

UML pour les bases de Données

Inspiré du livre de même titre de Ch. Soutou

Attention, pendant ce cours un rapprochement avec le modèle entité-relation sera fait.

Mireille Blay-Fornarino
IUT Nice-Sophia Antipolis

blay@polytech.unice.fr

<http://www.polytech.unice.fr/~blay>

Site web du module : <http://anubis.polytech.unice.fr/iut>

3 niveaux

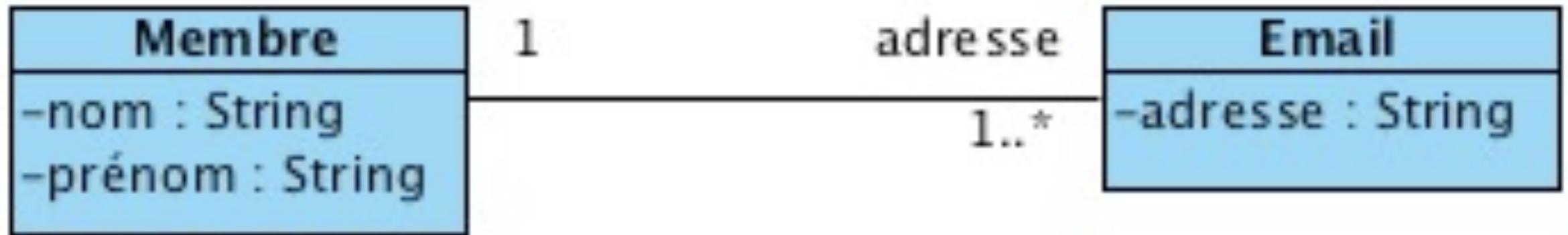
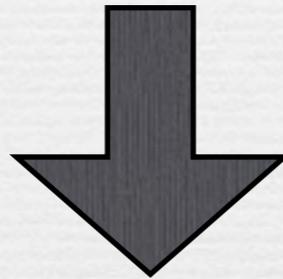
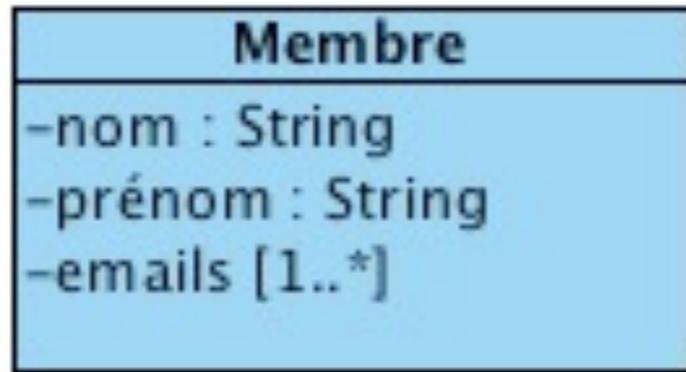
- Le niveau Conceptuel (=> Modèle Conceptuel de Données)
 - Représentation abstraite des informations du domaine géré par le futur système d'information
- Le niveau Logique (=> Modèle Logique de Données)
 - description des données prenant en compte les moyens informatiques de mémorisation et l'implémentation du système par un SGBD.
 - Le niveau interne diagrammes UML stéréotypés (ou MLD) ou notation mathématique
- Le niveau Physique
 - Expression des choix techniques et description des données de la base de données (dans la syntaxe du système de gestion adopté.)

Des classes bien construites

- Représentation abstraite des informations du domaine géré par le futur système d'information

Abstraite oui, mais pas n'importe quoi!

Un attribut a au plus une valeur à un instant t



Relatif à la Première forme normale

Notion de dépendance fonctionnelle

Considérons l'entité suivante et quelques une de ses occurrences :

1374 : Article
désignation = rateau LibelléCatégorie = Jardinage prixUnitaireHT = 10 noCategorie = J

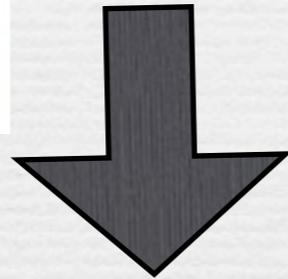
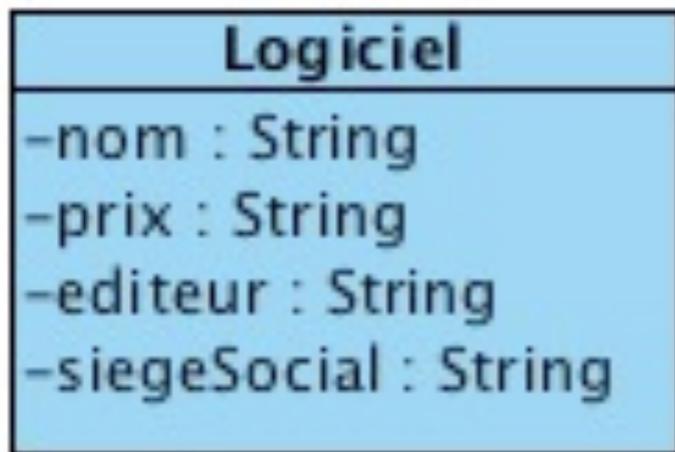
354YT : Article
désignation = Beche noCategorie = J prixUnitaireHT = 7 LibelléCatégorie = jardinage

452GT : Article
désignation = scie noCategorie = B prixUnitaireHT = 5 LibelléCatégorie = Bricolage

Article
<<PK>> -Référence : String -désignation : String -prixUnitaireHT : int -LibelléCatégorie -noCategorie : String

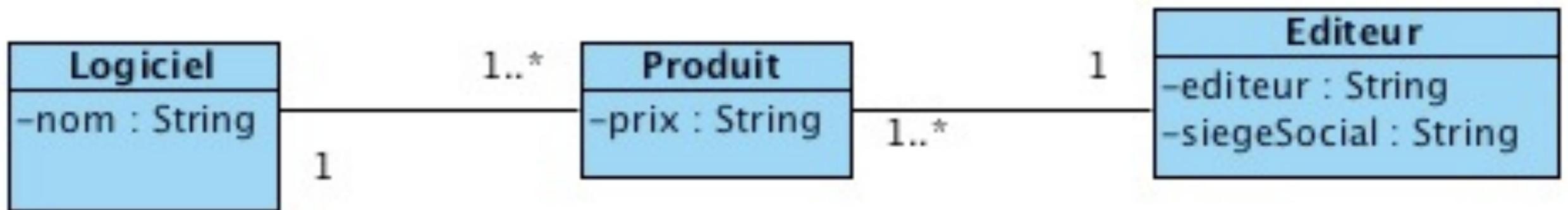
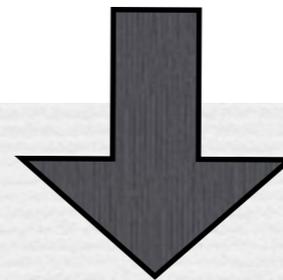
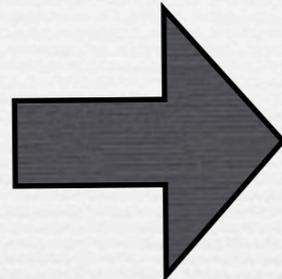
Cette entité est juste mais elle implique une redondance d'information relative à la catégorie. L'association entre le numéro de la catégorie et son libellé est en effet répétée dans chaque occurrence de l'entité ARTICLE.

Chaque attribut doit dépendre uniquement de la classe qui le définit



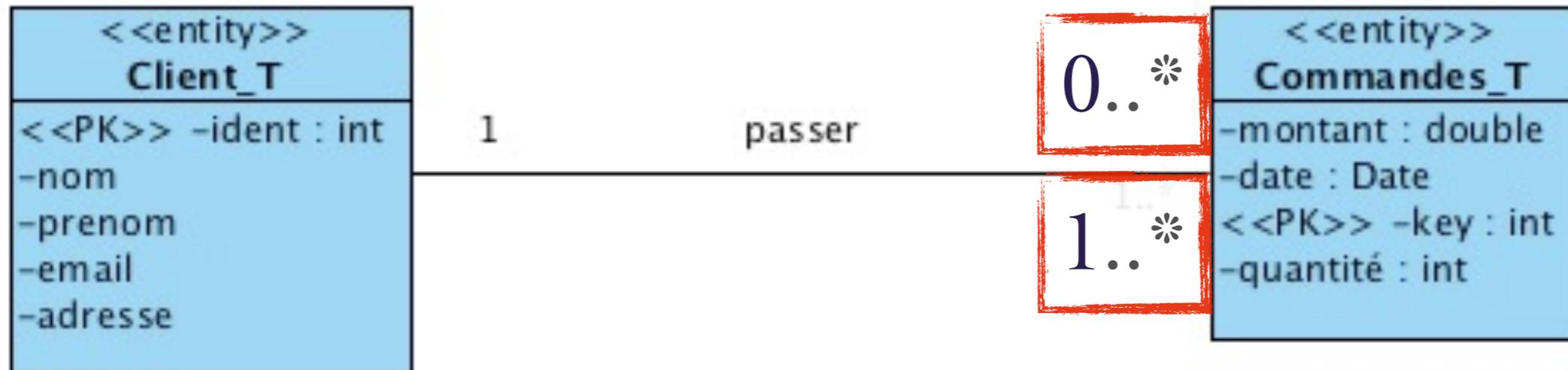
Relatif à la 2e forme normale

Chaque attribut doit dépendre uniquement de la classe qui le définit



Relatif à la 2e forme normale

Choix des modélisations ?



Une **commande** est toujours passée par au moins un client. Une commande est également passée au maximum par un client. Une commande est donc toujours passée par un et un seul client.

Un client passe au moins une commande et au maximum plusieurs (N) commandes. Cette modélisation ne tient pas compte des clients qui ne passent aucune commande. Un client est uniquement considéré comme tel s'il passe au moins une commande

Un client peut passer aucune commande et au maximum plusieurs (N) commandes. Cette modélisation tient compte des clients qui ne passent aucune commande.

Des classes aux bases de données

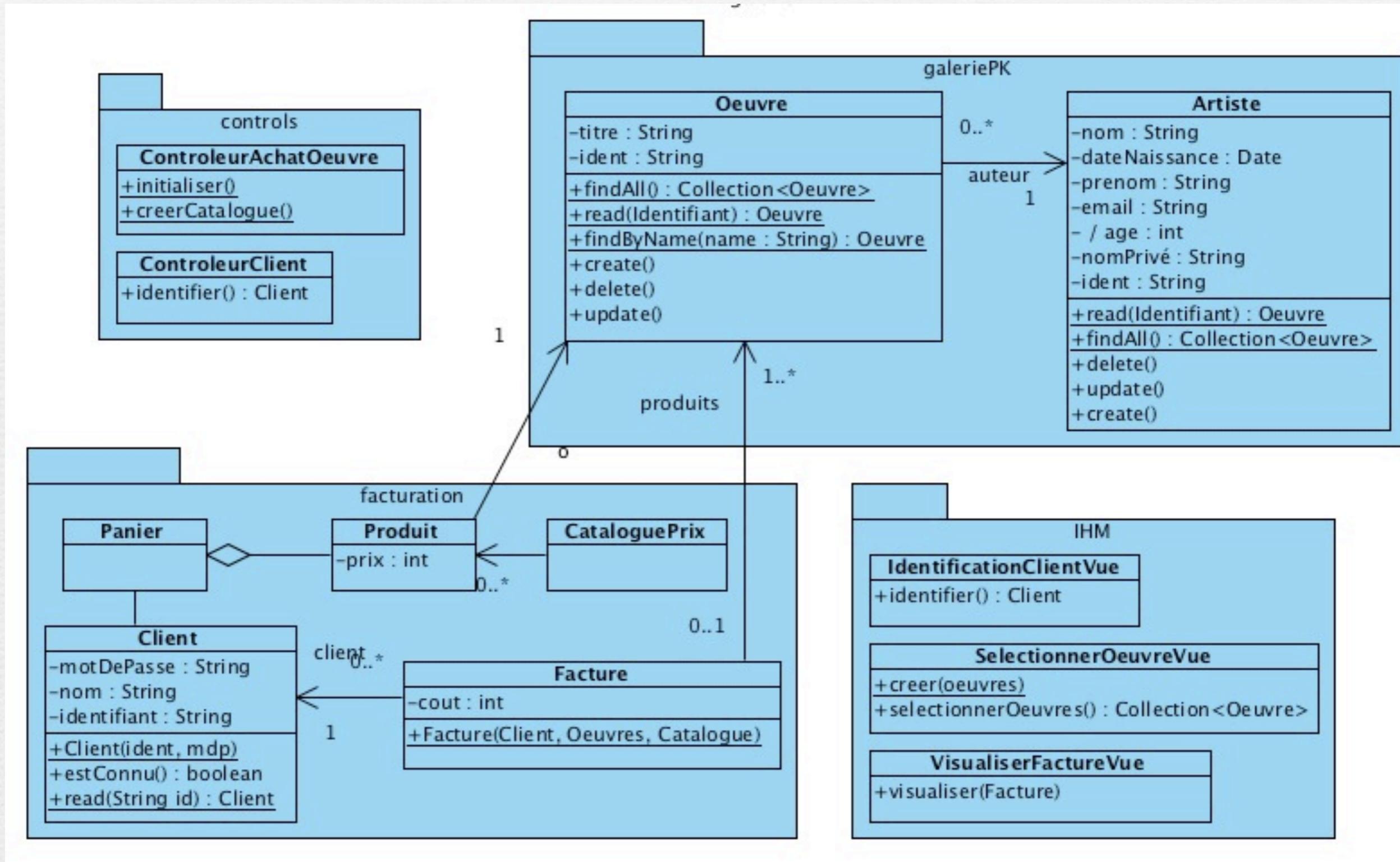
Conversion d'un modèle de classes en schéma relationnel

- À partir d'un diagramme de classes décrit en UML,
 - On sélectionne les classes qui correspondent à des données persistantes.
 - Puis on passe au schéma relationnel correspondant.
- L'objectif est donc de :
 - représenter **toutes** les informations présentes dans le diagramme des classes sélectionné (classes et associations) ;
 - éviter les redondances (répétition des mêmes associations) ;
 - limiter le nombre de relations (schéma complexe) ;
 - limiter les valeurs absentes (e.g., les valeurs NULL).

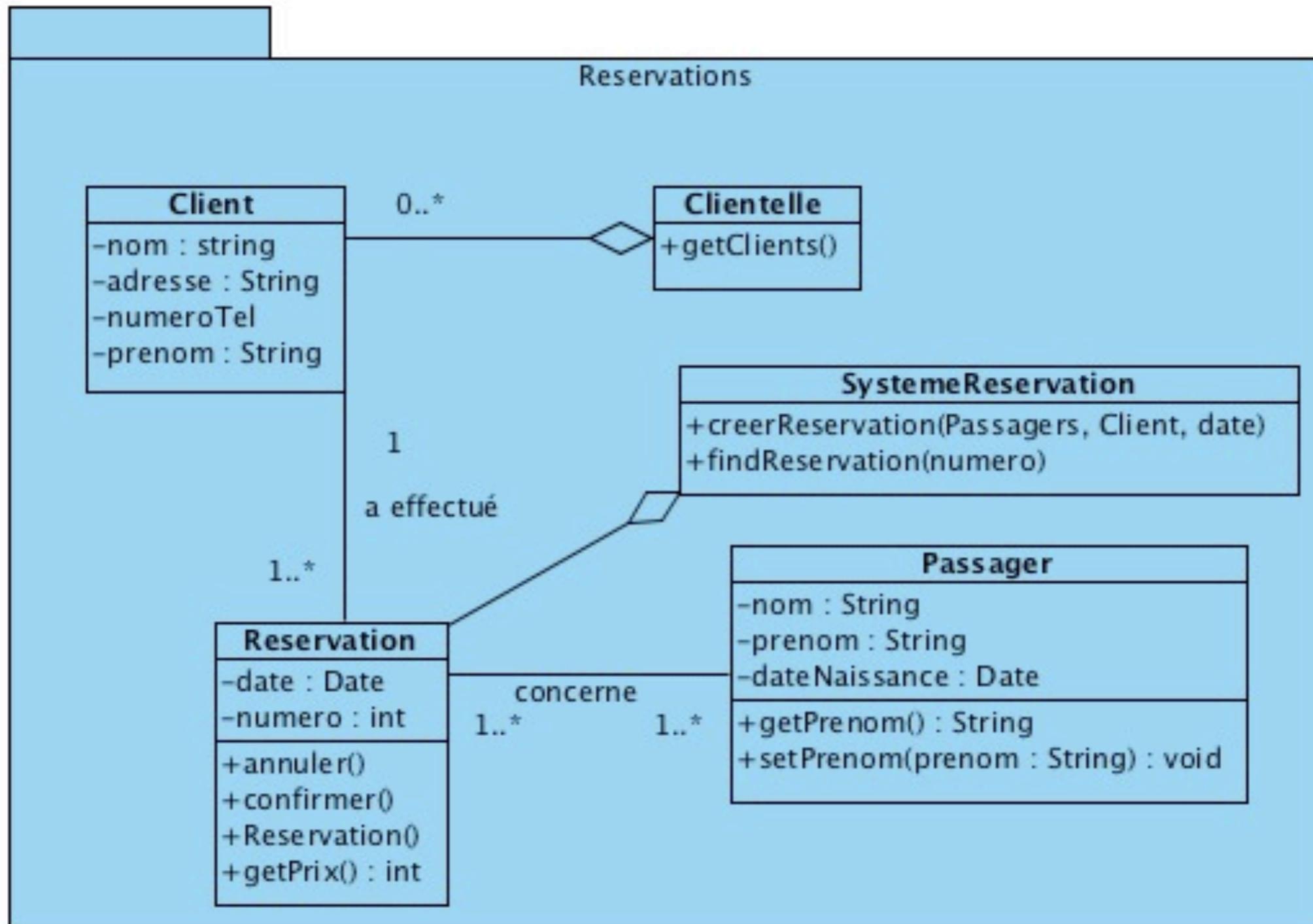
● <http://www.math-info.univ-paris5.fr/~pellier/teaching/others/bd/cours/Modelisation.pdf>

