

Les requêtes Sous MS ACCESS 2007

I. Introduction

Nous savons désormais stocker des informations structurées dans les tables d'une base de données relationnelle. Cette étape franchie, il nous faut maintenant apprendre à **gérer** ces informations, et à **retrouver** celles dont nous avons besoin quand cela s'avère nécessaire

Une base de données est souvent une **mine d'informations**, en particulier dans le domaine économique et financier. Il est très important pour le bon fonctionnement d'une entreprise que ces informations puissent être retrouvées rapidement et simplement par les personnes qui en ont besoin et qui sauront en faire bon usage. Pour ce faire, la **requête** constitue l'outil adéquat.

II. Création d'une requête

Le SGBD Access permet de créer des requêtes en utilisant soit une interface graphique, soit le langage SQL (Structured Query Language.)

A. Création d'une requête on utilisant le langage SQL

Nous avons étudié dans le chapitre 02 « Création d'une base de données Access 2007 » comment créer une base de données sous Microsoft Access 2007. Nous allons maintenant voir comment exploiter la base de données à l'aide de requêtes **SQL**.

SQL est un langage de programmation informatique destiné à **stocker**, à **manipuler** et à **retrouver** des données enregistrées dans des bases de données relationnelles On va prendre comme **exemple** la base de donnée créer dans le chapitre 02 et qui est constituer des deux table **Amis** et **Films**



Faculté des sciences de gestion et sciences économiques Département LMD : 2iem Année LMD

] /	Amis			
	ID 👻	Prénom 👻	Ville 👻	ID Film 👻
+	1	Marina	Riom	1
+	2	Thomas	Vichy	1
+	3	Delphine	Bordeaux	4
+	4	Damien	Paris	5
+	5	Raphael	Paris	6
+	6	Moktar	Clermont-Ferrand	7
+	7	Nicolas	Clermont-Ferrand	7
+	8	J-C	Le Cendre	8
+	9	Eric	Vichy	9
+	10	Vincent	Vichy	4

Films Nom du champ Type de données ID NuméroAuto Nom Texte Genre Texte Durée Numérique Date de sortie Date/Heure Réalisateur Texte Acteurs Texte Note Numérique

- > Ouvrer la base de données Films Amis
- > Dans la barre de menus en haut cliquez sur Créer

- Ensuite allez tout à droite dans la barre d'outils dans la section Autre et cliquez sur Création de requête
- La fenêtre suivante s'ouvre avec les différentes tables de votre base de données, vous allez passer directement en mode SQL pour cela cliquez sur le bouton Fermer :

- > Cliquez maintenant tout en bas à droite sur le bouton SQL :
- > La page principale affiche le résultat suivant :









Verr. num.

SQL 1



> Modifiez le résultat affiché comme ci-dessous :



 Cliquez ensuite sur le bouton Exécuter situé dans la barre d'outil en haut à gauche dans la section Résultats:



 Vous venez d'exécuter votre première requête SQL dont le résultat est le suivant

a	Requête1			
	ID 👻	Prénom 👻	Ville 👻	ID Film 👻
	1	Marina	Riom	1
	2	Thomas	Vichy	1
	3	Delphine	Bordeaux	4
	4	Damien	Paris	5
	5	Raphael	Paris	6
	6	Moktar	Clermont-Ferrand	7
	7	Nicolas	Clermont-Ferrand	7
	8	J-C	Le Cendre	8
	9	Eric	Vichy	9
	10	Vincent	Vichy	4
	11	Guillaume	Paris	1
*	(Nouv.)			

