Ecole Supérieure d'Agriculture d'Angers
55, rue Rabelais - 49000 Angers

Informatique

Les Bases de données

document d'informatique
< Les bases de données >

Edition complétée et remise à jour
Mai 2001

Didier Virion
Ingénieur ICAM - IEFSI

Plan Général

Première partie : De la théorie...
Chapitre 1 : Un peu d'histoire
Chapitre 2 : Les bases de données
Chapitre 3 : Conception d'une base de données

Seconde partie : ...à la pratique
Chapitre 1 : Le logiciel Microsoft Access
Chapitre 2 : Exemple d'utilisation
Chapitre 3 : D'autres conseils...

Complément : Information et Entreprise
Chapitre 1 : Information...
Chapitre 2 : ...et entreprise

En guise d'exercice : Un peu de réflexion...

Nous allons parler ici d'une forme particulière de traitement de l'information, encore que le terme de traitement soit mal adapté au cas présent. Il serait préférable de lui substituer celui de **gestion d'informations**, plus précis. C'est ce terme que l'on utilise dans l'expression « système de gestion de données », ou encore dans « système de gestion de base de données », en abrégé SGBD

En effet, par le terme traitement, on sous-entend souvent une transformation de l'information (comme dans un tableur, par exemple) alors que la gestion de données consiste essentiellement à structurer et à stocker des informations, afin de pouvoir retrouver rapidement celles dont on a besoin. La « transformation » de ces données relève d'un autre mode de traitement de l'information : celui des tableurs, déjà cité, et surtout celui des logiciels de calcul, comme par exemple les logiciels dédiés aux calculs statistiques.

Notez déjà ici ce que nous répéterons plus loin : il n'est pas question d'étudier la manière dont les informations sont réparties physiquement sur leur support. Nous nous plaçons délibérément du coté de l'utilisateur final, celui qui veut pouvoir se servir d'une base de données. Peu lui importe la « cuisine interne » du logiciel. Ce qui l'intéresse au premier chef, c'est de voir apparaître ce dont il a besoin sur son moniteur, ou sur son imprimante. C'est dans ce sens qu'à été conçue la seconde partie du document.

Cela n'empêche pas qu'il est utile de connaître certains rudiments de cette « cuisine », pour y ajouter à bon escient la pincée de sel ou la pointe d'ail qui donne toute sa saveur au travail réalisé. C'est là le but de la première partie du document.

Ce document se termine par un « complément » qui ne fait pas directement partie du programme de cette année, mais qui peut ouvrir des voies de réflexion à ceux qui prendront le temps de le lire attentivement.

Première partie : De la théorie...

Chapitre 1 : Un peu d'histoire

1.1.0 - Très brève introduction
1.1.1 - L'ancêtre, sur micro-ordinateur...
1.1.2 - Paradox entre en scène
1.1.3 - La chute d'Ashton-Tate
1.1.4 - Borland acquiert Ashton-Tate, Microsoft rachète Fox
1.1.5 - Le standard dBase
1.1.6 - Le boum des SGBD graphiques
1.1.7 - En guise de conclusion...

Chapitre 2 : Les Bases de Données...

1.2.1 - Les bases de données
1.2.2 - Qu'est-ce que le modèle relationnel ?
1.2.3 - Les 12 règles de Codd

Chapitre 3 : Conception d'une base

1.3.1 - Intérêt d'une bonne étude...
1.3.2 - Création d'une base de données
1.3.3 - Quelques règles simples
1.3.4 - Exemple

Chapitre 1 : Un peu d'histoire

Les quelques pages qui suivent ont pour but de tenter de vous donner une idée des bouleversements rapides dans le domaines des SGBD, et plus particulièrement des SGBD destinés aux micro-ordinateurs. En aucun cas, il ne s'agit d'être exhaustif. Bien au contraire, ce texte de « culture générale informatique » est limité à quelques « ténors » des années 80, jusque vers 1992/93, en prenant comme ligne directrice, le légendaire dBase...

Bien évidemment, comme chaque fois qu'il s'agit de retracer l'évolution d'un des nombreux aspects de l'informatique, le but n'est pas de vous faire un cours d'histoire des Sciences et Techniques, mais bien de vous permettre de mieux cerner l'évolution actuelle de ce domaine particulier.

1.1.0 -Très brève introduction

Il est possible de situer la naissance des premiers gestionnaires de fichiers vers 1960. En 1965, on découvre les premiers SGBD supportant une **structure hiérarchique**, ou navigationnelle, à l'aide de liens qui doivent être créés entre les fichiers-tables. Puis, en 1970, les **SGBD relationnels** se révèlent dans les bureaux d'études. Enfin, en 1980, les problèmes de volume et de rapidité d'exécution tendent à s'estomper : les systèmes d'exploitation UNIX, VMS et d'autres systèmes multitâches/multi-utilisateur, connectés sur réseaux, offrent de nombreuses solutions avec les notions de verrou, de sauvegarde, etc.

Depuis, les logiciels ont évolués, et chacune des sociétés dont nous allons citer les noms a connu des hauts et des bas, parfois même a disparu sous la pression des événements, ou à la suite des erreurs de ses dirigeants. Un des cas les plus remarquables, et les plus intéressants à tout point de vue, est celui des logiciels de la famille dBase. Il va d'ailleurs nous servir de fil conducteur...

Le 17 Mars 1988, 4000 personnes attendent dans une rue de Paris, près de la Maison de la Chimie. Elles sont parfois venues de loin pour assister à la première présentation française d'un certain **dBase IV**. Néanmoins, malgré une seconde séance improvisée, plusieurs centaines d'entre elles ne parviendront pas à s'introduire dans la salle réservée au show dBase. Hugues Leblanc, président de LCE et importateur du logiciel jubile. Moins d'un an après, l'enthousiasme est retombé. Le logiciel dBase IV, après un retard mémorable, s'est finalement montré au public qui s'attendait à mieux. Ashton-Tate, son éditeur, traverse des heures noires et n'arrivera pas à redresser la barre. Grandeur et décadence...

1.1.1 - L'ancêtre, sur micro-ordinateur...

Le **langage dBase** avait été originellement écrit par Wayne Ratliff pendant son temps libre vers la fin des années 70. L'ambition première de cet employé de la NASA est modeste : il veut pouvoir améliorer ses pronostics sur les matchs de football, en utilisant les résultats des matchs précédents, équipe par équipe. Pour cela, il développe un **langage de programmation** qui s'appuie sur une gestion de fichiers...

Aux alentours de 1980, deux membres d'une petite société d'édition (qui deviendra plus tard **Ashton-Tate**) se présentent à lui et proposent de prendre en main la distribution exclusive de dBase. Wayne Ratliff accepte. Il s'avère que ce logiciel répond à un besoin. La preuve en est qu'il rencontre immédiatement un succès important. En 1983, **Ashton-Tate** propose d'acquérir les droits de cette gestion de fichiers et sort **dBase II** pour différents systèmes, dont l'IBM PC[1]. Celui-ci constituera l'un de ses triomphes.

Le langage de programmation **dBase II** est suivi par **dBase III** en 1984, puis par **dBase III Plus**, à la fin de **1985**. Cette version consacre alors la complète domination d'Ashton-Tate sur le marché des gestionnaires de bases de données. Elle est plus conviviale que la précédente et peut exécuter des programmes en réseau (multi-utilisateur). Le logiciel dBase III Plus grimpe temporairement jusqu'à la première place des ventes américaines. **Vers 1986, Ashton-Tate détient 60% des parts de marché dans le domaine des SGBD**.

Pourtant, **les limitations de dBase III Plus** ne tardent pas à apparaître. Lorsqu'il s'agit de créer une relation (d'intégrer les données de plusieurs fichiers), il faut mettre en oeuvre une programmation complexe. L'exécution en réseau présente aussi des limitations : lorsqu'un utilisateur consulte un enregistrement, les autres ne peuvent y accéder. De plus ; les développeurs

[1] - On peut dire que dBase a été au PC ce que VisiCalc avait été à l'Apple II.

d'applications doivent recourir à des acrobaties pour écrire leurs programmes du fait que dBase III Plus ne peut ouvrir que quinze fichiers à la fois. Pour couronner le tout, les applications écrites sous cet environnement sont fort lentes.

1.1.2 - Paradox entre en scène

Une telle situation ne peut qu'encourager l'éclosion d'une âpre concurrence. Plusieurs gestions de base de données apparaissent : **R:Base** de Microrim, **DataEase** de la société du même nom... Des programmes compatibles dBase voient le jour et se distinguent notamment par une rapidité bien supérieure : **DBXL**, **Nantucket**, **FoxPro**. Mais de tous les systèmes de gestion de base de données, c'est le dénommé **Paradox** qui occupe alors le devant de la scène.

Il est l'oeuvre de deux informaticiens appelés **Richard Schwartz** et **Robert Shostak**. Ceux-ci ont passé plusieurs années à concevoir leur logiciel en travaillant lors de leurs soirées et week-ends. Ils sont partis de l'idée qu'il pouvait être judicieux d'appliquer des techniques d'intelligence artificielle au domaine des SGBD. En 1984, le projet des deux associés a la chance de séduire Ben Rosen, investisseur en capital-risque dont les précédents coups de maître concernent Lotus et Compaq. Ben Rosen, qui avoue fonctionner à l'enthousiasme, craque littéralement pour Paradox et aide au lancement d'Ansa, la compagnie de Schwartz et Shostak. Paradox se distingue de dBase III Plus par le fait qu'il est possible de se servir de ce produit sans jamais avoir à écrire une seule ligne de programmation. Il est donc plus facilement accessible aux utilisateurs « non spécialistes ».

L'un de ses atouts se trouve dans la fonction appelée QBE ou **Query By Example** (consultation par l'exemple). Grâce au QBE, l'utilisateur se contente d'indiquer ce qu'il recherche dans une base de données, le logiciel se chargeant de lui restituer les informations. Par ailleurs, deux ou plusieurs fichiers peuvent être mis en relation sans programmation. On peut facilement « **faire du relationnel** » sur des micro-ordinateurs...

Dès sa sortie, en 1985, Paradox s'impose. La presse spécialisée reconnaît en lui le véritable rival de dBase Ill Plus. Plusieurs comparatifs le placent en tête des SGBD en termes de performances globales. Séduites, de nombreuses grandes entreprises américaines, telles que Disney ou Elf Aquitaine Petroleum, adoptent immédiatement Paradox, des entreprises françaises également (Peugeot). Dès mars 1986, Ansa devient une compagnie rentable. En mai 1987, elle va plus loin et sort une **version réseau de Paradox** qui accroît un peu plus la distance qui le sépare de dBase III Plus. Lorsque plusieurs utilisateurs consultent un même fichier, les modifications apportées à partir d'un poste apparaissent quasi immédiatement sur les autres écrans (grâce à une technique dite de « rafraîchissement automatique »). En 1987, **Philippe Kahn**, fondateur de la société Borland, estime qu'il lui faut étendre sa gamme en intégrant un logiciel de plus grande envergure. Il a connu une énorme réussite avec le lancement de produits bon marché tels que Sidekick (utilitaires de bureau) ou Turbo Pascal (langage de développement). Le président de **Borland rachète donc Paradox** en juillet de cette même année.

1.1.3 - La chute d'Ashton-Tate

En 1988, chez Ashton-Tate, l'atmosphère reste bonne. A défaut d'être le plus performant des SGBD, dBase III Plus demeure indétrônable. Il faut dire qu'il a attiré à lui des dizaines de milliers de programmeurs qui affirment trouver leur bonheur dans cet environnement prolixe. On trouve plusieurs centaines d'outils capables d'améliorer la programmation sous dBase. Si besoin est, le compilateur **Clipper** de Nantucket ou le très rapide **FoxPro**, permettent de donner du tonus à un programme dBase III Plus qui, sinon, agirait avec trop de lenteur. De toute façon,

Ashton-Tate détient sa réponse aux critiques avec ce nouveau dBase IV qui doit officiellement sortir en juillet 1988.

Ed Esber, qui a pris la direction des affaires à la fin de 1984, partage le pouvoir avec un certain Luther Nussbaum. Problème : ce dernier n'est pas du tout apprécié à l'intérieur de sa société. Tandis que le premier se plaît à voyager fréquemment, à donner des conférences et, occasionnellement, à absorber d'autres compagnies (Multimate, Decision Ressources...), le second gère la société. Il acquiert la réputation d'un individu intimidant et irritant. Voilà qui n'est pas de nature à insuffler de la motivation aux troupes de programmeurs, considérés comme quantité négligeable. Wayne Ratliff se fâche et quitte la société. Ed Esber ne parviendra pas à lui trouver un remplaçant ayant la même qualité de vision.

Les **problèmes de dBase IV** apparaissent rapidement au grand jour. Les programmeurs s'escriment à faire tenir les 500 000 lignes de code (quatre fois la taille de son prédécesseur) dans les 640 Ko du DOS. La solution trouvée consiste naturellement à découper le programme en plusieurs parties qui sont chargées depuis le disque dur chaque fois que cela se révèle nécessaire. Seulement, une telle procédure ralentit l'exécution. Plus généralement, la difficulté du développement amène Ashton-Tate à repousser la date de sortie du produit. Lorsqu'il est enfin publié, en octobre 1988[2], il fait l'objet d'un accueil extrêmement mitigé : plusieurs magazines font état de bogues rendant difficile l'utilisation de dBase IV pour des développements importants. A la fin de cette même année, une estimation d'lDC (portant sur le nombre d'unités livrées) permet de découvrir que les parts de marché d'Ashton-Tate en matière de bases de données ont accusé une chute importante : 45% en 1988 contre 56% en 1987.

La **situation devient beaucoup plus malaisée** pour Ashton-Tate au cours de l'année 1989. En janvier, la compagnie finit par admettre officiellement l'existence des bogues et annonce l'écriture d'une version 1.1 destinée à les corriger. Luther Nussbaum "démissionne" quelques mois plus tard, sous des pressions aussi bien internes qu'externes.

Entre-temps, Ashton-Tate se distingue en intentant **un procès à Fox Software**. L'histoire est assez compliquée, et remonte à deux ans auparavant. Au début de l'année 1987, Ed Esber s'intéressait fortement à la société Fox Software et envisageait son acquisition. En effet, Fox publiait un SGBD compatible dBase III Plus, mais dix fois plus rapide que celui-ci. Les ingénieurs d'Ashton-Tate après avoir étudié la technologie mise au point par Fox arrivèrent à la conclusion qu'il serait bénéfique de l'intégrer dans les futurs SGBD d'Ashton-Tate. David Fulton, président de Fox Software, en demande alors 10 millions de dollars[3]. Ashton-Tate choisissant le parti du marchandage, **le haut management, qui n'est aucunement averti de l'aspect technique des logiciels, préfère finalement se passer de la technologie de Fox**.

Au début de 1989, Ed Esber prend la mouche, les performances de FoxPro éclipsent celles de dBase III Plus à un point tel que cela en devient gênant. Faute d'autre possibilité, il choisit la voie judiciaire et intente un procès à Fox Software, arguant que le langage de programmation employé s'inspirait trop ostensiblement de celui de dBase. A la fin de cette même année, la compagnie déclare des pertes de plus de 25 millions de dollars. L'ancien n°3 du logiciel termine son année avec 300 millions de dollars de chiffre d'affaires, tandis que le nouveau n° 1, Microsoft, s'envole vers son premier milliard...

La version 1.1 de dBase IV ne voit finalement le jour qu'en juillet 1990. Elle est jugée stable par les journalistes et par les consultants. Mais l'aura du SGBD leader de naguère est bel et bien tombée. Dans certaines listes de vente de logiciels, **Paradox** de Borland apparaît devant dBase IV et s'impose progressivement comme le nouveau numéro l des SGBD.

[2] - Avec seulement quatre mois de retard. Depuis, d'autres sociétés ont fait beaucoup mieux, si on peut dire, en accumulant jusqu'à deux années de retard.

[3] - A voir les autres transactions évoquées plus loin, on a envie d'ajouter « seulement »...

1.1.4 - Borland acquiert Ashton-Tate, Microsoft rachète Fox

Pourtant, dans l'ombre, Borland prépare un « coup » majeur, en juillet 1991, il annonce officiellement son intention de racheter Ashton-Tate. Philippe Kahn explique sa décision : "*Cela faisait un an et demi que nous étions en discussion avec eux. En interne, nous avions conçu un produit (le futur dBase pour Windows). Nous avons rencontré les gens d'Ashton Tate pour le leur présenter. Nous savions qu'ils étaient fort mal-en-point dans leurs propres développements. Ils ont été impressionnés par ce qu'ils ont vu. Notre produit apparaissait comme la continuation naturelle de leur dBase IV. Ashton-Tate a donc convenu de nous passer le relais*". Un mois plus tard, à la suite d'un échange d'actions portant sur 440 millions de dollars, la compagnie de Philippe Kahn absorbait son concurrent d'hier. L'une des premières décisions prises par Borland sera d'annuler le procès intenté par Ashton-Tate à Fox Software. Commentaire de Philippe Kahn : *"Nous nous battons sur le terrain de la technique et non pas sur celui des cours de justice"*.

En 1992, dans le monde des **logiciels pour compatibles PC**, **Borland contrôle environ 75 % du marché des bases de données**[4] . Il se prépare à offrir plusieurs mises à jour de dBase IV (une version DOS, dBase IV 1.2 et une version Windows, dBase IV 2), ce qui pourrait lui permettre de toucher 4 millions d'utilisateurs. Selon son président, dBase et Paradox vont même pouvoir coexister en harmonie. *"L'utilisateur de dBase aura la possibilité de manipuler les données de Paradox, et réciproquement"* Borland avait toutefois l'intention de privilégier Paradox plutôt que dBase.

Pourtant, cette situation paradisiaque a été de courte durée. En avril 1992, nouveau coup de théâtre : Microsoft annonce qu'il vient d'absorber Fox Software[5] . Pour prendre pied dans le domaine des SGBD, il était difficile d'opérer un choix plus judicieux : **FoxPro 2.0** a raflé à la fin de 1991 un nombre de prix considérables, allant jusqu'à être sacré meilleur logiciel de l'année. Un troisième challenger se profile, toujours dans l'univers dBase. Il s'agit de Computer Associates, éditeur bien implanté dans la mini-informatique et qui a successivement racheté les logiciels **dBFast** (clone de dBase sous Windows) et **Nantucket** (compilateur le plus populaire). Ce revirement de situation a obligé Borland à revoir ses priorités et à remettre dBase au premier plan.

1.1.5 - Le standard dBase

Les années 90 voient la concurrence se diversifier dans le domaine des SGBD. Des outils de développement en mode graphique, tels que **SQL Windows** de Gupta, ainsi que des propositions émanant d'éditeurs réputés dans le monde de la mini-informatique, comme **Ingres** ou **Oracle**, sont venus élargir l'offre. Pourtant, dBase devrait rester un standard important pour plusieurs raisons. Une présence vieille de plus de dix ans se traduit par des centaines de milliers d'applications développées à ce jour que les développeurs auront à coeur de faire évoluer. Son format de fichier (format dit **.DBF**) est reconnu par la très grande majorité des logiciels. La gamme des outils disponibles aux développeurs (logiciels annexes, magazines dédiés...) est incroyablement riche. Borland, Microsoft-FoxPro et Nantucket tentent de faire évoluer le langage (dont l'apprentissage s'avère fort simple) vers la programmation orientée objet. Il existait un risque de voir tous ces « **langages dBase** » apparaître de plus en plus divergents au fil de leurs versions réciproques. Mais cette situation semble appelée à se régulariser rapidement : un standard défini par des consultants indépendants a été retenu par l'ANSI en mai 1992. A l'époque, on estimait que la normalisation de ce langage devrait inciter tous les éditeurs à l'adopter.

[4] - La répartition en 1992 est de 42 % pour dBase et 33 % pour Paradox. Pour les principaux concurrents, les parts de ce marché sont de 6% pour Software Publishing (Superbase), et de 7 % pour Microsoft (Fox Pro)...

[5] - pour 173 millions de dollars...

1.1.6 - Le boum des SGBD graphiques

L'interface des logiciels évolue sans cesse pour le plus grand bien de l'utilisateur, qu'il soit novice ou expert. Avec la formidable percée des environnements graphiques et des langages de programmation événementielle, le mot-clef devient la convivialité. En bref, les éditeurs de SGBD rendent leurs produits plus intuitifs, pour l'utilisateur final comme pour le programmeur. Dans l'ensemble, le gain en puissance des machines et la standardisation des objets graphiques assurent une meilleure vitesse des traitements.

Avant que l'utilisateur puisse goûter aux délices des environnements graphiques, les éditeurs de SGBD ont longtemps négligé l'ergonomie de leurs produits. En effet, l'interface se voyait sacrifiée sur l'autel des performances (les techniciens parlaient aux techniciens). Aujourd'hui, il semble indispensable de s'adapter aux environnements graphiques, sous peine de disparaître. D'ailleurs, tous les grands ténors du domaine y sont venus, bon gré mal gré, avec plus ou moins de succès.

Sur PC, Windows rejoint peu à peu la convivialité qu'Apple avait instaurée au début des années 80 avec son système Macintosh. L'utilisateur peut de plus en plus souvent retrouver les mêmes applications dans les deux environnements : **MS.Works**, **ClarisWorks**, **Excel**, **FileMaker Pro**, **Omnis 7, 4°Dimension**... rien que pour la gestion de données. Ces applications, bâties autour d'un même noyau, sont compatibles d'un environnement à l'autre, aussi bien sur le plan des données que celui des développements.

1.1.7 - En guise de conclusion...

Attention, cette conclusion ne peut être que provisoire. Elle s'arrête au milieu des années 90, donc vers 1995/96, et elle n'a rien de définitif. Bien au contraire. En informatique, tout évolue très vite, trop vite même : la durée de vie d'un logiciel, entre deux révisions majeures, est souvent inférieure à deux années. Alors, ne comptez pas sur moi pour vous dire de quoi demain sera fait. Je ne suis pas Madame Soleil...

