

Présentation du cours

Objectifs du cours de BDA

- Présenter les **aspects avancés** des bases de données et **les techniques permettant de les implémenter**.
 - **Déclencheurs** et **facilité d'accès**
 - Optimisation des **performances**
 - Gestion des accès **simultanés** et **concurrents**
- Architecture et fonctionnement techniques des SGBD.
 - **Automatisation** de traitements
 - Organisation et **stockage physique** des données
 - Etapes du **traitement** de requêtes
 - **Optimisation** de requêtes
 - Gestion des **transactions**
 - Contrôle de **concurrency**
 - Gestion des **pannes**

Plan du cours de BDA

1. **Mise au point** : rappels sur les bases de données
 - Conception, modèle relationnel, algèbre relationnelle, langage SQL
2. **Compléments sur la manipulation des données**
 - Requêtes complexes
 - Vues, index,
 - Triggers et procédures stockées
3. **Méthodes d'accès** : Stockage et indexation de données
 - Stockage des données, organisation et indexation (hachage, arbre B, arbre B+)
4. **Evaluation et optimisation** de requêtes
 - Plan d'exécution de requêtes
 - Stratégies d'optimisation de requêtes
5. Gestion des **transactions**
 1. Gestion des pannes
 2. Contrôle de la concurrence

Rappels sur les bases de données

Dr N. BAME

Plan

- Base de données
- Systèmes de Gestion de Base de données
- Modèle Relationnel
- Algèbre Relationnel
- Langages SQL

Notion de base de données

- Ensemble cohérent de ***données structurées, fiables et partagées*** entre plusieurs utilisateurs
 - Collection de ***données opérationnelles enregistrées*** sur un support adressable et utilisées par les systèmes et les applications
 - Elles doivent être ***cohérentes*** (contraintes), ***non redondantes*** (formes normales) et accessibles ***simultanément par plusieurs*** utilisateurs

Système de Gestion de Base de données

- ***Ensemble d'outils logiciels*** permettant la ***gestion*** et ***l'exploitation*** des bases de données
- Fournit un ***langage*** de ***description*** des données et un langage de ***manipulation*** des ***données***
 - ***DDL***: Data Definition Language
 - ***DML***: Data Manipulation Language
 - ***DCL***: Data Control Language

Systeme de Gestion de Base de données

- **Contrôle intégré** des données
 - cohérence et intégrité
 - **partage**
 - performances d'accès
 - Sécurité
- **Indépendance** des données
 - **logique** : **cache** les détails de **l'organisation conceptuelle** des données
 - **physique** : **cache** les détails du **stockage physique** des données
- La **description** de la base est elle même **gérée** comme une base et s'appelle le **dictionnaire (catalogue)**

Fonctions d'un SGBD

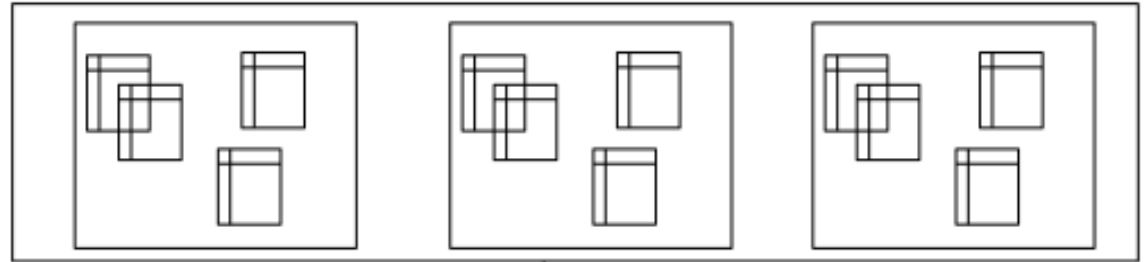
- **Schéma intégré**
 - vue uniforme des données, sous formes de *relations*
- **Intégrité déclarative et cohérence**
 - Primary key, foreign key, check, ...
 - l'utilisateur **spécifie** et le **SGBD valide**
- **Accès déclaratif**
 - avec un langage de requête (SQL), pour permettre à l'utilisateur d'exprimer ses besoins

