

SCR vs DEV vs SGBD

modélisation du
logiciel

DEV 2.1/SGBD

programmation du
logiciel

DEV 1.1/SGBD

optimisation du logiciel et
couche réseau/système

SCR

Plan du cours

- Introduction
- Modélisation conceptuelle des données
- Passage d'un modèle conceptuel de données à un modèle logique des données

Exemple

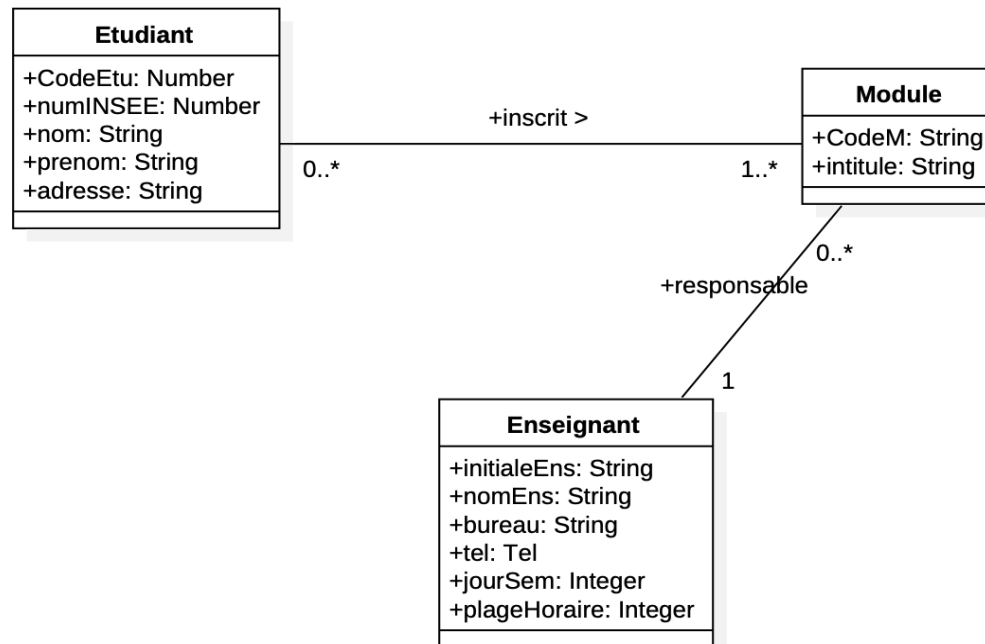
1.1 Inscription des étudiants

Lors de son inscription en début d'année scolaire, chaque étudiant remplit une fiche sur laquelle il indique certains renseignements comme son numéro d'identification nationale (*ninsee*), ses nom et prénom (*nom*, *prenom*), son adresse (*adresse*) et la liste des unités de valeurs (UV) qu'il s'engage à suivre (8 au plus sur les 15 possibles). Un code lui est automatiquement attribué (*codetu*).

Une UV est caractérisée par un code (*codeuv*) et un intitulé (*intuv*). Par exemple le code *UV3* identifie *Électronique numérique*. Chaque UV est placée sous la responsabilité d'un enseignant identifié par ses initiales (*initens*) et caractérisé par un nom (*nomens*), un numéro de bureau (*bureauens*) et un numéro de téléphone (*telens*). Cet enseignant se rend disponible un jour de la semaine (*jsem*) et durant une plage horaire précise (*hrens*) afin de donner tout renseignement concernant les UV qu'il dirige.

- Que faut-il modifier pour qu'un enseignant se rende disponible à différents moments (un seul créneau par UV qu'il dirige) ?
- Que faut-il modifier pour que différents créneaux soient disponibles par UV qu'il dirige ?

Diagramme de classes



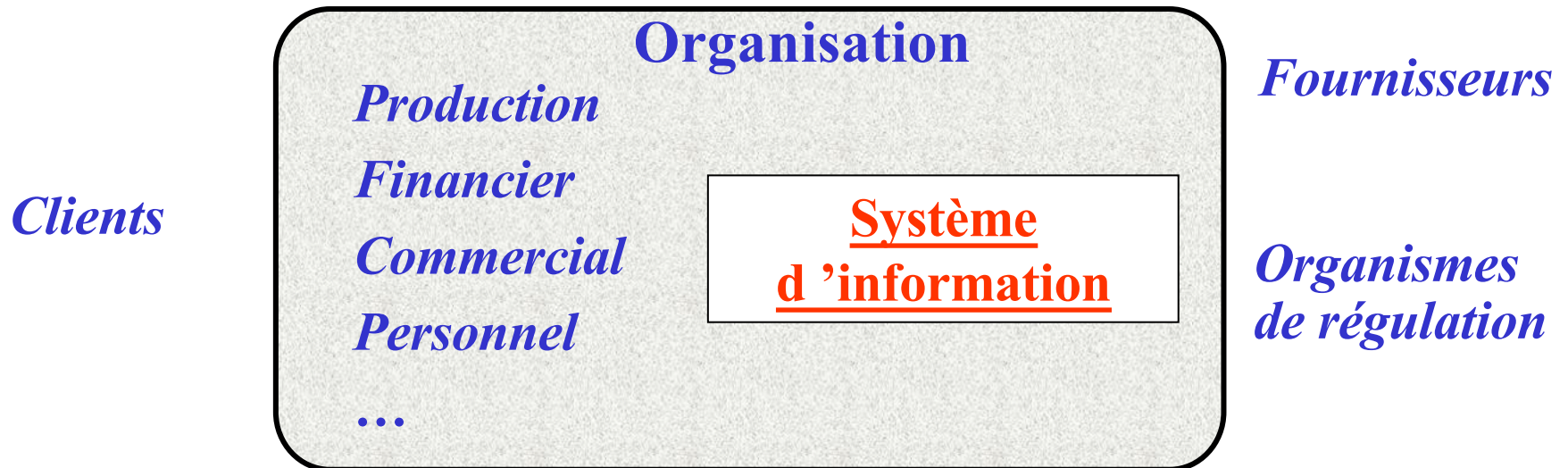
Code SQL

```
CREATE TABLE SCHEMA1.ENSEIGNANTS
(
  ENSEIGNANTS_ID NUMBER NOT NULL ,
  INITIALES_ENS  VARCHAR2(20) ,
  NOM            VARCHAR2(20) ,
  BUREAU        VARCHAR2(20) ,
  TÉLÉPHONE     VARCHAR2(20) ,
  JOUR_SEM      VARCHAR2(20) ,
  PLAGE_HORAIRE VARCHAR2(20) ,
  CONSTRAINT ENSEIGNANTS_PK PRIMARY KEY ( ENSEIGNANTS_ID ) ENABLE
);
CREATE TABLE SCHEMA1.INSCRITS
(
  INSCRITS_ID NUMBER NOT NULL ,
  ETUDIANTS_ID NUMBER ,
  UVS_ID      NUMBER ,
  CONSTRAINT INSCRITS_PK PRIMARY KEY ( INSCRITS_ID ) ENABLE
);
CREATE TABLE SCHEMA1.UVS
(
  UVS_ID      NUMBER NOT NULL ,
  CODE_UV     VARCHAR2(20) ,
  INTITULÉUV VARCHAR2(20) ,
  END_UV      NUMBER NOT NULL ,
  CONSTRAINT UVS_PK PRIMARY KEY ( UVS_ID ) ENABLE
);
CREATE TABLE SCHEMA1.ETUDIANTS
(
  ETUDIANTS_ID NUMBER NOT NULL ,
  CODE_ETU     VARCHAR2(20) ,
  NUM_INSEE    VARCHAR2(20) ,
  NOM          VARCHAR2(20) ,
  PRÉNOM      VARCHAR2(20) ,
  ADRESSE     VARCHAR2(20) ,
  CONSTRAINT ETUDIANTS_PK PRIMARY KEY ( ETUDIANTS_ID ) ENABLE
);
ALTER TABLE SCHEMA1.INSCRITS ADD CONSTRAINT INSCRIT_ETUDIANT FOREIGN KEY
(
  ETUDIANTS_ID
)
REFERENCES SCHEMA1.ETUDIANTS
(
  ETUDIANTS_ID
)
ON
```

1 - Introduction

1.1 Définition du Système d'Information (S.I.)

Environnement de l'organisation



Un système d'information est un ensemble organisé de ressources (matériel, logiciel, personnel, données et procédures) permettant d'acquérir, traiter, stocker, communiquer, etc., des informations nécessaires à la réalisation des processus de l'organisation

- **L'ORGANISATION ET SES PROCESSUS**
les conditions de fonctionnement du futur système dans l'organisation

- **LE SYSTEME D'INFORMATION AU SERVICE D'UNE ORGANISATION**
quels sont les éléments (informations) à représenter ?

- **LES TECHNOLOGIES DE L'INFORMATION SUPPORTENT PHYSIQUEMENT LE S.I**
le choix des technologies particulières et la définition de leur usage.

- **Champ d'application très vaste dû à la variété des systèmes :**

*Informatique de gestion, Informatique embarquée,
Informatique temps réel, Bureautique, Imagerie, ..etc.*

- **Complexité intrinsèque des Systèmes d'Information :**

Utilisation de nombreuses techniques pour la modélisation des systèmes (Théorie des Bases de données , les Langages de programmation, les automates, ...)



Besoin de méthodes

Des méthodes de conception de S.I. qui combinent:

- **des modèles de représentation de données (et de traitement)**
- **(des modèles de processus de conception du S.I.)**
- **des outils d'aide à la conception**

La méthode permet d'obtenir un modèle complet du futur système d'information qu'on peut valider avec les clients. Ce modèle sera ensuite mis en œuvre par le service de développement.

1) les modèles de représentation de données

Un modèle est une image de la réalité qui facilite la communication, il est établi pour répondre à un besoin particulier.

Exemples

- Le modèle Entité Association
- Les modèles de Merise (M.C.D, M.C.T, ...)
- Le modèle Objet UML (langage semi formel)
- Le langage B, le langage Z (langage formel)

**Le résultat de la modélisation est un schéma appelé
modèle conceptuel (ou schéma conceptuel)**

Exemple : Comment représenter le fait que les étudiants doivent faire un stage

Liste des étudiants
de 2ème année

Aurélie

Pierre

Meriem

Steve

Racem

Liste des propositions
de stage

L'Oréal

SNCF S1

SNCF S2

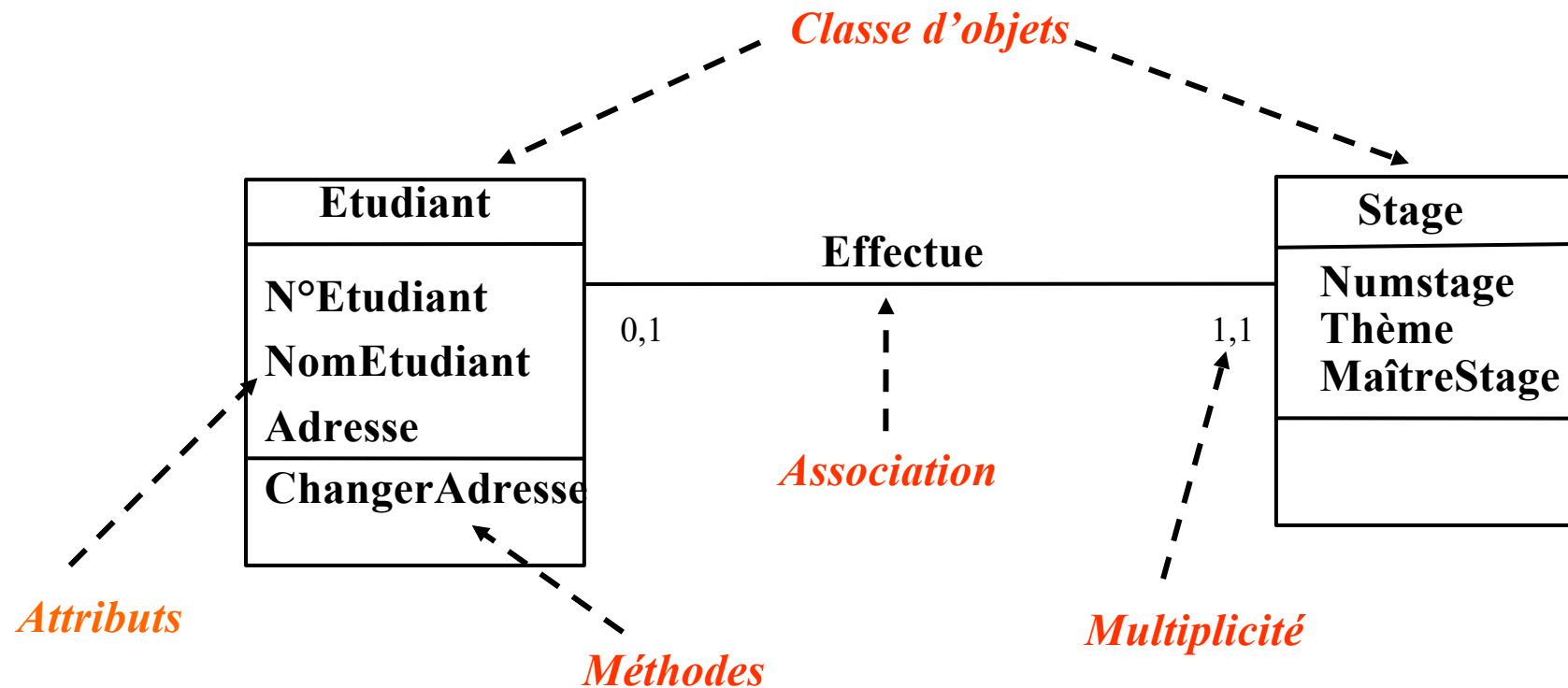
Avenade

Mahi-Mahi

SFR

Radan

Une solution avec le modèle Objet UML



2) Les processus de conception du S.I.

Les grandes phases des approches classiques de développement des systèmes d'information sont les suivantes :

- 1. Analyse des besoins des utilisateurs du système**
- 2. Modélisation des données et des traitements**
- 3. Conception de la base de données et des traitements**
- 4. Codage, Tests et intégration des modules**
- 5. Tests utilisateurs**
- 6. Mise en exploitation**

Plusieurs types de modèles de processus : modèle en cascade, modèle en V, processus unifié, etc.

3) Les outils d'aide à la conception

sont des environnements de conception et de développement de systèmes. Ils permettent la génération de modèles et automatisent en partie les étapes de production du logiciel.

Ces outils sont appelés **Atelier de Génie Logiciel (AGL, CASE en anglais :Computer Aided System Engineering)**).

Exemple

Win'Design, **StarUML**, RationalRose, Objecteering, JDeveloper, Umbrello, etc...

2 - Modélisation Conceptuelle des données

2.1 Introduction à la modélisation

2.2 Concepts et Règles de modélisation **Objet**

2.3 La démarche de construction

2.4 Notions complémentaires

2.5 Conclusion

2. 1 Introduction

- Il s'agit d'exprimer **l'ensemble des informations** que l'on veut prendre en compte dans le système d'information
- La solution est basée sur un **formalisme de représentation**
 - *guider le raisonnement du concepteur*
 - *obliger à respecter des normes*
 - *utiliser un langage commun*
- Le résultat est un **modèle de classes d'objets**
clair, cohérent, complet, fidèle et normalisé
- Ce résultat **est indépendant** de considérations techniques ou organisationnelles

2. 1 Introduction

Une bibliothèque

les Livres avec leur référence, leur titre et leur auteur

Les Abonnés avec leur nom, leur date de naissance et leur adresse

Les *emprunts* de livre par les abonnés

L456, Les misérables, V. Hugo

R589, Germinal, E. Zola

.....

L'organisation

**Modèle de
classes d'objet**

**Base de
données**

2.2 - Modélisation Conceptuelle des données

Introduction à la modélisation

2.2 . Concepts et règles de modélisation Objet

2.2.1 Les concepts de classe et d'objet

2.2.2 Le concept d'association

2.2.3 Les contraintes d'intégrité

La démarche de construction

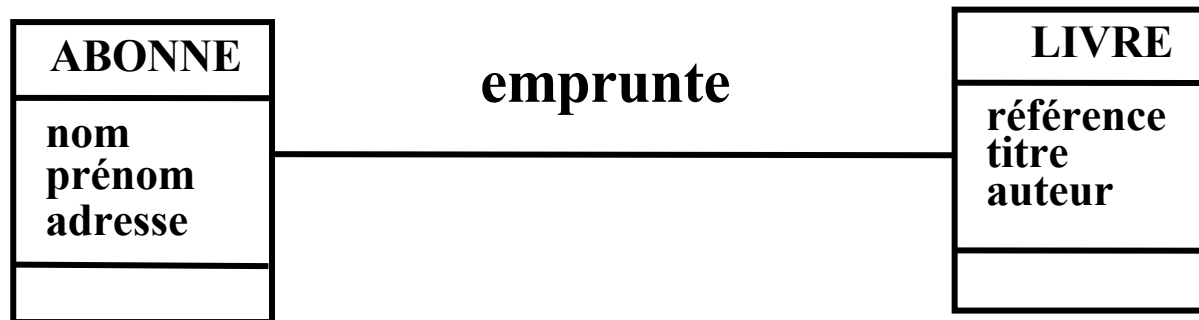
Notions complémentaires

2.2.1 Les concepts de modélisation O. O.

Les principaux concepts du modèle objet sont :

- **Objet, Classe, Attribut,**
- **Association**
- **Contraintes d'Intégrité**

L'utilisation d'un formalisme **graphique**



1) Le concept d'objet :

Définition : Abstraction ou entité ayant un sens propre (une valeur) dans le domaine ou pour une application donnée.

ex : le livre *les misérables*, l'abonné *Dupont*, la voiture *Peugeot 206*

Un objet présente trois caractéristiques : **un état (des valeurs)**, **un comportement** et **une identité**

Objet = État + Comportement + Identité

- **2 types d'objets** : Objets concrets (table, chaise, avion, voiture...) et Objets abstraits (compte en banque, commande...)

