

Fichiers et bases de données

Édition au 26-déc.-15 ATTENTION CETTE VERSION EST EN TRAVAUX !!!!!!!!!!!!!!!!!!!!!!!

1. Organisations des informations en fichiers

Pour vous familiariser avec la gestion de fichiers simples, vous pouvez réaliser votre carnet d'adresses à l'aide d'un logiciel comme Works ou Claris Works, intégrés incluant un gestionnaire de fichiers.

La structure d'information la plus simple est constituée d'un seul fichier qui contient une succession de fiches de même taille.

La fiche

L'élément fondamental d'un fichier de données est constitué par la "fiche" que l'on préfère nommer **enregistrement** (ou encore article).

Cet enregistrement est lui-même composé de plusieurs rubriques (ou zones) encore appelées les **champs**.

Si l'on prend comme exemple un fichier devant contenir des informations sur les livres d'une bibliothèque, il est facile d'imaginer le modèle de fiche (volontairement simplifié dans l'exemple choisi) que l'on devrait donner à l'imprimeur :

numéro d'inventaire	numéro d'inventaire
auteur	auteur
titre	titre
thèmes	thèmes
éditeur	éditeur
date d'achat	date d'achat
statut	<input type="radio"/> empruntable <input type="radio"/> non empruntable

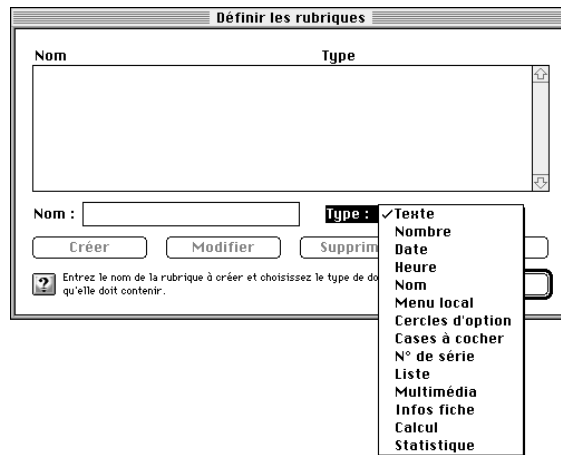
Voici un exemple de fiche complétée avec la saisie en cours dans une liste de l'éditeur :

numéro d'inventaire	21544
auteur	RAMBAUD Patrick
titre	La Bataille
thèmes	guerre, Napoléon
éditeur	Hachette
date d'achat	Flammarion
statut	<input type="radio"/> non empruntable

création de la fiche, type des champs

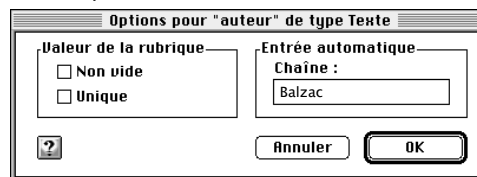
À l'instar de ce modèle, la première tâche lors de la création sera de définir la structure d'une fiche, c'est à dire de choisir les caractéristiques des champs qui la compose et éventuellement de sa présentation. Il faudra préciser, entre autres :

- l'**identificateur** (ou le nom) du champ
- son **type** (c'est à dire quelle est la nature des informations à y placer).

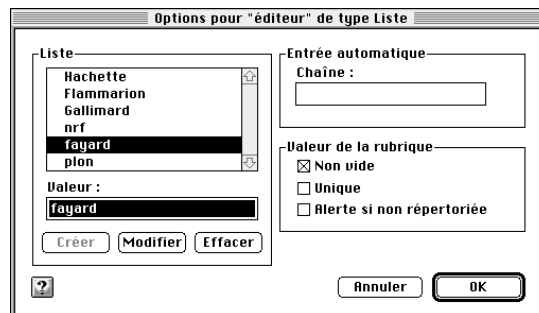


La forme des informations à saisir vont varier selon le logiciel utilisé; en particulier les types proposés sont plus ou moins nombreux. Le choix du type du champ dépend non seulement des informations qu'il contiendra, mais aussi des opérations que l'on effectuera sur ce champ.

