

Chapitre 3

Les ordres du SQL

1. Les bases du langage SQL

1.1 Les expressions

Dans la plupart des syntaxes Transact-SQL, il est possible d'utiliser des expressions ou des combinaisons d'expressions pour gérer des valeurs ou pour tirer parti de la programmabilité du langage. Les expressions peuvent prendre différentes formes :

Les valeurs constantes

Exemple

■ **Caractère** 'AZERTY', 'Ecole Nantaise d''Informatique'

Les chaînes de caractères sont entourées d'apostrophes. S'il doit y avoir une apostrophe dans la chaîne de caractères, il est nécessaire d'en mettre deux pour ne pas le confondre avec la fin de la chaîne de caractères.

■ **Numérique** 10, -15.26, 1.235e-5

Les valeurs numériques peuvent être écrites avec des chiffres en utilisant le point pour séparer la partie entière de la partie décimale. Il est possible d'utiliser la notation scientifique en utilisant le *e* avant la puissance de :

Date		
constante	date	heure
'231205'	5 Décembre 2023	00:00:00.000
'05/12/2023'	5 Décembre 2023	00:00:00.000
'05-12-23 8:30'	5 Décembre 2023	08:30:00.000
'8:30:2'	1 Janvier 1900	08:30:02.000
'5.12.2023 8:30pm'	5 Décembre 2023	20:30:00.000

Le format français jj/mm/aaaa est l'un des formats possibles pour les dates.

■ **Binaire** 0x05, 0xFF, 0x5aef1b
 ■ **Nulle** NULL

Les données binaires doivent être données en hexadécimal et sont préfixées par 0x.

Les noms de colonne

Un nom de colonne pourra être employé comme expression, la valeur de l'expression étant la valeur "stockée" de la colonne.

Les fonctions

On peut utiliser comme expression n'importe quelle fonction, la valeur de l'expression est le résultat retourné par la fonction.

Exemple

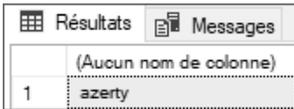
expression	valeur
SQRT(9)	3
substring('ABCDEF',2,3)	'BCD'

Les variables

Les variables peuvent être employées en tant qu'expression ou dans une expression, sous la forme @nom_de_variable. La valeur de l'expression est la valeur de la variable.

Exemple

```
DECLARE @x CHAR(10);  
SELECT @x='AZERTY';  
SELECT LOWER(@x);
```



Résultats		Messages	
(Aucun nom de colonne)			
1	azerty		

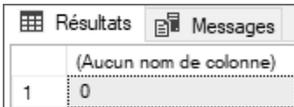
Les sous-requêtes

Une requête SELECT entre parenthèses peut être employée en tant qu'expression ayant pour valeur le résultat de la requête, soit une valeur unique, soit un ensemble de valeurs. Les requêtes SELECT sont présentées à la section L'extraction de lignes de ce chapitre.

Exemple

Stockage du nombre de clients dans une variable :

```
DECLARE @nombre INT;  
SELECT @nombre = (SELECT COUNT(*) FROM Clients);  
SELECT @nombre;
```



Résultats		Messages	
(Aucun nom de colonne)			
1	0		

Les expressions booléennes

Elles sont destinées à tester des conditions (IF, WHILE, WHERE, etc.). Ces expressions sont composées de la manière suivante :

```
expression1 opérateur expression2
```

1.2 Les opérateurs

Les opérateurs vont permettre de constituer des expressions calculées, des expressions booléennes ou des combinaisons d'expressions.

1.2.1 Les opérateurs arithmétiques

Ils permettent d'effectuer des calculs élémentaires et de renvoyer un résultat.

+ Addition

- Soustraction

* Multiplication

/ Division : division entière si les deux opérandes sont des entiers, division réelle sinon.

% Modulo (reste de la division entière)

(...) Parenthèses

Exemple

```
DECLARE @prixArticle NUMERIC(6,2) = 145.2;
DECLARE @TVA NUMERIC(2,2) = 20.0/100;
SELECT @prixArticle*(1+@TVA);
```

Résultats		Messages
(Aucun nom de colonne)		
1	174.2400	

```
DECLARE @nb INT = (SELECT COUNT(*) FROM Clients);
IF @nb%2=0 PRINT 'nombre de clients pair';
ELSE PRINT 'nombre de clients impair';
```

nombre de clients pair

Remarque

Les opérateurs + et - fonctionnent également sur les dates. L'unité est alors le jour. Exemple : `GETDATE()+1` permet d'obtenir la date du lendemain.

