Informatique
Base de données - Access

UFR L.E.A. - MST CI 2

Janvier 2004

A. Lemay

Contenu du document.

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 considéré 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, Paradox, MySQL, PostgreSQL parmi beaucoup d'autres.

La plupart de ces systèmes sont basés (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). **Access**a comme avantage par rapport à la plupart de ses concurrents de permettre une écriture en mode graphique des tables, de leur relations et de la plupart des requêtes. De plus, il intègre un système de création d'applications claires et simples pour chaque base de données.

Ce document est composé de cinq parties :

1. cette introduction,
2. une partie consacrée à la méthode Merise, qui indique comment concevoir une base de données,
3. une partie consacrée à la création de la base de données dans le logiciel **Access**,
4. une partie consacrée aux différents type de requêtes,
5. et une dernière partie consacrée à la finalisation de la base de données (notamment conception des formulaires, des états et des menus) qui permet d'avoir un produit fini, utilisable par un utilisateur n'ayant pas de connaissances particulière sur les SGBD.

2 Méthode Merise

2.1 Principes généraux

La méthode Merise a été créée en France dans les années 1970. Cette méthode utilise le système dit *d'entités-relations*. Il s'agit d'un outil et d'une technique d'analyse permettant de construire des schémas théoriques de raisonnement sur des applications tournant avec des bases de données dite *relationnelles* (comme celles d'**Access**).

A noter que nous ne présenterons ici qu'une partie de la méthode Merise, puisque la méthode Merise générale traite de l'intégralité de la conception de la base de données : elle ne s'intéresse pas uniquement de la partie correspondant au stockage des données, mais également à leur traitement.

La méthode Merise considère quatre phases dans la création d'une base de données :

1. **La phase d'analyse :** cette phase, qui ne sera pas étudié dans ce document, est une phase essentielle qui consiste à
   - étudier l'existant : y-a-t'il un système qui gère déjà tout ou partie de l'information, qu'il s'agisse d'un logiciel ou d'un ensemble de documents papiers ? Comment ces informations sont elles stockées ? Quelles sont les informations stockées ? Que manque-t'il ? Qu'est ce qui convient ou ne convient pas aux utilisateurs ?
   - interroger les futurs utilisateurs : qu'attendent-ils du futur SGBD ? Quelles sont les opérations qu'ils désirent automatiser ?
   - recueillir les informations existantes, étudier les divers liens qui peuvent exister entre ces informations, mettre en évidence les règles de gestion employées, ...
2. **La phase conceptuelle :** elle consiste à représenter l'organisation des données de manière générale. Elle aboutit sur la création du *modèle conceptuel des données* (MCD) dans lequel les données sont représentées sous forme d'entités liées entre elles par des relations.
3. **La phase logique ou organisationnel :** dans cette phase, la base de données sont représentées sous une forme *logique* plus proche de leur représentation réelle au sein du SGBD : les informations sont représentées uniquement sous forme de tables au sein d'un *modèle logique des données* (MLD).
4. **La phase physique ou opérationnelle :** elle consiste à construire réellement la base de données au sein du SGBD (ici **Access**). Cette partie ne sera pas décrite dans cette section, mais dans les suivantes.

A retenir

Les quatre phases de la méthode **Merise** :

1. analyse (étude de l'existant et enquête)
2. conceptuel (création du MCD)
3. logique (création du MLD)
4. physique (conception de la base de données dans Access)

2.2 Modèle Conceptuel de données

Après la phase d'analyse, nous pouvons commencer à représenter les informations sous forme conceptuelle. Le *Modèle Conceptuel de Données* (MCD) que nous allons construire contient deux éléments principaux : les *entités* et les *relations*.

Une entité est un élément du problème. Elle est définie par un ensemble de propriétés. Chacune des propriété est l'un des éléments qui caractérise l'entité. 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 (l'employé "Jean Martin").

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.

Après avoir fait une analyse aussi complète que possible du problème à informatiser, la construction du MCD se fait en quatre étape :

1. **repérage des entités**,
2. **construction des entités, choix des propriétés**,
3. **construction des relations**,
4. **choix des cardinalités.**

2.2.1 Repérage des entités

Une entité est un composant du problème : une personne, une facture, un livre, ... C'est la représentation d'un objet matériel ou immatériel pourvu d'une existence propre et conforme aux choix de gestion de l'entreprise. Comme dit plus haut, ce que l'on considère comme entité est un type général (ex : l'entité *personne* représente toutes les personnes) à ne pas confondre avec une occurrence d'entité (Jean Martin étant une personne, on le considère comme une occurrence de l'entité *personne*). Une entité doit avoir une existence indépendamment de tout autre entité.

Exemple : On considère le problème suivant :

Un libraire gère des œuvres littéraires. Une œuvre est une création littéraire. Une œuvre a au moins un auteur et est dans une édition (un livre). Une édition possède un ISBN unique et a un unique éditeur. Elle peut contenir plusieurs œuvres. 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 "œuvre" : Une création littéraire, un récit...
- l'entité "auteur" : une personne créateur d'œuvre,
- l'entité "édition" : un livre contenant une ou plusieurs œuvres littéraires,
- l'entité "éditeur" : la société qui va imprimer les livres,
- l'entité "exemplaire" : un exemplaire physique de livre.

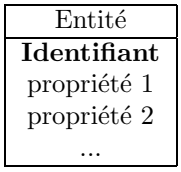
L'ISBN, par exemple, n'est pas une entité puisqu'il s'agit d'un élément qui caractérise l'entité "édition".

2.2.2 Construction des entités

L'étape suivante correspond à la construction des entités. On commence par donner un nom à chacune des entités. Il faut ensuite rechercher les propriétés 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),
- une propriété ne doit pas être "instable" ou "calculable" : si une propriété peut être obtenue par calcul à partir d'autres éléments qui vont apparaître dans la base de donnée (notamment d'autres propriétés), on ne doit pas la considérer,
- toute entité doit posséder une propriété particulière appelée sa **clé** (ou **identifiant**). Une clé doit caractériser de manière unique chaque occurrence de l'entité. Par exemple, le nom de famille d'une personne ne peut pas être considéré comme une clé d'une entité "personne" puisque deux personnes peuvent avoir le même nom de famille. Le numéro de sécurité sociale est par contre tout à fait acceptable. Il vaut mieux éviter les identifiants trop longs (on préférera un code de quelques chiffres à un intitulé d'une vingtaine de lettres par exemples),
- si aucune des propriétés "naturelles" ne peut servir de clé, on en rajoute une artificiellement (par exemple "CodeLivre" ou "IdAnimal").
- Chaque propriété ne doit dépendre que d'une seule entité.

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



Exemple : 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) :

- œuvre : **Idœuvre**, titre
- Auteur : **IdAuteur**, nom, prénom
- Édition : **ISBN**, titre, nb_pages
- Éditeur : **IdEditeur**, nom
- Exemplaire : **IdExemplaire**, état

Notons que, dans le cas de l'édition, l'ISBN est un identifiant tout à fait acceptable. Dans les autres cas, aucune des propriétés ne convenant comme identifiant, il a fallu créer une propriété particulière pour cet effet (Id...).

2.2.3 Construction des relations

L'étape suivante consiste à énumérer toutes les relations *possibles* entre entités. Si une relation a une chance d'apparaître (et de nous intéresser), alors on doit la considérer dans le MCD. On parle également parfois d'*association*.

Une relation se représente de la manière suivante :

On notera les points suivants :

- 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 néanmoins souvent possible (et préférable !) de se restreindre à des relations entre deux entités. Dans le cas ici, la relation **Vente** pourrait être remplacée par une entité **Acte de vente** qui est en relation avec l'acheteur, le vendeur et le lieu.

