

*Faculté des sciences de gestion et sciences économiques :Département LMD*

*1<sup>er</sup> Année Master*

*MODULE : INFORMATIQUE*

# **LES REQUETES SOUS MS ACCESS 2007**

*Présenté par :*

*HALAILI Mohamed*

# 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



# I. INTRODUCTION

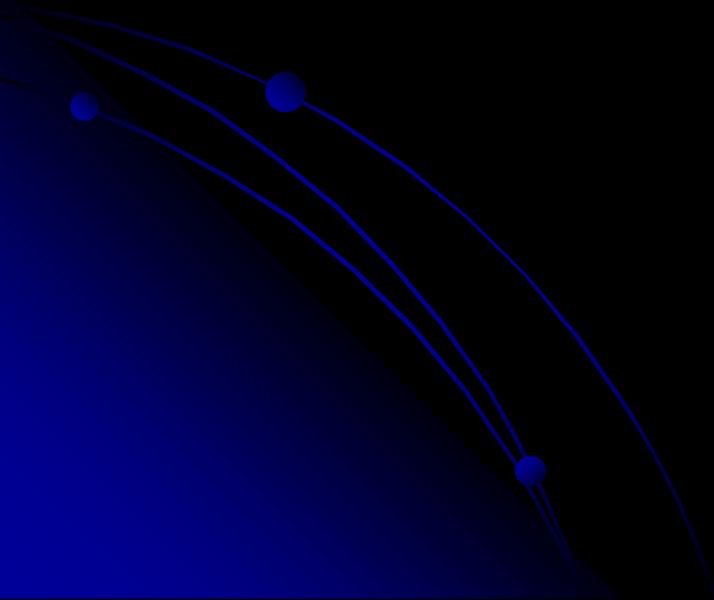
---

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**. 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.)

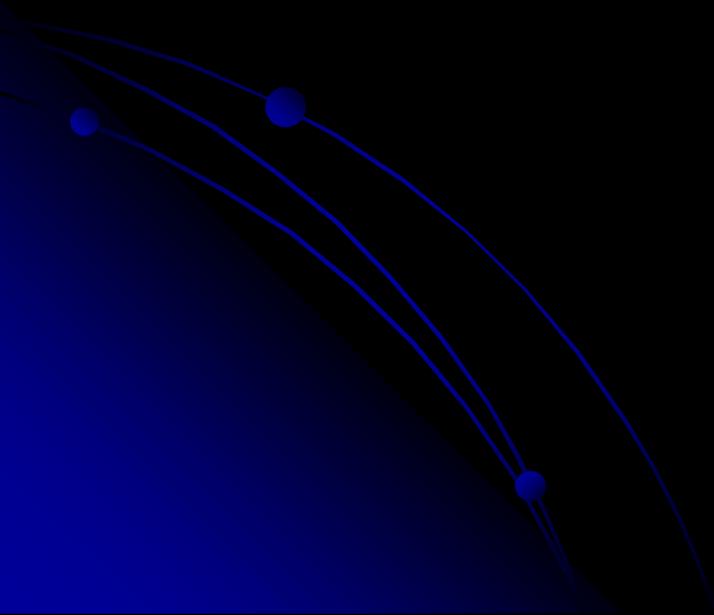


# LA REQUÊTE DANS LE LANGAGE SQL

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** cette base de données à l'aide de **requêtes SQL**.

# LA REQUÊTE DANS LE LANGAGE 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



# LA REQUÊTE DANS LE LANGAGE SQL

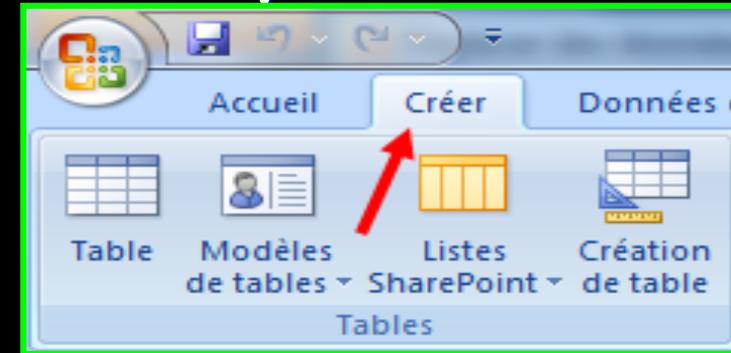
On va prendre comme exemple la base de donnée créer dans le chapitre 02 et qui est constituer des deux table Amis et Films

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

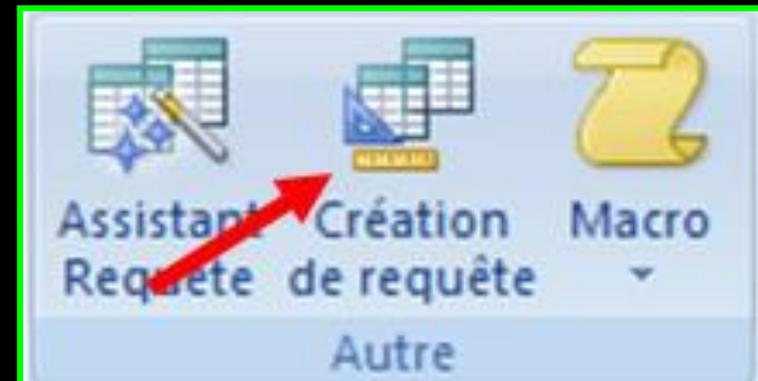
ID	Nom	Genre	Durée	Date de sortie	Réalisateur	Acteurs	Note
1	Ip Man	Arts Martiaux	105	16/01/2008	Wilson Yip	Donnie Yen, Simon Yam	3
2	Gran Torino	Drame	111	25/02/2009	Clint Eastwood	Clint Eastwood, Bee Vang	5
3	Le Parrain	Policier	175	02/08/2006	Francis Ford Coppola	Marlon Brando, Al Pacino	5
4	Les Evadés	Drame	140	01/03/1995	Frank Darabont	Tim Robbins, Morgan Freeman	4
5	La Liste de Schindler	Historique	195	02/03/1994	Steven Spielberg	Liam Neeson, Ben Kingsley	4
6	La Ligne verte	Fantastique	189	01/03/2000	Frank Darabont	Tom Hanks, David Morse	5
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
9	Inception	Science fiction	148	21/07/2010	Christopher Nolan	Leonardo DiCaprio, Marion Cotillard	3
10	Le Terminal	Comédie	128	15/09/2004	Steven Spielberg	Tom Hanks, Catherine Zeta-Jones	4

# LA REQUÊTE DANS LE LANGAGE SQL

- ❑ Ouvrir la base de données **FilmsAmis**
- ❑ Dans la **barre de menus** en haut cliquez sur **Créer**

