



Sommaire

1	Définitions.....	1
2	Historique.....	2
2.1	L'organisation en fichier.....	2
2.2	L'apparition des SGBD.....	2
2.3	Les SGBD relationnels.....	3
2.4	Les bases de données objet.....	3
2.5	Les SGBR actuels.....	3
2.5.1	Les SGBD commerciaux :.....	3
2.5.2	Les SGBDR libres.....	4
2.5.3	Le poids des acteurs du marché.....	4
3	Les objectifs d'un SGBDR.....	4
4	L'organisation d'un SGBDR.....	5
4.1	Notion de tables.....	5
4.2	Les types de données de base.....	5
4.2.1	Le type numérique.....	5
4.2.2	Le type texte.....	6
4.2.3	Le type Date/heure.....	6
4.3	Des types particuliers.....	6
4.3.1	Le type monétaire.....	6
4.3.2	Le type NuméroAuto.....	6
4.3.3	Le type Oui/Non.....	6
4.3.4	Le type Lien hypertexte.....	6
4.3.5	Le type mémo.....	6
4.3.6	Le type OLE.....	7
4.4	La notion d'intégrité.....	7
5	Rappel sur le modèle relationnel.....	7
5.1	Représentation du modèle relationnel.....	7
5.1.1	La représentation textuelle :.....	8
5.1.2	La représentation graphique.....	8
5.1.3	La représentation Access.....	8

1 Définitions

Une base de données contient l'ensemble des données informatisées d'un système d'information. Cette base est implantée physiquement sur le disque d'un ordinateur sous la forme d'un ou plusieurs fichiers. Le logiciel spécialisé dans la gestion d'une base de données s'appelle un SGBDR (système de gestion de bases de données relationnelles).

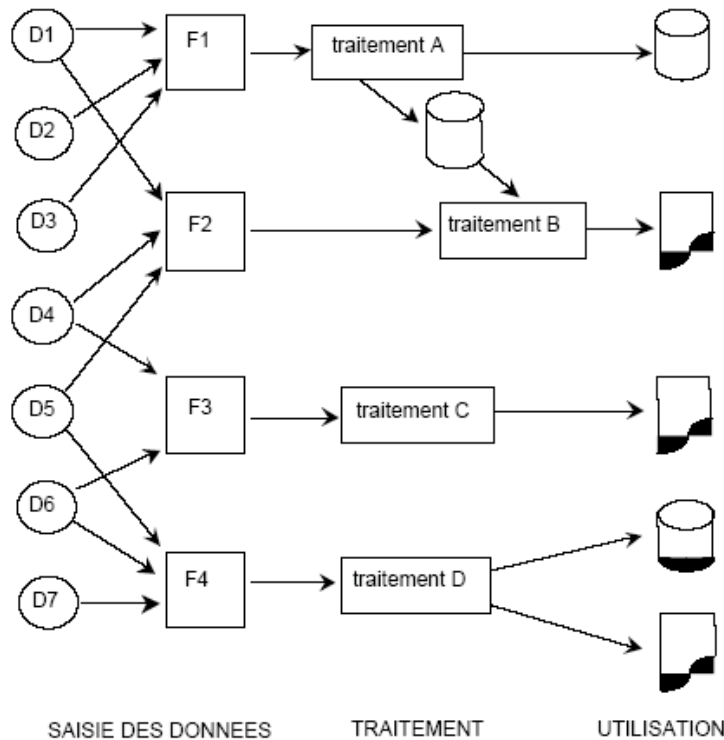
Un SGBD représente donc l'ensemble des programmes assurant structuration, stockage, maintenance, mise à jour et recherche des données d'une base + interfaces nécessaires aux différentes formes d'utilisation de la base.



2 Historique

2.1 *L'organisation en fichier*

Jusqu'aux années 60 : Les données étaient organisées en fichiers.

