


Introduction aux Bases de données



Principes, techniques
et outils

Plan

- Chapitre I : Concepts de base.
- Chapitre II : Conception des bases de données.
- Chapitre III : Modèle relationnel.
- Chapitre IV : Langage de requêtes SQL.
- Chapitre V : Protection de données.
- Chapitre VI : Outils de développement.
- Chapitre VII : Méthode de MERISE

Plan(I)

- **Introduction**
 - **Historique**
 - **Définition d'une base de données**
 - **Définition d'un Système de Gestion des Bases de Données (S.G.B.D)**
- **Modèles de représentation de données**
 - **Modèle réseau**
 - **Modèle hiérarchique**
 - **Modèle relationnel**
 - **Modèle orienté objet**

Plan(I)

- **Systemes de gestion des bases de données**
 - **Définition et rôle d'un SGBD**
 - **Structure générale d'un SGBD**
 - **Objectifs d'un SGBD**
 - **Fonctionnalités d'un SGBD**
 - **Types des SGBDs**
 - **Architecture fonctionnelle d'un SGBD**
 - **Gestion de fichiers (indexation et Hachage)**
 - **Différentes architectures des SGBDs.**

Plan(II)

- **Conception des bases de données**
 - **Généralités**
 - **Dictionnaire de données**
 - **Dépendances fonctionnelles**
 - **Définition**
 - **Dépendances fonctionnelles triviales et non triviales**
 - **Fermeture et transitivité des dépendances**

Plan(II)

- **Formes de normalisation**
 - **Normalisation de dictionnaire de données**
 - **Normalisation du premier niveau**
 - **Première forme normale**
 - **Deuxième forme normale**
 - **Troisième forme normale**
 - **Forme normale de Boyce/Codd**
 - **Cinquième forme normale**
 - **Sixième forme normale**

Plan(II)

- **Modèle conceptuel et logique de données**
 - **Les tables**
 - **Les propriétés**
 - **Les relations entre les tables**
 - **Les cardinalités**
 - **Règles de passage de MCD au MLD**
- **Études de cas**

Plan(III)

- **Modèle relationnel**
 - **Objectifs du modèle relationnel**
 - **Principe du modèle relationnel**
 - **Structure de la base de données**
 - **Règles d'intégrité structurelles**
 - **Règles d'intégrité non structurelles**
 - **Règles d'intégrité comportementales**

Plan(III)

- **Algèbre relationnel**
 - **Opérations ensemblistes**
 - ✓ **Union,**
 - ✓ **Produit cartésien,**
 - ✓ **Intersection,**
 - ✓ **Différence.**
 - **Opérations spécifiques**
 - ✓ **Jointure,**
 - ✓ **Semi-jointure**

Plan(III)

- **Algèbre relationnel**
 - **Autres opérations**
 - ✓ **Division,**
 - ✓ **Éclatement,**
 - ✓ **Complément.**

Plan(IV)

- **Les langages de requêtes :**
 - **Définition d'un langage de requête.**
 - **Langage de définition de données.**
 - **Langage de manipulation de données.**
 - **Langage de contrôle de données.**

Plan(V)

- **Protection de données.**
 - **Vue.**
 - **Intégrité de données.**
 - **SGBDs Actifs (Réflexe // Trigger).**
 - **Transactions et Reprises aux pannes.**
 - **Gestion de la concurrence.**
 - **Sécurité données.**

Plan(VI)

- **Les outils de développement :**
 - **Informix-SQL,**
 - **SGBD Access,**
 - **Visual Basic,**
 - **Serveur Apache et EasyPHP.**
- **Mini-Projets.**

Plan(VII)

- **Système d'information (MERISE)**
 - **Approche systématique**
 - **Les étapes de Merise**
 - **Modèle Conceptuel de Données (M.C.D)**
 - **Modèle Conceptuel de Traitement (M.C.T)**
 - **Modèle Logique de Données (M.L.D)**
 - **Modèle Organisationnel de Traitement (M.O.T)**
 - **Test et Validation**

Historique

Plus de 40 ans d'histoire

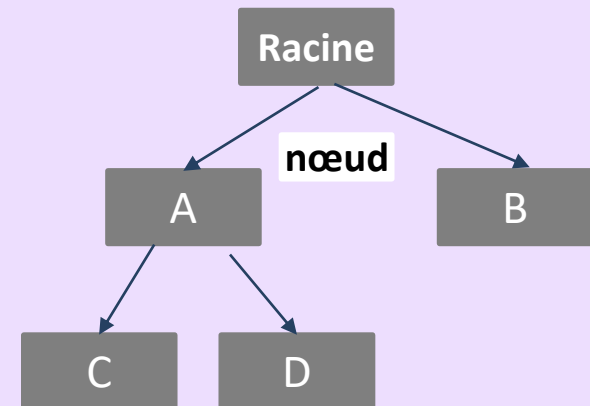
- Début des années 60 :
 - Liste de fichiers reliés par des pointeurs
 - Articles enchaînés accessibles par des clés
 - L'accès aux données se fait par le parcours de la chaîne d'articles (APOLLO/1965).

Historique

Première Génération (*Modèle Hiérarchique*)

Une **base de données hiérarchique** est une base de données dont le système de gestion lie les enregistrements dans une structure arborescente où chaque enregistrement n'a qu'un seul possesseur.

Base de de données hiérarchiques :
ADABAS, IMS (Programme APollo), ...

