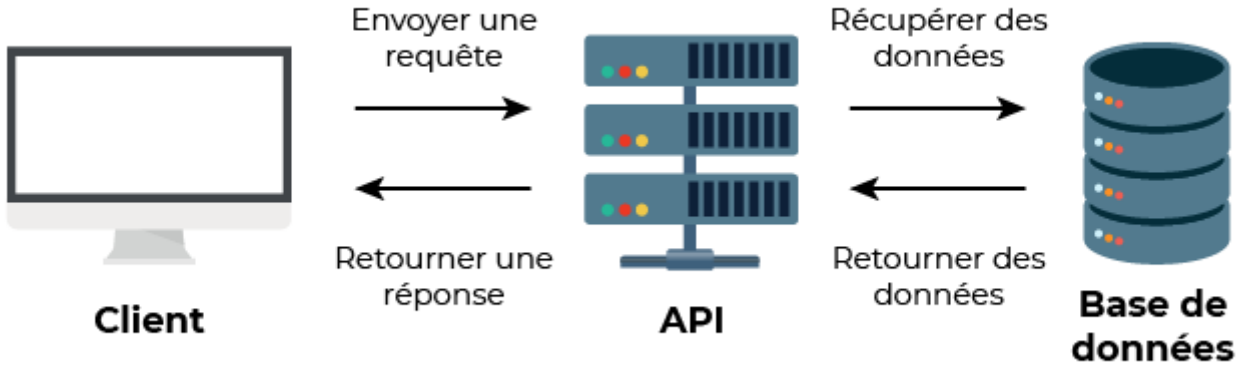
Introduction aux
Bases de Données

Fonctionnement API



Angular (TypeScript)

Spring (Java)

SQL (MySql)

Définition d'une DB

Qu'est-ce qu'une base de données ?

Temps de lecture : 5 mn

Une base de données est un ensemble d'informations qui est organisé de manière à être facilement accessible, géré et mis à jour. Elle est utilisée par les organisations comme méthode de stockage, de gestion et de récupération de l'informations.

Les données sont organisées en lignes, colonnes et tableaux et sont indexées pour faciliter la recherche d'informations. Les données sont mises à jour, complétées ou encore supprimées au fur et à mesure que de nouvelles informations sont ajoutées. Elles contiennent généralement des agrégations d'enregistrements ou de fichiers de données, tels que les transactions de vente, les catalogues et inventaires de produits et les profils de clients.

Généralement, l'administrateur de la base de données régule les accès des utilisateurs afin de contrôler leurs actions et d'analyser les usages. Pour garantir la cohérence des données et l'intégralité des transactions, toutes les transactions réalisées sur une base de données doivent répondre aux exigences de la conformité ACID :

- **Le principe d'Atomicité** garantit la bonne exécution de la transaction. Les transactions de base de données, comme les atomes, peuvent être décomposées en plus petites parties. Si une partie d'une transaction échoue, toute la transaction sera annulée.
- **La propriété de Cohérence** signifie que seules les données qui suivent des règles prédéfinies peuvent être écrites dans la base de données.
- **L'isolement** fait référence à la capacité de traiter simultanément plusieurs transactions de manière indépendante.
- **La durabilité** requiert de rendre les défaillances invisibles pour l'utilisateur final. Les données sont sauvegardées une fois la transaction terminée, même en cas de panne de courant ou de défaillance du système.

Example d'une DB

users

id	full_name	enabled	last_login
1	John Smith	f	2017-10-25 10:26:10.015152
2	Alice Walker	t	2017-10-25 10:26:50.295461
3	Harry Potter	t	2017-10-25 10:26:50.295461
5	Jane Smith	t	2017-10-25 10:36:43.324015

checkouts

id	user_id	book_id	checkout_date	return_date
1	1	1	2017-10-15 14:43:18.095143-07	
2	1	2	2017-10-05 16:22:44.593188-07	2017-10-13 13:05:12.673382-05
3	2	2	2017-10-15 11:11:24.994973-07	2017-10-22 17:47:10.407569-07
4	5	3	2017-10-15 09:27:07.215217-07	

books

id	title	author	published_date	isbn
1	My First SQL book	Mary Parker	2012-02-22 12:08:17.320053-03	981483029127
2	My Second SQL book	John Mayer	1972-07-03 09:22:45.050088-07	857300923713
3	My Third SQL book	Cary Flint	2015-10-18 14:05:44.547516-07	523120967812

addresses

user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA
3	3 Main Street	Boston	MA

reviews

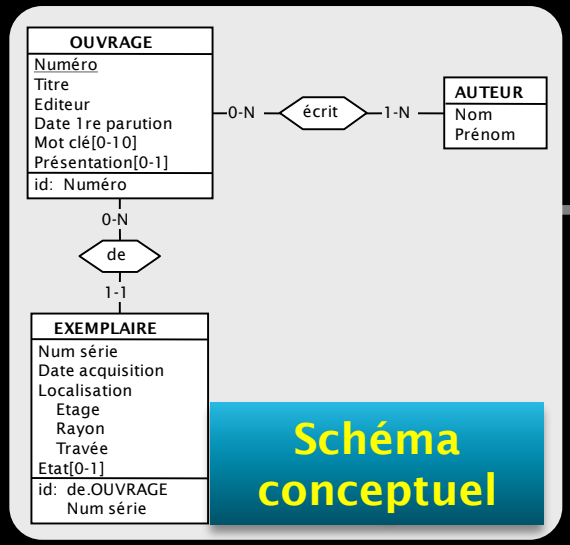
id	book_id	reviewer_name	content	rating	published_date
1	1	'John Smith'	'My first review'	4	2017-12-10 05:50:11.127281-02
2	2	'John Smith'	'My second review'	5	2017-10-13 15:05:12.673382-05
3	2	'Alice Walker'	'Another review'	1	2017-10-22 23:47:10.407569-07

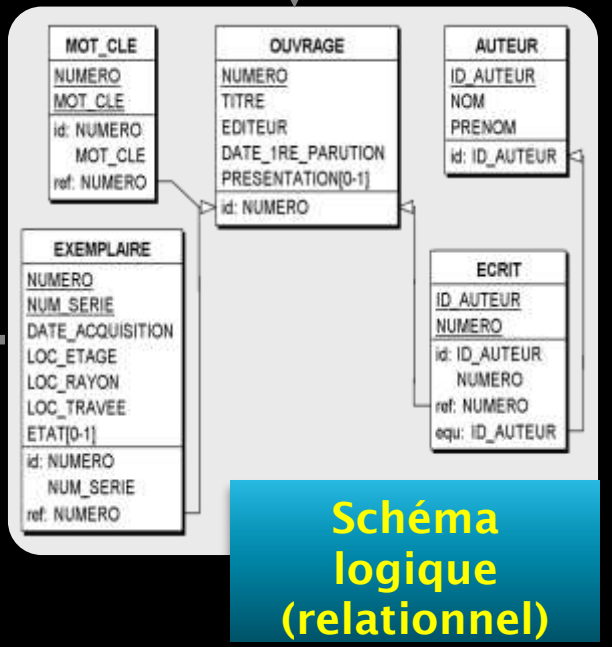
Chapitre 2: Le modèle Entité-Association

