

Cours

“Bases de données”

3° année (MIS)

Antoine Cornuéjols

www.lri.fr/~antoine
antoine.cornuejols@agroparistech.fr

Informations sur le cours

- 9 (10) **séances de 3h** (Cours + TD/TP)
- Polycopié
- Transparents
 - Disponibles sur :
<http://ouranos.inapg.inra.fr/wiki/>
 - Attention : remises à jour fréquentes

2

Informations sur le cours

- **Sources bibliographiques**
 - Andreas MEIER : “*Introduction pratique aux bases de données relationnelles*”, Springer-Verlag, 2006
 - Ramez ELMASRI & Shamkant NAVATHE : “*Conception et architecture des bases de données*”, Pearson Education, 2004

•

3

Introduction

PLAN Introduction

1. *Qu'est-ce qu'une base de données ?*
2. *Un système de gestion de base de données (SGBD)*
Pourquoi faire ?
3. *Les grands principes et architecture*
4. *Historique et évolution des SGBD*

4

Introduction

Bases de données

La bibliothèque de Leibniz (~ 1685)

🗄️ Caractéristiques

- 🗄️ Des données permanentes (livres)
- 🗄️ Un problème d'indexage
 - Précis
 - Fiable
 - Supportant des mises à jour

5

Introduction

Bases de données

● Définition (**base de données**)

- Une base de données est un **gros ensemble d'informations structurées** mémorisées sur un support **permanent** et qui peut être **partagée** par plusieurs applications et qui est **interrogeable par le contenu**.

● vs. Ensemble de fichiers

- *Lourdeur d'accès aux données*
- *Manque de sécurité*
- *Pas de contrôle de concurrence*

- Lourdeur d'accès : connaître le détail de l'implantation physique

- Manque de sécurité : N'importe qui peut modifier le fichier. Pas de contrôle de cohérence.

- Pas de contrôle de concurrence entre utilisateurs.

6

Introduction

Système de Gestion de Bases de Données (SGBD)

Définition (**SGBD**)

Un outil **informatique** permettant aux utilisateurs de **structurer**, **d'insérer**, de **modifier**, de **rechercher** de manière efficace des données spécifiques, au sein d'une grande quantité d'informations, stockées sur mémoires secondaires **partagée** de manière **transparente** par **plusieurs utilisateurs**.

7

Introduction

Bases de données vs. SGBD

🗄️ **Base de données**

Database

- 🗄️ **Grande masse** de données (Gigaoctets = 10^9 octets)
- 🗄️ **Mémoire secondaire** (disque dur, disque optique)
- 🗄️ **Liens abstraits/thématiques** entre données
- 🗄️ **Longévité** des données (des dizaines d'années)

🗄️ **SGBD**

Database Management System (DBMS)

- 🗄️ Gestion du **stockage** des données
- 🗄️ **Traitement** des données
 - Modification
 - Interrogation, extraction
 - Programmation d'applications

8

Introduction

Mémoire centrale vs. secondaire

● Mémoire **centrale** ou **primaire** ou **vive**

- Directement accessible par le processeur central
- Contient les instructions et les données d'un programme en exécution
- Rapidité
- Non persistance

● Mémoire **secondaire** ou **persistante**

- Indirectement accessible via des opérations d'entrée / sortie
- Lenteur
- Persistance

9

Introduction

Système de Gestion de Bases de Données (SGBD)

Définition (**SGBD**)

*Un système de gestion de bases de données (SGBD) est un logiciel de **haut niveau** qui permet de manipuler les informations stockées dans une base de données.*

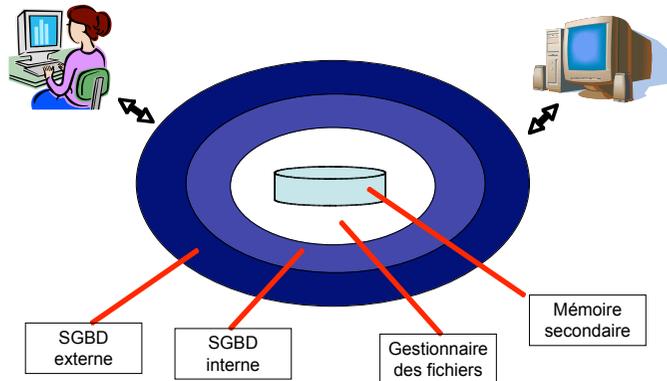
● Complexité (diversité des techniques mises en oeuvre) :