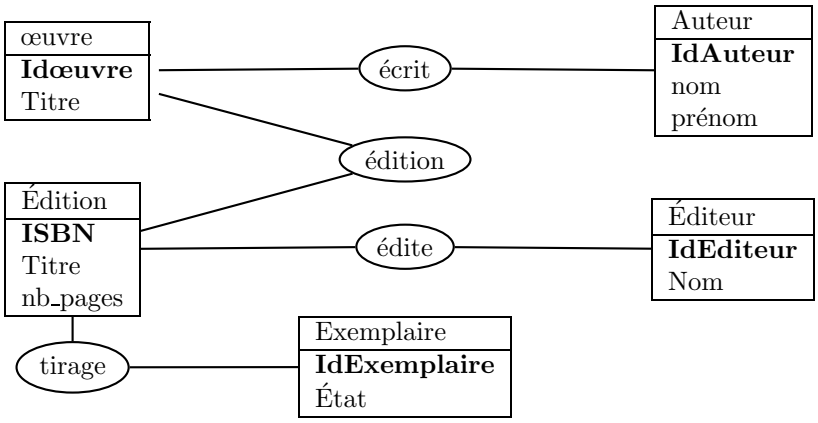
- Il est tout à fait possible d'avoir plusieurs relations entre deux entités.

- Il est également possible d'avoir une relation dite *réflexive*, c'est-à-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** lie 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 à une formation est systématiquement inscrit à tous les cours qui composent la formation.

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



2.2.4 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.

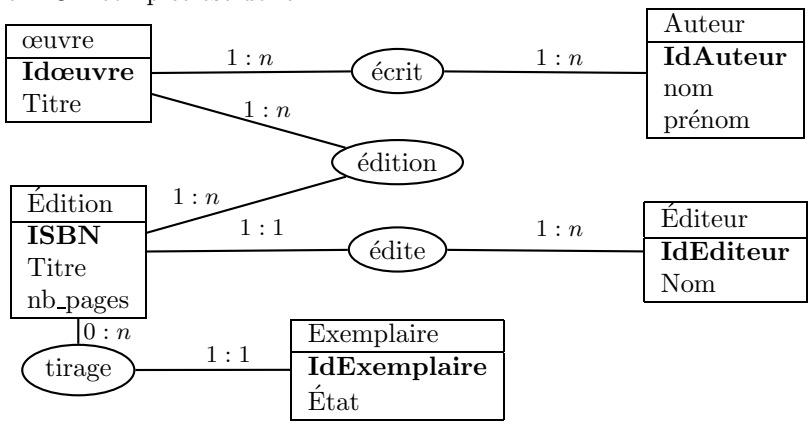
- min_A est le nombre *minimal* de fois où une occurrence de l'entité A participe à une relation du type considéré. Il s'agit en général de 0 ou 1.

- max_A est le nombre *maximal* de fois où une occurrence de l'entité B participe à la relation. Il s'agit en général de 1 ou n (n pour *plusieurs* fois, ou un nombre quelconque de fois).
- min_B et max_B 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 éditer au minimum une édition et au maximum *plusieurs* éditions (un nombre quelconque de fois). Ce qui nous donne :

Le MCD complet est donc :



2.2.5 Cas particuliers et pièges

Quelques points particuliers sont à garder à l'esprit lors de la réalisation d'un MCD.

- Un identifiant est obligatoire pour chaque entité.
- Il ne doit pas y avoir de redondance d'informations : une information quelconque ne doit pas être représentée plus d'une fois dans le MCD.
- Évitez autant que possible les relations entre plus de deux entités. Souvent, il est possible de remplacer la relation par une entité.
- Restez dans la mesure du possible avec des cardinalités de valeurs 0, 1 ou n. Il est de toute manière souvent possible de se ramener à ce cas dans les rares cas où des cardinalités d'un autre type semblent plus naturelles.
- Dans l'idéal, il faut trouver un bon compromis entre niveau de détail et "taille" de la base de données. Il est toujours possible de multiplier les entités, mais il vaut mieux le faire que si cela a vraiment du sens et un intérêt dans le problème. Par exemple, si on a une entité personne, on peut considérer l'adresse comme une entité séparée (reliée à personne par une relation "habite à") ou comme une propriété de la personne (ce qui est fait usuellement). En règle générale, il est plus économique de définir l'adresse comme une propriété, mais dans un cas ou il est courant que des personnes habitent au même endroit, la règle de non-redondance incite plutôt à utiliser une nouvelle entité.

A retenir

La méthode générale de construction du MCD :

1. recherche des *entités*,
2. recherche des *propriétés* (dont la *clé* de chaque entité),
3. recherche des *relations* entre entités,
4. recherche des *cardinalités* (0:1, 1:1, 0:n ou 1:n ?)

2.3 Modèle Logique de données

Une fois le MCD construit, l'étape suivante dans la conception de la base de données consiste 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.

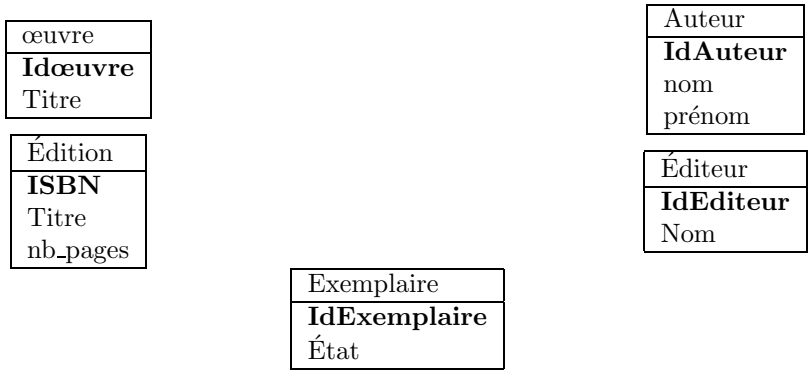
2.3.1 Construction des tables

La première étape consiste à 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 **champ**,
- un **identifiant** devient une **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.



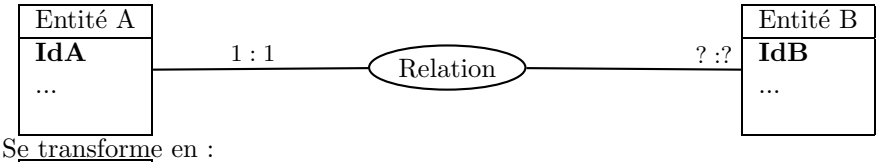
2.3.2 Transformation des relations en liens

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

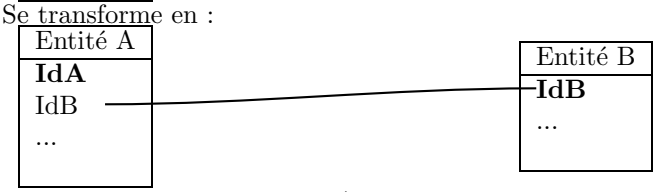
1. l'une des branches de la relation a une cardinalité de 1:1 ou 0:1
2. les autres cas

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 manière suivante :

- On ramène dans la table correspondant à l'entité "du côté du 1:1" (ou du 0:1) la clé primaire de l'autre table ainsi que toutes les éventuelles propriétés de la relations.
- 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 importées du côté du 1:1.



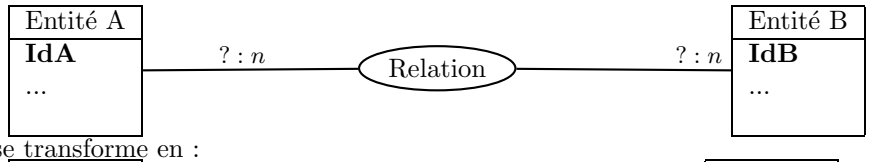
Se transforme en :



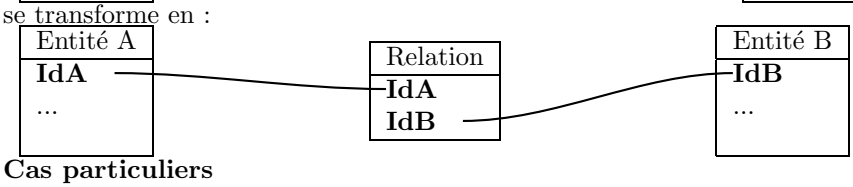
A noter que la clé importée (ici **IdB** qui se retrouve dans **table A**) ne devient pas une clé de la table : c'est une propriété comme une autre. Notons aussi que le lien se fait *entre champs* (on relie IdA à IdB) et non pas, comme dans le MCD, entre les tables.

