

# Laboratorul 3

De la BD

**Tematică:** lucrul cu funcții linie în SQL

## Cuprins

- 1 Funcții linie
  - 1.1 Funcții numerice
  - 1.2 Funcții ce iau ca argument date caracter și returnează valori numerice
  - 1.3 Funcții ce iau ca argument date caracter și returnează date de tip caracter
  - 1.4 Funcții pentru tipul DATE (date calendaristice)
  - 1.5 Funcții de conversie
  - 1.6 Alte funcții
- 2 Exerciții

## Funcții linie

Funcțiile de tip linie sunt funcții care, utilizate în interogări, sunt apelate pentru fiecare linie și returnează un rezultat pe linie pentru tabelul interogat. Argumentul funcțiilor poate fi o simplă constantă dată de utilizator, numele unei variabile, numele unei coloane sau o expresie. Pot fi utilizate în cadrul listei SELECT, în clauzele WHERE, ORDER BY. Apelate cu argument NULL funcțiile returnează NULL. Excepție fac CONCAT, NVL, REPLACE și REGEXP\_REPLACE.

Cele mai multe dintre funcțiile listate în continuare sunt specifice dialectului SQL de la Oracle.

### Funcții numerice

primesc ca argument și returnează date de tip numeric

CEIL (n)	Rotunjește superior pe n	SELECT CEIL(51.3) FROM DUAL;
FLOOR(n)	Rotunjește inferior pe n	SELECT FLOOR(51.3) FROM DUAL;
ROUND(n [,m])	Rotunjește pe n la m zecimale. Implicit m este 0, echivalent cu rotunjirea la cel mai apropiat întreg. m poate fi și negativ.	SELECT ROUND(21.365,2) AS "Rotunjire - pozitiv" FROM DUAL;  SELECT ROUND (21.665,-1) AS " Rotunjire - negativ " FROM DUAL;
TRUNC(n [,m])	Trunchează pe n la m zecimale; implicit m e 0; m poate fi și negativ.	SELECT TRUNC(21.365,2) AS "Trunchiere - pozitiv" FROM DUAL;

		SELECT TRUNC(21.665,-1) AS "Trunchiere - negativ" FROM DUAL;
MOD(m, n)	Returnează restul împărțirii lui m la n	SELECT MOD (26,11) AS "Mod" FROM DUAL;

La acestea se mai adaugă funcțiile trigonometrice uzuale (COS(n), ACOS(n), SIN(n), ASIN(n), TAN(n), ATAN(n)), funcțiile de calcul de logaritmi (LN(n), LOG(base, n)), ridicarea la putere (EXP(n), POWER(m,n), rădăcina pătrată (SQRT(n)).

### Funcții ce iau ca argument date caracter și returnează valori numerice

LENGTH (char)	Returnează lungimea șirului char.	SELECT LENGTH('cuvant') AS "Lungime cuvant" FROM DUAL;  SELECT prenume, LENGTH(prenume) FROM profesori WHERE UPPER(TRIM(prenume))='COSMIN'; -- de ce 10?
ASCII (char)	Returnează codul ASCII al primului caracter din șir	SELECT ASCII('A') FROM DUAL;
INSTR(char1, char2 [, n [, m ]])	Caută a m-a apariție a șirului char2 în șirul char1 începând căutarea de la poziția n; returnează poziția în char1 a primului caracter din subșirul identificat, relativ la n. Implicit n și m sunt 1. Dacă nu se identifică nici o apariție rezultatul este 0.	SELECT INSTR('CORPORATE FLOOR','OR',3,2) AS "Instring" FROM DUAL;  SELECT nume, INSTR(nume,'ra') FROM profesori;

### Funcții ce iau ca argument date caracter și returnează date de tip caracter

INITCAP(char)	Prima literă a fiecărui cuvânt e transformată în majusculă.	SELECT INITCAP('prima litera majuscula') FROM DUAL;
UPPER(char)	Returnează șirul char rescris cu majuscule	SELECT UPPER('Stefan') FROM DUAL;
LOWER(char)	Returnează șirul char rescris cu litere mici	SELECT LOWER('LOWER') FROM DUAL;
CONCAT(char1, char2)	Concatenează șirurile argument echivalent cu operatorul de concatenare	SELECT CONCAT( CONCAT(nume, ' este student in anul '), an) FROM studenti;

