

Bases de données :
cours 1 – Introduction

Elena Leroux

Description du cours

- **Objectifs du cours :**
 - Acquérir les fondements théoriques et pratiques des bases de données.
 - Pouvoir modéliser, créer, interroger, mettre à jour, etc. une base de données.

Bibliographie



- **Ouvrages sur les bases de données :**

- Avi Silberschatz, Henry F. Korth and S. Sudarshan, « Database System Concepts » , 5th edition, McGraw-Hill Higher Education, 2006.
- Raghu Ramakrishnan and Johannes Gehrke « Database Management Systems », McGraw-Hill Higher Education, 2003.

Organisation du cours



PARTIE 1 : Modèle Entités-Associations (MEA) et Modèle Relationnel (MR)

- **Séance 1 :**
 - **Cours :** Introduction (1 heure)
 - **Cours :** MEA et MR (3 heures)
- **Séance 2 :**
 - **TD :** MEA+MR : exercices de difficulté faible (2 heures)
- **Séance 2 :**
 - **TD :** MEA+MR : exercices de difficulté moyenne (2 heures)
 -
- **Séance 2 :**
 - **TD :** MEA+MR : exercices de difficulté forte (2 heures)
- **Semaine 4 :**
 - **Examen** sur le Modèle Entités-Associations et le Modèle Relationnel (1 heure 45 minutes)

Organisation du cours



PARTIE 2 : SGBD Oracle et SQL

- **Séance 1 :**
 - **Cours :** SQL (4 heures)
- **Séance 2 :**
 - **TP :** SQL (2 heures)
- **Semaine 3 :**
 - **TP :** SQL (2 heures)
- **Semaine 4 :**
 - **TP :** SQL (2 heures)
- **Semaines 5 :**
 - **Examen** sur SQL (1 heure 45 minutes)

Évaluation

- **Note du cours :**
 - 50% examen (modélisation) + 50% examen (SQL)



Contact

- **E-mail** : elena.leroux@univ-ubs.fr
- **Bureau** : D006, bât. ENSIBS, Vannes

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Introduction

- Les entreprises gèrent des volumes de données très grands :
 - giga-, terra-, péta-octets,
 - données numériques, textuelles, multi-média (images, films,...)
- Il faut pouvoir facilement :
 - archiver les données sur des 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 (exemples : salaire, note en BD, etc.)
 - données composées (exemples : CV, résultats de l'année scolaire, etc.)
 - identifiant (exemples : NSS pour humains ou P26215 pour voitures, etc.)
- **Qu'est-ce qu'une BD ?**
 - Collection de données structurées reliées par des relations.
 - Interrogeable et modifiable par des langages de haut niveau.

Où les bases de données sont-elle utilisées ?

- Banques :
 - toutes les transactions
- Compagnies aériens :
 - réservations,
 - consultation des horaires
- Universités :
 - enregistrement des étudiants,
 - gestion de notes
- Magasins :
 - clients,
 - produits,
 - achats
- Ressources humaines :
 - les dossiers des employés,
 - les salaires,
 - déductions fiscales
- Etc.

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Définition d'une base de données

- **Définition :**

Une **base de données (BD)** est un ensemble structuré de données enregistrées sur un ordinateur et accessibles de façon sélective par plusieurs utilisateurs.

- Un logiciel qui permet d'interagir avec une base de données s'appelle un **Système de Gestion de Base de Données (SGBD)**.

