

Baze de Date

Solutii – Laborator 2 – Saptamana 4

-- LABORATOR 2 - SAPTAMANA 4

5. Să se afișeze detalii despre salariații care au lucrat un număr întreg de săptămâni până la data curentă.

```
SELECT *  
FROM employees  
WHERE MOD(ROUND(sysdate - hire_date), 7) = 0;
```

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.

```
SELECT employee_id, last_name, salary,  
       round(salary + 0.15 * salary, 2) "Salariu Nou",  
       round((salary + 0.15 * salary) / 100, 2) "Numar sute"  
FROM employees  
--WHERE MOD(salary, 1000) != 0;
```

--WHERE MOD(salary, 1000) <> 0;

WHERE MOD(salary, 1000) ^= 0;

7. Să se listeze numele și data angajării salariaților care câștigă comision.

Să se eticheteze coloanele „Nume angajat”, „Data angajării”.

Utilizați funcția RPAD pentru a determina ca data angajării să aibă lungimea de 20 de caractere.

```
SELECT last_name AS "Nume angajat" , RPAD(to_char(hire_date),20,'X') "Data
angajarii"
```

```
FROM employees
```

```
WHERE commission_pct IS NOT NULL;
```

8. Să se afișeze data (numele lunii, ziua, anul, ora, minutul și secunda) de peste 30 zile.

```
SELECT TO_CHAR(SYSDATE + 30, 'MONTH DD YYYY HH24:MI:SS') "Data"
```

```
FROM DUAL;
```

9. Să se afișeze numărul de zile rămase până la sfârșitul anului.

```
SELECT ROUND(to_date('31-12-2023', 'DD-MM-YYYY') - sysdate)
```

```
FROM dual;
```

10. a) Să se afișeze data de peste 12 ore.

--Cat reprezinta 12 ore dintr-o zi

```
SELECT TO_CHAR(SYSDATE + 12/24, 'DD/MM HH24:MI:SS') "Data"
FROM DUAL;
```

b) Să se afișeze data de peste 5 minute

Obs: Cât reprezintă 5 minute dintr-o zi?

-- cat reprezinta 5 minute dintr-o ora

-- cat reprezinta 5 min dintr-o zi

```
SELECT TO_CHAR(SYSDATE + 1/12/24, 'DD/MM HH24:MI:SS') "Data"
FROM DUAL;
```

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”.

```
SELECT concat(last_name, first_name), hire_date,
       NEXT_DAY(ADD_MONTHS(hire_date, 6), 'monday') "Negociere"
FROM employees;
```

13. Să se afișeze numele angajaților și comisionul.

Dacă un angajat nu câștigă comision, să se scrie “Fara comision”.

Etichetați coloana “Comision”.

!!! NVL(EXPR1, EXPR2) - dacă expr1 este NULL, returnează expr2

- altfel returnează expr1
- tipurile celor două expresii trebuie să fie compatibile
- sau expr2 să se convertească implicit la expr1
- în caz contrar facem o conversie explicită

```
SELECT last_name, NVL(commission_pct, 'Fara comision') "Comision"
```

```
FROM employees; -- eroare deoarece cele două expresii din NVL nu au același tip
```

-- VARIANTA CORECTĂ - CONVERSIE EXPLICITĂ

```
SELECT last_name, NVL(TO_CHAR(commission_pct), 'Fara comision') "Comision"
```

```
FROM employees;
```

-- SAU

```
SELECT last_name,
```

```
    DECODE(commission_pct, NULL, 'Fara comision', commission_pct) "Comision"
```

```
FROM employees;
```

--SAU

SELECT last_name,

CASE WHEN commission_pct IS NULL THEN 'Fara comision'

ELSE TO_CHAR(commission_pct)

END"Comision"

FROM employees;

14. Să se listeze numele, salariul și comisionul tuturor angajaților al căror venit lunar (salariu + valoare comision) depășește 10 000.

