

Modélisation E/R des Données

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

1

1. Objectifs de la Modélisation

- ◆ Permettre une meilleure compréhension
 - Le monde réel est complexe
 - Concentration sur les aspects cruciaux du problème
- ◆ Réduire la distance sémantique entre le langage des utilisateurs et le langage des concepteurs
 - abstraction du réel perçu en termes compréhensibles et visibles
- ◆ Permettre une conception progressive
 - Abstractions et raffinements successifs, prototypage rapide
- ◆ Automatisation de la construction du schéma de la BD

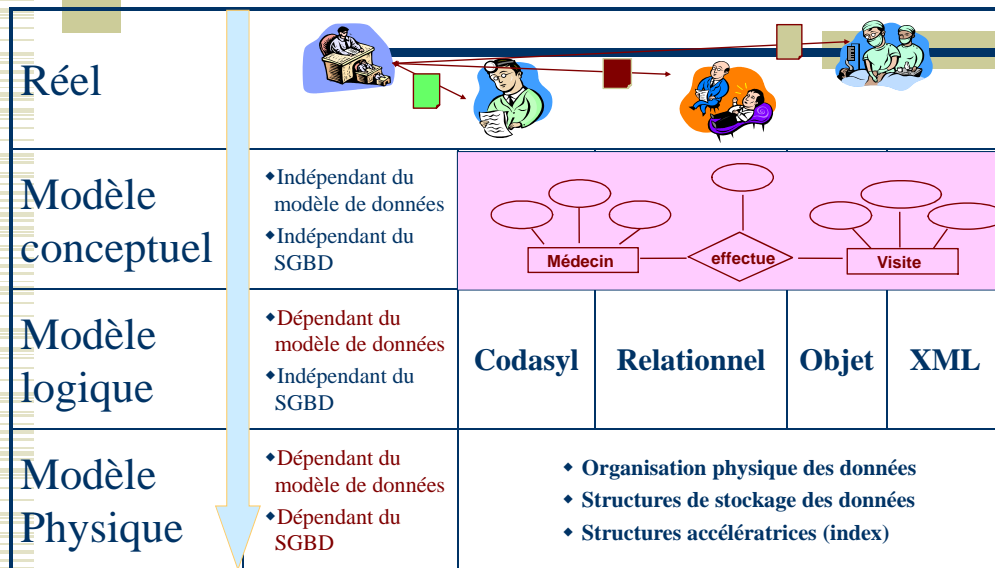
Élaborer un modèle conceptuel

- ◆ 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
 - Découvrir les relations entre ces concepts
- ◆ Faciliter la visualisation du système
 - Diagrammes avec notations simples et précises
 - 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 → Voitures; "Dupont" → "75AB75"

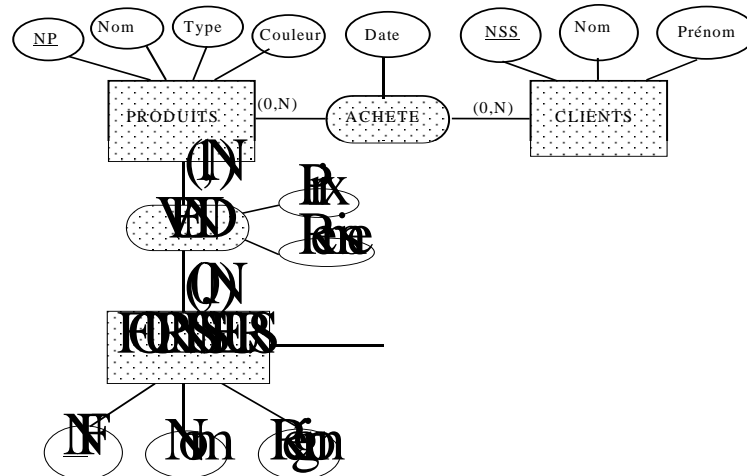
Modélisation à plusieurs niveaux



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

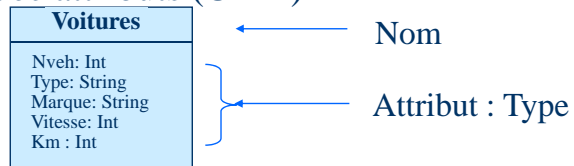


Entité

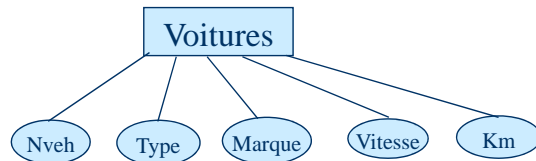
- ♦ Un objet du monde réel qui peut être identifié et que l'on souhaite représenter
 - La classe d'entité correspond à une collection d'entités décrites par leur type commun (le format)
 - L'instance d'entité correspond à un élément particulier de la classe d'entité (un objet)
 - Attention: on dit entité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