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 enfin la ou les clés primaires de la nouvelle table. L'idée générale est que chaque occurrence de cette entité doit pouvoir être identifiée de manière unique par ses clés primaires. Cela revient en général à choisir comme clés primaires l'ensemble des clés importés des autres tables.



se transforme en :



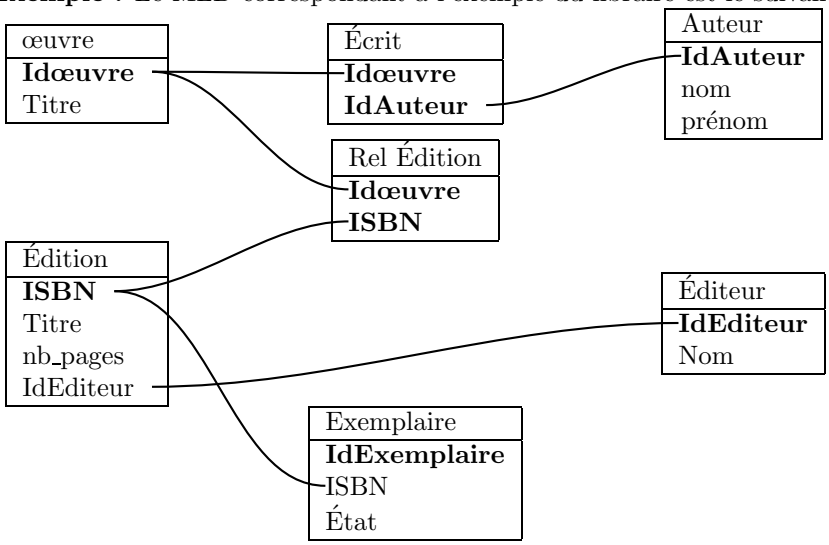
Cas particuliers
Quelques cas particuliers peuvent apparaître.

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

2.3.3 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 :



A retenir

Le méthode de transformation MCD - MLD :

1. les *entités* sont transformées en *tables* (sans modification)
2. les relations sont transformées en fonction de leurs cardinalités
   - pour une relation de type 1:1 - ?:? entre une entité A et une entité B se traduit par une importation de la clé primaire de l'entité B dans la table de A, et on ajoute un lien entre les deux clés,
   - une relation autre (sans 1:1 - ?:?) se transforme en table dans laquelle on retrouve les clés primaires de A et B.
3. les tables inutiles sont supprimés : il s'agit essentiellement des tables à un seul champ (leur clé).

3 Création de la base de données dans Access

Après avoir conçu un MLD, l'étape finale de la méthode Merise 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.

3.1 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. Nous traiterons ici de la création dans **Access**. L'importation sera traitée dans la section suivante. L'objectif final est de créer au sein d'**Access** toutes les tables présentes dans le MLD.

Tout d'abord, sélectionnez "créez une table dans Access" en mode création. La table créée est vide. Il faut maintenant créer tous les champs présents dans le MLD. Pour créer un champ, il suffit de cliquer sur une zone vide dans la liste des champs et de rentrer le nom du champ. Il faut ensuite préciser les différentes propriétés de ce champ présentes soit dans le tableau de la liste des champs, soit dans les onglets en dessous. Les propriétés indiquées d'une astérisque sont obligatoires.

Nom du champ*	le nom du champ, tel qu'il apparaît dans le MLD
Type de donnée*	à choisir parmi (essentiellement) Texte, Numérique, Date/Heure, NuméroAuto (un compteur qui s'incrémente automatiquement, très utile pour les clés), Oui/Non (champ à deux valeur possible : Oui/Non mais aussi Vrai/Faux...) et monétaire.
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 numéro de téléphone, code ISBN ou autre référence. (voir l'aide F1, le bouton ... contient quelques exemples utiles)
Valeur par défaut	Une valeur proposée par défaut pour le champ.
Valide Si	Condition pour lesquelles la valeur entrée par utilisateur est accepté (par exemple "H ou F" pour le sexe... voir F1)
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.
Chaîne 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.

important : si deux champs sont de même natures, ils doivent être de même type et de même taille. Par exemple, dans l'exemple du libraire, le numéro d'éditeur (champ **IdEditeur**) doit être de même type (texte, nombre ou numéroAuto) et de même taille de champ qu'il soit dans la table **éditeur** ou la table **édition**.

Note : Si un champ a un nombre limité de valeurs (par exemples, les catégories sociaux professionnelles, le statut marital, ...), il vaut mieux considérer une champ de type numérique plutôt qu'un champ texte. Préférera stocker 1,2 ou 3 plutôt que "célibataire", "marié" ou "veuf".

La dernière étape consiste à sélectionner la ou les clés de la table. Sélectionner ce ou ces champs (s'il y en plusieurs, vous pouvez les sélectionner en cliquant sur CTRL) et appuyez ensuite sur le bouton représentant une clé.

En pratique

Pour créer une table :

- sélectionnez **Table** dans le menu général, puis **Nouveau**.
- Créez ensuite tous les champs en précisant leur type et leurs propriétés.
- Précisez la clé ou les clés en sélectionnant le champ correspondant et en cliquant sur le bouton représentant une clé dans la barre d'outils.

3.2 Importation de table

La deuxième solution pour créer une table consiste à importer des données déjà existantes dans une autre base de données. Ce peut être (essentiellement) un fichier texte, un classeur Excel ou une table dans une autre base de données **Access**.

- Pour un fichier texte :
  1. s'assurer que le nom du fichier se termine bien par ".txt" ou ".tab", que ses données sont bien disposées en ligne (un enregistrement par ligne) et que sa première ligne est bien soit une ligne de titre, soit une ligne de données (supprimer les éventuelles lignes superflues),
  2. créez la table en passant par **Fichier**,**Données externes**, **Importer**, optez pour **Fichier texte** et sélectionnez votre fichier texte,
  3. choisissez si les champs de votre fichier sont séparés par un délimiteur (un caractère spécial) ou si elles sont de largeurs fixes (chaque champ prend un certain nombre de caractères). S'il s'agit de champs délimités, choisissez le délimiteur (le caractère qui sépare les champs, souvent la tabulation ou la virgule). S'il s'agit de champ de largeur fixe, indiquez quelles sont les zones qu'occupent chaque champ.
  4. Indiquez ensuite le nom et le type de chaque champ. Si la clé primaire n'est pas présente dans votre table, ajoutez la maintenant. Sinon, ne précisez pas de clé tout de suite
  5. indiquez le nom de la table et sauvegardez. La table est maintenant crée.
  6. Modifiez éventuellement la table. En particulier, sélectionnez éventuellement la ou les clé(s) primaire(s).
- Pour un fichier Excel :
  1. s'assurer que les données sont bien sur une seule feuille Excel qui ne contient que ces données avec une éventuelle ligne de titre. Supprimez les données superflues.
  2. Créez la table en passant par **Fichier**,**Données externes**, **Importer**, optez pour **Fichier Excel** et sélectionnez votre classeur Excel, puis la feuille qui contient les données.

3. Pour tous les champs, vérifiez et changez éventuellement leur noms et le type de données. Indiquez également les champs dont vous ne voulez pas dans votre table.
4. Si la clé primaire existe déjà dans la table, sélectionnez la, sinon vous avez la possibilité d'ajouter une clé primaire de type numérotation automatique. Si la table contient plusieurs clés primaires (dans le cas d'une table crée à partir d'une relation du MCD notamment), ne sélectionnez aucune clé pour l'instant.
5. Terminez en donnant un nom à la table, la table est créée.
6. Modifiez éventuellement la table. En particulier, ajoutez les clés primaires le cas échéant.

- Pour un fichier Access :

1. créez la table en passant par **Fichier**,**Données externes**, **Importer**, optez pour **Fichier access** et sélectionnez votre base de donnée,
2. sélectionnez la table qui vous intéresse,
3. modifiez éventuellement la table.

