

1. Les systèmes de Gestion de Bases de Données	2
1.1 LA BASE DE DONNÉES	2
1.2 LE SYSTÈME DE GESTION DE BASES DE DONNÉES	2
2. L'organisation des données dans la base de données	3
2.1 PRÉSENTATION DE L'EXEMPLE	3
2.2 L'ORGANISATION DES DONNÉES DANS UN SGBD	3
2.3 DIVISER POUR MIEUX RÉGNER	4
2.4 UTILISER PLUSIEURS TABLES	5
2.5 IL EXISTE TROIS TYPES DE RELATIONS ENTRE LES TABLES :	7
2.6 LES INDEX	12
3. Application	13
CREATION D'UNE BASE DE DONNEES AVEC ACCESS	17
3.1 CRÉATION D'UNE TABLE	19
3.2 LES TYPES DE DONNÉES	20
3.3 LES PROPRIÉTÉS DES CHAMPS	22
3.4 LE FORMAT D'AFFICHAGE	24
Pour les champs texte et mémo	24
Pour les champs Numériques et Monétaires	24
Pour les champs de type Date / Heure	26
Format d'affichage des dates / heures dans Windows	28
Pour les champs de type OUI/NON	29
3.5 LE MASQUE DE SAISIE	29
3.6 LES LISTES DE CHOIX	30
3.7 TRI PARMIS LES ENREGISTREMENTS	42
3.8 FILTRER DES ENREGISTREMENTS	43
Le filtre sur un seul critère :	43
Filtre sur plusieurs critères	44
3.9 RECHERCHER DES ENREGISTREMENTS	45

Cours Microsoft ACCESS

1. Les systèmes de Gestion de Bases de Données

1.1 La base de données

Une base de données est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun. Elle doit avoir un certain nombre de caractéristiques :

- ?? **Données structurées** : les informations contenues dans une base de données sont réparties en enregistrements , chaque enregistrement ayant une structure bien définie
- ?? **Données non redondantes** : Une même information ne sera pas répétée plusieurs fois dans la base de données.
- ?? **Données cohérentes** : Il ne doit pas être permis d'enregistrer dans une base des informations incohérentes entre elles
- ?? **Données accessibles** directement selon de multiples critères
- ?? **Indépendance des programmes et des données** : La base de données doit être indépendante des programmes qui y ont accès, on doit pouvoir utiliser un autre programme pour traiter différemment ces données sans avoir à toucher à ces données
- ?? **Sécurité des données stockées** : la base de données doit permettre un système de sécurité permettant de gérer les droits d'accès aux informations par les utilisateurs.

1.2 Le Système de Gestion de Bases de Données

Un Système de Gestion de Bases de Données (S.G.B.D.) représente un ensemble coordonné de logiciels qui permet de décrire, manipuler, traiter les ensembles de données formant la base. Il doit également assurer la sécurité et la confidentialité des données dans un environnement où de nombreux utilisateurs ayant des besoins variés peuvent interagir simultanément sur ces données.

Il doit pouvoir être utilisé par des non-informaticiens. Il doit assurer la définition des structures de stockage et des structures de données et le suivi de leur évolutions ; c'est ce qu'on appelle l'administration des données. Il doit pouvoir au maximum vérifier la cohérence des données. Le SGBD sert donc d'interface entre les programmes d'application des utilisateurs d'une part, et la base de données d'autre part.

Microsoft Access est un SGBD dont nous allons étudier le fonctionnement.

2. L'organisation des données dans la base de données

2.1 Présentation de l'exemple

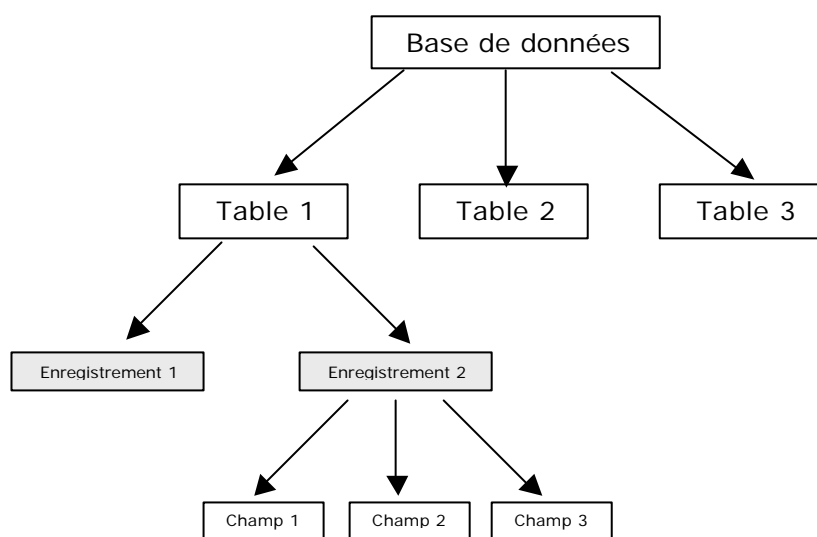
Avant de pouvoir exploiter les données contenues dans la base, il va falloir les modéliser, par modéliser, je veux dire trouver le meilleur moyen de représenter le monde réel en structurant la base de données de manière à pouvoir l'exploiter la plus simplement par la suite.

2.2 L'organisation des données dans un SGBD

Dans Access, comme dans les autres SGBD, les informations doivent être segmentées en données qui sont stockées dans des **tables**. Une **table** est donc un ensemble de données, organisées en lignes et en colonnes. On peut stocker dans une table n'importe quel type d'information (texte, chiffres, graphisme, son, etc...)

Chaque table est divisée en enregistrements, les enregistrements étant l'ensemble des données relatives à la même information. Prenons l'exemple d'un carnet d'adresses. Pour ce carnet d'adresses, nous allons utiliser une table «contacts » qui contiendra toutes les informations concernant chaque contact, chaque enregistrement contiendra les informations relatives à un (et un seul) contact (nom, adresse, téléphone, date de naissance, etc....).

Comme on vient de le voir, chaque enregistrement contient un ensemble d'informations (nom, adresse, téléphone) qui ont elles-même divisé l'enregistrement en plusieurs parties, chaque partie s'appelle un **champ**. Le champ contient une partie des informations de chaque enregistrement.



Organisation de la base de données

Si on reprend notre exemple de carnet d'adresse, nous allons avoir une table qui peut ressembler à ça :

Champ
↓

Nom	Adresse	Téléphone
Deuf John	Rue Barbe – 75001 Paris	01.02.03.04.05
Covert Harry	Rue Minant – 59000 Lille	02.03.04.05.06
Dalors Homère	Rue Tilant – 75003 Paris	03.04.05.06.07

← Enregistrement

Quelques définitions :

?? **La base de données** : C'est l'ensemble des tables utilisées pour gérer l'information

?? **La table** : c'est un ensemble de données organisées en rangées et en colonnes

?? **Le champ (ou colonne)** : C'est l'unité d'information dans une table. Une table est constituée de différents champs

?? **L'enregistrement (ou la ligne)** : C'est l'ensemble des données relatif à la même information (le 1^{er} enregistrement de la table contient les informations relatives au contact « John Deuf »)

2.3 Diviser pour mieux régner

Il faut toujours segmenter l'information en éléments les plus petits possibles. On aurait très bien pu créer une table contact avec un seul champ par enregistrement qui contiendrait toutes les informations en même temps. Mais une des règles d'or dans la création de la structure d'une base de données est de diviser le plus possible l'information pour pouvoir la traiter par la suite le plus simplement et le plus efficacement possible.

Le SGBD, pour retrouver les données qu'on lui demande, travaille avec le contenu des champs, il lui sera donc difficile, dans la table « Contacts » telle qu'elle est de retrouver les contacts habitant à Lille, la ville se trouvant mélangée au reste de l'adresse, de même, on ne pourra pas récupérer le nom seul de chaque contact, celui-ci se trouvant dans le même champ que le prénom.

Comme on ne peut pas toujours prévoir, lors de la création de la structure de la table, de l'usage qui en sera fait plus tard, il faut diviser le plus possible les données, sachant qu'il sera toujours compliqué (et risqué) de modifier la structure de la table par la suite lorsqu'elle sera remplie.

Une meilleure organisation de la table serait la suivante :

Nom	Prénom	Rue	Code Postal	Ville	Téléphone
Deuf	John	Rue Barbe	75001	Paris	01.02.03.04.05
Covert	Harry	Rue Minant	59000	Lille	02.03.04.05.06
Dalors	Homère	Rue Tilant	75003	Paris	03.04.05.06.07

L'organisation de la table est optimale, elle est décomposée en éléments aussi petits que possible, la ville est clairement séparée de l'adresse et le nom du prénom.

2.4 Utiliser plusieurs tables

Prenons l'exemple de la gestion d'une collection de disque. On pourrait être tenté de créer une seule table contenant les informations suivantes :

Collection
Nom Auteur
Prénom Auteur
Titre Album
Année
Genre
Support

Et nous n'aurions pas spécialement tort, toutes les informations relatives à un album s'y trouvent. Pourquoi faire plus compliqué ?

Néanmoins, nous allons voir qu'il est plus judicieux de créer ici deux tables :

<table border="1"> <thead> <tr> <th>Auteurs</th> </tr> </thead> <tbody> <tr> <td>Nom Auteur</td> </tr> <tr> <td>Prénom Auteur</td> </tr> <tr> <td>Date de naissance</td> </tr> <tr> <td>Adresse</td> </tr> </tbody> </table>	Auteurs	Nom Auteur	Prénom Auteur	Date de naissance	Adresse	et	<table border="1"> <thead> <tr> <th>Albums</th> </tr> </thead> <tbody> <tr> <td>Titre Album</td> </tr> <tr> <td>Année</td> </tr> <tr> <td>Genre</td> </tr> <tr> <td>Support</td> </tr> </tbody> </table>	Albums	Titre Album	Année	Genre	Support
Auteurs												
Nom Auteur												
Prénom Auteur												
Date de naissance												
Adresse												
Albums												
Titre Album												
Année												
Genre												
Support												

Pourquoi ?

On va regrouper ensemble dans une même table toutes les informations relatives au même « sujet ». On voit bien ici qu'on peut regrouper ensemble les données concernant uniquement l'identité de l'auteur et, dans une autre table, les informations relatives uniquement à l'album (par exemple, l'année de sortie d'un album n'a pas de rapport direct avec l'identité de l'auteur, on la trouvera dans une autre table).

Un bon moyen pour trouver les tables utilisées est de partir de la situation existante :

Un *auteur* fait des *albums*, a un auteur peut correspondre plusieurs albums

On en déduit directement une table « auteurs » et une table « albums ».

Ce qui, au passage, nous permet d'enrichir les informations sur les auteurs, on peut en profiter pour par exemple ajouter la date de naissance ou son adresse, ce qui aurait été pénible dans la 1^{ère} version, en effet, il aurait fallu, pour chaque album d'un auteur, ressaisir à chaque coup sa date de naissance et son adresse (sans compter les erreurs de frappe d'un album à l'autre), là les informations ne sont saisies qu'une fois, ce qui est d'abord moins fatiguant (et ça, c'est le plus important) et ensuite, évite les fautes de frappes.

Bon, c'est bien, maintenant on a deux tables, mais du coup, on ne sait plus qui a fait quoi, les informations sont éparpillées. Il va donc falloir trouver un moyen pour savoir exactement à quel auteur correspond chaque album. Pour cela on va ajouter dans la table *auteur* un champ qui va identifier de façon unique chaque enregistrement, ce champ d'identification unique est appelé **clef primaire**.

A quoi sert cette clef? A pouvoir retrouver de façon non équivoque n'importe quel enregistrement dans la table « Auteurs », on ne peut en effet utiliser le nom ou le prénom comme clef primaire car deux auteurs peuvent avoir le même nom (rare mais pas impossible).

Le numéro de sécurité sociale est un bon exemple de champ clef primaire car il identifie de manière unique un individu. Même si deux individus ont le même nom et le même prénom, ils ne peuvent avoir le même numéro de sécurité sociale.

La table « Auteurs » ressemble alors à ceci (la clef primaire est en gras)

Auteurs
N° Auteur
Nom
Prénom
Date de naissance
Adresse

Et pour pouvoir associer un album à un auteur, il va falloir ajouter un champ dans la table « Albums », ce champ contient le n° de l'auteur de l'album

Albums
N° Auteur
Titre Album
Année
Genre
Support

On voit ici que cette organisation répond à une des exigences des bases de données, à savoir la non redondance des informations, dans la première organisation, le nom et le prénom de l'auteur étaient répétés autant de fois qu'ils avait fait d'albums, ici il ne sera présent qu'une seule fois. Lorsqu'on voudra afficher les caractéristiques d'un album, Access, grâce au numéro de l'auteur présent dans l'enregistrement de la table « Albums » ira chercher automatiquement les informations le concernant dans la table « Auteurs ». Cette organisation a un énorme avantage, le jour ou on veut modifier le nom ou le prénom d'un auteur, cette modification sera répercutée automatiquement sur les albums qu'il aura fait.

Clef primaire réalisée à partir de plusieurs champs

Lorsqu'on est sûr que le contenu d'une combinaison de champs ne se répétera pas dans une table, on peut utiliser cette combinaison de champs comme clef primaire. On aurait pu, à la place d'un n° d'auteur, utiliser la combinaison « nom + prénom + date de naissance », parce qu'il est rare que deux auteurs différents aient le même nom et le même prénom et la même date de naissance. Mais ce type de combinaison complexe est difficile à maintenir (il est plus simple de taper un numéro qu'une combinaison complexe de noms et de date, sans parler des erreurs de frappes) et en plus prendrait plus de place dans la base de données (on ne voit pas trop l'intérêt de découper les informations en deux tables si c'est pour retrouver dans le n° de l'auteur utilisé dans la table « Albums » quasiment toutes les informations contenues dans la table « Auteurs » !). Et pour terminer, ce genre de clef primaire complexe fonctionne mal dans Access.

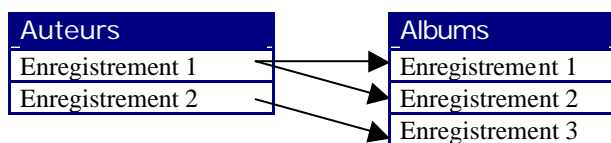
Grâce a cette clef, on va pouvoir mettre les deux tables en relation.

Établir une relation entre les tables de la base de données va permettre de réunir (pour les exploiter ensemble, les analyser, ...) les informations réparties dans différentes tables. La liaison se fait entre les tables à partir de la clef primaire, ici, on obtiendra la liste des albums d'un auteur grâce à la liaison entre la table *Auteur* et la table *Album* sur la clef primaire *n° auteur*.

2.5 Il existe trois types de relations entre les tables :

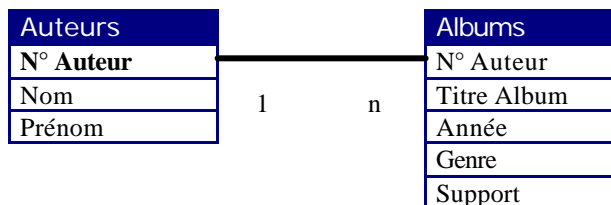
La relation de type « un à plusieurs » (1 – n)

Un enregistrement d'une table est en relation avec plusieurs enregistrements d'une autre table.



