

# Base de données

COURS 03 | MODÉLISATION AVANCÉE

## Le modèle entité-association-étendu

- ▶ Le modèle EA déjà présenté ne permet pas d'illustrer certains concepts plus élaborés.
- ▶ Le modèle **entité-association-étendu** (EAE) donne au concepteur des outils supplémentaires permettant :
  - ▶ d'augmenter le niveau de détail du modèle de données;
  - ▶ de préciser des liens importants entre les entités;
  - ▶ de présenter des contraintes essentielles;
  - ▶ de réduire significativement la redondance de données;
  - ▶ d'augmenter la robustesse, la modularité et la flexibilité du modèle créé.

GPA775

3

## Le modèle entité-association-étendu

- ▶ Ainsi, le modèle EAE permet une augmentation importante du niveau d'abstraction du modèle.
- ▶ Néanmoins, il existe certains inconvénients à cet accroissement du niveau d'abstraction. Attention aux découpages abusifs.
- ▶ Il n'existe aucun standard spécifique pour le modèle EAE :
  - ▶ autant pour les concepts couverts;
  - ▶ que pour la notation graphique utilisée.

GPA775

4

## Les constituants du modèle EAE

- ▶ Le modèle EAE est une extension du modèle EA. Ainsi, il inclut tout ce qui existe dans le modèle de base et y ajoute les concepts suivants :
  - ▶ le procédé de généralisation et de spécialisation;
  - ▶ la notion d'héritage (sous-classes et super classes);
  - ▶ la notion de catégorie (union);
  - ▶ L'abstraction d'association;
  - ▶ l'agrégation.

## Modèle EAE

### Procédé de généralisation et de spécialisation

- ▶ La généralisation est le procédé par lequel on factorise les attributs communs d'entités de même nature.
- ▶ Dans le même ordre d'idée, la spécialisation est le procédé inverse permettant de préciser des caractéristiques particulières à des entités similaires mais différentes.
- ▶ Ce procédé peut faussement ressembler aux notions d'entités faibles et fortes.
- ▶ Ces deux procédés sont à la base de la notion plus large d'héritage.
- ▶ Correspond souvent à la structure : EST UN

## Modèle EAE

### Héritage

- ▶ Le concept d'héritage est un concept très puissant de la programmation orienté objet qui permet une très grande réutilisabilité et adaptabilité des objets créés.
- ▶ Dans le cadre spécifique des bases de données relationnelles, le concept d'héritage est directement lié aux notions de généralisation et de spécialisation.
- ▶ Conceptuellement, l'héritage est le principe selon lequel un enfant hérite systématiquement de toutes les caractéristiques de son parent.
- ▶ Le concept permet de propager des informations communes du haut de l'« arbre généalogique » vers le bas tout en ajoutant des spécificités propres aux descendants.

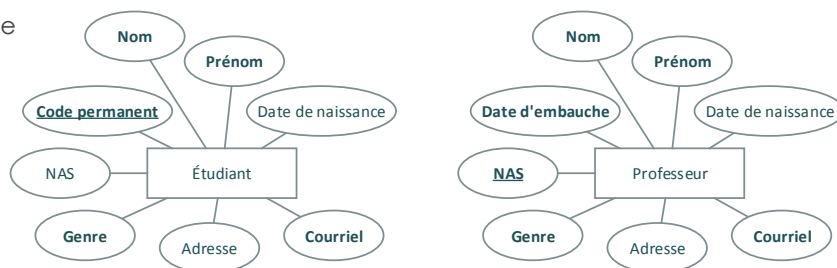
GPA775

7

## Modèle EAE

### Héritage – Super classe et sous-classe

- ▶ Dans le contexte de l'héritage une entité parent est appelé « super classe » alors qu'une entité enfant est appelé sous-classe.
- ▶ L'héritage multiple est possible mais souvent non recommandé à cause des ambiguïtés possibles.
- ▶ Voici un exemple de généralisation :

