

Secteur Tertiaire Informatique  
Filière « Etude et développement »

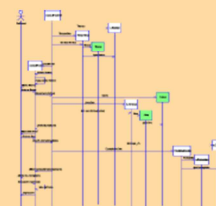
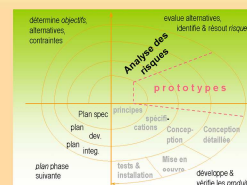
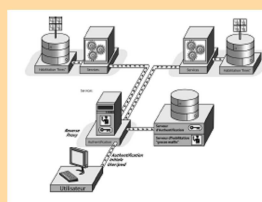
Séquence « Mettre en place la BDD »

Les SGBD

Apprentissage

Mise en pratique

Evaluation





# TABLE DES MATIERES

Table des matières .....	3
1. L'outil Base de Données : généralités .....	5
1.1 Définition.....	5
1.2 Facteurs à l'origine du développement.....	5
1.3 Rappel sur les systèmes de gestion de fichiers.....	5
1.4 Système de gestion de base de données (SGBD) .....	6
1.5 Historique des SGBD .....	7
1.6 Objectifs de l'approche BDD .....	7
1.6.1 Intégration et corrélation .....	7
1.6.2 Flexibilité .....	8
1.6.3 Disponibilité .....	8
1.6.4 Sécurité .....	9
1.7 Fonctionnement d'un SGBD.....	9
2. Différents langages d'un SGBD.....	10
2.1 Présentation.....	10
2.2 Classification des LMD.....	11
3. Rôle de l'administrateur de la base (DBA).....	11
4. En complément : Présentation des principaux modèles logiques de Bases de données .....	12
4.1 Le modèle hiérarchique.....	12
4.2 Le modèle en réseau .....	13
4.3 Le modèle relationnel.....	14

## Objectifs

Ce document a pour simple vocation de présenter les fondements des logiciels de gestion de bases de données. Il s'agit ici de les resituer dans leur historique et d'en définir les principales caractéristiques afin de mieux comprendre les principes des SGBD Relationnels et du langage SQL, massivement utilisés pour les applications de gestion.

## Pré requis

Aucun en particulier.

## Outils de développement

Aucun.

## Méthodologie

## Mode d'emploi

Symboles utilisés :



Renvoie à des supports de cours, des livres ou à la documentation en ligne constructeur.



Propose des exercices ou des mises en situation pratiques.



Point important qui mérite d'être souligné !


## Ressources

## Lectures conseillées

# 1. L'OUTIL BASE DE DONNEES : GENERALITES

## 1.1 DEFINITION

Le concept Base de Données (BDD) apparaît vers 1960 suite au nombre croissant d'informations que les entreprises doivent gérer et partager (évolution dans les systèmes informatiques). Historiquement, chaque application engendrait ses propres fichiers de données exploités par ses propres programmes. La création d'une base de données va à l'encontre de cette façon de procéder ; elle rend possible la centralisation, la coordination, l'intégration et la diffusion de l'information archivée.

 On peut en donner la définition suivante : **une base de données est un ensemble structuré de données enregistré sur des supports accessibles par l'ordinateur, pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun.**

En fait, une base de données enregistre des faits, des événements qui surviennent dans la vie d'un organisme, pour les restituer à la demande ; elle permet également de tirer des conclusions en rapprochant plusieurs faits élémentaires les uns des autres. Les données peuvent être manipulées par plusieurs utilisateurs ayant des vues différentes de ces données. Par exemple, dans le domaine des ressources humaines, le Service RH aura accès à toutes les données des salariés alors que le comité d'Entreprise ne pourra que consulter les données générales. La structure de l'ensemble requiert une description rigoureuse appelée **SCHEMA**.

## 1.2 FACTEURS A L'ORIGINE DU DEVELOPPEMENT

- nombre croissant d'informations à gérer ;
- évolution technologique des mémoires : augmentation des capacités, diminution des temps d'accès ;
- évolution des logiciels avec l'apparition sur le marché de produits fiables et diversifiés ;
- développement des systèmes 'temps réel' ;
- modification dans l'approche des problèmes de gestion par les entreprises. (approche globale par des données organisées de façon rationnelle plutôt qu'approche par des étapes basées sur les fonctions à réaliser).