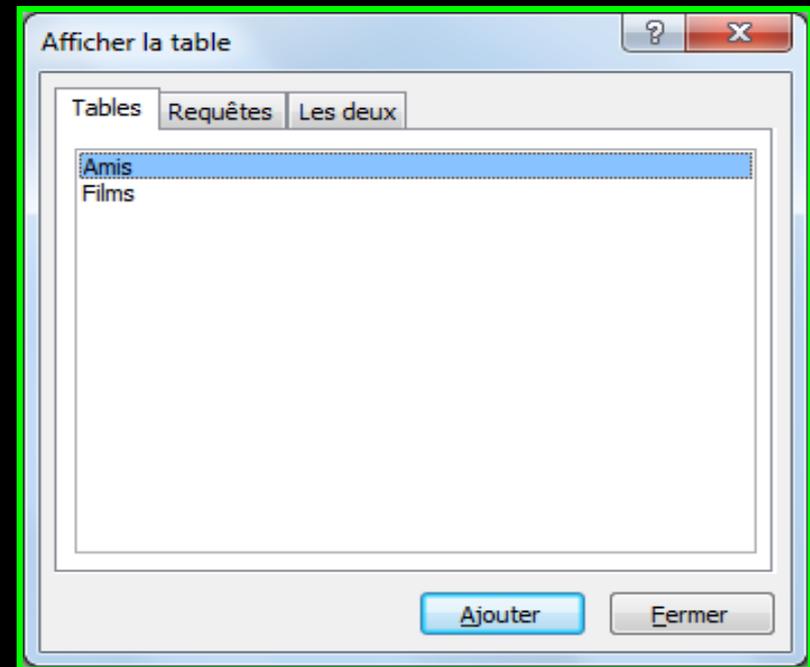


- ❑ Ensuite allez tout à droite dans la barre d'outils dans la section **Autre** et cliquez sur **Création de requête**



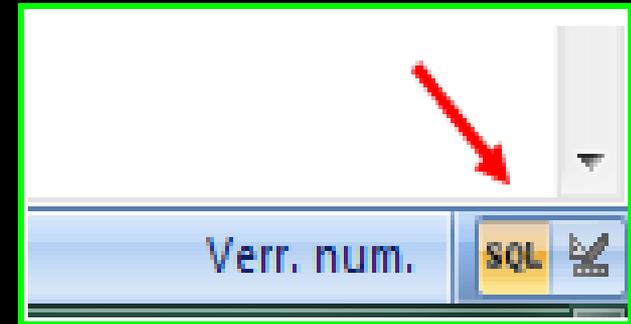
# LA REQUÊTE DANS LE LANGAGE SQL

□ 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** :



# LA REQUÊTE DANS LE LANGAGE SQL

- ❑ Cliquez maintenant tout en bas à droite sur le bouton SQL :



- ❑ La page principale affiche le résultat suivant :



# LA REQUÊTE DANS LE LANGAGE SQL

- ❑ Modifiez le résultat affiché comme ci-dessous

**SELECT \* FROM Amis** (  
Sélectionner tout les champs  
de la table Amis

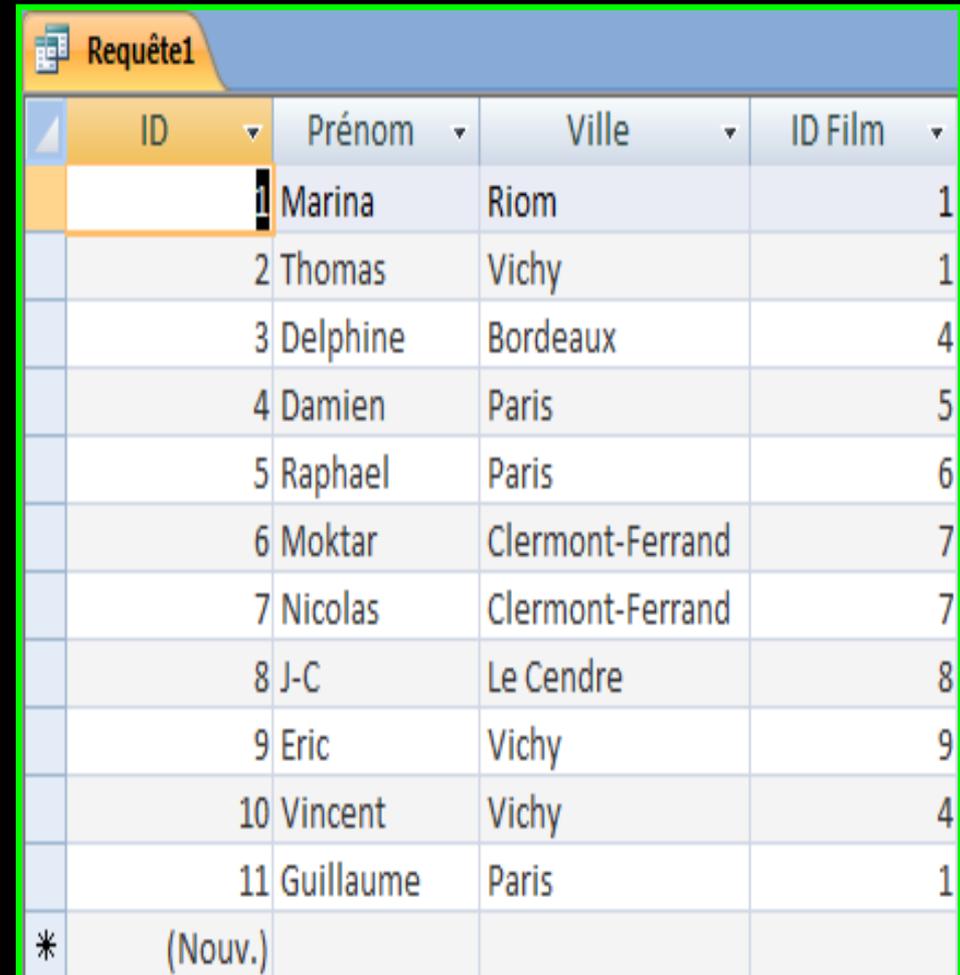


- ❑ Cliquez ensuite sur le bouton **Exécuter** situé dans la **barre d'outil** en haut à gauche dans la section **Résultats**:



