



# Cours LES SYSTÈMES DE GESTION DE BASES DE DONNÉES (SGBD) : MICROSOFT ACCESS

Université Cheikh Anta Diop (UCAD) de Dakar

Faculté des Sciences Economiques et de Gestion (FASEG)

Département de Mathématiques de la Décision (DMD)

Pr. Babacar M. NDIAYE & Dr. Mouhamadou A.M.T. BALDÉ. **Année 2022-2023.**

Laboratoire de Mathématiques de la Décision et d'Analyse Numérique (LMDAN)

*babacarm.ndiaye@ucad.edu.sn*, <http://lmdan.ucad.sn>

## Notes

Ces notes de cours correspondent à un enseignement de Masters 1 et 2 de la FASEG. Ce cours est le résultat d'une réflexion technique pédagogique dont nous espérons qu'il apportera aux étudiants une stimulation intellectuelle et un encouragement à persévérer, chaque fois que la compréhension d'un phénomène économique ou de gestion leur posera des difficultés. Bien qu'ayant relu attentivement toutes les notes, il reste plusieurs imperfections. Nous demandons aux étudiants de nous en excuser, et de nous les signaler afin d'en améliorer la qualité. Leurs camarades de l'année prochaine leur en seront reconnaissants.

# Table des matières

<b>1</b>	<b>LES SYSTÈMES DE GESTION DE BASES DE DONNÉES (SGBD)</b>	<b>3</b>
1.1	L'ORGANISATION DES DONNÉES DANS LA BASE DE DONNÉES . . . . .	3
1.1.1	La base de données . . . . .	3
1.1.2	L'organisation des données . . . . .	4
1.2	UTILISATION DE PLUSIEURS TABLES . . . . .	7
1.3	CLEF PRIMAIRE RÉALISÉE À PARTIR DE PLUSIEURS CHAMPS . . . . .	10
1.4	AMÉLIORATIONS DES SCHÉMAS . . . . .	11
1.4.1	Améliorations . . . . .	11
1.4.2	Les index . . . . .	12
<b>2</b>	<b>APPLICATIONS</b>	<b>14</b>
2.1	CRÉATION D'UNE BASE DE DONNÉES . . . . .	14
2.2	LES TABLES . . . . .	15
2.2.1	Les types de données . . . . .	17
2.2.2	Les propriétés des champs . . . . .	18
2.2.3	Le format d'affichage . . . . .	19
2.2.4	Le masque de saisie . . . . .	24
2.2.5	Les listes de choix . . . . .	27
2.2.6	Exemple . . . . .	27
2.3	RELATIONS . . . . .	28
2.3.1	La relation de type « un à plusieurs » (1 - n) . . . . .	28
2.3.2	La relation de type « plusieurs à plusieurs » (n - m) . . . . .	30
2.3.3	La relation de type « un - un » (1 - 1) . . . . .	35
2.3.4	Exemple . . . . .	35
2.4	LES REQUÊTES . . . . .	36
2.5	LES FORMULAIRES . . . . .	39
2.6	LES ÉTATS . . . . .	41
2.7	LES MACROS . . . . .	41

# 1 LES SYSTÈMES DE GESTION DE BASES DE DONNÉES (SGBD)

Nous abordons dans ce chapitre une introduction au base de données (son abréviation est BD, en anglais DB, database) qui est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

## 1.1 L'ORGANISATION DES DONNÉES DANS LA BASE DE DONNÉES

### 1.1.1 La base de données

**Définition 1.1.** *Une base de données est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun.*

Elle doit avoir un certain nombre de caractéristiques :

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

- **Sécurité des données stockées** : la base de données doit permettre un système de sécurité permettant de gérer les droits d'accès aux informations par les utilisateurs.

**Définition 1.2.** *Un Système de Gestion de Bases de Données (S.G.B.D.) représente un ensemble coordonné de logiciels qui permet de décrire, manipuler, traiter les ensembles de données formant la base.*

Il doit également assurer la sécurité et la confidentialité des données dans un environnement où de nombreux utilisateurs ayant des besoins variés peuvent interagir simultanément sur ces données.

Le SGBD sert d'interface entre les programmes d'application des utilisateurs d'une part, et la base de données d'autre part.

Il doit permettre **l'administration des données** c'est à dire assurer la définition des structures de stockage et des structures de données et le suivi de leurs évolutions.

### 1.1.2 L'organisation des données

Avant d'exploiter les données contenues dans la base, il faut trouver le meilleur moyen de représenter le monde réel en structurant la base de données de manière à pouvoir l'exploiter la plus simplement par la suite.

Dans Access, comme dans les autres SGBD, les informations doivent être segmentées en données qui sont stockées dans des tables. Une table est donc un ensemble de données, organisées en lignes et en colonnes (les tables sont développés au chapitre 2.2).

On peut stocker dans une table n'importe quel type d'information (texte, date, chiffres, son, monétaire, etc.)

**Chaque table est divisée en enregistrements**, les enregistrements étant l'ensemble des données relatives à la même information.

Prenons l'exemple d'un crédit à remboursement, pour les clients de la Société Générale de Banque au Sénégal(SGBS). Pour ce crédit à remboursement, nous allons utiliser une table noté «remboursement» qui contiendra toutes les informations concernant chaque client, chaque enregistrement contiendra les informations relatives à un (et un seul) client (numéro (permet d'identifier le client par ses coordonnées), remboursement en capital,

intérêts, etc.).

Comme on vient de le voir, chaque enregistrement contient un ensemble d'informations (numéro, remb. en capital, intérêts) qui ont elles-mêmes divisé l'enregistrement en plusieurs parties, chaque partie s'appelle un champ.

Le champ contient une partie des informations de chaque enregistrement (Figure 1).

**Bon à savoir :** Quelques définitions :

- La base de données : C'est l'ensemble des tables utilisées pour gérer l'information.
- La table : c'est un ensemble de données organisées en rangées et en colonnes (cf chapitre 2.2).
- Le champ (ou colonne) : C'est l'unité d'information dans une table. Une table est constituée de différents champs.
- L'enregistrement (ou la ligne) : C'est l'ensemble des données relatif à la même information (le 1er enregistrement de la table contient les informations relatives au client).

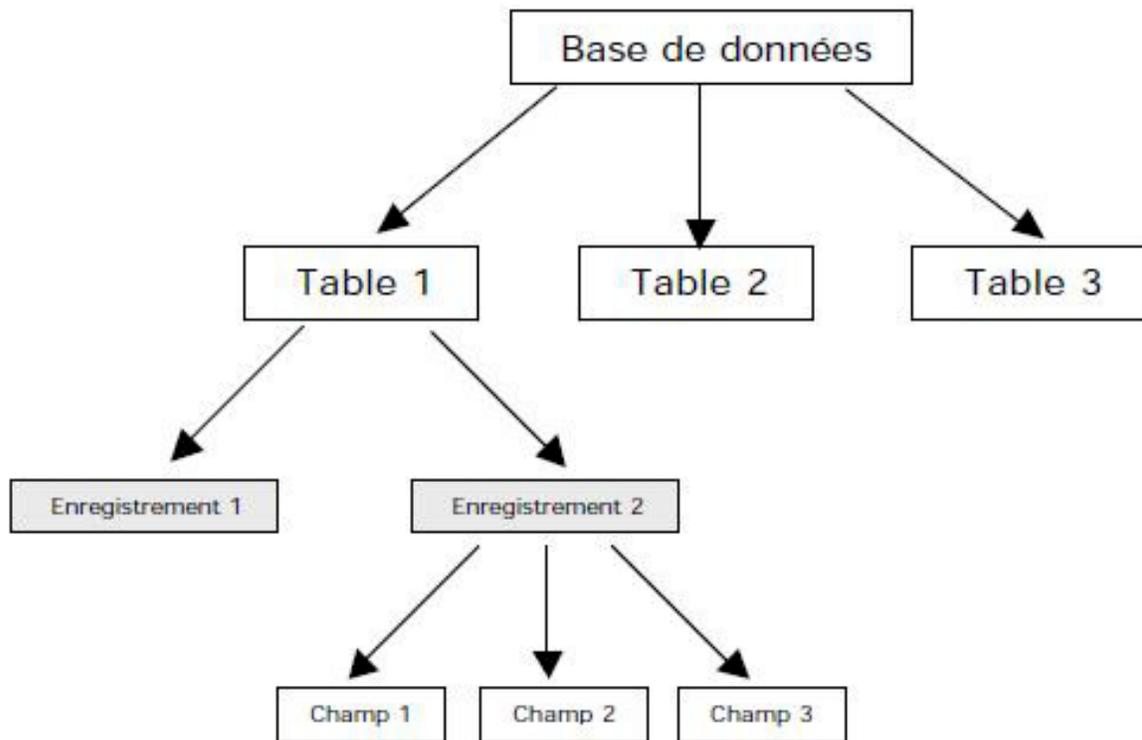


FIGURE 1 – Organisation de la base de données

Si nous reprenons notre exemple du crédit à remboursement, nous allons avoir une table qui peut ressembler à la Figure 2 : L'organisation de la table doit être optimale. La table

Numéro	Remb en capital	Intérêts
1	416.67	37.50
2	416.67	35.94
3	416.67	34.38
4	416.67	32.81
5	416.67	31.25
6	416.67	29.69
7	416.67	28.13
8	416.67	26.56
9	416.67	25.00
10	416.67	23.44

FIGURE 2 – Aperçu table

peut être décomposée en éléments aussi petits que possible.

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

Le SGBD, pour retrouver les données qu'on lui demande, travaille avec le contenu des champs, il lui sera donc difficile, dans la table «remboursement» telle qu'elle est, de retrouver les mensualités d'un client de l'agence du Point E se trouvant mélangée au numéro d'identifiant, de même, on ne pourra pas récupérer le capital restant dû, celui-ci pouvant se trouver dans le même champ que le numéro.

<b>Mensualités</b>	<b>Capital restant dû</b>	<b>Remb en capital</b>	<b>Intérêts</b>
454.17	10000.00	416.67	37.50
452.60	9583.33	416.67	35.94
451.04	9166.67	416.67	34.38
449.48	8750.00	416.67	32.81
447.92	8333.33	416.67	31.25
446.35	7916.67	416.67	29.69
444.79	7500.00	416.67	28.13
443.23	7083.33	416.67	26.56
441.67	6666.67	416.67	25.00
440.10	6250.00	416.67	23.44

FIGURE 3 – Exemple d’une meilleure organisation de la table

<b>Compte</b>
Nom
Prénom
Num. compte
Agence
Type
Solde

FIGURE 4 – La table compte

Comme on ne peut pas toujours prévoir, lors de la création de la structure de la table, de l’usage qui en sera fait plus tard, il faut diviser le plus possible les données, sachant qu’il sera toujours compliqué (et risqué) de modifier la structure de la table par la suite lorsqu’elle sera remplie. Une meilleure organisation de la table serait, par exemple, celle décrite en Figure 3.

## 1.2 UTILISATION DE PLUSIEURS TABLES

Prenons l’exemple de la gestion d’un compte bancaire. On pourrait être tenté de créer une seule table contenant les informations suivantes (Figure 4) : Et nous n’aurions pas spécialement tort, toutes les informations relatives à un Num. compte (Numéro de compte) s’y trouvent. Pourquoi faire plus compliqué ?

Néanmoins, nous allons voir qu’il est plus judicieux de créer ici deux tables (voir Figure 5) :

✓ Pourquoi ce choix ?

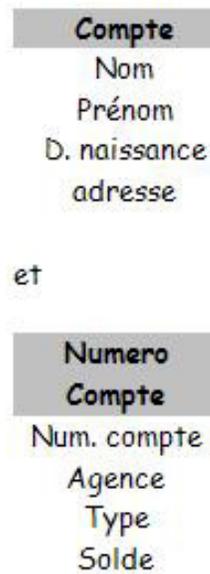


FIGURE 5 – Élargissement sous 2 tables

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

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

Un client peut avoir plusieurs numéros de comptes (compte courant, compte épargne, etc.), a un compte peut correspondre plusieurs opérations de retrait.

On en déduit directement une table «Compte» et une table «Numero Compte».

Ce qui, au passage, nous permet d'enrichir les informations sur les clients, on peut en profiter pour par exemple ajouter la date de naissance ou son adresse, ce qui aurait été pénible dans la 1ère version.

En effet, il aurait fallu, pour chaque numéro de compte d'un client donné, ressaisir à chaque coup sa date de naissance et son adresse (sans compter les erreurs de frappe d'un numéro de compte à l'autre), là les informations ne sont saisies qu'une fois, ce qui est

Compte
Nom
Prénom
Num. compte
Agence
Type
Solde

FIGURE 6 – Table Compte

d’abord moins fatigant (et ça, c’est le plus important) et ensuite, évite les fautes de frappes et autres.

