

QUELQUES ÉLÉMENTS DE SYNTAXE SQL

PC

SELECT ... FROM

SELECT * FROM Truc	Affiche toute la table Truc.
SELECT Bidule FROM Truc	Affiche la colonne bidule de la table Truc.
SELECT Bidule, machin FROM Truc	Affiche les colonnes machin et bidule de la table Truc.

SELECT ... FROM ... WHERE

SELECT * FROM Villes2 WHERE Pop_2010>100000	Affiche toutes les entrées (lignes) de la table Villes2 correspondant à des villes dont la population était supérieure à 100000 habitants en 2010 .
SELECT Nom FROM Villes2 WHERE Pop_2010>100000 AND NOT Nom LIKE 'P%'	Affiche les villes dont la population était supérieure à 100000 habitants en 2010 et dont le nom ne commence pas par la lettre P.

SELECT ... FROM ... ORDER BY

SELECT * FROM Villes2 ORDER BY POP_2010	Affiche les lignes de la table Villes2 ordonnées par population 2010 croissante (<i>par défaut, c'est l'ordre croissant qui est adopté</i>).
SELECT * FROM Villes2 ORDER BY POP_2010 DESC	Affiche les lignes de la table Villes2 ordonnées par population 2010 décroissante.
SELECT * FROM Villes2 ORDER BY POP_2010 DESC, Nom ASC	Affiche les lignes de la table Villes2 ordonnées par population 2010 décroissante, et, en cas d'égalité, par nom croissant (au sens lexicographique).

LIMIT et OFFSET

SELECT * FROM Villes2 ORDER BY POP_2010 DESC LIMIT 10	Affiche les 10 villes les plus peuplées en 2010.
SELECT * FROM Villes2 ORDER BY POP_2010 DESC LIMIT 5 OFFSET 10	Affiche les 5 suivantes.

FONCTION D'AGREGATION : MIN, MAX, SUM, COUNT, AVG

SELECT MAX(Pop_2010) FROM Villes2 WHERE Code_Postal LIKE '92%'	Renvoie la population de la ville des Hauts-de-Seine la plus importante.
SELECT COUNT(*) FROM Villes2 WHERE Code_Postal LIKE '%9%'	Renvoie le nombre de villes situées dans un département dont le numéro comporte le chiffre 9.

GROUP BY

SELECT * FROM Villes2 GROUP BY Dept	Renvoie une entrée (<i>la première possible de la table</i>) par département
SELECT Dept, Max(Pop_2010) FROM Villes2 GROUP BY Dept	Renvoie la liste des départements, avec, pour chacun d'entre eux, la population de sa ville la plus peuplée.

DISTINCT

SELECT DISTINCT TRUC FROM BIDULE	Renvoie toutes les entrées de la colonne Truc de la table Bidule, en évitant les doublons.
----------------------------------	--

GROUP BY ... HAVING

```
SELECT Dept, Max(Pop_2010) AS N
  FROM Villes2 GROUP BY Dept
  HAVING N BETWEEN 100000 AND 200000
```

Contrairement à WHERE, HAVING effectue la restriction après que le regroupement a été réalisé.

ALIAS : AS

```
SELECT TRUC AS MACHIN FROM BIDULE
  ou
SELECT TRUC MACHIN FROM BIDULE
```

Sélectionne l'attribut (colonne) "Truc" de la table Bidule, en le renommant "Machin". Ceci est souvent nécessaire lors de sélections imbriquées. Le « AS » n'est pas nécessaire, mais aide à la lisibilité.

UNION

```
SELECT COL1, COL2 FROM TABLE 1
UNION
```

Réunion des colonnes COL1, COL2 de la Table 1 et des colonnes COL3, COL4 de la Table2.

```
SELECT COL3, COL4 FROM TABLE 2
```

Les données de COL1 et COL3 doivent être du même type, de même pour les entrées de COL2 et COL4

INTERSECT

```
SELECT COL1 FROM TABLE1
```

No comment.

```
INTERSECT
```

```
SELECT COL2 FROM TABLE2
```

SELECTIONS IMBRIQUEES (Exemples)

```
SELECT AVG(N) FROM
  (SELECT COL1 AS N FROM TABLE1
    WHERE ...)
  WHERE ...
```

Des alias (renommages) sont le plus souvent nécessaires lors de requêtes imbriquées du type SELECT ... FROM (SELECT ...) WHERE ...

```
SELECT COL1 FROM TABLE 1
  WHERE COL1>
    (SELECT AVG(COL2) FROM TABLE2
      WHERE ...)
```

SELECT ... FROM ... WHERE (SELECT ...) est un autre type de sélections imbriquées.

JOINTURES : JOIN ... ON

```
SELECT X, Y, Z, T FROM Table 1
  JOIN Table2, Table3
    ON X=Z
```

Une jointure sans conditionnement « ON » se référant à la clé primaire de l'une des tables qui est clé étrangère pour les autres tables n'a généralement aucune utilité.

```
SELECT Table1.X, Table2.X, Table3.X, T
  FROM Table 1 JOIN Table2, Table3
    ON Table1.X=Table3.X
```

Lorsque des colonnes des différentes tables mises en jeu portent le même nom, on précise...

Quelques opérateurs

+ ; - ; / ; *

Opérateurs arithmétiques usuels.

= ; < ; > ; <= ; >= ; !=

Opérateurs de comparaison usuels (pas de == en SQL!).

NOT

Négation.

AND

ET

OR

OU

LIKE 'truc' est vraissi la chaîne de caractère se termine par 'Truc'.

X LIKE "%truc%" est vraissi la chaîne de caractère contient le mot 'Truc'.

