

Initiation aux bases de données

Chapitre 1 : Les concepts fondamentaux

1. Objectifs

- ✚ Les principes fondamentaux liés à la conception et à l'utilisation des bases de données.
- ✚ La manière dont les systèmes de gestion de bases de données ont évolué pour permettre une gestion efficace et cohérente des données.
- ✚ La manière dont les organisations conçoivent les bases de données et en assurent l'exploitation.
- ✚ Pourquoi est-il si important pour les organisations d'élaborer des modèles de données à divers niveaux d'abstraction ?

Qu'on le veuille ou non, les bases de données tout comme les technologies de l'information de manière générale sont omniprésentes dans les diverses activités de l'Homme moderne. Nous débutons ce chapitre en évoquant un certain nombre d'applications des bases de données non seulement pour illustrer leur importance centrale dans le fonctionnement de la société et des organisations mais aussi pour montrer les soins particuliers que leurs concepteurs doivent appliquer lors de leur planification et de leur réalisation afin de servir efficacement les utilisateurs.

Il est hors de question de faire ici un inventaire exhaustif des domaines d'application des bases de données. Il nous importe cependant de mentionner des exemples particulièrement significatifs d'applications conçues pour servir tant les individus que les organisations. Avant de les aborder, nous introduisons trois concepts pour lesquels nous proposons les définitions qui suivent. Ces concepts sont sous-jacents à l'illustration que nous comptons faire dans cette section.

Base de données (BD) : Ensemble structuré d'éléments **d'information**, souvent agencés sous forme de **tables**, dans lesquels les **données** sont organisées selon certains critères en vue de permettre leur exploitation pour répondre aux besoins d'information d'une organisation (*Database*).

En d'autre terme : Une base de données est un mécanisme qui permet de stocker des données de façon organisée et de les retrouver facilement suivant certains critères de recherche à condition qu'elle soit bien organisée. Dans une base de données il est possible d'ajouter, de modifier et de supprimer des données. Une base de données est dite existante quand elle est utilisée par une entreprise depuis plusieurs années.

Exemple : Une institution scolaire pourrait par exemple exploiter une seule base de données permettant de gérer l'admission des étudiants, d'assurer l'offre de cours à chaque session, d'inscrire les étudiants, de percevoir les frais d'inscription, de compiler les résultats et d'émettre les bulletins de notes.

Certaines de ces bases de données sont statiques, d'autres dynamiques.

Exemple :

- ✚ Une *carte routière* est une base de données statique (au sens abstrait), elle contient des informations telles que des états, des villes, des autoroutes, des directions, des distances, etc. En consultant une carte routière, nous pouvons rapidement tracer notre destination par rapport à notre situation actuelle. Les informations d'une carte ne changent pas. Il est impossible de les modifier sans imprimer et diffuser une nouvelle carte.
- ✚ Un *annuaire téléphonique* est également une base de données statique car les informations personnelles et commerciales y figurent pour une année donnée. Il est également impossible de les modifier sans imprimer et diffuser un nouvel annuaire.
- ✚ Le *cahier d'adresses* représente un bon exemple de base de données dynamique, les entrées sont modifiées à mesure que les amis et les membres de la famille déménagent ou changent de numéro de téléphone. On peut ajouter des personnes à l'occasion de nouvelles rencontres et en supprimer.
- ✚ Une *librairie en ligne* est également une base de données dynamique par le fait qu'elle traite sans cesse des commandes de livres, qu'elle ajoute régulièrement de nouveaux auteurs et titres, qu'elle en supprime, qu'elle met à jour l'inventaire, etc.

Application de bases de données : Utilisation de moyens informatiques pour répondre à un besoin déterminé en faisant appel de manière importante à une ou plusieurs bases de données à travers un **système de gestion de bases de données (SGBD) (Database application)**.

Exemple : Pour poursuivre avec l'exemple de l'institution scolaire, une application de bases de données pourrait être élaborée uniquement pour la gestion du volet comptable des frais de scolarité. Cette application serait conçue notamment pour émettre les factures pour les frais de scolarité et autres frais afférents, pour percevoir les paiements, rembourser l'étudiant à la suite d'un abandon, suspendre une inscription pour défaut de paiement etc...

Système de gestion de bases de données (SGBD) : Logiciel, le plus souvent produit par un éditeur commercial, qui gère et contrôle l'accès à une base de données, assurant ainsi une interface normalisée entre les applications et les bases de données (*Database management system*).

Plusieurs SGBD sont des logiciels commerciaux coûteux offerts sous forme de licences d'utilisation sur un serveur par des éditeurs de logiciel tels que IBM (DB 2), Oracle, Microsoft (SQL Server) pour ne nommer que les plus importants qui proposent des SGBD conçus pour exploiter les bases de données de grande envergure. Certains éditeurs offrent par ailleurs des SGBD bas de gamme, peu coûteux, destinés aux applications de bases de données de petites envergures. Microsoft Access est peut être le mieux connu et le plus utilisé des SGBD de cette catégorie, souvent appelé SGBD bureautique. Il existe aussi des SGBD dans le monde du logiciel libre, le plus populaire est sans aucun doute MySQL

2. Applications des bases de données

2.1. Le commerce électronique

L'utilisation des bases de données a connu un essor considérable dans le contexte du développement des échanges commerciaux sous forme électronique. Les entreprises de commerce de détail offrent de plus en plus à leurs clients la possibilité de consulter leur catalogue de produits par le biais de l'Internet de manière à diffuser le prix et la disponibilité de leurs produits et permettre à ces derniers, le cas échéant, de procéder à un achat en ligne. L'accès au catalogue, la possibilité de compléter une transaction d'achat avec autorisation de paiement par carte de crédit ne sauraient être mis en œuvre sans l'utilisation de plusieurs bases de données gérées soit par l'entreprise, soit par une institution financière partenaire. Lorsque le niveau de stock pour un produit atteint un seuil de rupture, le système informatique du commerçant peut émettre sur-le-champ une commande auprès d'un fournisseur par voie électronique. Là encore, une base de données du côté fournisseur va permettre de recevoir et de donner suite à la commande en confirmant une date de livraison, et en procédant ensuite à la facturation le moment venu.

Certaines entreprises offrent aux consommateurs la possibilité de faire des échanges commerciaux entre eux selon la formule d'une enchère électronique. La mise en vente ou la gestion des offres en temps réel ne saurait être possible sans une application de base de données sophistiquée qui assure l'impartialité du processus.

2.2. Les affaires électroniques

Les affaires électroniques précèdent et prolongent les échanges purement transactionnels liés au commerce électronique pour l'achat, la vente et le paiement des biens et services. Leur domaine d'application est plus large que le commerce électronique. Il concerne aussi bien l'organisation du travail dans une organisation que sa façon de communiquer et d'échanger des données avec ses clients, ses sous-traitants, ses fournisseurs et ses partenaires.

Les institutions financières ont été particulièrement innovatrices sur ce plan mais leurs initiatives n'auront été possibles qu'en adoptant des solutions exploitant de larges bases de données. L'introduction des guichets automatiques par les banques coïncide avec l'adoption de nouvelles technologies de communication d'une part, et de *systèmes de gestion de larges bases de données distribuées* d'autre part qui permettent au client d'effectuer des transactions sur ses divers comptes bancaires sans égard au propriétaire et à la localisation du guichet utilisé.

Les mêmes bases de données, accessibles traditionnellement via un guichet automatique, le sont aussi par Internet ou par des services téléphoniques automatisés.

La plupart des sociétés d'assurance offrent à la clientèle la possibilité d'obtenir en ligne une proposition pour une police d'assurance de dommages sur leurs biens. Elles mettent alors en œuvre des applications de bases de données qui dans un premier temps recueillent auprès du client des données sur la nature des biens et des couvertures souhaitées pour ces derniers. En accédant à des données provenant de sources diverses, tels que des bases de données sur la tarification ou l'historique des réclamations du client, l'application produit sur le champ une soumission et, si le client accepte la proposition, une police d'assurance sera émise.

Les institutions publiques ne sont pas en reste sur ce plan. Le citoyen peut remplir sa déclaration d'impôts. Des initiatives dites de gouvernement en ligne devraient permettre à terme au citoyen de traiter avec le gouvernement grâce à un seul guichet électronique. Ces initiatives ne sont possibles que par la mise en œuvre d'échanges de données entre les centaines de milliers de bases de données administrées par les multiples agences, organismes, sociétés du secteur public.

2.3. La gestion électronique des documents

Ce domaine d'application des bases de données concerne la gestion, par des moyens informatiques, du cycle de vie complet d'un document électronique, qu'il soit de nature textuelle, graphique, sonore, vidéo ou logicielle. Ce cycle va de sa création à sa destruction, en passant par sa modification, sa publication, sa diffusion. Cela en vue d'optimiser l'accès à ce document, à l'information qu'il contient ainsi qu'à d'autres documents apparentés.

Dans un tel contexte, les bases de données assurent le stockage et la diffusion de données multimédia, souvent qualifiées de données non structurées. La gestion électronique des documents a mené au développement de SGBD objet. Ce type de système de gestion de bases de données se distingue des SGBD relationnels utilisés traditionnellement par les organisations pour le stockage de données structurées sous forme de tables. Il sera question un peu plus loin des caractéristiques de ces divers types de SGBD.

Les bases de données dites objet sont centrales au fonctionnement des organisations qui œuvrent dans le secteur des communications, notamment les entreprises du monde de l'édition, de la production audio-visuelle et de la diffusion. La plupart des chaînes de télévision offrent à leurs auditeurs la consultation, à partir de leur portail Internet, d'un large éventail de documents électroniques. On peut y retrouver notamment la copie textuelle de la retranscription d'une émission, des extraits audio ou vidéo de la même émission ou encore l'émission dans sa version originale intégrale. Tout cela est rendu possible efficacement par la mise en œuvre de bases de données multimédia gérées par un SGBD objet ou de type hybride communément appelé SGBD relationnel-objet.

2.4. Le support à la décision

La décennie nonante a vu apparaître une catégorie de systèmes d'information permettant la recherche active et l'exploitation, sur le plan décisionnel, de l'ensemble des renseignements stratégiques essentiels qu'une entreprise doit posséder si elle veut faire face à la concurrence et occuper la première place dans son secteur d'activité. Ce type d'application appelé *système de veille stratégique* (connu en anglais sous le vocable de *business intelligence system*) fait appel à une masse considérable de données provenant de sources multiples, recueillies sur une large échelle de temps, regroupées dans une base de données que l'on appelle *entrepôt de données*.

Entrepôt de données : base de données spécialisée dans laquelle est centralisé un volume important de données consolidées à partir des différentes sources de renseignement d'une entreprise (notamment les bases de données internes) et qui est conçue de manière à ce que les personnes intéressées aient accès rapidement à l'information stratégique dont elles ont besoin (*Data warehouse*).

Exemple : imaginons qu'un important brasseur recueille quotidiennement, à partir de milliers de points de vente, des données sur la vente de ses produits et ceux de ses concurrents. Ces données sont consolidées dans un entrepôt de données qui contient de plus une foule de données démographiques

et socioéconomiques provenant d'un bureau de recensement. Les analyses menées quotidiennement sur ces données à l'aide de modèles mathématiques sophistiqués permettent au brasseur de cibler la clientèle pour un nouveau produit dans une région donnée, d'établir le moment opportun pour lancer une campagne de promotion sur certains produits, de proposer aux détaillants une présentation de leurs produits favorisant les achats croisés. Cette forme d'exploitation des bases de données est en pleine expansion non seulement dans le secteur du commerce au détail mais aussi dans les secteurs industriels et financiers.

3. Notions fondamentales en matière de gestion de données

Nous allons aborder dans cette section certains concepts fondamentaux du domaine de la gestion de données. Il importe notamment de définir les termes du vocabulaire courant qui dans le contexte du traitement électronique des données prennent une signification précise.

3.1. Donnée et information

Donnée : représentation d'un élément d'information, tel qu'un chiffre ou un fait, codé dans un format permettant son stockage et son traitement par ordinateur (Data).

Exemple : lorsqu'un client se présente à un point de vente pour payer un produit, la lecture du code barre donne accès à des données liées à ce produit, dont son prix et son nom. Dans ce cas, le code barre est une donnée au sens défini ci-dessus puisqu'il s'agit de la représentation codée d'une suite de chiffres, le code universel du produit (CUP). Par ailleurs, le prix et le nom du produit auxquels correspond le CUP sont aussi des données qui sont codées dans un format faisant appel à une norme largement adoptée en matière de codage binaire des données pour leur traitement par des ordinateurs. Cette norme porte le nom ASCII (*American Standard Code for Information Interchange*). Le format de codage d'une donnée dépend de son type.

Type de donnée : nature du codage utilisé pour représenter une donnée élémentaire et les opérations applicables à cette donnée. Les types les plus courants sont : entier, réel, texte, date, image, etc. (*Data type*).

Une donnée est une représentation. C'est pourquoi dans sa définition on réfère à la codification, donc à la syntaxe. Une information est davantage une interprétation de données dans un contexte particulier. Sa définition fait plutôt référence à la signification d'une donnée mais plus généralement d'un ensemble de données, c'est-à-dire à la sémantique, comme le montre la définition suivante.

Information : une information est une donnée ou un ensemble de données qui a ou ont été interprétée(s) (*Information*).

Les deux concepts sont intimement liés. Dans l'exemple précédent nous référions aux données d'une transaction de vente. La transaction, en plus du nom du produit, de son prix et de son CUP (code universel du produit), pourrait comporter le numéro du magasin où l'achat a été effectué. Si toutes les transactions de vente d'une grande chaîne de magasins sont enregistrées dans une même base de données, il sera possible en faisant un traitement statistique sur ces données de savoir quels produits se vendent le mieux dans les divers magasins de la chaîne. Le résultat de ce traitement et l'interprétation qu'en donneront les dirigeants représentent de l'information. Cette information pourrait mener à des décisions, notamment de retirer de ses tablettes certains produits dans certains magasins pour faire place à des produits qui se vendent mieux. Nous dirions en conclusion que

l'information induit assez souvent une action, soit une décision dans l'exemple qui nous occupe. Une masse de données, à moins qu'elle ne subisse un traitement approprié et qu'on en retire de l'information, induit rarement une action. Par analogie nous pourrions conclure que les données constituent la matière brute à partir de laquelle l'information est produite.

La vocation première d'une base de données est le stockage des données. Il s'agit donc d'assurer par le biais de cette technologie la persistance des données dans le temps. Les systèmes informatisés qui sont mis en œuvre pour le traitement des données peuvent être de simples outils de cueillette de données à la source. On les appelle *systèmes de traitement de transactions* (STT). De plus, ils peuvent être des systèmes qui traitent les données recueillies par les STT pour produire des informations utiles à la prise de décision. Cette deuxième catégorie de système fait appel tant à des ressources informatiques (équipement, logiciel, données) qu'à d'autres types de ressources telles que du personnel ou des procédures administratives. De tels systèmes portent le nom de système d'information, ou encore de système d'information organisationnel lorsque le système est conçu pour répondre aux besoins en matière d'information pour l'organisation dans son ensemble.

Système d'information organisationnel : système constitué des ressources humaines, des ressources informatiques (équipement, logiciel, données) et des procédures permettant d'acquérir, de stocker, de traiter et de diffuser les éléments d'information pertinents au fonctionnement d'une entreprise ou d'une organisation (*Management information system*).

Les systèmes d'information modernes exploitent tous une ou plusieurs larges bases de données à l'aide d'un ou de plusieurs systèmes de gestion de base de données. Les données stockées dans ces bases de données assurent le lien entre d'une part les ressources non informatiques que sont les utilisateurs et les procédures qu'ils doivent appliquer à l'intérieur du système et d'autre part les ressources informatiques. Les données agissent à la manière d'un pont qui relie les ressources informatiques et non informatiques d'un système d'information.

3.2. Caractéristiques des systèmes de gestion de bases de données (SGBD)

On s'entend généralement dire qu'un logiciel de gestion de données, pour qu'il puisse porter le nom de système de gestion de bases de données, doit posséder un certain nombre de caractéristiques fondamentales. Nous les regroupons ici en six catégories.

3.2.1. Indépendance entre les données et les applications

Les premières applications informatiques exploitant de grandes quantités de données étaient basées sur *des fichiers*. Les programmes informatiques devaient dans ce contexte comporter une description détaillée des données stockées dans ces fichiers, par exemple le type d'une donnée, sa taille ou son format. Les applications étaient alors fortement dépendantes de la structure des données des divers fichiers qu'elles utilisaient. Un SGBD doit permettre à un programmeur de développer une application sans avoir à encoder dans les programmes les aspects structurels des fichiers d'une base de données. De la sorte, si la structure de la base de données devait être changée, les programmes n'auront pas à être modifiés et, si cela était nécessaire, les modifications ne seraient que mineures. Cette caractéristique des SGBD permet une très grande productivité du personnel qui réalise des applications de base de données.

3.2.2. Contrôle centralisé des données pour éviter toute redondance

Le SGBD doit intégrer dans un même espace de stockage plusieurs fichiers, de manière à éviter toute redondance, c'est-à-dire courir le risque qu'une même donnée se retrouve dans deux ou plusieurs fichiers. Il n'est pas souhaitable qu'une donnée, par exemple l'adresse d'un client, se retrouve dupliquée dans plusieurs fichiers. Une telle situation pose des problèmes au moment où la donnée doit être mise à jour. Si l'adresse d'un client devait changer, il y a un risque que le changement ne soit enregistré que dans un seul fichier. Comment savoir alors quelle est l'adresse la plus récente, celle conservée dans le fichier A ou dans le fichier B ? Même si le changement d'adresse était enregistré dans tous les fichiers où elle est stockée, cela pose un problème au plan de la performance : pourquoi faire plusieurs fois ce qui logiquement ne devrait être fait qu'une seule fois ?

3.2.3. Partage des données et accès concurrents

Le besoin de centraliser le stockage des données de manière à assurer l'unicité de la saisie et de la validation des données a pour corollaire la mise en place de mécanismes techniques qui vont assurer aux utilisateurs un partage équitable de ces données, sans risque de concurrence, mais qui vont aussi empêcher les accès non autorisés. Le SGBD doit posséder un mécanisme d'accès concurrents à une base de données par plusieurs utilisateurs autorisés. Ce mécanisme doit éviter que des opérations simultanées mènent à des incohérences. Considérons par exemple deux utilisateurs qui tentent de modifier le solde d'un compte client. Le SGBD doit identifier, sur la base du principe « premier arrivé premier servi », quel utilisateur aura la priorité pour faire la lecture du solde. Il permettra au second utilisateur la lecture du solde quand le premier utilisateur aura complété sa modification. Ce mécanisme porte le nom de verrou. Il interdit tout accès à une donnée ou à un ensemble de données tant qu'un autre utilisateur procède à leur traitement.

