

**République Algérienne Démocratique et Populaire**

**Université Badji Mokhtar Annaba**

**Faculté des Sciences économique et de gestion**

**POLYCOPIE A L'INTENTION DES ETUDIANTS DE 1<sup>ère</sup> année Mater  
Finance et Banque**

**Conception et langages des BDD.**

**Exercices et Applications**

**Module : Informatique et Programmation**

**Elaboré par : Dr. Zoubir Layouni**

**Année 2020/2021**

# TABLE DE MATIERE

<b>TABLE DE MATIERE</b> .....	<b>1</b>
<b>PARTIE 1 SGBD PRINCIPE ET FONCTIONNEMENT</b> .....	<b>2</b>
I GENERALITES SUR LES BASES DE DONNEES .....	3
II OBJECTIFS DE L'APPROCHE SGBD .....	4
III ARCHITECTURE FONCTIONNELLE D'UN SGBD .....	4
IV FONCTIONNEMENT D'UN SGBD .....	6
V PRINCIPAUX MODELES LOGIQUES .....	8
<b>PARTIE 2 CONCEPTION ET DEMARCHE</b> .....	<b>8</b>
I CONCEPTION DE BASES DE DONNEES.....	9
II DEMARCHE DE CREATION D'UNE BASE DE DONNEES .....	10
III LES REDONDANCES .....	13
<b>PARTIE 3 CREATION ET MANIPULATION DES BASES DE DONNEE AVEC MS ACCESS</b> .....	<b>15</b>
I CREATION DES TABLES .....	18
II SAISIE DE DONNEE ET CREATION DE FORMULAIRE .....	26
III MANIPULATION DES DONNEE ET CREATION DES REQUETE .....	30
IV LANGAGE SQL .....	30
V LES FORMULAIRES ET SOUS- FORMULAIRES.....	32
VI LES ETATS .....	32
<b>PARTIE 4 TRAVAUX D'APPLICATION</b> .....	<b>35</b>
INTRODUCTION .....	36
CREATION DE TABLES .....	36
REPLISSAGE DES TABLES .....	36
PROPRIETE DES TABLES .....	37
LES REQUETES .....	38
LES FORMULAIRES1 .....	39
LES FORMULAIRES2 .....	40
LES ETATS .....	40
LES MACROS.....	41
GESTION DES COMMANDES .....	42
<b>PARTIE 5 EXERCICES D'APPLICATIONS (LANGAGE SQL)</b> .....	<b>45</b>
<b>PARTIE 6 EVALUATION</b> .....	<b>50</b>
<b>Exemple d'application : gestion des stages au sein de la faculté</b> .....	<b>85</b>

## PARTIE 1

### SYSTEME DE GESTION DE BASE DE DONNEES, PRINCIPE ET FONCTIONNEMENT

# SGBD PRINCIPE ET FONCTIONNEMENT

## I Généralités sur les bases de données

### Définition et Historique

Une base de données est un ensemble structuré de données enregistrées sur des supports informatisés, pouvant satisfaire simultanément plusieurs utilisateurs de façon sélective, en un délai raisonnable.

Le concept de Base de Données (BDD) est apparu vers 1960, face au nombre croissant d'informations que les entreprises devaient gérer et partager :

Chaque nouvelle application créait alors ses propres fichiers de données et ses propres programmes ;

le concept de base de données va à l'encontre de cette façon de procéder : il permet la centralisation, la coordination, l'intégration et la diffusion de l'information archivée.

La base de données enregistre les faits ou événements qui surviennent dans la vie d'un organisme, pour les restituer à la demande : elle permet également de tirer des conclusions en rapprochant plusieurs faits élémentaires.

Les données peuvent être manipulées par plusieurs utilisateurs ayant des **vues différentes** sur ces données ("points de vue" différents).

La structure d'ensemble des données suit une définition rigoureuse appelée **SCHEMA**.

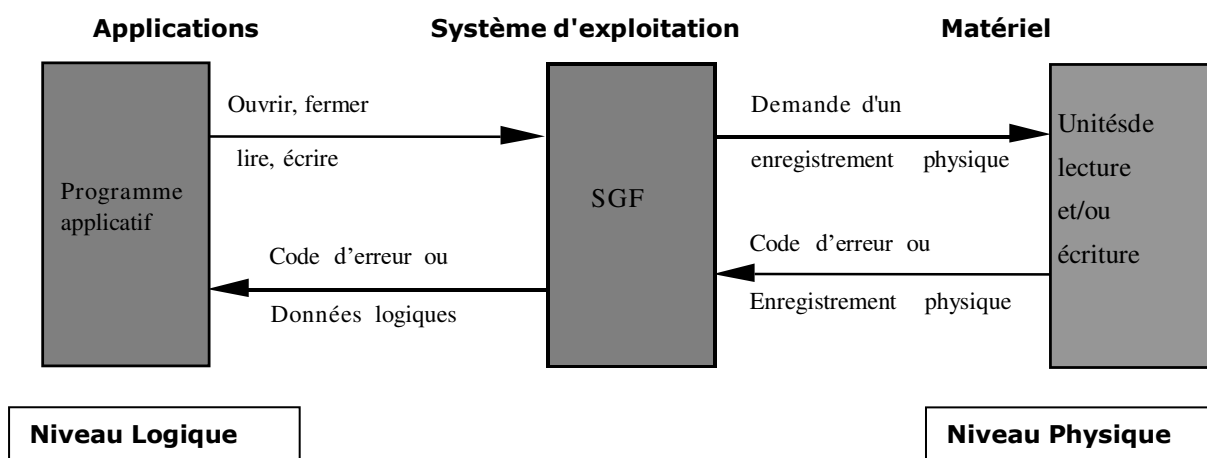
*Facteurs liés au développement des SGBD :*

- augmentation des capacités mémoire, et diminution des temps d'accès
- apparition sur le marché d'applications fiables et diversifiées, qui doivent partager leurs données
- développement des systèmes de gestion en temps réel : "Gestion transactionnelle"
- approche globale "orientée données" des problèmes de gestion : les données sont organisées de façon rationnelle plutôt que définies au coup par coup selon les applications à réaliser.

*Rappel sur les systèmes de gestion de fichiers.*

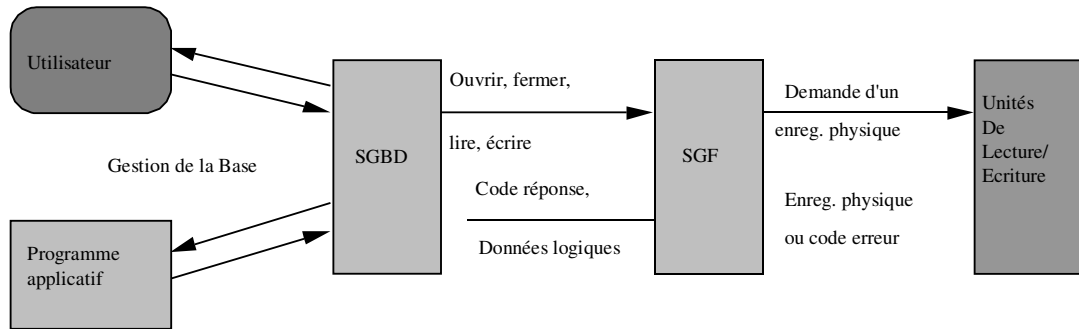
Toute manipulation de fichier exige trois niveaux d'intervention, et trois couches logicielles :

- Gestion du support physique : disques durs, disquette, streamers...  
⇒ *Pilote d'entrées-sorties (Driver)*
- Gestion des structures internes des fichiers, et des méthodes d'accès : ouverture, fermeture, lecture, écriture...  
⇒ *Système de gestion de fichiers (SGF)*
- Gestion des contenus : calculs, tests, affichages ...  
⇒ *Programmes applicatifs*



## Système de Gestion de Base de données : SGBD

- Ensemble des programmes et des langages de commande qui permettent de :
  - définir des "bases de données", et des relations entre les éléments de chaque base ;
  - spécifier le traitement de ces données : interrogations, mises à jour, calculs, extractions...
- Le SGBD reçoit des commandes aussi bien des programmes d'application que des utilisateurs : il commande les manipulations de données, généralement par l'intermédiaire d'un SGF.

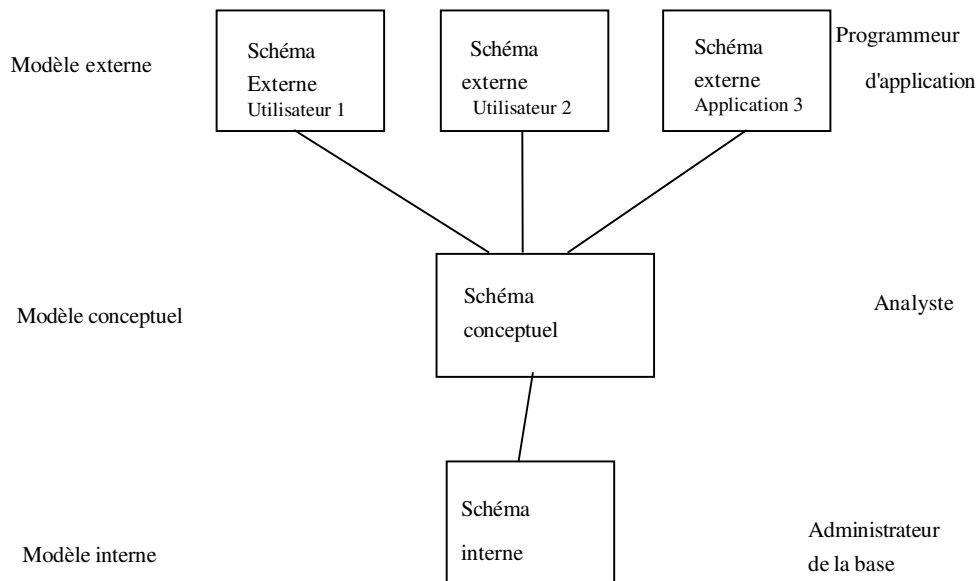


## II Objectifs de l'approche SGBD

- Pour pallier aux inconvénients des méthodes classiques de gestion de fichiers, les SGBD visent quatre objectifs : intégration et corrélation, flexibilité (indépendance), disponibilité, sécurité.
- Ces objectifs exigent une distinction nette entre les données et les procédures de manipulation de ces données : aux données, on associera une fonction **d'administration des données**, aux procédures de manipulation une **fonction de programmation**.

## III Architecture fonctionnelle d'un SGBD :

- architecture multi-niveaux : à chaque niveau fonctionnel, sont associés un modèle et un schéma de données, un langage de description de données (LDD) permettant de décrire les données du schéma, et un langage de manipulation de données (LMD) permettant de les utiliser (accès pour consultation, mise à jour...).

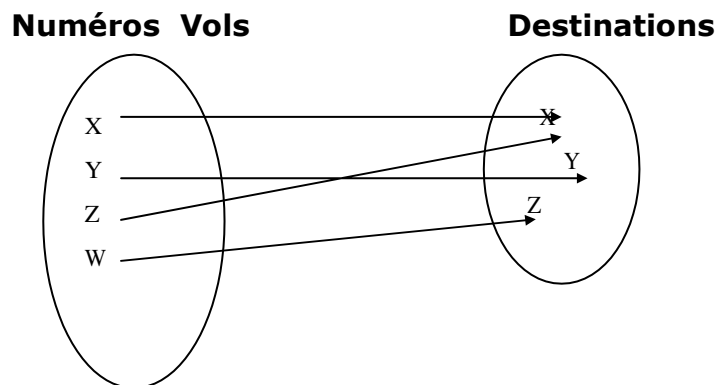


## Niveau conceptuel

- C'est une abstraction aussi fidèle que possible, de l'univers de l'entreprise, après modélisation et indépendamment de toute référence à l'utilisation et à l'implantation en machine. Le modèle conceptuel de données (MCD) permet le passage d'un concret inaccessible (l'univers réel) à un abstrait manipulable : le schéma conceptuel. Celui-ci peut donc être considéré comme la description du contenu de la base : c'est le résultat d'un travail d'analyse et de conception d'un système d'information automatisé.
- Pour aboutir au schéma conceptuel, l'analyste doit repérer dans le réel, et recenser de manière exhaustive, toutes les entités et toutes les associations :
  - Une **entité** peut être définie comme une personne, un objet, un lieu, un statut, un événement qui ont une existence dans le monde réel. C'est un objet concret ou abstrait, possédant un certain nombre de caractéristiques spécifiques (exemple : le produit x coûte y francs).
  - Généralement, les entités du monde réel se manifestent à travers des faits élémentaires.
  - Certains faits faisant intervenir plusieurs entités, il apparaît la notion d'**association**. Une association (ou **lien**) est un ensemble de deux ou plusieurs entités, chacune d'elles jouant un rôle particulier.
  - les liens fonctionnels notés N : 1

On a un lien fonctionnel N:1 de A vers B si toute occurrence de A détermine au plus une occurrence de B, et si à toute occurrence de B, correspond un nombre quelconque d'occurrences de A.

*Exemple : dans une compagnie aérienne, connaissant le numéro d'un vol, on en déduit d'une manière unique la destination, mais plusieurs vols peuvent avoir la même destination.*



- les liens hiérarchiques notés 1 : N.

On a un lien hiérarchique 1:N de A vers B si une occurrence de A peut déterminer un nombre quelconque d'occurrences de B et si, à une occurrence de B, correspond au plus une occurrence de A.

*Exemple : la polygamie est un lien 1 : N de "homme" vers "femme".*

- les liens maillés notés N : M.

On a un lien maillé de A vers B s'il n'existe aucune restriction sur le nombre d'occurrences de A et B intervenant dans le lien.

*Exemple : dans une faculté, un enseignant peut dispenser des cours dans plusieurs modules différents ; de la même façon, un cours peut être dispensé par plusieurs enseignants.*

## IV Fonctionnement d'un SGBD

### *Chronologie des opérations dans l'interrogation d'un SGBD*

Un programme d'application A émet une demande de lecture de données au SGBD sur une des bases :

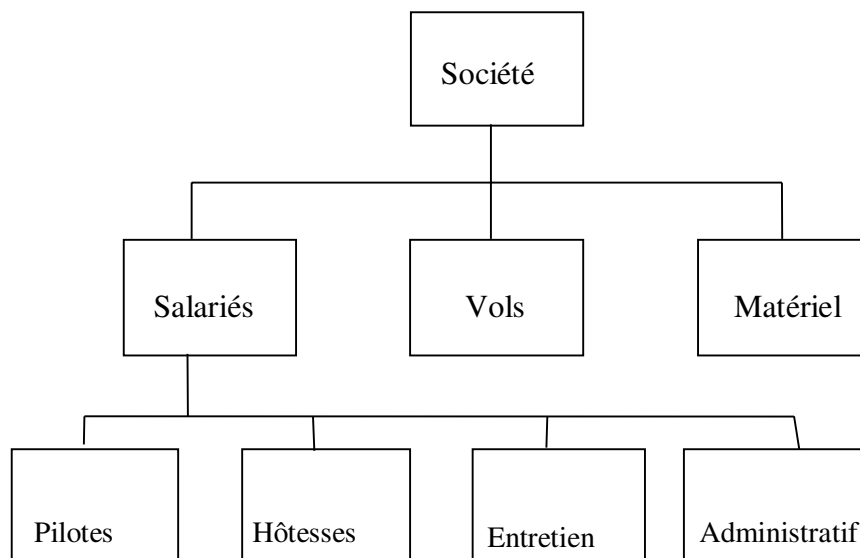
- Le SGBD traite la demande en consultant le sous-schéma externe relatif au programme d'application A, obtenant ainsi la description des données.
- Le SGBD consulte le schéma conceptuel et détermine le type logique de données à extraire.
- Le système examine la description physique de la base en rapport avec la requête logique et détermine le (ou les) enregistrement(s) physique(s) à lire.
- Le système lance une commande au système d'exploitation pour rechercher physiquement l'enregistrement désiré.
- Le système d'exploitation, par le biais de ses méthodes d'accès, accède à l'enregistrement physique.
- Les données demandées sont transférées dans les buffers, ou mémoires tampons.
- Le SGBD, à partir d'une comparaison entre le schéma logique global (conceptuel) et le sous-schéma externe de l'application A, extrait des données stockées dans le buffer, l'enregistrement logique réclamé par le programme d'application. Il effectue également les transformations éventuelles de format.
- Le SGBD transfère les données des buffers dans la zone de liaison du programme d'application A.
- Le SGBD fournit également des informations "d'état" au programme d'application, lui signalant en particulier les erreurs éventuellement constatées au cours du processus d'extraction.
- Le programme d'application, qui dispose des données et d'informations de "service" en assure la bonne exploitation !
- Les ordres d'écriture dans la base physique sont traités par un processus similaire, toute modification ou adjonction étant en général précédée d'une opération de lecture.
- A signaler que, dans la majorité des cas, le SGBD doit traiter simultanément plusieurs demandes de données en provenance de plusieurs programmes d'application, utilisant plusieurs schémas externes différents.

## V Principaux modèles logiques

Les trois principaux modèles sont, dans l'ordre chronologique de leur arrivée sur le marché, le modèle hiérarchique, le modèle réseau (ou navigationnel), le modèle relationnel.

### **Le modèle hiérarchique**

Exemple : le Système d'information d'une compagnie aérienne



## **Le modèle relationnel**

- C'est un article publié en 1969 par un mathématicien du centre de recherche IBM, **Codd**, qui définit les bases de ce modèle relationnel. Codd s'est intéressé au concept d'information et a cherché à le définir sans se préoccuper de la technique informatique, de ses exigences et de ses contraintes. Il a étudié un modèle de représentation des données qui repose sur la notion mathématique de "relation". Dans la pratique, une relation sera représentée par une **table** de valeurs.

*Exemple: représentation d'une table du personnel*

Matricule	Nom	poste	Salaire	N° dept
350	Semache	Pr	118000	320
780	Khames	MCB	115000	870
320	Toumi	MCA	215000	400
490	brahimi	MAA	150100	320

### *Définitions*

- Une relation est un ensemble de tuples (lignes), dont l'ordre est sans importance. Les colonnes de la table sont appelées attributs ou champs. L'ordre des colonnes est défini lors de la création de la table.
- Une clé est un ensemble ordonné d'attributs qui caractérise un tuple. Une clé primaire le caractérise de manière unique, à l'inverse d'une clé secondaire.
- On dit qu'un attribut A est un **déterminant** si sa connaissance détermine celle de l'attribut B (B dépend fonctionnellement de A).

### *Caractéristiques du modèle*

- Schéma de données facile à utiliser : toutes les valeurs sont des champs de tables à deux dimensions.
- Améliore l'indépendance entre les niveaux logique et physique : pas de pointeurs visibles par l'utilisateur.
- Fournit aux utilisateurs des langages de haut niveau pouvant éventuellement être utilisés par des non-informaticiens (SQL, L4G) et un ensemble d'opérateurs basé sur l'algèbre relationnelle : union, intersection, différence, produit cartésien, projection, sélection, jointure, division.
- Optimise les accès aux bases de données
- Améliore l'intégrité et la confidentialité : unicité de clé, contrainte d'intégrité référentielle
- Prend en compte une variété d'applications, en gestion et en industriel
- Fournir une approche méthodologique dans la construction des schémas.



## *PARTIE 2 CONCEPTION ET DEMARCHE*

# Conception et démarche

## I Conception de bases de données

### Les formes normales

Les formes normales permettent de construire un schéma conceptuel correct à partir des relations « brutes » issues des données recueillies auprès des clients.

#### 1<sup>ère</sup> forme normale

**Une relation est dite en première forme normale si chaque attribut possède une seule valeur (ce qui exclut les groupes), et si elle admet une clé primaire.**

#### Exemple:

L'exemple porte sur un ensemble de données concernant des tests de types différents, effectués sur les éléments matériels d'un système de production :

**R (libellé matériel, code marque, libellé marque, type de test, date du test, résultat du test)** n'est pas en 1<sup>ère</sup> forme normale car aucun attribut ne peut être clé primaire : le libellé matériel peut être identique pour plusieurs éléments.

**R (code matériel, libellé matériel, code marque, libellé marque, code type de test, libellé du test, date du test, résultat du test)** n'est pas en 1<sup>ère</sup> forme normale car on peut faire plusieurs tests sur un même matériel, ce qui exige de répéter les informations "code type de test", "libellé du test", "date du test", "résultat du test", dans un même tuple.

La relation doit être éclatée en deux, pour être exprimée en 1<sup>ère</sup> forme normale :

**R-MATERIEL (code matériel, libellé matériel, code marque, libellé marque) R-TEST (code matériel, code type, libellé test, date du test, résultat du test)**

Les deux relations ne comportent que des attributs sans répétition. Dans R\_TEST, la clé primaire est composée de "code matériel" et "code type" : un type de test peut concerner plusieurs matériels, un matériel peut être testé plusieurs fois, mais chaque matériel ne subit qu'une fois un type de test donné.

#### 2<sup>ème</sup> forme normale

**Une relation est dite en deuxième forme normale si elle est en première forme normale, et si tout attribut n'appartenant pas à la clé primaire ne dépend pas que d'une partie de cette clé.**

**R-TEST (code matériel, code type, libellé test, date du test, résultat du test)**

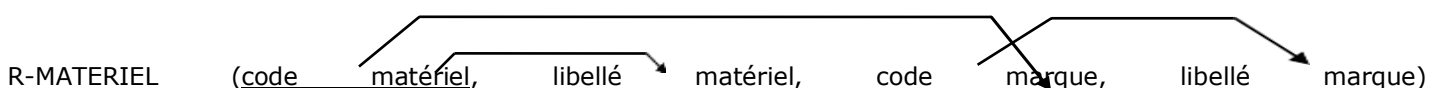
n'est pas en 2<sup>ème</sup> forme normale car l'attribut "libellé test" ne dépend que du "code type" et pas du "code matériel" ;

La relation doit être éclatée en deux, pour être exprimée en deuxième forme normale :

**R-TEST (code matériel, code type, date du test, résultat du test) R-TYPE TEST (code type, libellé test)**

#### 3<sup>ème</sup> forme normale

**Une relation est dite en troisième forme normale si elle est en deuxième forme normale, et si toutes les dépendances fonctionnelles issues de la clé primaire sont directes**



La dépendance entre "code matériel" et "libellé marque" n'est pas directe, "libellé marque" est en dépendance fonctionnelle directe avec le "code marque".  
 La relation doit être éclatée en deux, pour être exprimée en troisième forme normale :

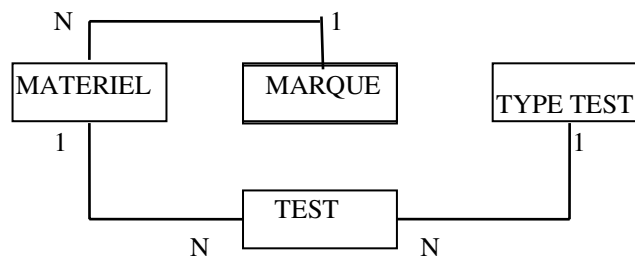
**R-MATERIEL (code matériel, libellé matériel, code marque)R-MARQUE (code marque, libellé marque)**

Le schéma conceptuel final de la base de données est donc :

**R-MATERIEL (code matériel, libellé matériel, code marque)R-MARQUE (code marque, libellé marque)**  
**R-TYPETEST (code type, libellé test)**  
**R-TEST (code matériel, code type, date du test, résultat du test)**

Commentaires:

- Le schéma conceptuel fait apparaître 3 relations entités : **R-MATERIEL, R-MARQUE, R-TYPETEST**
- et la relation association **R-TEST** qui réalise le lien **Matériel <--> Type test** de type N:M
- Le lien fonctionnel **Matériel <--> Marque** de type N:1 est réalisé par la présence du "codemarque" dans la relation R-MATERIEL.



## II Démarche de création d'une base de données

Avant la création de la base de données un travail d'analyse préalable est indispensable. Il est nécessaire d'analyser le problème à traiter en partant des résultats à obtenir (en sortie) avec leur fréquence. Etudions par exemple la création d'une base de données de gestion d'un Centre sportif.

### LE PROBLEME A RESOUDRE

Une salle de sport permet à ses adhérents d'utiliser des équipements sportifs, sur certains sites moyennant une cotisation dont le montant est fonction des équipements nécessités par l'activité choisie et le lieu de pratique.

Les activités possibles et les cotisations annuelles sont les suivants :

- |               |             |
|---------------|-------------|
| • Judo        | 2 035,00 DA |
| • Karate      | 4 200,00 DA |
| • GYMNASTIQUE | 1 815,00 DA |
| • MUSCULATION | 2 420,00 DA |
| • Aikido      | 1320,00 DA  |
| • TENNIS      | 3 500,00 DA |

Le montant du droit d'entrée par lieu de pratique est le suivant :

- |                |           |
|----------------|-----------|
| • Annaba ville | 150,00 DA |
| • Kouba        | 190,00 DA |
| • Sidi ammar   | 300,00 DA |
| • Saint Clout  | 90,00 DA  |
| • Belvedere    | 160,00 DA |

Les cotisations sont payées pour l'année au début du mois de janvier.

Les adhérents peuvent choisir plusieurs activités et utiliser les équipements correspondant aux activités choisies, quand bon leur semble, aux heures d'ouverture de chaque lieu. Chaque adhérent ne s'inscrit que dans un seul lieu de pratique.

Le responsable de la salle souhaite gérer sur Base de données les cotisations d'environ 500 adhérents. De plus, il voudrait gérer les renseignements concernant les adhérents, les activités, les lieux, les tarifs, et pouvoir éditer les états correspondants.

## LES RESULTATS A OBTENIR

Recenser tous les résultats que votre application doit pouvoir vous fournir. Il s'agit généralement d'états à produire. Ces états doivent contenir des données. Une maquette papier des états peut être réalisée afin de ne rien oublier.

Si nous reprenons notre exemple, les résultats à obtenir sont :

- la liste des adhérents avec leur code, nom, prénom, date de naissance, adresse, codepostal, ville et numéro de téléphone
- la liste des équipements mis à leur disposition avec le code, le nom et le tarif d'utilisation
- la liste des adhérents et des équipements qu'ils utilisent, ainsi que le montant payé.
- ...

## LE DICTIONNAIRE DES DONNEES

Il faut alors créer le dictionnaire des données c'est-à-dire recenser tous les renseignements à gérersans distinguer ce à quoi ils se rapportent.

Nous aurons donc :

- Nom adhérent
- Prénom adhérent
- Date de naissance
- Adresse
- Code postal
- Ville
- Numéro de téléphone
- Nom activité
- Tarif activité
- Lieu de pratique
- Droit d'entrée

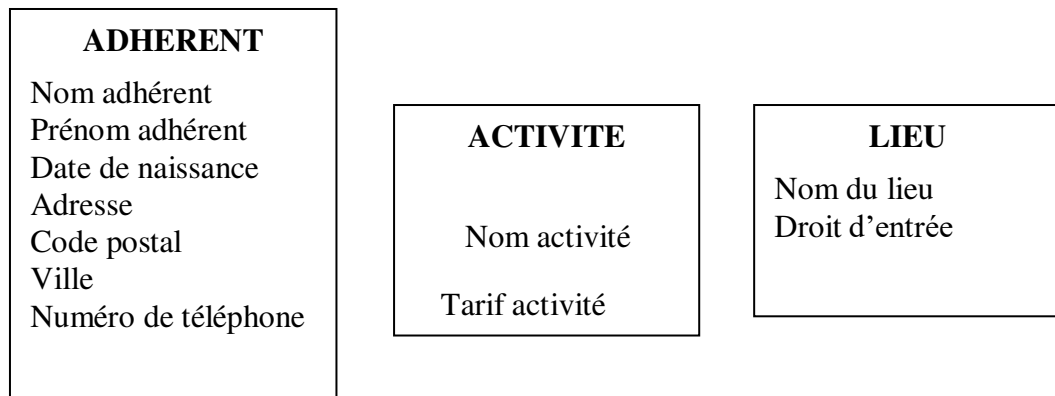
## LA DEFINITION DES ENTITES

L'entité peut être un individu (client, adhérent), un bien (article, dépôt, magasin, équipement...), un concept (description d'une commande, inscription...).

Nous voyons apparaître ici trois entités : les **adhérents** les **activités** et les **lieux de pratique**.

Il s'agit maintenant de définir à quelle entité se rapportent les données recensées plus haut, c'est-à-dire de quel **objet** ou **entité** elles deviennent l'**attribut** (ou la **caractéristique**).

Nous pouvons définir le schéma qui suit :



A chaque entité correspondra une table dans la base de données.

## LE MODELE ENTITE ASSOCIATION

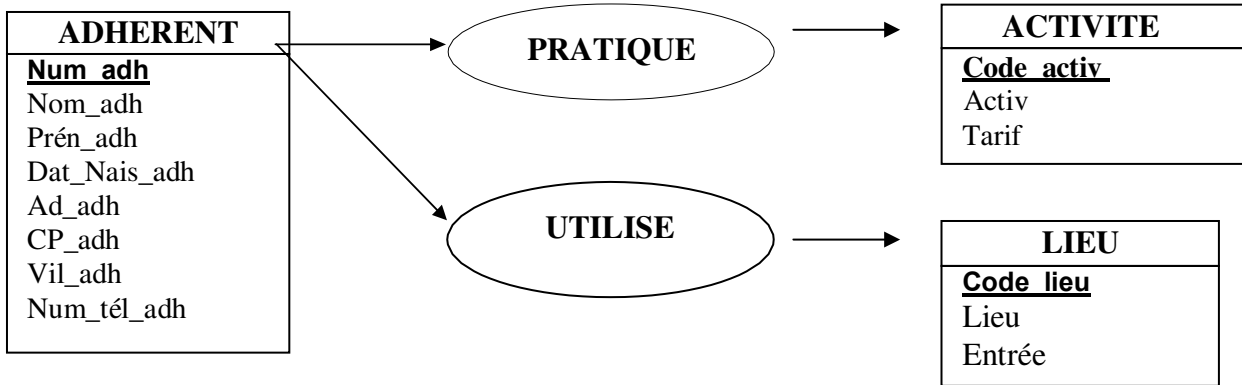
L'**association** est un lien entre 2 (ou plusieurs) entités.

Entre l'entité **ADHERENT** et l'entité **ACTIVITE**, l'association correspond à la notion de **PRATIQUE** de l'activité, et est matérialisée par le verbe **Pratiquer**.

Entre l'entité **ADHERENT** et l'entité **LIEU**, l'association correspond à la notion d'utilisation et est matérialisée par le verbe **Utiliser**

De plus nous allons rajouter un **identifiant unique** dans chaque table sous forme de **Numéro** ou de **Code**.

Le **modèle Entité Association** prend l'allure suivante :



**Num\_adh** est l'identifiant de la table **ADHERENT**. Ce champ sera défini comme clé primaire indexé sans doublon.

**Code\_activ** est l'identifiant de la table **ACTIVITE**. Il sera défini comme clé primaire indexé sans doublon.

**Code\_lieu** est l'identifiant de la table **LIEU**. Il sera défini comme clé primaire indexé sans doublon.

**LES RÈGLES DE GESTION**

Ce sont les règles qui régissent notre application. Ici un adhérent peut pratiquer **plusieurs activités**, sur **un seul lieu**. Il suffira qu'il paie le tarif des cotisations correspondant aux équipements utilisés, et le droit d'entrée sur le lieu de pratique.

**LE MODELE RELATIONNEL**

Nous devons maintenant créer le modèle relationnel. Les activités

- **Un adhérent peut pratiquer plusieurs ACTIVITES**
- **Une activité peut être pratiquée par plusieurs ADHERENTS**

Il y a donc une relation de **plusieurs à plusieurs** entre les tables **ADHERENT** et **ACTIVITE**.

Avec ACCESS, il n'est pas possible de créer un tel type de relation directement entre deux tables. Il faut nécessairement **transiter par une table intermédiaire**. Pour cela, il faut **remplacer l'association** matérialisée par le verbe utiliser par une nouvelle **table** qui servira de lien entre les 2 autres tables.

Cette nouvelle table que nous appellerons **PRATIQUE** comprendra donc les champs suivants :

*Num\_adh, Code\_activ*

**N.B** Ces deux champs correspondent aux clés primaires des deux autres tables.

Nous établirons une relation de type **un à plusieurs** entre le champ **Num\_adh** de la table **ADHERENT** et le champ **Num\_adh** de la table **PRATIQUE**.

Nous établirons une relation de type **un à plusieurs** entre le champ **Code\_activ** de la table **ACTIVITE** et le champ **Code\_activ** de la table **PRATIQUE**.

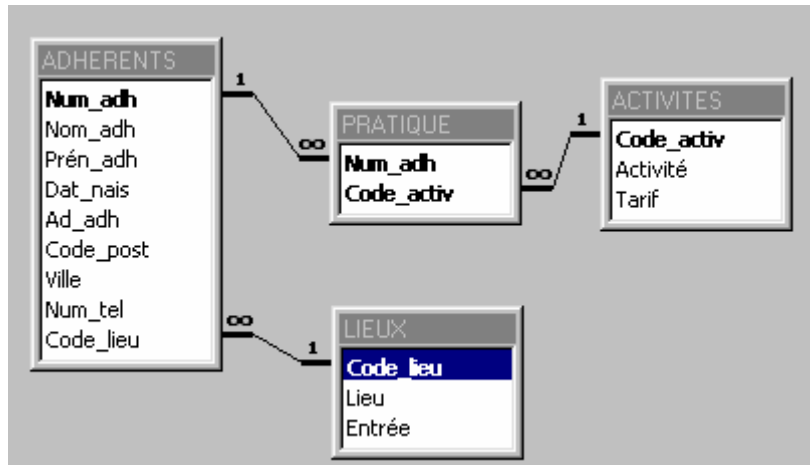
Le lieu de pratique

**Un adhérent ne pratique que sur un seul lieu. Un lieu peut recevoir plusieurs adhérents.**

**N.B** Nous avons donc une **relation de un à plusieurs** entre la table **LIEU** et la table **ADHERENT**.

Pour créer cette relation, nous allons devoir rajouter dans la table **ADHERENT** le code du lieu de pratique, afin d'établir la relation directe entre le champ **Code\_lieu** de la table **LIEU** et le champ **Code\_lieu** de la table **ADHERENT**.

Le modèle relationnel sera donc le suivant



### III les redondances

#### Exemple

Il est nécessaire d'éviter les redondances dans le modèle relationnel. Prenons par exemple la table suivante qui concerne les propriétaires de véhicules :

nom	date	tél	n° immat	marque	type	cv	coul
Anis	10/2/88	23.32.32.23	334419223	Renault	R25	9	bleu
Ines	8/10/88	62.62.52.55	778712025	Peugeot	405GR	7	vert
salim	7/7/89	76.45.34.34	55418524	Volvo	245	8	blanc
Zoubir	21/4/90	76.45.34.34	66719819	Peugeot	305	6	gris
Kamel	15/8/90	78.25.68.52	12917825	Renault	R25	9	blanc

Elle pose dans son utilisation un certain nombre de problèmes, liés à la redondance des données. Données redondantes

La table fait apparaître une personne et ses coordonnées autant de fois qu'elle possède un véhicule. Si Mr Salim change de N° de téléphone, il faut s'assurer que la mise à jour s'effectue bien sur les

Salim	7/7/89	76.45.34.34	5541852	Volvo	245	8	blanc
Zoubir	21/4/90	76.45.34.34	6671981	Peugeot	305	6	gris

deux enregistrements le concernant.

Une autre redondance est liée à la correspondance Marque, Type, CV

Anis	10/2/88	23.32.32.23	33441922	Renault	R25	9	bleu
Kamel	15/8/90	78.25.68.52	12917825	Renault	R25	9	blanc

Pour chaque propriétaire ayant une R25, il faudra saisir la marque et la puissance.

De plus, un même véhicule peut passer entre les mains de plusieurs propriétaires. Il faudra alors saisir toutes ces caractéristiques lorsqu'il changera de mains.

#### Solution

Les champs que nous trouvons dans cette table sont les attributs d'entités différentes. Nous allons rattacher ses attributs aux entités qu'ils caractérisent

**Nom et Numéro de téléphone** caractérisent l'entité **PROPRIETAIRE**

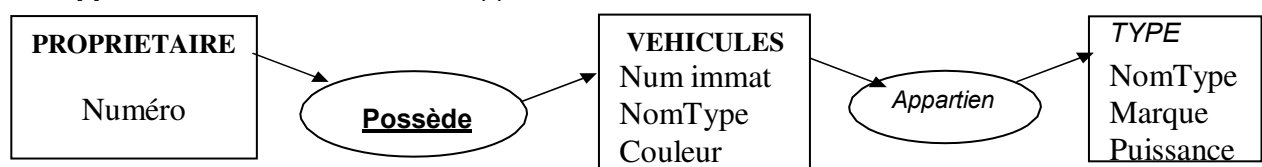
**Numéro d'immatriculation, Marque, Type et Couleur** caractérisent l'entité **VEHICULE**

**Puissance** caractérise l'entité **TYPE**.

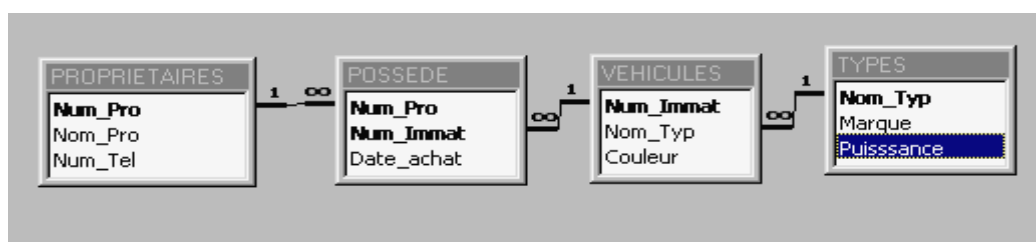
L'entité **PROPRIETAIRE** et l'entité **VEHICULE** sont liées par la notion de **Possession**. La relation est matérialisée par le verbe **Posséder** : En effet, un propriétaire possède un ou plusieurs véhicules. Mais, un même véhicule pourra avoir été possédé par plusieurs propriétaires successifs. Nous avons donc entre ces deux entités une relation de plusieurs à plusieurs.

L'attribut **Date d'achat** ne caractérise pas l'une des entités mises en évidence ci-dessus. Par contre elle caractérise le moment où le propriétaire va posséder le véhicule.

L'entité **VEHICULE** et l'entité **TYPE** seront liées par la notion d'appartenance. La relation est matérialisée par le verbe **Appartenir**. En effet un véhicule appartient à un **TYPE** et un seul.