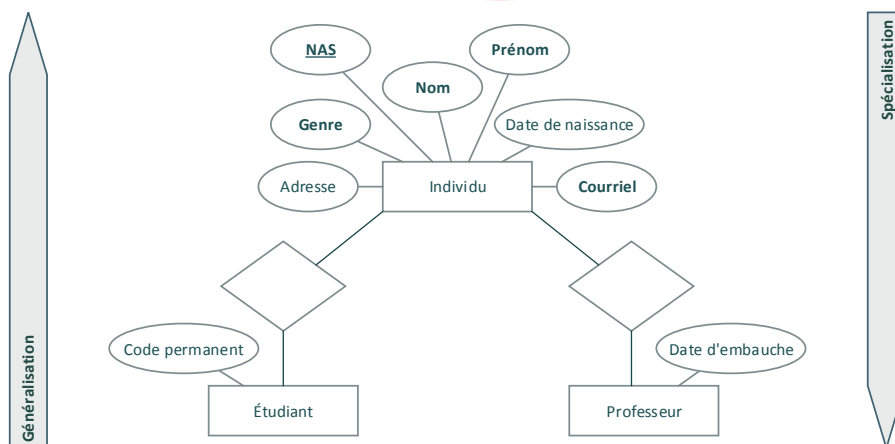


GPA775

8

## Modèle EAE

### Héritage, généralisation et spécialisation



## Modèle EAE

### Héritage, généralisation et spécialisation

- ▶ Spécialisation :
  - ▶ processus de modélisation « *top-down* » ;
  - ▶ consiste à distinguer plusieurs sous-ensembles d'entités à partir d'une entité commune.
- ▶ Généralisation :
  - ▶ processus de modélisation « *bottom-up* » ;
  - ▶ consiste à combiner un ensemble d'entités partageant des caractéristiques communes.
- ▶ Héritage :
  - ▶ généralisation + spécialisation => héritage de classes

## Modèle EAE

### Héritage et contraintes

- ▶ Il existe deux contraintes importantes à préciser pour les processus de généralisation et de spécialisation :
  - ▶ contrainte de cardinalité (disjointe ou avec recouvrement);
  - ▶ contrainte de participation (totale ou partielle).
- ▶ Aussi, une notation particulière est proposée pour chaque contrainte.

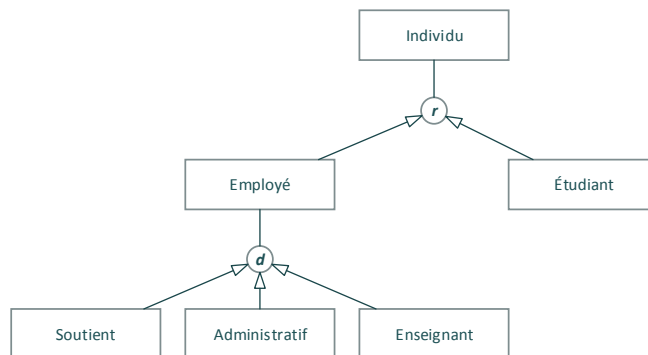
## Modèle EAE

### Héritage et contrainte de cardinalité

- ▶ Cette contrainte précise si une entité parent peut appartenir à un ou plusieurs entités enfants :
  - ▶ généralisation disjointe : l'entité parent ne peut appartenir qu'à une seule entité enfant;
  - ▶ généralisation avec recouvrement (« *overlapping* ») : l'entité parent peut appartenir à plusieurs entités enfants.
- ▶ On utilisera un cercle ayant la notation interne de **d** ou de **r** (pour disjoint ou recouvrement). De plus, une flèche sera ajoutée de l'entité enfant vers le cercle d'identification de contrainte.
- ▶ Les spécialisations singulières ne possèdent pas de cercle mais garde la flèche pour indiqué le lien parent/enfant.