3.2.4. Gestion de la cohérence et de l'intégrité des données

Les SGBD doivent pouvoir gérer le processus de mise à jour d'un ensemble de données comme un tout indissociable pour garantir la cohérence interne de la base de données. Dans le contexte de la gestion des comptes clients, supposons que l'application doive conserver dans la base de données à la fois les données de la transaction affectant le solde du client ainsi que la donnée sur le nouveau solde lui-même. Il est hors de question que, par suite d'une panne de l'ordinateur, les données de la transaction aient été enregistrées mais que le solde n'ait pu être recalculé. Ou l'inverse, que le nouveau solde ait été stocké mais que les données de la transaction n'aient pu être conservées. Un SGBD doit offrir un mécanisme qui fait en sorte que, si une étape d'une opération ne peut être complétée, toute l'opération doit être annulée automatiquement. Il est du ressort du programmeur de l'application de déterminer quelles sont les étapes indissociables d'une même opération et d'appliquer le mécanisme approprié. Ce mécanisme assure l'atomicité d'une transaction. C'est-à-dire qu'à la manière d'un atome, toutes les composantes d'une opération, ou d'une transaction, doivent être considérées comme un tout indissociable. Tout comme l'application d'un verrou sur une donnée, le mécanisme est totalement transparent à l'utilisateur car il est mis en œuvre par le programmeur. L'intégrité des données réfère à la nécessité de voir appliquer automatiquement par le SGBD des contraintes sur la validité d'une donnée. Là encore, il s'agit d'une responsabilité qui incombe aux concepteurs d'une base de données

Contrainte d'intégrité des données : ensemble de règles, définies par le concepteur d'une base de données, qui devront en tout temps être respectées par les données de la base de données. Ces règles sont gérées par le SGBD qui en assure l'application et informe l'utilisateur lorsque l'une d'elles est transgressée (Integrity constraint).

Des contraintes d'intégrité des données sont définies au moment de la conception d'une base de données afin d'écarter, dès leur saisie, les valeurs jugées inacceptables. Ces contraintes permettent donc de mettre en œuvre une validation automatique des données. Le concepteur d'une base de données peut non seulement établir pour une donnée son type ou sa taille, mais fixer par ailleurs une plage de valeurs acceptables, une limite inférieure, ou enfin une limite supérieure. Ces exemples de contraintes sont de type de contrainte d'intégrité sémantique car elles concernent la signification d'une donnée, donc l'interprétation unique et non ambiguë qui doit être accordée à cette donnée. Il existe un autre type de contrainte dite contrainte d'intégrité référentielle qui concerne les valeurs que peuvent prendre deux données associées dans la même base de données. Les SGBD relationnels, comme nous le verrons plus loin, peuvent assurer la satisfaction de contraintes d'intégrité référentielle en faisant en sorte que la valeur prise par une donnée placée dans un champ d'une première table doit être obligatoirement une des valeurs prises par un des champs d'une deuxième table.

3.2.5. Description des données stockées sous forme de métadonnées

La structure d'une base de données, telle que définie spécifiquement par le concepteur pour être prise en charge par un SGBD, est appelée le schéma physique de la base de données (MPD).

Schéma physique d'une BD : collection des composants constitutifs de la structure d'une BD, notamment les propriétés des données, les domaines des données (types de données), les fichiers, les contraintes d'intégrité, associés d'une manière ou d'une autre les uns aux autres. Ce schéma est défini par le biais d'un langage de définition de données faisant partie intégrante du SGBD (Database physical schema).

Le schéma physique est conservé dans la base de données sous forme de métadonnées au même titre que les données elles-mêmes. C'est précisément la nature réflexive ou auto descriptive des SGBD qui assure l'indépendance entre les données et les applications dont nous traitons plus haut. Un SGBD doit pouvoir en effet gérer non seulement des données mais aussi des métadonnées.

Métadonnées : données à propos des données qui sont stockées dans une BD (Metadata).

3.2.6. Gestion de la sécurité

La sécurité dans les SGBD possède plusieurs facettes. La première concerne les privilèges d'accès aux données accordés aux utilisateurs. Un SGBD doit permettre d'accorder des privilèges aux utilisateurs jusqu'au niveau d'une donnée pour des opérations de lecture, d'insertion, de mise à jour ou de suppression. Toutefois, en pratique, les privilèges sont généralement accordés pour un groupe d'utilisateurs, à un ensemble de données, soit en lecture (accès), soit en écriture (ajout et modification) c'est-à-dire pour les deux modes d'accès principaux aux enregistrements. Ces privilèges sont accordés par la personne à laquelle l'organisation a confié le rôle d'administrateur de bases de données. Les privilèges sont conservés sous forme de métadonnées dans la BD et sont rattachés au code d'accès et au mot de passe que l'administrateur a attribué à l'utilisateur.

Administrateur de base de données (ADB) : personne responsable de la réalisation physique de la BD sous forme de schéma physique, du contrôle de la sécurité et de l'intégrité, de la maintenance du SGBD et de la garantie de performance des applications de bases de données qui doit être assurée aux utilisateurs (Database administrator).

Il est aussi de la responsabilité de l'administrateur de bases de données de déterminer quelles seront les données qui devront être chiffrées, c'est dire rendues illisibles à ceux qui se seraient procurés frauduleusement des codes d'accès et pourraient en faire un usage inapproprié. Il va de soi que toutes les métadonnées relatives aux droits d'accès, aux codes d'utilisateur, pour ne mentionner que ceux-là, doivent être stockés chiffrés dans la BD.

Un deuxième volet de la sécurité concerne la sauvegarde de copies de sécurité d'une BD et la récupération des données à la suite d'une panne. Certains SGBD maintiennent en tout temps une copie miroir d'une BD et ce à la volée. C'est à dire que tout changement à une BD est reflété en temps réel dans une deuxième BD qui maintient une copie de la première souvent sur un site physiquement distant. La copie miroir peut par ailleurs être générée en différé par un mécanisme appelé synchronisation. Il s'agit d'alimenter, en temps opportun, une BD miroir incluant tous les changements effectués depuis la dernière synchronisation, par exemple au cours de la nuit quand les opérations sont suspendues.

Enfin, un SGBD doit pouvoir conserver une trace de toutes les opérations effectuées sur une BD, sous forme d'un journal de transactions. Ce journal complété automatiquement, donc sans intervention humaine, va permettre de reprendre le cas échéant certaines transactions qui n'ont pu être complétées à cause d'une panne, et ce, en toute transparence pour les utilisateurs. Le même journal est aussi un outil de vérification et de contrôle pour l'administrateur de la base de données.

3.3. Origine et évolution des SGBD

3.3.1. Les systèmes basés sur des fichiers

On ne saurait parler de l'origine des SGBD sans évoquer les premiers systèmes de traitement électronique des données basés sur des fichiers. Cela pour trois raisons principales :

- ✚ Ces systèmes ont permis de définir certains concepts toujours en usage même dans le contexte des SGBD les plus évolués, les notions de fichier de données, d'enregistrement, de champ et de type de données
- ✚ Bien que cette approche soit maintenant obsolète, il existe encore de nombreux systèmes de ce genre dans les organisations qu'il serait trop coûteux de remplacer par des applications de base de données. On les appelle systèmes hérités (ou en anglais legacy systems) car ils nous sont légués par des technologies de première génération, dépassées bien sûr, mais qu'il faut continuer d'utiliser tant que de nouveaux systèmes en cours de développement, ou prévus à l'intérieur d'un plan directeur de développement, ne les auront pas définitivement remplacées.
- ✚ Les systèmes basés sur des fichiers comportaient de nombreuses lacunes qui ont permis d'établir de manière empirique les caractéristiques fondamentales des SGBD modernes dont nous avons fait état précédemment.

Système basé sur des fichiers : ensemble de programmes d'application qui exploitent ses propres fichiers de données pour répondre aux besoins spécifiques d'un groupe d'utilisateurs finaux (File_based system).

Les premiers grands systèmes de traitement électronique des données élaborés dans les années 1950 et 1960, tels que les systèmes de traitement de la paie, les systèmes de gestion financière et comptable ou les systèmes de gestion des ressources humaines, étaient des systèmes basés sur des fichiers. Ces systèmes étaient conçus pour les besoins spécifiques d'une unité administrative et par conséquent les programmes d'application et les fichiers étaient généralement à l'usage exclusif du personnel de l'unité. Applications et fichiers formaient un ensemble fortement intégré dont l'exploitation était réservée à un groupe exclusif d'utilisateurs. Les concepteurs de ces applications faisaient dès lors appel à des technologies basées sur un certain nombre de concepts dont nous ferons ici état.

Les *fichiers* sont constitués d'enregistrements eux-mêmes décomposables en champs, chaque champ permettant le stockage d'une donnée. Ces concepts fondamentaux sont explicités à travers les définitions qui suivent.

Fichier de données : ensemble d'enregistrements identifié par un nom, qui constitue une unité logique de stockage de données pour un ordinateur (Data file).

Enregistrement : groupe de données apparentées, structuré et considéré comme un tout. Chaque donnée du groupe occupe un champ de l'enregistrement. Un champ est défini par son nom, une position dans l'enregistrement et le type de données qu'il permet de stocker (Record).

Un fichier est généralement représenté dans les ouvrages techniques sous forme d'un tableau comportant en en-tête une suite d'intitulés donnant le nom des champs, sous lequel sont inscrits les

enregistrements. La figure 0-1 illustre un petit fichier appelé Véhicule comportant trois enregistrements formés de cinq champs : No de série, Fabricant, Modèle, Année, Immatriculation. Chaque enregistrement possède une donnée par champ et, pour un champ en particulier, son type de données et sa taille sont systématiquement les mêmes quel que soit l'enregistrement. Dans cet exemple, le champ Année ne stocke que des valeurs numériques entières de quatre chiffres. Chaque ligne du tableau représente un enregistrement et tous les enregistrements ont la même structure, définie par les propriétés des champs qui le composent.

No de série	Fabricant	Modèle	Année	Immatriculation
YG100P9065QZ84	Ford	Taurus	2005	WWP 657
JK92876T6753W9	Nissan	Pathfinder XE	2004	KDF 324
PK8750927GH786	BMW	320 SI	2002	BGH 629

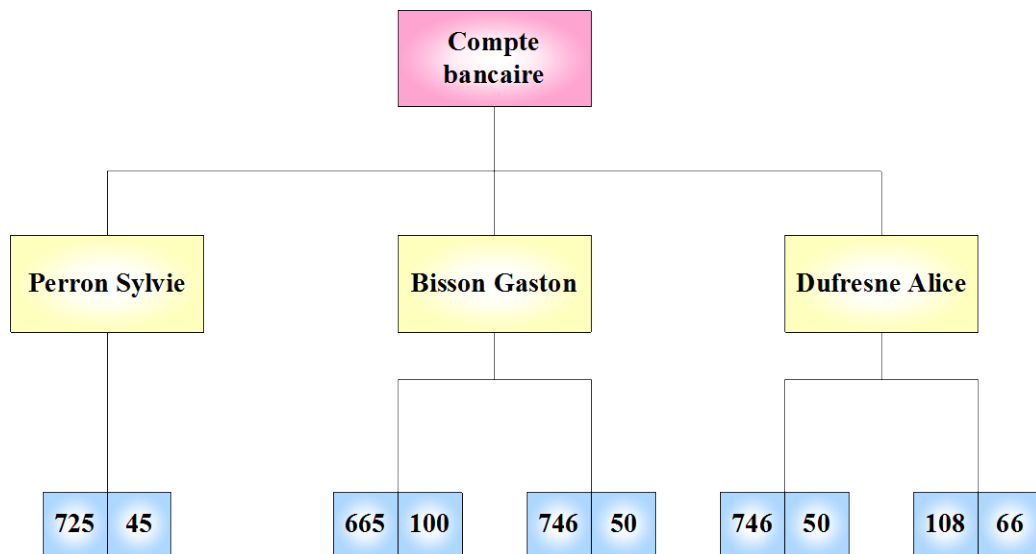
Figure 0-1 : fichier véhicule

3.3.2. Les bases de données hiérarchiques et réseaux

Les problèmes liés à l'exploitation des systèmes basés sur des fichiers sont notamment la forte intégration des applications et des fichiers, la redondance des données, la difficulté de partager les données et d'assurer leur intégrité. Enfin, la piètre sécurité des données ont donné lieu à un nouveau modèle en matière de gestion des données.

Une première génération de SGBD est née de travaux de recherche menés dans les années 1960 pour le compte de la NASA dans le cadre du projet Apollo et commercialisés dans les années 1970 par la société IBM sous le nom IMS (Information Management System). Le SGBD était basé sur le concept que les enregistrements conservés dans divers fichiers pouvaient être liés selon une certaine hiérarchie et constituer un assemblage arborescent. On a appelé *structure hiérarchique* cette forme de liaison entre des enregistrements de fichiers distincts. En vertu de ce modèle d'organisation des données, un enregistrement peut être le père de plusieurs enregistrements qui à leur tour peuvent avoir des fils. Par exemple, un fichier appelé Etudiant contenant des données sur les étudiants peut être lié à un autre fichier, Cours inscrits, contenant des données sur les cours suivis par les étudiants et le cas échéant le résultat obtenu. A l'aide d'un langage dit de navigation entre les fichiers, le programmeur accédant à un enregistrement du fichier Etudiant peut repérer automatiquement les enregistrements fils du fichier Cours inscrits. Un SGBD hiérarchique possède un avantage indéniable sur les systèmes basés sur des fichiers. Il permet en effet d'établir des liens un à plusieurs entre les fichiers d'une base de données : un enregistrement père peut être lié à plusieurs enregistrements fils.

Exemple de modèle hiérarchique



Dans une structure arborescente, les données sont représentées d'une façon hiérarchique comme un arbre considéré à l'envers. La lecture de cette structure se fait du haut vers le bas. Pour accéder à une donnée relative à un compte, il faut passer par la donnée client.

- Cette structure arborescente représente les clients d'une banque qui possèdent chacun un ou plusieurs comptes. Chaque compte peut en outre appartenir à un ou plusieurs clients. La cliente Sylvie Perron ne possède qu'un seul compte dont le numéro est 725. Le solde de ce compte est 45€. Le client Gaston Bisson possède deux comptes dont les numéros sont 665 et 746 et dont les soldes sont respectivement 100€ et 50€. La cliente Alice Dufresne possède aussi deux comptes, 746 et 108. Vous pouvez constater que les clients Bisson et Dufresne ont le même compte, le 746
- Dans cet exemple, nous n'envisageons que trois clients mais représentez-vous la structure composée de milliers de clients.
 - 1) Quels problèmes suscite cette structure ?
 - 2) Si le solde du compte 746 est modifié, combien de modifications faudra-t-il faire ?
 - 3) Comment repérer tous les clients qui possèdent un compte donné ?

✚ L'un des problèmes est qu'il y a redondance de l'écriture du solde pour tous les comptes communs. Ici, le solde 50€ est répété deux fois. Si ce solde est modifié, il faudra le mettre à jour autant de fois qu'il y a de comptes 746. De plus, comment savoir qu'il existe deux comptes 746 puisque l'accès aux données ne se fait que du haut vers le bas ? La structure arborescente ne nous permet pas de le trouver. Il est possible de repérer tous les comptes appartenant à un client donné mais l'inverse n'est pas prévu dans cette structure. En d'autres mots, il est difficile de trouver quels sont les clients qui possèdent un compte donné. Il faudra compter sur l'intelligence d'un programmeur pour trouver cette information.

➤ **Beaucoup d'espace en mémoire**

Ce système prend beaucoup d'espace en mémoire auxiliaire sur les unités de stockage. En effet, certaines valeurs doivent être répétées plusieurs fois. En plus d'occuper beaucoup d'espace, ce système favorise les erreurs de saisie et de mise à jour. Pourquoi?

➤ **Langage externe de haut niveau.**

Ce système utilise souvent un langage externe de haut niveau qui nécessite plusieurs lignes de code et un entretien coûteux.

➤ **Programmation et utilisation malaisées.**

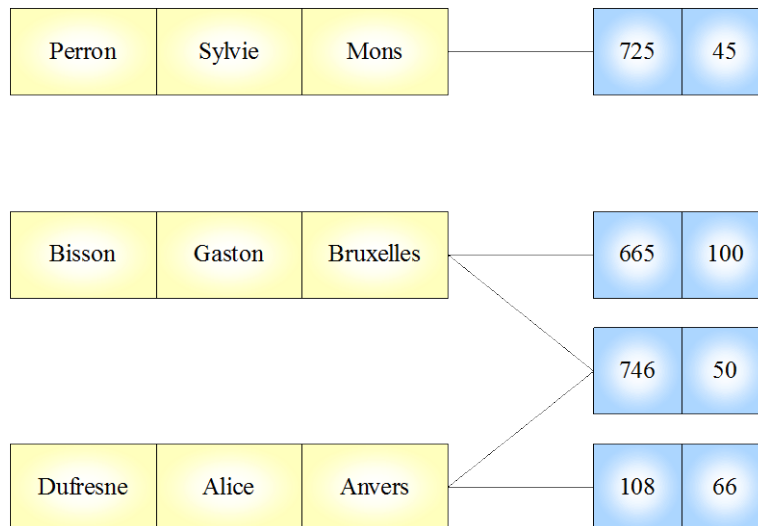
Par le fait que les données ne sont pas accessibles facilement, le programmeur doit trouver des astuces pour extraire les données nécessaires et produire les rapports demandés par les utilisateurs. Cela complique donc la programmation et l'entretien. Parfois, le programmeur ne parvient pas toujours à répondre aux besoins des utilisateurs.

Le modèle réseau contient de nombreux liens entre les différents éléments de données. L'accès aux données se réalise alors par des cheminements divers.

Dans la même période, des chercheurs de la société américaine General Electric eurent l'idée de généraliser l'approche des SGBD hiérarchiques en proposant un modèle d'organisation des données permettant des liens *plusieurs à plusieurs* entre les fichiers. La hiérarchie père-fils n'existe pas dans ce modèle car un enregistrement peut avoir plusieurs successeurs de même que plusieurs prédécesseurs. Les fichiers sont liés en réseau maillé à la manière des réseaux d'égout où à un point du réseau convergent des connecteurs **effluents** et des connecteurs affluents. Ces SGBD sont dits de type réseau (ou simplement SGBD réseau) et permettent de représenter des associations complexes entre les fichiers. Cette catégorie de SGBD a suscité beaucoup d'intérêt dans la communauté des chercheurs et des utilisateurs intéressés par les bases de données. Cela a donné lieu, à la fin des années 1960, à la création d'un groupe de travail visant à définir des normes pour le SGBD de type réseau portant sur les caractéristiques requises pour assurer la création des bases de données ainsi que la manipulation de données. Ces normes, connues sous le vocable CODASYL, décrivent en quelque sorte l'environnement requis pour la conception et l'exploitation des bases de données réseaux. Elles reconnaissent pour la première fois de manière formelle l'importance du schéma physique de la base de données, le rôle clé de la personne responsable de sa construction et de son entretien, c'est-à-dire l'administrateur de base de données (ABD). La norme met en évidence la nécessité de mettre à la disposition de l'administrateur de la base de données un langage pour la construction d'un schéma physique appelé le langage de définition de données ainsi que d'offrir au programmeur un langage pour exploiter les données, le langage de manipulation de données (**LMD**).