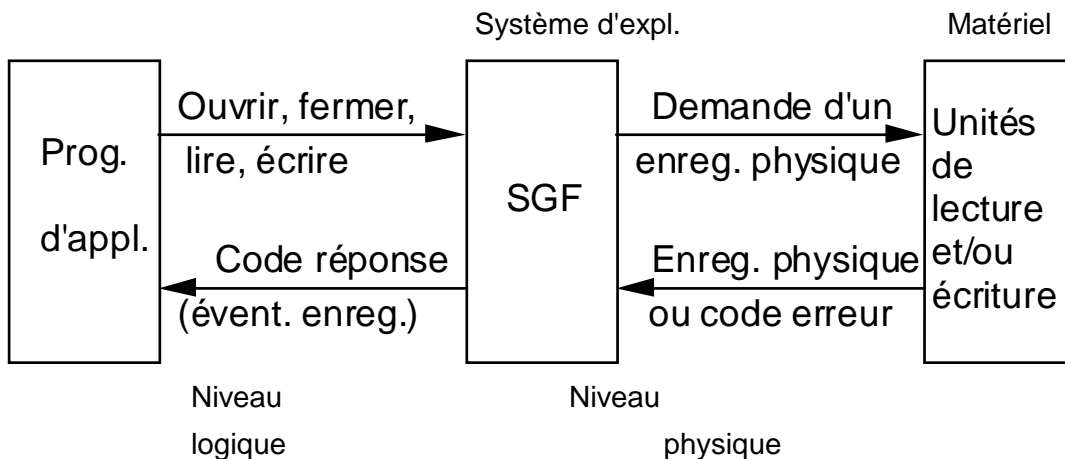
## 1.3 RAPPEL SUR LES SYSTEMES DE GESTION DE FICHIERS

Les données étant stockées sur des supports physiques, différents systèmes assurent des manipulations à 3 niveaux :

- support physique comme le disque dur (une sauvegarde de fichiers consiste en un passage des données d'un support à un autre) ;
- structures (opérations d'ouverture de fichier, fermeture, lecture, écriture) ;
- exploitation du contenu (calculs, tests...).

L'ensemble de ces fonctions est réalisé par:

- des programmes d'application ;
- des unités de lecture et/ou d'écriture ;
- des programmes de communication ou d'interface, l'ensemble de ces éléments constituant un S.G.F. (système de gestion de fichiers, assuré par le système d'exploitation de l'ordinateur), que l'on peut représenter selon le schéma suivant :

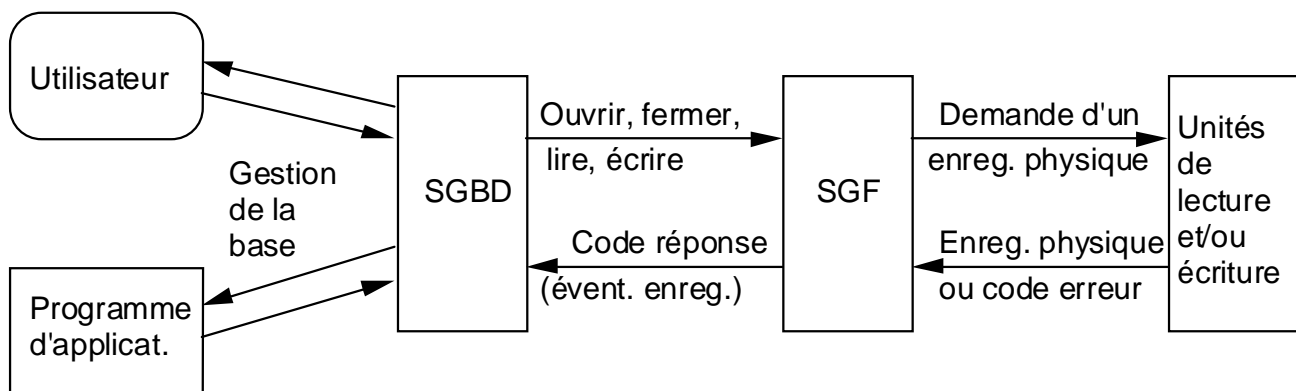


#### 1.4 SYSTEME DE GESTION DE BASE DE DONNEES (SGBD)

C'est l'ensemble des programmes et des langages de commande (livrés par le constructeur ou la société qui commercialise le produit) qui permettent :

