



Informatique

Initiation aux requêtes SQL



Auteur : C. Terrier ; <mailto:webmaster@cterrier.com> ; <http://www.cterrier.com>

Utilisation : Reproduction libre pour des formateurs dans un cadre pédagogique et non commercial

Le langage **SQL** (Structured Query Language) est un langage informatique normalisé d'interrogation des bases de données relationnelles (SGBDR). Il est devenu un standard international en 1986. Il est utilisé par toutes les bases de données (Oracle, dBase, Access...). C'est un langage est proche d'un langage naturel.

Sommaire

A - Société Exemple

B – Syntaxe des commandes

C – Etude des commandes

1. Requête sélection : **SELECT FROM**

1.1 Requête simple

1.2 Requête multi-tables : Jointure

1.3 Trier les données : **ORDER BY**

1.4. Affichage avec restriction : **WHERE**

1.4.1 Sélection sur un critère

- Critère alphabétique ou numérique simple
- Critère alphabétique élaboré : **LIKE**
- Opérateur : **NOT**

1.4.2 Sélection multi-critères : **AND, OR, BETWEEN, NOT**

- Opérateur : **AND**
- Opérateur : **OR**
- Opérateur : **BETWEEN**

1.5 Les fonctions arithmétiques : **MAX, MIN, AVG, SUM, COUNT**

1.6 Regrouper les données : **GROUP BY**

1.6.1 Regroupement simple

1.6.2. Regroupement avec calcul

1.6.3 Regroupement limitation : **HAVING**

2. Requête ajout : **INSERT INTO...VALUE**

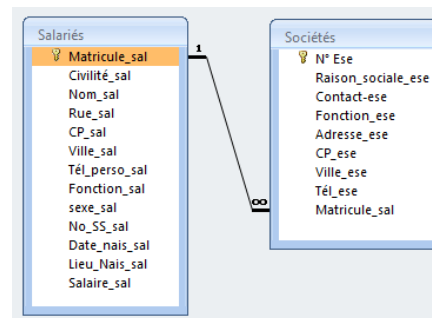
3. Requête suppression : **DELETE FROM**

4. Requête mise à jour : **UPDATE... SET**

A – Syntaxe des commandes SQL

Rappel :

- Dans une base de données relationnelle les informations sont stockées dans des champs qui sont regroupées dans des tables.
- Les tables regroupent les informations (champs) qui ont un lien commun (Information sur les salariés, sur les sociétés sur les visites,
- Une requête SQL affiche les contenus de champs qui appartiennent à des tables pour les afficher selon certains critères



MEMO DES PRINCIPALES COMMANDES

Select

SELECT "nom de champ" FROM "nom de table"
Affiche les données contenues dans les champs à partir des tables listées.

Order By

SELECT "nom de colonne"
 FROM "nom de table"
 [WHERE "condition"]
 ORDER BY "nom de colonne" [ASC, DESC]
Paramètre des critères de tri.

Where

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "condition"
Paramètre une condition de sélection.

Like

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "nom de colonne" LIKE {modèle}
Avec WHERE. Paramètre dans une condition de sélection, un critère sur une partie d'un champ.

Count, Avg, Max, Min, Sum

SELECT COUNT ("nom de colonne")
 FROM "nom de table"
Opérateur arithmétique : compte, moyenne, maximum, minimum, somme.

And – Or

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "condition simples"
 {[AND|OR] "condition simples"}+
Avec WHERE. Paramètre plusieurs conditions.

Not

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "condition simples"
 {[NOT] "condition simples"}
Avec WHERE. Paramètre exclue une condition.

Between

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "nom de colonne" BETWEEN 'valeur1' AND 'valeur2'
Paramètre un intervalle.

Group By

SELECT "nom de colonne 1", SUM("nom de colonne 2")

FROM "nom de table"
 GROUP BY "nom de colonne 1"
Paramètre un regroupement.

