

# Modélisation E/R des Données

1. Objectifs et principes
2. Le modèle Entité-Association (E/R)
3. Passage au relationnel
4. Conclusion

# 1. Objectifs de la Modélisation

- Permettre une meilleure compréhension
  - Le monde réel est trop complexes
  - Abstraction des aspects cruciaux du problème
  - Omission des détails
- Permettre une conception progressive
  - Abstractions et raffinements successifs
  - Possibilité de prototypage rapide
  - Découpage en modules ou packages
  - Génération des structures de données (et de traitements)

# Élaborer un modèle conceptuel

- **Isoler les concepts fondamentaux**
  - Que vont représenter les données de la BD ?
  - Découvrir les concepts élémentaires du monde réel
  - Décrire les concepts agrégés et les sous-concepts
- **Faciliter la visualisation du système**
  - Diagrammes avec notations simple et précise
  - Compréhension visuelle et non seulement intellectuelle

# Dériver le schéma de la BD

- Schéma
  - Définition de tous les types de données de la base et de leurs liens
- Agrégation de données
  - Type élémentaire (de base): Entier, Réel, String, ...
  - Type complexe (composé): Collection de types élémentaires
    - Tuple :
      - Exemple: Type Personne (nom: String, Prenom: String, age: Réel)
      - Instance ou occurrence Personne("Dupont", "Jules", 20)
    - Set :
      - Exemple : Voitures {id:String}; Voitures {"75AB75", "1200VV94"}
    - Bag, List, ...
- Possibilité d'intégrer des relations entre données (liens)
  - Exemple : Personne  $\rightarrow$  Voitures; "Dupont"  $\rightarrow$  "75AB75"

# Modélisation à plusieurs niveaux

Réel				
Modèle conceptuel	<ul style="list-style-type: none"> <li>◆ Indépendant du modèle de données</li> <li>◆ Indépendant du SGBD</li> </ul>			
Modèle logique	<ul style="list-style-type: none"> <li>◆ Dépendant du modèle de données</li> <li>◆ Indépendant du SGBD</li> </ul>	Codasy1	Relationnel	Objet XML
Modèle Physique	<ul style="list-style-type: none"> <li>◆ Dépendant du modèle de données</li> <li>◆ Dépendant du SGBD</li> </ul>	<ul style="list-style-type: none"> <li>◆ Organisation physique des données</li> <li>◆ Structures de stockage des données</li> <li>◆ Structures accélératrices (index)</li> </ul>		

# Génération de méthodes

- Méthodes d'analyse et de décomposition hiérarchiques
  - 1e génération basée sur des arbres fonctionnels
  - Diviser pour régner (Problème --> Sous-problème)
  - Warnier, SADT, Jackson, De Marco
- Méthodes d'analyse et de représentation systémiques
  - 2e génération basée sur entité-association
  - Séparation des données et traitements
  - Merise, Axial, SSADM
- Méthodes d'analyse et de conception orientées objets
  - 3e génération basée sur les objets
  - Réconciliation données et traitements
  - Réutilisation de composants

