

# Baze de date - Rezolvari Exercitii SQL

## 1 Exercitii

### 1.1 Functii pe siruri de caractere

1. Scrieti o cerere care afiseaza pentru fiecare angajat: <prenume angajat><nume angajat>castiga <salariu>lunar dar doreste <salariu de 3 ori mai mare>.

```
1 SELECT first_name || ' ' || last_name || ' castiga ' ||  
2 salary || ' lunar dar doreste ' || (salary * 3)  
   AS "Salariu ideal"  
3 FROM employees;
```

2. Afisati prenumele cu prima litera majuscula si numele cu majuscule pentru angajatii al caror nume incepe cu J sau M sau care au a treia litera din nume A. Rezultatul va fi ordonat descrescator dupa lungimea numelui.

```
1 SELECT INITCAP(first_name) AS Prenume, UPPER(last_name)  
   AS Nume  
2 FROM employees  
3 WHERE last_name LIKE 'J%' OR last_name LIKE 'M%' OR  
   last_name LIKE '__A%'  
4 ORDER BY LENGTH(last_name) DESC;
```

### 1.2 Functii aritmetice

3. Afisati codul salariatului, numele, salariul, salariul marit cu 15% (exprimat cu doua zecimale) si numarul de sute al salariului nou. Etichetati ultimele doua coloane "Salariu nou", respectiv "Numar sute".

```
1 SELECT employee_id, last_name, salary,  
2 ROUND(salary * 1.15, 2) AS "Salariu nou",  
3 TRUNC(salary * 1.15 / 100) AS "Numar sute"  
4 FROM employees;
```

4. Listati numele, salariul si o coloana care sa reprezinte nivelul venitului (pentru fiecare 1000 sa fie folosit cate un simbol \$).

```

1 SELECT last_name, salary,
2        RPAD('$', ROUND(salary/1000), '$') AS "Nivel
   venit"
3 FROM employees;

```

### 1.3 Functii si operatii cu date calendaristice

5. Afisati data (numele lunii, ziua, anul, ora, minutul si secunda) de peste 30 zile.

```

1 SELECT TO_CHAR(SYSDATE + 30, 'Month DD, YYYY HH24:MI:SS'
2        ) AS "Data peste 30 zile"
3 FROM dual;

```

6. Afisati pentru fiecare angajat numele si data negocierii salariului, care este prima zi de Luni dupa 6 luni de serviciu. Etichetati aceasta coloana "Negociere".

```

1 SELECT last_name,
2        NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY') AS "
   Negociere"
3 FROM employees;

```

7. Afisati numele, data angajarii si ziua saptamanii in care a inceput lucrul fiecare salariat. Etichetati coloana "Zi". Ordonati rezultatul dupa ziua saptamanii, incepand cu Luni.

```

1 SELECT last_name, hire_date,
2        TO_CHAR(hire_date, 'Day') AS "Zi"
3 FROM employees
4 ORDER BY TO_CHAR(hire_date, 'D');

```

### 1.4 Functii diverse

8. Afisati numele angajatilor si comisionul. Daca un angajat nu castiga comision, sa se scrie "Fara comision". Etichetati coloana "Comision".

```

1 SELECT last_name,
2        NVL(TO_CHAR(commission_pct), 'Fara comision') AS
   "Comision"
3 FROM employees;

```

9. Afisati numele, codul job-ului, salariul si o coloana care sa arate salariul dupa marire. Se presupune ca pentru IT\_PROG are loc o marire de 20%, pentru SA\_REP cresterea este de 25%, iar pentru SA\_MAN are loc o marire de 35%. Pentru ceilalti angajati nu se acorda marire.

```

1 SELECT last_name, job_id, salary,
2        CASE job_id
3            WHEN 'IT_PROG' THEN salary * 1.20
4            WHEN 'SA_REP' THEN salary * 1.25
5            WHEN 'SA_MAN' THEN salary * 1.35
6            ELSE salary
7        END AS "Salariu dupa marire"
8 FROM employees;

```

– Alternativ, folosind DECODE:

```

1 SELECT last_name, job_id, salary,
2        DECODE(job_id,
3            'IT_PROG', salary * 1.20,
4            'SA_REP', salary * 1.25,
5            'SA_MAN', salary * 1.35,
6            salary) AS "Salariu dupa marire"
7 FROM employees;

```