- la définition des différents "ensembles de données" de la base, et les relations existant entre eux ;
- le traitement, l'exploitation de ces données (interrogations, mises à jour, calculs, extractions...) ;

**Le SGBD reçoit des commandes aussi bien des programmes d'application que des utilisateurs; il commande les manipulations de données, généralement par l'intermédiaire du SGF de base de l'ordinateur.**



Les SGBD

## 1.5 HISTORIQUE DES SGBD

Les SGBD ont déjà près de 50 ans d'histoire. Les années 1960 ont connu un premier développement des bases de données sous forme de fichiers reliés par des pointeurs. Elles étaient en fait un ensemble d'enregistrements stockés les uns à la suite des autres et accessibles par des valeurs de données appelées clés.

A la fin des années 60, apparaissent les premiers SGBD, caractérisés d'une part par la séparation entre la description des données et leur manipulation par les programmes d'application, d'autre part par l'avènement des langages navigationnels (possibilité de se déplacer dans des structures de type graphe).



La 2ème génération, mise au point depuis les années 1970, s'appuie sur la théorie mathématique des relations. **Les données sont présentées aux utilisateurs sous forme de relations** entre domaines de valeurs ; **le langage d'accès, de type déclaratif, non procédural, permet de spécifier les données que l'on souhaite obtenir sans dire comment y accéder**. Le langage standard SQL en est un bon exemple. Ces SGBD supportent généralement une architecture répartie, au moins avec des stations "clients" transmettant leurs requêtes à de puissants "serveurs" gérant les bases de données. **Les SGBD/R ('R' pour 'relationnel'), actuellement massivement utilisés dans les applications de gestion, et basés sur le langage standard SQL en sont la meilleure illustration.**

Les modèles de base de données Orientés Objet, en gestation à la fin du XX<sup>e</sup> siècle, ne semblent pas être promis à un bel avenir... Le temps de gestation a été trop long et le consortium autour de ces projets a été dissous avec l'arrêt des travaux en 2001.

On s'oriente plus aujourd'hui vers des SGBD relationnels de type 'NoSQL' (Cassandra, MongoDB). Il ne s'agit pas ici d'oublier le modèle relationnel ni le langage SQL car NoSQL signifie littéralement 'Not Only SQL'.

Une autre piste récente apparaît avec la notion de 'Big Data' et la nécessité d'analyser rapidement d'énormes volumes de données selon de multiples critères afin d'en faire ressortir du sens ou des tendances (par exemple, analyser le détail des facturations d'un opérateur téléphonique ou les requêtes HTTP reçues par un serveur Web).

Affaires à suivre...

## 1.6 OBJECTIFS DE L'APPROCHE BDD

Pour pallier les inconvénients des méthodes classiques (à base de fichiers), les bases de données tentent d'atteindre 4 objectifs principaux : **intégration et corrélation**, **flexibilité** (indépendance), **disponibilité**, **sécurité**. Les 12 principes fondateurs ont été exprimés par un ingénieur IBM, E. F. Codd (voir annexe).

La solution à ces problèmes passe par une **distinction nette entre les données** (avec la structure qui leur est associée), **et les procédures de manipulation** de ces données. Aux données, on associera une fonction d'administration des données, aux procédures de manipulation une fonction de programmation.

