

# Structure de donnée: UML et Relationnelle

## *Introduction à la modélisation*

Damien Pellier - Xavier Clerc

Institut de Géographie Alpine

Damien.Pellier@imag.fr, Xavier.Clerc@imag.fr



# Pourquoi modéliser ?

## Introduction

### ● Pourquoi modéliser ?

- Le processus de modélisation
- Modèle Conceptuel de Données (MCD)
- Plan

## Conception d'un Système d'Information

### Diagramme de classes - Diagramme d'objets

### Agrégation et composition

### Association qualifiée

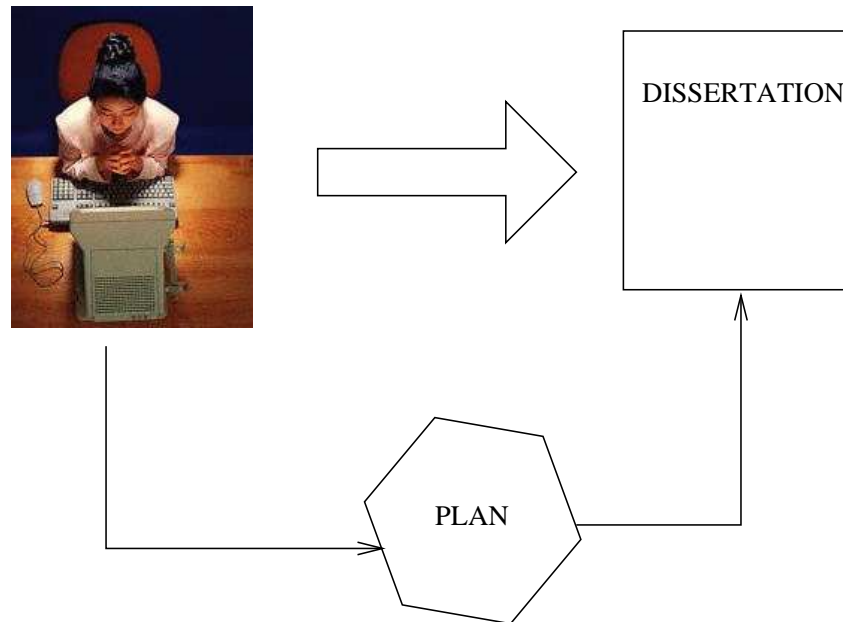
### Classe d'associations

### Traitement des hiérarchies de classes

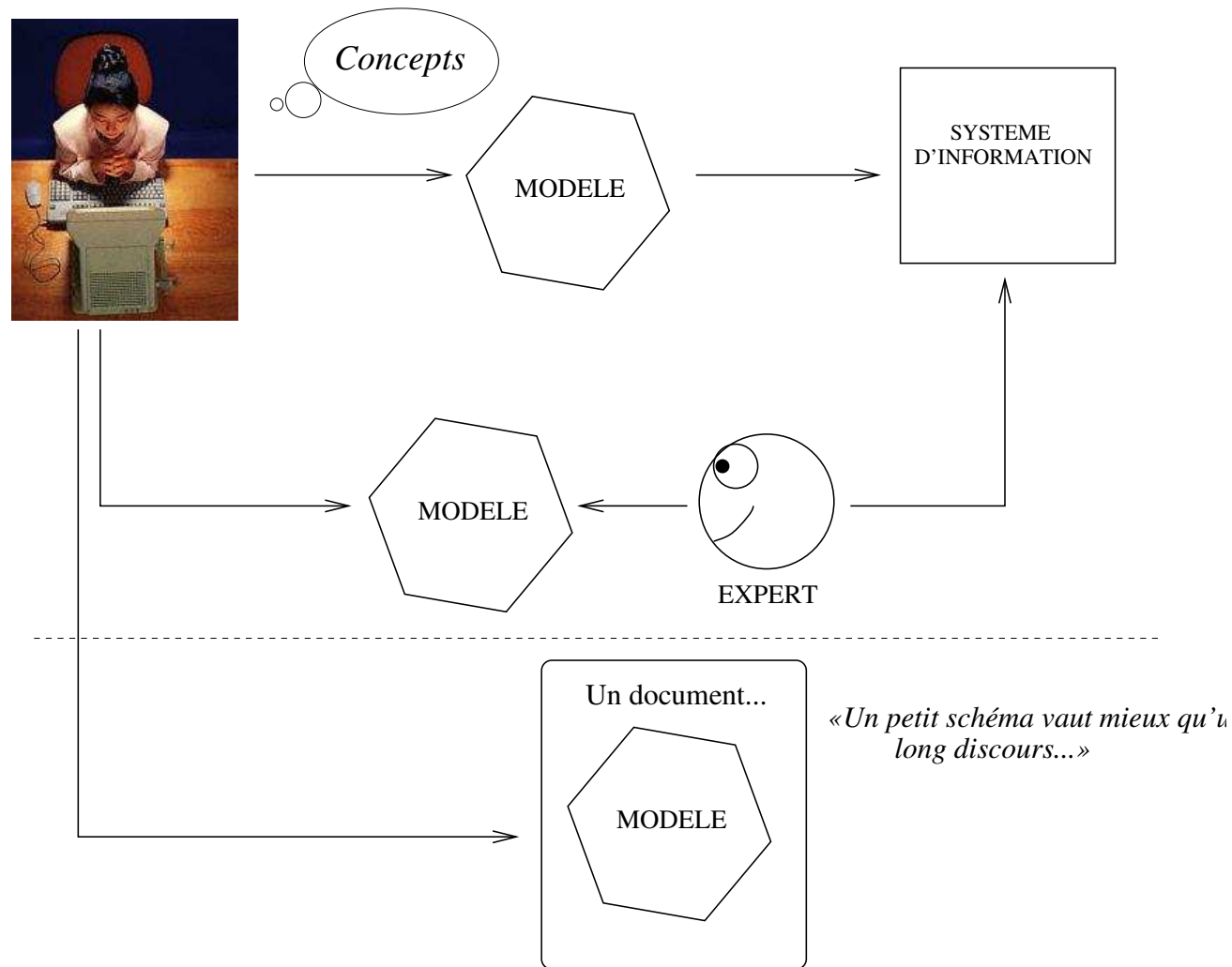
### Normalisation des relations

## Conclusion

- Résoudre un problème = réfléchir à ses «tenants et aboutissants» i.e. structurer sa pensée, représenter de façon abstraite le travail à réaliser  $\Rightarrow$  «archivage» de concepts ;
- Classer, ranger, structurer = préalable indispensable à l'exploitation des données ;
- La conception d'un SI n'échappe pas à ces principes ;



# Le processus de modélisation



## Introduction

- Pourquoi modéliser ?
- **Le processus de modélisation**
- Modèle Conceptuel de Données (MCD)
- Plan

## Conception d'un Système d'Information

### Diagramme de classes - Diagramme d'objets

### Agrégation et composition

### Association qualifiée

### Classe d'associations

### Traitement des hiérarchies de classes

### Normalisation des relations

## Conclusion



# Modèle Conceptuel de Données (MCD)

## Introduction

- Pourquoi modéliser ?
- Le processus de modélisation
- **Modèle Conceptuel de Données (MCD)**
- Plan

## Conception d'un Système d'Information

- Diagramme de classes -
- Diagramme d'objets

## Agrégation et composition

## Association qualifiée

## Classe d'associations

## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion

- Le MCD permet une description «statique» du système d'information ( $\neq$  au Modèle Conceptuel des Traitements) ;
- Les données sont décrites en termes d'entités, d'objets etc. (UML = Unified Modelling Language) ;
- Cette description graphique constitue un «langage commun» entre celui qui fait le cahier des charges et celui qui est chargé de sa réalisation.



# Plan

## Introduction

- Pourquoi modéliser ?
- Le processus de modélisation
- Modèle Conceptuel de Données (MCD)
- Plan

## Conception d'un Système d'Information

### Diagramme de classes - Diagramme d'objets

### Agrégation et composition

### Association qualifiée

### Classe d'associations

### Traitement des hiérarchies de classes

### Normalisation des relations

## Conclusion

- Conception d'un système d'information
- Références, liens, associations
- Diagramme de classes / Diagramme d'objets
- Agrégation et composition
- Association qualifiée
- Classes d'associations
- Contraintes d'intégrité
- Conversion d'un modèle en schéma relationnel
- Normalisation des relations



# Étapes de la conception

Introduction

Conception d'un Système  
d'Information

● Étapes de la conception

● Références, liens,  
associations

● Exemple préliminaire : la  
bibliothèque (1)

● Exemple préliminaire : la  
bibliothèque (2)

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Elle nécessite 4 étapes :

- Expression des besoins (cas d'utilisation, diagramme de séquences, etc.) ;
- Analyse fondée sur un MCD et un diagramme de séquences précis ;
- Conception = spécification des fonctionnalités (ou méthodes) + MCD complet ;
- Mise en œuvre (i.e. «programmation») + validation (i.e. tests).

# Références, liens, associations

## Introduction

## Conception d'un Système d'Information

### ● Étapes de la conception

### ● Références, liens, associations

### ● Exemple préliminaire : la bibliothèque (1)

### ● Exemple préliminaire : la bibliothèque (2)

## Diagramme de classes -

## Diagramme d'objets

## Agrégation et composition

## Association qualifiée

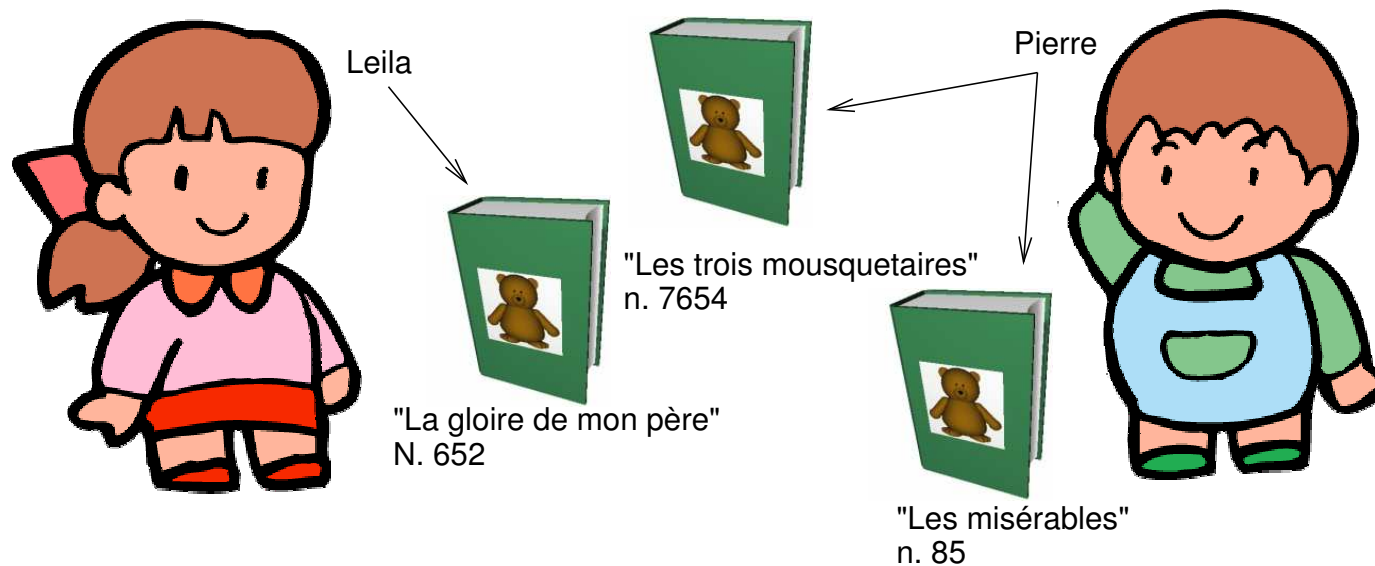
## Classe d'associations

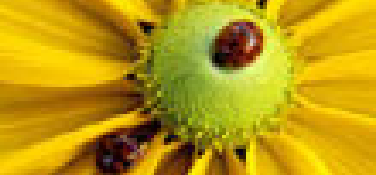
## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion

