

# **BASE DE DONNEES SQL**

## **LANGAGE STRUCTURE DE REQUÊTES**

---

(STRUCTURED QUERY LANGUAGE)

# 1. INTRODUCTION

## 1.1. Présentation de SQL

---

SQL (Structured Query Language) est un langage structuré de requêtes sur les bases de données relationnelles.

Il a été conçu par IBM dans les années 70. Il est devenu le langage standard des systèmes de gestion de bases de données (SGBD) relationnelles (SGBDR).

SQL est complet pour la gestion d'une base de données car il regroupe les langages suivants :

- un **langage d'interrogation des données (LID)** (SELECT...).  
Consultation des données, à travers des projections, restrictions.
- un **langage de définition des données (LDD)** (CREATE...) :  
Création des objets de la base de données. Exemples : tables, index, vues...
- un **langage de contrôle de l'accès aux données (LCD)** (GRANT,REVOKE)  
Définition des droits, des profils d'accès aux objets de la base.
- un **langage de manipulation des données (LMD)** (UPDATE, INSERT, DELETE)  
Mise à jour, insertion, suppression de données des tables.

Le langage SQL est utilisé par les principaux SGBDR du marché:

- SQL SERVER 2000, 2005,
- DB2,
- Oracle,
- Informix,
- Ingres,

NB : il faut noter certaines différences dans les syntaxes proposées par chacun des éditeurs de SGBD.

## 2. LE LANGAGE D'INTERROGATION DES DONNEES

Le langage d'interrogation a pour vocation de consulter à travers des requêtes SQL les données présentes dans une base de données.

Ceci est envisageable dès lors que l'utilisateur dispose des droits d'accès aux tables (structures physiques qui contiennent les données). La définition des droits d'accès est réalisée par l'administrateur de la base de données avec le langage de contrôle des données.

Une base de données met en œuvre des tables de données.

Ci-dessous le contenu des tables de notre base de données commerciale relative à la vente d'articles vestimentaires.

**Table ARTICLE**

	NumArt	Designation	PrixHT	NumTyp
+	JOG0943	TEO	43 J	
+	JOG4390	RAA	35 J	
+	JOG9094	ISIS	56 J	
+	PAN0909	BERMUDE	67 P	
+	PAN8934	CANARIE	24 P	
+	PYJ0909	TACOMA	23 Y	
+	PYJ0910	RAMSES	16 Y	
+	PYJ34095	OSIRIS	25 Y	
+	SHO00843	TRAVELLING	16 S	
+	SHO1245	KHEOPS	27 S	
+	SHO2345	PRINCE	25 S	

**Table CLIENT (extrait)**

numClt	RaisonSociale	adresseClt	CP	tel	fax
45	Lycée Elysée Reclus	19, porte du pommier	16000	05 45 4x xx xx	05 45 4x xx xx
77	Collège Mitterrand	18 rue du Cimetière	16200	05 45 4x xx xx	05 45 4x xx xx
245	Transports Rioux SA	30, chemin de St Jean	16200	05 45 xx xx xx	05 45 xx xx xx
258	Jarnac Télé SAV	26, place du clochet	16200	05 45 xx xx xx	05 45 xx xx xx
50	Poisson Occaz	34 bis, porte des peupliers	16300	05 45 xx xx xx	05 45 xx xx xx
272	Tout-Miroiterie SARL	19 ter, rue de St Joseph	16300	05 45 xx xx xx	05 45 xx xx xx
273	Goal-VU SA	62, boulevard de la marée	16300	05 45 xx xx xx	05 45 xx xx xx
1270	Lycée des Pommiers	94, rue des Pommiers	16300	05 45 4x xx xx	05 45 4x xx xx
107	Mambrade EURL	5 bis, porte de Ste Yolande	16400	05 45 xx xx xx	05 45 xx xx xx
302	Thoumain Max	25 bis, passage des cailloux	16400	05 45 xx xx xx	05 45 xx xx xx
986	Korrigan Services	43, avenue Ghandi	16400	05 49 xx xx xx	05 49 xx xx xx
280	Demelé Assistance	71, voie des libertés individuelles	16500	05 45 xx xx xx	05 45 xx xx xx
260	Bon-Garage Ouest	83 ter, chemin du paradis	16600	05 45 xx xx xx	05 45 xx xx xx
262	Raimon 24/24	31, avenue des fleurs	16600	05 45 xx xx xx	05 45 xx xx xx
338	Barsau SA	5 bis, rue de la partition	16600	05 49 xx xx xx	05 49 xx xx xx
657	Fromage de A à Z	68, rue des saules	16600	05 45 xx xx xx	05 45 xx xx xx