Exemple : a un auteur correspond un ou plusieurs albums (plusieurs flèches partent du même auteur), a chaque album ne correspond qu'un seul auteur.

La relation entre les deux tables se présente sous cette forme :



La relation « plusieurs à plusieurs » (n – m)

Un enregistrement de la table primaire peut être en relation avec plusieurs enregistrements de la table reliée et inversement, un enregistrement de la table reliée peut être en relation avec plusieurs enregistrements de la table primaire. Pour gérer ce type de relation, il faut la scinder en deux relations : une relation « un – plusieurs » et une relation « plusieurs – un ».

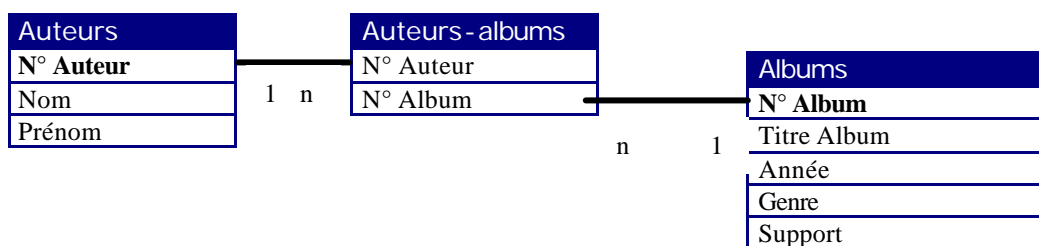
Exemple : On aurait pu avoir le cas où un album a plusieurs auteurs, là, ça ne marche plus, car telle que la table « Albums » est structurée, à un album ne peut correspondre qu'un auteur. Si on veut permettre qu'à un album correspondent plusieurs auteurs (on a alors une relation n – m), deux solutions s'offrent à nous :

1. La plus simple et la plus inélégante (les trucs les plus simples sont souvent inélégants) : ajouter quelques champs « auteurs » supplémentaires dans la table album pour arriver à quelque chose de ce genre, histoire de conserver une relation 1 – n :

Albums
N° Auteur
N° Auteur 2
N° Auteur 3
Titre Album
Année
Genre
Support

Ce genre de plaisanterie est à éviter à tout prix, en effet, si il n'y a qu'un seul auteur, les champs n° auteur 2 et n° auteur 3 ne vont rien contenir, et on va perdre de la place dans la base de données, et que va-t-il se passer si il y a quatre auteurs ?? vous répondrez qu'on peut prévoir quatre champs, ce à quoi je vous répondrez qu'il peut y avoir 5 auteurs, etc...

2. Il va donc falloir (on y coupe pas) créer une troisième table qui va servir d'intermédiaire entre la table « auteurs » et la table « albums » :



J'en entends déjà hurler !

Quelles sont les modifications ?

Nous sommes bien d'accord qu'avec un seul numéro d'auteur dans la table « albums » nous ne pouvons pas avoir plusieurs auteurs pour le même album. La solution 1 étant à proscrire (sauf dans le cas où on était sûr qu'il y aurait toujours 3 et seulement 3 auteurs), la troisième table est la seule solution.

Comment ? et bien, à chaque album correspond n enregistrements dans la nouvelle table « Auteurs-albums », chaque enregistrement renvoyant sur un seul enregistrement de la table « auteurs ». Votre esprit curieux aura noté qu'on a, pour les besoins de la création de cette nouvelle table, créé une clef primaire dans la table « albums ».

Pourquoi ? Pour faire correspondre un enregistrement de la table « Auteurs-albums » avec la table « Albums », il faut le faire sur la clef primaire, et comme on n'avait pas de clef primaire dans « Albums », il a bien fallu en créer une. Aucun des champs présents ne pouvait être utilisé comme clef primaire (la clef primaire, je le répète, doit identifier de façon UNIQUE un enregistrement) : deux albums peuvent avoir le même titre (rare mais pas impossible), peuvent être sortis la même année, être du même genre, etc..., on aurait pu utiliser une clef primaire complexe avec un couple (titre + année), plutôt rare que ce couple se retrouve deux fois, mais pour les raisons vues plus haut, on évitera d'utiliser ce type de clef.

Par exemple, on va avoir dans ces tables :

Auteurs

N° Auteur	Nom	Prénom
1	Crow	Sheryl
2	KS'Choice	
3	Sébastien	Patrick
4	Bezu	

Albums

N° Album	Titre Album	Année	Genre	Support
1	Cocoon Crash	1998	Good	CD
2	The Globe Sessions	1998	Good	CD
3	Tuesday Night Music Club	1993	Good	CD
4	Top NAZ Compil	1920	Debile-Blaireau	CD

Auteurs-Albums

N° Auteur	N° Album
1	2
1	3
2	1
3	4
4	4

Avec cette méthode, on peut affecter sans aucun problème n auteurs à m albums. (ici Sheryl Crow a deux albums, KS' Choice un seul, et l'album Top NAZ Compil a deux auteurs que je ne citerai pas)

Enfin, il existe un troisième type de relation, jamais utilisé: la relation de type «un – un»: un enregistrement d'une table est en relation avec un seul enregistrement d'une autre table et inversement. Ce type de relation ne doit pas se produire car les données de la table reliée n'ont aucune raison de se trouver là, comme elles ne correspondent qu'à un seule enregistrement de la 1^{ère} table, elles devraient être déplacées dans cette table.

Encore un peu de théorie avant de s'amuser avec Accès.

2.6 Améliorations

On peut encore améliorer le schéma de nos tables. En effet, pour la même raison qui nous avait fait séparer en deux tables « Albums » et « Auteurs » notre table de départ pour éviter à avoir à ressaisir les informations concernant un même auteur pour chacun de ses albums, il est ici possible d'effectuer à nouveau une division de la table « Albums ».

En effet, pour chaque album, on va indiquer le genre (rock, pop, reggae, rap etc...) et le support (CD, vinyl, K7, DAT, minidisc, etc...), ces informations vont être redondantes et pourraient être optimisées de la façon suivante :

- 1) On crée une table « genre » et une table « support »

Genre
N° Genre
Nom Genre

Support
N° Support
Nom support

- 2) On modifie la table « albums »

Albums
N° Auteur
N° Auteur 2
N° Auteur 3
Titre Album
Année
N° Genre
N° Support

Maintenant, au lieu d'indiquer explicitement pour chaque album son support et son genre, on indiquera un n° de genre et un n° de support. L'intérêt est triple : gain de place en mémoire (stocker un numéro prend moins de place que stocker une chaîne de caractères), gain de vitesse (entrer un numéro est plus rapide que taper le mot complet), optimisation de la structure (on pourra changer le nom d'un support, ce changement affectera instantanément l'ensemble des albums).

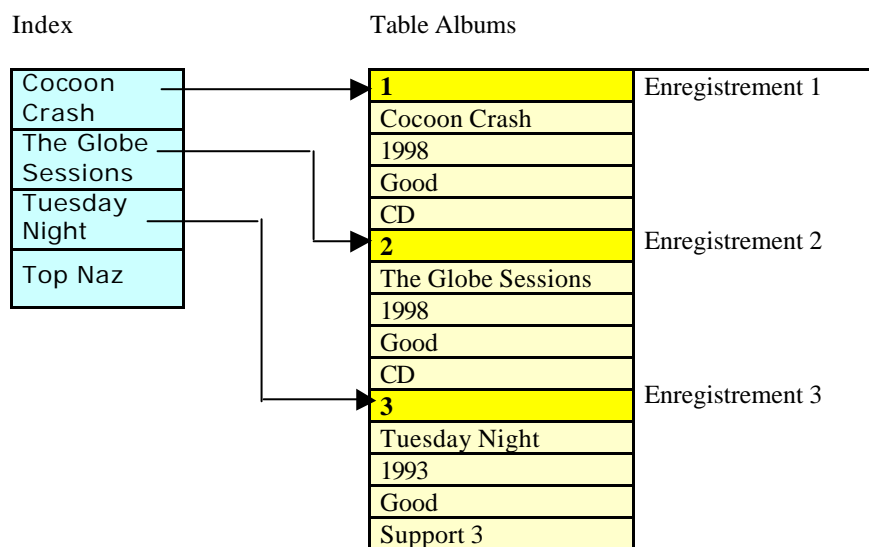
2.7 Les index

Pour optimiser les temps de recherche dans les tables, Access (et d'ailleurs tous les SGBD) ont besoin d'index. L'index permet à Access de retrouver plus rapidement les enregistrements concernés ; il accélère aussi les opérations de tri.

Comment ? Les index fonctionnent avec une table comme la table des matières avec un livre. Lorsqu'on indexe un champ, Access stocke hors de la table les valeurs de ce champ. Lorsqu'on effectue une recherche sur ce champ, plutôt que de le chercher dans la table, Access va faire sa recherche dans l'index, une fois qu'il aura trouvé, il affichera le contenu de l'enregistrement associé à cet index.

Petit crobard :

Supposons que la table « Albums » soit indexée sur le titre de l'album



Lorsqu'on cherchera un enregistrement de la table albums à partir du titre de l'album, Access parcourra l'index et affichera l'enregistrement correspondant à l'index qu'il aura trouvé.

Un index peut être composé d'un ou de plusieurs champs, on aurait pu indexer la table albums sur un index « titre + année » par exemple (si ça peut servir à quelque chose), de même, on peut utiliser plusieurs index pour une table, par exemple, on aurait pu indexer la table album sur les champs Titre, Genre et Support.

Pourquoi alors ne pas tout indexer ? L'idée peut sembler bonne, cela accélérerait les recherches et les tris mais ralentirait toutes les mises à jour de la table, en effet, à chaque fois qu'un champ index est modifié dans la table, il faut le modifier également dans l'index, ce qui peut coûter très cher en temps. On utilisera donc les index avec parcimonie et uniquement si ils s'avèrent utiles.

3. Application

Maintenant que vous avez tout compris au film, nous allons nous pencher sur un autre cas de figure. Supposons qu'on veuille gérer un magasin avec Access. Créez donc les tables et les relations pour notre magasin, au cas où certains ignoreraient le fonctionnement d'un magasin, voici comment ça marche :

- ?? Un client commande des produits
- ?? Un fournisseur livre des produits
- ?? Le stock contient des produits

Et voilà une solution qui semble honnête :

(Les clefs primaires sont en gras, les indexes sont soulignés)

Le choix des index est totalement subjectif, il sera surtout fait en fonction des besoins ultérieurs. Ce n'est pas grave si on n'a pas choisi les bons index, il est toujours possible de les modifier ultérieurement.

Une table « clients »

Clients
N° Client
Titre
<u>Nom</u>
Prénom
Adresse
Code Postal
Ville
Observations

Une table « produits »

Produits
Code produit
<u>Désignation</u>
Prix unitaire
Taux TVA
Stock

Une table « Commandes » qui va indiquer ce qu'a commandé le client

Commandes
N° Commande
<u>N° Client</u>
Date de la commande
Commande réglée

On est d'accord, une commande peut avoir plusieurs lignes (plus d'un produit commandé par commande, ça vaut mieux), pour cela, on va avoir une table « lignes de commande » qui va « relier » la commande aux produits.

Pourquoi n'avoir pas mis ça directement dans la table commande ? on retombe sur notre problème de tout à l'heure : parce qu'on ne sait pas par avance le nombre de lignes qu'on peut avoir dans une commande.

On peut voir qu'un produit peut se retrouver dans n commandes, et que une commande peut faire référence à m articles, on a donc entre les tables « commandes » et « articles », une relation n – m (plusieurs à plusieurs) qu'Access ne peut gérer (ni d'ailleurs aucun autre SGBD), il faut donc transformer cette relation en deux autres relations 1-n et n-1 en utilisant une table supplémentaire, d'où l'apparition de cette table.

Lignes commande
N° Commande
Code produit
Quantité

Pour les fournisseurs, on a quelque chose de très ressemblant

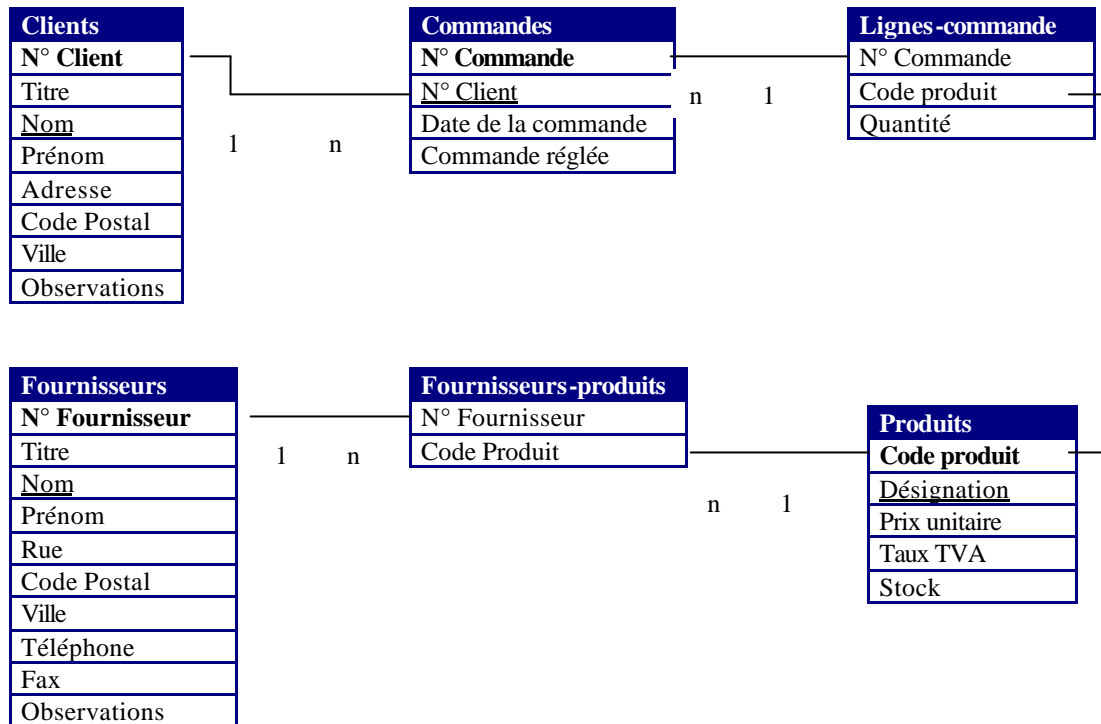
Une table « fournisseurs »

Fournisseurs
N° Fournisseur
Titre
<u>Nom</u>
Prénom
Rue
Code Postal
Ville
Téléphone
Fax
Observations

Même problème que précédemment, un fournisseur peut livrer plusieurs produits et le même produit peut être livré par plusieurs fournisseur, donc relation n-m, donc création d'une table intermédiaire

Fournisseurs-produits
N° Fournisseur
Code Produit

Et voilà le travail d'artiste, les relations entre tout ce beau monde donnent ceci



On aurait pu dire qu'un produit pouvait être livré par un seul fournisseur, dans ce cas, la table intermédiaire « Fournisseurs-Produits » aurait été inutile, pour savoir quel était le fournisseur d'un produit, il suffisait d'ajouter un champ « n° fournisseur » dans la table produit. Il aurait contenu, pour chaque produit, le n° de fournisseur le livrant.

