

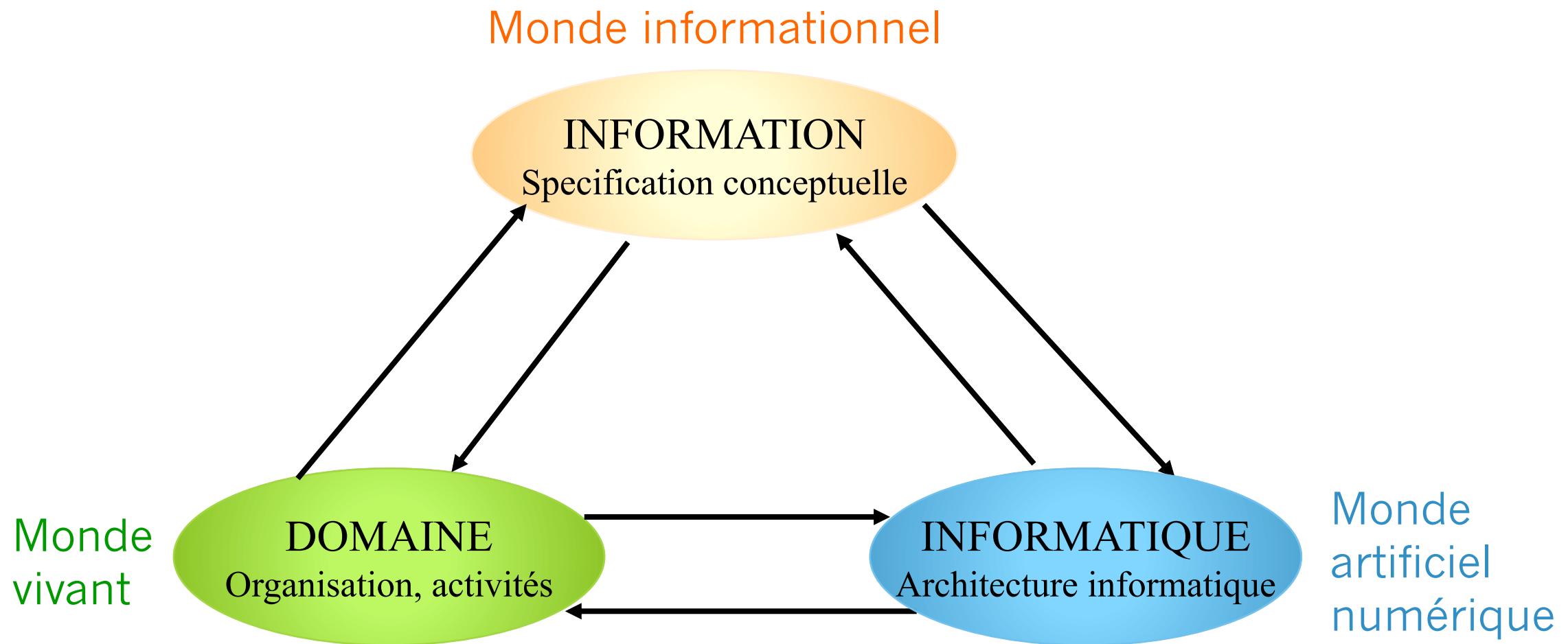
# MSIS Chapitre 4

## Le concept d'association

# Dans ce chapitre

- Qu'est-ce qu'une association? Comment l'identifier?
- Propriétés des associations
- Associations spécifiques
- Représentation des associations en langage formel relationnel

# Identification des associations

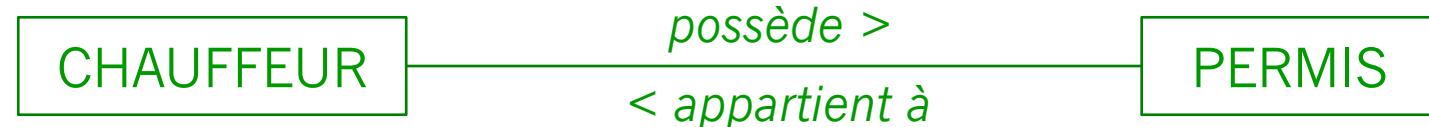


# Association dans le monde informationnel

INFORMATION

Association – une représentation d'un ensemble de liens entre les objets de deux classes ayant la même sémantique et décrits par les mêmes caractéristiques.

Chaque classe joue un rôle déterminé dans l'association.



# Association dans le monde vivant

DOMAINE

Si les classes représentent des concepts du domaine

ex. : Personne, Véhicule

alors les associations entre classes représentent des phrases formées sur ces concepts

ex. : une personne **est propriétaire** d'un véhicule,

une personne **conduit** un véhicule,

une personne **répare** un véhicule...

# Association dans le monde vivant

DOMAINE

De toute phrase décrivant un domaine d'activité, on peut extraire des classes et des associations entre des classes.

« Tel chauffeur conduit tel véhicule, qui est assuré auprès de telle compagnie d'assurances, pour effectuer tel trajet. »

Classes

Véhicule, Chauffeur, Trajet, Compagnie-Assurances

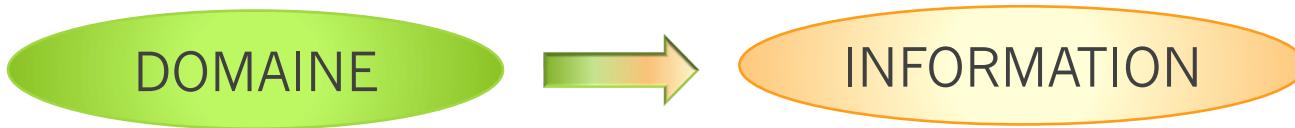
Associations

Conduire entre les classes Chauffeur et Véhicule

Etre assuré entre les classes Véhicule et Compagnie-Assurances

Effectuer formée sur Conduire et Trajet

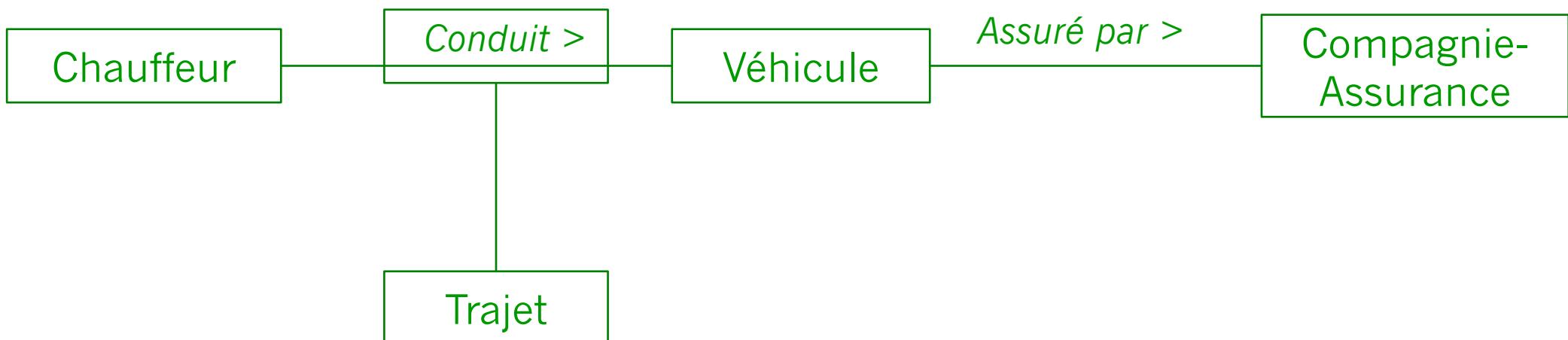
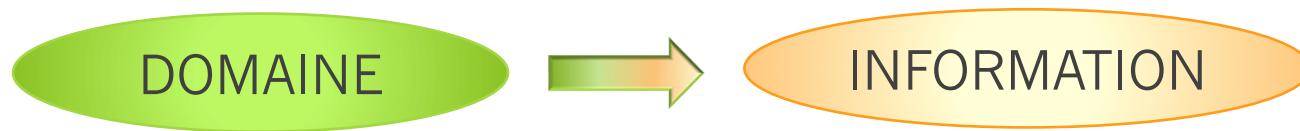
# Identification des associations



Quelles sont les phrases de la description du monde vivant qui donnent lieu à des représentations dans le modèle conceptuel du SI ?

- Les phrases peuvent mettre en jeu plusieurs classes : les associations ne sont pas toujours **binaires**, mais peuvent être **n-aires**.
- Les formulations des phrases peuvent avoir la **forme active ou passive** :  
*Cette compagnie assure ce véhicule.*  
*Ce véhicule est assuré par cette compagnie.*
- Il est possible de construire des **associations sur des associations** :  
*L'association Effectuer formée sur Conduire et Trajet.*

# Identification des associations



# Association dans le monde artificiel

INFORMATIQUE

Représente un chemin d'accès entre objets – des possibilités techniques de relier un objet à un autre objet.

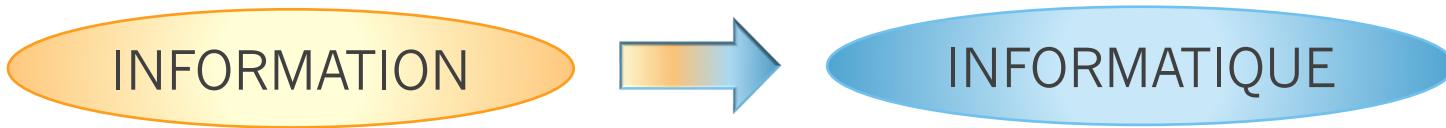
*Peut-on associer tout objet de toute classe à n'importe quel objet de n'importe quelle autre classe ?*

Potentiellement oui.

