

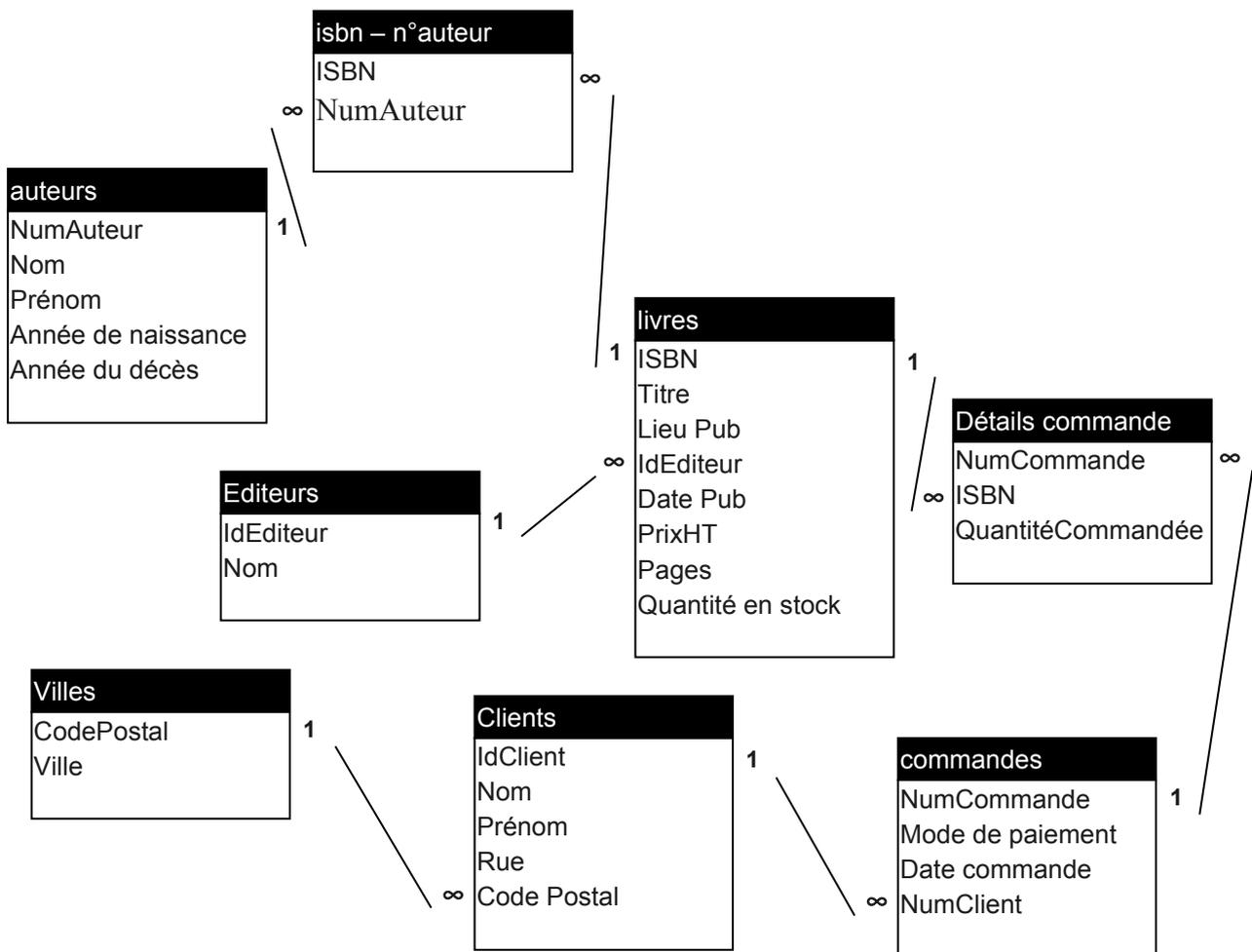
Gestion des bases de données (2^e partie)

Ce polycopié est destiné à présenter *Access* comme exemple de SGBD. Avant de commencer à construire une base de données sous *Access*, il est indispensable d’avoir créé le modèle logique des données (MLD) expliqué dans le polycopié précédent.

L’exemple qui servira de support de cours à la présentation d’*Access* est le dernier exercice d’application du précédent polycopié (création du MCD et du MLD) :

Une librairie veut informatiser sa gestion. Elle a des clients dans différentes villes qui lui passent des commandes. Une commande peut comporter plusieurs livres différents. Un livre est écrit par un ou plusieurs auteurs et est édité par un éditeur.

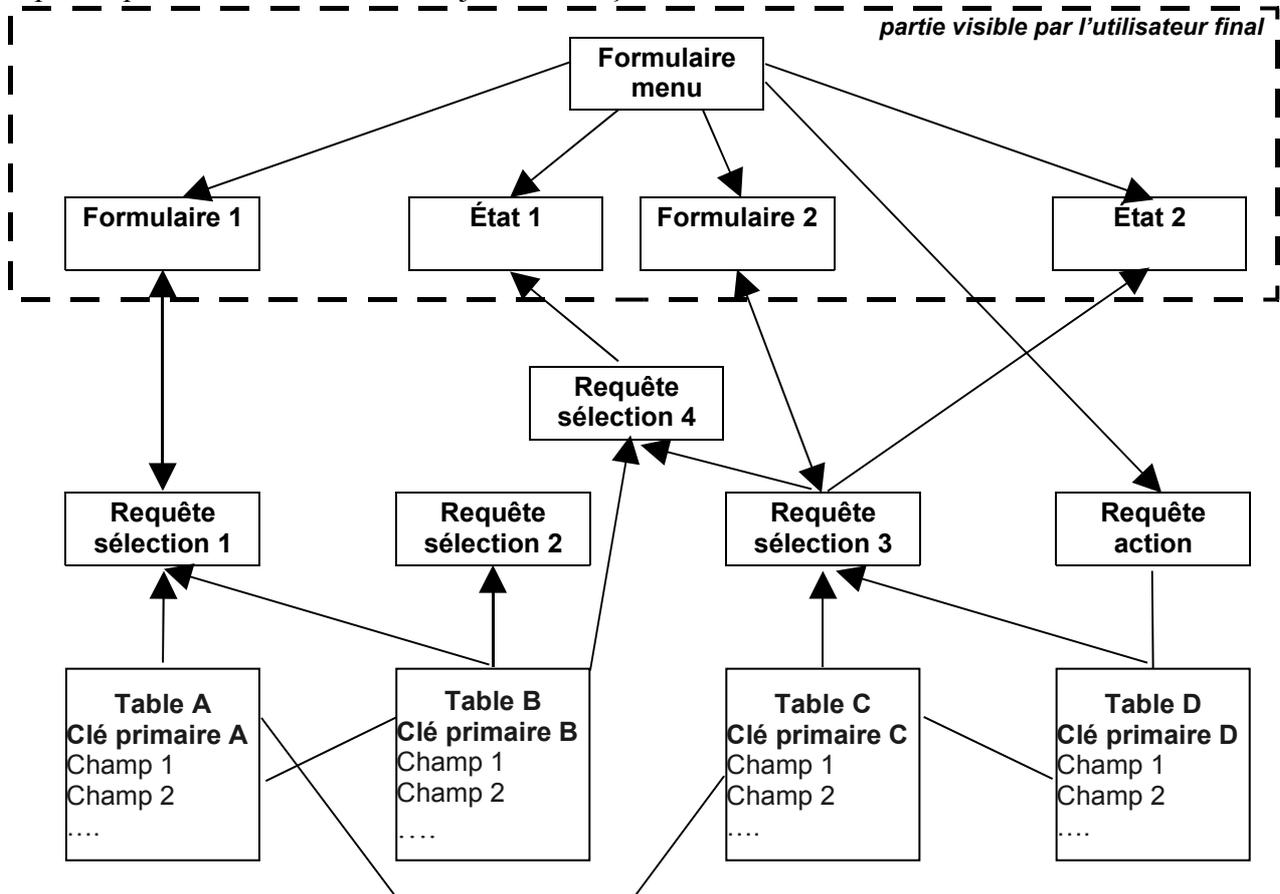
Pour mémoire, le MLD que nous avons construit est le suivant :



I. Présentation des principaux objets d'Access

Une base de données sous *Access* peut comprendre des tables (avec des relations entre ces tables), des requêtes, des formulaires, des états, des pages, des macros et des modules. Dans ce cours ne seront présentées que les tables et les requêtes. Les formulaires et les états (plus les menus qui sont des formulaires particuliers) seront présentés au semestre 2.

On peut représenter ces différents objets de la façon suivante :

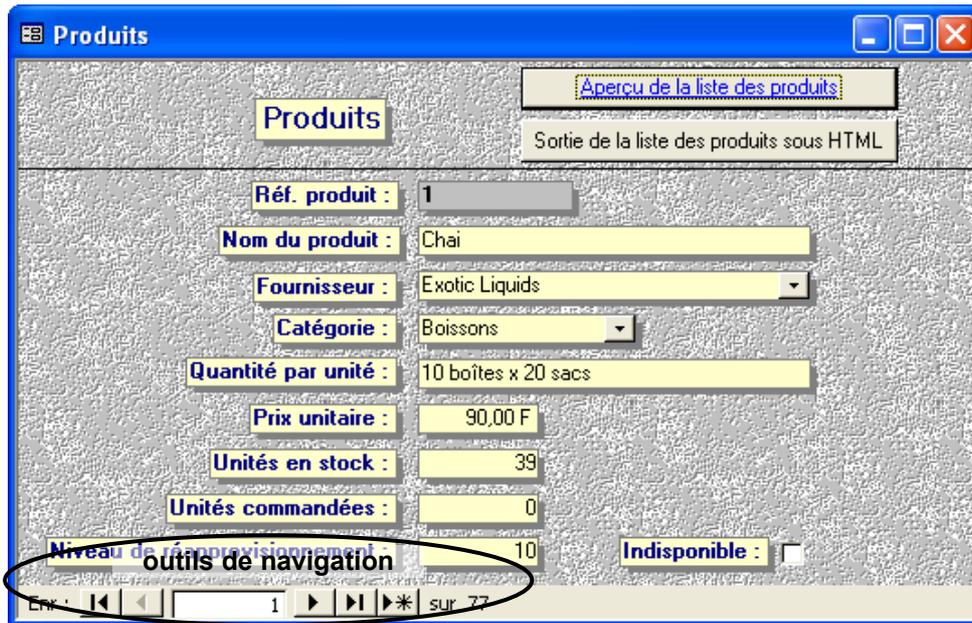


Un exemple de base de données livré avec *Access* va nous permettre de mieux comprendre le rôle de ces différents éléments. Lancez *Access* et appelez la commande **Exemples de bases de données** ▶ **Exemple de base de données les comptoirs** dans le menu ? (*Aide*).

Après un formulaire d'introduction (qui peut être désactivé), vous arrivez au menu suivant (un menu est un type de formulaire particulier) :



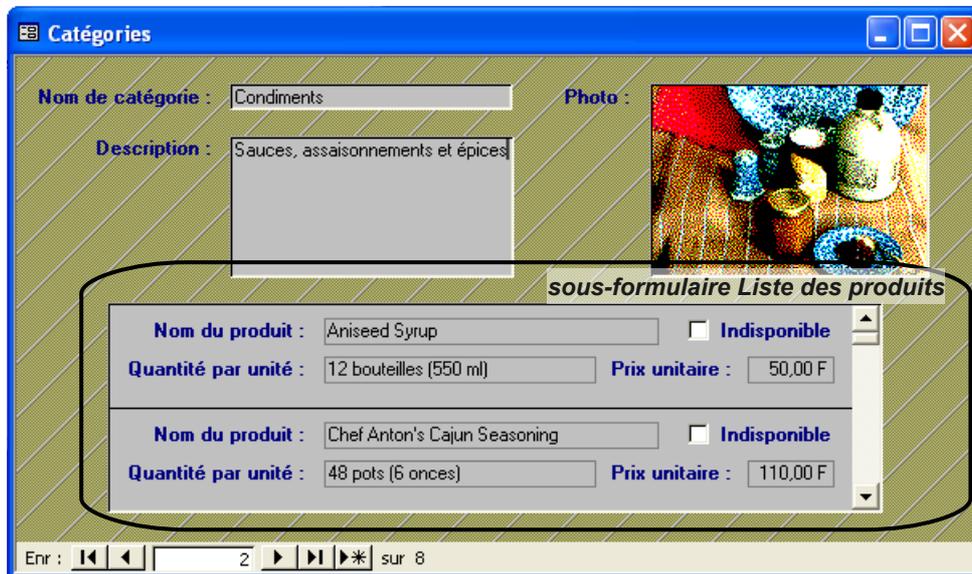
1) Cliquez sur le bouton **Produits**. Le formulaire suivant apparaît :



En bas de cette fenêtre, les flèches et permettent de parcourir la liste des produits (utilisez pour accéder directement au dernier et pour revenir au tout premier). On peut aussi accéder à un enregistrement en entrant son numéro dans la case. Le bouton permet de créer un nouvel enregistrement (référencer un nouveau produit).

Le bouton **Aperçu de la liste des produits** permet d'afficher l'état *Liste alphabétique des produits* : un catalogue des produits référencés par l'entreprise *Les comptoirs*.

2) Revenez au menu et cliquez sur le bouton **Catégories**. Ce bouton permet d'accéder au formulaire *Catégories*, qui contient le sous-formulaire *Liste des produits* :



Parcourez la liste des catégories et, pour chacune, visualisez la liste des produits correspondants.

3) Revenez au menu et cliquez sur le bouton **Fournisseurs**. Ce bouton permet d'accéder au formulaire *Fournisseurs*. Utilisez ce formulaire pour parcourir la liste des fournisseurs. En bas du formulaire, le bouton **Visualisation produits** permet d'afficher la liste des produits du fournisseur ; le bouton **Ajouter produits** sert à afficher le formulaire *Produits* de façon à ajouter un nouveau produit lié à ce fournisseur.

Utilisez ce formulaire pour ajouter un nouveau produit (Jeanlain – catégorie : *Boissons*) vendu par la société *Aux joyeux ecclésiastiques*. Complétez la fiche à votre gré.

Vérifier dans *Fournisseurs* et *Catégories* que le nouveau produit a été pris en compte

4) Revenez au menu et cliquez sur le bouton **Commandes** :

Produit	Prix unitaire	Quantité	Discount	Total
Original Frankfurter grüne Soße	65,00 F	2	20%	104,00 F
Raclette Courdavault	275,00 F	15	0%	4 125,00 F

Ajoutez une nouvelle commande (clic sur le bouton **►***) passée par le client *Blondel père et fils*, suivie par *Nancy Davolio* et livrée par *United*.