Le modèle relationnel aura cette allure :



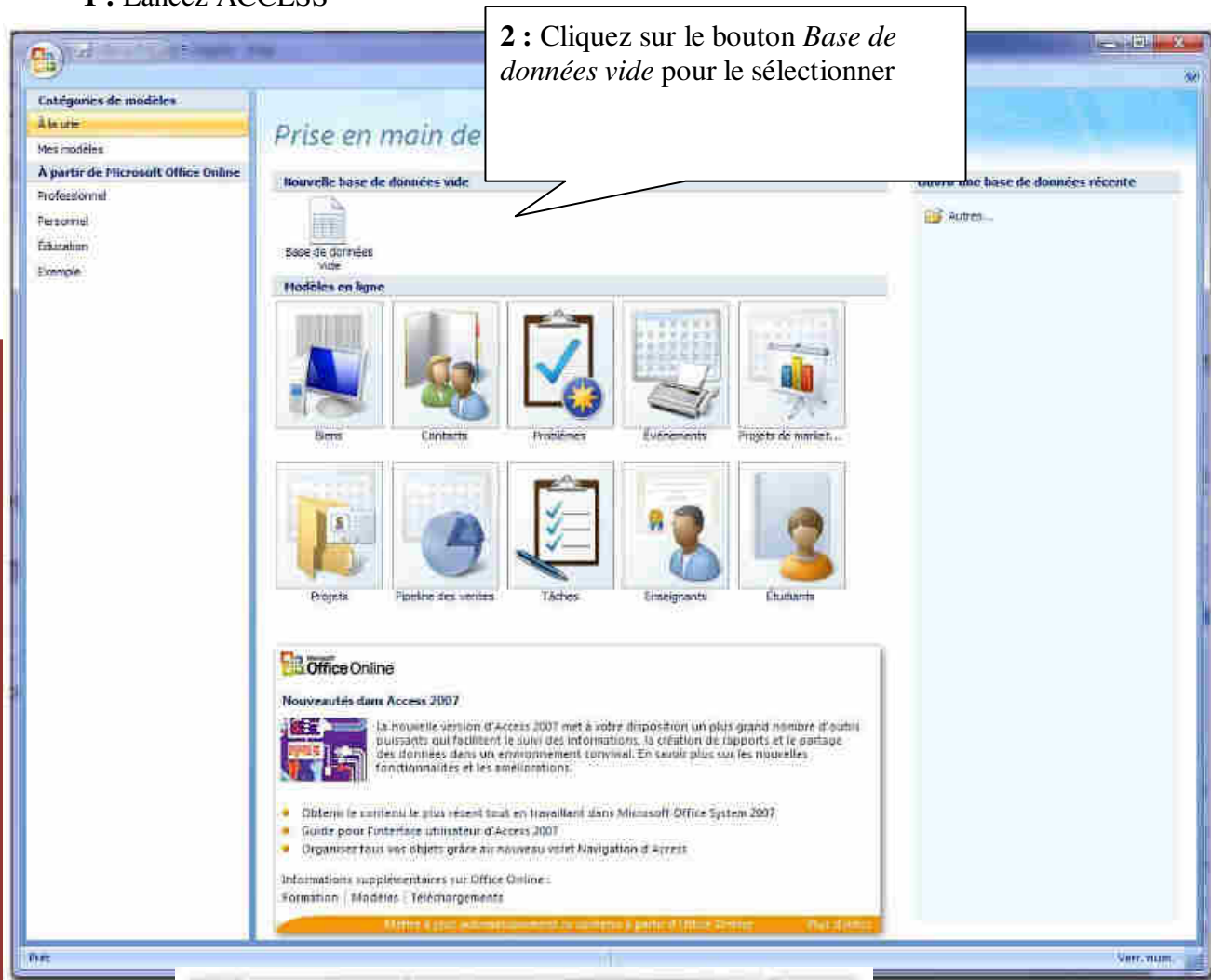
La relation entre la table **PROPRIETAIRES** et la table **VEHICULES** est matérialisée par la table **POSSEDE**. Celle-ci a comme attributs les deux clés primaires des deux tables et le champ **Date d'achat**, point de départ de la possession du véhicule.

La relation entre la table **VEHICULES** et la table **TYPES** ne nécessite pas la création d'une table intermédiaire puisqu'il s'agit d'une relation de **un à plusieurs** de la table **TYPES** vers la table **VEHICULES**.

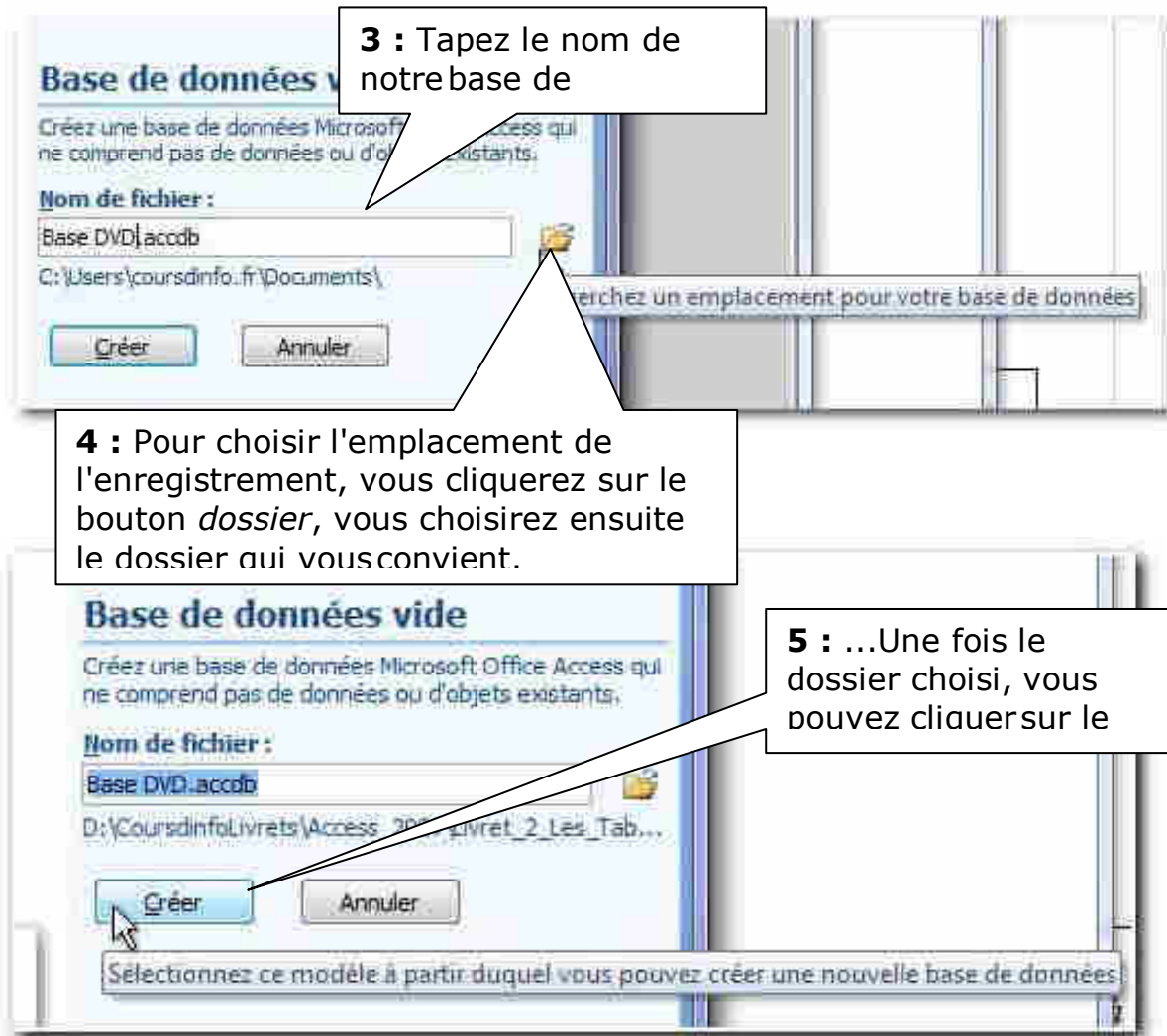
*PARTIE 3 CREATION ET MANIPULATION DES BASES DE DONNEE AVEC MS  
ACCESS*

## Démarche :

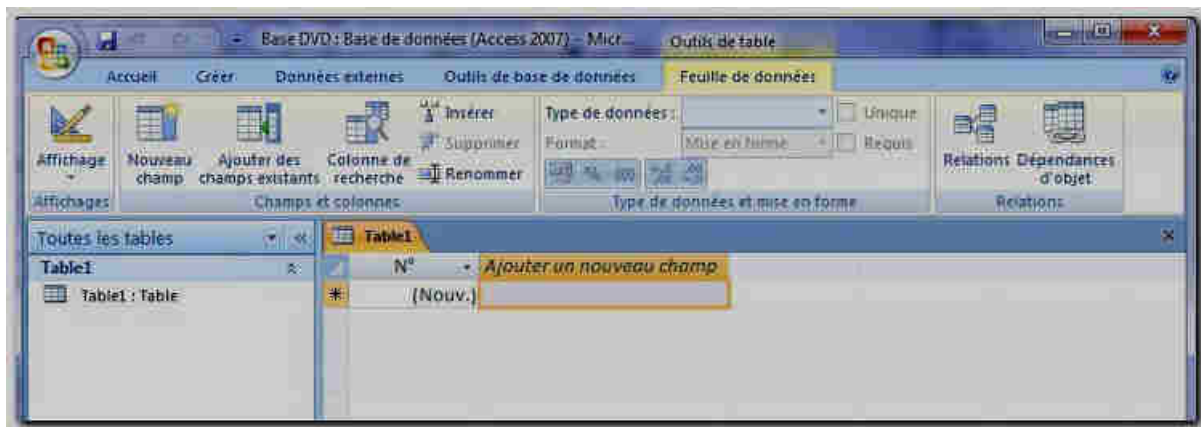
### 1 : Lancez ACCESS





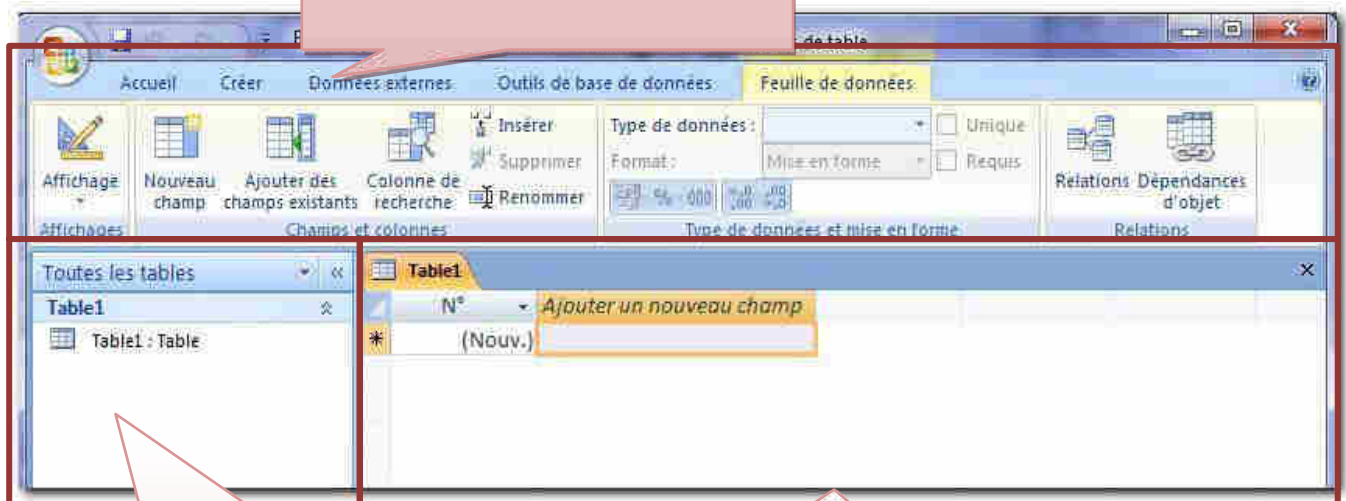


*Résultat la base de données est créée. Nous pouvons commencer à créer son contenu*



## Découvrir l'interface graphique

**Vous trouverez ici les rubans qui contiennent les commandes dont vous aurez besoin. Nous les examinerons au fur et à mesure.**



**Le volet de gauche est le volet de navigation. Pour le moment il ne contient que les tables, mais à mesure l'avancement de notre projet on y verra les requêtes, les formulaires...**

# CREATION ET MANIPULATION DES BASES DE DONNEE AVEC MS ACCESS

## I Création des tables

La **table** est Le principal organe de stockage de données d'ACCESS.

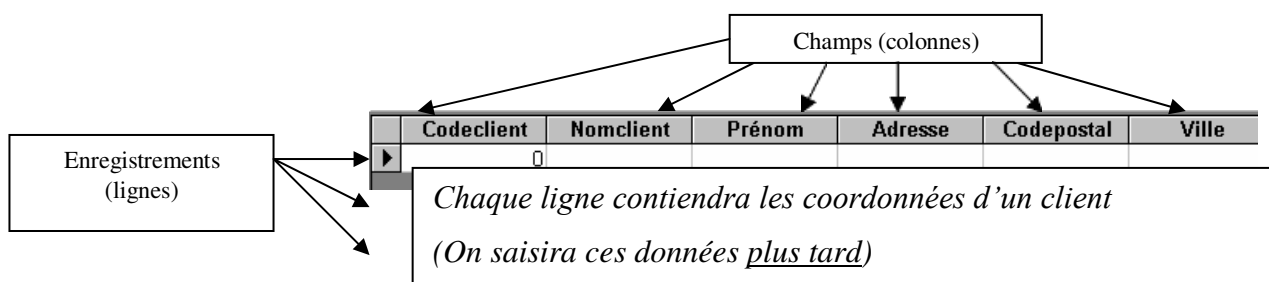
Pour qu'une base de données Access existe, il faut au moins une table. Il peut bien évidemment y en avoir plusieurs.

Les tables servent à emmagasiner les données stables (*quand on dit données stables, cela veut dire que leur **structure** est stable ; par exemple une table **clients** contiendra toujours des noms, des adresses, etc. et ces éléments se retrouveront à un emplacement déterminé*).

La table ressemble physiquement à une feuille de calcul Excel : il y a des **colonnes**, qui prennent ici le nom de **champs** et des **lignes** qu'on appelle **enregistrements**.

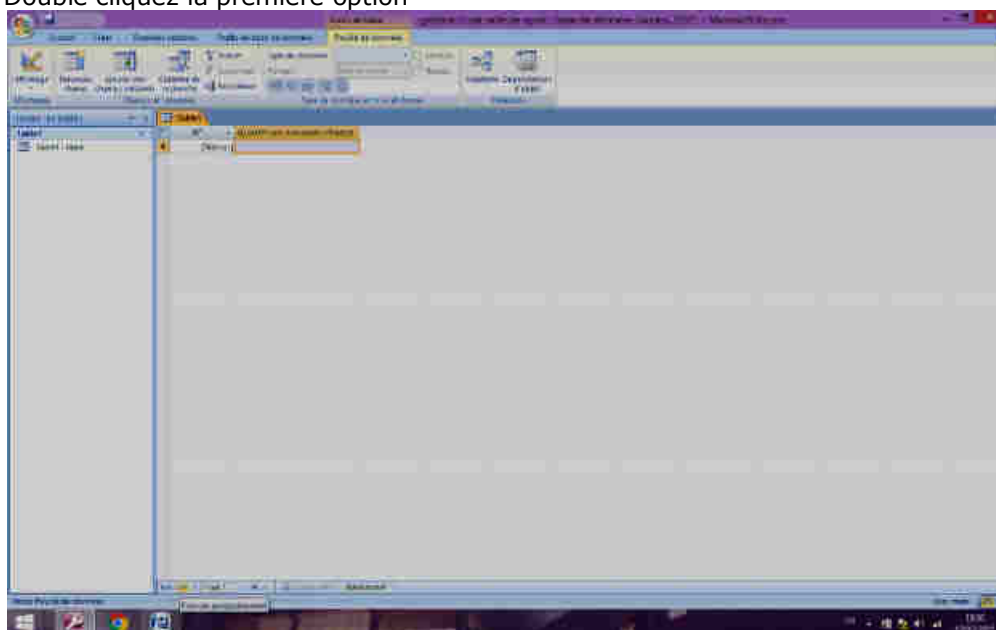
Créer une table, c'est d'abord décider de sa structure c'est-à-dire quels champs il faut créer et quel sera leur type de contenu (alphabétique, numérique, etc.)

### **Exemple de fichier clients (exemple de gestion d'un club de loisir)**

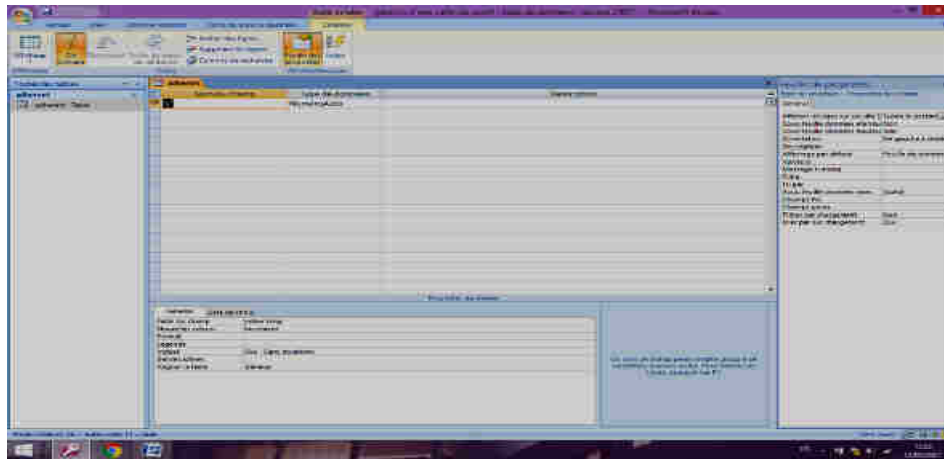


### **Création de la table ADHÉRENTS**

- Double cliquez la première option

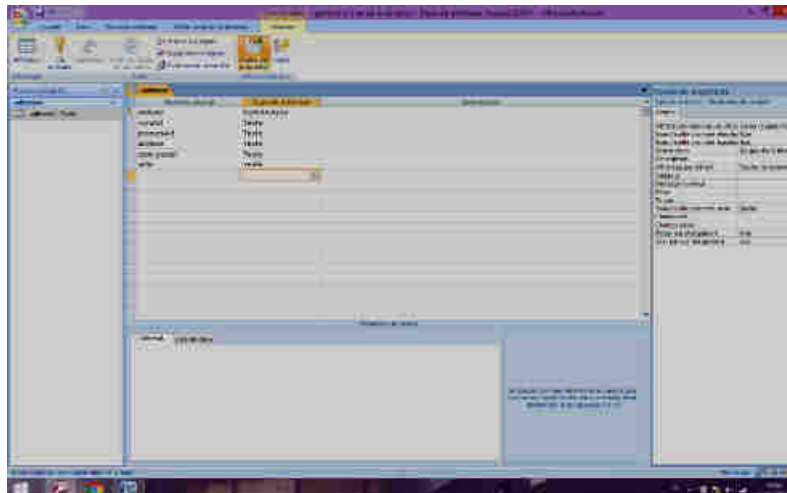


Vous obtenez le panneau de création de structure de table



Le travail va consister à décider les noms des champs puis à choisir le type de données qui figureront dedans. (Par exemple, le nom de l'adhérent contiendra du texte exclusivement ; ce sera donc un champ de type Texte)

Entrez les noms des champs tels qu'ils apparaissent ci-contre ; pour le moment ne touchez pas au type de données. Vous obtenez

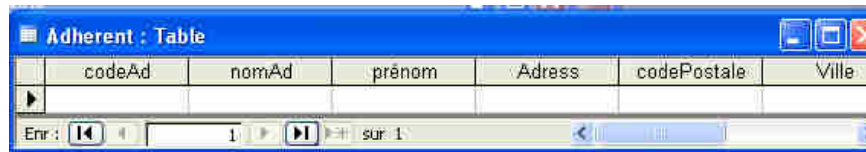


Jusque-là, vous travaillez sur la structure c'est-à-dire *l'envers du décor*. Pour visualiser ce que vous venez de créer, passez en mode feuille de données

Habituez-vous à passer du mode création  au mode feuille de données 



- Répondez **oui** à la question : *la table doit d'abord être enregistrée*, donnez-lui le nom **ADHÉRENTS** et répondez **non** à la question *laisser ACCESS définir une clé primaire*.
- Résultat (*rappel : vous vous trouvez en mode feuille de données*)



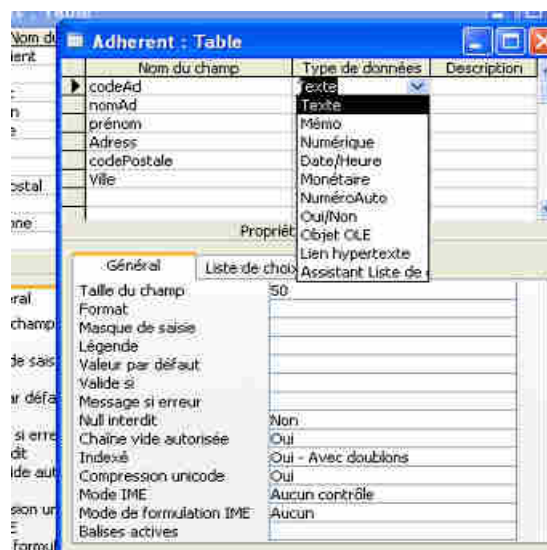
- Pour continuer à travailler sur la structure de la table, il faut repasser en mode création.



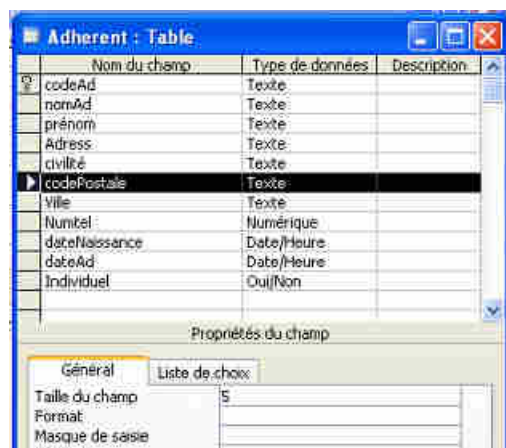
Cliquez sur l'icône

### Définir les caractéristiques des champs

- Vous allez maintenant définir le type de vos différents champs.



- Cliquez après le mot Texte sur la même ligne que CodeAd et choisissez Numérique. Cela veut dire que le code de vos adhérents sera constitué de chiffres. Remarque : un champ Texte peut contenir des lettres ou des chiffres, alors qu'un champ Numérique ne peut contenir que des chiffres.
- Laissez les autres champs en texte.
- Pour le CodePostal ; placez-vous sur sa ligne et limitez-le à 5 caractères en corrigeant les 50 en 5 dans la zone Taille de champ de l'onglet Général en bas



- Passez en mode Feuille de données pour voir, puis repassez en création.
- Vous allez maintenant ajouter d'autres champs à la table ADHÉRENTS.
- Sur la ligne après Ville, ajoutez Numtel. Vous allez le laisser en texte (nous avons vu qu'un champ texte peut contenir des chiffres), mais pour que la saisie soit plus aisée par la suite, vous allez définir un masque de saisie.

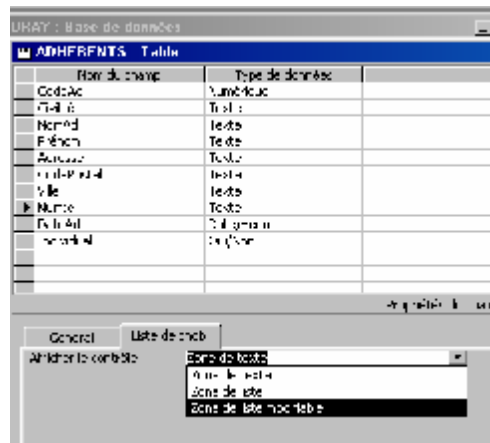
Cela veut dire que la zone à remplir pour le numéro de téléphone se présentera ainsi : \_\_ \_\_ \_\_ \_\_ et que vous n'aurez pas à saisir les espaces. Il faut savoir que tout champ contenant un masque de saisie, même s'il doit recevoir des chiffres, doit être obligatoirement un champ Texte.

- Le curseur étant sur la ligne de Numtel, cliquez dans [Masque de saisie] puis sur les pointillés qui vont lancer l'assistant. Laissez-vous guider par les écrans ; vous obtenez en fin de compte les signes suivants sur la ligne [Masque de saisie] : 00\ 00\ 00\ 00\ 00; (remarque : en tapant ces signes à la main, on obtiendrait le même résultat).
- Ajoutez à la table **ADHÉRENTS** les DateAd , champ de type Individuel pour saisir la date d'adhésion du membre pour savoir s'il appartient ou non à un comité d'entreprise Oui/Non

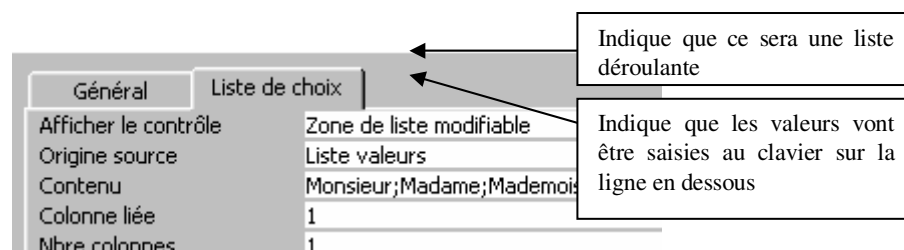
passer en mode feuille de données. pour voir le mode, cliquer sur mode création afin de voir si le masque est actif. Repassez en mode création.

Il manque un champ **Civilité** ; vous allez insérer une nouvelle ligne. Sélectionnez la ligne **NomAd** et appuyez sur la touche créée. [Insr] saisissez **Civilité** comme nom de champ. du clavier. Une nouvelle ligne

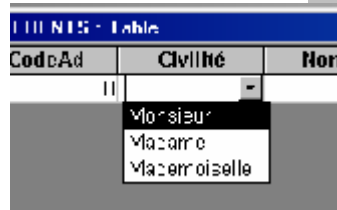
- Ce serait intéressant de créer une liste déroulante pour les civilités, c'est à dire une liste dans laquelle il suffirait de choisir au lieu d'avoir à saisir.



- Le curseur étant sur la ligne de Civilité, cliquez sur l'onglet Liste de choix puis sur la ligne Zone de texte et choisissez Zone de liste modifiable.



- Sur la ligne en-dessous, choisissez Liste de valeurs
- Sur la ligne en-dessous, saisissez : Monsieur;Madame;Mademoiselle (n'oubliez pas les points virgules et ne faites pas d'espace)
- Passez en mode visualisation, placez-vous dans la case sous **Civilité** : vous obtenez



### **Notion de clé primaire**

Dans un système de gestion de bases de données comme Access, pour toute table que l'on crée il faut se poser une question : **Quel est le champ qui contiendra pour chaque enregistrement**

## une caractéristique unique ?

Il est indispensable que toute table comporte un tel champ, de façon que le système puisse identifier de manière certaine chaque enregistrement.

Dans la table **ADHÉRENTS**, quel champ va servir d'identifiant ?

C'est bien sûr le Code Adhérent : chaque adhérent aura le sien.

Pour indiquer au système que le champ **CodeAd** est l'identifiant unique, vous allez poser dessus une clé (on dit une clé primaire). En mode **création**, sélectionnez la ligne de clé s'insère en début de ligne. **CodeAd**

cliquez sur l'icône de **clé**. Un symbole de **Pour Access, cela signifie qu'il est impossible de donner deux fois le même code dans la table.**

**Remarque** : la clé est généralement posée sur un **champ numérique** mais techniquement rien n'empêche de la poser sur un champ texte, dans la mesure où on est sûr que le contenu sera unique dans la table. Il est possible aussi de poser une clé sur deux champs en même temps : à ce moment-là ce sera l'association des deux contenus qui devra être unique.

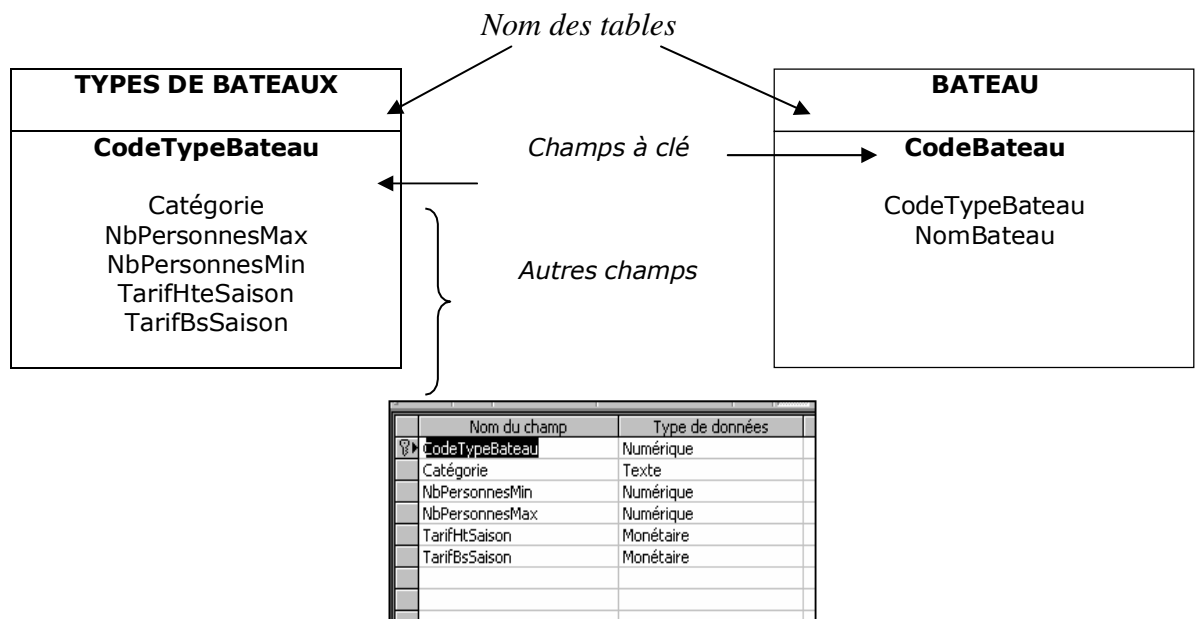
- Refermez la table **ADHÉRENTS** en enregistrant les modifications.

## Création des tables pour la location des bateaux

On peut louer trois types de bateaux, mais il y a plusieurs bateaux de chaque type ; vous aurez donc besoin de deux tables : la table **TYPES DE BATEAUX**, qui contiendra pour chaque enregistrement les caractéristiques du type de bateau. La table **BATEAUX** contiendra le nom propre de chaque bateau et sera rattachée à la table **TYPES DE BATEAUX** (de cette manière, vous n'aurez à saisir les caractéristiques qu'une fois par type de bateau). Dans la partie **Tables** de la fenêtre **Base de données**, cliquez **Créer une table en mode Création**.

Réfléchissez maintenant aux champs nécessaires dans chacune de ces deux tables. Il s'agit de chercher dans le descriptif de la flotte ce qui se rapporte au Type de bateau et ce qui se rapporte à un bateau en particulier.

On peut ainsi schématiser la structure de nos deux tables :



- Depuis la fenêtre **Base de données**, dans, faites **Créer une table en mode création**.
- Constituez la table **TYPES DE BATEAUX** comme sur le modèle ci-contre. N'oubliez pas de poser la clé sur le premier champ. Vérifiez que vous ne vous êtes pas trompé(e) de type de données pour chaque champ. Pour les tarifs, mettez en plus de **Monétaire** un format **DA** dans l'onglet du bas.

Nom du champ	Type de données
CodeBateau	Numérique
NomBateau	Texte
CodeTypeBateau	Numérique

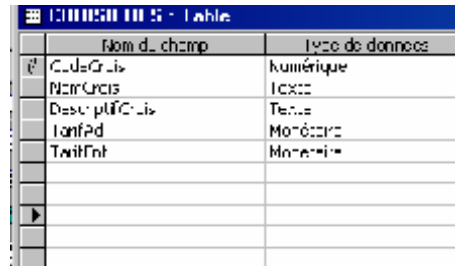
- Refermez et enregistrez la table sous le nom **TYPES DE BATEAUX** (les données seront saisies plus tard, de même que pour la table **ADHÉRENTS**)

- Créez la table **BATEAUX** comme ci-contre en posant la clé sur ce champ. Refermez et enregistrez la table sous le nom **BATEAUX**.

On pourrait aller chercher le `CodeTypeBateau` dans la table **TYPE DE BATEAUX**.

### Création de la table CROISIÈRES

Vous savez maintenant créer une table.



Nom du champ	Type de données
CodeCrois	Numérique
NomCrois	Texte
DescriptionCrois	Texte
TarifPd	Monétaire
TarifFnt	Monétaire

- Créez la table **CROISIÈRES** comme ci-contre. Pour les champs monétaires, demandez un format dinars.
- Refermez la table.

La base comporte maintenant 4 tables : **ADHÉRENTS**, **BATEAUX**, **TYPES DE BATEAUX** et **CROISIÈRES**.

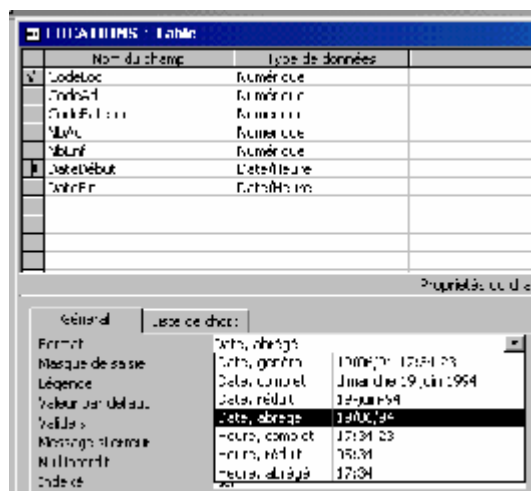
Ainsi juxtaposées, les 4 tables ne communiquent pas entre elles. Les liaisons que vous allez établir dans le chapitre suivant vont les rendre communicantes.

Quelles relations peut-on établir entre ces tables ? Pour trouver la réponse à cette question, on utilise des phrases avec sujet, verbe, complément :

un adhérent réserve une croisière,  
un adhérent loue un bateau.

Le fait de réserver ou louer nous amène à créer deux autres tables : une table des réservations et une table des locations dans laquelle seront stockés les éléments propres aux réservations ou aux locations (exemple : la date ; celle-ci est bien une caractéristique de la réservation et non du bateau).

### Création de la table LOCATIONS



Nom du champ	Type de données
CodeLoc	Numérique
TarifPd	Monétaire
TarifFnt	Monétaire
DateDébut	Date/heure
DateFin	Date/heure

Propriétés de l'objet

Général

Format: date, abrégé

Mask de saisie: [none]

Légende: [none]

Valeur par défaut: [none]

Valide: [none]

Messages d'erreur: [none]

Nullif: [none]

Indexé: [none]

- Créez la table **LOCATIONS** comme ci-contre
- Placez la clé sur le champ `CodeLoc`
- Pour `DateDébut` et `DateFin`, choisissez un format `date,abrégé` (en vous plaçant sur la ligne concernée, cliquer dans `Format`, onglet `Général`).
- Refermez la table **LOCATIONS**.

### Création de la table RÉSERVATIONS



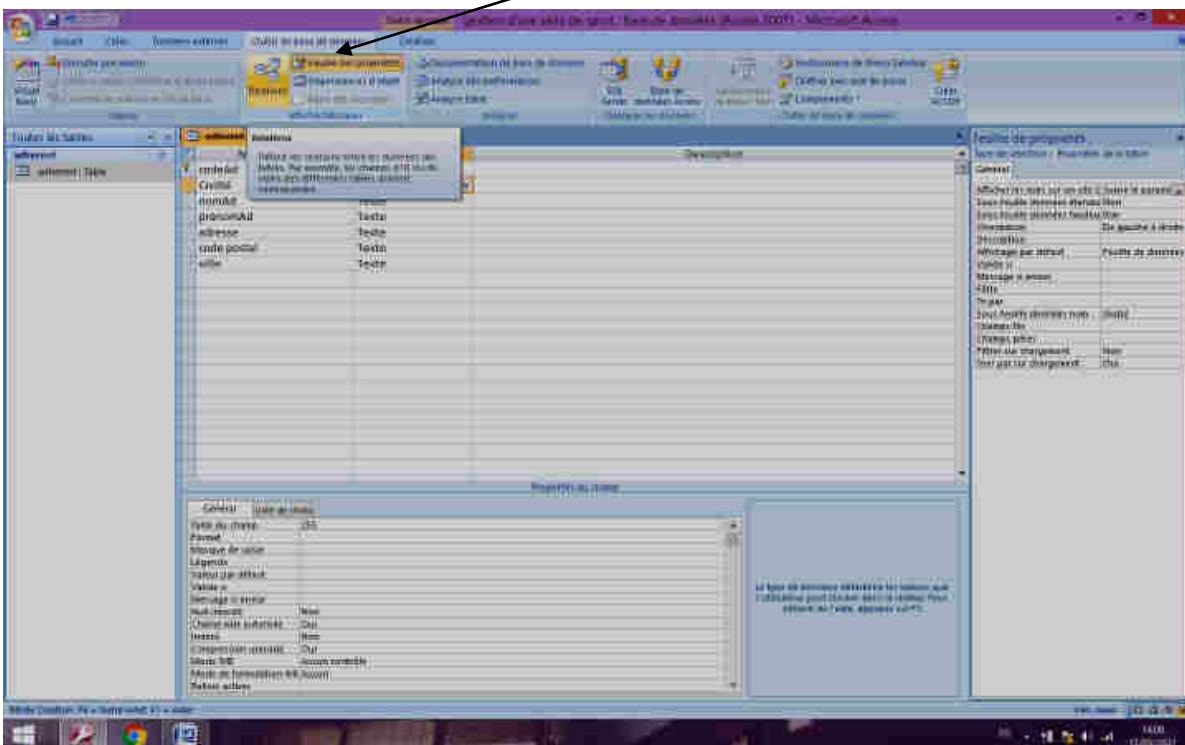
nom du champ	Type de données
CodeResa	Numérique
CodePd	Numérique
CodeCruis	Numérique
NbPd	Numérique
NbTmf	Numérique
Date	Date/Heure

- Créez la table **RÉSERVATIONS** comme ci-contre.
- Placez la clé sur le champ **CodeResa**
- Pour la date, prenez un format **Date**, abrégé

### Définition des relations

Cette partie du travail est particulièrement importante, car si vous rencontrez des problèmes ils se répercuteront tout au long du travail. Il faut donc impérativement que les relations fonctionnent parfaitement.