- Un *lien* caractérise le rôle que deux objets jouent l'un par rapport à l'autre : un objet fait *référence* à un autre objet ;
- Une *association* est le regroupement de liens entre mêmes classes et possédant la même sémantique ;
- Une association peut être *mono* ou *multivaluée* ; *partielle* ou *totale* (notion de cardinalité).





# Exemple préliminaire : la bibliothèque (1)

## Introduction

## Conception d'un Système d'Information

- Étapes de la conception
- Références, liens, associations
- Exemple préliminaire : la bibliothèque (1)
- Exemple préliminaire : la bibliothèque (2)

## Diagramme de classes - Diagramme d'objets

## Agrégation et composition

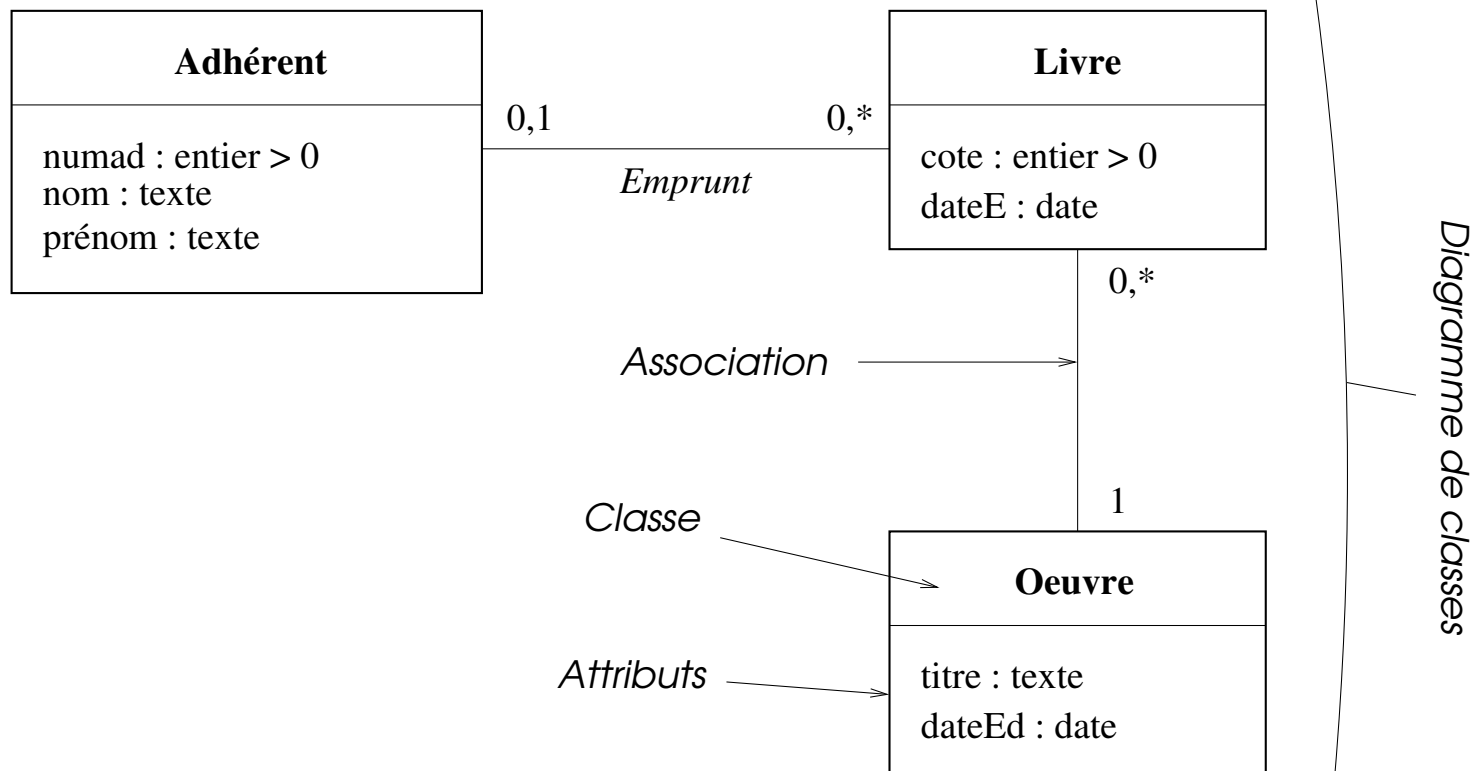
## Association qualifiée

## Classe d'associations

## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion







# Exemple préliminaire : la bibliothèque (2)

## Introduction

### Conception d'un Système d'Information

- Étapes de la conception
- Références, liens, associations
- Exemple préliminaire : la bibliothèque (1)
- Exemple préliminaire : la bibliothèque (2)

### Diagramme de classes - Diagramme d'objets

### Agrégation et composition

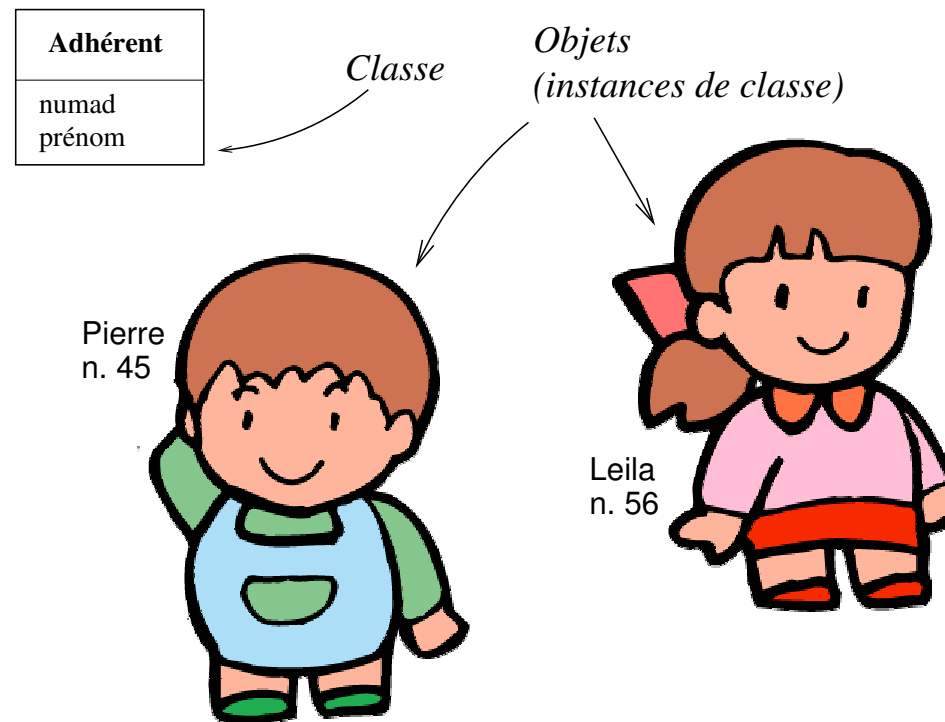
### Association qualifiée

### Classe d'associations

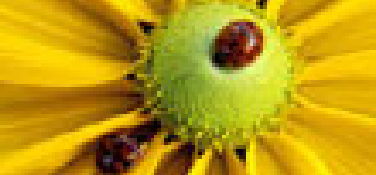
### Traitement des hiérarchies de classes

### Normalisation des relations

### Conclusion



Donner le diagramme de classes correspondant à l'application «Agence de voyages».



