

## Objectif

- Décrire l'intérêt des Bases de Données ( BD) et des Systèmes de Gestion des Bases de Données ( SGBD )
- Utiliser les fonctions de base d'un SGBD pour gérer les bases de données
- Utiliser les fonctions de base d'un SGBD pour charger des données, interroger une base de données et imprimer des données.

## Plan du chapitre

- I- Introduction générale
- II- Rappel de quelques concepts de base d'une base de données
- III- Réalisation d'une base de données
  1. Les tables
  2. Les relations
- IV- Création de requêtes
  1. Les requêtes de sélection
  2. Les requêtes paramétrées
  3. Les requêtes d'ajout
  4. Les requêtes de mise à jour
  5. Exercice d'application
- V- Création de formulaires
- VI- Création et impression d'états
- VII- Exercices
- VIII- Lecture

Le chapitre "**BASES DE DONNEES**" du manuel scolaire de la 3ème année, a présenté une introduction aux bases de données avec quelques concepts de manipulation de base d'une base de données.

Dans ce chapitre, en plus d'un bref rappel sur les notions qui ont été abordées en 3ème année, on va approfondir l'étude de ce chapitre en présentant les points suivants : la création d'une base de données, la création des requêtes, la création des formulaires et la création et l'impression d'états.

## I. Introduction générale

Au niveau du programme de la 3<sup>ème</sup> année on a introduit quelques concepts de bases de données et on a créé des exemples simples de bases de données. Dans ce qui suit on va étudier, de façon approfondie, ces concepts et on va créer des bases de données à travers des exemples du monde réel.

**Base de données :** Une base de données, comme on la définit, est un ensemble structuré de données enregistré sur des mémoires secondaires créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs.

Les bases de données contiennent des informations nécessaires au fonctionnement des entreprises, des sociétés et des établissements (gestion comptable, gestion commerciale, gestion de production, gestion des stocks,...)

**Système de Gestion de Bases de Données (SGBD) :** c'est l'outil permettant de créer de nouvelles bases de données de mettre à jour ces données (ajouter, modifier, supprimer), de rechercher efficacement des données spécifiques et de fournir différentes manières de les visualiser (à l'écran ou sur papier).

**Les SGBD Relationnels :** En 1970, Codd, mathématicien, chercheur chez IBM, propose le Modèle Relationnel, basé sur le concept de Relations de la théorie des ensembles. Les données sont organisées sous forme de tableaux de valeurs (Tables) indépendants (plus de pointeurs). Il définit l'Algèbre Relationnelle sur laquelle est basé SQL (Structured Query Language), le langage standard de manipulation (LMD) et de description des données (LDD) de tous les SGBD Relationnels actuels.

Les principaux SGBD Relationnels :

- Oracle, DB2, Sybase, Ingres, Informix, SQL Server sur mini et gros systèmes
- Access, Paradox, FoxPro, FileMaker, 4D sur micro
- MySQL, MSQL, Postgres, InstantDB freewares et sharewares

## II. Rappel de quelques principaux concepts d'une base de données

### 1. Modèle relationnel

Le modèle relationnel proposé est souvent considéré comme le plus simple et le plus élégant des modèles. Sa simplicité est due à une vision tabulaire des données très intuitive. Son élégance résulte de l'utilisation de bases formelles issues de la théorie mathématique des ensembles.

Les objectifs du modèle relationnel étaient différents des autres modèles ; parmi les lacunes que E.F. Codd souhaitait en apporter une solution on en retient deux :

- Permettre un haut degré d'indépendance entre les applications (programmes, interfaces) et la représentation interne des données (fichiers, chemins d'accès)
- Etablir une base de données solide pour traiter les problèmes de cohérence et de redondance des données.

Les données, dans le modèle relationnel, sont organisées sous forme de tableaux de valeurs appelés Tables.

**Exemple de table** : table **ELEVE** dans une base de données «GESTION DE NOTES»

ELEVE				
ID_ELEVE	nom	prénom	date de naissance	classe

La table **ELEVE** est conçue en colonnes, appelées attributs ou champs, et en lignes, appelées enregistrements ou n-uplets.

- Les champs de la table **ELEVE** sont : **ID\_ELEVE**, **nom**, **prénom**, **date de naissance** et **classe**.
- Le champ **ID\_ELEVE** représente l'identifiant unique des enregistrements de la table, on l'appelle clé primaire.

## 2. Schéma d'une table

Le schéma d'une table est donné par l'ensemble de ses attributs.

Le schéma de la table **ELEVE** est :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

### Activité : 1

Prenons l'exemple de la base de données «GESTION DE NOTES» dans l'application scolarité. Les tables de cette base de données peuvent être les suivantes :

ID_ELEVE	nom	prénom	date de nais	classe
E001	Ben Salem	Nizar	09/09/1987	4 Eco 2
E002	Kefi	Rym	17/03/1986	4 Eco 2
E003	Abrougui	Amal	03/02/1988	4 Eco 1
E004	Abassi	Maroua	28/11/1989	4 Eco 1
E006	Abbassi	Maram	03/12/1989	
E007	Boughanmi	Ahmed	03/06/1989	

Figure :  
1

	MAT_MATERE	libellé matière	coefficient
+	M01	Mathématiques	2
+	M02	informatique	1
+	M03	Gestion	2
+	M04	Economie	2

Figure : 2

ID_ELEVE	MAT_MATERE	note
E001	M01	15
E001	M03	13,25
E002	M02	10
E003	M01	10

Figure : 3

Donner les schémas des tables **MATIERE** et **NOTE**.

### Solution

Le schéma de la table **MATIERE** est :

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

et le schéma de la table **NOTE** est :

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

### 3. Schéma d'une base de données

Le schéma d'une base de données est l'ensemble des schémas des tables de la base.

Le schéma de la base de données «**GESTION DE NOTES**» est :

### Exemple

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

### III. Réalisation d'une base de données

#### Activité : 2

Soit à réaliser la base de données «GESTION DE NOTES», décrite avec le schéma suivant :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

#### 1. Les tables

##### a) Définition

Une entité équivaut à un sujet dans le domaine des bases de données. Des éléments distincts, tels que des élèves, les notes des élèves ou des matières, sont des exemples d'entités. Dans une base de données, chaque entité est représentée par une table.

Une table est une collection de données relatives à un sujet spécifique du monde réel. Ces données sont organisées en colonnes et en lignes :

- Les colonnes, appelées souvent **champs**, définissent la structure de la table.
- Les lignes, appelées souvent **enregistrements**, constituent les données de la table.

ID_ELEVE	nom	prénom	date de nais	classe
E001	Ben Salem	Nizar	09/09/1987	4 Eco 2
E002	Kefi	Rym	17/03/1986	4 Eco 2
E003	Abrougui	Amal	03/02/1988	4 Eco 1
E004	Abassi	Maroua	28/11/1989	4 Eco 1
E006	Abbassi	Maram	03/12/1989	
E007	Boughanmi	Ahmed	03/06/1989	

Figure : 4

#### Constatations

On remarque que les données d'une colonne ont le même type et la même signification :

- La colonne **ID\_ELEVE** contient des données de la même forme texte de quatre caractères,
- La colonne **date de naissance** contient des données de type date avec la même forme 'jj/mm/aaaa'.

*b) Les champs*

Chaque table contient un certain nombre de champs appelés aussi attributs. Les champs représentent les caractéristiques des éléments qu'on souhaite stocker dans la table. Par exemple, dans la table **ELEVE**, on peut enregistrer l'**ID\_ELEVE**, le nom, le prénom, la date de naissance et toutes les informations propres à chaque élève.

La table **ELEVE** admet cinq champs : **ID\_ELEVE**, nom, prénom, date de naissance et classe.

Lors de la création d'un champ, on doit spécifier son nom, son type et ses propriétés.



	Nom du champ	Type de données
	ID_ELEVE	Texte
	nom	Texte
	prénom	Texte
	date de naissance	Date/Heure
	classe	<div style="border: 1px solid black; padding: 2px;">                     Texte                      Mémo                      Numérique                      Date/Heure                      Monétaire                      NuméroAuto                      Oui/Non                      ...                 </div>

Figure : 5

Pour chaque champ, on spécifie une liste de propriétés qui varient selon le type du champ ; en effet, pour le champ date de naissance les propriétés sont spécifiées de la façon suivante :

Général	Liste de choix
Format	Date, abrégé
Masque de saisie	Date, général 19/06/1994 17:34:23
Légende	Date, complet dimanche 19 juin 1994
Valeur par défaut	Date, réduit 19-juin-94
Valide si	Date, abrégé 19/06/1994
Message si erreur	Heure, complet 17:34:23
Null interdit	Heure, réduit 05:34 pm
Indexé	Heure, abrégé 17:34
IME Mode	Aucun contrôle
IME Sentence Mode	Aucun

Figure : 6

### c) Les clés primaires

Dans la table **ELEVE** ci-dessus, le champ **ID\_ELEVE** identifie un élève de façon unique. Tous les autres champs contiennent les informations propres à chaque élève. Par exemple, l'**ID\_ELEVE** d'un élève détermine de façon unique son nom et sa classe ; alors que deux élèves peuvent avoir le même nom et la même classe, mais ils ne peuvent pas avoir le même **ID\_ELEVE**. L'**ID\_ELEVE** d'un élève est présenté dans la table sous une autre forme (par exemple en majuscule) pour indiquer qu'il s'agit d'un identificateur.

Dans ce cas, l'**ID\_ELEVE** représente la **clé primaire** de la table **ELEVE**. Donc toute table d'une base de données doit avoir une clé primaire.

En effet, une clé primaire identifie de façon unique chaque enregistrement de la table dont elle en fait partie et améliore les performances du serveur de bases de données.



ELEVE : Table		
	Nom du champ	Type de données
	ID_ELEVE	Texte
	nom	Texte
	prénom	Texte
	date de naissance	Date/Heure
	classe	Texte

Figure : 7

Dans la pratique, les **SGBD** présentent la clé primaire d'une table de façon différente à celle des autres champs. Par exemple **MS ACCESS** la précède par le symbole  comme indique la figure ci-dessus.

### d) Création d'une table

#### Activité :3

On se propose de créer la table **ELEVE** de la base de données «GESTION DE NOTES» de l'activité 1 avec les propriétés des champs suivantes :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

La clé primaire de la table est **ID\_ELEVE**


Champ	Type de données	Propriétés
ID_ELEVE	Texte	Taille =4
Nom	Texte	Taille =20
prénom	Texte	Taille =20
Date de naissance	Date/Heure	Format abrégé
classe	Texte	Taille =8



Reconnaître le menu et les commandes nécessaires pour créer cette table.

### Solution

Pour créer cette table on peut suivre la démarche suivante :

- Sélectionner l'objet «**Tables**» de la fenêtre base de données
- Cliquer sur le bouton «**Nouveau**»
- Choisir l'option « **Mode création** » puis confirmer
- Créer les champs de la table et fixer pour chacun ces propriétés :
  - Pour chaque champ :
    - Saisir son nom
    - Choisir son type de données
    - Fixer les propriétés indiquées plus haut
- Choisir la clé primaire ID\_ELEVE
  - Sélectionner le champ ID\_ELEVE
  - Cliquant sur l'outil  de la barre d'outils
- Enregistrer la table sous le nom ELEVE
- Fermer la table.

### Constatations

Dans la table ELEVE, la clé primaire est constituée d'un seul champ de même pour la table MATIERE. Par contre dans la table NOTE, la clé primaire est constituée de deux champs ID\_ELEVE et MAT\_MATIERE (car une note est attribuée à un élève dans une matière) comme indiqué dans le schéma de la table suivant :

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

Pour ce la, au moment du choix de la clé primaire, on doit :

- Sélectionner les deux champs ID\_ELEVE et MAT\_MATIERE
- Puis cliquer sur l'outil clé primaire de la boîte à outils.

### Application 1

Evaluer les propositions suivantes en mettant la lettre «V» si la proposition correspondante est vraie sinon la lettre «F» :

- Une base de données peut contenir une seule table
- Le SGBD accepte la création d'une table sans clé primaire
- Une clé primaire est l'identifiant unique des enregistrements dans une table
- Les données d'un seul champ peuvent avoir des propriétés différentes

Créer, de la même manière, les tables MATIERE et NOTE de la base de données «GESTION DES NOTES» en respectant les types ainsi que les propriétés des champs présentées dans les tableaux suivants:

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

Champ	Type de données	Propriétés
MAT_MATIERE	Texte	Taille =3
libellé matière	Texte	Taille =20
coefficient	Numérique	Taille = réel simple

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

Champ	Type de données	Propriétés
MAT_MATIERE	Texte	Taille =4
libellé matière	Texte	Taille =3
Note	Numérique	Taille = réel simple

**Retenons**

- Une table représente une entité du monde réel
- Un champ d'une table est caractérisé par un type et une liste de propriétés
- Dans une table, un ou un ensemble de champs forment sa clé primaire appelée aussi identifiant dont le principal rôle est de déterminer de façon unique un et un seul enregistrement de la table.

**2. Les relations**

**Activité : 4**

Reprenons le schéma de la base de données «GESTION DE NOTES» :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

Dans cette base de données, pourquoi les clés primaires des tables ELEVE et MATIERE figurent dans la table NOTE ?

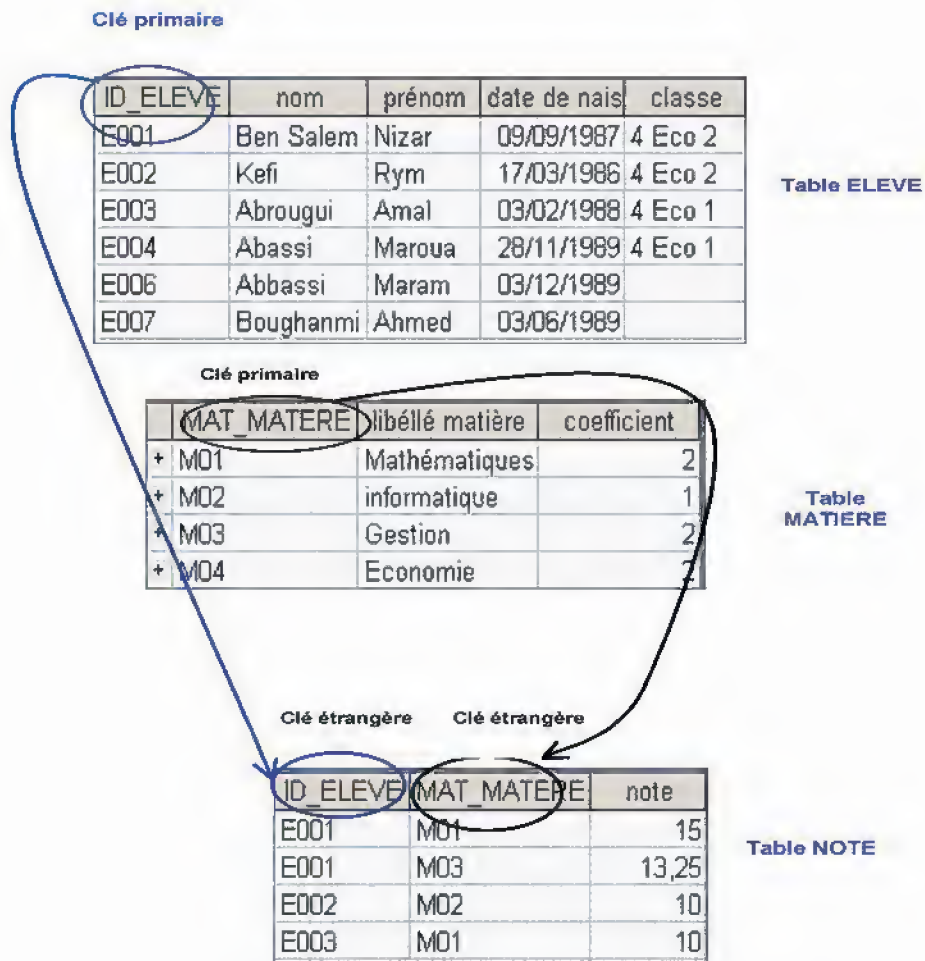
**Solution**

Pour assurer la cohérence de cette base de données, on peut appliquer les règles de gestion suivante :

- ✓ Une note, dans une matière, n'est affectée qu'à un seul élève
- ✓ Un élève peut avoir plusieurs notes
- ✓ Une note est associée à une matière
- ✓ plusieurs notes peuvent être affectées à une matière pour plusieurs élèves.

