
Introduction aux Bases de Données

Master Bio Informatique 1^{ère} année
Patricia Serrano Alvarado

Plan

1. Introduction
2. Objectifs des SGBD
3. Architectures des SGBD
4. Applications traditionnelles des SGBD

1. Introduction

- Les entreprises gèrent des volumes de données très grands
 - Giga, Terra, Péta –octets
 - Numériques, Textuelles, Multi-média (images, films,...)
- Il faut pouvoir facilement
 - Archiver les données sur mémoires secondaires permanentes
 - Retrouver les données pertinentes à un traitement
 - Mettre à jour les données variant dans le temps
- Les données sont structurées et identifiées
 - Données élémentaires ex: Votre salaire, Votre note en BD
 - Données composées ex: Votre CV, vos résultats de l'année
 - Identifiant humain ex: NSS ou machine: P26215
- Qu'est-ce qu'une BD ?
 - Collection de données structurées reliées par des relations
 - Interrogeable et modifiable par des langages de haut niveau

Un peu d'histoire

- Années 60:
 - Réceptifs logique de données → fichiers sur disque
 - Accès séquentiel puis sur clé
 - Lire (Nomf, Article), Ecrire (Nomf, Article)
 - Lire (Nomf, Article, Clé), Ecrire (Nomf, article, Clé)
- Années 70:
 - Avènement des Bases de Données Réseaux (BD)
 - Ensemble de fichiers reliés par des pointeurs
 - Langage d'interrogation par navigation
- Années 80:
 - Avènement des Bases de Données Relationnelles (BDR)
 - Relations entre ensemble de données
 - Langage d'interrogation par assertion logique

Systemes de fichiers

Caractéristiques



Comptabilité



Chirurgie

Consultations



Psychiatrie



Problèmes

Format des fichiers



Dupont

Symptomes : y
Turlututu : sqj
Symptomes : y
Turlututu : sdd
Analyses : xxx

Dupond

Turlututusqjisk
Symptom: yyyy
Analyses xxxx

Turlututudhjsd
Analyses :xx



Duhpon

Symptomes : yy
Analyses : xxxx
Symptomes : yy

Duipont

Turlututu : sq
Symptomyyyy
Analysesxxxx

Turlututudhjsd



Caractéristiques

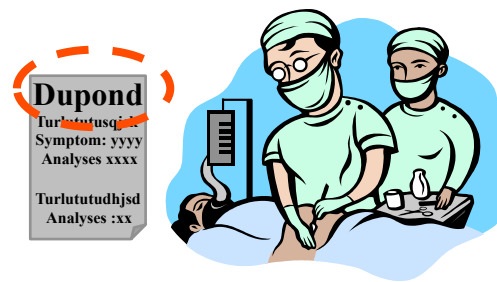
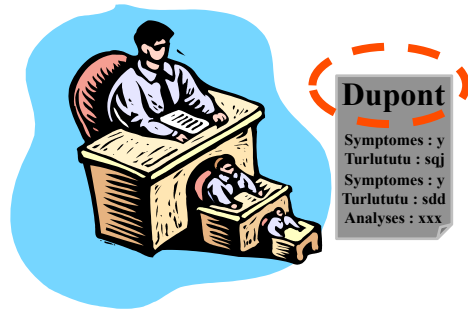
Plusieurs applications

- plusieurs formats
- plusieurs langages

Problèmes

- Difficultés de gestion

Redondance (données)



Caractéristiques

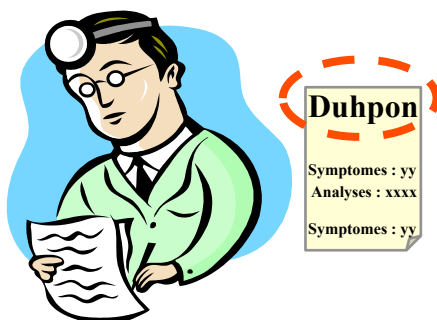
Plusieurs applications

- plusieurs formats
- plusieurs langages

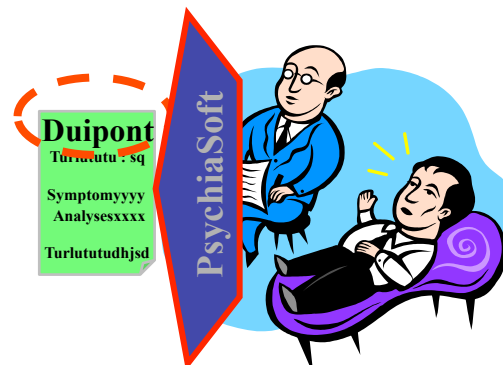
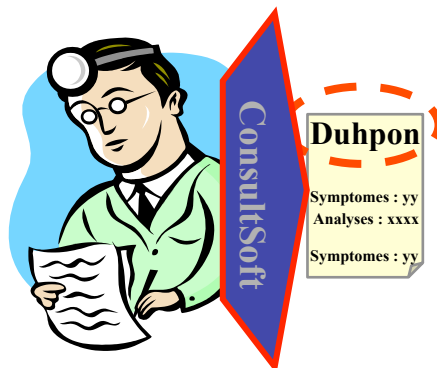
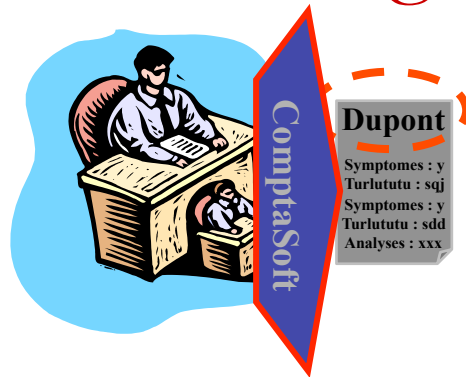
Redondance de données

Problèmes

- Difficultés de gestion
- Incohérence des données



Interrogations



Caractéristiques

Plusieurs applications

- plusieurs formats
- plusieurs langages

Redondance de données

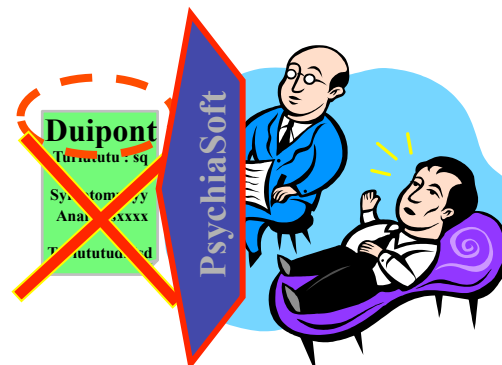
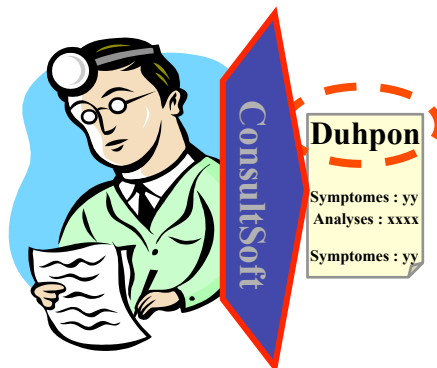
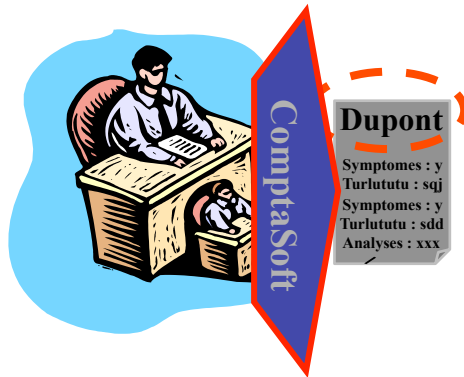
Pas de facilité d'interrogation

- Question ⇒ développement

Problèmes

- Difficultés de gestion
- Incohérence des données
- Coûts élevés
- Maintenance difficile

Pannes



Caractéristiques

Plusieurs applications

- plusieurs formats
- plusieurs langages

Redondance de données

Pas de facilité d'interrogation

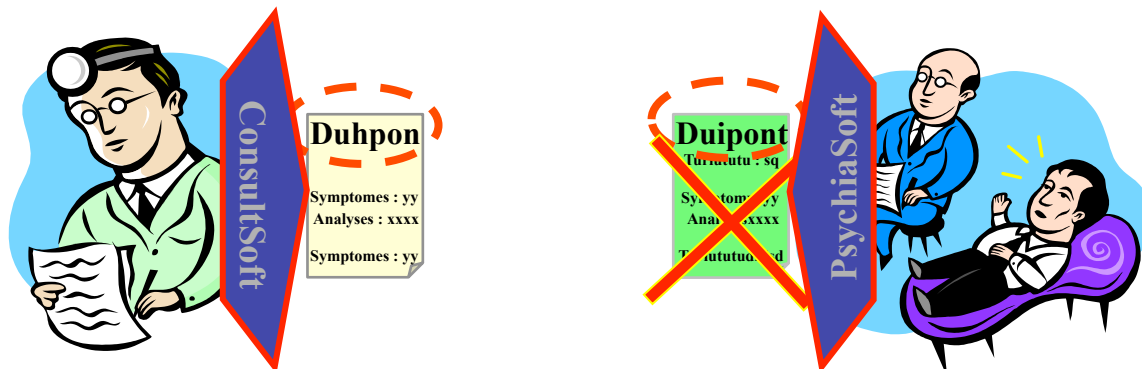
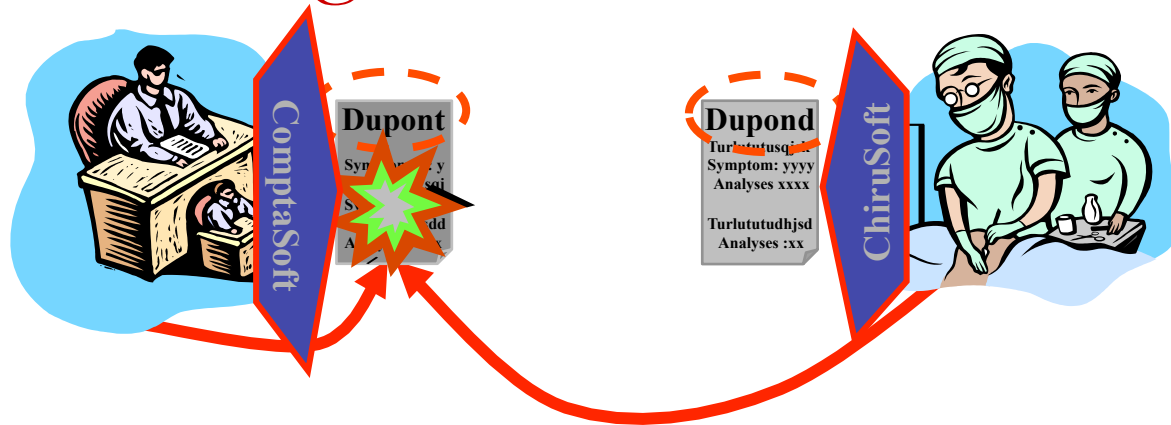
- Question ⇒ développement

Redondance de code

Problèmes

- Difficultés de gestion
- Incohérence des données
- Coûts élevés
- Maintenance difficile
- Gestion de pannes ???

