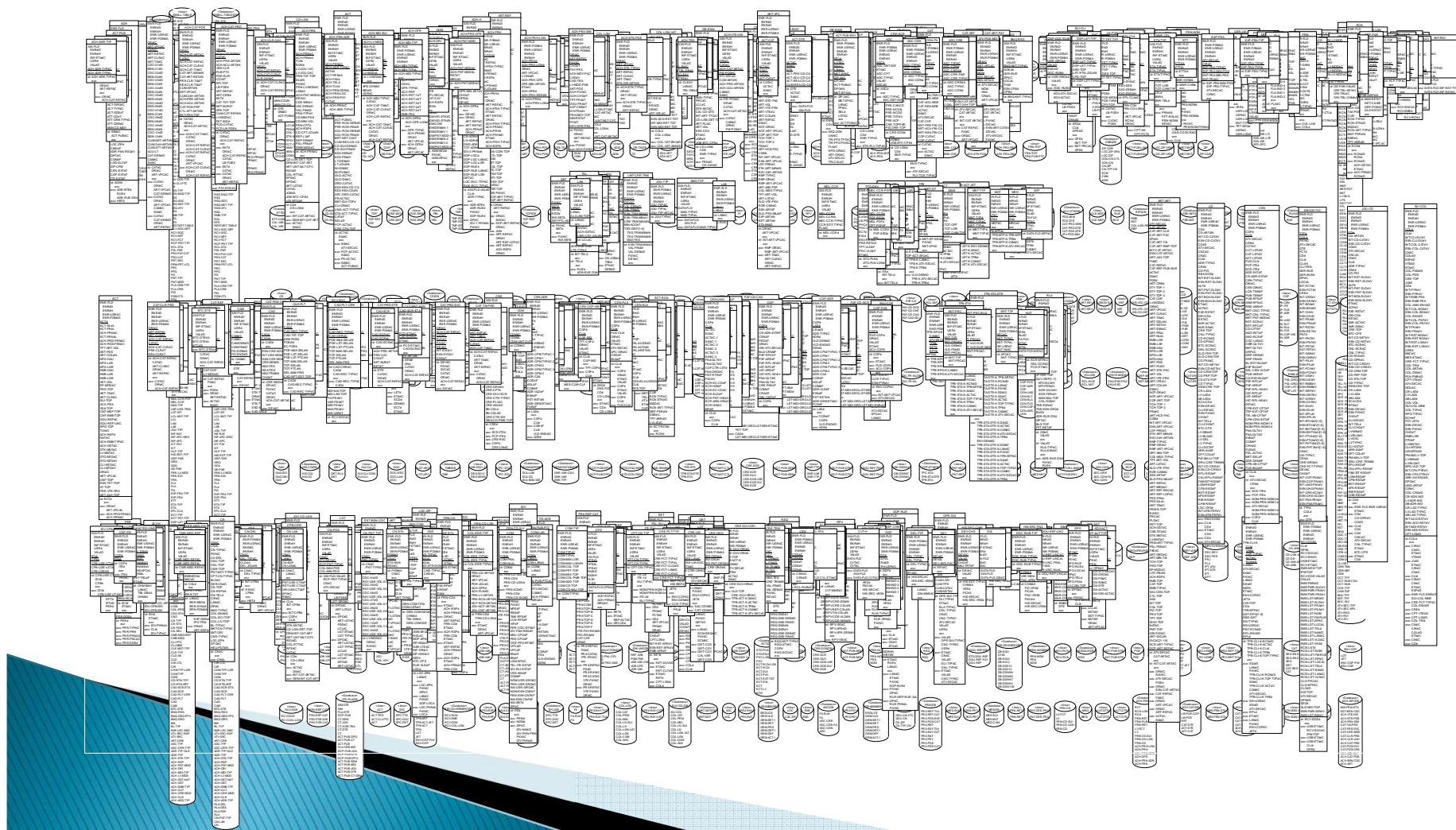
- ▶ Une base de données de taille moyenne:
 - comporte plusieurs centaines de tables
 - comporte plusieurs milliers de colonnes
 - ▶ Une base de données de grande taille:
 - comporte plusieurs milliers de tables
 - comporte plusieurs dizaines de milliers de colonnes
 - ▶ SAP utilise une BD de:
 - de près de 30.000 tables
 - de près de 200.000 colonnes
- Sans modèles de différents niveaux,
impossible de la comprendre!



Une exemple réel (1999)

Entreprise européenne de vente par correspondance

► 382 espaces de stockages et 853 tables



Le problème de modélisation

Comment traduire les besoins exprimés par les utilisateurs en structures de données ?

- ▶ Les utilisateurs = tous les agents (personnes, organismes, programmes et applications clients) impliqués dans la collecte, la production, la consultation, l'utilisation, la transmission, le traitement et la gestion des données
- ▶ Les besoins (ou exigences): ensemble des propriétés, caractéristiques et conditions que la base de données doit respecter pour satisfaire ses utilisateurs
 - besoins *fonctionnels*: contribuent à la fonction (au métier) des utilisateurs
 - besoins *non fonctionnels*: contribuent à la qualité des services rendus aux utilisateurs
- ▶ Les structures de données = le(s) schéma(s) de la base de données



Modéliser en 2 (?) étapes

Un ouvrage est une oeuvre littéraire publiée. Il est caractérisé par son numéro identifiant, son titre, son éditeur, sa date de première parution, ses mots-clés (10 au maximum), une brève note de présentation (ces notes sont en cours de constitution), le nom et le prénom de ses auteurs. A un ouvrage correspondent un certain nombre d'exemplaires, qui en sont la matérialisation physique. ...

```
create database BIB
create dbspace BIB_DATA;
create table OUVRAGE (
    NUMERO char(18) not null,
    TITRE varchar(60) not null,
    EDITEUR char(32) not null,
    DATE_1RE_PARUTION date not null,
    PRESENTATION varchar(255),
    primary key (NUMERO) in BIB_DATA;
    ...
alter table EXEMPLAIRE add constraint FKDE
    foreign key (NUMERO) references OUVRAGE;
    ...
create unique index IDOUVRAGE
    on OUVRAGE (NUMERO);
    ...
```



Conception de
base de données

besoins des
utilisateurs



Conception de
base de données

code SQL DDL

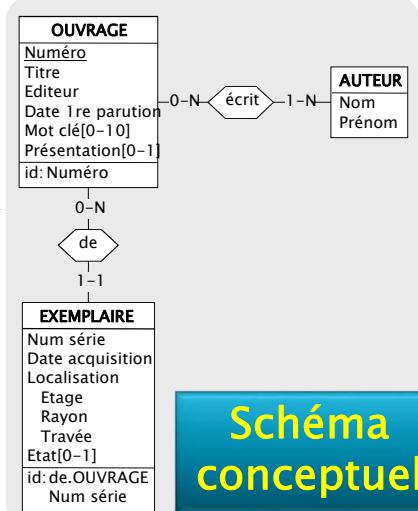
Méthodologie de développement de BD

Un ouvrage est une oeuvre littéraire publiée. Il est caractérisé par son numéro identifiant, son titre, son éditeur, sa date de première parution, ses mots-clés (10 au maximum), une brève note de présentation (ces notes sont en cours de constitution), le nom et le prénom de ses auteurs. A un ouvrage correspondent un certain nombre d'exemplaires, qui en sont la matérialisation physique. ...

Conception de base de données



Schéma conceptuel



```

create database BIB
create dbspace BIB_DATA;
create table OUVRAGE (
    NUMERO char(18) not null,
    TITRE varchar(60) not null,
    EDITEUR char(32) not null,
    DATE_1RE_PARUTION date not null,
    PRESENTATION varchar(255),
    primary key (NUMERO)) in
BIB_DATA;
...
alter table EXEMPLAIRE add constraint
FKDE
foreign key (NUMERO)references
OUVRAGE;
...
create unique index IDOUVRAGE
    on OUVRAGE (NUMERO);
...
    
```

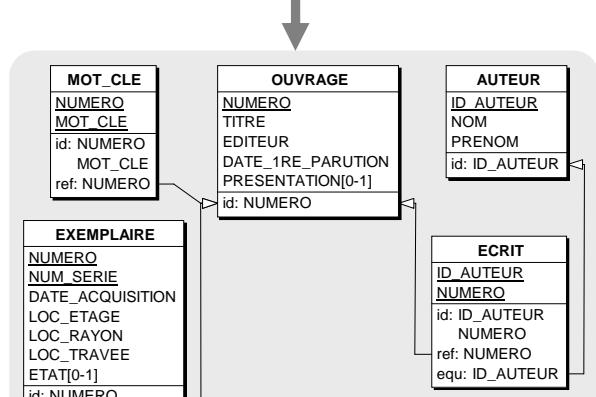


Schéma logique (relationnel)

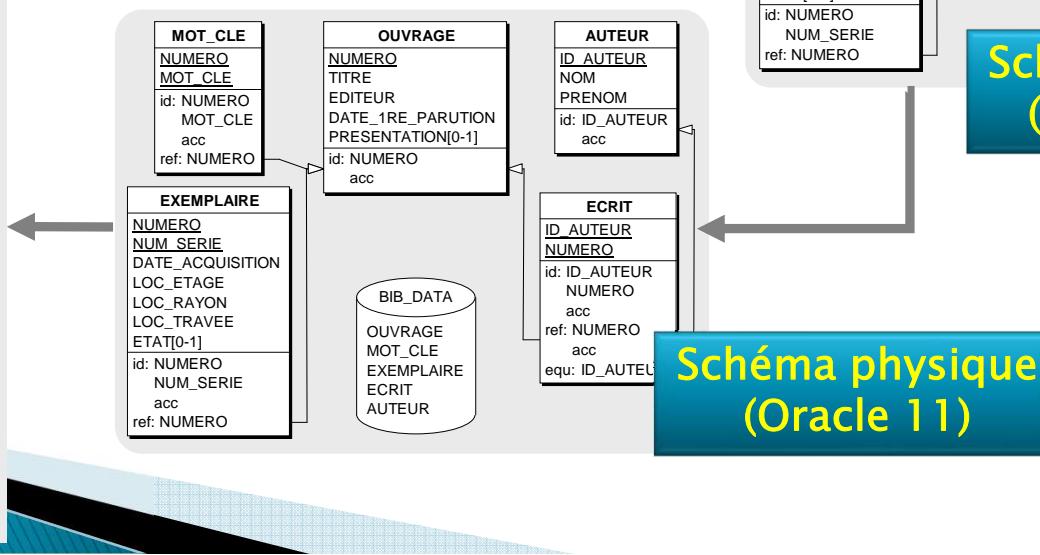


Schéma physique (Oracle 11)

Modèle VS. Schéma

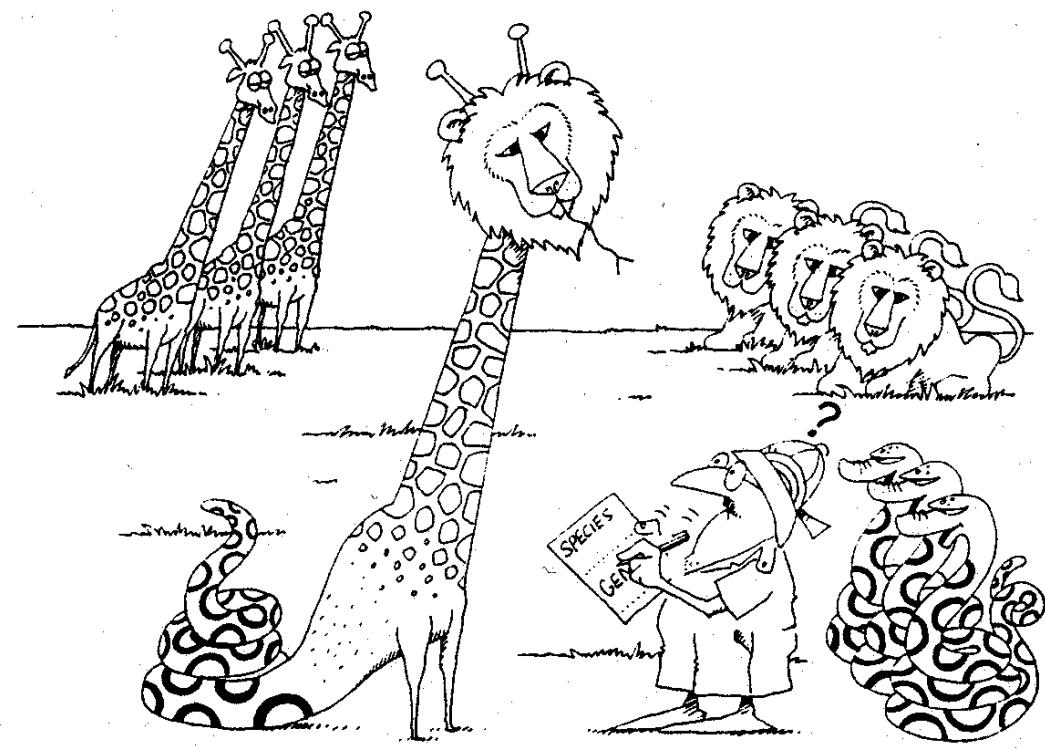
- ▶ Un modèle: un système formel de représentation de certains aspects de tous les domaines d'applications d'un certain type
 - Donne les constructeurs et la grammaire à suivre
 - En BD: représenter des objets, leurs attributs, leurs liens,...
- ▶ Un schéma: représentation des concepts statiques (et abstraits) d'un domaine d'application *particulier*
 - Un schéma peut être positionné avec différentes vues sur le domaine d'application
 - Un schéma peut être positionné à différents niveaux (ex: conceptuel, logique ou physique en BD)
- ▶ La population sont les différents instances d'un schéma particulier, à un moment particulier

Chapitre 2: Le modèle Entité-Association

» 2. La notion d'Entité et de
Classe d'Entité

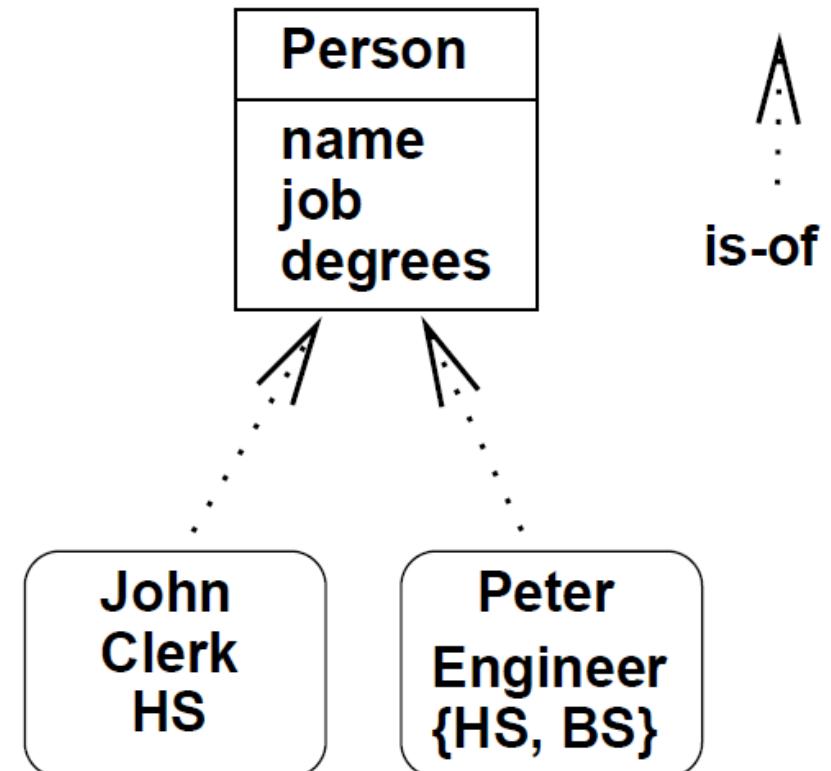
L'Entité

- ▶ Une Entité correspond à un objet du monde réel
 - Entité physique: des personnes, des machines, des produits,...
 - Entité conceptuelle/abstraite: une société, un travail, une commande,...



La Classe d'Entités

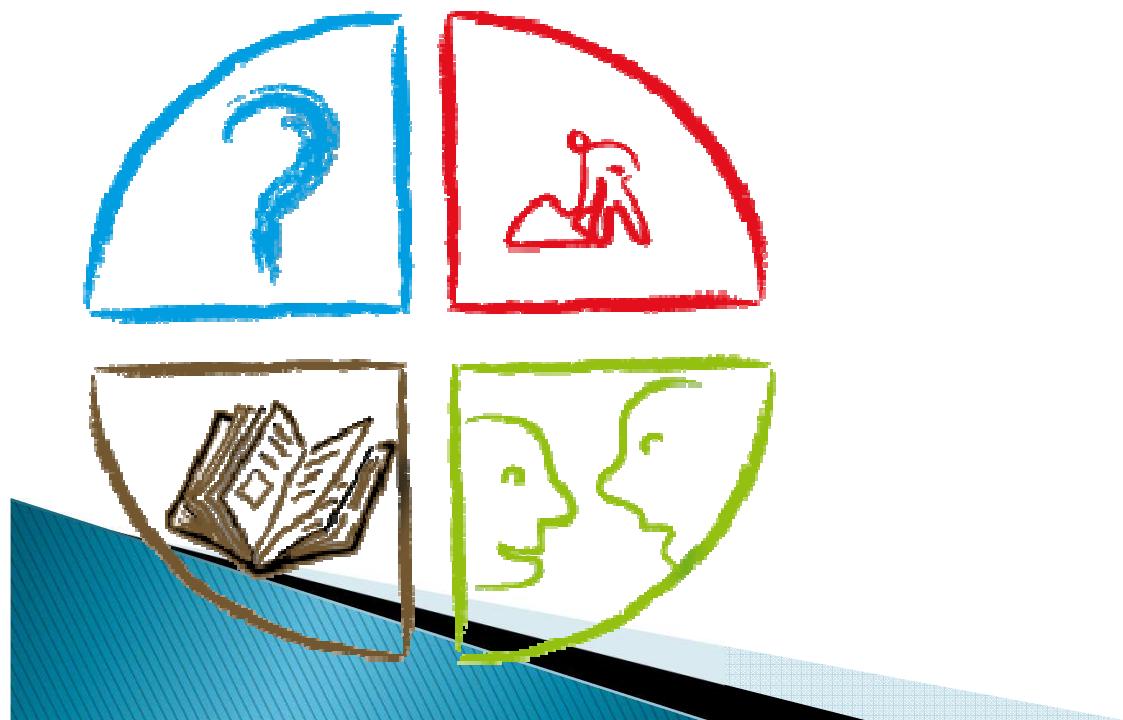
- ▶ Une Classe d'Entité
 - Concept générique dans lequel on classe/instancie plusieurs Entités ayant des caractéristiques similaires
 - Une Classe d'Entité définit un template commun pour un ensemble d'Entité



Exercices



- ▶ Quelques exercices pour exploiter les concepts d'Entité et de Classe d'Entités



Exercice 1

- ▶ Voici trois entités :
 - *Jules Dumoulin, 25 ans, habitant rue d'Erquelines n° 18 à 5983 Mont-sur-Pieds, marié, 1 enfant*
 - *Lisa Lambert, 78 ans, habitant rue d'Emailas n°134 à 1930 Souson, veuve, 4 enfants*
 - Henri, cheveux bruns, yeux bleus, 187cm, 56 kilogrammes, vendeur de meubles
- ▶ Identifiez l'entité ne faisant pas partie de la même classe que les autres
Expliquez pourquoi

Exercice 2

- ▶ Voici deux entités :
 - *Jules Dumoulin, 25 ans, habitant rue d'Erquelines n° 18 à 5983 Mont-sur-Pieds, marié, 1 enfant*
 - *Lisa Lambert, 78 ans, habitant rue d'Emailas n°134 à 1930 Souzon, veuve, 4 enfants*
- ▶ Nommez la classe pouvant regrouper ces deux entités et donnez des caractéristiques partagées

Exercice 3

- ▶ Voici la description d'une Classe d'Entités:
Film(Titre, AnneeProd, ActeurPrincipal, Langue, Réalisateur, NbrePrix)
- ▶ Voici l'extrait d'un article; donnez l'entité correspondant à la classe d'entité ci-dessus:
Le film mythique « Dear Friends » produit par Universal dans le Kansas, Etats-Unis, en 1987 et sorti dans les salles l'année suivante (1988) vient encore de remporter un prix récemment. Ce prix, nouveau triomphe pour l'équipe de production, a été remis à la veuve du réalisateur, Thomy Lee. L'acteur José Despero était évidemment présent. Son rôle central dans le film a encore été souligné. Sa partenaire, Monique Poncin, était également présente, bien que souffrante. Ce film va probablement être adapté pour être joué au théâtre selon le producteur de la troupe théâtrale « Moments Heureux ».