 Vous pouvez enregistrer la requête. Pour cela faites un clic droit sur l'onglet « Requête1" puis un clic gauche sur Enregistrer



Une fenêtre apparait vous demandant d'enregistrer sous le nom que vous souhaitez la requête. Tapez « Liste des Amis » puis cliquez sur le bouton OK :

Enregistrer sous		? ×
Nom de la requête	:	
Liste des Amis		
	ОК	Annuler

> On constate que la requête « Liste des Amis » apparait

dans la liste des tables à gauche et se distingue des

tables par un symbole qui précède son nom représentant

III. Les commandes SQL de base :

des tables liées :

Étudions à présent les commandes SQL de base en prenant des exemples concrets grâce à notre base de données FilmsAmis :

- SELECT & FROM : SELECT « nom du champ » FROM « nom de table » Exemple : SELECT * FROM Amis ;= Sélectionner tous les champs (=colonnes) provenant de la table « Amis »
- 2. DISTINCT : SELECT DISTINCT « nom du champ » FROM « nom de table » Exemple : SELECT DISTINCT Ville FROM Amis; = Sélectionner le champ « Ville » provenant de la table « Amis » mais exclure les doublons ce qui permet d'obtenir uniquement la liste des villes ou vous

		•	
avez	des	amis	:

3. WHERE : SELECT « nom du champ » FROM « nom de table » WHERE « condition » Exemple : SELECT Prénom FROM Amis WHERE Ville = "Clermont-Ferrand"; = Sélectionner le champ « Prénom » provenant de la table « Amis » dont le champ « Ville » = « Clermont-Ferrand » ce qui permet d'obtenir la liste des amis qui vivent à Clermont-Ferrand :

	Amis à Clermont-Ferrand						
	Prénom	-					
	Moktar						
	Nicolas						
*							

Clermont-Ferrand

Le Cendre Paris Riom Vichy 4. AND & OR : SELECT « nom du champ » FROM « nom de table » WHERE « condition simples » {[AND|OR] « condition simples »}

Exemple : SELECT Nom FROM Films WHERE Durée > 190 OR (Durée < 130 AND Durée > 120); = Sélectionner le champ « Nom » de la table « Films » dont le champ

« Durée » est supérieure à 190 ou inférieure à 130 et supérieure à 120 :

į.	Films > 190 min ou entre	e 120 et 130 min
	Nom 👻	
	La Liste de Schindler	
	Le Terminal	
*		

5. IN : SELECT « nom du champ » FROM « nom de table » WHERE « nom de colonne » IN ('valeur1', 'valeur2', ...)

Exemple : SELECT * FROM Films WHERE Genre IN ('Policier', 'Drame'); = Sélectionner tous les champs provenant de la table « Films » dont le champ « Genre » contient « Policier » et « Drame ». On obtient donc toutes les informations concernant les films du genre Policier ou Drame :

	Infos des films de genre Policier et Drame									
	ID ·	- Nom -	Genre 🝷	Durée 🔻	Date de sortie 🕞	Réalisateur 🔹	Acteurs 👻	Note	٠	
		2 Gran Torino	Drame	111	25/02/2009	Clint Eastwood	Clint Eastwood, Bee Vang		5	
		3 Le Parrain	Policier	175	02/08/2006	Francis Ford Coppola	Marlon Brando, Al Pacino		5	
		4 Les Evadés	Drame	140	01/03/1995	Frank Darabont	Tim Robbins, Morgan Freeman		4	
		8 Pulp Fiction	Policier	149	26/10/1994	Quentin Tarantino	John Travolta, Samuel L. Jackson		4	
*	(Nouv	.)								

6. BETWEEN : SELECT « nom du champ » FROM « nom de table » WHERE « nom de colonne » **BETWEEN** 'valeur1' **AND** 'valeur2'

Exemple : SELECT * FROM Films WHERE [Date de sortie] BETWEEN

#1/1/1990# AND #1/1/2000#; (les dates doivent toujours être encadrées par des # sous Microsoft Access) = Sélectionner tous les champs provenant de la table



« Films » dont le champ « Date de sortie » est compris entre le 01/01/1990 et le 01/01/2000 :