# Objectifs des méthodes

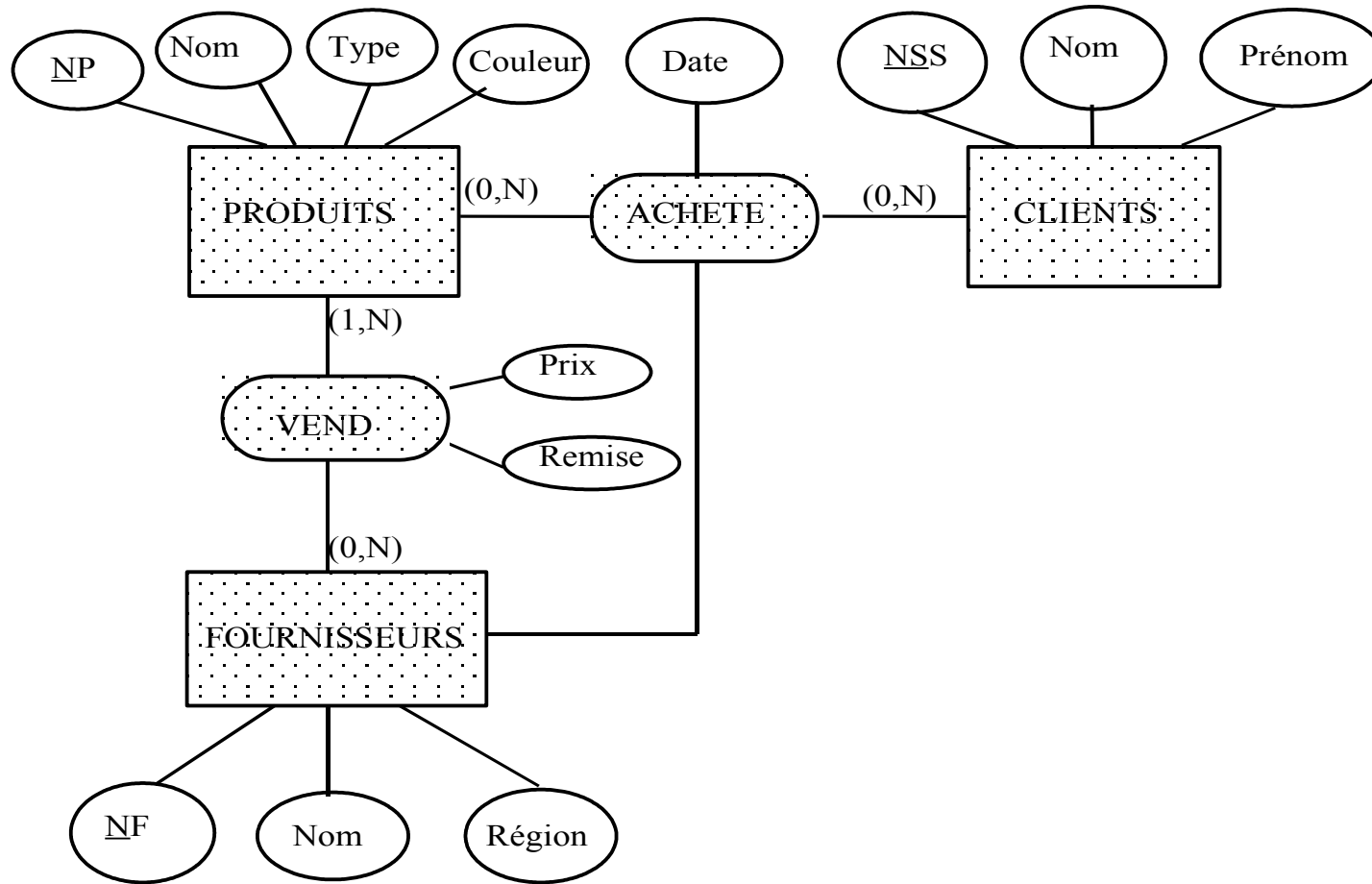
- Réduire la distance sémantique entre le langage des utilisateurs et le langage des concepteurs
  - meilleure communication entre utilisateurs et concepteurs
  - abstraction du réel perçu en termes compréhensibles et visibles
- Regrouper l'analyse des données et des traitements
  - meilleure compréhension des choses
  - plus grande cohérence entre l'aspect statique et l'aspect dynamique
- Simplification des transformations entre niveau conceptuel et niveau interne
  - implémentation directe éventuelle du schéma conceptuel
  - établissement possible de règles de transformations automatisées

## 2. Le Modèle Entité – Association (E/R Model)

- Ensemble de concepts pour modéliser les données d'une application (d'une entreprise)
- Ensemble de symboles graphiques associés
- Formalisé en 1976 par P. Chen
- Etendu vers E/R généralisé puis vers l'objet



