

Baze de date-Anul 2

Laborator 2 SQL

Funcții SQL

I. [Funcții SQL]

Funcțiile SQL sunt predefinite în sistemul *Oracle* și pot fi utilizate în instrucțiuni SQL. Ele 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 încearcă să **convertească implicit** argumentul înainte să evalueze funcția.

Dacă o funcție SQL este apelată cu un argument *null*, ea returnează automat valoarea *null*. Singurele funcții elementare 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/grup)

1. **Funcțiile single row** returnează câte o singură linie rezultat pentru fiecare linie a tabelului sau vizualizării interogate. Aceste funcții pot apărea printre coloanele din clauza *SELECT*, clauzele *WHERE*, *START WITH*, *CONNECT BY* și *HAVING*. În ceea ce privește tipul argumentelor asupra cărora operează și al rezultatelor furnizate, funcțiile *single row* pot fi clasificate în clase corespunzătoare.

Obs: Testarea funcțiilor prezentate se face de maniera : ***SELECT apel_funcție FROM dual***; astfel că vom omite comanda *SELECT* și vom da numai apelul funcției și rezultatul returnat.

❑ **Funcțiile de conversie** cele mai importante sunt:

Funcție	Descriere	Exemplu conversie
<i>TO_CHAR</i>	convertește (sau formatează) un număr sau o dată calendaristică în șir de caractere	<i>TO_CHAR(7) = '7'</i> <i>TO_CHAR(-7) = '-7'</i> <i>TO_CHAR(SYSDATE, 'DD/MM/YYYY') = '18/04/2007'</i>
<i>TO_DATE</i>	convertește (sau formatează) un număr sau un șir de caractere în dată calendaristică	<i>TO_DATE('18-APR-2007','dd-mon-yyyy')</i>
<i>TO_NUMBER</i>	convertește (sau formatează) un șir de caractere în număr	<i>TO_NUMBER('-25789', 'S99,999') = -25,789</i>

Obs: Există două tipuri de conversii:

- **implicit**, realizate de sistem atunci când este necesar;
- **explicit**, indicate de utilizator prin intermediul funcțiilor de conversie.

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*.

Datele calendaristice pot fi formatate cu ajutorul funcției **TO_CHAR(data, format)**, unde formatul poate fi alcătuit dintr-o combinație a următoarelor elemente:

<i>Element</i>	<i>Semnificație</i>
D	Numărul zilei din săptămâna (duminica=1; luni=2; ...sâmbătă=6)
DD	Numărul zilei din lună.
DDD	Numărul zilei din an.
DY	Numele zilei din săptămână, printr-o abreviere de 3 litere (MON, THU etc.)
DAY	Numele zilei din săptămână, scris în întregime.
MM	Numărul lunii din an.
MON	Numele lunii din an, printr-o abreviere de 3 litere (JAN, FEB etc.)
MONTH	Numele lunii din an, scris în întregime.
Y	Ultima cifră din an
YY, YYY, YYYY	Ultimele 2, 3, respectiv 4 cifre din an.
YEAR	Anul, scris în litere (ex: <i>two thousand four</i>).
HH12, HH24	Orele din zi, între 0-12, respectiv 0-24.
MI	Minutele din oră.
SS	Secundele din minut.
SSSSS	Secundele trecute de la miezul nopții.

❑ **Funcțiile pentru prelucrarea caracterelor** sunt prezentate în următorul tabel:

