



Laborator 4 - Limbajul SQL

Funcții referitoare la o singură înregistrare (single-row functions)

Funcțiile pe un singur rand cuprind următoarele tipuri de funcții:

1. funcții de tip caracter
2. funcții de tip numeric
3. funcții de tip data
4. funcții de conversie
5. funcții generale: NVL, NVL2, NULLIF, COALSECE, CASE, DECODE

1) Funcții de tip caracter (vezi curs 4)

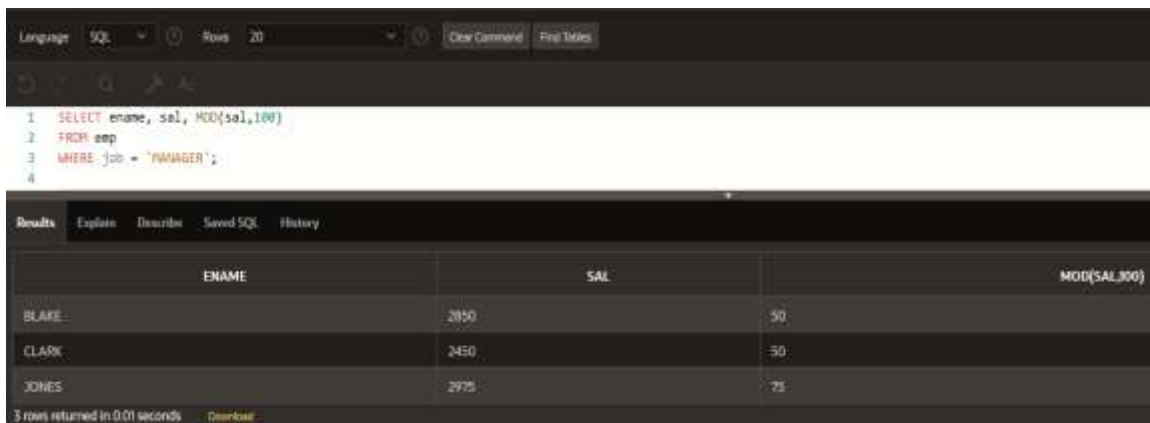
2) Funcții de tip numeric

Funcțiile pe un singur rand de tip numeric accepta ca date de intrare date de tip numeric.

Funcție	Descriere
ROUND(column expression, n)	rotunjește la o valoare cu <i>n</i> zecimale
TRUNC(column expression, n)	se obtine o valoare cu <i>n</i> zecimale prin trunchare
MOD(m, n)	returnează restul împărțirii lui <i>m</i> la <i>n</i>

Exemplu:

```
SELECT ename, sal, MOD(sal,100)
FROM emp
WHERE job = 'MANAGER';
```





3) Functii de tip data

Oracle retine datele intr-un format numeric intern, reprezentand secolul, anul, luna, ziua, ora, minutele si secunde. Formatul standard in Oracle este DD-MM-YY.

In Oracle datele valide sunt cuprinse intre 1 ianuarie 4712 B.C. si 31 decembrie 9999 A.D.

Funcția SYSDATE returneaza data si ora sistemului. SYSDATE se foloseste ca orice alt nume de coloana, se poate afisa data curenta selectand SYSDATE dintr-o tabela.

Deoarece o baza de date contine atat numere, cat si date, se pot folosi operatorii aritmetici ca adunarea si scaderea intre numere si date.

Operatie	Rezultat	Descriere
data + numar	Data	Adauga un numar de zile la o anumita data
data - numar	Data	Scade un numar de zile dintr-o anumita data
data - data	Numar de zile	Scade o data din alta data
data + number/24	Data	Adauga un numar de ore la o data

SYSDATE este o functie SQL care returneaza data curenta si ora.

SELECT SYSDATE FROM DUAL

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this, the 'SQL Commands' section is active, showing the query: `1 SELECT SYSDATE`
`2 FROM DUAL;`
`3`. The 'Results' tab is selected, displaying the output: `SYSDATE`
`02/15/2022`. At the bottom, it indicates '1 rows returned in 0.01 seconds' and provides a 'Download' link.

