

## Initiation à l'Utilisation de l'Informatique 2009-2010

### Introduction à Access 2003 (sur Windows 2000), création des tables

Le contenu de ce document, ainsi que tous les fichiers pédagogiques liés à ce document se trouvent sur le site de notre service MATsch à l'adresse [www.ulb.ac.be/soco/matsch/info-d-203](http://www.ulb.ac.be/soco/matsch/info-d-203). Vous y trouverez aussi des exemples de travaux (projets de bases de données, bases de données, descriptions Word du travail) réalisés par des étudiants des années précédentes.

Ce document reprend les étapes de conception et de création d'une base de donnée relationnelle. Il permet la création des principaux objets de la base de données "9<sup>e</sup> Art". Ce document sert donc de prise de contact avant la construction proprement dite de la base de données.

## Table des matières

1. Introduction aux bases de données
  - 1.1. Objets de la base de données
  - 1.2. Conception ou modification d'une base de données
  - 1.3. Définition de relations entre tables
  - 1.4. Quelques exemples de relations
  - 1.5. Description générale de la base de données
  - 1.6. Quelques exemples d'objets (Requêtes, Formulaires, ...)
    - 1.6.1. Les requêtes
    - 1.6.2. Les formulaires
    - 1.6.3. Les états
    - 1.6.4. Les macros
  - 1.7. En Conclusion
2. Création d'une nouvelle base de données
3. Création des tables de la base de données
  - 3.1. La table Tomes
  - 3.2. La table Clients
  - 3.3. Les autres tables
4. Création des relations
  - 4.1. Les relations " un à plusieurs "
  - 4.2. Les relations " plusieurs à plusieurs "
  - 4.3. Les relations avec une table de "Détails de Ventes"
  - 4.4. La "transitivité" des relations
  - 4.5. Voir les relations de type "un à plusieurs" dans les tables parents
5. Création des requêtes
  - 5.1. Les requêtes de type sélection
    - 5.1.1. Les requêtes à filtre
    - 5.1.2. Les requêtes paramétrées (requêtes paramètre)
    - 5.1.3. Les requêtes avec champs calculés
    - 5.1.4. Les requêtes « opération »
  - 5.2. Les requêtes de type action
    - 5.2.1. Les requêtes Mise A Jour (Update Query)
  - 5.3. Les autres requêtes
6. Création de formulaires
  - 6.1. Création de formulaires classiques
  - 6.2. Création de formulaires à sous-formulaire
  - 6.3. Les autres formulaires
  - 6.4. Quelques propriétés des contrôles dans les formulaires
  - 6.5. Construction d'un formulaire « Menu »
7. Construction des macros
  - 7.1. La macro Ouvrir Encodage Ventes
  - 7.2. La macro Fermer
  - 7.3. La macro Autoexec
  - 7.4. Les autres macros
    - 7.4.1. La macro Ouvrir
  - 7.5. Les actions les plus usuelles
8. Création d'états

- 8.1. Les états publipostages
- 8.2. Les états de regroupements
- 8.3. Les autres états
- 9. L'outil de compactage Access
- 10. Les F.A.Q. (Frequently Asked Questions, La Foire Aux Questions)

#### Fichiers de travail :

<08-9eArt-t.mdb> qui contient les tables de la base de données "9eArt".  
 <08-9eArt-tr.mdb> qui contient les tables et les relations de la base de données "9eArt".  
 <08-9eArt-trr.mdb> qui contient les tables, les relations et les requêtes de la base de données "9eArt".  
 <08-9eArt-trrf.mdb> qui contient les tables, les relations, les requêtes et les formulaires de la base de données "9eArt".  
 <08-9eArt-trrfm.mdb> qui contient les tables, les relations, les requêtes, les formulaires et les macros.  
 <08-9eArt.mdb> qui contient les tables, relations, requêtes, formulaires, macros et états.

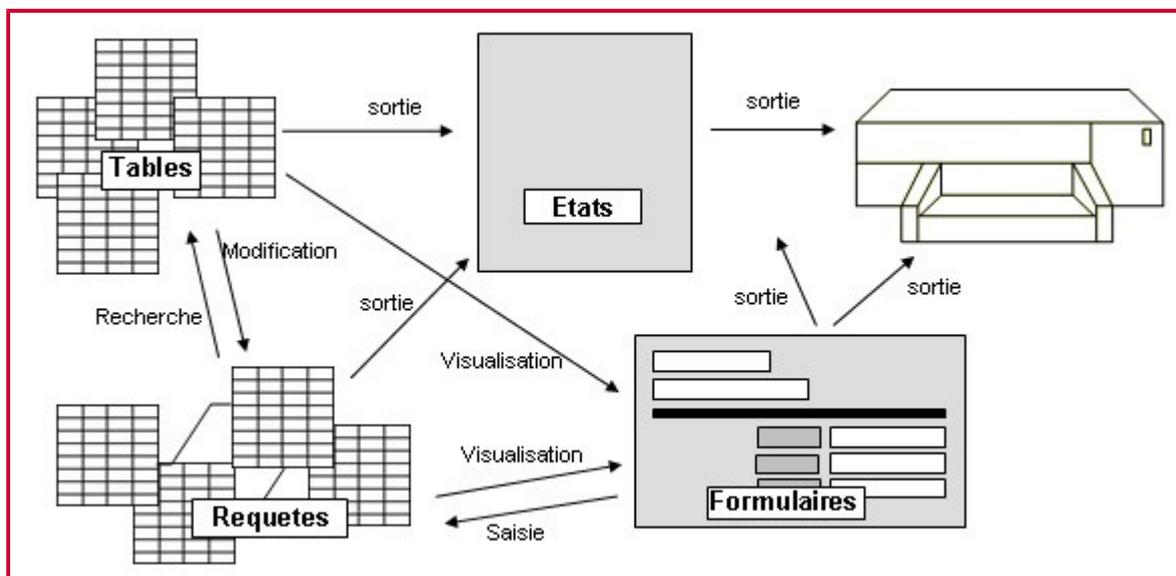
## 1. Introduction aux bases de données

### 1.1. Objets de la base de données

Le logiciel Access contient plusieurs types d'objets. Les objets de la base de données sont des objets qui apparaissent dans la fenêtre Base de données. C'est l'utilisateur qui crée ces objets. Nous utiliserons les objets suivants:

 Table:	Structure fondamentale du système de gestion de bases de données relationnelles. Dans Access, une table est un objet qui stocke des données dans des enregistrements (lignes) et des champs (colonnes). Les données sont généralement relatives à une catégorie particulière, par exemple des contacts dans vos agendas, ou des coordonnées dans le botin.
 Requête:	Une requête permet: <ul style="list-style-type: none"> <li>- de poser une question sur les données stockées dans les tables</li> <li>- d'accomplir une action sur ces données.</li> </ul>
 Formulaire:	Objet de base de données Access dans lequel vous placez des contrôles servant à saisir, afficher et modifier des données dans les champs.
 Etat:	Objet de base de données Access qui présente des informations mises en forme et organisées selon vos spécifications, par exemple <ul style="list-style-type: none"> <li>- des récapitulatifs de ventes,</li> <li>- des listes de numéros de téléphone</li> <li>- des étiquettes pour publipostage.</li> </ul>
 Macro:	Une action ou un ensemble d'actions destinées à automatiser certaines tâches.

Il existe aussi des Pages, pour la mise en commun des données sur réseau Intranet, et des Modules, pour la programmation. Nous ne verrons pas ces deux objets.



Les relations entre les objets, et leurs fonctions, sont représentées de manière simplifiée dans l'illustration ci-dessus. Les macros n'y sont pas représentées, car elles ne remplissent aucune fonction affectant l'organisation d'une base de données Access.

## 1.2. Conception ou modification d'une base de données

Dans une base de données relationnelle, telle qu'une base Access, les données sont regroupées, par entités, dans des tables. Pour qu'Access associe correctement les données réparties dans les tables, les liaisons entre les différents sujets doivent être précisées.

Pour concevoir une base de données

### 1. Définissez l'objectif de la base de données.

C'est au cours de cette phase que vous déterminez l'information que contiendra la base de données: Nos exemples et illustrations sont tirés de <08-9eArt.mdb> qui sera l'exemple type lors du cours. La base de données « 9eArt » se concentre sur la gestion d'un stock de bandes dessinées vendues en magasin spécialisé. Dans le cadre de vos travaux, évitez les sujets "passe-partout" tels que la gestion d'un stock de vins, de disques, de livres, ....

### 2. Définissez les tables.

Divisez l'information en entités principales. Chaque thème fera l'objet d'une table dans la base de données: La gestion de nos bandes dessinées se fait tome à tome. Nous nous restreignons à un ensemble d'environ 250 tomes. Afin de classer ces tomes, nous les regroupons en séries. De même, afin de classer ces séries, nous les regroupons en 8 maisons d'édition dont sont issus les tomes. Nous gérons les sorties d'albums: le tome vendu, la date de vente, et le client. Les données concernant ce client sont entièrement décrites dans la table Clients.

### 3. Définissez les champs.

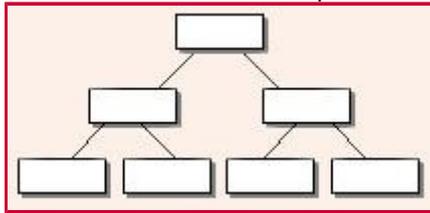
Déterminez les données à stocker dans chaque table. Chaque champ correspondra à une colonne dans la table. Nom et Prénom du client, par exemple, seront deux champs de la table Clients.

### 4. Définissez les relations entre les données et entre les tables.

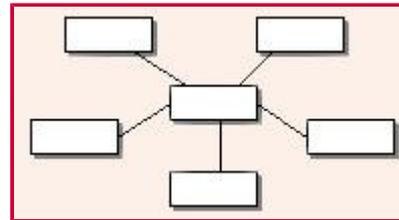
Examinez chaque table de la base de données et déterminez les liaisons qui peuvent exister entre les données des différentes tables. Au besoin, ajoutez des champs aux tables ou créez de nouvelles tables. Dans notre cas, il faut indiquer, pour chaque tome, de quelle série il est issu; pour chaque série, chez quel éditeur elle paraît; pour chaque vente, à quel tome elle correspond, et à quel client elle correspond. Une fois que vous avez créé les tables qui constituent une base de données, vous devez définir des relations entre ces tables. Ces relations indiquent à Access comment les données des différentes tables sont reliées entre elles ; elles facilitent également la création de requêtes, de formulaires et d'états qui incluent des tables multiples.

Les structures de bases de données sont variées. Des cas classiques de structures sont les structures hiérarchiques et structures en étoiles, mais d'autres types sont possibles. Des combinaisons de ces deux structures sont également permises. Méfiez-vous toutefois des relations cycliques: dans le cas général, il n'existe qu'un seul chemin pour passer d'une table à l'autre.

Structure hiérarchique



Structure en étoile



### 1.3. Définition de relations entre tables

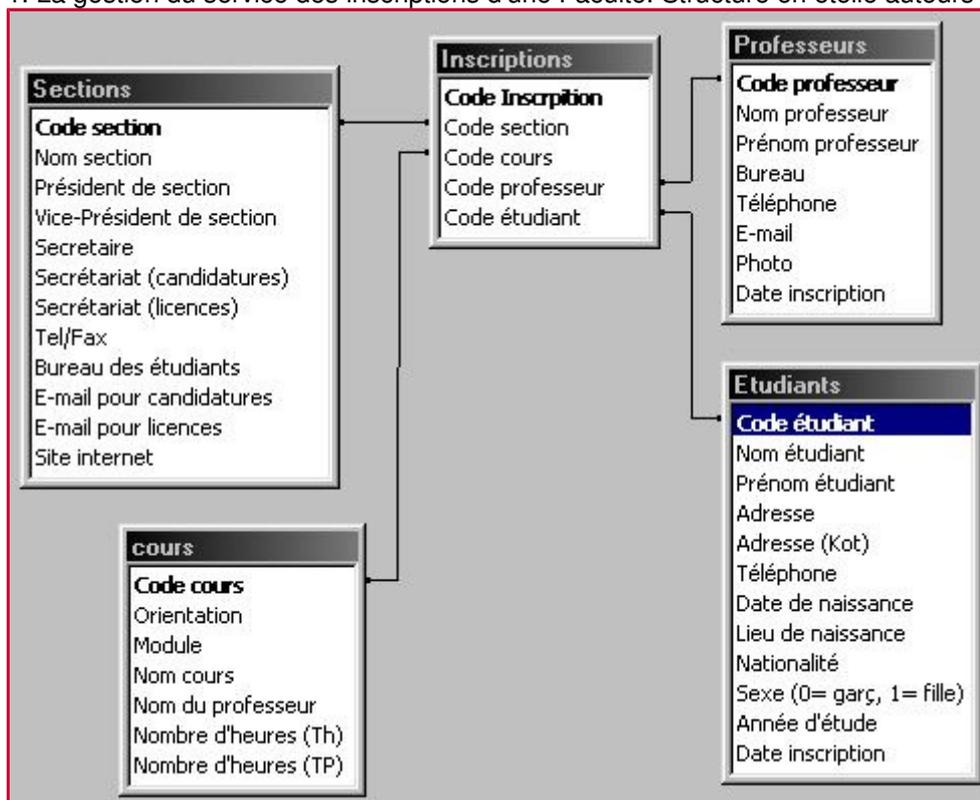
Si, par exemple, vous avez une table Clients et une table Ventas, la définition d'une relation par défaut entre ces deux tables vous fait gagner du temps lorsque vous créez des formulaires ou des états qui relient les informations de ces tables.

Pour définir des relations, vous faites correspondre les champs Clé primaire de la table parent aux champs correspondants de la table enfant (terminologie utilisée dans le cas d'une relation de un à plusieurs).

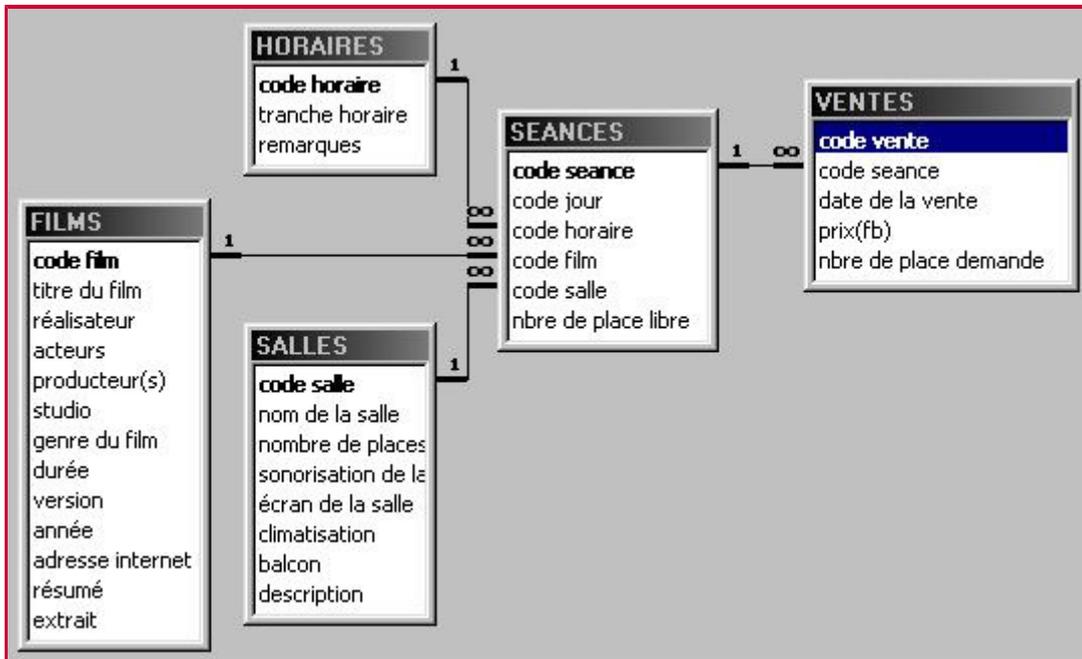
Pour voir les relations définies entre les tables, choisissez dans la barre d'outils le bouton Relations (Relationships) représenté par trois petites tables liées. Access affiche la boîte de dialogue Relations avec les représentations de toutes les tables, ainsi que les relations sous forme de traits entre les tables.

### 1.4. Quelques exemples de relations

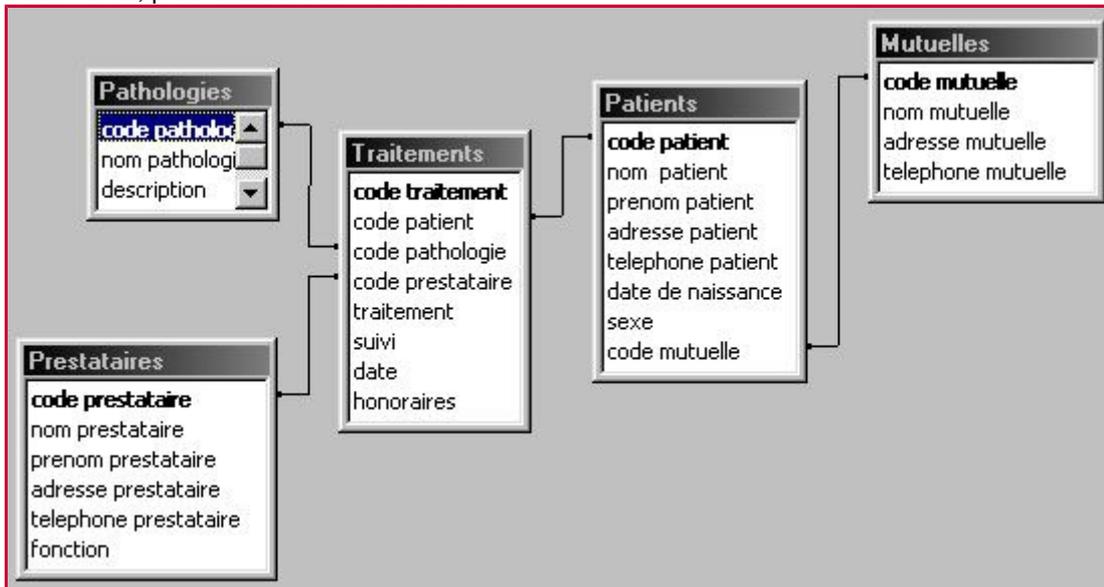
1. La gestion du service des inscriptions d'une Faculté. Structure en étoile autour de la table des inscriptions.