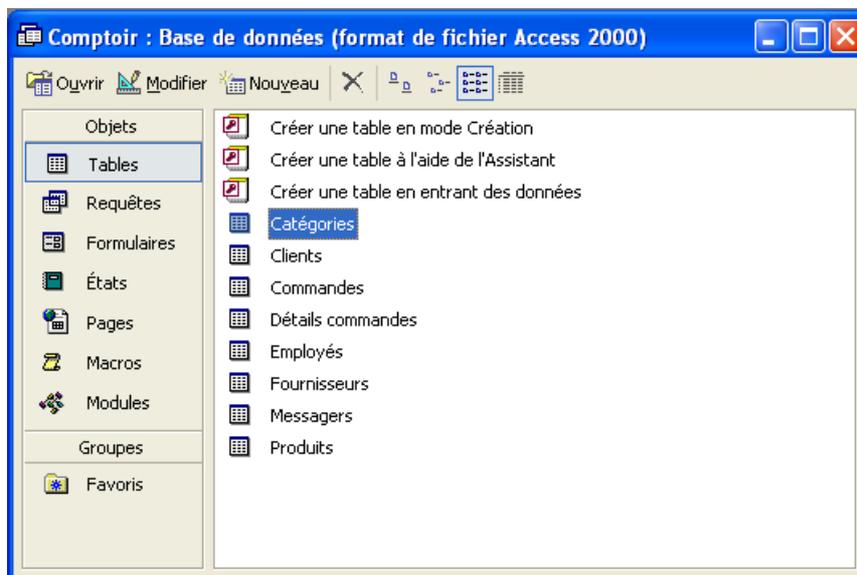
Essayez d'y mettre deux fois le même produit (*Jeanlain* par exemple). Un message d'erreur apparaît. Pourquoi ?

Finissez de saisir la commande mais ne cliquez pas sur **Imprimer facture**.

5) Revenez au menu et choisissez **Imprimer des rapports de vente**. Affichez les aperçus suivants :

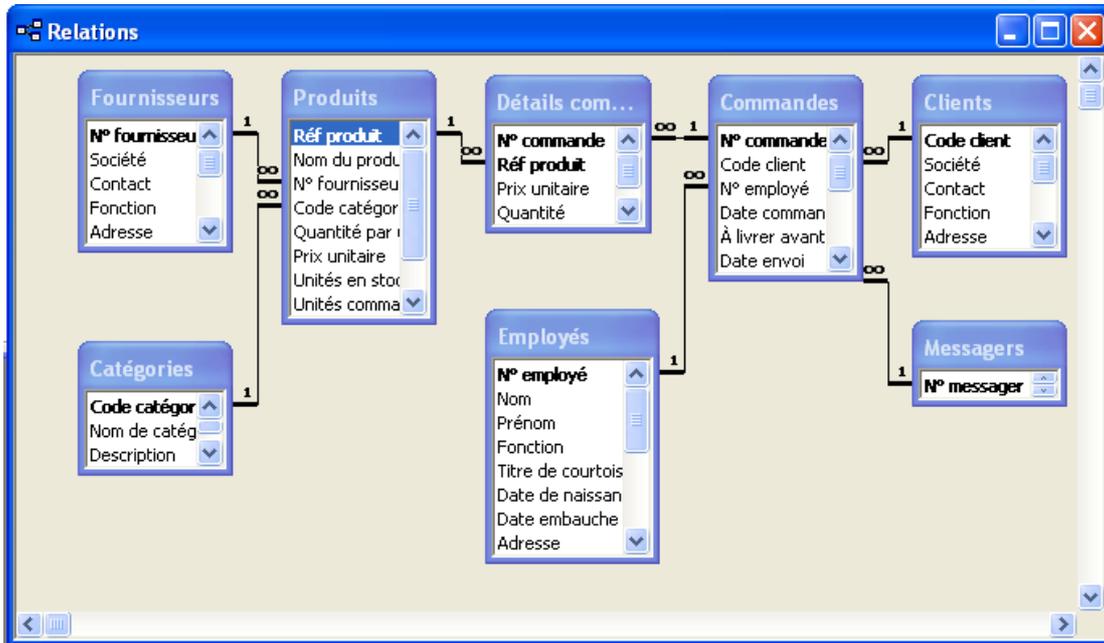
- ventes par représentant et par pays pour le premier trimestre 1997 ;
- totaux des ventes classés par montant ;
- statistiques de ventes pour les catégories *Desserts* et *Produits laitiers*.

Comme vous pouvez le constater l'utilisateur de la base de données n'accède via le menu qu'à des **formulaires** et des **états**. Fermez le menu (en cliquant sur le bouton **Afficher la fenêtre Base de données**) et examinez la structure des tables de cette base :



Pour afficher la liste des tables, cliquez sur le bouton correspondant à gauche de cette fenêtre. Vous pouvez afficher le contenu d'une table en double-cliquant sur son nom dans la liste.

La commande **Outils** ► **Relations** (ou le bouton  de la barre d'outils *Base de données*) vous permet de visualiser le modèle logique de données de l'application :



- 6) Examinez les tables *Produits*, *Commandes* et *Détails commandes* et vérifiez que les ajouts que vous avez introduits avec les différents formulaires sont présents dans les tables.

Avant de passer à la deuxième partie, fermez l'application « les comptoirs »...

II. Tables et relations

Nous allons maintenant créer les huit tables et les relations du modèle logique de données de l'application librairies. Pour commencer il faut créer l'application *Librairie* en appelant la commande **Fichier** ▶ **Nouvelle base de données...** Choisissez l'option *Créer base de données vide*. Vous indiquerez le nom (*Librairie_votre nom_votre prénom.mdb*) du fichier et l'emplacement où vous voulez l'enregistrer avant de cliquer sur le bouton **Créer**.

Attention ! A la différence des autres applications *Microsoft*, il ne vous sera pas possible de réenregistrer la base de données directement sous un autre nom et/ou à un autre emplacement par la commande **Fichier** ▶ **Enregistrer sous**.

Le format de fichier par défaut est le format *Access 2000* qui est compatible aussi bien avec *Microsoft Access 2000* qu'avec *Access 2002* (XP). Vous pouvez modifier ce format par défaut (fermez toutes les bases de données et appelez la commande **Outils** ▶ **Options** puis cliquez sur l'onglet *Avancé* ; dans la liste *Format de fichier par défaut*, sélectionnez *Access 2002*). Notez cependant qu'un fichier au format *Access 2002* ne peut être ouvert que dans *Access 2002*.

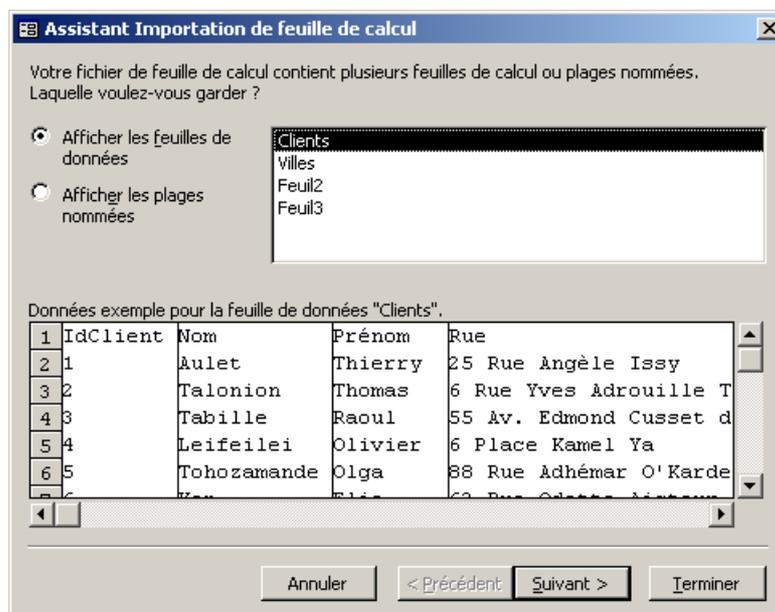
Il est possible de créer les tables soit en définissant la structure puis en saisissant des données, soit en important directement des données provenant de listes de données enregistrées dans des feuilles *Excel*.

1. Création de tables à partir de listes de données Excel

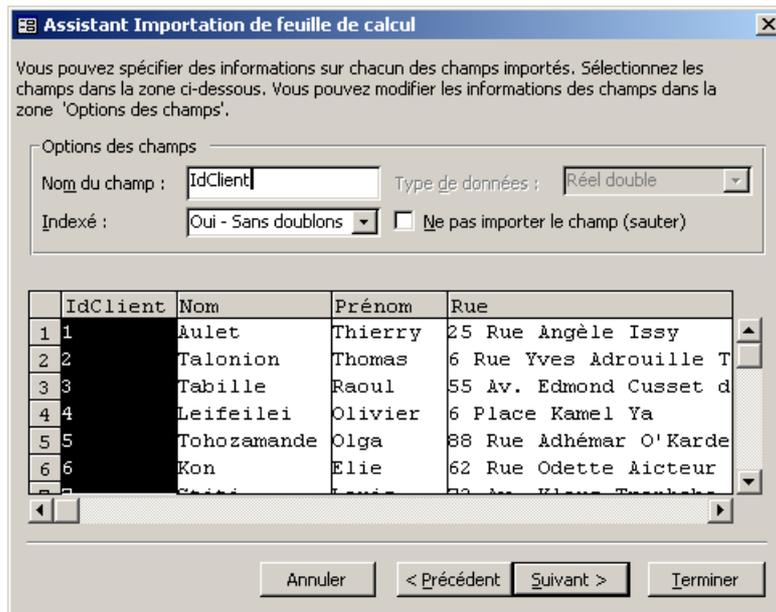
Afin de limiter des saisies fastidieuses nous allons utiliser la deuxième méthode pour créer les six premières tables de l'application. Les données se trouvent dans les deux classeurs *Excel* nommés *Bases Clients.xls* et *Bases Livres.xls* qui se trouvent sur le serveur dans le dossier *Sujets/master1 ANI+TCI* et que vous copierez sur votre ordinateur.

7) Importez la liste des clients depuis *Excel* :

- Appelez la commande **Fichier** ▶ **Données externes** ▶ **Importer...** N'oubliez pas de spécifier *Microsoft Excel* comme *Type de fichier* puis allez à l'emplacement où vous avez copié le fichier *Base Clients.xls*. Sélectionnez-le et cliquez sur **Importer** :



- Il est possible d'importer une feuille (*Afficher les feuilles de données*) ou une plage de cellules nommée (*Afficher les plages nommées*). Choisissez la première option et sélectionnez la feuille *Clients* puis cliquez sur **Suivant**.
- Activez l'option *Première ligne contient les en-têtes de colonnes* puis cliquez sur **Suivant**.
- Choisissez l'option *Dans une nouvelle table* puis cliquez sur **Suivant** :



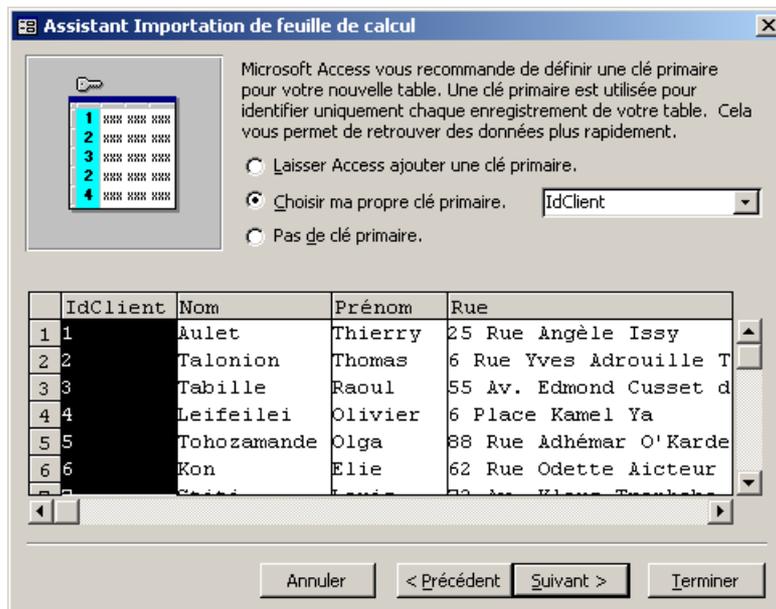
- Vous pouvez modifier certaines propriétés des champs de la table (nom, type de données), préciser ceux qui doivent être **indexés** et éventuellement en exclure certains. Cliquez sur **Suivant**.

Les index

Un index permet à *Microsoft Access* de rechercher et de trier plus rapidement des enregistrements. On peut créer des index basés sur un seul ou sur plusieurs champs (jusqu'à 10). Il est conseillé d'indexer les champs vérifiant les conditions suivantes :

- les données sont de type *Texte*, *Numérique*, *Monétaire* ou *Date/Heure* ;
- vous avez l'intention de rechercher des valeurs stockées dans le champ ;
- vous souhaitez trier les valeurs du champ ;
- le champ est destiné à prendre de nombreuses valeurs différentes.

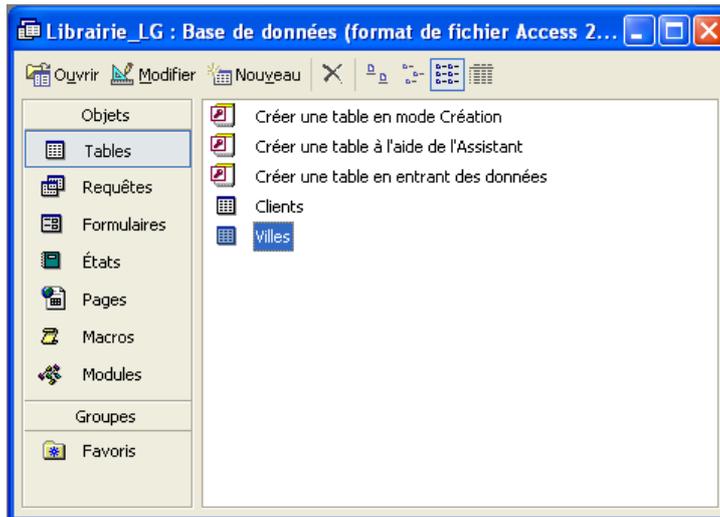
La clé primaire d'une table est indexée automatiquement.



- Précisez que la clé primaire est le champ *IdClient* puis cliquez sur **Suivant**.
- À la dernière étape de l'assistant, donnez un nom à la table (vous accepterez le nom proposé : *Clients*).

8) Importez de façon identique la liste des villes qui se trouve dans le fichier *Base Clients.xls*.