Par exemple il est préférable de choisir un type " caractère ou texte " pour un code postal, un numéro de téléphone, ... bien que ceux-ci soient constitués de chiffres : en effet, ces chiffres ne constituent pas à proprement parler des nombres et on n'effectuera pas de calculs arithmétiques sur un code postal comme on pourrait être amené à le faire sur un salaire par exemple; en revanche on pourra extraire les deux premiers caractères (département) d'un code postal. De plus, il est possible de laisser apparaître une proposition par défaut qui sera choisie sauf si l'utilisateur frappe autre chose.



Un type énuméré permettra de choisir dans une liste la réponse possible et évitera les erreurs de saisie.



On remarquera de plus que l'on peut obliger l'utilisateur à mettre une donnée (cochage de " non vide ").

Les types calculés correspondent à la possibilité pour le logiciel d'évaluer la valeur d'un champ à partir d'autres. Une valeur totale T.T.C. sera calculée à partir du prix unitaire hors taxe, de la quantité en stock et du taux de TVA.

À partir d'une date on pourra calculer simplement un nombre de jours. L'application vérifiera automatiquement la conformité d'une date si le champ est de type date.

On le voit donc, à un type de champ est associée tout une série d'opérateurs et de fonctions.

Pour les cercles d'options, il est indispensable lors de leur création de préciser l'élément " par défaut ", c'est à dire celui qui apparaîtra si tel est le désir du créateur du fichier, et de compléter des autres éléments. Rappelons qu'un seul cercle d'option pourra être choisi.

Enfin certaines applications intègrent le type image ou son. Il est ainsi possible de réaliser des fiches avec la photographie des individus, ou un catalogue avec les images des produits vendus.

la taille des champs

La taille de certains champs est automatiquement imposée par les logiciels. Par exemple un champ de type logique (qui ne peut prendre que deux valeurs -vrai/faux, ou oui/non-) occupera dans la fiche un octet (caractère).

Certains logiciels demandent à l'utilisateur de choisir la taille maximale des données du champ : il n'est pas judicieux de choisir une taille trop élevée pour éviter d'occuper inutilement de la place mémoire. Les logiciels modernes, au contraire, ont la capacité de gérer des fiches de longueur variable, et seuls les caractères saisis sont stockés sur le disque évitant le gaspillage de la place.

commandes accessibles

Un gestionnaire de fichier doit être capable au minimum des possibilités suivantes :

- Définir la structure d'un fichier et éventuellement la modifier

- Ajouter des fiches (dans la limite de l'espace disponible sur la mémoire de masse).
- Modifier une fiche
- Afficher, sur écran ou sur imprimante, le contenu des fiches.

En général de nombreuses autres possibilités sont présentes :

- Contrôle à la saisie en cas d'erreur de frappe (un caractère à la place d'un chiffre dans le prix d'achat par exemple)
- Formatage à la saisie ou à l'édition (mise en majuscule, cadrage ...)
- Tris selon une ou plusieurs clés (par exemple Nom + Prénom)
- Extraction sélective de fiches et de champs (par exemple liste des auteurs et des titres dont l'éditeur est Hachette)
- Actions globales portant sur un ensemble de fiches pour faire des totaux, des moyennes, des sommes, ...
- Possibilité de redéfinir la structure des fiches
- Outils de présentation pour les différentes éditions de fiches
- Présence d'un langage évolué pour l'interrogation du fichier
- Édition de rapports rassemblant les données sélectionnées et des opérations globales sur les fiches sélectionnées.

classement dans un fichier : tris et indexation

Parmi les opérations habituelles disponibles pour la manipulation des fichiers on dispose en général de la possibilité de trier les fiches sur une ou plusieurs clés.

Cela est nécessaire si l'on veut retrouver très vite une information. (imaginons ce qui se passerait si le dictionnaire n'était trié par ordre alphabétique).