Maintenant que nous avons deux tables, mais du coup, on ne sait plus qui a fait quoi, les informations sont éparpillées.

Il va donc falloir trouver un moyen pour savoir exactement à quel client correspond chaque numéro de compte.

Pour cela on va ajouter dans la table client un champ qui va identifier de façon unique chaque enregistrement, ce champ d’identification unique est appelé **clef primaire**.

✓ A quoi sert cette clef ?

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

Le numéro de la Carte d’Identification Nationale (CIN) est un bon exemple de champ clef primaire car il identifie de manière unique un individu. Même si deux individus ont le même nom et le même prénom, ils ne peuvent avoir le même numéro d’une pièce d’identité.

La table «Compte» ressemble alors à ceci (le champ qui va porter la clef primaire (CIN) est en gras) (voir Figure 7). Et pour pouvoir associer un numéro de compte à un client, il va falloir ajouter un champ dans la table «Numero Compte», ce champ contient le CIN du client ayant le numéro de compte (voir Figure 7).

Numero Compte
CIN
Num. compte
Agence
Type
Solde

FIGURE 7 – Table Numéro de Compte

Nous voyons ici que cette organisation répond à une des exigences des bases de données, à savoir **la non redondance des informations**.

Dans la première organisation, le nom et le prénom du client étaient répétés autant de fois qu'il avait ouvert un nouveau compte, ici il ne sera présent qu'une seule fois.

Lorsqu'on voudra afficher les caractéristiques d'un numéro de compte (nombres débiteurs, nombres créditeurs, etc.), Access, grâce au CIN du client présent dans l'enregistrement de la table «Numero Compte» ira chercher automatiquement les informations le concernant dans la table «Compte».

Cette organisation a un énorme avantage, le jour où on veut modifier le nom ou le prénom d'un client, cette modification sera répercutée automatiquement sur les comptes qu'il aura ouvert.

### 1.3 CLEF PRIMAIRE RÉALISÉE À PARTIR DE PLUSIEURS CHAMPS

Lorsqu'on est sûr que le contenu d'une combinaison de champs ne se répétera pas dans une table, on peut utiliser cette combinaison de champs comme clef primaire.

On aurait pu, à la place d'un CIN du client, utiliser la combinaison «nom + prénom + date de naissance», parce qu'il est rare que deux auteurs différents aient le même nom et le même prénom et la même date de naissance.

Mais ce type de combinaison complexe est difficile à maintenir (il est plus simple de taper un numéro qu'une combinaison complexe de noms et de date, sans parler des erreurs de frappes). En plus ceci prendrait plus de place dans la base de données (on ne voit pas trop l'intérêt de découper les informations en deux tables si c'est pour retrouver dans le

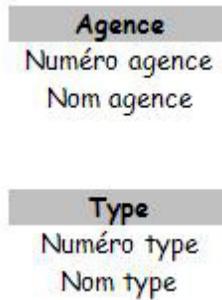


FIGURE 8 – Recréation de deux tables

CIN du client utilisé dans la table «Numero Compte» quasiment toutes les informations contenues dans la table «Compte»).

**Remarque 1.1.** *Ce genre de clef primaire complexe fonctionne mal dans Access.*

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

Établir une relation entre les tables de la base de données va permettre de réunir (pour les exploiter ensemble, les analyser, etc.) les informations réparties dans différentes tables. La liaison se fait entre les tables à partir de la clef primaire, ici, on obtiendra la liste des numéros de compte d'un client grâce à la liaison entre la table Compte et la table Numero Compte sur la clef primaire CIN.

## 1.4 AMÉLIORATIONS DES SCHÉMAS

### 1.4.1 Améliorations

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

En effet, pour chaque numéro de compte, on va indiquer l'agence (Point E, Centenaire, Baobab, etc.) et le type du compte (compte courant, compte épargne, etc.).

Ces informations vont être redondantes et pourraient être optimisées de la façon suivante :

1. On crée une table «Agence» et une table «Type», comme l'indique la Figure 8.

Numero Compte
CIN
CIN 2
CIN 3
Num. Compte
Numero Agence
Numero Type
Solde

FIGURE 9 – Modification de la table Numero Compte

2. On modifie la table «Numero Compte», comme l’indique la Figure 9.

✓ Maintenant, au lieu d’indiquer explicitement pour chaque numéro de compte son agence et son type, on indiquera un numéro d’agence et un numéro de type.

L’intérêt est triple :

- gain de place en mémoire (stocker un numéro prend moins de place que stocker une chaîne de caractères),
- gain de vitesse (entrer un numéro est plus rapide que taper le mot complet),
- optimisation de la structure (on pourra changer le nom d’un type de compte, ce changement affectera instantanément l’ensemble des numéros de compte).

#### 1.4.2 Les index

Pour optimiser les temps de recherche dans les tables, Access (et d’ailleurs tous les SGBD) ont besoin d’index.

**Définition 1.3.** *L’index permet à Access de retrouver plus rapidement les enregistrements concernés ; il accélère aussi les opérations de tri.*

✓ **Comment on procède ?**

Les index fonctionnent avec une table comme la table des matières avec un livre. Lorsqu’on indexe un champ, Access stocke hors de la table les valeurs de ce champ. Lorsqu’on effectue une recherche sur ce champ, plutôt que de le chercher dans la table, Access va faire sa

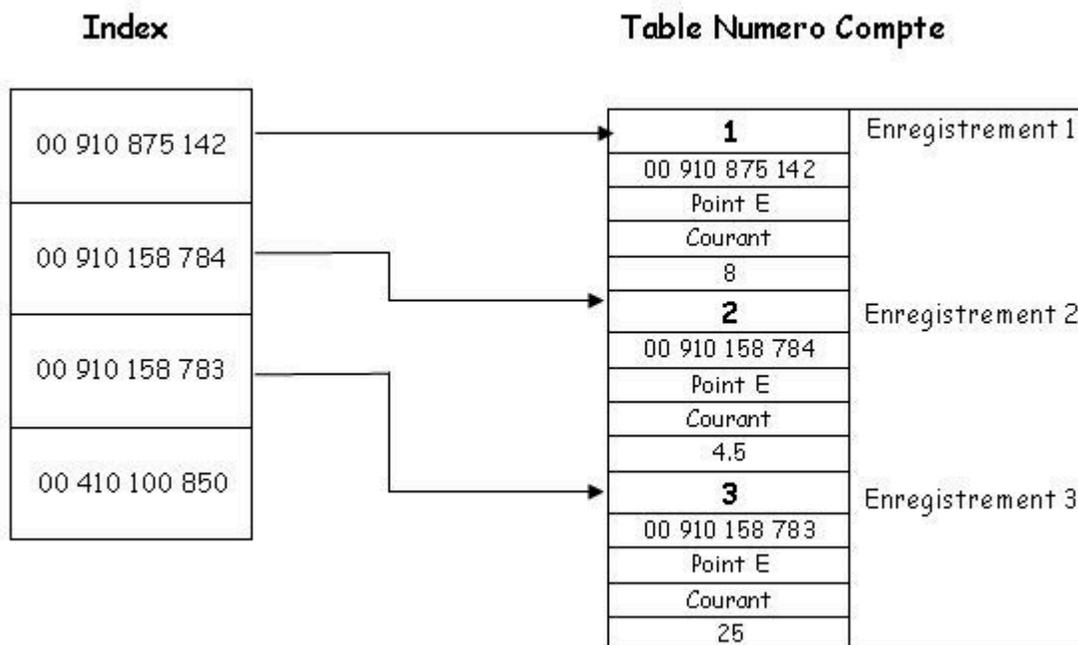


FIGURE 10 – Trouver un enregistrement via l'index

recherche dans l'index, une fois qu'il aura trouvé, il affichera le contenu de l'enregistrement associé à cet index.

**Exemple 1.1.** *Supposons que la table «Numero Compte» soit indexée sur le Num. compte comme indiqué dans la figure 10.*

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

Un index peut être composé d'un ou de plusieurs champs, on aurait pu indexer la table Numero Compte sur un index «solde» par exemple (si ça peut servir à quelque chose), de même, on peut utiliser plusieurs index pour une table, par exemple, on aurait pu indexer la table Numero Compte sur les champs Agence, Type et Solde.

### ✓ Pourquoi alors ne pas tout indexer ?

L'idée peut sembler bonne, cela accélérerait les recherches et les tris mais ralentirait toutes les mises à jour de la table, en effet, à chaque fois qu'un champ index est modifié dans la

table, il faut le modifier également dans l'index, ce qui peut coûter très cher en temps. On utilisera donc les index avec parcimonie et uniquement s'ils s'avèrent utiles.

## 2 APPLICATIONS

Nous décrivons dans ce chapitre comment créer une base de données, créer une table, faire un requête, etc. Le premier exemple sera noté ici par «Banque») qui va contenir les tables dont nous avons vu la structure dans la première partie.

### 2.1 CRÉATION D'UNE BASE DE DONNÉES

Nous considérons la base de données «Banque») contenant les tables dont nous avons vu la structure dans le chapitre précédent.

1. Dans «**Catégories de modèles**» cliquez sur «**À la une**».
2. Dans «**Nouvelle base de données vide**» cliquez sur «**Base de données vide**».
3. Entrez «**Banque**» comme nom de la base (à droite dans : nom de fichier).
4. Cliquez sur «**Créer**». Attention à l'emplacement du fichier ! Access l'enregistre par défaut dans «Mes documents», cf Figure 11.

**Remarque :** *On peut aussi créer une base de données via les modèles déjà conçus par Microsoft.*

*Pour cela dans «Catégories de modèles» cliquez sur «Mes modèles». Puis dans «Modèles en ligne» choisissez le modèle qui vous convient. ◇*

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

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

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

**Les Formulaires :** Le formulaire est utilisé pour faciliter la saisie et la modification



FIGURE 11 – Création de la base de données Banque

des données d'une table, par exemple, le formulaire «client» va permettre d'entrer de façon conviviale les informations concernant un client de la banque. On peut les saisir sans formulaire, ce que nous verrons, mais cette méthode est moins conviviale.

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

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

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

Toutes les objets composants la base (cf Figure 12) seront développés dans les chapitres qui suivent.

## 2.2 LES TABLES

Dans cette partie, nous présentons la méthode de création de table. Access propose cinq (5) méthodes pour créer une table :

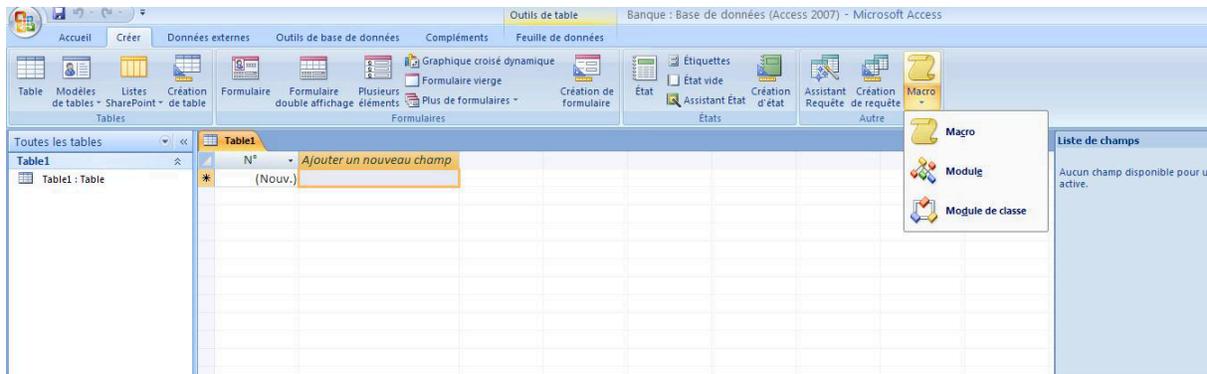


FIGURE 12 – Les différents objets de la base

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

✓ Nous utiliserons ici le **mode création**.

La création de la table s'effectue en remplissant les trois colonnes que sont :

- Nom du champ,
- Type de données,
- et Description.