**Table COMMANDE**

NumCde	DateCde	NumClt	ModReg
1	04/01/2000	577	CHQ
3	08/05/2000	41	ESP
4	19/02/2000	127	LCR
5	12/02/2002	1066	CHQ
6	22/02/2002	77	LCR
7	24/11/2002	302	ESP
8	12/05/2005	1207	CHQ
9	11/01/2006	1207	LCR
10	12/03/2007	1066	ESP
11	12/03/2007	1207	LCR
12	03/05/2007	107	CHQ
13	12/09/2007	77	LCR
14	23/09/2007	41	CHQ
16	10/02/2008	279	ESP
17	12/01/2008	279	LCR
18	18/01/2008	577	ESP

**Table LIGNE\_CDE (extrait)**

NumCde	NumArt	QtiteArt
1	JOG0943	25
1	JOG4390	2
1	JOG9094	2
1	PAN0909	3
1	PAN8934	8
1	PYJ0909	3
1	PYJ0910	10
1	PYJ34095	4
1	SHO1245	4
1	SHO2345	4
3	JOG0943	3
3	JOG4390	19
3	JOG9094	6
3	PAN0909	9
3	PAN8934	24
3	PYJ34095	12
3	SHO1245	12

**Table MODE\_REGLEMENT**

ModReg	Libellé
CB	Carte bleu
CHQ	CHEQUE
ESP	ESPECE
LCR	LCR

**Table TYPE\_ARTICLE**

NumTyp	Libelle
	Jeune
P	Pyjama
S	Sportwear
Y	Détente

**Table VILLE (extrait)**

CP	ville
16000	ANGOULEM
16100	ST LAURE
16200	JARNAC
16300	VIGNOLLE
16400	LA COURO
16500	ST MAURI
16600	TOUVRE
16700	VILLEGAT
17000	VILLENEU
17100	ROCHEFOR
17110	ST GEORG
17130	MONTENDR
17200	ROYAN
17230	CHARRON
17240	ST GERMA

**Question :** En vous appuyant sur ces tables, qui a passé la commande n°6 ?

Le modèle logique de données (ou schéma relationnel) est une représentation de la structure de la base de données :

**CLIENT** (NumClt, RaisonSociale, AdresseClt, Cp, Tel, Fax)

*NumClt* : clé primaire

*Cp* : clé étrangère en référence à *Cp* dans la table **VILLE**

**COMMANDE** (NumCde, DateCde, NumClt, ModReg)

*NumCde* : clé primaire

*NumClt* : clé étrangère en référence à *NumClt* dans la table **CLIENT**

*ModReg* : clé étrangère en référence à *ModReg* dans la table **CLIENT**

**ARTICLE** (NumArt, Designation, PrixHT, NumTyp)

*NumArt* : clé primaire

*NumTyp* : clé étrangère en référence à *NumTyp* dans la table **TYPE\_ARTICLE**

**LIGNE\_CDE** (NumCde, NumArt, QtiteArt)

*NumCde, NumArt* : clé primaire

*NumCde* : clé étrangère en référence à *NumCde* dans la table **COMMANDE**

*NumArt* : clé étrangère en référence à *NumArt* dans la table **ARTICLE**

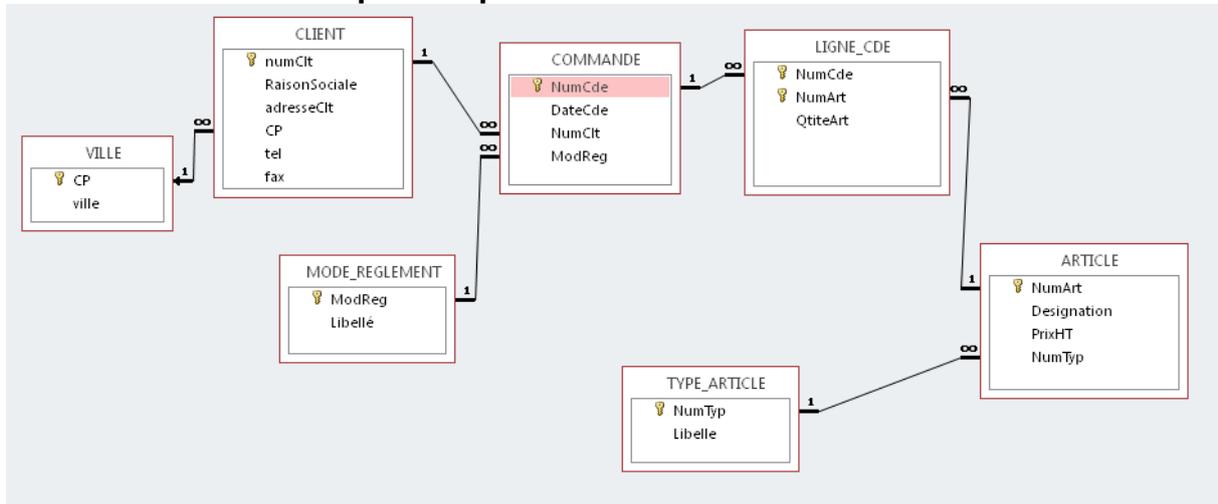
**MODE\_REGLEMENT** (ModReg, Libellé)

