

Fonctions, expression, opérateur et conditions (SQL-DML pour Oracle)

Table des matières :

1	INTRODUCTION	1
2	EXPRESSIONS	1
3	OPERATEURS.....	2
4	CONDITIONS	2
4.1	OPERATEURS DE COMPARAISON	2
4.2	OPERATEURS LOGIQUES.....	2
5	FONCTIONS	3
5.1	FONCTIONS SCALAIRES MATHEMATIQUES.....	3
5.2	FONCTIONS SCALAIRES CHAINES DE CARACTERES	5
5.3	FONCTIONS SCALAIRES DATES	7
5.4	FONCTIONS SCALAIRES DE CONVERSION.....	9
5.5	FONCTIONS DE COMPARAISONS	11
5.6	FONCTIONS SCALAIRES DIVERSES	11
5.7	FONCTIONS DE GROUPE	13

1 Introduction

Les instructions du DML permettent l'ajout, la suppression, la modification et la visualisation des lignes dans les tables existantes.

Les instructions SQL manipulent des expressions. Ces expressions font référence à des noms d'objets de la base, à des constantes, comportent des appels à des fonctions standardisées et composent ces éléments avec des opérateurs.

Des expressions logiques (conditions) permettent également de définir la portée des instructions.

2 Expressions

Les termes des expressions peuvent être :

- constantes caractères
exemple : 'chaîne de caractères' ; 'Ecole Nantaise d'Informatique'.
- constantes littérales date (format dépendant de la langue configurée pour l'instance)
exemple : '15-JAN-94'
- constantes numériques
exemple : 10 ; -123.459 ; -1.26e+6
- noms d'attribut de table
exemple : CLIENTS.NOCLI, ARTICLES.DESIGNATION
- fonctions

exemple : `SQRT(81) ; REPLACE('IAGADIGI', 'I', 'OU');SYSDATE`

- pseudo-colonnes

exemple : `nomsequence.NEXTVAL ; ROWID.`

3 Opérateurs

- arithmétiques + - / * ()

exemple : `1.15 * PRIX ; (2 * MTLIG)/5 ; SYSDATE +15`

- sur les chaînes de caractères : concaténation ||

exemple : `'Monsieur' || NOM`

4 Conditions

Les conditions mettent en jeu des expressions, des opérateurs de comparaison et des opérateurs logiques.

4.1 Opérateurs de comparaison

La valeur des expressions logiques peut être VRAI, FAUX ou INCONNU. Une comparaison sera évaluée comme INCONNU si au moins un de ses termes est NULL.

- Comparaison simple

`expression1 {=,!=,<>, <,<=, >, >=} expression2`

- Appartenance à un ensemble de valeurs

`expression1 IN (expression2,...)`

VRAI si `expression1` apparaît au moins une fois dans la liste `(expression2,...)`.

- Appartenance à un intervalle de valeurs

`expression1 BETWEEN expression2 AND expression3`

VRAI si `expression1` se situe entre les bornes `expression2` et `expression3`, bornes comprises.

- Comparaison par rapport à un format de chaîne de caractères

`expression1 LIKE 'format'`

Le format peut inclure les méta-caractères :

- "%" pour désigner une suite de 0 à n caractères
- "_" pour désigner un et un seul caractère.

4.2 Opérateurs logiques

Une expression NULL n'est ni VRAI ni FAUX.

- Négation d'une expression logique

`NOT expression`

- Combinaison d'expressions logiques

`expression1 { AND / OR } expression2`

5 Fonctions

Il existe deux types de fonctions :

- Les fonctions scalaires (scalar function) qui portent sur une seule ligne : la fonction est exécutée et retourne un résultat pour chaque ligne de la requête.
- Les fonctions sur un regroupement de lignes (group function) : la fonction est exécutée une fois et retourne un résultat pour un ensemble de lignes de la requête. Ces fonctions sont appelées fonctions d'agrégat.

Rq : Si une fonction est appelée avec un argument NULL, la valeur retournée est NULL. Cette règle s'applique à toutes les fonctions à l'exception de CONCAT, NVL et REPLACE, et des fonctions d'agrégat.

Notation des arguments des fonctions :

n expression numérique.

d expression date.

c expression caractère.

b expression logique.

5.1 Fonctions scalaires mathématiques

ABS (n)

valeur absolue de n.

CEIL (n)

premier entier supérieur ou égal à n.

COS (n)

cosinus.

COSH (n)

cosinus hyperbolique.

EXP (n)

e puissance n (e=2,71828183...).

FLOOR (n)

1^{er} entier $\leq n$.

LN (n)

logarithme népérien de n.