Caractérisation des données «persistantes»

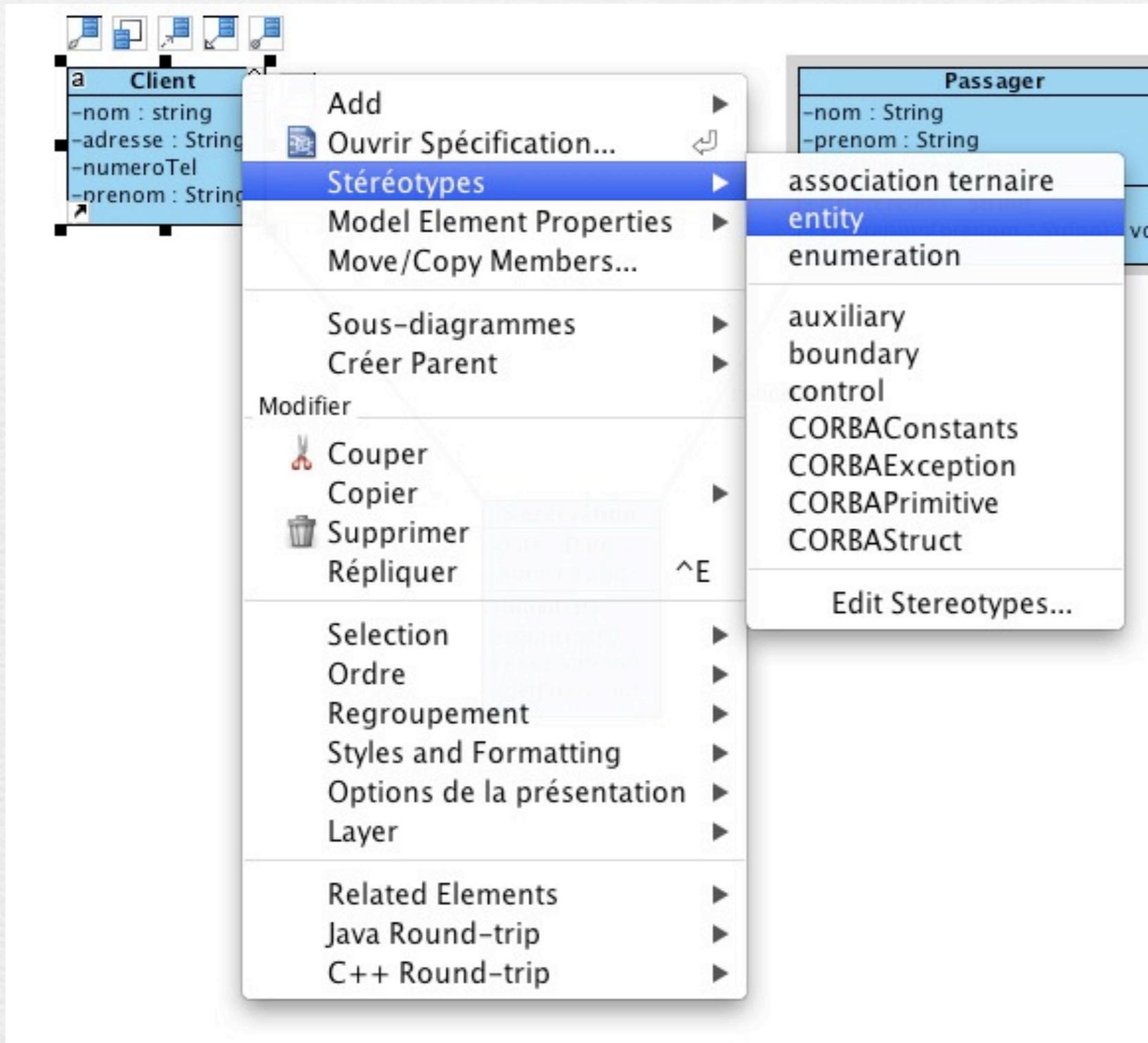
Données Persistantes ?



Données Persistantes ?

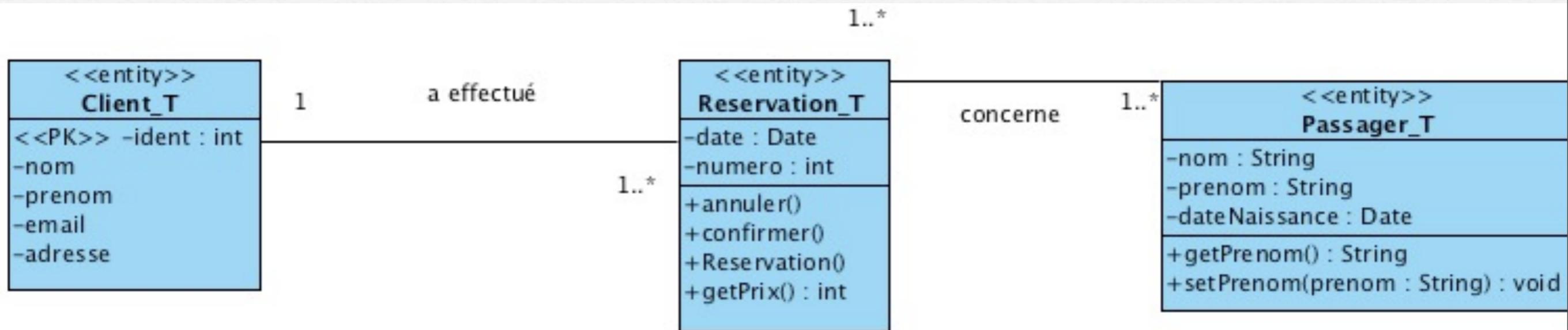


Données Persistantes ?



Attention une donnée persistante n'est pas seulement une table dans une base de données, elle peut exister aussi comme un objet à l'exécution : construction, calculs, ...

Données Persistantes ?



Attention une donnée persistante n'est pas seulement une table dans une base de données, elle peut exister aussi comme un objet à l'exécution :
 construction, calculs, ...

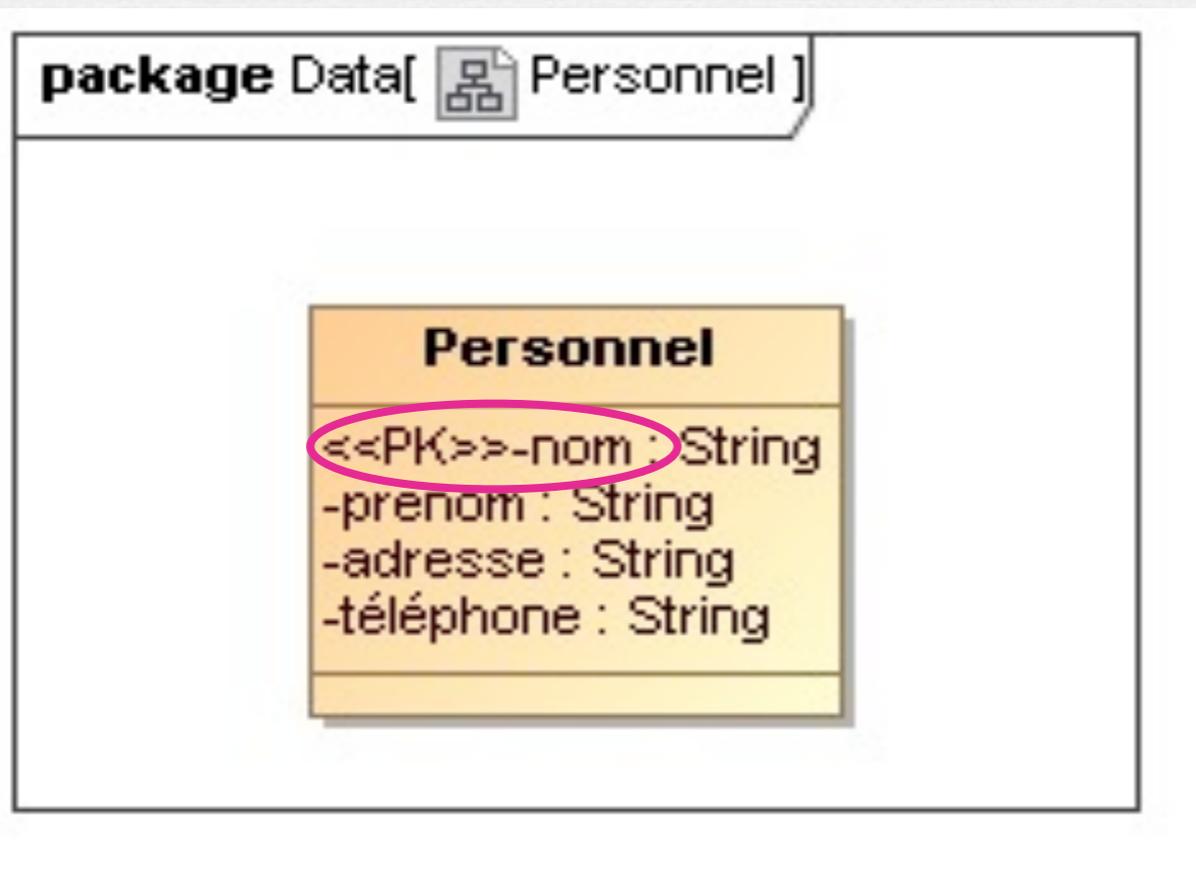
Détermination des «clefs» primaires

Affecter les clefs

- Clef naturelle – un attribut qui fait du sens dans l'environnement métier
 - (e.g. Numéro de sécurité sociale pour une personne... **Mais Danger !!**)
- Clef de substitution – une clé qui n'a aucune signification sens (e.g. a attribut «compteur»)
 - Utilisez pour la performance
 - Utiliser quand il n'est pas facile à identifier de manière unique une entité
- Clé composite – une clef de plus d'un champ
 - Peut être utilisé pour identifier l'entité de manière unique
- Clé étrangère - représente un ou plusieurs champs d'une autre entité utilisée pour maintenir les relations entre les «rangées»

Classe identifiée comme «entité»...

- Implémentations des attributs, généralisations, et associations sous forme de tables



```

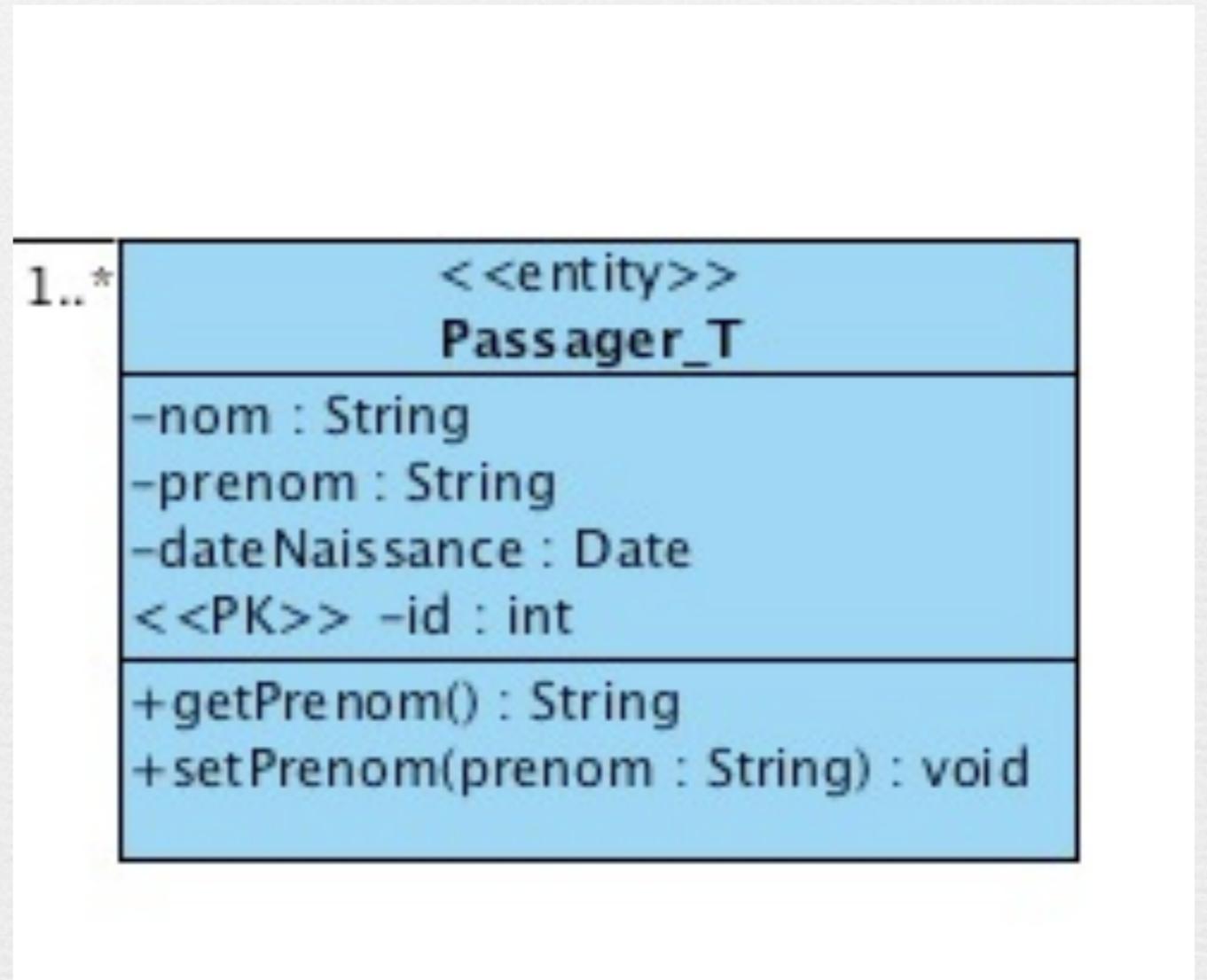
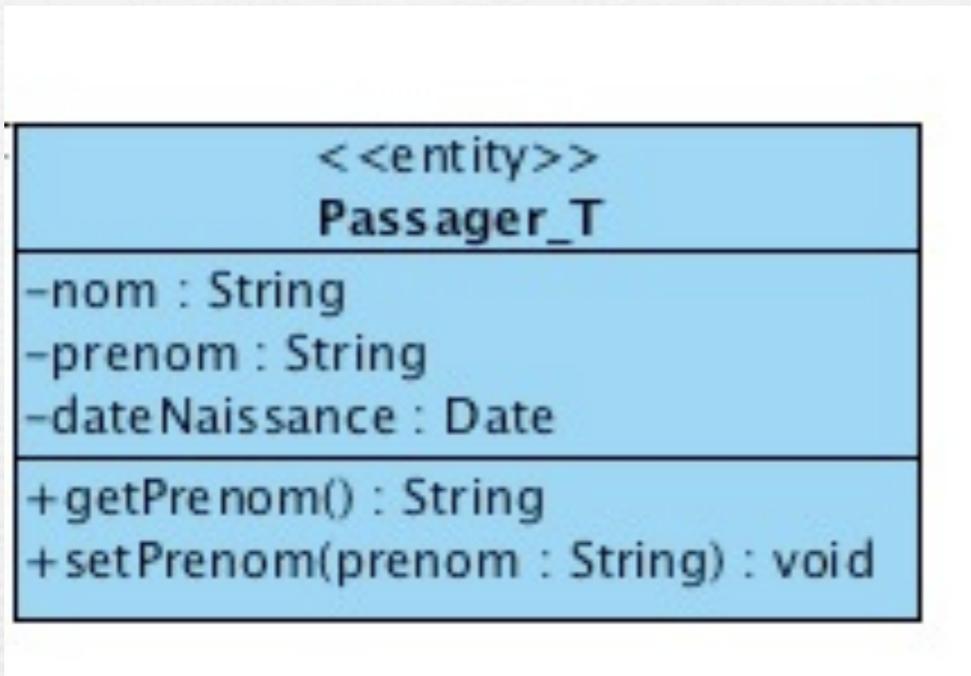
--
-- Structure de la table `personnel`
--
  
