

LABORATOR 2 SQL

FUNCȚII SQL (*single-row*)

Funcțiile SQL sunt predefinite în sistemul Oracle și pot fi utilizate în instrucțiuni SQL. Acestea nu trebuie confundate cu funcțiile definite de utilizator, scrise în PL/SQL.

- Dacă o funcție SQL este apelată cu un argument având un alt tip de date decât cel așteptat, sistemul convertește implicit argumentul înainte să evalueze funcția.
- Dacă o funcție SQL este apelată cu un argument *null*, atunci aceasta returnează valoarea *null*. Singurele funcții care nu urmează această regulă sunt CONCAT, NVL și REPLACE.
- Principalele funcții SQL pot fi clasificate în următoarele categorii:
 - Funcții *single-row*
 - Funcții *multiple-row* (funcții agregat)

Funcțiile single-row întorc câte o linie rezultat pentru fiecare linie a tabelului sau vizualizării interogate. Aceste funcții pot apărea în listele SELECT, clauzele WHERE, START WITH, CONNECT BY și HAVING.

1. Analizați următoarele funcții pentru prelucrarea șirurilor de caractere:

Funcție	Semnificație	Exemplu
LOWER (expr)	Convertește un șir de caractere la minuscule.	LOWER ('AbCd') = 'abcd'
UPPER (expr)	Convertește un șir de caractere la majuscule.	UPPER ('AbCd') = 'ABCD'
INITCAP (expr)	Convertește un șir de caractere la un șir care începe cu majusculă și continuă cu minuscule.	INITCAP ('AbCd') = 'Abcd'
CONCAT (expr1, expr2)	Concatenează două expresii de tip șir de caractere. Echivalent cu operatorul de concatenare ' '.	CONCAT ('Ab' , 'Cd') = 'AbCd'
SUBSTR (expr, m[, n])	Extrage din expresia <i>expr</i> <i>n</i> caractere începând cu poziția <i>m</i> . Dacă lipsește argumentul <i>n</i> , atunci extrage toate caracterele până la sfârșitul șirului. Dacă <i>m</i> este negativ numărătoarea	SUBSTR ('AbCdE' , 2, 2) = 'bC' SUBSTR ('AbCdE' , 2) = 'bCdE' SUBSTR ('AbCdE' , -3, 2) = 'Cd' SUBSTR ('AbCdE' , -3) = 'CdE'

	pozițiilor începe de la sfârșitul șirului de caractere spre început.	
LENGTH(<i>expr</i>)	Întoarce numărul de caractere al expresiei.	LENGTH('AbCdE')=5
INSTR(<i>expr1</i> , <i>expr2</i> [, <i>m</i>][, <i>n</i>])	Întoarce poziția la care se găsește a <i>n</i> -a ocurență a expresiei <i>expr2</i> în cadrul expresiei <i>expr1</i> , căutarea începând de la poziția <i>m</i> . Dacă <i>m</i> sau <i>n</i> lipsesc, valorile implicite sunt 1 pentru ambele.	INSTR(LOWER('AbC aBcDe'), 'ab', 5, 2)=0 INSTR(LOWER('AbCdE aBcD'), 'ab', 5)=7
LPAD(<i>expr1</i> , <i>n</i> [, <i>expr2</i>]) RPAD(<i>expr1</i> , <i>n</i> [, <i>expr2</i>])	Completează expresia <i>expr1</i> la stânga (LPAD), respectiv la dreapta (RPAD) cu caracterele specificate în expresia <i>expr2</i> , până la lungimea specificată de parametrul <i>n</i> . Dacă <i>expr2</i> lipsește se adaugă spații.	RPAD('AbC', 5, 'X')='AbCXX' LPAD('AbC', 5)=' AbC'
LTRIM(<i>expr1</i> [, <i>expr2</i>]) RTRIM(<i>expr1</i> [, <i>expr2</i>])	Trunchează expresia <i>expr1</i> la stânga (LTRIM) sau la dreapta (RTRIM) prin eliminarea succesivă a caracterelor din expresia <i>expr2</i> . Dacă <i>expr2</i> lipsește se elimină spații.	RTRIM('abcXX', 'X')='abc' LTRIM(' abc') = 'abc'
TRIM(LEADING <i>expr2</i> FROM <i>expr1</i>) TRIM(TRAILING <i>expr2</i> FROM <i>expr1</i>) TRIM(BOTH <i>expr2</i> FROM <i>expr1</i>) TRIM(<i>expr1</i>)	Permite eliminarea caracterelor specificate în <i>expr2</i> de la începutul (LEADING), sfârșitul (TRAILING) sau din ambele părți (BOTH) din expresia <i>expr1</i> .	TRIM(LEADING 'X' FROM 'XXabcdeXX') = 'abcdeXX' TRIM(TRAILING 'X' FROM 'XXabcdeXX') = 'XXabcde' TRIM(BOTH 'X' FROM 'XXabcdeXX') = 'abcde' TRIM(' abc ') = 'abcde'
REPLACE(<i>expr</i> , <i>expr1</i> , <i>expr2</i>)	Înlocuiește în șirul <i>expr</i> toate ocurențele șirului <i>expr1</i> cu șirul <i>expr2</i> .	REPLACE('%1%11', '%', '2') = '21211'

