

Les bases de données: introduction

Mohamed EL ADNANI

Plan

- Définitions générales
- Les objectifs d'un SGBD
- Utilités des BD
- Composantes d'une BD
- Architecture ANSI-SPARC
- Architecture fonctionnelle d'un SGBD
- Architectures évoluées

Base de Données: définition et manipulation

- **Définition** : c'est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun
- Elle supporte des *définitions* et des manipulations.
- **Définition de**
 - Données de la base
 - Structure intégrée
 - Lien sémantiques
 - Contraintes d'intégrité
 - Contraintes de sécurité
 - Vues de la base

BD

- ***manipulations de***
 - **Recherche de données**
 - Interactive
 - Assertionnelle
 - 4-GL
 - Logiquement et physiquement performante
 - **Insertion, Mise à jour, Suppression de données**
 - Cohérente
 - Partagée
 - Fiable
- Une BD efficace offre les opérations suivantes :
 - stockage de nouvelles données,
 - recherche de données particulières,
 - calculs divers sur des sous-ensembles de données

SGBD

- Système de Gestion de Base de Données (SGBD)
 - ❑ C'est un ensemble de programmes capable de gérer une base de donnée de façon satisfaisante en permettant à l'utilisateur de définir des données, de consulter ou de mettre à jour la BD.
 - ❑ Système logiciel gérant une BD
 - Peut avoir des composantes matériel
 - ❑ Mono ou multiordinateur
 - ❑ En général, peut gérer plusieurs BDs
 - ❑ Peut aussi accéder aux BDs d'autres SGBDs

Les objectifs d'un SGBD

■ **Les compétences premières**

- ❑ permettre les descriptions des données qui seront stockées dans la base. Cette description est faite en deux niveaux :
 - physique : sous l'angle du codage informatique
 - logique : sous l'angle intéressant l'utilisateur
- ❑ permettre l'extraction ou la modification des données. Le système doit permettre l'accès aux données par l'intermédiaire de langages déclaratifs (non procéduraux) et de haut niveau que l'on appelle classiquement langage de requêtes.

Les compétences premières

- ❑ Préserver l'intégrité de la base de données. Les données sont à tout moment valides et cohérentes par rapport à la réalité.
- ❑ Protéger certaines données. Vérifier que l'utilisateur a effectivement le droit de regarder ou de modifier la donnée demandée
- ❑ Gérer les concurrences d'accès quand plusieurs utilisateurs demandent en un même temps l'accès à une même donnée.
- ❑ Permettre la remise en état de la base et le redémarrage des opérations après un incident quel qu'il soit.

Qualités supplémentaires

- Une BD réelle est une entité vivante, elle doit pouvoir supporter des modifications au cours de son existence sans que celles ci perturbent la base existante ni les utilisateurs actuels.
- Pour obtenir cette qualité, on introduit la notion **d'indépendance données/programme**.
- On distingue deux niveaux d'indépendance :
 - ❑ l'indépendance physique :
 - ❑ l'indépendance logique.

l'indépendance P/L

■ l'indépendance physique :

- ❑ doit permettre de modifier les structures de stockage ou les méthodes d'accès aux données sans que cela ait des répercussions au niveau des applications.

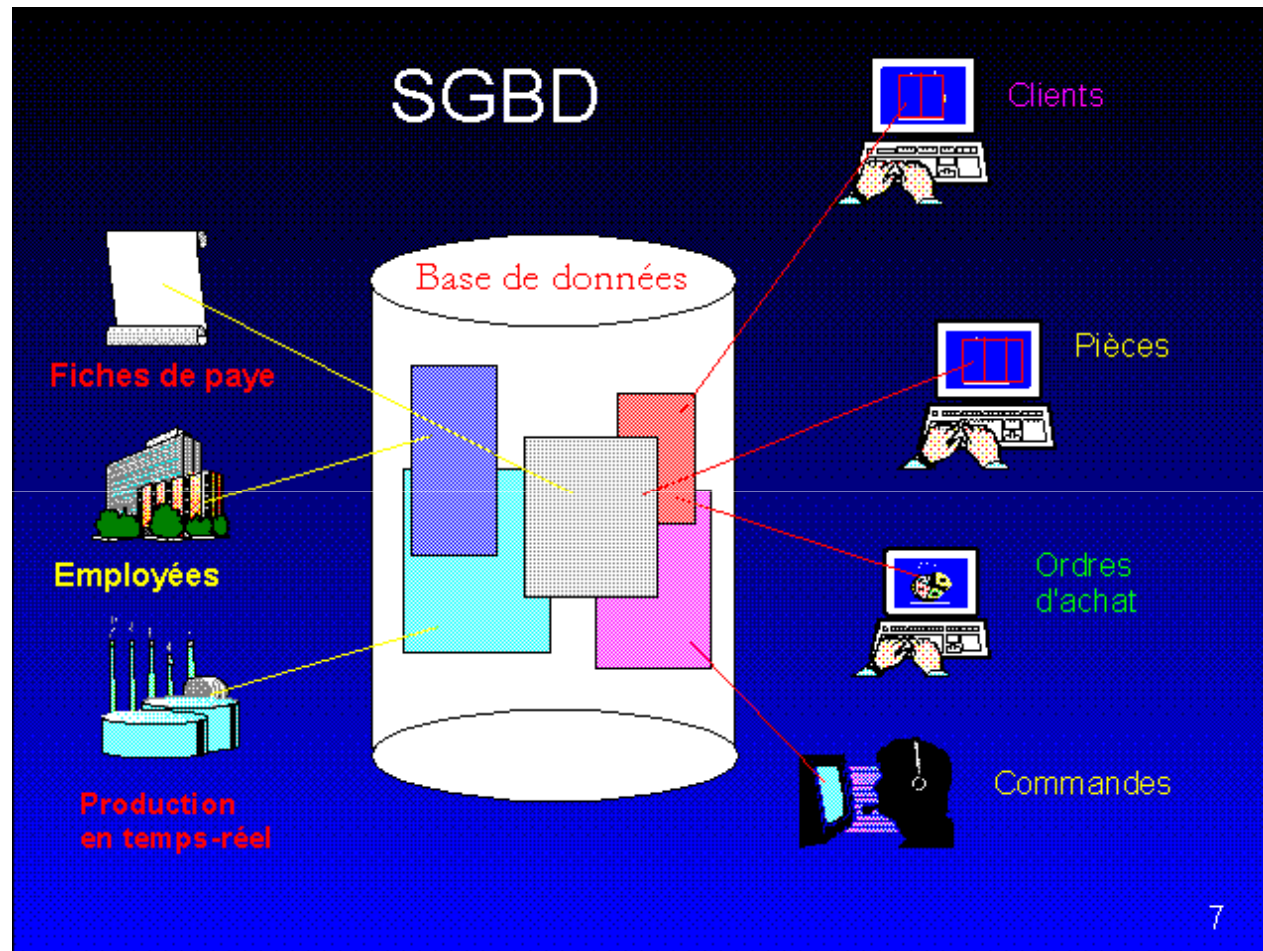
■ l'indépendance logique :

