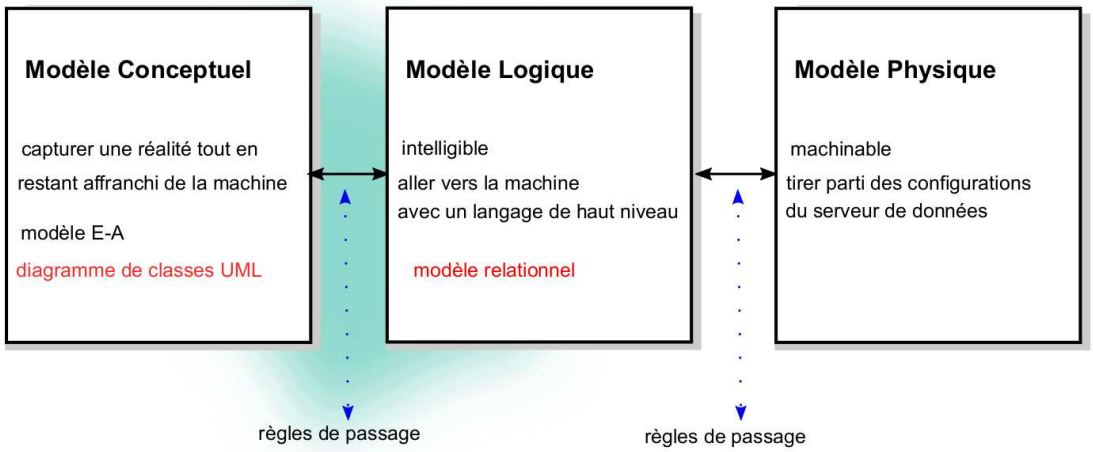
HMIN112M : Règles de passage UML vers le relationnel

I.Mougenot

UM

2020

Du diagramme de classes au schéma relationnel



Règles de passage

Trois règles principales

1. Toute classe devient une relation
2. Toute association devient une relation
3. Si l'association a une multiplicité de 1 ou de 0..1 à une de ses extrémités, il est possible de ne pas créer de relation, mais de garder l'information sous la forme de migration d'attributs

Règle 1

Toute classe devient une relation

Deux cas se présentent

1. un attribut ou un ensemble d'attributs assurent naturellement l'identification de chaque individu
2. aucun attribut (ou ensemble d'attributs) ne se dégage comme identifiant naturel

Exemple avec identifiant naturel

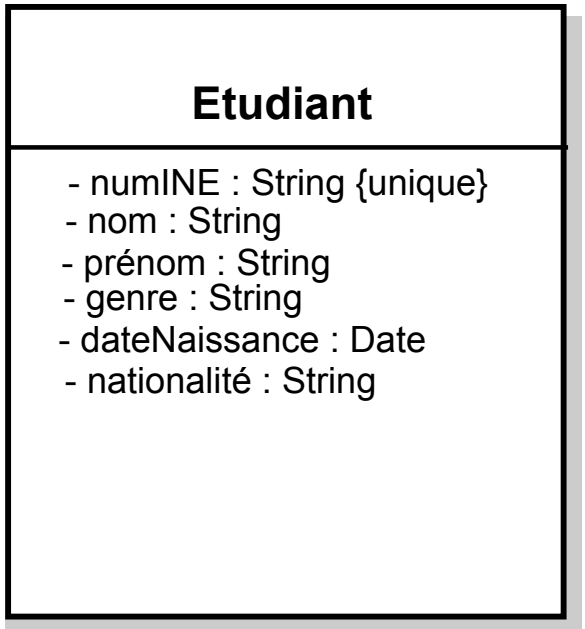


Figure: numINE va jouer le rôle de clé primaire de la relation

Définition de la relation Etudiant

Schéma relationnel

Etudiant(<u>numINE varchar(12)</u>, nom varchar(20), prenom varchar(20), genre char(1), dateNaissance Date, nationalite varchar(15))