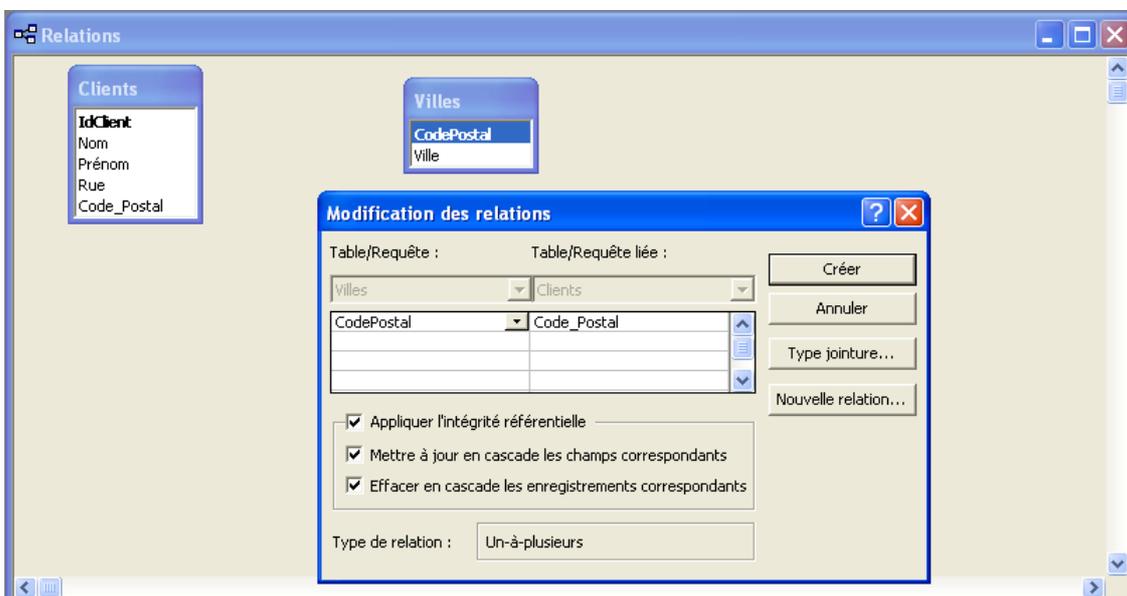
- 9) Vous disposez maintenant de deux tables dont vous pouvez examiner la structure avec le bouton  **Modifier** (affichage en mode *Création*) et dont vous pouvez visualiser, modifier et / ou compléter les données avec le bouton  **Ouvrir** (affichage en mode *Feuille de données*) :



- 10) Créez la relation entre les deux tables : dans le menu **Outils**, choisissez la commande **Relations** (ou cliquez sur le bouton  de la barre d'outils *Base de données*) :



Sélectionnez les deux tables *Clients* et *Villes* (touche *Ctrl* ou *Maj*) et cliquez sur **Ajouter**. Pour créer la relation entre les tables cliquez sur le champ *Code Postal* de la table *Villes* et faites glisser jusqu'à *CodePostal* de la table *Clients* :



Cochez les trois cases puis cliquez sur le bouton **Créer**.

L'intégrité référentielle

L'intégrité référentielle est un système de règles que *Microsoft Access* utilise pour garantir que les relations entre les enregistrements dans les tables liées sont valides et que vous ne supprimez ou ne modifiez pas accidentellement des données liées. Pour mettre en œuvre l'intégrité référentielle, il faut que les trois conditions suivantes soient réunies :

- le champ lié de la table primaire est une clé primaire ou a un index unique ;
- les champs liés ont le même type de données (exceptions : un champ *NuméroAuto* peut être lié à un champ *Numérique* de type *Entier long*, un champ *NuméroAuto* de type *N° de réplication* peut être lié à un champ *Numérique* de type *N° de réplication*) ;
- les deux tables appartiennent à la même base de données *Access*.

Lorsque l'intégrité référentielle est appliquée, les règles suivantes s'appliquent :

- On ne peut pas entrer dans le champ clé étrangère une valeur qui n'existerait pas dans la clé primaire de la table primaire. Toutefois, on peut entrer une valeur *Null* dans la clé étrangère pour indiquer que les enregistrements ne sont pas liés. Par exemple, une commande ne peut pas être attribuée à un client inexistant, mais elle peut n'être affectée à personne en entrant une valeur de type *Null* dans le champ *Numclient*.
- On ne peut pas effacer un enregistrement de la table primaire si des enregistrements correspondants existent dans une table liée. Par exemple, on ne peut pas effacer une personne de la table *Clients* s'il existe des commandes rattachées à cette personne dans la table *Commande*.
- On ne peut pas modifier une valeur de la clé primaire dans la table primaire si cet enregistrement a des enregistrements liés. Par exemple, il est impossible de modifier l'*Id-Client* d'un client qui aurait déjà passé des commandes.

Mises à jour et suppressions en cascade

Lorsque vous appliquez l'intégrité référentielle à une relation, vous pouvez spécifier si vous voulez qu'*Access* mette à jour ou supprime en cascade automatiquement des enregistrements liés. Si vous activez ces options, les opérations de suppression et de mise à jour qui sont normalement interdites par les règles d'intégrité référentielle sont alors autorisées, *Access* effectuant les modifications nécessaires dans les tables liées afin de maintenir l'intégrité référentielle :

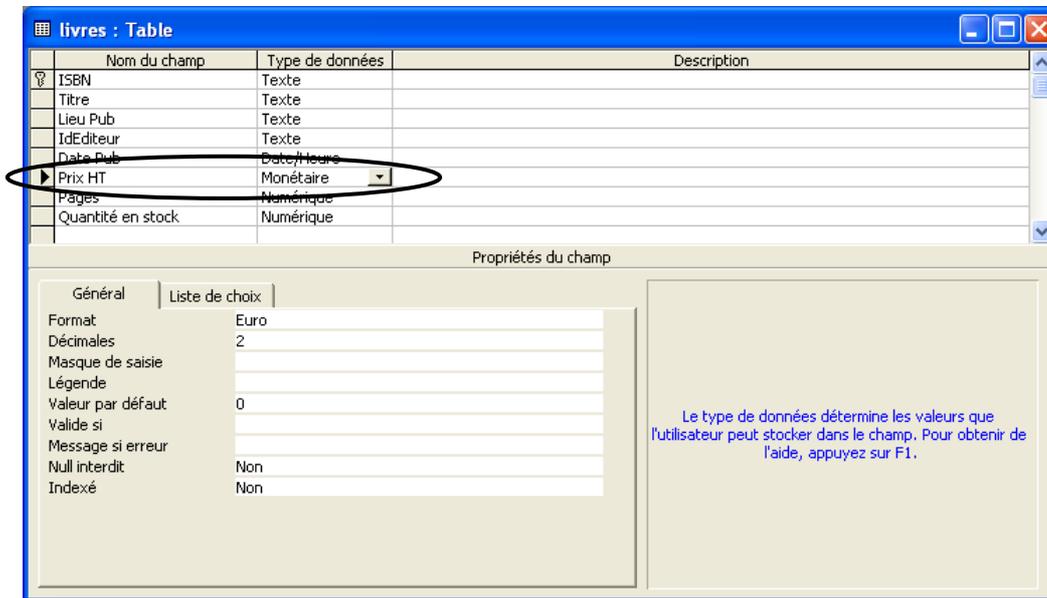
- *Mettre à jour en cascade les champs correspondants* : si vous modifiez la clé primaire d'un enregistrement de la table primaire (par exemple l'*IdClient* d'un client), *Access* met automatiquement à jour le champ correspondant dans tous les enregistrements liés (le champ *NumClient* de ses commandes) afin que la relation ne soit pas rompue.

Remarque : si la clé primaire dans la table source est un champ *NuméroAuto*, il est inutile d'activer cette option car on ne peut pas changer manuellement les valeurs de ce type de champ.

- *Effacer en cascade les enregistrements correspondants* : si vous supprimez un enregistrement de la table source, *Access* supprimera automatiquement tous les enregistrements de la table cible. Par exemple, si vous effacez un client de la table *Clients*, toutes ses commandes seront automatiquement supprimées de la table *Commandes* (ainsi que les enregistrements correspondants dans la table *Détails commandes*).

- 11) Ouvrez la table *Clients* et essayez d'ajouter un nouveau client dont le numéro est 190. Que se passe-t-il ? Pourquoi ?
- 12) Essayez d'ajouter un nouveau client dont le numéro est 206 et dont le code postal est 54000 (Nancy). Que se passe-t-il ? Pourquoi ? Comment régler ce problème ?

- 13) Importez les feuilles *livres*, *auteurs*, *ISBN N°auteur* (pas de clé primaire) et *editeurs* qui se trouvent dans le fichier *Base Livres.xls*.
- 14) Modifiez la table *ISBN N°auteur* en définissant les deux champs comme clé primaire.
- 15) Dans la table *livres*, changez le type de données du champ *PrixHT* de numérique à monétaire, format *Euro*, avec 2 décimales : sélectionnez la table et cliquez sur le bouton  Modifier :



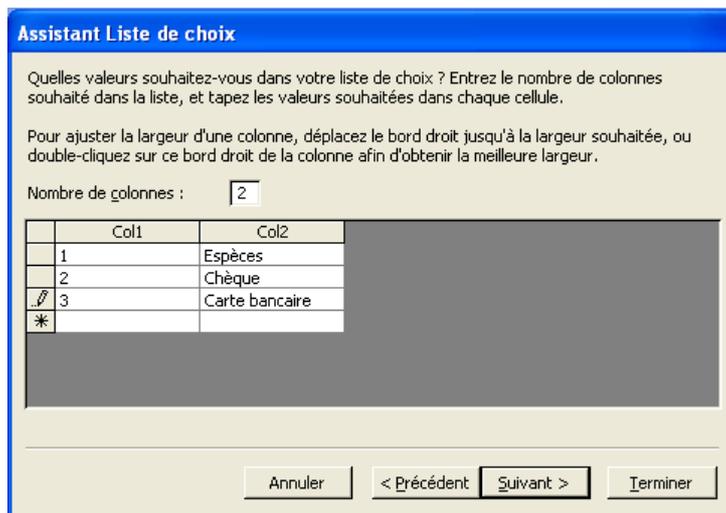
Cliquez dans la colonne *Types de données* à droite du nom du champ et sélectionnez le type voulu dans la liste déroulante. En bas de la fenêtre, à l'onglet *Général*, sélectionnez le format et le nombre de décimales.

- 16) Créez les relations entre les tables (en appliquant l'intégrité référentielle).

2. Création de tables ex-nihilo

Nous allons maintenant voir comment créer ex-nihilo une table dans *Access*, en nous appuyant sur l'exemple de la table *commandes*.

- 17) Cliquez sur le bouton  Nouveau, sélectionnez *Mode création* et créez la structure de la table *commandes* qui comprend les quatre champs suivants :
 - *NumCommande* : type de données *NuméroAuto*. Déclarez ce champ comme clé primaire (cliquez avec le bouton droit de la souris à gauche du nom du champ et choisissez **Clé primaire** dans le menu contextuel – la valeur de l'attribut *Indexé* devient *Oui sans doublons*).
 - *Mode de paiement* : choisissez comme type de données *Assistant liste de choix*. À la première étape de l'assistant, activez l'option *Je taperai les valeurs souhaitées* et cliquez sur **Suivant**. À l'étape suivante, choisissez 2 colonnes et saisissez les 3 lignes suivantes :



Cliquez sur **Suivant** et choisissez *Col1* comme champ à stocker. À la dernière étape, conservez *Mode de paiement* comme étiquette et validez en cliquant sur **Terminer**.
Observez le résultat dans l'onglet *Liste de choix*.

- *Date commande* : type de données *Date/Heure*, format *Date, abrégé*, valeur par défaut : *Date()* (c'est la fonction qui renvoie la date système).
- *NumClient* : type de données *Numérique*, taille *Entier long*.

18) Créez la relation avec la table *Clients*.

19) Saisissez les quelques enregistrements suivants :

Mode de paiement	Date commande	NumClient
2 (chèque)	date actuelle	11
1 (espèces)	date actuelle	23
3 (carte bleue)	22/10/2005	53

20) Essayez avec la commande suivante :

Mode de paiement	Date commande	NumClient
3 (carte bleue)	date actuelle	190

Que se passe-t-il ? Pourquoi ?

Il existe une autre façon de procéder, qui permet d'éviter de saisir une commande correspondant à un client inexistant :

- Supprimez tous les enregistrements de la table *commandes* (ouvrez-la en mode *Feuille de données*, sélectionnez tous les enregistrements – *Ctrl+A* – et appuyez sur *Suppr*) et la relation entre les tables *Clients* et *commandes* (affichez la fenêtre *Relations*, cliquez avec le bouton droit sur la relation entre les deux tables et choisissez **Supprimer** dans le menu contextuel).
- Ouvrez la table *commandes* en mode *Création* et recréez le champ *NumClient* en utilisant l'assistant *liste de choix* : à la première étape, choisissez l'option *Je veux que la liste de choix recherche les valeurs dans une table ou requête*. À l'étape suivante, choisissez la table *Clients* et cliquez sur **Suivant** :



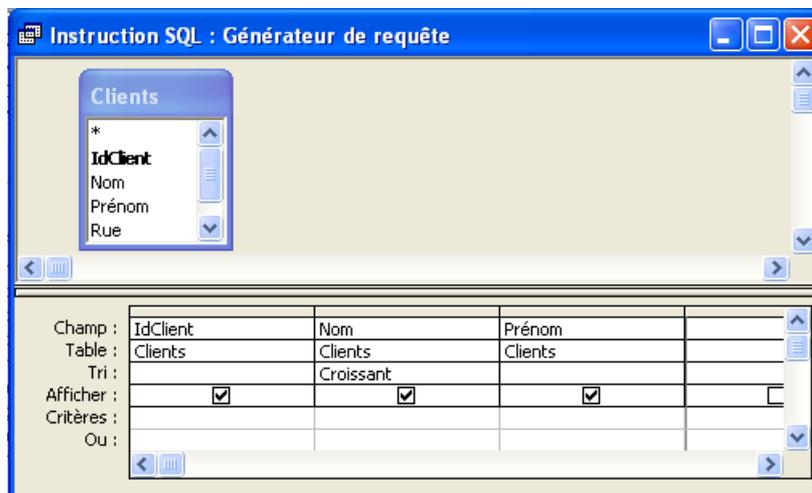
Sélectionnez les trois premiers champs (pour ajouter un champ cliquez sur son nom dans la colonne *Champs disponibles* puis sur le bouton **>** pour supprimer un champ, cliquez sur son nom dans la colonne *Champs sélectionnés* puis sur **<**) et cliquez sur **Suivant**.

À l'étape suivante, décochez l'option *Colonne clé cachée* et cliquez sur **Suivant**. La valeur à stocker est *IdClient*. Conservez *NumClient* comme étiquette et cliquez sur **Terminer**.

23) Passez dans la fenêtre *Relations*, ajoutez la table *commandes* et modifiez les propriétés de la relation.

24) Saisissez des commandes (une dizaine dont certaines par le même client).