SUBSTR(char, m [, n ])	Extrage din șirul char n caractere începând de la poziția m; omisiunea lui n semnifică finalul șirului; o valoare negativă pentru m semnifică numărarea de la stânga la dreapta	SELECT SUBSTR('ABCDEFGH',3,4) FROM DUAL;
RPAD(char1,n [,char2 ])	Returnează char1 cu spații adăugate la dreapta până la lungimea n, dacă se omite char2; altfel char2 este copiat de oricâte ori e nevoie pentru a completa char1 până la lungimea n	SELECT RPAD('nume',12,'ab') FROM studenti WHERE nume = 'Popescu';
LPAD(char1,n [,char2 ])	Similar RPAD cu diferența că alipirea se face la stânga	SELECT LPAD('Pagina',15,'*.') FROM DUAL;
CHR(n)	Returnează caracterul corespunzător codului binar n, ca tip VARCHAR2	SELECT CHR(68)  CHR(65)  CHR(84)  CHR(69) FROM DUAL;
TRIM (char)	Elimină spațiile goale de la începutul și sfârșitul textului	
LTRIM(char)	Elimină spațiile goale de la începutul textului	
REPLACE(char, search_string [, replacement_string])	Returnează șirul char în care subșirul search_string este înlocuit de replacement_string; absența ultimului reprezintă ștergerea tuturor aparițiilor subșirului căutat	SELECT REPLACE('Andrei','i','ea') FROM DUAL;
TRANSLATE(char, from, to)	Returnează char modificat astfel: fiecare apariție a unui caracter din șirul from este înlocuită de corespondentul (caracterul de pe	SELECT TRANSLATE('48KLM980', '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ', '9999999999XXXXXXXXXXXXXXXXXXXXXXXXXXXXX') "COD" FROM DUAL;

	aceeași poziție) din șirul to; dacă from este mai lung decât to, caracterele fără corespondent sunt șterse	
--	---	--

Funcțiile care schimbă cazul literelor (transformă textul în litere mici sau în majuscule) sunt frecvent utilizate în clauza WHERE pentru a regăsi text. De exemplu, dacă dorim identificarea datelor despre studentul Andrei și nu avem garanția că prenumele acestuia a fost inserat în baza de date cu toate litere majuscule sau doar cu prima literă majusculă, vom procesa atributul *prenume* aducându-l la forma dorită pentru testare:

```
SELECT * FROM studenti WHERE prenume = 'ANDREI'; --returneaza 0 linii
```

```
SELECT * FROM studenti WHERE UPPER(prenume) = 'ANDREI'; --returneaza înregistrările dorite
```

De asemenea funcția TRIM are o largă utilizare pentru regăsirea datelor atunci cand attributele sunt declarate de tip CHAR:

```
SELECT * FROM profesori WHERE UPPER(ume) = 'BREABAN'; --returneaza 0 linii
```

```
SELECT * FROM profesori WHERE TRIM(UPPER(ume)) = 'BREABAN'; --returneaza înregistrările  
dorite
```

## Funcții pentru tipul DATE (date calendaristice)

Formatul implicit pentru datele calendaristice este "DD-MON-YY" (ex: '10-JUN-15'), însă acesta poate fi modificat de către utilizator (ALTER SESSION SET NLS\_DATE\_FORMAT...). Pentru a depăși problemele de incompatibilitate, este recomandat ca lucrul cu constante de tip date calendaristice să facă apel la funcția TO\_DATE (descrișă în secțiunea următoarea) care va specifica formatul acesteia.

CURRENT_DATE	Returnează data curentă relativ la zona de timp (time zone) a sesiunii curente	SELECT CURRENT_DATE FROM DUAL;
SYSDATE	Returnează data curentă a sistemului	SELECT SYSDATE FROM DUAL;
MONTHS_BETWEEN(d1, d2 )	Numărul de luni dintre d1 și d2	SELECT MONTHS_BETWEEN( TO_DATE('02-02-2015','MM-DD-YYYY'), TO_DATE('01-01-2015','MM-DD-YYYY') ) AS "Months" FROM DUAL;
ADD_MONTHS(d, n)		SELECT data_nastere, ADD_MONTHS(data_nastere,1) FROM studenti WHERE nume= 'Popescu';
NEXT_DAY(d, char)	Returnează data calendaristică corespunzătoare zilei lucrătoare specificate de char imediat	SELECT NEXT_DAY('13- OCT-15','TUESDAY') FROM DUAL;