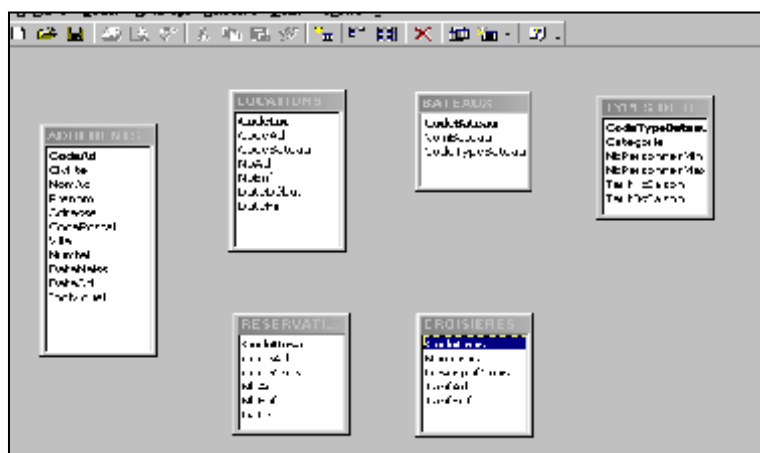
- Depuis la fenêtre de la Base de données, cliquez sur l'icône **Relations**



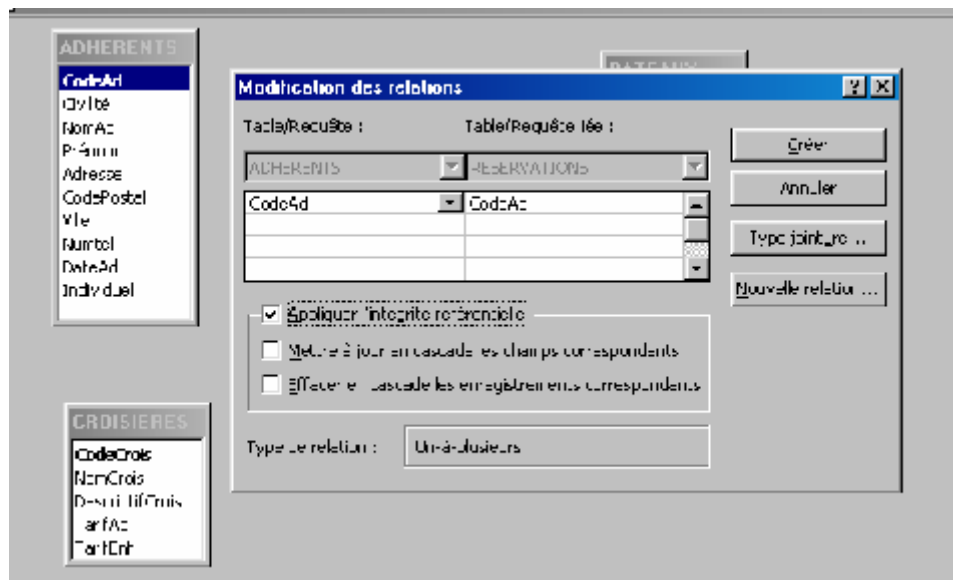
- Vous vous trouvez sur **ADHÉRENTS** ; cliquez **Ajouter**. Déplacez-vous sur **BATEAUX**, cliquez **Ajouter**, et ainsi de suite sur **CROISIERES** puis **LOCATIONS** puis **RÉSERVATIONS** puis **TYPES DE BATEAUX**.
- Les 6 tables sont affichées. Cliquez **Fermer**. (Si par mégarde vous avez ajouté deux fois une table, ôtez-la en appuyant sur la touche **Suppr** du clavier après l'avoir sélectionnée avec la souris)

Les tables peuvent être redimensionnées et déplacées comme n'importe quelle fenêtre Windows.

- Aménagez-les pour obtenir ceci :



Remarquez que dans chaque table, le champ à clé se présente en caractères gras. **Il reste à tracer les relations.**

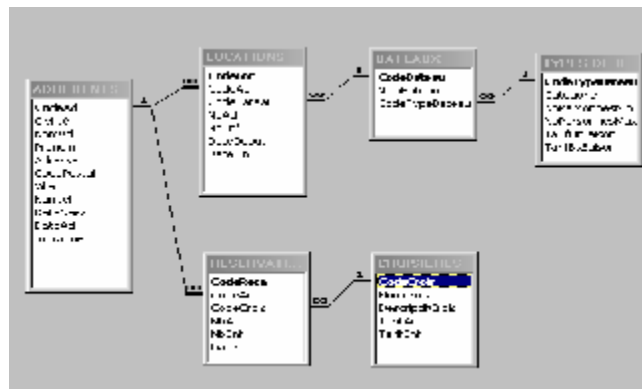


- Placez-vous sur `CodeAd` de la table **ADHERENTS** et sans lâcher le clic, tirez-le pour le déposer sur le champ `CodeAd` de la table **RÉSERVATIONS**.
- Dans la fenêtre qui apparaît, cochez `Appliquer l'intégrité référentielle` puis cliquez sur `Créer`. Le résultat est une ligne qui va de `CodeAd` de la table **ADHERENTS** jusqu'à `CodeAd` de la table **RÉSERVATIONS**.

#### Signification de cette ligne

Le système retrouve l'adhérent qui a réservé grâce à son code, indiqué dans la table des **RÉSERVATIONS**

- Le chiffre 1 et le symbole  $\infty$  (infini) signifient que pour un adhérent, plusieurs réservations sont possibles. En revanche, une réservation ne peut être attribuée qu'à un et un seul adhérent.
- Le sens de la relation, de 1 à  $\infty$  indique aussi que lorsque vous saisissez les données, il faudra d'abord saisir celles de la table du côté du 1 (*en clair : il ne sera pas possible d'enregistrer une réservation pour un adhérent qui n'existe pas encore*).
  - Définissez les autres relations pour obtenir ceci (*chaque fois il suffit de prendre le champ de démarrage et de le déposer sur le champ d'arrivée, attention de ne pas vous tromper*).
  - Vérifiez bien que votre écran est conforme à cette image



A partir de maintenant, le modèle est prêt. On sait quelles sont les tables, quels sont les champs qu'elles contiennent avec quel type de données à l'intérieur, on sait où se trouvent les clés, et on sait quelles relations unissent les tables.

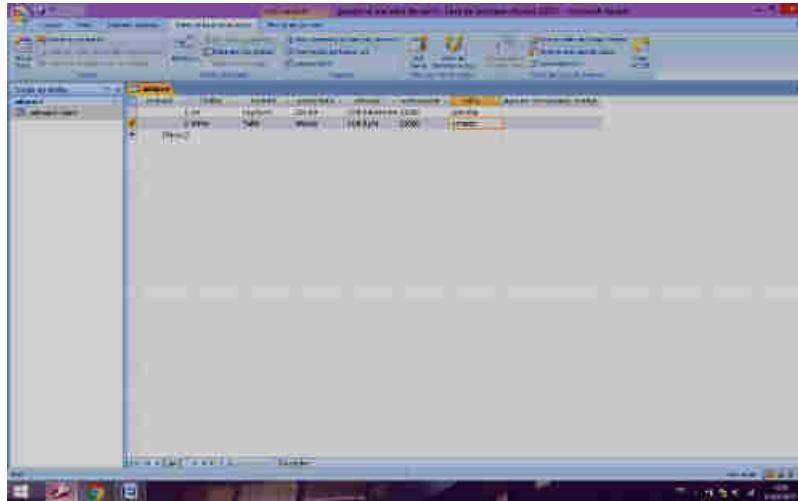
- Enregistrez les modifications apportées à la fenêtre des relations. L'essentiel de la base est constitué, la saisie des données va pouvoir se faire.

## II Saisie de donnée et création de formulaire

### Saisie de données en mode table

**Rappel** : d'après le modèle que vous avez déterminé, la saisie des données ne peut pas se faire dans n'importe quel ordre ; la table des réservations devra être saisie en dernier, puisque les chiffres 1 sont tous du côté des autres tables .

Depuis la fenêtre Base de données, dans les tables, double-cliquez sur **ADHÉRENTS** pour ouvrir la table en mode Feuille de données. Vous obtenez :

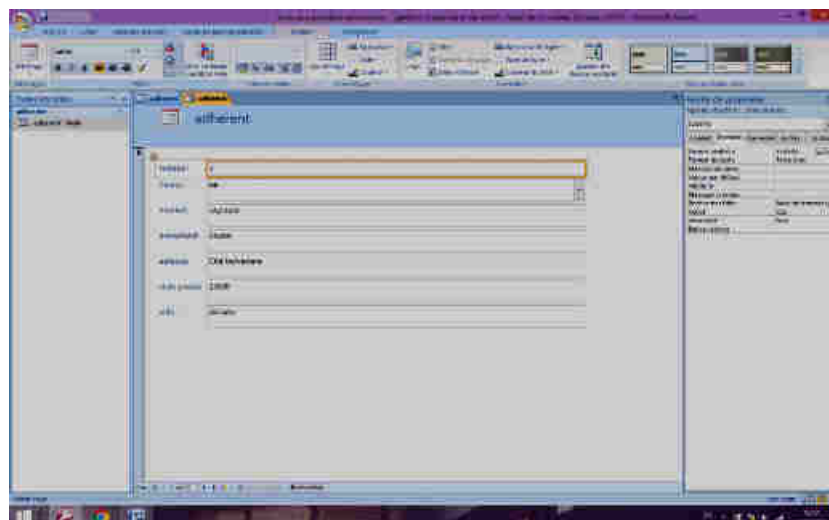


- Vous allez saisir le premier adhérent de la liste. Cliquez dans le champ CodeAd, 1<sup>ère</sup> ligne, et saisissez 1 puis tabulateur pour passer au champ Civilité.
- Cliquez sur la petite flèche de la liste déroulante et choisissez Madame.
- Continuez à saisir les données du premier adhérent. Constatez que le masque de saisie pour le téléphone a bien fonctionné. Un petit problème se pose cependant : la date de naissance n'a pas été prévue. Vous allez remédier à cela.
- La structure de la table peut encore être modifiée si cela ne touche pas au champ qui comporte la clé. De plus il ne s'agit que d'ajouter un champ. Cliquez sur l'équerre pour passer en mode création.

### Saisie de données en mode formulaire

Grâce aux assistants d'Access, le travail va être facile.

Depuis la fenêtre Base de données, placez-vous sur la table **TYPES DE BATEAUX** et développez le menu déroulant créer Formulaire



Vous avez devant les yeux un formulaire tout prêt. Il n'y a plus qu'à entrer les données dedans et améliorer sa présentation, mais ceci est un détail que nous verrons par la suite.

### Création du formulaire CROISIÈRES

- De la même manière que précédemment, créez à partir de la table **CROISIÈRES** un formulaire instantané et saisissez les données des croisières.
- Pour les descriptifs, utilisez une forme abrégée.
- Quand il n'y a qu'un tarif, mettez-le même pour adultes et enfants.
- Pour la Formule, mettez 0 dans tarif adulte.
- Refermez le formulaire en acceptant le nom **CROISIÈRES** proposé par le système.

Vous allez vérifier que les données saisies dans le formulaire sont bien présentes dans les tables.

- Dans la fenêtre Base de données, cliquez l'élément Tables et double-cliquez **CROISIÈRES** ; constatez que les données sont là. Refermez la table. Faites de même pour la table **TYPES DE BATEAUX**, puis pour la table **BATEAUX**.

### Création du formulaire LOCATIONS

- Créez un formulaire instantané à partir de la table **LOCATIONS**. Ne saisissez rien pour le moment.

### Gestion des événements .

#### Ajouter des enregistrements dans une table

Aujourd'hui, deux nouveaux clients s'inscrivent; avant de les enregistrer, créez un formulaire instantané à partir de la table **ADHÉRENTS**. Saisissez ensuite les données de :

<b>Layouni Zoubir</b> Cité Belvédère Tour E N0 62 0552933056	<b>Taleb Lahcen</b> Cité Rym Ts 3 N° 27 23000 Annaba 02 99 41 52 63
---	--

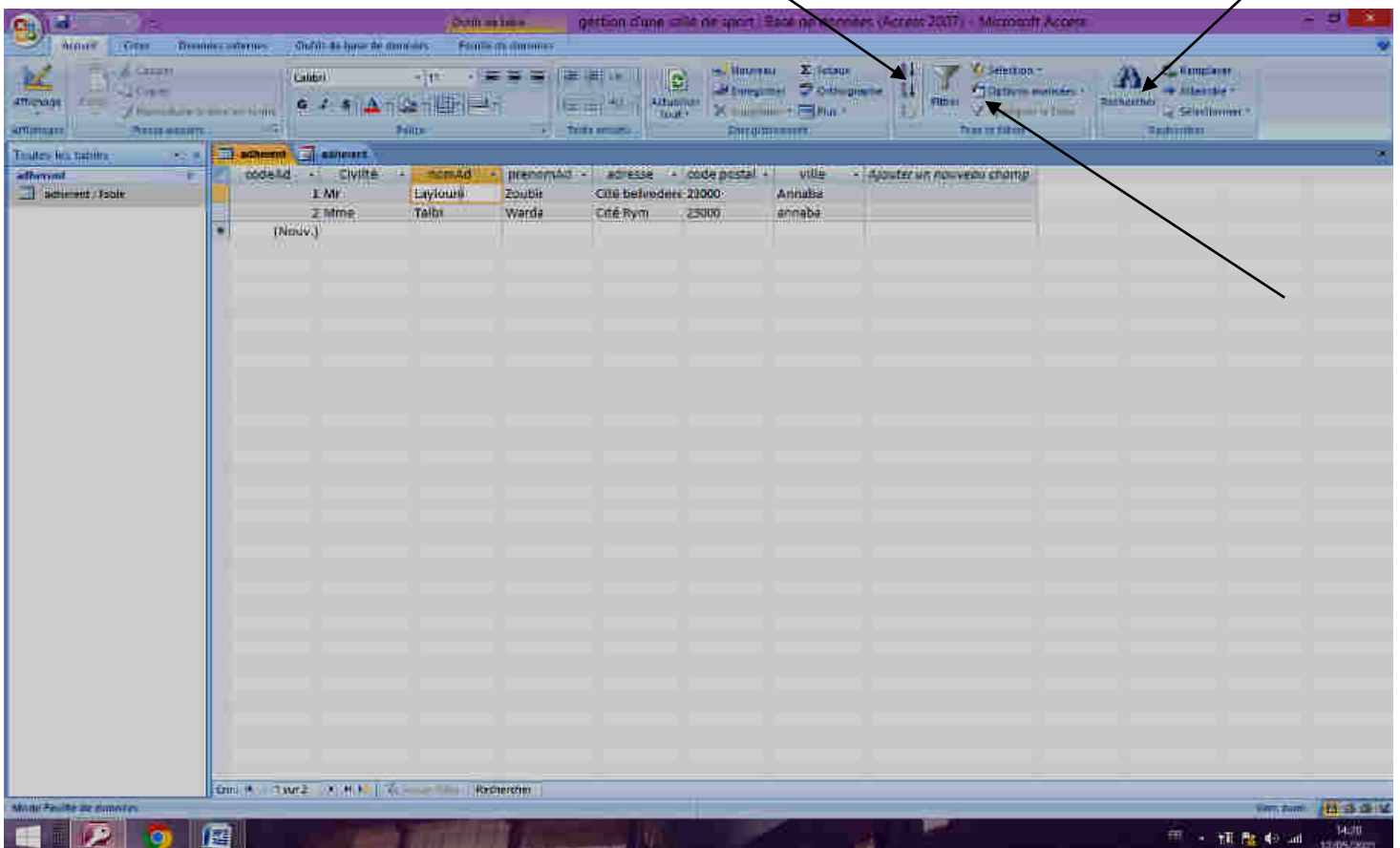
#### Modifier des données

Monsieur Layouni fait part de son nouveau numéro de téléphone : 06 14 30 45 78. Faites la modification (*utiliser le filtre par formule*).

#### Trier, filtrer, rechercher dans une table selon des critères

##### Tri

- Ouvrez la table **ADHÉRENTS**. Vous voulez obtenir un tri alphabétique par noms d'adhérents.



Sélectionnez le champ NomAd ; cliquez sur l'icône A/Z. Immédiatement, la table est triée.

- Constatez que les données de la ligne entière ont suivi le nom de l'adhérent.  
Remarque : sous Excel, il peut arriver que les données soient déstructurées, si vous avez fait une sélection malencontreuse ; avec Access, cela ne peut pas arriver.

### **Recherche**

- Vous recherchez l'adhérent **Layouni** Cliquez sur l'icône qui représente des jumelles. Remplissez la fenêtre dialogue comme suit.

Vous voulez ressortir uniquement les adhérents qui habitent Annaba

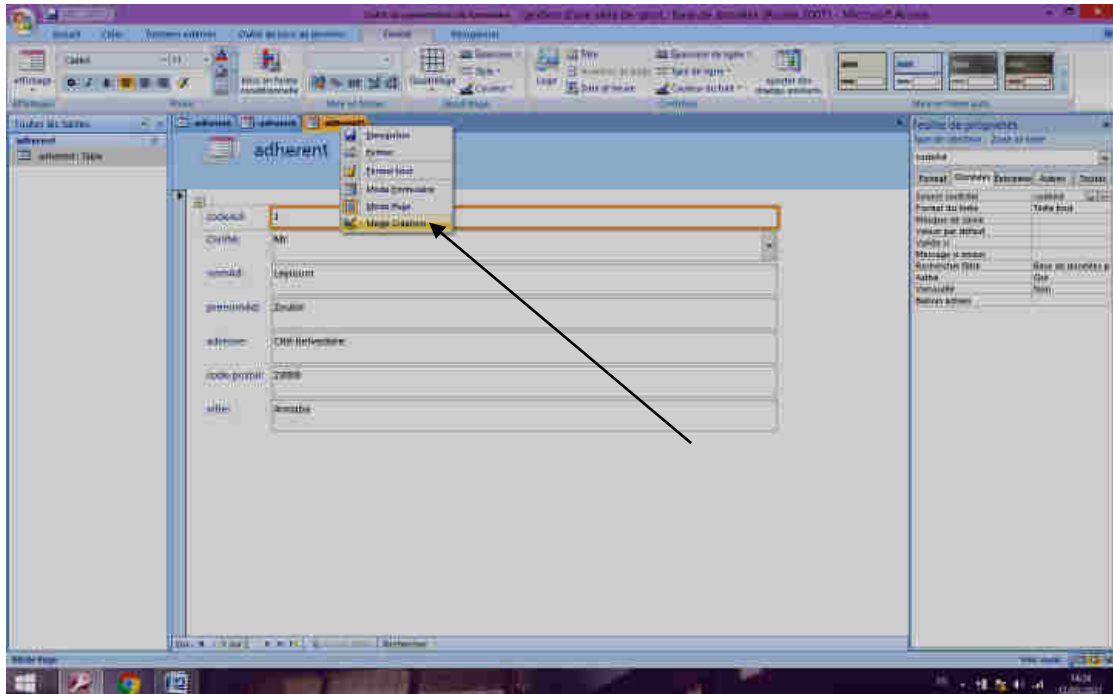
- Cliquez l'icône **Filtrer par formulaire** Placez-vous dans le champ Ville et choisissez **Annaba** dans la petite liste. Ensuite, cliquez sur **Appliquer le filtre**
- Vous avez devant les yeux la liste des adhérents de Annaba. Pour annuler le filtre, désactivez l'icône (qui s'appelle maintenant **Supprimer le filtre**).
- Supprimer le critère **Annaba**
- Filtrez les adhérents nés après le 1<sup>er</sup> janvier 1965 (*saisissez > 01/01/65 dans le champ **DateNaiss***).

Faites d'autres essais à votre guise, n'oubliez pas d'ôter les critères entre deux essais, sinon le système va chercher des enregistrements qui correspondent à la fois à plusieurs critères et n'aura peut-être pas de réponse).

### **Améliorer la présentation d'un formulaire**

Jusqu'ici nous ne nous sommes pas intéressés à la présentation car d'autres choses étaient plus importantes... toutefois il est toujours plus agréable de travailler sur un écran convivial et c'est pourquoi vous allez consacrer un peu de temps à améliorer votre cadre de saisie. Ceci est dans votre intérêt personnel, mais il faut penser que, dans l'entreprise, ce sont peut-être d'autres personnes qui devront saisir des données et qu'il est bon de leur faciliter aussi le travail.

- Depuis la fenêtre Base de données, dans l'élément **Tables**, placez-vous sur **ADHÉRENTS** et créez un formulaire instantané comme d'habitude.
- Vous obtenez :
- Passez en mode **Création**

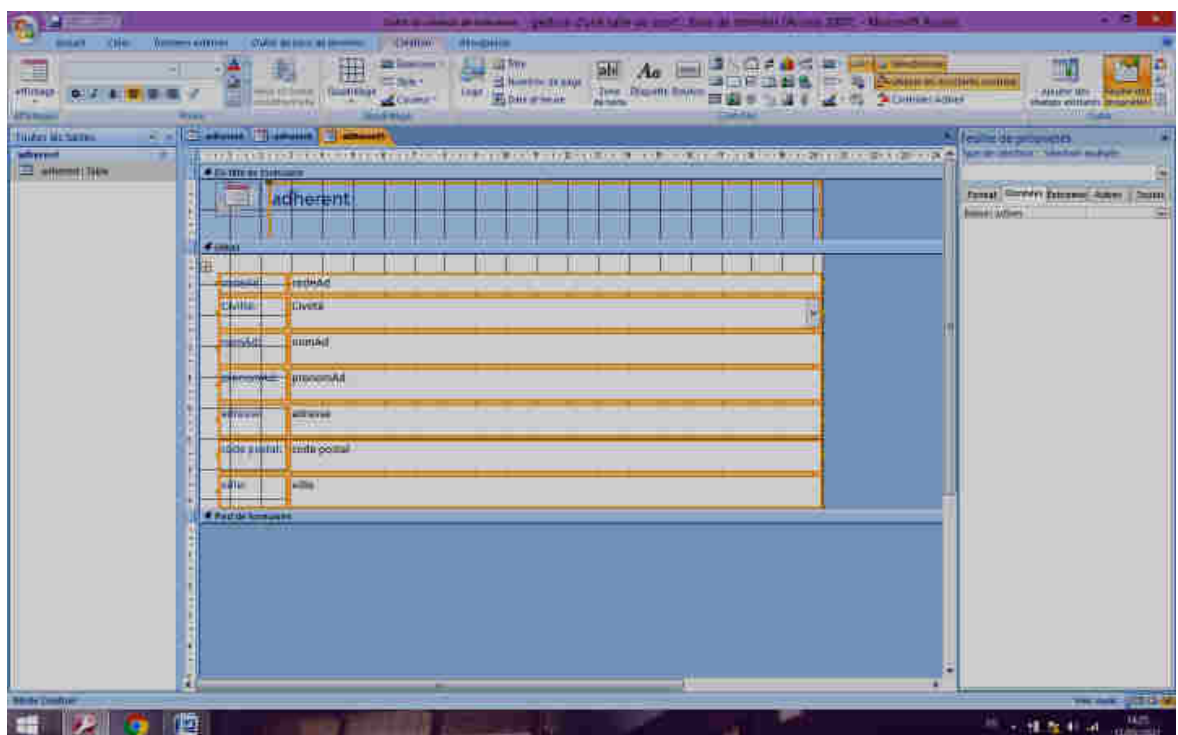


Vous allez travailler sur « l'envers » du décor. A tout moment, vous pouvez passer en mode Visualisation pour voir l'effet produit par vos manipulations.

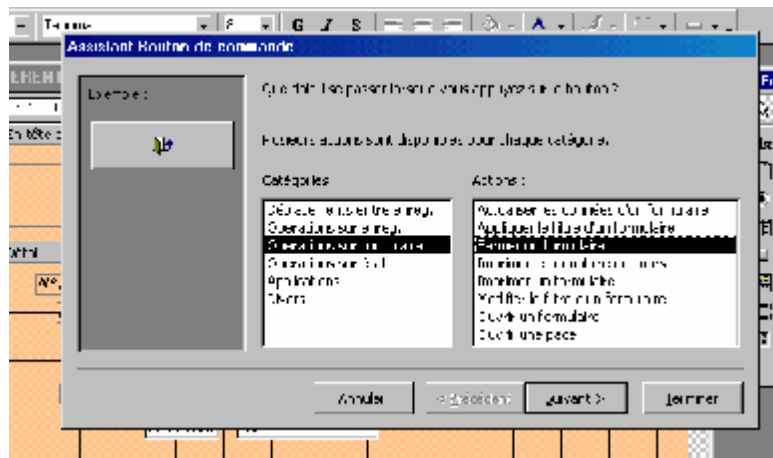
- Cliquez sur la zone de texte **CodeAd** : l'ensemble du contrôle est sélectionné.
- Déplacez-le vers la droite ; vous pouvez constater que l'ensemble du contrôle se déplace.
- Maintenant vous allez déplacer uniquement l'étiquette. Pour cela, cliquez précisément sur le petit carré noir en haut à gauche de l'étiquette. Le curseur prend la forme d'une main doigt levé.
- Rapprochez l'étiquette de la zone de texte. Observez bien de quelle manière apparaissent les marques de sélection dans les différents cas.
- Les étiquettes sont des emplacements où on peut écrire ce qu'on veut, alors que les zones de textes correspondent pour le système à des champs qui ont été définis dans la base de données.

Ainsi vous ne pouvez changer le mot **CodeAd** de la zone de texte à fond blanc sans perturber le fonctionnement. En revanche, le mot **CodeAd** sur fond gris de l'étiquette peut être modifié.

- Sélectionnez le mot CodeAd ainsi
- Tapez à la place N° Adhérent
- Agrandissez la fenêtre au maximum
- Ecrivez Adhérent à la place de Civilité dans l'étiquette
- Ecrivez **NOM** à la place de **NomAd**
- Supprimez l'étiquette **[CodePosta]** et l'étiquette **[Ville]**
- Vous obtenez quoi.
- Rapprochez les étiquettes des zones de texte concernées
- Cliquez Affichage En-tête/Pied de formulaire.
- Ouvrez la boîte à outils
- Cliquez l'outil **Aa**
- Créez une étiquette dans l'en-tête de formulaire (il faut la dessiner c'est-à-dire cliquer glisser depuis le coin gauche supérieur jusqu'au coin droit inférieur).
  - Tapez ADHERENTS dans l'étiquette. Sélectionnez le cadre pour mettre l'étiquette en 24 gras.
  - Cliquez avec le bouton droit sur le fond de l'en-tête de formulaire et dans l'option Couleur d'arrière-plan remplissage, choisissez une couleur **orange clair**. Faites la même chose pour la partie **[Détail]** du formulaire.
  - Vous allez formater en une fois toutes les étiquettes. Pour les sélectionner, cliquez sur la première (N° Adhérent) puis appuyez sur la touche **[MAJ]** du clavier et maintenez-la enfoncée puis cliquez sur chacune des autres étiquettes une par une. Lorsque la sélection est faite, mettez les étiquettes en gras italique. Peut-être certaines d'entre elles seront trop étroites... il vous suffira de les agrandir légèrement comme n'importe quel objet de dessin.
  - Vous allez améliorer l'alignement des zones de texte et des étiquettes. Sélectionnez les zones de texte (fond blanc) de la première partie, depuis **CodeAd** jusqu'à **CodePosta]**. Cliquez avec le bouton droit dans cette sélection et choisissez **Alignement** puis **[Gauche]**.
  - Faites la même chose pour les zones de texte de la deuxième partie.
- Aligned les étiquettes de la première partie sur la droite puis celle de la deuxième partie. Page suivante, voir résultat à obtenir



- Passez en mode **Feuille de données**
- Dans la boîte à outils, (affichez-la si nécessaire) choisissez l'outil **Bouton de commande**.
- Dessinez un petit rectangle avec cet outil dans la zone en-tête du formulaire, à droite de l'étiquette **ADHERENTS**.
- Choisissez les paramètres suivants : **Opérations sur formulaire** et **Fermer formulaire**



- Dans la boîte suivante, choisissez **Texte** **Fermer Formulaire**, **OK**, puis donnez comme nom à votre bouton **Fermer Adhérents**.
- Passez en mode **Affichage** pour juger du résultat. Testez le fonctionnement de votre bouton.

### III Manipulation des données et création des requêtes

En matière de bases de données, la requête est quelque chose d'important car cela sert à beaucoup de choses.

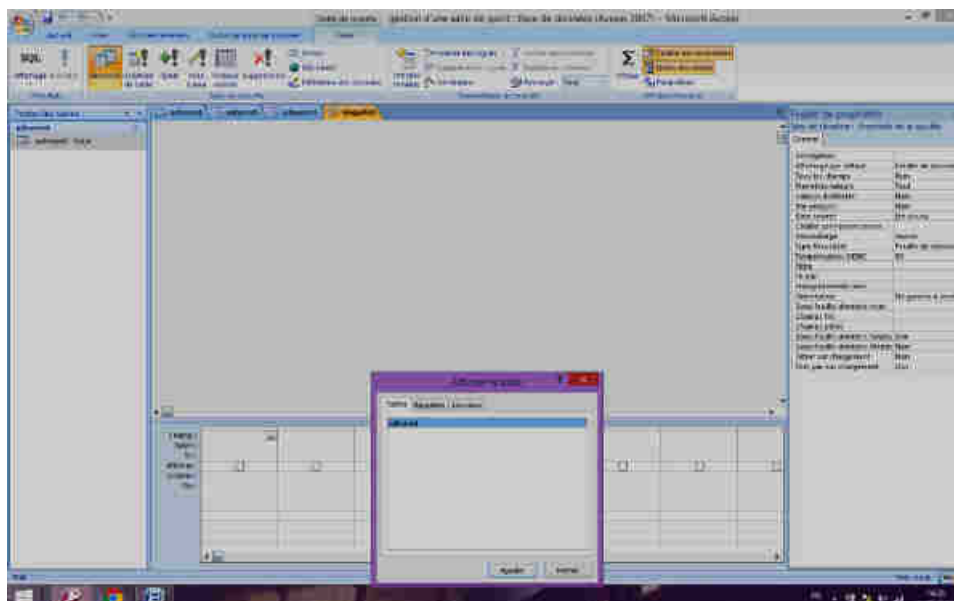
C'est d'abord une question qu'on pose au système (exemple : quels sont les clients qui habitent Annaba ? ou Quels sont les bateaux retenus pour telle période ?). Le système fournit une réponse sous forme de liste. L'avantage de définir une requête (par rapport à une simple interrogation) est que vous pouvez l'enregistrer et vous en resservir plus tard. La réponse ultérieure du système tiendra compte, bien entendu, des mises à jour qui auraient eu lieu entretemps.

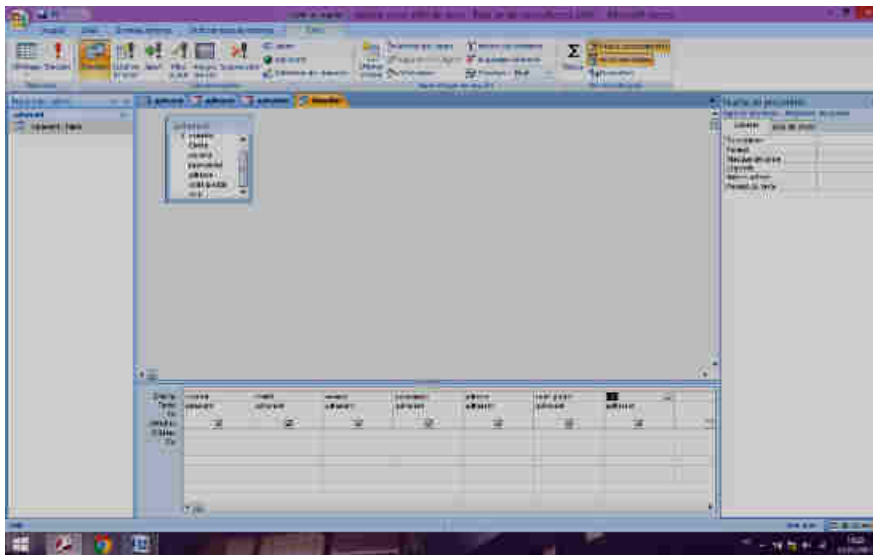
Avec une requête on peut aussi faire des calculs, des regroupements, etc.

Dans le cas qui nous intéresse, vous allez utiliser la requête pour regrouper des tables et ainsi créer un formulaire qui affiche les données venant de ces tables.

#### **Requête sélection**

- Dans la fenêtre **Base de données**, placez-vous dans l'onglet **Requêtes**, puis cliquez **Créer une requête en mode Création**.
- Double-cliquez **ADHERENTS** puis fermer.





- Double-cliquez les champs suivants : CodeAd, NomAd, Prénom, Ville
- Vous obtenez quoi.
- Sur la ligne Critères du champ Ville saisissez Annaba. Vous obtenez :
- Exécutez la requête en appuyant sur l'icône point d'exclamation de la barre d'outils.
- Le résultat s'affiche : il y a 2 clients à Annaba..
- Refermez la requête ; donnez-lui le nom
- ADHÉRENT Annaba.

### Requête Tri

Vous voulez disposer à tout moment d'une liste des clients triée par codes postaux puis alphabétiquement à l'intérieur de ce classement.

- Requête, Création d'une requête en mode Création
- Ajoutez la table **ADHÉRENTS**. Fermez.
- Ajoutez les champs CodePostal, NomAd, Prénom, Adresse, Ville
- Dans la ligne Tri des champs CodePostal et NomAd, choisissez tri Croissant. Un premier tri va s'effectuer sur le code postal, premier champ rencontré par le système puis un deuxième tri sur le champ Nom.
- Exécutez la requête puis refermez-la en lui donnant le nom Liste des clients par localités.

### Requête regroupement de tables

Cette requête a pour but de créer un formulaire regroupant les données de plusieurs tables.

- Créez une nouvelle requête en mode Création.
- Ajoutez les tables **RÉSERVATIONS**, **CROISIÈRES** et **ADHÉRENTS**. Fermez.
- Ajoutez tous les champs de **RÉSERVATIONS** et **CROISIÈRES** et le champ VILLE de la table **ADHÉRENTS**.
- Pour l'instant, inutile d'exécuter la requête, car il n'y a pas de données saisies dans les tables. Refermez-la et donnez-lui comme nom **Pour formulaire réservations**.
- Dans la fenêtre Base de données, placez-vous sur la requête que vous venez de créer et cliquez sur Formulaire instantané de façon à créer votre formulaire automatiquement en se basant sur les tables regroupées dans la requête.
- Supprimez le contrôle RESERVATIONS.CodeC
- Comme vous l'avez fait dans le formulaire LOCATIONS, vous allez remplacer CodeAd par une liste déroulante qui affichera les noms en clair. Supprimez le contrôle puis dessinez avec la boîte à outils une zone de liste modifiable puis laissez-vous guider par les écrans en choisissant la table ADHÉRENTS et en demandant l'affichage des champs CodeAd, NomAd, Prénom. N'oubliez pas de stocker la valeur dans le champ CodeAd.
- Remplacez aussi le contrôle CROISIÈRES.CodeCrois par une liste modifiable (choisissez la table **CROISIÈRES** et demandez l'affichage des 2 premiers champs) et stockez la valeur dans le champ CodCrois. Supprimez le contrôle NomCrois car cela ferait double emploi.
- Passez en mode Affichage et remplissez votre formulaire pour tester son fonctionnement avec cette réservation : Monsieur Saadi Moncef réserve une croisière AU FIL DE L'EAU pour le 20/07 pour un groupe de 32 personnes (17 adultes et 15 enfants)
- Constatez que lorsque vous choisissez une croisière dans la liste déroulante, ses caractéristiques se reportent automatiquement dans les autres champs. Ceci se fait en fonction de tout ce que vous avez déterminé dans le modèle de base. De même, la ville où habite l'adhérent se reporte automatiquement.
- Améliorez la présentation du formulaire en le mettant en couleur, en alignant mieux les différentes étiquettes et zones de texte. Insérer un en-tête de formulaire avec une étiquette de titre **RÉSERVATIONS DE CROISIÈRES**, comme ci-dessous.
- Le formulaire commence à prendre tournure. Toutefois vous pensez peut-être avec raison qu'il serait intéressant de pouvoir calculer directement le coût de cette réservation pour le client. Le



nombre d'adultes et d'enfants est connu, de même que les tarifs. Il faudrait pouvoir créer un champ qui calcule. C'est possible ! Mais rappelez-vous, le formulaire est basé sur une requête. C'est dans celle-ci qu'il faut préparer le champ calculé ; ensuite il suffira d'insérer ce nouveau champ dans le formulaire.

- Fermez le formulaire, ouvrez la requête Pour formulaire RÉSERVATIONS en mode Création.
- Placez-vous sur le premier champ libre (tout-à-fait à droite de tous les autres) et saisissez très exactement l'expression de calcul suivante (*attention, toute erreur de crochet ou de majuscules provoquera un dysfonctionnement*). N'oubliez pas le signe deux-points après Coût. Rappel : le crochet s'obtient en actionnant la touche AltGr et la touche 5 ou °.

**Coût:[NbAd]\*[TarifAd]+[NbEnf]\*[TarifEnf]**

- Exécutez la requête pour voir si le calcul se fait bien.
- Il reste à insérer le champ calculé dans votre formulaire.
- Fermez la requête en l'enregistrant. Ouvrez le formulaire **RÉSERVATIONS** en mode Création.
- Cliquez l'icône Liste des champs. Celle-ci a l'avantage de se tenir toujours à jour des modifications que l'on pourrait apporter au support de notre formulaire.
- En effet, dans la liste qui apparaît vous voyez le champ **Coût** que vous venez de créer. Prenez-le avec la souris et placez-le dans le formulaire.
- Passez en mode **Affichage** pour voir.
- Enregistrez les deux réservations suivantes dans votre formulaire.

