

Baze de Date

Solutii – Laborator 6 – Saptamana 9

LABORATOR 6 - SAPTAMANA 9

Subcereri Necorelate

1. Folosind subcereri, să se afișeze numele și data angajării pentru salariații care au fost angajați după Gates.

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date > (SELECT hire_date
                   FROM employees
                   WHERE INITCAP(last_name)= 'Gates'
                   );
```

2. Folosind subcereri, scrieți o cerere pentru a afișa numele și salariul pentru toți colegii (din același departament) lui Gates. Se va exclude Gates.

-- SUBCERERE PENTRU A PRELUA DEPARTAMENTUL LUI GATES

```
select department_id
from employees
where initcap(last_name) = 'Gates';
```

--CEREREA PRINCIPALA

```
select last_name, salary, department_id
from employees
where initcap(last_name) != 'Gates'
    and department_id = (select department_id
                        from employees
                        where initcap(last_name) = 'Gates'
                        )
```

--Se poate înlocui operatorul IN cu = ???

```
select last_name, salary, department_id
from employees
where initcap(last_name) != 'Gates'
    and department_id IN (select department_id
```

```
from employees  
where initcap(last_name) = 'Gates'  
);
```

--Se va inlocui Gates cu King;

```
select last_name, salary, department_id  
from employees  
where initcap(last_name) != 'King'  
and department_id = (select department_id  
from employees  
where initcap(last_name) = 'King'  
); -- "single-row subquery returns more than one row"
```

```
select last_name, salary, department_id  
from employees  
where initcap(last_name) != 'King'  
and department_id IN (select department_id  
from employees  
where initcap(last_name) = 'King'  
);
```

3. Folosind subcereri, să se afișeze numele și salariul angajaților conduși direct

de președintele companiei (acesta este considerat angajatul care nu are manager).

-- REZOLVATI INDIVIDUAL

-- CEREREA TREBUIE SA RETURNEZE 14 ANGAJATI

-- VEZI IMAGINEA ATASATA IN LABORATOR

```
SELECT last_name, salary
FROM employees
WHERE manager_id = (SELECT employee_id
                     FROM employees
                     WHERE manager_id IS NULL
                     );
```

4. Scrieți o cerere pentru a afișa numele, codul departamentului și salariul angajaților

al căror cod de departament și salariu coincid cu codul departamentului și salariul

unui angajat care câștigă comision.

```
SELECT last_name, department_id, salary
FROM employees
WHERE (department_id, salary) = (SELECT department_id, salary
                                FROM employees
                                WHERE commission_pct is not null
                                );
```

5. Să se afișeze codul, numele și salariul tuturor angajaților al căror salariu este mai mare decât salariul mediu.

```
SELECT employee_id, last_name, salary
FROM employees
WHERE salary > (SELECT ROUND(AVG(salary))
                FROM employees
                );
```

6. Scrieti o cerere pentru a afișa angajații care câștigă (castiga = salariul plus comision din salariu) mai mult decât oricare funcționar (job-ul conține șirul "CLERK").
Sortați rezultatele după salariu, în ordine descrescătoare;

--subcererea - selectam venitul functionarilor (job-ul contine şirul "CLERK")

```
select salary + salary * nvl(commission_pct, 0)
```

```
from employees
```

```
where upper(job_id) like '%CLERK%';
```

-- CEREREA COMPLETA

```
select employee_id, last_name, salary, commission_pct
```

```
from employees
```

```
where salary + salary * nvl(commission_pct, 0) > ALL
```

```
  (select salary + salary * nvl(commission_pct, 0)
```

```
   from employees
```

```
  where upper(job_id) like '%CLERK%'
```

```
);
```

--SAU

```
select employee_id, last_name, salary, commission_pct
```

```
from employees
```

```
where salary + salary * nvl(commission_pct, 0) >
```

```
  (select max(salary + salary * nvl(commission_pct, 0))
```

```
   from employees
```

```
  where upper(job_id) like '%CLERK%'
```

```
);
```

> ALL -> mai mare decat toate valorile => mai mare decat maximul

> ANY -> mai mare decat cel putin o valoare => mai mare decat minimul

7. Scrieți o cerere pentru a afișa numele angajaților, numele departamentului și salariul angajaților care câștigă comision, dar al căror șef direct nu câștigă comision.

-- REZOLVATI IN ECHIPA DE 2 PERSOANE

-- CEREREA TREBUIE SA RETURNEZE 5 ANGAJATI

-- VEZI IMAGINEA ATASATA IN LABORATOR