1.2.2 La concaténation de chaînes de caractères

La concaténation permet de constituer une seule chaîne de caractères à partir de plusieurs expressions de type caractère. L'opérateur de concaténation est le signe plus (+).

Exemple

```
DECLARE @nom CHAR(4) = 'LEVY';
DECLARE @prenom CHAR(5) = 'Simon';
SELECT 'M. ' + @nom + ' ' + @prenom;
```

Résultats		Messages
	(Aucun nom de colonne)	
1	M. LEVY Simon	

Il existe de plus la fonction CONCAT() qui permet de réaliser des concaténations. Contrairement à l'opérateur +, cette fonction appartient à la norme SQL. Elle a également l'avantage de pouvoir être utilisée avec des nombres sans que la somme soit effectuée.

1.2.3 Les opérateurs binaires

Ils permettent le calcul entre entiers, traduits implicitement en valeurs binaires.

& ET

| OU

^ OU Exclusif

~ NON

1.2.4 Les opérateurs de comparaison

Ils permettent la constitution d'expressions booléennes en comparant des expressions. Ces expressions peuvent être placées entre parenthèses.

exp1 = exp2

Égal.

`exp1 > exp2`

Supérieur.

`exp1 >= exp2`

Supérieur ou égal.

■ Remarque

`exp1 !> exp2` est également possible mais n'est pas dans la norme.

`exp1 < exp2`

Inférieur.

`exp1 <= exp2`

Inférieur ou égal.

■ Remarque

`exp1 !< exp2` est également possible mais n'est pas dans la norme.

`exp1 <> exp2`

Différent.

■ Remarque

`exp1 != exp2` est également possible mais n'est pas dans la norme.

`exp IN (exp1, exp2, ...)`

Indique si la valeur est parmi celles de la liste de valeurs.

`exp IS NULL` ou `exp IS NOT NULL`

Teste l'absence de valeur (NULL). Pour tester si une variable n'a pas de valeur affectée, il est indispensable d'utiliser l'opérateur IS NULL.

`exp LIKE 'masque'`

Filtre la chaîne de caractères ou la date suivant le masque spécifié.

Le masque peut être composé de :

—

Un caractère quelconque.

%

n caractères quelconques, avec n compris entre zéro et l'infini.

[ab...]

Un caractère dans la liste ab... .

[a-z]

Un caractère dans l'intervalle a-z.

[^ab...]

Un caractère en dehors de la liste ou de l'intervalle spécifié.

ab...

Le caractère lui-même.

Pour utiliser `_`, `%`, `[` et `^` en tant que caractères de recherche, il faut les encadrer de `[]`.

Exemple

masque	Chaînes correspondantes
'G%'	commençant par "G"
'_X%1'	deuxième caractère "X", dernier "1"
'%[1-9]'	se terminant par un chiffre compris entre "1" et "9"
'[^XW]%'	ne commençant ni par X ni par W
'LE[_]%'	commençant par "LE_"

exp LIKE 'masque' ESCAPE '\'

Lorsque le masque contient un caractère spécial comme `_` ou `&`, il est nécessaire de les faire précéder d'un caractère spécifique appelé caractère d'échappement. Afin de s'adapter à toutes les situations possibles, le caractère d'échappement est libre et il est simplement nécessaire de le spécifier après le mot-clé ESCAPE.

`exp BETWEEN min AND max`

Teste si la valeur de `exp` est incluse dans l'intervalle compris entre les valeurs `min` et `max` (bornes incluses).

`EXISTS` (sous-requête)

Renvoie VRAI si la sous-requête a renvoyé au moins une ligne.

Les opérateurs booléens

Ils permettent de combiner des expressions booléennes (`expb`) en renvoyant une valeur booléenne.

`expb1 OR expb2`

Vrai si une des deux expressions est vraie.

`expb1 AND expb2`

Vrai si les deux expressions sont vraies.

`NOT expb`

Vrai si `expb` est fausse.

1.3 Les fonctions

De nombreuses fonctions sont disponibles pour valoriser des colonnes ou pour effectuer des tests. Des exemples sont donnés avec l'instruction `SELECT`.

Parmi toutes les fonctions proposées en standard par SQL Server, il est possible de les regrouper par type : rowset, agrégation, ranking, scalaire. Pour ce dernier type de fonctions, il faut en plus les classer par catégories : mathématique, chaîne de caractères, date... car elles sont très nombreuses.

Certaines fonctions, plus particulièrement les fonctions permettant de manipuler les données de type caractère, prennent en compte le classement défini au niveau du serveur.