

Chapitre 1

Généralités sur les bases de données

I. Définition d'un SGBD

Une **base de données**, généralement appelée 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, recherche de données) et accessibles de façon sélective par plusieurs utilisateurs.

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le **Système de gestion de base de données (SGBD)**. Un SGBD héberge généralement plusieurs bases de données, qui sont destinées à des logiciels différents ou des thématiques différentes.

II. Fonctions d'un SGBD

D'une manière générale, un SGBD doit avoir les caractéristiques suivantes :

- **Indépendance physique** : Le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations.
- **Indépendance logique** : le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs.
- **Manipulabilité** : des personnes ne connaissant pas la base de données doivent être capables de décrire leurs requêtes sans faire référence à des éléments techniques de la base de données. Ceci peut se faire à travers un dictionnaire de données ou un catalogue système.
- **Rapidité des accès** : le système doit pouvoir fournir les réponses aux requêtes (recherches d'informations à partir d'une ou plusieurs bases de données) le plus rapidement possible, cela implique des algorithmes de recherche rapides.
- **Différents langage d'accès** : le SGBD doit au moins supporter un langage adressant les concepts du modèle (par exemple, dans le cas du modèle relationnelle, ce langage est SQL). Néanmoins, ce type de langage ne permet pas tous les types de manipulation. Par conséquent, soit les SGBD proposent un langage plus complet avec la possibilité de définir des accès à la base de données, soit ils proposent un couplage d'un langage de type SQL avec un langage de programmation conventionnel.
- **Administration centralisée** : le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier son intégrité de façon centralisée.
- **Limitation de la redondance** : le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage d'espace mémoire mais aussi des erreurs.
- **Vérification de l'intégrité** : les données doivent être cohérentes entre elles, de plus lorsque des éléments font références à d'autres, ces derniers doivent être présents.
- **Partage des données** : le SGBD doit permettre à plusieurs utilisateurs d'accéder à la base de données de manière simultanée et transparente.
- **Sécurité des données** : le SGBD doit permettre de spécifier qui a le droit d'accéder ou de modifier tout ou une partie d'une base de données. Il faut donc présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs afin de se prémunir contre les manipulations illicites intentionnelles ou accidentelles.
- **Résistance aux pannes** : le SGBD doit garantir la cohérence de l'information et des traitements en cas de panne. Les opérations sur les bases de données pouvant être très longues, il faut fournir un mécanisme de reprise en cas de panne matérielle ou logicielle, intentionnelle ou fortuite.

- **Capacité de stockage élevée :** le SGBD doit permettre de gérer des données très volumineuses, pouvant atteindre plusieurs milliards d'octets. Les unités de stockage sont passées du mégaoctet Mo (10^6 octets) au gigaoctet Go (10^9), puis au téraoctet To (10^{12}), puis petaoctet Po (10^{16}), voir même au exaoctet Eo (10^{18}) et zettaoctet Zo (10^{21}).

III. Avantages de l'utilisation des SGBD

Avant l'arrivée des bases de données (et encore dans beaucoup de cas aujourd'hui), chaque application écrite pour un organisme travaillait avec ses propres fichiers.

Une même information (l'adresse d'un client par exemple) peut alors être enregistrée dans plusieurs fichiers disjoints. Ceci occasionne des délais de mise à jour et peut amener les différentes applications à travailler sur des données contradictoires.

Quand la gestion se fait avec une base de données centralisée (centralisée « logiquement » mais pas nécessairement physiquement si la base de données est répartie sur plusieurs sites), chaque donnée n'est enregistrée qu'en un seul endroit de la base et il ne peut y avoir ce genre de problèmes. Cette centralisation facilite donc le maintien de l'intégrité des données.

Les facilités offertes par les SGBD pour contrôler l'accès des utilisateurs aux données de la base et les reprises automatisées après incident accroissent la sécurité dans le traitement des données.

Les SGBD offrent aussi des instructions très puissantes pour le traitement des données : un seul ordre *SELECT* du langage SQL peut correspondre à des dizaines de lignes de programmation dans un langage de troisième génération comme le langage C. La productivité des programmeurs est ainsi fortement augmentée.