1. Méthodologie de modélisation

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. ...






```
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 physique (Oracle 11)

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

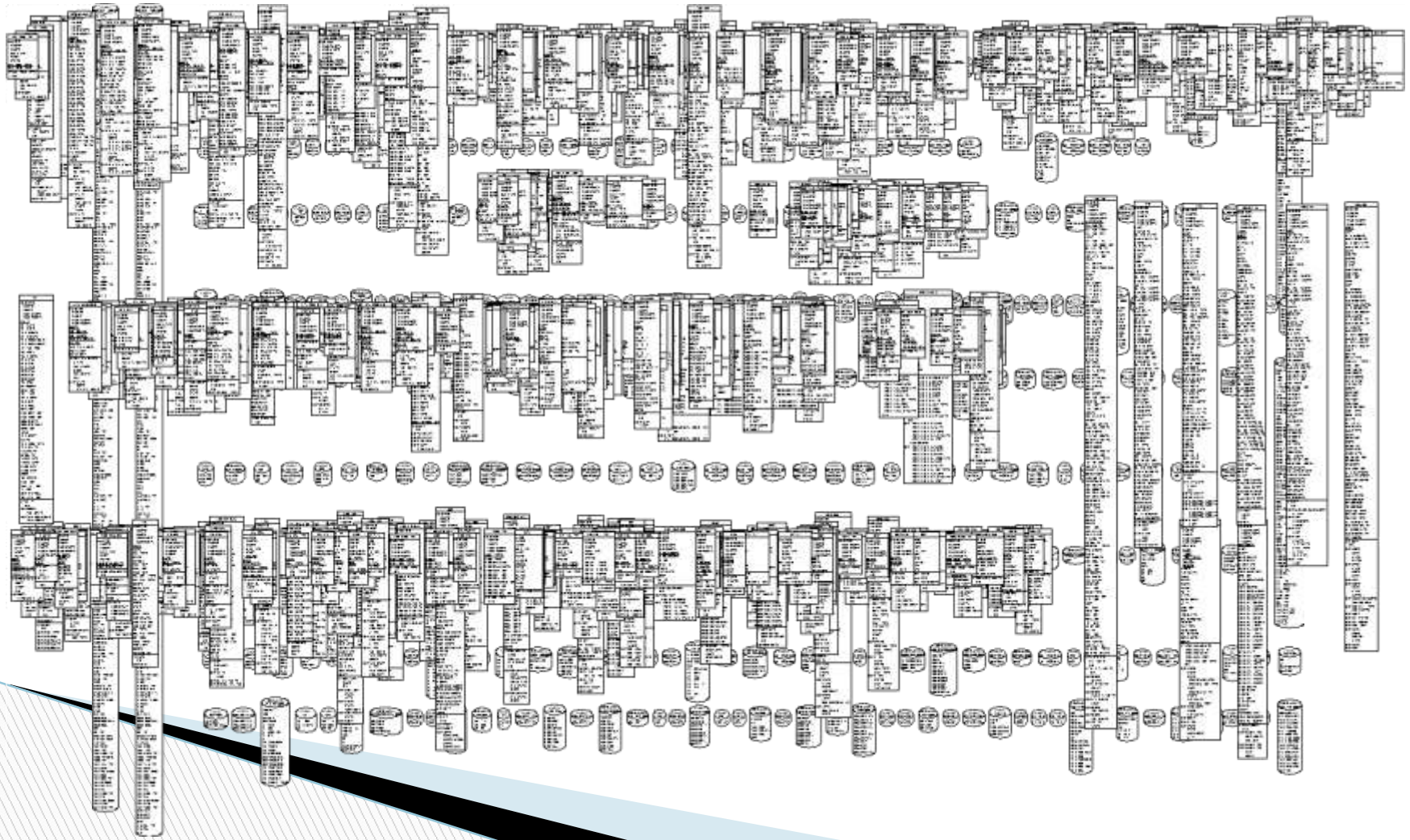
- Une base de données de taille moyenne:
  - comporte plusieurs centaines de tables
  - comporte plusieurs milliers de colonnes
- Une base de données de grande taille:
  - comporte plusieurs milliers de tables
  - comporte plusieurs dizaines de milliers de colonnes
- SAP utilise une BD de:
  - de près 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

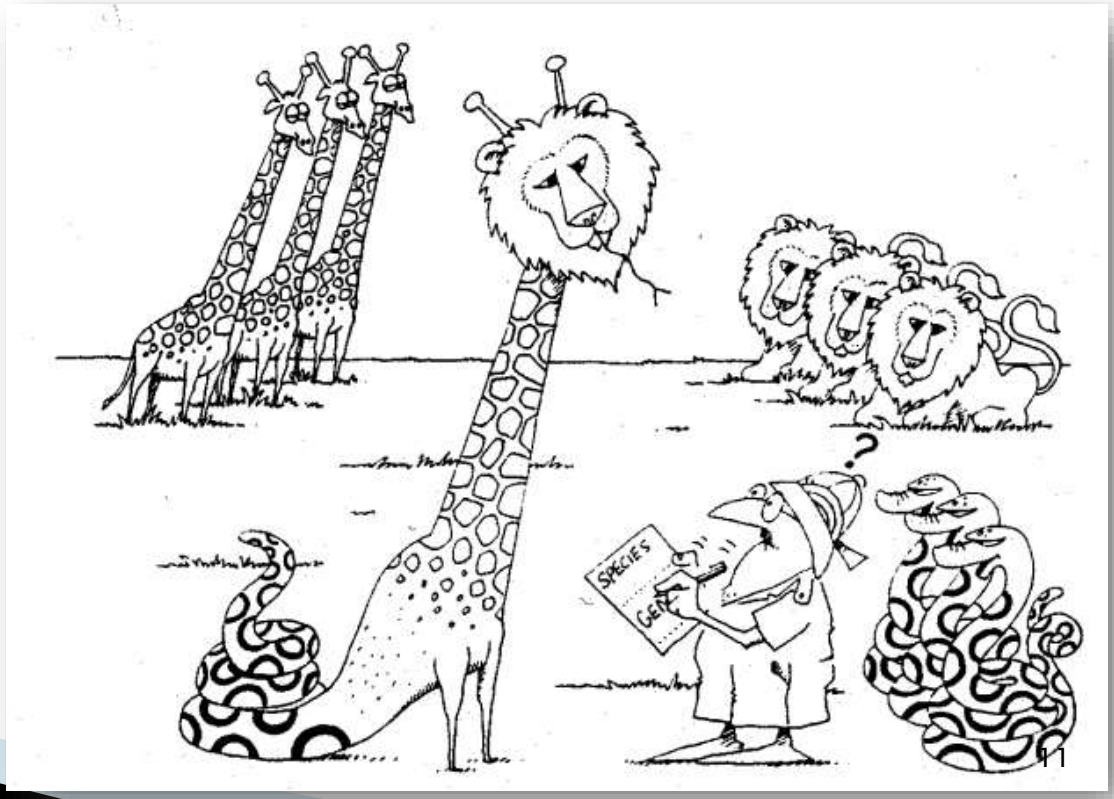


Chapitre 2: Le modèle Entité-Association

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

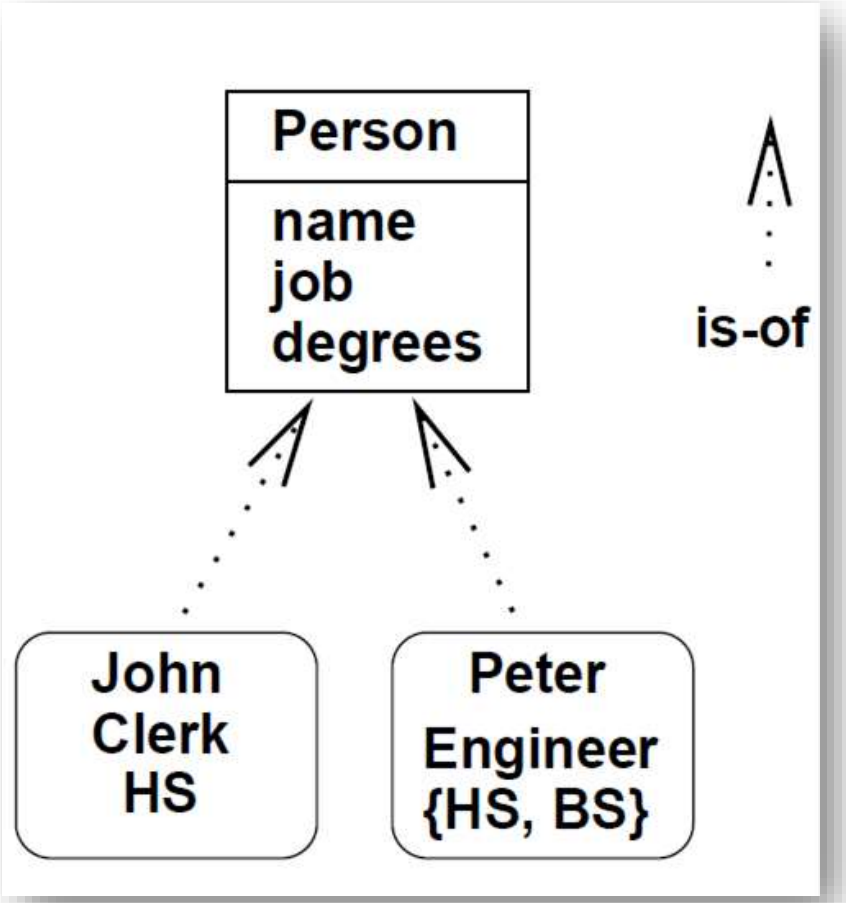
L'Entité

- 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 Souson, 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:

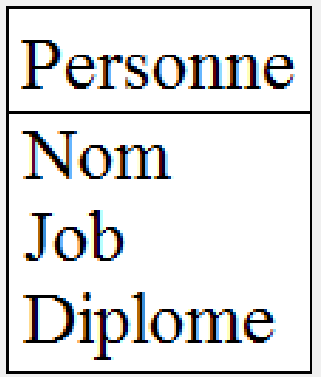
  Le film mythique « Dear Friends » produit par Universal dans le Kansas, Etats-Unis, en 1987 et sorti dans les salles l'année suivante (1988) vient encore de remporter un prix récemment. Ce prix, nouveau triomphe pour l'équipe de production, a été remis à la veuve du réalisateur, Thomy Lee. L'acteur José 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

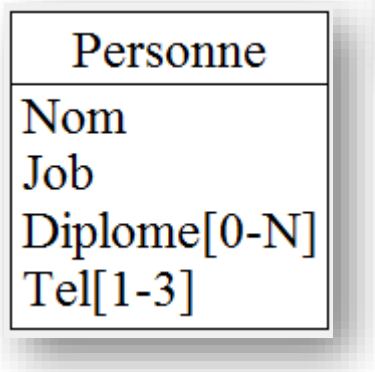
Les attributs: définition

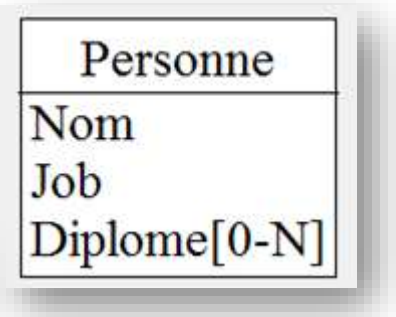
- Un **Attribut** d'une Classe est une caractéristique partagée par (toutes?) ses Entités
- Chaque Attribut a un domaine de valeurs (type) précis
- Chaque Entité a une valeur pour les (chaque?) attributs de sa Classe
- **Attribut atomique**



La cardinalité d'un Attribut

- Une **cardinalité**: concept mathématique définissant le nombre d'élément (ou de relations) dans/entre ensembles finis
- Soit la cardinatlité [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]