## Modèle EAE

### Héritage et contrainte de cardinalité



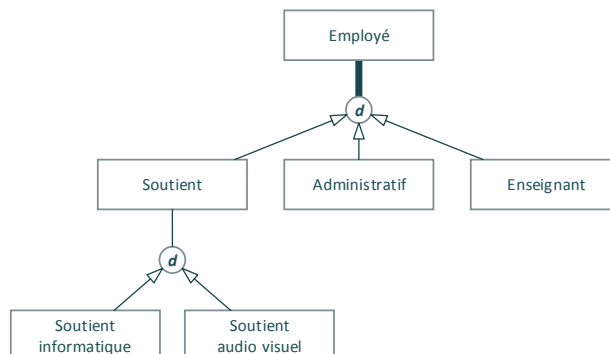
## Modèle EAE

### Héritage et contrainte de participation

- ▶ Cette contrainte précise si une entité parent doit forcément appartenir à l'une des spécialisations :
  - ▶ généralisation avec participation totale : doit appartenir à l'une des spécifications;
  - ▶ généralisation avec participation partielle : ne doit pas forcément être spécialisée.
- ▶ Les spécialisations singulières sont aussi sujettes à cette contrainte.
- ▶ On utilisera un trait épais entre l'entité parent et le cercle indiquant la contrainte de cardinalité.

## Modèle EAE

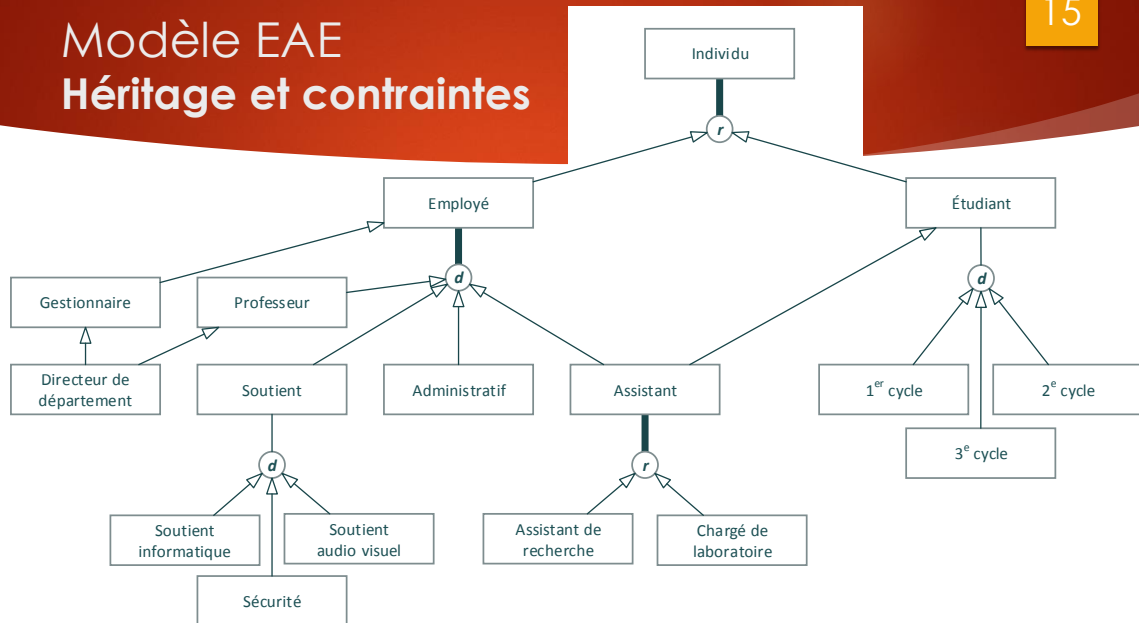
### Héritage et contrainte de participation