2.2.1 Les concepts d'Objet et de Classe

a) L'état regroupe les valeurs de tous les attributs d'un objet à un instant donné

- un attribut est une information qualifiant l'objet
- Chaque attribut peut prendre une valeur dans un domaine défini (le type similaire au langage de programmation)

*L'exemple ci après montre 2 objets abonnés décrit par les attributs suivants:
le nom, le prénom, l'adresse, le téléphone*

Un abonnéA

nom : *Leblanc*
prénom : *Eric*
Adresse : *20 rue Royale*
Téléphone : *0617889955*

Un abonnéB

nom : *Dupont*
prénom : *Martin*
Adresse : *45 rue Grande*
Téléphone : *0617546789*

2.2.1 Les concepts d'Objet et de Classe

b) Le comportement représente l'évolution de l'objet au cours du temps:

- Certaines valeurs d'attribut vont évoluer (valeur dynamique)
- D'autres vont être constantes (valeur statique)

*Changement du
N° téléphone*

Un abonnéA

nom : *Leblanc*
prénom : *Eric*
Adresse : *20 rue Royale*
Téléphone : *0617889955*

Un abonnéA

nom : *Leblanc*
prénom : *Eric*
Adresse : *20 rue Royale*
Téléphone : *0677891915*

2.2.1 Les concepts d'Objet et de Classe

- c) Un objet possède **une identité** qui caractérise son existence
- l'identité permet de distinguer chaque objet de façon non ambiguë
 - Un attribut est utilisé pour représenter l'identifiant permettant de distinguer les objets

Exemple

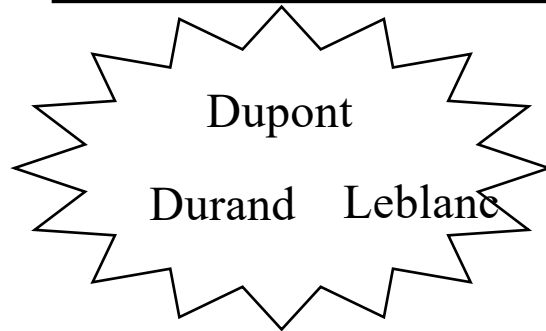
- Les voitures possèdent **un numéro d'immatriculation** qui permet de les différencier.
- **Le numéro de sécurité sociale** permet de différencier les employés d'une entreprise (même ceux qui sont homonymes).

2.2.1 Les concepts d'Objet et de Classe

2) Le concept de classe

Définition : représentation d'un ensemble d'objets de même nature, concrets ou abstraits et présentant un intérêt (c'est le type, le genre des objets)

La bibliothèque réelle



**Des abonnés de la bibliothèque
(des objets)**

Abstraire



Le schéma conceptuel



La classe d'objet

- **Niveau instance** : représente la valeur de l'information (les données)
- **Niveau type** : représente l'information (un sens précis)

2.2.1 Les concepts d'Objet et de Classe

3) Le concept d'attribut

Les attributs caractérisent des objets d'une classe

Nom de classe
Nom d'attribut: type = valeur initiale

ABONNE
Nom : chaîne
Adresse : chaîne
dateNaissance : date

La classe et ses propriétés

Objet1: *Dupont, 4 rue Duchemin 75012 Paris..., 27/06/78*

Objet2: *Durand, 2 rue Victor Hugo 75012 Paris, 21/09/56*

Objet3: *Leblanc, ...*

Objet4: *Martin, ...*

...-

Des objets de la classe

ABONNE

Chaque nom d'attribut est unique dans une classe.

4) Le concept d'association

Définition : relation sémantique entre classes.

C'est une abstraction de liens qui existent entre les objets des classes associées

*Exemple : Dupont **habitant** 4 rue Duchemin 75012 Paris **né le** 27/06/78 **a emprunté** "Les misérables" **de** V. Hugo*

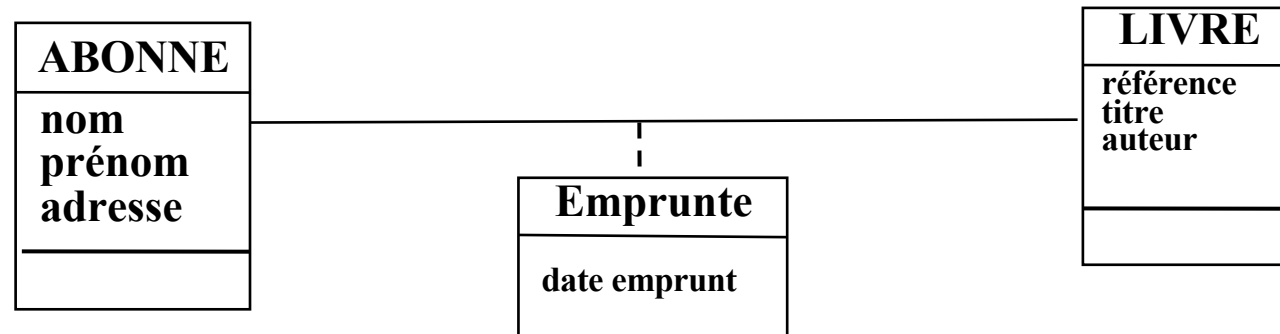
l'association ***Emprunte*** entre les classes ***Abonné*** et ***Livre***



2.2.2 Le concept d'Association

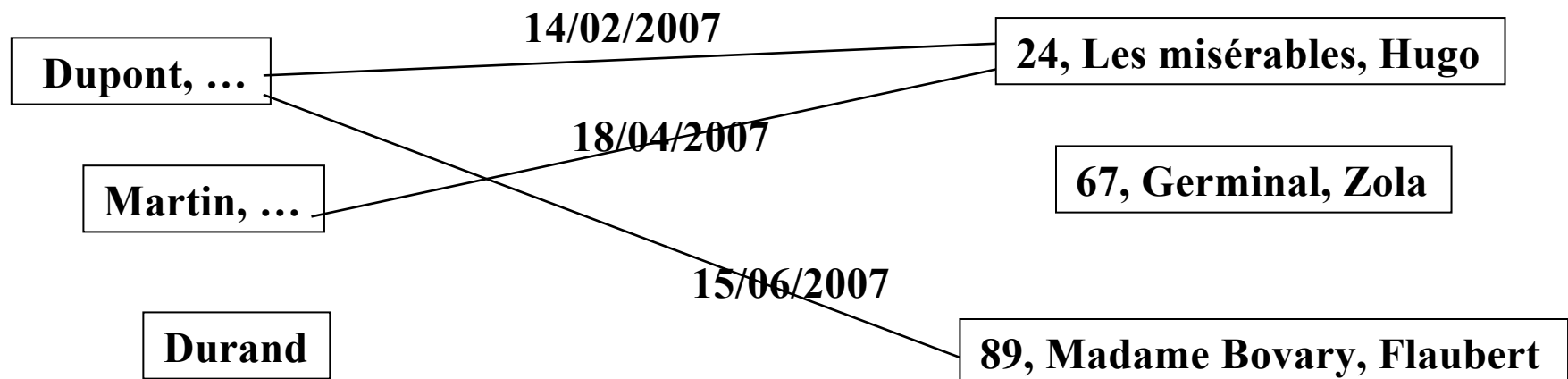
- Une association peut avoir des attributs

L'attribut 'date emprunt' d'un *livre* par un *abonné* concerne simultanément les 2 classes



Objets Abonné

Objets Livre



Trois liens de l'association «Emprunte»

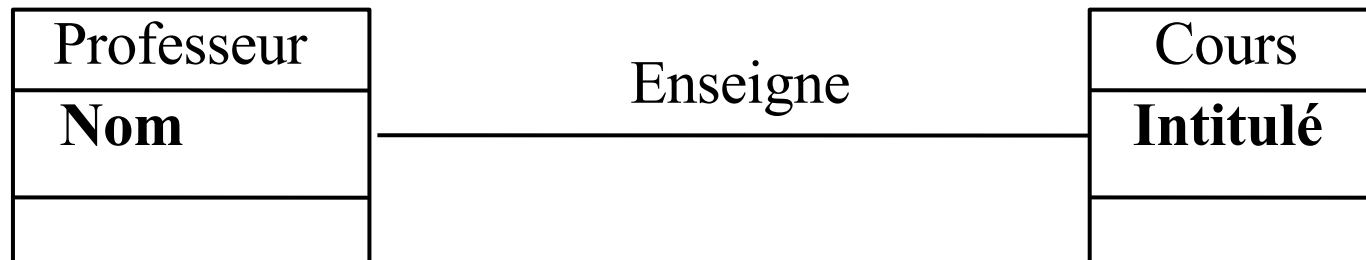
Le modèle utilisé pour obtenir *un modèle de classes d'objet* est composé des concepts suivants :

- **L'objet** : l'entité réelle
- **la classe d'objet** : regroupement des objets de même nature
- **l'attribut** : description des objets
- **l'association** : une relation sémantique entre classes

Exemple

Valeurs d'attribut

Le professeur *Hernandez* enseigne le cours d'*Algorithmique*
Le professeur *Laleau* enseigne le cours *SGBD*
Le professeur *Monnerat* enseigne le cours *Maths*



La classe Professeur a un attribut (Nom du professeur)

La classe Cours a un attribut (Intitulé du cours)

L'association Enseigne entre les classes Professeur et Cours

5) Le concept de **Contrainte d'intégrité** :

Une contrainte d'intégrité est définie comme une assertion qui doit être vérifiée par des données à des instants déterminés.

- **Les contraintes sur les propriétés**
- **La multiplicité d'une association**

a) Les Contraintes sur les propriétés

- Sur une propriété (attribut)
 - forme, liste de valeurs, fourchette de valeurs possibles
RéférenceLivre:entier, CatégorieClient:{10, 20, 30}, 14<âge<30
 - Stabilité (la valeur de la propriété ne change pas au cours du temps)
une date de naissance, N°Sécurité Sociale
- Sur plusieurs propriétés d'une même classe ou d'une même association
 - *heureDepart < heureArrivée ; dateEmprunt < dateRetour*
- Sur des propriétés de classe ou d'associations différentes
 - *Montant d'une commande est égal à la somme des montants des lignes de cette commande*

b) La multiplicité d'une association

C'est le nombre de participations d'un objet d'une classe dans une association.

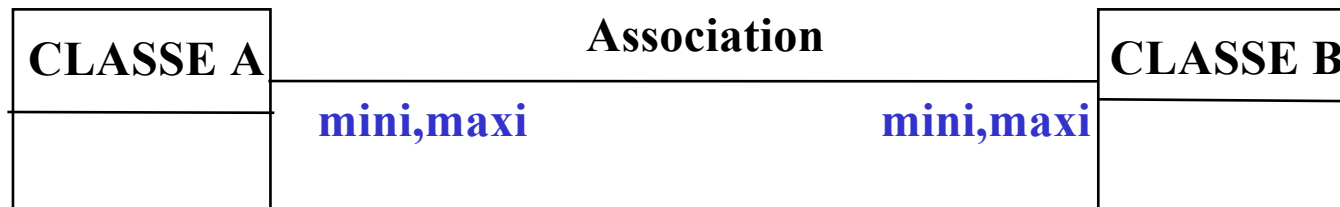
Elle s'exprime par 2 variables :

- **Multiplicité minimale : égal à 0 ou 1**

nombre de fois minimum qu'une instance d'une classe participe aux instances de l'association.

- **Multiplicité maximale : égal à 1 ou N (ou *)**

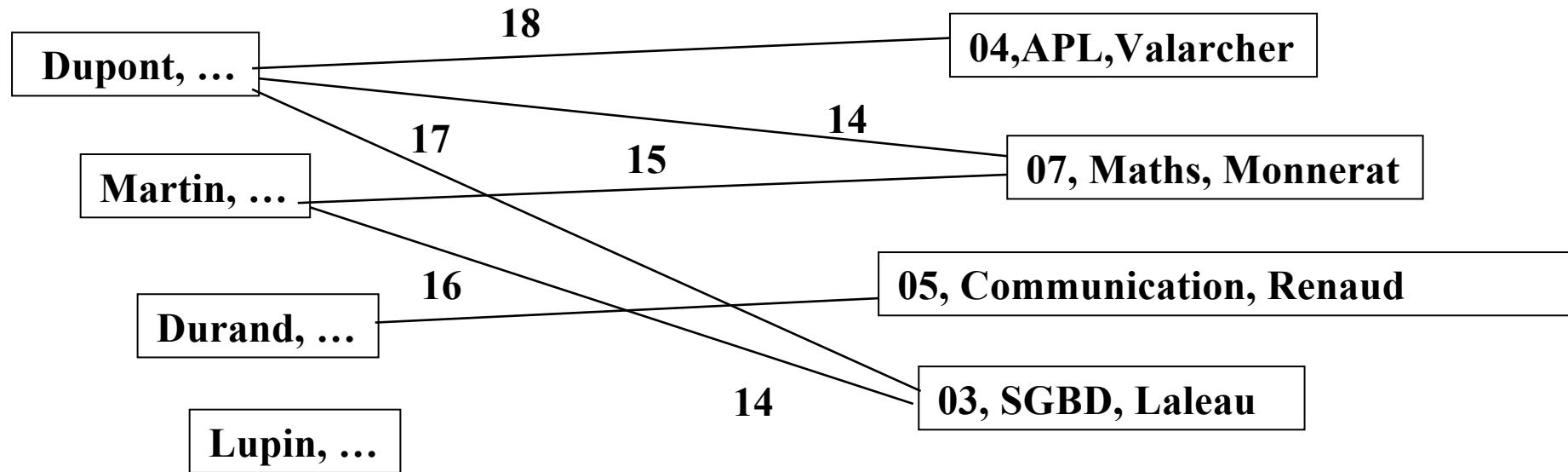
nombre de fois maximum qu'une instance d'une classe participe aux instances de l'association



