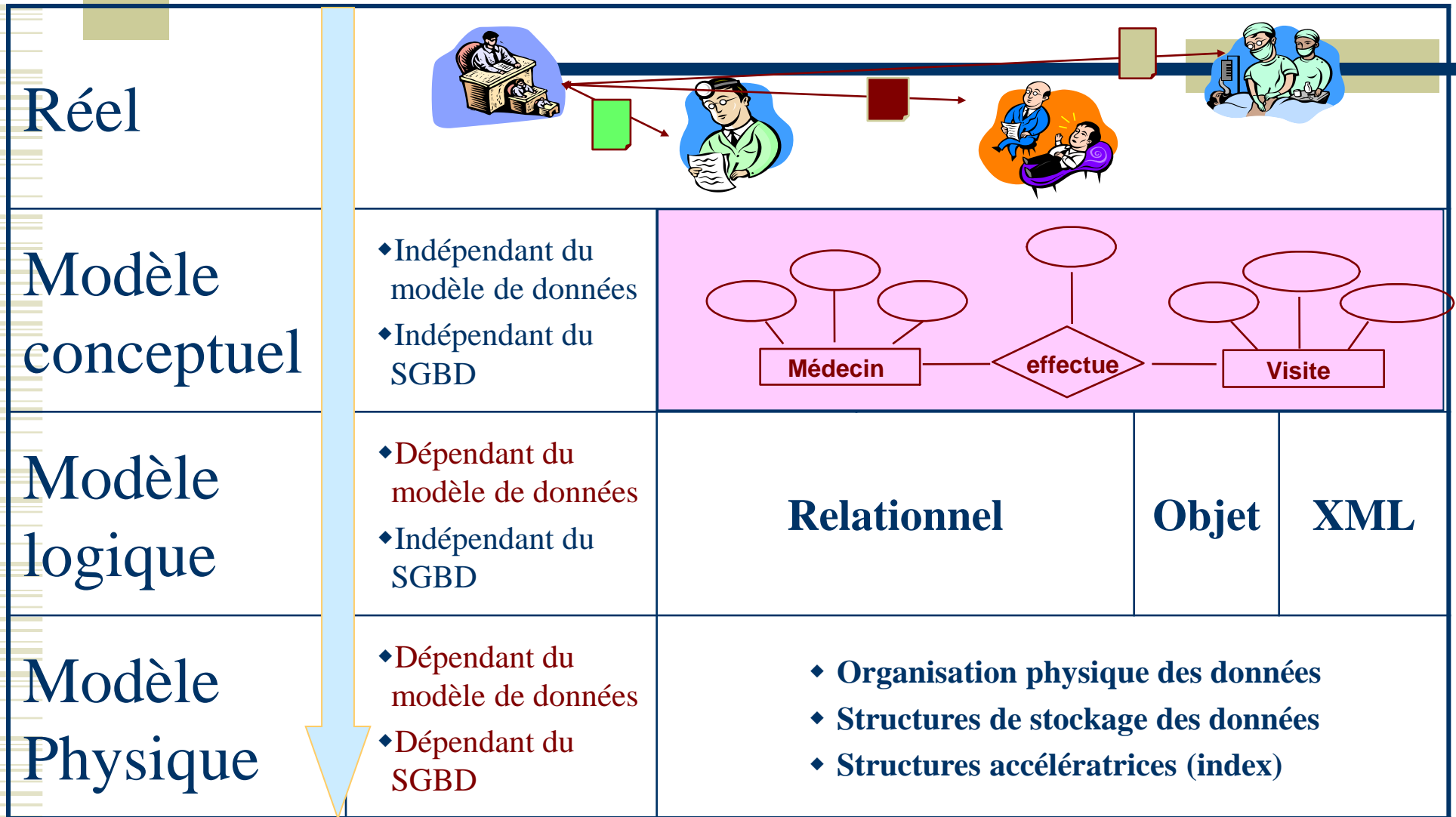


Modélisation E/R des Données

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

Modélisation à plusieurs niveaux



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
 - ◆ Exemple : Type Personne (nom: String, Prenom: String, age: Réel)
- ◆ Possibilité d'intégrer des relations entre données (liens)
 - ◆ Exemple : Personne → Voitures;
- ◆ Le schéma (abstrait) est utilisé pour créer de véritables objets du monde réel (instantiation)
 - Instance ou occurrence : Personne("Dupont", "Jules", 20)
 - Ensemble de Voitures {id:String}: Voitures {"75AB75", "1200VV94"}
 - Création d'une relation entre une personne et une voiture : "Dupont" → "75AB75"