- ❑ doit permettre à l'utilisateur de modifier l'organisation des données qui l'intéressent sans affecter les autres utilisateurs. Ainsi la BD peut évoluer selon les besoins par l'introduction ou le retrait de nouvelles classes de données

Qualités supplémentaires

- Pour aider à la réalisation de ces objectifs, le SGBD offre des outils de programmation adaptés :
 - un langage de description de données et des liens existant entre elles : LDD
 - un langage d'interrogation et de manipulation des données : LMD
 - la possibilité de définir des contraintes d'intégrité
 - la possibilité de définir des clauses de confidentialité sur les données et des droits d'accès pour les utilisateurs externes
 - des mécanismes plus ou moins transparents de stockage de données.

SGBD



Pourquoi une Base de Données ?

- En pratique, il est impossible de concevoir ces fonctions soi-même
 - Théorie complexe
 - Conception d'une BD
 - Formes normales...
 - Langages de définition et de manipulation d'une BD
 - Optimisation de requêtes assertionnelles
 - Transactions et partage de données
 - ...etc
 - Logiciel complexe
 - des centaines de milliers ou des millions de lignes de programmation

Différentes BDs

- Il y en a pour tous les goûts
- BDs personnelles
 - MsAccess etc.
 - 10 KO – 100 KO
- BDs professionnelles typiques
 - 100 KO – 100 GO
- BDs professionnelles très grandes
 - Very Large Databases (VLDB)
> 100 GO

Composantes d'une BD : Logiciel

- SGBD
 - Gère le niveau logique et physique de la base
 - Selon l'architecture ANSI-SPARC (voir plus loin)
 - Les outils frontaux (4-GL) ou complémentaires
 - Générateurs : de formes, de rapports, des applications
 - Intégrés au SGBD ou externes
 - Powerbuilder, Borland...
 - Interfaces WEB : HTML, XML...
 - Interfaces OLAP & Data Mining
 - Intelligent Data Miner (IBM)
 - Utilitaires: chargement, statistiques, aide à la conception...
-

Composantes d'une BD : Matériel

- Ordinateur générique
 - ❑ avec son CPU, RAM, disque pour la BD, bandes pour la sauvegarde
 - ❑ RAM est considérée traditionnellement trop petite pour une BD
 - Ce n'est plus toujours vrai
 - ❑ Problème classique d'organisation d'E/S pour une BD

Composantes d'une BD : Matériel

- Machine spécialisée (MBD)
 - ❑ Ne supporte que la BD
 - ❑ En général multiprocesseur à partage de rien
 - Teradata avec jusqu'au 1024 Pentiums
 - ❑ Les applications sont sur d'autres ordinateurs
 - ❑ Liaison par les réseaux LAN

Usagers: trois niveaux

- Interactifs (ad-hoc)
 - ❑ Cherchent les infos, sans connaître la BD
 - ❑ Interfaces visuelles: 4-GL, Web...
 - ❑ A la rigueur peuvent utiliser le langage QBE
- Programmeurs d'application
 - ❑ Construisent les interfaces pour les usagers interactifs
 - ❑ Spécialistes de SQL
- Administrateur : DBA
 - ❑ Définit et maintient la BD
 - ❑ A la priorité sur tous les autres usagers