2.2.3 Les contraintes d'intégrité

Objets Etudiants

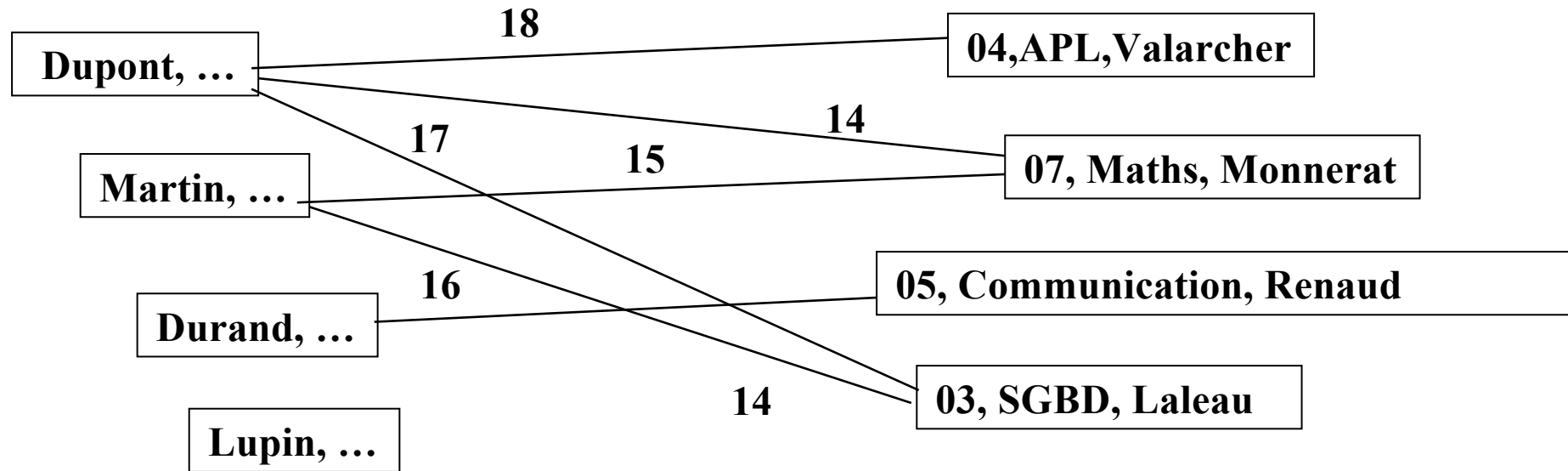
Objets Cours



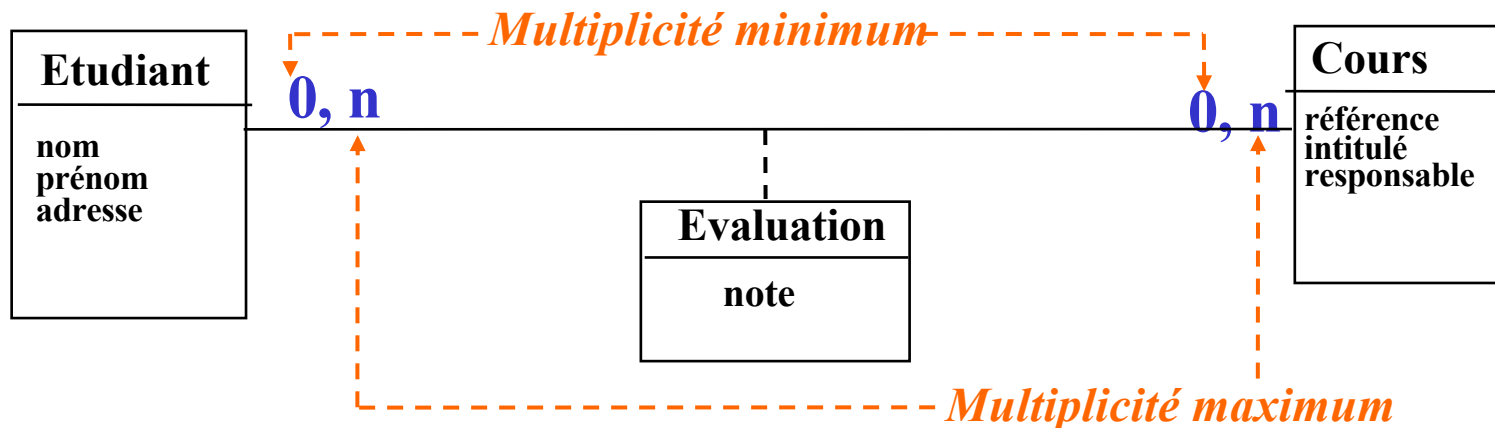
2.2.3 Les contraintes d'intégrité

Objets Etudiants

Objets Cours



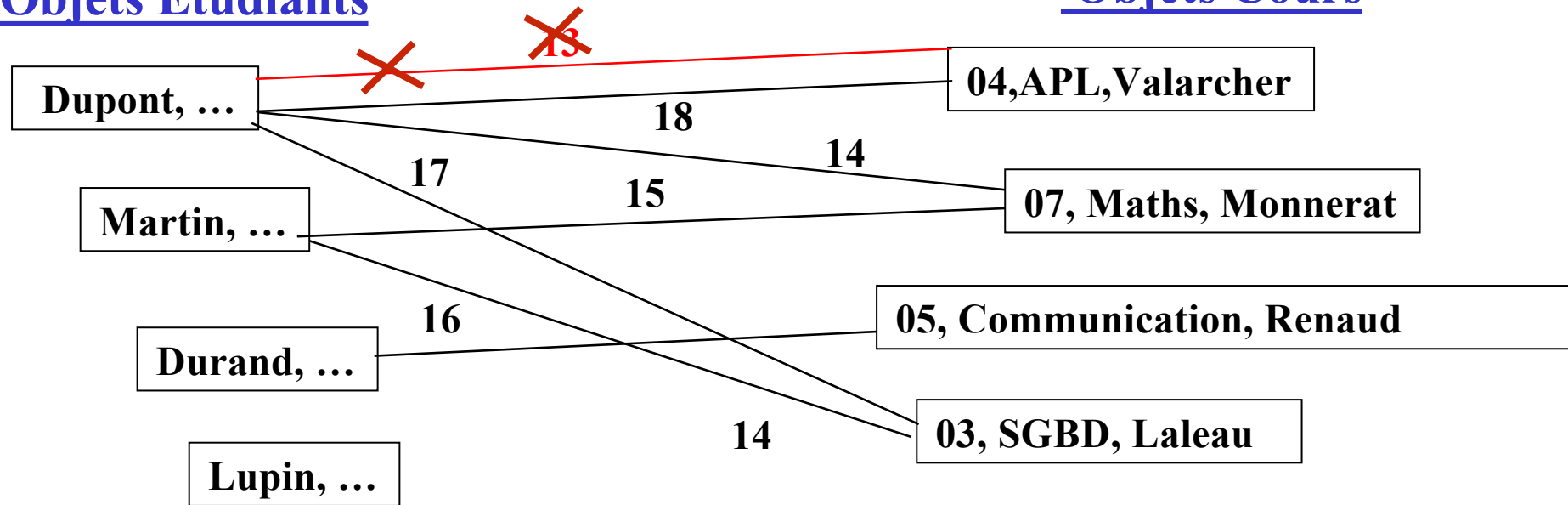
Le modèle de classes avec les multiplicités correspondantes



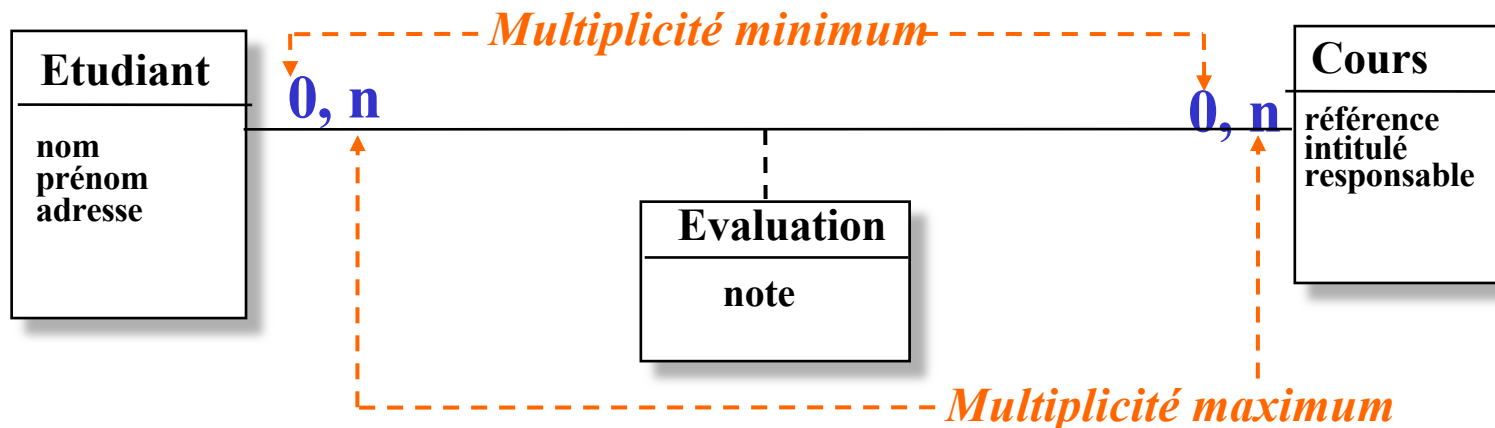
2.2.3 Les contraintes d'intégrité

Objets Etudiants

Objets Cours



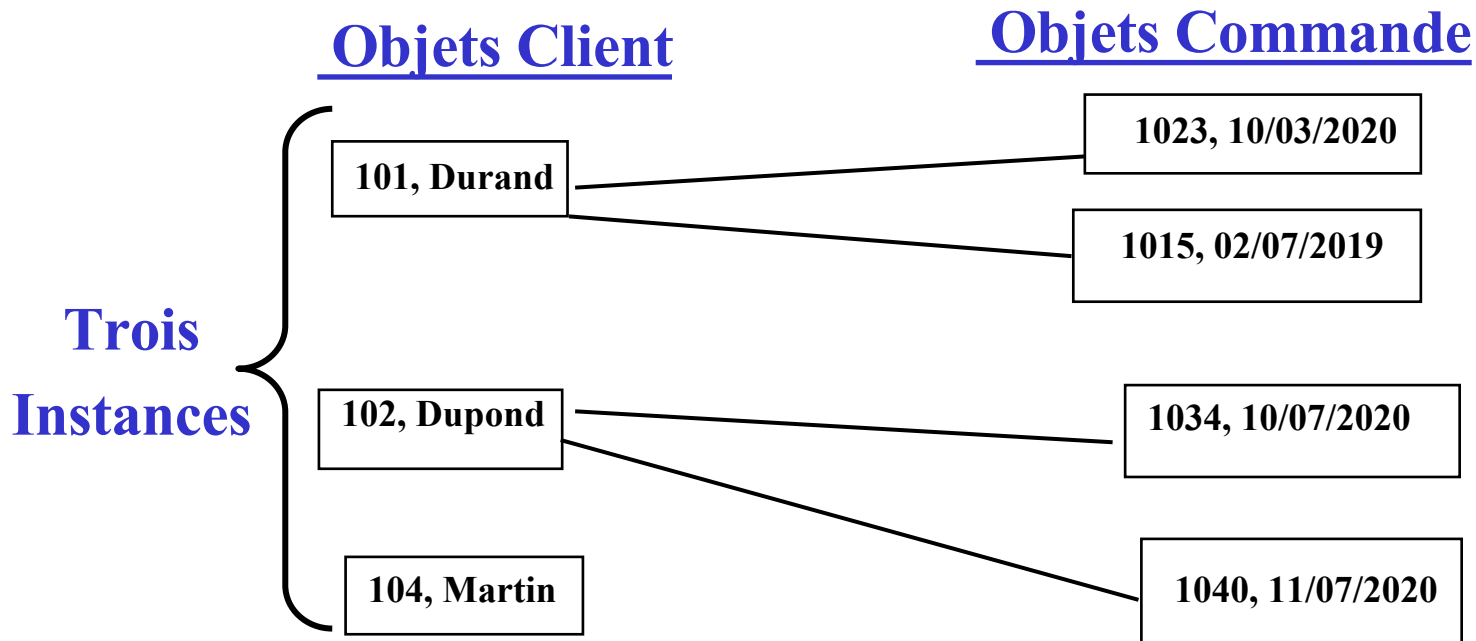
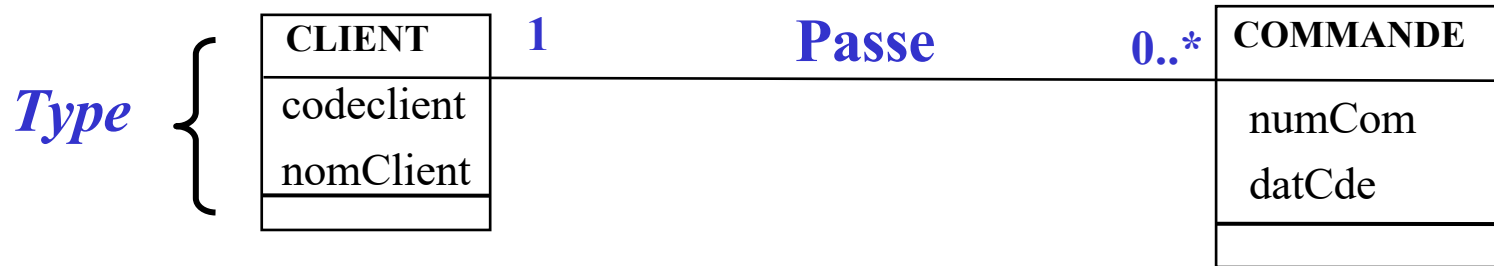
Le modèle de classes avec les multiplicités correspondantes



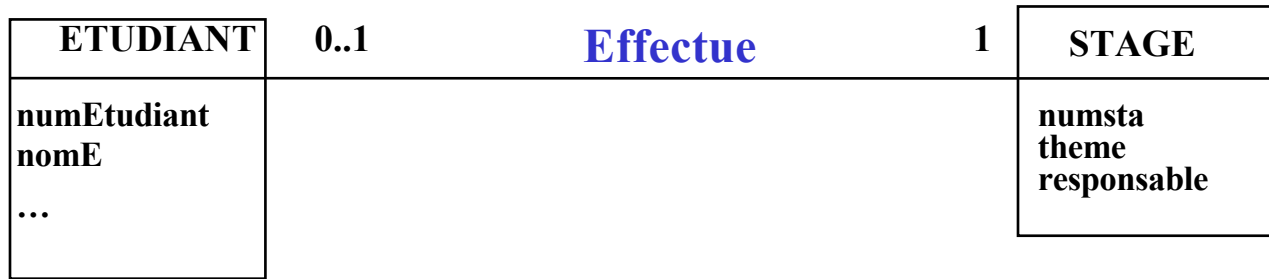
Les différentes multiplicités

- **0..1** est lié au minimum à 0 et au maximum à 1
- **1** est lié au minimum à 1 et au maximum à 1
- **0..N** ou **0..*** ou ***** est lié au minimum à 0
- **1..N** ou **1..*** est lié au minimum à 1
- **n..m** permet de spécifier des nombres autre que 0 ou 1

Exemple 1 : représenter le fait qu'un client passe plusieurs commandes mais une commande n'est passée que par un seul client

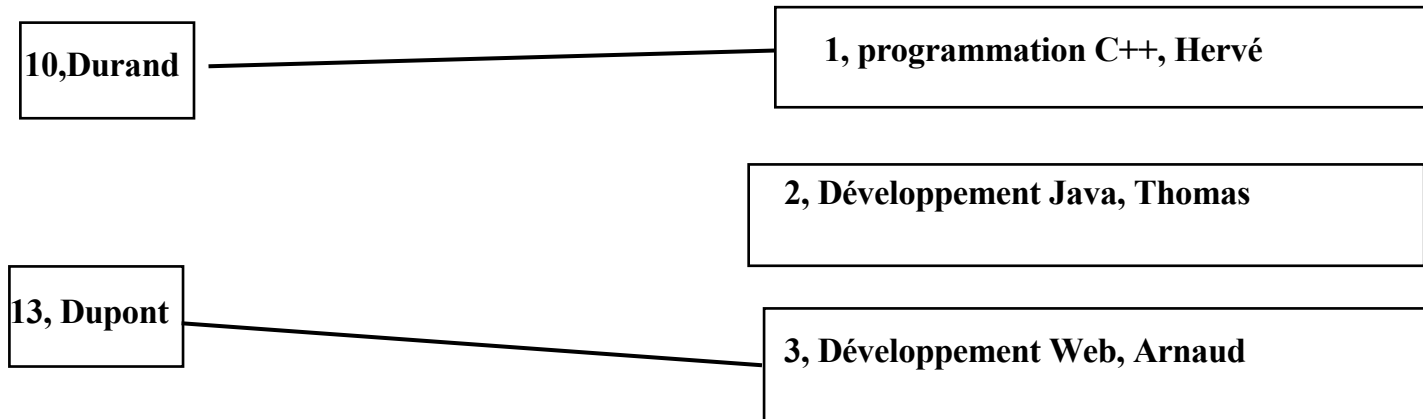


Exemple 2 : représenter le fait qu'un étudiant effectue un stage et qu'il existe des stages non affectés à des étudiants ou en attente d'affectation



Objets Etudiant

Objets stage



Le modèle utilisé pour obtenir *un modèle de classes d'objet* est composé des concepts suivants :

- **L'objet** : l'entité réelle
- **la classe d'objet** : regroupement des objets de même nature
- **l'attribut** : description des objets
- **l'association** : une relation sémantique entre classes
- **Contrainte d'intégrité**
 - multiplicité d'une association
 - propriétés

2.3. Démarche de construction d'un modèle Objet

2.1 Introduction à la modélisation

2.2 Concepts et Règles de modélisation Objet

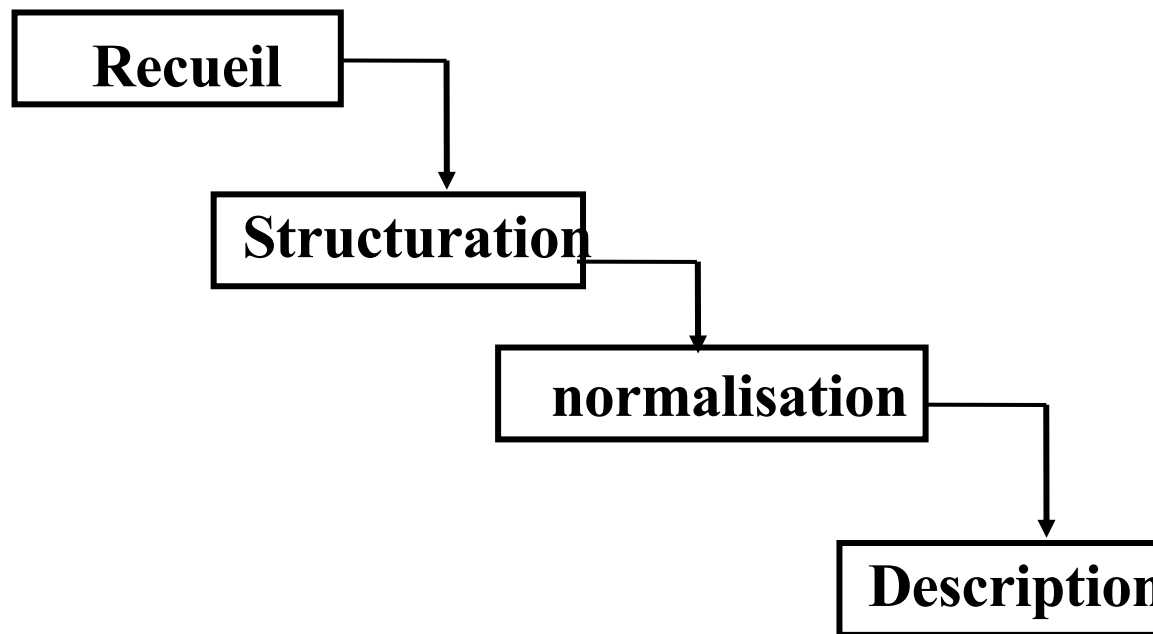
2.3 La démarche de construction

2.4 Notions complémentaires

2.5 Conclusion

2.3. Démarche de construction d'un modèle Objet

la démarche est un processus de construction d'un modèle de classes d'objet constitué de 4 étapes :