Dernier exemple pour la route

On veut modéliser le problème d'une partie de la gestion de la scolarité d'un établissement.
On connaît les règles suivantes :

- ?? Les enseignements sont dispensés sous forme de cours
- ?? Un étudiant peut s'inscrire à plusieurs cours
- ?? Un enseignant est rattaché à un ou plusieurs établissements et peut enseigner plusieurs cours
- ?? Un cours peut être enseigné par plusieurs enseignants
- ?? Certains enseignants peuvent être responsables des autres

A vous de jouer... Donnez les différentes tables qui vont être utilisées et les relations entre elles (on s'intéresse pas spécialement au contenu exact des tables)

Les tables qui vont être utilisées sont :

Cours
N° cours
Nom
...

Étudiants
N° étudiant
Nom
Prénom
...

Inscriptions
N° étudiant
N° cours
Note semestre
...

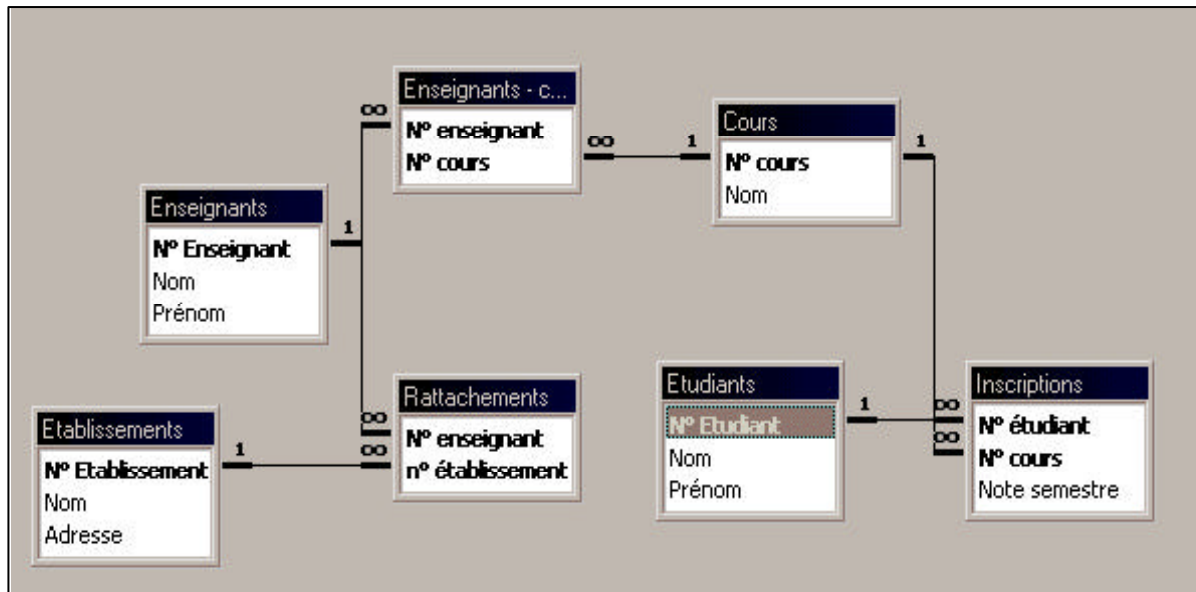
Enseignants
N° enseignant
Responsable
Nom
Prénom
...

Enseignants - cours
N° enseignant
N° cours

Établissements
N° établissement
Nom
Adresse
...

Rattachements
N° enseignant
N° établissement

Les relations vont être :

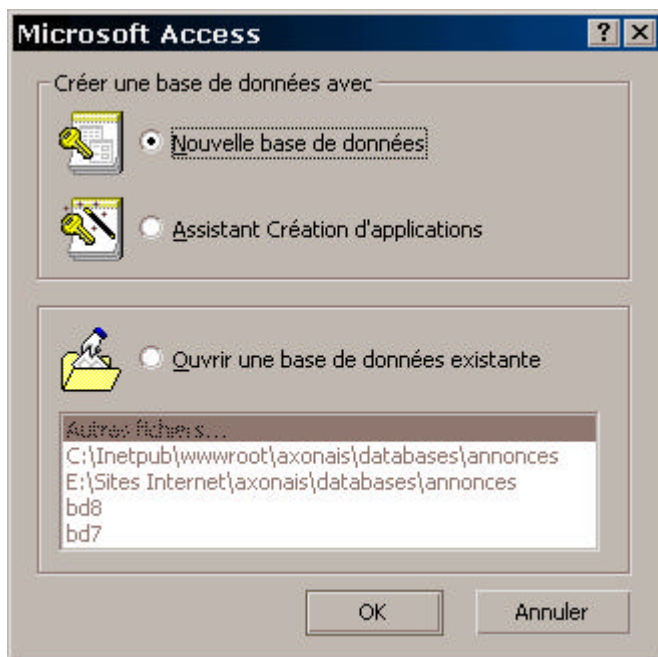


(fait avec Access !)

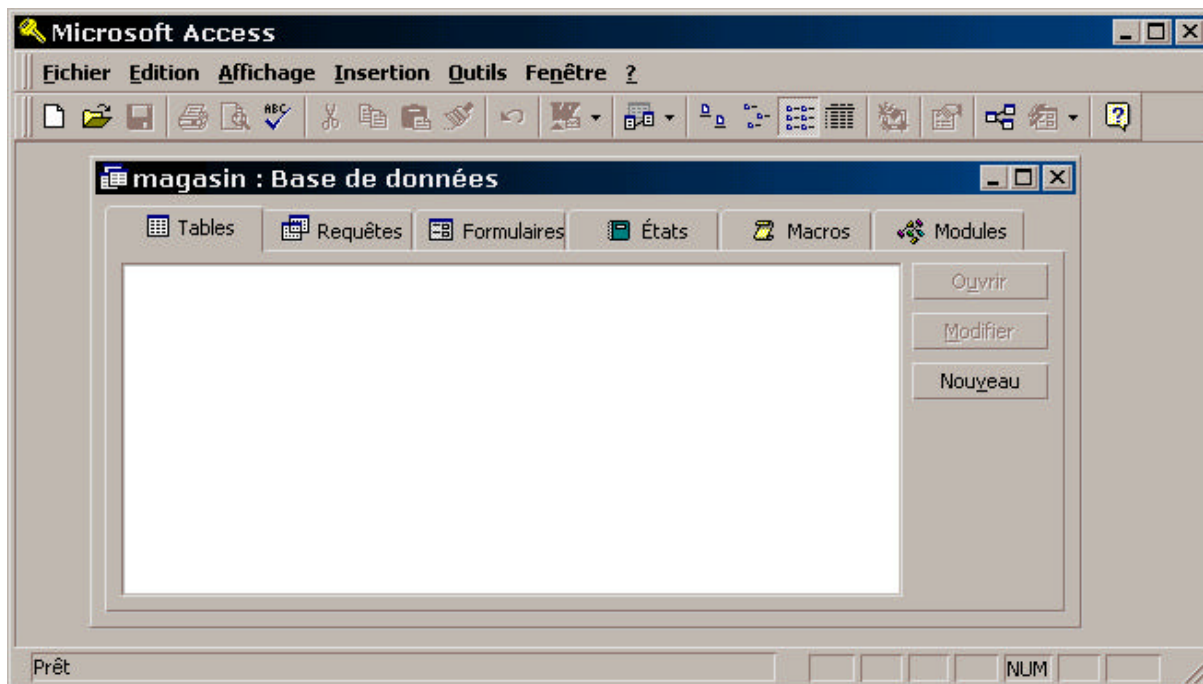
CREATION D'UNE BASE DE DONNEES AVEC ACCESS

Nous allons créer la base de données « Magasin » qui va contenir les tables dont nous avons vu la structure plus haut.

3.1 Création de la base de données



Cliquer sur "Nouvelle base de données" et entrer « magasin.mdb » comme nom de la base.



La fenêtre Base de Données s'affiche, elle contient tous les « objets » qui peuvent composer une base Access :

Les Tables : Chaque table de la base va contenir les données se rapportant à un sujet particulier, la table «clients» va contenir les informations sur les clients. C'est l'objet fondamental de la base de données, toute requête, formulaire ou état va être basé sur une ou plusieurs tables.

Les Requêtes : Les requêtes vont être utilisées pour obtenir des données contenues dans une ou plusieurs tables à partir d'une question. Par exemple, une requête va pouvoir nous afficher la liste des clients habitant à Paris.

Les Formulaires : Le formulaire est utilisé pour faciliter la saisie et la modification des données d'une table, par exemple, le formulaire "client" va permettre d'entrer de façon conviviale les informations concernant un client. On peut les saisir sans formulaire, ce que nous verrons, mais cette méthode est moins conviviale.

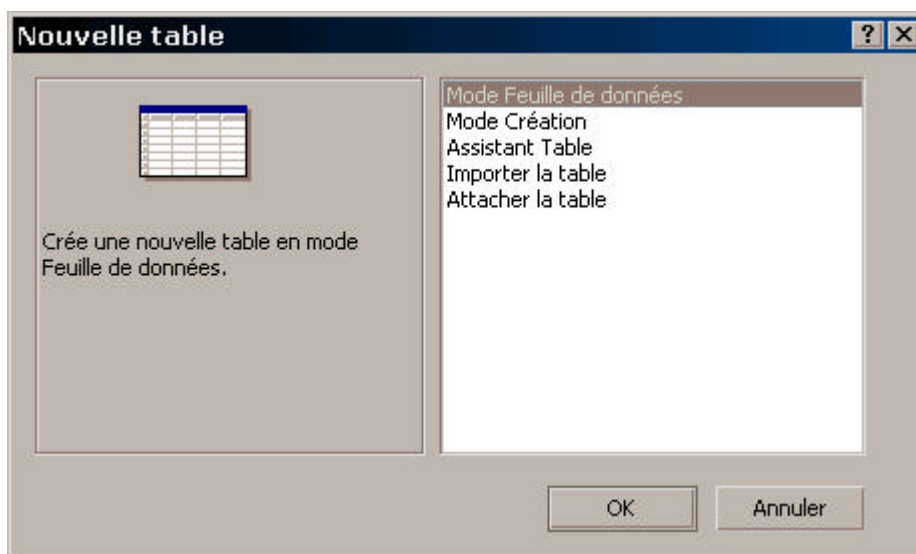
Les États : Les états permettent d'imprimer des données contenues dans des tables selon une présentation définie en y intégrant éventuellement des calculs.

Les Macros : Les macro commandes permettent d'automatiser une suite d'opérations répétitives.

Les Modules : Les modules sont des programmes écrits en Visual Basic for Application (VBA) pour réaliser des opérations qui seraient trop complexes en utilisant les seules fonctionnalités d'Access.

3.1 Création d'une table

Cliquer sur **Nouveau**



Access propose 5 méthodes pour créer une table :

- ?? **Mode feuilles de données** : ce mode permet de saisir directement des informations dans la table sans se préoccuper du nom ou du format des champs, la définition du nom et du format sera faite plus tard.
- ?? **Mode création** : On utilisera cette méthode le plus souvent, elle permet de créer les noms et la structure de chaque champ de la table.
- ?? **Assistant Table** : Access propose différents modèles de tables prédéfinis (carnet d'adresses, factures, etc...) et crée la table, vous devrez la personnaliser ensuite pour qu'elle réponde à vos besoins.
- ?? **Importer la table** : Si les données de votre table sont déjà contenues dans une autre table, dans un fichier ou dans une autre base de données, vous pouvez utiliser cette méthode pour importer vos données dans une nouvelle table.
- ?? **Attacher une table** : Cette option sert à attacher une table avec une autre table contenue dans une autre base de données, nous ne l'utiliserons pas.

Nous utiliserons ici le mode création :

La création de la table s'effectue en remplissant les trois colonnes « **Nom du champ** », « **Type de données** » et « **Description** »

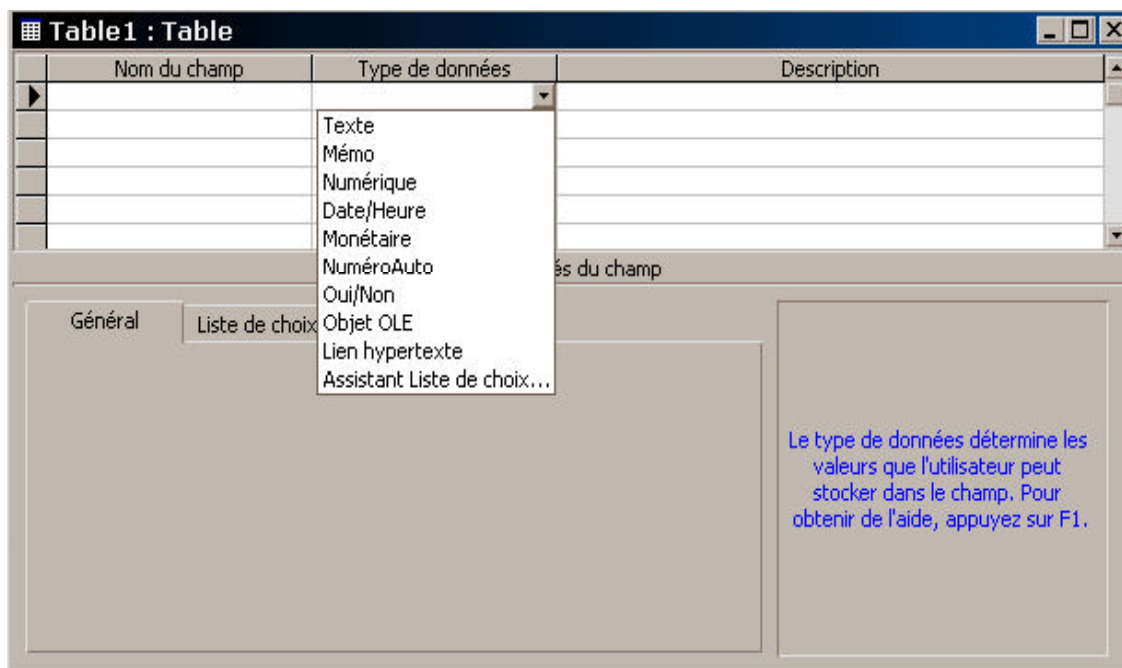
Chaque ligne de ce tableau contient les informations de chaque champ :

- ?? Le nom de chaque champ de la table (de 1 à 64 caractères)
- ?? Le type de données (voir plus bas) à choisir parmi 9 types proposés par Access
- ?? La description du champ (255 caractères maximum)

3.2 Les types de données

Chaque champ peut contenir des données de différents types : le nom du client va contenir des caractères alphabétiques, le code postal va contenir des chiffres, la date de livraison une date, un champ prix va contenir des valeurs monétaires.

Il faut choisir, pour chaque champ de la table, le type de données le plus approprié, en effet, rien n'interdit de choisir comme type de données pour un champ contenant une date ou un code postal un type « texte » ou pour champ contenant un prix un type « numérique », mais autant utiliser les types les plus appropriés !