- Modèles de données
- Langages de requêtes
- Techniques de stockage
- Organisation des fichiers
- Architecture
- Optimisation des requêtes
- Concurrence d'accès et reprise sur panne

10

Introduction

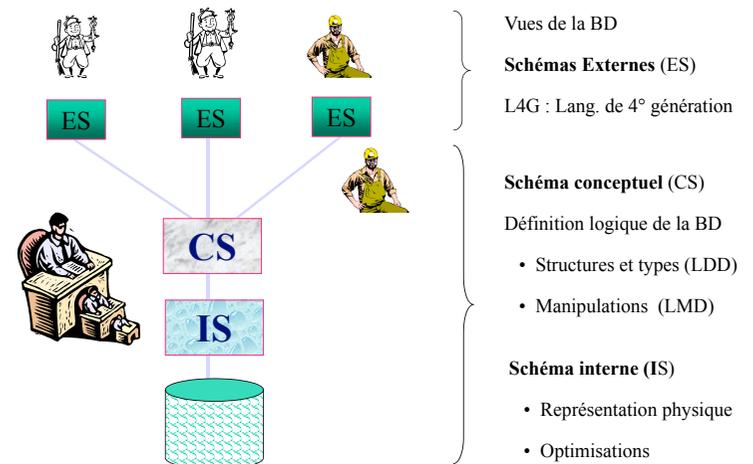
Système de Gestion de Bases de Données (SGBD)



11

Introduction

Architecture ANSI-SPARC



12

Introduction

Objectifs des SGBD (1)

- 🔑 **Indépendance *physique*** des programmes et des données :
 - 🔑 **Pouvoir modifier les schémas internes** sans modifier les schémas conceptuels et externes
- 🔑 **Indépendance *logique*** des programmes et des données :
 - 🔑 **Pouvoir modifier les schémas externes** sans modifier les schémas conceptuels
 - 🔑 Indépendance entre les différents utilisateurs
- 🔑 Manipulation des données par des **langages non procéduraux**
 - 🔑 Données facilement manipulables par les utilisateurs (interactifs ou programmeurs)
- 🔑 **Administration facile** des données
 - 🔑 Outils pour définir et modifier les définition de données

13

Introduction

Objectifs des SGBD (2)

- 🔑 **Efficacité d'accès** aux données :
 - 🔑 **Optimisation** : temps de réponse, débit, ...
 - 🔑 Optimisation des opérations d'E/S
- 🔑 **Redondance contrôlée** des données :
 - 🔑 Dans les **BD réparties** : redondance nécessaire, mais contrôlée
- 🔑 **Cohérence** des données
 - 🔑 Satisfaction de contraintes d'intégrité
- 🔑 **Partage** des données
 - 🔑 Permettre les accès concurrents
- 🔑 **Sécurité** des données
 - 🔑 Outils pour définir et modifier les définition de données
 - 🔑 Protection en cas de panne (du SGBD, de la machine, ...)
 - 🔑 Assurer l'atomicité des transactions et l'intégrité des données

14

Introduction

Applications

- 🔑 **Applications de gestion**
 - 🔑 Banque - Finance - Assurance
 - 🔑 Gestion du personnel / client des entreprises
 - 🔑 Gestion des stocks d'une entreprise commerciale
 - 🔑 Système d'information d'une entreprise
- 🔑 **Applications transactionnelles**
 - 🔑 Gestion des réservations
 - 🔑 Transport de passagers (train, avion, voiture, ...)
 - 🔑 Hôtels, Spectacles

Consultation / modifications très fréquentes

15

Introduction

Fonctionnalités d'un SGBD

1. **Persistence**
2. **Gestion du disque**
3. **Partage des données**
4. **Fiabilité des données**
5. **Sécurité des données**
6. **Indépendance logique / physique**
7. **Langage de description, d'interrogation et de traitement des données**

