

Chapitre 6 : Manipulation d'une base de données

I- Introduction :

Après avoir présenté, dans le chapitre précédent, les différentes commandes de création et de modification de la structure d'une base de données. Nous allons décrire dans ce chapitre comment manipuler le contenu d'une base de données en mode assisté et en mode commande.

II- Manipulation de données en mode assisté :

1) Mise à jour des données :

a- Insertion de lignes :

Activité :

Considérons la base de données dont le schéma est donné ci-dessous :

Employé (Nemp, Nomemp, Grademp, Chef, Daterecrut, Sal, Nodept#)

Département (Nodept, Nomdept, Ville)

- 1- Créer la base de données suivante et l'enregistrer sous le nom « Emploi » sous l'unité C : .
- 2- Insérer les 4uplets suivant dans la table département :

Nodept	Nomdept	Ville
10	Comptabilité	Kairouan
20	Recherche	Nabeul
30	Vente	Sousse
40	Opérations	Tunis
50	Commerce	Bizert

- 3- Insérer les 9uplets suivant dans la table Employé :

Nemp	Nomemp	Grademp	Chef	Daterecrut	Sal	Nodept
7369	Dorsaf	President	Null	11/05/80	5000	10
7499	Ali	Vendeur	7698	03/04/90	800	30
7521	Marwa	Vendeur	7698	05/08/81	1100	30
7566	Achref	Manager	7839	30/01/85	2000	20
7656	Fadwa	Analyste	7566	23/07/83	1050	20
7688	Haifa	Vendeur	7698	18/02/83	600	30
7782	Zied	Manager	7839	18/02/83	900	10
7839	Kawther	Analyste	7566	07/01/95	2500	10
7844	Aymen	secetaire	7698	12/03/89	1000	20

Démarche à suivre :

1. Dans la fenêtre « Base de données », sélectionner la table concernée.

2. Sélectionner le bouton, Ouvrir, une fenêtre qui vous permet d'introduire les données s'affiche.

Remarque :

Les boutons situés dans le coin inférieur gauche de la fenêtre vous permettent de naviguer entre les lignes de la table.

Ligne précédente

Ligne suivante



b- Suppression d'une ligne

Activité 2 :

1. Supprimer le dernier enregistrement de la table Département.

Démarche à suivre :

- 1- Se positionner sur la ligne à supprimer
 - 2- Presser la touche SUPPR ou supprimer l'enregistrement de menu contextuel
 - 3- Confirmer par ok
- c- Modification d'une ligne**

Activité 3 :

1. Modifiez le nom de la ville du département N°40 au lieu de « Tunis » mettez « Djerba ».

Démarche à suivre :

- 1- Se positionner sur la ligne à modifier
- 2- Appliquer la modification désirée.

2) Recherche de données : Requêtes

Une requête correspond à des ordres de programmation enregistrés au format SQL. Elle permet de rechercher une donnée selon un ou plusieurs critères, d'effectuer des calculs...

a- Démarche de création d'une requête :

Activité 4 :

1. Sélectionner l'onglet requête de la fenêtre « base de données »
2. Cliquer sur l'option « créer une requête en mode création »
3. Choisir la table employée puis fermer la fenêtre d'affichage de tables.
4. Afficher les numéros des employés qui ont le grade « vendeur »
5. Enregistrer la requête sous le nom RQ1
6. Modifier la requête précédente pour qu'elle affiche toutes les informations concernant les employés vendeur.
7. Enregistrer la requête
8. Créer une nouvelle requête qui permet d'afficher les noms des vendeurs qui appartiennent au département N°30.
9. Enregistrer la requête sous le nom RQ2

Démarche à suivre :

- 1- Sélectionner l'onglet requête de la fenêtre « base de données »
- 2- Cliquer sur l'option « créer une requête en mode création »
- 3- Choisir la ou les tables désirées, puis fermer la fenêtre d'affichage de tables.
- 4- Choisir les colonnes à afficher dans la requête (utiliser la souris pour glisser le champ de la table vers son emplacement dans la grille d'interrogation ou cliquer directement sur la cellule "Champ" et sélectionner l'attribut dans la liste déroulante proposée pour la ou les tables de la base)

Remarque : Pour afficher toute les colonnes d'une table sélectionner le champ *

- 5- Cocher les cases correspondantes aux colonnes qu'on désire afficher dans le résultat.
- 6- Mettre les critères qui doivent être respecté.
- 7- Visualiser le résultat de la requête tout en cliquant sur le bouton 
- 8- Enregistrer la requête.