	Films des années 90									
	ID	Ŧ	Nom 👻	Genre 👻	Durée 🔻	Date de sortie 🕞	Réalisateur 👻	Acteurs 👻	Note	*
		4	Les Evadés	Drame	140	01/03/1995	Frank Darabont	Tim Robbins, Morgan Freeman		4
		5	La Liste de Schindler	Historique	195	02/03/1994	Steven Spielberg	Liam Neeson, Ben Kingsley		4
		7	Forrest Gump	Comédie dramatique	140	05/10/1994	Robert Zemeckis	Tom Hanks, Gary Sinise		5
		8	Pulp Fiction	Policier	149	26/10/1994	Quentin Tarantino	John Travolta, Samuel L. Jackson		4
*	(Nou	v.)								

7. LIKE : SELECT « nom du champ » FROM « nom de table » WHERE « nom de colonne » LIKE {modèle}

Exemple : SELECT Réalisateur FROM Films WHERE Réalisateur LIKE « Fra* »;

(les caractères manquants sont exprimés par des * sous Microsoft Access) = Sélectionner les enregistrements du champ « Réalisateur » de la table « Films »

dont les enregistrements du champ « Réalisateur » contiennent le début des caractères « Fra » :

Réalisateurs contenant "I	Fra
Réalisateur 👻	
Francis Ford Coppola	
Frank Darabont	
Frank Darabont	

8. ORDER BY : SELECT « nom du champ » FROM « nom de table » [WHERE « condition »] ORDER BY « nom de colonne » [ASC, DESC] Exemple : SELECT Réalisateur FROM Films WHERE Réalisateur LIKE « Fra* » ORDER BY Réalisateur DESC; = Comme précédemment avec la commande SQL LIKE sauf que cette fois ci on a ajouté un tri du champ « Réalisateur » dans un ordre descendant. On constate que l'ordre de la liste est inversé en effet par défaut si on

utilise pas la commande SQL ORDER BY les enregistrements du champ sont classés dans un ordre montant :

-	Praisateurs contenant "Fra" Décroissant						
	Réalisateur 👻						
	Frank Darabont						
	Frank Darabont						
	Francis Ford Coppola						

Somme des Durées

1480

· 148

- 9. Fonctions SUM, AVG, MAX, MIN, COUNT : SELECT « nom de fonction » (« nom du champ ») FROM « nom de table »
 - Exemple : SELECT SUM(Durée) FROM Films; = Calcul de la somme de tous les enregistrements du champ « Durée » de la table « Films » :
 - ✓ Exemple : SELECT AVG(Durée) FROM Films; = Calcul de la moyenne de tous les enregistrements du champ « Durée » de la table « Films » :
 Moyenne des Durées
 - Exemple : SELECT MAX(Durée) FROM Films; = Sélectionner l'enregistrement du champ « Durée » de la table « Films » ayant la plus grande valeur :

Ē	Durée de fil	m la	plus élevée
	Expr1000	*	
		195	

Expr1000

Exemple : SELECT MIN(Durée) FROM Films; = Sélectionner l'enregistrement du champ « Durée » de la table « Films » ayant la plus petite valeur :



✓ Exemple : SELECT COUNT(Nom) FROM Films; = Calcul du nombre d'enregistrements du champ « Nom » de la table « Films » :

1	Nombre de	Film	5
	Expr1000	-	
		10	

10. GROUP BY : SELECT « nom du champ 1", SUM(« nom du champ 2") FROM « nom de table » GROUP BY « nom du champ 1" Exemple : SELECT Réalisateur, SUM(Note) FROM Films GROUP BY Réalisateur; = Sélectionner les enregistrements du champ « Réalisateur » et la somme des enregistrements du champ « Note » de la table « Films » groupés à partir du champ « Réalisateur » :

đ	Somme des Notes groupé par Réalisateurs						
	Réalisateur 👻	Expr1001 -					
	Christopher Nolan	3					
	Clint Eastwood	5					
	Francis Ford Coppola	5					
	Frank Darabont	9					
	Quentin Tarantino	4					
	Robert Zemeckis	5					
	Steven Spielberg	8					
	Wilson Yip	3					