On obtient ainsi un fichier trié, par exemple :

Fichier non trié

N°	Nom	Prénom
1	VERLAINE	Paul
2	LACAN	Jacques
3	POE	Edgar
4	ARAGON	Louis



Fichier trié

N°	Nom	Prénom
4	ARAGON	Louis
2	LACAN	Jacques
3	POE	Edgar
1	VERLAINE	Paul

Au cours de l'opération de tri il y a une réorganisation physique du fichier sur le disque, cette opération peut prendre beaucoup de temps avec de gros fichiers.

Si l'on ajoute la fiche de JARRY Alfred dans le fichier trié, celle-ci se place en fin de fichier on perd alors le bénéfice du tri précédent.

N°	Nom	Prénom
4	ARAGON	Louis
2	LACAN	Jacques
3	POE	Edgar
1	VERLAINE	Paul
5	JARRY	Alfred

On voit donc que le tri n'est pas très intéressant dans le cas présent. La plupart des logiciels de gestion de fichier propose la technique de l'indexation.

L'indexation d'un fichier consiste à gérer (le plus souvent de façon totalement transparente pour l'utilisateur) une table ordonnée constituée de la clé et d'une indication permettant d'accéder à la fiche correspondante (adresse relative de la fiche dans le fichier principal par exemple). La manipulation de cette table d'index est très rapide, l'ajout d'une fiche revient à ajouter la fiche en fin de fichier et à mettre à jour la table d'index en insérant, dans la table triée la nouvelle clé. Le système accède aux fiches *via* la table d'index. On évite ainsi les tris fastidieux sur le fichier. En outre rien n'empêche d'utiliser plusieurs tables d'index (un index sur les éditeurs et un index sur les auteurs par exemple) permettant d'accéder aux fiches de deux façons différentes très souplement et très rapidement.

Fichier non trié

N°	Nom	Prénom
4	ARAGON	Louis
2	LACAN	Jacques
3	POE	Edgar
1	VERLAINE	Paul
5	JARRY	Alfred

Table d'index

Clé	Prénom	Adresse de la fiche
ARAGON	Louis	4
JARRY	Alfred	5
LACAN	Jacques	2
POE	Edgar	3
VERLAINE	Paul	1

Un peu de terminologie

Le terme base de données bien que se rapportant à une notion bien précise pour les spécialistes est souvent confondu avec d'autres appellations comme banques de données, gestionnaires de fichiers, ...

Banques de données : Il s'agit d'un ensemble de données sur un sujet particulier (par exemple la banque de données Prolabo[®] concernant les fiches sécurités des produits chimiques distribués par cette firme, la banque de données Merck[®] contenant plus de 6000 références bibliographiques sur les techniques de HPLC ou encore la banque d'images médicales du CRIA).

Il s'agit donc en fait d'une bibliothèque d'informations accessibles au public. En général cet accès est payant puisqu'il faut créer la banque, tenir à jour les informations gérées ...

Ces informations peuvent d'ailleurs ne pas être "informatisées" et se présenter sous forme de document papier, film, microfilm,...

Fichiers et gestionnaires de fichiers : Un fichier est une collection de données (data en anglais) groupées sous la forme d'une suite de caractères (octets) et en général stockées sur un support magnétique (disquette, bande magnétique...).

Un gestionnaire de fichier est constitué d'un ensemble d'outils logiciels permettant à un utilisateur non spécialiste de gérer un fichier (créer des fiches, mettre à jour, supprimer, imprimer, ...)

Bases de données et Système de Gestion de Base de Données (S.G.B.D.) : Concept apparu au début des années soixante à la suite de travaux menés par I.B.M.

Une base de donnée est un ensemble d'informations, organisées en fichiers, accessibles par différents utilisateurs. Pour assurer un fonctionnement rationnel de la base, ces informations doivent être structurées et non redondantes, c'est à dire ne figurant pas plusieurs fois, sans que cela soit nécessaire, dans les fichiers composant la base de données.

