



Base de données

Modèle
Entité/Association

L3 Informatique

Antoine Spicher
antoine.spicher@u-pec.fr

Contexte du cours

■ Organisation du cours

- 1^{ère} partie (C. D.)
 - Modèle et algèbre relationnel
 - Langage SQL (essentiellement les requêtes : SELECT ... FROM ... WHERE ...)
- 2^{ème} partie (A. S.) : éléments pour la conception d'une base de données
 - Diagrammes E/A, traduction E/A → modèle relationnel (, forme normale)
 - Langage SQL (création et modification de table)

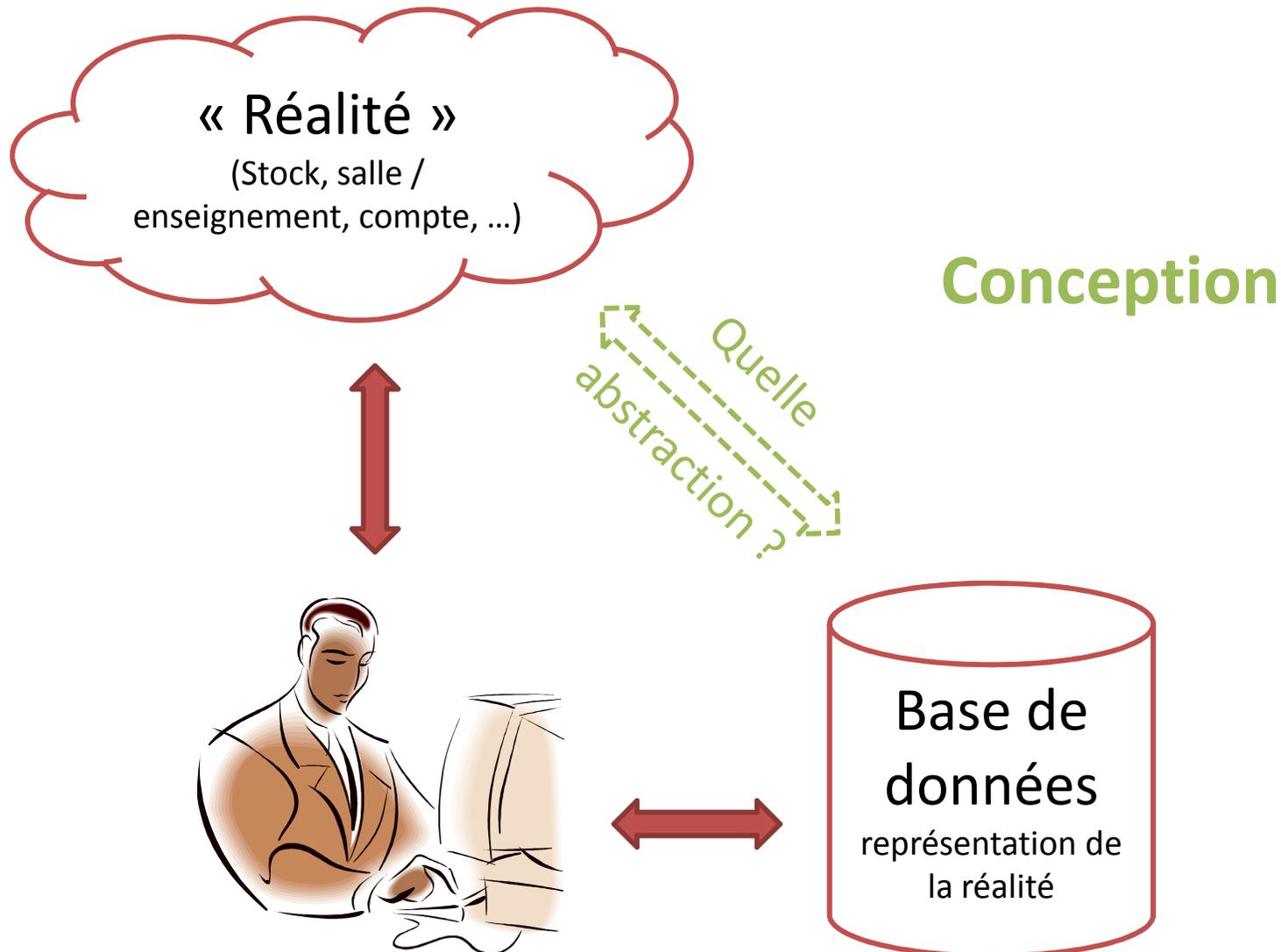
■ Ressource principale

Database System Concepts

A. Silberschatz & H. F. Korth & S. Sudarshan, 6^{ème} édition, 2010

Conception d'une base de données

■ Big Picture...



Conception d'une base de données

■ Processus complexe

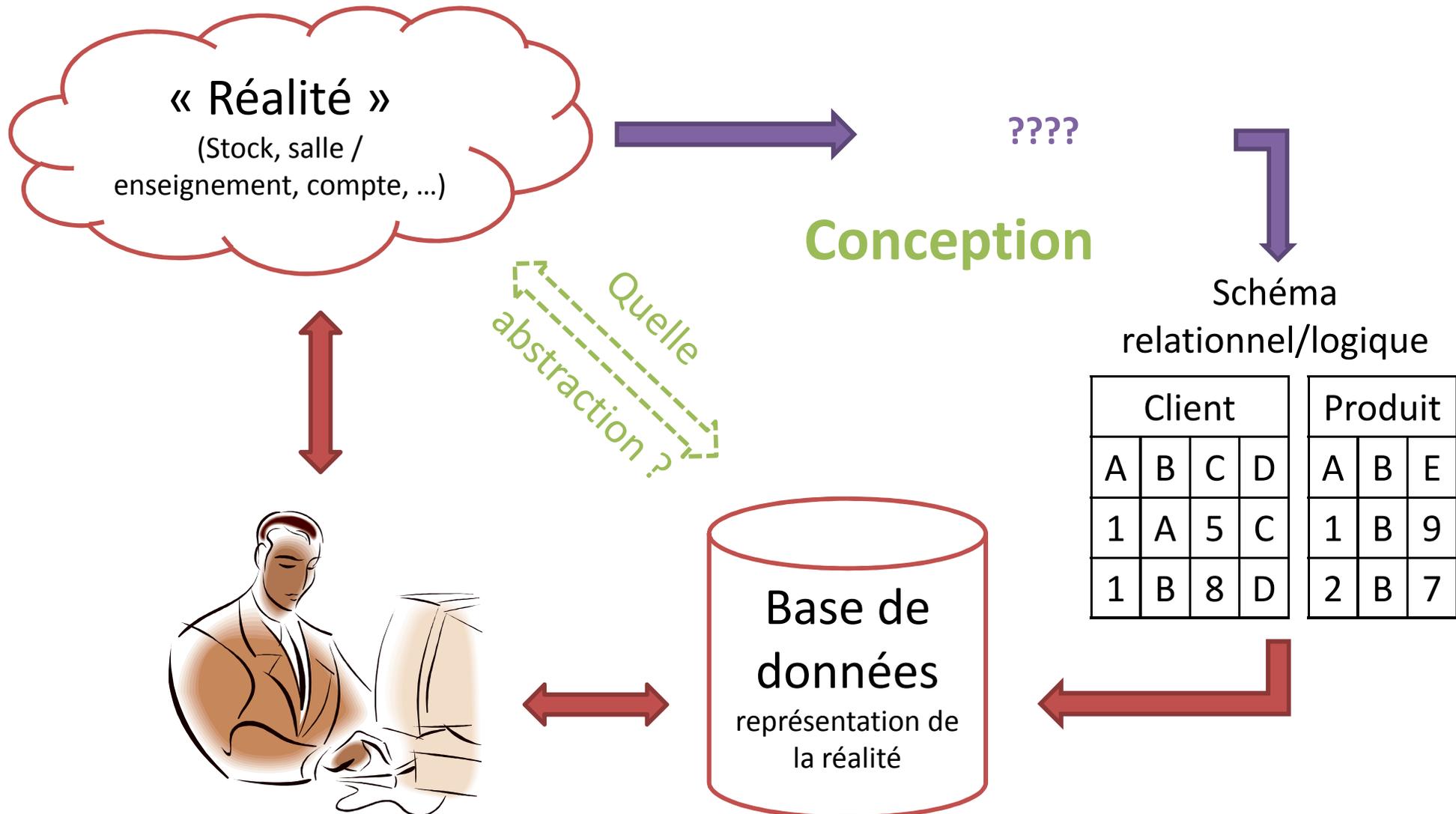
- Conception du schéma (*i.e.*, des tables)
- Conception des programmes accédant (lecture/écriture) aux données
- Conception du schéma de sécurité pour le contrôle d'accès (vues, permissions, etc.)

■ Quels outils ?

- Cas simple : utilisation directe du modèle relationnel
- Cas complexe (le plus courant)
 - Modèle relationnel : bas niveau (*i.e.*, proche de la BD)
 - peu compréhensible pour le non-initié (non-informaticien)
 - peu adapté à la représentation de relations complexes (utilisateur)
 - peu structuré, hiérarchisé (maintenance et évolutivité)
 - Nécessité d'un **modèle de haut-niveau**

Conception d'une base de données

■ Big Picture...



Conception d'une base de données

■ Étapes de développement d'un SIA

□ Aspects fonctionnels (**spécification**)

- Caractériser *complètement* les données
- Interaction avec les experts du domaine concerné et les (futurs) utilisateurs
- Support : en général textuel parfois diagrammatique (*e.g.*, UML : use case)

□ Élaboration d'une structure (**conception**)

- Choix d'un modèle de données adapté, traduction des spécifications
- Prise en compte de toute la spec.
- Éviter les pièges : redondance et incomplétude
- Support : présentation diagrammatique (**E/A**, UML, etc.)

□ Implantation (**réalisation**)

- Transformation de la vue abstraite et haut-niveau de la conception dans le modèle de données de l'implantation (*e.g.*, **E/A** → **schéma relationnel**)
- Réalisation physique de la base de données

■ Cf. cours de développement de programmes

Conception d'une base de données

■ Big Picture...

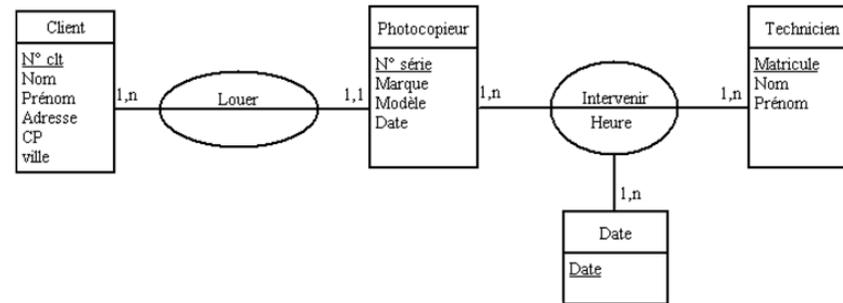


Schéma conceptuel
Entités/Associations

Conception

Schéma
relationnel/logique

Quelle
abstraction ?

Client				Produit		
A	B	C	D	A	B	E
1	A	5	C	1	B	9
1	B	8	D	2	B	7



Conception d'une base de données

■ Choix du modèle de données

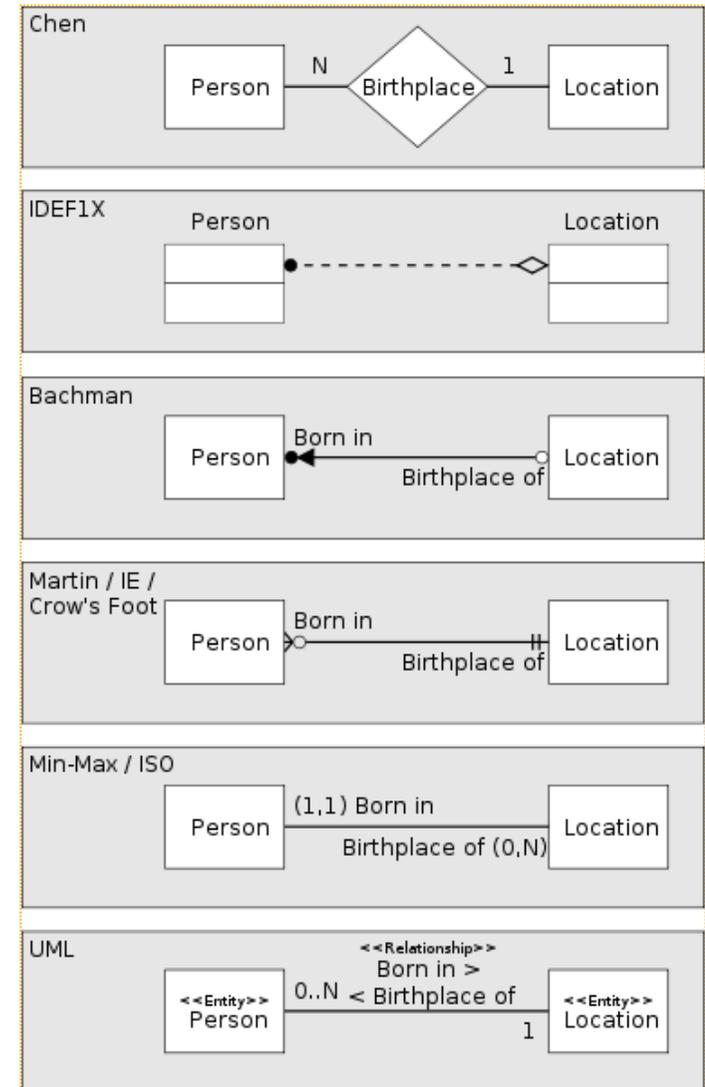
- Il en existe un grand nombre
- Entités/Associations (E/A) et UML

■ Modèle de données E/A

- Repose sur 3 notions
entité, association (ou relation¹) et attribut
- Il n'y a pas réellement de standard graphique
- Conventions de *Database System Concepts*

■ Cas d'UML

- Intersection commune importante avec E/A ...
... mais avec quelques différences
- Référence à UML quand cela est possible



(source : Wikipédia – Entity-relationship model)

¹ Attention à ne pas confondre avec la notion de relation en algèbre relationnelle...

Plan



- Introduction à la conception de BD
- Modèle entités/associations, concepts de base
- Modèle entités/associations étendu
- Quelques problématiques de conception

Modèle E/A, les bases

■ Origine

- Idées dans l'air du temps des années 1970
 - Représentation abstraite et conceptuelle des données
 - Représentation globale de la structure logique d'une BD
 - Modèle de données sémantique
- Peter Chen (1976)

The Entity Relationship Model: Toward a Unified View of Data



Peter Chen
(source : computer.org)

■ Trois concepts fondamentaux ...

- (Ensembles/classes d') *Entités*
 - « chose/objet » du monde réel distinct de tout autre objet
- (Ensembles/classes d') *Associations* : lien entre les « choses/objets »
- *Attributs* : propriétés descriptives d'une entité ou d'une association

■ ... et leur représentation diagrammatique

Modèle E/A, les bases – Entités

■ Notion d'*entité*

- Définition : « Une **entité** est une chose ou un objet du monde réel **identifiable sans ambiguïté** »
- Entité \equiv ensemble de propriétés appelées **attributs**
 - une valeur est associée à chaque attribut
 - un sous-ensemble des attributs, la **clef**, caractérise l'entité de façon unique

■ Notion d'*ensemble d'entités*

- Définition : « ensemble d'entités partageant les mêmes propriétés »
- Entité vs. ensemble d'entités \equiv objet vs. classe

Modèle E/A, les bases – Entités

■ Exemples d'entités

□ Université

l'UFR de science et techno., l'UE de BD, la salle P1-015, etc.

□ Allociné

MK2 Bibliothèque, Charlize Theron, UGC les Halles, *Star Wars épisode I*, *Avatar*, Jason Statham, Gaumont Opéra, etc.

■ Exemples d'ensembles d'entités

□ Université

■ les UEs (intitulé, nombre d'ECTS, référence Apogée, ...)

■ les salles (nom, nombre de places, ...)

■ les étudiants (numéro, nom, date de naissance, adresse, ...)

□ Allociné

■ les cinémas (nom, adresse, nombre de salles, ...)

■ les acteurs (nom, date de naissance, nationalité, ...)

■ les films (titre, année, ...)

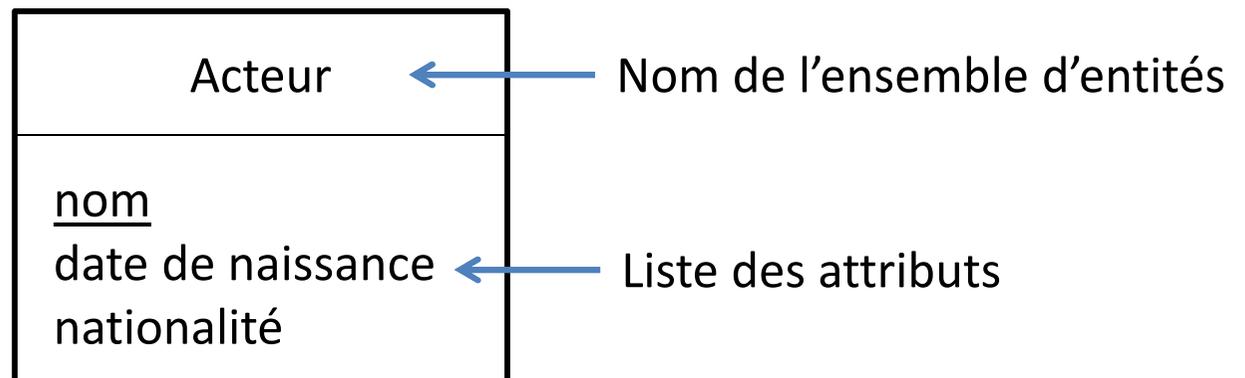
Modèle E/A, les bases – Entités

■ Représentation graphique

- Des entités, par extension, au sein de tables

nom	date de naissance	nationalité	...
Charlize Theron	07/08/1975	Afrique du Sud	...
Jason Statham	12/09/1972	Angleterre	...

