

Activité

Vous avez besoin de sauvegarder vos contacts et vous jugez judicieux de le faire en utilisant un classeur Excel :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	carnet d'adresses														
2	Nom	prenom	Adresse	code postal	ville	Telephone	Portable								
3	Tankeu	Romualde	Makepe	237	Douala	24345637333	674323470								
4															

Problèmes :

Tankeu a deux numéros portable

Elle a deux adresses (adresse, code postal, ville)

Possibles solutions

Ajouter une colonne intitulée Portable 2

Ajouter trois colonnes (adresse2, code postal2, ville2)

Question :

Comment indiquer quelle est l'adresse de la maison et celle du travail ? Une base de données gère très facilement cela.

id_client	Nom	prénom
1	Tankeu	Romualde

id_adresse	nom	Adresse	code postal	ville	#id_client
1	domicile	Makepe	237	Douala	1
2	maison	Village	237	douala	1

id_telephone	nom	numero	id_client
1	téléphone	243223454	1
2	portable maison	674323470	1
3	portable bureau	654365435	1

Nouveau problème :

J'ai Douala deux fois et écrits de façon différente. Si je fais une recherche des adresses dans « Douala » une seule va apparaître.

id_client	Nom	prénom
1	Tankeu	Romualde

id_adresse	nom	Adresse	code postal	# id_ville	#id_client
1	domicile	Makepe	237	1	1
2	maison	village	237	1	1

id_telephone	nom	numéro	id_client
1	téléphone	243223454	1
2	portable maison	674323470	1
3	portable bureau	654365435	1

id_ville	nom
1	Douala

Problématique : Comment enregistrer Tankeu Rita (la fille de Tankeu Romualde) en sachant qu'elle vit à la même adresse que sa mère.

id_client	Nom	prénom
1	Tankeu	Romualde
2	Tankeu	Rita

id_adresse	nom	Adresse	code postal	# id_ville
1	domicile	Makepe	237	1
2	maison	village	237	1

id_telephone	nom	numéro	id_client
1	téléphone	243223454	1
2	portable maison	674323470	1
3	portable bureau	654365435	1

id_habiter	#id_adresse	#id_client
1	1	1
2	1	1
3	2	2

id_ville	nom
1	Douala

Nous avons ainsi structure nos données, non pas pour avoir un fichier mais une base de donnée.

Introduction

Avant de réaliser un projet informatique, il faut pratiquer une analyse. Cette analyse consiste à comprendre et modéliser les systèmes d'information sur lequel on travaille. Pour obtenir une bonne communication, les informaticiens utilisent une méthode d'analyse.

Exemple : MERISE: Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'entreprise.
UML: Unified Modeling Language

Quel que soit la méthode, la conception de la base de données passe par l'élaboration de certains diagrammes normalisés tels le MCD.

D'autre part, un système d'information nécessite une structure pour conserver les données: Un fichier, Une base de données.

1. Système d'information

On appelle « système d'information » l'ensemble composé par les bases de données, le SGBD utilisé et les programmes associés.

Une base de données est un ensemble structuré de données enregistrées dans un ordinateur et accessibles de façon sélective par plusieurs utilisateurs.

Un SGBD (Système de Gestion de Base de Données) est un ensemble de programmes assurant la gestion et l'accès à une base de données.

2. Différences entre un fichier et une base de données

Dans un fichier, les informations sont stockées de manière séquentielle (les uns à la suite des autres). Cette méthode de stockage est très différente de celle des bases de données. On peut citer comme différences entre un fichier et une base de données:

- Le stockage interne : une base de données est composée de plusieurs fichiers virtuels appelés tables;
- La méthode de stockage : un fichier est très lié aux programmes qui permettent de le manipuler tandis qu'une base de données peut être manipulée par divers programmes;(pour accéder a un fichier faut l'ouvrir et le fermer à la fin)
- L'absence de redondance : dans une base de données, la même information est stockée une et une seule fois ;
- La recherche : pour retrouver une information dans un fichier, il faut parcourir toutes celles qui la précèdent tandis qu'on peut accéder directement à n'importe quelle information stockée dans une base de données.

3. Type de base de données

On distingue plusieurs types de base de données :

- **Hiérarchiques ou réseaux** : ici les enregistrements sont liés les uns aux autres ;
- **Relationnelles** : ici les tables sont reliées les unes aux autres par des champs communs. Ce sont les plus courantes ;
- **Objet** : Les données y sont représentées en tant qu'instances de classes hiérarchisées. Chaque champ est un objet. De ce fait, chaque donnée est active et possède ses propres méthodes d'interrogation et d'affectation;

Le SGBD est composé de trois couches successives:

– **Le système de gestion de fichiers.**

Il gère le stockage physique de l'information.

– **Le SGBD interne.**

Il organise le placement et l'assemblage des données, gère les liens et l'accès rapide.

– **Le SGBD externe.**

Il se charge de la présentation et de la manipulation des données aux concepteurs et utilisateurs. Il gère les langages de requêtes élaborées et les outils de présentation, tels que les états et formulaires.

4. Conception d'une base de donnée

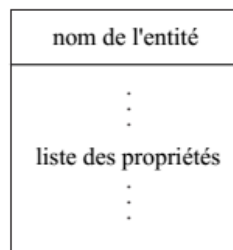
4.1.Définitions

- L'entité est la représentation, dans le système d'information, d'un objet matériel ou immatériel ayant une existence propre et conforme aux choix de gestion de l'entreprise. C'est une population homogène d'individus. Par contre les articles et les clients ne peuvent pas être groupes : leurs informations ne sont pas homogènes.

