

# Créer une application de gestion de bandes dessinées



par Thierry GASPERMENT ([arkham46.developpez.com/](http://arkham46.developpez.com/))

Date de publication : 19/03/10

Dernière mise à jour : 19/03/10

Apprenez à créer pas à pas une application Access de gestion de bandes dessinées. Ce tutoriel s'adresse aux débutants Access (et aux autres ;o) ).

I - Introduction.....	3
II - Spécifications fonctionnelles de l'application.....	3
II-A - Les catalogues.....	3
II-B - L'interface.....	3
III - Version ciblée.....	4
IV - Analyse par Merise.....	4
IV-A - Quelques mots de vocabulaire.....	4
IV-B - Présentation de l'outil utilisé : AnalyseSI.....	5
IV-C - Le modèle conceptuel de données (MCD).....	5
IV-D - Le modèle logique de données (MLD).....	12
IV-E - Le modèle physique de données (MPD).....	13
V - Création de la base de données et d'une première table.....	13
V-A - Avec Access 2007.....	13
V-B - Avec Access 2000 à 2003.....	14
VI - Création des tables en fonction du MLD.....	14
VI-A - Création des tables de référentiel.....	15
VI-B - Création des tables applicatives.....	17
VI-B-1 - Création de la table bds.....	17
VI-B-2 - Création d'une clé étrangère.....	17
VI-B-3 - Le travail de l'assistant.....	20
VI-B-4 - Création de la table exemplaires.....	21
VI-C - Création des tables issues des associations.....	22
VI-D - Vérification visuelle dans Access.....	23
VI-E - Téléchargement de la base de données.....	23
VII - Ajout de contraintes.....	23
VIII - Ajout de données de tests.....	25
IX - Création d'une requête.....	30
IX-A - Tri sur une clé étrangère.....	32
IX-B - Tri sur un champ texte contenant des nombres.....	34
X - Modification des jointures des relations.....	36
XI - Par où commencer?.....	39
XII - Maquette de l'interface.....	40
XIII - Conclusion.....	40
XIV - Téléchargement.....	41

## I - Introduction

Je vous invite à suivre **pas à pas** la création d'une base de données Access.  
Nous commencerons par quelques **spécifications**, puis l'**analyse** et la construction des **tables**.  
Nous construirons également l'interface à l'aide de **formulaires** et d'**états**.  
Lorsque cela serait nécessaire, nous écrivons du code **VBA**.

## II - Spécifications fonctionnelles de l'application

Avant toute chose, écrivons les spécifications de notre application.  
Ces spécifications décrivent les besoins utilisateurs de notre application.  
Aucune notion technique n'est définie ici.  
Pour ce tutoriel, nous allons définir des spécifications simplifiées.

### II-A - Les catalogues

L'application devra permettre à l'utilisateur de gérer un catalogue de bande dessinées.  
Chaque bande dessinées pourra contenir les informations suivantes :

- son titre;
- un commentaire libre;
- une note de 1 à 10;
- le numéro de tome;
- une ou plusieurs images;
- la série à laquelle elle appartient;
- les auteurs : dessinateur(s) et scénariste(s);

L'utilisateur devra pouvoir saisir ou modifier un catalogue de séries.

Chaque série pourra contenir les informations suivantes :

- son titre;
- un commentaire libre;
- une note de 1 à 10.
- une ou plusieurs images;

L'utilisateur devra pouvoir saisir ou modifier un catalogue d'auteurs.

Chaque auteur pourra contenir les informations suivantes :

- son nom et prénom;
- une note de 1 à 10.
- ses rôles (scénariste et/ou dessinateur);

L'utilisateur devra pouvoir saisir ou modifier un catalogue d'exemplaires possédés.

Chaque exemplaire pourra contenir les informations suivantes :

- la bande dessinée dont c'est un exemplaire;
- la date d'achat;
- l'état (mauvais, moyen, bon ou neuf);
- le format (petit format, grand format, tirage de tête).

### II-B - L'interface

L'application devra présenter sous forme de tableau la liste des bandes dessinées du catalogue.

Il devra être possible d'afficher :

- soit tous les exemplaires;
- soit seulement les exemplaires possédés;
- soit seulement les exemplaires manquants;

Différents filtres et recherches devront être disponibles :

- par titre de bande dessinée ou série (complet ou partiel);
- par auteur;
- par note (bd, série ou auteur);

- par auteur;
- par date d'achat;
- par état;
- par format.

Chacune de ces listes doit être affichable à l'écran et imprimable.

Les différents catalogues présentés au chapitre précédent doivent être modifiables afin d'ajouter, modifier ou supprimer des entrées.

### III - Version ciblée

Nous allons dans un premier temps cibler la version Access 2000.

Pour cette version, nous ne pourrions donc pas utiliser entre autres :

- le ruban;
- les champs pièces jointes pour les images.

Le développement peut tout de même être réalisé avec une version plus récente d'Access (2003 ou 2007 par exemple).

### IV - Analyse par Merise

Maintenant que nous connaissons les besoins de l'application, comment les intégrer à notre base de données Access?

La première chose à faire est de créer les tables nécessaires au stockage des données.

Se pose alors une question : de quelles tables avons-nous besoin?.

S'il apparaît évident que nous avons besoin d'une table pour contenir le catalogue de bande dessinées, d'autres tables sont moins évidentes.

Par exemple : comment gérer les auteurs?

Doit-on créer une table par rôle : c'est-à-dire une table pour les scénaristes et une pour les dessinateurs?

Ou une seule table pour tous les rôles?

Comment faire pour lier le catalogue aux auteurs?

Est-il nécessaire de créer une table supplémentaire pour associer chaque bande dessinée à ses auteurs?

Pour nous aider à répondre à toutes ces questions, nous allons faire une analyse à l'aide de la méthode **Merise**.

Cette phase de conception est très importante : une bonne analyse facilite ensuite la création des objets (requêtes, formulaire, états, ...) et la maintenance.

 *Consultez les tutoriels de **la page cours Merise**.*

 *Pour vos interrogations en phase d'analyse, utilisez le **le forum Conception Access**.*

#### IV-A - Quelques mots de vocabulaire

Si vous n'avez pas le courage ou le temps de lire les tutoriels au sujet de Merise, voici un petit résumé du vocabulaire à connaître pour suivre ce tutoriel :

- Modèle conceptuel de données (MCD) : définitions des concepts de l'application;
- Modèle logique de données (MLD) : définitions des tables de l'application, sans cibler de système de base de données en particulier;
- Modèle physique de données (MPD) : définitions des tables de l'application, en prenant en compte les particularités d'un système de base de données.
- Entité : Élément de base du MCD (peut être une bande dessinées, un état, une personne, ...);
- Association : Lien entre plusieurs entités, définie souvent par un verbe (appartient à, est un, ...);
- Cardinalité : Nombre d'occurrences possibles d'une association (0-n, 1-n, ...);
- Attributs : Propriété d'une entité ou d'une association (un nom, un identifiant, un commentaire,...);

Nous verrons plus en détail ces notions durant notre analyse.

## IV-B - Présentation de l'outil utilisé : AnalyseSI

Nous allons utiliser l'application **AnalyseSI**.

Cet outil est gratuit et, même s'il n'est plus maintenu, nous aidera efficacement à faire notre analyse.

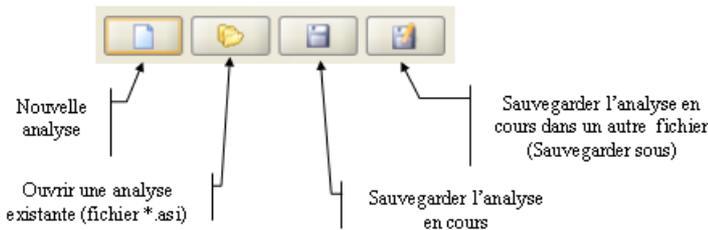
 *Voir les autres **outils de conception**.*

Téléchargez et décompressez le fichier de l'application *AnalyseSI*.

Exécutez ensuite le fichier **analysesi.exe** ou **analysesi\_opengl.exe**.

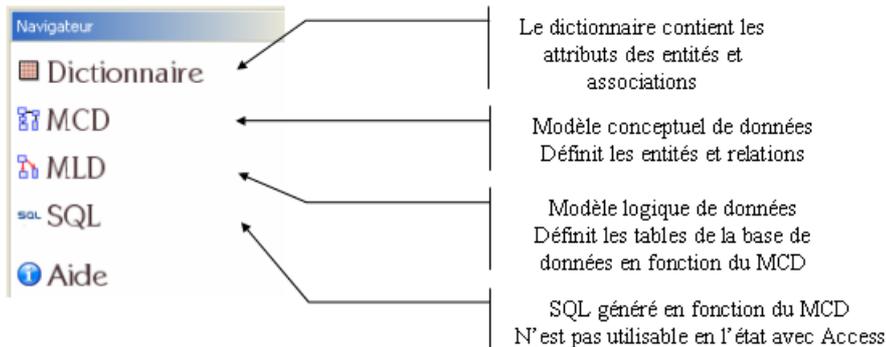
L'outil s'ouvre sur une nouvelle analyse.

Les boutons en haut permettent les opérations habituelles (ouverture, sauvegarde, ...).



 **Attention : l'outil ne propose pas de sauvegarde à la fermeture; pensez bien à sauvegarder votre analyse avant de le fermer.**

Le navigateur à gauche vous permet de changer de vue :



## IV-C - Le modèle conceptuel de données (MCD)

Nous allons créer directement les entités du MCD, et générer les entrées du dictionnaire au fur et à mesure.

 *Notez qu'il est possible de créer d'abord le dictionnaire de tous les attributs avant de créer les entités.*

*Les attributs non encore utilisés seront alors proposés lors de la modification d'une entité ou association.*

Cliquez sur le bouton MCD à gauche.

Le MCD est pour l'instant vide, nous allons y ajouter les entités.

Les entités représentent les informations dont nous avons besoin.

Par exemple, une entité peut être une bande dessinée, un auteur, ...

Nous nous aidons de notre spécification fonctionnelle pour savoir quelles entités ajouter au modèle.

Pour ajouter une entité, cliquez sur le bouton **Ajouter une entité**  en haut.

L'entité ajoutée est vide :

Entite 1

Double-cliquez sur l'entité pour l'éditer.

Notre première entité représente le catalogue de bandes dessinées; appelons la **BDs** en modifiant le nom en haut de la fenêtre d'édition.



Ajoutons maintenant les attributs de cet entité :

Chaque entité doit posséder un identifiant unique.

Nous ajoutons donc tout d'abord un attribut nommé **id\_bd** de type **uniqueidentifiant**.

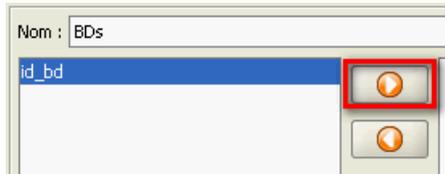
Dans la fenêtre d'édition de l'entité, saisissez en bas le nom de l'attribut et son type de données.



Cliquez sur le bouton **Ajout** pour ajouter l'attribut au catalogue.

L'attribut est alors disponible dans la liste de gauche.

Cliquez sur le bouton **Ajouter une nouvelle information** (la flèche droite au milieu) pour ajouter cet attribut à l'entité.



L'attribut s'affiche alors dans la liste de droite.

Il disparaît de la liste de gauche car un attribut ne peut être utilisé qu'une seule fois dans le modèle.

Une bande dessinée doit également avoir un titre, ajoutons un attribut **titre\_bd**.



Le titre est un **varchar** de taille **255**, c'est à dire une chaînes de caractère de taille 255.

Comme il n'y a pas de lien direct entre AnalyseSI et Access, le type de données est saisi à titre informatif.

Notez qu'on nomme l'attribut **titre\_bd**, ce qui permettra de le distinguer des autres titres du modèle.

De la même manière, on ajoute un attribut pour le commentaire, nommé **commentaire\_bd** (type varchar 255 par exemple).

Puis un attribut **note\_bd** de type **bigint** (entier long) qui contiendra la note donnée à la bande dessinée de 1 à 10.

Et enfin un attribut **tome\_bd** de type **varchar (10)** pour le numéro de tome de la BD.

Ce tome est une chaîne de caractères, ce qui nous permet de saisir des numériques mais aussi des caractères, par exemple pour des hors-série : HS01, HS02, ...

Nous avons ajouté les attributs les plus simples :

BDs
<u>id_bd</u>
titre_bd
commentaire_bd
note_bd
tome_bd

Viennent ensuite dans la spécification les images affectées à la bande dessinée.

On a spécifié qu'on souhaitait pouvoir définir une ou **plusieurs** images.

Si on se limitait à une image unique par bande dessinée, il suffirait d'un attribut **image\_bd**.

Pour ajouter plusieurs images, on pourrait imaginer plusieurs attributs **image\_bd1**, **image\_bd2**, ...

Ce modèle assez simpliste cache une grave erreur.

En effet si on a défini 2 attributs images et qu'on souhaite par la suite pouvoir en gérer 3, il faudrait revoir le modèle.

Ajouter un attribut est assez simple, mais implique de nombreuses modifications dans l'application :

- ajout d'un champ dans la table
- ajout de ce champ aux requêtes

- ajout de ce champ dans les formulaires d'affichage
- modification du code VBA si nécessaire
- ...

Ceci n'est pas envisageable, on ne doit donc pas créer plusieurs attributs porteurs d'une même information. Il faut alors définir pour les images une autre entité.

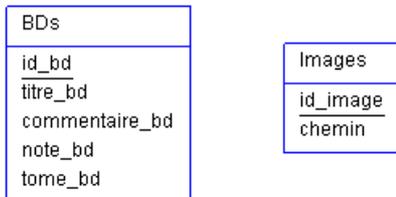
Créez une nouvelle entité et nommez-la **Images**.

Pour cette entité, ajoutez un attribut **id\_image** de type **uniqueidentifiant**.

Cet attribut est la clé de l'image qui permet de l'identifier de manière unique.

Puis ajoutez, toujours à l'entité Images, un attribut **chemin** (de type varchar 255).

Nous voici donc munis de deux entités :



Il nous faut maintenant lier ces entités; c'est le rôle des associations.

Pour ajouter une association, cliquez sur le bouton **Ajouter une association**  en haut.

Une association se traduit par un verbe : *appartient à, est de type, ...*

Ici par exemple, on traduit notre association par : **A pour image de bd.**

Double-cliquez sur l'association pour l'éditer et modifier son nom :



Ensuite il faut créer les liens entre l'association et les entités, ainsi que leur cardinalité.

Les cardinalités sont le nombre d'occurrences possible du lien.

Elles se définissent de chaque côté de l'association.

Par la suite, elle permettent de déduire le modèle logique de données (MLD).

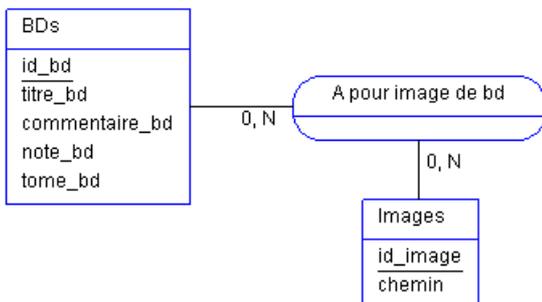
Pour ajouter un lien, cliquez sur le bouton **Ajouter un lien**  en haut.

Créez tout d'abord le lien entre l'entité **Bds** et l'association **A pour image de bd**.

Pour cela, après avoir cliqué sur le bouton **Ajouter un lien**, cliquez sur l'entité et faites glisser la souris jusque l'association sur laquelle vous relâchez la souris.

Ensuite, créez un lien entre l'entité **Images** et l'association **A pour image de bd**.

Par défaut la cardinalité des liens est **0,n**.



### Comment choisir les cardinalités?

Il faut faire des petites phrases, partant de l'entité :

- une BD peut avoir entre 0 et n images, donc une cardinalité **0,n** entre **Bds** et **A pour image de bd**;
- une image peut être utilisée dans 0 à n bandes dessinées, donc une cardinalité **0,n** entre **Images** et **A pour image de bd**;

Avec ces 2 petites phrases, on a donc décidé que :

- une bande dessinée pourra ne pas avoir d'images; si elle en a, elle pourra en avoir une ou plusieurs;
- une image pourra ne pas être utilisée, ou être utilisée par plusieurs bandes dessinées.

Notez que l'analyse ne permet pas de définir des cardinalités autres que 0,1 et n.

Si on souhaite limiter à 5 images par exemple, cela se fera plus tard lors de la conception de la base de données. On se rend compte dès maintenant qu'il faudra gérer un catalogue d'images. Les images étant maintenant gérées par l'association nouvellement créée, passons à la donnée suivante : la série à laquelle appartient la bande dessinée.

