

# Capitolul 5

## Alte functii singulare aplicate liniilor unei baze de date

In acest capitol vom trata functiile (de tip) referitoare la date calendaristice 'DATE', functiile de conversie si functii care accepta orice tip de data de intrare.

### Funcțiile de tip data calendaristica

Aceste functii se aplica asupra datelor ORACLE.

Toate functiile de tip data calendaristica intorc valoarea tipului DATE cu exceptia lui MONTHS\_BETWEEN care intoarce o valoare numerica.

### Stocarea datelor calendaristice ORACLE

ORACLE stocheaza datele calendaristice intr-un format numeric intern reprezentind:

- Secolul
- Anul
- Luna
- Ziua
- Ora
- Minutele
- Secundele

Formatul implicit de afisare sau intrare pentru o data calendaristica e DD-MON-YY. Datele calendaristice ORACLE pot varia intre 1 ian 4712 ien si 31 dec 4712 e.n

### Sysdate

Sysdate intoarce data curenta si timpul. Puteti folosi SYSDATE la fel ca folosirea oricarui nume de coloana. De exemplu , puteti afisa data curenta selectind SYSDATE dintr-o tabela. Este obisnuit sa se selecteze SYSDATE dintr-o tabela 'dummy' numita DUAL.

Tabela DUAL apartine utilizatorului 'SYS' si poate fi accesata de toti utilizatorii. Contine o singura coloana ,DUMMY si o linie cu valoarea 'x'.

Tabela DUAL e folositoare cind doriti sa obtineti o singura valoare-de exemplu , valoarea unei constante , pseudo-coloane sau expresii care nu e derivata dintr-o tabela cu data 'user'.

Pentru a afisa date curenta:

```
SELECT SYSDATE  
FROM SYS.DUAL;
```

Puteti selecta usor SYSDATE din EMP, dar 14 linii cu aceeasi SYSDATE vor fi intoarse , una pentru fiecare linie din tabela EMP.  
DUAL e preferata pentru ca e suficienta o singura linie intoarsa.

## Folosirea operatorilor aritmetici

Pornind de la faptul ca data calendaristica e memorata ca un numar , e posibil sa se faca calcule cu date calendaristice, folosind operatori aritmetici ca + sau -.Puteti aduna sau scadea constante numerice ca si date calendaristice.

Operatiile pe care le puteti face sunt:

data + numar

aduna un numar de zile la data, reintorcind o data calendaristica

data - numar

scade un numar de zile dintr-o data, producind o data calendaristica

date - date

scade o data dintr-o data , obtinind un nr. de zile.

date + numar/24

aduna un nr. de ore pentru a obtine o data calendaristica.

```
SELECT  HIREDATE,HIREDATE+7, HIREDATE-7,SYSDATE - HIREDATE
FROM    EMP
WHERE   HIREDATE LIKE '%JUN%';
```

HIREDATE	HIREDATE+7	HIREDATE-7	SYSDATE-HIREDATE
13-jun-83	20-jun-83	06-jun-83	1982.70628
11-jun-84	18-jun-84	04-jun-84	1618.70628
04-jun-84	11-jun-84	28-may-84	1625.70628
04-jun-84	11-jun-84	28-may-84	1625.70628

Scazind din SYSDATE ,HIREDATE din tabela EMP intoarce nr.de zile de la angajarea fiecarui om.

MONTHS\_BETWEEN(data1,data2)

gaseste nr. de luni intre data1 si data2. Rezultatul poate fi pozitiv sau negativ.

Daca data1 e mai tarzie decat data2 , REZULTATUL E POZITIV, daca data2 este mai tirzie decit data1 , REZULTATUL E NEGATIV.

```
SELECT  MONTHS_BETWEEN (SYSDATE,HIREDATE) ,
        MONTHS_BETWEEN('01-jan-84','05-nov-88')
FROM    EMP
WHERE   MONTHS_BETWEEN(SYSDATE,HIREDATE) >59;
```

```
MONTHS_BETWEEN(SYSDATE,HIREDATE) MONTHS_BETWEEN('01-jan-84','05-
nov-88')
-----
-----
65.0873622          -
58.129032          -
63
58.129032
```

```

60.5067171 -
58.129032
59.3454267 -
58.129032
59.3454267 -
58.129032
59.8292977 -
58.129032
6 records selected

```

Partea neintreaga a rezultatului reprezinta o portiune dintr-o luna.