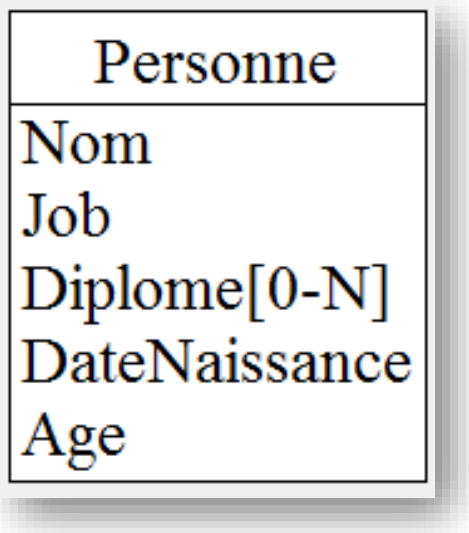
La valeur NULL

- 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 connaitre
  - **Inconnu actuellement, la valeur existe peut-être**: on ne connait 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 un moyenne ou compter un nombre d'enregistrement)

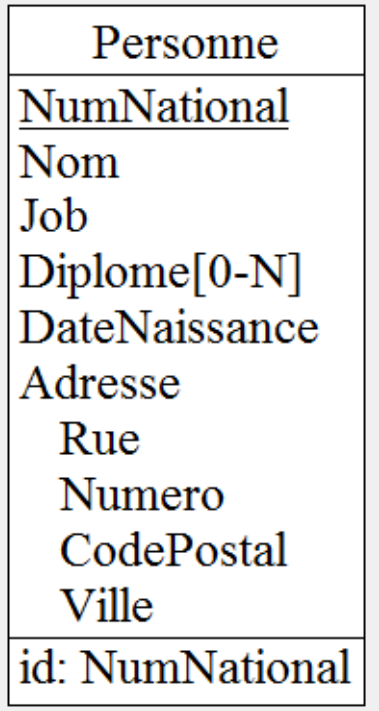
Attribut dérivé

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



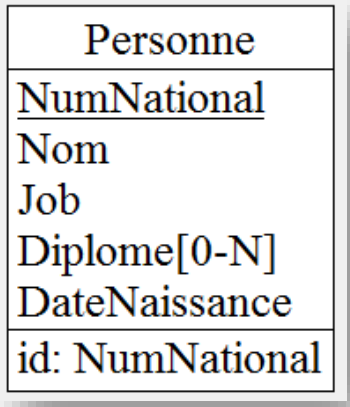
Attribut composite

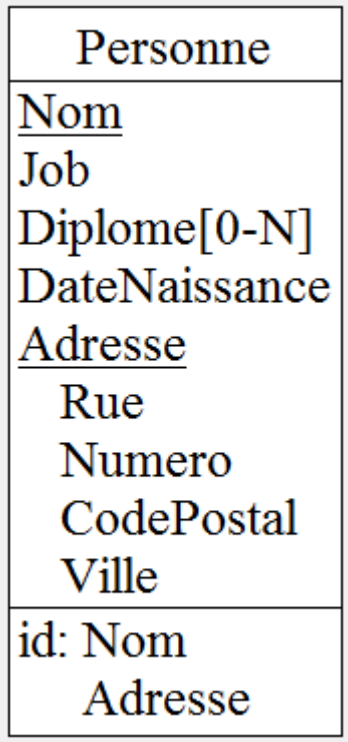
- Un **Attribut composite** est un attribut qui est décomposé en d'autres attributs
- Un attribut composite est donc un ensemble de valeurs mise en relation
- Un attribut atomique est donc un attribut indivisible
- C'est un groupe de valeurs qui n'est pas suffissant que pour être une entité en tant que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



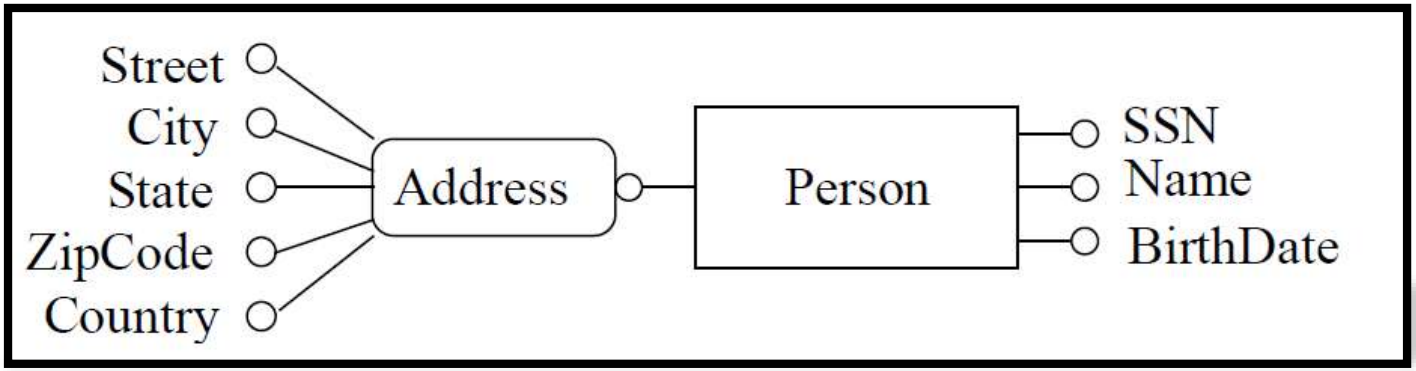
Attribut identifiant

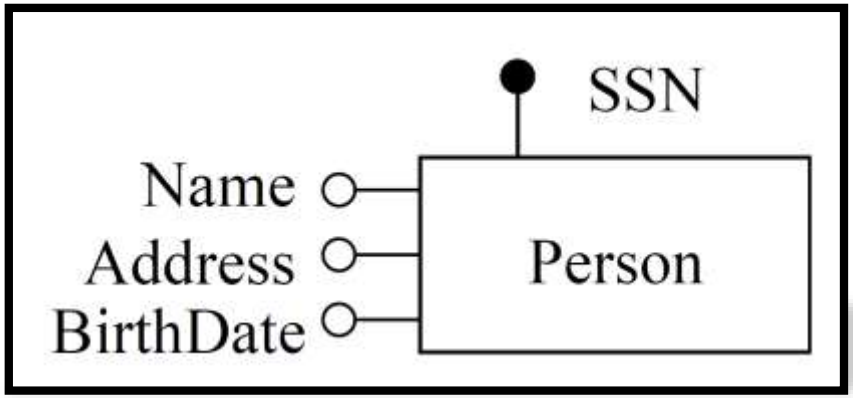
- Un **attribut identifiant** est un attribut (simple ou complexe) dont la valeur peut à elle seule identifier chaque entité de la Classe par rapport aux autres
  - ***Unicité***: pas deux fois la même valeur pour deux entités dans la classe
  - ***Valeur obligatoire*** : pas de valeur NULL =>cardinalité: [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





Autres notations pour les différents types d'Attributs



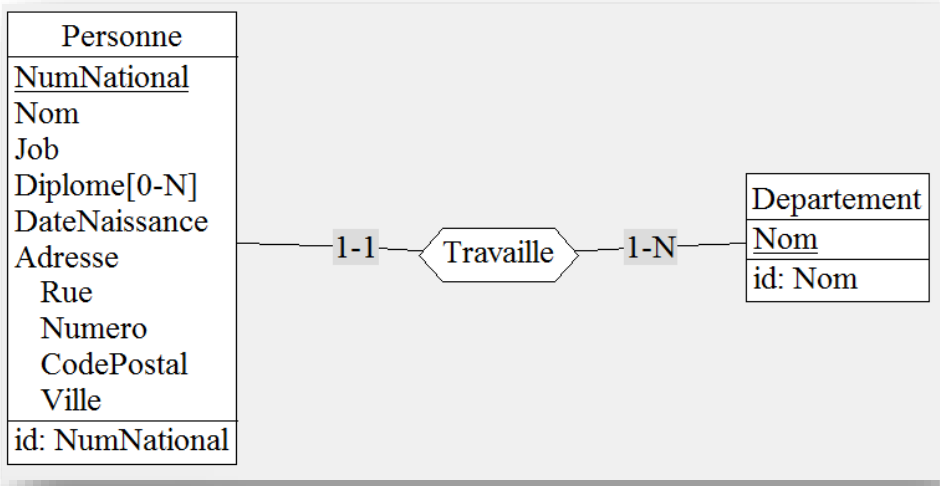


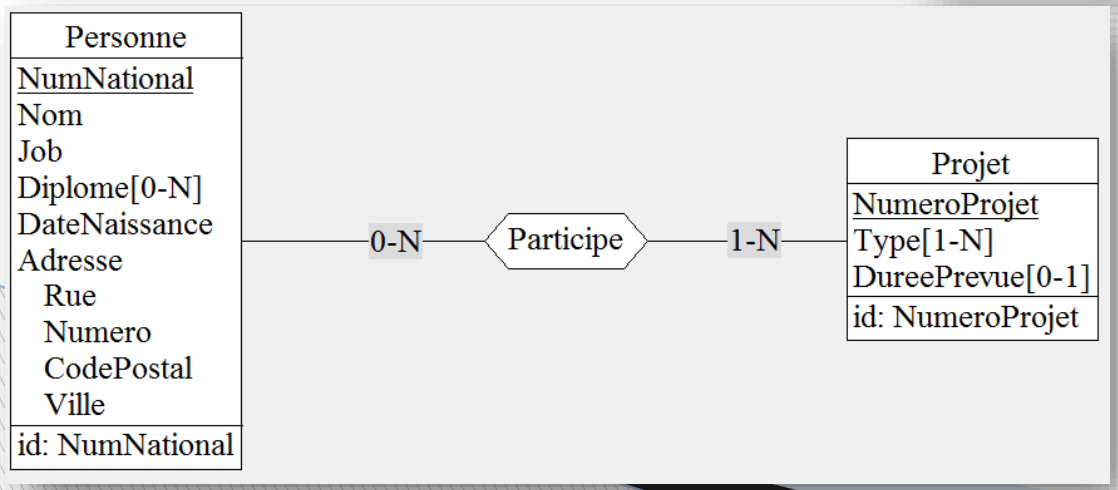
Chapitre 2: Le modèle Entité-Association

4. Les relations: l'Association

L'Association: définition

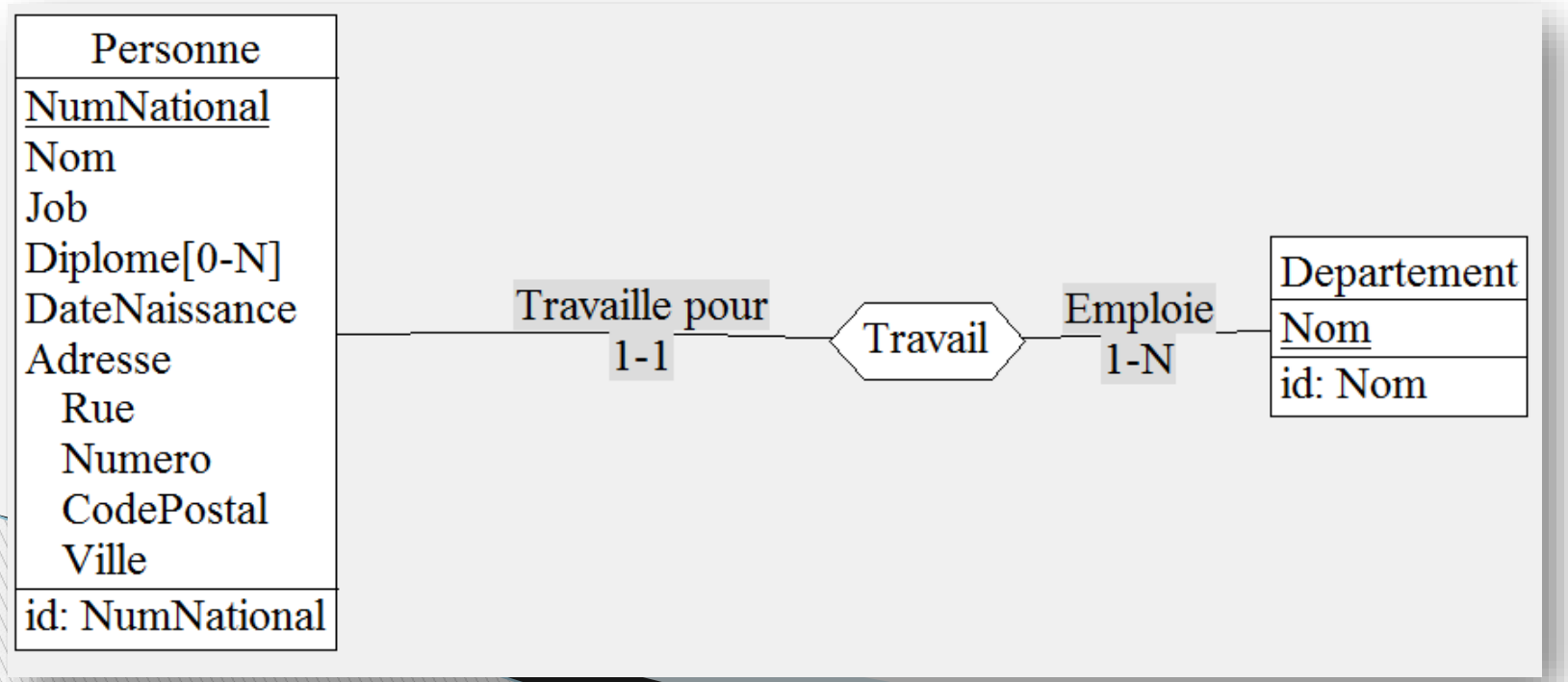
- Une **Association** capture une relation *statique* qu'ont des entités d'une classe avec des entités d'une autre classe
  - Se lie dans les deux sens!



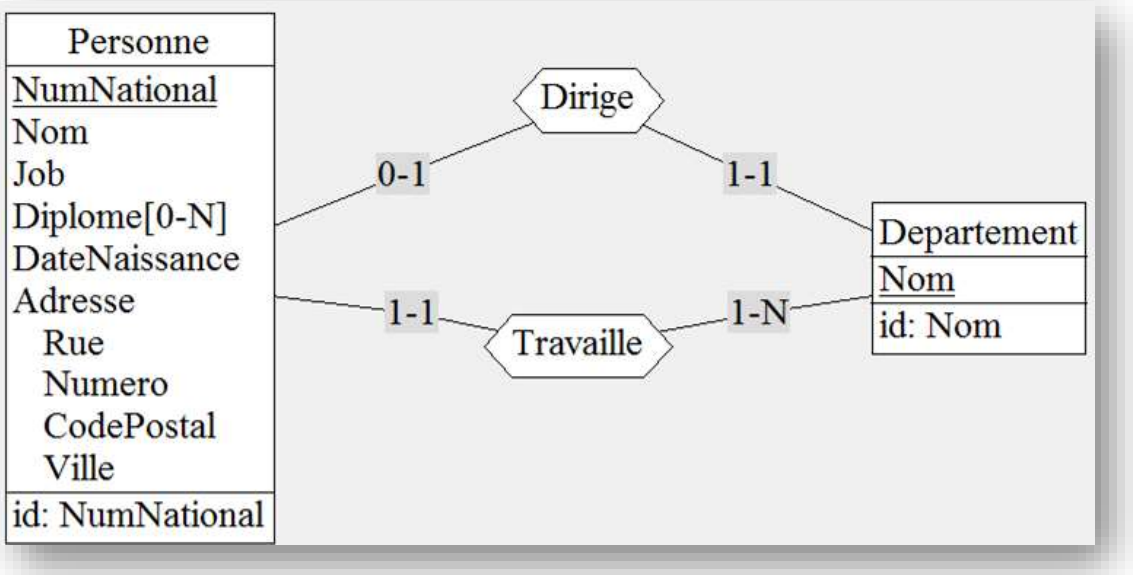


Les rôle d'une Association

- 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