	urmatoare datei d	
LAST_DAY(d)	Returnează data calendaristică corespunzătoare ultimei zile din luna specificată de d	SELECT SYSDATE, LAST_DAY(SYSDATE) AS "Last" FROM DUAL;
ROUND(d [, fmt ])	Rotunjește la cea mai apropiată dată calendaristică conform formatului dat.	SELECT ROUND (TO_DATE ('27-OCT-15','DD-MON-YY'),'YEAR') "New Year" FROM DUAL;
TRUNC(d [, fmt ])	Dacă formatul este omis, trunchiază la cea mai apropiată zi.	SELECT TRUNC(TO_DATE('27-OCT-15','DD-MON-YY'), 'YEAR') "New Year" FROM DUAL;  SELECT TRUNC(TO_DATE('27-OCT-15','DD-MON-YY'), 'MONTH') "New Year" FROM DUAL;

Studiați formatul fmt la adresa [http://download.oracle.com/docs/cd/B19306\\_01/server.102/b14200/functions230.htm](http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/functions230.htm)

Oracle stochează datele într-un format numeric intern reprezentând secolul, anul, luna, ziua, orele minutele și secunde. Fiind stocate în format numeric, datelor calendaristice le pot fi aplicați operatori numerici:

- adunarea sau scăderea unui număr la/dintr-o dată are ca rezultat o nouă dată:

SELECT SYSDATE + 10 FROM dual; Ce semnifică valoarea adunată? Cum putem adăuga un număr de ore la o data calendaristică?

- diferența a două date are ca rezultat numărul de zile

SELECT (SYSDATE-data\_nastere)/365 FROM studenti;

## Funcții de conversie

Oracle face implicit următoarele conversii de tipuri:

VARCHAR2 -> NUMBER

CHAR -> NUMBER

VARCHAR2 -> DATE

CHAR -> DATE

NUMBER -> VARCHAR2

DATE -> VARCHAR2

Pentru alte conversii există funcțiile:

TO_CHAR(char)	Convertește o valoare de tip char la VARCHAR2	
---------------	---	--

TO_CHAR(n [, fmt])	Convertește o valoare numerică n la VARCHAR2 conform formatului specificat fmt. Studiați formatul numeric la adresa: <a href="http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34570">http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34570</a>	SELECT TO_CHAR('01110' + 1) FROM dual; SELECT TO_CHAR(bursa, '\$99,999') FROM studenti;
TO_CHAR({ datetime } [, fmt ])	Convertește datetime la VARCHAR2 conform formatului specificat fmt	SELECT TO_CHAR (SYSDATE, 'Day, Month, DD, YYYY') "TO_CHAR example" FROM DUAL;
TO_DATE(char [, fmt ])	Convertește un șir de caractere în dată calendaristică conform formatului specificat. Studiați modul de specificare a formatului la adresa: <a href="https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements004.htm#i34924">https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements004.htm#i34924</a>	SELECT TO_DATE('January 26, 1996, 12:38 A.M.', 'Month dd YYYY HH:MI A.M.') FROM DUAL;
TO_NUMBER(char [, fmt ])	<a href="http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34570">http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34570</a>	SELECT TO_NUMBER('100.52','9,999.99') FROM DUAL;
CAST(expr AS type_name)		SELECT CAST(CURRENT_TIMESTAMP AS VARCHAR(30)) FROM DUAL;
CONVERT(char, dest_char_set [, source_char_set ])	Convertește șirul char într-o nouă codificare. Codificările uzuale sunt:  -US7ASCII: US 7-bit ASCII character set -WE8DEC: West European 8-bit character set -F7DEC: DEC French 7-bit character set -WE8EBCDIC500: IBM West European EBCDIC Code Page 500 -WE8ISO8859P1: ISO 8859-1 West European 8-bit character set -UTF8: Unicode 4.0 UTF-8 Universal character set, CESU-8 compliant -AL32UTF8: Unicode 4.0 UTF-8 Universal character set	SELECT CONVERT('Ä Ê Í Ó Ø A B C D E ', 'US7ASCII', 'WE8ISO8859P1') FROM DUAL;

Pentru a studia formatarea datelor calendaristice in cadrul functiei to\_char studiat materialul de la adresa: [http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/sql\\_elements004.htm#i34924](http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34924)