D'après ces règles de gestion, on remarque qu'une note est identifiée par un élève et une matière

*a) Définition d'une relation*



**Figure : 8**

Les relations d'une base de données représentent un ensemble de règles ou de méthodes qui régissent les données dans les tables. Si chaque note est attribuée à un élève et un élève peut avoir plusieurs notes, on peut créer une relation 1-N entre les élèves et les notes.

Une relation a pour principe la correspondance des données des champs clés de deux tables, ces champs ont généralement le même nom dans les deux tables.

Dans la plupart des cas, un tel champ est la clé primaire de l'une des tables, qui fournit un identificateur unique pour chaque enregistrement, et est une clé étrangère dans l'autre table. Par exemple, il est possible d'associer les élèves avec les notes en créant une relation entre la table ELEVE et la table NOTE.

### *b) Les types de relations*

✓ **Relation un à un (1-1)** : signifie qu'un enregistrement de la première table ne peut correspondre qu'à un seul enregistrement dans la deuxième table.

Par exemple, dans une base de données de gestion des personnels, si un employé ne peut être directeur que d'un seul département et un département n'a qu'un seul employé comme directeur. On définit ainsi une relation **un à un**.

✓ **Relation un à plusieurs (1-N)** : signifie que chaque enregistrement de la première table, contenant la clé primaire, peut être associé à plusieurs enregistrements de la deuxième table.

Par exemple, dans une base de données de gestion des personnels, si un département peut avoir plusieurs employés qui réalisent différentes fonctions mais chaque employé ne fait partie que d'un seul département. On définit ici une relation de type **un à plusieurs**.

✓ **Relation plusieurs à plusieurs (N-M)** : signifie que chaque enregistrement de la première table peut être associé à plusieurs enregistrements de la deuxième table et inversement

Par exemple, dans une base de données de gestion de production un type de produit peut être fabriqué en plusieurs usines et une usine donnée peut fabriquer plusieurs types de produits. On définit ici une relation de type **plusieurs à plusieurs**.

### *c) Création d'une relation*

#### **Activité : 5**

Soit la base de données «GESTION DE NOTES» définie plus haut. On se propose de créer les relations possibles entre ses tables.

Le schéma de cette base est :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

Pour simplifier le traitement on a admis, comme règle de gestion, qu'un élève ne peut avoir qu'une seule note pour une matière.

D'après le schéma de la base, on distingue l'existence de deux relations : la première entre la table ELEVE et la table NOTE et la deuxième entre la table MATIERE et la table NOTE.

Reconnaître le menu et les commandes nécessaires pour créer la relation qui lie la table ELEVE à la table NOTE.

### Solution

Pour créer la relation entre la table ELEVE et la table NOTE on peut suivre la démarche suivante :

- ✓ Ouvrir le menu «Outils» puis cliquer sur la commande «Relations...»
- ✓ Ajouter les tables ELEVE et NOTE en question
- ✓ Sélectionner la clé primaire ID\_ELEVE de la table ELEVE et le glisser, en maintenant le bouton gauche de la souris, vers la clé étrangère de la table NOTE
- ✓ Vérifier, dans la boîte de dialogue suivante, les champs puis cocher la case «Appliquer l'intégrité référentielle»

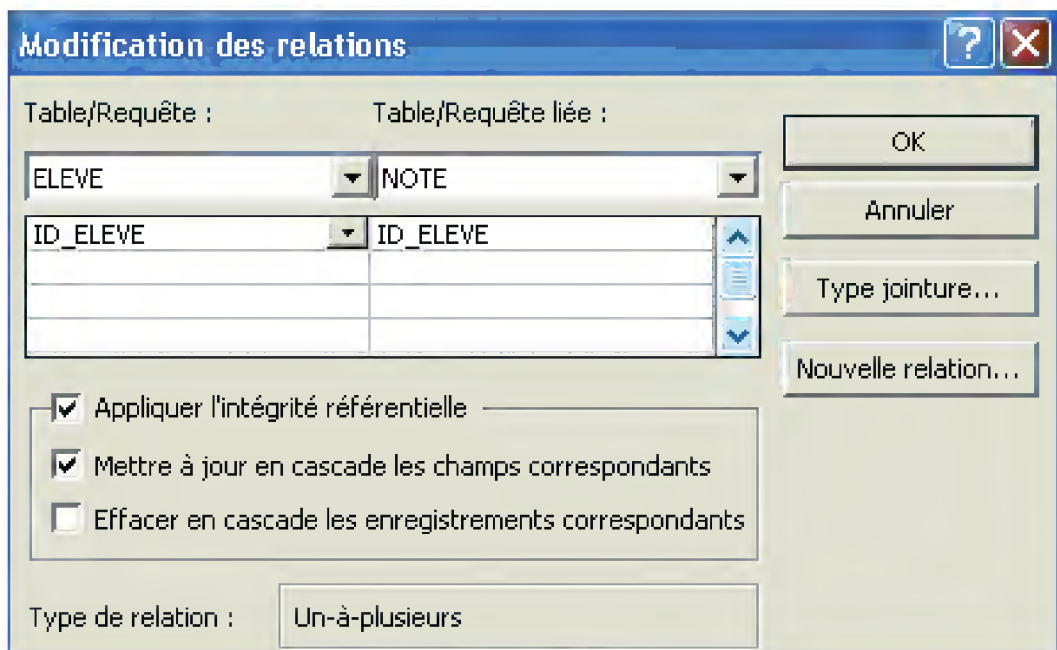


Figure : 9

- Valider ce paramètre en cliquant sur « Créer »
- Enregistrer les modifications apportées
- La relation est donnée par la figure suivante :

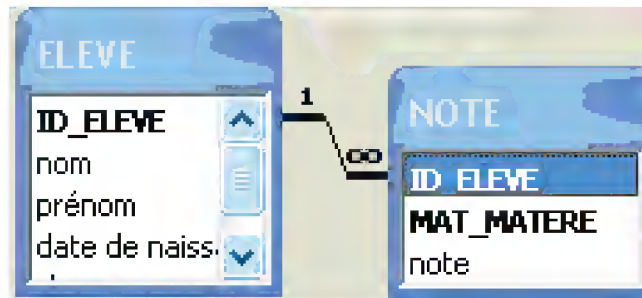


Figure : 10

### Constatations

- ✓ la clé primaire et la clé étrangère correspondante qui forment une relation sont de même type
- ✓ on ne doit pas créer une relation entre des champs qui n'ont pas le même genre d'information même s'ils ont le même type. Par exemple, les champs **nom** et **adresse**.
- ✓ les relations, dans une base de données, traduisent les règles de gestion de la base

### d) Application

Créer, de la même manière, la relation entre la table METIERE et la table NOTE

### Retenons

- Une relation est un lien entre deux tables de la base de données établi entre deux champs en commun.
- L'intérêt pratique de relier des tables sur des champs en commun est d'éviter la redondance (duplication d'une même information dans plusieurs tables).
- Dans la phase de conception d'une base de données, une relation de type N-M génère une nouvelle table intermédiaire et deux relations de type 1-N.

### Activité : 6

On souhaite créer une base de données « GESTION DES PROJETS » pour une entreprise. Une première étude a mis en évidence trois tables.

EMPLOYE (NumEmp, Nom, Prénom, Adresse, Tél, Grade, NumService)

SERVICE (NumService, NomService, Responsable, Tél)

PROJET (NumProjet, NomProjet, DateDeb, DateFin, NumService)

Avec les relations suivantes :



Figure : 11

Questions :

- Un employé peut-il travailler dans deux services ?
- Un employé peut-il travailler sur plusieurs projets ?
- Une personne peut-elle être responsable de plusieurs services ?

### Solution

- Un employé ne peut travailler que dans un seul service car le champ « NumService » ne peut avoir qu'une seule valeur.
- Un employé ne peut faire partie que d'un seul service et un projet ne peut être attaché qu'à un seul service. Donc un employé ne peut pas travailler sur deux projets en même temps.
- Une personne peut être responsable de plusieurs services.

### e) Exercice

Soit la base de données « GESTION STOCK » de schéma relationnel suivant :

PRODUITS (Réf produit, nom du produit, N° fournisseur, Quantité, Prix unitaire)

CLIENTS (Code client, nom, prénom, fonction, Adresse, Ville, Code postal, Tél)

FOURNISSEURS (N° fournisseur, Nom, prénom, Adresse, Ville, Code postal, Tél)

COMMANDES (N°commande, Code client, Date com, Date liv, Adresse liv, ville, liv)

DETAILS COMMANDES (N° commande, Réf produit, Prix unitaire, Quantité, Remise)

et soit les relations reliant les tables de la base suivantes :

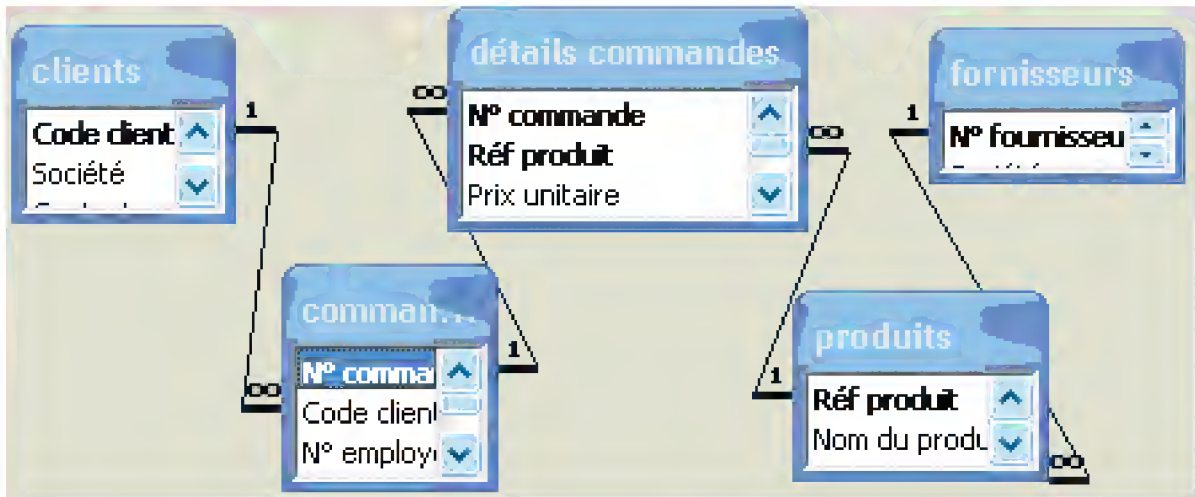


Figure : 12

**Question :** Créer la base de données « GESTION STOCK » ainsi que les relations entre ses tables.

#### IV. Création des requêtes

##### Activité : 7

Reprenons la base de données «GESTION DES NOTES», décrite par le schéma suivant :

**ELEVE (ID\_ELEVE, nom, prénom, date de naissance, classe)**

**MATIERE (MAT\_MATIERE, libellé matière, coefficient)**

**NOTE (ID\_ELEVE, MAT\_MATIERE, note)**

##### Solution

On se propose de répondre aux questions suivantes :

- Quels sont les élèves de la classe 4ème Eco 2 ?
- Quelles sont les matières dont le coefficient est supérieur ou égal à 2 ?
- Quels sont les élèves ayant une note supérieure à 12 en gestion ?

On parcourant les données des tables de la base de données «GESTION DES NOTES» on peut répondre, manuellement, aux questions posées plus haut.

- la liste des élèves de la classe 4ème Eco 2 est :

Nom	Prénom	Classe
Ben Salem	Nizar	4 Eco 2
Kefi	Rym	4 Eco 2



b. les matières dont le coefficient est supérieur ou égale à 2 sont :

libellé matière	coefficient
Mathématiques	2
Gestion	2
Economie	2

c. les élèves ayant une note supérieure à 12 en gestion sont :

Nom	Prénom	note
Ben Salem	Nizar	13,25

A l'aide de cette liste de questions on peut interroger la base de données afin de rechercher les informations dont on a besoin. C'est l'aspect le plus intéressant de l'étude des bases de données qui consiste à enchaîner astucieusement un petit nombre de requêtes bien choisies.

Ces questions représentent des requêtes à lancer pour avoir de façon précise une (ou un ensemble) de données.

Il existe plusieurs types de requêtes dont les plus utilisées sont :

- les requêtes de sélection
- les requêtes paramétrées
- les requêtes d'ajout
- les requêtes de mise à jour

### 1. Les requêtes de sélection

Stocker sans cesse des informations dans une base de données et en assurer la maintenance, n'est pas une fin en soi. Il faut pouvoir retrouver, chaque fois que cela est nécessaire, les informations pertinentes dont on a besoin. La requête de sélection a été créée dans ce but. Elle joue, dans les bases de données, un rôle très important.

La sélection représente l'outil courant de recherche d'informations dans les bases de données. D'une manière générale, une requête de sélection :

- s'applique soit à une seule table, soit à plusieurs tables liées par des relations ;
- permet de sélectionner les lignes par application d'un ou de plusieurs critères portant sur un ou sur plusieurs champs ;
- permet de choisir les colonnes que l'on veut conserver (comme la sélection simple) ;
- peut enregistrer le résultat recherché sous forme d'une table.

a) Création d'une requête de sélection

**Activité : 8**

Reprenons le schéma relationnel de la base de données «GESTION DES PROJETS» suivant :

**EMPLOYE (NumEmp, Nom, Prénom, Adresse, Tél, Grade, NumService)**

**SERVICE (NumService, NomService, Responsable, Tél)**

**PROJET (NumProjet, NomProjet, DateDeb, DateFin, NumService)**

On se propose de répondre aux requêtes suivantes :

Requête 1 : Donner la liste des ouvriers.

Requête 2 : Quels sont les projets dirigés par le service travaux ?

**Solution**

Requête 1 : Donner la liste des ouvriers.

Les données résultat de cette requête peuvent être formées des noms et des prénoms des employés dont leur grade est "ouvrier". La source de ces données est la table EMPLOYE.