# LA REQUÊTE DANS LE LANGAGE SQL

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

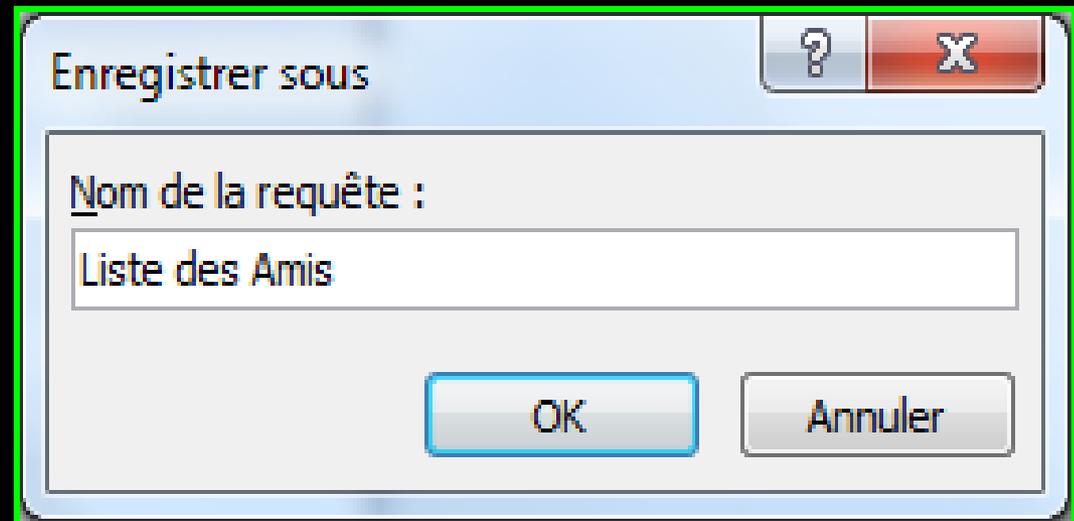


ID	Prénom	Ville	ID Film
1	Marina	Riom	1
2	Thomas	Vichy	1
3	Delphine	Bordeaux	4
4	Damien	Paris	5
5	Raphael	Paris	6
6	Moktar	Clermont-Ferrand	7
7	Nicolas	Clermont-Ferrand	7
8	J-C	Le Cendre	8
9	Eric	Vichy	9
10	Vincent	Vichy	4
11	Guillaume	Paris	1
*	(Nouv.)		



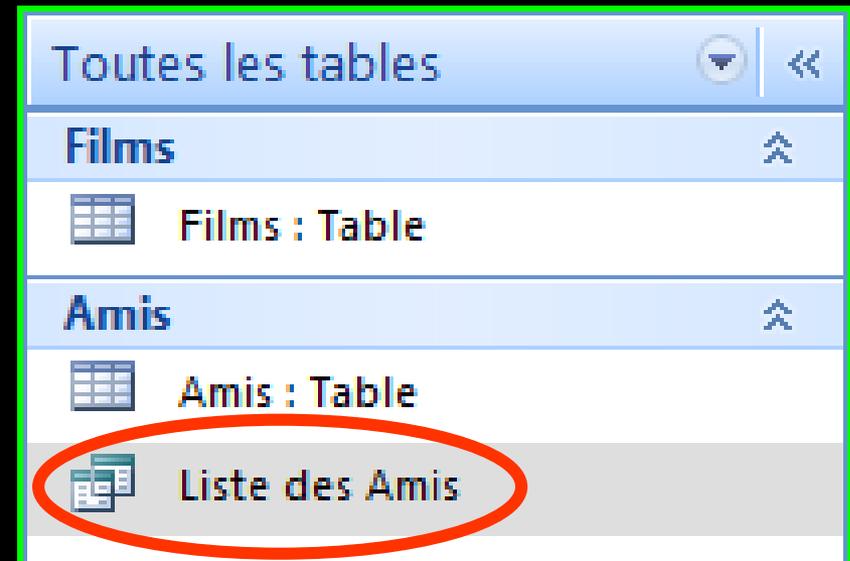
# LA REQUÊTE DANS LE LANGAGE SQL

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



# LA REQUÊTE DANS LE LANGAGE SQL

On constate que la requête « Liste des Amis » apparaît dans la liste des tables à gauche et se distingue des tables par un symbole qui précède son nom représentant des tables liées :



# LES COMMANDES SQL DE BASE

---

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



# LES COMMANDES SQL DE BASE

## 1. SELECT & FROM :

**SELECT** « nom du champ » **FROM** « nom de table »

### Exemple

**SELECT \* FROM** Amis ;= Sélectionner tous les champs (=colonnes) provenant de la table « Amis »

## Exercice

Dans la base de donnée **Gestion** crée :

1. Une **requête** qui affiche tout les champs de la table Clients
2. Une **requête** qui affiche tout les champs de la table Produits

# LES COMMANDES SQL DE BASE

## 2. DISTINCT

**SELECT DISTINCT** « nom du champ » **FROM** « nom de table »

### EXEMPLE

**SELECT DISTINCT** Ville **FROM** Amis;

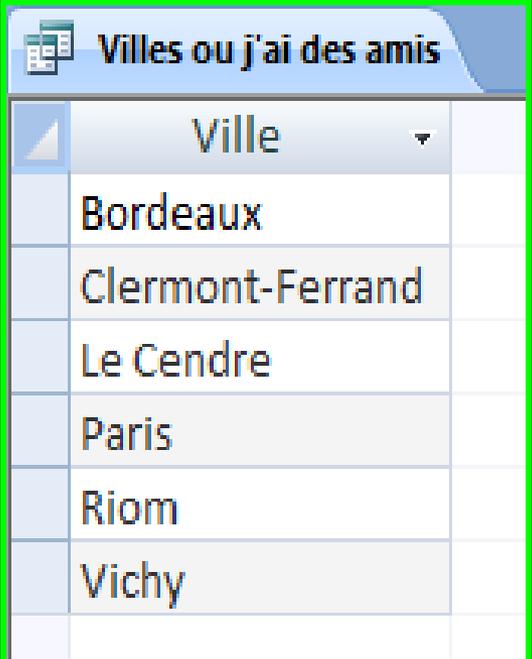
(Sélectionner le champ « Ville »

provenant de la table « Amis » mais

exclure les doublons ce qui permet

d'obtenir uniquement la liste des villes ou

vous avez des amis )



The screenshot shows a query window titled "Villes ou j'ai des amis". The query results are displayed in a table with a single column labeled "Ville". The data rows contain the following cities: Bordeaux, Clermont-Ferrand, Le Cendre, Paris, Riom, and Vichy.

