



# Modélisation

## Semaine 1

**420-KGB-JQ**  
**Gestion et exploitation de bases  
de données**

# Méthodologie de modélisation conceptuelle de données

## Notes:

- Le design de bd est un processus itératif qui possède un point de départ et une processus presque infinie d'affinages.
- Dans la cueillette des informations pour constituer un modèle conceptuel, il n'est pas toujours évident de savoir si un objet donné est une entité, une relation ou un attribut.
- Le design de la réalité d'une entreprise est **très subjectif** et des concepteurs différents obtiendraient très probablement des interprétations différentes mais toutes aussi valables.

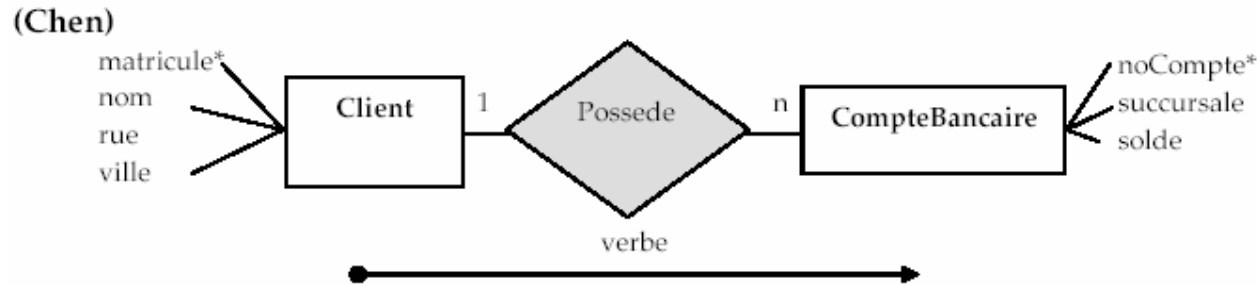
# La modélisation conceptuelle des données (MCD)

L'utilisation du modèle conceptuel repose sur le besoin **de raisonner au niveau générique** plutôt **qu'au niveau des instances** et leurs associations qui sont très nombreuses.

Plusieurs **formalismes** sont proposés, tous sont des variantes du formalisme **Entité/Association** de l'américain Peter Chen. Chaque variante (Merise, UML, OMT, ...) repose sur à peu près les mêmes notions, mais utilise parfois une **représentation graphique** particulière.

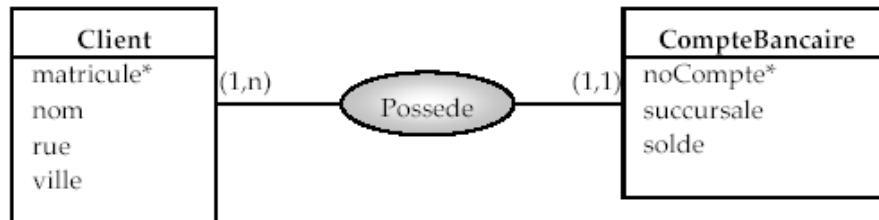
# MCD : Formalismes utilisés

- Formalisme Entité/Association selon Peter Chen :

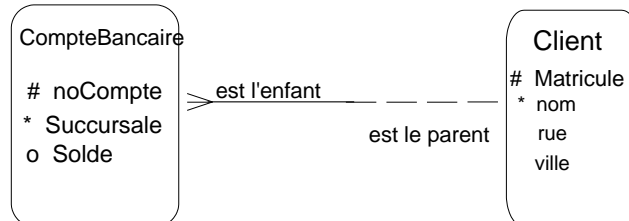


- Formalisme Entité/Association (entité-relation) selon Merise :

(Merise)



- Formalisme en pattes d'oie (utilisé par Oracle) :

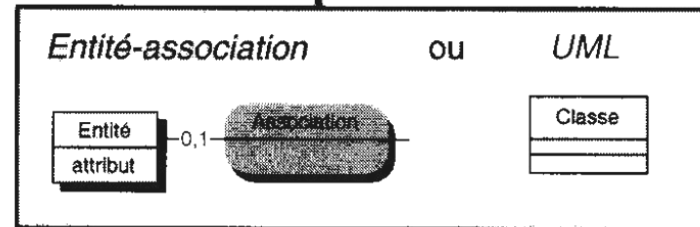


- Formalisme UML : (tel que développé dans le cours de Modélisation de données)