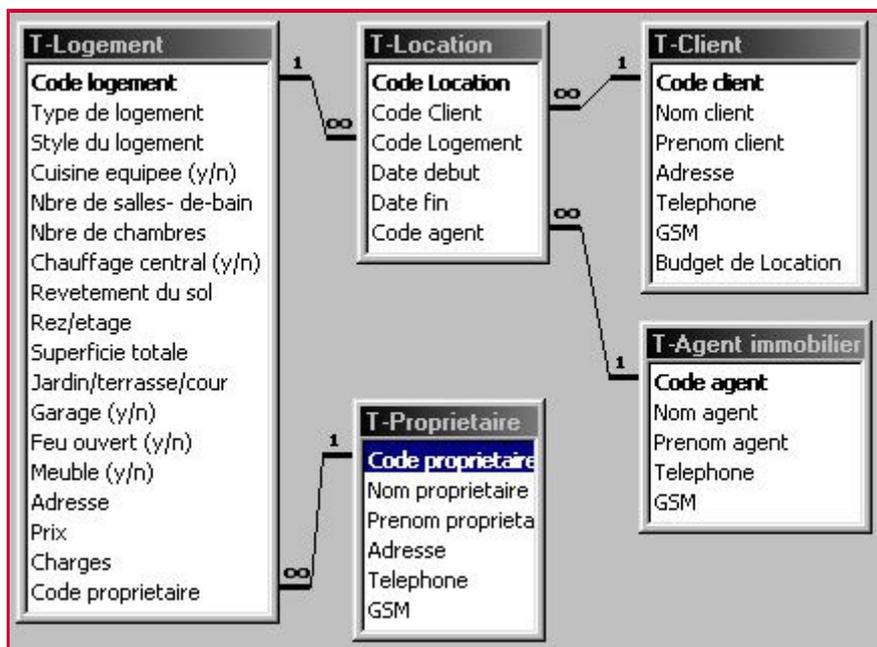
2. La gestion de la distribution des billets de cinéma. Structure en étoile autour de la table des séances, plus la table des ventes



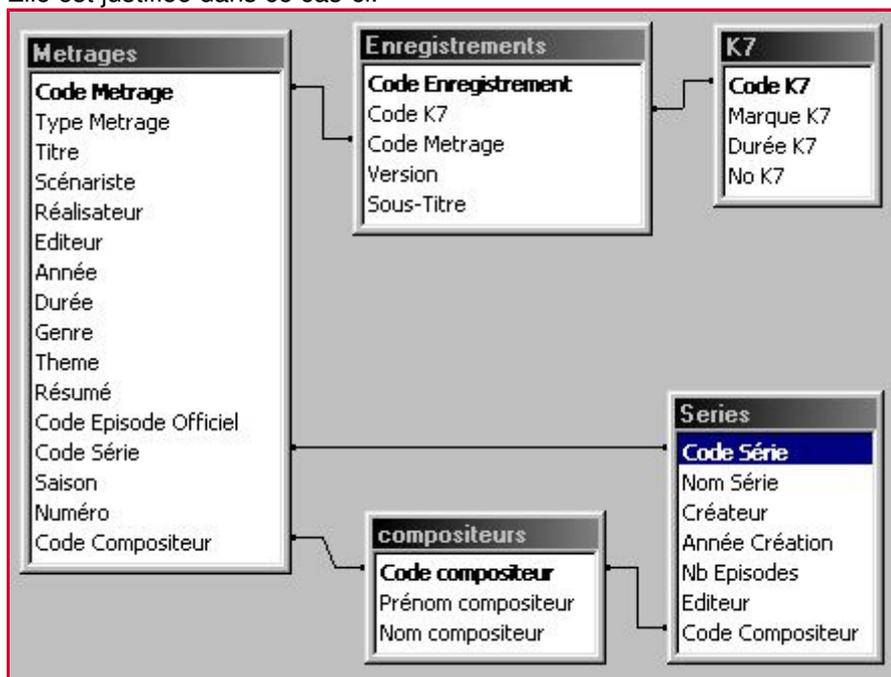
3. La gestion des traitements administrés dans un service hospitalier. Structure en étoile autour de la table des traitements, plus la table des mutuelles.



4. La gestion des locations pour des clients d'une agence immobilière. Structure en étoile autour de la table des locations, plus la table des propriétaires.



5. La gestion d'un stock de cassettes vidéos personnelles reprenant des films, téléfilms et séries. Structure de plusieurs à plusieurs entre les métrages et les cassettes, plus une table reprenant les titres de séries, plus une table reprenant les compositeurs de musiques. Attention à la "relation cyclique" entre Metrages, Series et Compositeurs! Elle est justifiée dans ce cas-ci.



Après avoir créé des tables, vous pouvez créer des requêtes, des formulaires, des états et d'autres objets base de données qui faciliteront l'utilisation des données.

## 1.5. Description générale de la base de données

L'exemple développé est basé sur la gestion d'un stock de bandes dessinées vendues en magasin spécialisé. Il s'agit de la base de données "9eArt" (<08-9eArt.mdb>). On se restreindra à un ensemble d'environ 250 tomes, que l'on appellera aussi parfois « albums », décrits dans la table Tomes. Afin de classer ces tomes, on créera une table Séries reprenant les données relatives aux 31 séries composées par les tomes. De même, afin de classer ces séries, on créera la table Editeurs reprenant les données relatives aux 8 maisons d'édition dont sont issus les tomes. On supposera, pour simplifier l'exemple, qu'une série d'albums n'a été éditée que par un seul éditeur. Pour gérer les ventes d'albums, on disposera d'une table Ventes reprenant le tome vendu, la date de vente, et le client. Les données concernant ce client seront décrites dans la table Clients.

L'utilisateur de la base de données n'est peut-être pas expert dans le maniement d'Access. Ainsi, une série de formulaires faciles d'emploi serviront à encoder les nouvelles séries, nouveaux tomes, nouveaux clients et

nouvelles ventes. L'utilisateur n'a qu'à se laisser guider par les différents boutons, messages, ... Nos formulaires de la base de données sont « Menu des Formulaires », « Encodage Clients », « Encodage Ventes », « Encodage Tomes », « Encodage Séries », « Acheteurs d'un Tome », « Séries d'un éditeur », « Tomes d'une série », « Sous-formulaire1 », « Sous-formulaire2 ».

On proposera une série de requêtes filtrant les tomes selon des critères basés sur le titre, le scénariste, la date de parution,... Des requêtes "regroupement" permettent d'extraire des données numériques à partir des données initiales, et permettront ensuite de créer des graphiques. Un autre type de requêtes, les requêtes action, permettra par exemple de mettre les prix des albums à jour chaque année. Les requêtes de la base de données sont « Rq Ventes », « Rq Van Hamme ou Rosinsky », « Rq Age d'Argent (1955-1970) », « Rq Tomes en Les ... », « Rq Tomes en ... », « Rq Ventes Mois », « Rq Compte Tomes par Série », « Rq Cote Moyenne Série », « Rq Inflation », « Rq Inflation (Annuler) », « Rq Anciens Clients », « Rq Ventes par Client », « Rq Catalogue des Editeurs », « Rq Editeurs », « Rq Séries », « Rq Tomes ».

Le gérant de la librairie aimerait avoir le rapport de ses ventes d'albums mois après mois. Ceci se fera grâce à un Etat approprié. Le gérant disposera d'autres Etats, comme les ventes par client, le catalogue des albums par éditeur, et les étiquettes de publipostage pour les commandes des clients, le double graphique des ventes par mois.

Un souci de présentation mènera à l'utilisation de boutons faisant appel à des macros très simples (ouvrir ou fermer un formulaire, diminuer le nombre d'albums en stock dès qu'un album est vendu, ..).

## 1.6. Quelques exemples d'objets (Requêtes, Formulaires, ...)

### 1.6.1. Les requêtes

Une requête permet de poser une question sur les données stockées dans les tables ou d'accomplir une action sur ces données.

1. Les requêtes sélection  posent des questions sur les données stockées dans les tables et renvoient une feuille de réponses dynamique sans modifier les données. Une fois la feuille de réponse dynamique affichée, vous pouvez examiner et modifier les données. Toute modification sera immédiatement enregistrée dans les tables source.

**Par exemple, Rq Ventes:** Elle reprend les données de la table Ventes et les données correspondantes dans les tables Tomes et Clients. Elle permet ainsi de retrouver sans se préoccuper des codes "qui a acheté quoi" et "ce qui a été acheté par qui".

**Par exemple, Rq " Van Hamme " ou " Rosinski " :** Cette requête affiche tous les tomes écrits par Van Hamme ou dessinés par Rosinski, quelles que soient les séries, quels que soient les éditeurs. Il est aisé de changer le critère et de retrouver les tomes écrits et/ou dessinés par d'autres auteurs.

2. Les requêtes action  modifient ou déplacent les données. Ajout, Suppression, Création de table et Mise à jour sont des requêtes action.

**Par exemple, Rq Inflation:** Cette requête, une fois exécutée, met la table " Tomes " à jour en augmentant (chaque année) les prix de 5%. Cette requête ne doit être utilisée qu'une fois par an sous risque de voir les prix être faussés.

**Par exemple, Rq Anciens Clients:** Cette requête, une fois exécutée, crée une nouvelle table répétant tous les champs de la table Clients, mais ne reprend pour enregistrements que les clients n'ayant plus fait d'achat depuis fin 1999.

3. Les requêtes analyse croisée  effectuent des totaux à partir des valeurs de chaque ligne et de chaque colonne. Ce sont les valeurs de deux champs (généralement) qui servent d'en-têtes de lignes et de colonnes de cette requête.

**Par exemple, Rq AC nombre de ventes par client par série:** On retrouve les séries classées sur les lignes, et les clients classés en colonnes. Les cellules du tableau indiquent le nombre d'achats répartis par séries de chacun des clients.

4. Les requêtes paramètre optimisent la souplesse d'emploi des autres requêtes en vous demandant de

modifier certains critères chaque fois que vous exécutez la requête.

**Par exemple, Rq Tomes en \*\*\* :** Cette requête demande un mot-clé devant apparaître dans tous les enregistrements. Chaque nouvelle ouverture de cette requête permettra d'entrer un nouveau mot-clé.

## 1.6.2. Les formulaires

Il s'agit d'un objet Access dans lequel vous placez des contrôles servant à saisir, afficher et modifier des données dans les champs. Lorsque vous faites appel à un Assistant pour vous seconder dans la création d'un formulaire, celui-ci vous pose un certain nombre de questions pour connaître les caractéristiques du formulaire souhaité, puis construit ce dernier sur la base de vos réponses.

Il existe plusieurs types de formulaires et plusieurs assistants:

1. Colonne Simple: Formulaire ou Etat affiche les valeurs de chaque enregistrement dans une seule colonne. Chaque valeur apparaît sur une ligne différente. Les enregistrements apparaissent les uns après les autres.

**Par exemple, Encodage clients:** Formulaire contenant les renseignements sur un client. Ce formulaire permet d'encoder les nouveaux clients. La photo du client s'insère grâce à la liaison Ole.

2. Tableau: Formulaire qui affiche les valeurs d'enregistrements dans des colonnes. Des intitulés sont affichés en haut de chaque colonne, et chaque ligne représente un enregistrement.

**Par exemple, Acheteurs d'un tome:** Listing des acheteurs d'un tome quelconque.

3. Graphique: Contenant un graphique basé sur des données numériques ou sur traitement de données textuelles.

**Par exemple, Graphe nombre de ventes par client**

4. Formulaire à Sous-formulaire: Les sous-formulaires peuvent servir à afficher une relation un-à-un ou un-à-plusieurs entre les données de tables ou de requêtes. Vous pouvez, par exemple, afficher le nom d'une catégorie de produit dans le formulaire principal et afficher la liste de tous les produits de cette catégorie dans un sous-formulaire.

**Par exemple, Tomes d'une série:** Dans le formulaire principal figurent les données concernant la série sous forme de colonne simple. Dans le sous-formulaire sont listés les tomes de cette série. C'est le champ [Code Série], lien entre les tables Séries et Tomes, qui permet d'afficher les tomes correspondants.

**Par exemple, Sous-formulaire2:** Listing de tous les tomes. Nécessaire comme sous-formulaire pour le formulaire " Tomes d'une série ".

Avec la version 2003 d'Access, il existe aussi des Formulaires Tables Pivots et des Formulaires Graphes Pivots, dont l'aspect se modifie directement selon vos choix de critères de tri, de filtre, de regroupement, ...

Même si l'Assistant n'offre pas exactement la présentation que vous voulez, il est cependant souvent plus rapide de l'utiliser pour créer un formulaire de base que vous modifiez ensuite.

Si l'utilisation d'un Assistant pour créer un formulaire reste la méthode la plus simple, elle risque cependant de ne pas produire la structure souhaitée. Si vous voulez concevoir vous-même un formulaire, créez un formulaire vierge, puis ajoutez les contrôles et les autres éléments dont vous avez besoin.

Vous pouvez demander à Access d'afficher un formulaire chaque fois qu'un utilisateur exécute une action déterminée, qu'il s'agisse d'ouvrir un autre formulaire ou un autre état, de cliquer sur un bouton de commande, ou de changer de champ actif. Un tel formulaire est alors appelé formulaire indépendant et ressemble fort à une boîte de dialogue. Un formulaire indépendant vous permet d'afficher un message ou de demander des informations complémentaires.

Ce formulaire peut être constitué d'une combinaison quelconque de textes, de contrôles ou d'images qui vous permettront d'afficher un message ou de demander à l'utilisateur de taper certaines informations. Par exemple, vous pourriez demander à l'utilisateur de fournir les dates de début et de fin pour un état, ou de fournir certaines données requises si les données contenues dans un champ déterminé ont été modifiées dans un formulaire.

Un formulaire indépendant contient généralement un ou plusieurs boutons de commande qui exécutent des macros, qui engendrent à leur tour d'autres actions à partir des informations saisies dans le formulaire indépendant. Vous pouvez également prévoir des boutons qui proposent à l'utilisateur des options spécifiques.

**Par exemple, Menu des Formulaires:** Certains des formulaires de « 9eArt » ne sont pas destinés à être ouverts directement. Ainsi, ce formulaire de présentation aura pour simple but de proposer de n'ouvrir que les formulaires intéressants: deux de listings, quatre d'encodage. Ce formulaire contient au total six **boutons de commande** renvoyant à la macro Ouvrir.



### 1.6.3. Les états

Lorsque vous faites appel à un Assistant pour vous seconder dans la création d'un état, celui-ci vous pose un certain nombre de questions pour connaître les caractéristiques de l'état souhaité, puis construit ce dernier sur la base de vos réponses. Bien que les noms suivants ne soient pas officiellement utilisés, Access offre plusieurs types d'assistants:

1. **Assistant Colonne simple:** Assistant Formulaire ou Assistant Etat qui sert à afficher les valeurs de chaque enregistrement dans une seule colonne. Chaque valeur apparaît sur une ligne différente apposée d'une étiquette à sa gauche. Les enregistrements apparaissent les uns sous les autres et du haut en bas de la page.
2. **Assistant Regroupements/totaux:** Assistant Etat qui sert à organiser les données en groupes affichés sous forme de tableau. Vous pouvez calculer la somme de chaque groupe et celle de tous les groupes. Des intitulés apparaissent en haut de chaque colonne de données.

**Par exemple, Etat Catalogue des Editeurs:** basé sur la requête Rq Catalogue des Editeurs.

Les enregistrements sont regroupés selon l'éditeur, puis selon la série. Le tri se fait ensuite sur le champ [N°Tome].

3. **Assistant Publipostage:** Assistant Etat qui sert à créer des étiquettes de publipostage.

**Par exemple, Etat Etiquettes:** basé sur la table Clients. Cet état permet d'envoyer des colis contenant la commande des clients grâce à l'impression d'étiquettes personnalisées.

**Par exemple, Etat Etiquettes Paramètre:** Le même état, mais ne propose que l'étiquette d'un client sélectionné (basé sur une requête paramètre).

4. **Assistant Graphique:** Assistant qui crée un état contenant un graphique.

**Par exemple, Etat Graphique des Ventes par Mois:** basé sur la requête Rq Ventes. Cet état présente en vérité deux graphiques. Il propose une représentation graphique des ventes par mois, sous forme de nombre de tomes vendus, et sous forme de chiffre d'affaire.

Si l'état créé par l'Assistant ne correspond pas exactement à ce que vous désirez, vous pouvez le **modifier** en mode Création ou recommencer l'opération en répondant différemment à l'Assistant.

Les Assistants facilitent grandement la création des états, mais ils n'offrent pas la méthode de création la plus flexible. Pour construire un état de toutes pièces, vous **créez un état vierge**, y ajoutez les niveaux de tri et de regroupement voulus, puis y placez les contrôles et les autres éléments de structure.

### 1.6.4. Les macros

Une macro est une liste d'instructions que vous créez dans le but d'exécuter automatiquement certaines tâches.

Dans Access, chaque instruction s'appelle une action. Lorsque vous exécutez la macro, Access réalise les actions, dans la séquence indiquée, en utilisant les objets et les données que vous avez spécifiés en tant qu'arguments des actions. Les macros servent à automatiser les tâches répétitives et à enrichir les possibilités de votre base de données.

Les actions sont déjà définies. Il suffit de les choisir dans la liste de propositions. Dans la partie inférieure de la fenêtre, spécifiez les arguments de l'action.