3.3 Relations

La deuxième phase consiste à intégrer dans la base de donnés les relations du MLD. Cela se fait en ouvrant la fenêtre des relations (menu Outils, Relations), en affichant toutes les tables (cf bouton droit) et en dessinant les relations : cliquez sur la clé primaire et faites glisser vers la clé importée.

Access vous demande si vous voulez activer la contrainte d'intégrité de votre base. Une contrainte d'intégrité empêche dans votre base des références à des clés inexistantes. Par exemple, si la relation entre *livre* et *auteur* a une contrainte d'intégrité, il sera impossible d'indiquer à un livre un numéro d'auteur inexistant dans la base. En règle général, activez la contrainte, cela évitera des problèmes dans votre base.

Deux autres options sont indiquées :

- mettre à jour en cascade : Si cette option est cochée et que vous changez la valeur d'une clé (par exemple le numéro d'Auteur dans la table auteur), la valeur de la clé sera modifiée partout (toute référence à l'ancien numéro d'auteur sera remplacée). Inversement, si la case n'est pas cochée, il sera impossible de modifier une clé primaire si elle est utilisée dans d'autres tables.

  En général, cette option peut être activée sans risque.

- Si cette option est cochée et que vous supprimez un enregistrement, tous les enregistrements de la table qui font référence à la clé primaire de cet enregistrement seront supprimés. Dans l'exemple, la suppression d'un auteur entraînerait la suppression de tous ses livres. Inversement, si la case n'est pas cochée, il sera impossible de supprimer un auteur tant que tous ses livres ne sont pas supprimés également.

  A vous de voir dans quels cas la suppression doit se faire automatiquement ou pas. Par sécurité, vous pouvez laisser cette option désactivée.

Un cas particulier peut arriver : le cas des relations *réflexives*. Si une relation existe entre une table et elle même, **Access** vous interdira de la dessiner directement. Il faudra ajoutez une deuxième fois la table (avec le bouton droit de la souris) et indiquer la relation entre la clé importée de la table de départ et la clé primaire de la deuxième table.

En pratique

Pour créer les relations de votre base :

1. allez dans le menu **Outils**, **Relations** pour accéder à la fenêtre représentant les relations,
2. avec le bouton droit de la souris, choisissez **afficher la table** et sélectionnez les tables de votre base,
3. dessinez les relations en faisant glissez les clés primaires vers les clés importées,
4. N'oubliez pas d'activer les contraintes d'intégrités.

4 Requêtes

4.1 généralités

Les requêtes constituent un moyen pratique et automatisé d'interroger ou de modifier la base de données. On distingue deux grands types de requêtes :

- les requêtes de **sélection** : ces requêtes sélectionnent une partie des données de la base, souvent dans le but de les montrer à l'utilisateur. Il s'agit de loin le type de requêtes le plus fréquent ;
- les requêtes d'**action** : ces requêtes modifient la base de données en ajoutant, retirant ou modifiant des enregistrements. On distingue parmi ces requêtes les requêtes de suppression, de mise-à-jour, de création de table ou d'ajout.

Toutes ces requêtes sont construites de manière assez similaire. Nous allons commencer par présenter les requêtes de sélection en montrant les différentes méthodes employées pour créer ces requêtes.

4.2 Requête de sélection

4.2.1 Requêtes mono-table

Une requête mono-table est la requête la plus simple qui soit : on désire sélectionner/montrer les valeurs d'un ou plusieurs champs d'une table. Pour créer une requête de ce type, il faut tout d'abord sélectionner la table qui contient les enregistrements qui nous intéresse, puis choisir parmi les champs de cette table ceux que l'on désire sélectionner.

En pratique

- Dans la partie **Requête** du menu principal, sélectionnez **Nouveau**, puis **Mode création**.
- Sélectionnez la table contenant les informations.
- Sélectionnez les champs qui contiennent les informations qui vous intéressent en double-cliquant dessus. Vérifiez bien que la case **afficher** soit cochée.

4.2.2 Requêtes multi-tables

Une requête concerne souvent plus d'une table. Par exemple, dans le cas du libraire, on peut avoir envie d'afficher la liste des auteurs avec pour chaque auteur, la liste de ses oeuvres. Cette requête fait intervenir deux tables : la table *auteur* et la table *oeuvre*. Il y a ici une relation entre ces deux tables que nous allons exploiter. Nous voulons donc afficher pour chaque auteur X, les oeuvres dont l'auteur est X, par conséquent, il faudra utiliser le lien qui existe entre le champ '`idauteur`' de la table auteur et le champ '`idauteur`' de la table oeuvre.

Remarque : Il est possible dans ce genre de requêtes de faire appel plusieurs fois à la même table. Supposons par exemple que nous disposons d'une entité *employé* qui dispose d'une relation réflexive *responsable* entre *employé* et *employé*. Si on désire la liste de tous les employés avec leurs responsables, il nous faudra utiliser deux fois la table employé dans la requêtes : l'une d'entre elle représentera l'employé, l'autre le responsable.

En pratique

- Dans la partie **Requête** du menu principal, sélectionnez **Nouveau**, puis **Mode création**.
- Sélectionnez les tables intervenant dans la requête.
- Vérifiez que les liens ont tous vraiment du sens dans votre requêtes, supprimez les liens inutiles.
- Sélectionnez les champs à afficher.

4.2.3 Requêtes avec critère

Lorsqu'une requête ne s'intéresse pas à tous les enregistrements d'une table de la base de données, mais seulement à certains, il nous faut utiliser des *critères* pour limiter la sélection. Les critères s'indiquent dans la ligne du même nom.

Supposons que nous nous désirons afficher toute la liste des auteurs dont le nom est "Dupont". Nous allons commencer par écrire une requête qui affichent tous les auteurs, puis nous allons restreindre cette requête en sélectionnant le champ "Nom" et en indiquant dans la zone "Critères" la valeur `="Dupont"` (ou simplement `"Dupont"`).

Access nous permet d'utiliser d'autres opérateurs mathématiques que =.

Opérateur	Signification
=	Égal
<>	Diférent
<	Inférieur
<=	Inférieur ou égal
>	Supérieur
>=	Supérieur ou égal

On peut également utiliser les opérateurs suivants

Opérateur	Signification	Exemple

Entre	Sélectionne les enregistrements pour lesquels la valeur d'un champ est comprise dans un intervalle de valeurs.	`Entre "A" et "C"`
		`Entre 10 et 20`
		`Entre #01/01/04# et #31/12/04#`

Dans	Sélectionne les enregistrements pour lesquels la valeur d'un champs est comprise dans une liste	`Dans("Lille";"Roubaix")`
Est	Sélectionne les enregistrements pour lesquels un champ est vide ou non	`est NULL` `est pas NULL`
Comme	Sélectionne les enregistrements contenant une donnée approximative	Comme "rue*"
Pas	Sélectionne les enregistrements ne correspondant pas au critère	Pas Entre "A" et "C"

Il est possible de faire référence à un autre champ dans les critères. Pour cela, utilisez le nom du champ entre crochets (Exemple : `[Nom]`. Si deux tables possèdent ce champ, on doit alors préciser la table en la mettant également entre crochet et en séparant la table et le champ par un point (exemple : `[Clients].[Nom]`).

Par exemple, la requête suivante sélectionne les livres dont l'auteur est le même que celui spécifié dans la table Auteur.

Champ :	IdAuteur
Table :	Livre
Tri :	
Afficher :	
Critère :	[Auteur].[IdAuteur]

Un certain nombre de fonctions sont également disponibles. En voici quelques unes.

