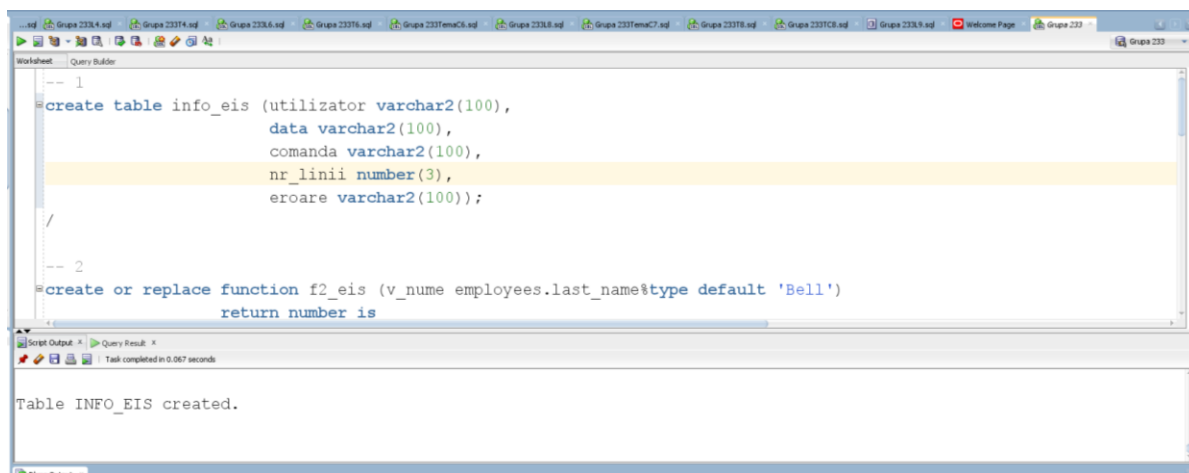


TEMA LABORATOR 9  
(Laborator PLSQL 4)

1. Creați tabelul info\_eis cu următoarele coloane:

- utilizator (numele utilizatorului care a inițiat o comandă)
- data (data și timpul la care utilizatorul a inițiat comanda)
- comanda (comanda care a fost inițiată de utilizatorul respectiv)
- nr\_linii (numărul de linii selectate/modificate de comandă)
- eroare (un mesaj pentru excepții).

```
create table info_eis (utilizator varchar2(100),  
                      data varchar2(100),  
                      comanda varchar2(100),  
                      nr_linii number(3),  
                      eroare varchar2(100));
```



2. Modificați funcția definită la exercițiul 2, respectiv procedura definită la exercițiul 4 astfel încât să determine inserarea în tabelul info\_eis a informațiilor corespunzătoare fiecărui caz determinat de valoarea dată pentru parametru:

- există un singur angajat cu numele specificat;
- există mai mulți angajați cu numele specificat;
- nu există angajați cu numele specificat.

```
create or replace function f2_eis (v_nume employees.last_name%type default 'Bell')  
    return number is  
    salariu employees.salary%type;  
    counter number;  
begin  
    select count(*) into counter  
    from employees  
    where last_name = v_nume;
```

```

select salary into salariu
from employees
where last_name = v_nume;

```

```

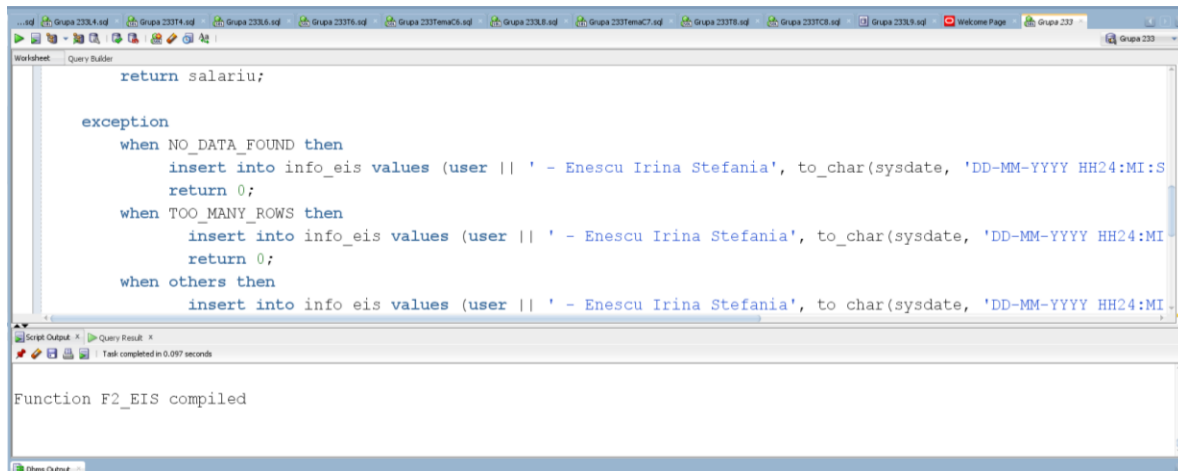
insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, null);
return salariu;

