

# Chapitre 1 : Présentation des bases de données

## 1. Notion de fichier

### 1.1. Définition

La notion de fichier (File) a été introduite en informatique durant les années 50.

Un fichier est un *réceptif d'information* caractérisé par un nom, constituant une mémoire secondaire idéale, permettant d'écrire des programmes d'application indépendants des mémoires secondaires [1].

Il y avait deux types de fichiers :

1. *Fichier des données* : des séquences d'enregistrements dont l'accès est séquentiel ou indexé.
2. *Fichier de traitement* : un ensemble d'instructions permettant la manipulation des données des fichiers (e.g., assembleur, Cobol).

### 1.2. Intérêt

L'objectif des fichiers était de simplifier l'utilisation des mémoires secondaires des ordinateurs. Les fichiers fournissent des réceptifs de données plus manipulables aux programmes et sont gérés par un système de gestion de fichiers.

Suite à la sophistication des systèmes informatiques, les données stockées dans des fichiers (informations système ou utilisateur) sont devenues structurées. En effet, aujourd'hui, les fichiers sont à la base des systèmes d'information. De ce fait, le premier niveau d'un SGBD (Système de Gestion de Base de Données) est la gestion de fichiers. Le SGBD sera présenté dans les paragraphes suivants.

Les données gérées par l'entreprise et les programmes spécifiant les traitements sur les données sont stockés dans des fichiers gérés par le système informatique. Par exemple, le traitement de l'application de comptabilité d'une entreprise de livraison de produits (gestion des comptes des clients et édition des factures, etc.).

La gestion des fichiers permet de traiter et de stocker des quantités importantes de données, et de les partager entre plusieurs programmes. De plus, elle sert de base au niveau interne des Systèmes de Gestion de Bases de Données.

### 1.3. Limites des systèmes de fichiers

Cette approche a souffert des problèmes suivants :

- Suite à la *redondance* des données (i.e. les doublons) dans les fichiers, le volume des données et le temps d'accès aux données augmentent et la validité du système diminue.
- La *mise à jour* dans l'approche système de fichier est pénible parce que le changement d'une donnée entraîne la nécessité de la changer partout pour conserver la validité du système (e.g., si l'adresse d'un client change, il faut la mettre à jour partout).

- L'accès à l'information est problématique : recherche et lecture/écriture de l'information.
- L'application est *dépendante* du mode de stockage des données.
- L'utilisation de fichiers oblige l'utilisateur de connaître : le *mode d'accès* à savoir séquentielle, indexé ou autre, la *structure physique* des enregistrements ainsi que la *localisation* des fichiers utilisés.
- Lorsqu'il s'agit des applications nouvelles, l'utilisateur doit : *écrire de nouveaux programmes, créer de nouveaux fichiers* contenant des informations qui existent déjà. Par conséquent, toute *mise à jour* de la structure des enregistrements (e.g., ajout d'un champ) conduit à la réécriture de tous les programmes dédiés à la manipulation de ces fichiers.
- Manque de sécurité : Dans le cas où tout programmeur peut accéder d'une manière directe aux fichiers, il n'est pas possible d'assurer la *sécurité* et l'intégrité des données.
- Dans un environnement où plusieurs utilisateurs accèdent aux mêmes fichiers, des problèmes de *concurrence* d'accès se posent.
- Les données *ne sont pas structurées*.

Une solution évidente à ces limites consiste à regrouper les fichiers de données en une seule entité appelée 'Bases de Données', dont les données et les traitements sont indépendants.

## 2. Définition de base de données (BD)

Plusieurs définitions ont été proposées, on peut citer entre autres les définitions suivantes :

- Selon Gardarin [1], une base de données est un ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique. Pour mériter le terme de base de données, un ensemble de données non indépendantes doit être interrogeable par le contenu, c'est-à-dire que l'on doit pouvoir retrouver tous les objets qui satisfont à un certain critère.
- Une Base de données est un ensemble structuré de données apparentées qui modélisent un univers réel. Une BD est faite pour enregistrer des faits, des opérations au sein d'un organisme (administration, banque, université, hôpital, ...). Les BD ont une place essentielle dans l'informatique [9].
- Une Base de données (BD) est un ensemble cohérent, intégré, partagé de données structurées défini pour les besoins d'une application [10].