Plan

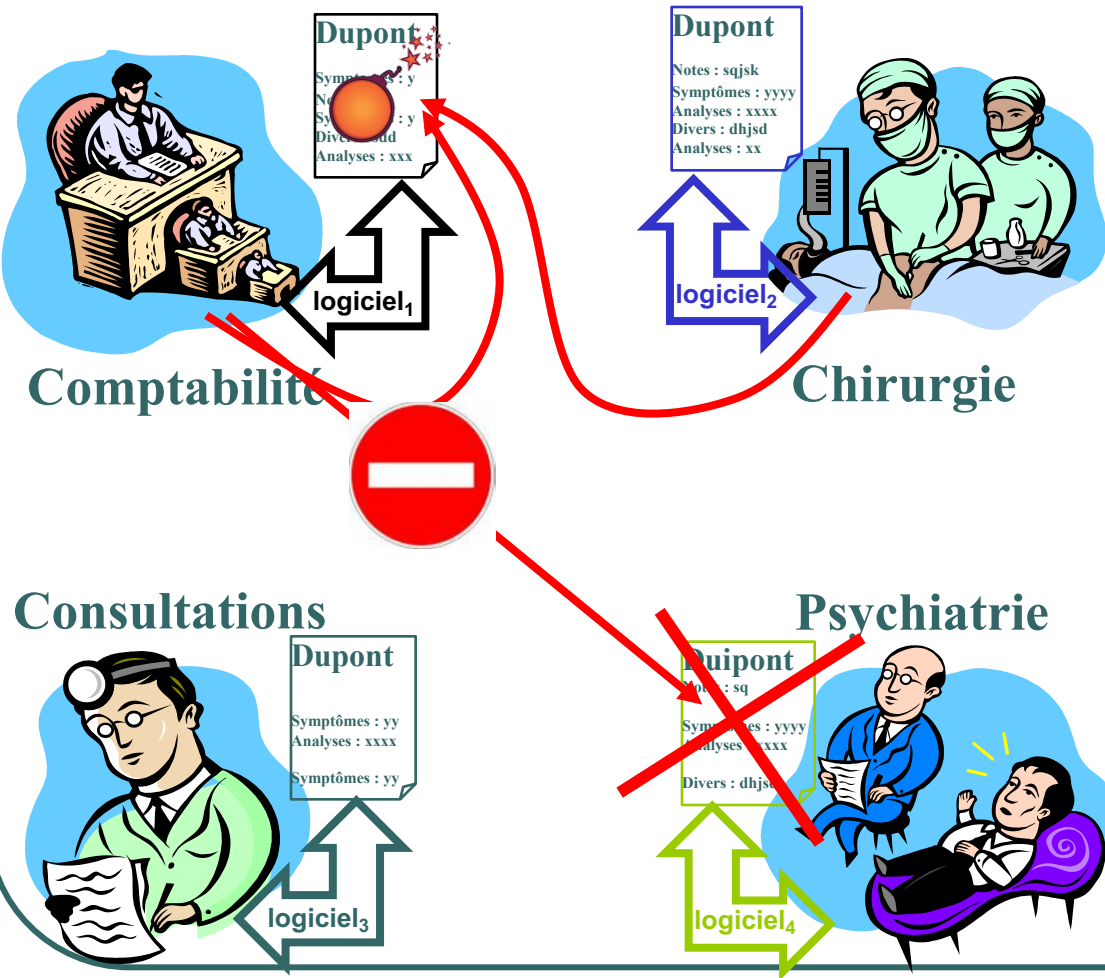


- Introduction
- Définition d'une base de données
- **Pourquoi doit-on utiliser des BD ?**
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Pourquoi doit-on utiliser des BD ?

: l'approche « système de fichiers »

Systèmes de fichiers



Caractéristiques

- Plusieurs applications
 - plusieurs formats
 - plusieurs langages
- Redondance de données
- Pas de facilité d'accès et d'interrogation

Problèmes

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

Pourquoi doit-on utiliser des BD ?

: l'approche "Bases de données"

- **La modélisation des données permet :**
 - d'éliminer la **redondance** de données :
 - de **centraliser** et d'**organiser** correctement les données,
 - d'avoir plusieurs niveaux de modélisation,
 - d'avoir divers outils de conception.
- **Logiciel «Système de Gestion de Bases de Données»**
 - **Factorisation** des modules de contrôle des applications :
 - Interrogation, cohérence, partage, gestion de pannes, etc.
 - **Administration** plus facile des données.

Pourquoi doit-on utiliser des SGBD ?

: résumé

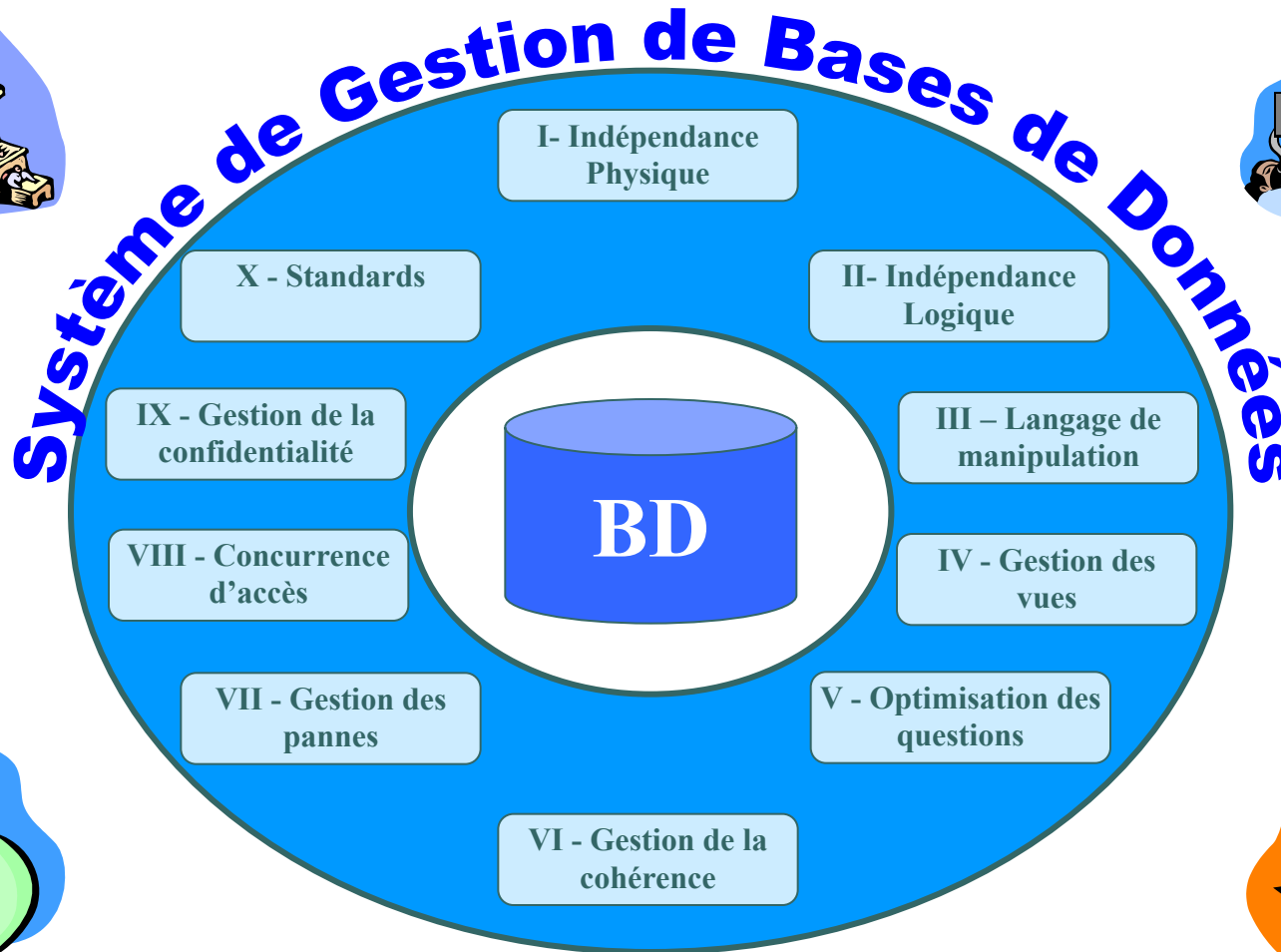
- Avant l'arrivée des bases de données (et encore dans beaucoup de cas aujourd'hui) :
 - Les informations d'un organisme étaient stockées dans divers fichiers.
 - Une même information (l'adresse d'un client par exemple) peut alors être enregistrée dans plusieurs fichiers disjoints.
 - Ceci occasionne des délais de mise à jour et peut amener les différentes applications à travailler sur des données contradictoires.
- Quand la gestion se fait avec une base de données centralisée (centralisée « logiquement » mais pas nécessairement physiquement si la base de données est répartie sur plusieurs sites) :
 - Chaque donnée n'est enregistrée qu'en un seul endroit de la base et il ne peut y avoir ce genre de problèmes.
 - Cette centralisation facilite donc le maintien de l'intégrité des données.