GPA775

15

## Modèle EAE Héritage et contraintes



GPA775

16

## Modèle EAE Catégorie

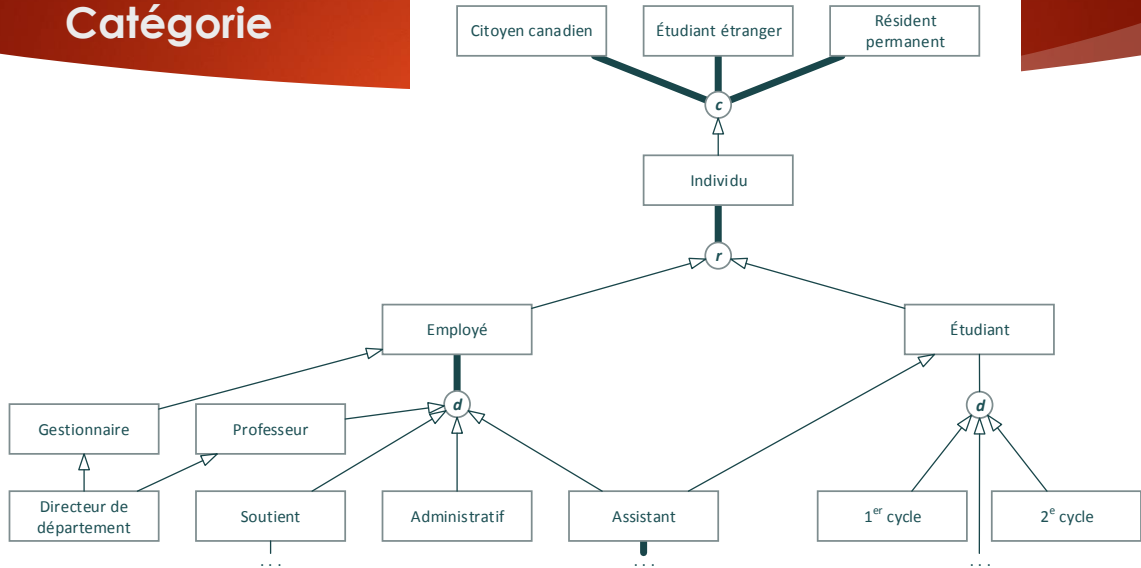
- ▶ L'héritage multiple implique que la sous classe doit absolument posséder un lien vers ses  $n$  super classes.
- ▶ Il est tout de même fréquent d'avoir un héritage exclusif qui est variable selon le contexte. Ainsi, selon l'instance courante, l'héritage peut être de parents différents.
- ▶ Ce type d'héritage est possible avec le modèle EAE : on les nomme union ou plus justement catégorie. Ainsi, il est possible d'avoir une définition formelle de la structure variable de l'héritage.
- ▶ Comme pour les autres types d'héritage, on utilise le cercle mais avec la lettre **C** (catégorisation). La contrainte de participation totale ou partielle est toujours possible.



GPA775

17

## Modèle EAE Catégorie



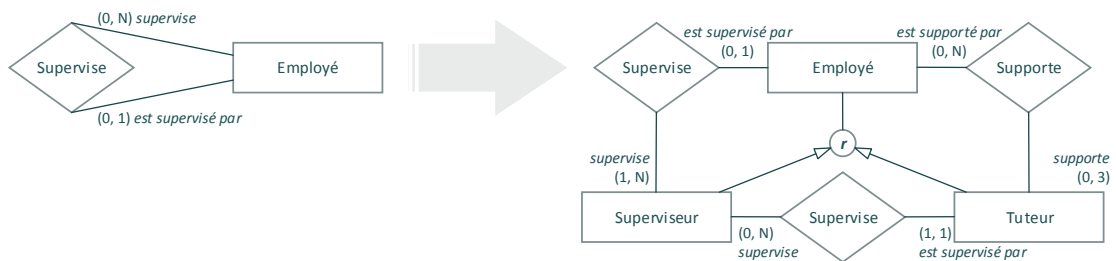
GPA775

18

## Modèle EAE Abstraction d'association

- ▶ Dans certaines situations, il est possible de créer un niveau d'héritage supplémentaire afin d'exprimer des associations présentes dans des super classes.
- ▶ Ce processus permet de préciser des rôles spécifiques.