Le S.G.B.D. est formé de l'ensemble des outils logiciels permettant de définir et de gérer une ou plusieurs bases de données.

Une base de données suppose donc obligatoirement une implémentation informatique.

2. Base de données

2.1 Les limites d'une gestion monofichier

Imaginons une gestion informatique traditionnelle dans une bibliothèque d'établissement scolaire organisée en fichiers indépendants.

La bibliothèque gère les emprunts avec le fichier BIBLIO précédent, il lui faut en outre pouvoir adresser un rappel à l'adresse de l'emprunteur, elle possède donc en plus un fichier des élèves avec leur adresse.

Le bureau des absences possède un fichier des élèves avec le relevé des absences et bien sûr leurs adresses

Le service de la scolarité possède aussi un fichier des élèves avec leur adresse et les résultats scolaires.

On le voit, les adresses, utilisées par tous sont présentes dans trois fichiers. Il y a une redondance des informations, redondance coûteuse en place occupée sur les disques.

De plus, que va-t-il se passer si un élève signale au service de la scolarité qu'il a changé d'adresse ? Qu'en sera t-il s'il informe le bureau des absences ?

Comment s'assurer que la mise à jour sera effectuée dans l'ensemble des fichiers comportant cette information ? Certaines informations des fichiers risquent rapidement devenir incohérentes.

Il n'est pas possible, non plus avec ce système d'établir des liens entre les informations détenues, par exemple ne serait-il pas intéressant de mettre en corrélation les résultats scolaires et la fréquentation de la bibliothèque ?

La structure monofichier n'apporte pas de réponse aux problèmes posés.

C'est à la suite de ces inconvénients que le concept de base de données est apparu.

2.2 Les différents modèles de bases de données

Les S.G.B.D. sont des logiciels capables de gérer simultanément plusieurs fichiers et les différents "liens" qui peuvent exister entre eux, ils doivent en outre disposer de puissantes possibilités d'édition et permettre des requêtes complexes, éventuellement à partir de plusieurs postes de travail.

Au moins trois modules sont nécessaires :

- un module de définition de la structure de la base (on utilise le terme de schéma de la base) : il s'agit de définir les différentes *entités* de la base (exemple l'entité client, fournisseur, facture, ...) et les liens entre ces entités. Cette phase de définition est capitale dans l'efficacité des applications qui seront développées autour de cette base.
- un module de manipulation des données (ajouts, suppressions, ...)
- un module de commande et d'interrogation de la base qui utilise un langage de requêtes spécialisé.

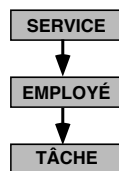
Les S.G.B.D. que l'on rencontre reposent sur trois modèles théoriques qui se différencient par les modalités de stockage et d'exploitation des données :

- Le modèle hiérarchique, le plus ancien.
- Le modèle en réseau, amélioration du modèle précédent.
- Le modèle relationnel, dernier né, basé sur une structure mathématique et qui est actuellement le plus développé.

Le modèle hiérarchique

Il s'agit d'une base de donnée dans laquelle toutes les informations sont organisées sous la forme d'une arborescence. Il convient bien pour des situations dans lesquelles on a un enregistrement "parent" dont dépend un nombre variable d'enregistrements sous-ensemble.

On peut illustrer cette arborescence par le schéma ci contre :



Cette structure est très efficace quand on accède aux informations par le sommet de l'arborescence, en revanche l'exploitation est beaucoup plus délicate, voire même impossible dans les cas où l'on recherche les informations par le "bas". dans l'exemple il est facile d'obtenir toutes les tâches effectuées par un employé d'un service donné, en revanche il est plus délicat de chercher le service où l'on effectue une tâche particulière.

Le modèle en réseau

Apparu pour remédier aux limites du modèle hiérarchique il reprend l'architecture hiérarchique en ajoutant des liens supplémentaires entre les informations. On obtient ainsi une structure de graphe plus générale et plus efficace.

Ce modèle est souple, mais il dépend étroitement de la structure de stockage des informations.