Restriction en BD : si un objet d'une classe A est associé à un objet de la classe B, alors tout objet de A peut être associé à un objet de B.

Un SGBD permet de naviguer d'objets en objets en passant de classes en classes et en suivant les chemins d'accès.

# Implémentation des associations



Le monde informatique des SGBD utilise des mécanismes d'associations de nature **binaire** :

- un objet d'une classe peut être associé à un ou plusieurs objets d'une autre classe,
- la **composition des associations** permet de connecter les objets de plusieurs classes.

Le monde informatique doit offrir une représentation fidèle aux raisonnements faits autant au niveau informationnel qu'au au niveau du monde vivant!

# Propriétés d'une association

Sémantique de l'association

Cardinalité des rôles d'association

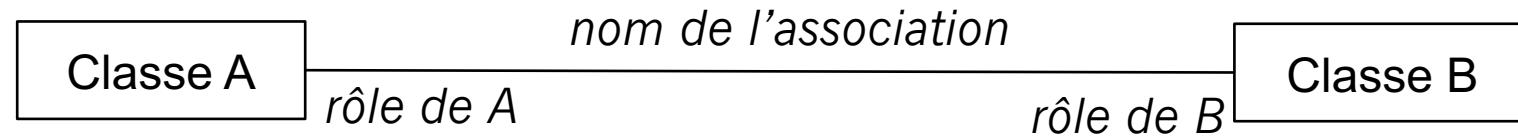
Contraintes entre associations

Transformation d'une association en classe

# Sémantique de l'association

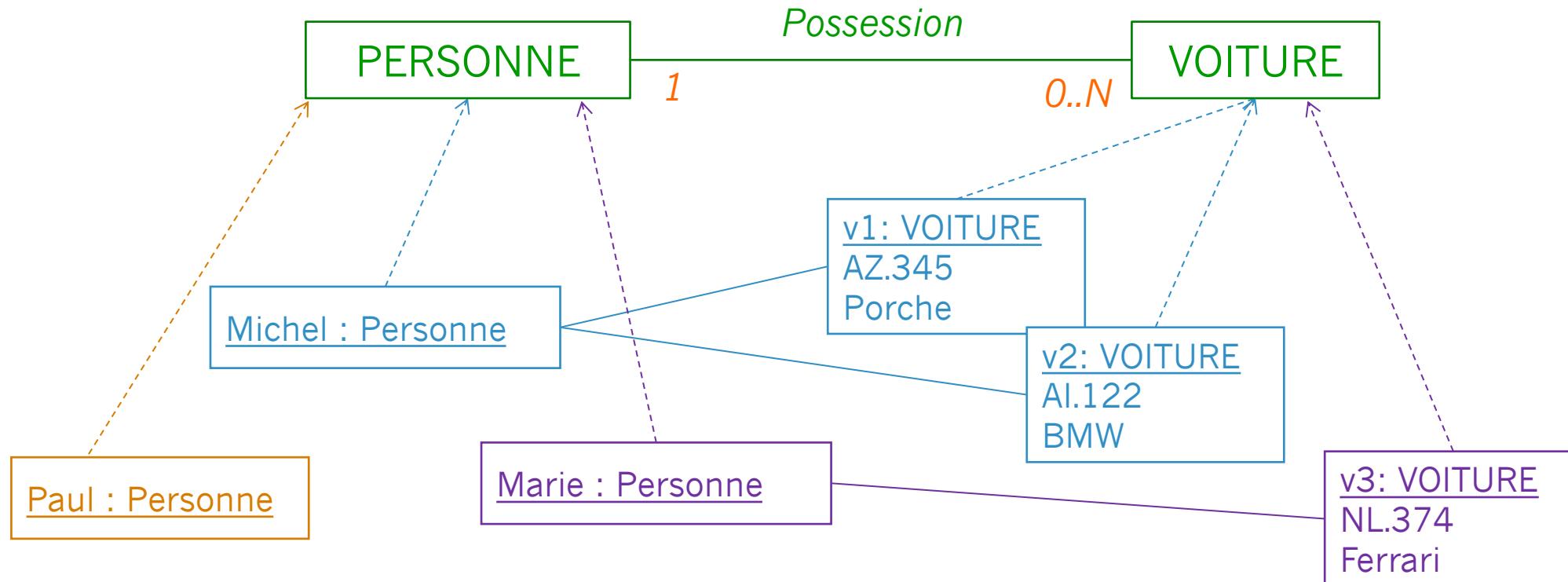
La **sémantique de l'association** est définie par son **nom** et les rôles que jouent les classes participant à l'association.

Pour la clarté du modèle il faut nommer l'association et/ou les rôles des classes.



# Multiplicité de liens dans une association

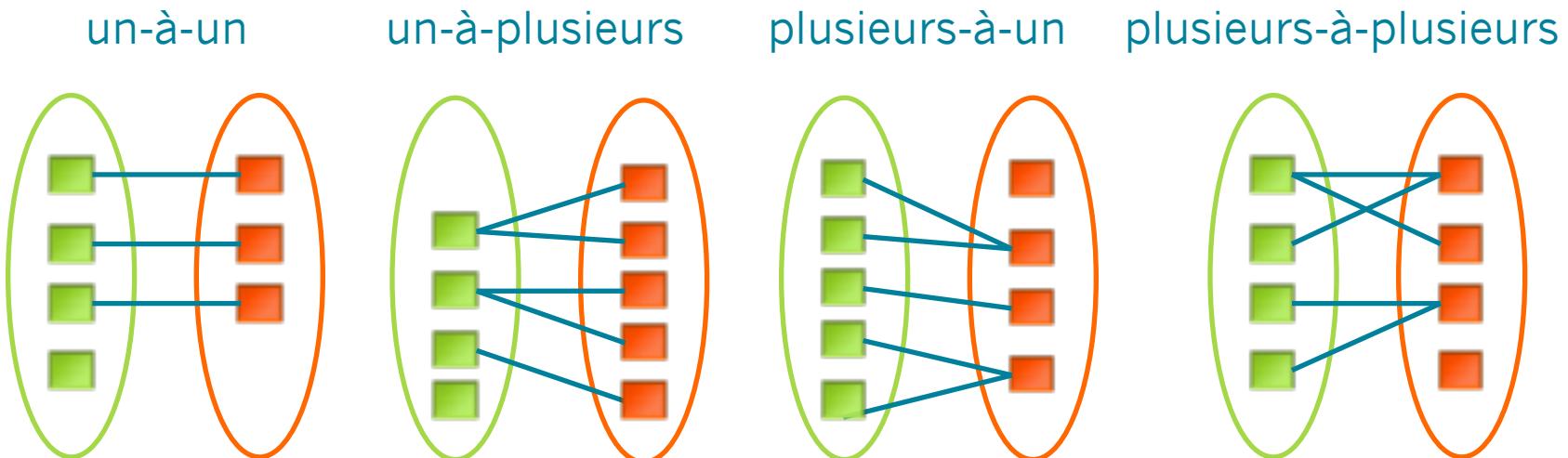
Une association entre deux classes représente de 0 à N liens entre les objets de ces classes.



# Cardinalités des rôles d'associations

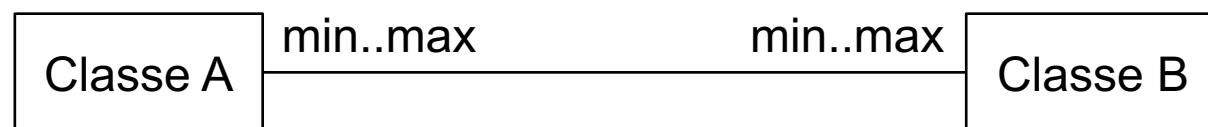
**Cardinalité** – le nombre d'objets avec lesquelles un objet peut être liée via un ensemble de liens.

Pour chaque rôle d'une association on spécifie sa cardinalité.



# Cardinalités des rôles d'associations

Une cardinalité est définie par deux valeurs numériques (min..max)



- min : le nombre de fois minimum qu'un objet participe à une association
  - max : le nombre de fois maximum qu'un objet participe à une association
- 
- Les valeurs possibles sont : (0..1), (1..1), (0..N), (1..N), (M..N) ou N indique un nombre indéterminé >1
  - N et (0..N) peut être remplacé par \* : (\*), (1..\*)



# Exemples des cardinalités



Il faut au moins 3 personnes pour ouvrir un club



On peut s'inscrire au plus 10 cours



Une personne peut posséder un nombre indéterminé de voitures

# Cardinalités doivent être déterminés pour les deux rôles de l'association



Une personne peut être membre de nombre indéterminé de clubs

Il faut au moins 3 personnes pour ouvrir un club



On accepte entre 5 et 20 étudiants par cours

On peut s'inscrire au plus 10 cours



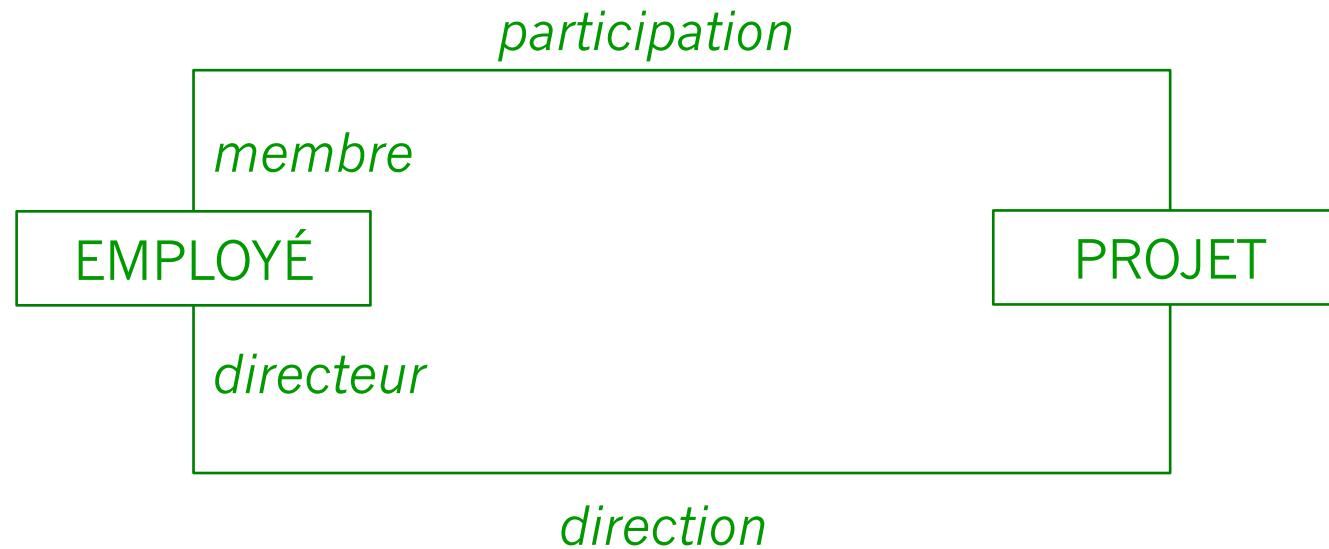
Une voiture appartient toujours à une personne et pas plus