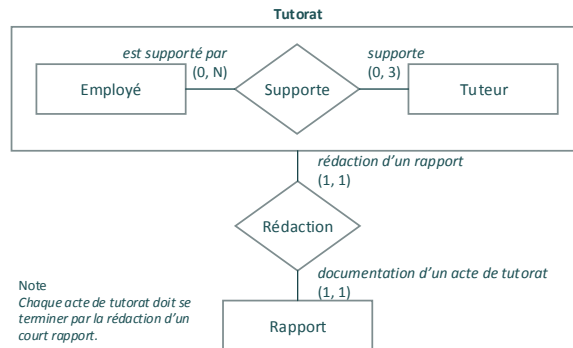
## Modèle EAE Abstraction d'association



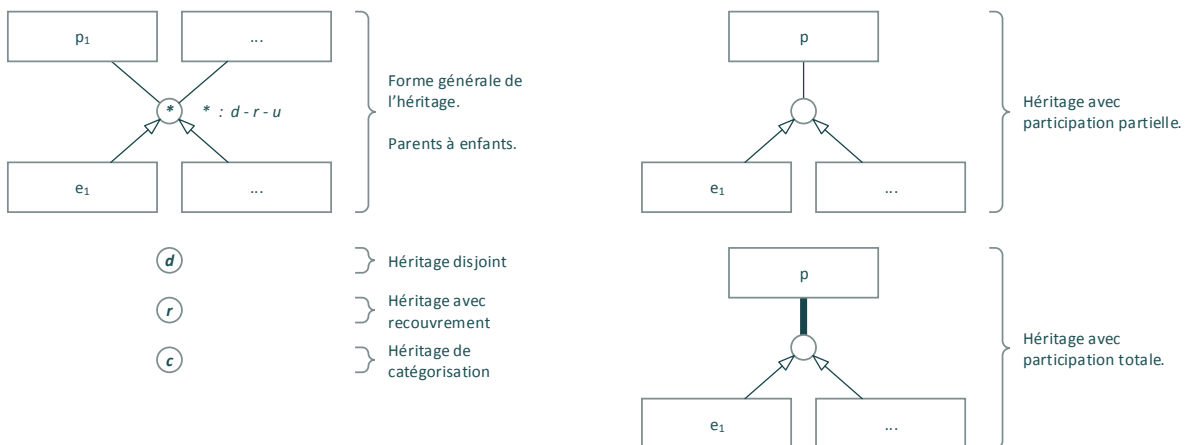
## Modèle EAE Agrégation

- ▶ L'agrégation est un concept d'abstraction consistant à créer une entité « virtuelle » à partir d'entités associées entre elles.
- ▶ Les agrégations peuvent être associées à d'autres entités par des associations ou un héritage quelconque.
- ▶ Souvent utilisé pour unir une association à une autre.

