

Memento SQL pour les MGTMN

Les commandes entre crochets sont celles qui ne sont pas utilisées dans chaque requête.

Requêtes sur une seule table

Affichage de valeurs présentes dans une table

Commandes	Commentaires
SELECT [DISTINCT] <i>col1</i> , <i>col2</i> , ... ou SELECT *	Sélection de colonnes (de champs) sélection de tout
FROM <i>table</i>	à partir d'une table
[WHERE <i>condition</i>]	ne garder que les lignes qui vérifient une condition
[GROUP BY <i>col</i>]	regrouper les résultats (n'affiche qu'une ligne du groupe)
[HAVING <i>condition</i>]	ne garder que les groupes vérifiant une condition
[ORDER BY <i>col</i>] ou [ORDER BY <i>col</i> DESC]	tri des lignes tri dans l'ordre décroissant
[LIMIT <i>nb</i>]	n'afficher que <i>nb</i> lignes
[OFFSET <i>debut</i>]	commencer à une certaine ligne

Les conditions

Les *conditions* utilisent souvent une comparaison :

égal à	différent de	supérieur	inférieur	supérieur ou égal	inférieur ou égal
=	!= ou <>	>	<	>=	<=

et parfois les opérateurs suivants :

appartient à une liste	n'appartient pas à une liste	« ressemble à »
IN	NOT IN	LIKE

Dans le cas de LIKE, le caractère « _ » remplace une lettre quelconque tandis que « % » remplace un texte quelconque.

Plusieurs conditions peuvent être utilisées en utilisant les mots clés AND, OR ou NOT. Attention dans ce cas à utiliser des parenthèses pour les conditions compliquées.

Exemples de requêtes

Les exemples ci-dessous sont en rapport avec [une base de données communes](#).

Requêtes	Effet
<pre>SELECT nom_comm FROM communes</pre>	Affiche les valeurs présentes dans la colonne <code>nom_comm</code> de la table <code>communes</code> .
<pre>SELECT nom_comm, code_dept FROM communes</pre>	Affiche les valeurs présentes dans la colonne <code>nom_comm</code> de la table <code>communes</code> .
<pre>SELECT DISTINCT code_dept FROM communes</pre>	Affiche les valeurs présentes dans la colonne <code>code_dept</code> sans répétition.
<pre>SELECT * FROM communes</pre>	Affiche la table <code>communes</code> en entier.
<pre>SELECT nom_comm FROM communes WHERE code_dept = 59</pre>	Affiche le nom des communes du Nord.
<pre>SELECT nom_comm, code_dept FROM communes WHERE code_dept != 59 ORDER By code_dept</pre>	Affiche le nom des communes et leur numéro de département pour celles qui ne sont pas du Nord. Les lignes sont triées suivant le code de département.
<pre>SELECT nom_comm FROM communes WHERE code_dept = 59 AND nom_comm LIKE "L%"</pre>	Affiche les noms des communes du Nord qui commencent par la lettre L.
<pre>SELECT * FROM communes WHERE code_dept = 59 OR population > 100</pre>	Affiche les informations des communes qui sont dans le Nord ou qui ont plus de 100000 habitants (renvoie aussi les communes qui vérifient les deux conditions).
<pre>SELECT code_dept, COUNT(*) FROM communes GROUP BY code_dept</pre>	Regroupe les communes par département et pour chaque groupe affiche une ligne avec le code et le nombre de lignes (donc de communes).
<pre>SELECT nom_comm FROM communes WHERE population > 100 GROUP BY code_dept HAVING COUNT(*) > 5</pre>	Prends les communes de plus de 100 mille habitants, les groupes par département et ne conserve que les groupes de plus de cinq communes. Affiche une seule commune par groupe...

Calculs à partir des valeurs présentes dans une table

Lors de l'utilisation de ces fonctions, une table d'une seule ligne est renvoyée.

