

# **Chapitre 1: Concepts fondamentaux**

**Dr. Chergui Leila**

### Introduction

Auparavant, les données sont stockées sous forme de fichiers et ils sont gérées par un système de gestion de fichiers, il y avait deux types de fichiers :

1. Fichier des données : représentent des séquences d'enregistrements dont l'accès est séquentiel ou indexé.
2. Fichier de traitement : représentent un ensemble d'instructions servant à manipuler les données des fichiers (Cobol, Pascal).

Mais cette approche a souffert des problèmes suivants :

1. Redondance d'information (Mise A Jours (MAJ) difficiles).
2. Problème d'incohérence, de fiabilité et de sécurité des données.
3. Manque de structuration des données.
4. Si un changement est effectué sur la structure d'un fichier de données tous les programmes qui l'utilisent doivent être modifiés et on va perdre du temps pour la maintenance.

**Solution :** regrouper les fichiers de données en une seule entité= BD, dont les données et les traitements sont indépendants.

### 1.1. Concepts de base

Dans ce qui suit on va expliquer le principe de quelques concepts de base.

#### 1.1.1. Base de Données (BD)

Une BD est une collection d'information modélisant une entreprise du monde réel, c'est un :

- Ensemble structuré de données.
- Enregistré sur des mémoires secondaires (disque dur).
- Créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs ou de programmes.

#### 1.1.2. Objectifs d'un Système de Gestion de Base de Données

Un Système de Gestion de Bases de Données (SGBD) (en anglais *DBMS* pour *Database Management System*) est un système qui permet de créer et de manipuler une BD partagée par plusieurs utilisateurs simultanément.

Il représente un ensemble de services (applications logicielles) permettant de :

- Définir la structure d'une BD (de décrire les données et les liens entre elles d'une façon logique sans se soucier du comment cela va se faire physiquement sur le disque) par un Langage de Description de Données (LDD).

- Manipuler des données : rechercher, insérer, modifier, supprimer des données dans une BD (Langage de Manipulation des Données : LMD).
- Assurer la cohérence des données : les données sont soumises à certains nombre de contraintes d'intégrité qui définissent un état cohérent de la base. Ces contraintes sont décrites dans le langage de description de données (LDD). Ex : l'âge des employés ne peut pas être supérieur à 55. Si un utilisateur ajoute une entité employé, le SGBD doit vérifier l'attribut âge, s'il ne respecte pas cette contrainte, il est rejeté. Une contrainte d'intégrité est une propriété que les objets décrits par le schéma (entité, association, attribut) doivent respecter de manière à représenter le problème.
- Assurer la sécurité des données : elle inclut la confidentialité et la sécurité après panne.
- Assurer la confidentialité des données : les données doivent pouvoir être protégées contre les accès non-autorisés. Pour cela, il faut associer à chaque utilisateur des droits d'accès aux données.
- Assurer la sécurité après panne : on peut tomber dans des situations où certains fichiers ne sont plus lisibles ou une panne au milieu d'une MAJ. Le SGBD doit assurer la reprise après panne. Parmi les méthodes les plus utilisées on cite la journalisation qui consiste à mémoriser les états successifs de la BD.
- Permettre le partage des données : permettre à plusieurs utilisateurs d'accéder aux mêmes données au même moment de manière transparente, en contrôlant les accès concurrents.
- Assurer une efficacité d'accès : en terme du temps de réponse et de débit global ainsi que des modes d'accès simples. Le temps de réponse est le temps d'attente moyen pour une requête. Le débit global représente le nombre de transactions exécutées par second.

### 1.1.3. Domaines d'application d'un SGBD

Les SGBD sont des logiciels complexes et stratégiques, utilisés dans de très nombreuses applications informatiques, parmi lesquelles :

1. Gestion d'entreprise (stock, personnel, client).
2. Banque (comptes, emprunte).
3. Système de réservation (avion, train).
4. Bibliothèque.
5. Géographie (carte routière : GPS).
6. Le E-commerce.
7. Les dossiers médicaux.

