

INTRODUCTION BASE DE DONNEES ET SGBD

Georges Gardarin, Base de données ;

Serge Miranda, Bases de données : Architectures, modèles relationnels et objets, SQL.

<https://stph.scenari-community.org/idl-bd/7/co/reUC084.html>

<https://www.w3schools.com/sql/default.asp>

<https://www.geeksforgeeks.org/sql-ddl-dml-tcl-dcl/>

PLAN DU COURS

- Préliminaire : Système d'information et système informatique ;
- Objectifs et architecture ;
- Modélisation (entités/association) ;
- Modèle relationnel (algèbre relationnelle, opérateurs ensemblistes et spécifiques) ;
- Langage SQL.

Systeme d'information (en bref)

1-Définition de concepts afférents

-Système :

Par système, il faut entendre un ensemble d'éléments matériels et immatériels mis en interaction dynamique dont certains pouvant être transformés au moyen de processus ou de mécanismes en vue d'atteindre un but (Cf Doc Système d'information PREPA détail).

Par système, on entend un ensemble ordonné et structuré de matériels formant un tout logique.

Un système consiste en un dispositif ou ensemble d'éléments ayant une fonction déterminée.

Un **système** est un ensemble d'éléments interagissant entre eux selon un certain nombre de principes ou de règles.

Un système peut être constitué d'un ensemble cohérent de notions (concepts, idées, d'archétypes, prototypes, paradigmes, spécimens, parangons, modèle ...).

Un système peut s'apparenter à un ensemble de méthodes ou de procédés pour produire un résultat.

Un système peut être considéré comme un assemblage d'éléments régi par des lois.

Exemples de systèmes :

Monde (le monde est un), être, être vivant, être humain, femme, homme, machine, entreprise, famille, Etat, système expert, ordinateur, vélo, virus, système d'extraction de pétrole, système de déforestation, astre, astéroïde, comète, grain de sable, goutte de pluie, terre, vent, eau, mer, feu, sac, feuille, univers, H1N1, grippe, énergie, ...

Système expert (logiciel utilisant une somme de connaissances propres à un domaine pour prendre des décisions ou résoudre des problèmes en simulant la démarche d'un être humain). C'est un système à base de connaissances et relevant de l'intelligence artificielle.

NB :

Il n'existe pas de système isolé véritablement. Les systèmes sont tous liés directement ou indirectement. Ils peuvent s'influencer positivement comme négativement. Le monde est un. Il constitue un tout. Notre univers, le monde y compris forment un graphe naturel.

Classification des systèmes selon leur degré de complexité croissant : 9 niveaux des systèmes

- a) Le système existe et est identifiable ;
- b) Le système agit ou réagit en fonction d'un but ;
- c) Le système a un comportement régulé (ou se régule) ;
- d) Le système fait son état (bilan interne et externe) : il s'informe sur son propre comportement et informe son environnement ;
- e) Le système décide de son comportement (de ses finalités) ;
- f) Le système mémorise les informations ;
- g) Il organise et coordonne : contrôle, dirige, ordonne, oriente, projette, planifie ;
- h) Il imagine : crée, invente, et innove. Il a des perspectives. Il détient des modules de tolérance de pannes et de fautes avancés et conçus pour son environnement (intérieur et extérieur).

i) Il s'auto finalise ou finalise : il tend vers une indépendance, une autonomie (manifestation du sommet de l'intelligentsia). C'est l'émergence des consciences. NASA (National Aeronautics and Space Administration) : Système américain sophistiqué au-delà du renseignement ; Systèmes produits par la NASA, Navigation aérienne et sous-marine.

-Donnée

La donnée est la représentation d'une information sous une forme conventionnelle en vue de son traitement par l'homme ou par une machine.

La donnée est la représentation d'une information sous une forme conventionnelle, convenant à une communication, à une interprétation ou un traitement par l'homme ou par une machine (moyens informatiques ou non).

La donnée est un fait, une notion, une instruction.

Elle est aussi l'enregistrement dans un code convenu d'une observation, d'un objet ou d'un phénomène (donnée factuelle, relative au fait), d'une image, d'un son, d'un texte.

-Information

D'après Boulding et Lemoigne, c'est aussi tout ce qui forme de manière significative une représentation imagée de la réalité.

L'information est alors tout le signifiant que l'on attache et que l'on peut déduire d'un ensemble de données, de certaines associations entre données.

L'information est un support de connaissance, c'est même la connaissance.

Elle peut avoir pour origine des moyens d'expressions humains ou non, être structurée ou non.

Quelques **formes** de l'information : écrite, orale, sonore, picturale, olfactive, tactile, gestuelle, électronique, numérique (digitale), (chiffre = digit).

Une donnée est une information élémentaire. En conséquence, l'information apparaît comme un agrégat de données pour en manifester sens et signification.

Infostructure : mot valise à partir des mots *information* et *infrastructure* pour traduire le fait que l'infrastructure du système d'information prime sur l'architecture logicielle ou/et matérielle.

L'informatique est devenue un ensemble d'outils de gestion et de contrôle de l'information. Elle évolue alors que le système d'information globale demeure.

-Connaissances

On définit la connaissance comme ce qu'on a appris par l'étude, par la pratique ou par l'expérience. C'est ce qu'on maîtrise. La connaissance est mise en évidence par une personne via son expertise.

Les techniques d'apprentissage plus poussées au plan de l'informatique actuel constituent une preuve de machines intelligentes à même de manipuler des connaissances et d'en produire à des fins d'intérêts multiples.

Ce niveau technologique du ressort du domaine de l'intelligence artificielle, n'a pas encore atteint la sagesse humaine en dépit des entraînements des machines.

-Sagesse

La sagesse est un état empreint de discernement et de jugement de bon sens. Elle englobe donnée, information et connaissance. L'émergence de conscience est ici manifeste. Jusque-là la sagesse est du ressort de l'homme.

2-Définition de l'entreprise :

Une entreprise est un système. Elle se distingue des autres systèmes à travers une définition plus spécifique.

[Terminologie :

Onomasiologie et sémasiologie (L'onomasiologie est le champ des désignations, des dénominations, des appellations, des terminologies alors que la sémasiologie, le champ de l'expression signifiante et définitionnel). Il s'agit de tenir un concept par un terme.

Sémasiologie c'est partir du terme de concept à ses significations notamment aux référents individuels, les instances. Et inversement, on a l'onomasiologie.

= > Definien et Definiendum (page 30).

On entend par entreprise, une entité socio économique et juridique regroupant des moyens humains, financiers et matériels en vue de la création de biens et de services (de la richesse).

Par entreprise, on entend, toute entité économique, quelle que soit sa forme juridique, permettant de rassembler et d'organiser de façon cohérente des ressources autour d'un objet social défini, sous une direction commune, et sous le contrôle des principaux apporteurs de ressources.

La structure d'une entreprise est la résultante d'un grand nombre de facteurs notamment la nature et le volume des activités, la personnalité des membres, la politique et la stratégie retenues, les usages, ainsi que l'environnement juridique et institutionnel.

Tout système est révélabile ou cernable à travers sa description ou dénommable dans une langue donnée.

Rôle de l'information dans l'entreprise

Trois rôles principaux sont assignés à l'information véhiculée et utilisée au sein d'une entreprise : Outil de communication interne, outil de communication externe et facteur de cohésion.

-Outil de communication interne :

C'est un système de communication en vue de répondre aux nécessités de gestion et aux obligations légales au plan interne :

Les nécessités de gestion porte sur la définition (structuration administrative verticale/horizontale, objectifs, activités et leur répartition.

Les obligations légales concernent la convention collective, les droit et devoir des parties composantes de l'entreprise, type langagier, respect mutuel, culture de l'entreprise.

-Outil de communication externe :

Il se focalise sur la visibilité et relation extérieures. En rapport avec les partenaires dont l'état en premier, il s'agit aussi de répondre aux nécessités de gestion (définition des objectifs, des activités et leur répartition) et aux obligations légales (convention et réglementation, exonération et imposition taxation).

-Facteur de cohésion sociale,

Il est question de savoir faire le management de l'information, afin de permettre à tout le monde d'être intéressé et participer à l'atteinte de l'objectif commun (tirer vers le haut et non vers le bas via des critiques constructives, participation active motivée, soutenue, volonté, intégration des individus, tendances véhiculées, éthique et moralité sociétales). On note l'entretien d'un environnement paisible et propice aux activités, aux affaires. il doit être de mise la sécurité des biens et des personnes, et d'aller et venir. Bien que, ce n'est pas tout on dit, mais aussi ce n'est pas tout on cache.

3-Système d'information

3.1-Définition :

Un système d'information est l'ensemble de toutes les informations qui circulent au sein de l'entreprise et les moyens mis en œuvre pour les gérer.

3.2-Finalités

- Aider à la prise de décision de qualité ;
- Cordonner l'ensemble des activités pour plus de synergie d'actions ;
- Contrôler l'évolution du système.

3. 3 -Rôles d'un SI

- recueillir,
- traiter et exploiter,
- mémoriser,
- diffuser.

3. 4-Typologie des systèmes d'information

Tout est en corrélation certes mais il importe de mettre en relief quatre typologies bien distinctives.

- Degré de formalisation des moyens
 - SI formel ;
 - SI informel.
- Niveau de décision
 - S.I de planification stratégique ;
 - S.I de planification tactique ;
 - S.I de planification opérationnelle .
- Nombres d'utilisateurs
 - S.I personnel ou utilisateur ;

- S.I collectif ;
 - S.I interentreprises.
- Degré d'informatisation ou d'automatisation des tâches
- S.I manuel ;
 - S.I mécanisé ;
 - S.I automatisé .

Vers une marche technologique forcée

Trois grandes époques sont très significatives dans la révolution liée au développement socio économique.

- Révolution agricole ;
- Révolution industrielle avec la découverte de l'énergie de transformation et de production ;
- Révolution des technologies de l'information et de la communication (TIC) marquée par trois grandes tendances :
 - informatique à architecture centralisée,
 - informatique à architecture distribuée ,
 - et informatique à architecture répartie.

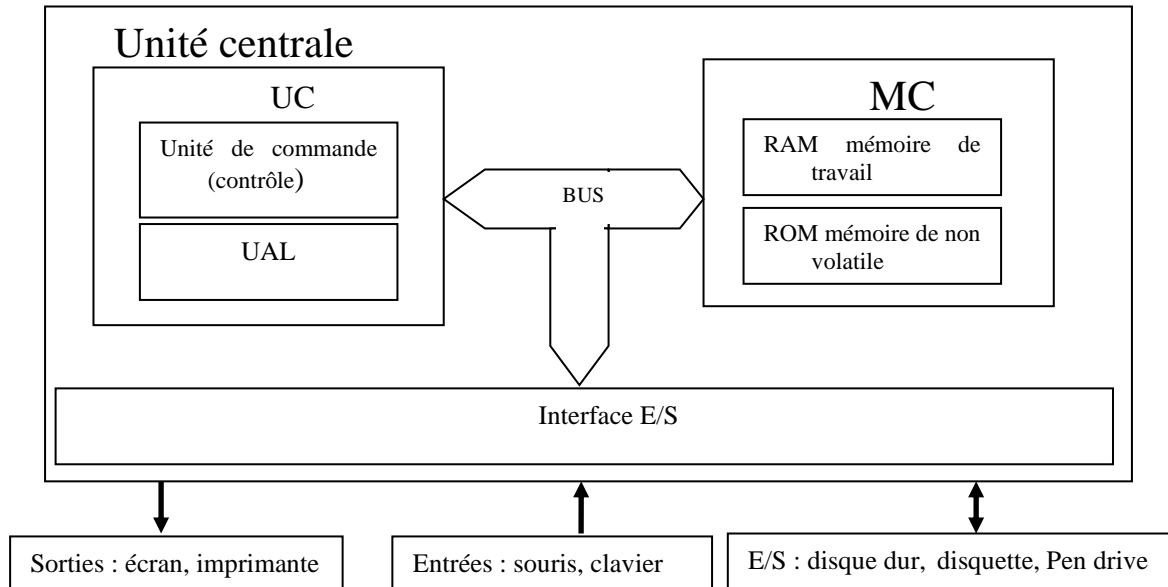
L'internet d'ubiquité fait son chemin. Il se fonde sur la miniaturisation matérielle et la factorisation logicielle. Il est une conséquence de la répartition en continue (3 tiers en n-tiers avec des entités machine à la fois serveur et client) relevant des TIC.

Système d'informatique (bref)

Définition d'informatique :

Science du traitement automatique et rationnel de l'information au moyen de l'ordinateur.

Définition de système d'informatique : ensemble de ressources informatiques (matériel et logiciel).



ANSI : American National Standard Institute

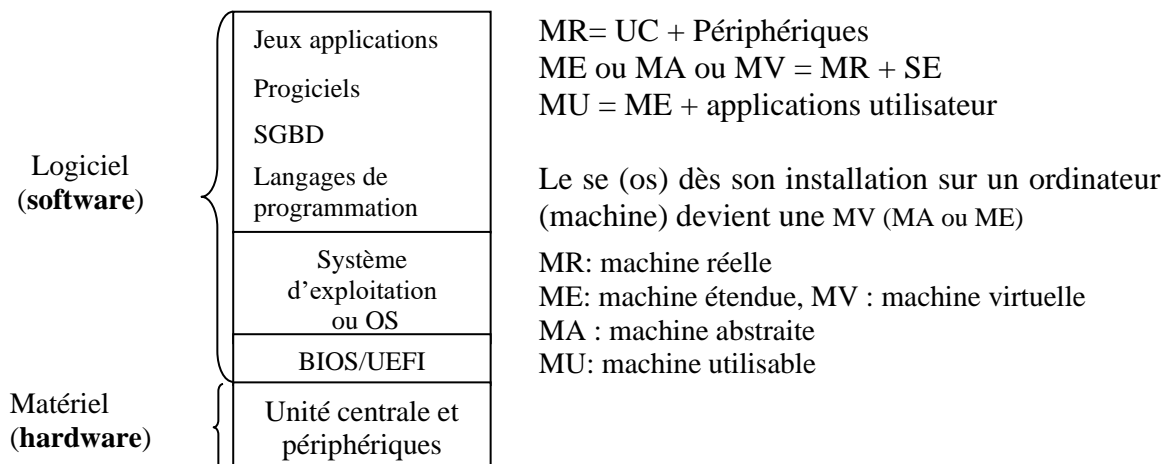
SPARC : Scalable Processor ARChitecture

Les technologies VLSI : Very Large Scale Integrator, ULSI : Ultra Large Scale Integrator sont les supports ou rempart en construction de la miniaturisation du matériel informatique et électronique.

Autres éléments de l'unité centrale : Carte mère, carte son, carte vidéo, carte tv, carte réseau, Boite d'alimentation, horloge

Le moniteur est en train de devenir inéluctablement un accessoire d'entrée sortie.

Cohabitation et agencement des composantes matérielle et logicielle.



BIOS : Basic Input Output System

UEFI : Unified Extensible Firmware Interface (en raison de l'extension sans cesse en taille des disques, TO, PO, EO, ..)

Le SE (operating system) a trois rôles fondamentaux :

- Machine abstraite (interface HM,) pour masquer les aspects fastidieux de la machine, offrir un environnement convivial et attractif de travail. Il masque les aspects fastidieux de l'ordinateur. C'est-à-dire il décharge l'utilisateur des programmes complexes de l'ordinateur.
- **gestionnaire des ressources matérielles et logicielles**. Il assiste le microprocesseur dans l'accomplissement de sa tâche.
- socle du développement et de la construction des logiciels.

Le deuxième rôle est plus englobant. Il prédomine.

En raison de l'architecture informatique actuellement répartie, Internet est le point de convergence de ces trois réseaux classiques d'antan. Ces trois types de réseaux sont :

- réseau informatique,
- réseau des télécommunications,
- Télédiffusion (câblo-opérateur ou réseau audiovisuel).

Ils sont en fusion de laquelle résulte le réseau multimédia (Internet) grâce à l'avènement de l'IP (Internet Protocole) de la famille des protocoles TCP/IP. L'IP a permis le convoyage ou la transmission sur le même signal les types de donnée : texte, image et vidéo, parole et son).

INTRODUCTION SYSTEME DE GESTION DES BASES DE DONNEES (SGBD)

D'une économie industrielle où prévalaient les matières premières et les ressources physiques, on passe à une économie informationnelle. L'information, fondement de toute décision doit désormais être traitée comme une ressource à part entière, c'est-à-dire gérée, contrôlée et organisée.

Un secteur quaternaire est quasiment né : celui des travailleurs de l'information avec un rôle transversal pour tous les secteurs d'activités et qui s'inscrit dans une dynamique double:

- l'accroissement de la complexité technique et organisationnel d'une part et d'autre part la fusion entre le monde informatique et le monde managériel.
- la réduction historique du temps décisionnel, depuis le rythme des saisons, les périodes lunaires, les heures de clochers, les secondes de l'électronique jusqu'aux nanosecondes de l'informatique transistorisée, par la maîtrise des SGBD voire des entrepôts de données (DataWarehouse).

Les BD, aujourd'hui, ont pris une place essentielle dans les systèmes informatiques et plus singulièrement en gestion. Elles représentent le creuset des efforts conjoints issus des Systèmes d'exploitation (SE), de l'intelligence artificielle, des systèmes repartis, du multimédia, etc.

Elles sont actuellement au cœur du système d'information des entreprises.

Les SGBD initialement disponibles sur les mainframes peuvent maintenant être installés sur tous les types d'ordinateurs y compris les PC et les Mac.

Avant l'arrivée des bases de données on utilisait les données organisées en fichiers fortement dépendant des traitements.

I- LIMITES A L'UTILISATION DES FICHIERS (SGF : système de gestion de fichier)

Dans les systèmes de traitements de données classiques, chaque application (fonction de traitement de l'entreprise), gère son propre ensemble de données utiles, ses propres fichiers. C'est l'approche utilisant directement un SGF. Dans toute entreprise modélisée avec une telle approche, il y a autant de couples (application, données propres/fichiers) que d'utilisations différentes de données.

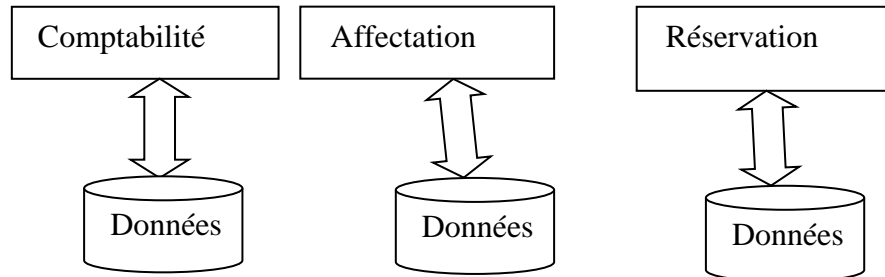
L'utilisation des fichiers impose à l'utilisateur de connaître l'organisation séquentielle, indexée, etc. des fichiers qu'il utilise afin de pouvoir accéder aux informations dont il a besoin puis d'écrire des programmes pour pouvoir effectivement manipuler ces informations.

Pour de nouvelles applications, l'utilisateur devra écrire un nouveau programme et ainsi, il est souvent emmené à créer de nouveaux fichiers qui contiennent des informations déjà présentes dans d'autres fichiers.

Il en résulte les conséquences suivantes :

- risque élevé de floraison de fichiers (de données) et de programmes disparates, redondants, difficilement compatibles et maintenables.
- application rigide (couplage fort), contraignante, longue et coûteuse, difficile à mettre en œuvre.
- données mal définies, peu fiables, peu accessibles de manière ponctuelle
- manque de corrélation et d'intégration.

- danger d'incohérence et d'intégrité des données
- difficultés pour la mise en œuvre des traitements ad hoc (non prévus à priori).
- toute modification de la structure des données nécessite ou implique la réécriture du programme donc données et traitements sont intrinsèquement ou intimement liés. (Couplage fort entre données et traitements) ;
- de déperdition (perte) de stockage, car une même donnée pouvant appartenir à Plusieurs applications.



Approche logique basée sur un SGF

Les données existantes ne peuvent donc pas être exploitées efficacement.

II-APPROCHE BD

1-Description

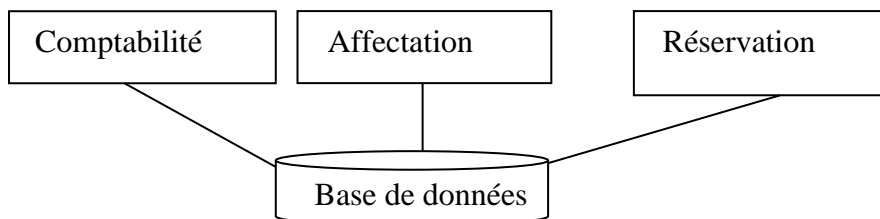
On a recherché des solutions prenant en compte les désirs (souhaits, besoins) des utilisateurs et les progrès technologiques. A ce titre, l'approche BD correspond à une triple évolution :

- l'évolution des entreprises (volume important des données, distribuées, centralisées ou réparties qui doivent être accessible en temps utile)
- l'évolution technologique (accroissement des performances du microprocesseur, intégration des composants, diminutions des coûts, miniaturisation due à VLSI, ULSI).
- l'évolution des systèmes d'exploitation et des architectures du système informatique.

On note l'extension logicielle et du matériel initial. Les SE réalisent une machine virtuelle très puissante qui définit un environnement pour des langages de haut niveau, architecture réseaux client-serveur voire n-tiers et, combinant de façon transparente des machines et des applications hétérogènes ou encore des ressources réparties et hétérogènes du point de vue matériel et logiciel).

Dans cette approche un réservoir commun de données est constitué. Ce réservoir représente une modélisation aussi fidèle que possible de tout le réel organisationnel qu'est l'entreprise. L'idée principale de l'approche BD réside dans l'intégration de toutes les informations de l'entreprise dans un réservoir unique où les applications viendraient y puiser les données les concernant. Ici, les traitements ad hoc sont envisageables, pas de duplication des données sauf cas particulier d'implantation distante relevant d'une question de sécurité.

Il n'y a plus de déperdition (perte) de stockage, ni de floraison de données, source d'incohérence.



Approche BD

2-Avantages de l'approche BD

Cette approche nous donne les principaux avantages. Ce sont :

- intégration et corrélation (cohérence des données, autrement dit correspondance entre données stockées et données réelles, centralisation des contrôles associés) ;
- flexibilité et indépendance (logique et physique) ;
- disponibilité et partageabilité (temps de réponse court, utilisation simultanée transparente aux utilisateurs, accessibilité) ;
- sécurité et fiabilité accrues (protéger l'unique réservoir (BD) contre tout accès invalide et illégal, mise en œuvre de l'intégrité, l'authentification, la confidentialité, la non répudiation, la tolérance de pannes) ;
- couplage faible entre base de données et traitements applicatifs inhérents à cette base de données.

3- Concepts de base relatifs au SGBD

-Lien entre SGBD et BD

Un SGBD (DBMS) permet à un utilisateur de communiquer avec une BD pour :

Décrire, spécifier et organiser les données sur les mémoires secondaires, (définir et modifier au niveau structurel) ; Manipuler les données (Rechercher, calculer, sélectionner, modifier...)

Un SGBD donne la possibilité à l'utilisateur de manipuler les représentations abstraites des données indépendamment de leurs organisations et de leurs implantations sur des supports physiques.

Un SGBD peut être considéré comme un interpréteur de langage de programmation de haut niveau qui, dans le cas idéal, permet à l'utilisateur de décrire précisément ce qu'il veut obtenir et comment l'obtenir.

-Base de données (Data base)

De façon formelle, on peut considérer une BD, comme une collection de données centralisée ou non, organisée de façon cohérente, servant pour les besoins d'un ou plusieurs utilisateurs.

C'est aussi un ensemble de données organisé de manière cohérente modélisant les objets d'une partie du monde réelle qui va servir de support à des applications informatiques.

Elle sous entend un type de données modélisant une partie du système d'information de l'entreprise.

C'est aussi un ensemble de données structuré, organisé de même manière cohérente et interrogeable.

Une base de données est un ensemble de données organisées qui modélise une partie du Système d'information et qui va servir de support à des applications informatiques.

On appelle aussi base de données, l'ensemble de données informatiques associées à un schéma de données et physiquement stockées en mémoire.

Remarque

Une démarche de conception de base de données doit conduire à la définition de données permanentes nécessaires aux besoins d'un ensemble d'utilisateurs. Cette base de données est relative à un domaine d'activité, à la gestion et au contrôle auquel elle doit contribuer ou même parfois qu'elle doit automatiser. Un tel domaine, dit domaine d'application de

gestion, pourrait être (ou désigner) une bibliothèque, un service du personnel, le département marketing d'une entreprise, la production de l'énergie, un Système d'information géographique (SIG), le traitement des infractions du code de la route ...

-SGBD (DBMS: Data base management system)

Quant aux SGBD, il peut être vu comme un ensemble de logiciels qui prend en charge la structuration, le stockage, la mise à jour et la maintenance des données.

Il constitue en fait une interface entre la base de données et les utilisateurs et programmes d'application.

En fait, le SGBD peut être aussi perçu comme un ensemble de logiciels systèmes permettant aux utilisateurs d'insérer, de modifier et de rechercher des données spécifiques dans une grande masse d'informations.

En résumé, un SGBD être vu comme un outil informatique permettant la sauvegarde, l'interrogation, la mise en forme des données structurées sur les mémoires secondaires.

Ils se distinguent des SGF par le fait qu'ils permettent la description des données (définition de type, des formats, de caractéristiques et parfois des opérations de manière séparée de leurs utilisations).

Le SGBD est le logiciel/matériel qui permet de gérer, manipuler et contrôler une BD.

Par gestion d'une BD, on entend création et maintenance de données (schémas, répertoires), optimisation des performances (prises de statistiques, définition de chemins d'accès rapides, évaluation de temps de réponses aux questions), maintien de la cohérence des données en cas d'erreur ou de panne (tolérance de panne) ou de malveillance.

Le SGBD offre une interface utilisateur « non procédurale » qu'on peut illustrer par l'image suivante : Supposons qu'un individu X arrive dans l'aéroport d'une ville inconnue et souhaite se rendre à une adresse Y. X choisit de prendre un taxi. Il va donner l'adresse Y au chauffeur et laisser à ce dernier le soin de trouver le meilleur chemin pour se rendre à la destination désirée dans les meilleurs délais. Les SGBDR sont de ce type de SGBD.

Exemple de SGBD : DB2, MICROSOFT ACCESS, MICROSOFT SQL Server ORACLE (Leader dans le domaine des SGBD, Forms builder pour les IHM), Progress, Sybase, Hyperfile de Windev (dont le langage WLanguage pour la construction des IHM ou interfaces).

SGBD = ensemble de logiciels qui va permettre de structurer, définir, manipuler les données de la BD.

Notons que les moteurs des SGBD évoluent dans deux directions complémentaires induites par de nouvelles niches applicatives des bases de données (besoins des entreprises):

- l'évolutivité (scalabilité) des serveurs avec une croissance en puissance attendue (vitesse en temps de calcul).
- l'extensibilité des types par les utilisateurs (traduisant de nouveaux besoins).

Cette évolutivité va de pair avec celle :

- du stockage bientôt dans l'ordre de Exa octet (EO) ; (YO>ZO >EO>PO> TO>GO>MO),
- du client de plus en plus léger (téléphone mobile internet, Personnel Digital Assistant (PDA)...).

REMARQUE

-Banque de données

Une base de données sous entend un type de données « factuelles » ou « primaires » alors qu'une banque de données, un type de données « référentielles » ou « secondaires ». L'information n'est pas directement obtenue après requête comme c'est le cas avec une base de données, mais des références à cette information nous sont dérivées. Elle peut inclure ou intégrer des bases de données.