```

```

exception
when NO_DATA_FOUND then
    insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'No data found!');
    return 0;
when TOO_MANY_ROWS then
    insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'Too many rows!');
    return 0;
when others then
    insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'Error!');
    return 0;
end f2_eis;
/

```

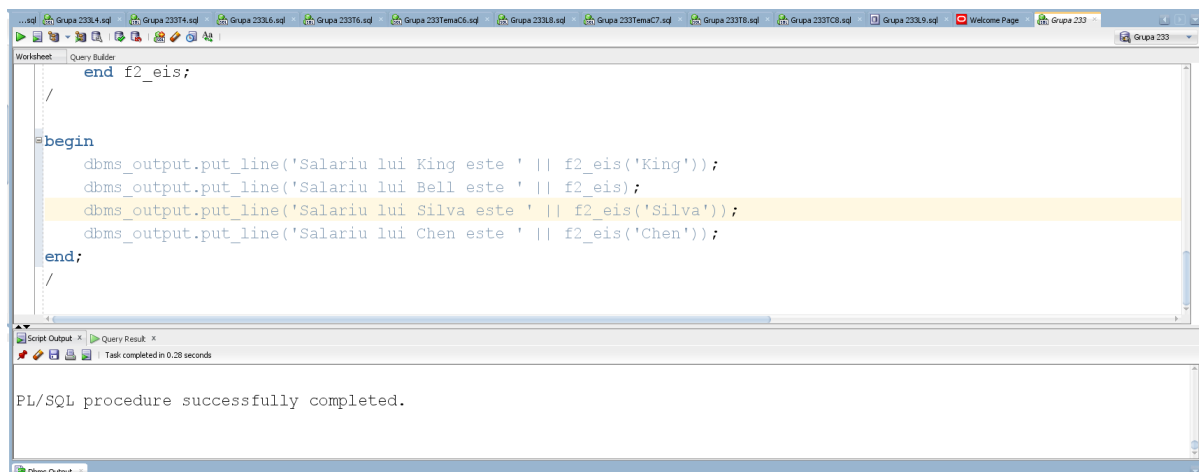


Am testat functia pe fiecare caz:

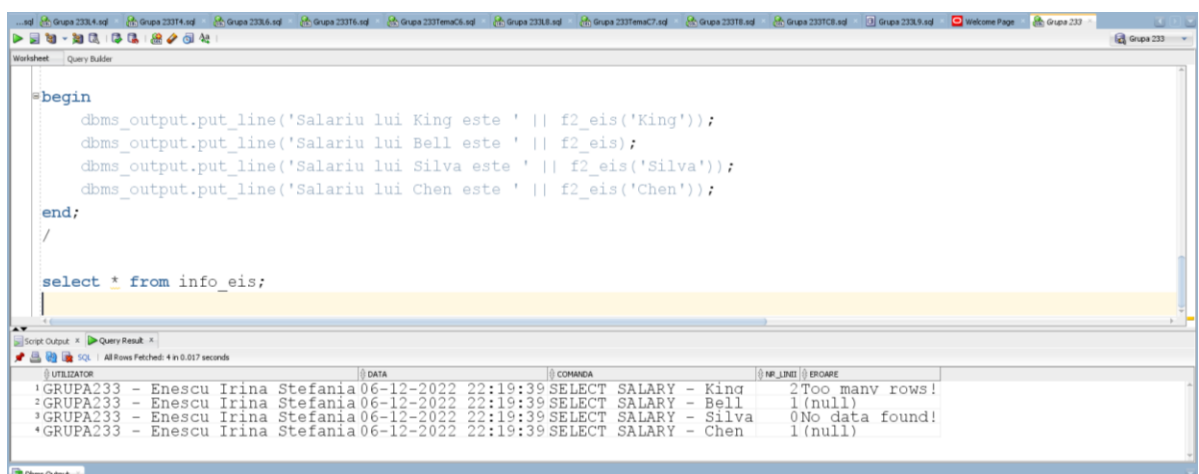
```

begin
    dbms_output.put_line('Salariu lui King este ' || f2_eis('King'));
    dbms_output.put_line('Salariu lui Bell este ' || f2_eis);
    dbms_output.put_line('Salariu lui Silva este ' || f2_eis('Silva'));
    dbms_output.put_line('Salariu lui Chen este ' || f2_eis('Chen'));
end;
/