		<code>REPLACE ('%1%11', '%1', '2')</code> <code>= '221'</code> <code>REPLACE ('%1%11', '%') =</code> <code>'111'</code>
<code>TRANSLATE (expr, expr1, expr2)</code>	Fiecare caracter din șirul de caractere <i>expr</i> care apare și în <i>expr1</i> este transformat în caracterul corespunzător (aflat pe aceeași poziție ca și în <i>expr1</i>) din șirul de caractere <i>expr2</i> .	<code>TRANSLATE ('%1%11', '%', '2')</code> <code>= '21211'</code> <code>TRANSLATE ('%1%111', '%1', '23')</code> <code>= ' 232333'</code>
<code>ASCII (expr)</code>	Întoarce codul ASCII al primului caracter din <i>expr</i> .	<code>ASCII ('curs') = 99</code> <code>ASCII ('c') = 99</code>
<code>CHR (expr)</code>	Întoarce caracterul corespunzător codului ASCII specificat.	<code>CHR (99) = 'c'</code>

2. Afișați numele și data angajării salariatului având codul 200. Etichetați coloanele conform semnificațiilor acestora (Nume, Data angajării), fără ca aliasurile să fie trunchiate la dimensiunea coloanei și respectându-se forma de scriere a acestora (aliasurile încep cu majusculă).

Indicație: `RPAD (TO_CHAR (hire_date), 20, ' ')`

3. Afișați pentru fiecare angajat din departamentul 20 un șir de caractere de forma "Funcția salariatului {prenume} {nume} este {cod functie}" (prenumele cu inițiala majusculă, numele cu majuscule, codul funcției cu minuscule).
4. Afișați pentru angajatul cu numele 'HIGGINS' codul, numele și codul departamentului. Cum se scrie condiția din WHERE astfel încât să existe siguranța ca angajatul 'HIGGINS' va fi găsit oricum ar fi fost introdus numele acestuia? Căutarea trebuie să nu fie case-sensitive, iar eventualele blank-uri care preced sau urmează numelui trebuie ignorate.

Indicație: `UPPER (TRIM (last_name))`

5. Afișați pentru toți angajații al căror nume se termină în litera 'n', codul, numele, lungimea numelui și poziția din nume în care apare prima data litera 'a'. Asociați aliasuri coloanelor afișate.