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

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

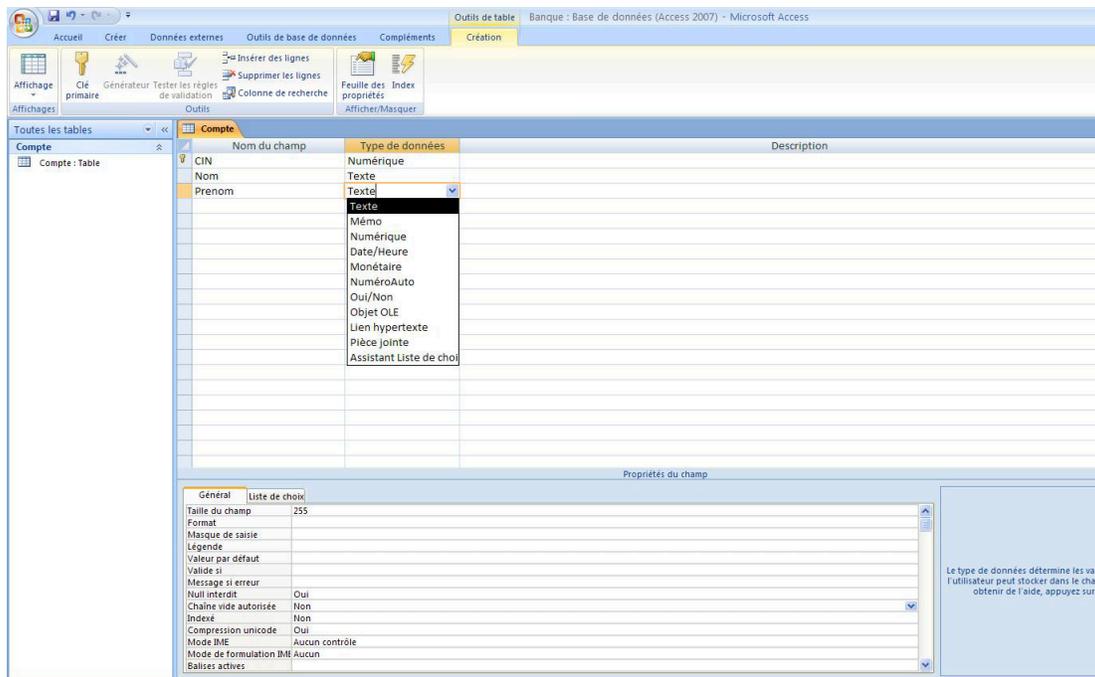


FIGURE 13 – Les différents types de données

— La description du champ (255 caractères maximum)

### 2.2.1 Les types de données

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

Il faut choisir, pour chaque champ de la table, le type de données le plus approprié.

En effet, rien n'interdit de choisir comme type de données pour un champ contenant une date ou un code postal un type «texte» ou pour champ contenant un solde un type «numérique», mais autant utiliser les types les plus appropriés ! Les différents types sont (cf Figure 13) :

**Texte** Pour un champ destiné à contenir des caractères alphanumériques (lettres et/ou chiffres). Attention, le champ de type texte ne peut contenir plus de 255 caractères, on l'utilise pour un nom, une adresse, une agence, etc.

**Mémo** Même utilisation que le type texte, mais le champ de type mémo peut contenir jusqu'à 65535 caractères (64 Ko). Utilisé pour une description longue par exemple.

**Numérique** Le champ ne pourra contenir que des nombres avec ou sans décimales.

(un code postal, un CIN, un solde, etc.).

**Date/Heure** Le champ ne pourra contenir que des dates ou des heures (Access vérifie la validité des dates ou des heures saisies).

**Monétaire** Valeurs présentées sous format monétaire (exemple : 4500.500 FCFA).

**NuméroAuto** Valeur numérique incrémentée automatiquement lors de la saisie de chaque enregistrement. Si vous choisissez ce type de champ, vous ne pourrez rien saisir dedans, c'est Access qui y placera un nombre automatiquement à chaque fois que vous créerez un nouvel enregistrement. On l'utilisera lorsqu'on veut être sûr que le contenu de ce champ ne se retrouvera pas dans un autre enregistrement de la table.

Par exemple, pour le numéro de client : chaque nouveau client de la banque aura un numéro unique automatiquement affecté par Access, c'est un champ de ce type qui sera le plus souvent utilisé pour être une clef primaire.

**Oui/Non** Seules deux données sont autorisées dans ce champ : Oui et Non (on utilisera ce type de données par exemple avec un champ «masculin» qui indiquera si un client est de sexe masculin ou non).

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

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

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

### 2.2.2 Les propriétés des champs

Comme vous pouvez le constater, chaque champ a des propriétés qui dépendent de son type de données, ces propriétés vont déterminer la façon dont les données qu'il contient vont être affichées ou comment les données vont être saisies dans ce champ.

Ces propriétés sont affichées dans l'onglet «Général» en bas de la fenêtre.

L'onglet «Général» affiche chaque propriété du champ et, à côté, dans des champs modifiables, la valeur de cette propriété.

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

FIGURE 14 – Propriétés des champs

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

La figure 14 indique les différentes propriétés des champs de type de donnée Texte et Mémo.

✓ Cependant, tous les autres types de données ont des propriétés similaires, certains ont des propriétés supplémentaires (cf Figures 15 et 16) .

### 2.2.3 Le format d'affichage

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

Propriétés des champs de type numérique	
<b>Taille du champ</b>	<p>Définit la valeur maximale qui va pouvoir être contenue dans ce champ. Il faudra faire attention et bien penser à l'avance quelle sera la valeur maximale permise dans ce champ pour pouvoir ajuster au mieux cette propriété :</p> <p><i>Octet</i> : de 0 à 255  <i>Entier</i> : de -32767 à +32768  <i>Long</i> : de -2147483648 à +2147483649  <i>Réel Simple</i> : de <math>-3.48^{E38}</math> à <math>+3.40^{E38}</math>  <i>Réel Double</i> : de <math>-1.79^{E308}</math> à <math>+4.96^{E324}</math></p> <p>Plus on descend, plus le champ occupera de la place en mémoire et sur le disque dur (le Réel Double prend 8 fois plus de place que l'octet), pensez-y si la table contient des milliers de lignes...</p>
<b>Décimales</b>	Nombres de chiffres après la virgule

FIGURE 15 – Champs de type Numérique

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

FIGURE 16 – Champs de type NuméroAuto

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

✓ **Attention !**

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

Pour modifier le format d'affichage, on utilise les codes suivants (quand ils marchent, etc.), Figure 15.

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

FIGURE 17 – Les codes du format d'affichage

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

FIGURE 18 – Pour les champs texte et mémo

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

FIGURE 19 – Pour les champs Numériques et Monétaires

✓ Toutefois, il y a des formats d'affichages plus spécifiques à certains types de données texte/mémo, Numériques/Monétaires et Date-Heure ; voir les Figures 18, 19 et 20.

✓ Pour les champs de type OUI/NON :

Access propose pour ce type de champ, trois types d'affichage par défaut : à la place de OUI ou NON, on peut afficher «Vrai» ou «Faut» ou alors «Actif» ou «Inactif».

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

FIGURE 20 – Pour les champs de type Date/Heure

Si on veut afficher un autre message, il faut dans le champ format de la propriété du champ taper la commande suivante : ;«texte 1» ;«texte 2»

alors, «texte 1» sera affiché à la place de Oui et «texte 2» à la place de Non.

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

;«OK»[vert] ;«NON»[Rouge]

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

**Remarque 2.1.** *N'oubliez pas que la ligne commence par le symbole «;».*

**Remarque 2.2.** *On peut définir son propre format d'affichage pour les champs numériques ou monétaires si le format que l'on désire ne fait pas partie des formats proposés ci-dessus. Pour cela, on utilise un format d'affichage spécial, ce format est défini par une chaîne de caractères qui a le format illustré dans la Figure 21 (vous tapez cette chaîne de*

; 
  ; 
  ; 
   
 >0      <0      =0      NULL

FIGURE 21 – Format d’affichage spécial

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

FIGURE 22 – Combinaison des symboles

*caractères dans le champ `format` de la propriété du champ).*

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

Chaque partie est une combinaison des symboles décrits dans la Figure 22.

Par exemple :

`$#,##0.00[vert];($#,##0.00)[rouge];«Zéro»;«Vide»`

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

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

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

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

FIGURE 23 – Combinaison des autres symboles

Comme pour les champs de type numériques ou date, on peut définir son propre format d'affichage en utilisant une combinaison des symboles illustrés dans les différentes Figures 23, 24 puis 25.

#### ✓ Format d'affichage des Dates/Heures dans Windows :

Pour définir le format d'affichage par défaut dans Windows, allez dans le panneau de configuration et double cliquez sur l'icône «Options régionales et linguistiques». Dans Options régionales cliquez sur Personnaliser puis sur Date (cf Figure 26).

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

#### 2.2.4 Le masque de saisie

Le masque de saisie permet de faciliter la saisie dans un champ en forçant l'utilisateur à entrer les données selon un format déterminé.

mm	Mois de l'année sur 2 chiffres (01 à 12)
mmm	Trois premières lettres du mois (jan à déc)
mmmm	Nom entier du mois (janvier à décembre)
t	Trimestre (1 à 4)
A	Jour de l'année (1 à 366)
aa	Deniers chiffres de l'année (00 à 99)
aaaa	Année complète (0100 à 9999)
h	Heure (0 à 23)
hh	Heure sur 2 chiffres (00 à 23)
n	Minutes (0 à 59)
nn	Minutes sur 2 chiffres (00 à 59)
s	Secondes (1 à 59)
ss	Secondes sur 2 chiffres (00 à 59)

FIGURE 24 – Combinaison des autres symboles(suite)

Tttt	Identique au format "Heure, Complet"
AM/PM	Heure sur 12 heures + AM ou PM
am/pm	Heure sur 12 heures + am ou pm
A/P	Heure sur 12 heures + A ou P
a/m	Heure sur 12 heures + a ou p
AMPM	Heure sur 12 heures + indicateur matin/après-midi défini dans le panneau de configuration

FIGURE 25 – Combinaison des autres symboles(suite)

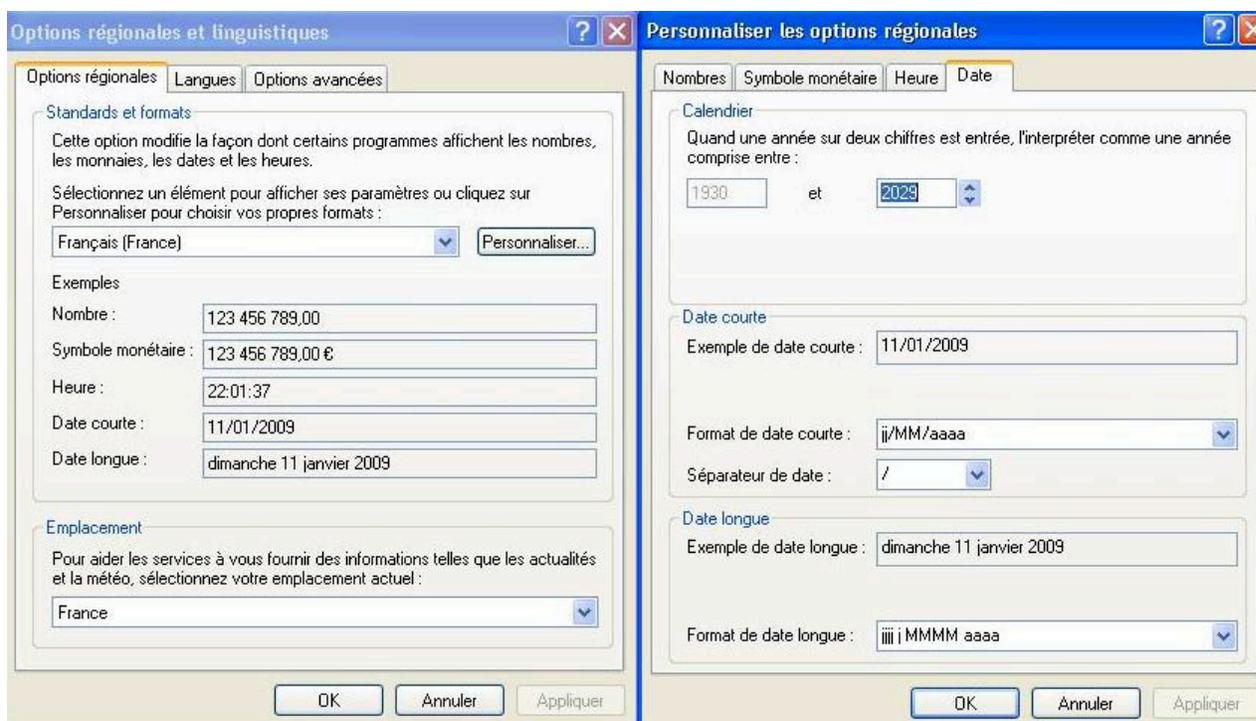


FIGURE 26 – Format d’affichage des dates/heures dans Windows

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

FIGURE 27 – Combinaison des caractères

Les informations stockées dans la table auront le format défini dans le masque de saisie contrairement au format d’affichage qui affectait seulement la façon dont les données contenues dans un champ allaient être affichées.