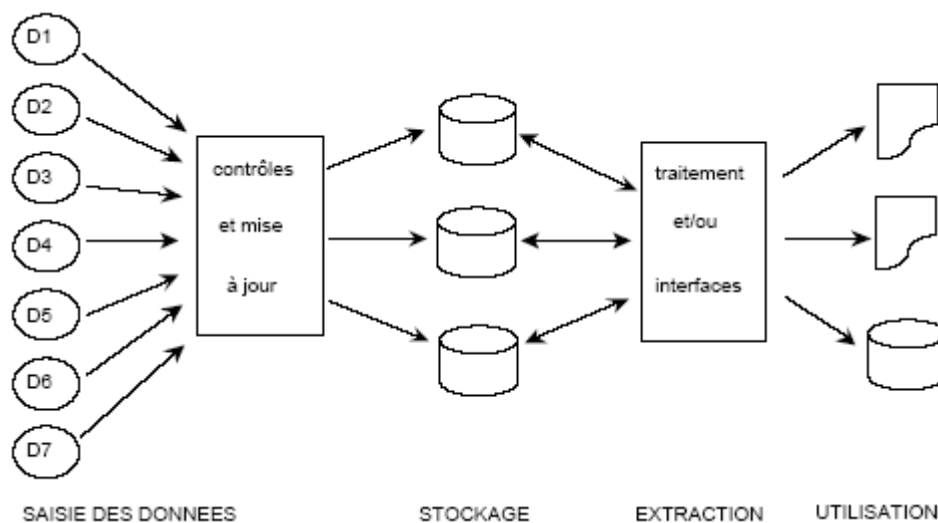


Les points particuliers de cette organisation étaient :

- Particularisation de la saisie en fonction des fichiers (un programme de mise à jour pour chaque fichier)
- Contrôle en différé des données → traitement à posteriori des erreurs, d'où augmentation des délais et du risque d'erreur
- Particularisation des fichiers en fonction des traitements
- Grande redondance des données
- Peu de standardisation dans les traitements
- Utilisation de nombreux supports intermédiaires

2.2 *L'apparition des SGBD*

A la fin des années 60 sont apparus les premiers SGBD qui organisaient la séparation de la description des données de leur manipulation par les programmes d'application : SOCRATE, TOTAL, IDMS, IDS2, IMS2.



Les points particuliers de cette nouvelle organisation étaient :

- Uniformisation de la saisie
- Contrôle immédiat de la validité des données saisies
- Partage d'une même structure de données entre plusieurs traitements
- Limitation de la redondance des données
- Standardisation des traitements généraux (consultation, restitution sous forme de listes ou tableaux)
- Gestion rationnelle des supports

2.3 Les SGBD relationnels

À partir de 1970 apparaît la deuxième génération de SGBD à partir du modèle relationnel. Les principales évolutions sont l'enrichissement et la simplification des SGBD afin de faciliter l'accès aux données pour les utilisateurs.

Commercialisation depuis 1982 : ORACLE, INGRES, SYBASE, INFORMIX, DB2, RDB.

2.4 Les bases de données objet

Il s'agit de la troisième génération de SGBD née dans les années 80 et basée sur le modèle objet: ONTOS, ObjectStore, VERSANT, ORION, O2.

2.5 Les SGBR actuels

2.5.1 Les SGBD commerciaux :

- Oracle (Unix, Linux, Windows)
- Microsoft : SQL serveur et Access (Windows)
- IBM : DB2, Informix
- Sybase : Sql Anywhere (Windows, Unix, Linux ...)
- Interbase (Windows, Linux)
- ...