Pour les texte		
Nbcar	nombre de caractère d'un texte	$Nbcar([Nom]) = "4"$ sélectionne les noms de 4 caractères
Droite	retrouve les caractères à droite d'un texte	$Droite([Nom]; 2) = "se"$ sélectionne les noms se terminant par "se"
Gauche	retrouve les caractères à gauche d'un texte	$Droite([Nom]; 2) = "mo"$ sélectionne les noms commençant par "mo"
Extracchaîne	retrouve les sous-chaînes d'un texte	$ExtracChaîne([Nom]; 2; 3) = "ach"$ sélectionne les noms qui contiennent la chaîne "ach" à partir de la deuxième lettre ("machin" par exemple)
Pour les dates et heures		
Mois	permet d'avoir le mois d'une date (il existe aussi jour et année)	$Mois([DateCommande]) = 6$ Affiche les commandes du mois de Juin.
diffdate	permet de faire une différence entre deux dates	$DiffDate("j"; [Date1]; [Date2]) > 100$ retrouve les éléments pour lesquels il y a plus de 100 jours entre date1 et date2 (mettre "m" pour le mois et "aaaa" pour l'année à la place de "j").
AjDate	permet de faire des opérations sur les dates	$AjDate("j"; -10; [DateLivraison])$ fournit une date antérieure de 10 jours à la dte de livraison.
Aujourdhui	permet d'avoir la date du jour (voir aussi **maintenant** pour avoir l'heure actuelle)	$= Aujourdhui()$ permet de sélectionner un champ date dont la valeur correspond à la date du jour.
Pour les champs numériques ou monétaires		
Ent	calcule la partie entière	$Ent([Prix]) = [Prix]$ sélectionne les prix ronds (sans centimes).
Abs	calcule la valeur absolue	$Abs([Variation]) < 10$ sélectionne les variation entre -10 et +10.

Access autorise la combinaison de plusieurs critères. Si ces critères sont tous sur la même ligne, la requête sélectionnera les enregistrements qui répondent à tous les critères à la fois. Si ces critères sont sur des lignes différents (lignes "ou"), la requête sélectionnera les enregistrements correspondant à l'une de ses lignes.

Champ :	Ville	Age
...		
Critères :	"Roubaix"	> 50
Ou :		

Champ	Ville	Age
...		
Critères :	"Roubaix"	
Ou :		> 50

Champ	Ville	Age
...		
Critères :	"Roubaix"	> 50
Ou :	"Lille"	

Ici, par exemple, la première requête sélectionnera les habitants de Roubaix de plus de 50 ans. La deuxième requête sélectionnera les habitants de Roubaix (indifféremment de leur age) et les personnes de plus de 50 ans (même n'habitant pas Roubaix). La troisième requête sélectionne les habitants de Roubaix de plus de 50 ans ainsi que tous les habitants de Lille.

La combinaison de critère peut aussi être réalisée à l'aide des opérateurs booléens : `OU`, `ET` ou `NON`. La requête suivante permet de sélectionner les habitants de Lille et de Roubaix (soit, ceux dont le champ ville vaut "Lille" **ou** "Roubaix")

Champ :	Ville
...	
Critères :	"Roubaix" Ou "Lille"
Ou :	

Enfin, il est possible de demander à l'utilisateur d'entrer un critère. Pour cela, mettre [*truc*] dans les critères (éventuellement dans une formule) où *truc* est un nom quelconque qui ne correspond pas à un nom de champ. L'exemple suivant permet de sélectionner un auteur dont le nom est demandé à l'utilisateur.

Champ :	Nom
Table :	Auteur
Tri :	
Afficher :	
Critère :	[Quel Nom]

4.2.4 Requêtes à champs calculés

Si on désire afficher une information qui n'est pas présente dans la base, mais qui est issue d'un calcul à partir d'autres champs, il est possible d'utiliser des formules pour créer ce nouveau champ. Ce champ s'appelle alors un *champ calculé*. Pour cela, dans la zone de champ, on écrit `nom du champ : formule` où `nom du champ` est le nom que l'on désire donner à notre nouveau champ et `formule` est une formule **Access** (écrite avec les mêmes opérateurs que dans le point précédent).

Par exemple, la requête suivante permet d'avoir l'âge d'une personne.

Champ :	Age : DiffDate("aaaa",[Date],Aujourdhui())
Table :	
Tri :	
Afficher :	
Critère :	

4.2.5 Requêtes de regroupements

La requête de regroupement est un important outil d'analyse et de synthèse. Un regroupement permet de faire un calcul sur un ensemble de valeurs d'un champ éventuellement en fonction d'un autre.

Prenons par exemple les données ci-dessous contentant le chiffre d'affaire de plusieurs agences d'une société à différentes dates.

Date	**Agence**	**CA**
06/01/2003	Nord	927,02 €
06/01/2003	Sud	1 098,46 €
06/01/2003	Est	561,29 €
07/01/2003	Nord	1 385,55 €
07/01/2003	Est	681,09 €
07/01/2003	Sud	1 401,56 €

Pour juger des performances des différentes agences, on peut avoir envie de faire des calculs sur le chiffre d'affaire en fonction de différentes données :

- par date : on peut avoir le chiffre d'affaire total par jour. le "par jour" est un indice fort du regroupement : on fera un regroupement sur la date, et un somme sur les chiffres d'affaires. Notons que la notion d'agence ne nous intéresse pas, elle ne devra pas apparaître dans la requête

- par agence : en sommant les chiffres d'affaires. Le regroupement se fait sur l'agence, on fait une somme sur le CA. Encore une fois la date ne doit pas apparaître dans la requête puisqu'elle ne joue aucun rôle dans le regroupement.

Lorsqu'on fait un regroupement, comme dans les deux exemples ci-dessus, il y a donc deux types de champs qui nous intéressent :

- les champs sur lesquels le regroupement porte (ici date et agence)
- les champs sur lesquels un calcul s'effectue

Tous les autres champs ne nous intéressent pas et n'ont pas à apparaître dans la requête. En particulier, si on désire ajouter une critère dans la requête sur un champ qui n'est ni un champ de regroupement, ni un champ de calcul, cela ne sera pas possible (directement en tout cas, voir les requêtes en cascade).

Différentes opérations sont possibles, en voici quelques-unes :

- **Somme :** effectue la somme de tous les champs
- **Compte :** calcule le nombre de fois où un champ apparaît. A ne pas confondre avec somme, dans les requêtes ci-dessus, il s'agit bien d'une somme de CA. Un compte indiquerait par exemple 3 (il y a 3 chiffres d'affaire pour chaque date).
- **Moyenne**
- **Var, EcartType :** variance et écart type.
- **Minimum, Maximum :** la plus petite/grande valeur
- **Premier, Dernier :** le premier/dernier de la liste (souvent couplé avec des tris)

Notons que bien évidemment la plupart de ces opérations n'ont de sens que sur des données de type numérique ou monétaire.

En pratique

Pour créer une requête de regroupement :

1. Commencez par créer votre requête en ne faisant intervenir que des champs sur lesquels vous ferez une calcul ou un regroupement (si vous avez besoin d'autres champs, il vous faudra plusieurs requêtes, voir la section "requête en cascades"),
2. Cliquez sur le bouton droit de la souris dans la zone de description de votre requête et sélectionnez "Totaux",
3. Pour chacun des champs, choisissez l'opération que vous comptez effectuer, ou "Regroupement" si le champ est un champ sur lequel vous allez faire un regroupement (il est obligatoire de renseigner cette partie : si vous pensez qu'il ne faut rien mettre, c'est probablement que le champ qui vous intéresse n'a pas à apparaître dans la requête).

4.2.6 Requêtes en cascades

Il arrive souvent qu'une requête ne puisse pas réaliser exactement ce que vous vouliez. Dans ce cas, il ne faut pas avoir peur de décomposer cette requête en plusieurs. Il est possible en effet de construire une requête en ouvrant une autre requête à la place d'une table. Pour cela, lors du choix des tables nécessaires pour une requête, choisissez l'onglet "Requête" et sélectionnez la requête dont vous avez besoin.

Une démarche générale pour créer une requête peut maintenant être proposée :

- commencez à concevoir votre requête principale,
- s'il vous manque une information, créez la requête qui indiquera cette information,
- répétez cette opération jusqu'à ce que votre requête principale ait tout ce dont elle a besoin.

Un exemple, supposons qu'une liste de personne soit indiquée avec leur nom, prénom et leur date de naissance. On désire connaître le nom de la ou des personne(s) les plus âgés.

- la requête sera une requête simple avec un critère : le champ nom sera affiché et la date de naissance doit être égale à la plus petite date de naissance.
- L'information "la plus petite date de naissance" n'est pas connue. Pour avoir cette information, il nous faut donc créer une requête : ce sera une requête par regroupement qui fait simplement un "min" sur l'ensemble des dates de naissance.
- Nous pouvons maintenant créer notre requête principale en se basant sur la requête qui nous permet d'avoir la date de naissance la plus ancienne.