Funcție	Descriere	Exemplu
<i>LENGTH(string)</i>	întoarce lungimea șirului de caractere <i>string</i>	<i>LENGTH('Informatica')=11</i>
<i>SUBSTR(string, start [,n])</i>	întoarce subșirul lui <i>string</i> care începe pe poziția <i>start</i> și are lungimea <i>n</i> ; dacă <i>n</i> nu este specificat, subșirul se termină la sfârșitul lui <i>string</i> ;	<i>SUBSTR('Informatica', 1, 4) = 'Info'</i> <i>SUBSTR('Informatica', 6) = 'matica'</i> <i>SUBSTR('Informatica', -5) = 'atica'</i> (ultimele 5 caractere)
<i>LTRIM(string [, 'chars'])</i>	șterge din stânga șirului <i>string</i> orice caracter care apare în <i>chars</i> , până la găsirea primului caracter care nu este în <i>chars</i> ; în cazul în care <i>chars</i> nu este specificat, se șterg spațiile libere din stânga lui <i>string</i> ;	<i>LTRIM(' info') = 'info'</i>
<i>RTRIM(string [, 'chars'])</i>	este similar funcției <i>LTRIM</i> , cu excepția faptului că ștergerea se face la dreapta șirului de caractere;	<i>RTRIM('infoXXXX', 'X') = 'info'</i>
<i>TRIM (LEADING TRAILING BOTH chars FROM expresie)</i>	elimină caracterele specificate (<i>chars</i>) de la începutul (<i>leading</i>) , sfârșitul (<i>trailing</i>) sau din ambele părți, dintr-o expresie caracter dată.	<i>TRIM (LEADING 'X' FROM 'XXXInfoXXX') = 'InfoXXX'</i> <i>TRIM (TRAILING 'X' FROM 'XXXInfoXXX') = 'XXXInfo'</i> <i>TRIM (BOTH 'X' FROM 'XXXInfoXXX') = 'Info'</i> <i>TRIM (BOTH FROM ' Info ') = 'Info'</i>
<i>LPAD(string, length [, 'chars'])</i>	adaugă <i>chars</i> la stânga șirului de caractere <i>string</i> până când lungimea noului șir devine <i>length</i> ; în cazul în care <i>chars</i> nu este specificat, atunci se adaugă spații libere la stânga lui <i>string</i> ;	<i>LPAD (LOWER('info'),6) = ' info'</i>
<i>RPAD(string, length [, 'chars'])</i>	este similar funcției <i>LPAD</i> , dar adăugarea de caractere se face la dreapta șirului;	<i>RPAD (LOWER('info'), 6, 'X') = 'infoXX'</i>
<i>REPLACE(string1, string2 [,string3])</i>	întoarce <i>string1</i> cu toate aparițiile lui <i>string2</i> înlocuite prin <i>string3</i> ; dacă <i>string3</i> nu este specificat, atunci toate aparițiile lui <i>string2</i> sunt șterse;	<i>REPLACE ('\$b\$bb','\$','a') = 'ababb'</i> <i>REPLACE ('\$b\$bb','\$b','ad') = 'adadb'</i> <i>REPLACE ('\$a\$aa','\$') = 'aaa'</i>
<i>UPPER(string), LOWER(string)</i>	transformă toate literele șirului de caractere <i>string</i> în majuscule, respectiv minuscule;	<i>LOWER ('InFo') = 'info'</i> <i>UPPER ('info') = 'INFO'</i>
<i>INITCAP(string)</i>	transformă primul caracter al șirului în majusculă, restul caracterelor fiind transformate în minuscule	<i>INITCAP ('info') = 'Info'</i>

<i>INSTR(string, 'chars' [,start [,n]])</i>	caută în <i>string</i> , începând de la poziția <i>start</i> , a <i>n</i> -a apariție a secvenței <i>chars</i> și întoarce poziția respectivă; dacă <i>start</i> nu este specificat, căutarea se face de la începutul șirului; dacă <i>n</i> nu este specificat, se caută prima apariție a secvenței <i>chars</i> ;	<i>INSTR (LOWER('AbC aBcDe'), 'ab', 5, 2)</i> = 0 <i>INSTR (LOWER('AbCdE aBcDe'), 'ab', 5)</i> = 7
<i>ASCII(char)</i>	furnizează codul <i>ASCII</i> al primului caracter al unui șir	<i>ASCII ('alfa') = ASCII ('a') = 97</i>
<i>CHR(num)</i>	întoarce caracterul corespunzător codului <i>ASCII</i> specificat	<i>CHR(97)= 'a'</i>
<i>CONCAT(string1, string2)</i>	realizează concatenarea a două șiruri de caractere	<i>CONCAT ('In', 'fo') = 'Info'</i>
<i>TRANSLATE(string, source, destination)</i>	fiecare caracter care apare în șirurile de caractere <i>string</i> și <i>source</i> este transformat în caracterul corespunzător (aflat pe aceeași poziție ca și în <i>source</i>) din șirul de caractere <i>destination</i>	<i>TRANSLATE('\$a\$aa','\$','b') = 'babaa'</i> <i>TRANSLATE('\$a\$aaa','\$a','bc') = 'bcbccc'</i>