Pour que les clients apparaissent dans l'ordre alphabétique, affichez la table *commandes* en mode *Création* (). Sélectionnez le champ *Numclient* et, dans l'onglet *Liste de choix*, cliquez sur la propriété *Contenu*. Un bouton **...** apparaît à droite de celle-ci ; cliquez dessus :



Sur la ligne *Tri*, en dessous du champ *Nom*, ajoutez *Croissant*. Fermez cette fenêtre en enregistrant les modifications. Enregistrez la table puis passez en mode *Feuille de données* ().

25) Créez la table *Détails commandes* avec les champs suivants :

- *NumCommande* : utilisez l'assistant *liste de choix*, en indiquant que les données doivent provenir de la table *commandes*, et en affichant les trois champs *NumCommande*, *Date-Commande* et *NumClient*. Décochez l'option *Colonne clé cachée*.
- *ISBN* : assistant *liste de choix*. Les données sont issues de la table *livres* dont on affichera les champs *ISBN*, *Titre* et *PrixHT*. Décochez l'option *Colonne clé cachée*. Vous afficherez les livres dans l'ordre alphabétique.
- *QuantitéCommandée* : type *Numérique*, taille *Entier* (pas long), valeur par défaut : 1.

Pour limiter les risques d'erreur de saisie, saisissez « >= 1 » (sans les guillemets) dans la propriété *Valide si* ; comme *Message si erreur*, entrez « La quantité commandée doit être positive ! ».

- 26) Définissez comme clé primaire la concaténation des deux premiers champs (sélectionnez-les en faisant glisser le pointeur de la souris, bouton gauche enfoncé, dans la marge à gauche des deux noms de champs puis cliquez avec le bouton droit et choisissez **Clé primaire**).
- 27) Passez dans la fenêtre *Relations*, ajoutez la table *Détail commandes*, et modifiez les propriétés des relations avec les tables *commandes* et *livres*.
- 28) Entrez des enregistrements (commande avec plusieurs livres, livre faisant l'objet de plusieurs commandes différentes).

3. Les types de champs

Les types de champ disponibles dans *Microsoft Access* sont les suivants :

- **Texte** : chaînes quelconques. On peut aussi utiliser ce type de champ pour des nombres qui ne nécessitent aucun calcul (numéros de téléphone, numéros INSEE, codes postaux...).
- *Taille du champ* : indiquez la longueur maximale autorisée (de 1 à 255 caractères).
- **Mémo** : utilisez ce type de champ pour de longues suites de caractères alphanumériques (jusqu'à 65536), telles que des notes ou des descriptions.
- **Numérique** : données numériques pouvant faire l'objet de calculs mathématiques.
La propriété *Taille du champ* peut prendre les valeurs suivantes :
 - *Octet* : nombres entiers compris entre 0 et 256.
 - *Décimal* : nombres compris entre $-(10^{28}-1)$ et $10^{28}-1$ avec jusqu'à 28 décimales.
 - *Entier* : nombres entiers compris entre -32 768 et 32 767.
 - *Entier long* : nombres entiers compris entre -2 147 483 648 et +2 147 483 647.
 - *Réel simple* : nombres compris entre $-3,402823 \cdot 10^{38}$ et $-1,401298 \cdot 10^{-45}$ pour les valeurs négatives et $1,401298 \cdot 10^{-45}$ et $3,402823 \cdot 10^{38}$ pour les valeurs positives.
 - *Réel double* : nombres allant de $-1,79769313486231 \cdot 10^{308}$ à $-4,94065645841247 \cdot 10^{-324}$ pour les valeurs négatives et de $+4,94065645841247 \cdot 10^{-324}$ à $+1,79769313486231 \cdot 10^{308}$ pour les valeurs positives.
 - *N° répliquon* : identifiant unique (GUID : séquence de 32 chiffres hexadécimaux).
- **Date/Heure** : données de type dates et/ou heures.
- **Monétaire** : stocke des valeurs de type monétaire. Ce type de champ empêche l'arrondi au chiffre supérieur lors des calculs.
- **NuméroAuto** : numéro séquentiel (augmentant d'une unité – choisissez *Incrément* dans la propriété *Nouvelles valeurs*) ou aléatoire (choisissez *Aléatoire*) généré automatiquement lors de l'ajout d'un enregistrement.
- **Oui/Non** : pour les données qui ne peuvent prendre que deux valeurs, comme Oui/Non, Vrai/Faux ou Actif/Inactif (à choisir dans la liste *Format*). Les valeurs *Null* ne sont pas autorisées.
- **Objet OLE** : ce type de champ permet de stocker des objets OLE (documents *Word*, classeurs *Excel*, images, sons...).
- **Lien hypertexte** : utilisez ce type de champ pour les liens hypertexte (liens locaux ou Internet).
- **Assistant Liste de choix** : cet assistant sert à créer un champ qui permet de choisir dans une liste déroulante une valeur provenant d'une autre table ou d'une liste.

Les propriétés suivantes sont communes à (presque) tous les types de champs :

- *Null interdit* : si cette propriété est fixée à *Oui*, l'utilisateur devra impérativement renseigner ce champ.
- *Chaîne vide autorisée* (champs de type *Texte* et *Mémo* uniquement) : si cette propriété a la valeur *Oui*, l'utilisateur peut saisir une chaîne vide (""), même si les valeurs *Null* sont interdites.

Quelle est la différence entre *Null* et une chaîne vide ? Une chaîne vide représente une information inexistante (par exemple le champ *Adresse2* pour les clients qui ne sont pas concernés), *Null* une information qui existe peut-être mais qui est inconnue. En résumé :

Null interdit	Chaîne vide autorisée	Action de l'utilisateur	Valeur enregistrée
Non	Non	Touche <i>Entrée</i> Touche <i>Espace</i> Chaîne vide ("")	<i>Null</i> <i>Null</i> (interdit)
Non	Oui	Touche <i>Entrée</i> Touche <i>Espace</i> Chaîne vide ("")	<i>Null</i> <i>Null</i> Chaîne vide
Oui	Non	Touche <i>Entrée</i> Touche <i>Espace</i> Chaîne vide ("")	(interdit) (interdit) (interdit)
Oui	Oui	Touche <i>Entrée</i> Touche <i>Espace</i> Chaîne vide ("")	(interdit) Chaîne vide Chaîne vide

- *Légende* : texte qui remplace le nom du champ dans les colonnes des feuilles de données, les étiquettes des formulaires et des états et pages d'accès aux données.

Remarques :

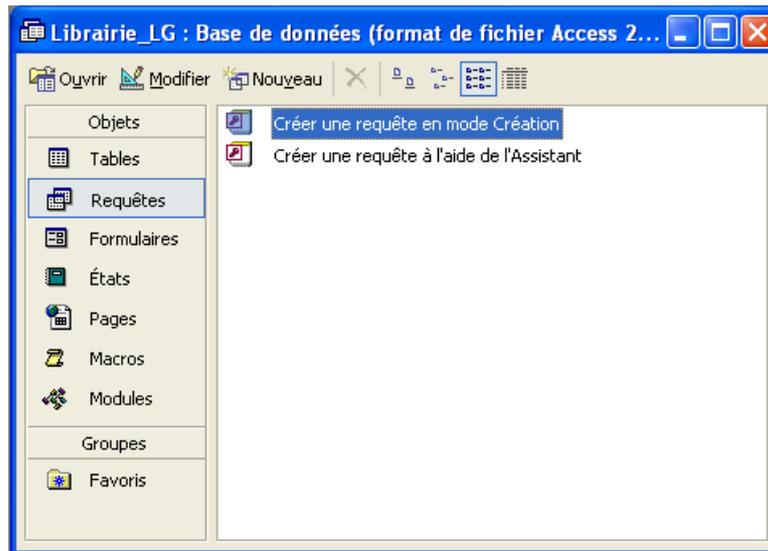
1. Choisissez une *Taille de champ* la plus petite possible car les données de petite taille prennent moins de place en mémoire et sont traitées plus rapidement.
2. On peut modifier la taille par défaut des champs de type *Texte* et *Numérique* : **Options** ▶ **Outils**, onglet *Tables/Requêtes*.
3. Si vous réduisez la taille d'un champ qui contient déjà des informations, vous risquez de perdre des données. Il devient impossible d'annuler les changements apportés aux données à la suite d'une modification de la propriété *Taille du champ* dès que celle-ci a enregistré.

III. Les requêtes sélection

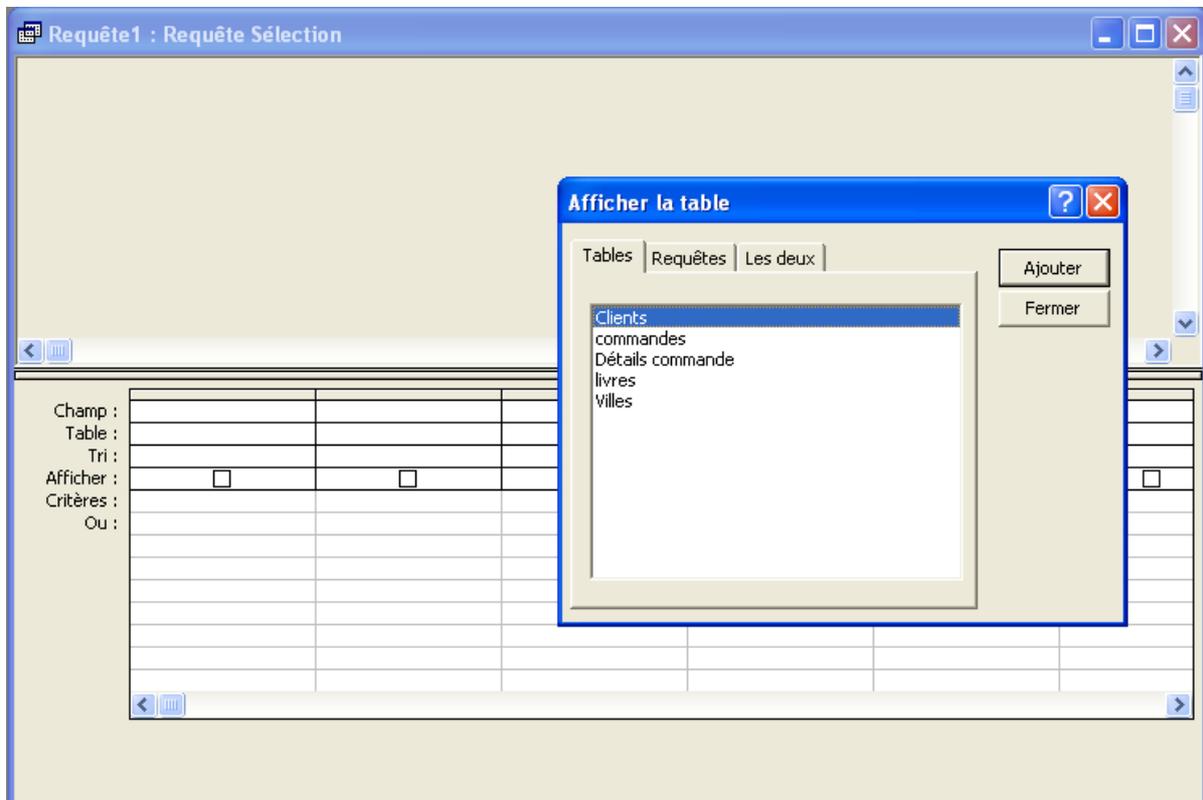
Il existe deux grands types de requêtes : les requêtes sélection et les requêtes action, que nous aborderons au prochain semestre. Une requête sélection récupère des données contenues dans une ou plusieurs tables et affiche les résultats sous la forme d'une feuille de données dans laquelle il est possible de travailler comme dans une table, même si, en réalité, les données se trouvent dans la (ou les) table(s) sur laquelle porte la requête.

1. Sélectionner des données depuis une table unique

Pour créer une nouvelle requête, revenez à la fenêtre de base de données et sélectionnez la catégorie *Requêtes* :



Double-cliquez sur *Créer une requête en mode création* :



La partie supérieure de cette fenêtre sert à afficher la table ou les tables qui serviront de base à la requête, avec leurs champs. Dans la partie inférieure, on trouve la grille QBE (*Query By Example*) qui va nous servir à créer la requête.

a. Critères et tri

Notre objectif est de créer une requête (nommée **cliNPdC**) qui affiche les clients du Nord-Pas-de-Calais triés par code postal, puis par nom.

29) Sélectionnez la table *Clients* et cliquez sur **Ajouter** puis sur **Fermer**.

30) Sélectionnez les champs qui devront intervenir dans la requête (pour nous, *IdClient*, *Nom*, *Prénom* et *CodePostal*).

Pour ajouter un champ, il existe plusieurs méthodes :

- double-cliquez sur son nom ;
- faites-le glisser de la table vers l'une des colonnes de la grille QBE.

Le champ * représente l'ensemble des champs de la table.

Pour supprimer un champ, sélectionnez la colonne (un clic sur l'en-tête) et appuyez sur *Suppr.*

Pour remplacer un champ déjà inséré dans la grille par un autre, cliquez sur son nom et sélectionnez le nom du champ de remplacement dans la liste déroulante.

L'option *À afficher* (activée par défaut), indique les champs qui seront affichés dans le résultat de la requête. Pour masquer un champ, il suffit donc de désactiver cette option.

31) Pour trier les enregistrements en fonction de la valeur d'un champ, cliquez dans la ligne *Tri* en face de celui-ci et choisissez l'ordre (*Croissant* ou *Décroissant*). Il est possible de trier en fonction de plusieurs champs. Dans ce cas, l'ordre des clés de tri est de gauche à droite ; pour déplacer un champ, sélectionnez la colonne (un clic sur son en-tête – pointeur ↓) puis saisissez l'en-tête de la colonne et faites-le glisser vers l'emplacement qui vous convient.

32) Pour sélectionner les clients du Nord-Pas-de-Calais (codes postaux commençant par 59 ou 62), cliquez dans la ligne *Critères* en dessous de *CodePostal* et saisissez « 59* » (sans les guillemets) puis « 62* » sur la ligne d'en dessous.

Vous pouvez aussi saisir directement sur une seule ligne l'expression « Comme "59*" Ou Comme "62*" ».

La requête terminée devrait ressembler à ceci :

Champ :	IdClient	Code_Postal	Nom	Prénom		
Table :	Clients	Clients	Clients	Clients		
Tri :		Croissant	Croissant			
Afficher :	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Critères :		Comme "59*"				
Ou :		Comme "62*"				

- 33) Enregistrez-la sous le nom *cliNPdC* et exécutez-la : cliquez sur le bouton  pour visualiser les résultats en mode *Feuille de données*.
- 34) Créez une requête (que l'on appellera **liv50a100**) qui extrait les livres dont le prix HT est compris entre 50 et 100 euros et les trie par prix décroissant
Dans la ligne *Critères*, entrez l'expression « Entre 50 Et 100 » ou « >=50 Et <=100 » (sans les guillemets).