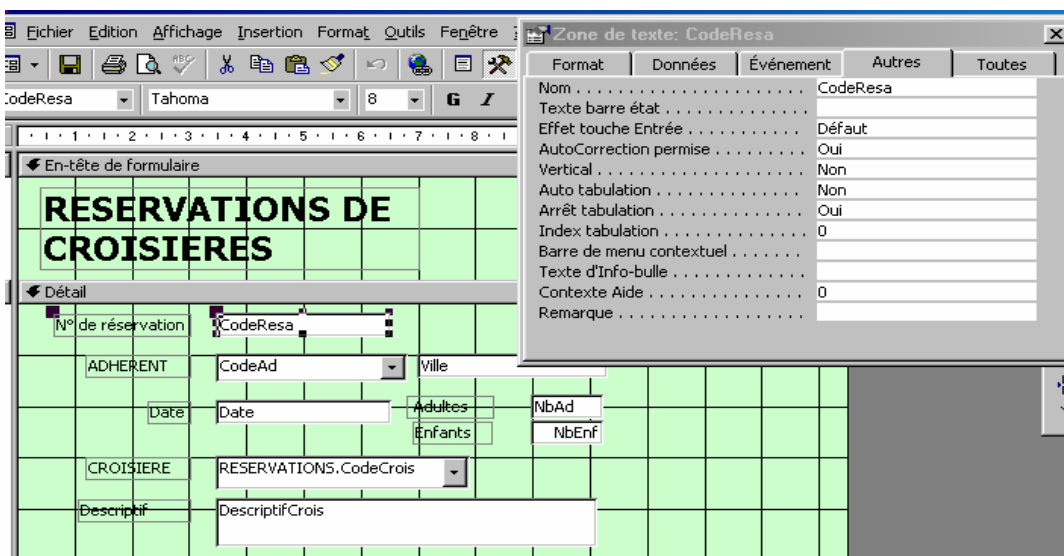
Monsieur Layouni Zoubir	Madame Saliha taty
réserve une croisière DETENTE	réserve une croisière EVASION
pour le 23/06 pour 40 adultes et 11 enfants entre 3 et 12 ans	pour le 7 juillet pour 52 personnes (35 adultes et 17 enfants)

### Propriétés des champs

Vous pouvez constater que dès que vous positionnez le nom de la croisière, le reste s'affiche de lui-même. Toutefois vous avez remarqué que la touche tabulation qui vous permet de passer d'un champ à l'autre passe un peu du coq à l'âne car les champs ont été déplacés ou rajoutés et cela ne suit pas l'ordre qui vous faciliterait la saisie. Il y a heureusement un remède.

- En mode Création, cliquez l'icône Propriétés de la barre d'outils.

Cette fenêtre vous montre toutes les propriétés de tous les éléments de votre écran. Ici vous êtes dans le formulaire Réservations. Cliquez sur le champ CodeResa et dans la fenêtre des propriétés, prenez l'onglet **Autres**. Regardez la ligne **Index Tabulation** il y a 0, ce qui veut dire que c'est la première zone à remplir.



- Sans fermer la fenêtre des propriétés, cliquez sur **CodeAd** ; mettez 1 à la place de l'index tab. Continuez de la même manière pour tous les autres champs, jusqu'au **Coût** qui sera le 10<sup>e</sup>. Dorénavant, la touche **tabulation** vous emmènera du début à la fin selon votre paramétrage. Un nouvel adhérent s'inscrit :

Layouni Zoubir
84 Cité Beveledere
Tour C N° 27
Tél. 02 97 45 78 12 Né le 7/3/62
Loue pour 3 semaines pour 4 personnes à compter du 10 août

- Vous allez l'enregistrer dans le formulaire **ADHÉRENTS** ; toutefois vous avez remarqué que vous êtes obligé de taper en majuscules le nom de famille ; il serait préférable de disposer d'un système où la saisie serait faite en majuscules même si on tape en minuscules. Il y a une solution !
- Ouvrez le formulaire en mode **Création**. Placez-vous sur le champ **NomAd**, ouvrez la fenêtre des propriétés, onglet **Format**. Sur la ligne **Format**, saisissez simplement le caractère > (*il forcera l'affichage en majuscules dans ce champ*).
- Repassez en mode **Affichage** pour saisir votre nouvel adhérent et constatez que votre manipulation a porté ses fruits.
- Enregistrez la location dans le formulaire **LOCATIONS**.

### Requête Mise à jour

Malheureusement, les prix des locations de bateaux augmentent de 10 %. Vous allez devoir changer les tarifs de la table **TYPES DE BATEAUX**. *La mauvaise solution serait de le faire manuellement ; bien entendu le cas de notre exemple est tellement petit que ce ne serait pas gênant de faire les opérations à la main. Mais il faut toujours penser « gros volumes » avec une base de données.*

Vous allez donc créer une requête qui procédera automatiquement à l'augmentation des prix.

- Dans l'élément **Requêtes** de la fenêtre Base de données, créez une requête en mode Création.
- Ajoutez la table **TYPES DE BATEAUX**, affichez tous les champs sauf les nombres de personnes.
- Cliquez sur le menu Requête et choisissez **Requête** **Mise à jour**, ce qui aura pour effet d'ajouter une ligne Mise à jour dans vos champs.
- Saisissez comme ci-dessous les expressions de calcul dans les champs des tarifs (attention, pas d'erreur de saisie, sans quoi cela ne marchera pas)

Champ :	CodeTypeBateau	Catégorie	TarifHtSaison	TarifBsSaison
Table :	TYPES DE BATEAUX	TYPES DE BATEAUX	TYPES DE BATEAUX	TYPES DE BATEAUX
Mise à jour :			[TarifHtSaison]*1,1	[TarifBsSaison]*1,1
Critères :				
Ou :				

- Exécutez la requête. Attention, ne faites la manœuvre qu'une seule fois ! Si vous la lancez plusieurs fois, les prix seront augmentés plusieurs fois de 10 %, et à chaque fois... sur la base déjà augmentée... Il faudra alors calculer le coefficient capable de ramener les choses à leur état initial. Donc vous l'exécutez une seule fois et vous cliquez sur Feuille de données pour voir le résultat.

### **Requête Somme/Regroupement**

Vous aimeriez disposer d'un moyen permanent de connaître ce qu'a rapporté globalement chaque type de croisière

- Créez une requête en mode Création, ajoutez la requête Pour formulaire **RÉSERVATIONS**, affichez les champs **Coût** et **CROISIERES.CodeCrois**.
- Cliquez l'icône  $\Sigma$  de la barre d'outils qui aura pour effet d'ajouter une ligne **Regroupement** dans vos champs.
- Dans cette ligne pour le champ **Coût**, déroulez la liste et choisissez **Somme**. Pour le deuxième champ, laissez **Regroupement**.
- Exécutez la requête. Fermez la requête et enregistrez-la sous le nom **Chiffre d'affaires croisières**. Vous pourrez à tout moment lancer cette requête et connaître ainsi votre chiffre d'affaires croisières.

### **Requête sélection.**

Vous souhaiteriez disposer à moments réguliers de la liste des clients ayant loué un bateau et de celle des clients ayant réservé une croisière. Cela pourra être édité sous forme d'état, ce que vous allez étudier dans le chapitre suivant. Mais il faut déjà disposer de la requête qui fait l'extraction de données.

- Créez une requête en mode **Création**, ajoutez les tables **ADHÉRENTS**, **RÉSERVATIONS**, **CROISIÈRES**.
- Affichez les champs : **Nom**, **Prénom**, **Ville**, **Individuel** de la table **ADHÉRENTS**, **NbAd** et **NbEnf** de la table **RÉSERVATIONS**, **NomCrois** de la table **CROISIÈRES**.
- Pour connaître le nombre total de personnes de chaque croisière, créez un champ calculé sur le premier champ vierge, ainsi paramétré :  
Nb Total :[NbAd]+[NbEnf]
- Exécutez la requête.

Vous voudriez que la liste soit triée par ordre alphabétique des clients. Que pouvez-vous modifier dans la requête pour l'obtenir ?

- Fermez et enregistrez la requête sous le nom **CROISIÈRES PAR CLIENT**.

*PARTIE 4 TRAVAUX D'APPLICATION*

# Travaux d'application

## Introduction

Les travaux d'applications sont conçu pour être une synthèse des acquis théorique vue dans le modules 5 « manipulation de base de données », ils sont poses sous forme d'atelier enchaîner qui abouti a l'élaboration d'une application de base de donnée touchant la totalité des fonctionnalités d'un SGBDR Tel que MS Access. Le projet est porté sur **la GESTION D'UN FOURNISSEUR DE PHARMACIES.**

## 1 Création de tables

### Objectifs :


- Créer une base de donnée sous MS Access
- Créer des tables sous MS Access
- Définir des Clés

Un laboratoire de médicaments désire informatiser la gestion de stock des médicaments ainsi que la gestion des commandes, et les médicaments périmés et d'autres opérations manuelles.

- 1) Créer une nouvelle base de données sur le disque dur, en précisant le nom de la base « gestion d'un Laboratoire de médicaments ».
- 2) Créer la table « médicaments » pour le stockage des informations concernant tous les médicaments.
  - Sur la 1<sup>ère</sup> colonne, vous préciserez les noms des champs,
  - Sur la 2<sup>ème</sup> colonne, vous préciserez les types des champs,
  - Sur la 3<sup>ème</sup> colonne, vous donnez les descriptions des champs.

La liste des champs de la table « **médicaments** » est la suivante :

Nom du champ	Type de données
Référence médicament	Numérique
Libelle médicament	Texte
Date de péremption	Date
Prix unitaire	Numérique
Quantité en Stock	Numérique
Quantité min	Numérique

- 4) Une fois les noms des champs spécifiés, vous devez définir la clé primaire pour la table en utilisant l'icône .

- 5) Créer la table Catégorie de médicament nommée (Catégorie)

La liste des champs de la table « **Catégorie** » est la suivante :

Nom du champ	Type de données
Numéro Catégorie	Numérique
Libelle Catégorie	Texte

- 6) Créer la table pharmacie

La liste des champs de la table « Pharmacie » est la suivante :

Nom du champ	Type de données
Numéro pharmacie	Numérique
Nom pharmacie	Texte
Adresse	Texte
Tel	Texte

- 7) Fermez la base de données « **Fournisseur de pharmacies** » avec **Fichier / Fermer.**

## 2- Remplissage des tables

### Objectifs :

- Remplir et saisir les données table sous MS Access
- Mettre en forme des tables et Champs
- Gérer les tables sous MS Access

Entrez les données des médicaments dans la table « **médicaments** », en cliquant sur le bouton table puis ouvrir.

- Informations à saisir dans la table médicaments :

Référence Médicament	Libelle Médicament	Date Péréemption	Prix unitaire	Quantité en Stock	Quantité en Seuil
0001	Doliprane	01/01/2004	12	100	20
0002	Aspégic	12/03/2004	50	200	50
...	...				
...	...				

- 1) Triez les enregistrements
- 2) Recherchez l'enregistrement n° 0007
- 3) Supprimer l'enregistrement n° 0007
- 4) Modifier la largeur des colonnes
- 5) Modifier la hauteur des lignes
- 6) Masquer la colonne « **Quantité min** »
- 7) Réafficher la colonne « **Quantité min** ».
- 8) Figez la colonne « **Libelle médicament** »
- 9) Changer la couleur de l'arrière plan de la table
- 10) Faites une copie de la table (Données et structure)
- 11) Renommez la table « Médicament »
- 12) Supprimez la table que vous venez de renommer.
- 13) Entrez les données des Catégories dans la table « **catégorie** »

Numéro catégorie	Libellé catégorie
01	Antibiotique
02	Anti-inflammatoire
03	Antalgique
...	...

- 14) Entrez les données des pharmacies dans la table « pharmacie »

Numéro pharmacie	Nom pharmacie	Adresse	Tel
001	Pharmacie LAYOUNI	Annaba	56-89-75-55
002	Mamipharme	Constantine	75-48-89-36
...	.....		

### 3- Propriété des tables

#### Objectifs :

- Définir et configurer les propriétés des tables
- Définir et configurer les propriétés des champs

- A) Copier la table « médicament » sous un autre nom « médicament1 » .
- B) Ouvrez la table médicament1 et définissez les propriétés des champs.

*Après avoir défini chaque nouveau format, basculez du mode création vers le mode feuille de données pour tester le format que vous venez de créer*

1. Référence médicament doit être numérique
2. Le Libellé du médicament doit avoir une entrée indispensable, sachant qu'une chaîne vide est interdite
3. La quantité min doit avoir une valeur par défaut « 10 ».
4. Le libellé du médicament doit être saisi en majuscule et affiché en bleu
5. Le tel domicile doit comporter uniquement des chiffres
6. La date de péréemption doit être supérieure à la date du système
7. La Quantité min doit être comprise entre 10 et 100
8. Pour la date de péréemption, elle doit être comprise entre date système +1 jour et la date système + 5ans.
9. Lorsque vous entrez une date qui ne respecte pas la règle de validité, Access ouvre une boîte de dialogue affichant un message d'erreur. Modifiez le message de cette boîte de dialogue.
10. Créer une règle de validité qui permet de contrôler si la quantité en stock est supérieure à la quantité min. (**Propriété de la table**). Afficher un message d'erreur personnalisé.

## 4- Les Requêtes

- Objectifs :**
- Créer des requêtes à l'aide d'interface graphique de MS Access
  - Créer des requêtes à l'aide du code SQL

1. Créez une requête n'affichant que les médicament(s) dont la date de péremption est inférieure à 2022.

Procédez comme suit :

- Appuyez sur le bouton **Requêtes** puis **Nouveau**
- **Mode création** puis **Ok**
- Sélectionnez la **table médicaments** et appuyez sur **Ajouter**, puis **Fermer**
- Glissez les champs que vous voulez afficher dans la requête
- Placez le critère **<=31/12/2022** dans le champ date de péremption.

2. Enregistrez la requête sous le nom « liste 2022 » avec **Fichier / Enregistrer**.
3. Ajoutez le champ « code catégorie » dans la table « médicament » (en utilisant la commande Insérer Liste de choix) afin de pouvoir créer une requête qui regroupe les deux tables « médicament » et « catégorie ».
4. Modifier la requête « liste des médicaments par catégorie », en affichant que les médicaments ayant la catégorie « Antibiotique ».
5. Modifier la requête « liste des médicaments par catégorie », en affichant que les médicaments ayant la catégorie « Antibiotique » et la date de péremption supérieure à 2003.
6. Créer une nouvelle table « Achat », pour chaque Achat on veut savoir le Nom pharmacie, le médicament, la quantité achetée, la date d'achat.
7. Créer la requête qui affiche la liste des achats effectués.
8. Modifier la requête en affichant que les achats effectués par une pharmacie spécifiée comme paramètre.
9. Faites une copie de la requête.
10. Supprimez la requête.
11. Modifiez la requête « Liste des médicaments par catégorie » par « liste des médicaments par catégorie et par date » en paramétrant la catégorie et la date.
12. Créer une requête sous le nom « Liste des médicaments commençant par C » permettant d'afficher l'ensemble des médicaments dont le nom commence par la lettre C.
13. Modifier la requête précédente de telle manière à pouvoir afficher toutes les médicaments dont le nom commence avec le caractère spécifié en tant que paramètres.
14. Créer une nouvelle requête nommée « liste des médicaments achetés pendant une période » permettant d'afficher la liste des médicaments achetés pendant une date début et une date fin.
15. Créer une requête de la table médicament qui permet de calculer automatiquement le nombre d'années restant de chaque médicament pour être périmé.
16. Créer une requête permettant l'affichage de la liste des pharmacies se trouvant à Annaba.
17. Créer une requête permettant l'affichage de la liste des pharmacies se trouvant à Annaba et dont le numéro de téléphone commence par le 77.
18. Créer une requête permettant l'affichage du nombre total d'achat par pharmacie pendant une année.
19. Créer une requête permettant l'affichage du nombre total de médicaments par catégorie donnée comme paramètre.
20. Créer une requête paramétrée permettant d'afficher le montant minimum d'achat d'une pharmacie.
21. Afficher le prix de médicament le plus cher d'une catégorie donnée.
22. Créer la requête qui affiche les noms des médicaments dont la date de péremption est aujourd'hui.
23. Augmenter de 10% le prix des médicaments dont le prix est supérieur à 100.

### 24. Requête Création de table

- Requête création de table pour copier tous les médicaments dont la quantité est inférieure à 30, dans une table **Rupture**.
- Requête création de table pour copier juste les codes et les désignations des médicaments dans une table médicament.
- Requête création de table pour copier la table pharmacie dans une nouvelle table **pharmacie sauve**.

## 25.Requête d'ajout

- Requête ajout permettant d'ajouter des pharmacies dans la table pharmacie à partir de la table pharmacie sauve.

## 26.Requête Suppression

- Requête suppression permettant de supprimer un médicament quelconque dans la table médicament.
- Requête suppression permettant de supprimer les achats passés avant le (01/01/2021).
- Requête suppression permettant de supprimer les achats d'une pharmacie donnée.

## 27.Requête mise à jour






- Requête mise à jour permettant d'effectuer une réduction de 10% sur les prix unitaire de tous les médicaments.
- Requête mise à jour permettant d'augmenter la quantité disponible de 20% pour tous les médicaments.

## 5- Les Formulaires

### Objectifs :

- Créer des Formulaires de saisie en mode création
- Mettre en forme les formulaires

### 1. Créez un nouveau formulaire de la table « médicament » en « mode création ».

- Appuyez sur le bouton **Formulaire**, puis **Nouveau**.
- Choisissez la table **médicament**.
- Mode création.
- Changez la couleur d'arrière plan.
- Insérez le titre « gestion du personnel » 
- Modifier le format du titre « 3D enfoncé ». 
- Insérez les champs de la table médicament. 
- Insérez un champ affichant la date du jour.
- Modifiez le format des champs (taille, police, couleur, style).
- Insérez une image. 
- Basculez en mode formulaire pour visualiser les modifications. 

### 2. Utilisez les boutons de navigation, situés en bas du formulaire, pour passer d'un enregistrement à un autre.

### 3. Créez un deuxième formulaire mode création de la table Catégorie.

### 4. Modifiez le format de votre formulaire en basculant en mode création.

### 5. Créez un formulaire de la table « médicament » dans le formulaire (représentant les valeurs en lignes et colonnes comme un tableau d'une feuille).

### 6. Insérez dans votre formulaire un graphique Excel représentant les données suivantes :

Année	Antibiotique	Antinflammatoire
2000	1012	1520
2001	1520	2006
2002	1789	2500

### 7. Créez un formulaire de type **graphique** représentant les données de la requête « **Liste des médicaments par catégorie** ».

### 8. Le formulaire suivant sera de type **sous-formulaire**. Etant entendu que dans une catégorie donné, il y a plusieurs médicament, il est très utile pour vous d'obtenir dans un tableau tout les médicaments d'une catégorie.

### 9. Jusqu'à présent, vous avez enregistré dans les tables des données rédigées par écrit. Le champ de type objet OLE permet d'enregistrer dans la table des images ou des photos. Voici pas à pas la méthode pour l'insertion d'un champ de type OLE dans une table.

- Ouvrez la table « **pharmacie** ».
- Ajoutez un champ de type **OLE**, nommé « **logo** ».
- Basculez en mode **feuille de données**.
- Positionnez le curseur sur le champ « **logo** » du 1<sup>er</sup> enregistrement.



- Choisissez le menu **Insertion / Objet**.
- Cherchez le nom du fichier à insérer à partir du bouton Parcourir.
- Spécifier le chemin de recherche et appuyer sur Ouvrir.
- Créer un nouveau formulaire de la table Pharmacie en insérant le nouveau champ « logo ».

## 6- Les Formulaires (suite)

### Objectifs :

- Créer des Formulaires de saisie en mode Assistant
- Créer et manipuler les contrôles

### Travail effectué avec:

- Un micro-ordinateur par 2 stagiaires
- Un logiciel de base de données MS ACCESS
- Une imprimante

1. Créez un nouveau formulaire de la table « **médicament** » en « **mode assistant** » pour ajouter de nouveau enregistrements sans modifier les anciens.
2. Créez un autre formulaire de la table médicament permettant de modifier les données des champs à part celui du code médicament.
3. Copier le formulaire et modifiez dans les propriétés de telle façon à pouvoir consulter les enregistrements sans mise à jour.
4. Créez un formulaire principal reliant l'ensemble des formulaires précédents. (Utilisation des boutons).
5. Ajoutez dans les formulaires (ajout modification et consultation) un bouton de fermeture ; pour pouvoir revenir au formulaire principal.
6. Créer un formulaire Création et mise à jour de la table « Pharmacie » en utilisant des boutons de commandes.
  - Bouton Création
  - Bouton Modification
  - Bouton Suppression
  - Bouton Consultation
  - Boutons de déplacements
  - Bouton Quitter qui revient au menu principal
7. Créer un formulaire qui permet l'affichage des informations de la table Achat en accordant une remise de prix pour chaque pharmacie de 0,75.

## 7- Les Etats

### Objectifs :

- Créer et imprimer les états basant sur des tables et des requêtes
- Créer des regroupement
- Mettre en forme les états

1. Créez un Etat représentant la liste des médicaments

Appuyez sur Etat puis Nouveau

Sélectionnez la table et le type d'état que vous souhaitez créer, cliquez sur OK.

Sélectionnez les champs à insérer dans l'état (de la même façon que l'insertion des champs dans un formulaire).

Cliquez sur Suivant.

Indiquez comment doit s'effectuer le regroupement des enregistrements et Cliquez sur Suivant.

Sélectionnez le ou les champs en fonction desquels les enregistrements doivent être triés lors de l'impression de l'état et cliquez sur suivant.

Choisissez la présentation souhaitée et cliquez sur suivant.

Tapez le titre de l'état.

Activez l'option Placer tous les champs sur une seule page pour que chaque enregistrement soit imprimé sur une seule ligne.

Cliquez sur le bouton Aperçu pour visualiser le résultat.

Basculez en mode création afin de modifier la présentation de votre Etat.

2. Créez un nouvel Etat permettant le regroupement des médicaments par catégorie.

3. Créez un Etat style tabulaire permettant d'éditer la liste des pharmacies.

4. Créez un Etat basé sur la table pharmacie pour laquelle il faut générer des étiquettes en vue d'un publipostage.

- Cliquez sur Etat puis sur le bouton Nouveau ;  
 Cliquez sur Assistant Etiquette ;  
 Choisissez la table pharmacie ;  
 Choisissez la mise en forme de vos étiquettes et le format de vos étiquettes ;  
 Placez les champs  
 Cliquez sur le bouton Terminer pour visualiser le résultat.
5. On vous propose la réalisation d'un état où les médicaments seront regroupés par ordre alphabétique.
  6. Créez un état l'Assistant Etat en vous basant sur la table médicaments ;  
 Placez les champs nécessaires pour obtenir un état;  
 Choisissez le champ Nom médicament comme champ de regroupement ;  
 Choisissez un type de présentation, puis un style et cliquez sur le bouton Terminer.
  7. Créer un état de la table médicament basé sur une mise en forme conditionnelle permettant de sélectionner les prix de médicament >50 DA.
  8. Créer un état basé sur une mise en forme conditionnelle permettant de sélectionner les médicaments dont la quantité stock < à la quantité minimal.
  9. Créer l'état qui permet d'afficher la somme des achats de chaque pharmacie par page.
  10. Créer l'état qui permet de regrouper les médicaments par ordre alphabétique (Utiliser l'option trier et regrouper) et changer la propriété de la zone (Gauche\$([libelle médicament] ;
  11. Créer l'état qui affiche à partir d'une requête croisé dynamique le nombre de médicaments par catégorie et par date de péremption.

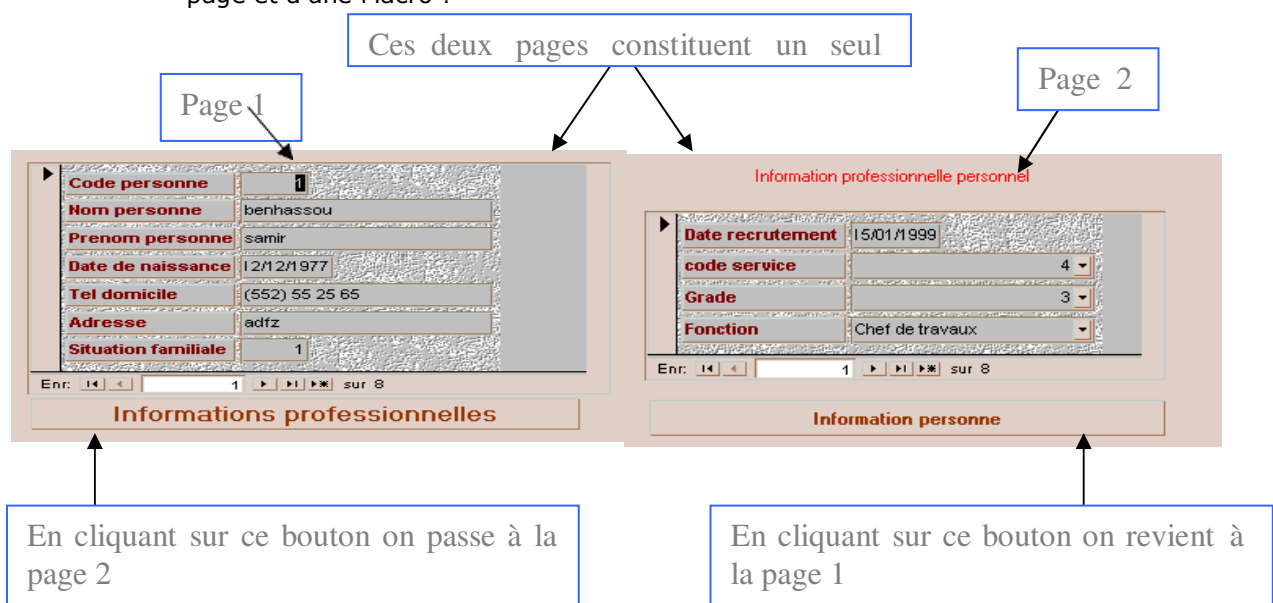
## 8- Les Macros

- Objectifs :**
- Consulter des données dépendants dans deux formulaires, en Utilisant les macros.
  - Gérer et naviguer entre les différents objets de MS Access en utilisant les macros.

### Travail effectué avec:

- Un micro-ordinateur pour 2 étudiants
- Un logiciel de base de données MS ACCESS
- Une imprimante

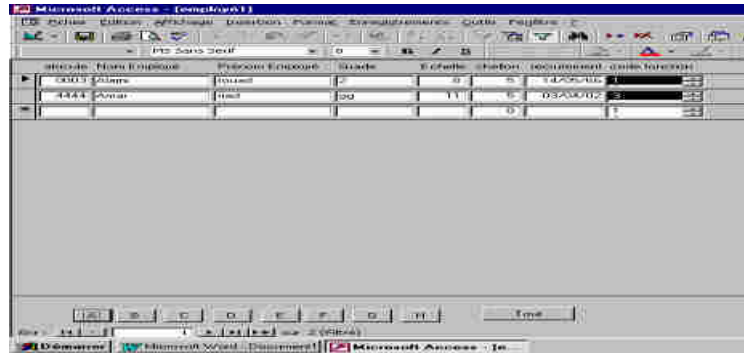
1. Comment synchroniser l'affichage de deux formulaires après ouverture via une macro  
 Exemple (Liste des médicaments par catégorie).
2. Créer des formulaires de la table médicament à plusieurs pages via le contrôle saut de page et d'une Macro ?



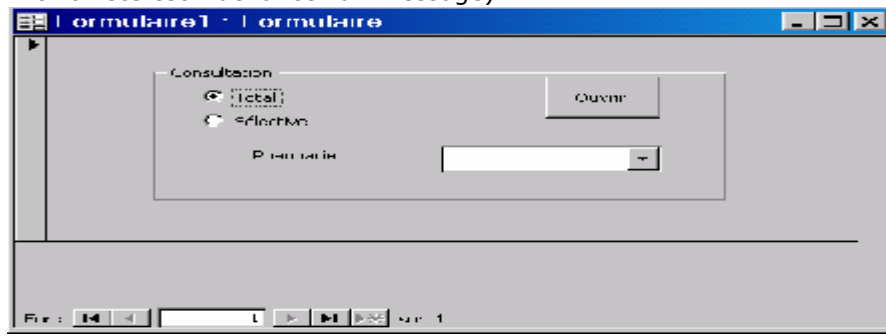
### Exemple

3. Créer la macro qui permet d'extraire des enregistrements de la table médicaments à condition d'avoir la possibilité de limiter des enregistrements dont le nom de médicament commence par une lettre.

### Exemple



1. Créer la macro qui permet d'agrandir un formulaire.
2. Créer le formulaire qui permet de consulter selon des critères
  - La liste des achats de toutes les pharmacies  
OU
  - La liste des achats d'une pharmacie choisie dans la liste de choix (avec un contrôle si la liste est vide lancez un message)



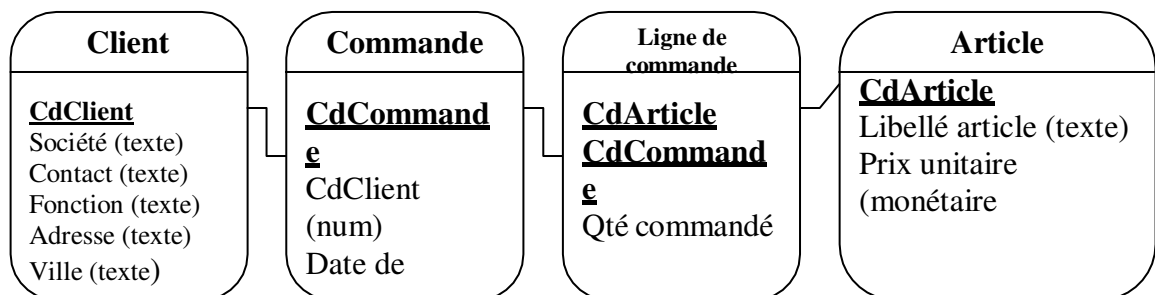
3. Créer la macro qui permet de créer une barre de menu

## 9- Gestion des commandes .

### Objectifs :

- Créer une application de base de donnée sous Access
- Créer un formulaire menu

1. Créez une nouvelle base de données « gestion des commandes ».
2. Créez les tables (Client, Commandes, Lignes-commande et article) pour le stockage des informations de la base de données.



3. Définissez la clé primaire de chaque table.
4. Entrez les données dans les tables pour pouvoir effectuer des tests d'extractions au niveau des requêtes.

### **Tables Clients**

Code client	Société	Contact	Fonction	Adresse	Ville	Pays	Tel	Fax
1	Sarl yahia	yazid	Représentant	Cité Rym	Annaba	Algerie	77-77-89	77-78-79
2	EURL med	mohammed	Assistant Export	Cité bevedere	Annaba	Algerie	77-89-45-88	56-89-78-45
3	Entreprise Med	Tijani Mohamed	Chef de produit	Cité Kouba	Annaba	Algerie	56-89-45	56-56-89
4	Cosider	Salmi brahim	Directeur de marketing	Cité Daksi	Constantine	Algerie	75-58-89	45-55-55
5	imprimerie père et fils	atrousse	Directeur marketing	Cité bellevue	Constantine	Algerie	12-56-89-56	44-44-77-44

### **Tables Commandes**

Code commande	Code client	Date commande	Date livraison
B01	1	05/01/2020	05/02/2020
B02	1	10/12/2021	11/12/2021
B03	2	25/04/2020	25/05/2021
B04	1	23/05/2020	12/06/2020

### **Tables Lignes commandes**

Code ligne	Code commande	Code article	Qté commandée
1	B01	0001	12
2	B01	0002	13
3	B01	0003	20
1	B02	0001	40
1	B03	0003	16
1	B04	0001	20
2	B04	0004	10
3	B04	0002	5
4	B04	0005	12

### **Table Article**

Code article	Libelle article	Prix unitaire
0001	Chaise	200
0002	Table	1000
0003	Bureau	2000
0004	Tableau blanc	1500
0005	Photocopieur	10000

5. Créez :
  - Une requête permettant l'affichage des commandes passées entre le 01/2020 et le 12/2020
  - Une requête paramétrée permettant l'affichage des commandes entre une date de début et une date de fin.
6. Créez une requête permettant l'affichage du nombre de commandes par client à Annaba.
7. Créer une requête permettant l'affichage de tous les clients étrangers.
8. Créer la requête qui permet de sélectionner les clients dont le pays est le Algérie ou dont la ville est Alger.
9. Créez une requête permettant l'affichage de l'entête d'une commande c'est-à-dire (l'ensemble des informations de la commande et le client qui a passé la commande).
10. Créez une requête permettant l'affichage du détail de la commande C'est-à-dire pour chaque ligne commande (code ligne commande, libellé article, prix unitaire, quantité commandée et une expression de calcul permettant de calculer le montant de ligne commande).
11. Créez une requête permettant l'affichage du pied de commande c'est-à-dire pour chaque

commande, le montant total des lignes commande.

- 12.**Créer un formulaire qui permet l'affichage des informations de la table article en accordant une remise de prix pour chaque article de 0,75.

**Prix Article avec Remise**

coda article

libelle

prix unitaire  **Prix unitaire avec remise**

- 13.**Créer le formulaire qui permet d'afficher la liste des bons de commandes par client.

- 14.**Créer un formulaire permettant l'affichage d'une commande.

code client  **BON COMMANDE N°**

Date commande   
Date livraison

détail commande

Code ligne	Code article	Libelle article	Qté commandé	Prix unitaire	montant
1	1	chaise	10	12,00 F	120,00
2	2	ta	15	10,00 F	150,00

Enr:  sur 2

**Total commande**


- 15.**Créer un état qui permet l'affichage de la liste alphabétique des clients.

- 16.**Créer l'état qui permet de regrouper les commandes livrées par semaine.

- 17.**Créer l'état qui permet l'affichage des informations de la tables article.

- 18.**Créer l'état qui permet d'afficher la liste des clients qui ont passé une commande

**Menu général Gestion de commandes**



**Clients** **Articles**

**commandes**

*Impression de la liste des clients qui ont passé des commandes*

Quitter Microsoft Access Quitter le menu général

## PARTIE 5 EXERCICES D'APPLICATIONS (LANGAGE SQL)

Gestion d'une bibliothèque.( l'énoncé complet du problème est sur la Plateforme, E-learning : elearning.univ-annaba.dz :

### Exercice 1 :

```
Select nom_éditeur, ville, région
from éditeurs
```

<i>nom_éditeur</i>	<i>ville</i>	<i>région</i>
<i>New Moon Books</i>	<i>Boston</i>	<i>MA</i>
<i>Binnet &amp; Hardley</i>	<i>Washington</i>	<i>DC</i>
<i>Algodata Infosystems</i>	<i>Bruxelles</i>	<i>(null)</i>
<i>Five Lakes Publishing</i>	<i>Chicago</i>	<i>IL</i>
<i>Ramona, éditeur</i>	<i>Lausanne</i>	<i>(null)</i>
<i>GGG&amp;G</i>	<i>Munich</i>	<i>(null)</i>
<i>Scootney Books</i>	<i>New york</i>	<i>NY</i>
<i>Editions Lucerne</i>	<i>Paris</i>	<i>(null)</i>

(8 ligne(s) affectée(s))

### Exercice 2 :

```
SELECT nom_employé, pn_employé, date_embauche, position_employé
FROM employé
WHERE (nom_employé LIKE 'L%') AND (date_embauche LIKE '%1990%') AND (position_employé
BETWEEN 10 AND 100)
```

<i>nom_employé</i>	<i>pn_employé</i>	<i>date_embauche</i>	<i>position_employé</i>
<i>Laid</i>	<i>Zohir</i>	<i>7/déc./1990 0:00</i>	<i>35</i>

(1 ligne(s) affectée(s))

### Exercice 3 :

```
SELECT nom_employé, id_éditeur
FROM employé
ORDER BY id_éditeur, nom_employé
```

<i>nom_employé</i>	<i>id_éditeur</i>
<i>Barizam</i>	<i>0736</i>
<i>Iskander</i>	<i>0736</i>
<i>Jordanien</i>	<i>0736</i>
<i>Karabaghli</i>	<i>0736</i>
<i>Lechheb</i>	<i>0736</i>
<i>Menaceur</i>	<i>0736</i>
<i>Odaifia</i>	<i>0736</i>
<i>Salhi</i>	<i>0736</i>
<i>Senhadji</i>	<i>0736</i>

...etc ... (43 ligne(s) affectée(s))

### Exercice 4 :

```
SELECT nom_auteur, pays, adresse
FROM auteurs
WHERE pays IN ('FR','CH', 'BE')
```

<i>nom_auteur</i>	<i>pays</i>	<i>adresse</i>
<i>Layouni</i>		<i>Cité belvedere Tour E</i>
<i>Bechker</i>	<i>BE</i>	<i>4, chemin de la Tour de benmhidi</i>

selmi CH 57, avenue des ecoles  
 Senhadji CH 55, rue Kouba  
 Chettah CH 48, rue de telemli  
 Chergrouche CH 48, rue de montplaisant  
 ...etc... (20 ligne(s) affectée(s))

---

**Exercice 5 :**

```
SELECT position_employé, count(*), MIN(date_embauche),MAX(date_embauche)
FROM employé
GROUP BY position_employé
```

Position employé

```
-----
32      1      6/nov./1989 0:00      6/nov./1989 0:00
35      3      9/nov./1989 0:00      8/déc./1992 0:00
64      1      7/juil./1992 0:00      7/juil./1992 0:00
75      3      1/janv./1990 0:00      5/déc./1992 0:00
78      1      9/déc./1988 0:00      9/déc./1988 0:00
80      1      11/nov./1993 0:00      11/nov./1993 0:00
..etc... (31 ligne(s) affectée(s))
```

---

**Exercice 6 :**

```
SELECT id_titre, MAX(droits)
FROM droits_prévus
GROUP BY id_titre
```

```
id_titre
-----
BU1032  12
BU1111  24
BU2075  24
BU7832  24
MC2222  20
MC3021  24
..etc... (16 ligne(s) affectée(s))
```

---

**Exercice 7 :**

```
select ville, nom_éditeur, count(*)
from éditeurs
GROUP BY ville, nom_éditeur
HAVING ville LIKE '%L%' OR ville LIKE '%B%'
```

```
-----
ville      nom_éditeur      count(*)
-----
Annaba     New Moon Books   1
Constantine data Infosystems 1
Blida     Ramona, éditeur  1
(3 ligne(s) affectée(s))
```

---

**Exercice 8 :**

```
SELECT droits, id_titre, minimum, maximum
FROM droits_prévus
ORDER BY droits
```

```

droits      id_titre minimum      maximum
-----
10          BU1032    0          5000
10          PC1035    0          2000
10          BU2075    0          1000
10          PS2091    0          1000
...etc...

```

```

count
=====
16

```

```

droits      id_titre minimum      maximum
-----
12          BU1032    5001       50000
12          PC1035    2001       3000
12          BU2075    1001       3000
12          PS2091    1001       5000
..etc...

