N°d'ordre: .....

République Algerienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la Recherche scientifique
Université Mouloud Mammeri de Tizi Ouzou
Faculté des Sciences
Département de Mathématiques

Mémoire de Master

Filière : Mathématiques
Spécialité : Recherche opérationnelle

Par

HATEB ASMA et MACHAINE OUIZA

Créer et gérer une base de données sur Ms Access

Soutenue le Sept 2022 devant le jury :

Dr.	GUETTAF Rabah	UMMTO	Président du jury
Md.	ZIDELMALE NACIRA	UMMTO	Examinateur
Dr.	AMIROU Ahmed	UMMTO	Encadreur

Année Universitaire : 2021/2022

Remerciements

Nous remercions dieu le tout puissant de nous avoir donné la force, la volonté et le courage pour accomplir ce travail.

Il nous est agréable d'adresser nos premiers remerciements à notre promoteur Monsieur AMIROU Ahmed ,qui a dirigé notre travail, pour ses précieux conseils et ses orientations ....

Nous adressons également nos plus profonds remerciements à nos enseignants.

Nos vifs remerciements s'adressent aux membres de jury qui ont accepté d'examiner notre travail.

Nous sincères sentiments vont à nos parents qui ont sacrifie jusqu'aujourd'hui et leurs encouragements tout le long de notre parcours.

Enfin, nous remercions toute personne ayant contribuée de près ou de loin à la réalisation de ce modeste travail.

Tizi-Ouzou, le 22 octobre 2022.

DEDICACE

Je dédie ce modeste traivail à :

Mes parents ;qui se sont sacrifié pour nous aves toute mon affection,mon amour,ma reconnaissance et tout mon respect.

"Que Dieu vous garde"

Mes soeurs mon frère pour leur affections et leur encouragemants qui ont toujours été pour moi des plus précieux.

Ma binome OUIZA et sa famile et tous ceux qui m'ont aidé de prés ou loin.

ASMA

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Une base de données est un ensemble recueil structuré d'informations.C'est ,en quelque sorte ,la version électronique des bacs dans lesquels sont rangés les fiches clients ,les descriptifs produits,les facteurs,etc.L'outil informatique apporte naturellement une plus grande souplesse dans la mise à jour des données et dans leur exploitation.
Avec ACCESS ,nous devenons concepteur de bases de données,et non plus simple utilisateur : à nous d'élaborer,de structure et de construire nos propres fiches de saisie.En centralisant l'information dans des containers à meme de l'accueillir ,nous pourrions ensuite l'exploiter à loisir. Objectivement,ACCESS est logiciel complexe ,qui s'apprivoise avec le temps.Inisan [15 juille 2007]

Concrètement on a procédé par partager notre travail en six chapitre ou avec chaque chapitre nous avons expliqué d'une manière détaillée comment créer et gérer une base de données en utilisant ACCESS , ces chapitre sont comme suit :

1. créer une base de données de bureau à l'aide d'un modèle et comment créer une base de données à partir de zéro en créant nos propres tables , formulaires , états et autres objets de base de données .
2. conception et création d'une base de données.
3. Interrogation d une base de données .
4. Expression de requête.
5. Conception de formulaires et d'états.
6. Partie pratique.

Dans cette dernière on base sur les précédents chapitres nous avons créé notre propre exemple qui est la facturation qui se compose de trois parties pratiques qui sont comme suivant :

Partie 1 : création des tables et des relations

Partie 2 : création d'un formulaire principal et de ses sous-formulaires.

Patrie 3 : création des états et de l'interface.

1 Création d'une base de donnée à l'aide d'un modèle et à partir d'une base de données vide

1.1 Introduction

Alors qu'au début de leur histoire les ordinateurs servaient essentiellement à calculer, leur utilisation principale de nos jours est la gestion d'informations. On les retrouve dans tous les secteurs d'activité .
Une grande quantité d'informations stockée dans un ordinateur s'appelle une base de données. Un logiciel permettant d'utiliser ces données est un système de gestion de base de données (SGBD). Différents logiciels existent permettant cette opération. Ainsi un tableur (tel qu'Excel) peut être considère comme un SGBD. Nous allons utiliser ici le logiciel Access comme SGBD.
Ce logiciel permet une conception aisée de bases de données de "petite" taille avec un nombre restreint d'utilisateurs. Il est à noter que plusieurs autres SGBD plus performants (mais également plus complexes) existent par ailleurs. On peut citer notamment Oracle, SQL Server, Paradoxe, MySQL, PostgreSQL parmi beaucoup d'autres. La plupart de ces systèmes sont bases (dont Access) sur le modèle relationnel et fonctionnent sur le même principes générale : les informations sont stockées dans des tables qui sont reliées entre elles par des relations. L'interrogation de la base de données se fait à l'aide de requêtes, ces requêtes étant écrites à l'aide d'un langage commun à la plupart des SGBD : le SQL (Structured Query Language).Lemay [2007]
De plus, il intègre un système de création d'applications claires et simples pour chaque base de données.
Ce chapitre est composé de :

— comment créer une base de données de bureau à l'aide d'un modèle .

— comment créer une base de données à partir d'une base de donnée vide en créant nos propres tables , formulaires , états et autres objets de base de données.

1.2 Quelques définitions liées aux Bases de données :

Définition 1.1 *Une base de données, usuellement abrégée en BD ou BDD, est un ensemble structuré et organisé d'informations. Les informations sont placées dans des fichiers, et organisées de manière à pouvoir être facilement triées, classées et modifiées par le biais d'un logiciel spécialisé appelé système de gestion de base de données (SGBD).Une base de données se traduit physiquement par un ensemble de fichiers présent sur une mémoire de masse (bien souvent un disque). La manière dont les informations sont organisées doit permettre de retrouver très rapidement n'importe quelle information sur un lot qui en contient plusieurs millions.doc*

Définition 1.2 *Le SGBD est un ensemble de services permettant de gérer les bases de données, c'est-à-dire il permet l'accès aux données de façon simple, autorise un accès aux informations à de multiples utilisateurs et manipule les données présentés dans la base de données (insertion, suppression, modification) .*
On peut dire que le SGBD est une interface entre les utilisateurs et la mémoire de masse, il facilite le travail des utilisateurs en leurs donnant l'impression que l'information est organisée comme ils le souhaitent.doc

Définition 1.3 *Un Système de Gestion de Base de Données peut être défini comme un ensemble de logiciels prenant en charge la structuration, le stockage, la mise à jour et la maintenance des données. Autrement dit, il permet de décrire, modifier, interroger et administrer les données. C'est en fait, l'interface entre la base de données et les utilisateurs. Un SGBDR permet de gérer une base de données relationnelle. Les SGBDRs peuvent être soit de type client/serveur (Oracle, SQL Server, MySQL, PostgreSQL, etc.), soit de type fichiers partagés (Access, SQL Server CE, Paradox, DBase, etc).doc*

Définition 1.4 *Microsoft Access souvent abrégé MS Access est un logiciel populaire de base de données application pour Windows , Access permet aux utilisateur de créer des bases de donnée personnalisées qui stockent des information dans une structure organisée. Le programme fournit également une interface visuelle pour la création de formulaires de tableaux et des SQL requêtes .Les données peuvent être entrées dans une base de données Access à l'aide de formulaire visuels ou d'un logiciel de base tableur interface .Les information stockées dans une base de données Access peuvent être consultées et accessible à partir d'autre programme . doc*

1.3 Avantages d'une base de données :

— **a)Centralisation** : les données peuvent être utilisées par plusieurs programmes et plusieurs utilisateurs.

b)-Indépendance entre données et programmmes :Dans une BD les données sont décrites indépendament des prpgrammes ce qui n'est pas le cas avec les fichiers.

C)-Intégration des liaisons entre les données : Pas besoin d'un programme pour retrouver les liens entre les données.

d)-Intégrité des données : Ce sont des régles de sécurité assurant la cohérance des données :
Unicité des enregistrements et interdiction de la suppression des données utilisées par d'autres données.

1.4 Ouvrire Accesss :

Ouvrir Access c'est tout simplement lancer une application Windows ,on suivant ces étapes :

- on clique sur le bouton Démarrer de windows ,choisissez successivemnt Tous les programmes puis Microsoft Office et enfin Microsoft Office Access 2007.
- Si on a utilisé Access il y a peu de temps on vas trouver son icône dans la partie gauche du menu Démarrer .
- OU bien par exemple on a un fichier de base de donnée ACCESS sur le bureau ou dans un dossier.faites double clique sur son icone .Access se lance et ouvre automatiquement ce fichier.

Access 2007 est capable d'ouvrir les fichier de base de données crées avec les anciennes versions de ce logiciel.il devrait supporter toutes les caractéristique de ces bases. Autrement dit ,notre base table formulaire et macros devraient s'ouvrir sans aucun souci.

1.5 Création d'une base de donnée à l'aide d'un modèle :

Si on veut organiser et gérer nos données avec Access ,mais ne veut simplement pas passer de temps à créer une base de données à partir d'une base de données vide .
La méthode la plus simple et la plus rapide pour créer notre base de données consiste à utiliser un modèle Access ; Access insérez une séries de modèles que nous pouvons utiliser comme un point de départ on n'essaye d'utiliser un modèle de base de données de bureau ou en trouver un en ligne .
Un modèle de base de données Access,tout comme un modèle word ou Excel, est un fichier à partir duquel on peut démarrer un nouveau projet, et qui une fois ouvert crée une application de base de données complète qui contient l'ensemble des tables,requêtes ,formulaire,macros et chaque modèle est conçu pour répondre à des besoins spécifiques de gestion de base de données , ils nous permettent de gagner du

temps et d'efforts et nous permettent de commencer à utiliser notre base de données immédiatement. Inisan [15 juille 2007]

Access fournit des modèles pour répondre aux besoins standards,comme l'illustre (la figure 1.1)

en mains.PNG

FIGURE 1.1 – *Les Modèles enligne*

On trouve dans le volet de gauche plusieurs catégories qui tentes de circonscrire les besoins de la plupart des utilisateurs.Chaque catégorie est associée à un ensemble de modèle qui viennent s'afficher dans le panneau central.Certains modèles peuvent se trouver sur notre disque dur,tandis que d'autre seront téléchargés sur le site Microsoft Office Online.
Si on souhaite utiliser un modèle installé sur notre ordinateur,on clique sur la catégorie Mes modèles visibles dans la zone Catégories des modèles . Le volet de gauche propose aussi une rubrique appelée A partir de Microsoft Office Online .Elle surmonte trois catégories : Professionnel,Personnel ,Éducation. Contact ,ventes,projets ,carnet d'adresses personnel....et la liste n'est pas exhaustive.
Lorsque on clique sur l'une des catégories Par exemples Microsofts office online ,le volet centrale change .Il montre le nom de la catégorie courante suivi d'une série de

grandes icônes qui correspondent à autant de modèles de base de données (voire la figure 1.2).