Pour créer cette requête, on peut suivre la démarche suivante :

- Choisir l'objet « Requêtes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau »
- Choisir le mode de création ; par exemple « Mode création »
- Dans la fenêtre « Afficher la table », ajouter la table EMPLOYE puis la fermer

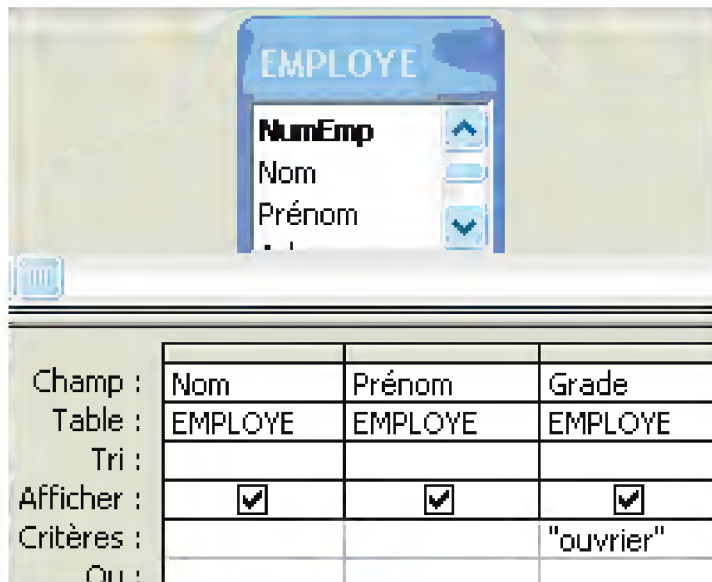


Figure : 13

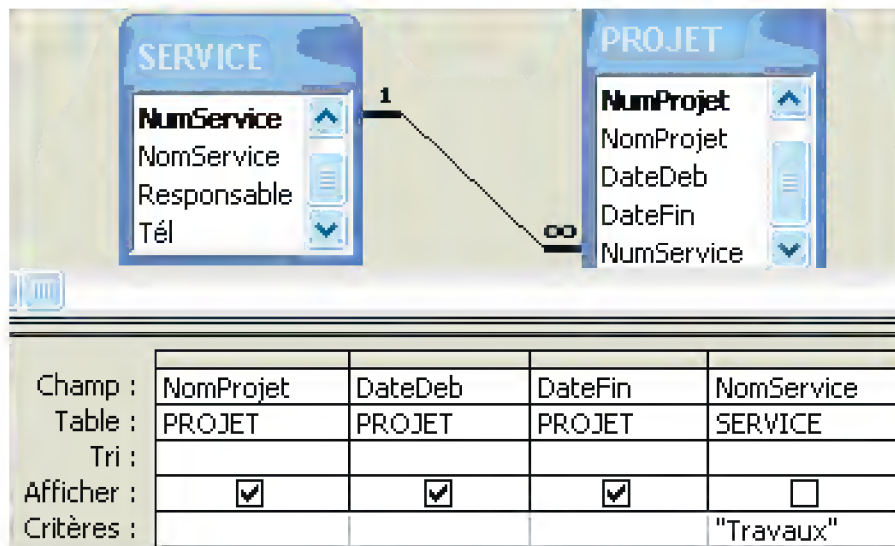
Dans cette fenêtre, et comme indique la figure précédente on doit fixer les paramètres de la requête :

- fixer la liste des champs à utiliser dans la requête (Nom, Prénom et Grade)
- fixer, pour chaque champ, la table dont il fait partie (table EMPLOYE)
- choisir l'ordre des données résultats à l'aide de l'attribut « Tri »
- cocher les cases de l'attribut « Afficher » des champs qui vont former le résultat (Nom et Prénom)
- spécifier, dans la clause « Critères », le ou les critères de sélection des données (Grade ="ouvrier")
- Enregistrer la requête sous le nom "liste des ouvriers" puis fermer la fenêtre.

**Requête 2 :** Quels sont les projets dirigés par le service des travaux ?

Le résultat peut être sous la forme de Nom du projet, date de début et date de fin. Dans ce cas, les données du résultat viennent de la table PROJET.

- Choisir l'objet « Requêtes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis sur « Mode création »
- Dans la fenêtre « Afficher la table », ajouter les tables SERVICE et PROJET puis la fermer



**Figure : 14**

Dans la grille de création de la requête, on doit :

- fixer la liste des champs à utiliser (NomProjet, DateDeb, DateFin et NomService)
- fixer, pour chaque champ, la table dont il fait partie (la table PROJET pour le NomProjet, DateDeb et DateFin et la table SERVICE pour le NomService)
- cocher les cases de l'attribut « Afficher » des champs qui vont former le résultat (NomProjet, DateDeb et DateFin)
- fixer dans la clause « Critères » les critères de sélection de la requête NomService = "Travaux".

### Constatations

- Dans la requête 1, on a utilisé une seule table (la table EMPLOYE) car les champs utilisés viennent de cette dernière.
- Dans la requête 2, on a utilisé les deux tables (EMPLOYE et PROJET)
- Les données, résultat d'une requête de sélection, sont souvent regroupées sous forme d'une table qu'on peut utiliser dans d'autres requêtes.
- Dans une requête, les critères de sélection sont des expressions qui peuvent utiliser des opérateurs logiques (OU, ET, NON) et les opérateurs relationnels (=, <, <=, >, >=)

#### b) Exécution d'une requête

##### Activité : 9

Exécuter la requête 1 de la base de données «GESTION DES PROJETS»

##### Solution

Pour ce faire, on peut suivre la démarche suivante :

- Choisir l'objet « Requêtes » de la fenêtre «base de données»
- double-cliquer sur la requête "liste des ouvriers" ou bien cliquer sur l'outil « Exécuter » de la boîte à outils.
- Le résultat s'affiche sous la forme d'une table comme suit :

Nom	Prénom	Grade
Ben Mahmoud	Anis	Ouvrier
Aouadi	Ilyes	Ouvrier
Issaoui	Mahmoud	Ouvrier

Figure : 15

#### c) Application

Reprenons la base de données **GESTION DES PROJETS**.

Créer puis exécuter la requête qui permet de donner tous les services qui ont supervisé des projets réalisés durant l'année 2006.

- La requête est le seul moyen pour interroger une base de données
- Le résultat d'une requête, sur une base de données, peut être considéré comme source de données dans d'autres requêtes.

## 2. Les requêtes paramétrées

##### Activité : 10

Soit le schéma relationnel de la base de données «GESTION DES PROJETS» :

**EMPLOYE** (NumEmp, Nom, Prénom, Adresse, Tél, Grade, NumService)

**SERVICE** (NumService, NomService, Responsable, Tél)

**PROJET** (NumProjet, NomProjet, DateDeb, DateFin, NumService)

Supposons que le directeur d'une entreprise veut consulter la liste des ouvriers d'un service donné.

Quelle solution peut - on proposer ?

**Solution**

Pour ne pas créer à chaque fois une nouvelle requête qui donne la liste des ouvriers d'un service donné, il est plus commode d'écrire une seule fois la requête et d'entrer à chaque exécution le nom du service dont on veut afficher ces ouvriers.

La requête sera nommée "requête paramétrée" et le paramètre est le nom du service.

La démarche à suivre pour réaliser cette requête est :

- Cliquer sur l'objet « Requêtes » dans la fenêtre « Base de données »
- Cliquer sur le bouton « Nouveau » puis sur « Mode création »
- Ajouter la table EMPLOYE et SERVICE

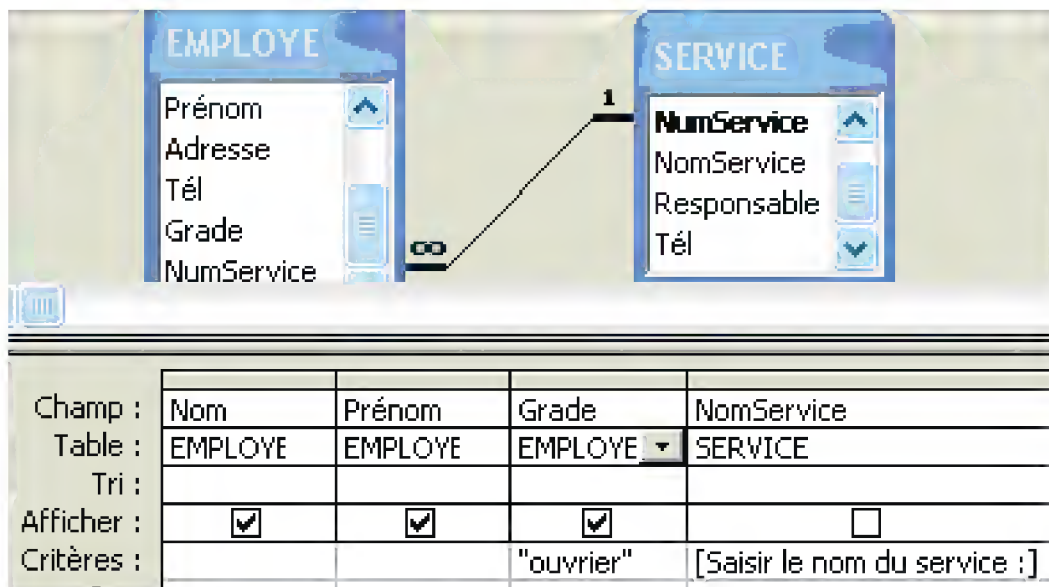


Figure : 16

- Dans la grille de création de la requête, le paramètre "NomService" est représenté par un message écrit entre crochets comme indiqué plus haut
- Enregistrer la requête sous le nom "requête paramétrée".

Au moment de l'exécution de la requête, la boîte de dialogue suivante s'affiche :

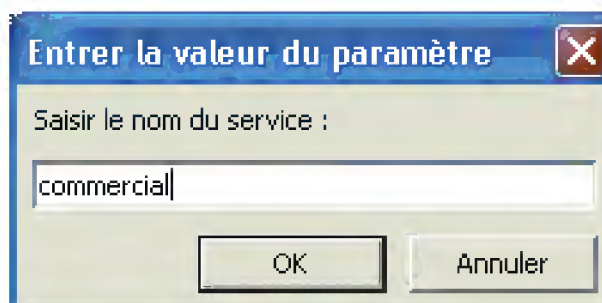


Figure : 17

Après avoir saisi le nom du service, on valide en cliquant sur le bouton "OK", le résultat de la requête sera affiché dans la table suivante :

Nom	Prénom	Grade
Ben Mahmoud	Anis	Ouvrier
Aouadi	Ilyes	Ouvrier

Figure : 18

### Remarque

En plus des requêtes de sélection, un **SGBD** offre aux concepteurs des bases de données la possibilité de créer des requêtes d'action. Une requête action est capable de modifier ou de déplacer un grand nombre d'enregistrements en une seule opération.

Dans la suite on va détailler deux exemples de requêtes action : les requêtes de mise à jour et les requêtes d'ajout.

### 3. Les requêtes d'ajout

Une requête d'ajout ajoute un groupe d'enregistrements d'une ou de plusieurs tables à la fin d'une ou de plusieurs tables.

#### Activité : 11

Soit le schéma relationnel de la base de données «GESTION DES PROJETS» :

**EMPLOYE (NumEmp, Nom, Prénom, Adresse, Tél, Grade, NumService)**

**SERVICE (NumService, NomService, Responsable, Tél)**

**PROJET (NumProjet, NomProjet, DateDeb, DateFin, NumService)**

et soit une liste de projets qui sont en cours de validation. Ces projets sont regroupés dans la table LISTE\_ATTENTE.

Le schéma de la table LISTE\_ATTENTE est analogue à celui de la table PROJET.

La commission de validation décide de valider tous les projets dont la date de fin « DateFin » est avant "31/12/2007".0

LISTE_ATTENTE : Table				
NumProjet	NomProjet	DateDeb	DateFin	NumService
0006	projet 8	02/01/2007	05/05/2007	01
0007	projet 15	07/01/2007	16/02/2008	02
0008	projet 13	28/02/2007	17/05/2007	03
0009	projet 20	12/01/2007	18/02/2008	04
0010	projet 14	20/02/2007	03/06/2007	01

Figure : 19

On veut ajouter les projets validés par la commission à la table PROJET.  
Pour éviter de taper toutes ces informations, quelle solution peut - on proposer ?

### Solution

Pour répondre à cette question on doit parcourir la table LISTE ATTENTE et extraire les enregistrements répondant à la contrainte ( $DateFin \leq '31/12/2007'$ ) puis les ajouter à la table ELEVE.

- Choisir l'objet « Requête » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Mode création »
- Ajouter la table LISTE ATTENTE qui représente la source des informations de la requête
- On remarque l'ajout du nouveau menu « Requête » à la barre des menus.

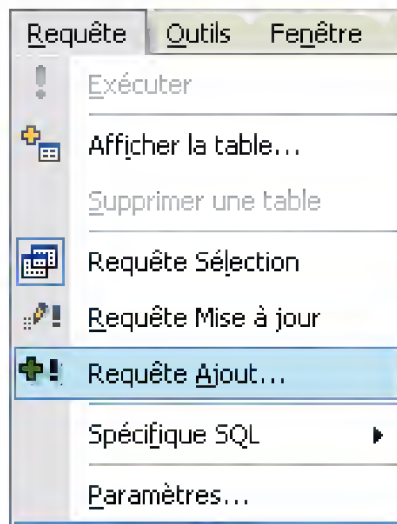


Figure : 20

- Choisir maintenant la commande « Requête Ajout »
- Dans la boîte de dialogue suivante qui s'affiche, choisir la table dans la quelle on va ajouter les enregistrements en question. Dans notre cas, choisir la table **PROJET**.

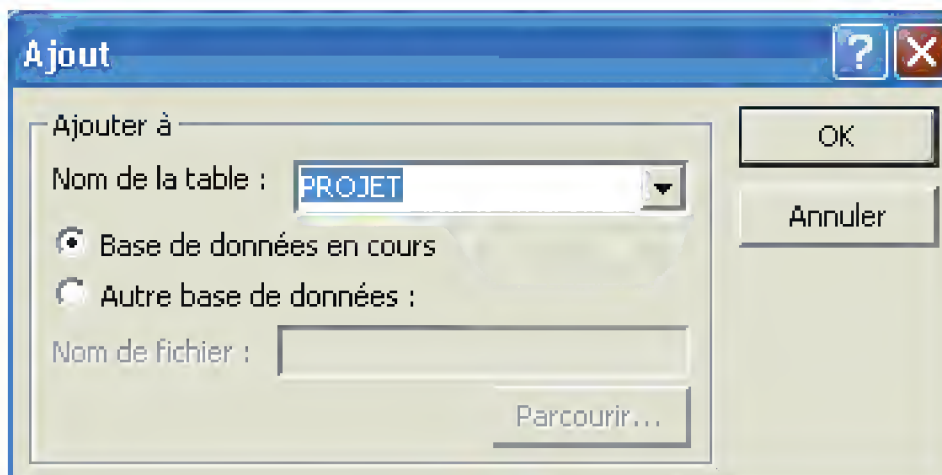


Figure : 21

Dans la grille de création de la requête, fixer les champs à ajouter à la table PROJET puis spécifier la contrainte (DateFin ≤ "31/12/2007") :

Champ :	NumProje	NomProjet	DateDeb	DateFin	NumServ
Table :	LISTE_AT	LISTE_AT	LISTE_A	LISTE_ATTENTE	LISTE_A1
Tri :					
Ajouter à :	NumProje	NomProjet	DateDeb	DateFin	NumServ
Critères :				<=#31/12/2007#	

Figure : 22

- Enregistrer la requête sous le nom « Ajout\_Projets ».