Ville
Bordeaux
Clermont-Ferrand
Le Cendre
Paris
Riom
Vichy

# LES COMMANDES SQL DE BASE

## 2. DISTINCT

**SELECT DISTINCT** « nom du champ » **FROM** « nom de table »

## EXERCICE

Dans la base de donnée **Gestion** crée :

1. Une **requête** qui affiche le champ Ville de la table Clients
2. Une **requête** qui affiche le champ d'éscription la table Produits

# LES COMMANDES SQL DE BASE

## 3. WHERE

**SELECT** « nom du champ » **FROM** « nom de table »

**WHERE** « condition »

### EXEMPLE

**SELECT** Prénom **FROM** Amis **WHERE**

Ville='Clermont-Ferrand'; = Sélectionner le

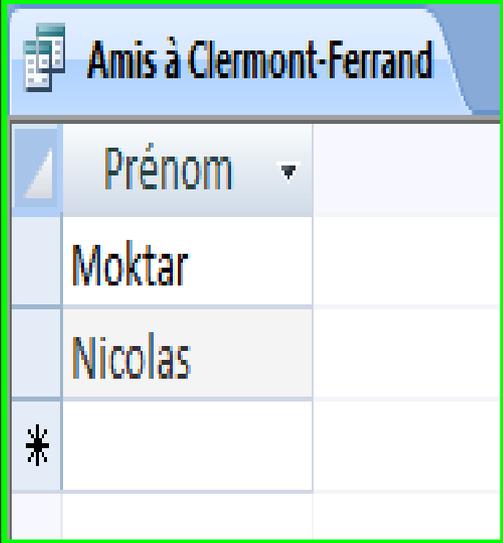
champ « Prénom » provenant de la table

« 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	
*	

# LES COMMANDES SQL DE BASE

## 3. WHERE

**SELECT** « nom du champ » **FROM** « nom de table »

**WHERE** « condition »

### EXERCICE

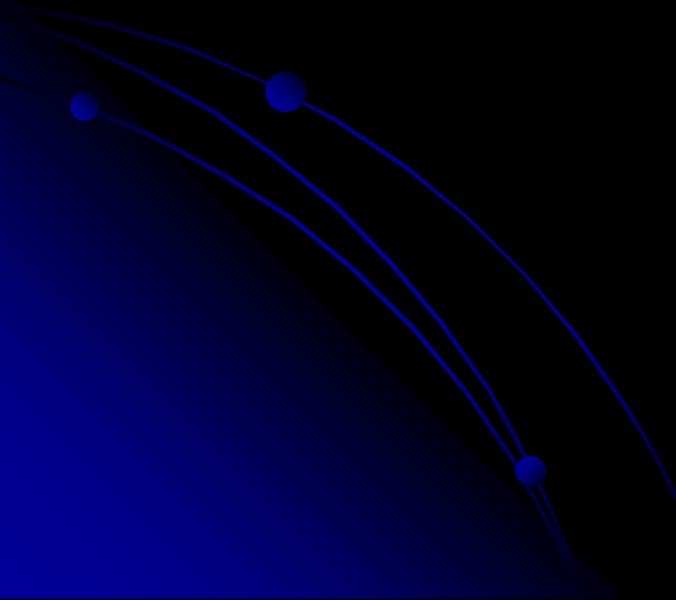
Dans la base de donnée **Gestion** crée :

1. Une **requête** qui affiche tout les noms de la table Clients dont la ville est **CARPENTRA**
2. Une **requête** qui affiche tout les noms de la table Clients dont le nombre de commande=2

# LES COMMANDES SQL DE BASE

## 4. AND & OR

**SELECT** « nom du champ » **FROM** « nom de table »  
**WHERE** « condition simples » **{[AND|OR]** « condition  
simples »}



# LES COMMANDES SQL DE BASE

## 4. AND & OR

**EXEMPLE 01** : Ecrire une requête qui affiche les nom des films dans la durée est comprise entre 110 et 150

```
SELECT Nom FROM Films WHERE Durée >= 110 AND  
Durée >= 150 ;
```

Sélectionner le champ « **Nom** » de la table « **Films** » dont le champ « **Durée** » est **comprise entre 110 et 150** c'est-à-dire **supérieure ou égale** à 110 et **inférieure ou égale** à 150

# LES COMMANDES SQL DE BASE

## 4. AND & OR

Durée >= 110 AND Durée >= 150

ID	Nom	Genre	Durée	Date de sortie	Réalisateur	Acteurs	Note
1	Ip Man	Arts Martiaux	105	16/01/2008	Wilson Yip	Donnie Yen, Simon Yam	3
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
5	La Liste de Schindler	Historique	195	02/03/1994	Steven Spielberg	Liam Neeson, Ben Kingsley	4
6	La Ligne verte	Fantastique	189	01/03/2000	Frank Darabont	Tom Hanks, David Morse	5
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
9	Inception	Science fiction	148	21/07/2010	Christopher Nolan	Leonardo DiCaprio, Marion Cotillard	3
10	Le Terminal	Comédie	128	15/09/2004	Steven Spielberg	Tom Hanks, Catherine Zeta-Jones	4

# LES COMMANDES SQL DE BASE

## 4. AND & OR

**Exemple 02:** Ecrire une requête qui affiche les noms des clients qui on un chiffre d'affaire compris entre 500 et 3000

```
SELECT Nom FROM client WHERE chiffre_dafaire >= 500  
AND chiffre_dafaire <= 3000;
```

N°Client	Nom	Prénom	Adresse	CP	Ville	Date de Nai	Remarque	Nom	Chiffre_dafaire
AA01	DEGREE	ANDRE	LA CARRIE	84410	BEDOIN	12/12/1995	PAIE EN RETAR	2	4 563,00 €
BM01	BALAN	MARC	RUE MOG	84200	CARPENTRA	15/03/1965	RAS	2	1 950,00 €
BM02	BRUN	MARIE	RUE DU CA	84200	CARPENTRA	04/05/1962	RAS	2	2 589,00 €
TA01	TALANT	ANDRE	RUE ROSE	84100	ORANGE	01/02/1952	RAS	1	630,00 €
TE01	THIBAUL	ERIC	CHEMIN D	84000	AVIGNAN	18/09/1958	PAIE EN RETAR	3	3 698,00 €

Chiffre Affaire entre 500 et 3000	
Nom	
BALAN	
BRUN	
TALANT	

# LES COMMANDES SQL DE BASE

## 4. AND & OR

**Exemple 03:** Ecrire une requête qui affiche les référence des produits de la table produits qui on un **Prix Unitaire** supérieure à 3000 ou une **Unité En Stock** supérieure à 5

**SELECT** Ref\_Produit **FROM** Produits **WHERE** PrixUnitaire > 3000 **OR** UniteEnStock > 5;

Ref_Produit	Description du produit	RefFourniss	UnitéEnStock	PrixUnitaire	N°Cli
BP60	BRET PENTIUM 1600	10	5	2 041,29 €	AA01
DC90	DIGITAL CELEBRIS 1000	10	2	3 433,15 €	TA01
EC59	ELONEX PC 800	10	1	2 893,48 €	BM02
EP90	ESCOM PENTIUM 1200	10	2	2 530,65 €	TE01
G200	GATEWAY 1200	10	1	1 353,75 €	BM02
IBM3	IBM GTX	10	4	2 253,20 €	BM02
IBM7	IBM ARTIVAL 1500	10	6	1 294,29 €	BM02
LM500	PC HP	10	10	1 895,50 €	AA01
PB02	PACKARD BELL 1600	10	2	2 285,21 €	AA01