Exemple sans identifiant naturel

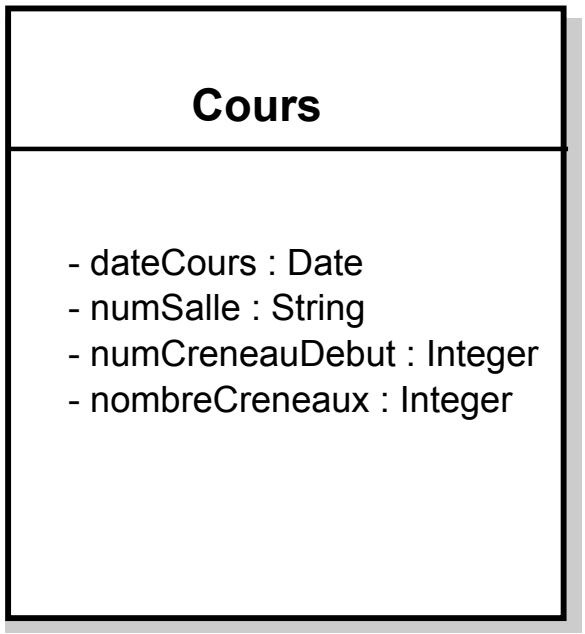


Figure: soit un identifiant artificiel, soit tous les attributs formant la clé primaire

Définition de la relation Cours

Schémas relationnels

– choix 1

Cours(dateCours Date, numSalle varchar(6), numCreneauDebut integer, nombreCreneaux integer)

– choix 2

Cours(numCours integer, dateCours Date, numSalle varchar(6), numCreneauDebut integer, nombreCreneaux integer)

Dériver une association

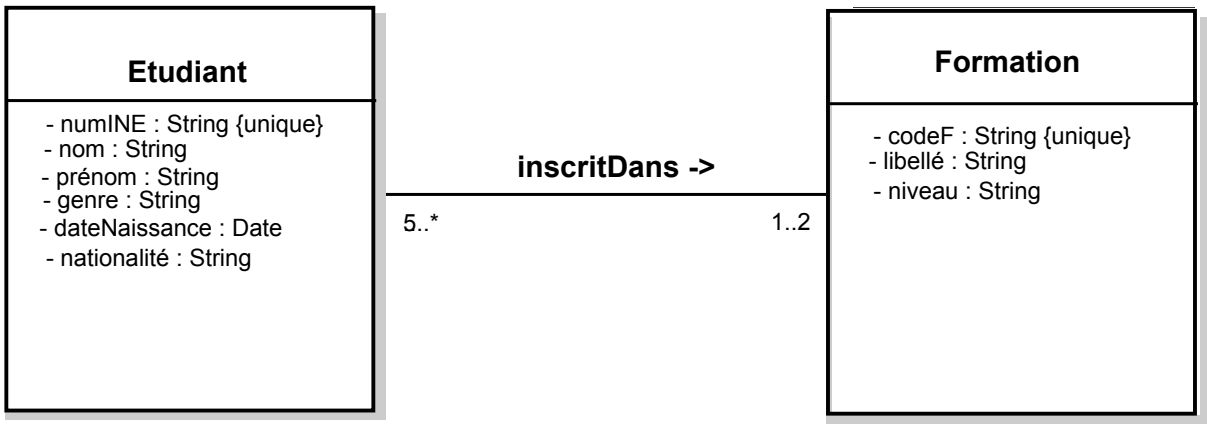


Figure: Les multiplicités ne vont pas permettre d'optimiser

Définition de la relation InscritDans

Schémas relationnels Etudiant et Formation

Etudiant(numINE varchar(12), nom varchar(20), prenom varchar(20), genre varchar(1), dateNaissance Date, nationalite varchar(15))
Formation(codeF varchar(8), libelle varchar(20), niveau varchar(4))

Schéma relationnel InscritDans

Une contrainte de clé primaire sur le couple numINE et codeF et deux contraintes de clé étrangère
InscritDans(numINE varchar(12), codeF varchar(8))
avec InscritDans(codeF) $\subseteq$ Formation(codeF)
et avec InscritDans(numINE) $\subseteq$ Etudiant(numINE)