Exemple de modèle réseau

Un SGBD structuré en réseau se compose d'enregistrements reliés les uns aux autres par des liens. Chaque enregistrement possède un certain nombre de champs dont chacun renferme la valeur d'une donnée. Un lien est une association entre deux enregistrements. Voici un exemple qui illustre ce concept :



Le diagramme qui représente la structure réseau se compose de rectangles et de lignes :

- a) les rectangles correspondent aux enregistrements
- b) les lignes caractérisent les liens entre enregistrements

➤ **Avantage de la structure réseau**

Sur ce diagramme, un client peut avoir plusieurs comptes et un compte peut appartenir à plusieurs clients. Le type de SGBD réseau possède donc un avantage par rapport au système hiérarchique. En effet, grâce à la structure réseau, la redondance des comptes est éliminée puisqu'un compte peut appartenir à plusieurs clients. Ainsi, le compte 746 n'apparaît qu'une seule fois bien qu'il soit la propriété de deux clients. Par conséquent, la mise à jour est améliorée et le risque d'erreurs est diminué.

➤ **Désavantages de la structure réseau**

L'un des résides dans la manière d'accéder aux données. De fait, pour accéder à un enregistrement, il faut « naviguer » le long de la base jusqu'à ce qu'on l'atteigne. L'accès aux données est séquentiel jusqu'à ce que la valeur soit trouvée ou que le statut indique la fin de la base de données. La recherche dans ce système se fait uniquement de gauche à droite. Il faut passer par le premier fichier pour atteindre le second.

L'autre désavantage est que le type réseau est difficile à programmer et requiert lui aussi un langage externe de haut niveau qui nécessite plusieurs lignes de code et un entretien coûteux.

Les SGBD hiérarchiques et réseau constituent la première génération de SGBD. Malheureusement ils comportaient certaines lacunes notamment au plan des fondements théoriques. De plus ils ne permettaient pas en pratique d'assurer l'indépendance tant souhaitée entre les applications et les données. En effet, comme le lien entre deux enregistrements était implanté à l'aide d'un pointeur, soit une sorte d'adresse permettant de repérer un enregistrement associé, cela donnait lieu à des programmes complexes même pour des requêtes simples. Les SGBD de première génération auront néanmoins permis de définir de manière détaillée les spécifications de l'environnement des bases de données relationnelles.

3.3.3. Les bases de données relationnelles

Le début des années 1970 a été marqué par une avancée importante dans le domaine des SGBD. E. J. Codd, chercheur au laboratoire de recherche IBM de San Jose en Californie, s'inspirant de l'algèbre relationnelle, proposa un mode d'organisation des données ne nécessitant pas de pointeurs pour lier les enregistrements. Non seulement ce modèle avait ses fondements théoriques solidement établis dans l'algèbre relationnelle, mais il était aussi d'une grande simplicité. La structure d'un fichier est définie comme une relation entre des données provenant d'un nombre fini de domaines. Les enregistrements sont des **tuples** et constituent des occurrences de la relation. Les liens sont assurés entre deux enregistrements sur la base d'un champ de même type, commun aux deux enregistrements. Si les champs communs possèdent la même valeur, les enregistrements sont logiquement liés. Il n'est dès lors plus nécessaire de gérer des pointeurs physiques pour assurer ces liens. De *physiques* qu'étaient les liens dans les SGBD hiérarchiques ou réseau, les liens sont dorénavant *logiques*, basés sur les valeurs des champs, ce qui rend la navigation entre les enregistrements beaucoup plus souple.

Ce mode d'organisation des données a donné naissance à une profusion de SGBD relationnels commerciaux et non commerciaux dont DB2 d'IBM, Oracle, Microsoft Access, MySQL, SQL Server, pour ne nommer que les plus connus. Les SGBD relationnels sont de deuxième génération et ils ont tous en commun intrinsèquement un langage appelé SQL (Structured Query Language), SQL agissant à la fois comme langage de définition (**DDL**) et langage de manipulation de données (**DML**).

Il existe deux types de bases de données relationnelles : les bases de données **OLTP** (*Online Transactional Processing*) et **OLAP** (*OnLine Analytical Processing*). Chaque type est associé à une utilisation particulière qui dépend de l'exploitation envisagée des données.

Une base de données **OLTP** dite *transactionnelle* permet de traiter des données de façon régulière. Par exemple : La programmation de cours et les inscriptions d'étudiants constituent un bon exemple d'utilisation de base de données transactionnelle. Supposons qu'une université propose plusieurs centaines de formations. Elle en ajoute, en supprime, en modifie en fonction des besoins. Chaque formation est assurée par un professeur au moins et concerne entre dix et trois cents étudiants. Des inscriptions et des abandons sont enregistrés toute l'année. Les données sont donc dynamiques et exigent un gros volume de saisie. Une gestion sur papier nécessiterait un personnel important, d'où l'intérêt d'utiliser une base de données.

Par contre le rôle principal d'une base de données **OLAP** consiste à fournir à l'utilisateur final des données en réponse aux demandes émises. En général, les chargements volumineux de données constituent la seule activité transactionnelle effectuée dans ce type de base. Les données OLAP (performances de l'entreprise, tendances...) servent de support à des décisions commerciales circonstanciées ou relatives au fonctionnement d'un organisme. Les deux principaux types de bases de données OLAP sont les DSS (*Decision Support Systems*) et les Data Warehouses (entrepôts de

données). Toutes deux sont normalement alimentées par une ou plusieurs bases de données OLTP, et utilisées pour prendre des décisions sur le fonctionnement d'un organisme. Un Data Warehouse diffère d'un DSS dans la mesure où il contient d'énormes volumes de données collectées auprès de toutes les parties d'un organisme. Les entrepôts de données sont spécialement conçus pour le stockage de gros volumes de données et pour satisfaire de façon performante les demandes des utilisateurs.

De plus, les bases de données permettent d'entretenir un historique. Les données *historiques* sont souvent apparentées et font généralement partie d'une base de données transactionnelle. Elles peuvent être stockées aussi bien dans une base OLTP qu'OLAP. Pour les entreprises qui souhaitent conserver des données pendant des années, il est généralement inutile de stocker en ligne toutes les données. Cela augmenterait la quantité globale de données à lire pour en récupérer ou en modifier une seule. Les informations d'historiques sont généralement stockées hors ligne, le cas échéant sur un serveur dédié, un lecteur de disques ou un lecteur de bandes. Par exemple, les données des trois dernières années d'exercice d'une entreprise peuvent être enregistrées sur une bande. La question qui se pose concerne la durée de stockage en ligne, seul le client peut y répondre.

Malgré leurs nombreuses qualités, les SGBD relationnels ont un certain nombre de lacunes pour l'expression de modèles de données à un haut niveau d'abstraction. Un effort considérable de recherche et de réflexion a été fait au cours des vingt dernières années pour développer des méthodes et des formalismes permettant d'exprimer un modèle de données dépouillé des contraintes du modèle relationnel mais qui puisse se traduire assez directement dans un SGBD relationnel.

3.3.3.1. Terminologie de l'approche relationnelle

1 Base de données relationnelles :

Les systèmes de gestion de bases de données sont fondés sur des modèles de données qui s'appuient sur une description logique sophistiquée des informations.

2 Système de gestion des bases de données :

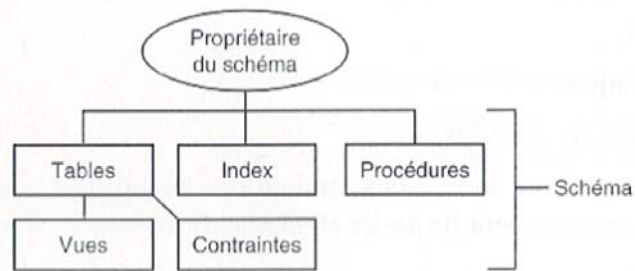
Un système de gestion des bases de données fondé sur le modèle de données relationnelles s'appelle un **S**ystème de **G**estion de **B**ases de **D**onnées **R**elationnelles (SGBDR). Dans ce type de base de données, les informations qui constituent le domaine d'application sont représentées sous forme d'un ensemble de tables et de relations.

3 Schéma :

Un schéma est un groupe d'objets d'une base de données apparentés et reliés entre eux. La table constitue l'élément le plus fondamental d'un schéma de base de données. D'autres types d'éléments peuvent résider dans un schéma de base de données : les index, les contraintes, les vues et les procédures.

Figure 1.4

Les objets constitutifs d'un schéma de base de données.

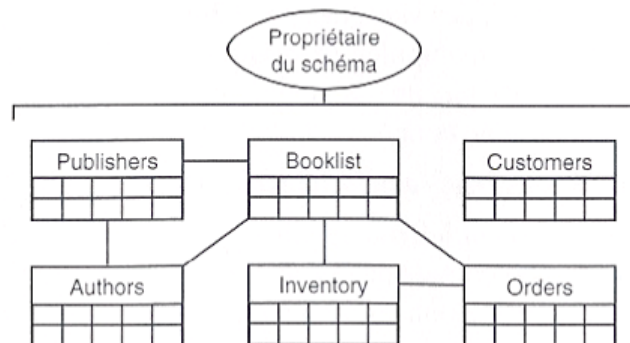


4 Table :

Dans un SGBDR les données sont *logiquement* perçues sous forme de tables. En effet, celles-ci représentent des structures de données logiques qui contiennent les informations que la base de données se propose de représenter. Les tables ne sont pas des structures physiques. Chaque table comporte un nom unique. Une table sert à stocker les données auxquelles l'utilisateur doit accéder. C'est l'unité fondamentale de stockage physique des données dans une base de données. Généralement, c'est aux tables que font référence les utilisateurs pour accéder aux données. On peut stocker dans une table n'importe quel type d'informations (texte, chiffres, graphisme, son, etc....).

Figure 1.5

Les tables d'une base de données et leurs relations.




Une base de données est constituée de plusieurs tables reliées entre elles. Selon leur fonction, on peut classer les tables en différents types :

- ✚ **les tables de données** : elles stockent l'essentiel des données d'une base de données
- ✚ **les tables jointes** : elles permettent de créer une liaison entre deux tables dans le cas d'une relation
- ✚ **les tables de sous-ensemble** : elles contiennent un sous-ensemble de données provenant d'une table de données
- ✚ **les tables de validation** : elles permettent de valider des données saisies dans d'autres tables de la base de données.


Il est possible de définir des contraintes pour contrôler les données autorisées dans une table.

Chaque table du SGBDR représente soit un objet sur lequel l'organisation veut conserver des données, soit une association entre deux ou plusieurs objets. Les objets peuvent être les clients d'une organisation, les fournisseurs, les factures, les produits, les employés, etc. Une table peut aussi être une association entre deux ou plusieurs objets. Par exemple, si une facture peut mentionner plusieurs produits et si un produit peut se retrouver sur plusieurs factures, une table représentera l'association entre facture et produit. Voici trois tables : la table CLIENT, la table COMPTE et la table COMPTE-CLIENT.


Exemple :

-  **TABLE CLIENT** : La table CLIENT représente un objet sur lequel l'entreprise veut conserver des données telles que le numéro du client, son nom, son prénom, son adresse, son numéro de téléphone, etc.

NOCLI	NOMCLI	PRENOMCLI	TELCLI	ADRCLI
PERS01	Perron	Sylvie	02/123.12.31	32, rue Marquette
BISG01	Bisson	Gaston	065/98.79.87	84, rue des Mésanges
DUFA01	Dufresne	Alice	064/32.13.21	1603, rue Brébeuf
LABP01	Labbé	Pierre	067/78.97.89	471, rue du Pinson

-  **TABLE COMPTE** : La table COMPTE représente un objet sur lequel l'entreprise veut conserver des données telles que le numéro du compte, son solde et peut-être d'autres données qui caractérisent ce compte.

NOCOMPTE	SOLDECOMPTE
108	66
665	100
725	45
746	50

-  **TABLE COMPTE-CLIENT** : La table COMPTE-CLIENT représente l'association entre la table CLIENT et la table COMPTE.

NOCOMPTE	NOCLI
108	DUFA01
665	BISG01
725	PERS01
746	DUFA01
746	BISG01
889	LABP01

✚ Cette table COMPTE-CLIENT permet de trouver :

1. A quel(s) client(s) appartient un compte donné ?

Par exemple, si nous voulons trouver à quel(s) client(s) appartient le compte 746, le SGBDR fait une recherche dans la colonne NOCOMPTE, repère toutes les rangées dont le numéro = 746 et trouve les clients correspondants Dufresne et Bisson.

2. Quel(s) compte(s) possède un client donné ?

Par exemple, si nous voulons trouver quel(s) compte(s) possède le client Bisson, le SGBDR fait une recherche dans la colonne NOCLI, repère toutes les rangées dont le numéro = BISG01, numéro qui correspond au client Bisson et trouve les comptes correspondants 665 et 746.

Vous constaterez que cette table d'association renferme deux colonnes (NOCOMPTE et NOCLI) qui se retrouvent dans chacune des deux tables objets (CLIENT et COMPTE). Ces deux colonnes permettent d'établir un lien entre ces deux objets.

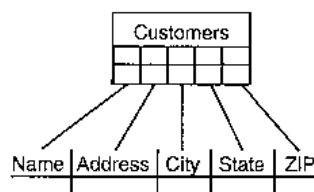
Une base de données relationnelle étant perçue comme une collection de tables où une table est une représentation matricielle composée de colonnes et de lignes de données.

5 Colonnes :

Les tables sont constituées d'un certain nombre de *colonnes*. Chaque colonne doit disposer d'un nom et deux colonnes ne peuvent avoir le même nom. Le nombre total de colonnes que comporte une table s'appelle le *degré* de la table. Une colonne ou champ est une catégorie particulière d'informations présentes dans une table. Une colonne représente une partie d'une table et constitue la plus petite structure logique de stockage d'une base de données. Chaque colonne se voit affecter un type de données et éventuellement des contraintes qui déterminent les valeurs autorisées dans la colonne correspondante. Dans la représentation d'une table, une colonne est une structure verticale qui contient des valeurs sur chaque ligne de la table.

Figure 1.6

*Colonnes d'une table
de base de données.*



Exemples de colonnes :

- ✚ Les colonnes de la table CLIENT sont : NOCLI (numéro du client), NOMCLI (nom du client), PRENOMCLI (prénom du client), TELCLI (téléphone du client) et ADRCLI (adresse du client).
- ✚ Les colonnes de la table COMPTE sont NOCOMPTE (numéro du compte) et SOLDECOMPTE (solde du compte).

- Les colonnes de la table COMPTE-CLIENT sont NOCOMPTE (numéro du compte) et NOCLI (numéro du client).

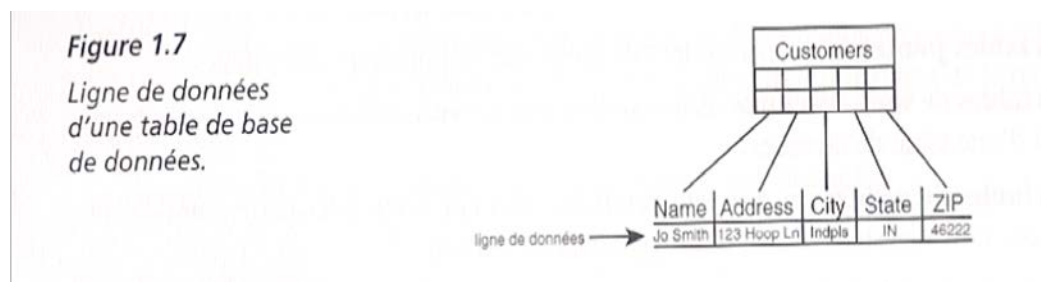
Chaque colonne de la table porte un nom unique. Cependant, un même nom peut exister dans plusieurs tables différentes.

Exemples:

- La table CLIENT comporte une et une seule colonne se nommant NOCLI mais ce nom se retrouve dans la table COMPTE_CLIENT.
- La table COMPTE comporte une et une seule colonne se nommant NOCOMPTE mais ce nom se retrouve dans la table COMPTE_CLIENT.

6 Ligne ou tuple :

Une ligne ou enregistrement est une collection de valeurs inscrites dans les colonnes successives d'une table, l'ensemble formant un enregistrement unique et un enregistrement étant l'ensemble des données relatives à la même information.



Une ligne ou enregistrement est une collection de valeurs inscrites dans les colonnes successives d'une table, l'ensemble formant un enregistrement unique et un enregistrement étant l'ensemble des données relatives à la même information.

Prenons l'exemple d'un carnet d'adresses. Pour ce carnet d'adresses, nous allons utiliser une table « contacts » qui contiendra toutes les informations concernant chaque contact, chaque enregistrement contiendra les informations relatives à un (et un seul) contact (nom, adresse, téléphone, date de naissance, etc....)

Exemples de lignes :

- La table CLIENT possède quatre lignes où chacune représente un client et renferme les caractéristiques de ce client.
- La table COMPTE possède quatre lignes où chacune représente un compte et renferme les caractéristiques de ce compte.
- La table COMPTE-CLIENT possède six lignes où chacune représente une association entre un numéro de compte et un numéro de client.

En d'autre terme : Les données de la table apparaissent comme un ensemble de lignes ou *n-tuple* où *n* est le nombre de colonne de la table. A chaque fois que le nombre de colonnes de la table est

connu, nous pouvons omettre le préfixe n et faire référence aux lignes des tables par les seuls mots lignes ou tuples. Appeler une ligne un n -tuple (pour un n , cela donné présente l'avantage d'indiquer le nombre d'entrées de la ligne). Toutefois cette terminologie n'est pas courante. La plupart des ouvrages se réfèrent aux lignes en évoquant des tuples et vice versa. Toutes les lignes d'une table ont le même format et représentent des objets ou relations du monde réel. Le nombre total de lignes présentes dans une table à *un moment donné* est appelé la *cardinalité* de la table. Dans les systèmes traditionnels, les termes *champ* et *enregistrement* sont utilisés respectivement comme synonymes des termes colonne et ligne.

Nous désignons le contenu d'une table à un moment particulier comme étant un *instantané* ou *instance* de la table. En général, lorsque les tables sont définies, le nombre de colonnes reste fixé pour toute la durée de la table. Toutefois, le nombre de lignes présentes dans la table varie car le contenu de la table reflète la dynamique de l'univers que la base de données se propose de représenter. De nouvelles lignes peuvent être insérées dans la table pour représenter de nouveaux faits. Ainsi, certaines lignes sont mises à jour pour refléter des modifications continues et d'autres lignes sont supprimées de la table car certains faits ne sont plus valables ou pertinents. Remarquez que les tables doivent comporter au moins une colonne mais pas forcément des lignes. Une table sans ligne est appelée une table vide. Le processus d'insertion de tuples la première fois dans une table est désigné par l'expression *charger une table*.