- D'un ensemble d'entités (modélisation des données)
 - Équivalente à la représentation des classes en UML
 - Les membres représentent les attributs
 - Pas d'opération



Modèle E/A, les bases – Attributs

■ Point de vue formel

- Propriétés descriptives d'une entité (ou d'une association)

$$e \in E \subseteq D_1 \times D_2 \times \dots \times D_n$$

- Un **attribut** est une *fonction* associant une valeur à une entité

$$A_i : E \rightarrow D_i \cup \{\text{null}\}$$

- D_i : domaine de valeurs de l'attribut A_i
- Cas de la valeur spéciale null
 - la valeur de l'attribut n'existe pas
 - la valeur de l'attribut existe mais n'est pas connue

■ Types d'attribut

- Attribut *simple* ou *composite*
 - **Simple** : la valeur est un objet atomique ne pouvant être décomposé
 - **Composite** : la valeur de l'attribut est structurée

Modèle E/A, les bases – Attributs

■ Point de vue formel

- Propriétés descriptives d'une entité (ou d'une association)

$$e \in E \subseteq D_1 \times D_2 \times \dots \times D_n$$

- Un **attribut** est une *fonction* associant une valeur à une entité

$$A_i : E \rightarrow D_i \cup \{\text{null}\}$$

- D_i : domaine de valeurs de l'attribut A_i
- Cas de la valeur spéciale null
 - la valeur de l'attribut n'existe pas
 - la valeur de l'attribut existe mais n'est pas connue

■ Types d'attribut

- Attribut simple ou composite
- Attribut *monovalué* ou *multivalué*
 - **Monovalué** : l'attribut à une seule valeur pour une entité (cardinalité de 1)
 - **Multivalué** : l'attribut à plusieurs valeurs à la fois (cardinalité *)

Modèle E/A, les bases – Attributs

■ Point de vue formel

- Propriétés descriptives d'une entité (ou d'une association)

$$e \in E \subseteq D_1 \times D_2 \times \dots \times D_n$$

- Un **attribut** est une *fonction* associant une valeur à une entité

$$A_i : E \rightarrow D_i \cup \{\text{null}\}$$

- D_i : domaine de valeurs de l'attribut A_i
- Cas de la valeur spéciale null
 - la valeur de l'attribut n'existe pas
 - la valeur de l'attribut existe mais n'est pas connue

■ Types d'attribut

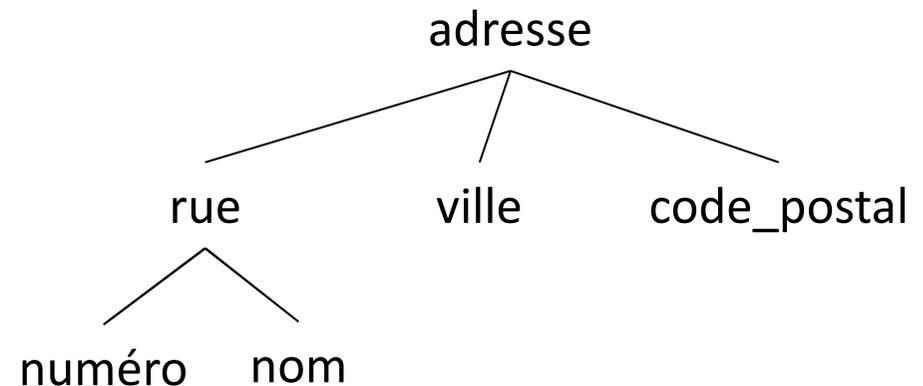
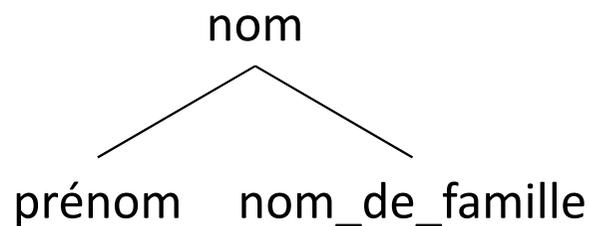
- Attribut simple ou composite
- Attribut monovalué ou multivalué
- Attribut **dérivé**

La valeur de l'attribut est calculée en fonction des autres attributs (cf. UML)

Modèle E/A, les bases – Attributs

■ Exemple

- *Attribut composite* : nom et adresse



- *Attribut multivalué* : numéro de téléphone (num_tel)

Une personne peut avoir plusieurs numéros de téléphone différents ou aucun

nom	num_tel	...
Nathalie	0123461711	...
	0662142315	
Nicolas	null	...

Modèle E/A, les bases – Attributs

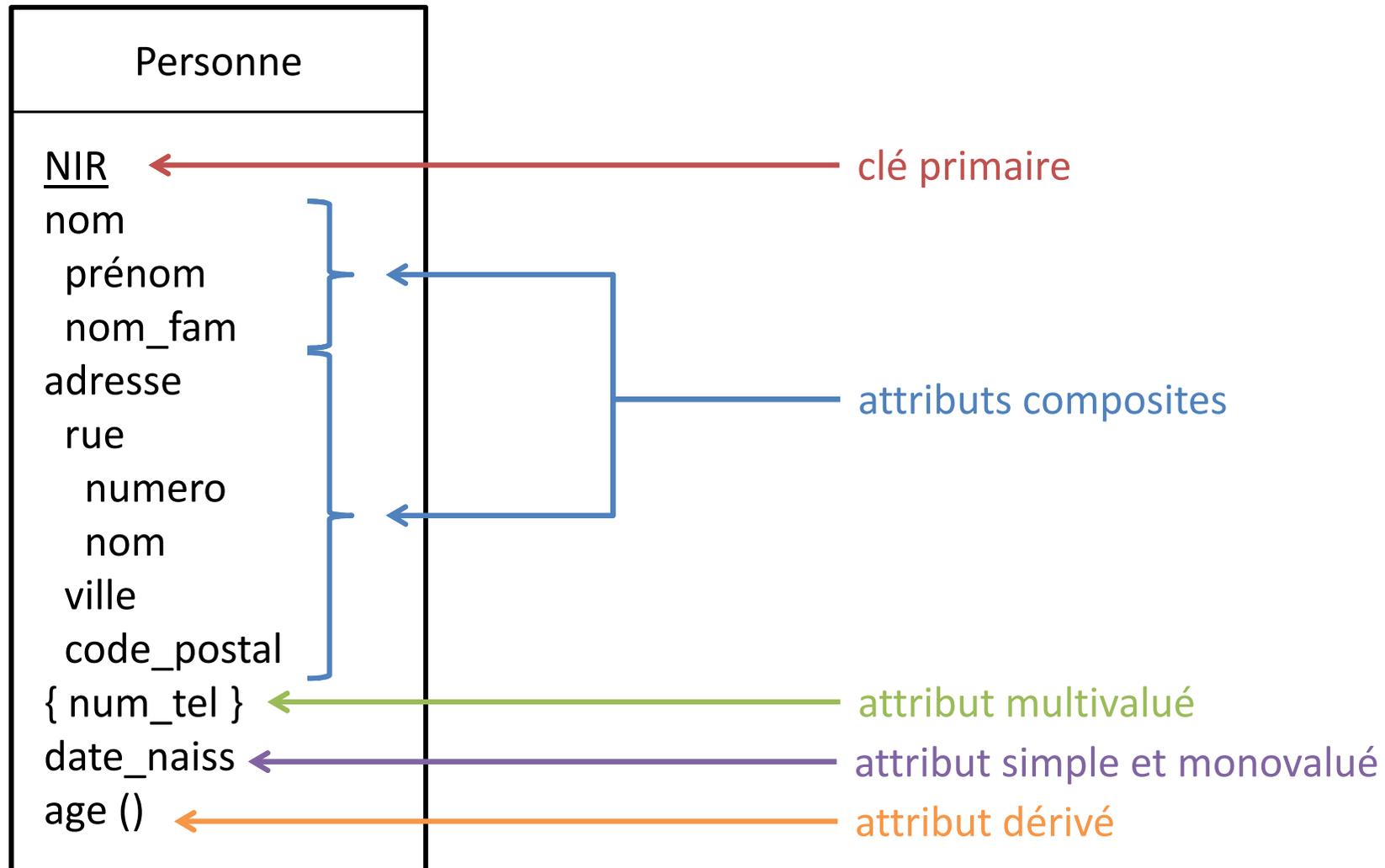
■ Attributs clés

- Hypothèse du modèle E/A : *toutes les entités sont différentes*
Les attributs clés assurent l'unicité des entités
- Quelques définitions
 - **Super clé** (*super key*) d'un ens. d'entités $E \subseteq D_1 \times D_2 \times \dots \times D_n$
 - Sous-ens. d'attributs $K = \{A_{k_1}, \dots, A_{k_p}\}$ distinguant les entités
 - $\forall e_1, e_2 \in E \quad e_1 \neq e_2 \Leftrightarrow \pi_K(e_1) \neq \pi_K(e_2)$
 - **Clé candidate** (*candidate key*) d'un ens. d'entités $E \subseteq D_1 \times D_2 \times \dots \times D_n$
 - Super clé minimale
 - K super clé de $E \Rightarrow \forall K' \subset K$ n'est pas une super clé de E
 - **Clé primaire** (*primary key*) d'un ens. d'entités $E \subseteq D_1 \times D_2 \times \dots \times D_n$
 - Il peut y avoir plusieurs clés candidates
 - *Primaire* : la clé candidate choisie pour la modélisation
- Cf. les notions de PRIMARY KEY et FOREIGN KEY en SQL

Modèle E/A, les bases – Attributs

■ Représentation graphique

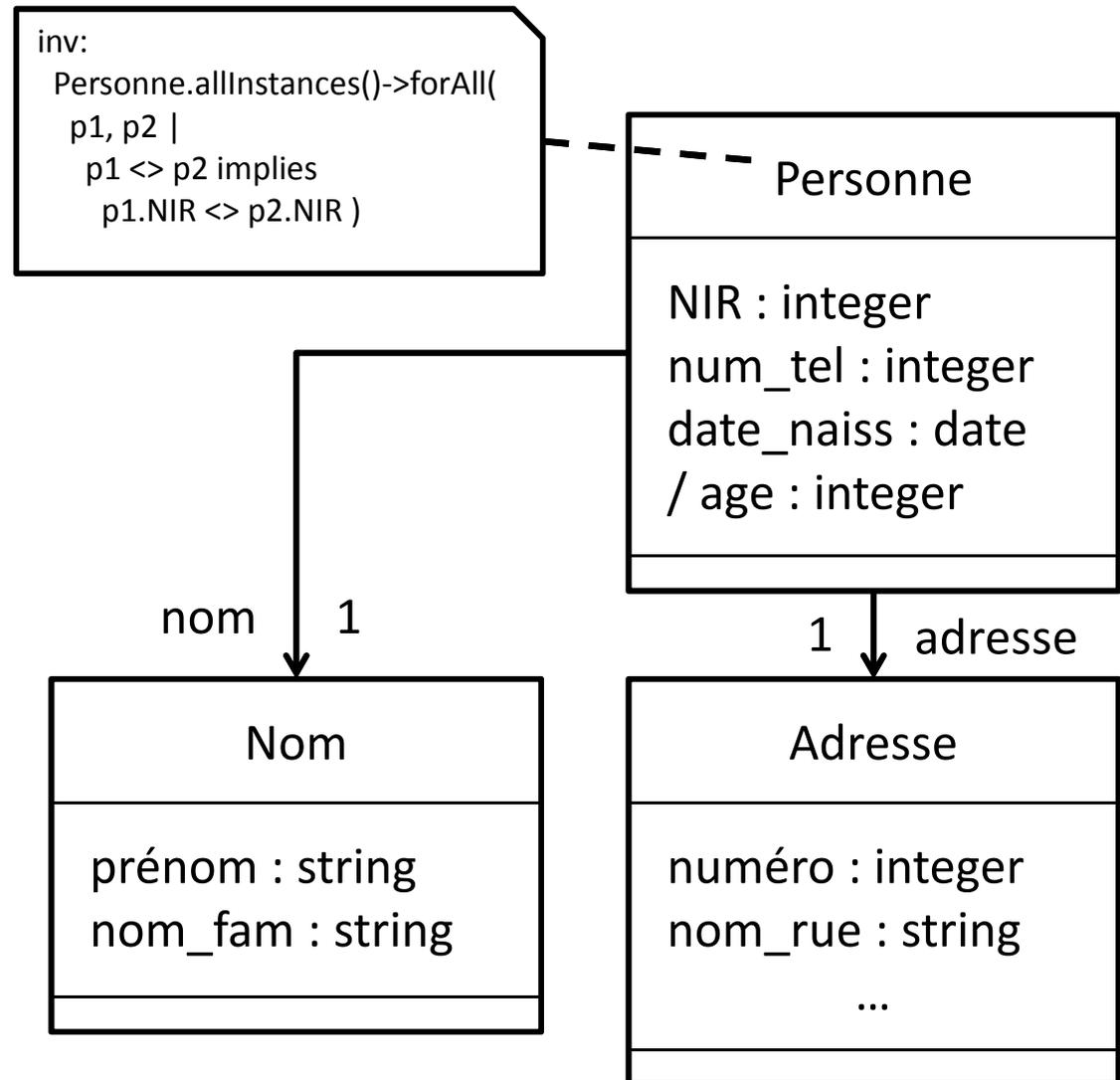
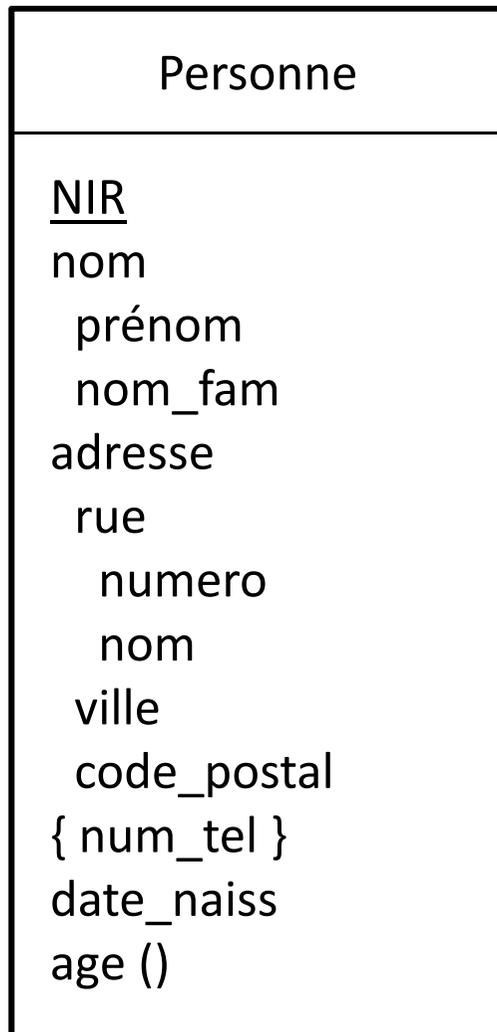
- Représentation adoptée dans le cours



Modèle E/A, les bases – Attributs

■ Représentation graphique

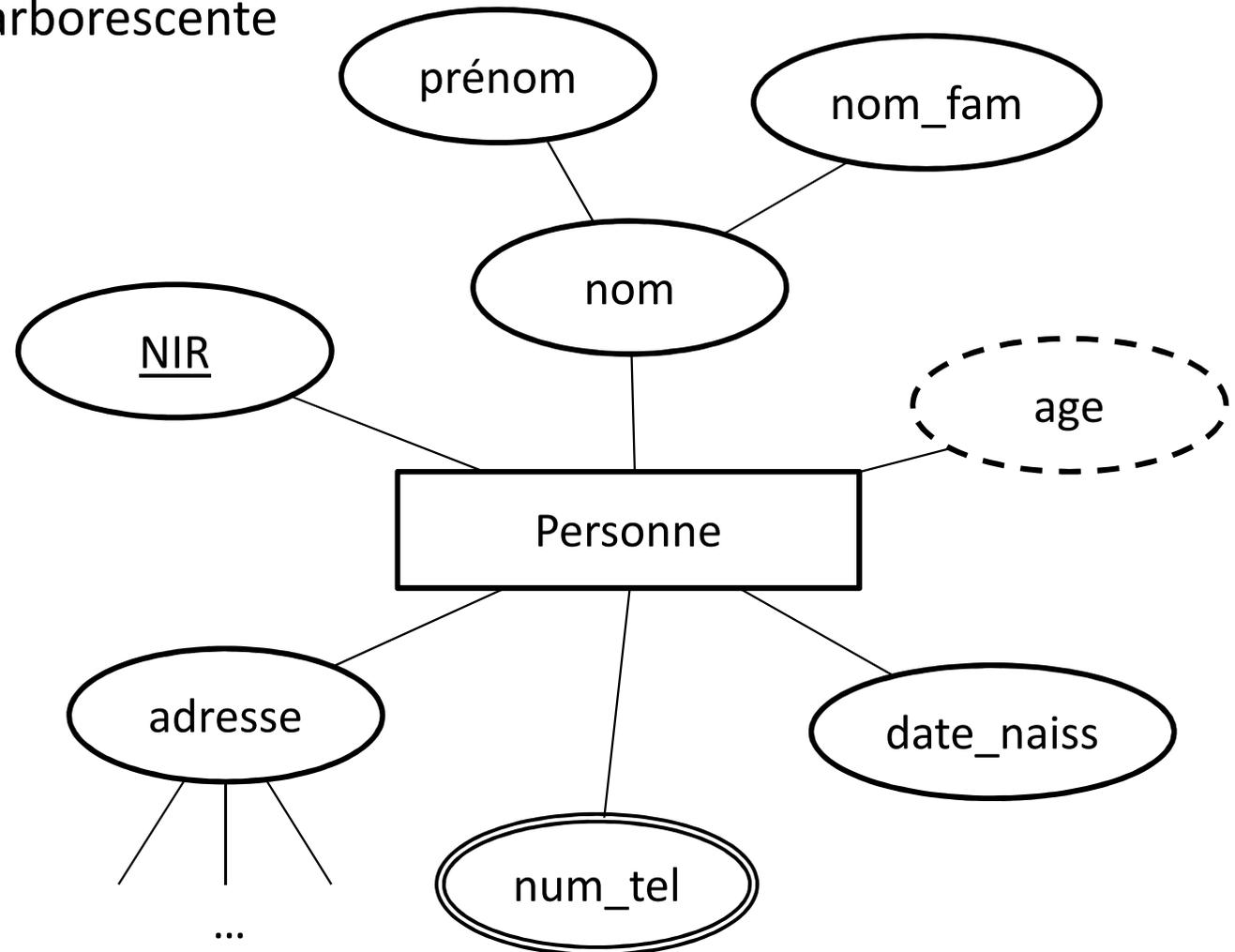
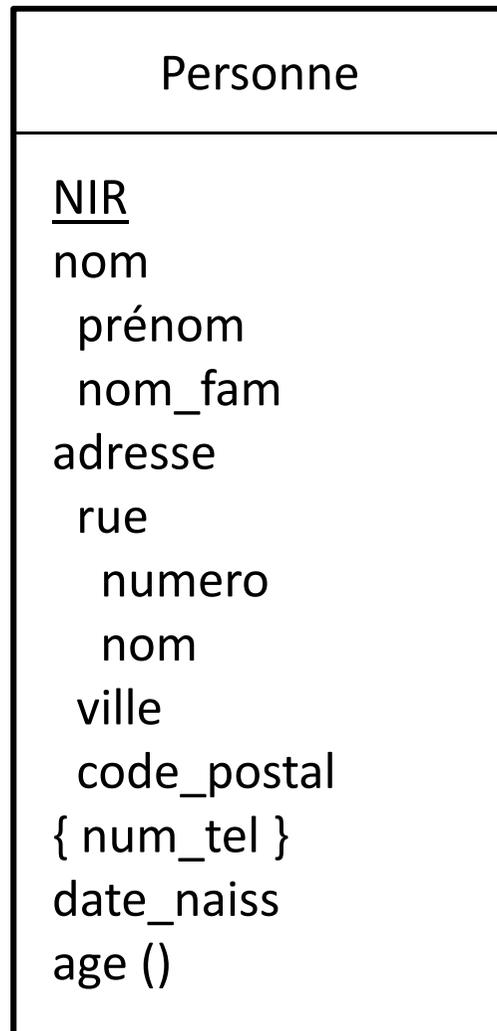
□ Comparaison avec UML