*ModReg : clé primaire*

**VILLE** (Cp, ville)

*Cp : clé primaire*

Ce schéma relationnel peut se présenter de la manière suivante :



L'interrogation utilise trois d'opérations fondamentales :

1. La projection,
2. la restriction,
3. la jointure.

## 2.1. Les opérations fondamentales : projection, restriction, jointure

### 2.1.1. La projection

La projection consiste à choisir une ou plusieurs colonnes d'une table à afficher.

#### TABLE CLIENT

NumClt	RaisonSociale	AdresseClt	Cp
23459	Dupont SARL	12 rue d'Alsace	49000
23460	SA Delta Pièces	24 avenue du 8 mai	13000
23461	Dubois et fils	6 impasse fontval	42300
23462	Palermo srl	345 bd St Germain	75000
23463	Laplace SARL	23 impasse des guerrins	42000
23464	Delporte SA	12 rue du Gal de Gaulle	44000
23465	Poitevin Frères	19 rue de la victoire	49000

Elle est réalisée avec la clause **SELECT** :

```
SELECT <Liste des attributs>  
FROM <tables>
```

Pour afficher le numéro des clients, la raison sociale et l'adresse des clients de la table, voici la requête SQL :

```
SELECT Numclt, RaisonSociale, AdresseClit  
FROM CLIENT
```

L'asterix \* permet d'obtenir toutes les colonnes d'une table sans les décliner explicitement dans le SELECT.

Exemple :

```
SELECT *  
FROM ARTICLE
```

cette requête renvoie toutes les colonnes et les lignes de la table ARTICLE.

La clause **ORDER** permet de trier le résultat de la requête.

```
SELECT <Liste des attributs>  
FROM <tables>  
ORDER BY <attributs>
```

Pour avoir un ordre croissant, il faut ajouter **ASC** après les attributs.  
Pour avoir un ordre décroissant, il faut ajouter **DESC** après les attributs.  
Il est possible de mixer les deux types de tri.

Exemples :

```
SELECT NumArt, Designation, NumTyp  
FROM ARTICLE  
ORDER BY Designation, NumTyp (tri croissant par défaut)  
->Tri du résultat : Par désignation, puis par prix (ordre croissant)
```

```
SELECT NumArt, Designation, NumTyp  
FROM ARTICLE  
ORDER BY Designation, NumTyp DESC
```

```
SELECT NumArt, Designation, NumTyp  
FROM ARTICLE  
ORDER BY Designation ASC, NumTyp DESC
```

La clause **DISTINCT** permet d'éliminer les doublons contenus dans le résultat d'une requête.

Exemple :

**SELECT Distinct Cp  
FROM CLIENT  
ORDER BY CP**

Cp
13000
42000
42300
44000
49000
75000

### 2.1.2. La restriction (ou sélection)

La restriction consiste à choisir une ou plusieurs lignes d'une table.

#### TABLE CLIENT

NumClt	RaisonSociale	AdresseClt	Cp
23459	Dupont SARL	12 rue d'Alsace	49000
23460	SA Delta Pièces	24 avenue du 8 mai	13000
23461	Dubois et fils	6 impasse fontval	42300
23462	Palermo srl	345 bd St Germain	75000
23463	Laplace SARL	23 impasse des guerrins	42000
23464	Delporte SA	12 rue du Gal de Gaulle	44000
23465	Poitevin Frères	19 rue de la victoire	49000

Elle est réalisée avec la clause **WHERE** :

```
SELECT <Liste des attributs>
FROM <tables>
WHERE <critères de restriction>
```

Les critères de restriction portent essentiellement sur les colonnes des tables. Ils utilisent les **opérateurs de comparaison** (=,<,>,<=,>=) ainsi que les **opérateurs logiques** (ET, OU, NON => AND, OR, NOT).