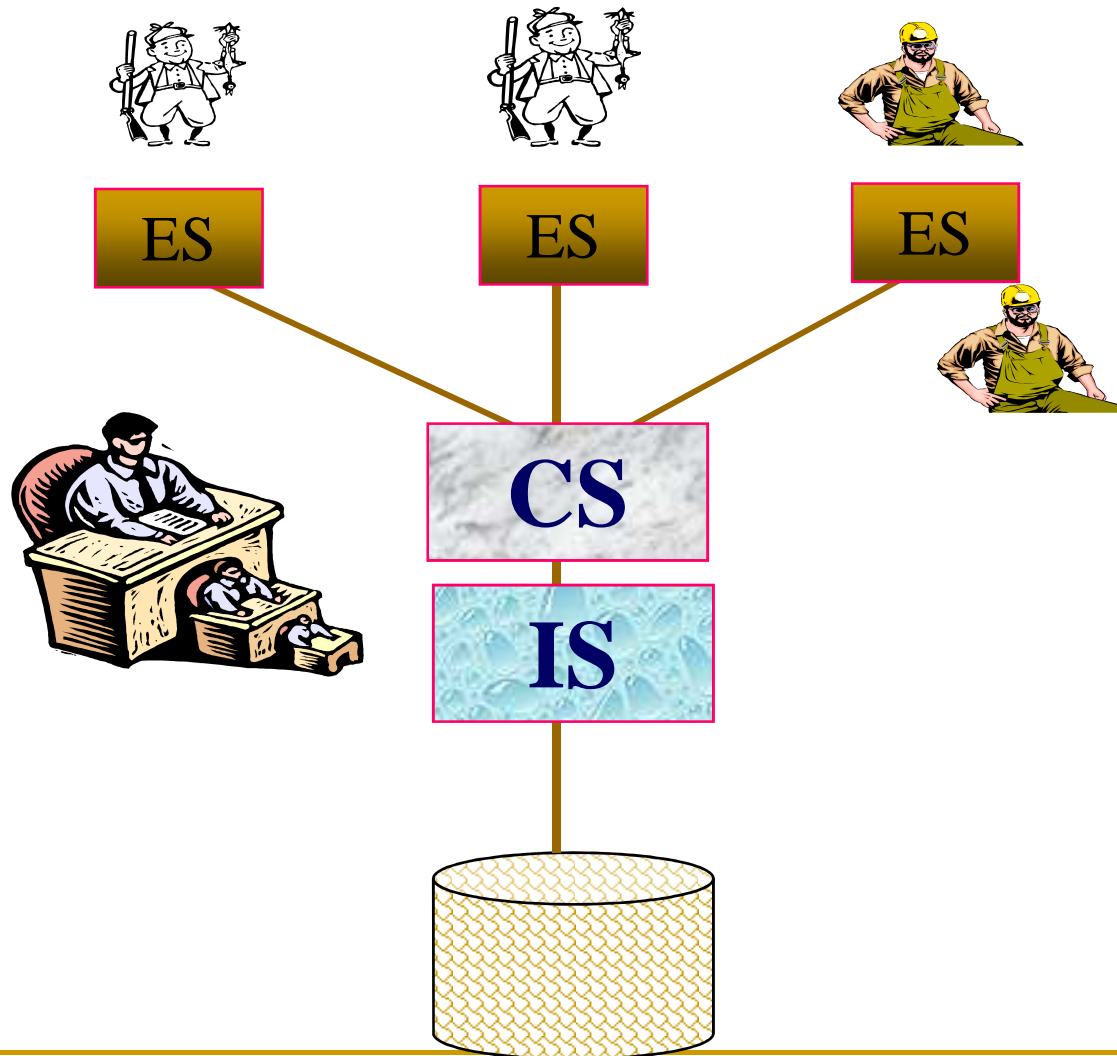
Types de SGBD: Par modèle de données

- 1ère génération 1950 – 65
 - Système de gestion de fichiers SGF,
 - SGF généralisés avec les langages booléens de manip.
- 2ème génération 1965 - 70
 - SGBD navigationnel
 - Hiérarchique (IMS), Réseau (Codasyl), Pseudo-relationnel
- 3ème génération 1969 - ...
 - SGBD relationnel (DB2, Oracle, Informix, MsAccess...
- SGBD OO 1990 - 1999
 - Pour supporter les nouvelles applications (O2, Objectstore, Objectivity..)
- SGBD relationnel – objet (RO) 1993 - ...
 - Évolution probable de tout SGBD relationnel

Types de SGBD : Par capacité

- « Mainframe ou cluster de mainframes »
 - DB2, Adabas...
- Machine base de données
 - Teradata
- Serveurs Unix & Windows 2000
 - DB2, Informix, Oracle, Sybase, Unify, SQL Server, ...
- Personnels
 - MsAccess, Paradox...

Architecture ANSI-SPARC



Architecture ANSI-SPARC

- Un standard pour tout SGBD digne de ce nom
- Proposé vers 1965 par Charles Bachman
 - Auteur du concept de la BD
 - Concepteur de IDMS-2
 - Le 1er SGBD moderne
 - Créateur du modèle de données réseau (Codasyl)
 - Plus tard de l'architecture Open System Interconnection (OSI)
 - Prix Turing
 - La plus haute récompense scientifique en informatique au USA

Architecture ANSI-SPARC

Schéma Conceptuel (CS)

- D'une manière abstraite: un *modèle* conceptuel de l'univers réel de la BD qui est le résultat de l'analyse de ce réel par application des méthodes de *conception* de la BD
- D'une manière appliquée : la définition **logique** de la BD
 - Une et une seule BD
 - Les données logiques, leurs structures et leurs types
 - Relations, attributs, domaines
 - Entités...
 - Objets, Types, Classes
 - Leur manipulations
 - procédures, fonctions, méthodes