# Diagramme de classes : Agence de voyages

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

## ● Diagramme de classes : Agence de voyages

## ● Diagramme d'objets - de classes : Polygone (1)

## ● Diagramme d'objets - de classes : Polygone (2)

## ● Exercice polygone (1)

## ● Exercice polygone (2)

## ● Exercice le cadastre (1)

## ● Exercice le cadastre (2)

## ● Exercice le cadastre (3)

## Agrégation et composition

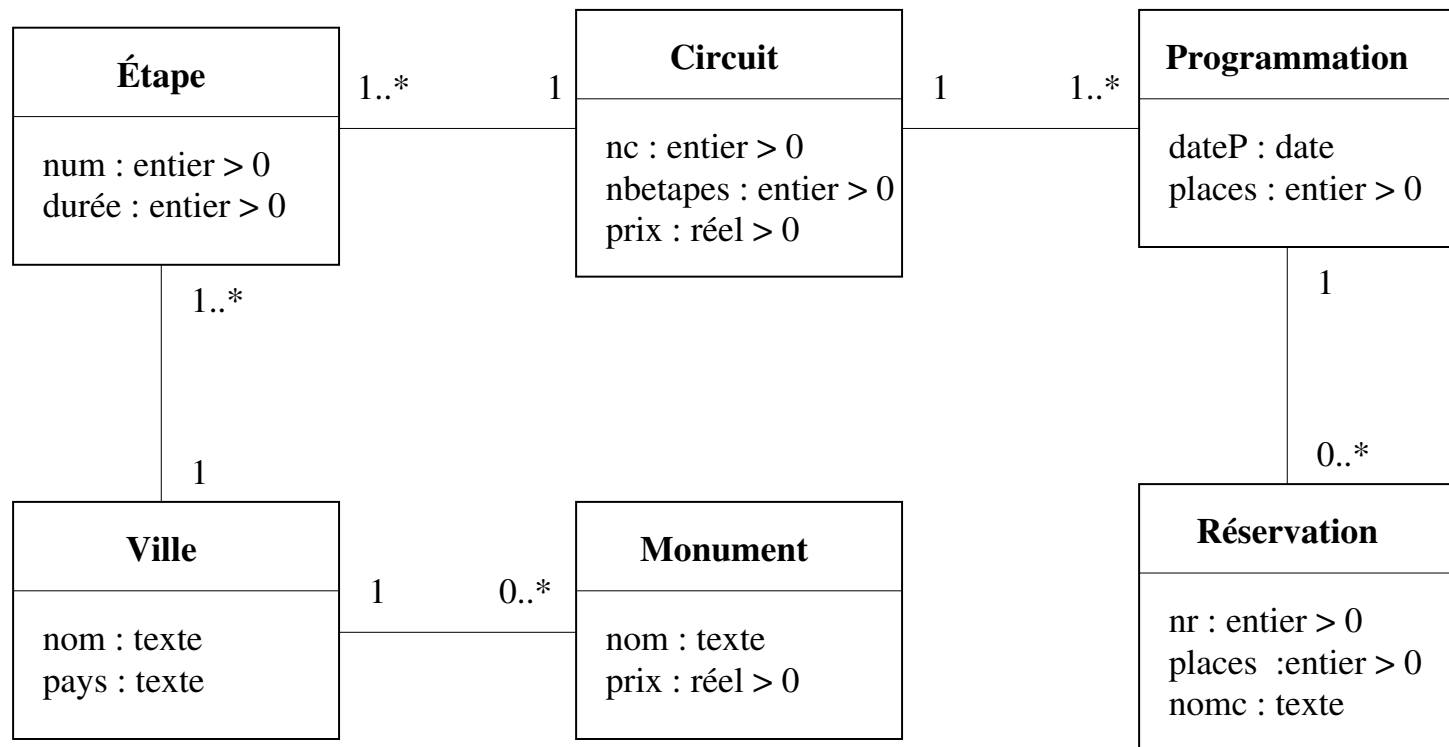
## Association qualifiée

## Classe d'associations

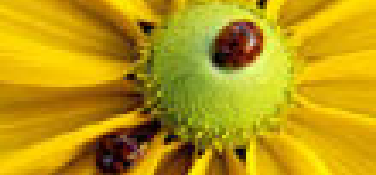
## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion



Les relations entre les classes permettent de définir les contraintes d'intégrité (cf. Access)



# Diagramme d'objets - de classes : Polygone (1)

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

- Diagramme de classes : Agence de voyages
- Diagramme d'objets - de classes : Polygone (1)
- Diagramme d'objets - de classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

## Agrégation et composition

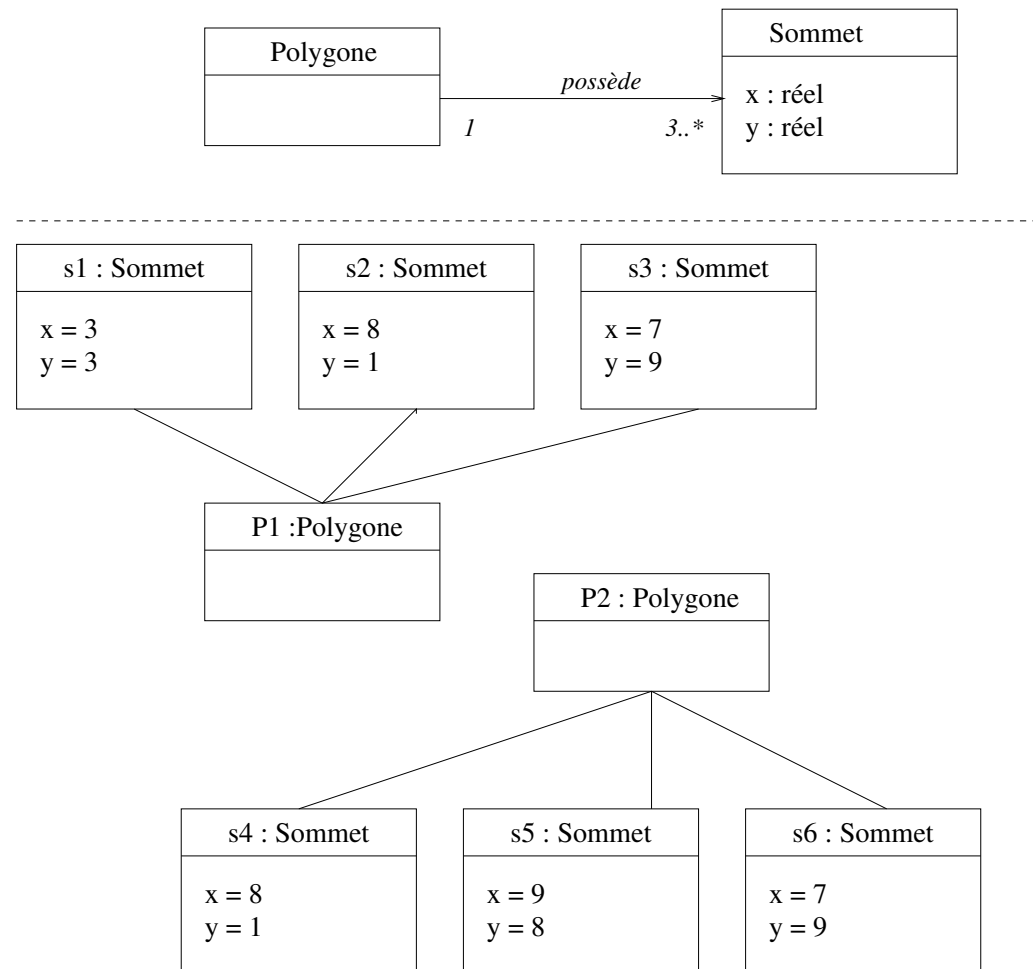
## Association qualifiée

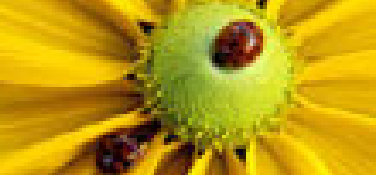
## Classe d'associations

## Traitement des hiérarchies de classes

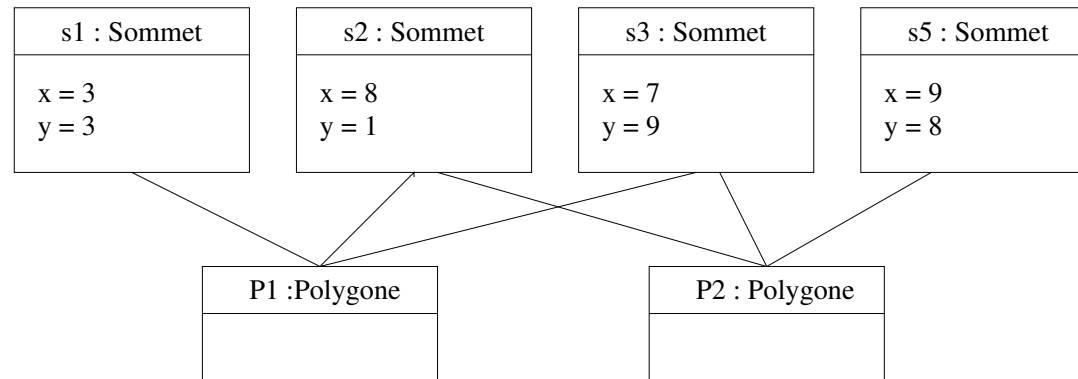
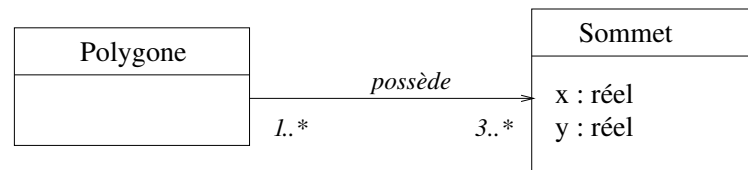
## Normalisation des relations

## Conclusion





# Diagramme d'objets - de classes : Polygone (2)



Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

- Diagramme de classes :  
Agence de voyages
- Diagramme d'objets - de  
classes : Polygone (1)
- Diagramme d'objets - de  
classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

# Exercice polygone (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

- Diagramme de classes :  
Agence de voyages
- Diagramme d'objets - de  
classes : Polygone (1)
- Diagramme d'objets - de  
classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