LOG (m,n)

logarithme de n base m.

MOD (n1,n2)

reste de $n1/n2$.

POWER (n1,n2)

n1 exposant n2.

ROUND (n1,[n2])

n1 arrondi à n2 positions décimales.

SIGN (n)

-1 si $n < 0$; 0 si $n = 0$; +1 si $n > 0$.

SIN (n)

sinus de n.

SINH (n)

sinus hyperbolique de n.

SQRT (n)

racine carrée de n.

TAN (n)

tangente de n.

TANH (n)

tangente hyperbolique de n.

TRUNC (n1,[n2])

n1 tronqué à n2 positions décimales.

Rq : La plupart des fonctions numériques retournent une valeur avec 38 chiffres significatifs. Cependant les fonctions COS, COSH, EXP, LN, LOG, SIN, SINH, SQRT, TAN et TANH retournent une valeur avec seulement 36 chiffres significatifs.

5.2 Fonctions scalaires chaînes de caractères

ASCII (c)

code ASCII du 1^{er} caractère de c.

ASCIISTR (c)

la chaîne de caractères passée en paramètre est convertie au format de la table ASCII de la base.

CHR (n)

caractère ASCII du code n.

COMPOSE (c)

la chaîne de caractères passée en paramètre est convertie en format UNICODE.

CONCAT (c1, c2)

concaténation de c1 et c2.

DECOMPOSE (c)

cette fonction est valide uniquement pour les chaînes au format unicode et elle retourne une chaîne contenant les différents caractères.

INITCAP (c)

renvoie c avec la première lettre de chaque mot en majuscule et les autres en minuscules.

INSTR (c1,c2[,n1[n2]])

position de la nème occurrence (défaut 1^{ère}) de c2 dans c1 à partir de la position n1 (1 par défaut).

INSTRB (c1,c2,n1[,n2])

identique à INSTR avec n1 et n2 en octets.

LENGTH (c)

nombre de caractères de c.

LENGTHB (c)

identique à LENGTH retourne le nombre d'octets.

LOWER (c)

renvoie c en minuscules.

LPAD (c1,n[,c2])

renvoie c1 sur une longueur n complété à gauche avec c2 (par défaut ' ').

LTRIM (c1[,c2])

renvoie c1 avec suppression à gauche de tous les c2 (par défaut ' ').

NLS_INITCAP (c,'param')

identique à INITCAP en tenant compte de param (code de pays).

NLS_LOWER (c,'param')

identique à LOWER en tenant compte de param (code de pays).

NLSSORT (c)

séquence d'octets utilisés pour le tri de la chaîne c.

NLS_UPPER (c,'param')

identique à UPPER en tenant compte de param (code de pays).

REPLACE (c1,c2,c3)

remplace dans c1 tous les c2 par c3 (c3 peut être nul).

RPAD (c1,n[,c2])

renvoie c1 sur une longueur n complété à droite avec c2 (par défaut ' ').

RTRIM (c1[,c2])

renvoie c1 avec suppression à droite de tous les c2 (par défaut ' ').

SOUNDEX (c)

représentation phonétique de c.

SUBSTR (c,n1[,n2])

renvoie une chaîne formée par n2 caractères de c à partir de la position n1.

SUBSTRB (c,n1[,n2])

identique à SUBSTR avec n1 et n2 en octets.

TRANSLATE (c1,c2,c3)

remplace dans c1 tous les c2 par c3 (c3 ne peut pas être nul).

TRIM ({LEADING | TRAILING | BOTH c1} | {c1}) FROM c2)

le caractère c1 est supprimé de la chaîne c2 soit en tête de chaîne (LEADING), soit en fin de chaîne (TRAILING), soit les deux (BOTH ou aucun mot clé).

UNISTR (c)

la chaîne de caractères passée en paramètre est convertie au format unicode.

UPPER(c)

renvoie c en majuscules.

5.3 Fonctions scalaires dates

5.3.1 Explications sur quelques fonctions

ADD_MONTHS (d,n)

date d plus n mois (n peut être négatif).

CURRENT_DATE

pour connaître la date et l'heure actuelle en prenant en compte le paramétrage de la zone horaire (TIME_ZONE) de la session.

CURRENT_TIMESTAMP

pour connaître la date et l'heure relative à la plage horaire de la session. Cette fonction retourne une valeur de type TIMESTAMP WITH TIME ZONE.

DBTIMEZONE

pour connaître la zone horaire en vigueur sur la base de données.

EXTRACT (format FROM d)

extraire un élément (jour, mois, année, heure, minute, seconde...) depuis un élément de type date ou bien un intervalle de temps calculé à l'aide des fonctions NUMTODSINTERVAL ou NUMTOYMINTERVAL.

