

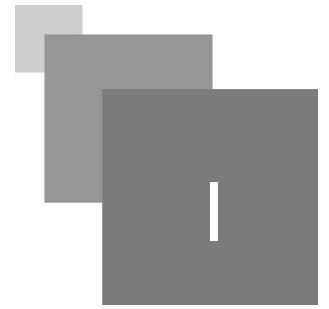
# Bases De Données

# Table of contents



<b>I - Chapitre 1 : Concepts Fondamentaux</b>	<b>3</b>
1. Objectif .....	3
2. Introduction .....	3
3. Données, Bases de données et SGBD .....	4
4. Que doit-on savoir pour utiliser un SGBD ? .....	6
4.1. Définition du schéma de données .....	7
4.2. Les opérations sur les données .....	8
4.3. Optimisation .....	9
4.4. Concurrence d'accès .....	9

# Chapitre 1 : Concepts Fondamentaux



Objectif	3
Introduction	3
Données, Bases de données et SGBD	4
Que doit-on savoir pour utiliser un SGBD ?	6

## 1. Objectif

A l'issue de ce cours l'apprenant sera capable d'*identifier* les principales fonctions d'un SGBD en se basant sur son architecture.

## 2. Introduction

Les BDD sont nées à la fin des années 1960 pour combler les lacunes des systèmes de fichiers et faciliter la gestion qualitative et quantitative des données informatiques. Les SGBD sont des applications informatiques permettant de créer et de gérer des BDD (comme Oracle ou PostgreSQL par exemple)

Les BDD relationnelles, issues de la recherche de Codd, sont celles qui ont connu le plus grand essor depuis les années, et qui reste encore aujourd'hui les plus utilisées. On utilise des SGBDR pour les implémenter. Le langage SQL est le langage commun à tous les SGBDR, ce qui permet de concevoir des BD relativement indépendamment des systèmes utilisés.

Les usages de BDD se sont aujourd'hui généralisés pour entrer dans tous les secteurs de l'entreprise, depuis les petites bases utilisées par quelques personnes dans un service pour des besoins de gestion de données locales, jusqu'aux bases qui gèrent de façon centralisée des données partagées par tous les acteurs de l'entreprise.

L'accroissement de l'utilisation du numérique comme outil de manipulation de toutes données (bureautique, informatique applicative, etc.) et comme outil d'extension des moyens de communication (réseaux) ainsi que les évolutions technologiques (puissance des PC, Internet,



etc.) ont rendu indispensable, mais aussi complexifié la problématique des BD.

Les conséquences de cette généralisation et de cette diversification des usages se retrouvent dans l'émergence de solutions conceptuelles et technologiques nouvelles, les bases de données du mouvement NoSQL particulièrement utilisées par les grands acteurs du web.

### 3. Données, Bases de données et SGBD

La première chose à faire est d'établir quelques points de terminologie. Qu'est-ce qu'une donnée ? C'est une information quelconque comme, par exemple : voici une personne, elle s'appelle Jean. C'est aussi une relation entre des informations: Jean enseigne les bases de données. Des relations de ce genre définissent

des structures. Une BDD est *un ensemble, en général volumineux, de telles informations, avec une caractéristique essentielle : on souhaite les mémoriser de manière permanente*. D'où les définitions :