Représentation d'une entité

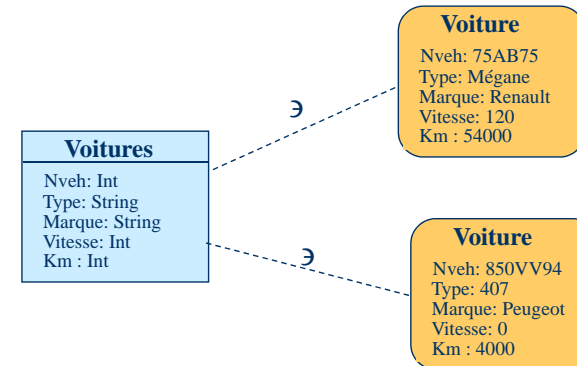
♦ Rectangle avec attributs (UML)



♦ Rectangle avec attributs accrochés (E/R)



Exemple d'instance d'entité



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

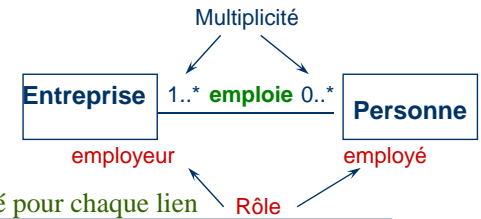
- ♦ Un identifiant aussi appelé clé est un (groupe d') attribut qui permet de retrouver une instance d'entité unique à tout instant parmi celles de la classe.
 - Exemple: Nveh dans Voitures, NSS dans Personnes
- ♦ Un identifiant peut être constitué de plusieurs attributs (clé composée)
 - Exemple:
 -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

Association

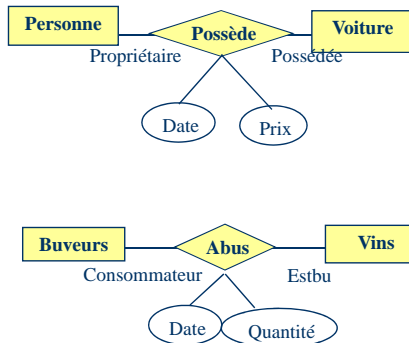
- ♦ Les entités sont reliées ensemble par des associations
 - Entre instances: exemple 1 véhicule est associé à 1 personne
 - Entre classes: abstraction des associations entre instances
- ♦ Une association peut avoir des attributs (propriétés)
 - Ex: la date de l'achat

Association: quelques définitions

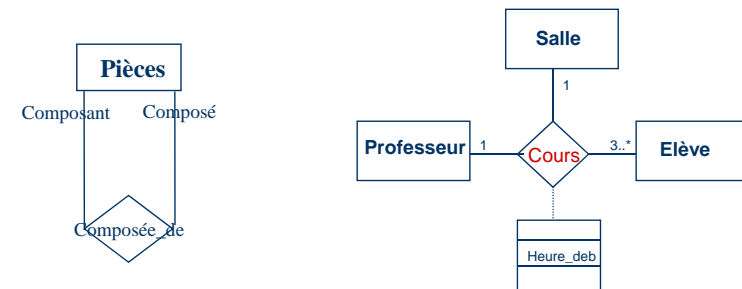
- ♦ Association (Association)
 - Une relation entre des instances de deux (ou plusieurs) classes
- ♦ Lien (Link)
 - Une instance d'association
- ♦ Rôle (Role)
 - Une extrémité d'une association
- ♦ Attribut de lien (Link attribute)
 - Un attribut de l'association instancié pour chaque lien
- ♦ Cardinalité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



