



Université du Québec

École de technologie supérieure

Conception de bases de données multimédias

Plan du cours

- **Difficultés associées à la représentation d'objets MM (6.1)**
- Modélisation conceptuelle (UML-ERD) (6.2)
- Transformation du modèle conceptuel en un modèle logique relationnel (6.2.1)



Difficultés en modélisation multimédia

- Temps
- Durée
- Synchronisation



Difficultés de la représentation MM

- Nombre et variété des dimensions caractérisant un objet MM
 - dimension spatiale
 - dimension temporelle
 - dimension hiérarchique
 - dimensions du contenu
 - dimensions des objets (régions)
 - dimensions des relations entre ces objets
 - dimension technique : métadonnées acquises automatiquement lors de la capture de l'objet MM
 - une vidéo incorpore toutes ces dimensions



Difficultés de la représentation MM

- Dimension temporelle

- ordre temporel : l'ordre entre les images d'une vidéo est essentiel
- gestion des intervalles de temps
 - pouvoir comparer des séquences entre elles : se chevauchent-elles, l'une est-elle incluse dans l'autre, l'une se produit-elle après l'autre ?
- gestion de la notion de durée
- synchronisation (vidéo) : bande sonore et images



Difficultés de la représentation MM

- Dimension spatiale

- gestion d'un système de coordonnées à 2 ou 3 dimensions
- gestion de volumes, surfaces, lignes et points
- nécessite des opérateurs de comparaison particuliers (comparaison d'objets selon leur position)
 - points : l'un est-il à droite/à gauche/au-dessus/au-dessous de l'autre ?
 - surfaces : l'une est-elle contenue dans l'autre, chevauche-t-elle l'autre;
 - ...



Difficultés de la représentation MM

- Dimension hiérarchique

- vidéo = séquence de clips
 - clip = séquence de scènes
 - scène = séquence de prises de vue
- livre = séquence de chapitres
 - chapitre = séquence de sections
 - section = séquence de paragraphes
 - paragraphe = séquence de phrases
 - phrase = séquence de termes (unités syntaxiques)

Difficultés de la représentation MM

- Incorporer des vidéos/bandes sonores/ images/textes à un modèle de données
 - BD classiques (info de gestion)
 - le problème se pose également lors de l'incorporation de données préexistantes à un système
 - mais ensuite, les données sont généralement acquises pour s'intégrer au système ... pour un certain temps
 - car les besoins d'affaires évoluent
 - un objet MM est créé pour lui-même, et non dans le but d'être incorporé à une BDMM; il n'y a donc pas de notion de besoins qui évoluent de la même manière



Plan du cours

- Difficultés associées à la représentation d'objets MM (6.1)
- **Modélisation conceptuelle (UML-ERD) (6.2)**
- Transformation du modèle conceptuel en un modèle logique relationnel (6.2.1)
- Définition de types de données utilisateur TDU (6.3)
- Manipulation de TDU (6.3.1)
- Introduction aux types Oracle InterMedia



Analyse : modèle conceptuel de données

- **Modèle conceptuel de données**
 - représentation graphique abstraite des informations à placer dans la BD
 - indépendante de la technologie utilisée pour l'implémentation
- **Sans modélisation conceptuelle c'est le chaos d'une conception hasardeuse !**



UML et Oracle Designer

- UML indépendant des implémentations
- Utilisé dans les standards internationaux comme MPEG
- Notation pour représenter l'analyse et la conception de systèmes informatiques dans leur ensemble
 - les données en sont l'une des composantes
 - issue du monde de l'OO (Orienté-Objet)
 - idéal pour conception en OO
 - mais permet de modéliser n'importe quel type de BD
- Oracle Designer
 - Notation particulière au fournisseur
 - Notation mixte de ERD et OO
 - Supporté par des outils de conception et génération automatisée
 - Très répandu dans le marché actuel



Conception OO et ERD

- L'industrie n'en est pas au BD-OO
- Les modèles commerciaux actuels n'appliquent pas la notion BD-OO
- Les difficultés avec les BD-OO ont considérablement ralenti le déploiement
- Actuellement c'est le relationnel-objet (RO) qui a le marché !



UML



Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

Conceptuel UML: notions de base

- Relationnel-Objet

- représentation d'une entité
 - du monde extérieur (un utilisateur)
 - ou de l'environnement informatique (un menu)
- un objet a
 - une identité, distincte et permanente, représenté par un OID unique à travers la BD
 - un état, représenté par des propriétés, changeantes avec le temps
 - des comportements (impliquant un changement d'état) et des opérations associées



Conceptuel UML: notions de base

- Type/classe : ensemble des attributs et opérations caractérisant le comportement d'un groupe d'objets
- Attribut/propriété
 - simple/composé : nom / adresse
 - mono/multivalué : nom / téléphones
 - de base/dérivé : date de naissance / âge
 - l'âge peut être déterminé par une opération impliquant la date de naissance et la date du jour



Notion d'objet et de classe

- **Objet (instance d'une classe)**
 - significatif pour le domaine d'application
 - caractérisé par
 - *identité*
 - *état*
 - *comportement*
- **Attribut (variable membre, variable d'instance)**
 - contenant pour une valeur
 - *représente son état*



Conceptuel UML: notions de base

- Un **objet** est représenté par un rectangle
 - La partie supérieure contient le nom de l'objet qui doit être souligné
 - Le centre est la liste des attributs, leurs types et leurs valeurs

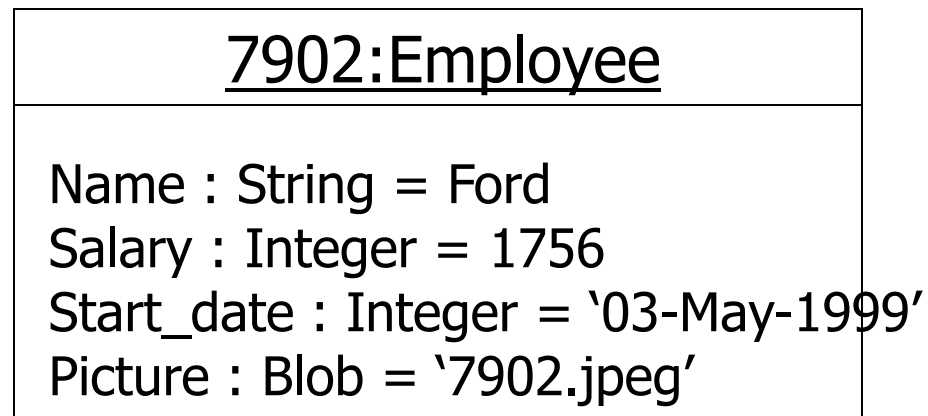
Conceptuel UML: notions de base

- Un objet a un état déterminé par les valeurs de ses attributs
- L'état peut être modifié par un programme en modifiant les valeurs de ses attributs



Représentation d'un objet en UML

Fig 6.4



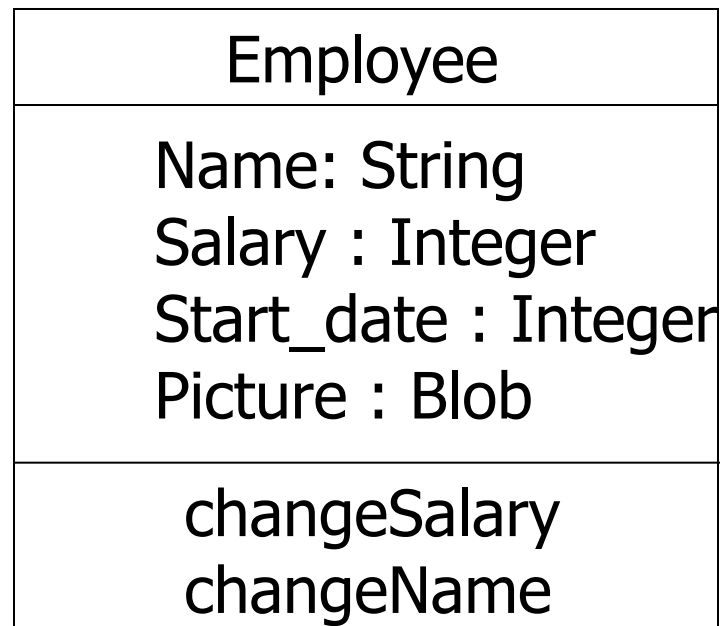
Classe UML

- Une classe est une abstraction qui représente les **caractéristiques communes** à un ensemble d'objets
 - Attributs
 - Associations
 - Opérations