### ADD\_MONTHS(data,n)

aduna n numar de luni calendaristice la 'data'. n trebuie sa fie intreg si poate fi si negativ.

```

SELECT HIREDATE, ADD_MONTHS (HIREDATE, 3), ADD_MONTHS (HIREDATE, -3)
FROM EMP
WHERE DEPTNO =20;

```

HIREDATE	ADD_MONTHS (HIREDATE, 3)	ADD_MONTHS (HIREDATE, -3)
13-jun-83	13-sep-83	13-mar-83
31-oct-83	31-jan-84	31-jul-83
05-mar-84	05-jun-84	05-dec-83
04-jun-84	04-sep-84	04-mar-84
05-dec-83	05-mar-84	05-sep-83

### NEXT\_DAY(data1,,char)

data urmatoarei zile a saptaminii(char) urmind data1. Char trebuie sa fie un numar reprezentind o zi sau un caracter.

```

SELECT HIREDATE, NEXT_DAY (HIREDATE, 'FRIDAY'), NEXT_DAY (HIREDATE, 6)
FROM EMP
WHERE DEPTNO =10;

```

HIREDATE	NEXT_DAY (HIREDATE, 'FRIDAY')	NEXT_DAY (HIREDATE, 6)
14-may-84	18-may-84	18-may-84
09-jul-84	13-jul-84	13-jul-84
21-nov-83	25-nov-83	25-nov-83

### LAST\_DAY(data1)

gaseste data reprezentind ultima zi a lunii care contine data 1.

```

SELECT SYSDATE, LAST_DAY (SYSDATE), HIREDATE, LAST_DAY (HIREDATE),
LAST_DAY ('15-FEB-88')
FROM EMP
WHERE DEPTNO =20;

```

SYSDATE	LAST_DAY (SYSDATE)	HIREDATE	LAST_DAY (HIREDATE)	LAST_DAY ('15-feb-88')
04-DEC-89	31-DEC-89	17-DEC-80	31-DEC-80	29-FEB-88
04-dec-89	31-dec-89	02-apr-81	30-apr-81	29-feb-88
04-dec-89	31-dec-89	09-dec-82	31-dec-82	29-feb-88

04-dec-89	31-dec-89	12-jan-83	31-jan-83	29-feb-88
04-dec-89	31-dec-89	03-dec-81	31-dec-81	29-feb-88

Funcția ROUND poate fi aplicată pe date calendaristice.

**Round(data1)**

întoarce data1 cu timpul setat la 12:00AM(noaptea). Aceasta e folositor când comparăm date care au timpuri diferite.

**ROUND (data1,'MONTH')**

întoarce prima zi a lunii conținând data1 dacă data1 este în prima parte a lunii, altfel întoarce prima zi a următoarei luni.

**ROUND(data 1,' YEAR')**

întoarce prima zi a anului conținând data1, dacă data1 este în prima jumătate a anului, altfel întoarce prima zi a următorului an.

```
SELECT SYSDATE,ROUND (SYSDATE, 'MONTH' ),ROUND (SYSDATE, 'YEAR' )
FROM SYS.DUAL;
```

SYSDATE	ROUND (SYSDATE, 'MONTH' )	ROUND (SYSDATE, 'YEAR' )
04-dec-89	01-dec-89	01-jan-90

**TRUNC(data1,'char')**

găsește prima zi a lunii care e conținută în data1, când char='MONTH'. Dacă char= 'YEAR', găsește prima zi a anului care conține data1.

```
SELECT SYSDATE,TRUNC (SYSDATE, 'MONTH' ),TRUNC (SYSDATE, 'YEAR' )
FROM SYS.DUAL;
```

SYSDATE	TRUNC (SYSDATE, 'MONTH' )	TRUNC (SYSDATE, 'YEAR' )
04-dec-89	01-dec-89	01-jan-89

TRUNC e folositor când vreți să ștergeți timpul dintr-o dată. Timpul component al zilei este șters implicit.

## Funcții de conversie

SQL furnizează un număr de funcții care controlează conversiile datelor calendaristice. Aceste funcții de conversie, convertește o valoare de la un tip de dată la alt tip.