### Par exemple, La macro Fermer:

Elle permet de fermer le formulaire " 1: Menu " et de retourner à la fenêtre Base de données.

action	
Fermer	<b>type objet:</b> Formulaire <b>nom objet:</b> 1: Menu Formulaires

## 1.7. En Conclusion

Malgré l'aspect très « construction » qu'impose la création de la base de donnée, il n'a été utilisé aucune notion de programmation. Vous serez capable, et serez tenu, d'ici la fin du cours, de créer et remettre une base de donnée semblable, selon les critères définis, à celle proposée ici.

La mise en forme de la base de donnée est telle que la base est utilisable, d'une part par ses concepteurs ou par des connaisseurs d'Access, d'autre part, et plus intéressant, par des utilisateurs ignorant tout d'Access. Ceci est dû aux formulaires indépendants, aux liste modifiables, aux messages d'informations, ...

## 2. Création d'une nouvelle base de données

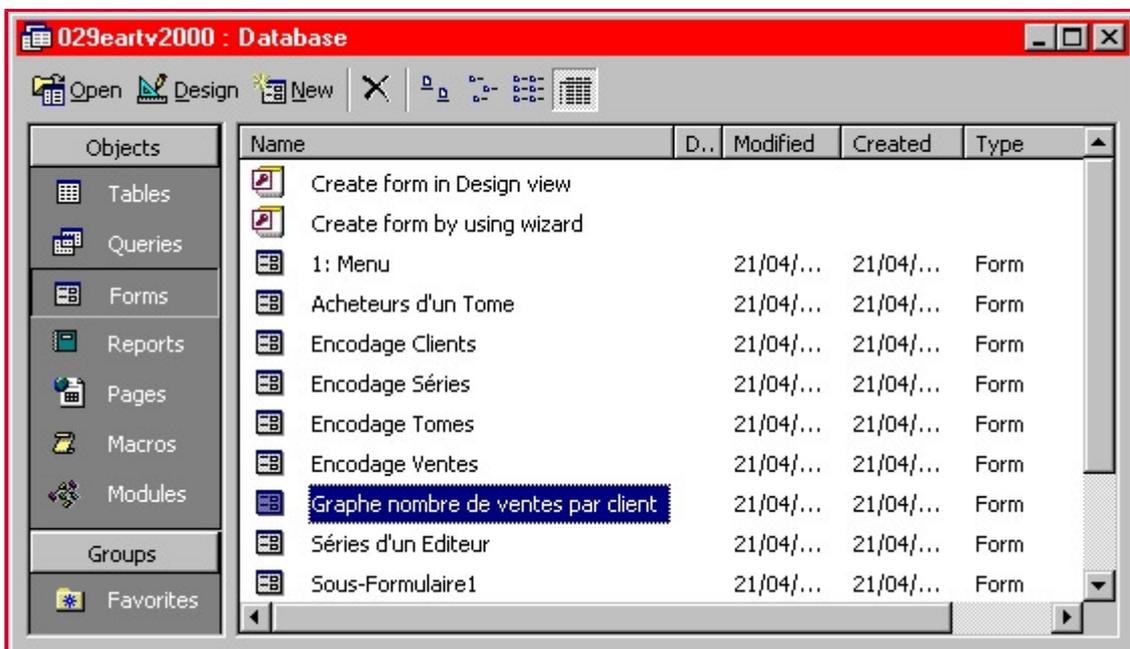
La première chose à faire est bien entendu de lancer Access par le menu Démarrer (Start). Access est le logiciel représenté par une clé mauve/rose.

Attention, il est possible d'obtenir au lancement d'Access un message d'avertissement prévenant que certaines expressions pourraient permettre à certains virus de pénétrer les bases de données Access. Le message est du type "Warning, unsafe expressions are not blocked. Do you want to block unsafe expressions?" Il suffit de choisir "No" pour continuer à travailler avec toutes les fonctionnalités d'Access. Lorsque la machine demande la confirmation de l'ouverture du fichier, "Do you want to open this file?", il faut confirmer par "Yes", et/ou par "Open".

Pour supprimer ces messages d'avertissements, et rendre moindre le niveau de sécurité d'Access, passez par le menu Outils/Macro/Sécurité (Tools/Macro/Security). Mettez le niveau de sécurité sur Bas (Low).

Office n'affiche par défaut que les commandes les plus souvent utilisées dans les menus. Les commandes moins utilisées peuvent être obtenues en attendant quelques instants dans le menu, ou en cliquant sur la flèche vers le bas dans la partie inférieure du menu. Cette option ralentit le travail de celui qui utilise souvent les menus. Il est possible pourtant d'afficher immédiatement toutes les commandes du menu en se rendant dans le menu Otions/Personnaliser (Options/Customize) en choisissant l'onglet Options, et en cochant la case "Always Show Full Menus". Cette manipulation est valable pour les logiciels Access, Excel, Word, PowerPoint.

Une fois Access complètement lancé, on choisit soit le bouton Nouveau Document de la barre d'outils, soit la commande Créer un Nouveau Document (Create New Document) dans le volet de droite. On choisit ensuite de créer une nouvelle base de données (Blank Database) dans le menu à droite. Access demande un nom de fichier pour la base. En prévision de la suite du travail, on proposera <08-9eArt0.mdb>. Ne pas oublier de créer ce fichier dans le dossier de travail (Save In: H:\DossierPersonnel). Plutôt que d'entrer le chemin soi-même, il est possible de voyager dans l'arborescence et de retrouver ce dossier personnel. Lorsque cela est fait apparaît la *fenêtre base de données*, l'objet qui caractérise en soi toute la base de données.



### 3. Création des tables de la base de données

La première chose à faire est de fournir à Access un minimum de données. Ces données seront mémorisées dans de grands tableaux, appelés tables. En appuyant sur le bouton Nouvelle (New), on nous propose de construire une nouvelle table en mode feuille de données (Datasheet View), en mode création (Design View), à l'aide d'un Assistant (Table Wizard), d'importer une table (Import Table) ou de lier une table d'une autre base de données (Link Table). On choisit le mode création (Design View), permettant de personnaliser entièrement notre travail.

#### 3.1. La table Tomes

Dans la suite de ce texte, les noms de champs sont indiqués entre crochets, et les mots techniques propres à l'informatique sont en italique.

La première *table* que l'on construit (en mode création (Design View)) est la table des **Tomes**. Elle doit contenir les données relatives aux albums dont dispose le vendeur. Dans la colonne Champ (Field Name) on entre successivement les champs [N° Tome], [Titre Tome], [Scénariste], [Dessinateur], [Année Parution], [Prix], [Couverture]. Si on précise à quelle série appartient l'album, on aura des répétitions (des dizaines de Tintin, des dizaines d'Astérix, ...). On réservera une table pour ces séries, et pour les données y référant. Pour l'instant, il suffira d'un code de série de trois lettres qu'on encodera dans le champ [Code Série].

Cette notion de code s'avère utile pour la mise en relation des tables, et pour l'utilisation de la clé primaire. La clé primaire est un champ ou une combinaison de champs qui désigne, de façon unique, chaque enregistrement dans une table. Elle fait office d'index principal de la table tout en permettant d'associer des données appartenant à des tables différentes. Notre clé primaire sera le [Code Tome] que l'on introduit en première position dans notre table grâce au bouton d'insertion de ligne. On y définit la clé primaire en sélectionnant la ligne de ce champ (en cliquant dans la colonne-marge) et en cliquant sur le bouton représentant une clé. 

On remarque que la machine a automatiquement choisi le type de données texte pour nos champs. On modifie le [N° Tome] par un type numérique (Number), l' [Année Parution] par un type Date/Heure (Date/Time), le [Prix] par un type Monétaire (Currency), et la [Couverture] par Liaison OLE (OLE Object). La Liaison OLE permet d'insérer dans le formulaire approprié une "miniature" (thumbnail) de la couverture.

Chacun de ces types peut être encore détaillé. Le champ [N° Tome] et le champ [Nb Exemplaires] sont choisis de format Octet (Byte). Leurs valeurs sont comprises entre 0 et 255. Ceci permet de ne pas perdre inutilement de la place en mémoire. Le champ [Année Parution] est choisi de format yyyy (en français aaaa), ne laissant ainsi paraître que l'année de parution (par exemple, 9/12/96 s'affiche comme 1996). Le champ [Prix] est choisi de format # " € ", faisant apparaître le nombre suivi du symbole " € " (Ce symbole est obtenu par la combinaison de touches [Alt Gr]+[e]). Ou alors on choisit le format "Euro" dans la liste prédéfinie, mais le symbole est alors affiché devant le montant. La taille (Field Size) du champ [Code Série] est réduite à 3 caractères. La taille du champ [Code Tome] est réduite à 6 caractères. On y entrera les trois lettres du code de série, et trois chiffres rappelant le numéro de l'album. La Clé Primaire (Primary Key) est sur le champ [Code Tome].

Concernant les formats de dates, voici les interprétations actuelles des formats abrégés pour les années.

Du 1/1/00 jusqu'au 12/31/29 est interprété comme du 1/1/2000 jusqu'au 12/31/2029

Du 1/1/30 jusqu'au 12/31/99 est interprété comme du 1/1/1930 jusqu'au 12/31/1999

Il est vivement conseillé d'entrer les dates sans abréviations. Par ailleurs, bien que le format soit yyyy, il faudra encoder les années sous la forme 1/1/1992. Une date telle que 1992 (sans précision de jour et de mois) ne sera pas acceptée.

On affecte une valeur par défaut (Default Value) au champ [Date Parution] dans le but de minimiser l'encodage des dates de parution des nouveaux albums parus. Le générateur d'expressions permet de construire ce genre de données. Pour obtenir le générateur d'expressions, on clique au niveau de la propriété *valeur par défaut*, puis sur le bouton à points de suspensions. Dans l'« arborescence » proposée, on choisit successivement Fonctions, Build-in-Fonctions, Date/Time, Date. En cliquant sur le bouton Coller (Paste), la fonction Date(), renvoyant la date actuelle, s'inscrit dans la partie supérieure de la fenêtre (vérifier que rien d'autre n'apparaît). On clique sur OK pour confirmer cette valeur par défaut.

Après avoir encodé une description dans la colonne du même nom, la table apparaît sous la forme suivante.

Champ	Type de champ	Description
Code Tome	Texte	Code représentant la série et le numéro de tome. exemple: tin001
N° Tome	Numérique	
Titre Tome	Texte	
Scénariste	Texte	
Dessinateur	Texte	
Année Parution	Date/Heure	
Prix	Monétaire	
Couverture	Liaison OLE	
Nb Exemplaires	Numérique	
Code Série	Texte	

En appuyant sur le bouton feuille de données (premier bouton de la barre d'outils), la table passe en mode feuille de données, et est prête pour un premier encodage. A noter que lors de ce premier passage d'un mode à l'autre, Access demande sous quel nom sauver la table. Le nom de la table est Tomes. Notons aussi que le bouton feuille de données est alors remplacé par le bouton création. Pour faciliter l'encodage, on rétrécit les colonnes [Code Tome], [N° Tome] et [Année Parution] en cliquant-glissant sur les séparations de colonnes (lorsque le pointeur a pris la forme d'une double flèche).

La méthode utilisée pour l'encodage consiste à saisir un nombre suffisant d'enregistrements (une dizaine) dans chaque table pour vérifier que les objets que l'on construira (formulaires, requêtes, ...) fonctionnent convenablement. Le reste de l'encodage se fera à partir des formulaires.

Notons entre autres choses que les champs numériques contiennent déjà une valeur « 0 », que la date par défaut est bien présente, ainsi que le symbole "€" pour le prix. L'encodage nous confrontera au problème suivant: une date de parution comme 1950 sera refusée car il ne s'agit pas d'une date complète. Les dépôts légaux de parution de bandes dessinées se font (souvent) par trimestre. Le mieux est alors d'entrer des dates comme 1/1/1950, 1/4/1950, ... Le résultat affiché sera le même (à savoir 1950) mais ce type de données pourrait s'avérer utile pour la construction de certains états (un regroupement par trimestre par exemple).

Voici quelques exemples d'encodage. On laisse le champ [Couverture] vide jusqu'à l'encodage dans les formulaires.

Code	N°	Titre Tome	Scenariste	Dessinateur	Date	Prix	Couverture	Nb	Code Série
ast001	1	Astérix le Gaulois	Goscinny	Uderzo	1961	350 FB		18	ast
ast002	2	La Serpe d'or	Goscinny	Uderzo	1962	300 FB		20	ast
ast003	3	Astérix et les Goths	Goscinny	Uderzo	1963	300 FB		20	ast
tin001	1	Tintin au Congo	Hergé	Hergé	1946	400 FB		19	tin
tin002	2	Tintin en Amérique	Hergé	Hergé	1945	400 FB		20	tin
tin003	3	Les Cigares du Pharaon	Hergé	Hergé	1955	350 FB		20	tin

### 3.2. La table Clients

Cette table contient les coordonnées et informations sur les acheteurs potentiels. Les étudiants construisent cette table de manière plus autonome que la précédente. Le principe est le même. La Clé Primaire est placée sur le champ [Code Client]. La taille (Size Field) du champ [Code client] est réduite à trois caractères. Le numéro de téléphone est de type texte ; cela laisse une plus grande liberté pour le format si on désire le modifier.

Champ	Type de champ	Description
Code Client	Texte	première lettre du prénom + 2 premières lettres du nom
Nom Client	Texte	
Prénom Client	Texte	
Adresse Rue	Texte	
Adresse No	Texte	
Adresse CP	Texte	
Adresse Commune	Texte	
Téléphone	Texte	

Lors du passage en mode feuille de données, la table est sauvée sous le nom « Clients ».

Voici quelques exemples d'encodages.

Code Client	Nom Client	Prénom Client	Téléphone
aaa	Aaron	Albert	0474 34 56 78
bba	Ballard	Bernadette	0478 45 67 89
cca	Calais	Caroline	0495 22 33 44

### 3.3. Les autres tables

Les autres tables de la base de données se construisent de la même façon. Il est donc inutile de passer à la création de chacune d'elles puisque le fichier <08-9eArt-t.mdb> les contient toutes. A ce stade, on ferme <08-9eArt0.mdb> et on ouvre <08-9eArt-t.mdb>. Ce fichier est complet au niveau de la structure des tables, ce qui permet aux étudiants « perdus » ou « en retard » de se remettre à jour. En outre, <08-9eArt-t.mdb> contient un nombre d'enregistrements suffisant, fournissant pour la suite du travail des résultats intéressants. La description des tables que l'on n'a pas construites est présentée ci-dessous.

Une visite rapide de chacune des tables permet d'avoir une idée des données avec lesquelles on va travailler. Surtout concernant la table Ventes qui fait appel à deux clés primaires. La transformation de <08-9eArt-t.mdb> va consister à lui appliquer des relations.

La table **Ventes** est « reliée » (cf paragraphe suivant sur les relations) aux tables Tomes et Clients et gère les ventes des albums:

Champ	Type de champ	Description
Code Vente	Autonumber	
Code Client	Texte	
Code Tome	Texte	
Date Vente	Date/Heure	

Le champ [Code Client] établit le lien avec la table Clients, et le champ [Code Tome] établit le lien avec la table Tomes. On étudiera le rôle de cette table dans les relations décrites ci-dessous.

La table **Clients (Archive)** reprend les mêmes champs que dans la table Clients, mais uniquement pour les clients n'ayant plus fait d'achat depuis fin 1999. Cette table n'est pas construite par le concepteur, mais par une requête de type action que nous étudierons plus loin.

La table **Séries** contient les données relatives aux 31 séries publiées.

Champ	Type de champ	Description
Code Série	Texte	3 premières lettres du nom de la série
Série	Texte	
Créateur	Texte	
Date Création	Date/Heure	
Description	Memo	
Code Editeur	Texte	Code de l'éditeur publiant la série

Le format du champ [Date Création] est lui aussi yyyy. Le champ [Code Editeur] permet de faire le lien avec la table " Editeurs ". Le nombre de caractères du champ [Code Série] est réduit à 3. La Clé Primaire est sur le champ [Code Série].

La table **Editeurs** contient les données relatives aux 8 éditeurs considérés:

Champ	Type de champ	Description
Code Editeur	Texte	3 premières lettres du nom de l'éditeur
Editeur	Texte	
Adresse	Memo	
Téléphone	Texte	
Responsable	Texte	
URL	Lien Hypertexte	

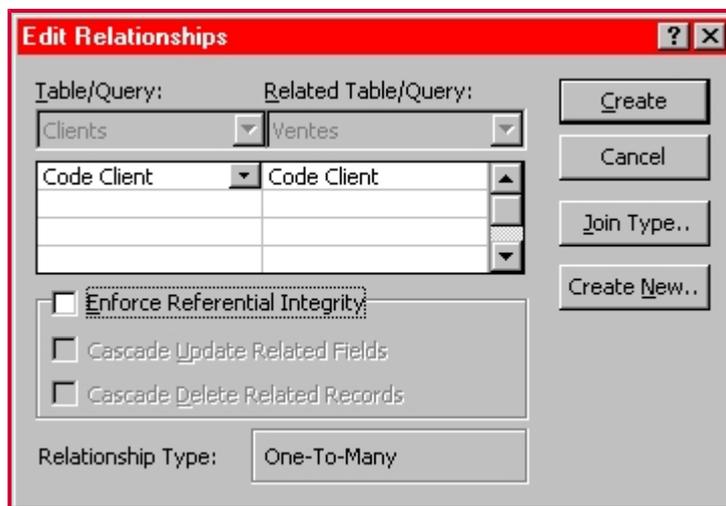
Le nombre de caractères admis pour le type texte a été modifié de 50 à 25 caractères. Pour le champ [Code Editeur], ce nombre a été réduit à 3. La Clé Primaire est sur le champ [Code Editeur]. Un lien hypertexte permet d'aller visiter le site de l'éditeur.

## 4. Création des relations

Une relation est une connexion établie entre des champs communs à deux tables. Une relation peut être du type un-à-un, un-à-plusieurs ou plusieurs-à-plusieurs. La création des relations de notre base a commencé lorsque l'on a décidé de mettre les données relatives aux séries dans la table Séries et d'utiliser un champ [Code Série] dans la table Tomes.