```

```

CREATE TABLE `personnel` (
  `nom` varchar(15) NOT NULL default "",
  `prenom` varchar(15) default NULL,
  `adresse` longtext,
  `telephone` varchar(12) default NULL,
  PRIMARY KEY (`nom`)
);
  
```

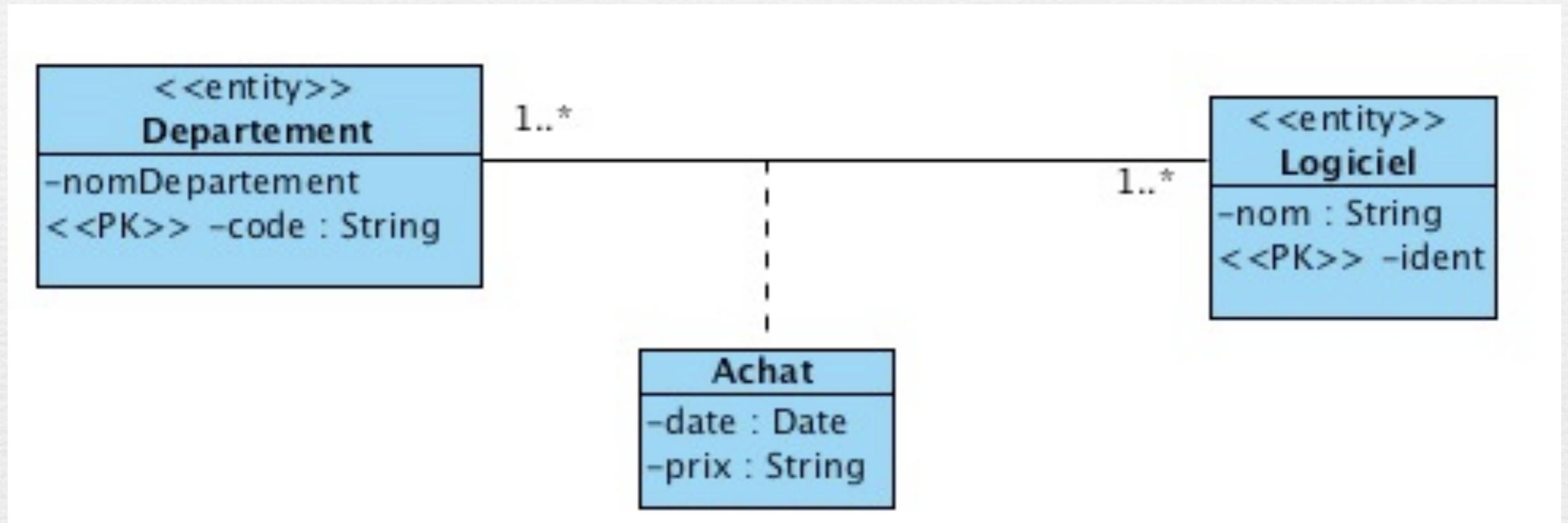
Identification des clefs primaires (ici utilisation de stéréotype)

Clefs...



Identification des clefs primaires (ici utilisation de stéréotype)

Clefs



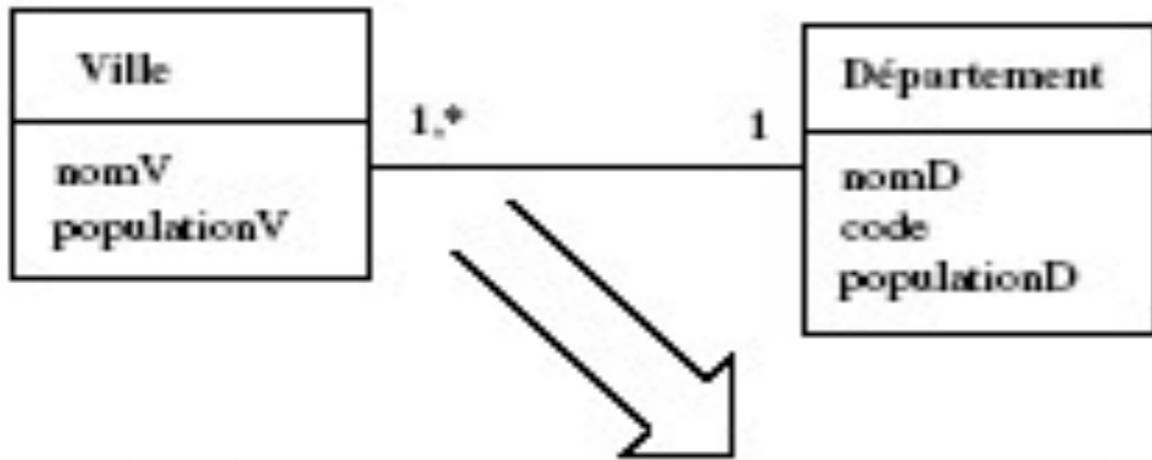
Transformation des associations

Transformation des associations

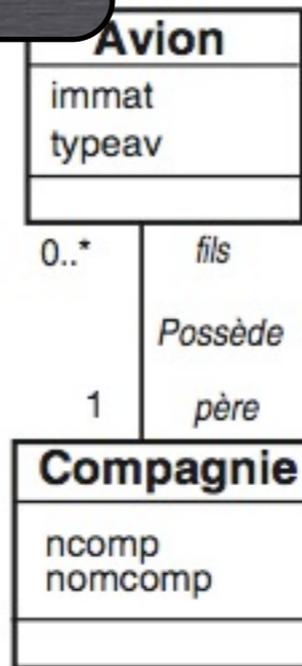
- Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations :
 - un-à-plusieurs ;
 - plusieurs-à-plusieurs ou classes-associations, et n-aires ;
 - un-à-un.

Association mono-multi

Ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.



LesDepartements(code, nomD, populationD)

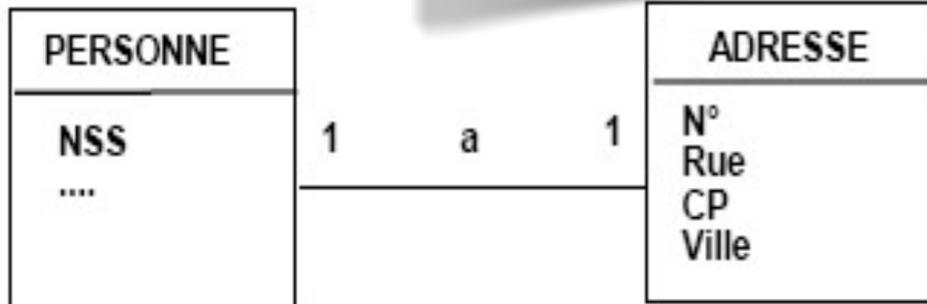


R1 et R2

Association mono-mono

Une seule table ?

Associations



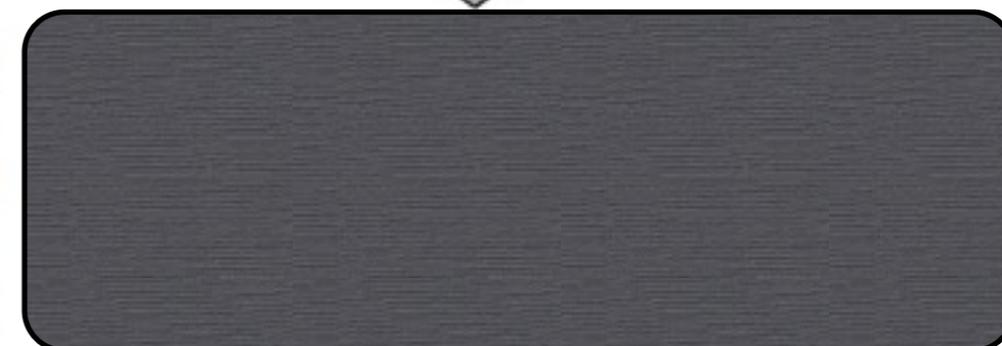
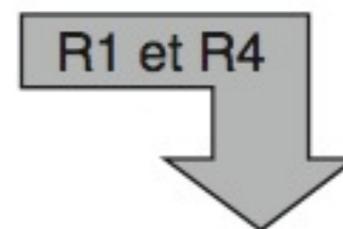
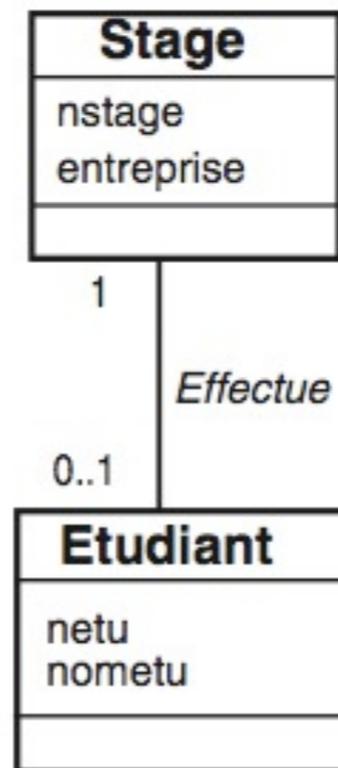
Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à un.

Schéma Personne

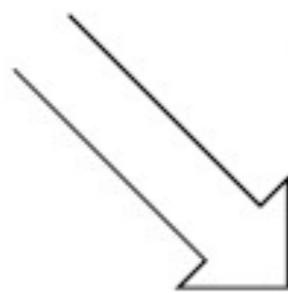
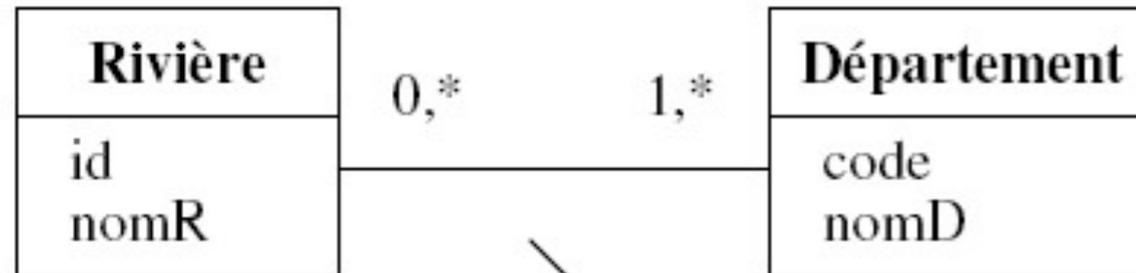
Clé primaire
NSS

Clé étrangère
Id_adresse

Attribut	Domaine	Non Null
<u>NSS</u>	String(13) ID	Oui
Nom	String(35)	Oui
<u>Id_adresse</u>	Identifiant	Oui

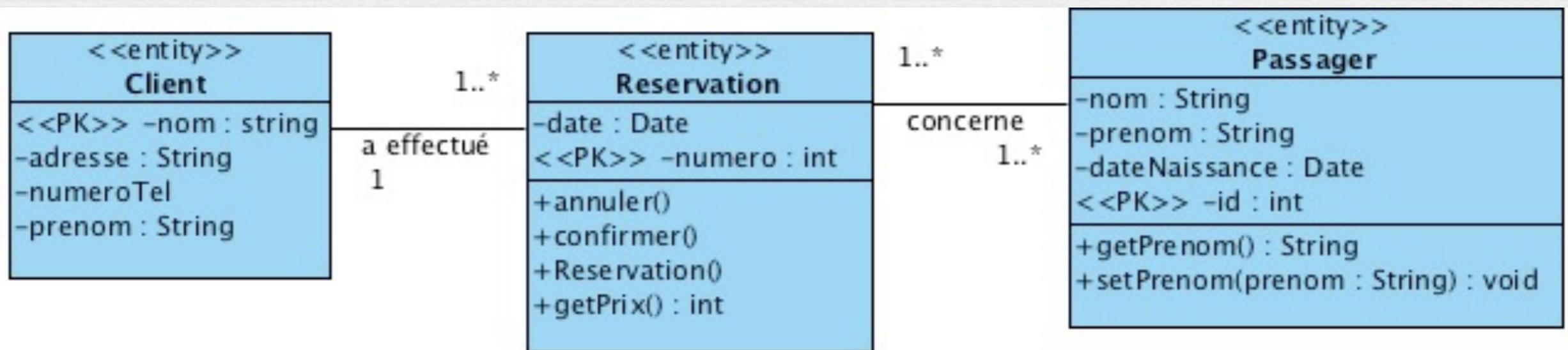


Association multi-multi



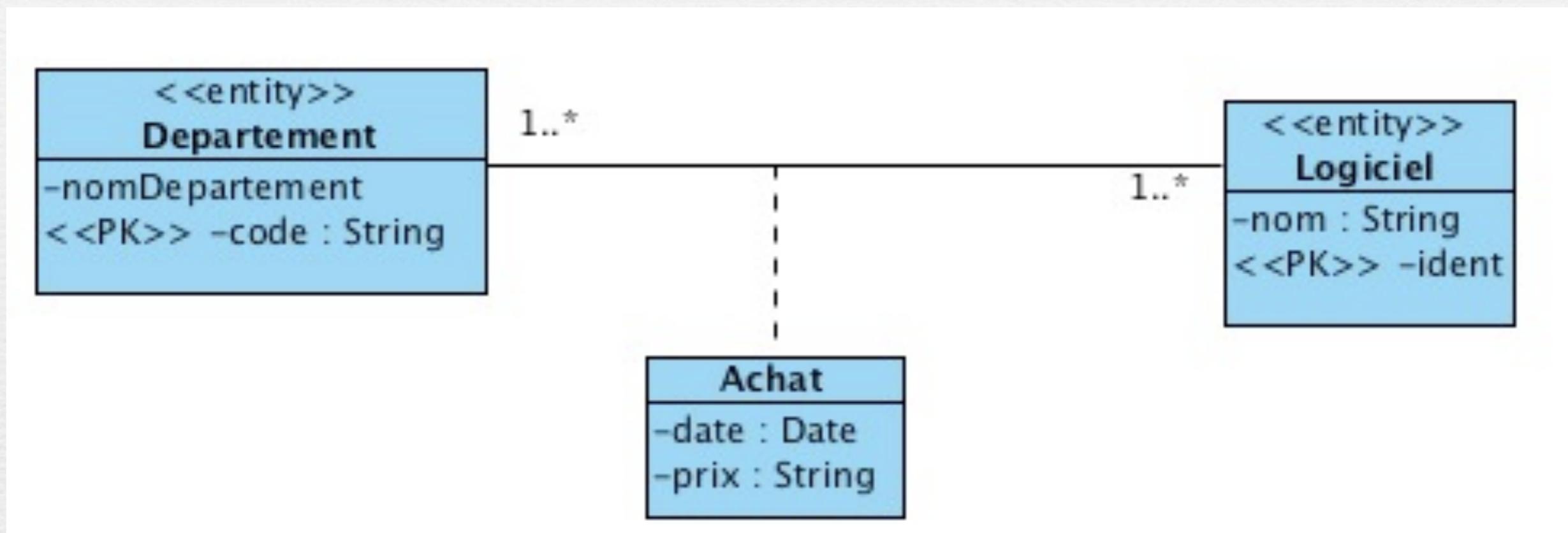
LesLocalisations(id, code)
 LesRivieres(id, nomR)
 LesDepartements(code, nomD)

L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des entités (classes) connectés à l'association.