3) La définition des clés de tri

On peut trier les données d'une colonne, selon une ou plusieurs clés de tri que se soit dans l'ordre croissant ou décroissant.

Activité 5 :

1. Modifier la requête RQ1 pour trier les noms des employés dans un ordre décroissant.

4) Les critères de filtre

Le filtrage : consiste à retrouver et afficher l'ensemble des lignes d'une table vérifiant une condition (pouvant porter sur plusieurs colonnes en utilisant les opérateurs logiques).

a- Recherche vérifiant un filtre relatif à une colonne de type texte :

Lorsque vous tapez un critère dans un champ texte, Access le place entre guillemets.

La recherche ne tient pas compte des majuscules et minuscules.

Attention aux caractères accentués « pêche » et « pêche » sont différents.

Il est possible d'exclure des lignes tout en écrivant <> ou **pas** devant le terme à exclure.

Il est possible d'utiliser les caractères spéciaux :

✓ * : remplace un ou plusieurs caractères quelconques.

✓ ? : remplace un caractère quelconque.

Voir RQ1 de l'activité 4 ou livre à la page 123.

Activité 5 :

1. Afficher les noms des employés qui commencent par la lettre A.
2. Afficher les noms des employés qui contiennent la lettre A
3. Afficher les noms des employés qui se terminent par la lettre A
4. Afficher tout les noms des employés sauf celui de Dorsaf.

b- Recherche vérifiant un filtre relatif à une colonne de type numérique :

Activité 6 :

1. Chercher tout les informations concernant les employés dont les salaires supérieur ou égal à 600.
2. Chercher tout les informations concernant les employés dont les salaires entre 1000 et 800.
3. Chercher tout les informations concernant les employés dont les salaires égaux à 600 ou à 800.

Pour les filtres relatifs à une colonne de type numérique on peut :

- Saisissez la valeur recherchée.

- On peut sélectionner des valeurs appartenant à un intervalle Ex : <1000 et >600. le **ou** est également utilisable dans ce genre de critère.

- Il est possible d'utiliser les opérateurs de comparaison suivant :

< Inférieur > Supérieur >= Supérieur ou égal

= égal <> Différent de <= Inférieur ou égal

c- Recherche vérifiant un filtre relatif à une colonne de type date :

On utilise les mêmes opérateurs utilisés pour le type numérique. Pour le type date Access rajoute automatiquement # autour de la date.

Activité 7 :

1. Afficher tout les employés dont la date de recrutement est inférieure au 11/02/83.

d- La combinaison de critères :

Il est possible d'utiliser plusieurs critères de sélection, il suffit de les placer dans les différentes lignes et colonnes de la grille ou sur la même ligne.

- Si les critères sont placés sur la même ligne, ils sont liés par ET (une ligne est affichée si les deux critères sont valables)

Activité 8 :

1. Afficher les numéros des employés qui on un salaire >800 **et** appartiennent au département N°30.
2. Afficher les numéros des employés qui on un salaire >800 **ou** appartiennent au département N°30.

e- La définition d'une requête paramétrée :

Une requête paramétrée, est une requête qui affiche dès son exécution une boîte de dialogue qui vous invite à lui transmettre des informations telle que des critères pour extraire des enregistrements ou une valeur à insérer dans un champ.

Activité 9 :

- 1- Ecrire une requête paramétrée qui permet d'afficher toutes les informations concernant l'employé Ali.
- 2- Ecrire une requête paramétrée qui permet d'afficher les numéros des employés recrutés en 18/02/83.

f- Autre critères :

- le critère **est null** ou **= null** permet de sélectionner les colonnes « vides ».
- le critère **est pas null** ou **<>null** permet d'afficher toutes les lignes répondant à la requête.

Activité 10 :

- 1- Afficher toutes les informations concernant les employés dont leurs chefs ne sont pas mentionnés dans la base.
- 2- Afficher les numéros non nuls des chefs.