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- **Objectifs des SGBD**
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Objectifs des SGBD



Objectifs des SGBD :

I. Indépendance physique

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

Objectifs des SGBD :

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	...	30
2	Fluisédal	...	20
3	Mucomyst	...	230
...

Cabinet du Dr. Masse

Patients

Id-P	Nom	Prénom
1	Lebeau	Jacques
2	Troger	Zoé
...

Visiteurs

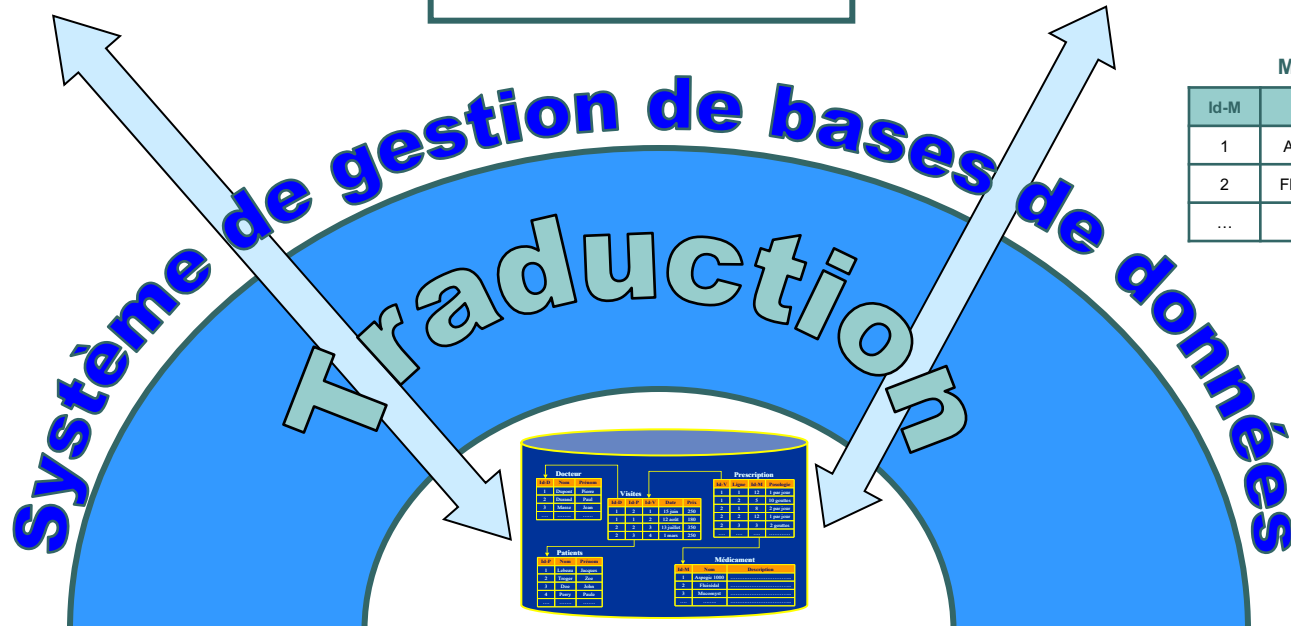
Id-D	Id-P	Id-V	Date	Prix
1	2	1	15 mai	25
2	3	4	1 mars	25

Prescription

Id-V	Ligne	Id-M	Posologie
1	1	12	1 par jour
1	2	5	10 gouttes
...

Médicament

Id-M	Nom	Description
1	Aspegic	...
2	Fluisédal	...
...



Objectifs des SGBD :

II. Indépendance logique – avantages

- 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 relations, renommage de champs.
- Possibilité **d'intégrer des applications existantes** sans modifier les autres.
- Possibilité de limiter les conséquences du partage : **données confidentielles**.

Objectifs des SGBD :

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** : **S**tructured **Q**uery **L**anguage
- Sémantique :
 - Logique du 1er ordre
- Syntaxe (aperçu !) :
 - **SELECT** <structure des résultats>
 - **FROM** <relations>
 - **WHERE** <conditions>

Objectifs des SGBD :

IV. Des vues multiples des données

- Les vues permettent d'implémenter l'indépendance logique en permettant de créer des **relations virtuelles**.
- 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
 - ...

Objectifs des SGBD :

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.

Objectifs des SGBD :

VI. Intégrité logique

- **Objectif :**

- Détecter les mises à jour erronées

- **Contrôle sur les données élémentaires :**

- Contrôle de types : par exemple, nom doit être alphabétique.
- Contrôle de valeurs : par exemple, salaire mensuel doit être entre 2500 et 5000 euros

- **Contrôle sur les relations entre les données :**

- Relations entre données élémentaires :
 - Exemple : Prix de vente > Prix d'achat
- Relations entre objets :
 - Exemple : Un électeur doit être inscrit sur une seule liste électorale.

Objectifs des SGBD :

VI. Intégrité logique – contraintes d'intégrités

- **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.

Objectifs des SGBD :

VII. Intégrité physique

- **Motivations : tolérance aux fautes**

- Transaction Failure : contraintes d'intégrité, annulation.
- System Failure : panne de courant, crash serveur, etc.
- Media Failure : perte du disque,
- Communication Failure : défaillance du réseau.