Représentation E/R



Degré d'une association



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

Cardinalités min et max

♦ Cardinalité maximum

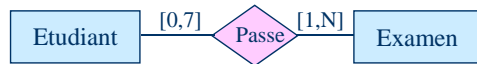
- Indique le nombre maximum d'instances d'une classe d'entité participant à une association

♦ Cardinalité minimum

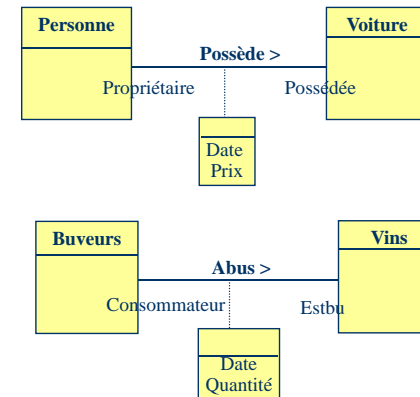
- Indique le nombre minimum d'instances d'une classe d'entité participant à une association

♦ Se lit

- Un étudiant passe entre 0 et 7 examens
- Un examen est passé par au moins un étudiant, sans limite supérieure de nombre



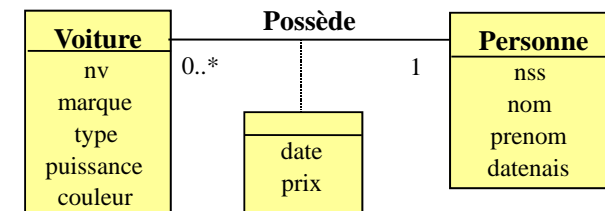
Représentation UML



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>	(0 à N), identique à *
<u>3..5</u>	limité (de 3 à 5)

Exemple en UML

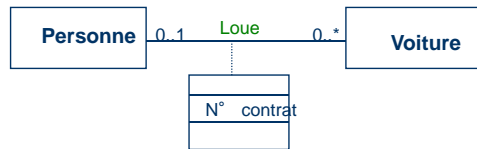


Se lit:

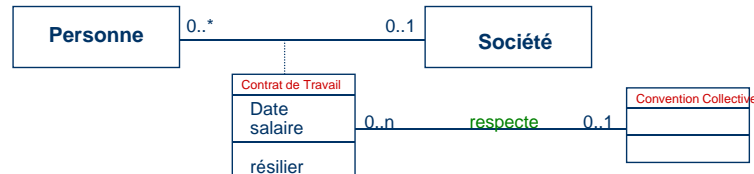
- Une voiture est possédée par une seule personne
- Une personne peut posséder un nombre quelconque de voitures
- ATTENTION : les cardinalités sont indiquées sur l'extrémité inverse du lien par rapport à E/R

Associations ...

- ♦ **Attribut d'association** = propriété du lien entre deux objets



- ♦ **Classe d'association** = élément ayant à la fois les propriétés d'une classe et d'une association



Agrégation et Composition

- ♦ **Agrégation**

- association particulière spécifiant une relation « tout-partie » entre l'agrégat et un composant.

- Notée ◇

- ♦ **Composition**

- forme forte d'agrégation avec cycle de vie des parties lié à celui du composite (la destruction du composite entraîne celle des composants)

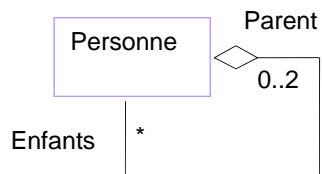
- Notée ◆

- ♦ **Distinction utile pour exprimer des contraintes d'intégrité**

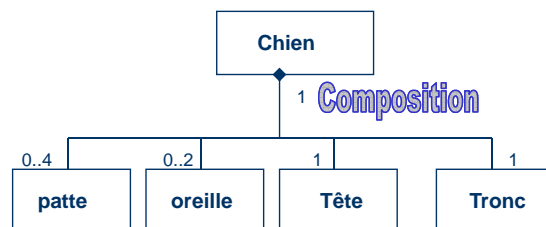
Exemples



Agrégation



<S'occupe de



Composition

Hiérarchies de classes

- ♦ **Gérer la complexité**

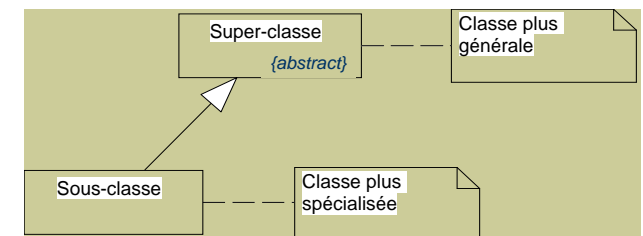
- Arborescences de classes d'abstraction croissante