Optimiser et ne pas créer la relation sur l'association

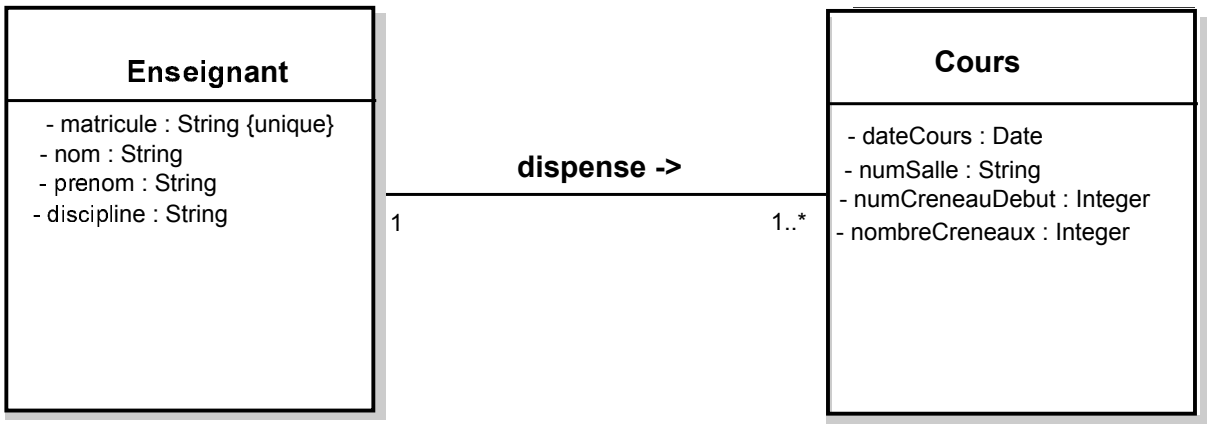


Figure: Faire jouer l'optimisation

Migration de l'attribut matricule dans Cours

Enseignant et Cours

Enseignant(matricule varchar(12), nom varchar(20), prenom varchar(20), discipline varchar(15))
– une contrainte de clé étrangère, l'enseignant qui dispense le cours est avant tout un enseignant
Cours(numCours integer, dateCours Date, numSalle varchar(6), numCreneauDebut integer, nombreCreneaux integer, matriculeEnseignant varchar(12))
avec Cours(matriculeEnseignant) $\subseteq$ Enseignant(matricule)

Exemple classe association

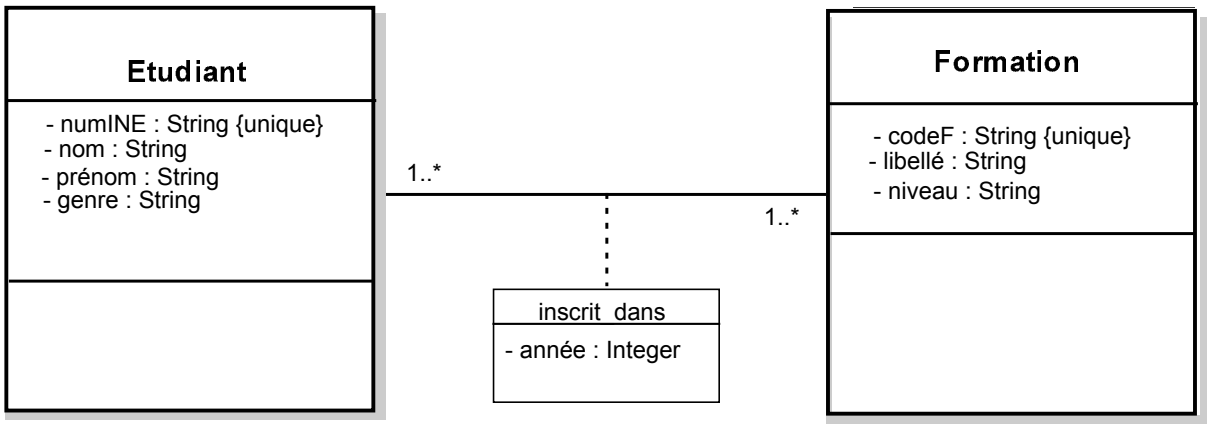


Figure: Attributs identifiants des classes reliées et attributs propres

Classe association InscritDans

Schémas relationnels Etudiant et Formation

Etudiant(numINE varchar(12), nom varchar(20), prenom varchar(20), genre varchar(1), dateNaissance Date, nationalite varchar(15))
Formation(codeF varchar(8), libelle varchar(20), niveau varchar(4))