Fonctions d'un SGBD

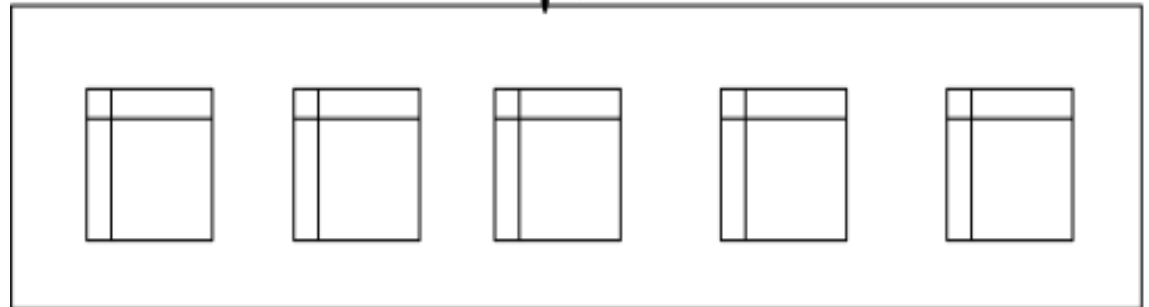
- **Traitement et optimisation de requêtes**
 - performances obtenues automatiquement
- **Transactions**
 - exécution des requêtes par des unités atomiques
 - indépendance à la concurrence multi-utilisateurs et aux pannes
- **Administration système**
 - Contrôle des accès aux données
 - outils d'audit et de réglage (tuning)
 - visualisation des plans d'exécution

Architecture d'un SGBD

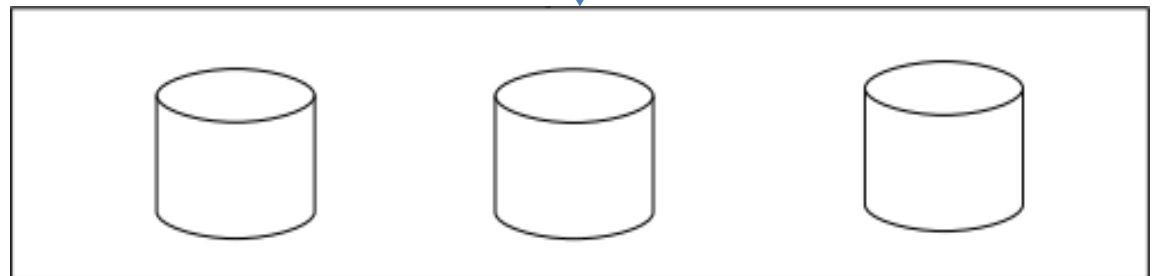
Niveau externe



Niveau logique



Niveau physique



Niveau physique

- S'appuie sur le **système de gestion de fichiers** de l'OS pour **stocker les données**
- Gère le **partage** des données et les **accès concurrents**
- Assure la **fiabilité** des données, notamment suite à des **pannes**
- La personne responsable de ce niveau est **l'administrateur**

Niveau logique

- Correspond à la ***vision des données*** indépendamment des applications individuelles
- Le ***langage de définition de données*** permet de mettre en place les données de manière structurée et correspondant au modèle élaboré
- Le ***langage de manipulation*** de données permet la ***consultation*** et la ***mise à jour*** de ces données

Niveau externe

- Correspond aux **vues** utilisateurs via leurs **applications**
- Les accès utilisateurs peuvent être distants, provenir de **diverses interfaces et langages**
- **Chaque utilisateur** ne voit que les données **qui lui sont utiles**
- Il permet le **dialogue courant** avec la base de données

Modèle relationnel

- Définition rigoureuse des données
- Règles d'intégrité garantissant la cohérence des données
- S'appuie sur les opérateurs de l'algèbre relationnelle
- **Concepts**
 - *Domaine* : ensemble de valeurs
 - *Relation* : tableau à deux dimensions
 - *Attribut* : nom d'une colonne du tableau

Schéma relationnel

- ***Schéma d'une relation*** : nom de la relation suivi de la liste des attributs et de la définition de leurs domaines
- ***Schéma relationnel*** : ensemble des schémas des relations d'une base de données
- ***Cardinalité de la relation*** : nombre de tuples
- ***Degré de la relation*** : nombre d'attributs

Exercice

- Donner un schéma relationnel d'une base de données pour une application de gestion commerciale.

Contraintes d'intégrité

- Expression logique qui doit être ***vérifiée*** en ***permanence*** dans la base
- Contrôler la ***cohérence*** des données
 - Contrainte de ***clé***
 - Contrainte de ***domaine***
 - Contrainte de ***références***
 - ...

