Baze de Date
Solutii – Laborator 2 – Laborator 3 – Saptamana 5

--LABORATOR 2 - SAPTAMANA 5

22. Scrieţi o cerere care afişează numele angajatului, codul departamentului în care acesta lucrează şi numele colegilor săi de departament.
Se vor eticheta coloanele corespunzător.

```
SELECT ang.employee_id, ang.last_name "Nume angajat", ang.department_id,
      coleg.last_name "Nume coleg", coleg.employee_id
FROM employees ang, employees coleg
WHERE ang.department_id = coleg.department_id
     and ang.employee_id > coleg.employee_id;
```

23. Creaţi o cerere prin care să se afişeze numele angajatilor, codul job-ului, titlul job-ului, numele departamentului şi salariul angajaţilor.
Se vor include și angajații al căror departament nu este cunoscut.

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

24. Să se afişeze numele şi data angajării pentru salariaţii care au fost angajaţi după Gates.

```
SELECT ang.last_name NumeAng, ang.hire_date DataAng,
       gates.last_name NumeGates, gates.hire_date DataGates
FROM employees ang, employees gates
WHERE ang.hire_date > gates.hire_date
      and initcap(gates.last_name) = 'Gates';

-- SAU UTILIZAND SUBCERERE IN WHERE
SELECT ang.last_name NumeAng, ang.hire_date DataAng
FROM employees ang
WHERE ang.hire_date > (SELECT hire_date
                       FROM employees
                       WHERE initcap(last_name) = 'Gates'
                       );
```

25. Să se afişeze numele salariatului şi data angajării împreună cu numele şi data angajării şefului direct pentru salariaţii care au fost angajaţi înaintea şefilor lor.

Se vor eticheta coloanele Angajat, Data_ang, Manager si Data_mgr.

```
SELECT ang.last_name Angajat, ang.hire_date Data_Ang,
       m.last_name Manager, m.hire_date Data_mgr
```

```
FROM employees ang, employees m
WHERE ang.manager_id = m.employee_id
    AND ang.hire_date < m.hire_date;

SELECT * FROM EMPLOYEES;

EMPLOYEES emp -> last_name, hire_date
employee_id, manager_id

EMPLOYEES mgr -> last_name, hire_date
employee_id
```

-- LABORATOR 3

-- RECAPITULARE JOIN

```
-- Join-ul este operaţia de regăsire a datelor din două sau mai multe tabele,
-- pe baza valorilor comune ale unor coloane. De obicei, aceste coloane reprezintă
-- cheia primară, respectiv cheia externă a tabelelor.
-- Reamintim că pentru a realiza un join între n tabele
-- o sa fie nevoie de cel puţin n – 1 condiţii de join
```

--TIPURI DE JOIN:

-- NONEQUIJOIN – condiţia de join conţine alţi operatori decât operatorul de egalitate

--Exemplu Nonequijoin:

```
SELECT last_name, salary, grade_level, lowest_sal, highest_sal
FROM employees, job_grades
WHERE salary BETWEEN lowest_sal AND highest_sal;

SELECT * FROM job_grades;
```

-- INNER JOIN (equijoin, join simplu)

-- corespunde situaţiei în care valorile de pe coloanele ce apar în condiţia

-- de join trebuie să fie egale

--EXEMPLE (folosind atat join-ul in WHERE cat si cel din standardul SQL3):

-- VARIANTA 1 - Condiția de Join este scrisă în clauza WHERE a instrucțiunii SELECT

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

```
SELECT DISTINCT last_name, department_id
FROM employees;
```

DISTINCT - este in standard

UNIQUE - nu este in standard

```
-- VARIANTA 2 - --JOIN SCRIS IN FROM (standardul SQL3) - 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 SCRIS IN FROM (standardul SQL3) - folosind USING
-- USING SE UTILIZEAZA dacă există coloane având acelasi nume
-- in acest caz coloanele referite nu trebuie sa contina calificatori
-- adica sa nu fie precedate de nume de tabele sau alias-uri

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