Partage de données



Caractéristiques

Plusieurs applications

- plusieurs formats
- plusieurs langages

Redondance de données

Pas de facilité d'interrogation

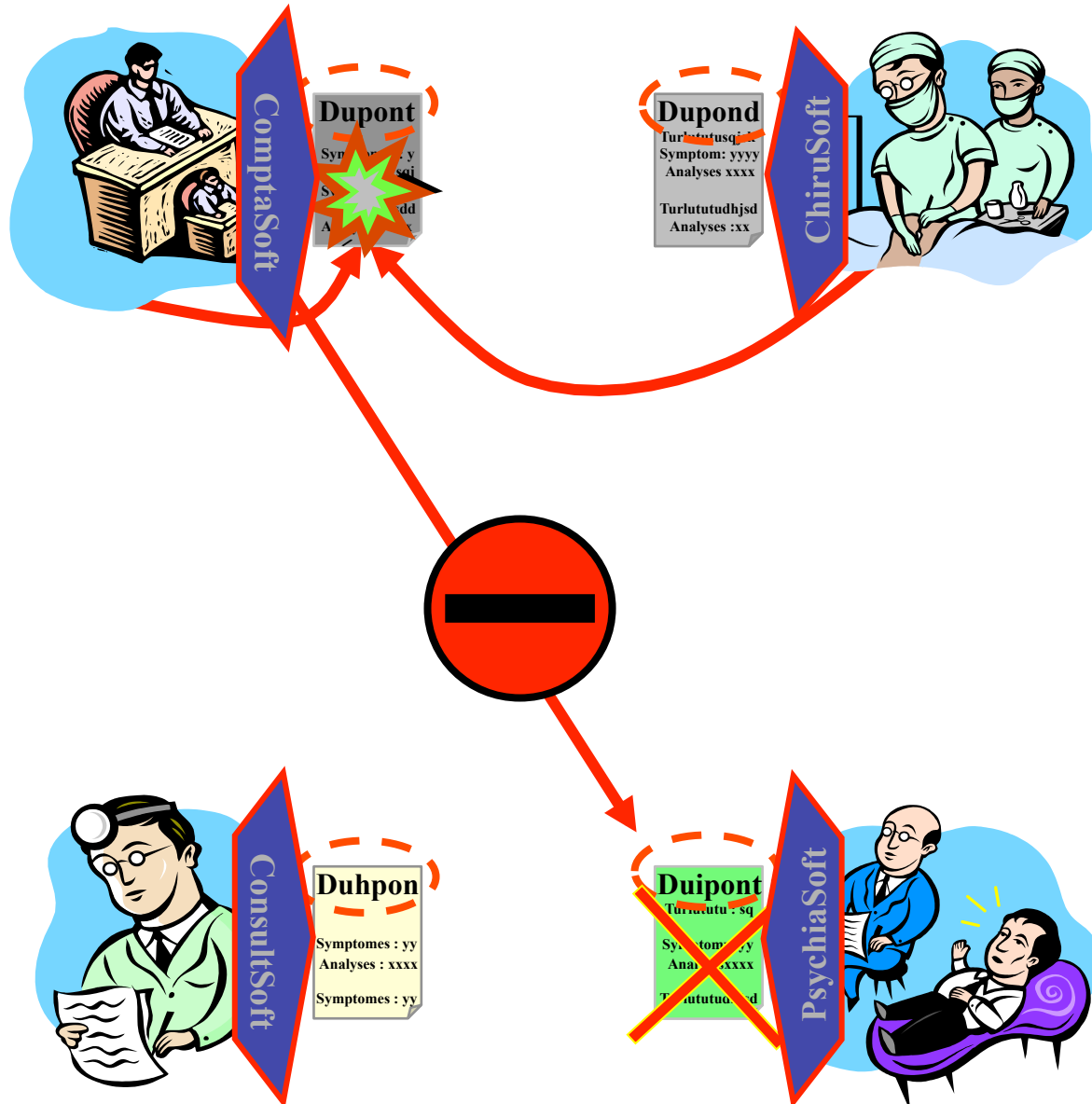
- Question ⇒ développement

Redondance de code

Problèmes

- Difficultés de gestion
- Incohérence des données
- Coûts élevés
- Maintenance difficile
- Gestion de pannes ???
- Partage des données ???

Confidentialité



Caractéristiques

Plusieurs applications

- plusieurs formats
- plusieurs langages

Redondance de données

Pas de facilité d'interrogation

- Question ⇒ développement

Redondance de code

Problèmes

- Difficultés de gestion
- Incohérence des données
- Coûts élevés
- Maintenance difficile
- Gestion de pannes ???
- Partage des données ???
- Confidentialité ???

L'approche ' 'Bases de données' '

- **Modélisation des données**

- Eliminer la **redondance** de données
- **Centraliser** et **organiser** correctement les données
- Plusieurs niveaux de modélisation
- Outils de conception

- **Logiciel «**S**ystème de **G**estion de **B**ases de **D**onnées»**

- **Factorisation** des modules de contrôle des applications
 - Interrogation, cohérence, partage, gestion de pannes, etc...
- Administration facilitées des données

Modélisation du réel

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

Modélisation Relationnelle (1)