Modèle E/A, les bases – Attributs

■ Représentation graphique

- Notation alternative arborescente



Modèle E/A, les bases – Associations

■ Notion d'association

- Concept proche de la notion d'association en UML

« *lien entre plusieurs (i.e., au moins 2) entités* »

- Formellement

- Soient n *entités* e_1, e_2, \dots et e_n avec $n \geq 2$
- Soient p *attributs* a_1, a_2, \dots et a_p avec $p \geq 0$
- Le $(n + p)$ -uplet $r = (e_1, e_2, \dots, e_n, a_1, a_2, \dots, a_p)$ est une **association**
- n est appelé le *degré* de r
- Si $p > 0$, r est appelée une **association attribuée**

■ Notion d'ensemble d'associations

Groupement des *associations de même type* dans un ensemble

- Soient n *ensemble d'entités* E_1, E_2, \dots et E_n avec $n \geq 2$
- Soient p *domaines d'attributs* D_1, D_2, \dots et D_p avec $p \geq 0$
- $R \subseteq E_1 \times \dots \times E_n \times D_1 \times \dots \times D_p$ est un **ensemble d'associations**

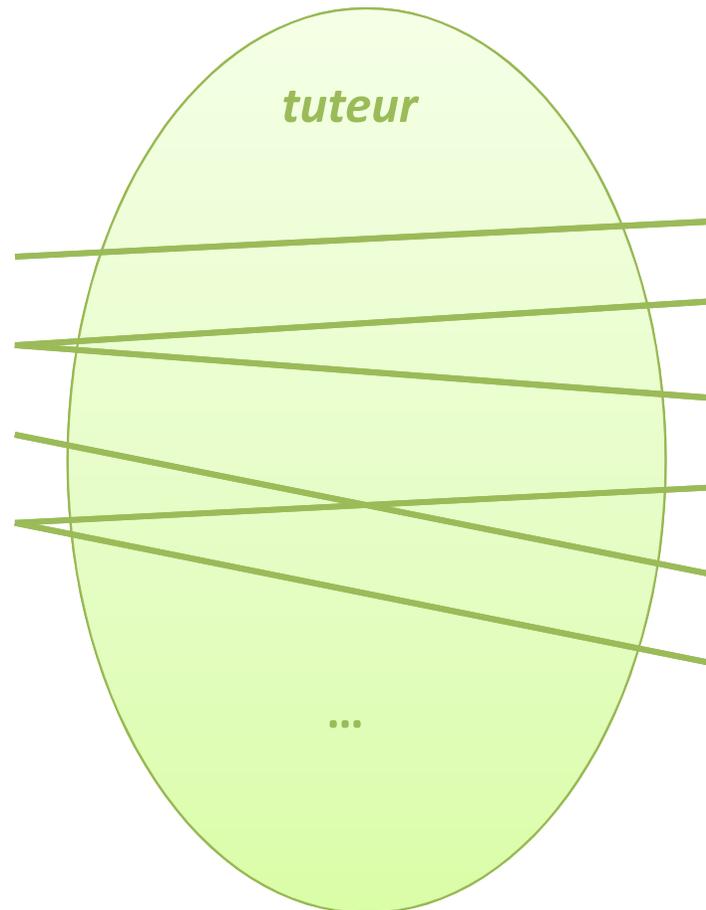
Modèle E/A, les bases – Associations

■ Exemple d'associations

- Deux ensembles explicites d'entités : *enseignant* et *étudiant*
- Une association *tuteur* de degré 2

<u>numen</u>	nom	...
76766	Crick	
45565	Katz	...
10101	Srinivasan	...
98345	Kim	...
76543	Singh	...
...

enseignant



<u>numero</u>	nom	...
98988	Tanaka	
12345	Shankar	...
00128	Zhang	...
76653	Brown	...
76543	Aoi	...
23121	Chavez	
...

étudiant

Modèle E/A, les bases – Associations

■ Exemple d'associations *attribuées*

- Deux ensembles explicites d'entités : *enseignant* et *étudiant*
- Une association *tuteur* de degré 2 avec 1 attribut

<u>numen</u>	nom	...
76766	Crick	
45565	Katz	...
10101	Srinivasan	...
98345	Kim	...
76543	Singh	...
...

enseignant



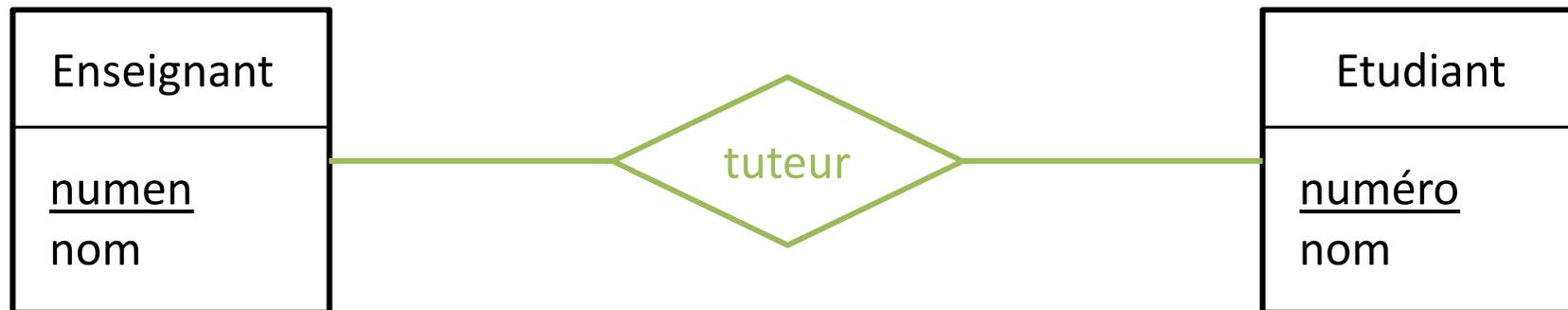
<u>numero</u>	nom	...
98988	Tanaka	
12345	Shankar	...
00128	Zhang	...
76653	Brown	...
76543	Aoi	...
23121	Chavez	
...

étudiant

Modèle E/A, les bases – Associations

■ Représentation graphique : *association simple*

- Adoptée pour le cours



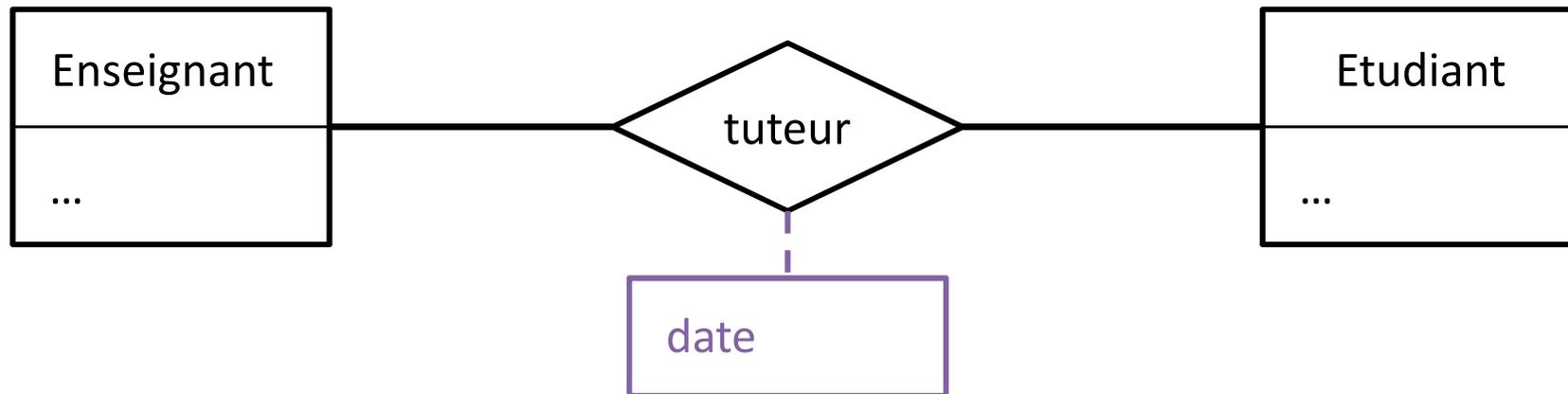
- Comparaison avec UML



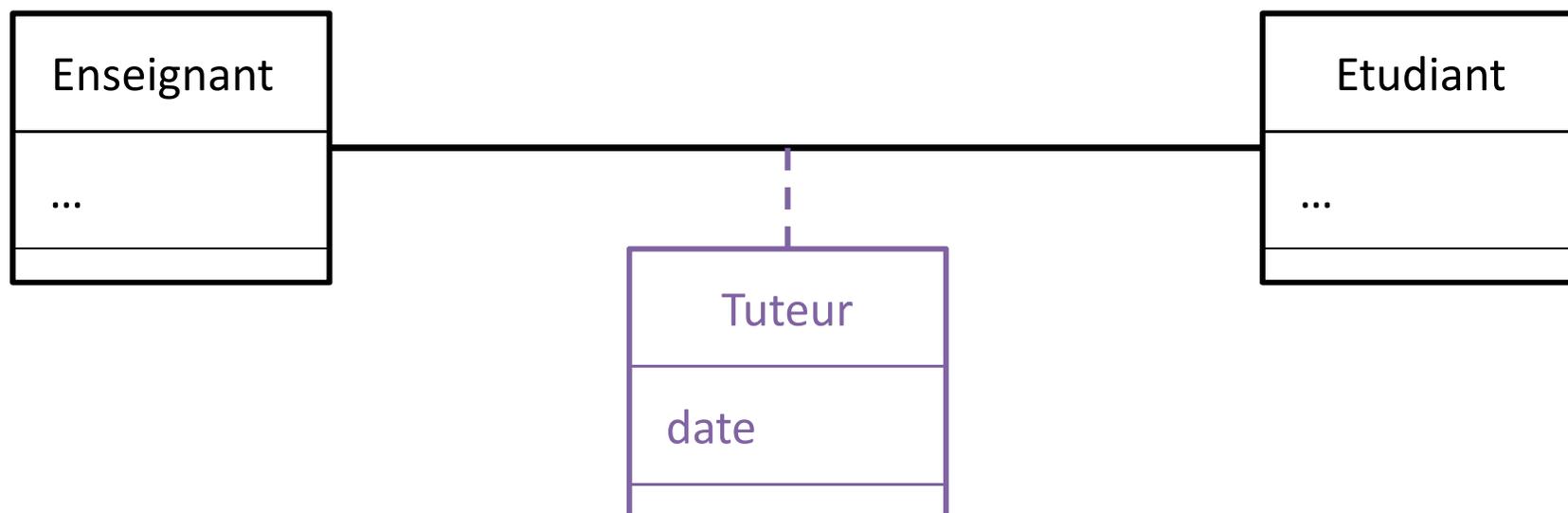
Modèle E/A, les bases – Associations

■ Représentation graphique : *association attribuée*

- Adoptée pour le cours



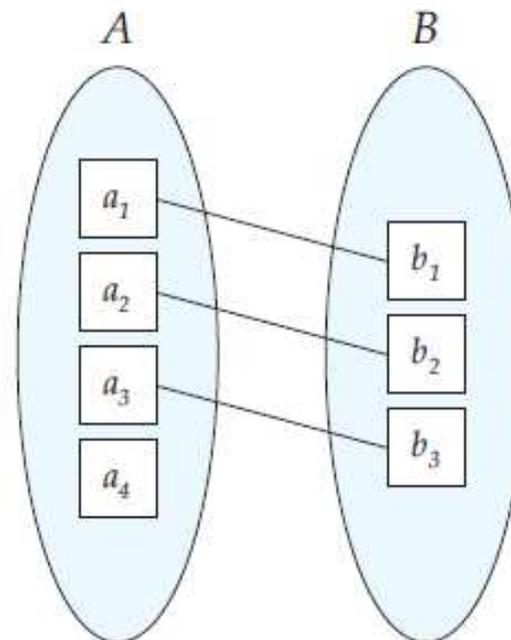
- Comparaison avec UML



Modèle E/A, les bases – Associations

■ Cardinalité

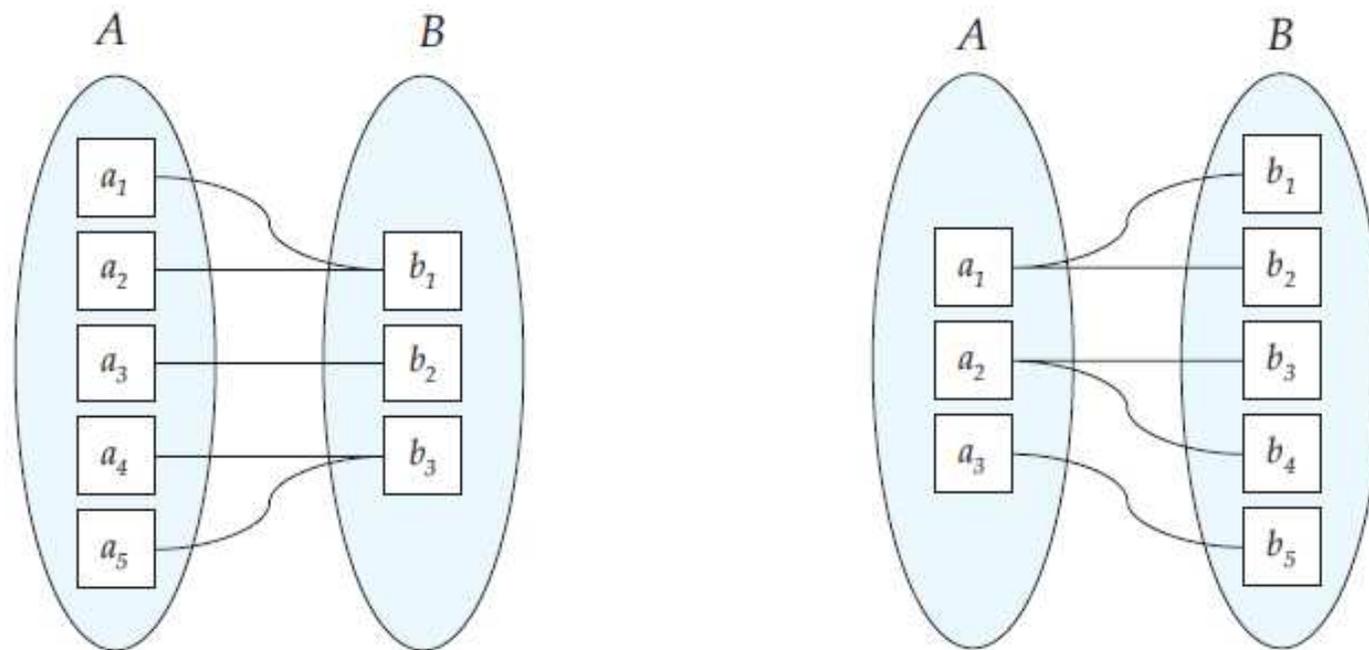
- Exprimer le *nombre* d'entités avec lesquelles une entité peut être en association via un ensemble d'associations
- Cas particulier des associations binaires (e.g., entre des ens. A et B)
 - *One-to-one*