Le masque de saisie, comme le format d’affichage est constitué d’une combinaison de caractères (cf Figure 27) :

Par exemple :

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

### 2.2.5 Les listes de choix

Nous avons vu précédemment que parmi les types de données est proposé un type de données particulier : la liste de choix.

La liste de choix n'est pas à proprement parler un type de données particulier, c'est un moyen de simplifier la saisie de données dans une table en proposant à l'utilisateur de cliquer sur un élément proposé dans une liste.

Le champ de la table sera rempli avec l'élément sélectionné dans la liste.

### 2.2.6 Exemple

Nous considérons une base de données M1FASEG contenant une table Etudiants avec les champs "NoCarte", "Nom", "Prenom", "Groupe" et "Option". Nous allons créer la base de données ainsi que la table dans Access 2013.

Tout d'abord création de la base de données. On ouvre access et on click sur base de données vide. Une boîte s'affiche pour vous permettre de mettre le nom de la base de données. Mettre le nom "M1FASEG". Vous pouvez changer le répertoire où sera stocker la base de données. Une table par défaut est crée par access. Vous pouvez la renommer en "Etudiants" puis modifier la structure des champs. Vous pouvez aussi la fermer sans l'enregistrer ce qui revient à la supprimer pour ensuite créer une table par vous-même. Si l'on choisit de fermer la table par défaut on doit en créer un autre soi-même. Pour ce faire on va dans le menu "CREER" puis on clique sur "Table" ou "Création table" dans la barre d'outils "Tables". Un fois la table crée et nommée on l'ouvre en mode création. Dans le mode création on met les noms des champs dans la colonne "Nom du champ". On met les types correspondant à chaque champ dans la colonne "Types de données". On modifie les propriétés du champ dans la partie "Propriétés du champ".

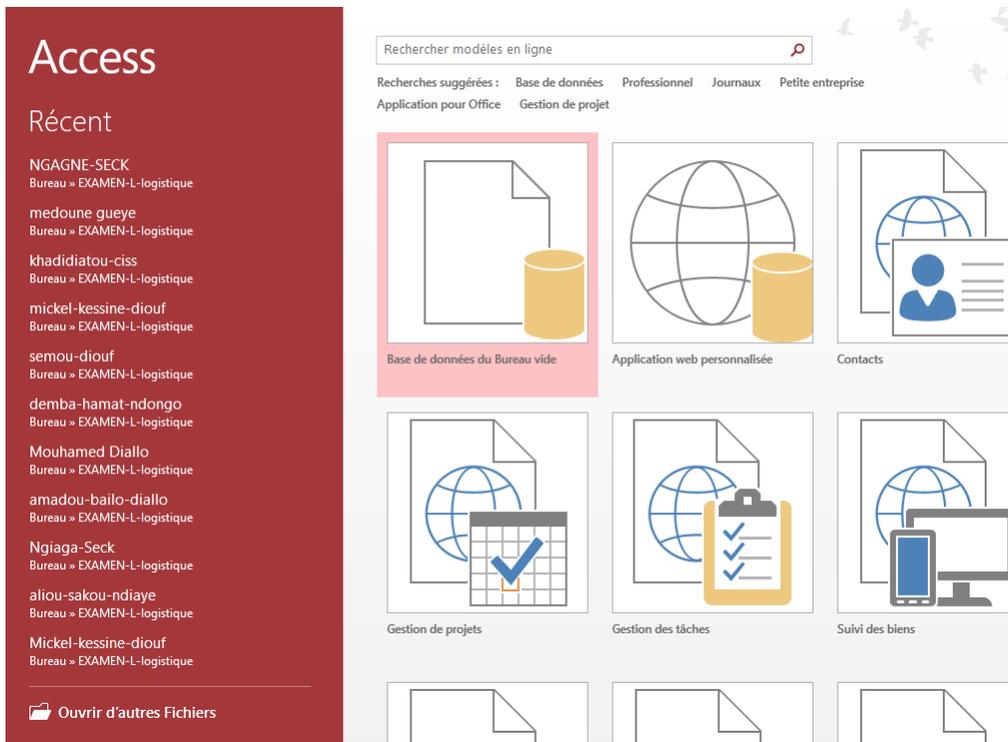


FIGURE 28 – Création de base de données vide

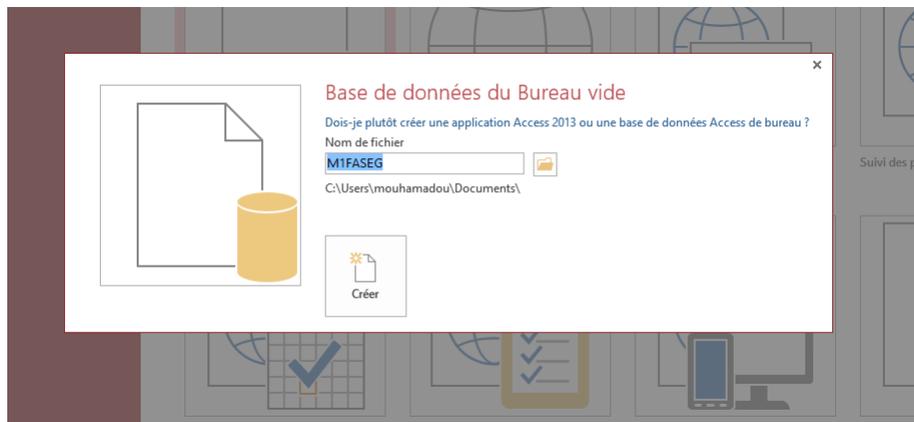


FIGURE 29 – Mettre le nom de base de données

## 2.3 RELATIONS

On distingue trois types de relation : un à plusieurs, plusieurs à plusieurs et un à un.

### 2.3.1 La relation de type « un à plusieurs » (1 - n)

Un enregistrement d'une table est en relation avec plusieurs enregistrements d'une autre table (Figure 38).

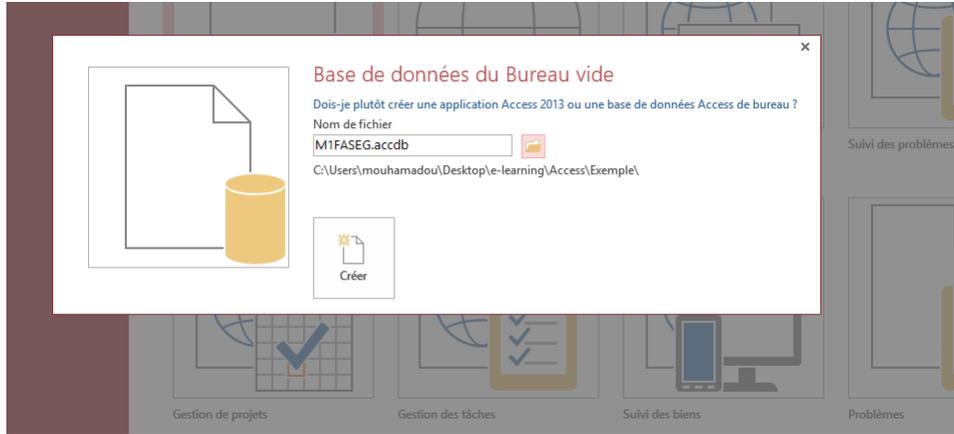


FIGURE 30 – Changer le répertoire de la base de données

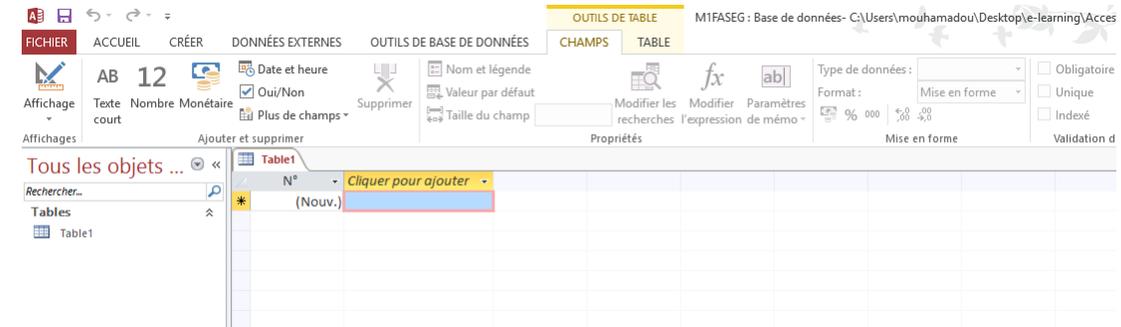


FIGURE 31 – Table par défaut

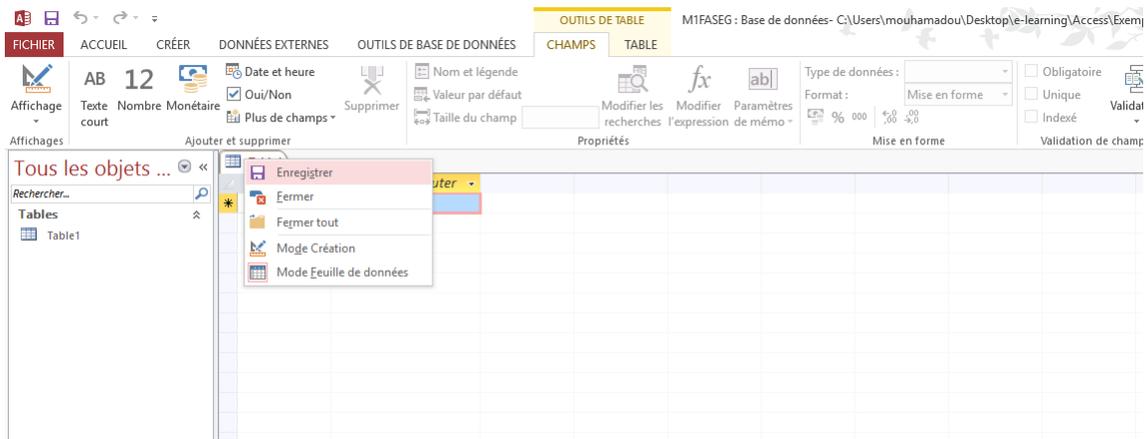


FIGURE 32 – Renommer la table par défaut

**Exemple 2.1.** *A un client correspond un ou plusieurs numéros de compte (plusieurs flèches partent du même client). Et à chaque numéro de compte ne correspond qu'un seul client.*

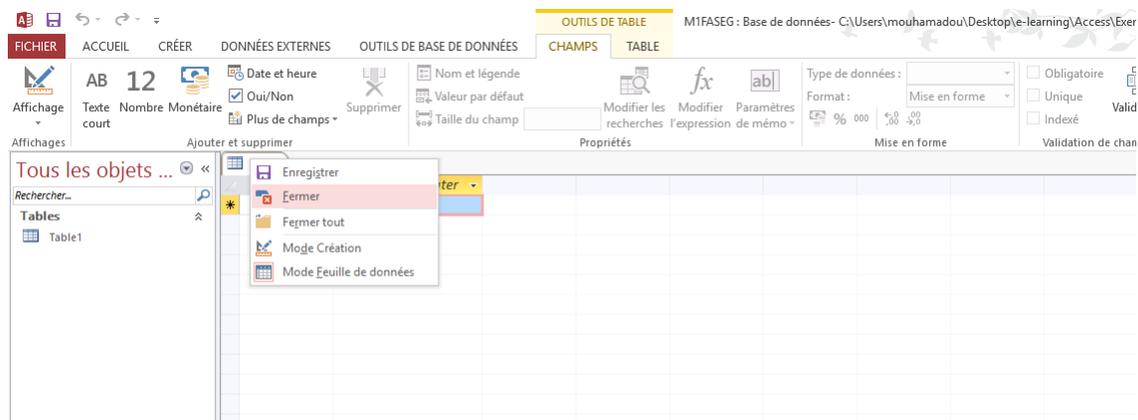


FIGURE 33 – Fermer la table par défaut

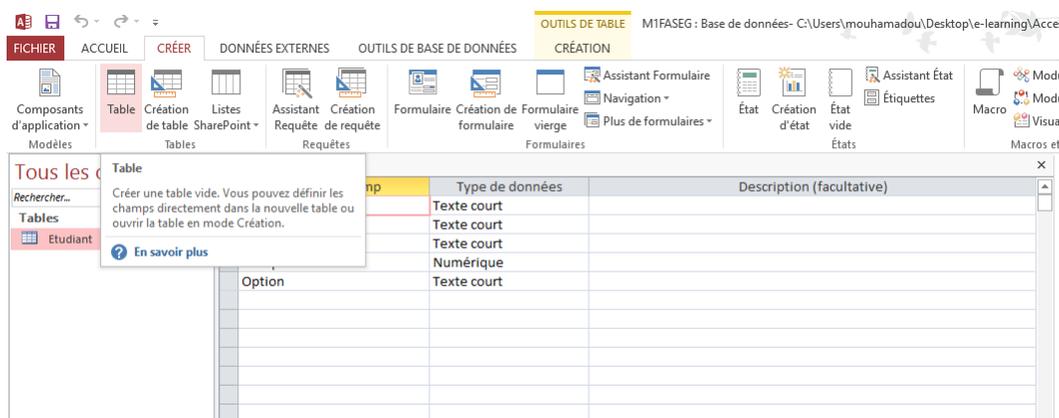


FIGURE 34 – Créer une table

La relation entre les deux tables se présente sous la forme de la Figure 39 :

### 2.3.2 La relation de type « plusieurs à plusieurs » (n - m)

Un enregistrement de la table primaire peut être en relation avec plusieurs enregistrements de la table reliée et inversement, un enregistrement de la table reliée peut être en relation avec plusieurs enregistrements de la table primaire.