### 1.6.1 Intégration et corrélation


Dans les systèmes dits "classiques", chaque application gère son propre ensemble de données utiles, ses propres "fichiers": il existe donc un risque d'avoir une floraison de fichiers et de programmes disparates, redondants et non compatibles. Il en résulte plusieurs inconvénients :

- **redondance d'où danger d'incohérence des données**

La même donnée peut appartenir à plusieurs applications, induisant une déperdition de stockage. Toute modification de cette donnée est à enregistrer plusieurs fois : si cette mise à jour multiple n'est pas effectuée correctement, les données deviennent incohérentes. De plus, les mises à jour étant multiples, le coût de la mise à jour augmente du fait de la multiplication des entrées-sorties physiques.

- **difficulté de mise en place de nouveaux traitements**

Mettre en place de nouvelles applications, non prévues a priori, entraîne dans un système classique, des duplications supplémentaires de données. De plus, pour les intégrer avec le reste des applicatifs en exploitation, des modifications, parfois importantes, sont nécessaires.

 **Dans l'approche BDD, un "réservoir" commun (intégration) est constitué, représentant une modélisation (corrélation) aussi fidèle que possible de l'organisation réelle de l'entreprise. Toutes les applications puisent dans ce réservoir les données les concernant, évitant les duplications.** A l'inverse, le partage des données communes par des utilisateurs ayant des besoins différents va poser de nouveaux problèmes de type contrôle de la concurrence d'accès.

### 1.6.2 Flexibilité

La notion de flexibilité est essentielle dans la conception de tout système de taille importante. Dans le cas des BDD, cette notion porte généralement le nom d'indépendance.

Dans les systèmes classiques, tout changement intervenant dans le stockage des données (support, méthode d'accès physique...) entraîne des modifications lourdes (voire une reprogrammation complète !) des applications correspondantes. L'approche BDD poursuit trois objectifs, correspondant au concept d'indépendance des données par rapport aux traitements :

- **indépendance physique** : tout changement de support, de méthode d'accès reste transparent au niveau de l'utilisation des données.
- **indépendance logique** : les programmes d'application sont insensibles à une modification dans l'organisation logique globale des données, grâce à la définition de sous-schémas couvrant les besoins spécifiques.
- **indépendance vis-à-vis des stratégies d'accès** : de plus en plus, l'utilisateur développeur n'a pas à prendre en charge l'écriture des procédures d'accès aux données. Il n'a donc pas à intégrer les modifications tendant à optimiser les chemins d'accès (ex: création d'index).

### 1.6.3 Disponibilité

Le choix d'une approche BDD ne doit pas se traduire par des temps de traitement plus longs que ceux des systèmes antérieurs. En fait, tout utilisateur doit (ou devrait !) pouvoir ignorer l'existence d'utilisateurs concurrents.

**L'aspect "performance" est donc crucial dans la mise en œuvre d'une base de données.**

Un tel objectif ne peut être atteint que si la conception d'une base de données est menée de façon rigoureuse avec un découpage fonctionnel adéquat. Les règles et contraintes inhérentes sont évoquées lors de l'apprentissage d'une méthodologie d'analyse (exemple MERISE).



#### 1.6.4 Sécurité

La sécurité des données (qui sera décrite plus en détail ultérieurement) est un terme générique qui recouvre plusieurs aspects :

- **l'intégrité**, ou protection contre l'accès invalide (erreurs ou pannes), et contre l'incohérence des données vis-à-vis des contraintes de l'entreprise ;
- la **confidentialité**, ou protection contre l'accès non autorisé ou la modification illégale des données.

Pour ne pas affecter les performances de façon sensible, la sécurité doit également être prise en compte dès la phase de conception.

#### 1.7 FONCTIONNEMENT D'UN SGBD

La chronologie des opérations élémentaires effectuées lors d'une demande de donnée de la base par un programme est la suivante :