Agrégation et composition

Association qualifiée

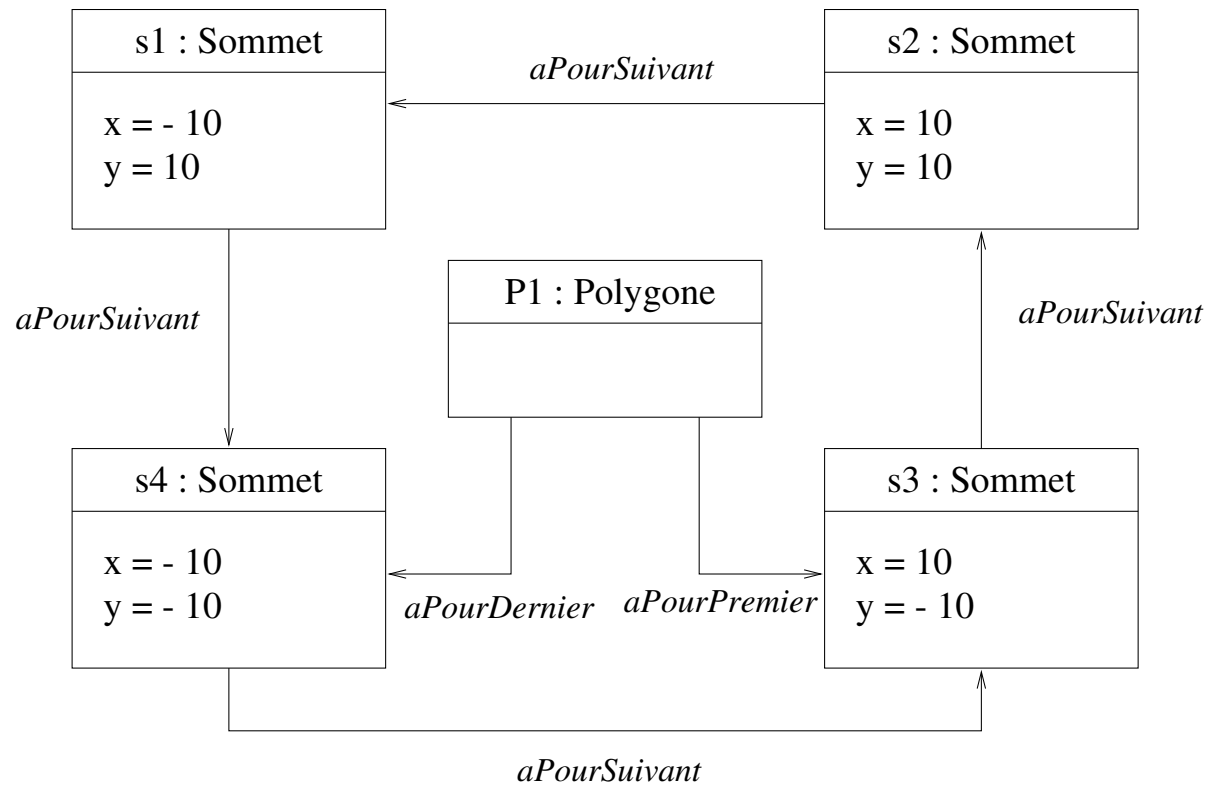
Classe d'associations

Traitement des hiérarchies de  
classes

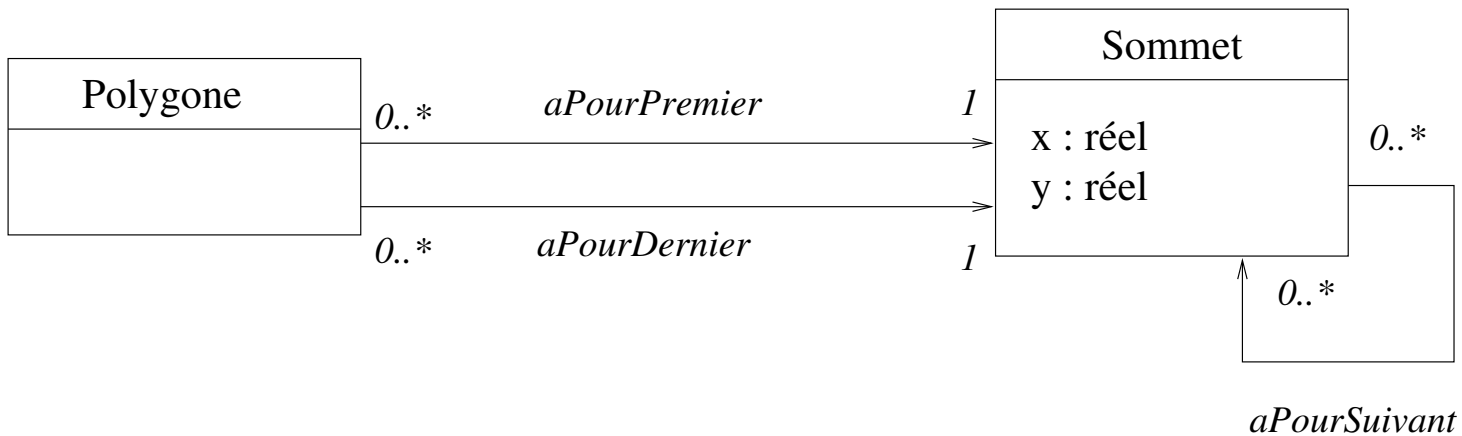
Normalisation des relations

Conclusion

Donner le diagramme de classes correspondant à :



# Exercice polygone (2)



Quelles méthodes pour la classe polygone ?

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

- Diagramme de classes :  
Agence de voyages
- Diagramme d'objets - de  
classes : Polygone (1)
- Diagramme d'objets - de  
classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)

- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

Agrégation et composition

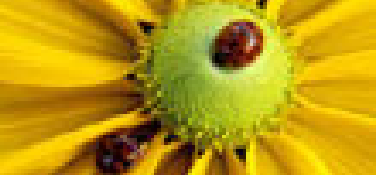
Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion



# Exercice le cadastre (1)

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

- Diagramme de classes : Agence de voyages
- Diagramme d'objets - de classes : Polygone (1)
- Diagramme d'objets - de classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

## Agrégation et composition

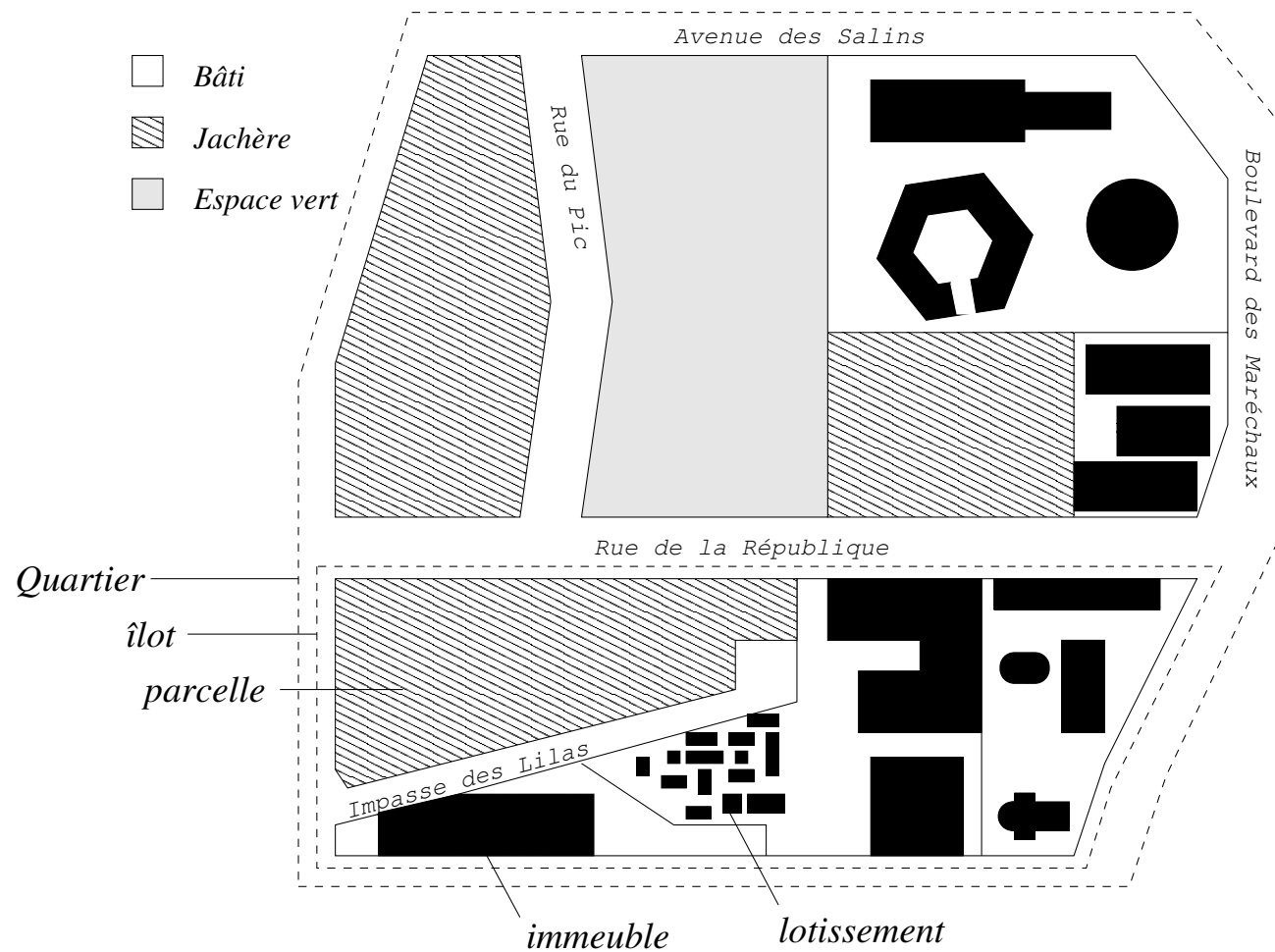
## Association qualifiée

## Classe d'associations

## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion





# Exercice le cadastre (2)

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

- Diagramme de classes : Agence de voyages
- Diagramme d'objets - de classes : Polygone (1)
- Diagramme d'objets - de classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- **Exercice le cadastre (2)**
- Exercice le cadastre (3)

## Agrégation et composition

## Association qualifiée

## Classe d'associations

## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion

- une parcelle est identifiée par un code. On considère qu'elle peut être bâtie, être en jachère ou contenir des espaces verts. Pour chaque parcelle, on désire connaître ces différents propriétaires au cours du temps ;
- les propriétaires ont un nom et une adresse ;
- pour une parcelle donnée, on veut pouvoir déterminer quelles sont les parcelles attenantes ;
- un quartier est délimité par plusieurs rues ;
- les rues ne traversent jamais une parcelle...



# Exercice le cadastre (3)

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

- Diagramme de classes : Agence de voyages
- Diagramme d'objets - de classes : Polygone (1)
- Diagramme d'objets - de classes : Polygone (2)
- Exercice polygone (1)
- Exercice polygone (2)
- Exercice le cadastre (1)
- Exercice le cadastre (2)
- Exercice le cadastre (3)