Pour gérer ce type de relation, il faut la scinder en deux relations : une relation «un - plusieurs» et une relation «plusieurs - un».

**Exemple 2.2.** *On aurait pu avoir le cas où un numéro de compte est associé à plusieurs clients, là, ça ne marche plus, car telle que la table «Numero Compte» est structurée, à un numéro de compte ne peut correspondre qu'un client.*

Si on veut permettre qu'à un numéro de client correspondent plusieurs clients (on a alors

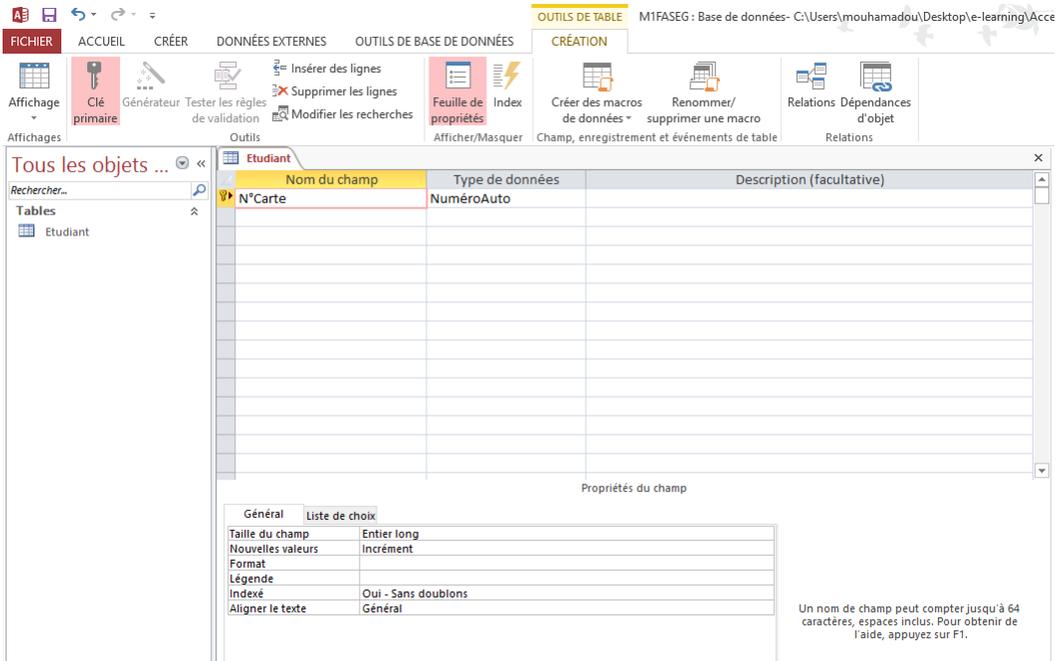


FIGURE 35 – Mettre les noms des champs

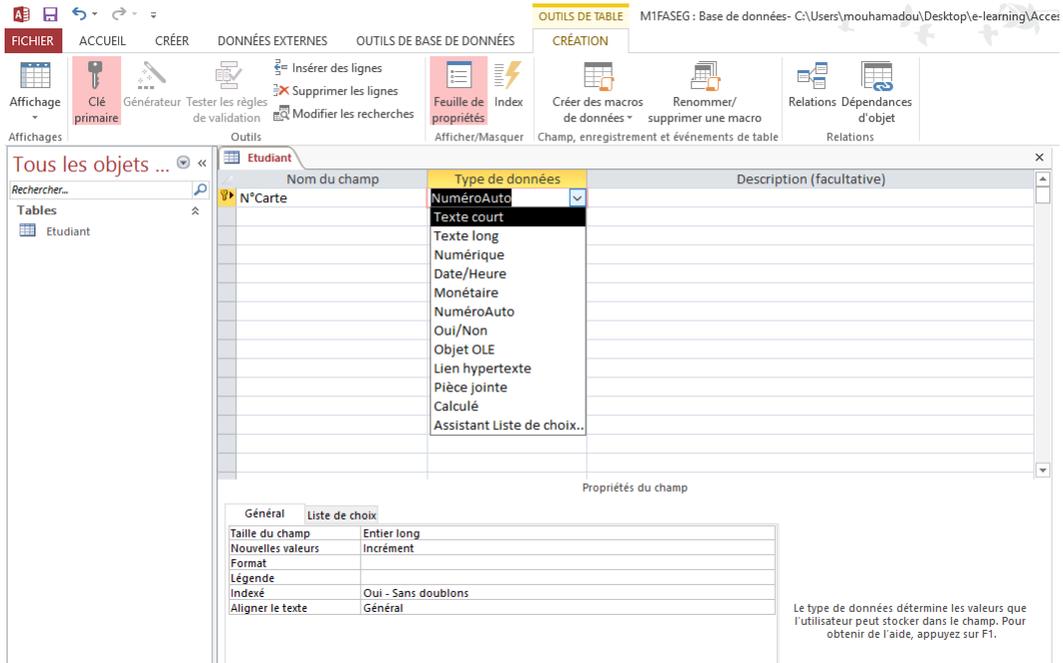


FIGURE 36 – Choix du type

une relation n - m), deux solutions s'offrent à nous :

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

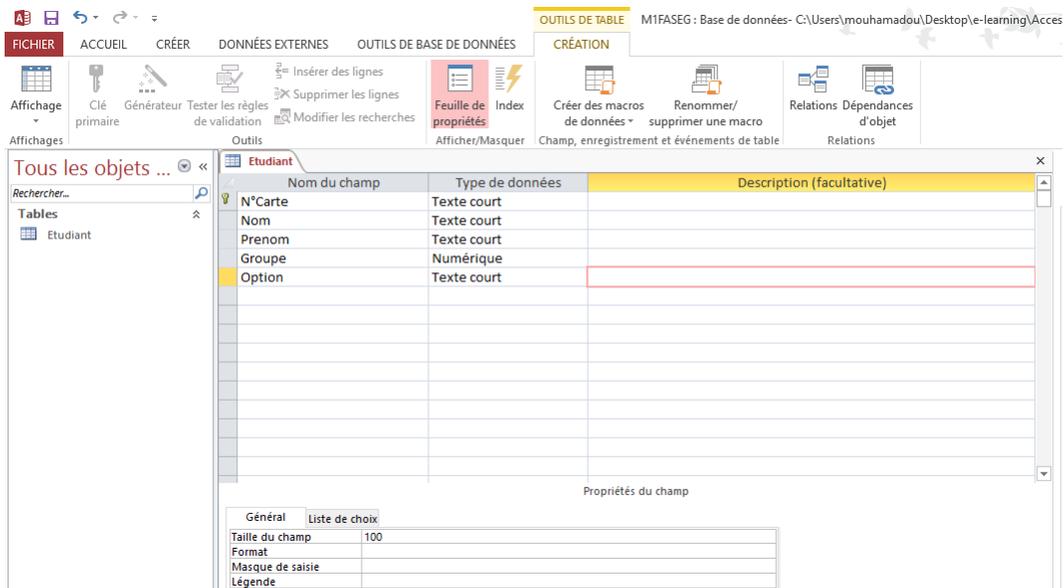


FIGURE 37 – Modifier les propriétés du champ

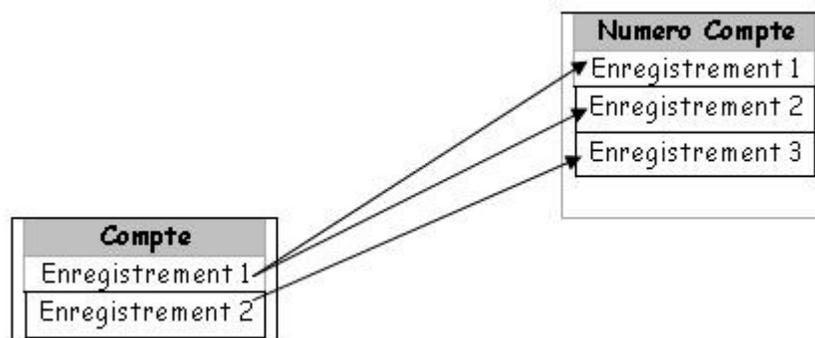


FIGURE 38 – Les types de relation

1 - n (Figure 40). Ce genre de plaisanterie est à éviter à tout prix, en effet, s'il n'y a qu'un seul client, les champs CIN 2 et CIN 3 ne vont rien contenir, et on va perdre de la place dans la base de données.

Que va-t-il se passer s'il y a quatre clients ?

Vous répondrez qu'on peut prévoir quatre champs, ce à quoi je vous répondrai c'est qu'il peut y avoir 5 clients, 10 clients, etc.

2. Il va donc falloir (on y coupe pas) créer une troisième table qui va servir d'intermédiaire entre la table «Compte» et la table «Numero Compte» (voir Figure 41)

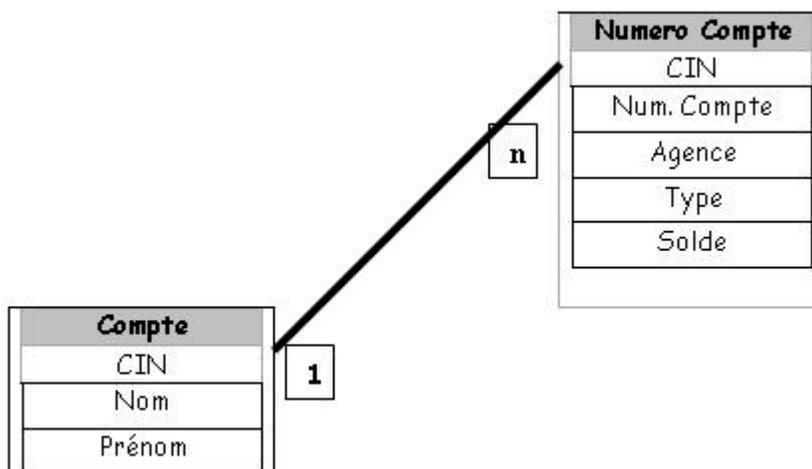


FIGURE 39 – Relations entre les 2 tables



FIGURE 40 – Relations plusieurs à plusieurs, 1<sup>er</sup> cas

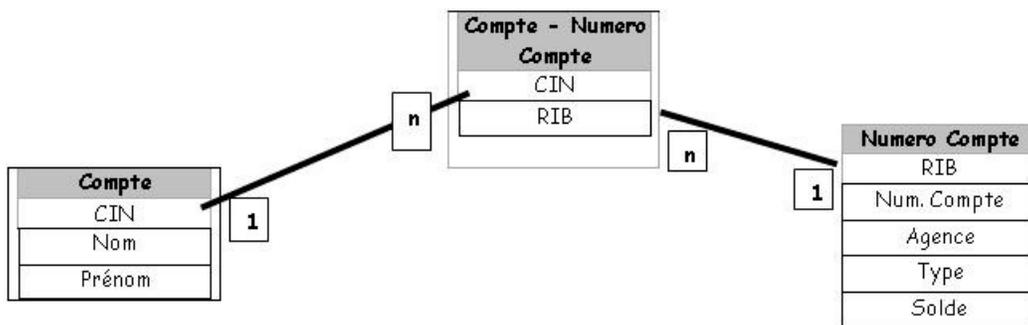


FIGURE 41 – Relations plusieurs à plusieurs, 2<sup>ieme</sup> cas

## Quelles sont les modifications ?

Nous sommes bien d'accord qu'avec un seul CIN du client dans la table «Numero Compte» nous ne pouvions pas avoir plusieurs clients pour le même numéro de compte. La solution 1 étant à proscrire (sauf dans le cas où on était sûr qu'il y aurait toujours 3 et seulement 3 clients), la troisième table est la seule solution.