Having

SELECT "nom de colonne 1", SUM("nom de colonne 2")
 FROM "nom de table"
 GROUP BY "nom de colonne 1"
 HAVING (condition fonction)
Paramètre une condition.

Insert Into

INSERT INTO "nom de table" ("colonne 1", "colonne 2", ...)
 valeurS ("valeur 1", "valeur 2", ...)
Ajoute d'enregistrements.

Delete From

DELETE FROM "nom de table"
 WHERE {condition}
Suppression d'enregistrements.

Update

UPDATE "nom de table"
 SET "colonne 1" = [nouvelle valeur]
 WHERE {condition}
Modifier un enregistrement.

Distinct

SELECT DISTINCT "nom de colonne"
 FROM "nom de table"
Sélectionne dans un champ les données différentes.

In

SELECT "nom de colonne"
 FROM "nom de table"
 WHERE "nom de colonne" IN ('valeur1', 'valeur2', ...)
Sélectionne des données précises.

Creat Table

CREATE TABLE "nom de table"
 ("colonne 1" "type de données colonne 1", "colonne 2"
 "type de données colonne 2", ...)
Crée une table.

Drop Table

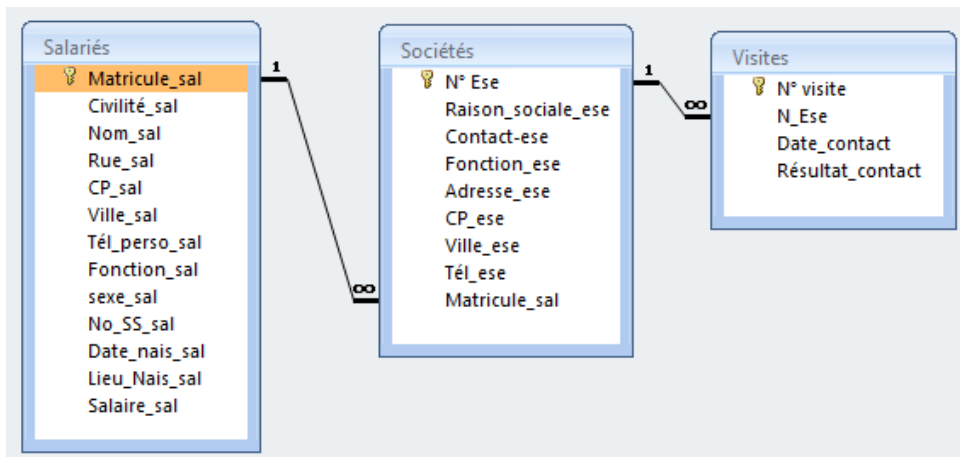
DROP TABLE "nom de table"
Supprimer une table.

Truncate Table

TRUNCATE TABLE "nom de table"
Supprime toute les données d'une table.

B - Société exemple

Pour illustrer ce cours nous allons travailler à partir de la base de données suivantes conçue sous Access :



Description de la base de données :

- Les salariés de cette société sont des commerciaux.
- Ils sont chargés d'assurer le suivi commercial des entreprises. Dans ce cadre chaque salarié est responsable de plusieurs sociétés. Inversement, une société ne peut être suivie que par un seul commercial.
- Chaque société est visitée plusieurs fois par an. La table Visites enregistre les comptes rendus de visites périodiques. Une entreprise fait l'objet de plusieurs visites par an. Chaque compte rendu concerne une seule entreprise.

Exemple : Liste des entreprises par vendeur

N° Ese	Raison_sociale_ese	Nom	Fonction_ese
1	EDIMCO SA	Floux Paul	PDG
3	BNP	Boulin Christophe	Directeur d'agence
18	NAPPERT	Brachut Marcelle	Directrice
19	SAFEX	Poulin Luc	Comptable
* (Nouv.)			

Exemple : Liste des visites par entreprises

Date_contact	Résultat_contact
22/08/2007	Envoyer documentation. A recontacter par téléphone dans 3 mois
15/12/2007	Souhaite installer un réseau dans un magasin
* (Nouv.)	

