

# Chapitre 1: Introduction aux bases de données et aux systèmes de gestion de base de données

## 1. Bases de données

### *Définition 1*

Une base de données (BD) est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour et recherche de données).

### *Définition 2*

Une base de données (BD) est une collection de données **persistantes** et **pertinentes** utilisées par des systèmes d'application de certaines **organisations**.

Trois mots clés sont utilisés dans cette définition :

- **données persistantes** : elles diffèrent d'autres données plus éphémères (de nature transitoire) telles que les données d'entrée et les données de sortie.
- **pertinentes** : données nécessaires et ciblées pour les différentes applications utilisatrices.
- **organisations** : peut être un simple individu avec une petite base de données privée (un médecin, un notaire, ...) ou une société complète avec une base de données partagée très importante (une banque, une université, un hôpital,...).

### **Avantages d'utilisation d'une base de données par rapport aux systèmes traditionnels (fichiers) :**

Dans le contexte classique, les fichiers sont conçus pour satisfaire les besoins d'une application. Il s'ensuit qu'une même information peut être représentée dans plusieurs fichiers. Si elle intervient dans deux applications, elle peut même se présenter différemment (format, codage,...). Cela entraîne un risque de conflit : lors d'une mise à jour par exemple, tous les duplicatas ne sont pas mis à jour en même temps par suite de leur non appartenance à l'application qui réalise cette mise à jour. L'utilisation de BD a les avantages suivants :

- **compacité** : plus besoin de fichiers manuels volumineux, fiches cartonnées par exemple où le risque de perte d'information est important,
- **rapidité** : recherche et mise à jour rapide d'information

- **données intégrées** : regroupement de plusieurs fichiers de données où toute redondance a été totalement ou partiellement éliminée. D'où la diminution de l'incohérence.

- **données partagées** : Plusieurs utilisateurs peuvent faire usage de ces données pour des buts divers. Ils peuvent également accéder aux mêmes données simultanément.

## 2. Système de Gestion de Base de Données (SGBD):

### 2.1. Définitions

#### *Définition1*

---

Un système de gestion de base de données (**SGBD**) est un logiciel qui permet la création et la manipulation de base de données.

C'est le logiciel qui prend en charge tous les accès à la BD.

#### *Définition2*

---

Un système de gestion de base de données (SGBD) est un ensemble de programmes qui permet la gestion et l'accès à une base de données. Il héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différentes.

### 2.2. Objectifs d'un SGBD :

#### **a) Définition des données**

permet de définir les données (schémas externes, schéma conceptuel , schéma interne et tous les liens correspondants) sous une forme non compilée et de les convertir dans la forme objet appropriée. Le SGBD doit donc être muni d'un langage de définition de données (LDD).

#### **b) Manipulation de données**

permet de traiter les requêtes de l'utilisateur pour interroger, modifier, supprimer ou insérer de nouvelles données.

Le SGBD doit être muni d'un langage de manipulation de données (**LMD**).

#### *Exemple*

---

SQL est un langage à la fois LDD et LMD

### **c) Non redondance**

La conception intégrée de la BD grâce à la notion de schéma de données (qui est une représentation globale des informations à l'aide de modèles de données pour un ensemble d'applications) évite les redondances constatées dans les SGF.

### **d) Cohérence**

Les données de la BD obéissent à des règles appelées contraintes d'intégrité (CI).

Une CI est une assertion que doit vérifier le SGBD à chaque fois que la donnée sur laquelle elle est définie est sollicitée (déchargement de l'utilisateur lors des opérations de création, modification, suppression).

#### *Exemple*

---

- spécialité licence a pour valeur ('ACAD', 'ISIL', 'GTR')
- une note est comprise entre 0 et 20

#### *Remarque*

---

Une BD cohérente est une BD dont les CI sont toujours vérifiées lors d'opérations sur la BD.

### **e) Efficacité des accès aux données**

Les accès aux données seront plus efficaces que dans les SGF grâce notamment :

- au développement d'index sophistiqués
- à l'existence de plusieurs chemins d'accès à une donnée
  - à l'existence de techniques d'optimisation de requêtes qui sélectionnent le chemin optimal à une donnée.

### **f) Administration centralisée**

C'est le rôle de l'administrateur de la BD qui a pour fonction de :

- Définir des structures de stockage,
- Définir des structures de données,
- Assurer le suivi et le contrôle de leur évolution.

### **g) Indépendance physique**

permet à l'administrateur de la BD de modifier l'organisation physique des données (ajouter une table d'index...) sans modification des programmes d'application déjà existants.