Schéma relationnel InscritDans

Une contrainte de clé primaire sur le triplet (numINE, codeF, annee) et deux contraintes de clé étrangère
InscritDans(numINE varchar(12), codeF varchar(8), annee integer)
avec InscritDans(numINE) $\subseteq$ Etudiant(numINE)
et avec InscritDans(codeF) $\subset$ Formation(codeF)

Association réflexive

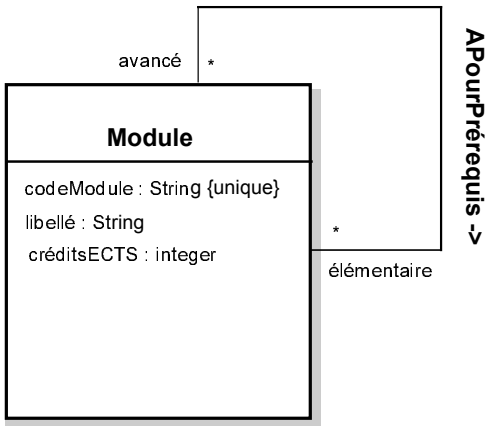


Figure: Utiliser les rôles en matière de sémantique

Association réflexive

Schéma relationnel Module

Module(codeModule varchar(8), libelle varchar(20), creditsECTS integer)

Schéma relationnel APourPrerequis

Une contrainte de clé primaire sur le couple (codeElementaire, codeAvance) et deux contraintes de clé étrangère
APourPrerequis(codeAvance varchar(8), codeElementaire varchar(8))
avec APourPrerequis(codeAvance) $\subseteq$ Module(codeModule)
et avec APourPrerequis(codeElementaire) $\subseteq$ Module(codeModule)

Association d'agrégation

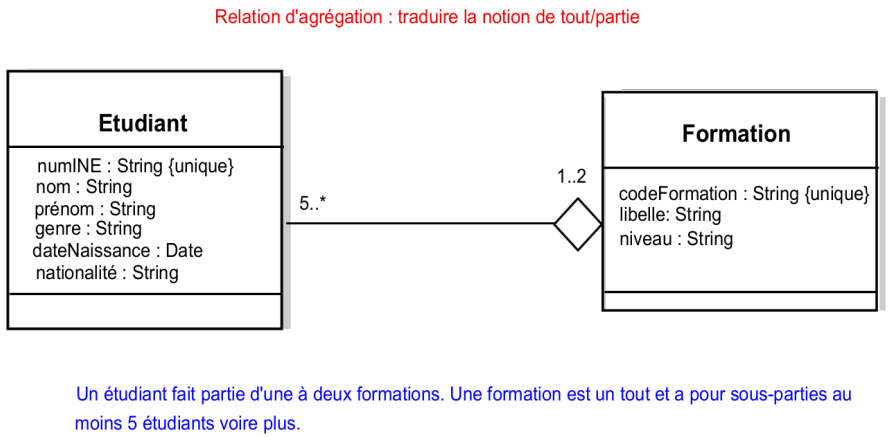


Figure: Nommer la relation d'agrégation : contient

Définition de la relation Contient

Schémas relationnels Etudiant et Formation

Etudiant(numINE varchar(12), nom varchar(20), prenom varchar(20), genre varchar(1), dateNaissance Date, nationalite varchar(15))
Formation(codeF varchar(8), libelle varchar(20), niveau varchar(4))

Schéma relationnel Contient

Une contrainte de clé primaire sur le couple numINE et codeF et deux contraintes de clé étrangère
Contient(codeF varchar(8), numINE varchar(12))
avec Contient(codeF) $\subseteq$ Formation(codeF)
et avec Contient(numINE) $\subseteq$ Etudiant(numINE)