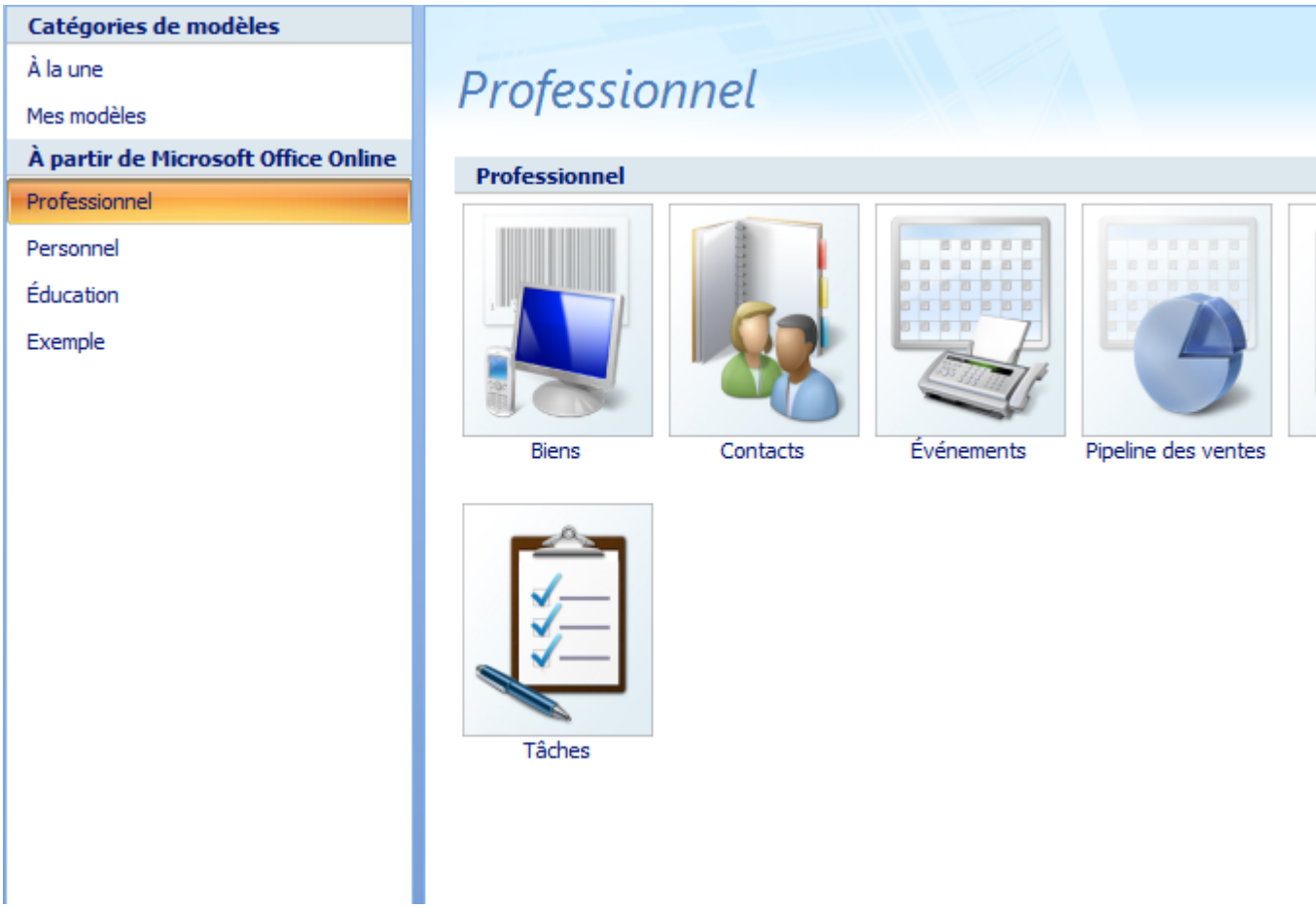


FIGURE 1.2 – *les modèles de base de données ,modèles professionnel*

Après avoir choisir le modèle de base de données que nous souhaitons d'utiliser ; la partie droite de la fenêtre affiche les option permettent de créer une nouvelle base de données basée sur le modèle sélectionné.
Dans le cas d'un modèle de Microsoft office Online on visualise la Taille de celui ci ainsi que l'évaluation, données dans la zone Nom de Fichier. L'emplacement du dossier dans lequel la nouvelle Bdd va être stockée et visible sous la zone Nom de fichier. on peut aussi cliquer sur le bouton ,puis on sélectionne le dossier dans lequel la BDD doit être stockée puis on clique sur OK. A la fin on clique sur le bouton Télécharger pour accéder au modèle.
Une fois qu'ACCESS à terminé de préparer le modèle ,le contenu de celui ci s'ouvre dans l'aaplication Acces .Un formulaire de la base de données s'affiche on permette ainsi de saisir des données .

On utilise ensuite les objets existant dans la base de données pour saisir ses données et créer éventuellement d'autre objets.La structure d'une base de données étant prédéfinie et on peut également modifier la structure des objets existant

1.6 Création d'une base de données :

Si on ne trouve pas le modèle dont on a besoin, on peut créer notre propre modèleen suivant ces étapes :
Access étant ouvert ,et le titre "Prise en main de Microsoft Office Access "étant affiché en lettres de feu sous la barre de titre.
(a) On clique sur le bouton Base de données vide (sous la bandeau Nouvelle base de données vide). Un volet apparait dans la parti droite de la fenêtre d'Access .il nous demande de définir le nom de notre base de données (voire la figure 1.3).

en mains.PNG

Figure 1.3 – *creation d'une base de données vide*

(b) On remplace la proposition par défaut **(Base de donnéesX.accdb)**par le nom que on veut attribuer à la base de données. Le X dans le nom de fichier ci-dessus

représente un nombre. Access numérote chaque nom par défaut selon un ordre croissant en fonction de nos décisions précédentes.

(c) Après on sélectionne l'emplacement dans lequel le fichier de base de données va être enregistré.Pour cela on suit ces étapes :

— On clique sur la petite icône qui représente un dossier ouvert ,ceci ouvre la boite de dialogue **Fichier Nouvelle base de données** .

— On choisis le dossier où on veut enregistré notre base de données après clique sur OK.
Le nom qu'on a attribué au fichier servira à ACCESS pour l'enregistrement du fichier dans son dossier de destination.

(d) Une fois revenu à la fenêtre d'Access ,on clique sur le bouton **Créer** Après la création de la base de données,on se retrouve par défaut sur **le Mode Feuille de données** c'est le mode qui permet de saisir des données.

A ce stade ,il est possible de foncer et saisir directement des données ou bien de commencer à définir les noms et les propriétés des champs.Les nom de ces champs apparaissent sur la ligne du haut,l'intitulé de ce en-tête indique pour l'instant **Ajouter un nouveau champ**.

on constate d'ailleurs qu'un champs **N** est automatiquement crée par Access. On peut sauvegarder tout de suite notre table , on cliquons sur l'onglet**Table1** et choisisse la commande **Enregistrer**.Access nous demandera de choisir une dénomination plus parlante et plus efficace que le Table1 par défaut.

Conception et création d'une base de données

2

2.1 Introduction

Une base de données correctement conçue nous donne accès à des informations précises et à jour. Étant donné qu'une conception correcte est essentielle pour atteindre nos objectifs en matière d'utilisation d'une base de données, il est tout à fait logique d'investir le temps nécessaire pour découvrir les principes de bonne conception. Au final, nous avons bien plus de chances de nous retrouver avec une base de données qui répond à nos besoins et qui peut facilement s'adapter au changement.

Dans ce chapitre ,on présente les conception d'une BDD et les étapes de la construction du modèle conceptuel de données (MCD) .Après on introduit le modèle logique et comment la transformer du modèle conceptuel en modèle logique ,et pour finir nous expliquons la création des tables au mode création et feuille de données.

2.2 Modèle conceptuel de données :

Le Modèle Conceptuel de Données (MCD) que nous allons construire contient deux éléments principaux :

— Les entités
— Les relations

Définition 2.1 *Une entité est un élément du problème. Elle est définie par un ensemble de propriété. Chacune des propriétés est l'un des éléments qui caractérise l'élément. Il faut distinguer une entité et une occurrence d'entité (ou instance). Une entité correspond au type général d'une donnée (ex : le type"employé) alors qu'une occurrence d'une entité est un représentant particulier de cette entité.Lemay [2007]*

Définition 2.2 *Une relation est un lien possible qui relie deux entités . Par exemple, si un employé peut être affecté à un entrepôt, il y aura une relation "affectation" entre l'entité entrepôt et l'entité "employé". Cela ne signifie pas nécessairement qu'il y aura affectation pour chacun des employé ,*

juste qu'il est possible qu'un employé soit affecté à un entrepôt. Une relation peut éventuellement être reliée à plus de deux entités et peut avoir certaines propriétés .Lemay [2007]

La construction du MCD se fait en-quatre étapes :

— Repérage des entités
— Construction des entités.
— Construction des relations
— Choix des cardinalités

2.3 Repérage des entités :

Une entité est un composant du problème : une personne, une facture, un livre, ... C'est la représentation d'un objet materiel ou immateriel pourvu d'une existence propre et conforme aux choix de gestion de l'entreprise .

Exemple : On considère le problème suivant : Un libraire gère des oeuvres litteraires. Une oeuvre une création littéraire .Une oeuvre a au moins un auteur et est dans une édition (un livre). Elle peut contenir plusieurs oeuvres. On veut mémoriser pour chaque édition le nombre d'exemplaires en stock et pour chaque exemplaire son état. Dans ce problème les entités sont :

— l'entité **"oeuvre"** :Une création littéraire,un récit
— l'entité **"auteur"** :une personne créateur d'oeuvre
— l'entité **"édition"** : un livre contenant une ou plusieurs livre littéraires.
— l'entité **"édition"** : la société qui vas imprimé les livres.
— l'entité **"exemplaire"** : un exemplaire physique de livre.

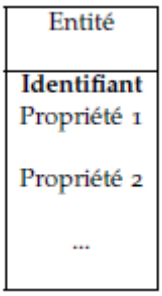
2.4 Construction des entités :

On commence par donner un nom à chacune des entités. Il faut ensuite rechercher les propriété de ces entités . On devra garder à l'esprit les points suivants :

— Toute propriété est élémentaire (elle n'est pas la composition d' éventuelles propriétés plus petites).
— Tout entités doit possédée une propriété particulière appelée clé ou identifiant. Une clé doit caractérisé de manière unique chaque occurrence de l'entité.Lemay [2007]
Si aucune des propriétés naturelles ne peut servir de clé ,on en rajoute une artificiellement(exp :"CodLivre" ou "IdAnimal")
— Chaque propriété ne doit dependre que d'une seule entité.

Une entité se représente ensuite graphiquement sous la forme d'un boite dans lequel on indique en titre le nom de l'entité suivi de toutes ses propriétés .On indique d'une

manière particulière l'identifiant.