- un programme d'application A émet une demande de lecture d'une donnée (ou d'un groupe de données) de la base : la demande est transmise au SGBD ;
- le SGBD traite la demande en consultant le sous-schéma (schéma externe) relatif au programme d'application A, obtenant ainsi la description des données ;
- le système consulte le schéma logique global (schéma conceptuel) et détermine le type logique de données à extraire ;
- le système examine la description physique de la base en rapport avec la requête logique et détermine le (ou les) enregistrement(s) physique(s) à lire ;
- le système lance une commande au système d'exploitation pour rechercher physiquement l'enregistrement désiré ;
- le système d'exploitation, par le biais de ses méthodes d'accès au matériel, accède à l'enregistrement physique ;
- les données demandées sont transférées dans des buffers, ou mémoires tampons ;
- le SGBD, à partir d'une comparaison entre schéma logique et sous-schéma A, extrait, des données stockées dans le buffer, l'enregistrement logique réclamé par le programme d'application. Il effectue également les transformations éventuelles de format ;
- le SGBD transfère les données des buffers dans la zone de liaison du programme d'application A ;
- le SGBD fournit également des informations "d'état" au programme d'application, lui signalant en particulier les erreurs éventuellement constatées au cours du processus d'extraction ;
- le programme d'application, qui dispose des données et d'informations de "service" en assure enfin la bonne exploitation !

#### Les SGBD

Les ordres d'écriture dans la base physique sont traités par un processus similaire, toute modification ou adjonction étant en général précédée d'une opération de lecture.

A signaler que, dans la majorité des cas, le SGBD doit traiter simultanément plusieurs demandes de données en provenance de plusieurs programmes d'application, utilisant plusieurs schémas externes différents.

## 2. DIFFERENTS LANGAGES D'UN SGBD

### 2.1 PRESENTATION

La présentation des SGBD réalisée dans les pages précédentes a fait apparaître la nécessité de bien différencier les deux étapes et objectifs différents que sont la définition des données a priori (rôle du DBA ou administrateur de la base) et leur utilisation a posteriori (par les utilisateurs ou les programmeurs d'application).

Le SGBD met donc à disposition deux types de langage :



- **Langage de Description de Données (LDD)**

Il permet de décrire précisément la **structure de la base** et le mode de stockage des données. Tandis que l'utilisation de fichiers permet une description de données interne au programme, **dans une approche Base de Données, on effectue la description de toutes les données une fois pour toutes** : elle est constituée de l'ensemble des tables et dictionnaires de la base, son schéma.

En particulier, **le LDD précise la structure logique des données** (nom, type, contraintes spécifiques...), **la structure physique** (mode d'implantation sur les supports, mode d'accès), la définition des **sous-schémas ou "vues"**.



- **Langage de Manipulation de Données (LMD)**

Il convient de rappeler que l'utilisation d'une BDD suppose un grand nombre d'utilisateurs (souvent non informaticiens) ayant tous des tâches et des besoins variés auxquels le LMD doit pouvoir répondre.

On peut répertorier :

- le langage d'interrogation ou langage de requête

Il évite le recours, en particulier pour des besoins ponctuels, à des langages généraux de programmation, qui impliquent des coûts et des délais de développement souvent fort longs. Il doit être à syntaxe souple, accessible aux non-spécialistes et permettre la formulation de demandes utilisant des critères variés et combinés. SQL en un exemple typique.

- le langage hôte

Pour les traitements réguliers ou mettant en œuvre d'importants volumes d'informations, **le SGBD doit pouvoir fournir une interface permettant l'utilisation de la base à l'aide des langages généraux** (COBOL, C, Basic, Java...).



**Dans les principaux SGBD Relationnels du commerce, LDD et LMD sont assurés par le langage SQL.**

## 2.2 CLASSIFICATION DES LMD

Les LMD se répartissent en 2 catégories principales :

- **langages navigationnels**

On les rencontre avec les SGBD 'hiérarchiques' ou 'réseaux'. Les requêtes du langage (ou questions) décrivent les chemins d'accès aux différentes données, celles-ci étant généralement chaînées entre elles.

- **langages algébriques** (ex : SQL)

On les rencontre avec les SGBD relationnels. Ils utilisent, pour fournir des résultats aux requêtes, les opérateurs de l'algèbre relationnelle.

## 3. ROLE DE L'ADMINISTRATEUR DE LA BASE (DBA)

Rappelons succinctement les différentes fonctions assumées par un SGBD :

- description de la structure de la base (schéma) ;
- organisation du stockage physique ;
- manipulation des informations (sélection, extraction, mise à jour) ;
- protection.