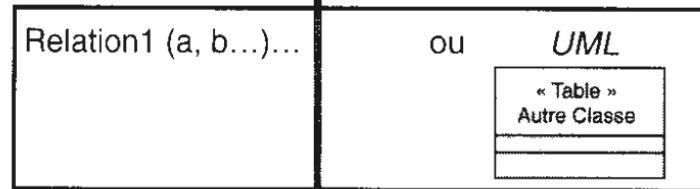
Systeme à modeliser



Niveau conceptuel



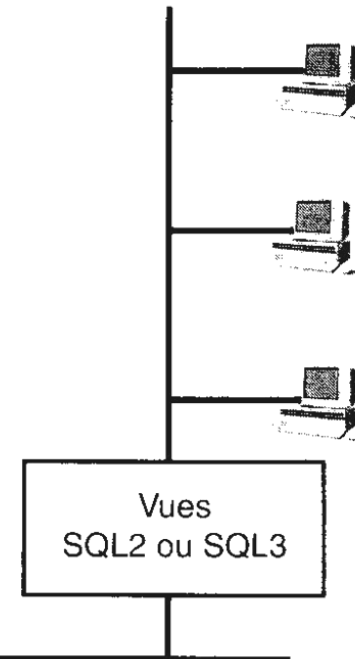
Niveau logique  
Modèle de données



Niveau physique

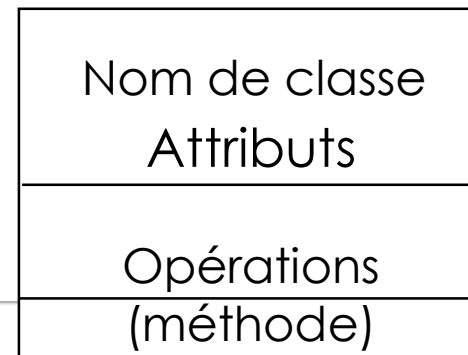


Niveau externe

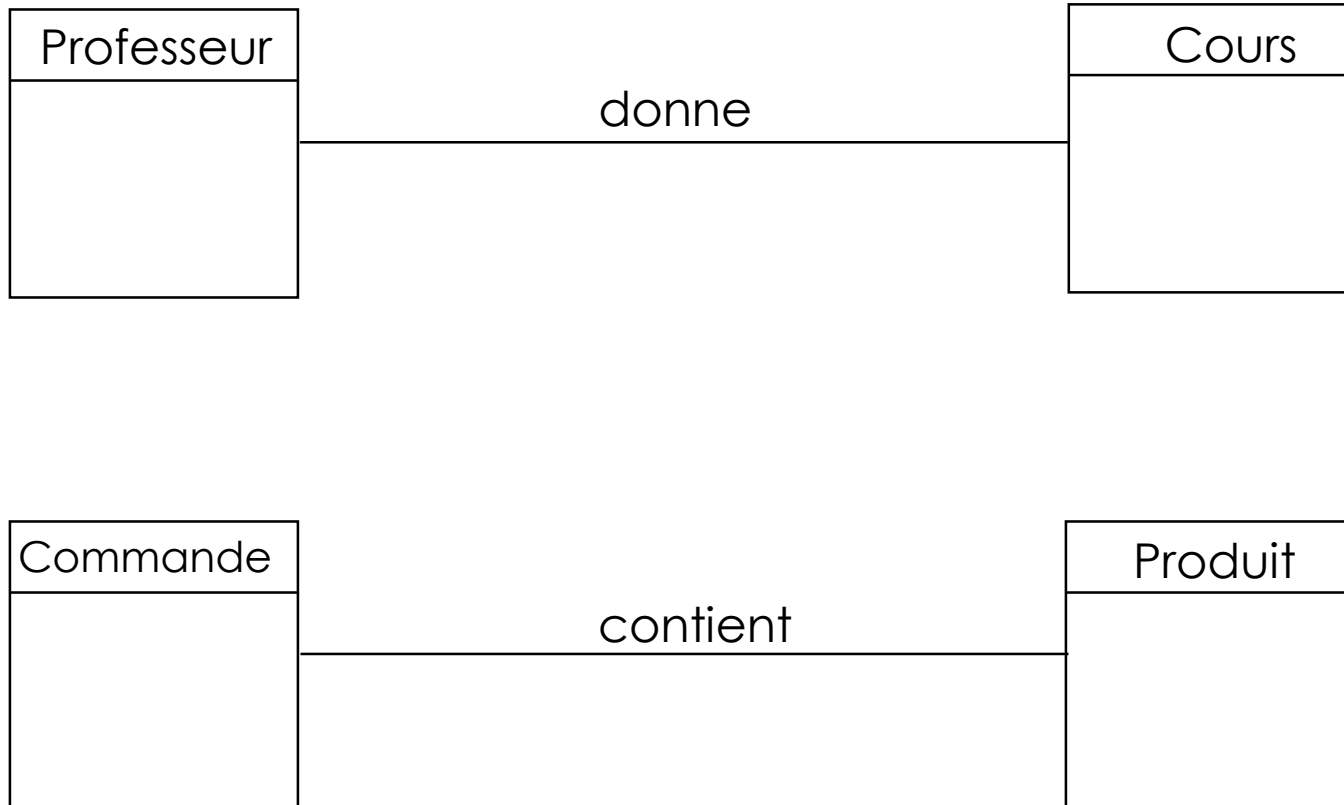


# Modélisation entité-association selon le formalisme UML : Terminologie

- ❖ Le diagramme des classes est une représentation, un modèle:
  - des objets et des intervenants, ou «entités»,
  - des associations, ou «relations», qui s'établissent entre les objets ou entités;
  - des éléments de données, attributs ou propriétés, qui décrivent ces objets.
- ❖ Représente une catégorie d'intervenants ou d'objets définis en utilisant les mêmes éléments de données.