Le modèle relationnel

La puissance des microordinateurs ayant sensiblement augmenté, ce modèle a pu être implémenté de façon satisfaisante sur ces machines.

L'exploitation de la base est construite sur les relations que l'on crée entre les éléments de différents fichiers, une information commune à deux fichiers permet d'en exploiter simultanément les données.

En reprenant l'exemple de la gestion de bibliothèque précédent on peut constituer les fichiers suivants :

On regroupe toutes les informations dans un même système composé de plusieurs fichiers, chaque service n'ayant accès qu'aux informations dont il a besoin.

Par exemple on peut imaginer 4 fichiers (on suppose dans cet exemple que le nom suffit à discriminer tous les étudiants) :

Fichier Biblio	Fichier Élèves	Fichier Absences	Fichier Notes
INVENTAIRE	NOM	NOM	NOM
AUTEUR	ADRESSE	DATE ABSENCE	MOY. PARTIEL1
TITRE	CODE POSTAL	MOTIF	MOY. PARTIELÉ
ÉDITEUR	VILLE		MOY. PARTIEL3

DATE ACHAT
THÈMES
PRIX
EMPRUNT
EMPRUNTEUR

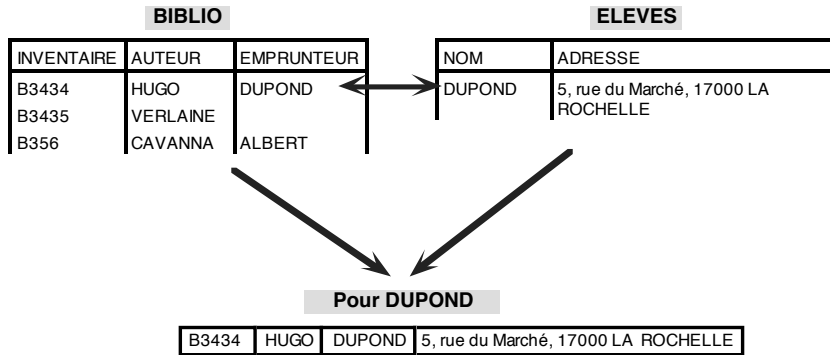
CLASSE

MOY. Examen
REÇU
MENTION

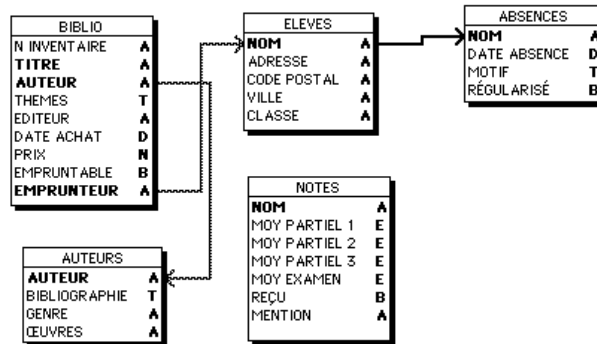
Seul le nom figure dans plusieurs fichiers, c'est lui qui va permettre de relier les fichiers entre eux.

On peut par exemple établir une relation entre le champ Emprunteur de BIBLIO et le champ nom de ELEVES.

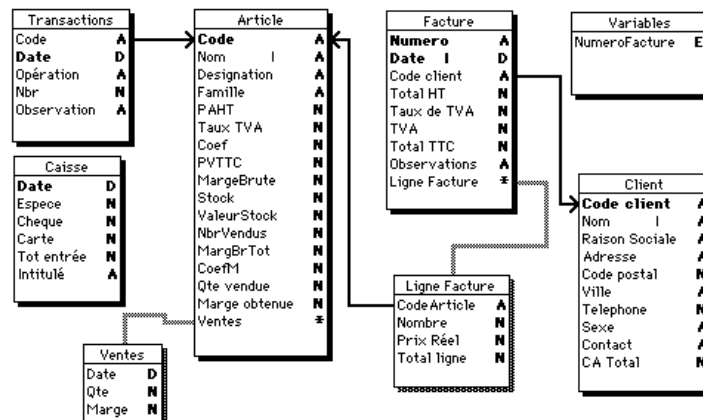
Le schéma ci-après illustre le lien établi entre les fichiers BIBLIO et ELEVES, une information commune aux deux (DUPOND) permet de relier deux enregistrements afin d'obtenir l'adresse de DUPOND qui a emprunté le livre de Hugo.