Après exécution de la requête d'ajout « Ajout\_Projets » la table PROJET devient :

PROJET : Table				
NumProjet	NomProjet	DateDeb	DateFin	NumService
0001	projet 1	02/01/2004	15/07/2005	01
0002	projet 5	10/03/2006	15/09/2006	01
0003	projet 3	28/10/2005	15/05/2006	03
0004	projet 2	12/01/2004	18/06/2005	04
0005	projet 4	20/01/2006	01/07/2006	01
0006	projet 8	02/01/2007	05/05/2007	01
0008	projet 13	28/02/2007	17/05/2007	03
0010	projet 14	20/02/2007	03/06/2007	01

Figure : 23

### Constatations

- Après exécution de la requête d'ajout « Ajout\_Projets », les enregistrements s'ajoutent à la fin de la table destination PROJET.

- On remarque que seul les projets, de la table LISTE ATTENTE dont leurs dates de début sont avant "31/12/2007", s'ajoutent à la table PROJET.



#### 4. Les requêtes de mise à jour

Une requête de Mise à jour apporte des changements globaux à un groupe d'enregistrements dans une ou plusieurs tables. On peut par exemple augmenter les prix de 5% sur tous les produits du stock, ou augmenter de 10% les salaires des ouvriers appartenant à une entreprise et ayant une certaine catégorie professionnelle. Une requête de Mise à jour permet de modifier les données d'une table existante.

#### Activité : 12

Reprendre le schéma relationnel de la base de données « GESTION STOCK » :

**PRODUITS** (Réf produit, nom du produit, N° fournisseur, Quantité, Prix unitaire)

**CLIENTS** (Code client, nom, prénom, fonction, Adresse, Ville, Code postal, Tél)

**FOURNISSEURS** (N° fournisseur, Nom, prénom, Adresse, Ville, Code postal, Tél)

**COMMANDES** (N° commande, Code client, Date com, Date liv, Adresse liv, ville, liv)

**DETAILS COMMANDES** (N° commande, Réf produit, Prix unitaire, Quantité, Remise)

Avec les relations suivantes :

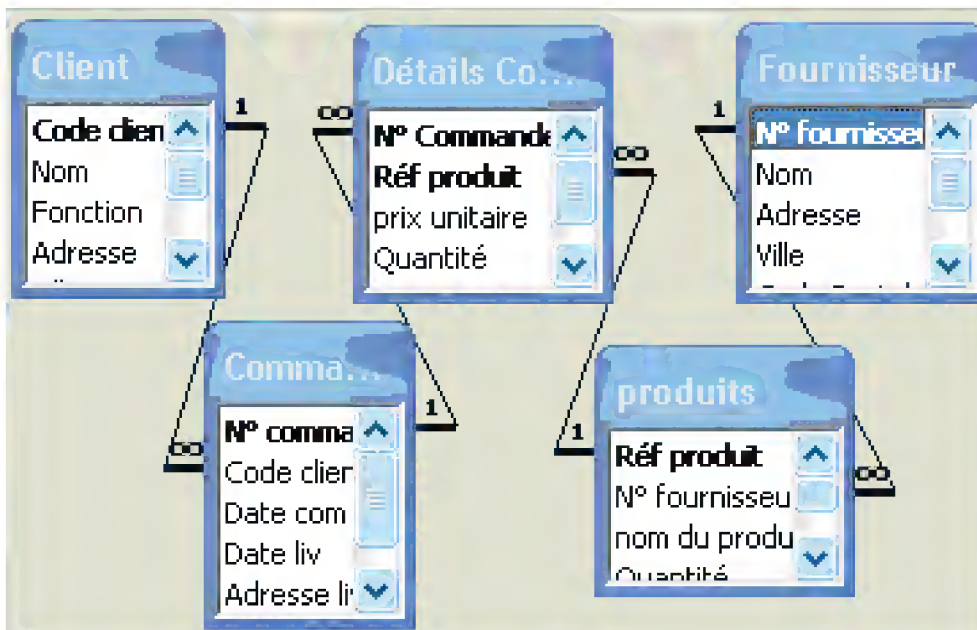


Figure : 24

#### Questions

- Ecrire la requête de mise à jour qui permet d'augmenter les prix unitaires des tous les produits de 5%.
- Ecrire la requête qui permet de changer la date de livraison « Date liv » de tous les produits fournis par le fournisseur Ali à la date "19/04/2007"

### Solution

1. Dans cette requête on va mettre à jour toutes les données de la colonne « prix unitaire » de la table PRODUITS pour qu'ils soient augmentés de 5%.

- Choisir l'objet « Requetes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Mode création »
- Choisir la commande « Requête Mise à jour » du menu « Requête »
- Dans la grille de création de la requête Mise à jour fixer le ou les champs à mettre à jour.
  - Fixer la table PRODUITS dans la clause « Table »
  - Choisir le champ « Prix unitaire » dans la clause « Champ »
  - Dans la clause « Mise à jour », taper la modification à effectuer :

**[Prix unitaire]+ [Prix unitaire]\*0,05.**

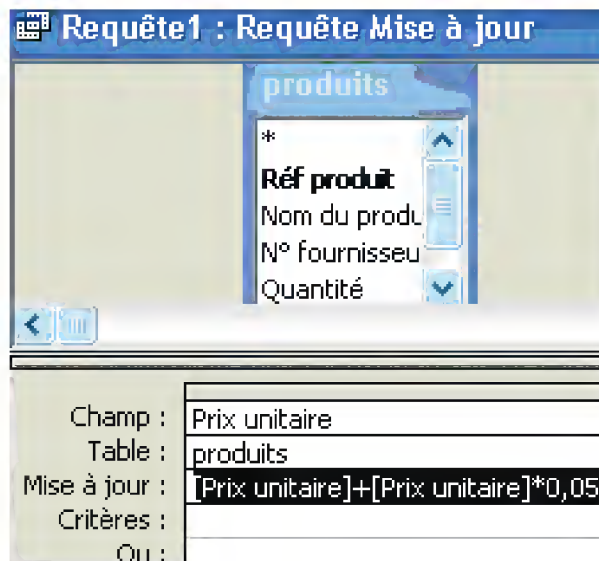


Figure : 25

- enregistrer puis exécuter la requête.

2. Dans cette requête, on remarque que :

- La mise à jour : remplacer les dates de livraison par « 19/04/2007 ».
- La contrainte (critère) : les produits fournis par le Fournisseur « Ali ».
- Les tables à consulter sont : **COMMANDES, DETAILS COMMANDES, PRODUITS et FOURNISSEURS**

La démarche à suivre pour réaliser cette requête est :

- Choisir l'objet « Requetes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Mode création »
- Ajouter les tables COMMANDES, DETAILS COMMANDES, PRODUITS et FOURNISSEURS

- Choisir la commande « Requête Mise à jour » du menu « Requête »
- Dans la grille de création de la requête :
  - Choisir le champ "Date liv" de la table COMMANDES
  - Choisir le champ "Nom" de la table FOURNISSEURS
  - Dans la clause « Mise à jour », taper la date "19/04/2007"
  - Dans la clause « Critères », taper le nom du fournisseur "Ali"

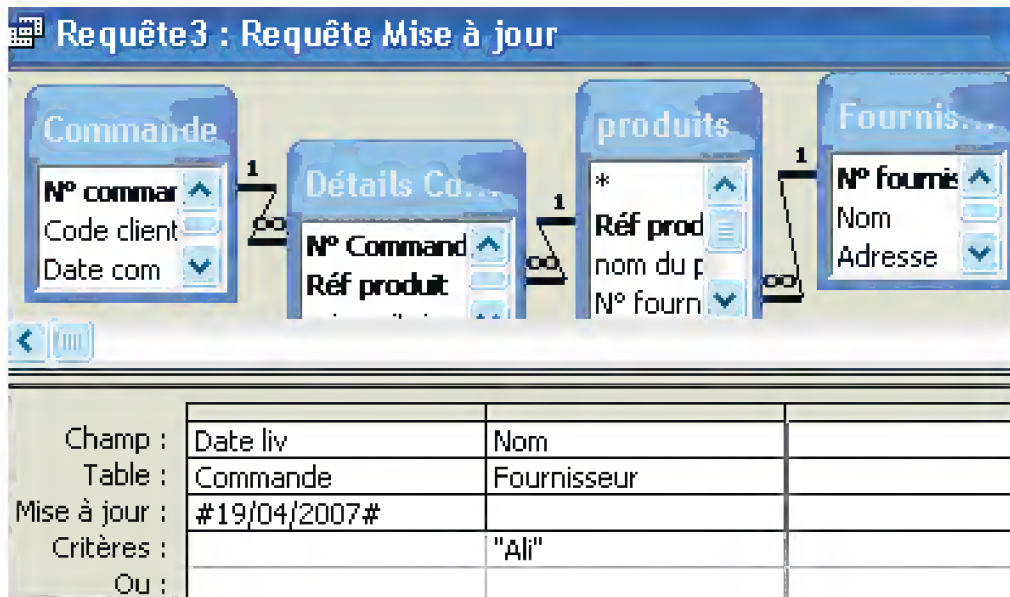


Figure : 26

- Enregistrer puis exécuter la requête.

### 5. Exercice d'application

Soit la base de données "FABRICATION" dont le schéma est donné ci-dessous.

**PIECE (NOP, DESIGNATION, COULEUR, POIDS)**

**SERVICE (NOS, INTITULE, LOCALISATION)**

**COMMANDE (NOP, NOS, QUANTITE)**

Avec :

- NOP : clé primaire de la table PIECE, il désigne le numéro de la pièce.
- NOS : clé primaire de la table SERVICE, il désigne le numéro du service.
- NOP et NOS : clé primaire de la table COMMANDE.

**Description de la table PIECE :**

Champ	Type de données	Propriétés
NOP	Texte	Taille = 5
DESIGNATION	Texte	Taille = 30
COULEUR	Texte	Taille = 20
POIDS	Numérique	Taille = réel simple

**Description de la table SERVICE**

Champ	Type de données	Propriétés
NOS	Texte	Taille =3
INTITULE	Texte	Taille =30
LOCALISATION	Texte	Taille = 40

**Description de la table COMMANDE :**

Champ	Type de données	Propriétés
NOP	Texte	Taille =5
NOS	Texte	Taille =3
QUANTITE	Numérique	Taille = entier

**Questions**

Répondre, brièvement, aux questions suivantes :

1. Créer les tables de la base de données "FABRICATION" ainsi que les relations qui les relie.
2. Créer les requêtes suivantes :
  - a. Les Noms des services ayant en commande la pièce "P1" avec une quantité supérieure à 10, dans l'ordre croissant.
  - b. La liste des pièces commandées par un service donné avec leur désignation et leur poids.
  - c. L'augmentation des quantités des pièces commandées par le service S1 de 50%

**Solution**

1. les relations entre les tables de la base de données "FABRICATION" sont données par la figure suivante :

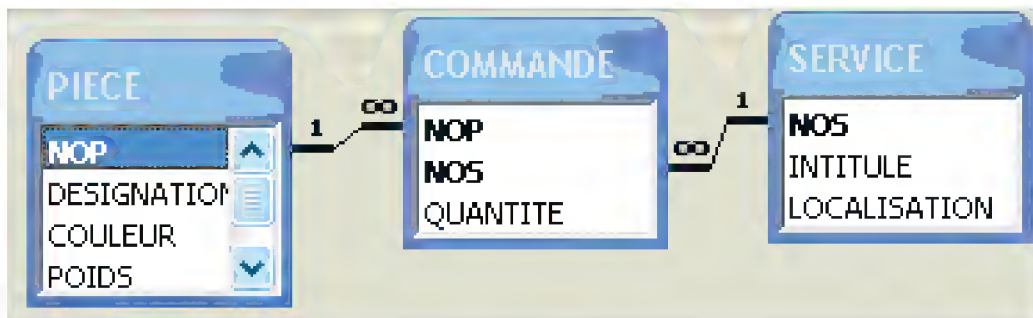


Figure : 27

a. On remarque que la requête est une requête de sélection. Pour répondre à cette dernière, on doit accéder aux trois tables :

- SERVICE : pour avoir les noms des services
- PIECE : pour comparer la désignation de la pièce avec le nom "P1"
- COMMANDE : pour vérifier que la quantité commandée est supérieure à 10

Pour créer la requête, on peut suivre la démarche suivante :

- Choisir l'objet « Requêtes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis sur « Mode création »
- Ajouter les tables PIECE, SERVICE et COMMANDE de la boîte de dialogue « Afficher la table »
- Dans la grille de création de la requête, fixer les champs et les critères comme indiqués plus haut :

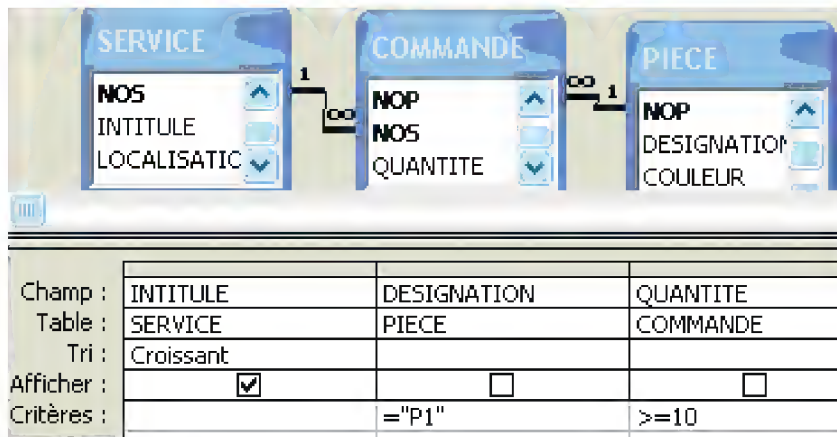


Figure : 28

b. Dans cette requête, on demande la liste des pièces commandées par un service donné. C'est-à-dire que le nom du service doit être saisi au clavier ; cela veut dire que la requête est paramétrée et le paramètre c'est le nom du service.

Dans ce cas, on a besoin aussi des trois tables. La grille de création suivante résume les champs à ajouter et le paramètre de la requête :

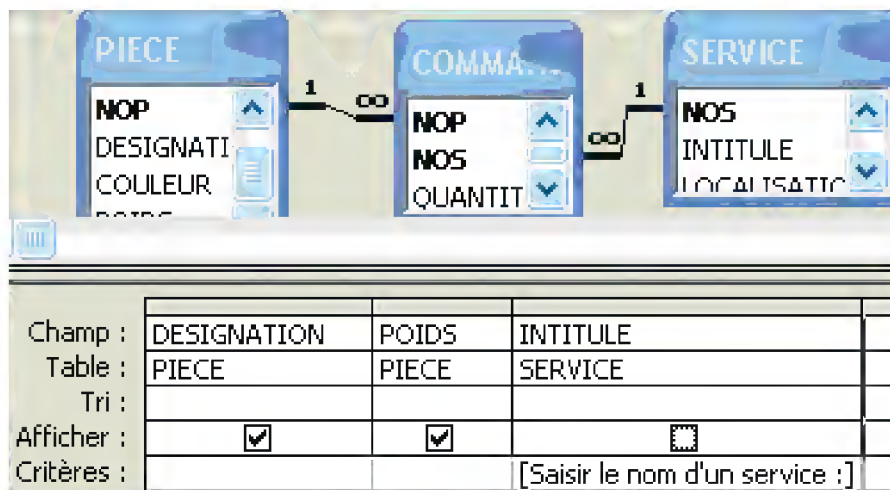


Figure : 29

c. La requête à créer dans ce cas est une requête de mise à jour car elle va modifier le contenu champ "QUANTITE" de la table COMMANDE pour qu'il soit augmenter de 50%.