Ici encore nous n'allons pas ajouter un simple attribut pour la série.

La série n'est pas un champ texte libre.

Nous avons défini, dans les spécifications, l'utilisation d'un catalogue de séries.

Cela permettra de proposer une liste des séries pour faciliter la saisie et éviter les erreurs.

Nous allons donc créer une entité **Séries**.

Comme vu précédemment : créez une nouvelle entité, nommez-la **Séries** et ajoutez lui un attribut **id\_série** de type **uniqueidentifiant**.

Nous compléterons plus tard ces autres attributs.

Créez maintenant une nouvelle association nommée **Appartient à la série**.

Ajoutez un lien entre chacune des entités (**Bds** et **Séries**) et l'association.

Puis écrivons les phrases nous aidant à définir les cardinalités :

- une BD peut appartenir à 1 série au maximum, donc une cardinalité **0,1** entre **Bds** et **Appartient à la série**;

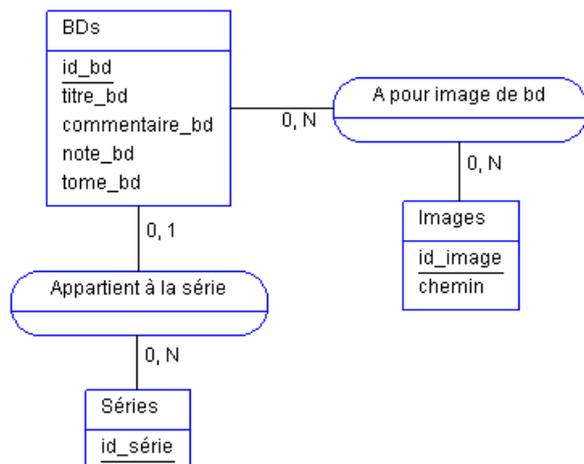
- une série peut être définie pour plusieurs BD, donc une cardinalité **0,n** entre **Séries** et **Appartient à la série**;

En effet une bande dessinée peut n'appartenir à aucune série, et ne peut appartenir à plus d'une série à la fois.

Une série quant à elle peut contenir plusieurs bandes dessinées, voir aucune si on vient de définir la série et qu'on n'a pas encore créé les BD associées.

Modifiez le lien entre **Bds** et **Appartient à la série** afin de définir la cardinalité à **0,1**.

(Double-cliquez sur un lien pour le modifier).



N'oublions pas que l'entité **Séries** n'est pas terminée.

Ajoutons lui comme défini dans la spécification :

- un attribut **titre\_série** (varchar 255).
- un attribut **commentaire\_série** (varchar 255).
- un attribut **note\_série** (bigint).

Pour les images, on procède comme pour les images de bandes dessinées.

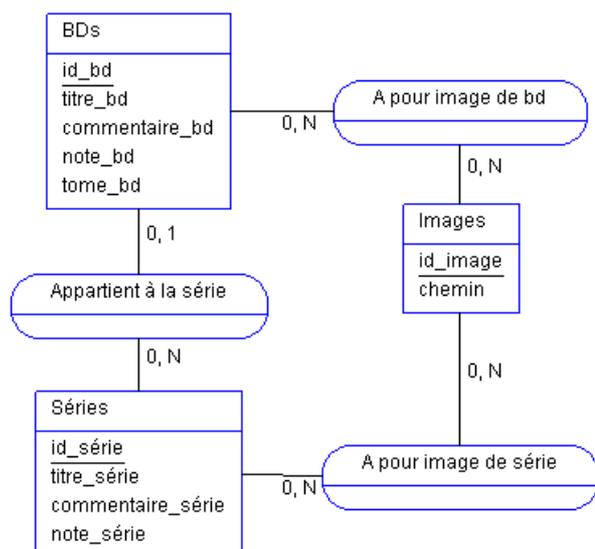
La même entité **Images** est utilisée.

Ainsi on pourra affecter la même image à une série et à une bande dessinée.

Nous créons par contre une nouvelle association nommée **A pour image de série**.

Créez un lien entre l'entité **Séries** et cette association, puis entre l'entité **Images** et cette association.

Les cardinalités de ces liens sont les mêmes que pour les images de BD : **0,n**.



Nous avons ajouté toutes les informations spécifiées pour les séries, retournons à notre entité **Bds**.

Il nous reste une information complexe : les auteurs de la bande dessinée.

Comme nous avons spécifié l'utilisation d'un catalogue d'auteurs, il seront naturellement à choisir parmi ce catalogue. Les auteurs ne seront pas des attributs de l'entité **Bds**, mais une association entre une entité **Auteurs** et l'entité **Bds**.

Menons une petite réflexion sur ces auteurs :

- nous avons défini 2 rôles : scénariste et dessinateur; il est possible qu'on souhaite par la suite ajouter de nouveaux rôles (coloriste par exemple) .
- un auteur peut être parfois scénariste, parfois dessinateur, et parfois les deux en même temps
- on peut également ne pas connaître l'auteur d'une bande dessinée (même si c'est écrit dessus, on n'a peut-être pas la BD sous la main)
- ou alors l'utilisateur peut tout simplement ne pas souhaiter gérer les auteurs, et nous ne souhaitons pas le forcer à le faire

Pour pouvoir facilement gérer les rôles, il nous faut une entité **Roles**.

Cette entité a pour attributs :

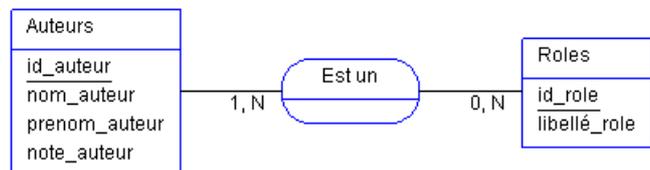
- un identifiant unique **id\_role** (uniqueidentifier);
- un libellé **libellé\_role** (varchar 50).

Les auteurs sont également une entité, nommée **Auteurs** avec comme attributs :

- un identifiant unique **id\_auteur** (uniqueidentifier);
- un attribut **nom\_auteur** (varchar 50).
- un attribut **prenom\_auteur** (varchar 50).
- un attribut **note\_auteur** (bigint).

Les auteurs sont liés au rôles par l'intermédiaire d'une association nommée **Est un** :

- un auteur doit avoir au moins un rôle, et peut en cumuler plusieurs, donc une cardinalité **1,n** entre **Auteurs** et **Est un**;
- un rôle peut être défini pour plusieurs auteurs, donc une cardinalité **0,n** entre **Rôles** et **Est un**;



Maintenant, il faut ajouter une association entre **Bds** et **Auteurs**.

On pourrait penser ajouter simplement une association pour chaque rôle, par exemple :

- une association **Est dessiné par**;
- une association **Est scénarisé par**.

Mais comment pourrait-on ajouter alors un nouveau rôle, par exemple le coloriste?

Il faudrait ajouter une nouvelle association, ce qui enfreint les règles.

On ne doit pas avoir à modifier le modèle de données pour ajouter une telle information.

Il faut ajouter une seule association, nommée **Est réalisé par**, qui sera liée à la fois aux auteurs et aux rôles.

On peut le faire avec une seule association, inutile donc d'en créer deux.

Créons cette association, nommée **Est réalisé par**.

Ajoutez un lien entre **Bds** et cette association.

Une bande dessinée peut être réalisée par plusieurs auteurs.

D'autre part, on souhaite que l'utilisateur puisse ne pas renseigner d'auteur.

Donc la cardinalité de ce lien est **0,n**.

Ajoutez maintenant deux liens entre cette association et les entités **Auteurs** et **Rôles**.

- un auteur peut avoir réalisé entre 0 et n BDs, donc une cardinalité **0,n** entre **Auteurs** et **Est réalisé par**;

- un rôle peut être associé à entre 0 à n BDs, donc une cardinalité **0,n** entre **Rôle** et **Est réalisé par**;

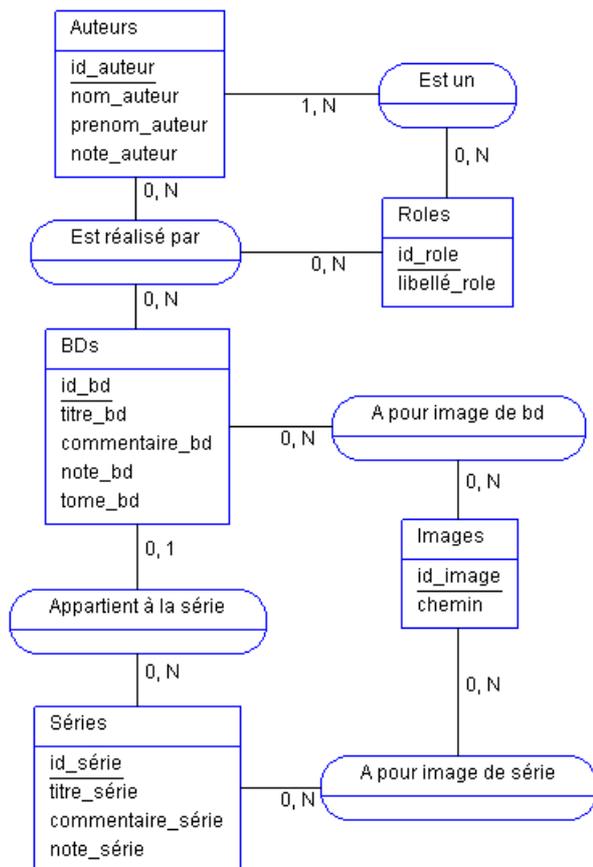
Il peut paraître étrange de définir par exemple qu'un auteur peut n'avoir réalisé aucune bande dessinée, mais cela nous permettra de créer dans le catalogue des auteurs sans avoir besoin de créer tout de suite les bande dessinées associées.

A noter : il devrait y avoir une contrainte à ce niveau.

En effet le couple auteur + rôle choisi dans l'association **Est réalisé par** doit être couvert par l'association **Est un**.

Il ne doit pas être possible par exemple de définir en dessinateur d'un BD un auteur qui n'est que scénariste.

AnalyseSI ne permet pas de gérer ces contraintes, nous la gérons par la suite lors de la conception.



A ce stade de l'analyse, nous avons défini : le catalogue de BD, les auteurs, les séries.

Tout est associé dans les règles, il nous reste maintenant à nous occuper du catalogue d'exemplaires possédés.

Le catalogue de bandes dessinées contient toutes les bandes dessinées ajoutées par l'utilisateur.

Quant à lui, le catalogue d'exemplaires possédés va permettre à l'utilisateur de définir quelles sont les BD du catalogue qu'il possède.

Une solution simple aurait été de rajouter un attribut dans le catalogue pour définir les BD possédées.

Ce serait très simple mais n'apporterait qu'une information binaire : je possède cette bande dessinée ou pas.

Nous souhaitons ajouter plus d'information : le format de la BD, son état, la date d'achat, ...

Et de plus il doit être possible de définir plusieurs exemplaires d'une même bande dessinée.

Pour cela créons un nouvelle entité nommée **Exemplaires**.

Premier attribut à ajouter à cet entité : un identifiant unique **id\_exemplaire** (uniqueidentifier).

Ajoutez un deuxième attribut pour la date d'achat : **date\_achat** de type **datetime**.

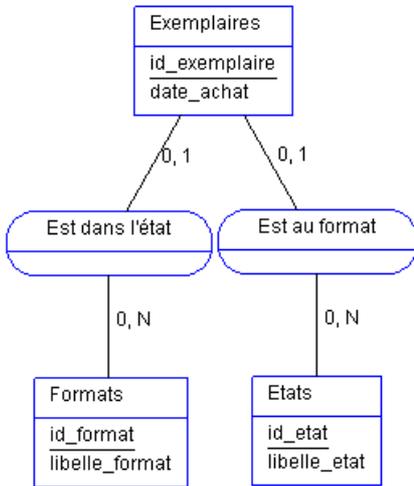
Pour gérer les formats et l'état, il nous faut de nouvelles entités nommées **Formats** et **Etats**.

Chacune de ces entités contient comme attribut :

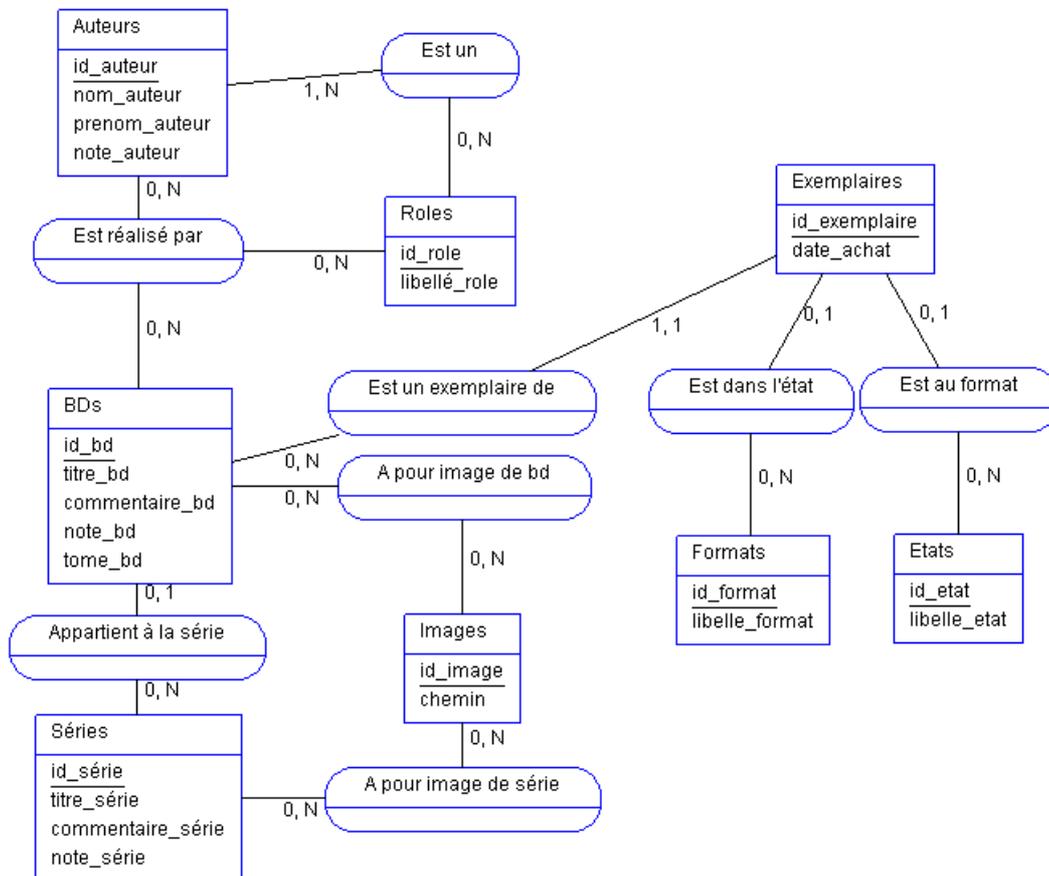
- un identifiant unique (uniqueidentifier) : **id\_format** et **id\_etat**;

- un libellé (varchar 50) : **libelle\_format** et **libelle\_etat**;

On associe un exemplaire à un format avec une association nommée **Est au format**.  
 On associe un exemplaire à un état avec une association nommée **Est dans l'état**.  
 Un exemplaire peut être dans un seul état au maximum (il ne peut pas être en bon état et en mauvais état à la fois).  
 Un exemplaire peut être dans un seul format au maximum (il ne peut pas être en petit format et grand format à la fois).  
 Les états et formats peuvent être définis pour aucun ou plusieurs exemplaires.  
 On déduit de ces trois dernières phrases les cardinalités de nos liens entre les exemplaires et les états et formats.



Reste à associer les **Bds** aux **Exemplaires** avec une association **Est un exemplaire de**.  
 - un exemplaire n'est exemplaire que d'une seule bande dessinée (et au moins une), donc une cardinalité **1,1** entre **Exemplaires** et **Est un exemplaire de**;  
 - une bande dessinée peut être définie pour aucun ou plusieurs exemplaires, donc une cardinalité **0,n** entre **Bds** et **Est un exemplaire de**;  
 Nous voici enfin arrivés à la fin de l'analyse : le MCD est terminé.



Vous pouvez télécharger le **document d'analyse (HTTP)** à ouvrir avec AnalyseSI

## IV-D - Le modèle logique de données (MLD)

 *Le modèle logique de données est proche de ce qu'on trouvera dans la base de données. Il définit, en fonction du MCD, les tables à créer pour couvrir l'analyse réalisée.*

Pour générer le MLD automatiquement avec AnalyseSI, cliquez sur le bouton **Construction**  en haut. Les entités deviennent des tables dans le MLD.