FROM_TZ (t, zone_horaire)

convertit une valeur de type TIMESTAMP en valeur de type TIMESTAMP WITH TIME ZONE.

LAST_DAY (d)

date du dernier jour du mois de d.

LOCALTIMESTAMP

pour connaître la date et heure courante par rapport à la zone horaire de la session. Cette fonction retourne une valeur de type TIMESTAMP.

MONTHS_BETWEEN (d1,d2)

différence en mois entre les deux dates.

NEW_TIME (d,z1,z2)

la date d de la zone z1 est convertie en date de la zone z2.

NEXT_DAY (d,j)

date du prochain jour j à partir de d.

NUMTODSINTERVAL (n,format)

convertit le nombre n passé en paramètre au format date. Le second argument représente l'unité dans laquelle n est exprimé. Les valeurs possibles sont 'DATE', 'HOUR', 'MINUTE' et 'SECOND'.

NUMTOYMINTERVAL (n, format)

convertit le nombre n passé en paramètre au format mois, année. Le format indiqué en second paramètre permet de préciser si le premier paramètre représente des mois ('MONTH') ou des années ('YEAR').

ROUND (d, format)

arrondit la date d à la date de début de l'élément donné en format le plus proche (Année, Mois, Jour, Heure...).

SESSIONTIMEZONE

pour connaître la zone horaire active dans la session actuelle.

SYSDATE

date système.

SYS_EXTRACT_UTC (d)

convertit une date et heure d'une zone horaire spécifique en date et heure au format UTC (*Universal Time Coordinated*) c'est-à-dire à l'heure de Greenwich.

SYSTIMESTAMP

pour connaître la date et l'heure, y compris les fractions de secondes, en s'appuyant sur la zone horaire configurée sur le serveur de base de données.

TRUNC (d, format)

tronque la date d au début de l'élément donné en format (Année, Mois, Jour, Heure...).

TZ_OFFSET (zone|format)

permet de connaître la zone horaire correspondant à la zone passée en paramètre.

5.3.2 Les formats de dates

Les formats de dates sont utilisés par les fonctions de conversion (TO_CHAR, TO_DATE) et les fonctions qui manipulent les dates (TRUNC, ROUND).

On trouve ci-dessous un tableau récapitulatif des principaux indicateurs de format de dates qui peuvent être utilisés.

- / ' . : ; 'texte'	Éléments de séparation présents dans le résultat.
D	Numéro du jour dans la semaine (1 à 7). Cet indicateur est à utiliser uniquement avec la fonction TO_DATE.
DAY	Nom du jour de la semaine.

DD	Numéro du jour dans le mois (1 à 31).
DDD	Numéro du jour dans l'année (1 à 366).
DY	Nom abrégé du jour de la semaine.
HH, HH12	Heure du jour (1 à 12).
HH24	Heure du jour (0 à 23).
IW	Numéro de la semaine dans l'année (1 à 53) basé sur la norme ISO.
MI	Minute (0 à 59).
MM	Numéro du mois sur 2 caractères (1 à 12).
MON	Nom abrégé du mois.
MONTH	Nom du mois.
Q	Numéro du trimestre (1 à 4).
RM	Numéro du mois en chiffre romain (I à XII).
SS	Secondes (0 à 59).
SSSS	Nombre de secondes depuis minuit (0 à 86399).
WW	Numéro de la semaine dans l'année. La semaine n°1 commence le premier jour de l'année et dure 7 jours.
W	Numéro de la semaine dans le mois (1 à 5). La première semaine du mois commence le premier jour du mois et se termine le septième jour.
YYYY	Année sur 4, 3, 2 ou 1 chiffre(s).
YYY	
YY	
Y	

5.4 Fonctions scalaires de conversion

BIN_TO_NUM (expr[, ...])

conversion d'un ensemble de nombres binaires en son équivalent décimal.

CHARTOROWID (char)

conversion de chaîne de caractères de forme 'nobloc.nolig.nofich' en type ROWID. Cette pseudo-colonne représente l'adresse physique de la ligne dans la base Oracle.

CONVERT (char,destination[,source])

conversion de jeu de caractères.

source ou **destination**

US7ASCII (défaut), WE8DEC (Dec Europe), WE8JHP (HP Europe), F7DEC (français DEC), WE8EBDCIC500 (IBM Europe Code page 500), WE8PC850 (IBM PC Code page 850), WE8ISO8859P1 (ISO8859-1 Europe 8 bits).