16

Introduction

Fonctionnalités d'un SGBD

1. Persistance

- Données stockées sur disque

2. Gestion du disque

- Techniques spécifiques pour de bonnes performances
 - Index, hash-coding
 - Regroupement des données sur disque
 - Optimisation des requêtes
 - Cache-mémoire

17

Introduction

Fonctionnalités d'un SGBD

3. Partage des données

- Chaque utilisateur doit avoir l'illusion d'être seul
 - Notion de transaction (begin, abort, commit)
- Cohérence des mises à jour effectuées par un utilisateur
- Cohérence collective : sérialisabilité

4. Fiabilité des données

- Vérification de contraintes d'intégrité
- Atomicité des transactions : transaction complètement effectuée ou pas du tout
- Résistance aux pannes
 - Si **panne mémoire** : restauration automatique de la base intégrant les dernières transactions validées avant la panne
 - Si **panne disque** : restauration d'une sauvegarde et déroulement du journal archivé
 - Mise en place d'un mécanisme de réplication synchrone de la base dans une base miroir : **mirroring**

18

Introduction

Fonctionnalités d'un SGBD

5. Sécurité des données

- Tous les utilisateurs ne peuvent pas tout faire sur toutes les données
 - Notion de groupes d'utilisateurs
 - Notion d'autorisation (lecture, écriture, exécution)
 - Granularité des autorisations : base de données, table, colonne, n-uplet, ...
 - Possibilité d'accorder ou de supprimer des droits

6. Indépendance logique / physique

- Organisation physique de la BD transparente pour le développeur d'application
- On doit pouvoir changer la répartition des données sur le disque (ex : un regroupement ou la pose d'un index) sans changer le code de l'application manipulant les données

19

Introduction

Fonctionnalités d'un SGBD

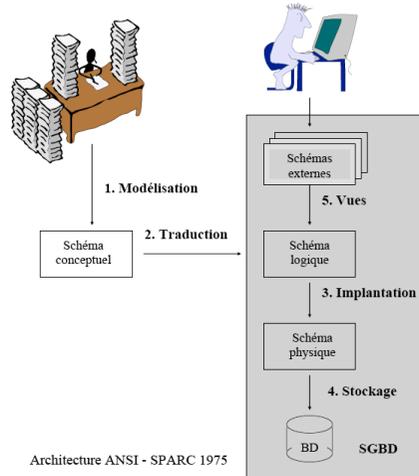
7. Langage de requête

- Les requêtes doivent être :
 - Simples
 - Déclaratives
 - Optimisées avant leur exécution
- État de l'art :
 - **SQL** (Structured Query Language) : mélange d'algèbre relationnelle et de calcul relationnel
 - **QBE** (Query By Example) : s'appuie sur le calcul relationnel, mais permet de formuler des requêtes et d'effectuer des manipulations de données à l'aide de représentations graphiques
 - **OQL** (Object Query Language) : conçu pour s'intégrer avec des programmes écrits en langage objet (C++, Smalltalk, Java)
 - **XQUERY** (XML Query language)

20

Introduction

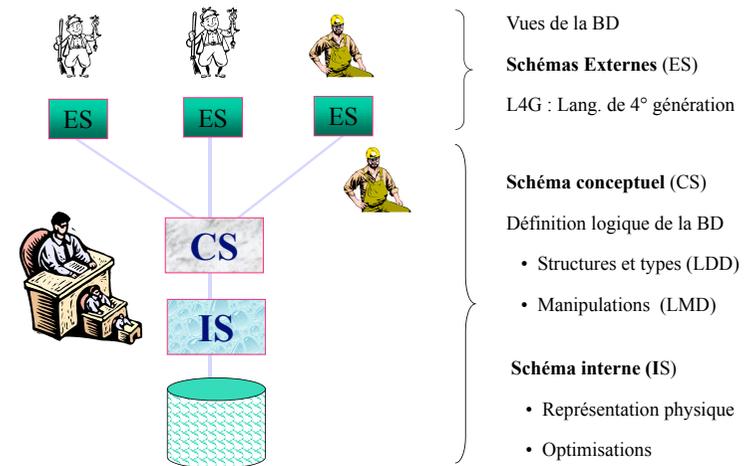
Conception d'une base de données



21

Introduction

Architecture ANSI-SPARC



22

Introduction

Architecture d'une base de données

3 niveaux :