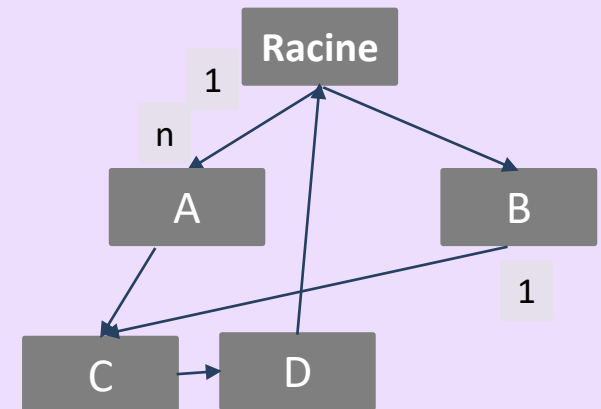


Historique

Première Génération (*Modèle Réseau*)

La **base de données réseau** est une extension de la base de données hiérarchique en donnant la possibilité d'établir des liaisons de type n-m et en définissant des associations entre tous les types d'enregistrements.

Base de de données réseaux :
TOTAL, IDMS, IDS, ...



Historique

Première Génération

(Modèle Réseau & Hiérarchique)

- **Fin des Années 60 :**
 - Recommandation de CODASYL (Conference on Data Systems Languages)
 - Séparation de la description des données de la manipulation de celles-ci
 - Modèle de données définit par un ensemble de fichiers reliés par des pointeurs
 - L'avènement des langages de navigation.

Historique

Deuxième Génération (*Modèle Relationnel*)

■ **Début des Années 70 :**

Les bases de données relationnelles présentent les données sous forme de relations entre domaines de valeurs, simplement représentées par des entités appelées tables. Elles utilisent les principes de l'algèbre relationnel et procèdent aux recherches et mises à jour à l'aide d'un langage non procédural standardisé appelé SQL (*Structured Query Language*).

Historique

Deuxième Génération (*Modèle Relationnel*)

- **Début des Années 70 :**
 - Facilite l'accès aux données pour les utilisateurs.
 - Améliore les fonctions de modèle d'accès de la 1ère génération.
 - Modèle de données sous forme de relations entre domaines de vecteurs.
 - Avènement des langages non procéduraux logique du 1^{er} ordre.

Historique

Troisième Génération (*Modèle Objet*)

- **Début des Années 80 :**
 - Support des modèles de données plus riches
 - Architectures mieux réparties (concurrents)
 - Réutilisation de l'existant
 - Intégration du support de règles dérivées de la modélisation logique des données
 - Structuration des programmes et des données en classes
 - Modèle de données constitué d'un ensemble d'objets (instances de classes)

Généralités

Qu'est-ce qu'une base de données ...?

- Un ensemble de données modélisant des objets d'une partie du monde réel ;
- Servant de support à une application informatique ;
- Non indépendantes ;
- Interrogeantes par le contenu ;
- Organisation cohérente de données permanentes et accessibles par plusieurs utilisateurs.

Généralités

Résumé

- C'est un ensemble structuré de données enregistré sur un supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en temps opportun.
- C'est un ensemble de données non redondantes logiquement liées et placées en ensemble, et qui permettent d'être utilisées par des applications différentes via des langages d'interrogation.

Généralités

Modèles de représentation physique de données

- » *Réseau*
- » *Hiérarchique*
- » *Relationnel*
- » *Orienté objet*

SGBD

C 'est quoi un S.G.B.D ?

Un ensemble de logiciels systèmes permettant

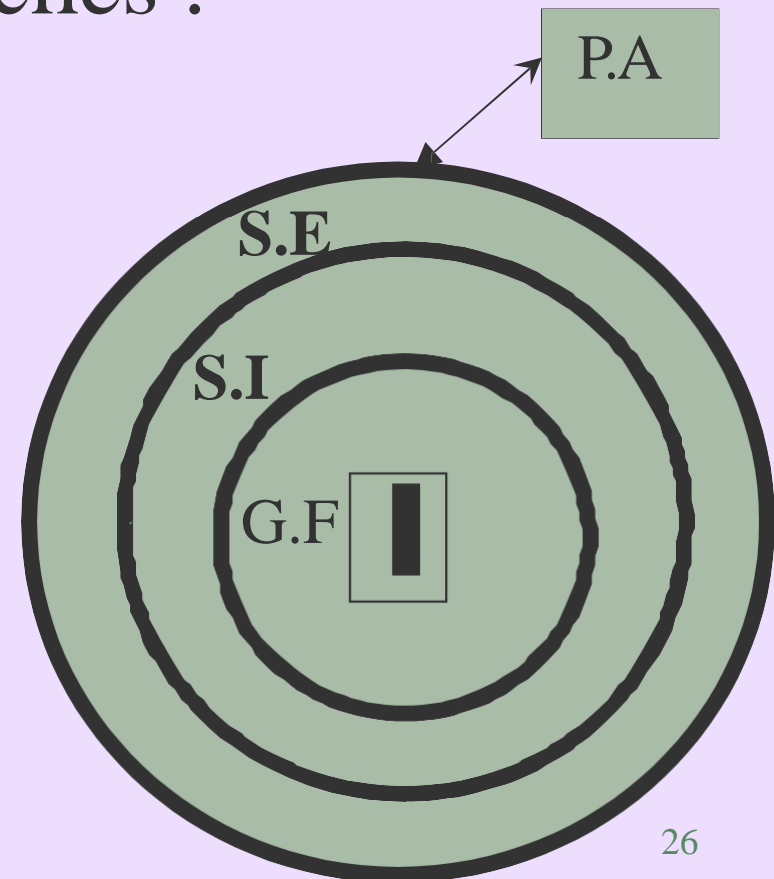
- » **Insérer**
- » **Modifier**
- » **Chercher**
- » **Supprimer**
- » **Partager**
- » **Protéger**

Efficacement, des données spécifiques dans une grande masse d'information.