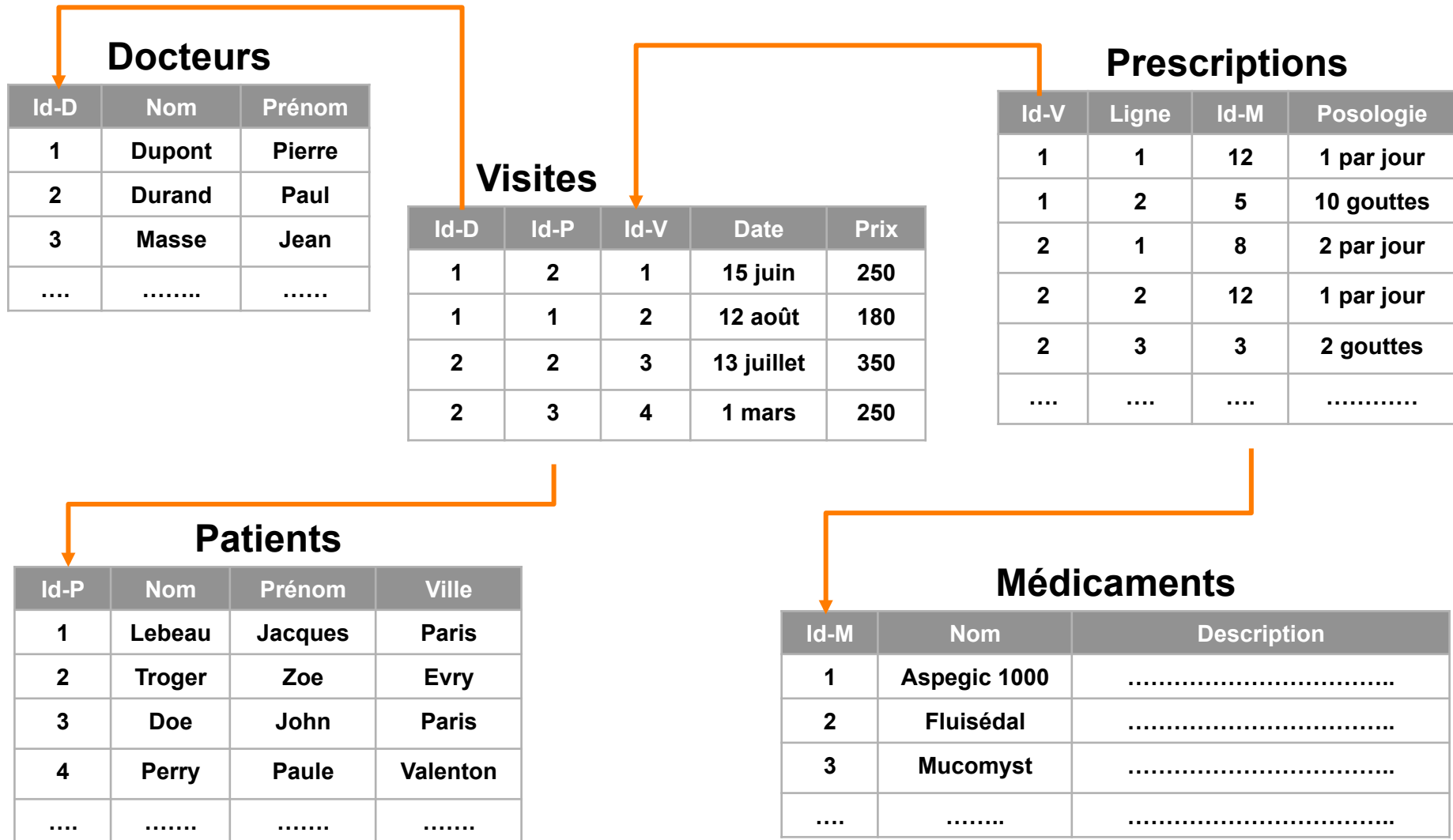
Relation ou table

Champs, attributs,
colonnes

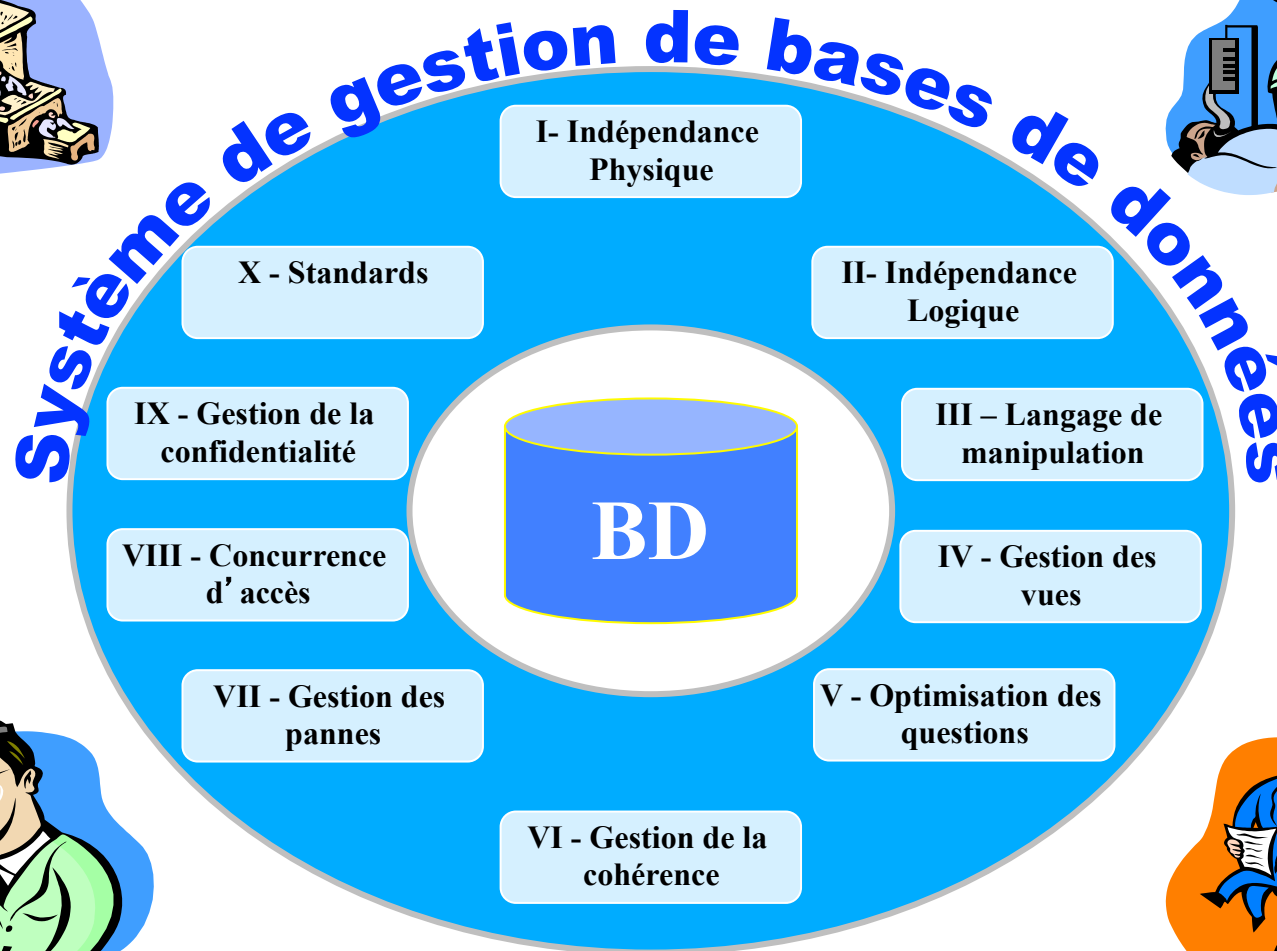
Id-D	Nom	Prénom
1	Dupont	Pierre
2	Durand	Paul
3	Masse	Jean
....

Tuples, lignes ou
n-uplets

Modélisation Relationnelle (2)



2. Objectifs des SGBD



I - Indépendance Physique

- **Indépendance des programmes d'applications vis à vis du modèle physique :**
 - Possibilité de modifier les **structures de stockage** (fichiers, index, chemins d'accès, ...) sans modifier les programmes;
 - Ecriture des applications par des **non-spécialistes des fichiers** et des structures de stockage;
 - Meilleure **portabilité** des applications et **indépendance** vis à vis du matériel.

II - Indépendance Logique

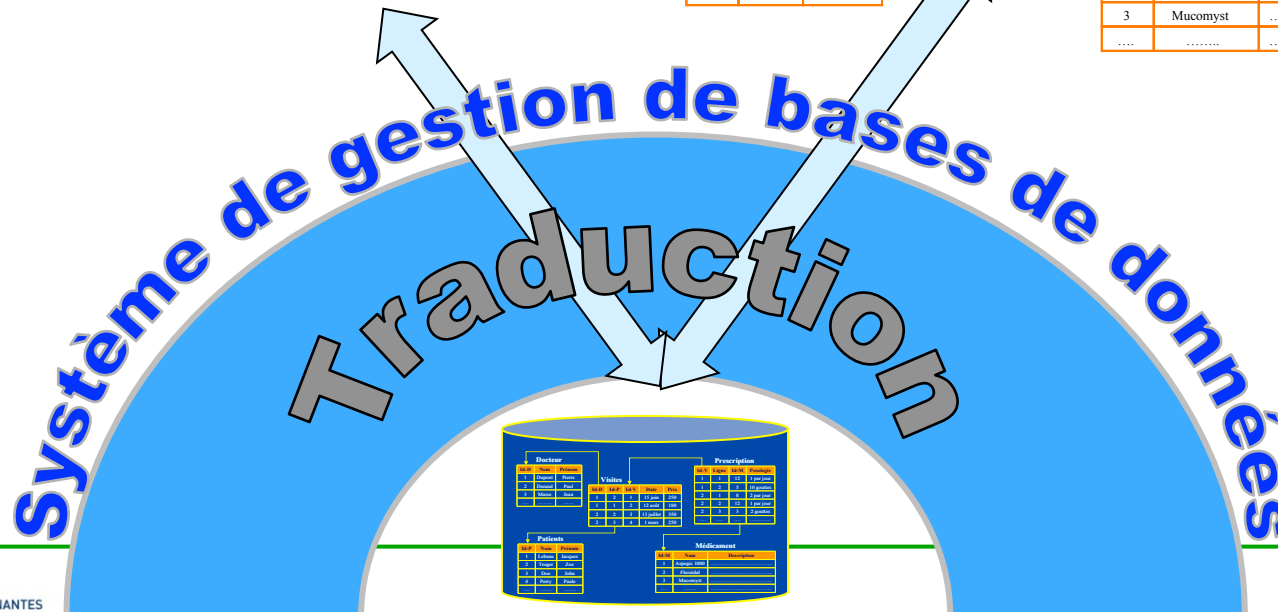
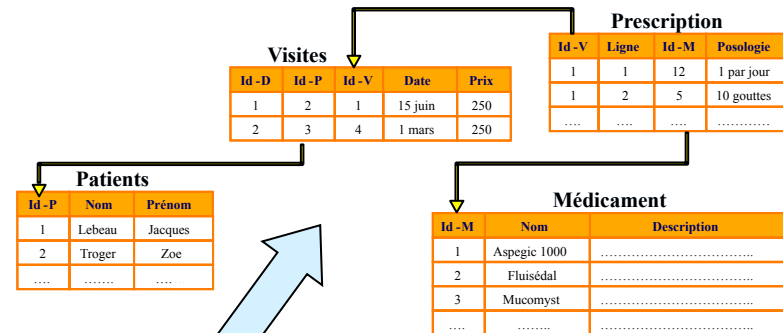
Les applications peuvent définir des **vues logiques** de la BD

Gestion des médicaments

Nombre_Médicaments

Id-M	Nom	Description	Nombre
1	Aspegic 1000	30
2	Fluisédal	20
3	Mucomyst	230
....

Cabinet du Dr. Masse



Avantages de l'indépendance logique

- Possibilité pour chaque application **d'ignorer** les besoins des autres (bien que partageant la même BD).
- Possibilité **d'évolution de la base de données** sans réécriture des applications :
 - Ajout de champs, ajout de relation, renommage de champs.
- Possibilité **d'intégrer des applications existantes** sans modifier les autres.
- Possibilité de limiter les conséquences du partage : **Données confidentielles.**

III - Manipulation aisée

- La manipulation se fait via un langage **déclaratif**
 - La question déclare l'objectif sans décrire la méthode
 - Le langage suit une norme commune à tous les SGBD
 - **SQL : Structured Query Language**
- Sémantique
 - Logique du 1er ordre ++
- Syntaxe (juste un aperçu...)
 - SELECT <structure des résultats>
 - FROM <relations>
 - WHERE <conditions>

IV – Des vues multiples des données

- Les vues permettent d'implémenter l'indépendance logique en permettant de créer des **relations virtuelles**
- Vue = Question/requête stockée
- Le SGBD stocke la **définition** et non le résultat
- Exemple :
 - La vue des patients parisiens
 - La vue des docteurs avec leurs patients
 - La vue des services statistiques
 - ...

V – Exécution et Optimisation

- Traduction **automatique** des questions déclaratives en programmes procéduraux :
 - ➔ Utilisation de l'algèbre relationnelle
- Optimisation **automatique** des questions
 - ➔ Utilisation de l'aspect déclaratif de SQL
 - ➔ Gestion centralisée des chemins d'accès (index, hachages, ...)
 - ➔ Techniques d'optimisation poussées
- Economie de l'astuce des programmeurs
 - Milliers d'heures d'écriture et de maintenance de logiciels.

VI - Intégrité Logique

- **Objectif : Détecter les mises à jour erronées**
- Contrôle sur les données élémentaires
 - Contrôle de types: ex: Nom alphabétique
 - Contrôle de valeurs: ex: Salaire mensuel entre 5 et 50kf
- Contrôle sur les relations entre les données
 - Relations entre données élémentaires:
 - Prix de vente > Prix d'achat
 - Relations entre objets:
 - Un électeur doit être inscrit sur une seule liste électorale

Contraintes d'intégrité des données

■ Avantages :

- **Simplification** du code des applications
- **Sécurité renforcée** par l'automatisation
- **Mise en commun** des contraintes

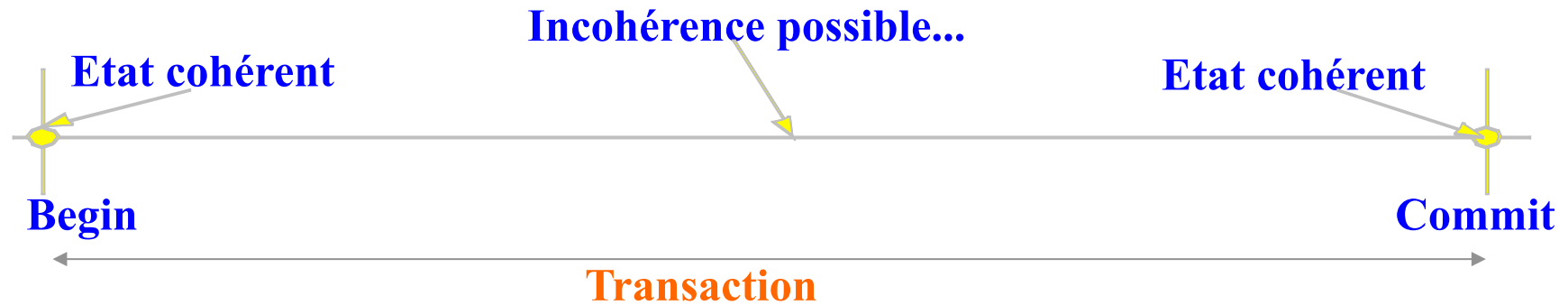
■ Nécessite :

- Un langage de définition de contraintes d'intégrité
- La vérification **automatique** de ces contraintes

VII - Intégrité Physique

- **Motivations : Tolérance aux fautes**
 - Transaction Failure : Contraintes d'intégrité, Annulation
 - System Failure : Panne de courant, Crash serveur ...
 - Media Failure : Perte du disque
 - Communication Failure : Défaillance du réseau
- **Objectifs :**
 - Assurer l'**atomicité** des transactions
 - Garantir la **durabilité** des effets des transactions commises
- **Moyens :**
 - Journalisation : Mémorisation des **états successifs** des données
 - Mécanismes de reprise

Transaction



Begin

$$CEpargne = CEpargne - 3000$$

$$CCourant = CCourant + 3000$$

Commit T1

Atomicité et Durabilité

ATOMICITE

Begin

CEpargne = CEpargne - 3000

CCourant = CCourant + 3000

Commit T1

Panne

→ Annuler le débit !!