Les différents types sont :

- ??**Texte** : Pour un champ destiné à contenir des caractères alphanumériques (lettres et/ou chiffres), attention, le champ de type texte ne peut contenir plus de 255 caractères, on l'utilise pour un nom, une adresse, etc...
- ??**Mémo** : Même utilisation que le type texte, mais le champ de type mémo peut contenir jusque 65535 caractères (64 Ko). Utilisé pour une description longue par exemple.
- ??**Numérique** : Le champ ne pourra contenir que des nombres avec ou sans décimales. (un code postal, un n° d'identification par exemple)
- ??**Date/Heure** : Le champ ne pourra contenir que des dates ou des heures (Access vérifie la validité des dates ou des heures saisies)
- ??**Monétaire** : Valeurs présentées sous format monétaire (exemple : 1 23,45F)

NuméroAuto Valeur numérique incrémentée automatiquement lors de la saisie de chaque enregistrement. Si vous choisissez ce type de champ, vous ne pourrez rien saisir dedans, c'est Access qui y placera un nombre automatiquement à chaque fois que vous créez un nouvel enregistrement. On l'utilisera lorsqu'on veut être sûr que le contenu de ce champ ne se retrouvera pas dans un autre enregistrement de la table, par exemple pour le numéro de client : chaque nouveau client aura un numéro unique automatiquement affecté par Access, c'est un champ de ce type qui sera le plus souvent utilisé pour être une clef primaire.

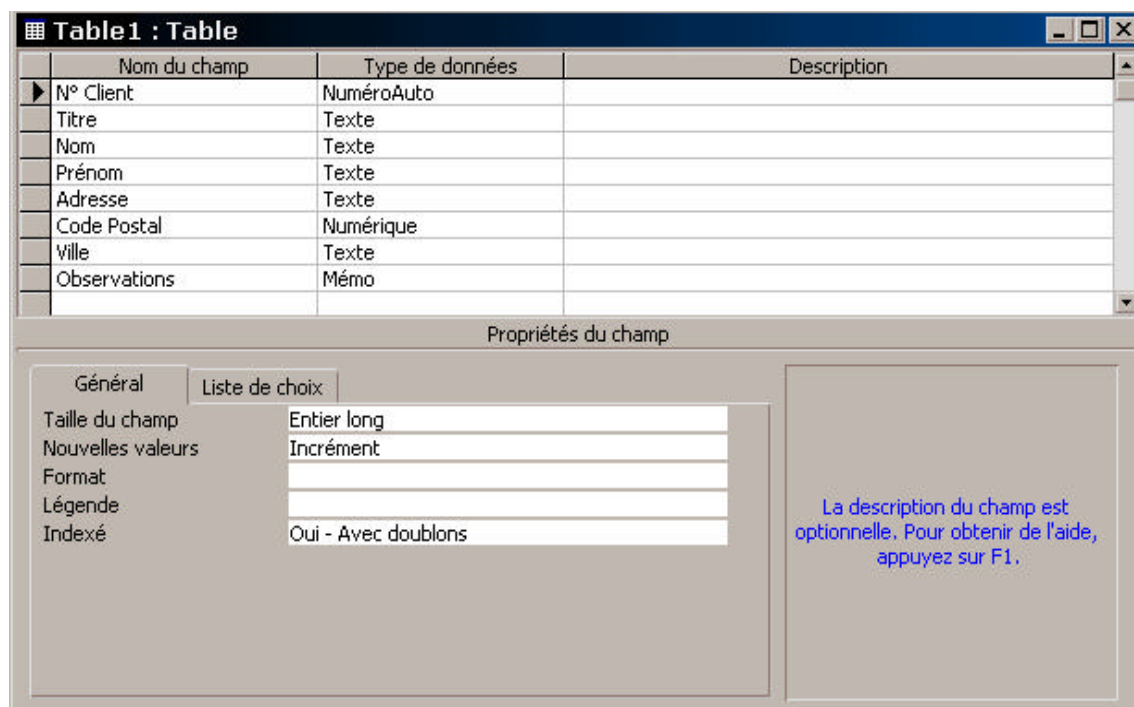
Oui/Non : Seules deux données sont autorisées dans ce champ : Oui et Non (on utilisera ce type de données par exemple avec un champ « réglé » qui indiquera si une facture a été réglée ou non)

Objet OLE : OLE (Object Linked and Embedded) est une technologie utilisée par Windows, elle permet d'insérer dans une application des objets provenant d'autres applications, on utilisera un champ de ce type pour insérer dans la table une image, un son, un fichier Word, etc...

Lien Hypertexte Un champ de ce type contiendra une adresse Internet sur laquelle on pourra cliquer directement.

La liste de choix n'est pas vraiment un type de données, nous l'étudierons plus tard.

Une fois la table « **Clients** » correctement renseignée, on devrait avoir quelque chose qui ressemble à ça :



3.3 Les propriétés des champs

Comme vous pouvez le constater, chaque champ a des propriétés qui dépendent de son type de données, ces propriétés vont déterminer la façon dont les données qu'il contient vont être affichées ou comment les données vont être saisies dans ce champ. Ces propriétés sont affichées dans l'onglet « Général » en bas de la fenêtre. L'onglet « Général » affiche chaque propriété du champ et, à côté, dans des champs modifiables, la valeur de cette propriété.

Ces propriétés varient selon le type de données (par exemple, un champ numérique aura une propriété « décimales » qui indiquent le nombre de chiffres après la virgule que ne peut avoir évidemment un champ de type texte)

Propriétés des champs de type Texte et Mémo	
Taille du champ	Nombre de caractères maximum qui peuvent être saisis
Valeur par défaut	Texte qui sera contenu par défaut dans ce champ
Valide si	Expression pour valider la saisie, exemple : <> « machin », interdira la saisie du mot « machin » dans ce champ
Message si erreur	Message d'erreur qui va être affiché si l'expression saisie n'est pas valide, dans notre exemple, on pourrait mettre ici «saisie de machin interdite »
Null Interdit	Si oui, il faudra obligatoirement saisir une valeur dans ce champ
Chaîne vide autorisée	La chaîne vide est "", on peut l'utiliser lorsqu'on n'a rien à entrer dans un champ obligatoire, si cette propriété est non, il faudra obligatoirement saisir du texte dans ce champ
Indexée	Le champ est un index avec ou sans doublons, si l'index n'a pas de doublons, il ne sera pas possible de saisir deux fois le même contenu pour ce champ dans la table
Format	Va définir comment le contenu du champ va être affiché (voir plus loin)
Masque de saisie	Oblige à saisir le contenu du champ selon un format précis (par exemple un numéro de téléphone), on verra ça plus loin

Tous les autres types de données ont des propriétés similaires, certains ont des propriétés supplémentaires :

Propriétés des champs de type numérique	
Taille du champ	<p>Définit la valeur maximale qui va pouvoir être contenue dans ce champ. Il faudra faire attention et bien penser à l'avance quelle sera la valeur maximale permise dans ce champ pour pouvoir ajuster au mieux cette propriété :</p> <p><i>Octet</i> : de 0 à 255 <i>Entier</i> : de -32767 à +32768 <i>Long</i> : de -2147483648 à +2147483649 <i>Réel Simple</i> : de $-3.48^{E}38$ à $+3.40^{E}38$ <i>Réel Double</i> : de $-1.79^{E}308$ à $+4.96^{E}324$</p> <p>Plus on descend, plus le champ occupera de la place en mémoire et sur le disque dur (le Réel Double prend 8 fois plus de place que l'octet), pensez-y si la table contient des milliers de lignes...</p>
Décimales	Nombres de chiffres après la virgule

Propriétés des champs de type NuméroAuto	
Nouvelles valeurs	Indique quelle sera la prochaine valeur contenue dans un champ de type NuméroAuto : avec Incrément , la valeur sera égale à la valeur créée dans le précédent enregistrement+1, avec Aléatoire , Accès remplira ce champ avec une valeur tirée au hasard (mais qui ne se retrouvera pas dans un autre enregistrement).

3.4 Le format d'affichage

Nous avons vu plus haut qu'on peut modifier la façon dont les données contenues dans les champs peuvent être affichées ou forcer leur saisie selon un format précis.

La façon dont les données sont affichées se fait par l'intermédiaire de la propriété "Format d'affichage".

Attention, la façon dont les données sont affichées dans un champ ne modifie en rien le contenu de ce champ dans la table. Par exemple, si vous forcez à afficher un '1' avant le contenu d'un champ, si ce champ contient '234', la valeur affichée sera '1234' mais le contenu du champ dans la table sera toujours '234'. La propriété 'format d'affichage' n'influe que sur la façon dont les données d'un champ vont être affichées.

Pour modifier le format d'affichage, on utilise les codes suivants (quand ils marchent...) :

Codes du format d'affichage	
"texte"	Affiche le texte "texte" dans le champ
!	Justifie à droite le contenu du champ
*c	Remplace les espaces par le caractère c
[couleur]	Affiche le contenu du champ dans la couleur "couleur"

Il y a des formats d'affichages plus spécifiques à certains types de données :

Pour les champs texte et mémo

Formats d'affichage spécifiques aux champs Texte et Mémo	
@	Va afficher un espace si on a rien saisi
<	Va afficher le texte en minuscules
>	Va afficher le texte en majuscules

Pour les champs Numériques et Monétaires

Formats d'affichage spécifiques aux champs Numériques et Monétaires	
Nombre général	Affiche le nombre tel qu'il a été saisi
Monétaire	Nombre avec séparateur de milliers + 2 chiffres après la virgule + symbole monétaire
Fixe	Affiche au moins un chiffre + 2 chiffres après la virgule
Standard	Nombre avec séparateur de milliers + 2 chiffres après la virgule
Pourcentage	Multiplie le nombre par 100 et ajoute le symbole %
Scientifique	Nombre au format décimal avec exposant

On peut définir son propre format d'affichage pour les champs numériques ou monétaires si le format que l'on désire ne fait pas partie des formats proposés ci-dessus.

Pour cela, on utilise un format d'affichage spécial, ce format est défini par une chaîne de caractères qui a le format suivant (vous tapez cette chaîne de caractères dans le champ 'format' de la propriété du champ) :

; ; ;
 >0 <0 =0 NULL

La première partie (avant le premier symbole ";") va définir comment va être l'affichage si le contenu du champ est supérieur à 0, la deuxième partie, si le contenu du champ est inférieur à 0, la troisième partie si le contenu du champ = 0 et la dernière partie si le contenu du champ est vide.

Chaque partie est une combinaison des symboles suivants :

Symbole	Signification
,	Séparateur de milliers
.	Séparateur décimal
0	Un chiffre ou zéro
#	Un chiffre ou rien
%	Pourcentage (* 100) + symbole %
E-	Affichage scientifique, exposant positif non signé
E+	Affichage scientifique, exposant positif signé

Par exemple :

\$#,##0.00[vert];(\$#,##0.00)[rouge];"Zéro";"Vide"

Que va-t-il se passer avec ce format d'affichage ?

- ?? Si le nombre est supérieur à 0, il sera affiché en vert avec au moins un chiffre avant la virgule et deux après la virgule.
- ?? Si il est inférieur à zéro, il sera affiché en rouge, entre parenthèses avec au moins un chiffre avant la virgule et deux après la virgule.
- ?? Si il est égal à zéro, il sera affiché "Zéro".
- ?? Si le champ est vide, il sera affiché "Vide".

Notez que les symboles # et 0 symbolisent chaque chiffre du nombre qui doit être affiché.

Pour les champs de type Date / Heure

Formats d'affichage spécifiques aux champs Date / Heure	
Date, général	Par défaut, affiche la date sous le format JJ/MM/AA HH:MM:SS (si il n'y a pas d'heure, affiche seulement la date)
Date, Complet	Affiche la date complète comme prévue dans le panneau de configuration de Windows Par exemple : lundi 1 janvier 1999
Date, Réduit	Affiche : 1-jan-99
Date, Abrégé	Affiche la date abrégée comme prévue dans le panneau de configuration de Windows Par exemple : 1/1/99
Heure, Complet	Affiche l'heure comme prévue dans le panneau de configuration de Windows Par exemple : 17:34:22
Heure, Réduit	Exemple : 5:34 PM
Heure, Abrégé	Exemple : 17:34

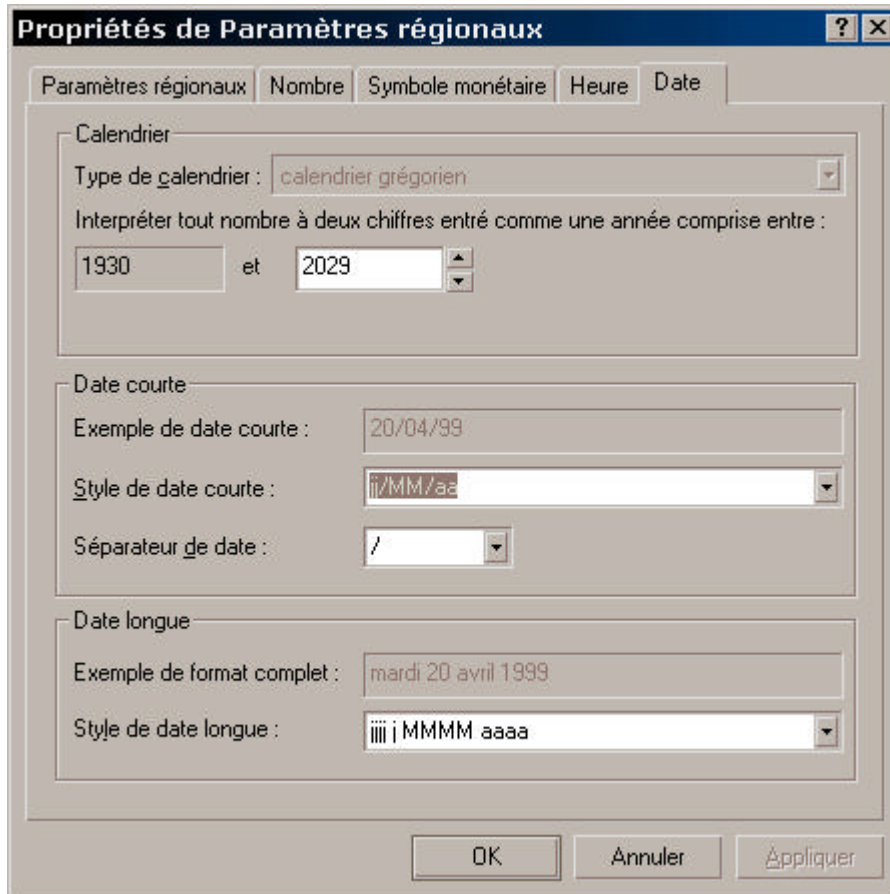
Comme pour les champs de type numériques ou date, on peut définir son propre format d'affichage en utilisant une combinaison des symboles suivants :

Symbole	Signification
:	Séparateur d'heure
/	Séparateur de date
c	Identique au format "Date, Général"
j	Jour du mois (1 à 31)
jj	Jour du mois sur deux chiffres (01 à 31)
jjj	Trois premières lettres du jour (dim à sam)
jjjj	Nom entier du jour de la semaine (dimanche à samedi)
jjjjj	Identique au format "Date, Abrégé"
jjjjjj	Identique au format "Date, Complet"
e	Jour de la semaine (1 à 7)
ee	Semaine de l'année (1 à 53)
m	Mois de l'année (1 à 12)

mm	Mois de l'année sur 2 chiffres (01 à 12)
mmm	Trois premières lettres du mois (jan à déc)
mmmm	Nom entier du mois (janvier à décembre)
t	Trimestre (1 à 4)
A	Jour de l'année (1 à 366)
aa	Deniers chiffres de l'année (00 à 99)
aaaa	Année complète (0100 à 9999)
h	Heure (0 à 23)
hh	Heure sur 2 chiffres (00 à 23)
n	Minutes (0 à 59)
nn	Minutes sur 2 chiffres (00 à 59)
s	Secondes (1 à 59)
ss	Secondes sur 2 chiffres (00 à 59)
Tttt	Identique au format "Heure, Complet"
AM/PM	Heure sur 12 heures + AM ou PM
am/pm	Heure sur 12 heures + am ou pm
A/P	Heure sur 12 heures + A ou P
a/m	Heure sur 12 heures + a ou p
AMPM	Heure sur 12 heures + indicateur matin/après-midi défini dans le panneau de configuration

Format d'affichage des dates / heures dans Windows

Pour définir le format d'affichage par défaut dans Windows, allez dans le panneau de configuration et double cliquez sur l'icône "**paramètres régionaux**" (une planète habituellement) :



Il y a deux onglets : Date et Heure qui définissent comment vont s'afficher la date et l'heure dans tous les logiciels Windows.