Chapitre 2: Le modèle Entité-Association

» 3. Les Attributs

Les attributs: définition

- ▶ Un Attribut d'une Classe est une caractéristique partagée par (toutes?) ses Entités
- ▶ Chaque Attribut a un domaine de valeurs (type) précis
- ▶ Chaque Entité a une valeur pour les (chaque?) attributs de sa Classe
- ▶ Attribut atomique

Personne
Nom
Job
Diplome

La cardinalité d'un Attribut

- ▶ Une cardinalité: concept mathématique définissant le nombre d'élément (ou de relations) dans/entre ensembles finis
- ▶ Soit la cardinalité $[x,y]$
 - x : borne inférieure; y : borne supérieure
 - Donne le nombre min x et max y de valeur que peut prendre une Entité pour cet attribut
 - $x \leq y$
- ▶ Optionnel VS. Obligatoire
 - Optionnel: $x = 0$
 - Obligatoire: $x \geq 1$
- ▶ Multiplicité d'une cardinalité
 - Unique (une valeur): $y = 1$
 - Multiple (plusieurs valeurs): $y > 1$
- ▶ Par défaut: $[1,1]$

Personne
Nom
Job
Diplome[0-N]

Personne
Nom
Job
Diplome[0-N]
Tel[1-3]

La valeur NULL

- ▶ Si une cardinalité est optionnelle, alors on peut ne pas avoir de valeur pour cet attribut lors de l'enregistrement/modification d'une entité particulière
- ▶ *NULL* (le "rien informatique")
 - Pas applicable: ne s'applique pas à l'entité
 - Inconnu actuellement, mais la valeur existe: on sait que l'entité possède une valeur pour l'attribut sans encore la connaître
 - Inconnu actuellement, la valeur existe peut-être: on ne connaît pas la valeur et on ne sait pas si elle existe
- ▶ Concept central à toujours garder dans un coin de la tête lorsque l'on réalise des requêtes SQL DRL
(et principalement des requêtes agrégatives, par ex. calculer une moyenne ou compter un nombre d'enregistrement)

Attribut dérivé

- ▶ Si deux attributs sont en relation, ils sont appelés attributs dérivés
 - Ex: Age peut-être dérivé de DateNaissance
 - **Redondance!**
- ▶ Garder l'attribut le plus pérenne dans le temps

Personne
Nom
Job
Diplome[0-N]
DateNaissance
Age

Attribut composite

- ▶ Un **Attribut composite** est un attribut qui est décomposé en d'autres attributs
- ▶ Un attribut composite est donc un ensemble de valeurs mise en relation
- ▶ Un attribut atomique est donc un attribut indivisible
- ▶ C'est un groupe de valeurs qui n'est pas suffisant que pour être une entité en tant que tel

Personne
<u>NumNational</u>
Nom
Job
Diplome[0-N]
DateNaissance
Adresse
Rue
Numero
CodePostal
Ville
id: NumNational

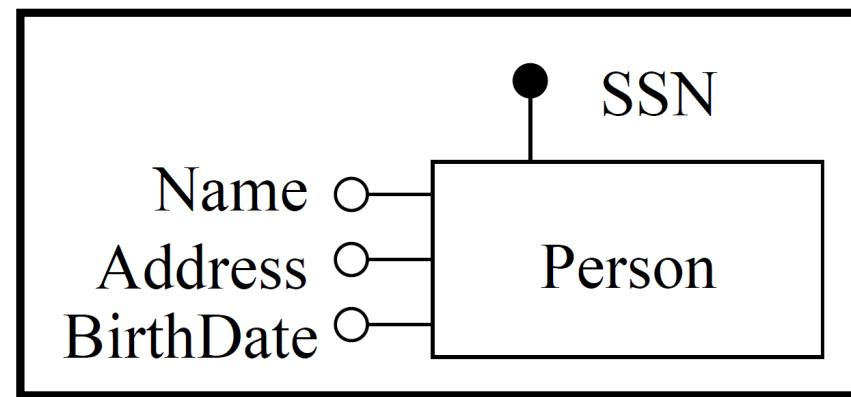
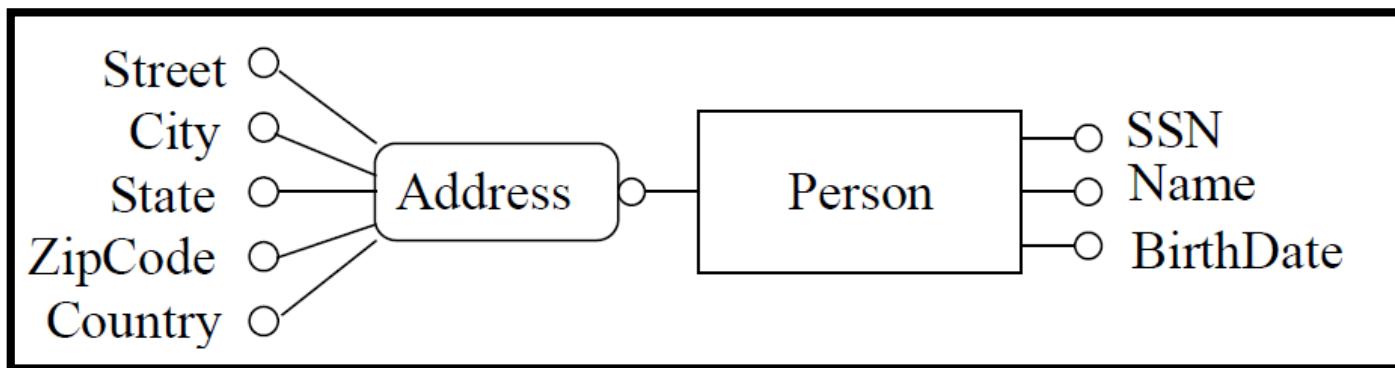
Attribut identifiant

- ▶ Un attribut identifiant est un attribut (simple ou complexe) dont la valeur peut à elle seule identifier chaque entité de la Classe par rapport aux autres
 - *Unicité*: pas deux fois la même valeur pour deux entités dans la classe
 - *Valeur obligatoire et unique*: pas de valeur NULL
=>cardinalité: [1,1]
- ▶ Une entité peut avoir plusieurs identifiants, un seul est choisi comme attribut identifiant
- ▶ Un identifiant peut être composé de plusieurs attributs simples ou composites

Personne
<u>NumNational</u>
Nom
Job
Diplome[0-N]
DateNaissance
id: NumNational

Personne
<u>Nom</u>
Job
Diplome[0-N]
DateNaissance
<u>Adresse</u>
Rue
Numero
CodePostal
Ville
id: Nom
Adresse

Autres notations pour les différents types d'Attributs

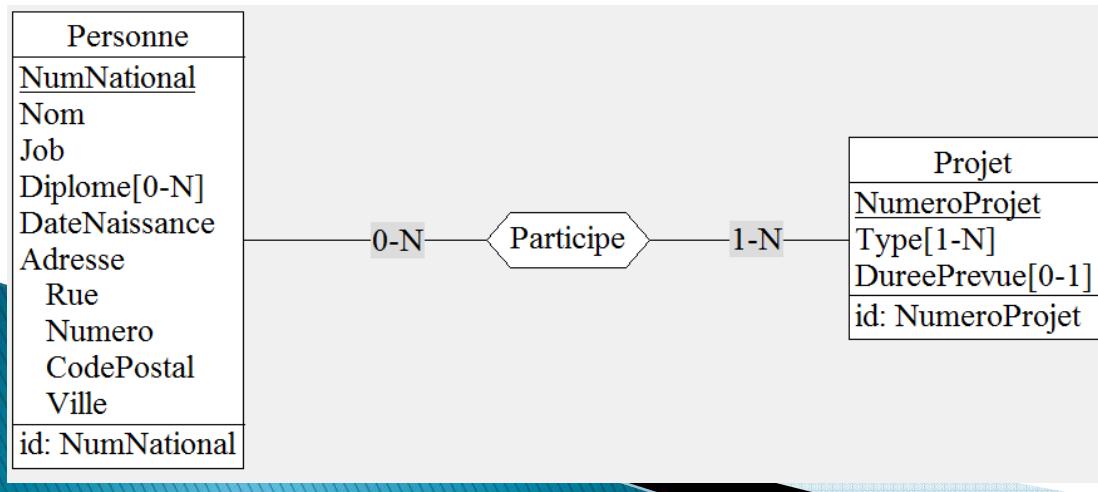
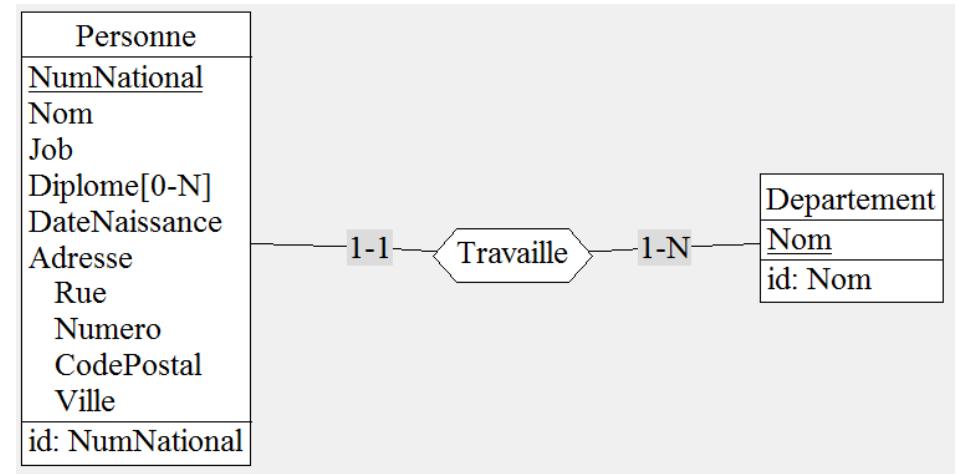


Chapitre 2: Le modèle Entité-Association

» 4. Les relations: l'Association

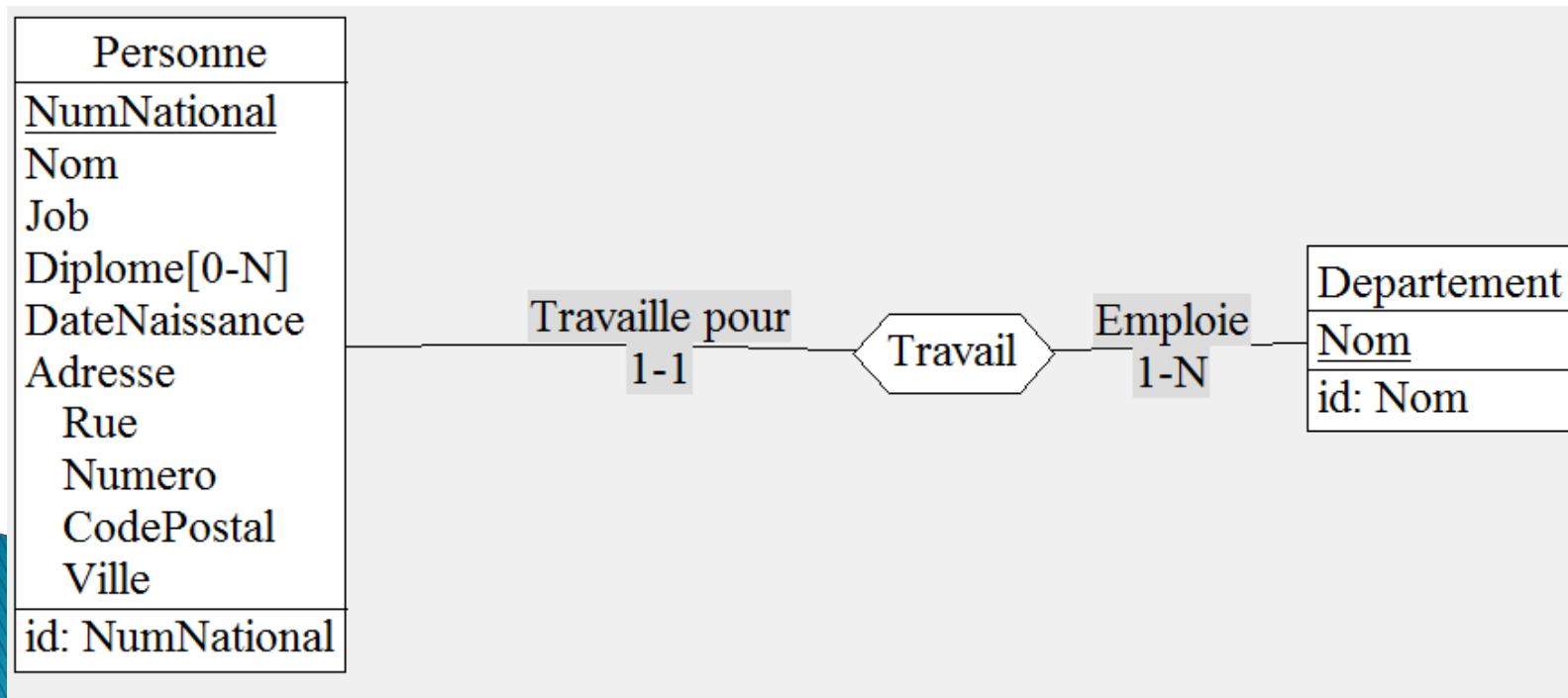
L'Association: définition

- ▶ 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!

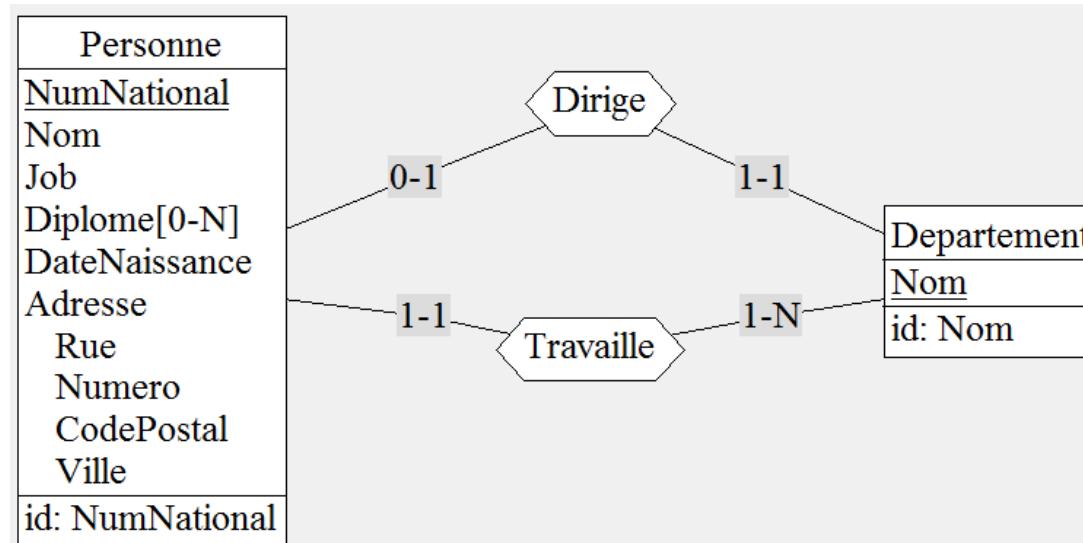


Les rôle d'une Association

- ▶ 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

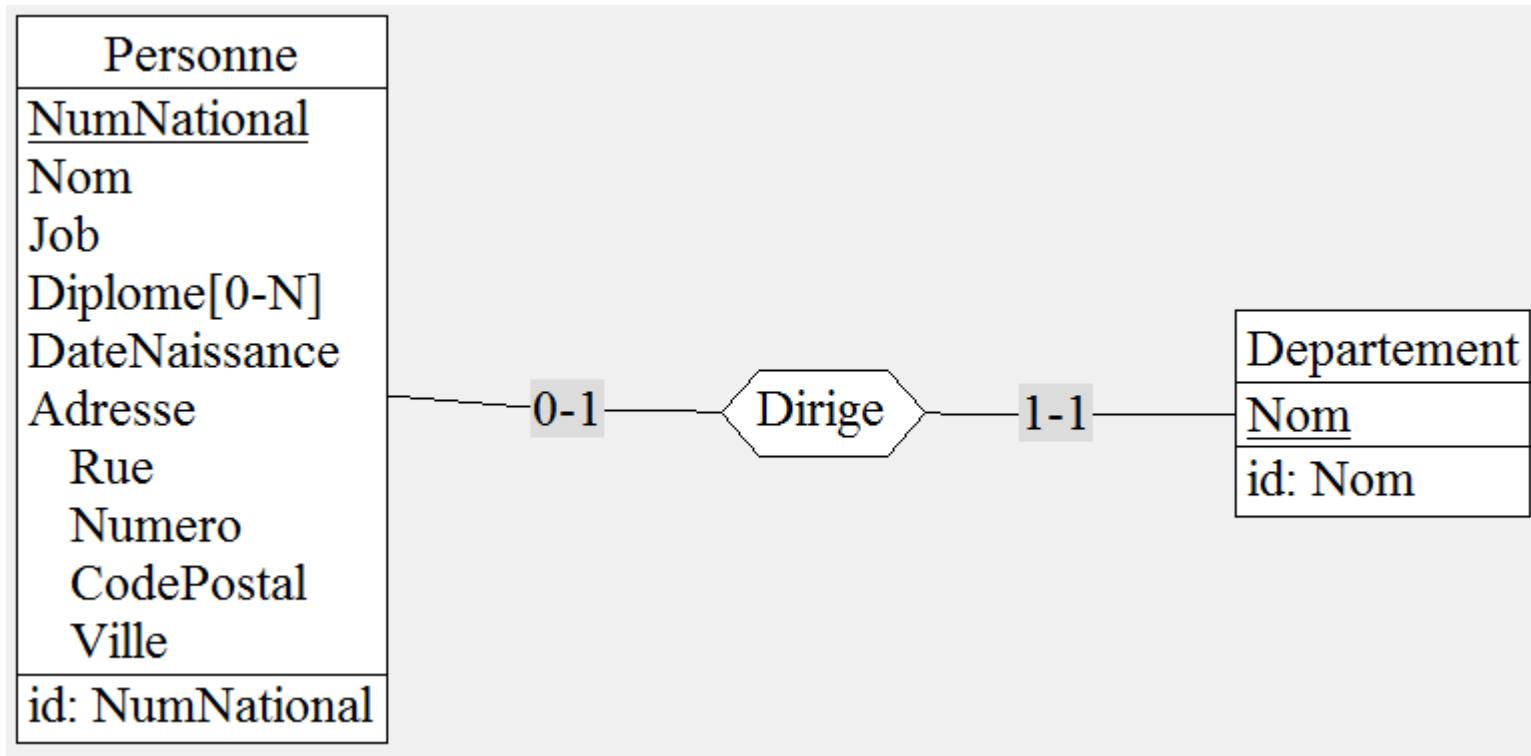


- ▶ Chaque association binaire a 2 cardinalités, une pour chacun des rôles
- ▶ Une cardinalité a une borne supérieure et une borne inférieure
- ▶ Les cardinalités de l'Association '*Travaille*' se lisent:
 - De 'Personne' vers 'Departement': "Une entité de la classe Personne a une et une seule relation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

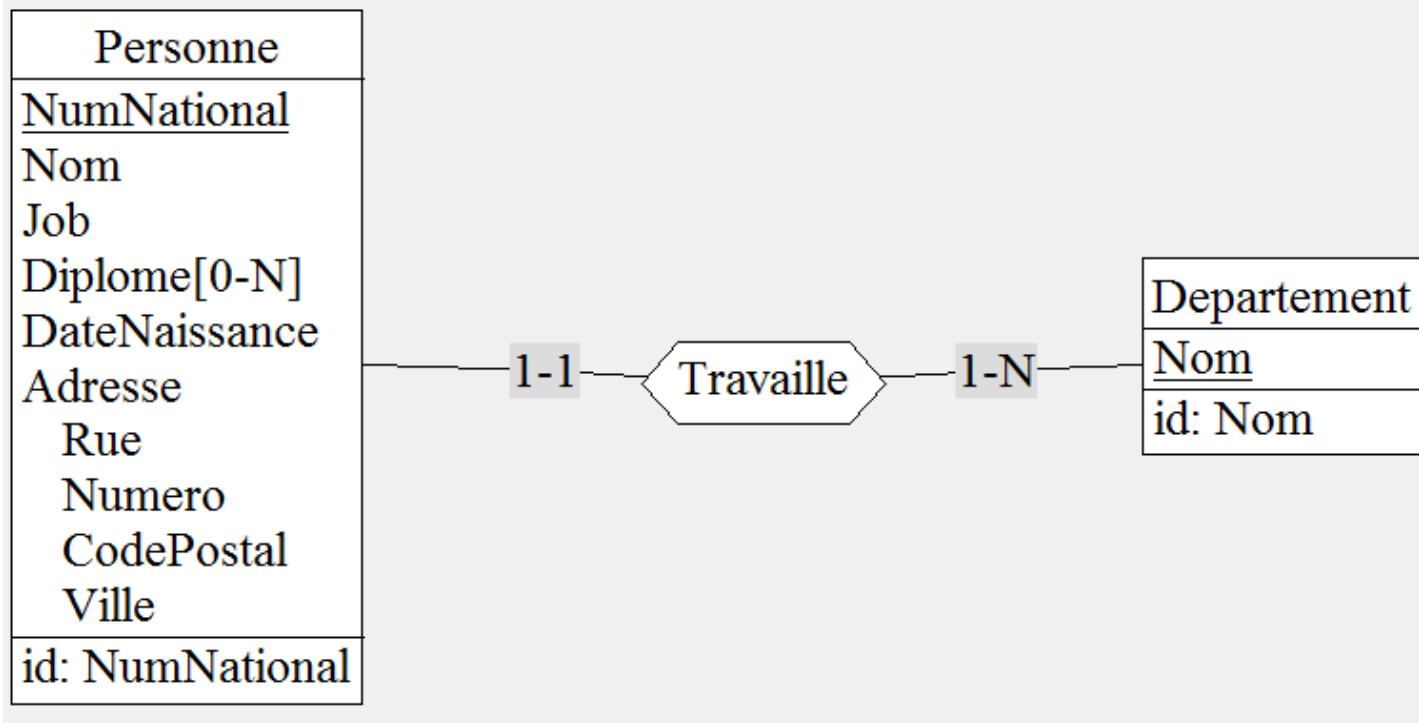
- ▶ Soit la cardinalité d'une Association (x,y)
- ▶ Optionnel VS. Obligatoire
 - **Optionnel:** $x = 0$
Une entité *peut* avoir une relation avec les entités de la classe mise en relation
 - **Obligatoire:** $x \geq 1$
Chaque entité *doit* avoir une relation avec les entités de la classe mise en relation
- ▶ 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