```

```

count
=====
16

```

..etc... (94 ligne(s) affectée(s))

### Exercice 9 :

```

SELECT nom_auteur, titre, prix
FROM auteurs a, titres t, titreauteur ta
WHERE (ville = 'Annaba') AND (a.id_auteur = ta.id_auteur) AND (ta.id_titre = t.id_titre)

```

```

nom_auteur      titre                                                                 prix
-----
Selmi           Guide des bases de données du gestionnaire pressé                    140,00
Layouni         Le stress en informatique n'est pas une fatalité !                   24,00
Senhadji        Toute la vérité sur les ordinateurs                                    136,00
Chettah         Phobie et passion informatique : éventail de comportements          147,00
(4 ligne(s) affectée(s))

```

### Exercice 10

```

SELECT nom_éditeur, nom_auteur, titre, qt
FROM auteurs a, titreauteur ta, titres t, éditeurs e, ventes v
WHERE (a.id_auteur = ta.id_auteur) AND (ta.id_titre = t.id_titre) AND (t.id_éditeur = e.id_éditeur)
AND (t.id_titre = v.id_titre)
ORDER BY nom_éditeur
COMPUTE SUM(qt) BY nom_éditeur

```

### Exercice 11

```

SELECT nom_auteur, SUM(qt)
FROM auteurs a, titreauteur ta, titres t, ventes v
WHERE (a.id_auteur = ta.id_auteur) AND (ta.id_titre = t.id_titre) AND (t.id_titre = v.id_titre)
GROUP BY nom_auteur
HAVING SUM(qt) > 20

```

### Exercice 12

```

SELECT nom_auteur, pn_auteur
FROM auteurs a
WHERE id_auteur IN (SELECT id_auteur FROM titreauteur) /* l'auteur doit avoir écrit au moins
un livre */
AND 100 = ALL (SELECT droits_pourcent
FROM titreauteur
WHERE a.id_auteur = id_auteur)
ORDER BY nom_auteur

```



<i>nom_auteur</i>	<i>pn_auteur</i>
selmi	zoubir
Senhadji	abesse
Chettah	mohamed
Chergrouche	Ali

(4 ligne(s) affectée(s))

---

**Exercice 13 :**

```
SELECT titre, prix
FROM titres
WHERE prix = (SELECT MAX (prix)
              FROM titres )
```

titre	prix
Est-ce vraiment convivial ?	156,00

(1 ligne(s) affectée(s))

---

**Exercice 14 :**

Afficher la liste des titres et le cumul de leurs ventes, tous magasins confondus, classés par ordre croissant des ventes

```
SELECT titre, somme = (SELECT SUM (qt)
                     FROM ventes v
                     WHERE t.id_titre = v.id_titre)
FROM titres t
ORDER BY somme
```

<i>Titre</i>	<i>somme</i>
<i>La psychologie des ordinateurs de cuisine</i>	(null)
<i>Guide des bonnes manières sur un réseau</i>	(null)
<i>Les festins de Parly 2</i>	10
<i>Guide des bases de données du gestionnaire pressé</i>	15
<i>Toute la vérité sur les ordinateurs</i>	15
<i>Privation durable d'informations : étude de quatre cas représentatifs</i>	15
<i>Phobie et passion informatique : éventail de comportements</i>	20
<i>Cinquante ans dans les cuisines de l'Elysée</i>	20
<i>La cuisine japonaise - la portée de tous</i>	20
<i>La cuisine - l'ordinateur : bilans clandestins</i>	25
<i>Vivre sans crainte</i>	25
<i>Equilibre émotionnel : un nouvel algorithme</i>	25
<i>Est-ce vraiment convivial ?</i>	30
<i>Le stress en informatique n'est pas une fatalité !</i>	35
<i>Les micro-ondes par gourmandise</i>	40
<i>Oignon, poireau et ail : les secrets de la cuisine méditerranéenne</i>	40
<i>Les secrets de la Silicon Valley</i>	50
<i>La colère : notre ennemie ?</i>	108

(18 ligne(s) affectée(s))

---

**Exercice 15 :**

Afficher le titre du livre le plus vendu de tous les magasins, et le nom de ce magasin

```
SELECT titre, nom_mag
FROM titres t, ventes v, magasins m
WHERE t.id_titre = v.id_titre
AND v.qt = ( SELECT MAX(qt)
            FROM ventes)
AND v.id_mag = m.id_mag
```

titre	nom_mag
-------	---------

**Exercice 16 :**

Rentrez vos noms, prénoms, dans la table auteurs, avec un identificateur qui n'existe pas déjà

```
insert auteurs (id_auteur, nom_auteur, pn_auteur, contrat)
values ('100-00-1020', 'Lécu', 'Régis', 1)
```

**Exercice 17 :**

Recopier toutes les caractéristiques d'un auteur en lui donnant un nouvel identificateur, et un nouveau nom

```
INSERT auteurs (id_auteur, nom_auteur, pn_auteur, téléphone, adresse, ville, pays, code_postal,
contrat)
```

```
SELECT '100-00-1200', 'toto', pn_auteur,téléphone, adresse, ville, pays, code_postal, contrat
FROM auteurs
WHERE nom_auteur = 'Layouni'
```

**Exercice 18 :**

Augmenter de 10% tous les prix des livres de l'éditeur « data Infosystems». Vérifier l'opération par une commande Select adéquate avant et après l'augmentation.

```
SELECT titre, prix
FROM titres t, éditeurs e
WHERE t.id_éditeur = e.id_éditeur
AND nom_éditeur = "data Infosystems"
AND prix is not NULL
```

```
UPDATE titres
SET prix = 1.10 * prix
WHERE prix is not NULL
AND id_éditeur = (SELECT id_éditeur
FROM éditeurs
WHERE nom_éditeur = "data Infosystems")
```

```
SELECT titre, prix
FROM titres t, éditeurs e
WHERE t.id_éditeur = e.id_éditeur
AND nom_éditeur = "data Infosystems"
AND prix is not NULL
```

**Exercice 19 :** Détruire les lignes créées dans la tables auteur, dans les exercices 16 et 17

```
DELETE auteurs
FROM auteurs
where nom_auteur ='Layouni'
```

## EVALUATION (Solution donnée sur la plateforme E-learning)

La compagnie NUMIDIA Travel de transport envisage d'intégrer dans son site web la gestion des réservations et paiement en ligne des voyages planifiés. A cet effet, le directeur de la compagnie veut implanter au préalable la nouvelle gestion des réservations et paiement en ligne sous Access et par la suite la migrer vers SQL Server.

Le MLD relationnel de la base de données se présente ainsi :

**Client** (*CINClient*, NomClient, PrénomClient, TelClien)t

**Billet** (*NumBillet*, CINClient, CodeVoyage, DateBillet, NumPlace, Reéglé, EtatPlace)

**Voyage** (*CodeVoyage*, HeureDepartVoyage, HeureArrivéVoyage, VilleArrivéVoyage, PrixVoyage)

**Place** (*NumPlace*)

### Structure de la base de données

**Table Client**

Nom de champ	Signification	Type	Taille/Format
<u><i>CINClient</i></u>	<u><i>CIN de Client</i></u>	Texte	10
NomClient	Nom de Client	Texte	20
PrénomClient	Prénom de Client	Texte	15
TelClien	Téléphone de Client	Texte	12

**Table Billet**

Nom de champ	Signification	Type	Taille/Format
<i>NumBillet</i>	Numéro de Billet	Numérique	Entier long
CINClient	CIN Client	Texte	10
CodeVoyage	Code Voyage	Numérique	Entier
DateBillet	Date de Billet	Date et heure	Abrégé
NumPlace	Numéro de Place	Numérique	Octet
Réglé	Reglé	Oui/Non	
EtatPlace	Etat de Place	Texte	7

**Table Voyage**

Nom de champ	Signification	Type	Taille/Format
<i>CodeVoyage</i>	<i>Code de Voyage</i>	Numerique	Entier
HeureDepartVoyage	Heure de Départ de Voyage	Date et heure	Heure Abrégé
Ville_Départ	Ville de départ de voyage	texte	15
HeureArrivéVoyage	Heure d'Arriver de Voyage	Date et heure	Heure Abrégé
VilleArrivéVoyage	Ville d'Arriver de Voyage	Texte	15
PrixVoyage	Prix de Voyage	Monetaire	Dh

**Table Place**

Nom de champ	Signification	Type	Taille/Format
NumPlace	Numéro de la place	Numérique	octet

**Travail demandé**

1. Pour chaque table ci-dessus, créer sa structure. Utiliser les mêmes clés primaires indiquées dans le MLD ci-dessus et créer les relations entrer ces tables
2. Saisie les enregistrements donnés dans l'annexe ci-dessus pour chaque table de la base donnée.
3. Créer une requête qui affiche les informations sur un client en introduisant son CIN et l'entreprise sous R\_CIN\_client
4. Créer une requête qui affiche les informations sur un voyage planifié en introduisant la ville de départ et la ville d'arriver puis l'enregistrer sous R\_Client\_CIN
5. Créer un formulaire base sur la requête R\_Client\_CIN et l'enregistrer sous F\_Client\_CIN
6. Créer une requête qui affiche les informations sur le billet + voyage et l'enregistrer sous R\_Réservation
7. Créer un formulaire base sur la requête R\_voyage et l'enregistrer sous F\_Voyage
8. Créer un formulaire base sur la table voyage (Ajouter/modifier/Supprimer) et l'enregistrer sous F\_Voyage
9. Créer un formulaire base sur la requête R\_Réservation et l'enregistrer sous F\_Réservation comme est montre dans la figure suivante :

### Réservation

<b><u>N° Billet</u></b>	<input type="text"/>	<b><u>Date</u></b>	<input type="text"/>
<b><u>CIN</u></b>	<input type="text"/>		
<b><u>Code Voyage</u></b>	<input type="text"/>		
<b><u>Heure Départ</u></b>	<input type="text"/>	<b><u>Heure Arrivé</u></b>	<input type="text"/>
<b><u>Ville Départ</u></b>	<input type="text"/>	<b><u>Ville Arrivé</u></b>	<input type="text"/>
<b><u>Place</u></b>	<input type="text"/>	<b><u>Etat</u></b>	<input type="text"/>
<b><u>Prix</u></b>	<input type="text"/>	<b><u>Payé</u></b>	<input type="text"/>

- Le bouton affiche les information sur un voyage a choisir et remplir les champ de voyage dans ce formulaire
- Le bouton nouvelle réservation vide tous le champs
- Le bouton billet imprime le billet de la place réservé
- Le bouton menu retourne vers le formulaire ci-dessus

### Menu Général

10. Créer l'état qui fait sortir des billes payées par un client donné dans une date de voyage comme le montre la figure suivante et l'enregistrer sous E\_billets et le lier au bouton « Billet » du formulaire du Réservation. (6 Pts)

### *SARL NUMIDIATravel*

N° : 0012500254 Date :01/01/21

<b><u>Heure Départ</u></b>	<b><u>Ville Départ</u></b>	<b><u>Heure Arrivé</u></b>	<b><u>Ville Arrivé</u></b>	<b><u>Place</u></b>	<b><u>Prix</u></b>
4 :00	Annaba	10 :00	Alger	3	350.00

## Recueil d'exercices :

### 1- gestion des cours avec requêtes

Soit les tables suivantes:

Elève (elv\_id, elv\_nom, elv\_date\_de\_naiss, elv\_addr, elv\_num\_de\_phone)

Class (class\_id, class\_description, class\_section)

Matière (mat\_id, mat\_description, mat\_coefficient)

Test (elv\_id, mat\_id, date, note)

Les contraintes sont :

a- La date de naissance de l'élève doit être de la forme « 10 Jan 1990 »

b- La class description doit être de cette forme « BAC3, BAC2, BIT1, et PAC2... »

ou les trois 1<sup>er</sup> caractères sont alphabétiques et en majuscules, le dernier caractère est un nombre, toutes les caractères sont obligatoire .

c- La description de Matière prend une valeur de la liste suivante (math, Access, droit, économie, comptabilité), ou Access est la valeur défaut.

d- La note ne peut pas être inférieure à 1 ou supérieure à 20.

e- L'élève numéros de téléphone est de la forme (00961) 06/123456, ou le zip code n'est pas obligatoire à saisir.

f- Le nom de l'élève est en majuscule.

Questions

1- Définir les clés primaires de toutes les tables, en justifiant votre choix pour la 4eme table.

2- Ajouter le champ « class\_id » à une des tables pour que chaque élève possède une class.

3- Déterminer les relations existantes entre les tables et montrer leurs cardinalités.

4- Ecrire les propriétés suivantes (nom champ, type de données, propriétés de champ) pour les champs mentionnés dans les contraintes précédentes.

Utilise les tables et les relations de l'exercice 1, répondre au :

Requête 1: Affiche en ordre décroissant les noms des élèves qui habitent à « blida » Requête 2: Affiche les matières description, ou la matière coefficient est plus grande que 10. Requête 3: Affiche en ordre croissant la class description de chaque élève.

Requête 4: Affiche la note et test\_id de l'élève « Nabih ».

Requête 5: Affiche les noms des élèves de la classe « BAC3 » et « BAD3 ».

Requête 6: Affiche les notes et les noms des élèves de la class « BAD2 » pour la matière « Math » en 17/3/2008.

Requête 7: Affiche les noms des élèves qui se terminent avec la lettre « A ».

Requête 8: Affiche test\_id des tests qui sont faites avant 5/5/2007.

Requête 9: Affiche les noms des élèves qui ont réussi pour la classe « BAC2 » en 13/4/2008. N.B : l'élève réussi quand il a 1000 et plus comme note final.

**Pour chaque Requête indiquer les tables nécessaires avec les relations entre ces tables.**

**Exercice Access : Requêtes Calcule et affichage avec solutions**

Affichez la somme de la masse salariale de la compagnie divisée par lieu de travail et par poste.

\* Utilisation de l'opération somme sur plusieurs critères.

Affichez la masse salariale des employés embauchés en 1993 repartit par lieu de travail.

\* Utilisation de l'opération OÙ.

Affichez le nombre de personnes pour l'entreprise par lieu de travail et par occupation.

\* Utilisation de l'opération Compte.

a- Le piège de l'opération Compte.

Déterminez le nombre de vendeurs "champions" ayant récolté une commission de 45 000 DA ou plus.

\* Utilisation des opérations Compte et OÙ.

---

**Affichez la somme de la masse salariale de la compagnie divisée par lieu de travail et par poste.**

Il s'agit aussi d'une requête qui a besoin d'une opération (somme). Il faut ajouter encore plus de détails que les deux dernières requêtes. Dans ce cas, il faut ajouter les champs Bureau et poste en plus de calculer les revenus.

- Appuyez sur le bouton **S** .

**OU**

-Du menu **Affichage**, sélectionnez l'option **Opérations**.

**Critère**

Champ :	Bureau	Poste	Revenu : salaire + commission
Opération :	Regroupement	Regroupement	Somme
Tri:			
Afficher :	X	X	X
Critère :			
Où :			

**Résultat**

Bureau	Poste	Revenu
Annaba	Gérant	50 000 DA
Annaba	Vendeur	133 000 DA
Constantine	Gérant	43 000 DA
Constantine	Vendeur	169 000 DA

Plus que vous ajoutez de champs, plus que la réponse sera détaillée. À l'exercice 11, il y avait seulement un champ pour le total des revenus. À l'exercice 12, il y avait en plus du champ des revenus celui des postes. Le résultat de la requête est plus détaillé qu'auparavant. Avec cette question et trois champs à l'affichage, le résultat est encore plus détaillé.

**Affichez la masse salariale des employés embauchés en 2020 repartit par lieu de travail.**

Vous avez remarqué dans les trois derniers exercices qu'à chaque fois que vous ajoutez un champ à la requête que la réponse devient de plus en plus détaillée. Mais, que faire lorsqu'il faut faire une opération avec des critères sans détailler sur ceux-ci ? Il existe une opération pour régler cette situation. L'opération "Où" est utilisée lorsqu'on veut ajouter une condition à une requête sans pour autant détailler la réponse sur cette condition. Pour l'exemple, le champ Embauche est utilisé pour trouver les employés qui ont été embauchés en 1993. Cependant, le revenu n'est pas distribué pour chacune des dates d'embauche de cette année.

**Critère**

Champ :	Embauche	Bureau	Revenu : salaire + commission
Opération :	Où	Regroupement	Somme
Tri:			
Afficher :		X	X
Critère :	Entre #93-01-01# et #93-12-31#		
Ou :			

**Résultat**

Bureau	Revenus
Annaba	88 000 DA
Constantine	112 000 DA

Si vous n'aviez pas utilisé l'opération "Où", et laissé à l'opération regroupement pour le champ Embauche, le résultat aurait affiché le cumulatif des revenus par bureau et par date d'embauche des employés de la compagnie.

**Affichez le nombre de personnes pour l'entreprise par lieu de travail et par occupation.**

Cet exercice est pour vous démontrer le fonctionnement de l'opération Compte. Cette opération, comme le nom l'indique, compte le nombre d'enregistrements qui répondent aux critères demandés. Aussi, comme pour toutes les opérations, à chaque fois que vous ajoutez un champ à la requête, la réponse devient de plus en plus détaillée.

- Appuyez sur le bouton **S** . **OU**-Du menu **Affichage**, sélectionnez l'option **Opérations**.

**Critère**

Champ :	Bureau	Poste	Prénom
Opération :	Regroupement	Regroupement	Compte
Tri:	Croissant	Croissant	
Afficher :	X	X	X
Critère :			
Ou :			

**Résultat**

Bureau	Poste	CompteDePrénom
Annaba	Gérant	1
Constantine	Vendeur	3
Annaba	Gérant	1



Constantine	Vendeur	2
-------------	---------	---

### Le piège de l'opération Compte

L'opération Compte calcule le nombre d'enregistrements qui répond aux critères que vous avez choisis. Il y a cependant une situation où celle-ci pourrait sous-estimer le nombre exact. Reprenons le dernier exemple avec une petite modification. Au lieu de faire le compte sur le champ **Prénom**, utilisez le champ **Commentaire**.

Champ :	Bureau	Poste	Commentaire
Opération :	Regroupement	Regroupement	Compte
Tri:	Croissant	Croissant	
Afficher :	X	X	X
Critère :			
Ou :			

### Résultat

Bureau	Poste	CompteDeCommentaire
Annaba	Gérant	0
Annaba	Vendeur	3
Constantine	Gérant	0
Constantine	Vendeur	2

Il n'y a plus de gérants! Pourquoi y a-t-il une différence? Access n'ajoute pas les enregistrements dont le contenu du champ est vide. Comme vous l'avez vu à l'exercice 7, le contenu du champ commentaire est vide pour les gérants de l'entreprise. Il est donc fortement suggéré de toujours utiliser avec l'opération Compte un champ dont on est certain qu'il y a un contenu. Vous pouvez l'un des autres champs de la requête ou le champ qui sert de clé primaire de la table. Par définition, celui-ci ne peut jamais être vide.

### Déterminez le nombre de vendeurs "champions" ayant récolté une commission de 45 000 DA ou plus.

L'exercice consiste à utiliser l'opération Compte pour connaître le nombre de personnes qui répondent aux critères mentionnés ci-dessus. Cet exercice ressemble beaucoup à l'exercice précédent. La différence est que cet exercice utilise l'opération Compte au lieu de l'opération Somme. Puisque les gérants de l'entreprise n'ont pas de commissions, le champ Poste est inutile pour compter seulement les vendeurs.

- Appuyez sur le bouton **S** . **OU** du menu **Affichage**, sélectionnez l'option **Opérations**. **Critère**

Champ :	Nombre:Prénom	Commission
Opération :	Compte	Où
Tri:		
Afficher :	X	
Critère :		>=45000
Ou :		

## Résultat

Nombre
3

## Exercice Access : Requêtes avec Fonction avec solutions

1- Affichez combien d'années de service a chaque employé de l'entreprise.

\* Fonction année() et date().

2- Affichez à côté du prénom et du nom des employés de l'entreprise le texte "champion" pour ceux qui ont une commission de 45 000 DA ou plus. Placez le texte "Désolé" à côté de ceux qui ne répondent pas au critère précédent.

\* Fonction VraiFaux(condition; si vrai; si faux) dont le résultat est du texte.

3- Affichez la commission des vendeurs si on donnait une prime de 5 000 DA à ceux ayant une commission de 45 000 DA ou plus.

\* Fonction VraiFaux(condition; si vrai; si faux) dont le résultat est un chiffre.

4- Affichez la somme de la masse salariale de la compagnie.

\* Utilisation de l'opération somme.

5- Affichez la somme de la masse salariale de la compagnie divisée par occupation.

\* Utilisation de l'opération somme sur plusieurs critères.

---

### 1- Affichez combien d'années de service a chaque employé de l'entreprise.

Cet exercice démontre un peu mieux l'avantage des fonctions dans des champs calculés. Elle démontre aussi la fonction qui est la date de l'ordinateur sur lequel vous travaillez. Il devient donc facile, avec la fonction année(), de faire une soustraction pour calculer le nombre d'années de service. Le résultat suivant est correct si la requête a été faite en l'an 2020.

#### Critère

Champ :	Prénom	Nom	Service: Année(date())-année([embauche])
Tri :			
Afficher :	X	X	X
Critère :			
Ou :			

#### Résultat

Prénom	Nom	Service
Anis	Layouni	10
Dounia	Layachi	10
Salim	Riad	9
halim	Bennacer	9
ilhem	houda	9
salhi	Lechheb	9
bilel	Gedyira	9

Il y a aussi d'autres fonctions du groupe Date/Heure qui pourrait vous intéresser. Il y a la fonction **maintenant()** qui donne non seulement la date mais aussi l'heure de l'ordinateur. Donc, au lieu d'avoir juste le 98-01-01 avec la fonction **date()**, il est possible d'avoir avec la fonction **maintenant()** 98-01-01 12:00:00 . Il y d'autres fonctions tel que **jour()**, **joursem()**, **mois()** et plusieurs autres qui sont disponibles dans la catégorie Date/Heure. Il ne faut pas oublier qu'Access vous offre 176 fonctions distribuées dans 16 catégories.

**Note sur les champs de type Date/Heure.**

Bien qu'Access affiche l'information sous forme de date ou d'heure, les informations sont conservées dans le logiciel sous forme numérique. Par exemple, le chiffre 37222 équivaut au 27 novembre 2020. Donc 37222,5 équivaut à midi le 27 novembre 2020.

**2- Affichez à côté du prénom et du nom des employés de l'entreprise le texte "champion" pour ceux qui ont une commission de 45 000 DA ou plus. Placez le texte "Désolé" à côté de ceux qui ne répondent pas au critère précédent.**

Cet exercice est pour vous démontrer la fonction Vraifaux pour afficher du texte. Elle fonctionne sur le même principe que la fonction =Si d'Excel. Elle lui faut trois informations, ou trois "paramètres", pour fonctionner correctement: la condition, que faire si vrai et que faire si faux. Ces paramètres sont séparés par un point-virgule (;). On vous demande ici d'écrire le texte "Champion" si la personne a une commission de 45 000 DA ou plus. Sinon, il faut afficher le texte "Désolé".

**Critère**

Champ :	Prénom	Nom	Critère:vraifaux([commission]>= 45000;"Champion";"Désolé")
Tri:			
Afficher :	X	X	X
Critère :			
Ou :			

**Résultat**

Prénom	Nom	Critère
Salim	Yazid	Désolé
ilhem	Layachi	Champion
Selmi	Reda	Champion
Iyes	fatmi	Désolé
Riad	Chami	Désolé
madoui	Latreche	Champion
baali	Mohammed yazid	Désolé

Pour cet exercice, la fonction Vraifaux a été utilisée pour afficher du texte. Vous pouvez aussi l'utiliser pour aussi afficher des chiffres comme dans le prochain exercice.

**3- Affichez la commission des vendeurs si on donnait une prime de 5 000 DA à ceux ayant une commission de 45 000 DA ou plus.**

C'est dans ce genre de situation de la fonction vraifaux est vraiment avantageuse. Elle s'applique seulement si les conditions requises sont remplies. Dans l'exercice précédent, la fonction Vraifaux() était utilisée avec du texte. Ici la fonction est utilisée pour calculer un chiffre.

## Critère

Champ :	Prénom	Nom	Ajusté:vraifaux([commission]>= 45000;[commission]+5000;[commission])
Tri:			
Afficher :	X	X	X
Critère :			
Où :			

## Résultat

Prénom	Nom	Ajusté
Riad	Latreche	0
Dounia	Layachi	50 000
Selmi	Réda	70 000
ilhem	Houda	23 000
Riad	Dafri	0
ilhem	Laterche	52 000
salim	Yazid	22 000

Pouvez-vous préparer une requête qui démontrait la différence à la masse salariale avec et sans cette prime? Vous devriez être capable si vous avez bien compris les exercices précédents.

### 4- Affichez la somme de la masse salariale de la compagnie.

À date, les questions affichaient les informations selon les critères demandés. Mais que faire lorsqu'on a besoin de trouver la somme, la moyenne ou le nombre d'enregistrements qui répondent à certains critères? C'est pour ces circonstances qu'il y a les opérations.

Il y a une fonction très puissante qui n'a pas été couverte jusqu'à présent: les opérations de regroupement. Il est possible avec les opérations de regrouper les enregistrements pour pouvoir les additionner, les compter, trouver la moyenne, le plus petit, le plus grand et plusieurs autres. C'est aussi une opportunité de revoir les champs calculés pour déterminer la masse salariale de l'entreprise. On utilise encore une fois un champ calculé pour déterminer le revenu (salaire + commission) de chaque employé.

-De la barre d'outils, appuyez sur le bouton **S OU** du menu **Affichage**, sélectionnez l'option **Opérations**. Une nouvelle ligne s'est insérée en dessous de la ligne des champs. C'est la ligne des opérations. Access vous permet d'accomplir des opérations mathématiques sur les enregistrements d'une requête. Si vous regardez les opérations possibles, vous retrouverez parmi d'autres la somme, la moyenne, le plus petit (min), la plus grande (max) etc.

-Dans la première colonne, écrivez la formule suivante: **Revenu: [salaire] + [commission]** .

-De la ligne des opérations, sélectionnez l'opération de **Somme** pour le revenu.

#### Note:

Plusieurs personnes mélangent les opérations Somme et Compte. L'opération Somme est utilisée pour l'addition de chiffres de champs de type numérique ou de type monétaire. L'opération Compte est utilisée pour compter le nombre d'enregistrements qui répondent aux critères sélectionnés.

Il est aussi à noter qu'il est impossible de mettre un critère sous les opérations sauf pour les opérations Regroupement, Compte, Expression et Où.

**Critère**

Champ :	Revenu : [salaire] + [commission]
Opération:	Somme
Tri:	
Afficher :	X
Critère :	
Ou :	

**Résultat**

Revenu
295 000 DA

Il y a aussi une autre manière de réaliser la requête en utilisant la fonction somme. Voici à quoi elle ressemblerait.

**Critère**

Champ :	Revenu : somme([salaire] + [commission])
Tri:	
Afficher :	X
Critère :	
Ou :	

Dans ce cas, elle n'est pas vraiment plus pratique que l'opération somme. Elle est cependant plus pratique lorsqu'elle est utilisée dans un champ calculé avec d'autres fonctions. Par exemple:  $\text{somme}([\text{commission}] + [\text{salaire}]) / \text{compte}([\text{bureau}])$ . En passant, cette dernière formule est l'équivalent de  $\text{moyenne}([\text{salaire}] + [\text{commission}])$ .

**5- Affichez la somme de la masse salariale de la compagnie divisée par occupation.**

Il s'agit encore une fois d'une requête qui a besoin d'une opération. Il faut cependant un champ de plus pour avoir le détail nécessaire. Dans ce cas, il faut ajouter le champ bureau pour pouvoir répartir les revenus par poste.

- De la barre d'outils, appuyez sur le bouton **S .OU** du menu **Affichage**, sélectionnez l'option **Opérations**.

**Critère**

Champ :	Poste	Revenu : salaire + commission
Opération:	Regroupement	Somme
Tri:		
Afficher :	X	X
Critère :		
Ou :		

## Résultat

Poste	Revenu
Gérant	93 000 DA
Vendeur	202 000 DA

La masse salariale de l'entreprise est maintenant réparti selon le poste de travail: gérant ou vendeur. Pour l'exemple, il y a seulement deux postes. S'il y aurait eu plus que cela, chaque poste serait affiché avec son revenu équivalent.

## Requêtes simples d'affichage avec solutions

1- Affichez le prénom et le nom des personnes ayant pour prénom " RIAD " .

\* Recherche simple.

2- Affichez le prénom, nom de toutes les personnes dont le nom de famille commence par la lettre " L " .

\* Comme \* .

3- Affichez le prénom, nom et salaire des personnes ayant un salaire supérieur à 45 000 DA.

\* , =, <, >, =, <, = et type numérique .

4- Affichez le prénom et le nom des personnes embauchées en 2020.

\* Entre et type Date.

5- Affichez le prénom et le nom des employés ayant la permanence selon l'ordre alphabétique de nom de famille et de prénom.

\* type logique et tri des enregistrements.

6- Affichez le prénom et le nom des employés n'ayant pas la permanence.

\* type logique et l'opérateur Pas.

---

### 1- Affichez le prénom et le nom des personnes ayant pour prénom " RIAD " .

#### 1. Choisir la ou les tables et les requêtes nécessaires.

-Créer une nouvelle requête en utilisant le mode création.

-De la liste des tables et requêtes, sélectionnez la table **Employés**.

#### 2. Choisir le type de requête.

-Assurez-vous d'avoir la requête de type sélection .

#### 3. Choisir le ou les champs nécessaires.

-De la liste des champs, ajoutez le champ de type "nom" et le champ "prénom" à la liste des champs.

#### 4. Déterminer si les champs ont besoin d'être triés.

-Pour cet exercice, aucun tri de champ n'est nécessaire.

#### 5. Cacher les champs au besoin.

-Pour cet exercice, aucun champ n'a besoin d'être caché.

#### 6. Déterminer les critères de sélection.

-Écrivez **RIAD** dans la première ligne de critère sous le champ prénom.

Voici ce que devrait ressembler la partie des critères par après.

## Critères

Champ :	Prénom	Nom
Tri :		
Afficher :	X	X
Critère :	" RIAD"	
Ou :		

Vous n'êtes pas obligé de mettre les guillemets. Access va les mettre pour indiquer que le champ est de type "texte". De plus, vous n'êtes pas obligé de mettre l'opérateur " = ". Il est mis par défaut si aucun autre opérateur n'est choisi. En plus, vous n'êtes pas obligé de mettre le texte en majuscules ou en minuscules. Access ne fait pas la différence lors de la recherche. Il faut cependant écrire le critère correctement. Par exemple, vous ne trouverez personne si vous avez écrit "RIIIIIAD" .

Pour les champs de type Date/Heure, Access placera automatiquement devant et après la date le caractère " #". Donc, vous n'êtes pas obligé de le mettre. Tout cela est pour être capable de différencier les types de champs; "" pour texte, # pour Date/Heure et rien pour les chiffres (monétaire ou numérique).

### 7. Exécuter la requête.

-Appuyez sur le bouton !.

Voici le résultat.

## Résultat

Prénom	Nom
RIAD	CHAMI
RIAD	SALHI

### 2- Affichez le prénom, nom de toutes les personnes dont le nom de famille commence par la lettre " L ".

Cette requête sert à voir les caractères spéciaux tel que " \* " et "? ". Le caractère "?" est utile pour remplacer un caractère dans la requête. Par exemple, une recherche sur b?lle pourrait donner les résultats suivants: balle, belle, bille, bulle. En plus d'être utile pour les champs de type texte, il peut aussi être utilisé avec les champs de type Date/Heure. Par exemple, #98-??-01# pourrait afficher tous les enregistrements du premier jour de chaque mois. Le caractère "\*" sert à remplacer une série indéterminée de caractères. On connaît le début mais pas la fin. Par exemple, une recherche en utilisant bal\* pourrait donner le résultat suivant: balade, balai, balance, balcon, baleine, balise, balle, ballerine, ballet ...

-Pour le critère du champ "nom", écrivez seulement l\*.

Access se chargera d'écrire Comme pour indiquer que les enregistrements recherchés commencent par la lettre "l".

## Critères

Champ :	Prénom	Nom
Tri :		
Afficher :	X	X
Critère :		Comme l*
Ou :		

-Appuyez sur le bouton !.

### Résultat

Prénom	Nom
RIAD	LAYOUNI
TOUFIK	LAYACHI
ILHEM	LASKRI

### 3- Affichez le prénom, nom et salaire des personnes ayant un salaire supérieur à 45 000 DA.

Cette requête démontre les possibilités d'utiliser les opérateurs  $=$  pour les champs de type numérique ou monétaire. Pour cet exercice, le critère de recherche utilise un champ de type monétaire. Les autres types de champs peuvent aussi être utilisés avec ces opérateurs.

### Critères

Champ :	Prénom	Nom	Salaire
Tri :			
Afficher :	X	X	X
Critère :			>45000
Ou :			

### Résultat

Prénom	Nom	Salaire
Riad	Chami	50000

Faites attention en lisant la question. Est-ce qu'on vous demande supérieur à X ou égale et supérieur à X? C'est un petit piège que plusieurs ne font pas attention.

### 4- Affichez le prénom et le nom des personnes embauchées en 1993.

Cette requête pourrait utiliser les opérateurs  $\geq$ , et,  $\leq$ . Mais l'exercice va démontrer l'opérateur " Entre " et le type de champ Date/Heure. Remarquez les " # " devant et après la date. Si vous ne les mettez pas, Access va les écrire pour vous. C'est pour identifier que le champ Embauche est de type Date/Heure.

### Critères

Champ :	Prénom	Nom	Embauche
Tri :			
Afficher :	X	X	X
Critère :			Entre #93-01-01# et #93-12-31#
Ou :			



## Résultat

Prénom	Nom	Embauche
Selmi	Reda	93-01-01
ilhem	Laskri	93-06-06
Reda	Chami	93-06-06
ilhem	Layachi	93-06-01
Sassi	Gendelou	93-01-01

Dépendant des options sélectionnées dans Windows et Access, il se peut que le format de la date soit différent. Vous devrez probablement mettre le mois devant et jour et l'année. Faites quelques essais en changeant l'ordre de l'année, du mois et de la journée.

### 5- Affichez le prénom et le nom des employés ayant la permanence selon l'ordre alphabétique de nom de famille et de prénom.

Cette requête regarde les possibilités avec le champ de type logique Oui/Non et aussi de trier le résultat de la requête. De plus, on utilise le champ **Permanence** pour filtrer les enregistrements. Mais celui-ci n'est pas affiché lors de la présentation de l'information. Assurez-vous que la case Afficher soit désactivée pour ce champ.

#### Critère

Champ :	Nom	Prénom	Nom	Permanence
Tri :	Croissant	Croissant		
Afficher :		X	X	
Critère :				Oui
Ou :				

Rappelez-vous. La priorité des tris va au champ trié le plus à la gauche; même si celui-ci n'est pas affiché

## Résultat

Prénom	Nom
Selmi	Reda
ilhem	Laskri
Reda	Chami
ilhem	Layachi
Sassi	Gendelou

### 6- Affichez le prénom et le nom des employés n'ayant pas la permanence.

Il serait facile de simplement mettre le critère **non** sous le champ permanence. Mais cet exemple est surtout pour démontrer l'opérateur **Pas**. Celui-ci affiche toutes les informations **sauf** ceux que vous avez sélectionnés. Dans ce cas, il affichera tous ceux qui sont différents de oui. Donc, la seule possibilité est *non*.

## Critère

Champ :	Prénom	Nom	Permanence
Tri :			
Afficher :	X	X	
Critère :			Pas oui
Ou :			

## Résultat

Prénom	Nom
Selmi	Reda
ilhem	Laskri

Note: Il y a une caractéristique des champs de type Oui/Non que vous devriez savoir. Bien qu'il soit affiché Oui/Non, Vrai/Faux à l'écran, l'information est conservée dans la table sous forme numérique. Access inscrit 0 (zéro) lorsque faux et -1 lorsque vrai. Avec un peu d'imagination, vous pouvez vraiment prendre avantage de cette situation.

## Travaux Pratiques.

Un service financier réalise un audit de données bancaires. Le schéma relationnel de ces données est le suivant :

- **compte(idCompte, idClient, solde);**
- **client(idClient, nom, prenom, adresse, annee);**
- **action(idAction, idCompte, montant);**

### Travail à faire :

- 1- Définir les clés primaires.
- 2- Relier les tables et reproduire les tables de la base de données sur votre feuille de réponse.
- 3- Suite aux relations que vous avez réalisées :  
Créer une liste de choix pour les clés étrangères en complétant le tableau suivant :

<b>Nom du champ</b>	
<b>Afficher le contrôle</b>	
<b>Contenu</b>	
<b>Colonne liée</b>	
<b>Nbre de colonnes</b>	

compte			action		
idCompte	idClient	solde	idAction	idCompte	montant
1	30	5	1	1	+10
			2	1	-5
			3	1	-10

client				
idClient	nom	prenom	adresse	annee
30	Hamdi	Jalal	1 rue Pasteur	1978

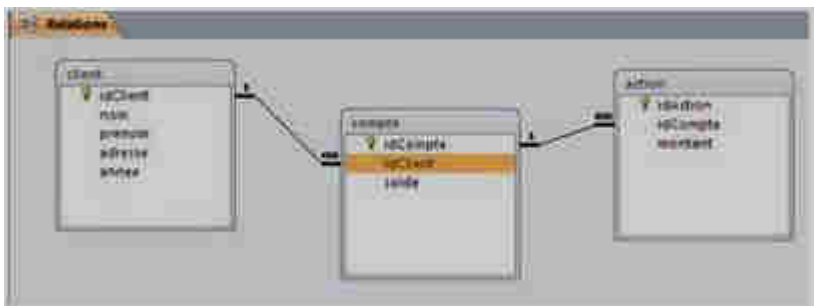
4-réaliser les requetes suivantes :

- a. Le nom et prénom des clients dont le numéro de client est inférieur strictement à 1000 et qui sont nés strictement après 1968;
- b. Les numéros de compte dont le solde est strictement supérieur à 1000 DA, et dont le propriétaire a pour nom 'Saadi' ;

1- Définir les clés primaires.

Table	Clé primaire
Client	idclient
Compte	idcompte
Action	idaction

2- Relier les tables et reproduire les tables de la base de données sur votre feuille de réponse.