Classe d'association

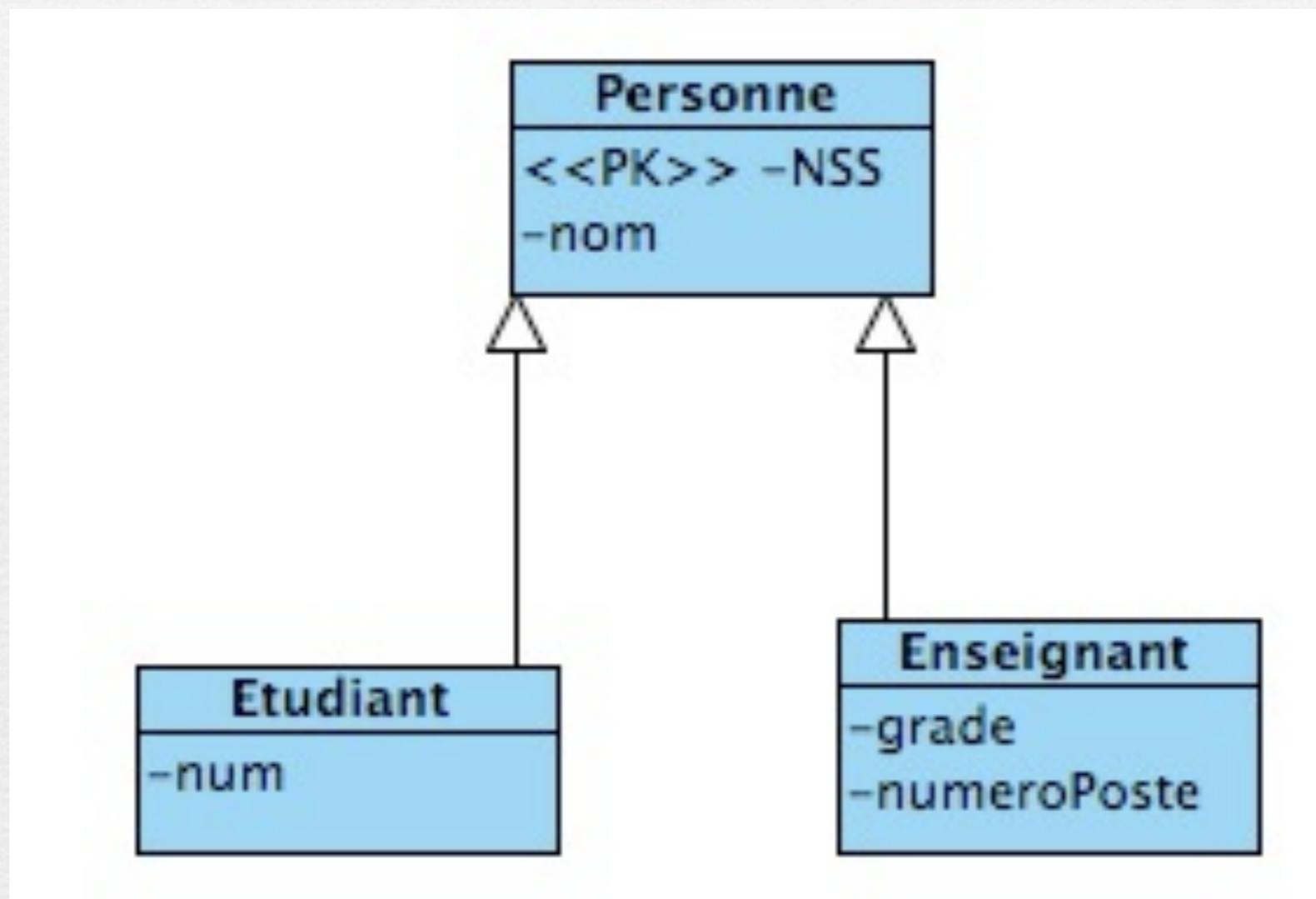
Comme pour une association multi-multi



Achat[code\$, ident\$, date, prix]

Transformation des relations d'héritage

Héritage



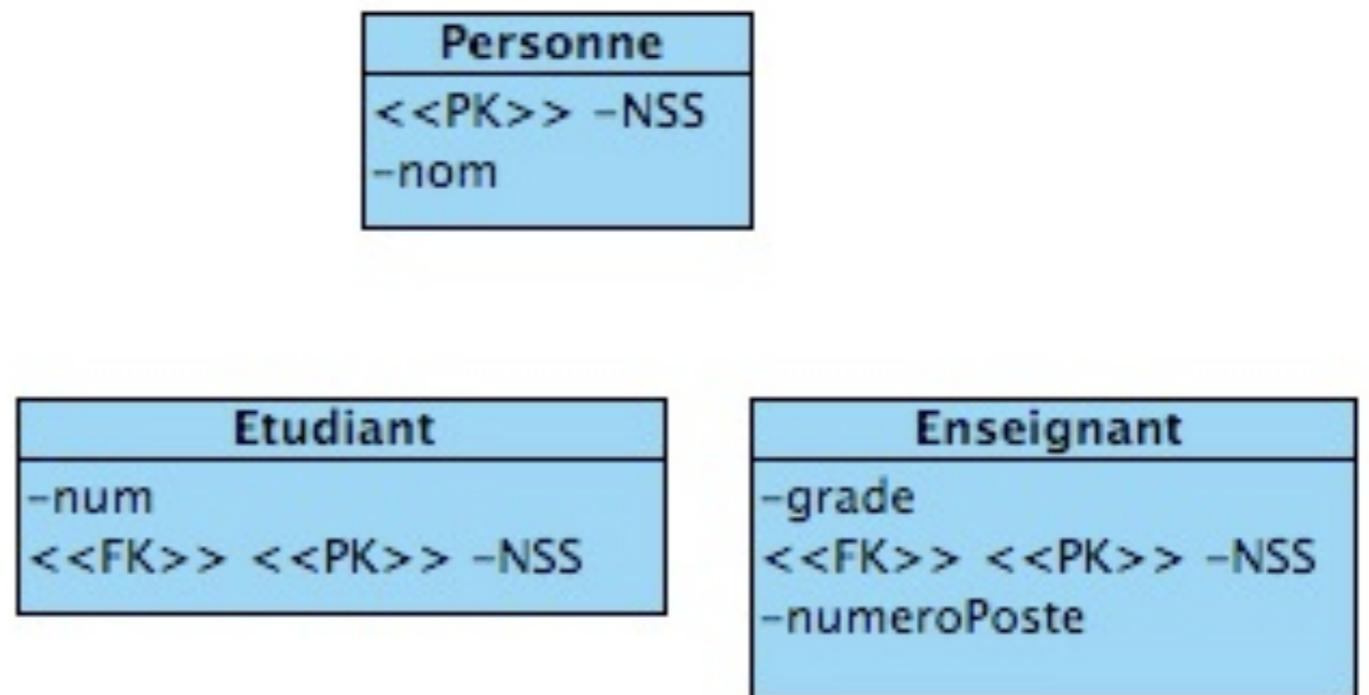
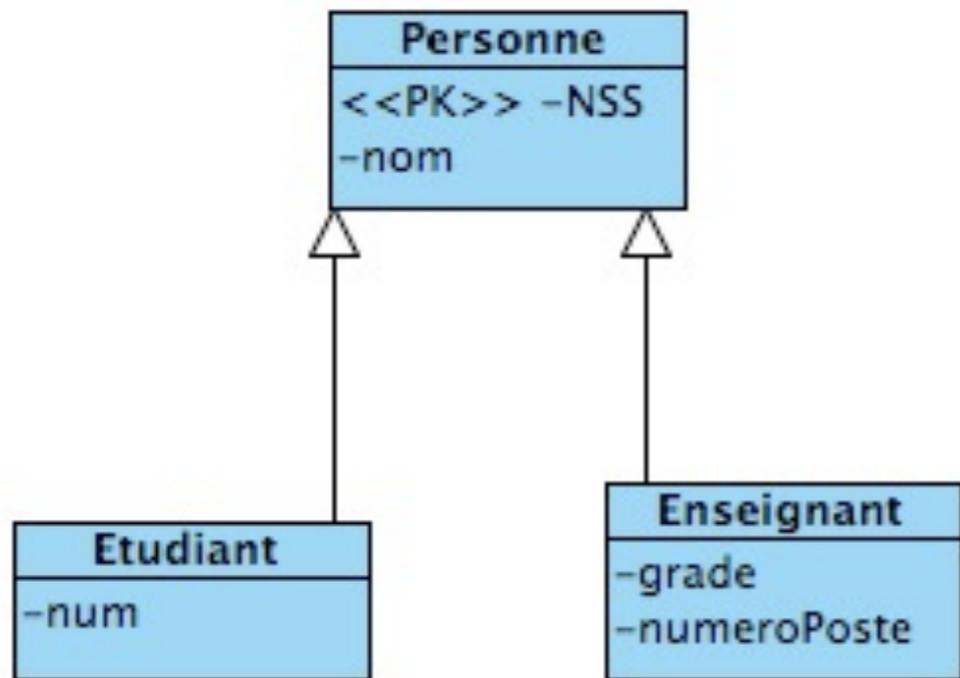
*UML 2 pour
les bases de données,
Christian Soutou*

- Trois décompositions sont possibles
- décomposition par distinction
 - décomposition descendante
 - décomposition ascendante

Décomposition par distinction

Chaque sous-classe est transformée en une relation.

La clé primaire de la sur-classe migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.

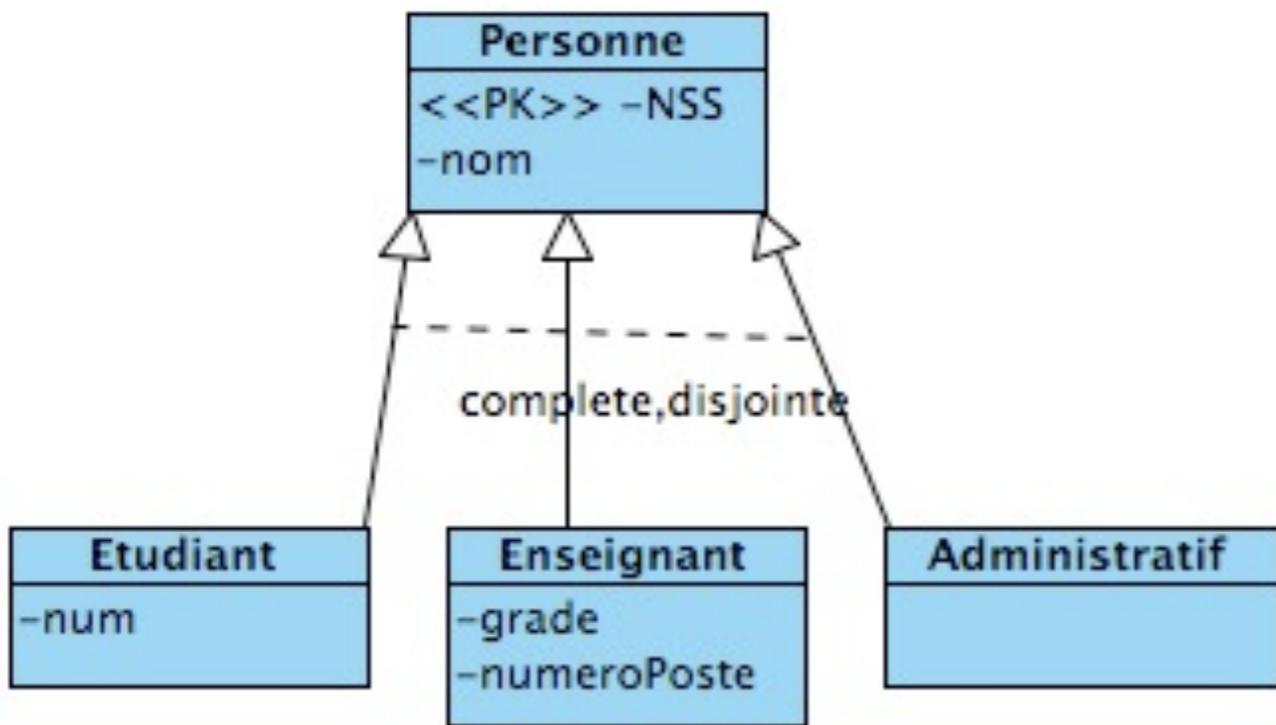


Personne [NSS, nom]

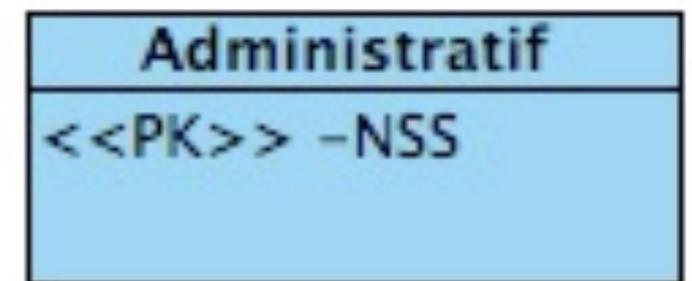
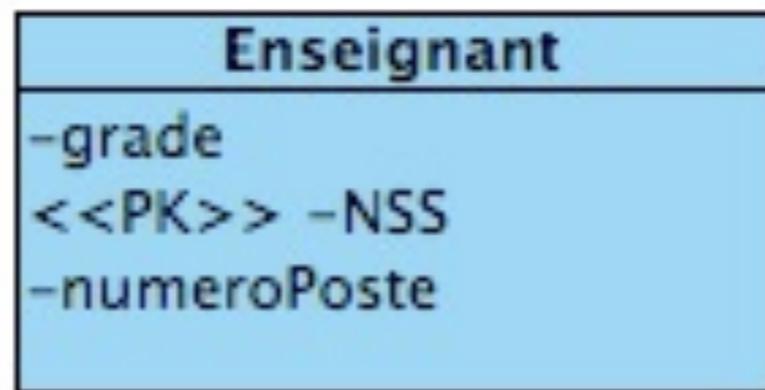
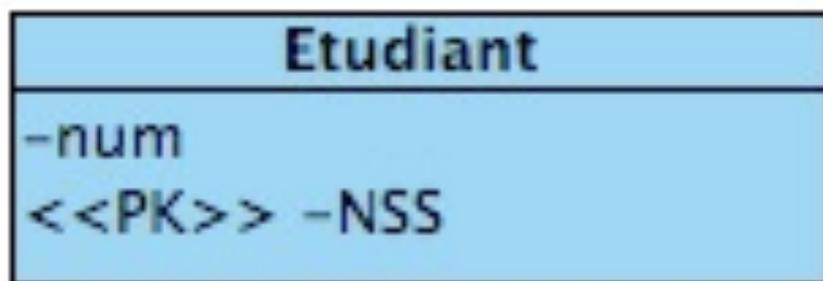
Etudiant [NSS#, num]

Enseignant [NSS#, numeroPoste, grade]

Décomposition descendante



S'il existe une contrainte de totalité ou de partition sur l'association, il est possible de ne pas traduire la relation issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).



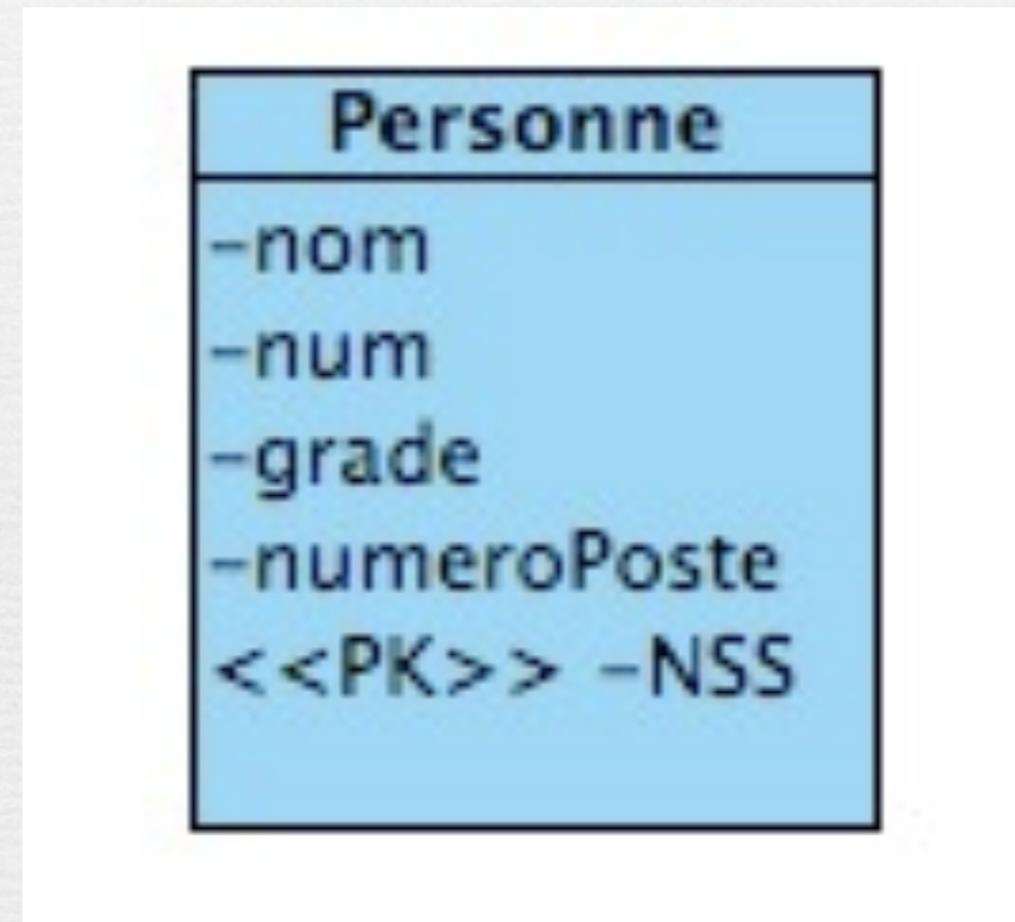
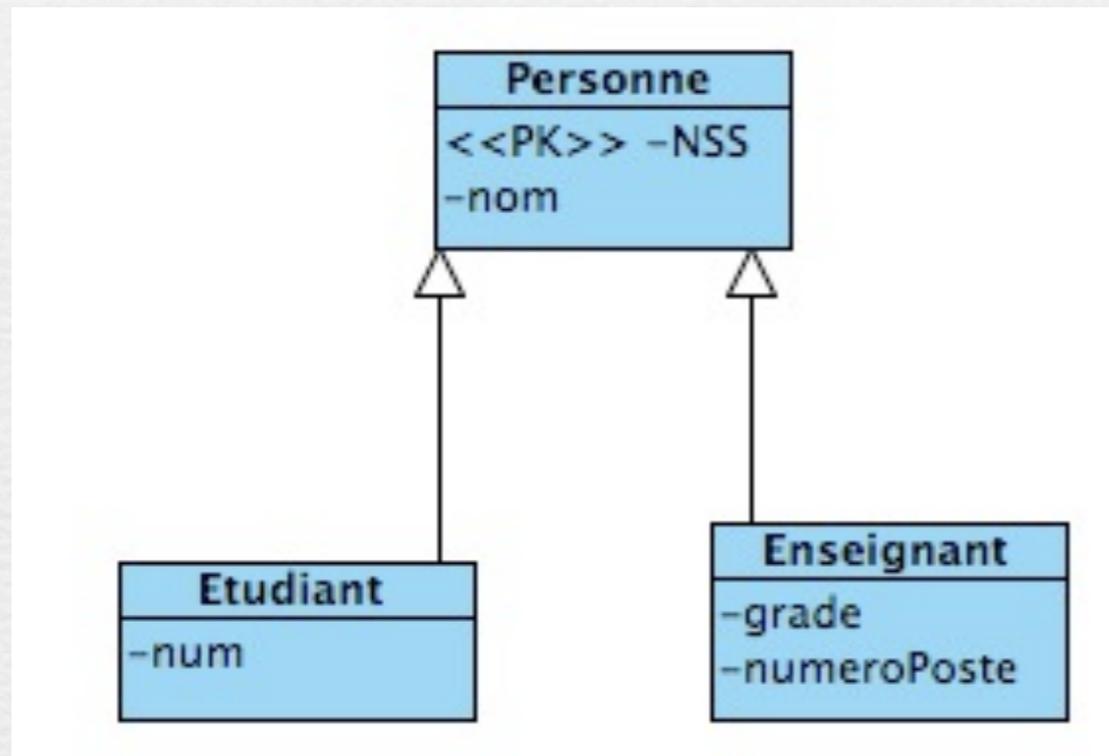
Qui a été oublié ?