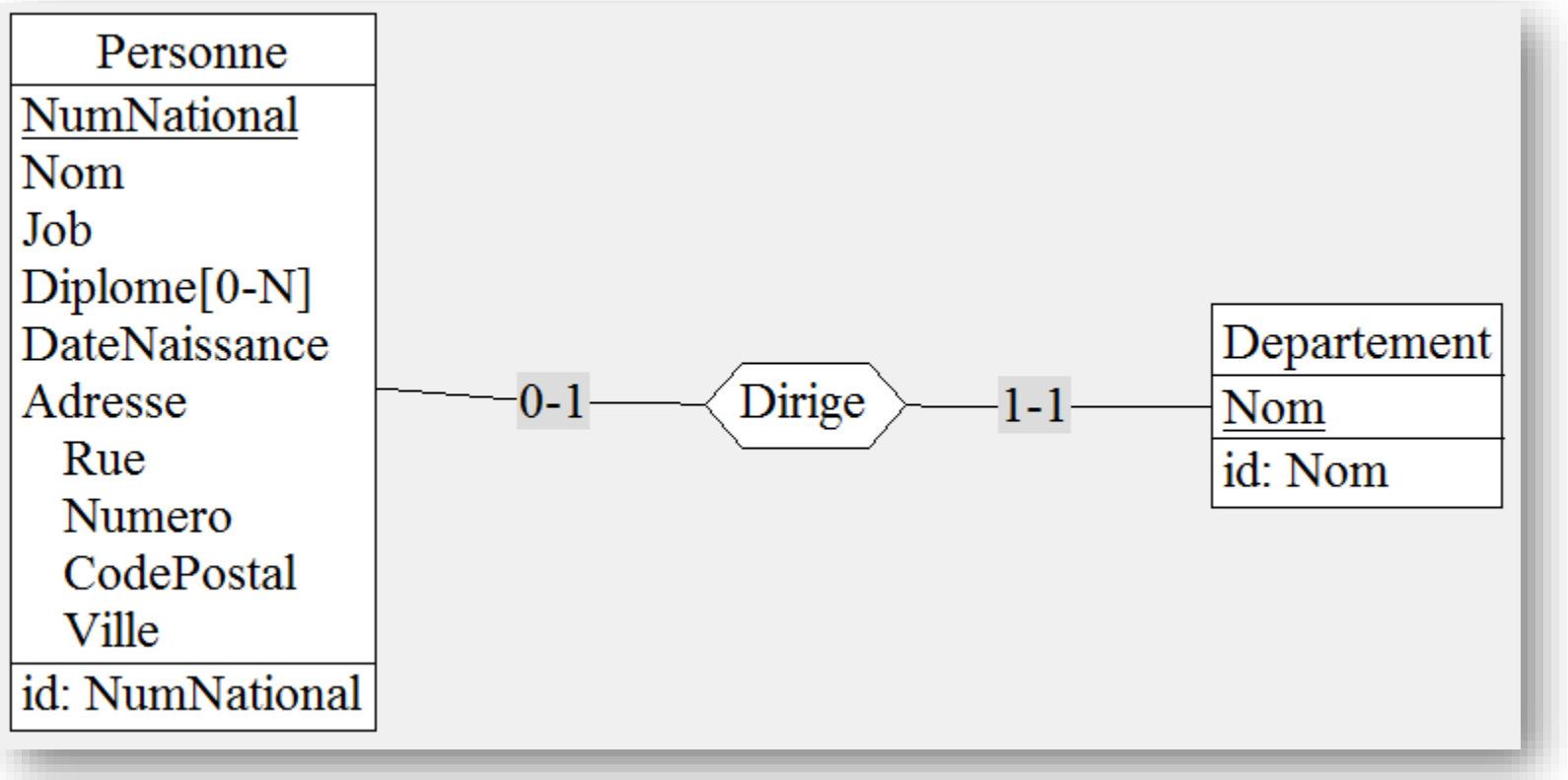


- Chaque association binaire a 2 cardinalités, une pour chacun des rôles
- Une cardinalité a une borne supérieure et une borne inférieure
- Les cardinalités de l'Association '*Travaille*' se lisent:
  - De 'Personne' vers 'Departement': "Une entité de la classe Personne a une et une seule relation Travaille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

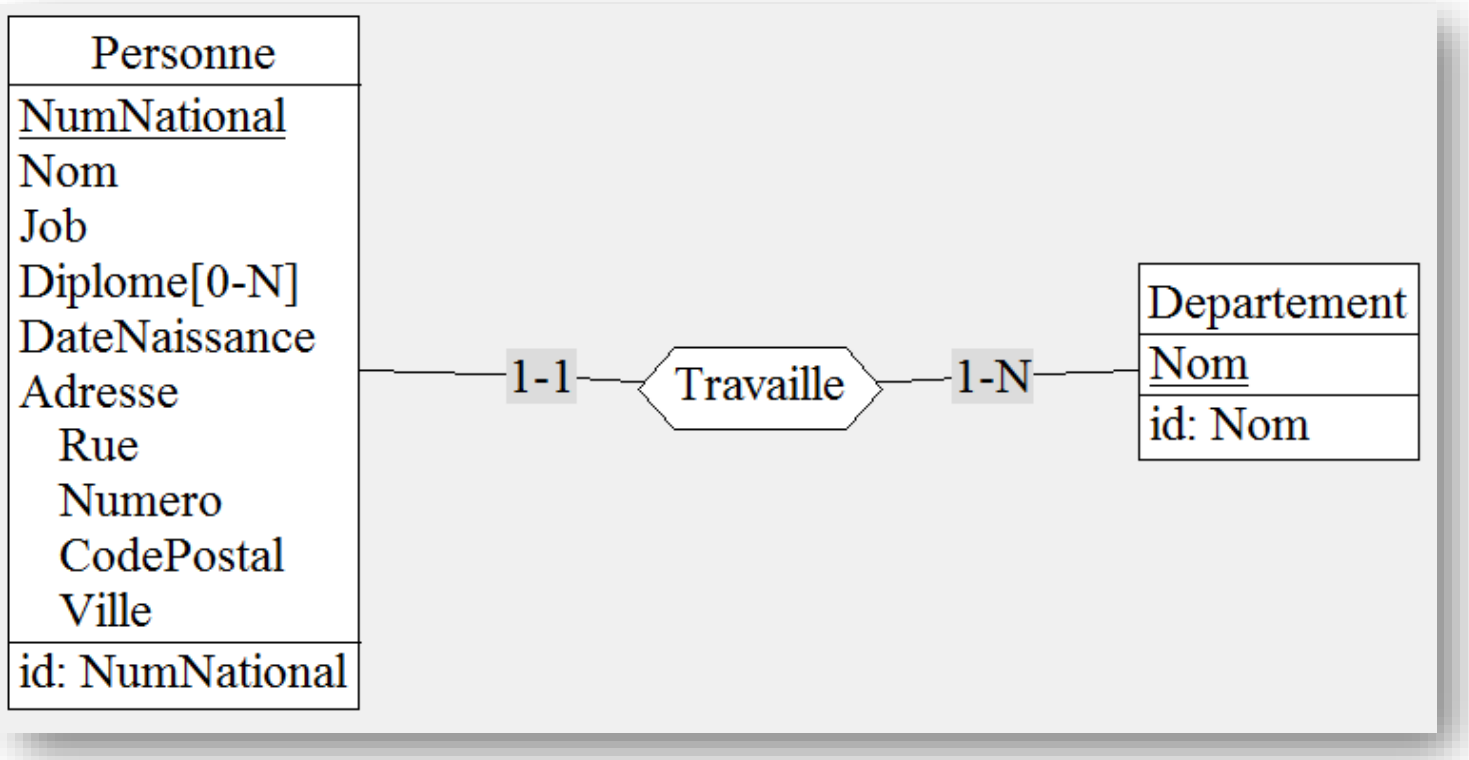
- Soit la cardinalité d'une Association (x,y)
- Optionnel VS. Obligatoire
  - **Optionnel**: x = 0
    Une entité *peut* avoir une relation avec les entités de la classe mise en relation
  - **Obligatoire**: x ≥ 1
    Chaque entité *doit* avoir une relation avec les entités de la classe mise en relation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



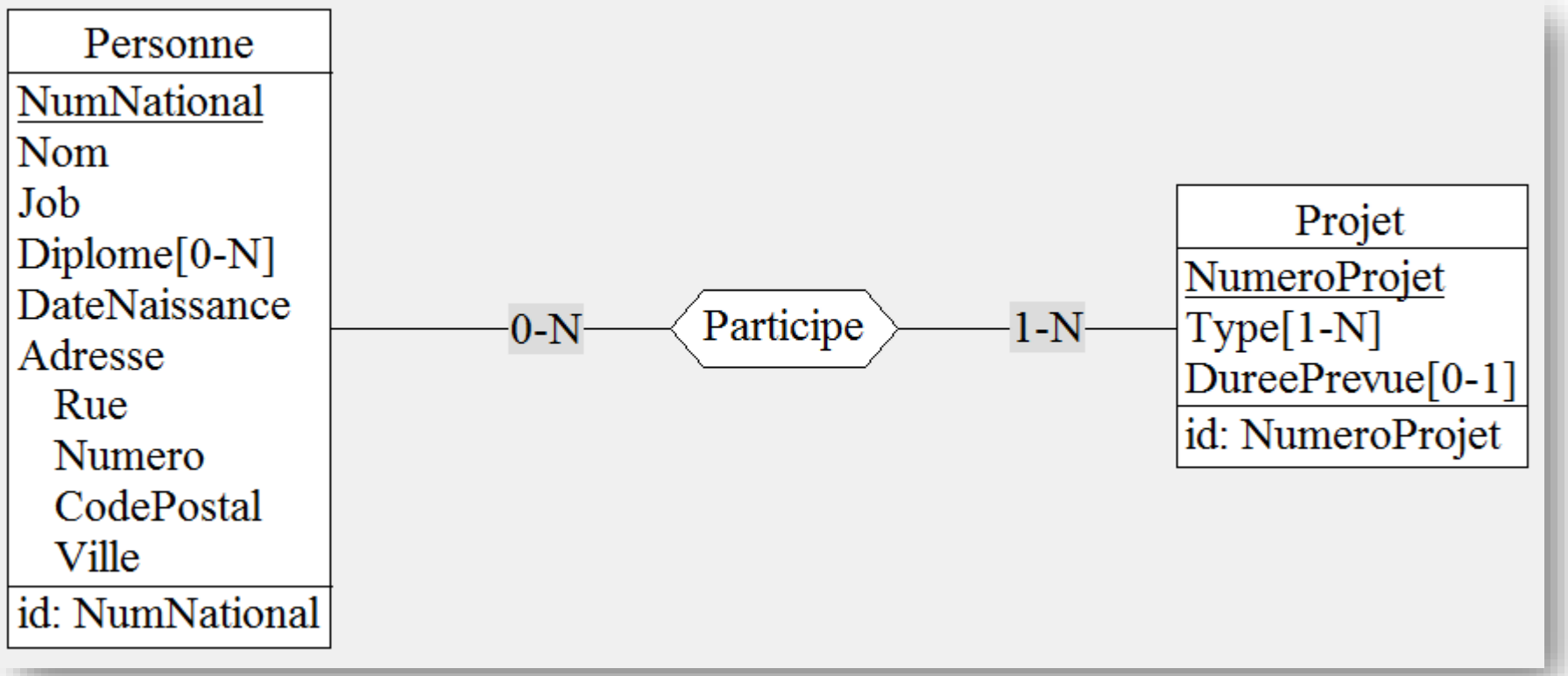
- Une ‘Personne’ peut diriger un seul ‘Departement’
- Un ‘Departement’ est dirigé par une et une seule ‘Personne’

Association One-to-Many



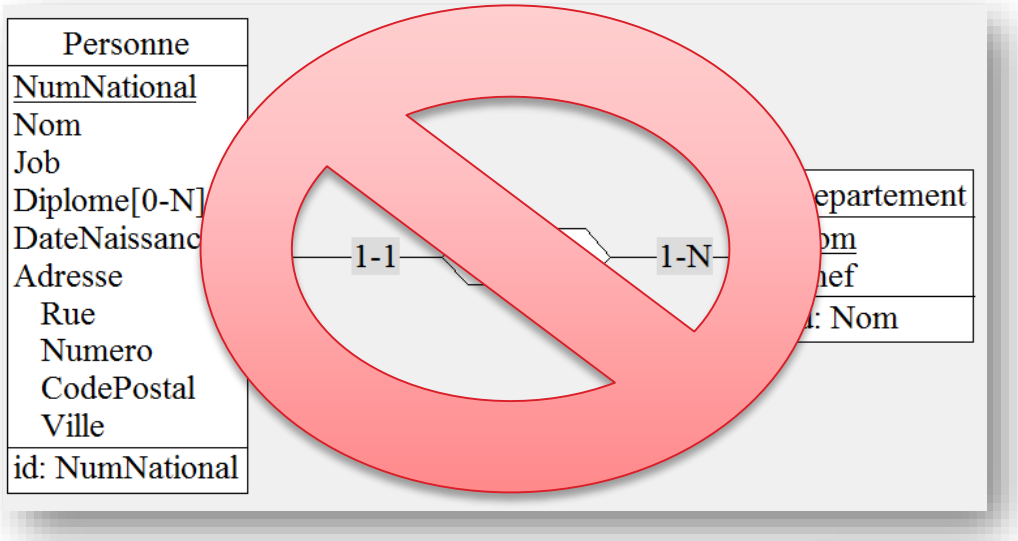
- Chaque ‘Personne’ travaille pour un seul ‘Departement’
- Chaque ‘Departement’ emploie de une à *n* ‘Personnes’
- Une association One-to-Many est similaire à une association Many-to-One (lecture des associations dans les deux sens!)

Association Many-to-Many

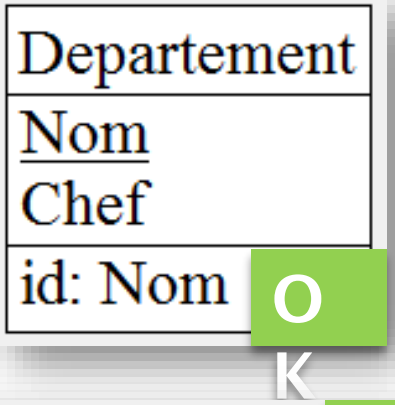


- Chaque entité ‘Personne’ peut travailler pour plusieurs ‘projets’
- Chaque entité ‘Projet’ utilise au moins une ‘personne’, jusque *n* ’personne’

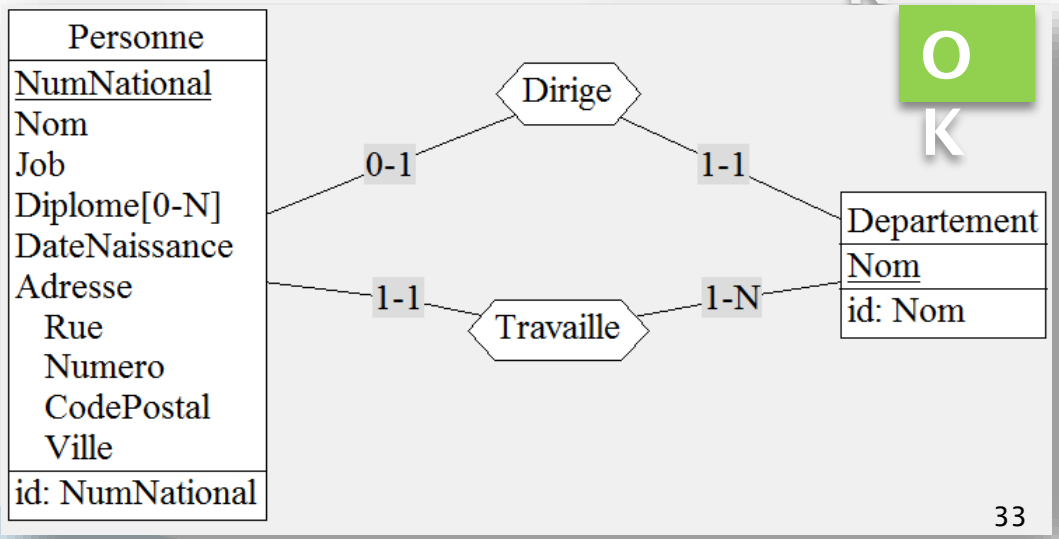
Utiliser adéquatement les Associations



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



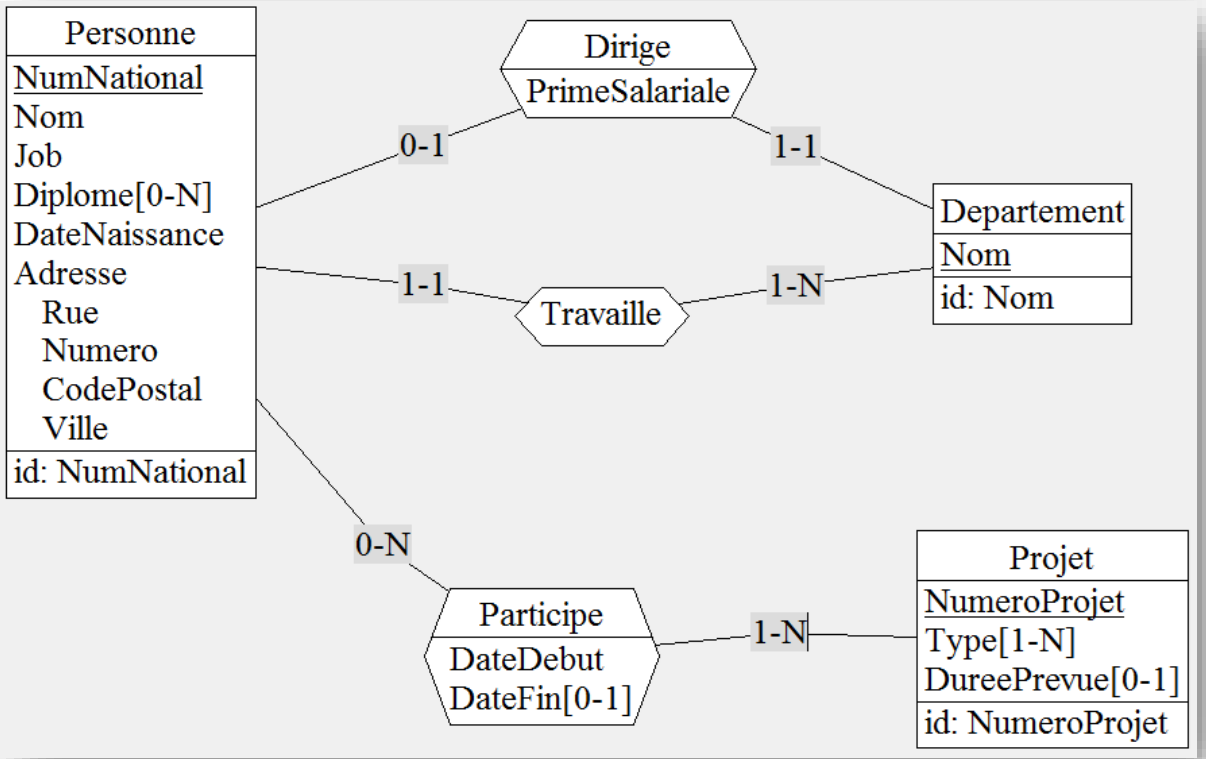
- Utiliser des relations pour capturer les liens du monde réel lorsque une classe de votre schéma capture la même donnée!



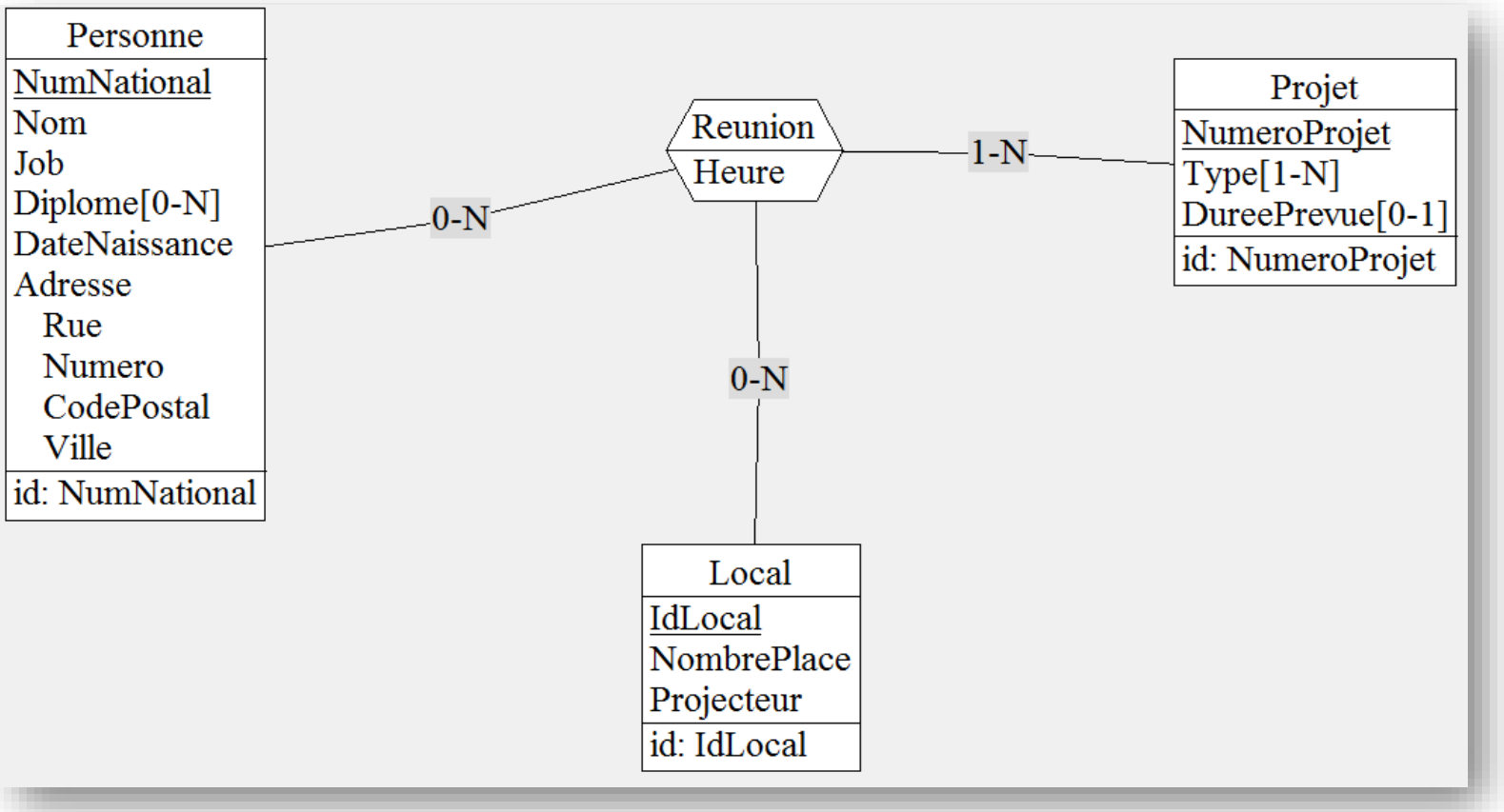
Les Attributs des Associations