Les relations sont créées grâce à la commande Relations (Relationships) du menu Outils (Tools), ou en appuyant

sur le bouton représentant trois tables liées par deux traits.  Une fenêtre grise apparaît. Pour ajouter des tables aux relations, il faut cliquer sur le bouton représentant une table et un "+" jaune (Show Table). On ajoute alors les différentes tables à lier en double-cliquant sur chacune. Dans notre cas, il s'agit dans l'ordre des tables Editeurs, Séries, Tomes, Ventes et Clients. Les relations à instaurer sont de type " un à plusieurs ". Elles se créent en cliquant-glissant du champ de la table contenant l'enregistrement « parent », jusqu'au champ équivalent dans la table contenant les enregistrements « enfants ». Access demande une confirmation avant de créer la relation. On confirme en cliquant sur le bouton Créer (Create).



Access propose avant la confirmation de choisir l'intégrité référentielle (Referential Integrity). Il s'agit de règles à suivre pour maintenir les relations définies entre les tables lorsqu'on ajoute ou supprime des enregistrements. Si on applique l'intégrité référentielle, Access empêche

- d'ajouter de nouveaux enregistrements dans une table reliée pour laquelle il n'existe pas d'enregistrement « parent »,
- de procéder à la modification de valeurs dans une table source qui entraînerait des enregistrements orphelins dans une table reliée,
- de supprimer des enregistrements d'une table source dans le cas où il existe des enregistrements « enfants » correspondants.

Les différentes relations entre les tables sont motivées comme suit.

#### 4.1. Les relations " un à plusieurs "

Une relation un-à-plusieurs est une relation entre des tables selon laquelle chaque enregistrement de la table source peut être associé à plusieurs enregistrements de la table reliée (chaque valeur de clé primaire peut figurer plusieurs fois dans la table reliée).

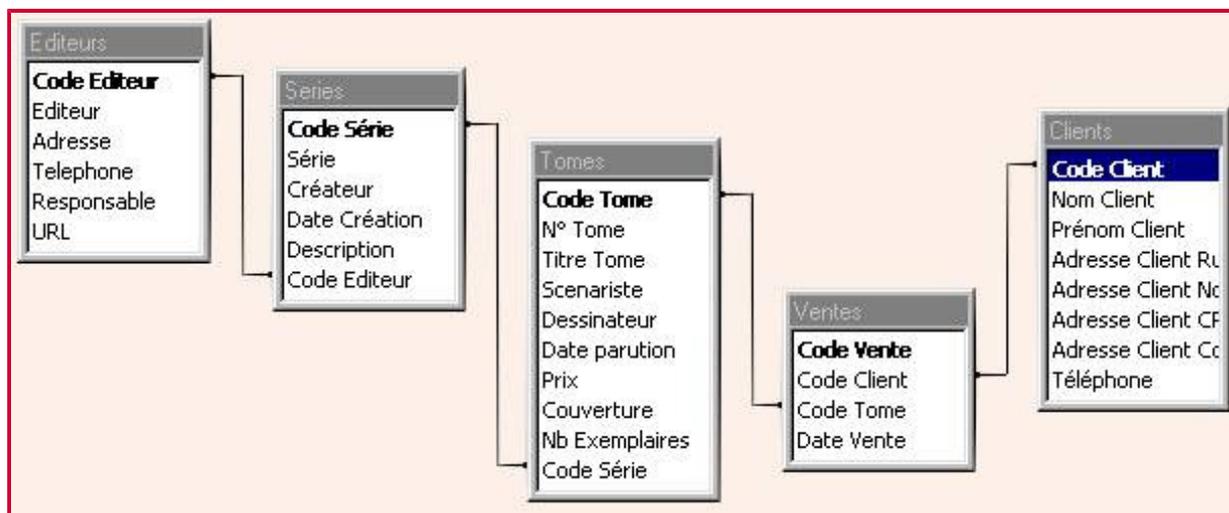
- Un éditeur publie différentes séries Exemple: Casterman publie Tintin, Le Chat, ... On crée une relation " un à plusieurs " entre les tables Editeurs et Séries grâce au champ [Code Editeur].
- Une série est composée de plusieurs albums. Exemple pour Tintin: Tintin au Congo, Tintin en Amérique, ... On crée une relation " un à plusieurs " entre les tables Séries et Tomes grâce au champ [Code Série].
- Un album pourra être vendu en plusieurs exemplaires. On crée une relation " un à plusieurs " entre les tables Tomes et Ventes grâce au champ [Code Tome].
- Un client pourra effectuer plusieurs achats. On crée une relation " un à plusieurs " entre les tables Clients et Ventes grâce au champ [Code Client].

#### 4.2. Les relations " plusieurs à plusieurs "

Dans une relation de type plusieurs-à-plusieurs entre deux tables, les enregistrements de chaque table peuvent être associés à plusieurs enregistrements de l'autre table. Pour définir une relation plusieurs-à-plusieurs entre deux tables, on doit créer une troisième table, appelée *table de jonction*, qui relie les deux premières. La table de jonction doit contenir les clés primaires des deux autres tables, et peut contenir de nouveaux champs.

- Les deux relations entre Tomes & Ventes et entre Clients & Ventes permettent de définir un autre type de relation: la relation " plusieurs à plusieurs " entre Tomes et Clients. Grâce à cette relation, il est aisé de déterminer tous les acheteurs d'un même album, ainsi que tous les albums vendus à un même client. La table Ventes sert de " jonction " entre les tables Tomes et Clients. Mise sous forme de formulaire, cette table Ventes est donc intéressante pour le libraire qui gère ses stocks grâce à elle.

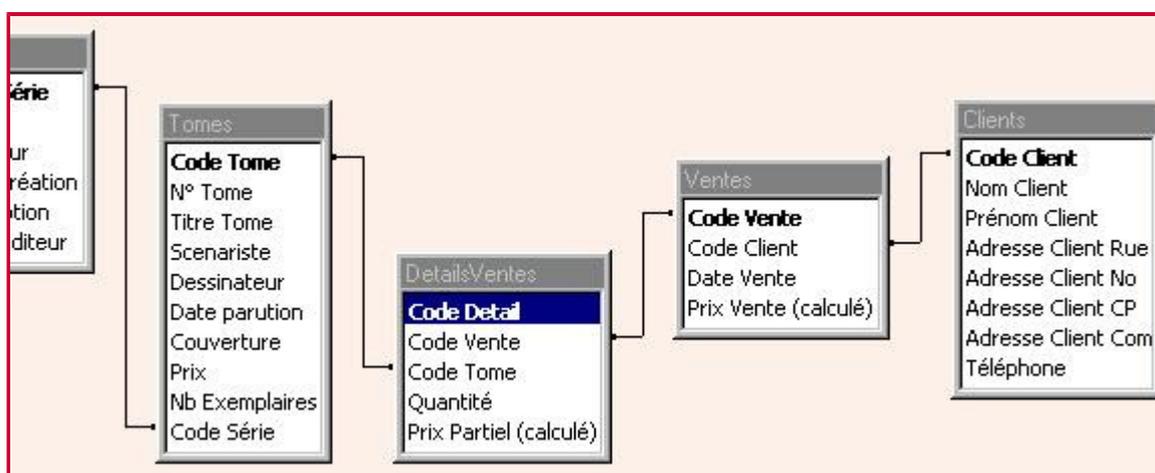
Voici une représentation schématique de nos relations.



### 4.3. Les relations avec une table de "Détails de Ventes"

Nous vous proposons ci-dessous un schéma de la structure de la base de données dans le cas où nous détaillons les ventes. La table des ventes telle que nous l'avons conçue ne permet en effet la vente que d'un album à la fois. Le formulaire des ventes doit être utilisé plusieurs fois si un client achète plusieurs albums, ou plusieurs exemplaires d'un même album. Le nom du client, en particulier, doit être à chaque fois ré-encodé.

La structure que nous proposons ci-dessous corrige ces maladrotes. Les ventes sont distinguées selon deux tables: la table Ventes et la table DetailsVentes. La table Ventes ne reprend plus que le Code Vente (clé primaire), le Code Client (encodé une seule fois si le client achète plusieurs albums), la date de la vente, et le Prix de Vente (calculé par Access comme la somme des prix partiels provenant de la table des détails). La table DetailsVentes reprend les champs Code Détail (clé primaire), Code Vente, Code Tome (pour les liens avec les tables Ventes et Tomes), la quantité de chaque album acheté (mise à "1" par défaut), et le Prix Partiel (calculé par Access comme le produit du prix du Tome (provenant de la table Tomes) et de la quantité (encodée lors de la vente)).



C'est à présent la table DetailsVentes qui sert de jonction entre les tables Tomes et Ventes. Il y a donc une relation "plusieurs à plusieurs" entre ces deux tables. Par extension, nous pouvons aussi dire qu'il existe une relation "plusieurs à plusieurs" entre Tomes et Clients.

Ce type de structure se retrouve souvent dans des bases de données de type commercial. Pensez à vos tickets de supermarchés ou autres factures: chaque ligne du ticket correspond à un enregistrement de la table DetailsVentes (l'objet, sa quantité, le prix partiel, ...), et l'en-tête et le pied du ticket correspondent aux données d'un enregistrement de la table Ventes (éventuellement le nom du client, la date, le prix total, ...).

En remarque, les champs [Prix Partiel] et [Prix Vente] ne sont pas nécessaires. Ils peuvent être calculés à tout moment par Access (à l'aide d'une requête par exemple) à partir du prix du tome et de la quantité achetée (en fonction du client, en fonction de la date, ...). Des états ou des formulaires qui récapituleraient les ventes d'un client ou les ventes par trimestres (par exemple) pourraient être conçus sans l'existence de ces champs. Leur présence n'est cependant pas interdite. Elle facilitera peut-être la création des états ou formulaires en question.

Nous laissons au lecteur le soin de chercher les "trucs" (requêtes, macros, calculs à partir des champs de formulaires, ...) qui permettent de mettre à jour les valeurs de ces champs calculés.

Par ailleurs, une macro de "9eArt" enlève un exemplaire du stock à chaque vente d'un album. Avec cette nouvelle

structure, la macro devrait être adaptée pour décompter le nombre d'exemplaires vendus lors de chaque vente.

#### 4.4. La "transitivité" des relations

Si une première table est liée à une seconde par une relation "un à plusieurs", et que cette seconde table est liée à une troisième par une relation "un à plusieurs", il n'est pas nécessaire de créer explicitement une relation "un à plusieurs" entre la première et la troisième.

Par exemple, imaginons un base contenant les tables Continents, Pays, Villes. Les relations "un à plusieurs" sont clairement établies entre Continents et Pays, et entre Pays et Villes. Il est clair qu'un continent contient également plusieurs villes. Toutefois, Access ne demande pas de matérialiser cette dernière relation. Elle sera prise en compte, implicitement, par l'intermédiaire des deux relations Continents/Pays et Pays/Villes.

En application, dans notre base "9eArt", un éditeur publie de nombreux tomes. Nous n'avons toutefois pas créé de lien entre les tables Editeurs et Tomes, car cette relation ("un à plusieurs") est prise en compte par l'intermédiaire de la table Séries.

#### 4.5. Voir les relations de type "un à plusieurs" dans les tables parents

Lorsque vous êtes dans une table parent, chaque enregistrement est précédé d'un signe "plus". Lorsque vous cliquez sur ce signe, des enregistrements supplémentaires s'affichent. Ces enregistrement sont les enfants de l'enregistrement parent sélectionné. Ce mode de visualisation permet d'avoir une vue d'ensemble plus large sur les données et les relations entre deux tables. Ce mode s'apparente à celui des formulaires à sous-formulaires.

Si une table est le parent de deux autres tables, Access vous demande dans une boîte de dialogue, lorsque vous cliquez sur le signe "plus", de quelle table enfant vous voulez voir les enregistrements.

### 5. Création des requêtes

Une requête permet de poser une question sur les données stockées dans les tables ou d'accomplir une action sur ces données. Il existe différents types de requêtes: les requêtes sélection (à filtre, à paramètres, analyse croisée, à champs calculés, opérations) et les requêtes actions (ajout, suppression, mise à jour, création de tables).

- Les requêtes à filtres extraient les données d'une ou plusieurs tables et affichent les résultats (un sous-ensemble d'enregistrements) dans une feuille de données dans laquelle les données peuvent être modifiées.
- Les requêtes à paramètres affichent une boîte de dialogue demandant à l'utilisateur des informations, données, valeurs, ... nécessaires à l'extraction des données désirées.
- Les requêtes analyse croisée affichent des valeurs calculées (somme, compte, moyenne) d'un champ d'une table, et les regroupent selon un champ commun classé verticalement dans la première colonne de la feuille de données et un autre champ classé horizontalement dans la première ligne.
- Les requêtes à champs calculés affichent dans leur feuille de données un champ nouveau dont le contenu est le résultat de calculs (pas uniquement numériques) effectués à partir d'autres champs.
- Les requêtes opérations regroupent des enregistrements selon un critère commun et calculent des sommes, comptes, moyennes et autres types d'opérations.
- Les requêtes actions modifient les données de plusieurs enregistrements en une seule opération. Elles permettent de supprimer des enregistrements d'une ou plusieurs tables, d'en ajouter, d'en mettre à jour, de créer une nouvelle table.



#### 5.1. Les requêtes de type sélection

Les requêtes de type sélection posent des questions sur les données stockées dans les tables et renvoient une feuille de réponses dynamique sans modifier les données. Une fois la feuille de réponses dynamique affichée, on peut examiner et modifier les données. Toute modification sera également enregistrée dans les tables sources.

##### 5.1.1. Les requêtes à filtre

des premiers rôles d'une requête est de rassembler des champs sélectionnés dans plusieurs tables. Ensuite, ce type de requêtes consiste à filtrer les enregistrements qui satisfont un critère prédéfini. Pour écrire ces critères, il suffit de se familiariser avec quelques symboles ou opérateurs dont certains sont peut-être connus: <, >, \*, [ ], **Et (And)**, **Ou (Or)**, **Comme (Like)**, ... (Voir la fin de la partie sur les requêtes pour un tableau récapitulatif des symboles et opérateurs intéressants.)

Quand on tape des critères dans plusieurs cellules, Access les combine au moyen de l'opérateur **Et (And)** ou **Ou (Or)**. Si les expressions occupent des cellules différentes mais dans la même ligne, Access utilise l'opérateur **Et (And)**. Si des expressions ne se trouvent pas dans la même ligne de la grille d'interrogation, Access utilise l'opérateur **Ou (Or)**

**Rq " Van Hamme " ou " Rosinski " :** Requête sélection, tables Tomes et Séries.

Comme premier exemple, on construit une requête qui rassemble la liste de tous les albums auxquels le scénariste Van Hamme ou le dessinateur Rosinski ont participé, ensemble, ou indépendamment.

Pour créer cette requête, on clique sur l'objet Requetes (Queries), puis sur Nouvelle (New). Les requêtes de cette base seront créées à partir du mode création (Design View). Les tables à utiliser sont Tomes et Séries. Les champs à utiliser sont [Code Tome], [N° Tome], [Titre Tome], [Scénariste], [Dessinateur], et le champ [Série] de la table Séries. En appuyant sur le bouton feuille de données, on remarque que la relation entre les deux tables est bien comprise par la requête (le champ [Code Série] est remplacé par le nom complet de la [Série]). On retourne au mode Création pour saisir nos critères. Sur la ligne Critères (Criteria), on entre Van Hamme dans le champ [Scénariste] et Rosinski dans le champ [Dessinateur]. Dès que l'on quitte une cellule, des guillemets se positionnent autour des critères. En mode feuille de données, on découvre les albums « Thorgal » produit par les deux artistes. Pour obtenir les albums produits par l'un **ou** par l'autre, il faut entrer le premier critère sur la ligne Critères (Criteria) et le second sur la ligne Ou (Or). En mode feuille de données, on redécouvre les albums « Thorgal » auxquels s'ajoutent les albums « Largo Winch » et « XIII ». On peut utiliser les opérateurs **Et (And)** et **Ou (Or)** pour générer des critères plus poussés.

Lors de la fermeture de la requête, on l'enregistre sous le nom « Rq " Van Hamme " ou " Rosinsky " ».

**Rq Age d'Argent (1955-1970):** Requête sélection, tables Tomes et Séries.

Les critères ne sont pas réservés au type texte. La requête suivante les utilise sur un champ de type Date/Heure. Cette requête extrait tous les albums parus entre 1955 et 1970.

Pour créer cette requête, on clique sur Nouvelle (New), puis sur Création (Design View). Les tables à utiliser sont Tomes et Séries. Les champs à utiliser sont [Code Tome], [N° Tome], [Titre Tome], [Scénariste], [Dessinateur], [Date Parution], et le champ [Série] de la table Séries. Il faut extraire les albums parus après le 1/1/1955 mais avant le 1/1/1970 (2 critères). Le multi-critère agit cette fois sur le même champ. On utilise l'opérateur **Et (And)** sous la forme <1/1/1970 **Et (And)** >=1/1/1955. Des caractères dièses apparaissent lorsqu'on quitte la cellule : <#01/01/1970# **Et (And)** >=#01/01/1955#. En demandant le tri croissant (Sort : Ascending) sur le champ [Date Parution], la requête affiche ces tomes dans l'ordre chronologique de leur parution.

Il est possible que les dates figurant en salle Renaissance ou sur d'autres machines que vous utiliserez soient les dates au format américain. Ceux-ci inversent le jour et le mois. Ainsi, le 12/10/1999 doit alors être interprété comme le 10 décembre et non le 12 octobre. Cela ne cause pas de problème lors du passage d'une machine à l'autre, Windows interprétant automatiquement la date de manière correcte, ou la corrigeant même lors de l'encodage.