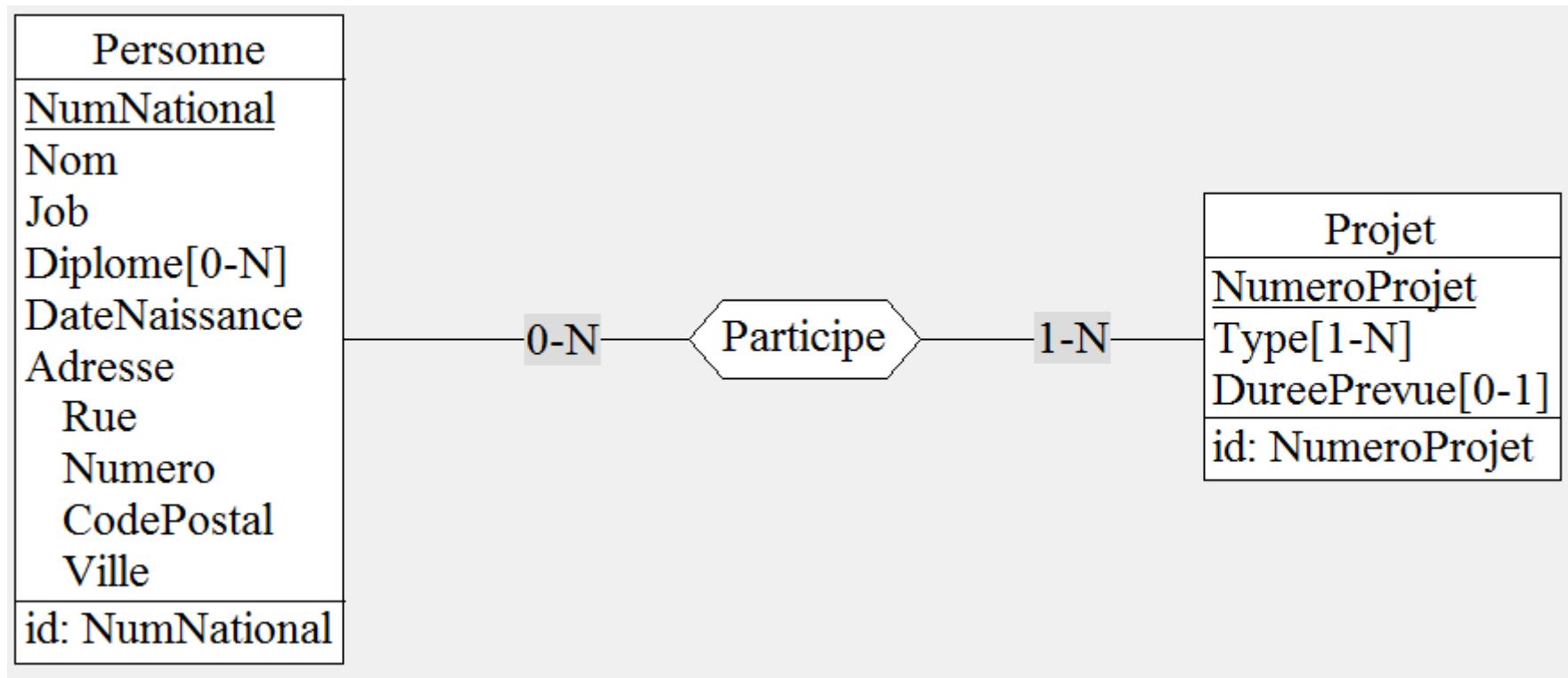
- ▶ Une ‘Personne’ peut diriger un seul ‘Departement’
- ▶ Un ‘Departement’ est dirigé par une et une seule ‘Personne’

Association One-to-Many



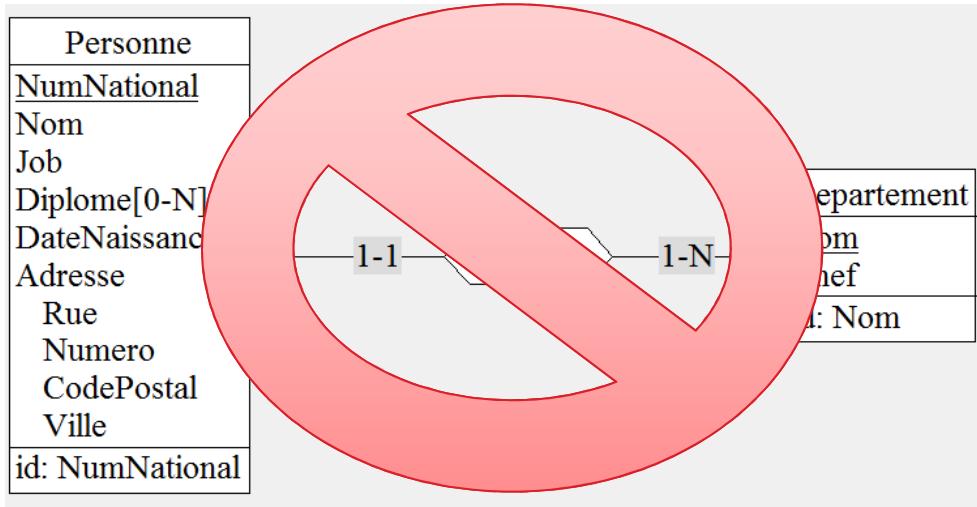
- ▶ Chaque ‘Personne’ travaille pour un seul ‘Departement’
- ▶ Chaque ‘Departement’ emploie de une à n ‘Personnes’
- ▶ Une association One-to-Many est similaire à une association Many-to-One (lecture des associations dans les deux sens!)

Association Many-to-Many

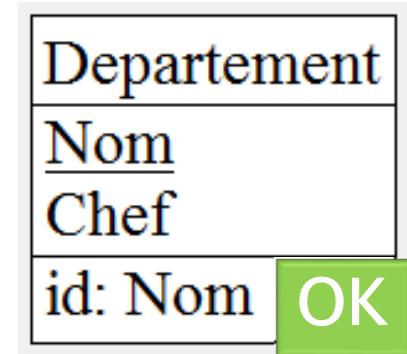


- ▶ Chaque entité ‘Personne’ peut travailler pour plusieurs ‘projets’
- ▶ Chaque entité ‘Projet’ utilise au moins une ‘personne’, jusque n ‘personne’

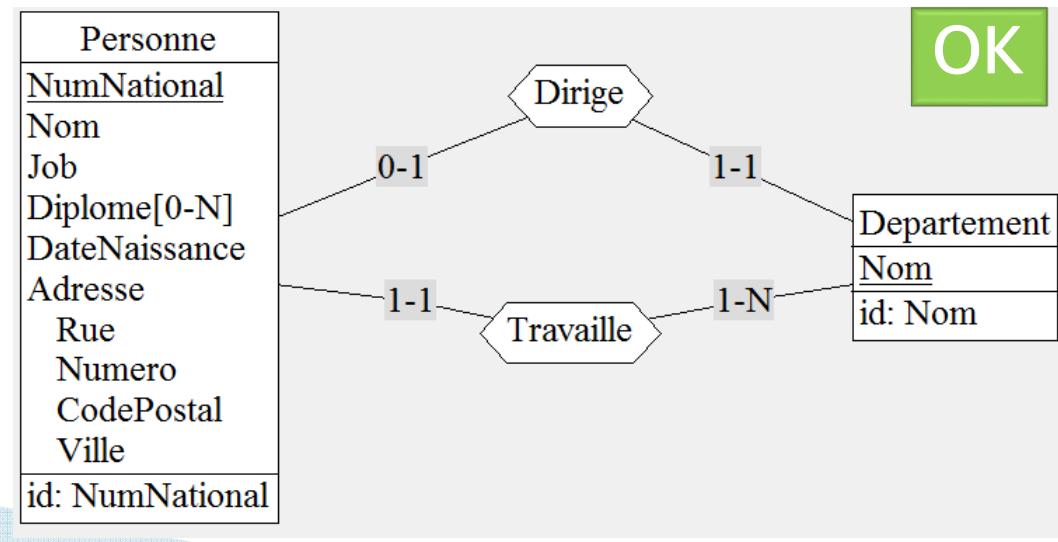
Utiliser adéquatement les Associations



- Si la classe ‘Personne’ n’est pas dans le schéma

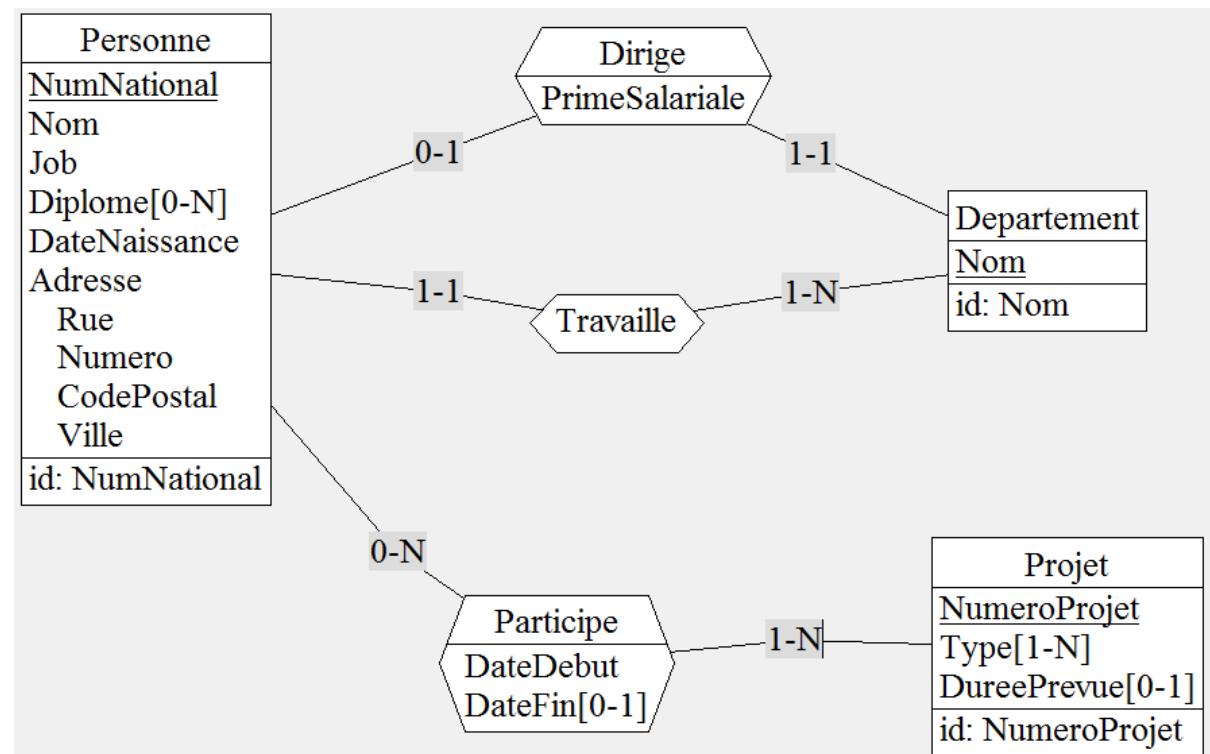


► Utiliser des relations pour capturer les liens du monde réel lorsque une classe de votre schéma capture la même donnée!

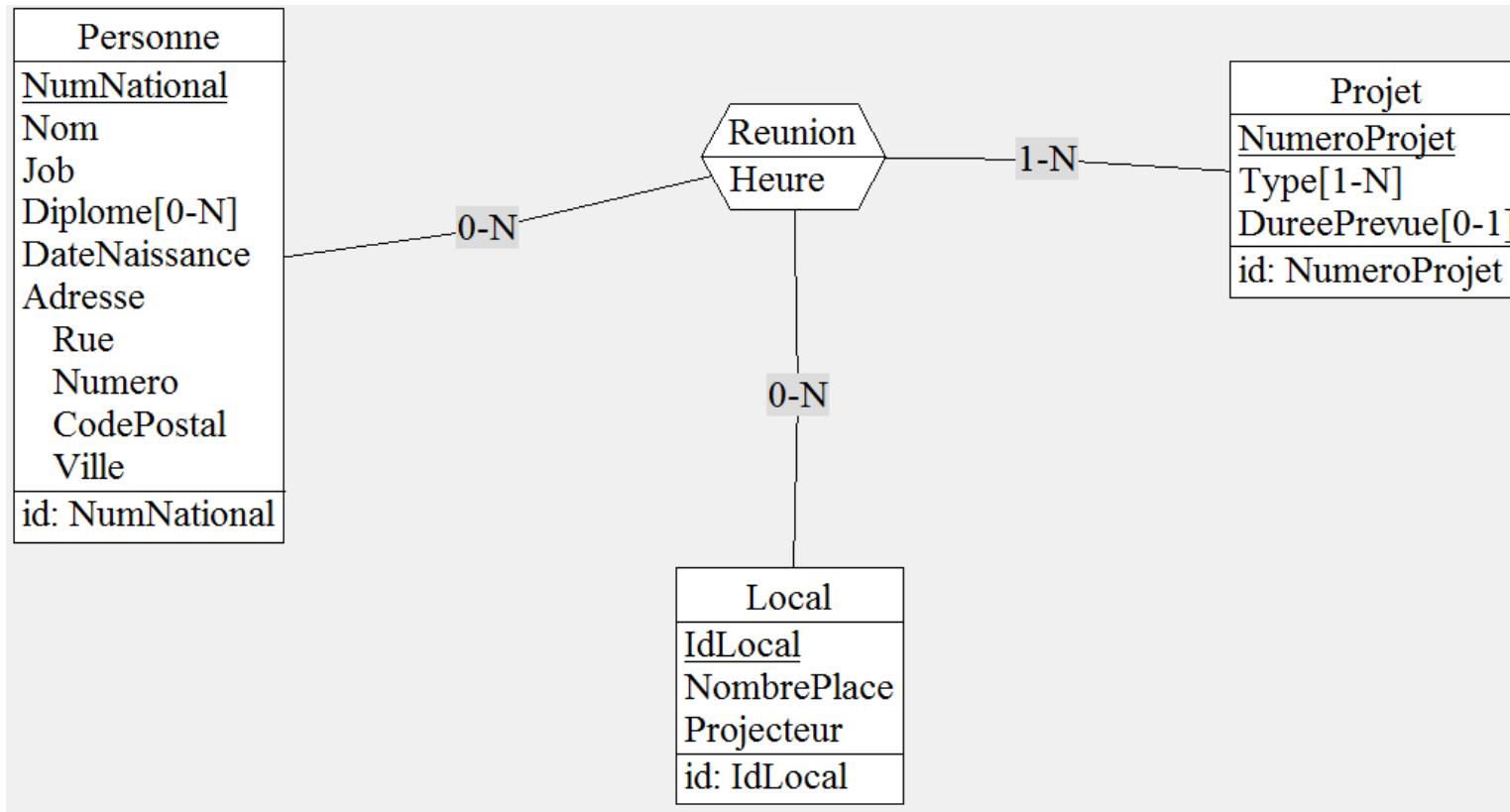


Les Attributs des Associations

- ▶ 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

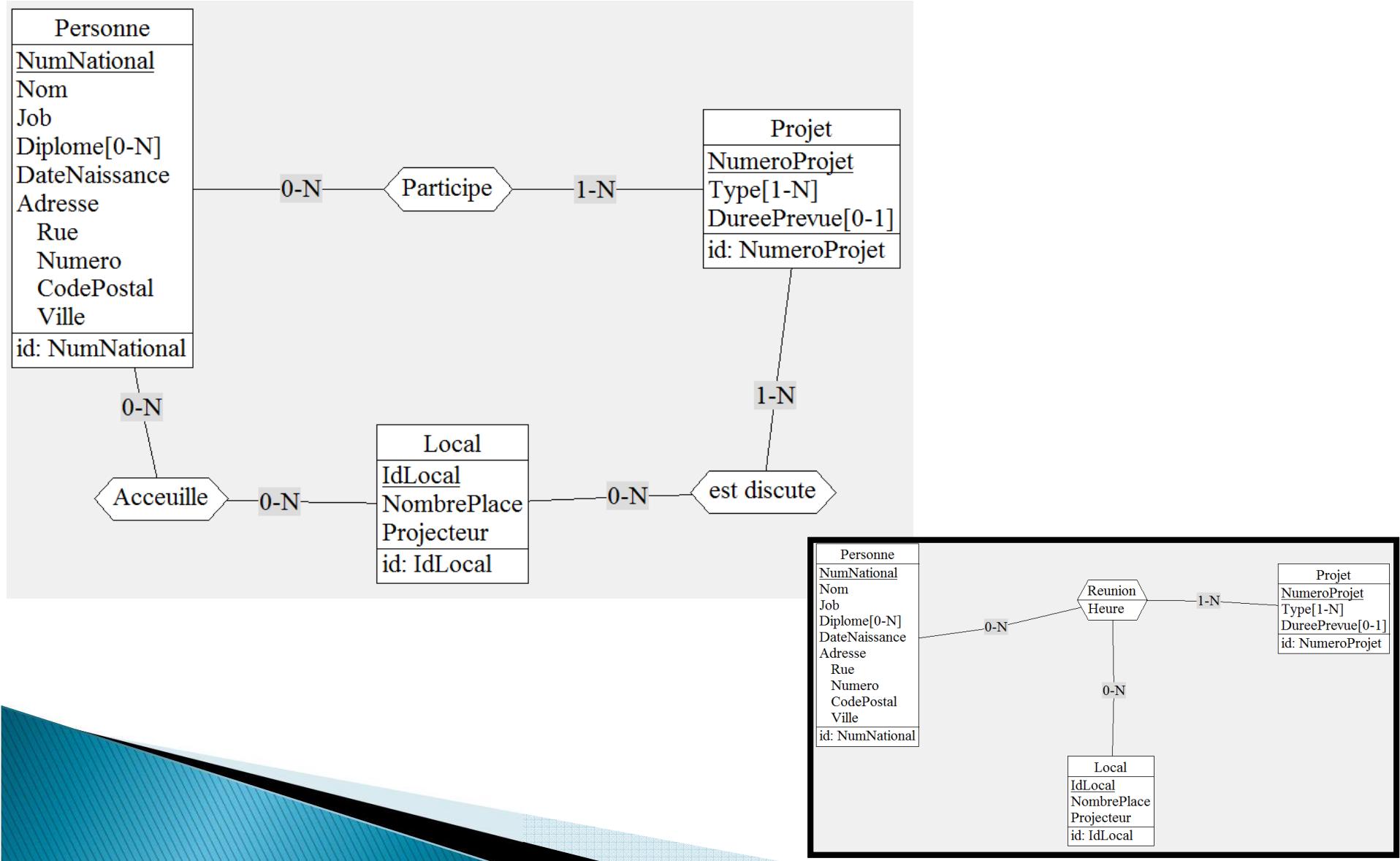


L'Association Ternaire



- ▶ 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

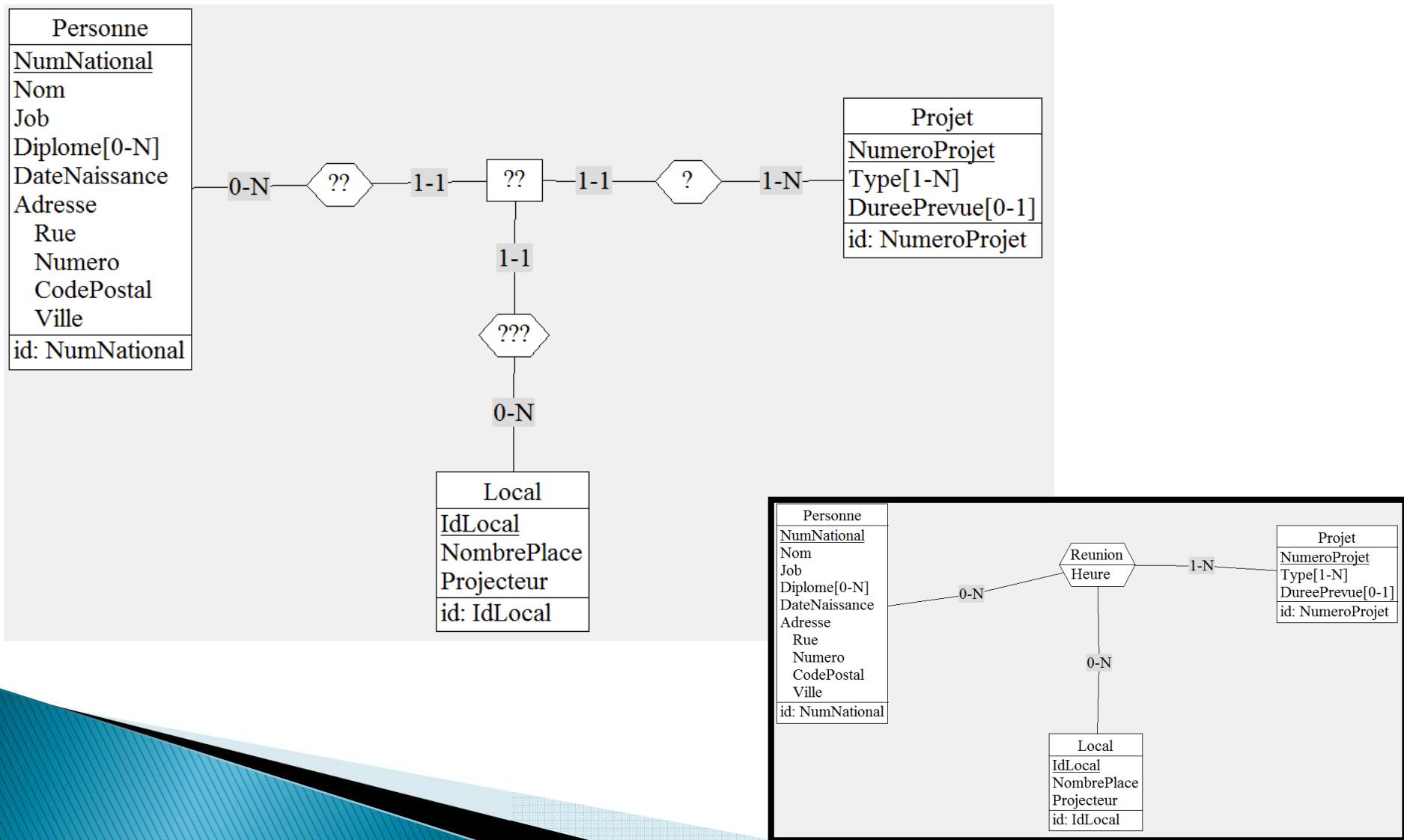
- ▶ Un exemple avec des instances
Reunion