Indicație: `LENGTH (last_name)`

`INSTR (UPPER (last_name), 'A')`

`SUBSTR (last_name, -1) = 'n'`

6. Analizați următoarele funcții aritmetice:

Funcție	Semnificație	Exemplu
<code>ROUND (expr [, n])</code>	Întoarce valoarea rotunjită a expresiei <i>expr</i> până la <i>n</i> zecimale. Dacă <i>n</i> este negativ sunt rotunjite cifre din stânga virgulei. Valoarea implicită pentru <i>n</i> este 0.	$\text{ROUND}(1.6) = 2$ $\text{ROUND}(1.4) = 1$ $\text{ROUND}(1234.56, 1) = 1234.6$ $\text{ROUND}(1230.56, -2) = 1200$ $\text{ROUND}(1260.56, -2) = 1300$
<code>CEIL (n)</code>	Întoarce partea întreagă superioară.	$\text{CEIL}(1.4) = 2$ $\text{CEIL}(1.6) = 2$ $\text{CEIL}(1234.56) = 1235$ $\text{CEIL}(-1234.56) = -1234$
<code>FLOOR (n)</code>	Întoarce partea întreagă inferioară.	$\text{FLOOR}(1.4) = 1$ $\text{FLOOR}(1.6) = 1$ $\text{FLOOR}(1234.56) = 1234$ $\text{FLOOR}(-1234.56) = -1235$
<code>TRUNC (expr [, n])</code>	Întoarce valoarea trunchiată a expresiei <i>expr</i> până la <i>n</i> zecimale. Dacă <i>n</i> este negativ sunt trunchiate cifre din stânga virgulei. Valoare implicită pentru <i>n</i> este 0.	$\text{TRUNC}(1234.56) = 1234$ $\text{TRUNC}(123456, -2) = 123400$ $\text{TRUNC}(1234.56) = 1234$ $\text{TRUNC}(1234.56, 1) = 1234.5$
<code>MOD (m, n)</code>	Întoarce restul împărțirii lui <i>m</i> la <i>n</i> .	$\text{MOD}(11, 4) = \text{MOD}(11, -4) = 3$ $\text{MOD}(-11, 4) = \text{MOD}(-11, -4) = -3$
<code>ABS (n)</code>	Întoarce valoarea absolută.	$\text{ABS}(-2) = 2$
<code>SQRT (n)</code>	Întoarce rădăcina pătrată.	$\text{SQRT}(2) = 1.41421356$
<code>EXP (n)</code>	Întoarce <i>e</i> la puterea <i>n</i>	$\text{EXP}(1) = 2.71828183$
<code>LN (n)</code>	Întoarce logaritm natural din <i>n</i>	$\text{LN}(2.71828183) = 1$
<code>LOG (n, m)</code>	Întoarce logaritm în bază <i>n</i> din <i>m</i>	$\text{LOG}(2, 4) = 2$
<code>POWER (n, p)</code>	Întoarce <i>n</i> la puterea <i>p</i>	$\text{POWER}(2, 5) = 32$
<code>SIGN (n)</code>	Întoarce semnul lui <i>n</i>	$\text{SIGN}(-2) = -1$ $\text{SIGN}(0) = 0$ $\text{SIGN}(2) = 1$
<code>COS, COSH</code>	Cosinus, cosinus hiperbolic	$\text{COS}(3.1415926) = -1$
<code>SIN, SINH</code>	Sinus, sinus hiperbolic	$\text{SIN}(3.1415926/2) = 0$

7. Afișați detalii despre salariații care au lucrat un număr întreg de săptămâni până la data curentă.

Indicație: `MOD(ROUND(SYSDATE - hire_date), 7)=0`

8. Analizați următoarele **operații pe expresii de tip dată calendaristică**:

Operație	Tipul rezultat	Descriere
<code>date -/+ number</code>	Date	Scade/Adaugă un număr de zile dintr-o / la o dată calendaristică.
<code>date1 - date2</code>	Number	Întoarce numărul de zile dintre două date calendaristice.
<code>date +/- number/24</code>	Date	Scade/Adaugă un număr de ore la o / dintr-o dată calendaristică.

9. Afișați data (luna, ziua, ora, minutul și secunda) de peste 10 zile.

10. Afișați numărul de zile rămase până la sfârșitul anului.

11. a. Afișați data de peste 12 ore.

Indicație: `SYSDATE+12/24`

b. Afișați data de peste 5 minute.