#### Exemples :

```
SELECT RaisonSociale, CP
FROM Client
WHERE CP= '49000'
```

**Cette requête donne la liste des clients du code postal 49000.**

```
SELECT RaisonSociale, CP
FROM Client
WHERE CP= '49000'
OR CP='45000'
```

**Cette requête donne la liste des clients du code postal 49000 et du 45000.**

L'opérateur **LIKE** permet de comparer le contenu d'une colonne avec une chaîne de caractères. Il utilise notamment le %.

```
SELECT *  
FROM CLIENT  
WHERE RaisonSociale like "SA%"  
→ clients dont la raison sociale commence par SA.
```

```
SELECT *  
FROM CLIENT  
WHERE RaisonSociale like "%SA"  
→ clients dont la raison sociale se termine par SA.
```

```
SELECT *  
FROM CLIENT  
WHERE RaisonSociale like "%SA%"  
→ clients dont la raison sociale contient SA.
```

 **Exercice : présentez les requêtes SQL permettant d'obtenir :**

1. La liste des articles (N°, désignation) dont le prix est supérieur à 12 euros.
2. La liste des articles (N°, désignation, prix) dont le prix est supérieur à 12 euros et inférieur à 25 euros.
3. La liste des commandes (N°, date commande) du mois de janvier 2008.
4. La liste des articles (N°, désignation) dont le type est 'S' ou 'P'.

### 2.1.3. La jointure

Cette opération consiste à lier des tables entre elles. Ce rapprochement va permettre d'élargir le nombre de colonnes candidates pour la projection.

Les jointures sont envisageables uniquement s'il existe des clefs étrangères.

Elles sont réalisées dans la clause **WHERE** :

```
SELECT <Liste des attributs>  
FROM table1, table2  
WHERE table1.ChampA=Table2.champB
```

**Exemple:**

```
SELECT NumClt, RaisonSociale, AdresseClt, Client.Cp , Ville.ville  
FROM CLIENT, VILLE  
WHERE CLIENT. Cp = VILLE.Cp
```

Cette requête affiche les coordonnées des clients et le nom de la ville



**Exercice : présentez la requête SQL permettant d'obtenir :**

1. La liste des commandes (N°commande, Raison sociale, Cp)
2. La liste des commandes (N°commande, Raison sociale, Cp) des clients du département 49.
3. La liste des commandes (N°commande, Raison sociale, Cp, Nom de la ville) triées par N° de commande.

## 2.2. Agrégations et regroupements

---

Il est souvent nécessaire d'effectuer des traitements sur les tuples extraits d'une ou plusieurs tables. Dans cette perspective, des fonctions d'agrégations sont nécessaires. Celles-ci permettent en effet, d'effectuer des opérations arithmétiques et statistiques (exemple : compter le nombre de lignes de commandes, faire la somme, la moyenne des quantités commandées, trouver la plus grande valeur pour un attribut, etc...).

Ces fonctions sont souvent associées à la notion de regroupement des tuples. En effet, le langage SQL permet d'appliquer les fonctions sur tous les tuples (exemple : la quantité totale commandée toute article confondu) ou sur un critère de regroupement (exemple : la quantité totale commandée par article).

### 2.2.1. Les fonctions d'agrégation

Elles permettent la réalisation de calculs ou traitements statistiques (dits verticaux) sur l'ensemble des valeurs d'un attribut.

Elles se retrouvent parmi les critères de projection de la requête.

Les fonctions souvent utilisées sont :

Fonctions	Description
<b>SUM</b>	Elle calcule la somme des valeurs d'un attribut
<b>AVG</b>	Elle calcule la moyenne des valeurs d'un attribut
<b>MAX</b>	Elle recherche la plus grande des valeurs d'un attribut
<b>MIN</b>	Elle recherche la plus petite des valeurs d'un attribut
<b>COUNT</b>	Elle compte le nombre de tuples qui répondent aux critères de restriction.

☛ Les fonctions SUM, AVG s'appliquent à des attributs de type numérique uniquement. Il faut être attentif au type de l'attribut afin de s'assurer de sa compatibilité avec la fonction d'agrégation envisagée.

La syntaxe générale est la suivante :

```
SELECT NomFonction(attributs(s))  
FROM <tables>
```

Remarque : les arguments placés à l'intérieur de la fonction peuvent être combiné avec des opérateurs mathématiques (+, -, /, \*) et correspondre à des attributs d'une même table ou de tables différentes. Il est également possible d'imbriquer des fonctions d'agrégation.