- **Objectifs :**

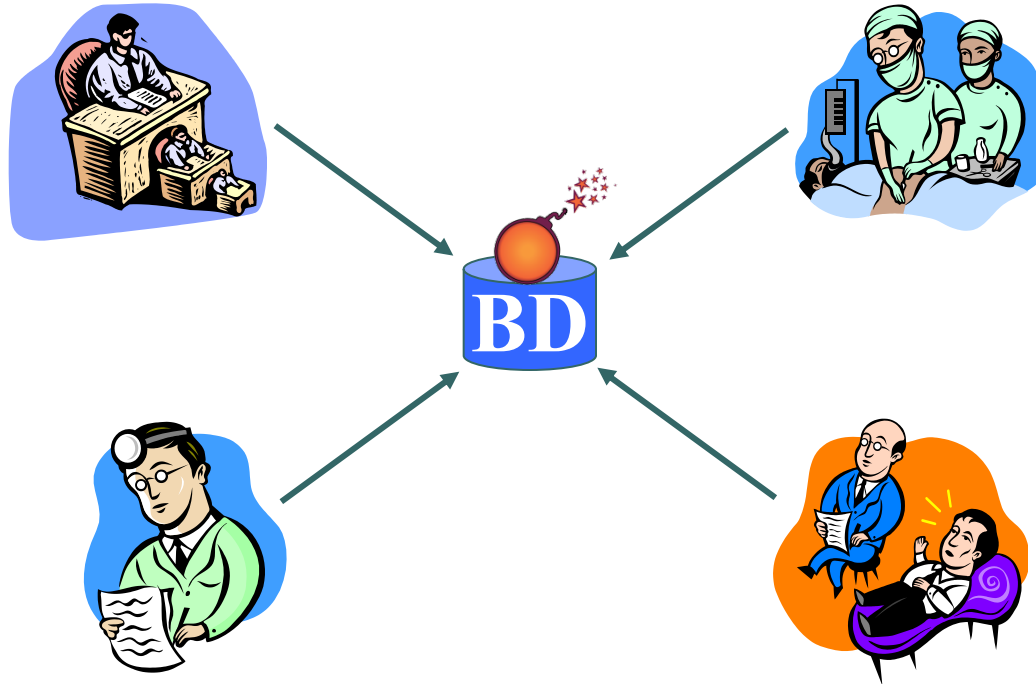
- Assurer l'**atomicité** des transactions.
- Garantir la **durabilité** des effets des transactions commises.

- **Moyens :**

- Journalisation : mémorisation des **états successifs** des données.
- Mécanismes de reprise.

Objectifs des SGBD :

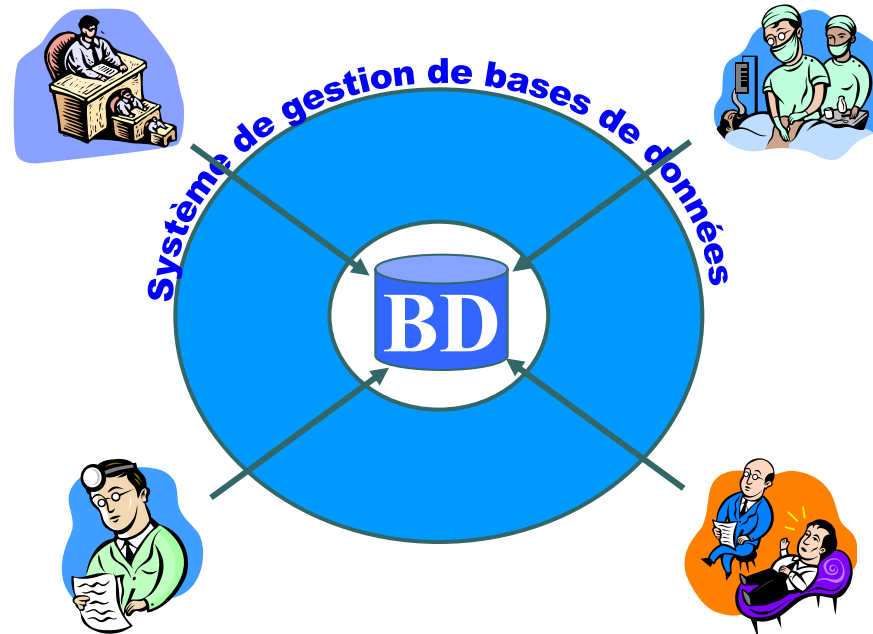
VIII. - Partage des données – problème



L'accès concurrent aux mêmes données
provoque un conflits d'accès !

Objectifs des SGBD :

VIII. - Partage des données



Le SGBD gère les accès concurrents :

- chacun à l'*impression* d'être seul (« isolation »)
- La cohérence de la base de données est conservée

Objectifs des SGBD :

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 (relations, vues, procédures, etc.) de la base.

Objectifs des SGBD :

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,
 - ...

Fonctions des SGBD : résumé

- Un SGBD doit permettre de :
 - décrire les données qui seront stockées,
 - manipuler ces données (ajouter, modifier, supprimer des informations),
 - obtenir des renseignements à partir de ces données (sélectionner, trier, calculer, agréger, etc.),
 - définir des contraintes d'intégrités sur les données (contraintes de domaines, d'existence, etc.),
 - définir des protections d'accès (mots de passe, autorisations, etc.),
 - résoudre les problèmes d'accès multiples aux données (blocages, inter-blocages),
 - prévoir des procédures de reprise en cas d'incident (sauvegardes, journaux, etc.),
 - écrire des applications indépendantes de l'implantation physique des données (codage, ordre dans lequel les données sont enregistrées, support d'enregistrement, etc.).

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- **Architectures logiques des SGBD**
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Architectures logiques des SGBD

- **Architectures logiques :**
 - historiquement les premiers SGBD étaient de **type hiérarchique**,
 - puis sont apparus les SGBD de **type réseau**.
 - actuellement, la plupart des nouveaux systèmes sont de **type relationnel**.
 - des SGBD de **type « objet »** commencent à apparaître sur le marché et ils remplaceront vraisemblablement une partie des SGBD relationnels.
- La différence essentielle entre ces types de SGBD réside dans les modèles sur lesquels ils s'appuient pour représenter les données.