A partir de 1990, la gestion sécurisée de gros volumes d'informations amène progressivement les entreprises à rendre homogènes leurs bases de données en standardisant leurs interrogations, en particulier à l'aide de SQL. Parallèlement, la diffusion des PC et leur connexion en réseau permettent aux architectures client-serveur de se multiplier. La lourdeur des gros systèmes est ainsi bousculée par les acquis de la micro : des standards s'imposent et, même s'ils sont parfois critiquables, ils sont préférables à une foule de produits géniaux mais malheureusement non compatibles entre eux.

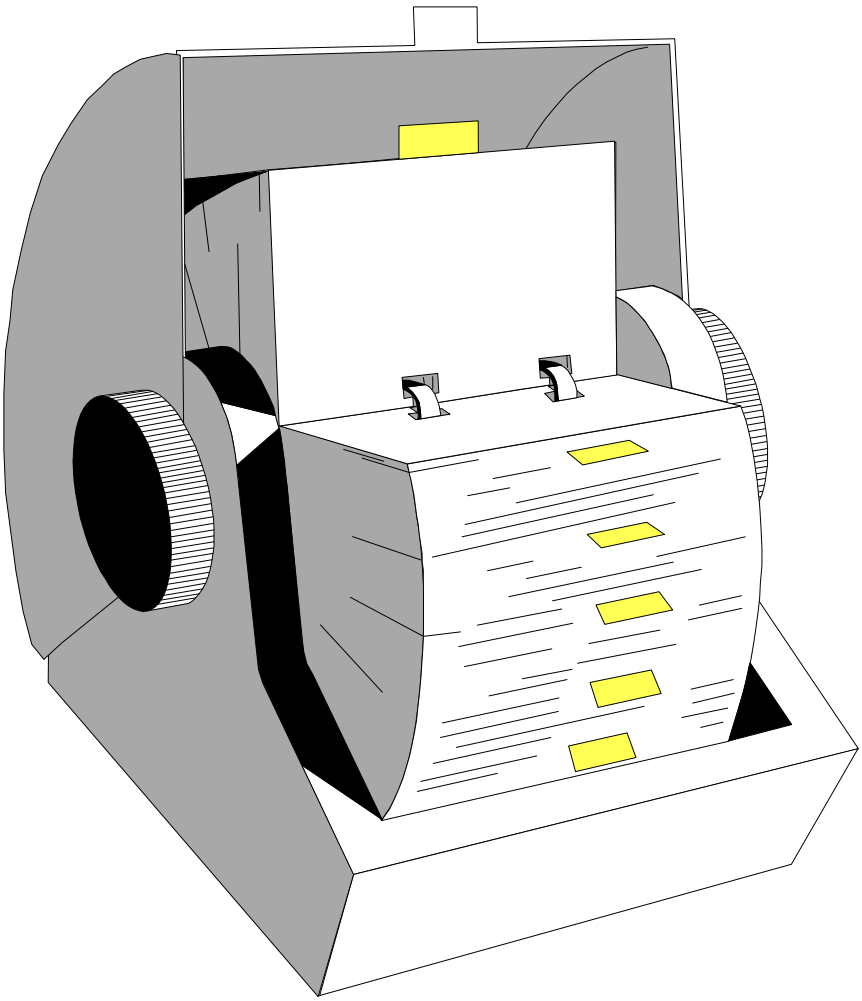
Enfin, avec la puissance accrue des machines et l'ouverture des bases de données, même les informations les plus gourmandes en ressources matérielles peuvent être rassemblées en structures logiques qu'il est possible de décentraliser.

L'amélioration des algorithmes de recherche ou de compression ainsi que les nouvelles possibilités de liaisons entre les applications (OLE, DAL et autre « **publication** », « **abonnement** » ou « **collage spécial avec liaison** ») ouvrent la voie aux bases de données « totales » ou, plus modestement, au support du multimédia.

A l'origine, les SGBD ne connaissaient que deux types d'informations : les caractères et les nombres. Aujourd'hui, la plupart des éditeurs incorporent dans leurs logiciels la possibilité de manipuler des images et du son. Avec ces outils, il devient possible de créer des bases de données « multimédia ». Parmi les nombreuses évolutions, il est possible d'imaginer une gestion de disques ou de cassettes dont la structure comporte des champs « images » (la pochette du dis-

que) et des champs « son » (les premières mesures d'un morceau). Ou encore une gestion de cave à vin permettant de visualiser les étiquettes des bouteilles... Il ne manque plus que le goût et l'odeur... Quoique, dans ce dernier domaine, des expériences (non concluantes) aient été tentées...

Gérer des sons est à présent aussi facile que du texte. Ils peuvent être déplacé d'un champ à l'autre presque aussi facilement qu'un mot. Il suffit d'importer (ou de coller) les sons, par exemple stockés dans des fichiers de type **.WAV**, dans le champ correspondant. Habituellement, sous Windows, cela se fait par l'intermédiaire de liaisons de type OLE qui « pointent » sur l'objet désiré, et chaque fois que l'on ouvre le document, le son se fait entendre...



Chapitre 2 : Les Bases de Données...

Souvent ignoré, parfois redouté car considéré comme un outil difficile à prendre en main, le gestionnaire de fichiers[6] est en fait une alternative simple au classement manuel d'informations et aux fiches cartonnées. Le tout est de bien définir ses besoins et d'être méticuleux lors de la réalisation.

Les logiciels de cette famille sont beaucoup plus nombreux qu'on ne le croit, et souvent assez faciles à mettre en oeuvre. Du plus simple au plus compliqué, du moins cher au plus coûteux, l'utilisateur n'a que l'embarras du choix. Parfois même, un simple tableur peut se révéler un outil suffisant pour de petites applications.

Le problème est justement de bien choisir son logiciel, non seulement pour aujourd'hui, mais aussi pour les besoins futurs que l'on n'imagine pas encore. Ce chapitre devrait vous aider à y voir plus clair.

[6] - Attention, il ne s'agit pas de l'ancien « gestionnaire de fichiers » de la version 3 de Windows, mais bien de logiciels permettant de gérer des fiches, c'est à dire de pouvoir créer, remplir, lire, etc. des fichiers de données.

1.2.1 - Les bases de données

A quoi sert un **gestionnaire de fichiers** et sa version plus élaborée, le **système de gestion de bases de données** (SGBD) ? De manière générale, à automatiser le traitement de collections d'informations de même nature. Très concrètement, à remplacer avantageusement les centaines de fiches cartonnées, les dizaines de dossiers suspendus et les épais répertoires qui servent à retrouver l'adresse d'un client, à classer des dossiers professionnels, ou encore à conserver un semblant d'ordre dans les centaines de disques de cette magnifique collection dont vous êtes, à juste titre (du moins vous le croyez), si fier.

En quoi l'ordinateur est-il mieux adapté à cette tâche que les systèmes de classement traditionnels ? Après tout, il va falloir se plonger dans l'étude d'un logiciel, consacrer du temps à la mise en place de son application, puis procéder à la saisie des informations... Mais le jeu en vaut largement la chandelle. Des produits haut de gamme, comme **Paradox** de Borland (pour PC et Windows); **FoxPro** de Microsoft (pour PC) ou **4°Dimension** d'ACI (pour Macintosh et Windows), des produits moins coûteux comme **Microsoft Access** (pour Windows), **Filemaker Pro** de Claris (pour Macintosh ou Windows) ou encore le module Données de **Microsoft Works,** comme celui de **Claris Works** (pour Macintosh ou Windows) vous permettront de trier très rapidement toutes vos fiches selon l'ordre de votre choix, avant de les imprimer.

Si l'on prend l'exemple d'une collection de disques, vous sortirez par exemple des listes de disques triées par artiste, par titre, par année de sortie, avec éventuellement le style de musique comme second critère de tri. Dans le cas d'un suivi de clientèle, vous pourrez facilement sélectionner d'un seul coup tous les clients dont le solde est débiteur, et imprimer à partir de cette sélection des étiquettes ou des enveloppes, voire des lettres de relance personnalisées. Quant au gain de place, il est considérable, puisqu'une simple disquette peut contenir l'équivalent de plusieurs étagères pleines de dossiers.

Tout d'abord, plus encore qu'avec toute autre catégorie de logiciels, il est impératif avec un gestionnaire de fichiers de définir précisément ses besoins. La moindre erreur, en particulier lors de la définition de la structure des données, pourra se traduire par d'importantes pertes de temps, surtout si on ne la découvre que lorsque toutes les données ont été saisies et que l'exploitation réelle a commencé. **Mettre la réalité en fiches n'est pas toujours aussi simple qu'on pourrait le penser**, il faut des années d'expérience pour être capable d'analyser correctement du premier coup un problème informatique complexe.

1.2.2 - Qu'est-ce que le modèle relationnel ?

Les logiciels dBase, Paradox, Quatrième Dimension, DataEase et autres FoxPro sont ce que l'on appelle couramment des **SGBDR** (ou systèmes de gestion de bases de données relationnelles). Seulement, que recouvre réellement le terme « **relationnel** » et s'applique-t-il réellement à tous les produits cités plus haut ?

1.2.2.1 - Origine et principe

C'est un dénommé **Codd**, travaillant chez IBM, qui a défini vers 1972 **le modèle relationnel de gestion des données**. A cette époque, les fichiers étaient gérés selon une approche plus complexe dite « hiérarchique ». Son principal défaut était son manque de flexibilité. Une fois que la structure d'un fichier avait été définie, il se révélait extrêmement difficile de la changer. Le modèle relationnel est par nature beaucoup plus souple, même si son concepteur a tenté d'en définir les tenants et les aboutissants avec douze règles devenues célèbres. En 1979, **Oracle** publiera

le premier SGBD vraiment relationnel sur mini-ordinateur. Il sera suivi par IBM (avec DB2) et par Ingres, un peu plus tard.

Malheureusement, dans le monde de la micro, **le terme « relationnel » va revêtir une signification beaucoup plus libérale, parfois d'ordre purement publicitaire**. Ainsi, le logiciel **dBase II**, qui apparaît vers 1982, est présenté au public comme un **SGBD.R**. Dans les faits, il ne respecte aucune des douze règles de Codd ! Ashton-Tate n'a utilisé cet adjectif que dans le but de mettre en avant une capacité de son logiciel : la possibilité (alors limitée) d'établir une relation (ou **jointure**) entre deux fichiers. D'autres programmes apparus à cette époque (K-Man, Open Access ou Omnis) se targuent également du qualificatif « relationnel ».

Malgré cet abus de langage, le pas en avant est important. Le fait de relier plusieurs fichiers (au sens de la jointure) évite de répéter plusieurs fois les mêmes données. Si une société vend un produit X, elle gère habituellement un fichier dans lequel se trouvent sa référence, son nom, son prix et d'autres informations. Avec une gestion mono-fichier il se pose en général le problème suivant : lorsque nous créons une facture, il faut introduire manuellement les renseignements relatifs au client, puis à chaque produit. Avec un système relationnel, même si ce « relationnel » se limite à de simples jointures, il suffit d'introduire la référence : les informations « nom du client » et « adresse » seront prises dans le fichier des clients, tandis que les informations « nom du produit » et « prix » seront récupérées du fichier des produits. Toutes ses informations seront incluses automatiquement dans la facture. Il suffira à l'utilisateur de taper le code du client, puis le code des produits (ainsi, évidemment, que la quantité achetée). La création de la facture (en fait une nouvelle table dans la base de données) sera ainsi facilitée..

C'est à partir de cette possibilité de créer des liens entre fichiers, de cette capacité de réaliser des jointures entre tables, que certains programmes s'attribuent le qualificatif de relationnel. Certains le sont vraiment, d'autres non. Mais le terme est à la mode, et c'est un excellent argument publicitaire...

1.2.2.2 - Les relations

En principe, trois types de jointure existent : un-à-un (1-à-1), un-à-plusieurs (1-à-N) et plusieurs-à-plusieurs (N-à-N ou M-à-N).

La **relation 1-à-1** est relativement peu utilisée, sauf dans des cas particulier. Elle permet de relier ensemble deux aspects d'une même entité, comme par exemple des données accessibles à tous, et des données confidentielles, protégées par un mot de passe.

La **relation 1-à-N** permet de lier N enregistrements d'un fichier à un enregistrement d'une table (fichier) maître. Une base listant des entreprises constitue un exemple typique. L'entête des états obtenus indique le nom et l'adresse des sociétés, et provient généralement d'un premier fichier (le fichier dit « maître » ci-dessus). Les lignes de détail indiquent les noms des principaux dirigeants et proviennent d'un autre fichier. Paradox et R:Base autorisent la définition d'une telle relation par le biais de masques de saisie. On peut arriver au même résultat sous MS.Access à l'aide d'une **requête**. Sous dBase IV, il est nécessaire de faire appel à la programmation.

La **relation M-à-N** autorise le lien de M enregistrements d'une table à N autres d'une seconde table. Elle permet de savoir que M personnes du nom de Dupont habitent à Lille, mais également qu'il y a des Dupont dans N autres villes. Une telle relation peut être gérée de façon dynamique (actualisée en permanence), en particulier sous les produits de la famille dBase, en passant par la programmation. Sous Paradox, il est possible d'obtenir une telle information par l'intermédiaire du module d'interrogation QBE, sans programmation. Notez tout de suite que ce type de relation va poser de gros problèmes lorsqu'on passera de l'étude théorique à la phase pra-

tique, de la conception à la réalisation. Chaque fois que possible, on tâchera de la remplacer par deux (ou plus) relations de type 1-à-N.

Dans les faits, d'autres opérations que la jointure devraient être réalisables sur les tables d'une base de données réellement relationnelle : restriction, projection, union, différence, intersection et division. Sans entrer dans le détail, nous retiendrons qu'elles permettent de générer des bases d'informations complexes à partir de plusieurs fichiers et qu'elles ont pour point de départ la théorie des ensembles. Plusieurs de ces opérations sont notamment accessibles sous Paradox par le biais du QBE, par des requêtes sous MS.Access, ou encore par de la programmation dans d'autres logiciels.

1.2.2.2 - Tables et Bases

La seconde notion à mettre en avant dans un environnement relationnel est celle de **base** et de **tables**. Par exemple, Paradox ou dBase gèrent des fichiers indépendants (des tables) qu'ils permettent de relier par jointure, formant ainsi une base de données gérable par un logiciel (un SGBDR). C'est également le cas de MS.Access, mais les tables sont alors enregistrées toutes ensemble, dans le même fichier. Elles ne sont donc pas disponibles directement, sauf par certains logiciels spécialisés.

D'autres systèmes, tels que R:Base ou SQL Server, se rapprochent davantage du modèle relationnel défini par Codd en ce sens que l'utilisateur manipule une véritable base d'informations au sein de laquelle il définit des fichiers, des « tables », indissociables de la dite base. L'avantage de cette seconde approche est évidemment qu'elle autorise une plus grande rigueur. Dans le cas de MS.Access, les bases de données se présentent sous l'aspect d'un fichier unique, incluant non seulement les tables, mais aussi d'autres éléments tels que les requêtes, les formulaires, les états et les modules de programmation. Toutefois, malgré cet amalgame, il existe une réelle indépendance des données par rapport aux moyens d'interrogation.

Les termes de fichier et de table sont à manier avec précaution. Le premier concerne davantage le mode d'enregistrement, tandis que le second correspond plutôt à l'utilisation qui en sera faite. Dans les logiciels « anciens », la confusion est possible : chaque table est enregistrée sous forme d'un fichiers, ou même de plusieurs, si l'on tient compte des index qui accompagnent la table. Au contraire, il arrive souvent que des logiciels plus récents enregistrent globalement la base de données, y compris les tables, en un seul fichier.

1.2.2.4 - Pensez relationnel

Lorsqu'une entreprise décide d'acquérir un SGBD.R, il faut naturellement qu'elle se pose la question de son utilisation. Dès lors que ce logiciel est supposé gérer les données vitales de l'entreprise, il se doit de répondre à certaines des règles énoncées par Codd afin de garantir la sécurité d'accès aux données et leur cohérence. **Voici quels sont les points essentiels :**

L'intégrité référentielle

Assurer l'intégrité référentielle consiste à mettre en oeuvre des moyens destinés à préserver la cohérence d'une base d'informations lors de ses mises à jour. Le cas le plus souvent évoqué est le suivant. Sur un SGBD micro, comme dBase, il est tout à fait possible de détruire un client qui a encore des factures impayées. Avec un moteur relationnel tel que SQL Server ou Oracle, une telle opération est **interdite** quel que soit le logiciel client qui interroge la base. Idéalement, il devrait être possible de définir des règles de validation des données directement dans la structure du fichier. SuperBase et R:Base possèdent une telle fonctionnalité,

absente de dBase ou de FoxPro pour lesquels il faut indiquer de telles règles dans chacun des masques de saisie, ou bien dans les programmes.

La gestion de l'intégrité référentielle par MS.Access est à considérer comme un cas particulier. Il est en effet possible de l'appliquer ou non lors de l'édition des relations. Si cette méthode est pratique lorsqu'il s'agit de récupérer des données en provenance d'autres logiciels, elle risque de poser des problèmes si l'on est pas extrêmement rigoureux lors du développement de nouvelles applications.

La notion de transaction

Le modèle relationnel implique la mise à jour de plusieurs données simultanément. L'exemple le plus classique est le suivant : une écriture comptable au débit doit s'accompagner automatiquement de la mise à jour d'un compte de crédit. Que se passe-t-il si une panne de courant survient avant que la seconde écriture ait pu être mise à jour ? Nous aboutissons à une base incohérente. Pour résoudre ce problème, les SGBD avancés offrent des possibilités de « rollback » (retour en arrière) ou de « commit » (validation de la transaction).

Une opération comptable telle que celle citée plus haut, avec des enregistrements multiples, est appelée une **transaction**. Elle est suivie pas à pas dans un journal. Lorsque les deux écritures ont été passées, le système reçoit un ordre de **commit** qui valide l'ensemble de la transaction. Si une panne s'est produite avant la fin des opérations, lors du redémarrage de la machine, le SGBD.R constate qu'il y a eu une ou plusieurs écritures non suivies d'un commit et déclenche un **rollback** : la base est restaurée en l'**état supposé cohérent** qu'elle avait avant le début de la transaction. Ces fonctions sont gérées par tous les SGBD.R serveurs (Oracle, SQL Server) mais aussi par certains produits « micro » tels que dBase IV, à condition de faire de la programmation.

La sécurité

Pour des raisons évidentes de sécurité et de confidentialité, l'accès aux données en réseau doit pouvoir être **limité** de diverses façons, et Codd énonce le principe des « vues ». Sur une table synthétisant les informations de plusieurs fichiers, il doit être possible de créer des **vues** ne faisant apparaître que les données désirées. L'exemple classique sera celui de la fiche d'un employé dans laquelle le salaire ou l'adresse resteront invisibles pour certaines catégories d'utilisateurs.

Certains SGBD autorisent une définition de mot de passe allant jusqu'au champ de chaque table. Dans le cas contraire, la séparation de la table en deux parties, reliées par une relation de type **1-à-1** est une solution acceptable. Une des deux tables est alors mise en « libre-accès », tandis que l'autre est protégée par un mot de passe.

1.2.3 - Les 12 règles de Codd

Vers 1970, R.F. Codd, alors chercheur chez IBM, a défini les règles sur lesquelles devrait être construit un modèle relationnel assurant l'intégrité des données. Les « douze règles de Codd » n'ont rien perdu de leur valeur aujourd'hui. En voici l'inventaire...

1. Représentation des informations

Les informations sont représentées au niveau **logique**, et non physique, ce qui signifie que l'on ne se préoccupe pas de l'implémentation réelle des données.

Elles sont décrites par des valeurs contenues dans des tables, organisées en lignes et colonnes. *(donc comme dans un tableur...)*

2. Accès aux données

Une donnée est accessible logiquement, c'est à dire sans connaissance de son implantation physique, grâce à la combinaison du nom de la table, de la clé primaire (ou d'un numéro d'article) et du nom de la colonne (*du nom du champ*).

3. Gestion des valeurs absentes

Cette règle précise que la valeur NULL (celle qui correspond à une absence d'information) est interprétable, et de valeur différente d'une chaîne de caractères vide (ou composée de « blancs ») ou encore d'une valeur numérique égale à zéro.

4. Dictionnaire de données

La description de la base de données est représentée par des informations accessibles comme s'il s'agissait de données ordinaires. Le langage relationnel permet donc en principe de manipuler indifféremment des données du système d'information ou des données décrivant la base elle-même. La description de la base est donc stockée dans des tables faisant partie de ce que l'on nomme le dictionnaire de données.

5. Le langage

Le SGBD doit inclure au moins un langage comportant l'ensemble des fonctionnalités suivantes : définition des données, définition des vues, manipulation des données, contraintes d'intégrité, autorisations, gestion des transactions.

6. La mise à jour à travers une vue

Une vue est un mode de représentation logique de la base de données. Si une vue peut être mise à jour, elle peut aussi l'être par le système.

7. La mise à jour des tables

Le langage relationnel doit disposer d'ordres de haut niveau s'appliquant non seulement à la lecture des données, mais aussi à la création, la mise à jour ou la suppression d'informations.

8. L'indépendance physique

Les programmes d'applications et les transactions interactives sont indépendants de la représentation physique des données et des méthodes d'accès sous-jacentes. Cela garantit la souplesse d'évolution du système d'information et exige que le SGBD dissocie efficacement la représentation logique d'une part et les aspects d'organisation physique d'autre part.

9. L'indépendance logique

Cette règle stipule que les programmes ne sont pas remis en cause lorsque des modifications (sans perte d'informations structurelles) sont opérées sur les relations de la base. Citons par exemple l'éclatement d'une table en deux, ou, à l'opposé, la fusion de deux tables.

10. L'indépendance vis-à-vis des contraintes d'intégrité

Les contraintes d'intégrité susceptibles d'évoluer dans le temps doivent pouvoir être formulées en dehors de tout programme applicatif et être référencées dans le dictionnaire des données. Une application informatique constitue une réponse à un problème qui se pose à un moment précis. Si l'entreprise et son environnement évoluent, le système d'information doit par conséquent pouvoir s'adapter à cette évolution, sans que soit remis en cause l'acquis applicatif.