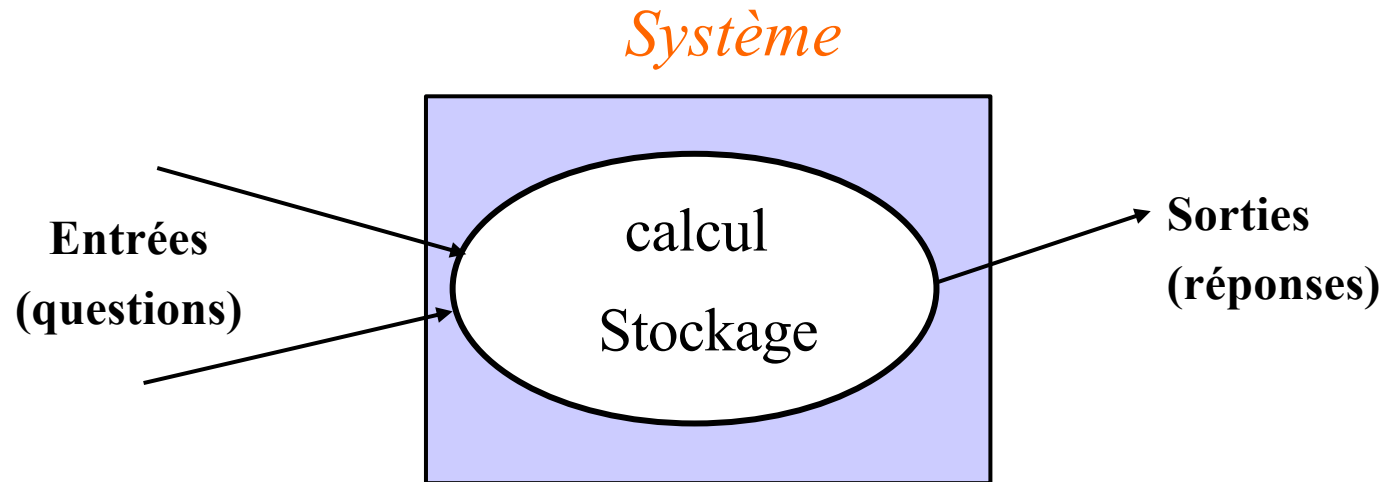


- **Le processus** est "le chemin qu'il faut parcourir pour atteindre le résultat".
- **Le modèle de classes d'objet** est le résultat à atteindre.

2.3. Démarche de construction d'un modèle Objet

2.3.1 Recueil des informations

- Consiste à recueillir, auprès des utilisateurs, les informations utiles.
- L'utilité d'une information se mesure en examinant les objectifs assignés au système

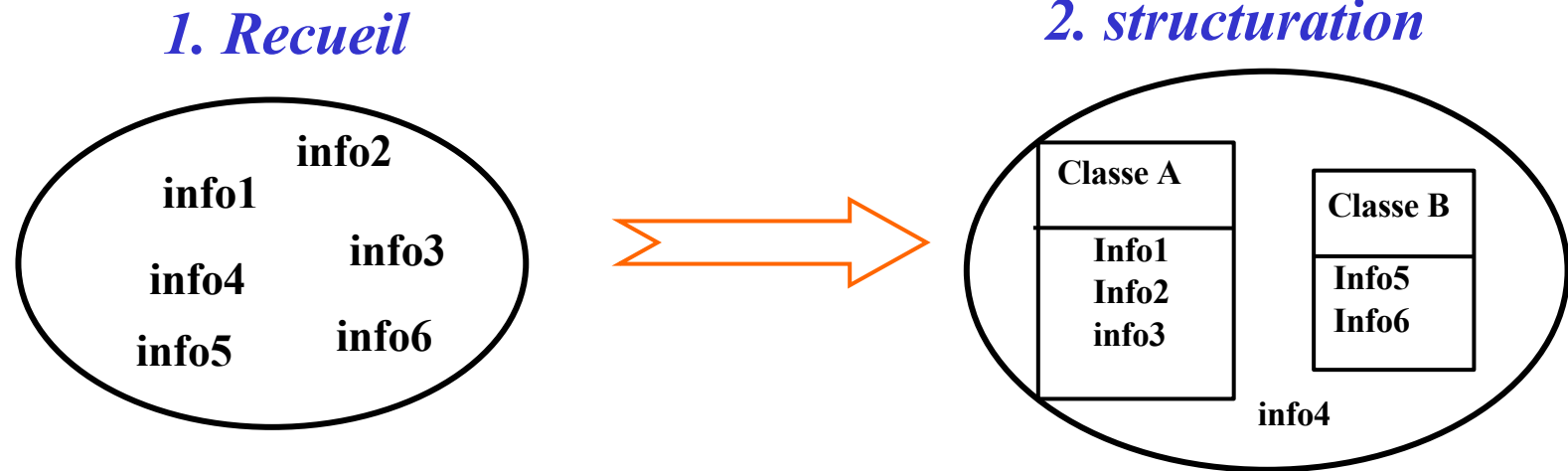


Cette étape permet d'identifier les besoins en informations auprès des utilisateurs du futur système

2.3. Démarche de construction d'un modèle Objet

2.3.2 Structuration

1) mettre en évidence les classes en regroupant les informations par affinité



Règles à respecter

- Les informations regroupées forment les attributs de la classe d'objet identifiée
- Une fois positionnée dans une classe, une information n'est plus disponible pour décrire une autre classe

2) Identifier les associations non porteuses d'information

Un verbe se traduit généralement dans un modèle par une association entre 2 ou plusieurs classes d'objet

3) Mettre en évidence les associations porteuses d'informations

Les informations n'appartenant à aucune classe sont des informations à placer dans des associations

4) Définir la multiplicité pour les classes d'objet impliquées dans une association

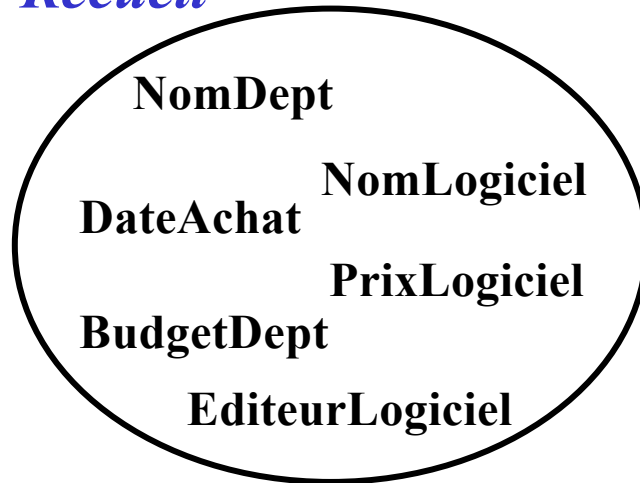
Pour déterminer les multiplicités d'une association entre 2 classes, la règle consiste à fixer une instance d'une classe A , puis à déterminer le nombre minimal et maximal d'instances liées à l'autre extrémité de la classe B.

2.3. Démarche de construction d'un modèle Objet

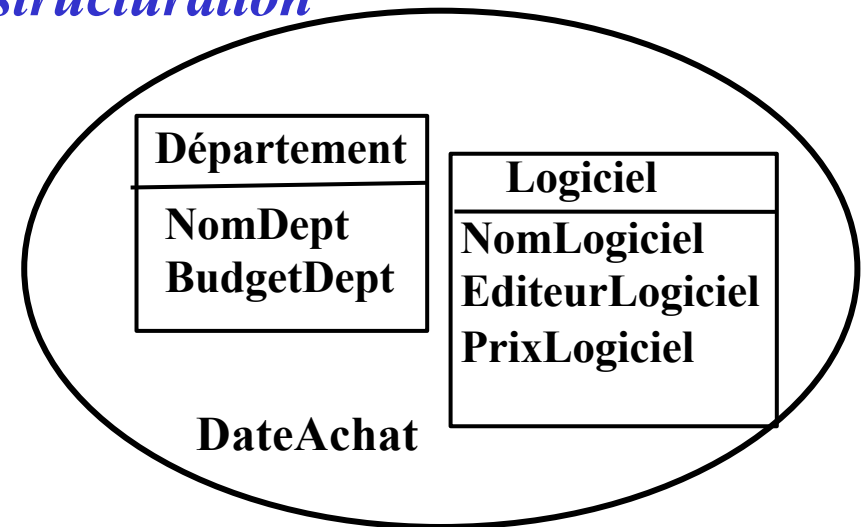
Exemple

Les départements de l'IUT achètent des logiciels chez des éditeurs de logiciels. Chaque département a un budget alloué pour payer les logiciels achetés.

1. Recueil



2. structuration



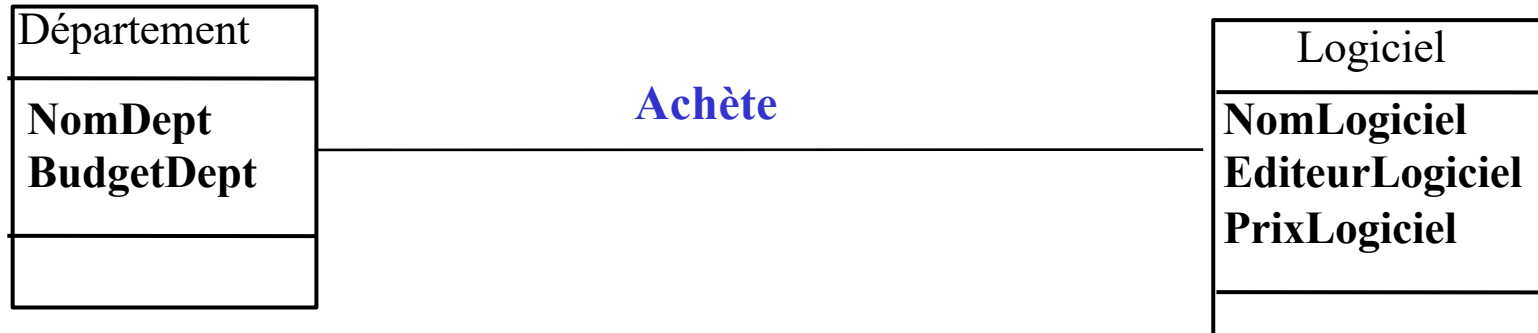
– Les informations 'NomDept' et 'BudgetDept' sont des attributs de la classe

Département

– Les informations 'NomLogiciel', 'EditeurLogiciel' et 'PrixLogiciel' sont des attributs de la classe **Logiciel**

2.3. Démarche de construction d'un modèle Objet

Solution



Questions

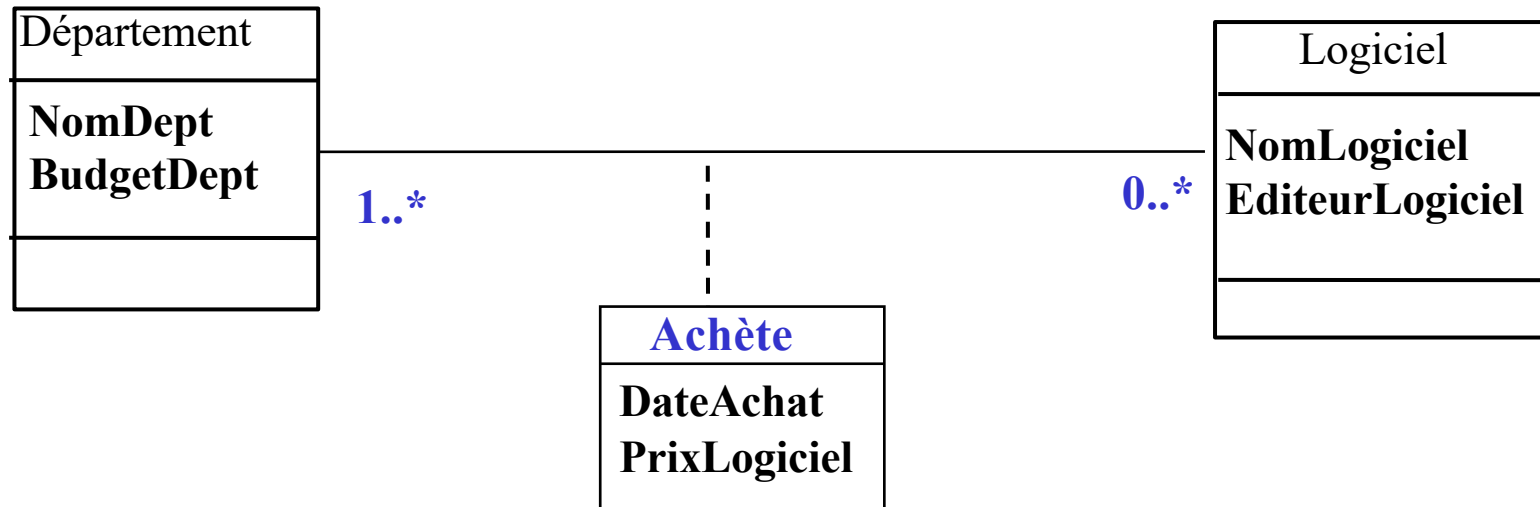
- L'information 'PrixLogiciel' est un attribut de la classe d'objet 'Logiciel':
signifie que les départements achètent le même logiciel au même prix
- L'information 'DateAchat': est ce que les départements achètent un logiciel à la même date ?
Réponse : Non
- L'information 'DateAchat': est ce qu'un département achète plusieurs fois le même logiciel à des dates différentes ?
Réponse : Non

Toujours se demander de la pertinence de l'information dans la classe.
Est ce que **cette information caractérise cette classe?**

2.3. Démarche de construction d'un modèle Objet

et si un logiciel est vendu à des prix différents selon les départements??

Solution



Multiplicité

- Un département, au minimum, n'achète pas de logiciel (=0) et au maximum plusieurs logiciels (=*)
- Un logiciel est acheté au minimum par un département (=1) et au maximum par plusieurs (=*)

2.3.3 Normalisation

Les informations retenues sont celles qui vont être mémorisées (stockées) dans le système. La normalisation consiste à vérifier que certaines règles de 'bonne qualité' sont respectées.

□ Règle 1 : caractère élémentaire d'un attribut

Tous les attributs sont atomiques dans le sens qu'ils doivent être non décomposables

Ex : l'attribut *adresse* composé d'un *numéro de rue*, du *nom de la rue* et du *code postal* n'est pas un attribut atomique.

Il doit être remplacé par les attributs atomiques : *numéro de rue*, *nom de la rue*, *code postal*

□ Règle 2 : identité et attribut

Un objet représente une entité du monde réel définie par un état (valeurs d'attribut, un comportement (évolution des données)) et une identité

Comment représenter cette identité dans le modèle de classes d'objets??