Important : Quand vous envisagez de faire des requêtes en cascade, pensez à afficher la clé primaire de vos tables dans toutes vos requêtes : les liaisons entre vos requêtes se feront plus facilement et vous ferez moins d'erreur.

4.2.7 Requête d'analyse croisée

Faire une analyse croisée consiste à effectuer une synthèse des données sur plusieurs niveaux. Cela ressemble à bien des points de vue à la création d'un Tableau Croisé Dynamique (TCD) en **Excel**. Dans **Access**, l'analyse croisée constitue une forme particulière de requête de sélection.

trois colonnes, et possédant des caractéristiques particulières. L'une des colonnes doit comporter des doublons, sur lesquels sera effectuée l'opération de regroupement (la colonne "U" de la figure ci-dessous). Une autre colonne (la colonne "W" de la figure ci-dessous) doit comporter un nombre restreint de valeurs distinctes, qui serviront à créer les nouvelles colonnes.

U	V	W
a		1
a		2
a		3
b		1
b		2
c		1
c		2
c		3

U	W1	W2	W3
a			
b			
c			

On peut ainsi réaliser des tableaux en deux dimensions pour visualiser l'évolution d'une données. Trois types de champs sont importants :

- **entête de ligne :** le champ qui sert à étiqueter les lignes,

- **entête de colonne :** le champ qui sert à étiqueter les colonnes,
- **valeur :** le champ qui va être à l'intérieur du tableau ; il faut lui associer une opération (voir *requêtes de regroupement*).

En pratique

Pour créer une requête d'analyse croisée :

1. créez une requête de sélection contenant les données
2. transformez la en requête d'analyse croisée en allant dans le menu **type de requête** et en sélectionnant **Analyse Croisée**,
3. indiquez quel champs utiliser comme entête de ligne, entête de colonne et valeur (dans *analyse*)

4.2.8 Requêtes en mode SQL

Le SQL (Structured Query Language) est le véritable langage utilisé pour faire des requêtes non seulement par **Access**, mais aussi par la quasi totalité des SGBD.

Il y aurait beaucoup à dire sur le SQL, et nous ne rentrerons pas ici dans les détails. Nous nous bornerons à montrer que chaque requête réalisée graphiquement en **Access** a une traduction en SQL (l'inverse n'est pas vrai), et qu'**Access** permet d'écrire des requêtes en SQL.

Le SQL est un langage assez simple qui permet d'écrire une requête sous forme de texte. Par exemple, la requête SQL suivante permet de sélectionner les noms de l'auteur ayant écrit "Frankenstein".

```
SELECT AUTEUR.nom FROM AUTEUR, OEUVRES where AUTEUR.idAuteur = OEUVRES.idAuteur AND
OEUVRES.titre='Frankenstein';
```

En quelques mots : la requête SQL indique qu'il affiche le champ `nom` de la table `AUTEUR`, qu'il a besoin pour cela des tables `AUTEUR` et `OEUVRES`, qu'il utilise le lien `AUTEUR.idAuteur - OEUVRES.idAuteur` et qu'il a un critère : `OEUVRES.titre='Frankenstein'`.

En pratique

Pour passer en 'Mode SQL' :

- créez une requête normalement ou ouvrez une requête existante ;
- au moyen de l'icône d'affichage (située sous fichier, complètement à gauche de la barre d'outils), sélectionnez le "Mode SQL" ;
- vous pouvez voir la traduction SQL de votre requête, et éventuellement modifier votre requête ;
- la requête modifiée ne peut éventuellement plus être vue dans Access (si elle ne peut pas être représentée graphiquement). On peut toutefois toujours l'exécuter (bouton avec point d'exclamation).

4.3 Requêtes d'actions

Jusqu'à présent, les requêtes que nous avons vu sont toutes des requêtes qui permettent de consulter la base de données. Il est également possible de faire des requêtes qui vont modifier la base de données :

- en supprimant des enregistrement,
- en les modifiant,
- en en ajoutant,
- ou en créant de nouvelles tables.

Nous voyons dans cette section comment réaliser ces différentes opérations. Il s'agit en règle générale de créer une requête de *sélection* (avec les méthodes vues ci-dessus) et en modifiant ces requêtes pour les transformer en requêtes d'*action*.

4.3.1 Requêtes de suppression

Une requête de suppression permet d'effacer des informations de la base de données. La conception d'une requête de suppression se fait très simplement en deux temps :

- création d'une requête de sélection qui va sélectionner les enregistrements à supprimer,
- transformation de la requête en requête de suppression.

Faîtes très attention lorsque vous utilisez ce genre de requête ! Cette opération est irréversible et par le jeu des relations, les suppressions peuvent se répercuter en cascade dans d'autres tables si l'option correspondante a été choisie lors de la création de la relation (voir la section sur l'intégrité réferentielle).

En pratique

Pour concevoir une requête de suppression :

- créer une requête de sélection qui va sélectionner les enregistrements à supprimer,
- transformez la en requête de suppression en allant dans le menu **type de requête** et en sélectionnant **Suppression**,
- vous pouvez passer en mode "feuille de données" pour afficher les enregistrements qui vous être supprimés (sans les supprimer réellement),
- pour exécuter la requête (et supprimer effectivement les enregistrements) appuyez sur le bouton "!" ou allez dans le menu Requête / Exécution, soit dans la liste des requêtes, double-cliquez sur votre requête.

4.3.2 Requêtes de mise à jour

Une requête de mise à jour permet de modifier les valeurs de certains enregistrement de la table. Il s'agit probablement du type de requête d'action le plus courant.

La conception d'une requête de mise à jour se fait en deux temps :

1. Conception de la requête comme d'une requête de sélection. On commence par créer une requête de sélection qui va permettre d'indiquer quels sont les enregistrements qui vont être modifiés.
2. Modification des enregistrement : on indique quels sont les nouvelles valeurs des champs indiqués.

Attention : il n'est pas possible d'utiliser des critères dans une requête de mise à jour. Si vous avez besoin d'utiliser des critères, utilisez alors des requêtes en cascade : une requête de sélection choisit les enregistrements qui vous intéressent, et une requête de mise à jour ouvre la requête de sélection et fait les modifications.

En pratique

Pour faire une requête de mise-à-jour :

1. créez une requête en mode sélection qui affiche les champs qui seront modifiés,
2. transformez cette requête en requête de mise à jour à partir du menu **Type de requête**,
3. Indiquez pour champs, sa nouvelle valeur (qui peut éventuellement être une formule).
4. testez en passant en mode feuille de données et exécutez avec le bouton "!",

4.3.3 Requêtes de création de table

La requête de création de table se fait également très simplement. Le résultat d'une requête de sélection peut se voir sous la forme d'une table : la requête de création de table crée simplement cette table réellement.

Il est à noter que ce type de requête ne devrait servir que très rarement, essentiellement lorsqu'il s'agit de modifier une base de données ou de créer une base de données à partir d'une autre (en vue de faire de l'importation de données).

En pratique

Pour faire une requête de création de table :

1. créez une requête en mode sélection qui affiche les informations qui seront présentes dans votre table,
2. transformez cette requête en requête de création de table à partir du menu **Type de requête**,
3. testez en passant en mode feuille de données et exécutez avec le bouton d'exécution ("!"),
4. à l'exécution le système demandera un nom de table.

4.3.4 Requêtes d'ajout

Ce type de requête ajoute un groupe d'enregistrements d'une ou plusieurs tables à la fin d'une ou de plusieurs autres tables. Sa création est également très aisée :

- on crée une requête de sélection comprenant les enregistrement à ajouter,
- on transforme la requête en requête d'ajout,
- et à l'exécution de la requête, on spécifie la table à laquelle sont ajoutées les données.

Il faut bien sûr que la requête de sélection ait bien les mêmes champs que la table à laquelle les données seront ajoutées.