Pour *chaque* colonne d'une table, il existe un ensemble de valeurs possibles appelé son *domaine*. Le domaine désigne toutes les valeurs permises qui peuvent apparaître dans la colonne. Le domaine d'une colonne quelconque peut être vu comme une réserve de valeurs dont nous pouvons en extraire certaines. Nous indiquerons le domaine d'une colonne donnée par *Dom(nom colonne)*. Observez que toute valeur qui apparaît dans une colonne doit appartenir à son domaine. Dans une table, il est possible que deux colonnes aient le même domaine. L'exemple 2.1 montre la représentation tabulaire d'une table appelée PERSONNEL.

Exemple

Etant donné la table PERSONNEL montrée ci-dessous, identifiez son degré et sa cardinalité. Pour chaque colonne, identifiez son domaine possible

Table : PERSONNEL				
<u>Id</u>	Nom	Prénom	Département	Salaire
1420472256987	Martin	Nicolas	Comptabilité	55000
1521275458231	Benakritis	Hervé	Marketing	33500
2621106365855	Chatelin	Sophie	Ressources humaines	40000

Solution

Dans cette table, nous pouvons identifier cinq colonnes Id (numéro d'identification), Nom, Prénom, Département et Salaire. Par conséquent le degré de la table est cinq. La table ne comporte que trois lignes ou tuples. Par conséquent la cardinalité de la table est trois.

Les domaines possibles pour chacun de ces attributs sont les suivants :

- ✚ Le domaine de l'attribut Id indiqué par $\text{Dom}(\text{Id})$ est un ensemble de valeurs numériques. Dans cet exemple, $\text{Dom}(\text{Id})$ est l'ensemble de 13 chiffres positifs.
- ✚ Le domaine de l'attribut Nom, $\text{Dom}(\text{Nom})$ est l'ensemble des noms autorisés. Dans ce cas, nous supposons que les noms sont formés d'une séquence de caractères alphabétiques accentués ou non et de certains autres symboles tels que l'apostrophe ou le tiret. Le nombre de caractères que peut comprendre un nom autorisé dépend des conventions établies par l'administrateur de la base de données ou par le créateur de la table. Le nombre de caractères généralement permis pour les plages de noms va de 20 à 256 caractères.
- ✚ Le domaine de l'attribut Prénom, $\text{Dom}(\text{Prénom})$ est l'ensemble des prénoms autorisés pour une personne. Nous supposons que les prénoms suivent la même convention que les noms.
- ✚ Pour une entreprise donnée, le domaine de l'attribut Département, $\text{Dom}(\text{Département})$ est l'ensemble des noms qui ont été sélectionnés comme noms de départements autorisés. Ici, la table n'indique que trois de ses valeurs.
- ✚ Le domaine de l'attribut Salaire, $\text{Dom}(\text{Salaire})$ est un sous-ensemble de l'ensemble des nombres réels positifs. Remarquez qu'un salaire négatif n'a pas de sens.

Par facilité, nous représenterons par la notation $t(A)$ la valeur de l'entrée du tuple t dans la colonne A , c'est-à-dire la valeur à l'intersection de la colonne A et de la ligne t .

Exemple

Pour la table COMMANDE_CLIENT montrée ci-après, quelles sont les valeurs de $t(A)$ si t est un tuple arbitraire et A est une colonne quelconque de la table.

Table : COMMANDE_CLIENT			
<u>Id</u>	Date_commande	Date_expédition	Type_paiement
1	11/08/2002	12/08/2002	Comptant
2	12/08/2002	12/08/2002	Bon d'achat
10	14/08/2002	15/08/2002	Crédit

Solution

Si nous appelons t le premier tuple montré dans la table, nous pouvons dire que $t(\text{Id}) = 1$, $t(\text{Date_commande}) = 11/08/2002$, $t(\text{Date_expédition}) = 12/08/2002$ et « $t(\text{Type_paiement}) = \text{comptant}$ ».

Si t est le second tuple, nous aurons que $t(\text{Id}) = 2$, $t(\text{Date_commande}) = 12/08/2002$, $t(\text{Date_expédition}) = 12/08/2002$ et $t(\text{Type_paiement}) = \text{bon d'achat}$.

Si t est le troisième tuple nous aurons que $t(\text{Id}) = 10$, $t(\text{Date_commande}) = 14/08/2002$, $t(\text{Date_expédition}) = 15/08/2002$ et $t(\text{Type_paiement}) = \text{crédit}$.

Remarquez que les tuples de cette relation sont de type 4-tuple. Cela signifie que chaque tuple comporte 4 entrées, soit une pour chacune des quatre colonnes de la table.

En résumé : A partir des définitions précédentes et de la notion de table, nous déduisons les conditions suivantes :

- ✚ La table a un nom unique
- ✚ Chaque colonne de la table comporte un nom unique → dans une même table, deux colonnes ne peuvent avoir des noms identiques.
- ✚ L'ordre des colonnes dans la table est sans importance.
- ✚ Toutes les lignes de la table ont le même format et le même nombre d'entrées.
- ✚ Les valeurs de chaque colonne appartiennent au même domaine (chaînes de caractères, valeurs entières, valeurs réelles, etc.).
- ✚ Chaque entrée (l'intersection d'une ligne et d'une colonne) de chaque tuple de la relation doit être une valeur unique. Ceci signifie qu'aucune liste ou ensemble de valeurs n'y sont permis.
- ✚ L'ordre des lignes est sans importance du fait qu'elles sont identifiées par leur contenu et non par leur position dans la table.
- ✚ Il n'y a pas deux lignes ou tuples identiques dans toutes leurs entrées.

7 Clé primaire :

Ensemble minimal de colonnes qui permet d'identifier de manière unique chaque tuple dans une table (*Primary key*).

Une clé primaire est souvent dite *simple*, si elle ne comporte qu'une seule colonne. S'il y a un risque en faisant appel à une seule colonne d'y trouver le cas échéant des doublons, cette clé ne peut être une clé primaire valide. Il est alors nécessaire de combiner la colonne à une autre, formant ainsi une clé primaire *composée* pour assurer l'identification unique des tuples. Une clé primaire composée peut être constituée de deux colonnes ou plus. Elle est quelque fois appelée clé *composite*.


Autrement dit, une clé primaire identifie de façon unique chaque ligne d'une table. Sa valeur doit être unique, c'est-à-dire qu'elle ne doit exister que pour une seule ligne de la table, elle ne doit pas se répéter à l'intérieur d'une même table. Ainsi, dans chaque table, il n'y a pas deux lignes exactement pareilles.

Exemples:


- ✚ La clé primaire de la table CLIENT est NOCLI puisque le numéro d'un client est unique et n'appartient qu'à ce client. Deux clients ne peuvent pas avoir le même numéro.
- ✚ La clé primaire de la table COMPTE est NOCOMPTE puisque le numéro d'un compte est unique. Deux comptes ne peuvent pas avoir le même numéro.
- ✚ La clé primaire de la table COMPTE-CLIENT est une clé combinée composée de deux colonnes : NOCOMPTE et NOCLI. La colonne NOCOMPTE ne peut pas être à elle seule la clé primaire de la table COMPTE-CLIENT puisqu'une même valeur se retrouve dans plus d'une rangée. Par exemple, le numéro du compte 746 est répété deux fois. Il n'est donc pas unique. Il en est de


même pour la colonne NOCLI dans laquelle le numéro des clients DUFA01 et BISG01 est également répété deux fois.

Si vous devez concevoir une base de données, vous devez tenir compte de plusieurs facteurs pour déterminer quel est ou quels sont le ou les champs qui constitueront la clé primaire de chaque table.

 **Valeur unique :** Vous devez vous assurer de l'unicité de la valeur de la clé primaire dans chaque table. Si une colonne n'est pas suffisante, vous devez ajouter autant de colonnes que cela est nécessaire pour que la valeur de la clé primaire soit unique.

On déconseille de prendre le numéro de téléphone comme clé primaire parce que ce numéro pourrait appartenir à plusieurs personnes qui sont enregistrées dans une même table et partageraient le même toit. Le nom d'une personne ne constitue pas non plus une clé primaire valable ; cela interdirait de référencer deux personnes portant le même nom. Il en est de même pour la combinaison nom, prénom.


 **Valeur de longueur raisonnable :** De surcroît vous devez choisir une clé primaire dont la valeur n'est pas trop longue. Certains SGBDR comme Oracle par exemple, empêchent le concepteur de prendre une colonne de type *LONG*. Votre choix devra à la fois minimiser le nombre de colonnes et le nombre de caractères tout en assurant l'unicité de la clé primaire. La clé primaire sera l'objet de nombreux calculs et traitements. Une longueur trop importante ralentirait le fonctionnement du SGBDR. De plus, comme décrit au point suivant, la clé primaire d'une table est reprise dans d'autres tables. Une clé primaire trop longue augmenterait la taille des enregistrements d'autres tables.

 **Valeur non modifiable :** Puisque la clé primaire d'une table est la seule valeur de la table qui peut se répéter plusieurs fois dans les autres tables et ainsi établir un lien entre les différentes tables de la base de données, il faut donc dans la mesure du possible éviter d'avoir à modifier cette valeur.

Supposons une table CLIENT et une table FACTURE, la colonne NOCLI est la clé primaire de la table CLIENT. Cette colonne se retrouve aussi dans la table FACTURE pour établir un lien entre les deux tables et déterminer à quel client est adressée chaque facture. Supposons qu'un client a reçu 250 factures, si le numéro du client était modifié, il faudrait d'abord le modifier dans la table CLIENT et ensuite le modifier 250 fois dans la table FACTURE.

Un numéro de carte d'identité ne constitue pas une clé primaire valable puisque ce numéro change en cas de remplacement de la carte. Par contre, le numéro d'inscription au registre national pourrait convenir. Pour une voiture, le numéro de plaque ne convient généralement pas, il change quand la voiture change de propriétaire. Le numéro de châssis convient. Pour identifier une société, le numéro d'entreprise convient.

Remarque: ces considérations sont valables pour la Belgique. Pour la France, le numéro de sécurité sociale est unique pour chaque personne et pourrait servir de clé primaire. Aussi le numéro de plaque d'une voiture ne change pas et pourrait convenir.

 **Valeur dédiée :** Il est possible de prendre pour clé primaire un champ ou une combinaison de champs contenant des valeurs utiles pour l'application. Une autre possibilité est de créer

un nouveau champ dont le seul but sera de servir de clé primaire. Les différents SGBDR ont un type de champ (généralement appelé auto incrément ou numéro automatique) qui sera utilisé comme clé primaire. La valeur de champ sera calculée automatiquement par le SGBDR de manière à être unique. Il s'agit généralement d'un entier codé sur 4 octets ce qui en fait une longueur raisonnable permettant de distinguer 232 enregistrements ; soit un peu plus de 4 milliards. Ce qui est suffisant pour la plupart des applications. La valeur étant calculée automatiquement par le SGBDR, l'utilisateur ne peut la modifier.

L'ajout de ce type de champ à usage de clé primaire répond parfaitement aux trois critères énoncés ci-dessus.

En d'autre terme : La notion de *clé* est un concept fondamental du modèle relationnel parce qu'elle fournit le mécanisme de base pour récupérer des tuples dans une table de la base de données.

Du fait qu'une table peut comporter plusieurs **clés candidates**, l'une d'elles doit être désignée comme *clé primaire*. Les valeurs de cette clé doivent permettre l'identification et jouer le rôle de mécanisme d'adressage de la relation, ceci revient à différencier les différentes lignes ou enregistrements de la table à partir de leurs valeurs de clé primaire. Dès lors qu'une clé primaire a été choisie, les autres clés candidates, si elles existent, sont parfois appelées *clés alternatives*. Un SGBDR n'autorise qu'une clé primaire par table. Cette clé est constituée d'un unique attribut (clé primaire unique) ou peut en comporter plusieurs (clé *primaire composée*). Les attributs qui font partie d'une clé (primaire ou alternative) sont dits *premiers* attributs. Dans ce cours, nous soulignons les attributs qui font partie de la clé primaire. Dans les exemples précédents, l'attribut Id est souligné parce qu'il constitue la clé primaire de la relation PERSONNEL, il n'existe pas alors deux employés ayant des valeurs identiques pour Id. De même, nous avons également souligné l'attribut Id de la relation COMIMANDE_CLIENT. Du fait qu'Id est la clé primaire de la relation COMMANDE_CLIENT, il n'existe pas deux commandes ayant la même valeur de Id. Remarquez que bien que ces deux clés primaires (Id de PERSONNEL et Id de COMMANDE_CLIENT) aient des noms identiques, elles doivent être considérées comme différentes parce leurs significations sous-jacentes sont différentes.

Du fait que la clé primaire identifie de manière unique les tuples, lignes ou enregistrements d'une table, aucune de ces colonnes ne doit être *NULL*. Cette obligation impose une condition supplémentaire ou *contrainte* sur les clés connue sous le nom de *contrainte d'intégrité*. Dans une relation, une valeur NULL représente des informations manquantes, inconnues ou des données inapplicables. *Rmq* : une valeur NULL n'est pas une valeur zéro mais il s'agit d'une valeur inconnue.

Exemple

Considérons la table DEPT et les lignes montrées à la suite. Expliquez si ces lignes peuvent être ou non insérées dans la table DEPT. Remarquez que DEPARTEMENT est la clé de la table.

Table : DEPT			
<u>DEPARTEMENT</u>	NOM	LIEU	BUDGET
20	Ventes	Annecy	1.700.000
10	Marketing	Lille	2.000.000

10	Recherche	Lille	1.500.000
	Comptabilité	Nancy	1.200.000
15	Informatique	Annecy	1.500.000

Solution

10	Recherche	Lille	1.500.000	Ligne 1
	Comptabilité	Nancy	1.200.000	Ligne 2
15	Informatique	Annecy	1.500.000	Ligne 3

Ligne 1 : Non, cette ligne ne peut pas être insérée. Elle viole la propriété d'unicité de la clé parce que le département 10 existe déjà dans la table

Ligne 2 : Non, cette ligne ne peut pas être insérée. Elle viole la contrainte d'intégrité de la clé parce que la clé département ne peut pas être NULL.

Ligne 3 : Oui, cette ligne peut être insérée sans problème parce qu'aucune contrainte n'est violée.

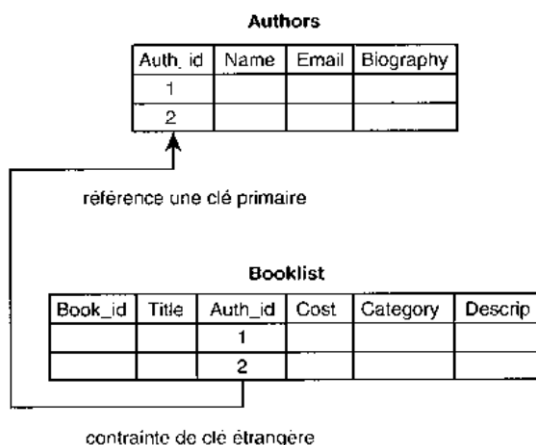
Les clés primaires sont définies au moyen d'instructions DDL (*Data Definition Language*) et sont automatiquement imposées par le SGBDR. Elles sont généralement définies au moment de la création des tables.

8 Clé étrangère :

Une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables. On y arrive en dupliquant la clé primaire de la deuxième table dans la première. On l'appelle aussi clé externe (*Foreign key*).

Une clé étrangère d'une table fait référence à une clé primaire d'une autre table. Elle est définie dans des tables enfant et assure qu'un enregistrement parent a été créé avant un enregistrement enfant et que l'enregistrement enfant sera supprimé avant l'enregistrement parent.

Figure 1.8
Intégrité référentielle
ou relations
parent/enfant.



Exemple : Pour supprimer un auteur de la base de données, on retire en premier lieu toutes les références à cet auteur dans la table enfant Booklist

Une rangée d'une table peut avoir un lien logique nommé relation avec une ou plusieurs rangées d'une ou de plusieurs autres tables. Cette relation est représentée par une clé étrangère. C'est la clé étrangère qui établit le lien logique entre les tables. Une clé étrangère est constituée d'une ou de plusieurs colonnes, elle représente la relation entre deux rangées faisant partie soit de deux tables différentes soit de la même table et correspond toujours à une clé primaire. Si elle correspond à une clé primaire combinée, elle est également combinée sinon elle est simple tout comme la clé primaire à laquelle elle correspond. Elle peut aussi correspondre à la clé primaire de la table dont elle fait partie. Dans ce cas, on dit que la relation est récursive ou circulaire et est définie sur une seule table. La clé étrangère permet d'aller chercher toutes les informations correspondant à sa clé primaire, c'est-à-dire toutes les colonnes qui ont un lien logique avec cette clé primaire. Cela évite de répéter les valeurs contenues dans ces colonnes. La seule valeur répétée est celle qui est contenue dans la clé primaire et cette valeur répétée devient la clé étrangère. La clé étrangère doit contenir soit une valeur nulle, soit la même valeur que la clé primaire à laquelle elle se réfère.

Exemples:

- ✚ Dans la table CLIENT, la rangée qui contient les données relatives à la clé primaire DUFA01 a une relation avec deux rangées de la table COMPTE-CLIENT. Cette relation est représentée par la colonne NOCLI qui est la clé primaire de la table CLIENT et une clé étrangère de la table COMPTE-CLIENT. Dans ce cas, on dit que la clé étrangère NOCLI de la table COMPTE-CLIENT correspond à la clé primaire NOCLI de la table CLIENT. Par cette relation, il est possible d'extraire tous les comptes appartenant à un client donné. Ici, grâce à la colonne NOCLI qui se retrouve dans les deux tables CLIENT et COMPTE-CLIENT, il est possible de savoir que la cliente Alice Dufresne possède les comptes 108 et 746.
- ✚ De même, dans la table COMPTE, la rangée qui contient les données relatives à la clé primaire 746 a une relation avec deux rangées de la table COMPTE-CLIENT. Cette relation est représentée par la colonne NOCOMPTE qui est la clé primaire de la table COMPTE et la clé étrangère de la table COMPTE-CLIENT. Dans ce cas, on dit que la clé étrangère NOCOMPTE de la table COMPTE-CLIENT correspond à la clé primaire NOCOMPTE de la table COMPTE. Par cette relation, il est possible d'extraire tous les clients possédant un compte donné. Ici, grâce à la colonne NOCOMPTE qui se retrouve dans les deux tables COMPTE et COMPTE-CLIENT, il est possible de savoir que le compte 746 appartient à deux clients, Alice Dufresne et Gaston Bisson.
- ✚ Ainsi la table COMPTE-CLIENT possède deux clés étrangères : NOCLI et NOCOMPTE. Dans une table, il peut y avoir plusieurs clés étrangères mais il n'y a toujours qu'une seule clé primaire même si cette clé est combinée.
- ✚ Dans la table CLIENT, toutes les colonnes (nom, prénom, adresse, téléphone, etc.) qui ont un lien logique avec la clé primaire NOCLI ne sont pas répétées dans une autre table. Seule la valeur de la clé primaire est répétée dans la table COMPTECLIENT et devient dans ce cas la clé étrangère de la table COMPTE-CLIENT.