C – ETUDE DES COMMANDES

1. Requête sélection : SELECT... FROM...

L'instruction **SELECT... FROM...** est utilisée pour afficher (Projeter) les enregistrements d'une base de données

- La commande **SELECT** sélectionne les champs,
- La commande **FROM** sélectionne les tables dans lesquelles sélectionner les champs.

1.1. Requête simple

Exemple : **Afficher les champs : matricule, nom et tél de la table Salariés**
`SELECT Salariés.matricule_sal, Salariés.nom_sal, Salariés.tél_sal`
`FROM Salariés`



- les noms de champs et de tables doivent strictement respecter la syntaxe utilisée dans la définition des tables.
- Le symbole * affiche tous les champs de la table.

Exemple : **Afficher tous les champs de la table Salariés**
`SELECT * FROM Salariés`

Entrainement	<p>The diagram shows three tables: Salariés (with fields: Matricule_sal, Civilité_sal, Nom_sal, Rue_sal, CP_sal, Ville_sal, Tél_perso_sal, Fonction_sal, sexe_sal, No_SS_sal, Date_nais_sal, Lieu_Nais_sal, Salaire_sal), Sociétés (with fields: N° Ese, Raison_sociale_ese, Contact_ese, Fonction_ese, Adresse_ese, CP_ese, Ville_ese, Tél_ese, Matricule_sal), and Visites (with fields: N° visite, N_Ese, Date_contact, Résultat_contact). Lines connect Salariés to Sociétés and Sociétés to Visites, indicating relationships.</p>
	Afficher les champs : raison sociale, contact et fonction de la table sociétés
	Afficher tous les champs de la table visites

1.2. Requête multi tables (jointure)

La jointure permet d'afficher des champs provenant de plusieurs tables. Les noms de champs sont placés dans la commande SELECT et sont précédé du nom de la table. De plus les noms des tables sont séparés par une virgule dans la commande FROM

Attention pour sélectionner uniquement les données en relation vous devez les filtrer à l'aide de la commande WHERE (Voir également 1.4)

Exemple : **Afficher les champs : matricule et nom de la table salariés ainsi que les champs : raison sociale, contact, fonction et téléphone de la table société.**
`SELECT Salariés.matricule_sal, Salariés.nom_sal, Sociétés.Raison_sociale_ese, Sociétés.Contact_ese, Sociétés.fonction_ese, Sociétés.tél_ese`
`FROM Salariés, Sociétés`
`WHERE Sociétés.Matricule_sal = Salariés.matricule_sal`

Entrainement	(Base de données ci-dessus)
Afficher les champs : matricule et nom de la table salariés ainsi que les champs N° ese, raison sociale et contact de la table Sociétés	
Afficher les champs : N° ese, Raison sociale, contact et fonction de la table société et les champs N° visite, date et résultat de la table visites.	

1.3. Trier les données : ORDER BY

La commande ORDER BY paramètre le nom du champ sur lequel trier les données ainsi que le critère de tri : ASC (croissant) ou DESC (décroissant)

Exemple : **Afficher les champs matricule, nom et tél de la table salariés en les triant sur le nom trié par ordre croissant**

```
SELECT Salariés.Matricule_sal, salaries.Nom_sal, Salariés.Tél_perso_sal
FROM Salariés
ORDER BY Nom_sal ASC
```

Il est possible d'indiquer plusieurs clés de tri en les saisissant les un à la suite des autres

Exemple : **Afficher les champs : matricule, nom et tél de la table salariés triés sur le matricule par ordre croissant et sur le nom par ordre décroissant**

```
SELECT Salariés.Matricule_sal, Salariés.Nom_sal, Salariés.Tél_perso_sal
FROM Salariés
ORDER BY Matricule_sal, ASC, Nom_sal DESC
```