11. L'indépendance vis-à-vis de la répartition des données

L'environnement applicatif n'est pas affecté par la répartition des données sur des sites et des supports physiques distincts, ou toute modification de cette éventuelle répartition.

12. La non-subversion

Cette dernière règle stipule que si le système dispose d'un langage de bas niveau, ce langage ne peut pas contourner ou remettre en cause les contraintes de sécurité et les règles d'intégrité énoncées au plus haut niveau.

Chapitre 3 : Conception d'une base

Une base de données bien conçue doit permettre d'accéder rapidement à l'information. Lorsque la base de données est structurée correctement, les temps de réponses sont plus courts. L'élaboration d'une base de données est un travail d'analyse méthodique qui prend du temps mais qui, à long terme, en économise beaucoup plus.

Pendant cette phase, il faut établir le schéma de la structure de la base de données. Une base bien conçue évite des corrections ultérieures et optimise la productivité de ses utilisateurs. De plus, cela retarde les inévitables problèmes de maintenance, et assure à la base une longévité accrue.

1.3.1 - Intérêt d'une bonne étude...

Pour bien comprendre le processus de création d'une base de données, **il faut différencier la manière dont on utilise les données de la manière dont un système de gestion de base de données relationnelle les stocke**. Vous utilisez les données en tant qu'informations présentées intelligiblement. Vous utilisez souvent une donnée, un élément d'information, dans plusieurs contexte, et dans chaque contexte, vous pouvez combiner cet élément à un ensemble différent de données. Ceci n'est pas le cas des systèmes de gestion de base de données relationnelle. Pour présenter efficacement l'information, un SGBD.R doit d'abord stocker les données par sujet, dans des tables contenant les données qui se rapportent à chacun de ces sujets. Ensuite, pour associer correctement les données, le logiciel doit connaître la(les) relation(s) qui existe(nt) entre les entités décrites dans les tables.

Créer une base de données consiste donc à établir la liste des données que devra contenir la base, puis à les regrouper par thème. Dans un second temps, vous déterminez, et notifiez au logiciel, les liens entre les entités, afin qu'il puisse les relier en temps voulu. Pour un particulier, cherchant à satisfaire l'un ou l'autre besoin personnel, on peut considérer que les phases de la création d'une base de données sont les suivantes :

Première phase :	Définition de l'objectif.
Seconde phase :	Définition des tables.
Troisième phase :	Définition des champs.
Quatrième phase :	Définition des relations.
Cinquième phase :	Mise au point de la structure.

Le développement d'une application professionnelle suit aussi, grosso modo, le même schéma de développement. Seulement, l'envergure différente du problème et la nécessité d'y associer d'autres personnes, ou d'autres services, conduisent à utiliser des méthodes plus structurées, comme la méthode Merise. **Dans tout ce qui suit, nous resterons à un niveau « personnel », pour ne pas dire « artisanal », dont la démarche est essentiellement de type « essai - erreur - correction »...**

1.3.2 - Création d'une base de données

1.3.2.1 - Etape 1 : Définition de l'objectif

C'est au cours de cette phase que vous déterminez l'information que contiendra la base de données.

La création d'une base de données commence par la définition de l'objectif de la base et des traitements qu'elle subira. C'est au cours de cette phase que vous précisez les tâches que le système d'information devra assurer. C'est à partir de cela que, **dans les étapes suivantes**, vous déterminerez d'abord les sujets pour lesquels vous voulez stocker des données (les tables) puis le détail des données elles-mêmes (les champs).

Vous devez connaître ce qui existe, et ce que l'on veut demander à la future base. Pour cela, interrogez-vous. Posez d'abord toutes les questions auxquelles devra répondre la base de données. Esquissez ensuite les états qui devront être générés. En même temps, dessinez les formulaires et les rapports que la base devra reproduire.

En fait, il s'agit surtout dans cette première phase de rassembler de la « documentation », à la fois sur les méthodes utilisées actuellement, et celles que l'on désire met-

tre en oeuvre. Vous aurez besoin de toutes ces informations tout au long de la création de la base de données. Une fois que vous avez rassemblé toutes ces informations, vous êtes prêt à passer à l'étape suivante.

1.3.2.2 - Etape 2 : Définition des tables

Une fois que l'objectif a été établi, vous déterminez les principaux sujets de la base. Chaque sujet correspondra à une table dans la base de données.

La définition des tables est la phase la plus délicate du processus de création d'une base de données. En effet, les résultats que vous attendez de la base de données, les états que vous voulez imprimer, les formulaires que vous voulez utiliser, les recherches que vous voulez effectuer ne fournissent pas nécessairement d'indices sur la structure même des tables à construire. Ils vous indiquent ce que vous voulez savoir, mais pas la manière d'organiser les données dans des tables.

Analyser l'information que vous voulez obtenir de la base de données et éclatez-la en thèmes ou sujets principaux. Chacun de ces sujets pourra donner naissance à une table. Une bonne technique consiste à examiner chaque information et à déterminer à quoi elle se rapporte. A chaque « sujet », à chaque « **objet** », ainsi désigné dans le domaine de l'étude, devra correspondre au moins une table, parfois plusieurs si l'objet est complexe. Ainsi, pour une agence immobilière, l'objet « *immeuble* » est en fait une collection d'objets « *appartement* ». A vous de déterminer quelles tables vous allez créer.

1.3.2.3 - Etape 3 : Définition des champs

Vous déterminez les données à placer dans chaque table. chaque catégorie de donnée dans une table s'appelle un champ, et correspond à une colonne de la table, présentée sous forme d'un tableau.

Au cours de cette phase, vous déterminez précisément l'information dont vous avez besoin à l'intérieur de chaque table. On peut dire que les champs sont les caractéristiques de la table. Chaque enregistrement (ou ligne) de la table contient obligatoirement le même ensemble de champs (ou colonnes), même s'ils sont vides.

En toute rigueur, un champ donné ne peut figurer que dans une seule table, afin d'éviter la répétition des informations, et faciliter les mises à jour. Si, au cours de la définition des champs, on est amené à se poser des problèmes de « doublons », cela peut provenir d'une mauvaise définition des tables. Cette phase peut ainsi vous conduire à les remettre en cause ! **En cas de besoin, n'hésitez pas à revenir en arrière. Il n'est pas encore trop tard.**

La puissance d'un SGBD.R réside dans sa capacité à rechercher et à relier les données de plusieurs tables. Pour réaliser ces tâches avec une efficacité maximale, chaque table doit inclure un (ou plusieurs) champ(s) qui identifie(nt) chaque ligne (ou enregistrement) de la table. Il doit s'agir d'un **identifiant unique**. En terminologie des bases de données, cette données s'appelle une **clé primaire**, ou une clé unique. D'un point de vue mathématique il y a bijection entre l'ensemble des valeurs de l'attribut et l'ensemble des enregistrements de la relation. La clé primaire peut être composée, c'est à dire qu'elle peut être formée de plusieurs attributs (de plusieurs champs). Elle doit obéir aux règles suivantes :

---> **elle est définie d'une manière unique**,
---> **elle n'est jamais mise à jour, ni modifiée**.

Quelques conseils...

--> *Chaque champ d'une table doit se rapporter directement au sujet de celle-ci*

Un champ qui décrit un autre sujet relève d'une autre table. Pour l'instant, il importe que chaque champ d'une table se rapporte exclusivement à celle-ci. Plus tard, lorsque vous définirez les liens entre les tables, vous verrez comment relier des champs provenant de plusieurs tables.

--> *Donnez des noms significatifs*

Il est conseillé de donner aux champs des noms descriptifs qui faciliteront leur identification lors de l'affichage ou de la modification des enregistrements. Par exemple, Nom, Prénom, Téléphone,... La plupart des logiciels actuels admettent des noms suffisamment longs pour que cela ne pose pas de problème.

--> *Prévoyez dès maintenant la nature des champs*

Au moment où vous passerez sur l'ordinateur, il faudra lui indiquer s'il s'agit d'un champ de type alphabétique, numérique, date, logique, graphique, ou autre. Selon la nature de chaque champ, les opérations susceptibles d'être faites ne sont pas identiques.

--> *Ne placez pas dans les champs des données calculées*

D'une manière générale, il vaut mieux demander au logiciel d'effectuer des calculs (moyennes, comptages, cumuls...) plutôt que de placer le résultat d'un calcul dans une table, à moins que les données à traiter ne soient justement ces valeurs.

--> *N'oubliez rien*

Certaines informations importantes peuvent vous échapper. Consultez à nouveau les documents que vous avez rassemblés au cours de la première phase. Assurez-vous que toutes les informations dont vous avez besoin se trouvent bien dans vos tables, ou peuvent en dériver. Pensez aux interrogations futures, et vérifiez à plusieurs reprises toutes les étapes antérieures...

1.3.2.4 - Étape 4 : Définition des Relations

Vous déterminez les liens qui peuvent exister entre les données d'une table et celles des autres tables. Au besoin, vous ajoutez des champs aux tables, ou alors, vous créez de nouvelles tables, pour clarifier les liens.

Une fois que vous avez réparti les informations dans les différentes tables, vous devez indiquer comment relier et présenter ces informations de manière cohérente. N'oubliez pas que le terme « relationnel » signifie que vous pouvez créer des liens entre les tables de la base de données, afin de ne pas devoir enregistrer plusieurs fois les mêmes données. Il serait dommage de ne pas chercher à utiliser au mieux cette possibilité. Il faut donc dès à présent prévoir ces liens, ces relations.

Pour établir une relation entre deux tables (une table A et une table B) vous devez ajouter la clé d'une table à l'autre, de façon à ce qu'elle apparaisse dans les deux tables. Mais comment savoir quel clé primaire, champ ou groupe de champ utiliser ? Pour définir correctement la relation, vous devez tout d'abord déterminer son type. En pratique, on peut en rencontrer trois :

---> - relation un-à-un **1-à-1.**
---> - relation un-à-plusieurs **1-à-N**,
---> - relation plusieurs-à-plusieurs **N-à-N**,

- relation un-à-un,

Dans une relation de type **un-à-un**, un enregistrement de la table A ne peut être relié qu'à un seul enregistrement de la table B, et inversement. Ce type de relation est rare et sa présence peut entraîner la révision de la structure de la base. En théorie, elle ne devrait pas exister, mais c'est parfois bien utile... Ces relations sont rares, parce que dans la majorité des cas, il est possible de combiner les données des deux tables en une seule. Par contre, elle peut servir à diviser une information en une table à accès libre, et une autre protégée par un mot de passe. En séparant les données en deux tables, on donne ainsi un libre accès aux données générales, tout en limitant l'accès aux données personnelles.

- relation un-à-plusieurs,

Dans une base de données relationnelle, la relation **un-à-plusieurs** est la plus courante. Avec cette relation un enregistrement de la table A peut être lié à plusieurs enregistrements de la table B, mais un enregistrement de la table B ne peut être relié qu'à un seul enregistrement de la table A. En pratique, pour établir la relation, on ajoute le ou les champs clés de la première table (coté « un » de la relation) dans la seconde table (coté « plusieurs »).

- relation plusieurs-à-plusieurs,

Dans une relation de type **plusieurs-à-plusieurs**, un enregistrement de la table A peut être associé à plusieurs enregistrement de la table B, et inversement. Pour détecter les relations plusieurs-à-plusieurs, il est important d'examiner la relation dans les deux sens. Ce type de relation peut entraîner la révision de la structure de la base de données. De plus, elles sont toujours difficiles à traiter directement, et il est préférable de les éviter. Dans certains cas, une solution consiste à créer une table supplémentaire pour éclater la relation de type plusieurs-à-plusieurs en deux relations de type un-à-plusieurs. En pratique, pour cela, on place la clé primaire de chacune des deux premières tables dans la nouvelle table.

1.3.2.5 - Etape 5 : Mise au point

Vous analysez la structure et éliminez les erreurs. Pour ce faire, vous créez les tables, saisissez quelques enregistrements, et vérifiez si vous obtenez les résultats escomptés. Ensuite, si c'est nécessaire, vous modifiez la structure en conséquence avant de recommencer les tests....

Une fois que vous avez défini les tables, les champs et les relations, vous devez tester l'ensemble pour détecter les éventuelles (et inévitables) imperfections de votre prototype de base de données.

Créez les tables, spécifiez les relations entre elles et saisissez quelques enregistrements dans chaque table. lancer des interrogations et vérifiez les résultats escomptés. Créer des ébauches de vos formulaires et états pour voir s'ils présentent les données attendues. Recherchez les doublons superflus et éliminez-les. Lors de vos tests, vous découvrirez certainement des aspects à modifier ou à améliorer. La liste ci-dessous énumère quelques points de contrôle :

--> Avez-vous omis des champs ? Toutes les informations sont-elles incluses ? Si ce n'est pas le cas, pouvez-vous les ajouter dans la structure actuelle ou devez-vous créer de nouvelles tables ?

--> Avez-vous défini une clé primaire correcte pour chaque table ? Si vous utilisez une clé pour rechercher certains enregistrements, est-elle facile à mémoriser ? Vérifiiez que toutes les valeurs qui devront entrer dans les champs de type « clé primaire » sont uniques. Si non, votre clé est mal choisie ! Il faut retourner à la troisième étape.

--> Saisissez-vous plusieurs fois la même donnée dans une table ? Vous pouvez probablement répartir ces données dans deux tables, avec une relation 1-à-N.

--> Certaines tables comportent-elles de nombreux champs, peu d'enregistrements et plusieurs champs vides par enregistrements ? La structure de ces tables gagnerait à être révisée.

1.3.2.5 - Première utilisation...

Cette fois, tout est prêt : les structures de fichiers, les masques avec leurs boutons, leurs boîtes à cocher et les boîtes de dialogue associées, les états imprimés. Ne vous précipitez pas pour autant sur les centaines de fiches en carton qui n'attendent que ça, à côté de votre ordinateur. Choisissez-en plutôt une vingtaine, qui contiennent tous les types de données que vous utilisez, et saisissez-les. C'est un moment émouvant, celui où l'on ressent pour la première fois le « look and feel » de son application, où l'on s'extasie devant les fonctions qui marchent... et où l'on peste contre celles qui « boguent ». C'est aussi le moment où l'on découvre tout ce que l'on a oublié. Mais, à ce stade, il est encore possible de modifier une structure de fichier, de retoucher un masque, de corriger l'action d'un bouton. Dans le pire des cas, celui où les erreurs empêchent de continuer avec les données déjà entrées, on n'aura que quelques fiches à ressaisir.

Cette mise au point doit être menée jusqu'au bout, c'est-à-dire jusqu'à ce que chaque masque ait été utilisé, chaque état imprimé, toutes les questions posées, sans qu'aucun des cas de figure possibles n'ait été oublié. Ensuite seulement, la saisie complète et l'exploitation réelle pourront commencer.

Et rassurez-vous : nous ne sommes plus à l'ère de l'informatique en temps différé et des gestionnaires de fichiers complètement rigides. Les produits actuels sont suffisamment souples pour que vous puissiez a posteriori ajouter des masques ou des états imprimés, modifier ceux déjà existants, ajouter, voire supprimer des champs à un fichier. Cela dit, moins on y recourt, mieux l'application fonctionnera. Et une modification peut entraîner des ajustements particulièrement fastidieux, voire dévoiler des problèmes passés jusque là inaperçus. **Surtout, n'oubliez pas, avant d'effectuer une opération délicate, de sauvegarder toutes les données ! Et cela, plutôt deux fois qu'une ! On n'est jamais trop prudent.**

1.3.3 - Quelques règles simples

Voici en guise de conclusion le rappel de quelques règles simples, nouvelles ou déjà vues dans ce qui précède, mais énoncées ici d'une manière plus condensée. Elles pourront vous être utiles pour créer une base de données. Attention, ces règles ne relèvent pas d'une méthode « professionnelle », mais bien d'un travail « artisanal », d'un travail d'amateur. Les professionnels utilisent des méthodes beaucoup plus structurées, et beaucoup plus efficaces.

Phase « Conception »

- Recenser tous les « objets » (les tables) qui vont composer la base,

- Recenser toutes les rubriques (les champs) qui décrivent chaque objet,

- Eviter les redondances d'information : une rubrique ne doit pas se répéter dans une même table, ni dans plusieurs, sauf si l'établissement de liens (de relations) l'exige.

- Si une même rubrique doit néanmoins figurer dans plusieurs tables, elle doit absolument avoir le même type et la même longueur partout, ainsi que, de préférence, le même nom.

- Etablir un schéma aussi complet que possible de la base de données, avec toutes ses tables et les relations qui les lient.

Phare « Réalisation »

- Vérifier que le logiciel prévu peut supporter les contraintes imposées (nombre de tables, de champs, types des champs, etc.).

- Simuler l'utilisation de la base à l'aide d'un jeu d'essai. Sans oublier les cas particuliers qui pourraient se poser lors de l'utilisation réelle.

1.3.4 - Exemple

A titre d'exemple, nous allons dire très rapidement quelques mots sur une petite base de données, très simple. En fait, il s'agit d'étudier les jurons du Capitaine Haddock à travers les Aventures de Tintin, des ouvrages bien connus...

1.3.4.1 - Définition de l'objectif

A priori, l'objectif est simple, et très limité : nous voulons pouvoir établir la liste des jurons utilisés par le Capitaine Haddock, et en même temps savoir dans quelles circonstances ils ont été proférés. Pour ceci, il nous suffira, dans notre étude, de savoir dans quel ouvrage, et à quelle page ils se trouvent.

1.3.4.2 - Définition des tables

Ici aussi, peu de problème. Il est assez évident que le domaine de notre étude ne comporte que deux « sujets », que deux « objets », selon le vocabulaire que nous utiliserons : d'un coté, il y a les **jurons**, de l'autre, il y a les **livres**...

1.3.4.3 - Définition des champs

En ce qui concerne les jurons proprement dit, à nouveau peu de problème. Il suffit d'un seul champ (contenant le juron lui-même) pour décrire l'objet « juron ». A la rigueur, on pourrait y ajouter un second champ, donnant la signification exacte du terme utilisé. Les affaires se compliquent un peu avec l'objet « **livre** ». A première vue, il suffirait de définir chacune des « **Aventures** » par son titre.

Seulement, là où l'affaire se complique, c'est que certaines ont été remaniées à plusieurs reprises (cf. « *L'île noire* » ou « *Tintin au pays de l'or noir* »), et parfois même, le vocabulaire a été changé (comme pour « *Coke en stock* », en ce qui concerne le Capitaine Haddock) à la demande de l'une ou l'autre de ces commissions qui se défendent de faire de la censure mais prétendent veiller sur la jeunesse française[7]... Dans le cas de « *Coke en stock* », il existe en fait trois versions : celle qui a paru dans le journal Tintin (en Belgique), et deux versions en album, la première (éditée en Belgique), et les réédition, modifiées à nouveau...

Le titre seul est donc insuffisant pour définir l'objet « **livre** ». Il faut préciser également, par exemple, l'éditeur et l'année d'édition. **Remarquer au passage que pour définir les**

[7] - C'est aussi l'une de ces commissions qui a obligé Lucky Luke à mâcher un brin d'herbe, à la place de sa cigarette, au titre de la lutte contre le tabagisme...

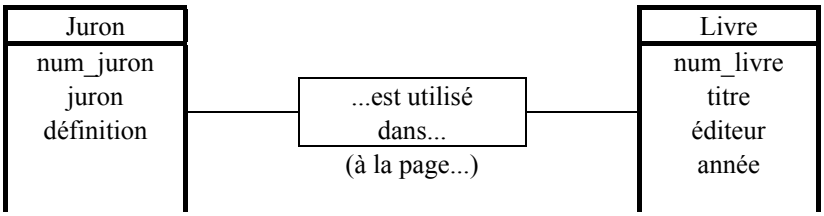
champs, il est nécessaire d'avoir une certaine connaissance du thème de la base de données, d'où l'obligation de dialoguer avec les utilisateurs, ou avec « ceux qui savent »...

1.3.4.4 - Définition des relations

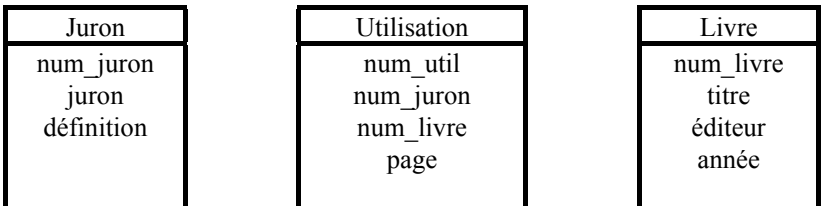
Ici, il nous sera plus facile de raisonner sur un petit dessin. Aussi, nous allons d'abord tenter de représenter les deux tables par les listes des champs qu'elles contiennent. Le premier est un peu spécial. En effet, nous savons déjà que dans chacune des tables doit figurer une **clé primaire**. Ce rôle sera joué par un champ nommé « numéro ». Dans une première approche, nous serons donc en présence des tables **Juron** et **Livre** avec les champs suivants :

Ce premier schéma fait apparaître un risque de confusion en ce qui concerne le vocabulaire. Le terme de « numéro », utilisé deux fois, n'est pas assez explicite pour l'utilisateur humain. Cela va nous conduire à changer les noms des champs « Numéro ».

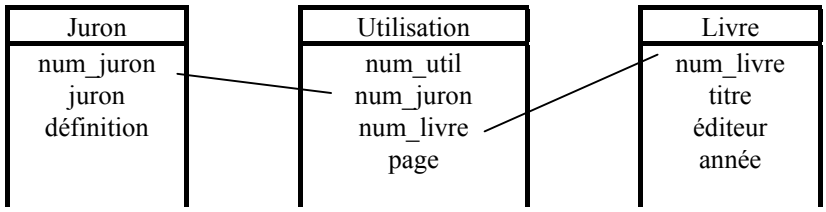
Pris dans sa globalité, le lien entre ces deux tables est apparemment du type N-à-N, ou plusieurs à plusieurs. En effet, rien n'empêche le capitaine de proférer plusieurs jurons dans le même album, par plus que d'utiliser le même juron dans plusieurs albums... Il faut donc réfléchir davantage. Sous sa forme la plus simple, nous pouvons dire qu'un juron est utilisé dans un livre...