Une personne peut posséder un nombre indéterminé de voitures

# Multiples associations

Il peut exister plusieurs associations entre deux classes.

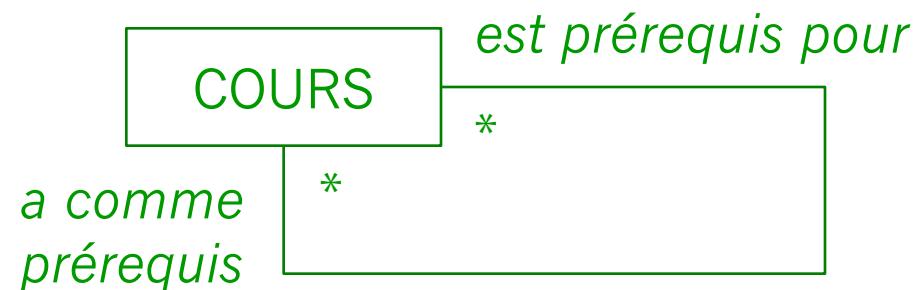
Les noms d'association et/ou des rôles aide à distinguer leur sémantique.



# Association récursive

L'association récursive relie les objets de la même classe entre eux.

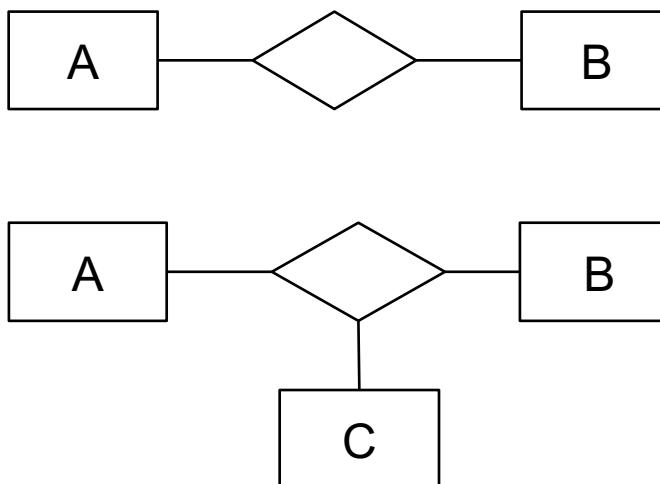
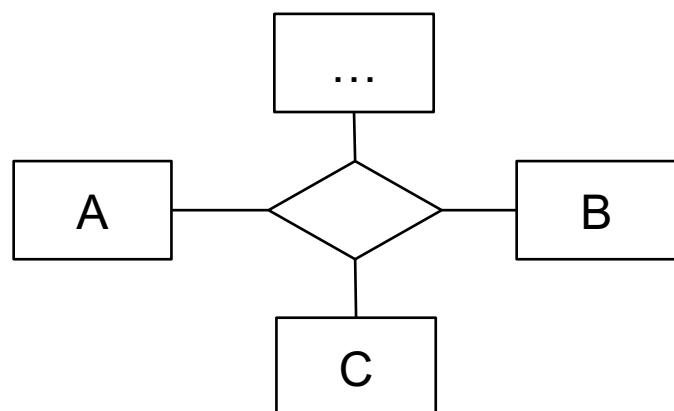
La spécification des rôles est indispensable pour comprendre le sens de l'association est récursive.



# Degré d'association

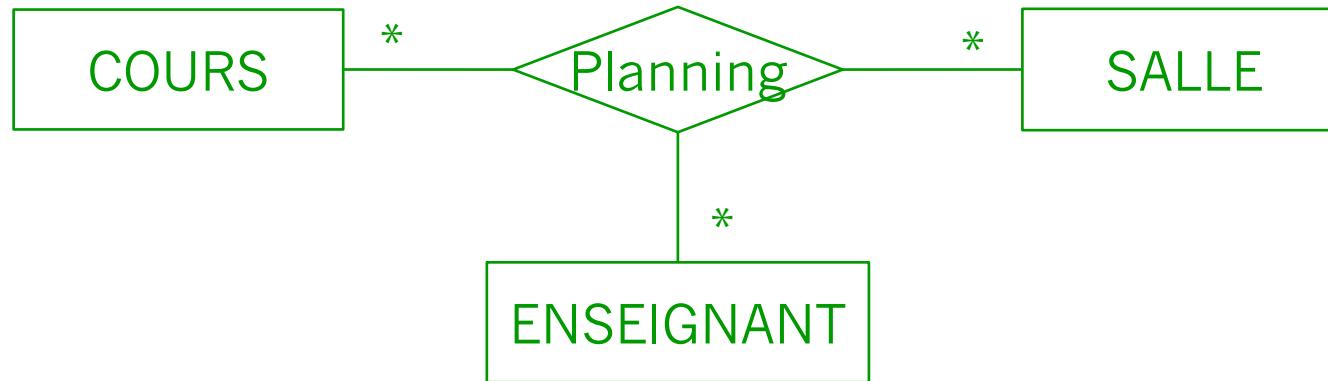
Degré d'une association – le nombre de classes participant dans l'association.

- Binaire = degré 2
- Ternaire = degré 3
- N-aire = degré N



# Exemple d'association ternaire

Un cours peut être donné dans plusieurs salles différentes et par plusieurs enseignants différents.



Peut-on savoir quel enseignant donne quel cours et dans quelle salle?

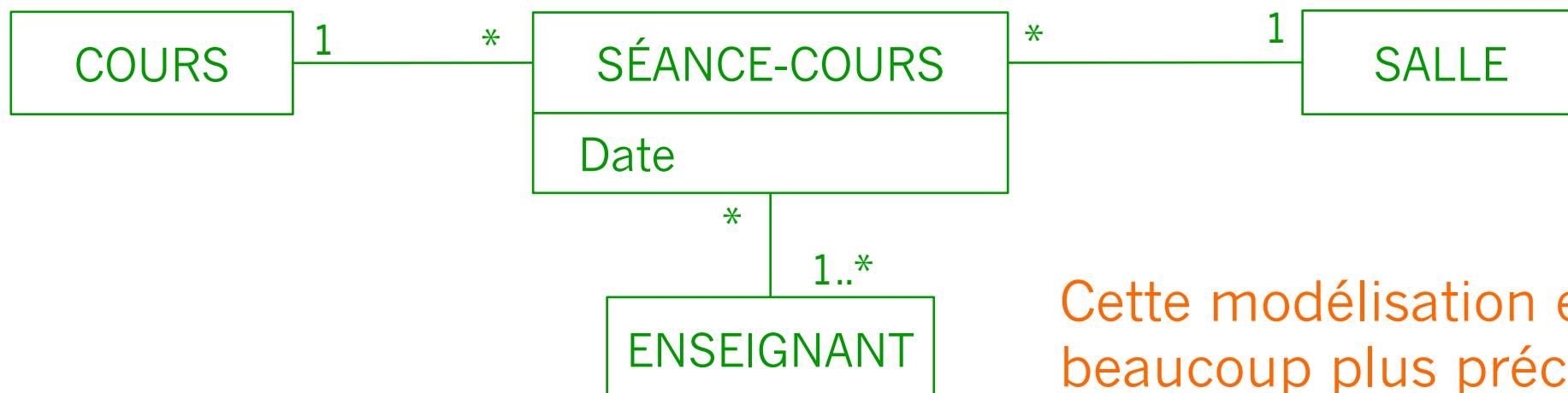
NON !

# Association ternaire vs Classe

On peut toujours transformer une association ternaire /n-aire en une classe.

Forte recommandation !

Les objets de la classe SÉANCE-COURS constituent le planning des cours. Une séance d'un cours à une date et se déroule dans une salle et peut être donnée par un ou plusieurs enseignants.



Cette modélisation est beaucoup plus précise !