## Agrégation et composition

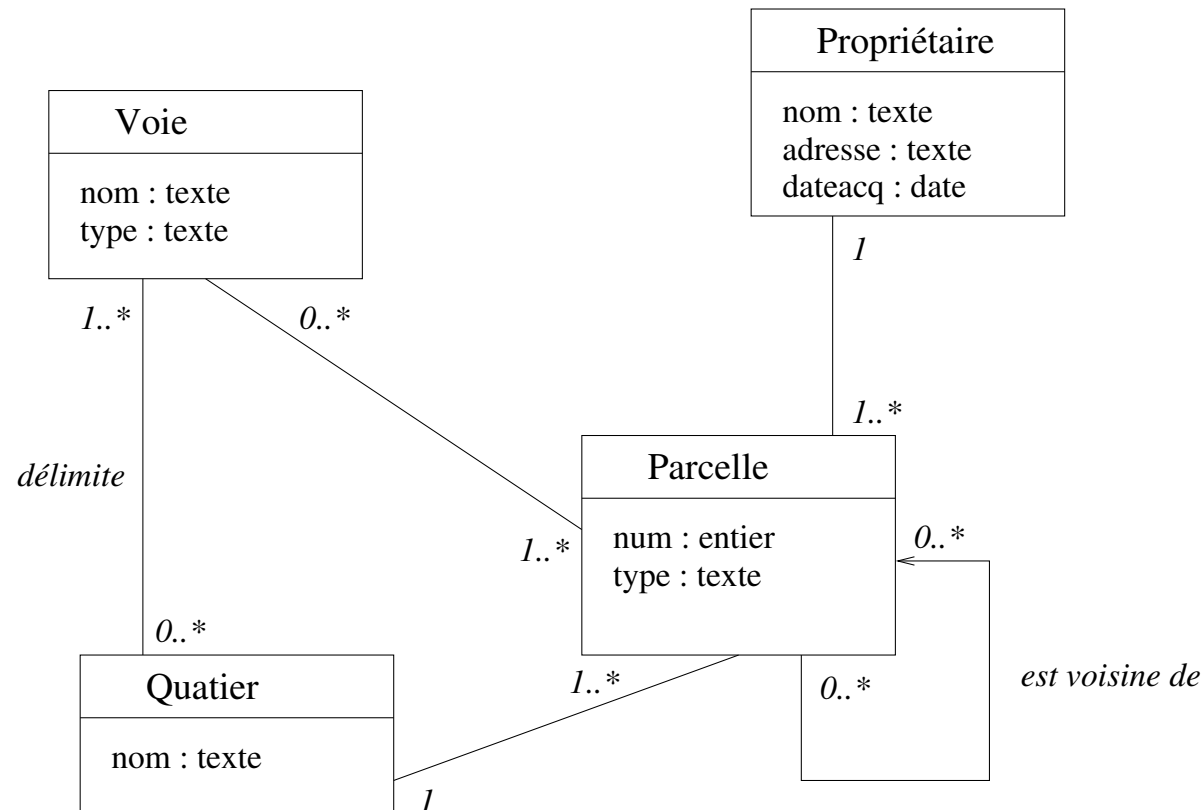
## Association qualifiée

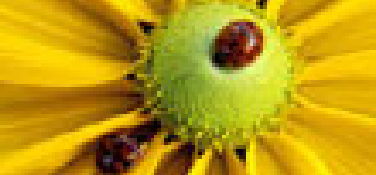
## Classe d'associations

## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion





# Définitions

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

● Définitions

- Association qualifiée
- Classe d'associations
- Exercice (1)
- Exercice (2)
- Conversion d'un modèle en  
schéma relationnel

Association qualifiée

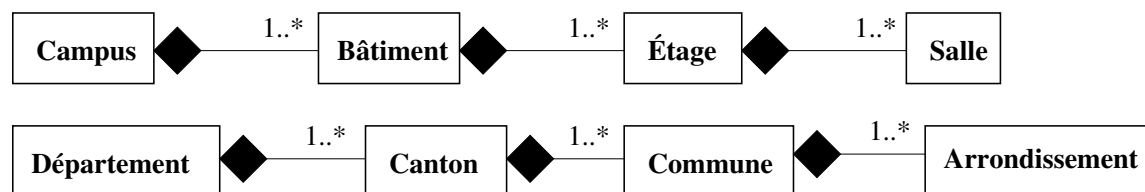
Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

- Agrégation : permet de représenter des associations du type « est une partie de... » Par exemple, un chapitre contient des sections qui elles-mêmes contiennent des sous-sections, etc.
- Composition : c'est une agrégation dont une sous-partie ne peut être partagée par plusieurs parties ; généralement, « l'accès » a une sous-partie n'est possible qu'à partir d'une seule partie et la suppression d'une partie entraîne la suppression de toutes ses sous-parties.



# Association qualifiée

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

● Définitions

● Association qualifiée

● Classe d'associations

● Exercice (1)

● Exercice (2)

● Conversion d'un modèle en  
schéma relationnel

Association qualifiée

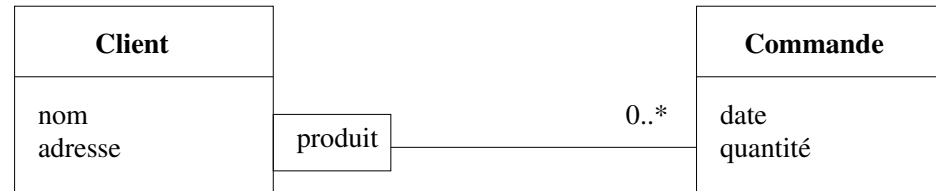
Classe d'associations

Traitement des hiérarchies de  
classes

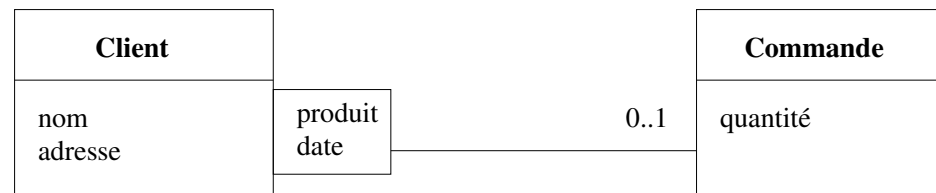
Normalisation des relations

Conclusion

Un ou des attributs d'une classe sont utilisés comme qualificatifs d'une association lorsque leurs valeurs permettent d'identifier un ensemble d'objets de la classe, dans une association donnée.



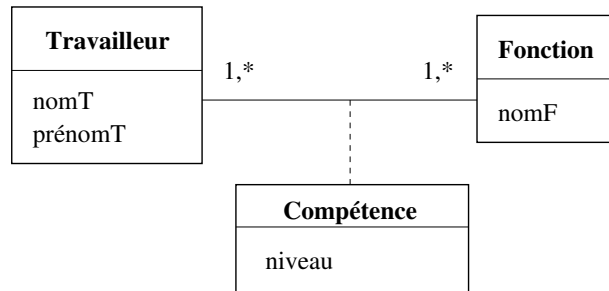
Ici, l'historique des commandes d'un client donné est indexé par rapport à un produit donné.



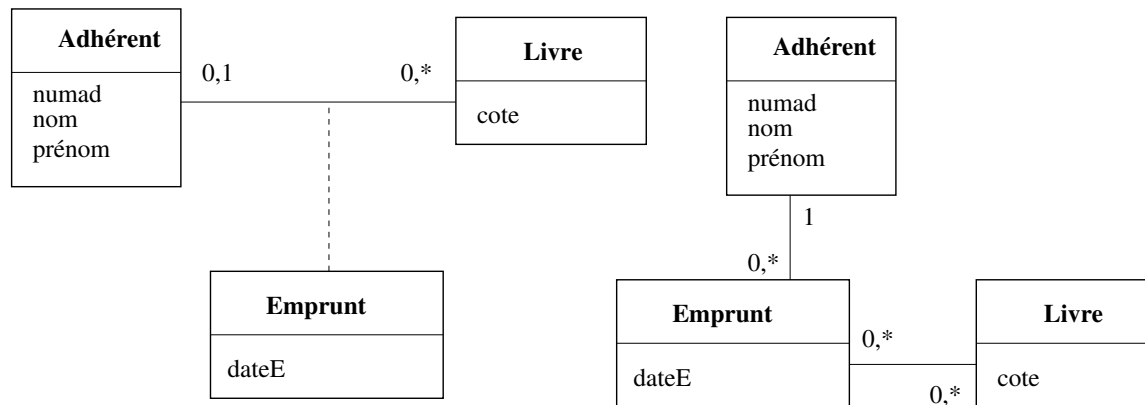
Dans ce cas, on représente la notion d'identifiant.

# Classe d'associations

Une association peut aussi être caractérisée par des attributs et des méthodes :



Exercice : Un des diagrammes ci-dessous est incorrecte. Lequel et pourquoi ?



Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

● Définitions

● Association qualifiée

● Classe d'associations

● Exercice (1)

● Exercice (2)

● Conversion d'un modèle en  
schéma relationnel

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion



# Exercice (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

- Définitions
- Association qualifiée
- Classe d'associations
- Exercice (1)
- Exercice (2)
- Conversion d'un modèle en  
schéma relationnel

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Un office du tourisme souhaite réaliser un système d'information permettant d'informer à travers un site Web ses visiteurs sur les films à l'affiche...

- un cinéma est identifié par son nom. On désire gérer son adresse et son numéro de téléphone. L'adresse du cinéma est donnée sous la forme d'un numéro dans une rue et du nom de la rue. Un cinéma dispose de plusieurs salles numérotées et chacune a une certaine capacité ;
- un film est identifié par son titre. On connaît son réalisateur et ses acteurs principaux. Un film a un ou plusieurs genres (e.g. comédie, policier, etc.) ;
- une séance a lieu dans une salle donnée. Elle est programmée à une heure donnée pour tous les jours de la semaine.



# Exercice (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

- Définitions
- Association qualifiée
- Classe d'associations
- Exercice (1)
- Exercice (2)
- Conversion d'un modèle en  
schéma relationnel

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

On souhaite modifier ce système afin de tenir compte des notions suivantes :

- une séance est programmée à une heure donnée d'un jour donné. Le programme des séances est ainsi différent selon les jours de la semaine ;
- un film peut être projeté en version française ou originale. Selon le cas, son titre est différent bien qu'il s'agisse du même film.



# Conversion d'un modèle en schéma relationnel

- Introduction
- Conception d'un Système d'Information
- Diagramme de classes - Diagramme d'objets
- Agrégation et composition
  - Définitions
  - Association qualifiée
  - Classe d'associations
  - Exercice (1)
  - Exercice (2)
  - Conversion d'un modèle en schéma relationnel
- Association qualifiée
- Classe d'associations
- Traitement des hiérarchies de classes
- Normalisation des relations
- Conclusion

À partir d'un diagramme de classes décrit en UML, on souhaite passer au schéma relationnel correspondant. L'objectif est donc de :

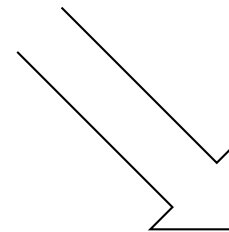
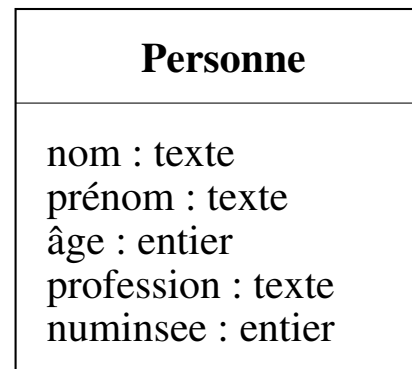
- représenter toutes les informations présentes dans le diagramme des classes (classes et associations) ;
- éviter les redondances (répétition des mêmes associations) ;
- limiter le nombre de relations (schéma complexe) ;
- limiter les valeurs absentes (e.g., les valeurs NULL).

Dans ce qui suit, on désigne par la lettre C une classe, X ses attributs et K un identifiant.