Solution

- Choisir une propriété (attribut) qui permet de distinguer les objets
ou
- Ajouter un attribut numéro **si aucune propriété ne permet de faire cette distinction**

Cas particulier : classe-association et classe qui dépend d'une autre classe (cf ci-dessous)

Exemple

Classe Abonné

ABONNE
<u>NumAbonné</u> : entier
Nom : chaîne
Adresse : chaîne
DateNaissance : date

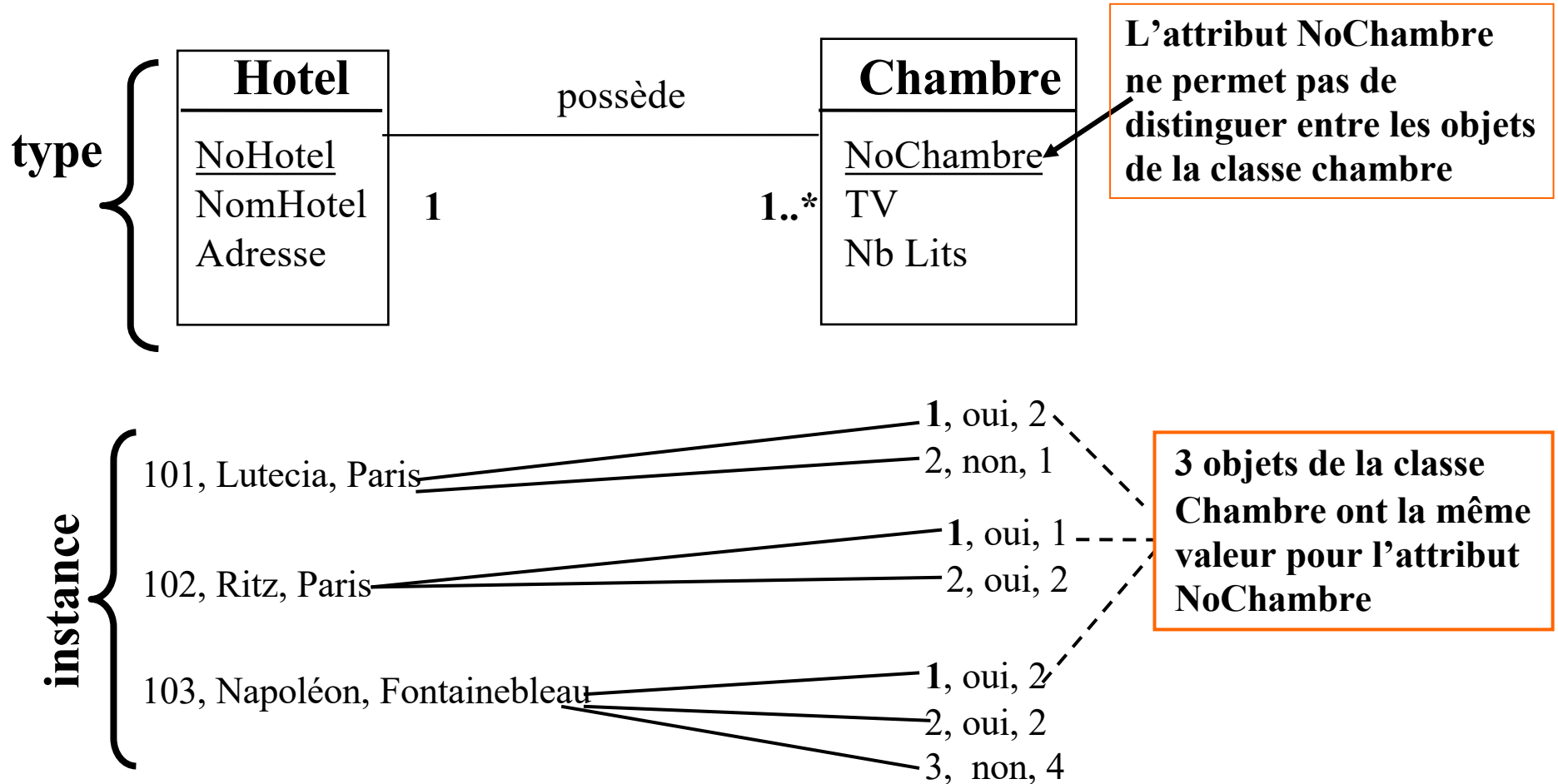
Des objets Abonnés

001, Dupont, 4 rue Duchemin 75013 Paris, 12/03/85
002, Durand, Cours Balzac 75012 Paris, 25/08/67
003, Martin, 4 rue des Lilas 75012 Paris, 18/09/88

L'attribut qui permet de distinguer entre les objets de la classe est souligné.

C'est l'**identifiant** de la classe d'objet

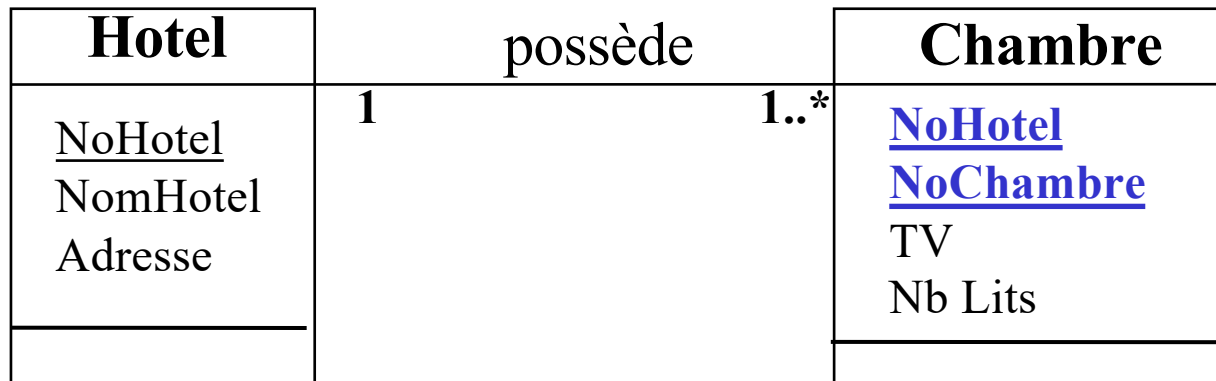
Exemple : représenter les hôtels et les chambres à louer



Une chambre dépend fortement de l'hôtel à laquelle elle appartient

Solution

L'identifiant de la classe d'objet est constitué du numéro de séquence et de l'identifiant de la classe d'objet dont il dépend.

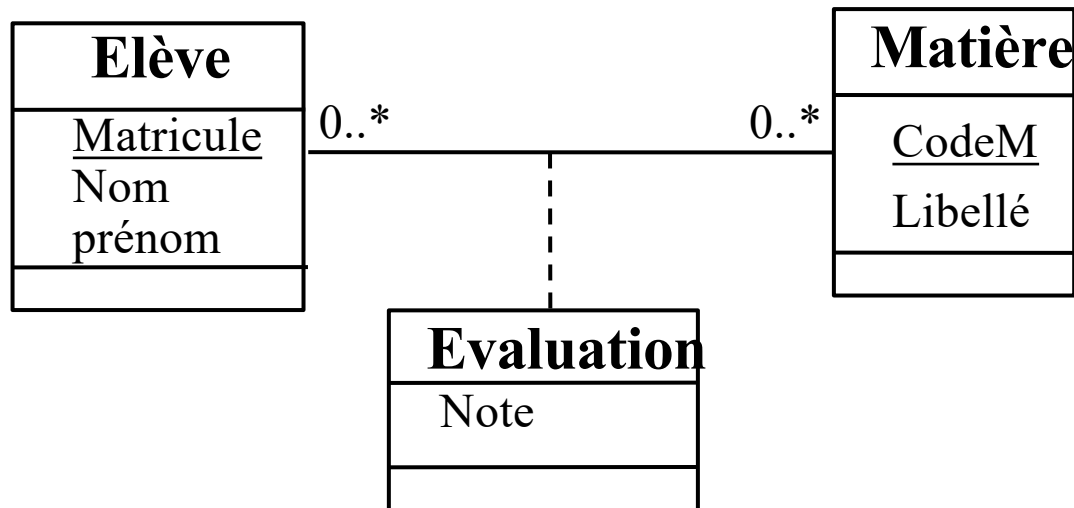


L'identifiant de la classe Chambre est composé des attributs NoHotel+ NoChambre.

On l'appelle identifiant relatif

L'identifiant d'une classe-association est constitué des identifiants de chaque classe qui participe à l'association.

Représenter les notes des élèves par matière, sachant que pour une matière donnée il y a une seule évaluation (note) par étudiant.



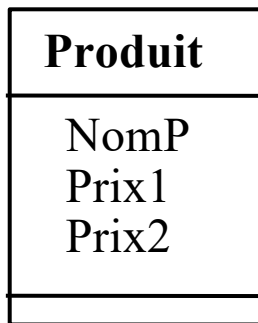
L'identifiant de la classe-association est le couple (Matricule, CodeM)

□ Règle 3 : Vérification de la non redondance

Un attribut figure *une seule fois* dans le modèle de classes d'objet soit dans une classe d'objet, soit dans une association avec ou sans attribut

Considérons un magasin de vente où un produit peut être vendu selon des prix différents : un prix X pour les clients (CA > 20000) et un prix Y pour les autres

Représentation de la classe Produit

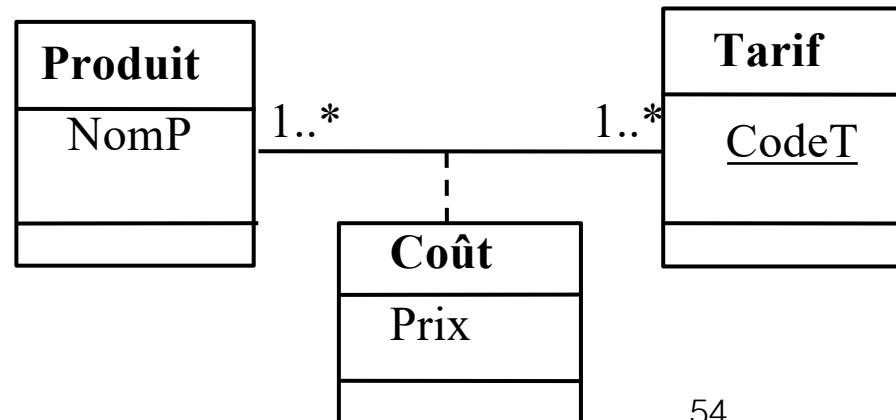


Cette solution n'est pas correcte car l'attribut Prix est représenté 2 fois (Prix1 & Prix2)

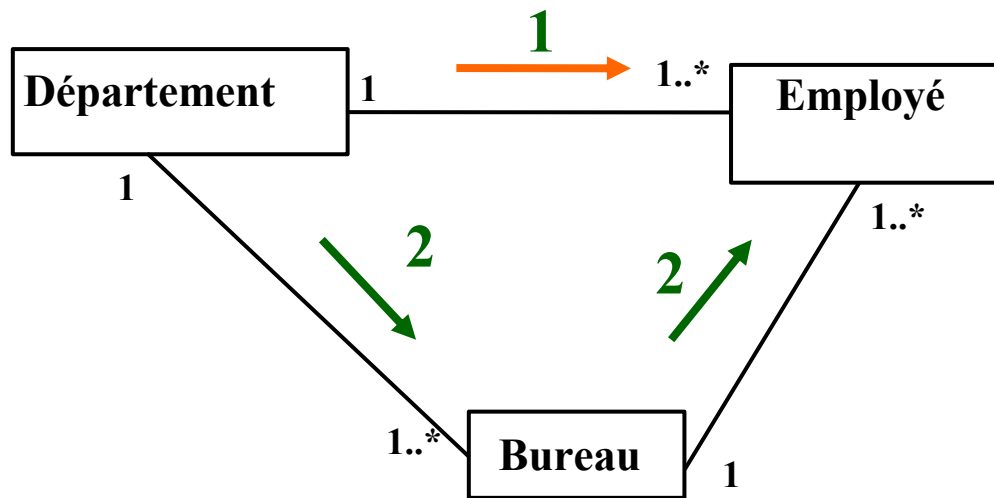
L'attribut Prix dépend simultanément des classes Produit et Tarif.

La classe Tarif regroupe les codes tarif définis selon les catégories de client.

Solution

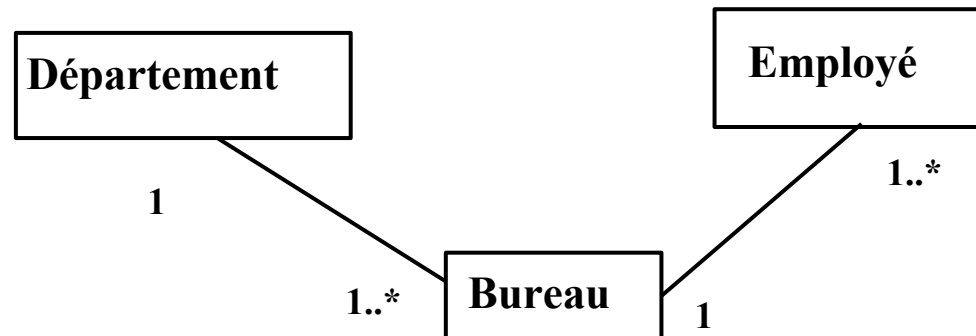


– Redondance association cyclique



Pour déterminer les employés d'un département, il y a 2 chemins

Solution : application de la règle de transitivité



□ Règle 4 : 1ere forme normale

Chaque attribut d'une classe (ou d'une classe–association) possède une valeur à un instant t.

Ex : l'attribut *Prénom* dans une classe *Personne* signifie que la personne *Dupont* ne peut avoir qu'un seul prénom

Alors comment faire si on veut représenter tous les prénoms d'une personne?

- Une classe est en première forme normale si chaque attribut est en dépendance fonctionnelle avec l'identifiant de la classe.
- En d'autres termes, à une valeur de l'identifiant, on a au plus une valeur de tout autre attribut de la classe

Conclusion normalisation

- Règle 1 : **caractère élémentaire d'un attribut**
- Règle 2 : **identité et attribut**
- Règle 3 : **Pas de redondance**
- Règle 4 : **1ère forme normale**

2.3.4 Description

Cette phase consiste à élaborer une documentation contenant une description de tous les éléments du modèle de classes

1. Décrire chaque classe du modèle
2. Décrire chaque association
3. Donner les contraintes d'intégrité liées à chaque classe
4. Donner les contraintes d'intégrité liées à plusieurs classes

2.4 Notions complémentaires

2.1 Introduction à la modélisation

2.2 Concepts et Règles de modélisation Objet

2.3 La démarche de construction

2.4 Notions complémentaires

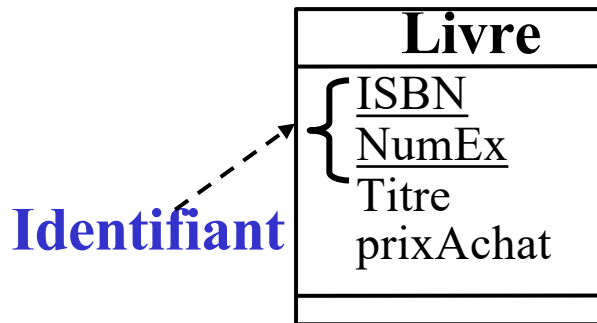
2.5 Conclusion

2.4 Notions complémentaires : l'identifiant

□ Règle de 2eme forme normale

1) Chaque attribut d'une classe dépend de l'identifiant de la classe ;

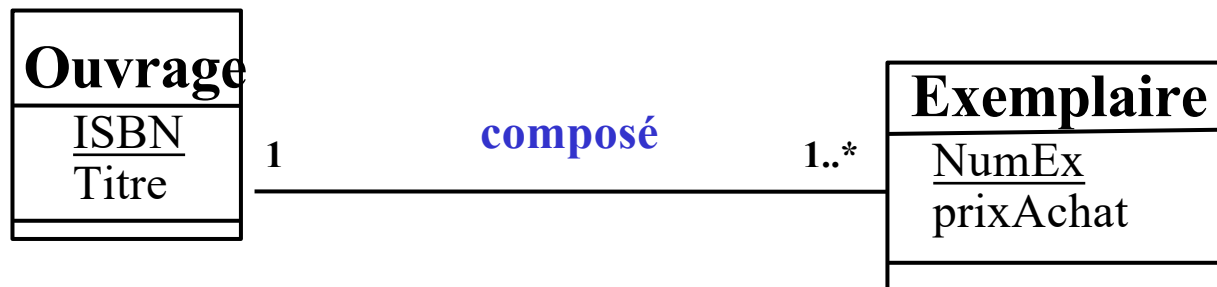
Dans une bibliothèque , certains livres existent en plusieurs exemplaires, comment les représenter dans le modèle de classes



- l'identifiant est composée de 2 attributs
- L'attribut Titre dépend de l'attribut ISBN
- PrixAchat dépend de ISBN+NumEx

**Solution
incorrecte**

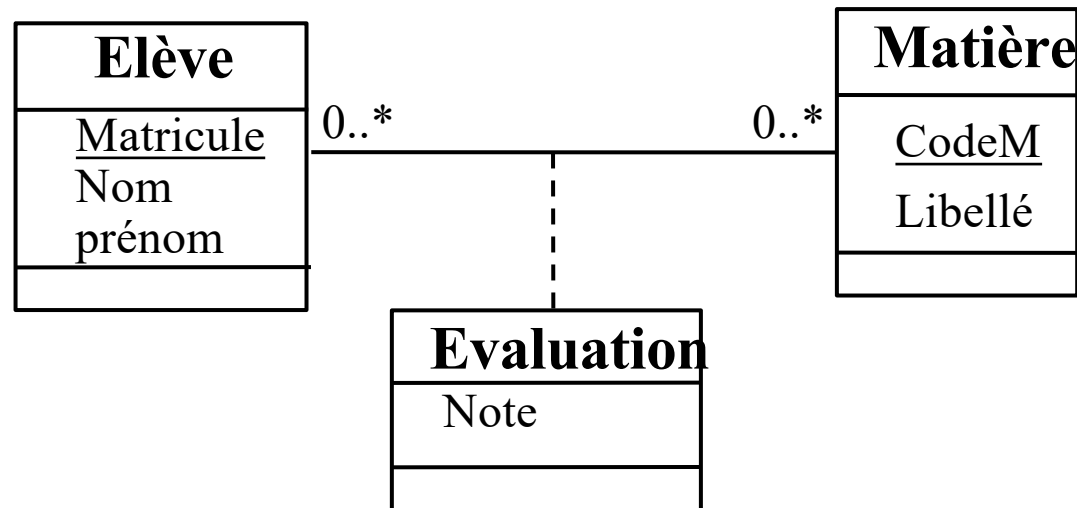
La solution est la suivante :