# Exemple de modèle E/R

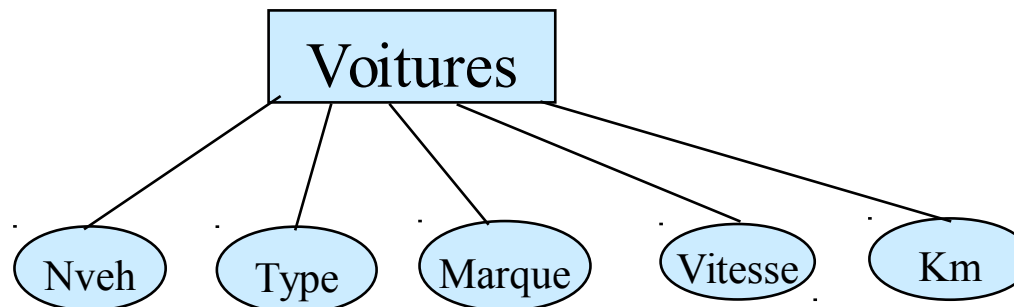
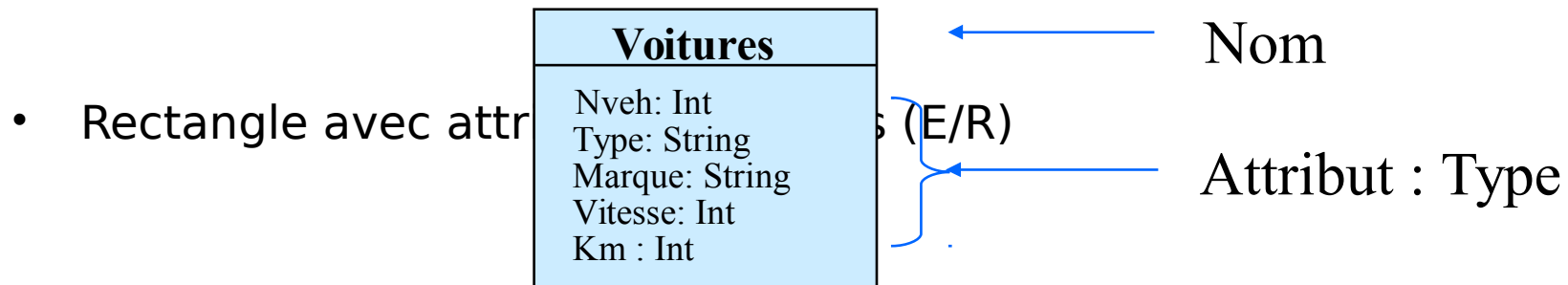


