**Requêtes SQL**

* **SELECT** colonne

**FROM** table : selection de colonne provenant de table

Plusieurs colonnes : séparation champs avec « , » et toutes avec « \* »

Ex pour ordre : SELECT \*

FROM table

WHERE condition

GROUP BY expression

HAVING condition

{ UNION | INTERSECT | EXCEPT }

ORDER BY expression

LIMIT count

OFFSET start

* SELECT **DISTINCT** colonnes

FROM table : évite doublons

* SELECT colonnes

FROM table

**WHERE** condition : extraction lignes selon condition. Si condition sur du texte ex : territoire = 'n2' Attention casse!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| =  | <> != (pas égal) | >  | <  | <=  | >= |

* + Opérateurs de comparaisons :
	+ **AND** / **OR** pour s’assurer de condition1 ET / OU condition2 et utilisation parenthèses si nécessaire
	+ **IN** (valeur1, valeur2, valeur3, …): liste de valeurs possibles 1 ou 2 ou 3 …, simplifie OR, peut-être associé à NOT
	+ **BETWEEN** ‘valeur1’ AND ‘valeur2’ : intervalle données, surtout pour nombres ou dates, peut-être associé à NOT, attention valeurs inclues/exclues !
	+ **LIKE** : début, milieu ou fin mot, avec % (chaine caractères joker) ou \_ (1 caractère jocker) on peut rechercher des fini par et/ou commence par
	+ **IS** (**NOT**) **NULL** : utile car NULL pas filtrable par opérateurs de comparaison
* SELECT colonne1, fonction(colonne2)

FROM table

**GROUP BY** colonne1: toujours après WHERE et avant HAVING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SUM()  | AVG() | COUNT()  | MAX()  | MIN()  |

* + Fonctions agrégation:
* SELECT colonne1, fonction(colonne2)

FROM table

GROUP BY colonne1

**HAVING** fonction(colonne2) operateur valeur: comme WHERE mais filtre avec les fonctions aggrégation

* SELECT colonne1, colonne2, colonne3

FROM table

**ORDER BY** colonne1 DESC, colonne2 ASC : tri (ascendant pas défaut)

* **AS**: renommer temporairement table ou colonne