

MÉMO REQUÊTES SQL

LA REQUÊTE SELECT

Exemple :

SELECT

t1.champ1,

t2.champ2,

expression,

agregation

-- on peut faire un commentaire avec -- pour une ligne.

-- Sur plusieurs lignes on utilise /* commentaire */

FROM

nom_schema23.nom_table1 **AS** t1,

nom_schema6.nom_table2 **AS** t2

WHERE

/* « and » veut dire « et », « or » veut dire « ou », in(valeur 1,valeur 2) veut dire égal à l'une des valeurs */

condition1

AND (condition2 **OR** condition3)

AND insee **IN** ('75105', '75106')

GROUP BY

champs_ou_expression

ORDER BY

champ_champs_ou_expression

LIMIT nombre ;

CRÉATION / ACTUALISATION / MODIFICATION D'UNE TABLE

/*Pour créer une table, il faut utiliser CREATE TABLE. Pour modifier une table, il faut la supprimer (DROP TABLE), puis la recréer (CREATE TABLE). */

DROP TABLE nom_de_la_table ;

CREATE TABLE nom_de_la_table **AS**

SELECT ... ;-- (ce que l'on veut dans la table)

--Pour modifier ou actualiser un champ, il faut utiliser UPDATE, SET.

UPDATE nom_de_la_table

SET nom_du_champ = valeur_ou_champ_ou_expression ;

--Pour ajouter/supprimer une colonne, il faut utiliser ALTER TABLE, ADD/DROP COLUMN.

ALTER TABLE nom_de_la_table

ADD COLUMN nom_du_champ **TYPE** (longueur du champ)

/*exemple pour ajouter une colonne*/ : **ADD COLUMN** population **INTEGER** (10)

/*pour supprimer une colonne/* **DROP COLUMN** population

OPÉRATIONS, FONCTIONS, AGRÉGATION

Les opérateurs mathématiques

Sigle	Description	Exemple	Résultats
+	addition	2 + 3	5
-	soustraction	4 - 2	2
*	multiplication	4 * 4	16
@	valeur absolue	@ -2	2
ROUND	arrondi	Round(18.5874, 2)	18,58
<	strictement inférieur		
<=	inférieur ou égal		
>	strictement supérieur		
>=	supérieur ou égal		
<>	différent		

LIKE

'abc' LIKE 'abc' VRAI	'abc%' LIKE 'abcdefgh' VRAI
'abc' LIKE 'a%' VRAI	'abc_' LIKE 'abcdefgh' FAUX
'abc' LIKE '_b_' VRAI	'abcd' LIKE 'abc%' VRAI
'abc' LIKE 'abcd' FAUX	'abc' LIKE 'ABC' FAUX
'abc_' LIKE 'abcd' VRAI	
'abc%' LIKE 'abcd' VRAI	

Si ... Alors

CASE
WHEN condition1 **THEN** valeur1
[**WHEN** condition2 **THEN** valeur2]
[...]
ELSE valeur_autre
END AS nom_nouveau_champ

Agrégation

SELECT val1, val2, count(*)
FROM ma_table
GROUP BY val1, val2
ORDER BY val1, val2

Fonctions d'agrégation

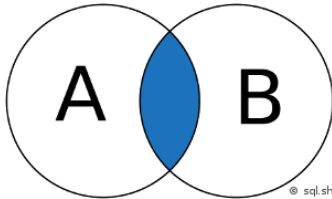
Count (*) → compte les lignes non nulles
Count(colonne1) → compte les lignes ou les valeurs dans la colonne1 sont non nulles
Count(DISTINCT colonne1) → compte les lignes ou les valeurs dans la colonne1 sont non nulles et dont les valeurs sont différentes entre elles
Min (colonne1) → le minimum de la colonne
Max (colonne1) → le maximum de la colonne
Avg (colonne1) → la moyenne de la colonne
Sum (colonne1) → la somme de la colonne

LES JOINTURES

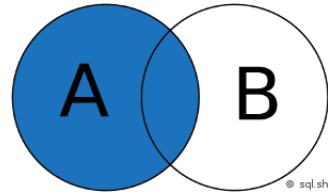
```
SELECT *  
FROM table1 AS t1  
JOIN table2 AS t2  
ON t1.var1 = t2.var2  
/* éventuellement la clause */ WHERE, GROUP BY, etc.
```

Les différents types de jointures

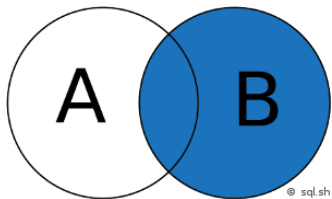
JOIN



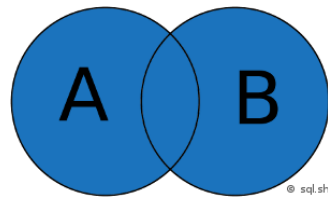
LEFT JOIN



RIGHT JOIN



FULL JOIN



LES PRINCIPAUX TYPES DE DONNÉES

Type	Description	Exemple de champs
CHARACTER VARYING	Texte de longueur maximale fixée	Identifiant, nom de commune
TEXT	Suite longue de caractère (sans limite de taille)	Descriptif, commentaire
INTEGER	Entier long	Population, surface
NUMERIC	Décimal	Prix, surface
DATE	Date du calendrier grégorien	Date
BOOLEAN	Vrai / faux	
Geometry (point, 2154)	Géométrie en point	Station de métro
Geometry (polygon, 2154)	Géométrie en polygone	Parcelle

LES PRINCIPALES FONCTIONS SPATIALES

st_area(geom) → retourne l'aire de l'objet. En m² (dépend du système géodésique)

st_distance(geom1,geom2) → retourne la distance séparant 2 objets

st_centroid(geom) → retourne le centre d'un polygone (à combiner avec st_astext pour pouvoir l'exploiter)

st_intersection(geom1,geom2) → construit la zone d'intersection entre 1 et 2

st_intersects(geom1,geom2) → retourne un booléen indiquant si 1 intersecte 2

st_contains(geom1,geom2) → retourne un booléen indiquant si l'objet 1 contient 2

st_within(geom1,geom2) → retourne un booléen indiquant si l'objet 1 est dans 2

st_buffer(geom,n) → retourne un objet tampon recouvrant tous les points situés à une distance \leq n de l'objet. (n dépend du système géodésique (en général m))

st_length(geom) → retourne la longueur ou le périmètre d'un objet

st_union(geom) → union de la géométrie des objets. Opère comme un *sum* → on peut utiliser *GROUP BY*

st_xmin, st_ymin, st_xmax, st_ymax(geom) → calcule les coordonnées minimum et maximum en x et y de la géométrie

Pour plus d'information, consulter la documentation : <https://docs.postgresql.fr/>