11. HAVING : SELECT « nom du champ 1", SUM(« nom du champ 2") FROM « nom de table » GROUP BY « nom du champ 1" HAVING (condition fonction) Exemple : SELECT Réalisateur, SUM(Note) FROM Films GROUP BY Réalisateur HAVING SUM(Note)>7; = Comme précédemment avec la commande SQL GROUP BY sauf que cette fois ci on a ajouté la condition que la somme des enregistrements du champ « Note » groupé par « Réalisateur » soit supérieur à la note de 7 :

Somme des Notes groupé par Réalisateurs et > à 7				
	Réalisateur 👻	Expr1001 🔹		
	Frank Darabont	9		
	Steven Spielberg	8		



• Alias : SELECT « table alias ». »nom du champ 1" « colonne alias » FROM « nom de table » « table alias » *Exemple : SELECT A1.Réalisateur AS [Nom Réalisateur], SUM(A1.Note) AS [Total Notes] FROM Films AS A1 GROUP BY A1.Réalisateur; (AS est spécifique à Microsoft Access) = Comme précédemment avec la commande SQL GROUP BY sauf que cette fois ci on a créé des alias afin que la colonne « Réalisateur » se nomme « Nom Réalisateur » et la colonne « Expr1001" se nomme « Total Notes » :*

🗐 Somme des Notes groupé par Réalisateurs avec Alias						
	Nom Réalisateur 📼	Total Notes 👻				
	Christopher Nolan	3				
	Clint Eastwood	5				
	Francis Ford Coppola	5				
	Frank Darabont	9				
	Quentin Tarantino	4				
	Robert Zemeckis	5				
	Steven Spielberg	8				
	Wilson Yip	3				

• Jointure Interne : Jointure entre 2 tables

Exemple : SELECT Prénom, Nom FROM Films INNER JOIN Amis ON Films.ID=Amis.[ID Film] WHERE Ville= »Clermont-Ferrand »; (Veillez à ce que le nom d'un champ contenant un espace soit encadré de crochets dans votre syntaxe SQL) = Sélectionner les champs « Prénom » et « Nom » des tables « Films » et « Amis » qui sont jointes dont les champs en relation sont « ID » de la table « Films » et « ID Film » de la table « Amis » et dont les enregistrements du champ « Ville » est « Clermont-Ferrand » :

Amis et Films de Clermont-Ferrand				
1	Prénom	*	Nom	Ŧ
	Moktar		Forrest Gump	
	Nicolas		Forrest Gump	

• Jointure Externe : Contrairement à la jointure interne qui respecte scrupuleusement la relation entre les tables jointes, la jointure externe permet d'aller plus loin en sélectionnant les données d'une des tables qui ne respectent pas le critère de jointure. Afin de vérifier cela ajoutez un enregistrement à la table « Amis ». Par exemple « Laetitia » qui habite à « Clermont-Ferrand » et dont l'ID Film est le « 4 » puis faites un clic droit sur le nom de l'onglet « Amis » puis un clic gauche sur Enregistrer :



Faculté des sciences de gestion et sciences économiques Département LMD : 2iem Année LMD

	Amis			
4	ID 👻	Prénom 👻	Ville 👻	ID Film 👻
	1	Marina	Riom	1
	2	Thomas	Vichy	1
	3	Delphine	Bordeaux	4
	4	Damien	Paris	5
	5	Raphael	Paris	6
	6	Moktar	Clermont-Ferrand	7
	7	Nicolas	Clermont-Ferrand	7
	8	J-C	Le Cendre	٤
	9	Eric	Vichy	9
	10	Vincent	Vichy	4
	11	Guillaume	Paris	1
	12	Laetitia	Clermont-Ferrand	4