Indicație: `SYSDATE+1/288`

12. Analizați următoarele **funcții pentru prelucrarea datelor calendaristice**:

Funcție	Semnificație	Exemplu
<code>SYSDATE</code>	Întoarce data și timpul curent	
<code>MONTHS_BETWEEN (data1, data2)</code>	Întoarce numărul de luni dintre <i>data1</i> și data <i>data2</i> . Rezultatul poate fi pozitiv sau negativ după cum <i>data1</i> este mai recentă sau nu față de <i>data2</i> . Zecimalele reprezintă părți dintr-o luna!	<code>ROUND (MONTHS_BETWEEN (SYSDATE+31, SYSDATE)) = 1</code>
<code>ADD_MONTHS (data, n)</code>	Adaugă <i>n</i> luni la o data specificată. Valoarea <i>n</i> trebuie să fie întreagă (pozitivă sau negativă).	<code>MONTHS_BETWEEN (ADD_MONTHS (SYSDATE, 3), SYSDATE) = 3</code>
<code>NEXT_DAY (data, expr)</code>	Întoarce data corespunzătoare primei zile a săptămânii specificate (<i>expr</i>) care urmează după <i>data</i> .	<code>NEXT_DAY ('15-DEC-2006', 'Monday') = '18-DEC-2006'</code>

		<code>NEXT_DAY ('15-DEC-2006', 2) = '18-dec-2006'</code>
<code>LAST_DAY (data)</code>	Întoarce data corespunzătoare ultimei zile din luna calendaristică ce conține data specificată.	<code>LAST_DAY ('15-FEB-2006') = '28-FEB-2006'</code>
<code>ROUND (data [, format])</code>	Întoarce data calendaristică rotunjită după formatul specificat. Valoarea implicită este 'DAY'.	<code>ROUND (TO_DATE ('27-OCT-00'), 'YEAR') = 01-JAN-01</code> <code>TO_CHAR (ROUND (TO_DATE ('15-feb-2006 13:50', 'dd-mon-yyyy hh24:mi')), 'dd-mm-yyyy hh24:mi') = '16-02-2006 00:00'</code> <code>TO_CHAR (ROUND (TO_DATE ('15-feb-2006 11:50', 'dd-mon-yyyy hh24:mi')), 'dd-mm-yyyy hh24:mi') = '15-02-2006 00:00'</code>
<code>TRUNC (data [, format])</code>	Întoarce data calendaristică trunchiată după formatul specificat. Valoarea implicită este 'DAY'.	<code>TO_CHAR (TRUNC (TO_DATE ('15-feb-2006 13:50', 'dd-mon-yyyy hh24:mi')), 'dd-mm-yyyy hh24:mi') = '15-02-2006 00:00'</code>
<code>LEAST (d1, ..., dn)</code> <code>GREATEST (d1, ..., dn)</code>	Dintr-o listă de date calendaristice, funcțiile întorc prima, respectiv ultima dată în ordine cronologică.	

13. Să se afișeze numele angajatului, data angajării și data negocierii salariului, care a avut loc în prima zi de Luni, după 6 luni de serviciu. Etichetați această coloană “Negociere”.

Indicație: `NEXT_DAY (ADD_MONTHS (hire_date, 6), 'Monday')`

14. Pentru fiecare angajat să se afișeze numele și numărul de luni de la data angajării. Etichetați coloana “Luni lucrate”. Să se ordoneze rezultatul după numărul de luni lucrate. Se va rotunji numărul de luni la cel mai apropiat număr întreg.

Indicație: `ROUND (MONTHS_BETWEEN (SYSDATE, hire_date))`

15. Analizați următoarele funcții de conversie:

Observație: Conversiile implicite asigurate de *server-ul Oracle* sunt:

- de la VARCHAR2 sau CHAR la NUMBER;
- de la VARCHAR2 sau CHAR la DATE;
- de la NUMBER la VARCHAR2 sau CHAR;
- de la DATE la VARCHAR2 sau CHAR.

```
SELECT last_name
FROM employees
WHERE TO_CHAR(hire_date, 'yyyy')=1994;

SELECT last_name
FROM employees
WHERE hire_date='07-JUN-1994';

SELECT employee_id||' '||last_name||' '||hire_date
FROM employees
WHERE department_id=10;
```