- Un Attribut est ajouté a 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

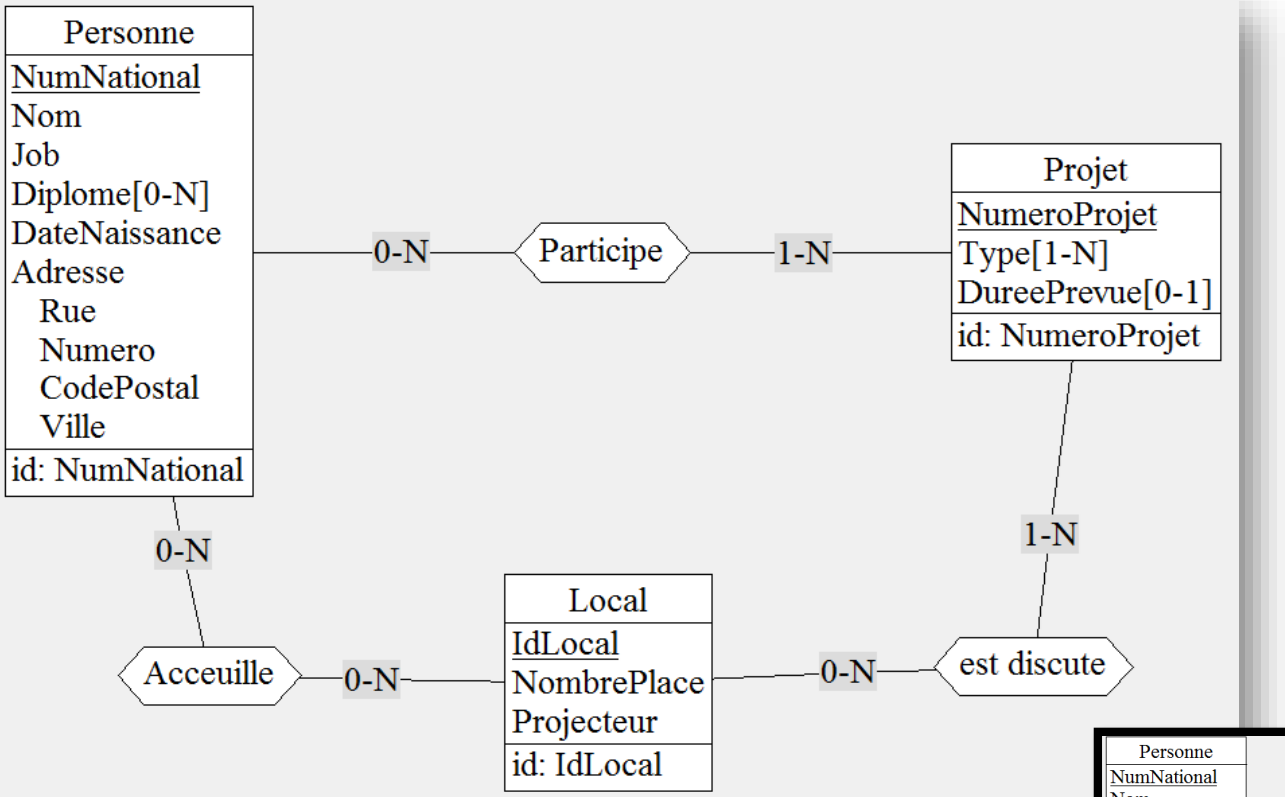


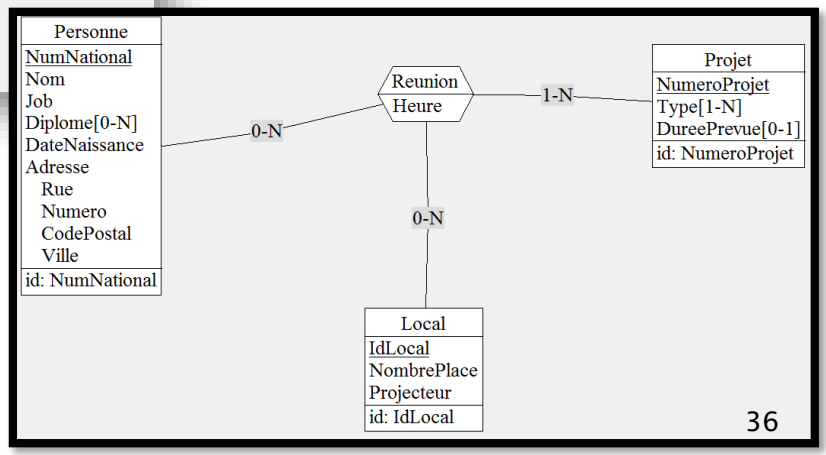
L'Association Ternaire



- **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

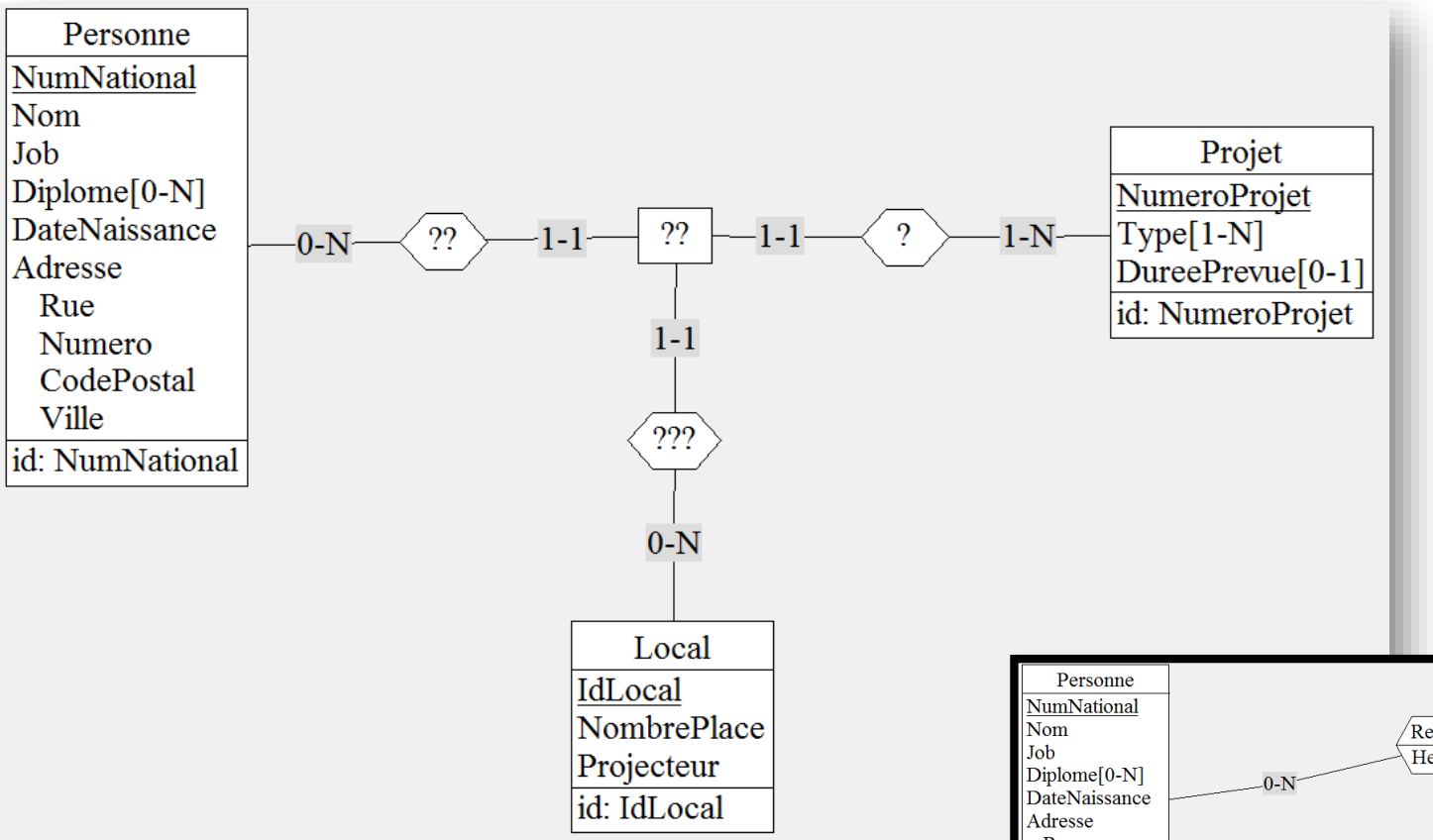
Personne	Local
Georges	Salle 17
Lucas	Salle 18
Georges	Salle 18

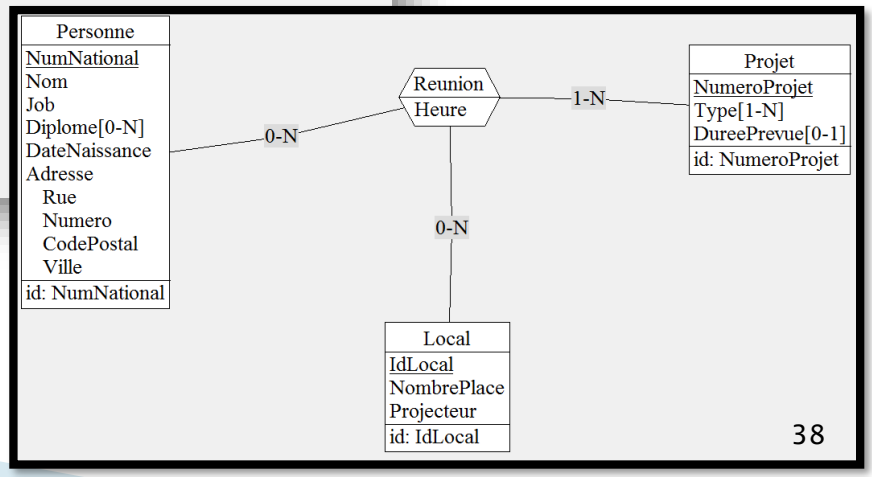
Est discute

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

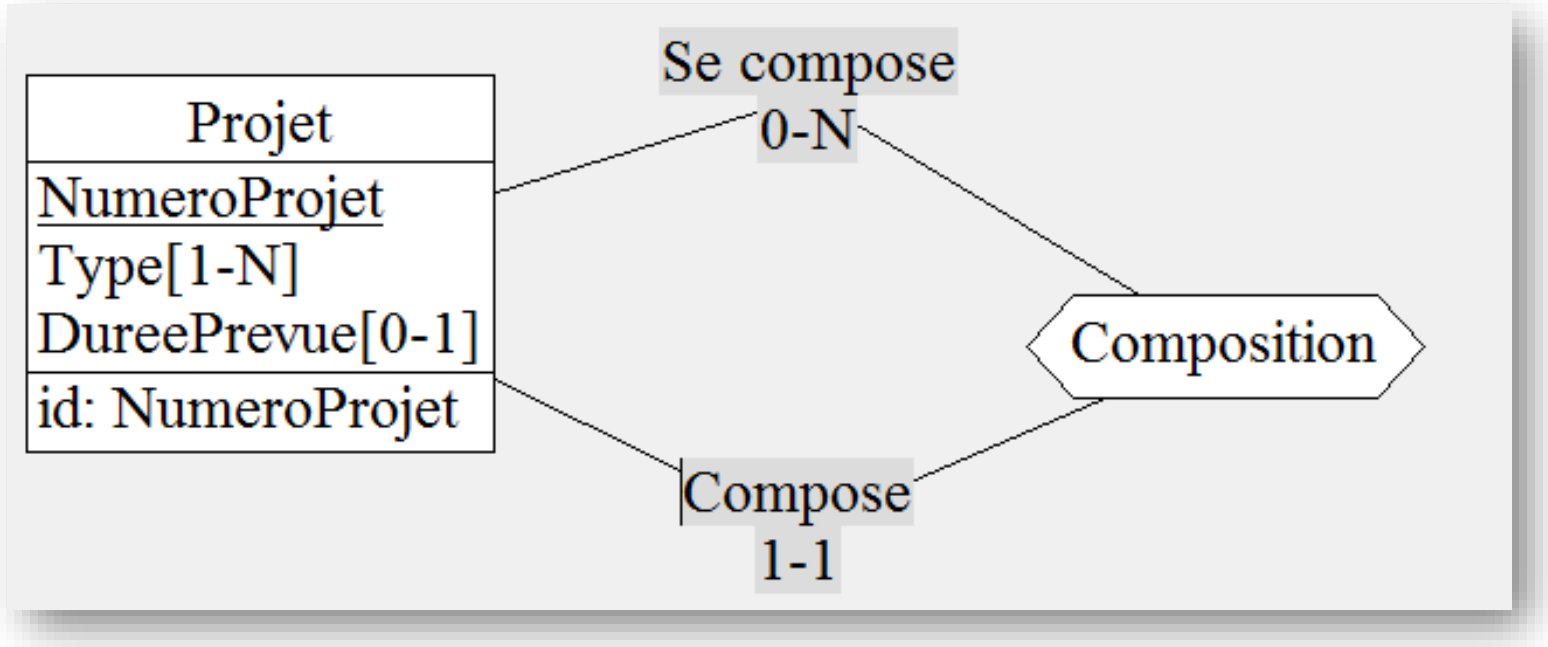
On peut inférer des relations dérivées avec les trois associations binaires qui n'existent pas dans la relation ternaire

Modélisation correcte d'une relation ternaire avec des associations binaires





L'Association 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!

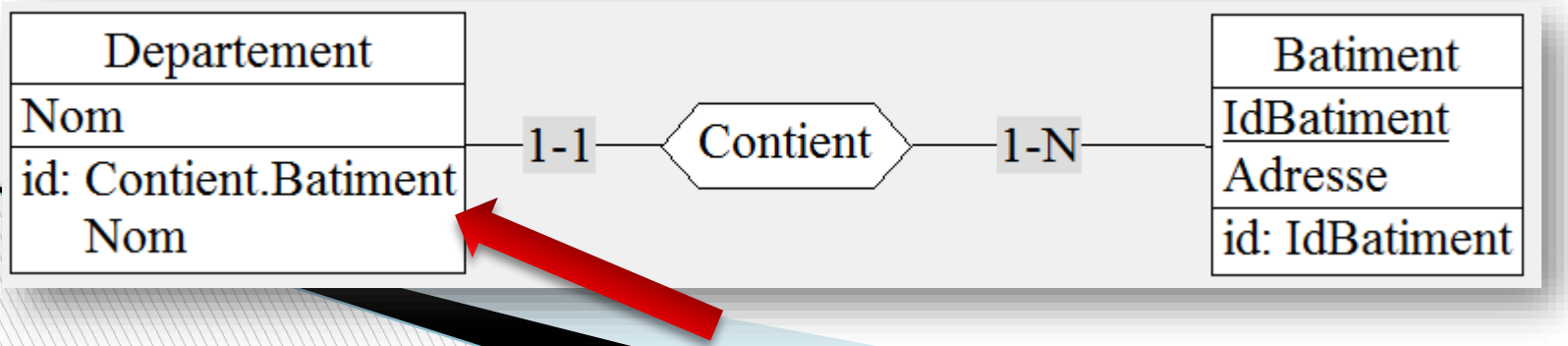
Un exercice d'illustration

- Modélisez la relation récursives suivantes: on identifie des employés dont certains sont supervisés par d'autres. Chaque employé ne peut être supervisé que par maximum un autre employé

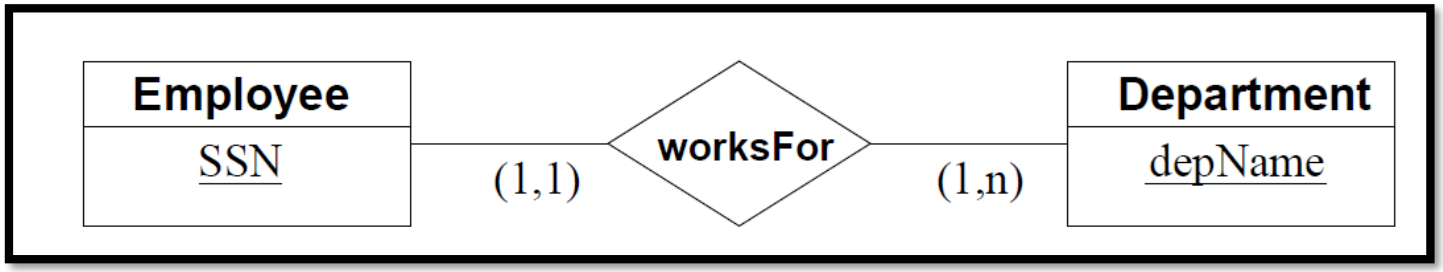
- *Quel est le gros risque de cette modélisation?*

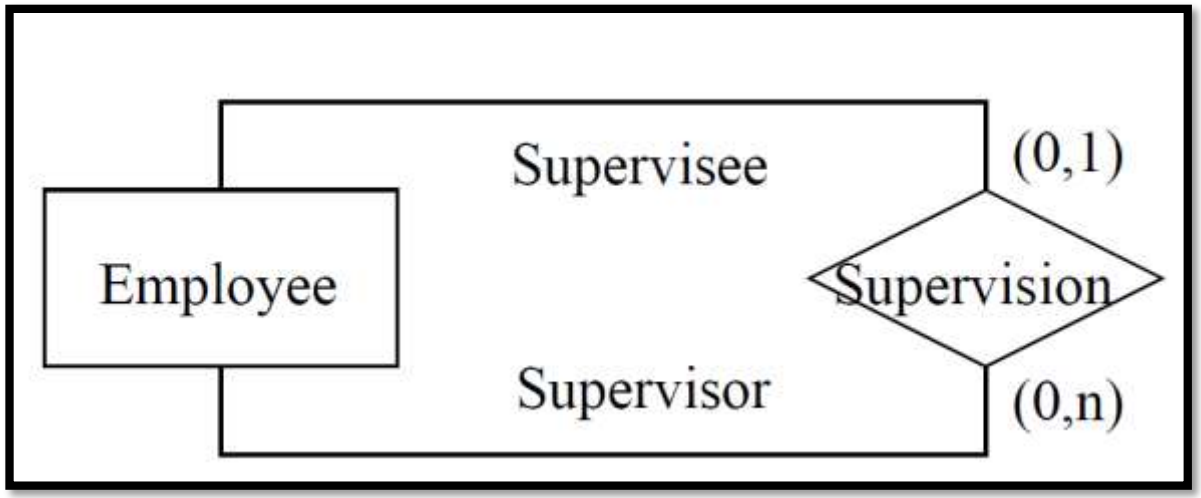
La Classe d'Entité faible

- 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*

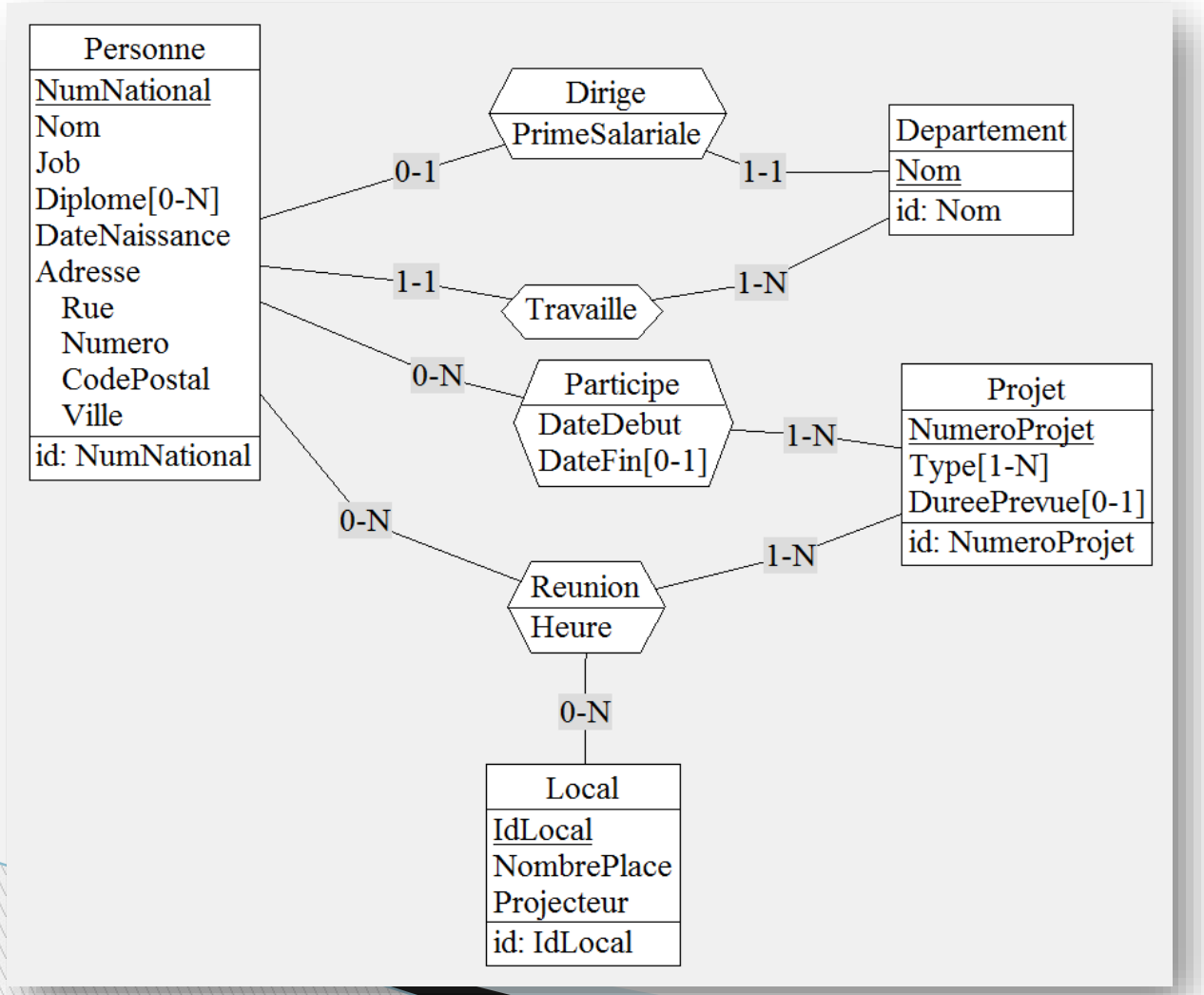


Autres notations pour l'Association





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

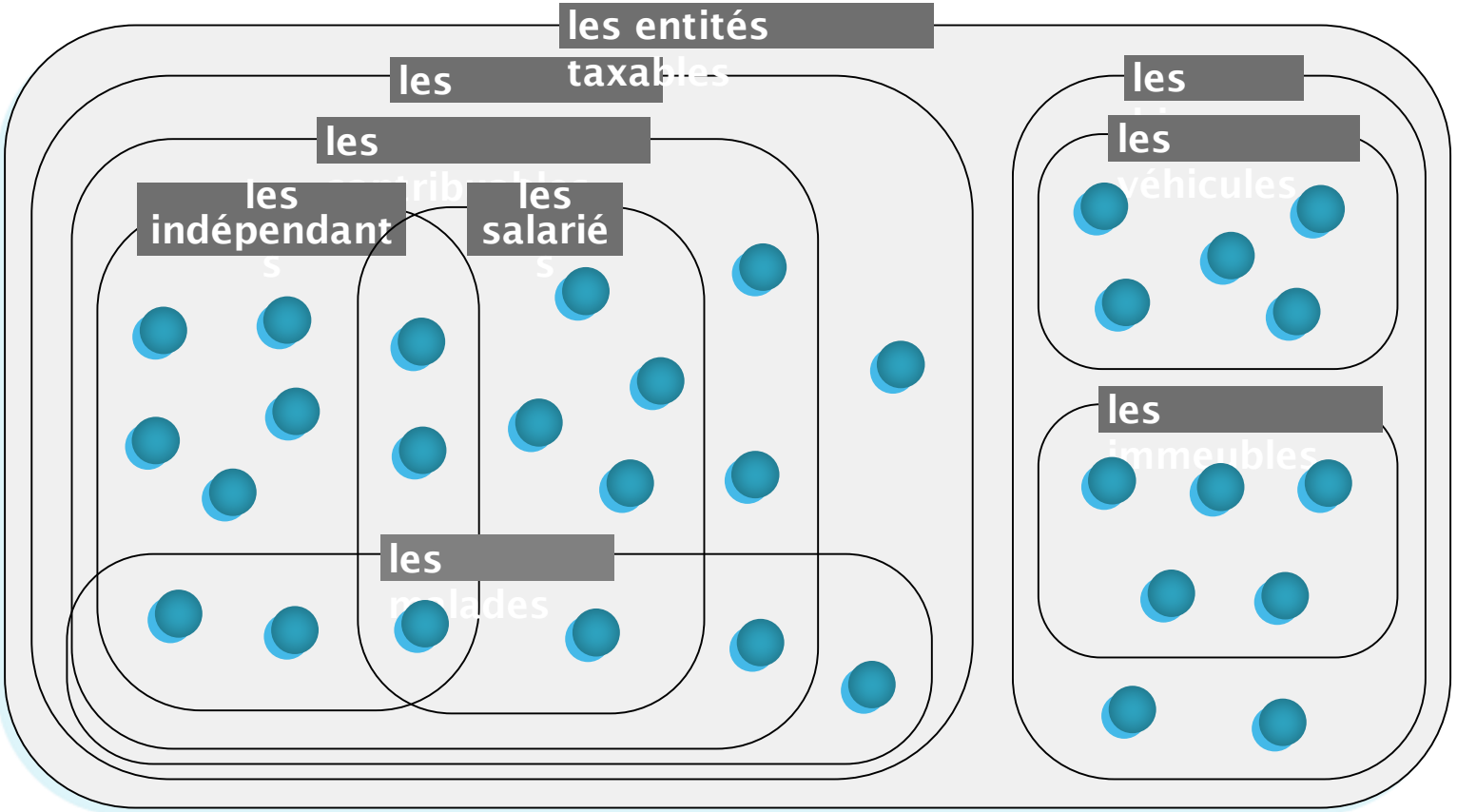
Exercices

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

Chapitre 2: Le modèle Entité-Association

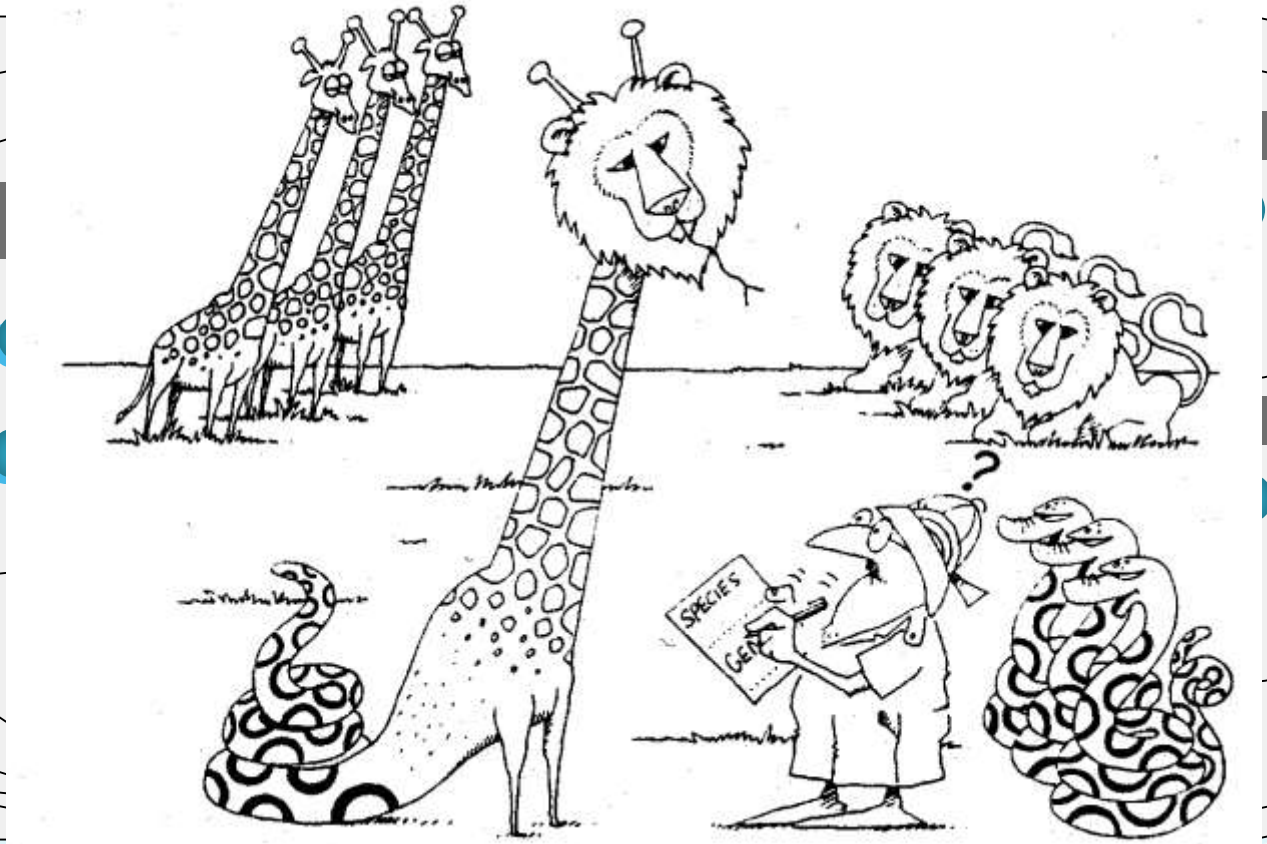
5. Les relations: la Spécialisation-Généralisation

Des Entités ayant plusieurs types



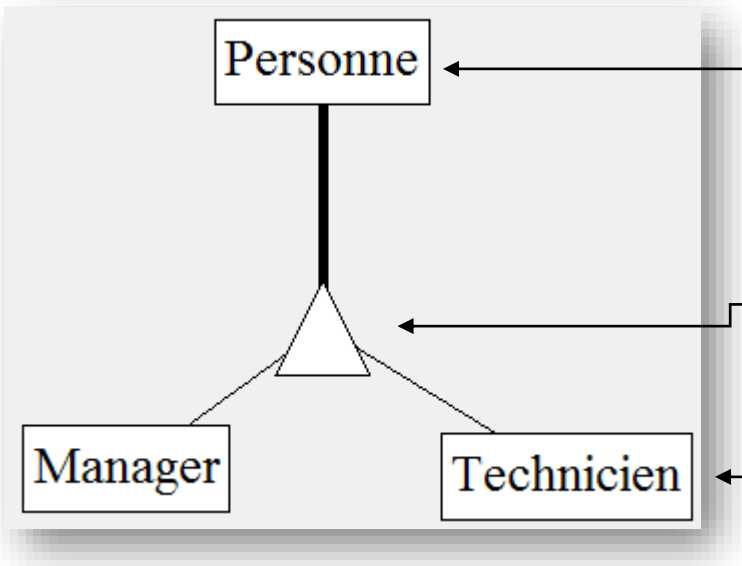
- Des entités peuvent avoir plusieurs types, c-à-d appartenir à plusieurs Classes d'Entités

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



Classe Parent / Classe de Généralisation (=surtype)

relation ***is-a***