**TO\_CHAR(numar/data,{'fmt'})**

convertește număr sau dată la caractere în format 'fmt'.

**TO\_NUMBER(char)**

convertește 'char' care conține un număr în 'NUMBER'.

**TO\_DATE('char','fmt')**

convertește valoarea 'char' reprezentând o dată calendaristică, într-o valoare dată în felul 'fmt' specificat. Dacă fmt e omis, formatul e DD-MON-YY.

## TO\_CHAR

Functia TO\_CHAR este folosita frecvent pentru a schimba un format de data de la cel implicat la un format de afisare alternativ. TO\_CHAR (data,'date picture') specifica ca data va fi convertita la un nou format de iesire.

Pentru a converti date curente de la formatul implicit (DD-MON-YY) la un nou format 'date picture':

```
SELECT TO_CHAR(SYSDATE, 'DAY, DDTH MONTH YYYY')
FROM SYS.DUAL;
```

```
TO_CHAR (SYSDATE, 'DAY, DDTHMONTHYYYY')
-----
TUESDAY      , 05TH SEPTEMBER 1989
```

Observati ca:

- 'Date picture', care trebuie incadrate de apostrofi poate contine orice format dintre cele tratate mai jos. Coloana si 'date picture' trebuie separate de virgula.
- DAY si MONTH la iesire sunt automat aduse la lungimea 9 adaugind blancuri.
- Pentru a sterge blancurile adaugate folositi prefixul FM(FILL MODE).

```
SELECT TO_CHAR (SYSDATE, 'fmDAY, ddth Month YYYY')
FROM SYS.DUAL;
```

```
TO_CHAR (SYSDATE, 'FMDAY, DDTHMONTHYYYY')
-----
Tuesday , 5th September 1989
```

FM poate fi folosit pentru a sterge zerourile de la inceput din formatul ddth ex:05TH este schimbat in 5th. Cazul in care 'date picture' este introdusa este cazul in care va fi afisata.

TO\_CHAR poate fi deasemenea folosita pentru extragerea timpului din zi , si afisarea lui intr-un format specificat.

Pentru afisarea timpului dintr-o zi:

```
SELECT TO_CHAR (SYSDATE, 'HH:MI:SS')
FROM SYS.DUAL;
```

```
TO_CHAR (SYSDATE, 'HH:MI:SS')
-----
08:16:24
```

Functia TO\_CHAR este de asemenea folosita pentru conversia unei valori de tip data numerica la o valoare de tip data caracter.

```
TO_CHAR (numar, 'number picture')
```

```
SELECT TO_CHAR (SAL, '$9,999')
FROM EMP;
```

```
TO_CHAR(SAL, '$9,999')
```

```
-----
```

```
$1,000
```

```
$1,600
```

```
$1,250
```

```
$2,975
```

Observati ca formatul este optional. Daca 'date picture' e omis, data este convertita la o valoare de tip caracter in format Oracle, implicit DD-MON-YY.

Daca 'number picture' nu e specificat, numarul e convertit intr-o valoare de tip caracter.

Observati de asemenea ca formatele nu afecteaza actuala reprezentare interna a valorii coloanei. Ele afecteaza doar cum valoarea coloanei este afisata cind e regasita cu o constructie SELECT.

## Formate pentru date calendaristice

Elementele unui format de data calendaristica sunt:

SCC or CC

Secol; Prefixul 'S' face ca datele i.e.n. sa fie prefixate de '-'.

YYYY or SYYYY

An; Prefixul 'S' face ca datele i.e.n. sa fie prefixate de '-'.

YYY or YY or Y

Ultimele 3,2,1 cifre din an

Y,YYY

An cu virgula in pozitie specificata.

SYEAR or YEAR

An in litere ; Prefixul 'S' face ca pt.anii i.e.n. sa se puna '-'.

BC or AD

Indicatorul BC/AD(pt.i.e.n. sau e.n.).

B.C.or A.D.

Indicatorul BC/AD cu puncte.

Q

Sfert de an.

MM

Luna.

MONTH

Numele lunii adaugind blancuri pina la lungimea 9.

MON

Numele lunii abreviat la 3 litere.

WW or W

Saptamina a anului sau a lunii.

DDD or DD or D

Zi a anului, lunii, saptaminii.

