select j.job\_title as Titlu\_Job\_PM, j.min\_salary as Salariu\_Min\_PM, j.max\_salary as Salariu\_Max\_PM,

e.last\_name || ' ' || e.first\_name as Nume\_PM, e.employee\_id as ID\_PM, d.department\_name as Departament\_PM

from jobs j left join employees e on j.job\_id=e.job\_id

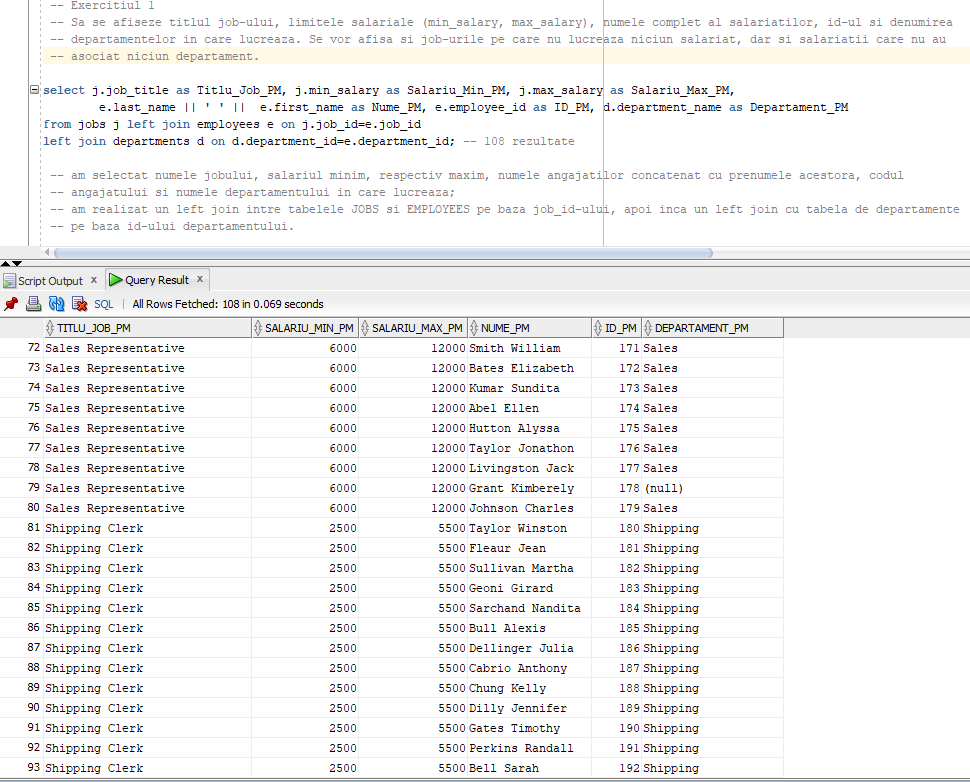
left join departments d on d.department\_id=e.department\_id;

-- am selectat numele jobului, salariul minim, respectiv maxim, numele angajatilor concatenat cu

-- prenumele acestora, codul angajatului si numele departamentului in care lucreaza;

-- am realizat un left join intre tabelele JOBS si EMPLOYEES pe baza job\_id-ului, apoi inca un left join

-- cu tabela de departamente pe baza id-ului departamentului.



select 'Departamentul ' || d.department\_name || ' este condus de ' || decode(d.manager\_id, null, 'nimeni', d.manager\_id) || ' si '

|| decode(count(e.employee\_id), 0, 'nu are salariati.', 'are salariati.') as Informatii\_PM

from employees e right join departments d on d.department\_id=e.department\_id

group by d.department\_id, d.department\_name, d.manager\_id;

-- rezultatul a fost afisat in formatul cerut prin concatenare;

-- pentru verificarea manager\_id-ului am folosit functia DECODE care returneaza string-ul "nimeni"

-- atunci cand manager\_id este null si valoarea acestuia in caz contrar;

-- am folosit functiile DECODE si COUNT pentru a verifica daca exista salariati;