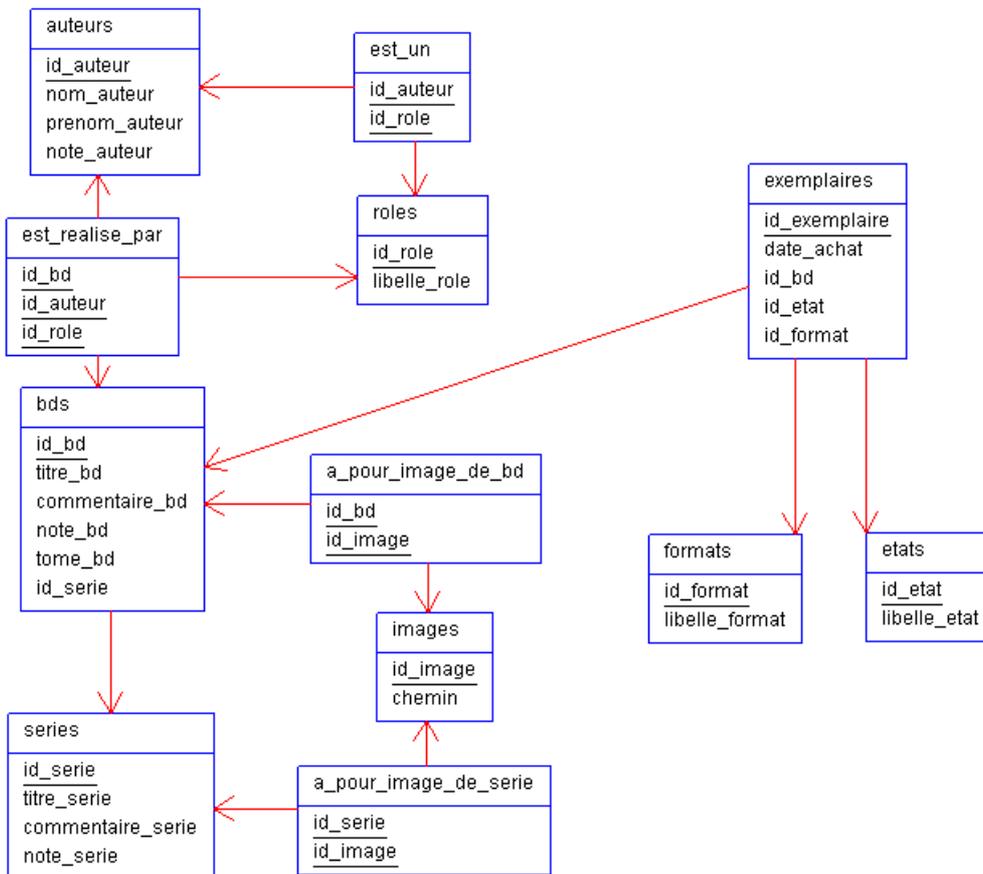
En fonction des cardinalités définies dans le MCD, les associations sont représentées soit par une nouvelle table, soit par une clé étrangère (foreign key) :

➔ Une table est nécessaire pour gérer une association multiple : par exemple les BDs qui peuvent être réalisées par plusieurs auteurs.

➔ Une clé étrangère suffit pour une association simple : par exemple les BDS qui ne peuvent appartenir qu'à une seule série.

La clé étrangère est en fait un simple champ qui contient l'identifiant d'une table : par exemple un champ dans **Bds** qui contient l'identifiant de la série.

Cliquez sur le bouton MLD à gauche pour voir le MLD : Voici le MLD généré par AnalyseSI :



Ce modèle n'est toujours pas spécifique à Access, il est généré sans tenir compte de la base de données ciblée.

Nous découvrons, lors du passage du MCD au MLD, tout l'intérêt de cette analyse.

Ce qui peut paraître évident avec l'expérience apparaît dans le MLD, par exemple :

- pour la série d'appartenance d'une bande dessinée, il suffit d'un champ **id\_serie** dans la table des bandes dessinées;
- par contre, pour les auteurs, une table supplémentaire **est\_realise\_par** est nécessaire pour pouvoir gérer plusieurs auteurs par BD.

 *Pour plus de détails sur le passage de MCD à MLD, consultez ce tutoriel : **Petit guide d'analyse des données à l'aide de la méthode MERISE.***

## IV-E - Le modèle physique de données (MPD)

Après le MCD et MLD, vient le MPD.

Ce modèle définit la structure de la base de données en tenant compte des spécificités de l'application utilisée. Il peut donc y avoir des différences entre le MLD et le MPD.

Par exemple pour la gestion des images :

- pour une version Access antérieure à 2007, il faut gérer une table des images avec leur chemin vers un fichier.
- par contre à partir de la version 2007, on peut utiliser un champ pièce-jointe pour contenir les images; la table **Images** est alors supprimée du MPD.

Nous n'allons pas définir un MPD ici.

Tout d'abord parce que AnalyseSI ne propose pas d'interface pour le définir (hormis le catalogue d'attribut qui contient le type de données des champs).

Et ensuite parce que notre projet de petite taille ne nécessite pas ce travail : le MPD serait très proche du MLD. Nous allons directement construire la base de données dans Access à partir du MLD.

## V - Création de la base de données et d'une première table

Nous y voilà : nous allons créer la base de données Access. Commencez par exécuter Access.

### V-A - Avec Access 2007

Par défaut, la base de données est créée au format **Access 2007** dans **Mes Documents**.

Si vous souhaitez modifier ces valeurs, cliquez sur le bouton **Office** en haut à gauche puis sur le bouton **Options Access** en bas à droite.

Sur l'onglet **Standard**, vous pouvez modifier le format de fichier par défaut, ainsi que le dossier par défaut. J'ai choisi un format de fichier Access 2000 pour ce tutoriel.

 *Vous pouvez convertir ensuite la base de données d'un format vers un autre (pourvu que les fonctionnalités utilisées sont compatibles avec la version de conversion). Si vous utilisez Access 2007, vous pouvez facilement déployer l'application avec le **runtime 2007** et les **extensions développeur** qui sont gratuits.*

Cliquez ensuite sur le bouton **Base de données vide**:

**Nouvelle base de données vide**



Choisissez éventuellement un autre dossier et modifiez le nom du fichier, puis cliquez sur le bouton **Créer**.

Par défaut, Access propose une nouvelle table Table1.

Fermez cette table en cliquant sur la petite croix à droite (pas tout en haut sinon ça ferme l'application, juste la croix de fermeture de la table en dessous).

Pour créer une nouvelle table choisissez l'onglet **Créer** (en haut dans le ruban), puis cliquez sur le bouton **Création de table**.



Ce mode création de table va nous permettre de maîtriser entièrement la création de la table.

Si vous cliquez sur le bouton **Table** à gauche, une table avec un champ N° est créée et les champs sont ajoutés en fonction des valeurs que l'on saisie dans la colonne.

Cet assistant peut paraître facile, mais n'offre aucune maîtrise sur les noms et les types de données de champs ajoutés.

Il faudrait ensuite passer quand même en mode création pour adapter la table créée par l'assistant.

- 💡 *Si vous souhaitez que le message d'avertissement de sécurité ne s'affiche pas à chaque démarrage, approuvez l'emplacement de la base de données :*
- Cliquez sur le bouton **Office** en haut à gauche, puis sur **Options Access**.*
- Choisissez **Centre de gestion de la confidentialité** puis cliquez sur **Paramètre du centre de gestion de la confidentialité...***
- Dans **Emplacements approuvés**, ajoutez le dossier de votre base de données (**Ajoutez un nouvel emplacement...**).*
- Ce paramétrage ne sera effectif que sur le PC sur lequel il a été réalisé.*

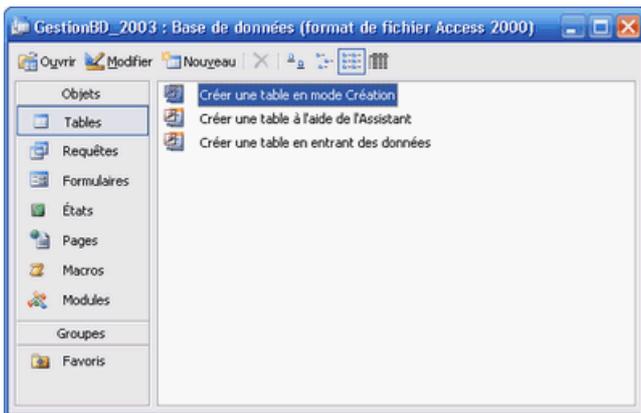
## V-B - Avec Access 2000 à 2003

Choisissez dans le menu : **Fichier => Nouvelle base de données**.

En fonction de la version d'Access, cliquez sur le bouton **Base de données** ou sur l'option créer **Base de données vide** (à droite).

Saisissez le chemin et le nom de la base de données, puis cliquez sur le bouton **Créer**.

La fenêtre de **Base de données** s'affiche :



Notez dans le titre de cette fenêtre le format de la base de données : Access 2000.

Il est possible de changer le format par défaut dans les options :

**Outils => Options... => onglet Avancé => Format de fichier par défaut.**

Il est ensuite possible de convertir la base de données dans un autre format : **Outils => Utilitaires de base de données => Convertir une base de données**.

Pour créer une nouvelle table choisissez **Tables** dans la fenêtre de base de données et double-cliquez sur l'option **Créer une table en mode création**.

Ce mode création de table va nous permettre de maîtriser entièrement la création de la table.

## VI - Création des tables en fonction du MLD

Notons tout d'abord que AnalyseSI génère du code SQL de création des tables.

Ce code n'est pas compatible avec Access, nous n'allons donc pas l'utiliser.

Nous allons créer les tables à la main, mais Access va nous aider à créer facilement et efficacement les liens entre nos tables.

Notre modèle logique de données contient une douzaine de tables.

Par quelle table devons nous commencer?

Il faut commencer par les tables qui sont en fin de chaîne dans les relations (au bout des flèches rouges dans le MLD).

C'est à dire **les tables de référentiel** qui contiennent des données qui apparaîtront notamment dans les listes de choix.

Ces tables de référentiel s'opposent aux **tables de données applicatives** qui contiennent souvent des données issues de ces tables de référentiel.

Ces tables sont généralement moins souvent modifiées que les données applicatives qui les utilisent.

Nous pouvons identifier comme tables de référentiel :

- formats;
- etats;
- images;
- series;
- auteurs;
- rôles.

Et comme tables de données applicatives :

- exemplaires;
- bds.

S'ajoutent à ces tables quelques tables issues des associations pour faire le lien entre les données applicatives et les données de référentiel :

- est\_un;
- est\_realise\_par;
- a\_pour\_image\_de\_bd;
- a\_pour\_image\_de\_serie.

Au sujet des ces *tables d'association*, notez que les relations (les flèches rouges) partent de ces tables. elles sont donc à créer en dernier, une fois que les tables de l'autre côté de la flèche seront créées.

Encore une fois on note l'intérêt de l'analyse réalisée.

En lisant le MLD, nous savons dans quel ordre créer les tables le plus efficacement possible.

Ayant essayé de créer directement les tables, je suis maintenant convaincu de l'intérêt de l'analyse :

J'avais d'abord créé la table des bandes dessinées (erreur...!) pour y greffer ensuite toutes les autres tables.

Après beaucoup de temps perdu à faire et défaire des tables et des champs un peu au hasard, je me suis lancé dans une analyse.

Maintenant que l'analyse est terminée, tout est beaucoup plus clair et on peut commencer sereinement à créer les tables.

## VI-A - Création des tables de référentiel

→ Commençons en douceur avec la table des formats.

A la fin du chapitre précédent, nous avons ouvert une nouvelle table en mode création.

Cette table s'appelle par défaut *Table1*, nous modifierons son nom lors de la sauvegarde.

Le tableau affiché est la liste des champs de la table : cette liste est vide.

Nous devons ajouter les champs vus sur le MLD.

Le champ identifiant : `id_format`

Dans la colonne **Nom du champ**, saisissez `id_format`.

Dans la colonne **Type de données**, choisissez dans la liste la valeur **NuméroAuto**.

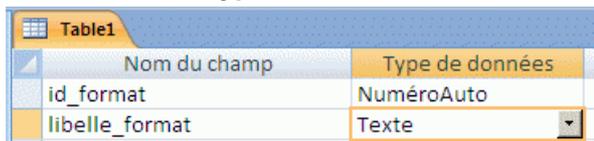
*NuméroAuto* correspond à *uniqueidentifier* que nous avons défini pour l'attribut de l'entité **Formats**.

Ce type de données très utile est un identifiant numérique auto-incrémenté par Access.

Le champ libellé : `libelle_format`

Dans la colonne **Nom du champ**, saisissez `libelle_format`.

Dans la colonne **Type de données**, choisissez dans la liste la valeur **Texte** (c'est la valeur par défaut).



Nom du champ	Type de données
id_format	NuméroAuto
libelle_format	Texte

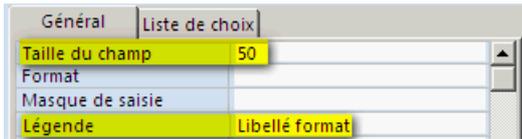
En bas de la table, sous la liste des champs, vous trouverez les propriétés des champs :

Parmi ces propriétés, il y a la **Taille du champ** : 255 caractères par défaut.

Modifiez la taille à 50 caractères, suffisant pour un libellé.

Nous utilisons également la propriété **légende** : celle-ci est utilisée en remplacement du nom de champ lors de l'affichage de la table ou lors de la génération d'un formulaire.

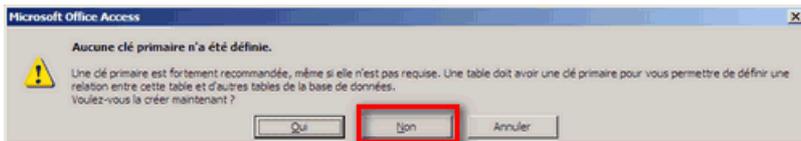
Définissez la légende : *Libellé format*.



Pour sauvegarder la table, cliquez sur la disquette en haut à gauche (ou au clavier CTRL + S). Access nous demande le nom de la table, nous allons utiliser le nom du MLD et le préfixer par **tbl** : **tblFormats**.

 Consultez ce tutoriel sur **les conventions typographiques**.

Nous voici en présence d'un message nous demandant si on souhaite créer une clé primaire : répondez **Non**.



Nous allons définir nous même la clé primaire.

Tout d'abord : qu'est ce qu'une clé primaire?

La clé primaire permet d'indexer les lignes de la table. Pour Access, c'est un peu comme une table des matières permettant de rapidement retrouver où se situe un enregistrement.

Cette clé doit être unique et peut être composée d'un ou plusieurs champs.

Ici encore, le MLD nous aide à choisir les champs qui compose la clé : ce sont les champs soulignés.

Pour la table **formats**, **id\_format** est souligné.

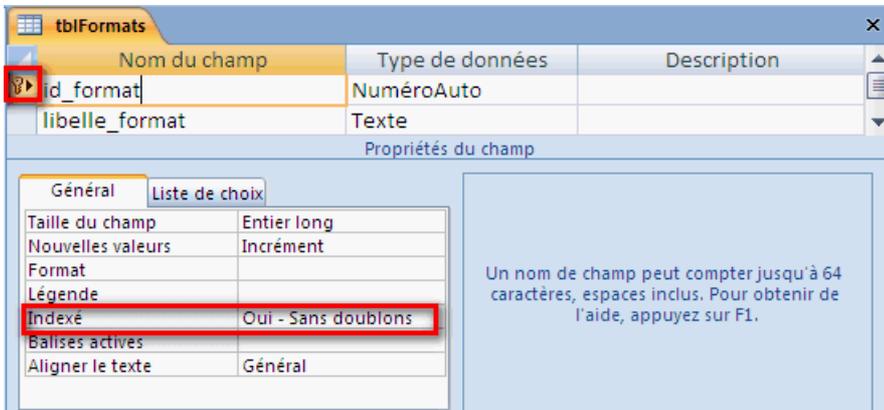


Cliquez sur ce champ puis sur le bouton **Clé primaire** du ruban (dans l'onglet **Création**) ou du menu.

**Clé primaire** est également accessible dans le menu contextuel sur clic du bouton droit de la souris.

Une petite clé apparaît alors à gauche du champ.

De plus ce champ est maintenant indexé sans doublon (car c'est un identifiant unique).



 Définir une clé primaire pour chaque table peut sensiblement améliorer les temps d'affichage des informations, même si aucun filtre et tri ne sont définis.

➔ Vous pouvez maintenant créer la table **etats** nommée **tblEtats** sur le même modèle :

- un champ **id\_etat** de type **NuméroAuto**;
- un champ **libelle\_etat** de type **Texte (50)** avec comme légende *Libellé état*;
- n'oubliez pas de définir **id\_etat** comme clé primaire.