RefProd PU Sup3000 ou US sup 5

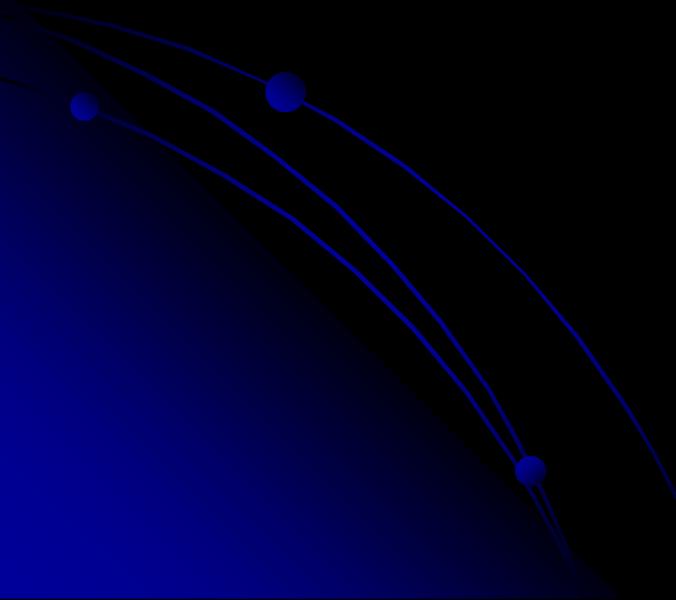
Ref_Produit
DC90
IBM7
LM500

# LES COMMANDES SQL DE BASE

---

## 5. IN

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



# LES COMMANDES SQL DE BASE

## 5. IN

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 :
```

# LES COMMANDES SQL DE BASE

## 5. IN

EXEMPLE :

```
SELECT * FROM Films WHERE Genre IN ('Policier',  
'Drame');
```



Infos des films de genre Policier et Drame

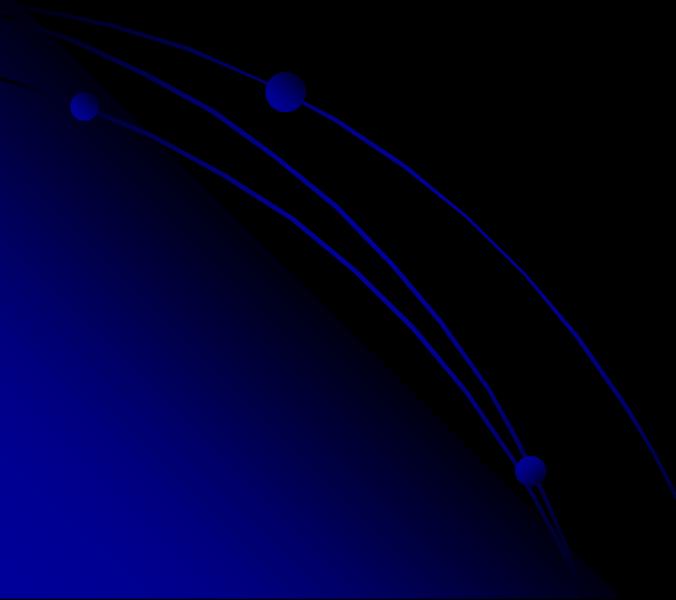
ID	Nom	Genre	Durée	Date de sortie	Réalisateur	Acteurs	Note
2	Gran Torino	Drame	111	25/02/2009	Clint Eastwood	Clint Eastwood, Bee Vang	5
3	Le Parrain	Policier	175	02/08/2006	Francis Ford Coppola	Marlon Brando, Al Pacino	5
4	Les Evadés	Drame	140	01/03/1995	Frank Darabont	Tim Robbins, Morgan Freeman	4
8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

# LES COMMANDES SQL DE BASE

---

## 6. BETWEEN

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



# LES COMMANDES SQL DE BASE

## 6. BETWEEN

**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 :**

# LES COMMANDES SQL DE BASE

## 6. BETWEEN

EXEMPLE :

```
SELECT * FROM Films WHERE [Date de sortie]  
BETWEEN #1/1/1990# AND #1/1/2000#;
```

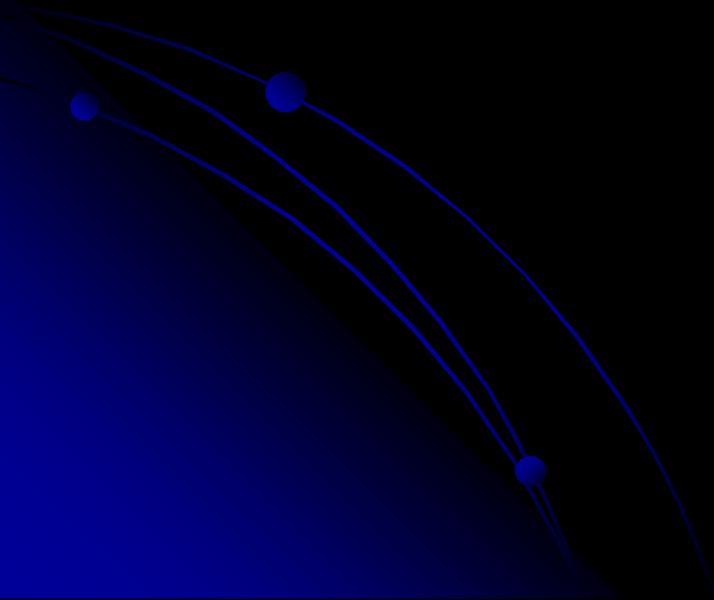
Films des années 90							
ID	Nom	Genre	Durée	Date de sortie	Réalisateur	Acteurs	Note
4	Les Evadés	Drame	140	01/03/1995	Frank Darabont	Tim Robbins, Morgan Freeman	4
5	La Liste de Schindler	Historique	195	02/03/1994	Steven Spielberg	Liam Neeson, Ben Kingsley	4
7	Forrest Gump	Comédie dramatique	140	05/10/1994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

# LES COMMANDES SQL DE BASE

---

## 7. LIKE

**SELECT** « nom du champ » **FROM** « nom de table » **WHERE** « nom de colonne » **LIKE** {modèle}



# LES COMMANDES SQL DE BASE

## 7. LIKE

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 »

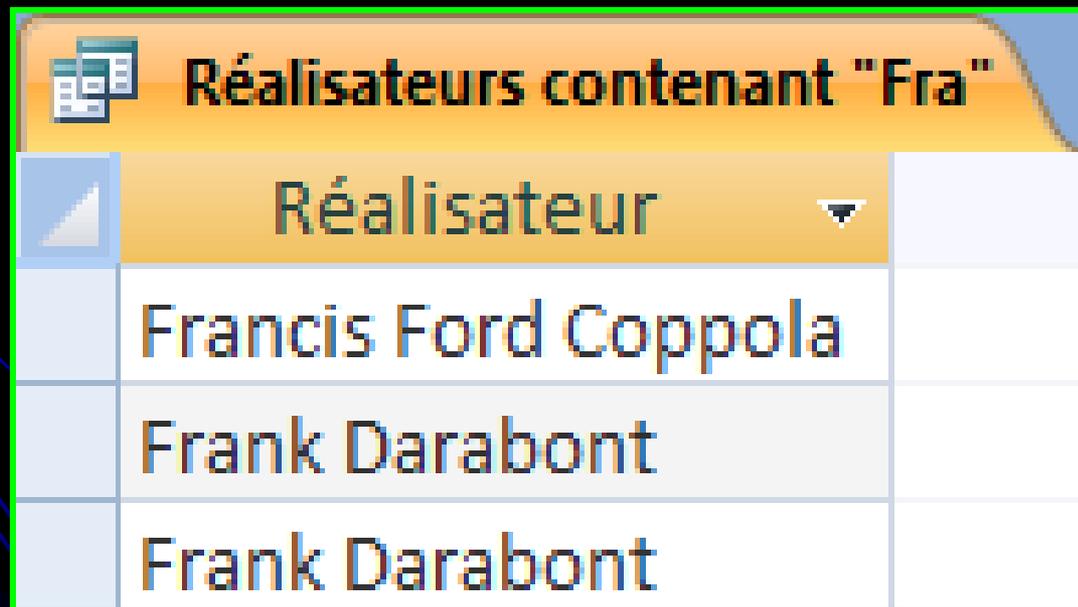
contiennent le début des caractères « Fra » :

# LES COMMANDES SQL DE BASE

## 7. LIKE

EXEMPLE :