Dans l'exemple du libraire, on peut construire les entités suivantes : (les propriétés sont indiquées après le nom de l'entité l'identifiant est en gras)

- Oeuvre : Idoeuvre, titre
- Auteur :IdAuteur, nom, prénom
- Editeure : IdEditeure, nom

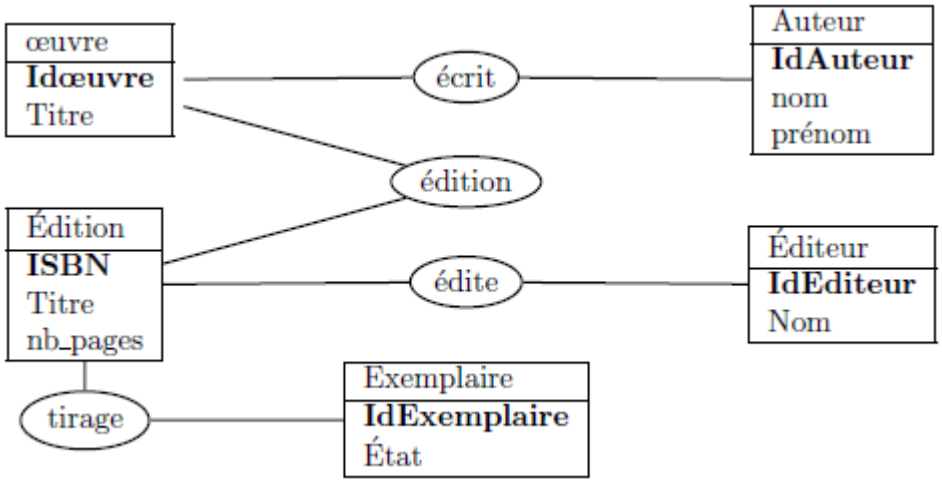
2.5 Construction des relations :

L'étape suivante consiste à énumérer toutes les relations possibles entre entités. Si une relation a une chance d'apparaître alors on doit la considérer dans le MCD. on parle parfois d'association .
Une relation se représente de la manière suivante : On notera les points suivant :

- Une relation est en général entre deux entités ;il est possible d'avoir des relations entre plus que deux entités. Par exemple,une relation vente entre Acheteur, Vendeur et Lieu pour une base de donnée de transaction immobilière.
- Il est tout à fait possible d'avoir plusieurs relations entre deux entités. Il est également possible d'avoir une relation dite réflexive, c'est-'a-dire entre une entité et elle-même. Par exemple, on peut avoir une relation**Responsable** entre une table **employé** et elle-même. Dans ce cas, il convient tout de même de remarquer que chacune des "pattes" de la relation a une signification différente. Ici, l'une des "pattes" signifiera est responsable de et l'autre signifiera a comme responsable. une relation peut avoir des propriétés. Par exemple, si une relation **Contient** lien l'entité **Facture** et l'entité **Produit**, elle possède certainement la propriété quantité (une facture contient un produit x en quantité y. D'ailleurs,

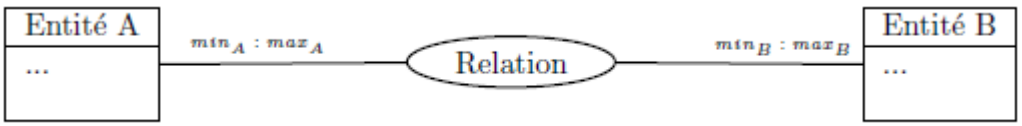
si une propriété dépend de plus d'une entité (comme c'est le cas ici avec la quantité qui dépend à la fois de la facture et du produit), c'est certainement qu'elle dépend d'une relation, et non pas d'une entité. Il faut éviter les relations que l'on peut déduire d'autres relations par transitivité. Par exemple dans une base de données gérant une université, si on dispose d'entités **étudiant** , **Formation** et **Cours** .
On a les relations **Fait partie** entre formation et cours (un cours fait partie d'une formation) et **inscription** entre étudiant et formation.
Il est inutile d'avoir en plus une relation **inscription** entre étudiant et cours tout étudiant inscrit 'a une formation est systématiquement inscrit à tous les cours qui composent la formation.

EXEMPLE :Dans l'exemple du libraire, on a les relations suivantes :



2.5.1 choix des cardinalités

Une fois les relations établies, il convient ensuite de caractériser le nombre de fois ou chacune de ces relations peut apparaître réellement. Ceci se fait à l'aide des cardinalités. Dans une relation classique (i.e. entre deux entités), quatre cardinalités sont à déterminer.



— minA est le nombre minimal de fois où une occurrence de l'entité A participe à une relation du type considéré. Il s'agit en général de 0 ou 1.

— maxA est le nombre maximal de fois o'u une occurrence de l'entité B participe 'a la relation.Il s'agit en général de 1 ou n (n pour plusieurs fois, ou un nombre quelconque de fois).
— minB et maxB fonctionnent de la même manière, mais en considérant l'entité B.

Notons qu'il est souvent difficile de choisir entre une cardinalité de type 0 : n et une cardinalité de type 1 : n. Il est important de noter que ce choix a souvent peu d'importance.

EXEMPLE :Dans l'exemple du libraire, considérons la relation **édite** qui existe entre les entités **éditeur** et **édition**. Ainsi, dans l'exemple du libraire, une édition (un livre) a toujours un et un seul éditeur (soit un minimum de un éditeur, et un maximum de un éditeur). Un éditeur par contre peut editer au minimum une édition et au maximum plusieurs éditions (un nombre quelconque de fois). Ce qui nous donne :

to n.PNG 

2.6 Modèle logique de données

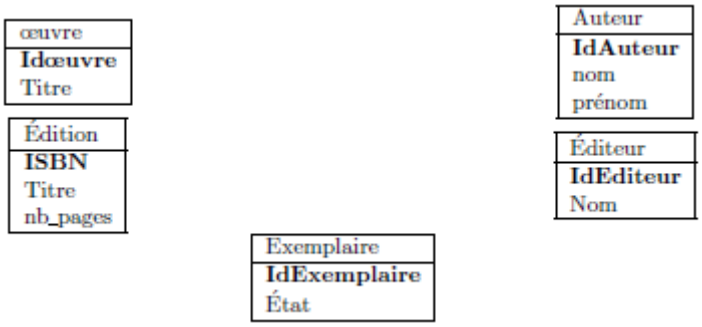
Une fois le MCD construit, l'étape suivante dans la conception de la base de données consist à concevoir le modèle logique de données, ou MLD. Ce MLD montre l'organisation des données sous forme de tables et est très proche de la manière dont les données vont être effectivement organisées dans Access. L'étape de transformation du MCD en MLD est assez simple et passe par trois étapes :

1. Transformation des entités en tables
2. Transformation des relations du MCD
3. Suppression des tables inutiles

2.6.1 construire des tables :

La première étape consiste 'a transformer toutes les entités du MCD en tables du MLD. Cette transformation est directe,il suffit de recopier les entités. Il s'agit essentiellement d'un changement de vocabulaire :
— Une **entité** devient **une table.**
— Une **propriété** devient **un champs.**
— Un **identifiant** devient **un clé primaire.**

A noter toutefois qu'il est essentiel qu'il n'y ait pas deux tables qui aient le même nom.
Exemple : la première partie de la construction du **MLD** du libraire est directe. Il suffit de recopier les entités



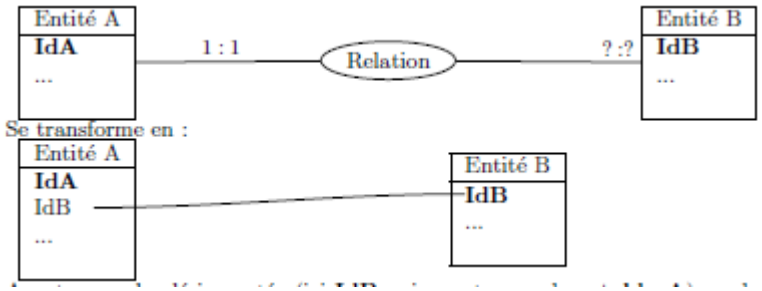
2.6.2 Transformation des relation en liens :

L'étape suivant consiste à transformer les relations du MCD en liens du MLD. Deux grands cas peuvent se présenter.

1. L'une des branches de la relation a une cardinalité de 1 :1 ou 0 :1
2. les autres cas

Premier Cas
Dans le cas d'une relation ou l'une des branches a une cardinalité de 1 :1 ou 0 :1, la transformation de la relation de la relation se fait de la maniére suivante :

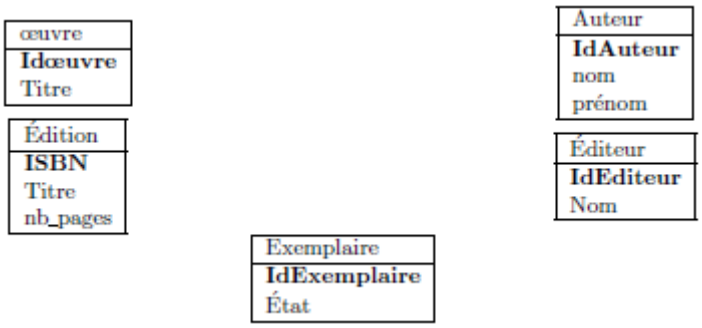
— On ramène dans la table correspondant à l'entité **"du coté du 1 :1"** (ou du 0 :1) la clé primaire de l'autre table ainsi que toutes les éventuelles propriétés de la relation.
— On lie la clé primaire ainsi importée avec la clé primaire de la deuxième table.
— Si la relation contenait des propriétés, celle-ci se retrouve également importée du coté du 1 :1.



On remarque que la clé importée ici IdB qui se retrouve dans table A ne devient pas une clé de la table : c'est une propriété comme une autre. Notons aussi que le lien se fait entre champs (on relie IdA à IdB) et non pas, comme dans le **MCD**, entre les tables.

Deuxième Cas
Dans tous les autres cas, la relation du **MCD** se transforme en une table du **MLD** :

— On crée une nouvelle table correspondant à la relation. Cette table contient toutes les éventuelles propriétés de la relation.
— On intègre à cette table les clés primaires des entités impliquées dans la relation.
— On relie les clés primaires des tables avec les clés importées dans la nouvelle table.
— On choisit en fin la ou les clés primaires de la nouvelle table. L'idée générale est que chaque occurrence de cette entité doit pouvoir être identifié de manière unique par ses clés primaires.
Cela revient en général à choisir comme clés primaires l'ensemble des clés importés des autres tables.



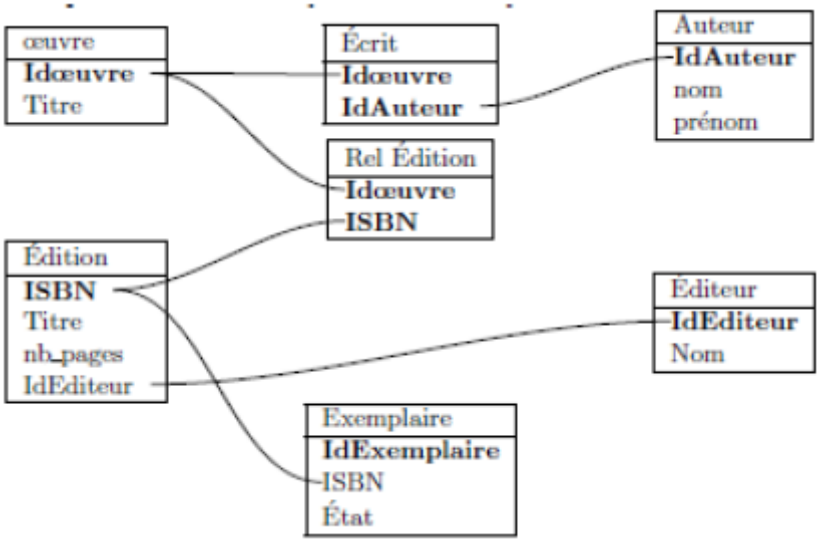
Cas particulière

— Si la relation est de type 1 :1 - 1 :1, on fusionne les deux entités en une. Ce type de relation rare est souvent due à un problème dans la conception du MCD.
— Si la relation est de type 0 :1 - 1 :1, on traite la relation comme une relation de type 1 :1 -? : ? (en ramenant la clé primaire du coté du 1 :1).
— Les relations réflexives (entre une entité et elle-même) se traitent comme les autres relations.
— Les relations ternaires (entre trois entités, ou plus), se traitent comme d'habitude. Si l'une des branches a une cardinalité de type 1 :1, on ramène les clés primaires des autres entités et les propriété de la relation dans l'entité

"du coté du 1 :1".
Si ce n'est pas le cas, la relation se transforme en table

2.6.3 Suppression des tables inutiles :

La dernière étape consiste simplement à supprimer les tables inutiles. En général (mais pas toujours), une table qui ne contient qu'un seul champ (sa clé) est inutile : elle ne nous apporte aucune information.
L'exemple le plus classique est une entité de type "date".
Exemple : Le MLD correspondant à l'exemple du libraire est le suivant :



2.7 Création de la base de données dans Access

Après avoir conçu un **MLD**, l'étape finale consiste à concevoir le Modèle physique de nos données. Il s'agit ni plus ni moins que d'intégrer le MLD au sein du SGBD(Access).
Cette opération comprend trois phases essentielles.

