

**Objectifs : comprendre une requête en langage SQL, concevoir des requêtes en langage SQL d'interrogation et de modification**

## I. LES REQUÊTES D'INTERROGATION DE LA BASE DE DONNÉES (RAPPELS)

### A. Définitions

#### ▪ Qu'est-ce qu'une requête ?

Une **requête** d'interrogation est une question que l'on pose à une base de données pour en extraire des informations. Elle pourra porter sur une ou plusieurs tables en fonction de l'endroit où sont stockés les renseignements à extraire.

Le résultat de la requête se présentera sous la forme d'une nouvelle table où seront rassemblés les renseignements demandés. Cette interrogation de la base de données est élaborée en utilisant des « outils » appelés « opérateurs relationnels » : **projection, restriction, jointure, tri.**

#### ▪ Quel rôle pour les opérateurs relationnels ?

Le **projection** est une opération qui a pour but d'indiquer les **champs de la table** qui sont retenus.

Le **restriction** est une opération qui consiste à ne retenir que certains enregistrements de la table en fonction d'un ou de plusieurs critères.

Le **jointure** permet le lien de deux tables afin d'utiliser les champs de tables différentes. Si les deux tables ne sont pas reliées directement, il faudra passer par une troisième.

Le **tri** consiste à ordonner les lignes d'une table en fonction d'un ou de plusieurs critères. Pour chacun des critères, on précise si le tri est croissant ou décroissant.

#### ▪ Pourquoi le SQL ?

Pour réaliser des requêtes, on peut utiliser le requêteur graphique du logiciel utilisé. Le requêteur graphique est un **langage visuel** propre à chaque logiciel, ses possibilités sont limitées lors de cas complexes. Les logiciels SGBDR intègrent également un **langage textuel** de programmation de requêtes : le **SQL** (Structured Query language) . Celui-ci est **universel, normalisé** et capable de traiter toutes les difficultés

### B. Structure des requêtes simples

Les requêtes en SQL ont généralement la structure suivante :

**SELECT** liste des champs à afficher (séparés par des virgules)

**FROM** liste des tables utilisées pour la requête (séparés par des virgules)

**WHERE** champ et critère de condition

**AND** champ commun aux tables utilisées

**ORDER BY** champ à trier **ASC** ou **DESC** ;

Les instructions **WHERE** et **AND** sont liées. Elles permettent d'introduire :

▪ les critères de conditions correspondant à l'opérateur de **RESTRICTION**.

▪ les champs communs aux tables utilisées correspondant à l'opérateur de **JOINTURE**.

L'instruction **ORDER BY** permet de demander un tri sur un champ (**ASC** pour croissant et **DESC** pour décroissant).

## II. LES INSTRUCTIONS DE CALCULS ET DE REGROUPEMENT

Diverses fonctions peuvent être intégrées aux requêtes afin d'obtenir des résultats calculés.

### A. Les instructions de calcul

Fonctions	INSTRUCTIONS	Rôle
Nombre	COUNT( )	Cette fonction permet de compter le nombre d'enregistrements d'un champ.
Somme	SUM( )	Calcule la somme des valeurs d'un ou plusieurs champs.
Moyenne	AVG( )	Cette fonction permet d'effectuer la moyenne des valeurs d'un champ.
Minimum	MIN( )	La fonction « Minimum » permet de retenir la plus petite valeur d'un champ.
Maximum	MAX( )	La fonction « Maximum » permet de retenir la plus grande valeur d'un champ.

Remarque : la fonction « **AS [ ]** », permet d'affecter au calcul un nom représentatif, qui peut être réutilisé dans les autres instructions de la requête.

Exemple : **SELECT SUM(nbheures) AS [Total]**

### B. Les instructions de regroupement

Il est possible de réaliser des opérations par groupe d'enregistrements.

#### ▪ La clause GROUP BY

La clause **GROUP BY** permet de faire porter les fonctions de calculs sur des groupes d'enregistrements.

**SELECT**    *Champs de regroupement, fonction (champ) AS [nom de l'attribut calculé]*  
**FROM**     *TABLES*  
**WHERE**    *Critère de jointure et/ou condition de restriction*  
**GROUP BY** *attributs de regroupement ;*

#### ▪ La clause HAVING

La clause **HAVING** vient après la clause **GROUP BY**. Elle permet de ne retenir que certains des enregistrements obtenus après le regroupement en appliquant un critère de restriction sur les groupes d'enregistrements.

**SELECT**    *Champs de regroupement, fonction ( ) AS [ ]*  
**FROM**     *TABLES*  
**WHERE**    *critère de jointure et/ou condition de restriction*  
**GROUP BY** *attributs de regroupement*  
**HAVING**    *condition de restriction sur le champ calculé ;*

Une instruction **HAVING** s'utilise donc conjointement à une instruction **GROUP BY**. Par contre, **GROUP BY** peut s'utiliser sans l'instruction **HAVING**.