# Contraintes entre associations

Ordonnancement



Exclusion



Inclusion



# Comment faire si l'association a des informations propres à elle?

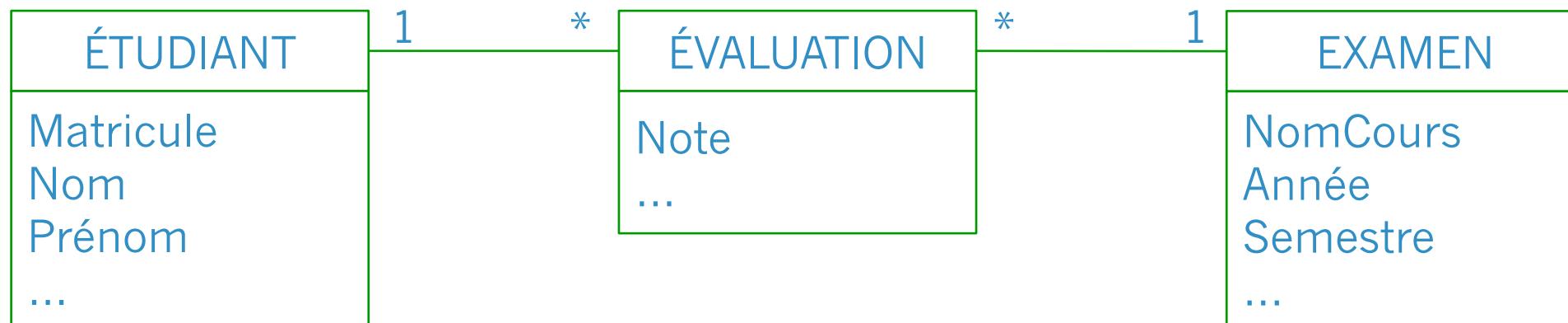
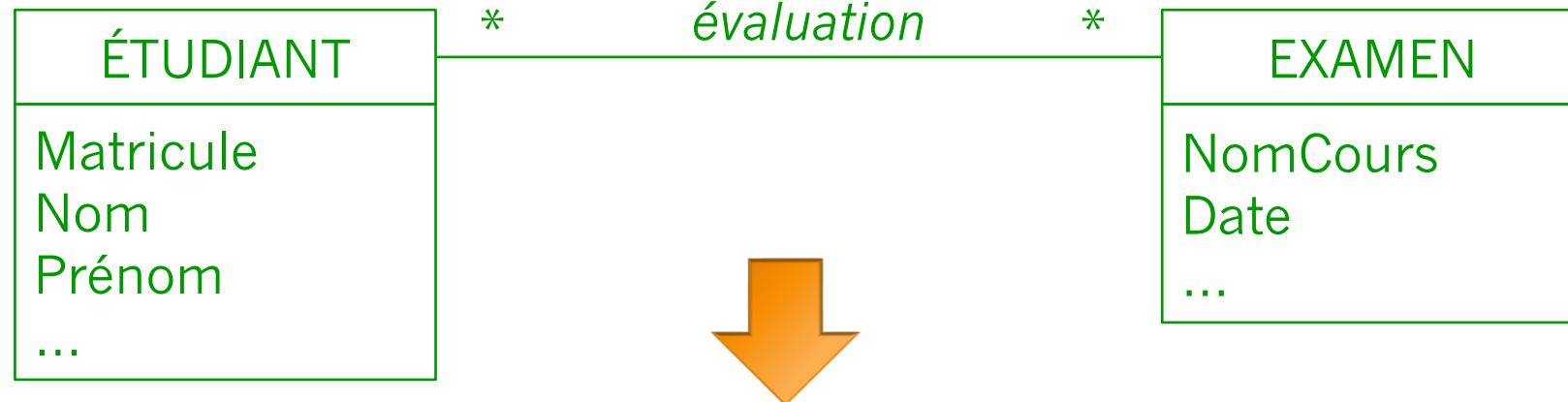


Où est-ce qu'on inscrit la note obtenue par un étudiant pour un examen ?

Option simple : transformer l'association **évaluation** en classe

Option UML : classe d'association

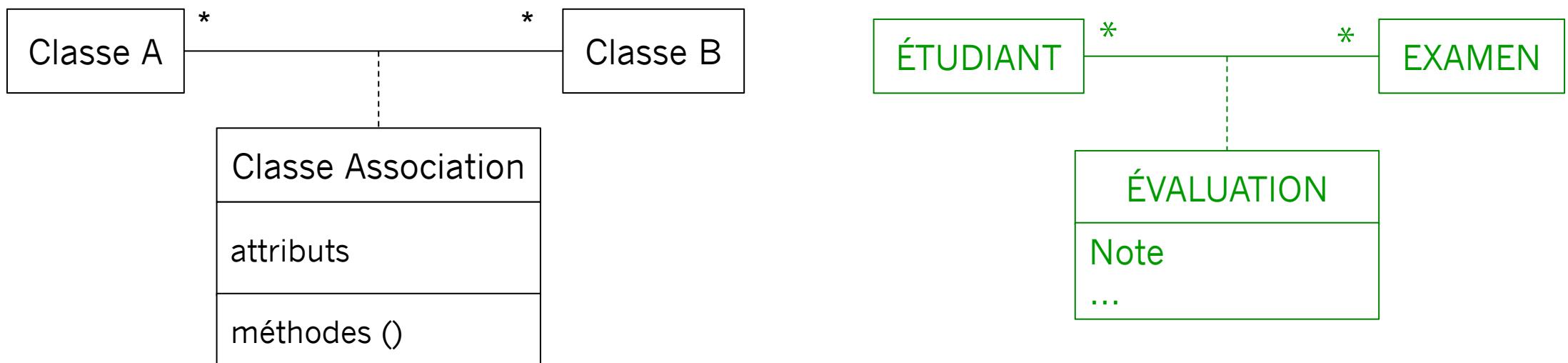
# Transformation d'une association en classe



# Classe d'association (UML)

**Classe d'association** – une association qui est représentée par une classe afin d'y ajouter des attributs et des opérations.

- La cardinalité maximale des deux rôles est N.
- A transformer en une classe normale avec deux associations!



Attention: seuls les outils de modélisation spécifiques UML supportent cette représentation !

# Associations spécifiques

Composition

Agrégation

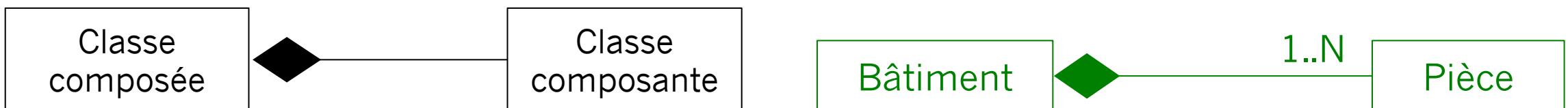
Spécialisation / généralisation

# Composition

Association de type « fait partie de »

Composition – l'objet de la classe composante est contenu dans un objet de la classe A (casse composée).

- Les cycles de vie des objets de la classe composante et du l'objet de la classe composée sont liés – si le composé est détruit, ses composants le sont aussi.
- Un objet composant ne peut pas exister tout seul (dans le système).
- Un objet composant n'appartient qu'à un seul objet composé.

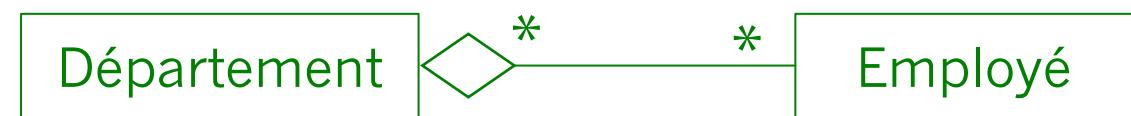
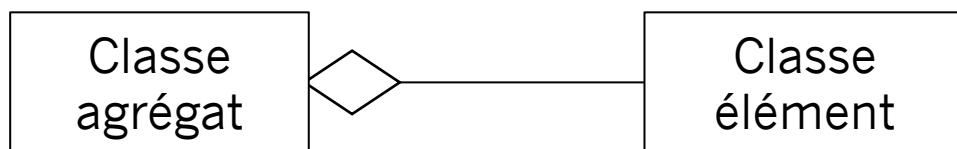


# Agrégation

Association de type « fait partie de », moins forte que la composition.

Agrégation – une association non symétrique dans laquelle une classe joue un rôle prédominant par rapport à l'autre classe.

- Une relation de type « agrégat / élément »
- Un objet de la classe élément peut être lié à plusieurs objets d'autres classes (l'élément peut être partagé).
- Un objet de la classe élément peut exister sans l'agrégat (et inversement) : les cycles de vies de l'agrégat et de ses éléments peuvent être indépendants.



# Composition, agrégation ou association

En modélisation conceptuelle de données **toujours privilégier l'association simple !**