SELECT last_name, salary, commission_pct

FROM employees

WHERE salary + salary * commission_pct > 10000;

-- varianta gresita deoarece comisionul poate fi si null

-- in acest caz expresia salary + salary * commission_pct este NULL

SELECT 10 + 0.5 * 10

FROM DUAL;

SELECT 10 + NULL * 10

FROM DUAL; -- operatiile matematice cu NULL sunt = cu NULL

-- VARIANTA CORECTA

```
SELECT last_name, salary, commission_pct,  
       salary + salary * NVL(commission_pct, 0) "Venit lunar"  
FROM employees  
WHERE salary + salary * NVL(commission_pct, 0) > 10000;
```

-- SAU

```
SELECT last_name, salary, commission_pct,  
       NVL(salary + salary * commission_pct, SALARY) "Venit lunar"  
FROM employees  
WHERE NVL(salary + salary * commission_pct, SALARY) > 10000;
```

-- SORTARE

```
SELECT last_name, salary, commission_pct,  
       NVL(salary + salary * commission_pct, SALARY) "Venit lunar"  
FROM employees  
WHERE NVL(salary + salary * commission_pct, SALARY) > 10000  
ORDER BY COMMISSION_PCT NULLS FIRST; -- sortare crescatoare (implicit)
```

- valorile null sunt plasate la final
- se poate utiliza NULLS FIRST pt plasarea valorilor null la inceput
- exista si NULLS LAST

15. Să se afișeze numele, codul funcției, salariul și o coloana 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.

Să se denumească coloana "Salariu renegociat".

--CASE VARIANTA 1:

```
SELECT last_name, job_id, salary, job_id,  
       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 renegociat"  
FROM employees;
```

-- CASE VARIANTA 2:

```
SELECT last_name, job_id, salary,  
  
CASE WHEN job_id = 'TT_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 renegociat"  
  
FROM employees;
```

-- DECODE:

```
SELECT last_name, job_id, salary,  
  
      DECODE(job_id, 'TT_PROG', salary * 1.1,  
              'ST_CLERK', salary * 1.15,  
              'SA_REP', salary * 1.2,  
              salary  
              ) "Salariu renegociat"  
  
FROM employees;
```



```
SELECT last_name,  
  
      DECODE(commission_pct, NULL, 'Fara comision', commission_pct) "Comision"  
  
FROM employees;
```



```
SELECT last_name,  
       CASE WHEN commission_pct IS NULL THEN 'Fara comision'  
            ELSE TO_CHAR(commission_pct)  
            END "Comision"  
FROM employees;
```

-- JOIN

```
SELECT * FROM EMPLOYEES;  
SELECT * FROM DEPARTMENTS;
```

```
SELECT employee_id, last_name, department_name  
FROM employees, departments; -- produs cartezian
```

-- VARIANTA 1 - JOIN IN WHERE

```
SELECT employee_id, last_name, department_name, d.department_id  
FROM employees e, departments d  
WHERE e.department_id = d.department_id; -- 106 angajati din totalul de 107
```

-- VARIANTA 2 - JOIN IN FROM FOLOSIND ON

```
SELECT employee_id, last_name, department_name, d.department_id
FROM employees e JOIN departments d
    ON (e.department_id = d.department_id);
```

-- JOIN IN FROM FOLOSIND USING

```
SELECT employee_id, last_name, department_name, department_id
FROM employees JOIN departments
    USING(department_id);
```

-- 106 angajati din totalul de 107

DE CE?

EXISTA UN ANGAJAT CARE NU ARE DEPARTAMENT -> ARE NULL PE
COLOANA DEPARTMENT_ID DIN EMPLOYEES

-- CUM PUTEM SA AFISAM SI ACEST ANGAJAT?

-- SE FOLOSESTE SIMBOLUL (+) IN PARTEA DEFICITARA DE INFORMATIE

-- AFISAM SI ANGAJATII FARA DEPARTAMENT

```
SELECT employee_id, last_name, department_name, d.department_id
FROM employees e, departments d
WHERE e.department_id = d.department_id (+);
```

-- DEPARTAMENTE FARA ANGAJATI

```
SELECT employee_id, last_name, department_name, d.department_id
FROM employees e, departments d
WHERE e.department_id (+) = d.department_id;
```

17. Să se listeze codurile și denumirile job-urilor care există în departamentul 30.

```
SELECT e.job_id, job_title, department_id
FROM jobs j, employees e
WHERE department_id = 30 and j.job_id = e.job_id;
```