- Une Base de données est un gros ensemble d'informations structurées mémorisées sur un support permanent [4].

### 3. Système de gestion de base de données

#### 3.1. Définition

Un SGBD (en anglais DBMS pour Database Management System) est un logiciel système qui permet de manipuler (insertion, suppression, mise à jour, recherche efficace) de grandes quantités de données stockées dans une base de données. Ces données peuvent atteindre quelques milliards d'octets partagée par de multiples utilisateurs simultanément. Les données stockées sont partagées en interrogation et en mise à jour d'une manière transparente. D'autres fonctions complexes peuvent être assurée par le SGBD telle que la protection des données partagées contre les incidents.

Contrairement aux systèmes de fichiers, les SGBD permettent de décrire les données de manière séparée de leur utilisation et de retrouver les caractéristiques d'un type de données à partir de son nom (par exemple, comment est décrit un article). La Figure 1.1 représente l'architecture générale d'un SGBD.

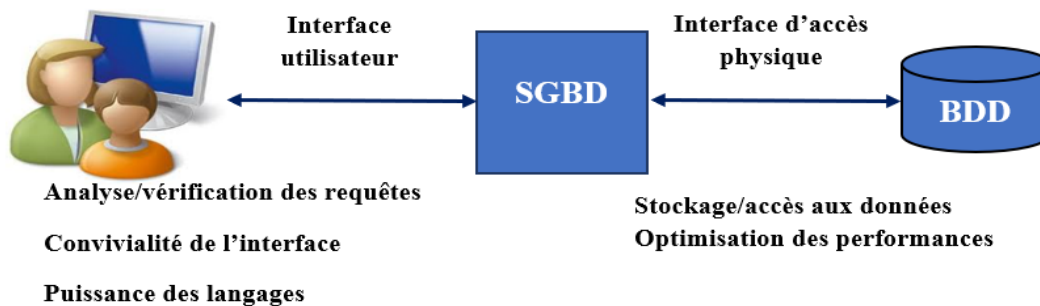


Figure 1.1. Architecture Générale d'un SGBD [11].

#### 3.2. Niveaux d'abstraction

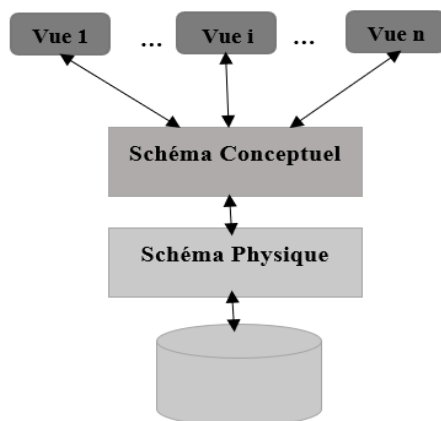
Un objectif majeur des SGBD est d'assurer une abstraction des données stockées sur disques pour simplifier la vision des utilisateurs [1]. Pour cela, trois niveaux de description de données ont été définis :

- **Niveau conceptuel** : Le niveau *conceptuel*, également appelé niveau *logique*, est le niveau central. Le schéma conceptuel permet de décrire l'ensemble des données de l'entreprise. Exemple : nom, prénom, adresse, etc. C'est une définition logique de la BD (représentation) via le modèle de données et est faite par l'administrateur de la BD qui identifie et décrit les regroupements de données et leurs interactions.
- **Niveau interne** : Ce niveau appelé aussi niveau *physique* permet la gestion : du stockage des données sur des supports physiques, des structures de mémorisation (fichiers) et d'accès (gestion des index, des clés, etc.).

- **Niveau externe** : Ce niveau est aussi appelé niveau *vue*, c'est le plus haut niveau d'abstraction de la base de données. A la différence du niveau conceptuel et interne dont les schémas décrivent toute une base de données, les schémas externes décrivent seulement la partie des données présentant un intérêt pour un utilisateur ou un groupe d'utilisateurs. En d'autres termes, le niveau externe associe à un utilisateur ou à un groupe d'utilisateurs une vue partielle du monde réel. En conséquence, il y a plusieurs vues d'une même BDD.