Par exemple : les articles vendus par une entreprise peuvent être groupes dans une même entité **Articles**, car d'un article à l'autre les informations ne changent pas de nature (désignation, prix unitaire,...).

- Un attribut est une propriété (donnée élémentaire et indécomposable) d'une entité ou d'une association.
Exemple : le **prix unitaire** est un attribut de l'entité **articles**, le **nom** de famille est un attribut de l'entité **clients**...

L'entité est composée de propriétés. Par exemple, une personne, une voiture, un client, un projet sont en général des entités.



- Identifiant de l'entité

Il s'agit de l'attribut (ou groupe d'attributs) qui identifie de manière unique une entité (le premier généralement).

- Une association est une liaison qui a une signification précise entre plusieurs entités.

Le nombre d'intervenants dans cette association caractérise sa dimension :

- réflexive sur une même entité ;
- binaire entre deux entités ;
- ternaire entre trois entités ;
- -n-aire entre n entités.



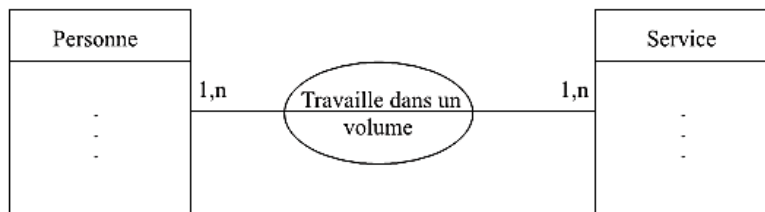
Des attributs peuvent être attachés aux associations. Par exemple, un employé peut passer 25 % de son temps dans un service, et 75 % de son temps dans un autre. L'association « travaille dans » qui relie une personne à un service portera dans ce cas la propriété « volume de temps passé ».

- Les cardinalités caractérisent le lien entre une entité et une association. La cardinalité d'une association est constituée d'une borne minimale et d'une borne maximale :

Syntaxe : (min,Max)

- minimale : nombre minimal de fois qu'une occurrence d'une entité participe aux occurrences de l'association, généralement 0 ou 1 ;
- maximale : nombre maximal de fois qu'une occurrence d'une entité participe aux occurrences de l'association, généralement 1 ou n.

Les cardinalités maximales sont nécessaires pour la création de la base de données. Les cardinalités minimales sont nécessaires pour exprimer les contraintes d'intégrité.



- Une contrainte d'intégrité est une contrainte sur les données de la base, qui doit toujours être vérifiée pour assurer la cohérence de cette base.

Les systèmes d'information sont souvent remplis de telles contraintes ; Par exemple :

- un identifiant doit toujours être saisi ;
- le salaire doit être compris entre 4 000F et 100 000F ;
- Un enseignant doit dispenser au moins un cours ;
- l'emprunteur d'un livre doit être un abonné du club ;

4.2. Conception d'un MCD (Modèle conceptuel de données).

Exercice 1 :

On souhaite mémoriser une partie des données d'un championnat international d'athlétisme comprenant des courses et des concours de lancer et de saut. Le domaine concerné est décrit ci-dessous :

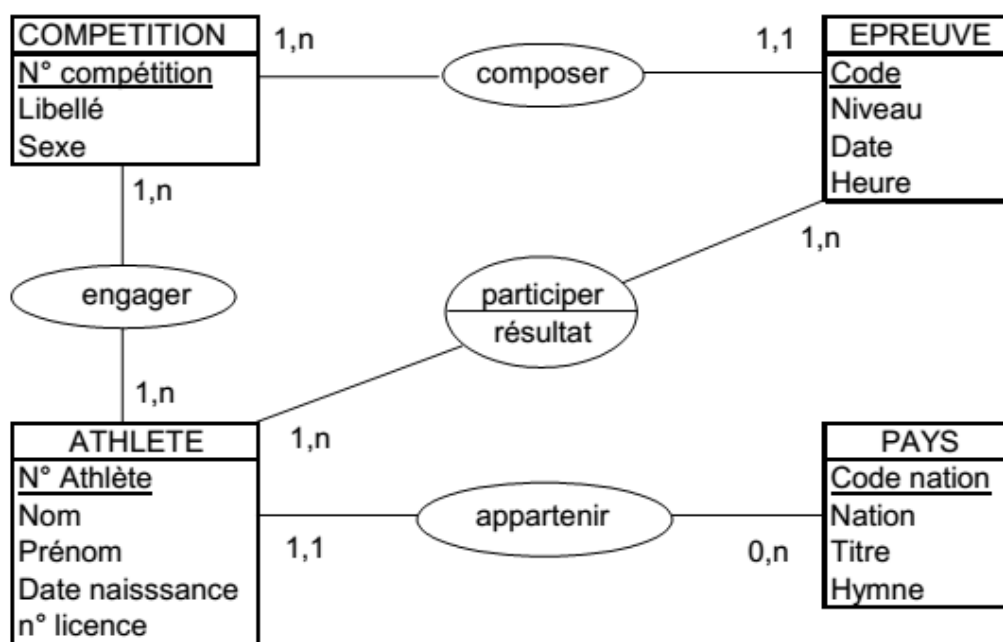
Le championnat est divisé en compétitions (100 m homme, 800 m femme, disque homme...) décrites par un numéro, un nom et le sexe des compétiteurs. Une compétition peut comprendre plusieurs épreuves de niveaux différents (finale, demi-finale... éliminatoires).