Architectures logiques de SGBD : hiérarchique

- Les données sont représentées sous forme d'une **structure arborescente** d'enregistrements.
- Cette structure est conçue avec des **pointeurs** et détermine le chemin d'accès aux données.

Architectures logiques de SGBD : réseau

- La structure des données peut être visualisée sous la forme d'un **graphe** quelconque.
- Comme pour le modèle hiérarchique, la structure est conçue avec des **pointeurs** et détermine le chemin d'accès aux données.
- Pour les modèles hiérarchiques et réseau :
 - les programmes ne sont pas indépendants de la structure logique de la base et du chemin d'accès aux données :
 - ils doivent décrire comment retrouver les données (on parle de navigation dans la base) et si, par exemple, on enlève un index, tous les programmes qui l'utilisaient doivent être réécrits.
 - De plus le langage de travail est complexe.

Architectures logiques de SGBD : relationnel (1/2)

- Il est fondé sur la **théorie mathématique des relations**.
- Il conduit à une représentation très simple des données sous forme de **tables** constituées de lignes et de colonnes.
- Il **n'y a plus de pointeurs** qui figeaient la structure de la base.
- La souplesse apportée par cette représentation et les études théoriques appuyées sur la théorie mathématique des relations ont permis le développement de **langages puissants non procéduraux** pour manipuler les données.

Architectures logiques de SGBD : relationnel (2/2)

- Dans ces langages, l'utilisateur ou le programmeur indique quelles informations il veut obtenir et c'est le SGBD qui trouve la manière d'arriver au résultat.
- Le programme ou l'utilisateur n'a plus à naviguer dans la base pour retrouver ses données.
- Ces langages peuvent être utilisés par des non-informaticiens et permettent l'écriture de programmes indépendants de la structure logique et physique des données.
- Le langage **SQL** est un standard parmi tous ces langages.

Architectures logiques de SGBD : objet

- Les données sont représentées sous forme d'**objets** au sens donné par les langages orientés objet :
 - pour simplifier, les données (au sens habituel) sont enregistrées avec les procédures et fonctions qui permettent de les manipuler.
- Les SGBD orientés objet (SGBDOO) supportent aussi la notion d'**héritage entre classes d'objets**.

Plan

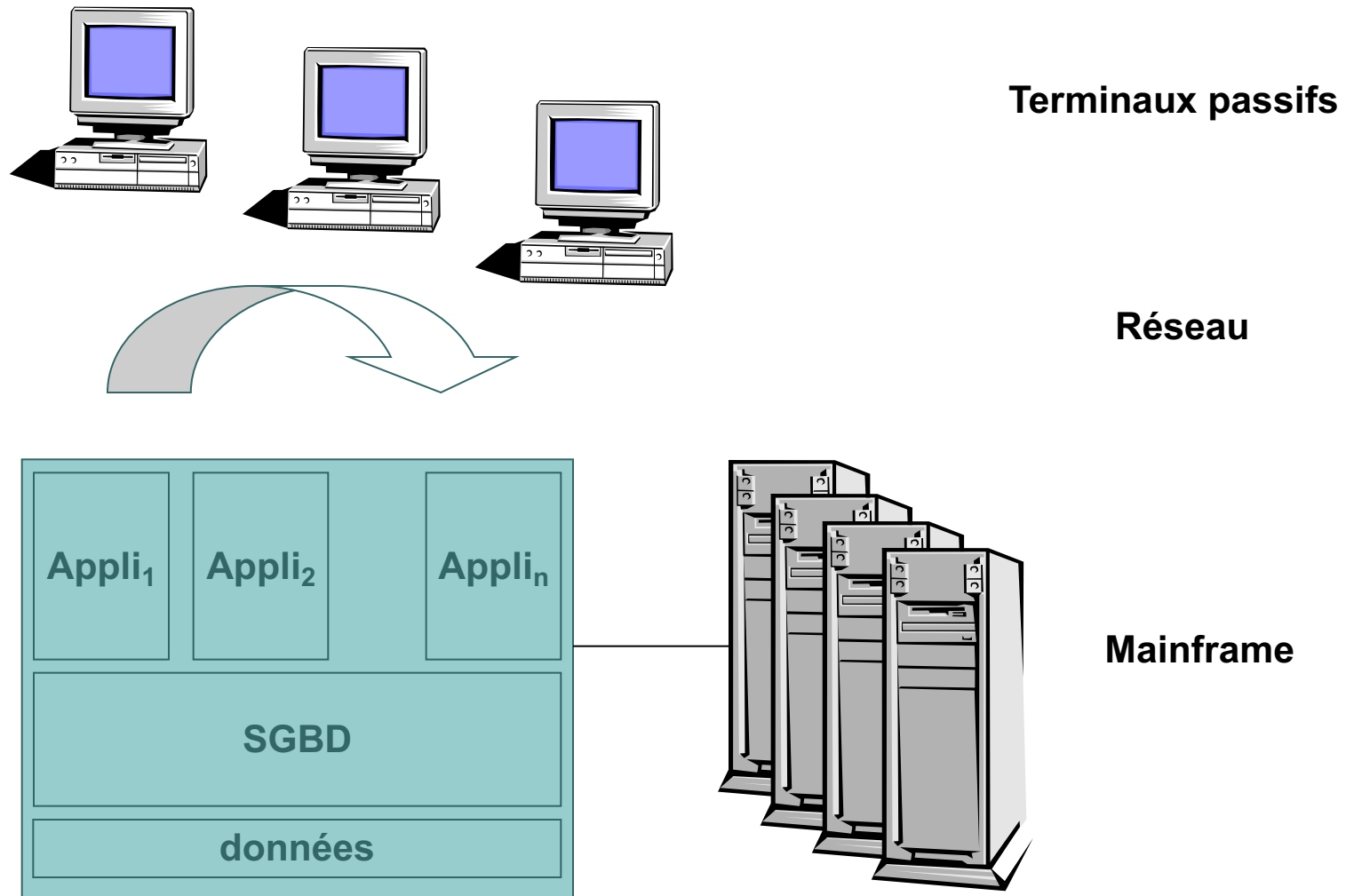


- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- **Architectures physiques des SGBD**
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