2.4 Notions complémentaires : l'identifiant

2) Chaque attribut d'une classe–association dépend simultanément des identifiants des classes participant à l'association

Représenter les notes des élèves par matière



A chaque couple (Matricule, CodeM) correspond une seule note

Rappel

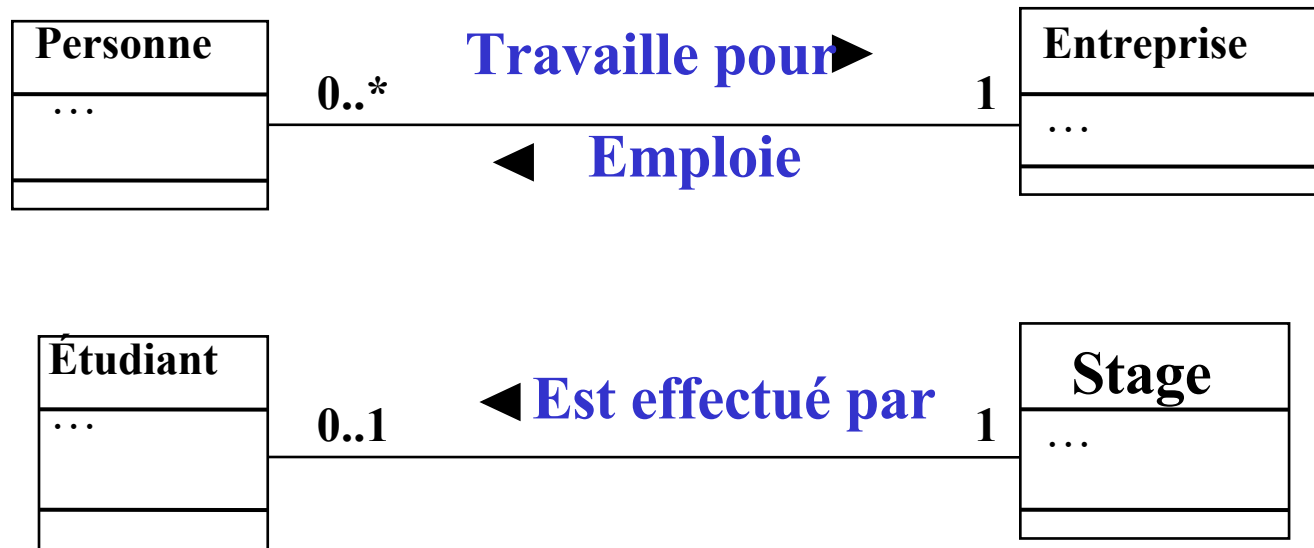
- Une association est une relation sémantique entre classes
- Un lien est une connexion physique entre objets, c'est l'instance d'une association

Degré d'une association

Le nombre de classes qui participent à l'association

Nom d'une association

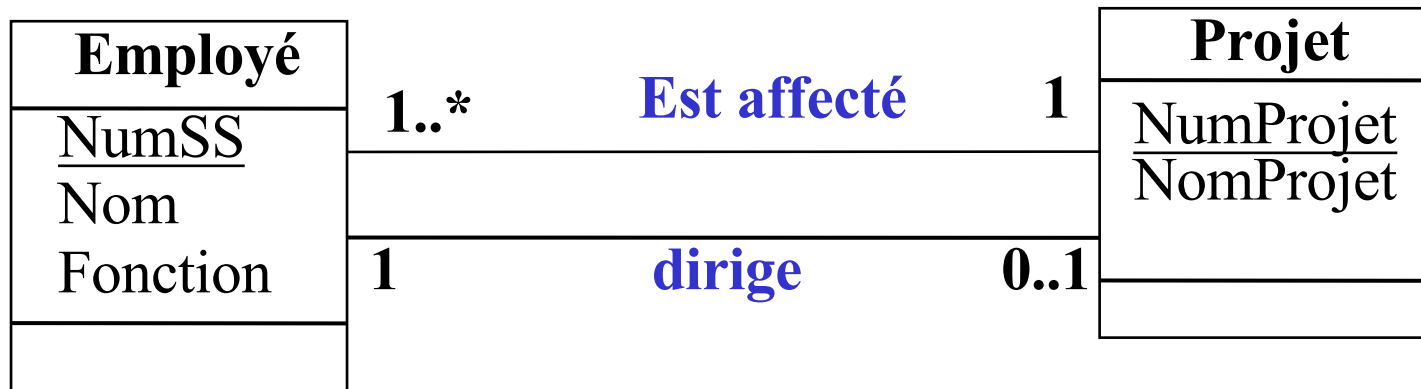
- Nommer les associations par une forme verbale active ou passive,
- Le sens de lecture peut être précisé au moyen d'un petit triangle dirigé vers la classe désignée par la forme verbale



2.4 Notions complémentaires : l'association

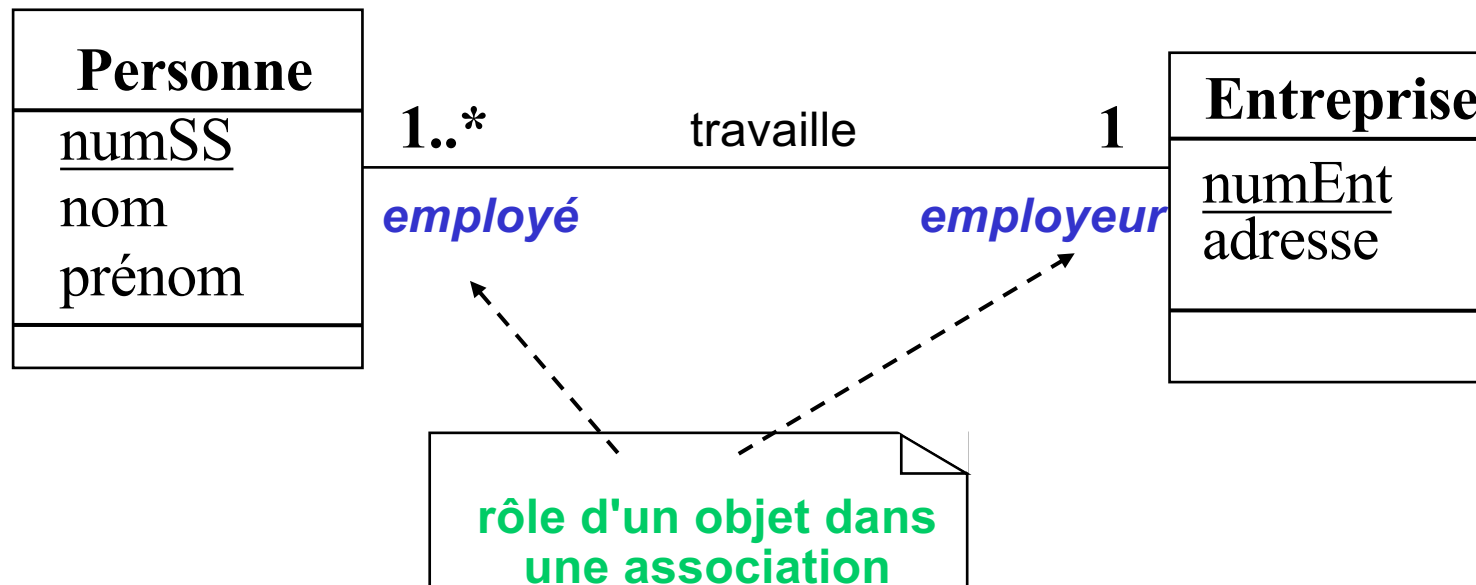
- **Plusieurs associations de nature différente peuvent concerner les mêmes classes d'objet**

Représenter les employés affectés à un projet et aussi le fait qu'un des employés soit le chef de projet



2.4 Notions complémentaires : l'association et le rôle

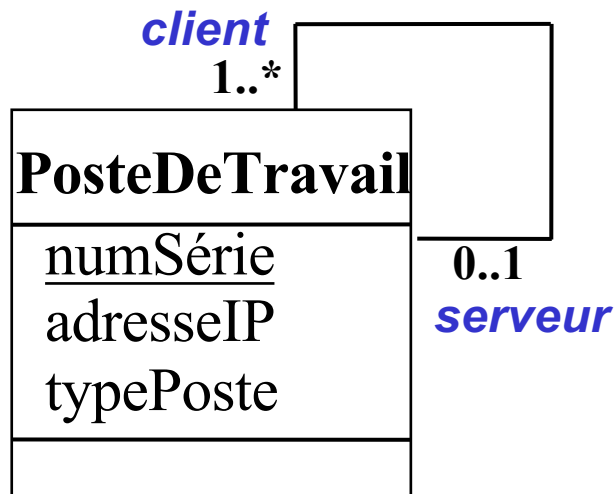
- **Rôle** des participants dans une association :
 - Un "rôle" peut être spécifié pour une extrémité de l'association.
 - Il exprime le rôle d'une classe dans l'association.
 - il facilite la lecture et la compréhension du modèle objet.



2.4 Notions complémentaires : l'association réflexive

- Une association réflexive est une association binaire qui fait intervenir deux fois la même classe

On veut représenter les postes de travail du département informatique, certains sont des postes client (salle TP), d'autres sont des serveurs



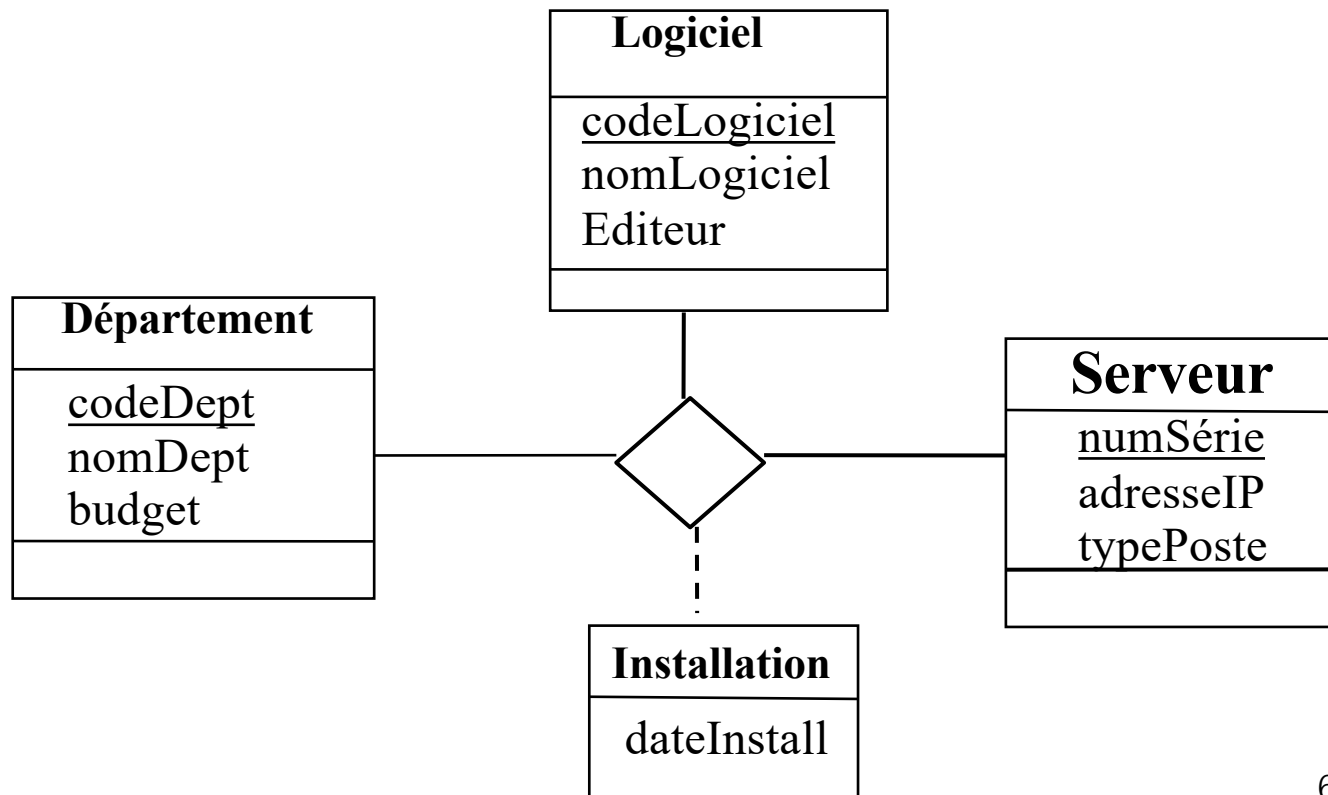
- Un poste serveur est lié à 1 ou plusieurs postes client
- Un poste client est relié à aucun serveur ou à un seul serveur

Le nommage des rôles est essentiel à la clarté du diagramme.

2.4 Notions complémentaires : l'association n-aire

Une association n-aire est une association qui fait intervenir plusieurs classes

Les logiciels sont installés sur des serveurs à l'initiative des départements, on veut représenter la date à laquelle le logiciel est installé



Comment déterminer les multiplicités?

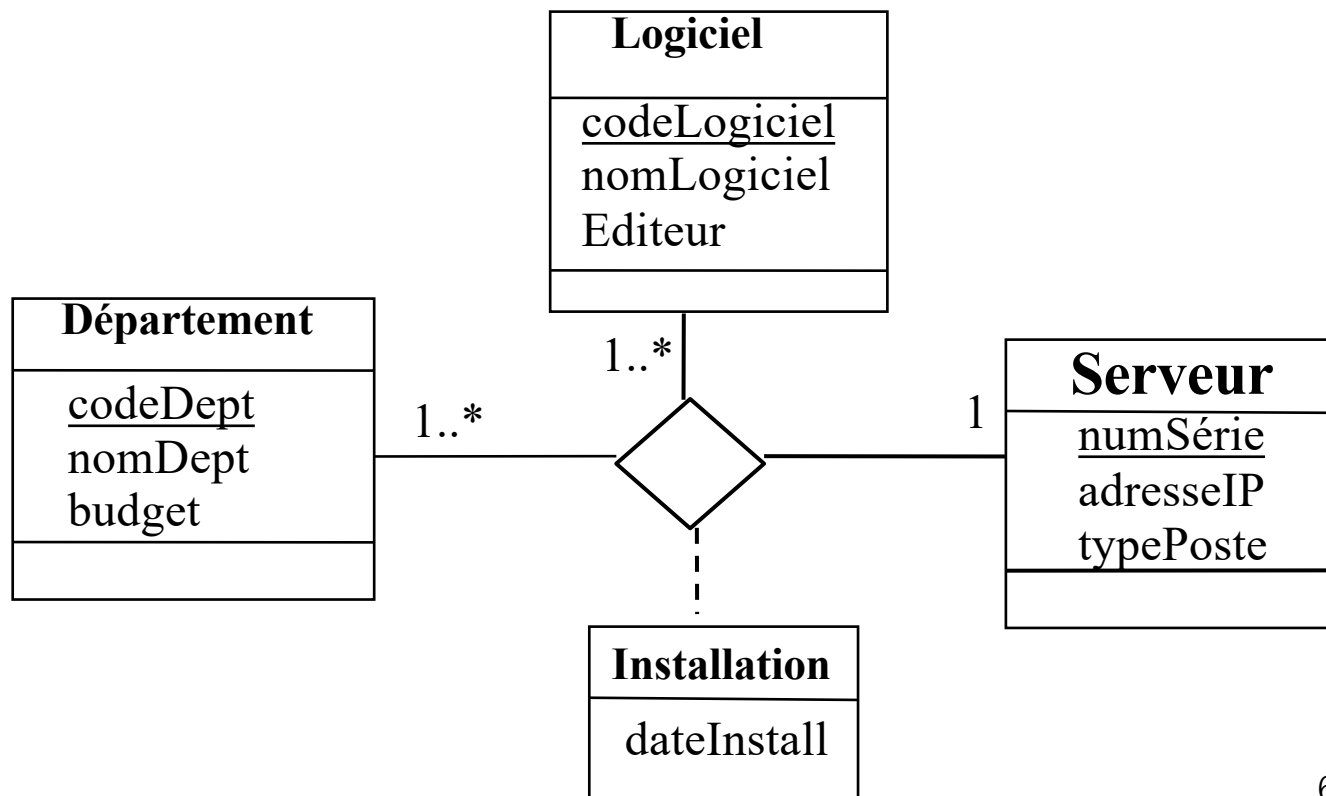
- **Classe logiciel** : combien de logiciels peuvent être associés à un couple (Département, Serveur)?
- **Classe serveur** : combien de serveurs peuvent être associés à un couple (Département, Logiciel)?
- **Classe département** : combien de départements peuvent être associés à un couple (Serveur, Logiciel)?