Pour personnaliser de façon fiable les accès à la base, il convient **d'identifier l'utilisateur** (code et mot de passe) et de vérifier qu'il est autorisé à effectuer sur les données les traitements qu'il demande (contrôle des droits d'accès).

- sécurité, **restauration** : possibilité de reconstituer la base dans un état satisfaisant après tout incident ;
- **optimisation des ressources** : le logiciel doit fournir des statistiques précises sur l'état de la base et permettre des réorganisations physiques périodiques qui éviteront la dégradation des performances globales du système ;
- **intégrité des données** (cohérence des informations les unes par rapport aux autres).

L'essentiel de la mise en œuvre de ces fonctions revient à une personne (ou une équipe) appelée **administrateur** de la BDD ou **DataBase Administrator**. Elle dispose d'un ensemble d'outils logiciels pour l'assister dans sa tâche. Son rôle consiste à :

- intervenir en tant que **conseil lors de l'étape conceptuelle** de l'analyse (responsabilité de la gestion des données) ;
- décider des **techniques d'accès et de l'implantation physique** (choix des disques, index et autres moyens d'optimisation) ;
- gérer les diverses **autorisations d'accès** ;
- définir les **stratégies de reprise en cas d'incident** ;
- **suivre régulièrement les performances** du système et réaliser en conséquence les modifications ou évolutions qui s'imposent.

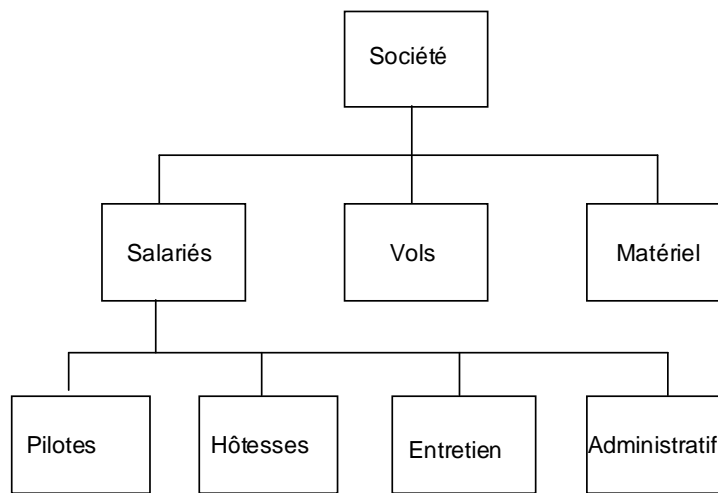
Les SGBD

## 4. EN COMPLEMENT : PRESENTATION DES PRINCIPAUX MODELES LOGIQUES DE BASES DE DONNEES

Un très grand nombre de modèles existent sur le marché, réalisant plus ou moins bien les diverses fonctions évoquées précédemment. Les 3 principaux, par le nombre d'installations, sont, dans l'ordre chronologique de leur arrivée sur le marché, le modèle hiérarchique, le modèle réseau (ou navigationnel), le modèle relationnel.

### 4.1 LE MODELE HIERARCHIQUE

Exemple : représentation (partielle !) du SI d'une compagnie aérienne



L'ancêtre, et le plus répandu, en est le SGBD IMS (Information Management System), développé et commercialisé par IBM dans les années 1970.

Sa caractéristique principale est la très forte dépendance entre la description de la structure des données et la manière dont celles-ci se trouvent enregistrées sur le support physique. Les éléments de base du modèle sont des enregistrements logiques reliés entre eux pour constituer un arbre ordonné ou arborescence valuée.

Les différentes entités (ou segments) constituent les nœuds, celui de plus haut niveau portant le nom de racine ; les branches (pointeurs logiques entre entités) constituent les liens. Chaque segment est une collection d'objets appelés champs (ou fields). **Chaque nœud, sauf la racine, a obligatoirement un père, mais un seul, et peut avoir plusieurs fils.**

De façon visuelle, cette représentation schématique fait apparaître :

- les avantages du modèle
  - rigueur des structures et des chemins d'accès, simplicité relative de l'implémentation, adéquation parfaite à une entreprise ou une application à structure arborescente (généalogie...).