1. **Création de la table** : il s'agit simplement d'ouvrir Access et de choisir "Nouvelle base de données"
2. **Création des tables** : toutes les tables du MLD doivent être créées. Les données peuvent éventuellement provenir d'un logiciel extérieur (feuille Excel, document textuel...). Il faut également spécifier le type de donnée que doit contenir chaque champ de la table et préciser quelle est la ou les clé(s) de la table.
3. **Création des liens** : tous les liens qui apparaissent dans le MLD doivent apparaître dans la base de données

2.7.1 Création des tables dans ACCESS :

Il existe deux principales méthodes pour créer une table dans Access. Soit la table est créée directement dans Access, soit elle provient d'un logiciel extérieur. On traiterons ici de la création dans Access. L'objectif final est de créer au sein d'Access toutes les tables présentes dans le MLD.

Définition 2.3 *Une table est ensemble de données relatives à une même entité,structurées sous forme d'un tableau . une table peut être appelée aussi "RELATION"*

Définition 2.4 *Une colonne (champ) représente une propriété élémentaire de l'entité décrite par cette table*

Définition 2.5 *Une ligne (enregistrement) représente une occurrence du sujet représenté par la table*

Définition 2.6 *La clé primaire d'une table est un champ ou un ensemble de champs permettant d'identifier de manière unique chaque enregistrement de la table.*

Mode feuille de données

: Il s'agite de créer une table vide à laquellle on va ajouter des champs en mode **Feuille de données**.

1. Dans l'onglet **Créer**,on clique sur le bouton **Table**.
2. Une nouvelle table contenant uniquement le champs **N** s'ouvre et la colonne **Ajouter un nouveau champ** est sélectionnée.
   Puis on ajoute les champs à la table après on spécifie le stypes de données pour chaque champ .
3. **Enregistrer la table :** on clique sur le bouton **Office** puis sur l'option **Enregistrer** ou on cliquons sur Enregistrer de la barre d'outils **Accés rapide**, ou encore on utilise le raccourci-claviet **Ctrl S** .
4. Ensuite on va saisié **le Nom de la table** dans la zone de saisie correspondante de la fenêtre qui s'affiche puis on clique sur le bouton **OK**.

2.7.2 Mode création

1. Dans l'onglet **Créer** ,on clique sur le bouton **Création de table** visible dans le groupe Table.
   Le fenêtre de création d'une table s'affiche à l'écran.On remarque l'apparition de l'onglet contextuel **Création** qui est sélectionné.
   Pour chaque champ à insérer dans la table :
2. On clique dans la première cellule vide de la colonne **Nom du champ** et on tape le nom du champ ,de 1 à64 caractères maximun à l'exception du point (.) , du point d'exclamation (!) ,de l'apostrophe inverse (') et des crochets ([]).

3. On sélectionne le **Type de données** qui on souhaite affecter à ce champ.On ovuvre la liste déroulante puis on clique sur le type souhaité.

2.7.3 Créer une clé primaire

Une clé primaire organise nos données en identifiant chaque enregistrement de manière unique.
Sans clé primaire notre base de données pourrait être totalement désorientée.si on construi une nouvelle table en mode création sans spécifier de clé primaire ,Access en ajoutera une lorsque on enregistre la table,une table ne peut posséder qu'une clé primaire .
cette clé primaire automatique appelée N et elle est de type
NuméroAuto.la clé ne peut pas etre un champ de type Mémo ,objet OLE, ou lien hypertexte .Laurie ulrich fuller [2007]
. Access indexe automatiquement le champ de clé primaire.

Pour créer une clé primaire on suie ces étapes :

1. On ouvre la table en mode création .
2. On clique sur le nom du champ à transformer en clé primaire .
3. Sur le ruban ,on clique sur le bouton géant illustré d'une clé et légendé **Clé primaire**.
   Un symbole de clé apparait sur la case qui précède le nom du champ dans le panneau central de l'espace de travail.

2.7.4 Modifier les propriétés des champs

Chaque champ possède des valeurs de propriétés qui sont les caractéristiques permettant de définir le champ .Les propriétés disponible peuvent changer en fonction du type du champ sélectionnée .Inisan [15 juille 2007]

— Dans le volet de Navigation on fait un clic sur la table contenant les champs pour lesquels les propriétés doivent etre modifiées puis on clique sur l'option **Mode Création**.
— Ensuite on clique sur le champ pour lequel on souhaite modifier les propriétés.
— On clique dans la zone de la propriété concernée puis nous effectuons nos modification :

Taille du champ : pour un champ de type texte, il s'agit de la longueur maximale (en nombre de caractère) d'un élément du champ .
Format : Permet de définir l'apparence du champ à l'écran.
Masque de saisie : permet l'utilisation d'un masque de saisie qui peut faciliter grandement la saisie des données. Utile notamment pour les numéros de téléphone, code ISBN ou autre référence.

Valeur par défaut : Une valeur proposée par défaut pour le champ.

Valide Si : Conditions pour lesquelles la valeur entrée par utilisateur est accepté.
Message si erreur : Message indiqué à l'utilisateur si son information n'est pas valide, selon la condition entrée dans Valide Si.
Null Interdit : Accepte-t'on que l'utilisateur laisse éventuellement ce champ vide.Dans le cas d'une clé importée, c'est OUI si la relation dans le MCD était de type 0 :1, NON si c'était 1 :1.
Chaine vide autorisée : Autorise-t'on une réponse vide, a priori laisser la valeur par défaut.
Indexé : Laisser la valeur par défaut.
Compression Unicode : Laisser la valeur par défaut.
Liste de choix : Pratique si on désire que l'utilisateur entre son information dans une liste. Il faut alors préciser soit les éléments de la liste, soit le champ de la table dans lequel se trouvent ces informations. Intéressant surtout pour les clés importées.

2.8 Des relations sur la table

Après avoir créé une table pour chaque sujet de la base de données, on doit donner les moyens à Access de renvoyer ces informations en cas de besoin. Pour ce faire, on vas insérer des champs communs dans les tables liées et définir des relations entre ces tables. On peut alors créer des requêtes, des formulaires et des états qui affichent des informations issues de plusieurs tables à la fois.
Avant de nous lancer dans la création de relations ,il faut que l'intérioriser les notion suivantes :

— on ne peut lier que des tables qui de trouvent dans la même base de données.
— Les tables que on veut mettre relation doivent posséder au moins un champ en commun.Et doivent contenir le même type de données.
— En générale le champ de liaison est la clé primaire d'une des deux tables .Mais il est rare qu'il s'agisse de la clé primaire des deux tables.

2.8.1 Types de relations :

Sous Access ,il existe plusieurs types de relations entre les tables .Trois pour être précis :

Un-à-plusieurs : Dans ce type de relation il est possible de lier un enregistrement de la première table à plusieurs enregistrements de la seconde .Laurie ulrich fuller [2007]

Un -à-un : Ici un enregistrement de la première table ne peut correspondre qu'à

un et un seul enregistrement de la seconde table. Laurie ulrich fuller [2007] les relations un-à-un sont plutôt rares .En fait si deux tables possèdent une telle relation il est généralement préférable de les fusionner.

Plusieurs -à-plusieurs : Cette fois plusieurs enregistrement de la première table peuvent être liés à plusieurs enregistrements de la seconde.

2.9 Relation table à table :

Pour faire des relations entre tables on doit en premier lieu ouvrir la fenêtre **Relations** .Pour cela :

1. On clique au-dessus du ruban sur l'onglet **Outils** de base de données .
2. Dans la rubrique **Afficher/Masquer**,on clique sur le bouton **Relations**. La fenêtre Relations apparaît en même temps que la boite de dialogue **Afficher la table**.On peut maintenant sélectionner et lier nos tables.

Sélectionner les tables :

— La fenêtre Relation étant ouverte,On clique sur le bouton **Afficher la table** . La boite de dialogue Afficher la table apparaît .Elle contient la liste des tables de la base de données courante.
— Pour chaque table que nous voulons mettre en relation ,on clique sur son nom puis sur le bouton **Ajouter**.
— Lorsque on a terminé ,on clique sur le bouton **Fermer**.

Créer des relation :
Une fois notre table sélectionnés ,on suis ces étapes pour la création d'une relation :

1. On commence par décider quelles tables on veut lier.
2. Pour sélectionner la clé primaire ;on place le pointeur de la souris au-desssus de la ligne correspondante de la table . on appuie sur le bouton gauche de la souris et on laisse le enfoncé.
3. Ensuite on glisse la ligne du champ de la tableX vers la tableY.Un signe plus apparaît à la base de données. La boite de dialogue **Modifier des relations** va apparaître.
4. Dans la boite de dialogue Modifier les relations ,On coche l'option **Appliquer l'intégrité référentielle.**
5. On clique ensuite sur **Créer**. Pour lier d'autres tables ,on répète les étapes précédentes .
6. Quand on termine la création ,on referme la fenêtre **Relations** confirmons l'enregistrement de nos définitions en cliquant sur OUI.

2.10 Finalisation de la base

Une fois la base de données terminée, On ne devrait pas avoir à manipuler directement les différentes tables et requêtes. on manipulera la base à l'aide d'états et de formulaires.
Un état permet une présentation agréable des données de la base de données et propose ainsi une vue soit sur une table de la base, soit sur les données sélectionnées par une requête de sélection. La navigation parmi ses différents éléments se fait classiquement à l'aide d'un menu.

Interrogation d'une base de données

3

3.1 Introduction

Dans une base de données, les informations sont structurées en rubriques, champs, ... L'utilisateur dispose généralement d'un langage de commandes (par exemple SQL) qui lui permet d'interroger la base de données afin d'obtenir une information précise. Les bases de données ont généralement une plus grande homogénéité interne que les hyperdocuments car l'ensemble les informations y sont répertoriées de façon standardisée. Cette contrainte permet l'interrogation (le système fait la recherche pour l'utilisateur), la réalisation de certains calculs (dans les 'spreadsheet') et de rapports (factures, devis, rapports d'examens médicaux,..).
Ce chapitre présente comment interroger des base de données sur MS Access et quelques définitions de langage SQL .Après on exprime les requetes select et action et comment l'utiliser pour isoler interroger des Bdd.

3.2 Généralité :

SQL(structured Query Language) est un langage d'interrogation de bases de données.Cela signifie qu'il permet de consulter et modifier le contenu d'une base de données.
En réalité SQL permet aussi bien de consulter et modifier la structure d'une base de données (créer ,modifier ou supprimer des tables et des relations) que de consulter et modifier les données (sélectionner ,insérer ,modifier ou supprimer des lignes à l'intérieure des tables).Il permet aussi de gérer les transactions et les droits d'accès .
SQL donc à la fois un langage de manipulation de données, de définition de données, de contrôle des transactions et ,de contrôle des données.
les instructions en SQL s'appellent des requêtes ce qui est somme tout aussi logique pour un langage assez sommaire en comparaison de langages de programmation.Il ne permet pas de faire des boucles ou des branchements conditionnels.En fait ne permet quasiment aucune forme d'algorithmique .C'est un

langage permet déclaratif c'est à dire permet tant de décrire le résultat recherché mais pas de préciser la manière de l'obtenir.les fondaments de l'informatique

3.3 REQUÊTES SELECT

La structure de base des requêtes SELECT est la suivante : **SELECT** [liste des colonnes à sélectionner séparées par des virgules ,] **FROM**[nom de la table qui contient les données recherchées].
Cela, précisons que si chaque **SGBD** a sa propre manière d'optimiser la traduction des requêtes en algorithmes, chaque système a aussi ses petites spécificités au niveau de la syntaxe du SQL.les fondaments de l'informatique
EXEMPLE : prenons comme exemple une liste des étudiants.
On Commence à présenté simplement par afficher la liste des étudiants (c'est-à-dire le contenu de la table Etudiant qui on a créer sur Ms Access).