❑ **Funcțiile aritmetice** *single-row* pot opera asupra:

- unei singure valori, și aceste funcții sunt: *ABS* (valoarea absolută), *CEIL* (partea întreagă superioară), *FLOOR* (partea întreagă inferioară), *ROUND* (rotunjire cu un număr specificat de zecimale), *TRUNC* (trunchiere cu un număr specificat de zecimale), *EXP* (ridicarea la putere a lui *e*), *LN* (logaritm natural), *LOG* (logaritm într-o bază specificată), *MOD* (restul împărțirii a două numere specificate), *POWER* (ridicarea la putere), *SIGN* (semnul unui număr), *COS* (cosinus), *COSH* (cosinus hiperbolic), *SIN* (sinus), *SINH* (sinus hiperbolic), *SQRT* (rădăcina pătrată), *TAN* (tangent), *TANH* (tangent hiperbolic);
- unei liste de valori, iar acestea sunt funcțiile *LEAST* și *GREATEST*, care întorc cea mai mică, respectiv cea mai mare valoare a unei liste de expresii.

❑ **Funcțiile pentru prelucrarea datelor calendaristice** sunt:

Funcție	Descriere	Exemplu
<i>SYSDATE</i>	întoarce data și timpul curent; Pentru completarea sintaxei obligatorii a comenzii <i>SELECT</i> , se utilizează tabelul <i>DUAL</i> :	<i>SELECT SYSDATE FROM dual;</i>
<i>ADD_MONTHS(expr_date, nr_luni)</i>	întoarce data care este după <i>nr_luni</i> luni de la data <i>expr_date</i> ;	<i>ADD_MONTHS('02-APR-2007', 3) = '02-JUL-2007'.</i>
<i>NEXT_DAY(expr_date, day)</i>	întoarce următoarea dată după data <i>expr_date</i> , a cărei zi a săptămânii este cea specificată prin șirul de caractere <i>day</i>	<i>NEXT_DAY('18-APR-2007', 'Monday') = '23-APR-2007'</i>
<i>LAST_DAY(expr_date)</i>	întoarce data corespunzătoare ultimei zile a lunii din care data <i>expr_date</i> face parte	<i>LAST_DAY('02-DEC-2007') = '31-DEC-2007'</i>

<i>MONTHS_BETWEEN</i> (<i>expr_date2</i> , <i>expr_date1</i>)	Întoarce numărul de luni dintre cele două date calendaristice specificate. Data cea mai recentă trebuie specificată în primul argument, altfel rezultatul este negativ.	<i>MONTHS_BETWEEN</i> ('02-DEC-2005', '10-OCT-2002') = 37.7419355 <i>MONTHS_BETWEEN</i> ('10-OCT-2002', '02-DEC-2005') = -37.7419355
<i>TRUNC</i> (<i>expr_date</i>)	Întoarce data <i>expr_date</i> , dar cu timpul setat la ora 12:00 AM (miezul nopții)	<i>TO_CHAR</i> (<i>TRUNC</i> (<i>SYSDATE</i>), 'dd/mm/yy HH24:MI') = '02/12/05 00:00'
<i>ROUND</i> (<i>expr_date</i>)	dacă data <i>expr_date</i> este înainte de miezul zilei, întoarce data <i>d</i> cu timpul setat la ora 12:00 AM; altfel, este returnată data corespunzătoare zilei următoare, cu timpul setat la ora 12:00 AM	<i>TO_CHAR</i> (<i>ROUND</i> (<i>SYSDATE</i>), 'dd/mm/yy hh24:mi am') = '03/12/05 00:00 AM'
<i>LEAST</i> (<i>d1</i> , <i>d2</i> , ..., <i>dn</i>), <i>GREATEST</i> (<i>d1</i> , <i>d2</i> , ..., <i>dn</i>)	dintr-o listă de date calendaristice, funcțiile întorc prima, respectiv ultima dată în ordine cronologică	<i>LEAST</i> (<i>SYSDATE</i> , <i>SYSDATE</i> + 3, <i>SYSDATE</i> - 5) = <i>SYSDATE</i> - 5 <i>GREATEST</i> (<i>SYSDATE</i> , <i>SYSDATE</i> + 3, <i>SYSDATE</i> - 5) = <i>SYSDATE</i> + 3

Operațiile care se pot efectua asupra datelor calendaristice sunt următoarele:

Operație	Tipul de date al rezultatului	Descriere
<i>expr_date</i> +/- <i>expr_number</i>	<i>Date</i>	Scade/adună un număr de zile dintr-o / la o dată. Numărul de zile poate să nu fie întreg (putem adăuga, de exemplu, un număr de minute sau de ore).
<i>expr_date1</i> - <i>expr_date2</i>	<i>Number</i>	Întoarce numărul de zile dintre două date calendaristice. Data <i>expr_date1</i> trebuie să fie mai recentă decât <i>expr_date2</i> , altfel rezultatul este negativ.