Opérateurs de comparaison

Les opérateurs de comparaison utilisables dans un critère sont <, =, <>, >, <=, >= et *Entre ... Et...*

Opérateurs logiques

Les principaux opérateurs logiques sont *Et*, *Ou*, *Ou_X* (Ou exclusif) et *Pas* (négation – par exemple le critère « Pas Comme "59*" » appliqué au champ *CodePostal* permet de trouver tous les clients qui n'habitent pas le Nord).

Jokers

L'astérisque (*) représente une chaîne de caractères quelconque, le point d'interrogation (?) un caractère quelconque et le dièse (#) un chiffre quelconque.

Critères portant sur des champs de type texte

Pour rechercher une valeur particulière, saisissez-la (entre guillemets) ; ainsi le critère "Nord" appliqué au champ *Département* permet de rechercher les clients du Nord.

Appliqués à un champ de type *Texte*, les opérateurs de comparaison portent sur l'ordre alphabétique (« > » signifie « vient après dans l'ordre alphabétique »). Ainsi le critère « <"Dupont" » appliqué au champ *Nom* de la table *Clients* permet de sélectionner les clients dont le nom vient avant « Dupont » dans l'ordre alphabétique.

Le mot-clé *Comme* permet de rechercher les valeurs d'un champ qui correspondent au modèle indiqué. Ainsi le critère « Comme "S*" » appliqué au champ *Nom* de la table *Clients* permet de sélectionner les clients dont le nom commence par la lettre S tandis que « Comme "[A-D]*" » permet de sélectionner les clients dont le nom a pour première lettre A, B, C ou D.

Les fonctions *Gauche()*, *Droite()* et *NbCar()* fonctionnent comme sous *Excel*.

Les listes

La fonction *In()* permet de rechercher les valeurs d'un champ qui appartiennent à une série donnée. Ainsi le critère « In("Nord";"Pas-de-Calais") » appliqué au champ *Département* permet de rechercher les clients du Nord-Pas-de-Calais.

Les dates

Dans un critère, les dates doivent être encadrées par des signes dièse. Ainsi, le critère « #02/11/2005# » appliqué au champ *DateCommande* permet de rechercher les commandes passées le 2 novembre 2005.

La fonction *Date()* renvoie la date système (la date actuelle, avec mise à jour). Les fonctions *Année()*, *Mois()*, *Jour()*, *Joursem()* fonctionnent comme dans *Excel*.

Valeur de champ vide (Null)

Null désigne un champ vide. Ainsi, le critère « Est Null » permettra de sélectionner les enregistrements pour lesquels le champ sélectionné n'a pas été renseigné tandis que « Est Pas Null » permet de rechercher ceux pour lesquels le champ a été complété.

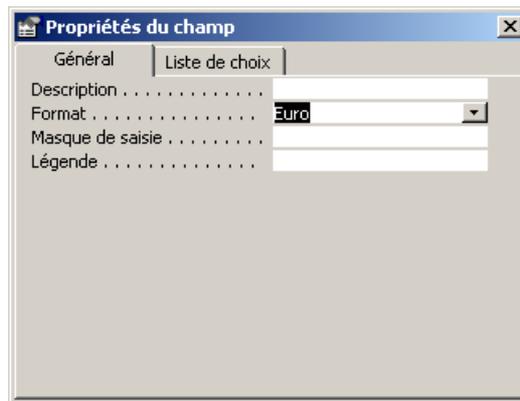
ET ou OU ?

Comme dans les filtres élaborés d'*Excel*, des critères se trouvant sur la même ligne sont reliés par l'opérateur *Et*, tandis que des lignes différentes de la grille QBE, sont reliées par *Ou*.

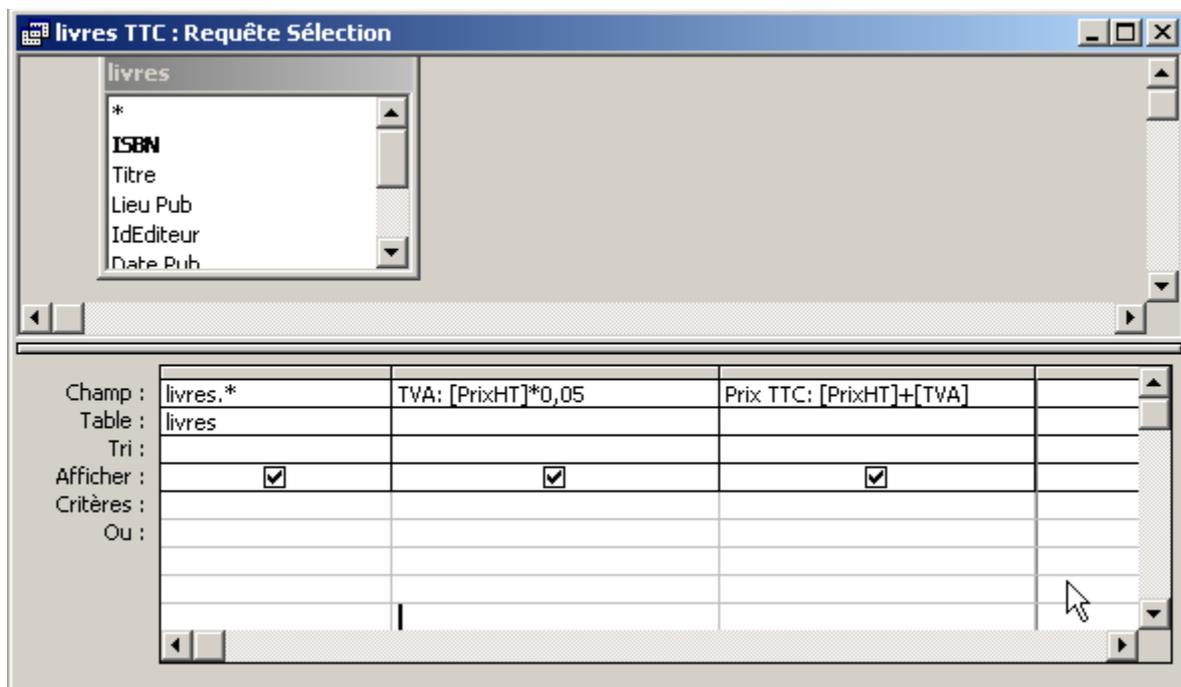
b. Champ calculé

Dans une requête, il est possible d'ajouter des champs calculés à partir des champs existants. On peut saisir directement la formule mais le plus simple est d'utiliser le générateur d'expression.

- 35) Créez une requête que vous enregistrerez sous le nom **livresTTC** qui comprend les champs calculés *TVA* (la TVA sur les livres est à 5,5 %), que l'on mettra au format monétaire, et *Prix TTC*.
- Dans la fenêtre de base de données, cliquez sur *Créer une requête en mode création*.
 - Ajoutez la table *livres* puis tous ses champs (champ *)
 - Dans la première ligne de la colonne suivante, saisissez l'expression suivante :
TVA: [PrixHT]*0,055 (le nom du champ calculé suivi de « : » et de la formule de calcul)
Les noms de champs intervenant dans le calcul doivent être mis entre crochets. Si vous ne faites pas appel au générateur d'expression, attention aux fautes de frappe !
 - Dans la première ligne de la colonne suivante, saisissez l'expression suivante :
PrixTTC: [PrixHT]+[TVA]
(un champ calculé peut faire appel un autre champ calculé **s'il est affiché**)
 - Pour modifier le format de ces deux champs, cliquez sur ceux-ci avec le bouton droit et choisissez *Propriétés* dans le menu contextuel :



En face de *Format*, choisissez *Euro* dans la liste déroulante.
La requête terminée ressemblera à ceci :



- 36) Créez une requête appelée **clients par zone** qui comprend le champ calculé *Zone* qui répartit les clients en trois zones : le Nord-Pas-de-Calais, la Région parisienne et le Reste de la France. On utilisera la fonction *VraiFaux* (équivalente de la fonction *SI* de *Excel*), la fonction *Gauche* et l'opérateur *Ou*. Vous trierez les clients par ordre alphabétique.

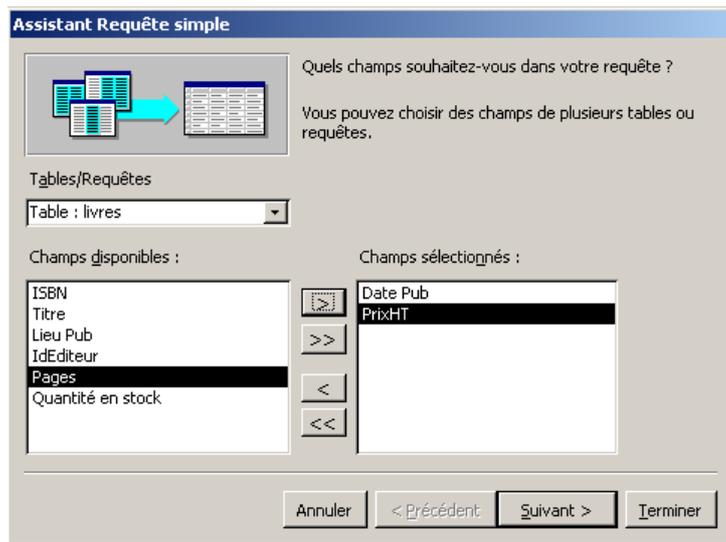
Zone: VraiFaux(Gauche([Code Postal];1)="9" Ou Gauche([Code Postal];2)="75";"Région parisienne";VraiFaux(Gauche([Code Postal];2)="59" Ou Gauche([Code Postal];2)="62";"Nord-Pas-de-Calais";"Reste de la France"))

c. Requêtes opérations

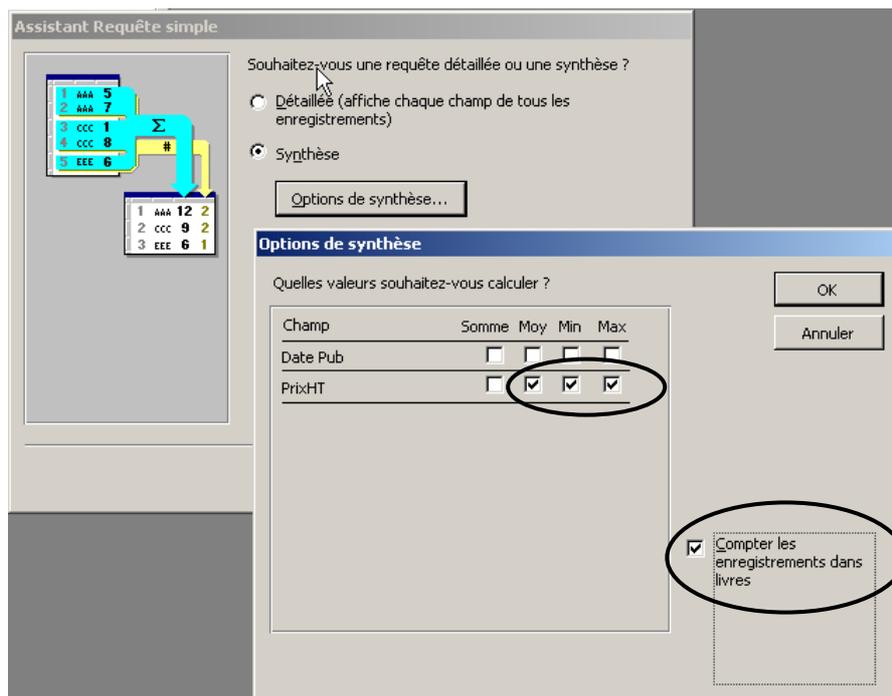
Ces requêtes, qui permettent d'effectuer des opérations comme des comptages ou des calculs de somme ou de moyenne, sont appelées aussi requêtes de synthèse ou requêtes de regroupement.

37) Créez une requête appelée **StatsLivParAnnée** qui pour chaque année d'édition des livres, permet de connaître le nombre de livre édités, le prix minimum, le prix maximum, le prix moyen....

- Dans la fenêtre de base de données, cliquez sur *Créer une requête à l'aide de l'assistant*.

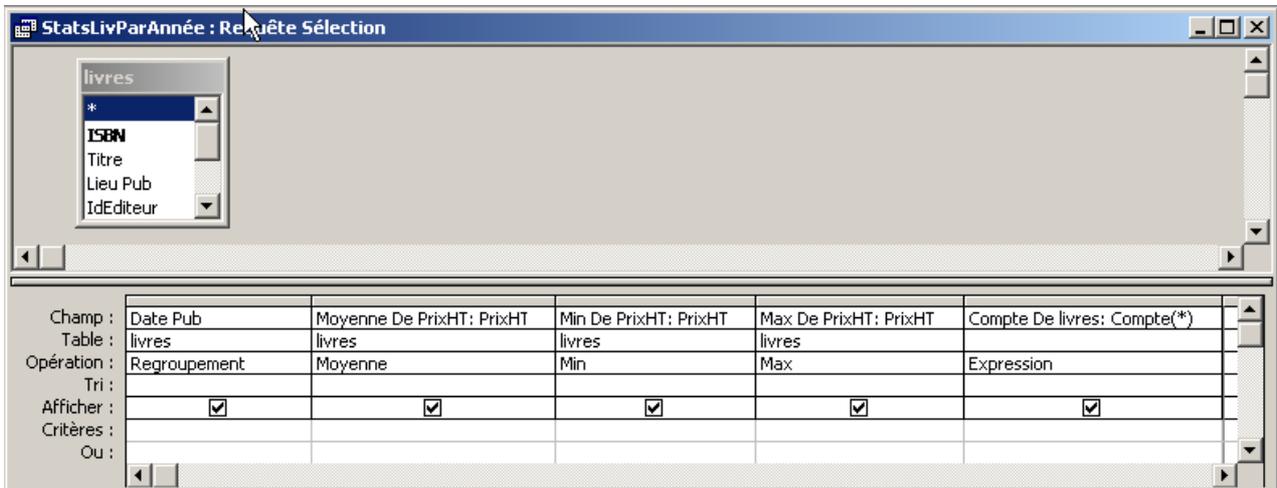


- Choisissez la table *livres*, puis ajoutez les champs *Datepub* et *PrixHT* et cliquez sur **Suivant**.
- Cochez la case *Synthèse* et cliquez sur le bouton **Options de synthèse** pour cocher les cases correspondant à ce que vous désirez obtenir :

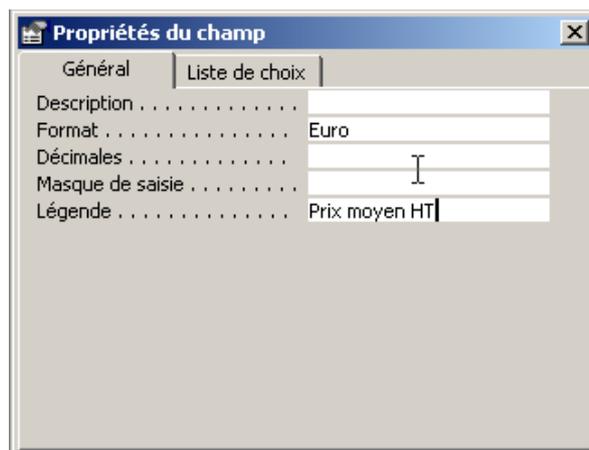


- Validez et cliquez sur **Suivant**.
- Donnez un nom à la requête, choisissez *Ouvrir la requête pour afficher les informations* et cliquez sur **Terminer**.