Pour les champs de type OUI/NON

Access propose pour ce type de champ, trois types d'affichage par défaut : à la place de OUI ou NON, on peut afficher "Vrai" ou "Faux" ou alors "Actif" ou "Inactif", si on veut afficher un autre message, il faut dans le champ format de la propriété du champ taper la commande suivante :

```
; "texte 1" ; "texte 2"
```

"texte 1" sera affiché à la place de Oui et "texte 2" à la place de Non.

On peut, bien sur, utiliser des codes d'affichages supplémentaires, par exemple :

```
; "OK" [vert] ; "NON" [Rouge]
```

affichera OK en vert à la place de Oui et NON en rouge à la place de Non.

N'oubliez pas que la ligne commence par le symbole ";"

3.5 Le Masque de saisie

Le masque de saisie permet de faciliter la saisie dans un champ en forçant l'utilisateur à entrer les données selon un format déterminé, les informations stockées dans la table auront le format défini dans le masque de saisie contrairement au format d'affichage qui affectait seulement la façon dont les données contenues dans un champ allaient être affichées.

Le masque de saisie, comme le format d'affichage est constitué d'une combinaison de caractères :

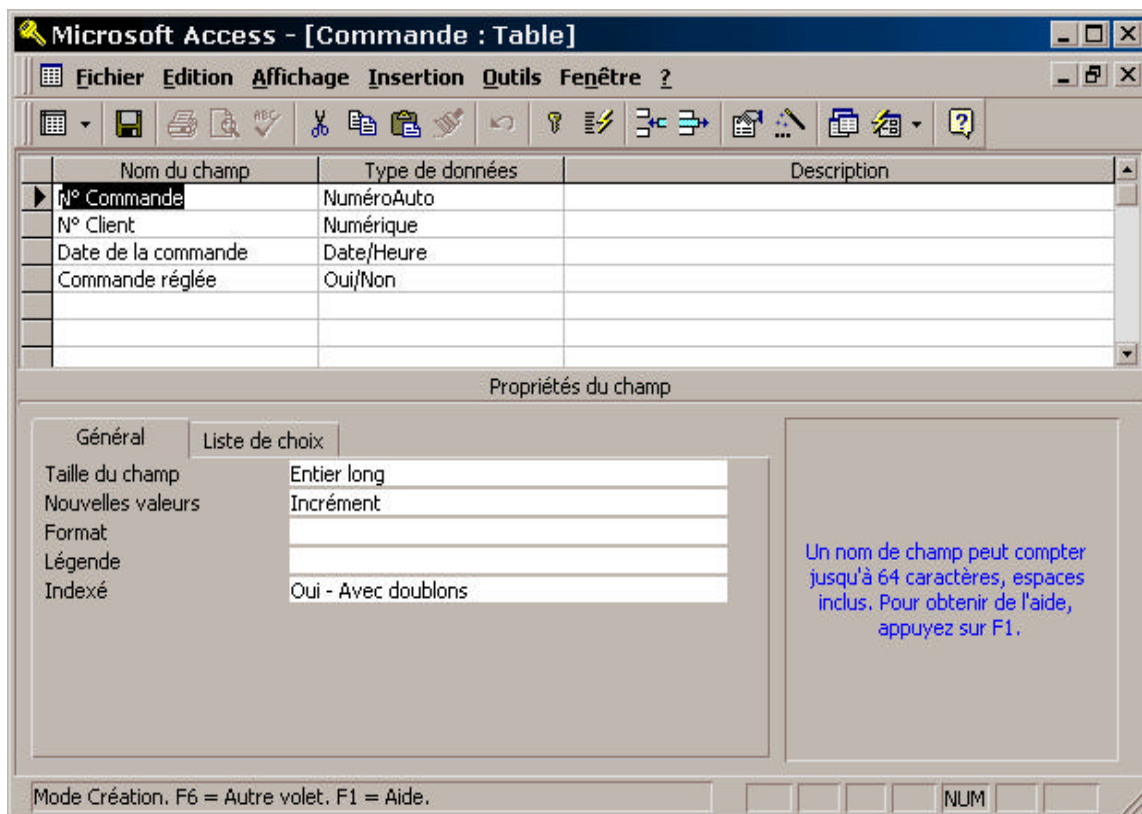
Symbole	Signification
0	Chiffre de 0 à 9 obligatoire
9	Chiffre ou espace facultatif
#	Chiffre ou espace ou + ou -
L	Lettre de A à Z obligatoire
?	Lettre de A à Z non obligatoire
A	Lettre ou chiffre obligatoire
a	Lettre ou chiffre facultatif
&	Caractère quelconque obligatoire
C	Caractère quelconque facultatif
<	Passe en minuscules
>	Passe en majuscules
!	Saisie à partir de la droite

Par exemple, si on veut saisir 5 chiffres obligatoirement pour un code postal, on utilisera : 00000, si on veut saisir une suite de 3 chiffres et de 3 lettres : 000LLL, si on veut saisir un numéro de téléphone : 00-00-00-00-00, ou un nom de famille dont la première lettre est toujours en majuscules : >L<???????????? (prévoir autant de ? que le nom peut comporter de lettres)

3.6 Les Listes de choix

Nous avons vu précédemment que parmi les types de données est proposé un type de données particulier : la liste de choix, la liste de choix n'est pas à proprement parler un type de données particulier, c'est un moyen de simplifier la saisie de données dans une table en proposant à l'utilisateur de cliquer sur un élément proposé dans une liste, le champ de la table sera rempli avec l'élément sélectionné dans la liste.

Pour pouvoir tester cela, nous allons créer la table "Commandes"



Elle doit ressembler à ça.

Nous allons maintenant retourner dans la table client et modifier le type de données du champ "Titre", Le champ "Titre" va contenir le titre du client : Monsieur, Madame, Mademoiselle. Au lieu de saisir à chaque fois le mot en entier, nous allons créer une liste de choix qui proposera ces trois titres, il suffira de cliquer sur l'un d'entre eux pour remplir automatiquement le champ.

Après avoir choisi le type "Liste de Choix" pour le champ "Titre", la fenêtre suivante s'affiche

Assistant Liste de choix

Cet Assistant crée une liste de choix, qui affiche une série de valeurs que vous pouvez sélectionner.

Comment souhaitez-vous que votre liste de choix obtienne ces valeurs ?

Je veux que la liste de choix recherche les valeurs dans une table ou requête.

Je taperai les valeurs souhaitées.

Annuler < Précédent Suivant > Terminer

Il y a deux façons de créer une liste de choix : soit à partir du contenu d'une autre table (ce que nous verrons après), soit à partir d'une liste que l'on va taper soi-même une fois pour toutes, c'est ce que nous allons choisir. (Cliquez ensuite sur Suivant)

Assistant Liste de choix

Quelles valeurs souhaitez-vous dans votre liste de choix? Entrez le nombre de colonnes souhaité dans la liste, et tapez les valeurs souhaitées dans chaque cellule.

Pour ajuster la largeur d'une colonne, déplacez le bord droit jusqu'à la largeur souhaitée, ou cliquez deux fois sur ce bord droit de la colonne afin d'obtenir la meilleure largeur.

Nombre de colonnes:

	Col1
	Madame
	Mademoiselle
	Monsieur
	*

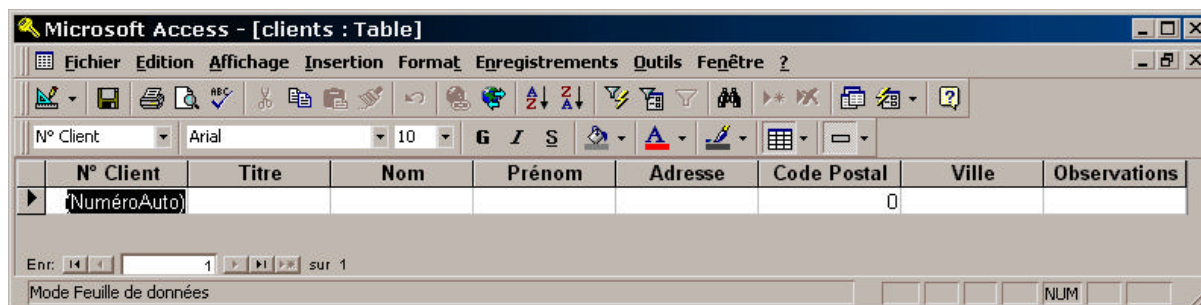
Annuler < Précédent Suivant > Terminer

On va taper dans la colonne Col1 la liste des titres possibles et cliquer sur Suivant une fois terminé.

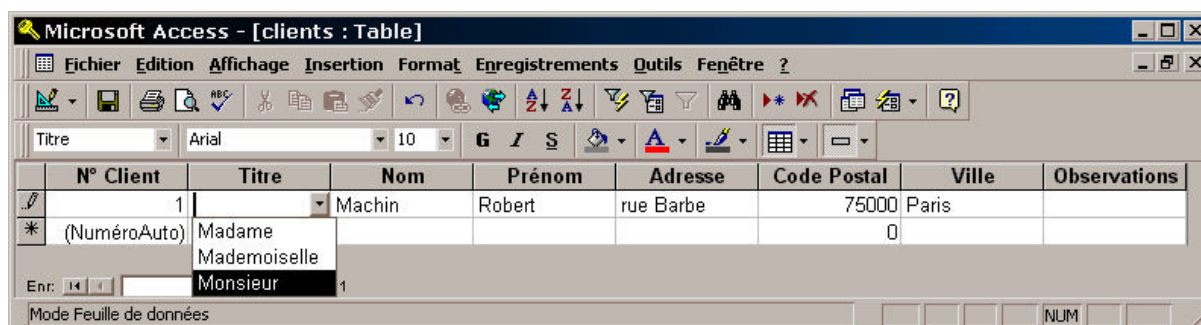


On nomme ici la liste de choix (peu importe le nom choisi) et on clique sur "Terminer"

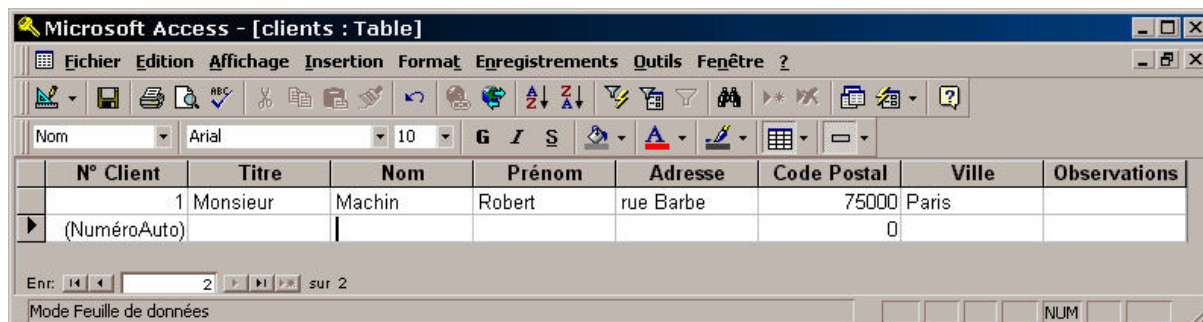
Maintenant, pour voir le résultat, nous allons saisir des données dans la table "Clients", dans la fenêtre où se trouve la liste des tables, on clique sur "**Ouvrir**", on arrive alors sur un tableau qui ressemble à ça :



Il y a une ligne par enregistrement, et une colonne par champ, on se positionne sur le champ désiré en cliquant dessus et on saisi directement les données.

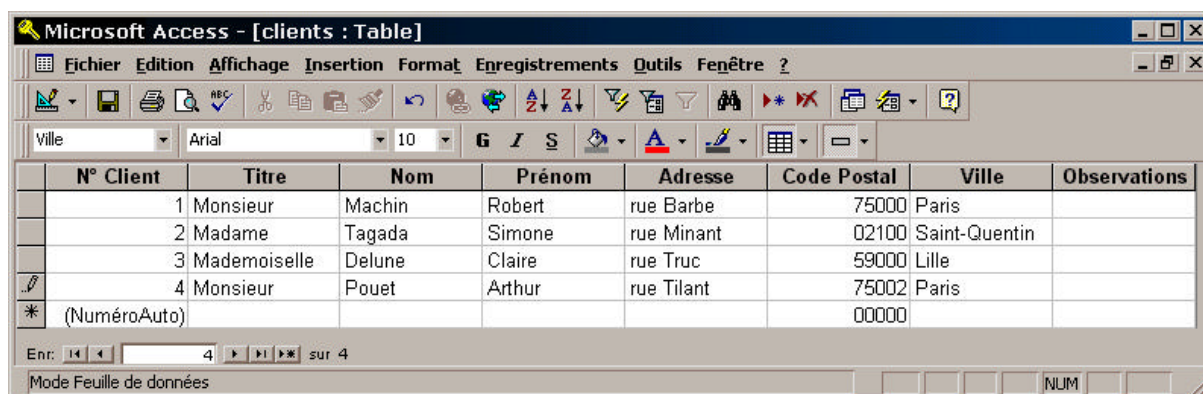


Et que se passe-t-il lorsqu'on clique sur "Titre" ? La liste de choix apparaît, on peut choisir ici un des titres proposés en cliquant dessus directement.



On peut voir que "Monsieur" a été entré directement dans le champ "Titre".