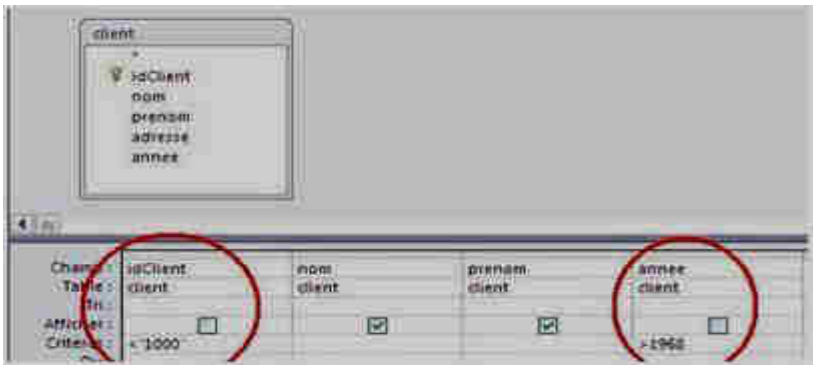
**3- Suite aux relations que vous avez réalisées :**

Créer une liste de choix pour les clés étrangères en complétant le tableau suivant :

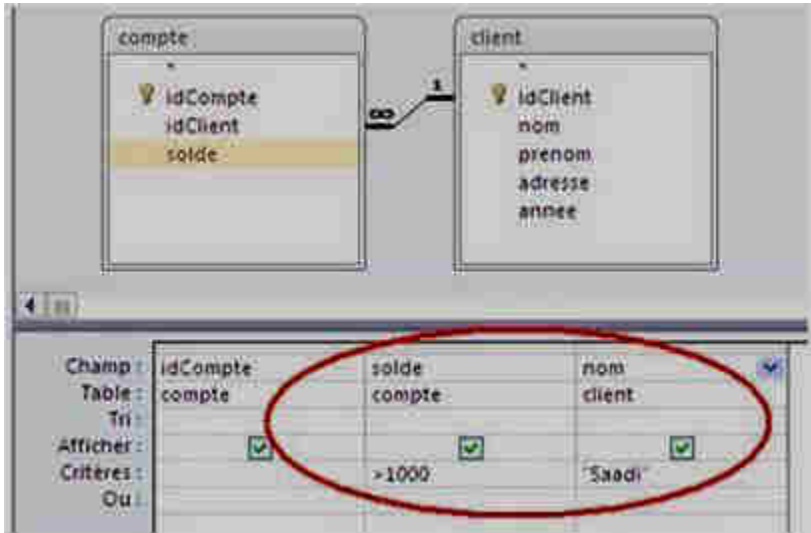
Nom du champ	Idclient
Afficher le contrôle	Zone de liste déroulante
Contenu	Client
Colonne liée	1
Nbre de colonnes	3

**4- Réaliser les requêtes suivantes en respectant le modèle ci-dessous :**

- a. Le nom et prénom des clients dont le numéro de client est inférieur strictement à 1000 et qui sont nés strictement après 1968;



b. Les numéros de compte dont le solde est strictement supérieur à 1000 Dirhams, et dont le propriétaire a pour nom 'Saadi' ;



## EXERCICES CORRIGÉS DE LANGAGE SQL

### Exercice 1 :

Soit la base de données d'un festival de musique : Dans une représentation peut participer un ou plusieurs musiciens. Un musicien ne peut participer qu'à une seule représentation.

Représentation (Num\_Rep , titre\_Rep , lieu)

Musicien (Num\_mus , nom , Num\_Rep #)

Programmer (Date , Num\_Rep # , tarif)

Ecrire la commande SQL permettant de rechercher :

La liste des titres des représentations.

La liste des titres des représentations ayant lieu au « théâtre Annaba ».

La liste des noms des musiciens et des titres et les titres des représentations auxquelles ils participent.

La liste des titres des représentations, les lieux et les tarifs du 25/07/2008.

Le nombre des musiciens qui participent à la représentation n°20.

Les représentations et leurs dates dont le tarif ne dépasse pas 20DH.

Après un certain nombre de représentation, le directeur du festival a constaté que certains musiciens participent à plusieurs représentations. Pourquoi la description proposée ne permet pas de traiter ce cas. Expliquer les opérations à faire pour résoudre ce problème. Traduire les étapes de la question en SQL afficher les listes des représentations du musicien numéro 128.

### Exercice 2 :

Soit la base de données suivante :

Départements:( DNO, DNOM, DIR, VILLE)

Employés: ( ENO, ENOM, PROF, DATEEMB, SAL, COMM, DNO#)

Exprimez en SQL les requêtes suivantes :

Donnez la liste des employés ayant une commission

Donnez les noms, emplois et salaires des employés par emploi croissant, et pour chaque emploi, par salaire décroissant

Donnez le salaire moyen des employés

Donnez le salaire moyen du département Production

Donnez les numéros de département et leur salaire maximum

Donnez les différentes professions et leur salaire moyen Donnez le salaire moyen par profession le plus bas

Donnez le ou les emplois ayant le salaire moyen le plus bas, ainsi que ce salaire moyen

### Exercice 3 :

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 97, dont une des étapes de type "contre la montre individuel" se déroule à Saint-Etienne :

EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif)

COUREUR(NuméroCoureur, NomCoureur, #CodeEquipe, #CodePays)

PAYS(CodePays, NomPays)

TYPE\_ETAPE(CodeType, LibelléType)

ETAPE(NuméroEtape, DateEtape, VilleDép, VilleArr, NbKm, #CodeType)

PARTICIPER(#NuméroCoureur, #NuméroEtape, TempsRéalisé)

ATTRIBUER\_BONIFICATION(#NuméroEtape, #NuméroCoureur, km, Rang, NbSecondes)

Exprimez en SQL les requêtes suivantes :

Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?

Quel est le nombre de kilomètres total du Tour de France 97 ?

Quel est le nombre de kilomètres total des étapes de type "Haute Montagne"?

Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

### Exercice 4 :

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

ETUDIANT(N°Etudiant, Nom, Prénom)

MATIERE(CodeMat, LibelléMat, CoeffMat)

EVALUER(#N°Etudiant, #CodeMat, Date, Note)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par Exprimez en SQL les requêtes suivantes :

Quel est le nombre total d'étudiants ?

Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?

Quelles sont les moyennes de chaque étudiant dans chacune des matières ?

Quelles sont les moyennes par matière ?

Quelle est la moyenne générale de chaque étudiant ?

Quelle est la moyenne générale de la promotion ?

Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion

Exercice 5 :

Ci-après, on donne la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre par, les étudiants peuvent ne pas suivre les mêmes matières.

ETUDIANT(CodeEt, NomEt, DatnEt)

MATIERE(CodeMat, NomMat, CoefMat)

ENSEIGNANT(CodeEns, NomEns, GradeEns, #CodeMat)

NOTE(#CodeEt, #CodeMat, note)

Ecrire les requêtes SQL permettant d'afficher :

Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom  
Les noms et les grades des enseignants de la matière dont le nom est 'BD'.

La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.

La liste des matières (Nom et Coefficient) qui sont suivies par l'étudiant de code 'Et321'.

Le nombre d'enseignants de la matière dont le nom est 'Informatique'

Exercice 6 :

Soit la base de données intitulée « gestion\_projet » permettant de gérer les projets relatifs au développement de logiciels. Elle est décrite par la représentation textuelle simplifiée suivante :

Developpeur (NumDev, NomDev, AdrDev, EmailDev, TelDev)

Projet (NumProj, TitreProj, DateDeb, DateFin)

Logiciel (CodLog, NomLog, PrixLog, #NumProj)

Realisation (#NumProj, #NumDev)

Ecrire les requêtes SQL permettant :

D'afficher les noms et les prix des logiciels appartenant au projet ayant comme titre « gestion de stock », triés dans l'ordre décroissant des prix

D'afficher le total des prix des logiciels du projet numéro 10. Lors de l'affichage, le titre de la colonne sera « cours total du projet ».

Afficher le nombre de développeurs qui ont participé au projet intitulé « gestion de stock »

Afficher les projets qui ont plus que 5 logiciels

Les numéros et noms des développeurs qui ont participé dans tous les projets.

Les numéros de projets dans lesquelles tous les développeurs y participent dans sa réalisation.

Exercice 7 :

On considère la base de données BD\_AIR\_MAROC suivante :

PILOTE (NUMPIL, NOMPIL, VILLE, SALAIRE)

AVION (NUMAV, NOMAV, CAPACITE, VILLE)

VOL (NUMVOL, #NUMPIL, #NUMAV, VILLE\_DEP, VILLE\_ARR, H\_DEP, H\_ARR)

Donnez la liste des avions dont la capacité est supérieure à 350 passagers.

Quels sont les numéros et noms des avions localisés à Alger ?

Quels sont les numéros des pilotes en service et les villes de départ de leurs vols ?

Donnez toutes les informations sur les pilotes de la compagnie.

Quel est le nom des pilotes domiciliés à Annaba dont le salaire est supérieur à 20000 DA?

Quels sont les avions (numéro et nom) localisés à Alger ou dont la capacité est inférieure à 350 passagers ?

Quels sont les numéros des pilotes qui ne sont pas en service ?

Donnez le numéro des vols effectués au départ de Alger par des pilotes de Annaba ?

Quels sont les vols effectués par un avion qui n'est pas localisé à Alger ?

Quelles sont les villes desservies à partir de la ville d'arrivée d'un vol au départ de Guelma ?

**Exercice 8 :**

Soit le schéma relationnel suivant :

Departement (NomD, N\_Dep, Directeur)

Employe (Matricule, Nom, Prénom, DateNaissance, Adresse, Salaire, #N\_dep, superieur)

Projet (NomP, N\_pro, Lieu, #N\_Dep)

Travailleur (#Matricule, #N\_Proj, Heures)

L'attribut supérieur dans la relation Employe contient le matricule du supérieur direct de l'employé. Chaque employé appartient à un département et travaille sur zéro, un ou plusieurs projets. Chaque projet est rattaché à un département qui peut être différent de celui des employés travaillant sur ce projet. Exprimer en SQL les requêtes suivantes :

Date de naissance et l'adresse de Tahar Layouni.

Nom et adresse des employés qui travaillent au département de recherche.

Nom et Prénom des employés dont le supérieur est Tahar Layouni.

Nom des employés qui travaillent plus de 10heures sur un projet à Guelmim

Nom des projets sur lesquelles travaillent Tahar Layouni et Dounia Mahmoudi.

Nom et prénom des employés qui ne travaillent sur aucun projet.

Numéro des projets qui ont au moins un participant de chaque département.

Nom des employés qui ne travaillent pas sur un projet à Guelmim.

**Exercice 9 :**

Soit le schéma relationnel suivant qui représente la base de données d'une agence de voyage en ligne.

CLIENT (NumCli, Nom, Prénom, e-mail, NumCB )

VOYAGE (CodeVoyage, Destination, Durée, Prix )

RESERVATION (#NumCli, #CodeVoyage, DateRes )

Formuler en SQL les requêtes suivantes :

Nom, prénom et e-mail des clients ayant une réservation en cours

Nom, prénom et e-mail des clients n'ayant aucune réservation en cours

Destination et liste des clients ayant réservés pour un voyage de plus de 10 jours et coûtant moins de 1000 DA.

Numéros de tous les clients ayant réservés sur tous les voyages proposés.

**Exercice 10 :**

Soit la base de données « cinéma » dont le schéma relationnel est donné ci-dessous :

VILLE (CodePostal, NomVille )

CINEMA (NumCine, NomCine, Adresse, #CodePostal )

SALLE (NumSalle, Capacité, #NumCine )

FILM (NumExploit, Titre, Durée)

PROJECTION (#NumExploit, #NumSalle, NumSemaine, Nbentrees)

Ecrivez les requêtes suivantes en SQL :

Titre des films dont la durée est supérieure ou égale à deux heures

Nom des villes abritant un cinéma nommé « MGM »

Nom des cinémas situés à Meknès ou contenant au moins une salle de plus 100 places

Nom, adresse et ville des cinémas dans lesquels on joue le film « Hypnose » la semaine 18

Numéro d'exploitation des films projetés dans toutes les salles

Titre des films qui n'ont pas été projetés

**Exercice 11 :**

À partir du système d'information de l'entreprise. le service des ressources humaines peut extraire et analyser les informations relatives à tous les personnels. Celui-ci lui permet en particulier d'exercer un suivi dans le domaine de la formation. Un extrait de ce domaine est présenté sous forme d'un schéma relation :

**Extrait du modèle relationnel du système d'information du service des ressources humaines**

ORGFORM (Numorgform, Nomorgform, Adresseorgform, CPorgform, Villeorgform, Nomorgform, Telorgform, Melorgform)  
 DATEFORM (Datedebutform)  
 FORMATION (Codeform, intituleform, Dureeform, Niveauform, Nbreplaceform, Coutform)  
 SALARIE (Matriculesal, Nomsal, Prenomsal, Qualificational, #Codecategorie)  
 CATEGORIE (Codecategorie, Libellecategorie)  
 SERVICE (Codeserv, Nomserv, Ancienneteserv)  
 PASSERBILAN (#Matriculesal, #Matriculesal, Datebilan)  
 AFFECTER (#Matriculesal, #Codeserv)  
 PROPOSER (#Codeform, #Matriculesal, Dateproposition)  
 SUIVRE (#Codeform, #Matriculesal, Datedebut)  
 REALISER (#Numorgform, #Codeform, #Datedebutform, NbInscrits)

Construire les requêtes en langage SQL permettant de répondre aux questions suivantes :  
 quel est le nombre de formations suivies par catégories de salariés ayant débuté au cours de la période du 01/06/2011 au 31/12/2011 ?

quelles sont les catégories pour lesquelles le nombre d'heures de formation est supérieur à la moyenne du nombre d'heures des formations suivies par l'ensemble des personnels ?

le responsable des ressources humaines souhaite intégrer dans la base de données une nouvelle formation liée au sertissage des boîtes de conserve. les nouvelles données à insérer sont les suivantes : "FORM587, sertissage niveau 1, 25j, perfectionnement, 12, 525 " Ecrire la requête permettant de mettre à jour la base.

**Exercice 12**

La société X utilise le logiciel de gestion de base de données Access pour gérer ses clients et ses représentants.

Voici la liste des tables créées dans Access :

Table représentants :

NUM REP	NOM REP	ADR REP	CP REP	VIL REP	AGE REP
1	TAYAR	Rue bel vue	23000	Annaba	22
2	BELAHCEN	Rue tiers	24000	Guelma	27
3	TOUMI	Rue du marché	16000	Alger	23
4	RAHEB	Rue des écoles	25000	Constantine	19
5	MAOUI	Rue malek	31000	Oran	30
6	SAKER	Rue djaout	17000	Tlemcen	40

**Table Courir**

NUM REP	COD REP
1	23000
1	24000
2	25000
2	26000
3	27000
3	28000
4	30000
5	31000
5	32000
6	33000
6	34000

**Table département :**

COD DEP	NOM DEP	CHEF SECTEUR
23000	ANNABA	TALEB
25000	CONSTANTINE	SAYEH
31000	ORAN	LAID

**Table client :**

CODE CLT	NOM CLT	NUM REP	NUM CAT
125444	SAHLI	TIYAR	2222



25487	RIHEB	TOUMI	3333
-------	-------	-------	------

Table categorie tarif.

NUM CAT	NOM CAT	REMISE
2222	Entreprise	10%
3333	Particulier	5%

Ecrire les requêtes suivantes :

- Afficher la liste des clients appartenant à la catégorie tarifaire n°1, classée par ordre alphabétique.
- Afficher la liste des clients (code, nom de client) rattachés au représentant **Toumi** .
- Afficher la liste des clients bénéficiant d'une remise de 10%.
- Afficher la liste des représentants (Numéro et nom) dépendant du chef de secteur .
- Afficher la liste des départements (code, nom, chef de secteur).
- Afficher la liste des chefs de secteur

### Exercice 13 :

Le responsable du SAV d'une entreprise d'électroménager a mis en place une petite base de données afin de gérer les interventions de ces techniciens. Le modèle relationnel à la source de cette base de données est le suivant :

**Client (Codecl, nomcl, prenomcl, adresse, cp, ville)**

**Produit (Référence, désignation, prix)**

**Techniciens (Codetec, nomtec, prenomtec, tauxhoraire)**

**Intervention (Numéro, date, raison, #codecl, #référence, #codetec)**

Le responsable vous demande d'écrire en langage SQL les requêtes suivantes :

La liste des produits (référence et désignation) classées du moins cher au plus cher.

Le nombre d'intervention du technicien n°2381.

La liste des clients ayant demandé une intervention pour des produits d'un prix supérieur à 300 DA.

Les interventions effectuées par le technicien : 'Mentri Mohamed' entre le 1er et le 31 août 2009.

Par ailleurs il vous informe que le produit référencé 548G a vu son prix augmenter (nouveau prix = 320 DA). Vous apprenez également par le directeur des ressources humaines qu'un nouveau technicien a été recruté : son code est le 3294, il s'appelle 'El hadi Ridha' et est rémunéré à un taux horaire de 15 DA.

### Exercice 14 :

La représentation textuelle suivante est une description simplifiée d'une base de données de gestion de facturation d'une entreprise commerciale.

**Client (Numcli, Nomcli, Prenomcli, adressecli, mailcli)**

**Produit (Numprod, désignation, prix, qte\_stock)**

**Vendeur (Idvendeur, Nomvendeur, adresse\_vend)**

**Commande (Numcom, #Numcli, #Idvendeur, #Numprod, date\_com, qte\_com)**

On suppose que **Numcli, Numprod, Idvendeur et Numcom** sont de type numérique. Le nom, le prénom et l'adresse des clients ainsi que les vendeurs sont des informations obligatoires, le mail peut ne pas être indiqué. La valeur par défaut de la quantité en stock des produits (qte\_stock) est égale à 0 Exprimer en SQL les requêtes suivantes :

Créer les tables : **Client, Produit, Vendeur et Commande.**

la liste des clients de Alger.

la liste des produits (Numprod, désignation, prix) classés de plus cher au moins cher.

noms et adresses des vendeurs dont le nom commence par la lettre 'M'.

la liste des commandes effectuées par le vendeur "Mohammed" entre le 1er et 30 janvier 2012.

le nombre des commandes contenant le produit n° 365.

Exercice 15 :

Soit la base de données suivante :

<i>Etudiant</i>				
numéro_carte_etudiant	Nom	Prénom	Date_naissance	Section
01234567	Ben Salah	Ahmed	12/08/1988	Informatique
01234568	Ben Mahmoud	Sami	02/09/1990	Math
01234569	Marzougui	Rami	23/01/1988	Informatique

<i>Matière</i>		
code_matière	nom_matière	coefficient
12508	Base de données	1.5
12518	Algorithmique	3

<i>Note</i>		
numéro_carte_etudiant	code_matière	note_examen
01234567	12508	15.5
01234567	12518	5.5
01234568	12518	10.5
01234569	12518	8.75

Ecrire les commandes SQL permettant de rechercher :

La liste de tous les étudiants.

Nom et coefficient des matières.

Les numéros des cartes d'identité des étudiants dont la moyenne entre 7 et 12.4.

La liste des étudiants dont le nom commence par 'ben'. Le nombre des étudiants qui ont comme matière '12518'.

La somme des coefficients des matières.

Les noms des étudiants qui une note examen >10.

Afficher les noms et les coefficients des matières étudiées par l'étudiant "01234568".

#### Exercice 16 :

Afin d'assurer la qualité des produits attendues par les Clients, l'entreprise cherche à optimiser la gestion des pannes pouvant survenir dans les infrastructures de production nécessaires à la fabrication du Ciment. Voici un extrait de la base de données :

**TECHNICIEN (idTech, nom, prénom, spécialité)**

**STATION (idstat, nom, Position, coordLat, coordLong, phase)**

**MACHINE (idmach, état, dateMiseEnService, dateDernièreRévision, #idStat)**

**TYPEINCIDENT (id, description, tempsRéparationPrévu)**

**INCIDENT (idInd, remarques, dateHeure, dateHeureCloture, #idmach, #idType)**

**INTERVENTION (idInterv, dateHeureDébut, dateHeureFin, #idInd, #idTech)**

Rédiger la requête SQL permettant d'obtenir la liste par ordre alphabétique des noms et prénoms des techniciens ayant réalisé une intervention sur la Machine identifiée par Ber001.

Rédiger la requête SQL permettant d'obtenir la liste des phases ayant connue un incident de "sur-chauffage" pour le mois Mai 2013.

Rédiger la requête SQL permettant d'obtenir le nombre d'incidents non clôturés.

Rédiger la requête SQL permettant d'obtenir la liste des noms des stations ayant eu plus de dix incidents.

Exercice 17 :

Soit la base de données gestion des ventes :

Produit (Ref, Designation, PrixUnitaire, Dimension, #code\_Machine)

Vente (Ncom, Ref, Qte, DateLiv)

Commande (Ncom, DateCmd, #CodeClt, #Code\_Salarie)

Produit\_concurrent(Ref, Designation, PrixUnitaire, PrixUnitaire, Dimension, #code\_Machine, Nom\_Concurrent)

Donner la requête qui permet d'obtenir le chiffre d'affaire mensuel de l'année en cours.

Donner la requête qui calcule le taux de vente de chaque produit.

Donner la requête qui affiche le produit le plus vendu du mois en cour.

La table produit concurrent est composée des informations sur les produits vedettes des concurrents ; Donner la requête qui permet d'ajouter tous les produits du concurrent « Alu » à la table Produits.

# Exercices Corrigés

## Exercice 1

Soit la base de données d'un festival de musique : Dans une représentation peut participer un ou plusieurs musiciens. Un musicien ne peut participer qu'à une seule représentation.

- Représentation (**Num\_Rep** , titre\_Rep , lieu)
- Musicien (**Num\_mus** , nom , #**Num\_Rep**)
- Programmer (Date, #**Num\_rep**, tarif)
- 

Ecrire la commande SQL permettant de rechercher :

1. La liste des titres des représentations.

```
1 SELECT * FROM Représentation
```

2. La liste des titres des représentations ayant lieu au « théâtre annaba ».

```
1 SELECT * FROM Représentation WHERE lieu="theatre_annaba"
```

3. La liste des noms des musiciens et des titres et les titres des représentations auxquelles ils participent.

```
1 SELECT M.nom, R.titre FROM Musicien M INNER JOIN Représentation R ON R.  
2 Num_rep=M.Num_rep
```

4. La liste des titres des représentations, les lieux et les tarifs du 25/07/2008.

```
1 SELECT R.titre, R.lieu, P.tarif FROM Programmer P INNER JOIN Représentation R ON P.  
2 Num_rep=R.Num_rep WHERE P.date="25-07-2008"
```

5. Le nombre des musiciens qui participent à la représentation n°20.

```
1 SELECT COUNT(*) FROM Musicien WHERE Num_rep=20
```

6. Les représentations et leurs dates dont le tarif ne dépasse pas 20DH.

```
1 SELECT R.Num_Rep, R.titre, P.Date FROM Représentation R  
2 INNER JOIN Programmer P ON R.Num_Rep=P.Num_Rep WHERE P.tarif<=20
```

## Exercice 2

Soit la base de données suivante :

- Départements : ( **DNO**, DNOM, DIR, VILLE)
  - Employés : ( **ENO**, ENOM, PROF, DATEEMB, SAL, COMM, #**DNO**)
- Exprimez en SQL les requêtes suivantes :

1. Donnez la liste des employés ayant une commission

```
1 SELECT * FROM Employés WHERE COMM NOT NULL
```

2. Donnez les noms, emplois et salaires des employés par emploi croissant, et pour chaque emploi, par salaire décroissant

```
1 SELECT ENOM, PROF, SAL FROM Employés ORDER BY PROF ASC, SAL DESC
```

3. Donnez le salaire moyen des employés

```
1 SELECT AVG(SAL) FROM Employés
```

4. Donnez le salaire moyen du département Production

```
1 SELECT AVG(E.SAL) FROM Employés E INNER JOIN Département D ON E.  
DNO=D.DNO WHERE D.DNOM="production"
```

2

5. Donnez les numéros de département et leur salaire maximum

```
1 SELECT DNO, MAX(SAL) FROM EmpIoyes GROUP BY DNO
```

6. Donnez les différentes professions et leur salaire moyen

```
1 SELECT PROF, MAX(SAL) FROM EmpIoyes GROUP BY PROF
```

7. Donnez le salaire moyen par profession le plus bas

```
1 SELECT AVG(SAL) as moy FROM EmpIoyes GROUP BY PROF ORDER BY  
2 moy ASC LIMIT 1
```

8. Donnez le ou les emplois ayant le salaire moyen le plus bas, ainsi que ce salaire moyen

```
1 SELECT PROF FROM EmpIoyes GROUP BY PROF  
2 HAVING AVG(SAL)=(SELECT AVG(SAL) as moy FROM EmpIoyes GROUP  
3 BY PROF ORDER BY moy ASC LIMIT 1)
```

## Exercice 3

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

- ETUDIANT(**NEtudiant**, Nom, Prénom)
- MATIERE(**CodeMat**, LibelléMat, CoeffMat)
- EVALUER(**#NEtudiant**, **#CodeMat**, Date, Note)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par # Exprimez en SQL les requêtes suivantes :

1. Quel est le nombre total d'étudiants ? `SELECT count(*) FROM ETUDIANT`
2. Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ? `SELECT MIN(Note) as 'plusbasse note', MAX(Note) as 'plus haute note' FROM EVALUER`
3. Quelles sont les moyennes de chaque étudiant dans chacune des matières ? `SELECT E.NEtudiant, E.Nom, E.Prénom, M.LibelléMat, M.CoeffMat, AVG(EV.Note) AS MoyEtuMat FROM EVALUER EV, MATIERE M, ETUDIANT E WHERE EV.CodeMat = M.CodeMat AND EV.NEtudiant = E.NEtudiant GROUP BY E.NEtudiant, M.LibelléMat`

4. Quelles sont les moyennes par matière ? Avec la vue MGETU de la question 3 ( MOYETUMAT)

```
1 SELECT LibelleMat, AVG(MoyEtuMat) FROM MOYETUMAT GROUP BY LibelleMat
```

5. Quelle est la moyenne générale de chaque étudiant ? Avec la vue MGETU de la question 3 ( MOYETUMAT)

```
1 SELECT NEtudiant, Nom, SUM(MoyEtuMat*CoeffMat)/SUM(CoeffMat) AS MgEtu FROM  
2 MOYETUMAT GROUP BY NEtudiant
```

6. Quelle est la moyenne générale de la promotion ? Avec la vue MGETU de la question 5 :

```
1 SELECT AVG(MgEtu) FROM MGETU
```

7. Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ? Avec la vue MGETU de la question 5

```
1 SELECT NEtudiant, Nom, Prenom, MgEtu FROM MGETU WHERE  
2 MgEtu >= (SELECT AVG(MgEtu) FROM MGETU)
```

## Exercice 4

Soit la base de données intitulée "gestion\_projet" permettant de gérer les projets relatifs au développement de logiciels.

Elle est décrite par la représentation textuelle simplifiée suivante :

- Developpeur (**NumDev**, NomDev, AdrDev, EmailDev, TelDev)
- Projet (**NumProj**, TitreProj, DateDeb, DateFin)

- Logiciel (**CodLog**, NomLog, PrixLog, **#NumProj**)
- Realisation (**#NumProj**, **#NumDev**) Ecrire les requêtes SQL permettant :

```
1 SELECT L.NomLog, L.PrixLog FROM Logiciel L INNER JOIN Projet P ON L.
2 NumProj=P.NumProj WHERE P.TitreProj="gestion_de_stock" ORDER BY L.
3 PrixLog DESC
```

2. D'afficher le total des prix des logiciels du projet numéro 10. Lors de l'affichage, le titre de la colonne sera « courstotal du projet ».

```
1 SELECT SUM(PrixLog) as "cout_total_du_projet" FROM Logiciel WHERE NumProj=10
```

3. Afficher le nombre de développeurs qui ont participé au projet intitulé « gestion de stock »

```
1 SELECT count(*) FROM Developpeur D INNER JOIN Realisation R
2 ON D.NumDev=R.NumDev INNER JOIN Projet P ON P.NumProj=R.NumProj
```

4. Afficher les projets qui ont plus que 5 logiciels SELECT NumProj, TitreProj FROM PProjet P INNER JOIN LogicielL ON P.NumProj=L.NumProj GROUP BY NumProj HAVING count(\*)>5

5. Les numéros et noms des développeurs qui ont participé dans tout les projets.

```
1 SELECT NumDev, NomDev FROM Developpeur D INNER JOIN Realisation R ON D.NumDev
2 =R.NumDev GROUP BY NumDev HAVING
3 count(*)=(SELECT COUNT(*) FROM Projet)
```

6. Les numéros de projets dans lesquelles tous les développeurs y participent dans sa réalisation.

```
1 SELECT NumProj, TitreProj FROM Projet P INNER JOIN Realisation R ON P.Num
2 Proj=R.NumProj GROUP BY NumProj HAVING
3 count(*)=(SELECT COUNT(*) FROM Developpeur)
```

## Exercice 5

Ci-après, on donne la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre par, les étudiants peuvent ne pas suivre les mêmes matières.

- ETUDIANT(**CodeEt**, NomEt, DatnEt)
- MATIERE(**CodeMat**, NomMat, CoefMat)
- ENSEIGNANT(**CodeEns**, NomEns, GradeEns, **#CodeMat**, **note**)
- NOTE(**#CodeEt**, **#CodeMat**, Ecrire les requêtes SQL permettant 'afficher :

1. Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique

```
SELECT * FROM ETUDIANT ORDER BY NomEt ASC
```

2. Les noms et les grades des enseignants de la matière dont le nom est 'BD'.

```
1 SELECT E.NomEns, E.GradeEns FROM ENSEIGNANT E INNER JOIN MATIERE M
2 ON M.CodeMat=E.CodeMat WHERE M.NomMat="BD"
```

3. La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.

```
1 SELECT DISTINCT(M.NomMat), M.CoeffMat FROM ENSEIGNANT E
2 INNER JOIN MATIERE M ON M.CodeMat=E.CodeMat WHERE E.GradeEns="Grd3"
```

4. La liste des matières (Nom et Coefficient) qui sont suivies par l'étudiant de code 'Et321'.

```
1 SELECT M.NomMat, M.CoeffMat FROM MATIERE M INNER JOIN NOTE N
2 ON M.CodeMat=N.CodeMat INNER JOIN ETUDIANT E ON E.CodeEt=N.CodeEt WHERE E.
3 CodeEt="Et321"
```

5. Le nombre d'enseignants de la matière dont le nom est 'Informatique' SELECT COUNT(\*) FROM ENSEIGNANTE INNER JOIN MATIERE M ON M.CodeMat=E.CodeMat WHERE

## Exercice 6

On considère la base de données BD\_AIR\_Algerie suivante :

- PILOTE (NUMPIL, NOMPIL, VILLE, SALAIRE)
- AVION (NUMAV, NOMAV, CAPACITE, VILLE)
- VOL (NUMVOL, #NUMPIL, #NUMAV, VILLE\_DEP, VILLE\_ARR, H\_DEP, H\_ARR)

1. Donnez la liste des avions dont la capacité est supérieure à 350 passagers.

```
1 SELECT * FROM AVION WHERE CAPACITE > 350
```

2. Quels sont les numéros et noms des avions localisés à Alger ?

```
1 SELECT NUMAV, NOMAV FROM AVION WHERE VILLE = 'Alger'
```

3. Quels sont les numéros des pilotes en service et les villes de départ de leurs vols ?

```
1 SELECT NUMPIL, VILLE_DEP FROM VOL
```

4. Donnez toutes les informations sur les pilotes de la compagnie.

```
1 SELECT * FROM PILOTE
```

5. Quel est le nom des pilotes domiciliés à Meknès dont le salaire est supérieur à 20000 DA ?

```
1 SELECT NOMPIL FROM PILOTE WHERE VILLE = 'Annaba' AND SALAIRE > 20000
```

6. Quels sont les avions (numéro et nom) localisés à Alger ou dont la capacité est inférieure à 350 passagers ?

```
1 SELECT NUMAV, NOMAV FROM AVION WHERE VILLE = 'Alger' AND CAPACITE < 350
```

7. Quels sont les numéros des pilotes qui ne sont pas en service ?

```
1 SELECT NUMPIL FROM PILOTE  
2 WHERE NUMPIL NOT IN (SELECT DISTINCT NUMPIL FROM VOL)
```

8. Donnez le numéro des vols effectués au départ de Alger par des pilotes de Annaba ?

```
1 SELECT DISTINCT V.NUMVOL FROM VOL AS V, PILOTE AS P  
2 WHERE V.NUMPIL = P.NUMPIL AND V.VILLE_DEP = 'Alger' AND P.VILLE = 'Annaba'
```

Ou

```
1 SELECT DISTINCT NUMVOL FROM VOL WHERE V.VILLE_DEP = 'Alger' AND NUMPIL NOT IN  
2 (SELECT NUMPIL FROM PILOTE WHERE VILLE = 'Annaba')
```

9. Quels sont les vols effectués par un avion qui n'est pas localisé à Alger ?

```
1 SELECT DISTINCT V.NUMVOL FROM VOL V, AVION A WHERE A.NUMAV = V.NUMAV  
2 AND A.VILLE != 'Alger'
```

10. Quelles sont les villes desservies à partir de la ville d'arrivée d'un vol au départ de Constantine ?

```
1 SELECT DISTINCT VILLE_ARR FROM VOL WHERE VILLE_DEP = 'Constantine' AND  
2 VILLE_DEP != VILLE_ARR
```

## Exercice 7

Soit le schéma relationnel suivant :

- Departement (NomD, **N\_Dep**, Directeur)
- Employe (**Matricule**, Nom, Prénom, DateNaissance, Adresse, Salaire, **#N\_dep**, superieur)
- Projet (NomP, **N\_pro**, Lieu, **#N\_Dep**)
- Travaille (**#Matricule**, **#N\_Proj**, Heures)

L'attribut supérieur dans la relation Employe contient le matricule du supérieur direct de l'employé. Chaque employé appartient à un département et travaille sur zéro, un ou plusieurs projets. Chaque projet est rattaché à un département qui peut être différent de celui des employés travaillant sur ce projet. Exprimer en SQL les requêtes suivantes :

1. Date de naissance et l'adresse de Tahar Layouni.

```
1 SELECT DateNaissance, Adresse FROM Employe WHERE Nom='Layouni' AND Prenom
2 = 'Tahar'
```

2. Nom et adresse des employés qui travaillent au département de recherche.

```
1 SELECT E.Nom, E.Adresse FROM Employe as E, Departement as D WHERE
2 E.N_dep=D.N_dep AND NomD='recherche'
```

3. Nom et Prénom des employés dont le supérieur est Taha Lamharchi.

```
1 SELECT Nom, Prenom FROM Employe
2 WHERE superieur=(SELECT Matricule FROM Employe WHERE Nom='Layouni' AND Prenom
3 = 'Tahar')
```

4. Nom des employés qui travaillent plus de 10 heures sur un projet à Constantine

```
1 SELECT E.Nom FROM Employe as E, Travaille as T, Projet P WHERE E.
2 Matricule=T.Matricule
3 AND T.N_proj=P.N_proj AND T.heures >=10 AND P.Lieu='Constantine'
```

5. Nom des projets sur lesquelles travaillent Taha Lamharchi et Dounia Mahmoud.

```
1 SELECT T.N_proj FROM Travaille as T, Employe as E WHERE T.Matricule=E.Matricule
2 AND E.Nom='Layouni' AND E.Prenom='Tahar'
3 INTERSECT
4 SELECT T.N_proj FROM Travaille as T, Employe as E WHERE T.Matricule=E.Matricule
5 AND E.Nom='Mahmoudi' AND E.Prenom='Dounia'
```

6. Nom et prénom des employés qui ne travaillent sur aucun projet.

```
1 SELECT Nom, Prenom FROM Employe
2 WHERE Matricule NOT IN ( SELECT Matricule FROM Travaille)
```

7. Numéro des projets qui ont au moins un participant de chaque département.

```
1 SELECT T.N_proj FROM Travaille as T, Projet as P, Employe as E WHERE
2 T.N_proj=P.N_proj AND T.Matricule=E.Matricule
3 GROUP BY T.N_proj
4 HAVING count(DISTINCT E.N_dep)=(SELECT count(*) FROM Departement)
```

8. Nom des employés qui ne travaillent pas sur un projet à Constantine.

```
1 SELECT Nom FROM Employe WHERE
2 Matricule NOT IN (SELECT T.Matricule FROM Travaille as T, Projet as P WHERE T.
3 N_proj=P.N_proj AND P.Lieu='Constantine')
```

## Exercice 8

Soit le schéma relationnel suivant qui représente la base de données d'une agence de voyage en ligne.