Représentation d'une classe en UML

Fig 6.5



Exemple : classes et objets

CLASSE
attributs
opérations

Vidéo
titre date sujet durée
segmenter compresser

Objet : CLASSE
attributs

<u>11231 : Vidéo</u>
titre : Bugs Bunny sujet : Carrot Cake durée : 4min:20sec

Identifiant d'objet (OID, object identifier)

- Mécanisme d'identification
 - pas deux objets avec le même OID
- Implicite
 - non visible (n'est pas représenté explicitement dans le diagramme)
 - réalisation traitée à l'instanciation
- Mécanisme de référence



Pas besoin d'identificateur explicite !

- Par opposition au relationnel

OID =154396

<u>: Prêt</u>
datePrêt : date = 10/10/2000

OID = 204395

<u>: Prêt</u>
datePrêt : date = 10/10/2000



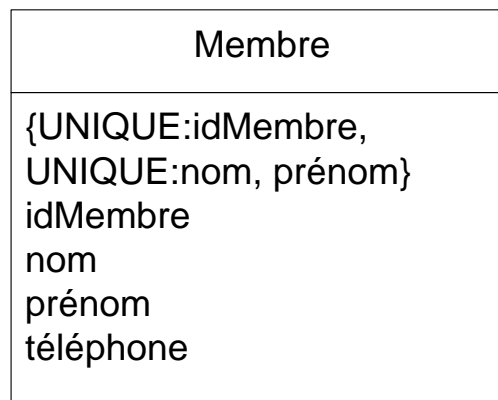
Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

***Identifiant naturel (ou clé «key»)* pour une classe**

- Ensemble minimal d'attributs qui identifie chacun des objets de manière unique {
▪ \sim clé candidate du relationnel
- Représentation par une contrainte UML (contrainte de l'attribut)



Ici deux objets membre ne peuvent pas avoir les mêmes valeurs pour l'attribut

Syntaxe générale pour la spécification des attributs en UML

- *[visibilité] nom [multiplicité] [: type] [= valeurInitiale] [{propriétés}]*
 - *visibilité* peut être :
 - + publique
 - # protégé
 - - privé
 - *nom* de l'attribut
 - *multiplicité* ([1..1] par défaut)
 - ✓ *téléphone*[1..2]: *String*
 - ✓ *adresse* [0..1]: *String*
 - ✓ *auteurs* [1..*]: *String*

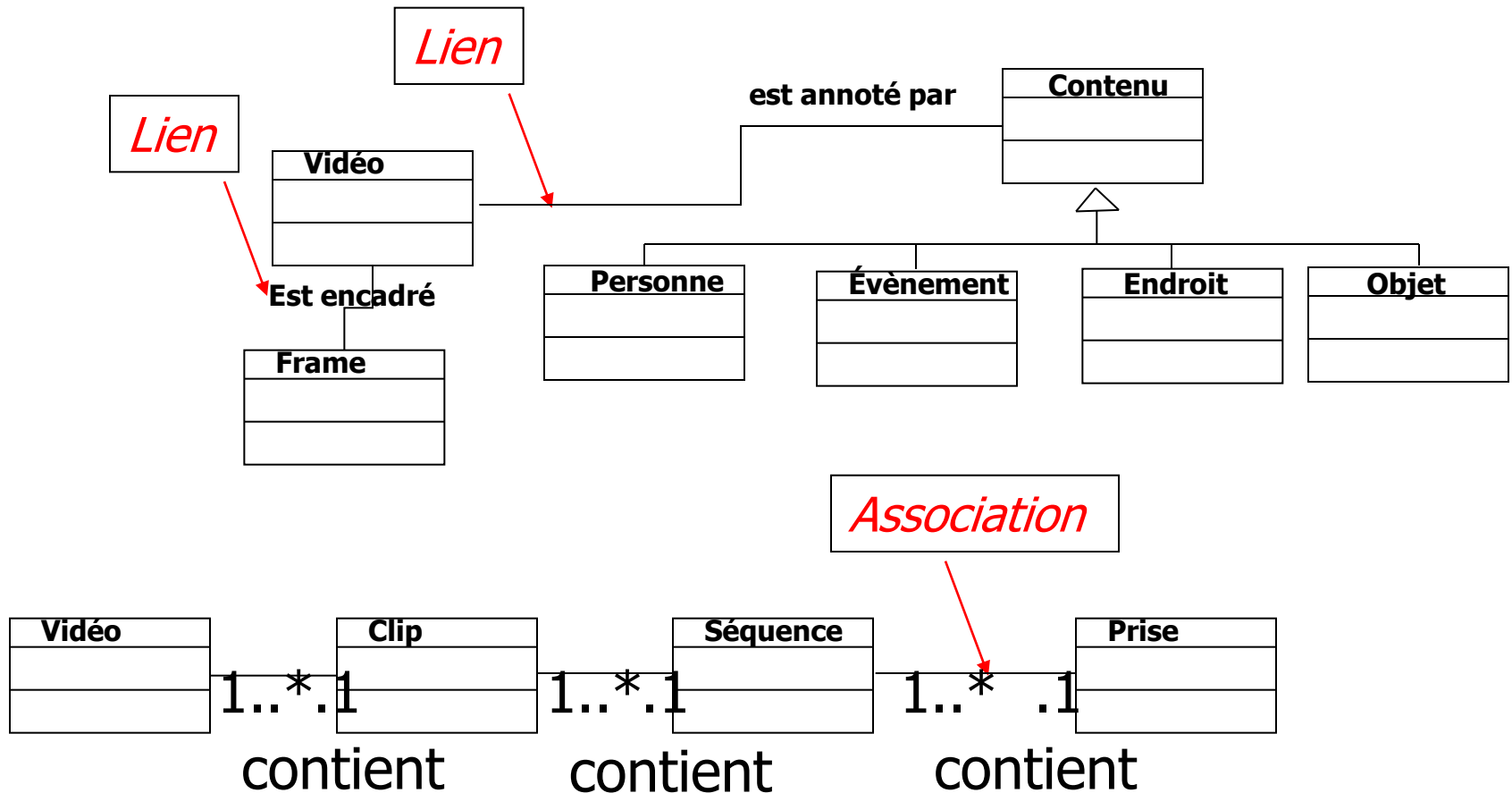


Syntaxe pour attributs (suite)

- [*visibilité*] *nom* [*multiplicité*] [*: type*] [*= valeurInitiale*]
[*{propriétés}*]
 - *type* :
 - *OCL (Object Constraint Language)*
 - *Boolean, Integer, Real, String, enum{valeur1,..., valeurN}*
 - types de la plate-forme visée
 - type non pré-défini
 - classe stéréotypée «*datatype*»