SGBD

Architecture de base

- Se compose de trois couches :
 - SGBD Externe
 - SGBD Interne
 - Gestionnaire de fichier



Modélisation de données

- **Niveaux d'abstraction**
 - **Un des objectifs majeurs des SGBDs est d'assurer une abstraction de données stockées. Pour cela trois niveaux de description ont été distingués par le groupe de normalisation ANSI/X3/SPARC :**
 - ✓ **Schéma conceptuel**
 - ✓ **Correspond à la structure canonique de données représentant la vue intégrée de tous les utilisateurs.**
 - ✓ **Schéma interne**
 - ✓ **Correspond à la structure de stockage supportant les données.**
 - ✓ **Schéma externe**
 - ✓ **Une description des données propre à chaque groupe d'utilisateurs**

SGBD Conceptuel

- **C'est le niveau central d'un SGBD. Il correspond à la structure canonique de données qui existent dans l'organisation :**
 - **Structure sémantique inhérente sans souci d'implémentation en machine.**
 - **Vue intégrée de tous les utilisateurs.**

SGBD Conceptuel

- **Présente la définition de base que modélise les données:**
 1. **Les types de données élémentaires, qui définissent les propriétés élémentaires des objets de l'entreprise.**
 2. **Les types de données composés, qui permettent de regrouper les attributs afin de décrire les objets du monde réel ou des relations entre les objets.**
 3. **Les types de données composés, qui permettent de regrouper les attributs afin de décrire les associations du monde réel.**
 4. **Les règles que devront suivre les données au cours de leur vie dans l'organisation.**

SGBD Externe

- **Cette couche consiste à**
 - **La mise en forme et la présentation des données aux programmes d'application selon le besoin des utilisateurs interactifs.**
 - **Exprime les critères de recherche des utilisateurs à l'aide des langages basés sur des procédures de recherche progressives ou sur des assertions logiques.**
 - **Assure l'analyse et l'interprétation des requêtes d'utilisateurs en primitives internes.**
 - **Assure l'indépendance entre la structure logique et les programmes qui manipulent celle-ci.**
 - **Permet d'assurer une certaine sécurité de données**

SGBD interne

- **Cette couche se base sur le modèle de données utilisé. Elle consiste à :**
 - **La gestion des données stockées dans les fichiers.**
 - **Assemblage des données en objets du module**
 - **Placement des objets dans les fichiers**
 - **Gestion de liens entre les objets**
 - **Gestion des structures permettant d'accélérer les accès aux objets (Objets enchaînés via des listes).**
 - **Présentation des objets demandés par l'utilisateur via S.G.B.D externe.**

Gestionnaire de fichiers

- Cette couche consiste à la gestion des réceptacles de données sur les mémoires secondaires. Elle fournit aux couches supérieures, des mémoires secondaires idéales adressables par objets. Pour l'accès deux mécanismes possibles :
 - Mécanisme de Hachage !!!
 - Mécanisme indexation !!!

Gestionnaire de fichiers

- **Architecture de base:**

Elle se compose de trois couches principales :

- 1. Méthodes d'accès;**
- 2. Analyseur;**
- 1. Modules d'entrées sorties.**

Gestionnaire de fichiers

- **Fonctions de base**
 - **Manipulation des fichiers**
 - **Adressage relatif**
 - **Allocation de l'espace mémoires**
 - **Localisation de fichiers sur les volumes**
 - **Classification de fichiers en hiérarchie**

Gestionnaire de fichiers

- **Manipulation des fichiers.**
 - **Opérations de base qui permettent l'accès au noyau de Gestionnaire de Fichiers pour la manipulation de ces derniers :**
 - » **Créer.**
 - » **Lire.**
 - » **Écrire.**
 - » **Détruire.**
 - » **Fermer.**

Gestionnaire de fichiers

■ **Adressage relatif**

① **Problématiques**

- **Discontinuité des fichiers sur la mémoire secondaire.**
- **Repérage indépendant de la localisation des fichiers sur la mémoire secondaire.**
- **Contrôle de débordement de la taille des fichiers.**
- **Minimiser le nombre des opérations d'entrées sorties.**
- **Répartition des enregistrements sur l'espace disque.**
- **Minimiser l'espace mémoire non utile (petite taille).**

Gestionnaire de fichiers

■ Adressage relatif

©Solution

- Division des fichiers en plusieurs pages.
- Taille de la page fixée pour qu'elle contienne plusieurs enregistrements (~4k).
- Chaque adresse relative se compose d'un numéro suivi d'un numéro d'octet dans la page.
- Possibilité d'accès direct ou bien séquentiel à une page.
- Utilisation des algorithmes de détection et annulation de blocage (dans les accès concurrents).
- Opérations de comptage

Gestionnaire de fichiers

Allocation d'espace mémoire, c'est la réservation des zones mémoires continues pour les fichiers.

- ① **Région** : Ensembles des zones mémoires secondaires (pistes) adjacentes allouées en une seule fois à un fichier.
- ① **Granule** : Unité minimale de la mémoire allouable à un fichier.
 - » Allocation statique (Table de bits)
 - » Allocation dynamique (Liste des Zones libres)

Gestionnaire de fichiers

■ Localisation des fichiers sur les volumes.