Ex : Moteur de recherche (Google), recherche documentaire.

Les systèmes documentaires ont pour fonction première d'offrir comme information une indirection sur un texte (dans un livre, un article...) contenant le résultat recherché, alors qu'un SGBD (BD) fournit directement ce résultat.

Posée à une banque de données de savoir l'état de santé du président des USA Wilson de - - 1914 à 1918 ? Cette date est donnée à la page 52 du livre contemporain Histoire contemporaine des USA édition 1980.

A une base de données cette réponse serait directe en donnant l'état « fou ».

-Datawarehouse : Entrepôt de données ou système d'information décisionnel. Il est alimenté en général par des bases de données dites bases de production. On a la possibilité de produire des données statistiques et d'élaborer une historisation des données. Il s'agit d'un système ayant pour support des bases de données (bases de production) pour générer ou dériver d'autres données ou informations à titre décisionnel (statistique).

Données ouvertes (open data)

Big Data : Le terme Big Data décrit des ensembles de volumes massifs de données à la fois structurées et semi-structurées (plusieurs réponses d'approximations sont données en conclusion en tendance vers la vérité. Algorithmique avec des degrés de complexité linéaire, polynomiale, exponentielle).

science de données (qualitative, quantitative) : analyse des données suivie de l'extraction de connaissance d'ensembles de données. Elle emploie des techniques et des théories multidisciplinaire (ou interdisciplinaire)

III- ORGANISATION EN COUCHES ET OBJECTIFS D'UN SGBD

1- ORGANISATION EN COUCHES

Un SGBD se compose en une approximation de trois (3) couches emboîtées de fonction depuis les mémoires secondaires vers les utilisateurs. Ce sont le gestionnaire de fichiers, le SGBD interne et le SGBD externe.

1.1- Gestionnaire de fichiers

La gestion des récipients de données sur les mémoires secondaires constitue traditionnellement la première couche. Ce gestionnaire de fichiers est encore appelé système de fichier. Celui-ci fournit aux couches supérieures des mémoires secondaires idéales, adressables par objet et capables de rechercher des données via des critères.

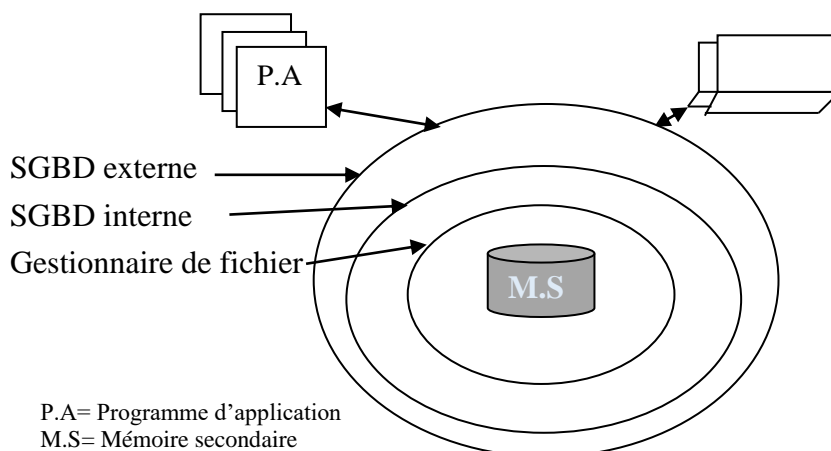
NB : Un fichier est un ensemble d'informations de même nature.

1.2- SGBD interne

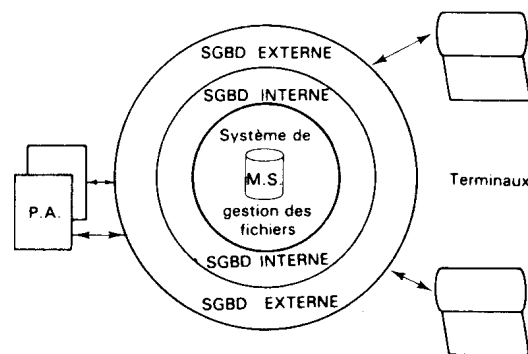
La gestion des données stockées dans les fichiers, l'assemblage de ces données en objets, le placement de ces objets dans les fichiers, la gestion des liens entre objets et les structures permettant d'accélérer les accès aux objets constituent la 2e couche (SGBD interne ou système d'accès aux données). Celui-ci repose généralement sur un modèle de données interne constitué par des câbles reliés par des pointeurs.

1.3- SGBD externe

La fonction essentielle de cette 3e couche consiste à la mise en forme et à la présentation des données aux programmes d'application et aux utilisateurs interactifs. Ceux-ci expriment leurs critères de recherche à l'aide du langage basé sur les procédures de recherches progressives ou sur des assertions logiques, en référençant des données dérivées de la base. C'est ce SGBD externe qui va assurer la transformation des données extraites de la BD échangée avec le monde extérieur.



Structure d'un SGBD en couches



Différentes couches d'un SGBD

Remarque

Ces couches de fonctions constituent sans nul doute environ la moitié du code d'un SGBD. En effet, au-delà des fonctions de recherche, de rangement et de présentation, un SGBD gère des problèmes difficiles de partage (exclusion mutuelle) et de cohérence des données. Il protège aussi les données contre les accès non autorisés. Ces fonctions qui peuvent paraître secondaires ou annexes sont souvent plus difficiles à réaliser et nécessitent beaucoup de code. Pour être complet notons qu'au dessus des SGBD, de plus en plus, aujourd'hui, des ateliers de génie logiciel permettent de modéliser des ensembles de données en BD et de représenter les traitements associés à l'aide d'interfaces graphiques et de

langage de spécification. Ces outils d'aide à la conception bien que non intégrés forcément dans le SGBD permettent de spécifier les descriptions des données. La puissance de PC et la mémoire à bon marché, l'approche orienté objet favorisent les SGBD à intégrer des objets beaucoup plus complexes répondant au besoin de nouvelles applications de types multimédia, web, environnement, télécommunication, audiovisuel, etc.

2- OBJECTIFS D'UN SGBD

Le principal objectif d'un SGBD est d'assurer l'indépendance des programmes aux données, c'est-à-dire la possibilité de modifier des schémas conceptuels et internes des données sans modifier les programmes d'application. Lors des modifications des structures logiques et physiques plus précisément, on distingue l'indépendance physique qui permet de changer les schémas internes sans changer les programmes et vis versa et l'indépendance logique qui permet de changer le modèle conceptuel sans modifier les programmes.

Les SGBD conduisent à mettre en commun les données d'une entreprise, ou du moins d'une application dans une base de données décrite par un dictionnaire de données. Cela ne va sans poser des problèmes d'efficacité notamment les accès simultanés aux données souvent situées sur le même disque. La base peut ainsi devenir un goulot d'étranglement. L'environnement multi usager de gérer les mises à jour concurrentes de sorte à garantir la cohérence des données. Notamment des données redondantes contrôlées doivent rester égales.

2.1 Classification des objectifs

Les objectifs d'un SGBD peuvent être classés en deux groupes dont le deuxième pouvant se déduire du premier.

Le premier comprend les objectifs suivants :

- Indépendance physique des programmes et données ;
- Indépendance logique des programmes aux données ;
- Manipulation des données par des langages non procéduraux (non informaticiens) ;
- Administration facilitée des données.

Le second groupe porte sur des objectifs additionnels à savoir :

- Efficacité des accès aux données ;
- Partageabilité des données ;
- Cohérence des données ;
- Redondance contrôlée des données ;
- Sécurité des données.

Quelques précisions sur certains objectifs :

-non redondance des données : saisir et mettre à jour plusieurs fois les mêmes données (éviter des MAJ non coordonnées).

-cohérence des données : vérification des contraintes d'intégrité et de la cohérence de la base ; les contraintes doivent être exprimées et gérées dans la base et non dans les applications.

-gestion de la concurrence d'accès aux données : plusieurs personnes ou applications doivent pouvoir accéder à la base de données en même temps. (Transaction ACID en vue de la gestion des susceptibilités de conflits d'accès, système de verrouillage d'accès aux données partagées en surtout modification multi-utilisateur).

-sécurité des données : protection des données contre les accès non autorisés ou mal intentionné ; il doit exister des mécanismes permettant d'autoriser, contrôler et enlever des droits d'accès à des utilisateurs ; Il faut tolérer les pannes (système de tolérance des fautes et des pannes) : capacité à faire passer la base de données d'un état cohérent à un autre état cohérent même s'il survenait une panne (électrique). Le système doit permettre un temps de reprise des données et de continuité de service le temps d'une panne.

2.2 Description des neufs (9) objectifs

2.2.1-Indépendance physique

Il s'agit de l'indépendance des programmes et des données (MCD, MLD, MPD) vis-à-vis des divers supports de stockage.

Les données élémentaires ou atomiques sont assemblées pour décrire les objets et les associations entre objets perceptibles dans le monde réel. Bien qu'en général, deux groupes de travail assemblent différemment ces données, il est possible au sein d'une entreprise bien organisée de définir une structure canonique des données (modèle conceptuel). C'est-à-dire un partitionnement en ensembles et sous ensembles ayant des propriétés bien définies et cohérentes avec les vues particulières. Cet assemblage peut être considéré comme l'intégration de ces vues. Il obéit à des règles qui traduisent l'essentiel des propriétés des données du monde réel. Cet assemblage correspond à un schéma conceptuel d'une BD.

Cet objectif va permettre de réaliser l'indépendance des structures de stockage aux structures de données du monde réel ; c'est-à-dire les schémas interne et conceptuel.

Bien que ces deux schémas décrivent les mêmes données à des niveaux différents, il s'agit donc de pouvoir modifier le schéma interne sans avoir à modifier le schéma conceptuel en tenant compte seulement des critères de performance et de flexibilité d'accès.

On pourra par exemple ajouter un index, regrouper deux fichiers en un, changer l'ordre ou le codage des données sans mettre en cause les entités et associations définies au niveau conceptuel.

2.2.2-Indépendance logique

D'une manière générale, un ensemble de données peut être vu comme une intégration de plusieurs centres d'intérêts. Ainsi, il est vu de manière distincte par différents groupes d'utilisateurs ; toutes ces visions spécifiques des données doivent être intégrées dans une vision globale. Le schéma conceptuel modélisant les objets et association entre objets du monde réel résulte d'une synthèse de vues particulières de chaque groupe de travail utilisant la BD, c'est-à-dire une intégration de schémas externes. En conséquence, chaque groupe de travail réalisant une application, doit pouvoir assembler différemment les données pour les former les entités et associations de son schéma externe, ou plus simplement des tables qu'il souhaite visualiser. Ainsi chacun doit pouvoir se concentrer sur les éléments constituant son centre d'intérêt, c'est-à-dire que chacun doit pouvoir ne reconnaître qu'une partie des données et ne voir que cette partie.

Il est donc souhaitable de permettre une certaine indépendance des données vues par les applications à la structure canonique des données de l'entreprise décrite dans le schéma conceptuel. L'indépendance logique est donc la possibilité de modifier un schéma externe sans modifier le schéma conceptuel. Cette indépendance est aussi assurée entre les différents utilisateurs (eux mêmes), chacun percevant une partie de la BD via son schéma externe selon une structuration voire un modèle particulier.

L'indépendance logique donne les avantages suivants :

- Permettre à chaque groupe de travail de voir les données comme il le souhaite
- Permettre l'évolution de la vue de chaque groupe de travail et de la vue de canonique de l'ensemble sans remettre en question au moins dans une certaine mesure le schéma conceptuel de l'ensemble.
- Permettre l'évolution d'un schéma externe sans remettre en cause les autres schémas externes.

En somme, il doit être possible d'ajouter des attributs, d'en supprimer d'autres ou de supprimer des entités dans des schémas externes mais aussi dans le schémas conceptuel (dans une certaine mesure) sans modifier la plus grande partie des applications, d'ajouter et de supprimer des ensembles d'associations.

2.2.3-Manipulation des données par des non-informaticiens

Les utilisateurs des données, parfois non professionnel de l'informatique doivent pouvoir manipuler facilement les données, c'est-à-dire les interroger, les mettre à jour sans préciser les algorithmes d'accès. Plus précisément, si les deux premiers objectifs sont atteints, les utilisateurs voient les données indépendamment de leur implantation en machine. De ce fait, ils doivent pouvoir manipuler les données au moyen de langages non procéduraux (en décrivant les données qu'ils souhaitent retrouver et mettent à jour).

Les langages non procéduraux (SQL) sont basés sur des assertions de logique de premier ordre. Ils permettent de définir les objets désirés aux moyens de relations entre les objets et les propriétés de ces objets.

Deux sortes d'utilisateurs manipulent la BD : les utilisateurs interactifs et les programmeurs. Les utilisateurs interactifs peuvent interroger, mettre à jour la BD. Ils sont parfois non informaticiens et réclament les langages de commande semi-formels jusqu'aux langages graphiques en passant par les interrogations par menus ou par formes.

Les programmeurs écrivent des programmes en utilisant des langages de 3e génération ou dits évolués (C, Ada, Cobol, Fortran, etc.), des langages plus récents, orientés objet (Java, C++) et des langages de 4e génération (Microsoft Visual Basic, Forte for java, Netbean, Eclipse, Jcreator, Jbuilder, PL/SQL). Ces derniers regroupent des instructions de programmation structurées, des expressions arithmétiques et logiques des commandes d'accès à la BD et des commandes d'édition et d'entrées de messages (menu déroulant, fenêtres, rapports imprimés, etc. Ils sont de plus en plus orientés objet (pour mieux simuler la réalité).

Dans tous les cas, il est important que le SGBD fournisse ou intègre des commandes nécessaires de manipulation de données par une intégration harmonieuse de langages de programmation pour des fins de traitements d'objet souhaités.

2.2.4-Administration facilitée des données

Un SGBD doit fournir des outils pour décrire les données, à la fois au niveau des structures de stockage et de leur représentation externe. Il doit permettre aussi le suivi de l'adéquation de ces structures au besoin des applications et autoriser leur évolution aisée ou en douceur. Les fonctions qui permettent de définir les données et de modifier leurs définitions sont appelés outils d'administrations des données.

Afin de permettre un contrôle efficace des données, de résoudre les conflits entre divers points de vue pas toujours cohérents, de pouvoir optimiser les accès aux données, et l'utilisation des moyens informatiques, on a pensé à centraliser ces fonctions entre les mains d'un petit groupe de personnel hautement qualifié appelé administrateur de BD. En fait, une telle centralisation a souvent conduit à des difficultés d'organisation.

L'évolution des SGBD modernes tend alors à fournir des outils permettant de décentraliser la description des données tout en assurant une cohérence entre les diverses descriptions partielles. Ainsi, pour aider les concepteurs de BD, un dictionnaire de données dynamique (méta base) est souhaité. Cette évolution rapide va dans le sens du développement d'outils (orientés objet et web) intégrés capables de faciliter l'administration des données et d'assurer la cohérence des descriptions. (Exemple d'outil d'administration du SGBD oracle : Toad 8).

2.2.5-Efficacité de l'accès aux données

Ces langages (ceux dont on a parlé) doivent permettre d'obtenir des réponses aux interrogations dans un temps raisonnable. Ils doivent donc être optimisés et aussi avoir un mécanisme permettant de minimiser le nombre 'accès disques'. Tout ceci est fait de façon transparente par (pour) l'utilisateur. Les performances en termes de débit (nombre de transactions par secondes) et de temps de réponse (temps d'attente moyen pour une requête type) sont des problèmes clairs des SGBD. L'objectif de débit élevé nécessite un overhead minimal (répartition générale des tâches) de la gestion des tâches accomplies par le système. L'objectif de bon temps de réponse implique qu'une requête courte d'un utilisateur n'attend pas une requête longue d'un autre utilisateur (mais sans abus de traitements courts ou longs).

Il faut donc partager les ressources (UC, unités d'entrée/sortie) entre les utilisateurs en optimisant l'utilisation globale et en évitant les pertes de commutation de contexte. Le goulot d'étranglement essentiel dans un SGBD reste les entrées/sorties(E/S) disques.

Une E/S coûte en effet quelques dizaines de millisecondes. Afin de les éviter ou les minimiser, on utilisera une gestion de tampon en mémoire centrale dans de véritables mémoires caches de disque afin qu'un grand nombre d'accès aux données se fasse en mémoire.

Un autre facteur d'accès limitatif est relatif (ou dû) à l'utilisation des langages non procéduraux (puissants mais ...) afin d'interroger et mettre à jour la BD.

En résumé, un SGBD devra chercher à optimiser une fonction de coût de la forme suivante :

$$C(Q) = a \times UC(Q) + b \times E/S(Q) \quad ; \quad Q \text{ étant ensemble typique de requêtes, } E/S(Q) \text{ est le}$$

nombre d'entrée/sortie, UC(Q) est le temps UC dépensé. a et b sont les facteurs convertissant respectivement temps UC(Q) et E/S(Q) en coût.

2.2.6-Redondance contrôlée des données

Dans les systèmes classiques à fichiers non intégrés, chaque application possède ses données propres. Ceci conduit généralement à de nombreuses duplications de données avec, outre la perte en mémoire secondaire associée, un gâchis important en moyens humains pour saisir et maintenir à jour plusieurs fois les mêmes données.

Avec une approche BD, les fichiers plus ou moins redondants sont intégrés en un seul fichier partagé par les applications. Ce qui conduit naturellement à la non duplication physique des données afin d'éviter les mises à jour multiples.

Avec l'avènement des BD surtout réparties sur plusieurs calculateurs interconnectés (agents, sites, machines), il est apparu plus souhaitable de faire gérer la base de données par le système des copies multiples de données (dans une certaine mesure une copie à des nœuds du réseau réduit la durée d'accès à une page web dynamique par exemple (1^{ère} fois lancée, arrêt, et 2^{ème} accès plus rapide, DNS, BDR).

Cela optimise les performances en interrogations en évitant les transferts sur le réseau et en permettant le parallélisme d'accès.

On considère aujourd'hui que la redondance gérée par le SGBD au niveau physique n'est donc pas forcément mauvaise. Il faudra par contre éviter la redondance anarchique, non connue du système, qui conduirait les programmes utilisateurs à devoir mettre à jour plusieurs fois une même donnée. Il s'agit donc de bien contrôler la redondance qui permet d'optimiser les performances, en la gérant de manière invisible (transparente) pour les utilisateurs. (RAID5 technologie permet la redondance physique des données pour des questions de sécurité)

2.2.7-Cohérence des données

Bien que les redondances anarchiques de données soient évitées par l'objectif précédent, les données vues par l'utilisateur ne sont pas indépendantes. Au niveau des ensembles de données, il peut exister une certaine dépendance entre données, par exemple une donnée représentant le nombre de commandes d'un client doit correspondre au nombre de commandes de (dans) la base. Plus simplement une donnée élémentaire doit respecter un format et ne peut prendre une valeur quelconque.

Exemple : Un salaire mensuel devant être compris entre 5000 f et 12000 f.

Un SGBD doit donc veiller à ce que les applications respectent ces règles lors des modifications de données et doit ainsi assurer leur cohérence. Ces règles et bien d'autres (CIF) doivent implicitement et explicitement suivre les données au cours de leur évolution sont appelées des contraintes d'intégrité.

2.2.8-Partage des données ou partageabilité des données

L'objectif est de partager les données de la BD dans le temps et également simultanément. Une application doit pouvoir accéder aux données comme si elle était seule à les utiliser sans attendre et aussi sans savoir qu'une autre application peut les modifier concurremment. En pratique, un utilisateur exécute des programmes généralement courts qui consultent et mettent à jour la BD. De tels programmes interactifs appelés transactions correspondent par exemple à l'entrée d'un produit en stock ou à une réservation d'une place dans un avion. Il est important que deux transactions concurrentes (2 réservations sur le même avion) ne s'emmêlent pas dans leur accès à la BD autrement dit qu'on ne puisse pas réserver le même siège pour deux passagers.

On cherche donc à assurer que le résultat d'une exécution simultanée de transactions reste le même que celui d'une exécution séquentielle dans un ordre quelconque de ces transactions. (Exclusion mutuelle dans les sections critiques, verrouillage/déverrouillage, mutex, sémaphore, moniteur comme en système d'exploitation et même l'utilisation de message).

2.2.9-Sécurité des données

Cet objectif a deux aspects.

Tout d'abord, les données doivent être protégées contre les accès non autorisés ou mal intentionnés. Il doit exister des mécanismes donc adéquats pour autoriser, contrôler ou enlever les droits d'accès pouvant dépendre de la valeur des données ou des accès précédemment effectués par les usagers. Par exemple, un employé pourra connaître les salaires des employés qu'il dirige mais pas celui de ses supérieurs.

D'un autre côté, la sécurité des données doit aussi être assurée en cas de panne d'un programme ou du système, voire de la machine. Un « bon » SGBD doit être capable de restaurer les données cohérentes après une panne disque, à partir des sauvegardes par

exemple via au recours à la technologie RAID5 (Redundant Array of Inexpensive Disks). A ce niveau, il est aujourd'hui possible, de prévoir des sauvegardes sur sites distants (sauvegarde à distance).

Une bonne gestion des transactions est celle devant pouvoir faire passer une BD d'un état cohérent à un autre état cohérent. Aussi, si une transaction commence (mise à jour d'un transfert depuis un compte bancaire sur un autre) et est interrompue par une panne pendant cette mise à jour (même après avoir débité votre compte), le SGBD doit assurer l'intégrité de la base (la somme d'argent gérée doit rester constante) et par la suite défaire la transaction qui a échoué (rollback sous oracle).

Une transaction doit être totalement exécutée ou pas du tout. C'est la propriété d'atomicité de la transaction prise dans un ensemble **de propriétés ACID** (Atomicité, Cohérence, Isolation, Durabilité).

En assurant l'atomicité des transactions, on contribue à garantir l'intégrité physique de la BD.

Cette sécurité intègre de plus en plus des systèmes cryptographiques et de chiffrement afin de parvenir à la confidentialité, à l'intégrité des données d'une part et d'autre part à l'authentification de l'utilisateur et à la non répudiation. Il est même fait usage d'éléments plus élaborés de sécurité notamment les signatures et certifications numériques.

[Cryptologie = cryptographie plus cryptanalyse ; Cryptographie symétrique et Cryptographie asymétrique où on développe le PKI ou ICP].

IV- MODELISATION DES DONNEES

Il s'agit de la description, de la spécification et de l'organisation des données sur les mémoires secondaires en produisant des modèles et des schémas.

Une base de données peut être pour certains, une collection de fichiers collés par des pointeurs multiples, aussi cohérents entre eux que possible, organisés de manière à répondre efficacement à une grande variété de questions.

Pour d'autres, une base de données peut paraître comme une collection d'informations modélisant une entreprise du monde réel.

Ainsi un SGBD peut donc paraître, à la fois comme un ensemble de logiciel système permettant de stocker et d'interroger un ensemble de fichiers interdépendants et également de gérer les données d'une entreprise.

Une idée centrale des bases de données est la séparation de la description des données effectuées par les administrateurs, de la manipulation les données effectuées par les programmes d'applications ou les user.

La description permet de spécifier les structures et les types de données l'application alors que la manipulation consiste à effectuer des interrogations et des mises à jour.

1- Niveaux d'abstraction

1.1- Concepts de base pour l'abstraction (Type, instance, modèle et schéma)

La description de données consiste à définir les propriétés d'ensembles de données (bloc, regroupement, paquet, groupement) modélisés dans la base de données et non pas d'objets particuliers (occurrences).

Les objets particuliers sont définis par les programmes d'application (manipulation des données) et ils doivent réunifier (encapsuler) les propriétés des ensembles auxquels ils appartiennent.

Il convient de noter que le recueil d'informations nécessite de circonscrire le problème, d'identifier les acteurs et utiliser quelques procédés de recherche (observation, entretien, écoute, interview, revue documentaire, internet, bibliothèque ...). Déjà un traitement ou prétraitement a lieu pour constituer la liste des informations pour le MCD, s'entendre sur la sémantique des termes conceptuels relatifs au domaine étudié, avant de voir même les intérêts spécifiques des utilisateurs devant constituer un second groupe de traitements (celui souvent modélisé).

Deux notions essentielles se dégagent par la définition des objets : type et instance.

-Type d'objet : il permet de spécifier les propriétés communes à un ensemble d'objets en termes de structure de données visibles et d'opérations d'accès. Le type d'objets est un ensemble des objets possédant des caractéristiques similaires ou partageant des propriétés communes et manipulables par des opérations identiques.

-Instance d'objet ou instantiation ou occurrence : elle correspond à un objet particulier identifiable parmi les objets d'un type (aussi appelée individu ou population).

La description d'un ensemble de données particulières correspondant par exemple à une application donne naissance à un schéma de données.

On distingue généralement le schéma source spécifié par les administrateurs de données et le schéma objet résultant de la compilation du précédent par une machine. Le schéma objet est directement utilisable (intelligible) par le SGBD afin de retrouver, de vérifier les propriétés des instances d'objets (qui seront) manipulées par les programmes d'application.

Modèle : un modèle est une abstraction de la réalité. Il comprend les éléments pertinents de cette réalité décrits ou à décrire.

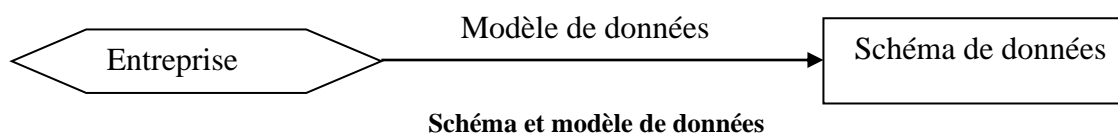
Schéma : C'est la description au moyen d'un langage déterminé, d'un ensemble de données particulières.

On retient que pour une base de données particulière, il existe un seul schéma interne et un seul schéma conceptuel. En revanche, il existe plusieurs schémas externes.

Schéma conceptuel (Modèle conceptuel) : c'est la description des représentations annonçant la machine (ordinateur), correspondant à une vue canonique globale d'une entreprise ou d'une partie de l'entreprise en termes de type d'objets à associer.

Modèle de données : Il doit être perçu comme un ensemble de structures et de données avec un ensemble d'opérations définies dessus et des mécanismes de contrôles associés correspondant respectivement aux termes **Définition, Manipulation, Contrôle**.

Schéma de données : C'est l'abstraction résultant de l'application d'un modèle de données à une entreprise.



NB : Modèle de description des données (data description model) : c'est l'ensemble des concepts et de règles permettant de décrire les données y compris le formalisme ou simplement leur représentation.

Le langage de description des données (data description language), c'est le langage supportant un modèle et permettant de décrire les données d'une base d'une manière assimilable par une machine.

Remarque

a- Le schéma est considéré comme un modèle plus opérationnel, pragmatique. Le schéma peut être dérivé du modèle.

b-Approche descendante ou ascendante de définition d'un schéma conceptuel

Le modèle conceptuel de données permet le passage d'un concret inaccessible (l'univers réel) à un abstrait manipulable : schéma conceptuel.

➤ Approche descendante de la conception d'un schéma conceptuel selon l'architecture de l'ANSI/X3/ SPARC

Cette approche consiste à :

- établir le **schéma conceptuel** comme corrélation aussi fidèle que possible à l'entreprise ;
- puis à dériver les schémas externes orientés utilisation et les schémas internes orientés performance d'implantation machine.

➤ Approche ascendante de la conception d'un schéma conceptuel selon Chen

Cette approche consiste à :

- définir d'abord les **schémas externes** pour coller au mieux les besoins utilisateurs, puis on dérive successivement les schémas conceptuels et internes. Les problèmes inhérents à cette pratique sont :

- difficulté de la définition d'un consensus entre les schémas externes ;
- absence de perception formelle et cohérente de l'entreprise prise dans sa globalité etc.