Composition → Association



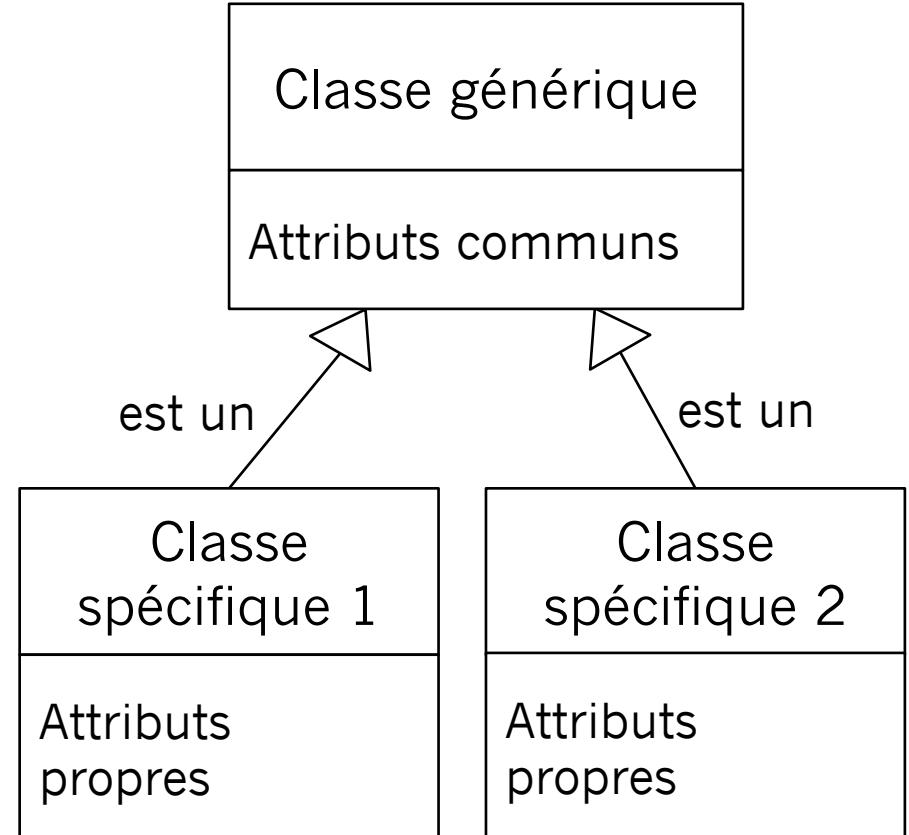
Agrégation → Association



# Spécialisation / généralisation

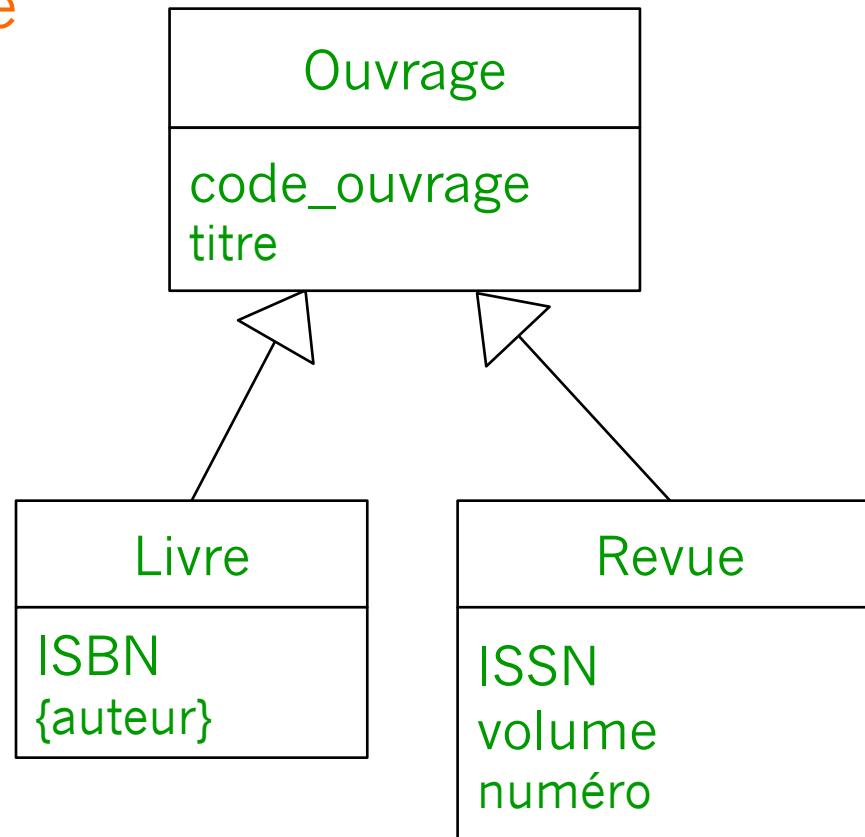
Un moyen de créer une classification des objets similaires

- **Spécialisation** – une association qui permet d'étendre les propriétés d'une classe sous forme de sous-classes, plus spécifiques.
- **Généralisation** – une association qui permet d'extraire les propriétés et les opérations communes à plusieurs classes et de les définir dans une classe générique.



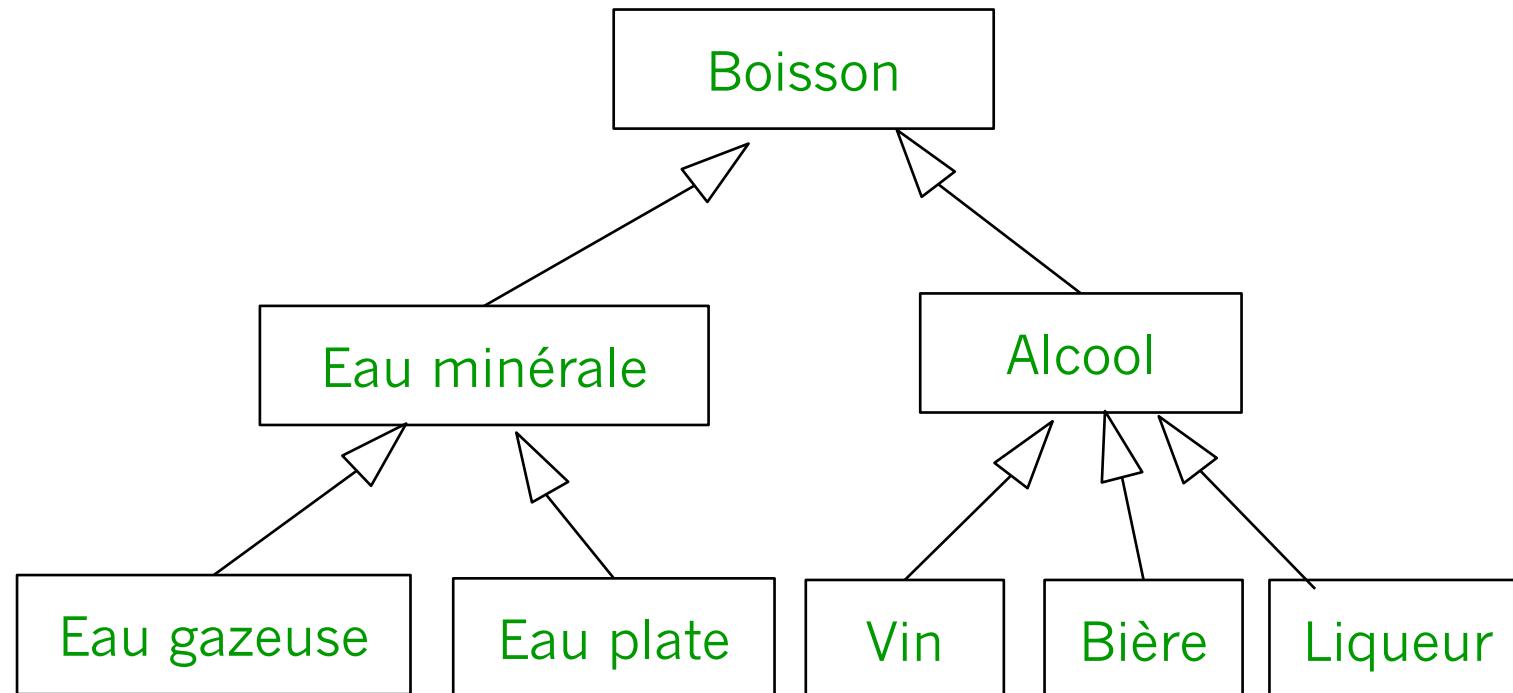
# Spécialisation / généralisation

Un moyen de créer une hiérarchie de classes de plus en plus spécialisées



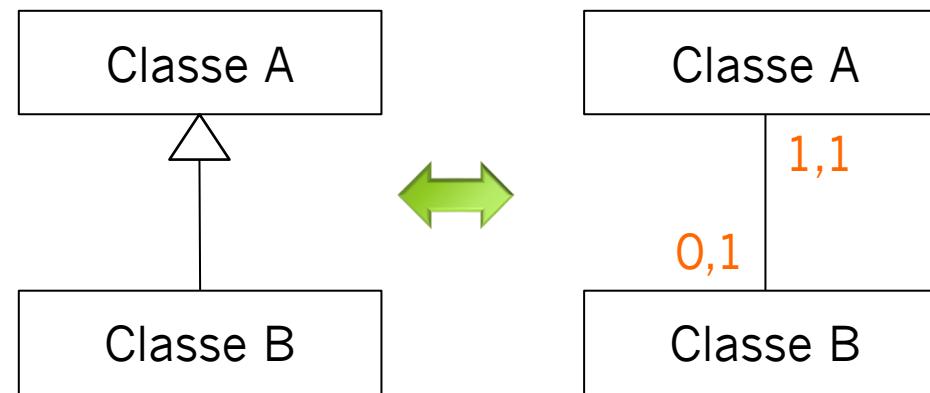
# Spécialisation / généralisation

Un moyen d'organiser un domaine, créer une taxonomie



# Spécialisation / généralisation

Spécialisation / généralisation – un type particulier d'association.



- Tout objet de B est associé à exactement un objet de A de manière permanente.
- Un objet de A est associé au maximum à un objet de B.

# Quel intérêt pour la spécialisation?

## Spécialisation logique ou d'utilisation ?



Spécialisation logique – basée sur des propriétés logiques des éléments du modèle, des règles et des traitements spécifiques sont appliqués en fonction de la spécialisation.

*Client\_Sénior est une spécialisation de Client – les clients de plus de 65 ans ont des priviléges spécifiques.*



Contrainte d'utilisation – la condition est gérée de façon extérieure (lors de l'utilisation de la BD).

*Les participants d'une conférence sont demandés de désigner leur régime alimentaire lors de l'inscription à la conférence.*

→ pas de spécialisation des participants par régime alimentaire, car cette information n'a aucun impact sur le traitement des participants.