5) Les formules :

Les intitulés des colonnes peuvent être utilisés comme des variables. Quand une requête est exécutée, les lignes d'une table sont lues les unes après les autres. A chaque lecture d'une ligne le contenu de chaque champ est mémorisé dans ces variables.

a- la concaténation :

Syntaxe :

Il faut écrire dans la case « Champ » de la grille d'interrogation la formule suivante :
 [Intitulé colonne1] & [Intitulé colonne 2]... & [Intitulé colonne n]

Activité 11 :

- 1- Afficher les noms des employés Concaténés avec leurs grades.

Solution :

[nomemp] & [grademp]

- Si on veut laisser un espace entre les deux mots : [nomemp] & " "& [grademp]

b- les champs calculés :

Une colonne calculée est composée de deux parties :

- Le nom donné à cette colonne
- Un calcul utilisant les différents opérateurs mathématiques : +, -, /, *, et les parenthèses.

Activité 12 :

- 1- Calculer le revient annuel de chaque salarié et affecter le résultat à une colonne appelée RA.

Solution :

RA : [sal] * 12

6) Les propriétés d'une colonne ou d'une requête :

Il est possible de mettre des propriétés sur les colonnes affichées. Ex : préciser le format de la date à afficher, le nombre d'enregistrements à afficher...

Activité 13 :

- 1- afficher les dates de recrutement des 5 premiers employés tout en précisant le nom du mois.

Démarche à suivre :

Pour afficher les 5 premiers enregistrements:

Cliquer sur la zone vide colorée en gris située à côté de la table employée.

Cliquer sur la commande « propriété » de menu contextuel.

Indiquer le nombre de lignes à afficher dans la zone « premières valeurs ».

Pour afficher la date de recrutement tout en précisant le nom du mois:

- 1) Dans la zone de critère, dérouler le menu contextuel
- 2) Choisir l'option propriété
- 3) Choisir le format désiré de la date.

7) Requête sélection basée sur des calculs :

Pour effectuer des calculs avec Microsoft Access, il faut activer la ligne opération tout en cliquant sur le bouton Σ . Elle permet principalement d'utiliser des fonctions de statistique suivantes :

- Somme : calcule la somme des valeurs d'une colonne.

- Moyenne : calcule la moyenne de toutes les valeurs d'une colonne.
- Minimum : calcule la valeur la plus petite dans une colonne.
- Compte : affiche le nombre total des lignes répondant à un critère.

Activité 14 :

- 1- Calculer le montant total des salaires versés
- 2- Afficher le salaire le plus élevé
- 3- Afficher l'effectif des salariés.
- 4- Calculer la moyenne des salaires des vendeurs.

8) Les requêtes multi-tables :

Pour établir une requête à partir de plusieurs tables, on doit associer les tables concernées. Cette opération s'appelle **jointure**, les tables à associer doivent avoir une colonne (clé) en commun. Ex : la colonne « Nodept » se trouve à la fois dans la table employée et la table département.

Activité 15 :

- 1- Créer une jointure entre la table employée et la table département.
- 2- Changer les propriétés de la jointure de façon qu'elle prenne en considération toutes les lignes de la table « employée » et seulement ceux de la table « département » pour lesquels les champs joints sont égaux

Démarche à suivre :

Voir livre page 130

Remarque :

La question 2 de l'activité permet :

Pour les enregistrements de la table employée pour lesquels il n'y a pas d'enregistrement associé, la requête crée une ligne avec des valeurs NULL pour les attributs issus de la table département.

9) Echange de données avec un tableur :

On peut prendre des informations à partir d'un tableur cette opération s'appelle : importation comme on peut envoyer des informations de l'Access vers l'Excel il s'agit donc de l'exportation.

Voir démarche d'importation et d'exportation dans les pages 131, 132, 133.