# Entité

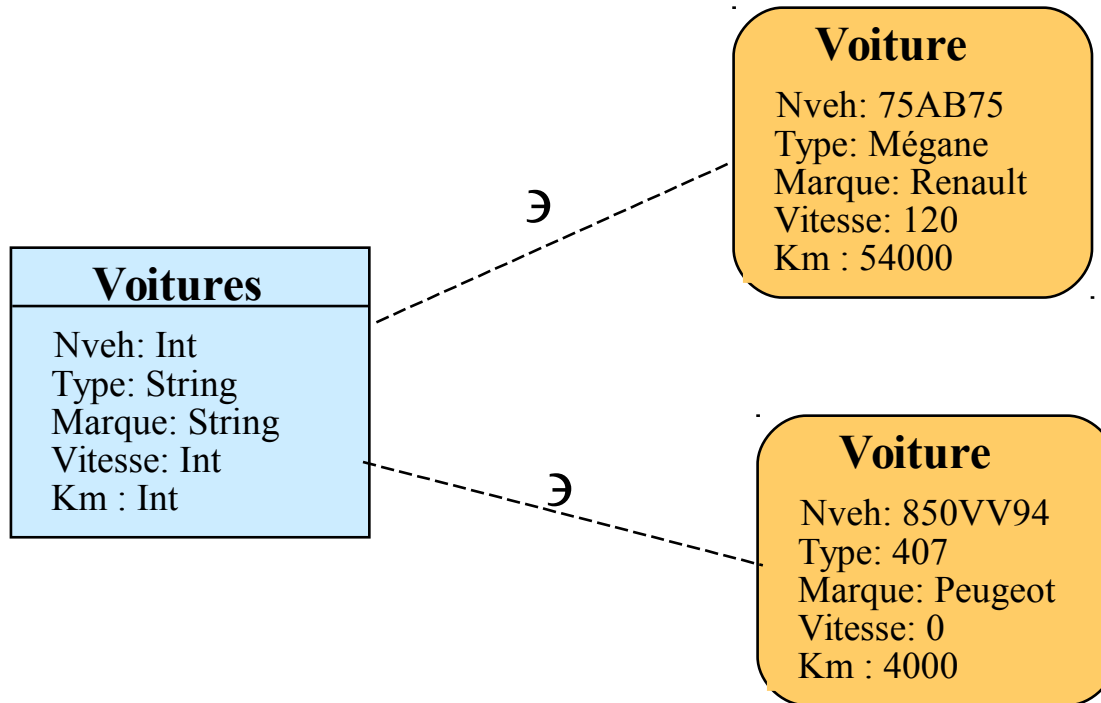
- Un objet du monde réel qui peut être identifié et que l'on souhaite représenter
  - La classe d'entité correspond à une collection d'entités décrites par leur type commun (le format)
  - L'instance d'entité correspond à un élément particulier de la classe d'entité (un objet)
  - Attention: on dit entité pour les deux ! Comprendre selon le contexte.
- Il existe généralement plusieurs entités dans une classe

# Représentation

- Rectangle avec attributs (UML)



# Exemple d'instance



# Attribut

- Description des propriétés des entités
- Toutes les instances d'une entité ont les mêmes attributs
  - Attribut simple: attribut ayant une valeur d'un type de base
  - Attribut composé: attribut constitué d'un groupe d'attributs
  - Attribut multi-valué: attribut pouvant avoir plus d'une valeur
- Avec le modèle E/R de base tout attribut est simple
- Avec le modèle E/R étendu, les attributs peuvent être complexes
  - Composés et multi-valués

# Identifiant ou Clé

- Un identifiant aussi appelé clé est un attribut qui permet de retrouver une instance d'entité unique à tout instant parmi celles de la classe.
  - Exemple: NVeh dans Voitures, NSS dans Personnes
- Un identifiant peut être constitué de plusieurs attributs (clé composée)
  - Exemple:
    - [N° , Rue, Ville] pour Maisons
    - [Nom, Prénom] pour Personnes

# Association

- Les entités sont reliées ensemble par des associations
  - Entre instances: par exemple 1 véhicule est associé à 1 personne
  - Entre classes: abstraction des associations entre instances
- Une association peut avoir des attributs (propriétés)
- Elle peut relier plusieurs entités ensemble
- Il est possible de distinguer le rôle d'une entité (elle peut en avoir plusieurs)