Chaque épreuve est décrite par un code, un niveau, une date et une heure de début. Les athlètes (n° d'athlète, nom, prénom, date de naissance, n° licence) appartiennent à différents pays (code-nation, nation, titre, hymne nationale). Ils sont engagés dans différentes compétition et participent à des épreuves dont les résultats sont mémorisés.

Interprétation du texte :

- « Le championnat est divisé en compétitions [...] décrites par un numéro, un nom et le sexe des compétiteurs » ⇒ création d'une entité COMPETITION avec les attributs numéro compétition, nom compétition, sexe.

- « Une compétition peut comprendre plusieurs épreuves » ⇒ création d'une entité EPREUVE associée à l'entité COMPETITION.
- « plusieurs épreuves de niveaux différents » ⇒ création d'une entité NIVEAU associée à l'entité EPREUVE.
- « Chaque épreuve est décrite par un code, un niveau, une date et une heure de début » ⇒ création des attributs code, niveau, date, heure de début dans l'entité EPREUVE.
- « Les athlètes [...] appartiennent à différents pays[...] » ⇒ Création d'une entité ATHLETE avec les attributs se trouvant dans la première paire de crochets, et associée à une nouvelle entité PAYS ayant pour attributs le contenu de la seconde paire de crochets.
- « Ils [les athlètes] sont engagés dans différentes compétitions » ⇒ création d'une association entre l'entité ATHLETE et l'entité COMPETITION.
- « Ils [les athlètes] [...] participent à des épreuves dont les résultats sont mémorisés » ⇒ création d'une association entre l'entité ATHLETE et l'entité EPREUVE.



4.3. Passage du MCD au MLD (plus tard aux tables relationnelles)

Tableau de conversion : les éléments de la colonne de gauche deviennent ceux de la colonne de droite.

MCD	MLD
entité	table
Propriété de l'entité	Colonne ou attribut(ou champs) de la table
Identifiant de l'entité	Clé primaire de la table
Relation(ou association) (x,n) a (x,n) avec x=0 ou 1	table
Relation (ou association) (x,1) a (x,n) avec x=0 ou 1	La clé primaire de la table (père) portant(x,n) migre vers la table(fils) portant(x,1)
Relation (ou association) (x,1) a (x,1) avec x=0 ou 1	FAIRE DISPARAITRE la table inutile

- Entité devient Table : Ensemble de données organisées suivant un modèle en colonnes (correspondant aux propriétés du sujet) et en lignes (correspondant aux occurrences du sujet) ;

- Attribut ou propriété d'entité devient Champ de table: colonne d'une table.
- Enregistrement: ligne d'une table;
- L'association inter-entité devient la Relation (entre les tables): Association de deux tables par la mise en commun d'un champ ;
- Clé primaire: Champ dont les valeurs permettent d'identifier de manière unique un enregistrement dans une table. Il ne peut être vide ;
- Clé étrangère : Clé primaire d'une table qui est attribut dans une autre table

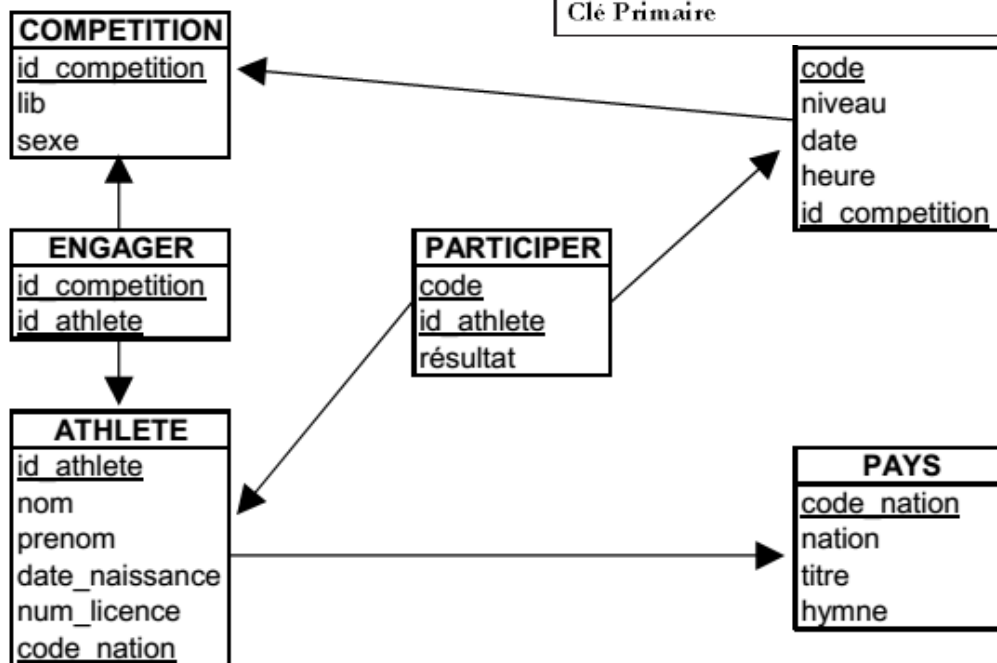
Champs ou attributs	
CodeMat	NomMat
C01	Mathématiques
C02	Français
C03	Histoire
C04	EPS
C05	Informatique