Note : Pour une BDD, la description conceptuelle et le schéma interne sont uniques. Par contre, plusieurs schémas externes en général peuvent être définis.

La Figure 1.2 présente les trois schémas d'une BD centralisée.



**Figure 1.2.** Les trois niveaux de schémas.

### 3.3. Objectifs des SGBD

L'objectif essentiel d'un SGBD est de garantir l'indépendance des données par rapport aux programmes (i.e. possibilité de modifier les schémas conceptuel et interne des données sans modifier les programmes). Le but est d'éviter une maintenance coûteuse des programmes lors des modifications des structures logiques et physiques des données. Par cela, on trouve l'indépendance *physique* et *logique*.

En plus, pour assurer une meilleure indépendance des programmes aux données il est important de *manipuler* les données aussi bien en interrogation qu'on mise à jour via des langages. De ce fait, l'*accès aux données* restent invisibles aux programmes d'application. Les descriptions de données sont établies par les administrateurs des données, donc, le SGBD doit faciliter cette tâche.

Lorsque le SGBD met en commun les données d'une entreprise dans une BD, et que plusieurs utilisateurs accèdent simultanément aux données, il est nécessaire de garantir : *l'efficacité des accès*, *le partage des données*, *la protection* de la BD contre les mises à jour erronées ou non autorisées et la *sécurité* des données en cas de panne.

En résumé, voici les objectifs des SGBD :

- **Indépendance physique** : permettre le changement du schéma physique (modifier l'organisation physique des fichiers, d'ajouter ou supprimer des méthodes d'accès) sans changer le schéma conceptuel. Cela présente deux avantages : le fait de ne pas manipuler des entités complexes rend les programmes d'application plus simples à écrire, la modification des

applications n'est pas obligatoire dans le cas de modification des caractéristiques du niveau physique.

- **Indépendance logique** : possibilité de modification du niveau conceptuel sans changement du schéma externe. Cela a deux avantages : pour les programmes d'application du niveau vue, il n'est pas nécessaire d'avoir une vue globale de l'entreprise. En outre, en cas de modification du schéma du niveau logique, les applications du niveau vue sont réécrites seulement si cette modification entraîne celle de la vue.
- **Manipulation des données** : permettre à tous types d'utilisateurs d'accéder à la base selon leurs besoins et connaissances. Par conséquent, un ou plusieurs :
  - *Administrateurs* de la base doivent avoir la possibilité de décrire les données aux niveaux interne et logique,
  - *Développeurs* d'applications écrivent des programmes d'application pour les utilisateurs finaux ou pour eux-mêmes, cela est à partir du niveau conceptuel ou externe,
  - *Utilisateurs* peuvent manipuler les données via un langage simple dont ils ont besoin.

- **Administration facilitée des données** :

En effet, la centralisation des descriptions de données faites par un groupe spécialisé entraîne une difficulté d'organisation. Le SGBD permet de décentraliser cette description à travers des outils. Pour cela, un dictionnaire de données dynamique peut aider les concepteurs de BD.

- **Redondance contrôlée des données** :

La suppression des données redondantes permet d'assurer la cohérence de l'information ainsi que la simplification des mises à jour. En effet, avec les BD réparties, il est préférable de gérer par le système des copies multiples de données. L'objectif est d'optimiser les performances en interrogation tout en évitant les transferts sur le réseau et permettre le parallélisme des accès. Conséquemment, parfois la redondance gérée par le SGBD est nécessaire spécialement au niveau physique des données.

Par contre, il faut éliminer la redondance anarchique qui force les programmes utilisateurs à mettre à jour une même donnée plusieurs fois. Donc, il faut bien contrôler la redondance.

- **Cohérence des données** :

Dans une approche BD, les données ne sont pas indépendantes (i.e. il peut exister certaines dépendances entre ces données). En d'autres termes, souvent une donnée ne peut pas prendre une valeur quelconque.

