

# **Bases de Données**



**Stella MARC-ZWECKER**

**Maître de conférences**

**Dpt. Informatique - Uds**

**[stella@unistra.u-strasbg.fr](mailto:stella@unistra.u-strasbg.fr)**

# Plan du cours

- 1. Introduction aux BD et aux SGBD
  - Objectifs, fonctionnalités et évolutions
  
- 2. Modèle relationnel
  - Définitions et concepts
  - Langage d'interrogation
    - Algèbre relationnelle
  
- 3. Langage SQL (Structured Query Lang.)
  - =>TP

# Plan du cours

- 4. Normalisation d'un schéma relationnel
  - | Dépendances fonctionnelles
  - | Formes normales
  
- 5. Modèle entité/association
  - | Conception du schéma logique
  - | Production du schéma de la BD
    - Passage du modèle E/A aux tables (SQL)

# Organisation

- Séances :

- | 12h cours

- | 14h TD

- | 10h TP

- Evaluation continue :

- | 1 interrogation écrite : 20% (~ octobre)

- | 1 TP noté : 30% (~ novembre)

- | 1 épreuve convoquée : 50% (~ décembre)

# Bibliographie

- Gardarin, G. *Bases de données*. Editions Eyrolles, 2003.
- Hainaut, J. L., *Bases de données et modèles de calcul (4e éd.) : Outils et méthodes pour l'utilisateur*. Dunod, 2005.
- Hainaut, J. L., *Bases de données (3e éd.): Concepts, utilisation et développement*. Dunod, 2015.

# **Chapitre 1**



## **Introduction aux Bases de Données (BD) et aux Systèmes de Gestion de Bases de Données (SGBD)**

# 1. Base de données

- | Bases de données utilisées dans :
  - Le domaine de l'informatique de gestion
  - Le monde du web et des applications client/serveur dynamiques
  - Entrepôts de données, « big data »
  
- | Discipline s'appuyant sur une théorie fondamentale
  
- | **Qu'est-ce qu'une base de données ?**
  - C'est un ensemble de données structuré et organisé permettant le stockage d'informations.
  
  - Les données doivent être interrogeables par le contenu et selon n'importe quel critère

# Exemple de schéma de base de données [Gardarin]

## BUVEUR

---

Nom	Prénom	Adresse
Dupont	Jean	Lyon
Schmidt	Titou	Colmar
Lopez	Carmen	Madrid
Costes	Maria	Toulouse
Chassin	Malik	Bordeaux

---

## VIN

---

Cru	Millésime	Quantité
Saint Emilion	2001	1243
Riesling	2003	734
Pinot Gris	1997	2874
Cahors	1985	112
Bourgogne	1966	56

---

- **Au niveau logique :**

les données sont vues par l'utilisateur comme des tables à 2 dimensions (abstraction)

- **Au niveau physique :**

le SGBD représente les données comme une suite d'octets pour que le système d'exploitation puisse écrire en mémoire ou sur le disque



## CONSOMMATION

---

Buveur	Vin	Date	Quantité
Costes	Saint Emilion	25/12/05	3
Chassin	Pinot Gris	10/07/05	2
Lopez	Saint Emilion	04/08/05	5
Lopez	Riesling	31/12/05	2
Schmidt	Riesling	05/11/05	6
Schmidt	Riesling	05/12/05	6
Schmidt	Riesling	05/01/06	6
Dupont	Cahors	15/01/06	7
Dupont	Bourgogne	30/01/06	8

---

- **BD « orientée lignes » :**

Le SGBD sérialise les valeurs d'une ligne ensemble (stockage ligne par ligne)

- **BD « orientée colonnes » :**

Le SGBD sérialise les valeurs d'une colonne ensemble (stockage colonne par colonne)