Au vu de la complexité du système à décrire et des difficultés de l'approche ascendante, on est amené à retenir l'approche descendante qui est beaucoup plus naturelle dans son appréhension du réel. En effet la conception d'une base de données (SGBD) devra suivre les quatre étapes suivantes :

- expression de la sémantique de la représentation du monde réel (niveau d'abstraction conduisant à la définition du schéma conceptuel) par des données et les liens entre ces données ;
- traduction du schéma conceptuel dans un modèle de données existant (hiérarchique, en réseau, relationnel, objet..) ;
- expression de l'utilisation (définition des vues, des accès logique et chemins...) ;
- expression de l'implantation en machine (choix des méthodes d'accès, des chemins d'accès..) permettant d'assurer de bonnes performances d'exploitation).

[page 37.....]

c- Caractéristiques d'un bon modèle ou schéma conceptuel (en polycopé).

Un modèle est une abstraction de la réalité. Une bonne représentation à travers ce modèle recherche l'équilibre entre la densité d'informations et la lisibilité en vue d'apporter une solution idoine à une problématique donnée. Il doit offrir les caractéristiques suivantes : puissance de représentation du réel, stabilité et flexibilité (ou évolutivité), simplicité (conception et utilisation) et base formelle.

-Puissance de représentation : il doit non seulement représenter les aspects structurels de l'entreprise modélisée mais aussi les aspects sémantiques et opératoires (tant au niveau de la description que de la manipulation des données) et notamment intégrer toutes les contraintes entre données existant dans l'univers réel (représenter aussi fidèle que possible la réalité, le perçu réel).

-Stabilité et flexibilité : Il doit être stable car devant apporter une solution durable à une problématique donnée. L'ajout d'une nouvelle donnée ou d'une nouvelle règle sémantique ne doit pas entraîner une modification sensible du schéma. Seul un changement de l'univers réel doit entraîner une mutation du modèle (schéma) et ceci sans effet de bord : réorganisation d'autres entités que celles concernées par le changement. Les fondamentaux demeurent donc. Le meilleur modèle est flexible et évolutif, car ses fondamentaux contiennent les éléments de son changement et de son adaptation au contexte dynamique sans disparaître totalement. Dans toute modélisation, il convient de veiller à atteindre un niveau (d'abstraction) d'invariabilité suffisant de sorte à garantir la stabilité au modèle

- *Simplicité* : cette propriété revêt deux aspects: la facilité de compréhension (entre pair de même domaine) et celle d'utilisation.

-La facilité de compréhension est relative à la conception du modèle. L'agrégation des données constituant le schéma doit être partagée par les pairs du même domaine. Toute symétrie existant dans l'univers doit apparaître.

-facilité d'utilisation porte sur le nombre réduit de primitives de manipulations pouvant être masquées. Ces primitives doivent être puissantes pour pouvoir exprimer des requêtes même complexes.

-Base formelle : la définition du schéma conceptuel ou modèle conceptuel doit s'appuyer sur une base rigoureuse, mathématique si possible, pour éviter toute ambiguïté d'interprétation et permettre une vérification formelle (complétude). Plus le modèle est formel, plus le schéma conceptuel correspondant sera fiable, stable, durable et viable.

1.2) Trois (3) niveaux d'abstraction : conceptuel, interne, externe

a-Niveau conceptuel

Il est le niveau central. C'est le dépositaire de la sémantique de la base des données. Ce niveau représente l'abstraction aussi fidèle que possible de l'univers modélisé (univers du discours). Il correspond à la structure canonique des données qui existent dans l'entreprise. C'est-à-dire, leur structure sémantique inhérente sans soucis d'implantation en machine et représentant la vue intégrée de tous les traitements (de tous les utilisateurs des applications). Il sert de référence globale aux vues particulières des schémas externes.

La définition du schéma conceptuel d'une application n'est pas un travail évident, cela nécessite un accord sur les concepts de base qui modélise les données.

Le schéma conceptuel permet donc de définir :

- les types élémentaires qui définissent les propriétés élémentaires de l'information dans l'entreprise ;
- les types de données composés qui permettent de regrouper les propriétés afin de décrire les objets du monde réel ou les relations entre ces objets ;
- les types de données composés qui permettent de regrouper les propriétés (attributs) afin de décrire les associations du monde réel.

- Eventuellement les règles que devront observer les données au cours de leur vie dans l'entreprise, ce sont les **règles de gestion**.

Le schéma conceptuel est défini en dehors de tout contexte informatique. Le schéma logique global est la traduction du schéma conceptuel par un modèle existant (hiérarchique, en réseau, relationnel, objet). (Schémas logique interne + logique externe). **P45**.

b- Niveau externe

Chaque groupe de travail utilisant des données possède une description des données perçues appelée schéma externe. Cette description est effectuée selon la manière dont le groupe voit la base via ce programme d'application. Ce niveau logique comprend les vues spécifiques définies pour la manipulation des données. Les besoins en données des utilisateurs sont ainsi exprimés à ce niveau à travers plusieurs schémas externes.

Autrement dit, une vue externe représente, d'une manière aussi fidèle que possible un type d'utilisation de la BD. C'est à ce niveau qu'agit l'utilisateur. Ses actions peuvent être de plusieurs types; un programme posant des requêtes ad hoc ou non et un autre posant des requêtes paramétrées ou non.

Alors qu'au **niveau conceptuel et interne les schémas décrivent une BD, au niveau externe, ils décrivent simplement la partie des données ayant simplement un intérêt pour un utilisateur ou un groupe d'utilisateurs.**

En conséquence, un schéma externe est aussi qualifié de vue externe. Le modèle externe utilisé est dépendant du modèle du langage de manipulation de la BD utilisé. Notons que la notion de schéma externe permet d'assurer une certaine sécurité des données décrites dans son schéma externe. Les autres données sont ainsi protégées des accès non autorisés, et surtout des utilisateurs mal intentionnés.

Certaines vues peuvent ne pas être construites mais déduites par calcul (ce niveau externe ci est abstrait ; production de statistiques).

NB : Le schéma externe, c'est la description d'une partie de la BD extraite ou calculée à partir de la BD physique correspondant à la vision du programme d'application ou d'un utilisateur, donc à un rangement particulier de certaines données.

L'ensemble des données est une vue intégrée de l'ensemble des traitements.(vue intégratrice)

c-Niveau interne

Il correspond à la structure de stockage supportant les données. La définition du schéma interne nécessite au préalable le choix d'un SGBD (Oracle, Microsoft Access, Progress, Sybase, DB2, Informix, Mysql, Microsoft SQLServer ...). Elle permet de décrire les données telles qu'elles sont stockées dans la machine :

- les fichiers qui les contiennent (nom ou identificateur, organisation, localisation)
- les articles de ces fichiers (langage, champ composant,...)
- les chemins d'accès à ces articles (table, index, chainage,...) (téléchargement inachevé d'un fichier d'une séquence vidéo donnera l'image mais pas l'entièreté des du film vidéo)

Ce niveau correspond à la représentation en machine aussi efficace que possible du schéma conceptuel. (L'efficacité doit tenir compte d'une part des contraintes d'implantation et d'autre part des critères d'utilisation)

NB : Schéma interne, c'est la description des données d'une BD en terme de représentation physique en machine correspondant à une spécification des structures de mémorisation et des méthodes de stockage et d'accès pour utiliser, ranger et retrouver ces données sur disque.

1.3-Synthèse des trois niveaux de schéma

Chacun de ces niveaux est caractérisé par un schéma, un modèle et un langage de définition et manipulation des données permettant de définir les données de ce schéma, de les décrire et de les manipuler.

Les fonctions ou rôles relatifs aux niveaux d'abstraction, peuvent être accomplis par la même personne ou par des personnes différentes. Un rôle essentiel est celui d'administrateur d'entreprise qui inclut la définition des informations que contient la BD au niveau sémantique par exemple s'appuyant sur des diagrammes entité-associations.

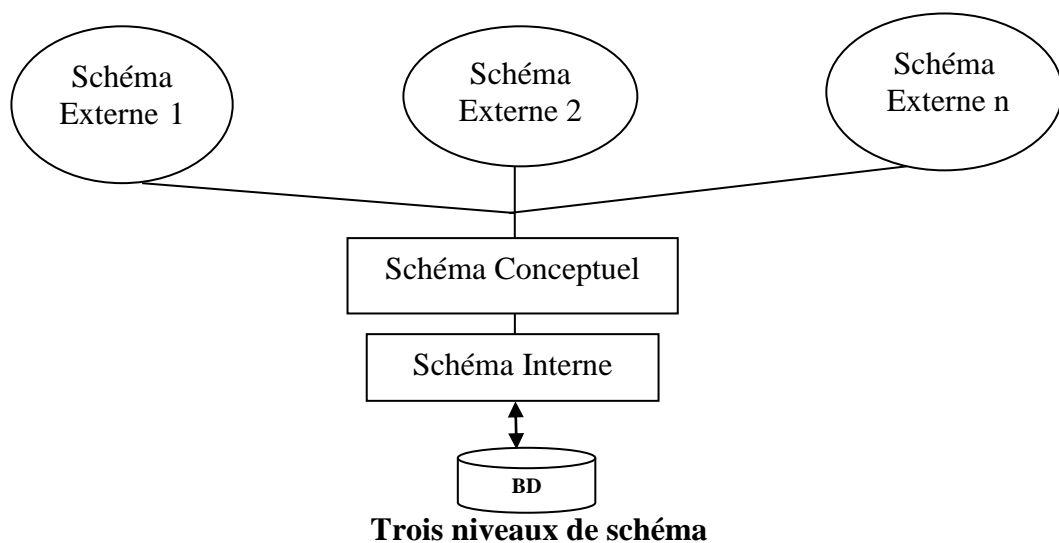
La plupart des SGBD modernes supportent seulement un schéma interne et plusieurs schémas externes. Le schéma conceptuel est défini en utilisant un outil d'aide à la conception (AGL) s'appuyant généralement sur des interfaces graphiques et permettant d'élaborer des diagrammes de type entités-associations.

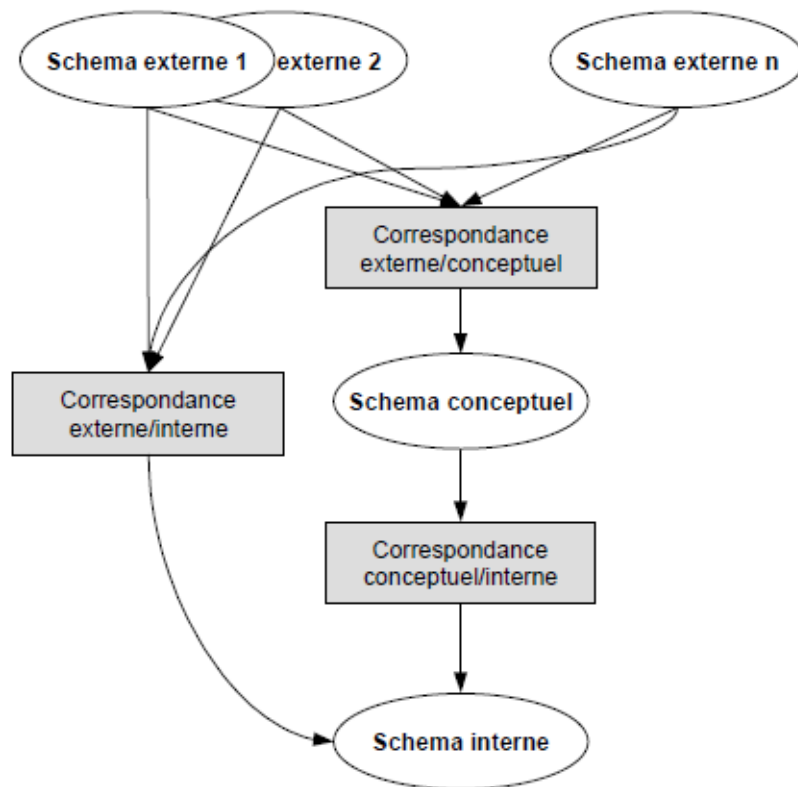
Quoiqu'il en soit, les différents schémas et procédures pour passer de l'un à l'autre, sont stockés dans le dictionnaire de données. Celui-ci peut être divisé en deux dictionnaires à savoir :

- le dictionnaire de l'entreprise qui contient le schéma conceptuel, les procédures et commentaires s'appuyant sur ce schéma.
- le dictionnaire des bases qui contient les schémas externes et internes ainsi que les procédures de passage de l'un à l'autre. (**catalogue**)

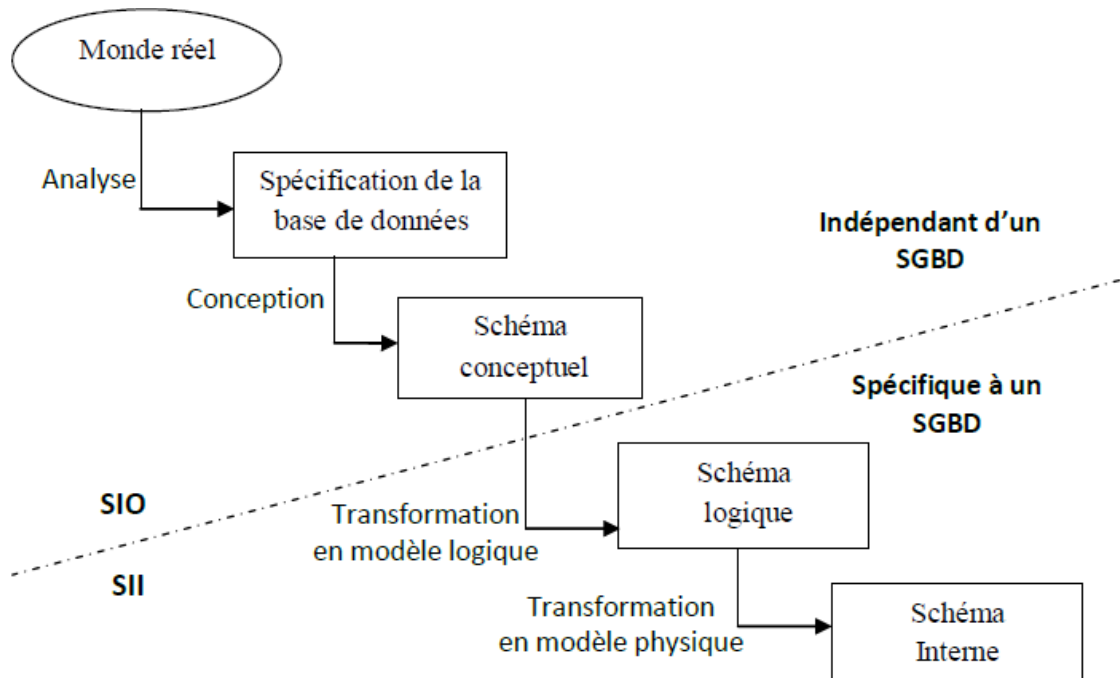
Tout dictionnaire contient en général des descriptions en langage naturel permettant de préciser la signification des données.

Un dictionnaire de données peut contenir des informations non strictement BD tels que les masques ou maquettes de données, d'écran ou de programmes. Les informations sont souvent en format source ou en format compilé. Tout ceci peut être spécifié dans un dictionnaire appelé **métabase** (base de données organisée sous forme de BD qui décrit toutes les autres bases).





Hiérarchie des modèles ANSI/X3/SPARC



Tout le processus de conception et mise en place d'une BD.

**Plusieurs cas d'implémentation SII sont possibles pour un seul cas de SIO.
PIM/PSM (Platform independent model / Platform specific model).**

Pour un PIM on a plusieurs PSM possible.

Concomitamment, trois cycles sont utilisés dans la conception et la mise en place d'une BD :

Cycle de décision,

cycle d'abstraction ou de raisonnement (modélisation)

et enfin le cycle de vie (ou de maîtrise).

2) Modèle entité/association : concepts et formalisme

Le modèle E/A est basé sur une perception du monde réel qui consiste à distinguer des agrégations de données élémentaires appelées entités et des liaisons ou liens entre ces entités appelées associations. Association et entités sont catégoriser en types.

-Entité : Une entité correspond à un objet du monde réel généralement définie par un nom (Voiture, Etudiant,...) C'est un regroupement de données élémentaires (propriété ou caractéristique spécifique).

-Type d'entité : Un type d'entité définit un ensemble d'entités constitué par des données du même type. Les types de données agrégés sont appelés les attributs de l'entité. En somme, une entité est un modèle d'objets identifiés du monde réel dont le type est défini par un nom et une liste de propriétés.

-Agrégation : C'est l'abstraction qui consiste à grouper des objets pour constituer des objets composés d'une concaténation d'objets composants. Une entité agrégation est une abstraction de données élémentaires.

-type d'association : Une association correspond à un lien logique entre deux entités ou plus. Elle est souvent définie par des attributs ou propriétés. Elle est alors porteuse de données. En résumé, une association est un lien logique entre entités dont le type est défini par un verbe et une liste éventuelle de propriétés.

-Attribut, Propriété ou caractéristique : Un attribut est une propriété d'une entité ou d'une association constituée par un nom (identificateur) et de type élémentaire, atomique (qu'on ne peut décomposer).

Trois niveaux de préoccupation (merise : méthode d'étude et de réalisation informatique pour les systèmes d'ensemble)

Trois niveaux de préoccupation et modèles

<i>Niveaux de préoccupation</i>	<i>Données</i>	<i>Traitement</i>
Niveau conceptuel	MCD	MCT
Niveau organisationnel et logique (modèle relationnel)	MOD MLD,	MOT MLT, ULT
Niveau physique (modèle opérationnel ou modèle opératoire)	MPD (BD)	MOpT

Les concepts propres à chaque modèle selon le niveau de préoccupation

Niveaux de préoccupation				
Niveau conceptuel	-Entité type ou type d'entité - type d'association	Association type ou type d'association	Propriété	Occurrence, instance, valeur
Niveau logique (modèle relationnel)	Relation	Relation (quand elle persiste et est maintenue)	Attribut ou (couple, triplet, quadruplet, ... n-uplet)	Tuple
Niveau physique (modèle opérationnel ou modèle opératoire)	Table	Table	Colonne ou champ	ligne ou enregistrement

L'approche systémique classique (Merise) et l'approche objet (PU/UML) sont différentes mais manipule les individus du même domaine de discours.

Les différents modèles sous UP sont dits digrammes.

Deux diagrammes sont importants et comparable MCD (SCD) de Merise et Diagramme de classes d'UP/UML.

Pour merise, tout traitement est séparé du regroupement de données en type d'entité. Quant à UP (avec ses déclinaisons) d'UML, des traitements dits opérations ou méthodes sont ajoutés au regroupement des données en classes.

L'ensemble des traitements en nombre peut s'avérer infini. Il possible de traitement ad'hoc non prévu au début d'un projet informatique de base de données applicative.

Si on peut cerner l'ensemble des données, ce n'est pas le cas de tous les traitements possibles.

Un regroupement de traitements sur les données constitue une application.

Les traitements vont des manipulations à base terminologique et ou calculatoire. ils se fondent sur les requêtes SQL.

-Dictionnaire de données

Via ce modèle, on recense les données brutes à épurer. C'est l'entame de la description du domaine de discours en rapport avec l'objectif défini sur la base des préoccupations et donc de la spécification des besoins.

Il comprend les propriétés qui seront utilisés par les outils de modélisations pour élaborer le modèle conceptuel des données (**MCD**) ou schéma conceptuel des données (SCD). Il constitue aussi la liste des entités types et leurs propriétés pour lesquelles l'on spécifie leur domaine de définition (numérique, alphabétique (textuel), date ...).

Par nom de la donnée ou simplement donnée, il faut comprendre propriété ou caractéristiques d'un type d'entité.

Quelques règles sur les données du dictionnaire des données sont :

- Données élémentaires (atomiques, non sécables) ;
- Données d'informations réduites ou calculées ainsi que les règles de calcul ;
- Données calculées de type situation ou historiques (total, taxes, commande/mois) ;

- Les paramètres utilisés dans les cas particuliers.

Exemple de dictionnaire de données

Nom de la donnée ou mnémonique	Signification	Type					Règle de calcul	Contrainte (d'intégrité)	Document
		format type (N,A,AND)	calculé	signalétique	situation	paramètre			
Nom-client		A		x					facture
Code-postal		N		x					facture
Ville-client		A		x					facture

T.V.A		N					18,6%		facture
Total HT		N	x			somme			
Nombre-Années-exercice-Personnel								< 37 ans	Attestation de travail

Pour le recueil des informations et de données, sont utilisés différents procédés de recherche avec une implication des acteurs à l'échelle. Comme procédés de recherche d'information pour une étude donnée, on a entre autres l'observation, la revue documentaire, l'Internet, les interviews des acteurs, sorties et visites de terrains.

Il y a lieu de comprendre les flux d'informations d'un acteur à un autre. Par acteur, il faut entendre un agent humain ou machine susceptible de mener au moins une des trois (3) tâches notamment recevoir, traiter et transmettre de l'information dans l'univers du discours circonscrite. On parlera de graphes de flux comme étant l'ensemble des envois et réception d'information dit message. Un flux met en relief un récepteur ou destinataire et un émetteur suite à traitement possible. Un flux porte sur un événement entrant et un événement sortant ou résultant.

S'agissant de la spécification des besoins, le demandeur ou le commanditaire peut être confronté à la formulation de ses propres besoins. Dans ce cas, des techniques d'élicitation peuvent être fournies au demandeur (avec ses futurs utilisateurs) par les maîtres d'œuvre ou de consultance. En général, le commanditaire est le maître d'ouvrage.

Les données à consigner dans le DD doivent respecter au minimum les règles de :

-non redondance, non polysémie (homonymies) et de non synonymie ;

Cela signifie qu'il faut éliminer les redondances, les synonymies et les polysémies.

Au terme de ce recueil, un ensemble de données et de règles de gestion sont assorties. Cet ensemble est soumis aux parties prenantes averties (utilisateur potentiel et porteurs de la conscience du projet en rapport avec les but et objectifs) pour être validé avec un minimum de **consensus**. Cette activité parachève l'admission de cette liste de données retenue en **dictionnaire de données (hors SGBD)**.

Inter-compréhension et élicitation

Un terme arrêté et utilisé pour désigner un ensemble d'individus (population d'individus ou d'instances) ou encore pour qualifier cet ensemble, doit être investi du sceau du consensus minimum. Bien entendu, ces individus...partagent les mêmes caractéristiques. Ils sont tout de même distincts par leur identifiant (eccéité ou heccéité).

[Onomasiologie/sémasiologie <=> intentionnalité/extensionnalité

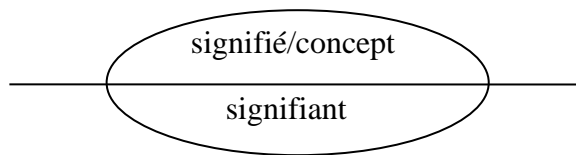
Par concept, il faut entendre une idée, un contenu de pensée, une représentation abstraite. C'est un signifié.

Deux grands courants de représentation du concept **concept** existent et émanent l'un de F Saussure (sémiologie) et l'autre de Peirce C Sanders (sémiotique).

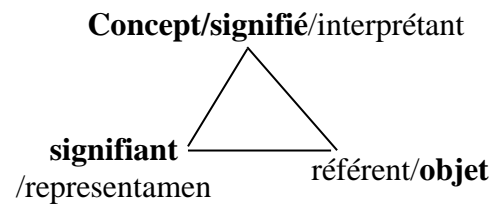
Sémiotique dite aussi sémiologie se fonde dans la linguistique (terme oral ou écrit) ou la croise.

Sémiotique signifie l'étude, la science ou la théorie des signes dans toutes leurs formes et dans toutes leurs manifestations. C'est l'étude pratique des systèmes signifiants.

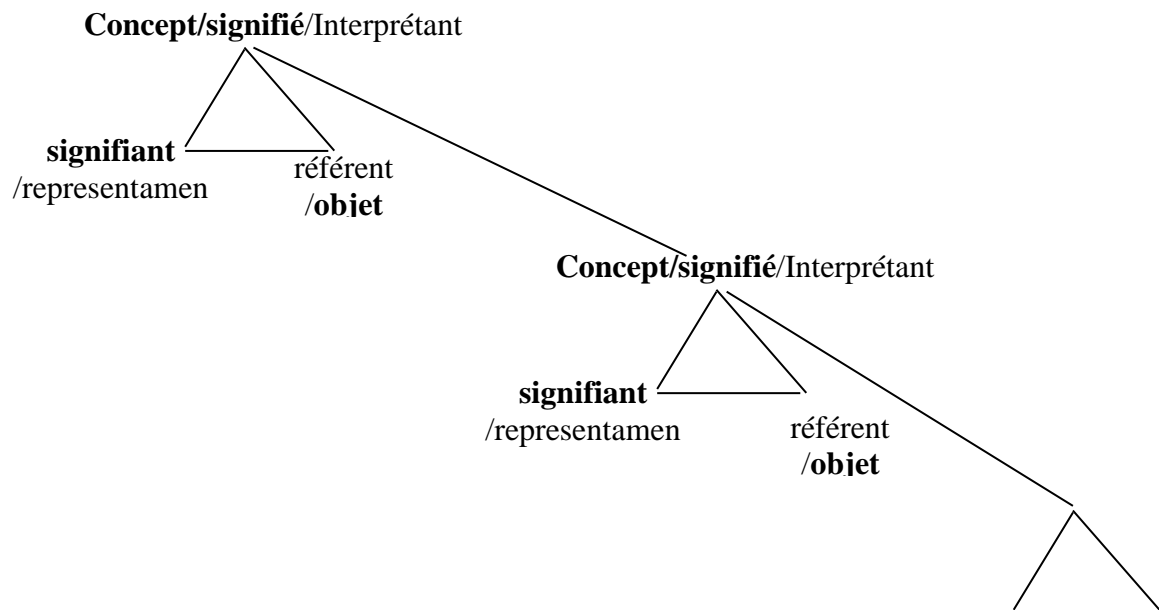
Approche d'obédience de F de Saussure



Approche d'obédience de Peirce C S



Sémiosis : Continuum d'interprétants successifs pour une personne humaine (pensée permanente de signe en signe, cognition permanente).



Le terme pour se saisir de ce concept est un mot ou un signifiant (terminologie en linguistique orale et écrite).

Aucune communication n'est possible sans un minimum d'iconicité, de signe.

Onomasiologie et sémasiologie

Obtenir un terme pour désigner des individus ou un concept relève de l'**onomasiologie**. C'est un acte d'intentionnalité. On contraint ainsi des individus dans l'emboîtement canonique et catégoriel.

Partir d'un terme ou disposer d'un terme pour chercher à lui donner une signification, un sens, c'est se forger à lui affecter des occurrences, des individus, des faits ou événements. Ce processus procède de la **sémasiologie**. C'est un acte d'extensionnalité.

Definiendum et definiens

Le definiendum est une agrégation d'autres definiendum (plus primitifs). Cette agrégation expressive est le definiens

Si un terme (**definiendum**) est défini, son expression (axiomatique) utilisant d'autres termes est appelée le **definiens**.

Exemple : Homme = Animal intelligent et raisonnable avec une certaine rationalité.

definiendum : Homme

definiens : animal intelligent et raisonnable avec une certaine rationalité.