Exemples : une note doit être supérieur ou égale à 0 et ne doit pas dépasser 20, un salaire mensuel doit être supérieur à 40 000 Dinar et doit raisonnablement rester inférieur à 200 000 Dinar.

Un SGBD doit veiller à ce que ces règles soient respectées par les applications lors des modifications des données et ainsi assurer la cohérence des données. Ces règles sont appelées contraintes d'intégrité.

- **Partage des données :**

L'objectif est ici de permettre le partage des données entre différents utilisateurs et applications. Un utilisateur n'a pas à se soucier si quelqu'un d'autre travaille sur les mêmes informations au même moment et peut accéder aux données en consultation ou en mise à jour comme s'il était seul. Le système doit gérer les conflits en refusant ou en retardant éventuellement un ou plusieurs accès. Il faut assurer que le résultat d'une exécution simultanée de transactions est le même que celui d'une exécution séquentielle. Par exemple, ne pas autoriser la réservation du même siège pour deux passagers différents.

- **Sécurité des données :**

La sécurité des données consiste à :

- Refuser les accès aux personnes non autorisées ou mal intentionnés. Le système doit présenter un mécanisme de vérification des droits d'accès aux objets de la base.  
Par exemple : un employé peut connaître seulement les salaires des personnes qu'il dirige mais pas le salaire des autres employés de l'entreprise.
- Protéger les données contre les pannes. Le système doit garantir des reprises après panne tout en restaurant la BD dans le dernier état cohérent avant la panne.

- **Efficacité des accès aux données :**

Les Entrées/Sorties disque reste problématique dans les systèmes de BD car une E/S disque coûtent quelques dizaines de millisecondes. L'interrogation et la mise à jour de la BD en utilisant des langages non procéduraux très puissants est une autre limite. Le SGBD prend en charge l'optimisation du coût et le nombre d'accès (E/S) d'une requête

Avant qu'une BDD prend sa forme finale (i.e. une forme utilisable par un SGBD), il faut passer par une étape de conception afin de décrire les objets de la réalité ainsi que les relations entre ces objets. Pour cela, la modélisation à travers des modèles est nécessaire.

## 4. Types de modèles de données

### 4.1. Modèle sémantique

Le modèle sémantique est parmi les modèles de bases de données les moins courants. Il comprend des informations sur la façon dont les données stockées sont rattachées au monde réel.

### 4.2. Modèle Entité-Association

Le modèle Entité-Association (EA) en français, ER en anglais (Entity Relationship) permet de décrire l'aspect conceptuel des données à l'aide d'entités et d'associations.

Le modèle entité-association, permet de modéliser des situations du monde réel décrites en langage naturel. Les noms correspondent aux entités, les verbes aux associations et les adjectifs ou compléments aux propriétés.

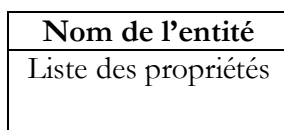
Par exemple : Le client a commandé un produit, ici deux entités sont identifiées : client et produit (voir Figure 1.6).

Le paragraphe suivant décrit les composant du diagramme entité/association.

### Entité

Une entité correspond à un objet du monde réel (matériel ou immatériel) défini en général par un nom (e.g., voiture, commande, etc.). Ces entités sont identifiables de manière unique, interagissent et font ou subissent des actions. La Figure 3 montre la représentation schématique d'une entité.

Par exemple : voiture, étudiant, patient, etc.



**Figure 1.3.** Modélisation d'une entité.

### Type d'entité

Le type d'entité désigne un ensemble d'entités ayant une sémantique et des propriétés communes (voir Figure 4).

Par exemple : Mohammed a prêté le livre d'Ahmed. Mohammed et Ahmed sont des entités de type entité Personne car ils ont les mêmes caractéristiques et le livre est une entité du type entité Livre. Notons que par abus de langage, on dit entité au lieu de type d'entité et c'est les types d'entité qui sont représentés dans le modèle et non pas les entités.



**Figure 1.4.** Modélisation d'un type entité.

### Attribut, valeur

Un attribut ou une propriété d'une entité ou d'une association caractérisée par un nom et un type.