Clé Primaire

Enregistrements

MLD Exercice 1 :

MLD Le championnat d'athlétisme



- L'association participer entre les entités ATHLETE et EPREUVE est une relation «plusieurs à plusieurs» qui est, par conséquent, devenu la table PARTICIPER dans le MLD. Cette table possède une clé constituée des clés des tables ATHLETE et EPREUVE, soit id_athlete et code. Elle possède une colonne résultat qui correspondait à l'attribut porté de l'association participer dans le MCD.
- L'association engager entre les entités COMPETITION et ATHLETE est une relation «plusieurs à plusieurs »qui est devenu la table ENGAGER dans le MLD. Cette table possède une clé constituée des clés des tables ATHLETE et EPREUVE, soit id_athlete et code.
- L'association composer entre les entités COMPETITION et EPREUVE est une relation « binaire à plusieurs ». Au moment du passage du MCD au MLD, l'association provoque l'exportation de la clé id_competition dans la table EPREUVE.
- L'association appartenir entre les entités PAYS et ATHLETE est une relation «binaire à plusieurs ». Au moment du passage du MCD au MLD, l'association provoque l'exportation de la clé code_nation dans la table ATHLETE.

MLD final :

competition (**id_competition** , libelle, sexe)
epreuve (**code** , niveau , date_epreuve , heure , # **id_competition**)
pays (**code_nation** , nation , titre , hymne)
athlete (**id_athlete** , nom , prenom , date_naissance , num_licence , #**code_nation**)
participer (#**code** , #**id_athlete**)
engager (#**id_competition**, #**id_athlete**)

4.4. Passage du MLD au MPD (Modèle physique des données)

Le Diagramme Physique des Données sera assimilé au script de création des tables du MLD. Il sera écrit en langage SQL (Structured query Language).

5. SGBD

5.1. Fonctions d'un SGBD

Il existe plusieurs SGBD sur le marché, mais tous (ou presque) remplissent les fonctions suivantes :

- **La définition des données** : elle permet la création et la mise à jour des structures de la base (tables, champs, etc.).
- **La manipulation des données** : elle permet de manipuler les enregistrements de la base et de les mettre à jour. Elle permet aussi de rechercher des informations en interrogeant la base ;
- **Le contrôle de l'accès aux données** : il permet de définir des droits d'accès aux différents utilisateurs de la base.

Exemple de SGBDR : Oracle, SQL Server, DB2, Access, MySQL, ...

Toutes ces fonctions peuvent être implémentées de deux façons :

- Au travers des programmes associés aux SGBD via des interfaces graphiques (assistants et mode création)
- A travers des requêtes écrites en langage SQL.

5.2. Le langage SQL

SQL (Structured Query Language) est un langage de commandes conçu en 1970 par la société IBM. Il permet la manipulation de la structure et des données des bases de données. SQL est présent dans presque tous les SGBD et est de ce fait un standard en termes de création et d'administration des bases de données.

Une commande SQL est appelée « requête ». Les requêtes sont regroupées en LDD, LMD et LCD. Si les requêtes de LDD et de LCD sont gérées par le créateur de la base de données, seules les requêtes de manipulation de données sont employées par les utilisateurs

5.3. Syntaxe SQL

SQL n'est pas sensible à la casse sauf pour ce qui est des accents. On préférera donc écrire sans accent. Toutefois, il est utile de mettre en majuscule les MOTS-CLES du langage. Par contre, toute instruction SQL se termine par un point-virgule.

5.4. Commandes SQL dans Access

Création en SQL

- Création d'une base de données
CREATE DATABASE nom_de_la_BD ;

- Création d'une table

```
CREATE TABLE Nom_de_la_table
(Nom_du_champ1 Type_de_donnée [NOT NULL AUTO_INCREMENT],
Nom_du_champ2 Type_de_donnée,
CONSTRAINT PRIMARY KEY (cle_primaire)
CONSTRAINT FOREIGN KEY (cle_etrangere)
REFERENCES (Nom_de_la_table_contenant_la_cle_etrangere));
```

Les types de données peuvent être :

- **INTEGER** : Ce type permet de stocker des entiers signés (+ ou -) codés sur 4 octets.
- **BIGINT** : Ce type permet de stocker des entiers signés codés sur 8 octets.
- **REAL** : Ce type permet de stocker des réels comportant 6 chiffres significatifs codés sur 4 octets.
- **NUMERIC** [(n, [d])] : nombre de n chiffres [d (optionnel) après la virgule]
- **CHAR** (longueur) : Ce type de données permet de stocker des chaînes de caractères de longueur fixe. longueur doit être inférieure à 255, sa valeur par défaut est 1.
- **VARCHAR** (longueur) : Ce type de données permet de stocker des chaînes de caractères de longueur variable. longueur doit être inférieure à 2000, il n'y a pas de valeur par défaut.
- **DATE** : Ce type de données permet de stocker des données constituées d'une date.
- **TIMESTAMP** : Ce type de données permet de stocker des données constituées d'une date et d'une heure.
- **TEXT** : Ce type de données permet de stocker des chaînes de caractères de longueur variable.

Exercice 2 :

Ecrire le code SQL qui permettra de créer la base de données et les tables du MLD obtenu plus haut.

6. Microsoft Access

Microsoft Access comme tout SGBD permet de créer, modifier et manipuler une base de données. Pour remplir toutes ces fonctions, Access le fait à travers des objets : Tables, formulaires, états, requêtes et macros.