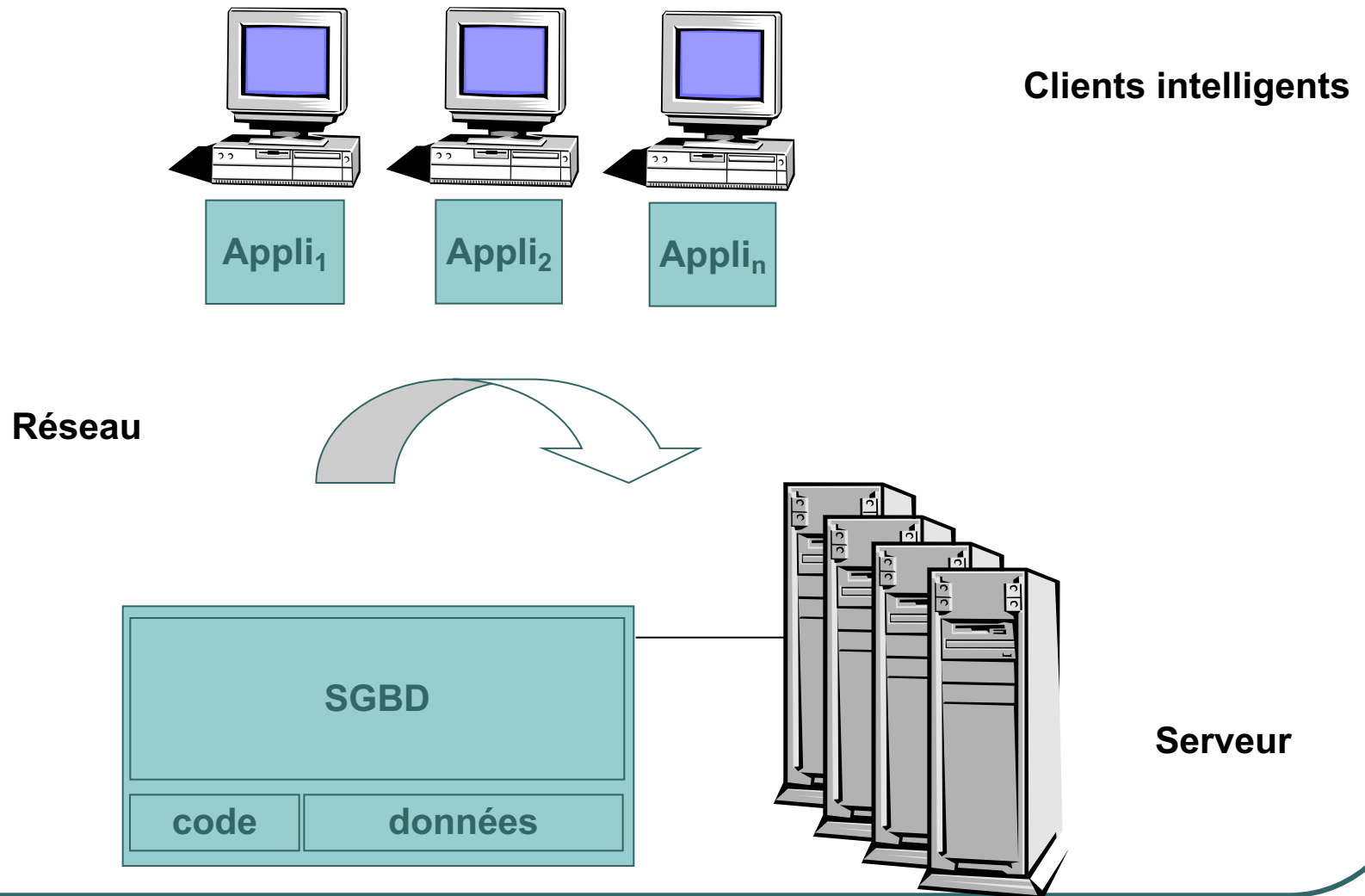
Architectures physiques des SGBD

- Les architectures physiques de SGBD sont très liées au mode de répartition.
- On distingue les architecture physiques suivantes :
 - centralisée,
 - client - serveur,
 - client - multi serveurs,
 - répartie,
 - hétérogène,
 - mobile.