38) Cliquez sur le bouton  pour visualiser la conception de la requête :



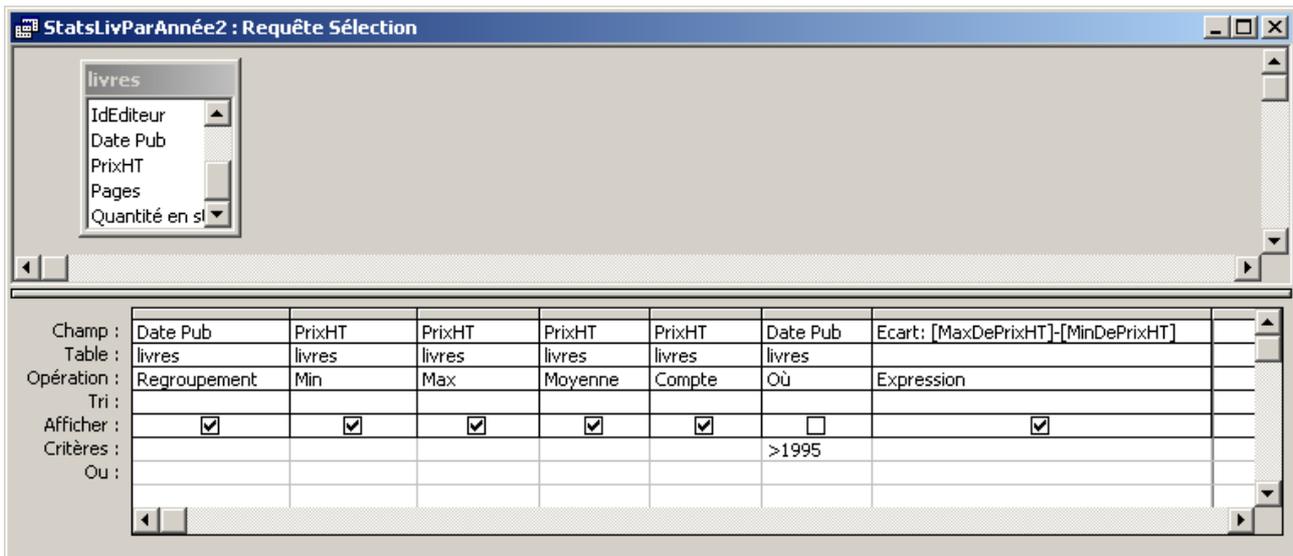
Remarque : on peut modifier l'intitulé des champs. Par exemple, cliquez avec le bouton droit sur le champ *Moyenne De PrixHT* et choisissez **Propriétés** dans le menu contextuel. Dans la case *Légende*, entrez « Prix moyen HT » :



39) Nous allons refaire cette requête sans l'aide de l'assistant (vous l'enregistrerez sous le nom **StatsLivParAnnée2**) :

- Créez une nouvelle requête en mode création.
- Ajoutez la table *livres*.
- Ajoutez le champ *Date Pub* et quatre fois le champ *PrixTTC*.
- Cliquez sur le bouton Σ (regroupement) de la barre d'outils *Base de données*.
- Dans la ligne *Opération*, laissez *Regroupement* en dessous du champ *Date Pub*, et choisissez *Min*, *Max*, *Moyenne* et *Compte* en dessous des quatre autres champs.
- Visualisez la requête et enregistrez-la.
- Rajoutez un champ calculé *Ecart: [MaxDePrixHT]-[MinDePrixHT]* et choisissez *Expression* dans la liste sur la ligne *Opérations*.
- Visualisez la requête et enregistrez-la.
- Nous souhaitons maintenant nous limiter aux livres publiés après 1995. Pour cela, ajoutez un champ *Date Pub* ; entrez « >1995 » dans la ligne *Critères* et choisissez « Où » dans la liste *Opération*. Il est inutile d'afficher ce champ.

La requête terminée doit ressembler à ceci :

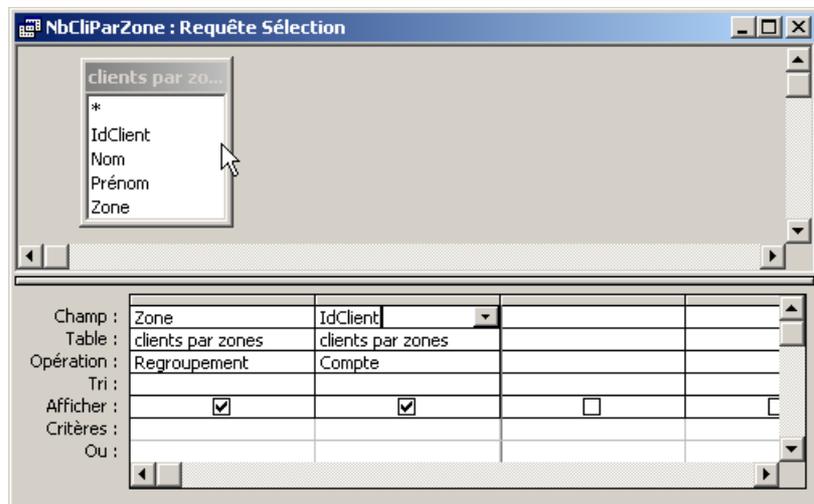


40) Créez une requête nommée **NbLivresParTranche** permettant de connaître le nombre de livres par tranches de prix HT de 25 € (tranche 1 : de 0 à moins de 25 €, tranche 2 : de 25 à moins de 50...)

d. Requetes en cascade

Au lieu de rechercher directement des données dans une table, une requête peut être basée sur une autre requête. Par exemple, nous souhaitons connaître le nombre de clients par zone :

41) Créez une nouvelle requête en mode *Création* et affichez la requête clients par zone. Ajoutez les champs *Zone* et *IdClient*. Effectuez un regroupement sur le champ *Zone* et un comptage sur le champ *IdClient*. N’oubliez pas de changer la légende du champ *IdClient* en *Nombre de clients*. La requête complétée doit ressembler à ceci :



42) Enregistrez cette requête sous le nom **NbCliParZone**, et affichez les données.

e. Requête pour trouver les doublons

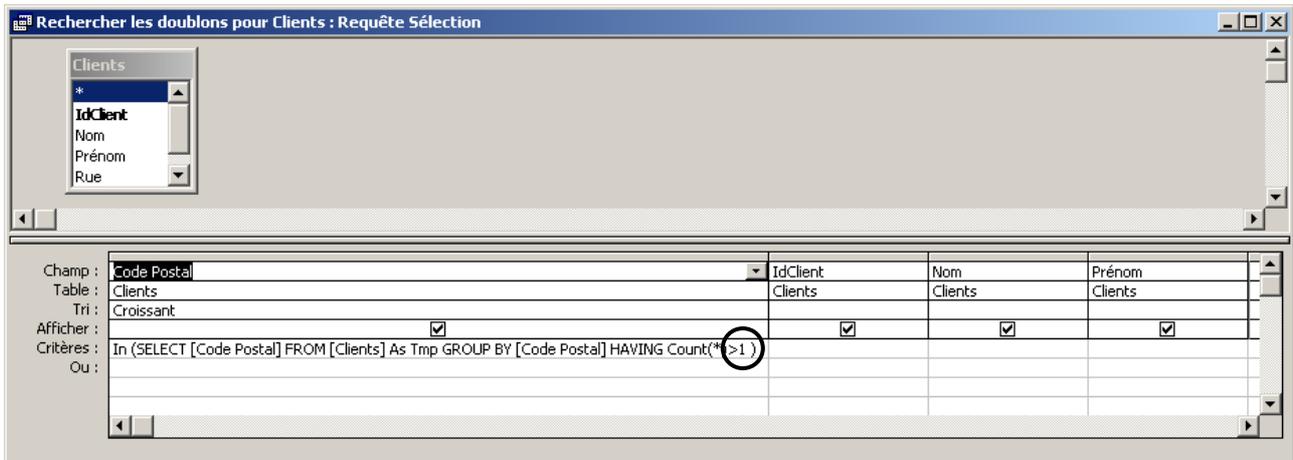
Nous souhaitons maintenant créer une requête appelée **VillesPlusieursClients** qui nous permettra de connaître les villes où il y a plus d’un client.

43) Cliquez sur le bouton **Nouveau**. Choisissez l’Assistant *Requête trouver les doublons* et cliquez sur **OK**. Les étapes de la construction de la requête sont les suivantes :

- (1) Choisissez la table *Clients*.
- (2) Indiquez *Code Postal* comme champ susceptible de contenir des doublons.

- (3) Comme champs supplémentaires à afficher, choisissez *IdClient*, *Nom* et *Prénom*.
- (4) Changez le nom de la requête (**VillesPlusieursClients**) et cliquez sur **Terminer**.

44) Passez en mode *Création* et observez la conception de la requête :

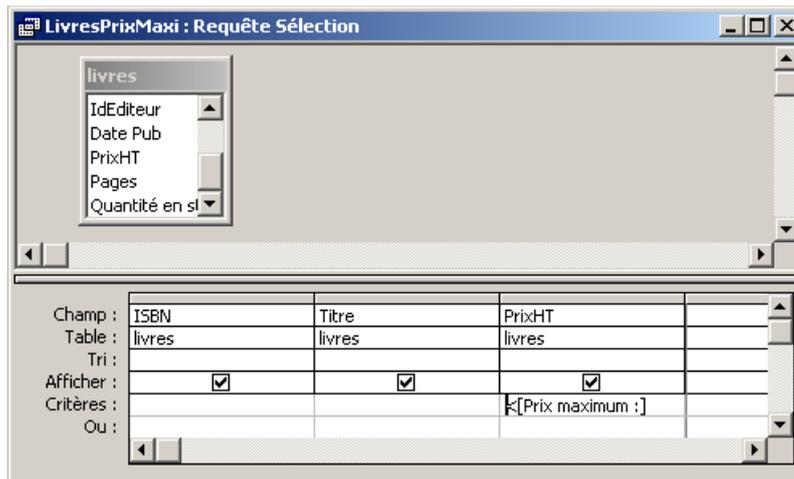


- 45) Nous souhaitons maintenant connaître les villes où il y a plus de 10 clients. Ouvrez la requête **VillesPlusieursClients** en mode création, remplacez 1 par 10. Appelez la commande **Fichier ▶ Enregistrer sous** et donnez comme nom **VillesPlus10Clients**. Exécutez la requête.

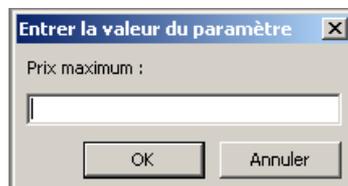
f. Requetes paramétrées

Une requête paramétrée permet de saisir une (ou plusieurs) valeur(s) intervenant dans un ou plusieurs critères au moment de l'exécution de la requête (et non lors de sa conception).

- 46) Nous souhaitons créer une requête (nommée **LivresPrixMaxi**) permettant d'afficher la liste des livres dont le prix est inférieur à une valeur X saisie par l'utilisateur. En regard du champ *PrixHT* dans la ligne *Critères*, saisissez « <[Prix maximum :] » (sans les guillemets) :



Lors de l'exécution de la requête, le texte placé entre crochets (attention à ne pas utiliser un nom de champ existant dans la requête) s'affiche dans une boîte de dialogue qui invite l'utilisateur à saisir la valeur qui le remplacera lors de l'exécution :



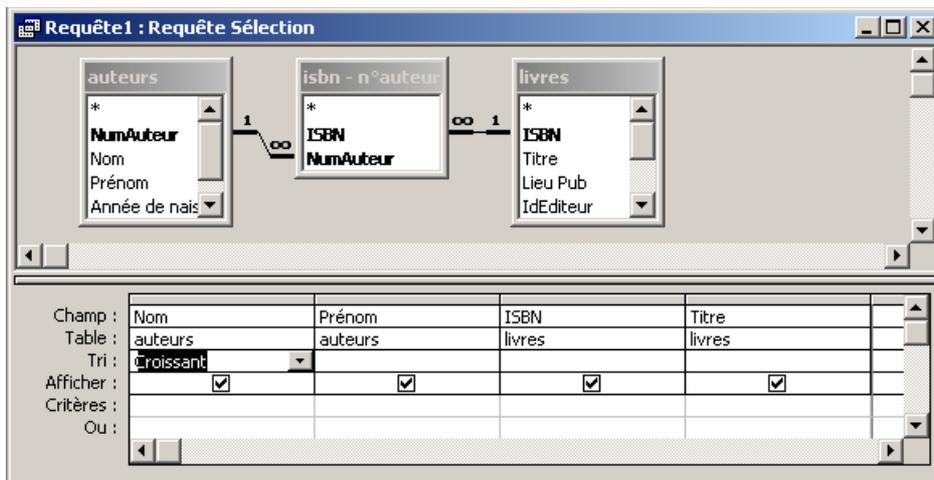
- 47) Créez une requête nommée **LivresAnnéeX** qui permet d’extraire les livres d’une année spécifiée par l’utilisateur lors de l’exécution de la requête.
- 48) Créez une requête nommée **NbLivresParAnnéeDepuisX** qui permet de connaître le nombre de livres par an, depuis une année spécifiée par l’utilisateur lors de l’exécution de la requête.