Baze de date - SQL



Principalele functii de tip data folosite in **Oracle** sunt:

1. ***MONTHS_BETWEEN (date1, date2)***: Returneaza numarul de luni dintre *date1* si *date2*. Rezultatul poate fi pozitiv sau negativ. Daca *date1* este mai mare decat *date2* rezultatul este pozitiv, daca *date1* este mai mica decat *date2* rezultatul este negativ. Valoarea zecimala reprezinta o parte a lunii.
2. ***ADD_MONTHS (date, n)***: Adauga *n* luni la o *date*. *n* trebuie sa aiba o valoare intreaga si poate fi negativ.
3. ***NEXT_DAY (date, 'char')***: Returneaza data urmatoare unei zile specificate. Argumentul *char* poate fi un numar sau un sir de caractere.
4. ***LAST_DAY (date)***: Returneaza ultima zi din luna.
5. ***ROUND (date [, 'fmt'])***: Returneaza data rotunjita la unitatea specificata de *fmt*. Daca formatul *fmt* lipseste data este rotunjita la cea mai apropiata zi.
6. ***TRUNC (date [, 'fmt'])***: Returneaza data truncata la unitatea specificata de *fmt*. Daca formatul *fmt* lipseste data este rotunjita la cea mai apropiata zi.



Exercitii rezolvate:

1. Sa se afiseze pentru fiecare angajat id-ul data anajarii si numarul de zile de munca in firma:

```
SELECT empno, hiredate, round(sysdate-hiredate,2) "Numar de zile"  
FROM emp;
```

EMPNO	HIREDATE	Numar de zile
7650	11/17/1981	14722.59
7668	01/01/1981	14722.59
7782	06/07/1981	14681.59
7566	04/02/1981	14650.59
7788	12/09/1982	14313.59
7502	12/03/1981	14706.59
7569	12/17/1980	15057.59
7499	02/20/1981	14992.59

2. Sa se afiseze pentru fiecare angajat id-ul data angajarii si numarul de saptamani de munca in firma:

```
SELECT empno, hiredate, round((sysdate-hiredate)/7,2) "Numar de saptamani"  
FROM emp;
```

EMPNO	HIREDATE	Numar de saptamani
7650	11/17/1981	2103.23
7668	01/01/1981	2118.8
7782	06/07/1981	2096.25
7566	04/02/1981	2115.94
7788	12/09/1982	2047.94
7900	12/03/1981	2100.94
7569	12/17/1980	2131.06
7499	02/20/1981	2141.8
7521	02/22/1981	2141.57

3. Sa se afiseze pentru fiecare angajat id-ul data anajarii si numarul de luni de munca in firma:

```
SELECT empno, hiredate, months_between(sysdate,hiredate) "Numar de luni"  
FROM emp  
ORDER BY hiredate desc;
```

Baze de date - SQL



EMPNO	HIREDATE	Numar de luni
9999	02/17/2022	75031362007968458781362007968458781362
7876	01/12/1983	469922464531063321385902037063321385902
7788	12/09/1982	471
7954	01/23/1982	481567625821385902037063321385902037063
7902	12/03/1981	481521278711708482676224611708482676225
7900	12/03/1981	481521278711708482676224611708482676225
7839	11/17/1981	4815761174208482676224611708482676224612
7654	09/28/1981	485406335498805256869772998805256869773
7844	09/08/1981	486051496789127837514934289127837514934

4. Sa se afiseze urmatoarea zi de Miercuri (Wednesday):

```
SELECT next_day(sysdate,'Wednesday')
FROM dual;
```

NEXT_DAY(SYSDATE,'WEDNESDAY')
03/16/2022

1 rows returned in 0.00 seconds [Download](#)

5. Sa se afiseze, pentru fiecare angajat, numarul de zile de miercuri pe care le-a lucrat in firma:

```
SELECT empno, to_char(hiredate,'Day'), round((sysdate-hiredate)/7) +
decode(to_char(hiredate,'Day'),'Wednesday',1,0) "Numar de zile de Miercuri lucrate"
FROM emp;
```