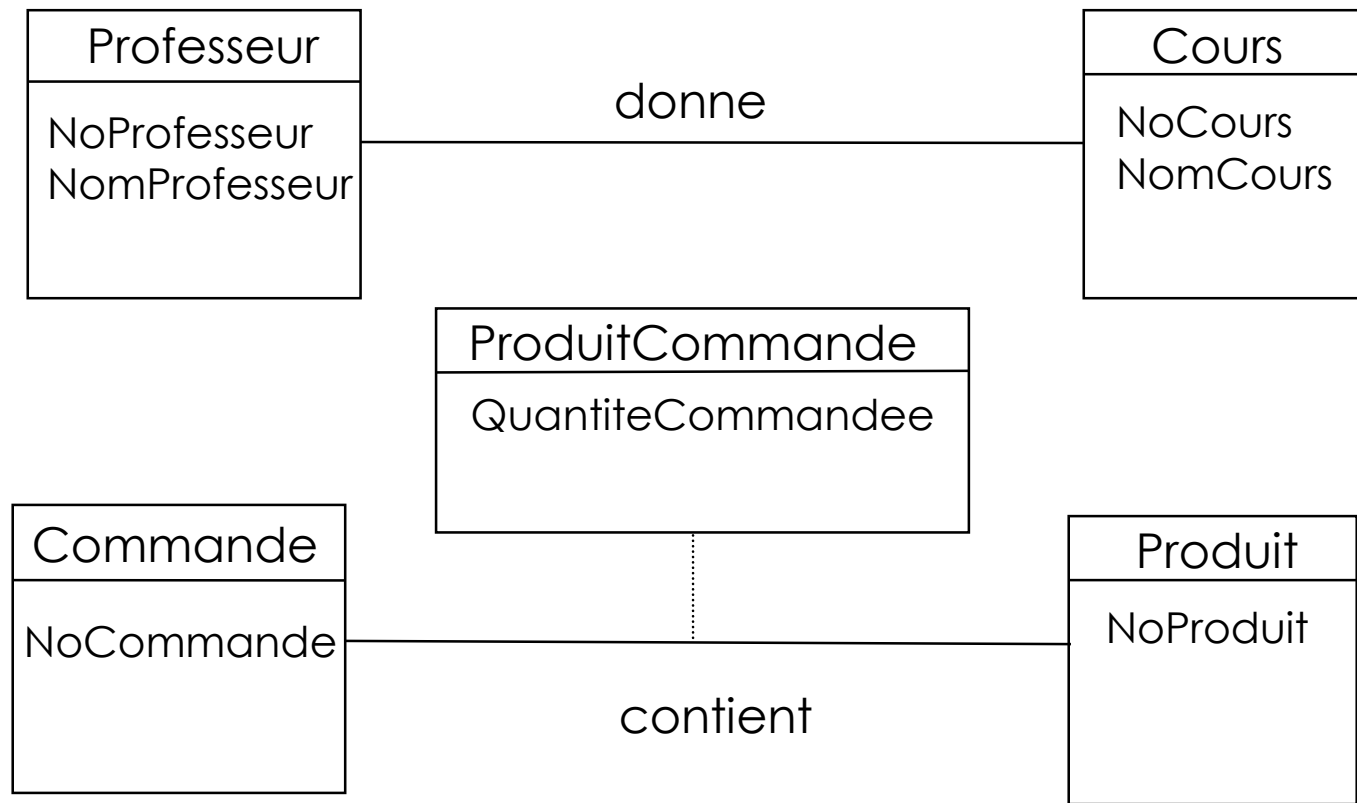


# Exemples



# Les classes d'association

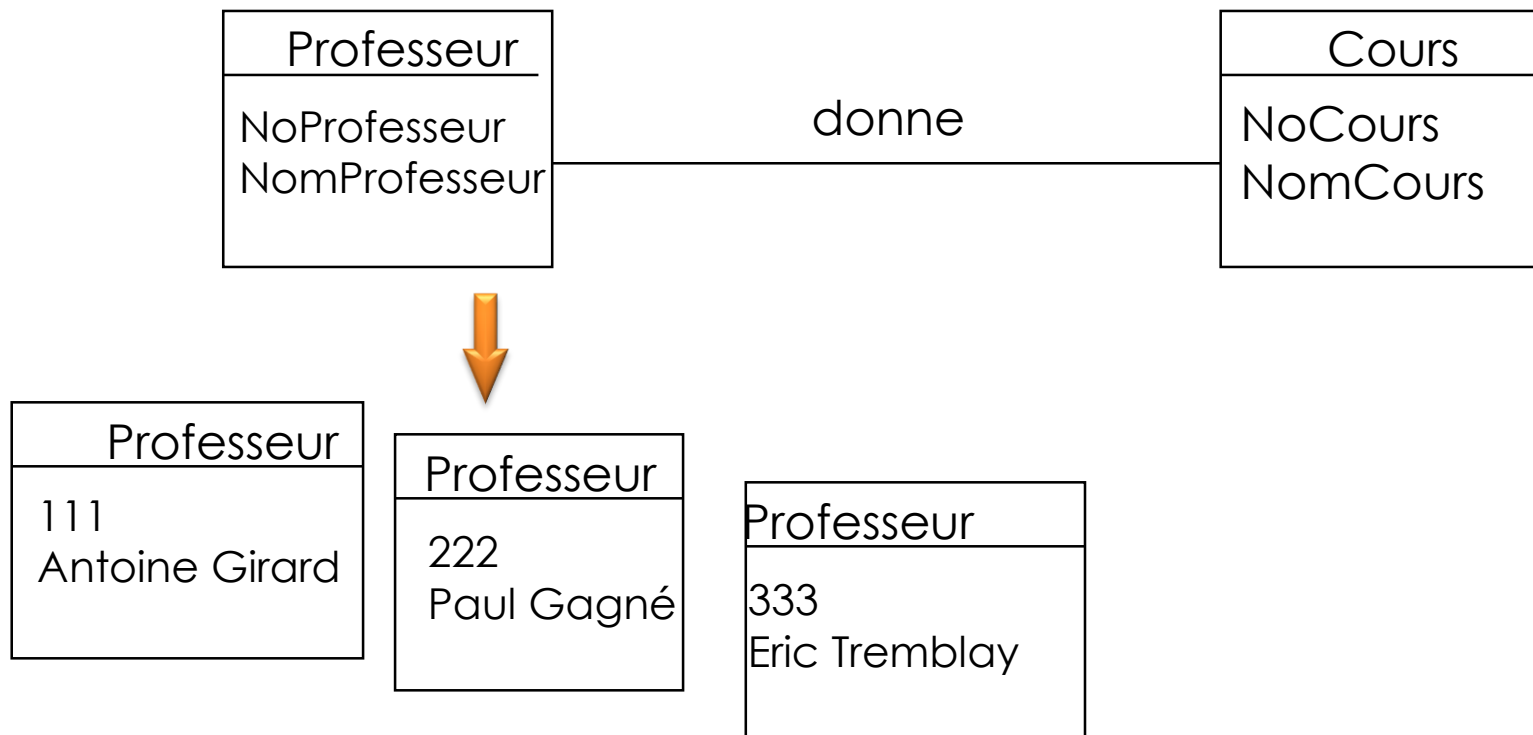
- Des attributs peuvent être associés aux classes mais aussi aux **associations**