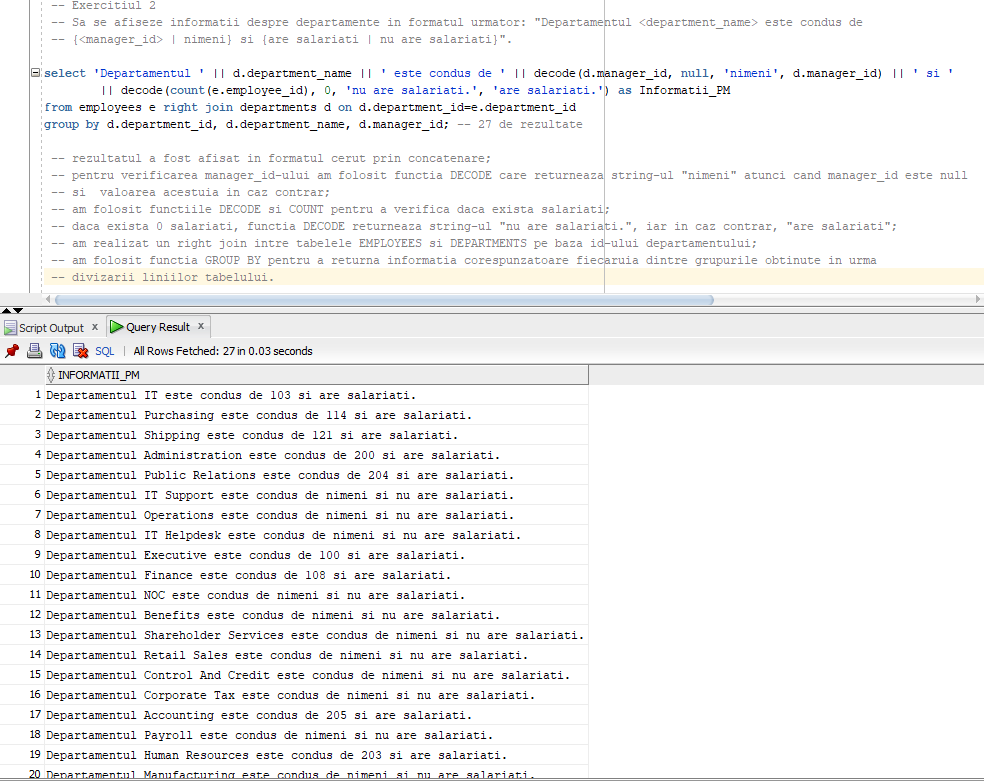
-- daca exista 0 salariati, functia DECODE returneaza string-ul "nu are salariati.", iar in caz contrar, "are

-- salariati";

-- am realizat un right join intre tabelele EMPLOYEES si DEPARTMENTS pe baza id-ului departamentului;

-- am folosit functia GROUP BY pentru a returna informatia corespunzatoare fiecaruia dintre grupurile

-- obtinute in urma divizarii liniilor tabelului.



select e.last\_name as Nume\_PM, e.first\_name as Prenume\_PM, length(e.last\_name) as Lungime\_Nume\_PM, l.city as Oras\_PM

from locations l right join departments d on l.location\_id=d.location\_id

right join employees e on d.department\_id=e.department\_id

where nullif(length(e.last\_name), length(e.first\_name)) is not null;

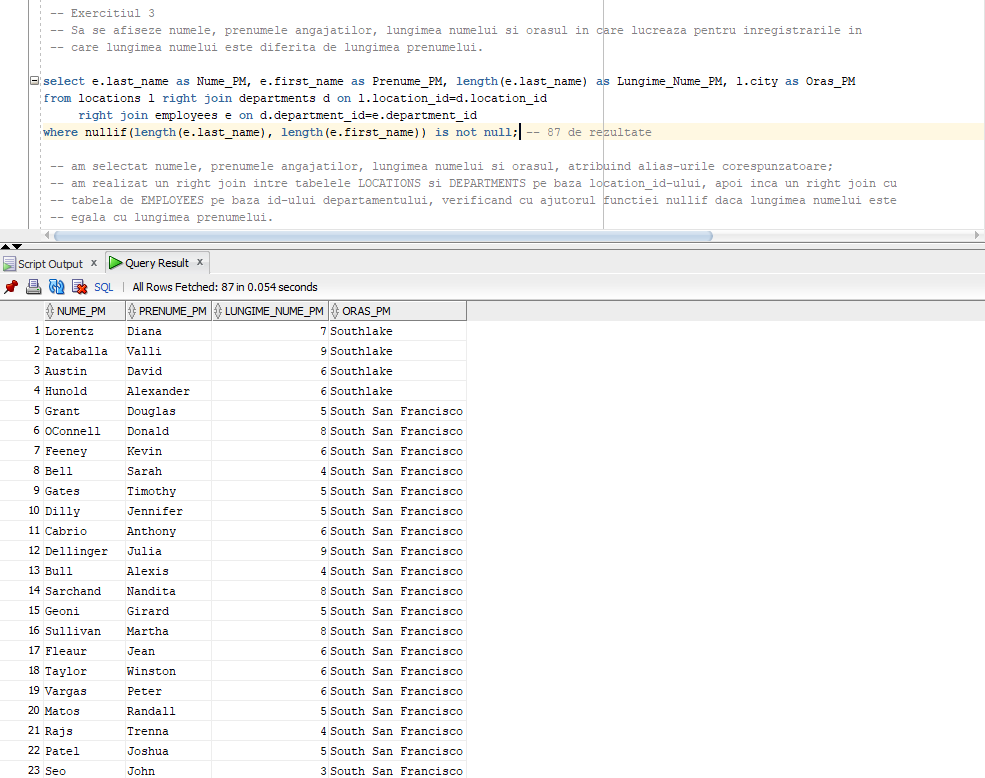
-- am selectat numele, prenumele angajatilor, lungimea numelui si orasul, atribuind alias-urile