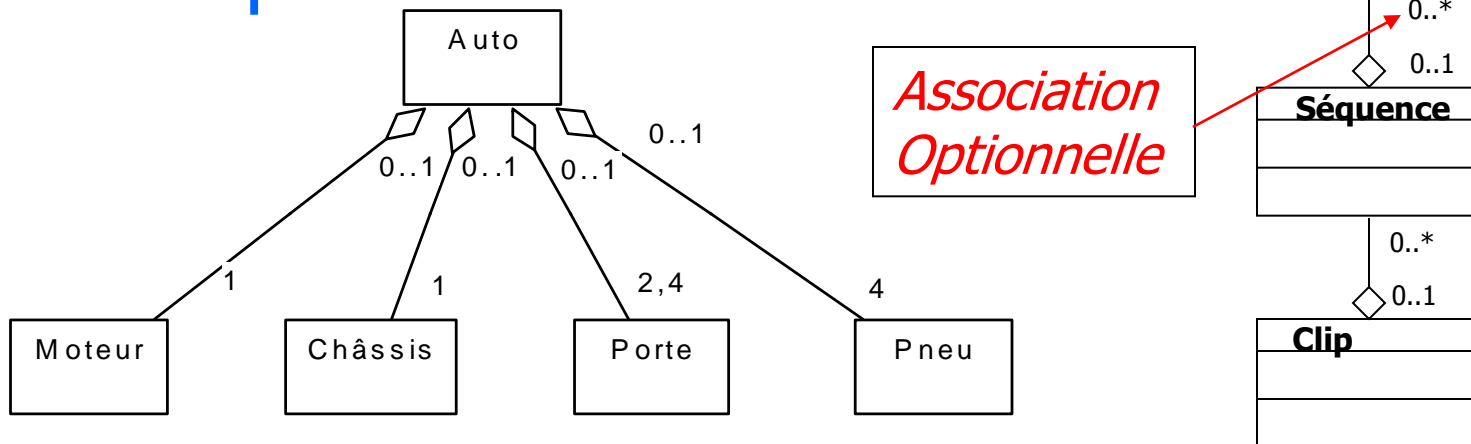


Notion de lien et d'association binaire



Agrégation

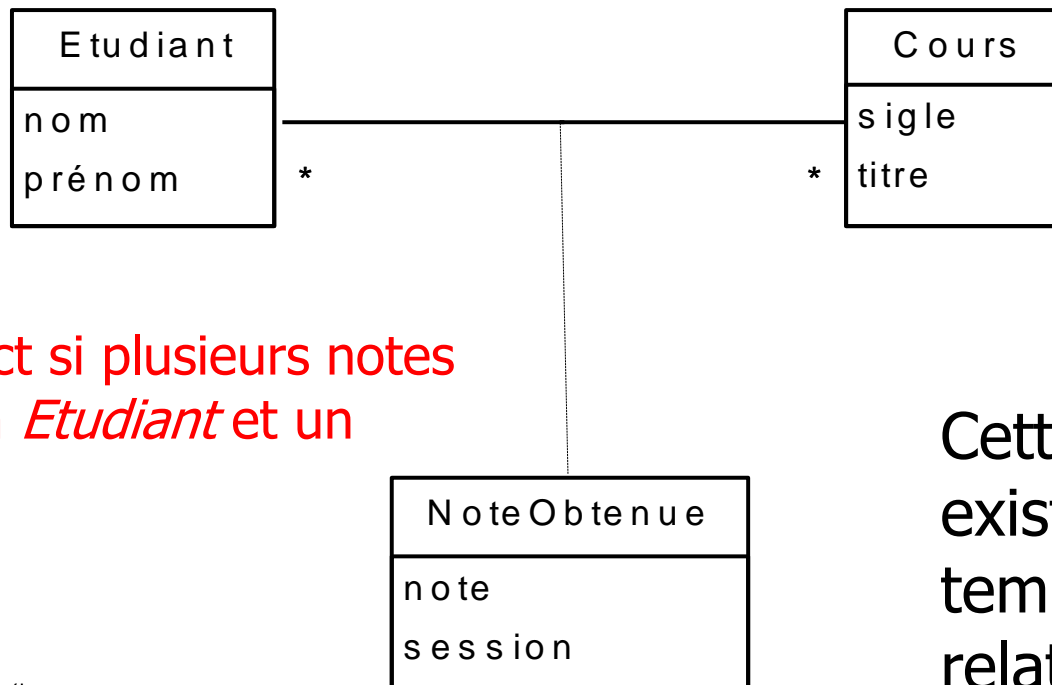
- Cas particulier d'association



Une association entre un tout et ses parties = agrégation

Classes associatives

- Association plusieurs à plusieurs (fig 6.7)