Conversiile explicite se realizează cu ajutorul funcțiilor de tip TO_{tip}

Funcție	Semnificație	Exemplu
TO_CHAR(expr[, format][, nlsparameters])	Convertește o valoare de tip numeric sau dată calendaristică, la un șir de caractere conform cu formatul specificat sau cu setările naționale specificate (NLS - National Language Support). Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii implicați. Formatul este case sensitive.	TO_CHAR('3') = ' 3' TO_CHAR(-12) = '-12' TO_CHAR(sysdate, 'DDMMYYYY') = ' 09122004' TO_CHAR(sysdate + 365 * 57, 'ddmmyyyy') = '25112061'
TO_NUMBER(expr[, format][, nlsparameters])	Convertește o valoare de tip șir de caractere la o valoare numerică conform cu formatul specificat. Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii implicați.	TO_NUMBER('-12.22', 'S99.99') = -12.22
TO_DATE(expr[, format][, nlsparameters])	Convertește o valoare de tip șir de caractere la o valoare de tip dată calendaristică în conformitate cu	TO_DATE('15-feb-2006', 'dd- mon-yyyy') = '15-feb-2006'

	formatul specificat. Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii implicați.	
--	---	--

16. Afișați numele și prenumele pentru toți angajații care s-au angajat în luna mai.

17. Afișați data următoarei zile de Vineri de peste 3 luni. Etichetați coloana. Rezultatul trebuie să fie formatat 'NumeZi, NumeLuna NumarZiLuna, an'.

18. Analizați următoarele **funcții SQL**:

Funcție	Semnificație	Exemplu
<code>NVL(expr1, expr2)</code>	Întoarce <i>expr1</i> dacă aceasta nu este NULL, <i>expr2</i> în caz contrar. Cele 2 expresii trebuie să aibă același tip sau <i>expr2</i> să permită conversia implicită la tipul expresiei <i>expr1</i> .	<code>NVL(NULL, 1) = 1</code> <code>NVL(2, 1) = 2</code> <code>NVL('c', 1) = 'c'</code> face conversie <code>NVL(1, 'c') - eroare</code> nu face conversie
<code>NVL2(expr1, expr2, expr3)</code>	Dacă <i>expr1</i> este nenulă atunci întoarce <i>expr2</i> , altfel întoarce <i>expr3</i> .	<code>NVL2(1, 2, 3) = 2</code> <code>NVL2(NULL, 2, 3) = 3</code>
<code>NULLIF(expr1, expr2)</code>	Dacă <i>expr1</i> = <i>expr2</i> , atunci funcția returnează NULL, altfel întoarce expresia <i>expr1</i> . Echivalent cu <code>CASE WHEN expr1 = expr2 THEN NULL ELSE expr1 END</code>	<code>NULLIF(1, 2) = 1</code> <code>NULLIF(1, 1) = NULL</code>
<code>COALESCE(expr1, expr2, ..., exprn)</code>	Întoarce <i>expr1</i> dacă nu este NULL, altfel <i>expr2</i> , dacă <i>expr2</i> nu este NULL,	<code>COALESCE(1, 2, 3) = 1</code> <code>COALESCE(NULL, 2, 3) = 2</code> <code>COALESCE(NULL, NULL, 3) = 3</code>
<code>UID, USER</code>	Întorc ID-ul, respectiv username-ul utilizatorului ORACLE curent	<code>SELECT UID, USER</code> <code>FROM DUAL;</code>
<code>VSIZE(expr)</code>	Întoarce numărul de octeți ai unei expresii de tip DATE, NUMBER sau VARCHAR2.	<code>SELECT VSIZE(hire_date)</code> <code>FROM employees</code> <code>WHERE employee_id=104;</code>

19. Afișați numele angajaților și comisionul. Dacă un angajat nu câștigă comision, afișați “Fara comision”. Etichetați coloana “Comision”.