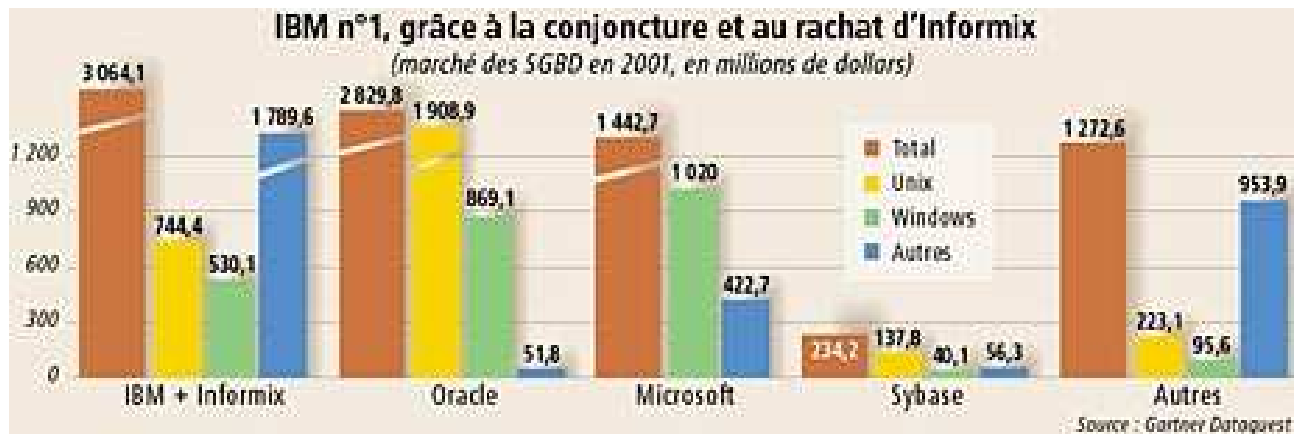


2.5.2 Les SGBDR libres

Généralement sous Windows et Linux

- My-SQL
- Postgre Sql
- Firebird
- Adabas, SAP-DB, Max DB
- ...

2.5.3 Le poids des acteurs du marché



3 Les objectifs d'un SGBDR

- Indépendance physique

Un remaniement de l'organisation physique des données n'entraîne pas de modification dans les programmes d'application.

- Indépendance logique

Un remaniement de l'organisation logique des données (ajout d'une nouvelle rubrique, ajout d'une nouvelle liaison...) n'entraîne pas de modifications dans les programmes d'application dont la "vision" logique n'a pas évolué.

- Manipulation des données par des langages non procéduraux



Des utilisateurs non informaticiens doivent pouvoir manipuler simplement les données, c'est-à-dire les interroger et les mettre à jour sans préciser d'algorithme d'accès.

- Administration facilitée des données

Un SGBD doit fournir des outils pour décrire les données, permettre leur suivi de ces structures et autoriser leur évolution. C'est la tâche des *administrateurs de données* : conception, création, maintenance, « arbitrage ».

- Efficacité des accès aux données

Nécessité de garantir un bon *débit* (nombre de transactions exécutées par seconde) et un bon *temps de réponse* (temps d'attente moyen pour une requête). Partage des ressources (CPU,

	BTS CGO 1 ^{ère} année P10	Chapitre 9 - Cours	
	<i>Les bases de données</i>		Page 5 / 8

disques...) entre les utilisateurs en optimisant l'utilisation globale afin d'éviter, par exemple, qu'une requête courte d'un utilisateur attende la fin d'une requête longue d'un autre utilisateur.

- Redondance contrôlée des données

Si redondance, volume de stockage plus important, opérations de mise à jour multiples, incohérences momentanées ou permanentes.

- Cohérence des données

Ex. L'âge d'une personne doit être un entier positif. Un SGBD doit veiller à ce que les applications respectent cette règle lors des modifications de données. Une telle règle est appelée *contrainte d'intégrité*.

- Partage des données

Diverses applications doivent pouvoir partager les données de la base dans le temps et simultanément, comme si elles étaient seules à les utiliser.

- Sécurité des données

Les données doivent être protégées contre les accès non autorisés ou mal intentionnés. La sécurité des données doit aussi être assurée en cas de panne d'un programme ou du système, voire de la machine.