Association de composition

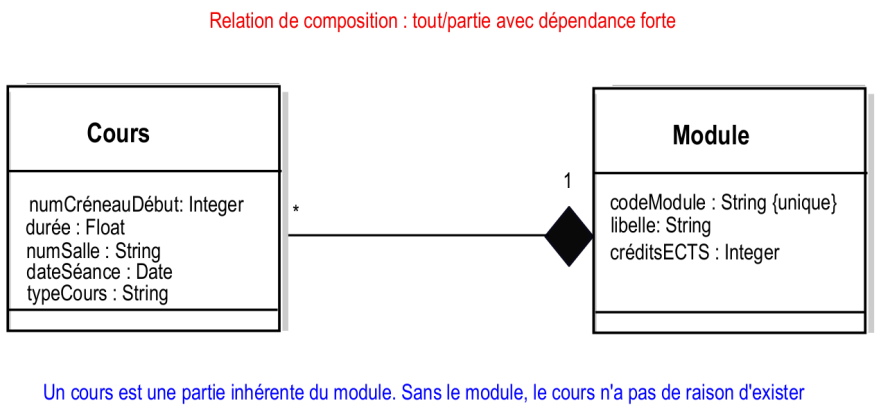


Figure: Dépendance existentielle - optimisation

Association de composition

Schéma relationnel Module

Module(codeModule varchar(8), libelle varchar(20), creditsECTS integer)

Schéma relationnel Cours

Une contrainte de clé primaire et une contrainte de clé étrangère
Cours(numCours integer, dateCours Date, numSalle varchar(6), numCreneauDebut integer, nombreCreneaux integer, codeModule varchar(8))
avec Cours(codeModule) $\subseteq$ Module(codeModule)

Association de composition : nouvel exemple

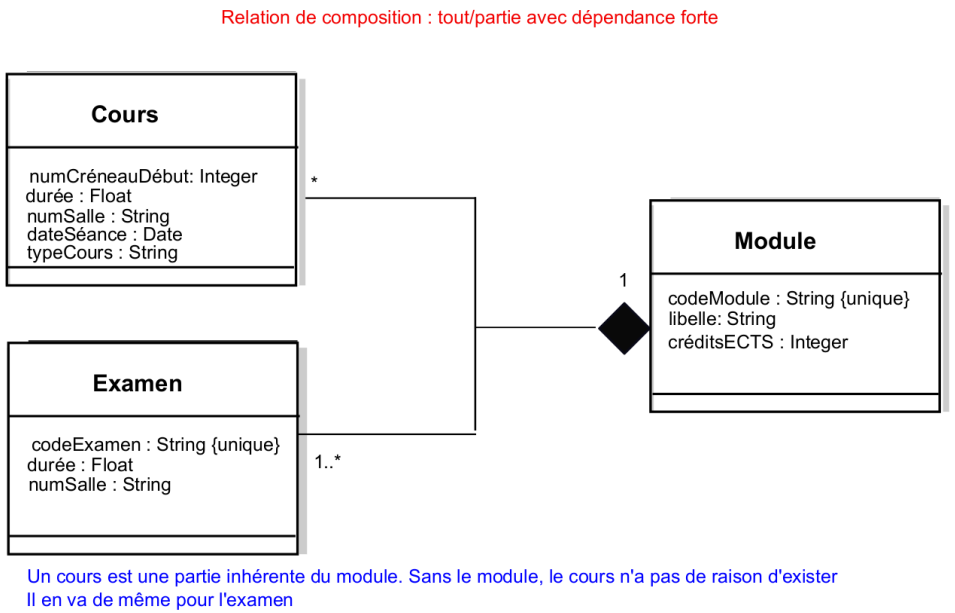


Figure: Identication partielle

Traduction (la relation Module reste inchangée)

Schéma relationnel Examen

Examen(codeExamen varchar(8), duree float, numSalle varchar(6), codeModule varchar(8))
avec Examen(codeModule) $\subseteq$ Module(codeModule)