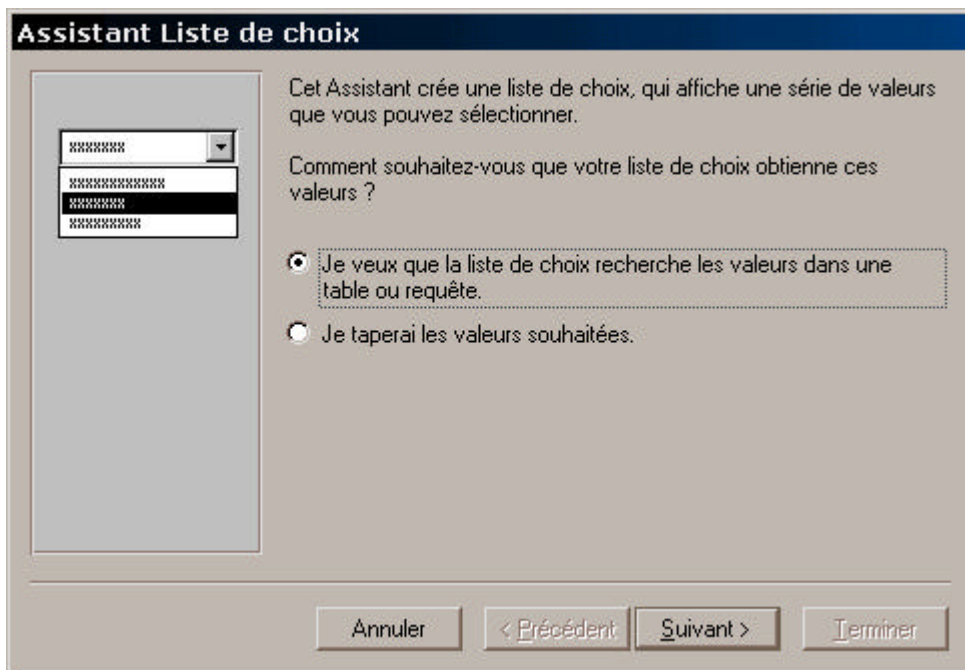
On va ainsi créer plusieurs clients :



Une fois ces clients saisis, nous allons aller modifier le type de données du champ "N° Client" de la table commande pour y mettre une liste de choix.

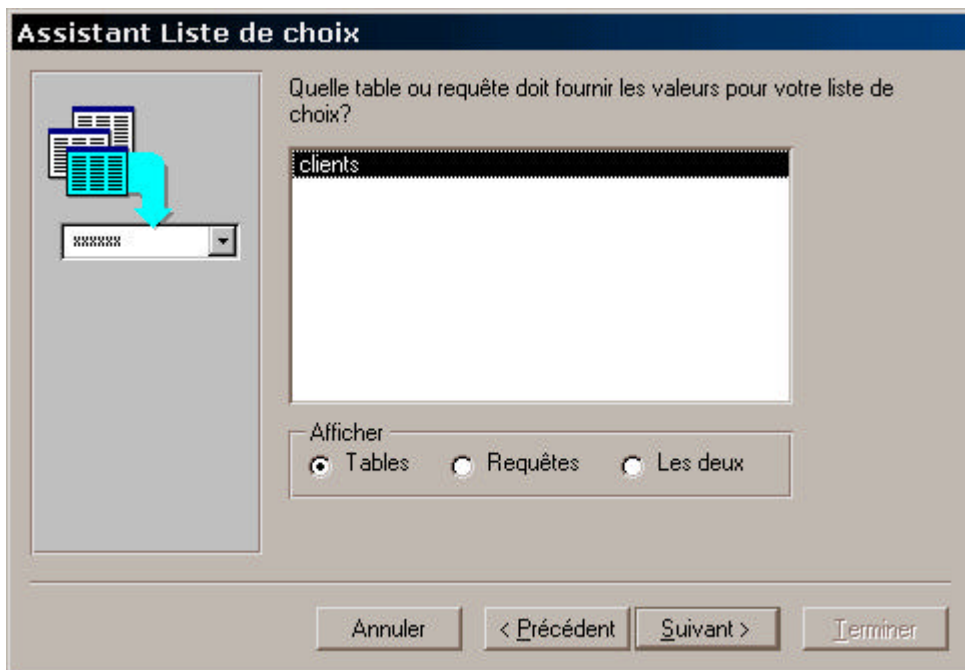
Pourquoi ? Une commande contient le numéro du client qui l'a passé, lorsque l'on saisi une commande et qu'on a que le nom du client, il faut se souvenir de son numéro pour le taper, la liste de choix va ici proposer la liste de tous les clients se trouvant dans la table "Clients", il suffira de cliquer sur le nom de l'un d'entre eux pour que son numéro soit automatiquement entré dans le champ "N° Client" de la table "Commande".

Voici comment faire :

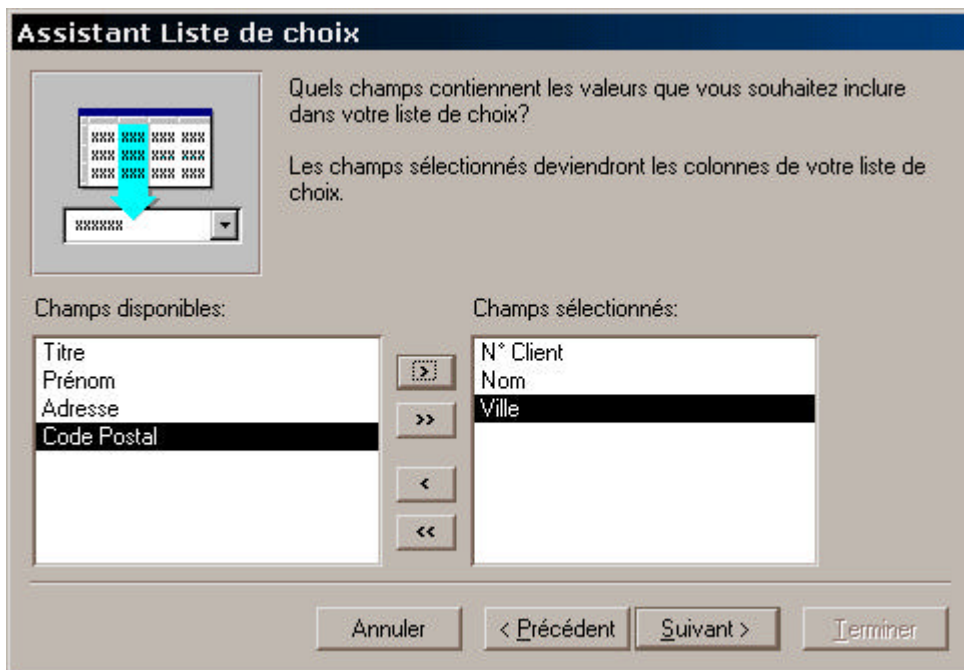


On change le type de données du champ "N° Client" dans la table "Commande" et on choisit "**Liste de Choix**". Cette fois-ci, nous n'allons pas saisir tous les éléments de la liste, ça n'aurait pas de sens, il faudrait remodifier cette liste à chaque fois qu'on ajoute un nouveau client. Nous allons dire à Access de construire sa liste de choix en allant chercher les éléments de cette liste dans une autre table, pour ça on clique juste sur "**Suivant**".

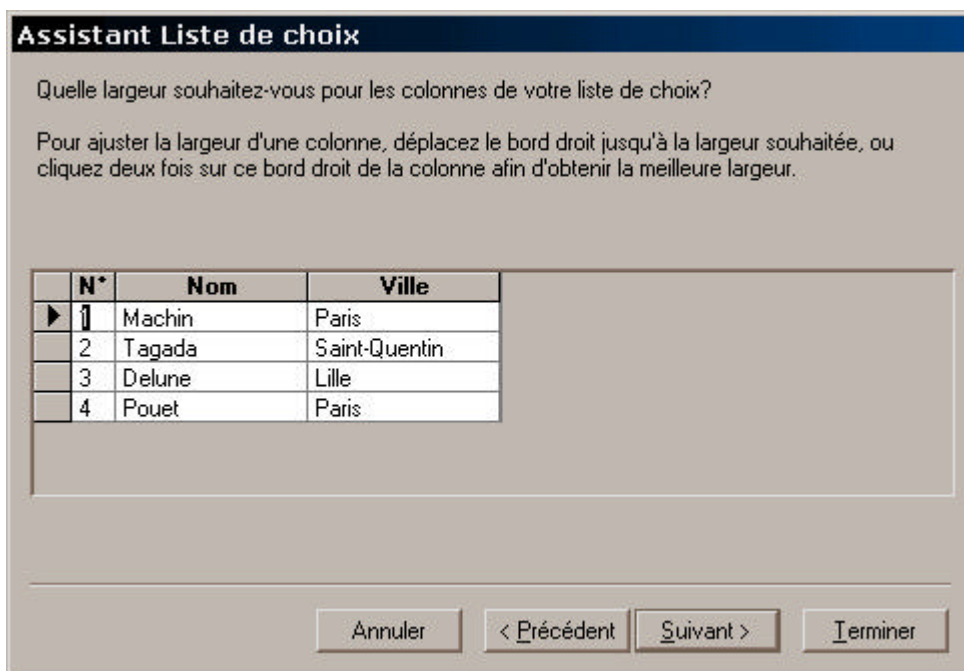
Access nous affiche les autres tables de la base :



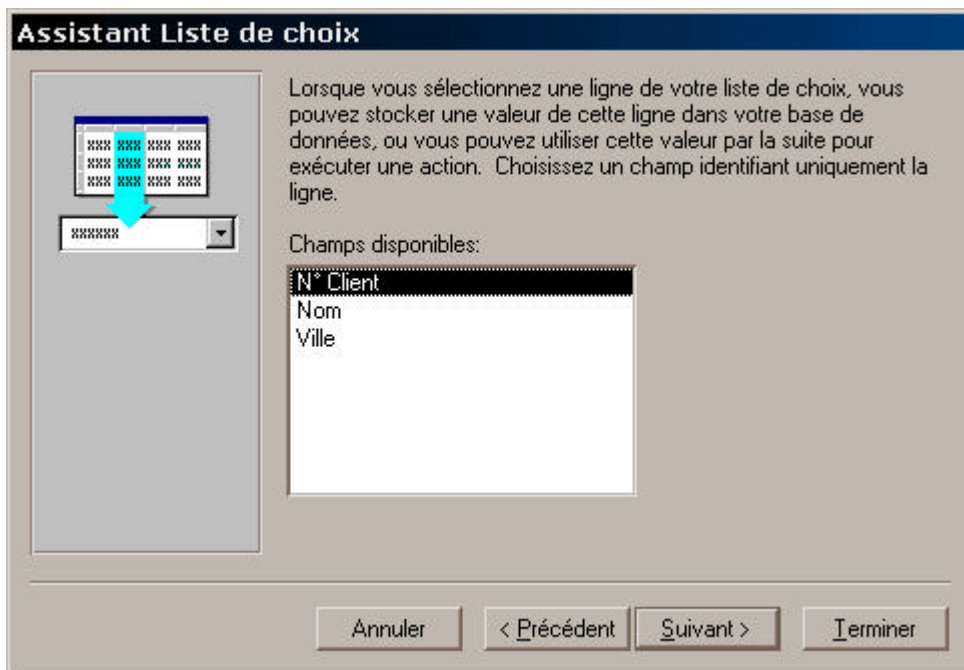
On choisit "Clients" (!) et on clique sur **Suivant**



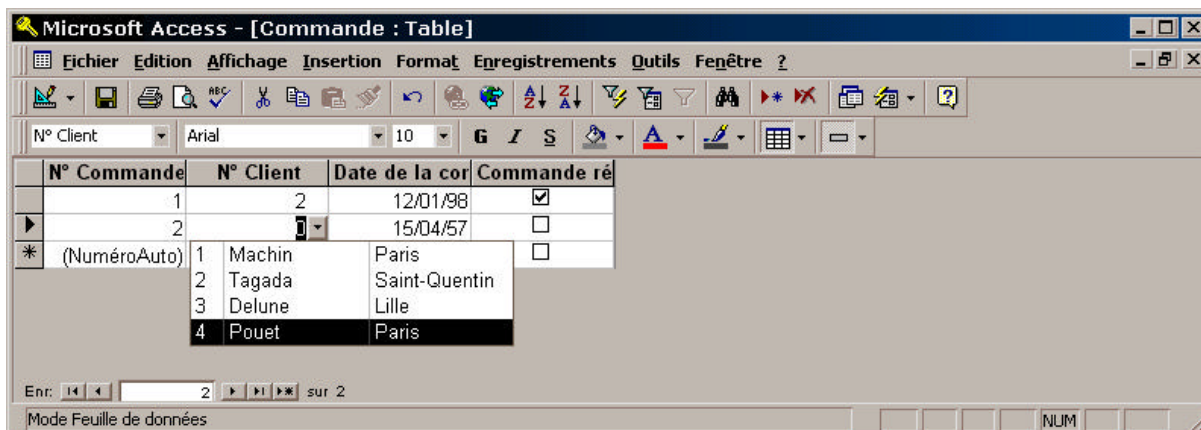
Access nous demande ici quels vont être les champs de cette table qui vont être affichés dans la liste, nous n'avons besoin que du numéro du client, de son nom et de sa ville (au cas où il y aurait deux clients homonymes). On sélectionne à gauche les champs que l'on désire voir apparaître dans la liste, et on clique sur ">" pour les ajouter, sur "<" pour les retirer, ">>" et "<<" ajoutent et retirent tous les champs.



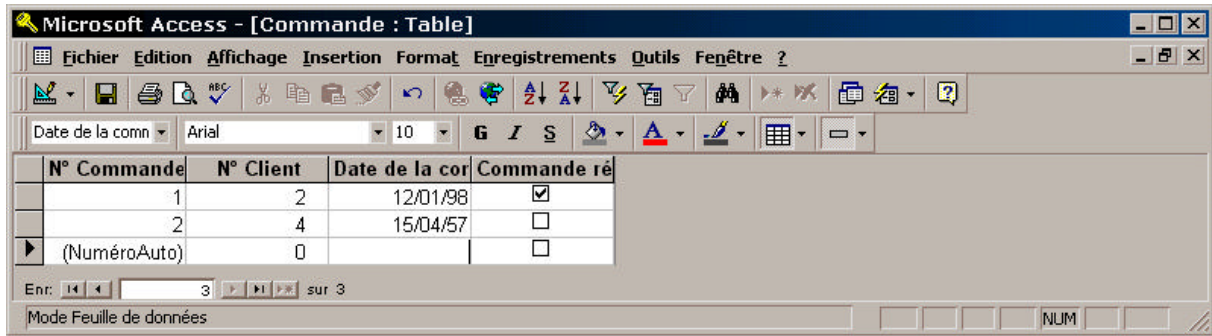
Access affiche un exemple du contenu de la liste. Vous pouvez ici dimensionner la taille des colonnes.



Nous avons choisi d'afficher trois champs de la table "Clients" dans notre liste de choix, Access nous demande lequel de ces trois champ va servir à initialiser le champ "N° Client" de la table "Commande", il est clair que nous voulons que le champ "N° Client" de la table "Commande" soit initialisé avec le n° de client du client que nous aurons sélectionné dans la liste, on choisit donc "n° Client" et on clique sur Suivant, on nomme ensuite la liste, et on peut commencer à saisir des commandes :



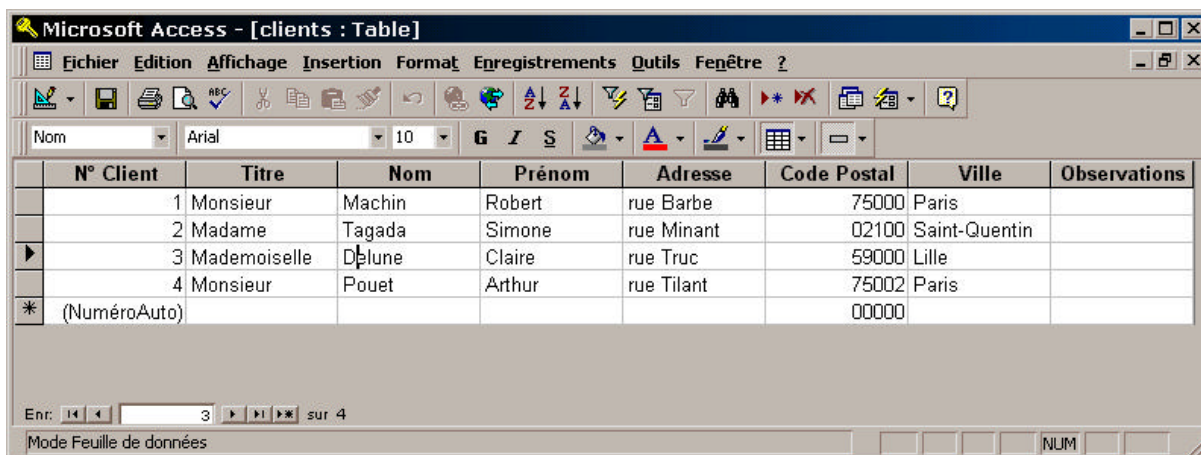
Que se passe-t-il lorsqu'on se positionne sur le n° de client ? Access va lire toute la table "Clients", et créer une liste de choix à partir de cette table. Cette liste contient les champs que nous lui avons indiqué. Il suffit de sélectionner un client dans la liste, et comme nous lui avons dit d'initialiser le champ N° Client de la table "Commande" avec la première colonne de la liste de choix, son numéro sera automatiquement copié.