```
SELECT Réalisateur FROM Films WHERE  
Réalisateur LIKE « Fra* »;
```



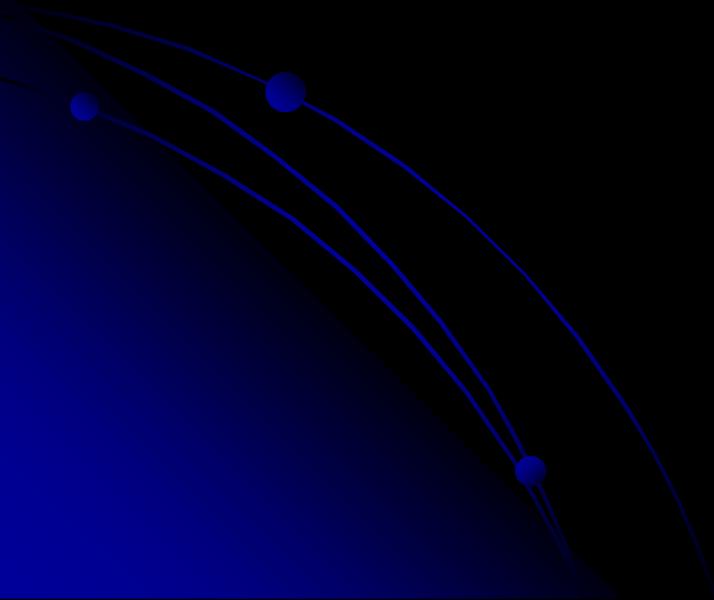
Réalisateurs contenant "Fra"	
Réalisateur	
Francis Ford Coppola	
Frank Darabont	
Frank Darabont	

# LES COMMANDES SQL DE BASE

---

## 8. ORDER BY

**SELECT** « nom du champ » **FROM** « nom de table »  
**[WHERE** « condition »] **ORDER BY** « nom de colonne »  
**[ASC, DESC]**



# LES COMMANDES SQL DE BASE

## 8. ORDER BY

**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**

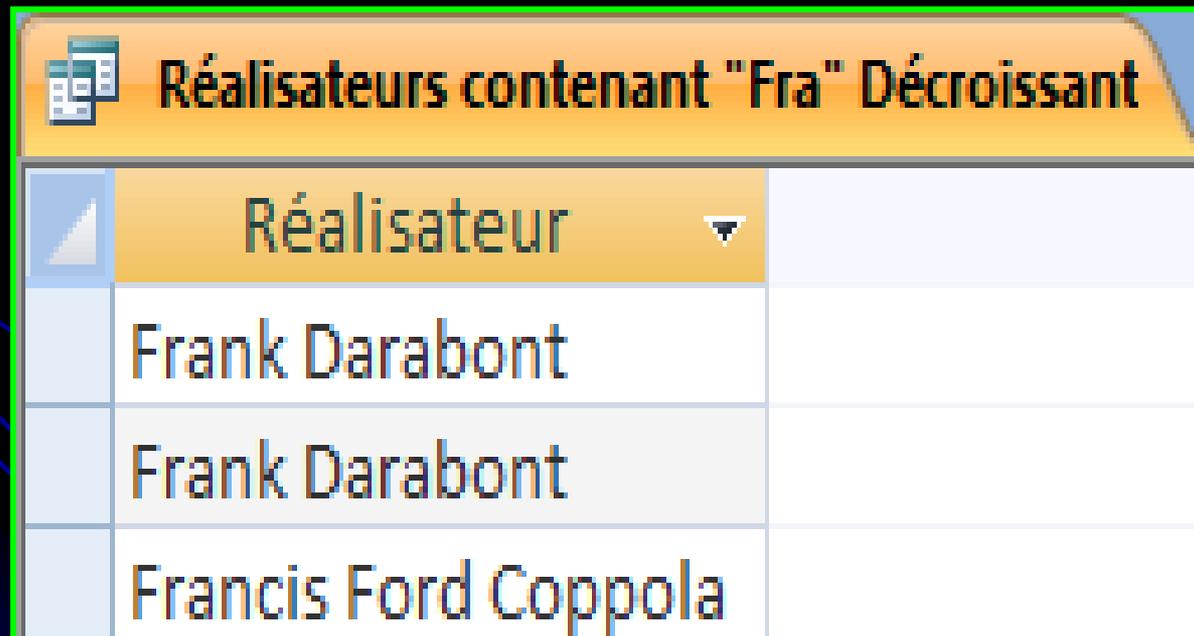
**BY** les enregistrements du champ sont classés dans un

ordre montant :

# LES COMMANDES SQL DE BASE

## 8. ORDER BY

**EXEMPLE : SELECT Réalisateur FROM Films WHERE Réalisateur LIKE « Fra\* » ORDER BY Réalisateur DESC;**



Réalisateurs contenant "Fra" Décroissant

Réalisateur
Frank Darabont
Frank Darabont
Francis Ford Coppola

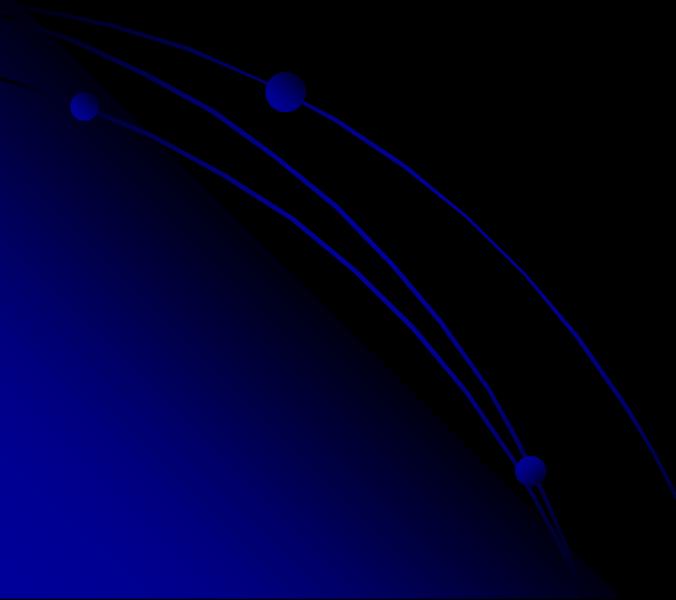
# LES COMMANDES SQL DE BASE

---

## 9. FONCTIONS SUM, AVG, MAX, MIN, COUNT

**SELECT** « nom de fonction » (« nom du champ »)

**FROM** « nom de table »



# LES COMMANDES SQL DE BASE

## 9. FONCTIONS SUM, AVG, MAX, MIN, COUNT

### EXEMPLE 1:

**SELECT SUM(Durée) FROM Films;** = Calcul de la somme de tous les enregistrements du champ « Durée » de la table « Films » :



The screenshot shows a query result window titled "Somme des Durées". It contains a single row with the expression "Expr1000" and the value "1480".

Expr1000
1480

# LES COMMANDES SQL DE BASE

## 9. FONCTIONS SUM, AVG, MAX, MIN, COUNT

### EXEMPLE 2:

**SELECT AVG(Durée) FROM Films;** = Calcul de la moyenne de tous les enregistrements du champ « Durée » de la table « Films »



The screenshot shows a query result window with a yellow header and a table below. The header contains a small icon of a table and the text 'Moyenne des Durées'. The table has two columns. The first column contains the text 'Expr1000' and a small downward-pointing triangle icon. The second column contains the numerical value '148'.