DURABILITE

Begin

CEpargne = CEpargne - 3000

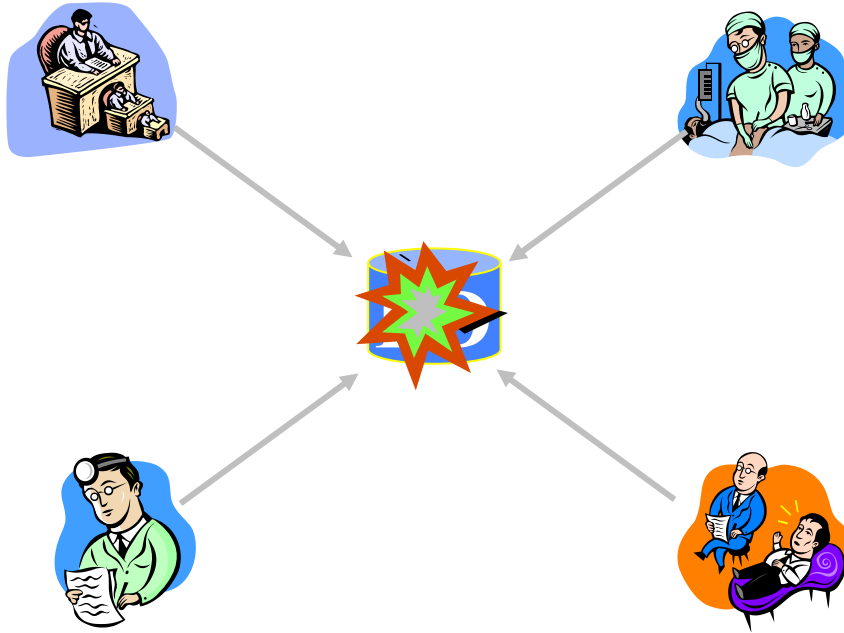
CCourant = CCourant + 3000

Commit T1

Crash disque

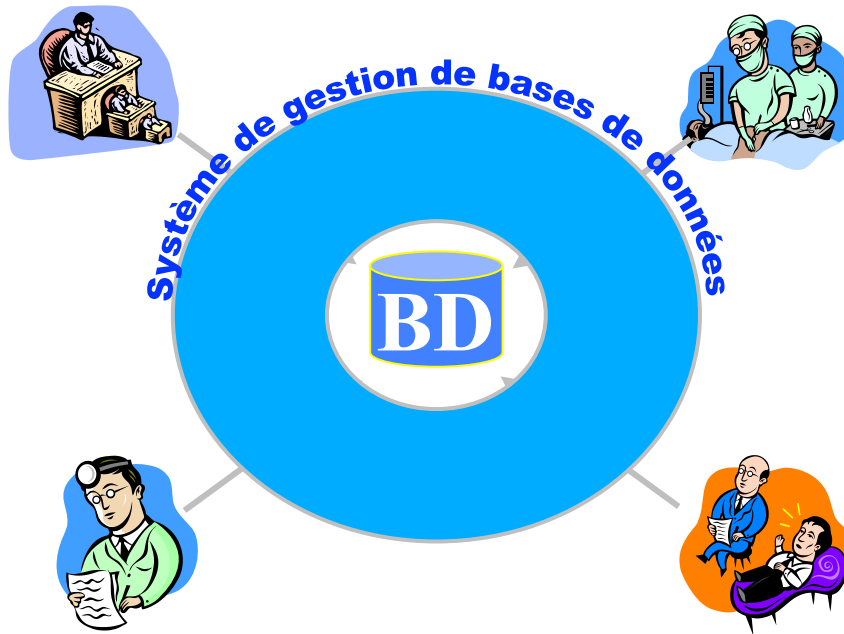
→ S'assurer que le virement a été fait !

VIII - Partage des données



- Accès concurrent aux mêmes données
- ➔ Conflits d'accès concurrent !!

Isolation et Cohérence



- Le SGBD gère les accès concurrents
 - ➔ Chacun à l' *impression* d' être seul (Isolation)
 - ➔ Cohérence conservée (Pas de *maj* conflictuelles)

IX – Confidentialité

- **Objectif : Protéger les données de la BD contre des accès non autorisés**
- Deux niveaux :
 - Connexion restreinte aux **usagers répertoriés** (mot de passe)
 - **Privilèges** d'accès aux objets de la base
- Usagers : Usager ou groupe d'usagers
- Objets : Relation, **Vue**, autres objets (procédures, etc.)

X - Standardisation

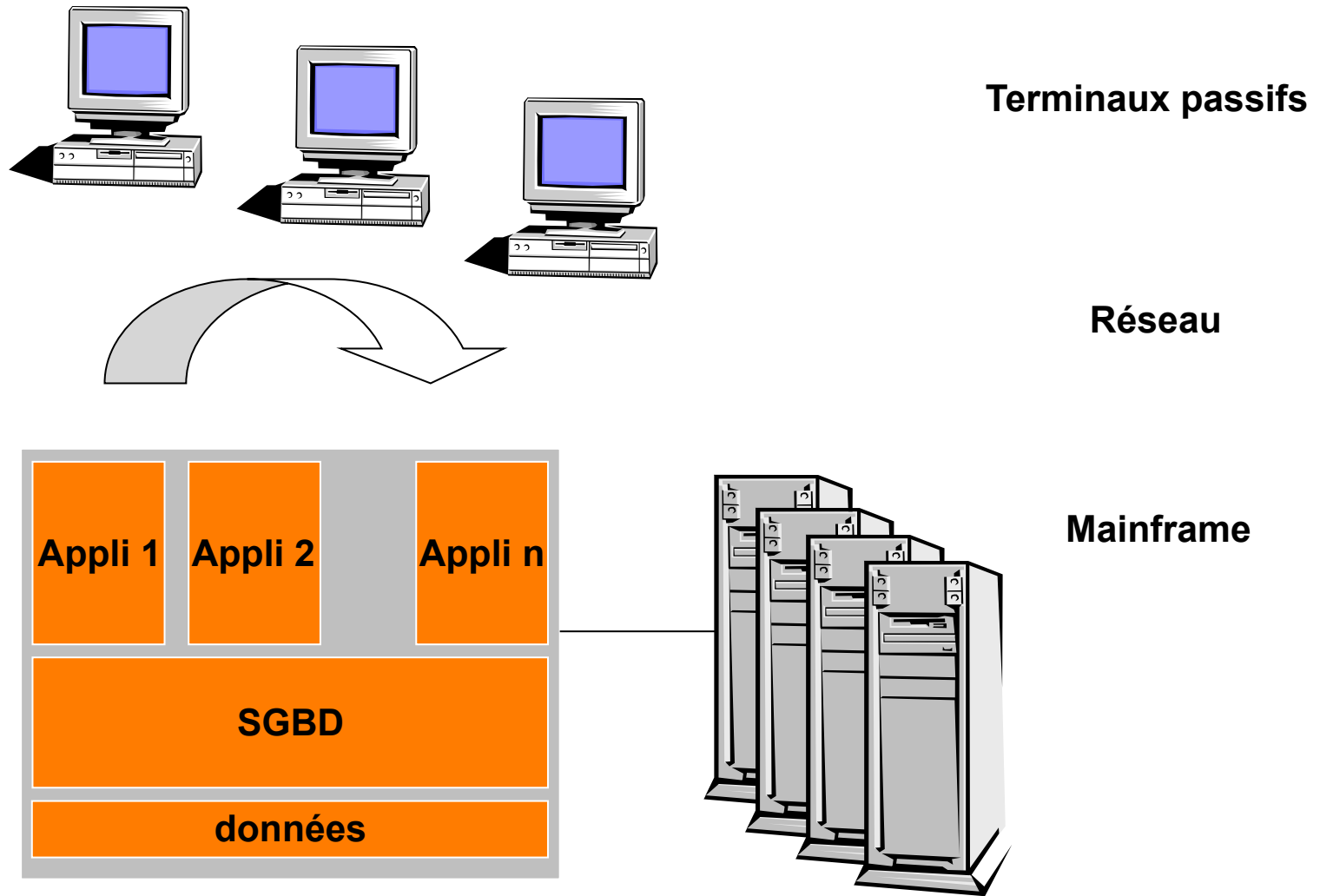
- L'approche bases de données est basée sur plusieurs standards
 - Langage SQL (SQL1, SQL2, SQL3)
 - Communication SQL CLI (ODBC / JDBC)
 - Transactions (X/Open DTP, OSI-TP)

- Force des standards
 - Portabilité
 - Interopérabilité
 - Applications multi sources...