Entrainement	(Base de données ci-dessus)
<pre> graph LR Salariés[Salariés] --- 1 à ∞ Sociétés[Sociétés] Sociétés --- 1 à ∞ Visites[Visites] </pre>	
Afficher les champs : Raison sociale, contact et fonction de la table sociétés trié sur le nom du contact	
Afficher tous les champs de la table visites trié sur la date et sur le numéro de l'entreprise	

1.4. Affichage avec restriction : WHERE

La commande **WHERE** permet de filtrer les enregistrements. La syntaxe de la commande est la suivante :
 WHERE « champs » **opérateur critère de filtre**

Une requête peut être simple (1.4.1) ou plus élaboré avec des conditions imbriquées (1.4.2) :

- Conditions :** Afficher les hommes => **Sexe = « Homme »**
Simple Afficher les personnes qui habitent à Lyon => **Ville = « Lyon »**
 Afficher les clients dont le chiffre d'affaires est > à 5 000 € => **CA > 5000**
 Afficher les clients qui ne sont pas de Lyon => **Ville NOT « Lyon »**
- Conditions :** Afficher les hommes de 20 ans => **Sexe = « Homme » AND age >20**
Imbriquées Afficher les clients de Lyon et Grenoble => **Ville = « Lyon » OR Ville = « Grenoble »**

Les opérateurs de sélection sont les suivants :

Opérateurs disponibles		
Opérateurs	Effets	Exemples
=	égal	WHERE salaire = 2000 €
<	inférieur à	WHERE salaire < 2000 €
>	supérieur à	WHERE salaire > 2000 €
<=	inférieur ou égal à	WHERE salaire <= 2000 €
>=	supérieur ou égal à	WHERE salaire >= 2000 €
<>	différent de	WHERE salaire <> 2000 €
PAS NULL	non vide	WHERE salaire = NULL
NULL	vide	WHERE salaire = PAS NULL
LIKE	contient	WHERE Prénom LIKE « Luc »

1.4.1. Sélection sur un critère

➤ **Critère alphabétique ou numérique simple**

Exemple : **Afficher les champs : matricule, nom et tél de la table salariés pour les hommes**
 SELECT Salaries.Matricule_sal, Salaries.Nom_sal, Salaries.Tél_perso_sal
 FROM Salariés
 WHERE Civilité_sal = "Homme"

Entrainement	<p>Salariés : Matricule_sal, Civilité_sal, Nom_sal, Rue_sal, CP_sal, Ville_sal, Tél_perso_sal, Fonction_sal, sexe_sal, No_SS_sal, Date_nais_sal, Lieu_Nais_sal, Salaire_sal</p> <p>Sociétés : N° Ese, Raison_sociale_ese, Contact_ese, Fonction_ese, Adresse_ese, CP_ese, Ville_ese, Tél_ese, Matricule_sal</p> <p>Visites : N° visite, N_Ese, Date_contact, Résultat_contact</p>
	Afficher les champs : numéro d'entreprise, rue, code postal et ville de la table sociétés pour les entreprises de Paris
	Afficher les champs : matricule, rue, code postal et ville, date de naissance de la table salariés pour les salariés de Lyon triés sur la date de naissance

➤ Critère alphabétique élaboré : LIKE

La commande **LIKE** signifie contient. Associé à la commande WHERE, elle permet de sélectionner une donnée contenue dans un champ. Exemple : je recherche le mot « République » dans le champ Rue.

Exemple **Afficher les salariés dont le nom de rue contient le mot République**
 SELECT Salariés.Nom_sal, Salariés.Rue_sal, Salariés.Ville_sal
 FROM Salariés
 WHERE Rue_sal LIKE « République »

La commande LIKE peut être associée au symbole % qui remplace une chaîne de caractère. Dans ce cas elle affiche uniquement les enregistrements qui contiennent le texte situé avant, après ou entre les signe % (chaîne de caractères)

Exemple 1 : **Afficher les villes qui commencent par la lettre : L** => WHERE Ville_sal LIKE "L%"
 Exemple 2 : **Afficher les villes dont le nom se termine par : ian** => WHERE Ville_sal LIKE "%ian"
 Exemple 3 : **Afficher les villes dont le nom contient les lettres : Du** => WHERE Ville_sal LIKE "%Du%"