-- corespunzatoare;

-- am realizat un right join intre tabelele LOCATIONS si DEPARTMENTS pe baza location\_id-ului, apoi

-- inca un right join cu tabela de EMPLOYEES pe baza id-ului departamentului, verificand cu ajutorul

-- functiei nullif daca lungimea numelui este egala cu lungimea prenumelui.



select e.last\_name as Nume\_PM, e.hire\_date as Data\_Angajarii\_PM, j.job\_title as Titlul\_Jobului\_PM, e.salary as Salariu\_PM,

decode(extract(year from e.hire\_date), 1989, salary \* 1.2, 1990, salary \* 1.15, 1991, salary \* 1.1, salary) as Marire\_PM

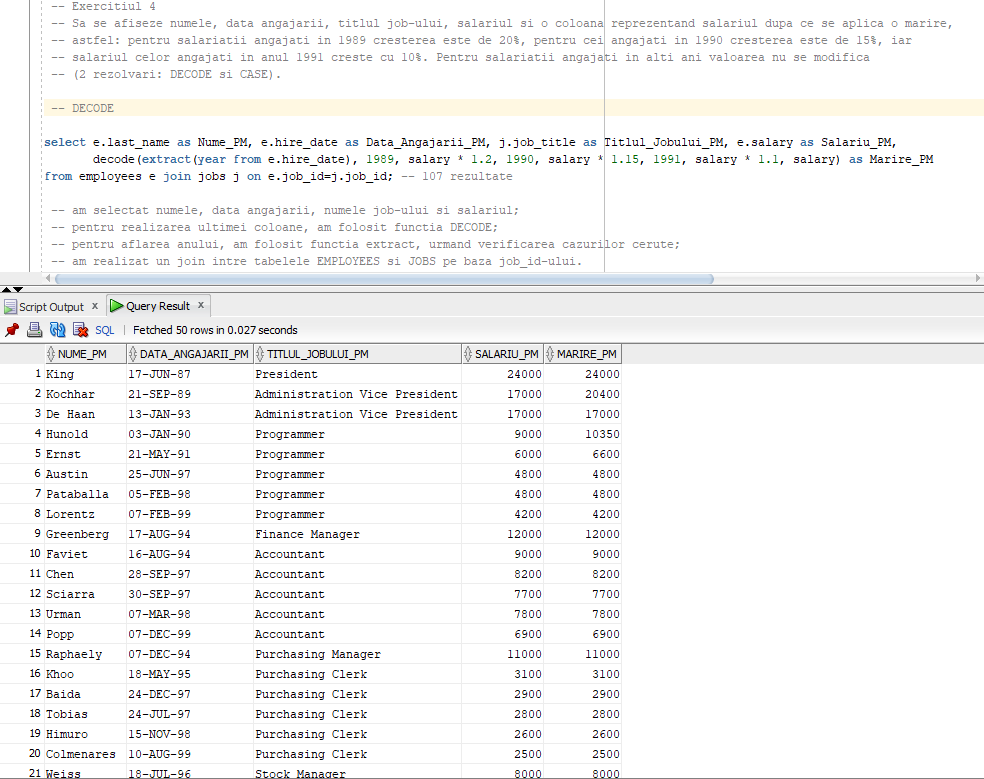
from employees e join jobs j on e.job\_id=j.job\_id;

-- am selectat numele, data angajarii, numele job-ului si salariul;

-- pentru realizarea ultimei coloane, am folosit functia DECODE;

-- pentru aflarea anului, am folosit functia extract, urmand verificarea cazurilor cerute;

-- am realizat un join intre tabelele EMPLOYEES si JOBS pe baza job\_id-ului.



select e.last\_name as Nume\_PM, e.hire\_date as Data\_Angajarii\_PM, j.job\_title as Titlul\_Jobului\_PM, e.salary as Salariu\_PM,

case extract(year from e.hire\_date)

when 1989 then salary \* 1.2

when 1990 then salary \* 1.15

when 1991 then salary \* 1.1

else salary

end as Marire\_PM

from employees e join jobs j on e.job\_id=j.job\_id;

-- acest cod difera de cel precedent doar prin folosirea functiei CASE in loc de DECODE cu ajutorul careia

-- am verificat cazurile cerute.