Modèle E/A, les bases – Associations

■ Cardinalité

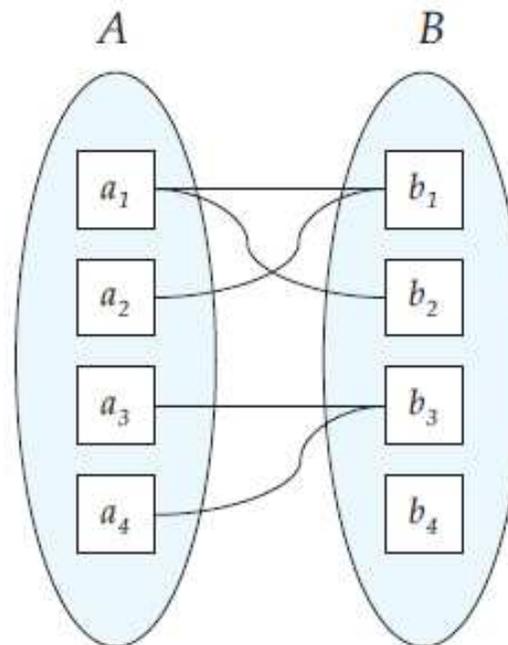
- Exprimer le **nombre** d'entités avec lesquelles une entité peut être en association via un ensemble d'associations
- Cas particulier des associations binaires (e.g., entre des ens. A et B)
 - *One-to-one*
 - *One-to-many* et *many-to-one*



Modèle E/A, les bases – Associations

■ Cardinalité

- Exprimer le **nombre** d'entités avec lesquelles une entité peut être en association via un ensemble d'associations
- Cas particulier des associations binaires (e.g., entre des ens. A et B)
 - *One-to-one*
 - *One-to-many* et *many-to-one*
 - *Many-to-many*



Modèle E/A, les bases – Associations

■ Cardinalité

- Exprimer le *nombre* d'entités avec lesquelles une entité peut être en association via un ensemble d'associations
- Cas particulier des associations binaires (*e.g.*, entre des ens. A et B)
 - *One-to-one*
 - *One-to-many* et *many-to-one*
 - *Many-to-many*
- Participation
 - Participation d'une entité E à une association $R : R \subseteq \dots \times E \times \dots$
 - Contrainte de participation
 - *Total* : toutes les entités de E apparaissent au moins une fois dans R
 - *Partielle* : certaines entités de E apparaissent au moins une fois dans R

Modèle E/A, les bases – Associations

■ Cardinalité

- Quatre possibilités de cardinalité

... on retrouve encore une fois des concepts introduits dans UML

	partielle	totale
<i>one</i>	0..1	1..1
<i>many</i>	0..*	1..*

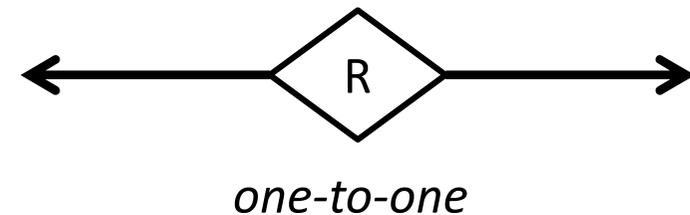
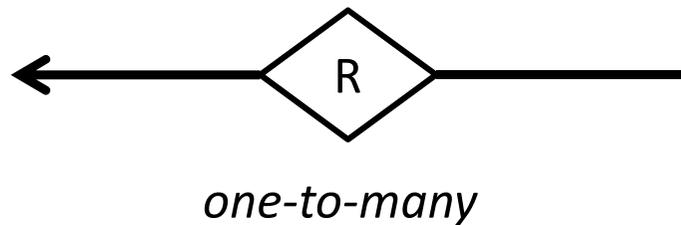
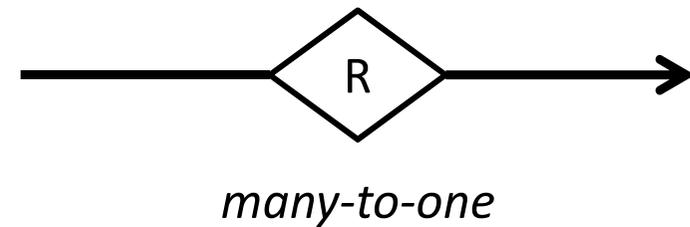
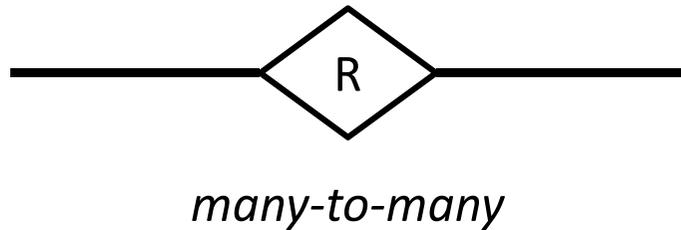
« ... au moins ... »

« ... au plus ... »

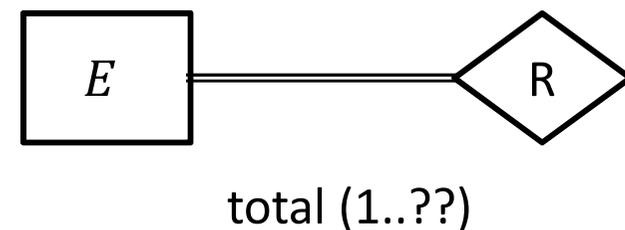
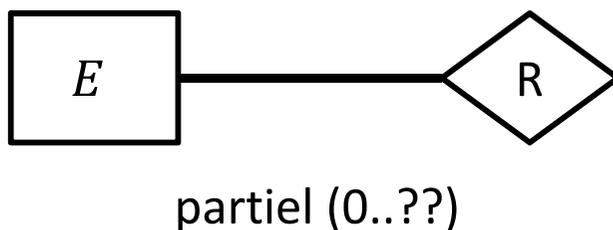
Modèle E/A, les bases – Associations

■ Cardinalité

- Notations graphiques pour les *assoc. binaires one-to-one, one-to-many, ...*



- Notations graphiques les contraintes de participation des *assoc. binaires*

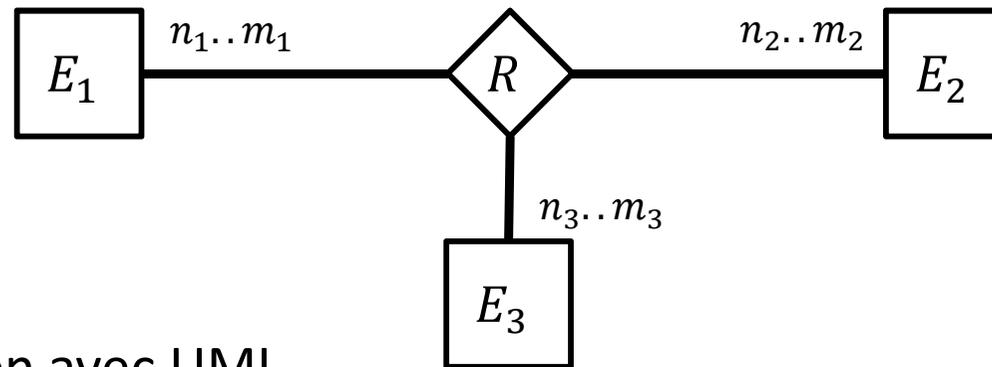


Modèle E/A, les bases – Associations

■ Cardinalité

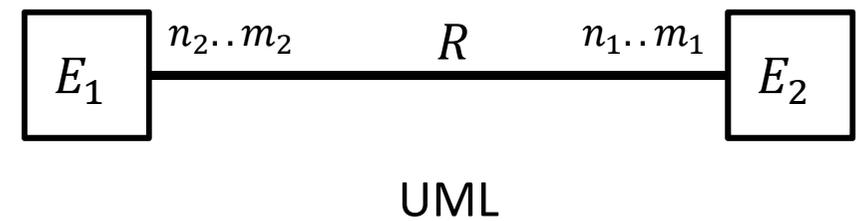
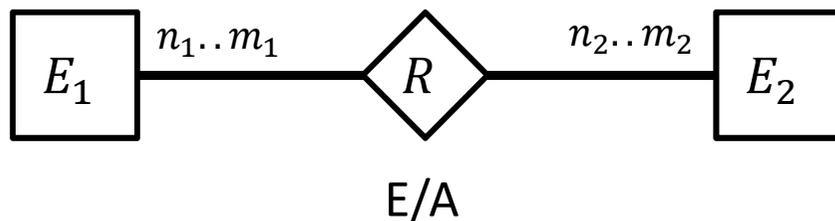
□ Notations graphiques dans le *cas général*

- Chaque ensemble d'entités E_i présente sa cardinalité dans R
 $e \in E_i$ apparaît dans au moins n_i et au plus m_i associations de R
- Utilisation autorisée pour le cas binaire



□ Comparaison avec UML

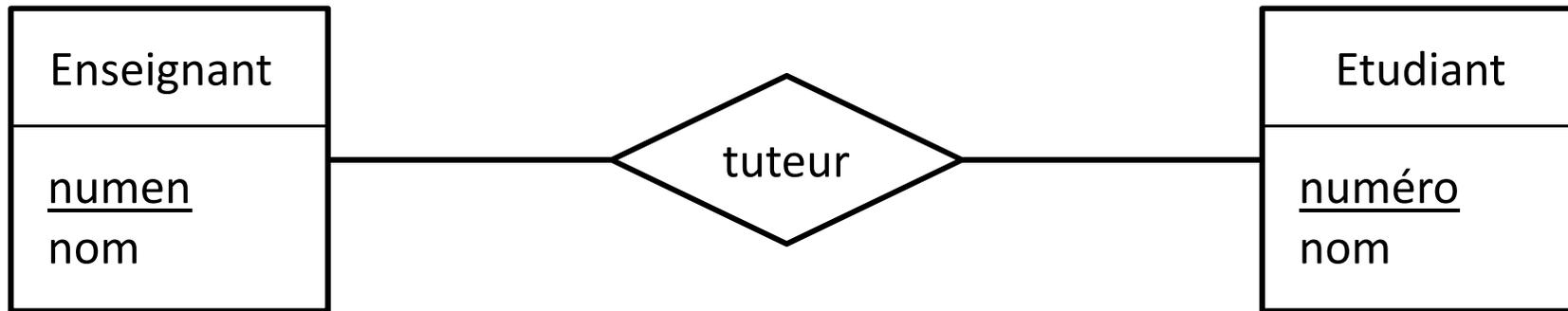
- Notation identique pour les relations n -aires avec $n \geq 3$
- Inconsistance pour $n = 2$ (inversion de la position des cardinalités)



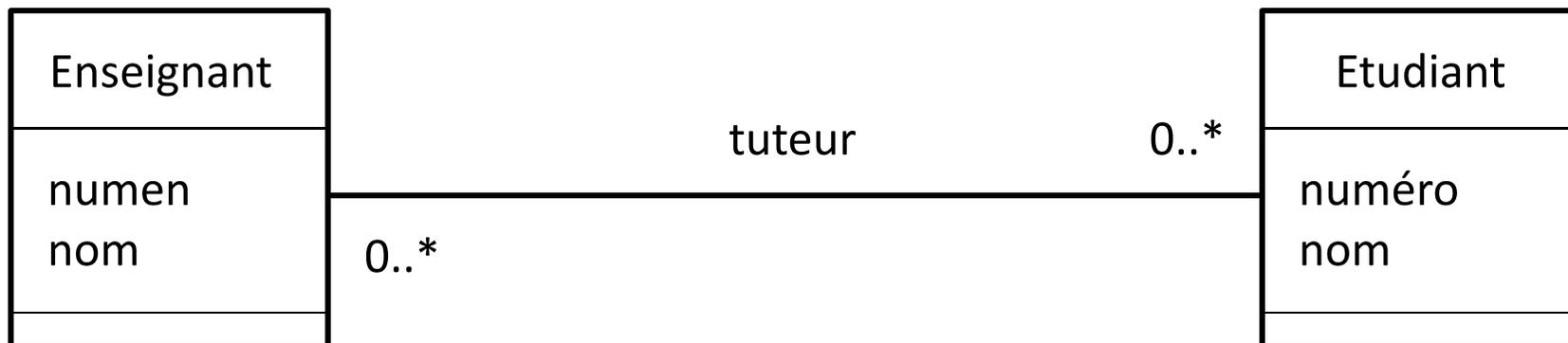
Modèle E/A, les bases – Associations

■ Exemples

□ Notation graphique



□ Équivalence avec UML



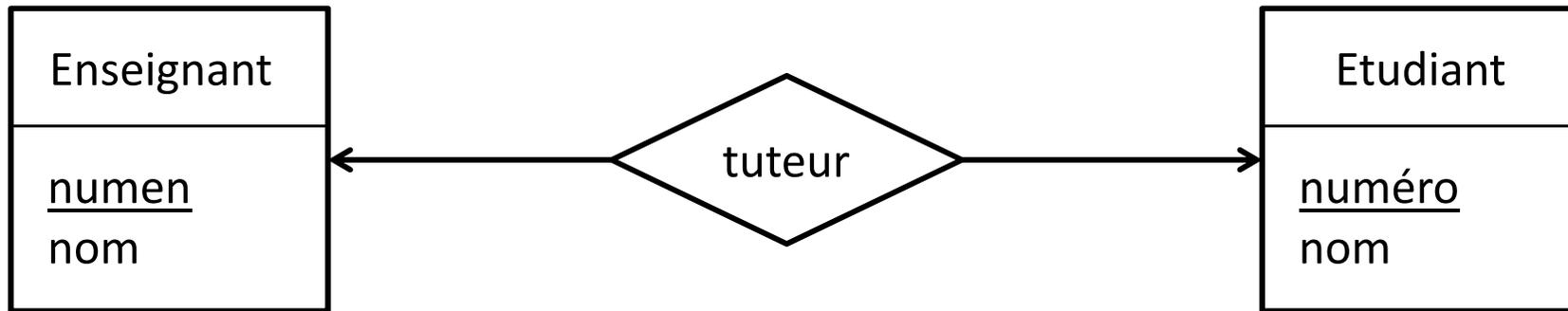
□ Interprétation

- Un enseignant **peut** être tuteur d'**un ou plusieurs** étudiants
- Un étudiant **peut** avoir **un ou plusieurs** enseignants tuteurs

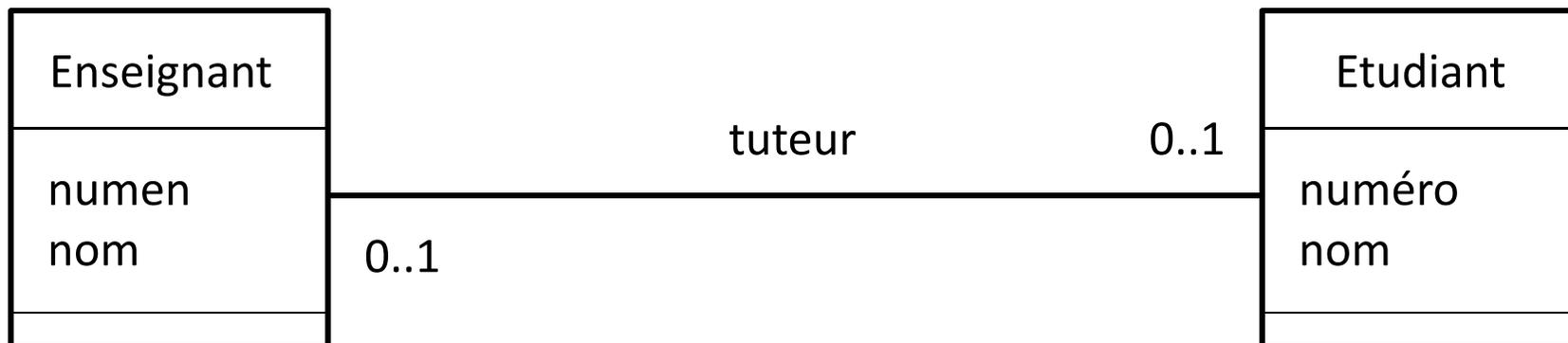
Modèle E/A, les bases – Associations

■ Exemples

□ Notation graphique



□ Équivalence avec UML



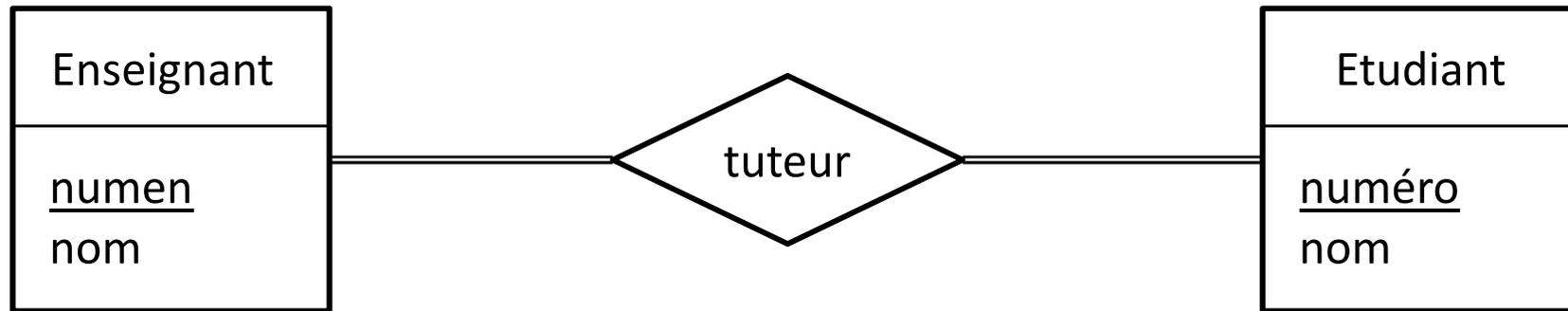
□ Interprétation

- Un enseignant **peut** être tuteur d'**au plus un** étudiant
- Un étudiant **peut** avoir **au plus un** enseignant tuteur