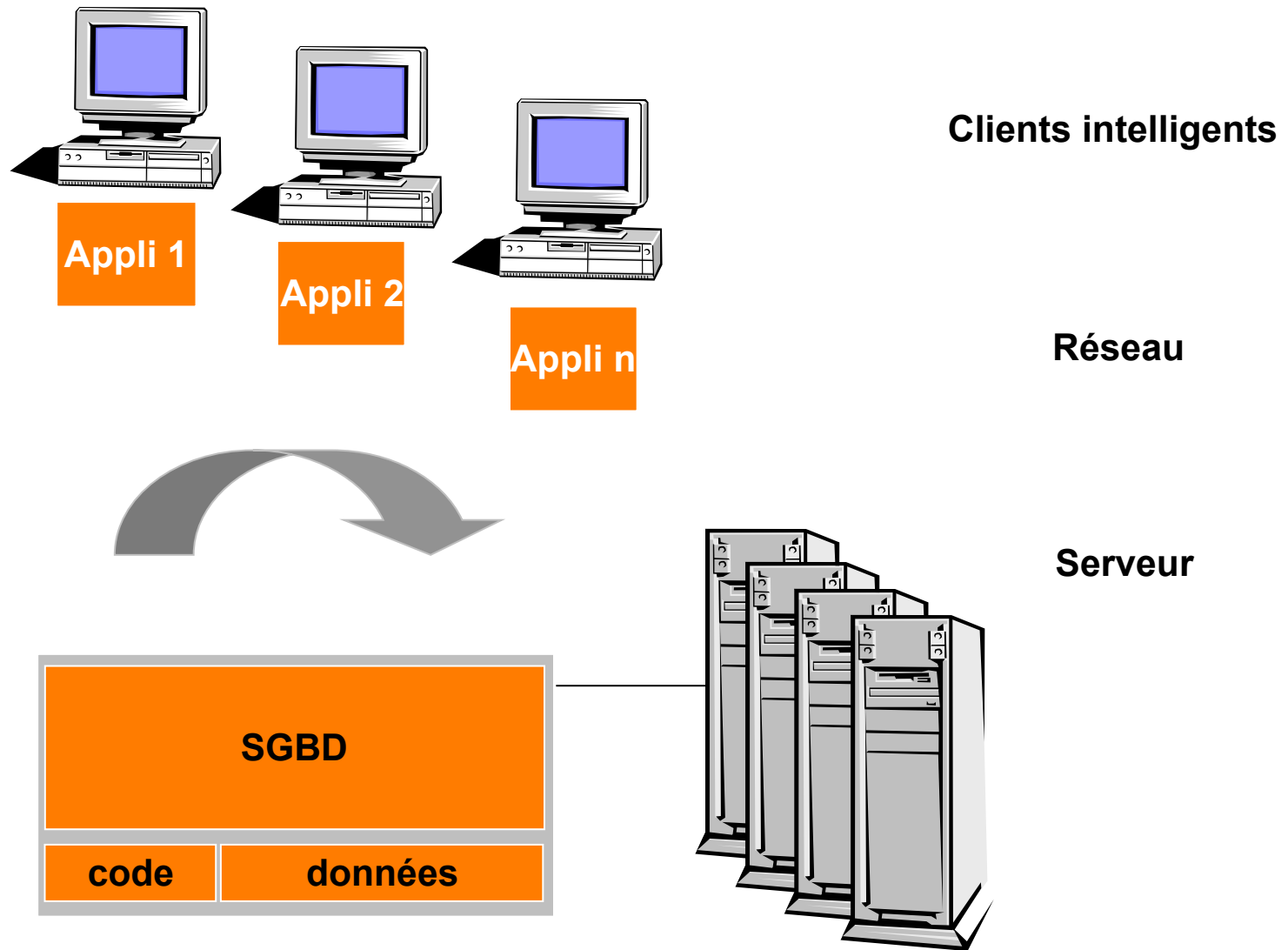
3. Architecture des SGBD

- ☞ **Les architectures physiques de SGBD sont très liées au mode de répartition.**
 - **BD centralisée**
 - **BD client/serveur**
 - **BD client/multi serveurs**
 - **BD répartie**
 - **BD hétérogène**
 - **BD mobile**
- ☞ **Le challenge se déplace des Péta-bases aux Pico-bases.**
 - **Péta-bases => parallélisme et grandes mémoires**
 - **Pico-bases => faible empreinte et forte sécurité**

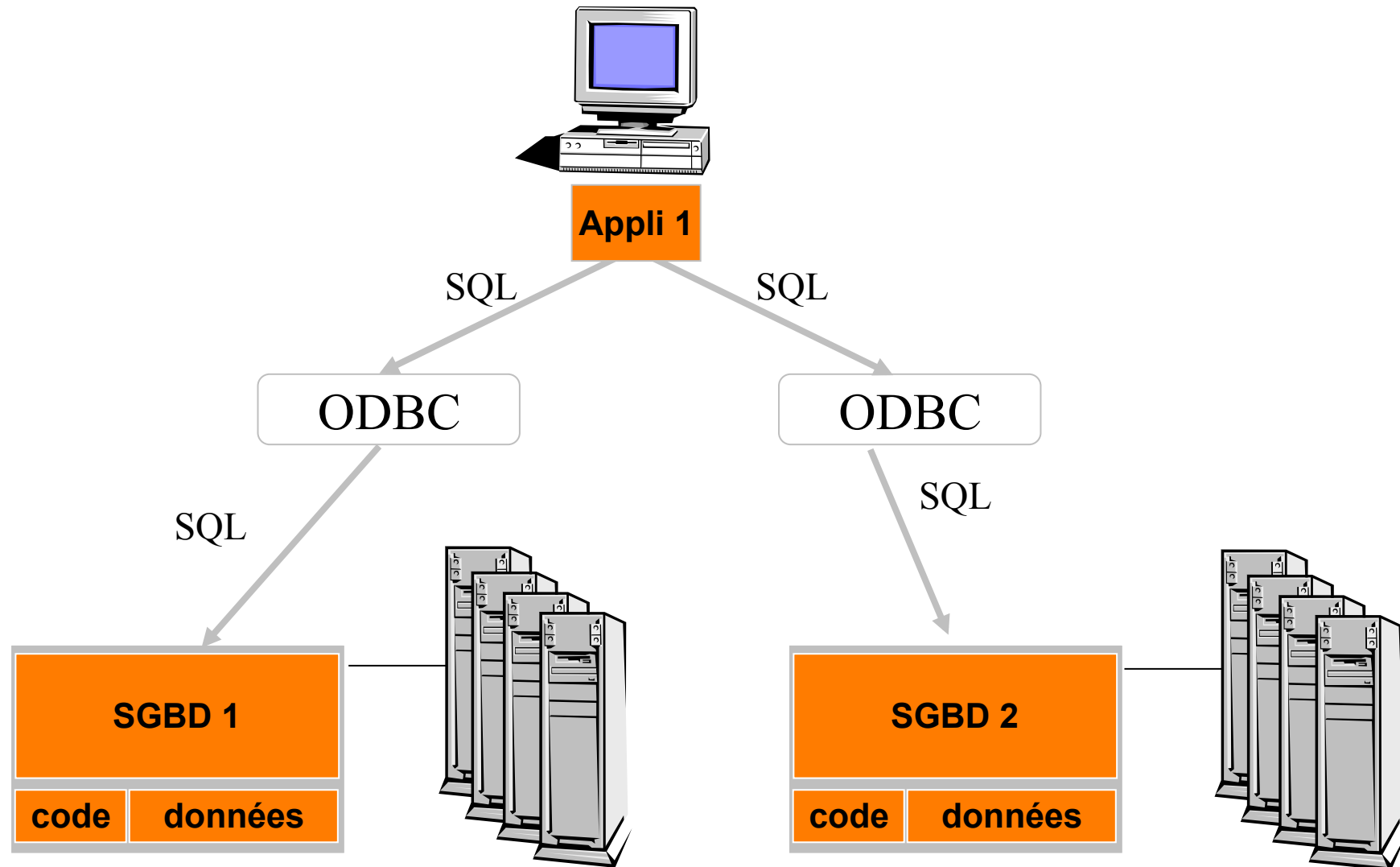
Architecture centralisée



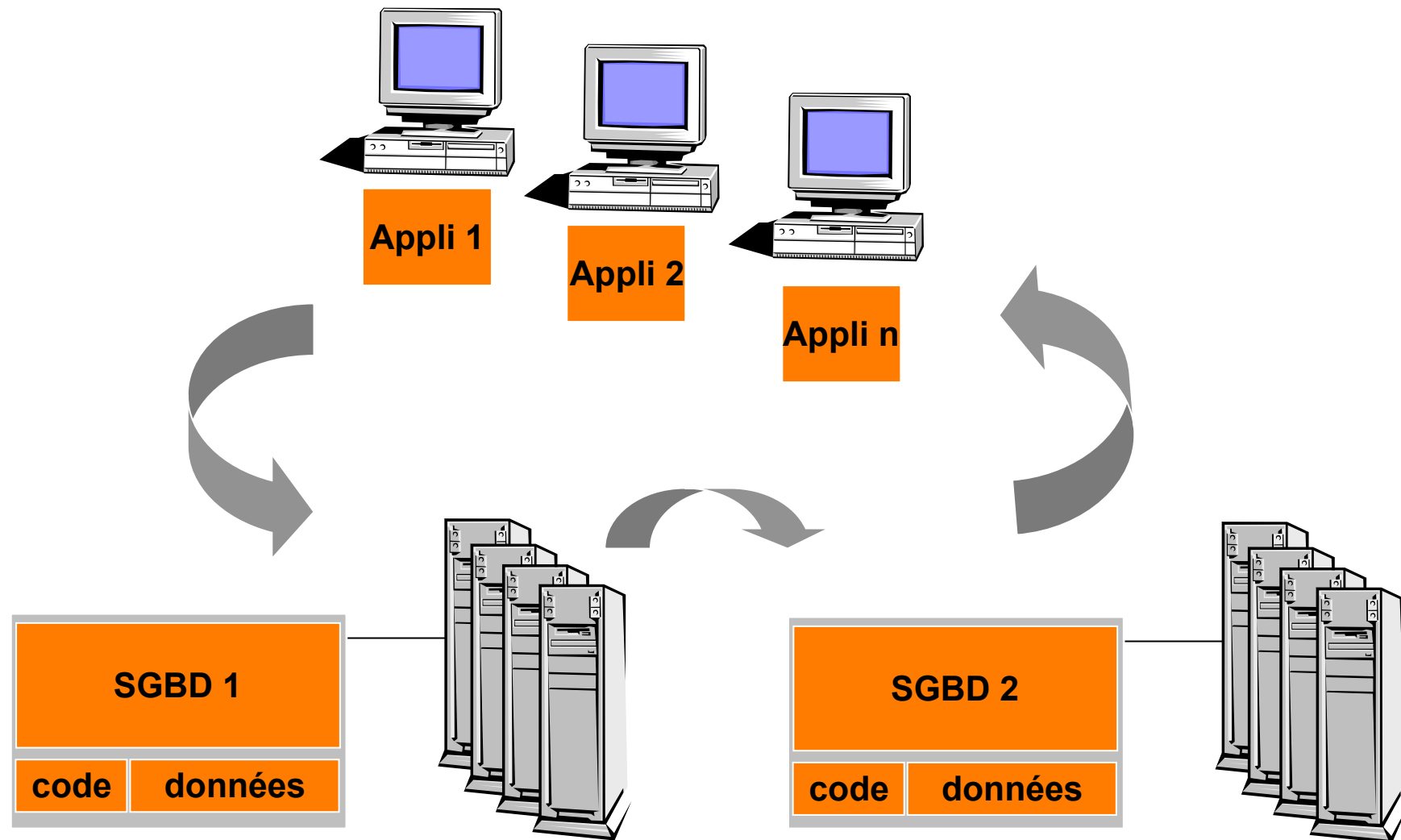
Architecture client-serveur



Architecture Client-Multiserveurs



Architecture répartie



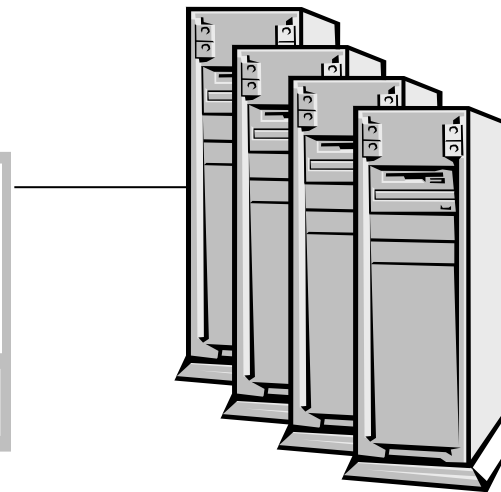
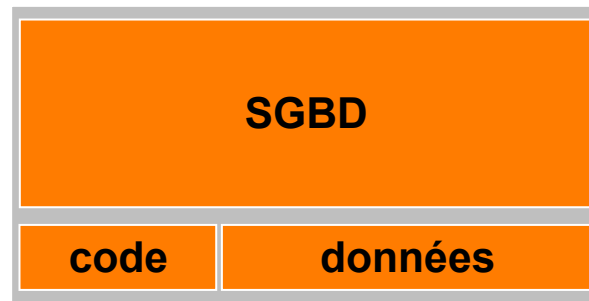
Architecture mobile