Architecture ANSI-SPARC

Schéma Conceptuel (CS)

- La définition logique de la BD (suite)
 - Les liens sémantiques
 - Données d'un même objet réel
 - Les contraintes d'intégrité
 - Mono-valeur
 - Référentielles
 - Variées
 - Les contraintes de sécurité
 - Qui peut manipuler quoi

Architecture ANSI-SPARC

Schéma Conceptuel (CS)

- Le CS est défini par l'administrateur seul pour assurer la :
 - Complexité
 - Sécurité
- Par l'intégration de données des différentes applications de la BD
 - Plusieurs méthodes de conception, plus ou moins formelles
 - Entité-Relations
 - Objets & Fonctions
 - Merise
 - Normalisation relationnelle

Architecture ANSI-SPARC

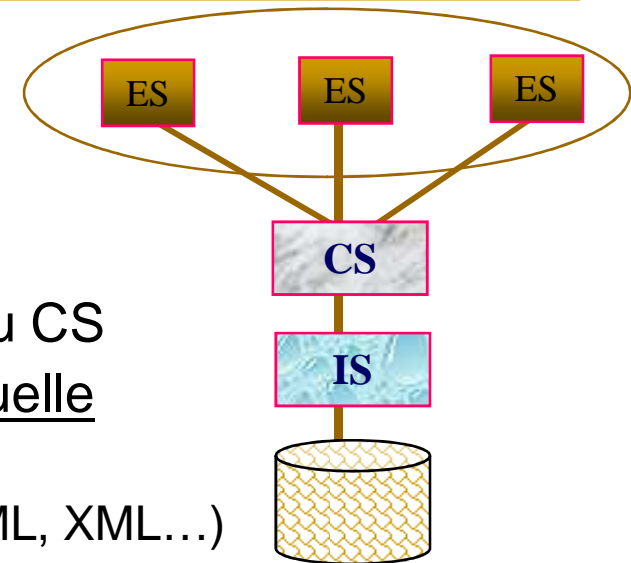
Schéma Conceptuel (CS)

- La BD (donc le CS) est défini en utilisant :
 - Le *langage de définition de données LDD*
 - Le *langage de manipulation de données LMD*
- Les deux sous langages forment:
 - Le *langage de base de données*
 - En général incomplet au sens de la machine de Turing
 - SQL pour une BD relationnelle

Architecture ANSI-SPARC

Schémas Externes (ESs)

- Un ES = un sous-schéma d'une BD dérivé du CS
- Définit une *vue* de la BD : une sous-base virtuelle
 - Pour des usagers de haut niveau ad-hoc
 - Vues 4-GL, orientés Web notamment (HTML, XML...)
 - Pour des programmeurs d'application
 - Vues SQL, pour SQL imbriqué ou Vbasic, ou procédures stockées...
- Une BD est en général munie de plusieurs différents ESs
- Mais tous ont le même CS comme racine commune
 - Donc tous doivent être acceptés par le DBA



Architecture ANSI-SPARC

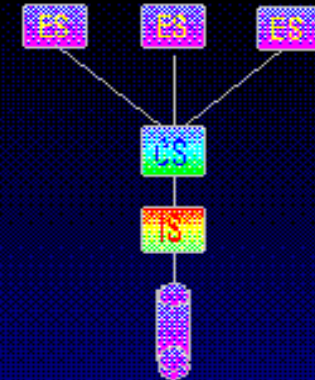
Schéma Interne (IS)

- Définit la représentation interne de la BD
 - Niveau interne ou *physique*
 - *Les disques, fichiers hachés, arbres-B... contenant la BD*
 - *La représentation physique de valeurs de données*
 - *Réel, entier, texte, OLE...*
 - *Encodage...*
- Définit l'application du CS sur le IS
 - Selon le principe de l'indépendance de niveaux logique et physique

Architecture ANSI-SPARC

Architecture ANSI-SPARC

Origine de l'idée



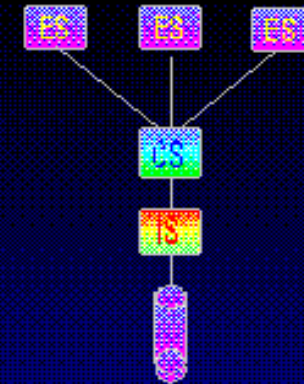
- ✓ Eviter les problèmes des SGF de 1ère génération
 - ✓ Existence de fichiers multiples et indépendants avec les données
 - ✓ dupliquées
 - ✓ hétérogènes
 - ✓ Difficultés d'assurer:
 - ✓ les mises à jour cohérentes de duplicata par les applications indépendantes
 - ✓ sécurité de l'ensemble de données
- ✓ L'existence du niveau conceptuel et physique réels **uniques** et, par contre, de vues externes **multiples** mais virtuels et est une réponse **théorique** à ce besoin