## Modèle EAE Agrégation



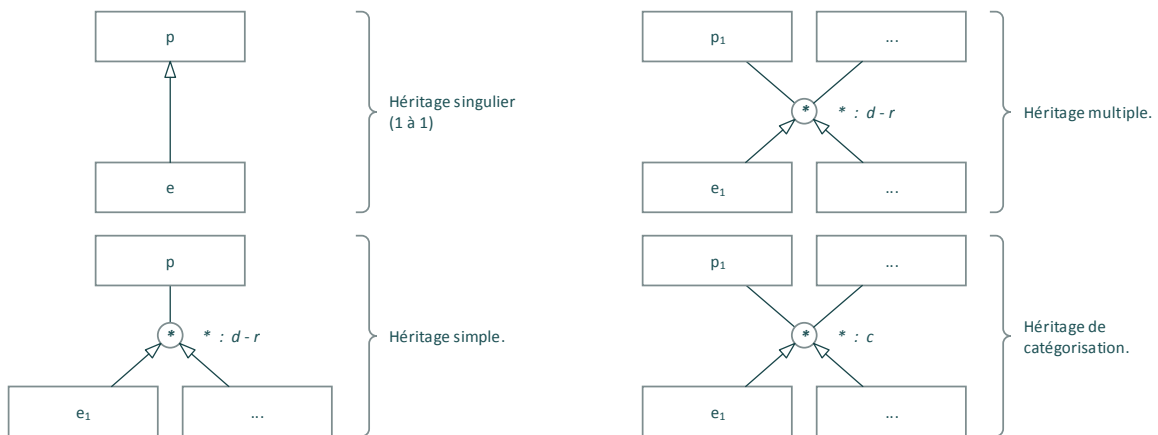
## Modèle EAE Représentation



GPA775

23

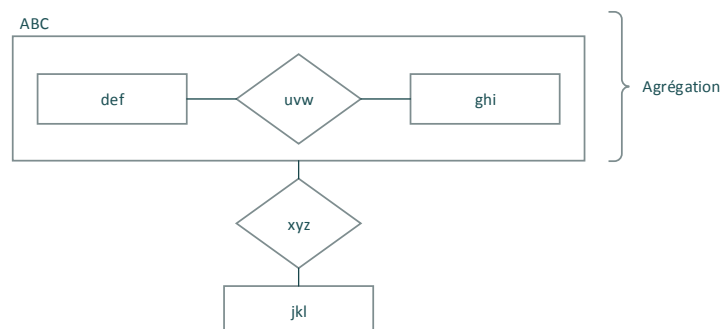
## Modèle EAE Représentation



GPA775

24

## Modèle EAE Représentation



## Modèle EAE

### Un exemple

## Modélisation UML

- ▶ UML vient de l'acronyme anglophone « *Unified Modeling Language* ».
- ▶ Ce n'est pas une méthode de conception mais plutôt un langage de modélisation.
- ▶ UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter une architecture logicielle.
- ▶ UML compte 14 diagrammes différents, complémentaires et dépendants hiérarchiquement. Tous les diagrammes se complètent afin de décrire tous les aspects du projet et pour toutes les étapes de son cycle de vie.

## Modélisation UML

- ▶ Il existe des bases de données orientées objets (BDOO) qui prennent en compte directement les notions d'objets.
- ▶ Tel que mentionné, le cadre du cours n'adresse que les bases de données relationnelles (BDR).
- ▶ Toutefois, le diagramme de classe reste pertinent et permet une conception rapprochée mais différentes du modèle EAE. Plusieurs concepts de la modélisation orientée objets peuvent être implémentés dans une BDR.

## Modélisation UML Diagramme de classe

- ▶ Le diagramme de classe est principalement constitué des éléments suivants :
  - ▶ la classe (et ses constituants);
  - ▶ les relations :
    - ▶ association;
    - ▶ agrégation;
    - ▶ compositions;
  - ▶ la généralisation.
- ▶ Plusieurs autres concepts existent mais ne sont pas couverts ici.

## Modélisation UML

### Diagramme de classe | Classe

- ▶ Une classe est la base d'UML.
- ▶ Elle est une représentation d'un ensemble d'objets concrets ou abstraits de même nature. Elle est le pendant de l'entité dans le modèle EA.
- ▶ Elle est constituée d'attributs mais aussi de méthodes.
- ▶ Elle est illustré par un rectangle divisé verticalement en 3 parties (nom, attributs et méthodes).
- ▶ Parfois, dans le contexte des BDR, on lui ajoute des contraintes, illustrées par une 4<sup>e</sup> sections (attention, ce n'est pas formellement du UML).
- ▶ Les attributs sont atomiques et le concept de clé primaire n'existe pas.

## Modélisation UML

### Diagramme de classe | Classe

Etudiant
codePermanent
noNAS
nom
prenom
genre
courriel
adresse
noTelephone
dateNaissance
quelAge
coteMoyenne
estInscrit
ageValide

Cours
sigle
nom
description
nbrCredit

## Modélisation UML

### Diagramme de classe | Relation

- ▶ Une relation représente un lien entre deux classes.
- ▶ Elle peut posséder un nom et, pour chaque lieu de connexion, le nom du rôle ainsi que l'une des 4 cardinalités disponibles (0..1, 1, 0..\*, 1..\*)
- ▶ Le modèle permet des relations *n-aires*. Néanmoins, il est plutôt d'usage courant d'utiliser des relations binaires.
- ▶ Les relations peuvent être :
  - ▶ une association, une agrégation ou une composition;
  - ▶ réflexive.

## Modélisation UML

### Diagramme de classe | Association

- ▶ L'association indique un lien quelconque entre deux classes.
- ▶ Elle peut être :
  - ▶ unidirectionnelle : indique qu'une seule classe peut accéder à la seconde;
  - ▶ bidirectionnelle : chacune des classes peut accéder à l'autre.
- ▶ Représenté par une ligne droite.
- ▶ Cet exemple se lit ainsi : un local accueil de 0 à N cours alors qu'un cours se donne dans un seul local.