➔ La table **roles** (nommée **tblRoles**) n'est pas très différente :

- un champ **id\_role** de type **NuméroAuto**;
- un champ **libelle\_role** de type **Texte (50)** avec comme légende *Libellé rôle*;
- **id\_role** est la clé primaire.

➔ Créez ensuite la table **auteurs** nommée **tblAuteurs** :

- un champ **id\_auteur** de type **NuméroAuto**;

- un champ **nom\_auteur** de type **Texte (50)** avec comme légende *Nom*;
- un champ **prenom\_auteur** de type **Texte (50)** avec comme légende *Prénom*;
- un champ **note\_auteur** de type **Numérique** avec comme légende *Note de l'auteur*;
- **id\_auteur** est la clé primaire.

Par défaut, la taille d'un champ numérique est **Entier long**.

Pour une note (de 0 à 10), la taille **Octet** est suffisante.

Vous pouvez modifier cette taille en bas dans les propriétés du champ **note\_auteur**.

➔ Au tour de la table **images**, nommée **tblImages** :

- un champ **id\_image** de type **NuméroAuto**;
- un champ **chemin** de type **Texte (255)** avec comme légende *Chemin de l'image*;
- **id\_image** est la clé primaire.

➔ Et on termine par la table **series** que l'on nomme **tblSeries** :

- un champ **id\_serie** de type **NuméroAuto**;
- un champ **titre\_serie** de type **Texte (255)** avec comme légende *Titre de la série*;
- un champ **commentaire\_serie** de type **Mémo** avec comme légende *Commentaire de la série*;
- un champ **note\_serie** de type **Numérique** (taille octet) avec comme légende *Note de la série*;
- **id\_serie** est la clé primaire.

 *Dans cette dernière table, nous avons utilisé un type de données **Mémo**.*

*Les champs **Mémo** sont très utiles puisqu'il permettent de stocker des textes longs, le type de données **Texte** étant limité à 255 caractères.*

*Nous l'utilisons donc pour des commentaires qui peuvent être plus longs que 255 caractères.*

## VI-B - Création des tables applicatives

Le référentiel étant créé, on peut désormais créer les tables applicatives; c'est-à-dire la table des bandes dessinées et des exemplaires.

On remarque sur le MLD un lien (en rouge) de la table des exemplaires vers la table des bandes dessinées : nous allons donc commencer par créer la table cible du lien, celle des bandes dessinées (bds).

### VI-B-1 - Création de la table bds

Créez une nouvelle table et ajoutez les champs comme nous avons maintenant l'habitude :

- un champ **id\_bd** de type **NuméroAuto**;
- un champ **titre\_bd** de type **Texte (255)** avec comme légende *Titre de la BD*;
- un champ **commentaire\_bd** de type **Mémo** avec comme légende *Commentaire de la BD*;
- un champ **note\_bd** de type **Numérique** (taille octet) avec comme légende *Note de la BD*;
- un champ **tome\_bd** de type **Texte (10)** avec comme légende *Tome de la BD*;
- **id\_bd** est la clé primaire.

Sauvegardez la table sous le nom **tblBds**.

### VI-B-2 - Création d'une clé étrangère

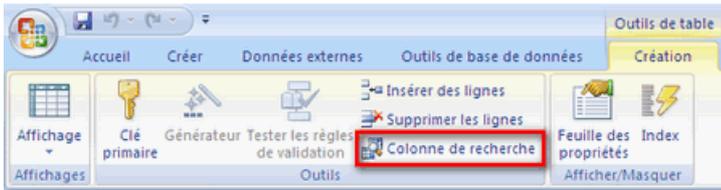
J'ai intentionnellement omis le champ **id\_serie**, tout simplement pour garder le meilleur pour la fin.

Ce champ est une *clé étrangère* : il a été ajouté lors de la génération du MLD (il n'est pas dans le MCD) et représente l'association **Appartient à la série**.

Ce champ **id\_serie** doit donc être créé de manière particulière, afin de lier la table des bandes dessinées à la table des séries.

Pour créer un tel champ, Access met à notre disposition un assistant.

➔ A partir de Access 2007, vous trouvez cet assistant dans l'onglet **Création** du ruban, c'est le bouton **Colonne de recherche**.

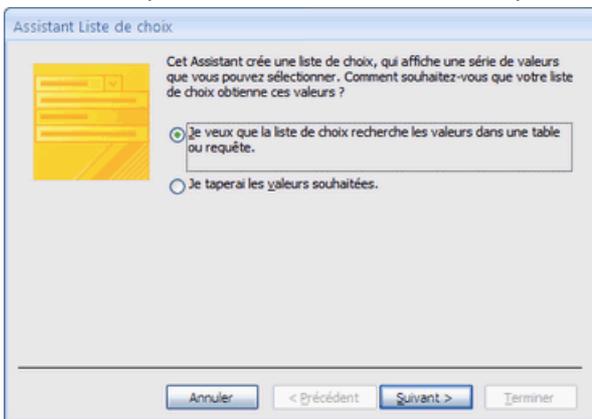


→ Avant Access 2007, vous trouvez cet assistant dans le menu : **Insertion => Champ de recherche ...**

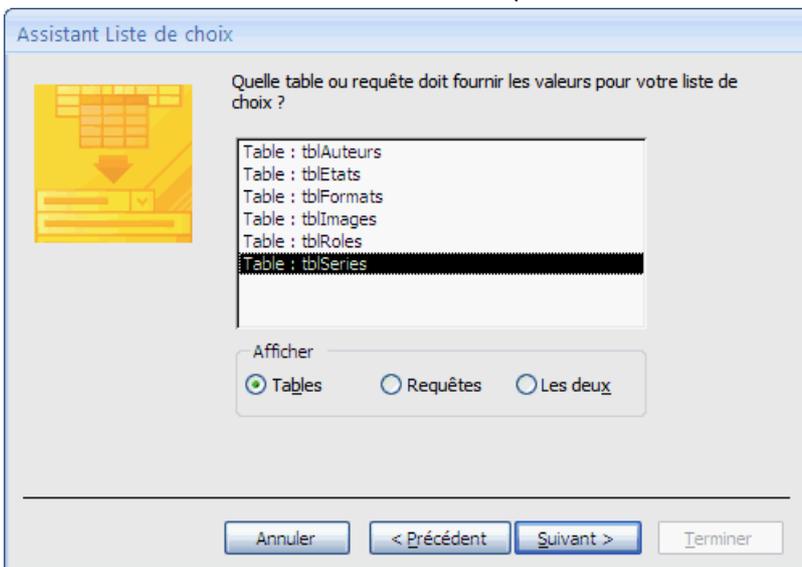


Dans la suite du tutoriel j'appellerai cet assistant comme il est nommé dans sa fenêtre: **assistant liste de choix**  
 Placez-vous alors sur un champ vide (une ligne en-dessous de la fin de la liste de champs) et lancez l'**assistant liste de choix**.

Nous souhaitons définir une liste de choix dans la table **bds** afin de nous aider à choisir la série de la table **series**.  
 Donc sur le premier écran, laissez le choix par défaut (liste à partir des valeurs d'une table) et cliquez sur **Suivant**.



Choisissez ensuite la table **tblSeries** et cliquez sur **Suivant**:



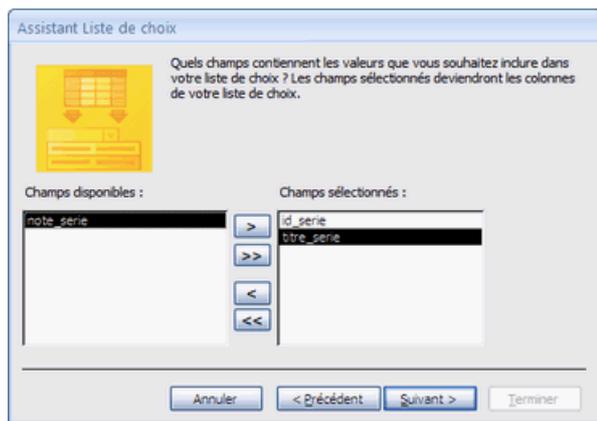
L'assistant nous demande alors quels champs contiennent les valeurs :

Il faut au moins choisir la clé, c'est-à-dire ici **id\_serie**.

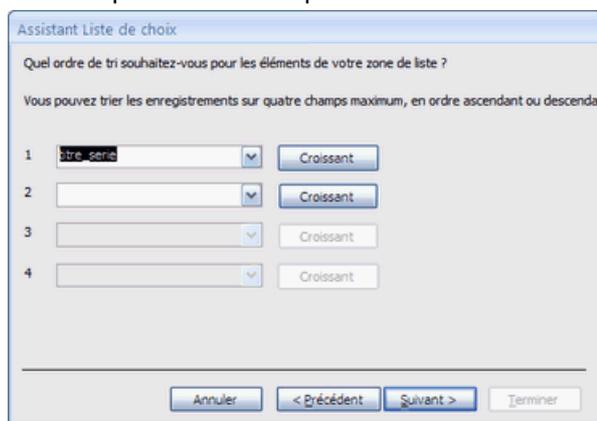
Un champ d'information (libellé, titre, ...) est souvent utile.

Ajoutons donc également le champ **titre\_serie** (double-cliquez sur un champ pour l'ajouter).

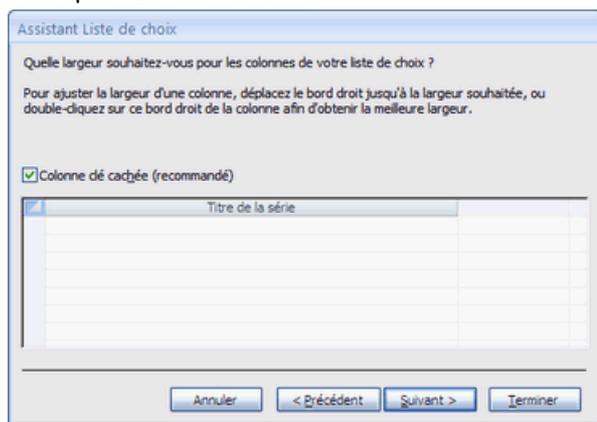
Et cliquez sur **Suivant**.



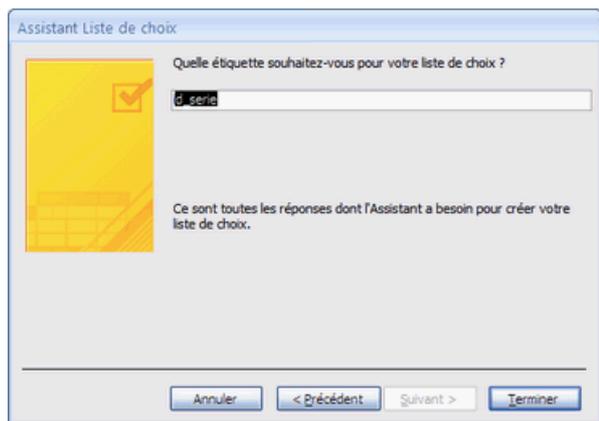
Le tri demandé sur l'écran suivant est celui utilisé pour l'affichage de la liste de choix. Il est pratique d'avoir les séries listées par titre, nous ajoutons donc le champ **titre\_serie** (on laisse un tri croissant, de A à Z).  
Et on clique sur **Suivant** pour continuer.



L'écran suivant permet de définir la largeur des colonnes de la liste de choix. Vous pouvez agrandir la colonne **Titre de la série** par exemple. Laissez la colonne clé cachée, il n'est pas utile de faire apparaître cette colonne à l'affichage. Et cliquez sur **Suivant**.



L'assistant nous demande une étiquette pour la liste de choix. Le terme n'est pas très bien choisi, l'étiquette est en fait le nom de champ qui va être créé. Notre champ s'appelle **id\_serie**. Cliquez sur **Terminer** pour finaliser la création du champ.



Si le programme le demande, validez l'enregistrement de la table en cliquant sur **Oui**

### VI-B-3 - Le travail de l'assistant

Il est toujours intéressant de savoir ce qu'à fait un assistant.

Premier constat : un champ **id\_serie** a été ajouté à notre table **tblBds** :

tblBds	
Nom du champ	Type de données
id_bd	NuméroAuto
titre_bd	Texte
commentaire_bd	Mémo
note_bd	Numérique
tome_bd	Texte
id_serie	Numérique

Ce champ est de type numérique, car il est lié à l'**id\_serie** de la table **series**, lui-même numérique.

Regardons les propriétés de ce champ en cliquant sur l'onglet **Liste de choix** est bas :

Propriétés du champ	
Général	Liste de choix
Afficher le contrôle	Zone de liste déroulante
Origine source	Table/Requête
Contenu	SELECT [tblSeries].[id_serie], [tblSeries].[titre_serie] FROM tblSeries ORDER BY [titre_serie];
Colonne liée	1
Nbre colonnes	2
En-têtes colonnes	Non
Largeurs colonnes	0cm;9,207 cm
Lignes affichées	16
Largeur liste	9,206cm
Limiter à liste	Oui
Autoriser plusieurs valeurs	Non
Autoriser les modifications	Oui
Formulaire Modifier les éléments	
Afficher uniquement les éléments	Non

C'est ici que l'assistant a effectué la plus grosse partie de son travail.

Il a défini une zone de liste déroulante pour la saisie.

L'origine de cette liste est **Table/Requête**.

Le contenu est en fait une requête écrite en SQL par l'assistant :

Contenu de la liste

```
SELECT [tblSeries].[id_serie], [tblSeries].[titre_serie] FROM tblSeries ORDER BY [titre_serie];
```

Nous retrouvons dans cette requête les éléments définis dans l'assistant :

- les champs **id\_serie** et **titre\_serie** de la table **tblSeries**;
- le tri sur le champ **titre\_serie**.

La colonne liée est la première colonne, c'est-à-dire le champ **id\_serie**.

En effet cet identifiant sera stocké dans la table **bds**.

Il faut toujours stocker l'identifiant unique, le titre sera recherché grâce à ce dernier. Stocker l'identifiant prend moins d'espace que de stocker le titre. De plus si le titre est modifié, il n'y a pas de modification à apporter à la table **tblBds**. Le nombre de colonne est défini à 2, car on a demandé 2 champs. Les largeurs de colonnes sont très importantes : elles sont définies à **0cm;9,207cm**. Cela correspond également à ce que nous avons défini dans l'assistant :

- la colonne clé (première colonne) est masquée, donc une largeur de 0cm;
- la deuxième colonne est le titre de la série, elle a été définie à 9,207cm et peut être modifiée ici si besoin.

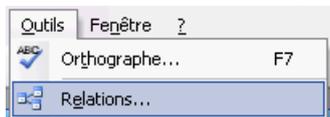
Notez que vous pouvez également définir le nombre de lignes affichées dans la liste de choix : ici 16 par défaut. La largeur de la liste peut-être plus petite que la somme des largeurs des colonnes; une barre défilement horizontale apparaîtra alors sur la liste.

Une dernière propriété intéressante : limiter à liste qui est égale à oui par défaut. Ainsi on ne pourra pas saisir une série qui n'est pas définie dans la table **tblSeries**. Le travail de l'assistant ne s'arrête pas là; il a également défini une relation entre les deux tables **tblBds** et **tblSeries**. Pour voir les relations :

- à partir de Access 2007, cliquez sur le bouton **Relations** de l'onglet **Outils de base de données**;



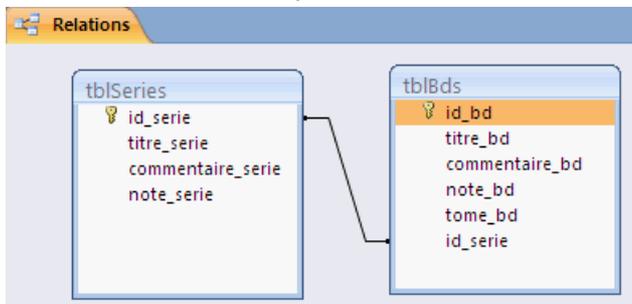
- avant Access 2007, cliquez dans le menu : **Outils => Relations**.



Si besoin, cliquez avec le bouton droit et choisissez **Afficher toutes les relations** afin de les voir.



Voici la relation mise en place :



## VI-B-4 - Création de la table exemplaires

Créez une nouvelle table. Ajoutez-y :

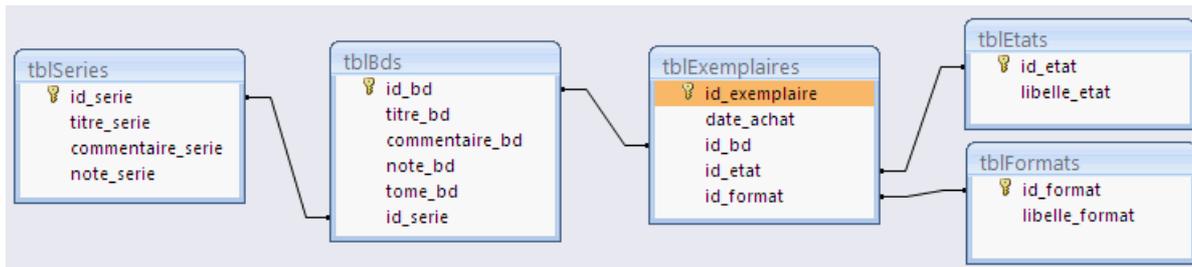
- un champ **id\_exemplaire** de type **NuméroAuto**;
- un champ **date\_achat** de type **Date/Heure**;
- définissez **id\_exemplaire** comme clé primaire.