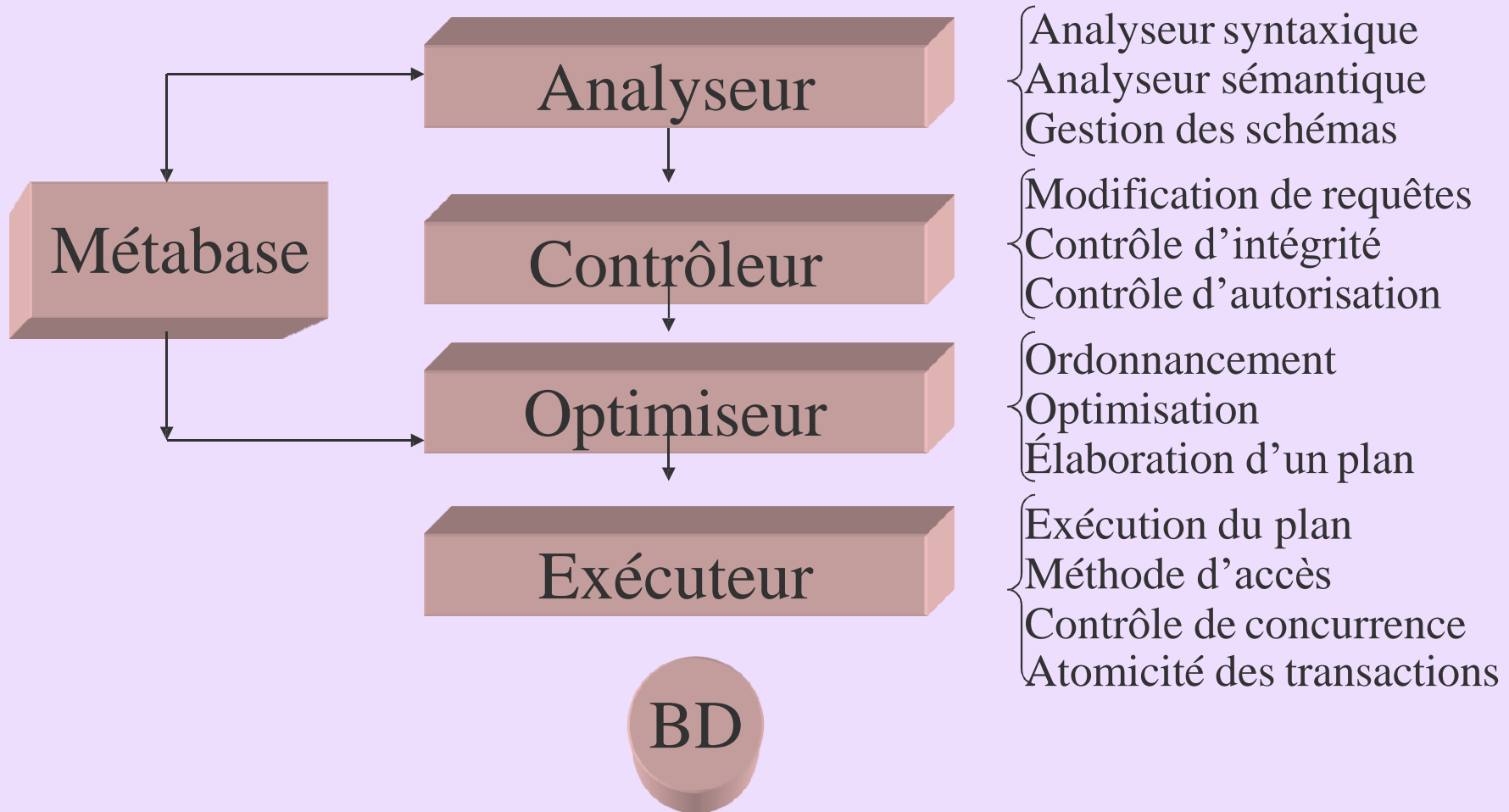
- **Label de volume** : premier secteur d'un volume permettant d'identifier ce volume et contenant son numéro.
- **Descripteur du fichier** : Ensemble des informations permettant de retrouver les caractéristiques d'un fichier, contenant son nom, sa localisation sur disque, sa taille...
- **Catalogue** : Table située sur un volume et contenant les descripteurs des fichiers du volume.
- **Catalogue hiérarchisé** : Catalogue constitué d'une hiérarchie de fichiers, chaque fichier contenant les descripteurs des fichiers immédiatement inférieurs dans la hiérarchie.

Gestionnaire de fichiers

- **Stratégie d'allocation de la mémoire secondaire**
 - **Objectifs d'une stratégie**
 - **Minimiser le nombre de régions à allouer à un fichier.**
 - **Minimiser le déplacement des bras de disques lors des lectures en séquentiel.**
 - **Minimiser le nombre de descripteurs de régions associés à un fichier.**
 - **Minimiser la distance qui sépare les régions successives d'un fichier.**

Différents types d'architectures de SGBD

Architecture fonctionnelle



Différents types d'architectures de SGBD



- Définition : c'est un dictionnaire de données organisé comme une base de données qui décrit d'autres bases et contient entre autres :
 - ✓ Entités
 - ✓ Relations
 - ✓ Vues
 - ✓ Règles sur la vie de données
 - ✓ Indexes
 - ✓ Procédures stockées

Différents types d'architectures de SGBD



- Rôle :
 - Réalise l'analyse lexical et syntaxique.
 - Contrôle de la sémantique de la requête.
 - Traduction des noms en format interne par des références.