Etudiant [NSS, num]

Enseignant [NSS, numeroPoste, grade]

Décomposition ascendante

Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la sur-classe.

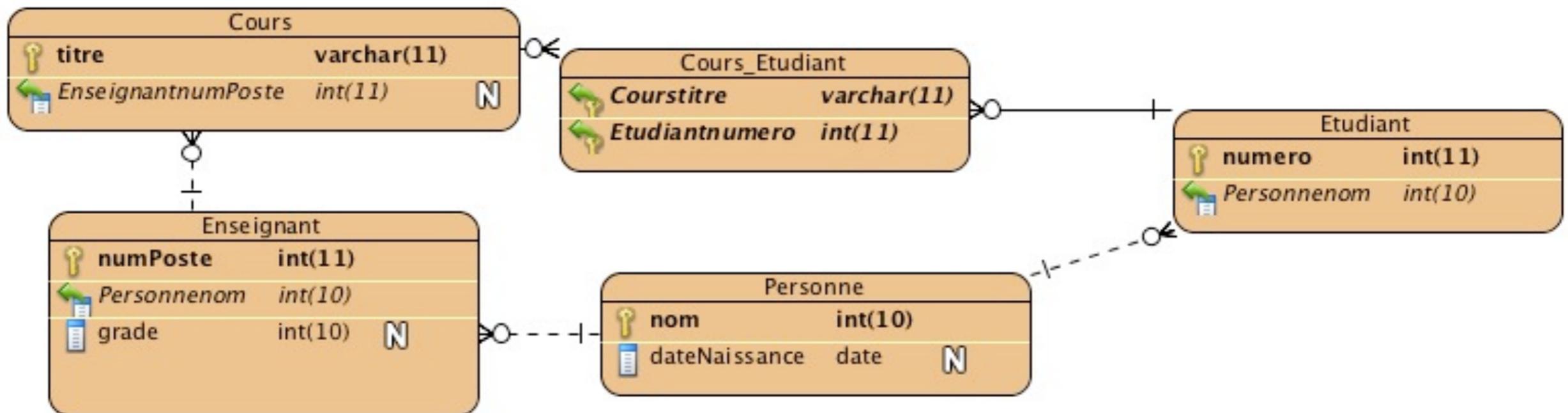
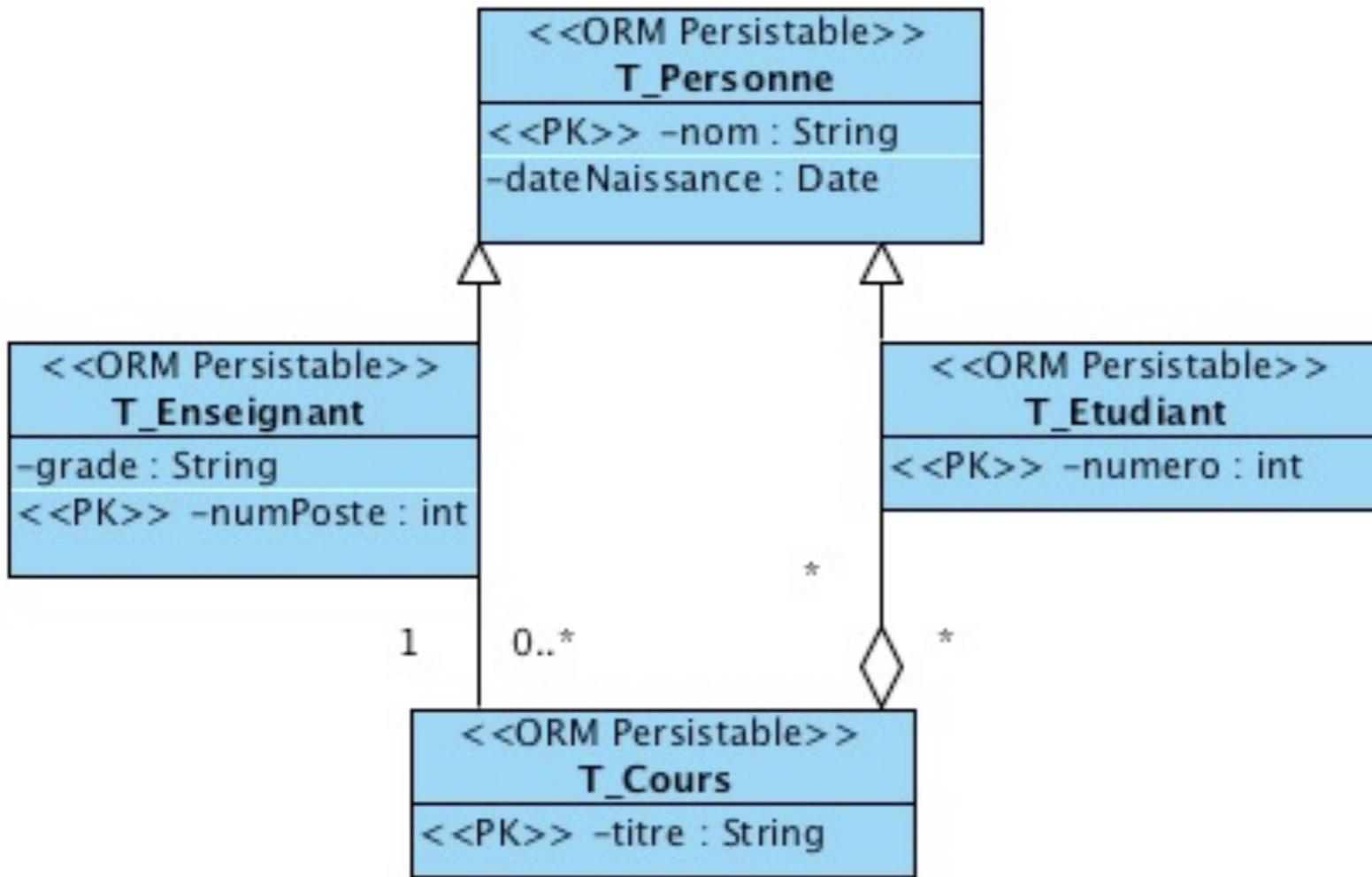


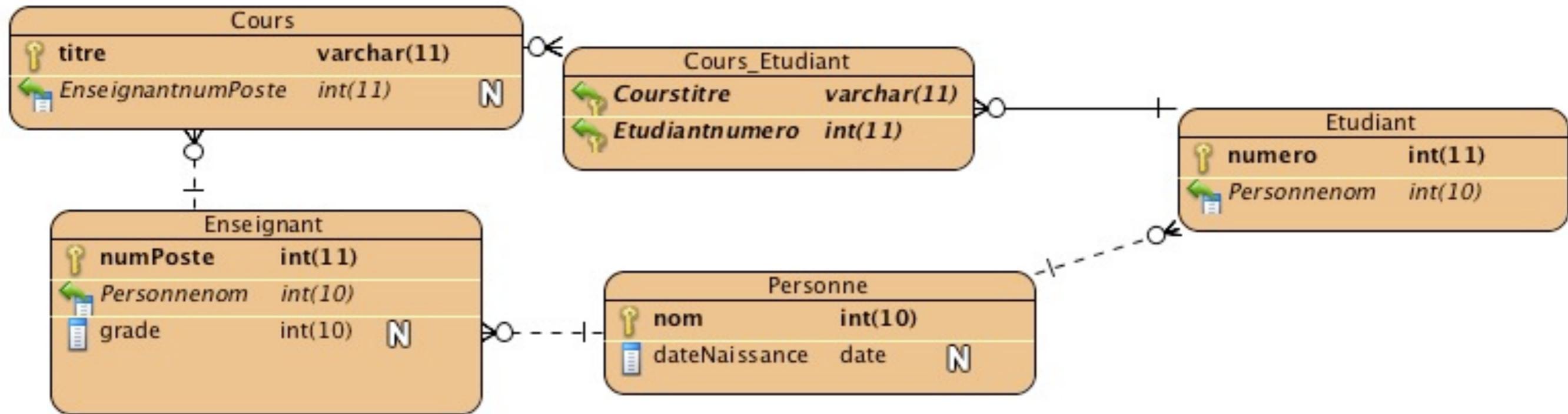
Personne [NSS, nom, grade, numeroPoste]

Caractéristiques du modèle logique de données

- D'un niveau de détails
 - ▶ Suffisant et sans ajout de détails inutiles
- Doit supporter les questions utilisateurs
 - ▶ Information needed by the user can be retrieved based on the information the model depicts
- Pas de redondance
- Indépendant des technologies de mise en oeuvre

Autre notation :
ORM



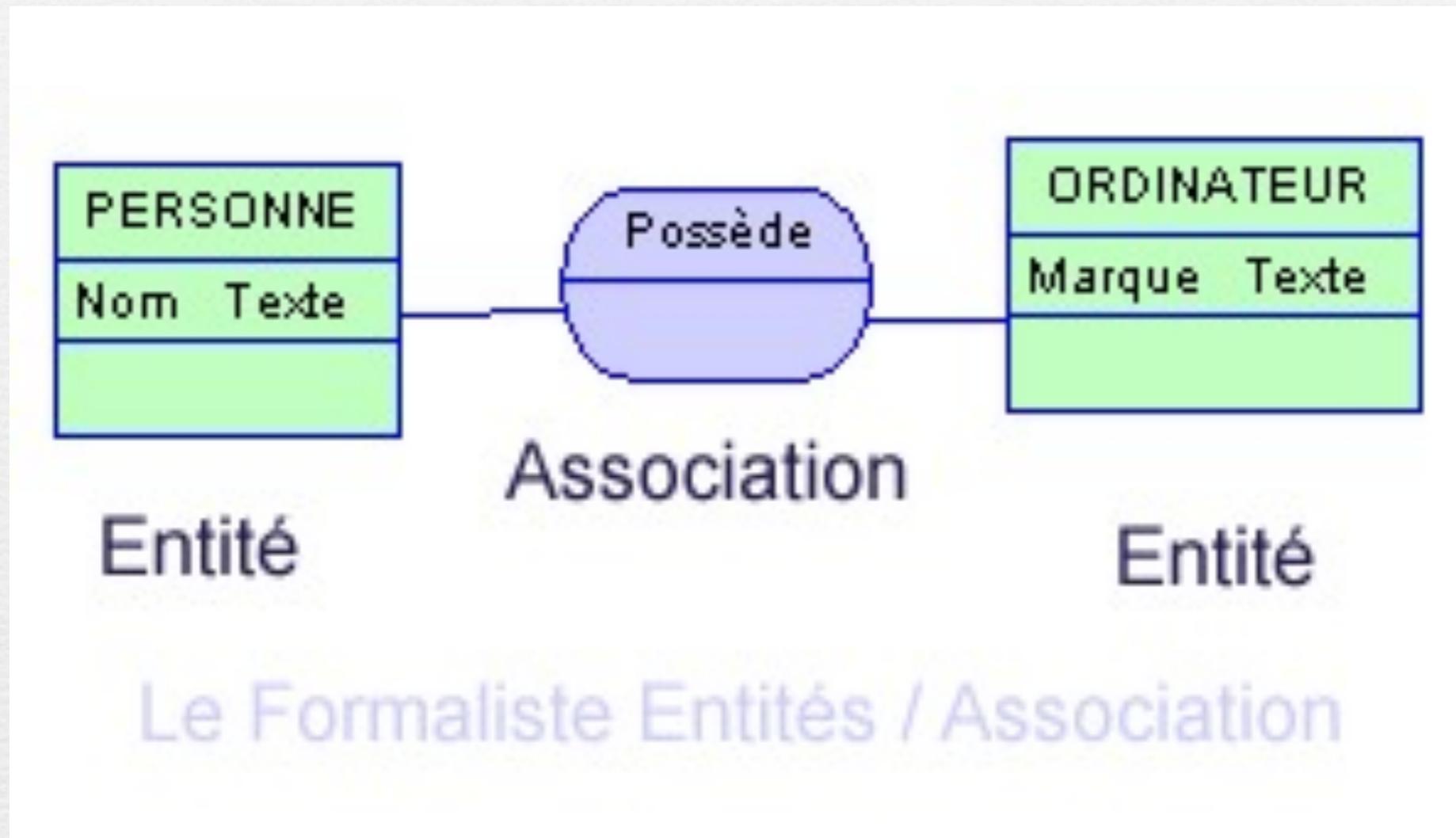


```

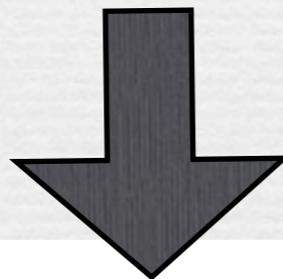
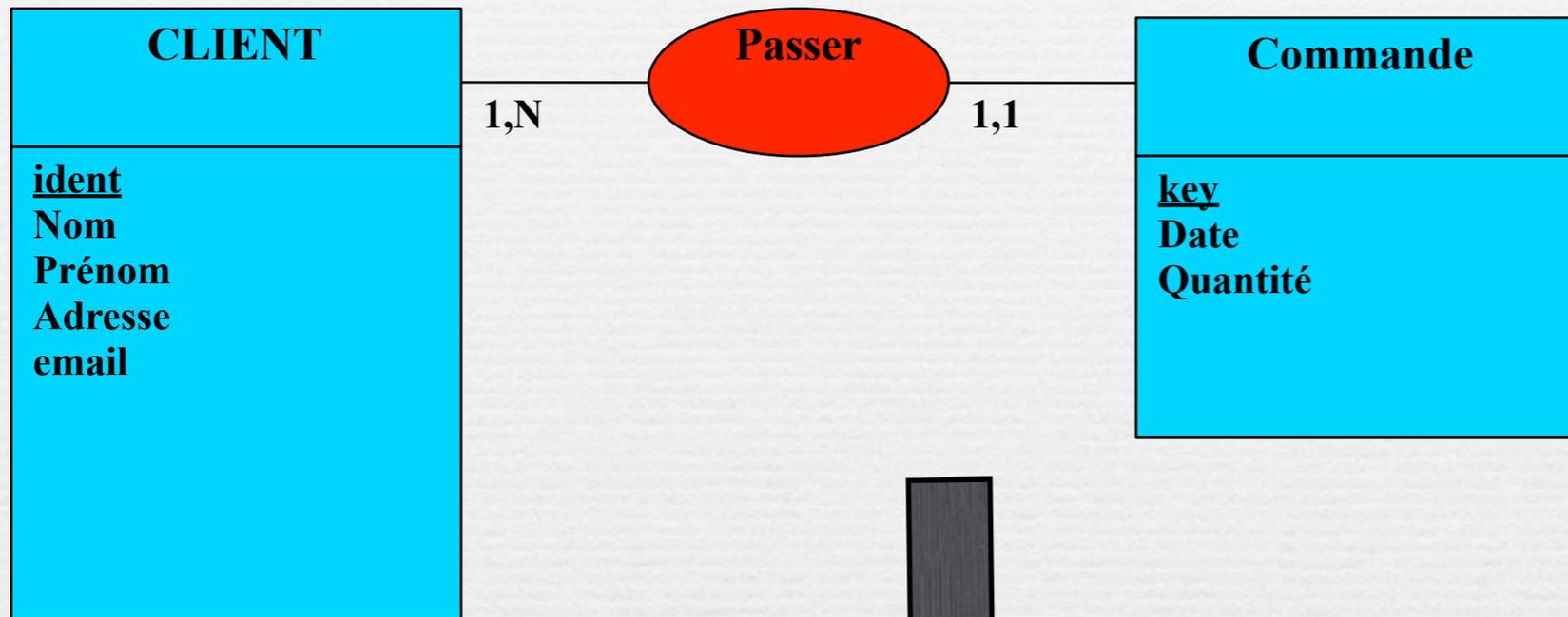
CREATE TABLE Cours_Etudiant (Courstitre varchar(11) NOT NULL, Etudiantnumero int(11) NOT NULL, PRIMARY KEY
(Courstitre, Etudiantnumero));
CREATE TABLE Etudiant (numero int(11) NOT NULL AUTO_INCREMENT, Personnenom int(10) NOT NULL, PRIMARY
KEY (numero));
CREATE TABLE Cours (titre varchar(11) NOT NULL, EnseignantnumPoste int(11), PRIMARY KEY (titre));
ALTER TABLE Cours ADD INDEX FK Cours908452 (EnseignantnumPoste), ADD CONSTRAINT FK Cours908452 FOREIGN
KEY (EnseignantnumPoste) REFERENCES Enseignant (numPoste);
ALTER TABLE Etudiant ADD INDEX FKEtudiant884191 (Personnenom), ADD CONSTRAINT FKEtudiant884191 FOREIGN
KEY (Personnenom) REFERENCES Personne (nom);
ALTER TABLE Enseignant ADD INDEX FKEnseignant439547 (Personnenom), ADD CONSTRAINT FKEnseignant439547
FOREIGN KEY (Personnenom) REFERENCES Personne (nom);
ALTER TABLE Cours_Etudiant ADD INDEX FK Cours_Etud35330 (Etudiantnumero), ADD CONSTRAINT FK Cours_Etud35330
FOREIGN KEY (Etudiantnumero) REFERENCES Etudiant (numero);
ALTER TABLE Cours_Etudiant ADD INDEX FK Cours_Etud846124 (Courstitre), ADD CONSTRAINT FK Cours_Etud846124
FOREIGN KEY (Courstitre) REFERENCES Cours (titre);
    
```