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

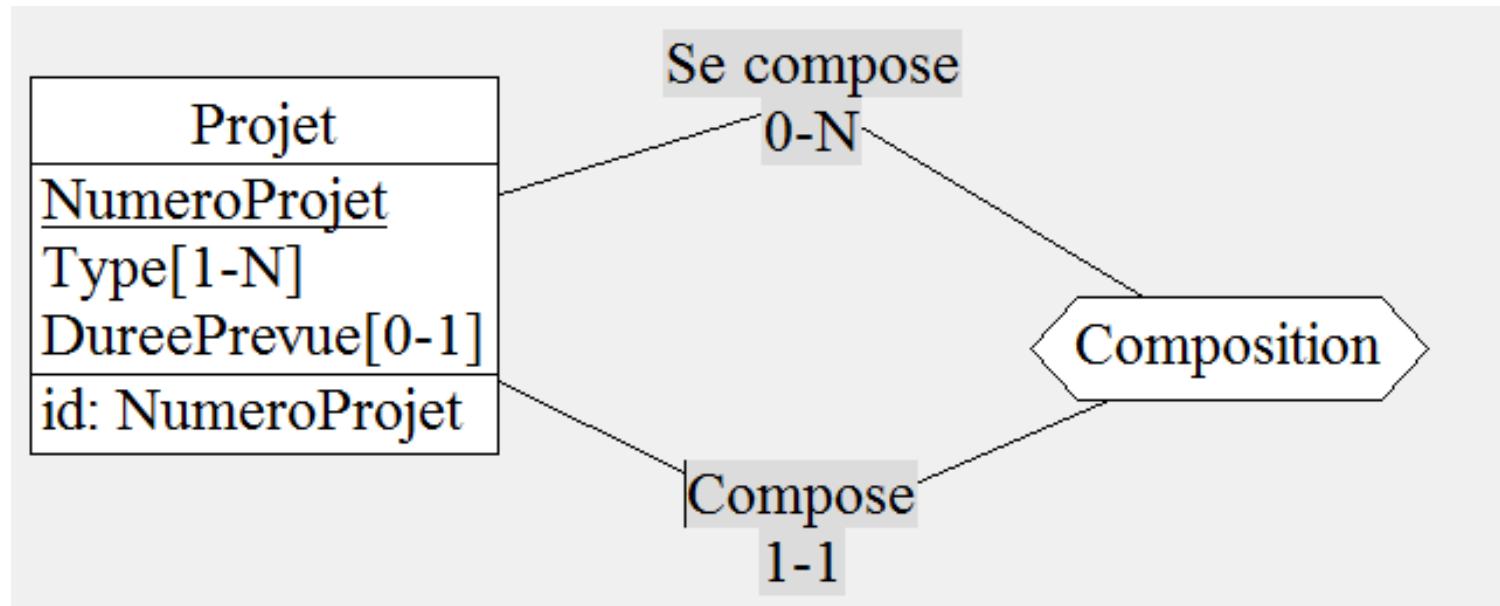
Participe		Accueille		Est discute	
Personne	Projet	Personne	Local	Local	Projet
Georges	Modélisation n°1	Georges	Salle 17	Salle 17	Modélisation n°1
Lucas	Modélisation n°2	Lucas	Salle 18	Salle 18	Modélisation n°1
Georges	Modélisation n°1	Georges	Salle 18	Salle 18	Modélisation n°2

On peut inférer des relations dérivées avec les trois associations binaires qui n'existent pas dans la relation ternaire

Modélisation correcte d'une relation ternaire avec des associations binaires



L'Association Réursive (ou Association cyclique)



- ▶ L'Association Réursive 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

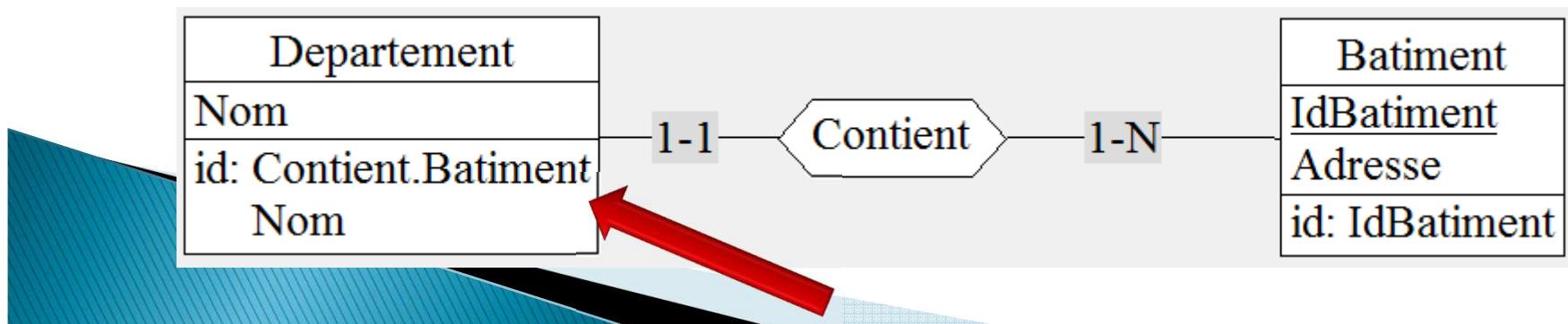


- ▶ Modélez la relation récursive suivante: on identifie des employés dont certains sont supervisés par d'autres. Chaque employé ne peut être supervisé que par maximum un autre employé
- ▶ *Quel est le gros risque de cette modélisation?*

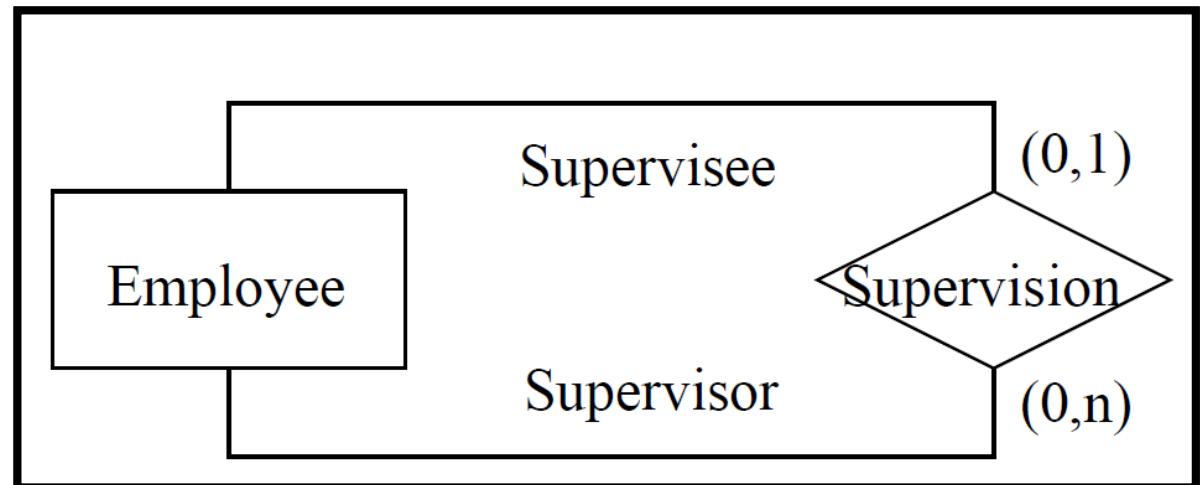
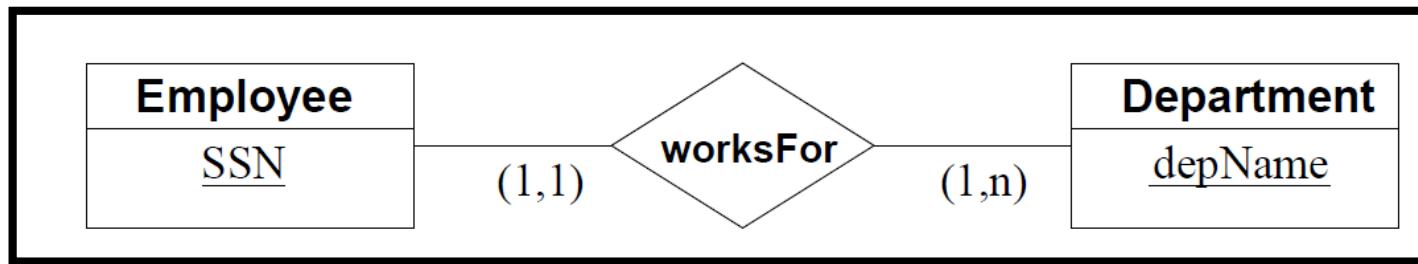


La Classe d'Entité faible

- ▶ 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*

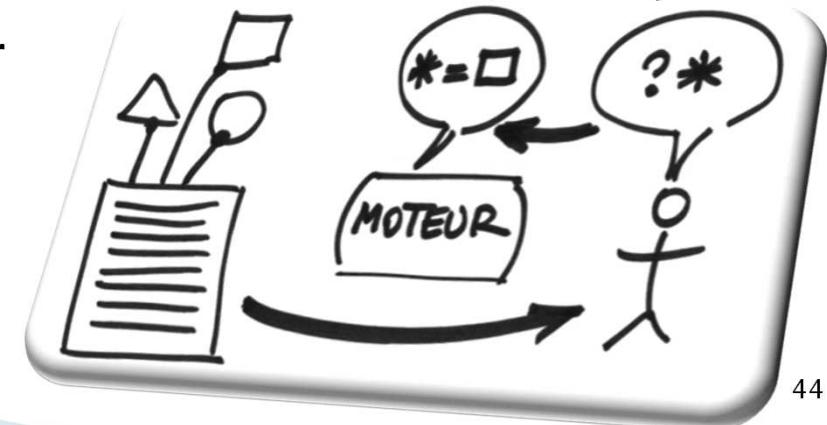


Autres notations pour l'Association

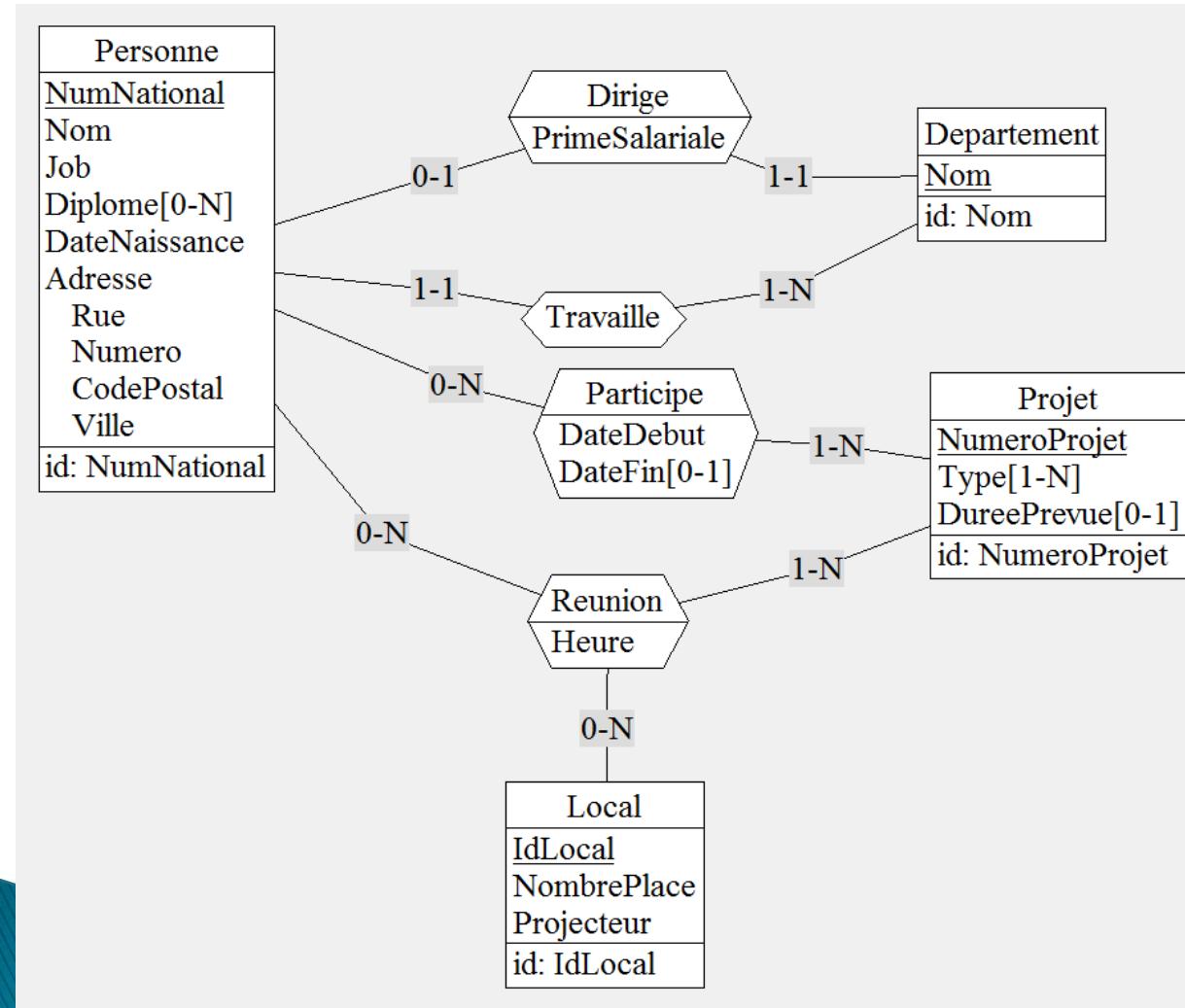


L'importance de la sémantique en modélisation

- ▶ 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

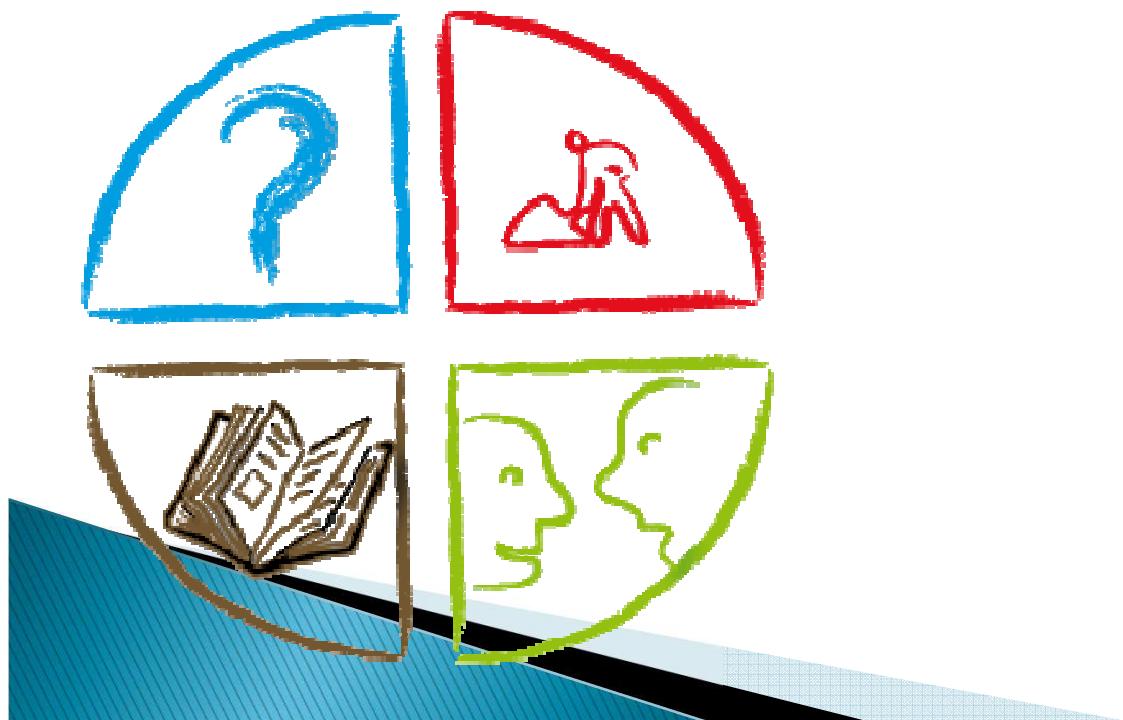


- ▶ **Personne**
=> Employe
- ▶ Truc & Astuce: la dénomination à utiliser est souvent donnée dans les phrases concernant le cas à modéliser

Exercices



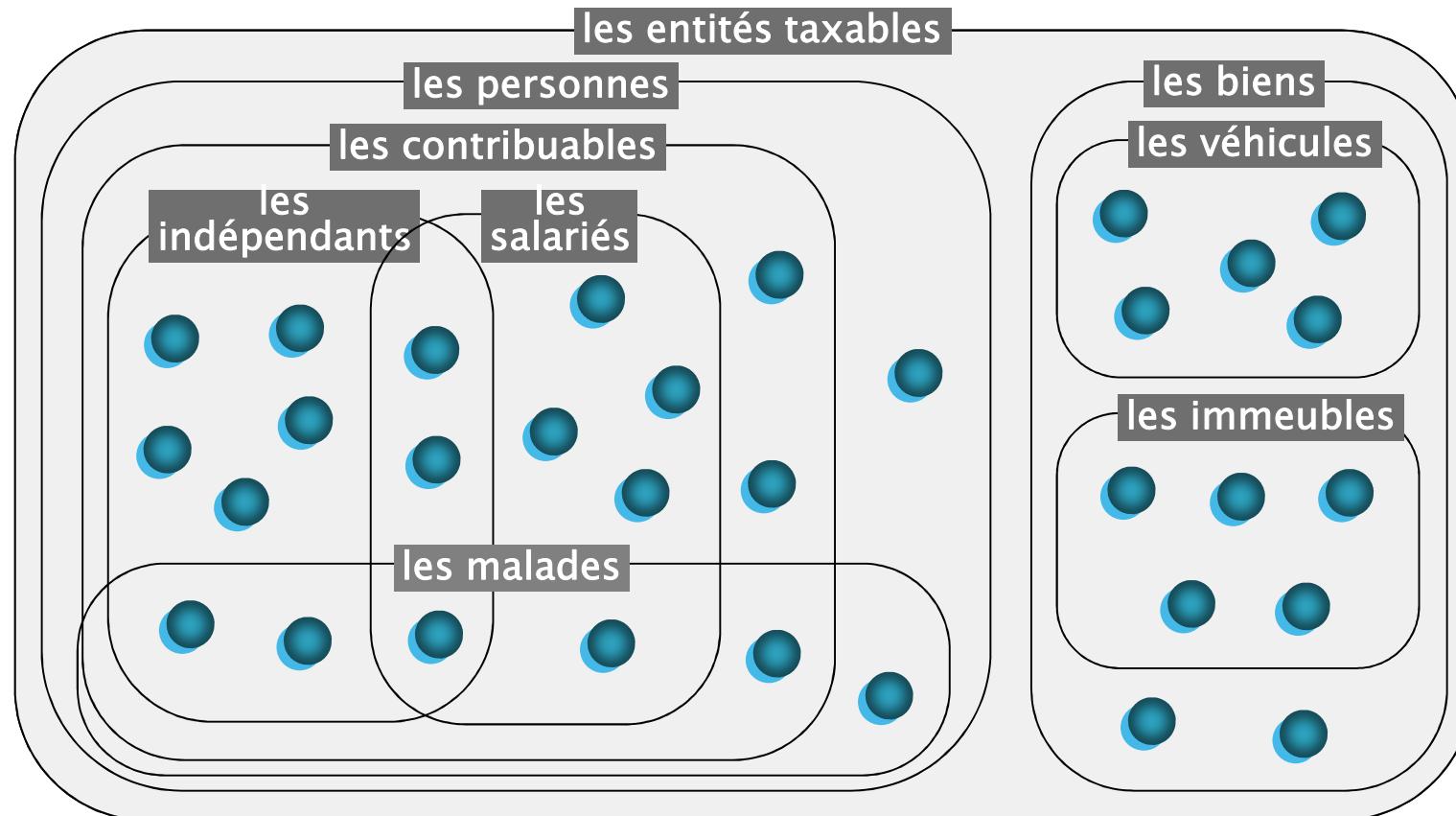
- ▶ Quelques exercices pour exploiter les bases du modèle Entité-Association