Schéma relationnel Cours

Cours(numOrdre integer, codeModule varchar(8), dateCours Date, numSalle varchar(6), numCreneauDebut integer, nombreCreneaux integer,)
avec Cours(codeModule) $\subseteq$ Module(codeModule)

Association d'héritage

Trois façons de faire, en fonction du contexte

1. garder tous les niveaux
2. aplatir vers le haut
3. aplatir vers le bas

Héritage : une vraie difficulté pour le relationnel : aucune solution idéale

Association d'héritage

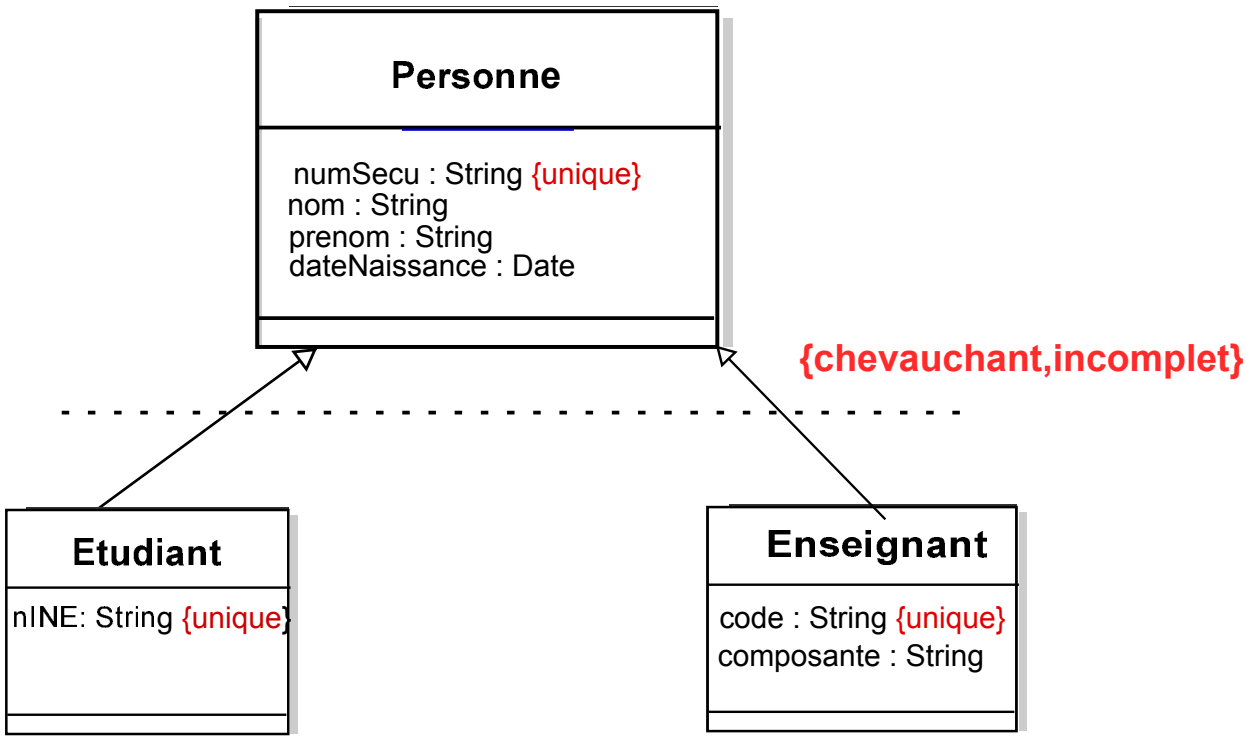


Figure: Premier exemple

Deux options possibles

Aplatir vers le haut ne peut pas convenir : un étudiant pouvant être aussi enseignant

1. garder tous les niveaux
2. aplatir vers le bas

Garder tous les niveaux

Schémas relationnels Personne, Etudiant et Enseignant

Personne(<u>numSecu varchar(12)</u>, nom varchar(20), prenom varchar(20), dateNaissance Date)
Etudiant(<u>numSecu varchar(12)</u>,numINE varchar(12))
avec Etudiant(numSecu) $\subseteq$ Personne(numSecu)
Enseignant(<u>numSecu varchar(12)</u>,code varchar(15), composante varchar(10))
avec Enseignant(numSecu) $\subseteq$ Personne(numSecu)