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

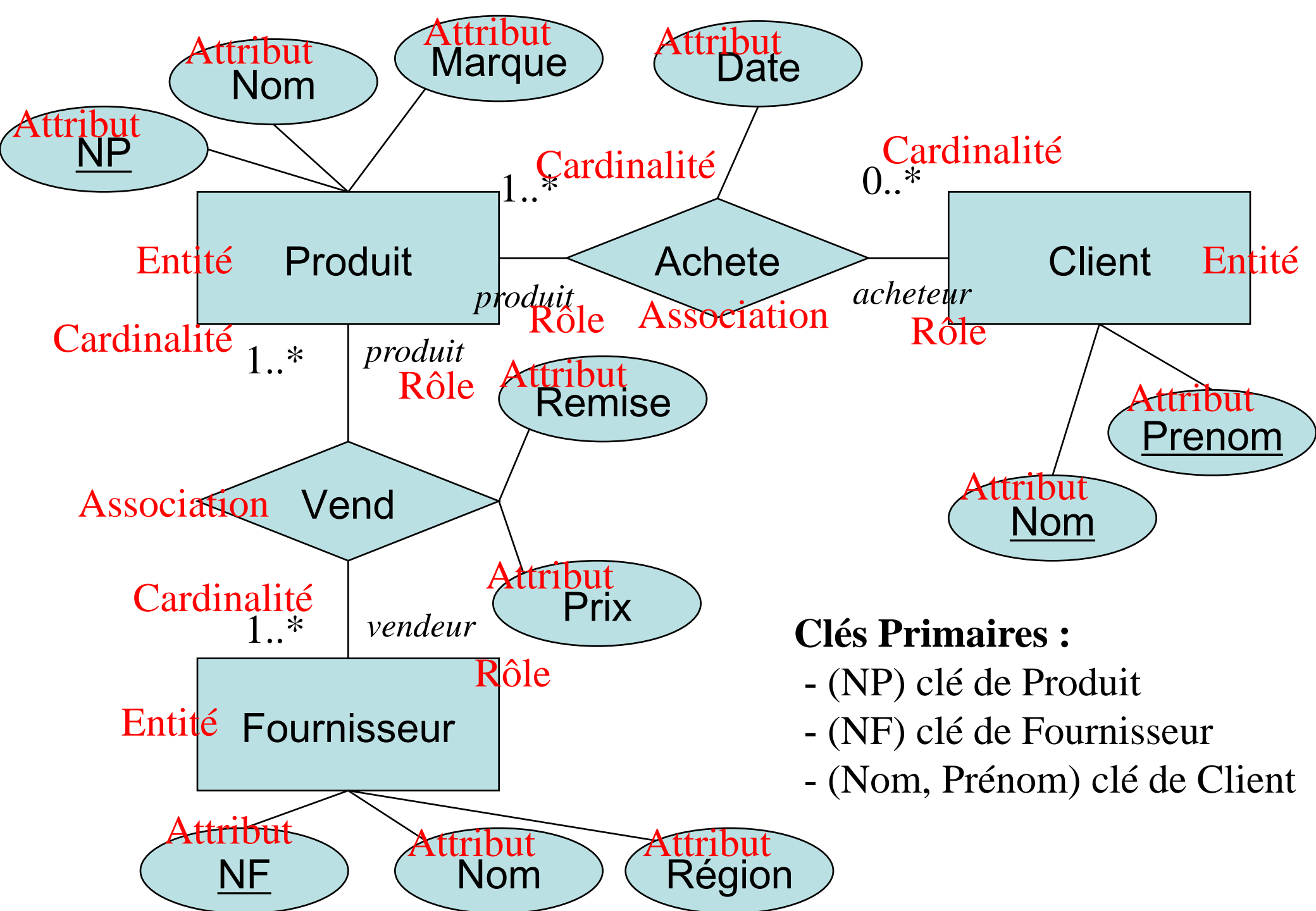
- ◆ Réduire la distance sémantique entre le langage des utilisateurs et le langage des concepteurs
 - meilleure communication entre utilisateurs et concepteurs
 - abstraction du réel perçu en termes compréhensibles et visibles
- ◆ Regrouper l'analyse des données et des traitements
 - meilleure compréhension 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 Peter Chen dans :
The Entity-Relationship model, towards a unified view of data, in
ACM Transactions on Database Systems, 1(1), pp 9-36, 1976
- ◆ Etendu vers E/R généralisé puis vers l'objet