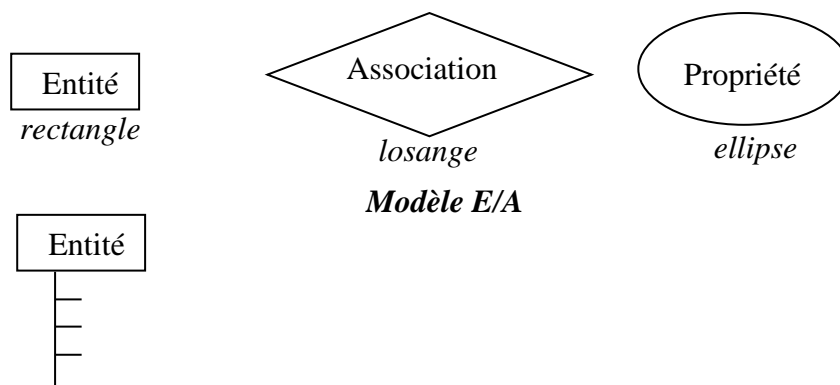
]

...

Graphe de dépendance des données/attributs/propriétés (GDF)

Modèle E/A

Le modèle E/A qui se résume par entité, association, propriété (attribut) permet de modéliser simplement des situations décrites en langages naturels. Un des mérites essentiels du modèle E/A est de permettre une représentation graphique élégante des schémas d'une BD. C'est le formalisme.



Modèle E/A

Modèle de Chen

-Domaine d'un attribut :

Le **domaine d'un attribut** caractérise à la fois le type de données (caractère, chaîne de caractère, entier ou réel, ...) et l'ensemble des valeurs admissibles de celui-ci.

Autrement dit, un domaine constitue un ensemble de valeurs admissibles sur lequel un ou plusieurs attributs du schéma prennent leurs valeurs effectives.

NB : Si deux attributs sont définis sur le même domaine, alors les comparaisons entre ces deux attributs (via l'usage de UNION ou JOINTURE) ont un sens.

Si ce n'est pas le cas, ces comparaisons sont absurdes sémantiquement.

-Identifiant

Un **identifiant** d'entité est une propriété particulière ou un ensemble de propriétés qui identifie de manière unique (ou univoque) une instance ou une occurrence d'entité.

Un attribut associe à chaque entité une **valeur** appartenant à un domaine.

Un domaine est un ensemble de valeurs acceptables par l'attribut considéré.

Un identifiant devient **clé** au niveau logique. A ce niveau, cette clé désigne plus globalement un groupe minimum d'attributs qui permet d'identifier de façon univoque un tuple dans une relation.

Lorsque le nombre d'attributs est supérieur à 1 (un) surtout dans une association transformée en relation, qui n'a donc pas disparu, alors une telle clé apparaît au niveau logique pour une association non hiérarchique. cette clé est aussi une **superclé**

(concaténation d'attributs). L'identifiant est un identificateur particulier dans la description des individus.

-Occurrence ou instance

un objet concret d'un type d'entité ou d'association, un exemplaire ou un individu du monde ou du domaine de discours.

-Cardinalité :

Elle est le nombre minimum et celui maximum de fois qu'une instance peut être intéressée (concernée) dans une association. Elle s'exprime en couple d'entiers naturels (C_{min} , C_{max}). $C_{min} \leq C_{max}$.

3-Concepts dérivés

-Formes d'association et cardinalités

Le type d'association caractérise le nombre de liens autorisés entre occurrences ou instances d'entités. Il sert à distinguer les associations en fonction du nombre d'occurrences et de classes d'entité qu'elles mettent en jeu. On distingue :

- Les associations **binaires**. Elle relie 2 entités.
- Les associations **n-aires** qui relient plus de 2 entités.
- Les associations **ternaires** qui relient de 3 entités.

Dans de telles associations, on a principalement les couples de cardinalités. On a principalement trois types de cardinalités que sont **(1;1)** ou (0 ;1), **(1;n)** ou (0;n), **(n;n)**.

NB : Il est recommandé de limiter le nombre de liens (ou de pattes) d'une association idéalement à deux ou trois.

Soient E et F deux entités liées par une association A. On dit aussi que E et F sont en association par A.

-Type **(1;1)** ou 1 à 1

Si à une occurrence d'entité E, peut correspondre via l'association A au plus une occurrence de l'entité F et que, réciproquement à une occurrence de E.

-Type **(1;n)** ou 1 à plusieurs

Si à une occurrence d'entité E peuvent correspondre via l'association A plusieurs occurrences de F mais à une occurrence de F au plus une occurrence de E. On parle d'association hiérarchique (dans le cadre d'association binaire).

Dans les autres cas, on parle d'association non hiérarchique, comme dans ce qui suit.

-Type **(n;n)** ou plusieurs à plusieurs

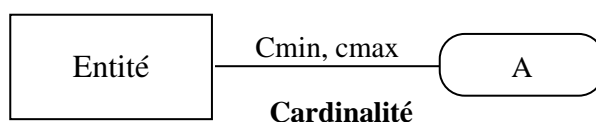
Si à une occurrence de l'entité E peuvent correspondre plusieurs occurrences de F et à une occurrence de l'entité F peuvent correspondre plusieurs occurrences de E via l'association.

Dans la pratique quatre types de cardinalité sont possibles sur les liens ou pattes des associations : **(0;1)**, **(1;1)**, **(0;n)**, **(1;n)**.

Pseudo formalisme :

La cardinalité se décompose en 2 objets:

- la cardinalité minimale (à 1) qui représente **l'obligation**, pour toute occurrence d'entité e_i de participer à l'association;
- la cardinalité maximale qui représente **le volume (nombre de fois)**, pour toute occurrence d'entité e_i de participer à l'association.



Le tableau ci-dessous représente les différentes valeurs possibles de Cmin et Cmax et leurs significations.

Valeurs de Cmin	Signification
0	Il peut exister dans l'entité E une occurrence e_i qui ne participe pas à l'association
1	Toute occurrence e_i de E participe obligatoirement à l'association A

Valeurs de Cmax	Signification
1	Si une occurrence e_i de E participe à l'association A elle ne peut y participer qu'une seule fois
N	Si une occurrence e_i de E participe à l'association A elle peut y participer plusieurs fois

Le tableau ci-après représente la signification des 4 couples de valeurs possibles en retenant comme critères:

- le caractère **optionnel ou obligatoire** de participation à l'association;
- **l'unicité ou la multiplicité** de participation à l'association.

		Cmin	
		Optionnel	Obligatoire
Cmax	Unicité	0,1	1,1
	Multiplicité	0,n	1,n

Souvent la cardinalité minimale est difficile à définir en début d'analyse, il est possible de mettre en lieu et place de 0 ou 1 le symbole -, en attendant de pouvoir se prononcer.

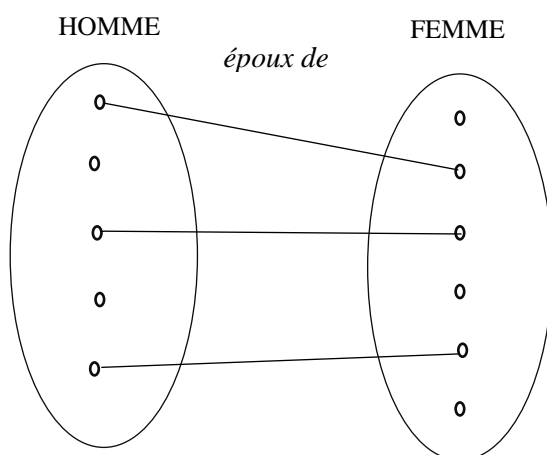
Remarque

Il est important de fixer les cardinalités avec une extrême rigueur car les valeurs à 1 de Cmin ou de Cmax peuvent engendrer un système bloquant. En effet ces cardinalités seront traduites dans la structure de la base de mémorisation des données en contraintes d'intégrité. Le tableau ci-après fixe les règles de contraintes qui seront appliquées :

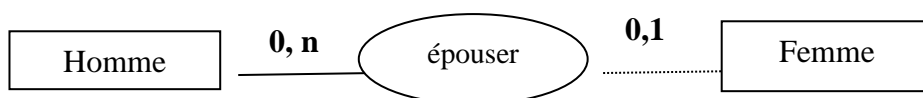
Cardinalité	Valeur	Caractéristique de l'association	Contrainte d'intégrité
Cmin	0	Optionnelle	-
Cmin	1	Obligatoire	Une occurrence ei de E ne peut être créée qu'avec une occurrence ai de l'association A
Cmax	1	Unicité	Une occurrence ei de E ne peut participer à la création d'une occurrence ai de l'association A que si elle ne participe pas déjà à une occurrence aj
Cmax	n	Multiplicité	-

Exemple des couples avec cardinalité

Cardinalité	Assertion pour l'exemple
0, 1	Un homme est l'époux d'une femme (Pas de bigamie!)

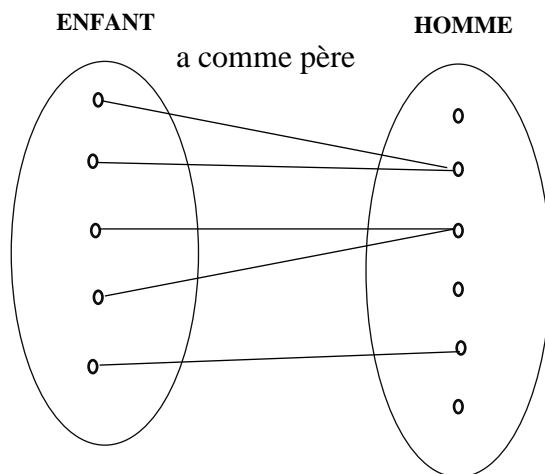


Cardinalité 0,1 / Représentation sagittale



Cardinalité 0,1 / Représentation E-A

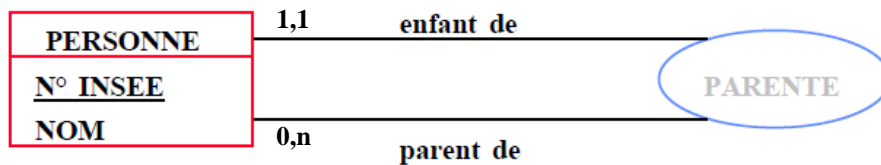
Cardinalité	Assertion pour l'exemple
1, 1	Un enfant a comme père



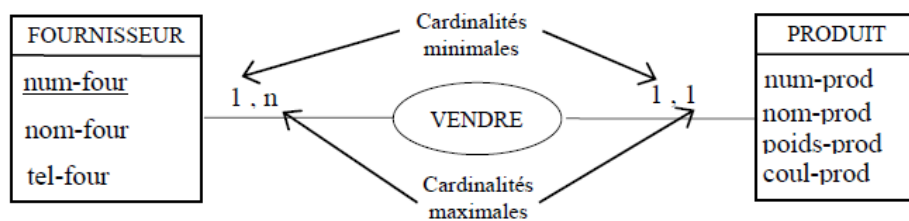
Cardinalité 1,1 / Représentation sagittale ou diagramme sagittal

Cardinalité	Assertion pour l'exemple
0, n	Un homme a comme enfant
1,1	Un homme a comme personne père

Association réflexive



Modèle ER de Codd



Modèle UP/UML



Dans le cadre de UP/UML, 1 peut se noter 0..1 ; 1..n peut se noter n.

Remarque : il y a inversement des cardinalités de mcd dans Merise pour le diagramme de classes de UP/UML.

4-Règles de bonne construction

Pour la cohérence du modèle E/A certaines règles doivent être adoptées :

- Élimination des redondances, des polysémies (homonymies), des synonymies ;
- Chaque entité possède un identifiant ;
- Chaque propriété d'une occurrence d'entité ne possède qu'au plus une valeur (monovaluée) ;
- Toutes les propriétés doivent être élémentaires (atomiques non décomposables) ;
- Toutes les propriétés autres que l'identifiant doivent dépendre pleinement de cet identifiant
- A chaque occurrence d'une association correspond une et une seule occurrence de chaque entité participant à l'association ;
- Pour une occurrence d'une association, il ne doit exister au plus qu'une valeur pour chaque propriété de cette association ;
- Chaque propriété d'une association doit dépendre pleinement de l'identifiant de cette association ;
- Une cardinalité (**0 ; 1**) ou (**1 ; 1**) indique une contrainte d'intégrité fonctionnelle (CIF) (dans le cadre ER).

NB : Des système multivalués existent et sont utilisés dans la limite de la maîtrise organisationnelle des données structurées et manipulées (MS Access 19). Par exemple, l'assignation de plus d'un numéro de téléphone à une personne. Le SQL évolue en conséquence.

NB : Une association est libellé en verbe d'action à l'infinitif.

Les concepts ou types d'entités sont libellés au singulier d'un substantif.

Base de données et dépendances fonctionnelles

Définition de schéma de base de données

- Un schéma de base de données est un ensemble de schémas de relations liés par des dépendances référentielles : attributs communs ou plus généralement des dépendances d'inclusion.
- Une base de données est alors un ensemble de relations (extensions) associé au schéma de base de données et vérifiant ses contraintes d'intégrité.

Un schéma de relation R est défini par un ensemble d'attributs U , le tout noté $R(U)$.

nomRelation (attribut: domaine, attribut:domaine, ...).

- Une relation r est une instance finie d'un schéma de relation, notée $r:R(U)$.

Définition des contraintes d'intégrité

Tout schéma de base de données doit être conçu pour imposer le respect d'un maximum de contraintes d'intégrité du réel perçu.

- Les **contraintes d'intégrité** sont des expressions logiques qui **doivent être satisfaites à tout instant** par une instance de base de données.

- Plusieurs types de contraintes d'intégrité
 - Contraintes sur les attributs : domaines, valeurs nulles, ... ;
 - Contraintes sur les n -uplets : la valeur d'un attribut peut dépendre d'un ou plusieurs autres attributs du même n -uplet (dates d'emprunt et de retour pour une bibliothèque), ... ;
 - Contraintes sur les relations : clés, cardinalité, ... ;
 - Contraintes sur la base de données : clés étrangères, ... ;
 - Contraintes temporelles : évolution chronologique (diplômes, état-civil), ...

Une C I peut découler d'un besoin technique ou des besoins émanant du cahier de charges (Spécification des besoins).

Elle peut relever des règles de gestion liées aux données (interviews et revue documentaire métier, domaine de discours).

Contraintes d'intégrité : notion de clé

Une relation est un ensemble de n -uplets. Par définition, un ensemble n'a pas d'élément en double, donc chaque n -uplet d'une relation est unique.

Pour identifier les n -uplets de façon unique sans en donner toutes les valeurs et respecter leur unicité une clé est nécessaire.

Définition de clé

- Groupe d'attributs minimum qui détermine un n -uplet de façon unique pour une relation.
- Plus formellement :

X clé de $R(U)$ avec $X \subseteq U$ ssi $\forall r : R(U), \forall t_1, t_2 \in r, t_1[X] = t_2[X] \rightarrow t_1 = t_2$

Propriétés

Toute relation possède au moins une clé : l'ensemble de ses attributs.

Si une relation possède plusieurs **clés candidates**, on choisit une clé qui sera privilégiée : **la clé primaire**. Aucun des attributs d'une clé primaire n'admet de valeur nulle (vide).

Conventions d'écriture

- .On souligne la clé primaire.
- .Les clés candidates sont soulignées en pointillés (souvent omis).

Définition de DF

- Principales contraintes d'intégrité formelles dans une BD.
- Une dépendance fonctionnelle (ou DF) indique une implication vérifiée universellement entre deux groupes d'attributs A et B : à une valeur pour A correspond toujours la même valeur de B.
- On la note $A \rightarrow B$ (signifie que A détermine B). A est parfois appelé « la source » et B « le but ou la cible ».

- Soit $R(A, B, C)$. L'attribut B est dit fonctionnellement dépendant de l'attribut A si étant donné 2 n-uplets $\langle a1, b1, c1 \rangle$ et $\langle a2, b2, c2 \rangle$ alors $a1 = a2 \rightarrow b1 = b2$
- Ou encore, A détermine B si étant donné une valeur de A, il lui correspond une valeur unique de B.

Remarques importantes

- Une DF est une assertion sur toutes les valeurs possibles (domaine des attributs) et non pas sur les valeurs actuelles (extension courante de la relation). Elle caractérise une intention et non pas une extension d'une relation : invariante au cours du temps.
- Les DF doivent être définies à partir d'un schéma de relation.

Par défaut, elles sont définies sur la relation universelle.

Utilités des DF

- Vérification que les extensions r d'un schéma R sont conformes au réel perçu.
- Spécifier (modéliser) les contraintes que devront vérifier toutes les relations d'un schéma relationnel.
- Outil de conception automatique d'un bon schéma relationnel. (WinDesign, PowerAmc)

Exemple

Soit une relation R exprimant l'emploi du temps d'un lycée construite sur les attributs suivants :

P (professeur), **H** (heure du cours), **S** (Salle), **C** (classe) et **M** (matière).

La signification d'un n-uplet de cette relation est :

Le professeur **P** enseigne la matière **M** à l'heure **H** dans la salle **S** à la classe **C**.

Donnez la liste des dépendances fonctionnelles.

Système de règle d'inférences définit par Armstrong en 1974 :

- Réflexivité : $Y \rightarrow X$ alors $(X \rightarrow Y)$ (ex. $A \rightarrow A$; $AB \rightarrow A$)
- Augmentation : $X \rightarrow Y$ alors $(\forall Z, XZ \rightarrow YZ \text{ ou } XZ \rightarrow Y)$
- Transitivité : $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

Autres règles déduites des 3 premières

Pseudo-transitivité : $X \rightarrow Y, WY \rightarrow Z$ alors $WX \rightarrow Z$

Démonstration

Augmentation à $X \rightarrow Y$ alors $WX \rightarrow WY$

Transitivité : $WX \rightarrow WY, WY \rightarrow Z$ alors $WX \rightarrow Z$

Union (ou composition) : $X \rightarrow Y, X \rightarrow Z$ alors $X \rightarrow YZ$

Démonstration

Augmentation à $X \rightarrow Y$ alors $X \rightarrow YX$

Augmentation à $X \rightarrow Z$ alors $XY \rightarrow ZY$

Transitivité : $X \rightarrow YX, XY \rightarrow ZY$ alors $X \rightarrow ZY$

Décomposition : $X \rightarrow Y$ et $Z \subseteq Y$ alors $X \rightarrow Z$

Démonstration

Réflexivité à $Z \subseteq Y$ alors $Y \rightarrow Z$

Transitivité : $X \rightarrow Y, Y \rightarrow Z$ alors $X \rightarrow Z$

1. Propriétés remarquables de dépendance fonctionnelle

Soient W, X, Y et Z des ensembles d'attributs non vides d'une relation R . Voici quelques propriétés remarquables :

Réflexivité : $(X \subseteq W) \Rightarrow (W \rightarrow^R X)$

Augmentation : $(W \rightarrow^R X) \Rightarrow (W, Y \rightarrow^R X, Y)$

Transitivité : $(W \rightarrow^R X \wedge X \rightarrow^R Y) \Rightarrow (W \rightarrow^R Y)$

Union : $(W \rightarrow^R X \wedge W \rightarrow^R Y) \Rightarrow (W \rightarrow^R X, Y)$

Pseudo-transitivité : $(W \rightarrow^R X \wedge X, Y \rightarrow^R Z) \Rightarrow (W, Y \rightarrow^R Z)$

Décomposition : $(W \rightarrow^R X \wedge Y \subseteq X) \Rightarrow (W \rightarrow^R Y)$

Propriétés des DF :

2. Définitions pour les dépendances fonctionnelles

Soient X, Y et Z des ensembles d'attributs non vides d'une relation R avec $X \rightarrow^R Y$.

Cette DF est alors dite :

Triviale : si et seulement si $Y \subseteq X$

Élémentaire ; si et seulement si $\forall Z$ tel que $Z \subset X$ on n'a pas $Z \rightarrow^R Y$

Canonique : si et seulement si Y n'a qu'un seul attribut

Directe : si et seulement si $X \rightarrow^R Y$ est élémentaire autrement dit

$\forall Z \wedge Z \neq X \wedge Z \neq Y$ on n'a pas $X \rightarrow^R Z \wedge Z \rightarrow^R Y$

3. Clé candidate

Rappel : Toute relation a au moins une clé candidate. Soit une relation R .

si R ne contient qu'un seul attribut,

cet attribut a une valeur distincte pour chaque occurrence et forme alors l'unique clé candidate de R .

si R contient plusieurs attributs

Soient X et Y deux ensembles non vides disjoints d'attributs de R tels que

$$X \cup Y = R.$$

Si on a $X \rightarrow^R Y$ avec X minimal, alors X est une clé candidate de R (il peut y en avoir plusieurs).

Si aucun X ne peut être trouvé ainsi, toutes les occurrences de R sont distinctes et l'unique clé candidate de R est formée de tous ses attributs.

Indication : Pour vérifier que X est minimal, il faut contrôler qu'il n'est pas possible de prendre un attribut dans X pour le mettre dans Y tout en maintenant $X \rightarrow^R Y$.

4. Pourquoi normaliser ?

La normalisation est utile :

- pour limiter les redondances de données,
- pour limiter les pertes de données,
- pour limiter les incohérences au sein des données et
- pour améliorer les performances des traitements.

-Formes normales

-**1FN** : données élémentaires + dépendance fonctionnelle **df**.

Ici, un attribut atomique n'est pas sécable. Il est non plus composé (structuré en sous-attributs), multivalué (liste de valeurs) ;

-**2FN** : 1FN + **dfe** (df élémentaire, aucune partie de la clé ne détermine un attribut non clé) :

-**3FN** : 2FN + **dfed** (élimination des cycles et transitivités) : Autre attribut non clé de la relation ne doit pas déterminer un attribut de la même relation.

NB : La répétition des propriétés et la duplication doivent être évitées dans l'élaboration du MCD ou SCD. En cas de non-respect de ces DF, les inconvénients sont l'incohérence de la BD, la modification difficile, le risque élevé de faire des erreurs de saisie surtout si c'est la même occurrence d'une propriété qui est répétée.

En cas de df entre attributs non clés (identifiants) dans une entité, il faut extraire la ou les propriétés de cette entité. Ces propriétés extraites feront l'objet d'une ou de nouvelles nouvelle (s) entité (s). Puis l'associer à l'entité d'où elle est extraite, et ce de manière hiérarchique.

Forme normale de Boyce Codd

La deuxième forme normale élimine les anomalies créées par des dépendances entre parties de clé et attribut non clé.

La troisième forme normale élimine les anomalies créées par des dépendances entre les attributs non clé (élimination de cycles et de transitivité).

Question : Quid des dépendances de parties de clés entre elles ou d'attributs non clé vers une partie de clé ?

La troisième FN (3FN) est bien insuffisante.

Afin d'éliminer les redondances créées par des dépendances entre parties de clé et celles éliminées par la 3FN, **Boyce** et **Codd** ont introduit la forme normale qui porte leur nom BCNF.

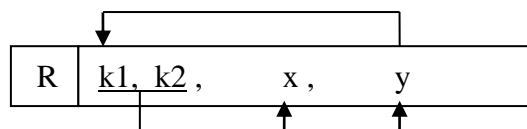
Une relation est en **BCNF** si et seulement si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé entière détermine un attribut.

BCNF est une relation respectant la 3FN et dont tous les attributs non-clé (hormis les clés candidates qui sont neutres et ne doivent pas être considérées) ne sont pas source de dépendance fonctionnelle.

Cette définition a le mérite d'être simple : pas de dépendance autre que $K \rightarrow A$, K étant la clé et A un attribut non clé.

Il a été montré qu'une relation a une décomposition en BCNF qui est sans perte. Par contre une décomposition en BCNF ne préserve en général pas les df.

Soit $R(\underline{k1}, \underline{k2}, x, y)$ une relation et $k1, k2, x$ et y ses attributs.



Le schéma ci-dessus est en 3FN mais pas en BCNF. Car Y détermine $k1$ une partie de la clé ($k1, k2$).

Une telle relation $R(\underline{k1}, \underline{k2}, x, y)$ doit être décomposable en deux autres que sont : $R1(\underline{k1}, \underline{k2}, x)$ et $R2(y, k1)$.

Exemple

Localisation (cru, pays, région, qualité)

Or $\text{région} \rightarrow \text{pays}$ (pays détermine région). Donc cette relation n'est pas en BCNF bien qu'en 3^{ème} FN puisque cru ne détermine pas la région (il y a du chablis en bourgogne mais aussi en californie).

Cru	Pays	Région	Qualité
Chenas	France	Beaujolais	Excellent
Juliéna	France	Beaujolais	Excellent
Chablis	Etats unis	Californie	moyen
Morgon	France	Beaujolais	Bon
Brouilly	Etats unis	Californie	Médiocre
Chablis	France	Beaujolais	Excellent

Pour la rendre BCNF, il faut la décomposer comme suit.

CRUS (cru, pays, qualité)

Region (region, pays).

La df cru, pays \rightarrow region est perdue. Mais la décomposition est sans perte.

Superclé

Dans une entité (ou entité type), une clé est un ensemble minimal d'attributs qui détermine toutes les autres.

Un ensemble d'attributs qui dénote une clé est appelé **superclé**. (Identifiant minimal).

Pour une même relation, il peut avoir plusieurs clés : En général, une désignée comme clé primaire. On parle également **clé candidate** pour désigner une clé quelconque.

Une clé est un ensemble minimal d'attributs qui détermine toutes les autres.

Un ensemble d'attributs qui inclut une clé est appelé superclé (Identifiant minimal).

Une clé choisie parmi les clés candidates pour indexer une base est appelée une **clé naturelle**.

Quatrième et cinquième formes normales

La BCNF est bien insuffisante pour éliminer complètement les redondances.

- DM : dépendances multivaluées : elle est une généralisation de la DF.

Quatrième forme normale (4FN)

La 4FN est une généralisation de BCNF destinée à décomposer les relations ayant des DM élémentaires.

Cinquième forme normale ou forme normale de projection jointure (5FN)

La 5FN est une forme normale de projection de jointure. P702

Dénormalisation et groupement de tables

Dénormalisation :

Technique consistant à implémenter la jointure de deux relations ou plus à la place des relations individuelles initiales.

Bien évidemment cela pourrait conduire à des redondances.

Groupement :

Une variante plus performante de la dénormalisation est le groupement, permis dans certains SGBD. Ce groupement connu sous le nom de placement à proximité est une technique permettant de regrouper dans une même page ou des pages voisines les tuples de deux tables (ou plus) selon un critère de jointure.

Partitionnement vertical et horizontal

Partitionnement vertical :

Il consiste à extraire d'une table ou relation via une projection selon les attributs ou champs ou colonnes. Dans cette projection, il est possible que les clés soient répétées

Partitionnement horizontal :

Il consiste au contraire à diviser la table en sous tables de même schéma, chacune conservant une partie des tuples (selon des critères de restriction)

Plus grand ensemble stationnaire F^+ de DF valides obtenu à partir d'un ensemble F initial en appliquant les propriétés des DF (axiomes d'Armstrong).

La fermeture transitive d'un ensemble de DF est unique. Elle ne dépend pas de l'ordre d'utilisation des propriétés.

Deux ensembles de DF sont équivalents ssi ils ont la même fermeture transitive $F1 \equiv F2$ alors $F1^+ = F2^+$

Contrairement à F^+ , la couverture minimale n'est pas unique.

Couverture minimale, ensemble essentiel pour effectuer des décompositions sans perte.

Algorithme en 3 étapes (ordre à respecter) à partir d'un ensemble de DF :

1. Décomposer les DF (un seul attribut en partie droite) ;
2. Rendre les DF élémentaires (membre gauche non décomposable) ;
3. Supprimer les DF redondantes (obtenues par transitivité) ;

Unicité de la couverture minimale (CM)

L'unicité de la CM va avec la compréhension de la même manière de la sémantique des données et leurs relations. Elle découle de l'interprétation commune des règles de gestion afférentes aux données et leurs corrélations.