### ✓ Comment ?

Et bien, à chaque numéro de compte correspond n enregistrements dans la nouvelle table «Compte-Numero Compte», chaque enregistrement renvoyant sur un seul enregistrement de la table «Compte».

Votre esprit curieux aura noté qu'on a, pour les besoins de la création de cette nouvelle table, créé une clef primaire dans la table «Numero Compte».

### ✓ Pourquoi ?

Pour faire correspondre un enregistrement de la table «Compte-Numero Compte» avec la table «Numero Compte», il faut le faire sur la clef primaire, et comme on n'avait pas de clef primaire dans «Numero Compte», il a bien fallu en créer une.

Aucun des champs présents ne pouvait être utilisé comme clef primaire (**la clef primaire, je le répète, doit identifier de façon UNIQUE un enregistrement**) : deux numéros de compte peuvent avoir la même solde (rare mais pas impossible), peuvent être créés le même mois, être du même type, etc., on aurait pu utiliser une clef primaire complexe avec un couple (solde + mois), plutôt rare que ce couple se retrouve deux fois, mais pour les raisons vues plus haut, on évitera d'utiliser ce type de clef.

Par exemple, on va avoir dans ces tables, les tables obtenues dans les figures 42, 43 et 44.

Avec cette méthode, on peut affecter sans aucun problème n clients à m numéros de compte (ici le client Taleb Fall a reçu deux RIB (Relevé d'Identité Bancaire), Alexandre Mendy un seul, et le RIB quatre est associé à deux clients que je ne citerai pas).

CIN	Nom	Prénom
1 785 81 02452	Fall	Taleb
2 785 79 51510	Mendy	Alexandre
2 002 45 44967	Diop	Mohamed
1 416 76 87500	Sagna	Véronique

FIGURE 42 – Compte

RIB	Num. Compte	Agence	Type	Solde (million FCFA)
1	00 910 875 142	Point E	Courant	8
2	00 910 158 784	Point E	Courant	4.5
3	00 910 158 783	Point E	Courant	25
4	00 410 100 850	Point E	Epargne	7

FIGURE 43 – Numero Compte

CIN	RIB
1 785 81 02452	2
1 785 81 02452	3
2 785 79 51510	1
2 002 45 44967	4
1 416 76 87500	4

FIGURE 44 – Compte-Numero Compte

### 2.3.3 La relation de type « un - un » (1 - 1)

Il existe un troisième type de relation, jamais utilisé : la relation de type « un - un » : un enregistrement d'une table est en relation avec un seul enregistrement d'une autre table et inversement.

Ce type de relation ne doit pas se produire car les données de la table reliée n'ont aucune raison de se trouver là, comme elles ne correspondent qu'à un seul enregistrement de la 1<sup>ère</sup> table, elles devraient être déplacées dans cette table.

### 2.3.4 Exemple

On suppose avoir deux tables "Etudiant"(NoCarte, Prenom, Nom, Groupe, Option) et "Validation"(Uv-Valider, NoCarte, Rang, mention). Mettre en relation ses deux tables sur le champ adéquat.

Pour faire la relation il nous faut choisir les deux champs des deux tables sur les quels on va faire le lien. On va faire le lien de sorte qu'à chaque étudiant de la table "Etudiant" on puisse voir les informations sur sa validation ou non. Il faudrait alors déjà que la table "Etudiant" ait une clef primaire. Etant donné que nous allons faire le lien sur le champ "NoCarte", on va le définir comme clef primaire au pour la table "Etudiant". Le champ "NoCarte" peut faire office de clef primaire puisque c'est un identifiant unique d'enregistrement. Pour la table "Validation" il n'est pas nécessaire de définir une clef primaire.

Pour commencer ouvrez access, si les deux tables sont déjà créées on passe à la suite sinon créez les comme indiqué dans le chapitre 2.2. Si les tables concernées par la relation sont ouverts, fermez les. Allez dans le menu "OUTILS DE BASE DE DONNEES" dans la barre d'outil "Relations" cliquez sur "Relations". Une boîte de dialogue "Afficher la table" s'affiche. Dans l'onglet des tables sélectionnez les deux tables "Etudiant" et "Validation". Puis cliquez sur "Ajouter" ensuite "Fermer". Au niveau de la table "Etudiant" sélectionnez et appuyez sur "NoCarte" faite glisser vers "NoCarte" de la table "Validation" et lâchez. Une boîte de dialogue "Modifier les relations" s'affiche. Tout en bas, Type de relation indique "Un-à-plusieurs". Il peut parfois indiquer "Non défini" si la relation est mal faite. Cochez toutes les options puis cliquer sur "Créer". Pour constater le résultat obtenu, ouvrez la table "Etudiant". Grâce à l relation, on peut donc voir dans la table "Etudiant" les UVs validés de chaque étudiant ainsi que le rang et la mention.

## 2.4 LES REQUÊTES

Les requêtes vont servir à afficher uniquement certaines données contenues dans les tables selon certains critères. Elles peuvent aussi faire des calculs sur vos données, ainsi que créer des sources de données pour les formulaires, les états ou même d'autres requêtes (on peut faire une requête sur le résultat d'une autre requête). Elles servent encore à modifier des tables existantes ou à en créer des nouvelles. Il existe différents types de requêtes que nous allons détailler après :

- **La requête sélection** : C'est celle qu'on utilisera le plus souvent. Elle permet de sélectionner des enregistrements, de faire des calculs et des regroupements. Elles ressemblent beaucoup aux filtres, mais permettent, en plus, de travailler sur plu-

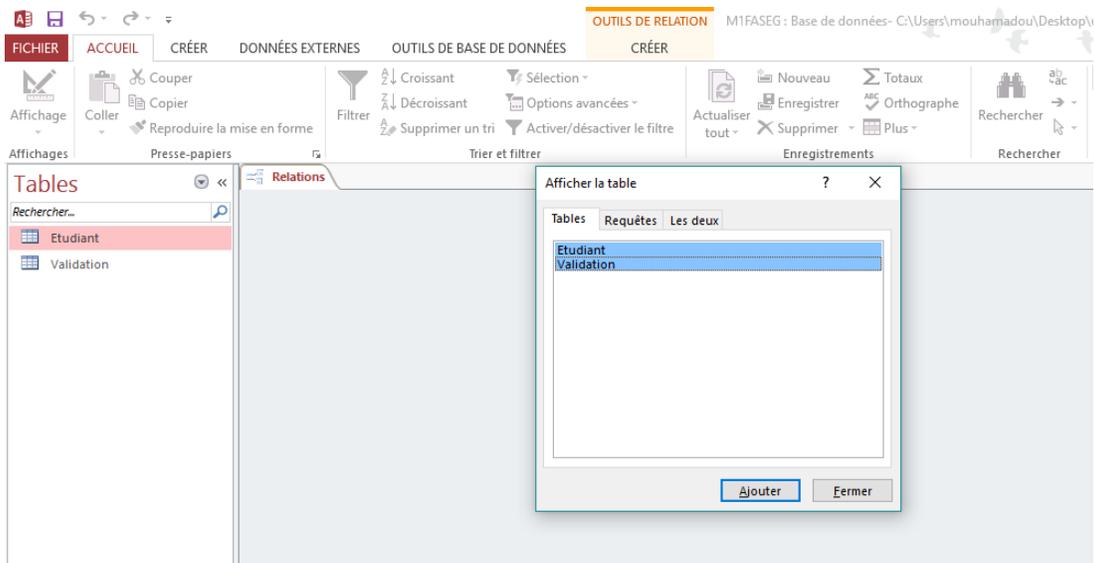


FIGURE 45 – Relation : afficher la table

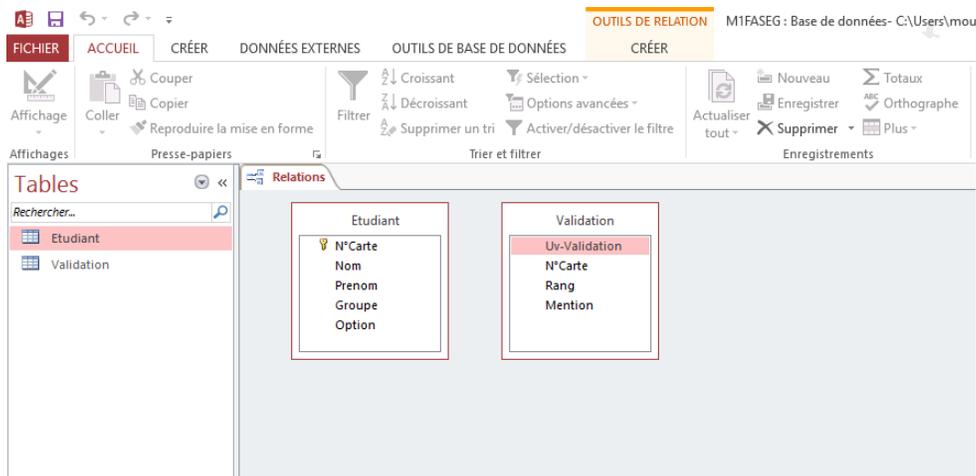


FIGURE 46 – Relation : ajout des deux tables

sièurs tables simultanément.

- **La requête d'Analyse croisée** : Cette requête présente ses résultats sous forme de tableau (de type Excel). On l'utilisera pour comparer des valeurs, dégager des tendances.
- **La requête de Création de table** : Cette requête crée une table à partir des données qu'elle a extraites dans une ou plusieurs autres tables.
- **La requête Mise à Jour** : Cette requête modifie le contenu d'un ou plusieurs champs d'une ou plusieurs tables. C'est le moyen le plus efficace pour mettre à jour un grand nombre d'enregistrements en une seule opération.

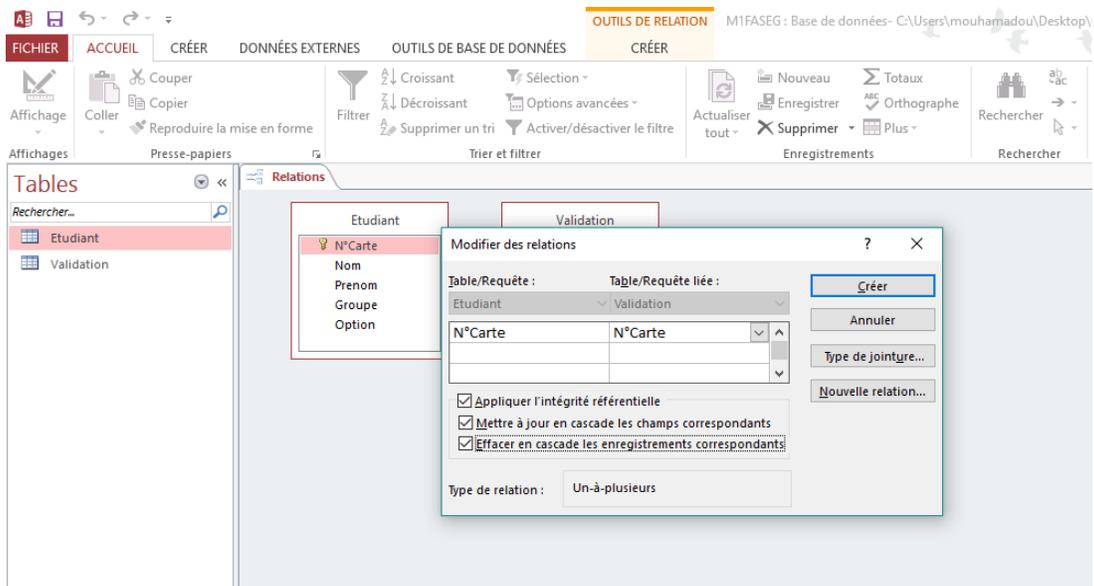


FIGURE 47 – Relation : Ouverture de la table de relation

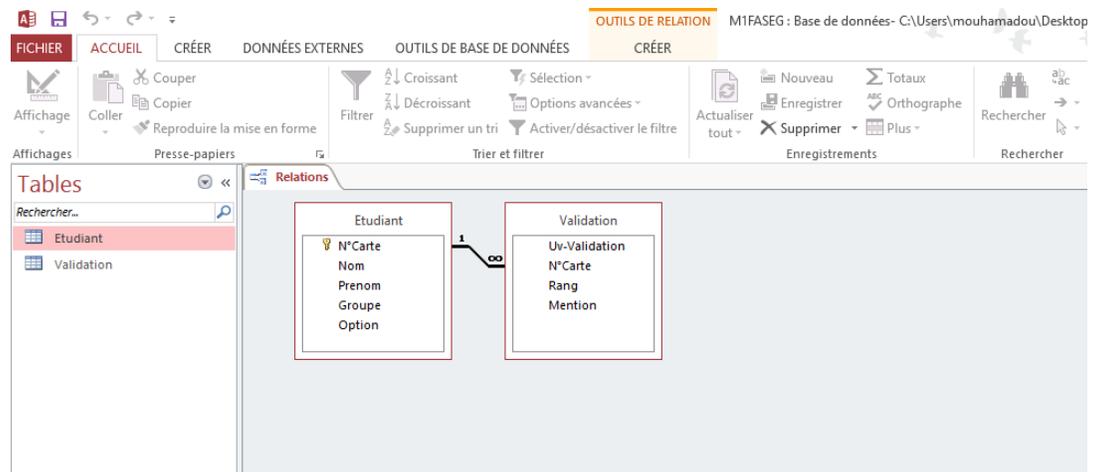


FIGURE 48 – Relation : relation effectuée entre les deux tables

- **La requête Ajout** : Cette requête ajoute les données qu'elle a extraites à la fin d'une table déjà existante.
- **La requête Suppression** : Cette requête supprime un ou plusieurs enregistrements dans une ou plusieurs tables. Les trois derniers types de requêtes ne seront pas étudiés :
- **La requête SQL direct**
- **La requête de définition de données**
- **La requête UNION**

Supposons qu'on veuille créer la requête suivante "afficher tous étudiants en Analyse".