Architecture ANSI-SPARC

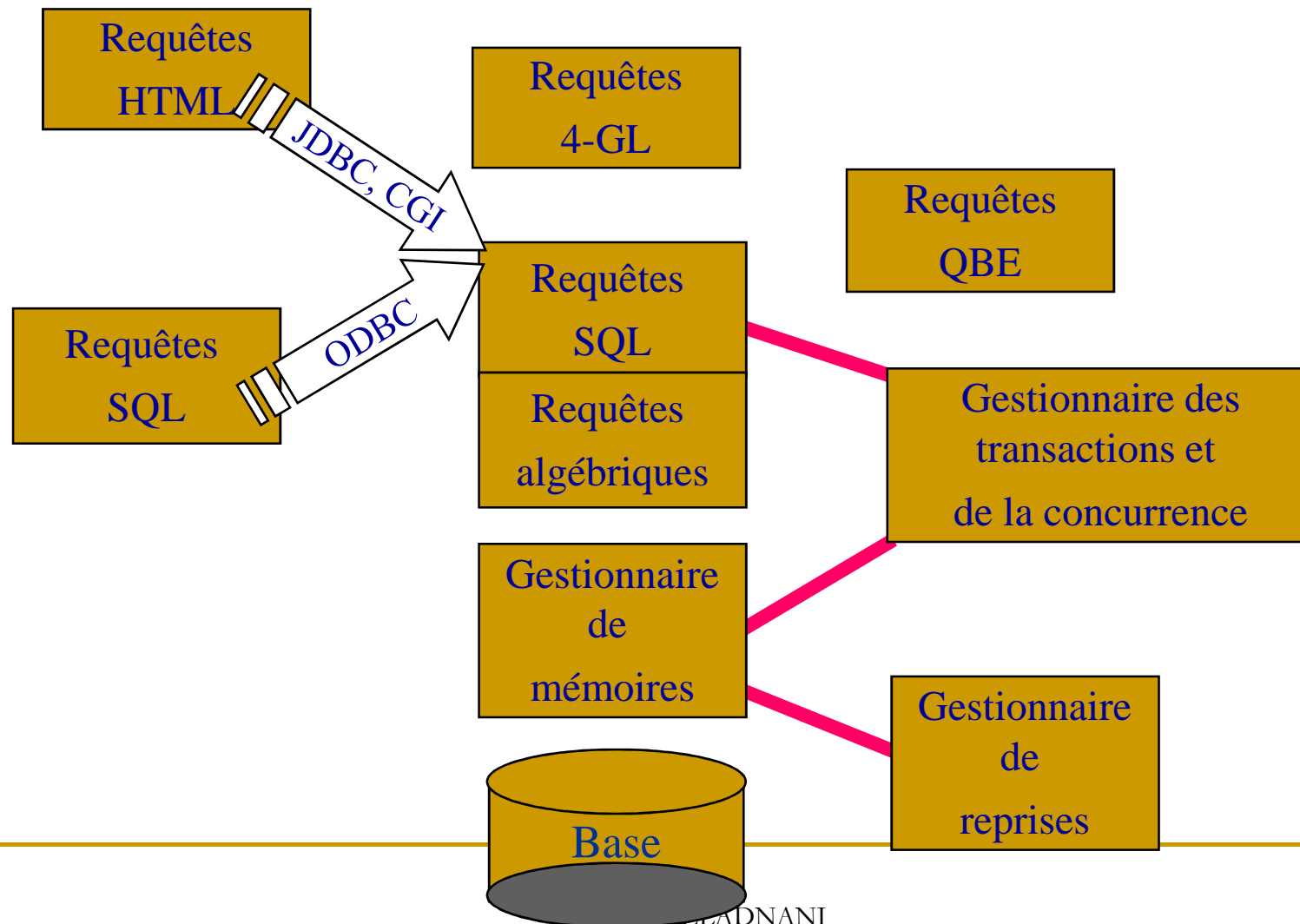
Architecture ANSI-SPARC

Limitations

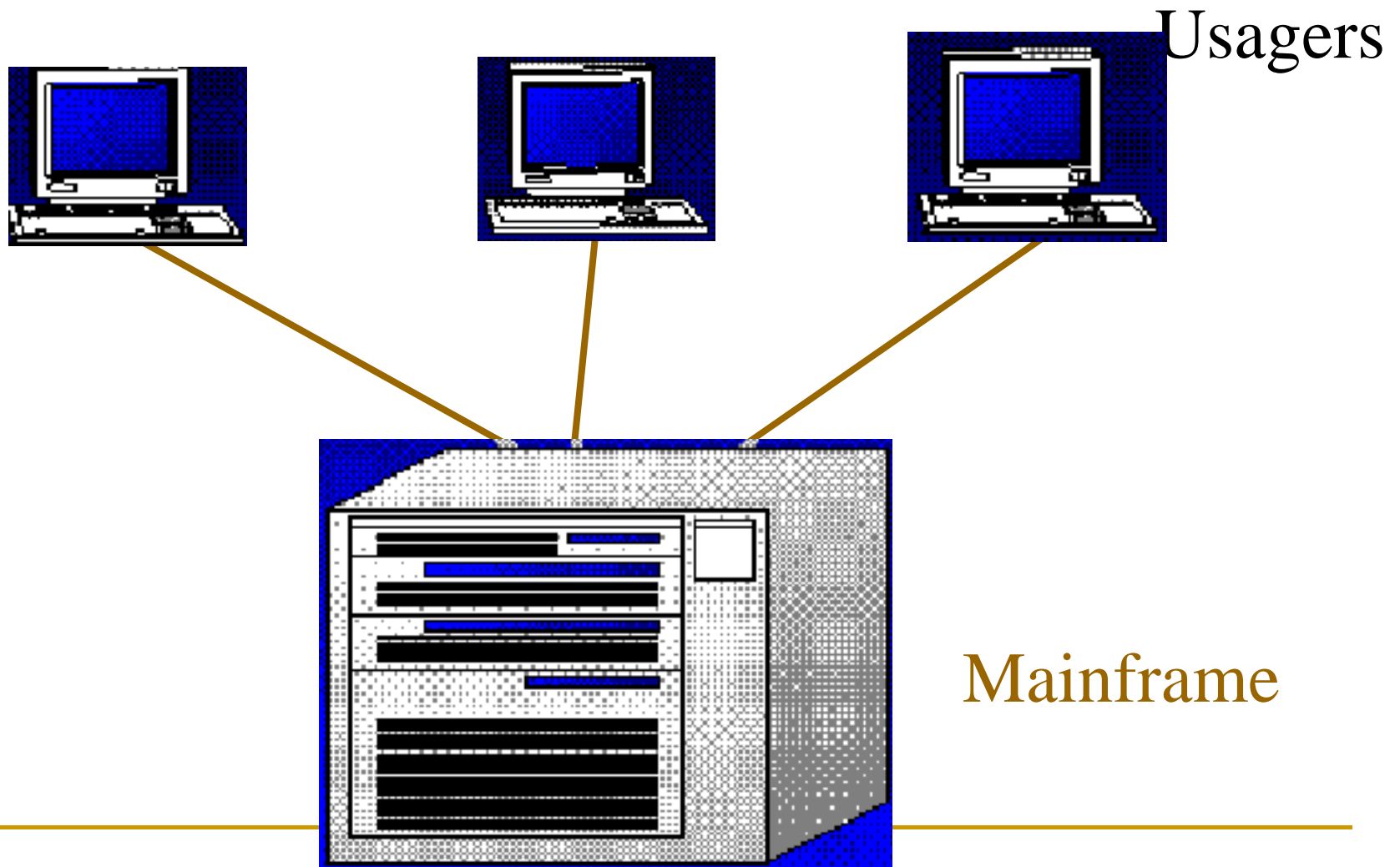
- Notion d'une base unique au niveau d'une entreprise
 - en général impossible en pratique
 - complexité organisationnelle
 - complexité technique
- Imprécisions de concepts fondamentaux
 - entreprise, BD...
- Notion de vue
 - limites théoriques sur la mise à jour de vues