8. Le web.

**1.1.4. Niveaux d'abstraction**

On peut distinguer trois niveaux d'abstraction comme il est illustré dans la figure 1.2 :

- Niveau externe : vue partielle des données suivant l'utilisateur (utilisateur final ou programmeur d'application). Ex : l'utilisateur n° 1 est chargé de paie, donc il voit que les informations employé. Le programmeur n° 1 est chargé de faire les facturations, donc il s'intéresse aux tables client et commande. Il existe plusieurs schémas externes.
- Niveau conceptuel : vue globale de l'organisation des données, c'est la définition logique de la BD (représentation) via le modèle de données, elle est faite par l'administrateur de la BD qui est chargé d'identifier et décrire les regroupements de données et leurs interactions. Ex : nom, prénom, adresse, etc. Il existe un seul schéma logique.
- Niveau interne : organisation physique des données qui concerne le type de stockage et les modes d'accès. Ce niveau est réalisé par le SGBD. Ex : nom : 20 octets, prénom : 20 octets, adresse : 40 octets, salaire : 4 octets.

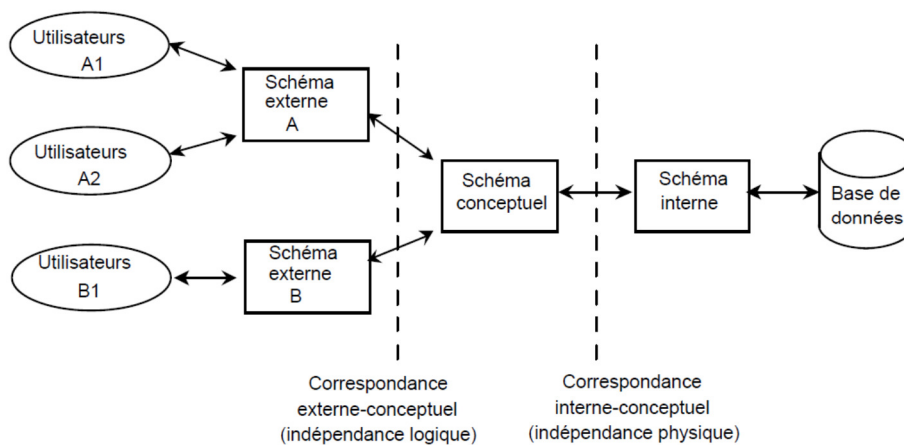


Figure 1.2. Niveaux d'abstraction dans un SGBD.

**1.1.5. Indépendance de données**

L'architecture à trois niveaux définie ci-dessus permet de garantir l'indépendance des données par rapport aux programmes : elle permet de modifier le schéma de la base de données à un niveau sans restructurer les autres.

On trouve deux types d'indépendance :

1. L'indépendance physique : est la possibilité de changer le schéma physique et de modifier l'organisation physique des fichiers, de rajouter ou supprimer des méthodes d'accès sans remettre en cause le schéma conceptuel.

2. L'indépendance logique : est la possibilité de modifier le niveau conceptuel sans changer le schéma externe.

### 1.1.6. Les modèles logiques

1. Le modèle hiérarchique : ou arbre, il lie les enregistrements dans une structure arborescente où chaque enregistrement n'a qu'un seul possesseur. Pour chaque nœud donné, un seul nœud père. Ex : ADATABASE (1970), IMS (1966), System 2000 (1967).
2. Le modèle réseau : ou graphe, c'est un modèle hiérarchique, mais permet en plus d'établir des relations transverses. Ex : TOTAL (1978), IDMS (1978), IDS2 (1978), SOCRATE (1973).
3. Le modèle relationnel : il stocke les informations décomposées et organisées dans des tables. On note que 80% des SGBD sont relationnelles. Ex : ORACLE (85% du marché), DB2, SQL Server, My SQL (libre), Postgre SQL (libre), ACCESS, PARADOX , DBASEV, INGRES, INFORMIX, RDB.
4. Le modèle orienté objet : il stocke les informations groupées sous forme de collections d'objets persistants dans des classes. Ex : Versan, Object store, O2, ONTOS, ORION.
5. Le modèle relationnel-objet.

Des exemples de modèle hiérarchique et modèle réseau sont affichés dans la figure 1.3.

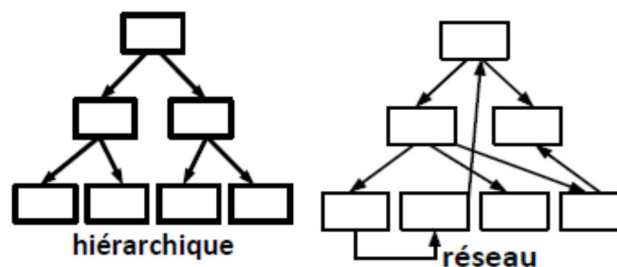


Figure 1.3. Exemples de modèles logiques.