

FUNCTȚII

INTRODUCERE

Funcțiile sunt folosite pentru manevra datelor. Ele acceptă unul sau mai multe argumente și întorc o valoare. Un argument este o constantă, variabilă sau o referire de coloană. Formatul pentru funcție este următorul: `function_name (arg1,arg2,...)`

Funcțiile pot fi folosite pentru:

- a calcula
- a modifica datele individuale
- a schimba formatul datelor pentru afisare
- a converti tipurile de date

Există diferite tipuri de funcții:

- pentru șiruri de caractere
- pentru valori numerice
- pentru date calendaristice
- de conversie
- universale
- de grup

Unele funcții operează pe un singur șir, altele pe grupuri de șiruri.

Funcții care se aplică unui singur șir.

- acționează asupra fiecărui șir returnat din cerere
- întoarce un rezultat pentru fiecare șir
- așteaptă unul sau mai multe argumente
- pot fi imbricate
- pot fi folosite acolo unde se folosesc variabile, coloane sau expresii; de exemplu, în construcțiile: `SELECT`, `WHERE`, `ORDER BY`.

Explicația notației:

Notație + Semnificație:

`col`

un nume de coloana dintr-o bază de date

`value(valoare)`

orice valoare literale (caracter/data/număr)

`n`

reprezintă o valoare numerică

`'string'`

reprezintă caracterele string
chars
reprezintă un număr de caractere specificate
date
reprezintă o dată calendaristică

Funcții numerice și pe caractere

Acestea acceptă la intrare un caracter și poate returna caractere sau valoare numerică.

Toate funcțiile pot fi testate utilizând tabela DUAL

LOWER(col/value)
trece la caractere scrise cu litere mici.

Exemplu :

```
Select LOWER('SQL course') from dual
```

UPPER(col/value)
trece la caractere scrise cu litere mari.

INITCAP(col/value)
trece prima litera a fiecarui cuvint la litera mare

CONCAT(str1, str2)
intoarce str1 concatenat cu str2 (Alternativa a operatorului ||)

Funcțiile LPAD și RPAD aduc sirurile de caractere la o lungime specificată.

LPAD(col/value,n,'string')
adauga la coloana sau la valoarea literală spre stânga până la lungimea totală n. Primele spații sunt umplute cu 'string'. Dacă 'string' e omis atunci sunt umplute cu blanuri.

```
SELECT LPAD('NUME',20,'*'),LPAD('NUME',20),LPAD('NUME',20,'. ')  
FROM dual;
```

RPAD(col/value,n,'string')
adauga la coloana sau la valoarea literală spre dreapta până la lungimea totală n. Ultimele poziții sunt umplute cu 'string' sau dacă acesta e omis cu blanuri.

În acest caz a doua coloană are în dreapta blanuri puse implicit.

Urmatoarele functii presupun numerotarea caracterelor din string de la stinga la dreapta, incepind cu 1.

SUBSTR(col/value,pos,n)

intoarce un string de n caractere lungime dintr-o coloana sau valoare literala, incepind de la pozitia pos. Daca n e omis, e extras sirul din pozitia pos la sfirsit.

INSTR(col/value,'string')

gaseste pozitia caracterului de la care apare prima data 'string'.

INSTR(col/value,'string',pos,n)

gaseste pozitia caracterului de la care incepe a n-a aparitie a lui string in coloana sau valoarea literala incepind din pozitia pos.

LTRIM si **RTRIM** sterg caracterele specificate din sir.

LTRIM (col/value,'char/s')

sterge de la stinga prima aparitie a caracterului specificat (sau o combinatie a caracterelor specificate). Daca nu e specificat nici un caracter sterge toate blancurile din stinga.

RTRIM (col/value,'char/s')

sterge de la dreapta, aparitiile lui char (sau combinatie de caractere specificate). Daca nu e specificat 'char/s', atunci sterge blancurile.

RTRIM poate fi in mod particular folositor in a sterge blancurile de la sfirsit la coloane.

LENGTH(col/value)

intoarce numarul de caractere din coloana sau din valoarea literala.

Functiile **TRANSLATE** si **REPLACE** sunt folosite pentru substitutia caracterelor.

TRANSLATE(col/value,from,to) translateaza la iesire caracterele 'from' la cele 'to'. Toate aparitiile din 'from' sunt inlocuite de corespondentele din 'to'.

REPLACE(col/value,string,replacement_string)

orice aparitie a 'string-ului', este inlocuita cu replacement_string. Daca 'replacement_string' e omis, toate aparitiile lui 'string' sunt sterse. Daca sunt omise si 'string' si 'replacement_string', atunci se semnaleaza eroare.

Functia **REPLACE** suplimenteaza functionalitatea furnizata de functia **TRANSLATE**. **TRANSLATE** furnizeaza caractere singulare, unul cite unul, pentru substitutie.

REPLACE va permite sa substituiti un sir cu altul ca si sa stergeti un sir (daca nu specificati 'replacement_string'). Observati ca 'string' si 'replacement_string' pot fi de orice lungime. REPLACE poate sa faca si conversia caracter la caracter REPLACE('C','T').