Au couple d'instances (a, b) des classes A et B, combien d'instances (c) de la classe C peuvent être associées à ce couple ?

2.4 Notions complémentaires : l'association n-aire

Une association n-aire est une association qui fait intervenir plusieurs classes

Les logiciels sont installés sur des serveurs à l'initiative des départements, on veut représenter la date à laquelle le logiciel est installé



2.4 Notions complémentaires : l'héritage

□ Le concept d'héritage

Considérons le cas suivant : comment représenter les personnels permanents et les personnels vacataires d'une entreprise?

Solution 1

instances

Personnel
NumE
nom
Prénom
Date d'entrée
salaire mensuel
date début vacation
durée de vacation

101, Dupond, Jean, 01/09/1997, 2000€, NULL, NULL

102, Durand, Annie, NULL, NULL, 19/10/2019, 25/02/2020

103, Ritz, Frédéric, 01/12/1975, 4000€, NULL, NULL

104, Napoléon, Bonaparte, 01/01/2000, 10000€, NULL, NULL

105, Michel, Emeline, NULL, NULL, 17/01/2020, 17/07/2020

2.4 Notions complémentaires : l'héritage

Solution 2

Personnel permanent
NumEP nom Prénom Date d'entrée salaire mensuel

Personnel vacataire
NumEV nom Prénom date début contrat durée de contrat

101, Dupond, Jean, 01/09/1997, 2000€

103, Ritz, Frédéric, 01/12/1975, 4000€

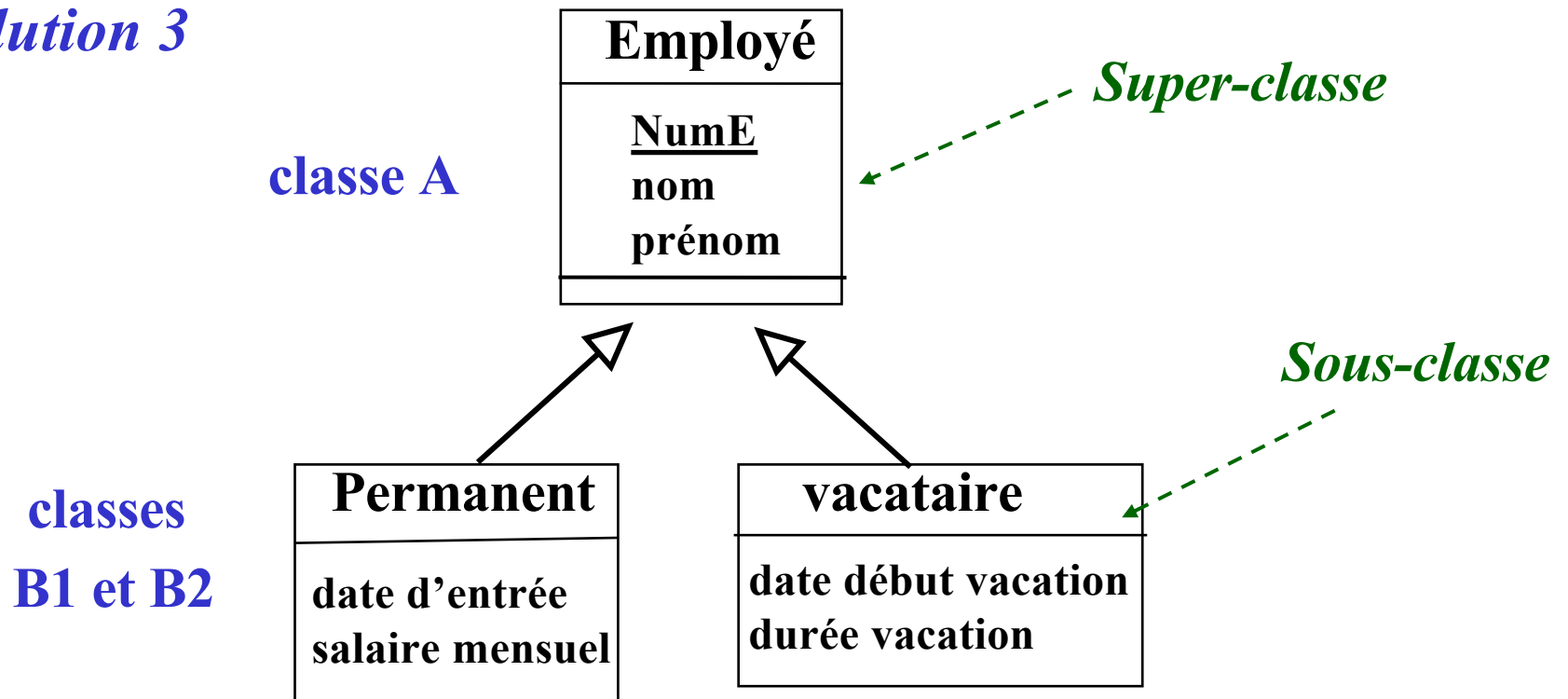
104, Napoléon, Bonaparte, 01/01/2000, 10000€

102, Durand, Annie, 19/10/2019, 25/02/2020

105, Michel, Emeline, 17/01/2020, 17/07/2020

2.4 Notions complémentaires : l'héritage

Solution 3

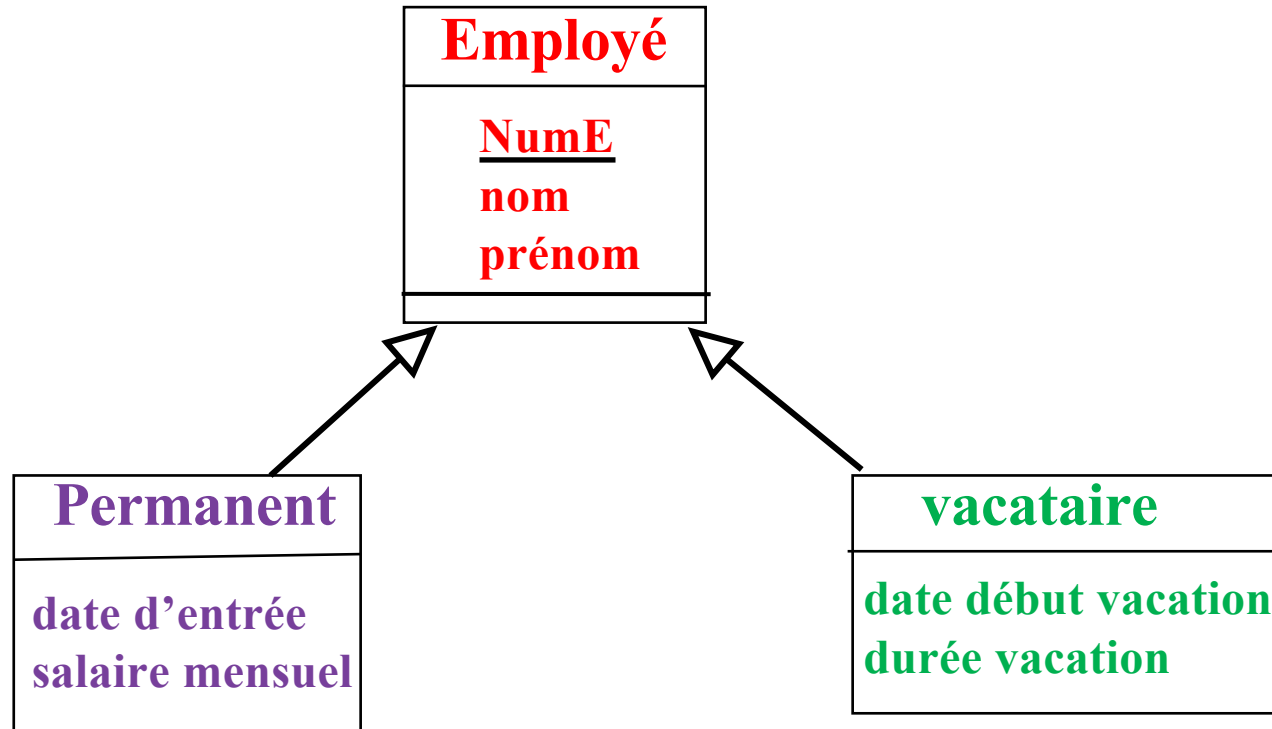


- *Les classes B1 et B2 héritent de la classe A* : signifie que tous les attributs de la classe A sont également ceux des classes B1 et B2

• L'héritage est une relation entre classes qui permet de représenter les attributs communs dans une classe (super-classe) et les attributs spécifiques dans une autre classe (sous-classes)

2.4 Notions complémentaires : l'héritage

Solution 3



101, Dupond, Jean, 01/09/1997, 2000€
103, Ritz, Frédéric, 01/12/1975, 4000€
104, Napoléon, Bonaparte, 01/01/2000, 10000€

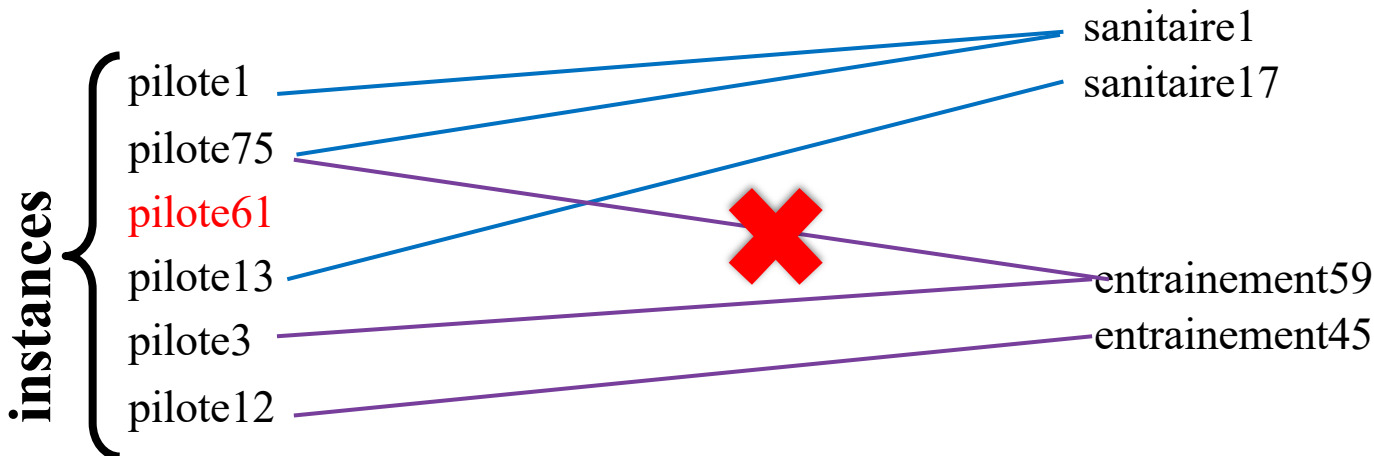
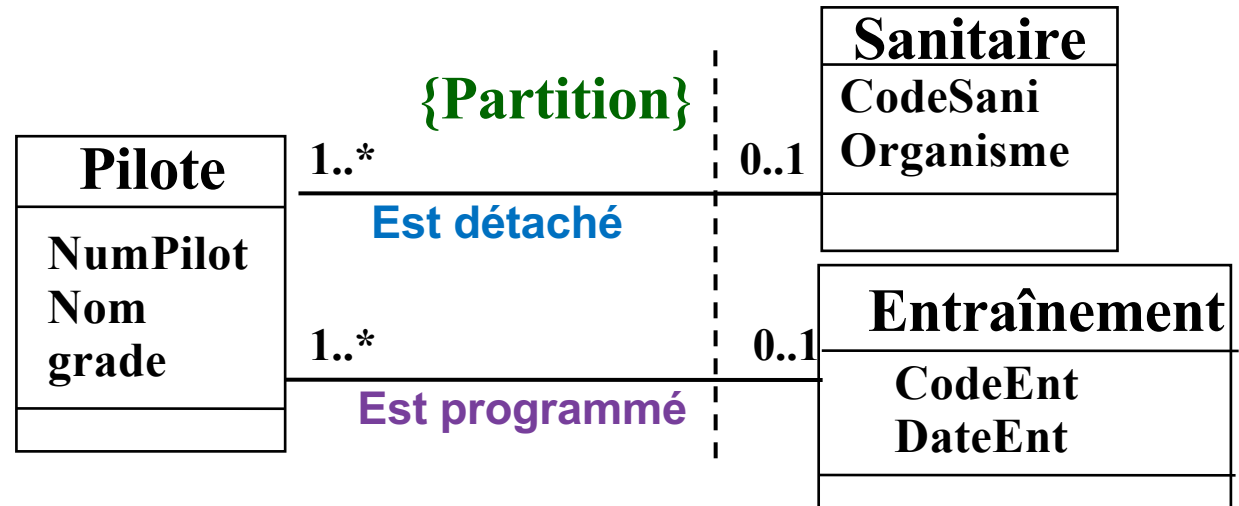
102, Durand, Annie, 19/10/2019, 3maois
105, Michel, Emeline, 17/01/2020, 5jours

2.4 Notions complémentaires : les contraintes d'intégrité

□ Les contraintes inter-association

a) **contrainte de partition** : tous les objets d'une classe participent obligatoirement à l'une des deux associations mais pas au deux.

Ex: des pilotes partent soit en mission sanitaire, soit en mission d'entraînement



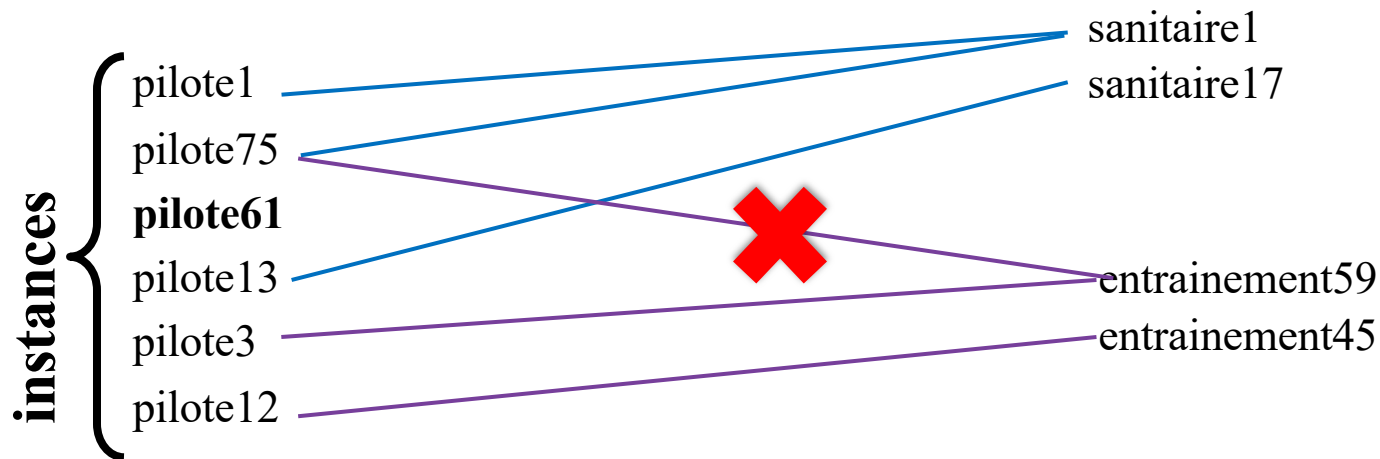
pilote61 doit être lié à une instance de Sanitaire ou de Entraînement

b) Contrainte d'exclusion

tous les objets d'une classe peuvent participer à l'une des deux associations, mais pas aux deux à la fois

Ex: des pilotes partent soit en mission sanitaire, soit en mission d'entraînement ou peuvent être au repos

Même notation et mettre **{Exclusion}**

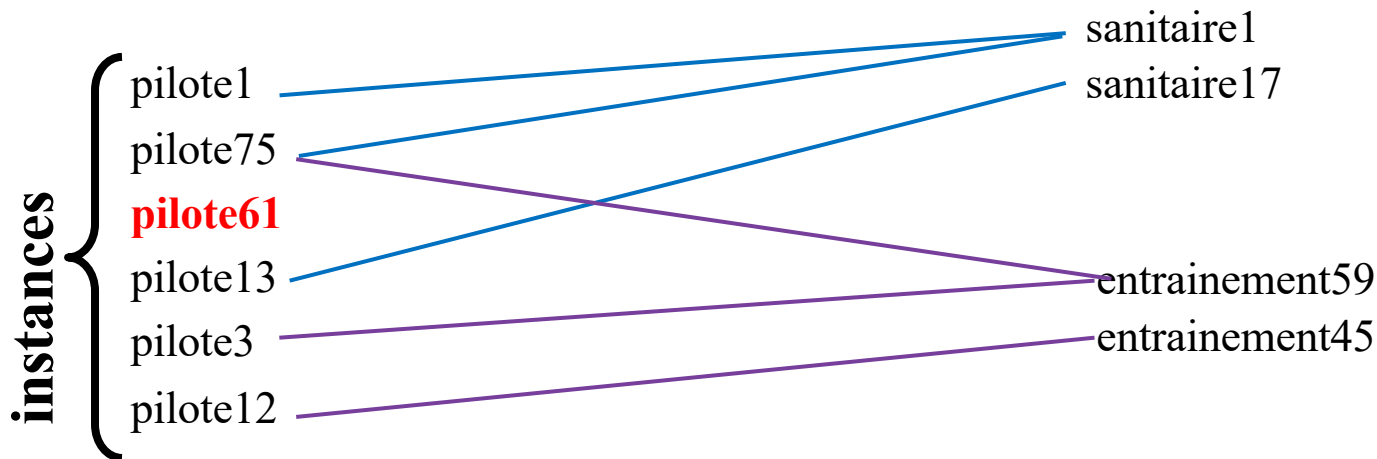


c) Contrainte de totalité

Tous les objets d'une classe participent au moins à une association

Ex: un pilote peut être affecté à la fois à une mission sanitaire et à une mission d'entraînement et que tous les pilotes participent au moins à une mission