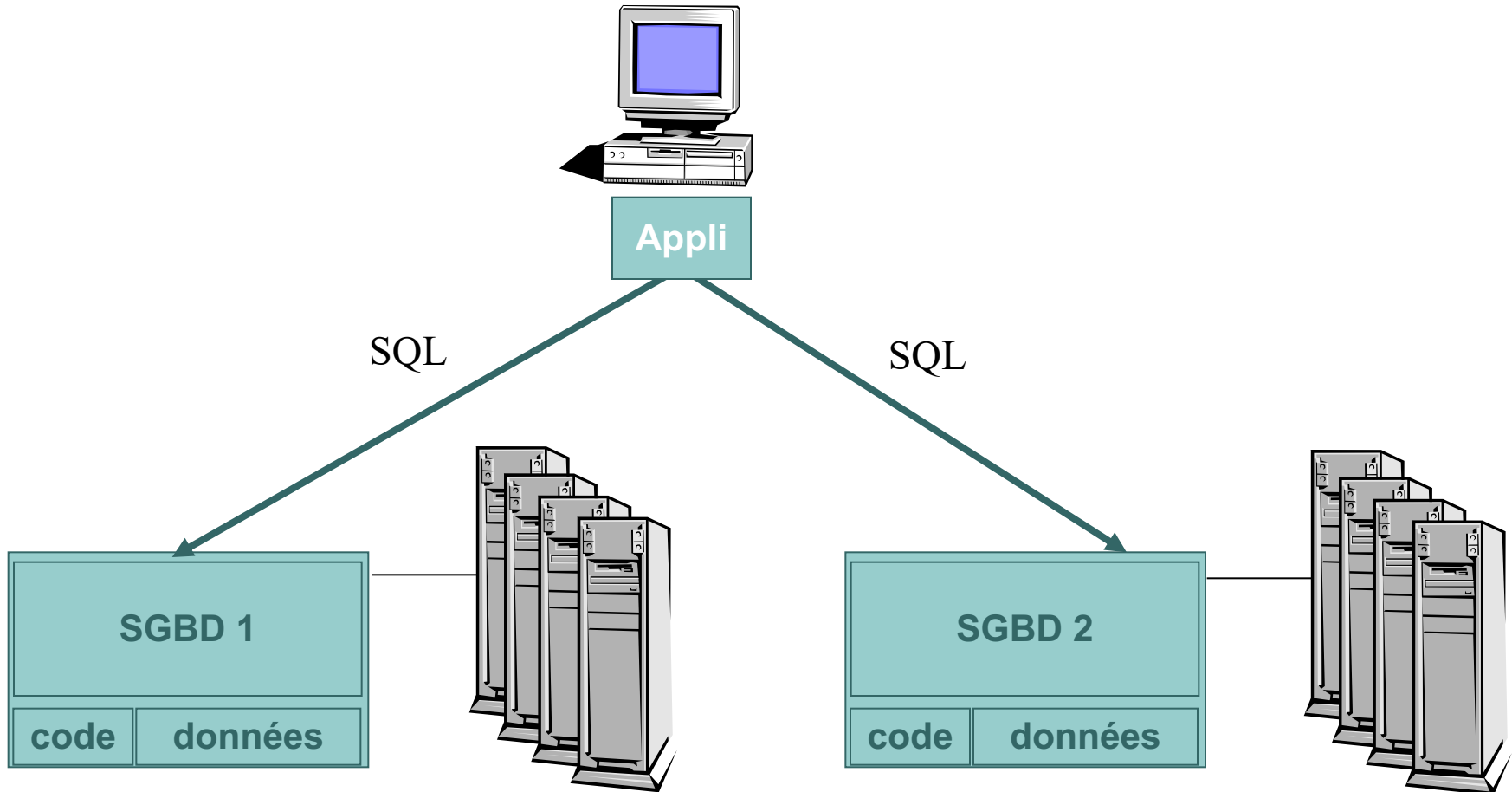
Architectures physiques des SGBD : centralisée



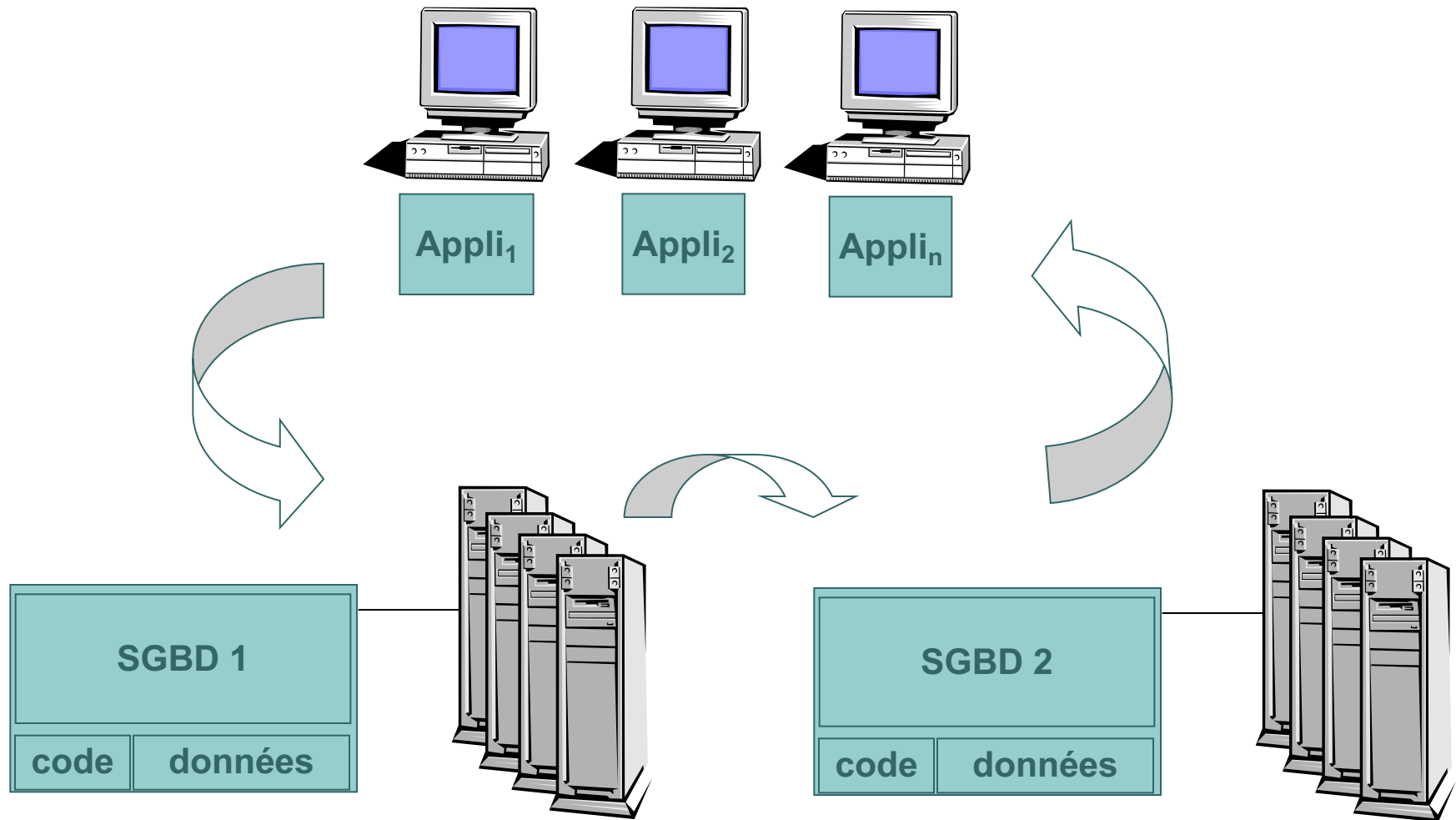
Architectures physiques des SGBD : client-serveur