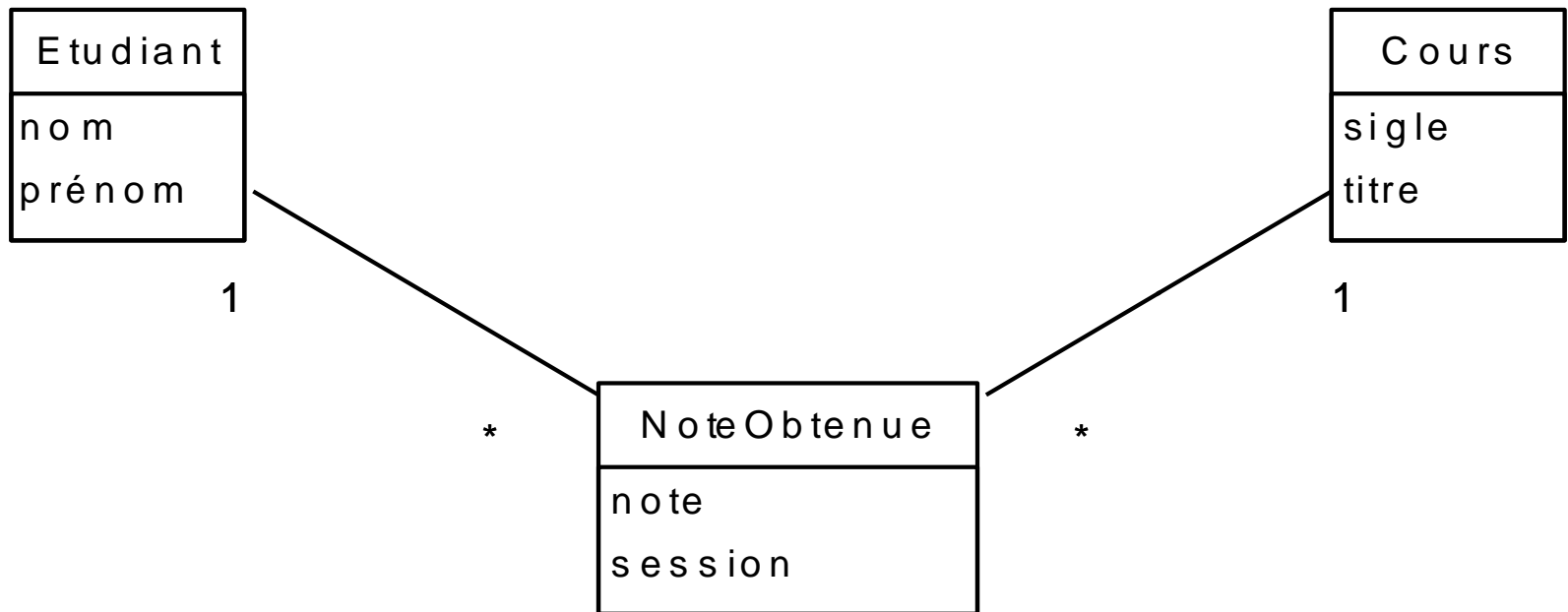


Incorrect si plusieurs notes
pour un *Etudiant* et un
Cours

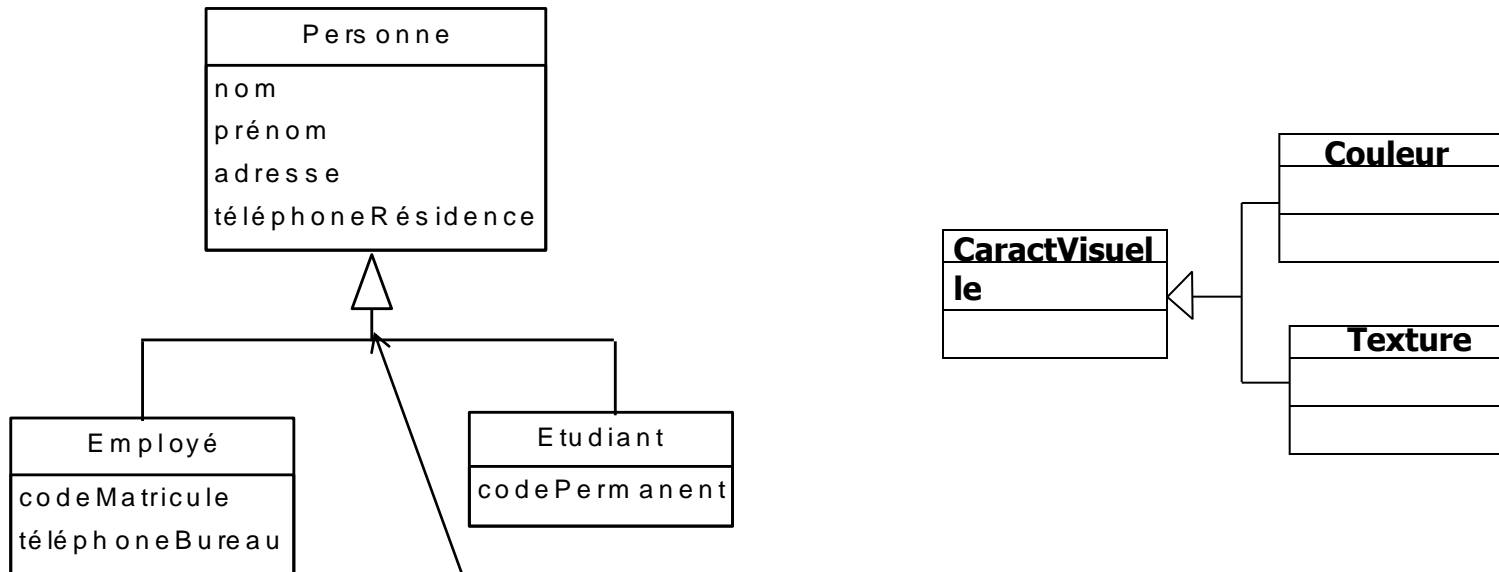
Cette situation
existe en tout
temps dans les
relations + à +.

Suite de l'exemple

- Si il y a plusieurs notes pour un *Étudiant* et un *Cours*



Lien d'héritage



Une relation de généralisation
est notée par un triangle

ERD



Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

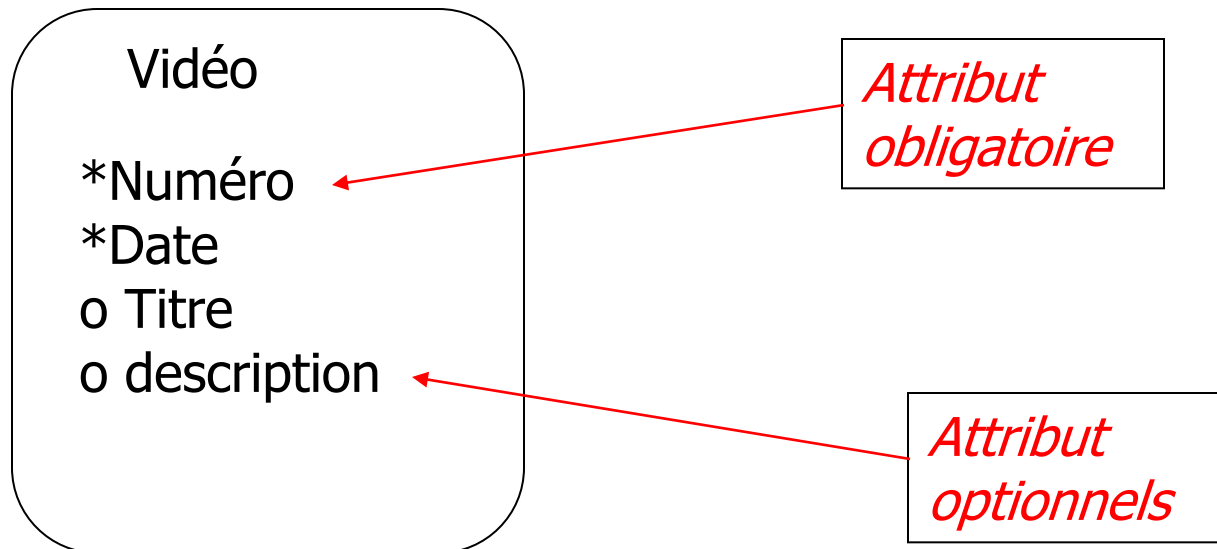
Conceptuel Entité-Associations: notions de base

- Entité

- Quelque chose de significatif pour l'entreprise pour lequel des données doivent exister
 - Même concept qu'un objet
 - Possède des instances
 - Possède des attributs
 - Possède des relations
 - Ne possède pas de OID; doit avoir un identifiant
 - Outils Designer qui génère la BD pour vous !