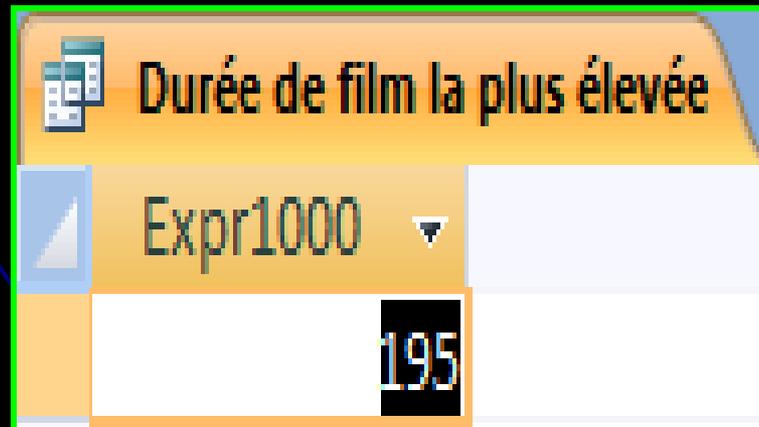
Moyenne des Durées	
Expr1000	148

# LES COMMANDES SQL DE BASE

## 9. FONCTIONS SUM, AVG, MAX, MIN, COUNT

### EXEMPLE 3:

**SELECT MAX(Durée) FROM Films;** = Sélectionner l'enregistrement du champ « Durée » de la table « Films » ayant la plus grande valeur



The screenshot shows a query result window with the title "Durée de film la plus élevée". The query text "Expr1000" is visible in the top bar. The result is displayed in a table with one row and one column, showing the value "195".

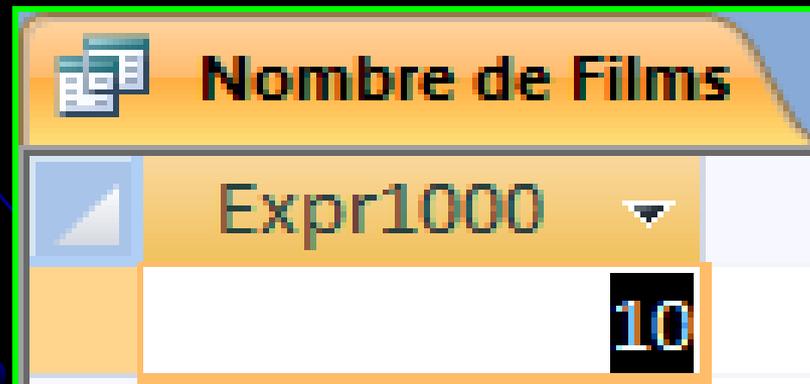
Expr1000
195

# LES COMMANDES SQL DE BASE

## 9. FONCTIONS SUM, AVG, MAX, MIN, COUNT

### EXEMPLE 4:

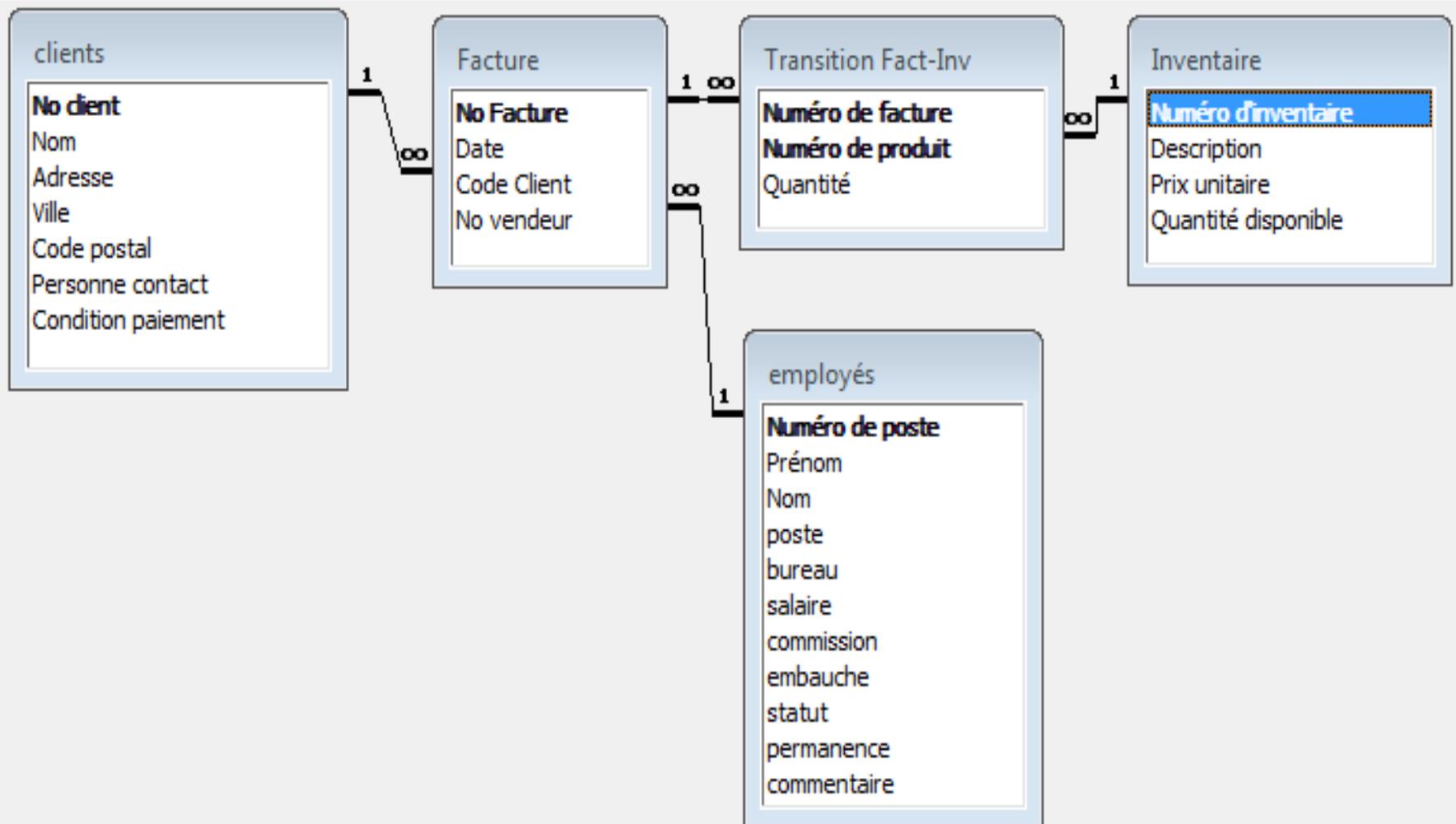
**SELECT COUNT(Nom) FROM Films;** = Calcul du nombre d'enregistrements du champ « Nom » de la table « Films »



The screenshot shows a query result window titled "Nombre de Films". It contains a single row with the expression "Expr1000" and the value "10".

Nombre de Films	
Expr1000	10

# EXERCICE D'APPLICATION



# EXERCICE D'APPLICATION

Ecrire les requêtes SQL qui permettent d'afficher :

1. le prénom et le nom des personnes ayant pour prénom "roger".
2. le prénom et le nom des personnes dont le nom de famille commence par L.
3. le prénom et le nom des personnes dont le salaire est supérieur à 45 000.
4. le prénom et le nom des personnes embauchées en 1993.
5. le prénom et le nom des personnes ayant la permanence trié selon le nom de famille.

# CORRECTION

Ecrire les requêtes SQL qui permettent d'afficher :

1. le prénom et le nom des personnes ayant pour prénom "roger".