Chapitre 2: Le modèle Entité-Association

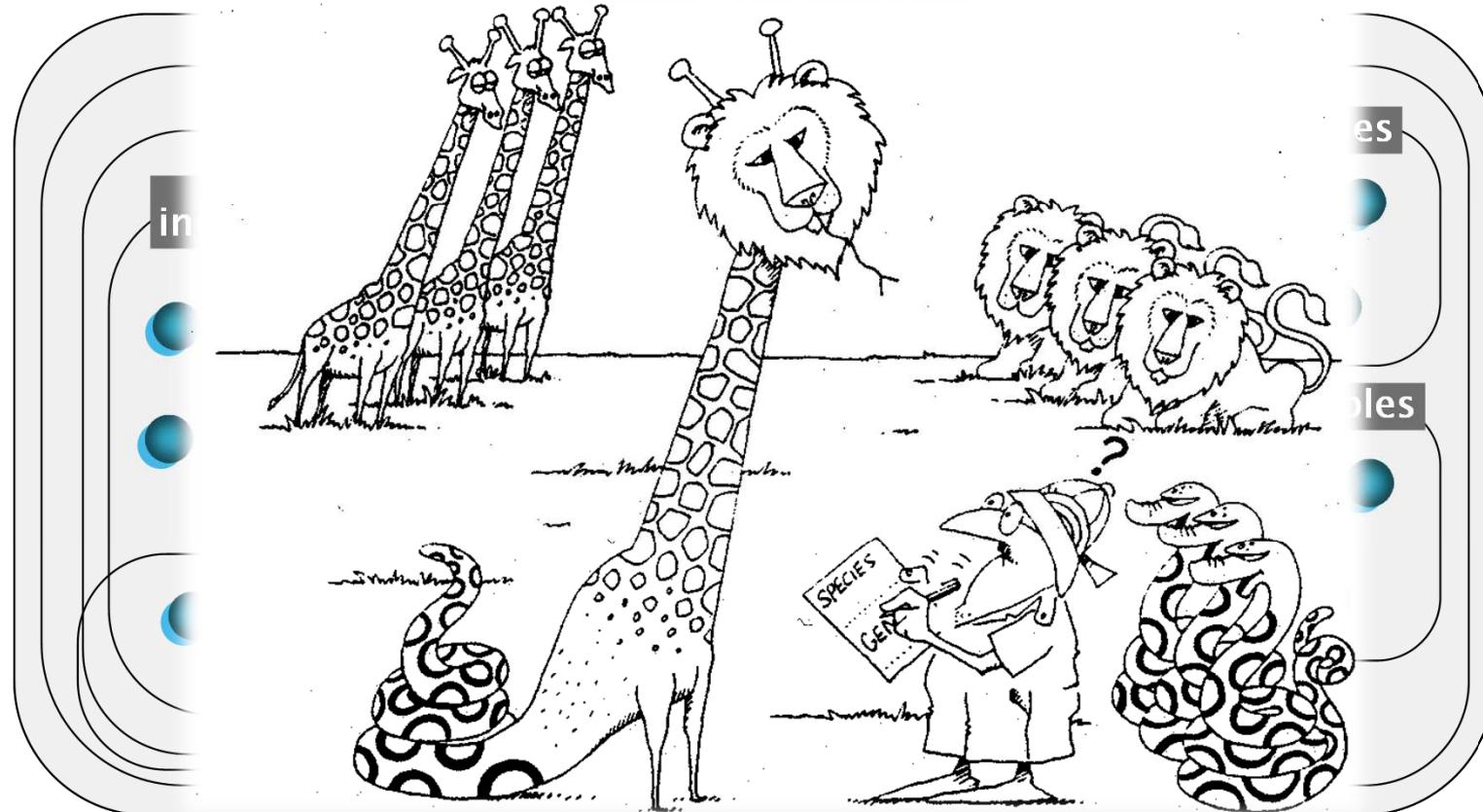
» 5. Les relations: la
Spécialisation-Généralisation

Des Entités ayant plusieurs types



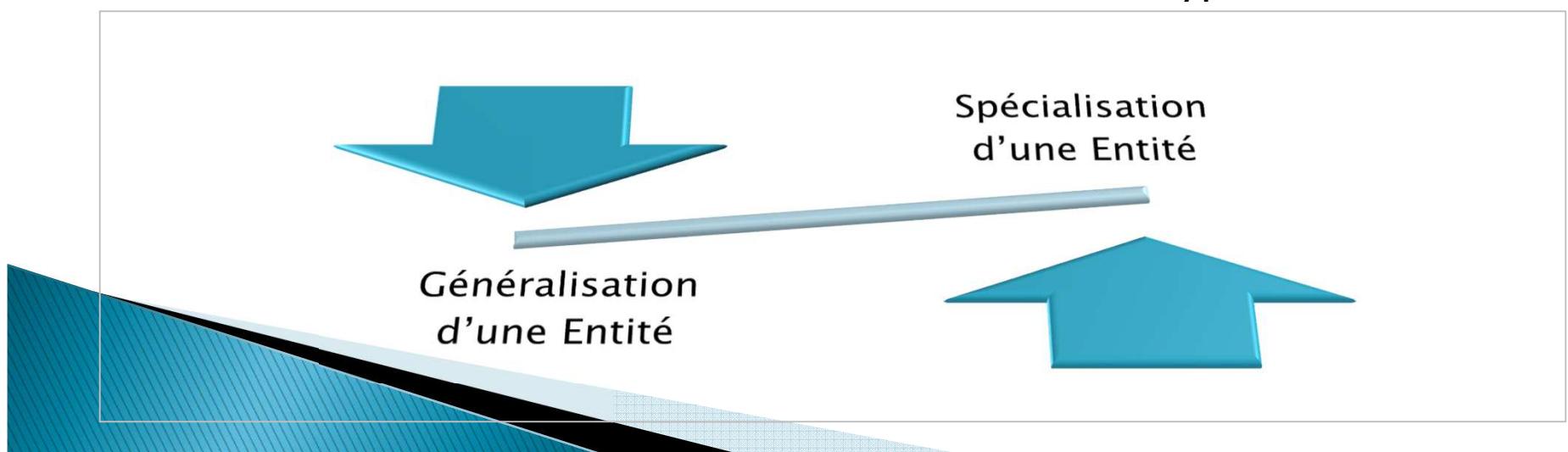
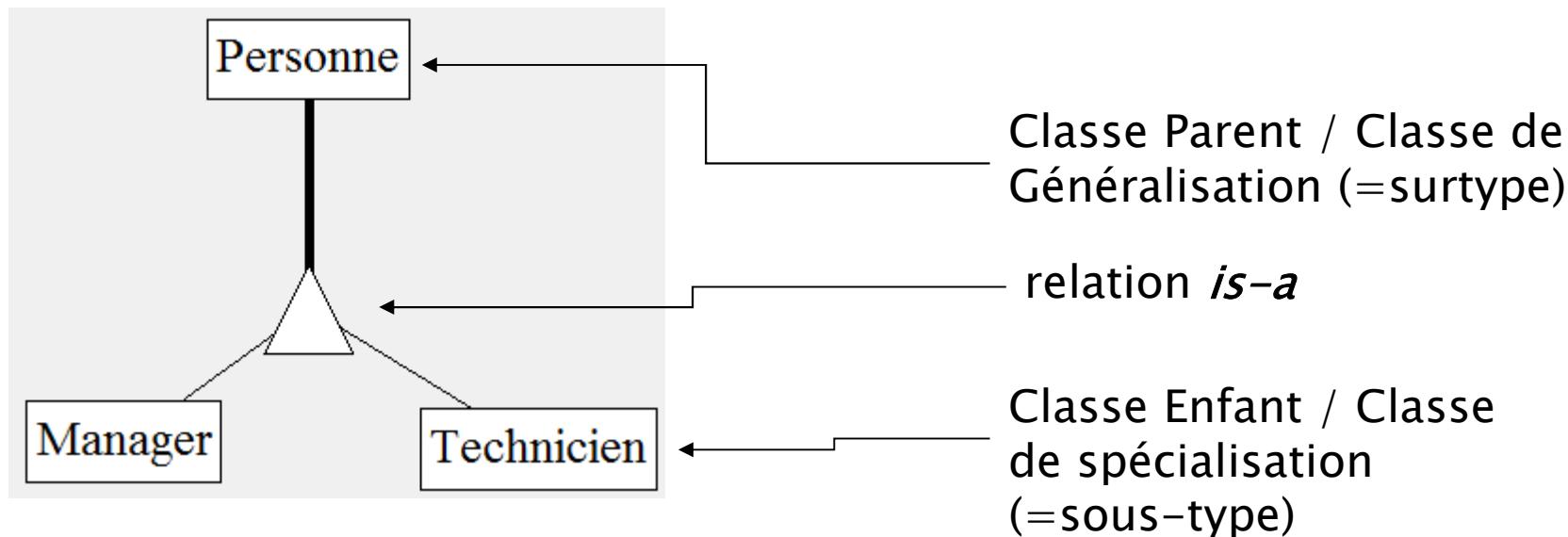
- ▶ Des entités peuvent avoir plusieurs types, c-à-d appartenir à plusieurs Classes d'Entités

Des Entités ayant plusieurs types



- ▶ Des entités peuvent avoir plusieurs types, c-à-d appartenir à plusieurs Classes d'Entités

La relation de Spécialisation–Généralisation

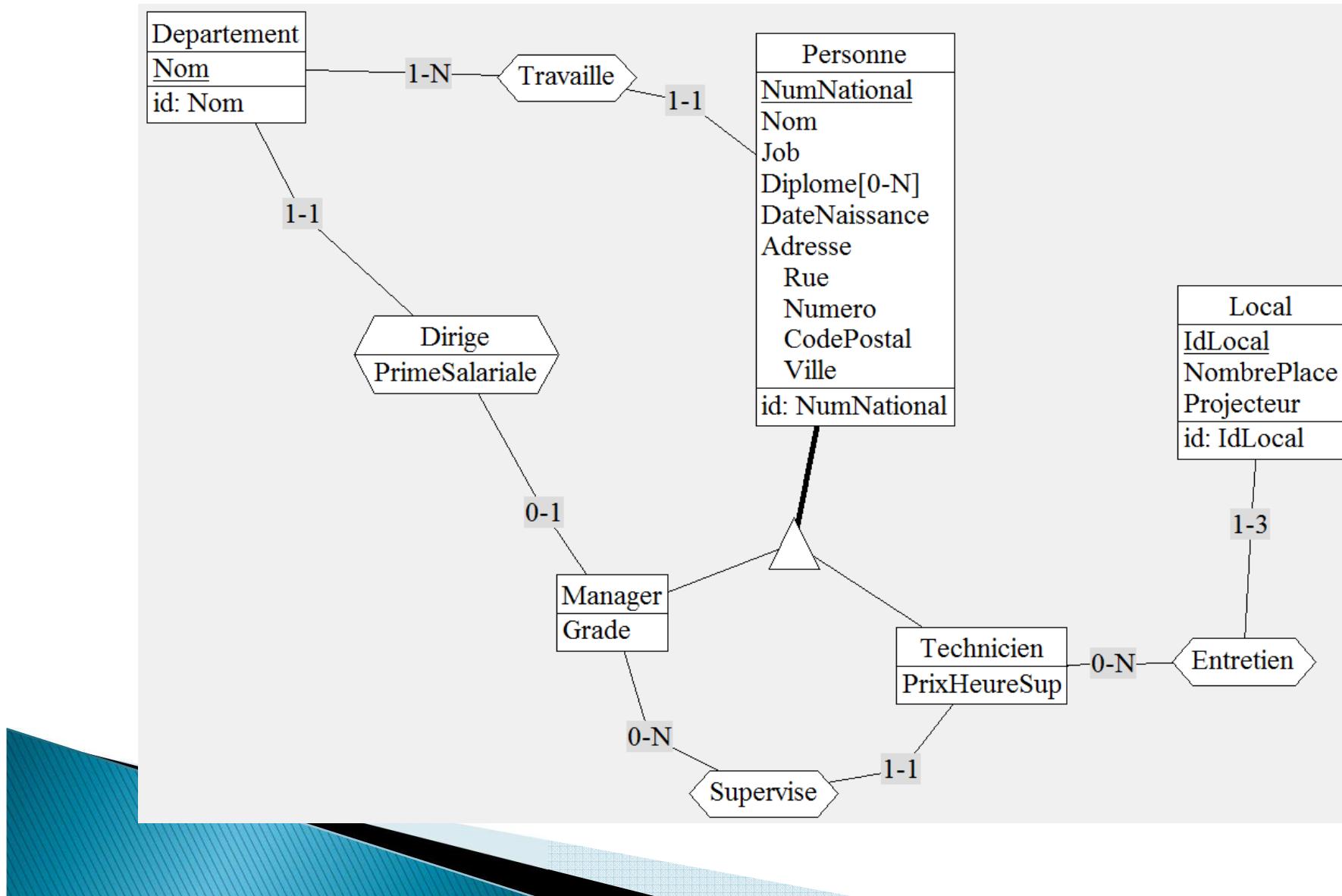


Mécanismes d'héritage dans les relations de Spécialisation – Généralisation

- ▶ Les Classes de Spécialisation héritent de toutes les caractéristiques des Classes de Généralisation
 - Les Attributs et Identifiants
 - Les Associations
- ▶ Les Classes de Spécialisations peuvent avoir des caractéristiques propres
 - Des Attributs et Identifiants
 - Des Associations
- ▶ Le mécanisme d'héritage est une conséquence de la propriété d'inclusion des populations entre les *surtypes* et les *sous-types* d'une entité particulière



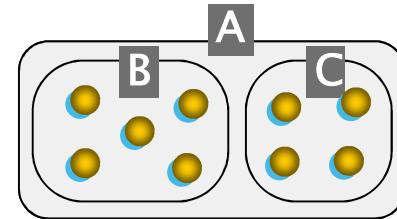
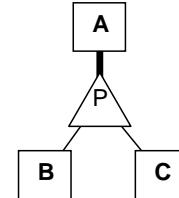
Mécanisme d'héritage: Illustration



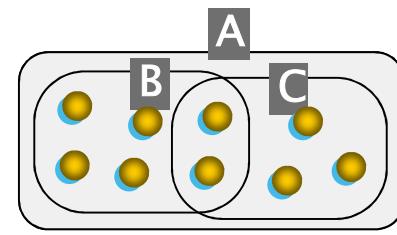
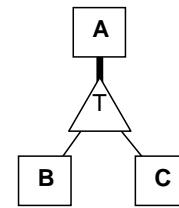
La cardinalité des relations de spécialisation – généralisations

- ▶ Une cardinalité peut être:
 - Total *ou* Partiel
 - Exclusive/Disjoint *ou* chevauchement ('overlapping')
- ▶ Total: toutes les Entités de la classe parent doivent être spécialisées
- Partiel: les entités de la classe parent peuvent être spécialisées (ou pas!)
- ▶ Exclusive: une entité spécialisée ne peut l'être qu'une seule fois
- ▶ Chevauchement: une entité spécialisée peut l'être dans plusieurs classes enfants

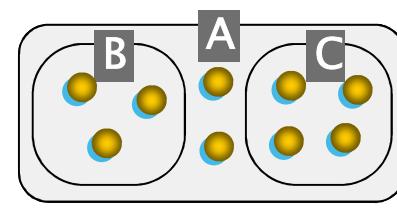
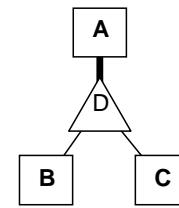
(T,E)
Partition



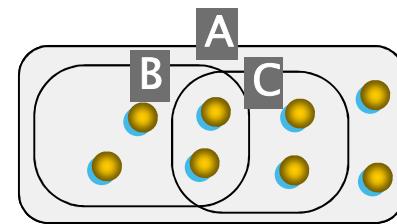
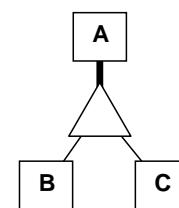
(T,O)



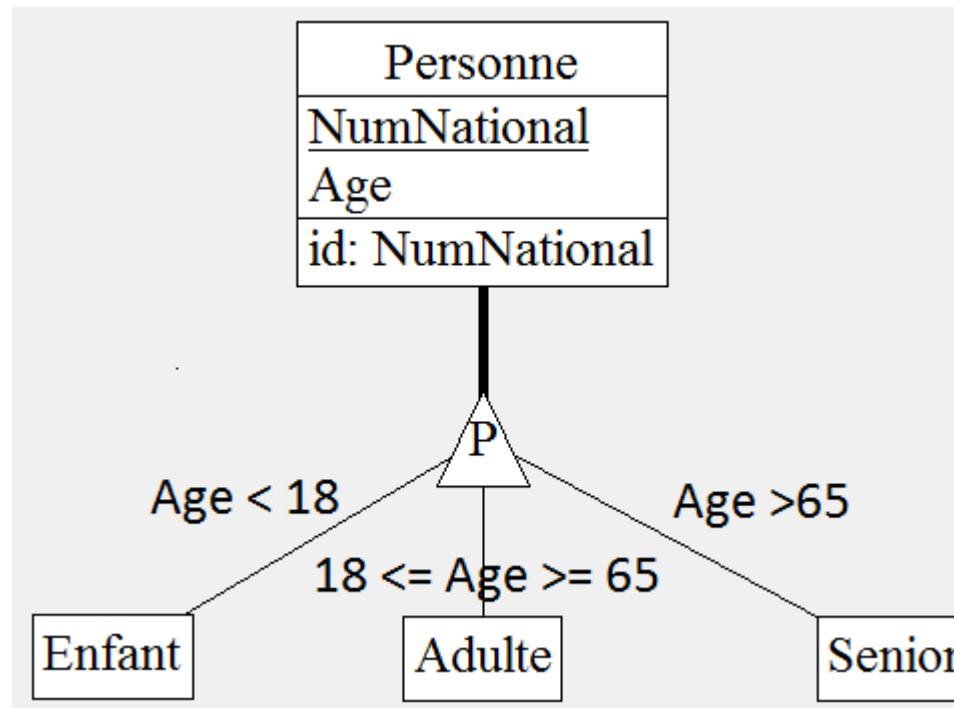
(P,E)



(P,O)