❑ **Funcții diverse:**

Funcție	Descriere	Exemplu
<i>DECODE</i> (<i>value</i> , <i>if1</i> , <i>then1</i> , <i>if2</i> , <i>then2</i> , ..., <i>ifN</i> , <i>thenN</i> , <i>else</i>)	returnează <i>then1</i> dacă <i>value</i> este egală cu <i>if1</i> , <i>then2</i> dacă <i>value</i> este egală cu <i>if2</i> etc.; dacă <i>value</i> nu este egală cu nici una din valorile <i>if</i> , atunci funcția întoarce valoarea <i>else</i> ;	<i>DECODE</i> ('a', 'a', 'b', 'c') = 'b' <i>DECODE</i> ('b', 'a', 'b', 'c') = 'c' <i>DECODE</i> ('c', 'a', 'b', 'c') = 'c'

<i>NVL(expr_1, expr_2)</i>	dacă <i>expr_1</i> este <i>NULL</i> , întoarce <i>expr_2</i> ; altfel, întoarce <i>expr_1</i> . Tipurile celor două expresii trebuie să fie compatibile sau <i>expr_2</i> să poată fi convertit implicit la <i>expr_1</i>	<i>NVL(NULL, 1) = 1</i> <i>NVL(2, 1) = 2</i> <i>NVL('a', 1) = 'a' -- conversie implicită</i> <i>NVL(1, 'a') -- eroare --nu are loc conversia implicită</i>
<i>NVL2(expr_1, expr_2, expr_3)</i>	dacă <i>expr_1</i> este <i>NOT NULL</i> , întoarce <i>expr_2</i> , altfel întoarce <i>expr_3</i>	<i>NVL2(1, 2, 3) = 2</i> <i>NVL2(NULL, 1, 2) = 2</i>
<i>NULLIF(expr_1, expr_2)</i>	Dacă <i>expr_1 = expr_2</i> atunci funcția returnează <i>NULL</i> , altfel returnează expresia <i>expr_1</i> . Echivalent cu <i>CASE WHEN expr1 = expr2 THEN NULL ELSE expr1 END</i>	<i>NULLIF(1, 2) = 1</i> <i>NULLIF(1, 1) = NULL</i>
<i>COALESCE(expr_1, expr_2, ..., expr_n)</i>	Returnează prima expresie <i>NOT NULL</i> din lista de argumente.	<i>COALESCE(NULL, NULL, 1, 2, NULL) = 1</i>
<i>UID, USER</i>	întorc <i>ID</i> -ul, respectiv <i>username</i> -ul utilizatorului <i>ORACLE</i> curent	<i>SELECT USER</i> <i>FROM dual;</i>
<i>VSIZE(expr)</i>	întoarce numărul de octeți ai unei expresii de tip <i>DATE</i> , <i>NUMBER</i> sau <i>VARCHAR2</i>	<i>SELECT VSIZE(salary)</i> <i>FROM employees</i> <i>WHERE employee_id=200;</i>

Utilizarea funcției *DECODE* este echivalentă cu utilizarea clauzei *CASE* (într-o comandă SQL). O formă a acestei clauze este:

<pre> CASE WHEN cond_1 THEN valoare_1 [WHEN cond_2 THEN valoare_2 ... WHEN cond_n THEN valoare_n] [ELSE valoare] END </pre>	<p>În funcție de valoarea <i>cond_i</i> returnează <i>valoare_i</i> corespunzătoare primei clauze <i>WHEN .. THEN</i> pentru care <i>cond_i</i> este adevărată; dacă nu corespunde cu nici o clauză <i>WHEN</i> atunci returnează valoarea din <i>ELSE</i>. Nu se poate specifica <i>NULL</i> pentru toate valorile de returnat. Toate valorile trebuie să aibă același tip de date.</p>
------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2. Funcțiile multiple-row (agregat) pot fi utilizate pentru a returna informația corespunzătoare fiecăruia dintre grupurile obținute în urma divizării liniilor tabelului cu ajutorul clauzei *GROUP BY*. Ele vor fi tratate într-un laborator dedicat.