III- Manipulation de données en mode commande :

1- mise à jour de données :

Creation de la base de données emploi :

```
create table employe (Nemp int(5) primary key,
Nomemp varchar(20) not null,
Grademp varchar(20) not null,
Chef int(5),
Daterecruit date not null,
Sal int(10) not null,
Nodept int(4) references departement (Nodept) on delete cascade on update cascade,);
```

```
create table departement (Nodept int(4) primary key,
nomdept varchar(20) not null,
ville varchar(20) not null,
);
```

a- Insertion de lignes :

Syntaxe :

```
INSERT INTO nom_table [liste_nom_colonne]
VALUES (liste_valeur) ;
```

Noter Bien :

- les types de données de « **liste_valeur** » doivent être compatibles avec ceux des colonnes de la table.
- Les attributs constituant la clé primaire doivent obligatoirement être présents.
- Il faut respecter les contraintes déclarées lors de la création de la table. Ex : Check, Not NULL, les contraintes d'intégrité référentielle appliqués aux clés étrangères.
- Il faut que les valeurs insérer soient dans le même ordre que les colonnes saisies dans « **liste_nom_colonne** »
- Il faut penser toujours à l'unicité de la clé primaire, si elle est saisie une fois dans une requête il ne faut pas le réinsérer dans une autre.
- L'apparition de la « **liste_nom_colonne** » dans la requête est facultative.
- les valeurs non saisies des colonnes qui ne subissent à aucune contrainte d'intégrité reçoivent la valeur null automatiquement.

Activité 16 :

- 1- Utiliser l'éditeur MYSQL pour créer la table employée et la table département
- 2- Insérer dans la table employée les enregistrements mentionnés dans la première activité.
- 3- Insérer les enregistrements de la table departement.

Solution :

On peut insérer le premier enregistrement par plusieurs méthodes :

1^{ière} méthode :

```
INSERT INTO employe (Nemp, Nomemp, Grademp, Chef, Daterecrut, Sal, Nodept)
VALUES (7369, "Dorsaf", "President", Null, "11/05/80", 5000, 10) ;
```

2^{ième} méthode :

```
INSERT INTO employe (Nemp, Nomemp, Grademp, Chef, Daterecrut, Sal, Nodept)
VALUES (7369, "Dorsaf" , "President", "11/05/80", 5000, 10) ;
```

3^{ième} méthode :

```
INSERT INTO employe VALUES ("7499" "Dorsaf", "President", "11/05/80",null,"5000", "10") ;
```

b- Suppression de ligne :

Syntaxe :

DELETE FROM nom_table [WHERE condition] ;

Noter Bien :

- La clause WHERE est optionnelle. Si elle est absente, dans ce cas, toutes les lignes de la table concernée seront supprimées.
- Le paramètre condition sert à indiquer la condition qui doit être vérifié par les lignes à supprimer.
- Il faut respecter les contraintes d'intégrité référentielles lors de la suppression d'une clé.
- Si lors de la création de la table on n'a pas déclaré la contrainte ON DELETE CASCADE pour la clé étrangère la requête Delete ne sera pas exécutée.

Activité 17 :

- 1- Supprimer de la table employée l'enregistrement correspondant à l'employé numéro7656.
delete from employé where nemp = "7656";

c- Modification de lignes :

Syntaxe :

**UPDATE nom_table
Set nom_colonne1 = Expression1 [, nom_colonne2 = Expression2...]
[WHERE condition] ;**

Noter bien

- la modification peut toucher une ou plusieurs colonnes. il suffit juste de les séparées par un virgule.
- Si on ajoute une condition, la valeur de la colonne sera mise à jour dans toutes les lignes vérifiant cette condition.

Activité 18 :

1-Modifier le nom de l'employé numéro 7499 à « Mohamed ».

```
UPDATE employe SET Nomemp = "mohammed" WHERE Nemp = 7699;
```

2-Pour le département numéro 30 modifier le nom de la ville à « Monastir »

```
Update departement set ville = "monstir" where nodept = 30 ;
```

2- Recherche de données : requêtes

Une recherche de données peut être réalisée en faisant référence à une ou plusieurs tables.