- les contraintes un peu lourdes qui en sont la contrepartie
  - Les accès se font normalement depuis la racine et uniquement depuis la racine.
  - La structure élude la possibilité de liens 'plusieurs pères-plusieurs fils', ne permettant que le lien '1 père-plusieurs fils'. La représentation d'autres relations impose de ce fait une redondance de l'information.  
Exemple : comment représenter un parc de véhicules et un ensemble de chauffeurs, chaque chauffeur pouvant conduire plusieurs véhicules, et un véhicule pouvant être conduit par plusieurs chauffeurs ?
  - On est amené à constater des "anomalies" lors des opérations de mise à jour (on parle de stockage, soit insertion, destruction, modification). Ainsi, l'élimination d'un nœud entraîne la suppression de tous les segments de niveau inférieur qui lui sont rattachés (risque de perdre des données uniques).
  - L'indépendance logique est très réduite puisque la structure du schéma doit refléter les besoins des applications.
  - Il n'existe pas d'interface utilisateur simple.

## 4.2 LE MODELE EN RESEAU

Il peut être considéré comme une évolution du modèle hiérarchique, en lui intégrant les résultats du travail du groupe CODASYL (comité de langage de programmation). Celui-ci a étudié l'extension de COBOL pour manipuler les bases de données et a rendu, en 1969, ses premières recommandations concernant syntaxe et sémantique du LDD et du LMD.

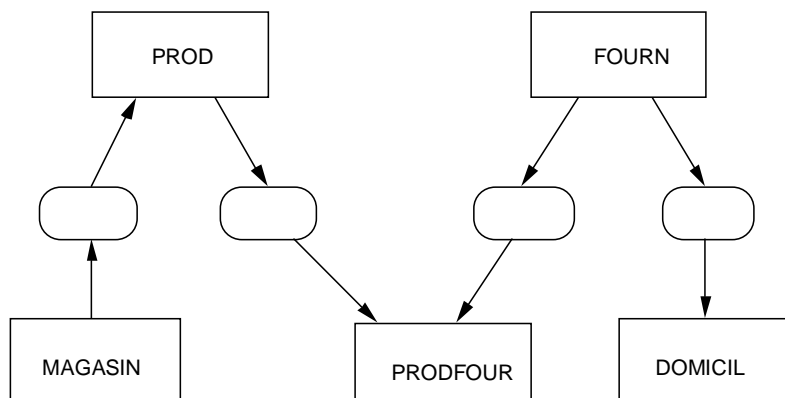
Même si cette vue est un peu simplificatrice, **une base en réseau peut être décrite comme un certain nombre de fichiers comportant des références des uns vers les autres**. Les entités sont connectées entre elles à l'aide de pointeurs logiques. Ce concept permet d'associer, à un enregistrement d'un ensemble de données A, une série d'enregistrements (ou 'records') d'un autre ensemble de données B. On constitue ainsi des SET, ou COSET, structure fondamentale du modèle en réseau :

- le lien entre les enregistrements de A et ceux de B est de type '1 père-plusieurs fils'.
- le COSET comporte un type d'enregistrement "propriétaire" (enregistrement de A = OWNER) et un type d'enregistrement "membre" (enregistrements de B = MEMBER).

On ne rencontre aucune restriction à la conception du modèle : un type de "record" peut à la fois être propriétaire et membre vis-à-vis de plusieurs sets ; de même, il peut être propriétaire (ou membre) de plusieurs sets, ce qui autorise au final l'implantation de relations 'plusieurs pères-plusieurs fils'.