2. Sélectionner des données depuis plusieurs tables

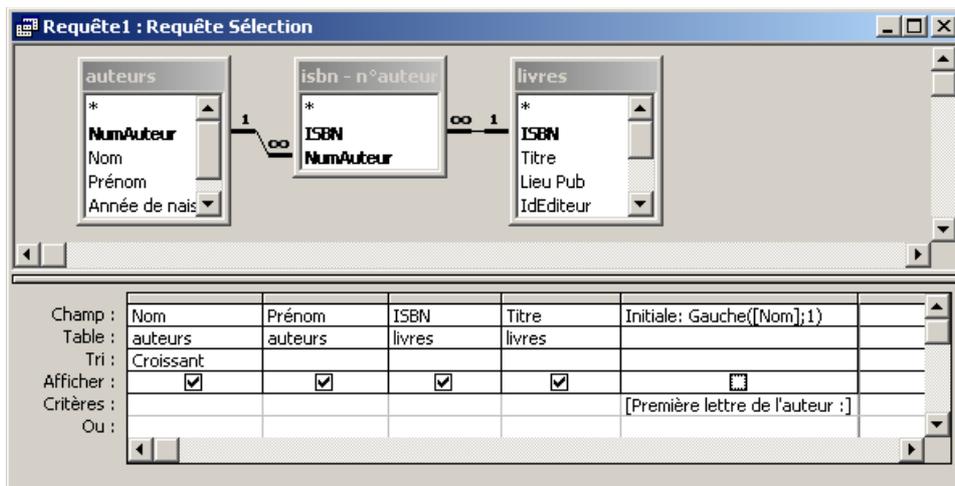
Il est souvent nécessaire d’extraire des données qui concernent plusieurs tables. On prendra soin de vérifier que les relations qui les unissent sont pertinentes par rapport à la requête (il est possible dans la requête de supprimer certaines relations et d’en rajouter d’autres).

a. Sélectionner des données

Comme premier exemple de requête portant sur plusieurs tables, nous allons construire une requête appelée **Auteurs et ses Livres** qui permet d’afficher les auteurs avec les livres qu’ils ont écrits, triés par ordre alphabétique des auteurs :



- 49) Créez une nouvelle requête en mode création et ajoutez les tables *auteurs*, *isbn – n° auteur* et *livre*. Insérez les champs *Nom* (avec tri croissant) et *Prénom* de la table *auteurs* ainsi que *ISBN* et *Titre* de la table *livres*.
- 50) Comme la liste est très longue, nous allons permettre à l’utilisateur de saisir l’initiale des auteurs à afficher dans un champ calculé :



Remarque : on pouvait se passer du champ calculé en faisant porter comme critère sur le champ *Nom* : « Comme [Première lettre de l'auteur :] & "*" ».

b. *Champs calculés portant sur plusieurs tables*

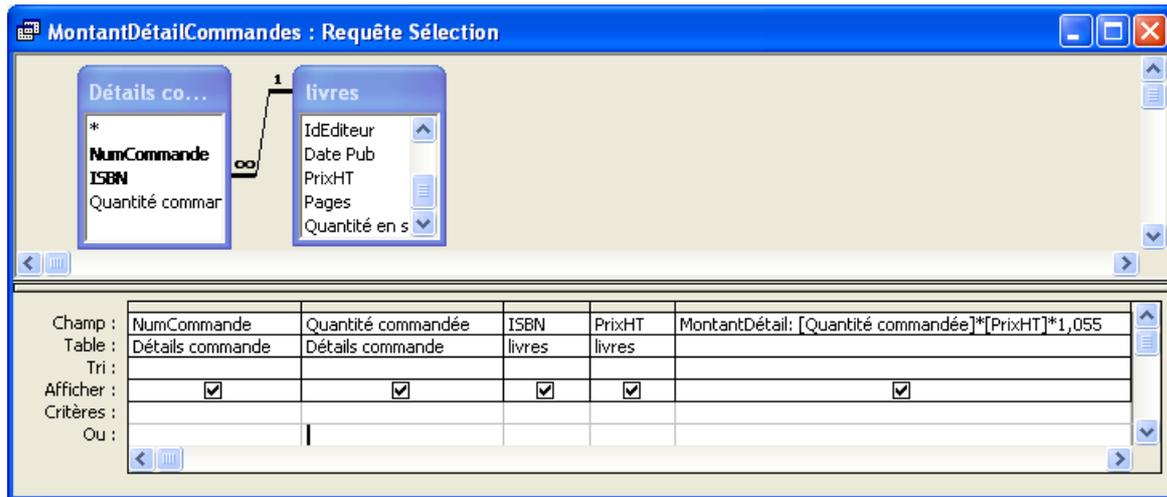
Dans une requête multitable, on peut insérer des champs calculés à partir de champs issus de tables différentes qui sont en relation.

51) Dans un premier temps, nous allons créer une requête opération appelée **MontantDétailCommandes** qui permet de connaître le montant TTC d'une ligne d'une commande :

Créez une nouvelle requête en mode *Création* et ajoutez les tables *Détails commandes* et *livres*. Affichez les champs *NumCommande*, *Quantité commandée* (de *Détails commande*), *ISBN* et *PrixHT* (de *livres*) ainsi que le champ calculé suivant :

« MontantDétail:[Quantité commandée]*[PrixHT]*1,055 »

N'oubliez pas de mettre ce champ au format *Euro*. La requête terminée doit ressembler à ceci :

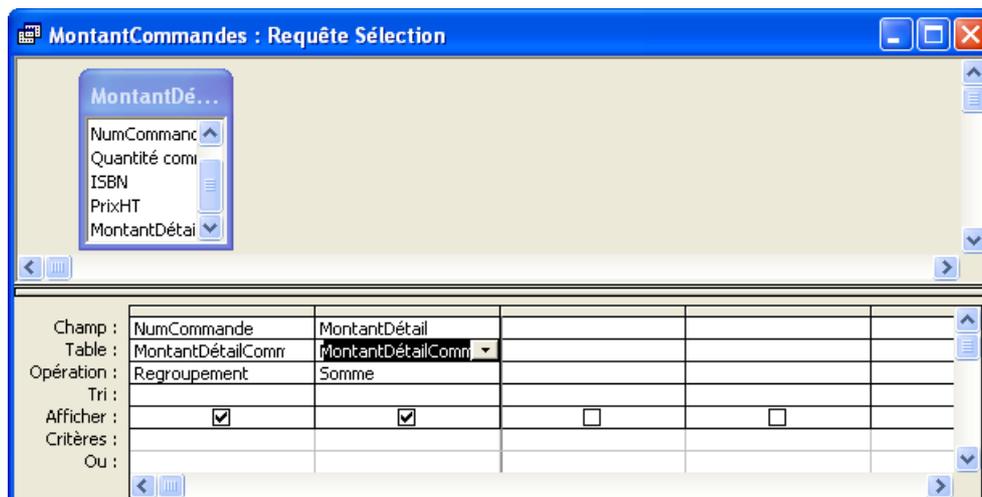


52) A partir de cette requête, nous allons créer une nouvelle requête **MontantCommandes** qui calcule le montant total de chaque commande.

Créez une nouvelle requête en mode *Création* et ajoutez la requête *MontantDétailCommandes*. Affichez les champs *NumCommande* et *MontantDétail*.

Cliquez sur le bouton Σ (regroupement) de la barre d'outils *Base de données* et, dans la ligne *Opération*, choisissez *Regroupement* pour le champ *NumCommande* et *Somme* pour *MontantDétail*. N'oubliez pas de mettre ce dernier champ au format *Euro*.

La requête terminée doit ressembler à ceci :



Remarque : nous verrons au second semestre des solutions plus rapides pour obtenir ce résultat dans les formulaires et les états.

c. **Requête de non-correspondance**

Le but d'une requête de non correspondance est de rechercher les enregistrements d'une table de données source qui ne sont liés à aucun enregistrement de la table de données cible. Par exemple, nous allons créer une requête nommée **AuteursSansLivres** qui permettra d'afficher les auteurs n'ayant pas écrit de livre.

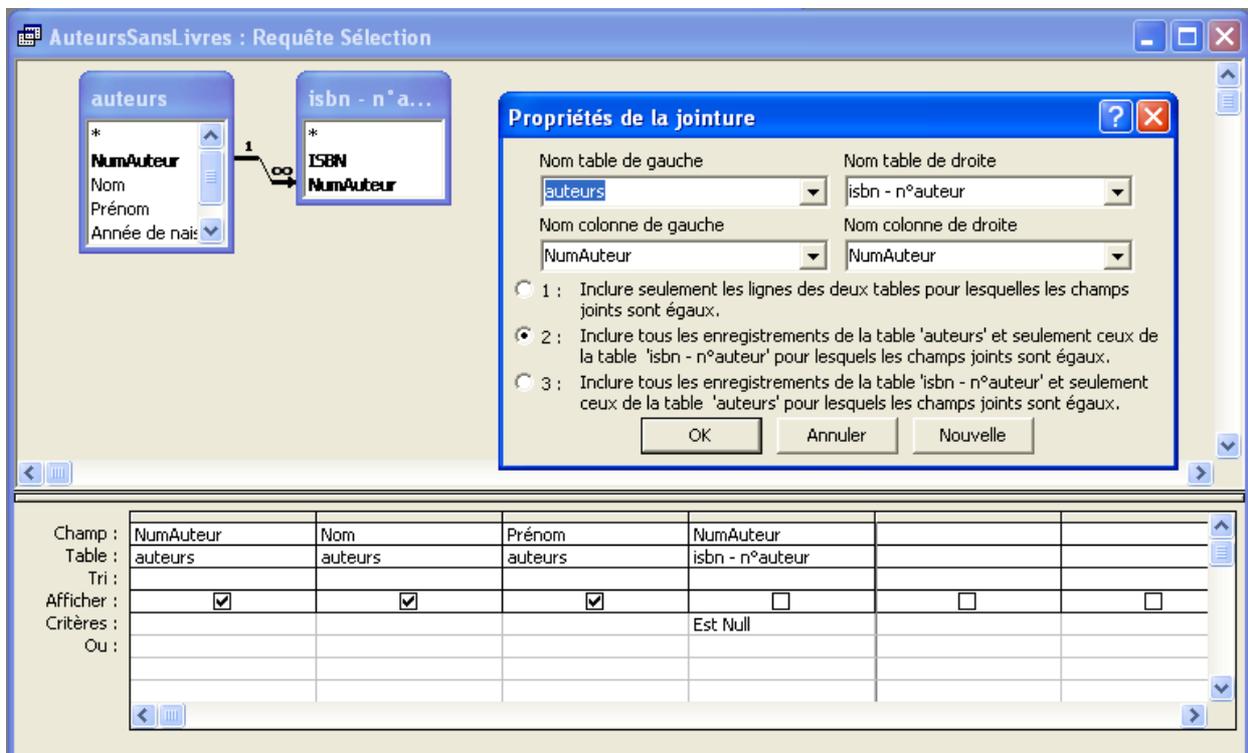
53) Pour pouvoir tester cette requête, ajoutez les auteurs suivants dans la table *auteurs* :

NumAuteur	Nom	Prénom	Année de naissance	Année du décès
111	Camus	Albert	1913	1960
112	Pascal	Blaise	1623	1662
113	Sartre	Jean-Paul	1905	1980

54) Nous allons créer cette requête à l'aide de l'Assistant Requête de non correspondance (cliquez sur le bouton **Nouveau** de la fenêtre de base de données. Les étapes de l'assistant sont les suivantes :

- (1) Sélectionnez la table *auteurs*.
- (2) La table en relation est la table *isbn - n°auteur*.
- (3) Les champs en relation sont les champs *NumAuteur* des deux tables. Sélectionnez donc *NumAuteur* dans la table *auteurs* et *NumAuteur* dans *isbn - n°auteur* puis cliquez sur le bouton **<=>**.
- (4) Indiquez que vous souhaitez afficher les champs *NumAuteur*, *Nom* et *Prénom*.
- (5) Nommez la requête **AuteursSansLivres** et cliquez sur **Terminer** pour afficher les résultats.

55) Passez en mode *Création* pour examiner la conception de la requête et notamment la propriété de la jointure (cliquez avec le bouton droit sur celle-ci et choisissez **Propriétés de la jointure** dans le menu contextuel) :

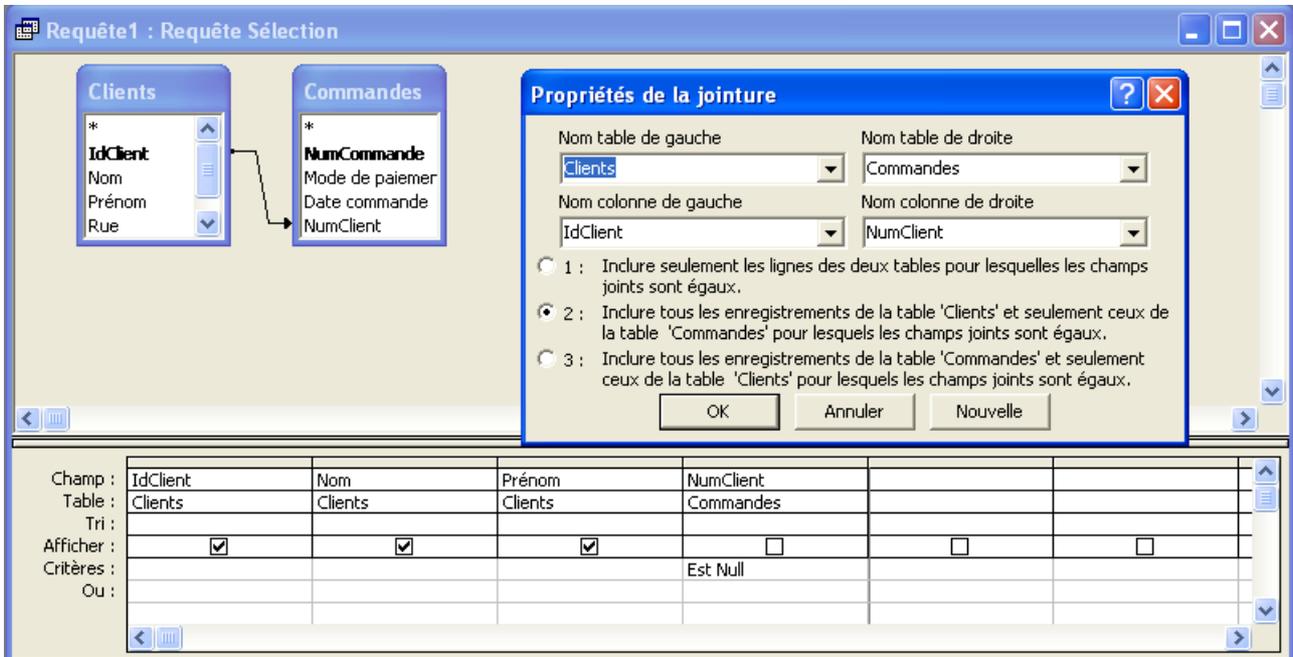


56) Construisons maintenant sans l'assistant une requête de même nature qui sera nommée **Clients-SansCommande** pour connaître les clients qui n'ont jamais passé de commandes :

- Créez une nouvelle requête en mode création.
- Ajoutez les deux tables *Clients* et *Commandes*.

- Insérez les champs *IdClient*, *Nom*, *Prénom* (de la table *Clients*) et *NumClient* (de la table *Commandes*).
- Modifiez les propriétés de la jointure (*Inclure tous les enregistrements de la table 'Clients' et seulement ceux de la table 'Commandes' pour lesquels les champs joints sont égaux*).
- Ajoutez « Est Null » comme critère pour le champ *NumClient* (il est donc inutile d'afficher ce champ !).