En effet, Microsoft Access crée une base de données, permet de visualiser, de saisir des données, d'effectuer des calculs et d'imprimer des résultats.

Si les formulaires permettent une présentation élégante des données à l'écran, ils ne sont pas adaptés à la présentation papier. Pour analyser les données et les mettre en page pour l'impression, on utilise un état. Celui-ci permet bien sûr l'affichage de plusieurs enregistrements simultanés mais aussi le calcul de cumuls, de regroupements et l'affichage de synthèses. Il permet aussi l'élaboration de graphiques de présentation, la gestion d'étiquettes ou de publipostages.

Ce sont les formulaires et états qui forment la partie visible d'une base de données

Access fournit des assistants qui génèrent automatiquement des formulaires et des états sans que le concepteur n'ait à connaître quoi que ce soit en programmation.

6.1.Création d'une base de données

Lorsque vous ouvrez Ms Access une fenêtre s'affiche vous proposant de créer une nouvelle base de données ou d'ouvrir une base de donnée existante. La base de données sous Access 2003 est un fichier qui porte l'extension .mdb, sous Access 2007 nous avons l'extension .accdb.

La procédure de création d'une base de données simple avec Access 2007 est la suivante :

- Lancer Access 2007
- Cliquer sur base de données vide
- Choisir l'emplacement où sera enregistré le fichier
- Saisir le nom du fichier.
- Cliquer sur créer.

6.2. Les objets dans Access

La fenêtre d'un SGBD contient les composants ou objets et pour accéder à un objet, il faut cliquer sur le bouton correspondant à sa catégorie. Ces objets sont :

- **Les tables**

C'est le point central de la base de données, car toutes les données vont être stockées dans les tables. Chaque table est identifiée par un nom et est composée de champs ou attributs (éléments caractéristiques d'une table)

- **Le formulaire**

Le formulaire permet d'entrer de façon conviviale l'information dans une ou plusieurs tables

- **Les requêtes**

L'objet requête permet d'afficher des informations répondant à des questions spécifiques. Les données d'une requête proviennent d'une ou de plusieurs tables, de requêtes ou d'une combinaison des deux.

- **Les états**

Ils permettent d'imprimer les données contenues dans une table selon une présentation définie et y intégrant éventuellement des calculs

- **Une macro**

Une macro permet d'automatiser certaines tâches dans Access ; elle peut simuler une suite d'actions qui auraient dû être faites par l'utilisateur .

6.2.1. Les tables

Dans une base de données simple, on peut avoir une seule table. Pour la plupart des bases de données, nous aurons besoin d'utiliser plusieurs tables.

La table est une collection de données relatives à un domaine bien défini exemple les élèves d'un établissement.

a) La procédure de création d'une table est la suivante :

- Cliquer sur le menu créer
- Cliquer sur création de table
- Saisir les différents champs (nom du champ, type de données, description). les champs sont les attributs des tables.
- Enregistrer la table

Exercice3 : créer les tables suivantes

Elèves
<u>Matricule</u>
Noms
Prénoms
Date de naissance
Classe

Matières
<u>Code</u>
Mathématiques
Français
TQG
Economie
Français
Matricule

Remarque :

Dans la table élèves, matricule est la clé primaire (elle se crée à partir du menu création). C'est un identifiant qui permet de distinguer de façon unique un enregistrement.

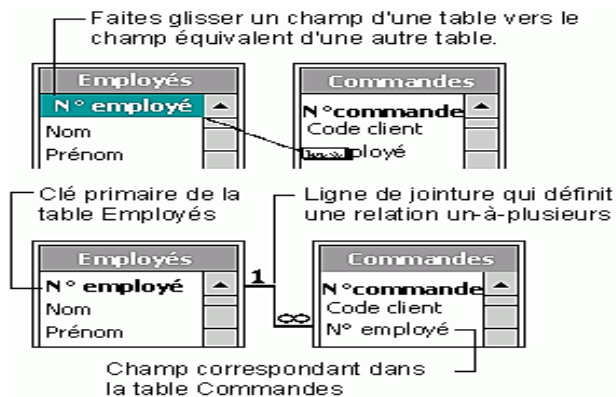
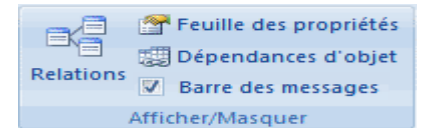
La création ou la modification d'une table se fait en mode création

Un champ correspond à une colonne de la table

b) Définition des relations entre les tables

Ici il est question de créer des relations entre les tables

- Sous l'onglet Outils de base de données du groupe
- Afficher/Masquer, cliquez sur Relations.
- **Sélectionnez plusieurs tables ou requêtes, puis cliquez sur Ajouter.** Une fois que vous avez terminé d'ajouter des tables et des requêtes dans l'onglet de document Relations, cliquez sur Fermer.
- **Faites glisser un champ (généralement une clé primaire) d'une table au champ commun (clé étrangère) dans l'autre table. Pour faire glisser plusieurs champs, appuyez sur CTRL, cliquez sur chaque champ, puis faites-les glisser.**
- Une boîte de dialogue Modifier des relations s'affiche.