```
-- Cele doua variante (join in where si join in from) sunt echivalente.
```

-- OUTER JOIN

-- pentru a afisa si angajatii care nu au departament se utilizeaza

-- simbolul (+) in partea deficitara de informatie

-- deficit de informatie (FARA DEPARTAMENT) -> afisam angajatii care au departament dar si angajatii FARA departament

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

-- deficit de informatie (FARA ANGAJATI) -> afisam departamentele care au angajati dar si departamente FARA angajati

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

-- In cazul standardului SQL3 se utilizeaza LEFT, RIGHT şi FULL OUTER JOIN

-- DISCUTIE!

-- DORIM SA AFISAM SI ANGAJATII CARE NU AU DEPARTAMENT

-- ADICA AFISAM TOTI ANGAJATII INDIFERENT DACA AU SAU NU DEPARTAMENT

-- CEEA CE INSEAMNA CA AFISAM ATAT ANGAJATII CARE AU DEPARTAMENT (INTERSECTIA)

-- CAT SI ANGAJATII CARE NU AU DEPARTAMENT

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

-- DORIM SA AFISAM SI DEPARTAMENTELE CARE NU AU ANGAJATI

-- ADICA AFISAM TOATE DEPARTAMENTELE INDIFERENT DACA AU SAU NU ANGAJATI

-- CEEA CE INSEAMNA CA AFISAM ATAT DEPARTAMENTELE CARE AU ANGAJATI (INTERSECTIA)

-- CAT SI DEPARTAMENTELE CARE NU AU ANGAJATI

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

```
-- FULL JOIN

-- AFISAM ATAT ANGAJATII CARE AU DEPARTAMENTE (INTERSECTIA)

-- CAT SI ANGAJATII CARE NU AU DEPARTAMENT

-- IMPREUNA CU DEPARTAMENTELE CARE NU AU ANGAJATI


SELECT employee_id, last_name, d.department_id, department_name

FROM employees e FULL JOIN departments d ON (e.department_id = d.department_id);



-- CROSS JOIN - produs cartezian

SELECT employee_id, last_name, e.department_id, department_name

FROM employees e CROSS JOIN departments d;



-- NATURAL JOIN

SELECT last_name, job_id, job_title

FROM employees NATURAL JOIN jobs;


SELECT last_name, e.job_id, job_title

FROM employees e, jobs j

WHERE e.job_id = j.job_id;
```

2. Să se afișeze codul şi numele angajaţilor care lucrează în același departament

cu cel puţin un angajat al cărui nume conţine litera “t”. Se vor afişa, de asemenea,

codul şi numele departamentului respectiv. Rezultatul va fi ordonat alfabetic după nume.

```
SELECT ang.employee_id, ang.last_name,
      coleg.employee_id, department_name
FROM employees ang JOIN employees coleg ON (ang.department_id =
coleg.department_id)
            JOIN departments d ON (ang.employee_id = d.department_id)
WHERE lower(coleg.last_name) LIKE '%t%'
     AND (ang.employee_id != coleg.employee_id)
ORDER BY ang.last_name;
```

3. Să se afișeze numele, salariul, titlul job-ului, oraşul şi ţara

în care lucrează angajații conduși direct de King.

```
SELECT e.last_name, e.salary, job_title, country_name, city
FROM employees e JOIN employees k ON (e.manager_id = k.employee_id)
            JOIN jobs j ON (e.job_id = j.job_id)
            JOIN departments d ON (e.department_id = d.department_id)
            JOIN locations l ON (d.location_id = l.location_id)
```

```
        JOIN countries c ON (l.country_id = c.country_id)
WHERE LOWER(k.last_name) LIKE 'king';
```

--5

-- FULL OUTER JOIN

SELECT employee_id, last_name, d.department_id, department_name

FROM employees e FULL JOIN departments d ON (e.department_id = d.department_id);

-- FULL JOIN = 123 = angajatii care lucreaza in departamente (106 angajati)

-- + angajatii care nu au depart (1 angajat)

-- + departamentele fara angajati (16 departamente)

-- SAU CU OPERATORI PE MULTIMI

-- left join afiseaza toti angajatii indiferent daca au sau nu departament

-- 106 (angajati care lucreaza in departamente - intersectia)

-- + 1 angajat care nu are departament (acesta se afla exclusiv in tabelul employees)

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

UNION -- ELEMENTELE COMUNE SI NECOMUNE O SINGURA DATA

-- right join afiseaza toate departamentele indiferent daca au sau nu angajati
-- 106 angajati care lucreaza in departamente, deci departamentele au implicit
angajati(INTERSECTIA)
-- + 16 departamente care nu au angajati (acestea se afla exclusiv in tabelul
departamente)

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