Matricule	nom	prénom	Date de naissance	adresse
298489	Marcel	Pascale	21/05/1994	Rue des Hirondelles 2
300487	Beraok	Nicolas	14/09/1994	Rue de Bréderode 14
310452	Pins	Robert	1/08/1995	Rue des lilas 13
325637	David	Emilie	19/04/1995	Place de l'Etoile 14
330777	Sineri	Hugues	13/11/1994	Square des Tilleuls 12
380654	Piesnet	Marie-Paule	8/06/1995	Avenue Pasteur 35

FIGURE 3.1 – *Table Etudiant*

Syntaxe :
SELECTmatricule,nom,prénom**From**Étudiant
La requête select après l'exécution est :

On peut ajouter une colonne calculant l'âge de chaque étudiant. Pour ce faire, il nous faut calculer la différence (en jours) entre la date courante et la date de naissance. Avec Access, on obtient la date courante du système avec la fonction Date(). Il suffit alors de calculer la différence et de diviser le résultat par 365.
Pour bien faire, on peut ensuite arrondir le résultat à l'entier le plus proche au moyen de la fonction Round(x,p) où x représente le nombre à arrondir, et p le nombre de décimales souhaité.
Syntaxe :
SELECT matricul,nom,prénom, Round((Date()-[date de naissance])/ 365,0) FROM Etudiant

On pourrait encore extraire les initiales de chaque étudiant et les afficher dans une nouvelle colonne. Cela supposera d'avoir recours à une fonction pour sélectionner une partie seulement d'un champ textuel (ici le premier caractère à

Matricule	nom	Prénom
298489	Marcel	Pascale
300487	Beraok	Nicolas
310452	Pins	Robert
325637	David	Emilie
330777	Sineri	Hugues
380654	Piesnet	Marie-Paule

FIGURE 3.2 – *Requête Select*

Matricule	nom	Prénom	Expr1
298489	Marcel	Pascale	20
300487	Beraok	Nicolas	20
310452	Pins	Robert	19

FIGURE 3.3 – *Requête Age*

gauche, ce qui s'obtient avec la fonction **Left(x,n)** où x est la chaîne de caractère de départ, et n le nombre de caractères que l'on souhaite conserver).
On peut aussi ajouter une colonne contenant une valeur arbitraire, par exemple l'année académique en cours.
Syntaxe :
SELECT matricule,nom,prénom,Left(prenom,1) et Left(nom,1),"2013-2014" FROM Etudiant

Matricule	nom	Prénom	Expr1	Expr2
298489	Marcel	Pascale	PM	2013-2014
300487	Beraok	Nicolas	NB	2013-2014
310452	Pins	Robert	RP	2013-2014
325637	David	Emilie	ED	2013-2014
330777	Sineri	Hugues	HS	2013-2014
380654	Piesnet	Marie-Paule	MP	2013-2014

FIGURE 3.4 – *Requête select*

Chaque fois que nous ajoutons une colonne à celles existantes, le SGBD donne à ce nouveau champ un nom arbitraire (ici Expr1). On peut tout aussi bien déterminer ce nom soi-même, voire même modifier l'intitulé des colonnes

existantes tel qu'il apparaît dans les résultats (cela ne modifie en rien le nom réel des colonnes concernées, d'ailleurs une requête SELECT ne modifie jamais rien, elle ne fait que produire une « lecture » des données).
On détermine l'intitulé d'une colonne avec le mot-clé AS (une spécificité d'Access, la plupart des autres SGBD n'ont pas besoin de ce mot-clé pour comprendre que ce qui suit le champ est un alias).

Syntaxe :
SELECT matricule, Left(prenom,1) Left(nom,1) AS Initiales, Round((Date()-[date de naissance])/365,0) AS Age,2013-2014 AS Année FROM Etudiant

3.3.1 Sélectionner les lignes recherchées

On a envisagé plusieurs manières de déterminer les colonnes à afficher :
sélection de colonnes existantes et création « à la volée » (c'est-à-dire pour les seuls besoins de la requête au moment de son exécution) de colonnes supplémentaires.
On va voir maintenant comment agir sur la sélection des lignes.
Le premier outil à notre disposition est celui qui consiste à exclure de la présentation des résultats les lignes exactement identiques, c'est-à-dire redondantes. En principe, dans nos tables, aucune ligne ne devrait être parfaitement identique, puisque chaque table contient au minimum une clé primaire qui rend chaque ligne unique. Mais quand une requête ne sélectionne pas toutes les colonnes, et en particulier pas la colonne qui sert de clé primaire, il se peut que l'exécution de la sélection fasse apparaître des lignes identiques. Le comportement par défaut des SGBD par rapport à cette question diffère de l'un à l'autre. Certains excluent par défaut les lignes redondantes, les autres les conservent sauf contrordre.

SQL résout le problème en permettant les deux options. Juste derrière le mot-clé SELECT, il suffit d'insérer le mot-clé **ALL** pour expliciter qu'on souhaite voir toutes les lignes,même les doublons, ou **DISTINCT** pour exiger que les lignes redondantes soient supprimées des résultats.

Pour déterminer spécifiquement quelles lignes on souhaite sélectionner, il nous faut un moyen de préciser dans la requête un ou plusieurs critères de recherche. C'est justement le rôle du mot-clé **WHERE** que l'on ajoute à la fin d'une requête et que l'on fait suivre par des critères de recherche (qui porteront logiquement sur la valeur des colonnes extraites ou calculées).
Les critères eux-mêmes sont définis en utilisant les opérateurs de comparaison (=, >=, (différent)) et sont combinés en utilisant les opérateurs logiques (AND, OR et NOT).

On peut utiliser les parenthèses pour lever toute ambiguïté sur l'ordre dans lequel des conditions multiples doivent être combinées.

EXEMPLE : Dans notre exemple on commence par afficher les étudiants, en sélectionnant seulement ceux dont le matricule est supérieur à 310000.

Syntaxe :
SELECT * FROM Etudiant WHERE Matricule >"310000"

Matricule	nom	prénom	Date de naissance	Adresse
310452	Pins	Robert	1/08/1995	Rue des lilas 13
325637	David	Emilie	19/04/1995	Place de l'Etoile 14
330777	Sineri	Hugues	13/11/1994	Square des Tilleuls12
380654	Piesnet	Marie-Paule	8/06/1995	Avenue Pasteur 35

FIGURE 3.5 – *Requête*

On voit que seuls 4 des 6 étudiants répondent à ce critère. On peut encore restreindre la sélection en ne conservant que ceux qui sont nés en 1995. Nous devons ajouter une condition supplémentaire dans la clause WHERE qui portera sur une valeur calculée : la fonction **Year(x)** nous sera utile pour obtenir l'année de chaque date de naissance.

Syntaxe : SELECT * FROM Etudiant WHERE Matricule >"310000" AND Year([Date de naissance])=1995

Matricule	nom	prénom	Date de naissance	adresse
310452	Pins	Robert	1/08/1995	Rue de Lilas 13
380654	piesnet	Marie Paule	8/06/1995	Avenue pasteur 35
325637	David	Emilie	19/04/1995	Place de l'Etoile 4

Quand on souhaite restreindre les résultats aux enregistrements dont l'une des colonnes en particulier est nulle (ou au contraire exclure celles-là des résultats), la chose peut se corser un peu. Un champ numérique vide par exemple ne veut pas dire qu'il contient la valeur 0 (on écrirait alors simple- ment WHERE champ=0), et un champ textuel vide ne veut pas dire qu'il ne comprend que des espaces. Non, on vous parle de la possibilité qu'un champ ne comprenne rien. Les SGBD ont un mot pour désigner l'absence de valeur dans un champ, elles disent simplement que le champ en question est nul (IS NULL). Pas un chiffre, pas une lettre, c'est vrai que c'est assez nul. Si on cherche les étudiants dont la date de naissance est manquante par exemple, la clause conditionnelle à appliquer devient WHERE [Date de naissance] IS NULL. Et pour, au contraire, exclure ces lignes-là : WHERE [Date de naissance] IS NOT NULL.

3.3.2 Trier les résultats :

Nous voici donc capables de sélectionner les lignes qui nous intéressent sur base de critères de sélection parfois assez subtils. Il nous manque à présent la possibilité de déterminer l'ordre dans lequel les lignes apparaissent dans les résultats. On peut par exemple souhaiter afficher la liste des cours classée par ordre alphabétique, ou bien par nombre de crédits décroissants (question d'afficher les cours les plus importants en premier).
Ordonner des informations "ORDER BY", et ses subtils appendices " ASC" (pour ascendant, c'est-à-dire le comportement par défaut) ou "DESC" (pour descendant).

EXEMPLE : On va trier la liste des cours par ordre alphabétique

Syntaxe : SELECT * FROM Cours ORDER BY Titre

mnemonique	titre	titulaire	ects	Discipline (FK)
GESTS203	Analyse des états financiers	Faska Comte	3	3
LANGS201	Anglais I	Ian Smith	4	5
DROIS201	Droit commercial	Françoise Juge	3	1
INFOS202	Introduction à l'informatique	Nicolas Code	5	4
MATHS201	Mathématique II	Marie Cosinus	5	6
STATS202	Probabilités et inférence statistique	Lise Peutet	10	6
ECONS201	Théorie macroéconomique I	Robert Ycroit	6	2

Ou bien par DESC ;
Syntaxe : SELECT * FROM Cours ORDER BY ects DESC

mnemonique	titre	titulaire	ects	Discipline (FK)
STATS202	Probabilités et inférence statistique	Lise Peutet	10	6
ECONS201	Théorie macroéconomique I	Robert Ycroit	6	2
MATHS201	Mathématique II	Marie Cosinus	5	6
INFOS202	Introduction à l'informatique	Nicolas Code	5	4
LANGS201	Anglais I	Ian Smith	4	5
GESTS203	Analyse des états financiers	Faska Comte	3	3
DROIS201	Droit commercial	Françoise Juge	3	1

Effectuer des regroupements et des traitements statistiques : le langage SQL peut réaliser des opérations sur les lignes, c'est-à-dire calculer des statistiques,comme par exemple calculer l'âge moyen des étudiants, ou la moyenne des notes obtenues à chaque cours.
Tout ce dont nous avons besoin, c'est de fonctions d'agrégation qui puissent s'exécuter à travers les lignes.
Pour rester simples, tous les SGBD offrent au minimum les cinq fonctions d'agrégation suivantes :

— AVG : Calcule la moyenne des valeurs d'une colonne
— MAX : Calcule la valeur maximale d'une colonne
— MIN : Calcule la valeur minimale d'une colonne
— SUM : Calcule la somme des valeurs d'une colonne

— COUNT : Compte le nombre de valeurs observées dans une colonne (ce qui revient à compter le nombre de lignes dans la table)

3.4 Requêtes Action

Après toutes ces consultations de données, passons enfin à l'action, c'est-à-dire aux requêtes qui permettent non plus de lire les données mais bien de les modifier. Commençons par la requête **DELETE**, c'est une requête génératrice de frissons et simple.