Par exemples : nom d'une personne, titre d'un livre, puissance d'une voiture, etc.

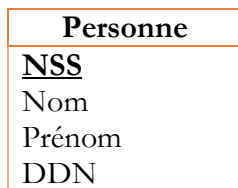
Chaque attribut possède un domaine qui définit l'ensemble des **valeurs** qui peuvent être choisies pour lui (entier, chaîne de caractères, booléen...). Au niveau de l'entité, chaque attribut possède une valeur compatible avec son domaine.

### Identifiant ou clé

Un identifiant (ou clé) d'un type entité ou d'un type association est l'ensemble minimum d'attributs qui identifient d'une manière unique une entité. Donc, il n'est pas possible qu'un identifiant d'un type

entité ou type association prene la même valeur pour deux entités (respectivement deux associations) distinctes.

Par exemple : numéro de client pour un client, code article pour un article, le code ISBN d'un livre pour un livre, numéro de sécurité sociale pour une personne.



**Figure 1.5.** Modélisation d'un type entité avec identifiant.

### Association

Une *association* (ou une relation) est un lien entre plusieurs entités.

Un *type association* (ou type relation) désigne un ensemble de relations qui possèdent les caractéristiques semblables et permet de décrire un lien entre plusieurs type entité.

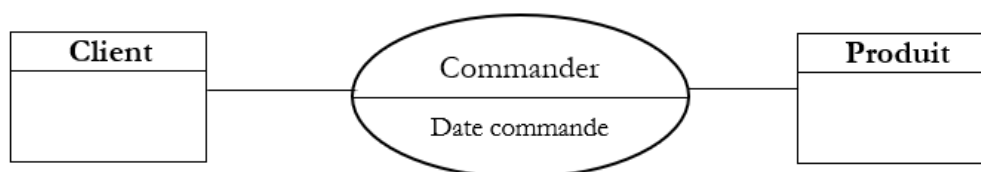
**Attention !** C'est par abus de langage qu'on utilise le mot association en lieu de type-association. Cependant, il ne faut pas confondre entre les deux concepts.

L'association est représentée par une ellipse. Une association peut avoir des propriétés particulières. Par exemple, l'association Commander a la propriété Date commande (voir figure 1.6).

Par exemple :

Type-association : un adhérent emprunte un livre.

Association : Ali a emprunté le livre des Bases de données, etc.



**Figure 1.6.** Exemple d'une association.

### Cardinalité

La cardinalité reliant un type d'association et un type d'entité représente le nombre de fois minimal et maximal de participations de chaque occurrence d'entité à une association.

La cardinalité minimale est 0 ou 1 et doit être inférieure ou égale à la cardinalité maximale et la cardinalité maximale est toujours 1 ou n. Donc, les cardinalités admises peuvent être expliquées comme suit :



- **0,n** : Une occurrence du type entité peut exister mais n'est pas impliquée dans aucune association et peut être impliquée, sans limitation, dans plusieurs associations.



Figure 1.7. Exemple de cardinalité 0,n.

- **1,1** : Une occurrence du type entité ne peut exister que si elle est impliquée dans exactement (au moins et au plus) une association.



Figure 1.8. Exemple de cardinalité 1,1.

- **0,1** : Une occurrence du type entité ne peut exister que si elle est impliquée dans au moins une association.



Figure 1.9. Exemple de cardinalité 0,1.

- **1,n** : Une occurrence du type entité ne peut exister que si elle est impliquée dans au moins une association.



Figure 1.10. Exemple de cardinalité 1,n.

### Dimension

La dimension d'un type-association est le nombre de types-entités qui y participent. On distingue :

- Les associations binaires qui relient 2 entités.
- Les associations ternaires qui relient 3 entités.
- Les associations n-aires qui relient plus de trois entités.
- Les associations réflexives qui relient une seule entité (e.g. Personne mariée à une personne).

### 4.3. Modèle hiérarchique

Le modèle hiérarchique (ou arbre) organise les données dans une structure arborescente, où chaque enregistrement n'a qu'un seul parent (racine). Les enregistrements frères et sœurs sont triés dans un ordre particulier. Ce modèle convient à la description de plusieurs relations du monde réel. Exemple : ADABASE (1970), System 2000 (1967).