```



Tabelul arata astfel:



create or replace procedure p4\_eis (v\_nume employees.last\_name%type) is

salariu employees.salary%type;

counter number;

begin

select count(\*) into counter

from employees

where last\_name = v\_nume;

select salary into salariu

from employees

where last\_name = v\_nume;

insert into info\_eis values (user || ' - Enescu Irina Stefania', to\_char(sysdate, 'DD-MM-YYYY HH24:MI:SS'), 'SELECT SALARY - ' || v\_nume, counter, null);

exception

when NO\_DATA\_FOUND then

```

        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'No data found!');
    when TOO_MANY_ROWS then
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'Too many rows!');
    when others then
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'SELECT SALARY - ' || v_nume, counter, 'Error!');

end p4_eis;

```

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL script in the editor is as follows:

```

begin
    select count(*) into counter
    from employees
    where last_name = v_nume;

    select salary into salariu
    from employees
    where last_name = v_nume;

    insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY HH24:MI:SS'),

exception
    when NO_DATA_FOUND then
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY HH24:MI:S

```

Below the editor, the 'Script Output' tab shows the message: 'Task completed in 0.188 seconds' and 'Procedure P4\_EIS compiled'.

Am testat procedura pe fiecare caz:

```

begin
    p4_eis('Santos');
    p4_eis('Grand');
    p4_eis('Smith');
    p4_eis('Olsen');
end;

```

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL script in the editor is:

```

end p4_eis;

/

select last_name, count(*) from employees group by last_name;

begin
    p4_eis('Santos');
    p4_eis('Grand');
    p4_eis('Smith');
    p4_eis('Olsen');
end;

/

```

Below the editor, the 'Script Output' tab shows the message: 'Task completed in 0.097 seconds' and 'PL/SQL procedure successfully completed.'

Tabelul arata astfel:

The screenshot shows a SQL query editor with the following query:

```
select last_name, count(*) from employees group by last_name;
```

Below the query editor, there is a script output window showing the results of the query. The output is as follows:

UTILIZATOR	DATA	COMANDA	MP_LINII	ERORERE
1 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:47:03 SELECT SALARY - Grand	0	No data found!
2 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:47:03 SELECT SALARY - Smith	2	Too many rows!
3 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:47:03 SELECT SALARY - Olsen	1	(null)
4 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:19:39 SELECT SALARY - King	2	Too many rows!
5 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:19:39 SELECT SALARY - Bell	1	(null)
6 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:19:39 SELECT SALARY - Silva	0	No data found!
7 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:19:39 SELECT SALARY - Chen	1	(null)
8 GRUPEA233 - Enescu Irina Stefania	06-12-2022	22:47:03 SELECT SALARY - Santos	0	No data found!

3. Definiți o funcție stocată care determină numărul de angajați care au avut cel puțin 2 joburi diferite și care în prezent lucrează într-un oraș dat ca parametru. Tratați cazul în care orașul dat ca parametru nu există, respectiv cazul în care în orașul dat nu lucrează niciun angajat. Inserați în tabelul info\_eis informațiile corespunzătoare fiecărui caz determinat de valoarea dată pentru parametru.

```
create or replace function f3_eis (v_oras in locations.city%type)
```

```
return number is
```

```
numar_angajati number := 0;
```

```
exista_locatie number := 0;
```

```
begin
```

```
select 1 into exista_locatie
```

```
from locations
```

```
where city = v_oras;
```

```
begin
```

```
select count(*) into numar_angajati
```

```
from employees e, departments d, locations l
```

```
where e.department_id = d.department_id and d.location_id = l.location_id
```

```
and l.city = v_oras;
```

```
if numar_angajati = 0 then
```

```
insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY  
HH24:MI:SS'), 'COUNTER EMP WITH MIN 2 JOBS IN CITY - ' || v_oras, numar_angajati, 'Employees not  
found!');
```

```
return 0;
```

```
end if;
```

```
select count(e.employee_id) into numar_angajati
```

```
from employees e, departments d, locations l
```

```
where e.department_id = d.department_id and d.location_id = l.location_id
```

```

and l.city = v_oras and e.employee_id in (select e.employee_id
                                         from employees e, job_history jh
                                         where e.employee_id = jh.employee_id
                                         group by e.employee_id
                                         having count(distinct jh.job_id) >= 2);

```

```

insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'COUNTER EMP WITH MIN 2 JOBS IN CITY - ' || v_oras, numar_angajati, null);
return numar_angajati;

```

```

end;

```

```

exception

```

```

when NO_DATA_FOUND then

```

```

insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'COUNTER EMP WITH MIN 2 JOBS IN CITY - ' || v_oras, 0, 'City not found!');
return exista_locatie;