D'un modèle
entité-relation à un
modèle à classe en
UML

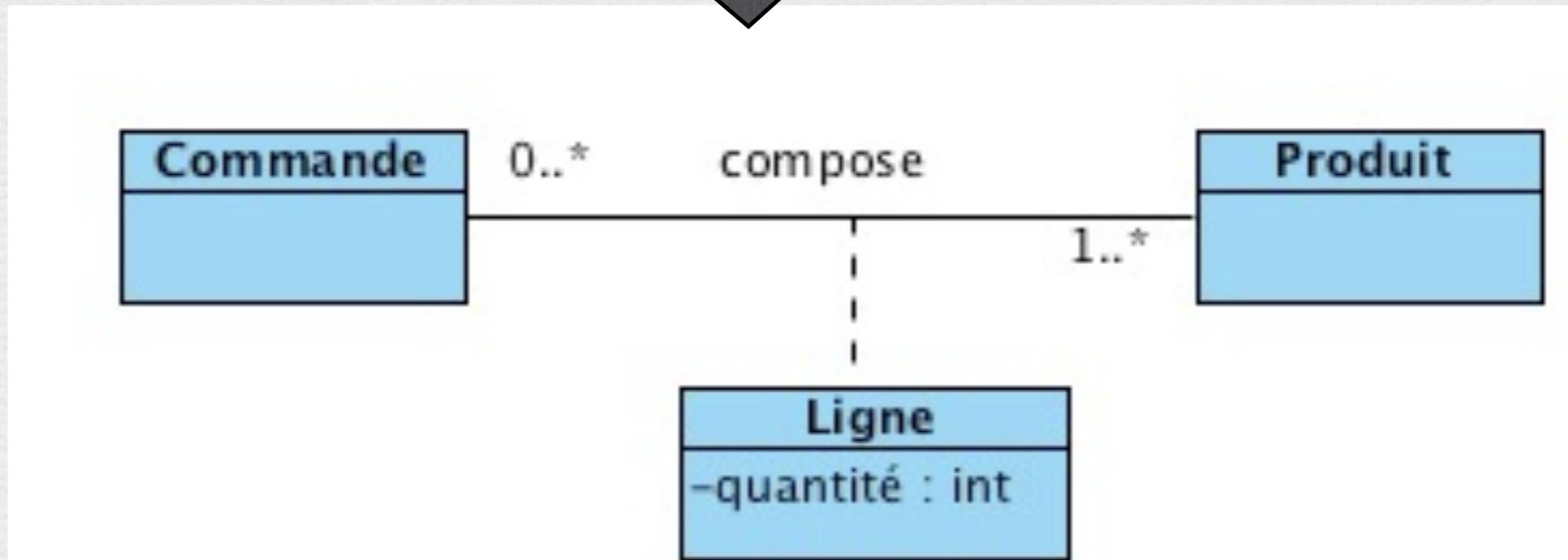
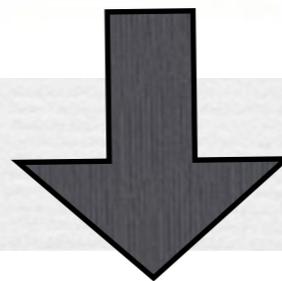
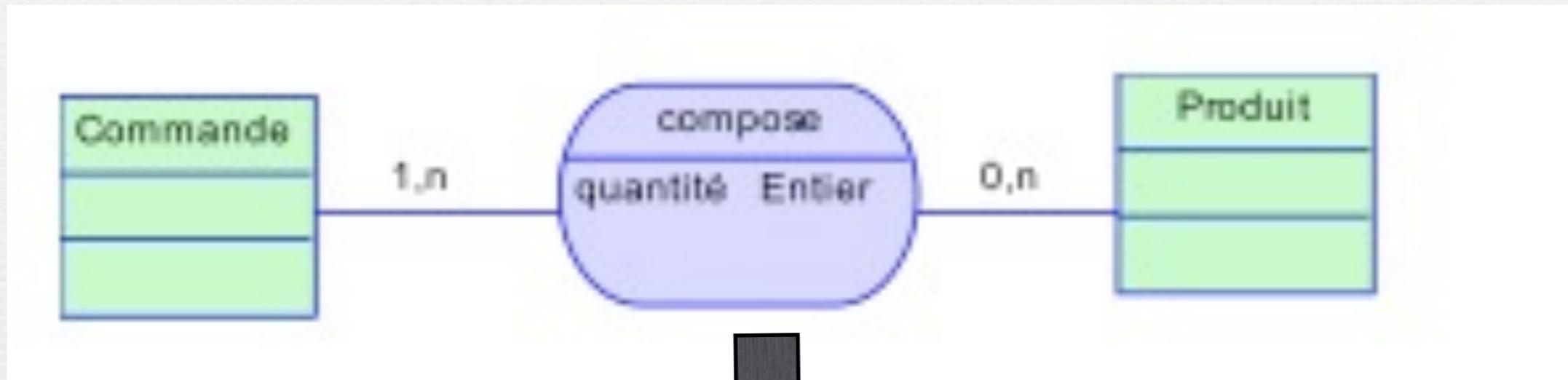
Modèles Entité - Association



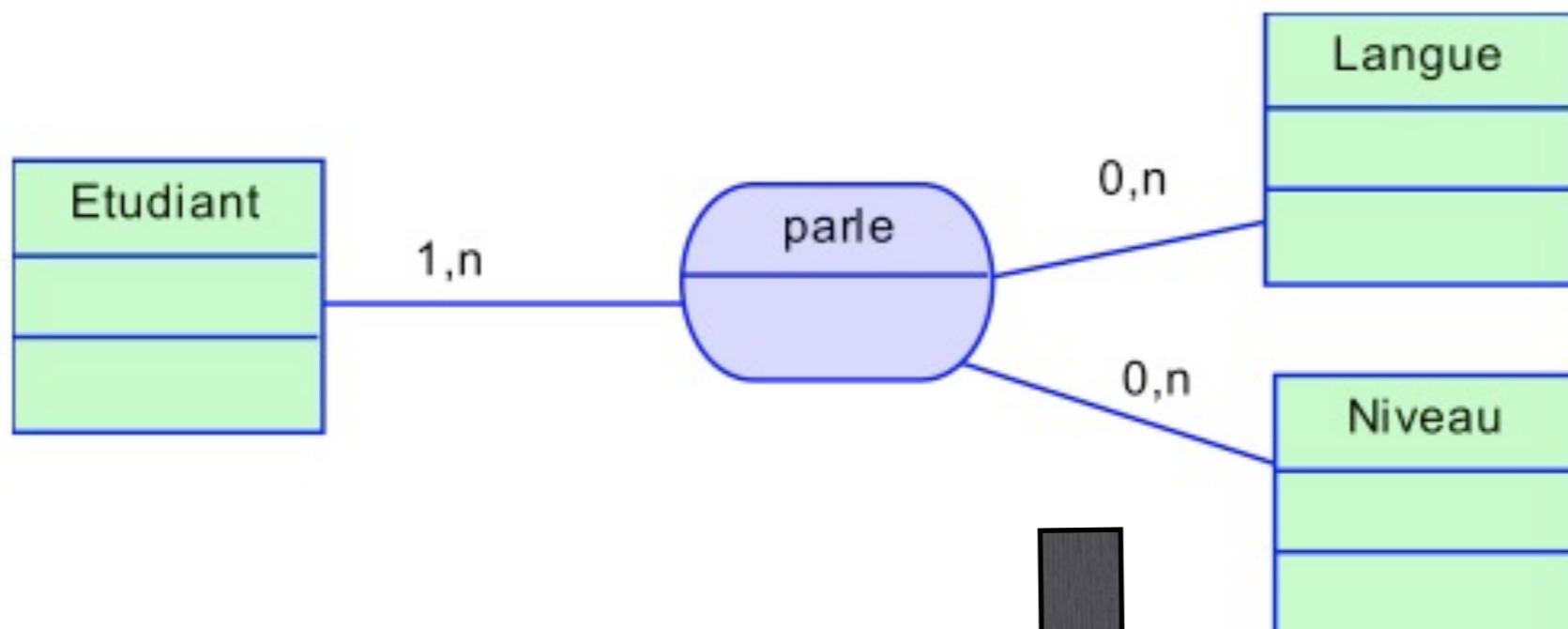
D'un modèle entité-association à un modèle de classe en UML



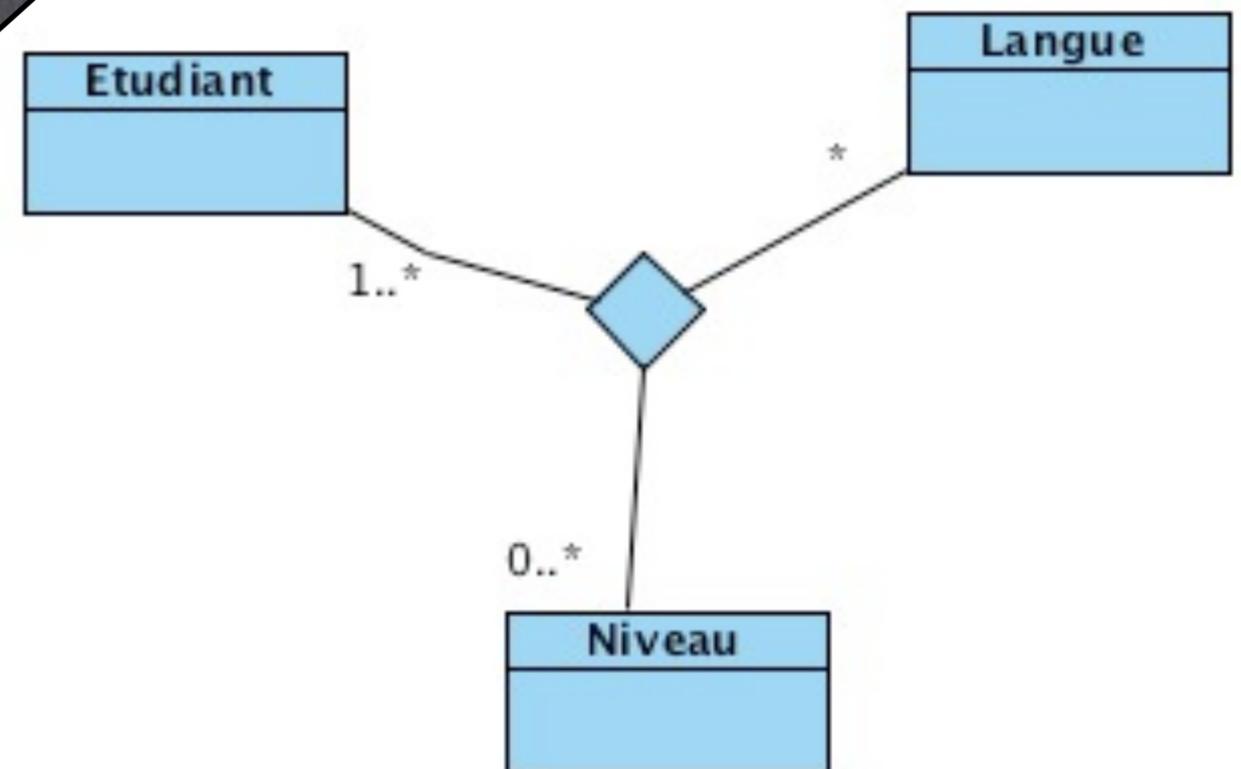
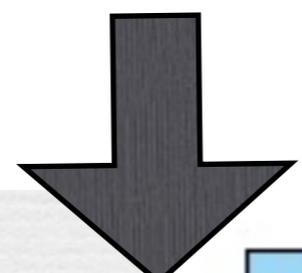
D'un modèle entité-association à un modèle de classe en UML



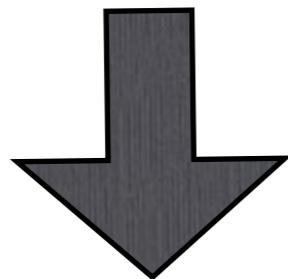
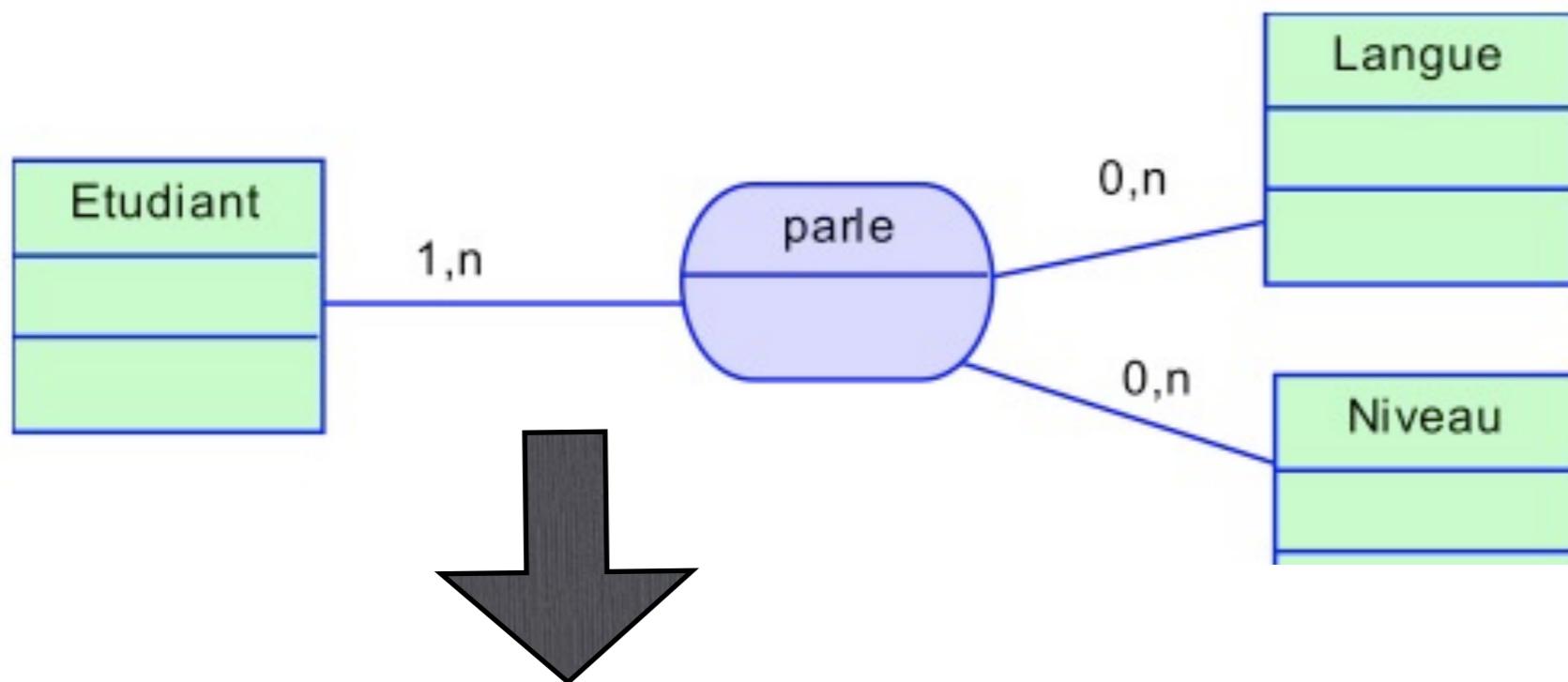
D'un modèle entité-association à un modèle de classe en UML