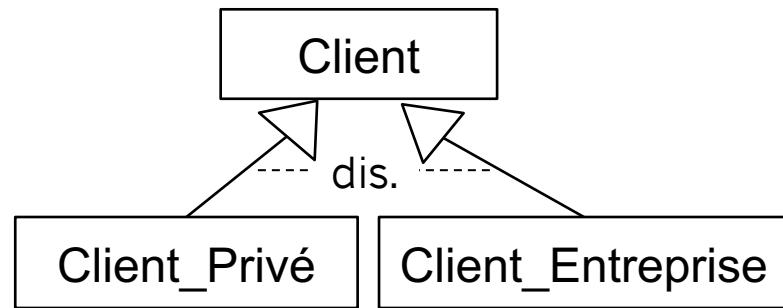
# Contraintes de spécialisation

Spécialisation disjointe ou chevauchante ?

Un objet peut-il appartenir à plusieurs spécialisations ou à une seule ?

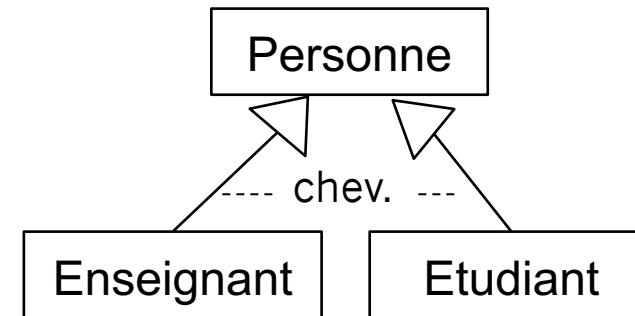
**Disjointe** – un objet ne peut appartenir qu'à une seule des spécialisations.

*Le même client ne peut être à la fois privé et entreprise.*



**Chevauchante** – un objet peut appartenir à plusieurs spécialisations.

*Une personne peut être à la fois enseignant et étudiant.*



# Contraintes de spécialisation

Spécialisation totale ou partielle ?

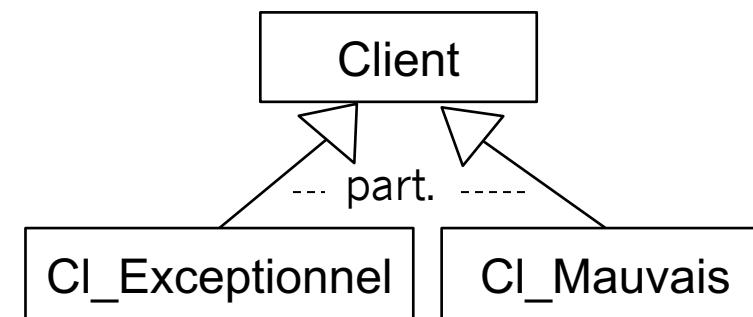
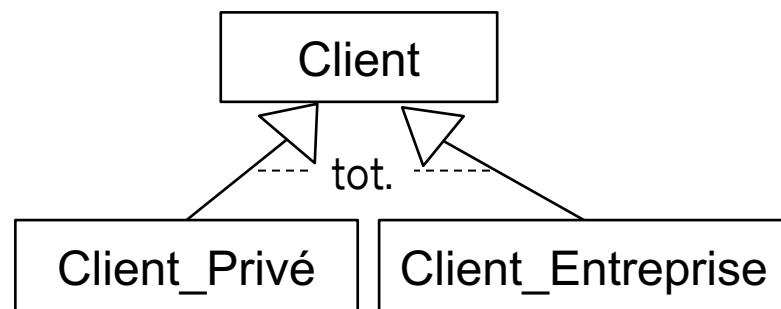
Un objet doit-il forcément appartenir à l'une des spécialisations ?

**Totale** – un objet doit absolument appartenir à l'une des spécialisations.

*Tout client est soit privé, soit entreprise.*

**Partielle** – un objet ne doit pas forcément être spécialisé.

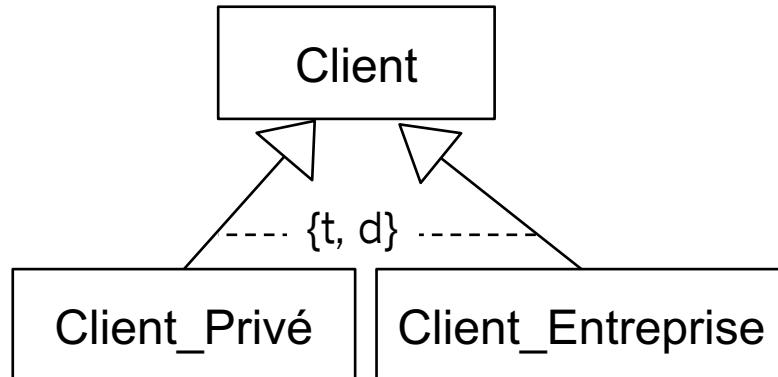
*Un client peut être ni exceptionnel ni mauvais.*



# Exemples de spécialisations

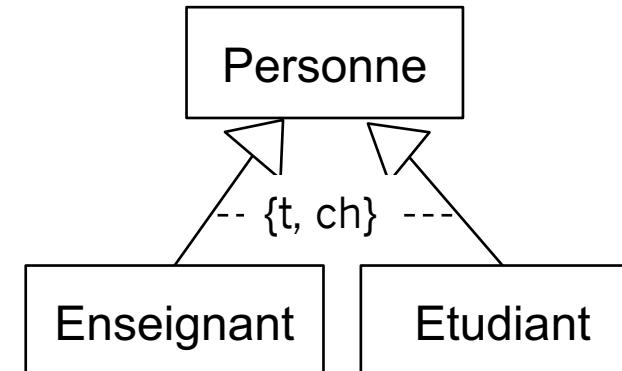
## Totale et disjointe

*Tout client est soit privé, soit entreprise, jamais les deux.*



## Totale et chevauchante

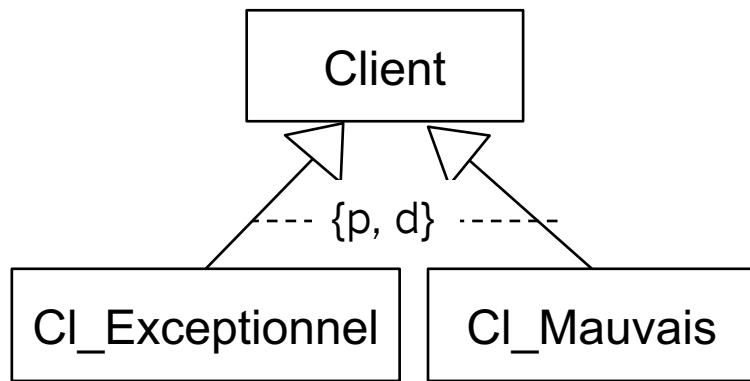
*Toute personne est soit enseignant, soit étudiant, soit les deux.*



# Exemples de spécialisations

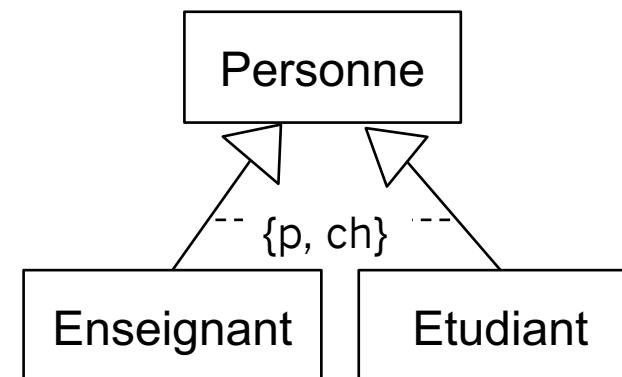
## Partielle et disjointe

- *Un client peut être classé comme exceptionnel ou comme mauvais, mais pas les deux à la fois.*
- *Les autres clients ne sont placés dans aucune des catégories.*



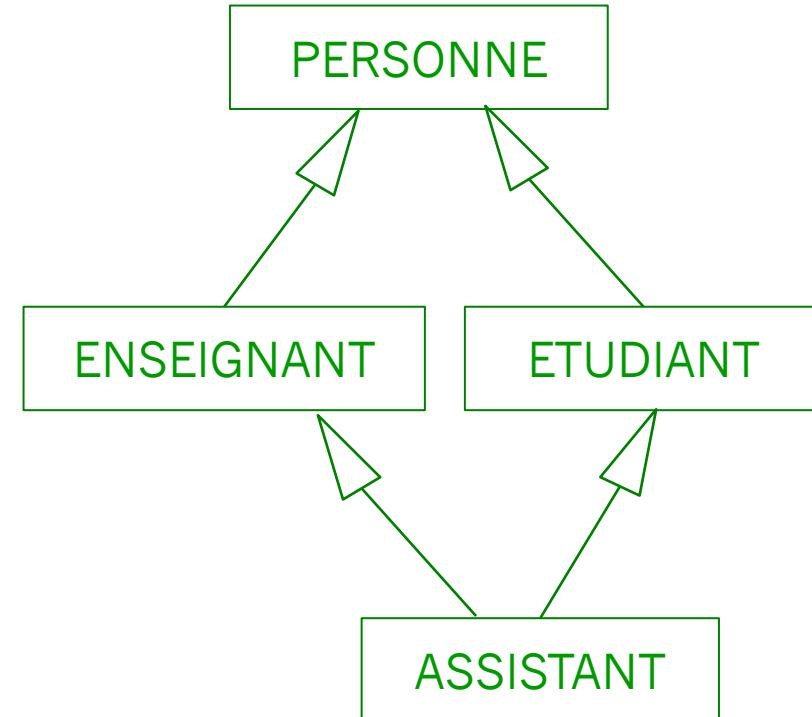
## Partielle et chevauchante

- *Une personne est soit un enseignant, soit un étudiant, soit non classée dans aucune de ces deux catégories (ex. secrétaire, comptable).*
- *Elle peut appartenir aux deux catégories à la fois.*



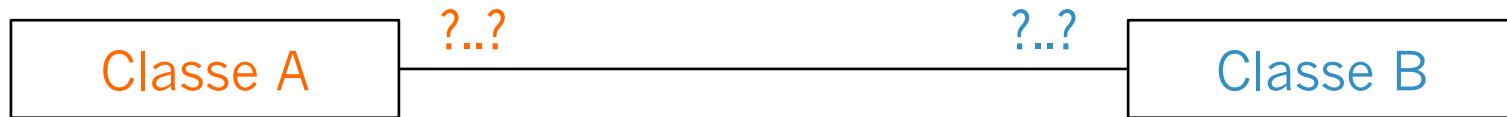
# Spécialisation multiple

Une classe peut être spécialisation de plusieurs classes.