Classe Enfant / Classe de spécialisation (=sous-type)

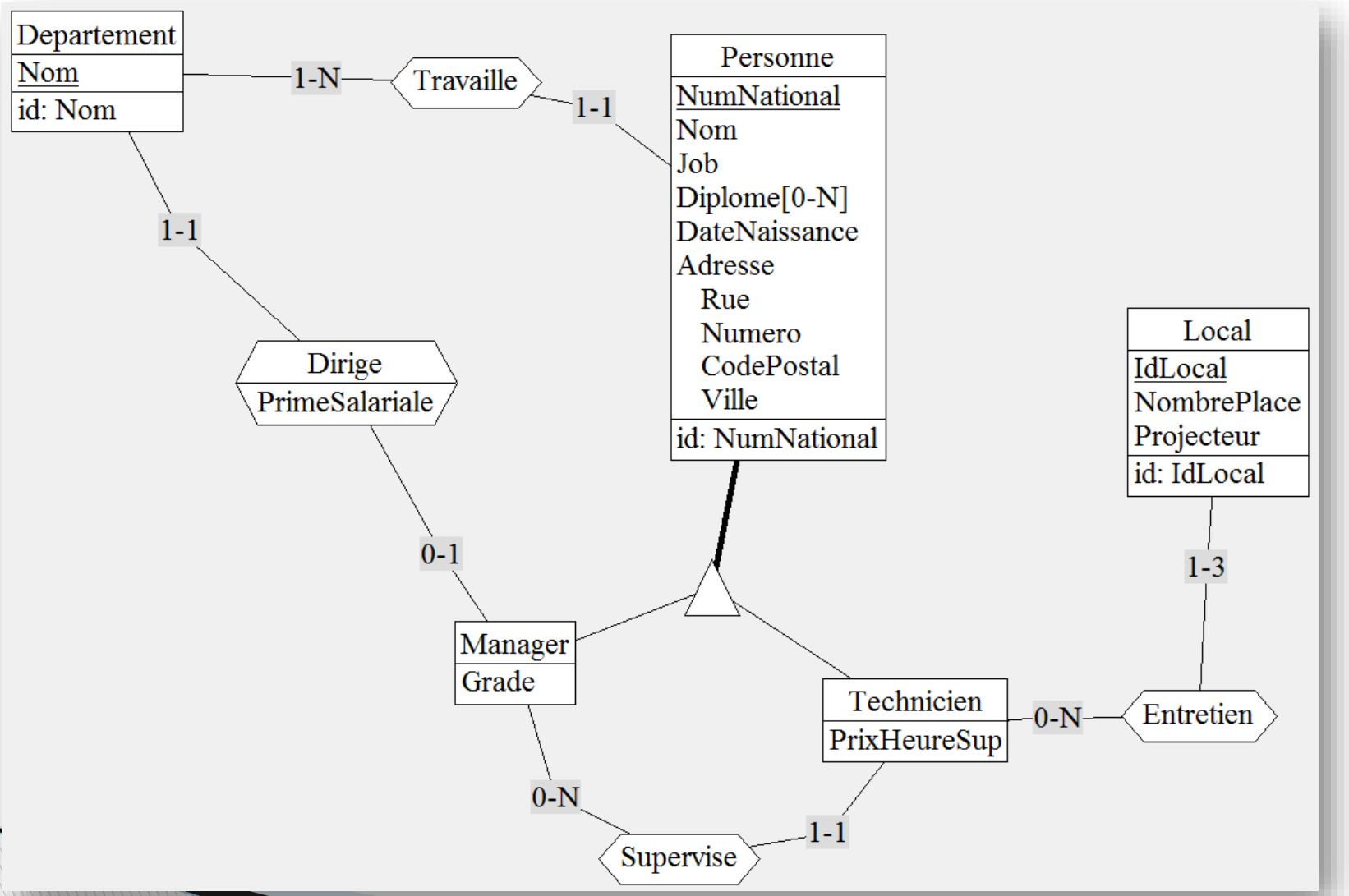
Généralisation d'une Entité

Spécialisation d'une Entité

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

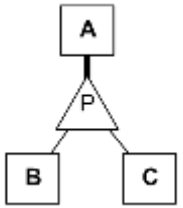
Mécanisme d'héritage: Illustration

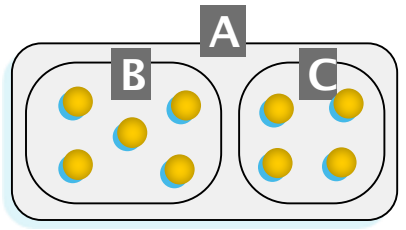


La cardinalité des relations de spécialisation – généralisations

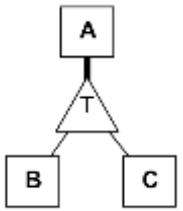
- Une cardinalité peut être:
  - Total *ou* Partiel
  - Exclusive/Disjoint *ou* chevauchement ('overlapping')
- **Total**: toutes les Entités de la classe parent doivent être spécialisées
  Partiel: les entités de la classe parent peuvent être spécialisées (ou pas!)
- **Exclusive**: une entité spécialisée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

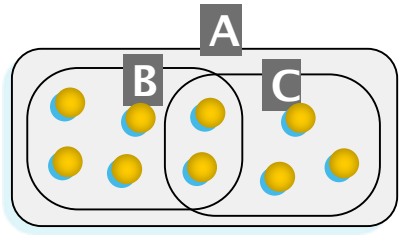
(T,E) Partition



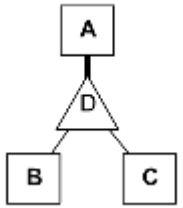


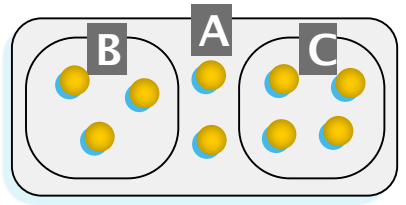
(T, O)



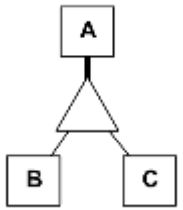


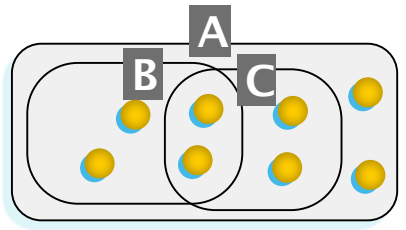
(P, E)



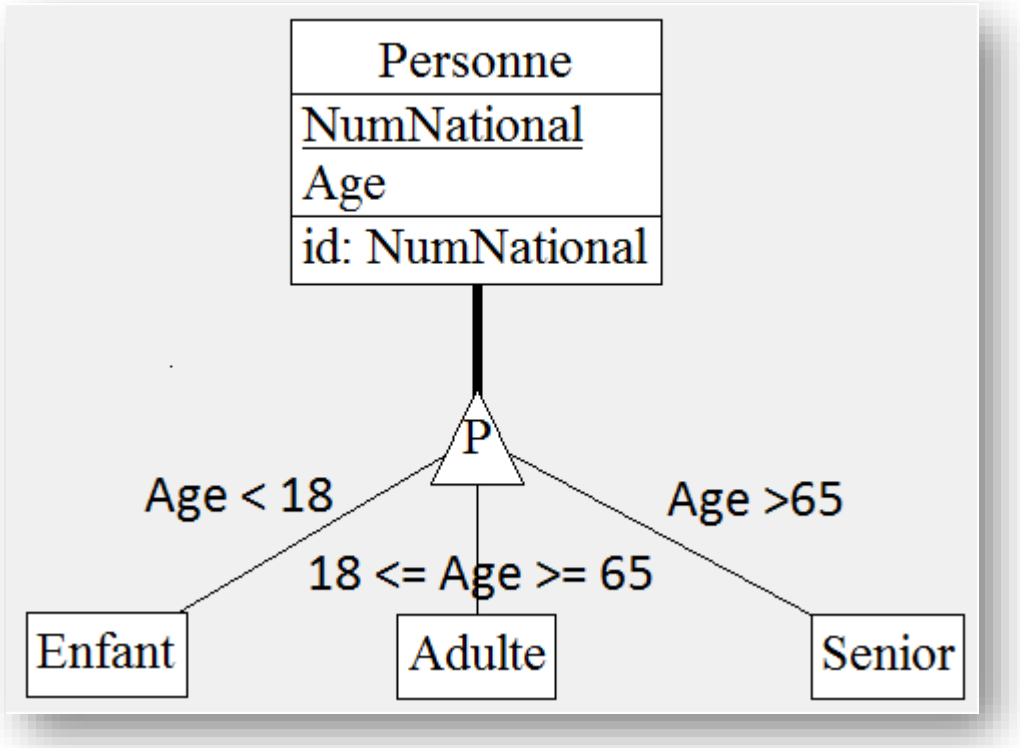


(P, O)



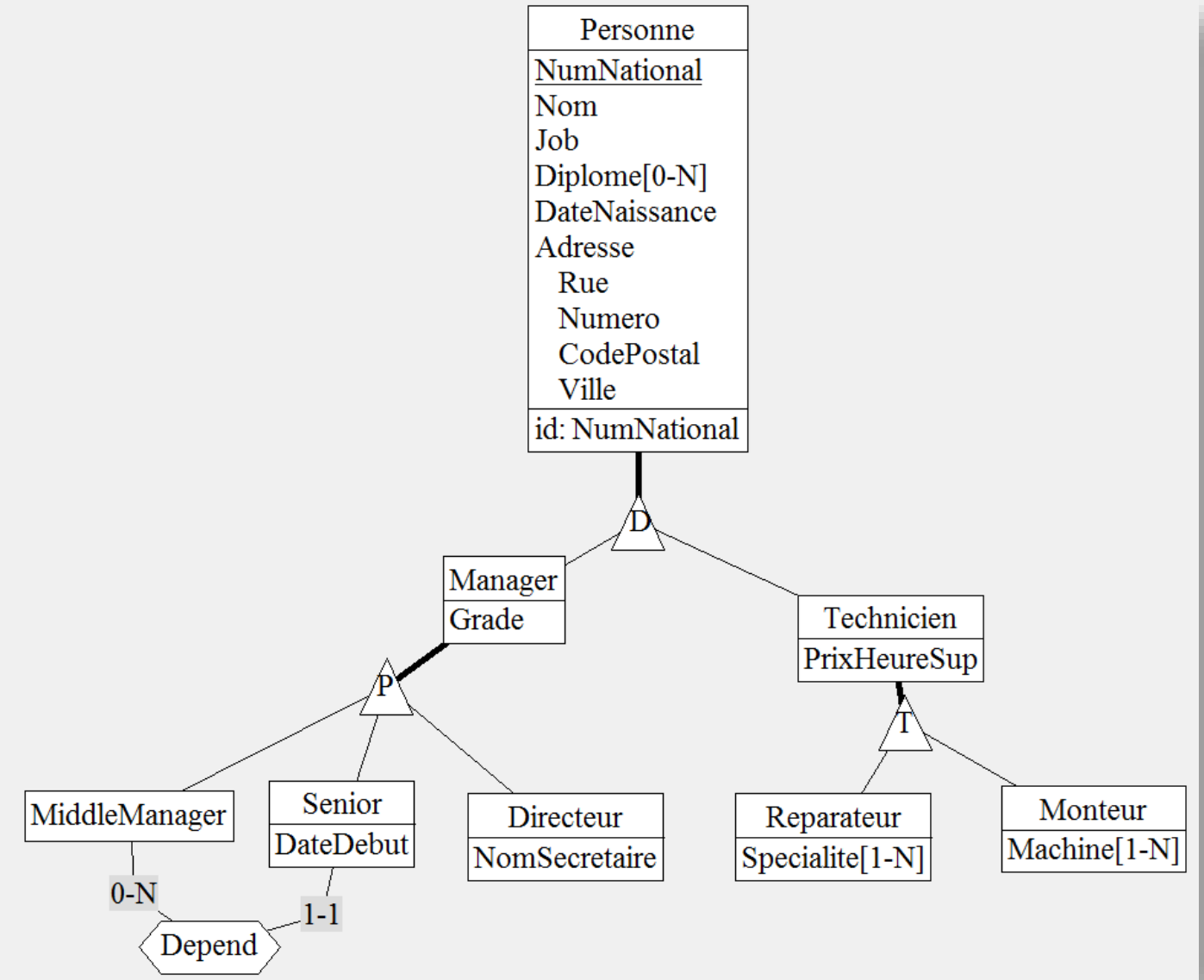


Utilisation de prédicats dans les spécialisation



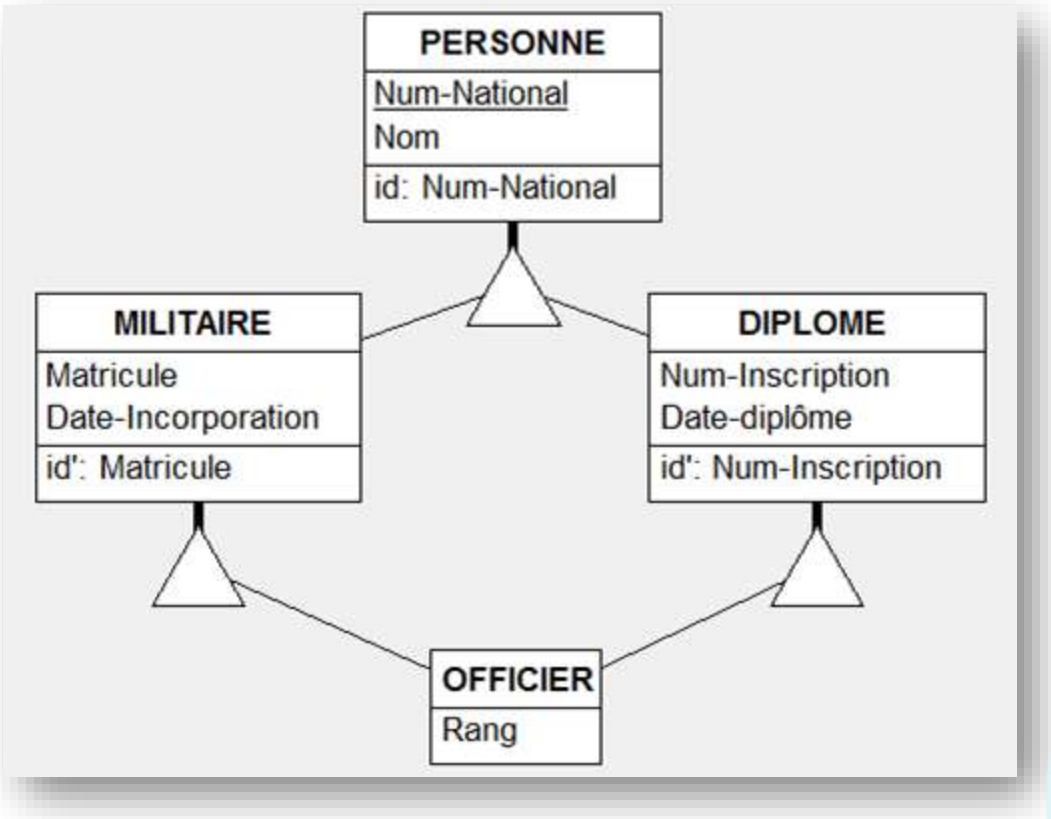
- On peut utiliser un ou plusieurs attributs de la classe parent pour spécifier la classe de spécialisation

Spécialisation multiple

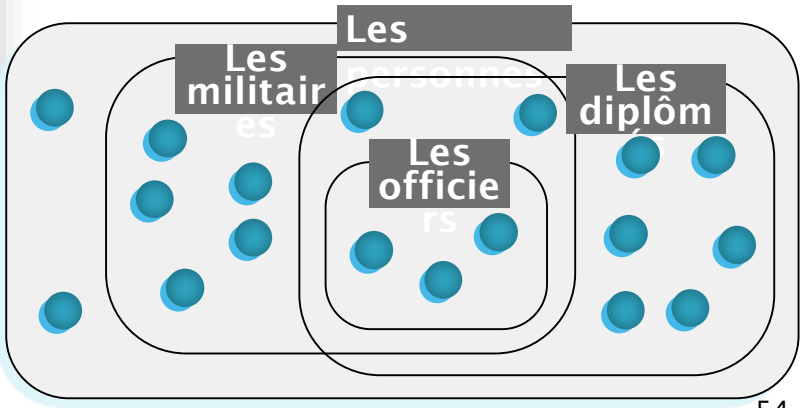


- Une classe d'entités peut à la fois être une classe de spécialisation et une classe de généralisation

Spécialisation multiple & Héritage multiple

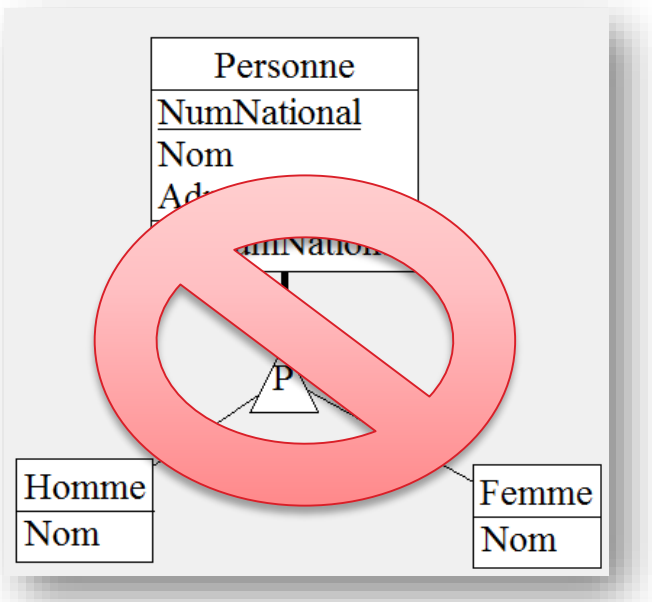


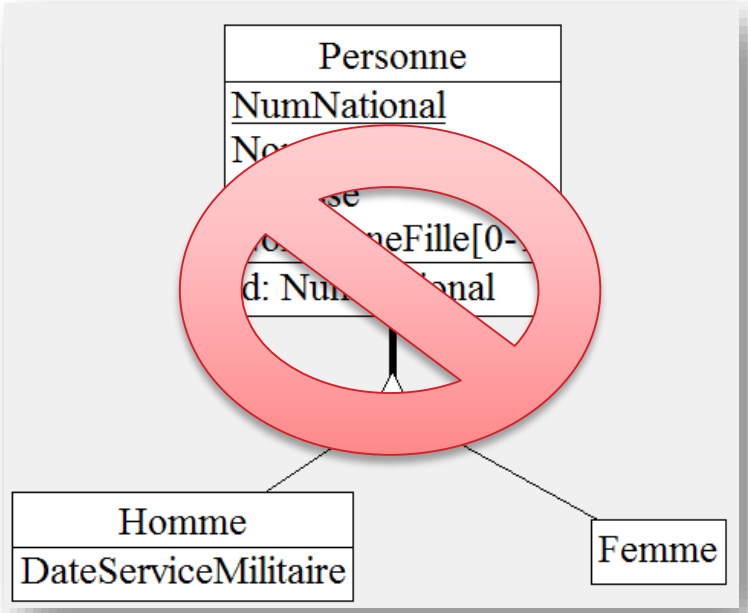
- Des officiers sont à la fois des personnes spécialisées comme militaires et des personnes classifiées comme diplômés

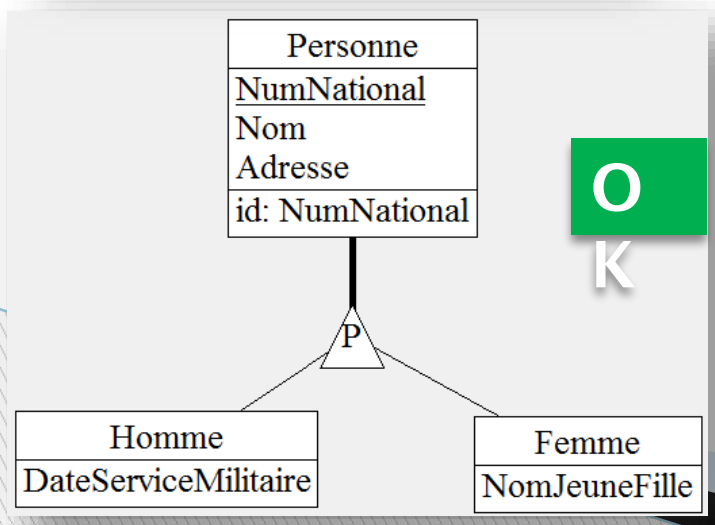


- Des militaires diplômés peuvent ne pas être des officiers

Mauvaise utilisation de la relation de Spécialisation – Généralisation