Différents types d'architectures de SGBD



Contrôleur

- Rôle :
 - Traduction des requêtes en format interne en une ou plusieurs requêtes référençant des objets existant dans la base.
 - Prendre en compte le contrôle de droit d'accès.
 - Contrôle des contraintes d'intégrité lors de la mise à jour.

Différents types d'architectures de SGBD



- Rôle :
 - **Élaborer un plan d'accès optimisé pour traiter les requêtes.**
 - **Décompose la requête en opérations d'accès élémentaires.**
 - **Choix d'un ordre d'exécution optimal.**
 - **Choix de la méthode d'accès à utiliser, en s'appuyant sur un modèle de coût.**
 - **Stockage du résultat en mémoire pour une multiple utilisation.**

Différents types d'architectures de SGBD



Exécuteur

- Rôle :
 - Exécuter le plan d'accès choisi et élaboré par l'optimiseur en s'appuyant sur les méthodes d'accès qui permettent d'accéder aux fichiers via des indexes.
 - Gérer les problèmes de la concurrence d'accès.
 - Gérer l'atomicité des transactions.

Différents types d'architectures de SGBD

Architecture Client/Serveur

Composé par :

1. Un serveur

- Gère les données partagées entre les clients.
- Exécute le code de SGBD, sous la demande des clients.

2. Des clients :

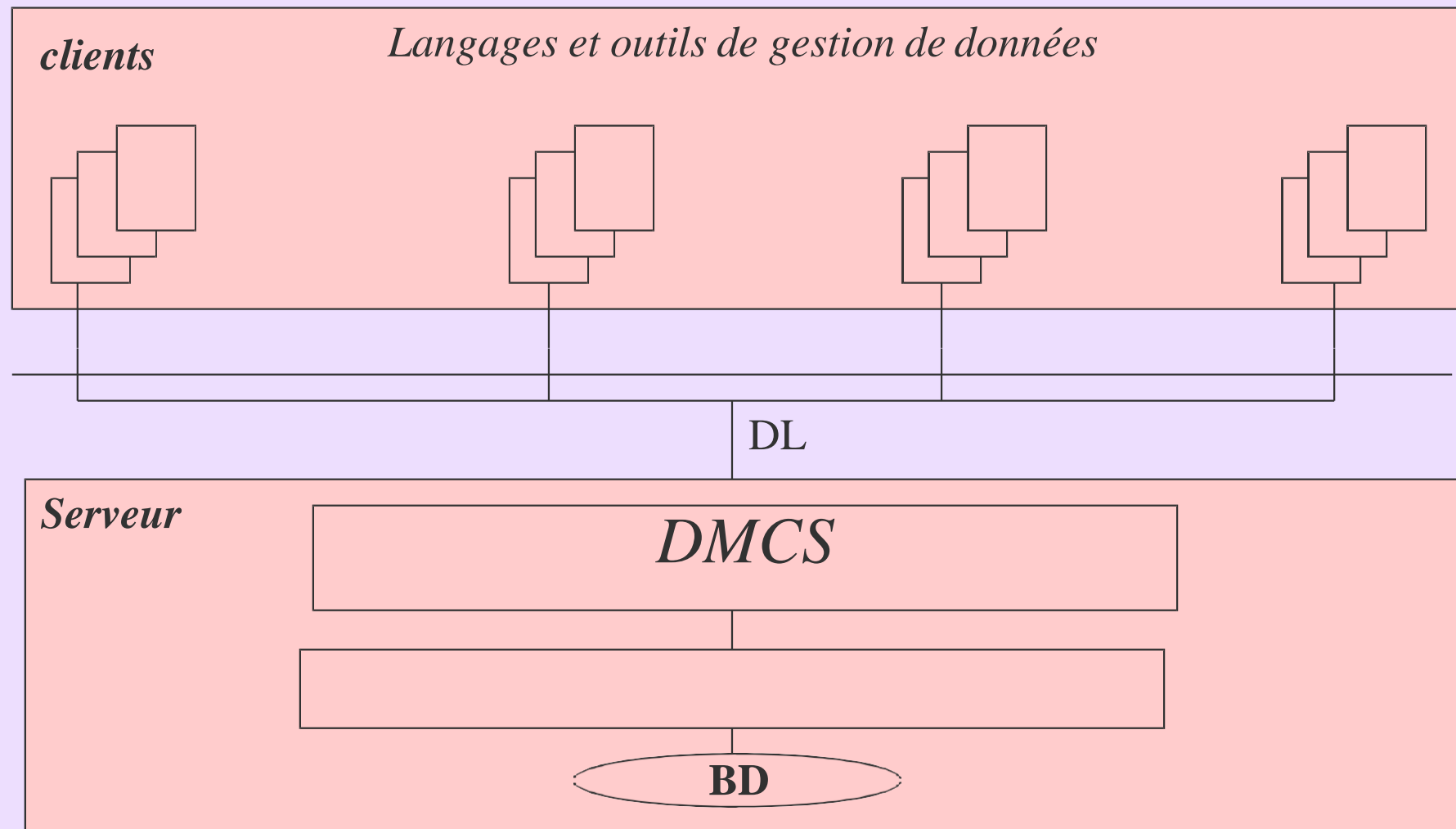
- Dialogue avec le serveur en utilisant des types de requêtes question/réponse.

3. Un réseaux :

- Assure l'acheminement des requêtes des clients et les réponses de serveur.