V- FONCTIONS D'UN SGBD

Un SGBD permet de décrire les données des bases, de les interroger, de les mettre à jour, de représenter les transactions de données, d'assurer les contrôles d'intégrité, de concurrence d'accès et de sécurité. Il intègre de plus en plus des fonctions avancées pour la gestion des procédures et des événements.

1) Description des données

La description concerne la définition des structures de données et le suivi de leur évolution et du stockage (volumétrie de la BD).

Un SGBD offre des interfaces pour décrire les données. La définition de différents schémas est effectuée par les administrateurs de BD (ou par des personnes jouant le rôle d'administrateurs de données). Dans un SGBD où l'environnement de description de BD supporte trois niveaux de schémas, on peut définir trois rôles d'administrateur de données ou simplement trois administrateurs. Cela est conforme à l'approche retenue par le comité ANSI-SPARC qui ajoute à ces rôles un quatrième d'où :

- L'administrateur d'entreprise ;
- L'administrateur de BD ;
- L'administrateur d'application ;
- Le programme l'application .

-Administrateur d'entreprise

Le porteur de ce rôle est chargé de l'élaboration du schéma conceptuel (description de toutes les entités et leurs propriétés avec les associations). C'est la vue du concepteur indépendamment de toute utilisation et de toute implantation machine (disque dur).

-Administrateur de BD (DBA)

L'exécutant de ce rôle est chargé de définir le schéma interne et les règles de correspondance entre le schéma interne et le schéma conceptuel

NB : Le schéma interne est la représentation du schéma conceptuel dans la machine. Ce schéma interne constitue la vue du système où le niveau global et le niveau conceptuel sont confondus.

Tâches d'un DBA :

- Administrer les systèmes de gestion de données de l'entreprise, en assurer la cohérence, la qualité et la sécurité
- Participer à la définition et à la mise en œuvre des bases de données et des progiciels retenus par l'entreprise
- Participer au choix des bases de données et à leur dimensionnement
- Créer des bases en liaison avec l'équipe d'administration et les Chefs de Projets

- Mettre en œuvre des logiciels de gestion de bases de données ainsi que des progiciels de gestion, adaptation et maintenance de ces logiciels
- Mettre en exploitation et en gestion des serveurs de données en relation avec l'administrateur système
- Créer des outils spécifiques d'aide à l'exploitation
- Assister les utilisateurs (formation, requête techniques ...)
- Être le support technique de second niveau pour l'ensemble des bases de données
- Être le correspondant technique entre les Chefs de projets et l'équipe administration
- Gérer les performances et l'optimisation des ressources
- S'assurer de l'intégralité des bases de données existantes en garantissant la sécurité physique et logique
- Mettre en œuvre des outils de surveillance
- Utiliser de façon optimale des bases en réglant leurs paramètres
- Veille technologique sur les SGBDR et les progiciels retenus par l'entreprise
- Tester et valider, pour les aspects techniques tous les logiciels et progiciels
- Définir des normes et standards d'utilisation et d'exploitation vers les SGBD
- Organiser, planifier et suivre la réalisation de projet lié aux bases de données
- Réaliser des documents (modèle de donnée, spécifications fonctionnelles et techniques...)

-Administrateur d'application

L'attributaire de ce rôle est chargé de définir les schémas externes correspondant à la vue de l'utilisateur ou groupe d'utilisateurs.

Un autre rôle dont il faut tenir compte au niveau applicatif, c'est le **programmeur d'application**. Le détenteur d'un tel rôle se consacre aux descriptions spécifiées par l'administrateur d'application.

NB : Un autre rôle existe celui de l'**administrateur système ou réseau**, surtout lorsque la base de données utilise le réseau ou encore dans le cadre d'une base de données répartie.

Remarque

Au regard des trois premiers rôles, il s'ensuit les trois **niveaux fonctionnels** retenus par le comité ANSI-SPARC (American National Standard Institute, Scalable Processor ARChitecture) : le niveau conceptuel, le niveau interne et le niveau externe déjà abordés (précédemment vus).

2) Recherche de données

Tout SGBD fournit des commandes de recherche des données. Les SGBD modernes offrent un langage d'interrogation assertionnel permettant de retrouver les données par le contenu, sans préciser la procédure d'accès. Une commande de recherche permet de retrouver les données de la base répondant à un critère de qualification (prédicat) plus ou moins complexe. Une recherche s'exprime à l'aide de requêtes du type suivant :

QUEL: RETRIEVE {liste d'attributs} FROM table WHERE *qualification*

SQL: SELECT {liste d'attributs} FROM table WHERE *qualification*

A priori, le SGBD doit offrir un langage complet c'est à dire permettant de poser toutes les questions possibles sur la BD. Les questions peuvent faire intervenir un grand nombre d'entités/associations, des calculs, des qualifications.

3) Mise à jour (update, alter)

La conception de mise à jour intègre à la fois l'insertion de données dans la base, la modification dans données et leur suppression. L'insertion de plusieurs instances d'entités peut s'effectuer facilement. Par contre, celle des instances d'associations est plus délicate car

il faut insérer un enregistrement référençant les entités de la base participant à ces associations à modifier. Il peut arriver que la modification des données s'effectue par recherche de données à modifier à l'aide d'une qualification. Aussi, il est possible de supprimer des tuples.

Il ne faut pas occulter non plus la mise à jour structurelle (*alter*) sans grande incidence d'organisation sinon refaire carrément un nouveau projet d'informatisation.

alter pour les contenant et *update* pour les contenus
create pour les contenant et *insert* pour les contenus ..

4) Transformation des données

Cette fonction incombe au SGBD. Comme il peut exister plusieurs niveaux de schémas pour décrire un ensemble de données, le SGBD doit pouvoir assurer le passage de données depuis un format correspondant à un niveau dans le format correspondant à un autre niveau (conceptuel à interne). Cette fonction est appelée **transformation des données**.

Il existe trois niveaux de schéma donc deux niveaux de transformations :

- Conceptuel à interne qui permet de faire passer des instances de données depuis le format conceptuel au format externe et réciproquement.
- Conceptuel externe qui permet de faire passer des instances de données depuis le format conceptuel au format externe et réciproquement.

5) Autres fonctions

1-Déclencheur (triggers) (fonctions et procédures stockées)

De nombreuses autres fonctions sont progressivement intégrées au SGBD au fil des ans. Par exemple, beaucoup savent aujourd'hui déclencher des procédures cataloguées (ou stockées) par l'utilisateur lors de l'apparition de certaines conditions sur les données ou lors de l'exécution de certaines opérations sur les entités et associations. Cette fonctionnalité est connue sous le nom de **trigger** (ou **déclencheur**), encore appelée **reflexe** dans le contexte des architectures client/serveur en relationnel.

Les déclencheurs (trigger) permettent de rendre les BD actives. Par exemple, on peut déclencher des procédures de correction lors de l'apparition de certains événements. Il s'agit là d'une fonctionnalité nouvelle qui prend de plus en plus de l'importance (ampleur).

De manière générale, les SGBD sont amenés à supporter des règles permettant d'inférer de nouvelles données (calcul par raisonnement logique) à partir des données de la base, lors des mises à jour ou des interrogations. Cela conduit à la notion de SGBD déductif capable de déduire des informations à partir de celles connues et de règles de déduction.

Enfin, les SGBD sont aussi amenés à gérer des objets de plus en plus complexes tels que les dessins ou images, des architectures ou cartes de géographie en capturant finement des découpages de ces objets en sous-objets. Ces objets pourront être atteints via des procédures elles-mêmes intégrées au SGBD. Cela conduit à la notion de SGBD à objet ou SGBD objet, capable de gérer des objets multiples manipulés par des fonctions d'utilisateurs. (Framework, module ou outils métiers, procédures stockées, oracle spatial). ERP, CRM.

Malgré la complexité de certaines contraintes, l'utilisation des déclencheurs permet de rejeter les mises à jour qui les violent. Ces déclencheurs permettent d'entraîner une opération conséquente suite à une première opération de base. Mieux, ils permettent de modéliser au sein de la BD les comportements réactifs des applications.

Les SGBD traditionnels sont passifs en ce sens qu'ils exécutent les commandes de mise à jour et de recherche.

Avec les déclencheurs, ils deviennent actifs et sont capables de réagir à des événements externes.

Par exemple, la surveillance d'un commerce électronique peut nécessiter le refus de vente à un client suspect. Une demande de réapprovisionnement en cas de rupture de stock. (Site web dynamique : sous tendu par une base de données ou des formulaires augmentant l'interactivité homme machine).

Tous ces événements peuvent être réalisés par des SGBD avec des déclencheurs appropriés. Ceci introduit la notion de BD active.

Une BD active permet donc de déplacer le comportement réactif des applications dans le SGBD.

5-2- Contrôle de l'intégrité des données

On appelle **contrainte d'intégrité**, toute règle implicite ou explicite que doivent suivre les données. Ces règles visent à assurer le maintien de la cohérence des données. Pour un SGBD supportant le modèle entité association, les contraintes possibles sont l'unicité de clé, la contrainte référentielle (clé secondaire) et la contrainte de données (domaines de variation ou type) et les cardinalités, les propriétés obligatoires.

Il faut veiller à ce que les valeurs des identifiants primaires soient stables durant le cycle de vie de l'entité pour en assurer la cohérence.

5-3- Gestion des transactions et sécurité

Elle permet d'assurer qu'un groupe de mise à jour soit totalement exécuté ou pas du tout. Cette propriété est connue sous le nom d'**atomicité** des transactions. Elle est garantie par le SGBD qui connaît l'existence de transaction à l'aide de commandes : Begin-transaction, End-transaction.

Une transaction est donc un groupe de mises à jour qui fait passer la base d'un état à un autre. Ces états successifs doivent être cohérents et donc respecter les contraintes d'intégrité. Cette propriété est connue sous le nom de correction des transactions (correctness, correction, **cohérence**).

Lorsqu'une transaction est partiellement exécutée, les données peuvent passer par des états incohérents transitoires qui seront corrigés par les mises à jour suivantes de la transaction.

(Transaction partiellement exécutée est justifiable par : une requête de mise à jour pas très bien formulée, mise à jour portant sur un trop grand nombre d'enregistrements, niveau de répartition des données élevé).

Pendant une période de l'activité, les effets de la transaction ne devront pas être visibles aux autres transactions. Cette propriété est connue sous le nom d'**isolation** de la transaction.

L'isolation doit donc être assurée par le SGBD.

Une fois la cohérence établie, elle doit l'être de manière durable. (Conserver la cohérence). Cette propriété est connue sous le nom de la **durabilité** de la transaction. Dès qu'une transaction valide ces modifications, le système doit garantir qu'elles seront conservées en cas de pannes (panne disque par exemple). (Persistance du système)

En résumé, toutes ces propriétés résumées sous le sigle ACID, doivent être assurées par un bon SGBD qu'il soit centralisé ou réparti.

Ces propriétés nécessitent deux types de contrôles qui sont intégrés dans les SGBD :

- Contrôle de la concurrence des accès

- Résistance aux pannes avec validation et reprise (réflexivité multi niveau)

En plus, il faut rappeler que la sécurité permet d'éviter les accès non autorisés aux données par des mécanismes de contrôle des droits d'accès (privilèges, rôle), et également de restaurer les données en cas de pannes ou d'erreurs.

Il est important de noter qu'en matière de sécurité on peut mettre à contribution plusieurs mécanismes et techniques : outils de supervision du réseau (dude), système RAID5 (redundant array of inexpensive disk), trois sauvegardes de la BD au minimum et souvent hors site, conseil et formation permanente des utilisateurs, veille technologique.

VI- ARCHITECTURE FONCTIONNELLE D'UN SGBD

1) Architecture à 3 trois niveaux de l'ANSI/X3/ SPARC-DBTG (Data Base Task Group) [voir polycop] X3 est une division de ANSI.

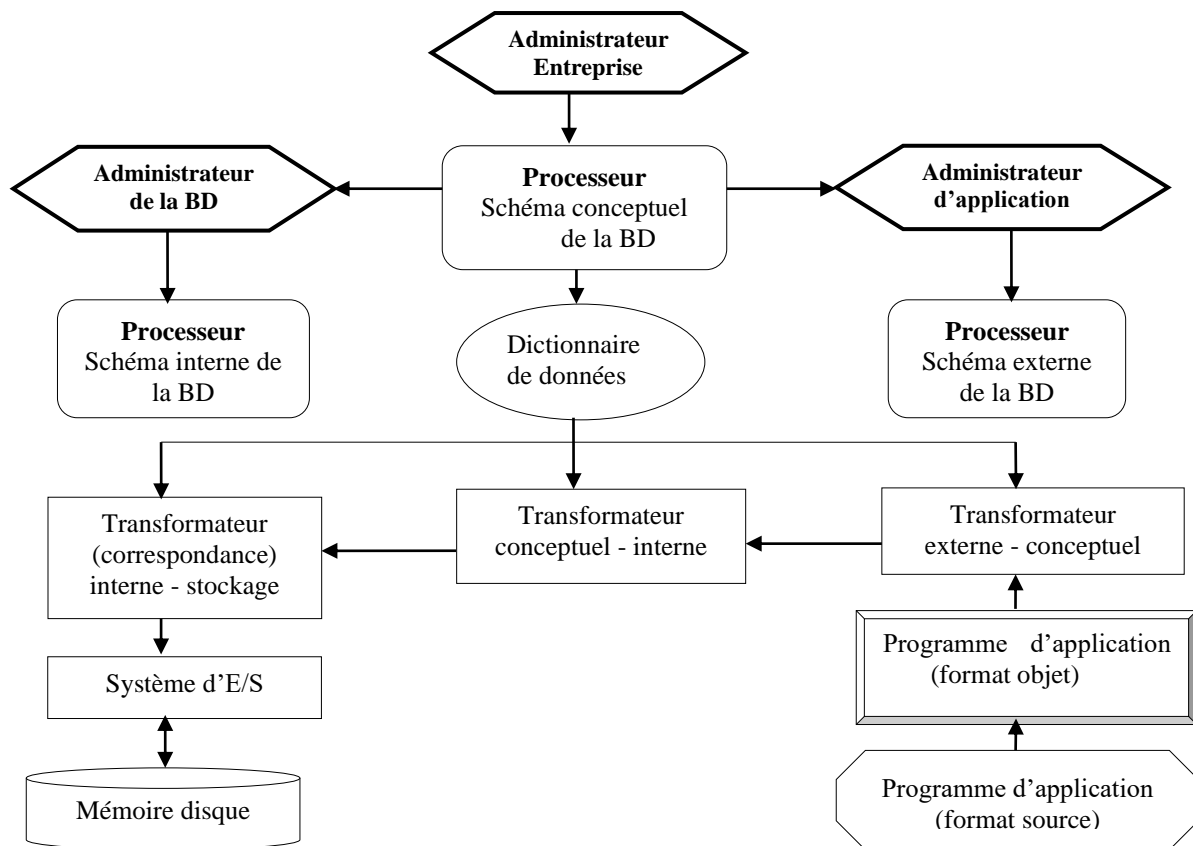
Le groupe (ou comité) ANSI/X3/SPARC-DBTG a proposé vers 1980, une architecture intégrant les 3 niveaux de schéma : externe, conceptuel et interne. Bien qu'ancienne, cette architecture permet de bien comprendre les niveaux de description et de transformation des données possibles dans un SGBD. En gros, cette architecture s'articule autour du dictionnaire de données et comporte 2 parties :

- un ensemble de modules (processeurs) permettant d'assurer la description des données et donc la constitution du dictionnaire de données.
- Une partie permettant d'assurer la manipulation des données c'est à dire l'interrogation et la mise à jour des bases.

Dans chacune des parties, on trouve les 3 niveaux : interne, conceptuel, externe (voir schéma architecture de référence). A chacun des trois niveaux correspond un processeur.

Les fonctions des trois processeurs sont les suivantes :

- Le processeur de schéma conceptuel compile le schéma conceptuel et, dans le cas où il n'y a pas d'erreur, range ce schéma compilé dans le dictionnaire de données (dictionnaire de données système).
- Le processeur de schéma externe compile les schémas externes et les règles de correspondance externe à conceptuel et, dans le cas où il n'y a pas d'erreur, range le schéma compilé et les règles de correspondance dans un dictionnaire de données (SQL).
- Le processeur de schéma interne a un rôle symétrique (semblable et opposé) au processeur de schéma externe.
- Le processeur de transformation externe à conceptuel transforme les manipulations externes en manipulations conceptuelles et inversement.
- Le processeur de transformation interne à conceptuel traduit les manipulations internes en manipulations conceptuelles et inversement.
- Finalement, le processeur de transformation interne à stockage traduit les manipulations internes en programmes d'accès au système de stockage et délivre les données stockées en format correspondant au schéma interne.



ARCHITECTURE DE REFERENCE D'UN SGBD A TROIS NIVEAUX (ANSI/X3/ SPARC-DBTG))

ANSI: American National Standard Institute

SPARC : *Scalable Processor ARChitecture* (« architecture de processeur adaptable »).

DBTG : Database Task Group

A la place de transformateur on peut mettre correspondance (alors mettre une flèche à sens double).

2-Architecture fonctionnelle typique

L'architecture à trois niveaux présentés ci-dessus permet de bien comprendre les niveaux de description et de manipulation des données. Cependant, la plupart des SGBD intègre le niveau interne et le niveau conceptuel dans un seul niveau. En fait, le véritable niveau conceptuel est pris en compte par les outils d'aide à la conception de l'application. Le schéma de l'architecture typique d'un SGBD est celui de référence le plus proche de des architectures des SGBD actuels basés seulement sur 2 niveaux de schéma : schéma (conceptuel et interne) et vues.

Nous allons passer en revue trois rôles du SGBD : analyseur, contrôleur, optimiseur, exécuter.

-Analyseur

Du point de vue de la description des données, un SGBD gère un dictionnaire de données appelé métabase (métadonnée), organisé comme une BD. Il est alimenté par les commandes de définition de schémas (exemple : create table..., create index...) et les définitions de vues (exemple : define view...).

Ces commandes sont analysées et traitées par le processeur d'analyse, **analyseur**, plus spécifiquement par la partie traitant le langage de description des données de ce processeur (qui fait souvent appel à des fonctions plus internes au SGBD pour gérer le dictionnaire comme une véritable BD).

Du point de vue de la manipulation des données, des requêtes (exemple : select, retrieve, modify, update, delete...) sont d'abord pris en compte par l'**analyseur de requêtes**. Celui-ci réalise l'analyse syntaxique (conformité à la grammaire) et sémantique (conformité à la vue référencée ou au schéma) de la requête. Celle-ci est alors traduite en format interne.

-Contrôleur

Une requête en format interne référençant une vue doit tout d'abord être traduite en une ou plusieurs requêtes référençant les données existant dans la base (schéma). Cette fonctionnalité est accomplie au niveau du **contrôleur** de requête ou modificateur de requête. Elle consiste à changer la requête en remplaçant les références aux objets de la vue par leur définition en termes d'objets du schéma. C'est aussi au niveau du contrôleur que sont pris en compte les problèmes de contrôle des droits d'accès (autorisation de lire ou d'écrire un objet) et du contrôle d'intégrité lors des mises à jour. Le contrôle d'intégrité consiste à vérifier que la base n'est pas corrompue (polluée) lors des mises à jour c'est à dire que les règles de cohérence des données restent vérifiées après les mises à jour.

-Optimiseur

L'**optimiseur** de requêtes est un composant clé des SGBD. Essentiellement, il élabore un plan d'accès optimisé pour traiter les requêtes. Pour ce faire, il décompose la requête en opérations d'accès élémentaires (sélection **d'index**, lecture d'article) et choisit un ordre d'exécution optimal ou plus proche de l'optimum pour ses opérations. Il choisit aussi les méthodes d'accès à utiliser. Pour effectuer les meilleurs choix, l'optimiseur s'appuie souvent sur un modèle de coût qui permet d'évaluer le coût d'un plan d'accès avant son exécution. Le résultat de l'optimisation peut être sauvegardé par des exécutions multiples ultérieures ou exécutées directement puis déduite. Pour rappel on a la détermination du coût d'une requête Q par :

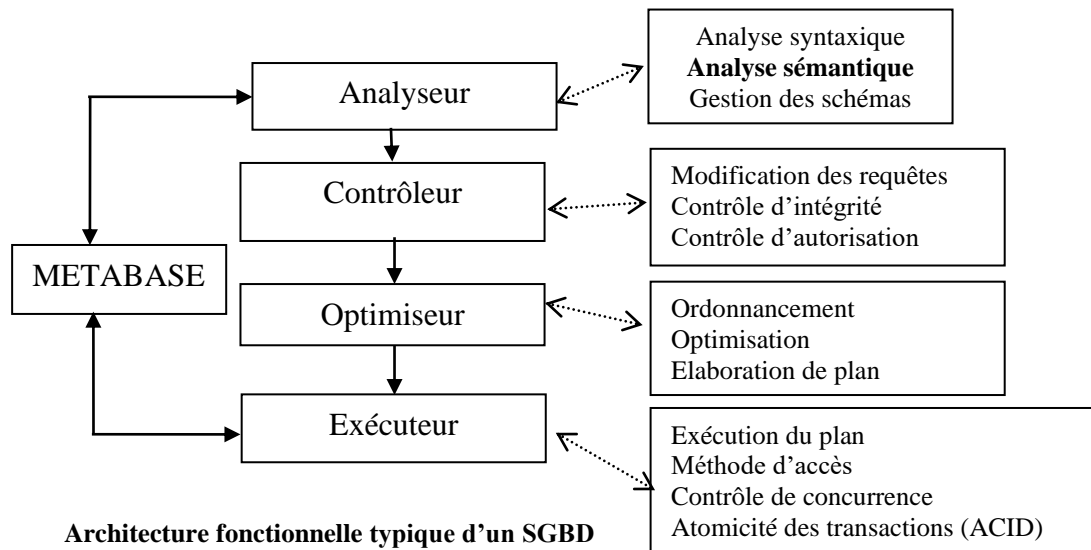
$C(Q) = a \times UC(Q) + b \times E/S(Q)$ avec Q étant un ensemble typique de requêtes, E/S (Q) est le nombre d'entrée/sortie, UC (Q) est le temps UC dépensé. a et b sont les facteurs convertissant respectivement temps UC(Q) et nombre d'E/S (Q) en coût.

-Exécuteur

L'**exécuteur** de plan a enfin pour rôle d'exécuter le plan d'accès choisi et élaboré par l'optimiseur. Il s'appuie sur les méthodes d'accès qui permettent d'accéder aux fichiers via des index et/ou des liens. C'est à ce niveau que sont gérés les problèmes de concurrence d'accès et d'atomicité de transactions. Les techniques utilisées dépendent de l'architecture opérationnelle du SGBD qui s'exprime en termes de processus et de tâches

Le schéma suivant donne une illustration succincte de ce qui vient d'être décrit.

Schéma de synthèse



3) SGBD opérationnel

Un SGBD au sens opératoire, est un véritable système qui doit fournir les services de base suivants :

- La gestion des données physiques en mémoire secondaire (la persistance) ; les fichiers de données doivent rester tels.
- La gestion de la sécurité des données avec contrôle de synchronisation pour les mises à jour de plusieurs processus concurrents ;
- La gestion des reprises sur pannes (système à tolérance de fautes ou d'erreurs ou de pannes).

4) Architecture répartie

a-Nouvelle tendance des SGBD

La nouvelle tendance au niveau des SGBD, c'est qu'ils intègrent les outils permettant d'atteindre l'ensemble des objectifs et deviennent de plus en plus répartis pour intégrer les fonctionnalités permettant à des applications internet de les utiliser aisément.

Il convient de noter qu'un SGBD offre une interface de description des données qui permet de documenter le dictionnaire de données. Le compilateur du langage de description des données gère cette méta base (dictionnaire de données).

Aussi, un SGBD offre une interface de manipulation des données (recherche, mise à jour,...) qui permet de modifier ou de retrouver des données dans la base. Le compilateur optimiseur de langage de manipulation gère des plans d'accès optimisés. Ceux-ci sont exécutés par le processus serveur qui gère aussi la concurrence d'accès et assure la fiabilité des données. Les requêtes peuvent être émises par des programmes d'applications écrits dans des langages traditionnels ou par des utilisateurs travaillant en interactivité.

Une BD à architecture répartie est composée de plusieurs serveurs coopérant à la gestion de BD composée de plusieurs bases gérées par un seul serveur. Mais apparaissant comme des bases (en environnement Unix) centralisées pour l'utilisateur.

Remarque :

Une BD à architecture répartie, peut être composée de serveurs munis de SGBD différents et spécialisés. C'est un exemple de BD réparties hétérogènes encore appelés BD fédérées.

Exemple : Oracle et Ingres sont des SGBD répartis.

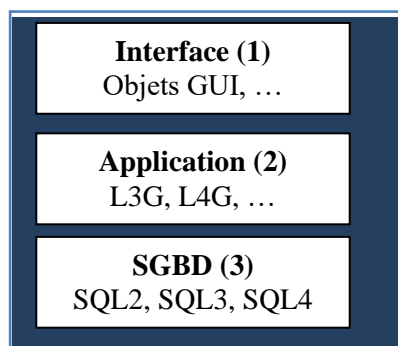
-Infostructures centrées base de données

Une Infostructure est un mot valise contraction de mots information et infrastructure pour montrer la prééminence de la ressource information et des bases de données dans les architectures matérielles et logicielles, ou encore la primauté des systèmes d'information sur le système informatique.

Ces architectures ont évolué au fil du temps depuis les architectures mainframes propriétaires, vers des architectures internet sans fil en conservant le serveur de base de données. Celles du futur mettront en jeu les architectures pair à pair (peer to peer computing). Le tableau suivant décrit cet état de fait.

Epoque 1 : Mainframe	Epoque 2 : Interface graphique	Epoque 3 : Client serveur (voir n-tiers)
Fiabilité, sécurité, vitesse d'accès disque, optimisation des ressources, performance, coûts	En plus de ce qui précède on a l'apport de l'autonomie et interface graphique	En plus de ce qui précède arrivée de nouveaux standards, indépendance des traitements, middlewares

b-Architecture fonctionnelle du développement d'une application



Dans une architecture centralisée (1), (2) et (3) sont sur le même serveur.

Dans une architecture client serveur (1) et (2) sont sur le poste client et (3) sur le serveur.

Dans une architecture Internet généralisée (n-tiers), on a une architecture comme suit :

- le (1) éclaté sur un ou des clients légers (téléphone mobile, PDA) et aussi un serveur web/portail ;
- le (2) est éclaté sur plusieurs serveurs d'applications ;
- le (3) est éclaté sur plusieurs serveurs de données (BD).

Exemple :

Serveur de base de données : Mysql ;

Serveur web : Appache ;

Serveur d'application : Php ;

Client : internet explorer (browser).

Remarque :

Un SGBD réparti peut mettre en jeu deux cas de figures :

- des SGBD de même type. On parle alors d'architecture en étoile (SQL server, Ingres Star, MS Access, System R ...)
- des SGBD hétérogènes suivant le même modèle (oracle, informix, Progress...) ou des modèles distincts (modèle hiérarchique avec IMS (Integrated Management System), modèle navigationnel avec IDS II (Integrated Data Store, modèle relationnel avec DB2).

Dans les deux cas (même modèle ou non), l'hétérogénéité peut être étendue au matériel et aux réseaux.