## 2. SGBD

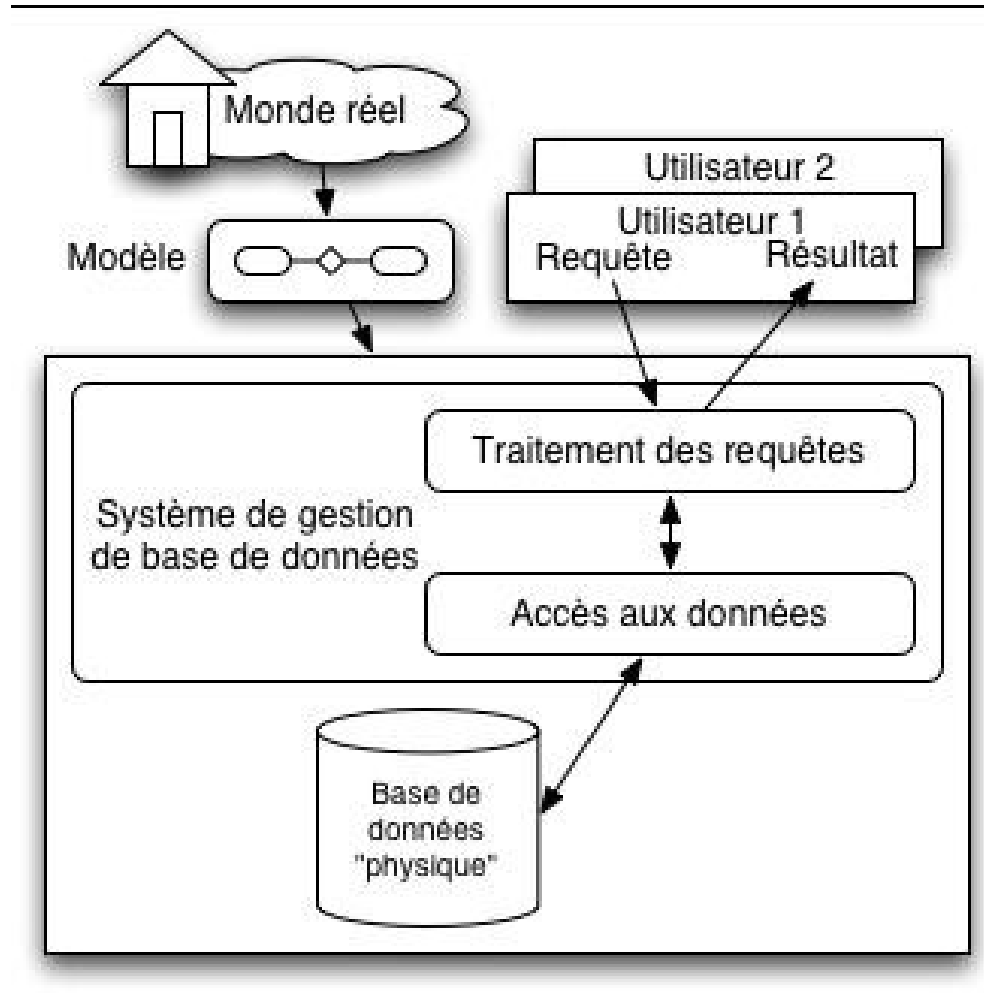
### Qu'est-ce qu'un SGBD ?

- Ensemble des logiciels permettant l'exploitation de la BD (mise à jour, recherche...)
- Le SGBD rend transparent le partage des données

### Objectifs d'un SGBD :

- Langage de manipulation des données
- Indépendance données/SGBD
- Fournir un accès efficace aux données
- Contrôler la redondance des données
- Cohérence des données
- Partage des données
- Sécurité des données

# Illustration du rôle d'un SGBD



## 2.1. Enjeux d'un SGBD

Développement de concepts, méthodes et algorithmes spécifiques permettant de :

- Gérer le stockage des données et l'accès aux données en mémoire secondaire (i.e. sur disque) :
  - Volumes de données de plus en plus importants (plusieurs dizaines de téra-octets)
  - Pérennité des données
  
- Gestion Multi-utilisateurs
  - Internet : plusieurs dizaines de milliers d'utilisateurs dans un contexte d'exploitation changeant
  - Gestion multi-agents
  
- Fouille de données, apprentissage, aide à la décision (OLAP : OnLine Analytical Processing)

# Exemples de SGBD

- **MySQL** : <http://www.mysql.org/> (domaine public )
- **PostgreSQL**: <http://www.postgresql.org/> (domaine public)
- **Oracle** : <http://www.oracle.com/> (Oracle Corporation)
- **IBM DB2** : <http://www-306.ibm.com/software/data/db2/>
- **Microsoft SQL** : <http://www.microsoft.com/sql/>
- **Sybase** : <http://www.sybase.com/linux>
- **Informix** : <http://www-306.ibm.com/software/data/informix/>