Un exemple de modèle Entité-Association

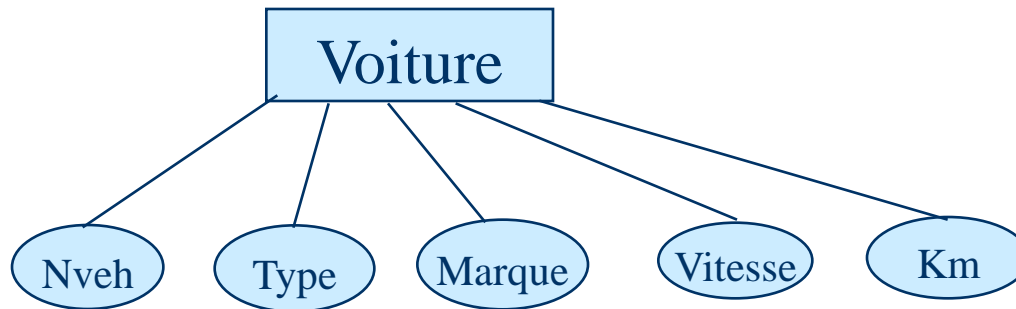


A- Classe d'Entité

- ◆ 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 *instances* d'entités dans une *classe* d'entité.

Classe d'Entité : Représentation

- ◆ Rectangle avec attributs accrochés (E/R)



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 (voir pbs de normalisation, cours master)

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
- ◆ **Clés candidates et clés primaires**
 - Une clé candidate est un ensemble d'attributs *permettant* d'identifier de manière unique une instance d'une entité
 - Parmi les clés candidates, on en choisit une, qu'on nomme clé primaire qui *va* identifier l'entité

B- Association

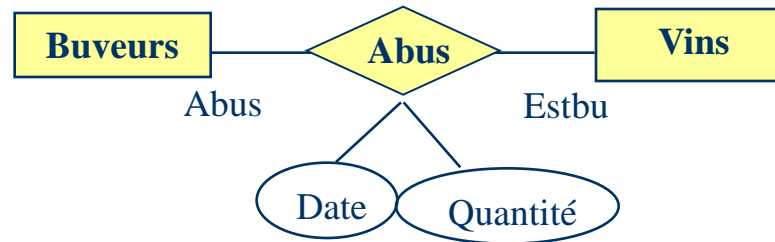
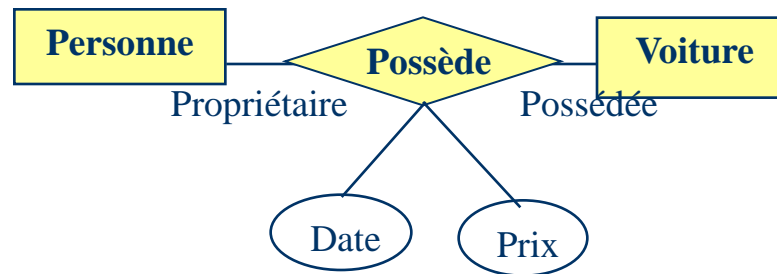
- ◆ Les entités sont reliées ensemble par des associations
 - Entre instances: par exemple 1 véhicule est associé à 1 personne
 - Entre classes: abstraction des associations entre instances
- ◆ Une association peut avoir des attributs (propriétés)
- ◆ Elle peut relier plusieurs entités ensemble
- ◆ Il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