#### Definition

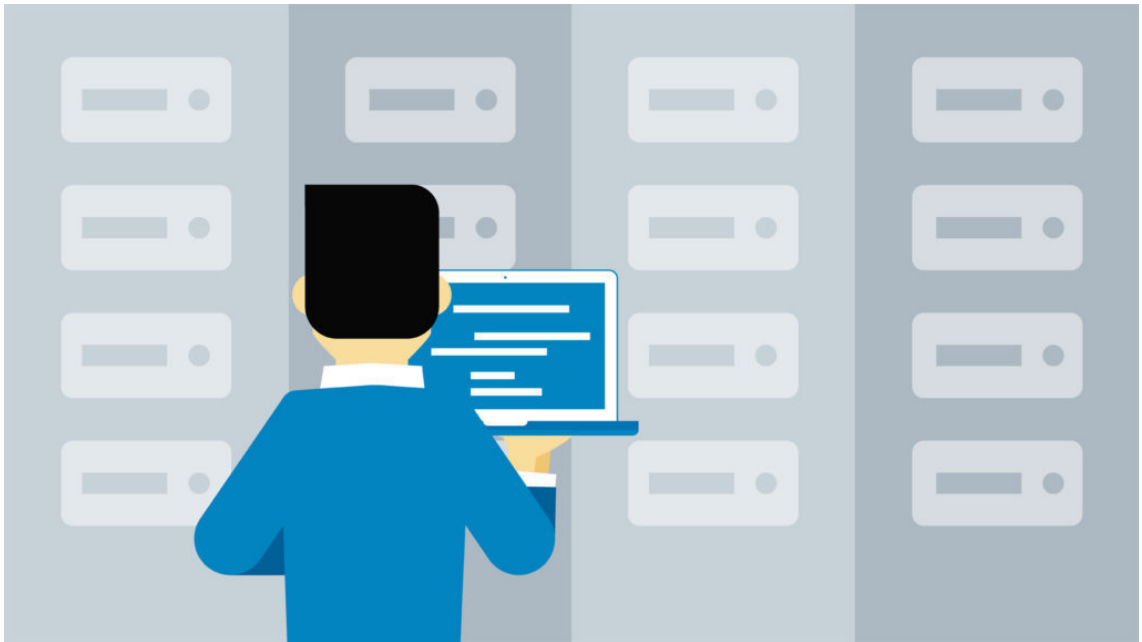
---

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

On peut remarquer qu'une organisation consistant en un (ou plusieurs) fichier(s) stockés sur mémoire secondaire est conforme à cette définition. Un ensemble de fichiers ne présentant qu'une complexité assez faible, il n'y aurait pas là matière à longue dissertation. Malheureusement l'utilisation directe de fichiers soulève de très gros problèmes:

1. Lourdeur d'accès aux données. En pratique, pour chaque accès, même le plus simple, il faudrait écrire un programme.
2. Manque de sécurité. Si tout programmeur peut accéder directement aux fichiers, il est impossible de garantir la sécurité et l'intégrité des données.
3. Pas de contrôle de concurrence. Dans un environnement où plusieurs utilisateurs accèdent aux mêmes fichiers, des problèmes de concurrence d'accès se posent.

D'où le recours à un logiciel chargé de gérer les fichiers constituant une base de données, de prendre en charge les fonctionnalités de protection et de sécurité et de fournir les différents types d'interface nécessaires à l'accès aux données. Ce logiciel (le SGBD) est très complexe et fournit le sujet principal de ce cours. En particulier, une des tâches principales du SGBD est de masquer à l'utilisateur les détails complexes et fastidieux liés à la gestion de fichiers. D'où la définition.



Base de données

### Definition

Un *Système de Gestion de Bases de Données (SGBD)* est un logiciel de haut niveau qui permet de manipuler les informations stockées dans une base de données.

La complexité d'un SGBD est essentiellement issue de la diversité des techniques mises en oeuvre, de la multiplicité des composants intervenant dans son architecture, et des différents types d'utilisateurs (administrateurs, programmeurs, non informaticiens, ...) qui sont confrontés, à différents niveaux, au système.

Voici quelques exemples illustrant tous les cas de figure qu'il faudrait envisager dans un cours exhaustif:

- Les modèles de données: entité-relation, réseau, hiérarchique, relationnel, orienté-objet, modèles sémantiques.
- Les langages de requêtes: fondements théoriques (logiques du premier ordre, du point fixe, algèbres diverses) et les langages comme SQL, SQL3, Datalog, OQL, etc.
- Les techniques de stockage : sur disque (optique), sur bande.
- L'organisation des fichiers: index, arbre-B, hachage, ...
- L'architecture : centralisé, distribué, sur d'autres bases accessibles par réseau.
- Les techniques d'évaluation et d'optimisation de requêtes.
- La concurrence d'accès et les techniques de reprise sur panne.

Pour mettre un peu d'ordre dans tout cela, on peut se raccrocher à une architecture standard conforme à la plus grande partie des SGBD existant, et offrant l'avantage de bien illustrer les principales caractéristiques d'un SGBD.

Cette architecture distingue trois niveaux correspondant d'une part à trois représentations équivalentes

de l'information, d'autre part aux champs d'interventions respectifs des principaux acteurs. Pour ces derniers, nous utiliserons la terminologie suivante :