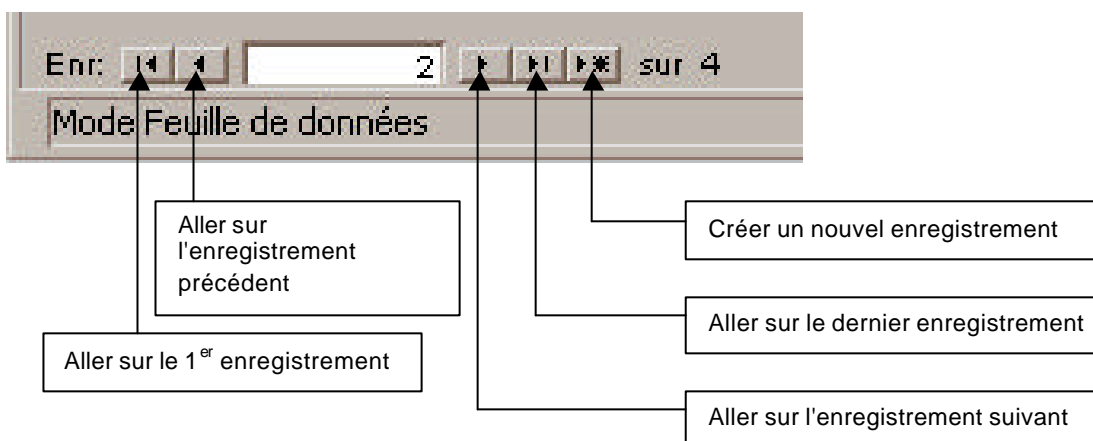
C'est beau non ?

3.6 La saisie des enregistrements

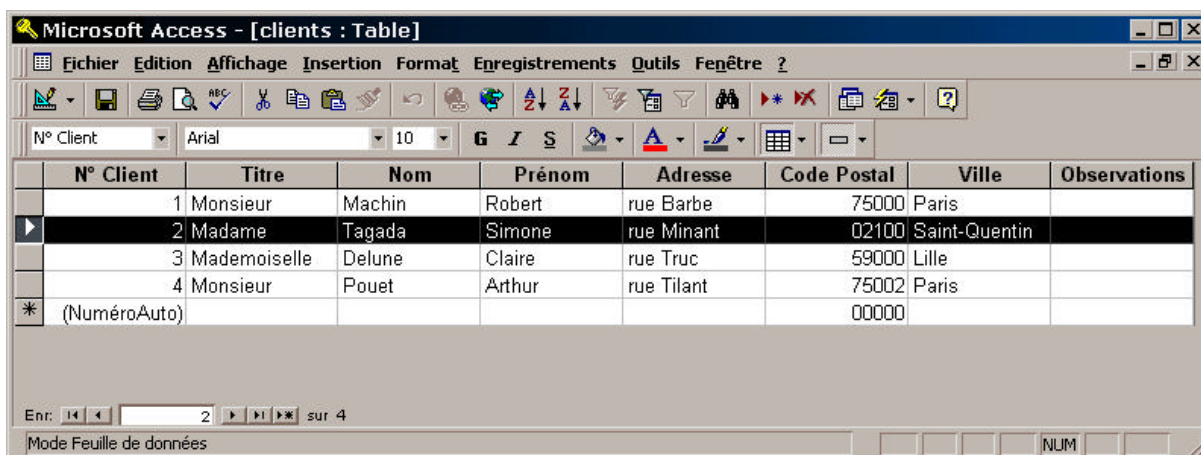
Jusqu'à présent, nous avons saisi rapidement des enregistrements dans les tables, attardons nous sur la saisie et les recherches / filtres .



Pour se déplacer parmi les enregistrements de la table, on utilise les icônes fléchés en bas à gauche de la fenêtre de saisie:



Pour supprimer un enregistrement, on clique à gauche de la ligne



et on appuie sur la touche "Suppr", Access vous demande alors une confirmation.

Deux petits trucs avant d'en finir avec la saisie de données dans les tables :

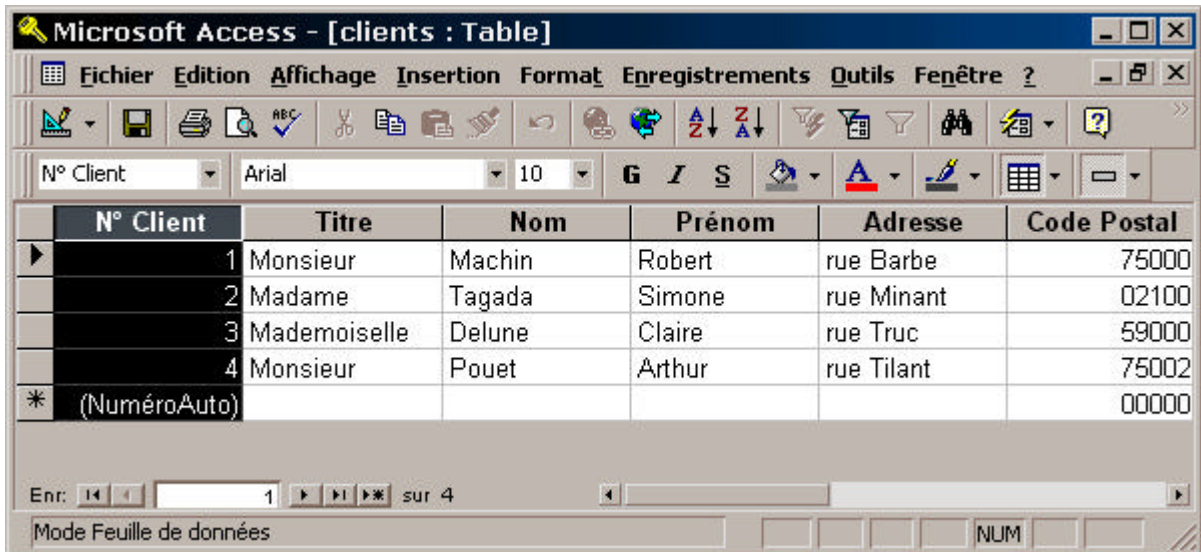
On peut cacher une colonne, pour ça, on va dans le menu "Format", et on clique sur l'option "Afficher les colonnes..."



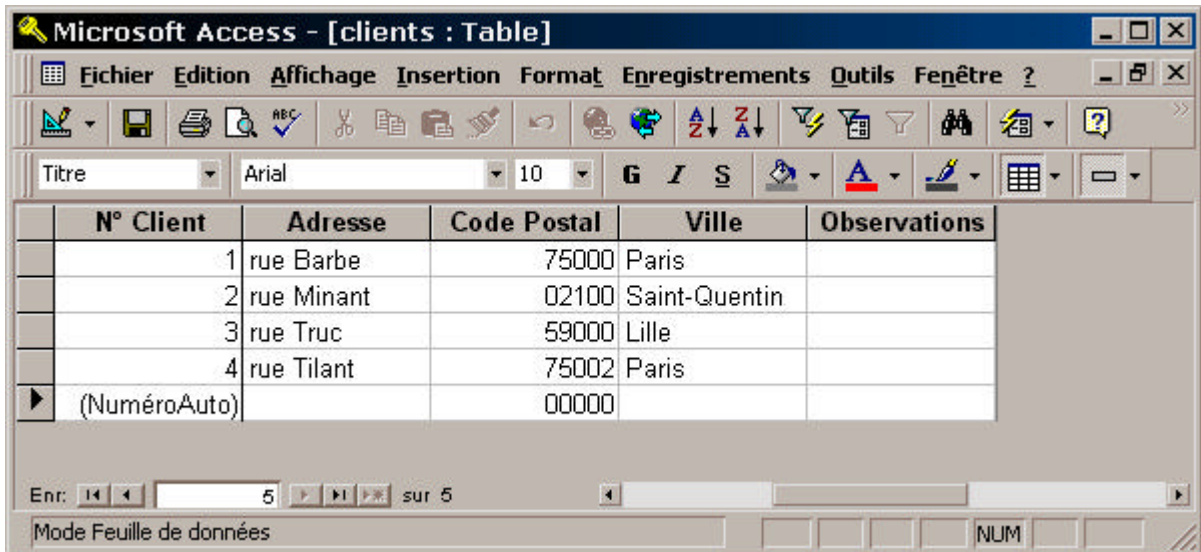
La liste des champs (colonnes) de la table s'affiche, vous sélectionnez celles que vous voulez voir apparaître, celles que vous n'avez pas sélectionné disparaîtront de l'affichage quand vous cliquerez sur "Fermer".

Attention, ça les fait juste disparaître de l'affichage pour la saisie, en aucun cas, le champ en question va être supprimé de la structure de la table.

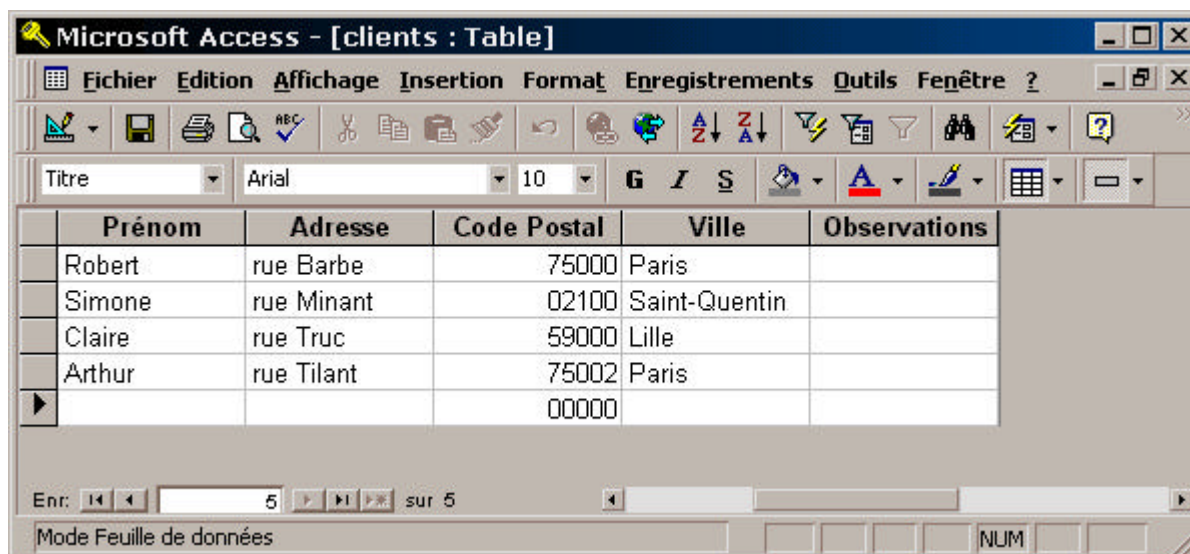
Autre truc interessant, on peut "figer" une colonne, à quoi ça sert ? Si pendant la saisie nous voulons avoir en permanence le numéro client , nous allons sélectionner la colonne en question en cliquant sur son titre :



Une fois la colonne sélectionnée, on va dans le menu "Format" et on clique sur "Figer les colonnes".
 Que se passe-t-il maintenant ? Quand on va déplacer l'ascenseur en bas pour modifier un des champs de l'enregistrement, la colonne "N° Client" sera toujours visible :



Si la colonne n'était pas figée, on aurait eu :



	Prénom	Adresse	Code Postal	Ville	Observations
	Robert	rue Barbe	75000	Paris	
	Simone	rue Minant	02100	Saint-Quentin	
	Claire	rue Truc	59000	Lille	
	Arthur	rue Tilant	75002	Paris	
			00000		

Enr: 5 sur 5
Mode Feuille de données

La colonne "N° Client" a bougé quand on a déplacé l'ascenseur, lorsqu'elle était figée, elle était bloquée.
Pour autoriser le déplacement de toutes les colonnes et annuler l'option "**Figer les colonnes**", dans le menu "**Format**", cliquez sur l'option "**Libérer les colonnes**".

3.7 Tri parmi les enregistrements

Pour trier la table sur un champ, on se positionne sur le champ en question (peu importe l'enregistrement) et on utilise un des deux icônes de la barre d'outils :



Celui de gauche permet le tri alphabétique croissant et celui de droite le tri alphabétique décroissant.