DAY

Numele zilei ,adaugind blancuri pina la lungimea de 9 caractere.

DY

Numarul zilei abreviat la 3 litere.

J

Nr.de zile din decembrie 31,4713 i.e.n.  
AM or PM Indicator meridian.  
A.M. or P.M. Indicator meridian cu puncte.  
HH or HH12 Ora din zi (1-12).  
HH24 Ora din zi (0-23).  
MI Minute.  
SS Secunde.  
SSSSS Secunde dupa miezul noptii(0-86399).  
/.,etc.  
"..." E reprodusa punctuatia in rezultat.  
"..." Sirul cu apostrofi reprodus in rezultat.

Prefixul de mai jos poate fi adaugat astfel:

fm 'FILL mode'.Prefixind MONTH sau DAY, suprima blancurile, lasind un rezultat de lungime variabila ,FM va suprima zerourile de inceput din formatul ddth.Nu e folositor cu alte coduri.A doua aparitie a lui 'fm' pune blan- curile din nou.

Sufixele de mai jos pot fi adaugate astfel:

TH nr.ordinal(ex:"DDTH" transformat in "4TH").  
SP nr.transformat in sir (ex.DDSP"pt"FOUR").  
SPTH sau thsp confirma numerele ordinale cu transformarea numar- cadru (ex.DDSPTH pt.FOURTH).

### **Observatie:**

? Codurile sunt case sensitive si vor afecta afisarea datelor calendaristice astfel:

DAY	MONDAY
Day	Monday
Month	July
ddth	14th
ddTh	14th

### **Formatul de date RR**

Daca-l folositi in locul lui YY, secolul variaza in acord cu anul specificat cu 2 cifre si ultimii 2 digiti ai anului curent.

Tabloul de mai jos rezuma comportamentul elementului RR.

## FORMATUL DE DATE RR(ORACLE 7)

### Pentru anii in alte secole

```

-----
|          |          |  Daca cei 2 digiti ai anului sint
|          |          |
|  Daca cei  -----
| 2 digiti |          |          0-49          50-99
|          |          |
| ai anului -----
| curent   |          |
| sunt     |  0   |  data intoarsa este   data intoarsa
este |    |  in secolul crt.     in secolul
anterior |    |
|         |  49 |
|         |    |
|         |    | -----
|         |  50 |  data intoarsa este   data intoarsa
este in|    |  in secolul  dupa     secolul crt.
|         |  -  |
|         |  99 |  cel crt.
|         |    |
|         |    | -----
-----

```

### EXEMPLE:

Anul crt.	Valoarea formatata (DD_MON_RR)	Anul interpretat
1994	27-oct-95	1995
1994	27-oct-17	2017
2001	27-oct-95	1995
2001	27-oct-17	2017

### Formatul pentru numere

Elementele modelului pt.formatul de numere sunt:



Format	Semnificatie	Exemple	
9	numere(nr.de 9 determina lung de afisare)	999999	1234
0	afis.zerourile de la inceput	099999	001234
\$	semnul dolar	\$999999	\$1234
.	punct zecimal in	999999.99	1234.00
,	pozitie specificata	999,999	1,234
MI	semnele minus la dreapta (valori negative)	999999MI	1234-
PR	paranteze pentru numere negative	999999PR	<1234>
EEEE	notatie stiintifica (formatul trebuie sa aiba 4E)	99.999EEEE	1.234E+03
V	inmultire cu 10 (n=numar de 9 dupa V)	9999V99	123400
B	afiseaza valori zero ca blancuri nu 0	B9999.99	1234.00

Observatie:

Formatele numerice afisate mai jos in tablou pot fi folosite cu comanda SQL\*PLUSCOLUMN(vezi cap.10).

### TO\_NUMBER

In exemplul urmatoare, functia TO\_NUMBER e folosita pentru a transforma un numar memorat sub forma unui caracter intr-un tip numar.

```
SELECT EMPNO, ENAME, JOB, SAL
FROM EMP
WHERE SAL > TO_NUMBER('1500');
```

### TO\_DATE

Pentru a afisa toti angajatii cu data angajarii 4 iulie 1984, puteti folosi functia TO\_DATE:

```
SELECT EMPNO, ENAME, HIREDATE
FROM EMP
```

```

WHERE HIREDATE =TO_DATE('June 4,1984','Month dd, YYYY');

EMPNO  ENAME      HIREDATE
-----
7844   TURNER      04-jun-84

```

Constanta e convertita intr-o data si apoi comparata cu valoarea HIREDATE.