# Classe non associative

Une relation pour la classe. Les attributs de la relation sont les attributs (non dérivés) de la classe. Les attributs dérivés sont représentés une formule.



LesPersonnes(numinsee, nom, prenom, age, profession)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

● Classe non associative

● Association qualifiée (1)

● Classe associative (1)

● Classe associative (2)

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion



# Association qualifiée (1)

## Introduction

## Conception d'un Système d'Information

## Diagramme de classes - Diagramme d'objets

## Agrégation et composition

## Association qualifiée

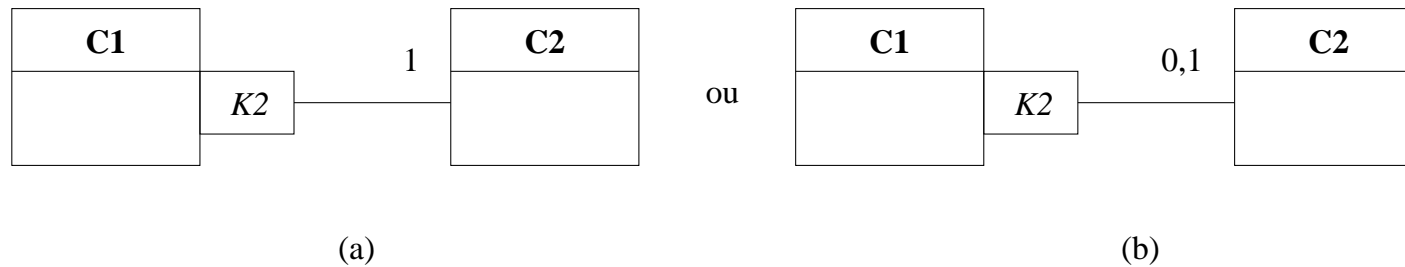
- Classe non associative
- Association qualifiée (1)
- Classe associative (1)
- Classe associative (2)

## Classe d'associations

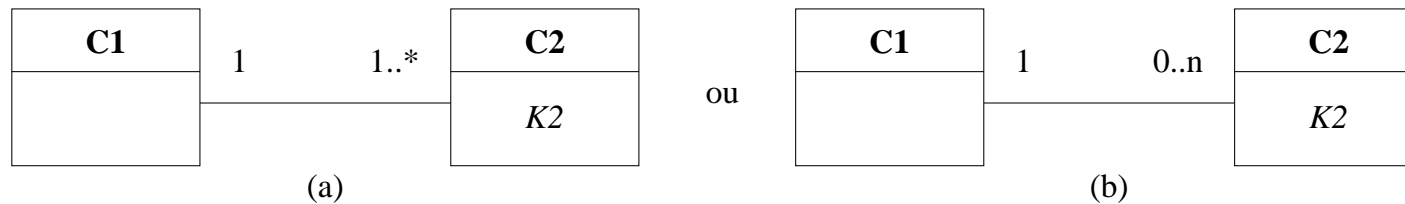
## Traitement des hiérarchies de classes

## Normalisation des relations

## Conclusion



Il faut se noter que cette association sans qualification est équivalente à :



# Classe associative (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

- Classe non associative
- Association qualifiée (1)
- **Classe associative (1)**
- Classe associative (2)

Classe d'associations

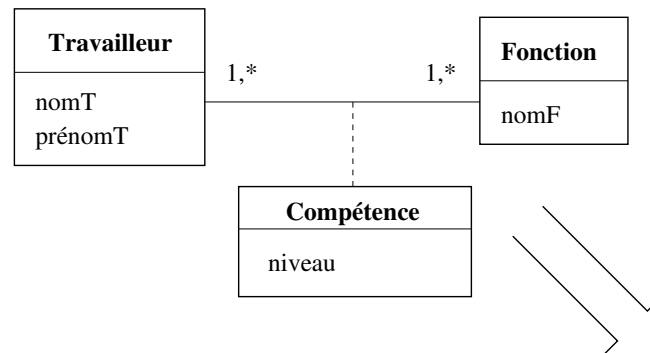
Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Soit R1, R2 les relations dédiées aux classes associées respectivement munies des identifiants K1 et K2. On crée une relation R pour la classe/association.

Exemple :



LesCompétences(nomF, nomT, prenomT, niveau)

LesFonctions(nomF)

LesTravailleurs(nomT, prenomT)



# Classe associative (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

- Classe non associative
- Association qualifiée (1)
- Classe associative (1)
- Classe associative (2)

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Les contraintes d'intégrité référentielles sont :

■  $R[K1] \subseteq R1[K1]$  ;

■  $R[K2] \subseteq R2[K2]$ .

Dans le cas d'association totale, ces contraintes sont plus fortes. Par exemple, supposons que l'association vers la classe C1 soit totale :  $R[K1] = R1[K1]$ .

Dans l'exemple précédent, on obtiendra donc :

■  $\text{LesCompétences}[\text{nomT}, \text{prénomT}] = \text{LesTravailleurs}[\text{nomT}, \text{prénomT}]$  ;

■  $\text{LesCompétences}[\text{nomF}] = \text{LesFonctions}[\text{nomF}]$ .



# Association binaire simple

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

● Association binaire simple

● 1er cas : l'association est  
«multi-multi» (1)

● 1er cas : l'association est  
«multi-multi» (2)

● 2e cas : l'association est  
«mono-multi» (1)

● 2e cas : l'association est  
«mono-multi» (2)

● 3e cas : l'association est  
«mono-mono» (1)

● 3e cas : l'association est  
«mono-mono» (2)

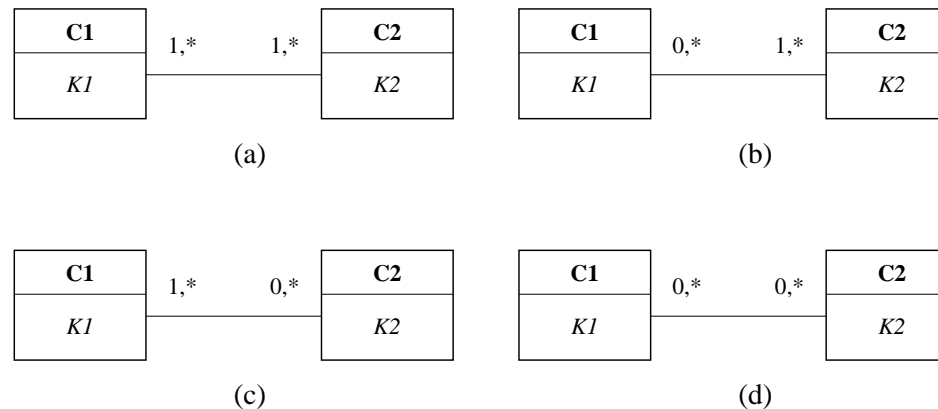
Traitement des hiérarchies de  
classes

Normalisation des relations

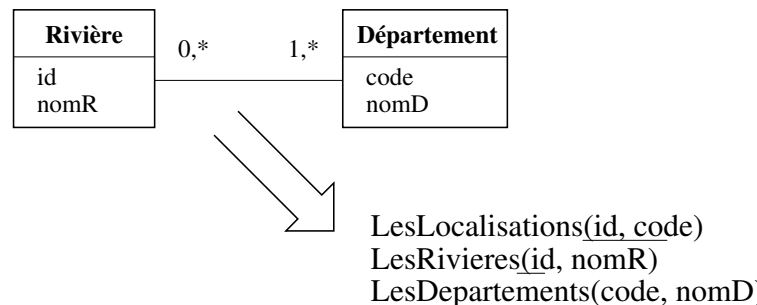
Conclusion

On distingue plusieurs cas selon la cardinalité de l'association :  
Soit C1 et C2 les classes associées. R1 est issue de C1 et R2 de C2. K1 est l'identifiant de R1 et K2 celui de R2.

# 1er cas : l'association est «multi-multi» (1)



Dans tous les cas, une relation  $R$  est définie pour l'association.  $K1$  et  $K2$  sont les attributs et l'identifiant de  $R$ .



*Remarque* :  $R$  est en quelque sorte une «table de liaison».

- Introduction
- Conception d'un Système d'Information
- Diagramme de classes - Diagramme d'objets
- Agrégation et composition
- Association qualifiée
- Classe d'associations
  - Association binaire simple
  - 1er cas : l'association est «multi-multi» (1)
  - 1er cas : l'association est «multi-multi» (2)
  - 2e cas : l'association est «mono-multi» (1)
  - 2e cas : l'association est «mono-multi» (2)
  - 3e cas : l'association est «mono-mono» (1)
  - 3e cas : l'association est «mono-mono» (2)
- Traitement des hiérarchies de classes
- Normalisation des relations
- Conclusion



# 1er cas : l'association est «multi-multi» (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

- Association binaire simple
- 1er cas : l'association est «multi-multi» (1)
- 1er cas : l'association est «multi-multi» (2)
- 2e cas : l'association est «mono-multi» (1)
- 2e cas : l'association est «mono-multi» (2)
- 3e cas : l'association est «mono-mono» (1)
- 3e cas : l'association est «mono-mono» (2)

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Dans tous les cas, les contraintes référentielles sont :

■  $R[K1]$  inclus dans  $R1[K1]$  ;

■  $R[K2]$  inclus dans  $R2[K2]$

Selon les cas, on ajoute d'autres contraintes :

■ (a) :  $R1[K1] = R[K1]$  et  $R2[K2] = R[K2]$  ;

■ (b) :  $R1[K1]$  inclus dans  $R[K1]$  (et donc  $R1[K1] = R[K1]$ ) ;

■ (c) :  $R2[K2]$  inclus dans  $R[K2]$  (et donc  $R2[K2] = R[K2]$ )

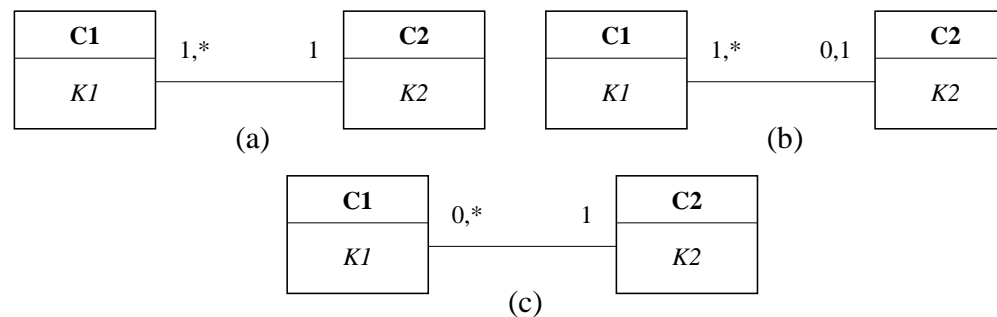
Dans l'exemple précédent, on obtiendra donc :

■  $\text{LesLocalisations}[\text{id}] = \text{LesRivières}[\text{id}]$  ;

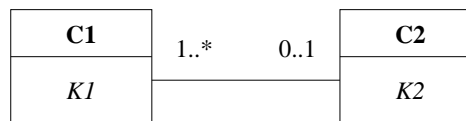
■  $\text{LesLocalisations}[\text{code}] \subseteq \text{LesDépartements}[\text{code}]$ .