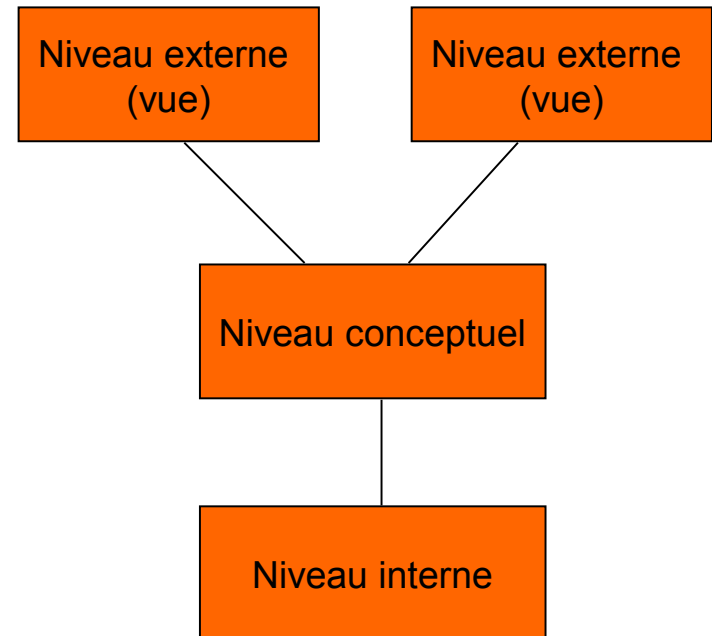
## 2.2. Modèles de données

- | Un **modèle de données** est un mode de représentation des informations caractérisé par :
  - **1.** Les structures des données
  - **2.** Les contraintes qui permettent de spécifier les règles que doit respecter une base de données.
  - **3.** Les opérations permettant de manipuler les données (interroger ou mettre à jour la base).
  
- | DDL : langage de description des données (1. et 2.)  
(Data Description Language)
  - langage utilisé pour décrire le schéma d'une base de données
  - Ex : SQL DDL (création, modification, destruction de tables)
  
- | LMD : langage de manipulation des données (3.)  
(Data Manipulation Language)
  - Ex : SQL DML (consultation, modification, suppression de données)

## 2.3. Architecture fonctionnelle

Trois niveaux de description du SGBD (ANSI/SPARC) :

- Le niveau externe
  - Niveau relatif aux utilisateurs
  - Manière dont ils voient les données
- Le niveau conceptuel
  - Niveau intermédiaire
  - Représentation « abstraite » de l'ensemble de la BD
- Le niveau interne
  - Niveau relatif à la mémoire physique
  - Manière dont les données sont réellement enregistrées



## 2.3.1. Niveau externe (vue)

- Pour un utilisateur la vue externe (sous-base virtuelle) correspond à la partie de la base de données à laquelle il a le droit d'accéder
- Protection des données (accès limité aux données du schéma externe pour un groupe donné)

### Buveurs de vins

-Nom  
-Prénom  
-NbCons  
-Cru  
-Millésime

### Identité Buveur

-Nom  
-Prénom  
-Adresse

### Vins consommés

-Cru  
-Millésime  
-Quantité



## 2.3.2. Niveau conceptuel (niveau logique)

- Ne se soucie pas de l'implémentation physique des données ni de la façon dont chaque groupe d'utilisateurs va se servir de la base de données
- Se concentre sur la description des entités, du type des données, des relations existant entre les entités et des opérations des utilisateurs.
  - Types de données élémentaires décrivant les propriétés des objets (ex: cru d'un vin, millésime)
  - Types de données composés décrivant les objets ou les relations entre objets (ex: vin, buveur)
  - Types de données composés décrivant les associations (ex: consommation de vin par un buveur)
  - Règles que devront suivre les données (ex: âge d'un buveur compris entre 14 et 120)

# Exemple de schéma conceptuel

- Types d'objets :

- BUVEUR (Nom, Prénom, Adresse)

- VIN (Cru, Millésime, Quantité)

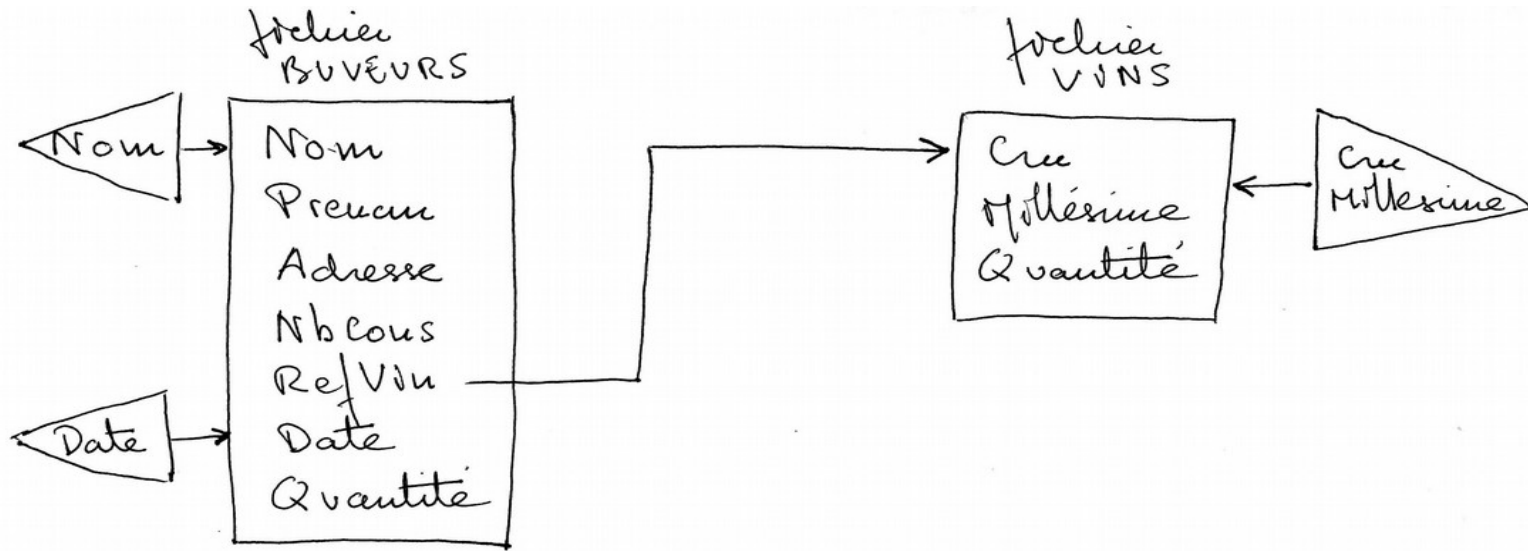
- Types d'associations :