ATTENTION : Il faut implémenter les contraintes d'intégrité référentielles avant l'admission de toute donnée dans la base pour éviter le stockage de données incohérentes ou sans relation, comme des enregistrements orphelins (enregistrements enfant sans aucune relation avec des enregistrements parent)

En d'autre terme : étant donné que des colonnes qui ont le même domaine sous-jacent peuvent être utilisées pour mettre en relation des tables d'une base de données, le concept de clé extérieure permet au SGBDR de maintenir la cohérence des lignes de deux tables. Ce concept se définit ainsi de façon formelle étant donné deux tables de la même base de données, un ensemble de colonnes de clés extérieures de la table est dit être une *clé extérieure* si les deux conditions suivantes sont satisfaites simultanément :

- la colonne d'une clé extérieure est le même domaine sous-jacent qu'une colonne d'une autre table qui est été définie comme clé primaire.
- les valeurs de clé extérieure d'un enregistrement quelconque de la table 2 soit sont NULL, soit doivent apparaître comme valeurs de clé primaire d'un enregistrement de la table 1.

D'un point de vue pratique, le concept de clé extérieure garantit que les enregistrements de la table 2 qui se réfèrent à des enregistrements de la table 1 se réfèrent bien à des enregistrements qui existent déjà. Cette condition imposée sur les clés extérieures s'appelle la contrainte *d'intégrité référentielle*. Certains auteurs désignent par *table enfant* celle qui comporte la clé extérieure et par *table parent* celle qui contient le ou (les) colonne(s) référencée(s). En appliquant cette terminologie, nous pouvons dire que la valeur de clé extérieure de chaque ligne d'une table enfant soit est NULL, soit doit correspondre à la valeur de clé primaire d'un enregistrement de la table parent.

Exemple

Soit les tables présentées ci-après. Supposons que la colonne DEPT_EMP soit une clé extérieure de la table PERSONNEL qui se rapporte à la colonne ID de la table DEPARTEMENT. Indiquez si les lignes montrées ci-dessous peuvent être insérées dans la table PERSONNEL.

Table : DEPARTEMENT		
<u>ID</u>	NOM	LIEU
10	Comptabilité	Lille
40	Ventes	Bordeaux

Table : PERSONNEL				
<u>ID_EMP</u>	NOM_EMP	CHEF_EMP	TITRE	DEPT_EMP
1234	Dupont		Président	40
4567	Duchemin	1234	Vice-président	40
1045	Valentin	4567	Directeur	10
9876	Parent	1045	Comptable	10
9213	Pasto	1045	Employé	30
8997	Maison	1234	Secrétaire	40
5932	Chatelier	4567	Employé	NULL

Solution

9213	Pasto	1045	Employé	30	Ligne 1
8997	Maison	1234	Secrétaire	40	Ligne 2
5932	Chatelier	4567	Employé	NULL	Ligne 3

Ligne 1 : non, cette ligne ne peut pas être insérée parce qu'elle viole la contrainte d'intégrité référentielle. IL n'y a pas de département 30 dans la table département.

Ligne 2 : oui, cette ligne peut être insérée sans problème parce qu'aucune contrainte n'est violée.

Ligne 3 : oui, cette ligne peut être insérée dans la table. Les valeurs NULL sont acceptables dans la colonne DEPT_EMP. La valeur NULL peut indiquer que l'employé n'a pas encore été attribué à un département.

Remarquez que dans l'exemple précédent, nous employons le mot-clé NULL pour indiquer explicitement l'absence d'un département alors que dans l'exemple 2.3 l'entrée de DEPARTEMENT a été laissée vide. Le lecteur doit savoir que certains systèmes autorisent ces deux moyens pour indiquer une valeur NULL alors que d'autres exigent que le mot-clé NULL soit indiqué explicitement.

Les clés extérieures sont généralement définies une fois toutes les tables créées et chargées. Ceci évite des problèmes tel que celui illustré dans l'exercice corrigé 2.9 ainsi que des problèmes de références circulaires. Ce dernier problème surgit lorsqu'une table se réfère à des valeurs d'une autre table qui, à son tour, se réfère à la première table. Les contraintes d'intégrité sont définies au moyen d'instructions DDL (*Data Definition Language*) et sont automatiquement imposées par le SGBDR.

9 Types de données :

Un type de données détermine l'ensemble des valeurs qu'il est possible de stocker dans une colonne de la base de données. De nombreux types de données existent. Trois sont plus particulièrement utilisés

- ✚ Le type de données alphanumérique permet de stocker des lettres, des nombres, des caractères spéciaux, et pratiquement n'importe quelle combinaison de données alphanumériques. Lorsqu'une valeur numérique est stockée dans un champ alphanumérique, elle est traitée comme un caractère, non comme un nombre, d'où l'impossibilité d'effectuer des opérations arithmétiques sur cette valeur
- ✚ Le type de données numérique sert à stocker uniquement des valeurs numériques
- ✚ Le type de données date et heure permet de stocker des valeurs de date et d'heure, lesquelles varient selon le système de gestion de base de données relationnelle utilisé

On peut trouver d'autres types de données qui peuvent varier selon le type de SGBDR utilisé.

10 Intégrité d'une base de données :

L'intégrité des données garantit que les données de la base sont exactes, en d'autres termes qu'elles vérifient des règles d'intégrité exprimées sous la forme de contraintes sur les colonnes. Ces contraintes valident les valeurs des données placées dans la base de données, garantissent l'absence de données dupliquées ou le respect des règles de gestion après modification ou ajout de données. Elles peuvent être implémentées aussi bien au niveau colonne qu'au niveau table.

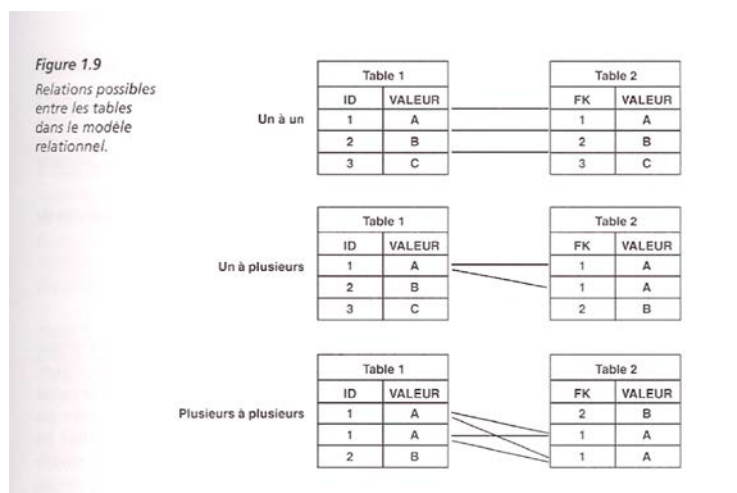
Quand les tables entretiennent des relations parent/enfant, les valeurs d'une colonne dépendent des valeurs d'une colonne d'une autre table. L'intégrité *référentielle* garantit que les données de tables apparentées sont cohérentes ou synchronisées. Ces données doivent vérifier des règles exprimées sous la forme de contraintes référentielles. La représentation de ces contraintes nécessite la définition de clés. Une *clé* est une valeur de colonne d'une table ou une combinaison de valeurs de colonnes qui permet d'identifier une ligne de cette table ou d'établir une relation avec une autre table. Il existe deux types de clés : primaires et étrangères

11 Relations :

Les bases de données comportent généralement plusieurs tables dont la plupart sont en relation les unes avec les autres. Dans les bases de données relationnelles, les relations sont souvent établies par l'intermédiaire de clés primaires et étrangères. On répartit les données dans des tables que l'on relie pour réduire la redondance des données. Le processus de réduction de la redondance de données dans une base de données relationnelle est baptisé normalisation.

Les trois types de relations entre deux tables sont :

- ✚ **Un à un** : un enregistrement d'une table est en relation avec un seul enregistrement d'une autre table
- ✚ **Un à plusieurs** : un enregistrement d'une table peut être en relation avec plusieurs enregistrements d'une autre table
- ✚ **Plusieurs à plusieurs** : un enregistrement d'une table 1 peut être en relation avec un ou plusieurs enregistrements d'une table 2 et un ou plusieurs enregistrements de la table 2 peuvent être en relation avec un ou plusieurs enregistrements d'une table 1. Dans le modèle relationnel, les tables parent peuvent avoir plusieurs tables enfant et les tables enfant plusieurs tables parent.



12 Index :

Un index sert à accélérer le processus de récupération des données d'une table. Par exemple, on peut créer un index sur le nom des clients, c'est-à-dire stocker les patronymes par ordre alphabétique pour que les utilisateurs ayant besoin de rechercher des clients par leur nom soient plus vite renseignés. Les lignes de l'index pointent vers les lignes correspondantes de la table qui contient le nom des clients, comme un index de livre indique une page donnée.

13 Domaine :

Ensemble des valeurs admises pour un attribut. Il établit les valeurs acceptables dans une colonne (*Domain*).

Une relation peut faire appel au même domaine pour plusieurs de ses attributs. La notion de domaine est fondamentale en matière de gestion de données. Elle permet d'exprimer des contraintes d'intégrité sémantique très fortes sur les données d'une BD.

14 Contraintes :

Une contrainte sert à imposer des règles aux données, plus précisément à contrôler les données autorisées dans une colonne et à assurer l'intégrité référentielle (relations des tables parent et enfant). Elles sont créées au niveau des colonnes.

15 Vues :

Une vue est une table virtuelle en ce sens qu'elle ressemble à une table et fonctionne comme elle. Elle est définie d'après la structure et les données d'une table. On peut l'interroger et parfois la mettre à jour. Les vues permettent d'assurer l'objectif d'indépendance logique. Grâce à elles, chaque utilisateur pourra avoir sa vision propre des données.

16 Triggers :

Un trigger (déclencheur) est une unité de code de programmation stockée dans la base de données. Il se déclenche en fonction d'un événement survenu dans la base et donne lieu à des modifications de données réalisées en fonction d'autres données modifiées ou jointes. On se sert des déclencheurs pour gérer les données redondantes.

17 Procédures stockées :

Une procédure est un programme stocké et exécuté au niveau de la base de données qui sert essentiellement à gérer les données et à effectuer des traitements par lots.

3.3.4. Les bases de données orientées objets et les autres

Le développement de langages orientés objets tels que Small talk et C++ a conduit à la mise au point de SGBD devant assurer la persistance des objets, soit le stockage permanent sur un support de mémoire auxiliaire des objets créés à l'aide de tels langages. Ce SGBD dit orienté objet ou simplement SGBD objet qui a connu un essor somme toute limité appartient à la troisième génération. Le SGBD Gem stone par exemple est une extension du langage Small talk. Le développement des SGBD objet a été freiné par des SGBD hybrides incorporant le modèle relationnel et le stockage d'objets. Les objets peuvent être soit de type structuré comme ceux créés par un langage orienté objet ou de type non structuré telles que des images, de la vidéo, des trames sonores. Le type non structuré est aussi appelé BLOB (Binary Large Object). On donne le nom SGBD relationnel-objet à cette forme hybride car il combine les propriétés du SGBD relationnel et du SGBD objet. L'éditeur américain Informix a été un précurseur en matière de SGBD relationnel-objet pour être suivi et vite dépassé par Oracle et IBM avec leur produit Oracle8 et DB2 Universal Database System respectivement.

A l'aube du XXI^e siècle, on parle d'une quatrième génération de SGBD. Il s'agit d'une catégorie hétérogène de SGBD conçus avant tout pour des applications spécialisées, comme par exemple et non exclusivement, les SGBD OLAP (On Line Analytical Processing) largement utilisés pour l'entreposage des données et l'exploration des données, les SGBD XML pour les applications Web, ou enfin les SGBD de contraintes pour les applications d'optimisation.

4. Environnement de bases de données

Comme nous l'avons vu à la section relatant les caractéristiques des SGBD, une base de données est par définition partagée par plusieurs utilisateurs finaux. Elle doit en conséquence satisfaire leurs besoins en information quel que soit l'utilisateur. L'utilisateur final est un élément central de l'environnement d'une base de données car il en est de fait le client. Mais il n'est pas la seule personne jouant un rôle majeur dans cet environnement. Nous avons évoqué plus tôt le rôle joué par **l'administrateur de bases de données**. Notons brièvement le rôle d'autres individus qui interviennent dans l'environnement d'une base de données.

✚ **Les concepteurs de bases de données** : Leur rôle consiste à élaborer le design d'une base de données. Ils traitent directement avec les utilisateurs dans une organisation pour identifier les données et les contraintes sur les données qui devront être conservées dans une BD. Les concepteurs sont aussi appelés des modélisateurs de données car leur travail consiste en outre à élaborer des modèles de données, soit des plans de plus en plus détaillés des données et des contraintes sur les données de la BD. Comme nous le verrons plus loin, leur travail consiste à élaborer des modèles du plus général au plus spécifique. Le modèle le plus général est le modèle conceptuel des données, indépendant du type de SGBD cible. Il ne fait état que des entités, de leurs attributs, des associations et contraintes sur les données. Les contraintes sur les données sont généralement exprimées sous forme de règles de gestion des données. Un modèle de données plus spécifique est le modèle logique de données qui est élaboré en fonction d'un type de SGBD tel que relationnel, réseau, hiérarchique ou objet. Enfin le modèle physique de données est la transposition d'un modèle logique dans le SGBD adopté pour implanter la BD et l'application de base de données.

✚ **Les développeurs d'applications** : Il s'agit essentiellement des personnes qui interviennent pour définir l'architecture et réaliser les programmes d'application qui vont fournir aux

utilisateurs les fonctionnalités pour accéder à la base de données dont le modèle physique vient d'être réalisé. Ce sont les architectes, les analystes et les programmeurs.

De manière à supporter efficacement le travail des concepteurs de BD et les développeurs d'application de base de données, bon nombre d'éditeurs d'outils de développement ou de SGBD adoptent une proposition faite par un organisme de normalisation américain, l'ANSI (American National Standards Institute), qui préconise une approche à trois niveaux pour réaliser le design d'une base de données. Bien qu'il ne s'agisse pas d'une norme en bonne et due forme, ce découpage fait désormais référence en la matière. Voyons en quoi consiste cette proposition qui a fait l'objet d'un premier rapport en 1975, autour duquel un large consensus s'est développé.

4.1. Les niveaux d'abstraction

4.1.1. Le niveau externe

En vertu de la proposition de l'ANSI, les données d'une base de données sont décrites selon trois niveaux d'abstraction. La façon dont un utilisateur perçoit les données s'appelle une *vue*. Chaque utilisateur, selon ses besoins et ses prérogatives, peut avoir une vue différente sur les données stockées dans une BD. Ce niveau d'abstraction est appelé le niveau externe. Il s'agit du plus haut niveau d'abstraction.

4.1.2. Le niveau interne

Au plus bas niveau d'abstraction se situe le niveau interne. Il représente la manière dont le SGBD perçoit les données stockées dans une BD. La description des données au niveau interne est donnée par deux modèles : le schéma logique des données et le schéma physique des données. Ces modèles sont fortement dépendants du type de SGBD choisi pour mettre en œuvre la base de données.

4.1.3. Le niveau conceptuel

De manière à assurer une totale indépendance de la description des données, soit en regard d'un groupe d'utilisateurs soit d'un SGBD en particulier, une couche intermédiaire doit exister et on l'appelle le *niveau conceptuel*. A ce niveau, le *schéma conceptuel de données* va décrire un domaine de données à prendre en charge par une BD sans égard à un utilisateur en particulier et il sera relativement stable dans le temps. Cela signifie qu'un changement à l'organisation physique d'une BD ou encore le choix d'un autre SGBD n'aura aucun impact sur les vues déjà offertes aux utilisateurs. Par ailleurs, l'ajout de nouvelles vues ou d'un nouveau groupe d'utilisateurs avec de nouvelles vues ne devrait avoir aucun impact sur l'organisation physique des données. Le niveau conceptuel assure ainsi une couche *d'isolation* entre les besoins des utilisateurs et l'organisation physique de la BD. Chaque vue est en fait un sous-ensemble du schéma conceptuel et est quelques fois appelée un *sous-schéma*.

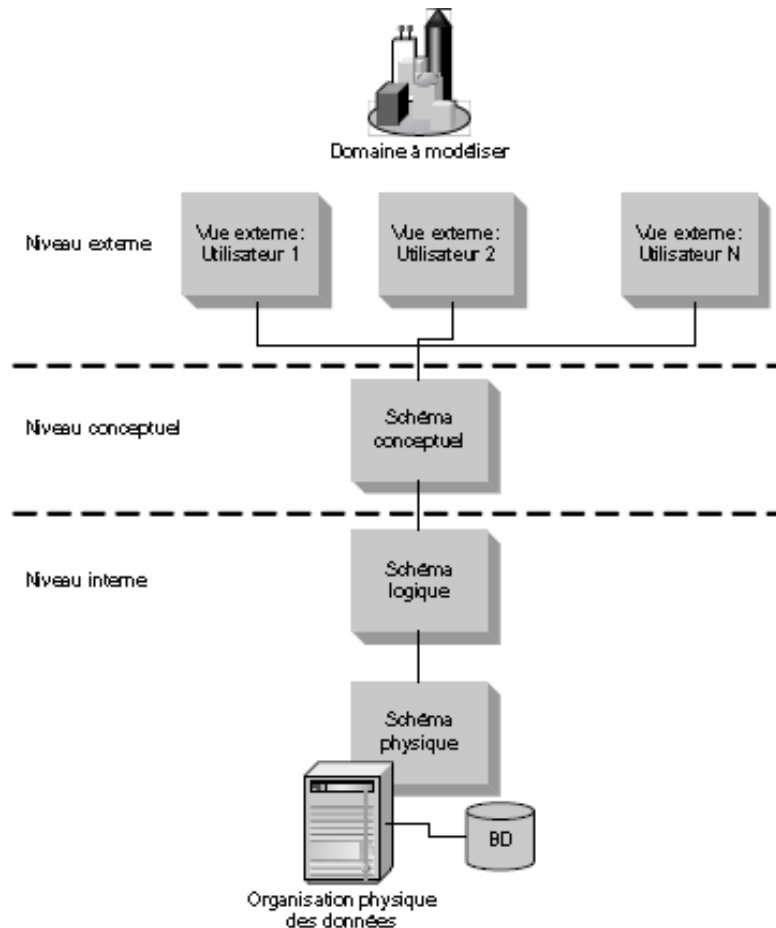


figure 0-2 : Les trois niveaux d'abstraction de l'ANSI

La figure 0-2 présente les trois niveaux d'abstraction sur les données tels que proposés par l'ANSI. Les trois niveaux d'abstraction et les modèles qui en découlent offrent un cadre rigoureux de description des données pour les concepteurs d'une base de données. L'élaboration du modèle physique de données et la réalisation des applications de base de données exigent par ailleurs des outils et des langages dont les développeurs font usage. Les travaux du groupe **CODASYL** ont permis d'identifier deux types de langage, destinés surtout aux développeurs mais aussi aux utilisateurs experts, qui devraient être présents dans l'environnement d'une base de données. Il s'agit du *langage de définition de données (LDD)* et du *langage de manipulation de données (LMD)*.