3.4.1 Supprimer des données : la requête DELETE

À la différence des requêtes sélection, DELETE ne demande que très peu d'arguments ou de précision. Le nom de la table dont on souhaite supprimer les données suffit .En fait , c'est qu'à la différence d'un traitement de texte ou d'un tableur, il est généralement vain de chercher la fonction « Annuler » d'un système de gestion de base de données, et plus encore de chercher la fonction « Enregistrer ».
Quand une base de données exécute une requête, elle enregistre les modifications apportées directement dans la table, sur le support de stockage secondaire. L'explication tient simplement dans la raison d'être et le mode de fonctionnement d'un système de base de données.
on sait que les SGBD existent pour gérer des fichiers en exploitant leur structure logique indépendamment du stockage physique. Et elles réalisent cela en accédant directement aux données là où elles se trouvent sur le disque dur en fonction des clés ou index.
Mais cela implique qu'à la différence d'un programme classique comme un traitement de texte, une base de données ne charge pas l'intégralité d'un fichier dans la mémoire vive de l'ordinateur avant de commencer à travailler. Elle travaille directement sur le disque dur. Et c'est très bien ainsi, surtout quand les bases de données sont volumineuses. Devoir attendre leur chargement complet en mémoire vive avant de pouvoir commencer à travailler serait particulièrement irritant, et bien souvent voué à l'échec tant la taille des bases de données a tendance à dépasser celle des mémoires vives.les fondaments de l'informatique
Dès lors, si on exécute une requête DELETE sur une table, le contenu de la table .
On manipule une requête DELETE avec beaucoup de délicatesse, et encore plus d'hésitation. Cela étant, pour brutale et irrémédiable que soit une requête DELETE, elle n'en est pas pour autant binaire,elle est parfaitement capable de supprimer des lignes beaucoup plus sélectivement. Et pour lui préciser quelle ligne on souhaite supprimer, on va utilise la même **clause WHERE** que celle que on aurai utilisée pour sélectionner des données.

D'ailleurs, ce n'est pas une mauvaise idée d'afficher avec une requête **SELECT** les lignes qui seraient effacées si on remplace simplement **SELECT** ... **FROM** par DELETE **FROM**.

3.4.2 Insérer des données : la requête INSERT INTO

En étant logique, on aurai dû entamer SQL avec la requête qui permet d'alimenter les tables de la base de données, c'est-à-dire de créer des enregistrements. Une seule requête permet de créer des enregistrements, c'est la requête **INSERT INTO**. C'est aussi la seule requête qui n'utilise pas le mot-clé **FROM** pour désigner la table sur laquelle elle travaille mais plutôt le mot-clé **INTO**. Dans la mesure où le terme est assez juste, ne lui en tenent pas rigueur.
C'est enfin la seule requête qui n'a que faire de la clause WHERE. Une requête d'insertion de données n'a besoin en réalité que de trois choses :

1. le nom de la table où on souhaite insérer des enregistrements
2. la liste des colonnes de cette table pour lesquelles on compte fournir des valeurs .
3. les valeurs en question.

Syntaxe : INSERT INTO table(colonnes) VALUES(valeurs).

En principe, une requête INSERT INTO n'ajoute qu'un seul enregistrement à la fois (sauf utilisation d'une sous-requête,).

Modifier des données : la requête UPDATE

Afin de boucler notre survol du langage SQL, il ne nous reste plus qu'à découvrir la modification des données avec la requête UPDATE. C'est une requête assez simple. Elle se contente de trois éléments :

— la table dont elle doit modifier les données,
— les modifications à apporter, et si besoin .
— les conditions précisant quelles lignes elle doit modifier.
Sa syntaxe a la forme suivante :

UPDATE table
SET champ1=valeur1, . . . , champK=valeurK
WHERE condition1. . .

par exemple on veut replier les notes des étudiants . on pourrai par défaut donner une note de 10/20 à tous les étudiants :
Syntaxe :
UPDATE
Etudiant SET

Note=10

3.5 Cachez des requêtes :

Tous les systèmes d'entreprise et la plupart des sites Web dynamiques reposent sur l'utilisation d'une base de données par les programmes. La question de l'interfaçage entre le programme (ou son langage de programmation) et la base de données qu'on interroge en SQL se pose dès lors avec acuité sitôt qu'on entreprend le développement d'une application de ce type.

En règle générale, cette intégration s'opère en utilisant dans le logiciel des interfaces prédéveloppées dans le kit de développement du langage utilisé (les « interfaces de programmation d'application » ou API).

Ces interfaces fournissent la capacité d'établir une connexion à une base de données à l'intérieur d'un programme, d'envoyer des requêtes SQL à la base de données, et de récupérer les résultats renvoyés par la requête, généralement sous la forme d'une liste .
Le langage de programmation peut alors reprendre la main pour manipuler les résultats contenus dans la liste, par exemple pour réaliser des calculs ou pour présenter à sa manière tout ou partie des résultats.
Conformément à l'évolution globale de l'informatique qui vise à en masquer la complexité et à la scinder en différentes couches aussi indépendantes que possible, les langages de programmation masquent de plus en plus la couche SQL et l'établissement des connexions aux bases de données en en déchargeant le développeur du logiciel. C'est particulièrement le cas des langages de programmation orientés objet contemporains (tel Python).

Concrètement, le développeur définit des classes correspondant aux tables de sa base de données, dont les attributs ont la même structure que les colonnes de la table correspondante. Ces classes définissent le modèle de données du programme, c'est-à-dire l'équivalent orienté objet du schéma relationnel de la base de données.

Expression des requêtes 4

4.1 Introduction

Les expressions sont généralement des techniques de programmation destinées à isoler des enregistrements contenant des séquences remarquables dans leurs champs. On les exploite pour trouver tout ce qui se rapproche de la recherche fournie, sachant qu'aucun champ ne contient précisément des expressions précises et identiques. doc

On va voir dans ce chapitre la définition d' une requête comment la créer et comment utiliser Les critères de requête pour centrer, appeler, isoler , notre recherche sur des éléments spécifiques d'une base de données Acces .

Définition 4.1 *Les requêtes sont principalement le moyen de rechercher et de compiler des données à partir d'une ou de plusieurs tables. La requête en cours est la même que si on pose une question détaillée sur notre base de données. Construire une requête en accès signifie qu'on définit une condition de recherche spécifique pour trouver exactement les données qu'on veut. Laurie ulrich fuller [2007]*

4.2 Création de requêtes

Pour créer une requête à une seule table

1. On sélectionne l'onglet de création sur le ruban et on recherche les requêtes ensuite on appuie sur la conception de requête.
   Access passera en mode **Création de requête**. Dans la boîte de dialogue afficher la table.
2. On sélectionne la table sur laquelle on souhaite exécuter une requête.
3. On Clique sur **Ajouter**, puis on clique sur l'option Fermer.

L'élément sélectionné apparaît sous la forme d'une petite fenêtre dans le volet relations entre objets. Dans la fenêtre de tableau, double-clique dans les noms de champs que on souhaite inclure dans notre requête.

EXEMPLE :
voici la table Employé qui on a créer en mode création Access :

N	Nom	Prénom	Titre	salaire	date	Categorie	Ville
1	Remou	siham	Ouvrier	25000	12/12/91	3	Tizi ouzou
2	Belaidi	Lyes	Vendeur	36000	02/03/91	4	Alger
3	Azarou	Ahmed	Secrétaire	19000	02/04/93	2	Blida
4	Oulouche	lilia	Admistrateure	69000	12/01/96	4	Bedjaia
5	Mukhtari	Ahmed	Administrateur	68220	30/01/99	1	Alger
6	Ramdani	Melissa	ouvrier	21000	04/04/95	0	Alger

TABLE 4.1 – *Table Employé*

— En haut de la fenêtre Access, on clique sur l'onglet Créer .
— Dans la section Requêtes du ruban, on clique sur le bouton Création de requête,
— Dans la boîte de dialogue qui suit, on sélectionne la table Employé.

  Dès lors, on clique sur le bouton Ajouter puis sur le bouton Fermer.
  Dans l'éditeur de requête, la table apparaît dans une représentation schématisée avec l'énumération de tous ses champs.
— Enregistrer la requête sous le nom : R-Employé.

4.2.1 Expression :

Une expression est une combinaison de tout ou partie des éléments suivants : fonctions intégrées ou définies par l'utilisateur, identificateurs, opérateurs, valeurs et constantes dont le résultat est une valeur unique.
Un critère de requête est une expression qu'Access compare aux valeurs des champs de la requête pour déterminer s'il faut inclure l'enregistrement contenant chaque valeur.

Modes d'utilisation des expressions

On peut utiliser des expressions pour :

— Calculer des valeurs qui n'existent pas directement dans nos données.
— Définir une valeur par défaut pour un champ de table ou un contrôle figurant sur un formulaire ou état.
— Créer une règle de validation afin de déterminer les valeurs que les utilisateurs peuvent entrer dans un champ ou un contrôle.
— Définir les critères d'une requête pour limiter les résultats à un sous-ensemble souhaité.

Composants des expressions :

Une expression comprend différents composants qu'on peut utiliser seuls ou combinés pour produire un résultat. Ces composants sont les suivants :

— Identificateurs Il s'agit des noms des champs de table ou des contrôles inclus dans les formulaires ou états, ou des propriétés de ces champs ou contrôles.
— Opérateurs Par exemple, les signes + (plus) ou - (moins).
— Fonctions Par exemple, SOMME ou MOYENNE.
— Constantes Il s'agit de valeurs immuables, comme les chaînes de texte ou les nombres qui ne sont pas calculés par une expression.
— Valeurs Chaînes (par exemple, « Entrer un nombre compris entre 1 et 10 ») ou nombres (par exemple, 1 254) utilisés dans les opérations. Enfin, on peut utiliser une expression pour définir des critères pour une requête. Par exemple, supposons qu'on veux afficher le prénom et le nom des personnes ayant pour prénom Ahmed ,on écrive Ahmed dans la deuxième ligne de critère sous le champ prénom .
Voici ce qui devrait ressembler :

Champ :	Nom	Prénom
Table :	Employé	
Tri :		
Afficher :	X	X
Critère :		"Ahmed"
ou :		

TABLE 4.2 – *critères*

Exécution ,on cliquons sur le bouton **!**
Résultat.

Requête 1	
Nom	prénom
Azarou	Ahmed
Mukhtari	Ahmed

Critère **Comme** est utilisé pour extraire tout ce qui ressemble à ce qui suit dans l'expression. exemple pour afficher le prénom ,nom de toutes les personnes dont le nom de famille commence par la lettre "A",on écrive seulement **comme A***.

Resultat.

Champ :	Nom	Prénom
Table :	Employé	
Tri :		
Afficher :	X	X
Critère :	comme A*	
ou :		

TABLE 4.3 – *critères*

Requête 2	
Nom	prénom
Azarou	Ahmed

Critère **" Entre "** et **"ET"**
on veut afficher le prénom ,nom des personne embauchées a la date 91.

Résultat.

Requête 3		
Nom	prénom	date
Remou	Sihame	12/12/91
Belaidi	Lyes	02/03/91

4.2.2 Fonctions :

Une fonction est une procédure qu'on peut utiliser dans une expression. Certaines fonctions, par exemple Date, ne nécessitent aucune entrée pour fonctionner. Toutefois, la plupart des fonctions nécessitent des entrées, appelées arguments.
Dans l'exemple du début , la fonction PartDate utilise deux arguments : un argument d'intervalle, avec une valeur de "yyyy", et un argument de date, avec une valeur de [Customers]![BirthDate]. La fonction PartDate requiert au moins ces deux arguments (intervalle et date), mais peut accepter un maximum de quatre arguments.
La liste suivante présente des fonctions qui sont fréquemment utilisées dans les expressions.