# Représentation des associations en langage formelle relationnel

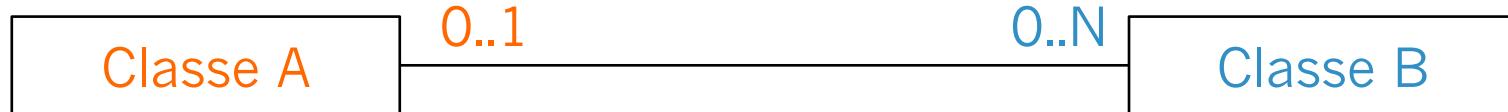
# Définition générique



Deux classes **A** et **B** sont reliées par une association et  
cléA est la clé de **A** et **A1, A2, A3,...** sont les attributs de **A**  
cléB est la clé de **B** et **B1, B2, B3,...** sont les attributs de **B**

La définition de l'association dépend des cardinalités de l'association.  
*A voir cas par cas dans les pages suivantes.*

# Association simple 0..1 – 0..N



Si deux classes **A** et **B** sont reliées par une association ayant comme cardinalités **0..1** du côté de A et **0..N** du côté de B, alors la définition de la classe **B** doit inclure la clé de la classe **A**.

A (**cléA** // A1, A2, A3,...)

B (**cléB** // **cléA**, B1, B2, B3,...)

Dans la définition de la classe **B** :

- **cléA** est appelée **clé étrangère** ou **contrainte de référence** ou encore **dépendance référentielle**,
- la valeur de **cléA** peut être obscure (inconnue) – un objet de B peut exister sans être relié à un objet de A.

# Association simple 0..1 – 0..N

Exemple : catalogue de formations d'une école

Le catalogue peut contenir des formations qui n'ont pas de responsable attribué (ex. la formation n'est pas ouverte)

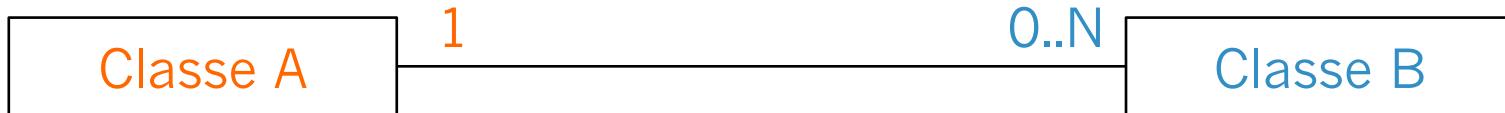


**RESPONSABLE** (`IdResp // Nom, Prénom, email,...`)

**FORMATION** (`Titre // IdResp, Durée, Description, Crédits,...`)

*ici la valeur de `IdResp` peut être obscure*

# Association 1 – 0..N



Si deux classes **A** et **B** sont reliées par une association ayant comme cardinalités **1** du côté de A et **0..N** du côté de B, alors la définition de la classe B doit inclure la clé de la classe A dont la valeur doit être claire pour tous les objets de B.

A (**cléA** // A1, A2, A3,...)

B (**cléB** // **cléA**, B1, B2, B3,...)

Dans la définition de la classe **B**:

- la valeur de **cléA** doit être **claire** – un objet de B ne peut pas exister sans être relié à aucun objet de A et il ne peut être relié qu'à un seul objet A.



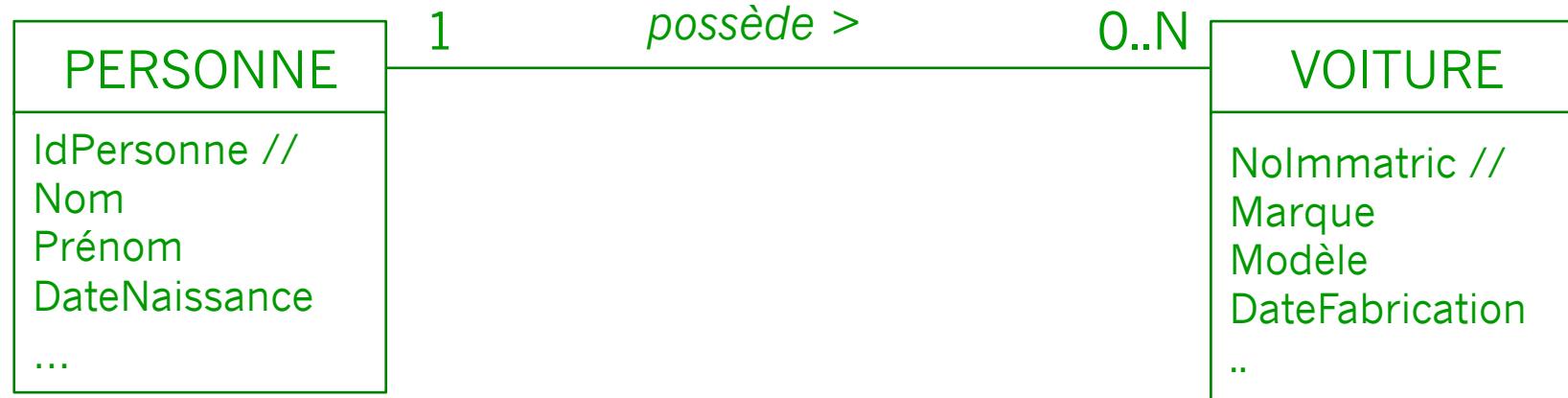
Attention à la cohérence de données

# Association simple 1 – 0..N

Exemple : gestion des immatriculations des voitures par le service des automobiles

Toute voiture immatriculée doit être reliée à une et une seule personne considérée comme propriétaire.

Toute personne peut posséder plusieurs voitures ou aucune.



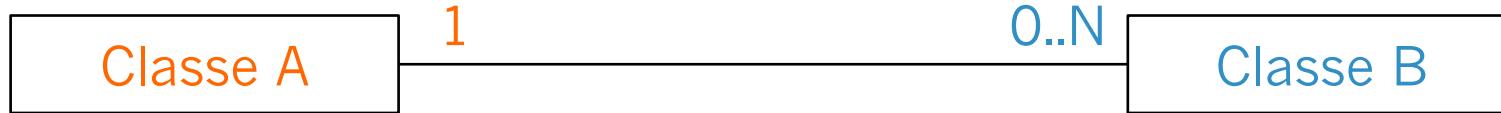
PERSONNE (IdPersonne // Nom, Prénom, DateNaissance,...)

VOITURE (Nolmmatric // **IdPersonne**, Marque, Modèle, DateFabrication,...)

*ici la valeur de IdPersonne doit être claire*

# Association 1 – 0..N

## Dépendance existentielle



La classe **B dépend existentiellement** de la classe **A**. Tout objet de B dépend existentiellement d'un objet de A.

- Un objet de B
  - ne peut pas exister sans être relié à aucun objet de A et ne peut être relié qu'à un seul objet de A,
  - est relié au même objet de A pour toute son existence.
- La clé de A fait partie de la clé de B
  - A (**cléA** // A1, A2, A3,...)
  - B (**cléA attributs cléB** // B1, B2, B3,...)

# Dépendance existentielle

Exemple : gestion des réservations d'hôtel pour une chaîne hôtelière

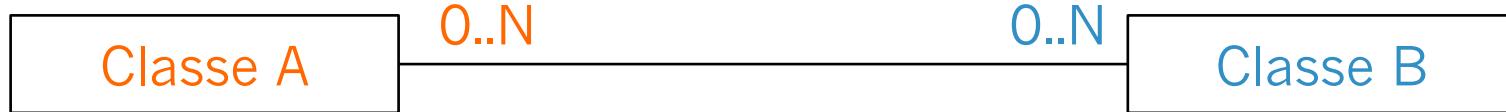
Les chambres sont numérotées de la même façon dans tous les hôtels. Le numéro de chambre ne suffit pas pour identifier une chambre, il faut aussi dire dans quel hôtel elle se trouve.



HOTEL (NomHotel // Catégorie, Adresse,...)

CHAMBRE (NomHotel NumChambre // Type, Description,...)

# Association complexe 0..N – 0..N



Une nouvelle classe **AB** doit être créée pour représenter l'association même si l'association n'a pas d'attributs propres.

La clé de **AB** est composée des clés de **A** et de **B**.

A (**cléA** // A1, A2, A3,...)

B (**cléB** // B1, B2, B3,...)