4.2. Langages de bases de données

En pratique cependant on ne retrouve dans un SGBD qu'un seul langage qui comporte à la fois des instructions pour la définition de données et d'autres pour la manipulation des données. Ce langage est en quelque sorte constitué de deux ensembles d'instructions ayant leur vocation propre. De plus, bien que les instructions écrites dans ce langage puissent être exécutées directement par le SGBD par l'intermédiaire d'un module appelé interprète, elles sont généralement enchâssées par le programmeur dans un programme réalisé avec un langage de haut niveau tel que C, C#, C++, Java ou Visual Basic, puis compilées avec les autres instructions du programme pour produire le programme d'application exécutable.

Langage de définition de données (LDD) : langage qui permet au concepteur du schéma physique ou à l'administrateur de la base de données de nommer les entités, les attributs et associations constituant la structure de la BD, ainsi que les contraintes d'intégrité et de sécurité associées. (Data Definition Language).

Le schéma physique est décrit à l'aide d'un script, c'est-à-dire une séquence d'instructions du LDD ne comportant ni condition, ni itération contrairement aux programmes écrits dans un langage de haut niveau. Le LDD est donc par définition *non procédural*. Le script suivant, écrit pour un SGBD relationnel dans le langage SQL, permet de créer l'entité **Ami** (une table dans le modèle relationnel) avec ses attributs et d'ajouter une contrainte sur l'attribut **RéfAmi** qui en fait la clé primaire de la table

```
CREATE TABLE Ami  
  ([RéfAmi] integer,  
   [Nom] varchar (30),  
   [Prénom] varchar (30),  
   [DateNaissance] date,  
   [Téléphone] varchar (10),  
   [Remarques] memo,  
   CONSTRAINT [Index1] PRIMARY KEY ([RéfAmi]) );
```

La compilation des instructions d'un script de LDD produit des métadonnées qui seront conservées avec la BD dans ce qu'il est convenu d'appeler le catalogue système.

Langage de manipulation de données (LMD) : langage qui fournit un ensemble d'opérations élémentaires de manipulation de données telles que l'insertion, la modification, la recherche, l'extraction et la suppression de données contenues dans une base de données (Data Manipulation Language).

Alors que le langage de définition de données est réservé aux spécialistes, concepteurs ou administrateurs, le langage de manipulation de données est très souvent utilisé, directement ou indirectement, par les utilisateurs finaux. Les instructions du LMD qui permettent la recherche et l'obtention de données constituent un langage de requête (query language). On réserve généralement le terme requête pour une instruction écrite dans un langage de requête pour la recherche et l'extraction de données dans une BD. Cette distinction peut paraître étonnante au novice car le langage SQL (Structured Query Language), malgré ce que laisse croire son nom, est un langage qui combine à la fois des instructions de définition et de manipulation de données. Sur le plan de la manipulation de données, il se présente non seulement comme un langage de requête au sens strict mais offre en plus des instructions pour la suppression, l'insertion ou la modification des données. SQL est donc un LMD complet, couplé d'un LDD.

Voici exprimé dans le langage SQL une instruction pour la recherche et l'extraction des enregistrements de la table **Ami** dont la valeur du champ **DateNaissance** est strictement supérieure au premier janvier 2000.

```
SELECT Ami.*  
FROM Ami  
WHERE ((Ami.DateNaissance) > #1/1/2000#);
```

Plusieurs SGBD relationnels offrent désormais une interface graphique pour faciliter à l'utilisateur la tâche de rédaction d'une requête. Il s'agit en fait de véritables langages de requête qui permettent

de construire une requête par l'exemple. Un représentant bien connu de cette catégorie est le QBE (Query-by-Example) intégré au SGBD Microsoft Access.

Ainsi, la requête exprimée ci-dessus en SQL peut être aussi définie à travers une grille comme le montre la figure 0-3. Tout comme SQL, QBE permet aussi de formuler des requêtes pour la suppression, l'insertion ou la modification d'enregistrements. L'utilisateur n'a donc plus à se plier à une syntaxe souvent rébarbative pour formuler sa requête. Sa tâche consiste alors essentiellement à compléter les paramètres de la requête à travers une grille.

The screenshot shows the Microsoft Access Query Design view for a query named 'Requête1'. The design grid is as follows:

Champ	Table	Tri	Afficher	Critères	Ou
Ami.*	Ami		<input checked="" type="checkbox"/>		
DateNaissance	Ami		<input type="checkbox"/>	>#1/01/2000#	
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		

Figure 0-3 : Requête de sélection exprimée graphiquement avec QBE

4.3. Architecture des SGBD multiutilisateurs

La première architecture mise en œuvre pour permettre à plusieurs utilisateurs d'accéder simultanément à une base de données est celle du télétraitement. Ce terme met en évidence le fait que toute la capacité de traitement des données est centralisée sur un ordinateur central auquel les utilisateurs accèdent par le biais de simples terminaux. Ces terminaux sont du type passif car ils n'interviennent nullement dans le traitement des données, ne servant qu'à transmettre, recevoir et disposer à l'écran les données reçues.

Une telle architecture impose à l'ordinateur central une charge de traitement considérable car ce dernier doit supporter non seulement les tâches d'accès à la base de données et d'exécution de l'application de base de données mais en outre celle de mettre en forme des données transmises aux terminaux qui n'auront qu'à les afficher selon les directives reçues. Cette architecture impose aux organisations l'exploitation et l'achat coûteux d'ordinateurs de grande taille. La figure 0-4 illustre cette architecture qualifiée aussi de traitement centralisé.

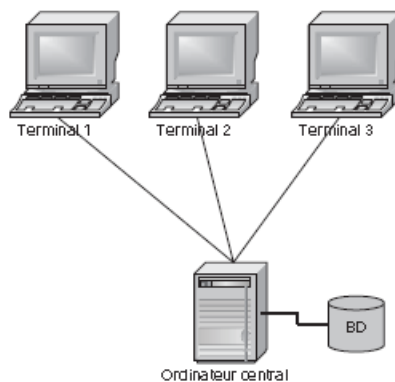


Figure 0-4 : Architecture de traitement centralisé

Le développement des ordinateurs personnels peu coûteux ainsi que des réseaux de communication à haut débit a permis l'élaboration d'une nouvelle architecture misant sur la capacité de traitement dévolue localement aux postes de travail des utilisateurs. La capacité de traitement n'est plus désormais l'apanage de l'ordinateur central. Les utilisateurs disposent alors de postes de travail qui prennent en charge en tout ou en partie le traitement des données. Cette architecture est dite à traitement distribué par opposition à la forme traditionnelle de traitement centralisé. La forme la plus connue de traitement distribué est l'architecture client-serveur.

Architecture Client-Serveur : spécification d'un système informatique dans lequel un processus appelé le serveur agit comme fournisseur de ressources pour d'autres processus qui demandent ces ressources, soit les processus clients. Le processus client et le processus serveur s'exécutent le plus souvent sur des machines différentes reliées au même réseau.

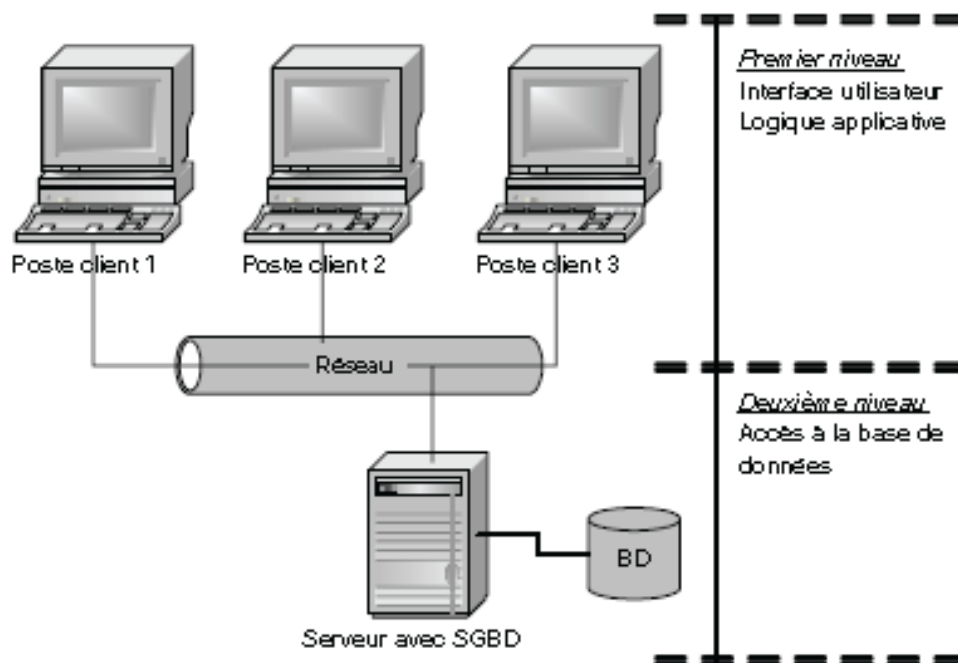


Figure 0-5 : Architecture client-serveur à deux niveaux

L'architecture client-serveur vise à décharger l'ordinateur central du plus grand nombre de tâches possible. La forme la plus simple que prend cette architecture est le modèle à deux niveaux (en anglais two-tier) illustré à la figure 0-5. Dans les applications de base de données, on retrouve généralement trois composants principaux : **la logique de présentation (interface utilisateur), la logique applicative aussi appelée la logique métier et la logique d'accès aux données.**

- ✚ **La logique de présentation** est responsable de la gestion de l'interface utilisateur notamment la présentation des données à l'écran et l'interaction avec l'utilisateur.
- ✚ **La logique applicative** prend en charge les traitements propres à l'application, soit les règles d'affaires d'un métier, la nature des calculs qui doivent être effectués.
- ✚ **La logique d'accès aux données** est relative à l'exécution des requêtes par le SGBD pour fournir à la logique applicative les données dont elle a besoin pour exécuter les processus du métier.

En vertu de l'architecture client-serveur à deux niveaux, le poste de travail de l'utilisateur, appelé le poste client, prend en charge tous les traitements liés à la fois à la logique de présentation et à la logique applicative. La logique d'accès est dévolue à un serveur qui agit comme serveur de données exclusivement. Le poste client compose des requêtes, dans le langage SQL par exemple, selon les besoins en données de l'application qui s'exécute sur le poste. Il transmet ensuite la requête à travers le réseau au serveur qui l'exécute à son tour et retourne en réponse au poste client le résultat de son exécution.

Une telle architecture permet aux organisations de se doter de ressources informatiques centrales de taille modeste tout en misant sur les ordinateurs personnels des utilisateurs, par ailleurs utilisés pour des tâches de bureautique, pour l'exécution d'applications de base de données. Cet avantage est obtenu au prix d'une administration beaucoup plus lourde du parc d'équipement, d'une plus grande complexité des applications et en conséquence de frais d'entretien et de mise à jour plus élevés.

De manière à simplifier l'entretien et la mise à jour de la logique applicative, un autre modèle client-serveur a été mis au point permettant de délester le poste client de ce composant. Il s'agit de l'architecture client-serveur à trois niveaux (three-tier). Dans ce modèle la logique applicative est prise en charge par un deuxième serveur, ne laissant au poste client que les tâches liées à la gestion de l'interface utilisateur. C'est ainsi qu'est né le concept de **client léger**, léger dans le sens que le poste client ne nécessite plus de ressources aussi importantes en matière d'espace de mémoire principale et de vitesse d'exécution que demandait le modèle à deux niveaux. Dans certaines applications client-serveur à trois niveaux, la gestion de l'interface utilisateur se fait avec un simple navigateur Web, avec pour conséquence une administration considérablement simplifiée des mises à jour des applications côté client. L'entretien de la logique applicative ne concerne désormais qu'une seule machine que l'on appelle le serveur d'application. La figure 0-6 illustre cette deuxième forme d'architecture client-serveur.

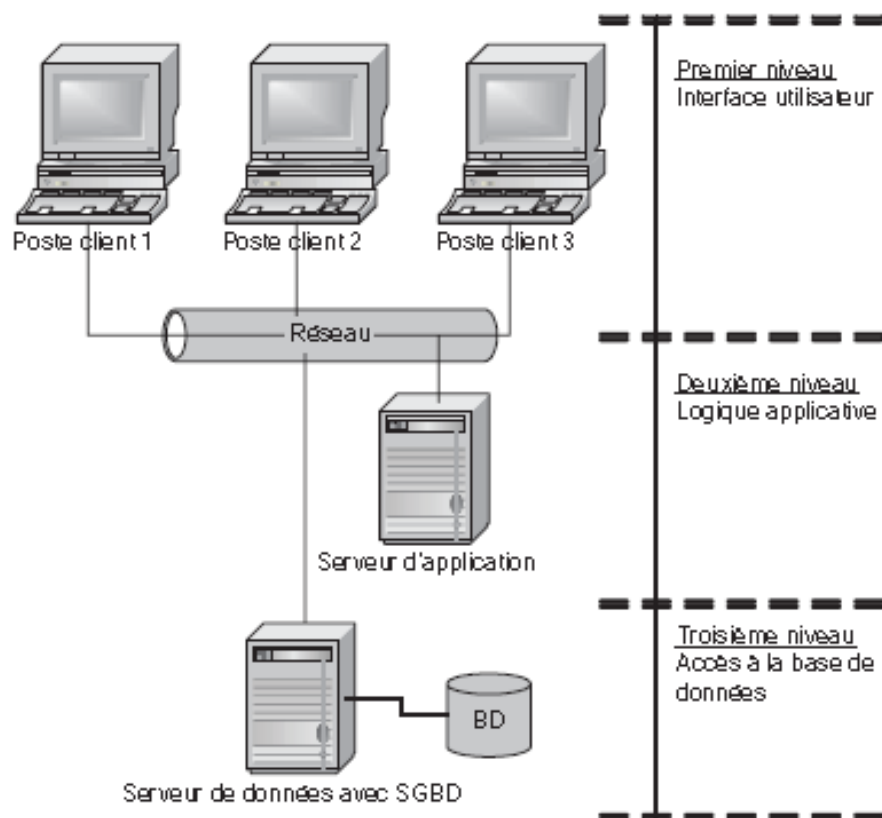


figure 0-6 Architecture client-serveur à trois niveaux

La facilité d'administration des mises à jour n'est pas le seul avantage d'un modèle à trois niveaux. Il est en revanche d'une grande souplesse. Il permet par exemple d'assurer aux utilisateurs un service sans interruption en cas de panne. En mettant en œuvre ce modèle avec plusieurs serveurs d'application, offrant tous la même logique applicative, la panne d'un serveur a peu d'impact sur les utilisateurs car les autres serveurs pourront prendre la relève en tout temps même si la performance globale s'en trouvera légèrement dégradée. De plus, le serveur d'application peut communiquer avec d'autres serveurs d'applications pour obtenir des ressources et des services spécialisés. Dans une application de commerce électronique par exemple, le serveur qui prend en charge la transaction du client peut faire appel à un autre serveur, spécialisé dans le traitement du paiement par carte de crédit, de manière à compléter l'achat. Le commerçant peut se procurer ce service auprès d'un fournisseur, sans avoir à développer ou à exécuter sur ses propres équipements ce composant du système.

Nous concluons cette section avec une illustration présentée à la figure 0-7 qui schématise l'environnement des bases de données que nous venons de décrire.