Entrainement	(Base de données ci-dessus)
Afficher les champs : RS, rue, code postal et ville de la table sociétés pour les entreprises dont le nom commence par BO	
Afficher les champs : civilité, nom, rue et ville de la table salariés pour les salariés qui habitent sur une place	

➤ L'opérateur NON (NOT)

L'opérateur NOT permet d'exclure de l'affichage les enregistrements qui satisfont la condition.
 Exemple : Afficher tous les clients qui ne sont pas des particuliers

Exemple : **Afficher les champs : matricule, nom et salaire des salariés qui ne sont pas cadres**
 SELECT Salariés.Matricule_sal, Salariés.Nom_sal,
 FROM Salariés
 WHERE NOT Fonction_sal = "Cadre"

Entrainement	Diagramme de base de données
Afficher les champs : numéro d'entreprise, rue, code postal et ville de la table sociétés des entreprises dont le code postal n'est pas 75000	<p>Le diagramme illustre la structure de la base de données avec trois tables : Salariés, Sociétés et Visites. Salariés a des champs : Matricule_sal, Civilité_sal, Nom_sal, Rue_sal, CP_sal, Ville_sal, Tél_perso_sal, Fonction_sal, sexe_sal, No_SS_sal, Date_nais_sal, Lieu_Nais_sal, Salaire_sal. Sociétés a des champs : N° Ese, Raison_sociale_ese, Contact_ese, Fonction_ese, Adresse_ese, CP_ese, Ville_ese, Tél_ese, Matricule_sal. Visites a des champs : N° visite, N_Ese, Date_contact, Résultat_contact. Les relations sont : Salariés (1) --- (8) Sociétés et Sociétés (1) --- (∞) Visites.</p>

Afficher les champs : matricule, rue, code postal et ville, date de naissance de la table salariés pour les salariés qui ne sont pas nés à Lyon

1.4.2. Sélection sur plusieurs critères

Une sélection peut imbriquer plusieurs critères entre eux :

- Afficher les clients de Lyon ou Grenoble => opérateur OU (OR)
- Afficher les clients de Lyon dont le chiffre d'affaires est > à 3000 € => opérateur ET (AND)
- Afficher les chantiers commencés entre le 01/01/08 et le 15/05008 => opérateur ENTRE (BETWEEN)

➤ L'opérateur ET (AND)

L'opérateur **AND** permet d'associer des critères qui s'ajoutent. (Les deux critères doivent être remplis).

Exemple : Je souhaite afficher les clients qui habitent à Lyon et dont le chiffre d'affaires est supérieur à 10 000 €

Exemple : **Afficher le matricule, le nom et le salaire des salariés masculin dont le salaire est supérieur à 2000 €**
 SELECT Salariés.Matricule_sal, Salariés.Nom_sal, Salarié.Civilité_sal Salariés.Salaire_sal
 FROM Salariés
 WHERE Civilité_sal = "Homme" AND Salaire_sal > 2000

Entraînement	(Base de données ci-dessus)
Afficher le numéro d'entreprise, la rue, le code postal et le ville des entreprises situées à Paris suivi par le salarié dont le matricule est 123	
Afficher le matricule et le nom des salariés de Lyon dont le salaire est supérieur à 3 000 €.	

➤ L'opérateur OU (OR)

L'opérateur **OR** permet d'associer des critères qui s'éliminent. (Un des deux critères doit être rempli).

Exemple : Je souhaite afficher les clients grossistes et les administrations

Attention : La formulation de la condition « OU » est « piègeuse » car pour afficher les grossistes et les administrations. Il faut utiliser l'opérateur OU. L'utilisation de l'opérateur ET revient à afficher les clients qui sont à la fois des grossistes et des administrations, ce qui est impossible. Lors d'une requête lorsque le résultat est vide, l'erreur provient le plus souvent de cette confusion.