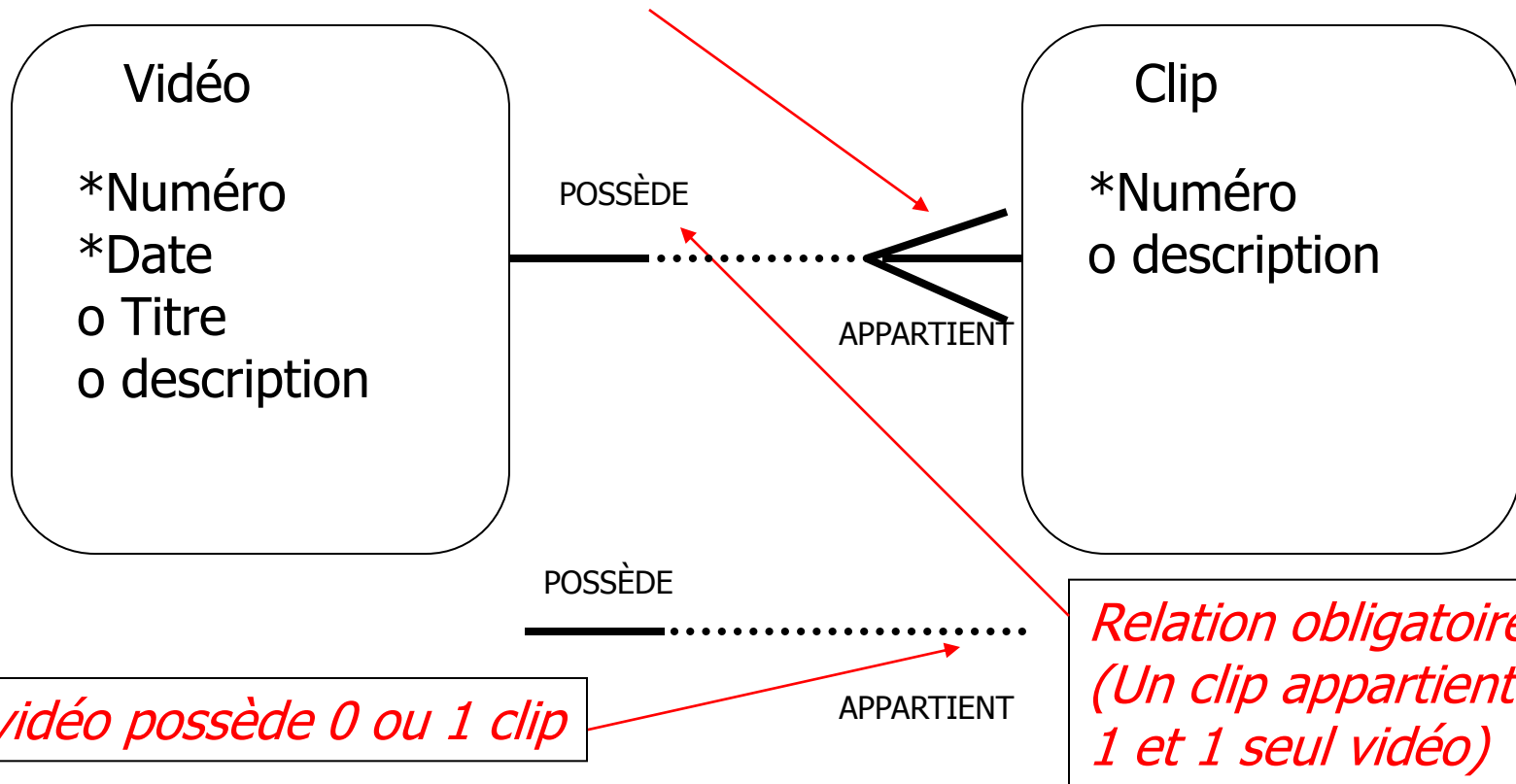


Représentation d'un entité avec Oracle Designer

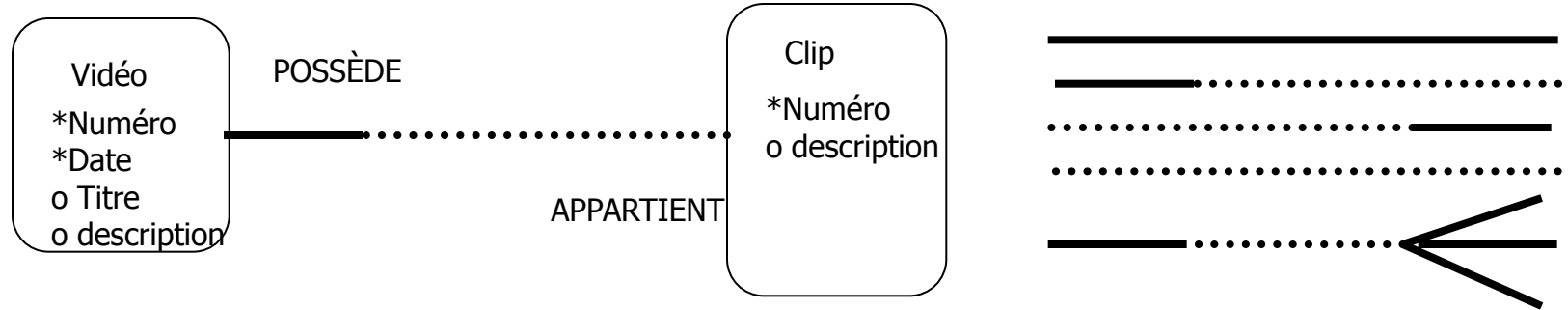


Représentation d'un relation avec Oracle (fig 6.1)

Un vidéo possède 1 ou plusieurs Clip(s)



Lire les relations



Un vidéo possède 0 ou 1 clip



Un clip appartient à 1 et 1 seul vidéo



Université du Québec

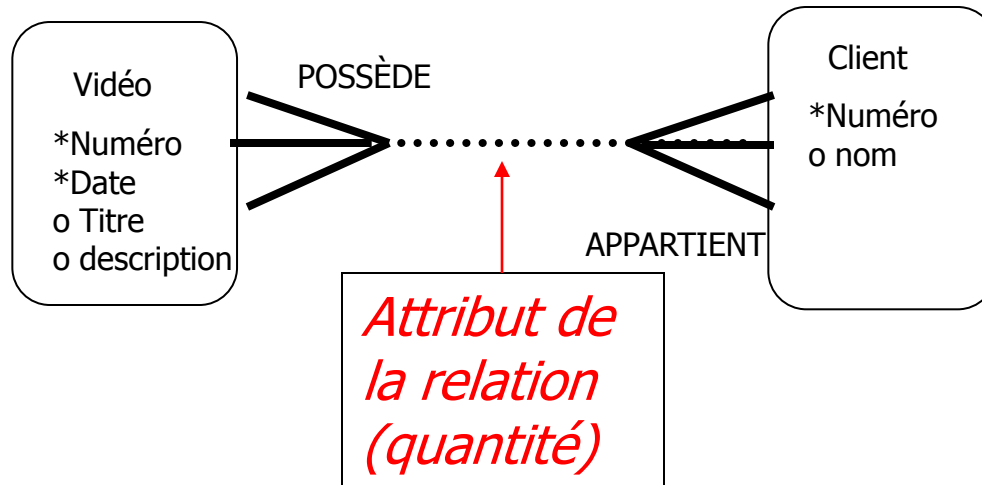
École de technologie supérieure

Département de génie logiciel et des TI

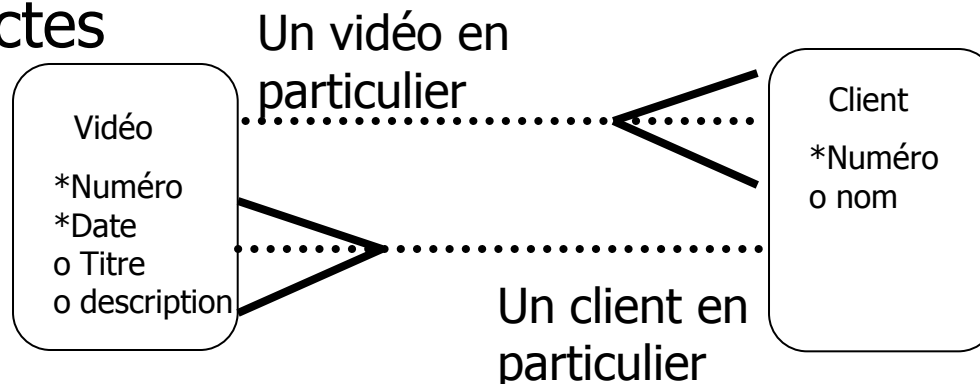
D'autres exemples de relations ERD

	[min..max]	[min..max]
	[1..1]	[1..1]
	[1..1]	[0..1]
	[0..1]	[1..1]
	[0..1]	[0..1]
	[1..1]	[0..*]
	[0..1]	[0..*]
	[0..1]	[1..*]
	[1..*]	[1..*]

Plusieurs à Plusieurs



En réalité, il s'agit de deux relations distinctes



Plan du cours

- Difficultés associées à la représentation d'objets MM (6.1)
- Modélisation Conceptuelle (UML-ERD) (6.2)
- **Transformation du modèle conceptuel en un modèle logique relationnel (6.2.1)**
- Définition de types de données utilisateur TDU (6.3)
- Manipulation de TDU (6.3.1)
- Introduction aux types Oracle InterMedia



Le modèle logique

- Transformation "mécanique" du schéma conceptuel en un schéma logique
- Ici schéma logique relationnel Oracle
- Assure la compatibilité de l'analyse conceptuelle avec le modèle physique de BD retenu
- En relationnel, colonnes uniquement simples et monovaluées



Le schéma logique : règles de normalisation

- Transformation des classes en tables
- Transformation des associations N:N (création d'entité d'intersection)
- Transformation des associations 1:N (ajout clé étrangère côté N)
- Transformation des hiérarchies

Le schéma logique : règles

- Transformation des classes en tables
 - Choix des clés primaires : invariantes, sans structure interne ni signification externe, de petite taille