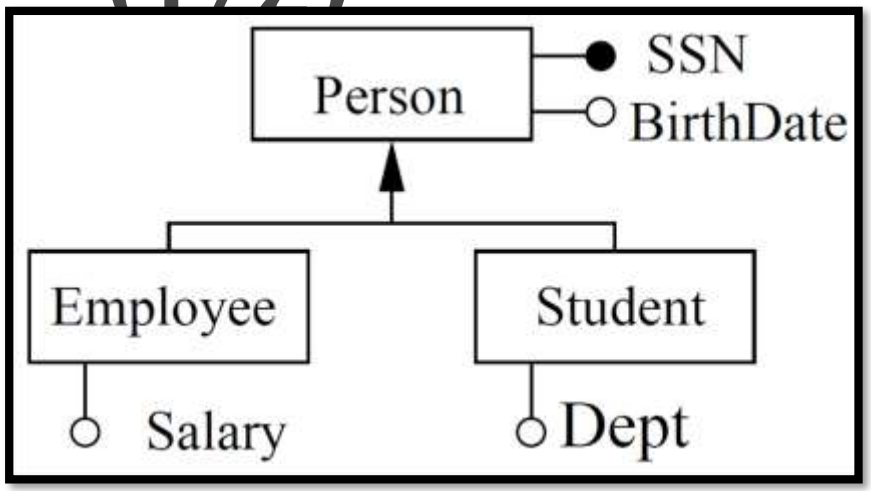


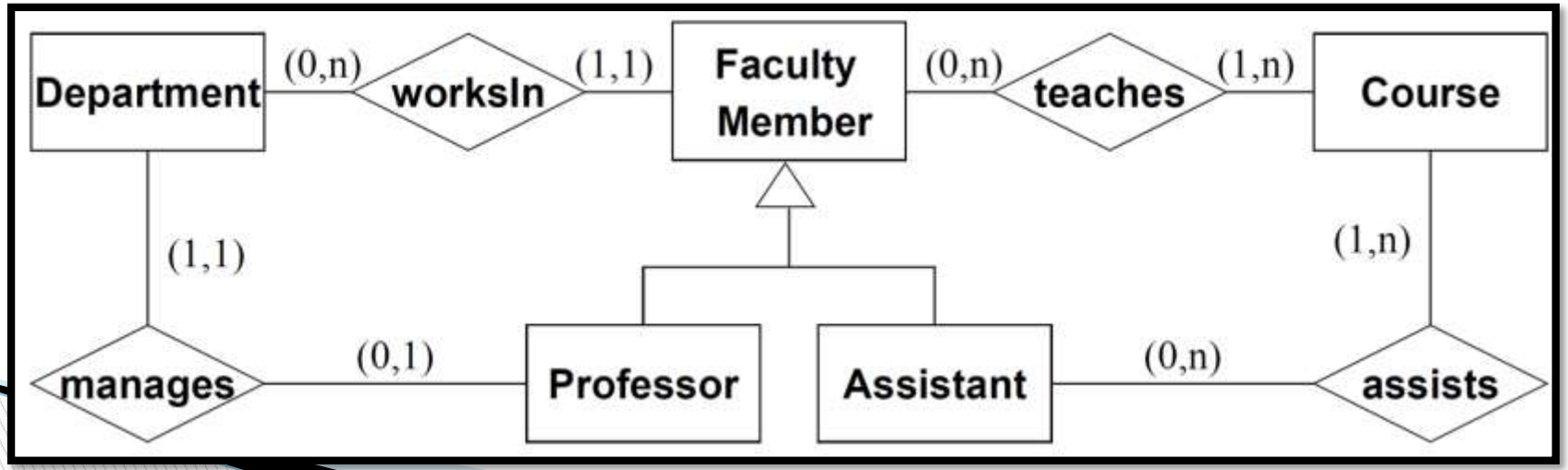




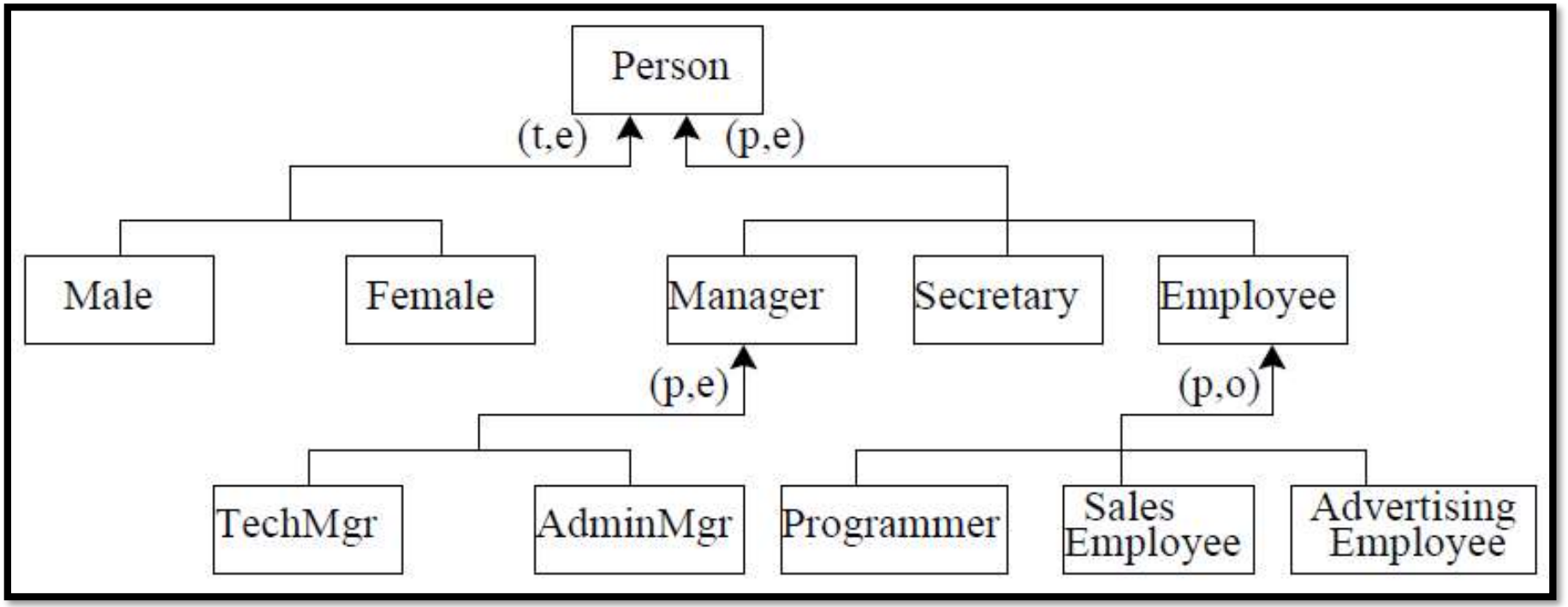
- *Attention*: ne pas confondre une relation d'Association avec une relation de Spécialisation – Généralisation!

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





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

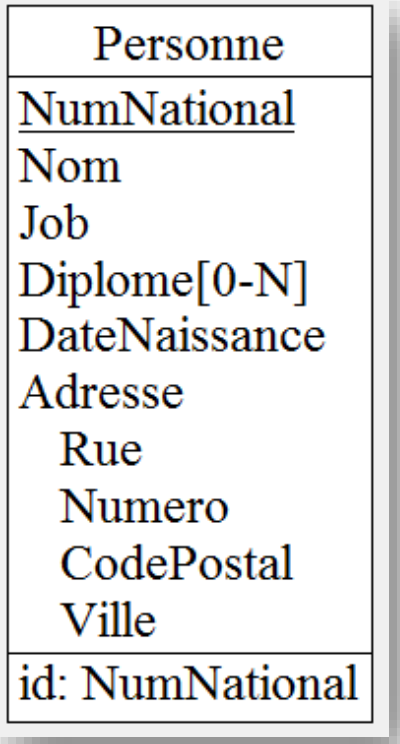


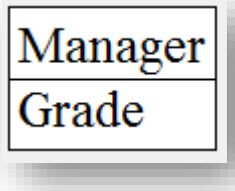
Chapitre 2: Le modèle Entité-Association

6. Les contraintes d'intégrités

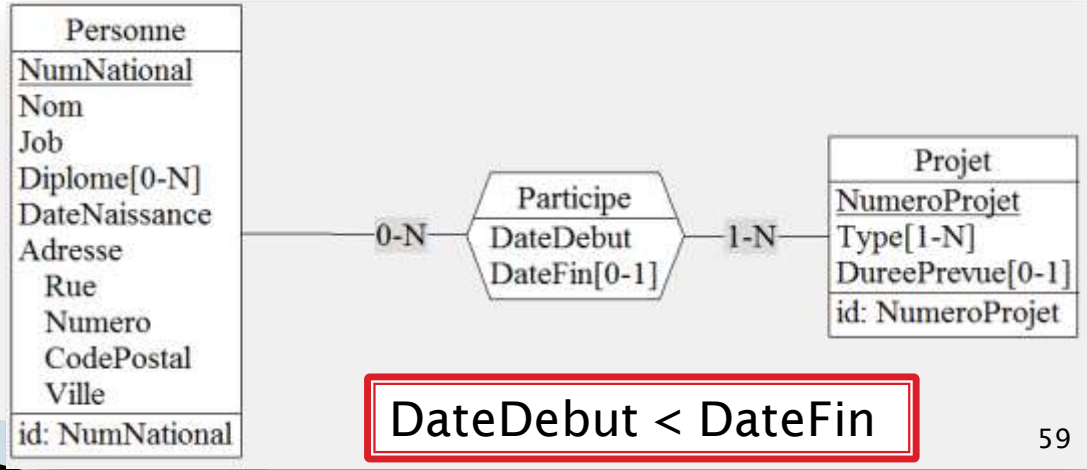
Les contraintes d'intégrités

- 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



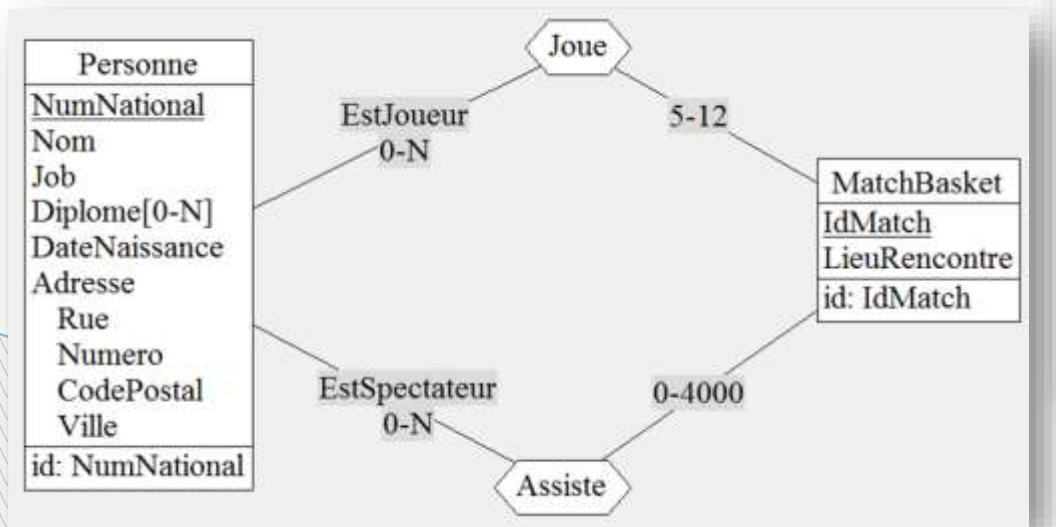


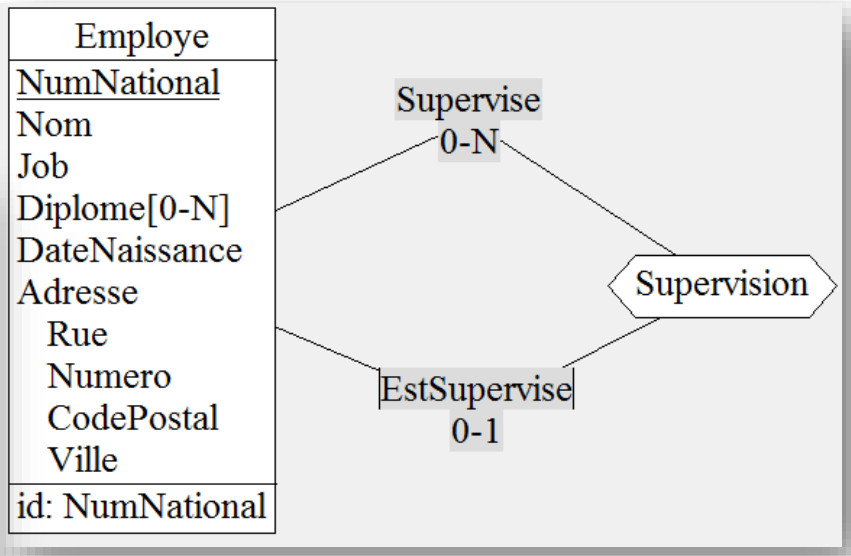
Grade = {"Low", "Middle", "High"}



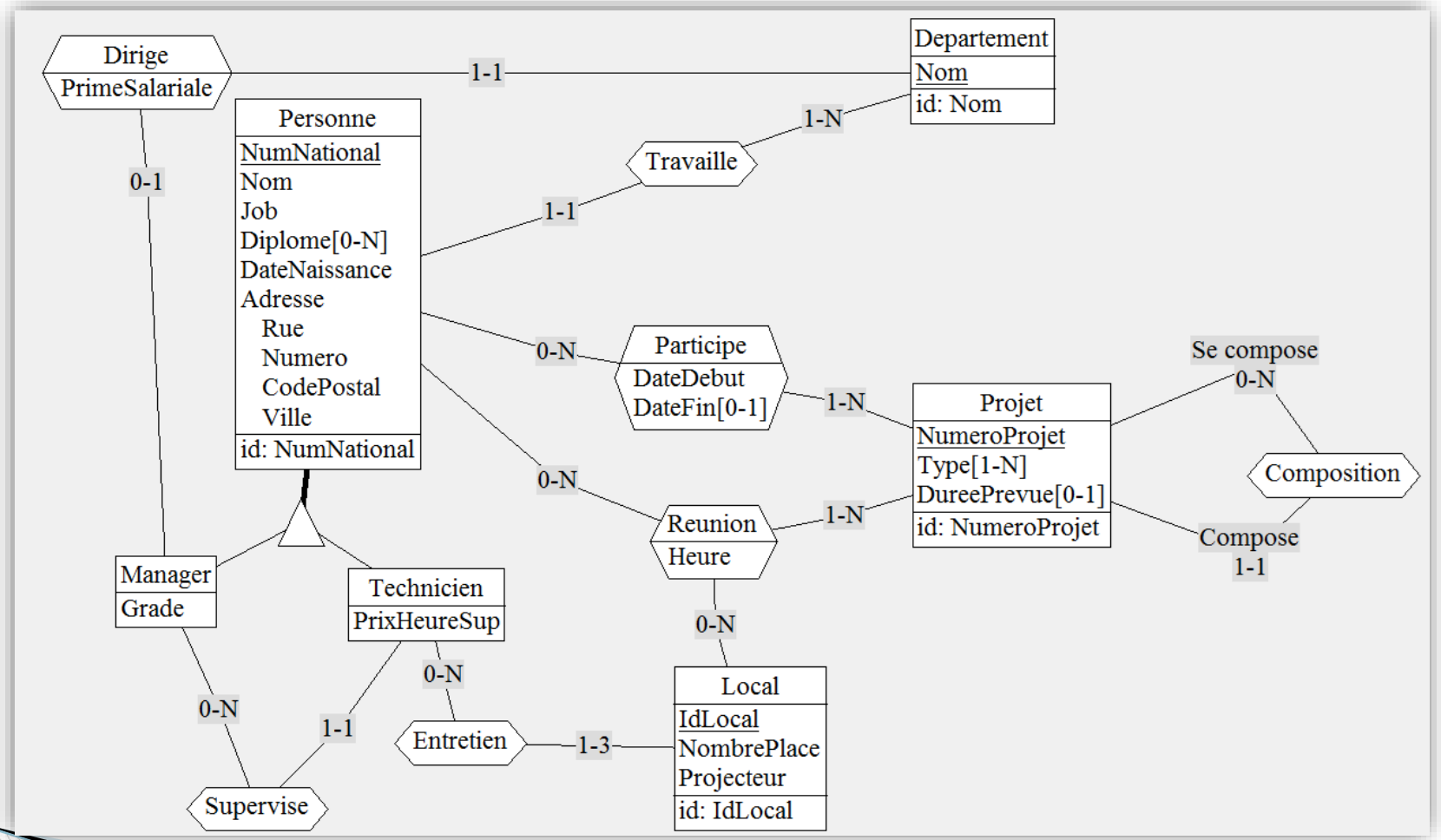
Les cycles dans les Associations (Récursives)

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

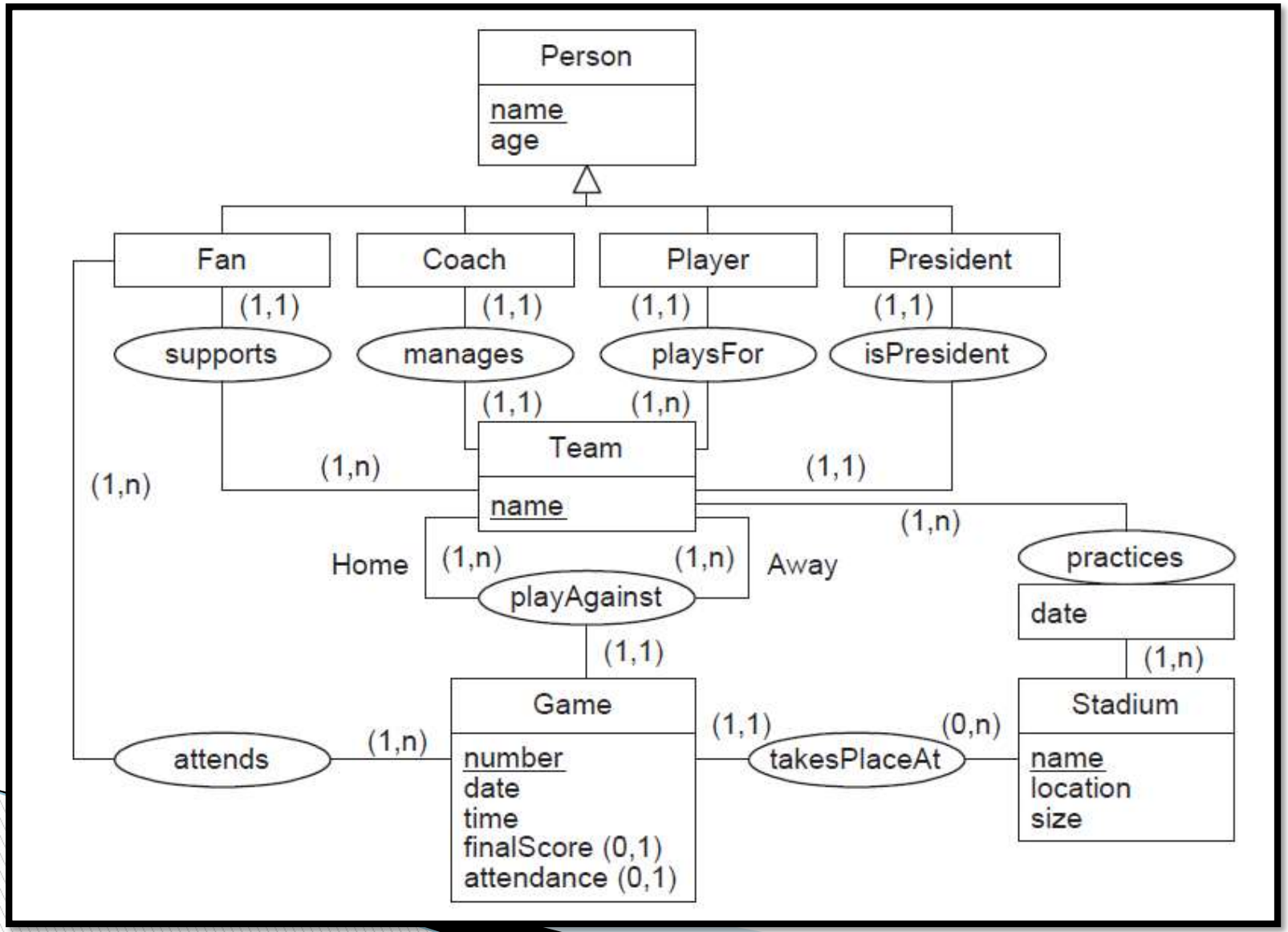




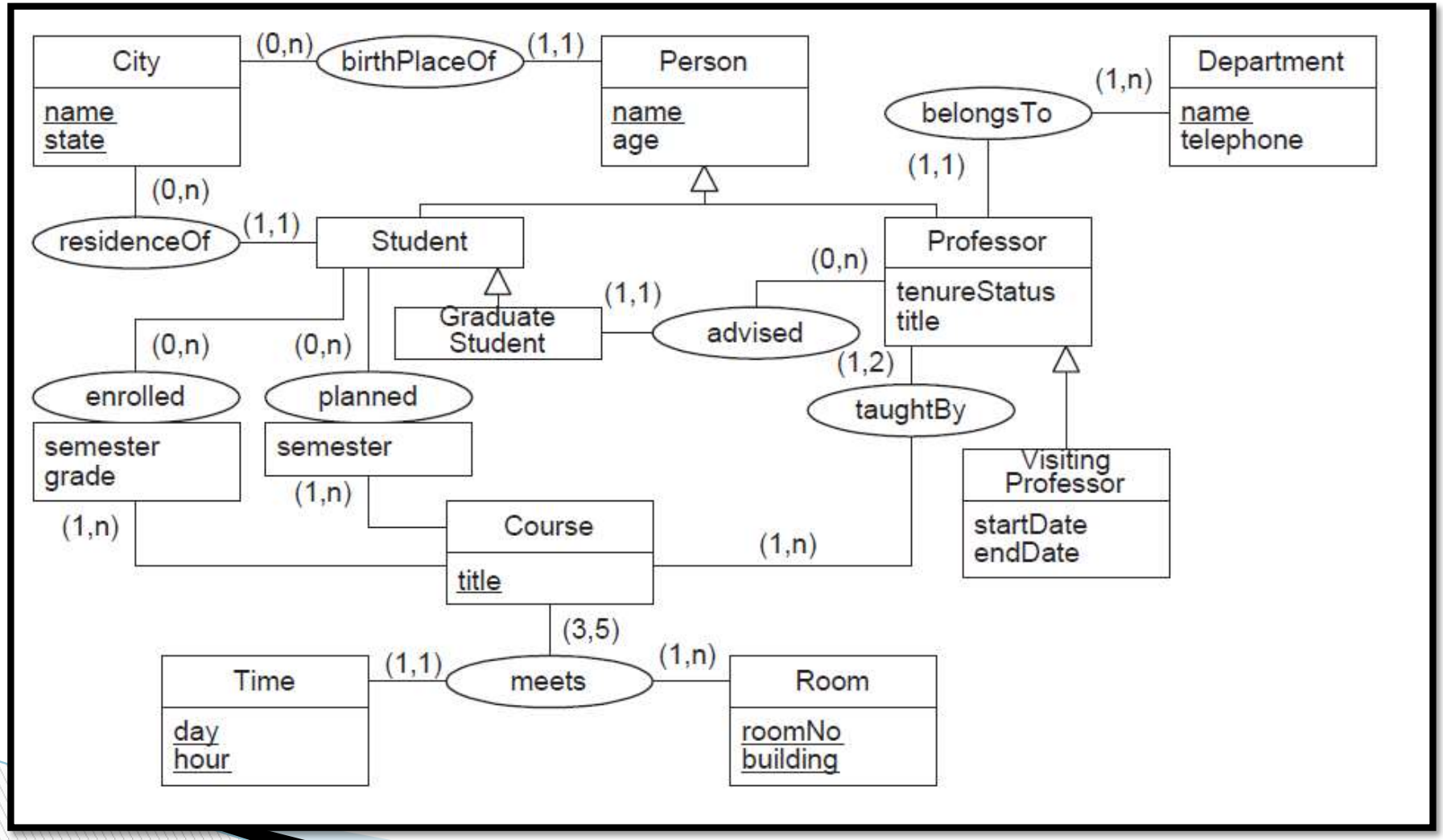
Lire un Schéma E-A (1/3)



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

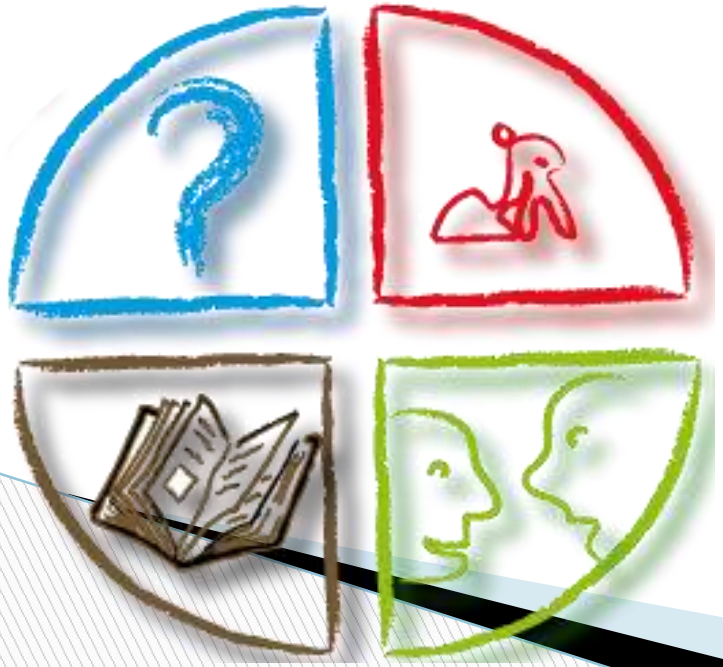


Table des matières

- 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

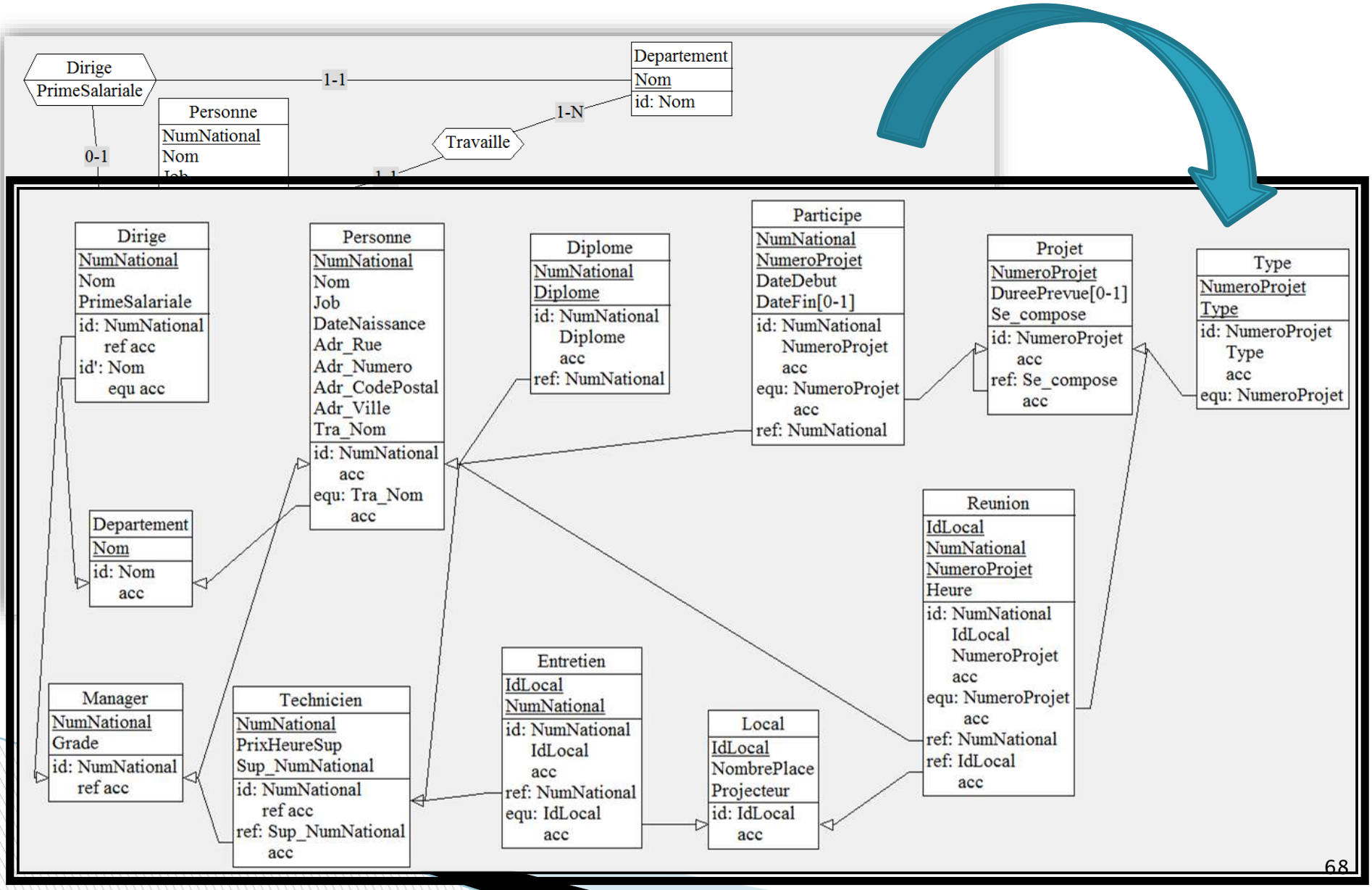
1. Le Schéma Relationnel: définition et concept

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!!

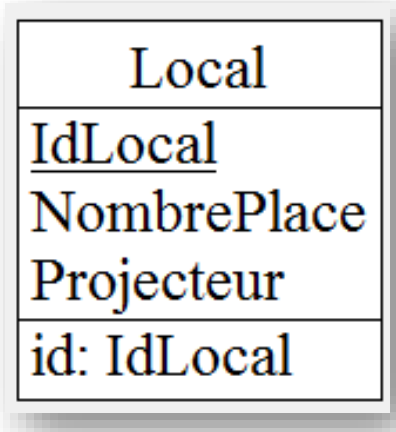
Chapitre 3: Le Schéma Relationnel

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		
IdLocal	NombrePlace	Projecteur

Traduire un attribut multiple ou composite

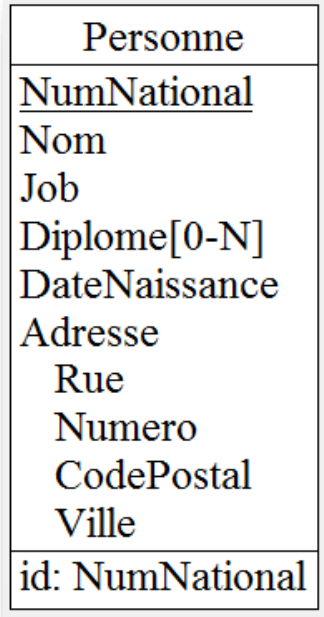
- Un attribut multiple est transféré dans une nouvelle table et référencé à l'aide du mécanisme de clé primaire – clé étrangère
- Une attribut composite est désagrégé

Personne

NumNational	Nom	Job	DateNaissance	AdRue	AdNumero	AdCodePostal	AdVille

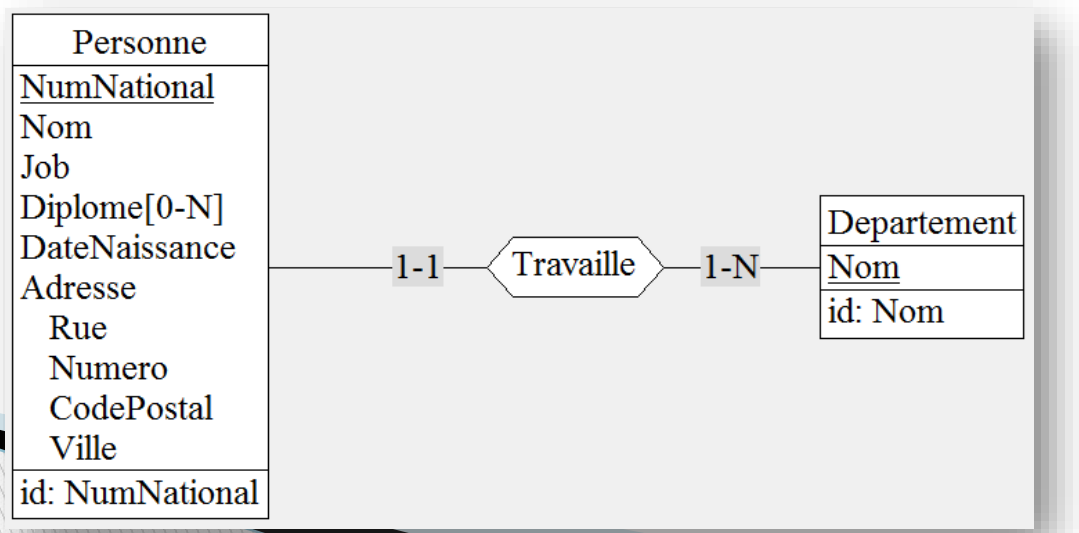
Diplome