Garder tous les niveaux : disposer de toutes les données pour un étudiant ou un enseignant

Jointure naturelle entre Personne et Etudiant, ou entre Personne et Enseignant

$$\text{Personnne P} \underset{P.numSecu=E.numSecu}{\bowtie} \text{Etudiant E}$$

$$\text{Personnne P} \underset{P.numSecu=E.numSecu}{\bowtie} \text{Enseignant E}$$

ou même entre Etudiant et Enseignant pour les étudiants également enseignants

$$\text{Personnne P} \bowtie \text{Etudiant E1} \bowtie \text{Enseignant E2}$$

Aplatir vers le bas

Schémas relationnels Etudiant et Enseignant

Etudiant(numSecu varchar(12), nom varchar(20), prenom varchar(20), dateNaissance Date,numINE varchar(12))
Enseignant(numSecu varchar(12), nom varchar(20), prenom varchar(20), dateNaissance Date,code varchar(15), composante varchar(10))

Reconstruire Personne

$\Pi_{numSecu,nom,prenom,dateNaissance}$ (Personne) $\equiv$
$\Pi_{numSecu,nom,prenom,dateNaissance}$ (Etudiant) $\cup$
$\Pi_{numSecu,nom,prenom,dateNaissance}$ (Enseignant)

Association d'héritage

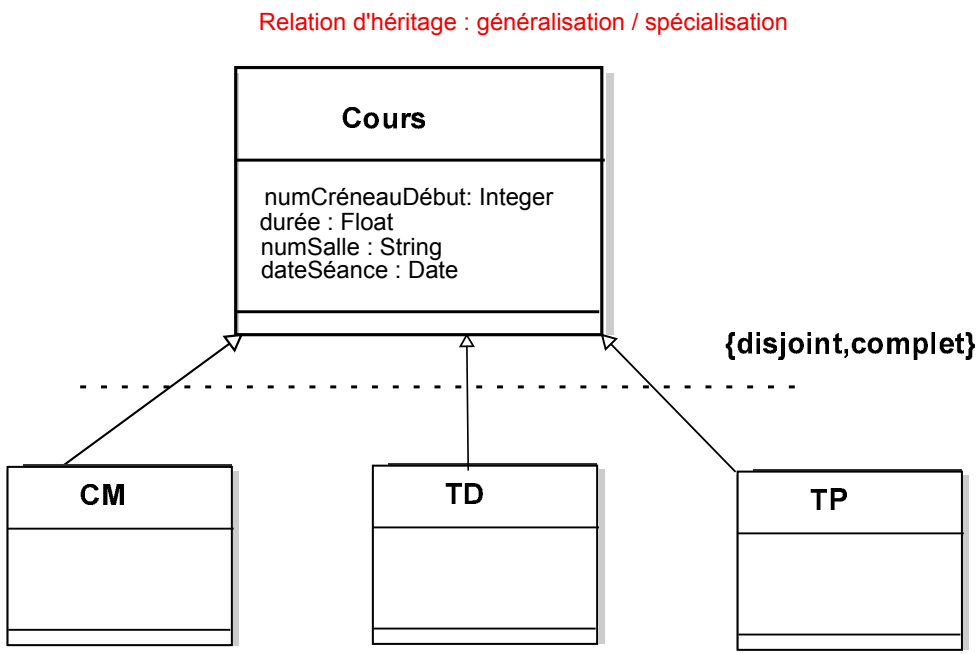


Figure: Second exemple

Aplatir vers le haut : le plus simple

Schéma relationnel Cours

Cours(numCours integer, numCreneauDebut integer, duree float, numSalle varchar(6), dateSeance Date, typeCours varchar(4)))

Inconvénient

Pourrait entraîner l'obligation de gérer des attributs souvent non renseignés, si CM, TD et TP avaient des attributs spécifiques