Functia TO\_DATE e frecvent folosita pentru a transforma o valoare in ORACLE intr-un format diferit de cel implicit. De exemplu , cind inserati o data , Oracle asteapta o data in formatul implicit DD\_MON\_YY. Daca nu vreti sa folositi formatul implicit , trebuie sa folositi functia TO\_DATE si masca formatului dorit.

De exemplu:

Pentru a introduce in tabela EMP - o linie cu un format nestandard introduceti:

```

INSERT INTO EMP(EMPNO,DEPTNO,HIREDATE)
VALUES (777,,20,TO_DATE('19/08/90','DD/MM/YY'));

```

Comanda INSERT e tratata in detaliu mai tirziu.

## **Functii care accepta orice tip de data la intrare**

### **DECODE**

DECODE este cea mai puternica functie SQL.

Aceasta faciliteaza interogările, facind munca unui 'case' sau a unei constructii 'if-then-else'.

Syntax:

```

DECODE (col/expression,
        search1,rezult1,[search2,rezult2,....,]
        default)

```

Col/expression e comparata cu fiecare valoare'search'si intoarce 'rezult' daca col/expression este egal cu valoarea 'search'.

Daca nu e gasita nici o egalitate, functia DECODE intoarce valoarea 'default'.Daca valoarea'default' e omisa, NULL e intors pentru cazurile de nepotrivire.

### **ARGUMENTE**

DECODE trebuie sa aiba minim 4 parametrii ca argumente.

### **COL/EXPRESSION**

numele coloanei sau expresie pentru a fi evaluate

### **SEARCH1**

prima valoare pentru testare

RESULT1

valoarea intoarsa daca se potriveste cu SEARCH1.

SEARCH1 si RESULT1 pot fi repetate de cite ori e necesar-[SEARCH2, RESULT2,...]

DEFAULT

valoarea pentru a fi reintoarsa daca nu exista nici o potrivire.

Observatie:

- col/expression pot fi orice tip de data
- SEARCH poate fi acelasi tip ca expression /col
- Valoarea intoarsa este fortata la acelasi tip ca al treilea argument (result1)

Exemplul urmatoar decodifica tipurile 'CLERK'si 'MANAGER';altele nu testate.Acestea devin implicite pentru UNDEFINE.

```
SELECT ENAME,  
       JOB,  
       DECODE (JOB, 'CLERK', 'WORKER',  
               'MANAGER', 'BOSS',  
               'UNDEFINED') DECODED_JOB  
FROM EMP;
```

ENAME	JOB	DECODED_JOB
SMITH	CLERK	WORKER
ALLEN	SALESMAN	UNDEFINED
WARD	SALESMAN	UNDEFINED
JONES	MANAGER	BOSS
MARTIN	SALESMAN	UNDEFINED
BLAKE	MANAGER	BOSS
CLARK	MANAGER	BOSS
SCOTT	ANALYST	UNDEFINED
KING	PRESIDENT	UNDEFINED
TURNER	SALESMAN	UNDEFINED
ADAMS	CLERK	WORKER
JAMES	CLERK	WORKER
FORD	ANALYST	UNDEFINED
MILLER	CLERK	WORKER

Pentru a afisa procentajele in functie de coloane GRADE din tabela SALGRADE:

```
SELECT GRADE,  
       DECODE (GRADE, '1', '15%',  
               '2', '10%',  
               '3', '8%',  
               '5%') BONUS FROM SALGRADE;
```

GRADE	BONUS
1	15%
2	10%
3	8%
4	5%

Acest exemplu ilustreaza ca valoarea intoarsa e fortata la tipul celui de al treilea argument, la folosirea functiei DECODE. Il sfatuim pe utilizator sa specifice ordinea in care informatia e afisata introducind o valoare la fiecare rulare:

```
select * from emp
order by decode (&orderby,
                1,
                sal,
                2,
                ename,
                sal);

Enter value for orderby:2
order by decode (2,1,sal,2,ename,sal)
*
ERROR at line 2:ORA -1722: invalid number
```

Observati ca aceasta comanda produce o eroare fiindca tipul lui 'ename'(char) difera de cel al lui 'sal'(numar) care e al treilea argument.