En plus de l'indépendance des programmes vis-à-vis de l'implantation physique des données, d'autres avantages importants sont apportés par l'utilisation des SGBD évolués des dernières générations (en particulier par les SGBD relationnels). Ces SGBD offrent :

- L'indépendance des programmes vis-à-vis de la structure logique des données (stratégie d'accès aux données, manière de regrouper les données,...)
- L'interrogation directe de la base par les utilisateurs dans un langage non procédural.

Ces différents points facilitent grandement la maintenance des applications et permettent plus de souplesse pour le traitement des données enregistrées.

IV. Niveaux de description des bases de données

Trois niveaux de description des données ont été définis par la norme ANSI/SPARC.

- **Niveau interne :** description du stockage des données au niveau des unités de stockage, des fichiers, ... On appelle cette description le schéma interne.
- **Niveau conceptuel :** description de la structure de toutes les données qui existent dans la base, description de leurs propriétés (relations qui existent entre elles) c'est-à-dire de leur sémantique inhérente, sans soucis d'implémentation physique ni de la façon dont chaque groupe de travail voudra s'en servir. On appelle cette description le schéma conceptuel.
- **Niveau externe :** description pour chaque utilisateur de sa perception des données. On appelle cette description le schéma externe ou vue.

Ce cours abordera plus particulièrement le niveau conceptuel.

V. Type d'utilisateurs d'une base de données

L'**administrateur** de la base est chargé du contrôle de la base de données. Il est chargé de permettre l'accès aux données aux applications ou individus qui y ont droit et de conserver de bonnes performances d'accès à ces données. Il est aussi chargé des sauvegardes et des procédures de reprise après panne.

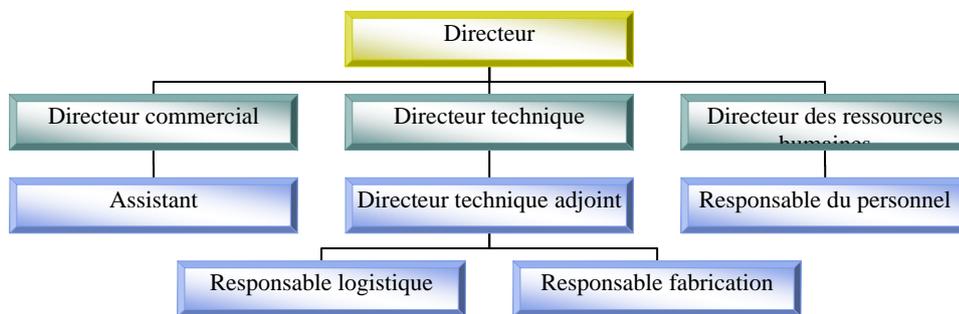
Le **programmeur** d'applications utilise la base de données pour construire ses applications. Il a droit de créer de nouvelles tables et les structures associées (vue, index, cluster,...). Il définit avec l'administrateur de la base les droits qui seront accordés aux utilisateurs des applications qu'il développe.

L'**utilisateur** final n'a accès qu'aux données qui lui sont utiles. L'administrateur de la base peut lui accorder certains droits : consultation, modification, suppression des données. En général, il n'a pas le droit de créer de nouvelles tables ni d'ajouter ou d'enlever des index.

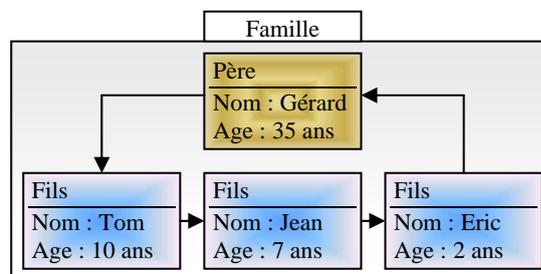
VI. Les différents types de SGBD

Le type d'un SGBD est caractérisé par le type du modèle de données qu'il implémente. Il existe ainsi **cinq types de modèles de données** :

- **Le modèle hiérarchique** : les données sont stockées hiérarchiquement, selon une arborescence descendante. Ce modèle s'apparente à l'organisation des répertoires d'un ordinateur. Ce type de SGBD est particulièrement adapté à la modélisation de nomenclatures. Si le principe de relation « 1 vers N » n'est pas respecté, alors la hiérarchie doit être transformée en un réseau. Les structures de données hiérarchiques ont été très utilisées dans les premiers SGBD car très simple d'implémentation.



