Ex 1: Folosind clauza with, pentru fiecare departament afisati informatii despre angajatii cu cel mai mare salariu din acel departament.

Daca intr-un departament nu lucreaza nimeni, veti afisa un mesaj corespunzator (diferit de "nu lucreaza nimeni").

with salarii\_departamente as(

select d.department\_id, d.department\_name, e.last\_name, e.salary as salariu\_maxim, count(\*) as nr\_ang\_salariu\_maximn

from departments d

join employees e on(e.department\_id = d.department\_id)

group by d.department\_id, d.department\_name, e.salary, e.last\_name -- daca intr-un departament ar exista mai mult de un angajat cu salariu maxim, s-ar afisa mai mult de un rezultat pentru acelasi department\_id (nu e cazul aici)

having e.salary = (

select max(e2.salary)

from employees e2

where e2.department\_id = d.department\_id

)

order by salariu\_maxim desc

)

select department\_id, department\_name, last\_name, salariu\_maxim, nr\_ang\_salariu\_maximn

from salarii\_departamente

union all -- fara union all si cu o incercare de join pt departments nu s-ar fi afisat decat niste departamente in care lucreaza cel putin un angajat, motiv pentru care nvl si decode nu ar fi mers

select d.department\_id, d.department\_name, 'Angajat inexistent momentan', 0, 0 -- determinarea departementelor in care nu lucreaza niciun angajat

from departments d -- doar pt corelare

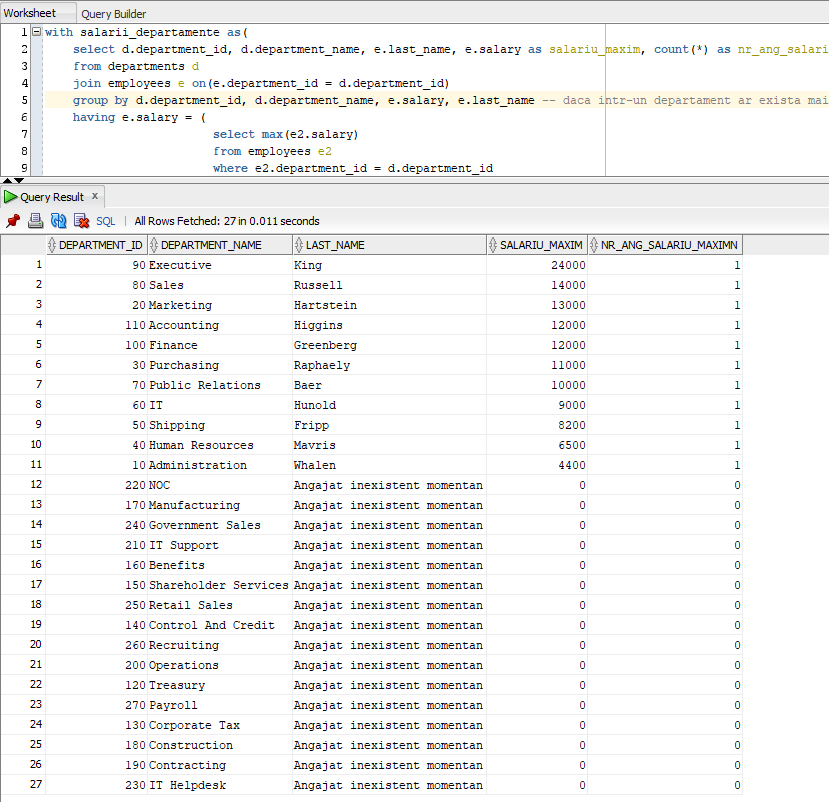
where not exists (

select 1

from employees e

where e.department\_id = d.department\_id

);



Ex 2: Afisati codul si numele departamentului si numele complet al managerului departamentelor pentru managerii de departament avand salariul mai mare decat media salariilor in care lucreaza angajatii al caror sef direct are un numar de telefon ce contine sirul "67".

select d.manager\_id, d.department\_name, e.first\_name || ' ' || e.last\_name as nume\_complet -- manageri departamente

from departments d

join employees e on(e.employee\_id = d.manager\_id)

where --d.manager\_id is not null -- exista departamente care nu sunt administrate, nu e o conditie necesara in acest caz

e.salary > (

select avg(e2.salary)