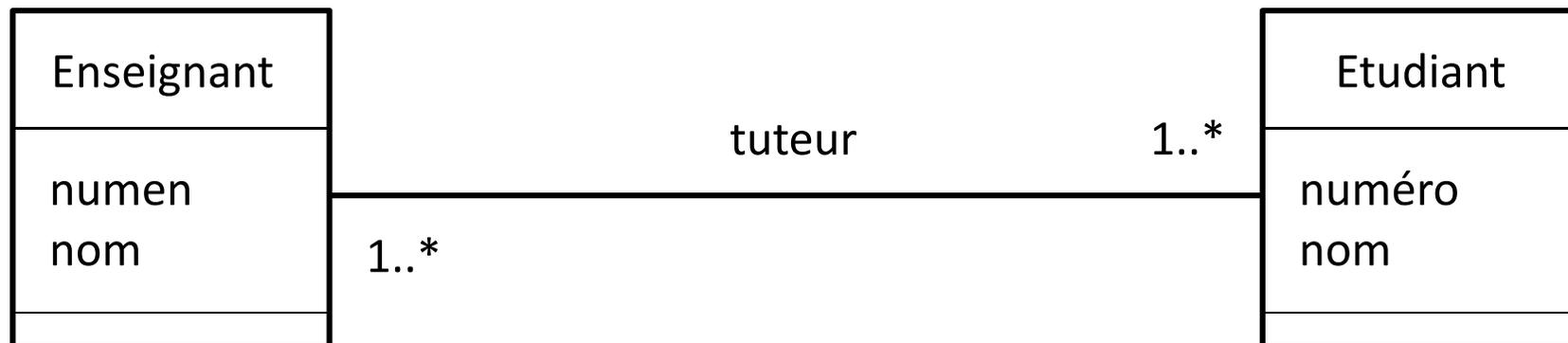
Modèle E/A, les bases – Associations

■ Exemples

□ Notation graphique



□ Équivalence avec UML



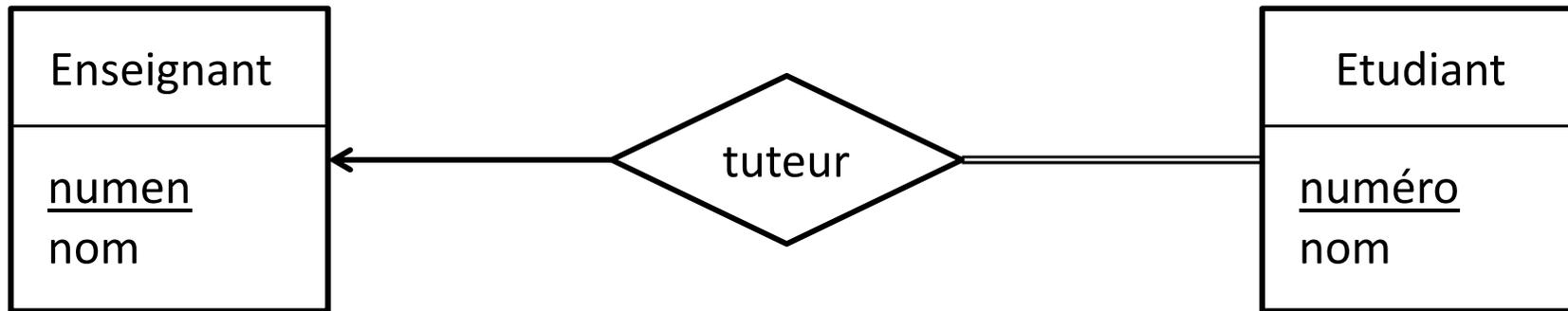
□ Interprétation

- Un enseignant **doit** être tuteur de **un ou plusieurs** étudiants
- Un étudiant **doit** avoir **un ou plusieurs** enseignants tuteurs

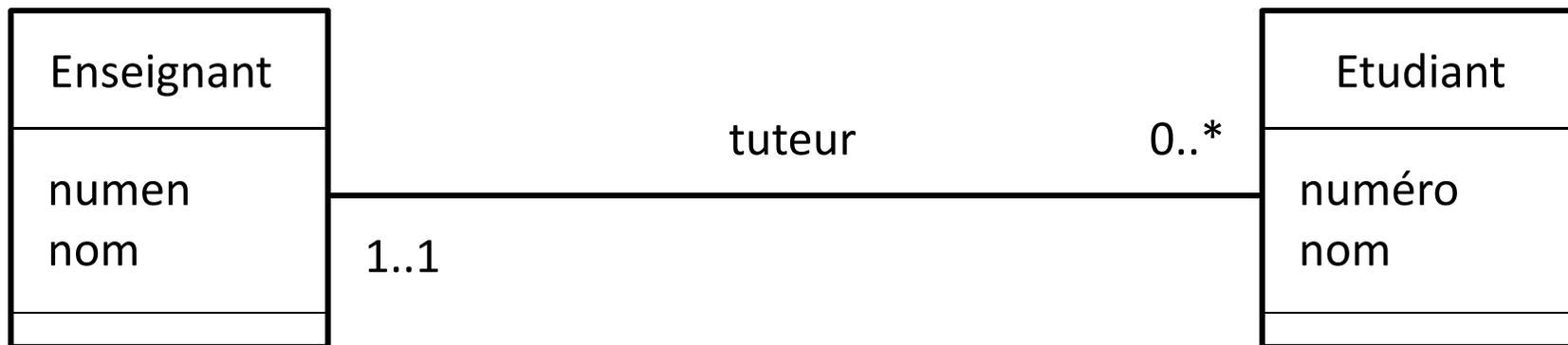
Modèle E/A, les bases – Associations

■ Exemples

□ Notation graphique



□ Équivalence avec UML



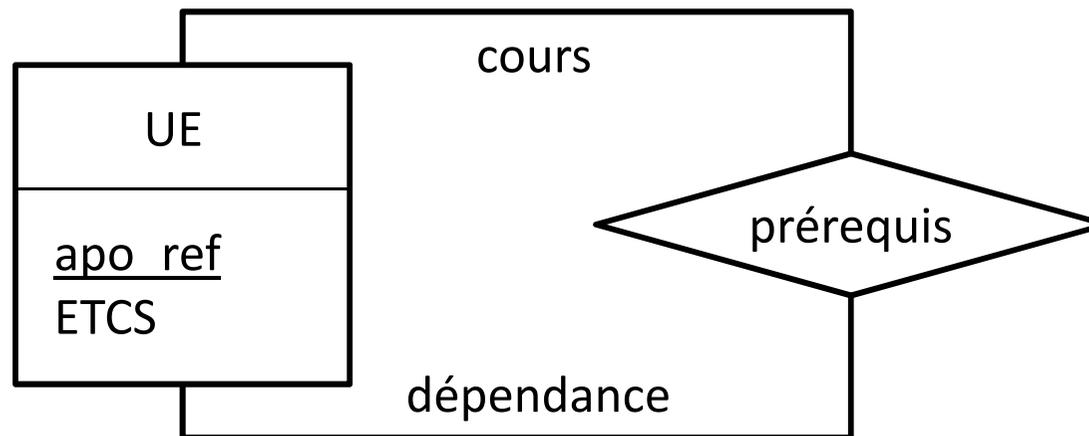
□ Interprétation

- Un enseignant **peut** être tuteur d'**un ou plusieurs** étudiants
- Un étudiant **doit** avoir **au plus un** enseignant tuteur (*i.e.*, **exactement un**)

Modèle E/A, les bases – Associations

■ Rôle

- Ambiguïtés dans les associations
lorsque le même ensemble d'entités est utilisé plusieurs fois dans une association
- Utilisation d'alias : les *rôles*
identique à l'approche UML
- Notation graphique sur un exemple



Modèle E/A, les bases – Entités faibles

■ Ensemble d'entités faible

- Définition : « ensemble d'entités sans clé primaire »
 - Possibilité d'avoir plusieurs entités différentes d'attributs identiques
 - Multi-ensemble d'entités
- Entités autorisées à la condition
Existence d'un *ensemble d'associations identifiantes*
 - Lien avec des entités fortes, appelées *ensemble d'entités identifiantes*
 - Participation de l'entité faible de cardinalité 1..1
- Clé primaire implicite
 - Soit K_S la clé primaire de l'ensemble d'entités identifiantes¹
 - Soit K_D un *discriminant* de l'ensemble d'entités faible (*souligné en pointillés*)
Une clé partielle des entités faibles
 - $K_S \cup K_D$ définit une clé primaire de l'ensemble d'entités faible

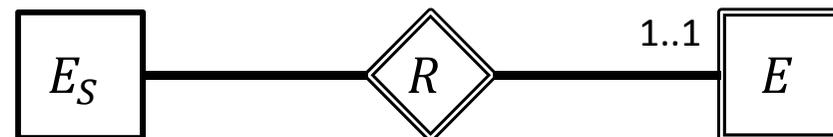
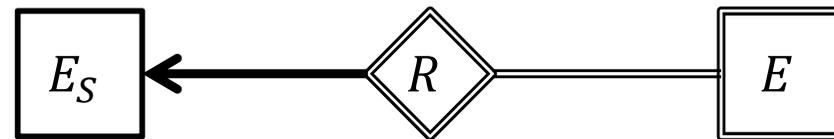
¹ On suppose qu'il n'y a qu'un seul ensemble d'entités identifiant ; s'il y en a plusieurs K_S représente l'union de leurs clés primaires

Modèle E/A, les bases – Entités faibles

■ Ensemble d'entités faible

□ Notation graphique

- E ensemble d'entités faible (le discriminant apparaît souligné en pointillés)
- R ensemble d'associations identifiant
- E_S ensemble d'entités identifiant

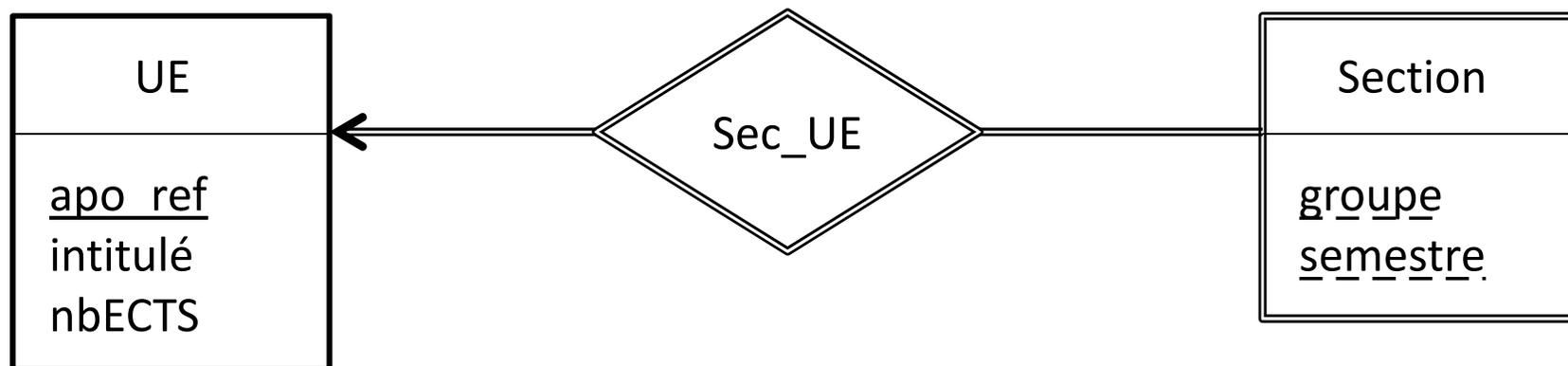


□ Pas d'équivalent UML

Modèle E/A, les bases – Entités faibles

■ Exemple

- UE divisée en sections
- Section décrite par le groupe et le semestre
- Exemples : « programmation impérative » et « architecture et système »



Plan



- Introduction à la conception de BD
- Modèle entités/associations, concepts de base
- Modèle entités/associations étendu
- Quelques problématiques de conception

Modèle E/A étendu

■ Modèle E/A de base

- Modélisation de la plupart des propriétés sur les données...
- ... mais absence des notions apparues plus tard

■ Deux extensions

- Éléments de modélisation objet (généralisation, spécialisation, héritage)
- Agrégation

Modèle E/A étendu – Généralisation

■ Spécialisation

- Processus de modélisation *top-down*
- Distinguer un sous-ensemble d'entités d'un ensemble d'entités
 - Spécification d'un ensemble d'entités de plus bas niveau
 - Attributs supplémentaires, participations particulières à des associations

■ Généralisation

- Processus de modélisation *bottom-up*
- Combiner des ensembles d'entités partageant certaines caractéristiques
 - Spécification d'un ensemble d'entités de plus haut niveau
 - Factorisations d'attributs ou de participations à des associations

■ Héritage

- Généralisation vs. Spécialisation \equiv hiérarchie de classes (modél. objet)
 - Entité de plus haut niveau : regroupe les attributs/associations *communes*
 - Entité de plus bas niveau : regroupe les attributs/associations *propres*

Modèle E/A étendu – Généralisation

■ Contraintes sur la généralisation

□ Généralisation *logique* ou *d'utilisation*

■ *Quelles entités sont membres d'un ensemble de plus bas niveau ?*

■ Contrainte *logique*

□ Propriété logique sur les éléments du modèle E/A

□ Exemple

ClSénior est une spécification de **Client**

les clients de plus de 65 ans sont des entités de **ClSénior**

■ Contrainte *d'utilisation*

□ La condition est gérée de façon extérieure (lors de l'utilisation de la BD)

□ Exemple

régime alimentaire d'un participant à une conférence

lors de l'inscription, le participant désigne son régime alimentaire

Modèle E/A étendu – Généralisation

■ Contraintes sur la généralisation

- Généralisation *logique* ou *d'utilisation*
- Généralisation *disjointe* ou *chevauchante*
 - *Une entité peut-elle appartenir à plusieurs spécialisations ou à une seule ?*
 - **Disjointe**
 - Ne peut appartenir qu'à une seule des spécialisations
 - Notion des proches des objets en programmation par exemple
 - **Chevauchante (*overlapping*)**
 - Peut appartenir à plusieurs spécialisations

Modèle E/A étendu – Généralisation

■ Contraintes sur la généralisation

- Généralisation *logique* ou *d'utilisation*
- Généralisation *disjointe* ou *chevauchante*
- Généralisation *totale* ou *partielle*
 - *Une entité doit-elle forcément appartenir à l'une des spécialisations ?*
 - **Totale**
 - Doit appartenir forcément à l'une des spécialisations
 - Analogie au cas où l'entité de plus haut niveau correspond à une classe abstraite
 - **Partielle**
 - Ne doit pas forcément être spécialisée

Modèle E/A étendu – Généralisation

■ Contraintes sur la généralisation

- Généralisation *logique* ou *d'utilisation*
 - Généralisation *disjointe* ou *chevauchante*
 - Généralisation *totale* ou *partielle*
- } cardinalité de la généralisation

	<i>partielle</i>	<i>totale</i>
<i>chevauchante</i>	rien ou ○	⊖
<i>disjointe</i>	⊙	⊕

« ... au moins ... »

« ... au plus ... »

Modèle E/A étendu – Généralisation

■ Notations graphiques et exemple

□ Proche d'UML

où les contraintes sur les cardinalités doivent être exprimées en OCL

□ Employés d'une entreprise

■ Classification suivant le type de profil

■ Productif (*e.g.*, consultant technique) ou commercial

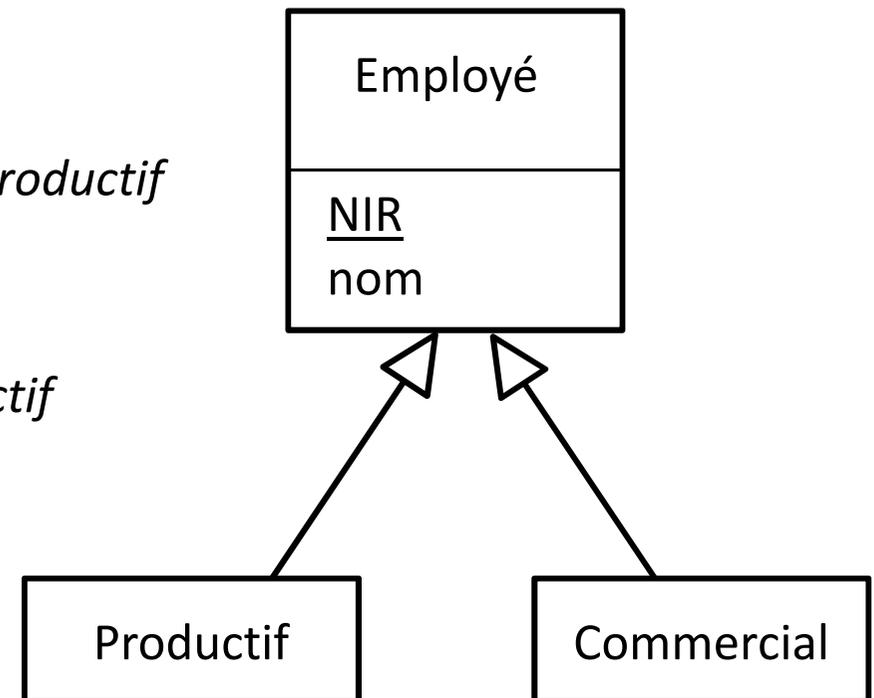
□ Sémantique : *partielle* et *chevauchante*