Différents types d'architectures de SGBD

Architecture Client/Serveur



Différents types d'architectures de SGBD

Architecture C/S à deux strates

■ Objectifs

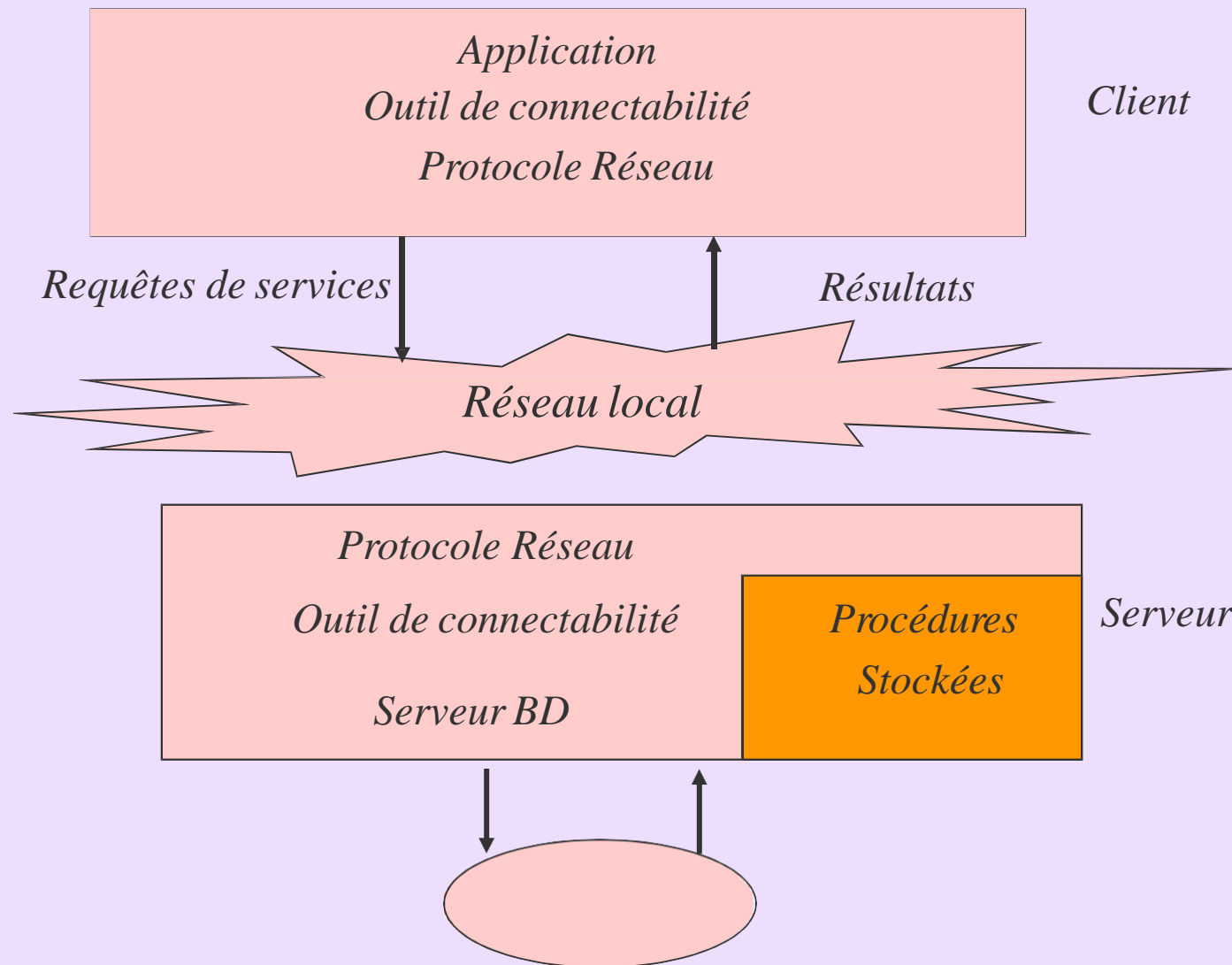
1. Réduire la communication entre les clients et le serveur.
2. Éviter des multiples commandes de transferts de données sur le réseau.

■ Principe

1. Les clients invoquent des procédures applicatives qui manipulent la base.
2. invocation des procédures stockées et transmission des paramètres de retour.