### III. LES REQUETES DE MODIFICATION DE LA BASE DE DONNEES

#### A. L'ajout de données

La clause **INSERT** permet d'ajouter une ou plusieurs lignes dans une table.

**INSERT INTO** *TABLE (valeur des champs en respectant le même ordre que celui de la table)*  
**VALUES** *(valeurs à insérer) ;*

#### B. La suppression de données

La clause **DELETE** permet de supprimer une ou plusieurs lignes dans une table

**DELETE** *Nom des champs à supprimer*  
**FROM** *TABLES*  
**WHERE** *clé primaire de l'enregistrement qui doit être supprimé ;*

#### C. La modification de données

La clause **UPDATE** permet de modifier (mettre à jour) les données contenues dans la base.

**UPDATE** *TABLE*  
**SET** *champ = « valeur »*  
**WHERE** *clé primaire de l'enregistrement qui doit être modifié ;*

L'entreprise « Aux saveurs BIO » décide d'informatiser la gestion de ses commandes. Elle vous demande d'effectuer plusieurs travaux.

Vous trouverez ci-dessous un extrait des données de la base implantée sur un SGBDR

CATEGORIES	
Codecat	Nomcat
ALIM	Alimentaire
BEAU	Beauté
MAIS	Maison
PARA	Parapharmacie

ARTICLES			
Numart	Nomart	Prixunit	Codecat
1	Crème hydratante	22,10	BEAU
2	Soin des lèvres	14,69	BEAU
3	Café expresso	4,26	ALIM
4	Bronchisirop	8,48	PARA
5	Mincivit	11,37	PARA
6	Riz long	2,56	ALIM
7	Chocolat noir	6,45	ALIM
8	Lessive	11,66	MAIS

CLIENTS						
Numcli	Nomcli	Prénomcli	Ruecli	CPcli	Villecli	Moderégl
C1	Dastugue	Eric	3, impasse des mimosas	06 150	Cannes	Chèque
C2	Rossi	Marcelle	17, rue du Baradat	65 500	Vic en Bigorre	Chèque
C3	Mota Vale	Christophe	128, bld de Mun	91 390	Morsang sur orge	Carte bancaire
C4	Lemaini	Farida	24, allée des Mimosas	94 230	Cachan	Carte bancaire

COMMANDES		
Numcde	Datecde	Numcli
BC1031	14/10/06	C2
BC1032	14/10/06	C4
BC1033	16/10/06	C1
BC1034	21/10/06	C3
BC1035	22/10/06	C2

CONCERNER		
Numcde	Numart	Quantité
BC1031	3	4
BC1031	7	6
BC1031	8	2
BC1032	1	1
BC1032	2	1
BC1033	3	5
BC1034	6	4
BC1034	8	2
BC1035	4	1

- **Elaborer le modèle relationnel**

▪ **Réaliser les requêtes suivantes :**

**Requête 1 :** afficher le numéro, le nom et la ville des clients.

Requête1 : Requête Sélection			
	Numcli	Nomcli	Villecli
▶	C1	Dastugue	Cannes
	C2	Rossi	Vic en Bigorre
	C3	Mota Vale	Morsang sur orge
	C4	Lemaini	Cachan
	*		

```
SELECT Numcli, Nomcli, Villecli
FROM CLIENTS ;
```

**Requête 2 :** afficher les numéros, noms, prix de l'ensemble des articles dont le prix est inférieur ou égal à 10 €.

Requête2 : Requête Sélection			
	Numart	Nomart	Prixunit
	3	Café expresso	4,26 €
	4	Bronchisirop	8,48 €
	6	Riz long	2,56 €
	7	Chocolat noir	6,45 €
▶			0,00 €

```
SELECT Numart, Nomart, Prixunit
FROM ARTICLES
WHERE Prixunit <= 10 ;
```

**Requête 3 :** afficher la liste des articles (noms et numéros) de la catégorie « Beauté »

Requête3 : Requête Sélection		
	Numart	Nomart
	1	Crème hydratante
▶	2	Soin des lèvres

```
SELECT Numart, Nomart
FROM ARTICLES, CATEGORIES
WHERE ARTICLES.Codecat = CATEGORIES.Codecat
AND Nomcat = « beauté » ;
```

**Requête 4 :** ensemble des articles (numéro, nom et prix unitaire) du bon de commande BC1031. Les articles sont classés du plus cher au moins cher.