Sauvegardez la table sous le nom **tblExemplaires** :

tblExemplaires	
Nom du champ	Type de données
 id_exemplaire	NuméroAuto
date_achat	Date/Heure

Utilisez ensuite l'assistant liste de choix pour créer les champs **id\_bd**, **id\_etat** et **id\_format**. Choisissez tour à tour les tables **tblBds**, **tblEtats** puis **tblFormats**.

Comme nous venons de le faire pour l'**id\_serie**, ajoutez la clé primaire (cachée) et un champ titre ou libellé. Voici l'écran des relations après ajout de ces champs :



## VI-C - Création des tables issues des associations

Maintenant que nos tables principales sont créées, nous pouvons créer les tables d'association. C'est-à-dire les tables :

- est\_un;
- est\_realise\_par;
- a\_pour\_image\_de\_bd;
- a\_pour\_image\_de\_serie;

Commençons par les associations d'images en créant la table **a\_pour\_image\_de\_bd** nommée **tblAPourImageDeBd**.

Créez une nouvelle table.

Cette table contient uniquement deux champs de lien avec les tables **tblBds** et **tblImages**.

Exécutez l'assistant liste de choix pour créer ces deux champs :

Choisissez d'abord la table **bds** pour créer le champ **id\_bd**.

Lorsque Access demande à sauvegarder la table, nommez-la **tblAPourImageDeBd**.

Choisissez de ne pas créer automatiquement de clé primaire.

Créez ensuite le champ **id\_image** en choisissant la table **tblImages** dans l'assistant liste de choix.

Nous avons refusé de créer automatiquement la clé primaire, mais nous allons manuellement en définir une.

Dans le MLD, les champs **id\_bd** et **id\_image** de la table **a\_pour\_image\_de\_b** sont soulignés.

Il composent à eux deux la clé primaire.

Sélectionnez les deux champs (en cliquant sur le sélecteur à gauche du champ et la touche CTRL) puis cliquez sur le bouton **Clé primaire**.

Une petite clé est alors affichée devant chacun des deux champs.

tblAPourImageDeBd	
Nom du champ	Type de données
id_bd	Numérique
id_image	Numérique

Procédez de la même manière pour créer la table **a\_pour\_image\_de\_serie** nommée **tblAPourImageDeSerie**.

tblAPourImageDeSerie	
Nom du champ	Type de données
id_serie	Numérique
id_image	Numérique

Créez ensuite la table **est\_un** (nommée **tblEstUn**) en créant avec l'assistant les champs **id\_auteur** lié à la table **tblAuteurs** et **id\_role** lié à la table **tblRoles**.

Ces deux champs composent la clé primaire à définir.

Pour la liste de choix des auteurs, vous pouvez ajouter le nom et le prénom en plus de l'identifiant.

La liste affichera le nom et le prénom de l'auteur.

tblEstUn	
Nom du champ	Type de données
id_auteur	Numérique
id_role	Numérique

Il reste enfin à créer la table **est\_realise\_par**, nommée **tblEstRealisePar**.

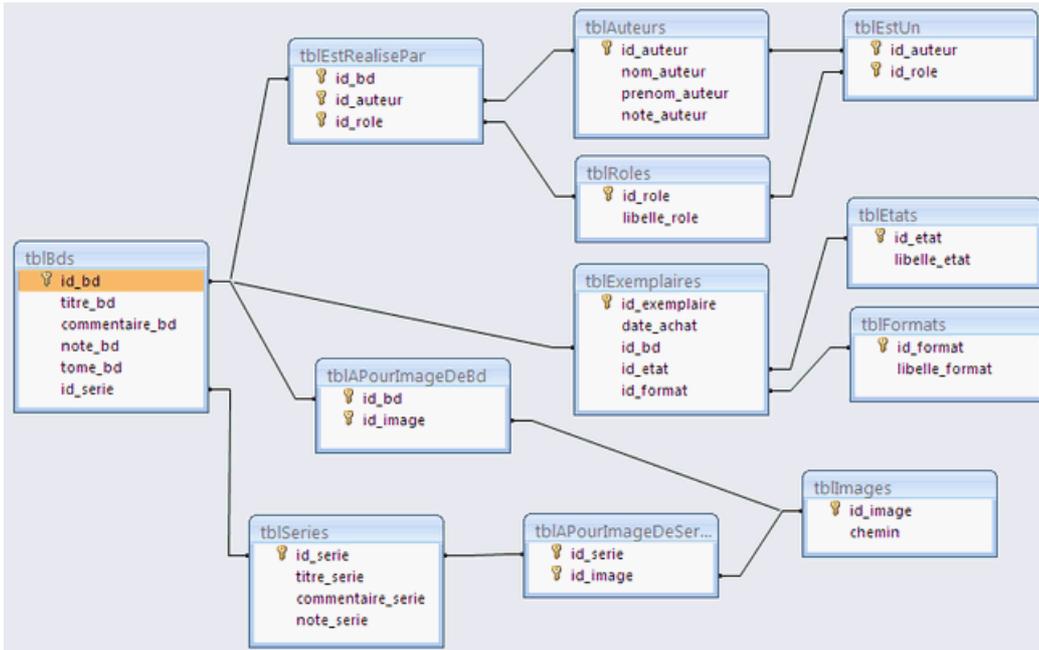
Les trois champs de cette table sont également créés avec l'assistant liste de choix.

tblEstRealisePar	
Nom du champ	Type de données
id_bd	Numérique
id_auteur	Numérique
id_role	Numérique

Toutes les tables sont maintenant créées.

## VI-D - Vérification visuelle dans Access

Affichez toutes les relations dans la fenêtre des relations.  
Si besoin, repositionnez les tables de manière plus lisible :



L'affichage des tables dans l'écran des relations Access est proche du MCD.  
Nous pouvons vérifier ici visuellement que nous n'avons rien oublié.

## VI-E - Téléchargement de la base de données

Vous pouvez télécharger la base de données contenant ces tables :  
**Les tables au format Access 2000 (HTTP)**

## VII - Ajout de contraintes

 Pour plus d'informations sur les contraintes, consultez ce tutoriel : **Contraintes avancées sous Microsoft Access.**

Les contraintes portent bien leur nom : elles vont contraindre l'utilisateur à saisir des valeurs qui répondent à des règles de validation.

Pour des règles simples, il est possible les définir directement dans le champ.

D'autres règles plus complexes nécessiteront plus tard du code en VBA.

→ **Un exemple simple : les notations :**

Nous avons en effet défini plusieurs champs de notation : **note\_bd**, **note\_serie** et **note\_auteur**.

Ces notes n'ont aucune autre contrainte que d'être numérique (le type de données du champ).

Nous souhaitons limiter la saisie à une note entre 0 et 10.

Ouvrez la table **tblAuteurs** en mode création.

A partir de Access 2007, les tables sont listées dans le volet de navigation à gauche.

Avant Access 2007, les tables sont listées dans la fenêtre de base de données.

Pour ouvrir une table en mode création, cliquez sur la table avec le bouton droit et choisissez **Mode création** (le bouton avec l'équerre).

Sélectionnez le champ **note\_auteur** et modifier les paramètres en bas :

- **Valide Si** :  $\geq 0$  Et  $\leq 10$  - **Message si erreur** : La note doit être comprise entre 0 et 10

Lors de la saisie de la note, le message inscrit dans *Message si erreur* s'affichera si la note ne respecte pas la règle *Valide si*.

On peut faire encore mieux en proposant une liste à l'utilisateur :

- Dans les paramètres du champ, cliquez sur l'onglet **Liste de choix**.

- Modifiez la propriété **Afficher le contrôle**, choisissez la valeur **Zone de liste déroulante**.

- Comme **Origine Source**, sélectionnez **Liste valeurs**.

- Dans **Contenu**, saisissez les valeurs possibles séparées par des points-virgules : **0;1;2;3;4;5;6;7;8;9;10**.

- Vous pouvez même limiter la saisie à la liste définie avec la propriété **Limiter à liste** (la contrainte *Valide si* devient alors inutile).

Petite remarque : une telle liste de choix peut être créée également à l'aide de l'assistant liste de choix, en choisissant l'option *Je taperai les valeurs souhaitées*.

Vous pouvez alors apporter les mêmes modifications aux champs **note\_bd** et **note\_serie**.

→ **Autre contrainte possible : imposer la saisie d'un champ** :

Par exemple pour la table des auteurs, on peut imposer la saisie du nom.

Ouvrez la table **tblAuteurs** en mode création et sélectionnez le champ **nom\_auteur**.

En bas, dans les propriétés du champ, définissez **Null interdit** égale à **Oui**.

Définissez également **Chaîne vide autorisée** égale à **Non**.

Il est important pour un champ texte de définir ces deux propriétés :

- **Null interdit** à Oui oblige l'utilisateur à saisir une données;

- **Chaîne vide autorisée** à Non interdit l'utilisateur de saisir une chaîne vide, c'est-à-dire de mettre un texte puis de le supprimer.

Par contre on laisse la possibilité de ne pas saisir de prénom, au cas où l'auteur ne soit connu que son nom ou pseudonyme.

On impose de la même manière la saisie d'un titre de bande dessinée dans le champ **titre\_bd** de la table **tblBds**.

Puis on impose la saisie d'un titre de série dans la table **tblSeries**.

→ **Autre contrainte, un exemplaire doit forcément être rattaché à une bande dessinée du catalogue** :

Imposons donc la saisie du champ **id\_bd** dans la table **tblExemplaires**.

Ce champ étant numérique, il n'y a pas de propriété *Chaîne vide autorisée*.

Définissez uniquement la propriété *Null interdit* = **Oui**.

→ **Ajoutons maintenant une contrainte d'unicité du couple Série/Tome** :

En effet, dans la table **tblBds**, on définit la série d'appartenance (**id\_serie**) et le tome (**tome\_bd**).

Il ne doit pas être possible de définir un même tome deux fois pour la même série.

Dans la table **tblBds** en mode création, sélectionnez les champs **id\_serie** et **tome\_bd**.

Cliquez ensuite dans l'onglet **Création** sur le bouton **Index** :



Avant Access 2007, vous trouverez ce bouton index dans la barre d'outils en haut, ou dans le menu : **Affichage** => **Index**.

Dans la fenêtre des index, vous trouvez la clé primaire nommée **PrimaryKey** composée du champ **id\_bd**.

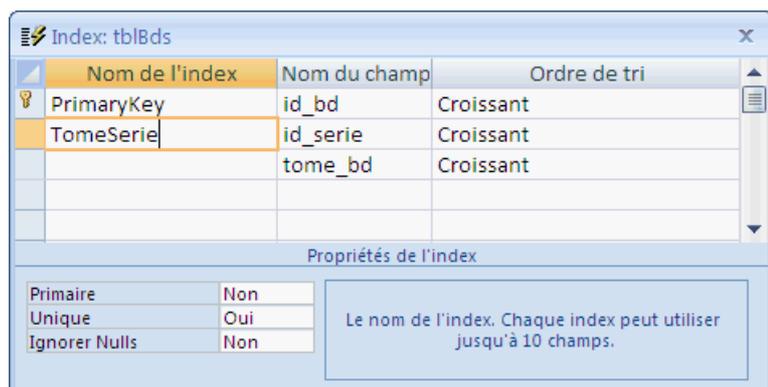
En effet la clé primaire est un index.

Ajoutez deux lignes pour créer notre index :

- Nom de l'index : TomeSerie, à mettre uniquement sur la première ligne;

- Pour la première ligne, modifiez en bas : *Unique* = **Oui** afin de ne pas accepter de doublons.

- Nom du champ sur chacune des lignes : **id\_serie** puis **tome\_bd**;



Fermer et enregistrez la table.

La création de cet index a deux effets :

D'abord il sera interdit de saisir des doublons de Série+Tome.

Ensuite Access gère un index, c'est à dire une table des matières qui va accélérer la recherche des BDs par Série et Tome.

L'insertion et la modification de données sera un peu plus lente car il faudra qu'Access gère à la fois la table et l'index. Par contre la lecture sera plus rapide car l'index agit comme une table des matières et permet de trouver rapidement un enregistrement.

## VIII - Ajout de données de tests

Avant d'aller plus loin dans la conception de notre base de données, nous allons ajouter quelques données de test. En renseignant ces données, nous allons pouvoir mettre en oeuvre les listes de choix et les contraintes que nous avons définies.

Si vous n'avez pas de bandes dessinées sous la main, le site **BD Gest'** regorge d'informations.

 *Si vous ne souhaitez pas remplir vous-même ces données de tests, vous trouverez à la fin de ce chapitre la base de données pré-remplie.*

Voici les quelques séries que j'ai choisies pour les données de tests :

- **Horologiom**;
- **Finkel**;
- **La Nef des fous**;
- **Les tuniques bleues**;

En toute logique, nous remplissons d'abord les tables de référentiels.

Ouvrez la table **tblRoles** : un double-click suffit à ouvrir la table en saisie.

Saisissez la colonne **Libellé rôle** les différents rôles possibles :

- 01- Scénariste;
- 02- Dessinateur;
- 03 -Coloriste;

 *Petite astuce : vous pouvez facilement ordonner les valeurs qui apparaîtront dans la liste en ajoutant un numéro au début.*

*Pour le tri, il est nécessaire de mettre la numérotation sur deux chiffres si le nombre de valeurs peut atteindre au moins 10.*

*Bien entendu, il faut avoir défini un tri (ici sur le libellé du rôle) lors de la création des listes de choix avec l'assistant.*

*On aurait également pu ajouter un champ numérique dédié pour le tri et le masquer à l'affichage.*

tblRoles	
id_role	Libellé rôle
1	01- Scénariste
2	02- Dessinateur
3	03 -Coloriste

Le champ **id\_role** étant un numéro automatique, Access incrémente lui-même ce champ.

Si Access vous demande si vous souhaitez enregistrer les modifications apportées à la mise en forme de la table, vous pouvez répondre Oui.

Ces modifications sont par exemple une modification de la taille des colonnes.

Les modification effectuée sur les données sont enregistrées automatiquement.

Ajoutez de la même manière les différents formats dans la table **tblFormats** :

tblFormats	
id_format	Libellé format
1	01 - Petit format
2	02 - Grand format
3	03 - Tirage de tête

Ajoutons ensuite quelques valeurs dans la table **tblEtats** :

tblEtats	
id_etat	Libellé état
1	01 - Etat neuf
2	02 - Bon état
3	03 - Etat moyen
4	04 - Mauvais état

Puis ajoutez les auteurs des bandes dessinées à la table **tblAuteurs**.

Commençons par la série Horologiom.

Le scénariste et le dessinateur sont une seule et même personne : *Lebeault Fabrice*.

Il suffit donc de renseigner son nom et prénom dans la table.

Le nom est imposé par une contrainte, ce que nous pouvons tester en essayant de saisir un enregistrement sans le nom.

Si vous souhaitez lui attribuer une note, vous pouvez la saisir dans le champ **Note de l'auteur**.

On en profite pour tester la liste de choix de la note, et la contrainte d'avoir une note entre 0 et 10.

La mise en couleur est assurée par *Breton Florence*, ajoutons-la également dans la table **tblAuteurs**.

Remarque : si on n'a pas d'avis sur la colorisation par exemple, il n'est pas nécessaire de noter l'auteur.

D'ailleurs, le champ **note\_auteur** a sa propriété **Null interdit** égale à Non.

Il est donc possible de laisser le champ égal à **Null**, c'est à dire sans aucune valeur.

Reste à ajouter les auteurs des autres séries :

tblAuteurs				
id_auteur	Nom	Prénom	Note de l'auteur	
1	Lebeault	Fabrice		8
2	Breton	Florence		
3	Convard	Didier		
4	Gine			
5	Rita			
6	Cauvin	Raoul		7
7	Salvérius	Louis		
8	Lambil	Willy		6
9	Turf			9

J'ai mis quelques notes pour les tests, un peu au hasard; ne le prenez pas mal si j'ai sous-noté votre auteur préféré. ;o).

Tous ces auteurs ont été ajoutés sans distinction dans la table auteurs.

Il faut maintenant préciser le ou les rôles de chacun dans la table d'association **tblEstUn**.