### 5.1.2. Les requêtes paramétrées (requêtes paramètre)

Les requêtes paramètre, qui ne constituent pas un type de requête différent, optimisent la souplesse d'emploi des autres requêtes en vous demandant de modifier certains critères chaque fois que vous exécutez la requête.

**Rq Tomes en \*\*\* :** Requête sélection, tables Tomes et Séries

On commence par construire une requête qui permet de retrouver tous les albums qui contiennent le mot "chat".

Pour créer cette requête, on clique sur Nouvelle (New), puis sur Création (Design View). Les tables à utiliser sont Tomes et Séries. Les champs à utiliser sont [Code Tome], [N° Tome], [Titre Tome], [Scénariste], [Dessinateur], et le champ [Série] de la table Séries. Sur la ligne Critères (Criteria), dans le champ [Titre], on

entre le critère **chat** (les guillemets apparaissent automatiquement). Le mode feuille de données n'affiche aucun enregistrement, car aucun album ne s'appelle « Chat ».

Pour préciser que « chat » n'est qu'un mot du titre, on utilise l'astérisque qui remplace n'importe quel caractère. Le nouveau critère est **\*chat\*** (la machine affiche **Like «\*chat\*»**). Le résultat en mode feuille de données est un ensemble d'albums contenant le mot « chat ».

Pour automatiser cette recherche avec un mot-clé quelconque sans chaque fois modifier le critère de la requête en mode modification, on remplace le critère par **Like \*&[Entrez le mot clé]&\***. Le champ [Entrez le mot-clé !] est un champ inconnu pour Access. Access demande alors une valeur pour ce paramètre à l'utilisateur. Chaque nouvelle ouverture de cette requête permettra d'entrer un nouveau mot-clé.

En fermant la requête, on la sauve sous le nom « Rq Tomes en \*\*\* »

### 5.1.3. Les requêtes avec champs calculés

Les champs extraits d'une requêtes ne se limitent pas aux champs des tables. Il est possible d'en "calculer" d'autres à partir de ceux existant. La syntaxe à utiliser pour le nom du champ calculé consiste à entrer un nom pour le nouveau champ, suivi d'un double point, suivi de la formule de calcul. Les noms de champs dans le calcul doivent apparaître entre crochets.

**Rq ventes mois:** Requête sélection, tables Tomes, Ventes, Clients.

Nous cherchons à construire le listing des ventes des 31 derniers jours. La machine construira ce listing en cherchant les albums dont la différence entre la date de vente et la date actuelle est inférieure à 31 jours. Cette différence est un champ que la machine peut calculer.

Pour créer cette requête, on clique sur Nouvelle (New), puis sur Création (Design View). Les tables à utiliser sont Tomes, Séries, et Clients (en prévision d'un prochain état). Les champs à utiliser sont [Titre Tome], [N° Tome], [Nom Client], [Prénom Client], et [Date Vente]. On y crée un champ supplémentaire appelé **[nb jours]** indiquant le nombre de jours qui se sont écoulés depuis la vente de l'album. Ce champ est un champ calculé selon la formule **nb jours : Date( )-[Date Vente]**. L'expression **Date( )** est une fonction indiquant la date courante (déjà utilisée dans la table Ventes) et changeant de jour en jour. **[Date Vente]** est la valeur entrée dans le champ [Date Vente]. Ainsi, **Date( )-[Date Vente]** représente le nombre de jours écoulés depuis les ventes. Sous ce champ, sur la ligne Critères (Criteria), on saisit **<31**, ne laissant plus figurer que les ventes désirées dans la feuille de données.

Si aucune nouvelle vente n'a été encodée durant les 31 jours précédant l'utilisation actuelle de la base de données, cette requête se trouvera tout naturellement vide d'enregistrements. En fermant la requête, on la sauve sous le nom « Rq ventes mois ».

### 5.1.4. Les requêtes « opération »

Les requêtes Opération permettent de réaliser un certain nombre de calculs sur tous ou partie des enregistrements. On peut par exemple regrouper les tomes d'une même série, faire des sommes, des moyennes, des comptes, extraire la première ou dernière valeur du groupe, ... Le bouton Opérations

(Totals) est celui comportant un symbole de sommation .

**Rq Compte Tomes par Série:** Requête Opération, basée sur les tables Séries et Tomes.

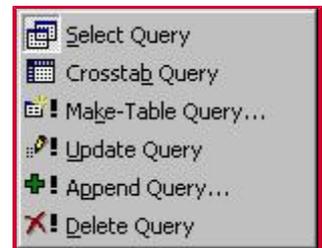
Nous cherchons les séries « durables », c'est à dire celles qui ont vu paraître un grand nombre de tomes. Pour cela, Access devra calculer le nombre de tomes parus dans chaque série.

Pour créer cette requête, on clique sur Nouvelle (New), puis sur Création (Design View). Les tables à utiliser sont Tomes et Séries. Les champs à faire apparaître sont [Série] et [N° Tome]. On clique sur le bouton Opérations (Totals). L'opération par défaut est le **Regroupement (Group By)**. On désire le **Regroupement** pour la [Série], mais le **Compte (Count)** pour le [N° Tome]. On en profite pour voir les autres opérations : **Somme (Sum), Moyenne (Avg), Minimum (Min), Maximum (Max)**, ... En passant en mode feuille de données, on vérifie que les 250 enregistrements sont regroupés en 31 séries avec à chaque fois le nombre d'albums parus dans la série.

En fermant la requête, on la sauve sous le nom « Rq Compte Tomes par Série »

## 5.2. Les requêtes de type action

Cette catégorie de requêtes comprend les requêtes Ajout (Append Query), Suppression (Delete Query), Création de table (Make-Table Query) et Mise à jour (Update Query). Les requêtes Suppression et Mise à jour modifient les données existantes ; les requêtes Ajout et Création de table déplacent les données existantes.



### 5.2.1. Les requêtes Mise A Jour (Update Query)

Access permet d'augmenter les prix de 5% pour tous les albums au moyen d'une simple requête de type action.

**Rq Inflation:** Requête action, Mise A Jour, basée sur la table Tomes.

Pour créer cette requête, on clique sur Nouvelle (New), puis sur Création (Design View). La table à utiliser est la table Tomes. Les champs à faire apparaître sont [Code Tome] et [Prix]. Pour modifier le type de requête (Query Type) on déroule la liste sous le bouton représentant deux feuilles de données superposées. On choisit le type Mise A Jour (Update Query). Dans le champ [Prix], sur la ligne Mise A Jour (Update To), on entre la formule de mise à jour : **[Prix] + [Prix] \* 0.05** (le prix de départ plus 5% de ce prix). La requête agit lorsqu'on appuie sur le bouton Exécuter (Run) symbolisé par un point d'exclamation. Un message nous avertit que les données vont être modifiées. On confirme. On vérifie en mode feuille de données que les prix ont été mis à jour. Notons que cette requête ne doit être utilisée qu'une fois par an sous risque de voir les prix rapidement faussés! A cet effet, nous avons construit la requête "Rq Inflation (Annuler)" qui effectue l'opération inverse

En fermant la requête, on la sauve sous le nom "Rq Inflation".

## 5.3. Les autres requêtes

A ce stade, on ferme <08-9eArt-t.mdb> et on ouvre <08-9eArt-trr.mdb>. Ce fichier est complet au niveau des tables et des requêtes, ce qui permet aux étudiants « perdus » ou « en retard » de se remettre à jour. On ne crée donc pas les requêtes qui suivent, mais on peut les visiter. Leur création étant semblable à un des types de requêtes rencontré ci-dessus, on ne les décrit que très brièvement.

**Rq Tomes " Les ... " :** Requête sélection, tables Tomes et Séries.

Cette requête affiche tous les tomes dont le titre commence par " Les ". Le critère entré sur la ligne " Critères " de la requête, sous le champ [Titres], est **Comme "Les \*" (Like "Les \*")**. L'opérateur **Comme (Like)** trie les titres semblables à ce qui suit entre guillemets. L'astérisque remplace n'importe quels caractères; elle sert de carte « Joker » Le champ [Série] de la table Séries permet de retrouver les séries dont sont tirés les albums sélectionnés.

**Rq Ventes par Clients:** Requête regroupement utilisée comme source de l'état correspondant.

**Rq Catalogue des Editeurs:** Requête regroupement utilisée comme source de l'état correspondant.

**Rq Prix Moyen Série:** Requête Opération, basée sur les tables Séries et Tomes.

Le premier effet de la requête est d'extraire les tomes dont un prix est affiché (certains albums ne sont pas entièrement encodés, ou leur prix est laissé à l'appréciation du vendeur) en entrant comme critère pour le prix l'expression **Est Pas Null (Is Not Null)**. Ensuite, les opérations sont le **regroupement (Group By)** sur le champ [Série], la **moyenne (Avg)** sur le champ [Prix], le **compte (Count)** sur le champ [N° Tome].

**Rq Editeurs:** Requête sélection, basée sur la table Editeurs.

Elle contient les champs [Editeur] et [Code Editeur]. Elle n'a pas d'intérêt direct, mais est utilisée comme liste modifiable dans le formulaire " Encodage Séries ".

**Rq Séries:** Requête sélection, basée sur la table Séries.

Elle contient les champs [Série] et [Code Série]. Elle n'a pas d'intérêt direct, mais est utilisée comme liste modifiable dans le formulaire " Encodage Tomes ".

**Rq Tomes:** Requête sélection, basée sur les tables Séries et Tomes.

Elle contient les champs [Série], [N°Tome], [Titre Tome] et [Code Tome]. Elle n'a pas d'intérêt direct, mais est utilisée comme liste modifiable dans le formulaire " Encodage Ventes ".

Le tableau ci-dessous reprend les syntaxes principales utilisées dans les critères de requêtes.

Like "S**"	Fournit tous les enregistrements pour lesquels le champ commence par S.
Like "**Imports"	Fournit tous les enregistrements pour lesquels le champ termine par le mot "Imports".
Like "[A-D]**"	Fournit tous les enregistrements pour lesquels le champ commence par une lettre entre A et D.
Like "**ar**"	Fournit tous les enregistrements pour lesquels le champ contient la séquence "ar".
Like "Acc?"	Fournit tous les enregistrements pour lesquels le champ commence par "Acc" et pour lesquels la quatrième et dernière lettre est inconnue.
Like [forms]![NomForm]![NomChamp]	Fournit tous les enregistrements pour lesquels le champ est identique au champ "NomChamp" du formulaire "NomForm".
< 100	Fournit tous les enregistrements pour lesquels le champ contient une valeur numérique inférieure à 100.
< 1/1/2000	Fournit tous les enregistrements pour lesquels le champ contient une date antérieure au 1/1/2000.
A And B	Fournit tous les enregistrements pour lesquels le champs satisfait le critère A et le critère B.
A Or B	Fournit tous les enregistrements pour lesquels le champs satisfait le critère A ou le critère B.
A Xor B	Fournit tous les enregistrements pour lesquels le champs satisfait le critère A ou le critère B de manière exclusive.
Null	Fournit tous les enregistrements pour lesquels le champs est vide.
Is not null	Fournit tous les enregistrements pour lesquels le champs n'est pas vide.

## 6. Création de formulaires

Maintenant que les données sont encodées et que des outils pour les analyser et modifier ont été créés, on va faciliter le travail de gestion de l'utilisateur grâce aux formulaires. On désire à présent faciliter l'encodage et soigner la présentation des données. En exemple, on va faire en sorte que la couverture des albums soit visible sur une fiche reprenant ses caractéristiques. En termes techniques, la transformation de <08-9eArt-trr.mdb> va consister à créer des formulaires, puis au cours suivant, à y ajouter des macros.

Il existe différents types de formulaires. Parmi les types principaux, on retiendra les formulaires colonnes, les formulaires tabulaires, les formulaires à sous-formulaires, les formulaires graphiques, les formulaires tables pivots, les formulaires indépendants.

- Les formulaires colonnes affichent les valeurs des champs les unes sous les autres, un enregistrement à la fois.
- Les formulaires tabulaires affichent les données telles que dans une table: les champs en colonnes, les enregistrements en lignes. La présentation des données est plus soignée que dans les tables.
- Les formulaires à sous formulaires sont composés d'un formulaire principal contenant les enregistrements parents d'une relation "un à plusieurs", et d'un sous-formulaire contenant les enregistrements enfants de cette relation.
- Les formulaires graphiques affichent les données sous formes de graphiques variés.
- Les formulaires tables pivots font les calculs que vous désirez, tels que la somme et le compte, basés sur la manière dont les données sont arrangées dans le formulaire. Par exemple, un formulaire table pivot peut afficher les valeurs d'un champ horizontalement ou verticalement, et ensuite calculer le total de la ligne ou de la colonne.
- Les formulaires indépendants ne contiennent pas de données provenant des tables et requêtes. Ils permettent à l'utilisateur de se guider dans la base de données, ou d'entrer des paramètres nécessaires aux fonctionnements des objets de la base.

## 6.1. Création de formulaires classiques

**Encodage clients:** Colonne simple, table Clients.

Pour créer un nouveau formulaire, il faut avant tout cliquer sur l'objet des formulaires, puis sur le bouton Nouveau (New). Regardons d'abord les différents Assistants disponibles. On choisit ensuite l'Assistant Formulaire (Form Wizard). La table à utiliser est la table Clients. Tous les champs sont à faire apparaître. Les champs apparaîtront en colonne (Columnar). Pour ce qui est du style, le choix est libre. Notons toutefois qu'un style est composé d'une image de fond. Une image de fond est soit encadrée, soit liée au formulaire. Dans le cas de l'encastrement, la taille de la base de donnée augmente en fonction de la taille de l'image. Dans le cas du lien, la base de donnée garde sa taille d'origine, mais l'image liée doit suivre la base de données lors de tous ses déplacements (disquette, autre PC, ...). Cette distinction se fait dans la propriété Type d'Image (Picture Type) du formulaire: Encadrée (Embedded) ou Liée (Linked).

L'intérêt d'un tel formulaire est un encodage aisé : il suffit d'atteindre la fiche vierge (5e bouton en bas du formulaire)

et de saisir les nouvelles données.

**Acheteurs d'un tome:** Tabulaire, requête Rq Ventes.

Les formulaires tabulaires présentent les données sous forme de tableaux, comme dans les tables, mais avec une mise en forme choisie par le concepteur. On va par exemple afficher le listing de tous les achats avec le code de l'album et les coordonnées de l'acheteur. Ce formulaire sera utilisé par la suite, lors de la création du formulaire des tomes (Il permettra de voir tous les acheteurs d'un tome choisi).

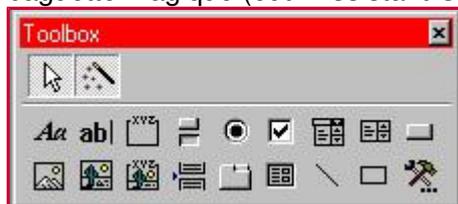
Pour créer ce formulaire, on clique sur le bouton Nouveau (New). On choisit l'Assistant Formulaire (Form Wizard). La requête à utiliser est la requête Ventes. Les champs à utiliser sont [Code Tome], [Nom Client], [Prénom Client] et [Téléphone]. Les données sont à trier par vente. Les champs apparaîtront en tableau (Tabular). Pour le style, le choix est libre. Une fois le formulaire créé, on le personnalise à partir du mode création (modification des couleurs, polices, largeur des zones de texte, ...).

**Encodage séries:** Colonne simple.

Ce formulaire permettra d'encoder les renseignements concernant une nouvelle série. Le formulaire contiendra une liste modifiable pour encoder l'éditeur. Le contenu de cette liste est le contenu de la requête " Rq Editeurs ". Ainsi, il ne faudra pas entrer le code de l'éditeur soi-même (le code étant la seule information disponible dans la table des séries, et ces codes n'étant pas toujours facilement mémorisables), mais simplement cliquer sur le bon nom dans la liste (issue de la table des éditeurs). Grâce aux relations, la machine identifie correctement l'éditeur à partir de son code.

Pour créer ce formulaire, on clique sur le bouton Nouveau (New). On choisit l'Assistant Formulaire (Form Wizard). La table à utiliser est la table Séries. Tous les champs sont à utiliser. Les champs apparaîtront en colonne (Columnar). Une fois le formulaire créé, on le personnalise à partir du mode création (déplacement de la zone de texte [description] et de son étiquette).

Pour obtenir la liste modifiable (Combo Box), on supprime la zone de texte [Code Editeur] et on la remplace par une liste modifiable. Pour une construction assistée, il faut d'abord avoir enfoncé l'outil représentant la baguette magique (outil Assistant Outils). Nous détaillons plus loin les quelques outils de la boîte à outils.



A la première étape de l'Assistant, on précise que les données proviennent (d'une table ou) d'une requête. A la deuxième étape, on précise qu'il s'agit de la requête « rq éditeurs ». A la troisième étape, on sélectionne tous les champs. A la quatrième étape, on précise le tri sur ces champs. A la cinquième étape, on détermine la largeur de chaque colonne. A la sixième étape, on indique que le champ [Code Editeur] identifie de manière unique la donnée à mémoriser (les autres colonnes ne sont donc présentes que pour informer l'utilisateur). A la dernière étape, on précise que le champ stockant l'information est le champ [Code Editeur] de la table Séries. On termine en modifiant l'étiquette de la liste modifiable.

## 6.2. Création de formulaires à sous-formulaire

L'intérêt d'une base de données relationnelle consiste en ses relations. Il est possible de construire des formulaires qui matérialisent ces relations. Il est possible de voir toutes les séries d'un même éditeur sur la fiche de l'éditeur, ou même, comme on va le faire de suite, de voir tous les albums publiés dans une même série sur la fiche de la série.

The screenshot shows an Access form titled "Series". At the top, there is a dropdown menu labeled "Choisissez la série:". Below it, there are several text boxes: "Code Série:" with the value "ast", "Créateur:" with "Goscinny", "Série:" with "Astérix", and "Date Création:". A "Description:" field contains the text "Goscinny devait avoir raison lorsqu'il prétendait que c'est Astérix qui a amené po". Below the form fields is a table titled "Tomes de la série". The table has columns for "N°", "Titre Tome", "Scenariste", "Dessinateur", "Date parution", "Prix", and "#". The table contains 6 rows of data for the Astérix series. At the bottom of the form, there are two record navigation controls: "Record: 1 of 10" and "Record: 1 of 28".