Projets
<u>Id_projet</u>
Nom_projet
Statut

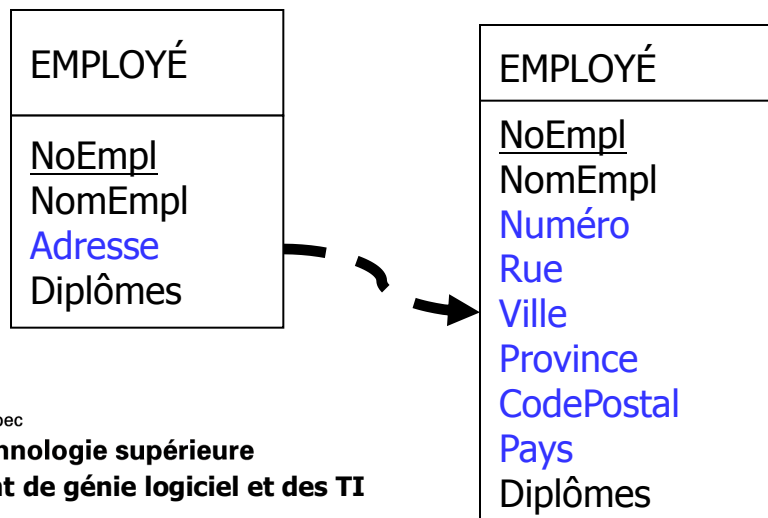
Périodes
<u>Id_période</u>
Date_début
Date_fin
Statut

Employés
<u>Id_employe</u>
No_employe
Matricule
Nom_employe
Prénom
Code_usager
Taux_horaire_base



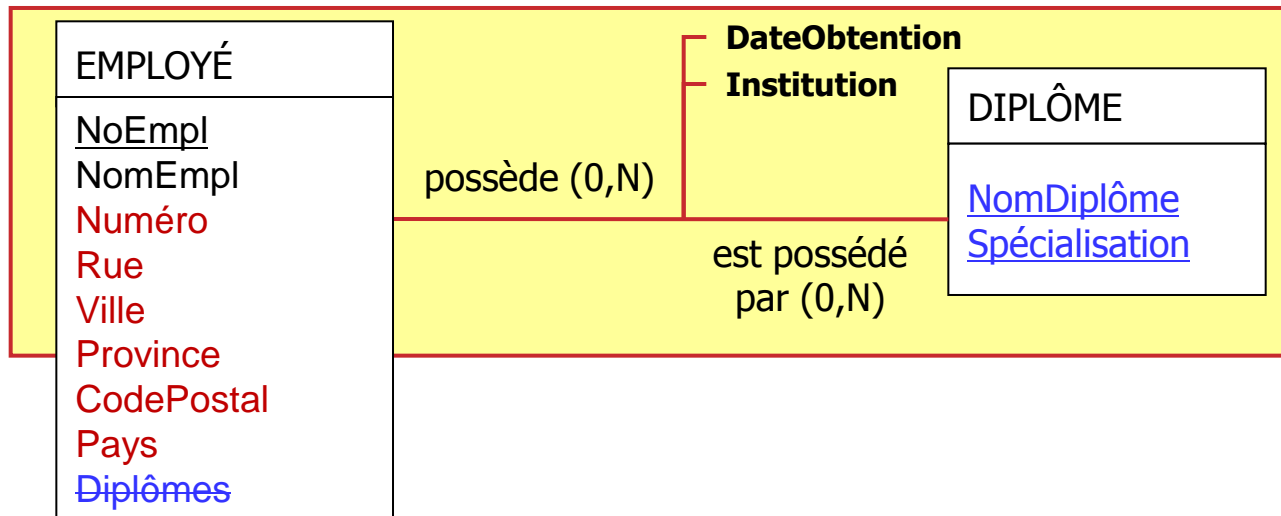
Le schéma logique : règles

- Transformation des classes en tables (suite)
 - Choix des clés primaires : invariantes, sans structure interne ni signification externe, de petite taille
 - Transformation des attributs structurés monovalués en plusieurs champs :
 - adresse -> rue, ville, province, code_postal



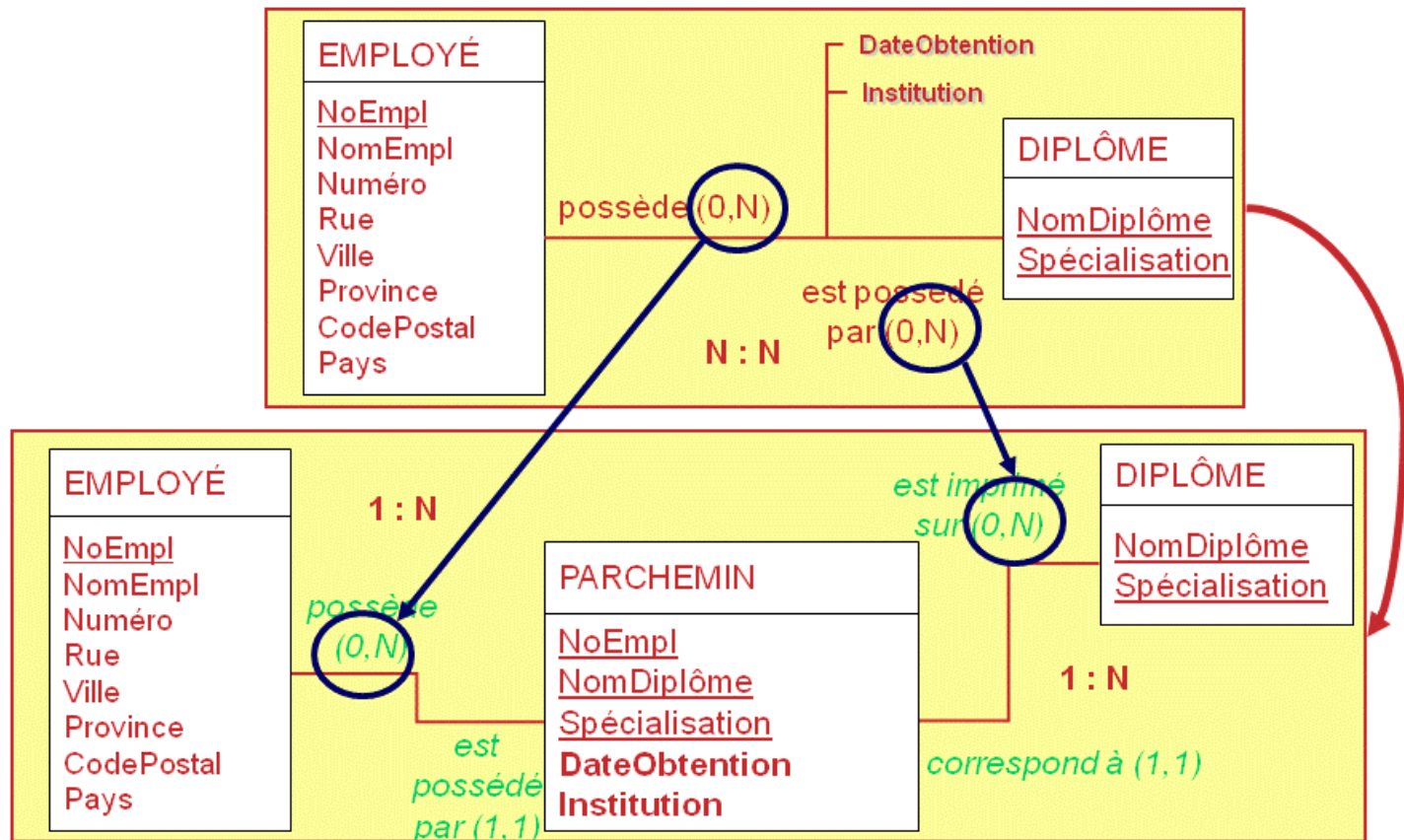
Le schéma logique : règles

- Transformation des classes en tables (suite)
 - Transformation des attributs multivalués en tables auxiliaires avec ajout d'une clé étrangère vers la table principale : table des téléphones