En fait, chacune des « utilisations » d'un juron est unique, en ce sens que les circonstances où il est proféré ne se renouvelleront pas. Il est donc (relativement) facile de dissocier notre relation de type « plusieurs à plusieurs » en deux relations plus simples, de type "un à plusieurs". Cela peut se faire très facilement à l'aide d'une table intermédiaire, comme par exemple dans le schéma suivant :



Sur ce dernier dessin, afin de rendre les relations plus visibles, il est possible de matérialiser les « liens » qui existent entre les tables en joignant par un trait les champ sur lesquels est réalisé ce lien, cette jonction (ici, les champs de même nom).


Juron
num_juron
juron
définition
Utilisation
num_util
num_juron
num_livre
page
Livre
num_livre
titre
éditeur
année

Seconde partie : ...à la pratique

Chapitre 1 : Le logiciel Access
2.1.1 - Le logiciel Microsoft Access
2.1.2 - Un peu de vocabulaire

Chapitre 2 : Exemple d'utilisation
2.2.1 - Comment créer une base (vide)
2.2.2 - Comment créer une table (vide)
2.2.3 - Comment créer un formulaire
2.2.4 - Comment créer une requête
2.2.5 - Comment créer un état
2.2.6 - Maintenance d'une base

Chapitre 3 : D'autres conseils...
2.3.1 - La base de données
2.3.2 - Définir des relations entre les tables
2.3.3 - Qu'est-ce que l'intégrité référentielle

Chapitre 1 : Le logiciel Microsoft Access

Une base de données bien conçue doit permettre d'accéder rapidement à l'information. Lorsque la base de données est structurée correctement, les temps de réponses sont plus courts. L'élaboration d'une base de données est un travail d'analyse méthodique qui prend du temps mais qui, à long terme, en économise beaucoup plus.

Avant d'aborder la phase de réalisation, il faut établir la structure générale de la base de données et de toutes les tables qui la compose. Une base bien conçue évite des corrections ultérieures et optimise la productivité de ses utilisateurs. De plus, cela retarde les inévitables problèmes de maintenance, et assure à la base une longévité accrue.

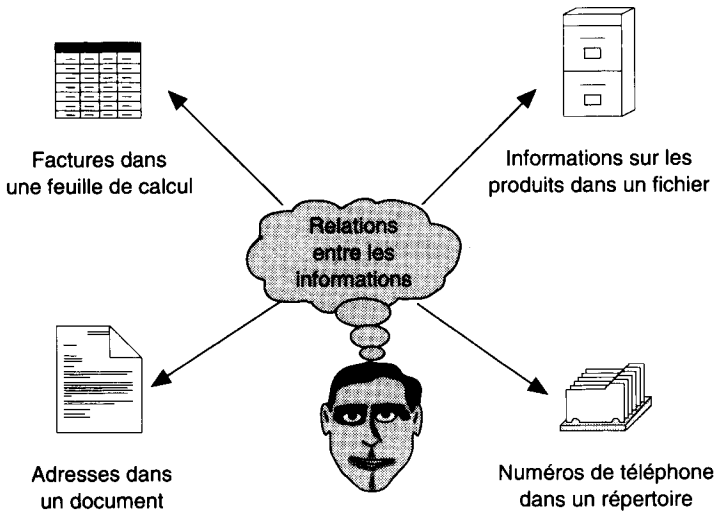
Attention, les dessins d'écran ne sont donnés qu'à titre d'exemple. Il peuvent être différentes de ceux observées, en fonction du logiciel utilisé, et de sa version. Dans les pages qui suivent, il s'agit du logiciel MS.Access (pour Windows 98).

2.1.1 - Le logiciel Microsoft Access...

Les lignes qui suivent sont extraites de la documentation fournie avec l'un des versions (précédentes...) du logiciel Microsoft. Access. Elles sont donc à prendre avec les précautions d'usage : jamais un fournisseur ne vous diras du mal de ses produits...

Une **base de données** consiste en un ensemble de données se rapportant à un sujet ou à un objectif donné. Un classeur regroupant des fiches d'employés, un album de timbres ou encore un carnet de ventes constituent des exemples de bases de données. Un **système de gestion de base de données** (SGBD) est un système qui permet de conserver et de rechercher des informations dans une base de données. Cela équivaut, dans les exemples précédents, au classeur, à l'album ou encore au carnet de ventes. Un SGBD informatisé correspond au programme utilisé pour enregistrer puis extraire vos données de l'ordinateur.

Microsoft Access est un **système de gestion de base de données relationnelle** (SGBDR). Cela signifie qu'avec Microsoft Access, vous pouvez organiser vos données par sujet, afin de pouvoir les gérer et les vérifier facilement. Vous pouvez également conserver les informations sur les relations établies entre ces sujets, afin de pouvoir retrouver aisément des ensembles de données. Pour avoir une meilleure idée de la puissance d'un tel système, supposez une base de données relationnelle non enregistrée sur ordinateur. Vous devez alors garder à l'esprit les relations existant entre les différentes données. Ce peut être le cas, par exemple, d'informations sur des produits et fournisseurs que vous conservez dans différents dossiers. Ainsi, un fichier traitement de texte pourrait contenir la liste des adresses de vos fournisseurs, tandis qu'une feuille de calcul contiendrait la liste des factures et des paiements. Les renseignements sur les produits et leur fournisseur respectif pourraient se trouver dans un fichier rangé dans un bureau et les numéros de téléphone des fournisseurs dans un répertoire téléphonique.



Une fois rassemblés, tous ces documents forment une base de données relationnelle. Chacun a une fonction particulière et, selon vos besoins, vous pouvez regrouper les informations de deux ou plusieurs de ces documents. L'élément maître de ce système demeure votre connaissance sur la façon dont les informations sont connectées entre elles.

Supposons que vous décidiez d'utiliser un ordinateur pour gérer vos données. Avec un système de gestion de base de données non relationnelle, vous pouvez créer dans votre base de données une grande table contenant toutes les informations dont vous avez besoin, à savoir: les numéros de téléphone, les informations produits, les factures, etc. Inévitablement, de nombreuses données sont dupliquées. Par exemple, chaque fois que vous ajoutez un nouveau produit, vous décidez d'inclure le numéro de téléphone du fournisseur pour le retrouver facilement. Si vous

achetez dix produits au même fournisseur, le même numéro de téléphone est enregistré dix fois. Non seulement votre table prend vite des dimensions démesurées, mais cette accumulation de données, ou *doublons,* nécessite un espace disque supplémentaire volumineux. Autre inconvénient de cette méthode : si vous devez mettre à jour des informations, vous devez alors le faire dans dix enregistrements différents dans notre exemple, mais peut-être dans des centaines d'enregistrements différents dans d'autres cas. C'est un processus long et peu fiable dans la mesure où une erreur de saisie est vite arrivée. Finalement, quand vous souhaitez changer la structure de vos données (ajout de numéros de télécopieur, par exemple, ou regroupement de produits par catégorie) la restructuration de la table peut devenir une tâche fastidieuse et causer des erreurs.

A l'inverse, une base de données peut contenir plusieurs petites tables, correspondant en quelque sorte aux différents dossiers dans votre bureau. Vous pouvez concevoir votre base de données de façon à ce qu'elle stocke également les informations concernant les relations établies entre les tables elles-mêmes. En principe, chaque élément d'information n'est stocké qu'une seule fois. Cela économise de l'espace disque, supprime les éventuels doublons et permet une mise à jour de vos informations plus rapide et plus fiable. Il vous est plus facile de restructurer une petite table dans une base de données relationnelle qu'une grande table dans une base de données non relationnelle. Autre avantage : vous pouvez ajouter de nouvelles tables contenant d'autres types d'informations sans devoir modifier l'organisation des tables existantes.

2.1.2 - Un peu de vocabulaire...

Base de données	Une base de données est un ensemble de tables, de requêtes, de formulaires, d'états, de macrocommandes Sous MS.ACCESS, l'ensemble de ces objets est enregistré dans un même fichier d'extension **.MDB**
Etat	Un état sert à imprimer les données dans un format et une mise en page personnalisées, par exemple des étiquettes de mailing, une liste de téléphone ou des rapports de ventes. Dans un état, les données peuvent être triées et regroupées avec des sous-totaux
Feuille de données (table)	Les données d'une table sont affichées sous la forme d'une feuille organisée en lignes et colonnes ; une ligne pour chaque enregistrement, une colonne pour chaque champ. La feuille peut être personnalisée, largeur et position des colonnes, hauteur des lignes, changement de police, colonnes figées ou libérées, colonnes masquées ou visibles. Les changements de présentation de la feuille ne peuvent pas être sauvegardés.
Feuille de réponse (requête)	Les données trouvées par une requête sont affichées dans une feuille de réponse organisée en lignes et colonnes. Celle-ci ressemble à une table, mais n'en est pas une. Il s'agit d'une vue dynamique de données appartenant aux tables sources de la requête. Les mises à jour apportées dans la feuille de réponse sont répercutées dans les enregistrements correspondant des tables sources, sous certaines conditions. La présentation de la feuille peut être personnalisée comme celle d'une feuille de données d'une table.
Formulaire	Un formulaire présente les données d'une table ou d'une requête sous une forme personnalisée. On utilise habituellement des formulaires pour saisir, modifier et mettre à jour plus facilement les données

Macro	Une macro (ou macrocommande) est une séquence d'actions enregistrées qui peut être exécutée automatiquement. Une macro peut servir à exécuter automatiquement des manipulations répétitives, ou exécuter une tâche en réponse à un événement (validation avant sauvegarde d'un enregistrement, ouverture d'un formulaire en cliquant sur un bouton de commande...).
Relation	Pour manipuler des données de plusieurs tables, il est nécessaire d'établir une relation entre ces tables, deux à deux. La relation est construite sur un champ de la première table et un champ de la deuxième table ; elle met en correspondance les enregistrements des deux tables qui ont la même valeur dans ces champs. La relation peut être définie au niveau des tables elles-mêmes, ou bien être établie au niveau de la requête.
Requête	Une requête est la formulation d'une interrogation sur le contenu d'une ou de plusieurs tables entre lesquelles sont établies des relations. Le résultat d'une requête est une feuille de réponses.
Table	Une table contient des données relatives à une certaine catégorie d'information (les employés, les factures...). Un enregistrement contient les données d'un élément de la catégorie (un employé, une facture...). Une tacle peut s'afficher sous forme d'une feuille de données dont les lignes sont les enregistrements et les colonnes les champs.

Chapitre 2 : Exemple pratique

Nous allons raisonner sur un cas très simple, peut-être même trop simple aux yeux de certains : celui d'un carnet d'adresse... Pour compliquer (très légèrement) la base de données, nous ferons en sorte que l'utilisateur puisse extraire ses adresses par département (en fait, un département à la fois), par exemple pour imprimer un listing...

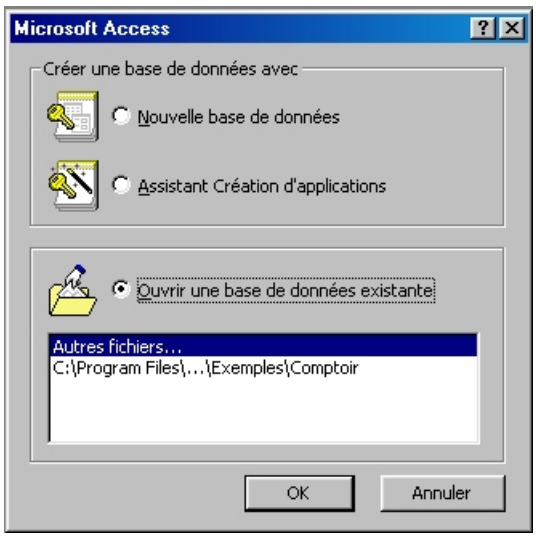
L'étude préliminaire, menée à partir du cahier des charges, a montré que, pour chaque individu, on devait enregistrer...

un titre de politesse
le nom de famille
le prénom
l'adresse
le code postal
le bureau distributeur
le numéro de téléphone
une zone de commentaires « libres »

Attention : ce que nous appelons ici l'adresse sera en réalité un « espace » formé de trois lignes de texte. L'expérience montre que cela permet de faire face à la quasi totalité des cas particulier

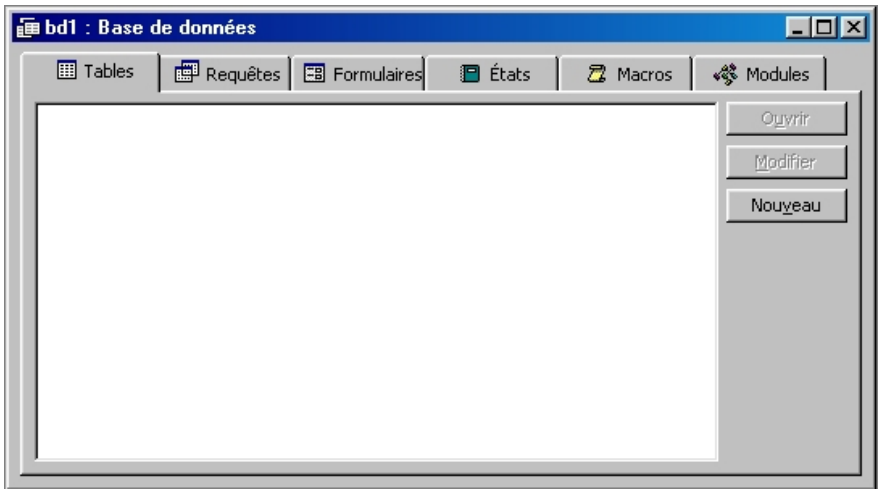
2.2.1 - Comment créer une base (vide)

A son ouverture, le logiciel ACCESS ne vous laisse pas beaucoup de choix, il vous demande immédiatment si vous voulez utiliser une base existante, ou en créer une nouvelle...



Le bon sens suffit pour deviner que la commande à sélectionner est ici la première. Elle ouvre une nouvelle fenêtre sur l'écran, afin de permettre à l'utilisateur de donner un nom à sa nouvelle base, et à indiquer à quel emplacement du disque il désire la placer... Cette fenêtre est pratiquement identique à celle que l'on retrouve avec la plupart des logiciels actuels en lançant la commande « Enregistrer Sous... ». Elle se passe donc de commentaires.

Une fois cette opération terminée, la base est créée. Bien évidemment, elle est vide : elle ne contient rien, ni table, ni requête, ni formulaires... Rappelons à cette occasion que les divers éléments d'une base de données Access sont tous enregistrés à l'intérieur d'un même fichier. Sur son disque, l'utilisateur ne verra donc que le fichier qu'il vient de créer ici. En ce qui nous concerne, nous ne chercherons pas à savoir comment tous ces éléments sont positionnés les uns par rapport aux autres dans ce fichier...



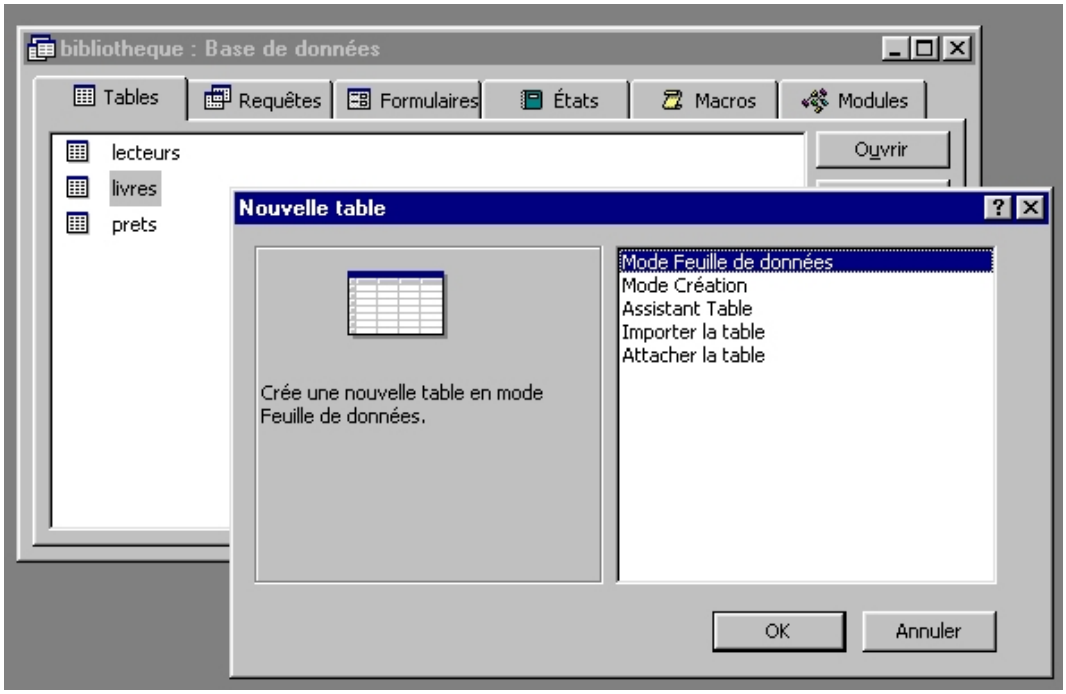
Notons tout de suite que le « travail » dans la base peut se décomposer en six grands « domaines », qui correspondent respectivement à un accès aux tables, aux requêtes, aux formulaires, aux états, ainsi qu'aux macro-commandes et aux « modules » de programmation:

les tables	contiennent les données. Elles se présentent un peu comme les feuilles d'un tableur
les requête	permettent de sélectionner, de trier, de classer... les données. Ce sont en fait des « interrogations » de la base de données
les formulaires	permettent d'afficher sur l'écran les données d'une table (ou d'une requête). Très utiles pour l'introduction des données
les états	sont destinés à être imprimés. Le plus souvent, il s'agira de « listes » pouvant inclure des calculs simples.
les macrocommandes et les modules	permettent de développer des applications plus complexes

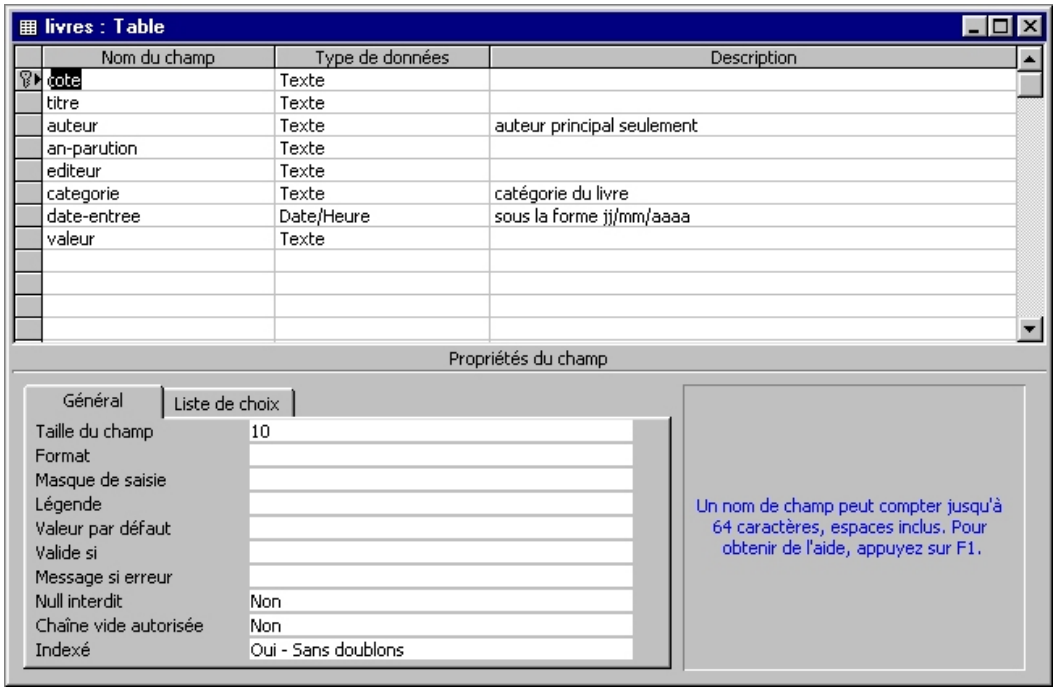
Comme il n'est pas question ici de faire de la programmation, mais uniquement une courte initiation à un logiciel de gestion de base de données (en prenant Access comme exemple), le cas des macro-commandes et des modules de programmation ne sera pas évoqué

2.2.2 - Comment créer une table (vide)

La création des table s'effectue à partir de l'écran précédent, qui va devenir, en quelque sorte, le point central de la base, la « plaque tournante » à partir de laquelle on va désormais travailler (beaucoup plus qu'à partir de la barre des menus de la fenêtre principale). Pour y aboutir, il suffit de s'assurer qu'on se trouve bien dans la fenêtre « tables », et de cliquer sur « nouveau ». On obtient ainsi un écran vierge... qu'il n'y a plus qu'à remplir... Seulement, à tous les niveaux du logiciel, l'utilisateur va rencontrer des « assistants ». Beaucoup d'entre eux sont utiles, d'autres moins. Dans le cas présent, il est souvent préférable de travailler « manuellement », en sélectionnant le mode dit « création »...



En principe, la phase d'analyse du travail à faire a permis de déterminer le type des champs à créer, ainsi que leur taille. Il est probable que le nom de ces champs est également déjà déterminé. Il ne reste donc plus qu'à remplir le tableau proposé.



Attention, ce n'est pas parce que le logiciel autorise des noms de 64 caractères qu'il faut les utiliser. Au contraire, il serait fastidieux d'utiliser des noms trop longs. De plus, il faut garder en tête qu'un jour où l'autre, il sera utile d'exporter des données vers d'autres logiciels. Il est donc conseillé de s'en tenir à des prescriptions « minimales », à savoir des noms courts, et sans espaces (par exemple en les remplaçant par le caractère de soulignement). Notez au passage que la fenêtre d'aide en bas de l'écran (à droite) correspond à la colonne dans la quelle se trouve le curseur. On peut ainsi avoir les « renseignements » suivants :

* Un nom de champ peut compter jusqu'à 64 caractères,
* Le type de données détermine les valeurs que l'utilisateur peut stocker dans le champ,
* La description du champ est optionnelle.
* ... ou d'autres indications...