20. Afișați numele, salariul și comisionul tuturor angajaților al căror venit lunar depășește 10000\$.

21. Analizați expresia CASE și funcția DECODE:

Funcție/Expresie	Semnificație	Exemplu
<pre> CASE expr WHEN expr_bool1 THEN return_expr1 [WHEN expr_bool2 THEN return_expr2 ... WHEN expr_booln THEN return_exprn] [ELSE return_expr] END </pre>	<p>În funcție de valoarea unei expresii întoarce valoarea primei perechi WHEN .. THEN care se potrivește sau dacă nu se potrivește nici una expresia din ELSE. Nu se poate specifica NULL pentru toate expresiile de returnat. (<i>return_expr1</i>). Toate expresiile trebuie să aibă același tip de date</p>	
<pre> DECODE (expr, expr_cautare1, expr_rezultat1, [expr_cautare2, expr_rezultat2, .. expr_cautaren, expr_rezultatn,] [rezultat_implicit]) </pre>	<p>Decodifică valoarea expresiei. Dacă valoarea expresiei este <i>expr_cautare1</i> atunci e returnată <i>expr_rezultat1</i>. Dacă nu se potrivește nici o expresie de căutare atunci e returnat <i>rezultat_implicit</i>.</p>	<pre> DECODE (1, 1, 2, 3) = 2 DECODE (2, 1, 2, 3) = 3 DECODE (3, 1, 2, 3) = 3 </pre>

22. Afișați numele, codul funcției, salariul și o coloană care să arate salariul după mărire. Se știe că pentru IT_PROG are loc o mărire de 10%, pentru ST_CLERK 15%, iar pentru SA_REP o mărire de 20%. Pentru ceilalți angajați nu se acordă mărire. Denumiți coloana "Salariu revizuit".

```

SELECT last_name, job_id, salary,
       DECODE(job_id, 'IT_PROG', salary*1.1,
               'ST_CLERK', salary*1.15,
               'SA_REP', salary*1.2,
               salary ) "salariu revizuit"
FROM employees;

SELECT last_name, job_id, salary,
       CASE job_id WHEN 'IT_PROG' THEN salary* 1.1
                  WHEN 'ST_CLERK' THEN salary*1.15
                  WHEN 'SA_REP' THEN salary*1.2
                  ELSE salary
       END "salariu revizuit"
FROM employees;

```

```
SELECT last_name, job_id, salary,  
       CASE WHEN job_id= 'IT_PROG' THEN salary* 1.1  
            WHEN job_id='ST_CLERK' THEN salary*1.15  
            WHEN job_id ='SA_REP' THEN salary*1.2  
            ELSE salary  
       END "salariu revizuit"  
FROM employees;
```

- 23.** Afișați numele salariatului și codul departamentului în care acesta lucrează. Dacă există salariați care nu au un cod de departament asociat, atunci pe coloana *id_department* afișați:
- textul “fara departament”;
 - valoarea zero.
- 24.** Afișați valoarea medie a comisionului lunar pentru:
- salariații care câștigă comision;
 - toți salariații.
- 25. a)** Afișați numele angajaților care nu au manager.
b) Afișați numele angajaților și codul managerilor lor. Pentru angajații care nu au manager să apară textul “nu are sef”.
- 26.** Afișați numele salariatului și:
- a)** venitul anual dacă are comision;
 - b)** salariul dacă nu are comision.
- Se va utiliza funcția NVL2.
- 27.** Pentru toți angajații, afișați valoarea *null* dacă lungimea numelui este egală cu cea a prenumelui, iar în caz contrar doar lungimea numelui. Se va utiliza funcția NULLIF.
- 28.** Modificați cererea anterioară astfel încât să afișați în locul valorii *null* textul “Valori egale”.
- 29.** Afișați numele salariatului, salariul și salariul revizuit astfel:
- dacă lucrează de mai mult de 200 de luni atunci salariul va fi mărit cu 20%;
 - dacă lucrează de mai mult de 150 de luni, dar mai puțin de 200 de luni, atunci salariul va fi mărit cu 15%;
 - dacă lucrează de mai mult de 100 de luni, dar mai puțin de 150 de luni, atunci salariul va fi mărit cu 10%;
 - altfel, salariul va fi mărit cu 5%.