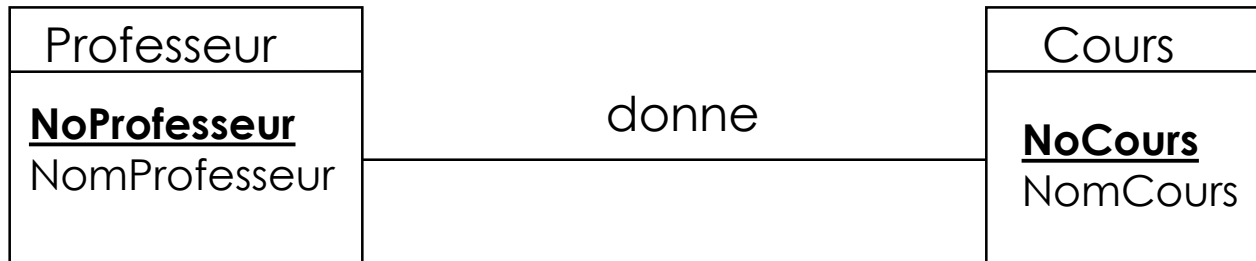
# Instance

- Un intervenant ou un objet particulier est appelé une instance



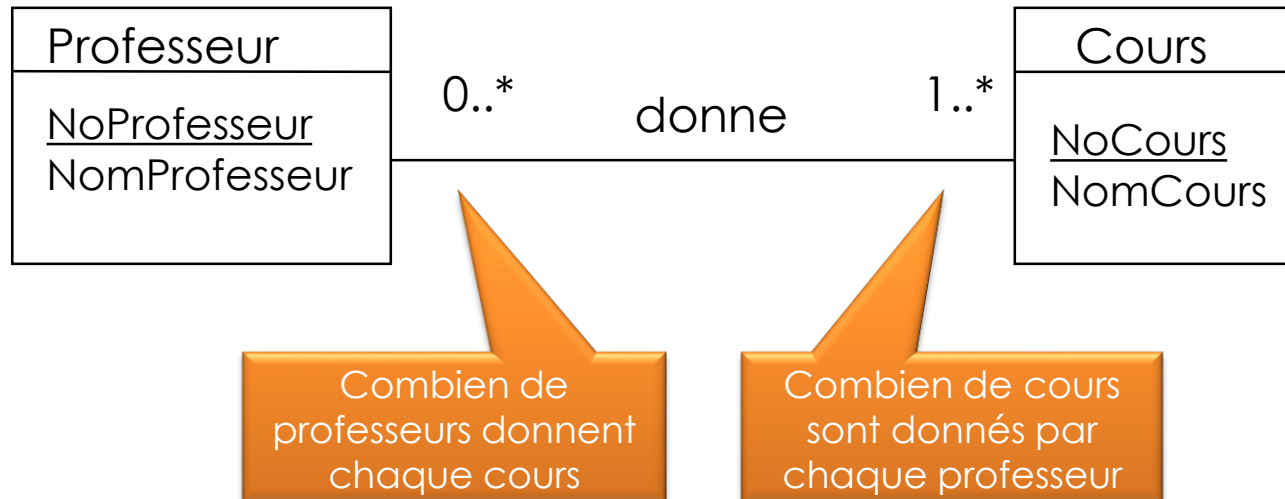
# Identifiant d'un objet

- Identifie chaque instance d'un objet de façon unique
- Dans le vocabulaire courant on l'appelle aussi clé



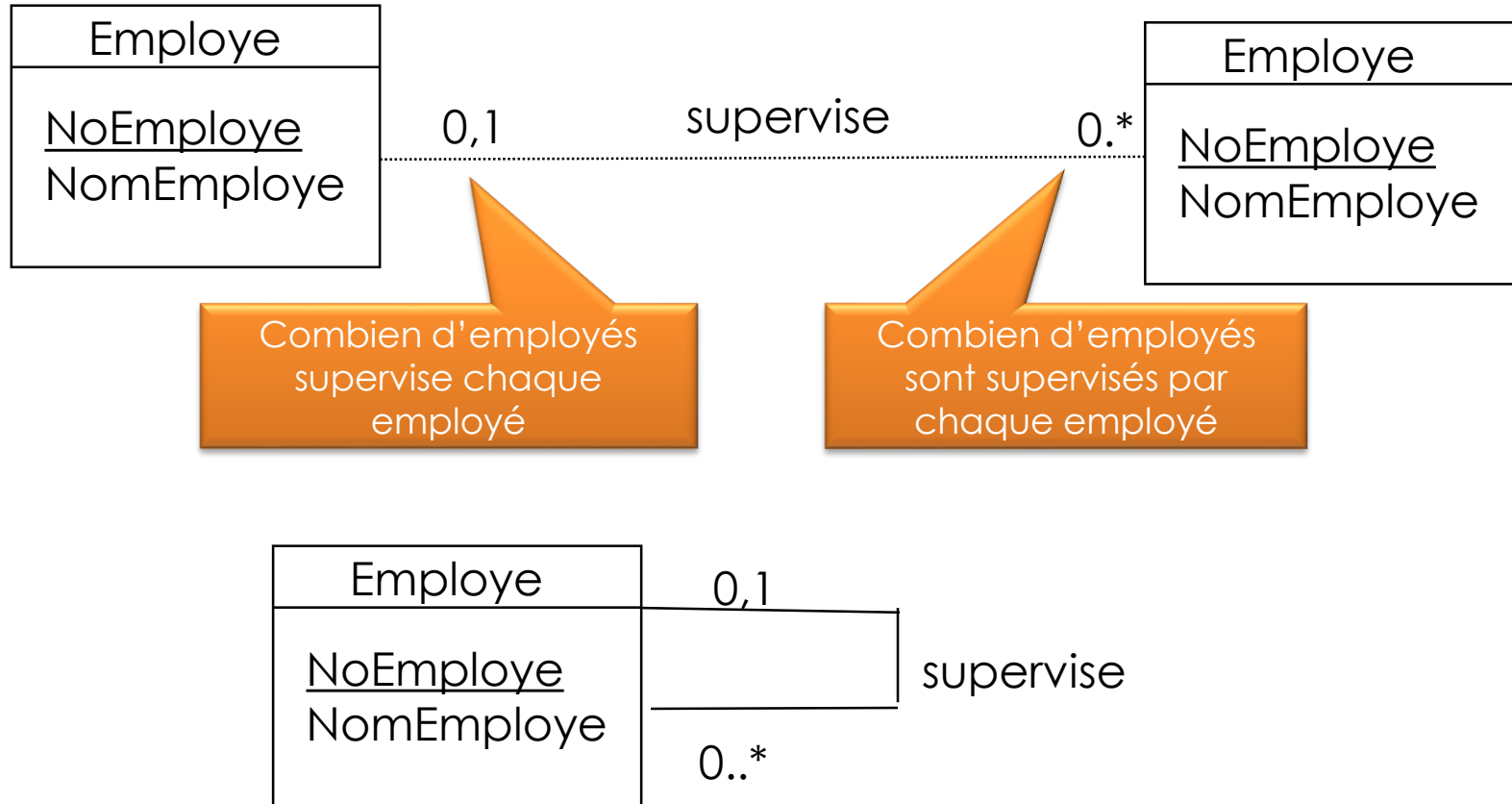
# Multiplicité des associations

- Le nombre de fois qu'une même instance de la classe peut être impliquée dans la relation qui l'associe à une autre classe