- CONSOMMATION ( BUVEUR, VIN, Date, Quantité)

## 2.3.3. Niveau interne (niveau physique)

- Décrit la structure de stockage physique de la base de données. Il s'appuie sur un système de gestion de fichiers pour définir la politique de stockage des données.  
⇒ dépend du SGBD
- Est responsable du choix de l'organisation physique des fichiers ainsi que de l'utilisation de méthodes d'accès en fonction de la requête.
- Les données sont décrites en termes de représentation physique en machine, i.e de structures de mémorisation, de méthodes de stockage et d'accès utilisées pour ranger et retrouver les données sur disque. Par ex :
  - Fichiers contenant les données (nom, localisation...)
  - Articles de ces fichiers (longueur, champs...)
  - Chemins d'accès à ces articles (index, chaînages...)

## Exemple de schéma interne [Gardarin]



- fichier **BUEURS** décrivant les buveurs et leurs consommations : chaque article contient le nom, prénom, adresse du buveur, suivi d'un groupe répétitif comprenant le nombre de consommations, et pour chaque consommation, un pointeur sur le vin bu, la date et la quantité
- un index sur le nom de buveur et sur la date permet d'accéder directement aux articles de ce fichier à partir de la valeur de ces attributs.
- fichier **VINS** : chaque article contient le cru, le millésime et la quantité de vin en stock. Ce fichier est indexé sur le couple (Cru, Millésime).

# 3. Objectifs et fonctionnalités d'un SGBD

## 3.1. Manipulation des données par un langage non procédural

- Manipulation 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 premier ordre ++
- Syntaxe
  - **SELECT** <structure de résultats>
  - **FROM** <relations>
  - **WHERE** <conditions>

## 3.2. Intégrité logique

- Objectif : détecter les mises à jour erronées
  - => contrôle sur les données :
    - contrôle de types
    - contrôle de valeurs
  - => contrôle sur les relations entre les données :
    - relations entre données élémentaires
    - relations entre objets
- Contraintes d'intégrité
  - => langage de définition des contraintes d'intégrité
  - => vérification automatique de ces contraintes

## 3.3. Intégrité physique

- Tolérance aux fautes
  - Transaction failure : contraintes d'intégrité non respectées
  - System failure : panne de courant, crash serveur
  - Media failure : perte du disque
  - Communication failure : panne du réseau
- Objectifs (ACID)
  - Atomicité des transactions
  - Correction des transactions
  - Isolation des transactions
  - Durabilité des transactions
- Moyens
  - Journalisation : mémorisation des états successifs
  - Mécanismes de reprise

# Intégrité physique

## Exemple de transaction

Begin

$CEpargne = CEpargne - 1000$

$CCourant = CCourant + 1000$

Commit T1

- **Atomicité**

Begin

$CEpargne = CEpargne - 1000$

>>>> panne

$CCourant = CCourant + 1000$

Commit T1

=> annuler le débit !