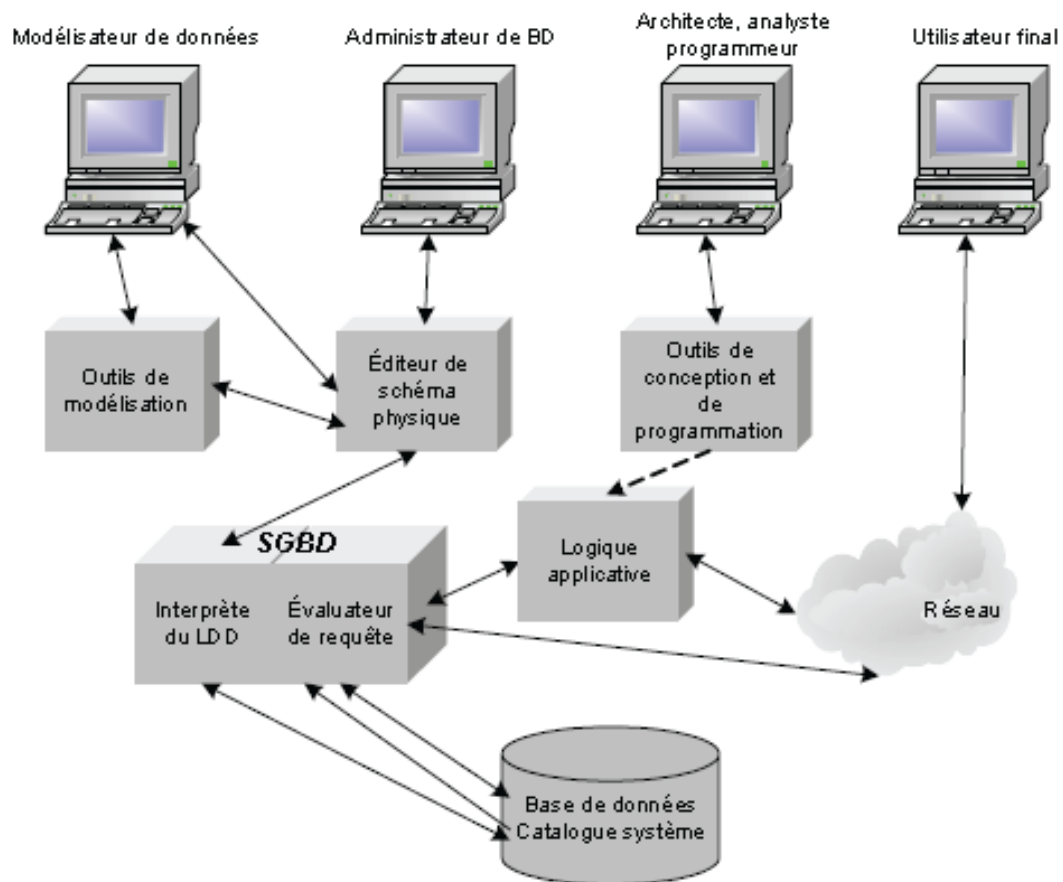


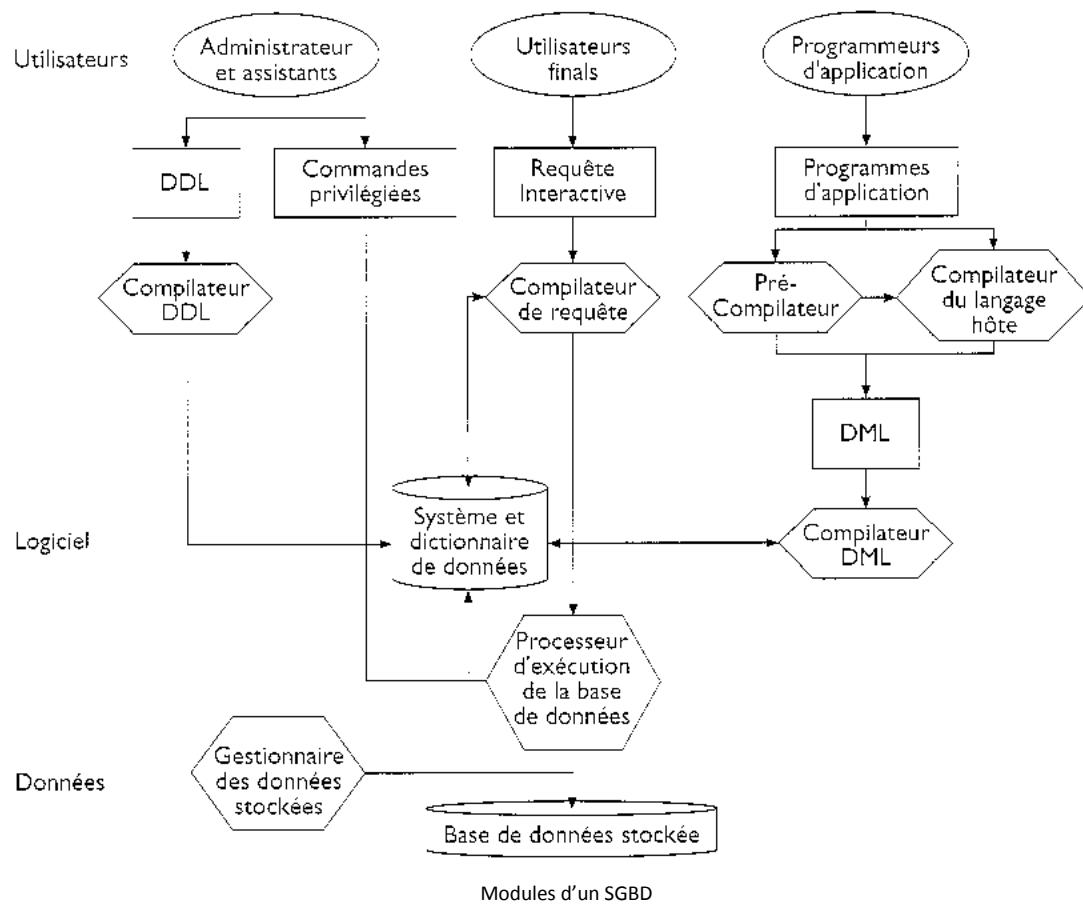
figure 0-7 : Environnement de bases de données

Regrouper les modules

Nous avons traité tous les composants du SGBD, nous allons nous intéresser aux divers modules de logiciels que l'on rencontre habituellement dans les SGBD et où ils se trouvent dans le système informatique considéré comme un tout. Il est probablement plus facile d'approcher cet environnement depuis le point de vue des différents utilisateurs abordé précédemment. Les figures ci-dessous montrent les relations des utilisateurs et des divers modules logiciels avec les données réelles.

Les données sont stockées sur un disque. L'administrateur et ses assistants peuvent exécuter les commandes et les instructions DDL (langage de définition de données) dont ils disposent et qui sont d'abord gérées par le compilateur DDL (langage de définition de données).

Le SGBD offre une protection complète des données par l'intermédiaire du dictionnaire des données, du gestionnaire d'exécution de la base de données et du gestionnaire des données stockées. Chaque accès aux données stockées passe par un ou plusieurs de ces composants. Le SGBD vérifie de façon cohérente par rapport au dictionnaire de données que tous les accès sont légaux et il traite ces commandes via le processeur d'exécution. Le gestionnaire d'exécution de la base de données gère chaque requête, qu'il s'agisse d'une requête de récupération ou de mise à jour. Seul l'administrateur et le personnel ont accès au gestionnaire des données stockées pour la création et la mise à jour de la structure effective de la table.



Exercices

Suivant les situations présentées dans les énoncés ci-dessous, veuillez réaliser sur Access la base de données permettant de répondre au mieux aux besoins spécifiés.

1. Personnes – Départements

Une personne appartient à un seul département. Dans un département il peut y avoir plusieurs personnes. L'entreprise veut connaître le nom des employés d'un département et le département dans lequel travaille un employé déterminé.

2. Cours – Professeurs

Un professeur peut dispenser plusieurs cours et un cours peut être dispensé par plusieurs professeurs. (Remarque : ici est considérée la capacité du professeur à dispenser le cours)

2^{ème} version : les cours donnés par un professeur à une date donnée

3. Clients – factures

Un client peut se voir adresser plusieurs factures mais une facture correspondra toujours à un seul client.

4. Clients – Comptes

Un client peut avoir plusieurs comptes et un compte peut appartenir à plusieurs clients

5. Factures – Produits

Une facture peut contenir plusieurs produits et un produit peut être repris sur plusieurs factures.

6. Employés – Départements - Ordinateur

Un employé ne peut travailler que dans un seul département mais par contre il peut utiliser plusieurs machines dans son département. L'entreprise désire retracer l'utilisation du matériel de chaque département sachant qu'il se peut qu'un employé n'utilise pas d'ordinateur et qu'un ordinateur ne soit jamais utilisé.

7. Employé et superviseur

Un employé supervise de 0 à plusieurs personnes. Il se peut qu'un employé n'ait pas de superviseur, s'il est le responsable. Il se peut également qu'un employé ne supervise personne. Dans ce cas l'entreprise désire retracer la liste des superviseurs ainsi que les personnes qui sont sous leur responsabilité. Elle souhaite également ressortir le nom du superviseur responsable d'un employé déterminé.

8. Infirmières – Médecins

1^{ère} version : Un médecin a sous ses ordres une ou plusieurs infirmières. Mais par contre une infirmière ne pourra être sous les ordres que d'un seul médecin.

2^{ème} version : Un médecin a sous ses ordres une ou plusieurs infirmières. Une infirmière pourra être sous les ordres de plusieurs médecins.

3^{ème} version : Ajouter les patients à ces deux versions sachant que l'on désire connaître à chaque visite de ce dernier le médecin et l'infirmière qui l'on pris en charge ainsi que la pathologie et le diagnostic mis en place.

9. Personnes – immeubles

Une personne gère un ou plusieurs immeubles sans en être nécessairement le propriétaire. Un immeuble peut être la propriété d'une personne qui ne le gère pas nécessairement. La commission des propriétaires souhaite connaître le propriétaire d'un immeuble ainsi que la personne responsable de sa gestion.

10. Historique des adresses

Une personne peut au cours de sa vie déménager plusieurs fois. L'état désire connaître les différents lieux de résidence présent et passé d'une personne.

11. Produits – Entrepôt

Une organisation a besoin du nom, de l'adresse et du numéro de téléphone des membres qui en font partie. L'entreprise possède plusieurs entrepôts. Elle a besoin de connaître la quantité en stock de chacun des produits dans chacun de ses entrepôts.

12. Facture – Produits

Une facture peut contenir plusieurs produits. Pour chacun des produits facturés nous avons besoin de connaître la quantité vendue et le prix unitaire.

13. Professeur - Imprimante

Dans une école, chaque professeur a un bureau. Sur chaque bureau est installée une imprimante reliée au serveur de l'école. Mais un professeur peut imprimer sur n'importe quelle imprimante reliée au réseau en cas de panne de son imprimante locale. L'école veut retracer quels professeurs utilisent quelles imprimantes.

14. Bibliothèque

Dans une bibliothèque, il y a plusieurs sections. Chaque section représente une catégorie de livres (Roman, Informatique, Biographie, ..). Chaque livre a un numéro unique et peut exister en plusieurs exemplaires. La bibliothèque veut retracer dans quelle section se trouve tel ou tel livre. Elle veut aussi conserver la description de chaque livre (auteur, titre, collection, édition, année, nombre de pages).

15. Restaurant

Un restaurant a deux serveuses et trois cuisiniers. Lorsqu'une serveuse reçoit une commande des clients, elle la communique aux cuisiniers et un des cuisiniers prépare la commande. Le restaurant veut retracer quel cuisinier a préparé quelle commande reçue par quelle serveuse.

16. Bureau de consultants

Un petit bureau de consultants a des clients pour lesquels il effectue des projets. Un projet doit toujours être associé à un et un seul client. Cependant, il est possible d'avoir un client sans avoir effectué de projets. Pour un projet, on fait appel à des pigistes. Ces pigistes peuvent être appelés à travailler sur plusieurs projets. Chez un client, il y a un ou plusieurs employés qui agissent comme personnes-ressources. Ces employés peuvent être associés à plusieurs projets.

17. Club de parapente

Un club de parapente souhaite gérer l'utilisation des parapentes qu'il loue à la journée aux membres d'une part et d'autre part, suivre les vols de ces mêmes membres. Chaque parapente est identifié de façon unique.

Le club désire connaître :

- la date de mise en service
- le modèle du parapente
- les lieux d'atterrissage et de décollage de chaque vol

De plus il est obligatoire que chaque membre ou pilote possède un numéro de licence

18. Accidents dans une usine

La direction d'une usine souhaite avoir une trace des accidents impliquant au moins une personne qui ont lieu dans un des différents sites de l'usine. Avant de penser à informatiser ce processus, il vous est demandé de représenter les différentes données qui sont susceptibles de composer la notion d'accident.

Dans le cadre de cet énoncé vous devez prendre en considération :

- ✓ Le ou les employés impliqués dans l'accident
- ✓ L'accident touchant au moins une personne
- ✓ Les lésions occasionnées aux personnes victimes de l'accident, ainsi qu'aux personnes responsables de l'accident.
- ✓ Les causes naturelles éventuellement à l'origine de l'accident
- ✓ Le lieu précis de l'accident

Les contraintes sont les suivantes :

- ✓ Une personne peut être victime d'aucun ou de plusieurs accidents sur l'année. Une personne pourrait être victime d'un accident mais par chance, ne se verrait occasionner aucune lésion. Par contre, une personne pourrait être victime de plusieurs lésions au cours d'un même accident.
- ✓ Un accident implique au moins une personne.
- ✓ Un accident peut être dû à une cause naturelle. Si c'est le cas, une seule cause naturelle sera à l'origine de l'accident. Une cause naturelle peut être à l'origine ou pas d'un accident.
- ✓ Un accident peut n'occasionner aucune lésion ou plusieurs lésions à plusieurs personnes. Il est possible qu'une lésion ne soit jamais reprise dans le cadre d'accidents. Un accident se déroule toujours à un et un seul endroit de l'usine.
- ✓ Dans un lieu de l'usine pourrait se dérouler plusieurs accidents au cours du temps, mais d'autres lieux pourraient ne jamais être concernés par un accident.
- ✓ Chaque accident se classe dans un type (et un seul).
- ✓ Un type d'accident peut concerner plusieurs accidents au cours du temps, mais ne pourrait en concerner aucun.

19. Les éditions LIVRIS

Les éditions LIVRIS effectuent le paiement des auteurs publiés chez elle une fois par an. Les auteurs sont identifiés par un numéro. Pour leur envoyer un chèque du montant dû, LIVRIS garde également leur nom et leur adresse. Un auteur est rémunéré par un pourcentage sur le prix total des ventes hors taxes de son livre, pourcentage allant de 10% à 15%. Lorsqu'un livre est écrit par plusieurs auteurs, le nombre d'auteurs est mentionné et la rémunération est partagée entre les différents auteurs. Les livres sont également identifiés par un numéro et sont de plus caractérisés par un titre et la quantité vendue au cours de l'année écoulée.

20. Référendums

L'arrondissement électoral de Bruxelles – Halle - Vilvoorde organise des référendums dans différents bureaux de vote. Vu qu'au maximum un seul référendum est effectué par jour, la date du référendum sert à identifier un référendum. Chaque référendum permet à la population de répondre à une seule question importante. La question posée doit être conservée par les autorités. Chaque bureau de vote de l'arrondissement comporte un numéro et un nom. Le numéro est unique par commune (il y a plusieurs bureaux de vote dans une commune). Chaque bureau de vote d'une même commune comporte un nom différent. Chaque commune possède également un n° national unique. C'est le conseil communal de chaque commune qui décide si un référendum proposé est susceptible d'intéresser ses concitoyens ou non. S'il pense que c'est le cas, la commune s'inscrit pour ce référendum. Quatre possibilités de résultats sont possibles : pour – contre – blanc et nul. Les résultats sont donnés par bureau de vote.

21. Club d'équitation

Un club d'équitation se compose des éléments suivants :

- Un cours d'équitation correspond à un niveau ;
- Un même cours est toujours donné par le même moniteur, toujours le même jour à la même heure : par exemple, le cours n°4 commence à 17h00 tous les lundis et est donné par M. Luke qui donne également d'autres cours ;
- Les cours sont caractérisés par un numéro (1, 2, 3...) ;
- Les moniteurs sont identifiés par leur nom ;
- Chaque membre du club possède une carte de membre comportant un numéro, un nom, une adresse, un code postal et une localité ;
- Un adhérent peut suivre des cours de niveaux différents ;
- A une date donnée, un adhérent suit un cours en montant un certain cheval : par exemples, M. Jamon, adhérent n°369, suit chaque lundi le cours n°4 de niveau "éperon d'argent"
- Pour faciliter l'attribution des chevaux, notamment aux nouveaux arrivants, on associe à chaque cheval un ou plusieurs chevaux regroupés selon des caractéristiques communes : pour ce rapprochement, la date de naissance, le sexe la race et la taille sont pris en compte. Cela permet, lorsqu'un cheval est souffrant, de pouvoir le remplacer par un autre cheval plus ou moins équivalent.

22.. Festival du film fantastique

Chaque année se déroule à Bruxelles le Festival du Film Fantastique. Afin d'attirer plus de monde encore, les organisateurs souhaitent installer un site Web informant le public sur :

- Les films (en 2004, 80 films différents numérotés de 1 à 80 étaient prévus) ;
- Les dates ;
- Les heures ;
- Les lieux (6 salles sont prévues : 4 au Passage 44, 2 au Nova) ;

L'affiche représentée ci-dessous est identique pour tous les films. Nous remarquons les choses suivantes :

- Pour éviter tout risque de perte des bobines pendant un transfert, un film est toujours donné au même endroit.
- Il n'y a qu'une seule représentation journalière pour un film.

- Toutes les données sont élémentaires et uniques (il n'y a jamais 2 acteurs ou 2 réalisateurs qui portent le même nom, par exemple).
- Le tarif d'un film est fonction de l'endroit et de la date (les films donnés au Nova sont moins chers car le confort est moindre, les films donnés l'après-midi sont moins chers pour aider les jeunes).
- Un acteur peut se retrouver dans plusieurs films de titres différents.
- Un film est réalisé par un seul réalisateur et la photographie est assurée par un seul photographe cinématographique.

PASSAGE 44 - 3	n° 16
LOST HIGHWAY De David Lynch	
Lundi 07 juin 20h30 Mardi 08 juin 18h00 Jeudi 10 juin 23h30	
Réalisateur : Photographie : Avec :	David Lynch Peter Deming Bill Pullman Patricia Arquette Balthazar Getty Tobert Loggia
Producteur :	Ciby 2000

23. Coopérative agro-alimentaire

Une coopérative agro-alimentaire regroupe différents acheteurs. Elle a comme activité l'achat de céréales et la revente en dégageant une marge bénéficiaire. Entre l'achat et la revente, elle stocke les différentes céréales (blé, orge, maïs...) dans dix-sept silos. Chaque silo est identifié par une lettre et ne contient bien sûr qu'une seule variété de céréale. Pour connaître les quantités stockées de chaque céréale, il peut être nécessaire d'inventorier plusieurs silos. Chaque variété de céréales achetée par un membre de la coopérative fait l'objet d'un contrat séparé. Chaque contrat comporte un numéro unique et mentionne les différentes quantités qui doivent être livrées à des dates d'échéance précises. Par exemple, Monsieur Bonvin, membre N° 7695 de la coopérative a signé le contrat 16442 prévoyant les livraisons suivantes de maïs : 200 kg le 05/10/2004 et 450 kg le 17/10/2004. Vous êtes chargé d'informatiser la gestion des contrats.

24. Fromagerie

Une fromagerie souhaite gérer au mieux la fabrication et la livraison de ses fromages à des clients principalement épiciers. Les fromages ont un code, une désignation, un prix et un type. Par exemple, "le Montagnard" vendu 1,2 € est du type "chèvre" alors que le "Brebicentenaire" vendu au prix de 1,4 € est un "brebis". Les épiciers sont connus de la fromagerie sous un numéro d'identification avec leur nom et leur adresse. Les épiciers déterminent à l'avance les quantités dont ils ont besoin pour un jour précis. Ainsi 20 "Montagnards" sont livrés chaque mardi à Mr Dubois, 10 "Montagnards" lui sont également livrés chaque jeudi. Les quantités à fabriquer sont alors déterminées pour une

semaine de l'année. Pour la 50^{ème} semaine, 156 "Brebicentenaires" ont été fabriqués. Pour chaque semaine également, les quantités de lait nécessaires à la fabrication des types de fromage sont commandées par la fromagerie. Par exemple, 30000 litres de lait de vache ont été commandés à la 50^{ème} semaine.

25. Achat d'articles

Un article en stock, caractérisé par un n° d'article, un libellé et un prix unitaire, est acheté en une certaine quantité par un client, caractérisé par un code client et un nom.

26. Club de karaté

La ligue Borraine de karaté organise tout au long de l'année des compétitions "kata" dans différents clubs qui lui sont affiliés. Ces compétitions consistent en démonstrations où chaque compétiteur montre son talent dans un enchaînement de mouvements appelé "kata".

Pour une compétition donnée, organisée par un seul club, tous les compétiteurs présentent le même "kata".