Modifier des relations

Table/Requête : Clients Table/Requête liée : Commandes

ID Client ID

☒ Appliquer l'intégrité référentielle

☐ Mettre à jour en cascade les champs correspondants

☐ Effacer en cascade les enregistrements correspondants

Type de relation : Un-à-plusieurs

OK Annuler Type jointure... Nouvelle relation...

- Vérifiez que les noms de champ affichés sont des champs communs pour la relation. Si un nom de champ est incorrect, cliquez dessus et sélectionnez le champ approprié dans la liste.
- Pour appliquer l'intégrité référentielle pour cette relation, activez la case à cocher Appliquer l'intégrité référentielle.
- Cliquez sur Créer.

Access dessine une ligne de relation entre les deux tables. Si vous avez activé la case à cocher Appliquer l'intégrité référentielle, la ligne est épaissie à chaque extrémité. En outre, si vous avez activé la case à cocher Appliquer l'intégrité référentielle, le chiffre 1 apparaît au-dessus de la partie épaissie sur un côté de la ligne de relation, et le symbole infini (∞) apparaît au-dessus de la partie épaissie sur l'autre côté, comme illustré ci-dessous.

c) Saisie des informations dans une table

Les informations d'une table sont saisies en mode feuille de données.
Saisir les informations suivantes dans la table Elèves

Table : Elèves				
Noms	Prénoms	Date de nais	Classe	matricule
Tagne	Jean	01/05/1995	Tle	CPSM01
Bissoko	Henri	05/11/2002	6 ^e	CPSM02
Uche	Willy	30/03/1996	Tle	CPSM03

6.2.2. Les formulaires

Un formulaire est un objet de base de données qui permet d'entrer, de modifier ou d'afficher les données d'une table ou d'une requête. Les formulaires permettent également de contrôler l'accès aux données en n'affichant, par exemple, que certains champs ou lignes de données. Par exemple, il est possible que certains utilisateurs n'aient besoin d'afficher que certains champs d'une table constituée d'une multitude de champs. En

leur proposant un formulaire qui ne contient que ces champs, il leur sera plus facile d'utiliser la base de données. Vous pouvez également ajouter des boutons et d'autres fonctionnalités à un formulaire afin d'automatiser les opérations fréquemment effectuées.

a) Créer un formulaire à l'aide de l'outil Formulaire


L'outil Formulaire permet de créer facilement et rapidement un formulaire. Lorsque vous utilisez cet outil, tous les champs de la source de données sous-jacente sont placés dans le formulaire. Vous pouvez commencer à utiliser le nouveau formulaire immédiatement, comme vous pouvez le modifier en mode Page ou Création pour l'adapter à vos besoins.

Utiliser l'outil Formulaire pour créer un formulaire :

- Dans le volet de navigation, cliquez sur la table ou la requête qui contient les données que vous souhaitez faire figurer dans le formulaire.
- Sous l'onglet **Créer**, dans le groupe **Formulaires**, cliquez sur **Formulaire**.

b) Créer un formulaire à l'aide de l'Assistant Formulaire

Pour être plus sélectif quant aux champs à faire figurer sur votre formulaire, vous pouvez utiliser l'Assistant Formulaire à la place des divers outils de création de formulaires mentionnés précédemment

Sous l'onglet **Créer**, dans le groupe **Formulaires**, cliquez sur **Plus de formulaires**, puis sur **Assistant Formulaire**. 

- Suivez les instructions fournies dans les pages de l'Assistant Formulaire.

REMARQUE Si vous souhaitez inclure des champs de plusieurs tables ou requêtes dans votre formulaire, ne cliquez pas sur **Suivant** ou sur **Terminer** après avoir sélectionné les champs de la première table ou requête dans la première page de l'Assistant Formulaire. Au lieu de cela, répétez les étapes afin de sélectionner une table ou une requête, puis cliquez sur les champs supplémentaires que vous souhaitez inclure dans le formulaire. Cliquez ensuite sur **Suivant** ou sur **Terminer** pour continuer.

- Dans la dernière page de l'Assistant, cliquez sur **Terminer**.

c) Insertion du bouton « OK » et le bouton « FERMER »

Pour insérer le bouton OK ou FERMER, on peut suivre la procédure suivante.

- Ouvrir le formulaire en mode création
- Dans le groupe contrôle de l'onglet création, vérifier que le bouton **utiliser les assistants contrôle** est activé.
- Cliquer sur bouton (bouton de contrôle formulaire)
- Cliquer et glisser sur une zone libre du formulaire puis suivre l'instruction de l'assistant bouton de commande

d) Insertion d'un sous formulaire

On a procédure suivante

- Choisir la 1ère table et les champs voulus
- Choisir la 2ème table et faire de même
- Sélectionner l'affichage souhaité (par élèves ou par matières)
- Indiquer l'affichage du sous-formulaire
- Choisir un style
- Donner un nom au formulaire et sous formulaire puis cliquer sur Terminer

La procédure d'insertion d'un sous formulaire dans un formulaire existant est la suivante

- Ouvrir le formulaire principal en mode création
- Agrandir la zone *Détail* en faisant glisser le *Pied de formulaire*
- cliquer sur **Sous-formulaire/Sous-état**, présent dans le groupe **Contrôles** de l'onglet **Création**
- Remplir la boîte de dialogue qui s'affiche en fonction des consignes de l'exercice.
- Cliquer sur terminer.