Pour ce faire il suffit d'aller dans le menu "CREER" puis dans la barre d'outils "Requêtes" choisir "Assistant Requête" ou "Creation de requête". Nous allons utiliser l'assistant requête puisque une fois la requête créée on peut l'ouvrir en mode création ce qui reviendrait à l'option "Creation de requête" sans l'avantage d'avoir été assisté par access dans la création.

Une fois l'assistant ouvert vous choisissez "Assistant Requête simple" puis cliquez sur "OK". Ensuite au niveau de "Tables/Requêtes" vous devez choisir l'objet sur lequel porte la requête. Ici on choisit la table "Etudiant". Choisissez les champs concernés par la requête en sélectionnant le champ et en cliquant sur la flèche ">". Ici on choisit les champs "NoCarte", "Nom", "Prenom" et "Option". Ensuite cliquez sur suivant. Donnez à la requête un nom significatif par rapport à question qu'elle pose par exemple "EtudiantAnalyse", puis cliquez sur "Terminer". Un premier résultat de la requête s'affiche lorsque au moment de cliquer sur "Terminer" l'option "Ouvrir la requête pour afficher les informations" est active. Pour compléter la requête, c'est à dire afficher uniquement les étudiants en analyse, on l'ouvre en mode création. En mode création en haut on voit "OUTILS DE REQUÊTES" dans le menu "CREER" au niveau de la barre d'outil "Type de Requête" on voit que "Sélection" est active, cela signifie que la requête que nous avons créer est de type sélection. Il est possible de modifier le type de requête en cliquant sur un des types de requête : Création de table, Ajout, Mise à jour, Analyse croisée, Suppression etc. Pour achever notre requête on met "Analyse" à l'intersection de "Critères" et du champ "Option" puis on tape sur "Entrée". On exécute à présent la requête en cliquant sur "Exécuter" dans "OUTILS DE REQUÊTES" dans le menu "CREER" au niveau de la barre d'outil "Résultats".

## 2.5 LES FORMULAIRES

Le formulaire va nous permettre d'afficher et de modifier le contenu d'une table de façon bien plus agréable que le mode "feuille de données " qui ne permet qu'un affichage en lignes et colonnes. De plus, le mode "feuille de données " ne permet l'affichage et la modification d'informations ne provenant que d'une seule table, le formulaire va nous permettre de manipuler au même endroit des informations provenant de plusieurs tables simultanément : par exemple, dans notre exemple de base de données magasin, nous

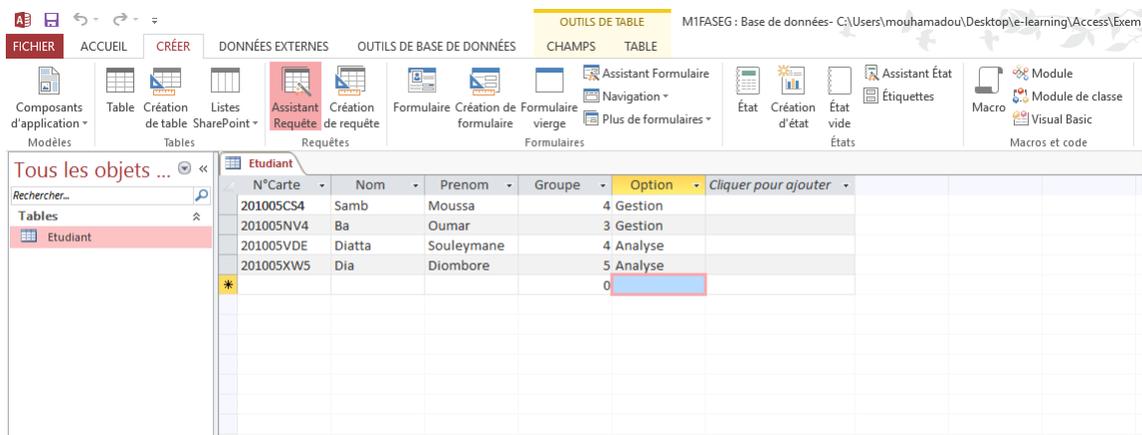


FIGURE 49 – Ouvrir Assistant Requête

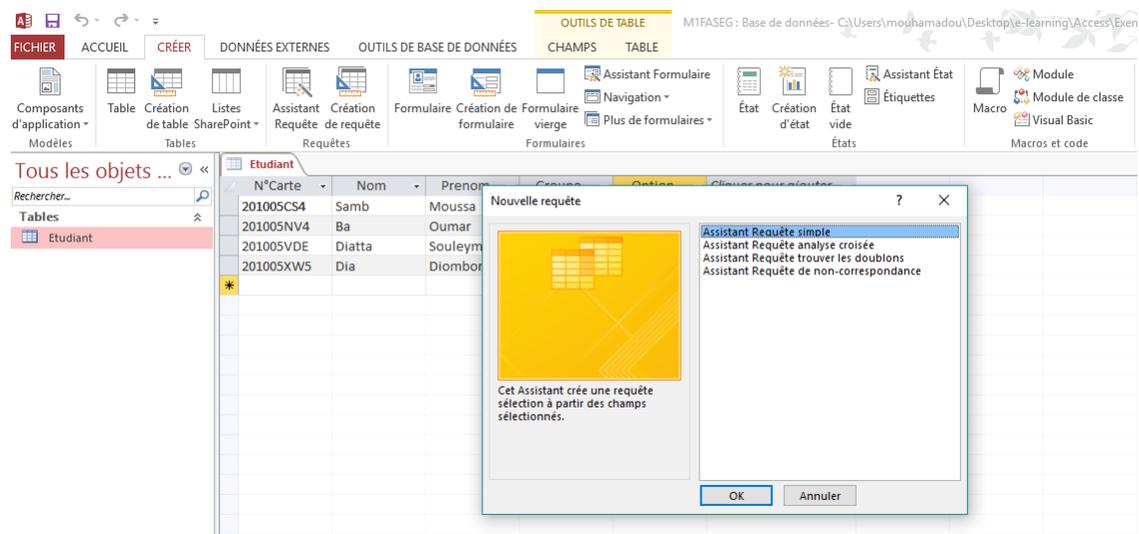


FIGURE 50 – Choix Assistant Requête simple

pourrons avoir un formulaire qui affichera dans la même fenêtre toutes les informations concernant une commande : informations générales sur la commande (provenant de la table “commande ”, informations sur le client ayant passé cette commande (provenant de la table “clients ” et le détail de cette commande (provenant des tables “lignes-commande ” et “produits ”) alors que précédemment, ces données étaient éclatées sur plusieurs feuilles de données. Les informations saisies ou modifiées dans le formulaire seront modifiées dans les tables à partir desquelles le formulaire a été créé.

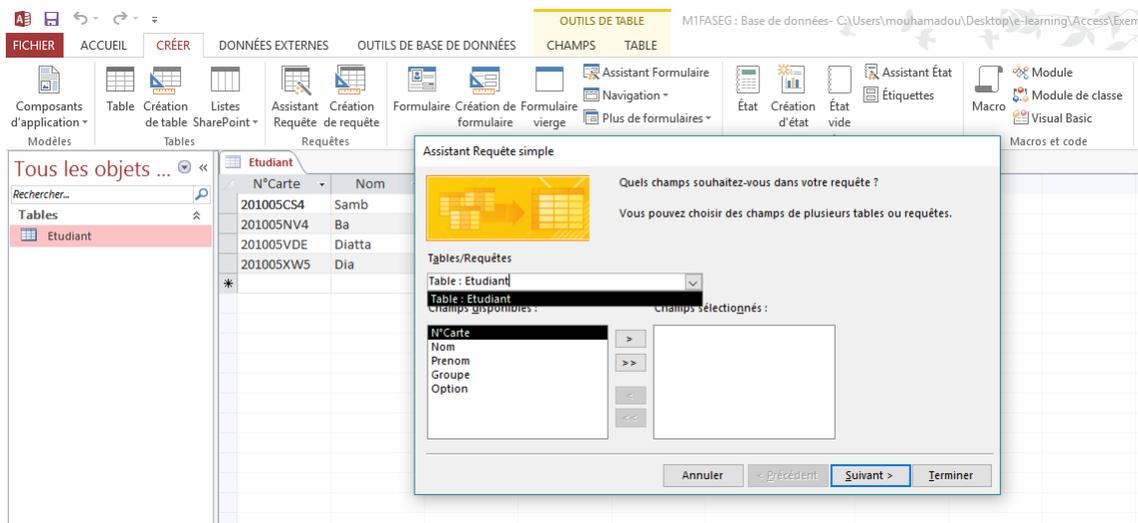


FIGURE 51 – Assistant Requête : choix objet

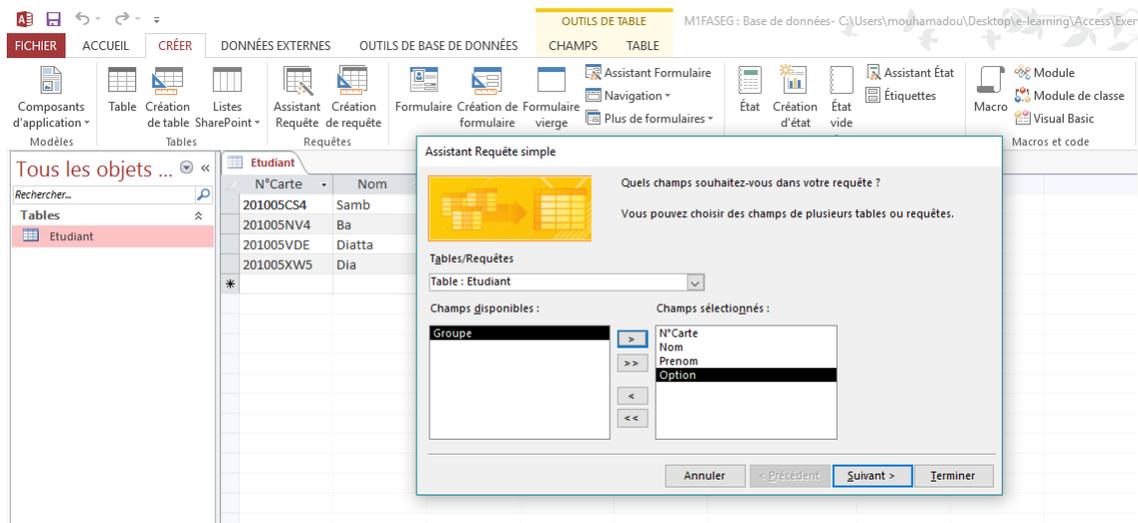


FIGURE 52 – Assistant Requête : choix champs

## 2.6 LES ÉTATS

Les états vont permettre l'impression d'enregistrements selon une présentation qui aura été définie préalablement. La création d'un état ressemble fortement à la création d'un formulaire.