FUNCTII NUMERICE

Acestea accepta intrare numerica si intorc ca rezultat un numar.

ROUND(col/value,n) rotunjeste la n zecimale. Daca n e omis se rotunjeste la cel mai apropiat intreg, daca e negativ, se aplica rotunjirea la inlocuirea cu 0 a ultimelor -n cifre .

TRUNC(col/value,n) trunchiaza la n zecimale, sau daca n e omis, trunchiaza prin lipsa la intreg. Daca n e negativ, se aplica trunchierea la inlocuirea cu 0 a ultimelor -n cifre.

CEIL (col/value) gaseste cel mai mic intreg mai mare sau egal cu coloana, expresie sau valoare.

FLOOR (col/value) gaseste cel mai mare intreg mai mic sau egal cu coloana expresie sau valoare.

POWER(col/value,n) ridica coloana, expresia sau valoarea la puterea n. Poate fi negativa.

EXP(n) intoarce e ridicat la puterea n e=2.71828183.

```
SELECT      EXP(4)
FROM        DUAL;
```

```
EXP(4)
-----
54.59815
```

SQRT(col/value) gaseste radacina patrata a coloanei sau valorii. Daca col/value e NULL sau negativ atunci e intors rezultatul NULL.

SIGN(col/value) intoarce -1 daca e coloana, expresie sau valoare e un nr. negativ , intoarce 0 daca e zero, +1 daca e nr.pozitiv.

Frecvent, functia SIGN este folosita pentru a compara doua valori. Urmatorul exemplu tipareste toti angajatii al caror salariu este mai mare decit comisionul lor.

```
select salary, commission_pct
from employees
where sign(salary-commission_pct)=1          ENAME          SAL          COMM
```

ABS(col/value) gaseste valoarea absoluta a coloanei sau valorii.

MOD(value1,value2) gaseste restul impartirii lui 'value1' la 'value2'.

Alte citeva functii matematice:

LOG(m,n)

intoarce logaritmul cu baza m si argument n.

SIN(n)

intoarce sinusul lui n

SINH(n)

intoarce sinusul hiperbolic al lui n

TAN(n)

intoarce tangenta lui n

TANH(n)

intoarce tangenta hiperbolica a lui n

COS(n)

cosinus de n

COSH(n)

cosinus hiperbolic al lui n

Functii de grup

Acest capitol explica cum informatii concise pot fi obtinute pentru grupuri de articole, folosind functiile de grup.

Functiile de grup se aplica unui set de articole. Ele intorc un rezultat bazat pe grupuri de articole, spre deosebire de un rezultat pentru fiecare articol. De regula, toate articolele din tabela sunt tratate ca un grup. Clauza GROUP BY din cadrul lui SELECT e folosita pentru a imparti articolele in mai multe grupuri.

Functii de grup:

AVG({DISTINCT} expr)

valoarea medie a expresie (ignorind valorile NULL!).

COUNT({DISTINCT} expr)

cate articole evalueaza expresia la altceva decit NULL.

COUNT(*)

cate articole sunt in grup.

MAX({DISTINCT} expr)

maximul valorilor expr.

MIN({DISTINCT} expr)

minimul valorilor expr.

SUM({DISTINCT} n)

suma valorilor expr (ignorind valorile NULL!).

DISTINCT face ca functia de grup sa considere doar valorile neduplicate. Toate fuctiile de grup cu exceptia lui COUNT (*) ignora valorile NULL.

Folosirea functiilor de grup

Pentru a calcula salariul mediu al tuturor angajatilor, introducem:

```
SELECT AVG (SALARY)
FROM EMPLOYEES;
```

Aici toate articolele din tabela EMP sint considerate ca un singur grup.

O functie de grup poate fi aplicata pe un subset de articole din tabela prin folosirea clauzei WHERE.

Pentru a numara angajatii din departamentul 20, introducem :

```
SELECT COUNT (*)
FROM EMPLOYEES
WHERE DEPARTMANT_ID = 20;
```

Clauza GROUP BY

Clauza GROUP BY poate fi folosita pentru a imparti articolele dintr-o tabela in grupuri. Functiile de grup pot fi folosite pentru a intoarce informatii pentru fiecare grup.

Pentru a calcula salariul mediu pentru fiecare slujba, introducem:

```
SELECT JOB_ID, AVG (SALARY)
FROM EMPLOYEES
GROUP BY JOB_ID;
```

JOB	AVG (SAL)
ANALYST	3000
MANAGER	2758.33333
PRESIDENT	5000

Exercitii

Formulati cererile:

1.

```
SELECT MIN(SAL) MINIMUM din angajati
```
2.

```
SELECT MAX(SAL), MIN(SAL), AVG(SAL) din angajati
```

3.

```
SELECT JOB, MAX(SAL) MAXIMUM, MIN(SAL) MINIMUM din angajati
```

4.

```
SELECT JOB, AVG(SAL) AVSAL, AVG(SAL*12 + NVL(COMM, 0)) AVCOMP  
Din angajati
```

5.

```
SELECT MAX(SAL)-MIN(SAL) DIFFERENCE  
Din angajati
```

6.

```
SELECT DEPARTMENT_ID, COUNT(*) din angajati
```