La gestion des données réparties requiert un coordinateur, qui peut être centralisé ou réparti, pour assurer :

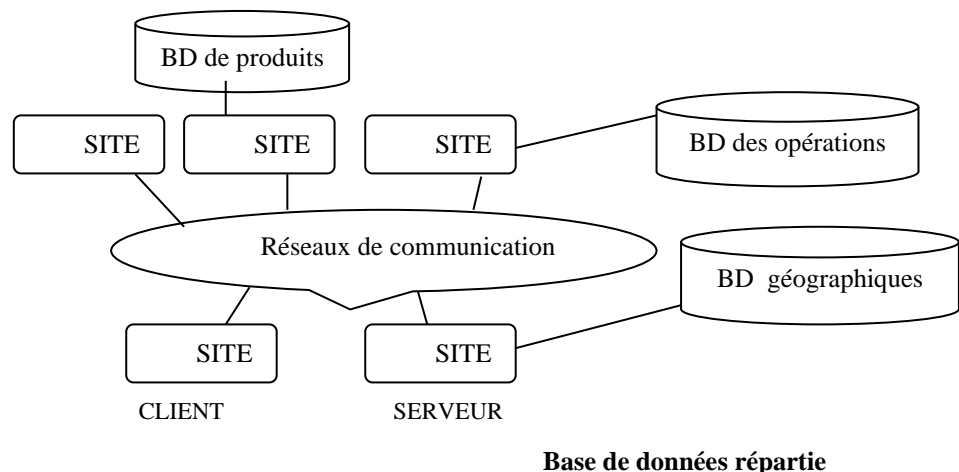
- le traitement global d'une requête SQL (décomposition en sous requête mono ou multi sites et optimisation de leur exécution).
- la synchronisation entre requêtes concurrentes et la cohérence globale en cas de pannes (protocole de terminaison multiphase)

Le traitement global d'une requête sous entend l'existence :

- d'un dictionnaire global (pouvant être centralisé ou distribué) dépositaires de l'information sur la localisation physique et assurant la transparence pour l'émetteur de la requête
- d'un traducteur pour accéder à plusieurs SGBD hétérogènes.

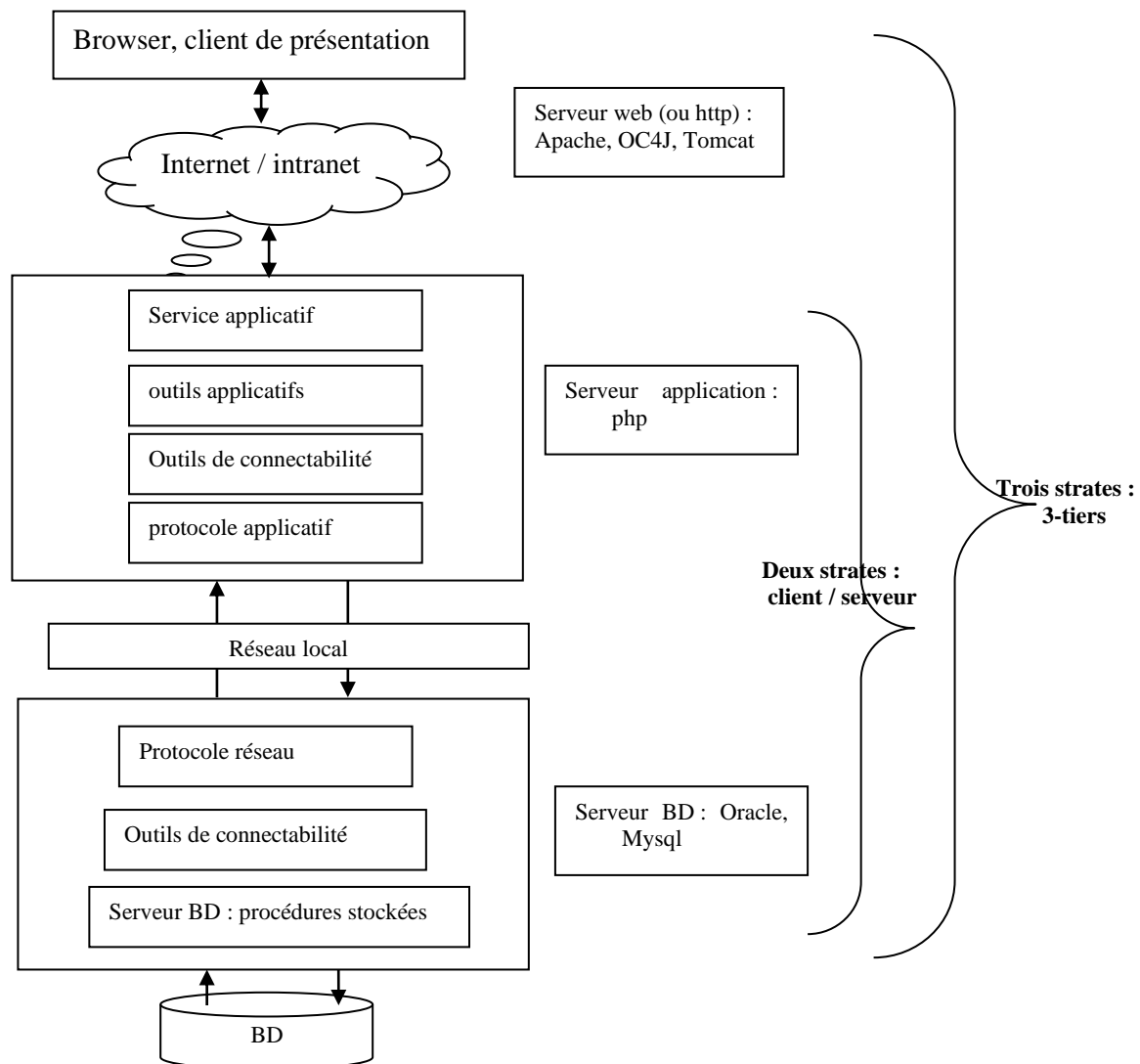
Un gestionnaire de ressources doit exister pour permettre et connaître l'état des ressources utilisées par chacun des traitements.

Les problèmes de conception d'applications réparties et d'interopérabilité hétérogène (SGBD, OS, Réseau) ne sont pas satisfaits pour mettre en œuvre des architectures performantes de ce type. Cela ne saurait tarder.



Ex : Googlearth

c-Architecture client- serveur trois niveaux (Web, Application, BD)



Architecture client- serveur trois niveaux (web, application, BD)

(Datacenter.)

d-Description client serveur (C/S)

- Cas de deux strates (C/S) :

L'application est écrite à l'aide d'un outil applicatif, souvent un L4G (L5G). Elle soumet ses demandes de services (requêtes au SGBD ou invocation de **procédures stockées** ou méthodes) au middleware qui les transfère au serveur. Le middleware comporte un composant client et un composant serveur qui permettent les échanges de commandes via le protocole réseau (outil de connectivité).

- Cas de trois strates (C/S) : (c/s généralisé).

Avec l'apparition d'internet et du web donc, les architectures C/S ont évolué vers des architectures à trois strates.

Le client est responsable de la présentation. Il utilise pour cela des web browsers.

Le serveur d'application exécute le code applicatif essentiel.

Le serveur de base de données supporte le SGBD et gère éventuellement des procédures stockées.

Dans l'architecture n-tiers, un client peut invoquer plusieurs serveurs. On parle alors d'architecture client multiserveur. L'inconvénient mais aussi l'avantage du C/S est de centraliser des données au niveau du serveur.

-Les différents modèles de bases de données (typologie)

- Les bases de données du modèle "relationnel" sont les plus répandues (depuis le milieu des années 80 environ jusqu'à aujourd'hui), car elles conviennent bien à la majorité des besoins des entreprises.

Le SGBD qui gère une BDD relationnelle est appelé "SGBD relationnel", ce qui est souvent abrégé en SGBDR. -D'autres modèles de bases de données ont été proposés : hiérarchique, en réseau, orienté objet, relationnel objet.

Aucun d'entre eux n'a pu détrôner le modèle relationnel, ni se faire une place notable sur le marché (sauf le relationnel objet, prôné par Oracle, qui connaît un certain développement).

-Malgré sa généralité, le modèle relationnel ne convient pas à toutes les BD rencontrées en pratique. Il existe donc des SGBD *spécialisés*. Les deux exemples les plus connus concernent la gestion des BD bibliographiques (ou documentaires), et celle des BD géographiques gérées à l'aide d'un SIG (Système d'Information Géographique, satellite).

MODELE RELATIONNEL

I- INTRODUCTION

Le modèle relationnel a été introduit par Edgard F. CODD dans les années 1970 au centre de recherche IBM à San José. Il existe en fait trois modèles de données principaux de haut niveau logique et organisationnel. Les premiers principaux modèles d'accès sont le modèle hiérarchique et le modèle réseau (navigationnel ou Codasyl : Conference On Data System Language). Le troisième qui a suivi est le modèle relationnel. La première volonté du modèle relationnel fut d'être un modèle ensembliste simple supportant des ensembles d'enregistrements aussi bien au niveau de la description que de la manipulation.

Le modèle relationnel est aujourd'hui à la base de nombreux systèmes et d'architecture permettant d'accéder depuis une station de travail à des serveurs de données. Le relationnel a atteint ses objectifs au delà de toute espérance en améliorant de façon significative (remarquable) les insuffisances des deux précédents.

Il convient de noter que depuis près de 40 ans trois grandes familles de modèles de données ont été définis en support d'implantation d'une base de données dans un SGBD.

Les modèles informatiques sont en général du ressort des informaticiens. Ici dans la partie définition du schéma et dans la partie manipulation des données, on a eu à utiliser le modèle hiérarchique et le modèle réseau sous le contrôle de DBTG (Data Base Task Group).

Les modèles mathématiques ou relationnels de CODD F.

Les modèles à sémantiques hybrides relationnel-objet.

Les modèles à sémantiques objets.

Edgar Frank « Ted » Codd (23 août 1923 - 18 avril 2003), informaticien britannique, considéré comme l'inventeur du modèle relationnel des SGBDR.

[rappel

L'établissement de ce cycle de conception a connu plusieurs étapes d'évolutions avant d'être adopté par la communauté des bases de données (BD). Cette évolution est liée aux propositions de différents modèles de données développés afin de gérer au mieux les données stockées. Les premiers modèles de données proposés à la fin des années 60, sont les modèles de données physiques.

Deux principaux modèles ont connu un certain succès et ont été utilisés dans d'importants systèmes industriels : le *modèle hiérarchique* utilisé dans le système IBM IMS, et le *modèle en réseau* utilisé dans le système **Codasyl** (Conference on Data Systems Languages) [49].

Ces deux modèles ont marqué une évolution importante dans le domaine des BD, dans la mesure où ils ont représenté les premiers modèles permettant de gérer et d'interroger l'ensemble de données d'une BD. Le modèle CODASYL a valu à son auteur **Charles Bachman**, le prix ACM Turing en 1973. Cependant, comme toute première proposition, ces modèles présentaient de nombreux inconvénients.

Le plus important est la difficulté de maintenance et d'évolution de la BD, due à la faible indépendance physique des données.

Afin de remédier à cet inconvénient, le modèle relationnel a été proposé par **Edgar F Codd** [50]. L'indépendance physique des données signifie la capacité de modifier le schéma interne de la BD sans avoir à modifier ni son schéma conceptuel, ni les schémas externes des utilisateurs.

Le modèle relationnel, a ainsi été conçu pour présenter une abstraction mathématique de ce qui est implémenté dans la BD. Le modèle relationnel est basé sur une représentation formelle. Il propose de stocker les données dans un modèle organisé en des structures simples et flexibles (des tables) pouvant représenter tout type d'information. Ce modèle marque une nouvelle évolution importante, sanctionnée également par un prix Turing en 1981. Le modèle relationnel a été implémenté dans d'importants projets académiques et industriels comme le projet System R, le projet Ingres et le projet Gamma [76].

]

1) Modèle hiérarchique

Bien avant ce modèle, on utilisait de système de fichiers pour les ensembles de données. (Voir en début du cours).

Une des structures les plus simples et les plus répandues existant dans tout l'univers réel est la structure hiérarchique. La conception de schéma dans le modèle hiérarchique dépend uniquement des liens d'association un à plusieurs (inter – entités).

Exemple : le SGBD IMS conçu en 1969 sur le système d'exploitation (OS) IBM/360, tous conçus par IBM (International Business of Machine).

-Avantages du modèle hiérarchique

- adéquation avec une entreprise à structure arborescente (déploiement) ;
- simplicité du modèle dont l'implantation est relativement facile ;
- rapidité d'accès aux données (lecture /écriture disque).

-Inconvénients

- impossible de représenter directement le lien d'association mailles (plusieurs à plusieurs) inter - entités ;
- beaucoup d'anomalies du modèle pour les opérations de stockages (modification mise à jour) ;
- absence de symétrie pour les requêtes de manipulation ;
- Indépendance logique très réduite ;
- absence d'interface utilisateur non procédural limitant les informations dans la définition et la manipulation de BD ;
- impossibilité de concevoir des requêtes mettant en relief des objets de la même génération ;
- coût prohibitif (excessif) des traitements ad hoc (imprévus).

2) Modèle réseau ou navigationnel (CODASYL : Conférences on data system language)

Il incorpore un langage d'interrogation (CODD Frédéric 1970-1978). Il conserve les liens un à plusieurs entre entités. L'usage de pointeurs logiques est nouveau.

Le schéma est représenté sous forme d'un graphe ou de réseau connectant les entités entre elles à l'aide de pointeurs logiques.

-Avantages du modèle navigationnel

- moins de rigidité donc accroissement de la flexibilité ;
- représentation manuelle des liens maillés ;
- absence d'anomalies pour les opérations de stockage ;
- rapidité des entrées /sorties disques.

-Inconvénients

- absence d'indépendance vis-à-vis des stratégies d'accès ;
- confusion du modèle conceptuel et du modèle des chemins d'accès ;
- procéduralité importante des langages de manipulation (l'utilisateur doit naviguer dans le réseau logique constitué par les enregistrements et les chaînes de pointeurs).

Certaines requêtes pour des besoins spécifiques sont impossibles.

2) Modèle relationnel

Le modèle relationnel vient combler les insuffisances des modèles précédents. Il ajoute quelques avantages :

- indépendance vis-à-vis des stratégies d'accès ;
- souplesse de manipulation de données d'une entité à une autre directement ou indirectement en association.

Il connaît tout de même des limitations ; ce qui nous a amenés au modèle objet pour mieux simuler la réalité.

II) MODELE RELATIONNEL ET SES 10 CONCEPTS FONDAMENTAUX

Le modèle relationnel est fondé sur une double base théorique mathématique : la théorie des ensembles et la théorie des prédicats.

Le modèle relationnel créé par E. F. Codd, connu sous le nom initial de modèle **V1**, constitue un des apports les plus remarquables à l'informatique dans son rapprochement vers le non-informaticien (non programmeur) résumé en quatre points :

a-Simplicité des concepts de base

Une relation par exemple, est un ensemble au sens mathématique qui va être visualisé à un instant donné par une table de valeurs. Le concept de table est un des concepts les plus faciles à appréhender. Il s'en suit également une possibilité d'apprentissage par des concepteurs ou utilisateurs non programmeur (dans les modèles hiérarchiques ou réseau des générations précédentes, concevoir et manipuler une base de données était exclusivement une affaire de programmeurs). Par là, le modèle relationnel satisfait plus l'indépendance des données.

b-Puissance des opérateurs de manipulation

Les opérateurs relationnels sont des opérateurs ensemblistes : l'application d'un opérateur génère une relation et donc un ensemble qui peut à son tour être manipulé. Un langage relationnel a donc un caractère dit non procédural.

c-Rigueur des concepts aussi bien des structures que des opérateurs avec la double base mathématique (théorie des ensembles et théorie des prédicats).

d-contribution à la cohérence du système d'information global de l'entreprise : un SGBD R peut être le support uniforme, s'appuyant donc sur une méthodologie et une formation unique des utilisateurs, pour toutes les applications de gestion ou de production, de Datawarehouse ou de serveur de données pour le commerce électronique.

10 concepts fondamentaux du modèle relationnel

Tout modèle de données doit avoir 3 composantes à savoir : définition, manipulation et contrôle c'est-à-dire :

- la **définition** concerne les structures permettant de spécifier les données.
- la **manipulation** concerne les opérateurs permettant de manipuler les données définies par les structures.
- le **contrôle** des règles d'intégrités induites par les structures et traduisant les contraintes liées à la définition ou la manipulation des données

Un SGBD qui implante un modèle de données doit automatiquement prendre en compte les règles d'intégrité et vérifier la cohérence de ces données.

Aussi et surtout de façon à éviter toute ambiguïté, les définitions d'un modèle de données doivent idéalement reposer sur des théories formelles. Le modèle relationnel a le privilège de reposer sur une double base formelle : la théorie ensembliste et la théorie des prédicats.

Composante définition des données

1-relation n-aire (tuples)

2-domaines –attributs

3-clé primaire

4-clé étrangère –domaine primaire

Composante intégrité des données

5--intégrité de domaines (liée à 2)

6-intégrité de relation (liée à 1 et 3)

7-intégrité de référence (liée à 4)

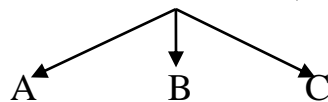
Composante manipulation des données

8-opérateurs sémantiques (liés aux domaines « domain driven operator »)

9-opérateurs ensemblistes (union, intersection, différence)

10-opérateurs relationnels (sélection, projection, jointure, division).

MODELE RELATIONNEL DE CODD (1970)



Les lettres A B et C sont définies pour traduire les dix concepts de base du modèle relationnel comme suit :

A : définition/structures	C : contraintes d'intégrité du modèle (pour le maintien de la cohérence)	B : manipulation (langage relationnel)
1- relation/attributs 2- domaines (sémantique) 3- clé primaire 4- domaine primaire (clé étrangère)	5- intégrité du domaine 6- intégrité d'entité (ou de relation) 7- intégrité de référence	8- équivalence avec une interface algébrique complète et opérateurs sémantiques (liés aux domaines) 8-opérateurs ensemblistes (union, intersection, différence) 10-opérateurs relationnels (sélection, projection, jointure, division).

DIX CONCEPTS DU MODELE RELATIONNEL

NB : Les autres concepts que l'on pourrait définir notamment contrôle de la concurrence, reprise sur pannes, sécurité, intégration de l'univers modélisé. Les vues ne sont pas spécifiques seulement au modèle relationnel mais caractéristiques de tout SGBD relationnel ou non.

Définition de relation et de domaine :

Le concept de relation étant le plus important dans le modèle de Codd, nous allons donner deux définitions une ensembliste et une prédictive.

-Définition ensembliste d'une relation :

Une relation R est définie comme un sous ensemble du produit cartésien de n ensembles D_i

R est inclus (large) dans le produit cartésien $D_1 \times D_2 \times \dots \times D_n$

Une relation est un ensemble composé d'éléments appelés n-uplets (les tuples au niveau logique sont les occurrences au niveau conceptuel).

Tuple en anglais et en français.

Couple, triplet, quadruplet, cinq uplets, ... , n-uplet.

Exemple

(100, Airbus, 300, Paris) est un quadruplet de la relation

AVION (AVNO#, AVNOM, CAP, LOCVILLE) avec 100 € AVNO (numéro d'avion), AIRBUS € AVNOM (nom d'avion), 300 € CAP (capacité) etc.

Le domaine LOCVILLE = {NICE, TOULOUSE, PARIS}.

Par ailleurs les attributs ADR de *PILOTE*, LOC de *AVION*, VD et VA de *VOL* prennent leurs valeurs dans le domaine *VILLE*.

Un domaine est un ensemble de valeurs caractérisé par un nom ou une propriété d'attribut.

-Définition prédicative d'une relation :

Prédicat : ce qui est dit de la chose ou d'une personne.

Une proposition est une forme d'énoncé qui est soit vraie ou fausse.

P1 : « il est beau » ;

P2 : « Socrate est mortel » ;

P3 « l'avion de numéro 100 est un Airbus de capacité 300 places localisé à Nice ».

Un tuple peut être perçu comme l'écriture concise d'une proposition. Par exemple à la proposition p3 correspond le tuple (100, Airbus, 300, NICE)

Une relation est un ensemble de tuples.

Objectifs du modèle relationnel selon CODD F.

Ce sont :

a1-Permettre un haut degré d'indépendance des programmes d'application et des activités interactives à la représentation interne des données en particulier au choix des ordres d'implantation des données dans les fichiers des indexes et chemin d'accès.

a2-Fournir une base solide pour traiter les problèmes de cohérence et de redondance des données (non atteint par le modèle hiérarchique)

Ces deux objectifs sont atteints grâce à la simplicité des vues relationnelles présentant les données sous forme de tables à deux dimensions et également grâce aux règles d'intégrités supportées par ce modèle et ses fondements logiques.

a3- Permettre le développement de LMD non procédural basé sur des théories mathématiques solides.

Cet objectif est atteint à l'aide de l'algèbre relationnelle qui permet de manipuler les données de manière très simple et formelle (via des opérations arithmétiques et aussi grâce au langage assertionnel basé sur la logique permettant de spécifier les données que l'on souhaite obtenir sans dire comment les obtenir).

Deux autres objectifs non prévus initialement par le modèle sont atteints à savoir :

a4-Etre un modèle extensible permettant de décrire et de manipuler simplement les données

tabulaires et pouvant être étendu à opérer de même sur des données beaucoup plus complexes.

a5- Ce modèle se veut devenir un standard pour la description et de la manipulation des BD.

Remarque : l'objectif 4 a permis d'intégrer de nouveaux concepts au relationnel notamment la plupart des concepts orientés objet (classe, encapsulation, polymorphisme, héritage, objet ...).

III) EXPRESSIONS DE L'ALGÈBRE RELATIONNELLE (Voir polycopé)

L'algèbre relationnelle s'inscrit dans le cadre typique du modèle relationnel. Elle a été inventée par CODD F. Comme une collection d'opérations formelles qui agissent sur des relations et produisent des relations en résultat. On peut considérer que l'algèbre relationnelle est aux relations, ce qu'est l'arithmétique aux entiers. Cette algèbre qui constitue un ensemble d'opérations élémentaires associés au modèle relationnel est sans doute une des forces essentielles du modèle. CODD a initialement introduit huit opérations dont certaines peuvent être composées à partir d'autres.

Les opérations de base peuvent être classées en deux (voire trois) catégories (opérations ensemblistes et opération spécifiques) :

- les **opérations ensemblistes** traditionnelles (union, différence, produit cartésien)
- les **opérations spécifiques** (projection, sélection ou restriction, jointure (équijointure, inéquijointure))
- **opérations dérivées** (intersection, division, complément, éclatement, jointure externe, semi jointure, fermeture transitive)

Les opérations ensemblistes sont des opérations binaires (à deux opérandes), c'est à dire qu'à partir de deux relations, elles en construisent une troisième. Ce sont l'union, la différence et le produit cartésien.

Les opérations spécifiques sont les opérations unaires de projections et restriction (sélection) qui, à partir d'une relation en produisent une autre, et l'opération binaire de jointure.

Notons qu'une relation est un ensemble de tuples.

1) Expression algébrique des opérations ensemblistes et formalisme

a- Union

L'union est l'opération classique binaire de la théorie des ensembles adaptée aux relations de même schéma.

C'est une opération portant sur deux relations de même schéma R1 et R2, consistant à construire une relation de même schéma R3 ayant des tuples ceux appartenant à R1 ou à R2 ou aux deux relations.

Notations utilisées :

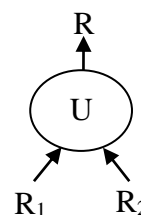
Expression ou notation algébrique

$$R = R1 \cup R2$$

$$R = \text{UNION} (R1, R2)$$

$$R = \text{APPEND} (R1, R2)$$

R est la relation resultat



Notation graphique

Exemple de schéma relationnel

En intention

Buveurs(nb, nom, adresse);

Vins (nv, Cru, mill, qualité, degré, Région, Couleur) ;

Bus(nb, nv, date, quantité).

En extensions de Vins suivies d'une union

Vins1	Cru	Mill	Région	Couleur
	Chenas	1983	Beaujolais	Rouge
	Tokay	1980	Alsace	Blanc
	Tavel	1986	Rhône	Rose

Vins2	Cru	Mill	Région	Couleur
	Tokay	1980	Alsace	Blanc
	Chablis	1985	Bourgogne	Rouge

Vins3 = Vins1 U Vins2	Cru	Mill	Région	Couleur
	Chenas	1983	Beaujolais	Rouge
	Tokay	1980	Alsace	Blanc
	Tavel	1986	Rhône	Rose
	Chablis	1985	Bourgogne	Rouge

NB: Les notations graphiques sont souvent celles utilisées dans la représentation d'une requête sous forme d'arbres algébriques.

[

Signification de CRU de la table VINS :

Un cru désigne un vignoble produisant un vin spécifique et, par extension, ce vin.

Le cru du vignoble : IG (vin avec Indication Géographique), VSIG (vin sans indication géographiques).

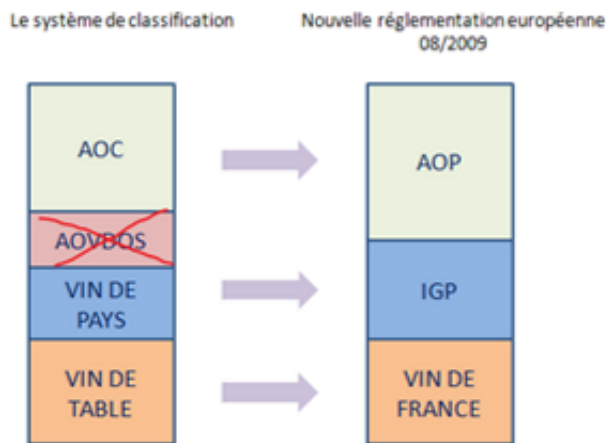
=> Vin de France,

Pour les VSIG : Le **vin de table** devient le **Vin de France** (oui, ça sonne mieux)

Pour les IG : Les AOC (Appellation d'Origine Contrôlée) deviennent les **AOP** (Appellation d'Origine Protégée).

Les vins de pays deviennent les **IGP**, (Indication Géographique Protégée)

Quant aux AOVDQS (Appellation d'origine vin délimité de qualité supérieure), ils disparaissent. Ils migrent au choix entre AOP ou IGP



]

b- Différence

La différence est également une opération classique binaire de la théorie des ensembles adaptée de même schéma. Les deux relations doivent avoir le même schéma.

C'est une opération portant sur deux relations de même schéma R_1 et R_2 , consistant à construire une relation de même schéma R_3 ayant des **tuples** ceux appartenant à R_1 et n'appartenant pas à R_2 .

C'est un opérateur non commutatif. L'ordre des opérandes est donc important.

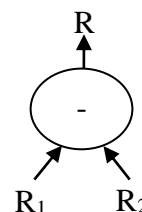
Notations utilisées :

$R = R_1 - R_2$

$R = \text{DIFFÉRENCE}(R_1, R_2)$

$R = \text{REMOVE}(R_1, R_2)$

$R = \text{MINUS}(R_1, R_2)$



Exemple

Notation graphique

Vins1	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
	Chenas	1983	Beaujolais	Rouge
	Tokay	1980	Alsace	Blanc
—	Tavel	1986	Rhône	Rose

Vins2	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
	Tokay	1980	Alsace	Blanc
	Chablis	1985	Bourgogne	Rouge

Vins4 = vins1 - Vins2	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
	Chenas	1983	Beaujolais	Rouge
	Tavel	1986	Rhône	Rose

c- Produit cartésien (cartesian product)

Le produit cartésien est une opération binaire adaptée aux relations. Les deux relations n'ont pas nécessité d'avoir même schéma.

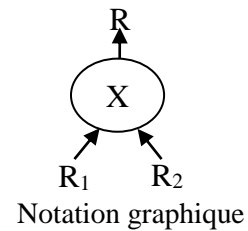
C'est une opération portant sur deux relations R1 et R2, consistant à construire une relation R3 ayant pour schéma la concaténation de ceux des relations opérandes et pour tuples toutes les combinaisons des tuples des relations opérandes.