L'utilisateur peut effectuer plusieurs types de recherche :

a- Recherche de colonne à partir d'une table : Projection

La projection est une opération qui vise à obtenir un ensemble de colonnes à partir d'une table. Il suffit juste de spécifier les noms des colonnes a sélectionnées séparés par des virgules puis dans la clause **From** on spécifie le nom de la table.

```
Select [DISTINCT] * / Liste _nom _ colonne  
From nom _table ;
```

Exp : SELECT DISTINCT * FROM employe

Noter bien :

- Si on souhaite obtenir tout les colonnes d'une table, le caractère * peut être utilisé à la place de la liste des noms de colonnes.
- Lors de l'affichage de résultat, SQL n'élimine pas les doubles à moins que ça ne soit explicitement précisé par le mot clé **DISTINCT**.
- Si la colonne clé primaire est listée on ne peut pas trouver les doublons dans la table résultat.
- Par défaut les colonnes de la table résultat porte les mêmes noms que la table de départ.
- Il est possible de donner aux colonnes de la table résultat des noms différents de ceux de la table d'origine. Cet entête est présenté juste après le paramètre **liste_ nom _colonne** et elle s'appelle généralement **Alias**.

Activité 19 :

1- lister les différents noms des employés.

2- lister les différents grades des employés. L'entête de la colonne doit être nommé « Grade »

3- Afficher tout les informations concernant la table employée.

Solution :

1- Select distinct Nomemp From employe ;

2- Select distinct grademp "grade" From employe ; ou Select distinct grademp AS grade From employe ;

3- Select * From employe ;

b- Recherche d'une ligne à partir d'une table : Sélection

La sélection ou appelée aussi restriction s'exprime à l'aide de la clause WHERE que l'on ajoute à la requête juste après la liste des tables utilisées.

La différence entre sélection et projection c'est que la projection consiste à lister un ensemble de colonnes et la sélection, en plus des colonnes, elle affiche les différentes lignes qui répondent au critère présenté dans la clause WHERE.

Syntaxe :

```
Select [DISTINCT] * / Liste _nom _ colonne  
From nom _table ;  
WHERE condition ;
```

Noter bien

- Le critère (ou appelé aussi prédicat) est donné sous la forme d'une expression logique dans laquelle on peut utiliser :

- Les opérateurs de comparaison : =, <, >, >=, <= et <>.
- L'opérateur BETWEEN pour les intervalles de valeurs.
- L'opérateur [NOT] IN pour les listes de valeurs.
- L'opérateur IS NULL et IS NOT NULL pour les valeurs indéterminées.
- L'opérateur [NOT] LIKE pour filtrer une chaîne de caractères.
- Les opérateurs logiques : AND, OR, NOT.

Activité 20 :

- 1- Afficher la liste des employés qui ont atteint ou dépassé un salaire de 800 dinars.
- 2- Afficher les noms des employés qui ont atteint ou dépassé un salaire de 800 dinars.
- 3- Déduire la différence entre les deux requêtes précédente.
- 4- Donner la liste des employés qui ont un salaire compris entre 1000 et 600 et dont le Nemp appartient à l'ensemble (7499, 7521, 7566, 7656, 7688), les employés doivent avoir un chef et un grade qui commence par la lettre V.

Solution

1)

Select*

from employe

where sal between 1000 and 600

and Nemp in (7499, 7521, 7566, 7656, 7688) and (chef is not null)

and (grademp like 'V%');

Remarque : le caractère "%" représente n'importe quel nbre de caractère

Le caractère "-" représente un seul caractère quelconque

c- Recherche de données à partir de plusieurs tables : Jointure

Pour établir une requête à partir de plusieurs tables, on doit associer les tables concernées. Une jointure se fait selon le schéma relationnel sur les clés des tables que l'on veut associer.

Syntaxe :

```
Select [DISTINCT] * / Liste_nom_colonne
From nom_table1[alias], nom_table2[alias]...
WHERE condition ;
```

- la jointure réalise une liaison entre deux tables en se basant sur l'égalité des valeurs entre l'une des colonnes de chaque table.

- l'alias permet de donner un nom abrégé à une table.

- le paramètre condition sert à préciser la condition de jointure. il ne faut pas la confondre avec la condition indiquée pour l'opération relationnelle de sélection.

Activité 21 :

- 1- Lister les numéros et les noms des employés qui appartiennent au département comptabilité.
- 2- Donner la liste des clients qui ont atteint les mêmes salaires. Que remarquez-vous ?