In exemplul de mai jos , dorim sa introducem salariul crescut in functie de meseria fiecarui angajat.

```
SELECT JOB,SAL,DECODE(JOB,'ANALYST',SAL*1.1,
                      'CLERK',SAL*1.15,
                      'MANAGER',SAL*0.95,
                      SAL) DECODED_SALARY FROM EMP;
```

JOB	SAL	DECODED_SALARY
CLERK	800	920
SALESMAN	1,600	1600
SALESMAN	1,250	1250
MANAGER	2,975	2826.25
SALESMAN	1,250	1250
MANAGER	2,850	2707.5
MANAGER	2,450	2327.5
ANALYST	3,000	3300
PRESIDENT	5,000	5000
SALESMAN	1,500	1500
CLERK	1,100	1265
CLERK	950	1092.5
ANALYST	3,000	3300
CLERK	1,300	1495

NVL(col/value,val) converteste o valoare NULL la 'val' Tipul datei trebuie sa se potriveasca cu (col/value si val).

```
SELECT SAL*12+NVL(COMM,0),NVL(COMM,1000),SAL*12+NVL(COMM,1000)
```

```
FROM EMP
WHERE DEPTNO=10;
```

```
SAL*12+NVL (COMM, 0) NVL (COMM, 1000) SAL*12+NVL (COMM, 1000)
```

```
-----
29400          1000          30400
60000          1000          61000
15600          1000          16600
```

GREATEST(col/value1,col/value2....) -intoarce cel mai mare dintr-o lista de valori.Toate 'col/value2' sint convertite la tipul col/value1 inaintea comparatiei.

```
SELECT GREATEST (1000, 2000) ,GREATEST (SAL, COM)
FROM EMP
WHERE DEPTNO=30;
```

```
GREATEST (1000, 2000)  GREATEST (SAL, COMM)
-----
                2000          1600
                2000          1250
                2000          1400
                2000
                2000          1500
                2000
```

6 records selected.

LEAST(col/value1,col/value2....) intoace cea mai mica valoare.  
Toate col/value 2 sint convertite la tipul col/value1 inainte de comparatie.

```
SELECT LEAST (1000, 2000) ,LEAST (SAL, COMM)
FROM EMP
WHERE DEPTNO=30;
```

```
LEAST (1000, 2000)  LEAST (SAL, COMM)
-----
                1000          300
                1000          500
                1000          1250
                1000
                1000          0
                1000
```

6 records selected.

VSIZE(col/value) - intoarce nr.de biti din reprezentarea interna ORACLE a lui col/value'.

```
SELECT DEPTNO, VSIZE (DEPTNO) , VSIZE (HIREDATE) , VSIZE (SAL) ,
       VSIZE (ENAME)
FROM EMP
WHERE DEPTNO=10;
```

```
DEPTNO VSIZE (DEPTNO) VSIZE (HIREDATE) VSIZE (SAL) VSIZE (ENAME)
-----
```

10	2	7	3	5
10	2	7	2	4
10	2	7	2	6

## FUNCTII IMBRICATE REVIZITATE

Amintim ca functiile pot fi imbricate la orice nivel si functiile interioare sunt evaluate intii mergind pina la functia cea mai exterioara. Este bine sa se tina nr. de paranteze de inceput (deschidere) si de sfirsit (inchidere). Trebuie sa fie acelasi numar din fiecare tip de paranteze.

Funcitiile de mai jos sint imbricate si sint evaluate dupa cum urmeaza.

```

SELECT      ENAME, NVL (TO_CHAR (MGR), 'UNMANAGEABLE')
FROM EMP
WHERE MGR IS NULL;

ENAME      NVL (TO_CHAR (MGR), 'UNMANAGEABLE')
-----
KING      UNMANAGEABLE

```

1. Coloana MGR e convertita la caracter cu functia TO\_CHAR
2. Functia NVL inlocuieste NULL mgr cu un sir de caractere 'UNMANAGEABLE'.