- **Durabilité**

Begin

$CEpargne = CEpargne - 1000$

$CCourant = CCourant + 1000$

Commit T1

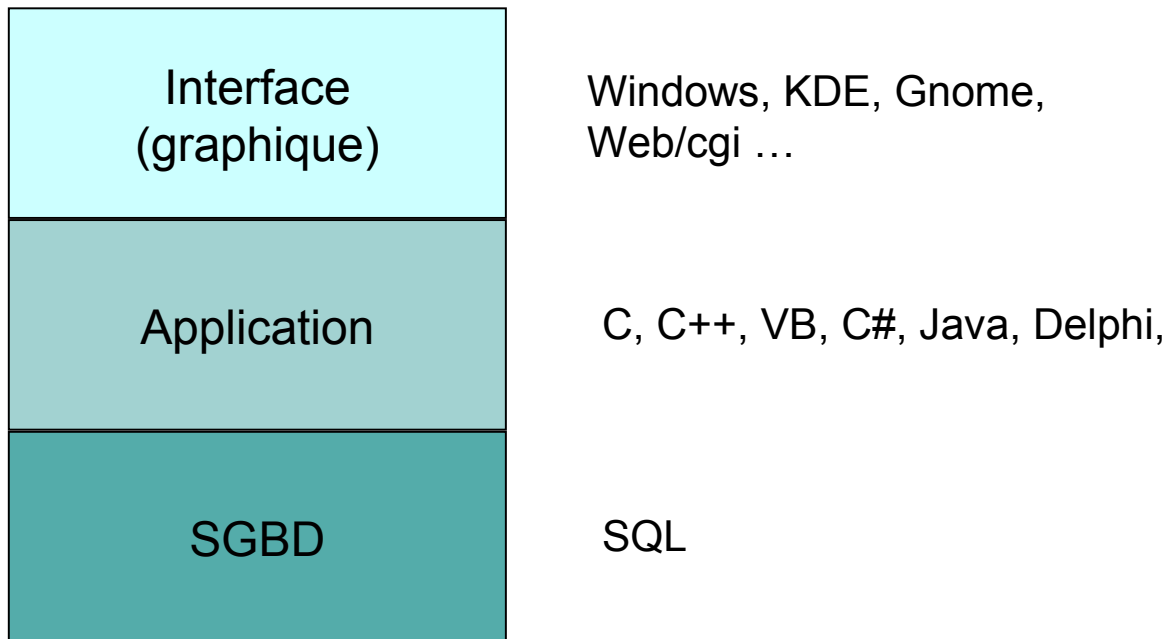
>>>>> crash disque

=> s'assurer que le virement n'a pas été écrasé !



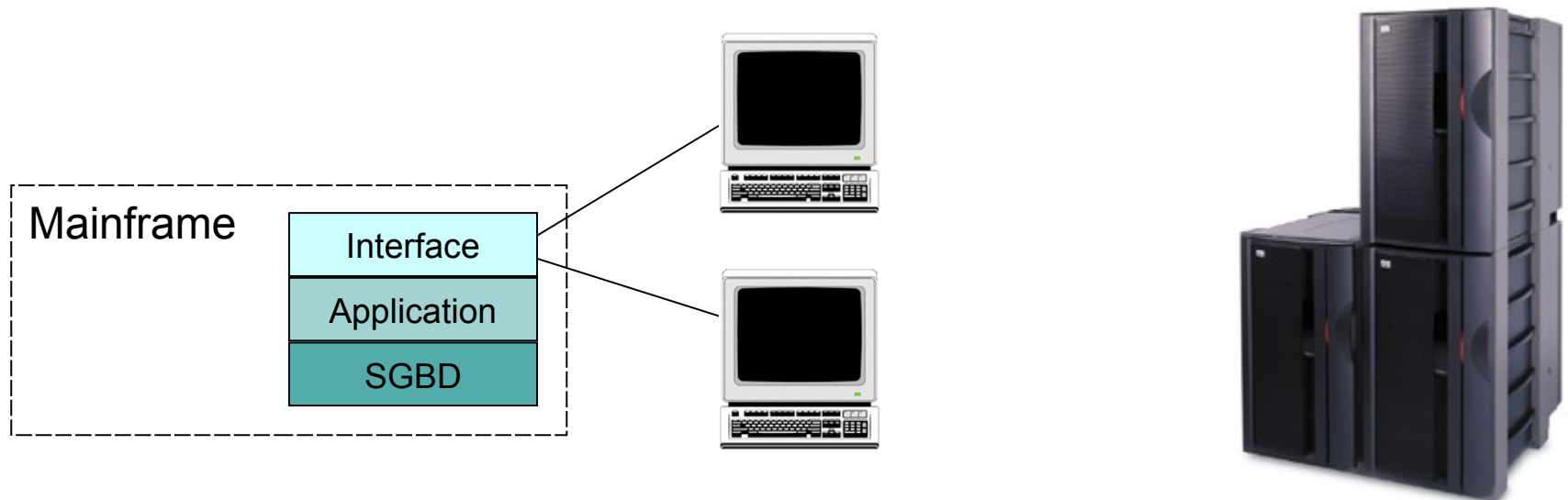
# 4. Evolution des architectures des SGBD

On peut diviser conceptuellement le système en 3 couches logicielles :



## 4.1. Architecture Mainframe (1960)

- Les 3 couches sont implémentées sur la même machine
- Systèmes propriétaires, non standardisés
- Terminaux passifs