## Modélisation UML

### Diagramme de classe | Agrégation

- ▶ L'agrégation indique qu'une classe est constituée (possède) d'une ou de plusieurs instances d'une autre classe;
- ▶ Le cycle de vie de la classe possédée n'est pas lié à la classe propriétaire.
- ▶ Représenté par une ligne droite muni d'un losange vide du côté propriétaire.
- ▶ Cet exemple se lit ainsi : Un département possède 1 ou plusieurs professeurs alors qu'un professeur est membre d'aucun à 3 départements.



## Modélisation UML

### Diagramme de classe | Composition

- ▶ La composition, semblable à l'agrégation, indique qu'une classe est constituée (possède) d'une ou de plusieurs instances d'une autre classe;
- ▶ Le cycle de vie de la classe possédée est lié à la classe propriétaire. Ainsi, si la classe propriétaire est détruite, les classes composées sont détruites aussi.
- ▶ Représenté par une ligne droite muni d'un losange noir du côté propriétaire.
- ▶ Cet exemple se lit ainsi : une université possède de 1 à plusieurs départements alors qu'un département n'appartient qu'à une université.



## Modélisation UML

### Diagramme de classe | Relation réflexive

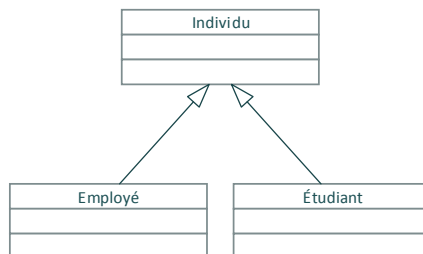
- ▶ Une relation réflexive est lorsqu'une classe est reliée à elle même.
- ▶ L'exemple de gauche se lit ainsi : un employé peut superviser de 0 à N employé alors qu'un employé est supervisé par aucun ou un seul employé.
- ▶ L'exemple de droite se lit ainsi : un cours possède de 0 à N cours préalable alors qu'un cours peut être préalable à aucun ou N cours.



## Modélisation UML

### Diagramme de classe | Généralisation

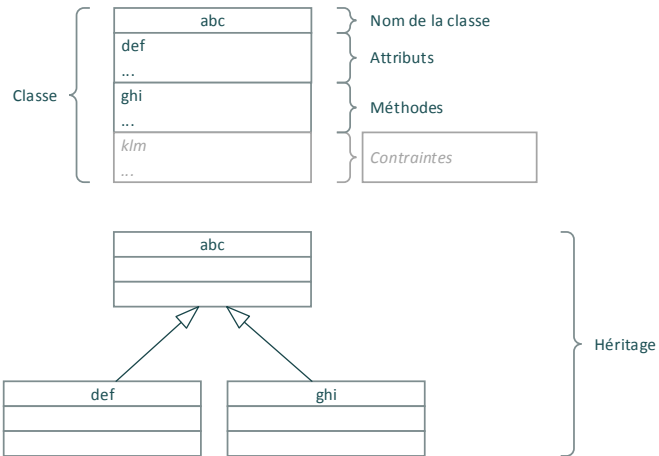
- ▶ La généralisation est encore le concept de l'héritage.
- ▶ Elle est représentée par une ligne munie d'une flèche vide pointant sur le parent.



GPA775

37

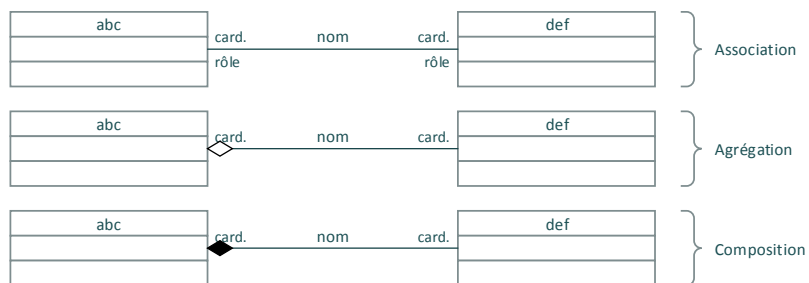
## Modélisation UML Représentation



GPA775

38

## Modélisation UML Représentation



# Modélisation UML

## Un exemple