Exemple :

N° Client	Titre	Nom	Prénom	Adresse	Code Postal
3	Mademoiselle	Delune	Claire	rue Truc	59000
1	Monsieur	Machin	Robert	rue Barbe	75000
4	Monsieur	Pouet	Arthur	rue Tilant	75002
2	Madame	Tagada	Simone	rue Minant	02100
(NuméroAuto)					00000

Ceci est l'état de notre table après l'avoir trié de façon croissante sur le champ "Nom"

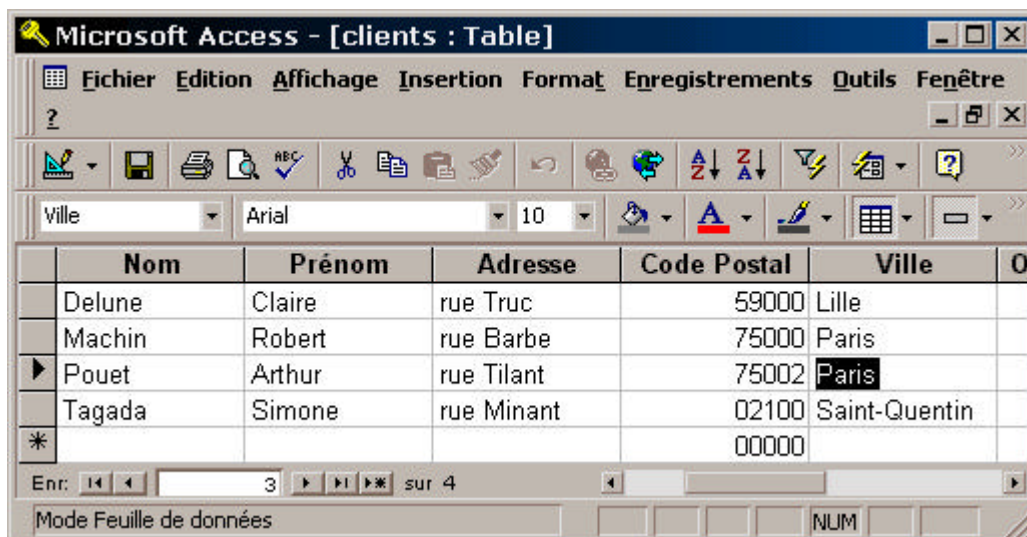
3.8 Filtrer des enregistrements

Les filtres permettent de limiter, de façon temporaire, les enregistrements affichés dans la table. On peut filtrer selon deux méthodes :

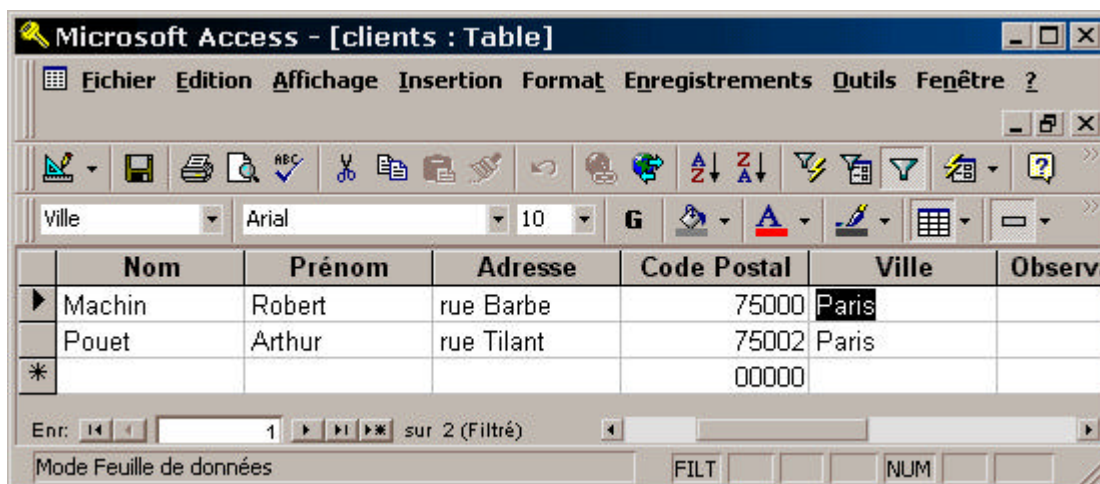
Le filtre sur un seul critère :

Par exemple, on ne veut afficher que les clients qui habitent Paris.

Pour cela, on se positionne sur l'enregistrement d'un client habitant Paris, et on sélectionne le mot "Paris" dans le champ "Ville":



Puis on clique sur l'icône du filtre dans la boîte à outils :



Ne s'affichent à présent que les client habitant à Paris. On peut aussi limiter le filtre à une partie d'un mot, par exemple, on aurait pu sélectionner seulement la lettre "P" du mot "Paris", ne se seraient alors affichés que les clients dont le nom de la ville commence par "P".

Pour faire réapparaître tous les enregistrements, on clique sur l'icône



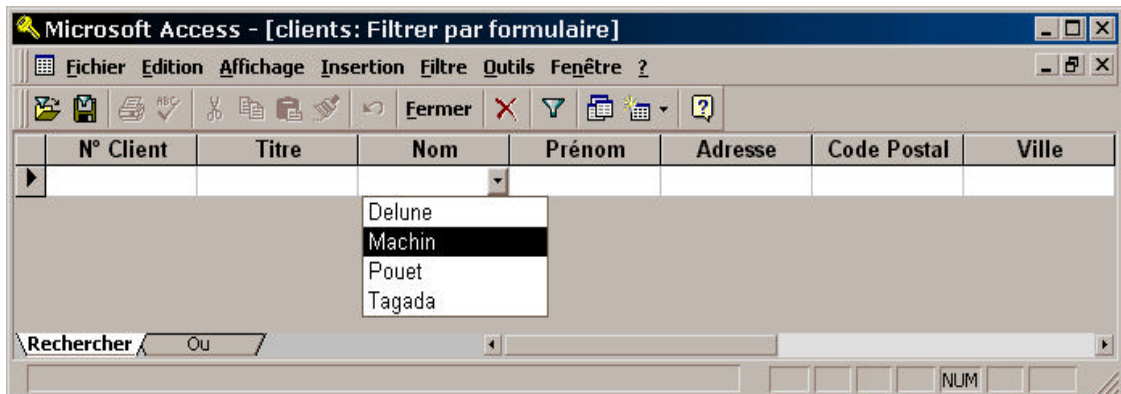
Filtre sur plusieurs critères

On clique sur l'icône :



Un enregistrement vide s'affiche.

Supposons que l'on veuille afficher les clients qui s'appellent "Machin" ou qui habitent à Saint-Quentin.

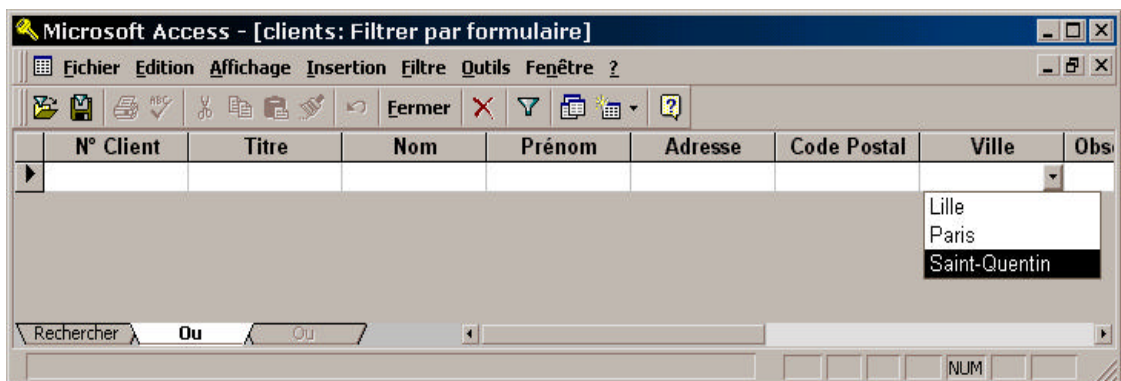


On va cliquer sur "Machin", si on voulait afficher les clients qui s'appellent "Machin" et dont le prénom est "Robert", on cliquerait sur "Robert" dans la colonne "Prénom".

Vous avez remarqué que, en bas, se trouvent deux onglets :



En cliquant sur l'onglet "Ou", vous obtenez un nouvel enregistrement vide dans lequel on va pouvoir saisir les critères de notre deuxième condition, le but étant d'avoir les clients qui s'appellent "Machin" OU qui habitent à Saint-Quentin, dans le deuxième onglet, on va sélectionner "Saint-Quentin" dans le champ ville :

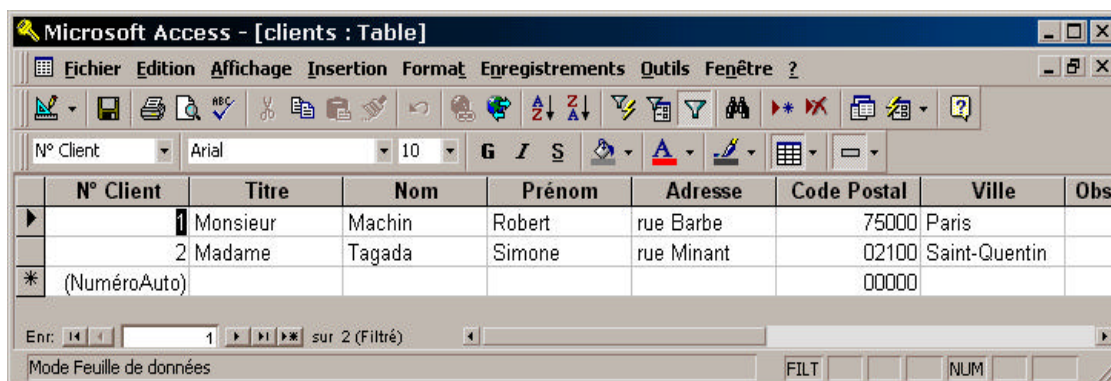


Au passage, notez qu'un troisième onglet est apparu au cas où on voudrait une troisième condition.

Une fois entrée la deuxième condition, on clique sur :



Et on obtient :

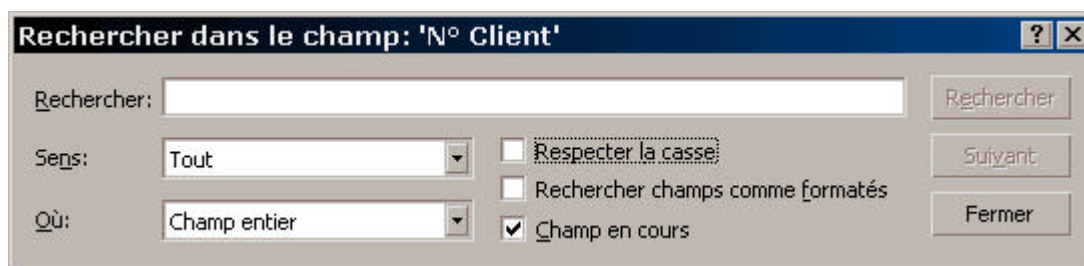


Soit la liste des clients dont le nom est soit "Machin" soit la ville est "Saint-Quentin".

Vous noterez au passage qu'on peut sauver son filtre multi critère sur disque en cliquant sur l'icône disquette dans la fenêtre de filtre.

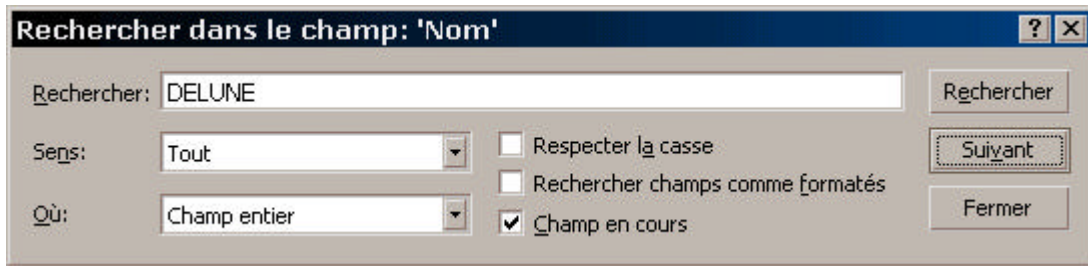
3.9 Rechercher des enregistrements

Pour faire une recherche, cliquez sur l'icône :

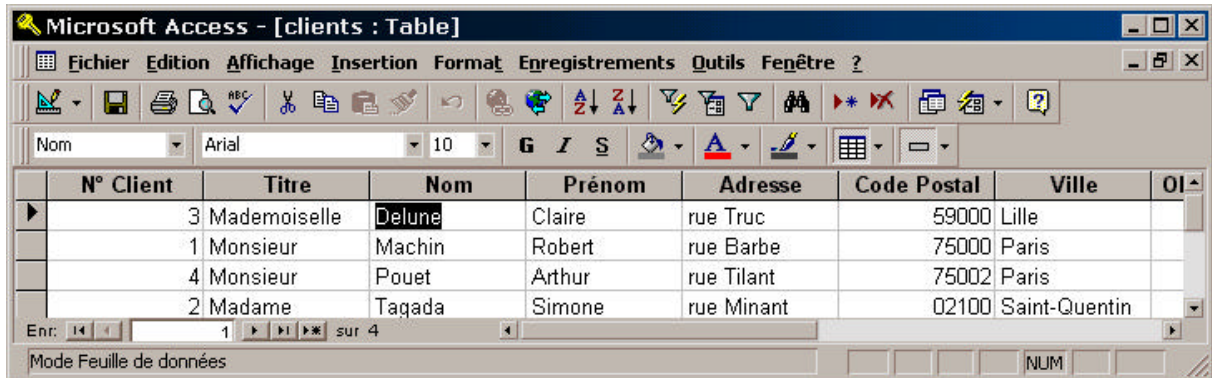


Rechercher	on indique la valeur à chercher
Sens	Indique dans quel sens va se faire la recherche : ?? <u>Tout</u> : dans toute la table ?? <u>Bas</u> : à partir de l'enregistrement courant jusque la fin de la table ?? <u>Haut</u> : à partir du début de la table jusque l'enregistrement courant.
Où	Sur quoi va se faire la recherche ?? <u>Champ entier</u> : La valeur cherchée doit être le contenu exact du champ ?? <u>Début de champ</u> : La valeur cherchée est le début du champ ?? <u>N'importe où</u> : La valeur cherchée peut se trouver n'importe où dans le champ
Respecter la casse	Doit-on différencier les majuscules des minuscules ?
Rechercher les champs comme Formatés :	Activez cette option si vous souhaitez qu'Access retrouve les données telles qu'elles sont affichées dans la table (en utilisant la propriété "Format" du champ) ou telles qu'elles sont stockées (c'est-à-dire telles qu'elles ont été saisies). Par exemple : pour le code postal, on a défini un format d'affichage : 00000 pour que si le code postal commence par 0 (comme 02100), il soit affiché correctement et pas sous la forme 2100. Si on active l'option "Rechercher champs comme formatés", on pourra rechercher le code postal 02100, sinon, il faudra chercher le code postal 2100.
Champ en cours	Indique si la recherche doit s'effectuer uniquement dans le champ en cours ou dans tous les champs de la table

Par exemple, pour rechercher le client "Delune" :

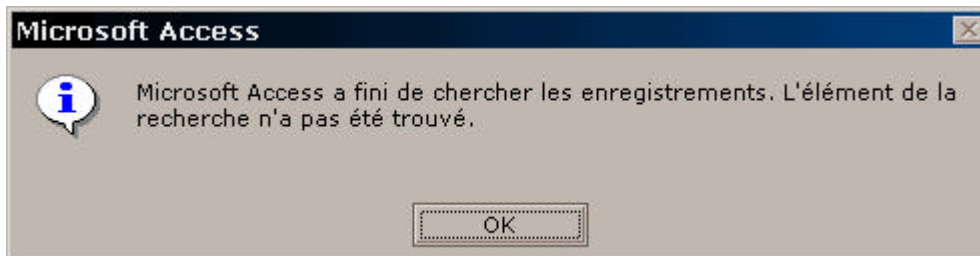


On clique sur "Rechercher"



Access se positionne sur le premier enregistrement dont le contenu du champ "Nom" est "Delune"

Pour trouver le suivant, on clique sur "Suivant"



En l'occurrence ici, il n'y en a pas d'autres, Access renvoie ce message.

On peut faire des recherches approximatives, pour cela on utilise des caractères génériques :

Caractère	Signification
*	Remplace une chaîne de valeur quelconque
?	Remplace un seul caractère
#	Remplace un chiffre unique
[]	Recherche des caractères parmi plusieurs
!	Exclu certains caractères de la recherche

Exemple :

La chaîne	Permet de retrouver	Mais ne retrouve pas
*rue	112 rue Boulevard Machin	
??rue	5 rue 8 rue	112 rue
##rue	10 rue	La rue
DUPON[TD]	DUPONT DUPOND	DUPONS
DUPON![TD]	DUPONS	DUPOND DUPONT

Prochain chapitre : les formulaires ...