- 📍 **Niveau externe** : Vues ; Environnement de programmation (intégration avec un langage de programmation) ; Interfaces conviviales et Langages de 4^e génération (**L4G**) ; Outils d'aide (e.g. conception de schémas) ; Outils de saisie, d'impression d'états.
- 📍 **Niveau logique** : Définition de la structure de données ; Langages de description des données (**LDD**) ; Consultation et mise à jour des données : Langages de Requêtes (**LR**) et Langage de Manipulation de Données (**LMD**) ; Gestion de la confidentialité (sécurité) ; Maintien de l'intégrité.
- 📍 **Niveau physique** : Gestion sur mémoire secondaire (fichiers) des données, du schéma, des index ; Partage des données et gestion de la concurrence d'accès ; Reprise sur pannes (fiabilité) ; Distribution des données et interopérabilité (accès réseaux).

📍 Trois types d'intervenants :

- 👤 Utilisateur naïf
- 👤 Concepteur et programmeur d'applications
- 👤 Utilisateur expert (administrateur de la base)

23

Introduction

Définition du schéma des données : trois modèles

- 📍 **Le modèle conceptuel** :
 - 👤 La description du système d'information
- 📍 **Le modèle logique** :
 - 👤 Interface avec le SGBD
- 📍 **Le modèle physique** :
 - 👤 Les fichiers

24

Introduction

Exemple : le modèle "entité-association"

Un modèle conceptuel :

- ☛ L'analyse du monde réel
- ☛ La conception du système d'information
- ☛ La communication entre différents acteurs de l'entreprise

Mais ne propose pas d'opérations

Des modèles logiques

- ☛ Un langage de définition de données (LDD)
 - Structure et contraintes
- ☛ Un langage de manipulation de données (LMD)

```
nom character 15, not null
âge integer between 0 and 150
débit = crédit
...
```

25

Introduction

Des langages abstraits

Le LDD

- ☛ indépendant de la représentation physique des données

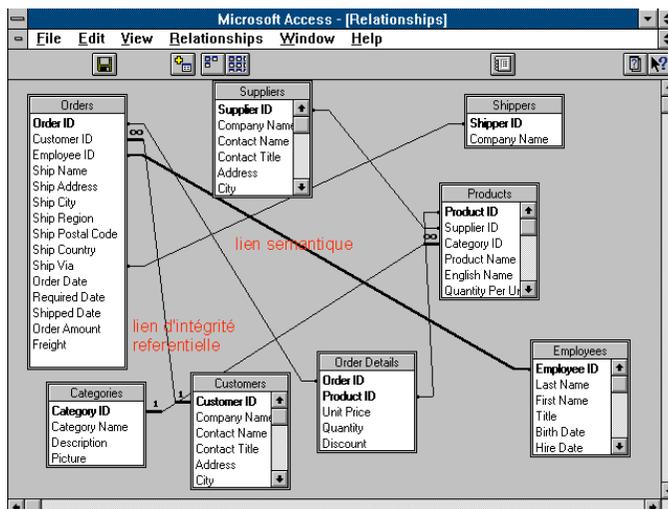
Le LMD

- ☛ indépendant de l'implantation des opérations

26

Introduction

Outils de description de schéma conceptuel



Introduction

Conception d'une application pour une BD

1. *Décider de la structure logique*
2. (Décider de la structure physique)
3. *Écrire des programmes d'application en utilisant la structure logique*
4. Le SGBD se charge de transcrire les commandes du LMD en instructions appropriées à la représentation physique

28

Introduction

Opérations sur les données

Quatre opérations classiques (LMD)

1. La **création** (ou *insertion*)
2. La **modification** (ou *mise-à-jour*)
3. La **destruction**
4. La **recherche**

- Faciles à exprimer
- Efficacité
- Fiabilité

29

Introduction

Optimisation

Optimisation d'une requête

- s'appuie sur l'organisation physique des données
 - Fichiers séquentiels
 - Index (denses, secondaires, arbres B)
 - Regroupement des données par hachage