Exemple :

Schéma représentant le sous-système d'information produits / magasins de stockage / fournisseurs / domiciliations bancaires



Avantages et inconvénients du modèle

- représentation naturelle des liens plusieurs pères-plusieurs fils.
- absence d'anomalies pour les opérations de mise à jour.
- commercialisation importante des systèmes correspondants (DMS, IDMS, TOTAL, IDS II, SOCRATE...) ; mais...
- pas d'indépendance par rapport aux stratégies d'accès ;
- lourdeur des langages de manipulation ; l'utilisateur doit "naviguer" dans le réseau logique constitué par les enregistrements et les chaînes de pointeurs.

### 4.3 LE MODELE RELATIONNEL

C'est un article publié en 1969 par un mathématicien du centre de recherche IBM, Codd, qui définit les bases de ce modèle 'relationnel'.

Codd s'est intéressé au concept d'information et a cherché à le définir sans se préoccuper de la technique informatique, de ses exigences et de ses contraintes. Il a étudié un modèle de représentation des données qui repose sur la notion mathématique de "relation". Dans la pratique, une relation sera représentée par une table de valeurs.

Exemple: représentation d'une table du personnel

Matricule	Nom	poste	Salaire	N° dept
350	Durand	Employé	8000	320
780	Dupond	Cadre	15000	870
320	Veillon	PDG	25000	400
490	Martin	Cadre	15000	320

## Théorie et définitions



- Une **relation** est un ensemble de tuples ou uplets (lignes ou articles), dont l'ordre est sans importance ; chaque tuple est unique.
- Les colonnes de la table sont appelées attributs ou champs ; leur ordre est défini lors de la création de la table.
- Une **clé** est un ensemble ordonné d'attributs qui caractérise un tuple. Une clé primaire le caractérise de manière unique, à l'inverse d'une clé secondaire.
- On dit qu'un attribut A est un *déterminant* si sa connaissance détermine celle de l'attribut B.

## Objectifs du modèle :

- proposer des schémas de données faciles à utiliser ;
- améliorer l'indépendance logique et physique ;
- mettre à disposition des utilisateurs des langages de haut niveau pouvant éventuellement être utilisés par des non-informaticiens ;
- optimiser les accès à la base de données ;
- améliorer l'intégrité et la confidentialité ;
- prendre en compte une variété d'applications ;
- fournir une approche méthodologique dans la construction des schémas.

## Caractéristiques du modèle :

- les structures de données sont simples : ce sont des tables à deux dimensions où les éléments sont des données élémentaires.
- un ensemble d'opérateurs appliqués aux relations (union, intersection, différence, produit cartésien, projection, sélection, jointure, division) ; l'algèbre relationnelle, permet la définition, la recherche et la mise à jour des données. Cette algèbre est la base des langages de manipulation non procéduraux, type SQL.
- un ensemble de contraintes d'intégrité (unicité de clé, contrainte référentielle) définit l'état cohérent, intègre de la base ;
- toute information de la base de données est représentée par des valeurs dans des tables ;
- il n'y a pas de pointeurs visibles par l'utilisateur entre tables.

## Avantages et inconvénients :

- indépendance de l'utilisateur vis-à-vis de la structure logique, de la structure physique et des stratégies d'accès, d'où simplicité et clarté dans l'utilisation ;
- puissance et uniformité de la représentation ;
- très grande souplesse d'expression dans la sécurité des données ;

## Les SGBD

- existence d'interfaces non procédurales ;
- indépendance des programmes vis-à-vis de la base : une modification du schéma ne nécessite pas de modification des applicatifs existants ;
- difficulté de conception d'un schéma conceptuel propre, étape indispensable à un fonctionnement satisfaisant du modèle ; des méthodologies comme la démarche Merise permettent d'assurer la production de schémas conceptuels propres.
- Grand nombre de produits commercialisés et en perpétuelle évolution : **DB2, INGRES, ORACLE, SQLServer, MySQL...**



## **CREDITS**

### **ŒUVRE COLLECTIVE DE L'AFPA**

**Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services**

#### **Equipe de conception (IF, formateur, mediatiseur)**

B. Hézard - Formateur

Ch. Perrachon – Ingénieure de formation

**Date de mise à jour : 08/02/16**

### **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Les SGBD

Afpa © 2016 – Section Tertiaire Informatique – Filière « Etude et développement »