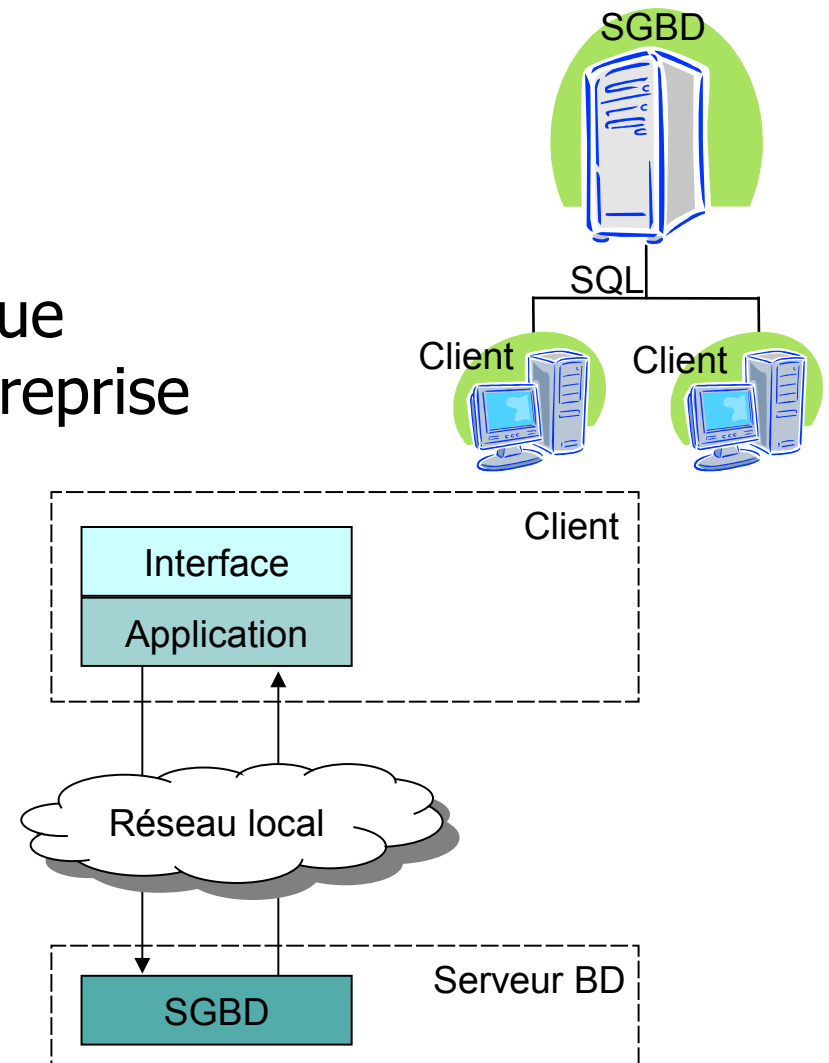
## 4.2. Architecture Client/Serveur (années 70)

### ■ Apparue avec :

- | Serveurs puissants
- | Ordinateurs de bureautique répanus au sein de l'entreprise
- | Réseaux locaux rapides

### ■ Avantage

- | Technologie standard
- | Milieu hétérogène



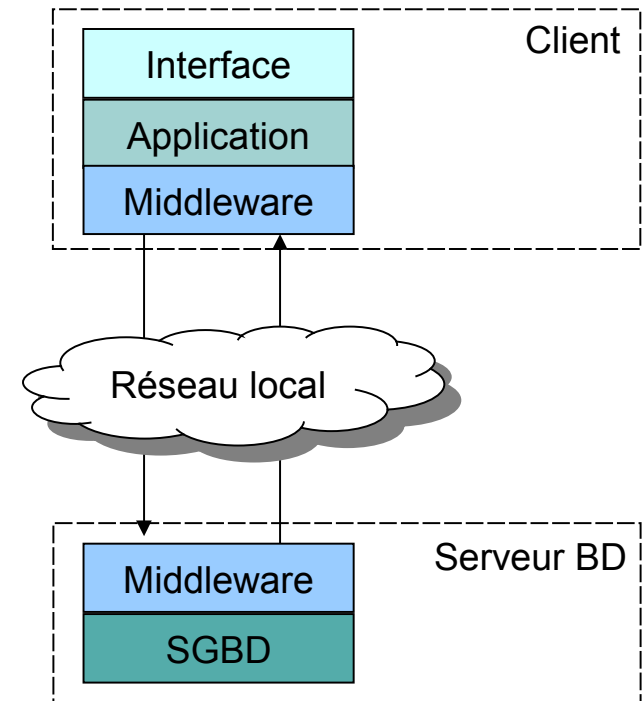
# Architecture Client/Serveur « 2-tiers » (2 strates)

## Principes [A. Montfort]

- 2 couches : **client et serveur**
- Fonction de présentation à la charge du client exclusivement
- Calcul réparti entre client et serveur
- Logiciel client spécifique au serveur
- Données accessibles via le serveur

Ex. de Middleware :

- SQL\*Net (Oracle)
- ODBC « Open DataBase Connectivity » (Microsoft)
- HTTP : middleware non transactionnel



# Architecture Client/Serveur « 2-tiers » (2 strates)

## Avantages

- Architecture simple à développer
- Données centralisées
- Interface utilisateur riche