N°	Titre Tome	Scenariste	Dessinateur	Date parution	Prix	#
1	Astérix le Gaulois	Goscinny	Uderzo	1961	8,68 €	20
2	La Serpe d'or	Goscinny	Uderzo	1962	8,68 €	20
3	Astérix et les Goths	Goscinny	Uderzo	1963	8,68 €	20
4	Astérix gladiateur	Goscinny	Uderzo	1964	8,68 €	20
5	Le Tour de Gaule d'Astérix	Goscinny	Uderzo	1965	8,68 €	20
6	Astérix et Cléopâtre	Goscinny	Uderzo	1965	8,68 €	20

On va voir comment se partagent les 250 albums entre les 31 séries. Pour créer ce formulaire, on clique sur le bouton Nouveau (New). On choisit l'Assistant Formulaire (Form Wizard). La table à utiliser est la table Séries. Tous les champs sont à utiliser. On en profite pour ajouter la table Tomes et tous ses champs (sauf peut-être le champ [Code Série] qui ne serait qu'une répétition). Ayant choisis des champs issus de plusieurs tables, l'étape suivante nous propose de construire un formulaire à sous formulaire, ou un formulaire lié à un autre formulaire. On choisit la première option. Les étapes suivants concernent la mise en forme de ces formulaires.

Un autre méthode de construction plus personnalisée consiste à créer le formulaire en 3 étapes:

A) Il faut construire les fiches des séries. B) Il faut créer le listing de tous les tomes. C) Il faut incorporer ce dernier formulaire dans le précédent. C'est toujours le champ [Code Série], lien entre les tables Séries et Tomes, qui permet d'afficher les tomes appropriés.

Notons que le formulaire « Tomes d'une Série » figurant dans la base de données <08-9eArt.mdb> contient une liste modifiable permettant d'atteindre l'enregistrement désiré. La construction de la liste modifiable se fait ainsi:

- A la première étape de l'assistant on précise que l'on doit atteindre un enregistrement du formulaire basé sur la valeur qui sera sélectionnée dans la liste modifiable.
- A la deuxième étape on précise que les champs à utiliser sont le code série et la série.
- A la troisième étape on précise la largeur des colonnes.
- La quatrième étape nous demande un nom pour l'étiquette (disons [Choix Code Série]).

**Encodage ventes:** Colonne simple, table Ventas.

Ce formulaire sera le formulaire de travail de l'utilisateur. Pour encoder une nouvelle vente, il suffira de suivre les instructions affichées dans les listes modifiables, à savoir, choisir le client dans une première liste modifiable, choisir l'album vendu dans une deuxième liste, vérifier que la date de vente est bien la date actuelle, et confirmer la vente. La vente sera encodée en faisant deux choix (le client et l'album) et en cliquant sur un bouton. En prime, le nombre d'albums de ce titre en stock sera automatiquement diminué d'une unité.

Pour créer ce formulaire, on clique sur le bouton Nouveau (New). On choisit l'Assistant Formulaire Colonne (Autoform Columnar). La table à utiliser est la table Ventes. En mode formulaire, on constate qu'il faut connaître les codes du client et de l'album pour un nouvel encodage. On revient en mode création pour transformer ces deux zones de texte en listes modifiables (Combo Box). On supprime la zone [Code Client]. On crée une liste modifiable à l'aide de l'Assistant. (Le menu contextuel permet de "changer" une zone de texte en une liste modifiable, mais les paramètres suivants doivent alors être indiqués manuellement, ce qui n'est pas aisé.) A la première étape de l'Assistant, on précise que les données proviennent d'une table (ou d'une requête). A la deuxième étape, on précise qu'il s'agit de la table « Clients ». A la troisième étape, on sélectionne les champs [Nom Client], [Prénom Client] et [Code Client]. A la quatrième étape, on confirme à la machine qu'elle doit cacher la colonne Clé Primaire (Hide Key Column). On regarde quand même l'effet produit si on refuse cette option. A la cinquième étape, on précise que le champ stockant l'information est le champ [Code Client]. On termine en changeant l'étiquette de la liste modifiable par le texte « Choisissez le client dans cette liste ».

On reproduit cette création de liste modifiable pour le [Code Tome]. La procédure est identique. Il ne faut pas choisir la table Clients, mais bien la requête « Rq Tomes » comme source des données. L'étiquette doit être modifiée en conséquence.

Lors de la leçon portant sur les macros, on construira un bouton qui aura pour effet de passer à la fiche vierge de ce formulaire, et qui diminuera le stock de tomes d'une unité. Cette macro n'est pas faisable avec l'Assistant Bouton car elle contient plus d'une action.

### 6.3. Les autres formulaires

On ferme <08-9eArt-trr.mdb> et on ouvre <08-9eArt-trrfm.mdb>. Ce fichier est complet au niveau des tables, requêtes et formulaires, ce qui permet aux étudiants « perdus » ou « en retard » de se remettre à jour. <08-9eArt-trrfm.mdb> contient en outre les macros nécessaires au bon fonctionnement de ces formulaires. On créera des macros au prochain cours. La création des formulaires suivants étant semblable à celle des formulaires rencontrés ci-dessus, ils ne sont décrits que très brièvement.

**Encodage tomes:** Colonne simple.

Formulaire contenant les renseignements d'un tome. Ce formulaire permet d'encoder les nouveaux tomes. La date de parution sera par défaut, pour les nouveaux tomes, la date courante (constante =Date() ).

En lançant le mode formulaire, on voit apparaître la couverture (de quelques tomes seulement pour ne pas trop alourdir la base de données) grâce à la liaison OLE (OLE Object). On peut modifier le format de l'objet OLE en mode création en tirant sur ses poignées. Pour l'insertion d'objets OLE (des images en particulier), nous vous conseillons de les copier/coller à partir d'un logiciel de lecture d'images. Attention, plus la taille de

vos images sera petite, moins votre base de données sera lourde. Une image de 5Ko peut faire croître la taille de la base de données de 100Ko! C'est pour cela que nous n'avons inséré qu'une dizaine de petites images dans notre base.

Le formulaire contient une liste modifiable pour l'encodage de la série. Voici ses propriétés.

Nom contrôle.....	Code Série	C'est sous ce nom que sera reconnu le contrôle dans le formulaire.
Source contrôle.....	Code Série	On ne crée pas un nouveau contrôle mais remplaçons la zone de texte
Origine source.....	Table/Requête	Le contenu de la liste existe déjà sous forme de requête, ...
Contenu.....	Rq Séries	... il s'agit de la requête " Rq Séries ".
Nbre colonnes.....	2	On désire le nom et le code de la série.
Colonne liée.....	2	Seul le code de la série (2e colonne) doit être mémorisé par ce contrôle.

Ce formulaire contient une liste modifiable permettant d'atteindre un enregistrement spécifique. Voici ses propriétés.

Nom contrôle.....	Choix Code Tome	C'est sous ce nom que sera reconnu le contrôle dans la macro.
Source contrôle....		Le contrôle n'est pas un contrôle d'origine de la table.
Origine source.....	Table/Requête	Le contenu de la liste existe déjà sous forme de requête, ...
Contenu.....	Rq Tomes	... il s'agit de la requête " Rq Tomes ".
Nbre colonnes.....	4	On désire le nom de la série, les n°, titres et codes du tome.
Largeur colonnes...	0cm;2cm; 0,5cm;3cm	
Colonne liée.....	1	Seul le code du tome (1e colonne) doit être mémorisé par ce contrôle.
Après MAJ.....	Trouver.Tome	Choisir un tome a pour effet de lancer la macro Trouver.

Ce formulaire contient aussi un bouton de commande ouvrant le formulaire " Acheteur d'un tome " indiquant la personne qui a acheté l'album considéré. Les propriétés principales de ce bouton de commande sont les suivantes:

Nom contrôle.....	Acheteurs	
Légende.....	Acheteurs de ce tome	Légende qui figurera sur le bouton.
Sur pression.....	Ouvrir.acheteurs	Appuyer sur le bouton a pour effet d'exécuter la macro Ouvrir.acheteurs.

**Séries d'un éditeur:** Formulaire Colonne, table Editeur.

Dans ce formulaire figurent les données concernant l'éditeur sous forme de colonne simple. On y insère le sous-formulaire1 dans lequel sont listées les séries de cet éditeur. C'est le champ [Code Editeur], lien entre les tables Editeurs et Séries, qui permet d'afficher les séries appropriées dans le sous-formulaire.

**Sous-formulaire1:** Formulaire Tabulaire, table Série.

Listing de toutes les séries. Inséré comme sous-formulaire dans le formulaire " Séries d'un éditeur ".

**Menu des Formulaires:** Certains des formulaires mentionnés ci-dessus ne sont pas destinés à être ouverts directement. Ce formulaire menu est un formulaire de présentation ayant pour but de proposer d'ouvrir les formulaires intéressants: deux de listings, quatre d'encodage. Ce formulaire contient au total six boutons de commande renvoyant à la macro Ouvrir. Il est plus commode de ne pas créer de macro pour tous ces boutons, mais d'utiliser l'Assistant Bouton, plus rapide et moins technique. Dans le cas où on crée les macros, il faut préciser leur nom au niveau de la propriété **sur Pression (On Click)** de chacun des boutons.

## 6.4. Quelques propriétés des contrôles dans les formulaires

Nous terminons ce chapitre sur les formulaire par un tableau récapitulatif des propriétés les plus souvent rencontrées lors de la création de contrôles dans les formulaires. Nous avons essentiellement repris les propriétés des zones de textes et des listes modifiables.

<b>Pour les zones de textes</b>	<b>Signification</b>
---------------------------------	----------------------

Name	Nom du contrôle
Control Source	Champ qui sert de source au contrôle
Default Value	Valeur par défaut du contrôle
Visible	Le contrôle doit-il être visible (Yes/No)
Can Grow / Shrink	Le contrôle peut-il agrandir/diminuer si son contenu est trop grand/petit (Yes/No)
Left / Top	Positionnement par rapport au bord gauche et supérieur du formulaire
Width / Weight	Largeur et épaisseur du contrôle
Back Style / Color	Style et couleur de fond
Border Style / Color / Width	Style, couleur et épaisseur de la bordure du contrôle
Font Name / Size / Weight	Nom, taille et épaisseur de la police
After Update	Evenement à effectuer après la mise à jour du contrôle
On Click / Dbl Click / Enter	Evenement à effectuer sur clic/double clic ou confirmation de la valeur

Pour les listes modifiables	Signification
Row Source Type	Les données proviennent-elles d'une table? requête? Liste de mots?
Row Source	Nom de la table, de la requête ou liste de mots servant de source
Column Count	Nombre de colonnes
Column Width	Largeur des colonnes (différentes largeurs séparées par des " ; ")
Bound Column	Numéro de la colonne liée au contrôle (une seule colonne est associée au contrôle)
List Rows	Nombre de lignes dans la liste modifiable

## 6.5. Construction d'un formulaire « Menu »

On commence en ouvrant <08-9eArt-trrf.mdb>. La transformation va consister à créer les macros afin de rendre les formulaires interactifs par la présence de boutons, de listes modifiables, ...

Nous avons précédemment ouvert le fichier <08-9eArt-trrfm.mdb> qui proposait un formulaire d'accueil de couleur rouge. Bien que les formulaires soient présents dans <08-9eArt-trrf.mdb>, le formulaire rouge n'apparaît pas car son lancement automatique se fait grâce à la macro autoexec qui n'existe pas encore. C'est l'occasion de faire un rappel sur les formulaires en construisant ensemble ce formulaire rouge (ou du moins les étapes intéressantes).



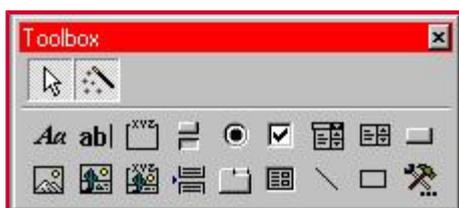
Ce formulaire est un formulaire de présentation ayant pour but de faciliter le travail d'encodage et de gestion du vendeur. Il propose d'ouvrir les formulaires intéressants : deux de listings, quatre d'encodage. Ce formulaire contient au total six boutons de commande renvoyant à la macro Ouvrir.

Pour créer ce nouveau formulaire, il faut avant tout cliquer sur l'onglet des formulaires, puis sur le bouton Nouveau (New). On choisit le mode création (Design View). Dans le cas de ce formulaire, il n'y a aucune table à utiliser. Le formulaire apparaît directement en mode création. Pour y placer un maximum de boutons, on clique et glisse (on drague) sur les bord pour agrandir la surface grise.

On commence par modifier la couleur de fond du formulaire en en choisissant une à partir de l'outil représentant un pot de peinture. Pour une image de fond, il faut choisir la propriété image (picture) du formulaire. (Les propriétés du formulaire, pas de ses objets, s'obtiennent en cliquant sur le carré gris dans le coins supérieur gauche du formulaire.) Le bouton assistant de cette propriété (bouton caché représenté par des points de suspension) lance un explorateur qui permet d'aller chercher l'image voulue sur le disque. La propriété Type d'Image (Picture Type) permet d'insérer (embedded) ou de lier (linked) l'image. (Cette dernière possibilité n'est pas active pour des Liaisons OLE.) L'intérêt de choisir de lier l'image est que son poids n'est pas comptabilisé dans celui de la base de données. Le travail ne sera donc pas trop lourds. Toutefois, Access retient le chemin "absolu" de l'image. C'est à dire que l'image ne sera plus visible si l'on change son chemin dans l'arborescence du disque. Ceci est un risque dans le cadre de notre travail puisque nous travaillons déjà sur plusieurs disques (C, D, F, H). Nous déconseillons donc le type d'image liée (linked) dans le cadre de votre travail.

Comme on va disposer des boutons sur ce formulaire, on crée une étiquette (label) qui propose à l'utilisateur un petit texte informatif quant à leur fonction. L'outil étiquette (label) est l'outil représentant le texte « Aa ». On clique sur cet outil. On drague sur notre formulaire de sorte à dessiner un rectangle, suffisamment grand pour qu'on puisse y écrire une phrase. On entre le texte « Que voulez-vous faire? » (on construit ici une version simplifiée du formulaire « 1 :Menu ») et on confirme en appuyant sur Enter. Pour que le texte ressorte mieux, on modifie la police, la taille, la couleur grâce à l'outil représentant un « A » sur un trait rouge, et on met ce texte en gras, italique, ...

On passe ensuite à la construction d'un bouton. On vérifie que l'outil Assistant (Control Wizards) de la barre d'outil (Toolbar) est enfoncé. On clique sur l'outil représentant un bouton (Command Button). On drague sur le formulaire de sorte à dessiner un rectangle, suffisamment grand pour que L'utilisateur n'ait pas de peine à cliquer dessus.



Ce bouton aura comme fonction d'ouvrir le formulaire des clients. L'action à choisir appartient donc à la catégorie Opérations sur les Formulaires (Form Operations) et s'appelle Ouvrir Formulaire (Open Form). A l'étape suivante, on précise que le formulaire à ouvrir est Encodage Clients. A l'étape suivante, on précise à l'Assistant que le formulaire s'ouvrira en proposant toutes les données. A l'étape suivante, on choisit de remplacer l'image sur le bouton par le texte « Formulaire d'encodage des clients ». A la dernière étape, on conserve le nom de bouton proposé par la machine.

Nous en profitons pour décrire les quelques outils intéressants de la boîte à outils (toolbox). Nous les décrivons à partir de la disposition "horizontale" de la boîte. Le premier bouton est le pointeur qui permet de sélectionner les objets. Le second est celui qui active l'assistant de création des outils.

Sur la première ligne d'outils à proprement parler:

- Outil étiquette
- Outil zone de texte
- Groupe d'options: regroupe sous un même contrôle (pour un même champ) différents outils (par exemple des boutons d'options)
- Bouton poussoir: utiles par exemple pour des champs de type Oui/Non
- Bouton d'options: utiles par exemple pour des champs de type Oui/Non ou dans un groupe d'options
- Case à cocher: utiles par exemple pour des champs de type Oui/Non
- Liste modifiable
- Liste de choix: Liste modifiable affichant à tous moments plusieurs lignes de choix
- Bouton

Sur la deuxième ligne d'outils

- Picture: pour insérer une image fixe sur le formulaire
- Cadre d'objet indépendant: pour insérer une image ou un objet fixe sur le formulaire
- Cadre d'objet dépendant: pour insérer une image ou un objet dépendant d'un champ de type Liaison OLE
- Saut de page
- Onglets: utile si de nombreux contrôles doivent être affichés sur un espace limité, ou regroupés par thèmes
- Sous-Formulaire ou Sous-Etat
- Ligne
- Rectangle
- d'autres contrôles ...

On passe en mode formulaire et on vérifie l'efficacité du bouton en cliquant dessus. On ferme le formulaire « Encodage Clients » et on repasse au mode création de notre formulaire de menu. Les étudiants construisent seuls un bouton ayant pour fonction d'ouvrir le formulaire « Encodage Ventes ». En quittant le formulaire, on l'enregistre sous le nom « Menu ».

## 7. Construction des macros

L'action qui consiste à ouvrir chaque formulaire à partir du formulaire "Menu" est mémorisée par la base de données sous forme d'un module. Un module est un des objets d'Access. Cet objet est très technique, et fait appel à des notions de programmation. Il est donc difficile, à notre niveau, de modifier une action enregistrée sous forme de module.

Un de ces boutons ouvre le formulaire « Encodage Ventes » au premier enregistrement. Mais pour encoder une nouvelle vente, l'utilisateur doit atteindre une fiche vierge. On désire plutôt produire deux actions de manière automatique: d'abord ouvrir le formulaire, puis atteindre la fiche vierge. L'Assistant Bouton ne nous le permet pas. Les macros le permettent.