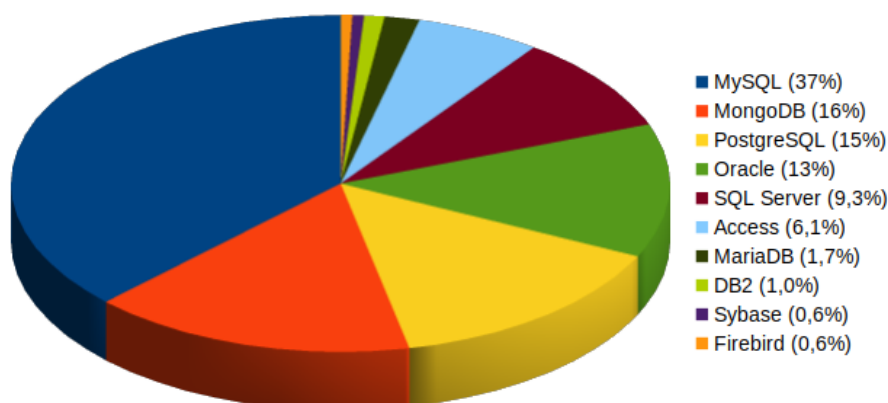
- Utilisateur naïf: du non spécialiste des SGBD au non informaticien.
- Concepteur et programmeur d'application : à partir des besoins des différents utilisateurs, écrit l'application pour des utilisateurs "naïfs".
- Utilisateur expert: informaticien connaissant le fonctionnement interne d'un SGBD et chargé d'administrer la base.

Chaque niveau du SGBD remplit (réalise) un certain nombre de fonctions:

- Niveau physiques: gestion sur mémoire secondaire (fichiers) des données, du schéma, des index ; Partage de données et gestion de la concurrence d'accès; Reprise sur pannes (fiabilité); Distribution des données et interopérabilité (accès aux réseaux).
- Niveau logique : Définition de la structure de données: Langage de Description de Données (LDD) ; Consultation et Mise à Jour des données: Langages de Requêtes (LR) et Langage de Manipulation de Données (LMD) ; Gestion de la confidentialité (sécurité); Maintien de l'intégrité ;
- Niveau externe : Vues; Environnement de programmation (intégration avec un langage de programmation); Interfaces conviviales et Langages de 4e Génération (L4G) ; Outils d'aides (e.g. conception de schémas); Outils de saisie, d'impression d'états.

#### Reminder

En résumé, un SGBD est destiné à gérer un gros volume d'informations, persistantes (années) et fiables (protection sur pannes), partageables entre plusieurs utilisateurs et/ou programmes et manipulées indépendamment de leur représentation physique



## 4. Que doit-on savoir pour utiliser un SGBD ?

Définition du schéma de données	7
Les opérations sur les données	8
Optimisation	9
Concurrence d'accès	9

L'utilisation d'un SGBD suppose de comprendre (et donc de savoir utiliser) les fonctionnalités suivantes:

Définition du schéma de données en utilisant les modèles de données du SGBD.

Opérations sur les données: recherche, mises-à-jour, etc.

Partager les données entre plusieurs utilisateurs. (Mécanisme de transaction).

Optimiser les performances, par le réglage de l'organisation physique des données. Cet aspect relève plutôt de l'administration et ne sera évoqué que dans l'introduction.

Reprenons dans l'ordre ces différents points.

See "SYSTEMES DE GESTION DE BASES DE DONNEES"

#### 4.1. Définition du schéma de données

Un schéma est simplement la description des données contenues dans la base. Cette description est conforme à un modèle de données qui propose des outils de description (structures, contraintes et opérations). En fait, dans un SGBD, il existe plusieurs modèles plus ou moins abstraits des mêmes objets, e.g. :

- Le modèle conceptuel : la description du système d'information
- Le modèle logique : interface avec le SGBD
- Le modèle physique : fichiers.

Ces différents modèles correspondent aux niveaux dans l'architecture d'un SGBD. Prenons l'exemple du modèle conceptuel le plus courant : le modèle Entité/Association. C'est essentiellement une description très abstraite qui présente les avantages suivants:

- l'analyse du monde réel
- la conception du système d'information
- la communication entre différents acteurs de l'entreprise

En revanche, il ne propose pas d'opérations. Or définir des structures sans disposer d'opérations pour agir sur les données stockées dans ces structures ne présente pas d'intérêt pratique pour un SGBD. D'où, à un niveau inférieur, des modèles dits "logiques" qui proposent

1. Un langage de définition de données (LDD) pour décrire la structure, incluant des contraintes.
2. Un langage de manipulation de données (LMD) pour appliquer des opérations aux données.

Ces langages sont abstraits: le LDD est indépendant de la représentation physique des données, et le LMD est indépendant de l'implantation des opérations. On peut citer une troisième caractéristique : outre les structures et les opérations, un modèle logique doit permettre d'exprimer des contraintes d'intégrité sur les données.