Dans la démarche de création, et après avoir choisi le mode de création, on doit spécifier la nature de la requête en cliquant sur la commande « Requête Mise à jour » du menu « Requête ».

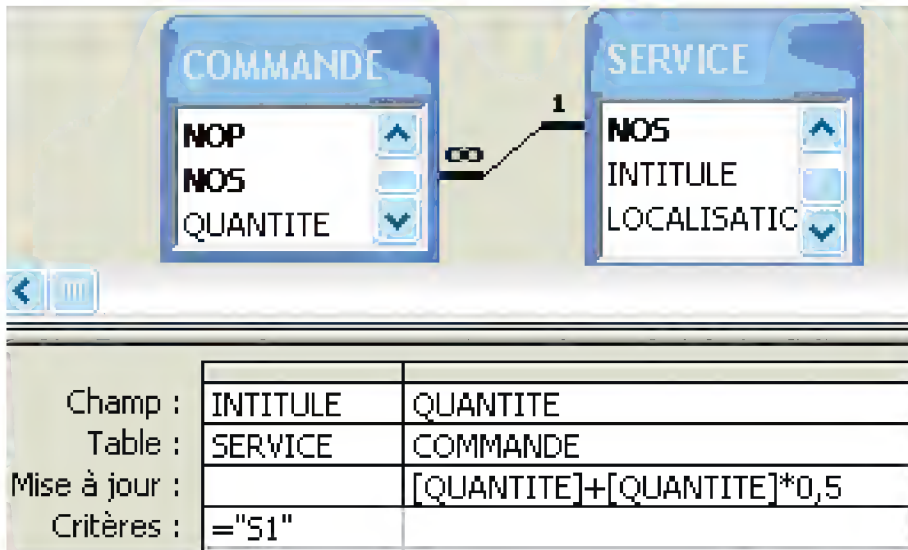


Figure : 30

On remarque, dans la grille de création de la requête, l'existence de la clause « Mise à jour » qui contient la mise à jour proposée.

## V. Création de formulaires

Un formulaire est un type d'objet d'une base de données utilisé essentiellement pour saisir et afficher les données dans une base de données. On peut également utiliser un formulaire comme :

- Menu général qui ouvre d'autres formulaires, des requêtes et des états dans la base de données,
- Boîte de dialogue personnalisée qui permet aux utilisateurs d'entrer des données et de les utiliser.

Dans la base de données « GESTION STOCK », la figure suivante représente un formulaire de saisie de produits

Figure : 31

- 1 : Etiquette, qui représente le titre du formulaire
- 2 : Les zones de texte, Chaque zone est associée à un champ de la table PRODUITS.
- 3 : Barre de défilement des enregistrements.
- 4 : Bouton de commande, utilisé pour quitter le formulaire.

**Activité :13**

Reconnaître le menu et les commandes nécessaires pour créer le formulaire de la figure 31.

**Solution**

Le formulaire à créer est utilisé pour saisir les données dans la table PRODUITS. Pour cela, il est possible d'utiliser l'assistant des formulaires comme indique la démarche suivante :

- Choisir l'objet « Formulaires » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir l'un des modes de création (par exemple, assistant formulaire).

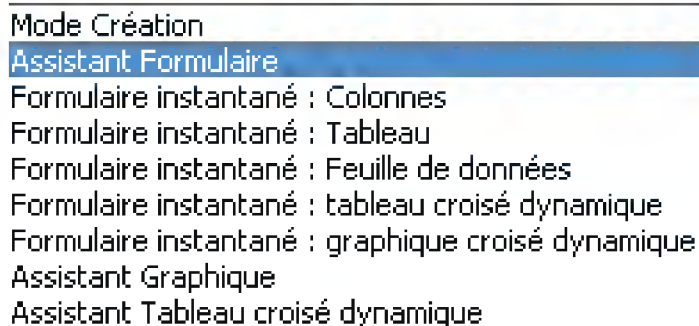


Figure : 32

- Fixer la table PRODUITS, source des données puis valider.

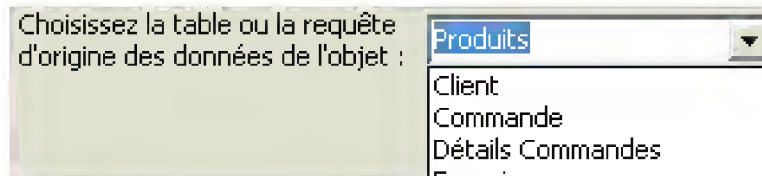


Figure : 33

- Ajouter les champs qu'on veut afficher dans le formulaire.
- Choisir la liste des champs, le style et le nom du formulaire puis l'enregistrer.

Le formulaire s'affiche en mode affichage avec un état initial. On peut modifier (ajout, suppression, mise en forme, ...) les objets du formulaire en basculant du mode affichage au mode création.

### Constatations

Un formulaire admet deux modes :

- Mode création : utilisé pour créer ou modifier les objets du formulaire ainsi que leurs propriétés.
- Mode affichage : utilisé pour consulter ou pour saisir les données de la base de données.

Dans un formulaire, on peut insérer une étiquette, une zone de texte, une case d'options, des cases à cocher, ...etc. Ces objets sont regroupés dans une boîte dite boîte à outils.

Pour insérer un objet on peut suivre la démarche suivante :

- Cliquer sur l'objet en question de la boîte à outils.
- Maintenir le bouton gauche de la souris enfoncé puis glisser
- Relâcher le bouton de la souris
- Accéder aux propriétés de l'objet et les modifier afin de répondre aux besoins.

	Etiquette
	Zone de texte
	Groupe d'options
	Bouton bascule
	Case d'option
	Case à cocher
	Zone de liste déroulante
	Zone de liste
	Bouton de commande
	Image
	Cadre d'objet indépendant
	Cadre d'objet indépendant
	...



Chaque objet du formulaire dispose d'une liste de propriétés qu'on peut consulter et/ou modifier.

Pour modifier les propriétés d'un objet du formulaire :

- Sélectionner le ou les objets en question
- Choisir la commande « propriétés » du menu « Affichage »



Figure : 34

Chaque objet d'un formulaire dispose d'une liste de propriétés qui dépendent de son type. Par exemple, pour les objets de type zone de texte, les propriétés sont présentées sous forme d'onglets : « Format », « Données », « Évènement » et « Autres ».

Dans l'onglet « Format », on peut spécifier les mises en forme de l'objet.

Dans l'onglet « Données », on peut spécifier la source de contrôle (le champ qui lui correspond dans la table PRODUITS), le masque de saisie, la valeur par défaut, ...

En plus des formulaires de saisie des données, on peut créer un formulaire dont le rôle est de rendre facile l'accès aux autres formulaires, d'exécuter les requêtes et d'imprimer les états de la base de données. Ce genre de formulaire joue le rôle d'un menu général pour la base de données.



Figure : 35

Dans ce formulaire, on remarque l'existence des boutons de commandes. Chacun, à l'aide d'un clic, permet d'ouvrir un formulaire. Par exemple, en cliquant sur le bouton « Saisie des produits » on ouvre le formulaire créé dans l'activité 11.

Chaque bouton de commande est un lien vers un autre objet qui peut être un formulaire, une requête, un état, ... etc.

### Activité : 14

Reconnaître les menus et les commandes nécessaires pour créer, dans le menu général, « GESTION STOCK » de la base de données « gestion stock » :

- un bouton de commande
- un lien entre un bouton et un objet

### Solution

#### Création d'un menu général :

Un menu général pour une base de données est un formulaire qui joue le rôle d'interface entre l'utilisateur et la base de données.

Pour créer ce formulaire on peut suivre la démarche suivante :

- Choisir l'objet « Formulaires » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode de création
- Après validation, le formulaire vierge suivant s'affiche en « mode formulaire » :

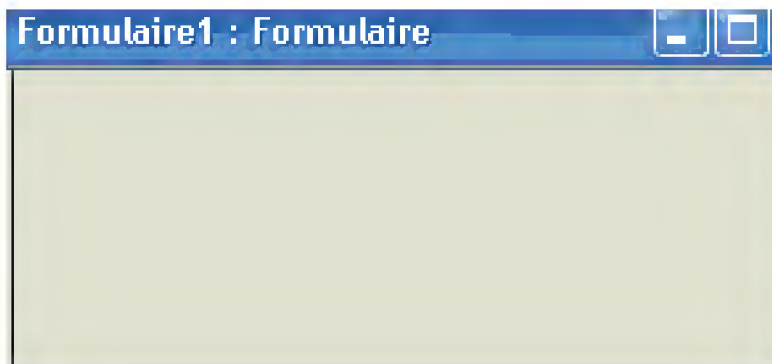


Figure : 36

Formulaire vierge en mode formulaire

Pour modifier le contenu du formulaire, on doit basculer du mode formulaire en mode création en cliquant sur la commande « Mode création » du menu « Affichage » ou bien en cliquant sur l'outil



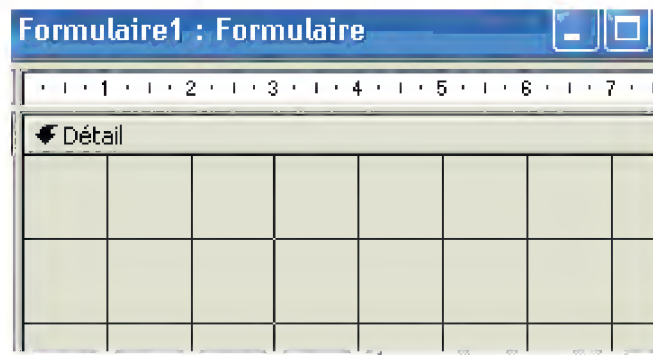


Figure : 37 Formulaire vierge en mode création

### Création des objets dans un formulaire :

Les objets qu'on peut ajouter sont regroupés dans la boîte à outils

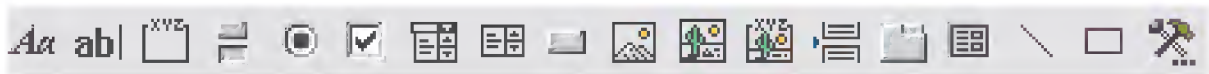



Figure : 38 Boîte à outils

La création d'un objet de la boîte à outils se fait de façon manuelle ou bien à l'aide d'un assistant qui peut effectuer à notre place toutes les étapes de création.

Dans cette activité, on ne va créer que les boutons de commande dans le formulaire « Menu général » présenté précédemment. On peut donc suivre la démarche suivante :

- Cliquer, dans la boîte à outils, sur l'objet « Bouton de commande »
- Activer l'outil « Assistants de contrôle » 
- Cliquer dans le formulaire à l'endroit où on veut le placer
- Suivre les instructions des boîtes de dialogue de l'Assistant

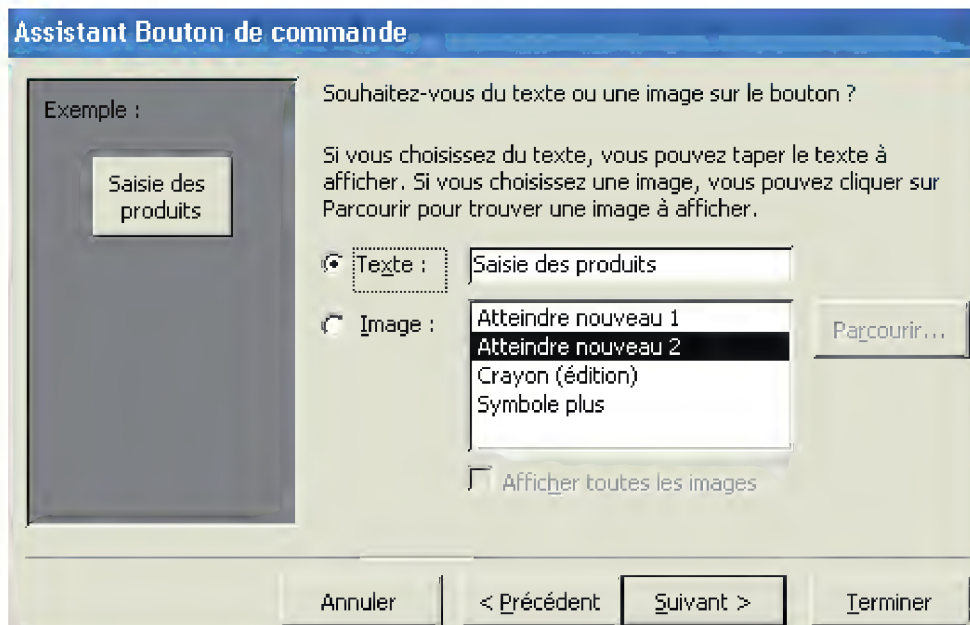


Figure : 39

- Dans la dernière boîte de dialogue, cliquer sur Terminer pour afficher le bouton de commande en mode Création.

### Création de l'action d'un bouton de commande :

Dans un formulaire, un bouton de commande permet d'exécuter une action ou un ensemble d'actions définies dans une macro.

Une macro est un ensemble composé d'une ou de plusieurs actions qui effectuent chacune une opération particulière, comme ouvrir un formulaire ou imprimer un état. Les macros permettent d'automatiser les tâches les plus courantes dans une base de données.

Pour créer une macro on peut suivre la démarche suivante :

- Choisir l'objet « Macro » de la fenêtre base de données
- Cliquer sur le bouton « Nouveau »
- Dans la colonne « Action » fixer l'action en question. Dans notre cas l'action est « OuvrirFormulaire »
- Fixer le nom du formulaire comme indique la figure suivante :

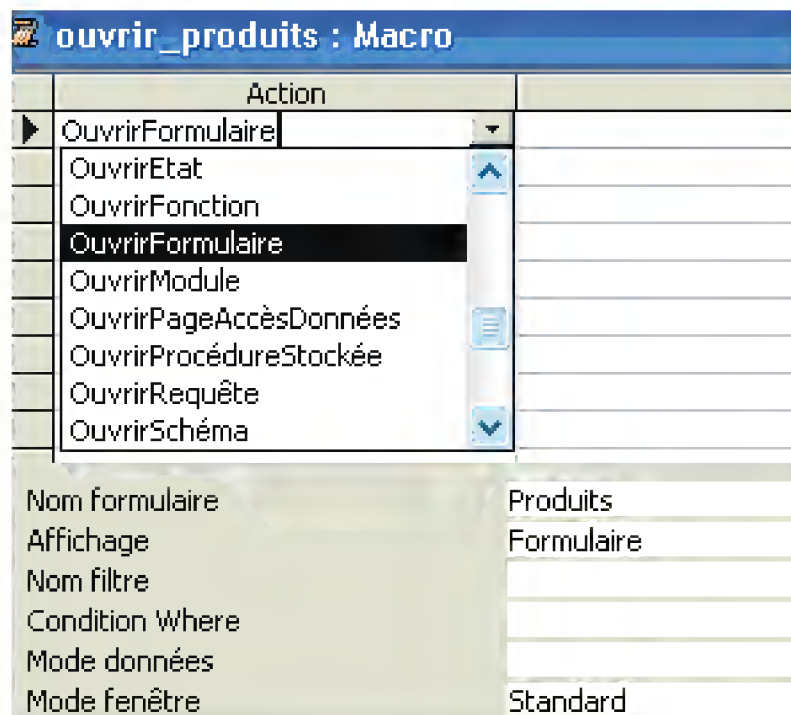


Figure : 40

- Enregistrer la macro sous le nom Ouvrir\_produits.