Une macro est une liste d'instructions créées dans le but d'exécuter automatiquement certaines tâches. Chaque instruction s'appelle une action. Lorsqu'on exécute la macro, Access réalise les actions, dans la séquence indiquée, en utilisant les objets et les données spécifiés en tant qu'arguments des actions. Les macros servent à automatiser les tâches répétitives et à enrichir les possibilités de la base de données. Contrairement aux modules, il n'est pas nécessaire de savoir programmer pour créer une macro.

### 7.1. La macro Ouvrir Encodage Ventes

Pour créer une nouvelle macro, il faut avant tout cliquer sur l'onglet des macros, puis sur le bouton Nouvelle (New). Deux colonnes apparaissent.



Dans la colonne de gauche, on inscrira les actions. Il ne faut pas connaître toutes les actions de mémoire puisqu'elles sont accessibles à partir d'un menu déroulant. Dans la colonne de droite on inscrira une description afin de se souvenir des buts de la macro. Sous les deux colonnes, on introduira les arguments de chaque action. Les arguments à fournir varient en fonction de chaque action. La plupart d'entre eux sont aussi accessibles à partir d'un menu déroulant.

L'outil contenant « xyz » au dessus de deux carrés est l'outil Nom de Macro (Macro Name) qui permet de regrouper plusieurs macros sous un même nom général, en les différenciant par un "sous-nom" (une sorte de prénom ; chaque prénom détermine une « sous-macro »). La macro « Trouver » est basée sur ce principe : elle contient deux sous-macros: "Trouver.Tome" et "Trouver.Serie" qui sont deux macros basées sur le même principe, mais appliquées à deux objets différents. L'outil représentant un petit organigramme est l'outil Conditions qui permet l'exécution ou non d'une macro selon qu'une condition introduite est satisfaite ou non.

Dans la suite de ce document, on décrit les actions des macros dans la colonne de gauche. Le texte à droite de l'action n'est pas la description de la macro, mais la liste des arguments. Il y a la version à n'utiliser que sur la version française d'Access, et la version à n'utiliser que sur la version anglaise d'Access (Il existe des différences de syntaxe d'une version à l'autre).

La première macro est constituée de deux actions décrites ci-dessous. Elle ouvre donc le formulaire « Encodage Ventes » sur une fiche vierge.

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OuvrirFormulaire	<b>Nom formulaire:</b> Encodage Ventes
AtteindreEnregistrement	<b>Atteindre:</b> nouveau

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OpenForm	<b>Form Name:</b> Encodage Ventes
GoToRecord	<b>Record:</b> new

En fermant la macro, on l'enregistre sous le nom "Ouvrir Encodage Ventes".

Il faut à présent ouvrir le formulaire « Menu » et y remplacer l'action construite à l'aide de l'Assistant par cette nouvelle macro. Dans le formulaire « Menu » ouvert en mode Création, on fait apparaître les propriétés du bouton qui permet d'ouvrir notre formulaire « Encodage Ventes ». La propriété **Sur Pression (On Click)** contient la mention [Event Procedure] (il s'agit de l'action que nous avons précédemment construite à l'aide de l'assistant bouton). En faisant apparaître le menu déroulant dévoilant la liste des macros, on remplace la mention [Event Procedure] par « Ouvrir Encodage Ventes ». Un test en mode formulaire permet de vérifier que le formulaire « Encodage Ventes » s'ouvre effectivement sur une fiche vierge.

## 7.2. La macro Fermer

On crée un bouton sur le formulaire « Menu » qui fermera ce même formulaire.

La méthode décrite ci-dessus consiste à créer la macro, puis le bouton, et affecter la macro au bouton. La méthode décrite maintenant consiste à créer le bouton et y construire directement la macro à partir d'un générateur d'expression.

On désactive l'outil Assistant Outil (Control Wizards) de la barre d'outils (Toolbar). On dessine un nouveau bouton et on fait apparaître ses propriétés. On clique sur la propriété **Sur Pression (On Click)**, puis sur le bouton du générateur d'expression (bouton avec points de suspension). On appelle le Générateur de Macro (Macro Builder). Le nom de la macro sera « Fermer Menu ». La seule action dans ce cas-ci sera Fermer (Close). L'objet (Object Type) à fermer est un formulaire (Form). Son nom (Object Name) est « Menu ».

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
Fermer	<b>type objet:</b> Formulaire <b>nom objet:</b> Menu

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
Close	<b>Object type:</b> Form <b>Object name:</b> Menu

## 7.3. La macro Autoexec

Lors de l'ouverture de la base de données, le formulaire « 1 : Menu » s'ouvre automatiquement. Ceci se produit grâce à la macro « Autoexec ». Le nom de cette macro est un nom réservé. Lors de l'ouverture de la base de données, Access exécute, si elle existe, la macro Autoexec.

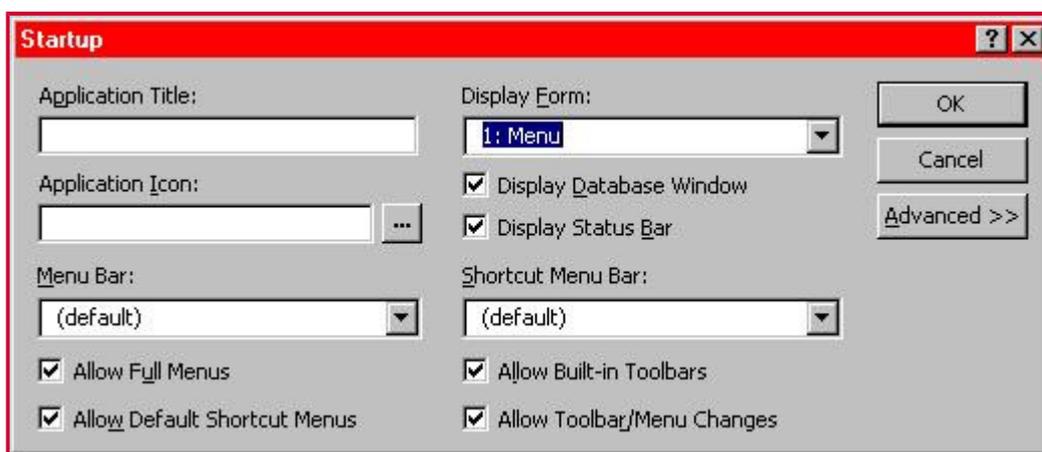
Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OuvrirFormulaire	<b>Nom formulaire:</b> 1:Menu

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OpenForm	<b>Form name:</b> 1:Menu

L'autre macro dont le nom est réservé est la macro AutoKeys qui affecte à une touche ou à un jeu de touches des macros prédéfinies.

Quand la première action à exécuter n'est que l'affichage d'un formulaire, on peut l'effectuer à l'aide du menu Outils/Démarrage/AfficherFormulaire (Tools/Startup/DisplayForm), en choisissant dans la zone Display Form le

formulaire à afficher automatiquement.



## 7.4. Les autres macros

A ce stade, on ferme <08-9eArt-trrf.mdb> et on ouvre <08-9eArt-trrfm.mdb>. Ce fichier est complet au niveau des tables, requêtes, formulaires et macros, ce qui permet aux étudiants « perdus » ou « en retard » de se remettre à jour. On ne crée pas les macros qui suivent, mais on peut les visiter et/ou vérifier leur action. Leur création étant semblable à une des macros rencontrées ci-dessus, on ne les décrit que très brièvement.

### 7.4.1. La macro Ouvrir

Ouvre le formulaire demandé dans le formulaire " 1:Menu des Formulaires ", ou dans le formulaire " Encodage Tomes ". Une macro peut contenir plusieurs sous-macros. Les sept sous-macros de la macro Ouvrir sont " acheteurs ", " séries d'un ", " tomes d'une ", " encod sér ", " encod tom ", " encod cli ", et " encod ach ". Voici les actions et arguments des macros " Ouvrir.acheteurs ", " Ouvrir.séries d'un " et " Ouvrir.encod sér ".

**Ouvrir.acheteurs:** Ouvre le formulaire " Acheteurs d'un Tome " à partir du formulaire " Encodage Tomes ", mais ne montre que les acheteurs du tome courant grâce à la Condition Where.

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OpenForm	<b>Form Name:</b> Acheteurs d'un Tome <b>Condition Where:</b> [Code Tome] = Forms ! [Encodage Tomes] ! [Code Tome]

Ce dernier argument signifie que le formulaire à ouvrir ne doit afficher que les enregistrements dont la valeur du champ [Code Tome] correspond à la valeur du champ [Code Tome] dans le formulaire principal qui est " Encodage Tomes " (noté "Forms ! [Encodage Tomes] ! [Code Tome]").

**Ouvrir.séries d'un:** Ouvre le listing des séries d'un éditeur quelconque

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OpenForm	<b>Form Name:</b> Séries d'un Editeur

La macro " Tomes d'une " est semblable à cette macro.

**Ouvrir.encod sér:** Ouvre le formulaire d'encodage des séries à l'enregistrement vierge, le dernier.

Action	Paramètres de l'action (à déterminer en bas à gauche de la fenêtre)
OpenForm	<b>Form Name:</b> Encodage Séries
GoToRecord	<b>Record:</b> new

Les macros " encod tom " et "encod cli " sont semblables à cette macro.

## 7.5. Les actions les plus usuelles

Vous trouverez dans le tableau ci-dessous un bref descriptif des actions de macros les plus souvent utilisées.

Close	Ferme l'objet en cours.
-------	-------------------------

FindRecord	Trouve le premier enregistrement qui correspond au critère de recherche spécifié.
GoToRecord	Atteint un enregistrement spécifié.
GoToControl	Atteint le contrôle ou le champ spécifié de l'enregistrement courant.
Maximize/Minimize	Agrandit ou réduit la fenêtre de l'objet en cours.
OpenForm	Ouvre le formulaire spécifié.
OpenQuery	Ouvre et exécute la requête spécifiée.
OpenReport	Ouvre l'état spécifié.
OpenTable	Ouvre la table spécifiée.
RunApp	Exécute la ligne de commande spécifiée, par exemple pour ouvrir le logiciel Excel.
RunMacro	Exécute la macro spécifiée.
Save	Sauve l'objet en cours.
SetValue	Définit une valeur spécifiée pour le contrôle spécifié.
...	...

## 8. Création d'états

Un état est un objet de base de données Access qui présente des informations mises en forme et organisées selon nos spécifications, par exemple des récapitulatifs de ventes, des listes de numéros de téléphone ou des étiquettes pour publipostage. Les états de la base de données "9e Art" sont tous créés à l'aide des Assistants. Ils sont ensuite modifiés afin d'être personnalisés. Par exemple, on enlève la date en anglais qui se trouve en bas de chaque page, on joue sur les propriétés "Can Grow/Can Shrink" des contrôles, on améliore l'ergonomie de la page,...

Il existe différents types d'états: les états colonnes, les états tableaux, les états graphiques, les états de publipostage, les états de regroupement.

- Les états colonnes affichent les valeurs des champs les uns sous les autres, un enregistrement à la fois.
- Les états tabulaires affichent les données telles que dans une table: les champs en colonnes, les enregistrements en lignes.
- Les états graphiques affichent les données sous formes de graphiques variés.
- Les états de publipostage permettent l'impression des données sur des étiquettes autocollantes ou sur des enveloppes.
- Les états anciennement appelés de regroupement sont des états tabulaires pour lesquels sont calculés pour certains groupes de données des sommes, moyennes, comptes, ... Les réponses données aux questions de l'Assistant amènent à la distinction entre état tabulaire et état de regroupement.

### 8.1. Les états publipostages

**Etat Etiquettes:** basé sur la table Clients.



Nous allons créer un état qui permet d'imprimer des étiquettes portant le nom et l'adresse de clients.

Pour créer un nouvel état, il faut avant tout cliquer sur l'onglet des états, puis sur le bouton Nouveau (New). La première fois que l'on crée un état, on peut visiter les différents Assistants disponibles. On choisit l'Assistant Etiquettes (Label Wizard). La table à utiliser est la table Clients. La première étape suivante consiste à choisir le type d'étiquettes utilisées. Mais les types ne correspondent pas toujours aux étiquettes disponibles dans le commerce. Une bonne méthode consiste à rechercher dans la liste proposée le type le plus proche des étiquettes possédées, puis, au final, de modifier manuellement les paramètres de longueur et largeur obtenus par l'Assistant. L'étape suivante n'est qu'une question de mise en forme. L'étape suivante consiste en la composition du texte de l'étiquette : les champs apparaissent dans la zone de gauche, et le reste du texte est à entrer à partir du clavier. Voici un exemple de composition :

Vous trouverez dans ce colis les albums que vous avez commandés.  
 {Prénom Client} {Nom Client}  
 {Adresse No} {Adresse Rue}  
 {Adresse CP} {Adresse Commune}

Le tri (Sort) se fera sur le nom du client. L'état est sauvé sous le nom « Etat Etiquettes ».

Après un premier aperçu du résultat, on passe en mode création afin de personnaliser le résultat. On peut par exemple allonger ou rétrécir la longueur et la largeur de l'étiquette en draguant le long des bords de la surface. On peut délimiter les bords des étiquettes au moyen d'une bordure. On appelle pour cela l'outil Rectangle. Le rectangle dessiné voile le contenu de l'étiquette. Il faut modifier la propriété **Style de Fond (Back Style)** en choisissant Transparent. Il est toujours possible de modifier le format des caractères de chacune des zones de texte (en mettant par exemple le texte introductif en italique et les coordonnées en gras).

Notez que cet état affiche (et donc imprime) des étiquettes pour tous les clients, ce qui n'est pas très réaliste. Pour n'afficher l'étiquette que d'un client spécifique, nous avons l'Etat Etiquettes Parametre, basé sur la requête "Rq Etiquettes Parametre". Lors de l'ouverture de l'état, il suffit de préciser le nom du client pour lequel l'étiquette doit être imprimée (ou par exemple saisir "a\*" pour avoir tous les clients dont le nom commence par a).

## 8.2. Les états de regroupements

L'Assistant Etat (Report Wizard) sert à organiser les données en groupes. On peut calculer la somme de chaque groupe et celle de tous les groupes.

**Etat Ventes du mois:** basé sur la requête "Rq Ventes Mois".

Date Vente	Série	N° Titre Tome	En stock	Nom Client	Prénom
<b>Tuesday, September 12, 2000</b>					
	Akira	2 Cycle Vivans	20	Beckett	Barbara
	XIII	1 Le Jour du soleil noir	16	Galy	Géraldine
	XIII	1 Le Jour du soleil noir	16	Derivet	Daniel
Nombres de ventes du jour: 3					
<b>Wednesday, September 13, 2000</b>					
	XIII	1 Le Jour du soleil noir	16	Beckett	Barbara
	XIII	1 Le Jour du soleil noir	16	Abel	Albert

Cet état indique les albums vendus pendant le mois. On se base sur la requête « Rq Ventes Mois » traduite sous forme d'état.

Pour créer cet état, on clique sur le bouton Nouveau (New). On choisit l'Assistant état (Report Wizard). La requête à utiliser est la requête « Rq Ventes du Mois ». Tous les champs sont à utiliser. Les données sont à trier par date de vente. Il n'est pas nécessaire d'ajouter d'autres niveaux de regroupement (grouping level). Il n'est pas nécessaire de trier les données sur un des champs restants (mais on peut). La suite n'est qu'une question de mise en forme.

Lorsque l'Assistant demande des niveaux de regroupement, il est possible de préciser le type de regroupement désiré. Par exemple, les dates/heures peuvent être regroupées par jour, par semaine, par mois, par trimestre, ... Lorsque l'Assistant demande un tri dans un groupe, il est possible d'ajouter des opérations de regroupement dans ce groupe. Ces opérations sont la somme, la moyenne, le minimum, le maximum.

Une fois l'état créé, on le personnalise à partir du mode création (modification des couleurs, polices, largeur des zones de texte,...). Cela est nécessaire puisque, par exemple, les étiquettes d'en-tête sont trop larges pour que le texte s'inscrive en entier. On change aussi le titre de l'état par un titre plus explicite quant à sa fonction. Lorsque un album est répété plusieurs fois de suite, ses données ne s'affichent qu'une fois. Par exemple, pour le champ [N° Tome],

2		2
1		1
1		
1		
2		2
1		1
1		
3		3

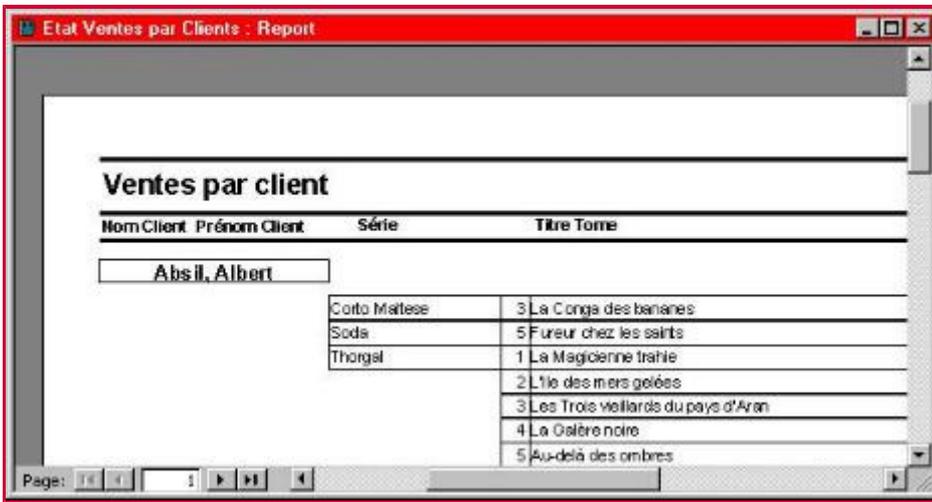
devient

ce qui n'a pas de sens. Il faut donc modifier les propriétés des champs [N° Tome], [Titre Tome] et [Nb Exemplaires] (mais pas [Série] sinon le regroupement devient inutile). La propriété **Cacher les Doublons (Hide Duplicates)** doit recevoir la valeur Non (No).