Ouvrez cette table **tblEstUn** (double-clique).

Notez comme la saisie est facile avec les listes de choix :

tblEstUn	
id_auteur	id_role
*	
Breton	Florence
Cauvin	Raoul
Convard	Didier
Gine	
Lambil	Willy
Lebeault	Fabrice
Rita	
Salvérius	Louis
Turf	

Et voici la table qui définit le rôle de nos auteurs:

tblEstUn	
id_auteur	id_role
Lebeault	01- Scénariste
Lebeault	02- Dessinateur
Breton	03 -Coloriste
Convard	01- Scénariste
Gine	02- Dessinateur
Rita	03 -Coloriste
Cauvin	01- Scénariste
Salvérius	02- Dessinateur
Lambil	02- Dessinateur
Turf	01- Scénariste
Turf	02- Dessinateur
Turf	03 -Coloriste

Deux remarques :

- Certains auteurs peuvent cumuler plusieurs rôles;
- Access affiche le premier champ visible de la liste de choix, mais c'est bien l'identifiant qui est stocké!

Nous sommes prêts à saisir les données de séries, dans la table **tblSeries** bien évidemment :

tblSeries				
	id_serie	Titre de la série	Commentaire de la série	Note de la série
+	1	Horologiom	Pour les amateurs de rouage.	7
+	2	Finkel	Pour les marins. Pour les amoureux du voyages et des poissons.	6
+	3	Tuniques bleues (Les)	Incontournable!	8
+	4	Nef des fous (La)		8

Il suffit de saisir le titre de la série, un commentaire si nécessaire, et une note si vous le souhaitez.

Deux remarques cette fois aussi :

- Appuyez sur les touches CTRL + ENTREE pour passer à la ligne dans le commentaire;
- Rejetez les articles en fin de titre (Le, La , Les, ...) pour des tris plus pertinents.

Il maintenant grand temps de saisir le catalogue de bandes dessinées : la table **tblIBDs**.

Il suffit de saisir :

- Le titre de la bande dessinée;
- Eventuellement un commentaire et une note;
- Le numéro de tome;
- La série d'appartenance.

Vous pouvez tester l'index sans doublon positionné sur le couple Série+Tome.

Il n'est pas possible de saisir deux fois le même tome pour une même série.

Pour saisir la série, utilisez efficacement la liste de choix. Saisissez la première lettre de la série et Access proposera la série correspondante.

Notez qu'on a saisi des numéros de tomes non numériques, pour les hors-série et les intégrales.

Là encore j'ai mis des notes au hasard qui ne reflète en rien la qualité des albums.

J'ai également ajouté une bande dessinée importantes pour nos tests, sans numéro de tome et sans série.

Cela nous permettra de tester les différentes jointures lorsque nous aborderons les requêtes.

id_bd	Titre de la BD	Commentaire de la BD	Note de la BD	Tome de la BD	id_serie
1	L'homme sans clef			7 1	Horologiom
2	L'instant du Damokle			7 2	Horologiom
3	Nahédig			6 3	Horologiom
4	La nuit du requinqueur			8 4	Horologiom
5	Le grand rouage			6 5	Horologiom
6	L'enfant de mer			7 1	Finkel
7	Océane			4 2	Finkel
8	Génos			5 3	Finkel
9	Le secret			7 4	Finkel
10	Origine			7 5	Finkel
11	Esta			8 6	Finkel
12	Corruption			5 7	Finkel
13	Premier Cycle	Intégrale.		INT1	Finkel
14	Eauxfolles			9 1	Nef des fous (La)
15	Pluvior 627			6 2	Nef des fous (La)
16	Turbulences			8 3	Nef des fous (La)
17	Au turf			4	Nef des fous (La)
18	Puzzle			6 5	Nef des fous (La)
19	Les Chemins énigmatiques			6	Nef des fous (La)
20	Terminus			9 7	Nef des fous (La)
21	Puzzle Gribouillis	Hors série		COF2	Nef des fous (La)
22	La genèse	Hors série		HS01	Nef des fous (La)
23	Histoire d'Eauxfolles - Cour	Hors série		HS02	Nef des fous (La)
24	Le petit Roy	Hors série		HS03	Nef des fous (La)
25	Le grenier d'Eauxfolles	Hors série		HS04	Nef des fous (La)
26	La chambre du Roy - Le bric	Hors série		PF	Nef des fous (La)
86	Un chariot dans l'Ouest			1	Tuniques bleues (Les)
87	Du nord au sud			2	Tuniques bleues (Les)

La BD sans tome ni série :

id_bd	Titre de la BD	Commentaire de la BD	Note de la BD	Tome de la BD	id_serie
143	Adios Palomita	Western		9	

Une remarque importante sur les numéros automatiques :

Vous noterez que la numérotation passe directement de 26 à 86.

En effet lorsque vous éditez une ligne, le numéro est incrémenté.

Si vous annuler la saisie (touche Echap par exemple) ou que vous supprimez la ligne, la numérotation reste incrémentée.

Il est donc possible qu'il y ait des "trous" dans ce numéro.

Cela ne gêne en rien le fonctionnement de la base de données : ce numéro est à usage interne et ne sera pas visible de l'utilisateur.

Pour information, le type de données *Entier long* a comme valeur maximale : 2 147 483 647.

Il est très peu probable d'atteindre ce maximum.

Le catalogue de bandes dessinées étant saisi, il faut maintenant définir les auteurs de ces bandes dessinées.

Cela est fait dans la table **tbiEstRealisePar** :

tblEstRealisePar			
	id_bd	id_auteur	id_role
	L'homme sans clef	Lebeault	01- Scénariste
	L'homme sans clef	Lebeault	02- Dessinateur
	L'homme sans clef	Breton	03 -Coloriste
	L'instant du Damokle	Lebeault	01- Scénariste
	L'instant du Damokle	Lebeault	03 -Coloriste
	L'instant du Damokle	Breton	03 -Coloriste
	Nahédig	Lebeault	01- Scénariste
	Nahédig	Lebeault	02- Dessinateur
	Nahédig	Breton	03 -Coloriste
	La nuit du requinqueur	Lebeault	01- Scénariste
	La nuit du requinqueur	Lebeault	02- Dessinateur
	La nuit du requinqueur	Breton	03 -Coloriste
	Le grand rouage	Lebeault	01- Scénariste
	Le grand rouage	Lebeault	02- Dessinateur
	Le grand rouage	Breton	03 -Coloriste

Il ne sera pas nécessaire de gérer les auteurs dans l'application, nous le faisons pour avoir quelques données de tests. Je n'ai d'ailleurs pas eu le courage de définir tous les coloristes.

Pour la bande dessinées *Adios Palomita*, je n'ai même pas défini d'auteurs.

Une dernière table à remplir : la table des exemplaires **tblExemplaires** :

Je vais remplir cette table avec les exemplaires que je possède :

tblExemplaires					
	id_exemplaire	dateachat	id_bd	id_etat	id_format
	1		L'homme sans clef	02 - Bon état	02 - Grand format
	2		L'instant du Damokle	02 - Bon état	02 - Grand format
	3		Nahédig	01 - Etat neuf	02 - Grand format
	4		La nuit du requinqueur	02 - Bon état	02 - Grand format
	5	01/06/2008	Le grand rouage	01 - Etat neuf	02 - Grand format
	6		L'enfant de mer	01 - Etat neuf	
	7		Océane	02 - Bon état	
	8		Génos	02 - Bon état	
	9		Le secret	03 - Etat moyen	
	10		Origine	04 - Mauvais état	
	11		Esta	03 - Etat moyen	
	12		Corruption	02 - Bon état	
	13	01/02/1994	Eauxfolles	02 - Bon état	
	14	01/03/1995	Pluvior 627	03 - Etat moyen	
	15	01/08/1997	Turbulences	02 - Bon état	
	16	01/09/2002	Au turf	02 - Bon état	
	17	01/06/2008	Puzzle	01 - Etat neuf	
	18	01/02/2010	Les Chemins énigmatiques	02 - Bon état	
	19	01/02/2010	Terminus	01 - Etat neuf	
	20		La prison de Robertsonville	03 - Etat moyen	
	21		Blue Retro	04 - Mauvais état	
	22		Des bleus et des bosses	02 - Bon état	
	23		La rose de Bantry	01 - Etat neuf	
	24	01/04/2008	Drummer Boy	02 - Bon état	
	25		Les bleus en folie	02 - Bon état	

Les dates d'achat, formats et états sont optionnels.

Remarques pour les images :

La gestion des images n'étant pas aisée, nous laissons vides ces tables pour le moment.

Nous traiterons cette gestion des images dans un chapitre à part.

Les données de tests sont maintenant renseignées.

Cela a été un peu laborieux, nous allons bien sûr développer une interface qui permettra à l'utilisateur de remplir bien plus facilement ses données.

Vous pouvez télécharger la base de données contenant les tables avec leurs contraintes et les données de tests :

## Les tables et données au format Access 2000 (HTTP)

### IX - Création d'une requête

 Pour les débutants, consultez ce tutoriel : **Créer des requêtes simples.**

 Pour plus d'informations sur les requêtes et les jointures, consultez ces tutoriels :

- **Comprendre les jointures dans Access.**
- **Access - Les Bases : V-3. Les requêtes multi-tables.**

Jusqu'à présent, nous avons uniquement afficher des tables.

Pour aller plus loin, il est utile de créer des requêtes.

Plusieurs types de requêtes sont disponibles : on peut utiliser les requêtes pour ajouter des lignes à une table, modifier des données ou simplement les afficher.

Un des avantages des requêtes est qu'elles nous permettent d'afficher des données issues de plusieurs tables, le lien entre ces tables étant réalisé par des jointures.

Elles permettent également de définir des tris et des filtres.

Nous allons dans un premier temps, et pour nous familiariser avec les requêtes, créer une requête de sélection pour afficher les bandes dessinées correctement triées.

En effet lorsqu'on ouvre la table, les bandes dessinées sont présentées dans l'ordre où elles ont été saisies.

Il est même possible parfois que les lignes ne soient pas dans cet ordre.

Il est donc important de définir un tri d'affichage pour garantir l'ordre des lignes.

On pourrait définir un tri au niveau d'un futur formulaire mais nous allons plutôt le définir dans une requête.

La requête pourra ainsi être réutilisée à plusieurs endroits : dans un formulaire, dans un état, ou même dans une autre requête.

Que souhaitons-nous?

Tout simplement une requête qui affiche la liste des bandes dessinées **triées par série et par numéro de tome.**

Pour créer une requête à partir d'Access 2007 : dans l'onglet **Créer**, cliquez sur le bouton **Création de requête.**

Le bouton **Assistant requête** permet de lancer un assistant pour vous faciliter le travail.

Pour créer une requête avant Access 2007 : dans la fenêtre de base de données, affichez les requêtes et cliquez sur **Créer une requête en mode création.**

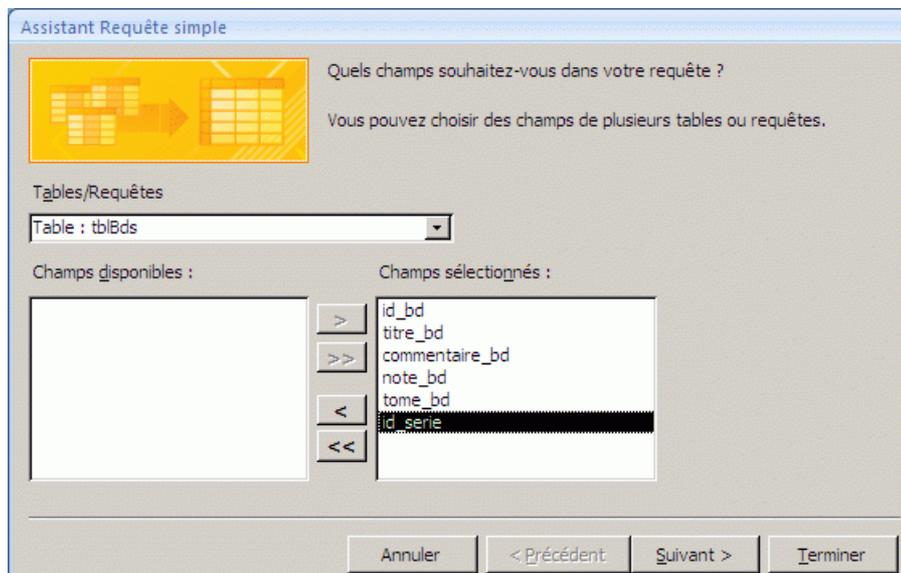
**Créer une requête à l'aide de l'assistant** permet de lancer un assistant pour vous faciliter le travail.

Lancez l'assistant de création de requête.

Dans la fenêtre qui s'affiche, choisissez **Assistant Requête simple** et cliquez sur OK.

Dans la liste, sélectionnez la table **tblBds**.

Dans un premier temps, sélectionnez tous les champs, il sera ensuite possible de les supprimer s'ils ne sont pas nécessaires.



Remarque : à ce stade de l'assistant, on pourrait ajouter des champs d'une autre table en modifiant la table sélectionnée dans la liste.

Cliquez sur **Suivant** pour continuer.

Laissez la sélection à **Détaillée**, l'autre option étant utile pour faire des regroupements : compte, somme, ...

Cliquez sur **Suivant** pour poursuivre.

Donnez un nom à la requête, par exemple : **qryBds**.

Préfixer les noms de requête avec **qry** par exemple est utile car parfois les tables et requêtes apparaissent dans une même liste.

On peut ainsi facilement les différencier.

Cliquez sur **Terminer** pour continuer.

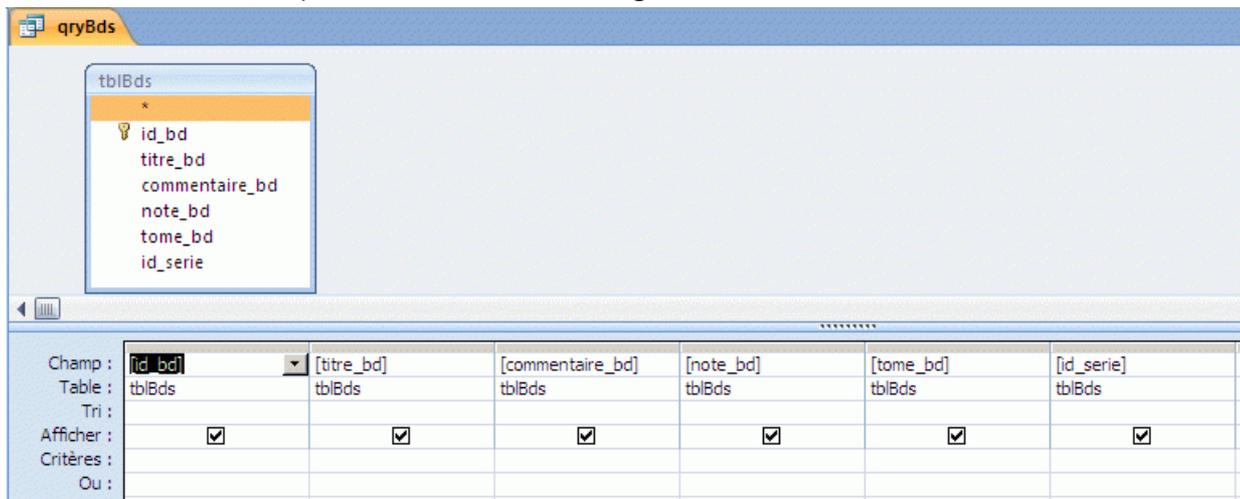
La requête créée s'ouvre alors en affichage.

On ne voit pas de différence par rapport à l'affichage de la table car on s'est contenté d'ajouter les champs de la table.

Passer en mode création pour visualiser la structure de la requête :

- A partir d'Access 2007, dans l'onglet **Accueil** choisissez **Mode Création**;

- Avant Access 2007, cliquez dans le menu : **Affichage => Mode création**.



Cet affichage est appelé **QBE** (Query By Exemple).

Dans la partie haute, les tables sont affichées : ici on n'en a qu'une seule.

Dans la partie basse, on trouve en colonne les champs de la requête ainsi que la table d'où ils sont extraits.

La ligne de tri est vide, ainsi que les lignes de critères.

On remarque également que tous les champs sont cochés pour affichage.

Comme nous souhaitons trier par série et par tome, ajoutez un tri à chacun de ces champs.

Choisissez **Croissant** comme tri des champs **tome\_bd** et **id\_serie**.

Les tris s'effectuent dans l'ordre d'affichage des colonnes, c'est-à-dire par tome puis par série.

Cliquez sur la colonne **tome\_bd** tout en haut (le curseur se change alors en flèche vers le bas) pour la sélectionner, puis faites-là glisser après la colonne **id\_serie**.

ote_bd]	[id_serie]	[tome_bd]
tblBds	tblBds	tblBds
	Croissant	Croissant
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

On demande alors un tri d'abord par série puis par tome.