Enfin, autre conseil, même si la description du champ est optionnelle, il est toujours intéressant de la faire figurer. N'hésitez jamais à « documenter » au maximum votre travail. Il faut faire en sorte que celui qui sera amené à regarder votre base de données puisse comprendre ce que vous avez voulu faire.

Voici donc la description de la table que nous voulons créer :

Nom du champ	type	taille	description
politesse	texte	32	le titre de politesse
nom	texte	120	le nom de famille
prenom	texte	120	le prénom
adr_1	texte	120	l'adresse : ligne 1
adr_2	texte	120	l'adresse : ligne 2
adr_3	texte	120	l'adresse : ligne 3
copost	texte	10	le code postal
bureau	texte	120	le bureau distributeur

phone	texte	25	le numéro de téléphone
observ	texte	120	un commentaire libre

Dans le tableau « vierge », il n'apparaît que trois colonnes. En effet, la taille du champ doit être saisie dans le tableau, qui apparaît en bas à gauche en fonction du type de champ que l'on vient de choisir. Par exemple, pour la première ligne, inscrivons « politesse » comme nom du champ. Dès que nous « cliquons » dans la seconde colonne pour donner le type, le mot « texte » apparaît immédiatement, avec une flèche destinée à ouvrir un menu déroulant, ainsi que la grille à laquelle nous venons de faire allusions. Notez tout de suite que le type par défaut d'un champ est le type « texte ». Dans notre exemple, nous n'avons donc pas à chercher plus loin. De plus, pour l'instant, la principale indication à modifier dans la grille de définition du champ est sa taille (par défaut, égale à 50 caractères).

A l'aide du tableau ci-dessus, il serait donc assez rapide de remplir la grille de définition de notre première (et unique) table. Seulement attention, les choses ne sont pas toujours aussi simples qu'elles peuvent paraître.

Selon la version du logiciel utilisé, vous pouvez avoir, en bas de la grille de définition des champs, trois lignes importantes : **null interdit**, **chaîne vide autorisée** et **indexé**. Dans notre exemple, il faudra veiller d'une manière générale à ne pas interdire le « null », c'est à dire à autoriser la présence d'une « chaîne vide » (d'une absence de données), et à ne pas indexer le champ. Ceci n'aura que trois exceptions :

* le champ **politesse** : la chaîne vide ne sera pas autorisée... (réponse *NON*)
* le champ **nom** : la chaîne vide ne sera pas autorisée... (réponse *NON*)
* le champ **copost** : il sera « indexé avec doublons »... (réponse *OUI - avec doublons*)

Pour les deux premiers, cela signifie tout simplement que nous nous interdisons (dans la structure de la base) de saisir une adresse sans formule de politesse ou sans nom. Pour le troisième champs, cela revient à demander à l'ordinateur de tenir à jour un index permettant de sortir les adresses dans l'ordre des codes postaux. La notion de « **Avec doublons** » autorise à utiliser plusieurs fois le même code dans la table. Si on le désire, on peut aussi indexer le champ nom, mais, pour notre exemple, ce n'est pas nécessaire.

Une fois cette grille remplie, pour créer la table, il suffit d'en enregistrer la définition, par la commande **Enregistrer sous...** du menu **fichier**[8]. Sauf problème particulier, le logiciel doit alors vous poser une question. En effet, chaque table peut comporter une « **clé primaire** » (c'est à dire un index SANS doublons). C'est même fortement recommandé. Étant donné qu'aucun champ de notre table ne possède cette propriété, le logiciel demande s'il doit créer cette « clé primaire ». Répondez « **oui** ».

Aux champs existant, le logiciel ajoute un champ nommé N°, de type « compteur » ou « numero automatique », selon la version utilisée. Ce champ sera automatiquement incrémenté d'une unité à chaque introduction d'une nouvelle adresse, créant ainsi un « identifiant unique » : chaque adresse va avoir un numéro différent... Attention, un champ de type « compteur » est obligatoirement une clé primaire...

[8] - N'oubliez pas que vous ne verrez jamais cette table sur votre disque : elle est enregistrée quelque part à l'intérieur de votre base de données, qui sera le seul « élément » visible « de l'extérieur »...

2.2.3 - Comment créer un formulaire

Un formulaire est, à peu de chose près, une grille que l'on va faire apparaître sur l'écran afin de pouvoir accéder à une table autrement qu'en mode « tableau ». C'est aussi l'équivalent du formulaire « papier » que l'on vous demande de remplir dans de multiples occasions, avec chaque champ de la table représenté par une ligne sur laquelle on doit écrire, ou par un rectangle...

Ici, nous allons faire appel à l'un des (très et parfois trop) nombreux « assistants » de MS.Access. Pour cela, il faut d'abord cliquez sur l'onglet « formulaire », puis sur la commande « nouveau ». Ensuite, nous indiquons que vous voulons utiliser la table « adresses », en faisant appel à l'assistant (ne demandez pas à avoir un formulaire vierge : vous auriez effectivement une « feuille vierge », sans rien dessus !). Le modèle le plus simple est celui dit en « **formulaire instanténé : Colonnes** ». Comme son nom l'indique, il va mettre tous vos champs sous forme d'une simple colonne. Autre point intéressant, ce modèle est aussi celui qui se prête le mieux à des modifications[9], et donc à une « personnalisation »...

En règle générale, une fois le modèle sélectionné, l'assistant vous guide pas à pas dans la construction de votre formulaire. Vous indiquez les champs à faire figurer, ainsi que leur ordre, puis le type de présentation (choisissez « standard »), avant de donner un nom à votre formulaire, et de confirmer au logiciel que c'est terminé. En quelques secondes, vous obtenez un formulaire tout à fait correct. Il ne vous reste plus qu'à l'utiliser pour remplir votre table...

Attention : n'oubliez pas de l'enregistrer, par la commande **Enregistrer sous...** qui se trouve dans le menu **fichier**[10]. Sinon, vous risquez de le perdre. Dans notre exemple, nous pourrions l'enregistrer sous le nom de « Saisie des adresses ».

Bien évidemment, ce formulaire standard n'est certainement pas le magnifique formulaire dont vous rêvez, mais il est toujours possible, dans une seconde étape de le modifier pour l'adapter à votre cas particulier. Toutefois, n'oubliez pas que cela demande des connaissances un peu plus poussées, et que cela risque de vous prendre du temps. Si vous êtes pressé, conservez votre formulaire tel qu'il est. Si vous voulez le transformer en une œuvre d'art, prenez quelques heures sur votre temps libre pour apprendre comment modifier un formulaire existant.

2.2.4 - Comment créer une requête

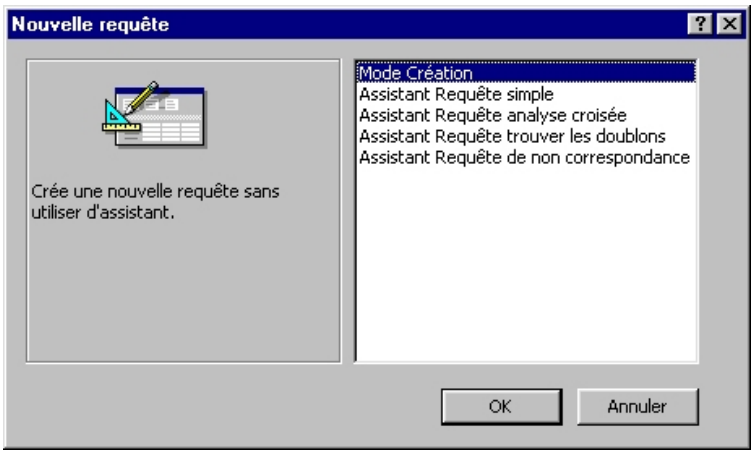
Une requête peut être considérée comme une sorte d'interrogation, préparée à l'avance, ou encore comme une sorte de « filtre » que l'on va appliquer à une table. De plus, cette interrogation peut porter sur une table unique, ou sur plusieurs tables liées entre elles. Ce lien entre deux tables doit être fait entre deux champs (un pour chacune des tables) représentants une même « caractéristique », et il va permettre d'obtenir une table virtuelle, formée de tous les champs sélectionnés (lors de la création de la requête) dans l'une et l'autre tables.

Etant donné que notre « base de donnée » ne comporte qu'une table, nous n'envisagerons ici que l'aspect « interrogation » (ou filtre) sur une table. Pour commencer, demandons la feuille « Requête » dans la « plaque tournante » du logiciel, puis «nouveau» pour signaler que nous voulons créer une nouvelle requête.

[9] - Mais pour cela, consultez le mode d'emploi de votre logiciel, ou la fonction d'aide.

[10] - N'oubliez pas que vous ne verrez jamais cette table sur votre disque : elle est enregistrée quelque part à l'intérieur de votre base de données, qui sera le seul « élément » visible « de l'extérieur »... Par la suite, nous ne rappellerons plus que chaque élément de la base doit être enregistré dans la base. De toute façon, en cas de besoin, le logiciel se charge de vous le rappeler avant de vous laisser « fermer » une fenêtre.

Si on utilise la version 1 du logiciel MS.Access, on obtient immédiatement la liste des « tables » utilisables. Notez qu'il s'agit aussi bien des **tables réelles** (celle que nous avons créées) que des **tables virtuelles**, définies par une requête : n'oubliez pas que le résultat d'une requête est une table (presque) comme les autres, avec toutes les propriétés qui s'attachent à ce type d'objet. Par contre, si on utilise une version de ce même logiciel, il va proposer l'utilisation d'un assistant, destiné à guider l'utilisateur pas à pas vers la solution cherchée...

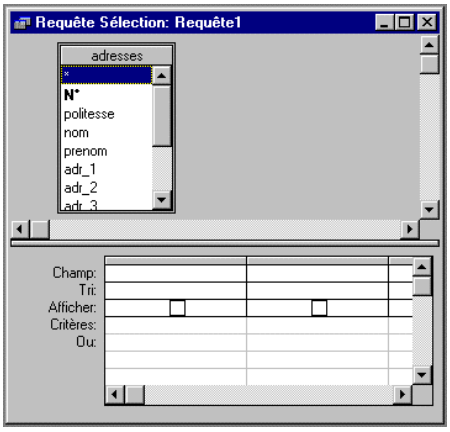


Dans le cas présent, l'utilisation systématique d'un assistant est déconseillée, car il est très simple d'écrire soi-même les éléments de sa requête. Lorsque nous aurons confirmer le choix du « mode création », le logiciel va proposer de choisir la ou les tables (table **réelle**, crée comme celle qui précède**,** ou table **virtuelle**, résultant d'une requête comme celle que nous sommes en train de réaliser) que l'on veut mettre dans notre nouvelle requête.

Notez au passage ce que nous venons de dire : le résultat d'une requête est similaire à une table, et en a toutes les propriétés, sauf une : elle n'a pas d'existence « physique », et n'est « construite » (par le logiciel) qu'au moment de son utilisation.

Dans notre cas particulier, bien évidemment, nous n'avons pas beaucoup de choix : notre base de données ne contient encore qu'une seule table. Le choix est donc vite fait... Nous choisissons d'ajouter la table « **adresses** » à notre future requête, (en cliquant sur le nom de la table, puis sur « ajouter », et nous fermons cette fenêtre...

L'écran obtenu doit à présent ressembler à ceci :



Il ne reste plus qu'à faire sélectionner les champs que l'on désire voir paraître dans la requête, et à les placer dans la ligne « champ » du cadre inférieur. La méthode la plus simple est d'effectuer un « **glisser-déplacer** », mais on pourrait aussi cliquer dans la case de la première ligne, ce qui fait apparaître une flèche sur la droite de cette même case. Cettez flèche donne alors accès à un menu déroulant donnant toutes les variables (tous les champs) de la (ou des) tables(s) sélectionnée(s). En fait, ici, nous allons prendre tous les champs, l'un après l'autre... Pour les voir tous, il faut parfois utiliser l'ascenseur verticale placé le long de la table, tandis que l'ascenseur horizontal en bas de la partie inférieure de le fenêtre contenant les différentes colonnes où l'on place les champs permet de se déplacer de l'une à l'autre. Attention : le logiciel refuse les colonnes vides (sauf après le dernier champ sélectionné). Il faut donc les remplir de gauche à droite, dans l'ordre où l'on veut voir figurer les champs, et sans laisser de « vide »...

Attention : il serait possible d'utiliser l'étoile qui figure en tête de la liste des champs, qui signifie effectivement « tous les champs », mais cela rendrait impossible certaines opérations sur des champs particuliers. Une méthode donc, à n'utiliser qu'avec précaution.

N'oubliez pas d'enregistrer votre requête en cours d'écriture, sans attendre qu'elle soit terminée. Ceci se fait d'abord par la commande « **enregistrer Sous... »** qui permet de donner un nom à la requête, puis, les fois suivantes, en cliquant sur l'icône d'enregistrement.

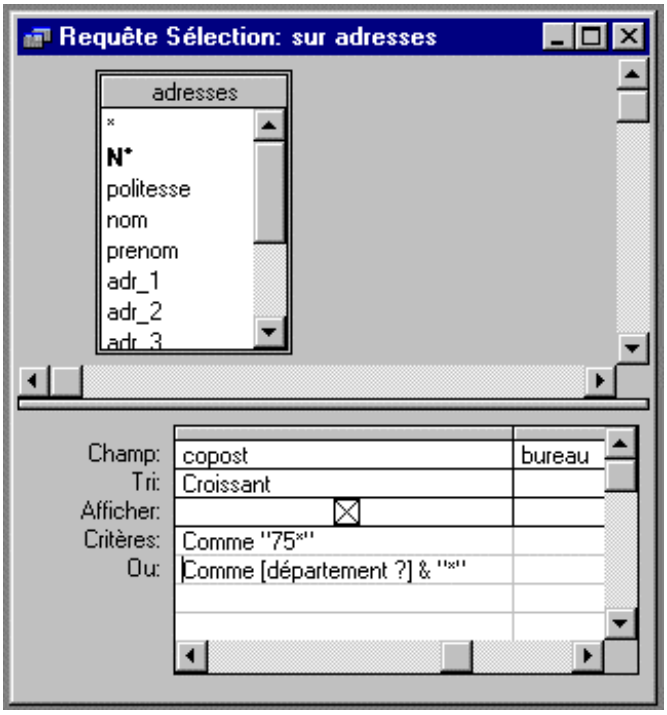
Quelles « contraintes » peut-on imposer aux champs de cette requête, et comment ? Le plus simplement du monde, en utilisant les lignes situées en dessous du nom du champ. Regarder par exemple la colonne consacrée au champ « **copost** » (c'est à dire au code postal). La ligne «tri» permet de trier la table sur ce champ, en ordre croissant ou décroissant; la ligne « afficher » permet de faire figurer, ou non, le champ sur les écrans (le champ existe toujours, mais il devient invisible). La ligne « critère » permet d'imposer des critères particulier de sélection. Les possibilités sont très nombreuses, et trop nombreuses pour qu'il soit possible de les évoquer toutes ici. Nous nous contenterons de deux exemples...

Comme «75*»

va « laisser passer » tous les codes postaux qui commencent par 75...

Comme [département ?] & «*»

va d'abord ouvrir une **boite de dialogue** pour demander la valeur à donner à l'expression entre **crochets**, puis va ajouter une étoile à cette valeur. Le symbole **&** que l'on voit ici est celui de la concaténation des deux chaînes de caractères, celle que l'on va donner et l'étoile. Ainsi, si on tape ***49***, l'ordinateur comprend ***49****, c'est à dire tous les codes postaux commençant par « 49 » **.** On se retrouve alors dans un cas analogue au précédent, mais avec la possibilité de « choisir » le département que l'on veut éditer...



Attention, la disposition sur deux lignes différentes correspond à une **fonction OU**. Les deux critères indiqués ici, inscrits dans deux lignes différentes, ne « laisseront passer » que les adresses dont les codes postaux commencent par 75, **ou** par le département indiqué dans la boite de dialogue évoquée plus haut... En outre, en raison de la ligne « tri », ces adresses seront données dans l'ordre croissant des codes postaux.

Si au contraire, on désire une liste triée sur les noms, il faut faire disparaître l'indication « croissant » de cette colonne, et la mettre dans la colonne correspondant au nom. Pour cela, la technique est simple : on clique dans la cellule, cela fait apparaître une flèche, sur la droite, qui elle-même donne accès à un menu déroulant. Il ne reste plus à l'utilisateur qu'à choisir (non trié, croissant, décroissant) en cliquant sur l'option désirée.

2.2.5 - Comment créer un état

Un état est, à priori, un document destiné à être imprimé, contrairement au formulaire qui est en principe fait pour être visualisé sur un écran. En d'autres terme, le but de ces deux moyens de diffusion de l'information est identique, au choix du support près. Cela est tellement vrai qu'il est toujours possible d'imprimer un formulaire, ou de regarder un état sur l'écran de l'ordinateur (par exemple par un « **aperçu avant impression** »).

La création d'un état ressemble fort à celle d'un formulaire. Il faut d'abord décider de lutilisation d'un assistant... Le plus simple est « Etat instantané : Tableau ». C'est aussi le plus utilisé. Dans des cas plus complexe, l'assistant « Assistant Etat », le plus général, pourra vous permettre de résoudre la plupart de vos problèmes.

De question en question, cet assistant va vous guider vers un résultat qui, s'il n'est pas exactement ce que vous voulez obtenir, pourra être modifié assez facilement. Si le résultat est vraiment trop éloigné de votre idée, n'hésitez pas à recommencer toute le démarche : il est presque toujours plus simple, de tout recommencer que de tenter de modifier un état mal conçu au départ.

Attention : n'oubliez pas que votre état ne pourra utiliser que les champs contenu dans votre table ou dans votre requête. Il faut donc savoir ce que vous voulez imprimer dès que vous commencer à concevoir votre requête.

2.2.6 - Maintenance d'une base

Lorsque vous ajoutez des données dans une table, le volume de la base de données augmente. C'est normal, et facile à comprendre. Par contre, lorsque vous supprimez des données, la taille de la base ne diminue pas ! Habituellement, il en est de même si vous supprimer un formulaire ou un état : la place occupée pas votre base reste inchangée... En réalité, les suppressions que vous faites, ne sont pas de véritables suppressions, mais en quelque sorte la mise en place d'un index qui indique que tel ou tel enregistrement n'est plus valable, et de doit plus être affiché... Si vos données sont de type « permanent », ce phénomène n'a pas beaucoup d'importance. Par contre, si vous faites de fréquentes modifications, vous serez conduit un jour ou l'autre à « compacter » votre base de données, c'est à dire, en quelque sorte, à la « nettoyer », à la purger de tous les « résidus » indésirables qui se sont accumulé au fur et à mesure des utilisations.

Cela se fait avant d'ouvrir une base, à partir du menu fichier, en choisissant l'option « compacter une base de données ».

En principe, l'opération est sans danger, mais, comme toujours, il est fortement conseillé une sauvegarde de votre base avant de la compacter : personne n'est à l'abri d'une fause manoeuvre, d'une coupure de courant, ou de n'importe quel incident... Sachez aussi que rien ne vous empêche de compacter une base en conservant son nom : la base « nettoyée » prendra la place de l'ancienne, et les utilisateurs ne se rendront compte de rien... Vous conserverez intacts tous les liens qui peuvent exister entre la base de donnée et d'autres logiciels (comme par exemple un traitement de texte pour effectuer du publipostage).

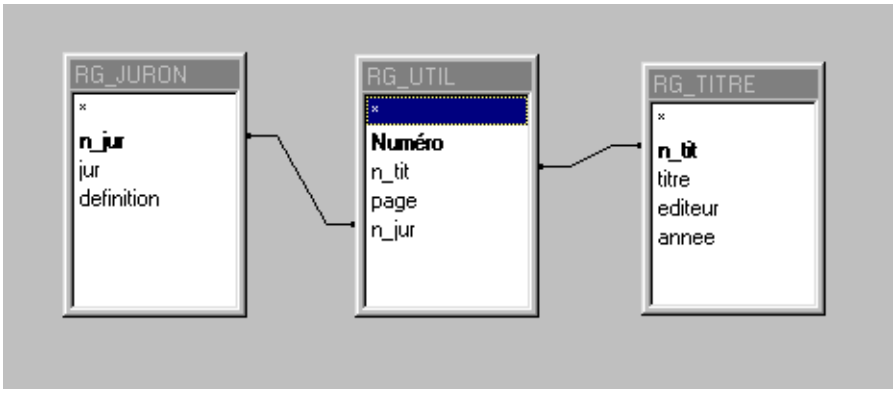
Chapitre 3 : D'autres conseils...

L'exemple précédent a porté sur une base de données formée d'une seule table. C'est naturellement un cas possible, mais dans ce cas, n'importe quel gestionnaire de données peut faire l'affaire. Point n'est besoin d'un logiciel « relationnel ».

En complément de ce qui précède, nous allons aborder dans ce chapitre quelques caractéristiques propres à un « **Système de Gestion de Base de Données Relationnelle** ».

2.3.1 - La base de données

Il va s'agit tout simplement de la base évoquée plus haut, dans la première partie, concernant les jurons du Capitaine Haddock. Toutefois, les noms des tables ont été changés. Rappelons que sa structure doit être la suivante[11] :



La création des trois tables ne pose pas de problème particulier : que l'on veuille créer une table ou plusieurs, la méthode reste la même. Il suffit de l'appliquer autant de fois que l'on veut créer de table...