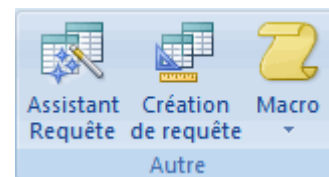
6.2.3. Les REQUETES

Une base de données contient des informations qui lorsqu'elles sont nombreuses leur gestion devient assez laborieuses. L'objet requête apporte une solution car il permet d'afficher des informations répondant à des questions spécifiques. Les données d'une requête proviennent d'une ou de plusieurs tables, de requêtes ou d'une combinaison des deux.

a) Création d'une requête simple en mode création

La procédure est la suivante :

- Sous l'onglet **Créer**, dans le groupe **Autre**, cliquez sur **Création requête**.
- Dans la boîte de dialogue **Afficher la table**, double-cliquez sur les tables desquelles vous souhaitez extraire des données. Chaque table s'affiche sous forme de fenêtre dans la section supérieure du Concepteur de requêtes. Cliquez sur **Fermer** une fois l'ajout de vos tables terminé.
- Dans chaque table, double-cliquez sur chacun des champs que vous voulez utiliser dans votre requête. Chaque champ apparaît dans une cellule vide de la ligne **Champ** de la grille de création.
- Ajoutez éventuellement des critères à la ligne **Critères** de la grille de création. (Comme F*), < 12



b) Création d'une requête simple en mode SQL

Les requêtes SQL permettent de manipuler les données dans une base de données. La création d'une requête peut se faire grâce aux instructions suivantes :

1. **Drop** : permet de supprimer une base de données ou une table.

Syntaxe :

DROP DATABASE nom_BD ;

DROP TABLE nom_table ;

2. **Alter** : permet de modifier une table

Syntaxe :

ALTER TABLE nom_table ADD nom_du_champ Type_de_donnees ; (permet d'ajouter un champ a une table)

Syntaxe :

ALTER TABLE *nom_table* **MODIFY** *nom_du_champ* *Type_de_donnees* ;

(Permet de modifier le type de données d'un champ dans une table)

Syntaxe :

ALTER TABLE *nom_table* **DROP** *nom_du_champ* ;

(Permet de supprimer un champ dans une table)

3. **Update** : permet de créer une requête de mise à jour qui modifie les valeurs des champs d'une table spécifiée, selon des critères déterminés.

Syntaxe :

UPDATE *nomtable*

SET *nouvellevaleur*

WHERE *critères* ;

Table : nom de la table contenu les données à modifier

Nouvellevaleur : valeur à insérer dans un champ particulier des enregistrements mis à jour.

Critères : expression qui détermine quels enregistrements sera mise à jour

4. **SELECT** : permet d'interroger une base de données

La syntaxe minimale d'une instruction select est la suivante :

SELECT *nomchamps* **FROM** *nomtable* ; Affiche tous les enregistrements (certains champs) d'une table

Ou

SELECT * **FROM** *nomtable* ; Affiche tous les enregistrements (tous les champs) d'une table.

Pour une sélection répondant à un critère, on a la syntaxe suivante :

SELECT *nomchamps*

FROM *nomtable*

WHERE *critères* ;

Exemple

SELECT noms, prénoms

FROM participants

WHERE noms like "N*";

Cette requête affiche les noms et prénoms des participants dont les noms débutent par N.

5. **INSERT TO** : permet d'insérer les valeurs dans les champs d'une table. La syntaxe est la suivante

INSERT INTO *nomtable* (*nomchamps1*, *nomchamps2*, *nomchamps3*,)

VALUES (*valeurchamps1*, *valeurchamps2*, *valeurchamps3*,);

6.2.4. Les Etats

Les états permettent d'imprimer les données contenues dans une table selon une présentation définie et y intégrant éventuellement des calculs.

a) Création d'un état à l'aide de l'outil état

L'outil État est le moyen de création d'état le plus rapide, car la génération d'état est automatique, sans demande d'informations. L'état affiche tous les champs provenant de la table ou de la requête sous-jacente.

La procédure est la suivante :

- Dans le volet de navigation, cliquez sur la table sur laquelle vous voulez baser le rapport ou la requête.
- Sous l'onglet **Créer**, dans le groupe **États**, cliquez sur **État**.
- Access crée et affiche l'état en mode Page.



Une fois l'état affiché, vous pouvez l'enregistrer, puis fermer, soit l'état et les données sous-jacentes, soit la requête utilisée comme source d'enregistrement. À la prochaine ouverture de l'état, Access affichera les données les plus récentes de la source d'enregistrement.

b) Création d'un état à l'aide de l'assistant état

Vous pouvez utiliser l'Assistant État pour affiner la sélection des champs constitutifs de votre état. Vous pouvez également spécifier le mode de regroupement et de tri des données, et utiliser des champs provenant de plusieurs tables ou requêtes, à condition d'avoir défini au préalable les relations entre les tables et les requêtes.

La procédure est la suivante

- Sous l'onglet **Créer**, dans le groupe **États**, cliquez sur **Assistant État**
- Suivez les instructions de l'Assistant État. Sur la dernière page de l'Assistant, cliquez sur **Terminer**.

REMARQUE

Si vous souhaitez inclure des champs provenant de plusieurs tables et requêtes dans votre état, ne cliquez pas sur **Suivant** ou **Terminer** après la sélection des champs de la première table ou requête, dans l'Assistant État. Répétez les étapes de sélection d'une table ou d'une requête, et cliquez sur les champs supplémentaires à inclure dans l'état. Cliquez ensuite sur **Suivant** ou sur **Terminer** pour continuer.