- Un "kata" est désigné par un nom. Chaque compétition de "kata" a un numéro d'ordre chronologique ;
- Pour participer à une compétition en tant que compétiteur, il faut être membre d'un club ;
- Pour être membre d'un club, il faut être titulaire d'une licence de membre. Un membre ne peut s'inscrire que dans un seul club. Chaque club possède un numéro et il exerce ses activités dans un lieu précis qui ne change pas. Les clubs sont animés et dirigés par un ou plusieurs entraîneurs ;
- Un entraîneur est titulaire d'une licence professionnelle d'entraîneur. Un entraîneur n'est pas membre d'un club : Il peut exercer son activité dans plusieurs clubs. Il est classé selon son niveau de compétence technique.
- Les membres d'un club ne participent pas tous aux compétitions ;
- Le "kata" présenté par le compétiteur est noté par 5 juges, chaque juge donnant une note sur 10. Les juges sont choisis parmi les entraîneurs pour une compétition et ils sont numérotés de 1 à 5. A la fin de la compétition, chaque compétiteur est classé par rang en fonction du total corrigé. Ce total est ajouté au total déjà obtenu pour les autres compétitions.

27. Société de location d'emplacements de parking

La société LOCABOX est une entreprise spécialisée dans la gestion des emplacements de parking (i.e. des boxes fermés ou ouverts) situés dans le sous-sol de plusieurs immeubles de diverses villes. Dans le cadre de la gestion des boxes, elle désire conserver une trace des informations relatives aux boxes des locations, présents et passés (pour les archives de la société LOCABOX).

Elle est amenée à utiliser les propriétés suivantes :

Propriété	Signification
ADR_IMM	Libellé de l'adresse de l'immeuble concerné
CHA_LOC	Montant variable des charges de la location d'un box
CP_IMM	Code Postal de l'immeuble concerné
DAT_LOC	Date de début de la location
DUR_LOC	Durée de la location (en jours)
LOY_BOX	Montant fixe du loyer pour un box
NOM_LOC	Nom du locataire du box
NUM_BOX	N° de matricule d'un box au sein de LOCABOX
NUM_IMM	N° de matricule d'un immeuble au sein de LOCABOX
NUM_LOC	N° de matricule d'un locataire au sein de LOCABOX
NUM_VEH	N° d'immatriculation du véhicule occupant le box
PRE_LOC	Prénom du locataire
VIL_IMM	Ville dans laquelle se trouve l'immeuble

La société LOCABOX utilise:

- NUM_BOX pour identifier un box parmi tous ceux qu'elle possède ;
- NUM_IMM pour identifier un immeuble qui la concerne ;
- NUM_LOC pour identifier un de ses locataires ;
- NUM_VEH pour identifier un véhicule parmi tous ceux enregistrés.

Les principales règles de gestion sont les suivantes :

- Un même locataire (habitant ou non l'immeuble) peut louer plusieurs boxes différents sur une même période. De même il peut louer le même box sur plusieurs périodes différentes (ce qui est très souvent le cas) ;
- LOCABOX exige de connaître le numéro d'immatriculation du véhicule (appartenant ou non au locataire) garé habituellement dans un box. Elle accepte un maximum de 2 véhicules pouvant se garer alternativement dans un box donné. Un même véhicule est autorisé à se garer dans plusieurs boxes différents ;
- Un locataire peut éventuellement louer un box sans y garer de véhicule.

28. Société de location de voiture

Vous êtes chargé d'informatiser la gestion de la société CARWORLD, société internationale de location de véhicules. Cette société regroupe plusieurs sociétés réparties dans plusieurs pays.

Cette société dispose d'un parc de véhicules de différentes marques et modèles, chaque véhicule étant sous la responsabilité d'une seule agence (p. ex., le véhicule immatriculé ASP034 dépend de

l'agence EUBEBX, agence se trouvant à Bruxelles, est de marque Renault et de modèle "Mégane TDI").

Chaque véhicule appartient à un type de véhicule (il y en a six actuellement : camion, utilitaire, tourisme, 4x4, moto, scooter) et est classé dans une catégorie (p. ex., dans le type tourisme existent cinq catégories : "économique", "standard", "moyenne", "haut de gamme", "luxe" ; dans le type utilitaire, les catégories "haut de gamme" et "luxe" n'existent pas).

Le tarif de location journalier dépend du type de véhicule, de la catégorie et de l'agence où s'effectue la location. Un client peut s'adresser à une agence de la Société et louer un véhicule disponible dans cette agence. Toutefois, il peut, s'il le souhaite, restituer le véhicule dans une autre agence de son choix (p. ex., quelqu'un peut louer une voiture à l'agence EUBEBX et la restituer à l'agence de Rotterdam EUNLRO).

Lors d'une location, l'agence de départ établit un contrat avec le client où sont stipulés entre autres la date de début de location, la durée prévue, le prix journalier convenu (le prix peut parfois être inférieur au prix tarif), le montant de l'acompte versé... L'agence note l'identification du véhicule loué, le nombre de Km figurant au compteur et le n° du permis du client.

Lors du retour du véhicule, on note l'agence de destination, le nombre de Km au compteur (la différence permet de calculer les Km parcourus). Les véhicules étant fournis avec le plein de carburant, le client est tenu de remettre le véhicule dans le même état. Si le plein n'est pas fait, l'agence fait le plein et note le montant pour facturer le client. Au terme de la location, l'agence éditte une facture contenant le montant dû qu'elle remettra au client.

Il est prévu une informatique centralisée, chaque agence étant reliée au Siège Central par une liaison spécialisée. La future base de données devra permettre de :

- Gérer la location des véhicules ;
- Gérer le parc des véhicules (âge, nombre de Km, état...) ;
- Établir la facture du client ;
- Établir des statistiques (Km moyen parcouru par agence, par client,... nombre de location par agence par mois, par jour...)

Parmi les différents attributs repris sur votre schéma devront se retrouver au moins les attributs suivants:

Propriété	Signification
NUM_IMM	N° immatriculation du véhicule
NUM_PER_CLI	N° de permis du client
NOM_CLI	Nom du client
PRE_CLI	Prénom du client
ADR_CLI	Adresse du client
VIL_CLI	Ville du client
CP_LI	Code postal du client
PAY_CLI	Pays du client
TYPE_VEH	Type du véhicule
CODE_CAT	Code catégorie du véhicule
LIB_CAT	Libellé catégorie
PUISS_VEH	Puissance véhicule
MAR_VEH	Marque du véhicule

LIB_MOD	Libellé du modèle
DATE_ACHAT	Date achat du véhicule
KM_COMPPT	Km au compteur
NUM_CON	N° de contrat
DATE_CON	Date du contrat
PRIX_JOUR_C	Prix de location par journée (prix prévu)
DATE_DEP	Date départ en location
DATE_RET_P	Date prévue de retour du véhicule
DATE_RET	Date de retour du véhicule
KM_DEP	Km au départ en location
KM_RET	Km au retour de location
NUM_FAC	N° de facture
DATE_FAC	Date de facturation
TOT_FAC	Total de la facture
COD_AGEN	Code agence
ADR_AGEN	Adresse agence
VILLE_AGEN	Ville agence
CP_AGEN	Code postal agence
PRIX_JOUR	Tarif journalier
TOT_CARB	Montant de carburant à facturer au client

29. Clinique médicale

Vous devez informatiser la gestion d'une clinique médicale. Après avoir complété votre analyse, vous avez recueilli les renseignements suivants :

- Chaque patient de la clinique a son médecin habituel. Le patient peut cependant être reçu par un autre médecin en cas d'urgence.
- A chaque visite du patient, le médecin qui le reçoit inscrit une entrée dans son dossier. Cette entrée contient la date et le but de la visite, le diagnostic et au besoin le traitement recommandé.

30. Prêt de livres

Vous devez développer un système qui permettra la gestion des prêts de livres d'une bibliothèque. Ce système doit permettre de connaître le fournisseur de chaque volume acheté.

Les abonnés de la bibliothèque possèdent chacun une carte de membre. Ils peuvent emprunter jusqu'à cinq livres en même temps pour une période de temps déterminée par la bibliothèque.

Les livres sont placés dans des sections numérotées. Ces sections possèdent un emplacement spécifique (rez-de-chaussée, deuxième étage, etc.) et une spécialité particulière (roman, biographie, histoire, etc.).

La bibliothèque achète ses livres chez différents fournisseurs. Il est à noter que deux livres identiques peuvent être achetés chez deux fournisseurs différents. Il peut donc y avoir plusieurs exemplaires d'un même livre.

31. Chaîne de magasins de fruits et légumes

Vous devez créer un système de gestion informatisée pour une chaîne de magasins de fruits et de légumes.

La chaîne possède neuf magasins répartis dans trois régions. Chaque région a son code unique de deux lettres et chaque succursale possède un numéro unique.

Un des employés de chaque succursale est le gérant de la succursale. Chaque région possède un directeur. Cependant, la succursale située sur la rue Rockland à Verdun n'a pas de gérant car celui-ci a quitté la compagnie. Le plus vieux magasin est celui de la rue Peel à Montréal car il a ouvert le 1^{er} septembre 1968.

A la fin de la journée, chaque succursale rend compte du nombre et du type de chaque produit vendu. Chaque produit (fruits et légumes) a un numéro de produit, une description, un prix et une unité de vente (kilo, unité, sac, douzaine, etc.)

Chaque succursale peut modifier son inventaire selon les quantités de marchandises reçues. Le système central peut voir l'inventaire de toutes les succursales et savoir pour chaque succursale quelles sont la quantité actuelle et la quantité requise de façon à acheminer les bons produits à temps. La quantité requise peut être différente d'une succursale à l'autre et d'un produit à l'autre.

32. Société de location immobilière

Un organisme exploite une entreprise de location de locaux (appartement, espace de stationnement, local commercial). Chaque local appartient à un groupe d'immeubles et un groupe d'immeubles peut posséder plusieurs locaux.

Un locataire peut louer un ou plusieurs locaux. Un local peut ne pas être loué ou il peut être loué par un seul locataire.

L'échéance du loyer est mensuelle. A chaque échéance, un montant de location est calculé pour chaque local loué par un locataire.

Les locataires en retard de paiement font l'objet d'étapes successives de poursuite. L'entreprise veut savoir pour chaque poursuite quel loyer n'a pas été payé par quel locataire et pour quel local. Chaque loyer non payé entraîne une poursuite composée de plusieurs étapes.

L'entreprise est donc intéressée par le suivi de la location des locaux ainsi que des étapes de poursuite. Elle veut connaître :

- Les coordonnées (nom, adresse et téléphone) des locataires et des propriétaires des locaux.
- L'adresse et le type de chaque local.
- Le montant du loyer pour chaque local, la date d'échéance et la date du paiement.
- La date de début, la date de fin et le résultat de la poursuite.
- La date, la description et le résultat de chaque étape de poursuite.

33. Société financière

La Compagnie Agria vend des produits financiers à ses clients. Dans le système actuel, chaque client est identifié par son numéro d'assurance sociale. On conserve pour chaque client son nom, son adresse, son numéro de téléphone et le nom de son employeur. Chaque produit financier est

identifié par le numéro d'assurance sociale du client auquel s'ajoute un numéro de séquence généré pour distinguer les différents produits du même client. Un produit financier est toujours vendu à un et à un seul client. On distingue deux sortes de produits : les assurances vie (AV) et les régimes enregistrés d'épargne retraite (RÉER). Les informations à conserver pour les RÉER sont la date d'achat, le taux d'intérêt, le montant initial et le terme. Pour les AV, on conserve la date d'achat, la prime mensuelle et le montant assuré. On conserve aussi pour chaque AV le numéro d'assurance sociale, le nom, l'adresse et le numéro de téléphone du bénéficiaire désigné. Les produits sont vendus par des vendeurs à commissions. On veut donc pouvoir retracer le vendeur de chaque produit vendu afin de calculer les commissions. Chaque vendeur est identifié par son numéro d'assurance sociale. On conserve pour chaque vendeur son nom, son adresse, son numéro de téléphone et un code identifiant la catégorie du vendeur.

34. Collège

Dans un collège, chaque étudiant inscrit fait partie d'un groupe. Chaque groupe suit différents cours. L'étudiant reçoit une note pour chaque cours auquel il est inscrit. Le collège engage des professeurs pour donner un ou plusieurs cours. Un cours peut être donné par plusieurs professeurs mais un groupe qui suit un cours n'a qu'un seul professeur et n'utilise qu'un seul local par cours. Chaque local correspond à un type particulier.

D'après l'étude des besoins, l'analyste a établi ainsi les informations que le collège aimerait obtenir :

- Le nom de l'étudiant, son adresse, son téléphone, son code permanent, son sexe, sa date de naissance, sa note et le nombre d'absences pour chacun des cours.
- La date de création du groupe, la date à laquelle le groupe termine le programme et le programme auquel il est inscrit.
- La description du cours, le nombre d'heures, le nombre de périodes théoriques, le nombre de périodes de laboratoire, le nombre de semaines requis, le nombre de crédits et le type de local requis.
- Le nom du professeur, son adresse, son téléphone, sa compétence et sa préférence pour chaque cours qu'il donne.
- La description du local, son emplacement et son type.
- Pour chaque groupe et pour chaque cours, la date de début du cours, la date de fin du cours, le numéro du local et le numéro du professeur.

35. Société organisatrice de spectacle

Le centre d'art Hérisson présente des spectacles depuis maintenant dix ans. Le groupe de direction du centre accorde une attention particulière à la sélection des spectacles et à la définition des plans d'abonnement au début d'une nouvelle année. La stratégie de la direction est d'offrir des spectacles de types différents pour rejoindre une grande partie de la population et d'augmenter le nombre d'abonnements annuels. Le centre détient un bon réseau de points de vente de billets répartis aux points stratégiques de la ville et ses banlieues. Pour les abonnements, les billets sont réservés, imprimés et envoyés au début de l'année.

Le groupe de direction du centre d'art doit choisir les spectacles au début de chaque nouvelle année. Le même spectacle peut être offert par des agences différentes à des prix différents. Lorsqu'il y a entente avec une agence de spectacle, un contrat est signé. Ce contrat couvre l'ensemble des représentations du spectacle vendu par l'agence de spectacle. La direction doit donc décider pour chaque spectacle choisi le nombre de représentations et leurs dates.

L'entreprise veut faire un suivi de tous les spectacles et des agences qui les offrent, du nombre de billets vendus pour chaque point de vente ainsi que des clients qui achètent soit des billets, soit des plans d'abonnement. Elle veut être capable d'imprimer les billets, les contrats et un programme mensuel indiquant toutes les représentations du mois.

- Sur le billet, elle veut imprimer le nom de l'agence, l'endroit de la représentation, le nom et le type de spectacle, la date, l'heure, le nom de la section (parterre, balcon, etc.), les numéros de siège et de rangée et le prix du billet.
- Sur le contrat, elle veut avoir le numéro du contrat, la date du contrat, le nom de l'agence avec laquelle elle a signé, le type du spectacle et la date de chacune des représentations du spectacle.

36. Organisation des élections au Canada

On veut concevoir une application de gestion des données relatives au processus électoral. On suppose que les données ne portent que sur une seule élection. Tous les citoyens éligibles doivent être recensés. On veut pouvoir retrouver pour chaque citoyen son numéro d'assurance sociale, son nom, son numéro de téléphone et le pôle électoral où il doit exercer son droit de vote (un et un seul par citoyen). La carte électorale subdivise le territoire en comtés et chaque comté est subdivisé en pôles.

Dans le système actuel, un comté est identifié par son nom et un pôle électoral est identifié par le nom du comté auquel on ajoute un numéro de pôle généré pour distinguer les différents pôles du même comté. Le même numéro de pôle peut être utilisé dans des comtés différents.

On veut pouvoir retrouver l'adresse de chaque pôle. Chaque citoyen vote pour un candidat au poste de député de son comté. Chaque comté possède un ensemble de candidats au poste de député. Pour être candidat, un citoyen doit représenter un parti officiellement reconnu. Un candidat ne peut pas se présenter dans deux comtés. On veut pouvoir retrouver pour chaque candidat, son numéro d'assurance sociale, son nom, son numéro de téléphone, son âge, son sexe, son parti et le comté où il se présente au poste de député. On veut aussi conserver l'information suivante au sujet des partis : le nom du parti, l'adresse du local officiel et le numéro de téléphone. Pour chaque parti, on veut pouvoir connaître le chef qui doit être un candidat officiellement reconnu. Pour le chef de chaque parti, on veut conserver en plus des informations relatives à son statut de candidat au poste de député, la date de son élection en tant que chef du parti.

Chaque citoyen peut faire une ou plusieurs contributions monétaires à un ou à plusieurs partis. Pour chaque citoyen, on veut conserver s'il y a lieu le montant total qu'il a donné à chaque parti.

37. Compagnie de téléphone Canadienne

Etablie depuis 1990, la compagnie de téléphone Miles Inc. s'impose maintenant comme un des chefs de file dans le domaine des télécommunications.

Les abonnés de Miles paient un tarif mensuel de base qui varie selon la région où ils habitent. Le territoire de l'Amérique du Nord a donc été découpé en régions précises (exemple : Montréal métropolitain, Prairies Nord, Terre-Neuve Ouest, etc.). A chaque région correspond un tarif de base précis qui sert à défrayer le coût des appels locaux (à l'intérieur de la région).

A ce tarif de base s'ajoutent les frais de communications interurbaines. Le montant facturé pour une communication interurbaine dépend du taux et du nombre de minutes (exemple : 5 minutes à 2,75 \$/min). Le taux dépend de la région de la personne qui appelle et de la région de la personne appelée (exemple : une communication entre Montréal métropolitain et Terre-Neuve Ouest coûte 1,95 \$/min.). Le prix dépend aussi du type d'appel. Les appels transmis par le central ou les appels facturés à la Carte Internationale Miles coûtent 20% plus cher. Les appels de personne à personne entre 18:00 et 06:00 coûtent 30% moins cher. Les appels acheminés vers l'Europe ou l'Amérique du Sud ont un taux fixe à 7,50 \$/min.. Les appels vers l'Asie et l'Océanie ont un taux fixe de 12,50 \$/min. Pour chaque appel interurbain, il faut donc connaître la région d'où émane l'appel, la région appelée, l'heure et le type d'appel (par le central, par la carte d'appel ou de personne à personne).

Au tarif de base et aux frais de communications interurbaines s'ajoutent les tarifs des services spéciaux. Les clients ont la possibilité de s'abonner à des services spéciaux tels que la mise en attente, le service de répondeur téléphonique informatisé, le renvoi d'appel, la téléconférence et l'afficheur d'appel. Chaque service spécial possède un taux mensuel fixe (exemple : 9,00 \$/mois pour le service de téléconférence). Il est important de noter que chez Miles aucun frais de raccordement (ligne ou services spéciaux) n'est facturé au client. Le tarif de base, les communications interurbaines et les services spéciaux sont toutefois taxables.

Votre rôle consiste à établir un système informatique en mesure de faire le suivi et la facturation des services de Miles Inc.