Rétroconception : Exemple de TP

Schéma relationnel d'origine

DEPT(n_dept integer, nom varchar(14), lieu varchar(13))
EMP(num integer, nom varchar(10), fonction varchar(15), n_sup integer, embauche Date, salaire number(7,2), comm number(7,2), n_dept integer)
avec EMP(n_dept) $\subseteq$ DEPT(n_dept)
avec EMP(n_sup) $\subseteq$ EMP(num)

Dérivation inversée

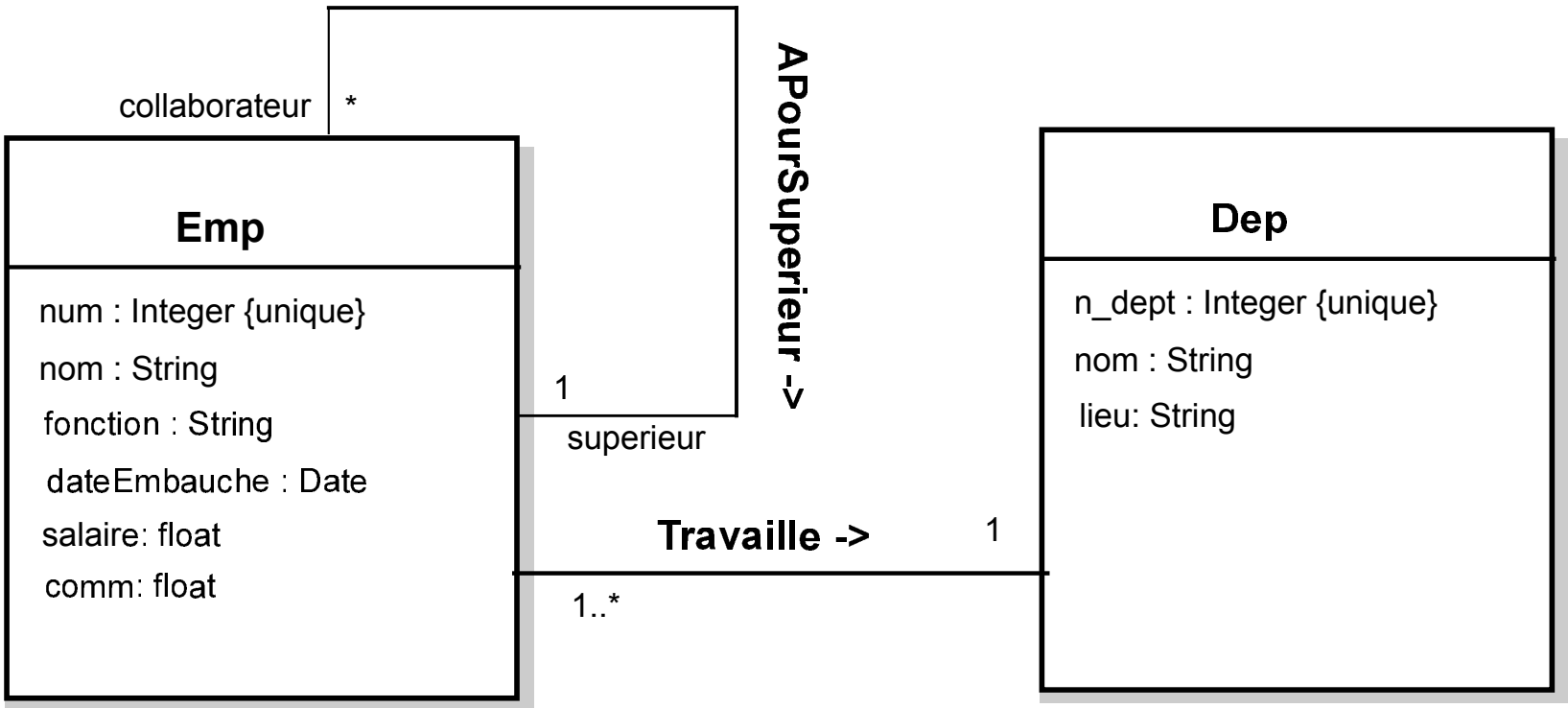


Figure: Rétroconception