EMPNO	TO_CHAR(HIREDATE,'DAY')	Numar de zile de Miercuri lucrate
7839	Tuesday	2101
7699	Friday	2112
7782	Tuesday	2106
7566	Thursday	2156
7788	Thursday	2048
7902	Thursday	2101
7569	Wednesday	2152
7499	Friday	2142
7521	Sunday	2142
7654	Monday	2110

Baze de date - SQL



6. Sa se afiseze numele, job-ul si salarial fiecarui angajat impreuna cu valoarea de trei ori a salariului, toate aceste informatii concatenate intr.-o singura propozitie:

```
SELECT ename || ' ' || job || ' castiga ' || sal || ' pe luna dar doreste sa castige ' || sal * 3 AS  
"Coloane concatenate"  
FROM emp;
```

Coloane concatenate
KING PRESIDENT castiga 5000 pe luna dar doreste sa castige 15000
BLAKE MANAGER castiga 2850 pe luna dar doreste sa castige 8550
CLARK MANAGER castiga 2450 pe luna dar doreste sa castige 7350
JONES MANAGER castiga 2975 pe luna dar doreste sa castige 8925
SCOTT ANALYST castiga 3000 pe luna dar doreste sa castige 9000
FORD ANALYST castiga 3000 pe luna dar doreste sa castige 9000
SMITH CLERK castiga 800 pe luna dar doreste sa castige 2400
ALLEN SALESMAN castiga 1600 pe luna dar doreste sa castige 4800
WARD SALESMAN castiga 1250 pe luna dar doreste sa castige 3750
MARTIN SALESMAN castiga 1250 pe luna dar doreste sa castige 3750

7. Sa afiseze numele angajatului cu prima litera majuscula, apoi job-ul si lungimea numelui. Se vor afiza doar acele nume care incep cu literele 'j' sau 'm' sau care au a treia litera din nume litera 'a'. Rezultatul interogarii va fi afisat in ordine descrescatoare dupa lungimea numelui:

```
SELECT initcap(ename) "Nume", upper(job) "Job", length(ename) "Lungimea numelui"  
FROM emp  
WHERE lower(ename) like 'j%' or lower(ename) like 'm%' or lower(ename) like '__a%'  
ORDER BY length(ename) desc;
```

Nume	Job	Lungimea numelui
Martin	SALESMAN	6
Miller	CLERK	6
Jones	MANAGER	5
James	CLERK	5
Adams	CLERK	5
Clark	MANAGER	5
Blake	MANAGER	5

7 rows returned in 0.03 seconds

Baze de date - SQL



Exemple:

```
SELECT ename, hiredate  
FROM EMP  
WHERE ename like '%S';
```

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is: `SELECT ename, hiredate FROM EMP WHERE ename like 'S';`. The results table displays three rows:

ENAME	HIREDATE
JONES	04/02/1981
ADAMS	01/12/1983
JAMES	12/03/1981

3 rows returned in 0.01 seconds

```
SELECT ename, ROUND((SYSDATE - hiredate)/7) AS SAPTAMANI  
FROM EMP  
WHERE job = 'SALESMAN';
```

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is: `SELECT ename, ROUND((SYSDATE - hiredate)/7) AS SAPTAMANI FROM EMP WHERE job = 'SALESMAN';`. The results table displays four rows:

ENAME	SAPTAMANI
ALLEN	2189
WARD	2158
MARTIN	2107
TURNER	2110

4 rows returned in 0.01 seconds

Baze de date - SQL



```
SELECT empno, hiredate, ROUND(hiredate, 'MONTH'), TRUNC(hiredate, 'MONTH')
FROM EMP
WHERE hiredate LIKE '%80';
```

The screenshot shows the Oracle SQL Developer interface. The SQL Command window contains the following query:

```
1. SELECT empno, hiredate, ROUND(hiredate, 'MONTH'), TRUNC(hiredate, 'MONTH')
2. FROM EMP
3. WHERE hiredate LIKE '%80';
4.
5.
```

The Results window shows the following output:

EMPNO	HIREDATE	ROUND(HIREDATE, MONTH)	TRUNC(HIREDATE, MONTH)
7566	12/17/1980	01/01/1981	12/01/1980

1 rows returned in 0.01 seconds. Download

4) Functii de conversie

In unele cazuri Oracle foloseste date de un anumit tip atunci cand astepta date de un alt tip. Cand se intampla acest lucru, Oracle poate converti automat datele in tipul de date dorit.