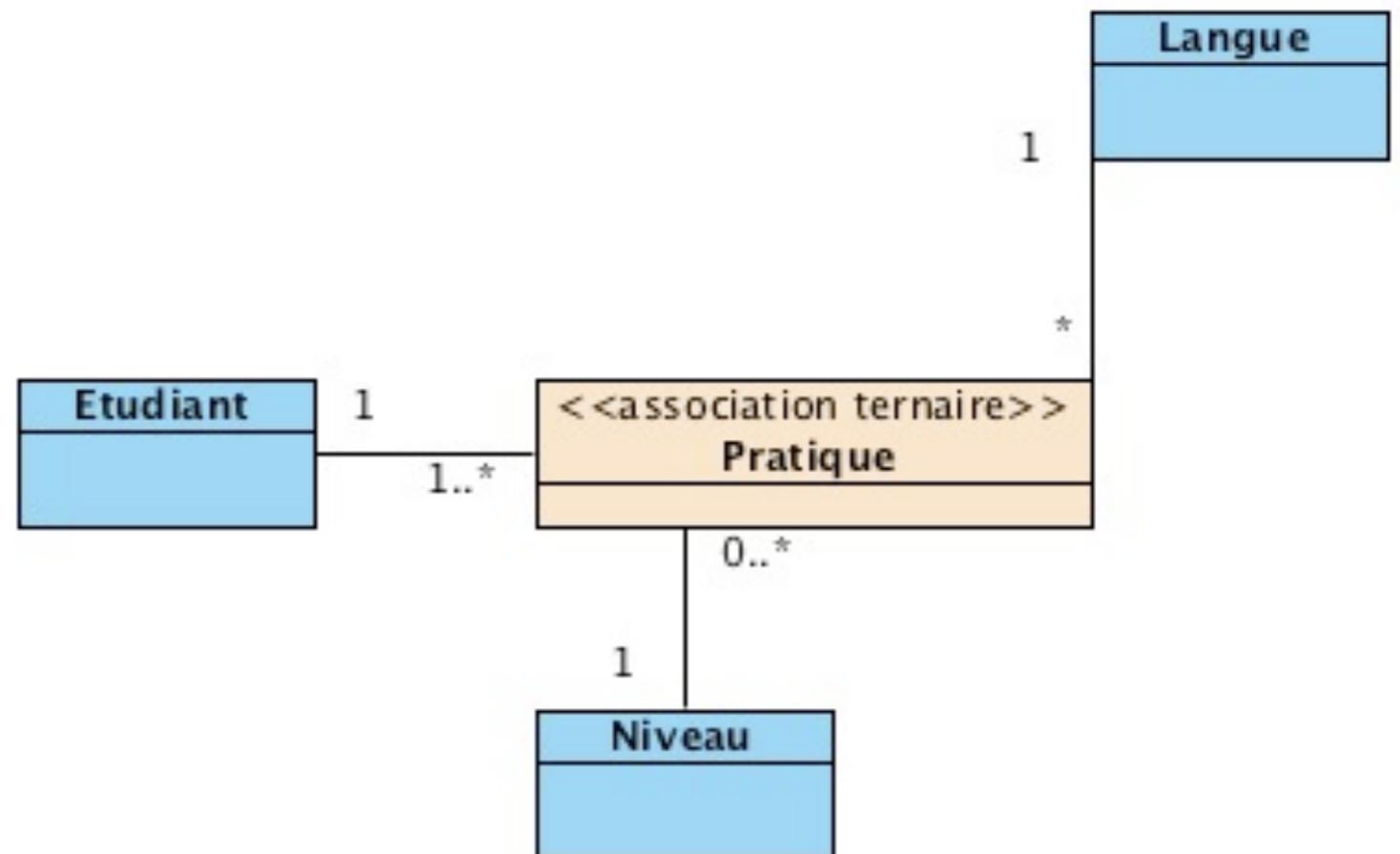
Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.



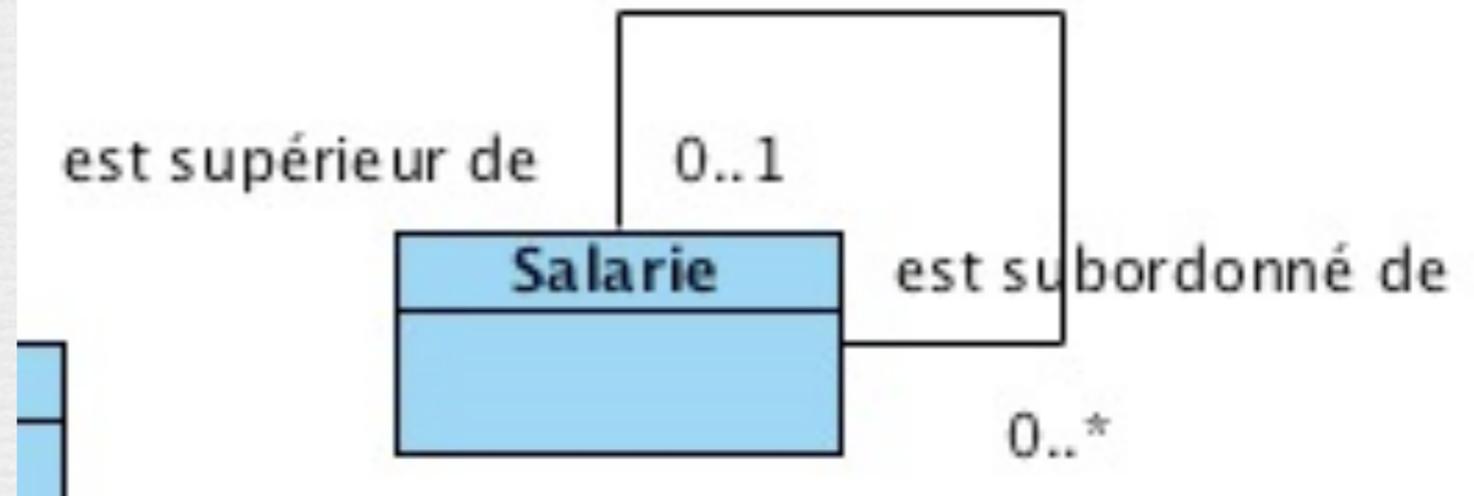
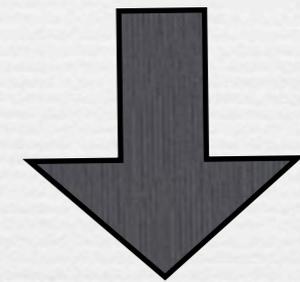
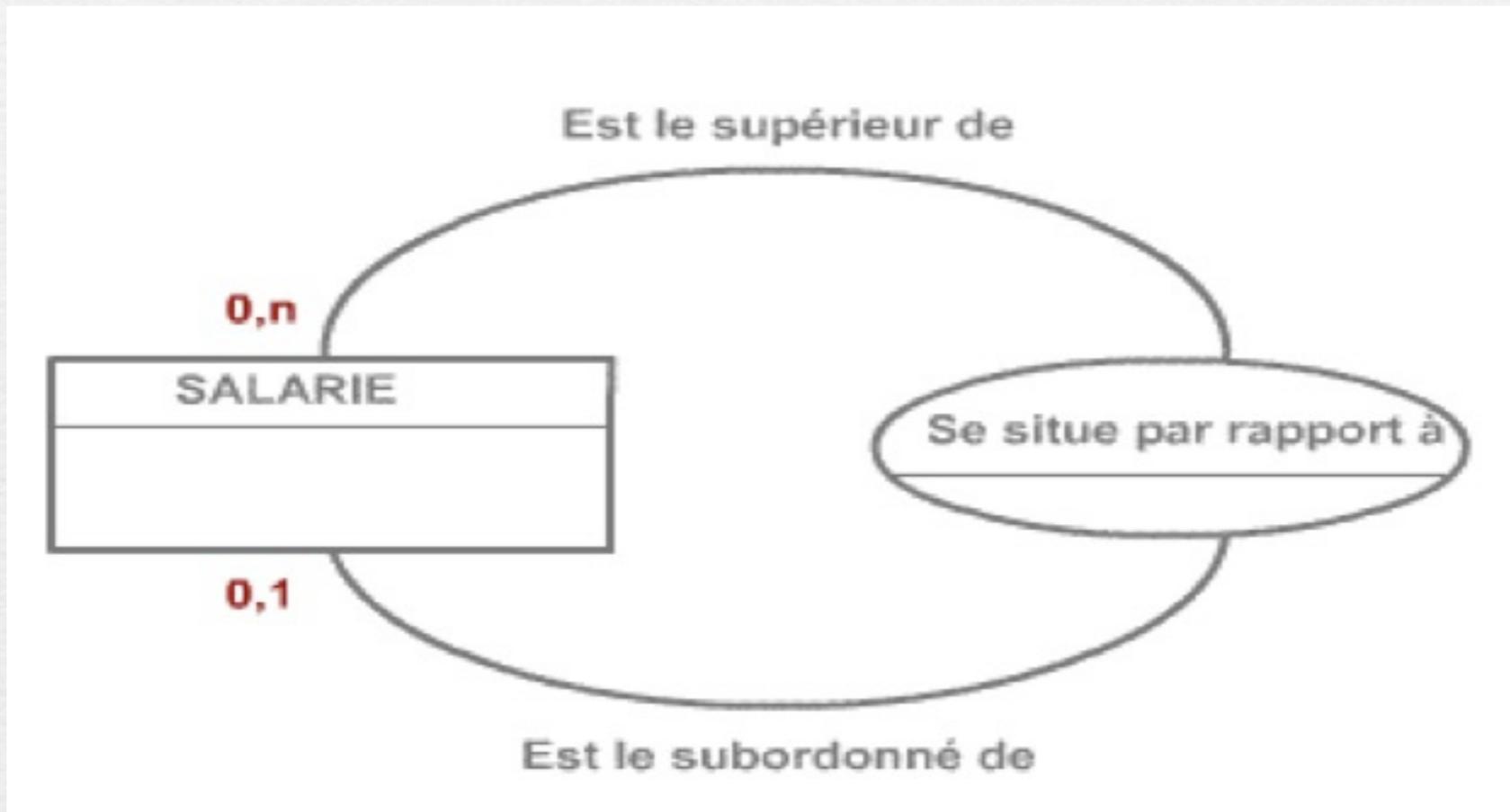
D'un modèle entité-association à un modèle de classe en UML

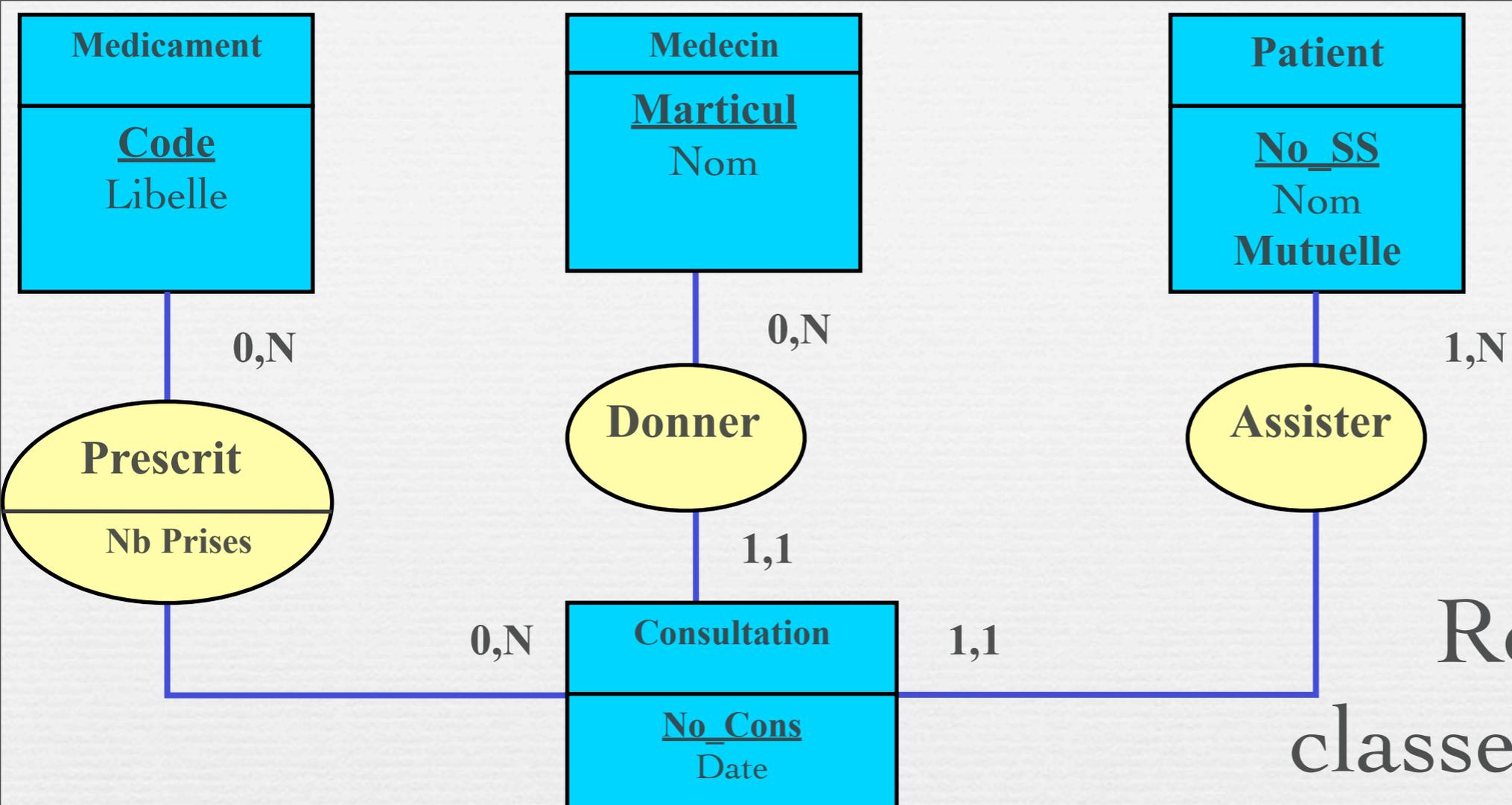


Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.