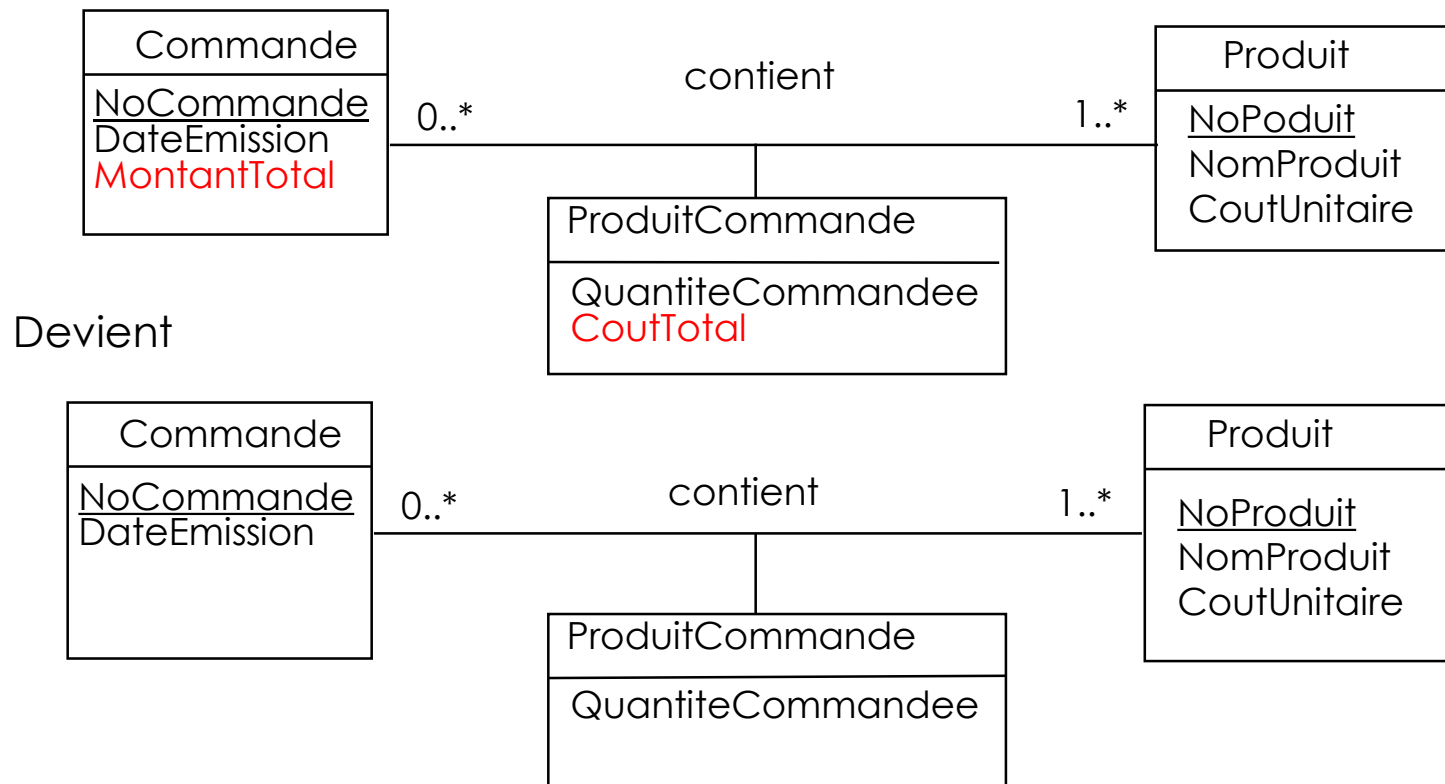
- Les différentes possibilités sont 0..1, 1..1, 0..\* ou 1..\*

# Une association réflexive



# Les données calculées

- Le modèle ne doit contenir que des données élémentaires et non des données calculées

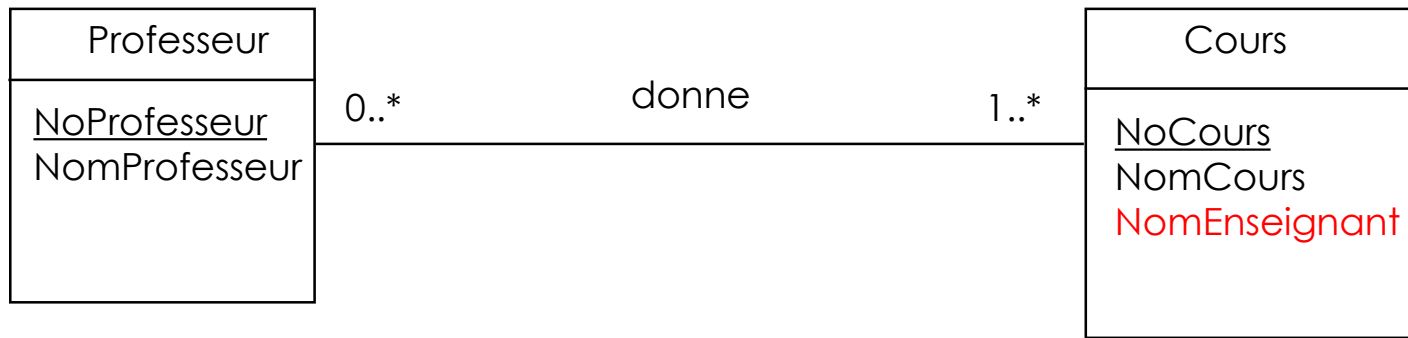


# Données calculées & dénormalisation

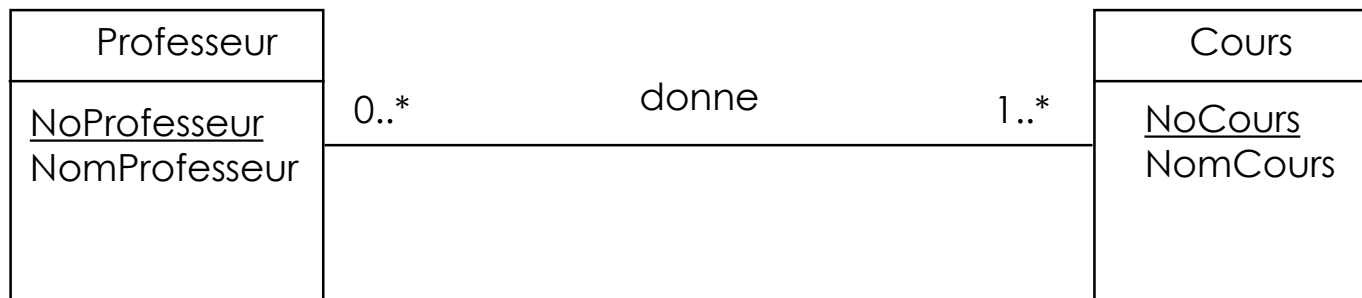
- Pour des raisons de performances, il est parfois acceptable que des données calculées soient introduites dans une base de données.
- Lorsque ce choix est considéré, on parle de « **dénormalisation** » et ça ne doit pas se faire à l'étape du modèle conceptuel.