II. [Exerciții]

[Funcții pe siruri de caractere]

1. Scrieți o cerere care are următorul rezultat pentru fiecare angajat:

```

<prenume angajat> <nume angajat> castiga <salariu> lunar dar doreste <salariu de 3 ori mai mare>.

```

Etichetați coloana "Salariu ideal". Pentru concatenare, utilizați atât funcția *CONCAT* cât și operatorul "||".
2. Scrieți o cerere prin care să se afișeze prenumele salariaților cu prima literă majusculă și toate celelalte litere minuscule, numele acestuia cu majuscule și lungimea numelui, pentru

angajații al căror nume începe cu J sau M sau care au a treia literă din nume A. Rezultatul va fi ordonat descrescător după lungimea numelui. Se vor eticheta coloanele corespunzător. Se cer 2 soluții (cu operatorul *LIKE* și funcția *SUBSTR*).

3. Să se afișeze pentru angajații cu prenumele „Steven”, codul, numele și codul departamentului în care lucrează. Căutarea trebuie să nu fie *case-sensitive*, iar eventualele *blank*-uri care preced sau urmează numelui trebuie ignorate.
4. Să se afișeze pentru toți angajații al căror nume se termină cu litera 'e', codul, numele, lungimea numelui și poziția din nume în care apare prima data litera 'a'. Utilizați *alias*-uri corespunzătoare pentru coloane.

[Funcții aritmetice]

5. Să se afișeze detalii despre salariații care au lucrat un număr întreg de săptămâni până la data curentă. Este necesară rotunjirea diferenței celor două date calendaristice?
6. Să se afișeze codul salariatului, numele, salariul, salariul mărit cu 15%, exprimat cu două zecimale și numărul de sute al salariului nou rotunjit la 2 zecimale. Etichetați ultimele două coloane “Salariu nou”, respectiv “Numar sute”. Se vor lua în considerare salariații al căror salariu nu este divizibil cu 1000.
7. Să se listeze numele, salariul și o coloana care să reprezinte nivelul venitului (pentru fiecare 1000 să fie folosit câte un simbol \$). **Ex:** 6750 -> ‘\$\$\$\$\$’

[Funcții și operații cu date calendaristice]

8. Să se afișeze data (numele lunii, ziua, anul, ora, minutul și secunda) de peste 30 zile.
9. Să se afișeze numărul de zile rămase până la sfârșitul anului.
10. a) Să se afișeze data de peste 12 ore.
b) Să se afișeze data de peste 5 minute
11. Să se afișeze numele și prenumele angajatului (într-o singură coloană), data angajării și data negocierii salariului, care este prima zi de Luni după 6 luni de serviciu. Etichetați această coloană “Negociere”.
12. 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.
13. Să se afișeze numele, data angajării și ziua săptămânii în care a început lucrul fiecare salariat. Etichetați coloana “Zi”. Ordonați rezultatul după ziua săptămânii, începând cu Luni.

14. Sa se afiseze numele și data angajării pentru fiecare salariat care a fost angajat in 1987. Se cer 2 soluții: una în care se lucrează cu formatul implicit al datei și alta prin care se formatează data.

Obs: Elementele (câmpuri ale valorilor de tip datetime) care pot fi utilizate în cadrul acestei funcției EXTRACT sunt: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND.

[Funcții diverse]

15. Să se afișeze numele angajaților și comisionul. Dacă un angajat nu câștigă comision, să se scrie "Fara comision". Etichetați coloana "Comision".

16. Să se listeze numele, salariul și comisionul tuturor angajaților al căror venit lunar (salariu + valoare comision) depășește 10000.

17. Să se afișeze numele, codul job-ului, salariul și o coloană care să arate salariul după mărire. Se presupune că pentru IT_PROG are loc o mărire de 20%, pentru SA_REP creșterea este de 25%, iar pentru SA_MAN are loc o mărire de 35%. Pentru ceilalți angajați nu se acorda marire.