Requête4 : Requête Sélection			
	Numart	Nomart	Prixunit
	8	Lessive	11,66 €
	7	Chocolat noir	6,45 €
▶	3	Café expresso	4,26 €

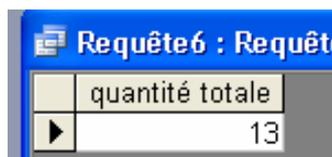
```
SELECT ARTICLES.Numart, Nomart, Prixunit
FROM ARTICLES, CONCERNER
WHERE ARTICLES.numart = CONCERNER.numart
AND Numcde = "BC1031"
ORDER BY Prixunit DESC ;
```

**Requête 5 :** afficher le nombre d'articles de la catégorie « alimentaire »

Requête5 : Requête	
	Nombre d'articles
▶	3

```
SELECT COUNT (Numart) AS [Nombre d'articles]
FROM ARTICLES, CATEGORIES
WHERE ARTICLES.Codecat = CATEGORIES.Codecat
AND Nomcat = "alimentaire" ;
```

**Requête 6** : quelle est la quantité totale des articles commandés par le client Rossi ?



quantité totale
13

```
SELECT SUM(Quantité) AS [quantité totale]
FROM CONCERNER, COMMANDES, CLIENTS
WHERE CONCERNER.Numcde = COMMANDES.Numcde
AND COMMANDES.Numcli = CLIENTS.Numcli
AND Nomcli = " Rossi" ;
```

**Requête 7** : quel est le prix moyen des articles proposés ?



Prix moyen
10,1962 €

```
SELECT AVG (Prixunit) AS [prix moyen]
FROM ARTICLES ;
```

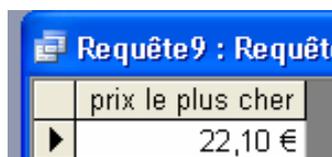
**Requête 8** : quel est l'article le moins cher ?



prix le moins cher
2,56 €

```
SELECT MIN (Prixunit) AS [prix le moins cher]
FROM ARTICLES ;
```

**Requête 9** : quel est l'article le plus cher



prix le plus cher
22,10 €

```
SELECT MAX (Prixunit) AS [prix le plus cher]
FROM ARTICLES ;
```

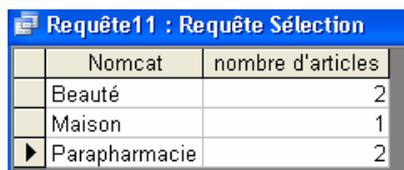
**Requête 10** : quel est le nombre d'articles par catégorie ?



Nomcat	Nombre d'articles
Alimentaire	3
Beauté	2
Maison	1
Parapharmacie	2

```
SELECT Nomcat, COUNT (Numart) AS [Nombre d'articles]
FROM ARTICLES, CATEGORIES
WHERE ARTICLES.Codecat = CATEGORIES.Codecat
GROUP BY Nomcat ;
```

**Requête 11** : Les catégories d'articles qui contiennent strictement moins de 3 articles



Nomcat	nombre d'articles
Beauté	2
Maison	1
Parapharmacie	2

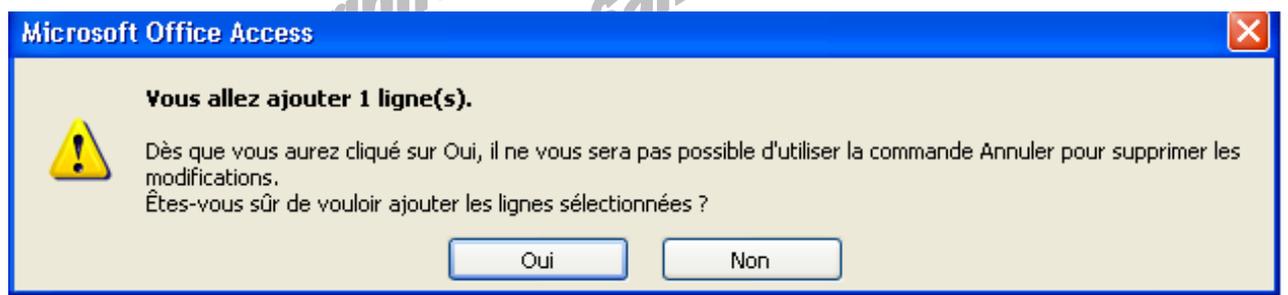
```
SELECT Nomcat, COUNT (Numart) AS [nombre d'articles]
FROM ARTICLES, CATEGORIES
WHERE ARTICLES.Codecat = CATEGORIES.Codecat
GROUP BY Nomcat
HAVING COUNT (Numart) <3 ;
```