■ Commercial \cup Productif \neq Employé

*Un employé **n'est pas** forcément un productif
ou un commercial*

■ Commercial \cap Productif $\neq \emptyset$

*Un employé **peut être** à la fois productif
et commercial*



Modèle E/A étendu – Généralisation

■ Notations graphiques et exemple

□ Proche d'UML

où les contraintes sur les cardinalités doivent être exprimées en OCL

□ Employés d'une entreprise

■ Classification suivant le type de profil

■ Productif (*e.g.*, consultant technique) ou commercial

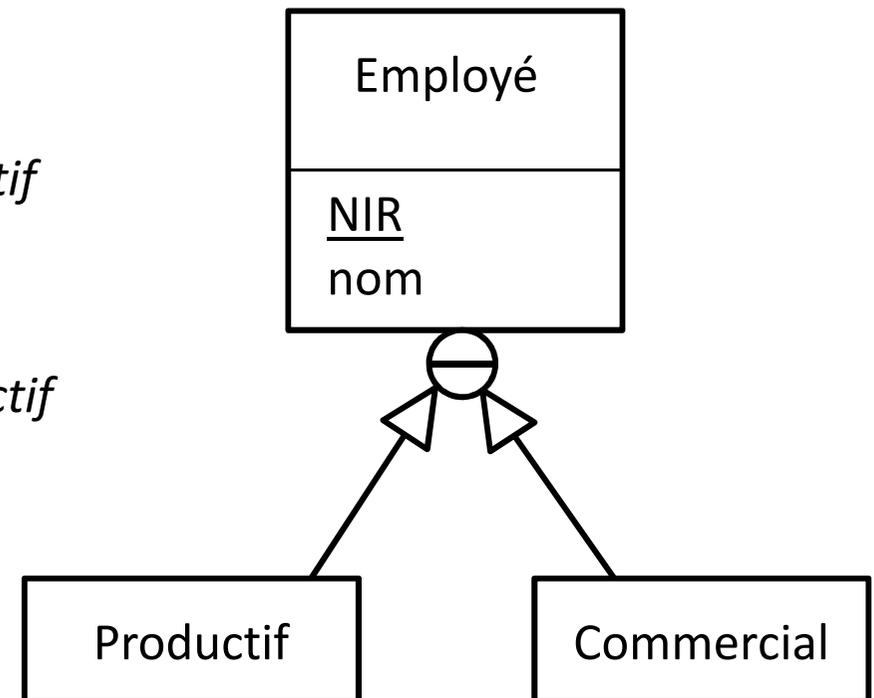
□ Sémantique : *totale* et *chevauchante*

■ Commercial \cup Productif = Employé

*Un employé **est** forcément un productif
ou un commercial*

■ Commercial \cap Productif $\neq \emptyset$

*Un employé **peut être** à la fois productif
et commercial*



Modèle E/A étendu – Généralisation

■ Notations graphiques et exemple

□ Proche d'UML

où les contraintes sur les cardinalités doivent être exprimées en OCL

□ Employés d'une entreprise

■ Classification suivant le type de profil

■ Productif (*e.g.*, consultant technique) ou commercial

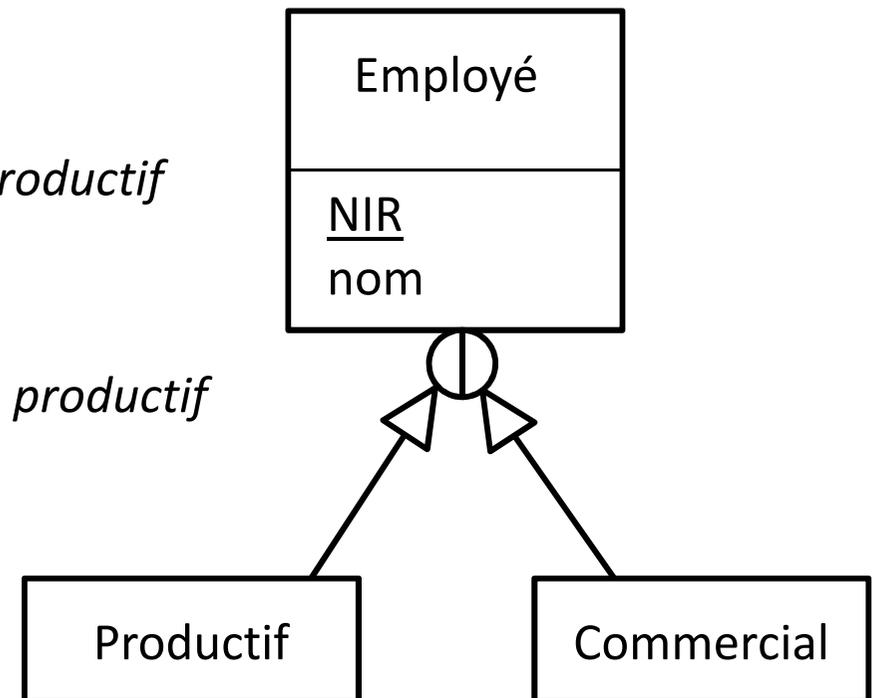
□ Sémantique : *partielle* et *disjointe*

■ Commercial \cup Productif \neq Employé

*Un employé **n'est pas** forcément un productif ou un commercial*

■ Commercial \cap Productif = \emptyset

*Un employé **ne peut pas être** à la fois productif et commercial*



Modèle E/A étendu – Généralisation

■ Notations graphiques et exemple

□ Proche d'UML

où les contraintes sur les cardinalités doivent être exprimées en OCL

□ Employés d'une entreprise

■ Classification suivant le type de profil

■ Productif (*e.g.*, consultant technique) ou commercial

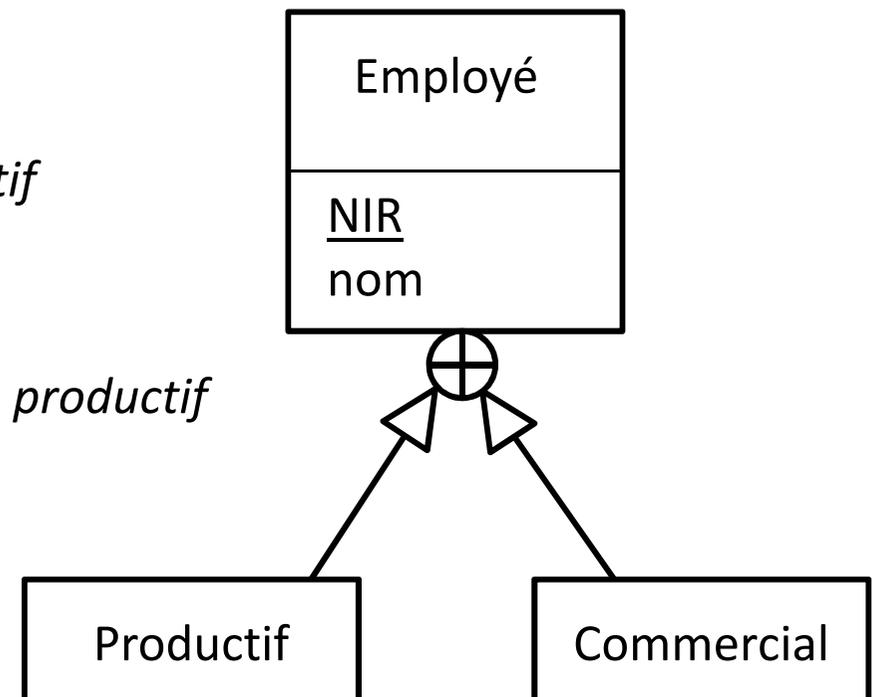
□ Sémantique : *totale* et *disjointe*

■ $\text{Commercial} \cup \text{Productif} = \text{Employé}$

*Un employé **est** forcément un productif
ou un commercial*

■ $\text{Commercial} \cap \text{Productif} = \emptyset$

*Un employé **ne peut pas être** à la fois productif
et commercial*



Modèle E/A étendu – Agrégation

■ Association comme entité

- Par exemple : association entre associations...
- Limitation du modèle E/A de base
Pas « d'ordre supérieur »

■ Agrégation

- Abstraction où les associations sont considérées comme des entités d'ordre supérieur
- *Attention*

Notion totalement différente des agrégations en UML

Les agrégations UML permettent d'abstraire les notions de collections, tableaux, listes, etc. en programmation

Modèle E/A étendu – Agrégation

■ Notations graphiques et exemple

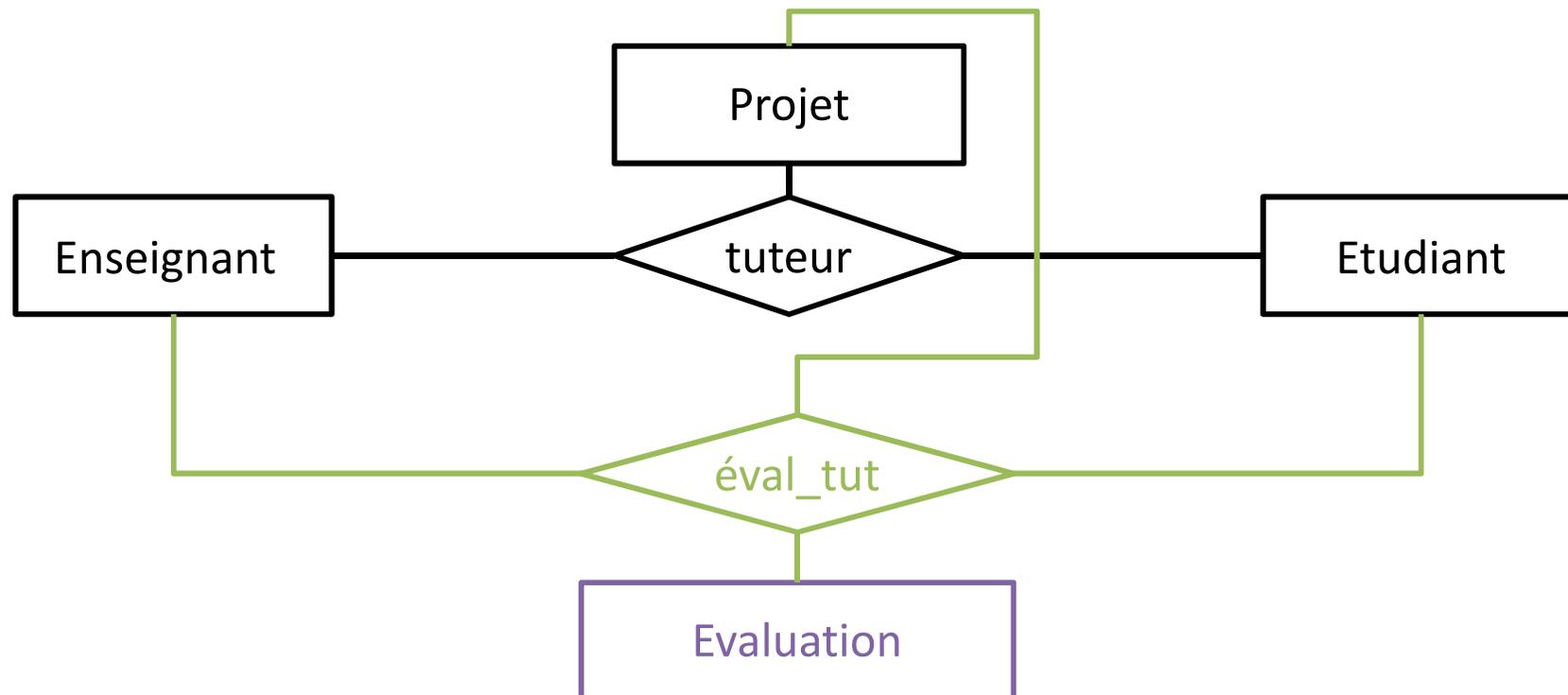
□ Pas d'équivalent en UML

□ Exemple : tutorat entre enseignants et étudiants

■ Ajout d'une entité Projet : tutorat est une association ternaire

■ Entité Evaluation

« l'évaluation d'un tutorat dépend du projet, de l'enseignant et de l'étudiant »

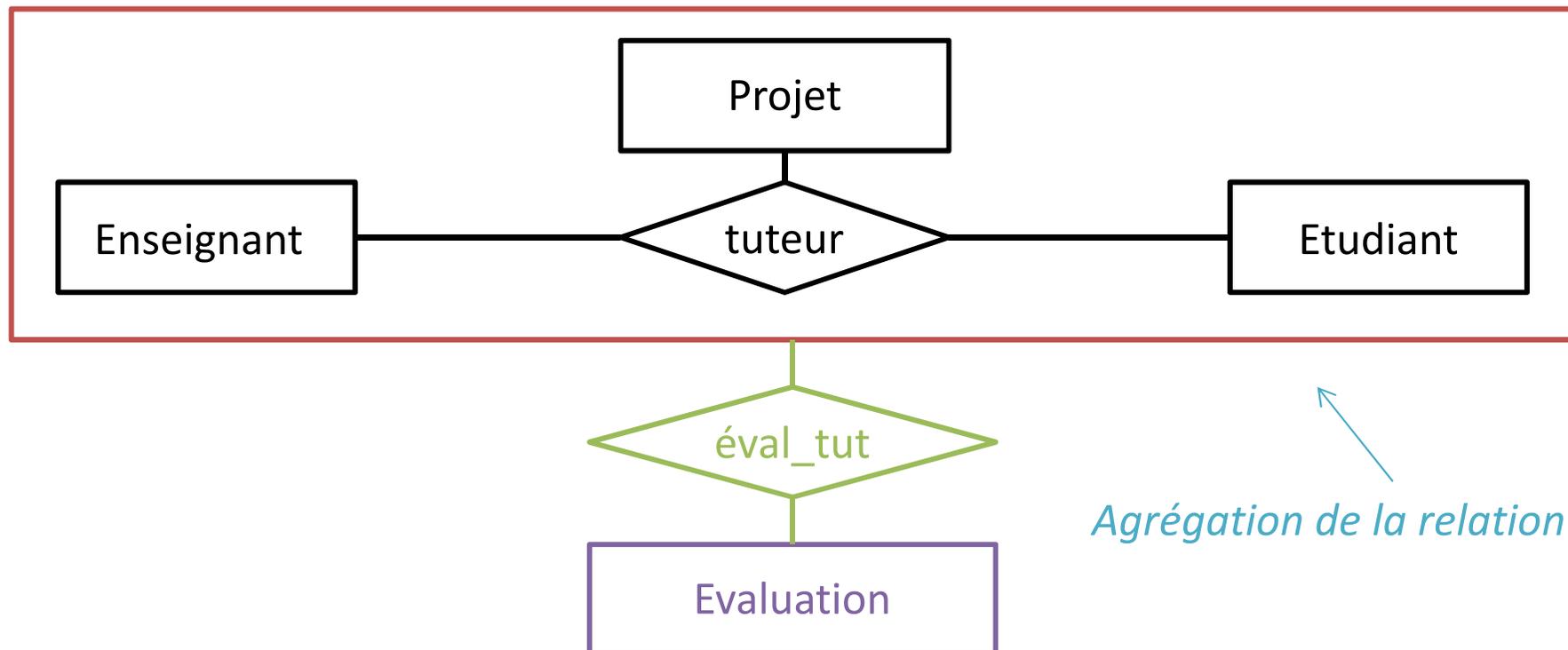


Modèle E/A étendu – Agrégation

■ Notations graphiques et exemple

- Pas d'équivalent en UML
- Exemple : tutorat entre enseignants et étudiants
 - Ajout d'une entité Projet : tutorat est une association ternaire
 - Entité Evaluation

« l'évaluation d'un tutorat dépend du projet, de l'enseignant et de l'étudiant »



Agrégation de la relation tuteur

Plan

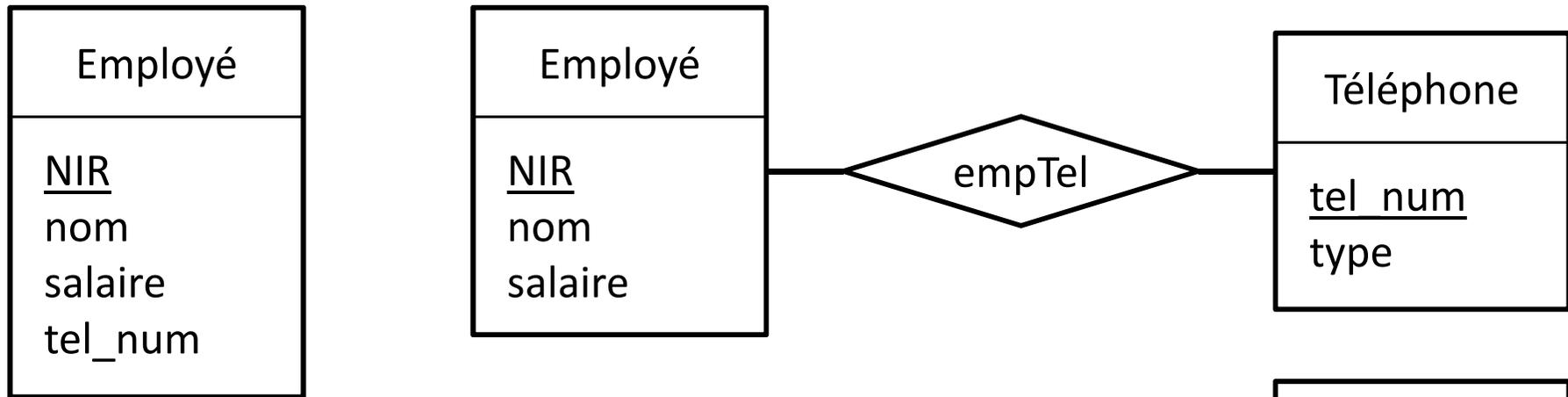


- Introduction à la conception de BD
- Modèle entités/associations, concepts de base
- Modèle entités/associations étendu
- Quelques problématiques de conception

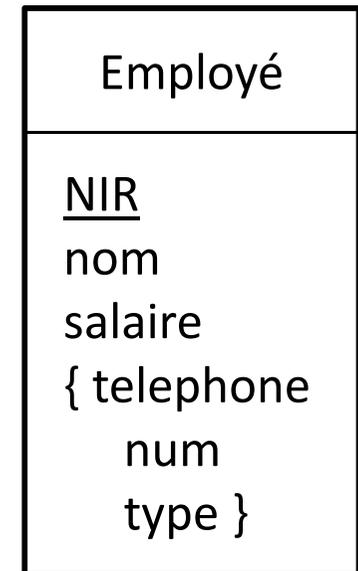
Éléments de conception – FAQ

■ Ensemble d'entités vs. attribut ?

- Exemple : employé et numéro de téléphone



- Aucune réponse simple...
- Préconisations
 - L'attribut est un objet en soi, a une importance dans la BD
Par exemple, utiliser la BD pour générer un annuaire
 - L'attribut a lui-même des attributs
 - Un attribut composite ou multivalué
cache souvent une entité non explicitée dans les spec.