Exemple : **Afficher le matricule, le nom et le salaire des salariés de Lyon et Grenoble**
 SELECT Salariés.Matricule_sal, Salariés.Nom_sal, Salariés.Salaire_sal, Salarié.Ville_sal
 FROM Salariés
 WHERE Ville_sal = "Lyon" OR Ville_sal = "Grenoble"

Entrainement	
Afficher les champs : numéro d'entreprise, rue, code postal et ville de la table sociétés pour les entreprises de Paris ou de Lyon	
Afficher les champs : RS, contact et Télé des sociétés dont le numéro de matricule du représentant est 123 ou 125	

➤ **L'opérateur ENTRE (BETWEEN)**

L'opérateur BETWEEN permet de paramétrer un intervalle de valeur.

Exemple : Afficher tous les clients dont le chiffre d'affaires est compris entre 1 000 et 2 000 €.



Cet opérateur peut être remplacé par l'opérateur AND :

- Afficher les salariés dont l'age est compris entre 20 et 30 ans
- Afficher les salariés dont l'age est > 20 et < 30

Exemple : **Afficher le matricule, le nom et le salaire des salariés masculin dont le salaire est supérieur à 2 000 € et inférieur à 3 000 €.**

```
SELECT Salariés.Matricule_sal, Salariés.Nom_sal, Salariés.Salaire_sal
FROM Salariés
WHERE Salaire_sal BETWEEN 2000 AND 3000
```

Entrainement	
Afficher les champs : numéro d'entreprise, rue, code postal et ville de la table sociétés pour les entreprises dont le code postal est compris entre 38000 et 39000	
Afficher le matricule, la rue, le code postal, la ville et la date de naissance des salariés dont la « date_d'embauche » de la table salariés est comprise entre le 01/01/2000 et le 31/12/2000	

1.5. Les fonctions arithmétiques: MAX, MIN, AVG, SUM, COUNT

Les fonctions arithmétiques permettent d'afficher des statistiques sur des champs numériques ou monétaires ou de compter le nombre d'occurrence d'un champ.

La fonction est placée devant le nom du champ qui est mis entre parenthèse.

Exemple 1 : **Afficher le nombre de salariés**
 SELECT COUNT(Salariés.Matricule_sal)
 FROM Salariés

Exemple 2 : **Afficher le salaire moyen des cadres**
 SELECT AVG(Salaire_sal), Fonction_sal
 FROM Salariés
 WHERE Fonction_sal = "Cadre"

Les fonctions disponibles sont les suivantes :

Fonctions	Effets
MAX	Affiche la valeur maximum d'un champ
MIN	Affiche la valeur minimum d'un champ
AVG	Affiche la moyenne des données d'un champ
SUM	Affiche la somme des données d'un champ
COUNT	Affiche le nombre de valeur d'un champ

Entrainement	
	<p>Salariés: Matricule_sal, Civilité_sal, Nom_sal, Rue_sal, CP_sal, Ville_sal, Tél_perso_sal, Fonction_sal, sexe_sal, No_SS_sal, Date_nais_sal, Lieu_Nais_sal, Salaire_sal</p> <p>Sociétés: N° Ese, Raison_sociale_ese, Contact_ese, Fonction_ese, Adresse_ese, CP_ese, Ville_ese, Tél_ese, Matricule_sal</p> <p>Visites: N° visite, N_Ese, Date_contact, Résultat_contact</p>
	Afficher le nombre de salariés de l'entreprise
	Afficher le salaire maximum et le salaire minimum des salariés

1.6. Regrouper les données : GROUP BY

1.6.1. Regroupement simple

La commande GROUPE BY regroupe à l'affichage les données sur le champ spécifié.

Exemple : Par défaut les visites sont triées par ordre chronologique. Dès lors les visites concernant une même entreprise sont séparées à l'affichage. Le regroupement des visites sur le nom de la société permet d'afficher les visites qui concernent une même entreprise les unes à la suite des autres.