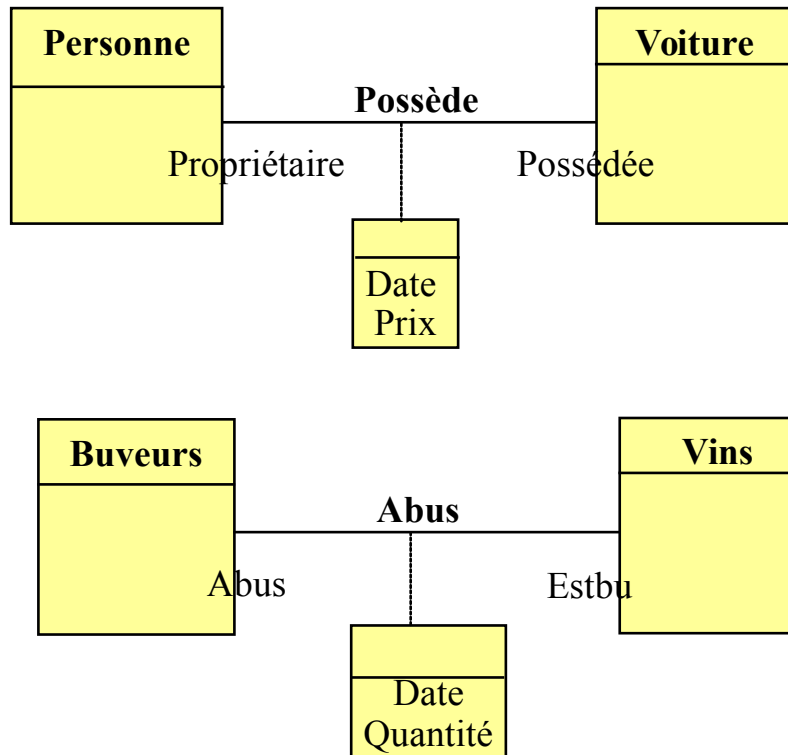
# Association: quelques définitions

- Association (Association)
  - Une relation entre des instances de deux (ou plus) classes
- Lien (Link)
  - Une instance d'association
- Rôle (Role)
  - Une extrémité d'une association
- Attribut de lien (Link attribute)
  - Un attribut de l'association instancié pour chaque lien
- Cardinalité (Multiplicity)
  - Le nombre d'instance d'une entité pour chaque instance de l'autre