Éléments de conception – FAQ

■ Ensemble d'entités vs. ensemble d'associations ?

- Exemple : étudiant et UE



- Aucune réponse simple...
- Préconisation

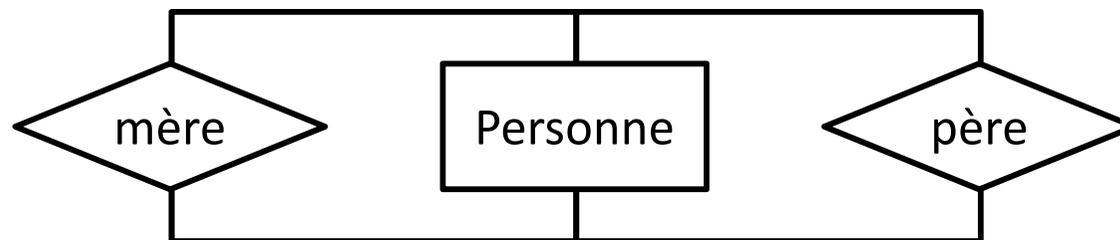
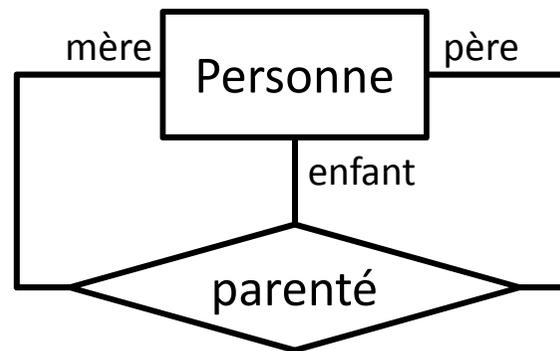
Désigner une relation par un verbe décrivant l'*action* entre les entités

Éléments de conception – FAQ

■ Association n -aire vs. association binaire ?

□ Transformation d'une association n -aire en associations binaires...

■ Exemples : enfant et parent



Éléments de conception – FAQ

■ Association n -aire vs. association binaire ?

□ Transformation d'une association n -aire en associations binaires...

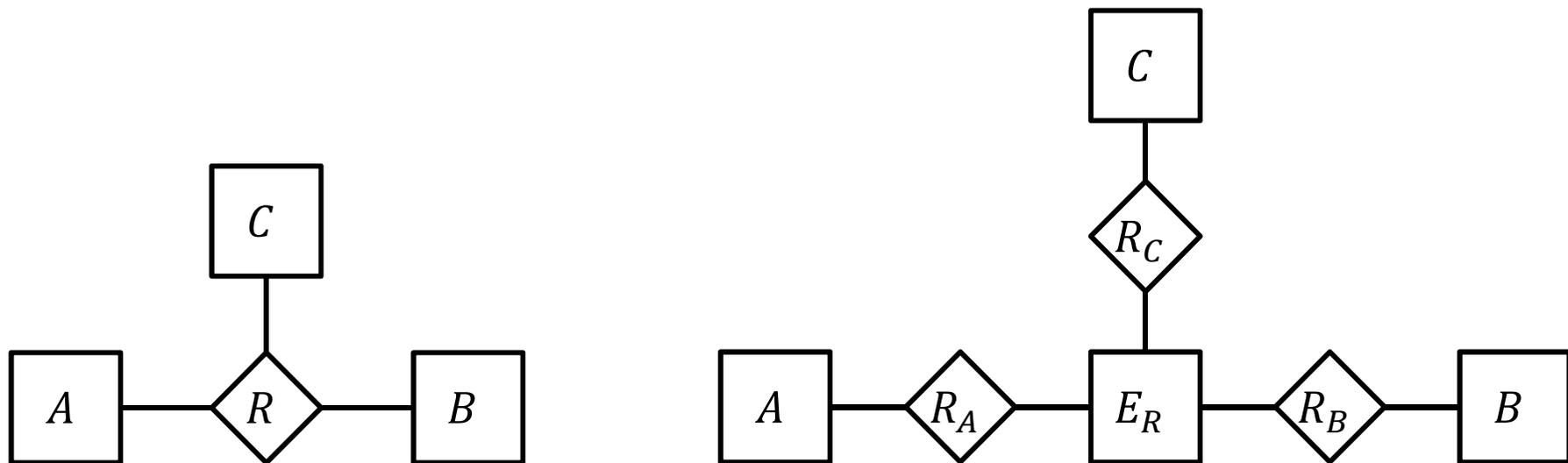
■ Cas général (pour $n = 3$)

□ Ajout d'un attribut identifiant pour E_R (pas indiqué sur le schéma)
utilisation possible d'une entité faible pour E_R

□ Pour chaque association $(a_i, b_i, c_i) \in R$

- création d'une entité $e_i \in E_R$

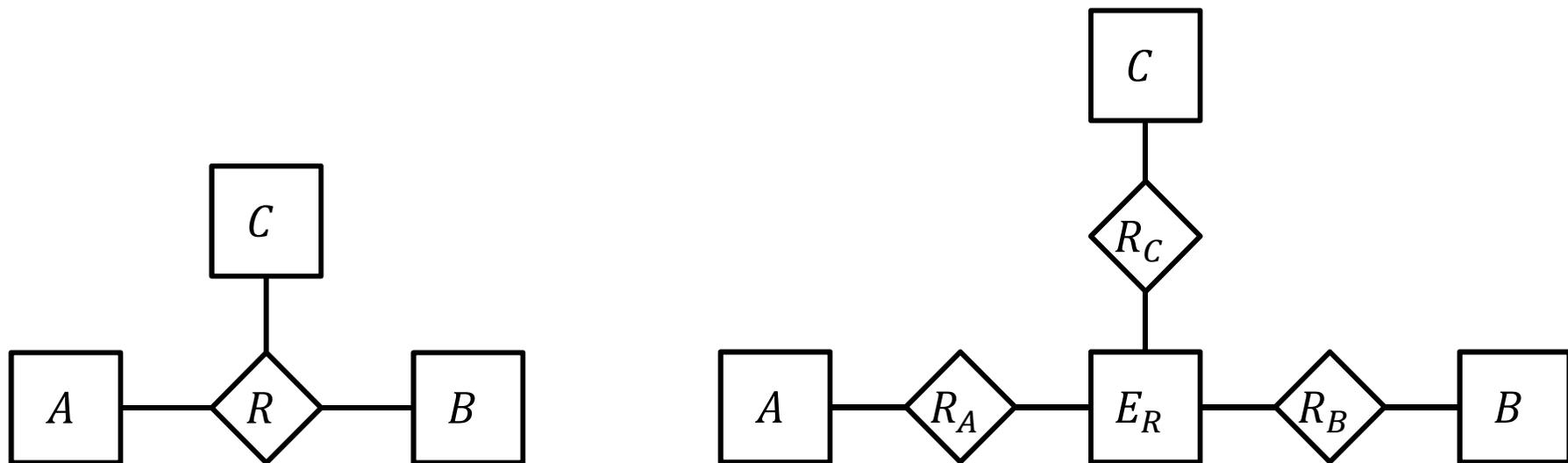
- avec $(e_i, a_i) \in R_A, (e_i, b_i) \in R_B, (e_i, c_i) \in R_C$



Éléments de conception – FAQ

■ Association n -aire vs. association binaire ?

- Transformation d'une association n -aire en associations binaires...
- ... mais parfois non désirée
 - Complexifie le schéma
 - Traduction E/A => schéma relationnel, moins de jointure
 - Certains cas de cardinalité ne peuvent être traduits
 - Considérer le cas R *many-to-one* de A et B vers C
chaque paire (a, b) est associé à au plus un c



Éléments de conception – Validation E/A

- Validation d'un modèle E/A
 - Vérification de la complétude du modèle
 - Vérification de la cohérence du modèle
 - Simplification du modèle

Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- **Vérification de la complétude du modèle**
- Vérification de la cohérence du modèle
- Simplification du modèle

■ Complétude du modèle

- Objectifs : vérifier que l'ensemble des spécifications sont présentes
- Présentation de chaque élément du modèle E/A
 - Entité

Nom	Enseignant
Définition	Personnel de l'université délivrant une quantité d'enseignement non nulle
Liste des attributs	NIR, nom, prénom, grade, section
Clé primaire	NIR

Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- **Vérification de la complétude du modèle**
- Vérification de la cohérence du modèle
- Simplification du modèle

■ Complétude du modèle

- Objectifs : vérifier que l'ensemble des spécifications sont présentes
- Présentation de chaque élément du modèle E/A
 - Association

Nom	Tutorat
Définition	Représente le suivi d'un étudiant sur un projet par un enseignant
Liste des entités (cardinalité+rôle)	Enseignant (0..N, tueur), Etudiant (1..1, tutoré), Projet (1..N)
Liste des attributs	-

Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- **Vérification de la complétude du modèle**
- Vérification de la cohérence du modèle
- Simplification du modèle

■ Complétude du modèle

- Objectifs : vérifier que l'ensemble des spécifications sont présentes
- Présentation de chaque élément du modèle E/A
 - Attribut

Nom	Grade
Définition	Corps auquel appartient un enseignant
Structure	Simple – monovalué
Domaine de valeur	{ MCF, Pr, PRAG, ... }
Appartenance à la clé	Non

Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- **Vérification de la complétude du modèle**
- Vérification de la cohérence du modèle
- Simplification du modèle

■ Complétude du modèle

- Objectifs : vérifier que l'ensemble des spécifications sont présentes
- Présentation de chaque élément du modèle E/A
 - Contrainte d'intégrité

Nom	Tutorat de recherche
Éléments E/A impliqués	Association Tutorat, Attribut Grade (Enseignant) Attribut Type (Projet)
Assertion	Seuls les enseignants de grade MCF et Pr, peuvent encadrer un projet de type de Recherche

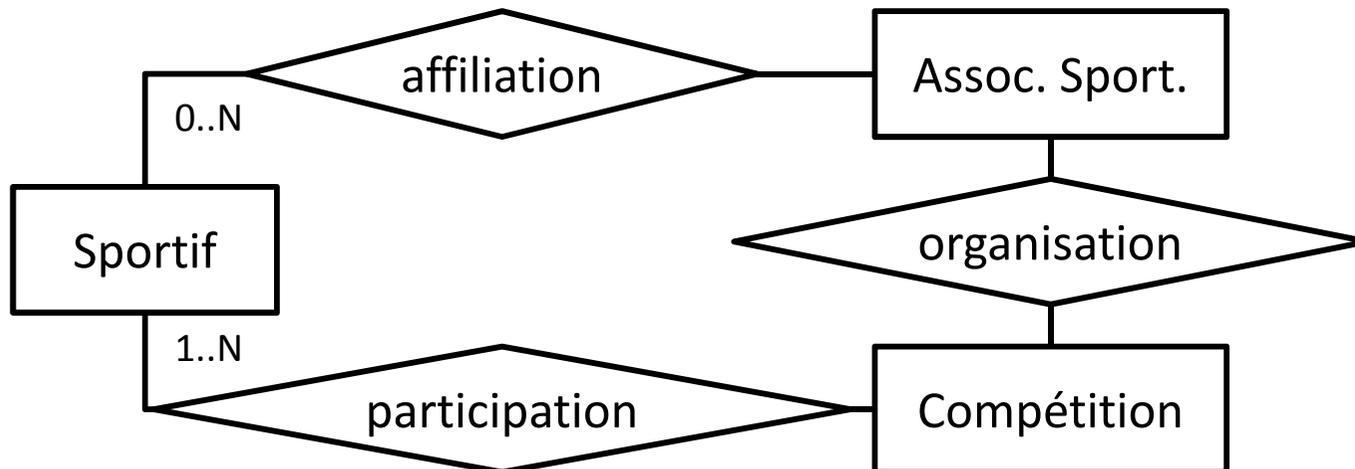
Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- Vérification de la complétude du modèle
- **Vérification de la cohérence du modèle**
- Simplification du modèle

■ Cohérence du modèle

- Existe-t-il des contraintes contradictoires ?
Ex: niveau BD, vérifier s'il est possible de peupler les tables
- Aucune approche systématique



Contrainte d'intégrité :
Un sportif ne peut participer à une compétition que s'il est affilié à une association

Éléments de conception – Validation E/A

■ Validation d'un modèle E/A

- Vérification de la complétude du modèle
- Vérification de la cohérence du modèle
- **Simplification du modèle**

■ Simplification du modèle

- Objectifs
 - Élimination de la redondance et des ambiguïtés
 - Clarté
 - Concision
 - Stabilité du schéma (limiter les besoins de restructuration lors de l'ajout d'entités/associations)
 - Facilité de mise-à-jour (éviter l'introduction d'anomalies lors de l'exploitation de la BD)
- Application d'une série de règles de simplification
Différent d'une mise sous forme normale (rien d'automatique)

Éléments de conception – Simplification E/A

■ Absence d'homonyme et de synonyme

□ Homonymie

- Noms identiques pour différents éléments E/A¹
introduction d'ambiguïtés (*e.g.*, penser aux jointures naturelles)
- Transformation
 - Éléments de sémantiques différentes : renommage
 - Éléments de sémantiques identiques (modélisation redondante)
supprimer un des éléments et restructurer le modèle E/A

□ Synonymie

- Éléments E/A de sémantiques équivalentes nommés différemment
Redondance dans le schéma et ambiguïté
- Transformation
Supprimer un des éléments et restructurer le modèle E/A

¹ On appelle « élément E/A » une entité, une association, un attribut ou une contrainte d'intégrité

Éléments de conception – Simplification E/A

■ Minimalité des clés primaires

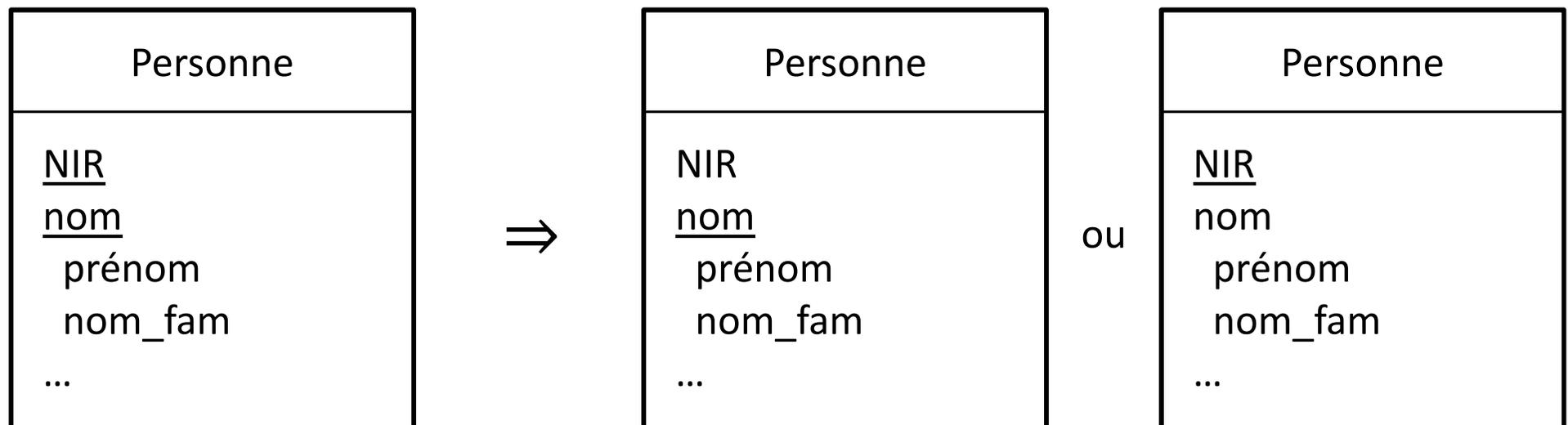
□ Définition (cf. déf. clé minimale)

« Si la clé d'une entité (faible ou non) est constituée de plus d'un attribut alors il n'existe pas au sein de ce groupe d'attributs un sous-groupe qui forme une clé »

□ Transformation

- Utiliser le sous-groupe comme clé primaire (ou discriminant)
- Attention, le choix de la clé est primordial pour la future BD

□ Exemple



Éléments de conception – Simplification E/A

■ Mise en évidence des attributs dérivables

□ Définition (cf. attribut dérivable)

« Un attribut est dérivable si sa valeur peut être calculée à partir de la valeur d'autres attributs »

□ Transformation

- Créer une contrainte d'intégrité donnant la règle de calcul

Éléments de conception – Simplification E/A

■ Élimination des éléments redondants

□ Définition

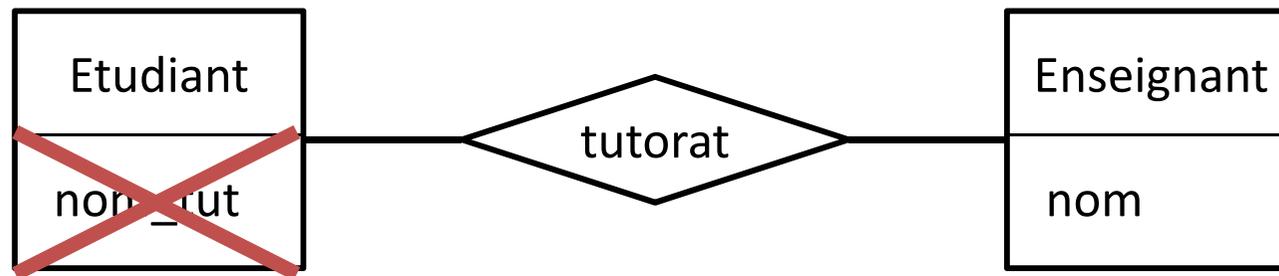
« unicité de l'obtention structurelle d'une information »

□ Transformation

■ Supprimer la redondance et restructurer le diagramme

■ Deux types possibles

□ Attribut redondant à travers une association



Éléments de conception – Simplification E/A

■ Élimination des éléments redondants

□ Définition

« unicité de l'obtention structurelle d'une information »