4.4. Modèle réseau

Modèle réseau ou graphe est un modèle hiérarchique étendu qui autorise relations transverses (i.e. relations plusieurs-à-plusieurs entre des enregistrements liés). Un enregistrement peut être un membre ou un enfant dans plusieurs ensembles. Cela permet de traduire des relations complexes. Ex : TOTAL (1978).

4.5. Modèle relationnel

Dans le modèle relationnel, les informations décomposées et organisées sont stockées dans des tables. Exemple : 80% des SGBD sont relationnelles, ORACLE (85% du marché), DB2, SQL Server, ACCESS, etc. Le schéma relationnel est l'ensemble des RELATIONS qui modélisent le monde réel ; tel que les relations représentent les entités du monde réel (par exemple : des personnes, des objets, etc.) ou les associations entre ces entités.

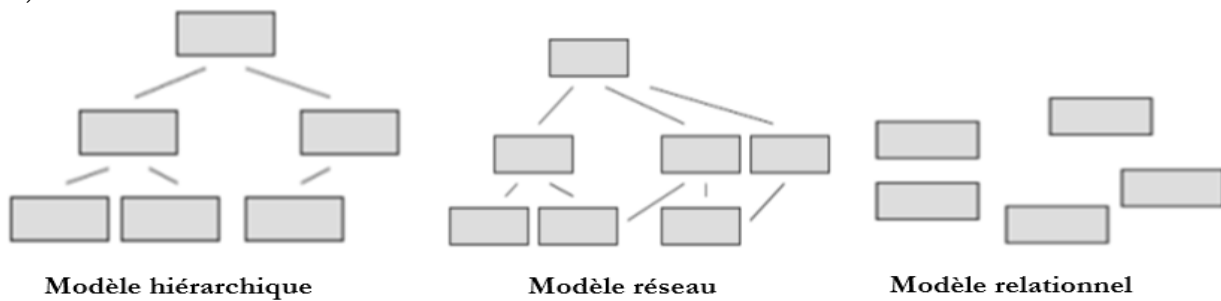


Figure 1.11. Modèles de données.

Le passage d'un schéma conceptuel E-A à un schéma relationnel se fait suivant des règles qui seront présentées dans le prochain chapitre.

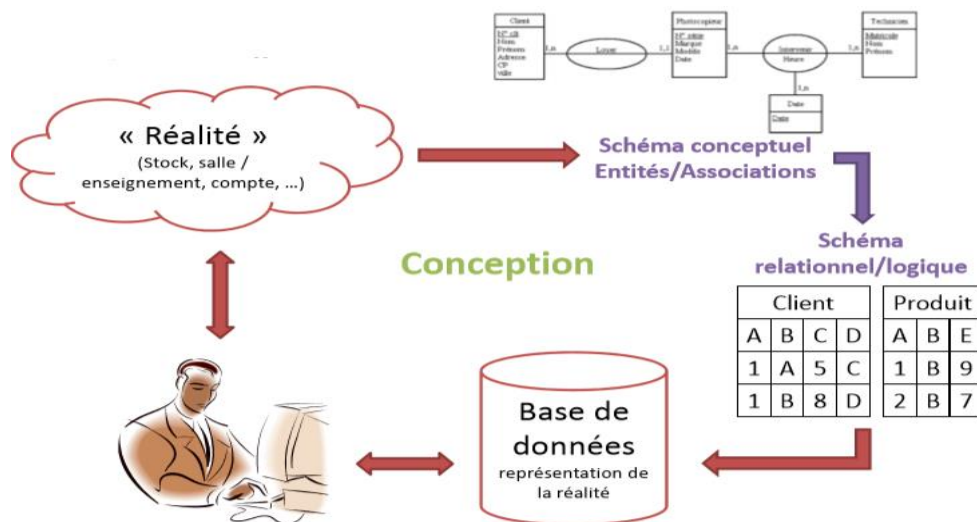


Figure 1.12. Transformation du modèle E/A au modèle relationnel [13].