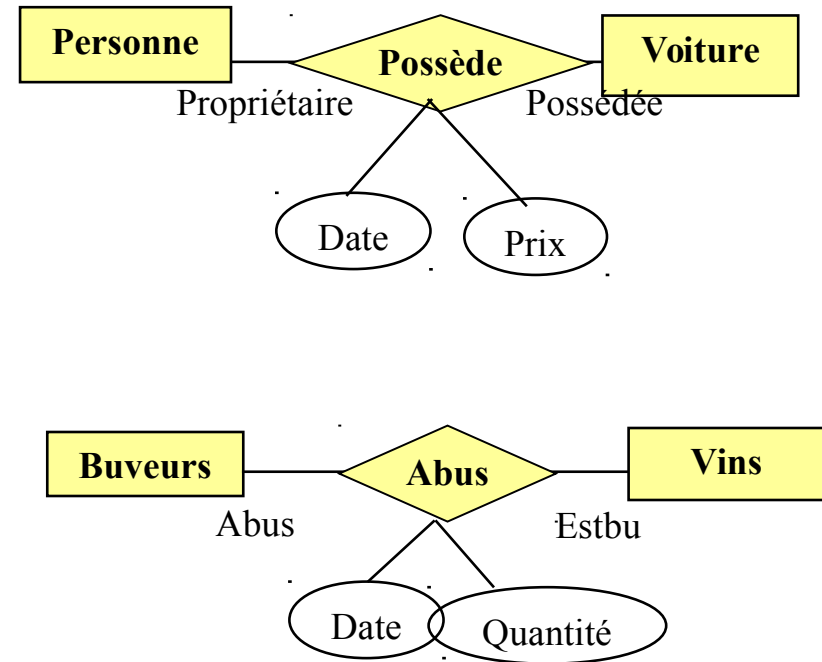


# Représentation

UML

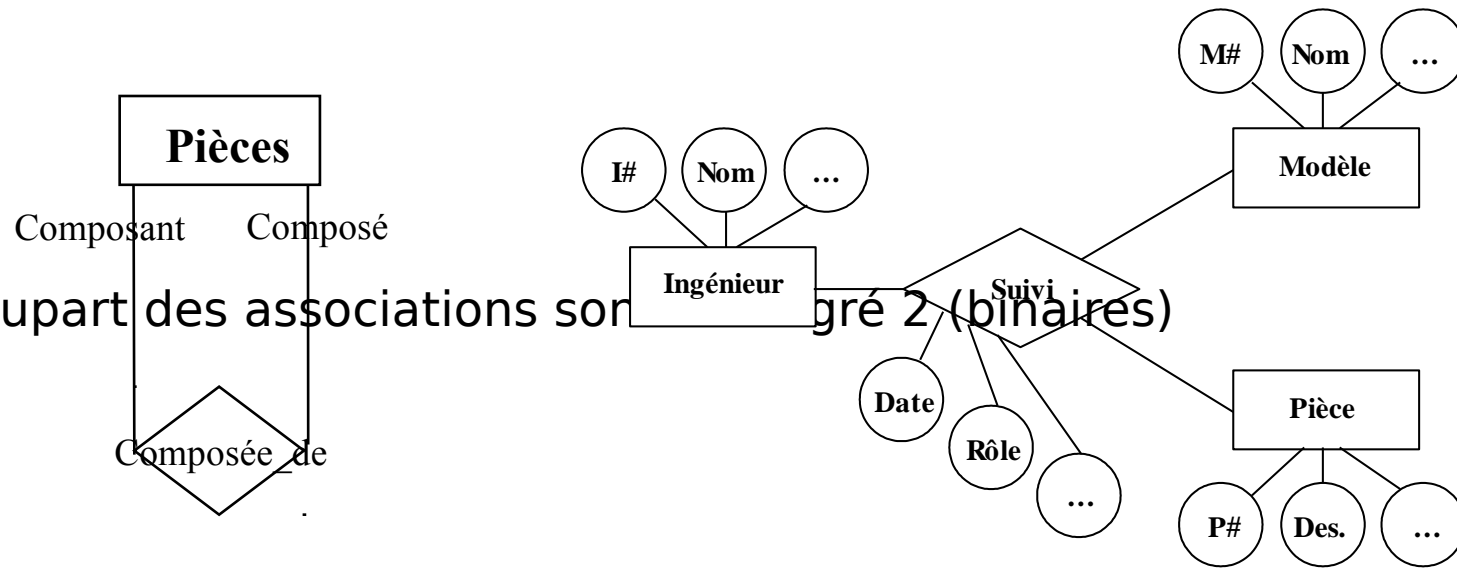


E/R



# Degré d'une association

- La plupart des associations sont de degré 2 (binaires)

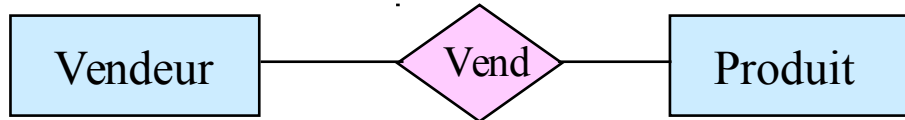
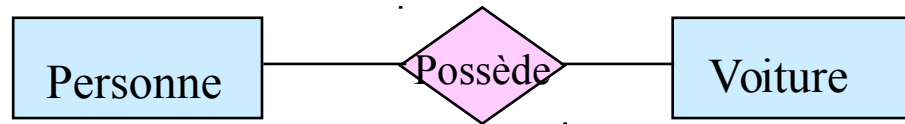
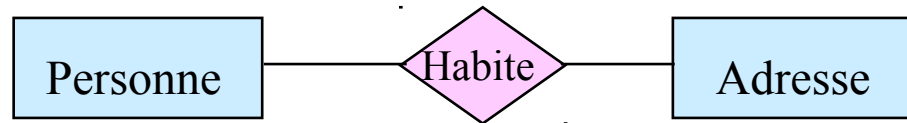