- ♦ **Généralisation**

- Super-classes

- ♦ **Spécialisation**

- Sous-classes

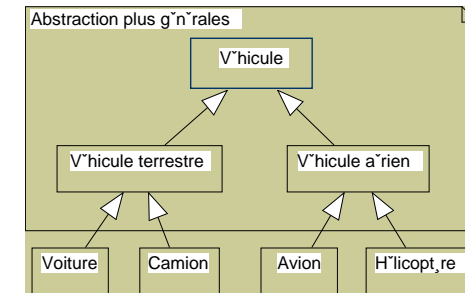


Généralisation

- ♦ **Généralisation** = relation taxonomique entre un élément plus général et un élément plus spécifique qui est entièrement conforme avec le premier élément, et qui ajoute de l'information supplémentaire
- ♦ **Héritage** = mécanisme par lequel des éléments plus spécifiques incorporent la structure et le comportement d'éléments plus généraux

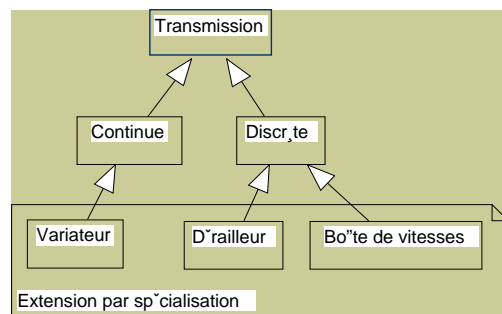
Généralisation

- ♦ Factoriser les éléments communs
 - attributs, opérations et contraintes



Spécialisation

- ♦ Extension cohérente d'un ensemble de classes



Domaines

- ♦ Ensemble nommé de valeurs
 - Un attribut peut prendre sa valeur dans un domaine
 - Généralisation des types élémentaires
- ♦ Exemples
 - Liste de valeurs (1,2,3)
 - Type contraint ($0 < \text{val} < 100$)
- ♦ Permettent de préciser les valeurs possibles des attributs
- ♦ Utile pour déclarer des contraintes d'intégrité

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

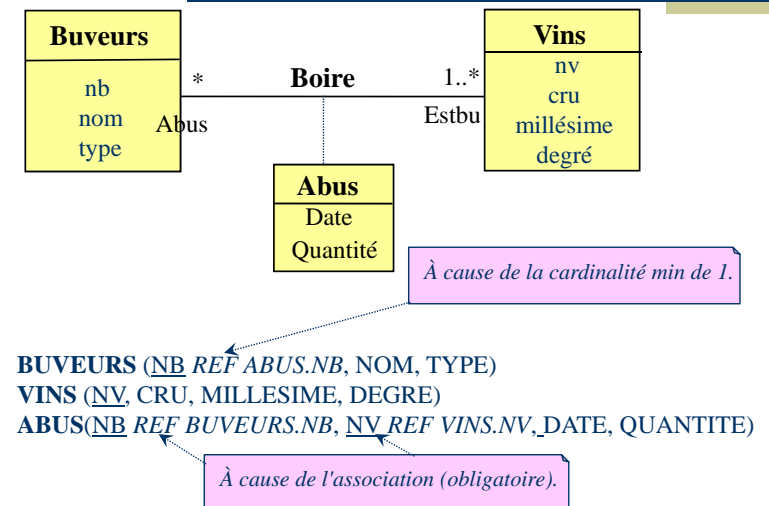
3. Passage au relationnel

- ♦ Implémentations des entités et associations sous forme de tables
 - mémorisent les états des entités et liens
- ♦ Les attributs correspondent aux colonnes des tables
 - nom attribut → nom colonne
 - Ensemble de valeurs → domaine
- ♦ Exemple
 - **Etudiant** (Ne: integer, Nom: string, Prénom: string, Age: integer)

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
 - diagrammes standards
- ♦ Passage au relationnel semi-automatique
 - outils du commerce utilisables (Objecteering, Rose, etc.)