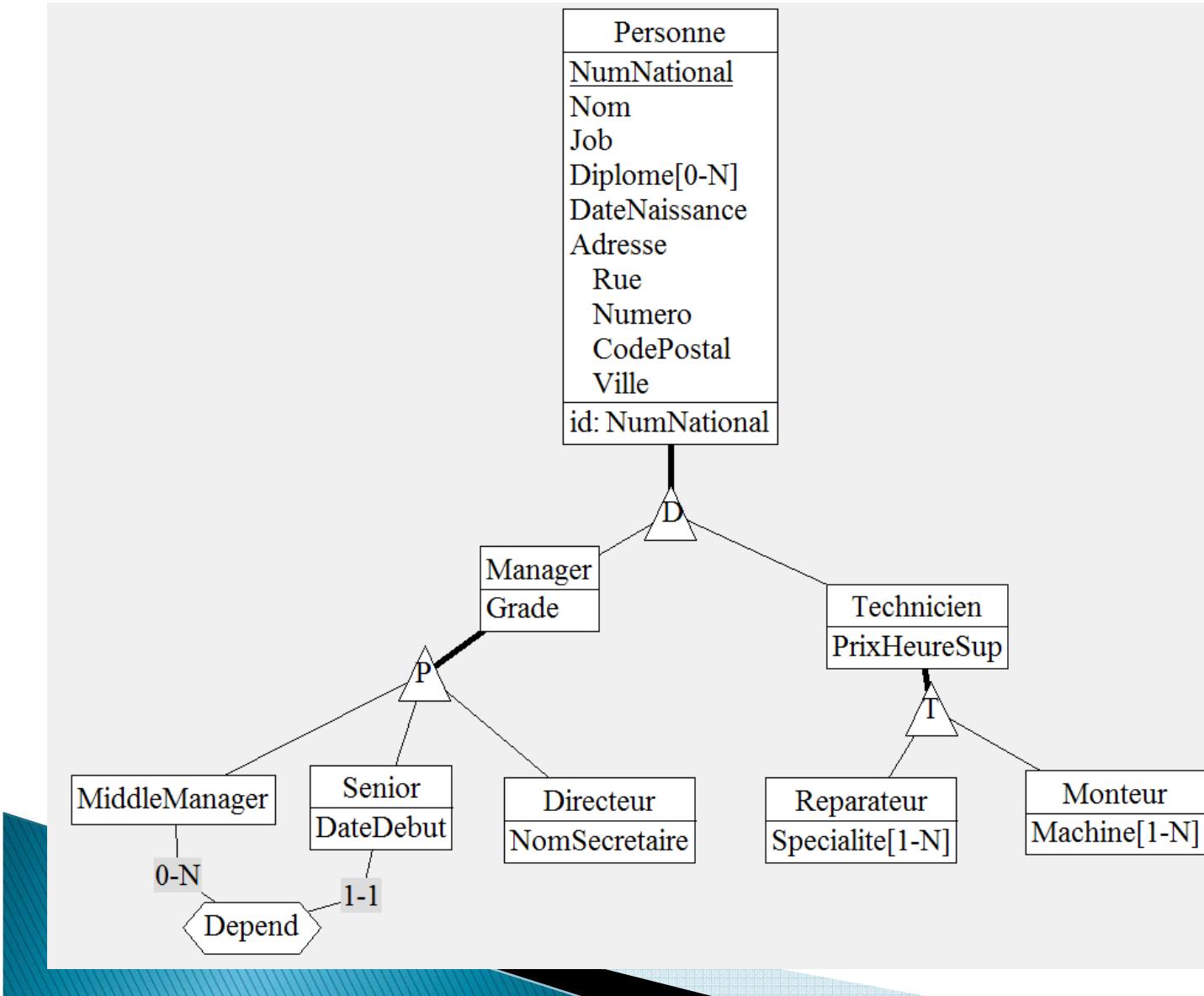


Utilisation de prédictats dans les spécialisation



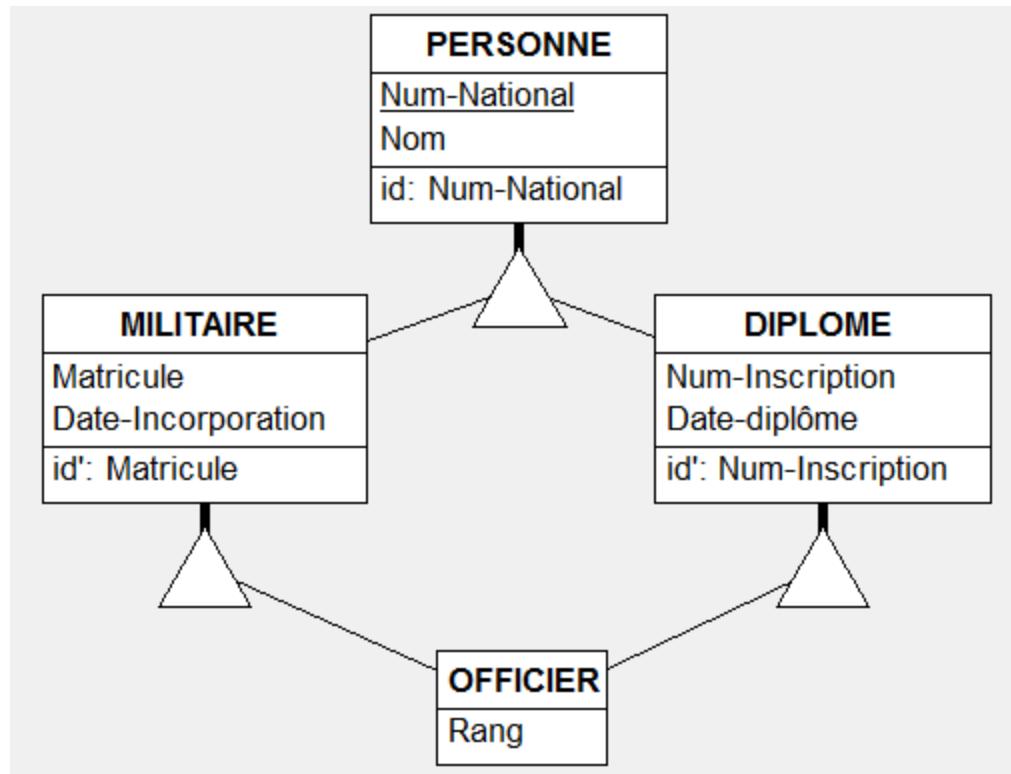
- ▶ On peut utiliser un ou plusieurs attributs de la classe parent pour spécifier la classe de spécialisation

Spécialisation multiple



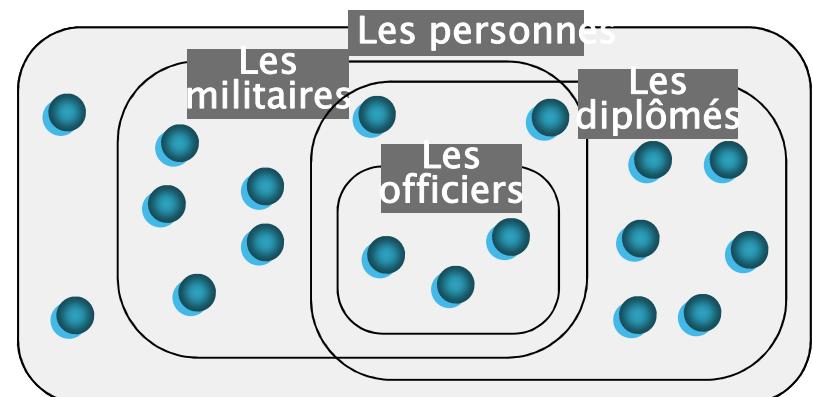
- ▶ Une classe d'entités peut à la fois être une classe de spécialisation et une classe de généralisation

Spécialisation multiple & Héritage multiple

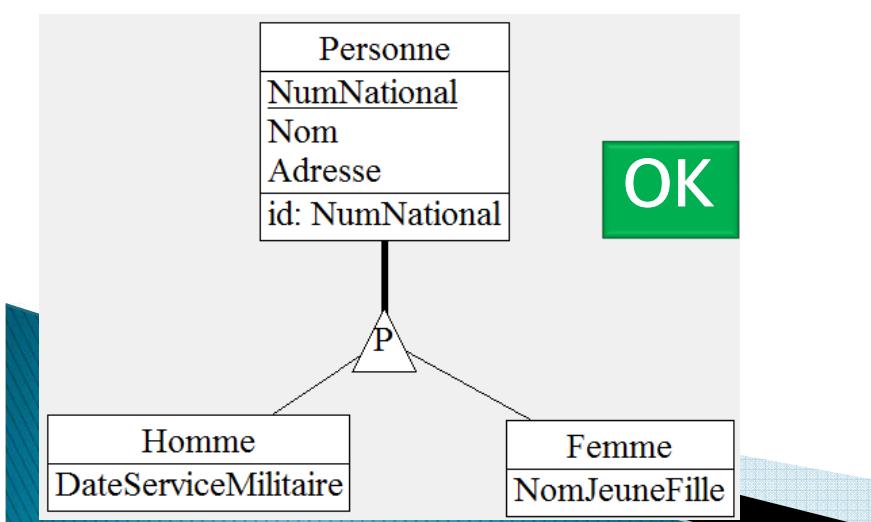
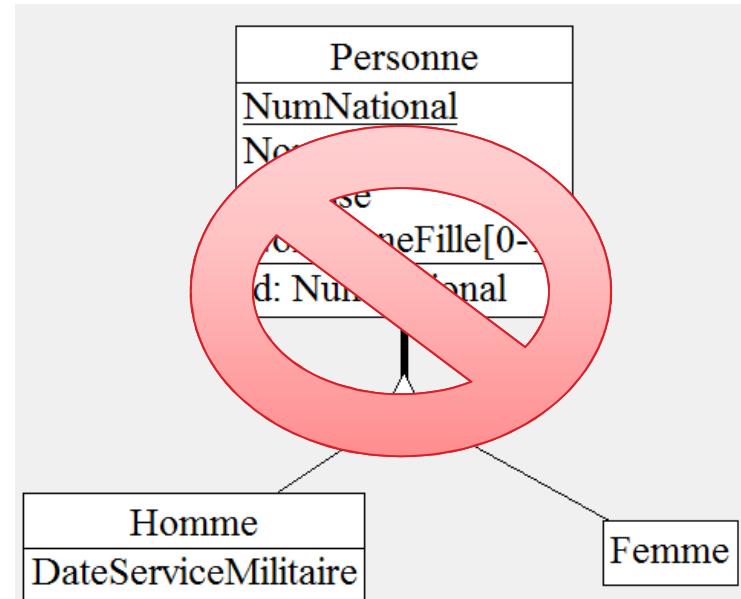
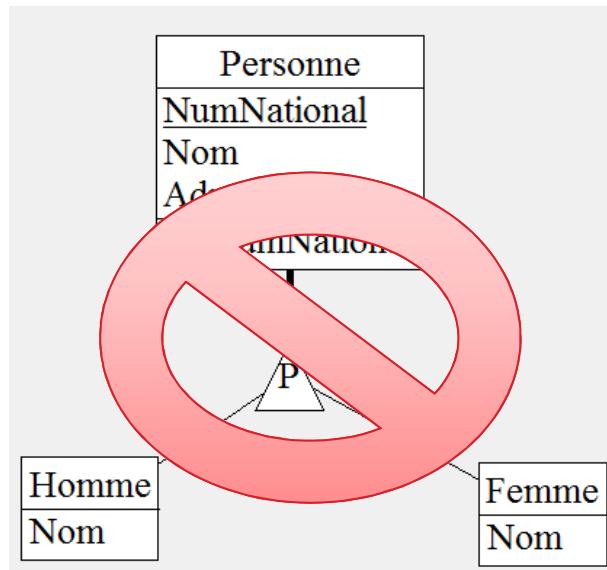


- Des militaires diplômés peuvent ne pas être des officiers

- Des officiers sont à la fois des personnes spécialisées comme militaires et des personnes classifiées comme diplômés

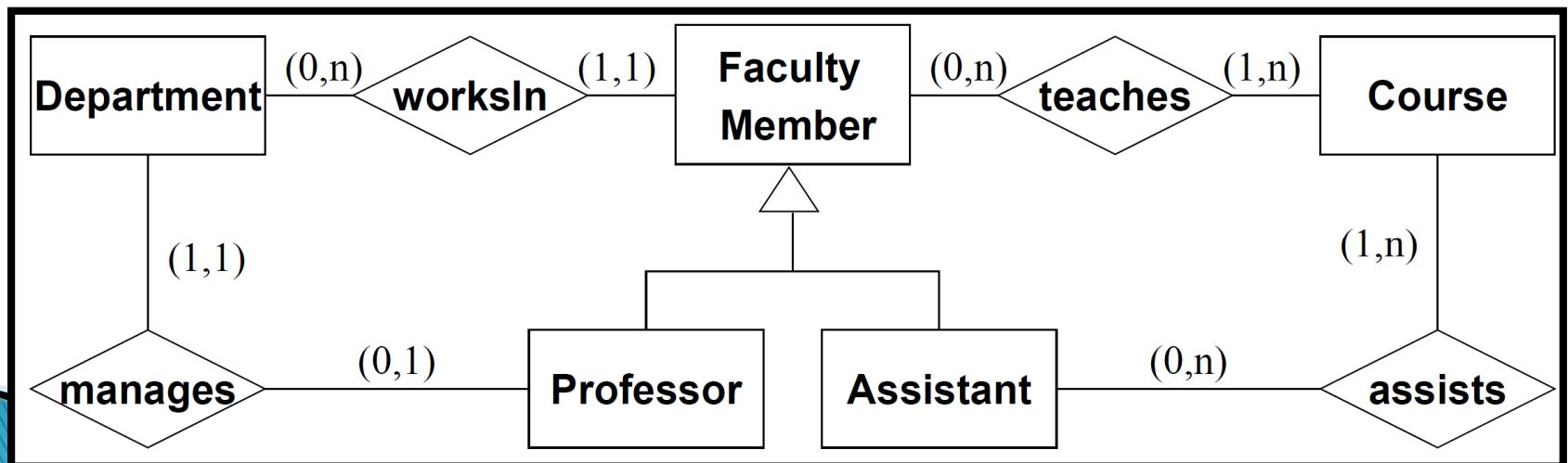
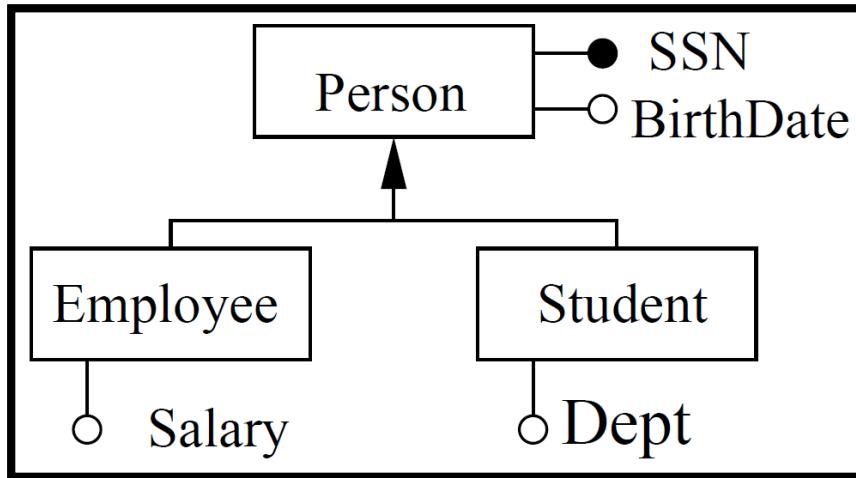


Mauvaise utilisation de la relation de Spécialisation – Généralisation

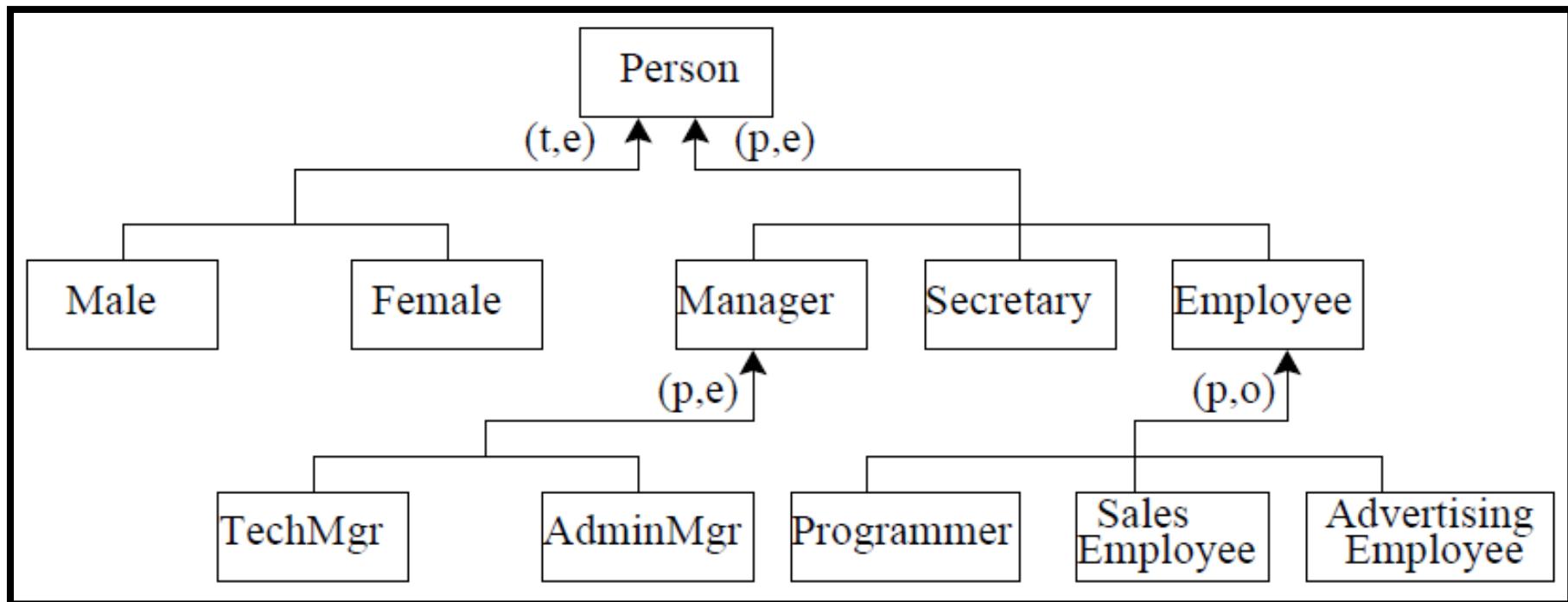


- ▶ *Attention:* ne pas confondre une relation d'Association avec une relation de Spécialisation – Généralisation!

Autres notations pour la relation de Spécialisation – Généralisation (1 / 2)



Autres notations pour la relation de Spécialisation – Généralisation (2/2)

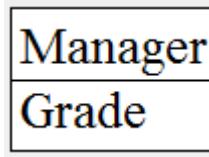


Chapitre 2: Le modèle Entité-Association

» 6. Les contraintes d'intégrités

Les contraintes d'intégrités

- ▶ Une contrainte d'intégrité est propriété d'un schéma E-A que des entités (instances) doivent respecter
 - lors l'insertion de l'entité (= contrainte statique/de structure)
 - lors de la modification et la suppression de l'entité (= contrainte dynamique)
- ▶ Une contrainte d'intégrité est valable pour tous les niveaux de schéma (conceptuel, logique et physique) même si l'outil de modélisation ne permet pas *directement* sa définition

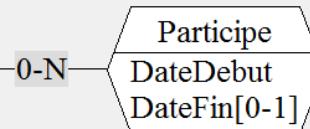


Grade = {"Low", "Middle", "High"}

Personne
NumNational
Nom
Job
Diplome[0-N]
DateNaissance
Adresse
Rue
Numero
CodePostal
Ville
id: NumNational

DateNaissance
 $\geq (1/1/1885)$

Personne
NumNational
Nom
Job
Diplome[0-N]
DateNaissance
Adresse
Rue
Numero
CodePostal
Ville
id: NumNational

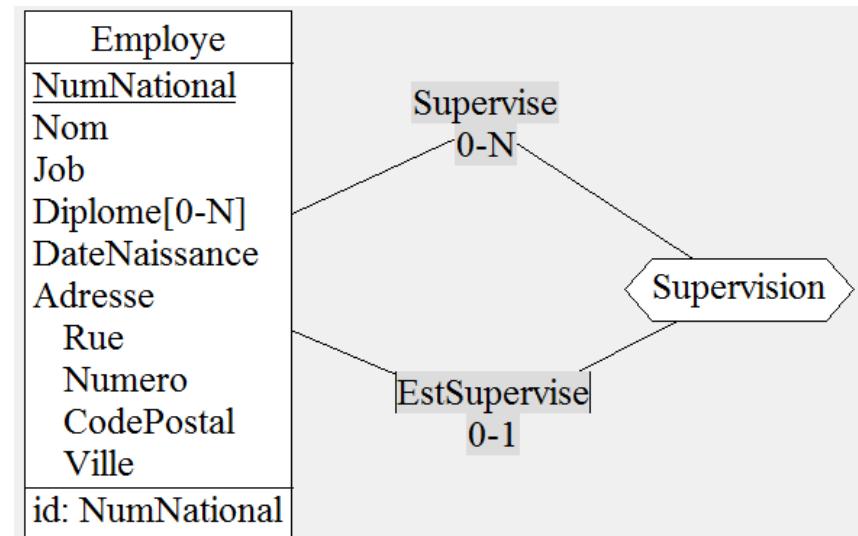
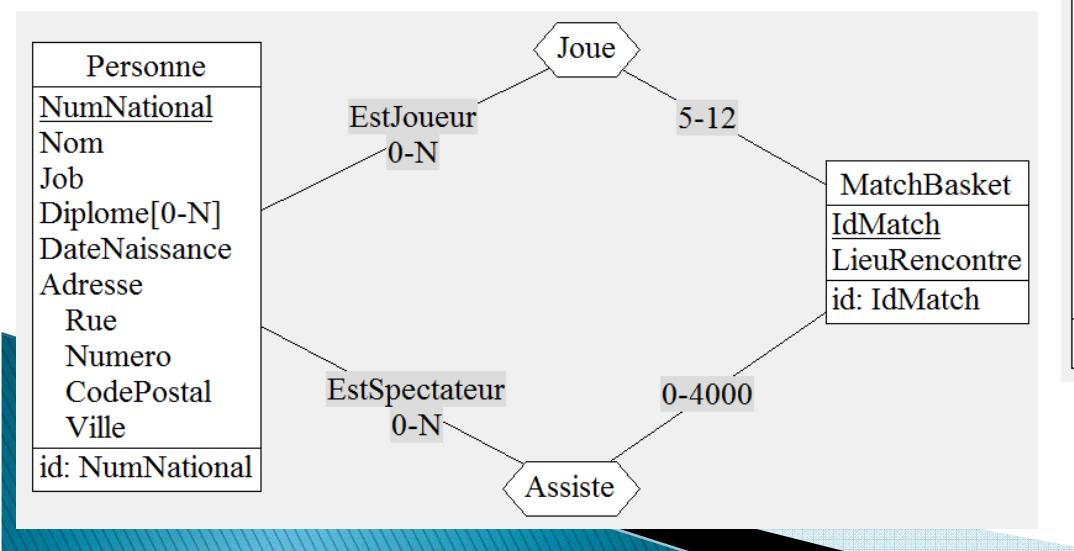