HEXTORAW (char)

conversion d'une valeur hexadécimale contenue dans une chaîne de caractères en une expression binaire de type RAW.

RAWTOHEX (raw), RAWTONHEX (raw)

fonction inverse de la précédente.

ROWIDTOCHAR(rowid), ROWIDTONCHAR (rowid)

fonction inverse de CHARTOROWID.

TO_CHAR (caractère)

conversion d'une chaîne de caractères au format NCHAR, NVARCHAR2, CLOB ou NCLOB dans le jeu de caractères utilisée dans la base de données.

TO_CHAR (n,[format[, 'param']])

conversion numérique en caractères.

format : chaîne de caractères représentant le format numérique (les éléments possibles de cette chaîne de caractères sont présentés dans le tableau des formats de date vu précédemment).

param : convention de pays.

TO_CHAR (d, format)

conversion d'une date en chaîne de caractères.

TO_CLOB (chaîne)

conversion d'une chaîne de caractères ou de type LOB au format CLOB.

TO_DATE (char, format)

fonction inverse de TO_CHAR (caractère en date).

TO_DSINTERVAL (chaîne)

conversion d'une chaîne de caractères en données de type intervalle Jour/Seconde.

TO_NCHAR (c)

conversion d'une chaîne de caractères dans le jeu de caractères national (deuxième jeu de caractères de la base).

TO_NCHAR (d,format)

conversion d'une date en chaîne, dans le jeu de caractères national.

TO_NCHAR (n)

conversion d'un nombre en chaîne de caractères dans le jeu de caractères national.

TO_NCLOB (c)

conversion d'une chaîne de caractères en élément de type NCLOB.

TO_NUMBER (char)

conversion d'une chaîne de caractères en numérique.

TO_TIMESTAMP (c[,format])

conversion d'une chaîne de caractères en élément de type TIMESTAMP en s'appuyant sur le format de la date indiqué en second paramètre.

TO_TIMESTAMP_TZ (c[,format])

conversion d'une chaîne de caractères en élément de type TIMESTAMP WITH TIME ZONE en s'appuyant sur le format de la date indiqué en second paramètre.

TO_YMINTERVAL (chaîne)

conversion d'une chaîne de caractères en données de type intervalle Mois/Année.

Exemple d'utilisation de fonctions de conversion

Dans l'exemple ci-dessous, la fonction SYSDATE permet de connaître la date et l'heure actuelle puis ces informations sont affichées proprement à l'aide de la fonction de conversion de TO_CHAR. Pour convertir la date en chaîne de caractères, on utilise un format de conversion dont les indicateurs sont donnés dans le tableau précédent.

```
SQL> select to_char(sysdate,'DAY DD MONTH YYYY HH24:MI:SS') from dual;

TO_CHAR(SYSDATE,'DAYDDMONTHYYYYHH24
-----
LUNDI    23 AVRIL    2001 11:39:01

SQL>
```

5.5 Fonctions de comparaisons

NULLIF (expr1, expr2)

Compare expr1 et expr2. Si les deux expressions sont égales, alors la valeur NULL est retournée, sinon expr1 est retournée. Il n'est pas possible de préciser la valeur NULL dans expr1.

5.6 Fonctions scalaires diverses

BITAND (arg1, arg2)

effectue une opération ET entre les deux arguments bit par bit.

COALESCE (expr,[, ...])

retourne la première expression non nulle passée en paramètre.

DECODE (colonne, valeur 1, resul1 [,valeur2, resul2 ...],[défaut])

si colonne a la valeur valeur 1, la forme sera resul1.

DUMP (exp[,format[,départ[longueur]]])

les formats possibles sont :

8 (le résultat est exprimé en octal),

10 (le résultat est exprimé en décimal),

16 (le résultat est exprimé en hexadécimal),

17 (le résultat est exprimé en caractères).

Les options départ et longueur permettent de préciser la partie de la chaîne à traiter.

GREATEST (exp[,exp...])

renvoie la plus grande des expressions.

LEAST (exp[,exp.....])

renvoie la plus petite des expressions.

NVL (exp1, exp2)

si expression1 est NULL renvoie expression2.

NVL2 (exp1, exp2, exp3)

si expression1 est NULL, alors la fonction NVL2 retourne expression2, sinon c'est expression3 qui est retournée. Les expressions 2 et 3 peuvent être de n'importe quel type de données à l'exception du type LONG.

UID

numéro identificateur de l'utilisateur.

USER

nom de l'utilisateur.

USERENV (opt)

renvoie des caractéristiques de l'environnement.

opt : ENTRY ID ; SESSION ID ; TERMINAL ; LANGUAGE ; LANG ; ISDBA ; CLIENT_INFO.