## 2e cas : l'association est «mono-multi» (1)

- On ajoute K2 dans la relation R1. Valuation obligatoire pour K2 dans R1 et  $R1[K2] = R2[K2]$



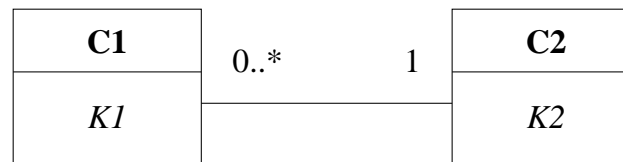
- Pour éviter les valeurs absentes dans R1, une autre relation R est définie pour l'association. K1 et K2 sont les attributs et K1 est l'identifiant de R. Les contraintes référentielles sont :  $R[K1]$  inclus dans  $R1[K1]$  et  $R[K2]$  inclus dans  $R2[K2]$



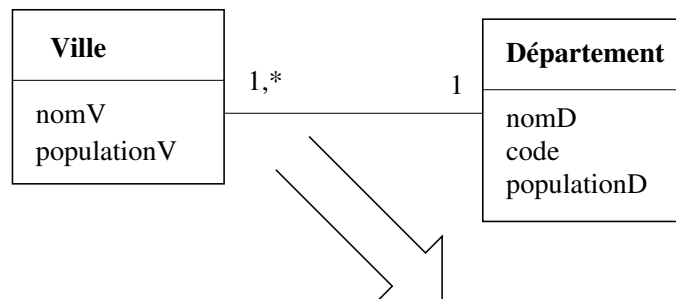
- Association binaire simple
- 1er cas : l'association est «multi-multi» (1)
- 1er cas : l'association est «multi-multi» (2)
- 2e cas : l'association est «mono-multi» (1)
- 2e cas : l'association est «mono-multi» (2)
- 3e cas : l'association est «mono-mono» (1)
- 3e cas : l'association est «mono-mono» (2)

## 2e cas : l'association est «mono-multi» (2)

- On ajoute K2 dans la relation R1. Valuation obligatoire de K2 dans R1 ;  $R1[K2] \subseteq R2[K2]$



Prenons un exemple :



LesDepartements(code, nomD, populationD)

LesVilles(nomV, populationV, code)

LesDepartements[code] = LesVilles[code]

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

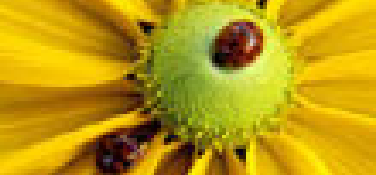
- Association binaire simple
- 1er cas : l'association est «multi-multi» (1)
- 1er cas : l'association est «multi-multi» (2)
- 2e cas : l'association est «mono-multi» (1)
- 2e cas : l'association est «mono-multi» (2)
- 3e cas : l'association est «mono-mono» (1)
- 3e cas : l'association est «mono-mono» (2)

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion





# 3e cas : l'association est «mono-mono» (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

- Association binaire simple
- 1er cas : l'association est «multi-multi» (1)
- 1er cas : l'association est «multi-multi» (2)
- 2e cas : l'association est «mono-multi» (1)
- 2e cas : l'association est «mono-multi» (2)
- 3e cas : l'association est «mono-mono» (1)
- 3e cas : l'association est «mono-mono» (2)

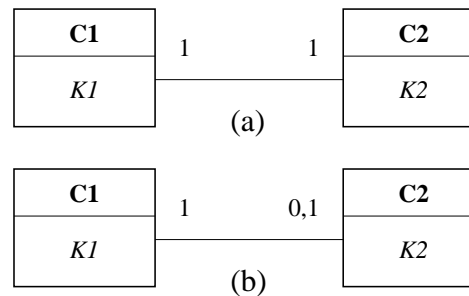
Traitement des hiérarchies de  
classes

Normalisation des relations

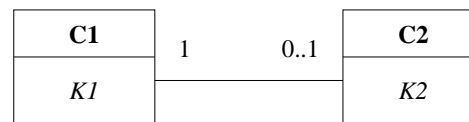
Conclusion

## ■ Deux solutions sont possibles :

- ◆ Soit on ajoute  $K1$  dans  $R2$  et  $R2[K1] = R1[K1]$ ,  $K1$  est un autre identifiant dans  $R2$  ;
- ◆ Soit on ajoute  $K2$  dans  $R1$  et  $R1[K2] = R2[K2]$ ,  $K2$  est un autre identifiant dans  $R1$ .



## ■ On ajoute $K1$ dans $R2$ et $R2[K1] \subseteq R1[K1]$ , $K1$ est un autre identifiant dans $R2$ .



# 3e cas : l'association est «mono-mono» (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

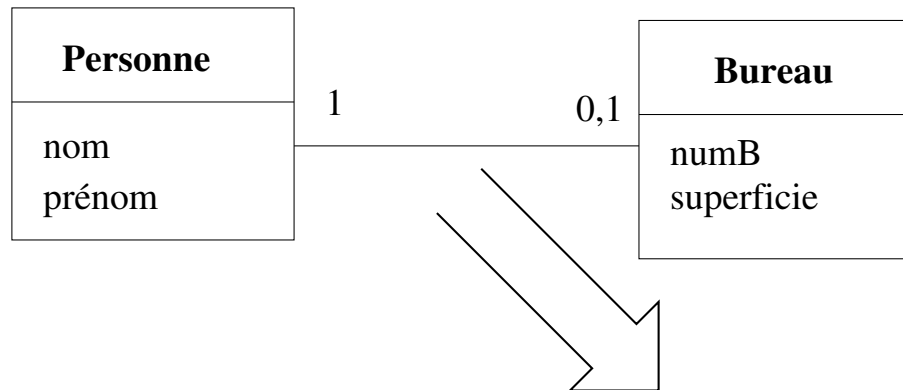
- Association binaire simple
- 1er cas : l'association est «multi-multi» (1)
- 1er cas : l'association est «multi-multi» (2)
- 2e cas : l'association est «mono-multi» (1)
- 2e cas : l'association est «mono-multi» (2)
- 3e cas : l'association est «mono-mono» (1)
- 3e cas : l'association est «mono-mono» (2)

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

Prenons un exemple :



LesBureaux(numB, superficie, nom, prenom)

LesPersonnes(nom, prenom)

LesBureaux[nom, prénom]  $\subseteq$  LesPersonnes[nom, prenom]



# Traitement des hiérarchies de classes (1)

Introduction

---

Conception d'un Système d'Information

---

Diagramme de classes -  
Diagramme d'objets

---

Agrégation et composition

---

Association qualifiée

---

Classe d'associations

---

Traitement des hiérarchies de  
classes

---

● Traitement des hiérarchies de  
classes (1)

● Traitement des hiérarchies de  
classes (2)

● Traitement des hiérarchies de  
classes (3)

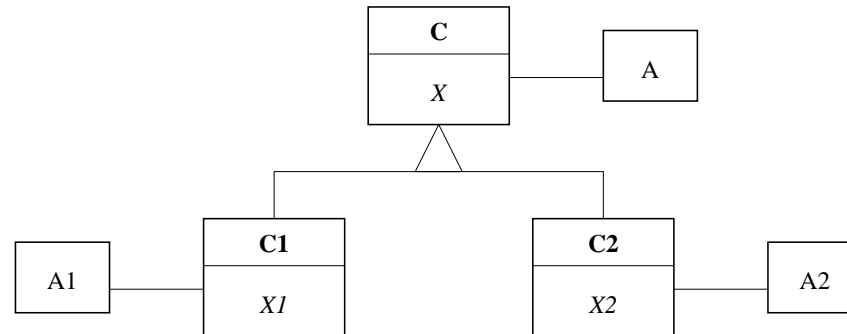
Normalisation des relations

---

Conclusion

---

Soit C une super-classe d'attributs X et associée à une classe A. Soit C1 et C2 deux sous-classes de C, d'attributs respectivement X1 et X2, et associées respectivement à A1 et A2.



On pose K l'identifiant de la relation qui serait associé à C.



# Traitement des hiérarchies de classes (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

● Traitement des hiérarchies de  
classes (1)

● Traitement des hiérarchies de  
classes (2)

● Traitement des hiérarchies de  
classes (3)

Normalisation des relations

Conclusion

On distingue plusieurs cas selon les contraintes sur l'héritage :

- **Complet** : un objet de C est instance d'au moins une sous-classe. On définit deux relations  $R1(X, X1)$  et  $R2(X, X2)$  (K est identifiant dans R1 et dans R2). L'ensemble des objets de C est représenté par la vue :  $R1[X] \cup R2[X]$ . L'association avec A est traitée avec C1 et avec C2. L'association avec A1 (resp. avec A2) est traitée avec C1 (resp. avec C2) ;
- **Incomplet** : un objet de C est instance de zéro sous-classe. On définit 3 relations :  $R(X)$ ,  $R1(K, X1)$  et  $R2(K, X2)$  (K est identifiant dans R, dans R1 et dans R2).  $R1[K] \subseteq R[K]$  et  $R2[K] \subseteq R[K]$ . L'association avec A est traitée avec C. L'association avec A1 (resp. avec A2) est traitée avec C1 (resp. avec C2).

# Traitement des hiérarchies de classes (3)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

● Traitement des hiérarchies de  
classes (1)

● Traitement des hiérarchies de  
classes (2)

● Traitement des hiérarchies de  
classes (3)

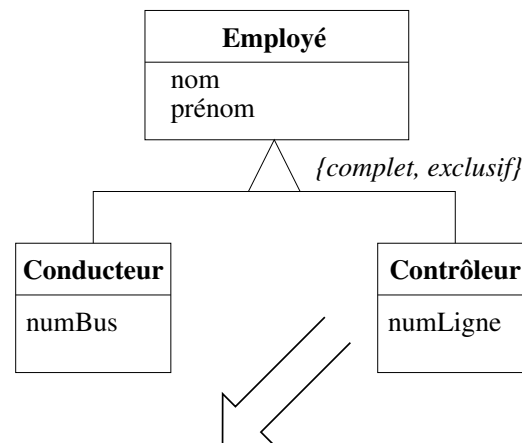
Normalisation des relations

Conclusion

On ajoute des contraintes selon les cas :

- Inclusif : un objet de C est instance de plusieurs sous-classes. Pas de contrainte supplémentaire.
- Exclusif : un objet de C est instance d'au plus une sous-classe. L'intersection de  $R1[K]$  et de  $R2[K]$  doit être vide.