### 8.3. Les autres états

A ce stade, on ferme <08-9eArt-trrfm.mdb> et on ouvre <08-9eArt.mdb>. Ce fichier contient la base de données complète. On ne crée pas les états qui suivent. Leur création étant semblable à une des macros rencontrées ci-dessus, on ne les décrit que très brièvement.

**Etat Ventes par clients:** basé sur la requête Rq Ventes par clients.

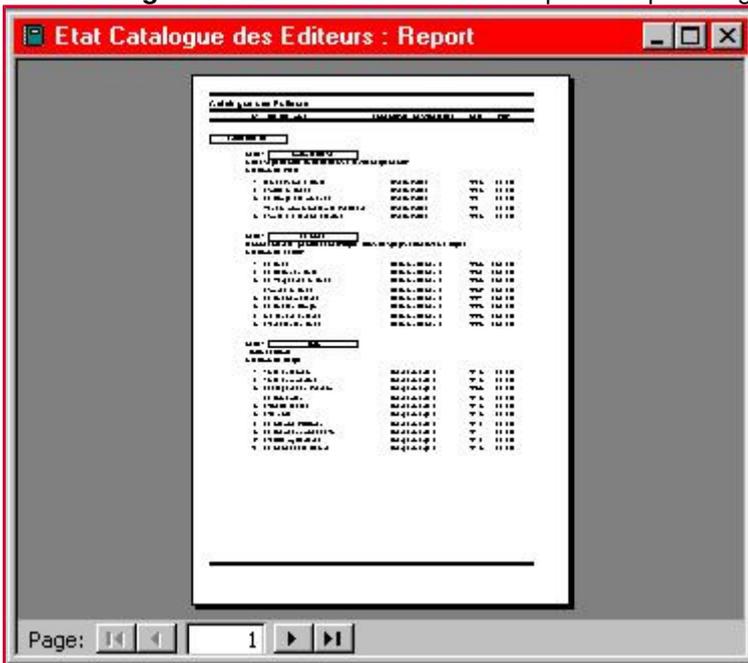


On trie cette fois les ventes en les regroupant par client. Les deux contrôles [Nom Client] et [Prénom Client] sont rendus invisibles et remplacés par une zone de texte dont la source est l'expression **= [Nom Client] & ", " & [Prénom Client]**. Il s'agit d'une concaténation de deux champs, avec une virgule et un espace entre le nom et le prénom : les noms de champs sont entre crochets, les textes invariants sont entre guillemets, et le symbole "&" est utilisé comme opérateur de concaténation.

Dans les sous-groupes, les tomes sont triés selon la série, puis le numéro du tome.

Dans le pied de groupe [Nom Client], la formule **=Compte([N° Tome]) (=Count([N° Tome]))** calcule le nombre d'achats par client. Pour expliquer ce compte, la source de la zone de texte du pied de groupe reçoit l'expression **"Nombre d'achats pour "&[Prénom Client]& " "&[Nom Client]&":"** (cette expression peut être obtenue à l'aide du générateur d'expressions; le symbole "=" doit cependant être mis manuellement).

**Etat Catalogue des Editeurs:** basé sur la requête "Rq Catalogue des Editeurs".



Les enregistrements sont regroupés selon l'éditeur, puis selon la série. Le tri se fait ensuite sur le champ [N°Tome]. Comme pour les formulaires précédents, les modifications du formulaire ne concernent que la mise en forme des contrôles: position, longueur, bordure, police, concaténation de champs, ...

## 9. L'outil de compactage Access

Pour des performances optimales, vous devez compacter et réparer régulièrement les bases de données Access. Faites cela par exemple lorsque vous avez fini de construire un objet, ou à chaque fois que vous fermez votre base de données. Dans le menu Outils, pointez sur Utilitaires de base de données, puis cliquez sur Compacter et Réparer une base de données (Tools/Database Utilities/Compact And Repair Database).

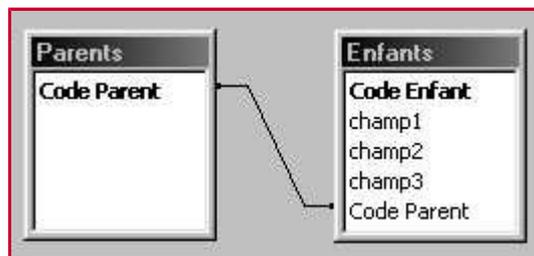
Vous serez peut-être surpris de la place mémoire récupérée après un compactage. Mais Access ne supprime rien de votre travail, et ne fait que libérer de l'espace mémoire inutilisé après la création de vos objets. Rien n'est donc perdu.

Notez aussi que cette méthode a comme avantage de garder "lisible" la base de données; il est inutile de "décompacter" celle-ci. Il ne s'agit pas d'un compactage classique au sens où on ne crée pas un nouveau fichier appelé "archive" (comme avec les logiciels de compactage, tel Winzip).

## 10. Les F.A.Q. (Frequently Asked Questions, La Foire Aux Questions)

1. **question:** Comment se définit une relation de type "un à plusieurs"? Que signifient les expressions "table parent" et "table enfant"?

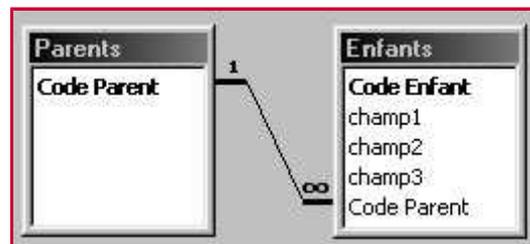
**solution:** A un enregistrement de la table parent correspondent plusieurs enregistrements dans la table enfant. Voici une simulation la clé primaire de la table parent est le "code parent" et la clé primaire de la table enfant est le "code enfant". Le champ commun dans les deux tables est le "code parent". La relation s'établit en cliquant de gauche à droite (dans ce cas-ci), de "code parent" vers "code parent"



2. **question:** Quelle est cette notion "d'Intégrité référentielle"? Pourquoi ne vois-je pas les "1" et les "infinis" sur ma relation?

**solution:** Pour voir les "1" et les "infinis", il faut appliquer l'intégrité référentielle. On bénéficie alors de ses avantages, mais ceux-ci s'avèrent parfois, surtout au début, des inconvénients: Access empêche alors

- d'ajouter de nouveaux enregistrements dans une table reliée pour laquelle il n'existe pas d'enregistrement "parent",
- de procéder à la modification de valeurs dans une table source qui entraînerait des enregistrements "orphelins" dans une table reliée,
- de supprimer des enregistrements d'une table source dans le cas où il existe des enregistrements "enfants" correspondants.



3. **question:** Comment supprimer ces tables non désirées qui apparaissent dans les relations à l'ouverture de la fenêtre des relations ou des requêtes?

**solution:** Cliquer sur la table et appuyer sur Delete n'est pas suffisant; la table réapparaît à l'ouverture suivante. Pour la supprimer définitivement, il faut, dans la fenêtre des relations, d'abord supprimer toutes les relations impliquant cette table (cliquer bien précisément sur le lien), puis la table concernée.

4. **question:** Pourquoi dois-je me contenter d'un "mini-encodage" après la création de mes tables? N'est-il pas plus efficace d'encoder directement toutes les données?

**solution:** Lorsque les tables et les relations sont créées, on encode une douzaine d'enregistrements par table. Après le "mini-encodage", on crée les requêtes, formulaires et états. Un petit nombre d'enregistrement permettra de vérifier "à l'oeil" s'ils fournissent les bons résultats. Si les résultats ne sont pas ceux attendus, cela peut être dû à une erreur de structure des tables, où à un encodage erroné. Il est alors encore temps de passer aux modifications. Les autres enregistrements seront encodés ultérieurement dans les formulaires. Un encodage terminé sur une structure à revoir et à modifier signifie beaucoup d'heures de perdues.

5. question: Pour sauvegarder régulièrement mes données, je dois donc cliquer sur l'outil "enregistrement" représenté par une disquette?

problème: Sous la plupart des logiciels, le bouton disquette permet de sauvegarder le fichier dans l'état où il se trouve.

solution: Sous Access, la sauvegarde des informations encodées se fait lorsqu'on passe d'un champ à un autre. Il est donc inutile de cliquer sur ce bouton lors de l'encodage. Par contre, il permet de sauvegarder les requêtes, formulaires et états à tout moment de leur création ou de leur modification.

6. question: J'ai supprimé des enregistrements. Il y a donc un "trou" dans la suite des nombres de mon champ autonumber. Comment mettre à jour ces valeurs?

solution: Il n'est pas possible de "combler ce trou" sous Access. Cela n'est pas gênant puisque les champs de type autonumber s'emploient exclusivement pour des codes clés primaires, et que ces codes ne doivent pas être visibles par l'utilisateur. Il est toutefois possible de rectifier la suite de nombres à l'aide d'un export vers Excel.

7. question: Mon critère de requête est une longue expression. Comment le voir en entier sans donner une largeur démesurée à la cellule?

solution: Il existe un outil très simple, le zoom, pour voir le contenu d'une cellule de requête, de table, de formulaire, ... On l'obtient en faisant MAJ+F2 (Appuyez sur la touche F2 tout en maintenant la touche de majuscules enfoncée.)

8. question: Je teste une requête avec un critère simple, bien choisi pour me fournir des résultats. Pourquoi la requête me donne-t-elle alors 0 résultat?

solution: La cause est très fréquemment une simple faute d'orthographe, soit dans le critère, soit dans l'encodage des données. Vérifiez les mots au pluriel, l'orthographe des noms propres, tapez bien 1 espace entre deux mots et non 2, ... Il est même arrivé qu'une requête fournisse 0 résultat car les étudiants n'avaient encore rien encodé...

9. question: Pourquoi ma requête multitable ne me donne-t-elle pas tous les résultats attendus, ou même pas de résultat?

problème: Lorsque l'on crée une requête portant sur deux tables liées par une relation de un à plusieurs, il arrive que tous les enregistrements de la table parent n'apparaissent pas.

solution:

- La première chose à vérifier est l'orthographe (cf réponse ci-dessus).
- Vérifiez aussi que le lien soit bien établi entre deux champs communs (pas une date avec un nom par exemple).
- Peut-être le "mini encodage" n'était-il pas suffisant. Y-a-t'il assez d'enregistrements dans la table parent? Et dans la table enfant?
- Par défaut, Access n'affiche que les enregistrements parents qui ont un équivalent dans la table enfant. Pour afficher tous les parents, même ceux sans enfant, il faut faire appel à la propriété de jonction (Join Properties) en double-cliquant sur la relation et en appuyant sur le bouton Type de joint... (Joint Type...). Dans la boîte affichée, vous sélectionnez la deuxième option (la troisième option pour afficher plutôt les enregistrements enfants ET les orphelins).



10. **question:** A l'inverse de la question précédente, pourquoi ma requête multitable me donne-t-elle plus de résultats que ceux attendus?

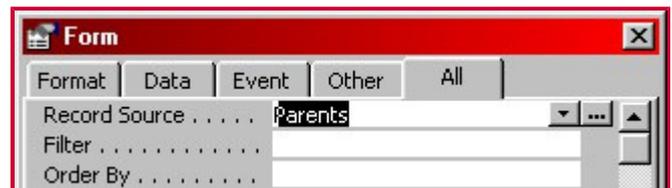
**problème:** Il arrive parfois qu'une requête donne plus de résultats que ceux attendus. On reconnaît qu'il y a une erreur en découvrant des enregistrements absurdes.

**solution:** Il y a beaucoup de probabilités que les tables de la requête ne soient pas liées. Access combine alors tous les enregistrements de la première par tous les enregistrements de la seconde. Pour corriger cet effet, ajouter, le cas échéant, dans la requête toutes les tables intermédiaires entre les tables liées. S'il n'existe pas de lien direct entre les tables, vous pouvez toujours créer un lien temporaire dans la requête, mais cette solution est peu conseillée.

11. **question:** Pourquoi mon formulaire ne reconnaît-il plus la requête qui contient ses données? Pourquoi mon sous-formulaire est-il remplacé par un rectangle blanc?

**problème:** Le formulaire retient comme "source" le nom de la requête ou de la table qui contient les données. Si vous changez le nom de la requête ou de la table, le formulaire ne fait pas automatiquement le changement de source. Il ne retrouve plus les données à afficher, et affiche soit un message d'erreur, soit un cadre blanc.

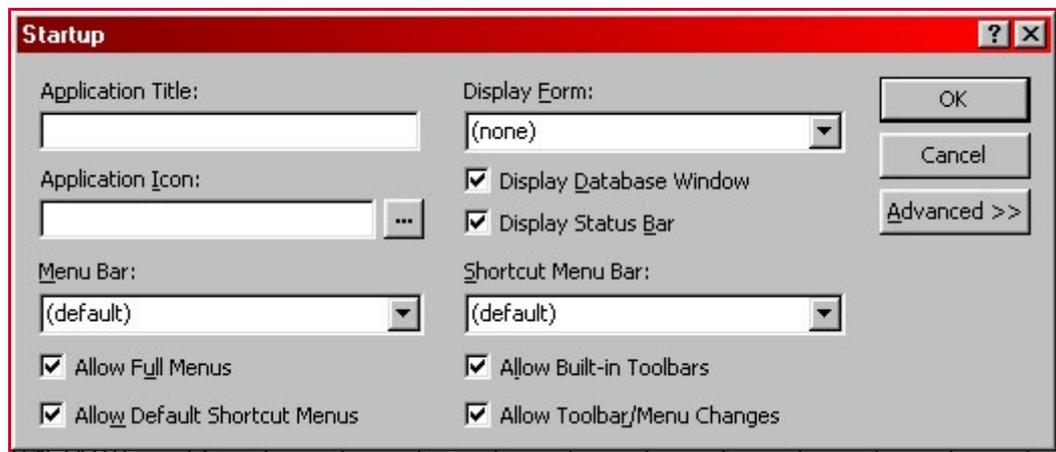
**solution:** Choisissez au plus tôt les noms de vos tables, champs, requêtes, formulaires, ... et n'en changez pas. Si vous êtes obligé de les changer, n'oubliez pas de modifier la propriété "source" dans tous les formulaires, états ou macros qui utilisent cette requête.



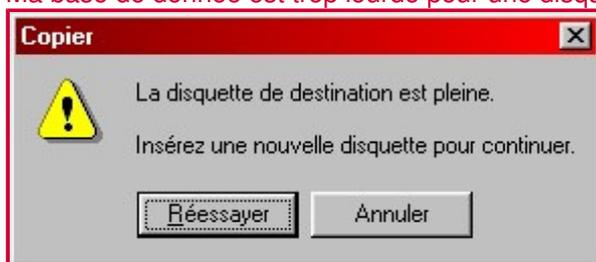
12. **question:** Comment faire apparaître un menu de démarrage à l'ouverture de la base de données?

**solution:**

- La macro autoexec se lance à l'ouverture de la base de données. Créez donc une macro autoexec qui aura pour action d'ouvrir le formulaire de démarrage de votre choix.
- Une autre possibilité est d'utiliser la commande démarrage du menu outils (Tools/Startup). Dans la boîte de dialogue, demandez d'afficher (Display Form) le formulaire de votre choix.



13. question: Ma base de donnée est trop lourde pour une disquette. Comment la transporter?



problème: Les bases de données, pour peu qu'elles contiennent quelques images, voient leur taille se compter en méga-octets plutôt qu'en kilo-octets. Mais une disquette ne peut transporter que 1440 kilo-octets (1.38Mo).

solution:

- La première chose à faire est de compacter la base de données avec le compacteur interne d'Access. Utilisez le menu Outils/Utilitaires de la base de données/Compacter et Réparer la base de données (Tools/Database Utilities/Compact and Repair Database). Cette méthode a comme avantage de garder "lisible" la base de données; il est inutile de "décompacter" celle-ci. La seule chose que l'utilitaire fait est en quelque sorte de supprimer des espaces blancs, des espaces de travail devenus inutiles après création de vos objets. Il ne s'agit pas d'un compactage classique au sens où on ne crée pas un nouveau fichier appelé "archive" (comme avec les logiciels de compactage).
- Si ce résultat n'est pas suffisant, il faut employer un logiciel de compactage externe à Access. Ceux utilisés en salles Renaissance sont WinZip et RAR1.55. Grâce à eux, il est possible de découper le fichier en différentes archives afin de faire tenir le travail sur plusieurs disquettes. Pour l'utilisation de ces logiciels, référez-vous aux notes de cours sur Windows sur le serveur des salles informatiques ou sur notre site web ([www.ulb.ac.be/soco/matsch/info-d-203](http://www.ulb.ac.be/soco/matsch/info-d-203)).
- Les ordinateurs des salles Renaissance 1 et 2 sont équipés de ports USB (à l'arrière) et de graveurs. Avec une clé USB ou un CD réinscriptible (ou un paquet de CDs non-réinscriptibles) la taille de fichiers ne devrait plus être un problème.

14. question: Comment faire lorsque rien ne sort de l'imprimante?

problème: Vous avez envoyé un document à l'imprimante, et rien ne se passe...

solution:

- La bonne réponse est de chercher pourquoi rien ne s'est passé et résoudre le problème. La mauvaise réponse est de relancer l'impression en espérant que cela marche. Certains utilisateurs continuent d'envoyer leur travaux en espérant qu'ils seront finalement imprimés. Le résultat est plutôt gênant quand quelqu'un résout finalement le problème (imprimante éteinte, bourrage papier, câble débranché, ...) et regarde 20 versions du même travail s'imprimer.

- En salle Renaissance, il est possible aussi que votre travail soit imprimé sur une imprimante d'une autre salle. Pour contrôler votre impression, ne cliquez pas sur le bouton représentant l'imprimante, mais passer par les menus: Fichier/Imprimer (File/Print), et choisissez l'imprimante de votre salle.