Optimiseur dans le SGBD

- Choisit le meilleur séquençement des opérations

30

Introduction

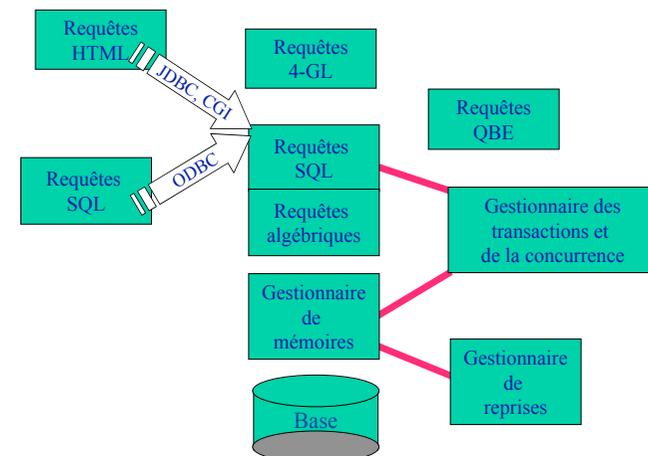
Concurrence d'accès

- **Gérer les conflits**
 - si plusieurs utilisateurs font des mises-à-jour simultanées
- Offrir un **mécanisme de retour en arrière**
 - si annulation d'opérations
- **Donner une image cohérente** des données
 - si un utilisateur fait des requêtes et un autre des mises-à-jour

31

Introduction

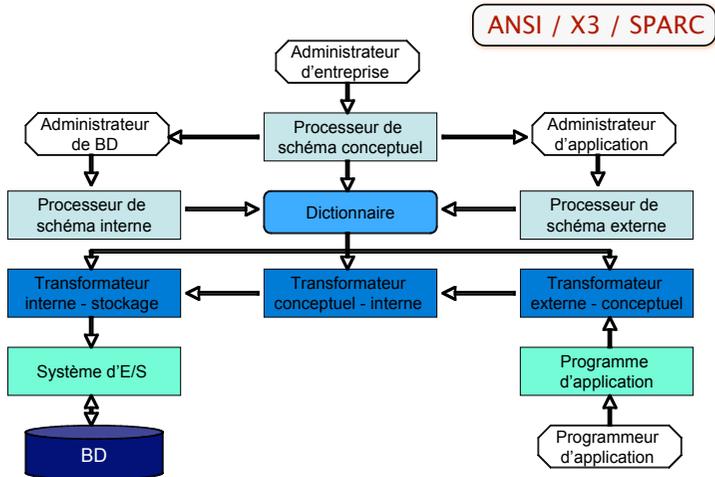
Architecture fonctionnelle d'un SGBD relationnel



32

Introduction

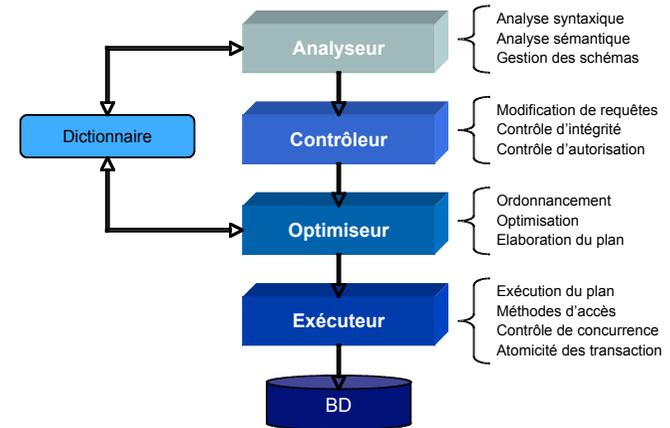
Architecture fonctionnelle d'un SGBD relationnel



33

Introduction

Architecture fonctionnelle typique d'un SGBD



34

Introduction

Avertissements

Un SGBD est un système complexe !

Spécificités d'un SGBD

- Trois niveaux d'abstraction
- Fiabilité
- Concurrence
- Stockage massif sur disque

Prix à payer

- Interfaces entre niveaux
- Journalisation
- Verrouillage
- Techniques spécialisées d'accès aux données sur disque

35

Introduction

Avertissements

Un SGBD est un système difficile à bien utiliser !