Notations utilisées :

$R = R1 \times R2$

$R = \text{PRODUIT}(R1, R2)$

$R = \text{TIMES}(R1, R2)$



VinsA	<i>Cru</i>	<i>Region</i>	<i>Couleur</i>
	Chenas	Beaujolais	Rouge
	Tokey	Alsace	Blanc
	Tavel	Rhone	Rose

X

Année	<i>Mill</i>
1980	
1985	

↓

Vins6 = VinsA X Année	<i>Cru</i>	<i>Région</i>	<i>Couleur</i>	<i>Mill</i>
	Chevas	Beaujolais	Rouge	1980
	Tokey	Alsace	Blanc	1980
	Tavel	Rhone	Rose	1980
	Chevas	Beaujolais	Rouge	1985
	Tokey	Alsace	Blanc	1985
	tavel	Rhone	Rose	1985

2) Expression algébrique des opérations spécifiques (relationnelles) et formalisme

a- Projection (coupure verticale)

La projection est une opération spécifique aux relations qui permet de supprimer des attributs d'une relation. Elle est unaire. Son nom provient du fait qu'elle permet de passer d'une relation n-aire à une relation p-aire avec $p \leq n$; donc d'un espace à n dimensions à un espace à moins de dimensions.

C'est une opération sur une relation R1 consistant à composer une relation R2 en enlevant à la relation initiale R1 tous les attributs non mentionnés en opérandes (aussi bien au niveau du schéma que des tuples) et en éliminant les tuples de R1 en doubles qui sont conservés une seule fois.

Notations utilisées :

En désignant par A_1, A_2, \dots, A_n les attributs de la projection on a :

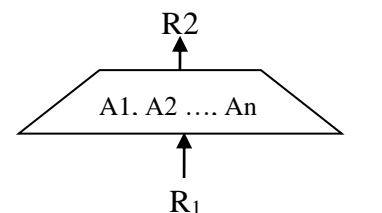
$R2 = \Pi A_1, A_2, \dots, A_n (R1)$

$R2 = R1[A_1, A_2, \dots, A_n]$

$R2 = \text{PROJECT}(R1, A_1, A_2, \dots, A_n)$

$R2 = \text{PROJECTION}(R1, A_1, A_2, \dots, A_n)$

$R2 = \text{PROJECTION } R1 (A_1, A_2, \dots, A_n)$



Notation graphique de la projection

Exemple

Vins1	Cru	Mill	Région	Qualité
	Volnay	1983	Bourgogne	A
	Volnay	1980	Bourgogne	B
	Chenas	1986	Beaujolais	A
	Julienas	1986	Beaujolais	C

Π Cru, Région

↓

Π (Vins1)	Cru	Région
	volnay	Bourgogne
	chenas	Beaujolais
	julienas	Beaujolais

NB : La projection est une coupure verticale.

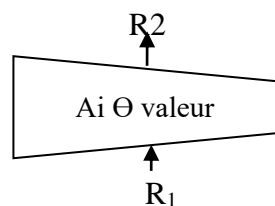
b- Sélection ou restriction (coupure horizontale)

La restriction est aussi une opération spécifique unaire, qui produit une nouvelle relation en enlevant des tuples à la relation opérande selon un critère (prédicat ou qualification).

C'est une opération sur une relation R1 produisant une relation R2 de même schéma, mais comportant les seuls tuples qui vérifient la condition précisée en arguments.

Les conditions possibles sont du type : $\langle \text{attribut} \rangle \langle \text{opérateur} \rangle \langle \text{valeur} \rangle$ où l'opérateur est un opérateur de comparaison choisi parmi $\{=, <, >, <=, >=, \neq\}$.

Il est possible d'utiliser des compositions ou combinaisons logiques de critères comprenant des conjonctions (et) et des disjonctions (ou) ou des combinaisons des deux.



Notation graphique de la sélection

Notations utilisées :

$$R2 = \delta_{\text{condition}}(R1)$$

$$R2 = R1[\text{condition}]$$

$$R2 = \text{RESTRICT}(R1, \text{condition})$$

$$R2 = \text{RESTRICTION}(R1, \text{condition})$$

Exemple

Vins1	Cru	Mill	Région	Qualité
	volnay	1983	Bourgogne	A
	volnay	1980	Bourgogne	B
	chenas	1984	Beaujolais	A
	julienas	1986	Beaujolais	C

$\delta_{\text{mill} > 1983}$

↓

Vins1	Cru	Mill	Région	Qualité
	chenas	1984	Beaujolais	A
	julienas	1986	Beaujolais	C

c- jointure

La jointure est une des opérations essentielles de l'algèbre relationnelle, sans doute la plus difficile à réaliser par les systèmes. Elle permet de composer deux relations à l'aide d'un critère de jointure. Elle peut-être vue comme une extension du produit cartésien avec une condition permettant de comparer les attributs. Elle est binaire.

C'est une opération consistant à rapprocher selon une condition les tuples de deux relations R1 et R2 afin de former une troisième relation R3 qui contient l'ensemble de tous les tuples obtenus en concaténant un tuple de R1 et un tuple de R2 vérifiant la condition de rapprochement.

Les conditions possibles sont du type : $\langle \text{attribut1} \rangle \langle \text{opérateur} \rangle \langle \text{attribut2} \rangle$ où attribut1 appartient à R1 et attribut2 à R2.

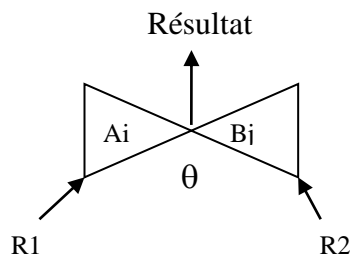
Selon le type d'opérateur on distingue :

- L'équi-jointure si l'opérateur est $=$.
- L'inéqui jointure dans les autres cas c'est-à-dire avec un des opérateurs $\{<, >, <=, >=, \neq\}$.

Il est possible d'utiliser des compositions logiques de critères comprenant des conjonctions (et) et des disjonctions (ou).

Dans le cas de l'équi jointure, les deux attributs égaux apparaissent chacun dans le résultat. Il y a donc duplication d'une même valeur dans chaque tuple. Afin d'éliminer cette redondance, on définit la jointure naturelle. Jointure naturelle comme suit :

Opération consistant à rapprocher les tuples de deux relations R1 et R2 afin de former une troisième relation R3 dont les attributs sont l'union des attributs des relations R1 et R2, et dont les tuples sont obtenus en composant un tuple de relation R1 et un tuple de R2 ayant même valeur pour les attributs de même nom.



Représentation graphique de la jointure

Notations utilisées :

$R1 \bowtie R2$
condition


Join (R1, R2, condition)


Où A_i appartient à R1 et B_j à R2.


Exemples

Les exemples suivants illustrent cette opération de jointure sur les relations VINS et LOCALISATION.

Qualité de vin par ordre décroissant donné par les lettres A, B, C, D, E.


VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Chablis	1983	A
	juliéna	1986	C


LOCALISATION	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B


VINS_REG	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	1983	A	Bourgogne	A
	Volnay	1979	B	Bourgogne	A
	Chablis	1983	A	Bourgogne	A
	Chablis	1983	A	Californie	B


Il s'agit d'une jointure naturelle des relations VINS et LOCALISATIONS

Condition : $Qualité \neq QualMoy$

VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Chablis	1983	A
	juliéna	1986	C


 $Qualité \neq QualMoy$

LOCALISATION	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B

VINS_LOC	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	1983	A	Chablis	Californie	B
	Volnay	1979	B	Volnay	Bourgogne	A
	Volnay	1979	B	Chablis	Bourgogne	A
	Chablis	1983	A	Chablis	Californie	B
	Julienas	1986	C	Volnay	Bourgogne	A
	Julienas	1986	C	Chablis	Bourgogne	A
	Julienas	1986	C	Chablis	Californie	B

C'est une inéquijointure des relations VINS et LOCALISATION.

Au regard de la condition ou critère de la jointure, il est fait usage des attributs Qualmoy et Qualité pour la formaliser.

3) Expression algébrique des opérations dérivées et formalisme

Les opérations suivantes sont parfois utilisées pour manipuler des relations. Elles peuvent en général être obtenues par combinaisons des précédentes.

a -Intersection

L'intersection est une opération classique de la théorie adaptée aux relations de même schéma.

C'est une opération portant sur deux relations R_1 et R_2 de même schéma, consistant à construire une relation R_3 ayant pour les tuples ceux appartenant à la fois à R_1 et à R_2 .

Notations utilisées :

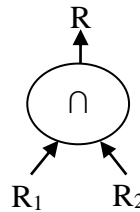
Expression algébrique

$R = R_1 \cap R_2$

$R = \text{INTERSECT}(R_1, R_2)$

$R = \text{INTERSECTION}(R_1, R_2)$

$R = \text{AND}(R_1, R_2)$



Représentation graphique de l'intersection

Exemple

En extension

	Vins1	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
		Chenas	1983	Beaujolais	Rouge
		Tokay	1980	Alsace	Blanc
		Tavel	1986	Rhône	Rose

	Vins2	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
		Tokay	1980	Alsace	Blanc
		Chablis	1985	Bourgogne	Rouge

Vins 5 = vins 1 \cap Vins 2	<i>Cru</i>	<i>Mill</i>	<i>Région</i>	<i>Couleur</i>
	Tokay	1980	Alsace	Blanc

Remarque: $A \cap B = A - (A - B) = B - (B - A)$

L'intersection est à la fois une opération dérivée et ensembliste.

b-division

La division est une opération ensembliste permettant de rechercher dans une relation les sous tuples complétés par tous ceux d'une autre relation. Elle permet ainsi d'élaborer la réponse à des questions de la forme « quelque soit, x, trouvez y ».

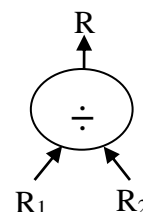
C'est une opération consistant à construire le quotient de la relation

$D(A_1, A_2, A_3, \dots, A_p, A_{p+1}, \dots, A_n)$ par la relation $d(A_{p+1}, \dots, A_n)$ comme la relation $Q(A_1, A_2, A_3, \dots, A_p)$ dont les tuples sont ceux qui concaténés à tout tuple de d donnent un tuple de D .

De manière plus formelle, désignons par a_i , une valeur quelconque de l'attribut A_i . Un tuple est alors une suite des valeurs $\langle a_1, a_2, a_3, \dots \rangle$. Le quotient Q de D par d est défini par : $Q = \{ \langle a_1, a_2, \dots, a_p \rangle \text{ tel que, quelque soit } \langle a_{p+1}, \dots, a_n \rangle \text{ de } d, \langle a_1, a_2, \dots, a_p, a_{p+1}, \dots, a_n \rangle \text{ appartient à } D \}$. Les notations possibles sont :

$Q = D \div d$

$Q = \text{division}(D, d)$ avec Q relations résultat.



Représentation graphique la division

$$R = R1 \div R2$$

Exemple

VINA	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
÷	Volnay	1983	A
	Volnay	1979	B
	Chablis	1983	A
	Chablis	1979	A
	Chevas	1986	A

QUALITE	<i>Mill</i>	<i>Qualité</i>
↓	1983	A
	1979	A

VINSB	<i>Cru</i>
	Chablis

$$VINSB = VINS A / QUALITE$$

Remarque :

L'intersection est une opération dérivable des opérations typiques de base (union, différence, produit cartésien). Elle est donc une opération redondante avec les opérations de base, puisque qu'il est possible de l'obtenir à partir de la différence à l'aide d'une des formules suivantes :

$$A \cap B = A - (A - B) = B - (B - A)$$

La division peut être obtenue à partir d'opération de base à savoir la différence, le produit cartésien, la projection comme suit :

$$D \div d = R1 - R2 \text{ avec}$$

$$R1 = \Pi A_1, A_2, \dots, A_p (D) \text{ et}$$

$$R2 = \Pi A_1, A_2, \dots, A_p ((R1 \times d) - D)$$

c-Complément

le complément permet de trouver les tuples qui n'appartiennent pas à une relation. Il suppose à priori que les domaines sont finis. (sinon on obtient des relations infinies)

C'est un ensemble des tuples du produit cartésien des domaines des attributs d'une relation n'appartenant pas à cette relation.

Notations utilisées :

-Not (R1)

-Comp (R1)

Domaines CRU = {chablis, volnay, médoc}

Couleur = {Rouge, Blanc, Rosé}

Exemple

COUL_CRU	<i>Cru</i>	<i>couleur</i>
	Chablis	Rouge
	Chablis	Rosé
	Volnay	Rouge
	Médoc	Rosé
	Médoc	Blanc

Not(COUL_CRU)	<i>Cru</i>	<i>couleur</i>
	Chablis	Blanc
	Volnay	Rosé
	Volnay	Blanc
	Médoc	Rouge

Parlant de produit cartésien de cu et de couleur, on obtient 9 tuples. Le complément sur 5 tuples donne trivialement 4 tuples.

d-Eclatement

L'éclatement est une opération qui n'appartient pas vraiment à l'algèbre relationnelle puisqu'elle donne deux relations en résultats, à partir d'une. Elle est cependant utile pour partitionner une relation horizontalement en deux sous relations. A ce titre elle est concernée par une extension de l'algèbre relationnelle.

C'est une opération consistant à créer deux relations à partir d'une relation R1 et d'une condition, la première contenant les tuples de R1 respectant la condition et la deuxième ceux ne la vérifiant pas.

Notations utilisées :

R1 = RESTRICT (R, condition)

R2 = RESTRICT(R, not (condition))

e-Jointure externe

- Jointure externe

Les jointures telles que définies précédemment perdent des tuples d'au moins une relation quand les jointures n'ont pas de projections identiques sur l'attribut de jointure. Pour préserver toutes les informations dans tous les cas, il est nécessaire de définir des jointures qui conservent les tuples sans correspondant avec des valeurs nulles associées quand nécessaire. C'est dans ce but que CODD (1979) a introduit les jointures externes.

C'est donc une opération générant une relation R3 à partir de deux relations R1 et R2, par jointure de ces deux relations et ajout des tuples de R1 et de R2 ne participant pas à la jointure, avec des valeurs nulles pour les attributs de l'autre relation.

Cette opération est très utile, en particulier pour proposer ces vues sans perte d'information.

Notation

- RELATION1  RELATION2

- EXT-Join (RELATION1, RELATION2)

Remarque : La jointure externe permet par exemple de joindre des tables clients et commandes sur un numéro de client commun, en gardant les clients sans commande et les commandes sans clients associés.

Exemple

VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Juliéas	1986	C



LOCALISATION	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B



VINSREG	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	1983	A	Bourgogne	A
	Volnay	1979	B	Bourgogne	A
	Chablis	-	-	Bourgogne	A
	Chablis	-	-	Californie	B
	Juliéas	1986	C	-	-

-Dérivées de la jointure externe

Aussi, on peut distinguer la jointure externe droite qui garde seulement les tuples sans correspondant de la relation de droite. Elle se note :

- R1 R2 ou REXT-JOIN

- R1 R2 ou LEXT-JOIN

Exemples (en exercices)

- *jointure externe gauche R1= vins et R2 = Localisation*

VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Juliéas	1986	C



LOCALISATION	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B



VINSREG	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	1983	A	Bourgogne	A
	Volnay	1979	B	Bourgogne	A
	Juliéas	1986	C	-	-

- *jointure externe droite R1= vins et R2 = Localisation*

VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Juliéas	1986	C



LOCALISATION	<i>Cru</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B



VINSREG	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>	<i>Région</i>	<i>QualMoy</i>
	Volnay	1983	A	Bourgogne	A
	Volnay	1979	B	Bourgogne	A
	Chablis	-	-	Bourgogne	A
	Chablis	-	-	Californie	B

f-Semi-jointure

Dans certains cas, lors de l'exécution d'une jointure, il est de conserver tous les attributs de deux relations résultats : seuls les attributs d'une des deux relations sont conservés. A ce titre une opération spécifique appelée semi-jointure, très utile pour optimiser l'évaluation des questions a été définie.

C'est donc une opération portant sur deux relations R1 et R2, donnant en résultat les tuples de R1 qui participent à la jointure des relations.

Notation :

- R1 R2

- SEMI-JOIN (R1, R2)

Elle est équivalente à la jointure des relations R1 et R2 suivis par une projection du résultat sur les attributs de la relation R1.

Exemple

VINS	<i>Cru</i>	<i>Mill</i>	<i>Qualité</i>
	Volnay	1983	A
	Volnay	1979	B
	Chablis	1983	A
	Juliéna	1986	C



LOCALISATION	Cru	Région	QualMoy
	Volnay	Bourgogne	A
	Chablis	Bourgogne	A
	Chablis	Californie	B



VINSREG	Cru	Mill	Qualité
	Volnay	1983	A
	Volnay	1979	B
	Chablis	1983	A

g-Fermeture transitive

C'est une opération très particulière permettant d'ajouter des tuples à une relation.

Elle n'appartient pas à l'algèbre relationnelle, mais peut être vue comme une de ses extensions. Il n'est pas possible de constituer cette opération avec un nombre fixe d'opérations de l'algèbre relationnelle : elle peut être effectuée par une série de jointure/projection/union. Mais le nombre d'opérations à effectuer dépend du contenu de la relation.

C'est donc une opération sur une relation R à deux attributs (a1, a2) de même domaine consistant à ajouter à R tous les tuples qu se déduisent successivement par transitivité, c'est à dire que si l'on a des tuples <a,b> et <b,c>, on ajoute <a, b>.

Notation

-close(R)

-R⁺

-T (R)

...

Calcul

Il est possible de générer des données non tuples à partir des tuples des relations. Ce sont les agrégats u données calculées pour lesquelles des fontions mathématiques existent : sum, avg, count ...

Remarque :

Il est possible de générer des données non tuples à partir des tuples. Ce sont les agrégats ou données calculées pour les quelles des fonctions mathématiques existent : sum, average, count, ...

Expression de l'algèbre relationnelle

A partir des opérations de l'algèbre relationnelle, il est possible de composer un langage d'interrogation de base de données. Une question peut alors être représentée par un arbre d'opérateurs relationnels ou algébriques.

Exercices : Base de données Dégustation dont le schéma est :

BUVEURS (nb, nom, prénom, adresse, type)

VINS (nv, cru, mil, qualité, degré)

ABUS (nb, nv, date, quantité)

1) Donner les degrés des vins de crus Morgan et de Millésime 1978.

Solution 1

R1 = Restrict (VINS, crus = "Morgan") ou R1 = SELECTION (VINS, crus = "Morgan")

R2 = Restrict (VINS, Mill = "1978")

R3 = Intersect (R1, R2)

Résultat = Project (R3, degré)

Deux autres versions

R1 = Restrict (VINS, crus = "Morgan")

R2 = Restrict (R1, Mill = "1978")

Résultat = Project (R2, degré)

R1 = Restrict (VINS, crus = "Morgan" et Mill = "1978")

Résultat = Project (R1, degré)

2) donnez les noms et les prénoms des buveurs de morgon ou de chevas.

Solution2

R1 = Restrict (VINS, cru = "morgon")

R2 = Restrict (VINS, cru = "chevas")

R3 = Union (R1, R2)

R4 = Join (R3, Abus)

R5 = Join (R4, Buveurs)

Résultat = Project (R5, nom, prénom)

3) Donnez les noms et adresse des buveurs ayant but plus de 10 bouteilles de chablis 1976 avec le degré de vins.

Solution3

R1 = Restrict (Abus, Qualité > 10)

R2 = Restrict (VINS, Cru = "Chablis")

R3 = Restrict (R2, Mill = 1976)

R4 = Restrict (R1, R3)

R5= Project (R4, Nb, Degré)
R6= Join (R5, Buveurs)
Résultat= Project (R6, Nom, Adresse, Degré).

Ou

R1= Restrict (VINS, Cru= "Chablis")
R2= Restrict (VINS, Mill= 1976)
R3= Restrict (Abus, Qualité>10)
R4=Intersect (R1,R2)
R5=Project (R4,degré)
R6=Join(R3,R5)
R7=Join(R6,Buveur)
R8=Project (R7, nom,adresse,degré)

4) *Donnez les noms des buveurs n'ayant bu que du morgon.*

Solution4

R1= Join (Buveurs, Abus, Vin)
R2= Restrict (R1, Cru= "Morgon")
R3= Project (R2, nom)
R4= Restrict (R1, Cru ≠ "Morgon")
R5= Project (R4, nom)
Résultat= Minus (R3, R5).

ARBRE ALGEBRIQUE

Une question exprimée sous forme d'un programme d'opérations de l'algèbre relationnelle peut être représentée par un arbre relationnel ou arbre algébrique..

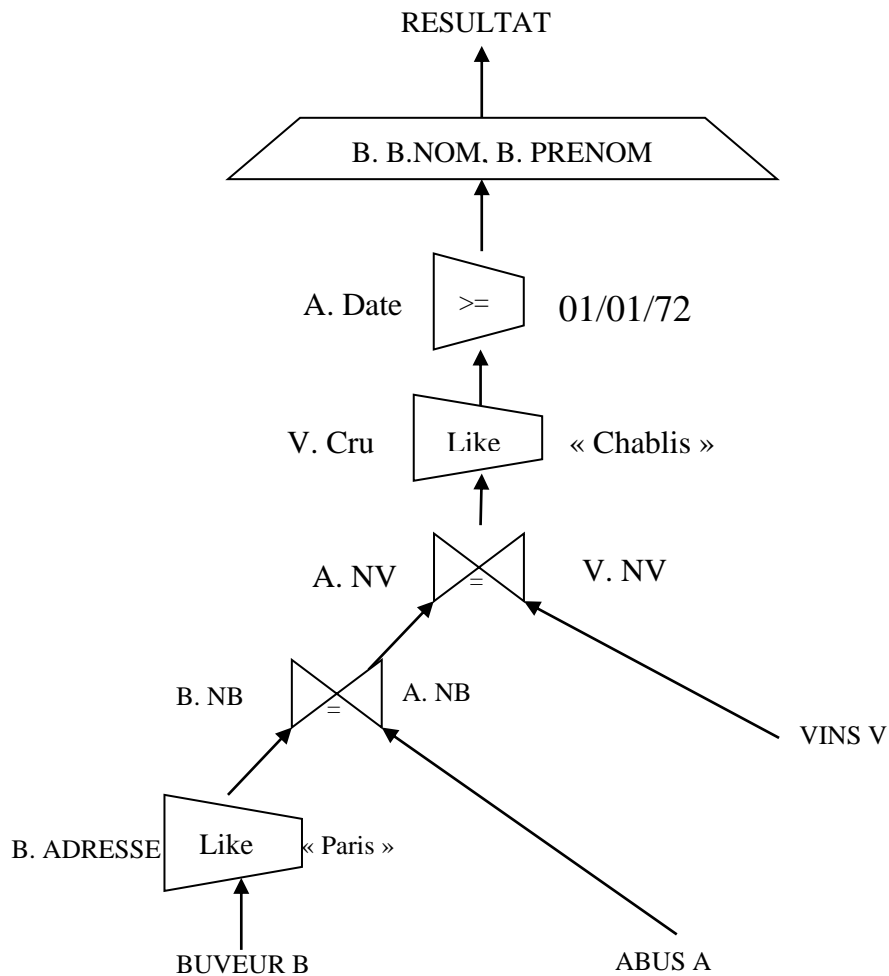
C'est un arbre dont les nœuds correspondent à des opérations algébriques de relation et les arcs à des relations de base ou temporaires représentant des flots de données entre opérations.

Ainsi la question de trouver les ***noms et prénom des buveurs habitants Paris ayant bu du chablis depuis 1972*** peut être exprimée à l'aide d'un arbre dit arbre relationnel.

Plusieurs arbres équivalents peuvent être déduits d'un arbre donné à l'aide de règles de transformation simples telles que :

La permutation des jointures et restrictions, la permutation des projections et des jointures, le regroupement des intersections sur une même relation, etc. Ces transformations sont à la base des techniques d'optimisation des questions.

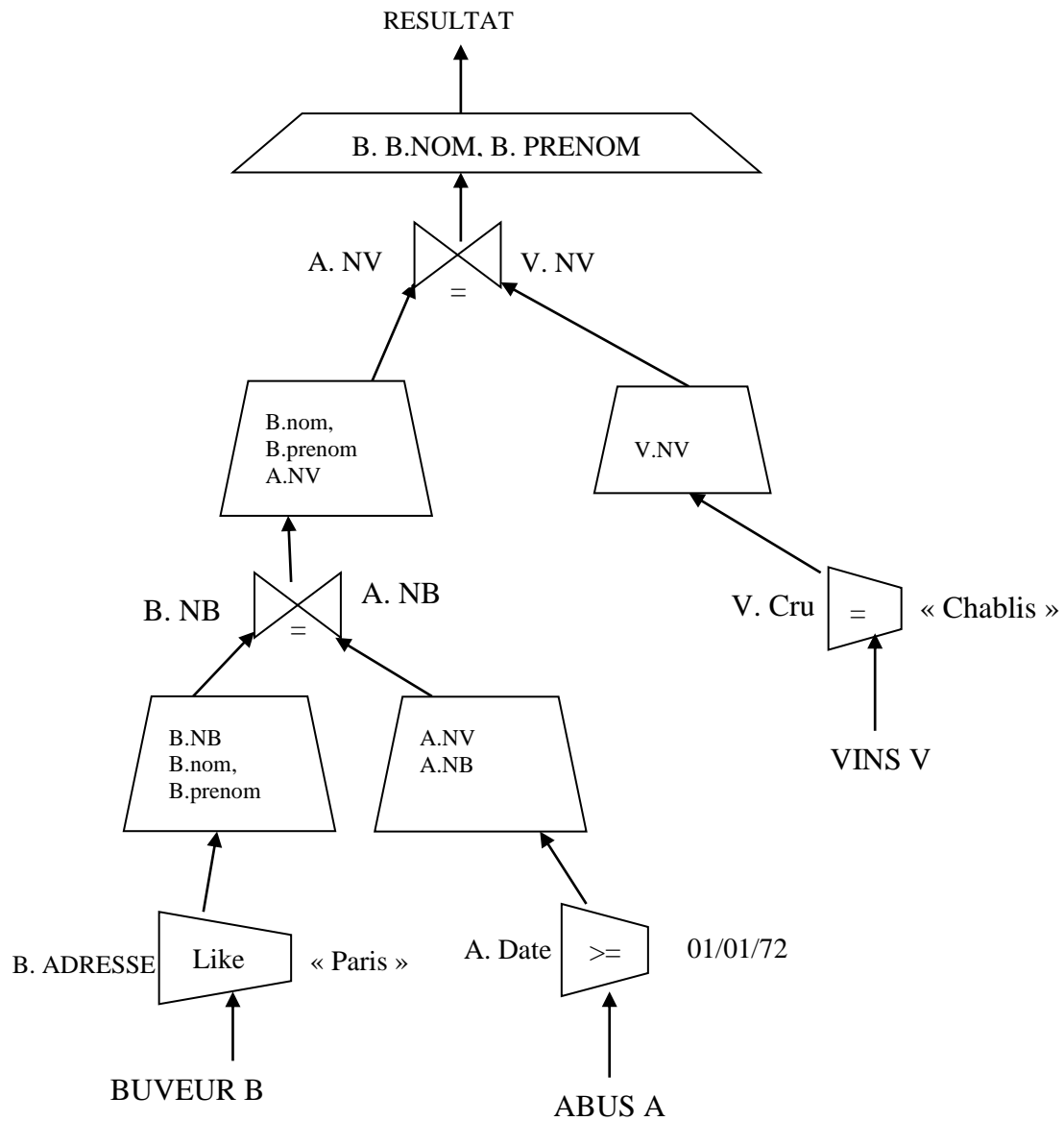
Exemple



La composition d'opérations de l'algèbre relationnel ne nécessite pas toujours d'attendre le résultat de l'opération précédente pour exécuter l'opération suivante. Restriction, projection et jointure peuvent être exécutées par des algorithmes à flots de données. Des opérateurs se succédant sur un arbre, peuvent être donc exécutés en parallèle par des algorithmes « pipe line ». Des opérateurs figurant sur des branches distinctes d'un arbre peuvent aussi être exécutés en parallèle de manière indépendante.