**Example**

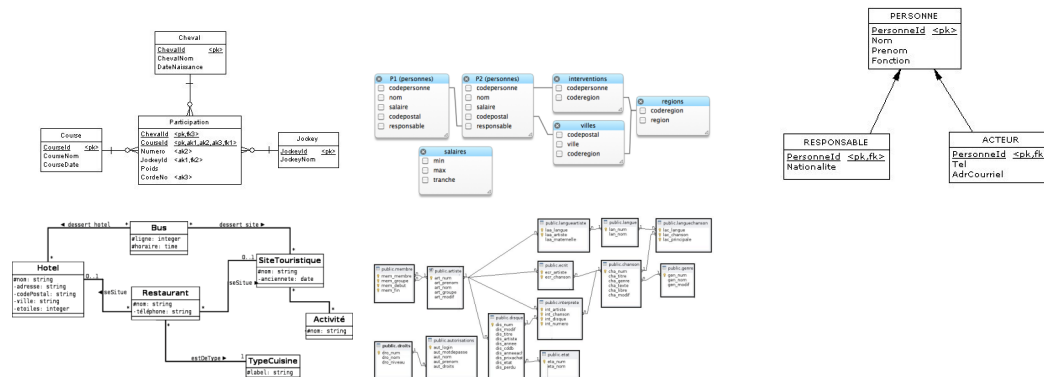
```

1 nom character 15, not null;
2 âge integer between 0 and 120;
3 débit = crédit;
4 ...
    
```

Bien entendu, le SGBD doit être capable de garantir le respect de ces contraintes. Quand on conçoit une application pour une BD, on tient compte (plus ou moins consciemment) de cette architecture en plusieurs niveaux. Typiquement : On décide la structure logique, on décide la structure physique, on écrit les programmes d'application en utilisant la structure logique, enfin Le SGBD se charge de transcrire les commandes du LMD en instructions appropriées appliquées à la représentation physique. Cette approche offre de très grands avantages qu'il est important de souligner. Tout d'abord elle ouvre l'utilisation des SGBD à de utilisateurs non-experts: les langages proposés par les modèles logiques sont plus simples, et donc plus accessibles, que les outils de gestion de fichiers. Ensuite, on obtient une caractéristique essentielle : l'indépendance physique. On peut modifier l'implantation physique sans modifier les programmes d'application. Un concept voisin est celui d'indépendance logique : on peut modifier les programmes d'application sans toucher à l'implantation.

Enfin le SGBD décharge l'utilisateur, et en grande partie l'administrateur, de la lourde tâche de contrôler la sécurité et l'intégrité des données

*Modèles logiques de données et modèles conceptuels de données*



**4.2. Les opérations sur les données**

Il existe 4 opérations classiques (ou requêtes):

1. La création (ou insertion).
2. La modification (ou mise-à-jour).
3. La destruction.
4. La recherche.

Ces opérations correspondent à des commandes du LMD. La plus complexe est la recherche en raison de la variété des critères.



Pour l'utilisateur, une bonne requête a les caractéristiques suivantes. Tout d'abord elle s'exprime facilement : l'idéal serait de pouvoir utiliser le langage naturel, mais celui-ci présente trop d'ambiguïtés. Ensuite le langage ne devrait pas demander d'expertise technique (syntaxe compliquée, structures de données, implantation particulière ...). Il est également souhaitable de ne pas attendre trop longtemps (à charge pour le SGBD de fournir des performances acceptables). Enfin , et peut-être surtout, la réponse doit être fiable.

Une bonne partie du travail sur les SGBD consiste à satisfaire ces besoins. Le résultat est ce que l'on appelle un langage de requêtes, et constitue à la fois un sujet majeur d'étude et une caractéristique essentielle de chaque SGBD. Le langage le plus répandu à l'heure actuelle est SQL.

### 4.3. Optimisation

L'optimisation (d'une requête) s'appuie sur l'organisation physique des données. Les principaux types d'organisation sont les fichiers séquentiels, les index (denses, secondaires, arbres B) et le regroupement des données par hachage.

Un module particulier du SGBD, l'optimiseur, tient compte de cette organisation et des caractéristiques de la requête pour choisir le meilleur séquençement des opérations.

### 4.4. Concurrence d'accès

Plusieurs utilisateurs doivent pouvoir accéder en même temps aux mêmes données. Le SGBD doit savoir:

- Gérer les conflits si les deux font des mises-à-jour.
- Offrir un mécanisme de retour en arrière si on décide d'annuler des modifications en cours.
- Donner une image cohérente des données si l'un fait des requêtes et l'autre des mises-à-jour.

Le but : éviter les blocages, tout en empêchant des modifications anarchiques