Funcitiile imbricate pot fi de asemenea folosite pentru a afisa date de vineri , la doua luni de azi in formatul Day dd Month YYYY.

```

SELECT SYSDATE,
       TO_CHAR (NEXT_DAY (ADD_MONTHS (SYSDATE, 2), 'FRIDAY'), 'Day
dd
       Month YYYY')
FROM SYS.DUAL;

SYSDATE
TO_CHAR (NEXT_DAY (ADD_MONTHS (SYSDATE, 2), 'FRIDAY'), 'DAYDDMONTHYYYY')
-----
04-dec-89 Friday      09 Februarie 1990

```

1. Functia ADD\_MONTHS aduna 2 luni la luna curenta (decembrie)
- 2.
3. Functia NEXT\_DAY gaseste vineri la 2 luni de SYSDATE
- 4.
5. Functia TO\_CHAR converteste coloana data la tipul CHAR pentru
6. afisarea in formatul Day dd Month YYYY
- 7.

## Capitolul 5 Exerciții - Funcții

Aceste exerciții acopera funcțiile discutate în acest capitol, ca și

pe cele din capitolul 4.

1. Afisati pt.fiecare angajat din departamentul 20 numele si data
2. angajarii.
3. Fiti siguri ca specificati aliasul 'DATE\_HIRED' dupa expresia
4. voastra altfel coloana respectiva va fi pierduta. Se folo-
5. seste o lungime de 80 de caractere care este lungimea implicita
6. pentru coloane de caracter.
7. ENAME DATE\_HIRED
8. -----
9. SMITH June,Thirteenth 1983
10. JONES October,Thirty-First 1983
11. SCOTT March,Fifth 1984
12. ADAMS June,Fourth 1984
13. FORD December,Fifth 1983
- 14.
- 15.
16. Afisati pt.fiecare angajat numele, data angajarii si adaugati
17. un an la data angajarii pt.fiecare. Ordonati iesirea in
18. ordinea crescatoare a datei angajarii la care s-a adaugat
19. un an.
20. ENAME HIREDATE REVIEW
21. -----
22. SMITH 13-jun-83 13-jun-84
23. ALLEN 15-aug-83 15-aug-84
24. JONES 31-oct-83 31-oct-84
25. MILLER 21-nov-83 21-nov-84
26. MARTIN 05-dec-83 05-dec-84
27. FORD 05-dec-83 05-dec-84
28. SCOTT 05-mar-84 05-mar-85
29. WARD 26-mar-84 26-mar-85
30. CLARK 14-may-84 14-may-85
31. TURNER 04-jun-84 04-jun-85
32. ADAMS 04-jun-84 04-jun-85
33. BLAKE 11-jun-84 11-jun-85

```

34.      KING      09-jun-84  09-jun-85
35.      JAMES     23-jun-84  23-jun-85
36.      14 records selected
37.
38.

39.  Afisati lista angajatilor , pentru fiecare afisind salariul
40.      daca acesta e mai mare ca 1500, daca e egal cu 1500
afiseaza
41.      'On Target', daca e mai mic decit 155, afiseaza
'BELOW 1500'.
42.      ENAME      SALARY
43.      -----
44.      ADAMS      Below 1500
45.      ALLEN      1600
46.      BLAKE      2850
47.      CLARK      2450
48.      FORD       3000
49.      JAMES      Below 1500
50.      JONES      2975
51.      KING       5000
52.      MARTIN     Below 1500
53.      MILLER     Below 1500
54.      SCOTT      3000
55.      SMITH      Below 1500
56.      TURNER    On Target
57.      WARD       Below 1500
58.
59.      14 records selected
60.
61.
62.

63.  Scrieti o cerere care intoarce ziua din saptamina
64.      (ex:MONDAY), pt. fiecare data introdusa in formatul
65.      DD.MM.YY.
66.      Enter value for anydate:12.11.88
67.
68.      DAY
69.      -----
70.      SATURDAY
71.
72.

73.  Scrieti o cerere pt. a determina cit timp au fost
angajati ai
74.      companiei .Folositi DEFINE pentru a evita repetarea
tipului
75.      functiilor.
76.      Enter value for employee_name:king
77.
78.
79.      ENAME      LENGTH OF SERVICE
80.      -----
81.      KING       4 YEARS 4 MONTHS
82.
83.

84.  Dindu-se un sir in formatul 'nn/nn', verificati daca primele
85.      si ultimele 2 caractere sunt numere si daca
caracterul din

```



86. mijloc este '/'.

