

SQL AIDE MÉMOIRE

Instruction SELECT

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table;
```

Clause DISTINCT

```
1 SELECT DISTINCT colonne1, colonne2 ... colonneN
2 FROM nom_table;
```

Clause WHERE

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE CONDITION;
```

Clause AND/OR

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE CONDITION-1 {AND|OR} CONDITION-2;
```

Clause IN

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE nom_colonne IN (val-1, val-2,...val-N);
```

Clause BETWEEN

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE nom_colonne BETWEEN val-1 AND val-2;
```

Clause LIKE

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE nom_colonne LIKE { PATTERN };
```

Clause ORDER BY

```
1 SELECT colonne1, colonne2 ... colonneN
2 FROM nom_table
3 WHERE CONDITION
4 ORDER BY nom_colonne {ASC|DESC};
```

Clause GROUP BY

```
1 SELECT SUM(nom_colonne)
2 FROM nom_table
3 WHERE CONDITION
4 GROUP BY nom_colonne;
```

Clause COUNT

```
1 SELECT COUNT(nom_colonne)
2 FROM nom_table
3 WHERE CONDITION
```

Clause HAVING

```
1 SELECT SUM(nom_colonne)
2 FROM nom_table
3 WHERE CONDITION
4 GROUP BY nom_colonne;
5 HAVING (condition de la fonction arithmétique);
```

OPÉRATEURS SQL – AIDE MÉMOIRE

Opérateurs arithmétiques	
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Reste de la division euclidienne
Opérateurs de comparaison	
A = B	A égal à B
A <> B	A différent de B
A > B (A < B)	A supérieur à B (A inférieur à B)
A >= B (A <= B)	A supérieur ou égal à B (A inférieur ou égal à B)
A !> B (A !< B)	A n'est pas supérieur à B (A n'est pas inférieur à B)
Opérateurs de comparaison	
A BETWEEN B AND C	A est compris entre B et C
A LIKE 'chaîne de caractères'	A contient la chaîne de caractères
A IN (B1, B2, B3, ...)	A est présent dans la liste (B1, B2, B3, ...)
A IS NULL	A n'a pas de valeur (à ne pas confondre avec 0)
Opérateurs de logique	
ALL	L'opérateur ALL permet de comparer une valeur à toutes les valeurs d'un autre jeu de valeurs.
ANY	L'opérateur ANY est utilisé pour comparer une valeur à une valeur applicable de la liste conformément à la condition.
BETWEEN	L'opérateur BETWEEN permet de rechercher des valeurs comprises dans un ensemble de valeurs, en fonction de la valeur minimale et de la valeur maximale.
EXISTS	L'opérateur EXISTS est utilisé pour rechercher la présence d'une ligne dans une table spécifiée qui répond à un certain critère.
IN	L'opérateur IN est utilisé pour comparer une valeur à une liste de valeurs littérales spécifiées.
LIKE	L'opérateur LIKE est utilisé pour comparer une valeur à des valeurs similaires à l'aide d'opérateurs génériques.
IS NULL	L'opérateur IS NULL est utilisé pour comparer une valeur avec une valeur NULL .
UNIQUE	L'opérateur UNIQUE recherche dans chaque ligne de la table spécifiée l'unicité (pas de doublons).
OR	L'opérateur OR est utilisé pour combiner plusieurs conditions dans la clause WHERE d'une instruction SQL.
AND	L'opérateur AND permet l'existence de plusieurs conditions dans la clause WHERE d'une instruction SQL.
NOT	L'opérateur NOT inverse la signification de l'opérateur logique avec lequel il est utilisé. Ex. : NOT EXISTS , NOT BETWEEN , NOT IN , etc. Ceci est un opérateur de négation.
Opérateurs pour les chaînes de caractère	
%	Le symbole pourcentage représente zéro, un ou plusieurs caractères joker.
_	Le symbole underscore représente un seul caractère joker.
	Concaténation. Exemple : SELECT name ' (' status ')' AS name_ AND _status FROM entity ; Attention, il faut utiliser concat avec MySQL.