# Cardinalité d'une association

- 1:1

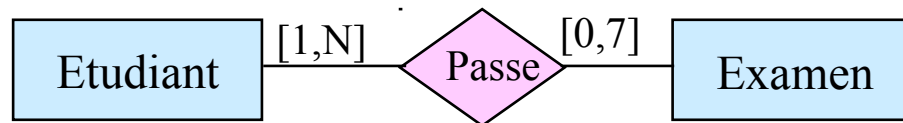
- 1:N

- N:M



# Cardinalités min et max

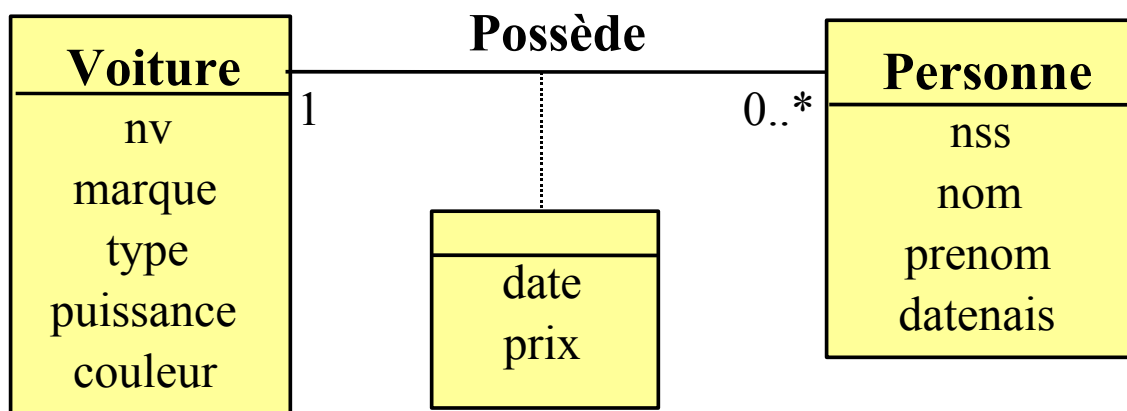
- Cardinalité maximum
  - Indique le nombre maximum d'instances d'une classe d'entité participant à une association
- Cardinalité minimum
  - Indique le nombre minimum d'instances d'une classe d'entité participant à une association



# Cardinalités: notations UML

<u>1</u>	1
<u>*</u>	plusieurs (0 à N)
<u>0..1</u>	optionnel (0 ou 1)
<u>1..*</u>	obligatoire (1 ou plus)
<u>0..*</u> {ord}	ordonné (0 à N)
<u>3..5</u>	limité (de 3 à 5)

# Exemple



# Domaines

- Ensemble nommé de valeurs
  - Un attribut peut prendre valeur dans un domaine
  - Généralisation des types élémentaires
- Exemples
  - Liste de valeurs (1,2,3)
  - Type contraint ( $0 < \text{int} < 100$ )
- Permettent de préciser les valeurs possibles des attributs
- Réduisent les ambiguïtés

# La pratique de la conception

- Bien comprendre le problème à résoudre
- Essayer de conserver le modèle simple
- Bien choisir les noms
- Ne pas cacher les associations sous forme d'attributs
  - utiliser les associations
- Faire revoir le modèle par d'autres
  - définir en commun les objets de l'entreprise
- Documenter les significations et conventions
  - élaborer le dictionnaire