**Clients intelligents
mobiles**

**Données répliquées
et/ou personnelles**

Réseau sans fil















Serveur

4. Applications traditionnelles des SGBD

- **OLTP** (On Line Transaction Processing)
 - Cible des SGBD depuis leur existence
 - Banques, réservation en ligne ...
 - Très grand nombre de transactions en parallèle
 - Transactions simples
- **OLAP** (On Line Analytical Processing)
 - Données historisées et agrégées
 - Entrepôts de données, DataCube, Data Mining ...
 - Faible nombre de transactions
 - Transactions très complexes

Évolution des BD

	BD d'entreprise	BD personnelles	BD 'light' (PDA / Tél.)	PicoDBMS carte à puce
Capacité				
Prix				
Nombre				

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 complexe
 - 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 → Voitures; "Dupont" → "75AB75"

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

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.
- Représentée par un rectangle

Attribut

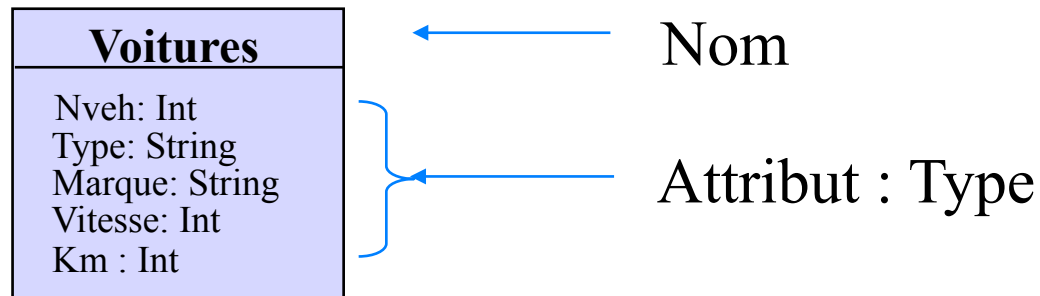
- Description des propriétés des entités
- Toutes les instances d'une entité ont les mêmes attributs
 - Attribut simple: caractérisé par un type de base
 - Attribut composé: caractérisé par un groupe d'attributs
 - Attribut multi-valué: caractérisé par plusieurs valeurs
- Avec le modèle E/R de base tout attribut est simple
- Avec le modèle E/R étendu, les attributs peuvent être complexes
 - Composés et multi-valués
- Représenté par un ovale

Domaines

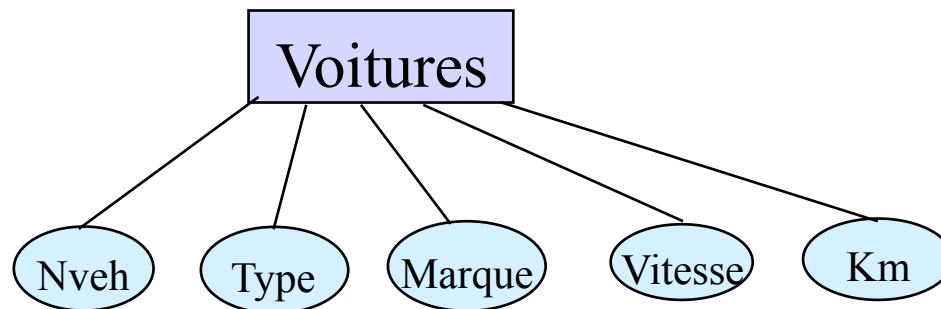
- Ensemble nommé de valeurs
 - Un attribut peut prendre une valeur dans un domaine
 - Généralisation des types élémentaires
- Exemples
 - Liste de valeurs (1,2,3)
 - Type contraint ($0 < \text{int} < 100$)
 - Type simple (int, varchar, etc.)
- Permettent de préciser les valeurs possibles des attributs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

Représentation

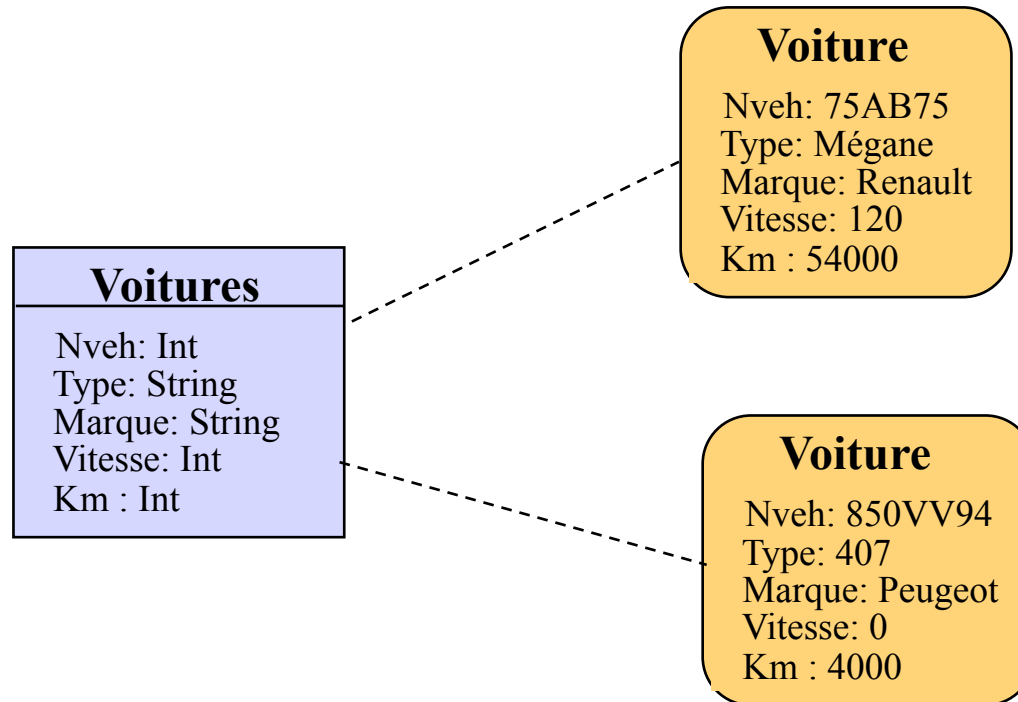
- Rectangle avec attributs (UML)



- Rectangle avec attributs accrochés (E/R)



Exemple d'instance



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

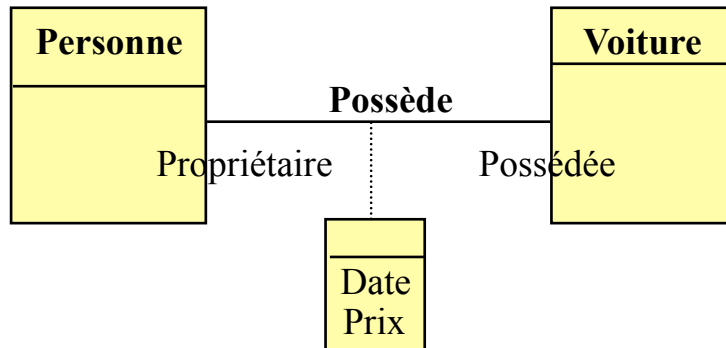
- Les entités sont reliées ensemble par des associations
 - Entre instances: par exemple 1 véhicule est associé à 1 personne
 - Entre classes: abstraction des associations entre instances
- Une association peut avoir des attributs (propriétés)
- Elle peut relier plusieurs entités ensemble
- Il est possible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

Association: quelques définitions

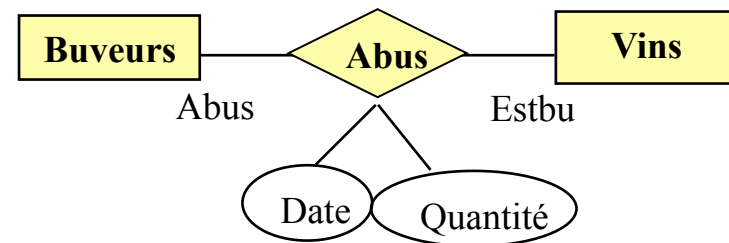
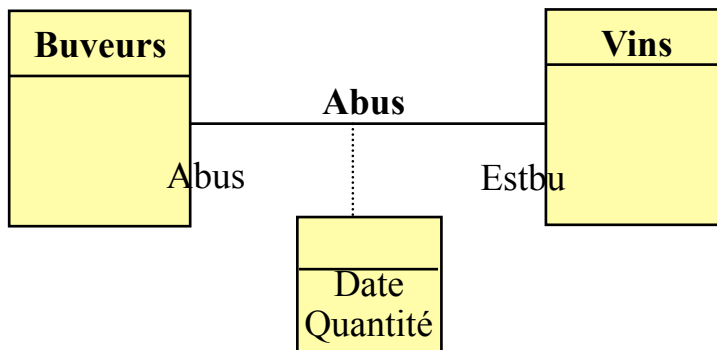
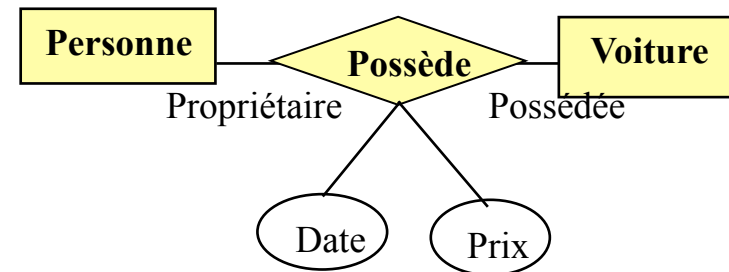
- Association (Association)
 - Une relation entre des instances de deux (ou plus) classes
- Lien (Link)
 - Une instance d'association
- Rôle (Role)
 - Une extrémité d'une association
- Attribut de lien (Link attribute)
 - Un attribut de l'association instancié pour chaque lien
- Cardinalité (Multiplicity)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