Projet
NumeroProjet
Type[1-N]
DureePrevue[0-1]
id: NumeroProjet

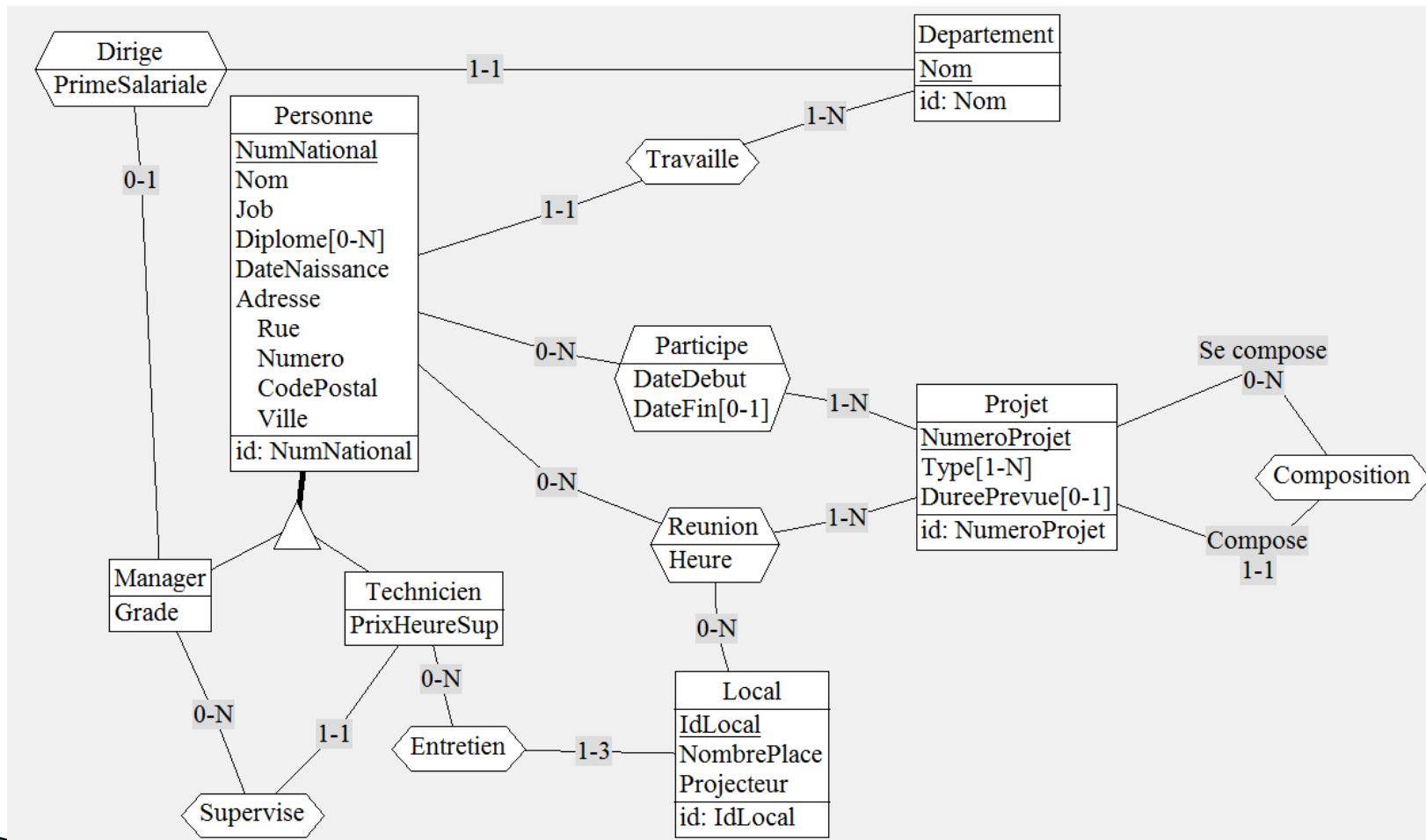
DateDebut < DateFin

Les cycles dans les Associations (Récuratives)

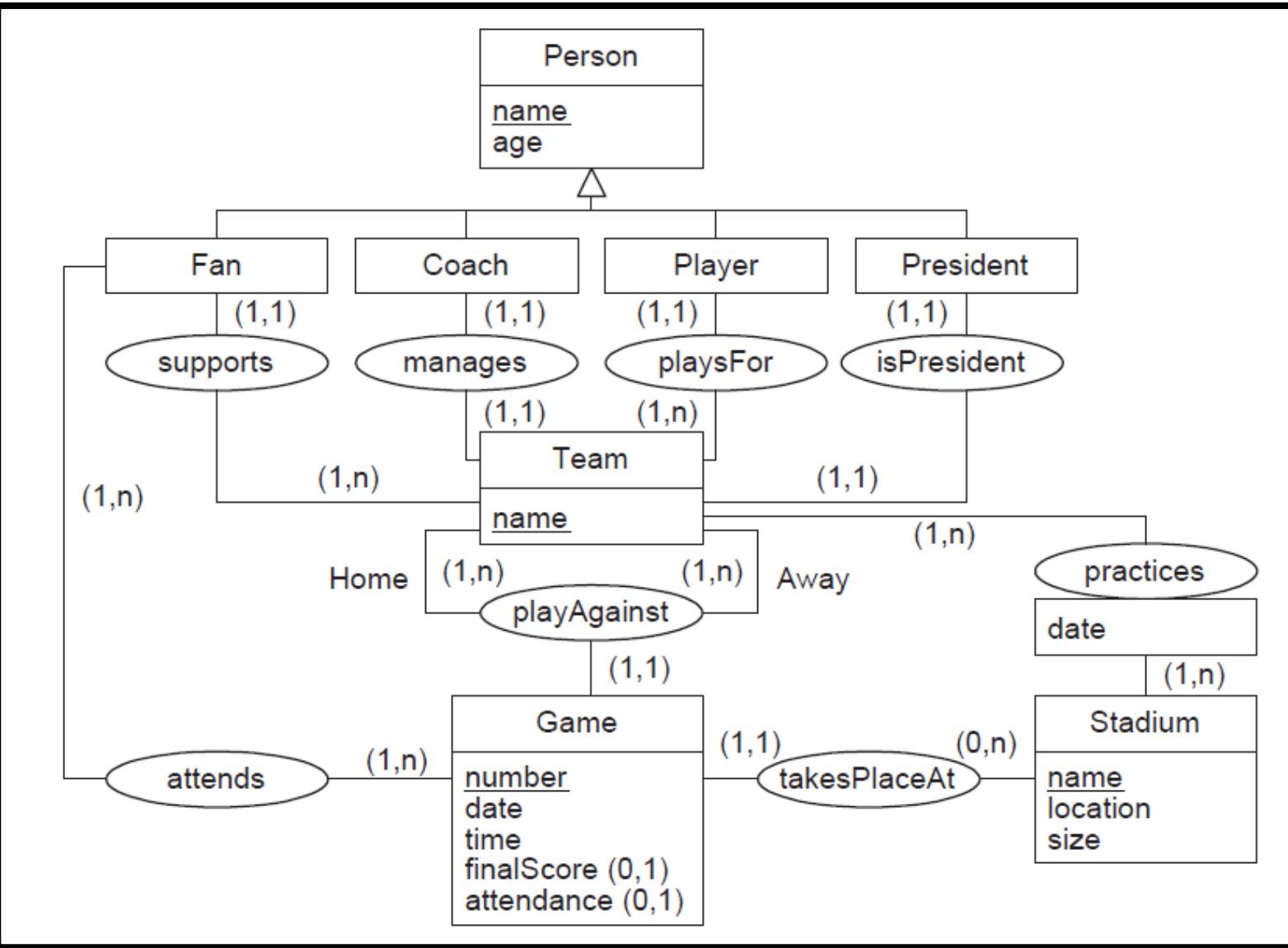
- ▶ Il faut toujours indiquer une contrainte d'intégrité dans une Association Réursive pour éviter les boucles
- ▶ On peut également retrouver des cycles dans les associations binaires modélisées sous la forme d'une boucle



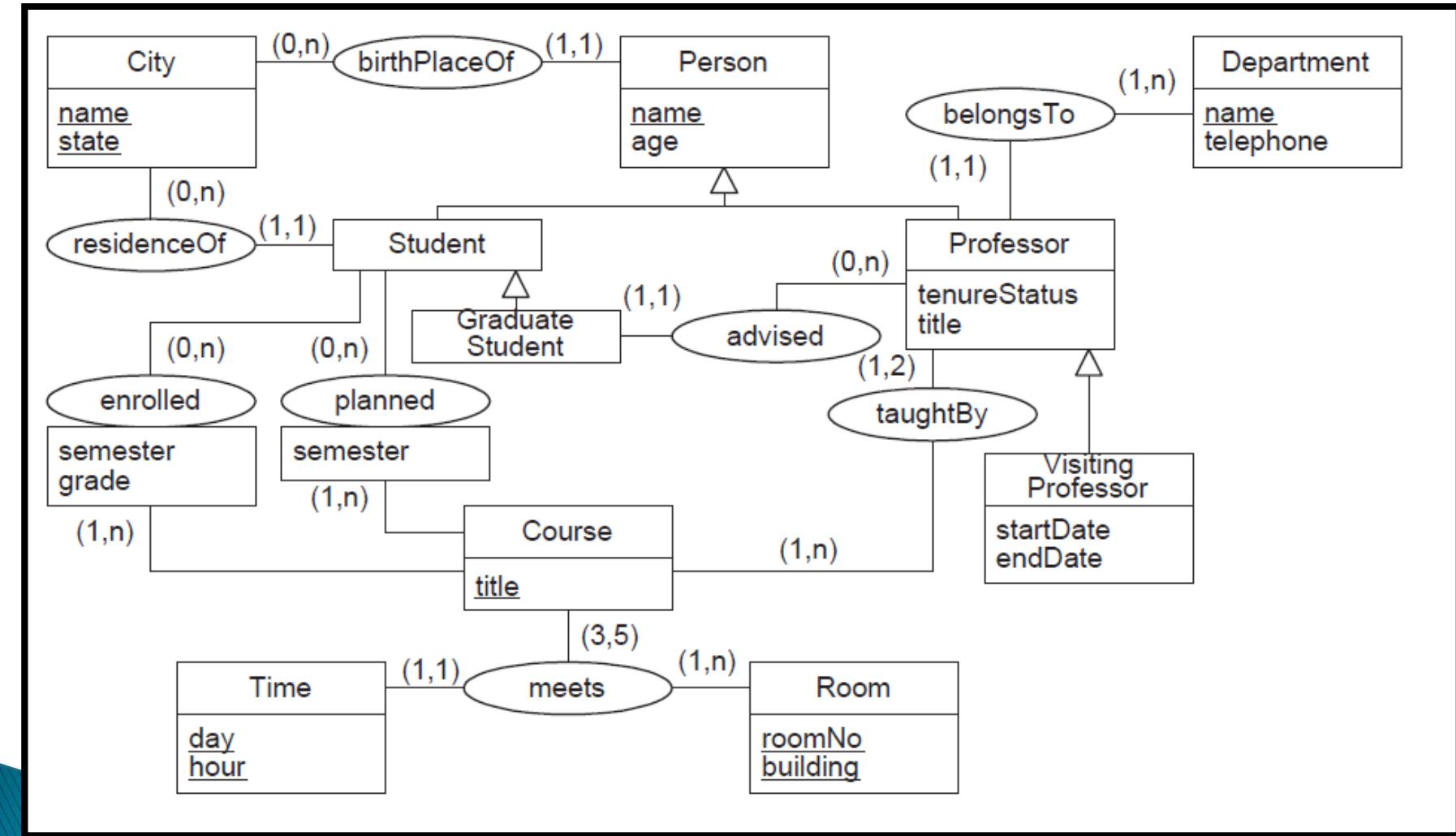
Lire un Schéma E-A (1 / 3)



Lire un Schéma E-A (2/3)



Lire un Schéma E-A (3 / 3)



Exercices



- ▶ Quelques exercices pour approfondir les connaissances du modèle Entité–Association



Table des matières



- ▶ Chapitre 1: Introduction aux bases de données
- ▶ Chapitre 2: Le modèle Entité–Association
- ▶ Chapitre 3: Le Schéma Relationnel
 - 1. Le Schéma Relationnel: définition et concept
 - 2. Les types de contraintes
 - 3. Les règles de traductions de l'E-A vers le schéma relationnel
- ▶ (Chapitre 4: SQL)

Chapitre 3: Le Schéma Relationnel

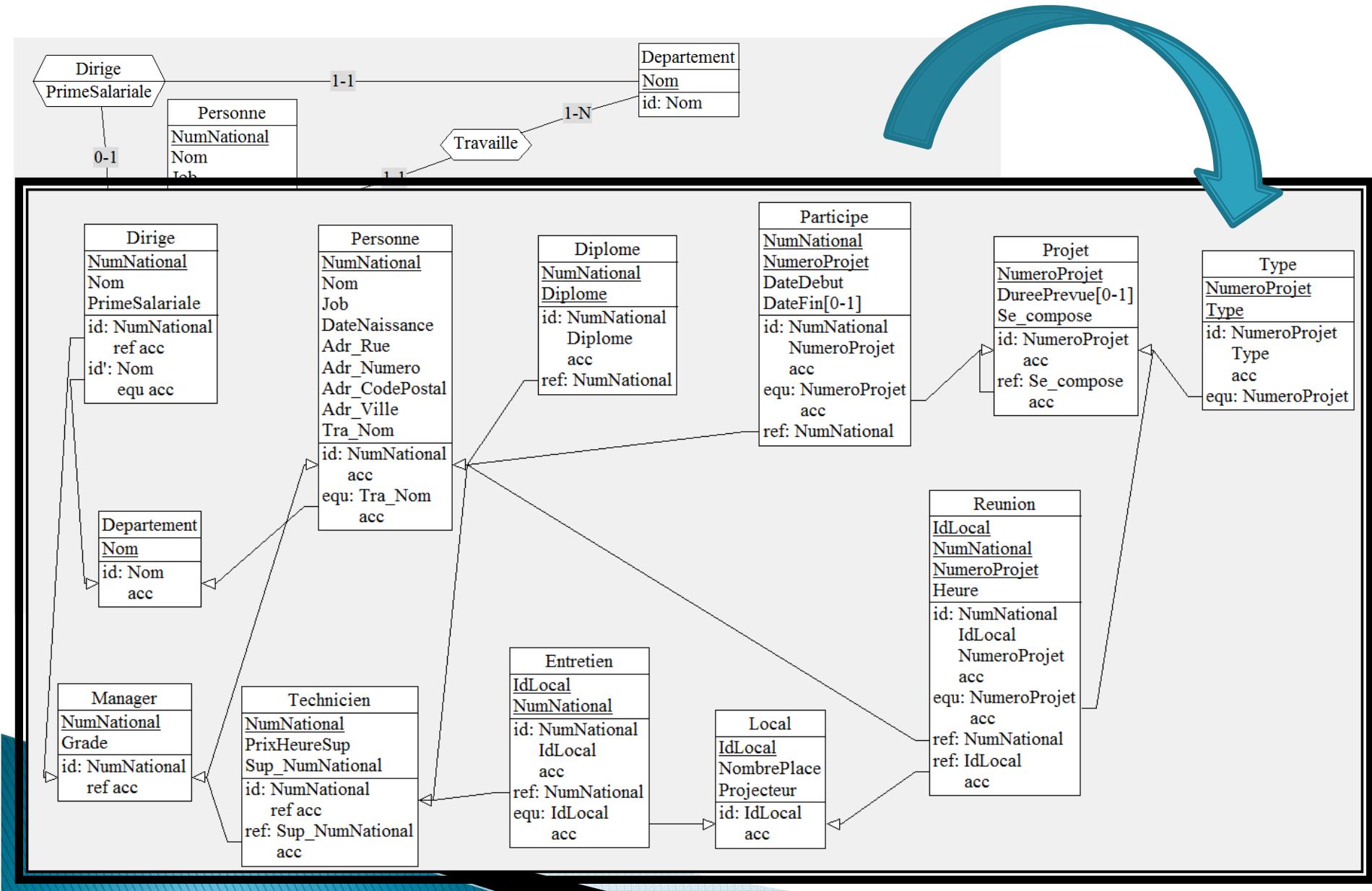
» 1. Le Schéma Relationnel:
définition et concept

Le Schéma relationnel: Définition

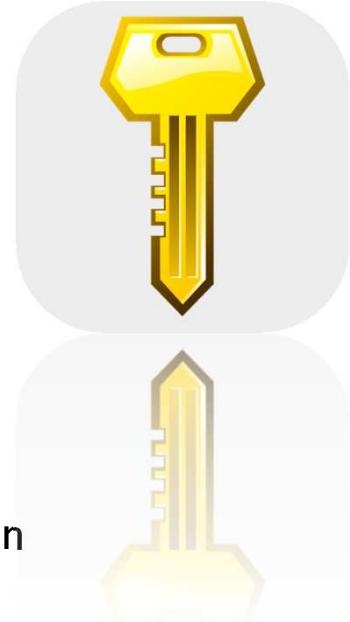
- ▶ Une base de données relationnelle = ensemble de relations de 2 dimensions
 - Un enregistrement dans une table
 - Un référence entre deux enregistrements via la redondance d'information
=> Mécanisme de clé primaire – clé étrangère
- ▶ Le schéma relationnel définit la manière dont on pourra utiliser la base de données par après
 - SQL DDL (contraintes spécifiques)
 - SQL DML
 - SQL DRL



Exemple: E-A => schéma relationnel



Le mécanisme clé primaire - clé étrangère



- ▶ Une **clé primaire** identifie de manière unique tout enregistrement d'une table par rapport aux autres enregistrements
- ▶ Deux propriétés:
 - Aucune valeur NULL acceptée
 - Tout nouvel enregistrement (ou modification de valeur d'un ancien enregistrement) doit avoir une valeur unique pour sa clé primaire
- ▶ Une clé primaire peut être composé de plusieurs colonnes
- ▶ La **clé maximale** = l'ensemble des colonnes d'une table (on ne peut pas enregistrer deux fois le même objet!)
- ▶ La **clé minimale** = clé primaire
- ▶ La **clé étrangère** traduit une relation (Association ou Spécialisation) avec une autre table en référençant la clé primaire de cette table
- ▶ Contrainte d'intégrité référentielle:
 - La clé étrangère doit avoir le même domaine que la clé primaire qu'elle référence
 - Chaque valeur enregistrée dans la clé étrangère doit être une valeur existante de la clé primaire
- ▶ Clé étrangère = clé secondaire (autre appellation)



Chapitre 3: Le Schéma Relationnel

» 2. Les types de contraintes
d'intégrité

Les types de contraintes d'intégrités

- ▶ **Contrainte d'intégrité d'une clé primaire:** toute clé primaire ne peut être NULL et doit avoir une valeur distincte des valeurs existantes
- ▶ **Contrainte d'intégrité référentielle:** mécanisme de liens à l'aide d'une clé primaire et d'une clé étrangère donnant un domaine et une liste de valeurs identiques entre le clé étrangère et sa clé primaire (ex: le type d'une colonne)
- ▶ **Contrainte d'intégrité de domaine:** contrainte définissant l'ensemble des valeurs que les champs d'une colonne peuvent prendre
- ▶ **Contrainte de ligne:** contrainte vérifiée lors de l'insertion/modification d'un enregistrement
- ▶ **Contrainte de colonne:** contrainte vérifiée pour toute valeur d'une colonne (à l'insertion et/ou à la modification)

Violation de contraintes

- ▶ Si une insertion/modification viole une contrainte, soit:
 1. Rejet de l'insertion
 2. Correction de l'insertion (l'élément de l'enregistrement violant la contrainte est adapté automatiquement)
- ▶ Si une suppression viole la contrainte d'intégrité référentielle, soit:
 1. La suppression est interdite
 2. On indique une valeur NULL comme valeur dans les clés étrangères
 3. La suppression est réalisée en cascade: tous les enregistrements référençant la clé primaire supprimée sont eux aussi supprimés!!