```
SELECT last_name, department_name, salary, commission_pct, e.manager_id
FROM employees e JOIN departments d ON (e.department_id =
d.department_id)
WHERE commission_pct IS NOT NULL AND
      e.manager_id IN (SELECT employee_id
                        FROM employees
                        WHERE commission_pct IS NULL
                        );
```

8. Să se afișeze numele angajaților, codul departamentului și codul job-ului salariaților

al căror departament se află în Toronto.

-- REZOLVATI IN ECHIPA DE 2 PERSOANE

-- CEREREA TREBUIE SA RETURNEZE 2 ANGAJATI

-- VEZI IMAGINEA ATASATA IN LABORATOR

```
SELECT last_name, department_id, job_id
FROM employees
WHERE department_id = (SELECT department_id
                        FROM departments d JOIN locations l ON (d.location_id =
l.location_id)
                        WHERE initcap(city) = 'Toronto'
                        );
```


--SAU:

SELECT last_name, department_id, job_id

FROM employees

WHERE department_id in (

 SELECT department_id

 FROM departments

 WHERE location_id in

 (SELECT location_id

 FROM locations

 WHERE lower(city) = 'toronto'

)

);

-- SAU

SELECT last_name, e.department_id, job_id

FROM employees e JOIN departments d

ON(e.department_id=d.department_id)

 JOIN locations l ON (d.location_id=l.location_id)

WHERE l.city = 'Toronto';

9. Să se obțină codurile departamentelor în care nu lucreaza nimeni (nu este introdus nici un salariat în tabelul employees). Sa se utilizeze subcereri;

-- OPERATORI PE MULTIMI

```
select department_id
from departments -- din lista tuturor departamentelor
```

minus -- eliminam

```
select department_id
from employees -- departamentele cu angajati
```

-- => departamentele care nu au angajati

--SUBCERERE

```
select department_id
from departments
where department_id NOT IN (select department_id
                             from employees
                             where department_id is not null
                             );
```

-- SAU

```
select department_id
from departments
where department_id NOT IN (select NVL(department_id, -1)
                             from employees
                             );
```

!!!

ATENTIE LA NOT IN SI NULL

-- SAU PUTEM UTILIZA IN

```
select department_id
from departments
where department_id IN
    (select department_id
     from departments
     minus
```

```
    select department_id
    from employees
    ); -- subcererea returneaza departamentele fara angajati
```

10. Este posibilă introducerea de înregistrări prin intermediul subcererilor (specificate în locul tabelului).

Ce reprezintă, de fapt, aceste subcereri? Să se analizeze următoarele comenzi INSERT:

```
INSERT INTO emp (employee_id, last_name, email, hire_date, job_id, salary,  
commission_pct)
```

```
VALUES (252, 'Nume252', 'nume252@emp.com', SYSDATE, 'SA_REP',  
5000, NULL);
```

```
SELECT employee_id, last_name, email, hire_date, job_id, salary,  
commission_pct
```

```
FROM emp
```

```
WHERE employee_id = 252;
```

```
ROLLBACK;
```

```
INSERT INTO
```

```
(SELECT employee_id, last_name, email, hire_date, job_id, salary,  
commission_pct
```

```
FROM emp)
```

```
VALUES (252, 'Nume252', 'nume252@emp.com', SYSDATE, 'SA_REP',  
5000, NULL);
```

```
SELECT employee_id, last_name, email, hire_date, job_id, salary,  
commission_pct
```

```
FROM emp
```

```
WHERE employee_id = 252;
```

```
ROLLBACK;
```

11. Sa se creeze tabelul SUBALTERNI care sa contina codul, numele si prenumele angajatilor

care il au manager pe Steven King, alaturi de codul si numele lui King.

Coloanele se vor numi cod, nume, prenume, cod_manager, nume_manager.

```
DESC employees;
```

```
CREATE TABLE SUBALTERNI
```

```
(cod number(6) constraint pkey_sub primary key,
```

```
nume varchar2(25) constraint nume_sub not null,
```

```
prenume varchar2(20),
```

```
cod_manager number(6),
```

```
nume_manager varchar2(25) constraint nume_man not null
```

```
);
```

```
INSERT INTO SUBALTERNI (cod, nume, prenume, cod_manager,  
nume_manager)
```

```
(SELECT ang.employee_id, ang.last_name, ang.first_name,  
ang.manager_id,
```

```
man.last_name
```

```
FROM employees ang join employees man on (ang.manager_id =  
man.employee_id)
```

```
WHERE ang.manager_id = (select employee_id
```

```
from employees
```

```
where lower(last_name||first_name) = 'kingsteven'
```

```
)
```

```
);
```

```
commit;
```

```
select * from subalterni;
```