Affichage sans regroupement			Affichage avec regroupement		
Date	Entreprise	Ventes	Entreprise	Date	
15/01/2008	Bernard	1000 €	Bernard	15/01/2008	1000 €
17/01/2008	Paule	800 €		20/02/2008	600 €
20/02/2008	Bernard	600 €		28/02/2008	1200 €
28/02/2008	Bernard	1200 €		15/03/2008	300 €
03/03/2008	Paule	400 €	Paule	17/01/2008	800 €
15/03/2008	Bernard	300 €		03/03/2008	400 €

La fonction est placée devant le nom du champ qui est mis entre parenthèse.

Exemple : **Afficher entreprises regroupées par salariés**
 SELECT Salariés.matricule_sal, Salariés.nom_sal, Sociétés.Raison_sociale_ese,
 Sociétés.Contact_ese, Sociétés.Fonction_ese, Sociétés.Tél_ese
 FROM Sociétés, Salariés
 WHERE sociétés.Matricule_sal = Salariés.matricule_sal
 GROUPE BY Sociétés.Matricule_sal

Entrainement
<p>Salariés Matricule_sal Civilité_sal Nom_sal Rue_sal CP_sal Ville_sal Tél_perso_sal Fonction_sal sexe_sal No_SS_sal Date_nais_sal Lieu_Nais_sal Salaire_sal</p> <p>Sociétés N° Ese Raison_sociale_ese Contact_ese Fonction_ese Adresse_ese CP_ese Ville_ese Tél_ese Matricule_sal</p> <p>Visites N° visite N_Ese Date_contact Résultat_contact</p>
Afficher le numéro de l'entreprise, la ville et le numéro de téléphone regroupés par salariés responsable
Afficher les adresses des salariés regroupés par ville

1.6.2. Regroupement avec calcul

La commande GROUPE BY est utilisée pour réaliser des calculs concernant un même enregistrement.

Exemple : Calculer le total des ventes pour chaque client :

Table Visites			Affichage avec regroupement		
Affichage sans regroupement					
Date	Entreprise	Ventes	Entreprise	Date	
15/01/2008	Bernard	1000 €	Bernard	15/01/2008	1000 €
17/01/2008	Paule	800 €		20/02/2008	600 €
20/02/2008	Bernard	600 €		28/02/2008	1200 €
28/02/2008	Bernard	1200 €		15/03/2008	300 €
03/03/2008	Paule	400 €		Total	4100 €
15/03/2008	Bernard	300 €	Paule	17/01/2008	800 €
				03/03/2008	400 €
				Total	1200 €

Exemple : **Calculer le chiffre d'affaires total réalisé avec chaque client dans le tableau de la page précédente**

```
SELECT Entreprise, SUM(Ventes)
FROM Visites
GROUPE BY Entreprise
```

Entrainement	Table : Lignes_Commande Champs : N°_commande, référence, quantité, N°_client
Calculer les quantités totales commandées par type de référence	
Calculer le nombre de commandes passées par client	

1.6.3. Regroupement limitation : HAVING

La commande HAVING jointe à un calcul permet de restreindre l’affichage uniquement aux sociétés qui satisfont un critère

Exemple : Calculer le total des ventes pour chaque client et afficher les résultats uniquement pour les entreprises dont le chiffre d'affaires est supérieur à 3 000 €.

Table Visites			Affichage avec regroupement		
Affichage sans regroupement					
Date	Entreprise	Ventes	Entreprise	Date	
15/01/2008	Bernard	1000 €	Bernard	15/01/2008	1000 €
17/01/2008	Paule	800 €		20/02/2008	600 €
20/02/2008	Bernard	600 €		28/02/2008	1200 €
28/02/2008	Bernard	1200 €		15/03/2008	300 €
03/03/2008	Paule	400 €		Total	4100 €
15/03/2008	Bernard	300 €			