## Alte funcții

USER	Returnează numele schemei curente	SELECT USER FROM DUAL;
NVL(expr1, expr2)	Dacă expr1 este NULL returnează expr2, altfel returnează expr1	SELECT nume, NVL(bursa,0) "bursa" FROM studenti; SELECT nume, NVL(TO_CHAR(bursa),'Nu are bursa') "bursa" FROM studenti;

NULLIF(expr1, expr2)	Dacă expr1 este egală cu expr2 returnează NULL, altfel returnează expr1	
DECODE(expr, search1, result1 [, search2, result2]... [, default ])	Compară expr cu fiecare valoare search pe rând. Dacă obține egalitate returnează valoarea result corespunzătoare. Dacă nici o potrivire nu e găsită este returnată valoarea DEFAULT; dacă aceasta nu e precizată returnează NULL. Dacă expr e NULL, se consider egalitate doar dacă search e NULL.	SELECT DECODE(bursa, 450, 'premiul 1', 350, 'premiul 2', 250, 'premiul 3', '?' ) FROM studenti;
LEAST(expr1 [, expr2 ]... )	Returnează cel mai mic element din listă	SELECT LEAST('Mihai','Andrei','Anca') "LEAST" FROM DUAL;
GREATEST (expr1 [, expr2 ]... )	Returnează cel mai mare element din listă	SELECT GREATEST('Mihai','Andrei','Anca') "GREATEST" FROM DUAL;

Funcția NVL este deseori utilă pentru a evita rezultatele nule la aplicarea unor operatori/funcții. De exemplu, dacă toți studenții facultății ar fi finanțați cu o sumă dată (de ex. 100), suma totală ridicată de un student nu se poate obține prin

```
SELECT 100+bursa FROM studenti;
```

ci prin

```
SELECT 100+NVL(bursa,0) FROM studenti;
```

## Exerciții

1. Scrieți o interogare pentru a afișa data de azi. Etichetați coloana "Astăzi".
2. Pentru fiecare student afișați numele, data de naștere și numărul de luni între data curentă și data nașterii.
3. Afișați ziua din săptămână în care s-a născut fiecare student.
4. Utilizând funcția de concatenare, obțineți pentru fiecare student textul 'Elevul <prenume> este în grupa <grupa>'.  
<br>
5. Afișați valoarea bursei pe 10 caractere, completând valoarea numerică cu caracterul \$.
6. Pentru profesorii al căror nume începe cu B, afișați numele cu prima literă mică și restul mari, precum și lungimea (nr. de caractere a) numelui.
7. Pentru fiecare student afișați numele, data de naștere, data la care studentul urmează să își sărbătorească ziua de naștere și prima zi de duminică de după.
8. Ordonați studenții care nu iau bursă în funcție de luna când au fost născuți; se va afișa doar numele, prenumele și luna corespunzătoare datei de naștere.
9. Pentru fiecare student afișați numele, valoarea bursei și textul: 'premiul 1' pentru valoarea 450, 'premiul 2' pentru valoarea 350, 'premiul 3' pentru valoarea 250 și 'mențiune' pentru cei care nu iau bursa. Pentru cea de a treia coloană dați aliasul "Premiu".
10. Afișați numele tuturor studenților înlocuind apariția literei i cu a și apariția literei a cu i.
11. Afișați pentru fiecare student numele, vârsta acestuia la data curentă sub forma '<x> ani <y> luni și <z>

zile' (de ex '19 ani 3 luni și 2 zile') și numărul de zile până își va sărbători (din nou) ziua de naștere.

12. Presupunând că în următoarea lună bursa de 450 RON se mărește cu 10%, cea de 350 RON cu 15% și cea de 250 RON cu 20%, afișați pentru fiecare student numele acestuia, data corespunzătoare primei zile din luna următoare și valoarea bursei pe care o va încasa luna următoare. Pentru cei care nu iau bursa, se va afișa valoarea 0.
13. Pentru studenții bursieri (doar pentru ei) afișați numele studentului și bursa în stelute: fiecare steluta valorează 50 RON. În tabel, aliniați stelutele la dreapta.

Adus de la „[http://85.122.23.37/BD/index.php?title=Laboratorul\\_3&oldid=341](http://85.122.23.37/BD/index.php?title=Laboratorul_3&oldid=341)”

- 
- Ultima modificare efectuată la 09:15, 29 octombrie 2015.
  - Conținutul este disponibil sub Creative Commons Atribuire, exceptând cazurile în care se specifică altfel.