Dans certains environnements ces relations peuvent être visualisées sous forme de traits, comme dans cet exemple en 4° DIMENSION® :



Le schéma d'une base de donnée peut-être très complexe comme cette comptabilité en 4° Dimension® :



2.3 Des langages d'interrogation

Un certain nombre de règles élaborées par Codd permet de construire des structures relationnelles qui respectent au mieux l'intégrité des données et la non redondance des informations.

Pour gérer la base (et en particulier formuler des requêtes) il faut disposer d'un langage. Ce langage peut-être propre à l'application et permettre l'interrogation, voire même la programmation de modules spécifiques à l'application, il en est ainsi du langage de dBASE® ou encore du langage PAL de PARADOXE®.

En DBASE III plus , pour obtenir la liste des emprunteurs de livres avec leur adresse il faut effectuer les commandes suivantes successivement :

```

SELECT 2
USES ELEVES INDEX noms && on suppose avoir créé cet index
SELECT 1
USE BIBLIO
SET RELATION TO Nom INTO ELEVES
DISPLAY ALL titre,auteur, emprunteur, eleves->adresse,eleves->codepostal, eleves->ville
FOR      emprunteur # ' '
    
```

Ces langages supposent la connaissance de la structure de la base et sont insuffisants pour pouvoir communiquer avec l'ensemble des serveurs de base de données.

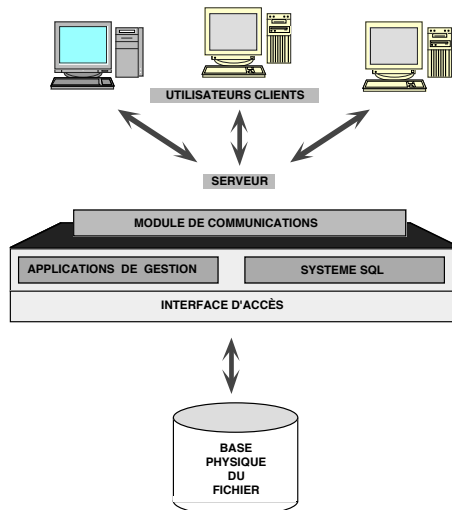
On utilise d'autres formes pour adresser des requêtes qui ne présupposent pas connaître la méthode (procédure) que doit utiliser le programme de gestion pour obtenir le résultat.

On utilise plus particulièrement :

- QBE (Query By Example) disponible avec PARADOX® ou DBASE IV® qui, comme la traduction l'indique, consiste à formuler une requête en indiquant au logiciel ce que l'on souhaite obtenir plutôt que la procédure à suivre pour l'obtenir.
- Le langage SQL (Structured Query Language = langage d'interrogation structuré) qui permet d'accéder aux bases de données qu'elles soient sur micro-ordinateur ou sur gros système. Il s'agit en fait de tout un système disponible depuis peu sur micro-ordinateurs qui sera introduit directement dans les systèmes d'exploitation (SQL Server de Microsoft ou SQL CL/1 pour Apple). Il s'agit d'un langage non procédural¹ que des applications "frontales"² utilisent pour communiquer avec le serveur, le tout étant de plus en plus transparent pour l'utilisateur.

¹. non-procédural : qui agit par description de l'objet recherché; par opposition à procédural qui décrit les moyens employés (procédure) pour atteindre l'objectif.

² Application que l'utilisateur a devant lui par exemple DBASE VI par opposition aux applications centrales qui sont implémentées sur le serveur SQL Server par exemple.



Corrélat :