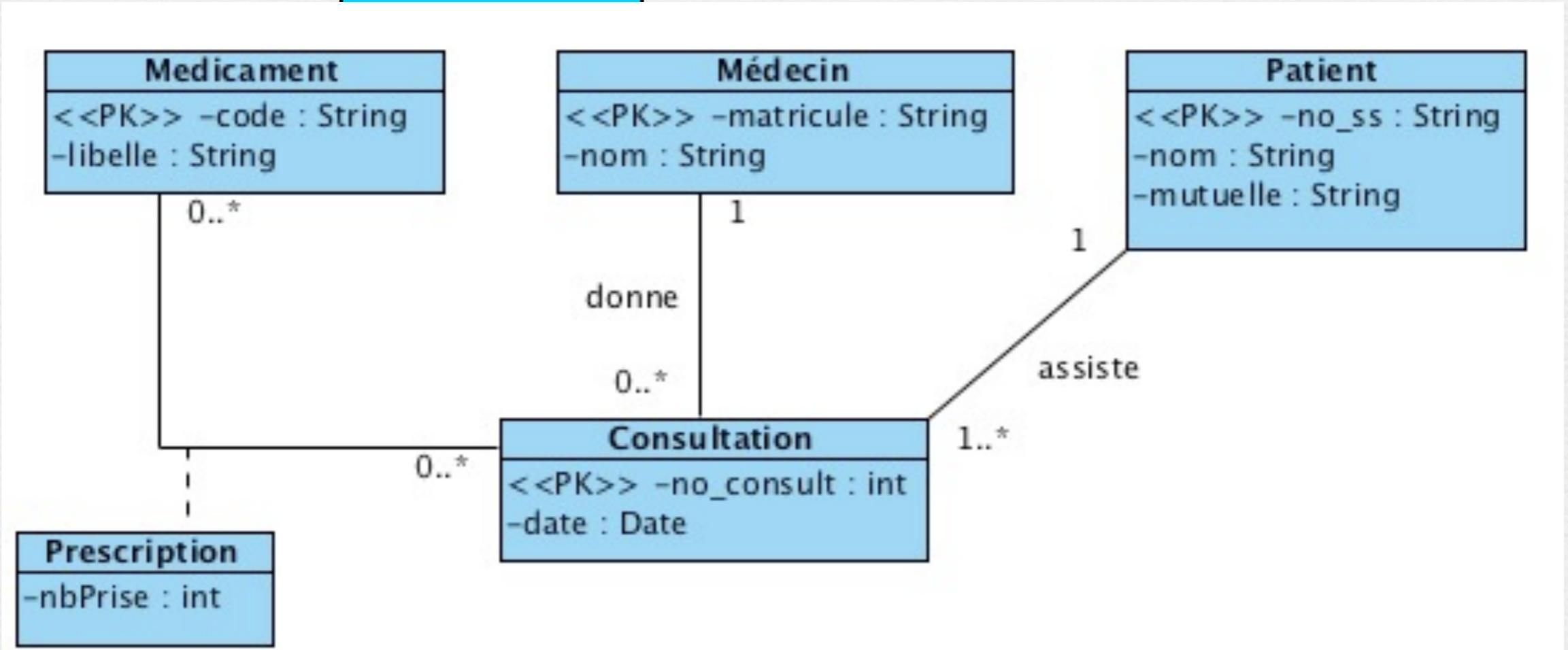


D'un modèle entité-association à un modèle de classe en UML

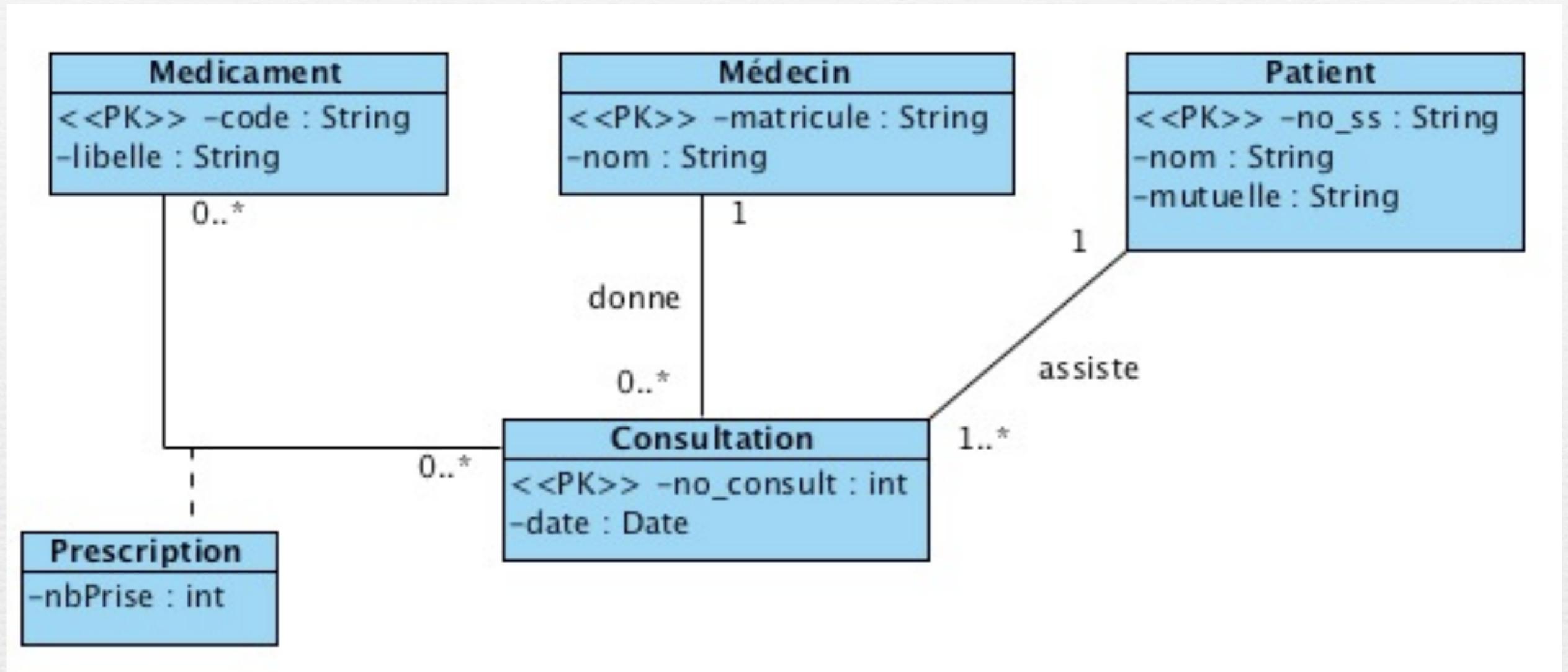




Relations et classe d'association



Savoir lire



1. Un patient peut-il effectuer plusieurs visites?
2. Un médecin peut-il recevoir plusieurs patients dans la même consultation?
3. Peut-on prescrire plusieurs médicaments dans une même consultation?
4. Deux médecins différents peuvent-ils prescrire le même médicament.

Oui
Non
Oui
Oui

Exercice 3 " LabDB "

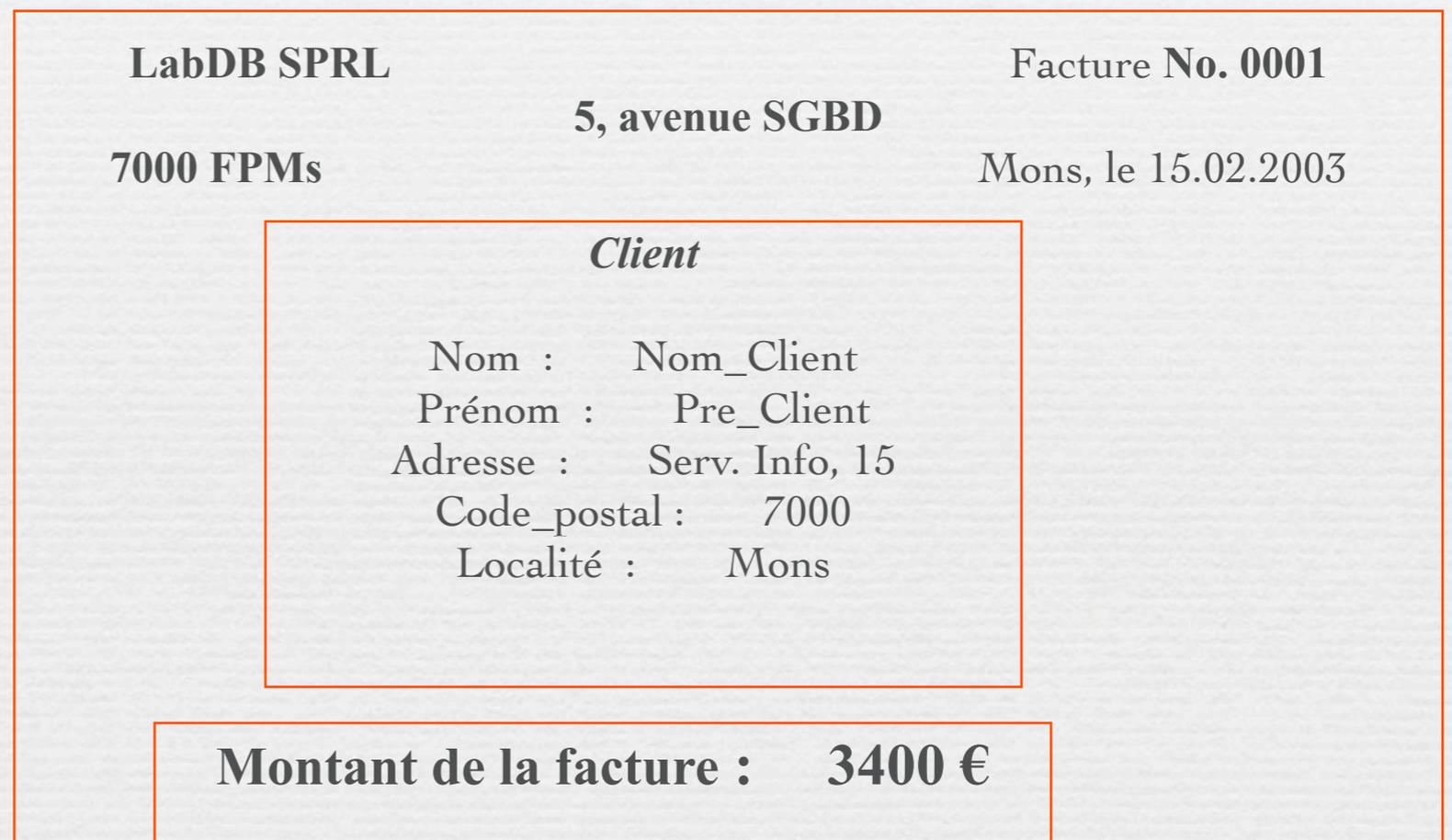
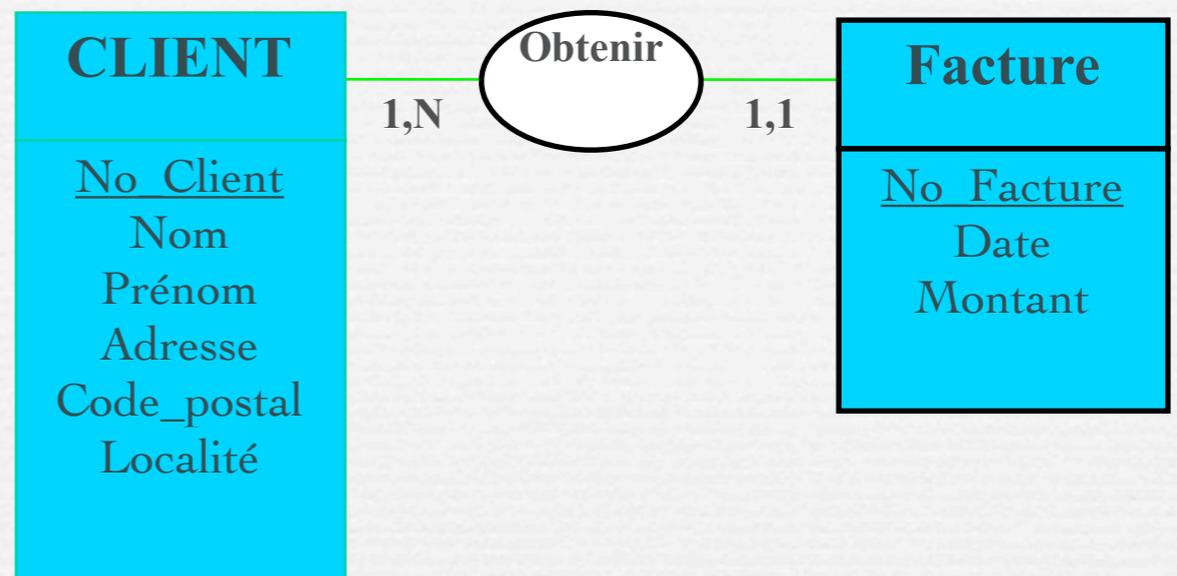
PARTIE 1

No Client en + propriété **artificielle** définit comme identifiant. Sinon il faut définir un identifiant composé de plusieurs propriétés.

La société "LabDB" désire informatiser son système de facturation. Les factures devraient se présenter de la façon suivante:

Modéliser les données du système d'information, sachant que:

- Un client peut bien sûr recevoir plusieurs factures, mais il est uniquement considéré comme tel à partir du moment où il reçoit sa première facture.
- Une facture concerne un et un seul client.



PARTIE 2

Le responsable de la facturation de la société désire rendre les factures plus informatives. Comme un client peut acheter plusieurs articles différents en même temps, la facture devrait indiquer pour chaque article le numéro, un libellé, le prix unitaire, la quantité vendue et le prix total pour ce type d'article.

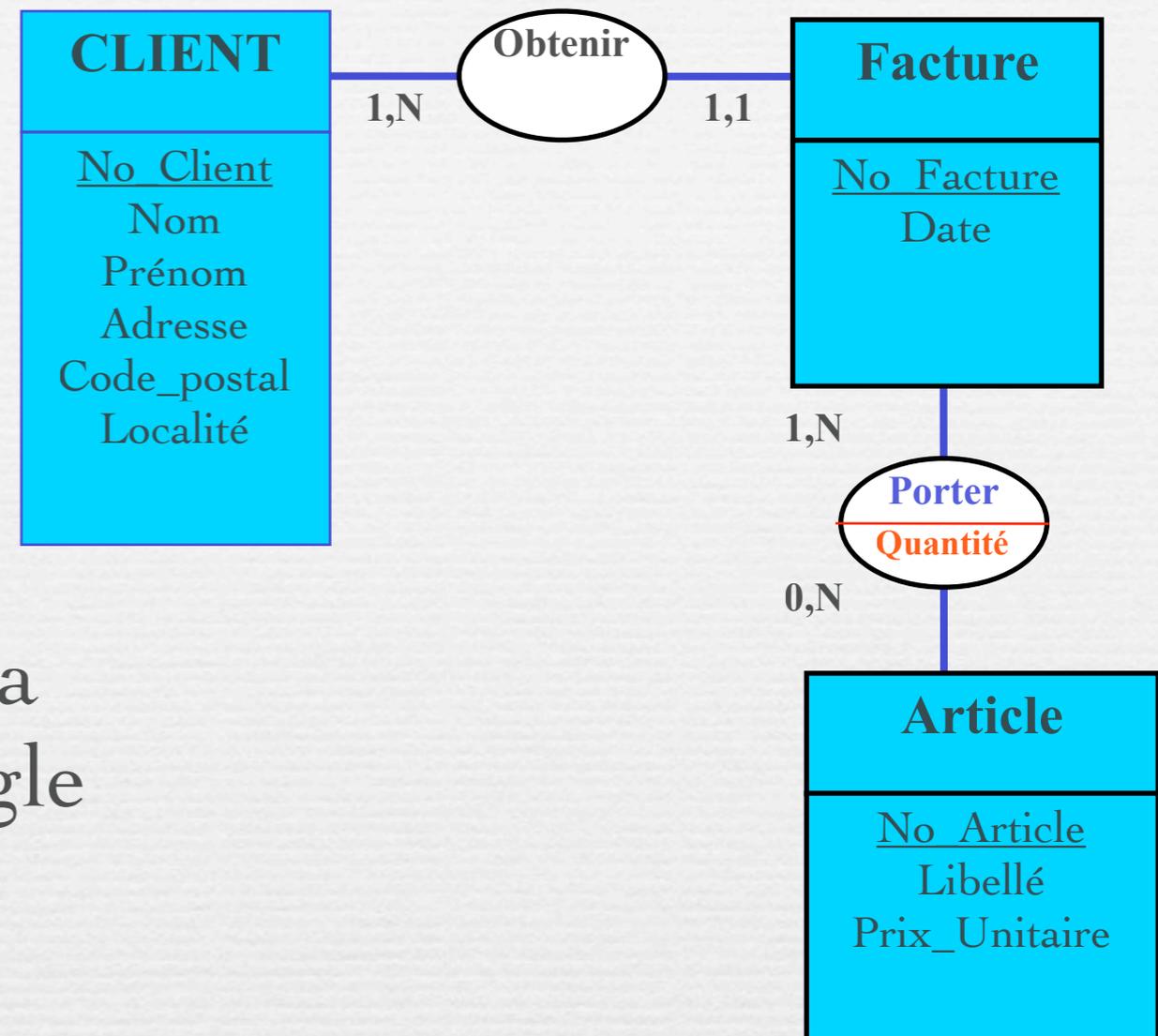
la facture devrait avoir:

Voici l'aspect que la facture devrait avoir:

Tous les articles disponibles sont stockés (p.ex. No=233 Libellé="Analyse" PU=1000 €). Même si un article n'est pas encore considéré par une facture, il existe dans le système d'information.

LabDB SPRL	Facture No. 0002																				
5, avenue SGBD																					
7000 FPMs	Mons, le 15.02.2003																				
<i>Client</i>																					
Nom : Nom_Client Prénom : Pre_Client																					
Adresse : Serv. Info, 15																					
Code_postal : 7000																					
Localité : Mons																					
<table border="1"><thead><tr><th>No. Article</th><th>Libellé</th><th>Prix unitaire</th><th>Quantité</th><th>Prix</th></tr></thead><tbody><tr><td>233</td><td>Analyse</td><td>1000 €</td><td>1</td><td>1000 €</td></tr><tr><td>025</td><td>MCD</td><td>700 €</td><td>2</td><td>1400 €</td></tr><tr><td>142</td><td>MLD</td><td>1000 €</td><td>1</td><td>1000 €</td></tr></tbody></table>		No. Article	Libellé	Prix unitaire	Quantité	Prix	233	Analyse	1000 €	1	1000 €	025	MCD	700 €	2	1400 €	142	MLD	1000 €	1	1000 €
No. Article	Libellé	Prix unitaire	Quantité	Prix																	
233	Analyse	1000 €	1	1000 €																	
025	MCD	700 €	2	1400 €																	
142	MLD	1000 €	1	1000 €																	
Montant total de la facture : 3400 €																					

PARTIE 2



Remarque:

L'entité Facture ne contient plus la propriété Montant. Il existe une règle générale de conception qui dit:

Aucune propriété qui peut être calculée à partir d'autres propriétés existantes ne devra être stockée dans le Modèle.

FIN

Pour cette année ...

Bibliographie

 Mohammed BENJELLOUN, Les Bases de données relationnelles, Faculté Polytechnique de Mons, [Mohammed.Benjelloun@fpms.ac.be](mailto: Mohammed.Benjelloun@fpms.ac.be)