Association: quelques définitions

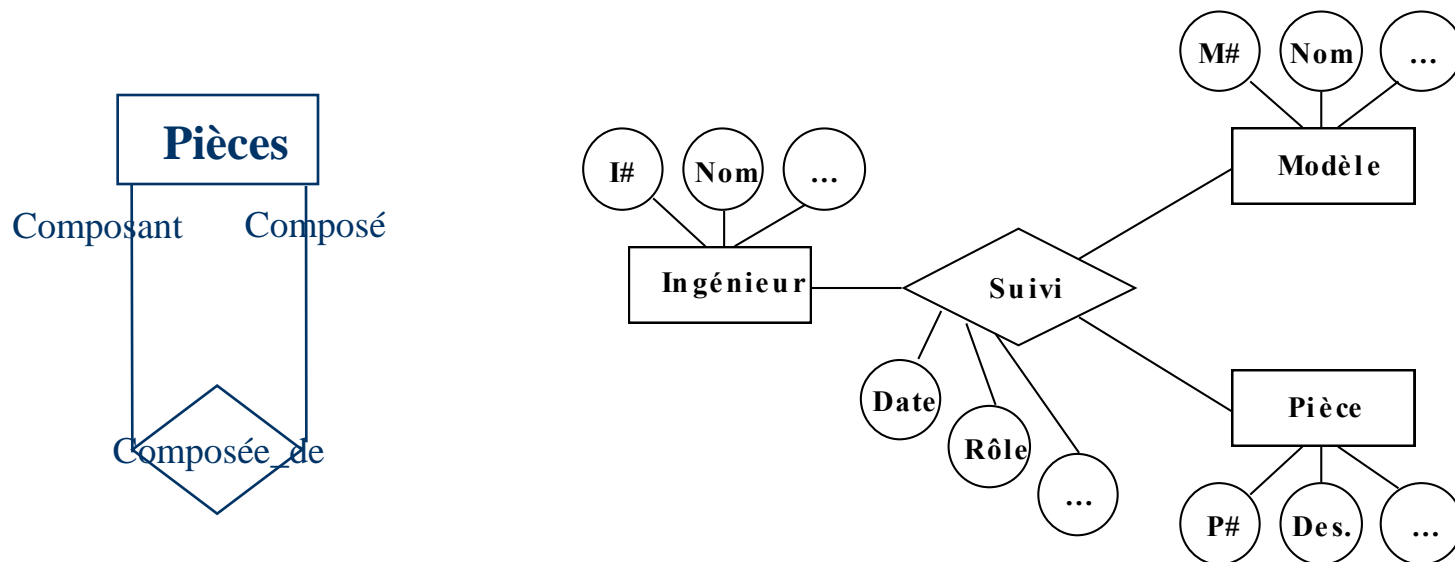
- ◆ Association (Association)
 - Une relation entre des instances de deux (ou plus) classes
- ◆ Lien (Link)
 - Une instance d'association
- ◆ Rôle (Role)
 - Une extrémité d'une association
- ◆ Attribut de lien (Link attribute)
 - Un attribut de l'association instancié pour chaque lien
- ◆ Cardinalité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

Représentation

E/R



Degré d'une association



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

Cardinalité (max) d'une association

◆ 1:1



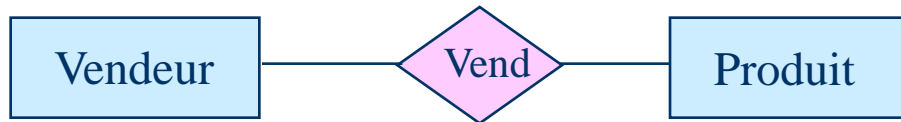
one-to-one

◆ 1:N



one-to-many

◆ N:M



many-to-many

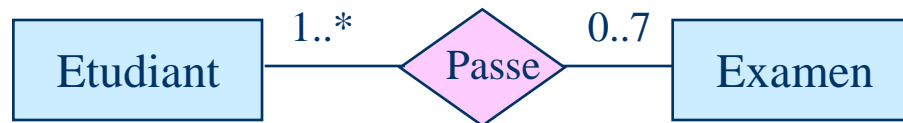
Cardinalités min et max (Notations UML)

♦ 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



Domaines

- ◆ Ensemble nommé de valeurs
 - Un attribut peut prendre des 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

3. Conclusion :

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

Pour aller plus loin ...

- ◆ 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