Fonctions	Commentaires
<code>MAX(col)</code>	renvoie la valeur maximale dans la colonne <code>col</code>
<code>MIN(col)</code>	renvoie la valeur minimale dans la colonne <code>col</code>
<code>SUM(col)</code>	renvoie la somme des valeurs de la colonne <code>col</code>
<code>AVG(col)</code>	renvoie la moyenne des valeurs de la colonne <code>col</code>
<code>STD(col)</code>	renvoie l'écart type des valeurs de la colonne <code>col</code>
<code>COUNT(col) WHERE condition</code>	renvoie le nombre de valeurs de la colonne <code>col</code> dans les lignes vérifiant une certaine <code>condition</code>

Exemples de requêtes

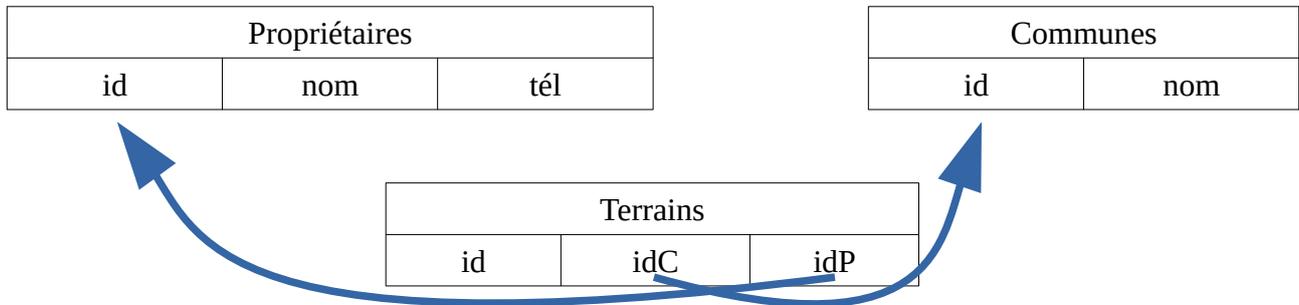
Requêtes	Effet
<code>SELECT MAX(population), nom_comm FROM communes</code>	Affiche la plus grande population et la commune correspondante.
<code>SELECT SUM(population) FROM communes</code>	Affiche la population totale des communes.
<code>SELECT AVG(population) FROM communes WHERE code_dept = 59</code>	Affiche la population moyenne des communes du Nord.
<code>SELECT COUNT(*) FROM communes WHERE code_dept = 59</code>	Affiche le nombre de communes du Nord (le nombre de lignes vérifiant la condition).
<code>SELECT code_dept, COUNT(*) FROM communes GROUP BY code_dept</code>	Crée des groupes de communes par département et affiche le code des départements et le nombre de communes dans chacun.

Requêtes sur plusieurs tables (jointures)

Pour interroger plusieurs tables, il faut utiliser les mots clés JOIN ... ON ... :

Exemples de requêtes

Ces exemples reposent sur le schéma relationnel suivant :



Requêtes	Effet
<pre> SELECT Terrains.id,Communes.nom FROM Terrains JOIN Communes ON Terrains.idC = Communes.id </pre>	Affiche les identifiants des terrains et leur commune.
<pre> SELECT Propriétaires.nom,Terrains.id FROM Propriétaires JOIN Terrains ON Terrains.idP = Propriétaires.id JOIN Communes ON Terrains.idC = Communes.id WHERE Communes.nom = "Roscoff" </pre>	Affiche les propriétaires de Roscoff et l'identifiant de leur terrain. Attention à l'ordre des jointures.
<pre> SELECT P.nom,COUNT(*) FROM Propriétaires AS P JOIN Terrains AS T ON T.idP = P.id JOIN Communes AS C ON T.idC = C.id WHERE Communes.nom LIKE "R%" GROUP BY P.nom HAVING COUNT(*) >= 1 ORDER BY P.nom </pre>	Sélectionne les résultats dans les communes commençant par R, regroupe ceux-ci par nom de propriétaire et ne retient que les groupes (donc les propriétaires) qui ont au moins un terrain. Trie les résultats par ordre alphabétique de nom de propriétaire.