- **Le modèle réseau** : ce modèle est une extension du modèle précédent, il est alors possible d'établir des relations « 1 vers N » en définissant des associations entre tous les types d'enregistrements. Cependant, pour retrouver une donnée dans une telle modélisation, il faut connaître le chemin d'accès, ce qui rend les programmes encore dépendants de la structure de données.



- **Le modèle déductif** : les données sont modélisées dans des tables, mais leur manipulation se fait par calcul de prédicats. C'est le modèle le moins courant.
- **Le modèle objet (SGBDOO, Système de Gestion de Bases de Données Orienté Objet)** : les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées classes présentant des données membres qui les décrivent et représentent leur état. Les objets contiennent aussi la logique qui permet de les utiliser et de les modifier. Tous ces objets sont classés hiérarchiquement dans une base de données à objets.
- **Le modèle relationnel (SGBDR, Système de Gestion de Bases de Données Relationnelles)** : les données sont modélisées dans des tables ou relations. La manipulation de ces données se fait selon la théorie mathématique des relations.

PERSONNE				
ID	Nom	Prénom	DateNaiss	VilleNaiss
1	Dupont	Bob	01/01/1950	1
2	Durand	Gérard	29/04/1998	3
3	Marchand	Daniel	26/12/2005	1

VILLE				
ID	Nom	Population	Superficie	Région
1	Paris	2 153 600	105.40 km ²	2
2	Lyon	466 400	47.95 km ²	3
3	Marseille	820 900	260.62 km ²	4

La souplesse apportée par cette représentation et les études théoriques appuyés sur la théorie mathématique des relations ont permis le développement de langages puissants non procéduraux. Dans ces langages, l'utilisateur ou le programmeur indique quelles informations il veut obtenir et c'est le SGBD qui trouve la manière d'arriver au résultat. Le programme ou l'utilisateur n'a plus à naviguer dans la base pour retrouver ses données. Ces langages peuvent être utilisés par des non-informaticiens et permettent l'écriture de programmes indépendants de la structure logique et physique des données. Le langage SQL est un standard parmi tous ces langages.

L'inventeur du modèle relationnel est Codd qui travaillait dans les laboratoires d'IBM. Il a énoncé douze règles que doivent vérifier les SGBD pour être relationnels. En fait, aucun SGBD ne respecte actuellement toutes ces règles. Certains cependant s'approchent de cette « perfection » relationnelle.

Voici ces douze règles (certaines notions seront développées plus tard dans ce cours) :

- Règle 1 : toutes les informations sur les données sont représentées au niveau logique et non physique (pas besoin d'informations sur la façon dont sont enregistrées physiquement les données).
- Règle 2 : les données sont accessibles uniquement par la combinaison du nom de la table, de la clé primaire et du nom de la colonne (pas de chemin à donner).
- Règle 3 : une valeur spéciale doit représenter l'absence de valeur (valeur NULL).
- Règle 4 : la description de la base de données doit être accessible comme les données ordinaires (un dictionnaire des données est enregistré dans la base).
- Règle 5 : un langage doit permettre de définir les données, définir des vues (visions particulières de la base, enregistrées comme des relations), manipuler les données, définir les contraintes d'intégrité, des autorisations et gérer des transactions.
- Règle 6 : on peut faire des mises à jour par les vues lorsque c'est logiquement possible.
- Règle 7 : le langage doit comporter des ordres effectuant l'insertion, la mise à jour et la suppression de données (un seul ordre pour effectuer chacune des fonctions).
- Règle 8 : indépendance des programmes vis-à-vis de l'implantation physique des données.
- Règle 9 : indépendance des programmes vis-à-vis de l'implantation logique des données (si les informations manipulées par les programmes n'ont pas été modifiées ou supprimées).
- Règle 10 : les contraintes d'intégrité doivent pouvoir être définies dans le langage relationnel et enregistrées dans le dictionnaire des données.
- Règle 11 : indépendance vis-à-vis de la répartition des données sur divers sites.
- Règle 12 : on ne peut jamais contourner les contraintes (d'intégrité ou de sécurité) imposées par le langage du SGBD en utilisant un langage de plus bas niveau (par exemple le langage C).

A la fin des années 90 les bases relationnelles sont les bases de données les plus répandues (environ trois quarts des bases de données).