Différents types d'architectures de SGBD

Architecture C/S à deux strates



Différents types d'architectures de SGBD

Architecture C/S à trois strates

- Objectifs

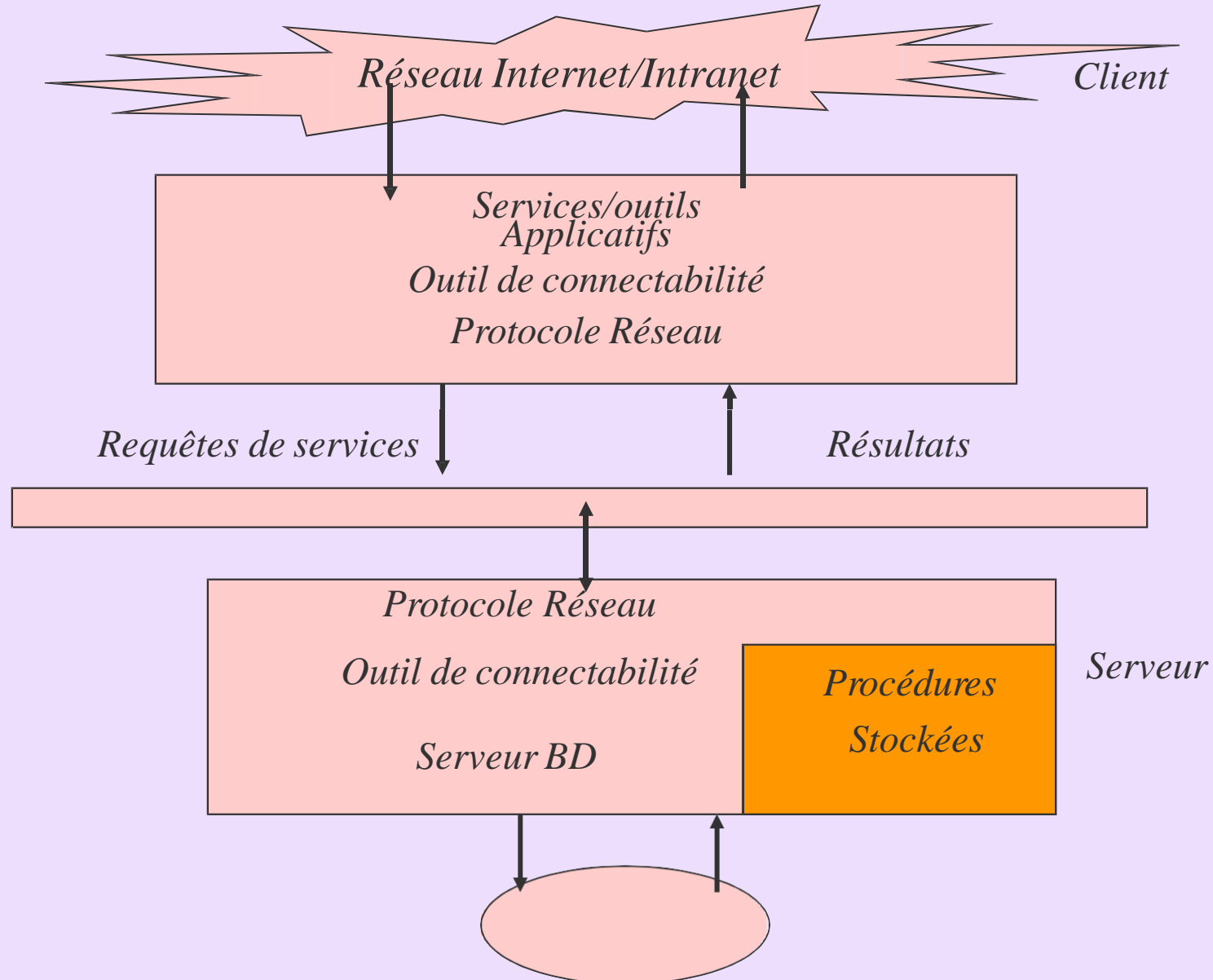
1. Le client est responsable de la présentation de données, en utilisant des browsers Web.

- Principe

1. Serveur de données exécutant le SGBD et éventuellement des procédures applicatives.
2. serveur d'application exécutant le corps des applications
3. Client responsable des dialogues et de présentation des données selon le standard Web.

Différents types d'architectures de SGBD

Architecture C/S à trois strates



Objectifs de S.G.B.D.

- Indépendance physique.
- Indépendance logique.
- Manipulation des données avec des langages non procéduraux.
- Efficacité d'accès aux données.
- Contrôle de redondance.
- Cohérence et partage de données.

Indépendance physique

■ Objectifs

- Réaliser l'indépendance des structures de stockage aux structures de données de monde réel.
- Modification de schéma interne sans avoir à modifier le schéma conceptuel.
- Indépendance de l'évolution technologique sur les applications:

■ Avantages:

- On peut ajouter des index, regrouper des fichiers changer l'ordre de codage des données dans un article, sans mettre en cause les entité association définies au niveau conceptuel
- Changer les support physiques de données plus performant sans mettre en cause le schéma conceptuel.

Indépendance logique

■ Objectifs

- Réaliser l'indépendance des données vue par les applications à la structure canonique des données décrite dans le schéma conceptuel.
- Possibilité de modifier un schéma externe sans modifier le schéma conceptuel.

■ Avantages:

- Permettre à chaque groupe de travail de voir les données comme il souhaite
- Permettre l'évolution de la vue d'un groupe de travail remettre en cause le schéma conceptuel.
- Permettre l'évolution d'un schéma externe sans remettre en cause les autres schéma conceptuel.
- Assurer la sécurité et l'intégrité de données.

■ Objectifs

- Interroger et mettre à jour les données sans préciser les algorithmes d'accès.
- Décrire les données qu'on souhaite retrouver sans décrire la manière de les retrouver, qui est propre à la machine.

■ Avantages:

- Simplicité d'utilisation de ces langages par des utilisateurs non informaticiens.
- L'exploitation de l'indépendance des applications et la manière de définition et stockage de données.