Architectures physiques des SGBD : client – multi serveurs



Architectures physiques des SGBD : répartie



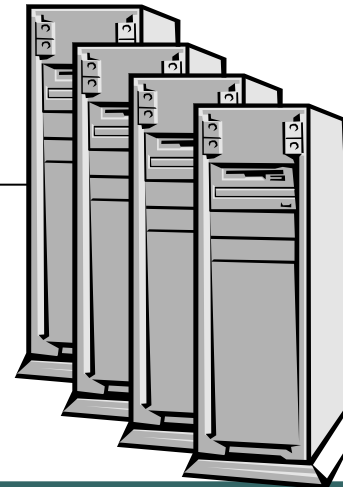
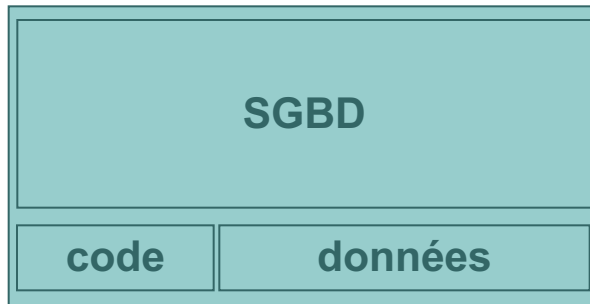
Architectures physiques des SGBD : mobile



Clients intelligents
mobiles

Données répliquées
et/ou personnelles

Réseau sans fil



Serveur

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- **Types d'utilisateurs d'une BD**
- Niveaux de description des BD
- Un peu d'histoire

Types d'utilisateurs d'une BD : administrateur

- L'**administrateur** de la base est chargé du **contrôle de la base de données**.
- En particulier, il est chargé de :
 - permettre l'**accès aux données** aux applications ou individus qui y ont droit,
 - conserver de **bonnes performances d'accès** à ces données,
 - réaliser des **sauvegardes** et
 - réaliser des **procédures de reprise** après les pannes.

Types d'utilisateurs d'une BD : programmeur d'applications

- Le **programmeur d'applications** utilise la base de données pour **construire ses applications**.
 - Il a le droit de créer de nouvelles tables et les structures associées (vues, index, cluster, etc.).
 - Il définit avec l'administrateur de la base les droits qui seront accordés aux utilisateurs des applications qu'il développe.

Types d'utilisateurs d'une BD : utilisateur final

- **L'utilisateur final** n'a accès qu'aux données qui lui sont utiles.
 - L'administrateur de la base peut lui accorder certains droits :
 - consultations, modification, suppression des données.
 - En général il n'a pas le droit de créer de nouvelles tables ni d'ajouter ou d'enlever des index.
 - Il interroge la base de données à l'aide de requêtes.
 - **Exemples :**
 - Les personnes qui accèdent à la base de données par Internet.
 - Les employés de banque.
 - etc.

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- **Niveaux de description des BD**
- Un peu d'histoire

Niveaux de description des BD (1/2)

- La norme ANSI/SPARC définit 3 niveaux pour décrire l'architecture d'une base de données :
 - le **niveau externe** correspond aux différentes façons dont les utilisateurs peuvent voir les données de la base. La représentation de la base est donc composée de plusieurs schémas externes.
 - le **niveau conceptuel** correspond à une vision globale de la base. Il n'y a qu'un seul schéma conceptuel. Les données sont décrites ainsi que les contraintes d'intégrité liées aux données.
 - le **niveau interne** correspond à la manière dont la base est implantée sur les ordinateurs (description des enregistrements contenant les données, des index, etc.).

Niveaux de description des BD (2/2)

- Ce découpage facilite la maintenance des applications :
 - il permet de dissocier autant que possible les applications des contraintes liées au matériel ou au système d'exploitation (seul le niveau interne en dépend),
 - la séparation entre les niveaux externe et conceptuel permet de conserver les mêmes applications même si le schéma conceptuel est modifié ou enrichit.
- Ce cours concerne plus particulièrement les niveaux **externes** et **conceptuels**.

Plan



- Introduction
- Définition d'une base de données
- Pourquoi doit-on utiliser des BD ?
- Objectifs des SGBD
- Architectures logiques des SGBD
- Architectures physiques des SGBD
- Types d'utilisateurs d'une BD
- Niveaux de description des BD
- Un peu d'histoire

Un peu d'histoire (1/2)

- **Les années 50 et le début des années 60 :**
 - Les données sont stockées sur les bandes magnétiques :
 - accès séquentiel aux données
 - Les cartes perforées sont utilisées pour les données d'entrée.
- **La fin des années 60 et les années 70 :**
 - L'arrivée des disques durs permet un accès direct aux données.
 - Les architectures de type hiérarchique ou réseau sont utilisées pour organiser les données et y accéder.
 - Ted Codd définit le modèle relationnel de données et gagne le prix de ACM Turing pour ces travaux.
 - IBM commence le développement du prototype « System R »
 - UC Berkeley commence le développement du prototype « Ingres ».

Un peu d'histoire (2/2)

- **Les années 80 :**

- La recherche sur des prototypes relationnels pénètre dans l'industrie :
 - Le langage SQL devient un standard industriel.
- Utilisation d'architectures parallèles et distribuées pour l'organisation physique des données.
- Apparition de bases de données objet.

- **Les années 90 :**

- Augmentation du nombre d'applications utilisant des bases de données.
- Augmentation significative en taille des bases de données.
- Émergence du commerce Web.

- **Les années 2000 :**

- Apparition des standard XML et XQuery.
- Automatisation de l'administration des bases de données.