- Distinction entre les langages de déf. et de manip.
 - Une vue est souvent définie par un langage de manipulation
 - Les objets sont définis par des manipulations qu'ils supportent
- **Néanmoins, l'architecture ANSI-SPARC était une idée capitale pour l'informatique moderne**



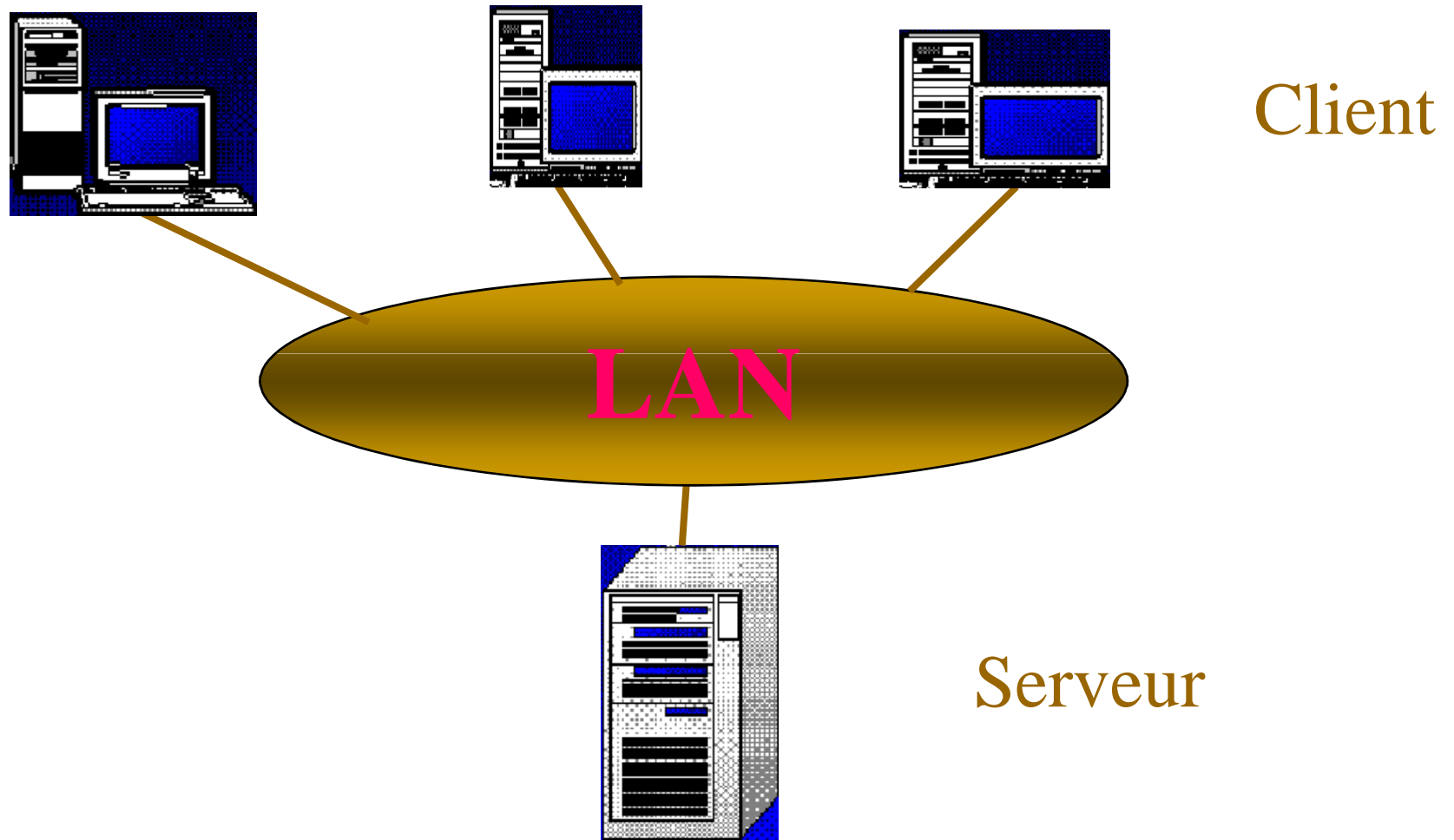
Architecture fonctionnelle d'un SGBD relationnel