Depuis la liste des tables et Requêtes ouvrez la requête « Amis et Films de Clermont-Ferrand » en faisant un clic droit sur la requête puis un clic gauche sur **Ouvrir** ou si vous ne l'avez pas enregistré créez de nouveau la requête précédente : **SELECT Prénom, Nom FROM Films INNER JOIN Amis ON Films.ID=Amis.[ID Film] WHERE Ville= »Clermont-Ferrand »**;

Toutes les tables						
Films	*					
Films : Table						
🗐 Amis et Films de 🤇 🛁	and Farman d					
Durée de film la r	<u>O</u> uvrir					
	<u>M</u> ode création					
Durée de film la p	Exporter >					
📑 🛛 Films > 190 min o 🍓	Co <u>l</u> lecter et mettre à jour les données par messagerie					
Films des années	Renommer					
📑 Infos des films de	Masquer dans ce groupe					
Moyenne des Dui	Supprimer					
📑 Nombre de Films 🐰	Cou <u>p</u> er					
🗊 Réalisateurs cont 🗈	<u>C</u> opier					
🗊 Réalisateurs cont 🖺	C <u>o</u> ller					
🗊 Somme des Durée 😭	Propriétés de l'o <u>bj</u> et					
Somme des Notes grou	upé par Réalisateurs					
Somme des Notes grou	upé par Réalisateurs avec Alias					
Somme des Notes grou	upé par Réalisateurs et > à 7					
Amis	*					
Amis : Table	Amis : Table					
Amis à Clermont-Ferrar	Amis à Clermont-Ferrand					
Amis et Films de Clermo	Amis et Films de Clermont-Ferrand					
📑 Liste des Amis						
Villes ou j'ai des amis						



On constate que « Laetitia » a bien été ajoutée à la liste des Amis habitant à Clermont-Ferrand et qu'elle possède le Film « Les Evadés » :

Amis et Films de Clermont-Ferrand						
4	Prénom	•	Nom	Ŧ		
	Moktar		Forrest Gump			
	Nicolas		Forrest Gump			
	Laetita		Les Evadés			

Supprimez maintenant de la table « Amis » le champ « ID Film » de « Laetitia » qui est « 4 » et n'oubliez pas d'enregistrer la table avant de la refermer puis exécuter de nouveau la requête « Amis et Films de Clermont-Ferrand ». On constate qu'on a de nouveau juste « Moktar » et « Nicolas » car Laetitia n'a plus de critère de jointure :

Amis et Films de Clermont-Ferrand				
	Prénom	*	Nom	Ŧ
	Moktar		Forrest Gump	
	Nicolas		Forrest Gump	

Pour que Laetitia apparaisse dans la requête il faut exécuter une jointure externe :

Exemple : SELECT Prénom, Nom FROM Films RIGHT JOIN Amis ON Films.ID=Amis.[ID Film] WHERE Ville= »Clermont-Ferrand »; (dans les versions antérieures de Microsoft Access 2007 la syntaxe était RIGHT OUTER JOIN si vous l'ajoutez ça fonctionne aussi) = C'est la même requête que pour la jointure interne sauf qu'on a remplacé INNER par RIGHT :

Amis et ou Films de Clermont-Ferrand				
Prénom	*	Nom	Ŧ	
Moktar		Forrest Gump		
Nicolas		Forrest Gump		
Laetita				

Le **RIGHT JOIN** (ou **RIGHT OUTER JOIN**) implique que l'on sélectionne toutes les lignes respectant le critère de jointure, puis on ajoute toutes les lignes de la table « **TableDroite** » qui ont été rejetées car elles ne respectaient pas le critère de jointure.

Le **LEFT JOIN** (ou **LEFT OUTER JOIN**) implique que l'on sélectionne toutes les lignes respectant le critère de jointure, puis on ajoute toutes les lignes de la table « **TableGauche** » qui ont été rejetées car elles ne respectaient pas le critère de jointure.