Maintenant on doit associer la macro au bouton de commande « saisie des produits » du formulaire « Gestion Stock » en suivant la démarche suivante :

- Ouvrir le formulaire « menu général » en mode création
- Sélectionner le bouton de commande «saisie des produits» puis cliquer sur «propriétés» du menu «Affichage»
- Dans l'onglet «Evènement», choisir la macro " Ouvrir\_Produits" dans la clause «Sur clic».



Figure : 41

- Fermer la fenêtre

### Constatations

- On ne peut modifier un formulaire que s'il est affiché en mode création.
- On plus des champs des tables, un formulaire peut contenir d'autres objets tels qu'un bouton de commande, une image, une case à cocher, ...etc.
- Chaque objet, dans un formulaire, admet des propriétés qui dépendent de sa nature.
- L'action à exécuter en cliquant sur un bouton de commande doit être définie dans une macro ou bien dans une procédure événementielle.

### Application

Reprendre le formulaire « Gestion Stock » et créer :

- le bouton de commande «saisie des clients » qui, sur clic, ouvre le formulaire « saisie des clients » à l'aide d'une macro.
- le bouton de commande «saisie des commandes » qui, sur clic, ouvre le formulaire « saisie des commandes » à l'aide d'une macro.
- le bouton de commande «saisie des fournisseurs » qui, sur clic, ouvre le formulaire « saisie des fournisseurs » à l'aide d'une macro.
- le bouton de commande «les requêtes » qui, sur clic, ouvre le formulaire « les requêtes » à l'aide d'une macro.
- le bouton de commande «les états » qui, sur clic, ouvre le formulaire « les états » à l'aide d'une macro.

## VI. Création et impression d'états

Dans une base de données, l'objet état est utilisé pour mettre en forme les données destinées à être imprimées. Lorsque la matérialisation de données issue d'un SGBD présente un caractère répétitif, et plus encore lorsqu'il est envisagé d'automatiser cette opération, le recours à un état constitue la meilleure solution. A ce titre, l'état constitue donc le troisième objet des SGBD par ordre d'importance décroissante, après les tables et les requêtes.

Un état est pratiquement toujours construit sur le résultat d'une requête, et ce pour les raisons suivantes :

- a. les bases de données contiennent souvent des quantités considérables d'informations, et il n'est pas question de tout imprimer. Il faut donc commencer par sélectionner l'information particulière que l'on veut reproduire avant d'imprimer ;
- b. dans une base de données, les informations sont réparties dans des tables multiples, et il faut les rassembler avant de les imprimer. On peut, donc, introduire dans un même état des champs provenant de plusieurs tables liées par des relations ;
- c. souvent, les informations à imprimer doivent se présenter dans un certain ordre. Il faut donc réaliser un tri lors de la création de l'état.

Dans la base de données « GESTION STOCK », la figure suivante représente un état sur les produits fournis par un fournisseur donné.

<i>Liste des produits par fournisseur</i>				
vendredi 12 janvier 2007				
<i>Nom</i>	<i>Réf prod</i>	<i>Nom du prod</i>	<i>Quantité</i>	<i>Prix unitaire</i>
<b>Abassi Maram</b>				
	<b>3</b>	<b>Plateau GM</b>	<b>10</b>	<b>50,5</b>
	<b>1</b>	<b>Hricot</b>	<b>15</b>	<b>10,5</b>
	<b>9</b>	<b>Popinel GM</b>	<b>3</b>	<b>150</b>
	<b>8</b>	<b>Boite à instrument P</b>	<b>5</b>	<b>65</b>
	<b>5</b>	<b>Verre à pied</b>	<b>15</b>	<b>5</b>
	<b>4</b>	<b>pèse bébé</b>	<b>3</b>	<b>420</b>

Figure : 42

Comme pour un formulaire, un état admet deux modes : « Mode création » et mode affichage appelé « Aperçu avant impression ».

Dans le mode création, on distingue, dans la figure suivante, cinq zones :

Figure : 43

- En-tête d'état : cette zone est imprimée en haut de la première page uniquement ;
- En-tête de page : cette zone est imprimée en haut de toutes les pages ;
- Détail : cette zone permet d'imprimer le contenu des requêtes ;
- Pied de page : cette zone est imprimée en bas de chaque page ;
- Pied d'état : cette zone s'imprime uniquement après le détail de la dernière page ;

En mode création, on distingue trois catégories de zone de texte :

- la zone de texte dépendante : son contenu provient d'un champ d'une table ou d'une requête. Les zones de texte dépendantes se trouvent généralement dans la zone "Détail" de l'état ;
- la zone de texte indépendante : son contenu ne provient pas d'un objet de la base de données. On l'utilise pour afficher un texte informatif (exemple : le titre de l'état), une image (exemple : le logo de l'entreprise), ou des éléments de décoration, principalement dans l'en-tête de l'état ;
- la zone de texte calculée : elle contient le résultat d'un calcul (exemples : somme, moyenne, fonctions statistiques, date, page, etc.). Ce résultat est remis à jour automatiquement chaque fois que les données utilisées dans le calcul sont modifiées. Les zones de texte calculées se trouvent généralement dans les pieds de page ou dans le pied de l'état.

**Activité : 15**

Reconnaître le menu et les commandes nécessaires pour créer l'état de la figure 44

**Solution**

L'état qu'on vient de créer utilise comme source de données une requête paramétrée qui a comme résultat tous les produits fournis par un fournisseur donné. Donc on doit commencer par créer cette requête :

Création de la requête paramétrée « liste des produits »

- Choisir l'objet « Requetes » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Mode création »
- Ajouter les tables PRODUITS et FOURNISSEURS
- Dans la grille de création de la requête, fixer les champs ainsi que le paramètre « Nom du fournisseur : »

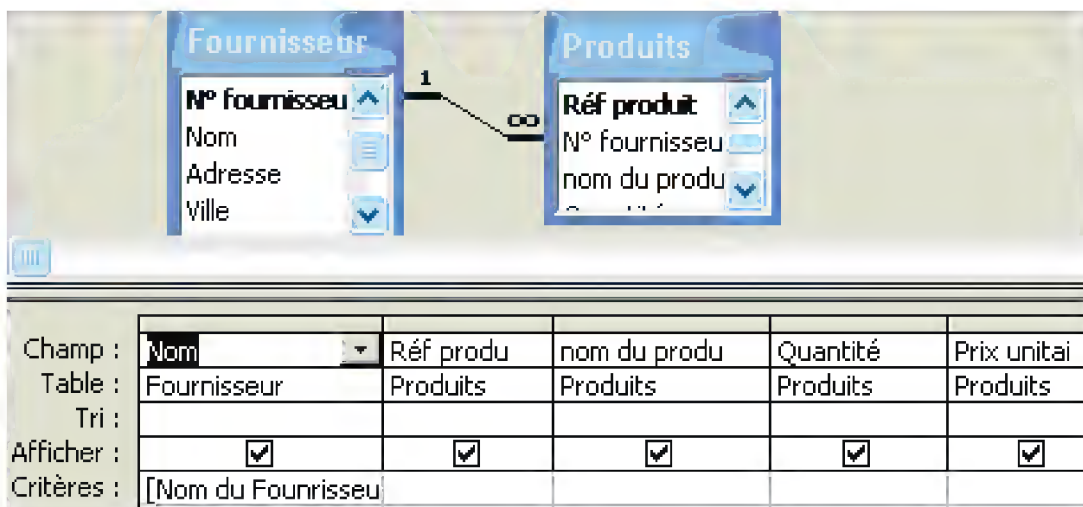


Figure : 44

- Enregistrer la requête sous le nom " liste des produits".

**Création de l'état**

- Choisir l'objet « Etats » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Assistant Etat»
- Sélectionner la source de données souvent une requête. Dans ce cas la requête "liste des produits" ; puis valider.



- Choisir l'objet « Etats » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Assistant Etat»
- Sélectionner la source de données souvent une requête. Dans ce cas la requête 'liste des produits' ; puis valider.

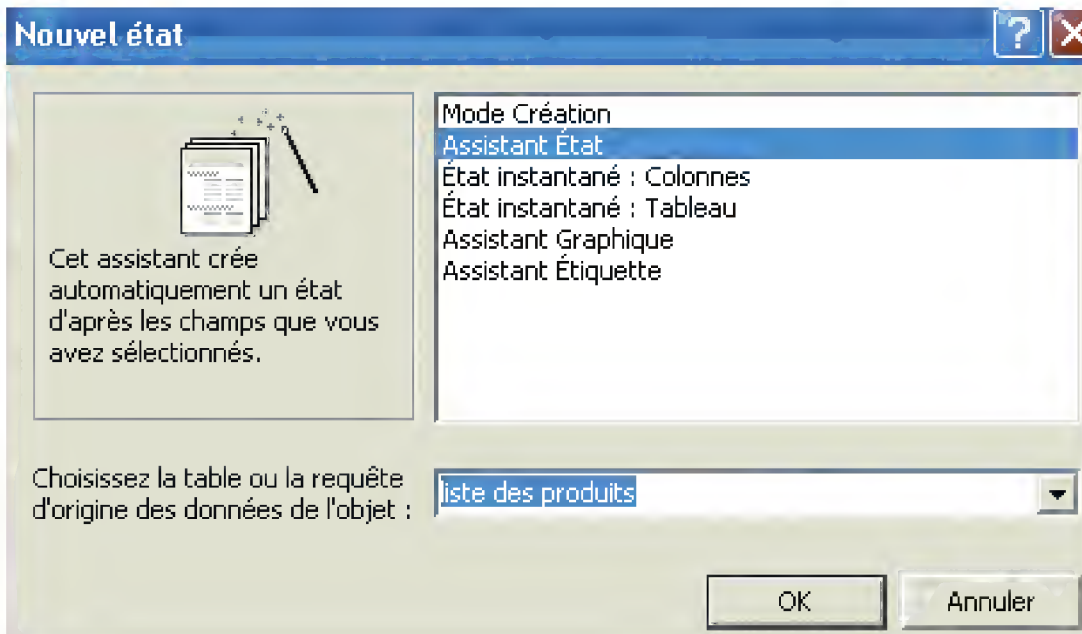


Figure : 45

- Suivre les instructions de l'assistant pour :
  - sélectionner les champs,
  - afficher des données,
  - fixer le niveau de regroupement des champs,
  - spécifier l'ordre de tri des données
  - choisir la disposition des données (échelonné, bloc, contour) ainsi que l'orientation de la page (paysage ou portrait)
  - choisir un style pour l'état
  - visualiser l'aperçu de l'état
- Enregistrer l'état sous le nom 'liste des produits par fournisseur'.

L'état s'affiche en mode « Aperçu avant impression », on peut basculer vers le mode création à l'aide de la commande « Mode création » du menu « Affichage » pour mettre en forme les champs de l'état, ajouter des lignes séparatrices, ajouter la date ou une image, ...

**Affichage de l'état "liste des produits par fournisseur"**

- Double cliquer sur l'état "liste des produits par fournisseur" dans la fenêtre base de données. La boîte de dialogue suivante s'affiche :
- Taper le paramètre de la requête source de l'état :

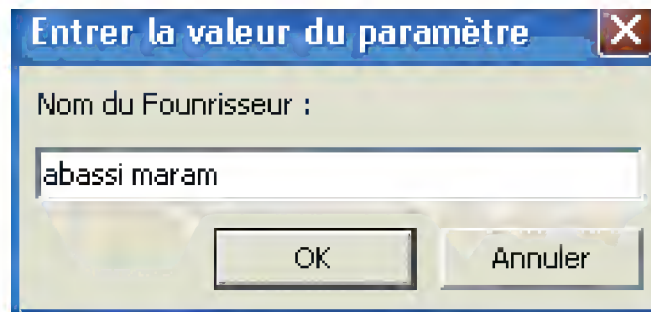


Figure : 46

- Après avoir cliqué sur le bouton "OK", l'état suivante sera afficher en mode "Aperçu avant impression"

***Liste des produits par fournisseur***

vendredi 12 janvier 2007

<i>Nom</i>	<i>Réf prod</i>	<i>Nom du prod</i>	<i>Quantité</i>	<i>Prix unitaire</i>
<b>Abassi Maram</b>				
<b>3</b>		<b>Plateau GM</b>	<b>10</b>	<b>50,5</b>
<b>1</b>		<b>Hricot</b>	<b>15</b>	<b>10,5</b>
<b>9</b>		<b>Popinel GM</b>	<b>3</b>	<b>150</b>
<b>8</b>		<b>Boite à instrument P</b>	<b>5</b>	<b>65</b>
<b>5</b>		<b>Verre à pied</b>	<b>15</b>	<b>5</b>
<b>4</b>		<b>pèse bébé</b>	<b>3</b>	<b>420</b>

Figure : 47

**VII. Exercice de synthèse : problème corrigé**

On s'intéresse, dans ce problème, à la base de données « Gestion des records » stockant les performances des différents athlètes participant aux meetings de la Fédération Internationale d'Athlétisme.

La base de données « Gestion des records » est décrite par les relations suivantes :

**ATHLETE (NumLicence, Nom, Prénom, NumFéd)**  
**FEDERATION (NumFédération, NomPays, Continent)**  
**RESULTAT (NumLic, NomMeeting, Epr, Rang)**  
**RECORD\_DU\_MONDE (Epreuve, NumLic, Valeur)**

Où :

- L'attribut NumFéd dans ATHLETE correspond au NumFédération dans FEDERATION.
- L'attribut NumLic dans RESULTAT correspond au NumLicence dans ATHLETE.
- L'attribut Epr dans RESULTAT correspond à Epreuve dans RECORD\_DU\_MONDE.

La table RECORD\_DU\_MONDE contient le record du monde actuel pour chaque épreuve.

L'attribut Epreuve a des valeurs comme «100m masculin», «Saut en hauteur féminin» à qui correspond une valeur du record. Par exemple, pour « 100m masculin » la valeur du record est « 9s79 ».

### Travail demandé :

On demande de donner les démarches à suivre qui permettent de répondre aux questions suivantes :

1. Créer les tables de la base de données « Gestion des records »
2. Créer les relations entre les tables de la base de données
3. Créer les requêtes suivantes :
  - a. Donner les athlètes d'un pays donné (le nom du pays doit être saisi au clavier)
  - b. Donner le nom de l'athlète qui détient le record du monde du 10000m masculin
4. Créer l'état qui permet d'imprimer la liste des athlètes d'un pays donné
5. Créer les formulaires suivants :
  - a. menu principal : contenant le titre ' Gestion des records ', trois boutons de commandes :
    - le 1er pour afficher le formulaire de saisie des athlètes
    - le 2ème pour afficher le formulaire de saisie des résultats
    - le 3ème pour afficher un aperçu avant impression de l'état créé dans la question 4
  - b. un formulaire de saisie des athlètes
  - c. un formulaire de saisie des résultats

## Solution

### 1. Créer les tables de la base de données "Gestion des records".

Pour créer cette table on peut suivre la démarche suivante :

- Sélectionner l'objet «Tables» de la fenêtre « base de données »
- Cliquer sur le bouton «Nouveau»
- Choisir l'option « Mode création » puis confirmer
- Créer les champs de la table ATHLETE et fixer pour chacun ces propriétés :
  - ✓ Saisir le champ "NumLicence" et fixer les propriétés : clé primaire, type de données = texte, taille du champ = 4 caractères,
  - ✓ Saisir le champ "Nom" et fixer les propriétés : type de données = texte, taille du champ = 20 caractères,
  - ✓ Saisir le champ "Prénom" et fixer les propriétés : type de données = texte, taille du champ = 20 caractères,
  - ✓ Saisir le champ "NumFéd" et fixer les propriétés : type de données = texte, taille du champ = 3 caractères,
- Enregistrer la table sous le nom ATHLETE, puis la fermer.