Acesta conversie poate fi facuta *implicit* de serverul Oracle sau *explicit* de utilizator. Conversia *implicita* se face conform anumitor reguli, iar conversia *explicita* se face folosind functii de conversie.

Funcțiile de conversie convertesc o valoare de un anumit tip in alt tip. SQL contine trei astfel de functii:

Funcție	Descriere
TO_CHAR (number date, [fmt])	Converteste o valoare de tip numeric sau data intr-un sir de caractere ce are formatul specificat optional de <i>fmt</i> .
TO_NUMBER (number date, [fmt])	Converteste un sir de caractere ce contine cifre intr-un numar in formatul specificat optional de <i>fmt</i> .
TO_DATE (number date, [fmt])	Converteste un sir de caractere reprezentand o data intr-o valoare de tip data avand formatul <i>fmt</i> specificat. Daca formatul nu este specificat, formatul este DD-MON-YY.

Baze de date - SQL



Exemple:

```
SELECT empno, TO_CHAR(hiredate, 'MM/YY') Luna_Angajarii  
FROM EMP  
WHERE ename = 'SMITH';
```

```
1 SELECT empno, TO_CHAR(hiredate, 'MM/YY') Luna_Angajarii  
2 FROM EMP  
3 WHERE ename = 'SMITH';  
4  
5
```

EMPNO	LUNA_ANGAJARII
7369	12/80

1 rows returned in 0.01 seconds [Download](#)

```
SELECT ename, TO_CHAR(hiredate, 'fmDD Month YYYY') AS DATA_ANGAJARII  
FROM EMP;
```

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20 Clear Command Find Tables

```
1 SELECT ename, TO_CHAR(hiredate, 'fmDD Month YYYY') AS DATA_ANGAJARII  
2 FROM EMP;
```

ENAME	DATA_ANGAJARII
FRIG	17 November 1981
BLAKE	1 May 1981
CLARK	9 June 1981
XPES	7 April 1981
SCOTT	9 December 1982
FORD	3 December 1981
SMITH	17 December 1980
ALLEN	20 February 1981
WARD	22 February 1981
MARTIN	28 September 1981
TURNER	8 September 1981
ADAMS	12 January 1983
JAMES	8 December 1981
MILLER	23 January 1982

14 rows returned in 0.01 seconds [Download](#)

Baze de date - SQL



```
SELECT TO_CHAR(sal, '$99,999.00') SALARIU
FROM EMP
WHERE ename = 'FORD';
```

```
1 SELECT TO_CHAR(sal, '$99,999.00') SALARIU
2 FROM EMP
3 WHERE ename = 'FORD';
4
5
```

Results	Explain	Describe	Saved SQL	History
				SALARIU
\$3,000.00				
1 rows returned in 0.00 seconds				Download

5) Functii generale

Funcțiile generale sunt: **NVL**, **NVL2**, **NULLIF** și **COALESCE**, aceste funcții lucrează cu orice tip de date.

Funcție	Descriere
NVL (expr1, expr2)	Converteste o valoare nula într-o valoare actuală.
NVL2 (expr1, expr2, expr3)	Dacă <i>expr1</i> nu este nula, NVL2 returnează <i>expr2</i> . Dacă <i>expr1</i> este nula, NVL2 returnează <i>expr3</i> . Argumentul <i>expr1</i> poate fi orice tip de date.
NULLIF (expr1, expr2)	Compara două expresii și returnează valoarea nula dacă sunt egale sau prima expresie dacă nu sunt egale.
COALESCE (expr1, expr2, ..., expr n)	Returnează prima expresie non-nula din lista de expresii.

Exemple:

Funcția **NVL** converteste o valoare nula într-o valoare actuală. Funcția **NVL** se poate folosi pentru a converti orice tip de date, dar valoarea rezultată are întotdeauna același tip cu *expr1*.