- CLIENT (**NumCli**, Nom, Prénom, e-mail, NumCB )
- VOYAGE (**CodeVoyage**, Destination, Durée, Prix )
- RESERVATION (**#Numcli**, **#CodeVoyage**, DateRes)

Formuler en SQL les requêtes suivantes :

1. Nom, prénom et e-mail des clients ayant une réservation en cours

```
1 SELECT Nom, Prénom, e-mail FROM CLIENT
2 WHERE NumCli IN (SELECT DISTINCT NumCli FROM RESERVATION)
```

2. Nom, prénom et e-mail des clients n'ayant aucune réservation en cours

```
1 SELECT Nom, Prénom, e-mail FROM CLIENT
2 WHERE NumCli NOT IN (SELECT DISTINCT NumCli FROM RESERVATION)
```

3. Destination et liste des clients ayant réservés pour un voyage de plus de 10 jours et coûtant moins de 1000 DA.

```
1 SELECT C.Nom, C.Prenom, V.Destination FROM CLIENT as C, VOYAGE as V, RESERVATION as R
2 WHERE C.NumCli=R.NumCli and V.CodeVoyage=R.CodeVoyage
3 AND Duree>=10 AND Prix<1000
4
```

4. Numéros de tous les clients ayant réservés sur tous les voyages proposés.

```
1 SELECT NumCli FROM RESERVATION GROUP BY NumCli HAVING
2 count(*)=(SELECT count(*) FROM VOYAGE)
```

## Exercice 9

Soit la base de données « cinéma » dont le schéma relationnel est donné ci-dessous :

- VILLE (CodePostal, NomVille )
- CINEMA (**NumCine**, NomCine, Adresse, **#CodePostal** )
- SALLE (**NumSalle**, Capacité, **#NumCine** )
- FILM (**NumExploit**, Titre, Durée)
- PROJECTION(**#NumExploit**, **#NumSalle**, NumSemaine, Nbentrees)

Ecrivez les requêtes suivantes en algèbre relationnelle :

1. Titre des films dont la durée est supérieure ou égale à deux heures

```
1 SELECT NumExploit, Titre FROM FILM WHERE Duree>=2
```

2. Nom des villes abritant un cinéma nommé « MGM »

```
1 SELECT NomVille FROM VILLE
2 WHERE CodePostal IN (SELECT CodePostal FROM CINEMA WHERE NomCine='MGM')
```

3. Nom des cinémas situés à Annaba ou contenant au moins une salle de plus 100 places

```
1 SELECT NumCine FROM CINEMA WHERE CodePostal=(SELECT CodePostal FROM
2 VILLE WHERE NomVille='Annaba')
3 OR NumCine IN (SELECT NumCine FROM SALLE WHERE Capacité>=100)
```



4. Nom, adresse et ville des cinémas dans lesquels on joue le film « Hypnose » la semaine 18

```
1 SELECT C.NomCine, C.Adresse, V.NomVille FROM CINEMA as C, VILLE as V WHERE C.
2 CodePostal=V.CodePostal
3 AND C.NumCine IN (SELECT S.NumCine FROM SALLE as S, FILM as F, PROJECTION as P
4 WHERE P.NumExploit=F.NumExploit AND P.NumSalle=S.NumSalle AND F.
5 Titre='Hypnose' AND P.NumSemaine=18)
```

5. Numéro d'exploitation des films projetés dans toutes les salles

```
1 SELECT NumExploit FROM PROJECTION GROUP BY NumExploit
2 HAVING count(*)=(SELECT count(*) FROM SALLE)
```

6. Titre des films qui n'ont pas été projetés

```
1 SELECT Titre FROM FILM WHERE NumExploit NOT IN (SELECT NumExploit FROM PROJECTION)
```

## Exercice 10

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 97, dont une des étapes de type "contre la montre individuel" se déroula à Saint-Etienne :

- EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif)
- COUREUR(NuméroCoureur, NomCoureur, **#CodeEquipe**, **#CodePays**)
- PAYS(CodePays, NomPays)
- TYPE\_ETAPE(CodeType, LibelleType)
- ETAPE(NuméroEtap, DateEtap, VilleDép, VilleArr, NbKm, #CodeType)
- PARTICIPER(**#NuméroCoureur**, **#NuméroEtap**, TempsRealisé)
- ATTRIBUER\_BONIFICATION(**#NuméroEtap**, **#NuméroCoureur**, km, Rang, NbSecondes)

Exprimez en SQL les requetes suivantes :

1. Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?

```
1 SELECT NumeroCoureur, NomCoureur, NomPays FROM EQUIPE A, COUREUR B, PAYS C
2 WHERE A.CodeEquipe=B.CodeEquipe And B.CodePays=C.CodePays
3 And NomEquipe="FESTINA"
```

2. Quel est le nombre de kilomètres total du Tour de France 97 ?

```
1 SELECT SUM(Nbkm) FROM ETAPE
```

3. Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?

```
1 SELECT SUM(Nbkm) FROM ETAPE A, TYPE_ETAPE B
2 WHERE A.CodeType=B.CodeType And LibelleType="HAUTE_MONTAGNE"
```

4. Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

```
1 SELECT NomCoureur FROM COUREUR
2 WHERE NumeroCoureur NOT IN (SELECT NumeroCoureur FROM ATTRIBUER_BONIFICATION)
```

5. Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

```
1 SELECT NomCoureur FROM PARTICIPER A, COUREUR B
2 WHERE A.NumeroCoureur=B.NumeroCoureur GROUP BY
3 NumeroCoureur, NomCoureur
4 HAVING COUNT(*)=(SELECT COUNT(*) FROM ETAPE)
```

6. Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

```

1 SELECT NomCoureur, CodeEquipe, CodePays, SUM(TempsRealise) AS Total FROM
2 PARTICIPER A, COUREUR B
3 WHERE A.NumeroCoureur=B.NumeroCoureur and NumeroEtape<=13 GROUP BY
4 A.NumeroCoureur, NomCoureur, CodeEquipe, CodePays ORDER BY Total
5

```

7. Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

```

1 SELECT NomEquipe, SUM(TempsRealise) AS Total FROM
2 PARTICIPER A, COUREUR B, EQUIPE C
3 WHERE A.NumeroCoureur=B.NumeroCoureur And B.CodeEquipe=C.CodeEquipe And
4 NumeroEtape<=13
5 GROUP BY B.CodeEquipe, NomEquipe ORDER BY Total

```

## Exercice 11 :

À partir du système d'information de l'entreprise. le service des ressources humaines peut extraire et analyser les informations relatives à tous les personnels. celui-ci lui permet en particulier d'exercer un suivi dans le domaine de la formation. Un extrait de ce domaine est présenté sous forme d'un schéma relation :

**Extrait du modèle relationnel du système d'information du service des ressources humaines**

ORGFORM (Numorgform, Nomorgform, Adresseorgform, CPorgform, Villeorgform, Nomorgform, Telorgform, Melorgform)

DATEFORM (Datedebutform)

FORMATION (Codeform, Intituleform, Dureeform, Niveauform, Nbreplaceform, Coutform)

SALARIE (Matriculesal, Nomsal, Prenomsal, Qualificationsal, #Codecategorie)

CATEGORIE (Codecategorie, Libellecategorie)

SERVICE (Codeserv, Nomserv, Ancienneteserv)

PASSERBILAN (#Matriculesal, #Matriculesal, Datebilan)

AFFECTER (#Matriculesal, #Codeserv)

PROPOSER (#Codeform, #Matriculesal, Dateproposition)

SUIVRE (#Codeform, #Matriculesal, Datedebut)

REALISER (#Numorgform, #Codeform, #Datedebutform, NbInscrits)

Construire les requêtes en langage SQL permettant de répondre aux questions suivantes :

1. quel est le nombre de formations suivies par catégories de salariés ayant débuté au cours de la période du 01/06/2011 au 31/12/2011 ?

```

1 SELECT Libellecategorie, count(distinct Codeform) FROM SUIVRE, SALARIE,
2 CATEGORIE WHERE SUIVRE.Matriculesal=SALARIE.Matriculesal AND SALARIE.
3 codecategorie=CATEGORIE.codecategorie
4 AND Datedebut BETWEEN "01/06/2011" AND "31/12/2011"
5 GROUP BY Libellecategorie

```

2. quelles sont les catégories pour lesquelles le nombre d'heures de formation est supérieur à la moyenne du nombre d'heures des formations suivies par l'ensemble des personnels ?

```

1 SELECT Libellecategorie FROM SUIVRE, SALARIE, CATEGORIE, FORMATION
2 WHERE SUIVRE.Matriculesal=SALARIE.Matriculesal AND
3 SALARIE.codecategorie=CATEGORIE.codecategorie AND
4 FORMATION.Codeform=SUIVRE.Codeform

```

```
GROUP BY Libellecategorie
HAVING SUM(Dureeform) > (SELECT AVG(Dureeform) SUIVRE, FORMATION
WHERE SUIVRE.Codeform=FORMATION.Codeform)
```

3. le responsable des ressources humaines souhaite intégrer dans la base de données une nouvelle formation liée au sertissage des boîtes de conserve. Les nouvelles données à insérer sont les suivantes : "FORM587, sertissage niveau 1, 25j, perfectionnement, 12, 525 "

Ecrire la requête permettant de mettre à jour la base.

```
INSERT INTO FORMATION VALUES("FORM587","sertissage_niveau_1",600,
"perfectionnement",12,525)
```

## Exercice 12

La société X utilise le logiciel de gestion de base de données Access pour gérer ses clients et ses représentants. Voici la liste des tables créées dans Access :

Ecrire les requêtes suivantes

- Afficher la liste des clients appartenant à la catégorie tarifaire n°1, classée par ordre alphabétique

```
SELECT CODE_CLT, NOM_CLT FROM client WHERE NUM_CAT=1 ORDER
BY NOM_CLT ASC
```

- Afficher la liste des clients (code, nom de client) rattachés au représentant Sony

```
SELECT CODE_CLT, NOM_CLT FROM client, representant WHERE client.
NUM_REP=representant.NUM_REP AND NOM_REP="Sony"
```

- Afficher la liste des clients bénéficiant d'une remise de 10%

```
SELECT CODE_CLT, NOM_CLT FROM client, categorie_tarifaire WHERE client.
NUM_CAT=categorie_tarifaire.NUM_CAT AND REMISE="10%"
```

- Afficher la liste des représentants (Numéro et nom) dépendant du chef de secteur SONYBIS

```
SELECT NUM_REP, NOM_REP FROM representant, couvrir, departement WHERE
representant.NUM_REP=couvrir.NUM_REP AND
couvrir.CODE_DEP=departement.CODE_DEP AND CHEF_SECTEUR="
Sonybis"
```

- Afficher la liste des départements (code, nom, chef de secteur)

```
SELECT * FROM departement
```

- Afficher la liste des chefs de secteur

```
SELECT DISTINCT CHEF_SECTEUR FROM departement
```

## Exercice 13

Le responsable du SAV d'une entreprise d'électroménager a mis en place une petite base de données afin de gérer les interventions de ces techniciens. Le modèle relationnel à la source de cette base de données est le suivant :

- **Client (Codecl, nomcl, prenomcl, adresse, cp, ville)**
- **Produit (Référence, désignation, prix)**
- **Techniciens (Codetec, nomtec, prenomtec, tauxhoraire)**
- **Intervention (Numéro, date, raison, #codecl, #référence, #codetec)**

Le responsable vous demande d'écrire en langage SQL les requêtes suivantes :

1. La liste des produits (référence et désignation) classées du moins cher au plus cher.

**select Reference, designation from produit order by prix**

2. Le nombre d'intervention du technicien n°2381.

**select count (\*) from Intervention where codetec =2381**

3. La liste des clients ayant demandé une intervention pour des produits d'un prix supérieur à 300 DA.

```
1 select nomcl from Client clt, Produit prod, Intervention int where clt.codecl=int.codecl
2 and prod.Reference=int.Reference and prod.prix>300
```

4. Les interventions effectuées par le technicien : 'Mentri Mohamed' entre le 1er et le 31 août 2009.

```
select Numero , date , raison from Intervention int , Techniciens tec
where int.codetec=tec.codetec and tec.nomtec="Mentri" and tec.prenomtec="Mohamed" and int.
date between "2009-08-01" and "2009-08-31"
select Numero , date , raison from Intervention int , Techniciens tec
where int.codetec=tec.codetec and tec.nomtec="Mentri" and tec.prenomtec="Mohamed" and MONTH(
int.date)=8 and YEAR(int.date)=2009
```

5. Par ailleurs il vous informe que le produit référencé 548G a vu son prix augmenter (nouveau prix = 320 DA).

```
update Produit set prix=320 where Reference="548G"
```

6. Vous apprenez également par le directeur des ressources humaines qu'un nouveau technicien a été recruté : son code est le 3294, il s'appelle 'sari Mohamed' et est rémunéré à un taux horaire de 15 DA.

```
insert into Technicien values(3294,"sari","Mohamed",15)
```

## Exercice 14

La représentation textuelle suivante est une description simplifiée d'une base de données de gestion de facturation d'une entreprise commerciale.

- **Client (Numcli, Nomcli, Prenomcli, adressecli, mailcli)**
- **Produit (Numprod, désignation, prix , qte\_stock)**
- **Vendeur (Idvendeur, Nomvendeur, adresse\_vend)**
- **Commande (Numcom, #Numcli, #Idvendeur, #Numprod, date\_com, qte\_com)**

On suppose que Numcli, Numprod, Idvendeur et Numcom sont de type numérique.

Le nom, le prénom et l'adresse des clients ainsi que les vendeurs sont des informations obligatoires, le mail peut ne pas être indiqué.

La valeur par défaut de la quantité en stock des produits (qte\_stock) est égale à 0

Exprimer en SQL les requêtes suivantes :

1. Créer les tables : Client, Produit, Vendeur et Commande. create table Produit( Numprod int primary key , designation varchar(30), prix float , qte\_stock int default 0 ) create table commande( Numcom int primary key , Numcli int , idvendeur int , Numprod int date\_com date qte\_com int FOREIGN KEY(Numcli) REFERENCES Client(Numcli), FOREIGN KEY(idvendeur) REFERENCES Vendeur(idvendeur), FOREIGN KEY(Numprod) REFERENCES Pro- duit(Numprod) )
2. la liste des clients de Alger.

```
select * from Client where adressecli like "%Alger%"
```

3. la liste des produits (Numprod, désignation, prix) classés de plus cher au moins cher.

```
select Numprod , designation , prix from Produit order by prix ASC
```

4. noms et adresses des vendeurs dont le nom commence par la lettre 'M'.

```
select Nomvendeur , adresse_vend from Vendeur where Nomvendeur like "M%"
```

5. la liste des commandes effectuées par le vendeur "Mohammed" entre le 1er et

```
1 select Numcom, Numcli, Idvendeur, Numprod, date_com, qte_com from Commande cmd, Vendeur vend
2 where cmd.Idvendeur=vend.Idvendeur and vend.Nomvendeur="mohammed"
3 and cmd.date_com between "2012-01-01" and "2012-01-30"
```

30 janvier 2012.

6. le nombre des commandes contenant le produit n° 365.

```
select count (*) from Commande where Numprod =365
```

## Exercice 15

Soit la base de données suivante :

Ecrire les commandes SQL permettant de rechercher :

1. La liste de tous les étudiants.

```
select * from Etudiant
```

2. Nom et coefficient des matières.

```
select nom_matiere , coefficient from Matiere
```

3. Les numéros des cartes d'identité des étudiants dont la moyenne entre 7 et 12.

### Etudiant

numéro_carte_etudiant	Nom	Prénom	Date_naissance	Section
01234567	Ben Salah	Ahmed	12/08/1988	Informatique
01234568	Ben Mahmoud	Sami	02/09/1990	Math
01234569	Marzougui	Rami	23/01/1988	Informatique

### Matière

code_matière	nom_matière	coefficient
12508	Base de données	1.5
12518	Algorithmme	3

### Note

numéro_carte_etudiant	code_matière	note_examen
01234567	12508	15.5
01234567	12518	5.5
01234568	12518	10.5
01234569	12518	8.75

```
select numero_carte_etudiant from Note, Matiere mat where Note.
code_matiere=Mat.code_matiere
group by numero_carte_etudiant
having (sum(note_examen*coefficient)/sum(coefficient)) between 7 and 12
```

4. La liste des étudiants dont le nom commence par 'ben'.

```
select * from Etudiant where Nom like "Ben%"
```

5. Le nombre des étudiants qui ont comme matière '12518'.

```
select * from Note where code_matiere=12518
```

6. La somme des coefficients des matières.

```
select sum(coefficient) from Matiere
```

7. Les noms des étudiants qui une note\_examen >10.

```
select distinct Nom from Note , Etudiant
where Note.numero_carte_etudiant=Etudiant.numero_carte_etudiant and note_examen >10
```

8. Afficher les noms et les coefficients des matières étudiées par l'étudiant "20/01234568".

```
select nom_matiere , coefficient from Note , Matiere where
Note.numero_carte_etudiant="20/01234568"
```

## Exercice 16

Afin d'assurer la qualité des produits attendus par les Clients, l'entreprise cherche à optimiser la gestion des pannes pouvant survenir dans les infrastructures de production nécessaires à la fabrication du Ciment.

Voici un extrait de la base de données :

- **TECHNICIEN (idTech, nom, prénom, spécialité)**
- **STATION (idstat, nom, Position, coordLat, coordLong, phase)**
- **MACHINE (idmach, état, dateMiseEnService, dateDernièreRévision, #idStat)**
- **TYPEINCIDENT (id, description, tempsRéparationPrévu)**
- **INCIDENT (idInc, remarques, dateHeure, dateHeureCloture, #idmach, #idType)**
- **INTERVENTION (idInterv, dateHeureDébut, dateHeureFin, #idInc, #idTech)**

1. Rédiger la requête SQL permettant d'obtenir la liste par ordre alphabétique des noms et prénoms des techniciens ayant réalisé une intervention sur la Machine identifiée par Ber001.

```
select nom , prenom from TECHNICIEN tec , INCIDENT inc , INTERVENTION int where tec
.idTech=int.idTech and int.idInc=inc.idInc and idmach="Ber001" order by nom ASC ,
prenom ASC
```

2. Rédiger la requête SQL permettant d'obtenir la liste des phases ayant connue un incident de "sur-chauffage" pour le mois Mai 2013.

```
select distinct phase from STATION st, MACHINE mch, INCIDENT inc, TYPEINCIDENT type where inc.
idmach=mch.idmach and st.idstat=mch.idstat and type.id=inc.idType
and type.description="sur-chauffage" and MONTH(dateHeure)=5 and YEAR(dateHeure)=2013
```

3. Rédiger la requête SQL permettant d'obtenir le nombre d'incidents non clôturés.

```
select count(*) from INCIDENT where dateHeureCloture is NULL
```

4. Rédiger la requête SQL permettant d'obtenir la liste des noms des stations ayant eu plus de dix incidents.

```
select nom from STATION st, MACHINE mch, INCIDENT inc where inc.idmach=mch.idmach and st.idstat=mch.idstat  
GROUP by nom having count (*) >10
```

## Exercice 17

Voici un extrait de la base de données gestion des ventes :

- **Produit** (**Ref**, Designation, PrixUnitaire, Dimension, #code\_Machine)
- **Vente** (**Ncom**, Ref, Qte, DateLiv)
- **Commande** (**Ncom**, DateCmd, #CodeClt, #Code\_Salarie)
- **Produit\_concurrent** (**Ref**, Designation, PrixUnitaire, PrixUnitaire, Dimension, #code\_Machine, Nom\_Concurrent)

1. Donner la requête qui permet d'obtenir le chiffre d'affaire mensuel de l'année en cours.

```
select sum(Qte*PrixUnitaire), MONTH(DateCmd) from Produit, Vente, Commande where  
Produit.Ref=Vente.Ref and Vente.Ncom=Commande.Ncom and  
YEAR(DateCmd)=YEAR(NOW()) group by MONTH(DateCmd)
```

2. Donner la requête qui calcule le taux de vente de chaque produit.

```
select Ref, sum(Qte)/(select sum(Qte) from Vente) from Vente group by Ref
```

3. Donner la requête qui affiche le produit le plus vendu du mois en cours.

```
1 select Ref, Designation, tot from (select Ref, Designation, sum(Qte) as tot from Produit,  
2 Vente where Produit.Ref=Vente.Ref group by Ref) order by tot DESC LIMIT 1
```

4. La table produit concurrent est composée des informations sur les produits vedettes des concurrents ; Donner la requête qui permet d'ajouter tous les produits du concurrent GleenAlu à la table Produits.

```
insert into Produit (Ref,Designation,PrixUnitaire,Dimension,code_machine)  
(SELECT Ref, Designation, PrixUnitaire, Dimension,code_machine FROM Produit_concurrent where  
Nom_Concurrent="GleenAlu")
```

## Exemple d'application

### *Application : Gestion des stages au sein de la faculté.*

Ce projet traite la gestion des stages au niveau de la faculté des sciences de l'ingénieur, cette gestion passe par plusieurs procédures ou a la dernière année d'étude, l'étudiant peut demander un stage au sein d'une entreprise. le responsable des stages au sein de la faculté organise et affecte des étudiants à faire des stages dans des organismes d'accueils afin de préparer un rapport de fin d'étude.

Ce responsable établi des conventions avec les entreprises, informe et affecte les étudiants selon leurs spécialités. Ce stage peut être accepté ou refusé selon les cas.

Au début l'étudiant commence par demander un stage et consulte les possibilités offertes, le responsable prend contacte avec un organisme d'accueil et établi des conventions en accord avec l'étudiant et l'entreprise. s'il ya un désaccord l'étudiant annule son stage. Dans le cas favorable l'inscription au stage est faite et l'étudiant est affecté.

Les objectifs du projet :

- Etude de l'existant sur les problèmes de la gestion des stages au sein de la faculté
- Elaborer une solution automatisé qui facilité le traitement de la gestion des stages
- Conception d'un système d'information
- Réalisation d'une application avec Access qui permettra de tracer un modèle entité association
- Etablir des requêtes suivant les demandes de l'utilisateur

Etapes a suivre:

- Collecte des données sur la procédure de la gestion des stages
- Faire un dictionnaire des données
- Création des tables
- Création de formulaires de saisie,
- Création des états pour l'édition,
- Conception des requêtes pour l'interrogation de la base de données.

La gestion des stages au sein de la faculté est très importante pour alléger le travail du responsable des stage et garder des traces des affectation des étudiants et des conventions qui sont établis au fur et a mesure, cela permettra d'avoir une base de données complète concernant les organismes qui permettent aux étudiants de faire des stages dans leur locaux et apprendre les principes opérationnelles.

### **1.2- Objectifs.**

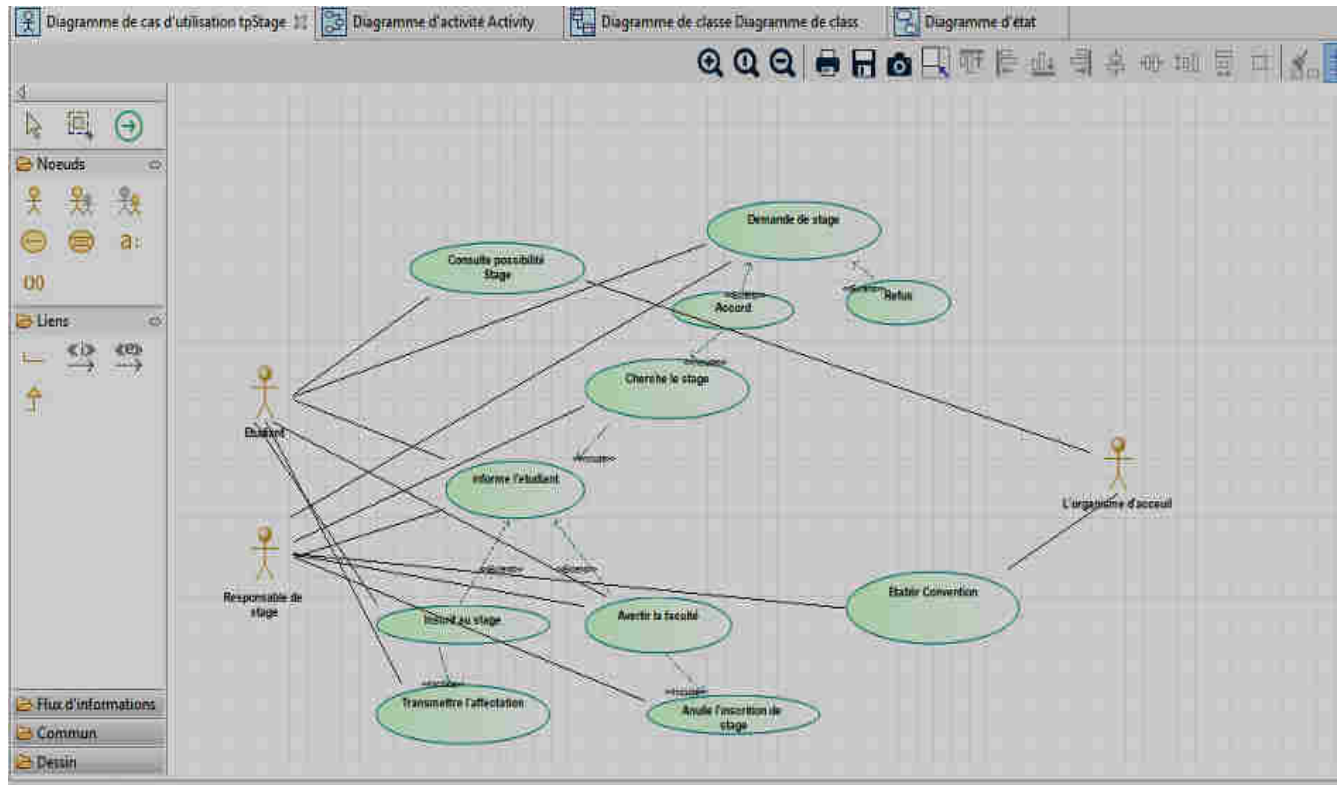
Le projet consiste a la mise au point d'une solution informatique qui permettra de gérer les stages automatiquement pour gagner du temps et simplifier les procédures, l'objectif principale de l'application est l'affectation des étudiants dans des organisme d'accueils ( généralement des entreprises économiques ) pour les besoins d'apprentissage.

L'application en Access permettra de gérer les étudiants ainsi que leurs affectation et voir la conventions établis.

## **2- Diagramme de cas d'utilisation**

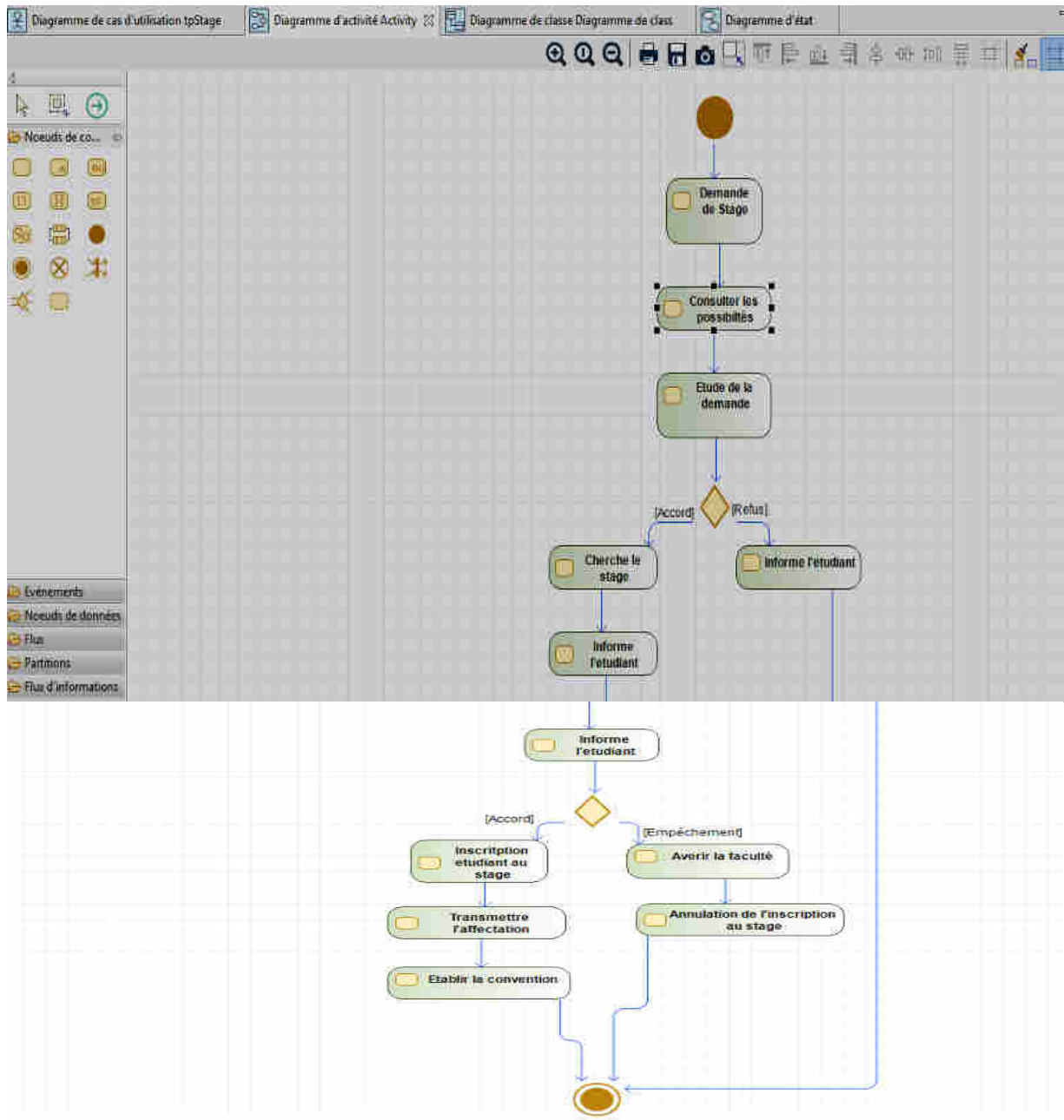
Le diagramme de cas d'utilisation nous permet de capturer les besoins des utilisateurs pour l'application que nous allons développer du système d'information GESTION DES STAGES

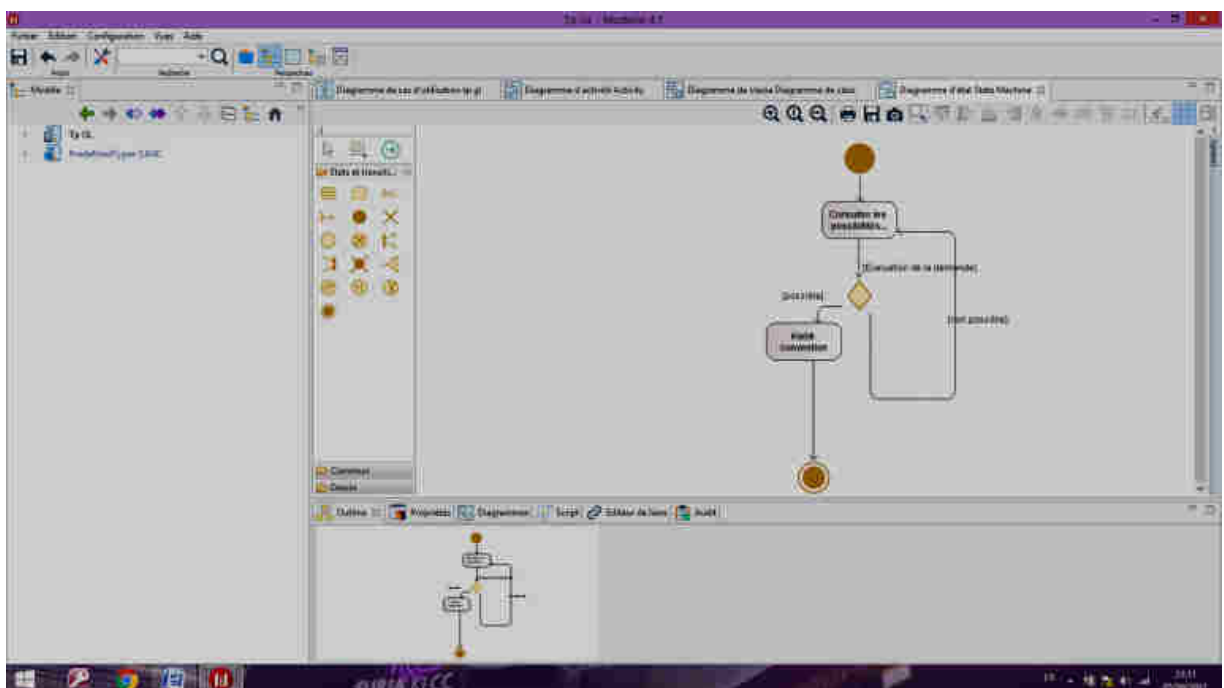
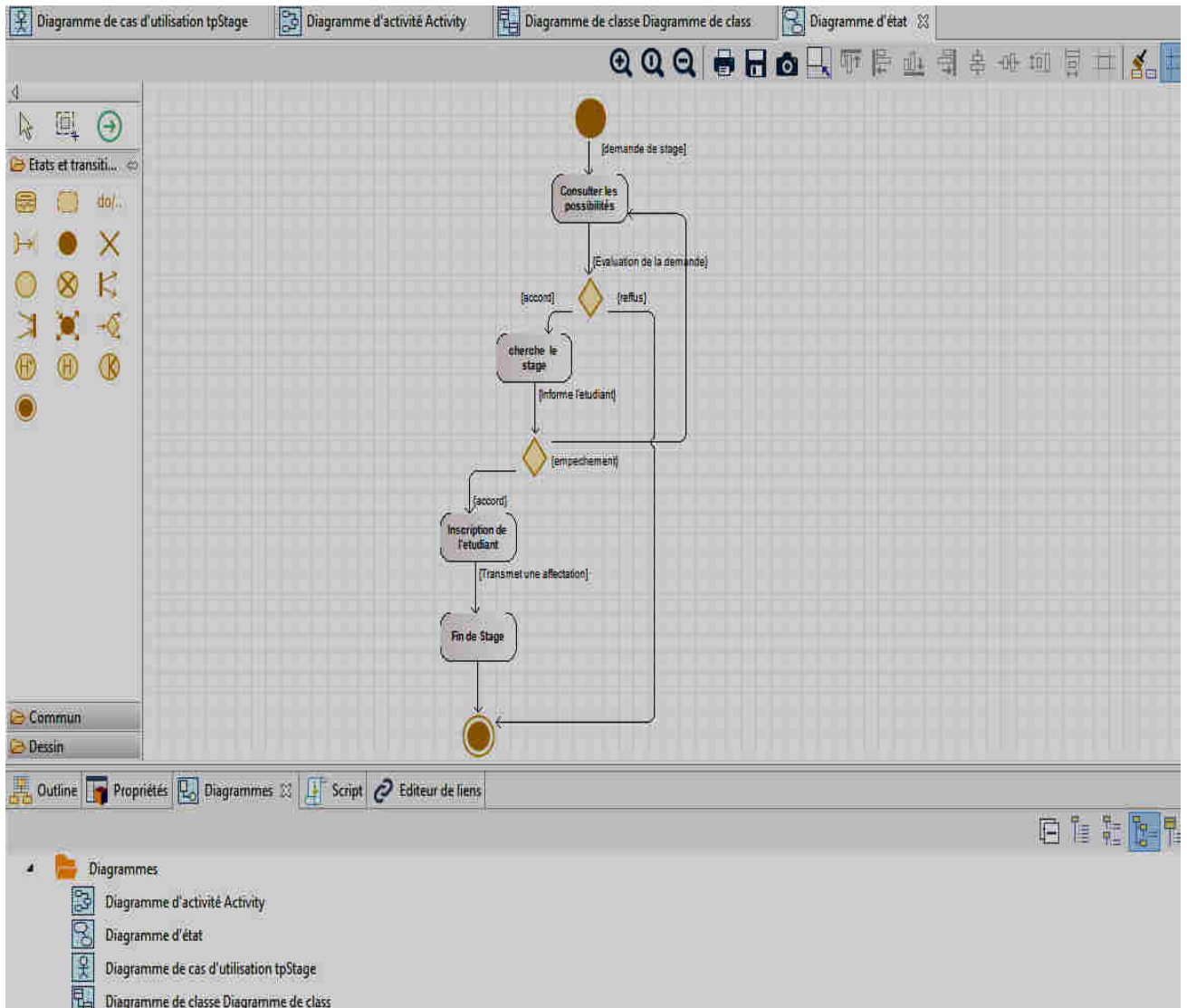




# 1. Diagramme d'activité

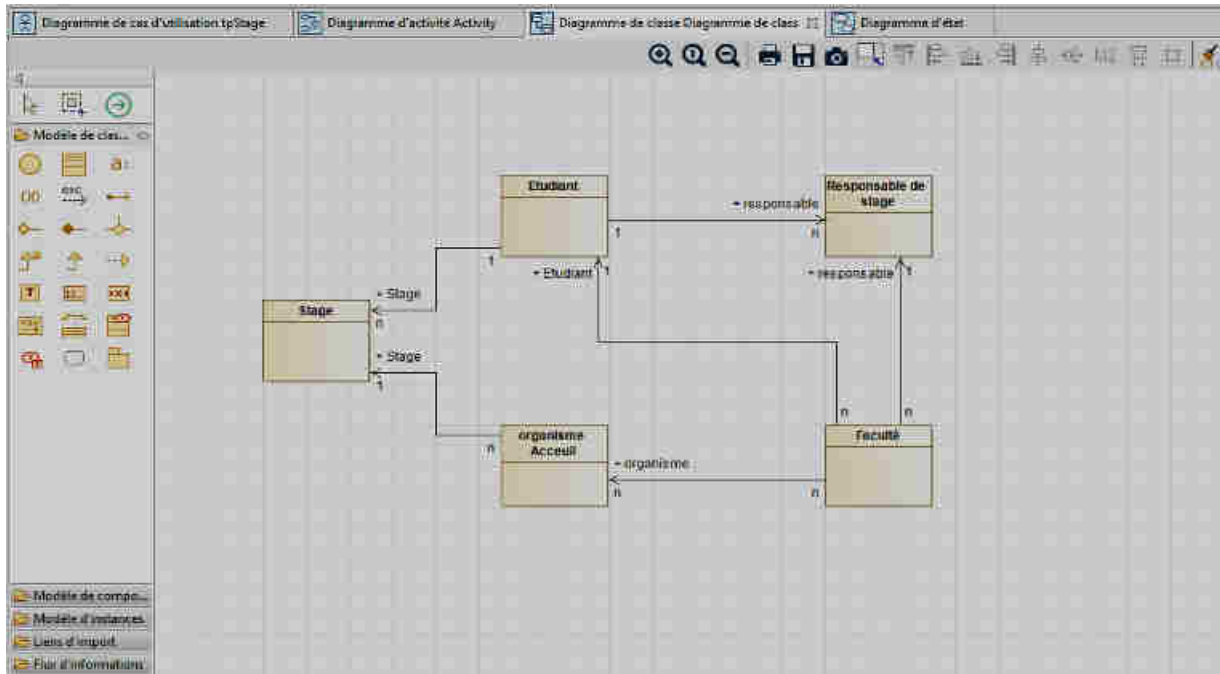
Le diagramme d'activité nous permet de modéliser les tâches qui doivent être effectuées par les acteurs et le système et leur ordonnancement. Les dépendances entre les tâches des activités et leurs dépendances.



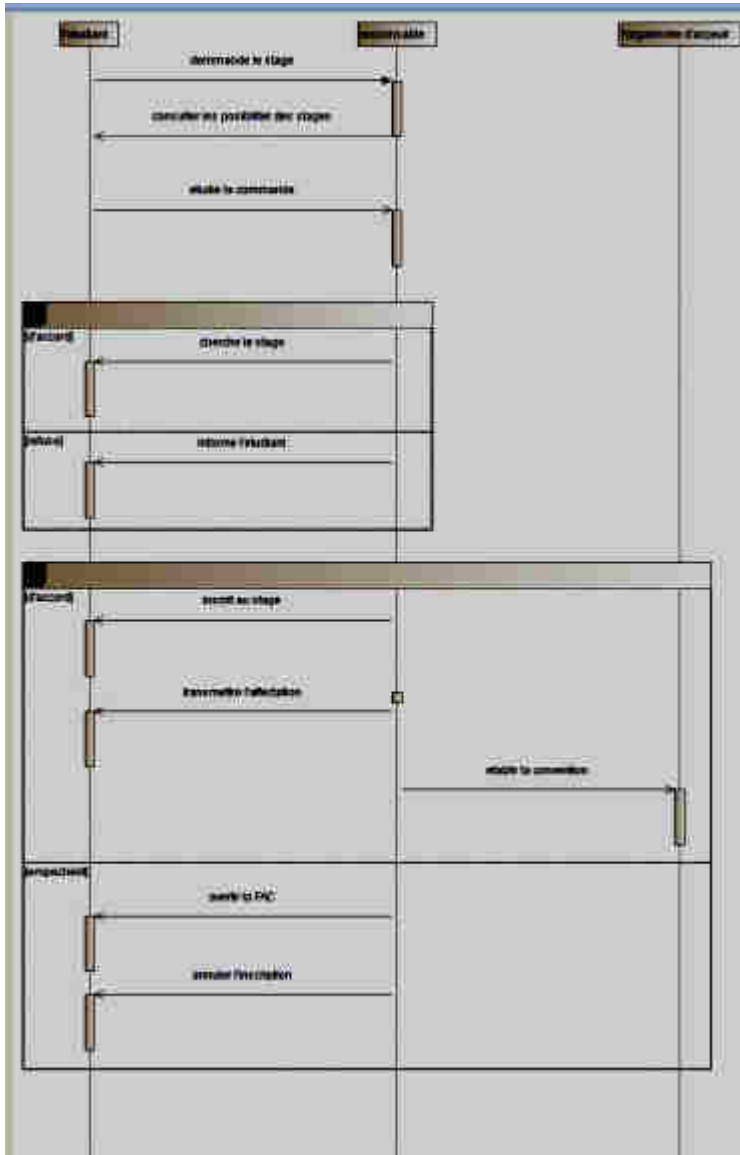


## Diagramme de classe.

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.



## 4. Diagramme de séquence



## La Modélisation des Données

Pour la représentation des données mémorisables du S.I, il convient de modéliser les données et les traitements. Nous présentons dans cette section le modèle conceptuel des données.