Méthodes de développement et maintenance

- Analyse des besoins de l'application
- Conception du schéma conceptuel
- Conception du schéma interne
- Réglage (tuning) des applications

36

Introduction

Architectures

Architecture BD centralisée



Usagers



Mainframe

- programme d'application et SGBD sur même machine (même site)
- premiers systèmes

37

Introduction

Architectures opérationnelles

Du point de vue opérationnel

- Un SGBD est un ensemble de processus et de tâches qui supportent l'exécution du code du SGBD pour satisfaire les commandes des utilisateurs.

- Depuis les années '80, les SGBD sont basés sur une **architecture clients-serveur**

Serveur : Gère les données partagées et exécute le code du SGBD

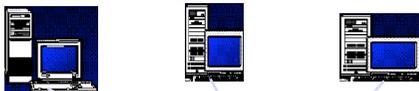
Clients : Communiquent avec le serveur. Requêtes de type question / réponse

38

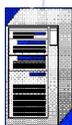
Introduction

Architectures

Architecture client-serveur



Clients



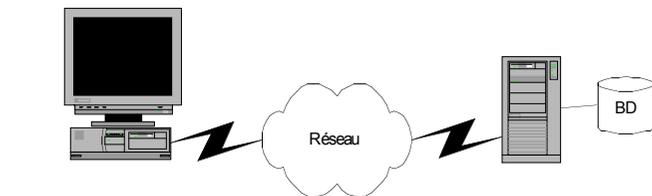
Serveur

39

Introduction

Architectures

Architecture client-serveur



Client

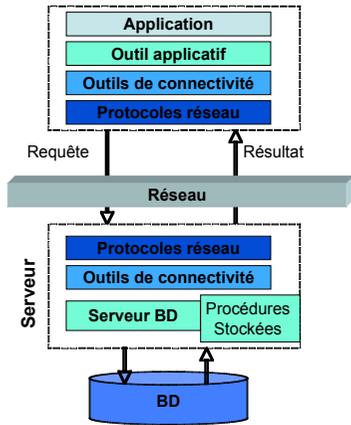


Serveur

40

Introduction

Architectures *Architecture deux strates*



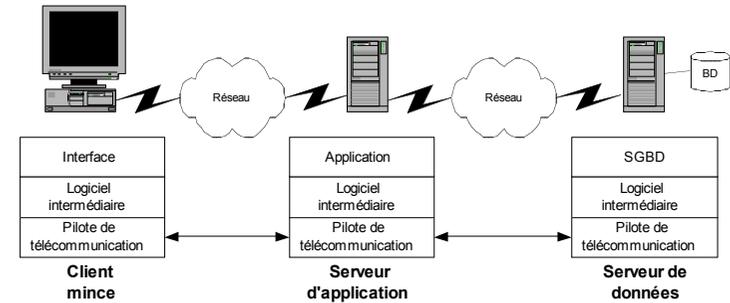
- Faiblesses :
- > Nécessité de spécialistes réseau, BD, PC
 - > Interfaces hétérogènes (Windows, Mac, ...)
 - > Peu portables
 - > Coûts élevés pour la maintenance