**Requête 12** : liste des clients dont la quantité totale des articles commandés le 14/10/06 est supérieure à 10.

Requête12 : Requête Sélection			
	numcli	Nomcli	Quantite commandée
▶	C2	Rossi	12

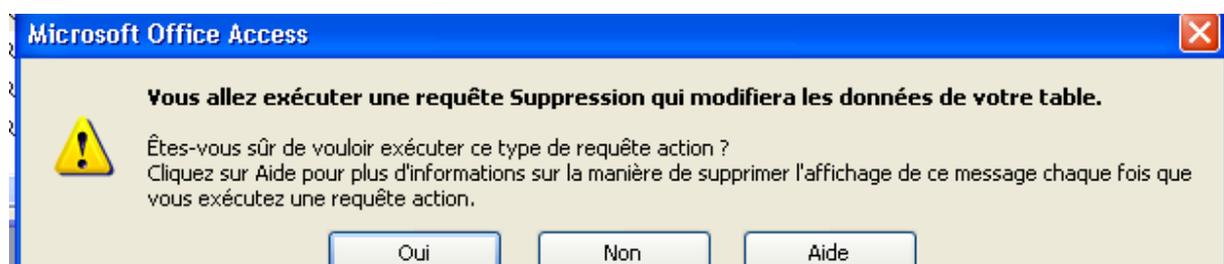
```
SELECT CLIENTS.numcli, Nomcli, SUM (Quantité) AS [Quantite commandée]
FROM CLIENTS, COMMANDES, CONCERNER
WHERE CLIENTS.Numcli = COMMANDES.numcli
AND COMMANDES.Numcde = CONCERNER.Numcde
AND Datecde = #14/10/2006#
GROUP BY CLIENTS.Numcli, Nomcli
HAVING SUM(Quantité)>10;
```

**Requête 13** : l'entreprise souhaite ajouter le client Bouvil Youri habitant 13 allée de la Garenne 33127 Martignas et réglant par carte bancaire.



```
INSERT INTO CLIENTS (Numcli, Nomcli, Prénomcli, Ruecli, CPcli, villecli, Moderègl)
VALUES (« C5 », « Bouvil », « Youri », « 13 allée de la garenne », « 33127 », « Martignas »,
« carte bancaire ») ;
```

**Requête 14**: L'article 8 n'est plus commercialisé.



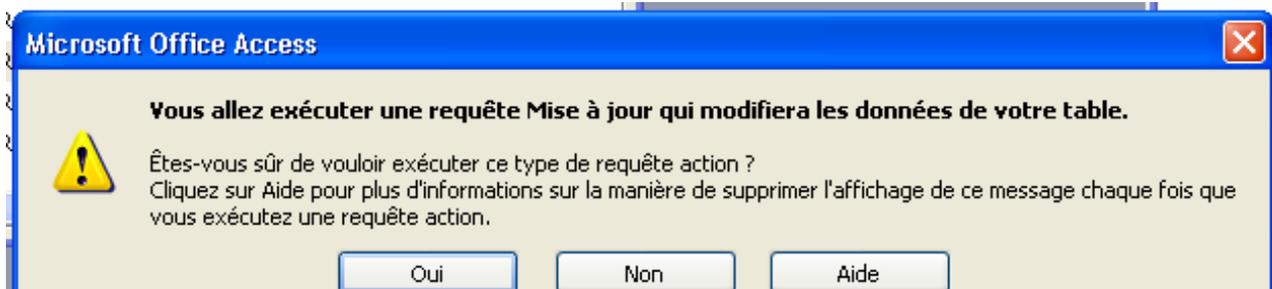
```
DELETE Numart, Nomart, Prixunit, Codecat
FROM ARTICLES
WHERE Numart = « 8 » ;
```

**Requête 15** : le client C2 a changé d'adresse. Nouvelle adresse : 257 rue de Paris 95150 Taverny.

**UPDATE CLIENTS**

**SET Ruecli = « 257 rue de paris », CPcli = « 95150 », Villecli = « Taverny »**

**WHERE Numcli = « C2 » ;**



**Requête 16** : les prix des produits de la catégorie BEAU doivent être tous augmentés de 10 %.

ARTICLES : Table				
	Numart	Nomart	Prixunit	Codecat
+ 1		Crème hydratante	24,31 €	BEAU
+ 2		Soin des lèvres	16,16 €	BEAU
+ 3		Café expresso	4,26 €	ALIM
+ 4		Bronchisirop	8,48 €	PARA
+ 5		Mincivit	11,37 €	PARA
+ 6		Riz long	2,56 €	ALIM
+ 7		Chocolat noir	6,45 €	ALIM
+ 8		Lessive	11,66 €	MAIS
▶			0,00 €	

**UPDATE ARTICLES**

**SET Prixunit = Prixunit\*1.1**

**WHERE Codecat="BEAU";**