87. Afisati expresia 'Yes' daca e asa , sau 'No' altfel.  
Folositi

88. urmatoarele valori pentru a testa solutia voastra:  
89. '12/34',01/1a','99/88'  
90. VALUE VALID?  
91. -----  
92. 12/34 YES  
93.  
94.

95. Angajatii pe 15 ale fiecărei luni sau înainte sint platiti  
96. in ultima vineri a acelei luni . Cei angajati dupa  
data de  
97. 15 sunt platiti in ultima vineri a urmatoarei luni.  
98. Afisati o lista a angajatilor , cu data de angajare  
si prima  
99. data de plata.Sortati in functie de data angajarii.

100.	ENAME	HIREDATE	PAYDAY
101.	-----		
102.	SMITH	13-jun-83	24-jun-83
103.	ALLEN	15-aug-83	26-aug-83
104.	JONES	31-oct-83	25-nov-83
105.	MILLER	21-nov-83	30-dec-83
106.	MARTIN	05-dec-83	30-dec-83
107.	FORD	05-dec-83	30-dec-83
108.	SCOTT	05-mar-84	30-mar-84
109.	WARD	26-mar-84	27-apr-84
110.	CLARK	14-may-84	25-may-84
111.	TURNER	04-jun-84	29-jun-84
112.	ADAMS	04-jun-84	29-jun-84
113.	BLAKE	11-jun-84	29-jun-84
114.	KING	09-jul-84	27-jul-84
115.	JAMES	23-jul-84	31-aug-84
116.			
117.	14 records selected		
118.			
119.			

## CAPITOLUL 5 SOLUTII

```

1.      SELECT  ENAME,
2.             TO_CHAR(HIREDATE, 'fmMonth, Ddspth YYYY')
3.             date_hired
4.      FROM    EMP
5.      WHERE   DEPTNO=20;
6.
7.
8.      SELECT  ENAME, HIREDATE, ADD_MONTHS (HIREDATE, 12)
REVIEW
9.      FROM    EMP
10.     ORDER  BY  ADD_MONTHS(hiredate, 12);

```

```

11.
12.
13.          SELECT  ENAME,
14.                DECODE (SIGN (1500-SAL), 1, 'BELOW 1500', 0, 'On
15.                Target', SAL)
16.                SALARY
17.          FROM    EMP
18.          ORDER BY ENAME;
19.
20.
21.          SELECT  TO-
22.          CHAR (TO_DATE ('&ANYDATE', 'DD.MM.YY'), 'DAY') DAY
23.          FROM    SYS.DUAL;
24.
25.          DEFINE TIME=MONTHS_BETWEEN (SYSDATE, HIREDATE)
26.          SELECT  ENAME,
27.                FLOOR (&TIME/12) || 'YEARS' ||
28.                FLOOR (MOD (&TIME, 12)) || 'MONTHS '
29.                "LENGTH OF SERVICE"
30.          FROM    EMP
31.          WHERE   ENAME=UPPER ('&EMPLOYEE_NAME');
32.
33.
34.          SELECT  '12/34' VALUE
35.                DECODE (
36.                TRANSLATE ('12/34', '1234567890',
37.                '9999999999'),
38.                '99/99', 'YES', NO) "VALID?"
39.          FROM    SYS.DUAL;
40.
41.
42.          SELECT  ENAME,
43.                HIREDATE,
44.                DECODE (SIGN (TO_CHAR (HIREDATE, 'DD') -15
45.                1, NEXT-DAY (LAST-
46.                DAY (ADD_MONTHS (HIREDATE, 1)),
47.                'FRIDAY') -7,
48.                NEXT-DAY (LAST_DAY (HIREDATE), 'FRIDAY') -
49.                7)
50.                PAYDAY
51.          FROM    EMP
52.          ORDER BY HIREDATE;
53.
54.          sau
55.          SELECT  ENAME, HIREDATE,
56.                NEXT_DAY (LAST_DAY (ROUND (HIREDATE, 'MONTH')) -7
57.                'FRIDAY') PAYDAY
58.          FROM    EMP
59.          ORDER BY HIREDATE;

```