Refaire cette démarche pour créer les autres tables de la base de données "Gestion des records" : ATHLETE, FEDERATION, RESULTAT et RECORD\_DU\_MONDE.

### 2. Selon le schéma relationnel de la base, on distingue les trois relations suivantes :

- a. La première entre la table FEDERATION et la table ATHLETE : la clé primaire de cette relation est "NumFédération" et la clé étrangère est "NumFéd"
- b. La deuxième entre la table ATHLETE et la table RESULTAT : la clé primaire de cette relation est "NumLicence" et la clé étrangère est "NumLic"
- c. La troisième entre la table FEDERATION et la table ATHLETE : la clé primaire de cette relation est "Epreuve" et la clé étrangère est "Epr"

Pour créer ces relations on peut suivre la démarche suivante :

- ✓ Ouvrir le menu «Outils» puis cliquer sur la commande «Relations...».
- ✓ Ajouter les tables : ATHLETE, FEDERATION, RESULTAT et RECORD\_DU\_MONDE, puis fermer la boîte de dialogue

-Créer la première relation :

- Maintenir le bouton gauche de la souris enfoncé sur la clé primaire "NumFédération", glisser vers la clé étrangère "NumFéd" puis relâcher le bouton de la souris
- Vérifier, dans la boîte de dialogue qui s'affiche, les champs puis cocher la case « Appliquer l'intégrité référentielle »
- Valider en cliquant sur « Créer »

- Refaire les trois dernières étapes pour créer la deuxième et la troisième relation
- Enregistrer

On aura comme résultat graphique les relations de la figure suivante :

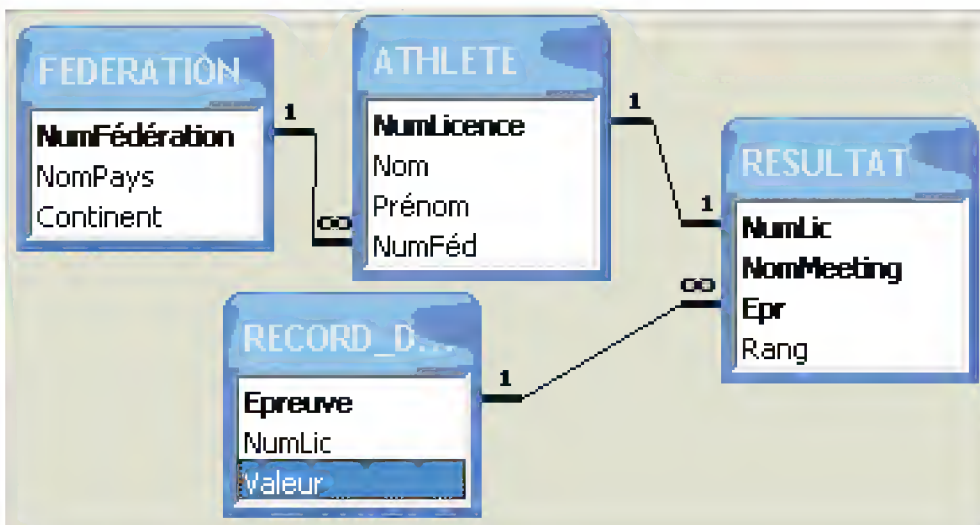


Figure : 48

**d. Donner les athlètes d'un pays donné (le nom du pays doit être saisi au clavier)**

La requête est une requête paramétrée, le paramètre est le nom du pays

- Cliquer sur l'objet « Requêtes » dans la fenêtre « Base de données »
- Cliquer sur le bouton « Nouveau » puis sur « Mode création »
- Ajouter les tables ATHLETE et FEDERATION
- Dans la grille de création de la requête, fixer les champs "NumLicence", "Nom" et "Prénom" de la table ATHLETE et le champ "NomPays" de la table FEDERATION comme indique la figure suivante :

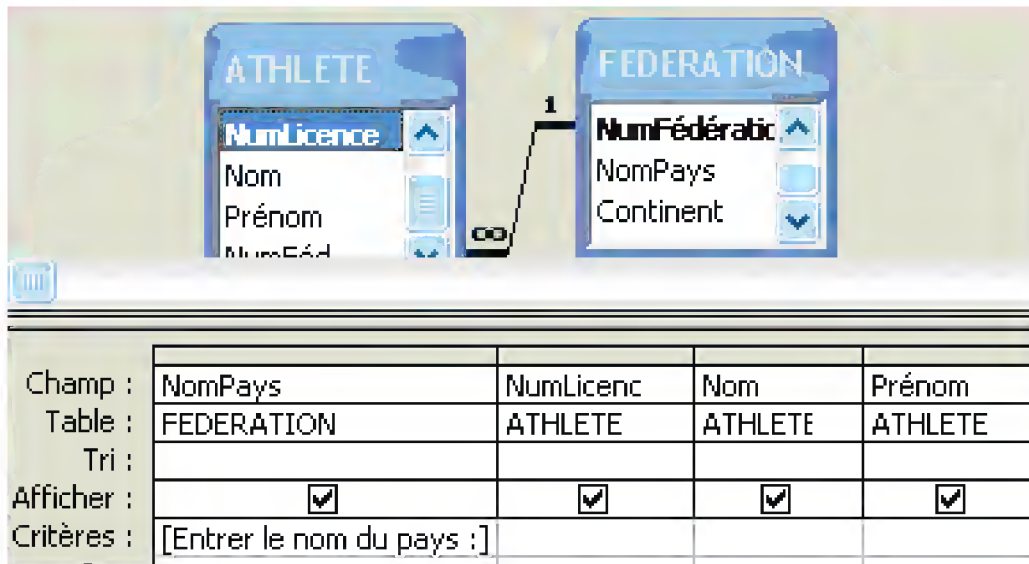


Figure : 49

- Enregistrer la requête sous le nom "liste des athlètes".

**e. Donner le nom de l'athlète qui détient le record du monde du 10000m masculin**

- Cliquer sur l'objet « Requetes » dans la fenêtre « Base de données »
- Cliquer sur le bouton « Nouveau » puis sur « Mode création »
- Ajouter les tables ATHLETE, FEDERATION, RESULTAT et RECORD\_DU\_MONDE
- Dans la grille de création de la requête, fixer le champ "NumLic" de la table RECORD\_DU\_MONDE, les champs "Nom" et "Prénom" de la table ATHLETE, et le champ "NomMeeting" de la table RESULTAT comme indique la figure suivante :

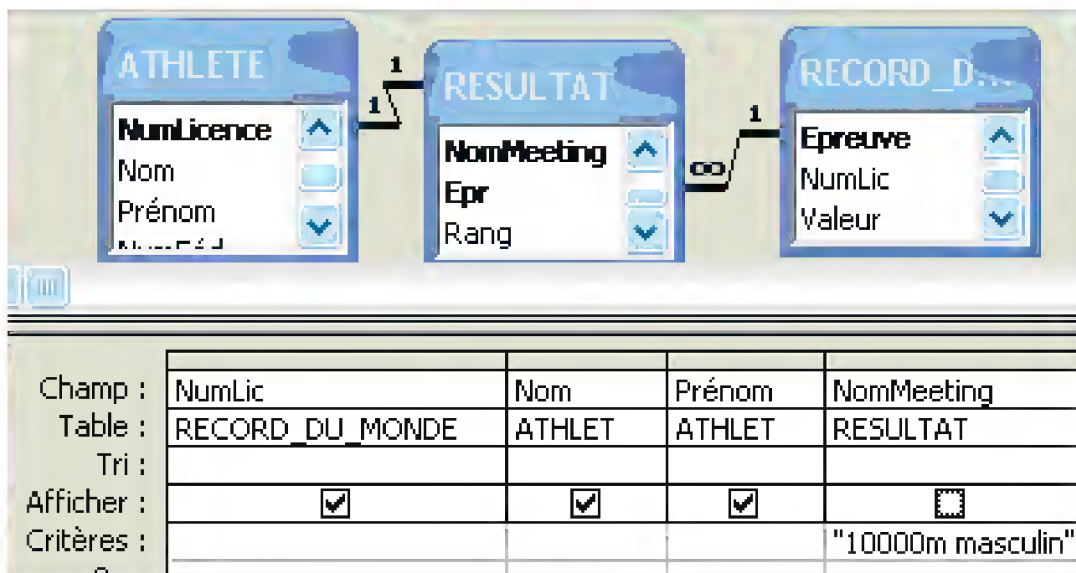


Figure : 50

- Enregistrer la requête sous le nom "record du monde".

### 3. Création de l'état

- Choisir l'objet « Etats » de la fenêtre « Base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode « Assistant Etat»
- Sélectionner la requête 'liste des athlètes'; puis valider.
- Suivre les instructions de l'assistant pour :
  - ✦ sélectionner les champs,
  - ✦ afficher des données,
  - ✦ fixer le niveau de regroupement des champs,
  - ✦ spécifier l'ordre de tri des données
  - ✦ choisir la disposition des données et le style de l'état
- Réaliser les mises en forme et les mises en pages de l'état en mode création.
- Enregistrer l'état.

### 4. Création des formulaires

#### a. menu principal

Les boutons de commande, nécessitent la création des macros qui permettent l'ouverture des formulaires et l'impression de l'état. Donc on doit commencer par créer les macros.

#### Création des macros

- Choisir l'objet « Macro » de la fenêtre base de données
- Cliquer sur le bouton « Nouveau »
- Dans la colonne « Action » fixer l'action ("OuvrirFormulaire" pour les deux premiers boutons et "OuvrirEtat" pour le troisième bouton)
- Fixer le nom du formulaire pour chaque macro

#### Création des formulaires

- Choisir l'objet « Formulaires » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau » puis choisir le mode de création
- Afficher le formulaire en mode création puis créer les trois boutons de commande de la manière suivante :
  - ✦ Choisir l'outil « Bouton de commande » de la boîte à outils
  - ✦ Le placer dans le formulaire
  - ✦ Accéder à la propriété « Sur clic » de l'onglet « Evènement »
  - ✦ Choisir la macro correspondante
  - ✦ Fermer la boîte de dialogue des propriétés.
  - ✦ Refaire les étapes précédentes pour créer le 2ème bouton de commande en choisissant comme source de données la table RESULTAT.
  - ✦ Refaire les étapes précédentes pour créer le 3ème bouton de commande en choisissant comme source de données l'état à imprimer.

### b. Formulaire de saisie des athlètes

- Choisir l'objet « Formulaires » de la fenêtre « base de données »
- Cliquer sur le bouton « Nouveau »
- Fixer la table ATHLETE, source des données, puis valider.
- Choisir la liste des champs, le style et le nom du formulaire puis l'enregistrer.

### c. Formulaire de saisie des résultats :

Appliquer la même démarche pour créer le formulaire de saisie des résultats.

### Exercices non corrigés

#### Exercice 1

Soit une base de données de gestion des projets qui contient les relations suivantes:

Employé (Matr, NomE, Grade, DatEmb, Sup, Salaire, Commission, NDept)

Département (NumDept, NomDept, Lieu)

Projet (CodeP, NomP)

Participation (MatrEmp, CdeP, Heures)

Où :

- L'attribut NDept dans Employé correspond au NumDept dans Département.
- L'attribut CdeP dans Participation correspond au CodeP dans Projet.
- L'attribut MatrEmp dans Participation correspond au Matr dans Employé.

#### Travail demandé :

- 1) Créer les tables de la base de données ainsi que les relations qui les relient.
- 2) Créer les requêtes suivantes :
  - liste des départements de Tunis
  - les noms des employés attachés à un département donné
- 3) Créer, pour chaque requête un état permettant l'impression des données résultat.

#### Exercice 2

Soit le schéma de la base de données 'compétition' suivante :

COUREUR (NuméroCoureur, NomCoureur, CodeEquipe, CodePays)

EQUIPE (CodeEquipe, NomEquipe, DirecteurSportif)

PAYS (CodePays, NomPays)

ETAPE (NuméroEtape, Date, VilleDep, VilleArr, NbKm)

PARTICIPER (NuméroCoureur, NuméroEtape, TempsRéalisé).



2. Créer les tables de la base de données.
3. Créer un formulaire de saisie pour les entités COUREUR, EQUIPE, ETAPE et PARTICIPER.
4. Créer un formulaire qui joue le rôle d'un menu principal de la base de données.
5. Créer un état permettant l'impression de tous les coureurs tunisiens.

### Exercice 3

Une entreprise implantée à l'échelle nationale, spécialisée dans la vente de produits informatiques, possède une base de données pour la gestion de son service de vente. Le schéma relationnel de cette base de données est le suivant :

Projets (**num\_projet**, désignation, succursale, budget)

Ventes (**num\_vendeur**, **num\_pièce**, **num\_projet**, **quantité**, **date\_vente**)

Pièces (**num\_pièce**, désignation, poids, num\_succ\_stock, prix\_unitaire, quantité\_stockée, seuil\_stock)

Succursales (**num\_succursale**, désignation, ville, surface, capital)

Vendeurs (num\_vendeur, nom, prénom, ville, succursale, salaire\_fixe, commission)

Les clés primaires des différentes relations sont indiquées en gras et les clés étrangères sont indiquées en italique.

Num\_succ\_stock dans la relation Pièces référence la succursale de stockage de la pièce.

Succursale dans la relation Vendeurs référence la succursale de rattachement du vendeur.

Succursale dans la relation Projets référence la succursale qui a en charge le projet.

Le triplet num\_vendeur, num\_pièce, num\_projet dans la table Ventes référence le vendeur qui a effectué la vente, le numéro de la pièce objet de la vente et le numéro du projet auquel est destinée la pièce.

a. Créer la base de données (tables et relations)

b. Ecrire les requêtes suivantes :

✦ Liste des pièces qui ont atteint "seuil\_stock"

✦ Liste de tous les vendeurs des pièces fournis pour le compte d'un projet donné.

c. Créer le formulaire de saisie des ventes.

d. Créer un état qui permet d'imprimer les ventes réalisées durant le mois de janvier 2007

### Exercice 4

On se propose de réaliser une base de données pour la gestion du parc auto d'une société. On dispose du schéma relationnel suivant :

VEHICULE (**Num\_Immat**, Date\_achat, Code\_modèle, Consom\_moyenne)

MODELE (**Code\_modèle**, Désignation, Code\_constructeur, Puissance)

CONSTRUCTEUR (**Code\_constructeur**, Nom, Adresse)

PRIX\_ACHAT (**Date**, **Code\_modèle**, Prix)

REPARATION (**Num\_réparation**, Date\_entrée, date\_sortie, Num\_immat, coût\_rép)

PIECE ( **Code\_pièce**, Désignation, Quantité\_en\_stock, prix)

CONSOMMATION\_CARBURANT (**Num\_immat**, Date, Quantité\_Carburant)

CONSOMMATION\_PIECE (**Num\_réparation**, **Code\_pièce**, Quantité, Prix\_unitaire)

### Questions

Créer la base de données, les tables et les relations qui les relient et les requêtes suivantes :

- i. les véhicules dont le coût de réparation dépasse 500 D
- ii. la quantité de carburant utilisée pour un véhicule donné à une date donnée.