from employees e2

where e2.manager\_id in(

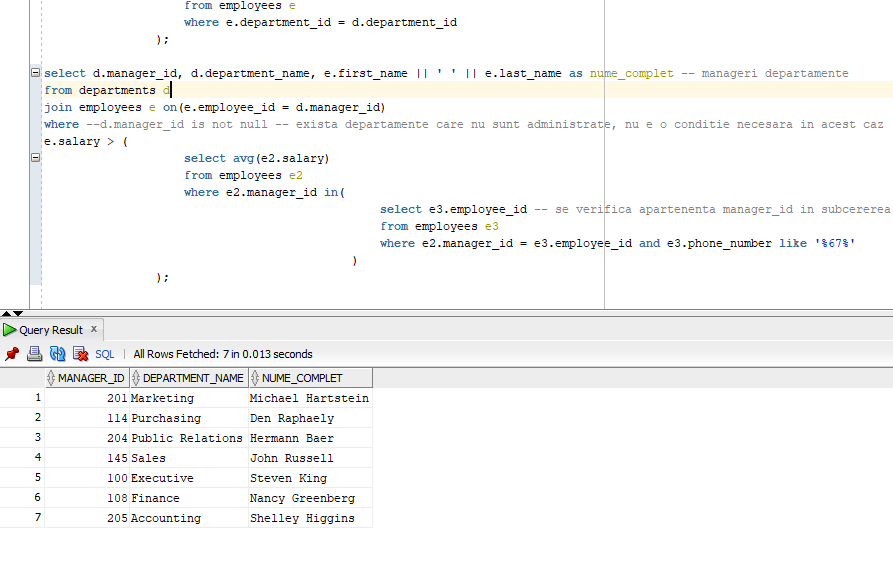
select e3.employee\_id -- se verifica apartenenta manager\_id in subcererea corelata pt a determina un average de salariu pentru subordonati

from employees e3

where e2.manager\_id = e3.employee\_id and e3.phone\_number like '%67%'

)

);



Ex 3: Afisati departamentele in care lucreza angajati al caror manager direct are un numar de telefon ce contine fix 10 CIFRE. (Atentie la forma actuala a numerelor de telefon din baza de date. Folositi doar functiile prezente in fisierele oferite ca suport la laborator.) Se vor afisa doar departamentele a caror medie salariala este mai mare decat media salariala a intregii firme.

select d.department\_id, d.department\_name

from departments d

join employees e on(d.department\_id = e.department\_id) -- doar pentru a afisa numele departamentului

where e.manager\_id in ( -- corelare sef direct

select e2.employee\_id

from employees e2

where e.manager\_id = e2.employee\_id and length(replace(e.phone\_number, '.', '')) = 10

)

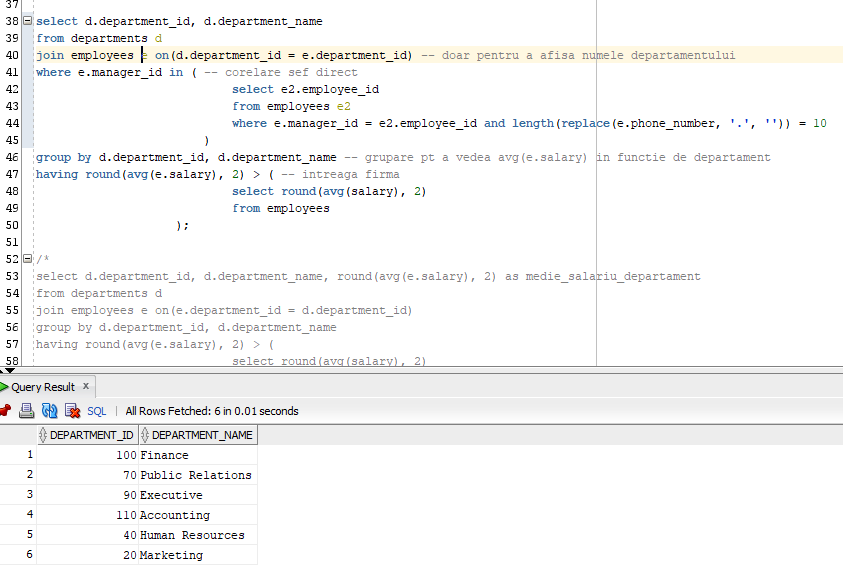
group by d.department\_id, d.department\_name -- grupare pt a vedea avg(e.salary) in functie de departament

having round(avg(e.salary), 2) > ( -- intreaga firma

select round(avg(salary), 2)

from employees

);



Ex 4: Sa se listeze angajatii (codul si numele acestora) care au lucrat pe toate proiectele nelivrate la termen.

-- sapt 12 division lab

select distinct a.employee\_id

from works\_on a

where not exists (

(

select p2.project\_id -- toate proiectele pe care lucreaza angajatul (in mod implicit daca lucreaza si la proiectul 2, inseamna ca diferenta dintre multimi nu e vida, deci nu respecta criteriul)

from project p2

join works\_on b on(b.project\_id = p2.project\_id)

where b.employee\_id = a.employee\_id

)

minus

( -- proiecte nelivrate la deadline per angajat dar si overall (delivery date e stabilita ca fiind data cei mai inaintata la care un angajat termina de lucrat la proiect)

select p.project\_id

from project p

where a.end\_date > p.deadline and p.delivery\_date > p.deadline

)

);