NumNational	IntituléDiplome

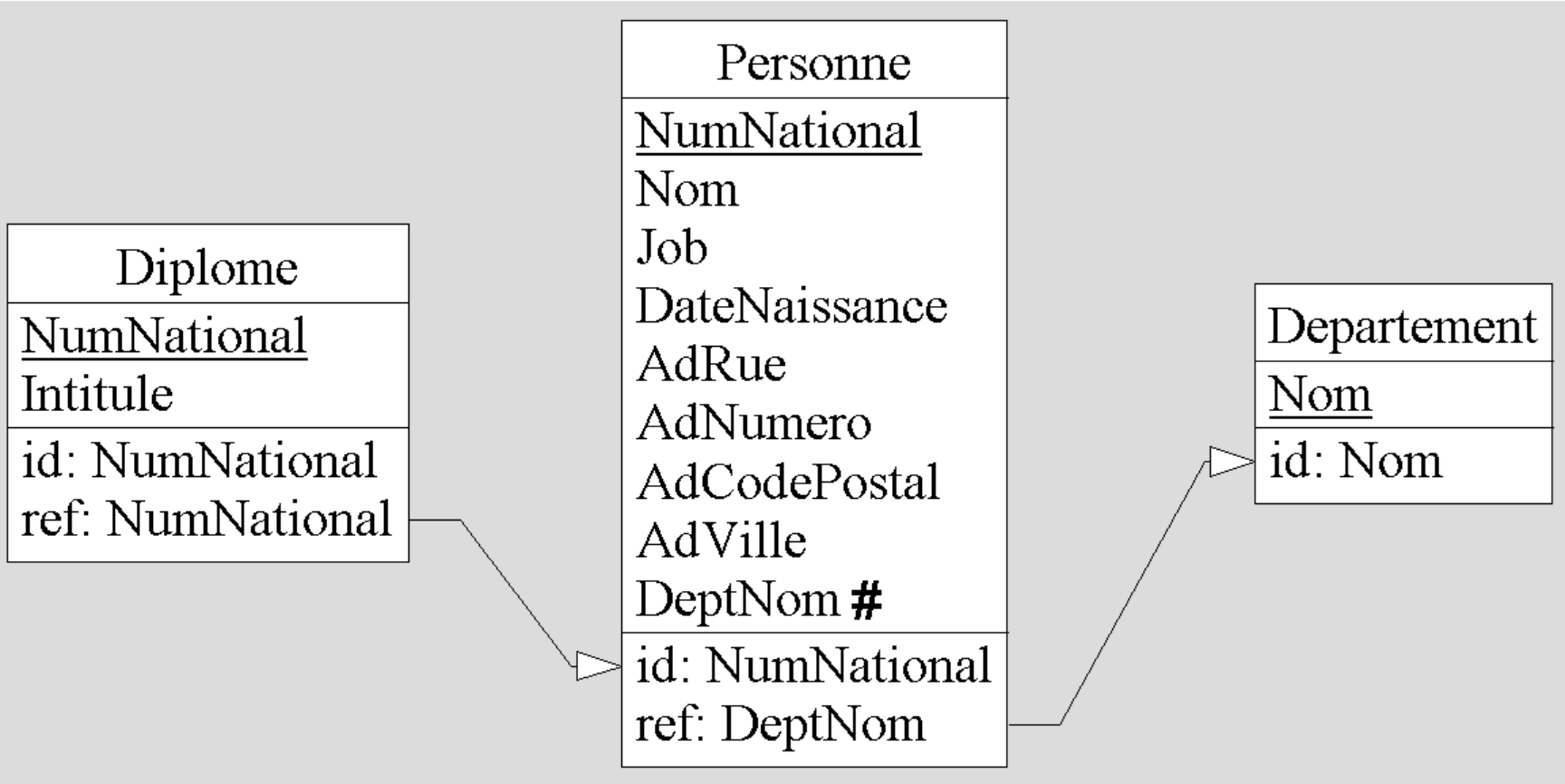


Traduire une Association binaire One-to-Many

- 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 référencie 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)

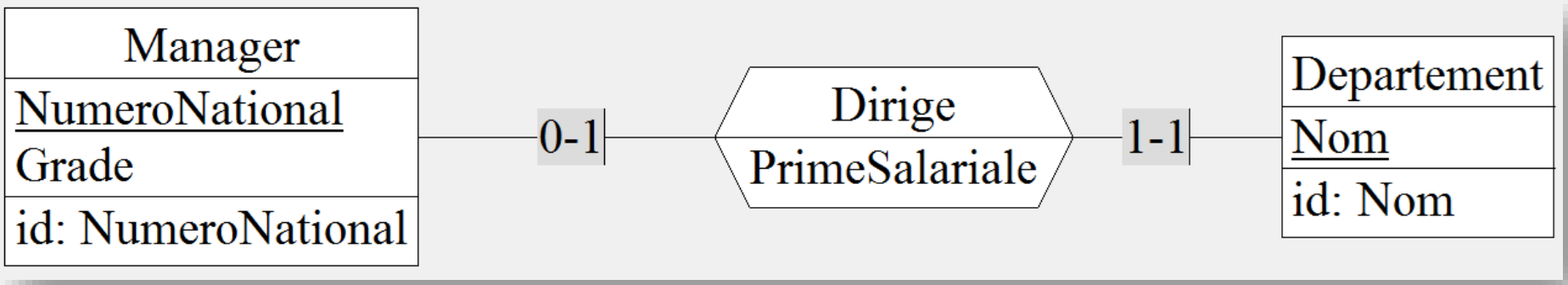


Traduire une Association binaire One-to-Many



Traduire une Association binaire One-to-One

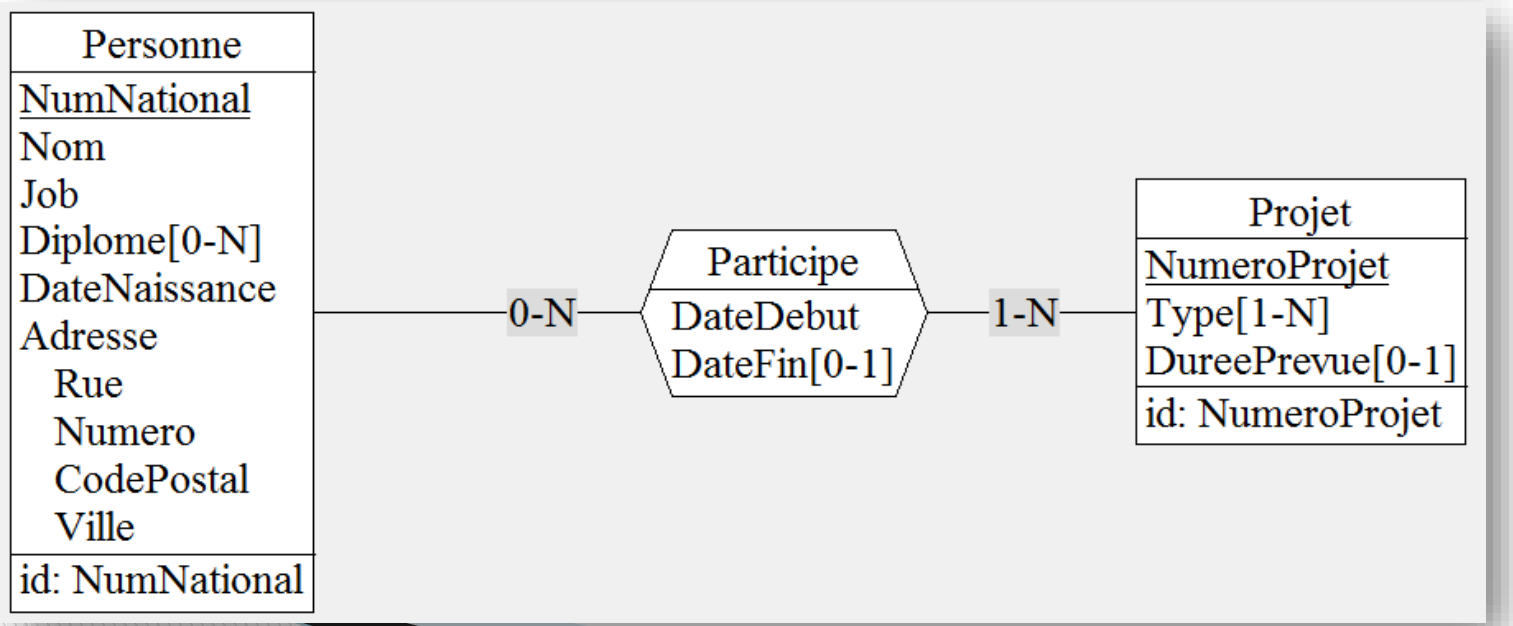
- Une association binaire one-to-one est traduite comme une association de type one-to-many
- On privilégie toujours la table du côté de la cardinalité obligatoire, c-à-d que la clé étrangère est ajouté du côté de la cardinalité obligatoire



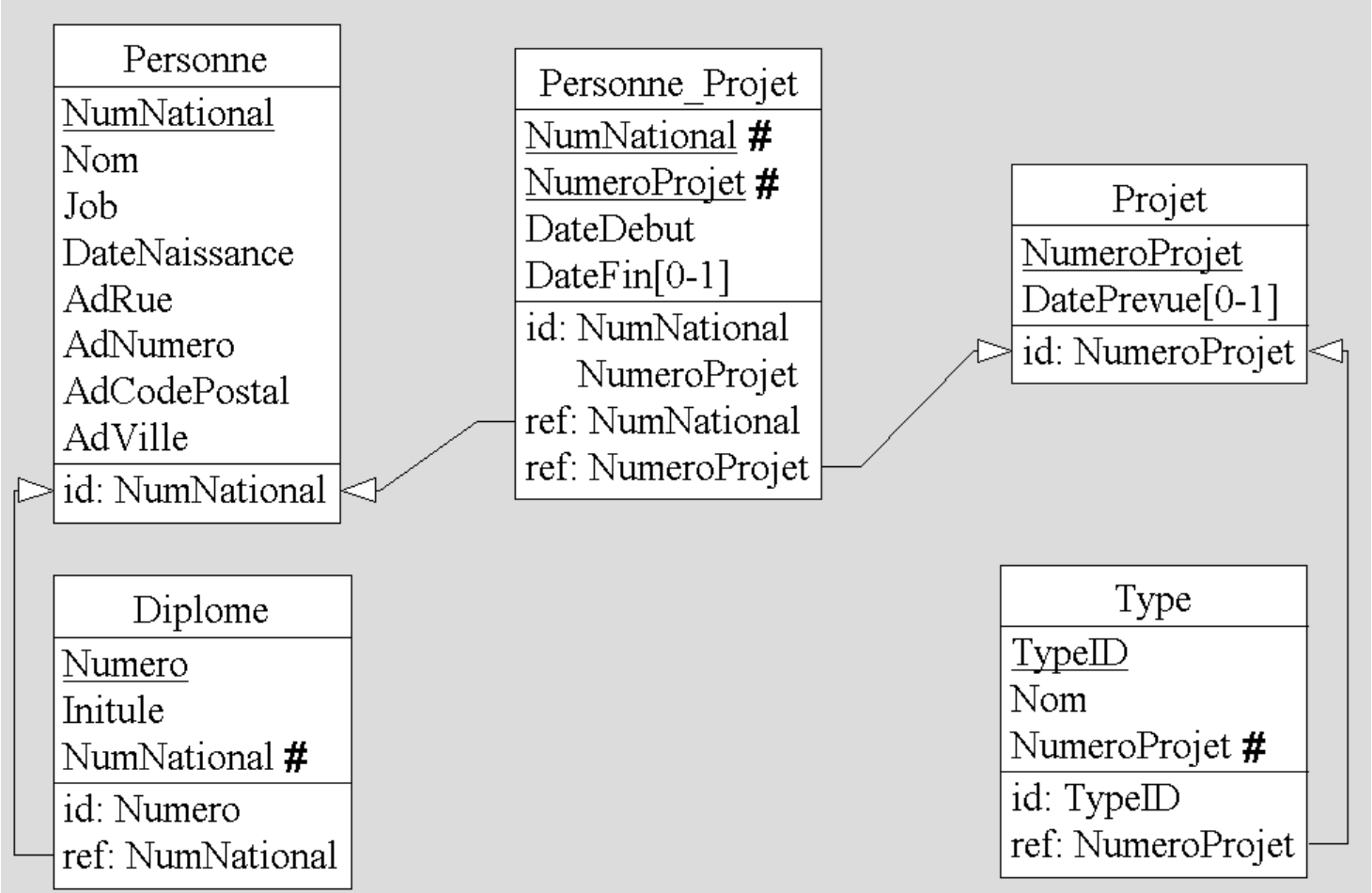
Traduire une Association binaire One-to-One

Traduire une Association binaire Many-to-Many

- Une association binaire de type many-to-many disparait 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

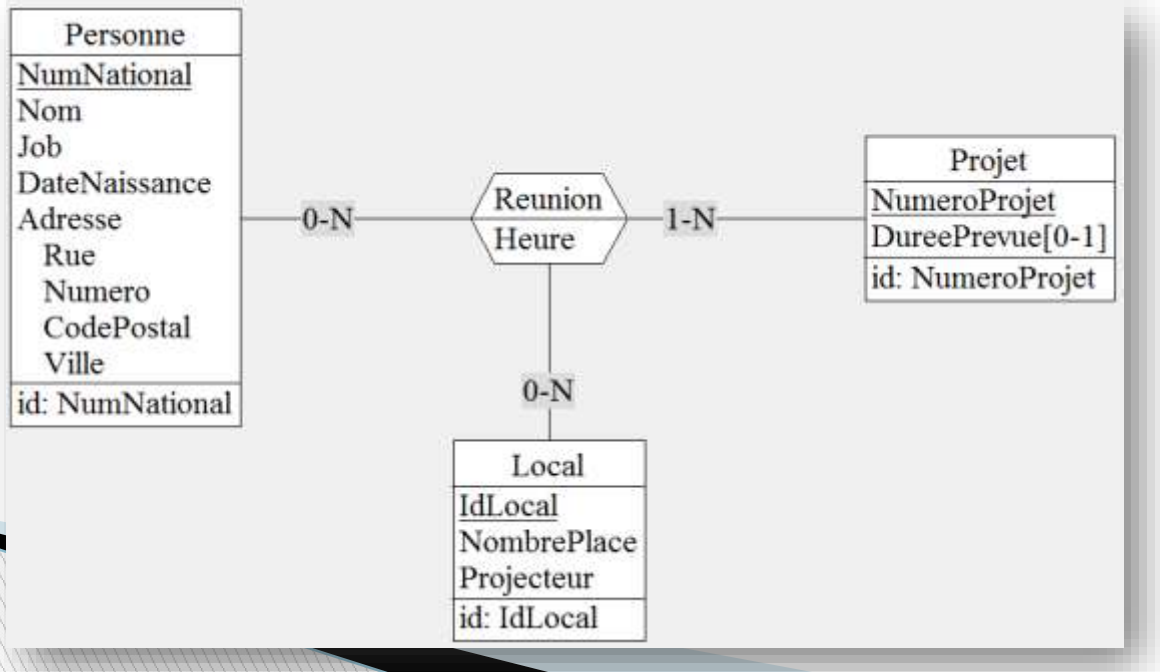


Traduire une Association binaire Many-to-Many



Traduire une Association n-aire

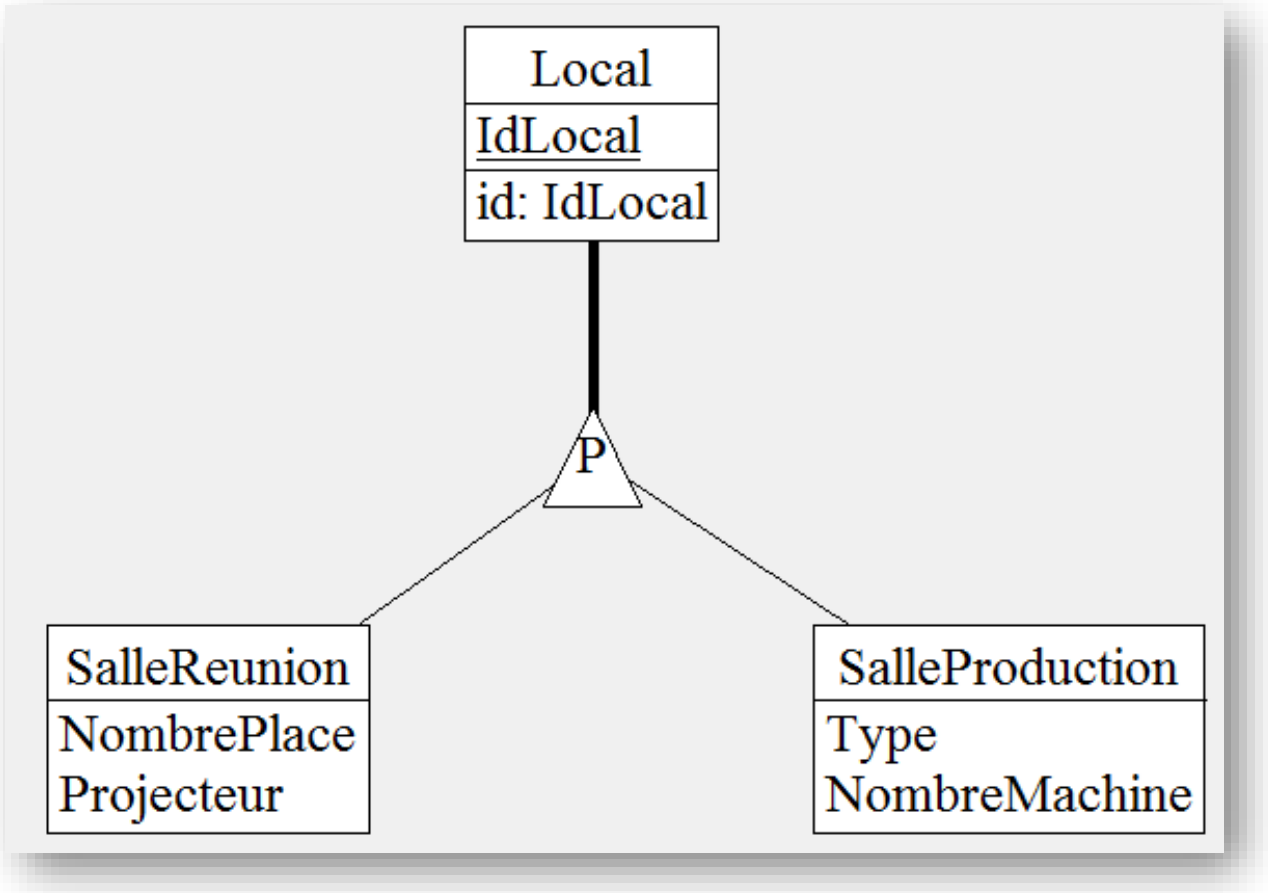
- Une association non-binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que de classes d'entités en association
- Les attributs éventuels de l'association deviennent les colonnes de cette nouvelle table



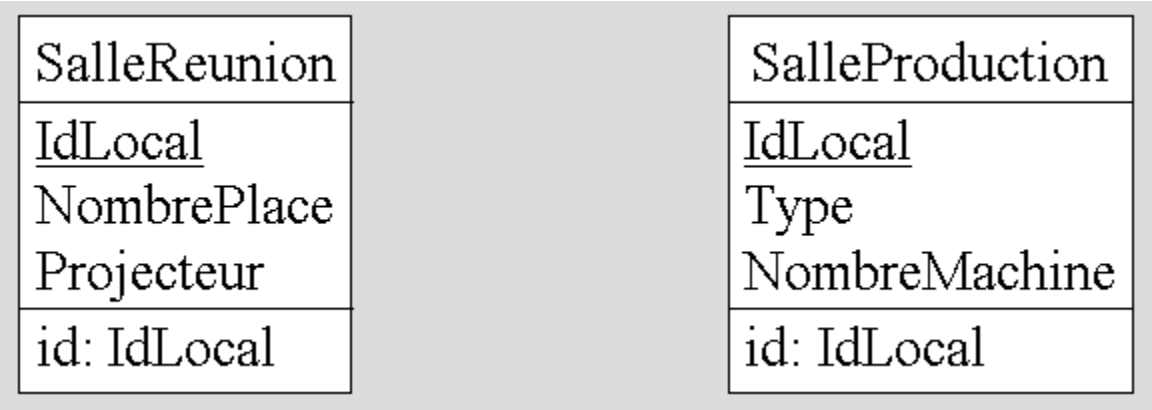
Traduire une relation de spécialisation – Généralisation

- 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*

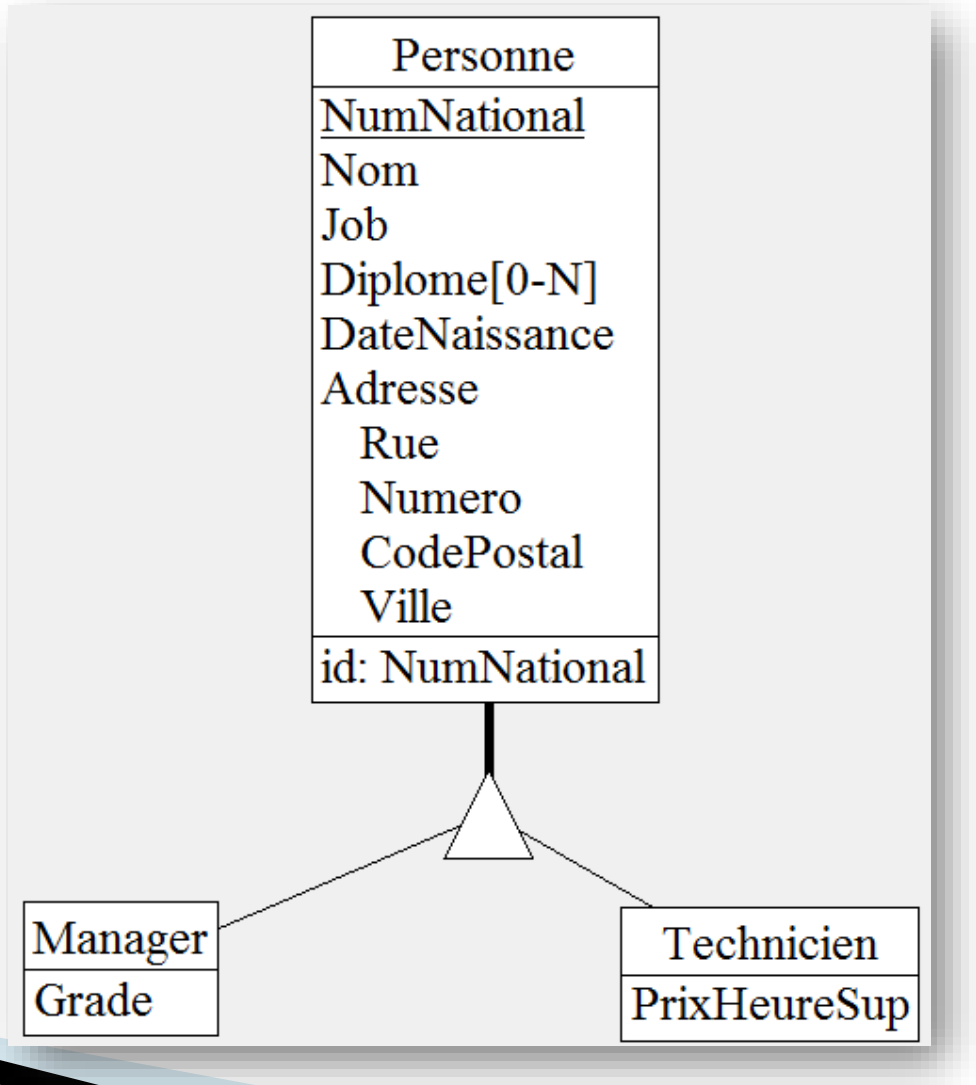
Traduire une spécialisation Totale



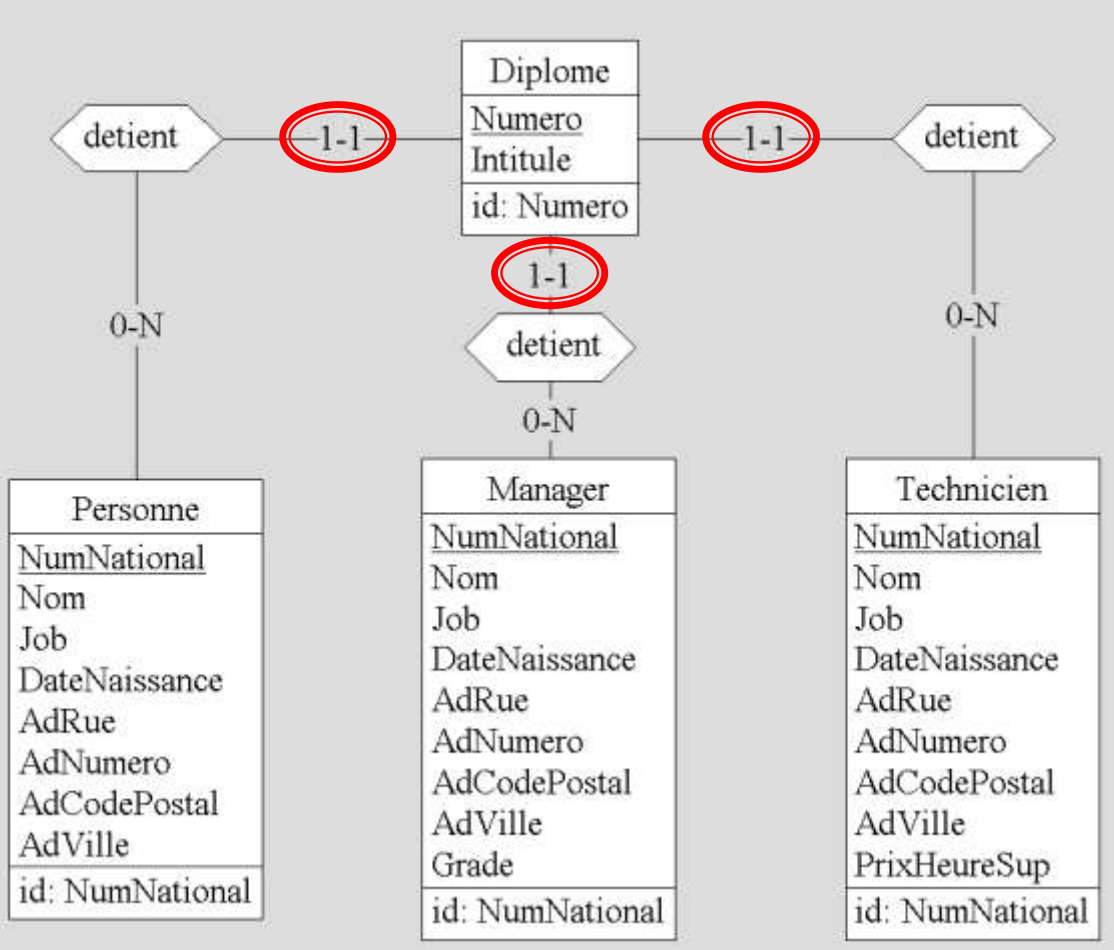
Traduire une spécialisation Totale



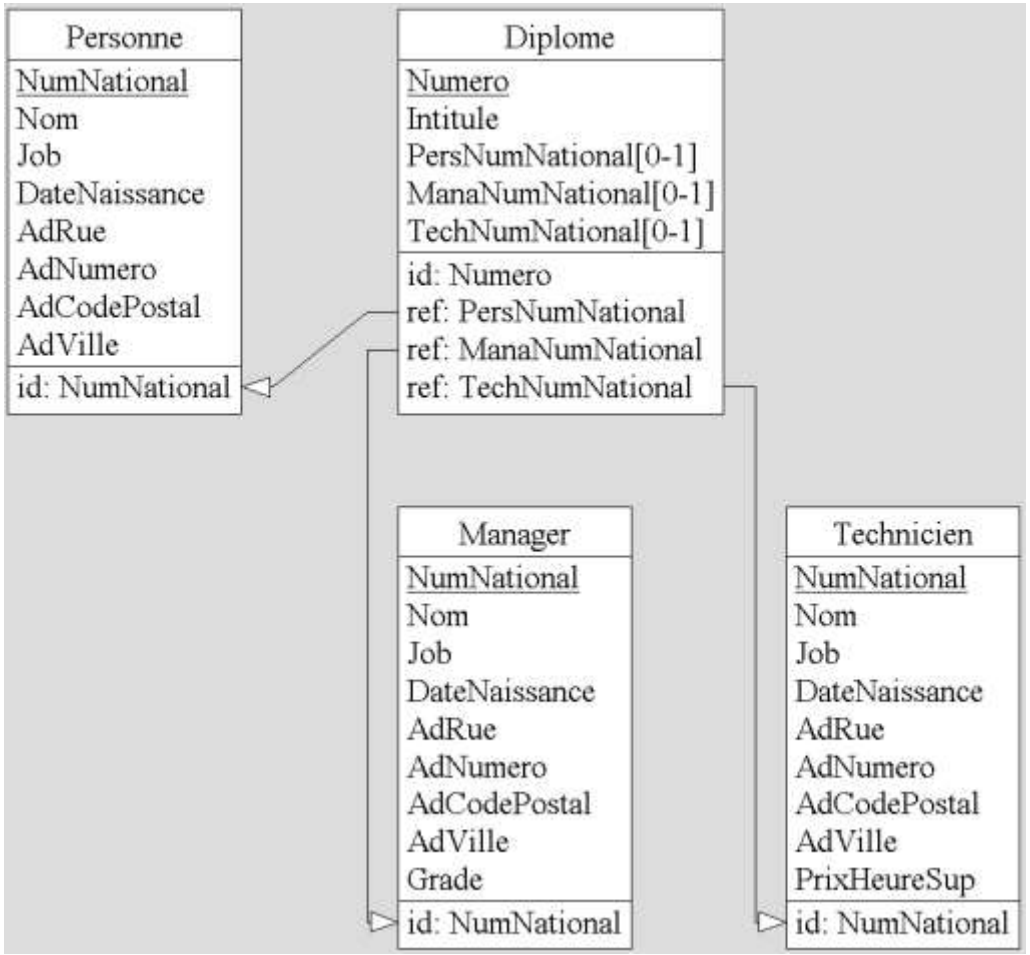
Traduire une spécialisation Partielle



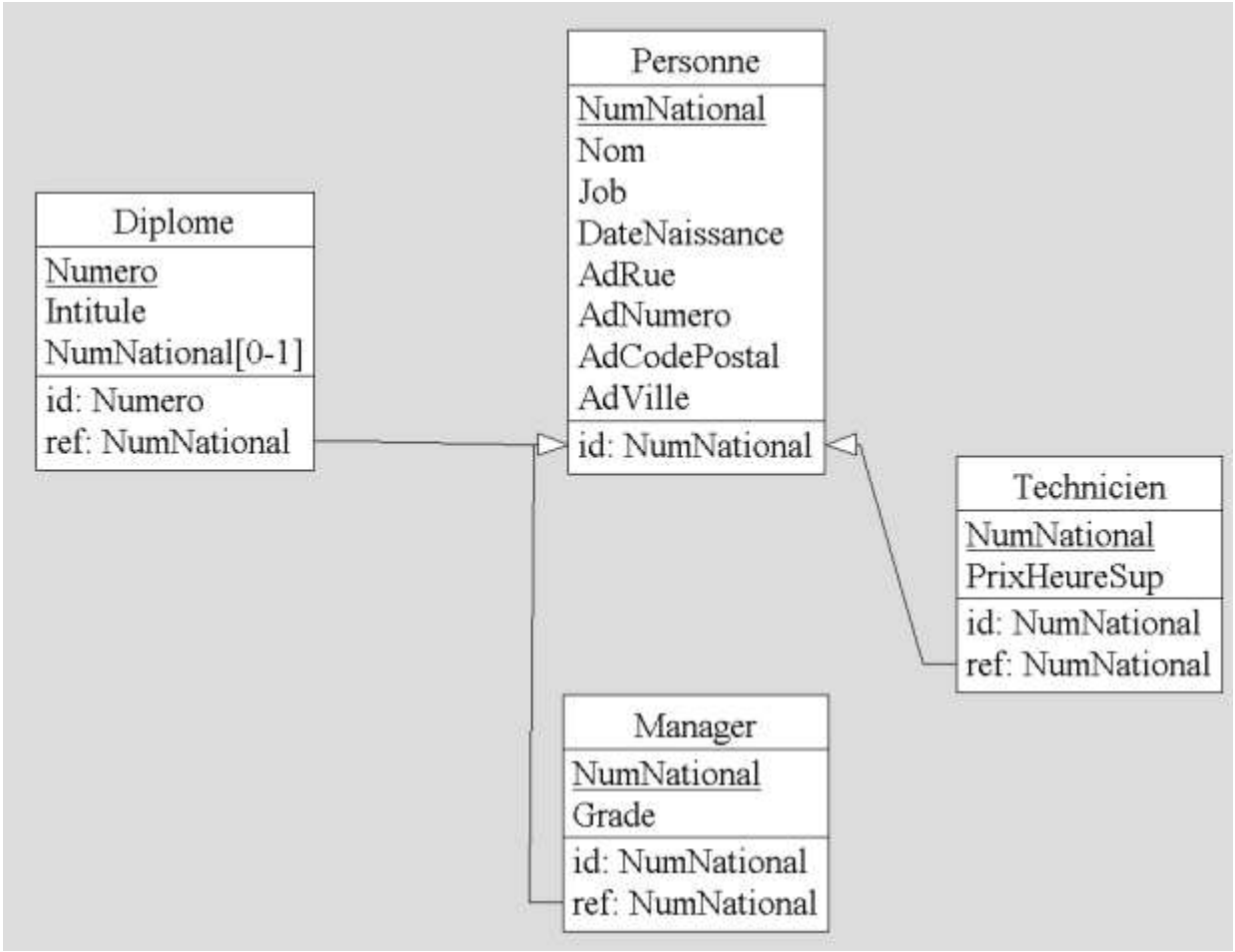
Traduire une spécialisation Partielle



Traduire une spécialisation Partielle



Traduire une spécialisation Partielle



Exercices

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