Exemples en UML et E/R

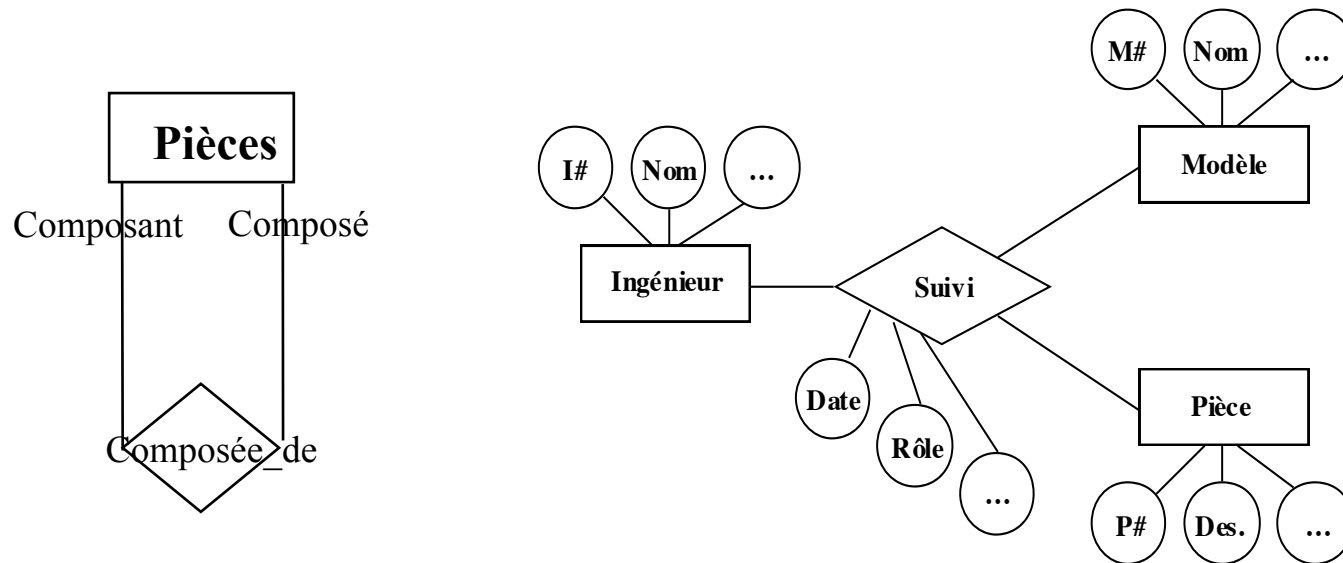
UML



E/R

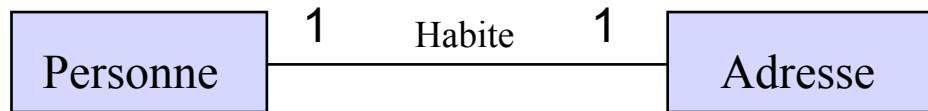


Degré d'une association

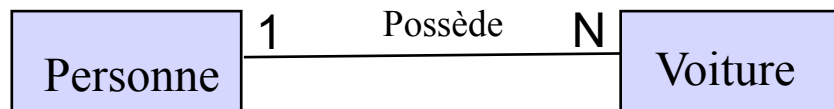


- La plupart des associations sont de degré 2 (binaires)

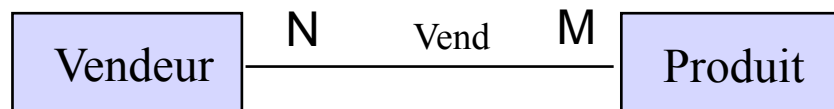
Cardinalité d'une association



- 1:1 -> Une personne habite à une adresse et à une adresse habite 1 personne



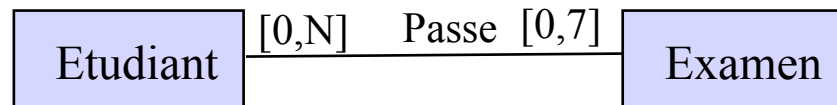
- 1:N -> Une personne possède N voitures et une voiture est possédée par 1 personne



- N:M -> un vendeur peut vendre M produits et un produit peut être vendu par N vendeurs

Cardinalités min et max

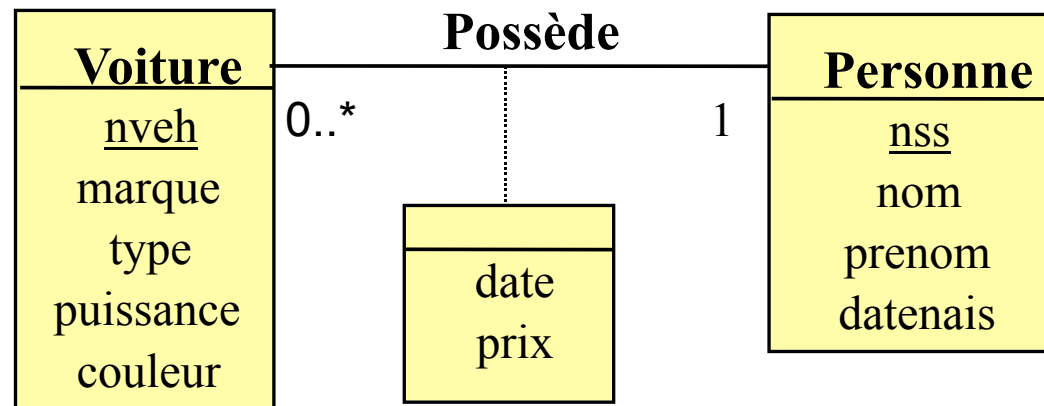
- Cardinalité maximum
 - Indique le nombre maximum d'instances d'une classe d'entité participant à une association
- Cardinalité minimum
 - Indique le nombre minimum d'instances d'une classe d'entité participant à une association



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>	ordonné (0 à N)
<u>{ord}</u>	
<u>3..5</u>	limité (de 3 à 5)

Exemple



La pratique de la conception

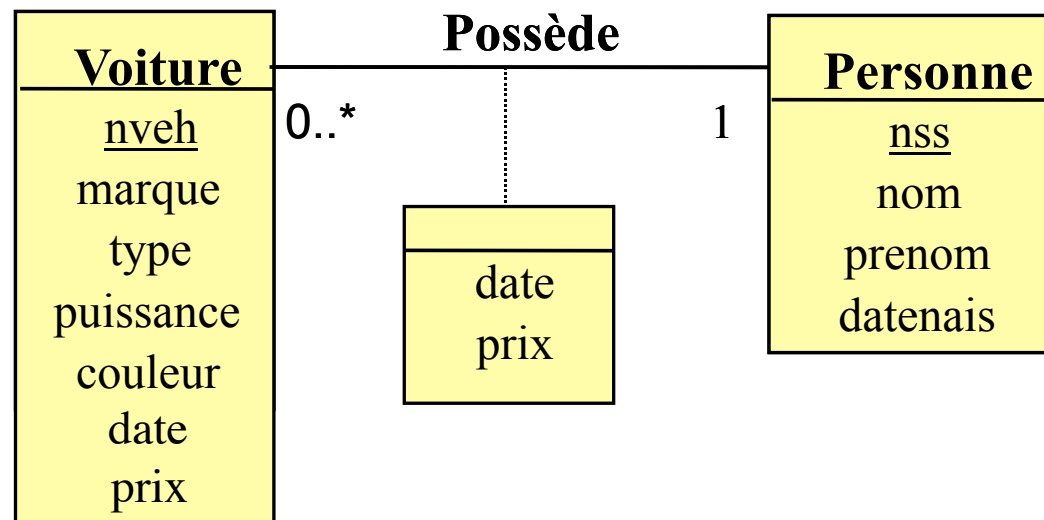
- Bien comprendre le problème à résoudre
- Essayer de conserver le modèle simple
- Bien choisir les noms
- Ne pas cacher les associations sous forme d'attributs
 - utiliser les associations
- Faire revoir le modèle par d'autres
 - définir en commun les objets de l'entreprise
- Documenter les significations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

3. Passage au relationnel

- Implémentations des entités et associations sous forme de tables
 - Règle 1 : Une **entité** est représentée par une relation (table) de même nom ayant pour attributs la liste des attributs de l'entité.
 - Règle 2 : Une **association** est représentée par une relation de même nom ayant pour attributs la liste des clés des entités participants et les attributs propres de l'association.
 - Exemples :
 - POSSEDE (NSs, NVeh, Date , Prix)
 - ABUS (NVeh, Nb, Date, Quantité)

Cont.

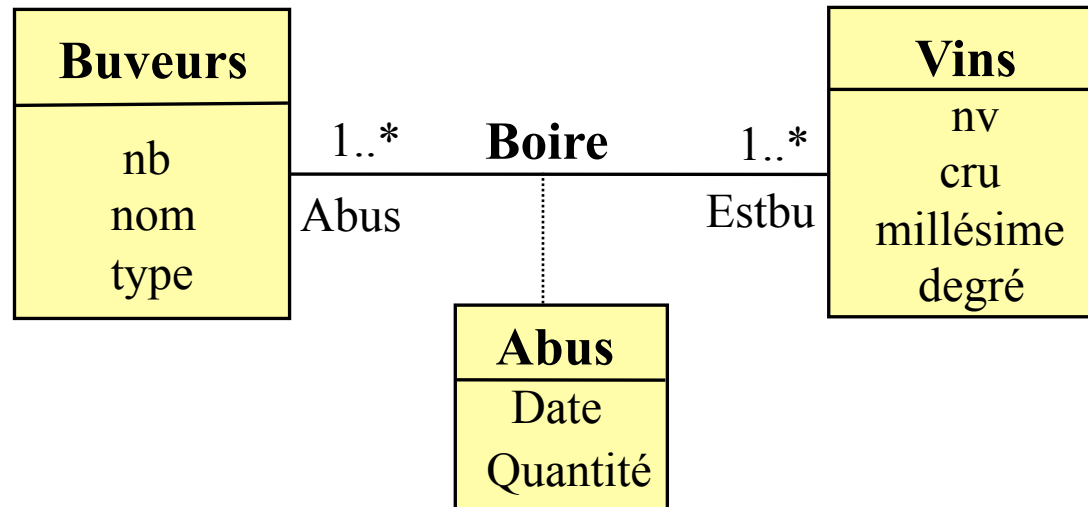
- Amélioration possible si un tuple d'une table référence un ou 0..1 tuple de l'association.
- Règle 3 : Regrouper les associations 1:1 ou 1:0..1 avec l'entité d'origine
 - Exemple :
 - VOITURE et POSSEDE : regroupées en une seule table si toute voiture est possédée par un ou aucun propriétaire



Contraintes référentielles

- ...Règle 6 : Toute association $E1 \rightarrow R \rightarrow E2$ représentée par une table R non intégrée à $E1$ ou $E2$ donne naissance à 2 contraintes référentielles (une pour chaque clé de $E1$ et $E2$).