Exercice 4:

Créer une table unique Agenda dans laquelle on trouvera notamment les nom, prénom, adresse, code postal, ville, téléphone, portable, fax, email, remarques, deuxième adresse, etc., pour chaque individu.

Créer un formulaire présentant des données qui soient accessibles uniquement en lecture, puis un formulaire de saisie et un formulaire de modification.

Vous proposerez ensuite un état détaillant toutes les données et un état restreint sous format d'étiquette par exemple, ne présentant que les nom, prénom et adresse.

Vous terminerez enfin par un menu général d'appel des formulaires et états créés.

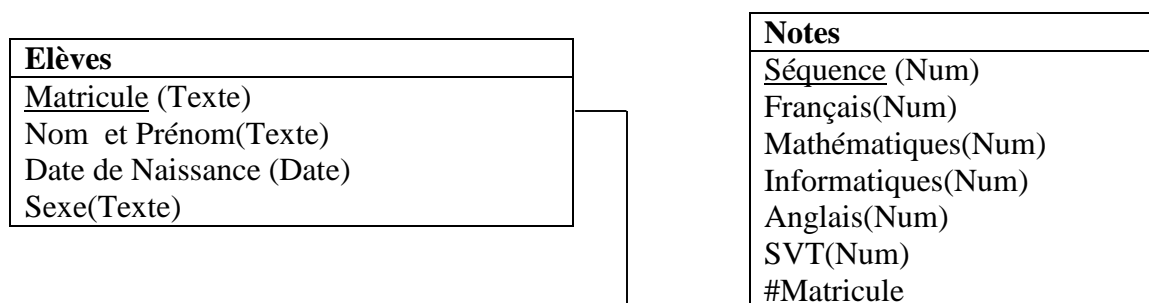
Exercice 5 :

Un établissement scolaire de la place vous demande de créer une base de données qui gère les relevés de notes des élèves. Votre travail consiste à créer les tables élèves et notes.

Travail à faire

1. Lancer Ms Access 2007
2. Créer la base de données qui porte votre nom ou le groupe de travail
3. À l'aide du Modèle logique de Données (MLD) de la **figure 1**, créer les différentes tables

Figure 1



4. Créer les relations dans Access
5. Insérer les enregistrements suivants

Matricule	Nom et Prénom	Date de Nais	Sexe	Français	Maths	Info	Anglais	SVT
Mat4011	Um Jeannette	15/07/1998	Fém.	10.5	12	09	12	08.5
Mat2442	Voffou	05/10/2000	Masc.	08	10	15	09	12

	Sylvestre							
Mat4111	Essomba Antoinette	20/08/2001	Fém.	11	11.5	12.5	08	11.5
Mat6019	Oumarou Ali	25/02/1990	Masc.	12	8.5	13	10.5	09.5
Mat0213	Dakole Ibrahim	21/11/1999	Masc.	7.5	11.5	10	10	11

6. Créer les requêtes suivantes

- 6.1. La requête nommée « **Garçons** » qui sélectionne tous les élèves de sexe Masculin.
- 6.2. La requête nommée « **filles** » qui sélectionne tous les élèves de sexe féminin
- 6.3. La requête nommée « **note maths** » qui sélectionne les élèves et leurs notes de mathématiques.

Corrigé exercices:

1. CREATE TABLE competition (id_competition NUMBER(6) NOT NULL, libelle VARCHAR2(20), sexe CHAR(1) NOT NULL, CONSTRAINT competition_pk PRIMARY KEY (id_competition));
2. CREATE TABLE epreuve (code NUMBER(6) NOT NULL, niveau NUMBER(2) NOT NULL, date_epreuve DATE, heure DATE, id_competition NUMBER(6) NOT NULL, CONSTRAINT epreuve_pk PRIMARY KEY (code), CONSTRAINT epreuve_id_competition_fk FOREIGN KEY(id_competition) REFERENCES competition(id_competition));
3. CREATE TABLE pays (code_nation NUMBER(6) NOT NULL, nation VARCHAR2(20), titre VARCHAR2(20), hymne VARCHAR2(20), CONSTRAINT pays_pk PRIMARY KEY(code_nation));
4. CREATE TABLE athlete (id_athlete NUMBER(6) NOT NULL, nom VARCHAR2(15), prenom VARCHAR2(15), date_naissance DATE, num_licence NUMBER(12), code_nation NUMBER(6) NOT NULL, CONSTRAINT athlete_pk PRIMARY KEY (id_athlete), CONSTRAINT athlete_code_nation_fk FOREIGN KEY(code_nation) REFERENCES pays(code_nation)) ;
5. CREATE TABLE participer (code NUMBER(6), id_athlete NUMBER(6), CONSTRAINT participer_pk PRIMARY KEY (code, id_athlete), CONSTRAINT participer_id_athlete_fk FOREIGN KEY(id_athlete) REFERENCES athlete(id_athlete), CONSTRAINT participer_code_fk FOREIGN KEY (code) REFERENCES epreuve(code));
6. CREATE TABLE engager (id_competition NUMBER(6), id_athlete NUMBER(6), CONSTRAINT engager_pk PRIMARY KEY (id_competition, id_athlete), CONSTRAINT engager_id_athlete_fk FOREIGN KEY(id_athlete) REFERENCES athlete(id_athlete), CONSTRAINT engager_id_competition_fk FOREIGN KEY (id_competition) REFERENCES competition(id_competition));