La requête terminée devra ressembler à ceci :

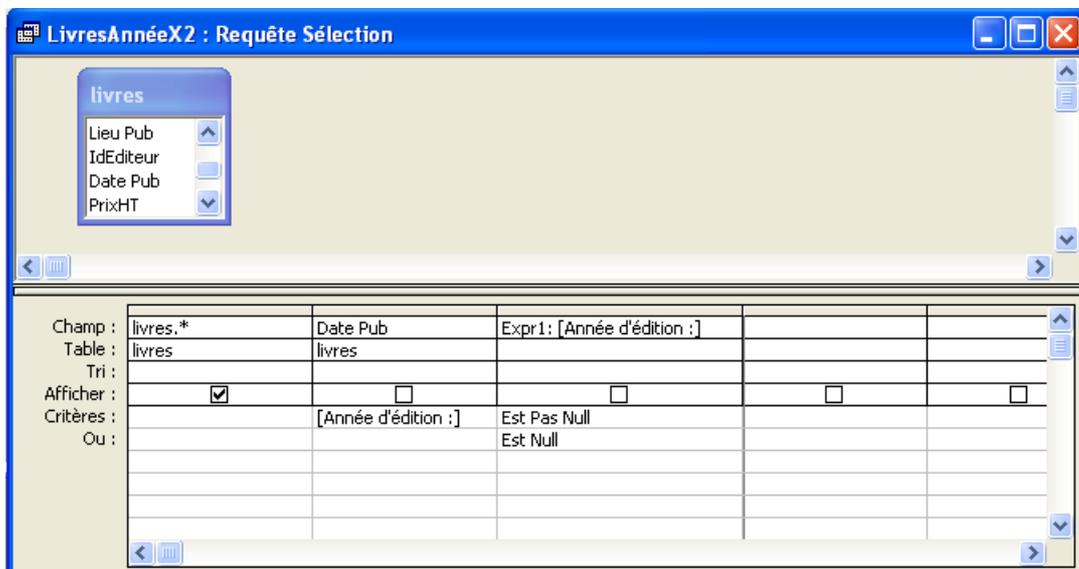


3. Quelques requêtes plus complexes

a. Requetes avec paramètres facultatifs

Si, lors de l'exécution d'une requête paramétrée, l'utilisateur ne saisit pas de valeur pour le paramètre, aucun enregistrement n'est affiché. Si l'on souhaite au contraire que dans ce cas tous les enregistrements soient affichés, il faut prévoir de traiter les deux éventualités.

57) Nous allons nous appuyer sur l'exemple de la requête **LivresAnnéeX**. Ouvrez-la en mode création et enregistrez-en une copie sous le nom **LivresAnnéeX2** puis modifiez-la comme suit :



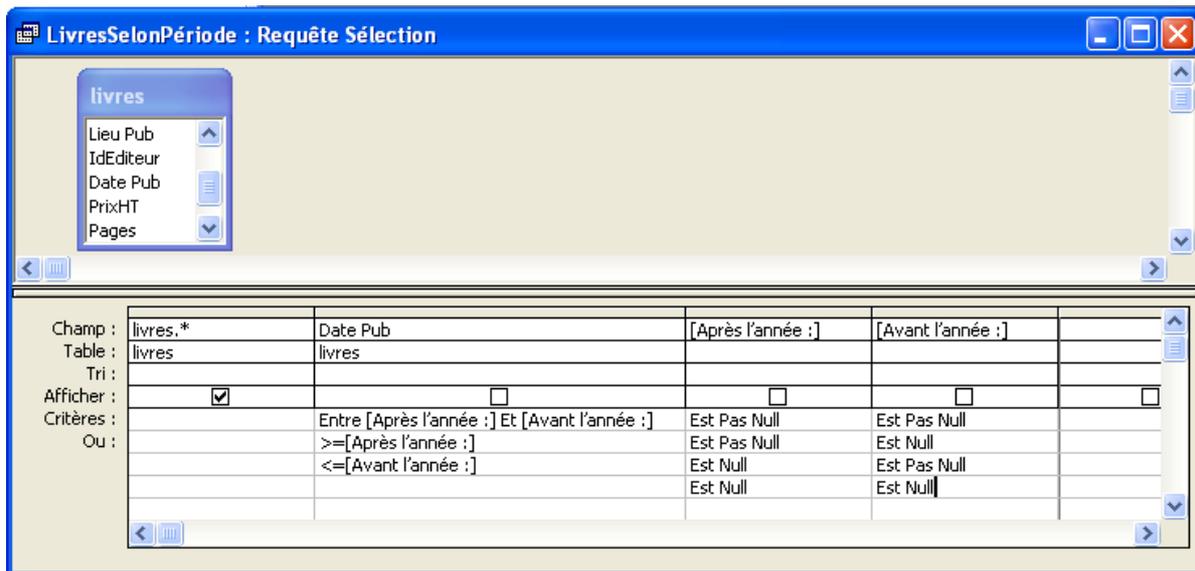
Pour pouvoir distinguer si le critère [Année d'édition] a été saisi ou pas, on ajoute un champ calculé qui se borne à reprendre la valeur de ce paramètre : « Expr1: [Année d'édition :] ».

Il reste à traiter les deux cas :

- le critère a été saisi (n'est pas *Null*) et alors la *Date Pub* des livres doit être égale à ce paramètre.
- le critère n'a pas été saisi (il est *Null*) et alors aucun critère ne doit être appliqué.

Attention ! Il est recommandé de faire des copier/coller pour respecter la syntaxe exacte du paramètre (en cas de faute de frappe, *Access* reconnaîtrait deux paramètres différents).

- 58) Créons maintenant une requête nommée **LivresSelonPériode** qui affiche les livres entre une année X et une année Y saisies par l'utilisateur. Si celui-ci ne saisit pas l'année de départ, on affichera tous les livres édités avant l'année Y ; s'il ne saisit pas d'année de fin, on affichera tous les livres publiés après l'année X ; s'il ne saisit aucun paramètre, on affichera tous les livres :



Avec deux paramètres facultatifs, le problème est un peu plus complexe puisqu'il faut distinguer entre quatre éventualités :

- les deux paramètres ont été saisis : le critère est « Entre [Après l'année :] Et [Avant l'année :] » ;
- seul [Après l'année :] a été saisi : le critère est « >=[Après l'année :] » ;
- seul [Avant l'année :] a été saisi : le critère est « <=[Avant l'année :] » ;
- aucun paramètre n'a été saisi : pas de restriction.

b. Autres requêtes complexes

Nous souhaitons créer une requête qui affiche les autres clients habitant la même ville (même code postal) qu'un client dont le nom sera saisi par l'utilisateur. Il faut procéder en plusieurs étapes :

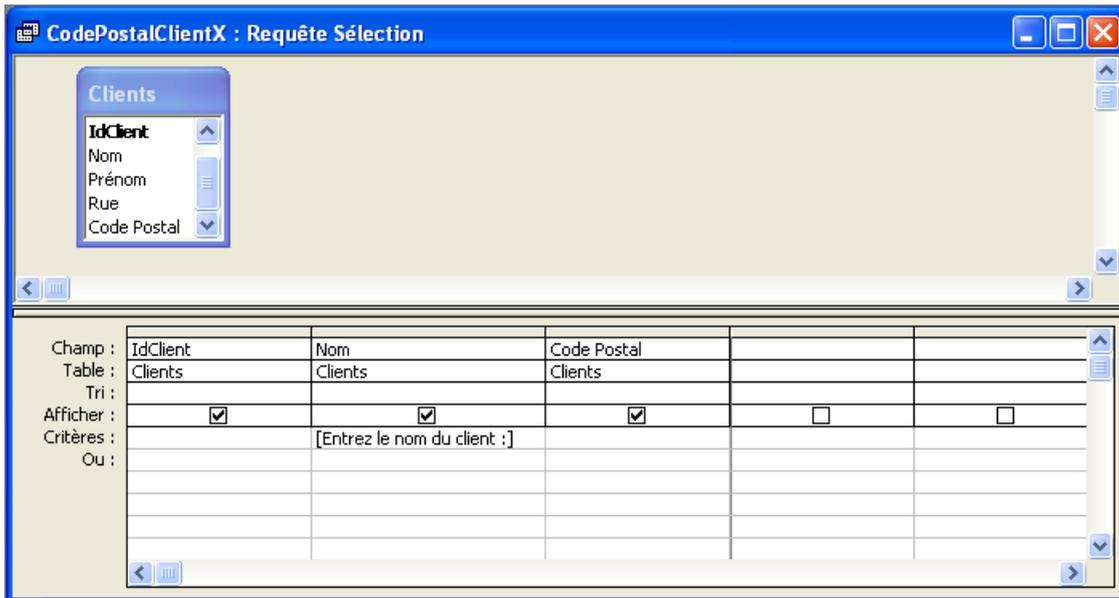
- (1) Créer une requête paramétrée **CodePostalClientX** qui recherche le code postal du client dont le nom est entré par l'utilisateur.
- (2) Créer une requête **AutresClientsDeLaVilleDeX** qui s'appuiera sur la première requête et la table clients.

- 59) Créez une nouvelle requête en mode création et ajoutez la table *Clients*.

Insérez les champs *IdClient* (ce champ n'est pas indispensable à ce niveau, mais il servira pour affiner la requête *AutresClientsDeLaVilleDeX*), *Nom* et *Code Postal*.

Comme critère pour le champ *Nom*, entrez le paramètre suivant : [Entrez le nom du client :] (pour limiter le risque de doublons, vous pouvez entrer un deuxième critère paramétrique pour le champ *Prénom*). Enregistrez la requête sous le nom **CodePostalClientX**.

La requête terminée doit ressembler à ceci :



60) Créez une nouvelle requête en mode création et ajoutez la table *Clients* et la requête *CodePostalClientX*.

Insérez les champs *IdClient*, *Nom*, *Prénom* et *Code Postal* de la table *Clients*.

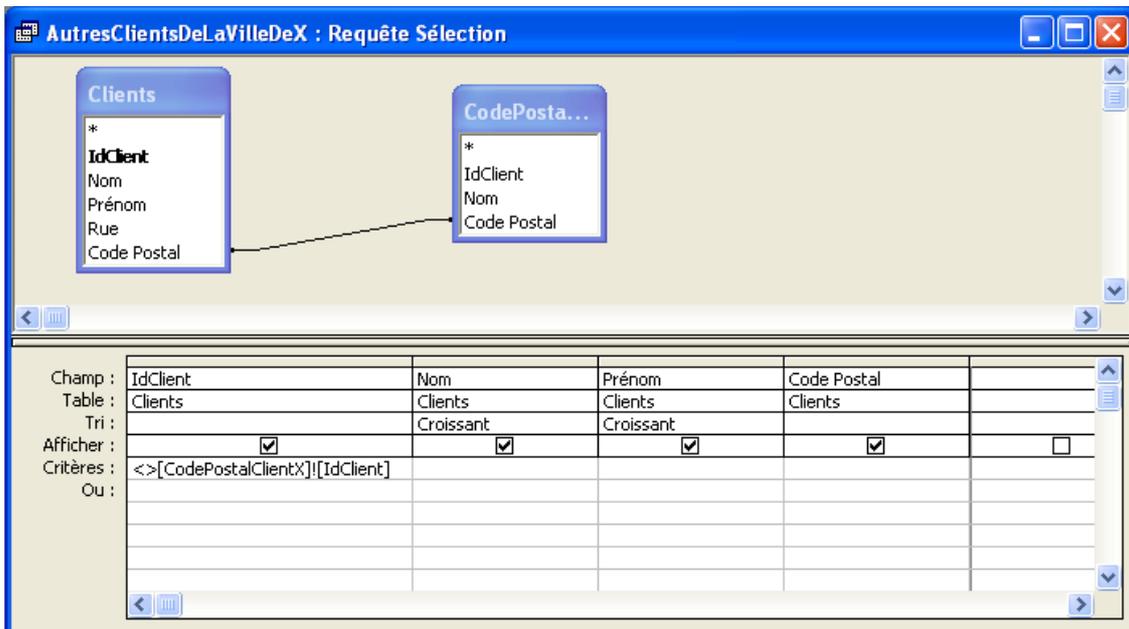
La relation proposée ne nous intéresse pas. Supprimez-la et ajoutez une relation entre les deux champs *Code Postal*.

Enregistrez la requête sous le nom **AutresClientsDeLaVilleDeX**. Vous pouvez la tester avec Sansonner (02000) ou avec Deulisse (59000)

Remarque : si vous ne voulez pas que le client dont vous avez saisi le nom figure dans le résultat de votre requête, ajoutez le critère « <>[CodePostalClientX]![IdClient] » pour le champ *IdClient* (utilisez le générateur d'expression pour éviter les fautes de frappe)

Dans cette requête, le champ *IdClient* intervient deux fois : dans la table *Clients* et dans la requête *CodePostalClientX*. Pour lever l'ambiguïté, il faut indiquer auquel des deux on fait référence. Ainsi [CodePostalClientX]![IdClient] ([nom de la table/requête]![nom du champ]) renvoie au champ *IdClient* de la requête *CodePostalClientX*.

Si vous le souhaitez, vous pouvez aussi trier les clients par ordre alphabétique. La requête terminée devrait ressembler à ceci :



61) Autre exemple : nous souhaitons afficher les autres livres écrits par le(s) auteur(s) d'un livre dont le n° ISBN est X (on pourra tester avec le livre dont le n° ISBN est 2853944603) :

The screenshot displays two query design views in Microsoft Access. The top view, titled "Autres livres de auteur(s) du livre X : Requête Sélection", shows a relationship between three tables: "auteurs", "isbn - n°auteur", and "livres". The "auteurs" table has fields "NumAuteur", "Nom", "Prénom", and "Année de nais". The "isbn - n°auteur" table has fields "ISBN" and "NumAuteur". The "livres" table has fields "ISBN", "Titre", "Lieu Pub", and "IdEditeur". The relationships are: "auteurs" (1) to "isbn - n°auteur" (∞), "isbn - n°auteur" (∞) to "livres" (1), and "auteurs" (1) to "Auteur(s) de ISBN" (∞). Below the design grid, the "Afficher" row shows checkboxes for "ISBN", "NumAuteur", "Titre", "Nom auteurs", and "Prénom auteurs". The "Critères" row contains the criteria: "<>[Auteur(s) de ISBN]![ISBN]" and "[Auteur(s) de ISBN]![NumAuteur]".

The bottom view, titled "Auteur(s) de ISBN : Requête Sélection", shows a query design based on the "isbn - n°auteur" table. The "Afficher" row has checkboxes for "ISBN" and "NumAuteur". The "Critères" row contains the criterion: "[Entrez l'ISBN :]".