— La fonction **Date** permet d'insérer la date système actuelle dans une expression. Elle est fréquemment utilisée avec la fonction Format, mais également avec des identificateurs pour les champs qui contiennent des données de date/heure.
syntaxe : =Date()

— La fonction **PartDate** permet de déterminer ou d'extraire une partie d'une date. En règle générale, il s'agit d'une date obtenue à partir d'un identificateur de champ, mais il peut aussi s'agir d'une valeur de date renvoyée par une autre fonction, telle que la fonction Date.

  Syntaxe : DatePart ("yyyy", Date())
— La fonction **DiffDate** permet de déterminer la différence entre deux dates, généralement entre une date obtenue à partir d'un identificateur de champ et une date obtenue à l'aide de la fonction Date.
  Syntaxe : =DateDiff("d", Now(), [Orders].[ReceiveBefore])-10
— La fonction **Format** permet d'appliquer un format à un identificateur et aux résultats d'une autre fonction.
  Syntaxe : Format([Date],"ww")=Format(Now(),"ww")-1
— La fonction **VraiFaux** permet d'évaluer une expression comme étant vraie ou fausse, puis de renvoyer une valeur si le résultat de l'expression est Vrai et une autre valeur si le résultat de l'expression est Faux.
  Syntaxe : =IIf([CountryRegion]="Italy", "Italian", "Some other language")
— La fonction **DansChaîne** permet de rechercher la position d'un caractère ou d'une chaîne au sein d'une autre chaîne. La chaîne dans laquelle la recherche est effectuée est généralement obtenue à partir d'un identificateur de champ.
— Les fonctions Gauche, ExtracChaîne et Droite permettent d'extraire des caractères d'une chaîne, en commençant par le caractère le plus à gauche (Gauche), une position spécifique au milieu (ExtracChaîne) ou le caractère le plus à droite (Droite). Elles sont fréquemment utilisées avec la fonction DansChaîne. La chaîne à partir de laquelle ces fonctions extraient des caractères est généralement obtenue à partir d'un identificateur de champ.
  syntaxe :
  Left([ProductName], 1)
  Right([AssetCode], 2)
  Mid([Phone],2,3) +
  ++0.

CONCEPTION DE FORMULAIRE ER D'ÉTATS 5

5.1 Introduction

Access nous offre un certain nombre d'outils qui nous aident à créer rapidement des rapports attrayants et faciles à lire qui présentent les données de la manière la plus adaptée aux besoins de ses utilisateurs. on peut utiliser les commandes de l'ongletCréer pour créer un formulaire et état simple d'un simple clic. on peut utiliser l'Assistant État pour créer un formulaire , état plus complexe, ou créer un formulaire, état en ajoutant nous-même toutes les données et les éléments de mise en forme. Quelle que soit la méthode choisie, nous apporterons probablement quelques modifications à la conception pour qu'il affiche les données comme nous le souhaitons.Inisan [15 juille 2007]
Ce chapitre traite du processus général de création des formulaire et d'états, puis nous montre comment ajouter des éléments de conception spécifiques et des modification à notre formulaires et notre états.

5.2 création d'un formulaire dans Access

Pour créer un formulaire à partir d'une table ou d'une requête dans notre base de données, dans le volet de navigation, on clique sur la table où la requête contenant les données pour le formulaire, puis, dans l'onglet Créer, on clique sur Formulaire.
Access crée un formulaire et l'affiche en mode Page. On peut apporter des modifications de conception, comme ajuster la taille des zones de texte en fonction des données à afficher, si nécessaire.

EXEMPLE : On créer un formulaire à partir de la table "Employée"

1. on sélectionne la table "Employée"
2. Après on clique créer formulaire

Résultat :

FIGURE 5.1 – *formulaire*

5.3 MODIFICATION D'UN FORMULAIRE :

Le formulaire étant crée, il est ensuite possible de le modifier. La modification d'un formulaire en suivant une méthode similaire à celle utilisée pour modifier un état. Pour modifier un formulaire, il faut passer en mode création (dans le menu affichage ou par l'icône mode création). On voit alors la structure du formulaire, à partir de là, il est possible de réaliser les opérations suivantes :

— Déplacer ou redimensionner un élément,
— Supprimer des éléments changer les propriétés des éléments (en cliquant sur l'élément avec le bouton droit de la souris)
— Ajouter des éléments (bouton, ligne, zone de texte...), pour cela appuyer sur le bouton "boite à outils", Lorsqu'on ajoute un éléement, il est possible de les ajouter directement pour ensuite modifier l'élément, ou, plus simplement, en utilisant un assistant.

Pour cela, on sélectionne sur la boite à outils le bouton représentant une "baguette magique" avec de créer l'élément. On trouve les élément

suivants :

— intitulé : il s'agit d'une zone de texte qui ne sert qu'à la présentation. Pour le créer, on sélectionne le bouton correspondant ("Aa") et on clique sur le formulaire à l'endroit où on veut placer votre intitulé et on écrive notre texte.
— zone de texte : une zone de texte est une zone qui sert à afficher la valeur d'un champ. Il est également possible d'ajouter des champs calculés à partir d'autres champs présents dans le formulaire.Pour la créer, on sélectionne le bouton correspondant ("abj") et on clique sur le formulaire à l'endroit où on veut placer notre zone de texte. Si on veut activé l'assistant ("baguette magique"), on demandera alors à quel champ doit correspondre la zone de texte. Sinon, dans la zone de texte, indiquez le champs qui nous intéresse
("[salaire]" par exemple, ou "=[salaire] + [catégorie]" si on veut une formule).
— Groupe d'options : un groupe d'options nous permet de choisir entre différents choix à l'aide de boutons ou de cases. Le groupe d'options est particulièrement adapté aux champs contenant un nombre limité de valeurs numériques "codés" (par exemple, les catégories sociaux professionnelles).
— liste déroulante : permettent le choix d'une valeur dans une liste.

5.4 UTILISATION DES IMAGES :

il est facile de définir une image en fond de formulaire pour agrémenter le coté graphique de notre application .
Au moment de la création du formulaire ,l'assistant formulaire propose quelques images prédéfinies ,par le biais de l'icône **Mise en forme automatique**.
Un compromis est envisageable pour conserver les images sans pénaliser l'application ;on procède ainsi :
— on apparaitre les propriétés du formulaire
— Sous l'onglet Format on donne la valeur Attachée à la propriété Type d'image.
Lorsqu'une image est intégrée à la base ,elle stockée dirctement dans le fichier .accdb,ce qui explique l'accroissement de taille.Inversement ,une image attachée reste un fichier indépendant sur le disque dur ;seul son chemin est mémorisé dans la base de données.hervé inisan

5.5 Reproduire la mise en forme :

L'icône Reproduire la mise en forme permet d'homogénéiser le style de divers objets sur le formulaire ,plutôt que d'appliquer du gras ,de l'italique ou des couleurs objets par objets.on suit ces étapes :

1. On clique sur un objet du formulaire (ou de létats) puis on clique sur l'icône Reproduire la mise en forme.
2. on clique sur l'objet "à peindre".
3. pour arrêter le processus ,on clique une dernière fois sur l'icône Reproduire la mise en forme.

5.6 Sous formulaire :

Les sous formulaires permettent d'éditer plusieurs tables en même temps. Supposons par exemple que, dans l'exemple du "Employée", on désire avoir un formulaire qui affiche tous les renseignements concernant les employées(nom, prénom, ...) .
L'opération peut se faire "à la main" ou à l'aide de l'assistant. Nous présentons ici la méthodes de la création.
La création du formulaire se fait de la manière suivante.
— On créer une requête qui contient les informations que devra contenir le formulaire (dans l'exemple, il s'agira d'une requête contenant tous les nom ,prénoms des employées et
— on crée un formulaire en se basant sur cette requête à l'aide de l'assistant.
— on Suit l'assistant. La procédure est la même que pour un formulaire simple à une nuance près. On nous demande comment on souhaite afficher nos données, on choisis le niveau de regroupement voulu (comme pour un état). puis "formulaire avec sous formulaire".

5.7 États :

Un état sert à visualiser les données. Il ne permet aucune modification. Les états sont en particulier tout à fait adaptés à l'impression des données : ils fournissent une présentation sous forme de feuilles prêtes à être imprimées.
La création d'un état se fait dans la section "état" puis "Nouveau". Différentes méthodes de création sont proposées :

— Mode création : création de l'état sans aucune aide.
— Assistant état : conseillé pour la plupart des utilisations.

— État Instantané Colonnes / Tableau : permet la création trés rapide d'états, mais sans aucune option de personnalisation.
— Assistant graphique : pour créer un graphique.
— Assistant Etiquette : génère des états sous forme d'étiquettes.

5.8 Modification de l'état :

Un état étant crée, il est possible de le modifier. Pour cela on clique sur le bouton "Modifier" après avoir sélectionné notre état (ou passez en mode "Création").

On observe alors les différentes parties de l'état :
— l'entête de l'état : c'est ce qui va être imprimé au début de l'état ; ce ne sera imprimé qu'une seule fois.
— l'entête de page : c'est ce qui est afficher au début de chaque page
— l'entête de groupe : c'est ce qui va etre imprimé au début de chaque groupe,le groupe étant l'un des niveaux de regroupements choisis lors de la création de l'état
— le détail : c'est ici que vont être imprimés chaque ligne de l'état
— le pied de groupe : c'est ce qui va être imprimé à la fin de chaque groupe,
— le pied de page : ce qui est imprimé en bas de chaque page (souvent la date et le numéro de pages),
— le pied d'état : imprimé une seule fois à la fin du document.

5.9 Section d'états :

Chaque état possède une ou plusieurs sections d'état. La section qui est présente dans chaque état est la section Détail. Cette section se répète une fois pour chaque enregistrement dans la table où la requête sur la base de l'état. Les autres sections sont facultatives et se répètent moins souvent, et sont généralement utilisées pour afficher des informations communes à un groupe d'enregistrements, à une page d'état ou à l'état entier.
La figure suivante décrit l'emplacement de chaque section et la manière dont elle est utilisée.

Section	Localisation	Contenu classique
Section En-tête d'état	N'apparaît qu'une seule fois, en haut de la première page du rapport.	▪ Titre du rapport ▪ Logo ▪ Date actuelle
Section Pied d'état	Apparaît après la dernière ligne de données, au-dessus de la section Pied de page sur la dernière page de l'état.	Totaux des états (sommes, comptages, moyennes, et ainsi de suite)
Section En-tête de page	Apparaît en haut de chaque page de l'état.	▪ Titre du rapport ▪ Numéro de page
Section Pied de page	Apparaît en bas de chaque page du rapport.	▪ Date actuelle ▪ Numéro de page
Section En-tête de groupe	Apparaît juste avant un groupe d'enregistrements.	Champ sur
Section Pied de groupe	Apparaît juste après un groupe d'enregistrements.	Total du groupe (sommes, nombres, moyennes, et ainsi de suite)

Figure 5.2 – *section d'états*

5.10 Ajouter des sections d'état ou d'en-tête de page et de pied de page

Dans le volet de navigation, on clique avec le bouton droit sur l'état à modifier, puis on clique sur Mode Création dans le menu raccourci. Les sections sont séparées par des barres horizontales ombrées appelées sélecteurs de section. L'étiquette de chaque sélecteur de section

indique ce que la section se trouve juste en dessous. Chaque état comprend une section détail et peut également contenir des sections En-tête d'état, En-tête de page, Pied de page et Pied d'état.