Exemple



Buveurs ([nb](#), nom, type)

Vins ([nv](#), cru, millesime, degre)

Abus([nb](#) REF buveurs, [nv](#) REF vins, _date, quantite)

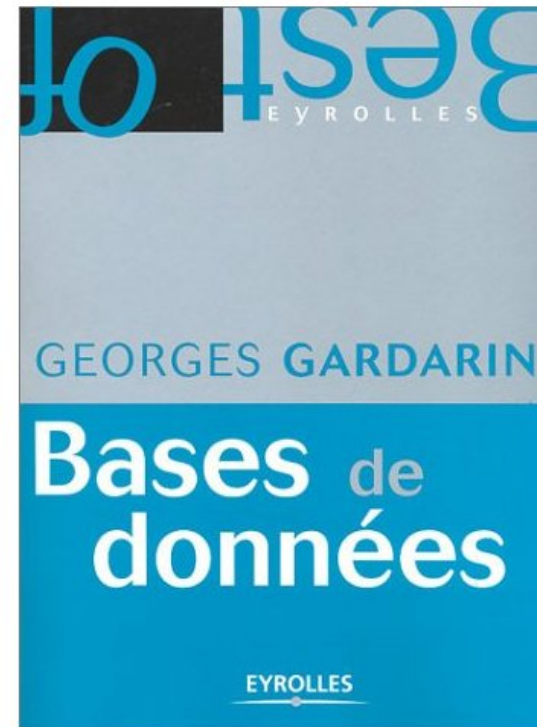
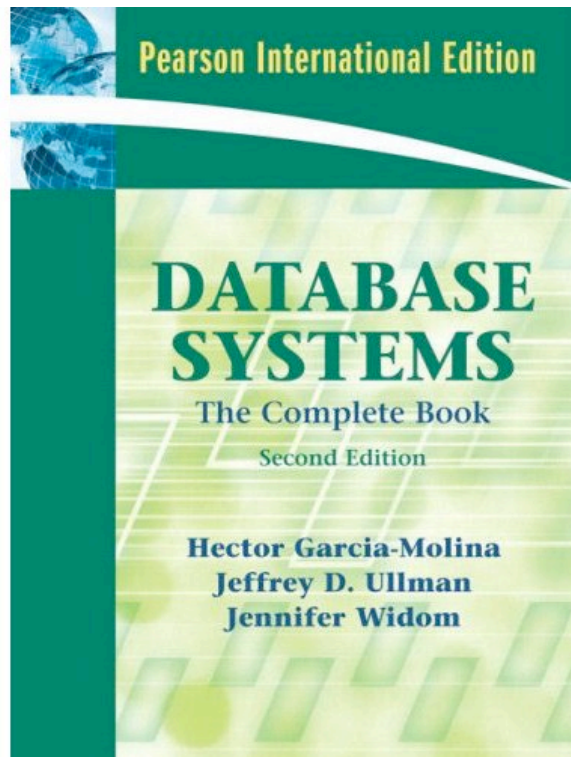
Règle 6 : À cause de l'association (obligatoire).

6. Conclusion

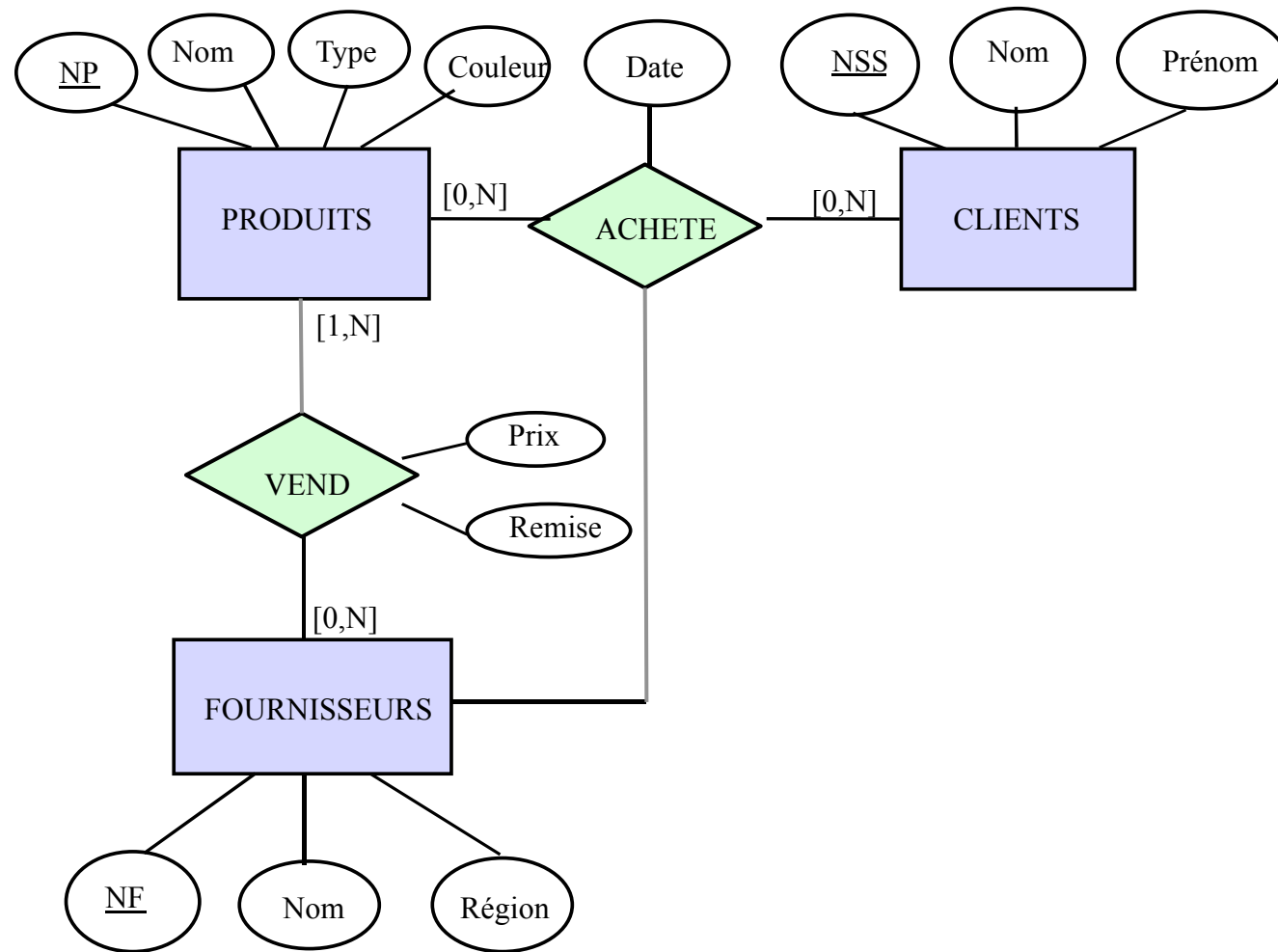
- Intérêt de l'utilisation d'une méthode de conception
 - proche du monde réel
 - démarche sémantique claire
 - diagramme standards
- Passage au relationnel semi-automatique
 - outils du commerce utilisables (Objecteering, Rose, etc.)
 - supporteront les extensions objet-relationnel à venir
- Extensions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

Bibliographie

- Hector Garcia Molina, Jeffrey D. Ulman and Jennifer Widom. Database Systems. Second Edition, Pearson Prentice Hall, International Edition. 2009.
- Bases de données. George Gardarin, 5^e tirage, 2003.



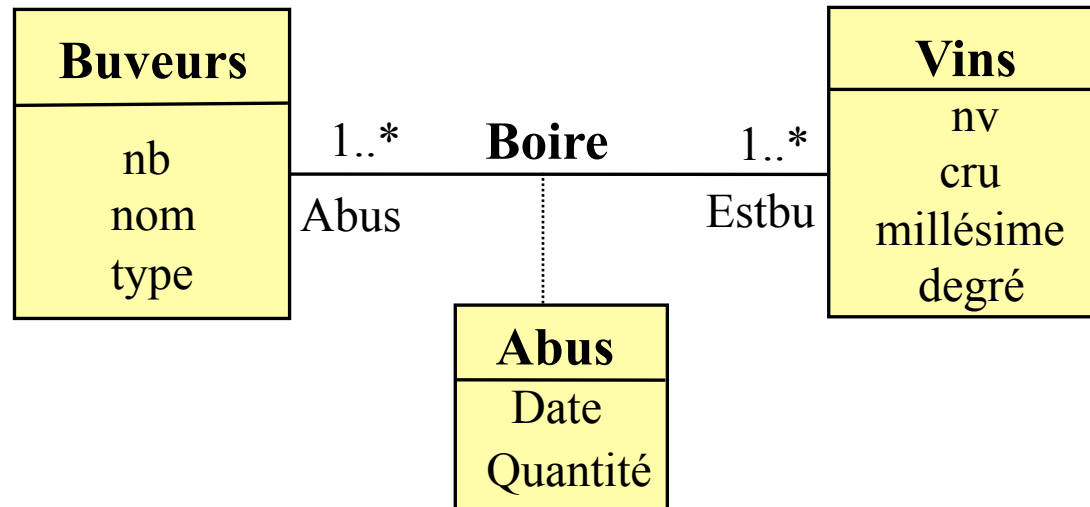
Exemple de modèle E/R



Contraintes référentielles

- ...Règle 6 : Toute association $E1 \rightarrow R \rightarrow E2$ représentée par une table R non intégrée à $E1$ ou $E2$ donne naissance à 2 contraintes référentielles (une pour chaque clé de $E1$ et $E2$).
- Règle 7 : Toute association $E1 \rightarrow R \rightarrow E2$ de cardinalité minimale 1 sur $E2$ représentée par une table non intégrée à $E1$ donne naissance à une contrainte référentielle additionnelle (la clé de $E1$ référence son instance dans R).

Exemple



E: A cause de la cardinalité min de 1

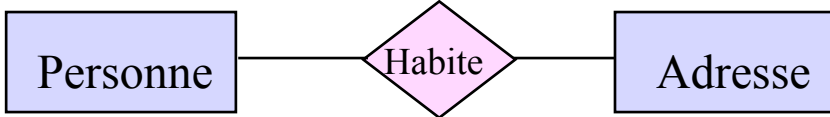
Buveurs ([nb](#) REF Abus.nb, nom, type)


Vins ([nv](#), cru, millesime, degre)


Abus([nb](#) REF buveurs, [nv](#) REF vins, _date, quantite)

D: À cause de l'association (obligatoire).

Cardinalité d'une association

- 1:1 

Personne — Habite — Adresse
- 1:N 

Personne — Possède — Voiture
- N:M 

Vendeur — Vend — Produit