## Inconvénients

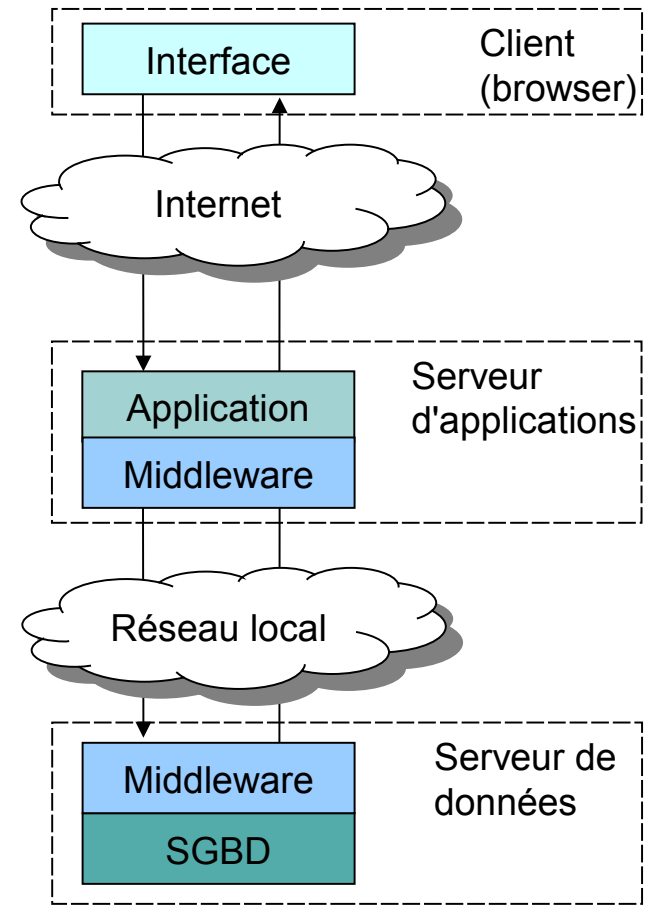
- Client lourd
- Logiciel spécifique au serveur  
=> contrôle des évolutions de versions et d'applications
- Aspect propriétaire de l'application client (entreprises volatiles)  
=> problème de viabilité à long terme

# Architecture client/serveur « 3-tiers »

## Principes :

3 couches :

- couche **présentation** des données associée au client (client « léger »)
- couche **traitement** des données assurée par un serveur d'applications web
- couche **données** liée au serveur de base de données (SGBD)



# Architecture client/serveur « 3-tiers »

## ■ Avantages

- Séparation Client / Serveur / SGBD  
=> spécialisation des développeurs sur chaque tiers de l'architecture
- Client léger (navigateur)  
=> requêtes client plus simples
- Facilité de maintenance
- Portabilité du tiers serveur => meilleure évolutivité des applications