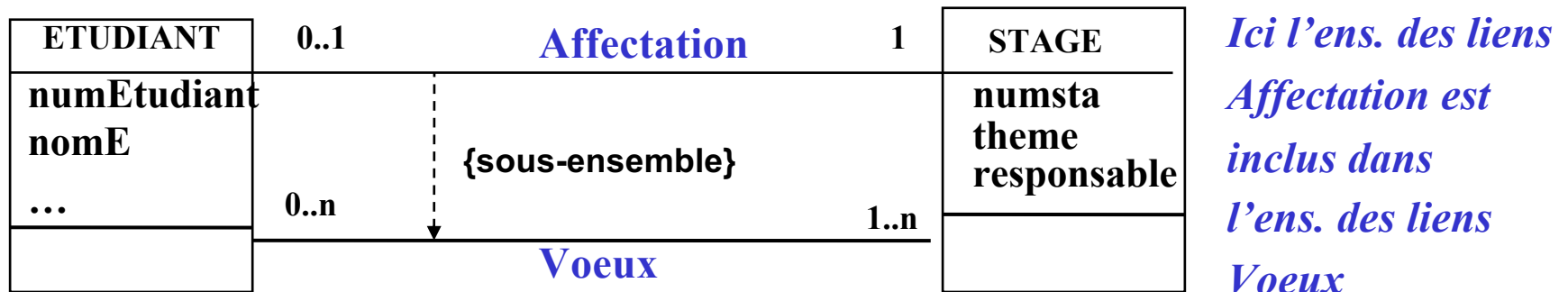
Même notation et mettre **{Totalité}**



pilote61 doit être lié
à une instance de
Sanitaire ou de
Entraînement

d) Contrainte d'inclusion

Toutes les occurrences d'une association doivent être incluses dans les occurrences d'une autre association



e) Contrainte de simultanéité

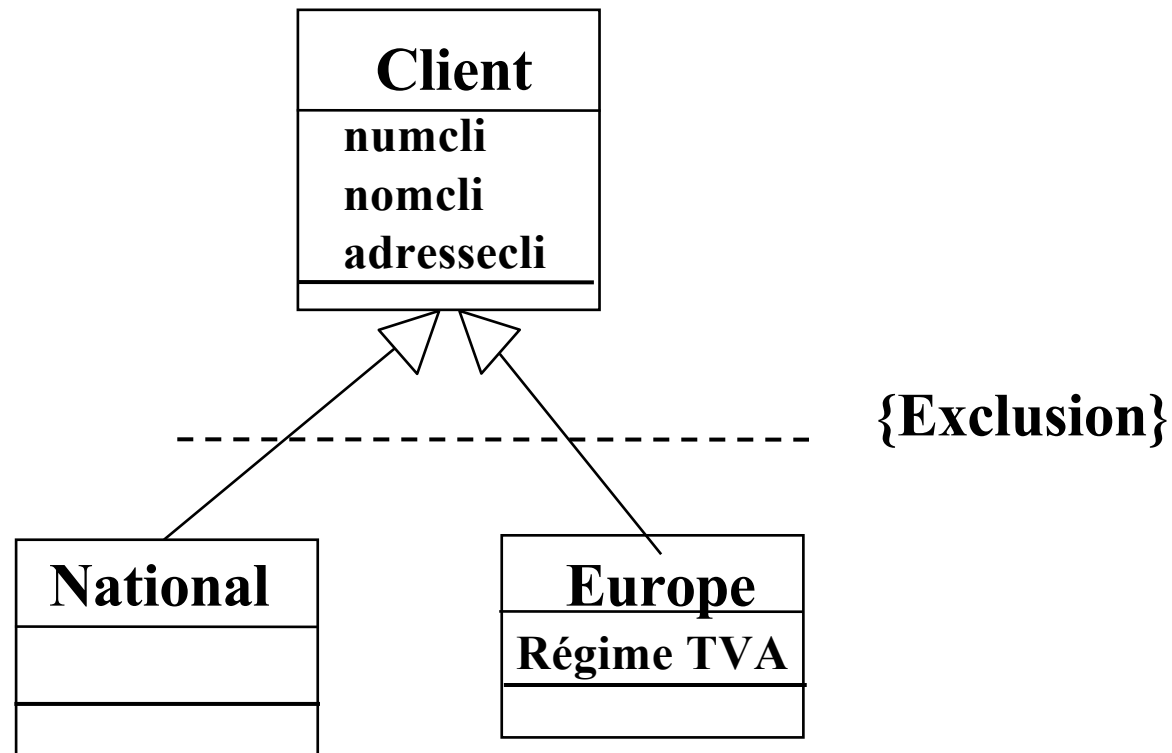
Si l'occurrence d'une classe participe à l'une des deux associations, elle participe également à l'autre

Ex: un pilote peut être au repos, mais s'il est affecté à une mission d'entraînement, il doit aussi être affecté à une mission sanitaire et vice versa

Même notation et mettre {Simultanéité}

b) Exclusivité (X)

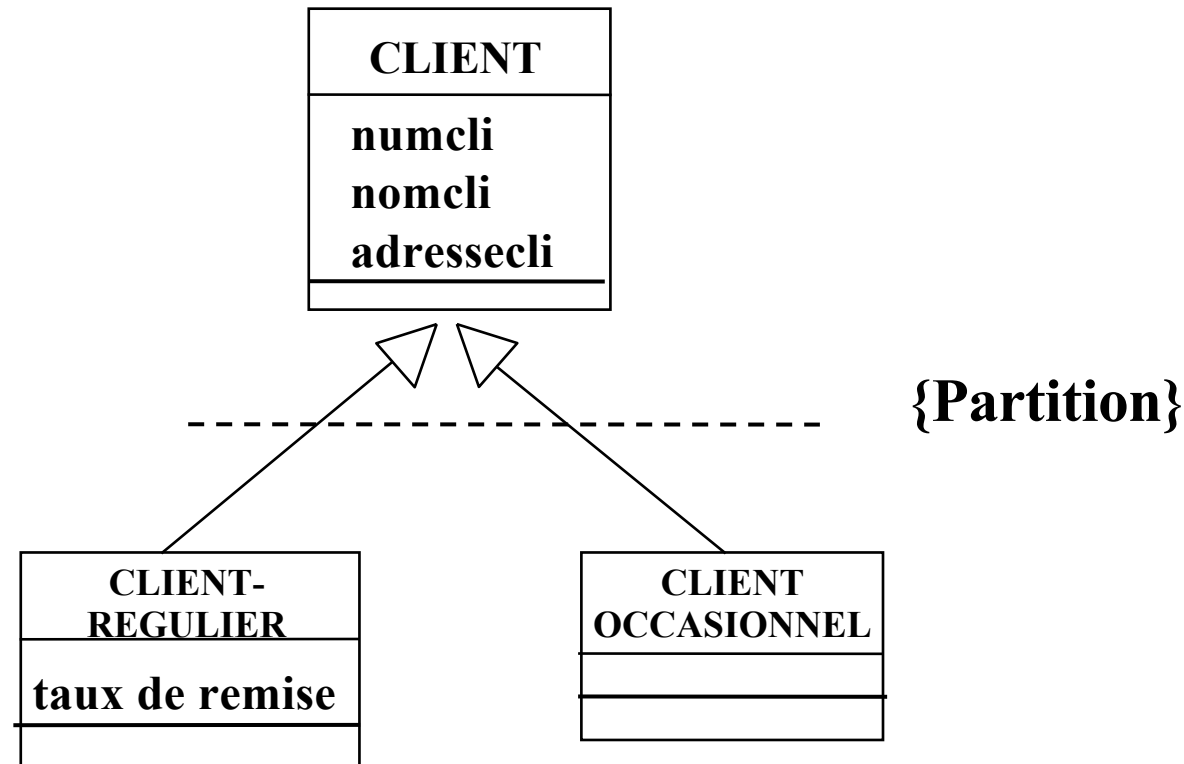
un client peut être l'un ou l'autre, ni l'un ni l'autre mais pas les deux à la fois



□ Contraintes sur l'héritage

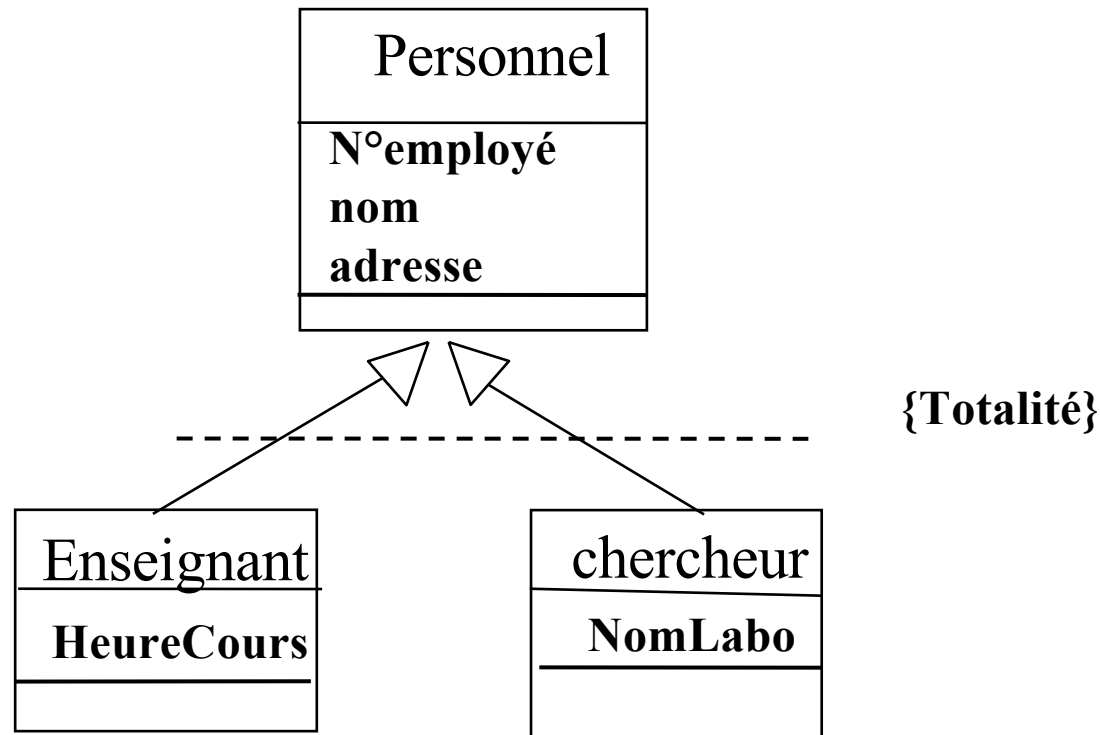
a) Exclusivité et totalité (XT)

tout client est soit un client régulier, soit un client occasionnel



c) Totalité (T)

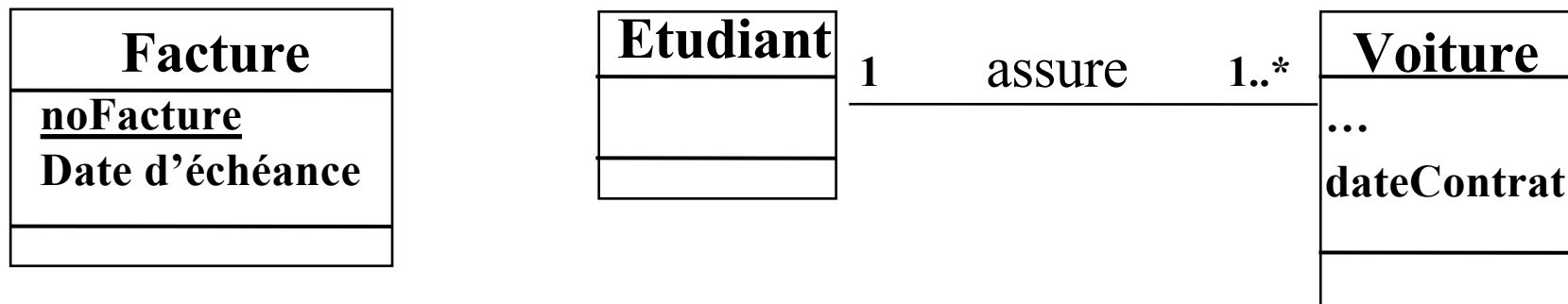
un employé peut être soit un enseignant, soit un chercheur, soit les deux à la fois



3) La modélisation du temps

Modélisation de propriétés à valeurs calendaires

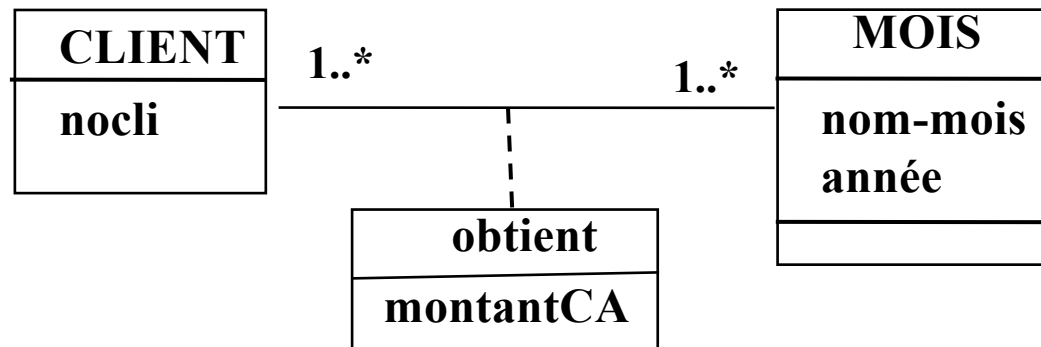
- date de naissance, date de livraison... sont représentées dans un schéma par des attributs



Le temps est considéré comme un attribut

2.4 Notions complémentaires : les contraintes d'intégrité

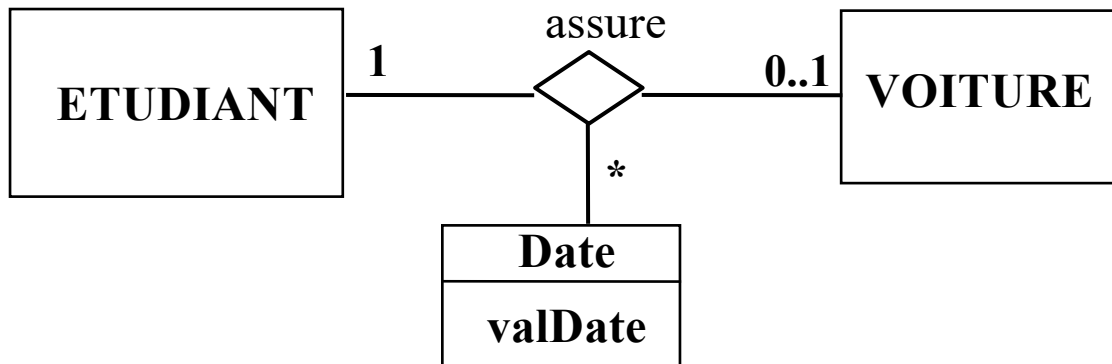
- **Modélisation de l'historique**
 - Le chiffre d'affaires mois par mois des clients



2.4 Notions complémentaires : les contraintes d'intégrité

- **Modélisation de l'historique**

- Un étudiant assure plusieurs fois la même voiture



Le temps est un élément discriminant, il est représenté dans une classe à part entière, appelée

Classe Temporelle

(et1, v1, 04/04/17)
(et1, v2, 01/12/19)
(et2, v1, 22/10/08)
(et1, v1, 13/01/19)
(et2, v2, 16/01/20)

2.5 Conclusion

Le modèle de classes d'objet sert à représenter une situation du monde réel, il représente l'un des résultats essentiels dans le développement des Systèmes d'Information (S.I.) car :

- il constitue une base d'accord et de dialogue entre informaticiens et utilisateurs
- il permet d'évaluer l'adéquation du système d'information informatisée aux besoins des gestionnaires
- il est le point de départ d'une conception détaillée

Pour concevoir un système d'information, on a besoin d'un modèle, d'une démarche de conception et d'un outil. Pour obtenir un bon modèle de classes d'objet, il faut respecter certaines règles.