VSIZE (exp)

nombre d'octets stockés pour exp.

WIDTH_BUCKET (expr, mini, maxi, n)

construit n intervalles équivalents sur la plage mini à maxi puis indique dans quel intervalle se trouve chaque expression.

5.7 Fonctions de GROUPE

(ou fonctions statistiques, ou fonctions d'agrégat)

coln : Colonne numérique.

colonne : Colonne de tout type.

AVG (coln)

Moyenne des valeurs de colonne.

COUNT (colonne)

Pour chaque regroupement, nombre de lignes regroupées où colonne est non NULL.

COUNT (*)

Pour chaque regroupement, nombre de lignes regroupées.

MAX (colonne)

Valeur maximum de la colonne pour chaque regroupement.

MIN (colonne)

Valeur minimum de la colonne pour chaque regroupement.

STDDEV (coln)

Écart-type des valeurs de colonne pour chaque regroupement.

SUM (coln)

Somme des valeurs de colonne pour chaque regroupement.

VARIANCE (coln)

Variance des valeurs de colonne pour chaque regroupement.

CORR (col1, col2)

Coefficient de corrélation entre les 2 colonnes.

COVAR_POP (col1, col2)

Covariance d'une population.

COVAR_SAMP (col1, col2)

Covariance d'un échantillon.

CUME_DIST (col, pourcentage)

Distribution cumulative.

DENSE_RANK (col)

Numéro d'ordre d'une ligne dans un ensemble ordonné de lignes.

FIRST (col) LAST (col)

Retourne la première (dernière) ligne ordonnée par rapport à un critère tri (DENSE_RANK par exemple).

GROUP_ID

Distingue les groupes dupliqués apparus suite à l'application de la clause GROUP BY.

GROUPING (expr)

Permet d'identifier les lignes de cumul lorsque les instructions ROLLUP ou CUBE sont utilisées.

GROUPING_ID (expr)

Comme pour la fonction GROUPING, permet d'identifier si la ligne est ou non le résultat d'une commande ROLLUP ou CUBE.

PERCENTILE_CONT (expr)

Distribution inverse continue.

PERCENTILE_DISC (expr)

Distribution inverse discrète.

PERCENT_RANK (expr)

Similaire à CUME_DIST.

RANK (expr)

Calcule le rang d'une valeur dans un groupe de valeurs.

REGR_(expr)

Calcul de la régression linéaire.

STDDEV_POP (expr)

Écart-type d'une population.

STDDEV_SAMP (expr)

Écart-type d'un échantillon.

VAR_POP (expr)

Variance d'une population.

VAR_SAMP (expr)

Variance d'un échantillon.

Rq : Dans le cas où le calcul d'agrégats porte sur une colonne particulière, les valeurs NULL de cette colonne ne sont pas prises en compte pour les calculs.

Rq : Oracle dispose de quelques autres fonctions de calculs d'agrégats qui sont spécialisées dans le calcul statistique.

Exemples

Quel est le nombre de clients par ville ?

L'attribut ville ayant été défini de type CHAR(30), il faut éliminer les éventuels espaces à droite pour que le regroupement se fasse bien sur les caractères significatifs de la valeur de la colonne.

```
SQL> select rtrim(VILLE), count(*) from CLIENTS group by rtrim(VILLE);
```

RTRIM(VILLE)	COUNT(*)
LE MANS	1
NANTES	3
PARIS	1
TOURS	1

```
SQL>
```

Prix le plus élevé et moyenne des quantités en stock par famille d'articles (deux premiers caractères de la référence) :

```
SQL> select * from articles;
```

REFA	DESIGNATION	PRIX	CODETVA	CATEGORIE	QTESTK
AB22	Taps Persan	1375,11	2	IMPORT	5
CD50	Chaine HiFi		2	IMPORT	7
CD21	Platine laser	500	2	IMPORT	20
ZZZZ	Article bidon			DIVERS	25
AA00	Cadeau		0	DIVERS	8
AB03	Carpette	165	2	SOLDES	116
AB	Tapis		2	DIVERS	2
ZZ01	Lot de tapis	500	2	DIVERS	0
AB10	Tapis de chine	1650	2	IMPORT	10

9 ligne(s) sélectionnée(s).

```
SQL> select substr(refart,1,2), max(prix), avg(qtestk)
2 from articles
```

```
3 group by substr(refart,1,2);
```

```
SU  MAX(PRIX)  AVG(QTESTK)
```

```
--  -----  -----
```

AA		8
AB	1650	33,25
CD	500	13,5
ZZ	500	12,5

```
SQL>
```