Notez qu'il aurait aussi été possible de laisser les colonnes dans l'ordre tome puis série, et de rajouter une colonne tome à la fin pour le tri.

On ne demanderait alors pas d'affichage pour cette colonne de tri.

te_bd	tome_bd	id_serie	tome_bd
tblBds	tblBds	tblBds	tblBds
		Croissant	Croissant
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pour notre cas, on peut faire passer la colonne tome à droite de la colonne série, cela a du sens de toute manière d'avoir d'abord la série puis le tome.

Il sera ensuite possible lors de la création des formulaires et états d'ordonner les colonnes comme on le souhaite indépendamment de l'ordre mis dans la requête.

Repasser en affichage *Feuille de données* pour voir le résultat :

id_bd	Titre de la BD	Commentaire de la BD	Note de la BD	id_serie	Tome de la BD
143	Adios Palomita	Western	9		
1	L'homme sans clef		7	Horologiom	1
2	L'instant du Damokle		7	Horologiom	2
3	Nahédig		6	Horologiom	3
4	La nuit du requinqueur		8	Horologiom	4
5	Le grand rouage		6	Horologiom	5
6	L'enfant de mer		7	Finkel	1
7	Océane		4	Finkel	2
8	Génos		5	Finkel	3
9	Le secret		7	Finkel	4
10	Origine		7	Finkel	5
11	Esta		8	Finkel	6
12	Corruption		5	Finkel	7
13	Premier Cycle	Intégrale.		Finkel	INT1
86	Un chariot dans l'Ouest			Tuniques bleues (Les)	1
95	Des bleus et des tuniques			Tuniques bleues (Les)	10
96	Des bleus en noir et blanc			Tuniques bleues (Les)	11
97	Les bleus tourment cosaques			Tuniques bleues (Les)	12
98	Les bleus dans la gadoue			Tuniques bleues (Les)	13
99	Le blanc-bec			Tuniques bleues (Les)	14
100	Rumberley			Tuniques bleues (Les)	15
101	Bronco Benny			Tuniques bleues (Les)	16
102	El Padre			Tuniques bleues (Les)	17
103	Blue Retro			Tuniques bleues (Les)	18
104	Le David			Tuniques bleues (Les)	19
87	Du nord au sud			Tuniques bleues (Les)	2
105	Black Face			Tuniques bleues (Les)	20
106	Les cinq colons			Tuniques bleues (Les)	21

Le résultat obtenu n'est pas celui qu'on espérait, il y a apparemment deux erreurs de tri :

- le tri sur la série est incorrect : Horologiom est avant Finkel alors que F est avant H dans l'alphabet;
- le tri sur le tome paraît chaotique : 2 est après 19 ...

Prenons ces deux problèmes un par un.

## IX-A - Tri sur une clé étrangère

Le tri sur la série est un tri sur une clé étrangère.

On ne trie pas sur le libellé mais sur l'identifiant **id\_serie** dont la valeur est masquée par la liste de choix qui affiche le titre.

Dans la table **tblSeries** on voit bien que :

- Horologiom a comme identifiant 1;
- Finkel a comme identifiant 2;

Donc dans le tri Horologiom est avant Finkel.

Il faut alors effectuer le tri sur le champ **titre\_serie** pour assurer l'ordre alphabétique.

Ce champ est dans la table **tblSeries** et n'est donc pas disponible dans la requête.

Nous allons l'ajouter.

Retournez sur la requête **qryBds** en mode création.

Dans la partie haute (où est affichée la table **tblBds**), cliquez avec le bouton droit et choisissez **Afficher la table ...**.

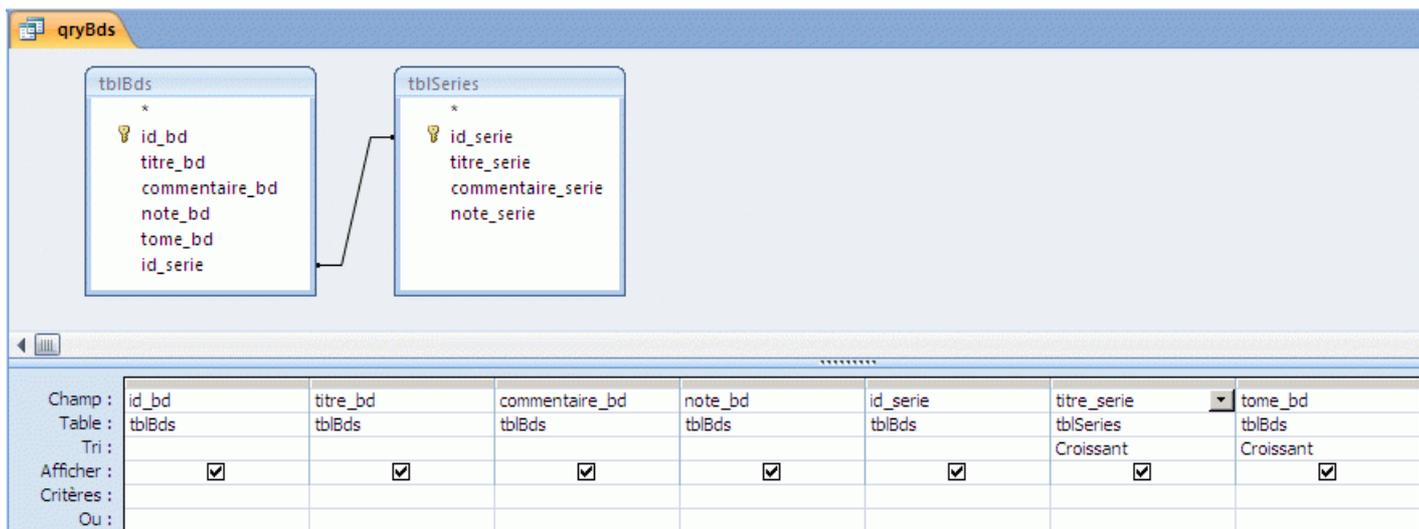
Sélectionnez la table **tblSeries**, cliquez sur **Ajouter** puis **Fermer**.

La table **tblSeries** s'affiche au côté de **tblBds**, avec le lien que nous avons vu dans la fenêtre des relations.

Ajoutez à la requête le champ **titre\_serie**, soit en double-cliquant sur le champ de la table, soit en le sélectionnant sur la ligne champ d'une nouvelle colonne.

Définissez un tri croissant pour ce champ, et retirez le tri sur le champ **id\_serie**.

Faites glisser la colonne du champ **titre\_serie** avant la colonne **tome\_serie** pour définir l'ordre de tri d'abord par titre de série puis par tome.



Passer en mode feuille de données :

id_bd	Titre de la BD	Commentaire de la BD	Note de la BD	id_serie	Titre de la série	Tome de la BD
6	L'enfant de mer			7	Finkel	1
7	Océane			4	Finkel	2
8	Génos			5	Finkel	3
9	Le secret			7	Finkel	4
10	Origine			7	Finkel	5
11	Esta			8	Finkel	6
12	Corruption			5	Finkel	7
13	Premier Cycle	Intégrale.		Finkel	Finkel	INT1
1	L'homme sans clef			7	Horologiom	1
2	L'instant du Damokle			7	Horologiom	2
3	Nahédig			6	Horologiom	3
4	La nuit du requinqueur			8	Horologiom	4
5	Le grand rouage			6	Horologiom	5
14	Eauxfolles			9	Nef des fous (La)	1
15	Pluvior 627			6	Nef des fous (La)	2
16	Turbulences			8	Nef des fous (La)	2

Finkel est bien maintenant trié avant Horologiom.

Problème résolu? Non pas tout à fait.

Regardez bien le nombre d'enregistrements en bas : il y en a 83.

Si on regarde dans la table **tblBds**, on en voit 84.

Une bande dessinée a donc disparu!

L'explication de cette disparition se trouve dans la relation (=jointure) entre nos tables.

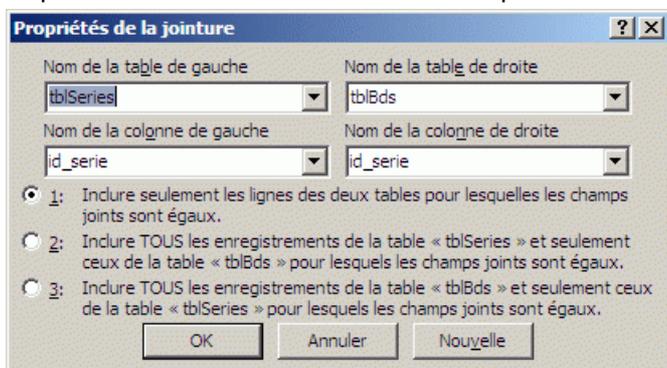
Nous avons laissé la relation par défaut qui est **une jointure équivalente**.

C'est-à-dire que ne s'affiche que les enregistrements pour lesquels il y a une entrée dans chacune des deux tables.

Hors nous avons une bande dessinée (*Adios Palomita*) qui n'est pas associée à une série; elle n'apparaît donc pas dans notre requête.

Nous allons vite corriger cela.

Repassez en mode création et double-cliquez sur le lien entre les deux tables :



Dans les propriétés de la jointure, on voit bien nos deux tables et les champs **id\_serie** de chacun d'entre-elles.

Puis en-dessous il y a trois boutons très importants qui définissent le type de jointure.

On a actuellement : *Inclure seulement les lignes des deux tables pour lesquelles les champs joints sont égaux.*

Ce n'est pas ce que nous souhaitons, nous voulons voir apparaître toutes les bandes dessinées et leur série si disponible.

Il faut donc choisir la troisième option : *Inclure TOUT les enregistrements de la table "tblBDs" et seulement ceux de la table "tblSeries" pour lesquels les champs joints sont égaux.*

Cliquez donc sur le troisième bouton puis sur OK.

Dans la partie haute, on voit une modification de la relation qui se transforme en flèche de **tblBDs** vers **tblSeries**.

Affichez la requête en mode feuille de données :

La BD *Adios Palomita* apparaît et nous avons bien l'intégralité de nos 84 BDs.

id_bd	Titre de la BD	Commentaire de la BD	Note de la BD	id_serie	Titre de la série	Tome de la BD
143	Adios Palomita	Western		9		
6	L'enfant de mer			7	Finkel	1
7	Océane			4	Finkel	2
9	Génes			6	Finkel	3

La BD qui n'a pas de série a donc un champ **id\_serie** vide.

 *Maintenant que nous avons mis en évidence l'intérêt des type de jointure, nous les modifierons dans un chapitre suivant dans la fenêtre des relations.*

*Ainsi on aura par défaut les jointures désirées.*

*Si besoin on pourra définir un autre type de jointure pour une requête en particulier.*

*Les jointures définies dans la fenêtre des relations peuvent être modifiées dans chaque requête comme nous venons de le voir.*

## IX-B - Tri sur un champ texte contenant des nombres

Le tri alphabétique sur la série étant maintenant corrigé, occupons-nous du tri sur le tome.

Ce tome a été défini en champ texte afin de permettre la saisie à la fois de numériques et de textes.

Le problème lors du tri est que un champ texte est trié par ordre alphabétique : donc par exemple 2 vient après 12.

Il est possible de tricher en rajoutant les 0 non significatifs à la saisie : par exemple 02 au lieu de 2.

Ainsi 02 est bien trié avant 12.

Mais il faut alors imposer la saisie de ces 0 non significatifs, et il faut aussi se fixer un nombre de chiffres obligatoire.

Si on a plus de 99 albums, il faudrait alors imposer la saisie sur 3 chiffres.

Ceci étant peu pratique, nous allons garder une saisie libre et essayer de trouver une solution lors du tri.

Il est tout simplement possible de formater les nombres et de trier sur ces nombres formatés.

Il faut ajouter un champ calculé à la requête.

En effet il est possible de faire des calculs dans les requêtes (ce "calcul" sera en fait simplement un formatage du champ).

Si vous connaissez déjà les fonctions disponibles, vous pouvez directement les écrire à la place du nom du champ.

Sinon cliquez avec le bouton droit sur la ligne champ dans la première colonne vide et cliquez sur le bouton **Créer...**

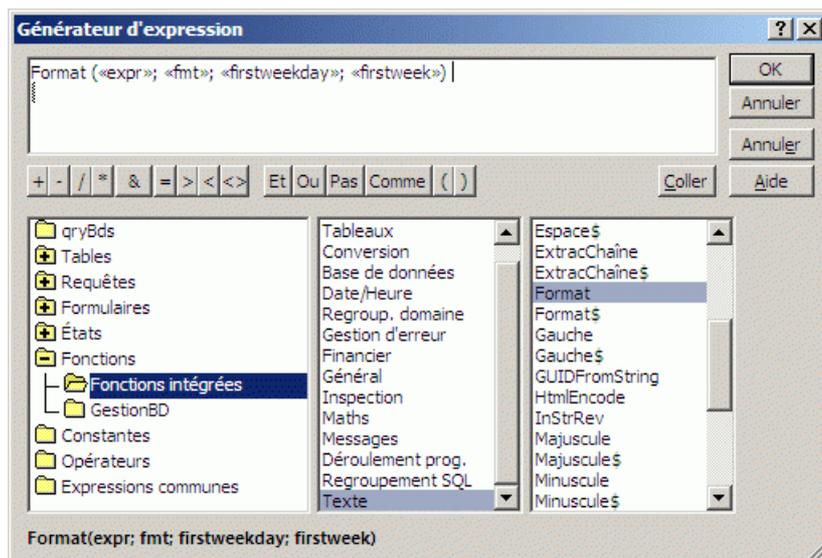
Le générateur d'expression s'ouvre; il permet de lister les différentes expressions disponibles.

Choisissez **Fonctions** puis **Fonctions intégrées**.

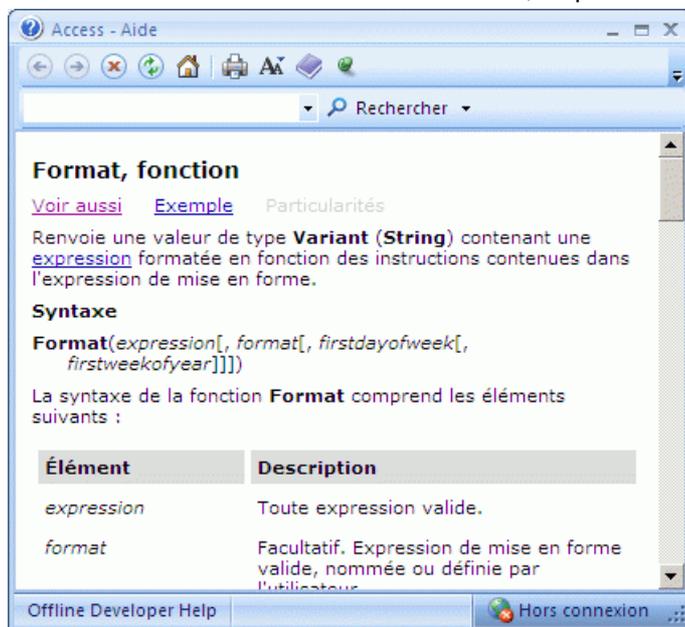
Dans la liste du milieu, sélectionnez **Texte** car nous souhaitons travailler sur du texte.

Puis dans la liste à droite, vous avez toutes les fonctions disponibles pour manipuler les textes.

La fonction qui nous intéresse est la fonction **Format**, double-cliquez dessus.



La fonction apparait dans la case en haut, avec ses arguments à remplir. Pour obtenir de l'aide sur la fonction choisie, cliquez sur le bouton **Aide**.



L'aide de la fonction s'affiche; pour plus d'information sur le formatage de nombre, cliquez sur **Voir aussi** puis sur **Formats numériques définis par l'utilisateur**.

On lit alors dans l'aide que "0" peut être utilisé dans la fonction format pour afficher des zéros non significatifs.

Modifions donc l'appel de la fonction :

- remplacez «**expr**» par la valeur à formater : le champ **[tome\_bd]**;
- remplacez «**fmt**» par le format désiré : **"0000"**.
- supprimer «**firstweekday**» e t«**firstweek**» qui servent au formatage de dates.

Notez ici l'utilisation des crochets qui encadrent le nom du champ; il est important d'utiliser des crochets pour différencier un nom de champ d'une chaîne de caractères.

Souvent, Access comprend tout seul qu'il s'agit d'un champ; parfois il est nécessaire d'ajouter les crochets.

Pour le format par contre, on l'encadre de guillemets car c'est une chaîne de caractères.

On choisit par exemple quatre caractères 0 afin de formater nos numéros de tomes sur quatre chiffres.

On est alors tranquille jusqu'au tome 9999, c'est largement suffisant.

L'expression est donc : *Format ([tome\_bd]; "0000")* .