Nous présentons le dictionnaire de données puis le modèle conceptuel de données et enfin le modèle relationnel.

### 5.1. Dictionnaire de Données

Les informations utilisées ont été collectées au travers des entretiens et des documents utilisés.

**Etudiant**(mat\_etud, nom\_etud, prenom\_etud, section, groupe, , année universitaire)

**Faculté** (cod\_fac, nom\_fac, departement)

**Organisme d'accueil** (cod\_org, libellé\_org, adresse, code postal, ville, Tel, Fax, secteur activité, nom\_coodinateur)

**Stage** (Num\_Stag, **mat\_etud**, **cod\_org**, date\_début stage, dat\_fin stage, tuteur, secteur stage)

**Responsable de stage**( cod\_resp, nom\_resp, prenom\_resp)

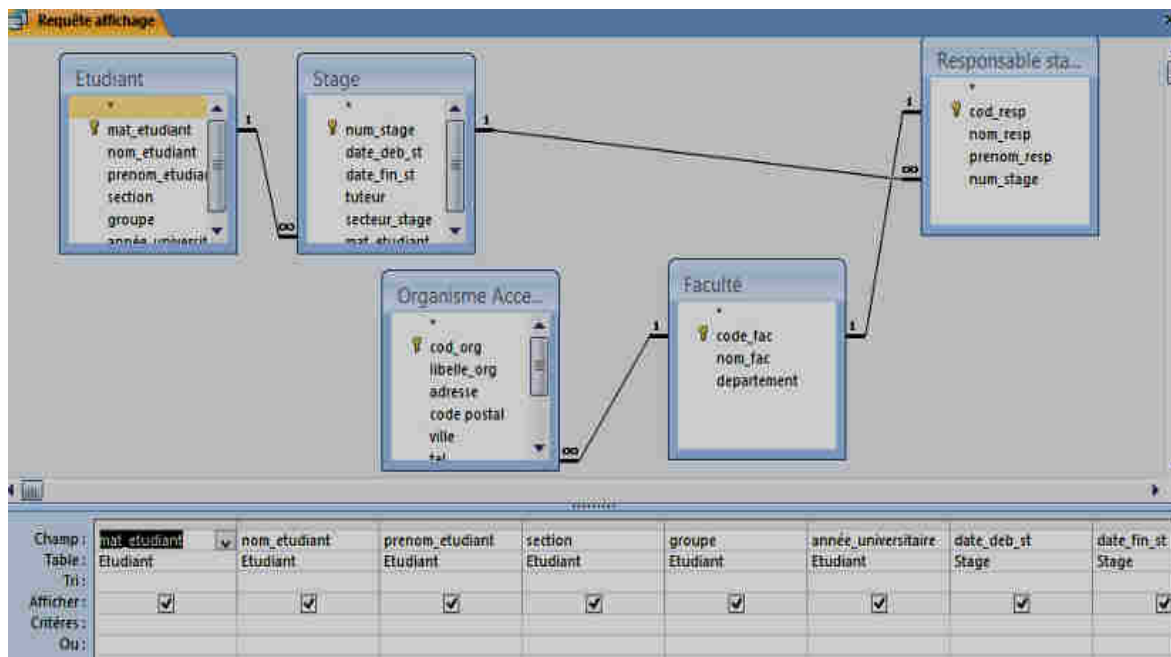
**APPARTIENT**(mat\_etud ,année Universitaire, cod\_fac, departement)

<b>codes</b>	<b>idonnées</b>	<b>type</b>
Mat_etud	Matricule etudiant	N
Nom_etud	Nom etudiant	A
Prénom_etud	Prénom etudiant	A
Section	section	AN
groupe	groupe	AN
Année universitaire	Année universitaire	Date
Cod_fac	Code faculté	N
Nom_fac	Nom de la faculté	A
departement	département	A
Cod_org	Code organisme d'accueil	N
Libellé_ent	Libellé entreprise	A
adresse	Adresse entreprise	AN
Code postal	Code postal	N
Ville	Ville	A
tel	Téléphone	N
fax	Fax	N
Secteur activite	Secteur d'activité	A

responsable	Responsable	A
Dat_debut_stage	Date début du stage	Date
Dat_fin_stage	Date fin de stage	Date
tuteur	Tuteur	A
Secteur stage	Secteur stage	A
Code_resp	Code du responsable	N
Nom_resp	Nom du responsable	A
Pren_resp	Prénom du responsable	A

## 5.2. Le Modèle Conceptuel des données.

Pour la modélisation conceptuelle des données nous utilisons le modèle Entité-Association qui est fondé sur les concepts d'Entité, Association, Propriété et cardinalité.



## 5.3. Le Modèle Relationnel

### Liste des Tables

**Etudiant** (mat\_etud, section, nom, prenom, année universitaire)

**Faculté** (cod\_fac, nom\_fac, departement)

**Organisme d'accueil** (cod\_org, libellé\_ent, adresse, code postal, ville, Tel, Fax, secteur activité, responsable)

**Stage** (mat\_etud, cod\_org, année, date\_début stage, dat\_fin stage, nom responsable, tuteur, secteur stage)

**Responsabel stage**( cod-res, nom prenom)

**APPARTIENT**(année Universitaire, mat\_etud, cod\_fac, departement)

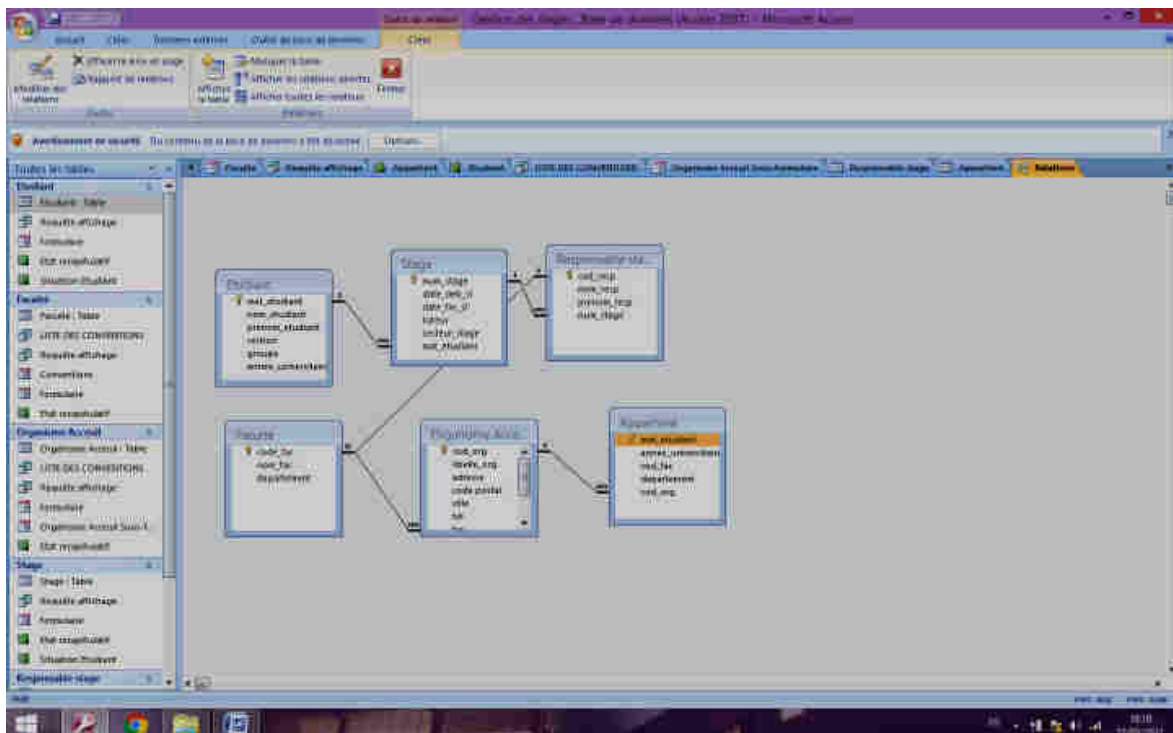
# 5.Réalisation de l'application

L'application que nous avons réalisée avec l'outil Access nous a permis d'automatiser le système d'information sur agence de voyage

Nous présentons les formulaires que nous avons conçu, les patterns de requêtes utiles et les états demandés par les utilisateurs.

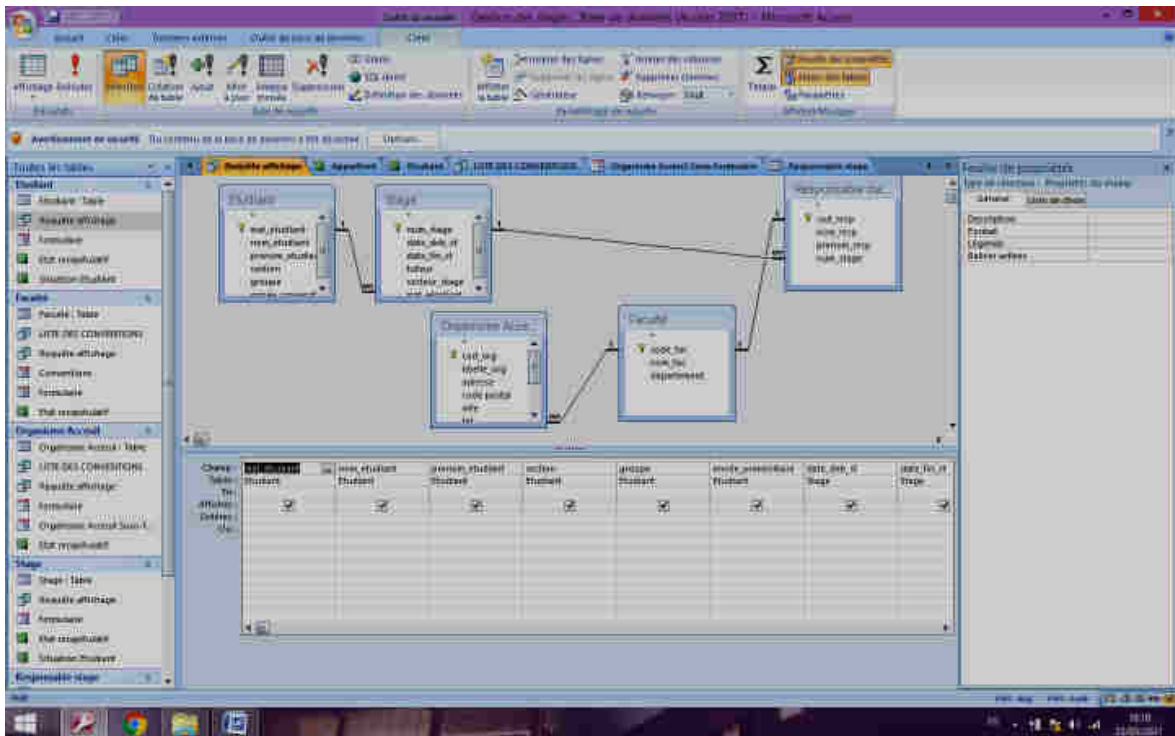
### SOLUTION :

Forme générale de l'application (tables, requêtes, formulaires et états)

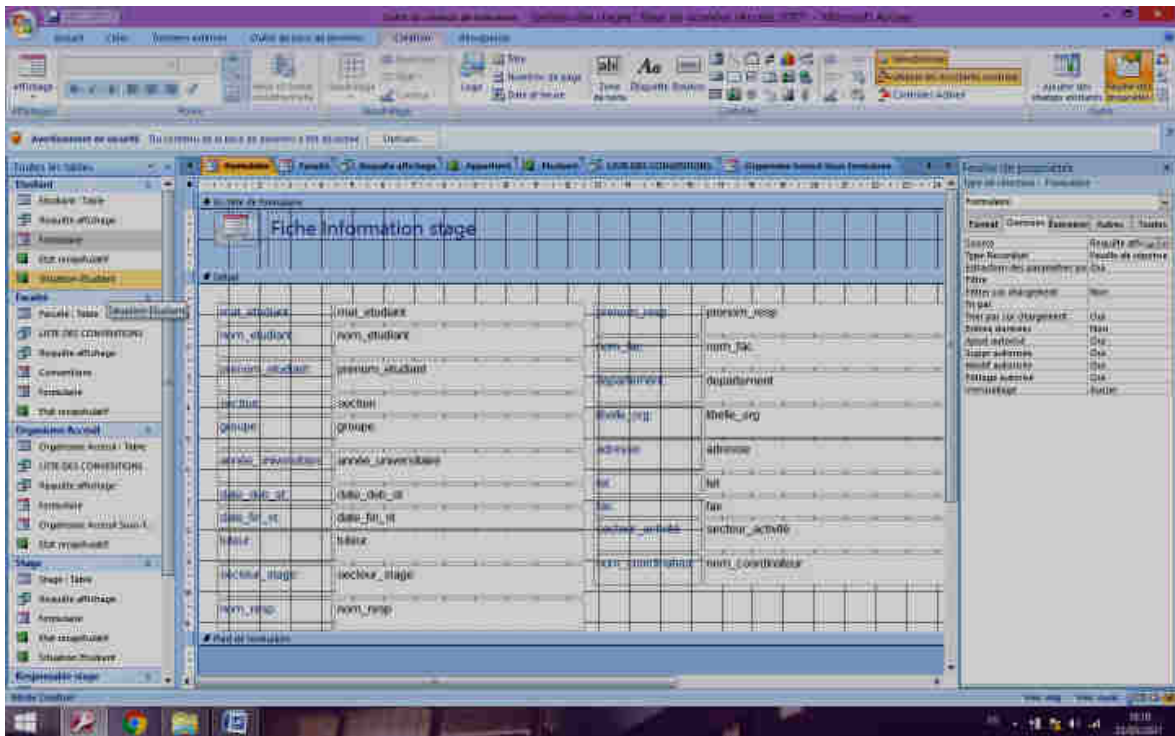




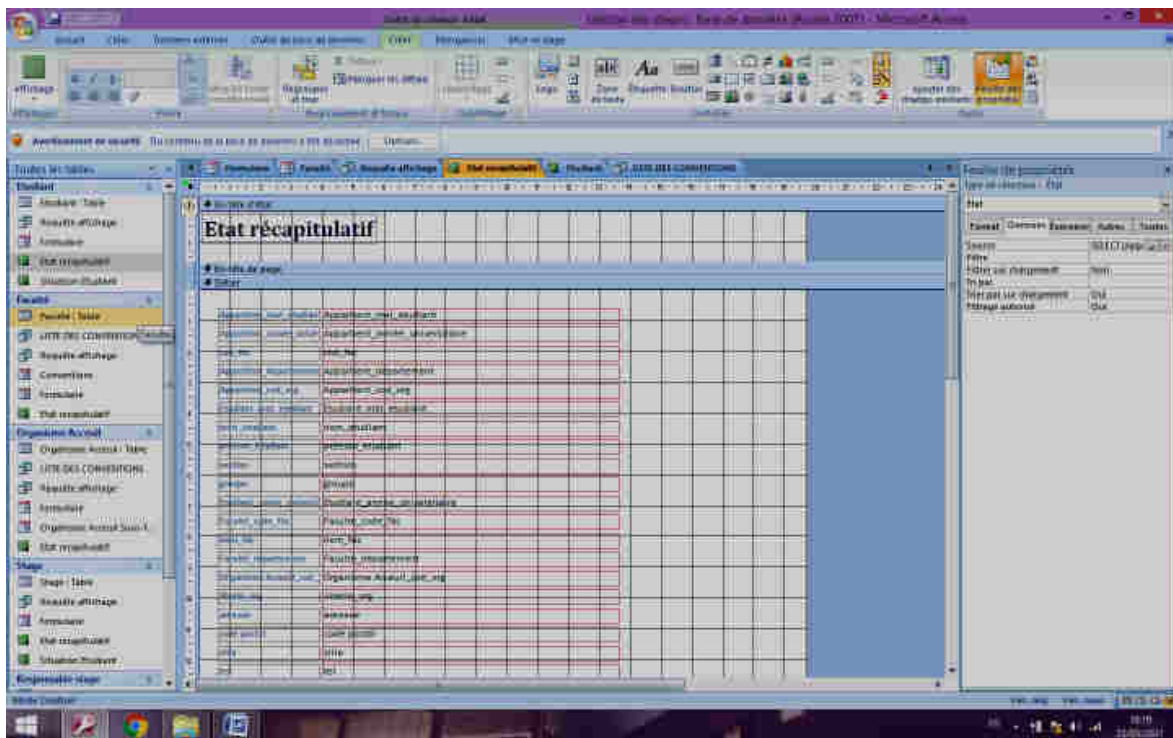
## 1- créations des requêtes en mode créations.



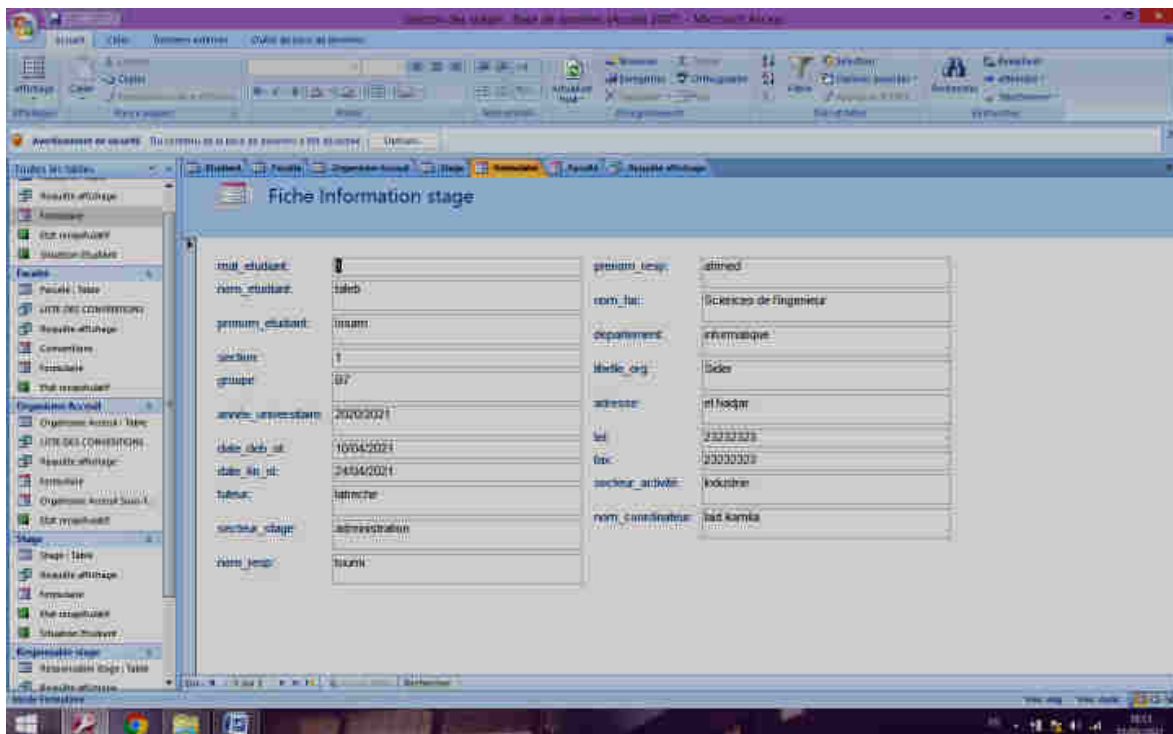
## 2- Création du formulaire ( Mode création)



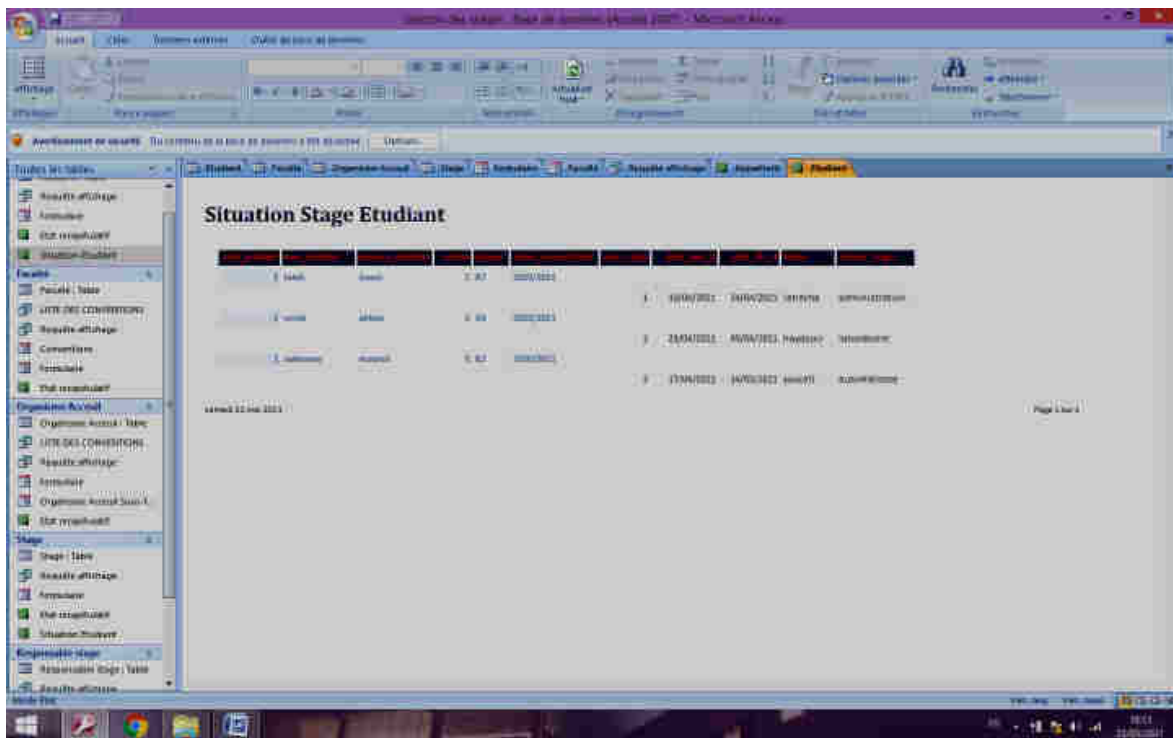
### 3- Création d'un état ( plus un état récapitulatif)



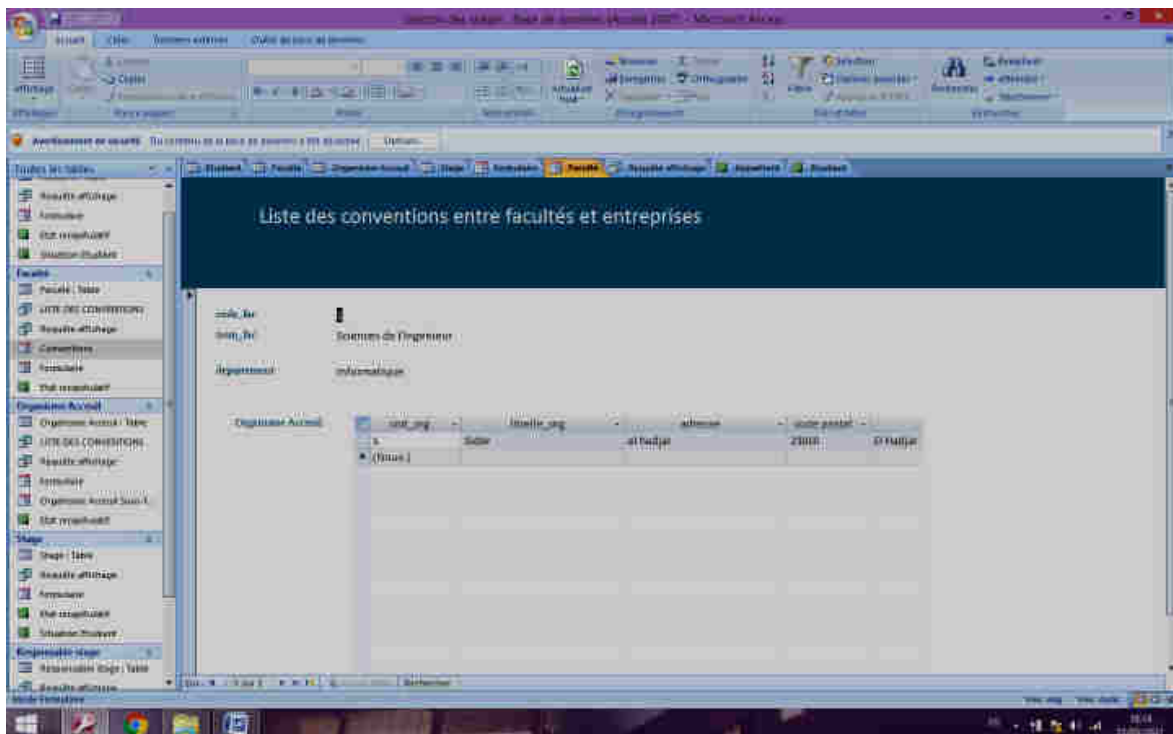
### 4- Création de la fiche stage d'un étudiant.



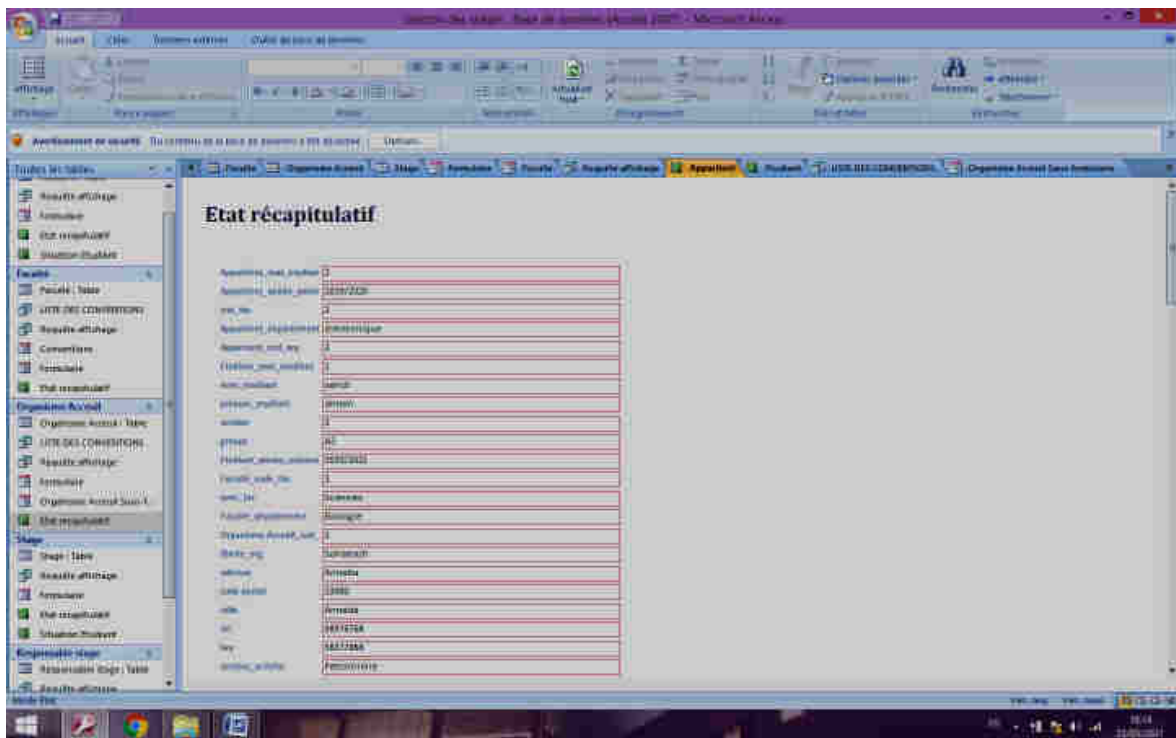
5- Création d'un état situation de l'étudiant par rapport a son stage.



6- Liste des conventions Facultés – Entreprises.



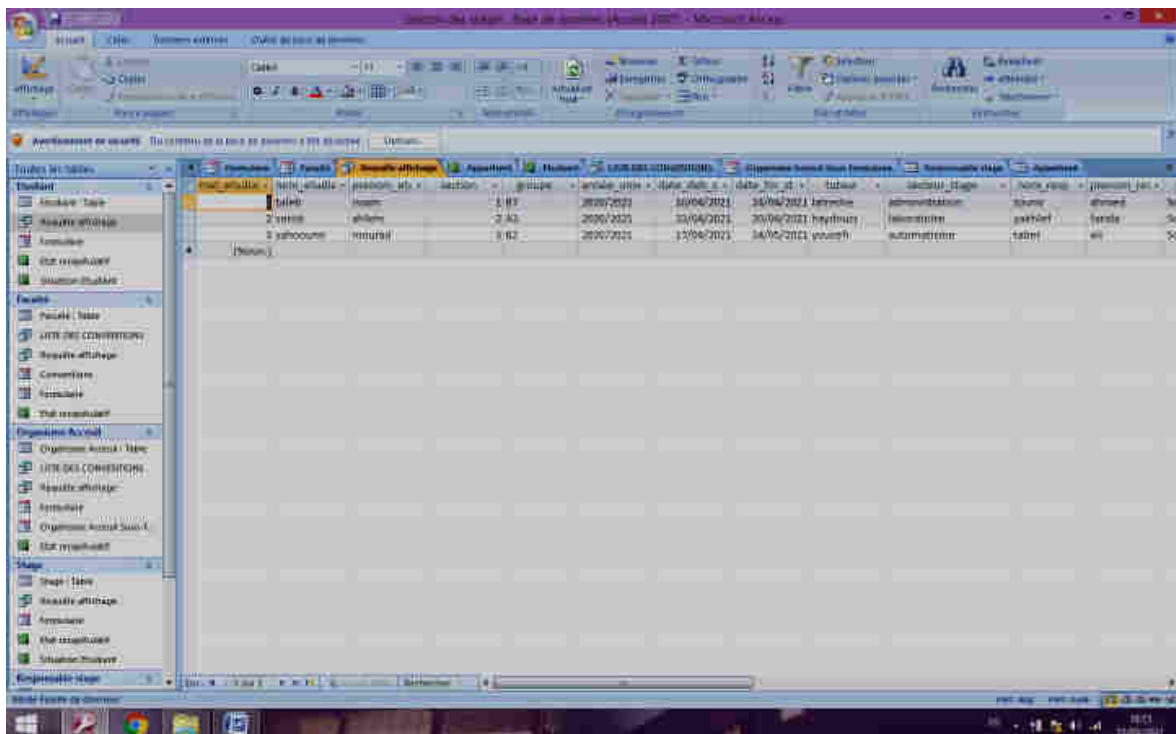
## 7- Autres états (a la demande)



**Etat récapitulatif**

Apprentissage, état, section	0
Apprentissage, année, date	2020/2021
act, nb	2
Apprentissage, appellation	Administratif
Apprentissage, act, nb	2
(statut, code, section)	1
Act, statut	actif
person, profil	000001
statut	1
profil	001
Personne, année, section	2020/2021
Statut, code, nb	1
act, nb	20000000
Apprentissage, appellation	Administratif
Apprentissage, Act, act	1
act, nb	20000000
act	Administratif
act	00010000
nb	00010000
nb	00010000
nb	00010000

## 8- Requête affichage.



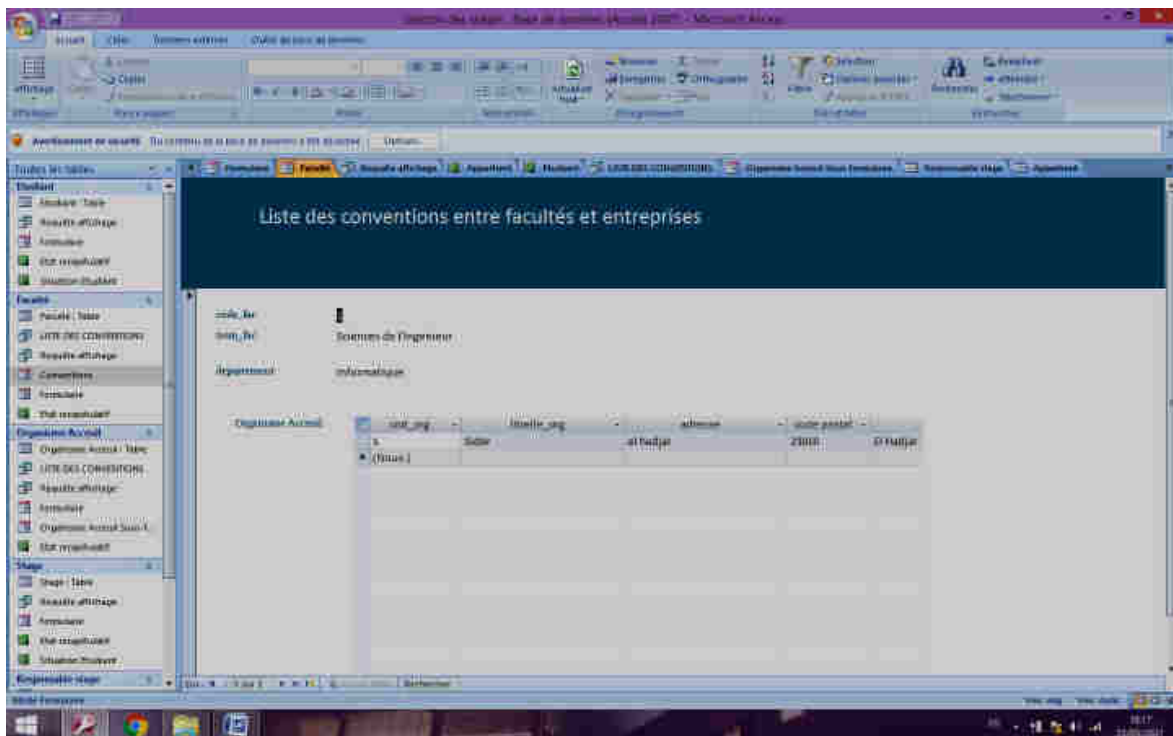
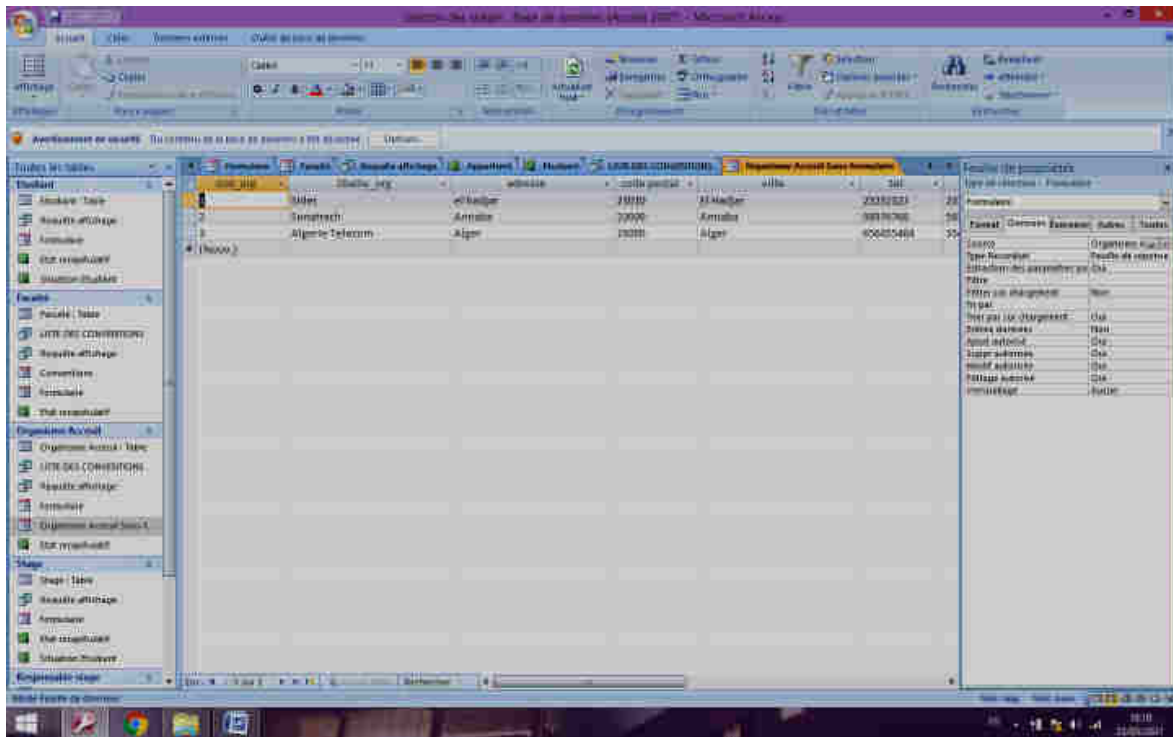
appellation	statut	section	annee	date_debut	date_fin	titre	statut_stage	act, nb	person, act, nb
Administratif	actif	000001	2020	20/06/2021	25/06/2021	Administratif	Administratif	actif	000001
Administratif	actif	000001	2020	20/06/2021	20/06/2021	Administratif	Administratif	actif	000001
Administratif	actif	000001	2020	20/06/2021	24/06/2021	Administratif	Administratif	actif	000001

## 9- Liste des conventions par requêtes.

nom_con	departement	ville	adresse	tel	fax	secteur	statut
2 Sciences de la	informatique	Alger	23000 El Hadje	2522212	2522222	industrie	actif
2 Sciences de la	Biologie	Souk Ahrasj	23000 Ahrasj	2522212	2522222	industrie	actif
3 Sciences de la	gèologie	Alger	10000 Alger	45655444	45655444	Telecommunication	actif

nom_con	departement	ville	adresse	date_creation	date_modif	date_validation	statut
2 sciences	informatique	Alger	23000 El Hadje	25/06/2021	25/06/2021	25/06/2021	actif
2 sciences	informatique	Alger	23000 El Hadje	25/06/2021	25/06/2021	25/06/2021	actif
3 sciences	gèologie	Alger	10000 Alger	12/06/2021	12/06/2021	12/06/2021	actif

10 – liste des organismes d’accueils



## **BIBLIOGRAPHIE**

- 1. HAINAUT Jean-Luc. Bases de données : concepts, utilisation et développement. Dunod, 2012, 2ème édition, 700 pages, Ouvrage complet et très pédagogique, traitant non seulement les concepts fondamentaux, mais aussi les méthodologies de conception.**
- 2. BROUARD Frédéric, BRUCHEZ Rudi, SOUTOU Christian. SQL. Pearson, 2010, 3ème édition, 300 pages, Ouvrage pédagogique rédigé par des praticiens de ce langage.**
- 3. RAMAKRISHNAN Raghu, GEHRKE Johannes. Database management systems. Mcgraw-Hill , 2002, 3ème édition, 1 100 pages, Ouvrage de référence très complet sur les bases de données.**
- 4. DATE Chris. SQL and relational theory: how to write accurate SQL code. O'Reilly, 2011, 2ème édition, 450 pages,**
- 5. GARDARIN Georges. Bases de données. Eyrolles, 2003, 790 pages, gratuit en ligne**