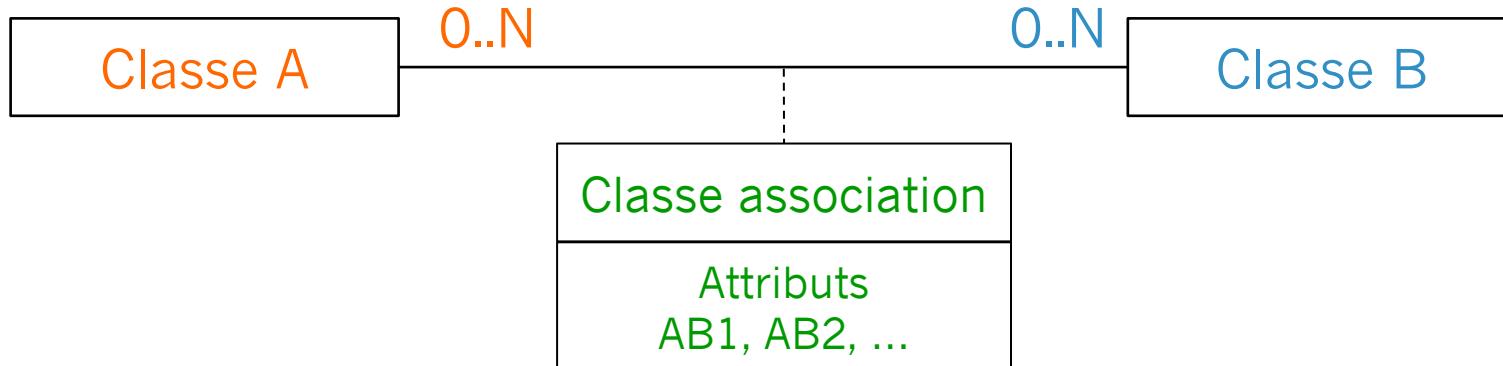
AB (**cléA cléB** //)

ou

AB (**cléA cléB attributs clé AB** //)

Il est nécessaire de compléter la clé de AB par des attributs supplémentaires si un objet de A peut être relié au même objet de B plusieurs fois.

# Association complexe 0..N – 0..N



Quand l'association a des attributs propres à elle – classe association

A (**cléA** // A1, A2, A3,...)

B (**cléB** // B1, B2, B3,...)

AB (**cléA cléB** // **AB1, AB2, ...**)

A (**cléA** // A1, A2, A3,...)

B (**cléB** // B1, B2, B3,...)

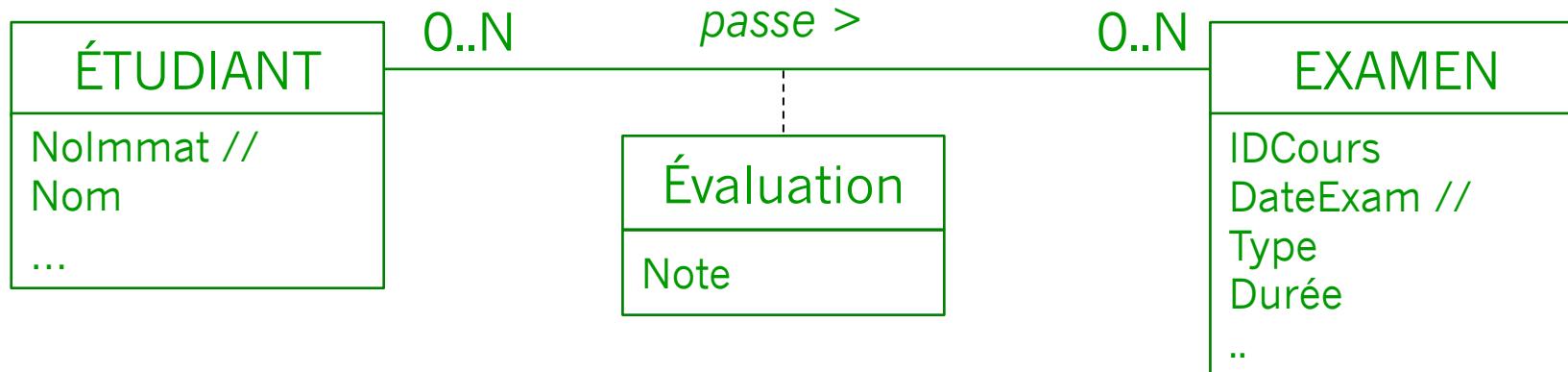
AB (**cléA cléB attributs clé AB** // **AB1, AB2, ...**)

# Association complexe 0..N – 0..N

Exemple : gestion des examens

Chaque étudiant peut passer plusieurs examens et chaque examen est ouvert à plusieurs étudiants.

La note obtenue est un attribut propre à l'association.



ÉTUDIANT (`Nolmat // Nom, ...`)

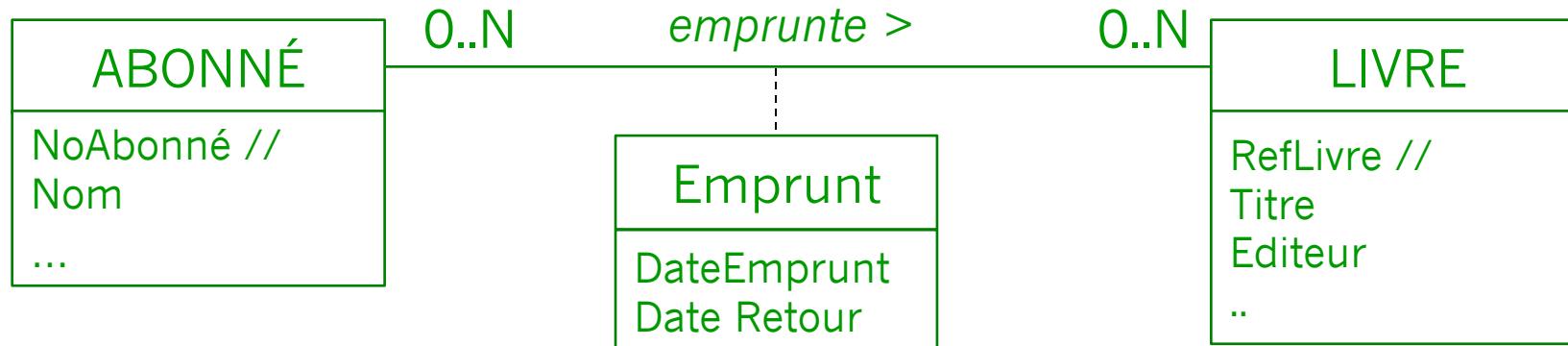
EXAMEN (`IDCours DateExam // Type, Durée, ...`)

ÉVALUATION (`Nolmat IDCours DateExam // Note`)

# Association complexe 0..N – 0..N

Exemple : gestion des emprunts de livres

Un abonné peut emprunter le même livre plusieurs fois.



ABONNÉ (NoAbonné // Nom,...)

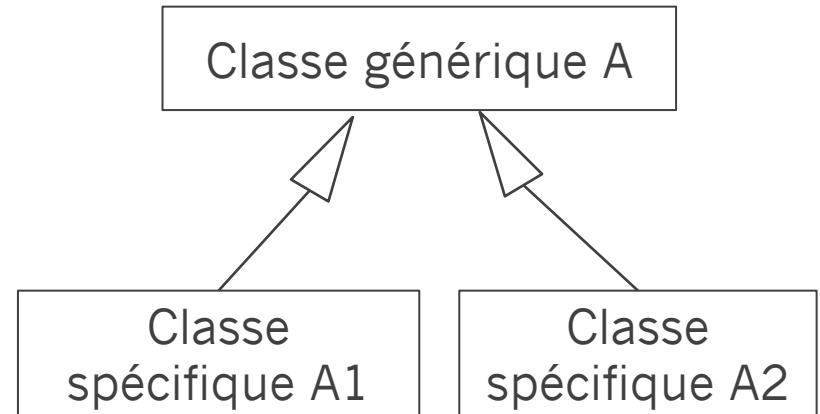
LIVRE (RefLivre // Titre, Editeur,...)

EMPRUNT (NoAbonné RefLivre DateEmprunt // DateRetour)

# Spécialisation / généralisation

Si les classes A1 et A2 sont des spécialisations de la classe A alors A1 et A2 sont en dépendance existentielle avec A.

- Tout objet de A1/A2 doit être relié à un et un seul objet de A2 pour toute son existence.
- La clé de A est également la clé des A1 et A2.
- A1 et A2 peuvent aussi avoir des clés propres.



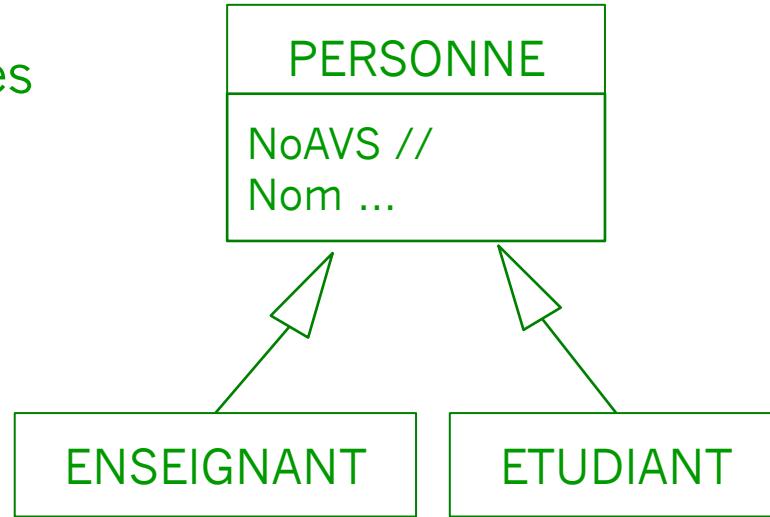
A (**cléA** // A1, A2, A3,...)

A1 (**cléA** / **cléA1** // A11, A12, A13,...)

A2 (**cléA** / **cléA2** // A21, A22, A23,...)

# Spécialisation / généralisation

Exemple : gestion des membres académiques



Classe générique

PERSONNE (NoAVS // Nom, Prénom, email, Adresse, DateNaissance)

Classes spécifiques

ENSEIGNANT (NoEns / NoAVS // DateEmbauche, Position)

ETUDIANT (NoEtudiant / NoAVS // DateInscription, EtatEdutdiant)