2.3.2. - Définir des relations entre les tables

Pour le logiciel Access, une relation s'établit à priori entre la clé primaire d'une table principale, ou « source », et un autre champ d'une seconde table. Dans le cas des premières versions du logiciel, cela se faisait par la commande « relation » du menu « édition ». A condition d'avoir pris un certain nombre de précautions, le logiciel pouvait même suggérer des relations. Il proposait automatiquement de créer un lien entre le champ clé primaire de la table « source » et un champ portant le même nom dans la table « destination ». Toutefois, il était nécessaire que ces deux champs soient de même nature. On comprend facilement pourquoi : il est impossible d'établir des liens entre des objets différents, comme des nombres et du texte, des nombres entier et des nombres réels...

Toutefois, ces relations ne respectaient pas obligatoirement les contraintes de l'intégrité référentielle. Pour cela, il fallait le demander au logiciel, en cochant une case dans la fenêtre de travail.

Avec les dernières versions, le logiciel crée automatiquement des relations entre tables d'une requête, à conditions que les conditions vues plus haut soient respectées. Toutefois, ces relations (valables uniquement pour la requête où elles ont été mises en place) ne respectent pas les règles de l'intégrité référentielle. La contrainte précédente subsiste. Seule la création d'une relation permanente entre deux tables permet de respecter cette règle.

Cette création est un peu plus complexe que dans les premières versions. Voici de que dit à ce sujet la fonction d'aide d'Access :

1 Fermez toutes les tables que vous avez ouvertes. Vous ne pouvez pas créer ou modi-

[11] - L'étoile qui figure en haut de chacune des tables est un « raccourcis » permettant de sélectionner tous les champs en même temps, ainsi que nous l'avons déjà évoqué en parlant des requêtes dans le chapitre précédent.

fier des relations entre des tables ouvertes.

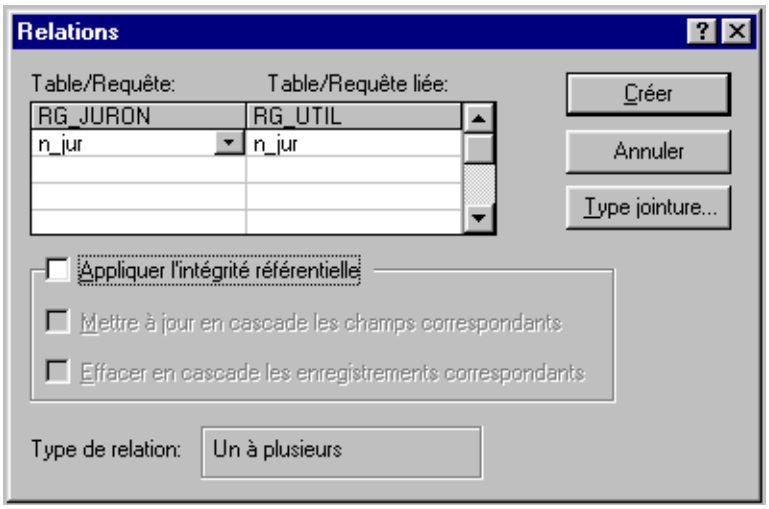
2 Dans la Fenêtre Base de données, cliquez sur dans la barre d'outils.

3 Si vous n'avez défini aucune relation dans votre base de données, la boîte de dialogue Ajouter tables/requêtes s'affiche automatiquement. Si vous devez ajouter les tables que vous voulez relier et que la boîte de dialogue Ajouter tables/requêtes ne s'affiche pas, cliquez sur dans la barre d'outils. Si les tables que vous voulez relier sont déjà affichées, passez directement à l'étape 5.

4 Cliquez deux fois sur le nom des tables que vous souhaitez relier, puis fermez la boîte de dialogue Ajouter tables/requêtes.

5 Faites glisser à partir d'une table le champ que vous souhaitez relier au champ associé d'une autre table.

Dans la plupart des cas, vous reliez le champ Clé primaire (qui est affiché en gras) d'une table avec un champ correspondant d'une autre table (souvent du même nom) appelé la clé externe. Les champs reliés ne doivent pas obligatoirement avoir le même nom, mais ils doivent posséder le même type de données (à une exception près) et contenir le même type d'informations. En outre, lorsque les champs correspondants sont des champs de type Numérique, ils doivent avoir le même paramétrage (réel, entier, etc...). La seule exception à cette règle est la possibilité de relier un champ « **Numérotation automatique incrémentielle** » (c'est à dire un champ « compteur ») à un champ « **Numérique Entier long** ».



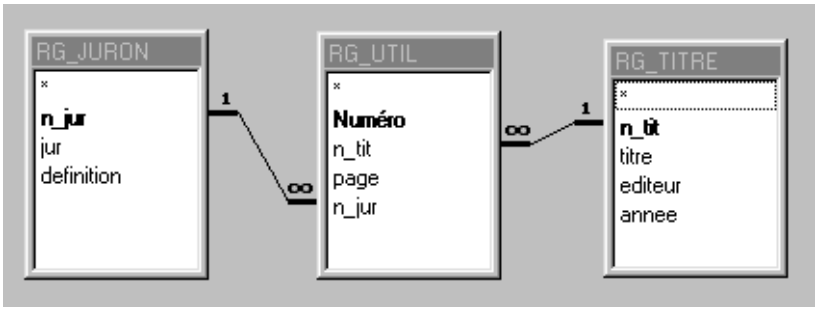
6 La boîte de dialogue Relations s'affiche (voir ci-dessus). Vérifiez les noms des champs affichés dans les deux colonnes et assurez-vous qu'ils sont corrects. Vous pouvez les modifier si nécessaire. Définissez les options de relation si nécessaire. Pour des informations sur un élément spécifique de la boîte de dialogue Relations, cliquez sur ?, puis cliquez sur l'élément.

7 Choisissez le bouton Créer pour créer la relation.

8 Répétez les étapes 5 à 8 pour chaque couple de tables à relier.

Lorsque vous fermez la fenêtre Relations, Microsoft Access vous demande si vous voulez enregistrer la mise en forme. Que vous le fassiez ou non, les relations que vous avez créées sont enregistrées dans la base de données.

Pour savoir si les relations mises en place dans une requête sont liées à cette requête, ou permanente, il suffit de regarder comment elles apparaissent sur l'écran. Comparer la figure du paragraphe 2.3.1 à celle qui suit :



Ici, la nature des relation est clairement indiquée :il s'agit de liens de type 1-à-N.. Cela signifie, par exemple, qu'un numéro de juron, utilisé une seule fois dans la table RG_JURON, peut être utilisé un nombre infini (?) de fois dans la table RG_UTIL.

2.3.3 - Qu'est-ce que l'intégrité référentielle?

L'intégrité référentielle est un système de règles utilisées pour garantir que les relations entre les enregistrements dans les tables reliées sont valides et que vous n'effacez ou ne modifiez par erreur des données liées. Vous pouvez définir l'intégrité référentielle quand toutes les conditions suivantes sont réunies :

Le champ correspondant de la table primaire est une clé primaire ou possède un index unique.

Les champs associés ont le même type de données.

Les deux tables appartiennent à la même base de données Microsoft Access. Si les tables sont attachées, elles doivent être des tables format Microsoft Access, et vous devez ouvrir la base de données dans laquelle elles sont stockées pour définir l'intégrité référentielle. L'intégrité référentielle ne s'applique pas aux tables attachées de bases de données utilisant d'autres formats.

Lorsque l'intégrité référentielle est d'application, vous devez observer les règles suivantes :

Vous ne pouvez pas entrer une valeur dans le champ clé externe de la table liée qui n'existe pas dans la clé primaire de la table primaire. Toutefois, vous pouvez entrer une valeur de type Null dans la clé externe, qui spécifie que les enregistrements ne sont pas liés. Par exemple, une commande ne peut pas être affectée à un client qui n'existe pas, mais elle peut n'être affectée à personne en entrant une valeur de type Null dans le champ N° client.

Vous ne pouvez pas effacer un enregistrement de la table primaire si des enregistrements correspondants existent dans une table liée. Par exemple, vous ne pouvez pas effacer l'enregistrement d'un employé dans la table Employés s'il y a des commandes affectées à celui-ci dans la table Commandes.

Vous ne pouvez pas changer une valeur clé primaire dans la table primaire si cet enregistrement possèdent des enregistrements liés. Par exemple, vous ne pouvez pas changer le N° d'un employé dans la table Employés si des commandes lui sont affectées dans la table Commandes.

Si vous souhaitez que Microsoft Access applique ces règles pour une relation, sélectionnez la case à cocher **Appliquer l'intégrité référentielle** lorsque vous créez la relation. Si l'intégrité référentielle est appliquée et que vous ne respectez pas une des règles pour des tables liées, Microsoft Access affiche un message et ne permet pas la modification.

2.3.4 - Travailler sur plusieurs tables

Cette création d'une requête ne pose pas plus de problème qu'à partir d'une seule table. En fait, à partir du moment où les trois tables sont liées, on peut considérer qu'elles constituent *une table virtuelle*, dont il est possible d'utiliser le contenu exactement comme celui d'une table réelle... En ce qui concerne les états, le problème est exactement le même : la requête se comporte exactement de la même manière d'une table.

Toutefois, il peut se poser des problèmes lorsqu'on tente de réaliser un formulaire à partir d'une requête, en particulier lorsqu'on désire l'utiliser pour faire de la saisie. Il faut alors avoir recours à des formulaires plus complexes, réalisés à partir d'un formulaire dédié à la table principale et incluant des sous-formulaires pour les autres tables mais ceci est une autre histoire... en dehors des limites de ce document d'initiation.

Complément : Information et Entreprise

Chapitre 1 : Information...

- 3.1.1 - Les données et l'information
- 3.1.2 - Les qualités de l'information
  - L'information doit être utile
  - L'information doit être exacte
  - L'information doit être complète
  - L'information doit être ponctuelle
- 3.1.3 - Les sources d'informations
  - Les sources primaires
  - Les sources secondaires
  - Les sources internes
  - Les sources externes
- 3.1.4 - L'information dans le temps
  - La régularité
  - La période
- 3.1.5 - La présentation de l'information

Chapitre 2 : ...et entreprise.

- 3.2.1 - L'entreprise en tant que systèmee
  - Le système opérant
  - Le système de pilotage
  - Le système d'information
- 3.2.2 - Les fonctions du système d'information
  - La génération des informations
  - La mémorisation des informations
  - La communication et la diffusion
  - L'exécution des traitements
- 3.2.3 - La constitution d'un système d'information
  - L'environnement
  - Les intrants
  - Les extrants
- 3.2.4 - Les sources d'information de l'entreprise
  - La comptabilité et les finances
  - Le marketing
  - Les ressources humaines
  - La production
  - Le management

L'idée principale de cette partie est de montrer la différence entre système d'information et système informatique. Le second n'est qu'un ensemble matériel au service du premier. Les systèmes de gestion de base de données sont un des outils dont disposent les responsables des systèmes informatiques...

De tout temps, les entreprises et les gestionnaires ont dû disposer d'informations afin d'être en mesure de prendre les meilleures décisions possibles dans les domaines qui les intéressent. Le besoin d'information n'est donc pas un phénomène récent. Il remonte en fait au début même de la gestion. Ce qui a changé, cependant, ce sont d'une part la quantité d'information accessible et d'autre part les outils de traitement de cette information.

Les entreprises sont des systèmes complexes qui utilisent plusieurs types de ressources. Ces « intrants » peuvent se présenter sous forme de matière premières, de ressources humaines, technologiques ou financières. Mais cette liste demeure incomplète tant qu'on n'y ajoute pas un élément qui prend de plus en plus d'importance dans la gestion moderne : l'information.

L'information peut servir dans la mise au point de l'innovation, dans l'accumulation des connaissances requises par rapport au processus de fabrication, ou dans des recherches sur les marchés potentiels de l'entreprises ou sur ses concurrents. Souvent, elle constitue même le produit final, la raison d'être de certaines entreprises.

Chapitre 1 : Information...

Toute « information » constitue-t-elle vraiment de l'information ? Bien que cette question puisse paraître un peu bizarre, elle garde toute son importance. On utilise, dans le langage de tous les jours, le mot « information » à toutes les sauces. On dit qu'on a reçu de l'information à tel ou tel sujet, que l'on a pris des informations, que tel document fournit une mine d'informations, et ainsi de suite. Sans oublier les « informations » de la radio ou de la télévision...

Mais lorsqu'on traite de l'information, (par exemple en rapport avec la gestion), on doit différencier deux niveaux : le renseignement et l'information. Un renseignement ne constitue pas nécessairement une information.

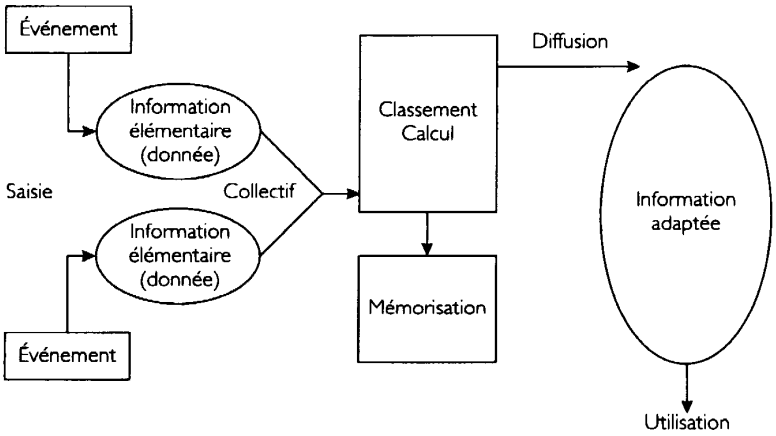
3.1.1 - Les données et l'information

Certains types de renseignements ne sont en fait que des données, des unités de connaissances qui n'apportent rien en tant que telles à celui ou celle qui les possède. **Les données en elles-mêmes ne véhiculent pas de message**. Elles doivent être organisées, traitées, de façon à présenter un sens, avant de pouvoir être considérées comme de l'informations. Ainsi, le fait de savoir qu'il fera -15° à Moscou, aujourd'hui, ne constitue pas, à priori, une information. Cette bribe de connaissance ne m'est d'aucune utilité : je n'ai pas l'intention de me rendre à Moscou, et je ne connais personne qui y soit actuellement... C'est donc un renseignement qu'il peut être amusant de posséder, mais qui demeure un renseignement. Par contre, ce renseignement peut constituer une information utile pour quelqu'un d'autre. Le voyageur dont l'avion s'envole à 13 h en direction de cette ville sera très certainement heureux de connaître ce renseignement : cela l'aidera à décider quels vêtements il va porter. Devant ce renseignement, une autre personne choisira peut-être de différer son voyage.

Il existe donc des milliards de données dont il est possible de disposer, mais cela ne signifie pas que toutes ces données constituent de l'information.

Pour avoir une valeur informative, les données doivent ajouter aux connaissances de l'utilisateur, et présenter une certaine utilité. Une donnée prendra un sens et se transformera en information au moment où elle permettra de mieux comprendre une situation, de se faire une opinion, de prendre une décision. Tant et aussi longtemps qu'un renseignement ne joue pas ce rôle, il demeure tout simplement une donnée de plus à disposition d'un individu ou d'un organisme.

Pour le responsable de formation, la répartition par groupe d'âge des habitants de la région dans laquelle son établissement oeuvre constituera de l'information, parce qu'il pourra se servir de ces données pour être mieux en mesure de prévoir les effectifs des années à venir, de même que leurs besoins de formation. Par contre, un renseignement sur la nouvelle politique de prix d'une chaîne de magasins ne lui sera que de peu d'utilité. Ce qui représente de l'information pour l'un peut très bien n'être qu'un renseignement pour son voisin.



Différents auteurs ont établi de la façon suivante la différence entre les termes « données » et « information » : les données sont des faits ou des chiffres, tandis que l'information est la connaissance acquise par le traitement des données. Une données peut être définie comme la représentation de faits, d'observations ou de répétitions d'un événement. L'information, quant à elle, consiste en des données modifiées en une forme utile et pourvue de sens pour certains besoins particuliers de l'être humain.

Nous pouvons donc affirmer que toute information est une données, mais toute donnée ne constitue pas nécessairement une information. L'information peut se présenter sous différentes formes. Elle peut aussi être quantitative (ou numérique) ; elle indique alors une évaluation précise comme le profit brut d'une entreprise, son ration d'endettement, ou n'importe quel ratio comptable. elle peut aussi être qualitative (ou symbolique), auquel cas on la présente normalement au moyen d'un texte, comme c'est le cas de ce document.

3.1.2 - Les qualités de l'information

L'information doit être utile

Pour qu'un renseignement puisse détenir une information, comme nous l'avons dit plus haut, il doit avoir une certaine utilité pour celui qui le reçoit. Ainsi, pour le gestionnaire d'un marché d'alimentation, le fait de connaître les prix de l'essence à la station-service la plus proche ne constitue d'un renseignement. Par contre, le fait de connaître les prix de vente de son concurrent le plus proche sur les boissons gazeuses est une information particulièrement utile.

L'information doit être exacte

Il nous apparaît évident qu'une information, pour être utile, se doit d'être exacte ; mais cette évidence dans la vie courante prend un tout autre aspect dans le contexte de la prise de décision dans l'entreprise. En effet, une information erronée ou imprécise peur être lourde de conséquences, car le gestionnaire qui considérera cette information comme exacte prendra une décision en fonction de celle-ci, ce qui peut l'amener à faire un mauvais choix. Ainsi, le gestionnaire d'un marché d'alimentation que l'on informerait du fait que son concurrent le plus proche s'apprête à accorder un rabais important sur le prix des boissons gazeuses pourrait très bien décider de la faire lui-même. Il annoncera alors ses prix réduits, avant de se rendre compte que le concurrent en question n'a pas baissé les siens. Il aura donc subi, plus ou moins inutilement, un manque à gagner.

L'information doit être complète

Une information partielle peut être tout aussi dangereuse qu'une information inexacte. Le gestionnaire d'un organisme doit disposer d'une information détaillée pour être en mesure de prendre une bonne décision. Il arrive évidemment qu'on prenne une bonne décision sans connaître toutes les données d'un problème, mais c'est souvent l'effet du hasard, ou de la chance... Pour poursuivre l'exemple précédent, avant de prendre une décision quant à sa réaction à l'attaque de son concurrent, le gestionnaire du marché d'alimentation cherchera à savoir de quel ordre sera la réduction de prix prévue. S'agit-il de quelques francs sur le « pack » de 6 bouteilles, ou de centimes sur celui de 24 ? Il est certain que la réaction de ce gestionnaire dépendra de l'importance de l'action du concurrent.

L'information doit être ponctuelle

Certains renseignements peuvent prendre une valeur au fil des ans. On base souvent les prévisions de ventes sur les ventes des périodes antérieures, on se fie au temps qu'il a fait ces derniers jours pour planifier nos activités des prochains jours, etc. Mais il existe aussi un grand nombre de renseignements qui finissent par perdre toute valeur. Ainsi, le gestionnaire d'un marché d'alimentation se servira-t-il de la courbe des ventes des dernières semaines pour prévoir les ven-

tes de la semaine prochaine ; il consultera peut-être les informations des années antérieures pour tenter d'établir un comportement plus ou moins régulier des ventes selon les saisons ou les périodes de l'année. Ce sont là des informations qui ne souffrent pas du passage du temps, mais tous les types d'information ne connaissent pas ce sort.

A quoi peut-il servir au gestionnaire de notre marché d'alimentation de savoir que son concurrent a fait un rabais éclair, jeudi dernier, sur les boissons gazeuses, et qu'il en a vendu des milliers de caisses ? Evidemment ce renseignement peut être utile, du fait que les ventes effectuées la semaine dernière auront probablement des effets négatifs sur ses ventes futures, mais c'est là une bien mince consolation. Ce renseignement lui aurait été réellement utile s'il l'avait eu avant la promotion en question, de sorte qu'il aurait pu réagir immédiatement, et contrer les principaux effets négatifs de cette action du concurrent.

Il est possible de résumer ce qui précède sous forme d'un tableau :

Qualité	Définition	Exemple
Utile	L'information est pertinente si elle est bien adaptée aux besoins des utilisateurs	Connaître la population d'un quartier pour savoir si on peut y implanter un magasin. Le chiffre d'affaire d'un magasin d'une autre ville n'est pas utile dans ce cas de figure.
Exacte	L'information est digne de confiance si elle vient d'une bonne source, si est bien vérifiée, si elle est à jour...	Une information issue de rumeurs peu vérifiables, ou découlant de calculs approximatifs, n'est pas fiable et peut inciter à prendre de mauvaises décisions
Complète	L'information touche tous les aspects importants pour la compréhension des phénomènes et la prise de décision	Le fait de savoir que les ventes ont augmenté de tel pourcentage ne signifie rien si on ne connaît pas la cause de cette variation
Ponctuelle	L'information est facile à obtenir, requiert peu de manipulation et est accessible directement	Possibilité de retrouver rapidement le chiffre d'affaire par catégorie et par client à l'aide de deux ou trois frappes au clavier de l'ordinateur.

3.1.3 - Les sources d'information

Préalablement à toute utilisation de l'information, les données doivent être **recueillies** et **organisées**. On doit donc déterminer quelles données il faudra recueillir, à quel moment on devra le faire, à quelle fréquence et de quelle façon. Par exemple, pour un marchand, il est très important, lors de chaque vente, de retenir le prix de vente afin d'assurer un suivi des rentrées de fond. Toutefois, ce n'est pas suffisant : il a aussi besoin de savoir quels articles ont été vendus afin de contrôler son inventaire. Si, de plus, les vendeurs reçoivent un pourcentage des ventes, il devra enregistrer le nom (ou le numéro) de chaque vendeur lors des transactions.

L'information provient de multiples sources. Il est important pour une organisation de les identifier et de les connaître. Chaque source possède à la fois des avantages et des inconvénients, et doit être choisie en fonction de l'objectif visé et des capacités de traitement de l'information fournie. La même source peut être utile à plusieurs services. Ainsi l'inventaire fait par les responsables de la gestion des stocks d'une entreprise servira aussi bien au service comptable qu'à celui du marketing et à celui de la distribution...

On distingue deux bases de différenciation de l'information : d'une part les sources primaires et les sources secondaires, et d'autre part les sources internes et les sources externes.