Algèbre relationnelle

- ***Ensemble d'opérations*** qui s'applique ***sur*** des ***relations*** pour donner de ***nouvelles relations***
- Ces opérations s'effectuent grâce à un certains nombres ***d'opérateurs***
- On distingue les opérateurs ***ensemblistes*** issues de la théorie des ensembles et les opérateurs purement ***relationnels***

Opérateurs de l'algèbre relationnelle

- **Projection** : création d'un schéma sous-ensemble d'un autre schéma
- **Sélection** : Même schéma mais réduction du nombre de tuples grâce à un critères (ou prédicat)
- **Produit de cartésien** : Création d'un schéma à partir de 2 tables avec concaténation des attributs et combinaison systématique des tuples
- **Jointure** : Création d'un table à partir de 2 tables avec *union des attributs* et *concaténation des tuples* satisfaisant à la *condition* de jointure

Opérateurs de l'algèbre relationnelle

- **Union** : création d'une table à partir de 2 tables ayant le *même schéma*. La table résultant contient *l'ensemble* des tuples
- **Différence** : création d'une table à partir de 2 tables ayant le *même schéma*. La table résultante contient les *tuples appartenant à une* table et pas à l'autre
- **Intersection** : création d'une table des tuples *communs* de 2 tables ayant le même *schéma*
- **Division** : création d'une table à partir de 2 tables. Une table a un schéma sous-ensemble de l'autre. Les tuples résultant sont ceux dont les valeurs sont celles correspondant au sous-ensemble

Opérateurs de base

- **Opérateurs de base**
 - ✓ l'union
 - ✓ la différence
 - ✓ le produit cartésien
 - ✓ la restriction (sélection)
 - ✓ la projection
- **Opérateurs dérivés**
 - ✓ L'intersection
 - ✓ La jointure
 - ✓ La division

Exercice

- Soient les relations suivantes :
 - **Employe**(eno, nom, profession, dateEmbauche, salaire, #numeroDept)
 - **Departement**(numeroDept, nom, ville, #directeur)
- Exprimer les requêtes suivantes à l'aide de l'algèbre relationnelle :
 1. Donner les noms et les salaires des employés.
 2. Donner les professions des employés.
 3. Donner les dates d'embauche des techniciens.
 4. Donner les noms des employés et les noms de leur département.
 5. Donner les numéros des employés travaillant à Louga.

Utilisation des opérateurs

- Dans les requêtes seuls les opérateurs les plus maniables sont utilisés : ce sont *l'union* et la *différence* pour *l'insertion* et la *suppression* de tuples dans la base et la *restriction*, la *projection* et la *jointure* pour la *recherche sélective* de tuples.
- Les opérateurs de l'algèbre relationnelle sont à la base des langages de manipulations de données (SQL)

Langage SQL

- Langage de **définition** de données (LDD) et de **manipulation** de données (LMD) des **bases de données relationnelles**
- Langage de *définition* de données permet de **créer**, *modifier*, *supprimer* les éléments du **schéma relationnel**
- Langage de *manipulation* de données permet *l'interrogation*, *l'ajout*, la *modification*, la *suppression* de données

Les instructions SQL

- **Extraction de données**
 - SELECT
- **Manipulation de données**
 - INSERT
 - UPDATE
 - DELETE
- **Définition de données**
 - CREATE
 - ALTER
 - DROP
 - RENAME
 - TRUNCATE

Les instructions SQL

- **Contrôle de transactions**
 - COMMIT
 - ROLLBACK
 - SAVEPOINT
- **Contrôle des données**
 - GRANT
 - REVOKE

Exercice SQL 1

- Soit la base RH avec les relations suivantes :
 - **Employe**(eno, nom, profession, dateEmbauche, salaire, #numeroDept)
 - **Departement**(numeroDept, nom, ville, #directeur)

Donner les ordres SQL permettant de :

- Créer la base de données RH et s'y connecter
- Créer la table Employe avec la contrainte de clé primaire et les contraintes de vérification sachant que le salaire minimal est 400000 et que la date d'embauche doit être antérieure à la date courante.
- Créer la table Departement avec les contraintes de clé primaire et de clé étrangère.
- Modifier la table Employe pour ajouter la contrainte de clé étrangère.

Exercice SQL 2

- Soient les relations suivantes :
 - **Employe**(eno, nom, profession, dateEmbauche, salaire, #numeroDept)
 - **Departement**(numeroDept, nom, ville, #directeur)
- Exprimer les requêtes suivantes en SQL:
 1. Donner les noms et les salaires des employés.
 2. Donner les professions des employés.
 3. Donner les dates d'embauche des techniciens.
 4. Donner les noms des employés et les noms de leur département.
 5. Donner les numéros des employés travaillant à Louga.