# Unicité des attributs

- Un attribut est unique à une association ou à une classe
- Attention aux synonymes

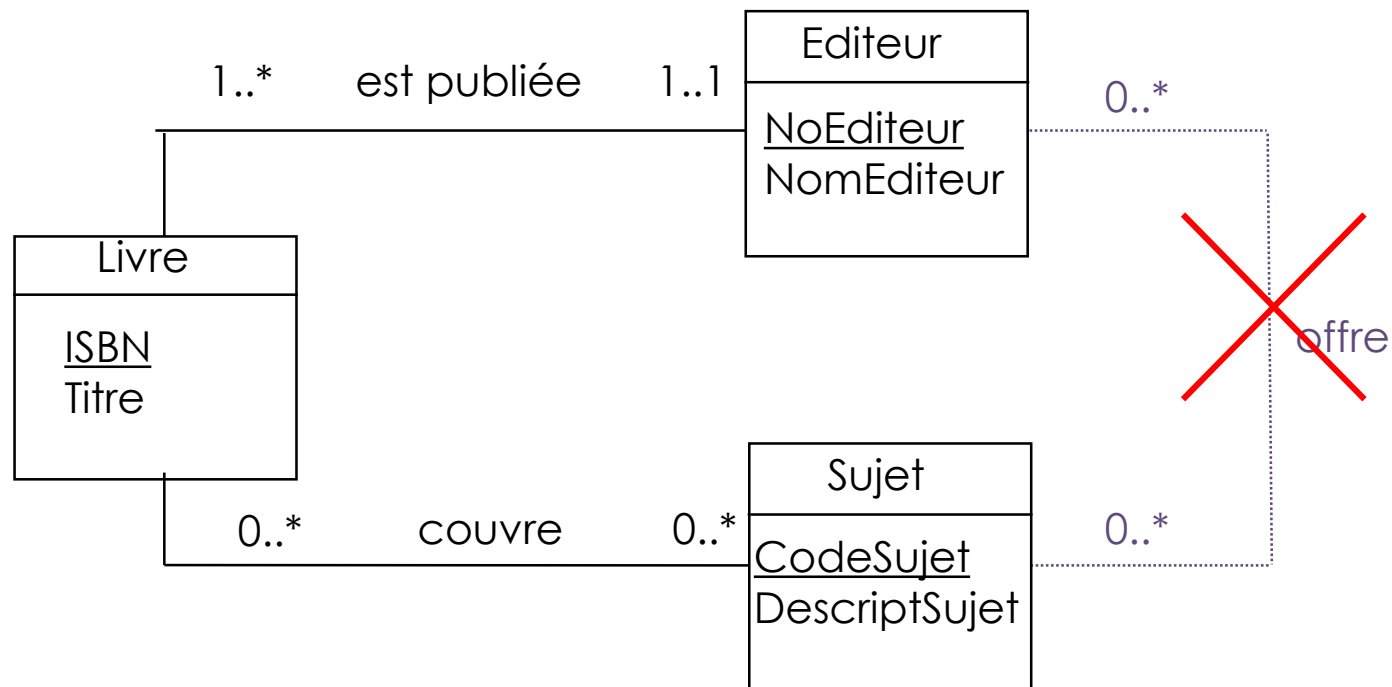


Devient



# Associations pouvant être déduites

- Lorsqu'une association peut être déduite des autres associations, elle n'est généralement pas représentée





# Qu'est-ce que la normalisation

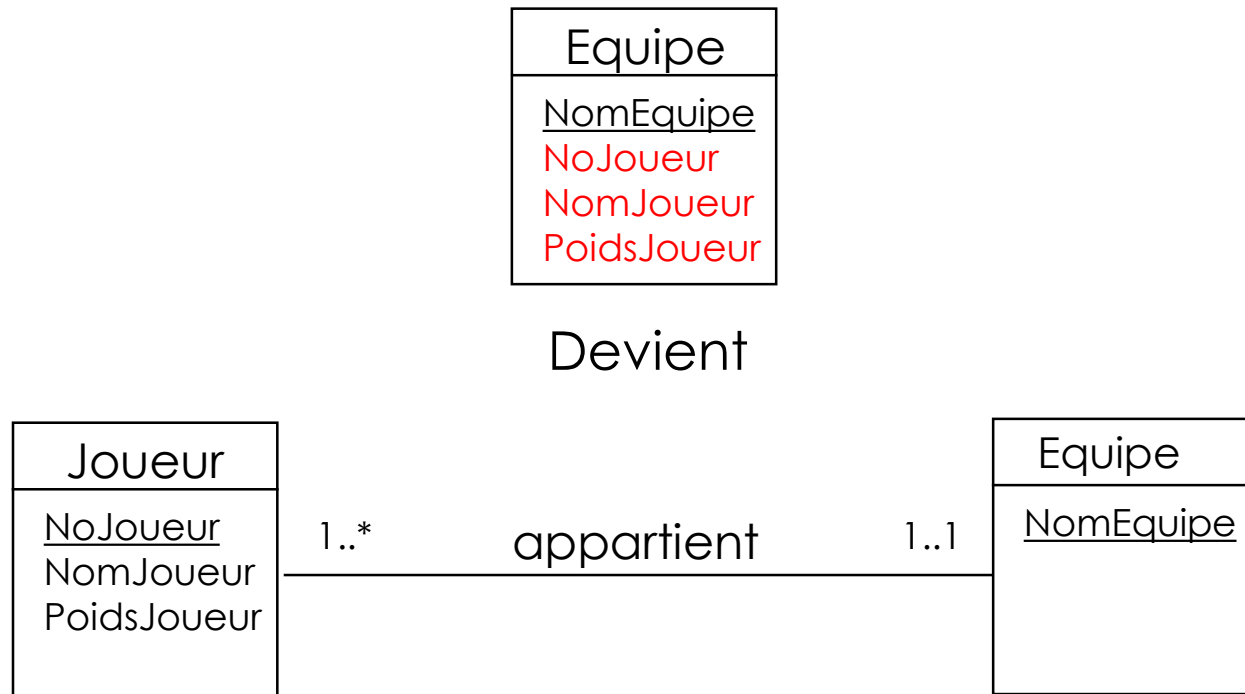
- La normalisation est un processus qui permet de s'assurer qu'un modèle de données est bien construit.
- La normalisation est un ensemble de règles développées pour éviter les incohérences dans une bd relationnelle provenant des opérations de mises à jour.
- Elle vise d'abord à éliminer la redondance dans les données et améliorer la cohérence.
- Elle permettra d'obtenir une structure des données plus stable qui supportera plus facilement les modifications futures.