Baze de date - SQL



SELECT ename, sal, NVL(comm, 0), (sal * 12) + (sal * 12 * NVL(comm, 0)) AN_SAL
FROM EMP;

```
1 SELECT ename, sal, NVL(comm, 0), (sal * 12) + (sal * 12 * NVL(comm, 0)) AN_SAL
2 FROM EMP;
3
4
```

ENAME	SAL	NVL(COMM,0)	AN_SAL
KING	5000	0	60000
BLAKE	2850	0	34200
CLARK	2450	0	29400
JONES	2975	0	35700
SCOTT	3000	0	36000
FORD	3000	0	36000
SMITH	800	0	9600
ALLEN	1600	300	579200
WARD	1250	900	756000
MARTIN	1250	1400	2105000
TURNER	1500	0	18000

SELECT ename, sal, com, (sal * 12) + (sal * 12 * comm) AN_SAL
FROM EMP;

```
1 SELECT ename, sal, com, (sal * 12) + (sal * 12 * comm) AN_SAL
2 FROM EMP;
3
4
```

ENAME	SAL	COMM	AN_SAL
KING	5000	-	-
BLAKE	2850	-	-
CLARK	2450	-	-
JONES	2975	-	-
SCOTT	3000	-	-
FORD	3000	-	-
SMITH	800	-	-
ALLEN	1600	300	579200
WARD	1250	900	756000
MARTIN	1250	1400	2105000
TURNER	1500	0	18000

Baze de date - SQL



Funcția **NVL2** examinează prima expresie. Dacă prima expresie nu este nulă, atunci funcția NVL2 returnează a doua expresie. Dacă prima expresie este nulă, atunci a treia expresie este returnată.

```
SELECT ename, sal, comm, NVL2(comm, 'SAL + COMM', 'SAL') INCOME  
FROM EMP  
WHERE deptno = 30;
```

```
1 SELECT ename, sal, comm, NVL2(comm, 'SAL + COMM', 'SAL') INCOME  
2 FROM EMP  
3 WHERE deptno = 30;  
4  
5
```

ENAME	SAL	COMM	INCOME
BLAKE	2850	-	SAL
ALLEN	1600	300	SAL + COMM
WARD	1250	500	SAL + COMM
MARTIN	1250	1400	SAL + COMM
TURNER	1500	0	SAL + COMM
JAMES	950	-	SAL

6 rows returned in 0.01 seconds [Download](#)

Funcția **NULLIF** compară două expresii. Dacă cele două expresii sunt egale, funcția returnează valoare nulă. Dacă cele două expresii nu sunt egale, funcția returnează prima expresie. Funcția NULLIF este echivalentă cu expresia CASE.

```
SELECT ename, LENGTH(ename) EXPR_1, job, LENGTH(job) EXPR_2, NULLIF  
(LENGTH(ename), LENGTH(job)) RESULT  
FROM EMP;
```

```
1 SELECT ename, LENGTH(ename) EXPR_1, job, LENGTH(job) EXPR_2, NULLIF  
2 FROM EMP;  
3  
4  
5
```

ENAME	EXPR_1	JOB	EXPR_2	RESULT
KING	4	PRESIDENT	9	4
BLAKE	5	MANAGER	7	5
CLARK	5	MANAGER	7	5
JONES	5	MANAGER	7	5
SCOTT	5	ANALYST	7	5
FORD	4	ANALYST	7	4
SMITH	5	CLERK	5	-
ALLEN	5	SALESMAN	8	5
WARD	4	SALESMAN	8	4
MARTIN	6	SALESMAN	8	6

Baze de date - SQL



Funcția **COALESCE** returnează prima expresie care nu este nulă.

```
SELECT ename, COALESCE (comm, sal, 10) COMM  
FROM EMP  
ORDER BY comm;
```

```
1 SELECT ename, COALESCE (comm, sal, 10) COMM  
2 FROM EMP  
3 ORDER BY comm;  
4  
5
```

ENAME	COMM
TURNER	0
ALLEN	300
WARD	500
SMITH	800
JAMES	950
ADAMS	1100
MILLER	1300
MARTIN	1400
CLARK	2450
BLAKE	2850