Exemple : **Calculer le chiffre d'affaires total pour chaque client et afficher ceux dont le total est supérieur à 3000**

```
SELECT Entreprise, SUM(Ventes)
FROM Visites
GROUPE BY Entreprise
HAVING SUM(Ventes) > 3000
```

Entrainement	Table : Lignes_Commande Champs : N°_commande, référence, quantité, N°_client
Calculer les quantités totales commandées par type de référence et afficher uniquement ceux dont la quantité moyenne est supérieur à 50 articles	
Calculer le nombre de commandes passées par client et afficher uniquement ceux dont le nombre de commandes est supérieur à 12 articles	

2. Requête ajout : INSERT INTO VALUE

La commande INSERT INTO permet d'ajouter des enregistrements aux champs d'une table

Exemple : **Ajouter à la table visites l'enregistrement suivant :
Date 15/04/2008 ; Entreprise : Paule ; Vente 600 €**
 INSERT INTO "Visites" ("Date", "Entreprise", "Ventes")
 VALUES ("15/04/2008", "Paule", 600,)

Table Visites		
Affichage sans regroupement		
Date	Entreprise	Ventes
15/01/2008	Bernard	1000 €
17/01/2008	Paule	800 €
20/02/2008	Bernard	600 €

Entrainement	<p>Salariés: Matricule_sal, Civilité_sal, Nom_sal, Rue_sal, CP_sal, Ville_sal, Tél_perso_sal, Fonction_sal, sexe_sal, No_SS_sal, Date_nais_sal, Lieu_Nais_sal, Salaire_sal</p> <p>Sociétés: N° Ese, Raison_sociale_ese, Contact_ese, Fonction_ese, Adresse_ese, CP_ese, Ville_ese, Tél_ese, Matricule_sal</p> <p>Visites: N° visite, N_Ese, Date_contact, Résultat_contact</p>
Ajouter l'enregistrement suivant à la table Sociétés : N° ese : 23, Société : BongrainSA , Contact : GoudardLuc , Fonction : PDG , Adresse : 2 rue du Mont, 33000, BORDEAUX, 03 45 12 32 65, 02	

3. Requête suppression : DELETE FROM

La commande DELETE FROM permet de supprimer un ou des enregistrements d'une table

Exemple : **Supprimer de la table visites les l'enregistrement du 17/01/2008 :**
 DELETE FROM Visites
 WHERE Date = "17/01/2008"

Table Visites		
Affichage sans regroupement		
Date	Entreprise	Ventes
15/01/2008	Bernard	1000 €
17/01/2008	Paule	800 €
20/02/2008	Bernard	600 €

Entrainement	(Base de données ci-dessus)
Supprimer de la table sociétés les enregistrements qui concernent la société BREAL SA	
Supprimer de la table visites les enregistrements qui concernent la société dont le matricule est 025	

4. Requête mise à jour : UPDATE... SET

La commande UPDATE SET permet de modifier des enregistrements d'une table

Exemple : **la date du 17/01/2008 est fausse il fallait saisir 19/01/2008:**

```
UPDATE Visites
SET Date = "19/01/2008"
WHERE Date = "17/01/2008"
```

Table Visites		
Affichage sans regroupement		
Date	Entreprise	Ventes
15/01/2008	Bernard	1000 €
17/01/2008	Paule	800 €
20/02/2008	Bernard	600 €

Entrainement	<p>The diagram shows three tables: Salariés, Sociétés, and Visites. Salariés has a primary key Matricule_sal. Sociétés has a primary key N° Ese. Visites has a primary key N° visite. There is a one-to-many relationship between Salariés and Sociétés, and between Sociétés and Visites.</p>
	<p>Le salarié Dupont (matricule 033) est muté, dorénavant les sociétés qu'il gérait seront suivies par le salarié Grémont Paul dont le matricule est 028. Mettre à jour la table sociétés.</p>
	<p>Le salarié Dupont (matricule 033) a changé de N° de téléphone. Ancien numéro : 04 78 23 56 89 ; Nouveau numéro : 04 78 43 33 28</p>