**LECTURE****BASES DE DONNEES  
INTRODUCTION ET DEFINITIONS**

L'informatique évolue vers le traitement de masses d'informations de plus en plus grandes dans des environnements répartis géographiquement où doivent cohabiter des matériels hétérogènes. Dans ce contexte, les bases de données sont utilisées de façon intensive pour de nombreux domaines d'application tels que le domaine médical, les administrations ou les associations. Les applications concernées par l'utilisation d'un SGBD possèdent des caractéristiques différentes tant au niveau du volume de données concernées qu'au niveau de la complexité de ces données et des traitements informatiques à réaliser. Néanmoins, le regroupement des données dans une base de données gérée par un système de gestion de base de données apporte de nombreux avantages dans la plupart des cas d'utilisation. De manière intuitive, il est possible de définir une base de données de la façon suivante :

**Base de données**

Une base de données est une collection de données sur un domaine d'application particulier où les propriétés des données ainsi que les relations sémantiques entre ces données sont spécifiées en utilisant les concepts proposés par le modèle de données sous-jacent.

**Modèle de données**

La notion de modèle de données est essentielle et c'est elle qui souvent motive le choix de l'utilisation d'une base de données. En effet, la résolution d'un problème par un automate nécessite de représenter l'information sur le domaine traité appelé parfois mini-monde ou univers du discours sous une forme digitale qui soit interprétable et manipulable par un ordinateur. Le modèle doit donc être spécifié en utilisant des données codées et stockées en mémoire ainsi que par des opérations (programmes) qui déterminent comment ces données peuvent être utilisées pour résoudre le problème posé [Diettrich94]. Un modèle peut se définir comme une représentation abstraite de l'information et éventuellement des opérateurs de manipulation de l'information.

**Système de Gestion de Bases de données**

Un système de gestion de bases de données (SGBD) est une collection de logiciels permettant de créer, de gérer et d'interroger efficacement une base de données indépendamment du domaine d'application.

D'un point de vue fonctionnel, les apports escomptés d'un SGBD sont les suivants :

**- Supporter les concepts définis au niveau du modèle de données.**

Ceci afin de pouvoir représenter les propriétés des données. Ce niveau de représentation n'est pas nécessairement lié à la représentation interne sous forme de fichiers. Il regroupe en général la définition de types spécifiques et la définition de règles de cohérence.

**- Rendre transparent le partage des données entre différents utilisateurs.**

Ceci signifie que plusieurs utilisateurs doivent pouvoir utiliser la base de façon concurrente et transparente. Le problème posé ici est du au fait que le SGBD pour des raisons évidentes de performances (partage du CPU) doit permettre des exécutions concurrentes sur une même base de données.

**- Assurer la confidentialité des données.**

Il est nécessaire de pouvoir spécifier qui a le droit d'accéder ou de modifier tout ou partie d'une base de données. Il faut donc se prémunir contre les manipulations illicites qu'elles soient intentionnelles ou accidentelles. Cela nécessite d'une part une identification des utilisateurs, la constitution de groupes d'utilisateurs avec des recouvrements potentiels et d'autre part, une spécification des droits ajout, suppression, mise à jour). Il est patent que garantir la confidentialité des données engendre un surcoût en temps au niveau des manipulations.

**- Assurer le respect des règles de cohérence définies sur les données.**

A priori, après chaque modification sur la base de données, toutes les règles de cohérence doivent être vérifiées sur toutes les données. Evidemment, une telle approche est irréalisable pour des raisons de performances et il faut déterminer des moyens de trouver précisément quelles règles et quelles données sont susceptibles d'être concernées par les traitements réalisés sur la base de données. Ces traitements doivent pouvoir être effectués sans arrêter le système.

**- Fournir différents langages d'accès selon le profil de l'utilisateur.**

En général, on admet que le SGBD doit au moins supporter un langage adressant les concepts du modèle. Dans le cas du modèle relationnel, ce langage est le langage SQL. Néanmoins ce type de langage ne permet pas tous les types de manipulation et les SGBD proposent soit un langage plus complet au sens Turing du terme avec la possibilité de définir des accès à la base de données, soit un couplage d'un langage tel que SQL avec un langage de programmation conventionnel (tels que le langage C ou le langage Cobol). La définition d'une interface entre une base de données et le Web pose ce type de problème de spécification et de navigation dans une base de données.

**- Etre résistant aux pannes.**

Ceci afin de protéger les données contre tout incident matériel ou logiciel qu'il soit intentionnel ou fortuit. Il faut donc garantir la cohérence de l'information et des traitements en cas de panne. Les applications opérant sur des bases de données sont souvent par nature amenées à opérer des traitements longs sur d'importants volumes de données. Les possibilités de panne en cours de traitement sont donc nombreuses et il faut fournir des mécanismes de reprise en cas de panne.

**- Posséder une capacité de stockage élevée**

Permettre ainsi la gestion de données pouvant atteindre plusieurs milliards d'octets. Les capacités de stockage des ordinateurs sont en augmentation croissante. Cependant, les besoins des utilisateurs sont également en croissance forte. Avec l'essor des données multimédia (texte, image, son, vidéo) les besoins sont encore accrus. Les unités de stockage sont passées du Méga-octets (10<sup>6</sup>) au Giga-octets (10<sup>9</sup>), puis au Téra-octets (10<sup>12</sup>), Peta-octets (10<sup>16</sup>), et l'on commence à parler d'Exa-octets (10<sup>18</sup>) voir de Zetta-octets (10<sup>21</sup>).

**- Pouvoir répondre à des requêtes avec un niveau de performances adapté.**

Une requête est une recherche d'information à effectuer sur une ou plusieurs bases de données qui peut impliquer des caractéristiques descriptives sur l'information ou des relations entre les données. La puissance des ordinateurs n'est pas la seule réponse possible à apporter aux problèmes de performance. Une requête peut généralement être décomposée en opérations élémentaires. L'ordre d'exécution des opérations en fonction de leurs propriétés (associativité, commutativité) ainsi que le regroupement de certaines opérations utilisant le même ensemble de données sont des éléments qui permettent de diminuer significativement le temps d'exécution d'une requête.

**- Fournir des facilités pour la gestion des méta-données.** Par exemple à travers un dictionnaire de données ou un catalogue système. Les méta-données concernent les données sur le schéma de la base de données (relations, attributs, contraintes, vues), sur les données (vues), sur les utilisateurs (identification, droits) et sur le système (statistiques). Ces données doivent être gérées et consultées de la même manière que les données afférentes à l'application. Cette notion de catalogue assure également une certaine flexibilité au niveau de l'utilisation du SGBD. Cette flexibilité permettant l'ajout sous contrôle de nouveaux utilisateurs ainsi que la modification de structures de données existantes sous certaines conditions. De plus, ce type d'information permet entre autre à l'administrateur de la base de données ou au SGBD lui-même d'adapter la politique de stockage en fonction du contenu.

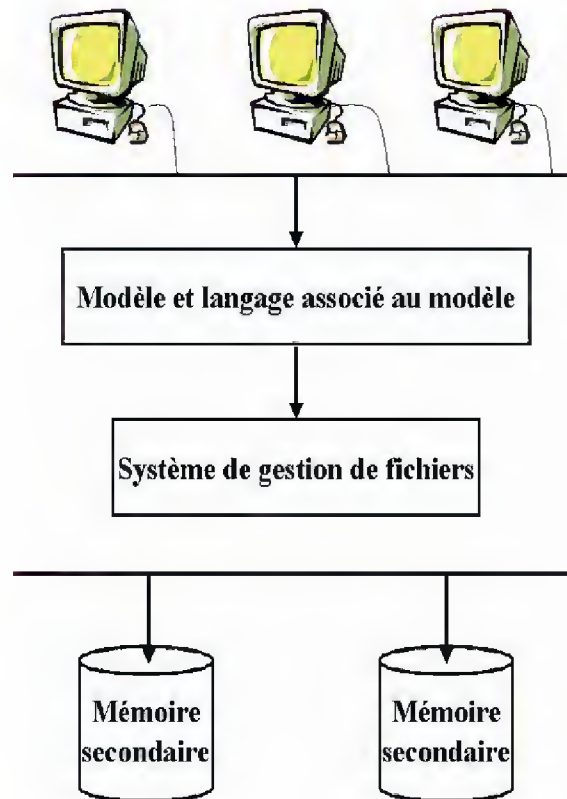
Il existe des SGBD de complexité variable qui possèdent tout ou partie des propriétés ci-dessus. Prenons en exemple deux produits assez caractéristiques : le SGBD relationnel Oracle 7 et le SGBD relationnel Access. Le SGBD Oracle 7 est un SGBD relationnel utilisé pour des applications critiques et qui offre un maximum des caractéristiques présentées ici. Le SGBD Access est un SGBD dans le monde de l'informatique individuelle qui présente l'avantage d'une grande facilité d'utilisation et qui peut convenir à des applications de taille réduite ou moyenne. L'aspect convivial de ce dernier étant évident. En revanche, les niveaux de performance et de sécurité ne sont pas comparables.

### **Architecture de SGBD**

Les SGBD reposent sur trois niveaux d'abstraction qui assurent l'indépendance logique et physique des données, autorisent la manipulation de données, garantissent l'intégrité des données et optimisent l'accès aux données. L'architecture ANSI/SPARC [ANSI75] spécifie cette architecture à trois niveaux pour un SGBD :

- **Le niveau externe.** Il regroupe toutes les possibilités d'accès aux données par les différents usagers. Ces accès, éventuellement distants, peuvent se faire via différents types d'interfaces et langages plus ou moins élaborés. Ce niveau détermine le schéma externe qui contient les vues des utilisateurs sur la base de données c'est à dire le sous-ensemble de données accessibles ainsi que certains assemblages d'information et éventuellement des informations calculées. Il peut donc exister plusieurs schémas externes représentant différentes vues sur la base de données avec des possibilités de recouvrement.
- **Le niveau conceptuel.** Il correspond à la vision des données générale indépendante des applications individuelles et de la façon dont les données sont stockées. Cette représentation est en adéquation avec le modèle de données utilisé. Dans le cas des SGBD relationnels, il s'agit d'une vision tabulaire où la sémantique de l'information est exprimée en utilisant les concepts de relation, attributs et de contraintes d'intégrité. Le niveau conceptuel est défini au travers du schéma conceptuel.
- **Le niveau physique.** Il regroupe les services de gestion de la mémoire secondaire. Il s'appuie sur un système de gestion de fichiers pour définir la politique de stockage ainsi que le placement des données. Cette politique est définie en fonction des volumes de données traitées, des relations sémantiques entre les données ainsi qu'en fonction de l'environnement matériel disponible. Comme le suggère la figure 1, il est tout à fait possible de répartir les données sur différents supports de stockages distribués sur un réseau. Le niveau physique est donc responsable du choix de l'organisation physique des fichiers ainsi que de l'utilisation de telle ou telle méthode d'accès en fonction de la requête. Ce niveau doit également assurer le partage des ressources, la gestion de la concurrence et des pannes. La personne responsable de ce niveau est un administrateur de bases de données. Son rôle est à la fois d'assurer la mise en place et le contrôle des procédures systèmes liées à la gestion de la base mais aussi de gérer les droits d'accès à la base.

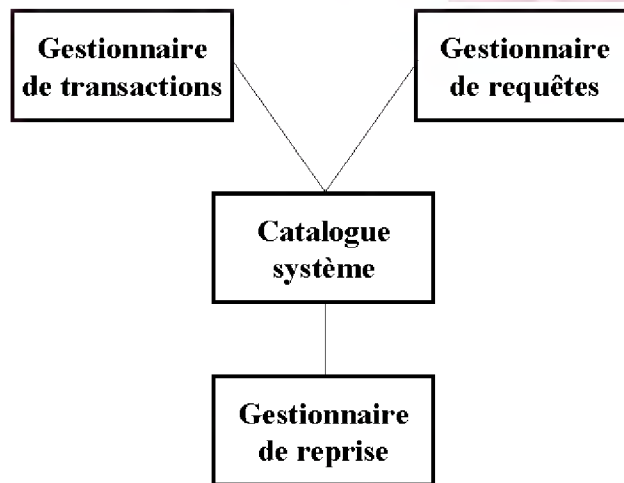
Cette architecture vise à identifier différents niveaux de structuration dans un SGBD comme cela est illustré sur la figure 1. L'objectif visé par cette structuration est l'obtention d'une indépendance maximale entre les niveaux.



- **Indépendance logique.** Possibilité de modifier le schéma conceptuel sans remettre en cause les schémas externes ou les programmes d'application. L'ajout de nouveaux concepts ne doit pas modifier des éléments qui n'y font pas explicitement référence.

- **Indépendance physique.** Possibilité de modifier le schéma physique et la politique de stockage (modification de l'organisation physique des fichiers, ajout ou suppression de méthodes d'accès) sans remettre en cause le schéma conceptuel et donc le schéma externe. Le but de ce niveau d'indépendance est de rendre transparente la gestion physique des données aux programmes d'application.

Cette architecture logique permet donc d'identifier la logique de structuration d'un système de gestion de bases de données. D'un point de vue fonctionnel, il est possible d'identifier plusieurs composants que l'on retrouve dans la plupart des SGBD. La figure 2 illustre cette composition.



Le catalogue système ou dictionnaire de données est un composant au cœur de la communication entre les autres composants. Il contient toutes les méta-données utiles au système. Ces méta-données correspondent à la description des données (type, taille, valeurs autorisées, etc.), aux autorisations d'accès, aux vues et autres éléments systèmes. Le catalogue correspond à la réalisation de l'architecture à trois niveaux décrite précédemment. Le catalogue contient la description des différents schémas (physique, conceptuel et externes) ainsi que les règles de passage d'un schéma vers l'autre.

Le gestionnaire de requêtes est responsable du traitement des commandes des utilisateurs visant à stocker, rechercher et mettre à jour des données dans la base de données. En utilisant les informations stockées dans le catalogue, il interprète ces requêtes et les traduit en des requêtes d'accès physique aux données susceptibles d'être traitées par le système de gestion de fichiers.

Le gestionnaire de transactions est responsable de traiter les transactions. Une transaction est un ensemble d'opérations d'accès et de mise à jour de données émises par un utilisateur. Ces opérations doivent être traitées comme un tout et le gestionnaire de transaction doit assurer d'une part l'indivisibilité des opérations soumises, la cohérence du système après l'exécution de l'ensemble des opérations, l'isolation des traitements par rapports aux autres traitements qui peuvent être soumis de façon concurrente et enfin la persistance des résultats une fois l'ensemble des opérations achevées.

Le gestionnaire de reprise est un élément essentiel d'un SGBD qui doit remplacer le système de gestion de fichier traditionnel afin de minimiser les effets d'une panne sur la base de données. Un tel système ne peut évidemment pas être sûr ou sécurisé à 100 %. Néanmoins, il est indispensable que le gestionnaire de reprise puisse pallier à certaines pannes qui peuvent avoir différentes causes telle qu'une division par zéro dans un programme, à un problème de disque défectueux ou à une panne d'alimentation. L'objectif essentiel du gestionnaire de reprise est de restaurer la base de données dans un état cohérent. Vue les causes très différentes de panne et les difficultés liées à cette gestion, cet élément est un élément central qui concerne environ 10 % ou plus du code d'un SGBD.