Architecture BD Centralisée



Architecture Client-Serveur



L'architecture Client-Serveur

- Définition
 - ❑ modèle d'architecture applicative où les programmes sont répartis entre processus clients et serveurs communiquant par des requêtes avec réponses.
- Une répartition hiérarchique des fonctions
 - ❑ données sur le serveur partagées entre N clients
 - ❑ interfaces graphiques sur la station de travail personnelle
 - ❑ communication par des protocoles standardisés
 - ❑ distribution des programmes applicatifs afin de minimiser les coûts

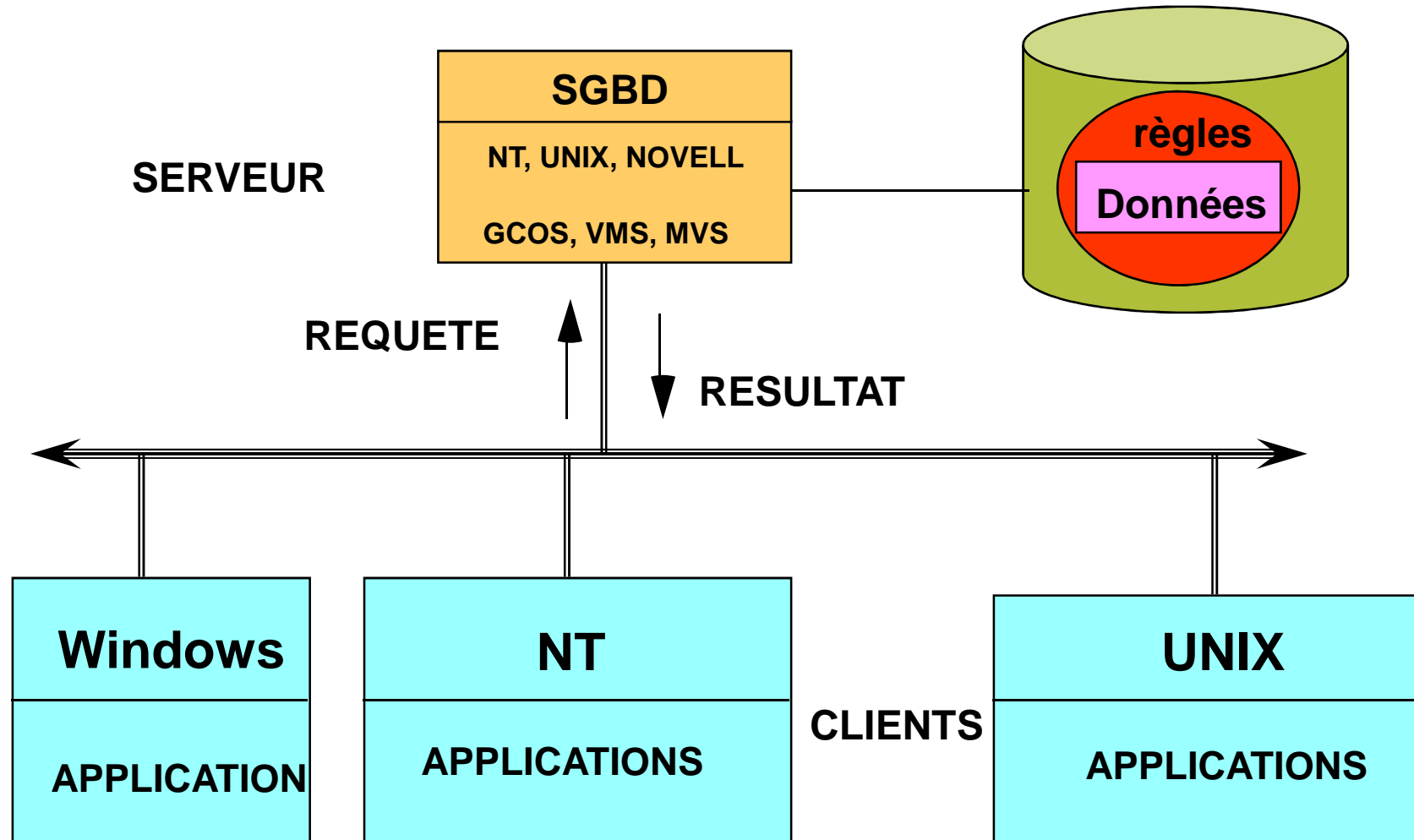
Pourquoi le C/S ?