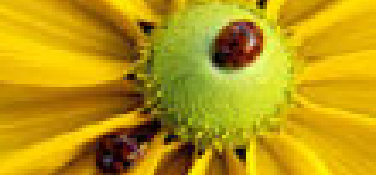
Prenons un exemple :



$\text{LesConducteurs}(\underline{\text{nom}}, \underline{\text{prenom}}, \text{numBus})$

$\text{LesContrôleurs}(\underline{\text{nom}}, \underline{\text{prenom}}, \text{numLigne})$

$\text{LesConducteurs}[\text{nom}, \text{prenom}] \cap \text{LesContrôleurs}[\text{nom}, \text{prenom}] = \emptyset$



# Normalisation des relations

- Introduction
- Conception d'un Système d'Information
- Diagramme de classes - Diagramme d'objets
- Agrégation et composition
- Association qualifiée
- Classe d'associations
- Traitement des hiérarchies de classes
- Normalisation des relations
  - Normalisation des relations
  - Anomalies de mise à jour
  - Notion de dépendance fonctionnelle (DF)
  - Première forme normale (1FN)
  - Deuxième forme normale (2FN) (1)
  - Deuxième forme normale (2FN) (2)
  - Troisième forme normale (3FN) (1)
  - Troisième forme normale (3FN) (2)
- Conclusion

Le but est d'éviter les «anomalies de mise à jour» et éliminer autant que possible les redondances faisant croître la base de façon «géométrique»...

Soit la relation LesCommandes(numProd, quantité, idFournisseur, adresse) :

LesCommandes(	<u>numProd</u> ,	quantité,	idFournisseur,	adresse)
	101	300	901	St-Égrève
	104	1900	902	Grenoble
	112	900	904	Grenoble
	103	19	901	St-Égrève
	...	...	...	...



# Anomalies de mise à jour

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

● Normalisation des relations

● Anomalies de mise à jour

● Notion de dépendance  
fonctionnelle (DF)

● Première forme normale  
(1FN)

● Deuxième forme normale  
(2FN) (1)

● Deuxième forme normale  
(2FN) (2)

● Troisième forme normale  
(3FN) (1)

● Troisième forme normale  
(3FN) (2)

Conclusion

- Anomalie de modification : si on souhaite mettre à jour l'adresse d'un fournisseur, il faut le faire pour tous les n-uplets concernés.
- Anomalie d'insertion : pour ajouter un nouveau fournisseur, il faut obligatoirement fournir des valeurs pour *numProd* et *quantité*.
- Anomalie de suppression : la suppression d'un produit peut faire perdre toutes les informations sur un fournisseur.



# Notion de dépendance fonctionnelle (DF)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

- Normalisation des relations
- Anomalies de mise à jour
- Notion de dépendance  
fonctionnelle (DF)
- Première forme normale  
(1FN)
- Deuxième forme normale  
(2FN) (1)
- Deuxième forme normale  
(2FN) (2)
- Troisième forme normale  
(3FN) (1)
- Troisième forme normale  
(3FN) (2)

Conclusion

Soit la relation  $R(X, Y, Z)$  où  $X$ ,  $Y$  et  $Z$  sont des ensembles d'attributs :  $Y$  **dépend fonctionnellement de**  $X$  ( $X \rightarrow Y$ ) si à chaque valeur de  $X$  ne correspond qu'une seule valeur de  $Y$ .

Exemple : soit LesParcelles(numP, nomP, adresse) ;  
 $numP \rightarrow nomP$  et  $nomP \rightarrow adresse$ . D'où  
 $numP \rightarrow nomP, adresse$ .





# Première forme normale (1FN)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

- Normalisation des relations
- Anomalies de mise à jour
- Notion de dépendance  
fonctionnelle (DF)
- Première forme normale  
(1FN)
- Deuxième forme normale  
(2FN) (1)
- Deuxième forme normale  
(2FN) (2)
- Troisième forme normale  
(3FN) (1)
- Troisième forme normale  
(3FN) (2)

Conclusion

*Une relation est en 1FN si tout attribut n'est pas décomposable.*

Exemple : la relation LesParcelles(numP, nomP, adresse) n'est pas en 1FN car l'attribut adresse est du type (3, avenue Gambetta) et peut donc être décomposé en numéro de voie, type de voie, nom de voie, etc.



# Deuxième forme normale (2FN) (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

- Normalisation des relations
- Anomalies de mise à jour
- Notion de dépendance  
fonctionnelle (DF)
- Première forme normale  
(1FN)
- Deuxième forme normale  
(2FN) (1)
- Deuxième forme normale  
(2FN) (2)
- Troisième forme normale  
(3FN) (1)
- Troisième forme normale  
(3FN) (2)

Conclusion

Elle permet d'éliminer les redondances dues aux répétitions de valeurs.

*Une relation est en 2FN si :*

- *elle est en 1FN ;*
- *tout attribut n'appartenant pas à la clé primaire dépend entièrement de celle-ci.*

Exemple : la relation LesClients(numc, nom, prénom, numVoie, typeVoie, codePostal, ville) est en 2FN.



# Deuxième forme normale (2FN) (2)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

- Normalisation des relations
- Anomalies de mise à jour
- Notion de dépendance  
fonctionnelle (DF)
- Première forme normale  
(1FN)
- Deuxième forme normale  
(2FN) (1)
- Deuxième forme normale  
(2FN) (2)
- Troisième forme normale  
(3FN) (1)
- Troisième forme normale  
(3FN) (2)

Conclusion

Exemple : la relation LesCommandes(numProd, idFournisseur, quantité, ville) n'est pas en 2FN car

$numProd, idFournisseur \rightarrow quantité$  et  $idFournisseur \rightarrow ville$ .

La décomposition suivante donne deux relations en 2FN :

LesCommandes(numProd, idFournisseur, quantité) et  
LesFournisseurs(idFournisseur, ville). Exercice : la relation

AchatsParcelles(nomProprio, numP, date, numActe, commune, département) n'est pas en 2FN. Pourquoi ?  
Comment la décomposer pour obtenir des relations en 2FN ?



# Troisième forme normale (3FN) (1)

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

- Normalisation des relations
- Anomalies de mise à jour
- Notion de dépendance fonctionnelle (DF)
- Première forme normale (1FN)
- Deuxième forme normale (2FN) (1)
- Deuxième forme normale (2FN) (2)
- Troisième forme normale (3FN) (1)
- Troisième forme normale (3FN) (2)

Conclusion

Elle permet d'éliminer les redondances dues aux relations de transitivité qui peuvent exister entre attributs. *Une relation est en 3FN si :*

- *elle est en 2FN ;*
- *il n'existe aucune DF entre deux attributs n'appartenant pas à la clé primaire.*

Exemple : la relation LesCommandes(numCom, date, idFournisseur, numVoie, typeVoie, nomVoie, ville, codePostal) n'est pas en 3FN. En effet, on a par exemple :

$numCom \rightarrow idFournisseur$ ,  $idFournisseur \rightarrow ville$  et  $numCom \rightarrow ville$  (transitivité). Cette transitivité introduit des problèmes : caractéristiques des fournisseurs répétées à chaque commande les concernant si changement d'adresse, répercuter cette modification dans tous les n-uplets correspondants. Comment décomposer pour éviter cela ?



# Troisième forme normale (3FN) (2)

**Remarque importante :** la 3FN ne permet pas de résoudre tous les problèmes de mise à jour et de redondance ; il existe d'autres formes normales...

- Introduction
- Conception d'un Système d'Information
- Diagramme de classes - Diagramme d'objets
- Agrégation et composition
- Association qualifiée
- Classe d'associations
- Traitement des hiérarchies de classes
- Normalisation des relations
  - Normalisation des relations
  - Anomalies de mise à jour
  - Notion de dépendance fonctionnelle (DF)
  - Première forme normale (1FN)
  - Deuxième forme normale (2FN) (1)
  - Deuxième forme normale (2FN) (2)
  - Troisième forme normale (3FN) (1)
  - Troisième forme normale (3FN) (2)
- Conclusion



# Références

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

● **Références**

● En conclusion

- P.-C. Scholl, M.-C. Fauvet, F. Lagnier, F. Maraninchi - Cours d'Informatique - Langages et programmation. Masson 1993

# En conclusion

Introduction

Conception d'un Système  
d'Information

Diagramme de classes -  
Diagramme d'objets

Agrégation et composition

Association qualifiée

Classe d'associations

Traitement des hiérarchies de  
classes

Normalisation des relations

Conclusion

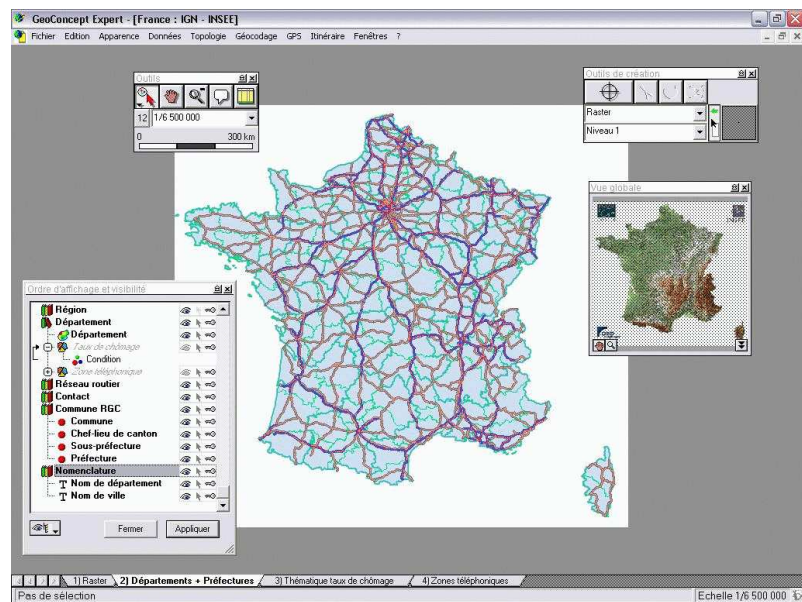
● Références

● En conclusion



*Hier...*

*...aujourd'hui !*



*Maintenant, à vous de jouer...*