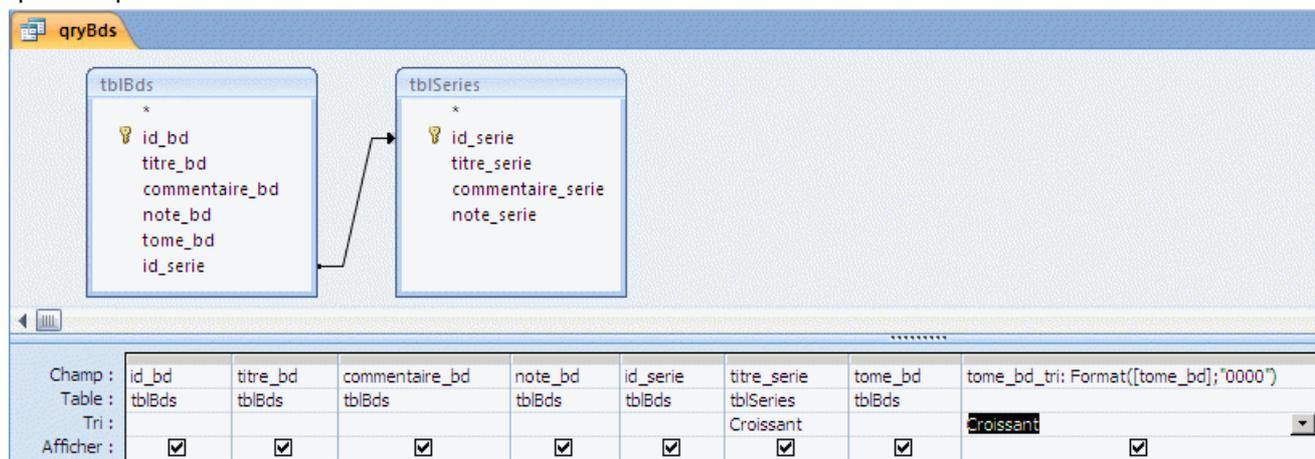
Cliquez sur OK, l'expression est ajoutée sur la ligne Champ.

Access a ajouté **Expr1**: au début de l'expression : c'est un alias.

C'est-à-dire que notre champ calculé s'appelle **Expr1**.

Donnons-lui un nom plus adéquat : **tome\_bd\_tri** par exemple.

Ce champ étant destiné au tri, ajoutez-lui un tri croissant et retirez en même temps le tri sur la colonne **tome\_bd** qui n'est plus utile.



Affichez les données pour vérifier le tri :

id_bd	Titre de la BD	Commentaire de la BC	Note de la BD	id_serie	Titre de la série	Tome de la BD	tome_bd_tr
143	Adios Palomita	Western		9			
6	L'enfant de mer			7	Finkel	1	0001
7	Océane			4	Finkel	2	0002
8	Génos			5	Finkel	3	0003
9	Le secret			7	Finkel	4	0004
10	Origine			7	Finkel	5	0005
11	Esta			8	Finkel	6	0006
12	Corruption			5	Finkel	7	0007
13	Premier Cycle	Intégrale.			Finkel	INT1	INT1
1	L'homme sans clef			7	Horologiom	1	0001
2	L'instant du Damokle			7	Horologiom	2	0002
3	Nahédiq			6	Horologiom	3	0003
4	La nuit du requinqueur			8	Horologiom	4	0004
5	Le grand rouage			6	Horologiom	5	0005
14	Eauxfolles			9	Nef des fous (La)	1	0001
15	Pluvior 627			6	Nef des fous (La)	2	0002
16	Turbulences			8	Nef des fous (La)	3	0003
17	Au turf				Nef des fous (La)	4	0004
18	Puzzle			6	Nef des fous (La)	5	0005
19	Les Chemins énigmatique				Nef des fous (La)	6	0006
20	Terminus			9	Nef des fous (La)	7	0007
21	Puzzle Gribouillis	Hors série			Nef des fous (La)	COF2	COF2
22	La genèse	Hors série			Nef des fous (La)	HS01	HS01
23	Histoire d'Eauxfolles - Co	Hors série			Nef des fous (La)	HS02	HS02
24	La nuit Riv	Hors série			Nef des fous (La)	HS03	HS03

Les tris sont désormais corrects : la liste des BDs est triée par série, puis au sein d'une série les BDs sont triées par tomes.

Nous remarquons également que le formatage n'a pas affecté les tomes non numériques (HS01, HS02, ...) ce qui nous arrange beaucoup.

Le champ **tome\_bd\_tri** qui ne sert qu'au tri peut être masqué à l'affichage : pour cela en mode création décochez la case **Afficher** de cette colonne.

## X - Modification des jointures des relations

Nous l'avons vu lors de la précédente requête, les jointures par défaut ne nous conviennent pas toujours.

Nous allons donc passer en revue les relations entre nos tables afin de mieux les définir.

Ouvrez la fenêtre des relations :

Soit dans le ruban : onglet **Outils de base de données**, bouton **Relations**.

Soit avant Access 2007: menu **Outils => Relations**.

Si besoin : clic droit => **Afficher toutes les relations**.

Nous allons d'abord modifier la relation vu précédemment entre **tblBds** et **tblSeries**.

Double-cliquez sur le lien entre ces deux tables.



Notez que la relation est définie de la table **tblSeries** vers la table **tblBds** (de la table à gauche vers la table à droite). On trouve sur cette fenêtre une case à cocher **Appliquer l'intégrité référentielle**.

Si cette case est cochée, cela signifie qu'il ne peut y avoir d'enregistrements orphelins.

C'est-à-dire pour notre cas qu'on ne peut pas avoir dans la table **tblBds** une bd appartenant à une série qui n'existe pas dans la table **tblSeries**.

Très intéressante, cette fonctionnalité empêche notamment de supprimer une série si des BDs sont définies comme lui appartenant.

Une fois l'intégrité référentielle activée, il est possible de demander une mise à jour ou suppression en cascade des enregistrements liés.

Pour la modification ce n'est pas possible le champ **id\_serie** de **tblSeries** et auto-incrémenté et ne peut être modifié. Pour la suppression, c'est utile dans certains cas mais pas tous : si on supprime une série, toutes les Bds associés serait supprimées.

Il sera la plupart du temps préférable pour notre application de ne pas cocher cette case de suppression en cascade.

Cochez donc la case **Appliquer l'intégrité référentielle**.

Cliquez ensuite sur le bouton **Type Jointure...**



Nous retrouvons à cet endroit le type de jointure comme il était présenté dans la requête.

Choisissez comme tout à l'heure la troisième option : *Inclure TOUT les enregistrements de la table "tblBds" et seulement ceux de la table "tblSeries" pour lesquels les champs joints sont égaux.*

Cliquez deux fois sur OK pour valider.

La relation affichée présente alors une flèche de **tblBds** vers **tblSeries**.

La jointure mise en place s'appelle une **jointure à droite**.

*A droite* car on affiche tous les enregistrements de la table de droite (**tblBds** étant à droite dans la fenêtre de la jointure)

Définir le type de jointure dans cette fenêtre des relations n'a pour effet que de présenter ce type de jointure par défaut lors de la création d'une requête.

Par contre l'intégrité référentielle a une réelle incidence sur la saisie des données.

Essayer maintenant de supprimer une série dans la table série, un message d'erreur est affiché :



Nous n'aurons ainsi par d'erreur de suppression de série : il faudra d'abord supprimer les BDs avant de supprimer une série.

Poursuivons avec les autres relations :

→ Entre **tblSeries** et **tblAPourImageDeSerie** :

Option 2 = Inclure TOUT les enregistrements de la table "tblSeries" et seulement ceux de la table "tblAPourImageDeSerie" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblAPourImageDeSerie** et **tblImages** :

Option 3 = Inclure TOUT les enregistrements de la table "tblAPourImageDeSerie" et seulement ceux de la table "tblImages" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblBds** et **tblAPourImageDeBd** :

Option 2 = Inclure TOUT les enregistrements de la table "tblBds" et seulement ceux de la table "tblAPourImageDeBd" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblAPourImageDeBd** et **tblImages** :

Option 3 = Inclure TOUT les enregistrements de la table "tblAPourImageDeBd" et seulement ceux de la table "tblImages" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblBds** et **tblEstRealisePar** :

Option 2 = Inclure TOUT les enregistrements de la table "tblBds" et seulement ceux de la table "tblEstRealisePar" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblEstRealisePar** et **tblAuteurs** :

Option 3 = Inclure TOUT les enregistrements de la table "tblEstRealisePar" et seulement ceux de la table "tblAuteurs" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblEstRealisePar** et **tblRoles** :

Option 3 = Inclure TOUT les enregistrements de la table "tblEstRealisePar" et seulement ceux de la table "tblRoles" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Entre **tblAuteurs** et **tblEstUn** :

Option 2 = Inclure TOUT les enregistrements de la table "tblAuteurs" et seulement ceux de la table "tblEstUn" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

**Appliquez la mise à jour et la suppression en cascade.**

Cela nous permettra lors de la suppression d'un auteur de supprimer automatiquement ses rôles.

→ Entre **tblRoles** et **tblEstUn** :

Appliquez l'intégrité référentielle.

Option 2 = Inclure TOUT les enregistrements de la table "tblRoles" et seulement ceux de la table "tblEstUn" pour lesquels les champs joints sont égaux.

→ Entre **tblExemplaires** et **tblEtats** :

Appliquez l'intégrité référentielle.

Option 3 = Inclure TOUT les enregistrements de la table "tblExemplaires" et seulement ceux de la table "tblEtats" pour lesquels les champs joints sont égaux.

- Entre **tblExemplaires** et **tblFormats** :

Appliquez l'intégrité référentielle.

Option 3 = Inclure TOUT les enregistrements de la table "tblExemplaires" et seulement ceux de la table "tblFormats" pour lesquels les champs joints sont égaux.

→ Entre **tblExemplaires** et **tblBds** :

Option 3 = Inclure TOUT les enregistrements de la table "tblExemplaires" et seulement ceux de la table "tblBds" pour lesquels les champs joints sont égaux.

Appliquez l'intégrité référentielle.

→ Pour terminer, ajoutons une relation entre les tables **tblEstUn** et **tblEstRealisePar**.

Nous avons en effet évoqué lors de la conception une contrainte sur le couple auteur+rôle.

On ne doit pas pouvoir ajouter un auteur dans un rôle qui n'est pas défini pour lui.

Ajoutez une relation entre **tblEstUn** et **tblEstRealisePar** :

- faites glisser la souris du champ **id\_auteur** de la première table vers le champ **id\_auteur** de la deuxième table.

- ajoutez le champ **id\_role** à la liste.
- dans *type de jointure*, choisissez l'option 3 = *Inclure TOUT les enregistrements de la table "tblEstRealisePar" et seulement ceux de la table "tblEstUn" pour lesquels les champs joints sont égaux.*
- appliquez l'intégrité référentielle.

La contrainte d'existence du couple auteur+rôle sera vérifié par cette relation grâce à l'intégrité référentielle.

Pour ces jointures, on a défini uniquement des jointures à gauche ou à droite.

Cela nous évitera des problèmes au cas où un enregistrement serait manquant.

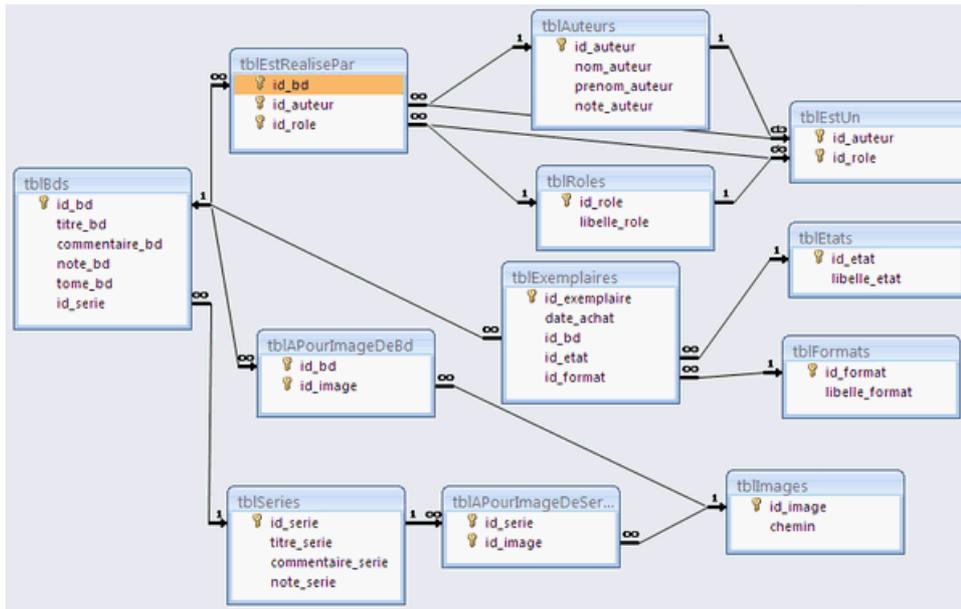
Par exemple si on ne définit pas d'auteur pour une BD, la BD ne disparaîtra pas de nos requêtes.

L'auteur sera vide, tout simplement.

Nous pourrons par la suite modifier ces jointures indépendamment dans chaque requête.

D'un autre côté, l'intégrité référentielle refusera par exemple de nous laisser supprimer un auteur si des Bds lui sont associées.

Notez que lors de la mise en place de l'intégrité référentielle, Access a ajouté des 1 et des signes infini :



Ne confondez pas ces symboles avec les cardinalités du MLD.

Ces signes sont définis à 1 si le champ de la jointure est sans-doublons (clé primaire par exemple), sinon à infini.

Vous pouvez télécharger la base de données contenant les tables avec leurs relations et la première requête :

**Les tables et relations au format Access 2000 ([HTTP](http://arkham46.developpez.com/articles/access/didacticiel-bd/))**

## XI - Par où commencer?

Nous voici prêt à créer l'interface utilisateur de notre application.

Nous sommes confrontés au même problème que lors de la création des tables, encore une fois le syndrome de la page blanche.

Avant tout je vais vous présenter brièvement les objets à notre disposition pour créer l'interface :

Tout d'abord les formulaires :

Ceux-ci servent à **afficher** des informations à **l'écran**.

Il est possible de les lier à une table ou une requête : dans ce cas les ajouts/modifications/suppressions sont repercutés dans la table.

Dans le cas d'un formulaire non lié à un table, on parle de formulaire indépendant.

On distingue deux types de formulaires :

- les formulaires **simples** qui affichent un seul enregistrement à la fois: ce sont des fiches.



- les formulaires **continus** qui affichent chaque enregistrement l'un en-dessous de l'autre : ce sont des listes.



Généralement les formulaires continus affichent moins d'informations par enregistrement que les formulaires simples. Par exemple on affichera une liste de bandes dessinées dans un formulaire continu, avec des informations allégées (sans le commentaire).

Et on pourra ouvrir la fiche d'une BDS pour saisir des informations complémentaires (le champ commentaire qui n'était pas visible sur la liste par soucis de lisibilité).

Ensuite les états :

Les états sont destinés à l'**impression** des données.

Il n'est pas conseillé d'essayer d'imprimer un formulaire : le résultat obtenu est souvent peu lisible.

On créera un état pour toute impression papier car l'état permet de formater correctement la page d'impression.

Comme pour les formulaires, il est possible de le lier à une source de données (table ou requête).

Les états permettent notamment de gérer des groupements, ce que le formulaire ne propose pas.

Il sera par exemple possible de regrouper les bandes dessinées par série dans un état.

Nous détaillerons ces possibilités dans un prochain chapitre lors de la création de nos formulaires et états.

Enfin les barres de menus ou le ruban :

Les barres de menus peuvent être :

- soit fixes en haut de la fenêtre, comme la barre de menu standard d'Access par exemple qui contient *Fichier, Edition, ...*;

- soit flottants; on parle alors de menus contextuels : ce sont ceux auxquels on accède par click droit avec la souris.

A partir d'Access 2007, les menus sont remplacés par **un ruban**.

Les menus contextuels existent toujours dans cette version mais ne sont modifiables que par code VBA.

Afin de se lancer tête baissée dans la création de ces objets, prenons le temps de réfléchir à l'interface utilisateur à mettre en place.

## XII - Maquette de l'interface

N'étant pas formé à l'utilisation d'un quelconque outil pour modéliser l'interface, je vais vous proposer la méthode papier+crayon.

Commencer par le formulaire des auteurs.

Ajouter le nombre de bd par auteur (une requête + un champ avec fonction de domaines = 2 tutos à mettre en liens).

## XIII - Conclusion

## XIV - Téléchargement

**Application de gestion de BD ([HTTP](#))**

**Le document d'analyse pour AnalyseSI ([HTTP](#))**