## 2.7 LES MACROS

La macro commande permet d'automatiser certaines tâches d'Access, elle peut simuler une suite d'actions qui auraient du être faites par l'utilisateur. Une macro est composée

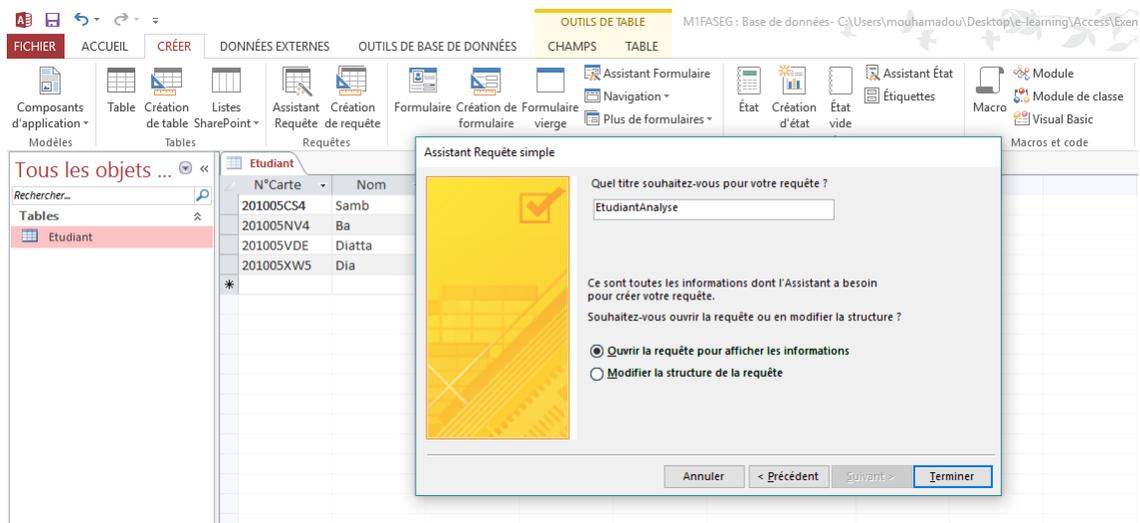


FIGURE 53 – Assistant Requête : renommer la requête

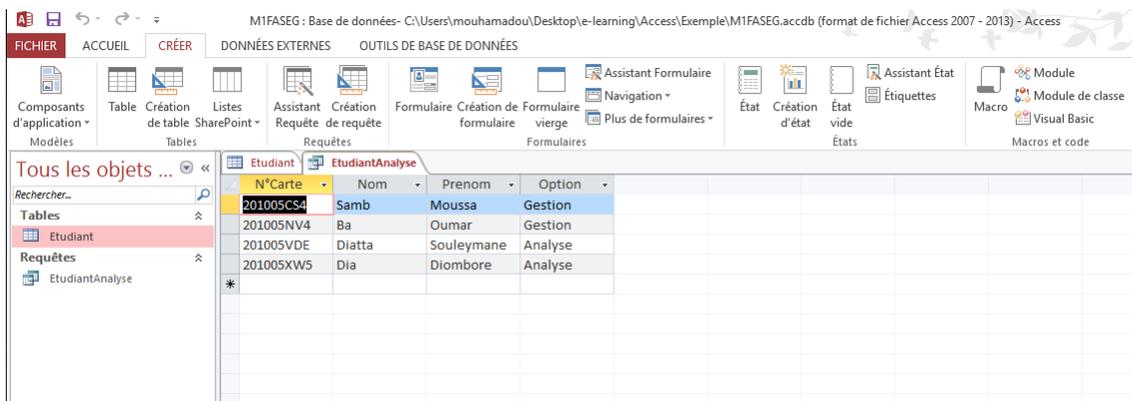


FIGURE 54 – Affichage de la requête

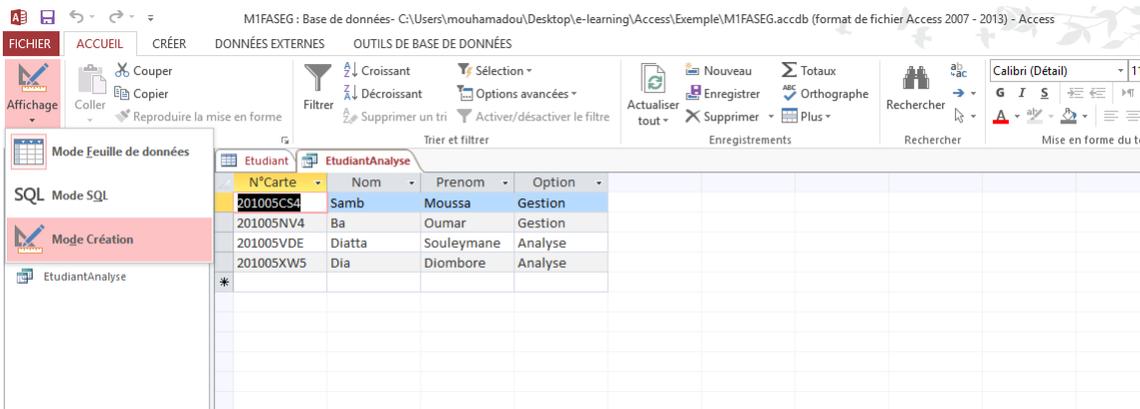


FIGURE 55 – Ouverture requête en mode création

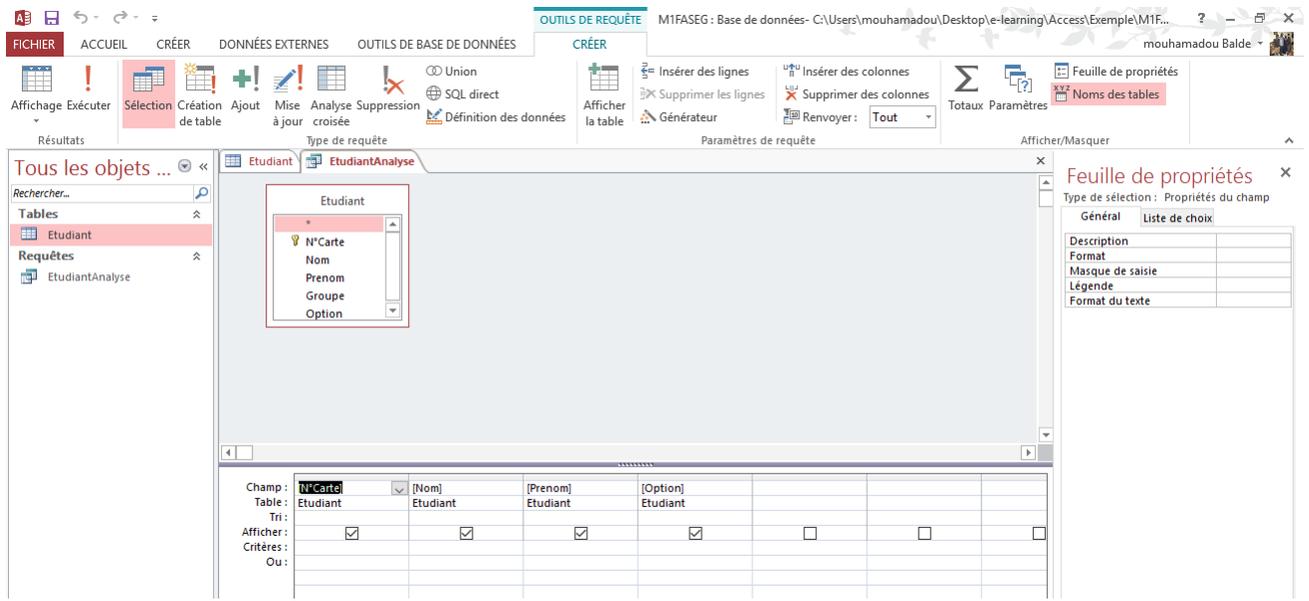


FIGURE 56 – Requête en mode création

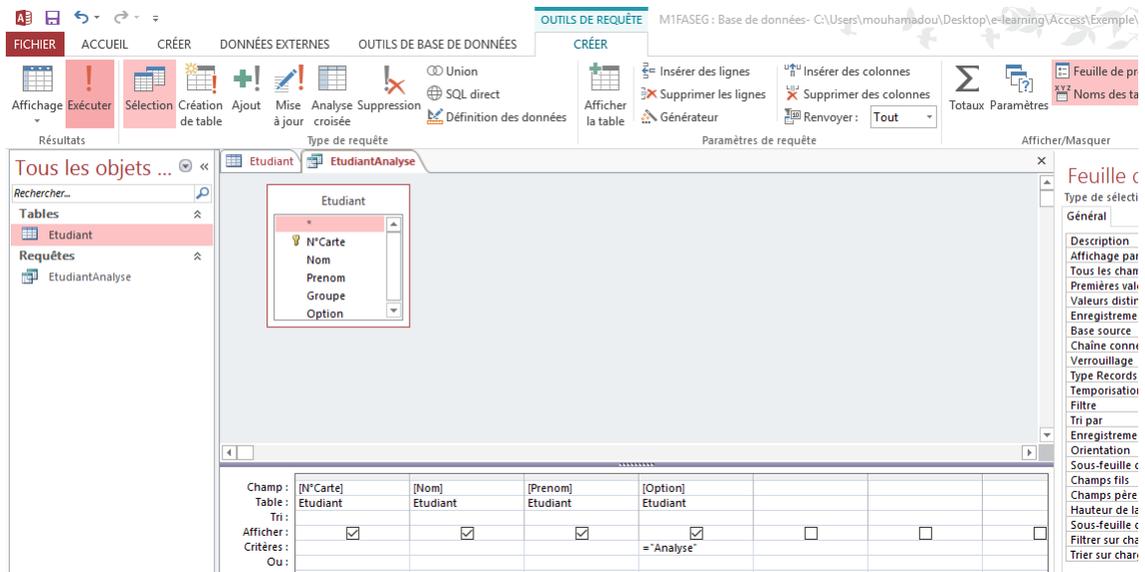


FIGURE 57 – Mettre le critère “Analyse” au niveau de “Option”

d’Actions, chaque action correspond à une tâche : lorsque vous exécutez la macro, Access exécute automatiquement les actions qu’elle contient. Certaines de ces actions, plus complexes, vous permettent d’afficher des boîtes de dialogue, de tester la réponse fournie par l’utilisateur, d’afficher une barre de menus personnalisée? et de développer une application autonome sans avoir besoin de programmer des modules en Visual Basic (bien que les actions fassent référence à des instructions en Visual Basic). La méthode de création

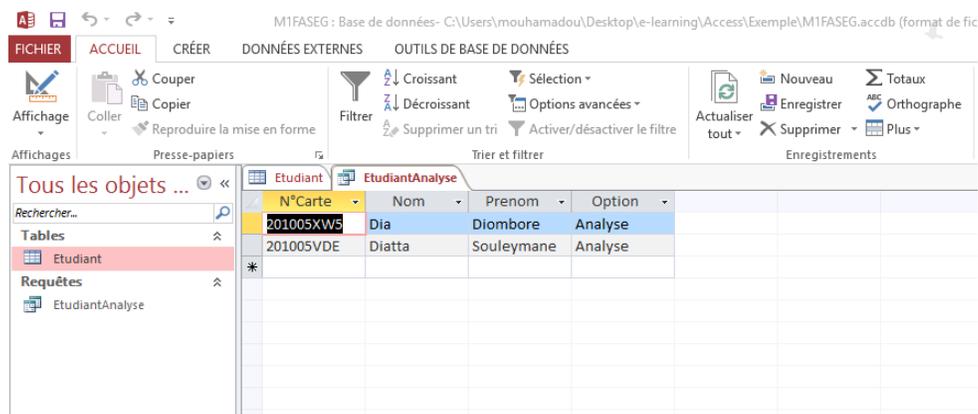


FIGURE 58 – Exécution de la requête

d'une macro est liée à deux facteurs importants :

- L'environnement de départ de la macro : certaines macros peuvent être exécutées quelle que soit la fenêtre active, d'autres sont liées à un objet de la base de données (formulaire, état, ?)
- L'événement qui va déclencher l'exécution de la macro : dans un formulaire, ce peut être un clic sur un bouton, ce peut être l'ouverture du formulaire, la valeur du contenu d'un contrôle, ou encore en fonction de la mise en page d'un état.

## Références

- [1] Microsoft Access Home Page, <https://products.office.com/fr-fr/access>.
- [2] Catégorie Microsoft Access du site apprentissage-virtuel, <http://www.apprentissage-virtuel.com/access>.
- [3] Philip J. Pratt and Joseph J. Adamski, Concepts of Database Management, Cengage Learning - 2007.
- [4] Saeed K. Rahimi and Frank S. Haug, Distributed Database Management Systems : A Practical Approach, John Wiley and Sons, 2010.
- [5] Club des développeurs et IT pro, <http://access.developpez.com/>