En pratique

Pour faire une requête d'ajout :

1. créez une requête en mode sélection qui affiche les informations qui seront ajoutées,
2. transformez cette requête en requête d'ajout à partir du menu **Type de requête**,
3. testez en passant en mode feuille de données et exécutez avec le bouton d'exécution ("!"),
4. à l'exécution le système demandera le nom de table à laquelle seront ajoutées les données.

5 Finalisation de la base

Une fois la base de données terminée, l'utilisateur final ne devrait pas avoir à manipuler directement les différentes tables et requêtes. Il 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.

5.1 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 imprimés.

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 très rapide d'états, mais sans aucune option de personnalisation. L'assistant colonnes permet une présentation sous forme de colonnes avec à gauche le nom du champ et à droite sa valeur, l'assistant tableau génére une présentation proche de celle des tables (noms des champs en haut puis la liste des valeurs).
- **Assistant graphique** : pour créer un graphique (non abordé ici).
- **Assistant Etiquette** : génère des états sous forme d'étiquettes. Non abordé ici, la création se fait cependant d'une manière très similaire à celle des états classiques.

Nous verrons ici uniquement l'utilisation de l'assistant état. Celui-ci nous pose différentes questions :

- **Choix des champs :** tout d'abord, sélectionnez une table ou une requête puis faites glisser les champs qui seront affichés dans l'état. Notez que si les champs qui vous intéressent appartiennent à plusieurs tables ou requêtes, alors il vaut mieux commencer par créer une requête qui regroupe ces différentes informations et bâtir l'état sur cette nouvelle requête.
- **Niveau de regroupement :** si l'état nécessite un regroupement, c'est ici qu'on l'indique. Par exemple, si on désire la liste des livres par auteurs, nous choisirons un regroupement par auteur (sur IdAuteur). Notez que si tous les champs ne sont pas à leur bon niveau de regroupement, ce sera rectifiable par la suite (par exemple, le regroupement par IdAuteur laissera le nom de l'auteur au même niveau que les informations sur les livres, ce sera rectifiable plus tard).
- **Le tri :** on choisit les champs à partir duquel les lignes de détails seront triés (les lignes de détails sont les lignes qui seront affichés pour chaque enregistrement).
- **La présentation :** on choisit ici le modèle de présentation de l'état.
- **Le style :** on choisit les différents éléments de style (couleur,...) parmi une liste de quelques choix.
- **Nom de l'état :** le nom avec lequel l'état sera stocké.

L'état est ensuite crée.

5.2 Modification de l'état

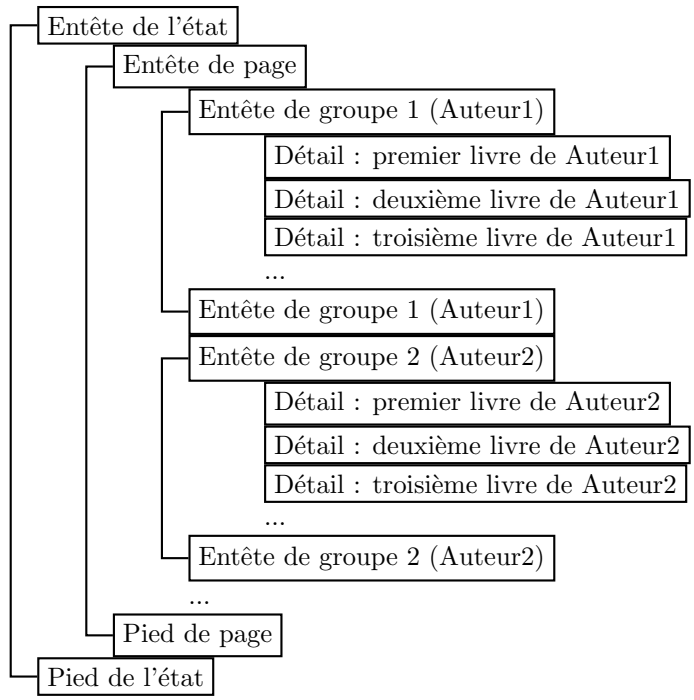
Un état étant crée, il est possible de le modifier. Pour cela, cliquez sur le bouton "Modifier" après avoir sélectionné votre état (ou passez en mode "Création").

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 affiché au début de chaque page
- **l'entête de groupe :** c'est ce qui va être imprimé au début de chaque groupe, un 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.

Notez qu'il peut y avoir plusieurs groupes (s'il y avait plusieurs niveaux de regroupements) et qu'il est possible que certaines section indiqués ici manquent (pied de groupe est souvent absent), on peut choisir les sections à afficher à l'aide du bouton "Trier/Regrouper".

Par exemple, si on utilise un état qui affiche tous les auteurs, avec tous leurs livres (avec donc un regroupement sur **Auteur**), l'impression du document se fait ainsi :



A partir de là, il est possible de faire différents types de modifications :

- déplacer un élément, y compris d'une section à une autre,
- supprimer des éléments,

- ajouter des éléments de décoration (lignes, boîtes, images, textes), pour cela appuyer sur le bouton "boîte à outils",
- les sauts de pages peuvent également être ajoutés (voir propriété des différentes sections, en général dans "pied de groupe").

5.3 Création d'un formulaire

Un formulaire permet de modifier et d'afficher le contenu d'une 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 peut nous permettre de manipuler au même endroit des informations provenant de plusieurs tables simultanément. Par exemple, dans la base de données du libraire, il est possible au sein d'un même formulaire d'ajouter ou modifier les informations sur un auteur et sur l'ensemble des oeuvres qu'il a écrit.

La création d'un formulaire se fait d'une manière très similaire à la création d'état. Après avoir fait "nouveau formulaire", plusieurs choix vous sont proposés.

- **Mode création :** la création se fait sans aucune aide.
- **Assistant formulaire : Access** nous guide pas à pas dans la création du formulaire
- **formulaires instantanés (Colonnes / Tableau / Feuille de données) :** crée un formulaire rapidement pour une table, sans poser de questions. L'aspect est rudimentaire et ne donne pas de grandes possibilités de personnalisations
- **Assistant graphique et tableau croisé dynamique :** non abordé ici.

Si vous créez un formulaire en utilisant la méthode "Assistant formulaire", **Access** vous demande les renseignements suivants :

- **La table :** quelle est la table qui contient les informations que l'on souhaite modifier.
- **Les champs :** quels sont les champs que nous désirons afficher dans le formulaire (en général tous, mais pas toujours. Il n'est par exemple pas nécessairement utile de choisir la clé si celle-ci est issu d'un champ de type NumératoAuto).
- **La présentation :** la présentation générale de l'état, les plus souvent utilisés étant "Colonne simple" et "Tabulaire".
- **Le style :** le style général (couleur, fond...).
- **Le nom :** le nom de l'état.

Une autre méthode pour créer le formulaire consiste à utiliser l'un des assistants "formulaire instantané". Pour cela, à la création, choisissez l'un de ces assistants plutôt que l'assistant standard :

- l'assistant **formulaire instantané : Colonnes** crée un formulaire standard avec un enregistrement par feuille,
- l'assistant **formulaire instantané : Tableau** crée un formulaire sous la forme d'un tableau reprenant l'ensemble des enregistrements,
- l'assistant **formulaire instantané : Feuille de données** permet la création immédiate de formulaire reprenant la même présentation que les feuilles de données (c'est-à-dire moche).

L'assistant vous demande simplement la table ou la requête à partir de laquelle vous voulez créer votre formulaire et le crée instantanément. Ceci a l'avantage de permettre une création très rapide mais ne propose pas beaucoup de possibilité de personnalisation.

5.4 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 "boîte à outils",

Lorsqu'on ajoute un élément, il est possible de les ajouter directement pour ensuite modifier l'élément, ou, plus simplement, en utilisant un assistant. Pour cela, sélectionnez sur la boîte à outils le bouton représentant une "baguette magique" avec de créer l'élément. On trouve les éléments suivants :

- **intitulé :** il s'agit d'une zone de texte qui ne sert qu'à la présentation.