```
SELECT prénom, nom FROM employés WHERE prénom  
LIKE "roger" ;
```

2. le prénom et le nom des personnes dont le nom de famille commence par L.

```
SELECT prénom, nom FROM employés WHERE nom LIKE  
« "L*" ;
```

# CORRECTION

3. le prénom et le nom des personnes dont le salaire est supérieur à 45 000.

```
SELECT prénom, nom FROM employés WHERE salaire > 45000 ;
```

4. le prénom et le nom des personnes embauchées en 1993.

```
SELECT prénom, nom FROM employés WHERE embauche=1993 ;
```

# CORRECTION

---

5. le prénom et le nom des personnes ayant la permanence trié selon le nom de famille.

```
SELECT prénom, nom FROM employés WHERE  
permanence=(Date) ORDER BY nom ASC
```

# EXERCICE D'APPLICATION 2

Soit La base de données suivante qui est constituée d'une table Clients et d'une table Produits

N°Client	Nom	Prénom	Adresse	CP	Ville	Date de n	Remarque	Nom	Chiff
AA01	DEDRE	ANDRE	LA CARRIERE	84410	BEDOIN	12/12/1955	PAIE EN RETARD	2	4563
BM01	BALAN	MARC	RUE BAC	84200	CARPENTRAS	15/03/1965	RAS	2	1950
BM02	BRUN	MARIE	RUE DU CANAL	84200	CARPENTRAS	04/05/1962	RAS	2	2589
TA01	TALANT	ANDRE	RUE ROSE	84100	ORANGE	01/02/1952	RAS	1	630
TE01	THIBAUT	ERIC	CHEMIN DU LAC	84000	AVIGNON	18/09/1958	PAIE EN RETARD	3	3698

PRODUITS						
Réf Produit	Description	RéfFourniss	UnitéEnStoc	PrixUnitaire	N°Client	
BP60	BRET PENTIUM		10	5	2 041,29 €	AA01
DC90	DIGITAL CELEBI		10	2	3 433,15 €	TA01
EC59	ELONEX PC 800		10	1	2 893,48 €	BM02
EP90	ESCOM PENTIL		10	2	2 530,65 €	TE01
G200	GATEWAY 1200		10	1	1 353,75 €	BM02
IBM3	IBM GTX		10	4	2 253,20 €	
IBM7	IBM APTIVAL 1		10	6	1 294,29 €	BM02
PB02	PACKARD BELL		10	2	2 285,21 €	AA01

# EXERCICE D'APPLICATION 2

Ecrire les requêtes SQL qui permettent d'afficher :

1. Tous les champs des personnes ayant pour prénom "ANDRE".
2. Le prénom et le nom des personnes ayant pour prénom "ANDRE".
3. le prénom, nom et adresse des personnes dont le nom de famille commence par B.
4. les adresses des personnes dont la ville est CRPENTRAS
5. Le nom des produits dont le prix est supérieur à 2500,00€.
6. Le nom des produits dont le prix est compris entre 2000,00€ et 2500,00€.