4 L'organisation d'un SGBDR

4.1 *Notion de tables*

Une base de données est constituée par des tables **reliées** entre elles, c'est pourquoi on parle de bases de données relationnelles. Une table est un tableau. Ses lignes sont appelées les **enregistrements** et l'intitulé de ses colonnes les **champs**. A l'intersection d'une ligne et d'une colonne figure une **valeur**

La structure d'une table donne la liste des champs (attributs) qui la composent. Pour chaque champ, elle précise ses propriétés, essentiellement :

- Son **nom** codé.
- Son **type**.
- Son **caractère obligatoire ou facultatif** (si une valeur est ou non requise pour ce champ).
- Sa **plage de valeurs**, éventuellement
- ...

4.2 *Les types de données de base*

Pour chaque champ, il faut préciser le type d'informations que celui-ci sera capable de stocker. A chaque type correspond une place en octets nécessaire sur le disque pour enregistrer les valeurs du champ. Les différents types sont:

4.2.1 *Le type numérique*

- Il permet de stocker des données quantitatives sur lesquelles des calculs pourront être faits.



- Il occupera 1, 2, 4 ou 8 octets selon que les nombres à enregistrer seront des octets, des entiers ou des nombres réels.
- Ce type comprend des sous-types (octet, entier, entier long, réel simple, réel double).

4.2.2 Le type texte

- Il permet de stocker des données sous forme de texte ou chaîne de caractères.
- Sa longueur peut être définie de façon fixe ou variable (type Varchar)

4.2.3 Le type Date/heure

- Il permet de gérer le temps.
- C'est un type qui permet de prendre en compte le nombre de jours par mois, y compris le cas particulier du mois de février.
- Il permet de programmer très simplement des calculs sur les dates et les heures.
- Il permet de faire des extractions sur la date (mois, année ...)
- Ce type occupe 8 octets.

4.3 Des types particuliers

Il s'agit souvent de dérivés des types de bases qui sont déclinés en fonction de la base de données. Il peut également s'agir simplement de mise en forme pour l'affichage. Ils sont très répandus dans les bases de données à interface graphique (Access)

4.3.1 Le type monétaire

- Valeurs monétaires et données numériques utilisées dans les opérations mathématiques sur des données de 1 à 4 décimales. Précision à 15 chiffres à gauche de la virgule et à 4 chiffres à droite.
- Il occupe 8 octets.

4.3.2 Le type NuméroAuto

- Numéro séquentiel unique (incrémenté de 1 en 1) ou numéro aléatoire attribué par le SGBDR lorsqu'un nouvel enregistrement est ajouté à une table.
- Les champs NuméroAuto ne peuvent pas être modifiés.
- Ce type occupe 4 octets.

4.3.3 Le type Oui/Non



- Il est aussi appelé type logique ou type booléen : valeurs Oui et Non et champs qui contiennent exclusivement une valeur duale (Oui/Non, Vrai/Faux ou Actif/Inactif).
- Il n'occupe qu'un octet.

4.3.4 Le type Lien hypertexte

- Il permet d'enregistrer un chemin d'accès vers un fichier, un site Internet.

4.3.5 Le type mémo

- Il s'agit d'un champ texte pouvant aller jusqu'à 65 535 caractères

	BTS CGO 1 ^{ère} année P10	Chapitre 9 - Cours	
	<i>Les bases de données</i>		Page 7 / 8

4.3.6 Le type OLE

- Il s'agit d'un objet (tel qu'une feuille de calcul Excel, un document Word, une image, un son ...) lié ou incorporé.

4.4 La notion d'intégrité

Les contraintes d'intégrité font partie du schéma conceptuel. Parmi celles-ci, on distingue :

- Les contraintes de domaines sont des contraintes liées aux colonnes des tables (attribut non nul, ou dans une fourchette de valeurs ...)
- La contrainte d'intégrité d'entité précise qu'une table doit toujours avoir une clé primaire
- La contrainte d'intégrité référentielle est liée aux clés étrangères.