### **h) Indépendance logique**

permet de modifier le schéma conceptuel (ajouter de nouvelles classes d'objets, de nouvelles associations,...) sans modifier les programmes d'applications.

### ***i) Partageabilité***

Le SGBD doit permettre à plusieurs applications de partager les données. Pour cela, il doit gérer les conflits d'accès (les détecter et les solutionner) . Ceci est possible grâce à la notion de **transaction** et de définition d'**algorithmes de gestion de la concurrence**.

#### *Remarque*

---

Une transaction est une unité de traitement séquentiel exécutée pour le compte d'un utilisateur. C'est une suite finie d'actions (opérations) portant sur des objets de la BD dont l'exécution la maintient dans un état cohérent.

#### *Exemple de transaction*

---

Opération de transfert d'une somme d'argent X d'un compte bancaire C1 à un compte C2.

### ***j) Sécurité et confidentialité***

Les données doivent être protégées contre les pannes et contre les accès mal intentionnés.

- Protection contre les pannes : On distingue deux types de pannes

- Pannes simples caractérisées par la perte du contenu de la mémoire centrale
- Pannes graves caractérisée par la perte du contenu des mémoires secondaires

Dans les deux cas, il est possible de récupérer un état cohérent des données grâce à l'existence et à la gestion de journaux de transactions, qui gardent la trace d'exécutions antérieures.

-Protection contre les accès mal intentionnés : La définition de droits d'accès, de mots de passe etc. gérés par le SGBD permet la confidentialité des données de la base et leur inviolabilité.

### 3. Le modèle relationnel

Les bases de données sont apparues à la fin des années 60, à une époque où la nécessité d'un système de gestion de l'information souple se faisait ressentir. Il existe plusieurs modèles de SGBD, différenciés selon leurs représentations des données ou modèles de données.

#### Exemples de modèles de données:

- Les modèles de 1ère génération (années 60) : Hiérarchique et Réseau
- Les modèles de 2ème génération (années 70-80): **Relationnel**, Entités/Association (E/A), ...
- Les modèles de 3ème génération (décennie 90): Modèle orienté objet

Dans le **modèle relationnel**, les données sont représentées sous forme de tables à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations. Ce modèle est appelé "relationnel" car le terme mathématique **relation** est le terme le plus approprié pour une table.

#### Exemple :

Etudiant				
Nom	Prénom	Matricule	Groupe	Section

Module	
Cod-Mod	Coefficient

Notes			
Matricule	Cod-Module	EMD1	EMD2

### Terminologie du modèle relationnel :

Une relation correspond à une table ayant un schéma et une extension. Un n-uplet (tuple) correspond à une ligne d'une table et un attribut à une colonne.

#### Exemple :

Schéma de la relation Etudiant :

Etudiant ( Matricule, Nom, Prénom, Groupe, Section)

Extension de la relation Etudiant :

Matricule	Nom	Prénom	Groupe	Section
23564	kaci	Ali	1	A
23654	Madi	Reda	1	A
	.....			

Attributs

Etudiant

un tuple

Domaine

**Scalaire** : c'est la plus petite unité sémantique de données (valeur atomique) telle que la valeur du matricule d'un étudiant, de son nom, prénom ... Par exemples, 23564 est un scalaire, Ali, 1, A, Madi sont des scalaires...

**Domaine** : ensemble donné de valeurs scalaires, toutes de même type. Par exemple, le domaine des matricules des étudiants

**Attribut** : champ définissant une propriété. Il doit être défini sur exactement un seul domaine : les valeurs des attributs doivent être prises dans ce domaine. Dans les langages de programmation, on définit un domaine par un type de données.

**Relation** : Une relation  $R$ , sur un ensemble de domaines  $D_1, D_2, \dots, D_n$  (non nécessairement distincts) est constituée de deux parties, un en-tête (schéma de la relation) et un corps (ensemble de lignes).

L'en-tête est un ensemble fixé d'attributs  $A_1, A_2, \dots, A_n$ , tel que chaque attribut  $A_i$  est défini sur un domaine  $D_j$  ( $j = 1, 2, \dots, n$ ). Les noms des attributs  $A_1, A_2, \dots, A_n$  sont tous distincts.

Le corps consiste en un sous-ensemble de tuples ou n-uplets.

**Tuple** : (appelé aussi uplet ou enregistrement) correspond à une ligne d'une relation

**Cardinalité** : c'est le nombre de tuples d'une relation.

**Degré** : c'est le nombre d'attributs d'une relation.