Pour ajouter des sections d'en-tête et de pied de page ou des sections d'en-tête et de pied d'état à notre état,on clique avec le bouton droit sur un sélecteur de section, puis on clique sur En-tête de page/Pied de page ou En-tête/Pied d'état dans le menu raccourci.

Access ajoute toujours des sections d'en-tête et de pied de page de page et d'état par paires. Autrement dit, nous ne pouvons pas ajouter une section d'en-tête de page ou d'état sans ajouter la section de pied de page correspondante.

La partie pratique

6

6.1 Créer une facture avec Access

Partie 1/3 - Création des tables et des relations :

on vas présenter ce chapitre en 3 parties,créer une facture sous Access. Si on gére la liste de clients sur Access, le fait de créer nos factures à partir de cette même base de données nous fera gagner un temps précieux.
Il y a différentes manières de créer cette "application" de gestion de factures, mais on va montrer comment en créer une de façon simple et nous pourrons tout à fait la personnaliser selon nos besoins.

on va faire de nombreuses manipulations dans cette partie et, pour éviter d'avoir à tout ré-expliquer dans le détail sur la raison pour laquelle on fait certaines actions, on doit savoir :
Créer une base de données Access(voir le chapitre 1 page 08 chapitre 2 page 18 – Création d'une table, chapitre 5 – Création d'une requête page32, et chapitre 5 pour – la Création d'un formulaire page37 , et d'un état page 40)

Créer des relations entre les tables (voir le chapitre 02 des relation sur la table s comment les créer)

Créer un sous-formulaire (voir l'article Créer un sous-formulaire Access chapitre 5 pages 39)

Ce chapitre est divisé en 3 parties
— Partie 1 : création des tables et des relations.
— Partie 2 : création du formulaire principal et de ses sous-formulaires.
— Partie 3 : création des états et de l'interface.
Pour chaque objet de la base de données, on placerai un préfixe avant son nom afin de mieux le retrouver : T pour table, R pour requête, F pour formulaire, S/F pour sous-formulaire et E pour état.
Création des tables Access :

Nous allons donc commencer par créer nos tables :
•T Clients : coordonnées de chaque client.
• T Date facture : servira en tant que sous-formulaire et répertoriera toutes les factures établies.
• T Facture : servira en tant que sous-formulaire.
•T Tarifs : répertorie tous les produits à vendre avec leurs tarifs datés. De cette manière, en cas de modification de tarifs, les factures antérieures ne seront pas modifiées, mais les factures ultérieures auront le nouveau tarif.

Les 2 tables suivantes vont être créées manuellement.
on crée la table T DATE FACTURE avec les champs suivants :

• ID Date facture : NumératoAuto. Ce champ est la clé primaire ;

• ID Client : Numérique ;

• Date Facture : Date/Heure. on Choisis le Format Date abrégé et le Masque de saisie du même nom en cliquant sur ;

•Mode de paiement : Assistant liste de choix (dans la boîte de dialogue, on sélectionnera on taperai les valeurs souhaitées et saisir les données suivantes les unes en dessous des autres : Chèque, Virement, Espèces, CESU. Cocher la case Limiter à la liste).

on Crée la table T FACTURES avec les champs suivants :
• ID Facture : NumératoAuto. Ce champ est la clé primaire ;

• ID Date facture : Numérique. Choisissez l'Index avec doublons ;

• ID Tarif : Numérique.on Choisis l'Index avec doublons ;

• Désignation : Texte court ;

• Quantité : Numérique ;

• Prix unitaire : Monétaire

On Crée nos relations de manière à avoir un lien :

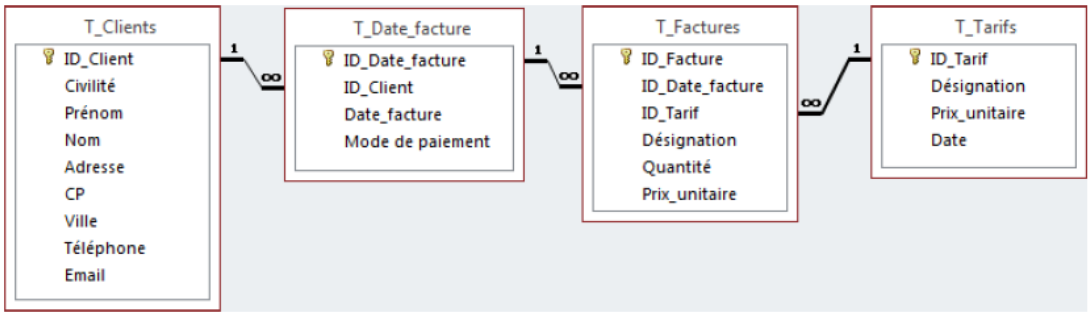
— Entre le champ ID Client de la table T Clients et le champ ID Client de la table T Date facture ;

— Entre le champ ID Date facture de la table T Date facture et le champ ID Date facture de la table T Factures ;

— Entre le champ ID Tarif de la table T Tarifs et le champ ID Tarif de la table T Factures.

Pour chaque relation, on coche les cases Appliquer l'intégrité référentielle, Mettre à jour en cascade les champs correspondants et effacer en cascade les enregistrements correspondants.

Nos relations devraient ressembler à ça :



Partie2 :Création du formulaire principal Access :
Nous allons maintenant créer le formulaire. Pour cela,

1. on ouvre l'Assistant Formulaire de l'onglet Créer,et on choisis la table T Clients et on sélectionne tous les champ
2. Après on choisis la disposition en Colonne simple et on clique sur Terminer en choisissant Modifier la structure du formulaire.
3. Nous allons modifier la présentation du formulaire, mais on peut bien sûr l'adapter à nos besoins.
4. On Supprime le titre T Clients en cliquant dessus et en appuyant sur la touche Suppr et réduiser la partie En-tête de formulaire en ramenant la barre de Détail au plus près.on Supprime également les étiquettes Civilité, Prénom, Nom, CP et Ville qui n'indique que leur titre.
5. on Modifie l'étiquette ID Client en N Client.

On peut dimensionner nos contrôles de manière identique en affichant la Feuille de propriétés de l'onglet Création et en modifiant les données des lignes Largeur et Hauteur de l'onglet Format.

on essaye d'éviter de trop réduire les contrôles auquel cas nos données seraient masquées en **Mode formulaire** et nous serions obligé de cliquer dessus et de nous déplacer à l'intérieur.

— On sélectionne tous les contrôles en traçant un rectangle avec notre souris et indiquer une hauteur de 0,552 cm. On applique une police Gras pour les étiquettes. Pour ces étiquettes, on les sélectionne puis on clique droit >Taille > Au contenu et réajustez la hauteur à 0,552 cm.

— On centre les étiquettes ID Client, Civilité, Prénom, Nom et CP.
Voici les tailles en largeur de chacun de nos contrôles :

— D Client : 0,989 cm ;
— Civilité : 1,905 cm ;
— Prénom : 2,989 cm ;
— Nom : 3,811 cm ;
— Adresse : 5,423 cm ;
— CP : 1,217 cm ;
— Ville : 5,714 cm ;
— Téléphone : 3,811 cm ;
— Email : 6,693 cm.

Dans la Feuille de propriétés , on sélectionne la sélection Formulaire et on indique une Largeur de 35 cm. puis on Sélectionne tous les contrôles et on retire le contour en allant dans Format > Contour > Transparent.
Enfin, on déplace les éléments de manière à avoir ce genre de présentation
Pour terminer la présentation de ce formulaire, on insére un bouton de contrôle grâce aux contrôles de l'onglet Création.

Comme nous n'utiliserons pas ce formulaire pour modifier les coordonnées des clients (même si c'est possible), nous n'avons pas besoin de connaître l'état de l'enregistrement. Nous allons donc masquer le sélecteur qui est la ligne verticale avec une flèche à gauche du formulaire en Mode formulaire

Dans la Feuille de propriétés , à l'onglet Format, à la ligne Afficher sélecteur, on change le Oui en Non.
On Ferme le formulaire en enregistrant les modifications.
Enfin on renomme le formulaire en cliquant droit dessus dans le panneau de navigation > Renommer > F Clients.

Conclusion générale

Actuellement, la plupart des logiciels de bases de données (SGBD) reposent sur le modèle relationnel. Une base de données est vue comme un ensemble de tables, structurées selon des formes normales et manipulées grâce à des logiciel et des langages non algorithmiques. Apres avoir mené à bien notre travail ayant comme intitulé " comment créer et gérer une base de données sur Ms Access , nous avons évoquer la façon d'utiliser le logiciel Microsoft Access . Par ailleurs ce travail a eu pour objectif :

— création d'une base de donnée à laide d'un modèles et à partir d'une base de données vide.

— construction des entités et transformation des relations. — constructions des tables et des relations.

— création de formulaires et d'états.

Perspectives

Dans la continuité directe de notre travail de thèse, Il serait vraiment intéressant de pouvoir faire une réelle mise à grande échelle de notre base de donnée ce que ne nous pouvons pas faire , faute de moyen. . . .

Bibliographie

hervé inisan. *les meilleure truc ;astuces et secrets rigoureusement*.

Hervé Inisan. *créez et utilisez une base de données*. 15 juille 2007.

john kaufeld Laurie ulrich fuller, ken cook. *Access pour les nuls*. 2007.

A. Lemay. *Informatique Base de données- Access :UFR L.E.A. - MST CI 2 ,*. 2007.

les fondaments de l'informatique. *les base de l'interrogation de données en sql*.

Livre « Je me lance avec Access 2007»,Hervé Inisan, janvier2007

taonix.fr

maths.unsw.

techlib.fr

Base de données Microsoft Access ,Riadh BOUSLIMI,2011

support.microsoft.com

Livre « Access 2007 Créez et utiliser une base de données »,Corinne Hervo,2007

Livre « Astuces le best of ,les meilleurs trucs,astuces et secrets rigoureusement sélectionnés»,Hervé Inisan,juin 2007

Livre « VBA Access 2016 programmer sous Access »,Jean

على تقدير نماذج الانحدار الذاتي

يعتمد عملنا في نهاية الدراسات على بحرسفت اكحسس ٢٠٠٧ من خلال التعامل خطوة بخطوة مع جميع الوظائف التي تسمح بإنشاء قاعدة بيانات وإدارتها : وصف بيئة اكحسس واستخدام التعليمات وإنشاء قاعدة بيانات وإدارة الكائنات التي يؤلفها إنشاء واستغلال الجداول والنماذج والتقارير ، وإدارة السجلات من خلال ورقة بيانات ونموذج ، وكذلك اختيار وحذف السجلات في مساعدة الاستعلام

Résumé

Notre travail de fin d'études se base sur Microsoft Access 2007 en abordant pas à pas toutes les fonctions permettant la création et gestion d'une base de données : description de l'environnement Access et utilisation de l'Aide , création d'une base de données et gestion des objets qui la composent, création et exploitation des tables , formulaires et états , gestion des enregistrements par l'intermédiaire d'une feuille de données et d'un formulaire , ainsi que sélection et suppression d'enregistrements à l'aide de requête.

Abstract

Our thesis is based on Microsoft Access 2007 and covers step by step all the functions allowing the creation and management of a database: description of the Access environment and use of the Help, creation of a database and management of its objects, creation and use of tables, forms and reports, management of records through a data sheet and a form, as well as selection and deletion of records using a query.