Les sources primaires

L'information provenant d'une source primaire est une information qui a été recueillie dans un objectif particulier. Par exemple, un sondage est une source primaire d'information. Il permet de connaître l'opinion d'un large groupe de personnes sur un sujet précis. Par contre, les sondages sont coûteux, longs à réaliser et la préparation des questionnaires demande un soin minutieux.

L'observation est aussi une source primaire. Cette méthode permet d'éviter les délais ainsi que les biais occasionnés par les intermédiaires. Il faut toutefois faire attention au fait que les résultats peuvent être faussés si l'observation d'un phénomène influence (directement ou non) l'objet observé.

Une troisième source possible est l'expérimentation. Il s'agit d'une façon efficace de maîtriser l'ensemble des conditions entourant un phénomène. Toutefois, il est souvent très difficile de reproduire les conditions réelles d'une situation.

Une autre source primaire d'information consiste tout simplement à demander l'avis d'une personne. L'estimation d'un expert est quelque fois la seule ou la meilleure information possible sur un sujet donné. Il est alors très important de bien choisir sa source d'expertise et de lui décrire avec précision le phénomène.

Enfin, les fichiers des entreprises sont généralement considérés comme une source primaire d'information. C'est la cas, par exemple, des fichiers sur les clients, ou de ceux utilisés dans la gestion des inventaires. Ces fichiers permettent d'accumuler systématiquement l'information nécessaire à la bonne marche des opérations de l'entreprise.

Les sources secondaires

La caractéristique des sources secondaires est qu'elles présentent une information déjà existante, indépendamment de l'objectif fixé par le gestionnaire. Ces sources sont très variées, nous n'en verrons que quelques unes. Une première source secondaire est constituée par la masse d'informations déjà recueillies et emmagasinées par chaque entreprise. Cette information accessible à peu de frais est reliée au contexte particulier de l'entreprise. Elle peut, par contre, se révéler parfois périmée ou difficilement utilisable sous sa forme originale. Elle peut aussi n'avoir qu'un intérêt purement historique.

L'information peut être obtenue de firmes externes. Cela permet d'avoir accès à des renseignements qu'il serait autrement impossible d'obtenir dans des délais et à des coûts raisonnables. De nombreuses firmes se spécialisent dans l'acquisition et la revente de l'information. Certains banques de données sont accessibles par Minitel ou des systèmes informatisés. Un client (en général un abonné) peut alors se servir sur la base à l'aide de son propre ordinateur et d'un modem.

Une troisième source secondaire d'information est l'ensemble des organismes gouvernementaux, nationaux ou européens. Ces organismes fournissent de l'information sur une foule de sujets, en quantité phénoménale, et sous une forme standardisée qui se prête généralement bien aux traitements minimaux qu'il faudra lui faire subir pour l'adapter aux besoins de l'entreprise.

Finalement, une quatrième source d'information provient des publications diverses (publications scientifiques ou techniques, magazines et journaux). Ces publications peuvent aussi fournir une masse d'information à un coût très faible. Cependant, l'information, dans ce dernier cas, est souvent biaisée et peu rigoureuse. Il faut alors la traiter avec soin pour en faire un produit utile. En fait, il s'agira plus souvent d'un indicateur que d'une information.

Les sources internes

Les membres d'une entreprise constituent des sources internes d'information pour une organisation. Les données sont alors les connaissances des individus, les rapports produits par la firme, ou une étude de marché effectuée par des membres du personnel. Notons qu'une source interne d'information peut utiliser des données provenant de l'extérieur de l'entreprise. La source est interne dans la mesure où les éléments ne deviennent de l'information qu'après avoir été saisis par l'organisation. La plupart des données utilisées par les ordinateurs de gestions sont de nature interne : les factures, les relevés de salaires, les relevés de ventes ou d'achats en sont des exemples courants.

Les sources externes

Les sources externes proviennent de l'extérieur de l'entreprise et les renseignement recueillis possèdent déjà un caractère informationnel avant d'être récupérés par l'entreprise. Il s'agit par exemple des publications d'affaires, des publications gouvernementales ou des rapports en provenance des sociétés de consultation.

3.1.4 - L'information dans le temps

La régularité

Une caractéristique importante de l'information et de sa valeur porte sur la régularité ou la fréquence de sa mise à jour. Certaines informations requièrent une mise à jour quotidienne, voire permanents, comme le débit d'une autoroute, alors que d'autres ne peuvent être obtenue d'une ou deux fois par semaine, ou tous les mois, tous les ans, voire encore à plus longue échéance, comme les données des recensements.

La période

Une autre caractéristique de l'information est la période sur laquelle elle porte. Il peut s'agir de données historiques, de données présentes sur les activités de l'entreprise ou de ses concurrentes, ou même de données sur une situation future. Il serait faux de prétendre que les informations historiques ne sont pas aussi pertinentes que les informations actuelles. Les informations historiques qui concernent la même période au cours des deux dernières années peuvent être beaucoup plus utiles que les informations actuelles lorsqu'on veut prévoir les ventes de la période à venir, par exemple.

De même, certaines informations ayant trait à des données futures peuvent être, contrairement à ce que certains croient, tout aussi exactes que des informations actuelles. Ainsi on connaît de façon très précise le nombre d'individus qui auront 16 ans l'an prochain : ce sont ceux qui ont 15 ans cette année... Ce n'est pas parce qu'une information porte sur une période à venir

qu'elle est nécessairement inexacte ou fausse. Certes, il existe toujours une marge d'incertitude lorsqu'on se tourne vers l'avenir, mais cette marge est souvent faible.

3.1.5 - La présentation de l'information

Tous les exemples que nous avons donnés précédemment touchaient des informations d'ordre quantitatif présentées sous forme de données chiffrées. Mais ce ne sont pas les seules informations nécessaire à un gestionnaire d'entreprise.

Il existe tout un éventail d'informations qualitatives qui s'avèrent précieuses pour les personnes qui les possèdent. Ainsi, il sera intéressant de connaître l'opinion des gens sur tel produit qu'on offre, on voudra mieux connaître les préférences des consommateurs en ce qui concerne l'emballage des boissons gazeuses, on cherchera à comprendre comment le consommateur moyen en arrive à prendre une décision d'achat ou de non achat.

De même, la présentation sous forme de données chiffrées ne constitue pas la seule façon de transmettre une information. On peut communiquer le même message par le biais d'un graphique ou d'un dessin. Dans certains cas, cette approche est même préférable à la communication par le seul texte. De nombreux travaux ont tentés de mettre en évidence des relations générales entre la forme de la représentation utilisée et la qualité du processus de compréhension. c'est ainsi que l'on a pu repérer quelques aspects intéressants :

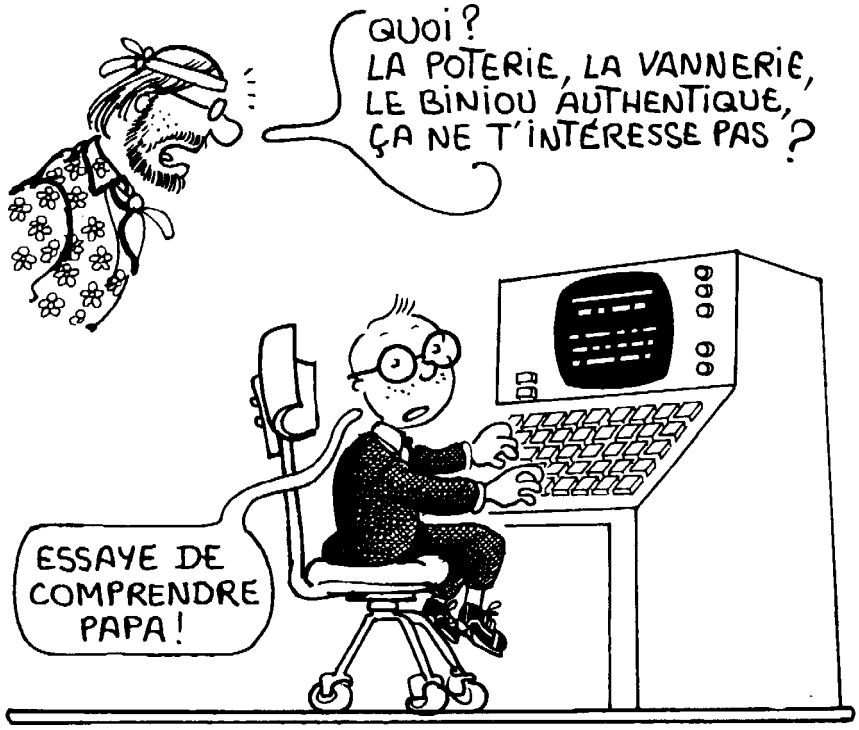
1-/ **L'information présentée sous forme de graphiques est plus rapidement perçue dans sa globalité** que l'information présentée sous forme de tableaux (ce qui ne fait que confirmer le vieil adage connu de tous : un bon dessin vaut mieux qu'un long discours)

2-/ **Les risques de perception erronée sont plus grand lors de l'utilisation de graphiques** que lors du recours aux tableaux de chiffres. Par exemple, le « lecteur » montre une grande sensibilité au choix des échelles dans les graphiques.

3-/ **Il n'est pas possible de définir, d'une manière générale, la « meilleure forme » d'une représentation**. La pertinence formelle dépend des conditions d'utilisation et des caractéristiques du « récepteur », en particulier de son style cognitif (de sa manière de s'approprier l'information).

Les expériences ont montrées que les individus du style « systématique, analytique » qui utilisent une approche méthodique, déductive, des problèmes préfèrent des représentations sous forme de tableau, alors que les individus de type « intuitif, heuristique » apprécient davantage les représentations imagées et les commentaires qualitatifs. La pertinence de la forme est essentiellement subjective.

Ces résultats sont quelques peu fragmentaires, et observés dans des contextes expérimentaux, ils ne peuvent être considérés comme des lois universelles. Il est à peu près certain que la forme utilisée pour les représentations a un impact sur le processus d'assimilation de l'information, mais, en l'état actuel de nos connaissances, peu de conclusions sont scientifiquement validées.


QUOI ?
LA POTERIE, LA VANNERIE,
LE BINIOU AUTHENTIQUE,
ÇA NE T'INTÉRESSE PAS ?
ESSAYE DE COMPRENDRE PAPA !

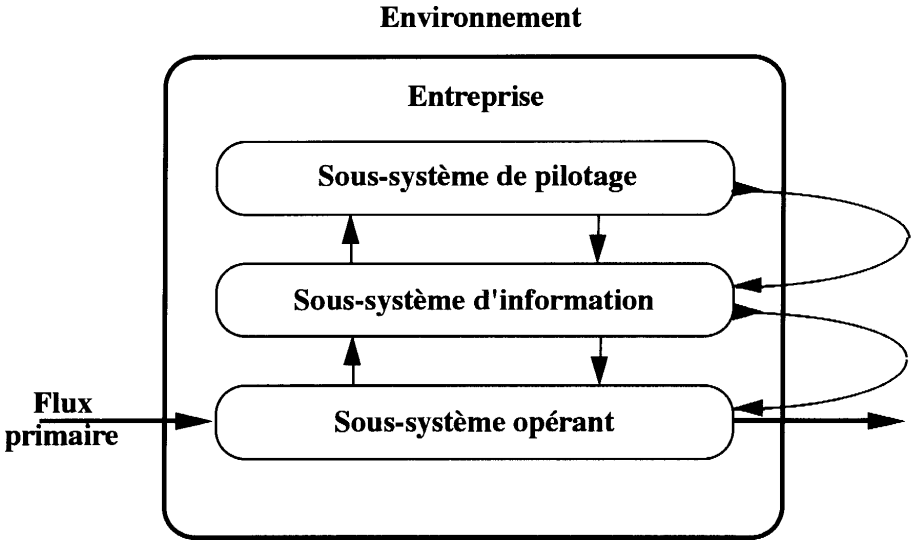
Chapitre 2 : ...et Entreprise

La science des systèmes, appelée aussi systémique, est une discipline scientifique autonome relativement récente, puisqu'elle date de la fin des années 70. La systémique se définit plutôt par son projet que par son objet. Elle prend ses racines principalement dans la théorie des systèmes, la théorie de la commande, la théorie du contrôle, la cybernétique.

La systémique a pour projet « ***la modélisation des phénomènes perçus ou conçus complexes : modélisation, à fin d'anticipation, d'éventuelles interventions intentionnelles et de leurs conséquences enchevêtrées*** ». La science des systèmes a ainsi pour finalité de proposer des modèles pour l'action ou la compréhension d'objets ou de phénomènes complexes dans les domaines les plus variés (biologie, sciences sociales, gestion...).

3.2.1 - L'entreprise en tant que système

La figure suivante illustre la modélisation systémique de l'entreprise, avec ses trois sous-systèmes : système **opérant**, système **d'information** et système **de pilotage**.



Décrivons de façon succincte le rôle de chacun de ses systèmes composant ensemble l'entreprise-système.

Le système opérant transforme et produit

Le système opérant est le siège de l'activité productive de l'entreprise. Cette activité consiste en une transformation de ressources ou flux primaires. Ces flux primaires peuvent être des flux de matière, des flux financiers, des flux de personnel, des flux d'actif ou enfin des flux d'information.

Notons au passage que ce dernier type de flux concernera des informations-matière première qui ne seront pas directement utilisées par l'entreprise à des fins de pilotage (comme dans la presse). Très souvent, dans le secteur tertiaire, la transformation de flux D'information de ce type est la principale activité de L'entreprise. Cela revient à assimiler le système opérant à un système transformant de l'information, on parlera alors de système d'information de production.

Le Système de pilotage réfléchit, décide et contrôle

Le système de pilotage est le siège de l'activité décisionnelle de l'entreprise. Cette activité décisionnelle est très large et est assurée par tous les acteurs de l'entreprise,à des niveaux divers, depuis les acteurs agissant plutôt dans l'activité productrice de l'entreprise,à ceux dirigeant cette dernière. Elle permet la régulation, le pilotage mais aussi l'adaptation de l'entreprise à son environnement. C'est cette activité qui conduira l'évolution, décidera notamment de l'organisation et de l'évolution des systèmes opérants et d'information.

Dans un contexte classique de gestion, cette activité concerne entre autres l'allocation des ressources impliquées (prévision, planification...), ainsi que leur suivi (contrôle de gestion, contrôle budgétaire).

Le système d'information mémorise, traite et diffuse

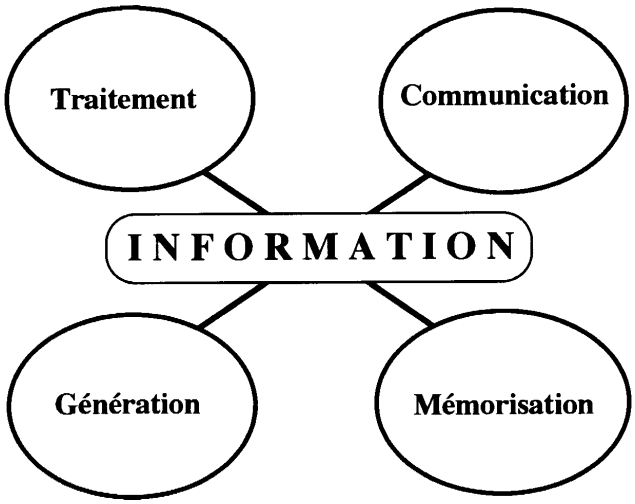
Le système d'information peut être considéré, pour l'instant, comme un système de mèmorisation, d'enregistrement, dont le rôle est de permettre au système de pilotage d'assurer ses fonctions, notamment en assurant son couplage avec le système opérant.

D'après le « Codasyl Developpement Commitee », **un système d'information traite des objets et des événements du monde réel intéressant les utilisateurs de ce système d'information**. Ces objets réels et événements, appelés entités, sont représentés dans le système par des données. L'information relative à une entité particulière se présente sous forme de valeurs qui décrivent qualitativement et/ou quantitativement un ensemble d'attributs qui ont une signification dans le système.

3.2.2 - Les fonctions du système d'information

Dans une entreprise, ce système d'information est destiné...
- au système de pilotage,
  pour pouvoir connaître et maîtriser le fonctionnement du système opèrant,
- au système opèrant ,
  lorsque les flux transformés sont de nature « information »

Le système d'information assure dans l'entreprise, vue en tant que système, les fonctions primaires présentées dans la figure...



La génération des informations

Le système d'information est conçu à l'initiative du système de pilotage. Le système de pilotage doit alors faire preuve d'imagination dans la définition de l'information nécessaire à l'émergence du système d'information et de ses fonctions.

La génération de l'information est une fonction indispensable que le système de pilotage doit exercer pour permettre la conception du système d'information. Cette génération est un préalable nécessaire à toute mémorisation de l'information, elle permettra toute saisie future de l'information et elle est propre à chaque organisation.

Cette génération de l'information consiste à donner à toute information un nom et une définition, reconnus et partagés au sein de l'entreprise. Cela revient en fait à définir le vocabulaire spécifique à l'entreprise. Elle concerne également la définition des événements déclarés d'intérêt pour l'organisation, conduisant aussi à définir les réactions que l'organisation devra développer en réponse à ses événements.

La mémorisation des informations

(transfert des informations dans le temps)

La fonction de mémorisation (collective) des informations a un rôle central : sans mémoire, pas d'apprentissage, pas d'intelligence. La nature et la signification des informations à mémoriser seront des éléments essentiels de la conception d'un système d'information.

La communication et la diffusion des informations

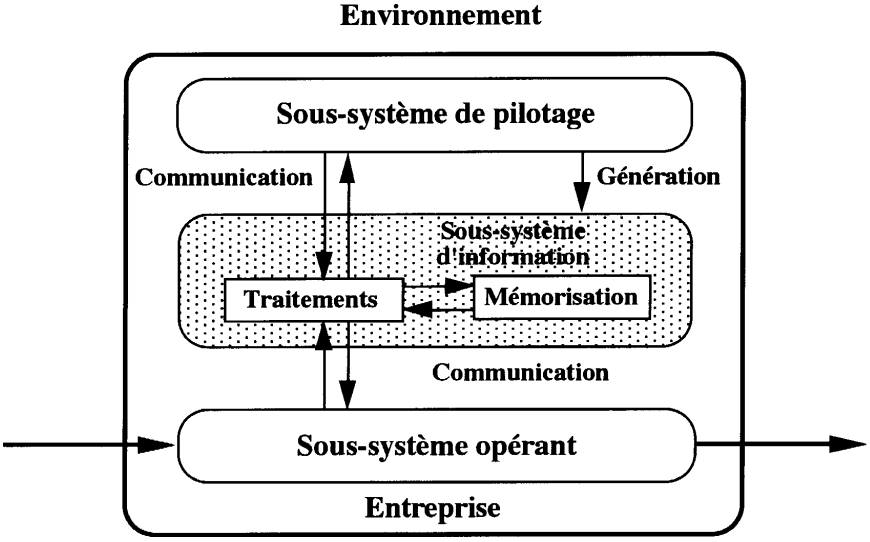
(transfert des informations dans l'espace)

Le système d'information assure les échanges (acquisition et restitution) d'informations avec le système opérant et le système de pilotage. L'organisation de l'acquisition et de la restitution des informations constituera un autre élément important de la conception.

L'exécution des traitements

(transfert des informations dans la forme)

Par référence à l'approche système, les traitements sont soit des activités de transformation d'information-matière première (relevant donc du système opérant), soit des activités de décision, élémentaires ou complexes (relevant du système de pilotage). Le système d'information accueille, pour le compte du système de pilotage ou du système opérant, les traitements suffisamment formalisés et répétitif.



Ainsi défini, le système d'information ne fait aucune hypothèse sur les moyens le supportant ; le système d'information existe indépendamment (et bien avant l'apparition) des techniques informatiques. Toutefois, ces techniques vont permettre d'amplifier les fonctions de

mémorisation, de communication et de traitement des informations. L'informatisation du système d'information comporte ainsi deux préoccupations majeures : d'une part la compréhension et l'explicitation du système d'information (activité, information, organisation), et d'autre part la construction de logiciels (fichiers, programmes...) support du système d'information.

La conception du Système d'information d'une entreprise demande une approche rationnelle. C'est pourquoi on a vu apparaître en France la méthode Merise. Par ailleurs, les développements se trouvent simplifies grâce aux langages dits de quatrième génération (les L4G) avec lesquels, parait-il, « il suffit de savoir ce que l'on veut »...

3.2.3 - Constitution d'un système d'information

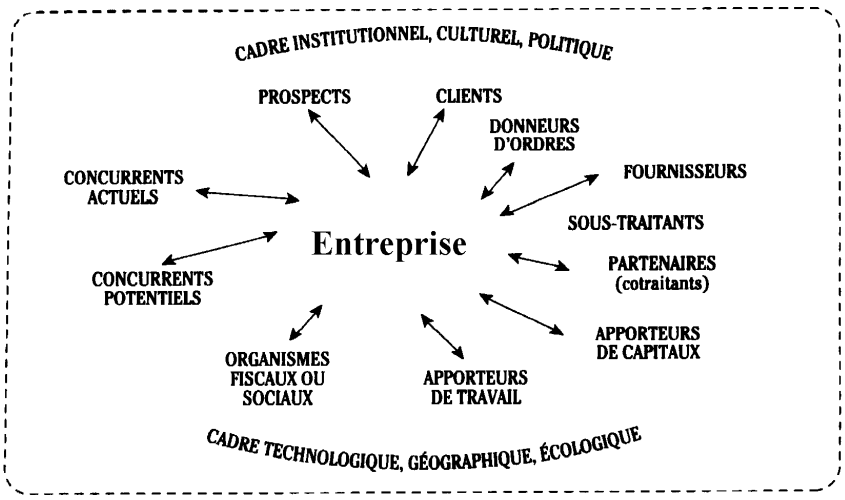
Un système d'information, c'est d'abord un système, c'est à dire un ensemble de composantes qui ont constamment des interactions entre elles, et cela dans le but d'atteindre un objectif précis. C'est ensuite un système basé sur l'information, c'est à dire sur la communication ou l'utilisation de données afin de rendre ces renseignements utiles aux récepteurs éventuels.