## ■ Inconvénients

- Architecture plus complexe
- Coûts plus élevés au départ

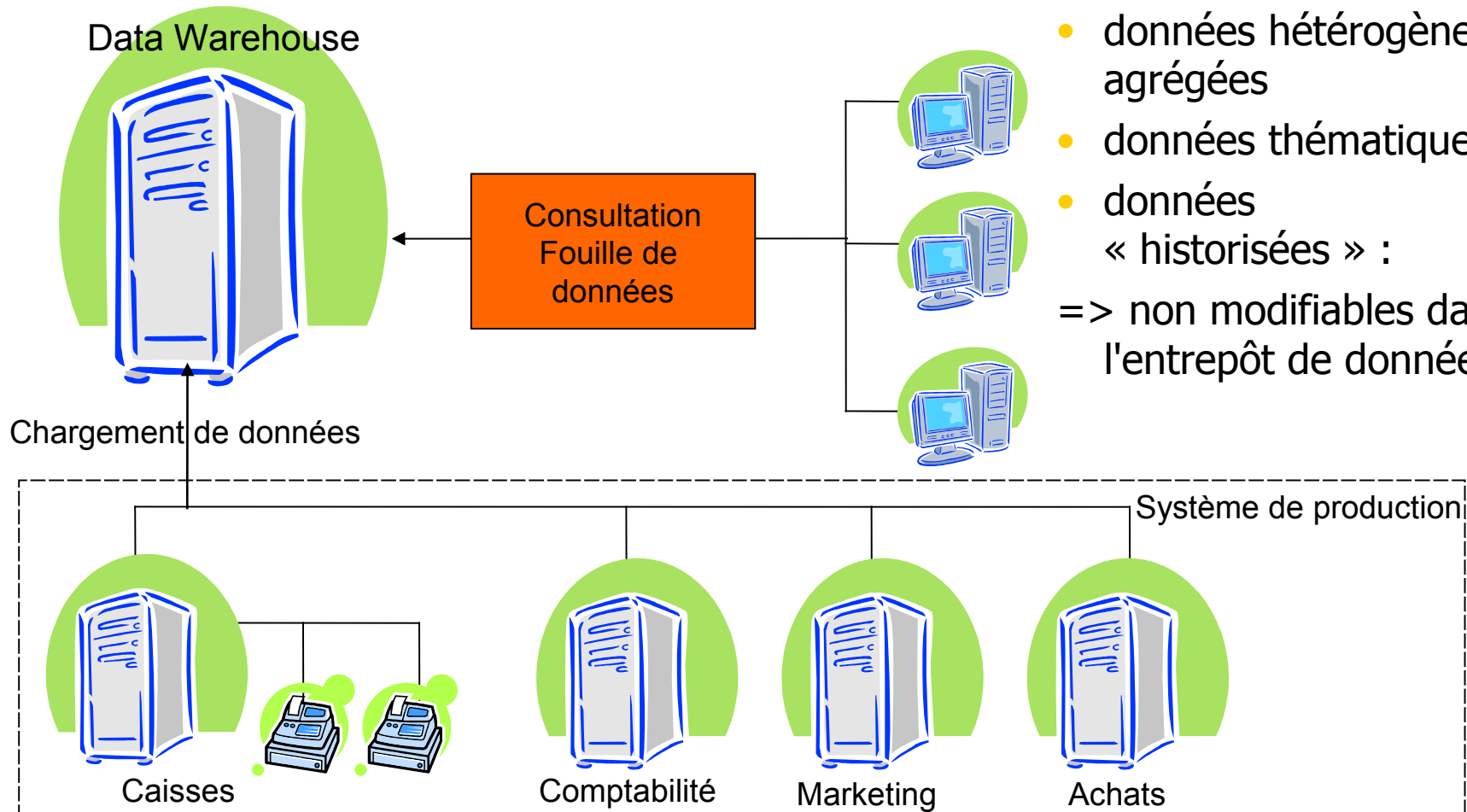




## 4.4. BD décisionnelles, entrepôts de données (« data warehouse »)

Aide à la décision dans l'entreprise :

- données hétérogènes agrégées
- données thématiques
- données « historisées » :  
=> non modifiables dans l'entrepôt de données



# OLAP (« On Line Analytical Processing »)

- | Architecture et outils d'analyse décisionnelle d'entrepôts de données
- | Analyse multidimensionnelle des données

## ■ Exemple : les lignes d'une facture

- | Réunion des informations nécessaires à l'analyse (date, produit, famille de produits, client, pays du client, magasin ou vendeur, quantité, chiffre d'affaires)
- | Différents niveaux d'agrégation (axes d'analyse) :
  - Catégories de produits : alimentaire / non alimentaire
  - Temps d'achat : jour, mois, année, période de marketing
  - Géographique : pays, région, magasin
  - Personnes : client, vendeur

## ■ Mise en oeuvre :

- | => « dénormalisation » de la base pour construire une table unique
- | => agrégation (redondance !) : chaque hiérarchisation des axes d'analyse correspond à une question (comparaison des vendeurs, des clients, des années)
- | => nécessite des performances exceptionnelles du SGBD et des navigateurs puissants

## 5. Evolution des SGBD

### | Avantages du modèle relationnel

- Simplicité de la structure
- Bases de données normalisées
- ACID : transactions sécurisées, données cohérentes

### | Inconvénients du modèle relationnel

- Langage mal adapté aux structures de données complexes (ex: données multimédia)
- Les jointures de tables sont coûteuses sur des tables volumineuses

- Extension objet des modèles relationnels :
  - Conserve les acquis du relationnel
  - Définit des sous-types par héritage
  - Utilisation de règles logiques pour le maintien de cohérence
  - Exemple : Oracle 8, DB2 Universal Database
- Meilleur support d'Internet et du Web
  - => interrogation d'objets multimédia distribués
  - => extraction de connaissances : fouille de données
  - => analyse multidimensionnelle
  - => NoSQL (« Not Only SQL »)

# NoSQL (1998): « scalabilité »

Bases de données non relationnelles qui manipulent d'énormes volumes de données (« Big Data »)

- adaptées aux applications manipulant des pages web : analyses, statistiques, etc.
- bases de données distribuées
- la table n'est plus l'unité logique => système « clé-valeur » avec base de données qui est un tableau « associatif » avec des milliards d'entrées
- SQL n'est en général pas utilisé

=> ne garantissent pas ACID : le maintien de la cohérence des données distribuées est très coûteux en temps

=> **logiciels de stockage de données plutôt que SGBD**

- Les grands acteurs d'Internet utilisent NoSQL :
  - Facebook utilise Cassandra (2500 fois plus rapide que MySQL) :
  - Google utilise « Big Table » (SGBD orienté colonnes adapté aux « Big Data »)
  - Amazon (Dynamo), LinkedIn (Project Voldemort)