# La normalisation

- Quand effectuer la normalisation ?
  - À n'importe quel stade de conception d'une bd: conceptuel ou logique.  
On dit alors que les entités / tables sont normalisées.  
**La normalisation est donc une opération qui débute avec la conception de la bd et qui doit être revue au cours de son cycle de vie.**
  - Lors de l'évolution du modèle, il faut appliquer à nouveau les algorithmes de normalisation pour garantir que les modifications apportées aux entités / tables respectent toujours les principes de normalité.
- Anomalies de mise à jour
  - Anomalies d'insertion
  - Anomalies de suppression
  - Anomalies de modification

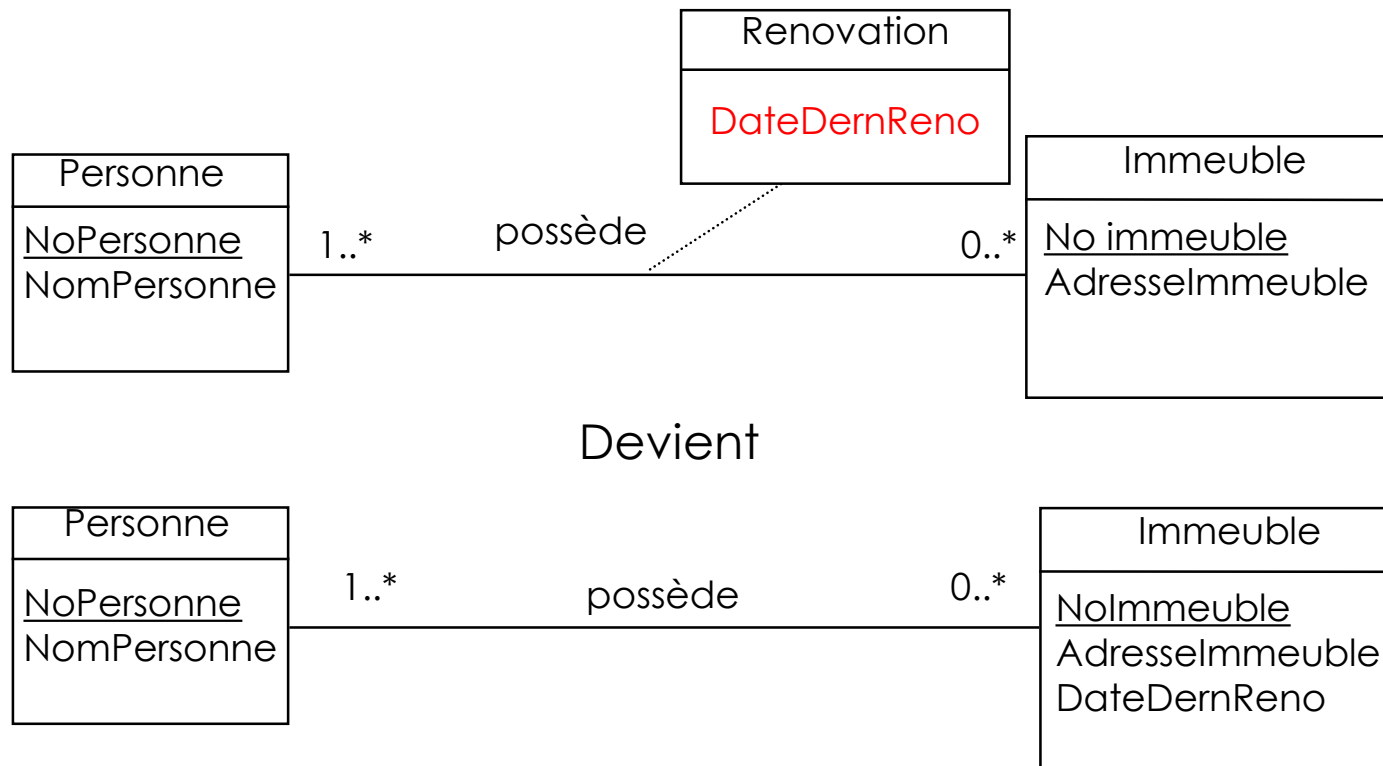
# Première forme normale

- Éliminer les données répétitives



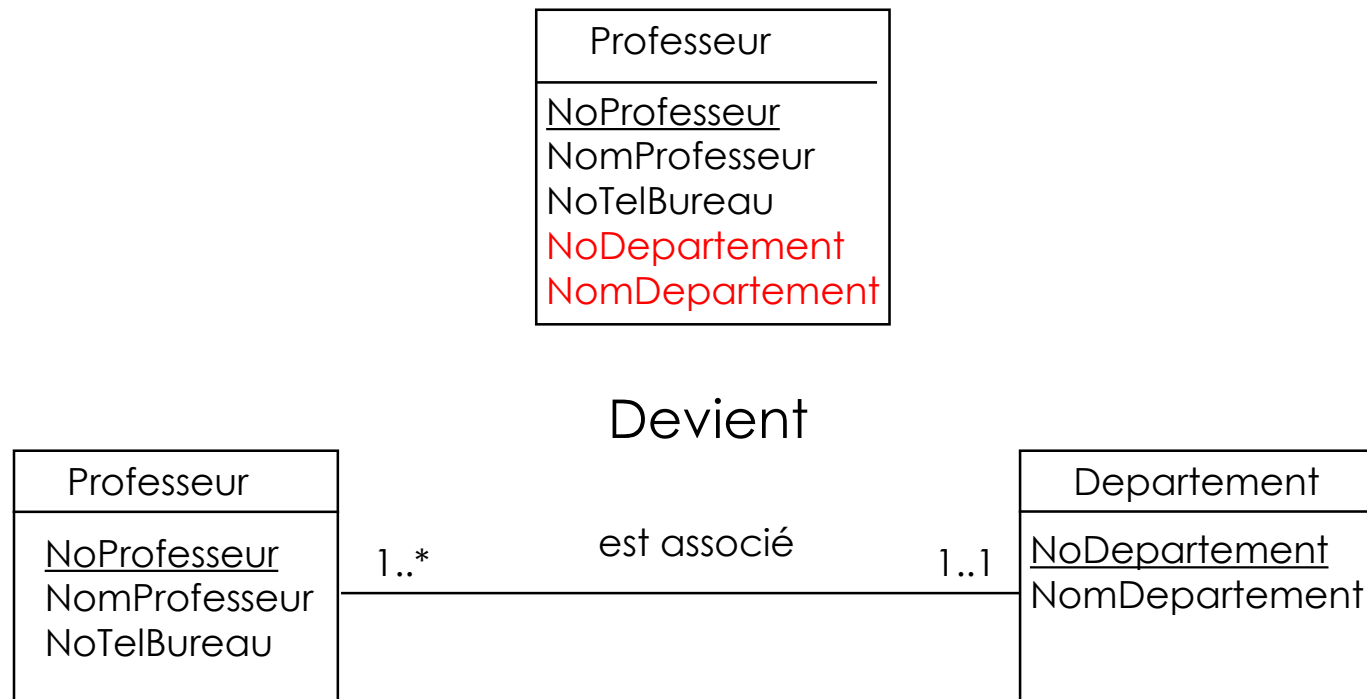
# Deuxième forme normale

- Les propriétés doivent dépendre de tout l'identifiant et non seulement d'un sous-ensemble - garantir la dépendance fonctionnelle