- Évolution des besoins de l'entreprise
 - Augmentation de productivité, rapidité de réactivité souhaitée
 - Utilisation des micros assurant flexibilité et faibles coûts
 - Besoin de décisionnel et transactionnel sur gros volumes
- Évolution des technologies
 - Systèmes ouverts permettant l'usage de standards
 - Environnements de développement graphiques
 - Explosion de la puissance des micros et des serveurs (parallèles)

Pourquoi le C/S ?

- Solutions techniques séduisantes
 - ❑ Les données partagées enfin accessibles simplement
 - ❑ Mise en commun des services (règles de gestion, procédures)
 - ❑ Gestion de transactions et fiabilité au niveau du serveur

Architecture 1e génération



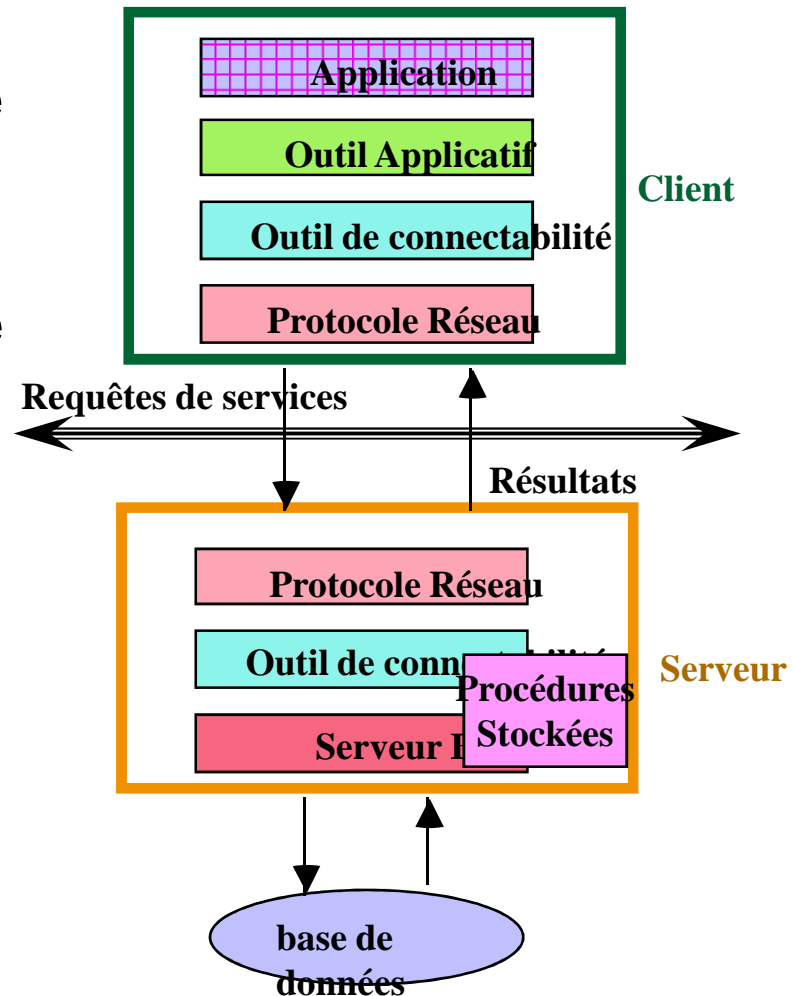
Le C/S de 2e génération

■ Procédure stockée

- ❑ Procédure accomplissant une fonction de service sur les données
- ❑ Exemple : Entrée ou sortie de stock

■ Architecture orientée services plutôt que requêtes

- ❑ Distribution des traitements
- ❑ Peut être automatisée



Le C/S de 2e génération

- Évolution et passage à l'échelle
 - Possibilité de serveurs multiples, avec redondances
 - Possibilité de données privées sur les clients

Intérêt du C/S de 2e génération

■ Réduction des transferts réseaux

- ❑ non nécessité de monter les données dans le client pour les modifier
- ❑ appel de services plus compact

■ Distribution automatique des applications

- ❑ développement sur le poste de travail
- ❑ partitionnement par tirer-déposer (drag & drop)

■ Simplification des outils de développement

- ❑ principe de la fenêtre unique
- ❑ modélisation uniforme des objets applicatifs
- ❑ invisibilité du modèle de données à l'extérieur du serveur

Faiblesses du client-serveur

- **Une mise en œuvre difficile**
 - ❑ nécessité de spécialistes réseaux, BD, PC
 - ❑ des outils hétérogènes et peu portables
 - ❑ les évolutions sont difficiles

Vers le C/S Universel (3e gén.)

- Intégration du Web et du client-serveur
 - ❑ navigateur à présentation standard pour le client
 - ❑ possibilité de petites applications (applets) sur le client
 - ❑ très grande portabilité (Réseau Privé Virtuel, Intranet, Internet)
- Architecture à 3 strates (3-tiered)
 - ❑ Base de données avec procédures stockées
 - ❑ Services applicatifs partagés
 - ❑ Présentation hypertexte multimédia avec applets

Vers le C/S Universel (3e généré.)

- Support de l'hypermédia
 - types de données variées et extensibles (texte, image, vidéo)
 - hypertexte et navigation entre documents et applications

Conclusion et bilan C/S

- Les SGBD fonctionnent tous en C/S
- Trois niveaux de fonctions distinguées :
 - données (SGBD)
 - application (L4G)
 - présentation (Web, Windows, Motif)