Le SQL : syntaxe et exemples

Définition

SQL= Structured query language = *Langage de requêtes structuré*

SQL est un langage de définition de données (LDD, ou en anglais DDL *Data definition language*), un langage de manipulation de données (LMD, ou en anglais DML, *Data manipulation language*) et un langage de contrôle de données (LCD, ou en anglais DCL, *Data control language*), pour les bases de données relationnelles.

Mettre des commentaires en SQL

-- suivi d'un espace puis du commentaire sur une ligne : (norme SQL92, auparavant ##)
/* suivi du commentaire sur une ligne ou plusieurs lignes ***/**

Ressources : http://sql.developpez.com/

Légende
[optionnel]
<à remplacer>
variable en italique

Exemples de requêtes SQL

Sélections (respecter l'ordre des clauses)

```
SELECT  *  FROM nom_table ;                                  // * pour tous les champs
SELECT  nom_col2,  nom_col4, nom_col3                        // liste de champs séparés par virgule
        FROM nom_table
        WHERE <condition>                                    // pour sélectionner des enregistrements
        ORDER BY nom_col,  nom_col1 DESC ;                   // tri DESCendant (par défaut ASC)
SELECT  nom_col1, nom_col4 AS mon_alias1, nom_col2 mon_alias2   // alias de champ (AS facultatif)
        FROM nom_table ;
SELECT  nom_col1, nom_col4 AS 'mon alias1', nom_col2 'mon alias2'   // alias de champ avec espace ' ' ou` `
        FROM nom_table ;
SELECT  DISTINCT nom_col2, nom_col3                          // DISTINCT pour éviter les doublons
        FROM nom_table;
SELECT  COUNT(*),                                            // nb total d'enregistrements
        COUNT(DISTINCT nom_col2),                            // nb de valeurs distinctes non null
        FROM nom_table ;
SELECT  nom_col4, COUNT(nom_col2) AS Nb,
        MIN(nom_col2) AS Min,
        AVG(nom_col2) AS Moyenne                             // alias de champ (AS facultatif)
        FROM nom_table
        GROUP BY nom_col4                                    // regroupement (agrégat)
        WHERE <condition>
        HAVING <condition sur le regroupement>               // condition sur le regroupement
        ORDER BY Moyenne ASC, Nb DESC
        LIMIT 3,10 ;                                         // limiter le nombre de résultats
```

Sélection avec jointure en SQL1 (1986)

```
SELECT  t1.nom_col1, t2.nom_col
        FROM nom_table1 AS t1, nom_table2 AS t2              // alias de table (AS facultatif)
        WHERE t1. nom_col4= t2.nom_col8
        AND <conditions éventuelles de sélection> ;
```

Sélection avec jointure en SQL2 (1992)

```
SELECT  t1.nom_col1, t2nom_col2
        FROM nom_table1 t1 [INNER] JOIN nom_table2 t2;
        ON t1. nom_col4= t2.nom_col8
        WHERE < conditions éventuelles de sélection> ;
```

Insertion d'enregistrement

sous forme complète :

```
INSERT  INTO nom_table (nom_col2, nom_col4,nom_col5)
        VALUES (12.5, null, 'montexte') ;
```

sous forme incomplète :

```
INSERT  INTO nom_table
        VALUES ('', 12.5, NULL, 'montexte') ;
```

Suppression de tous les enregistrements

```
DELETE  FROM nom_table ;
```

```
TRUNCATE TABLE nom_table ;
```

Suppression d'enregistrements

```
DELETE  FROM nom_table
        WHERE <condition> ;                                  // Ne pas oublier sinon la table se vide
```

Mise à jour d'enregistrements

UPDATE nom_table
SET *nom_col1= 'val1', nom_col2= nom_col2*105/100,nom_col4=nom_col3*nom_col1, nom_col5=null*
WHERE *<condition> ;*

Création de table

CREATE TABLE *nom_table* (
nom_col1 TINYINT(2) unsigned NOT NULL AUTO_INCREMENT,
nom_col2 DECIMAL(3,2) NOT NULL default 0,
nom_col3 VARCHAR(28) NULL,
nom_col4 VARCHAR(4) NOT NULL,
PRIMARY KEY (nom_col1),
FOREIGN KEY (*nom_col4*) REFERENCES *nom_table2*(*nom_col8*)
) ;

Suppression de table

DROP TABLE *nom_table3, nom_table2* ;

Modification de table

ALTER TABLE *nom_table*
ADD [COLUMN] *nom_col type_de_col* AFTER *nom_col3* ; // ajout de colonne après une autre
ALTER TABLE *nom_table*
DROP *nom_col* ; // suppression de colonne
ALTER TABLE *nom_table*
CHANGE *nom_col nom_col type_de_col* ; // changement de nom ou type de colonne

Expression de conditions

nom_col2 <>12.5 AND (*nom_col3* < '*val1*' OR *nom_col3* = '*val2*' OR *nom_col3* IS NOT NULL)
nom_col NOT IN ('*val1*', '*val2*', '*val3*')
nom_col NOT BETWEEN *10.5* AND *15*
nom_col LIKE '*%exp2*' // 2 caractères joker :
nom_col NOT LIKE '*exp1%exp2*' // % remplace 0 ou n caractères
nom_col LIKE '*_ _exp3%*' // _ remplace 1 et 1 seul caractère

Opérateurs - Prédicats

= > >= < <= != <>		// opérateurs de comparaison
IN	NOT IN	// appartenance à une liste de valeurs
BETWEEN	NOT BETWEEN	// appartenance à un intervalle de valeurs
LIKE	NOT LIKE	// à utiliser avec % et _
IS NULL	IS NOT NULL	// test de valeur Null

Fonctions d'agrégation (calculs effectués sur un ensemble d'enregistrements)

COUNT(*) // nb d'enregistrements total
COUNT(DISTINCT *nom_col2*) // nb de valeurs distinctes non null
SUM(*nom_col2*) // somme des valeurs
MIN(*nom_col2*) // minimum des valeurs
MAX(*nom_col2*) // maximum des valeurs
AVG(*nom_col2*) // moyenne des valeurs

Quelques fonctions

UPPER('chaîne') : met en majuscules
LOWER('chaîne') : met en minuscules
LTRIM('chaîne') RTRIM(' chaîne ') TRIM(' chaîne ') : supprime les espaces à gauche, à droite ou des 2 côtés
NOW( ) : Renvoie la date et heure courante (année, mois, jour, heure, minute, seconde, fraction de seconde).
CURRENT_DATE( ) ou CURDATE( ) : Renvoie la date courante (année, mois, jour, heure).
YEAR('date') : extrait l'année
MONTH('date') : extrait le mois (de 1 à 12)
DAY('date') : extrait le jour (de 1 à 31)
DATE_FORMAT('date','%d/%m/%Y') : Affiche la date formatée, ici au format jj/mm/aaaa.
CHAR_LENGTH('chaîne') : renvoie la longueur d'une chaîne de caractères
CONCAT(exp1, exp2,exp3) : concatène les expressions indiquées
ROUND(nombre, n) : renvoie le nombre arrondi à n décimales
SUBSTRING('chaine',p,n) : renvoie la sous-chaîne de n caractères de 'chaine', à partir du p^{ième} caractère