  Pour le créer, sélectionnez le bouton correspondant ("*Aa*") et cliquez sur le formulaire à l'endroit où vous voulez placer votre intitulé et écrivez votre 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, sélectionnez le bouton correspondant ("ab|") et cliquez sur le formulaire à l'endroit où vous voulez placer votre zone de texte. Si vous avez activé l'assistant ("baguette magique"), on vous demandera alors à quel champ doit correspondre la zone de texte. Sinon, dans la zone de texte, indiquez le champs qui vous intéresse (`"[salaire]"` par exemple, ou `"=[salaire] + [commission]"` si vous voulez une formule).

- **bouton bascule, bouton d'option et case à cocher :** ils fonctionnent de la même manière que les zones de textes, mais pour des champs de type Oui/Non.

- **groupe d'options :** un groupe d'options vous 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).

  Pour le créer, utilisez l'assistant, sélectionnez le bouton correspondant et cliquez sur le formulaire à l'endroit où vous voulez placer votre groupe d'options. L'assistant vous demande quelles sont les différentes étiquettes de votre liste ("patron", "cadre moyen", "ouvrier",...). Il s'agit de la valeur affichée dans le groupe d'options. On vous demande ensuite une éventuelle valeur par défaut, puis la valeur correspondante à chacune des étiquettes. Il s'agira de la valeur contenue réellement dans le champ. On vous demande ensuite le champ correspondant au groupe d'options (celui dans lequel l'information sera stockée) et le style (bouton ou case). La dernière question concerne la légende de votre groupe d'options.

- **liste déroulante :** permettent le choix d'une valeur dans une liste.

  Pour la créer, utilisez l'assistant, sélectionnez le bouton correspondant et cliquez sur le formulaire à l'endroit où vous voulez placer votre liste déroulante. De là, trois choix vous sont proposés : le premier indique que la liste est remplie à partir des valeurs d'une table ou d'une

requête, la deuxième correspond à des valeurs remplies "à la main", et la troisième offre une fonction de recherche que nous n'aborderons pas ici.

Si on choisit le premier choix, l'assistant commence par demander la source des informations (une table ou une requête). Il demande ensuite le ou les champs concernés. Il est possible de mettre plusieurs champs : le champ qui contient réellement l'information à stocker et un ou plusieurs autres champs plus parlants qui serviront à renseigner l'utilisateur. On souhaite ainsi souvent faire une liste sur un ensemble de clés primaires (IdAuteur par exemple), alors qu'on préfère afficher d'autres informations (nom et prénom de l'auteur). Si on a choisit plusieurs champs, l'assistant demande s'il faut masquer ou non le premier champ. Il demande ensuite la largeur souhaitée pour la liste et enfin le champ qui contiendra les données de la liste.

Avec le second choix, la démarche est similaire. On commence là par remplir la liste des valeurs.

- **bouton de commande :** un bouton de commande permet de réaliser certaines opérations par le biais d'un bouton. Il est ainsi possible de passer d'un enregistrement à un autre, d'effectuer des recherches, d'ouvrir un autre formulaire ou un autre état.

  Pour le créer, utilisez l'assistant, sélectionnez le bouton correspondant et cliquez sur le formulaire à l'endroit où vous voulez placer votre bouton. l'assistant vous demandera ensuite quel opération vous voulez réaliser avec votre bouton et la présentation qu'il aura.

- **boîte, trait, image :** il s'agit d'éléments purement graphiques qui peuvent améliorer la présentation de votre formulaire.

- **Onglet :** ce contrôle offre deux avantages. Il regroupe les informations à une même place et prend moins d'espace à l'écran. Il suffit de cliquer sur l'onglet pour voir les informations de la catégorie choisie.

- **Sous formulaire / Sous état :** cet élément permet d'intégrer un sous formulaire. Ce point sera abordé plus loin.

- **autre contrôles :** Access contient également d'autres contrôles qui permettent d'autres types de saisie, par exemple le contrôle Calendrier pour les dates

5.5 Sous formulaire

Les sous formulaires sont un peu aux formulaires ce que sont les regroupements aux états. Ils permettent d'éditer plusieurs tables en même temps. Supposons par exemple que, dans l'exemple du libraire, on désire avoir un formulaire qui affiche tous les renseignements concernant les auteurs (nom, prénom, ...) ainsi que la liste des livres qu'ils ont écrits. L'opération peut se faire "à la main" ou à l'aide de l'assistant. Nous présentons ici deux méthodes, une simple et rapide, une un peu moins simple (mais tout de même) qui permet de faire plus de choses.

Méthode simple

La création du formulaire se fait de la manière suivante.

1. Créez une requête qui contient les informations que devra contenir le formulaire (dans l'exemple, il s'agira d'une requête contenant tous les auteurs et tous les livres de ces auteurs).

2. Créez un formulaire en se basant sur cette requête à l'aide de l'assistant.

3. Suivez l'assistant. La procédure est la même que pour un formulaire simple à une nuance près. On vous demande comment vous souhaitez afficher vos données, choisissez le niveau de regroupement voulu (comme pour un état). Choisissez "formulaire avec sous formulaire".

Méthode un peu moins simple (mais tout de même)

La méthode présentée ici a le même résultat que ci-dessus mais permet éventuellement plus de souplesse. On peut par exemple en suivant cette méthode avoir plusieurs sous-formulaires au sein d'un même formulaire.

Nous reprenons ici le même exemple : on désire tous les auteurs avec tous les livres de chaque auteur.

1. créez la requête du sous-formulaire : il s'agira ici d'une requête affichant tous les livres avec leur auteur (IdAuteur). Les données qui nous intéresse sont effectivement les informations sur les livres, et le champ IdAuteur nous permettra de faire le lien avec le formulaire principal.

2. Créez le sous formulaire : il s'agira d'un formulaire classique sur la requête précédente. Ce sera ce formulaire qui sera intégré dans le formulaire principal.

3. Créez le formulaire principal.

4. Modifiez ce formulaire : à l'aide du bouton "Sous formulaire" de la barre d'outils, importez le sous formulaire (en ayant activé la "baguette magique"). L'assistant vous demandera comment lier le sous-formulaire avec le formulaire : dans notre exemple, il s'agit de faire un lien par le champ IdAuteur (on affichera ainsi uniquement les livres dont l'auteur est celui du formulaire).

5.6 Menus et autres personnalisations

5.6.1 menu général

Une étape importante de la finalisation de la base de donnée est la conception d'un menu. Ce menu permettra à l'utilisateur de retrouver facilement les différents éléments dont il a besoin sans avoir à manipuler directement la base de donnée. Idéalement, l'utilisateur doit pouvoir réaliser toutes les opérations dont il a besoin à partir du menu. La méthode est la suivante.

1. Commencez par réfléchir à ce que vont contenir toutes les différentes pages de votre menu.

2. Créez votre menu principal. Pour cela, aller dans "Outils/Gestionnaire du menu général". Acceptez d'avoir un menu général.

3. Vous avez maintenant accès à une fenêtre dans laquelle se trouve un élément (menu général). Dans cette fenêtre se trouveront toutes les pages de votre système de navigation. Créez-les (avec Ajouter)

4. Vous pouvez maintenant les modifier. Chaque page de menu contient différents éléments. Chaque élément sera représenté par un bouton qui effectuera une action précise. En particulier, on peut :

   - aller sur une autre page du menu,
   - ouvrir un état,
   - ouvrir un formulaire en mode création (s'ouvre sur une page blanche) ou en mode modification (s'ouvre sur le premier enregistrement),
   - Quitter l'application...

Attention : Access semble être très sensible aux manipulations sur son menu. En théorie, vous pouvez consulter le menu qui est présent dans les formulaires. Évitez de trop modifier ce menu, et surtout ne l'effacez pas sous peine de voir **Access** se "planter" ou vous empêcher de créer un nouveau menu...

5.6.2 Démarrage de l'application

On peut ensuite personnaliser d'avantage notre application en indiquant ce que va voir l'utilisateur lorsqu'il va démarrer notre base de données. Pour cela, allez dans `outils`$\rightarrow$`démarrage`. De là on peut (entre autres) :

- indiquer le nom du menu qui est affiché lors du démarrage de l'application,
- l'icône de l'application,
- le nom de l'application