La représentation par arbre algébrique met ainsi en évidence des possibilités de parallélisme et des enchaînements nécessaires. Les propriétés des arbres relationnels sont importantes pour optimiser les questions dans les contextes parallèles.

Exemple :



Arbre équivalent à l'arbre précédent

L'optimisation des requêtes se base sur des notions en termes de commutativité ou d'associativité de certaines opérations.

LANGAGE SQL2

Les serveurs de données relationnels présentent de nos jours une interface externe sous forme d'un langage de recherche et de mise à jour. Cette interface permet de spécifier les ensembles de données à sélectionner ou à mettre à jour à partir de propriétés de valeur sans dire comment trouver les données.

Ainsi, les opérateurs directement utilisables par les usagers sont en général ceux des langages dits assertionnels (QUEL, QBE, SQL).

SQL est normalisé depuis les années 197-1989 par l'ISO.

SQL est le nom générique d'un standard constitué par une famille de langages de définition, de manipulation et de contrôle de données relationnelles.

Il est issu du langage SEQUEL Structured English as a QUERY Language.

Il constitue le standard d'accès aux BD assertionnelles. Les autres interfaces par menu, fenêtre, grille, ...ou de programmation de type langage de 3^e ou 4^e génération sont le plus souvent offertes au dessus du langage SQL. Celui-ci constitue donc le point d'entrée obligatoire dans les SGBD relationnels. De manière générale, SQL utilise des critères de recherche (encore appelés **qualification**) conçus à partir de la logique des prédicats de premier ordre.

Il comporte quatre opérations de base (données) utilisant des clauses (règles de base, primitives, mots clés).

- recherche (mot clé SELECT en SQL ; RETRY en QUEL)
- Insertion (mot clé INSERT en SQL ; APPEND en QUEL)
- Suppression (mot clé DELETE en SQL ; SUPPRESS en QUEL)
- Modification (ALTER en SQL).

What is SQL?

SQL stands for Structured Query Language.

SQL lets you access and manipulate databases.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

What Can SQL do?

SQL can execute queries against a database.

SQL can retrieve data from a database.

SQL can insert records in a database.

SQL can update records in a database.

SQL can delete records from a database.

SQL can create new databases.

SQL can create new tables in a database.

SQL can create stored procedures in a database.

SQL can create views in a database.

SQL can set permissions on tables, procedures, and views.

Modification structurelles (schéma)

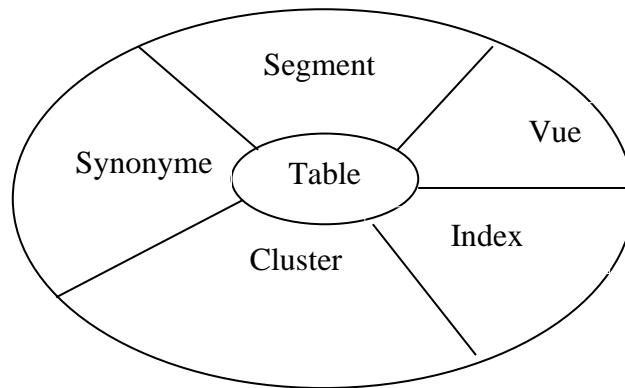
ALTER table <nom-table> <alteration>

Ajout d'une colonne : add column

Modification d'une colonne : alter column

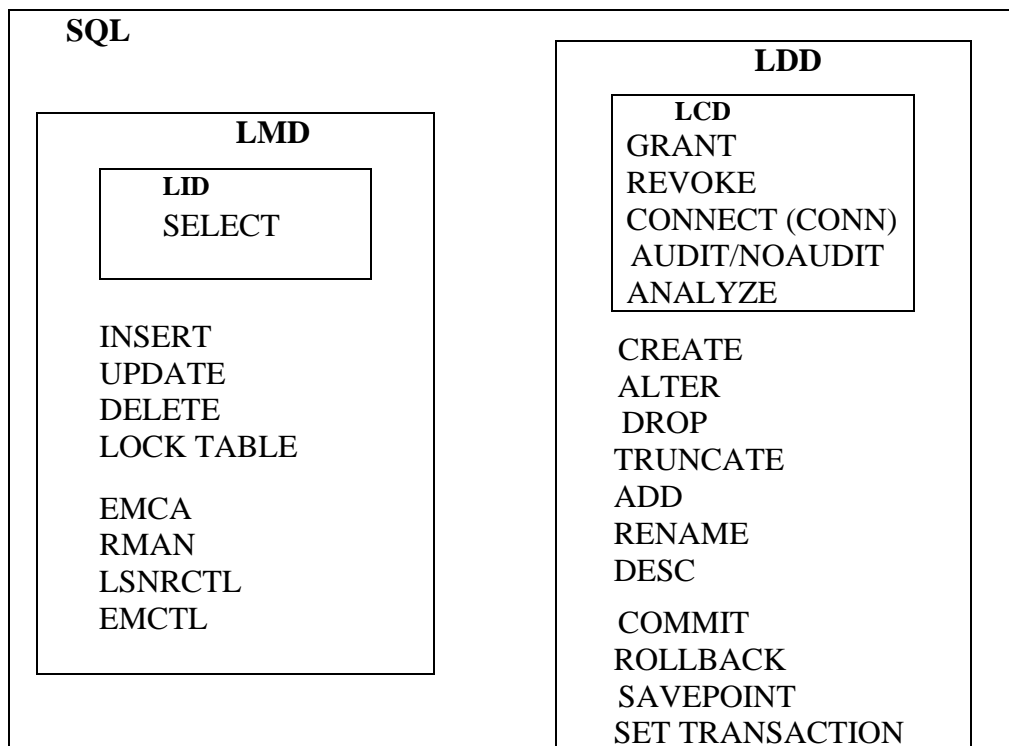
Ajout d'une contrainte : drop constraint

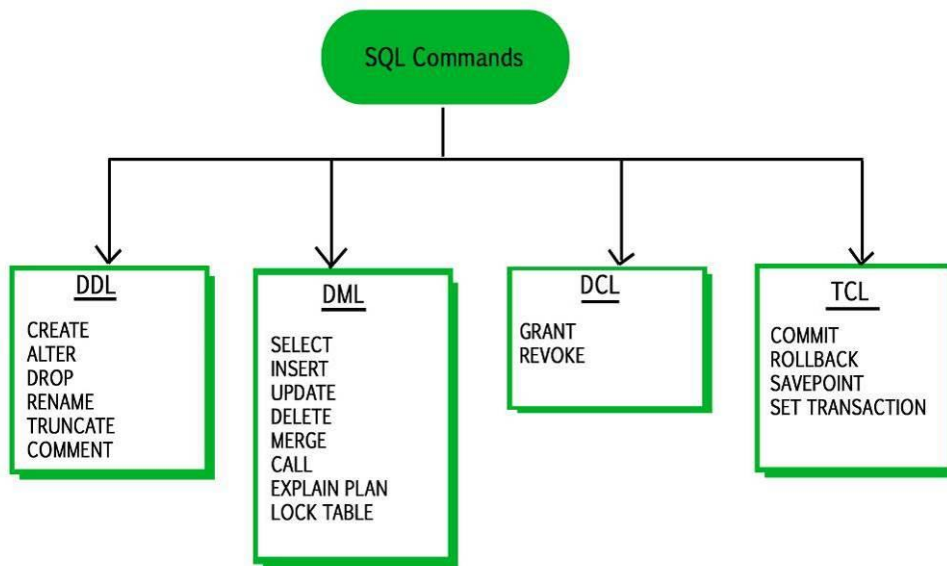
La table constitue l'élément central autour duquel gravitent d'autres objets (index)



-Il a deux grandes composantes **LDD (DDL)** et **LMD (DML)**.

-Serveur de base de données = Instance BD + BD





]

----------***-----***-----

(Dans ORACLE enterprise Linux : oraclce pwd kouame)

Se connecter comme **sysdba**

Pour se déconnecter de la base oracle (orcl), faire un shutdown immediate pour relancer la Base de données :

faire

faire \sqlplus / as sysdba

Faire **startup**

Faire **open**

Se reconnecter en tant que utilisateur connu sinon les objets seront accédés comme suit : scott.emp ;

----------***-----***-----

Oracle linux enterprise installé en virtualisation (VMWare Worstation 10.0.0 build-1295980)

Pour lancement

sqlplus /nolog

puis

conn / as sysdba,

starup

ou

sqlplus / as sysdba

starup

Une fois la bd de donnée est montée on peut avoir accès selon les droits :

desc dict

select

desc cats

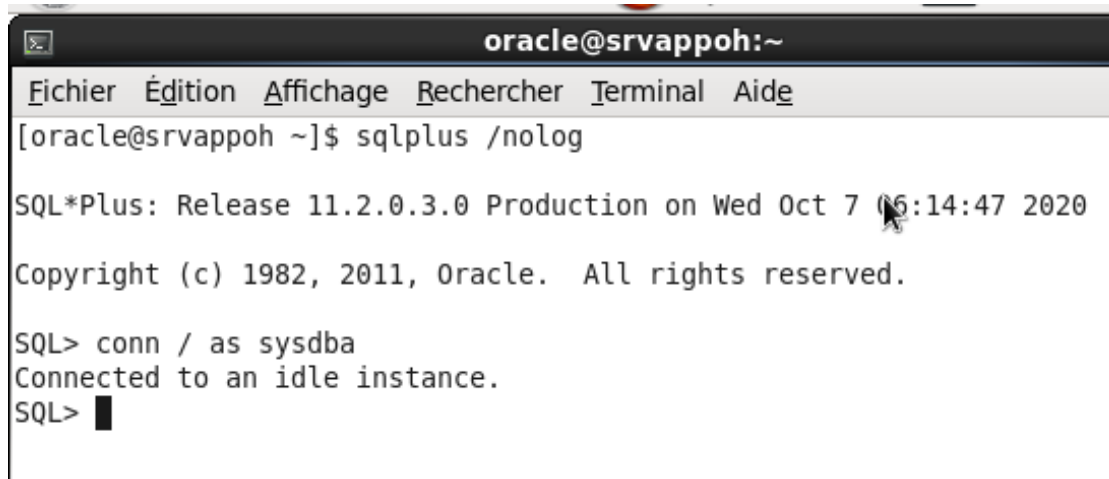
desc cat

select substr(table_name, 1, 10) from cat

desc v\$database

Pour fermeture (démonté la BD)

shutdown immediate

A screenshot of a terminal window titled 'oracle@srvappoh:~'. The window has a menu bar with 'Fichier', 'Édition', 'Affichage', 'Rechercher', 'Terminal', and 'Aide'. The terminal shows the following text:

```
[oracle@srvappoh ~]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Wed Oct 7 06:14:47 2020

Copyright (c) 1982, 2011, Oracle. All rights reserved.

SQL> conn / as sysdba
Connected to an idle instance.
SQL> █
```

----------***-----***-----

' /u01/app/oracle/product/11.2.0/dbHome/bin ' qui comprend un grand nombre de commandes sysdba, pour l'administration de la BD oracle)

Trois types de fichiers de gestion de la base de données sous oracle : données, contrôle et log.

- v\$controlfile :
- v\$logfile :
- v\$datafile .

v\$database

----------***-----***-----***-----***-----***-----***-----


```
SQL> select member from v$logfile;
```

MEMBER

```
-----  
/u01/app/oracle/oradata/orcl/redo03.log  
/u01/app/oracle/oradata/orcl/redo02.log  
/u01/app/oracle/oradata/orcl/redo01.log
```

```
SQL> desc v$controlfile;
```

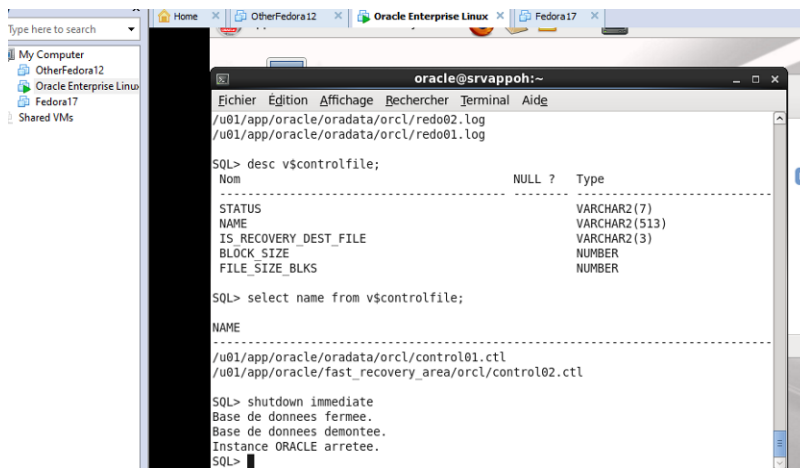
Nom	NULL ?	Type
STATUS		VARCHAR2(7)
NAME		VARCHAR2(513)
IS_RECOVERY_DEST_FILE		VARCHAR2(3)
BLOCK_SIZE		NUMBER
FILE_SIZE_BKLS		NUMBER

```
SQL> select name from v$controlfile;
```

NAME

```
-----  
/u01/app/oracle/oradata/orcl/control01.ctl  
/u01/app/oracle/fast_recovery_area/orcl/control02.ctl
```

```
SQL>
```



----------***-----***-----

```

oracle@srvappoh:~
Fichier Édition Affichage Rechercher Terminal Aide
FILE_SIZE_BLKs          NUMBER

SQL> select name from v$controlfile;

NAME
-----
/u01/app/oracle/oradata/orcl/control01.ctl
/u01/app/oracle/fast_recovery_area/orcl/control02.ctl

SQL> shutdown immediate
Base de donnees fermee.
Base de donnees demontee.
Instance ORACLE arreteee.
SQL> exit
Deconnecte de Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - Product
ion
With the Partitioning, OLAP, Data Mining and Real Application Testing options
[oracle@srvappoh ~]$ ls
afiedt.buf  MCD2013.SQL  Images  Modèles  Téléchargements
appohh.gz  dd           install_procedure  Musique  Vidéos
Bureau     Documents    MCD2013.SQL      Public
[oracle@srvappoh ~]$ pwd
/home/oracle
[oracle@srvappoh ~]$

```

Créer la base de données dans oracle à partir de la génération du script de cette base.

```

oracle@srvappoh:~
Fichier Édition Affichage Rechercher Terminal Aide
Base de donnees montee.
Base de donnees ouverte.
SQL> desc dict
  Nom                                NULL ?    Type
-----
TABLE_NAME                          VARCHAR2(30)
COMMENTS                            VARCHAR2(4000)

SQL> @'/home/oracle/bureau/essai/MCD2015.sql';
SP2-0310: impossible d'ouvrir le fichier "/home/oracle/bureau/essai/MCD2015.sql"
SQL> @'/home/oracle/MCD2013.SQL';

Table supprimee.

Table supprimee.

CREATE DATABASE MCD1
*
ERREUR a la ligne 1 :
ORA-01501: echec de CREATE DATABASE
ORA-01100: base de donnees deja montee

Table creee.

Commentaire cree.

Index cree.

```

Remarque

Tout modèle de données obéit à trois composantes principales (du point de vue SGBD) :

- *Définition* ;

- Manipulation ;
- Contrôle.

Aperçu de la constitution d'une BD. Une des tâches essentielles du développement des BD est la conception des schémas de base. L'objectif est de structurer le domaine d'application de sorte à le représenter sous forme de type ou de table. La représentation doit être juste pour éviter les erreurs sémantiques.

Elle doit être aussi complète pour permettre le développement des programmes d'applications souhaitées.

Elle doit être évolutive (sécable) afin de supporter la prise en compte rapide de nouvelles demandes.

Le concepteur ou l'administration de la base effectuent doivent répondre à ces choix.

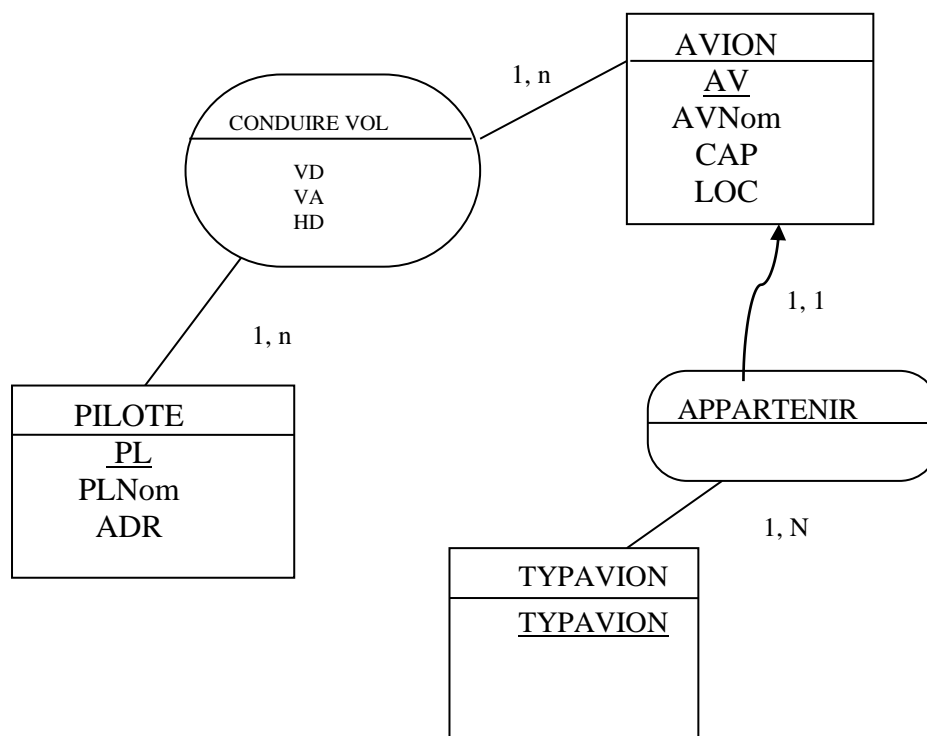
PILOTE (PL # INT (3), PLNOM CHAR (12), ADR CHAR (12) PRIMARY KEY (PL #))

AVION (AV # INT (3), AVNOM CHAR (12), CAP INT (3), LOC CHAR (20), PRIMARY KEY (AV #))

VOL (VOL CHAR (5) PRIMARY KEY, PL # INT (3) REFERENCE PILOTE (PL #), AV # INT (3) REFERENCE AVION (AV #), VD CHAR (20)

Not null, VA CHAR (20) HD DECIMAL (2, 2))

FORMATION (PL # NUMBER (4) not null REFERENCES PILOTE (PL #), TYPAV CHAR (2) not null, DATEF DATE, PRIMARYKEY (PL #, TYPAV).



REQUETES ET REPONSES:

Q1: Quels sont les types d'avion de la Compagnie avec leur capacité, tire par ordre croissant de capacité pour les capacités > 250.

Remarque : Tous les avions de même nom ont la même capacité.

R1 (Solution de Q1) : SELECT DISTINCT AVNom, cap FROM AVION WHERE cap > 250 ORDER BY cap DESC;

NB: Les termes connus du langage sont des clauses, les autres sont des prédicats.

Q2 : insérer la ligne (110,A320, 320, Nice)

R2: Insert into AVION values (110, 'A320', 'NICE');

Q3: Augmenter de 10% la capacité des avions localisés à Nice.

R3: Update AVION set cap = 11, 1 * cap where Loc = 'Nice';

Q4: supprimer les vols de NICE au départ après 20h.

R4: Delete * from Vol where VD = 'Nice' and HD>20

Q5: Pour chaque avion en service donnez son n°, type et sa capacité.

R5: Select AV, AVNom, cap from Avion, Vol
where vol.AV = AVION.AV

Q6: Quel est le nombre d'avion de la Cie?

R6: Select count (AV) from Avion;

Q7: Quel est le nombre d'avion en service?

R7: Select Count (AV) from VOL

Q8: Pour chaque avion en service donner le nombre des pilotes différent qui le conduisent.

R8: Select VOL.AV, count (Distinct VOL.PL) from Vol group by #AV

VOL1	PL#	AV#	VD
	1	100	Nice
	1	101	Paris
	1	102	Toul
	2	100	Paris
	2	105	Nice

Vol2	PL#	AV#	VD
		100	Nice
		101	Paris
		102	Toulouse

VOL3	PL#	#	VD
		100	Paris
		105	Nice

Q9: Quels sont les numéros des pilotes qui assurent plus 2 vols?

R9: select PL from VOL group by PL Having Cunt (*) >2;

Q10: Quels sont les n° de pilote (avec le nombre des vols assurés) qui assurent plus de 2 vols au départ de Nice ?

R10: Select PL, count (*) From VOL where PL in (Select PL From Vol where VD = 'Nice' Group By Having Count (*) >2;

2e Partie Méthode ensembliste et Méthode prédictive

Q1: Quels sont les noms des avions avec leur n° et leur localisation?

• **Méthode ensembliste** : un ordre est caractérisé par

- un **seul SFW par table**
- le 1^{er} SFW traduit le résultat
- association des SFW par l'appartenance ensembliste

• **Méthode prédicative** : un ordre est caractérisé par :

- un **seul SFW pour toute la question**
- la liste de tables intervenant dans l'écriture de la requête est dans FRM
- la traduction de la jointure par une égalité entre les colonnes

Q2 : Quels sont les noms des pilotes qui assurent un vol au départ de paris

Q3 : Quels sont les noms des pilotes qui conduisent un Airbus ?

S2:

- (méthode prédicative)

Select PLNom from pilote, vol where pilote.PL = vl.PL and vol.VD = 'Paris'

-(méthode ensembliste)

Select PLNom from pilote where PL# = any (Select PL# from vol where VDD = <<Paris>>);

S3: Select PLNom from pilote where PL in (Select PL from vol where AV in (Select AV from Avion where AVNom = 'Airbus'))

Q4: Quels sont les noms des pilotes qui habitent dans les localisations d'un 'Airbus'

Extension de SQL 2 (relationnel) est SQL 3 et SQL4 (objet) mais qui comprennent le relationnel.

Exercices /projets

- Base de données dégustation
- BD état civil
- BD ligne budgétaire
- BD météorologique
- Gestion du personnel,
- Gestion du patrimoine
- Organisation de concours
- Gestion de suivi des armes de guerre
- Gestion de passation des charges
- gestion d'une chaîne de complexes hôteliers
- suivi des sources et origines des armes de guerres
- gestion des zones de pèlerinage (basilique)
- gestion des pannes et réparations d'un parc automobile d'une entreprise

Autres :

A-Gestion d'une bibliothèque (trois tours de 10 étages dont 5 en sous-sol et 5 en hauteur)

B- Gestion des bus d'une métropole (A chaque arrêt de bus se trouvent notifiées chaque fois les heures d'arrivée des deux prochains bus A et B. A et B arriveront respectivement à t1 et t2 ($t_1 < t_2$). Faire en sorte que maximalement les arrêts aller et retour soient en face à face. Prendre en compte les dépôts ou stations puis les centres de formations de pilotes/chauffeurs et de réparation).

C- Gestion de moyens de locomotions communes (vélos, voitures à quatre place). Tout usager peut se servir d'un véhicule pendant une certaine durée, et après usage le stationner à des dépôts dédiés.

1-Trace des armes de conflit

2-Gestion d'un schéma directeur de développement à continuité et à l'échelle (projet régional, planification, court moyen long terme, budget, secteur de soutien, santé, éducation, emploi, ministère, Institutions de base...)

3-Passation des charges d'administration de responsabilité tant publique que privée

4-Gestion rationnelle à la proportionnelle utile de la reprise des travaux de mémoire d'ingéniorat (école, pays, niveaux de résultats, auteur, type d'innovation, ...)

5-Gestion d'une chaîne de complexes hôteliers au plan national

6-Gestion de retrait et de dépôt d'argent (GAB et mobile money à la fois) VISA. Admission des banques et de mobile money transfert en visa.

7-Gestion des vols et plans de vols

****_**

8-Gestion des explorations minières des régions au plan national

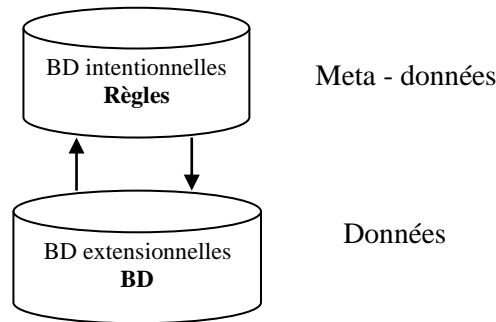
9-Gestion innovante de passation des marchés publics :

Règles de gestion MCD pour le modèle MCD

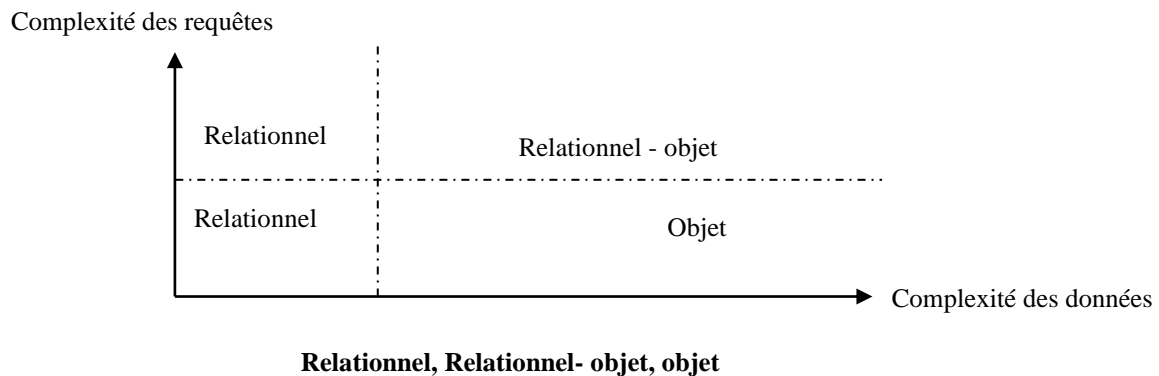
Base de données déductives :

Prédicat intentionnel : prédicat calculé par un programme constitué de règles logiques dont les instances ne sont pas stockées dans la base de données.

Prédicat extensionnel : prédicat dont les instances sont stockées dans la base de données sous forme de tuples.



Prédicats extensionnels et prédicats intentionnels



Technologie de l'information :

Technologie : ensemble des techniques industrielles, science des techniques ou technique.

Technique (adj) : propre à un art, à une science à une activité, à un savoir faire, à un fonctionnement

La technique : la technologie, ensemble de précédés d'un art ou d'un métier

Evaluation de l'objectivité d'une base de données à objet par RICE

Extension de java (Java futur : **AspectJ**)

Disque :

IDE : Integrated Device Electronics

S-ATA Serial Advanced Technology Attachment S-DATA

SCSI : Small Computer System Interface? (Système d'interface pour micro-ordinateur).

Interface parallèle standard à débit rapide, définie par le comité X3T9.2 de l'American National Standard Institute (ANSI). Enfin, certaines interfaces peuvent être assimilées à de véritables bus, permettant le raccordement de plusieurs périphériques à la suite : c'est le cas de l'interface SCSI ou du futur bus universel USB (Universal Serial Bus).

Processeur :

CISC : Complex Instruction Set Computing, processeur à jeu d'instructions complexes)

RISC: Reduced Instruction Set Computing, processeur à jeu d'instructions réduit)