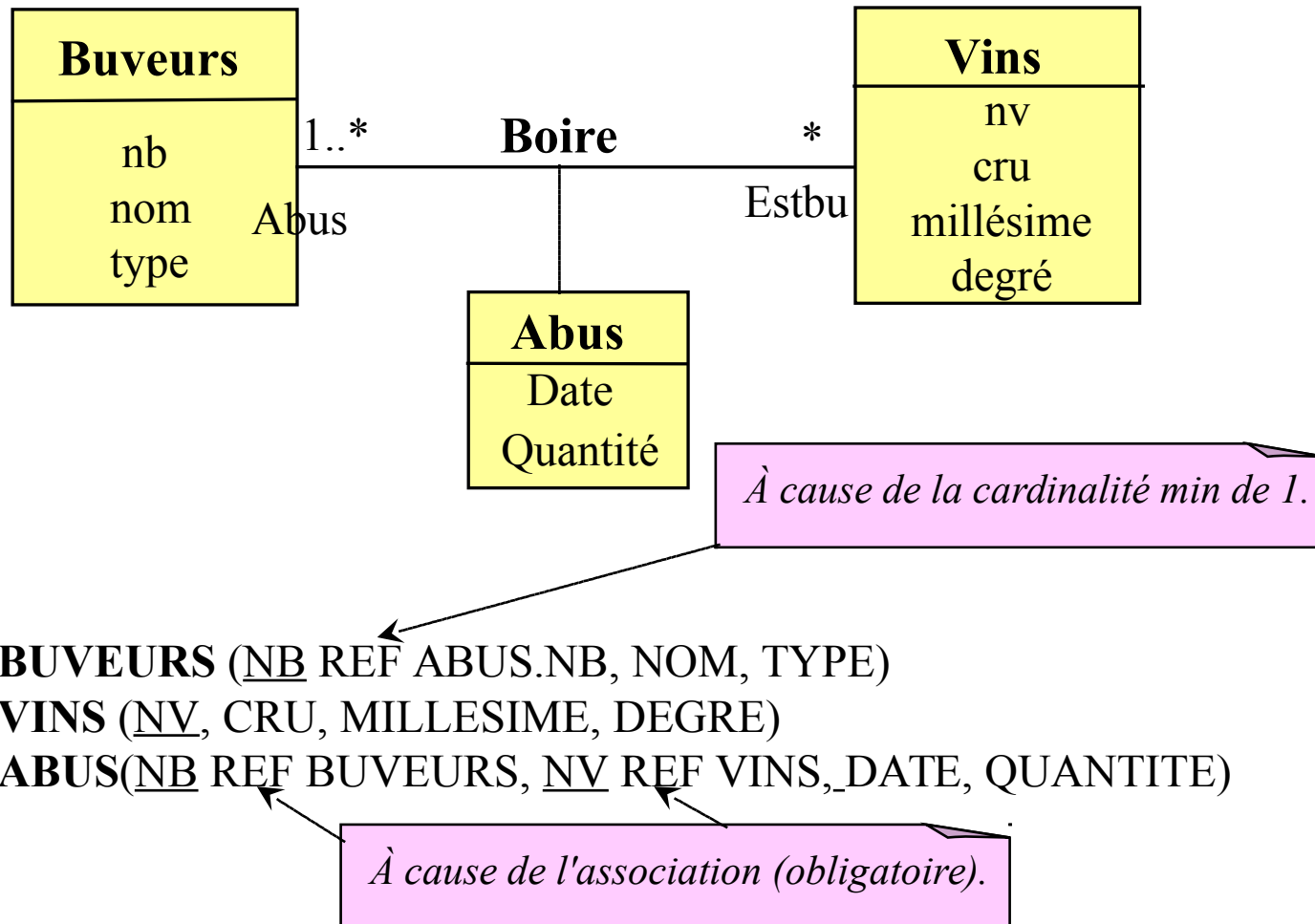
# 3. Passage au relationnel

- Implémentations des entités et associations sous forme de tables
  - mémorisent les états des entités et liens
  - pas nécessaire d'avoir une BD E/R
- Les attributs correspondent aux colonnes des tables
  - nom attribut → nom colonne
  - Ensemble de valeurs → domaine

# Traduction des associations

- Règle de base
  - Une association est représentée par une table dont le schéma est le nom de l'association et la liste des clés des entités participantes suivie des attributs de l'association
  - Exemples :
    - POSSEDE (N° Ss, N° Veh, Date , Prix )
    - ABUS (Nv, Nb, Date, Quantité)
- Amélioration possible
  - Regrouper les associations 1 --> n avec la classe cible
  - Exemple :
    - VOITURE (N°VEH, MARQUE, TYPE, PUISSANCE, COULEUR)
    - POSSEDE (N° SS, N° VEH, DATE , PRIX )
    - regroupés si toute voiture a un et un seul propriétaire

# Exemple



# 6. Conclusion

- Intérêt de l'utilisation d'une méthode de conception
  - proche du monde réel
  - démarche sémantique claire
  - diagramme standards
- Passage au relationnel semi-automatique
  - outils du commerce utilisables (Objecteering, Rose, etc.)
  - supporteront les extensions objet-relationnel à venir
- Extensions à venir avec la conception objet