Manipulation des données avec des langages non procéduraux

- **Limite :**
 - **L'absence des langages qui peuvent satisfaire tous les besoins des utilisateurs (Proche au langage naturel)**
- **Solutions & Problématiques :**
 - **Utilisation des langages procéduraux (3ème génération) avec les langages d'interrogation, on parle de langage de 4ème génération.**
 - **Utilisation des langages Orienté Objet comme C++ ou bien JAVA.**
 - **Le problème principal est le dysfonctionnement des langages procéduraux et ensembliste.**

Efficacité d'accès aux données

■ Objectifs

- **Augmenter la performance en terme de débit : Nombre des transactions type exécutées par seconde [ES(Q)].**
- **Augmenter la performance en terme de temps de réponse : le temps d'attente moyenne pour une requête type [UC(Q)].**
- **Équation à optimiser est :**

$$C(Q) = a * ES(Q) + b * UC(Q)$$

Contrôle de redondance

■ Objectifs

- **Une conception proche de la réalité : les objets communs doivent être uniques et partagés entre tous les utilisateurs.**
- **La non duplication physique de données afin d'éviter les mises à jour multiples.**
- **Optimiser l'utilisation de l'espace disque.**

Contrôle de redondance

■ Problématiques :

- **Dans un environnement réparti, l'interrogation d'une base de données nécessite un temps de réponse très important dû au transfert des données sur le réseau.**
- **L'utilisation d'une seule base de données centralisée, empêche les SGBD distants de fonctionner en parallèle.**
- **Si le serveur de données tombe en panne, toutes les données vont être perdues.**

■ Solutions :

- **La réplication efficace des données d'une manière invisible et transparent pour l'utilisateur.**

Cohérence de données

■ Objectifs

- Les données doivent représenter des entités du monde réel.
 - L'âge d'une personne compris entre 0 et 180 ans
- les données stockées doivent être cohérentes avec d'autres données.
 - Le nombre des commandes du client doit correspondre au nombre de commandes dans la base de données.
- Les données à traiter doivent être stockées
 - Le client d'une commande doit être enregistré dans la base de données.

■ Solutions :

- Spécification de certaines règles sémantiques et d'intégrité. Et le système contrôle à ce que les applications respectent ces règles lors de modification.

■ Objectifs

- **Permettre aux applications de partager les données de la base dans le temps mais aussi simultanément.**
 - ① **Plusieurs utilisateurs peuvent manipuler les mêmes données en même temps (sans blocage).**
 - ① **La duplication de mêmes données sur des supports physique différents, tous en assurant leurs cohérences.**
- **Une application doit pouvoir accéder aux données comme si elle était seule à les utiliser.**

■ Objectifs

- **Assurer que le résultat d'une exécution simultanée de transaction reste le même que celui d'une exécution séquentielle dans un ordre quelconque des transactions.**
- **Assurer la disponibilité des données même dans le cas des pannes.**
- **Empêchement de certaines modifications non autorisée aux utilisateurs.**

Partage Sûreté et Sécurité de données

■ Objectifs

- **Protection des données contre les accès non autorisés ou mal intentionnés, par l'accordement des droits d'accès.**
- **Restaurer les données cohérentes en cas des pannes du système ou la machine, par l'utilisation des mécanisme de reprise aux pannes.**

■ Solutions :

- **Pour plus de détail voir la partie des Bases de données Avancées Chapitres Bases de Données Réparties (protection de données)**

Fonctions des S.G.B.D.

- Description de données
- Recherche de données
- Mise à jour des données
- Transformation des données
- Protection de données
- Contrôle de l'intégrité de données
- Gestion de transactions et de sécurité
- Sûreté des données et reprise aux pannes

Description des données

- Dictionnaire de données
 - Ensemble de schémas et des règles de passage entre les schémas associés à une base de données, combinés à une description de signification de données et les contraintes de leurs évolutions.
 - ① Structurelle.
 - ① Comportementale.

Description des données

- Administrateur de données:
 - Administrateur de bases de données
 - ① Exécution du schéma interne et des règles de correspondance entre les schémas internes à conceptuel.
 - Administrateur d'entreprise
 - ① Définition du schéma conceptuel
 - Administrateur d'applications
 - ① Définition des schémas externes et des règles de correspondance entre les schémas externes et conceptuel.

Description des données

- Commande de base pour définir les schémas interne, conceptuel et externe.
 - Créer une base de données
 - CREATDB <Nom_base>
 - Créer une entité
 - CREATE ENTITY <Nom_entité>
(nom_attribut Type)
 - Créer une relation
 - CREATE RELATIONSHIP <Nom_relation>
(Nom_entite, ..., Nom_Attribut)
 - Détruire une entité, une association ou une base
 - DESTROY Nom_relation|Nom_association
 - DESTROYDB Nom_DB

Manipulation de données

- Les SGBDs offrent un langage d'interrogation assertionnel permettant de retrouver ou mettre à jour les données par le contenu sans préciser la procédure d'accès.

Transformation de données

C'est une fonction effectuant la restauration des instances de données conformes à un schéma en instance de données conformes à un autre schéma.

- Transformation conceptuelle-interne
 - Permettant de faire passer les instances de données depuis le format conceptuel au format interne de données.
- Transformation externe-conceptuelle
 - Permettant de faire passer des instances de données depuis le format conceptuel au format externe.

Contrôle de l'intégrité de données

Contrainte d'intégrité de données, c'est toute règle implicite ou explicite que doit suivre les données, pour assurer la cohérence et la non redondance de données .

- Unicité de la clé.
- Contrainte de domaine.
- Contrainte référentielle.
- Valeur nulle.

Gestion des transactions

Permet d'assurer qu'un groupe de mises à jour est totalement exécutée ou pas des tout.

(ACID)

- Atomicité des transactions
- Correction des transactions
- Isolation de transactions
- Durabilité de transactions

Types de S.G.B.Ds

-
- » *Réseaux & hiérarchiques*
 - » *Relationnels*
 - » *Orientés objet*
 - » *Répartis*
 - » *Déductifs*
 - » *Géographiques*
-