Dans tout système d'information, on trouve deux niveaux : le premier niveau qui est le système conceptuel, c'est à dire la structure ou la logique du système, et un second niveau qui constitue le système physique, c'est à dire les différentes composantes informatiques ou tangibles qui servent de support au système conceptuel.

Un système physique est nécessairement construit à l'aide d'un ensemble de composantes. Le système d'information n'échappe pas à cette règle. Ses principales composantes sont l'environnement, les intrants et les extrants...

L'environnement

Lorsqu'on traite de système, on fait souvent la différence entre un système ouvert et un système fermé, selon que le système est influencé ou non par son environnement et qu'il l'influence ou non en retour. En ce sens, un système d'information constitue un des plus beaux exemple de système ouvert : il prend en effet toutes ses informations dans un environnement et les retransmets dans le même environnement, ou parfois dans un environnement différent.



Ainsi, on peut fournir au système d'information différentes données portant sur l'environnement concurrentiel de l'entreprise de sorte qu'après leur traitement les gestionnaires de

l'entreprise puissent disposer d'une information qui leur servira pour la prise de décision. De même, on peut donner au système un ensemble de données comptables qui deviendront des informations utiles à la prise de décision. Ces deux exemples démontrent que le système d'information ne peut vivre sans interagir avec son environnement ; une grande partie des renseignements originaux (les intrants) proviennent de l'extérieur du système d'information et ce qui en sort (les extrants), est lui aussi, la plupart du temps, utilisé à l'extérieur du système.

Les intrants

Les intrants constituent l'ensemble des éléments qui entrent dans le système d'information. Il peut s'agir des renseignements provenant de l'extérieur de l'entreprise comme de ceux provenant de l'intérieur.

Mais attention : tous les renseignements ne sont pas forcément des intrants. Comme nous l'avons déjà signalé, pour qu'un renseignement puisse devenir une information, il doit posséder certaines qualités, soit être utile, exact, complet et ponctuel. Les gestionnaires responsables du système d'information auront donc pour tâche de faire une sélection de ces renseignements. Ils n'accepteront que ceux qui se conforment aux critères établis.

De même, les renseignements devront subir une transformation minimale pour être intégrés au système d'information : ils seront codés. Les données fournies par les organismes spécialisés sont en général de grande qualité, mais la forme sous laquelle ils sont fournis ne correspond pas nécessairement à la structure du système d'information de l'entreprise. Celle-ci devra alors modifier le format de leur présentation, éventuellement les recodifier afin de les intégrer à son système.

Il faut donc accomplir certaines opérations de base pour que des renseignements, qu'ils soient internes ou externes, puissent constituer des intrants dans le système d'information. On doit d'abord sélectionner les renseignements qui remplissent les critères d'acceptation, on doit ensuite les codifier (ou les recodifier) pour qu'ils correspondent au format utilisé par le système. On doit enfin transmettre ces renseignements au système lui-même.

Les extrants

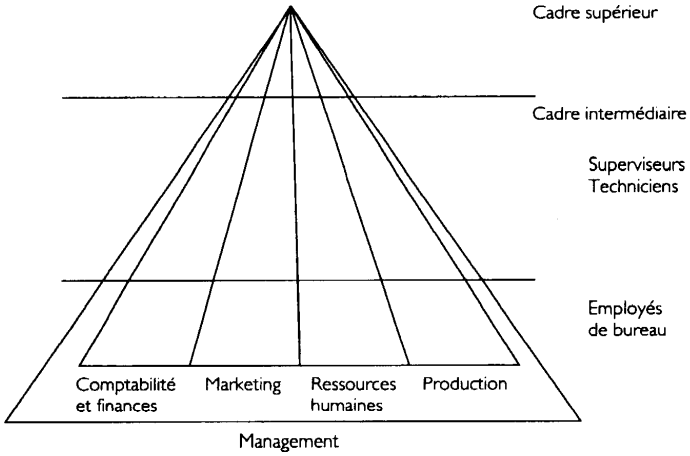
De même que le système doit recevoir des intrants pour pouvoir fonctionner, de même il doit pouvoir produire différents extrants, sans quoi il serait totalement inutile. Ces extrants doivent présenter sensiblement les mêmes qualités que les intrants, soit, être utiles, exacts, complets et ponctuels.

A quoi pourrait servir en effet un extrant qui n'aurait aucune utilité ? Ce ne serait qu'un renseignement, et non une information. Dans la même veine, une information inexacte risquerait bien plus de créer des difficultés que de rendre service à l'entreprise. Par ailleurs, une information incomplète, ou qui n'arriverait pas en temps opportun risquerait de perdre une grande partie de son utilité et de son intérêt.

Les gestionnaires de systèmes d'informations doivent donc s'assurer que les extrants répondent à tous les critères mentionnés précédemment, mais il doivent aussi veiller à ce qu'ils soient transmis à l'aide d'un signal que les récepteurs seront en mesure de décoder sans risque d'erreur. Le matériel informatique actuel offre des possibilités presque infinies de codification de l'information (par le langage écrit ou chiffré, par les graphiques...). La décision du responsable de l'information n'en sera que plus complexe.

3.2.4 - Les sources d'information dans l'entreprise

Dans les situations où l'on trouve un système d'information, on pourra utiliser un autre genre de classification de l'information, ce que l'on pourrait nommer les types d'information? Dans le cas d'un système d'information de gestion, les types d'information seront le plus souvent constitués par les secteurs de gestions suivants : la comptabilité et les finances, le marketing, les ressources humaines, la production et le management



La comptabilité et les finances

Une partie importante de l'information accessible dans l'entreprise provient du secteur de la comptabilité et des finances. Comme les lecteurs le savent probablement déjà, la principale tâche des services comptables consiste à enregistrer les données chiffrées de l'entreprise. Ce service est donc un important fournisseur de renseignements.

Seulement, le rôle de ce service ne se limite pas à l'accumulation de données et de renseignements. Il a aussi la responsabilité de traiter les renseignements qu'il collecte de façon à les présenter sous une forme standardisée, compréhensible et utile aux gestionnaires ou aux individus à l'extérieur de l'entreprise, qui y ont accès. En deux mots, le service comptable doit aussi transformer les renseignements en informations. C'est ce service qui prépare les états financiers de l'entreprise de même que les études de coûts et revenus, de même que l'évolution de la situation financière. Le service financier, s'il est distinct du service comptable, est un utilisateur de l'information. C'est en effet en se basant sur les informations fournies par le service comptable que les gestionnaires des finances prendront toutes sortes de décisions visant à améliorer la situation financière de l'entreprise et à profiter des possibilités qui s'offrent à elle.

En fait, ces services traitent une si grande quantité de renseignements et d'informations qu'on a tendance à croire que les information comptables et financières représentent l'ensemble des informations de l'entreprise. Bien qu'il soit vrai que ces services occupent une place importante dans les systèmes d'information de gestion, ils constituent seulement une composante parmi plusieurs autres.

Les services comptables et financiers préparent ou utilisent de nombreux documents, tels que ceux-ci :

1-/ **Les états de fabrication et des résultats**. Ces états rendent compte des coûts reliés à la fabrication des produits de l'entreprise, des ventes effectuées ainsi que des dépenses engagées.

2-/ **Les budgets**. Ils permettent aux gestionnaires de déterminer la situation de leur trésorerie et de prévoir les besoins futurs ou les surplus de fonds éventuels.

3-/ **Le bilan**. Ce document fournit un portrait de la situation d'une entreprise, à un moment précis.

4-/ **Les rapprochements bancaires**. Ces états confirme au gestionnaire l'état de son avoir en banque (ou de son découvert...)

Le marketing

Le service du marketing est à la fois un fournisseur et un utilisateur d'information. Les gestionnaires ont en effet besoin d'une grande quantité d'information pour être en mesure de prendre les meilleures décisions possibles. Grâces à ces informations, ils pourront, entre autres choses :

1-/ **Effectuer les prévisions des ventes**. Le gestionnaire doit alors disposer d'informations qui lui permettront de déterminer, aussi précisément qu'il pourra, les ventes de l'entreprises au cours des prochaines semaines ou des prochains mois. Mais ces informations ne sont pas seulement utile au marketing. Grâce à elles, il sera possible de prévoir les quantités à produire, les achats à effectuer, le personnel à embauche...

2-/ **Evaluer le marché potentiel**. L'entreprise qui songe à offrir un nouveau produit cherchera à connaître sa clientèle potentiel, tant sur le plan quantitatif que sur le plan qualitatif, la portion de la population qui est susceptible de s'intéresser à ce produit, les régions où se trouvent les acheteurs potentiels...

3-/ **Définir les possibilités d'innovation**. Une entreprise qui désire étendre sa gamme de produits doit déterminer quel produit elle ajoutera, dans quel emballage, à quel prix...

4-/ **Procéder à une évaluation des dépenses**. Le gestionnaire doit prévoir les dépenses qu'il engagera durant les prochaines périodes en publicité, en promotion...

5-/ **Évaluer l'efficacité du travail de communication**. La publicité est-elle réellement efficace ? Qu'en est-il de telle promotion ? Les représentants sont-ils efficaces ? Est-il possible de répartir les budgets autrement, de façon à en retirer un meilleur rendement ?

Les ressources humaines

Ce secteur est souvent négligé par les responsables de l'information. En effet, tout le monde s'accorde pour reconnaître le besoin d'information dans le service du marketing, ou dans celui des finances, mais lorsqu'il s'agit d'évaluer ce besoin dans le service des ressources humaines, les opinions divergent. Pourtant, le besoin d'information est tout aussi important dans ce service que dans les autres. On aura, selon la taille et l'organisation de l'entreprise, des besoins dans les domaines suivants...

1-/ **Les dossiers de candidatures**. Les entreprises d'une certaine envergure, qui engagent régulièrement du personnel, désireront probablement garder en mémoire toutes les candidatures reçues, afin d'être en mesure de retracer rapidement le profil des candidats potentiels à un poste.

2-/ **Les dossiers des employés**. Les gestionnaires se doivent d'avoir à leur disposition un certain nombre d'information sur leurs employés, s'ils veulent gérer au mieux leur service. Un dossier comprend notamment le salaire actuel de l'employé, la date de sa dernière évaluation de rendement, les possibilités de promotion interne...

3-/ **Les données sur les accidents du travail**. Point de plus en plus important dans la gestion des ressources humaines, les accidents du travail font maintenant l'objet d'un contrôle très serré non seulement de la part des gestionnaires de la production, mais aussi de la direction de l'entreprise.

La production

La production est sans doute le second plus important producteur et utilisateur d'informations, après le service comptable et financier. On y traite les informations sur les sujets suivants :

1-/ **Les fournisseurs**. Il s'agit des différentes sources d'approvisionnement pour chacune des matières premières ou des fournitures utilisées dans l'entreprise, des prix offerts par chaque fournisseur, de la courbe des achats...

2-/ **Les stocks**. L'information porte sur l'état des stocks de matière première, de produits en cours de fabrication, de produits finis...

3-/ **Les plans de production**. Selon les prévisions de ventes établies par le service marketing et selon l'état des stocks, il faut planifier la production future de façon a avoir toujours sous la main les quantités suffisantes de produits pour répondre à la demande.

4-/ **Le contrôle de qualité**. Quels sont les standards de qualité de l'entreprise pour chacun des produits, quels sont les niveaux de qualité atteints, quelles corrections doit-on apporter ? Voilà des questions auxquelles le responsable de la production doit être en mesure de répondre.

5-/ **Le transport et les douanes**. Le gestionnaire doit bien connaître les entreprises de transports avec lesquelles il fait affaire, les lois et règlements sur l'importation et l'exportation, les fournisseurs de produits d'emballages...

Le management

Les responsables de tous les services de l'entreprise ont besoin d'informations courante pour être en mesure de prendre des décisions jour après jour. Un bon système d'information fournira alors des informations sur les ventes de la journée précédente, sur l'état actuel des stocks, sur la situation financière, sur l'état des négociations avec les représentants syndicaux...

Un peu de réflexion...

Nous vous proposons ici un exercice sur la conception d'une base de données. Bien évidemment, l'idéal serait, une fois la phase d'analyse terminée, de pouvoir réaliser une maquette de la base. Toutefois, dites-vous bien que ce n'est pas l'essentiel : si la base est mal conçue dès le départ, il est inutile de chercher à la mettre en œuvre.

La phase de réalisation, certes intéressante, est secondaire, dans tous les sens du terme par rapport à la phase d'analyse. Une analyse mal conduite ne pourra jamais aboutir à une application utilisable, même si vous maîtriser toutes les « ficelles » de votre logiciel de gestion de base de données...

Dans cet exercice, votre démarche est guidée pas à pas, afin d'aboutir à un certain résultat. Toutefois, sachez qu'un problème de ce genre n'a jamais de solution unique. Comme souvent, il y a plusieurs solutions possibles, plus ou moins bonnes. Dans le cas présent, celle que l'on vous suggère n'est donc pas la seule possible... Toutefois, le rédacteur de ces lignes estime que c'est la meilleure (ou la moins mauvaise...)

D'abord le problème, brutal dans sa brièveté...

On vous demande de dessiner la structure d'une base de données permettant d'organiser le logement d'étudiants dans différentes Résidences Universitaires. On suppose, à priori, qu'il s'agit de chambres individuelles.

Ensuite des remarques...

Dessiner la structure de la base (certains dirait réaliser un modèle de cette base) revient à déterminer quelles sont les champs à utiliser (les données à prendre en compte), les tables à mettre en place, et les liens (les relations) à établir entre ces différentes tables.

Or, il est évident que, dans « l'univers » qui nous concerne, il existe trois catégories d'**objets** : les étudiants, les chambres et les Résidences Universitaires. Il est tout aussi évident qu'il existe des **liens** entre ces objets, comme par exemple, en vrac et dans le désordre...

un étudiant occupe une chambre durant un espace de temps donné,
une chambre peut être utilisée successivement par plusieurs étudiants,
une chambre est située dans une Résidence Universitaire,
une Résidence Universitaire comporte un certain nombre de chambre.

Avec cela, vous devriez pouvoir vous débrouiller, essayer de répondre à ce problème avant de tourner la page...

La **première étape** vers la solution consiste donc à étudier chacun des **objets** en présence(étudiants, chambres, résidences), afin de déterminer les champs nécessaires à leur description. En d'autres termes, on va chercher la structure des trois **tables** qui correspondent à ces trois objets. Dans cette phase, on ne s'intéresse pas aux liens qui vont les relier ensemble, mais uniquement aux tables...

Dans un **second temps**, on va étudier ces liens. On sait par exemple qu'un étudiant occupe une chambre durant un espace de temps donné, et qu'une chambre peut être utilisée successivement par plusieurs étudiants. Cela devrait conduire à « matérialiser » ce lien par une nouvelle table (qui pourrait s'appeler **occupation**). Cette table devra pouvoir « pointer » vers la table **étudiant** (par la « clé primaire de cette table) en même temps que vers la table **chambre** (également par la clé primaire de cette autre table). En outre, d'autres indications sont nécessaires à ce lien (qui matérialise le logement d'un étudiant), comme par exemple la **date d'arrivée** de l'étudiant, ainsi que sa **date de départ**. Ceci afin de savoir qui occupe, ou a occupé cette chambre. Cela permettra en même temps de savoir si la chambre est disponible. Si une chambre est inconnue dans la table **occupation,** cela signifie qu'elle n'a pas encore été attribuée. De même si le dernier occupant est parti, la chambre peut être attribuée à un autre étudiant...

Le second lien, celui entre la résidence et la chambre est du même type, mais, contrairement au précédent, il ne risque pas d'être modifié ! Son seul rôle est de ne pas écrire une centaine de fois l'adresse de la résidence, si cette dernière propose une centaine de chambre... On voit rapidement que cette table supplémentaire pourra ne comporter que deux champs, la référence de la résidence, et la référence de la chambre...

En fonction de ce qui précède, la base de données, sous sa forme finale, pourrait donc comporter au maximum 5 tables... Toutefois, puisque ce lien est définitif, il serait plus simple de placer directement dans la table chambre la référence de la résidence dans laquelle elle se trouve, ce qui ne ferait que quatre tables, et donc une structure un peu plus simple.

Toutefois, on peut adopter une autre solution si on décide de placer les caractéristiques de la résidence parmi les caractéristiques de la chambre. Dans ce cas, on n'utilisera que les tables **étudiant**, **logement** et **chambre**... Le défaut de ce choix est que, dans ce cas, il faudra recopier les caractéristiques d'une résidence (par exemple son adresse) autant de fois qu'elle propose de chambre. Ce que l'on a l'impression de gagner en simplicité au niveau de la structure est perdu par le nombre de données supplémentaires à écrire, et par l'inévitable augmentation du volume de la base de données.

On complique le problème...

Certaines résidences universitaires disposent de place de parking (numérotées et identifiables individuellement). On envisage d'attribuer ces places, dans la mesure du nombre disponible, à des étudiants possédant une voiture. Compléter votre modèle.

Encore des remarques...

La première partie a pu vous conduire à prévoir quatre tables, il faut à présent ajouter un objet supplémentaire (**places de parking**) dans votre base, et à trouver les liens entre ce nouvel objet et ceux que vous avez déjà mis en place...

Une solution intéressante sera peut-être de prendre en compte un autre objet, cité dans l'énoncé du problème : l'objet voiture... Seulement, n'oubliez pas qu'une voiture n'a qu'un seul propriétaire, et qu'on considère ici qu'un étudiant ne possède qu'une seule voiture !

Vers une solution...

Attention, sachez qu'il n'existe pas de **Solution** unique (et magique) au problème posé, mais des solutions, plus ou moins adaptées aux besoins exprimés. On a bien dit plus haut que l'on compliquait le problème. Selon l'angle sous lequel vous allez l'aborder, vous pouvez aboutir à des solutions différentes....

Quel que soit le problème posé, il n'existe pas de Solution unique (et magique) au problème posé, mais des solutions, plus ou moins adaptées aux besoins exprimés.

Selon l'angle sous lequel vous allez aborder le problème posé, vous pouvez aboutir à des solutions différentes...

Table des Matières

PREMIÈRE PARTIE : DE LA THÉORIE... 3

CHAPITRE 1 : UN PEU D'HISTOIRE 5

1.1.0 -Très brève introduction 6
1.1.1 - L'ancêtre, sur micro-ordinateur... 6
1.1.2 - Paradox entre en scène 7
1.1.3 - La chute d'Ashton-Tate 7
1.1.4 - Borland acquiert Ashton-Tate, Microsoft rachète Fox 9
1.1.5 - Le standard dBase 9
1.1.6 - Le boum des SGBD graphiques 10
1.1.7 - En guise de conclusion... 10

CHAPITRE 2 : LES BASES DE DONNÉES... 13

1.2.1 - Les bases de données 14
1.2.2 - Qu'est-ce que le modèle relationnel ? 14
1.2.2.1 - Origine et principe 14
1.2.2.2 - Les relations 15
1.2.2.2 - Tables et Bases 16
1.2.2.4 - Pensez relationnel 16
1.2.3 - Les 12 règles de Codd 17

CHAPITRE 3 : CONCEPTION D'UNE BASE 21

1.3.1 - Intérêt d'une bonne étude 22
1.3.2 - Création d'une base de données 22
1.3.2.1 - Etape 1 : Définition de l'objectif 22
1.3.2.2 - Etape 2 : Définition des tables 23
1.3.2.3 - Etape 3 : Définition des champs 23
1.3.2.4 - Étape 4 : Définition des Relations 24
1.3.2.5 - Etape 5 : Mise au point 25
1.3.2.5 - Première utilisation 26
1.3.3 - Quelques règles simples 26
1.3.4 - Exemple 27
1.3.4.1 - Définition de l'objectif 27
1.3.4.2 - Définition des tables 27
1.3.4.3 - Définition des champs 27
1.3.4.4 - Définition des relations 28

SECONDE PARTIE : ...À LA PRATIQUE 31

CHAPITRE 1 : LE LOGICIEL MICROSOFT ACCESS 33

2.1.1 - Le logiciel Microsoft Access 34
2.1.2 - Un peu de vocabulaire... 35

CHAPITRE 2 : EXEMPLE PRATIQUE 37

2.2.1 - Comment créer une base (vide) 38

2.2.2 - Comment créer une table (vide) *39*
2.2.3 - Comment créer un formulaire *42*
2.2.4 - Comment créer une requête *42*
2.2.5 - Comment créer un état *45*
2.2.6 - Maintenance d'une base *46*

CHAPITRE 3 : D'AUTRES CONSEILS 47

2.3.1 - La base de données *48*
2.3.2. - Définir des relations entre les tables *48*
2.3.3 - Qu'est-ce que l'intégrité référentielle? *50*
2.3.4 - Travailler sur plusieurs tables *51*

COMPLÉMENT : INFORMATION ET ENTREPRISE 53

CHAPITRE 1 : INFORMATION 55

3.1.1 - Les données et l'information *56*
3.1.2 - Les qualités de l'information *57*
L'information doit être utile 57
L'information doit être exacte 57
L'information doit être complète 57
L'information doit être ponctuelle 57
3.1.3 - Les sources d'information *58*
Les sources primaires 59
Les sources secondaires 59
Les sources internes 60
Les sources externes 60
3.1.4 - L'information dans le temps *60*
La régularité 60
La période 60
3.1.5 - La présentation de l'information *61*

CHAPITRE 2 : ...ET ENTREPRISE 63

3.2.1 - L'entreprise en tant que système *64*
Le système opérant transforme et produit 64
Le Système de pilotage réfléchit, décide et contrôle 64
Le système d'information mémorise, traite et diffuse 65
3.2.2 - Les fonctions du système d'information *65*
La génération des informations 65
La mémorisation des informations 66
La communication et la diffusion des informations 66
L'exécution des traitements 66
3.2.3 - Constitution d'un système d'information *67*
L'environnement 67
Les intrants 68
Les extrants 68
3.2.4 - Les sources d'information dans l'entreprise *69*
La comptabilité et les finances 69
Le marketing 70
Les ressources humaines 70
La production 71
Le management 71

UN PEU DE RÉFLEXION 73