Chapitre 3: Le Schéma Relationnel

» 3. Les règles de traductions de
l'E-A vers le schéma relationnel

**La liste suivante énumérant les règles de transformation n'est pas exhaustive!
Les règles les plus simples sont toujours privilégiées si plusieurs existent**

Traduire une classe d'entité

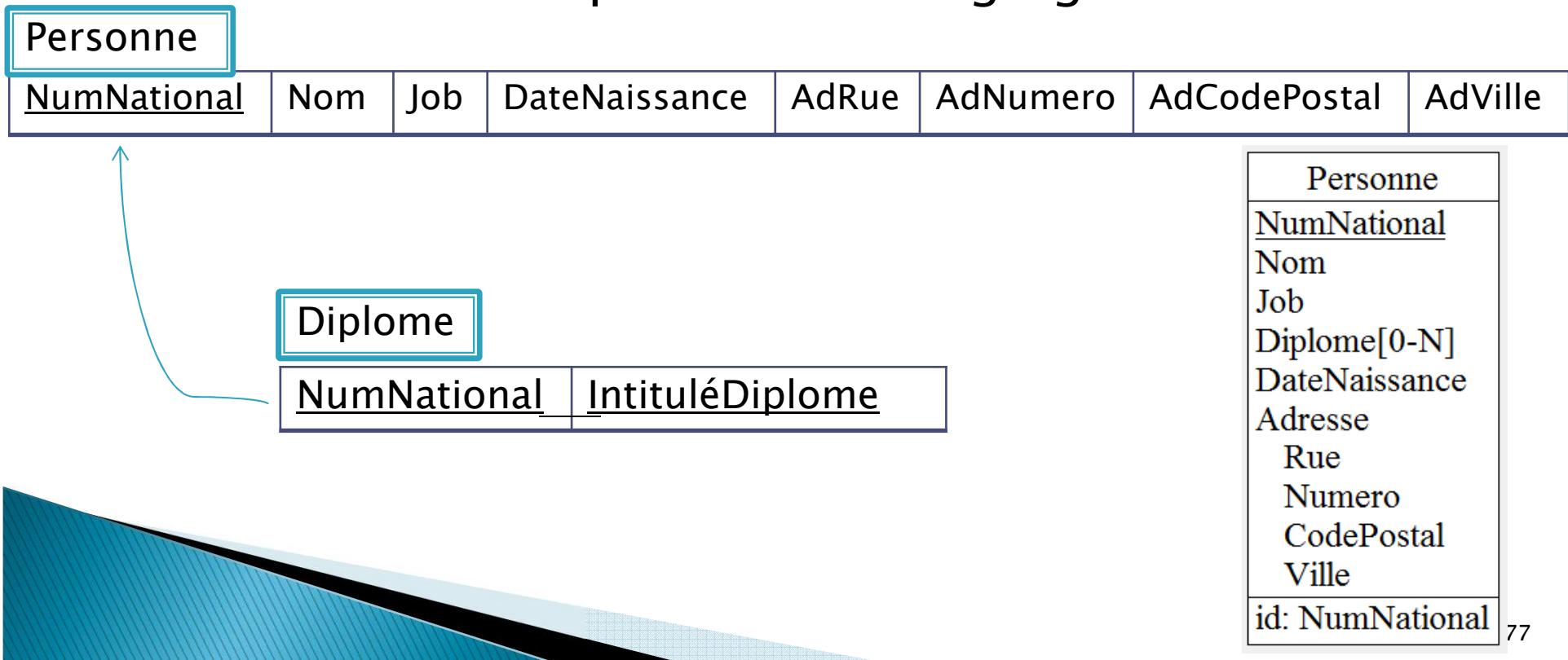
- ▶ Toute classe d'entités devient une table dans laquelle les attributs deviennent les colonnes de la table.
- ▶ L'(es) identifiant(s) de la classe d'entités devient(nent) la clé primaire de la table

Local
<u>IdLocal</u>
NombrePlace
Projecteur
id: IdLocal

Local		
<u>IdLocal</u>	NombrePlace	Projecteur

Traduire un attribut multiple ou composite

- ▶ Un attribut multiple est transféré dans une nouvelle table et référencé à l'aide du mécanisme de clé primaire – clé étrangère
- ▶ Une attribut composite est désagrégé



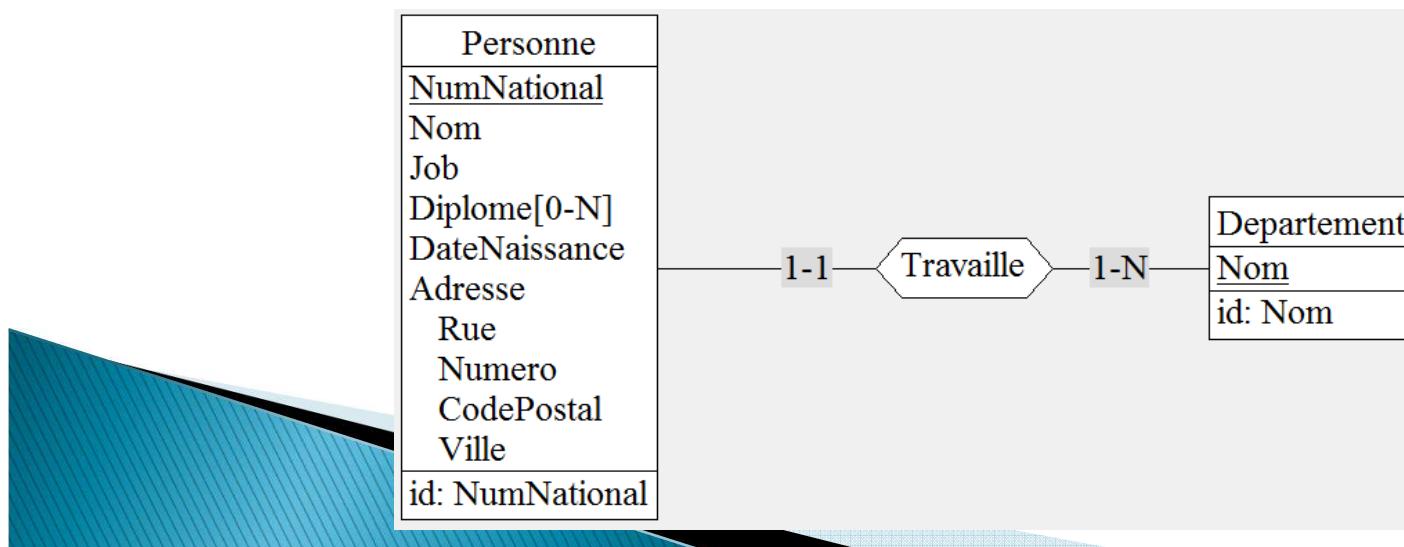
Traduire une association

- ▶ Garder les deux règles primordiales en tête :
 - Éviter la redondance d'information
 - S'arranger pour que le modèle final comporte le moins de valeurs nulles possibles. Cela doit être décidé au cas par cas, en fonction du problème posé
- ▶ De manière générale :
 - Deux cas possibles :
 - apparition d'une clé étrangère dans l'une des tables de l'association (celle qui a la cardinalité (1,1) ou (0,1). Cette clé étrangère pointera toujours vers la clé primaire de l'autre table
 - Création d'une table intermédiaire constituée de 2 clés étrangères, chacune pointant vers une table différente de l'association
 - Une table dont la cardinalité est (1,N) dans une association ne reçoit jamais de clé étrangère
 - Les attributs des associations suivent l'apparition de la clé étrangère dans la table concernée



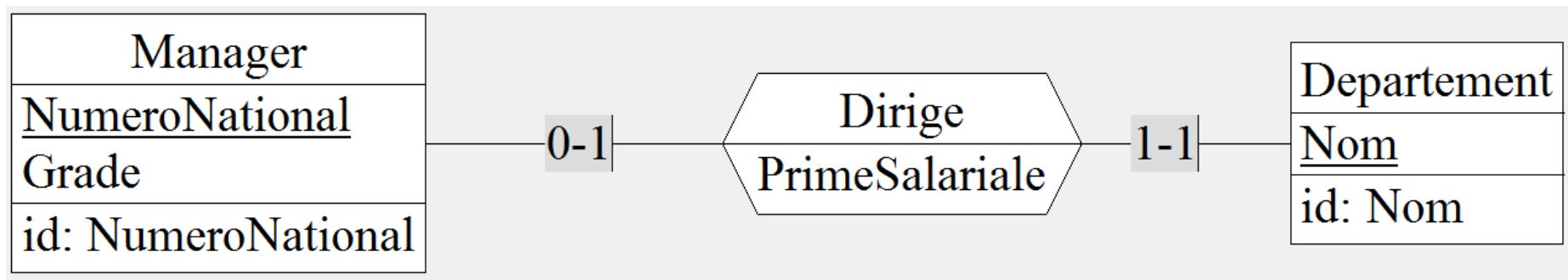
Traduire une Association binaire One-to-Many

- ▶ Une association binaire de type one-to-many disparaît au profit d'une clé étrangère dans la table du côté de la cardinalité (0,1) ou (1,1). Cette clé étrangère référence la clé primaire de la table ayant une cardinalité (0,n) ou (1,n)
- ▶ Les éventuels attributs de l'association deviennent des colonnes de la table du côté de la cardinalité (0,1) ou (1,1)
- ▶ La clé étrangère ne peut recevoir une valeur NULL que si la cardinalité était optionnelle, c-à-d : (0,1)



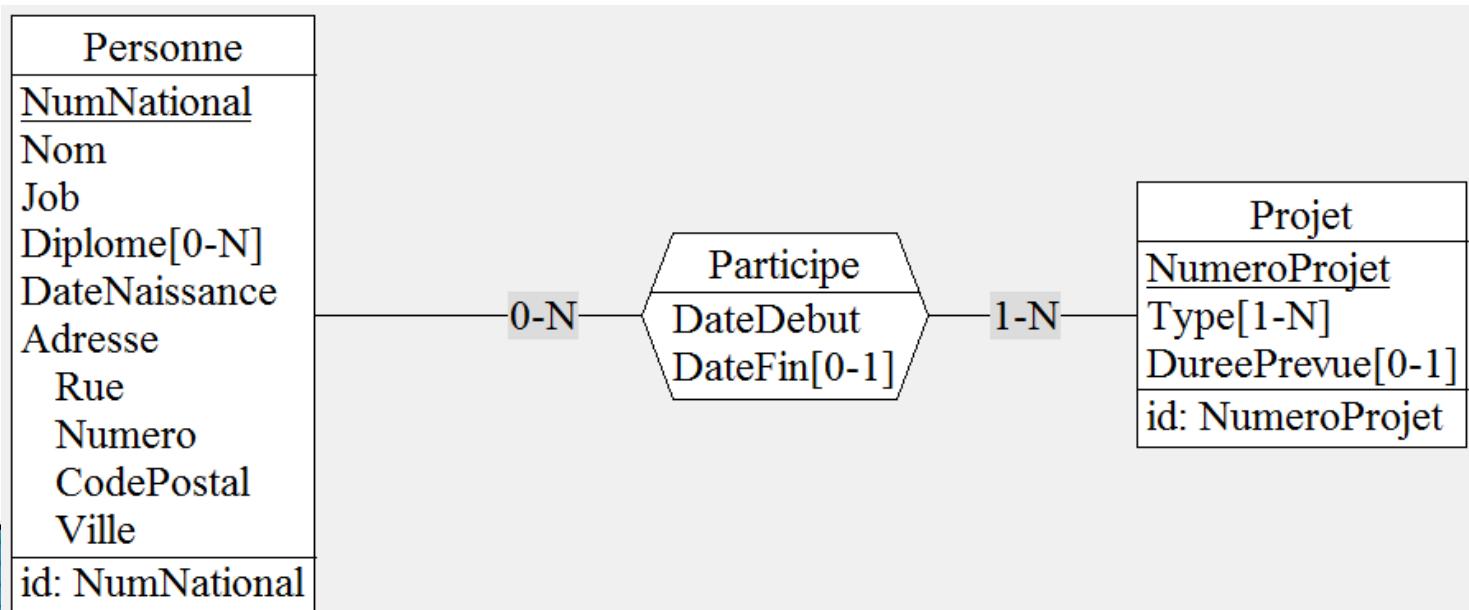
Traduire une Association binaire One-to-One

- ▶ Une association binaire one-to-one est traduite comme une association de type one-to-many
- ▶ On priviléie toujours la table du côté de la cardinalité obligatoire, c-à-d que la clé étrangère est ajouté du côté de la cardinalité obligatoire



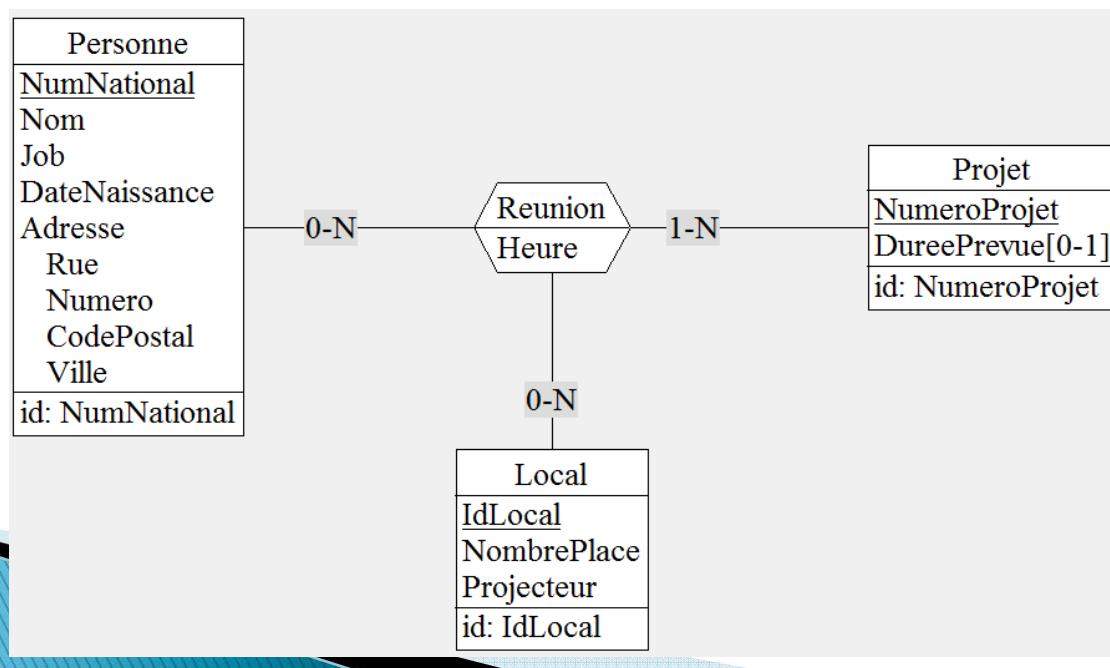
Traduire une Association binaire Many-to-Many

- ▶ Une association binaire de type many-to-many disparaît au profit d'une table supplémentaire dont la clé primaire est formée des deux clés étrangères
- ▶ Les attributs éventuels de l'association deviennent des colonnes dans la nouvelle table



Traduire une Association n-aire

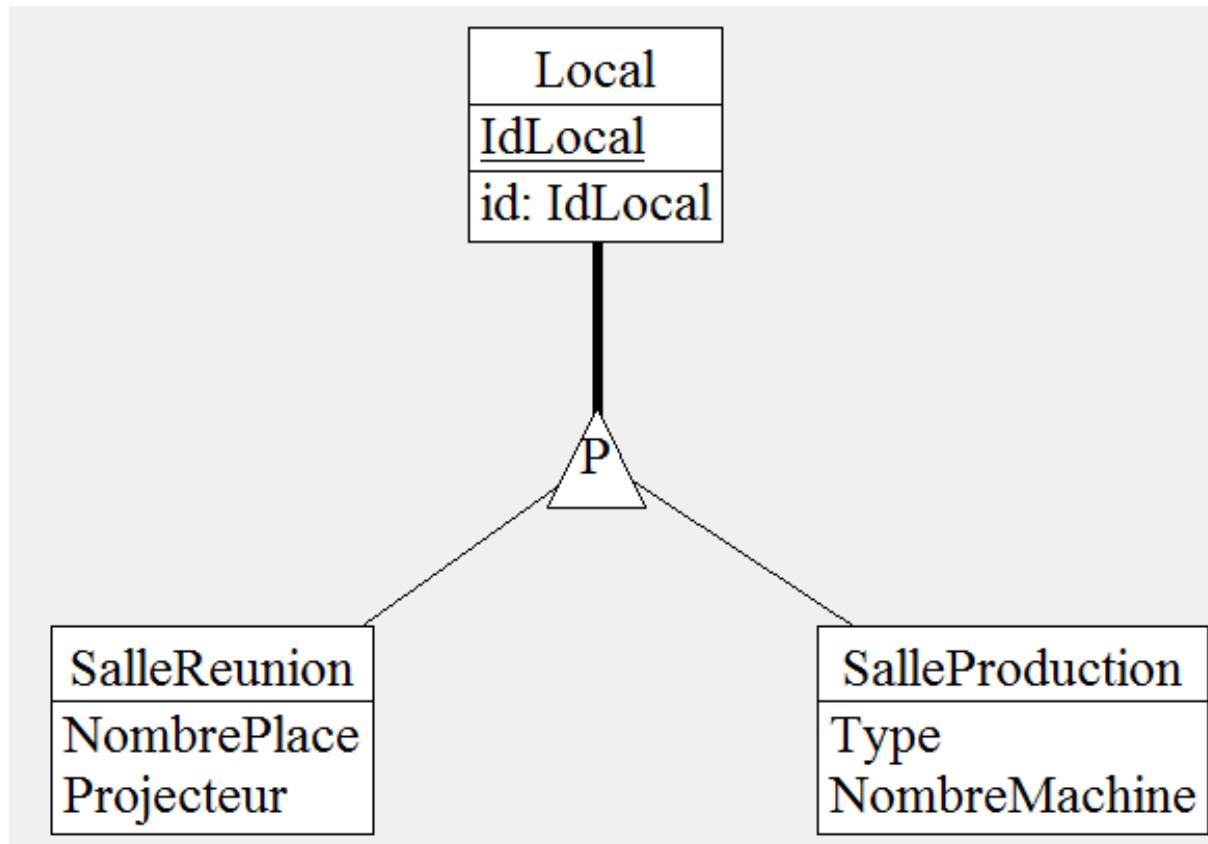
- ▶ Une association non-binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que de classes d'entités en association
- ▶ Les attributs éventuels de l'association deviennent les colonnes de cette nouvelle table



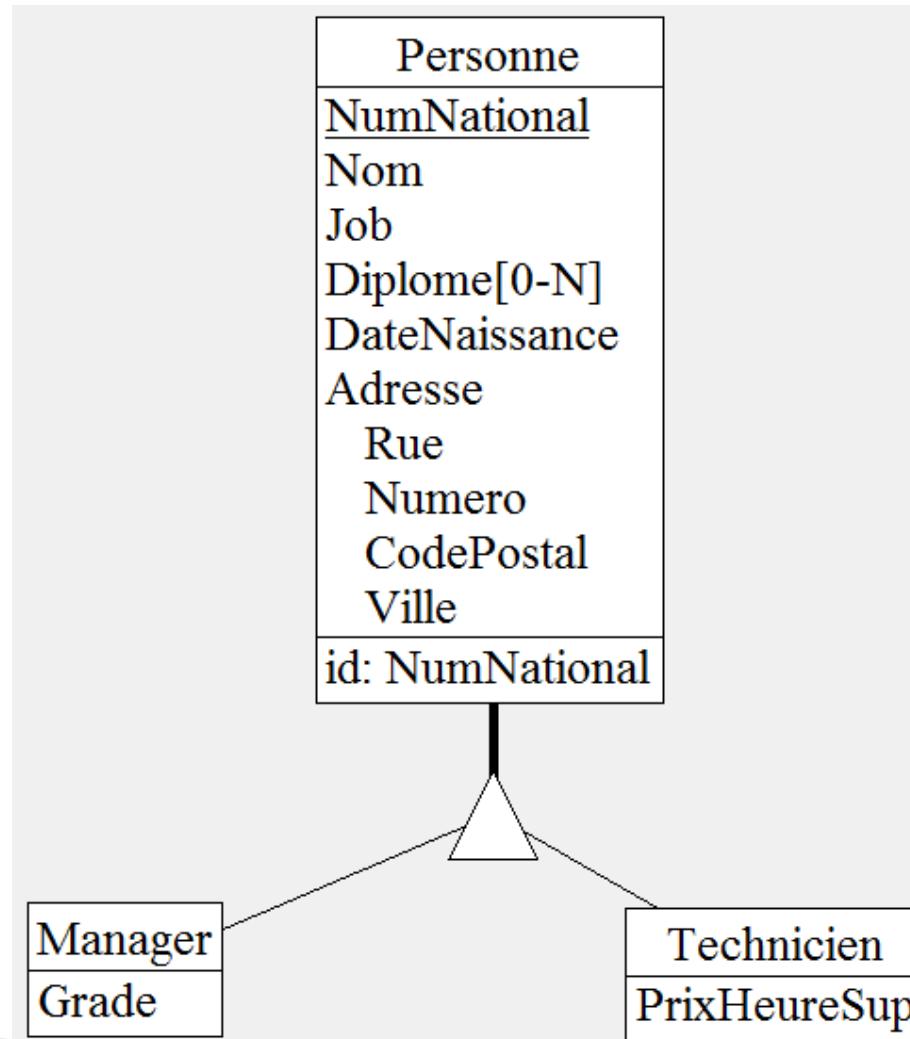
Traduire une relation de spécialisation – Généralisation

- ▶ Les relations de spécialisation – généralisation doivent être désolidarisées (tout en gardant une relation dans les tables)
 - Si la cardinalité est Totale: conservation des classes enfants et suppression de la classe parent
Les attributs/associations de la classe parent sont transmis à chacune des classes enfants
 - Si la cardinalité est Partielle:
 - Conservation de la classe parent
 - OU, création d'une nouvelle classe enfant "autre"
- ▶ *Il faut toujours garder à l'esprit qu'il faut toujours permettre la même expressivité après la transformation*

Traduire une spécialisation Totale



Traduire une spécialisation Partielle



Exercices



- ▶ Quelques exercices pour utiliser les règles de traduction de l'Entité-Association vers le Schéma Relationnel