Avec l'apparition du Web
 ➔ architectures à trois strates

Introduction

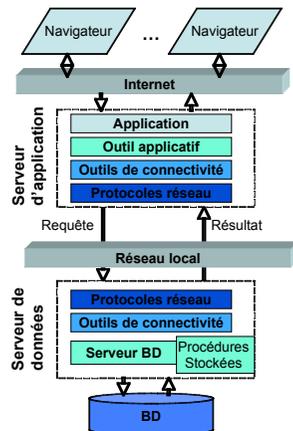
Architectures

Architecture à trois niveaux : three tier



Introduction

Architectures *Architecture trois strates*



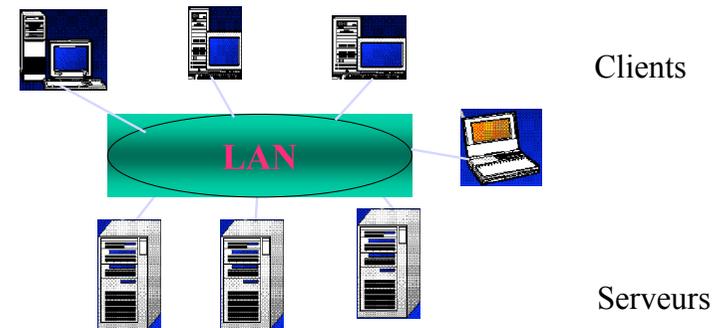
- Avantages :
- > Portabilité
 - > Interfaces homogènes pour les clients (navigateurs)
 - > Support de l'hypermédia
- Inconvénient (mais aussi un avantage)
- > La centralisation de la gestion des données au niveau du serveur

➔ architectures réparties

Introduction

Architectures

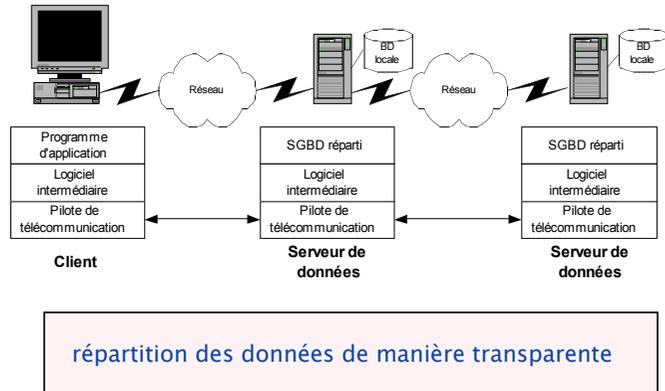
Architecture BDR / BDP / MDP (BD réparties / BD parallèles / Multi DB)



Introduction

Architectures

Base de données répartie, distribuée



45

Introduction

Architectures

Architecture BDR / BDP / MDP (BD réparties / BD parallèles / Multi DB)

Base de Données Répartie (BDR)

Une BD avec les fragments dans les SGBDs sur différents sites d'un réseau

Si c'est un réseau local et la répartition est une partition au niveau physique, alors c'est une *BD parallèle* (BDP)

Multibase (MBD)

Une collection de bases de données munie d'un langage multibase

Sur différents sites ou le même

On les appelle aussi BDs intéroperables

En général ces BDs sont autonomes et hétérogènes

46

Introduction

Historique

60-70 SGBD Réseau

IDS (General Electric), APL, DMS 1100, ...,
ADABAS (Software AG)

60-70 SGBD Hiérarchique

ISM IBM www.software.ibm.com
System 2000 MRI

gestion de pointeurs entre enregistrement

problème : pas d'*indépendance physique*

47

Introduction

Historique

- Modèle *relationnel* (conceptuel)
 - parmi ses pères : *Ted Codd* (1970)
 - un modèle simple description = table
 - un modèle mathématique
 - théorie des ensembles
 - logique
- *SQL* (Structured Query Language)
 - un langage normalisé !!!

recherche ← 20 ans → systèmes commerciaux

48

Introduction

Historique

70-80 **SGBD Relationnel** :

System R (San José), **Ingres**, ...

Informix	Informix	
Oracle	Oracle	www.oracle.com
DB2	IBM	www.software.ibm.com
Sybase	Sybase	www.sybase.com
SQL Server	Microsoft	www.microsoft.com

49

Introduction

Historique

- **Les machines:**

1960 : Stocker un giga-octet → **plusieurs disques**.

SGBD → système complexe + coûteux +

SGBD → ordinateur puissant

SGBD → **machine base de données**

2000 : Un giga-octet → **c'est rien!**

SGBD → installation sur un portable

aussi commun qu'un traitement de texte!

... parfois confondu avec les outils de bureautique!

SGBD → **Client-Serveur**

50

Introduction

Historique

- **Les besoins**

1960 : Un Giga octet → **énorme**

Mémoire	coûteux
Applications	gestion, stock
Données	type alphanumérique

2000 : Un Giga octet → **peu**

Mémoire	bon marché
Applications	CAO , Genome, Mulimédia
Données	image, son, video
Besoin	Tera-octets (1000 Giga-octets= 10^{12} octets)

1 heure de video = 1 Giga octets

images satellite = peta octets (10^{15} octets)

51