Solution :

1- SELECT nemp, nomemp

FROM employe e, departement d

WHERE e.nodept = d.nodept AND d.nomdept = 'comptabilité';

2- Select * from employe x, employe y where x.salaire=y. salaire;

Dans cette requête on a utilisé deux fois la même table il s'agit d'une auto-jointure dans ce cas l'alias est indispensable.

d- Recherche de données avec Tri :

On peut déterminer l'ordre de tri des enregistrements extraits par l'instruction SELECT.

Syntaxe :

```
Select [DISTINCT] * / Liste_nom_colonne
FROM nom_table1 [alias], nom_table2 [alias]...
[WHERE condition]
```

ORDER BY nom_colonne1 [ASC/DESC][nom_colonne2 [ASC/DESC]... ;

- Les mots **ASC** et **DESC** sont facultatifs, ils indiquent l'ordre de tri qui est par défaut c'est **ASC** (ascendant).
- Les noms **_colonne** qui figurent après le paramètre **ORDER BY** doivent figurer dans la clause **Select**.
- Les éléments **NULL** des colonnes concernées par le tri sont donnés ensemble de manière successive.

Activité 22 :

1- Afficher les noms et les grades des employés par ordre croissant des noms

Solution :

```
select nomemp, grademp
from employe
order by nomemp asc, grademp desc ;
```

e- Recherche tout en utilisant des fonctions de calculs (Fonctions Agrégat) :

Les fonctions agrégat réalisent un calcul sur les enregistrements d'une table :

- **SUM ()** donne la somme des valeurs d'un champ pour tous les enregistrements satisfaisant la condition de la requête. Le champ doit bien sur être de type numérique
- **AVG ()** donne la moyenne des valeurs d'un champ pour tous les enregistrements satisfaisant la condition de la requête
- **MAX ()** donne la valeur la plus élevée d'un champ pour tous les enregistrements satisfaisant la condition de la requête.
- **MIN ()** donne la valeur la plus petite d'un champ pur tous les enregistrements satisfaisant la condition de la requête.
- **COUNT (*)** Renvoie le nombre d'enregistrements satisfaisant la requête.

Activité 23 :

Donner le nombre des employés. Donner la moyenne des salaires.

```
SELECT
```

```
COUNT( *) FROM employe ;
```

Donner le minimum , maximum et la moyenne des salaries

```
SELECT MIN(sal), MAX(sal), AVG(sal) from employe;
```

Trouver les nom des employés dont le salaire est supérieur à la moyenne des salaires

```
Select nomemp From employe
```

```
Where sal > (select avg (sal) From employe);
```

donner les employes et leurs salaries annuels qui sont >100000

```
SELECT nemp, sal * 12 AS "salaire annuel"
```

```
FROM employe
```

```
WHERE sal * 12 > 100000;
```

e.1 La fonction Group By : Cette clause sert à **grouper** les résultats d'une requête. Cela permet une meilleure lisibilité.

Par exemple on voudra avoir la moyenne des salaries des employés par département. Pour cela on tape la requête suivante :

```
Select nodept, AVG (sal) From EMPLOYE Group by nodep ;
```

nodept	AVG (sal)
10	2800
20	1350
30	833.33

On a donc le résultat suivant

Activité :

1) Calculer la moyenne des salaires par département et par grade au sein de chaque département et compter le nombre de personnes.

Réponse :

```
SELECT nodept, grademp, AVG( sal ) , count(*)  
FROM EMPLOYE  
GROUP BY nodept, grademp; (on les groupe d'abord par nodept puis par grade)
```

On a donc le résultat suivant:

nodept	grade	COUNT(*)	AVG (sala)
10	President	1	5000
10	Manager	1	900
10	Analyste	1	2500
20	Manager	1	2000
20	analyste	1	1050
20	secrétaire	1	1000
30	Vendeur	3	833.33

e.2 La fonction Having :

Pour appliquer des restrictions sur ces " sous-tables " on utilise la commande **Having** qui est l'équivalent du **Where** pour les groupes.

Activité :

2) On veut la liste des grades avec leur moyenne de salaire mais seulement s'il y a plus d'une personne qui a ce grade.

Solution :

```
Select grademp, COUNT(*), AVG (sal) From EMPLOYE Group by grademp Having count(*) > 1 ;
```

grade	COUNT(*)	AVG(sala)
vendeur	3	833.33