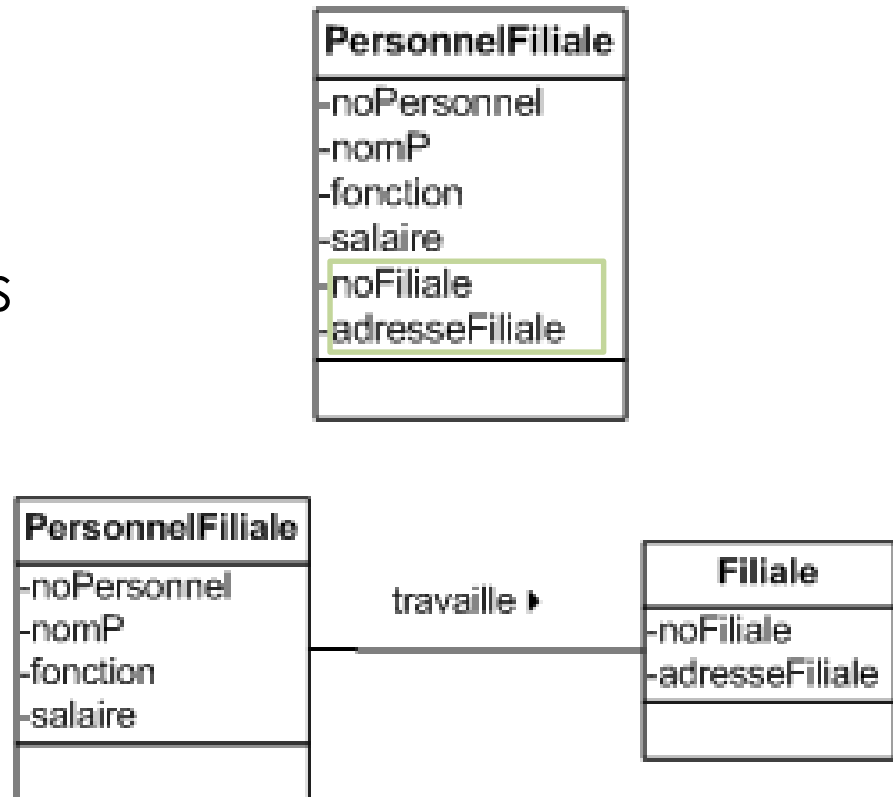
# Troisième forme normale

- Les propriétés ne dépendent que de l'identifiant et non d'une autre propriété - éliminer les dépendances transitives



# Exemple d'anomalie d'insertion

L'attribut adresseFiliale peut engendrer des incohérences entre les différentes occurrences de cette classe/entité.



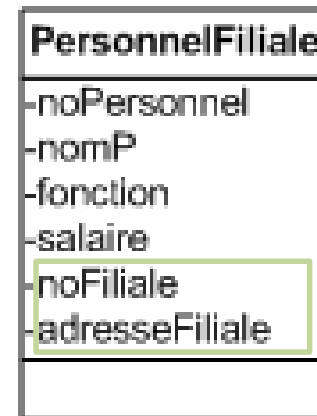
# Exemple d'anomalie de suppression

- Lorsqu'on détruit la dernière occurrence de cette classe/entité, alors il ne subsistera plus aucune trace des filiales de la compagnie.

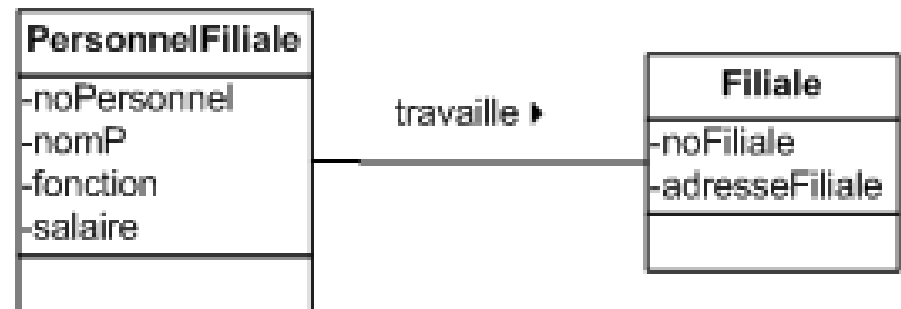
PersonnelFiliale
-noPersonnel
-nomP
-fonction
-salaire
-noFiliale
-adresseFiliale

# Exemple d'anomalie de modification

- Si vous désirez changer l'adresse de la filiale, vous devez changer cette information dans toutes les occurrences qui concernent cette filiale en particulier.



- Dans ce cas-ci, vous n'avez à changer l'information qu'une seule fois.



**Pour éviter les anomalies de mises à jour, il est recommandé de poursuivre la normalisation jusqu'en 3<sup>e</sup> forme normale.**



# Facteurs critiques de succès en modélisation de données

- Travailler autant que possible de manière interactive avec les utilisateurs.
- Respecter une méthodologie structurée tout au long du processus de modélisation des données.
- Suivre une approche orientée données.
- Incorporer des considérations structurelles et d'intégrité dans les modèles de données.
- Combiner les techniques de conceptualisation, de normalisation et de validation des transactions dans la méthodologie de modélisation des données.
- Employer des diagrammes pour représenter la plus grande partie possible des modèles de données.
- Faire appel à un langage de design de bd (LDBD) pour représenter les sémantiques supplémentaires des données qu'un diagramme ne représente pas assez aisément.
- S'animer de la volonté de répéter et de reprendre les étapes.