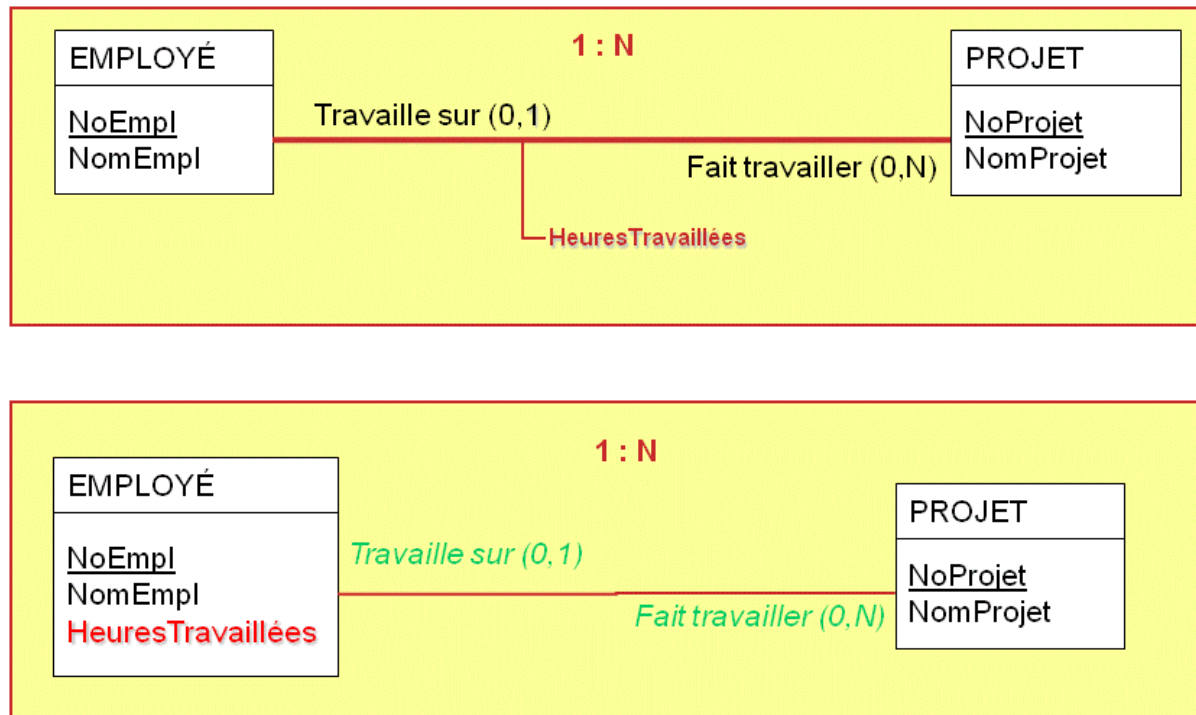
Le schéma logique : règles

- Transformation des associations N:N



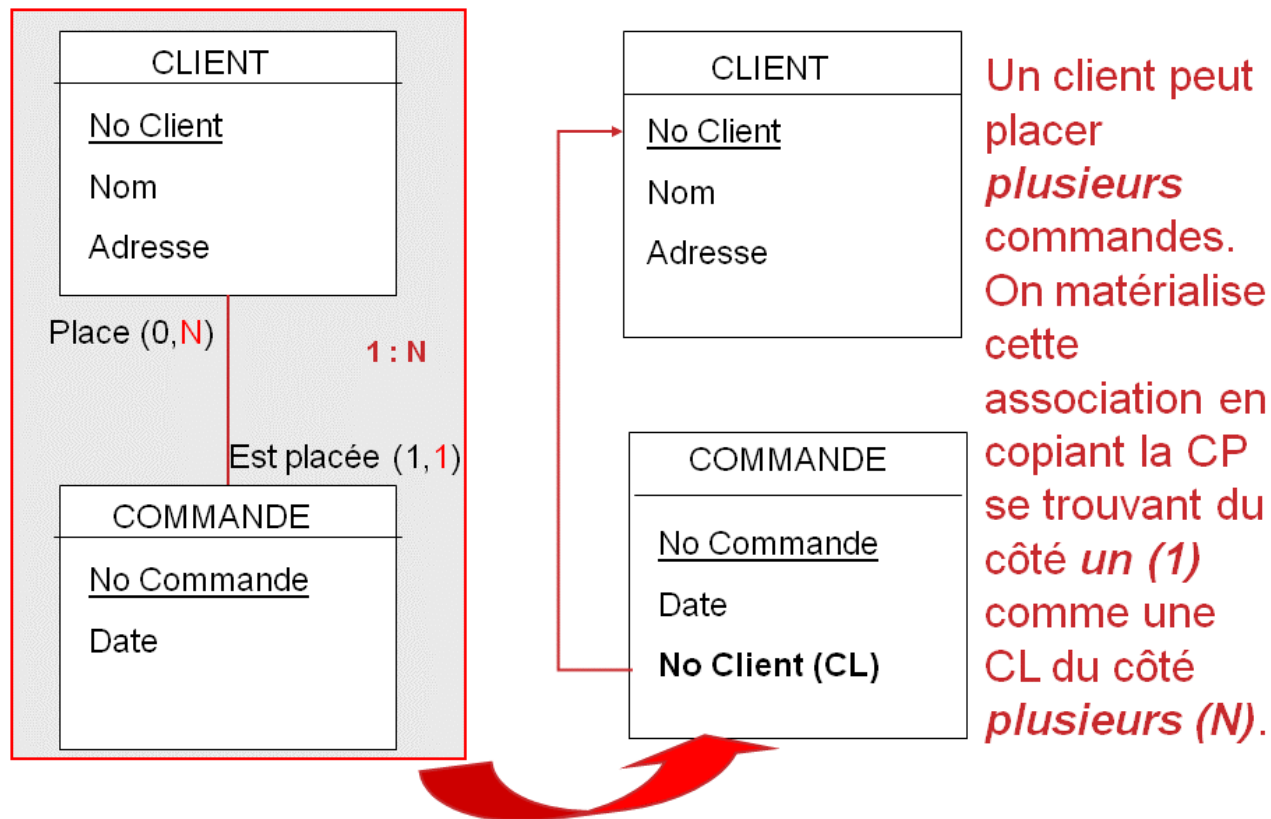
Le schéma logique : règles

- Éliminer les attributs d'association



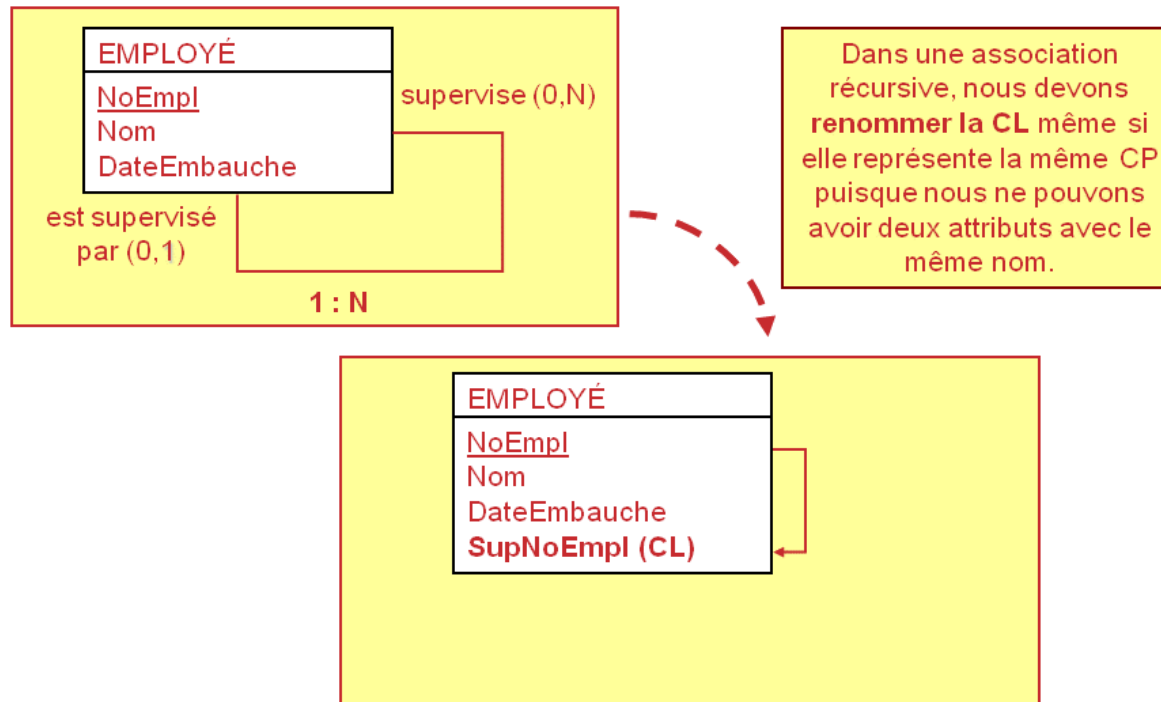
Le schéma logique : règles

- Transformation des associations 1:N



Le schéma logique : règles

- Transformation des associations 1:N



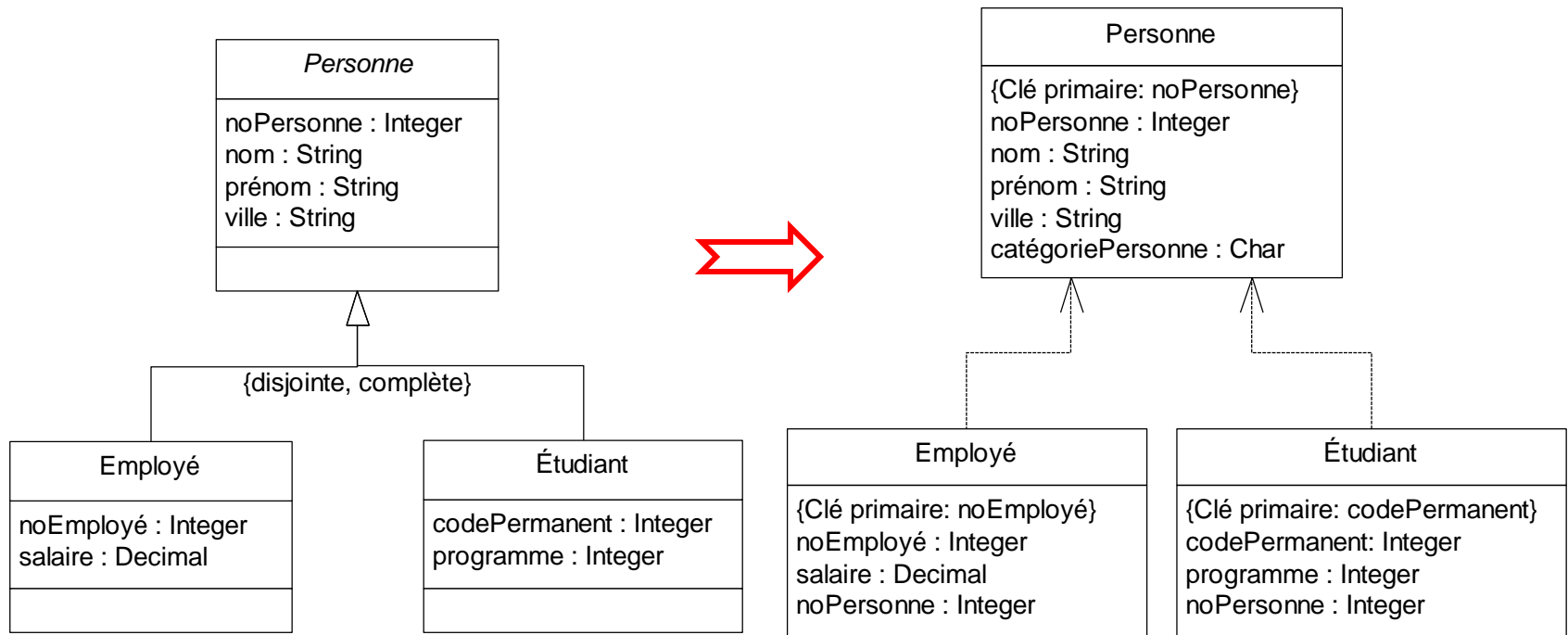
Le schéma logique : règles

Traduction des hiérarchies : trois approches

1. Délégation : une table par classe