Exemples :

```
SELECT Sum(QtiteArt) as 'Total'  
FROM LIGNE_CDE  
WHERE NumArt = 'PYJ34095'
```

Cette requête affiche le total des quantités commandées pour l'article au n° 'PYJ34095'

```
SELECT Count(NumArt) as 'Résultat'  
FROM ARTICLE
```

Cette requête affiche le nombre d'articles dans la table article.

```
SELECT Max(PrixHT) as 'Résultat'  
FROM ARTICLE
```

Cette requête affiche le prix le plus élevé.

```
SELECT Count(NumArt) as 'Total articles'  
FROM ARTICLE
```

```
WHERE PrixHT between 25 and 45
```

Cette requête affiche le nombre d'articles dont le prix est compris entre 25 et 45.

```
SELECT Sum(QtiteArt * PrixHT) as 'Total'  
FROM ARTICLE A, LIGNE_CDE L
```

```
WHERE A.NumArt = L.NumArt
```

Cette requête affiche le chiffre d'affaire total.



**Exercice : présentez la requête SQL permettant d'obtenir :**

1. Le nombre de commandes prises durant le mois de janvier 2008 :
2. Le nombre de lignes de commandes prises entre le 01/02/08 et le 15/03/08 pour l'article dont le numéro est 'SHO00843' :
3. La quantité totale commandée et le chiffre d'affaire généré par les articles de type 'P'.
4. Le chiffre d'affaire moyen des commandes.

### 2.2.2. Les regroupements

Les requêtes précédentes fournissent une ligne pour résultat.

L'application d'un critère de regroupement permet d'avoir un résultat plus détaillée en terme de présentation. En effet, pour chacun des critères de regroupement (critères de rupture de en fait), les lignes de regroupées sont soumises à la même fonction d'agrégation.

Exemple :

#### **Nombre de clients par département**

- Par département :  
il s'agit du critère de regroupement.
- Nombre de clients :  
la fonction Count() sera utilisée pour agréger les tuples relatifs à un même département.

→ Résultat :

<b>CP</b>	<b>Nbre clients</b>
<b>49000</b>	<b>123</b>

<b>49300</b>	<b>39</b>
.....	
<b>79230</b>	<b>45</b>

Les regroupements avec le langage SQL s'effectuent avec la clause **GROUP BY** suivi des critères de regroupement (liste d'attributs).

La syntaxe générale est la suivante :

<pre><b>SELECT</b> attribut(s), NomFonction(attributs(s)) <b>FROM</b> &lt;tables&gt; <b>GROUP BY</b> attribut(s)</pre>
--

●\* Si des attributs ne sont pas incorporés dans une fonction d'agrégation dans la projection, ces derniers devront nécessairement être précisés dans la clause **GROUP BY**.

Exemple :

```
SELECT Cp, Count(NumClt) as 'Nbre de clients'  
FROM CLIENT  
GROUP BY Cp  
ORDER BY Cp
```

Cette requête affiche un résultat similaire au suivant:

<b>CP</b>	<b>Nbre clients</b>
<b>49000</b>	<b>123</b>
<b>49300</b>	<b>39</b>
.....	
<b>79230</b>	<b>45</b>

 **Exercice : présentez la requête SQL permettant d'obtenir :**

1. Le nombre de commandes prises durant le mois de janvier 2008 par client (Numéro client, Nom client, nbre de commandes).
2. Le nombre de lignes de commandes prises entre le 01/02/08 et le 15/03/08 par article (Numéro article, désignation, nbre de lignes)
3. Le chiffre d'affaire par type article (Type d'article, CA)
4. Le chiffre d'affaire par client (Raison Sociale, CA)
5. Le chiffre d'affaire, et le nombre de commandes par date de commande pour la période allant du 01/02/08 au 15/02/08 (Date de commande, CA, Nombre de commandes).

## **Les restrictions dans les requêtes avec regroupements**

Les critères de restriction relatifs aux regroupements sont implémentés avec la clause HAVING. En effet, « HAVING » s'applique aux fonctions d'agrégation en particulier.

Exemple :

```
SELECT Cp, Count(NumClt) as 'Nbre de clients'  
FROM CLIENT  
GROUP BY Cp  
HAVING Count()=5
```

Cette requête affiche les codes postaux, le nbre de clients. Ne s'affichent que les CP qui comportent 5 clients.

Exercices :

1. La liste des clients (Numéro de client, Raison sociale, CA) qui ont un chiffre d'affaire supérieur à 10 000 euros en 2007. Le résultat sera trié par CA décroissant :
2. La liste des clients (Numéro de client, Raison sociale) qui ont passé plus de deux commandes pour le mois de janvier de l'année en cours:
3. La liste des clients (Numéro de client, Raison sociale) des départements 49, 53, qui ont un CA moyen inférieur à 5 000 euros pour l'année en cours