Appliquer l'intégrité référentielle implique :

- Dans le champ clé étrangère vous ne pouvez pas entrer une valeur qui n'existe pas dans la clé primaire (Un client ne peut être saisi dans une commande s'il n'existe pas).
- Vous ne pouvez pas effacer un enregistrement de la table primaire si des enregistrements correspondants existent dans une table liée
- Vous ne pouvez pas modifier une valeur clé primaire dans la table primaire si cet enregistrement a des enregistrements liés

La notion d'intégrité **d'intégrité référentielle** est au cœur même de la base de données. Elle lui confère son caractère relationnel. **Référentielle** signifie que la clé étrangère doit faire référence à une clé primaire.

5 Rappel sur le modèle relationnel

Le modèle relationnel représente la structure de la base de donnée, à savoir

- La structure des tables (nom, nom des champs)
- Les liens entre elles.
- La **clé primaire**, obligatoire pour chaque table, permet d'identifier de façon unique chaque enregistrement de la table.
- La **clé étrangère** matérialise la liaison de la table vers une autre table.

Le schéma relationnel est indépendant de la base de données utilisée.

5.1 Représentation du modèle relationnel

Cette représentation n'est pas normalisée, mais fait appel à des habitudes. Il est important dans cette représentation de représenter explicitement:

- Les tables avec leurs noms
- Les champs (ainsi que leurs noms) associés aux tables
- Les clés primaires
- Les clés étrangères
- Les liaisons clés primaires, clés étrangères

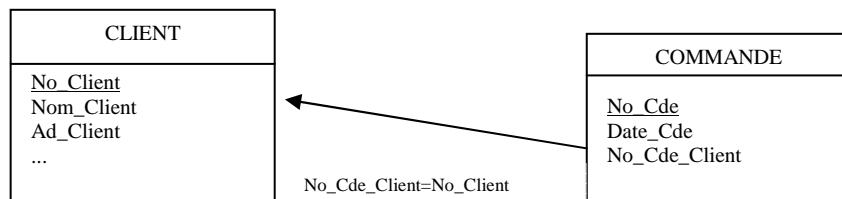
Elle existe sous différentes formes :

5.1.1 La représentation textuelle :

CLIENT (No_Client, Nom_Client, Ad_Client ...)
 COMMANDE (No_Cde, Date_Cde, No_Client_Cde#)

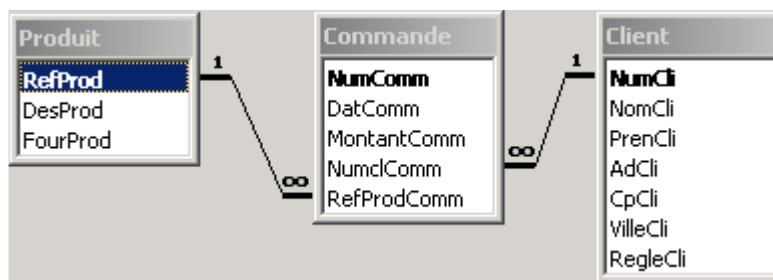
Cette représentation non graphique est commode, car rapide à écrire, mais elle est pauvre et insuffisante dès que le nombre de table se multiplie. Le lien entre la clé étrangère et la clé primaire qu'elle référence n'est pas symbolisé. Il faut donc soit connaître le plan de codification mis en œuvre, soit donner à la clé étrangère le même nom que la clé primaire qu'elle référence, soit l'indiquer en clair par une phrase, ce qui est de plus en plus souvent réalisé dans les études de cas.

5.1.2 La représentation graphique



Cette représentation est plus visuelle, s'adapte à des schémas beaucoup plus complexes et met en relief les liens entre table.

5.1.3 La représentation Access



Il s'agit d'une représentation graphique particulière qui visualise de plus les règles de gestion sur la base (application de l'intégrité référentielle ou non). Il s'agit d'une particularité d'Access.