2. Fusion : fusion du parent et des enfants; une seule grande table
3. Concaténation : duplication des colonnes du parent dans chaque enfant; la table parent est éliminée

Le schéma logique : règles

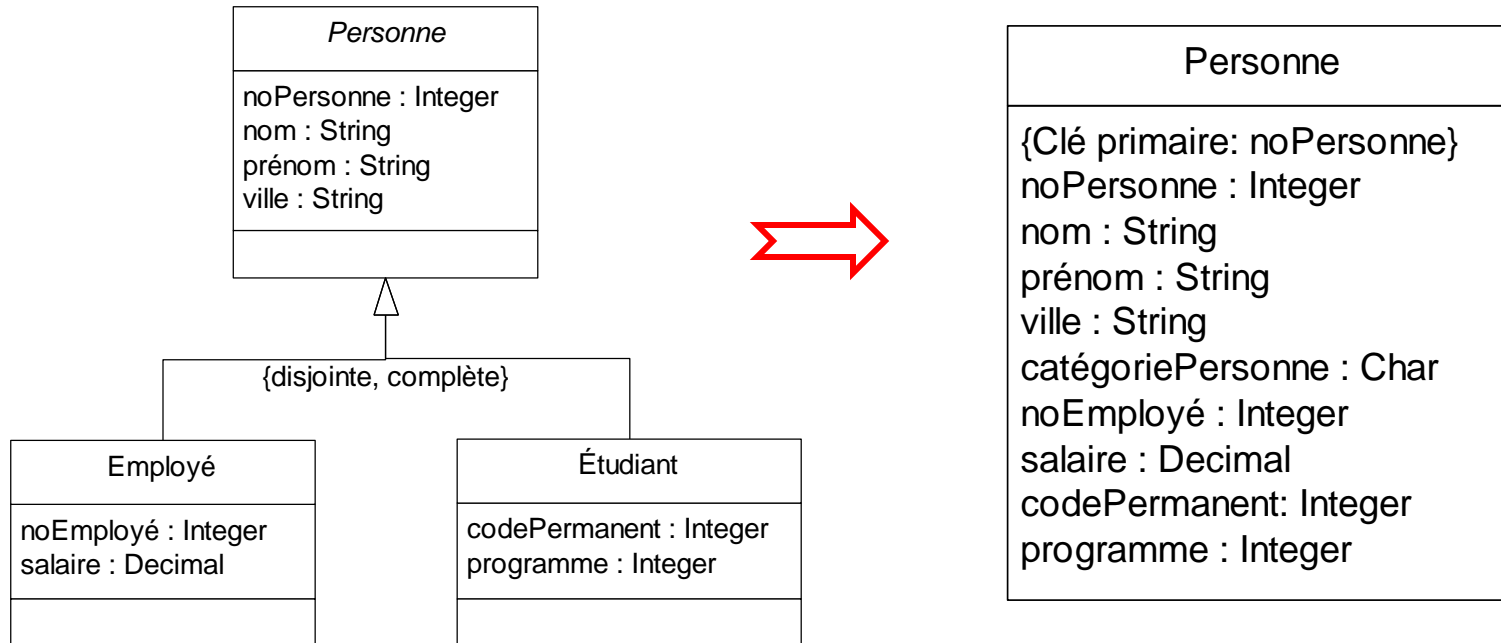
1. Délégation : une table par classe



- Attribut discriminant optionnel
- Jointures obligatoires

Le schéma logique : règles

2. Fusion : une seule grande table



- **Attribut discriminant obligatoire**

Le schéma logique : règles

3. Concaténation : une table par enfant