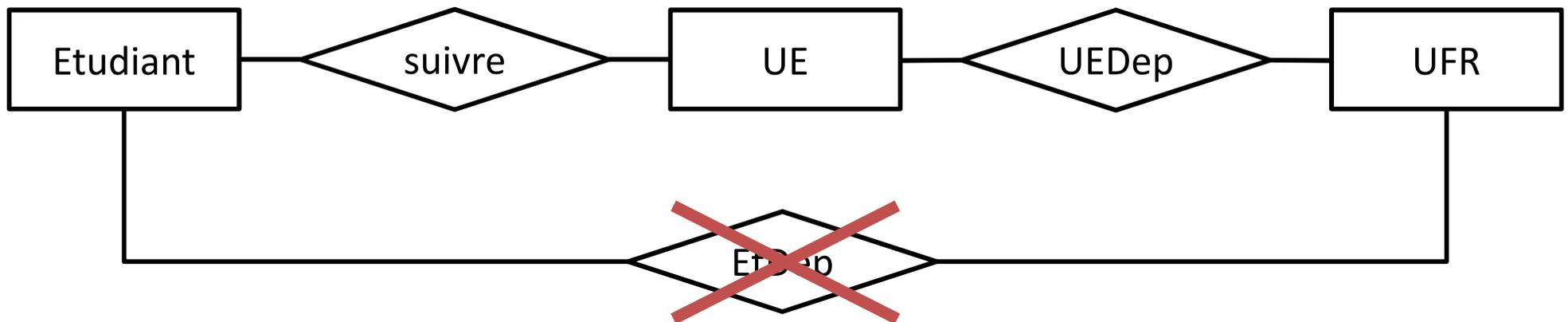
□ Transformation

■ Supprimer la redondance et restructurer le diagramme

■ Deux types possibles

□ Attribut redondant à travers une association

□ Association redondante à travers d'autres associations



Éléments de conception – Simplification E/A

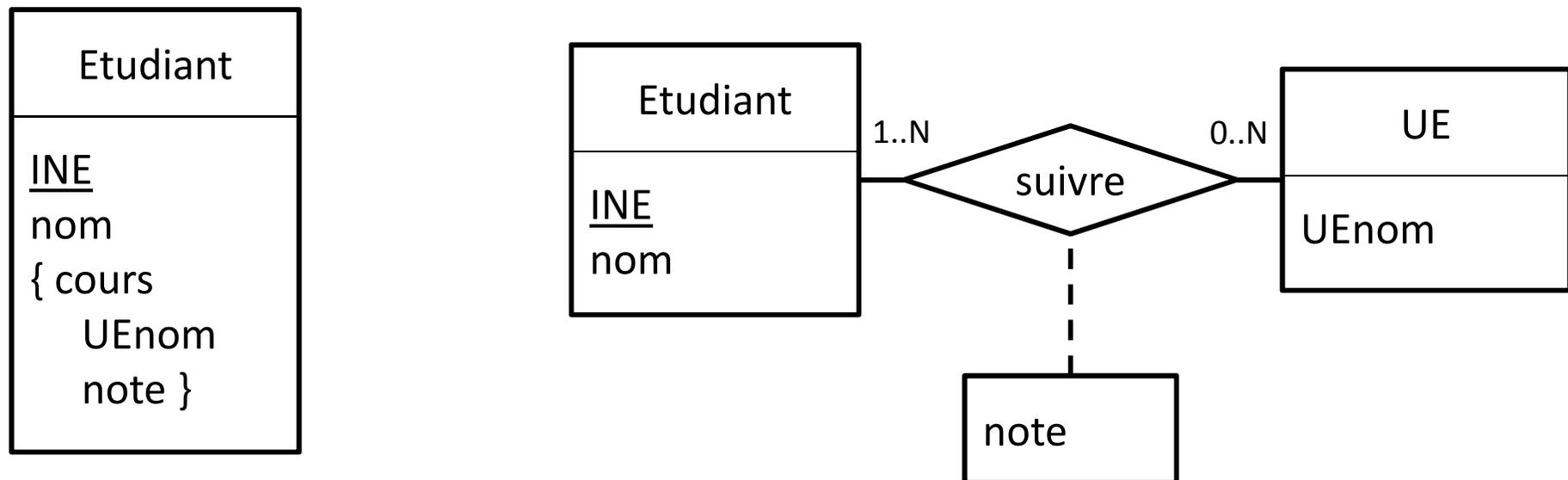
■ Désagrégation d'une entité

□ Définition

« Une entité est *désagrégable* lorsqu'au moins un de ses attributs exprime un objet de la réalité perçue représentable sous la forme d'une association ou d'une entité »

□ Transformation

■ Cas d'un attribut composite ou multivalué



Éléments de conception – Simplification E/A

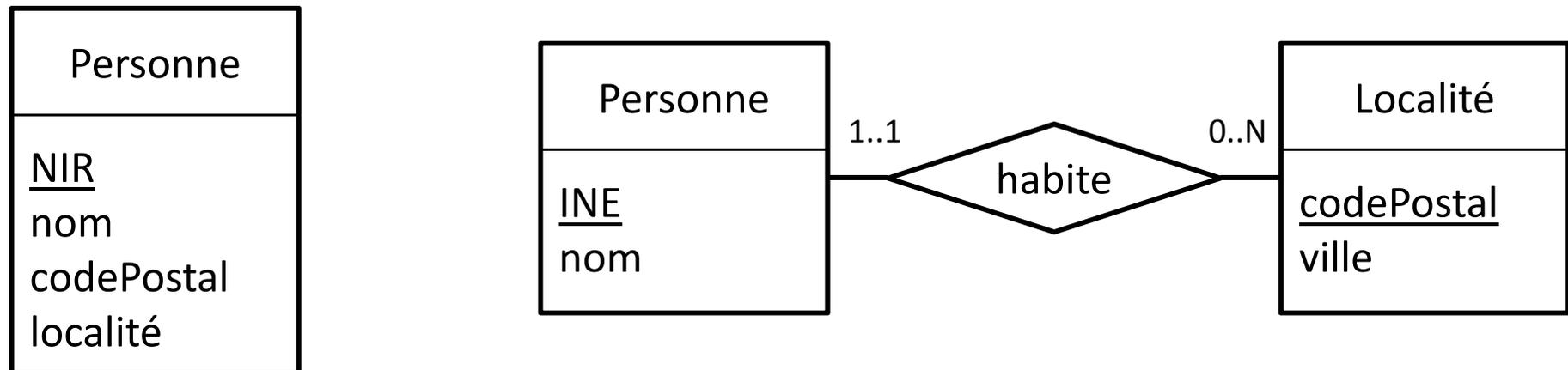
■ Désagrégation d'une entité

□ Définition

« Une entité est *désagrégable* lorsqu'au moins un de ses attributs exprime un objet de la réalité perçue représentable sous la forme d'une association ou d'une entité »

□ Transformation

- Cas d'un attribut composite ou multivalué
- Cas d'une dépendance entre attributs (autre que la clé)



Éléments de conception – Simplification E/A

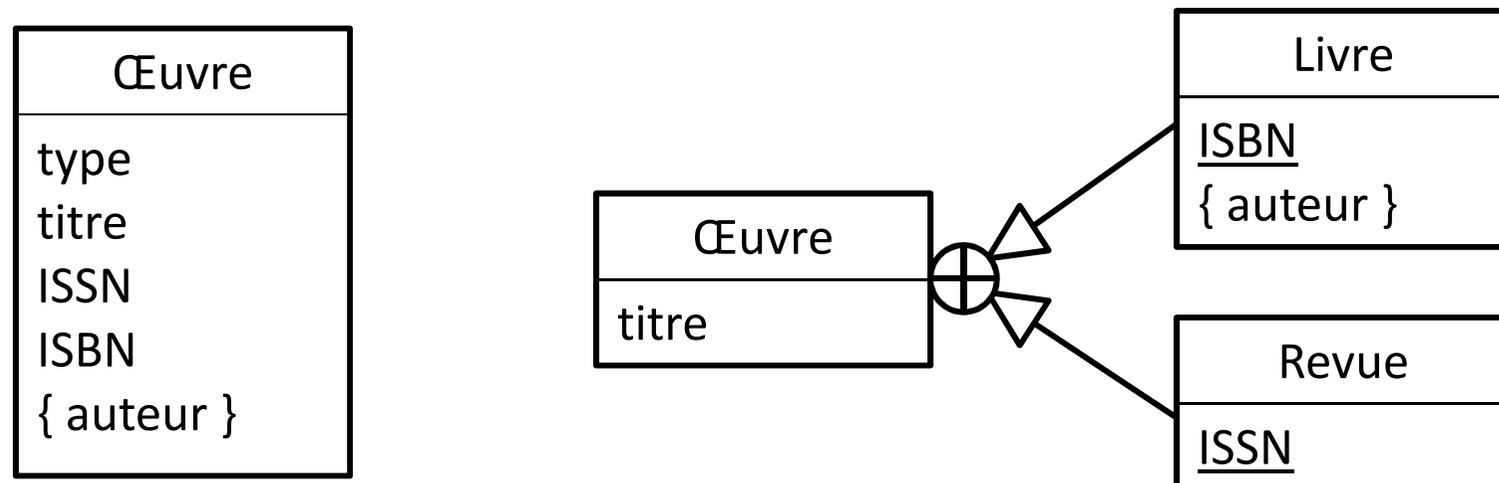
■ Désagrégation d'une entité

□ Définition

« Une entité est *désagrégable* lorsqu'au moins un de ses attributs exprime un objet de la réalité perçue représentable sous la forme d'une association ou d'une entité »

□ Transformation

- Cas d'un attribut composite ou multivalué
- Cas d'une dépendance entre attributs (autre que la clé)
- Cas d'un attribut prenant la valeur null suivant la valeur d'autres attributs



Éléments de conception – Simplification E/A

■ Désagrégation d'une association

□ Définition

Analogue à celle d'une association

□ Transformation

■ Processus plus complexe

■ Création de nombreuses entités

Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

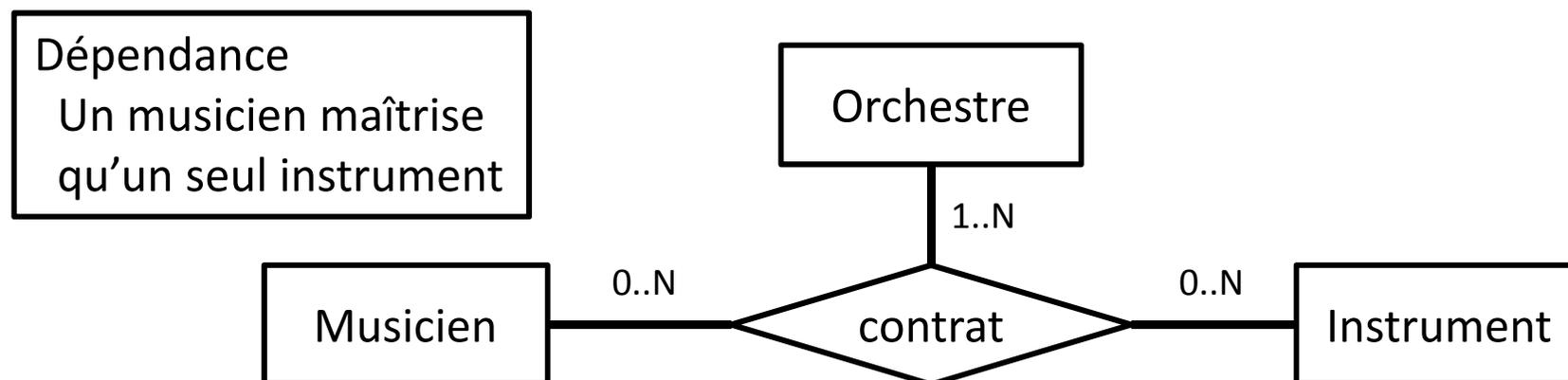
« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

- Pas de cardinalité [?..1] et au moins une dépendance entre rôles traduit souvent la présence d'une imbrication deux associations



Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

- Pas de cardinalité [$?..1$] et au moins une dépendance entre rôles traduit souvent la présence d'une imbrication deux associations



Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

□ Pas de cardinalité [?..1] et au moins une dépendance entre rôles traduit souvent la présence d'une imbrication deux associations

$\left\{ \begin{array}{l} \text{Soit un ensemble d'associations } R(E_1, \dots, E_n) \\ \forall i \text{ la cardinalité maximale de } E_i \text{ dans } R \text{ est } N \\ \text{Soit } i, j \text{ tel que } E_j \text{ dépend de } E_i \text{ dans } R \end{array} \right. \Rightarrow \left\{ \begin{array}{l} R_1(E_i, E_j) \\ R_2(E_1, \dots, E_{j-1}, E_{j+1}, \dots, E_n) \end{array} \right.$

Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

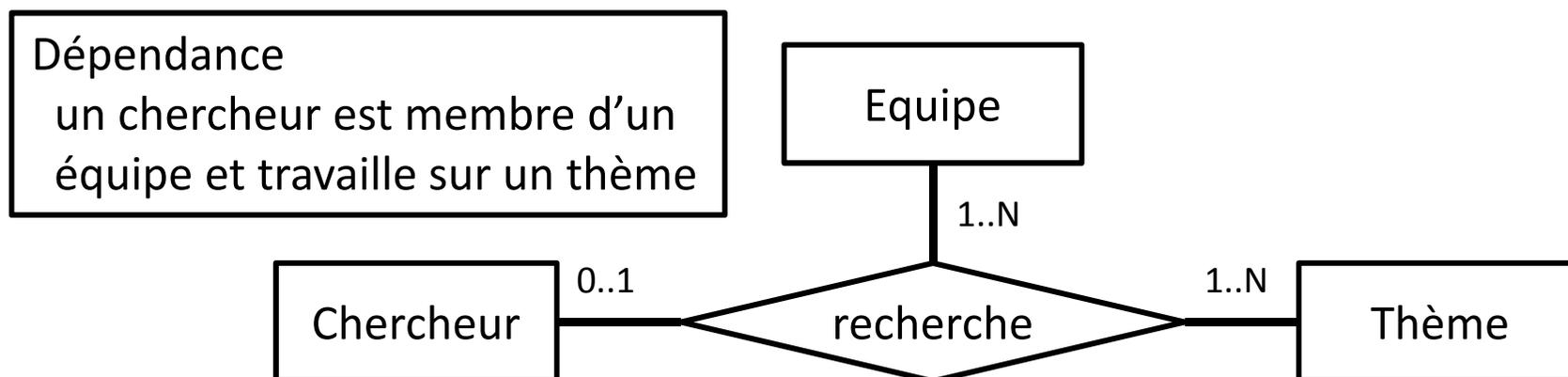
□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

□ Pas de cardinalité [?..1] et au moins une dépendance entre rôles

□ Un rôle avec cardinalité [?..1] et $n - 1$ dépendances avec les autres rôles



Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

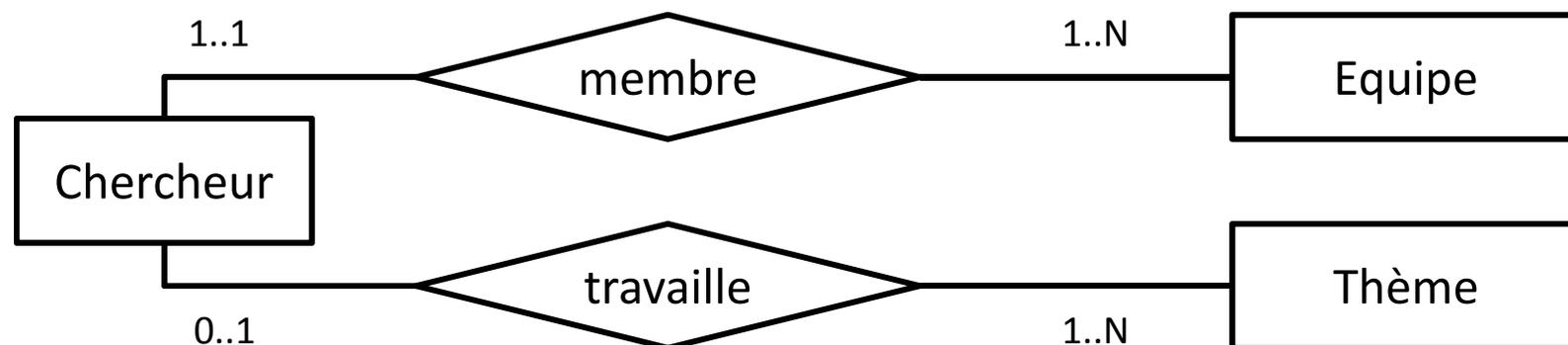
□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

□ Pas de cardinalité [?..1] et au moins une dépendance entre rôles

□ Un rôle avec cardinalité [?..1] et $n - 1$ dépendances avec les autres rôles



Éléments de conception – Simplification E/A

■ Décomposition d'une association

□ Définition

« La **décomposition** d'une association consiste à remplacer, *sans perte d'information*, une association de degré $n \geq 3$ par plusieurs associations dont le degré sera au plus égal à $n - 1$ »

□ Transformation

■ Uniquement si le modèle s'en trouve simplifié

■ Deux types de décomposition

□ Pas de cardinalité [?..1] et au moins une dépendance entre rôles

□ Un rôle avec cardinalité [?..1] et $n - 1$ dépendances avec les autres rôles

$$\left[\begin{array}{l} \text{Soit un ensemble d'associations } R(E_1, \dots, E_n) \\ \exists i \text{ la cardinalité maximale de } E_i \text{ dans } R \text{ est } 1 \\ \forall j \neq i E_j \text{ dépend de } E_i \text{ dans } R \end{array} \right. \Rightarrow \left[\begin{array}{l} R_1(E_i, E_1) \\ \dots \\ R_{i-1}(E_i, E_{i-1}) \\ R_{i+1}(E_i, E_{i+1}) \\ \dots \\ R_n(E_i, E_n) \end{array} \right.$$



-- FIN --