```

```

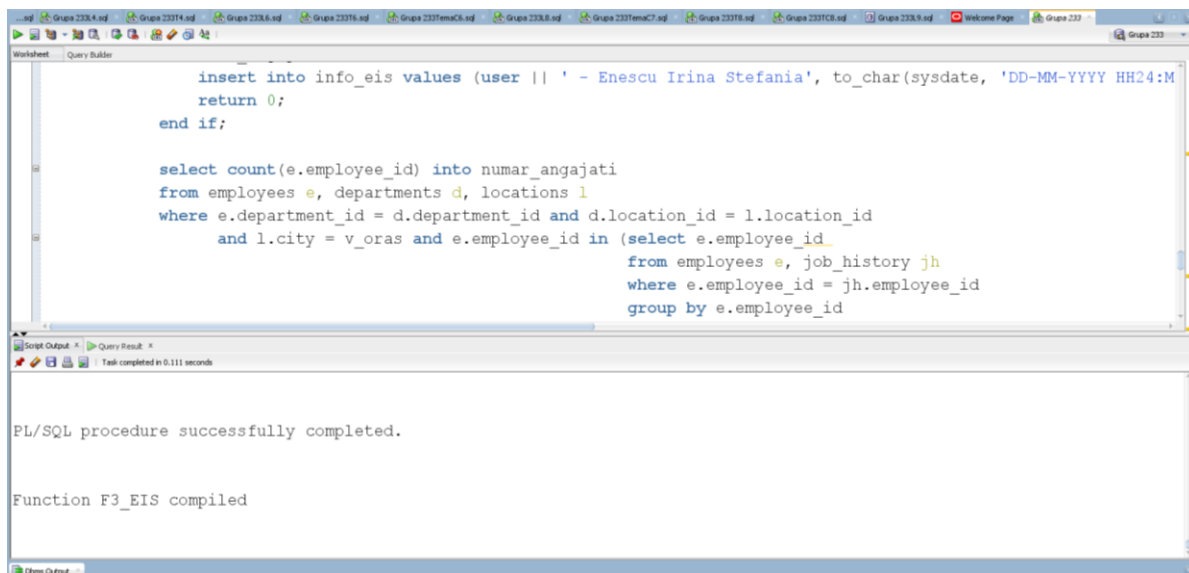
end f3_eis;

```

```

/

```



Am testat functia pe fiecare caz:

```

begin

```

```

dbms_output.put_line('Numarul de angajati este ' || f3_eis('Seattle'));
dbms_output.put_line('Numarul de angajati este ' || f3_eis('New York'));
dbms_output.put_line('Numarul de angajati este ' || f3_eis('London'));
dbms_output.put_line('Numarul de angajati este ' || f3_eis('Roma'));

```

```

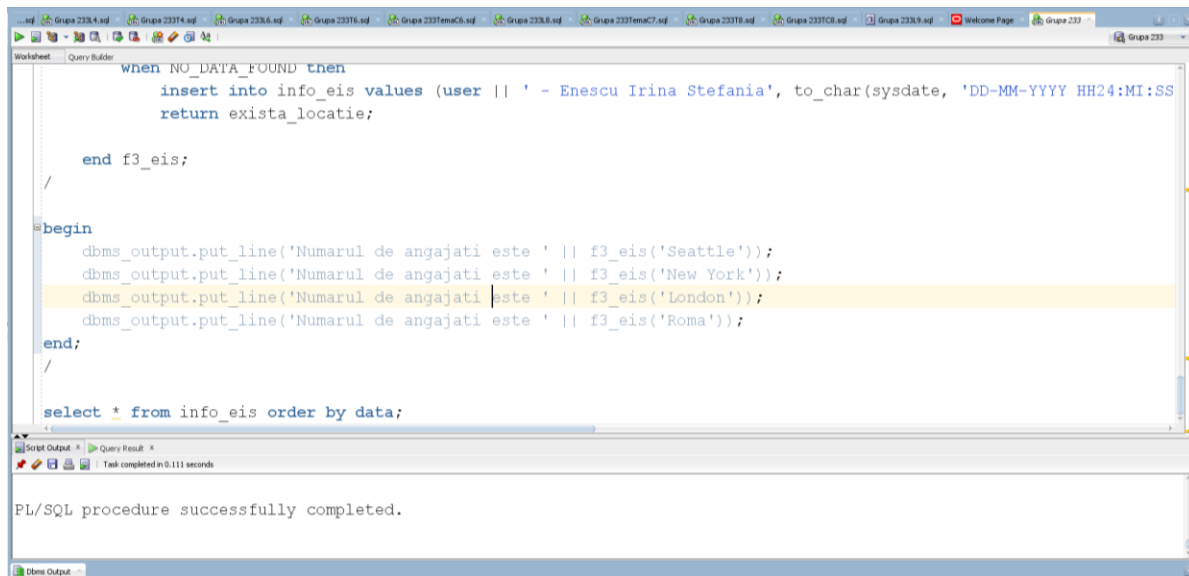
end;

```

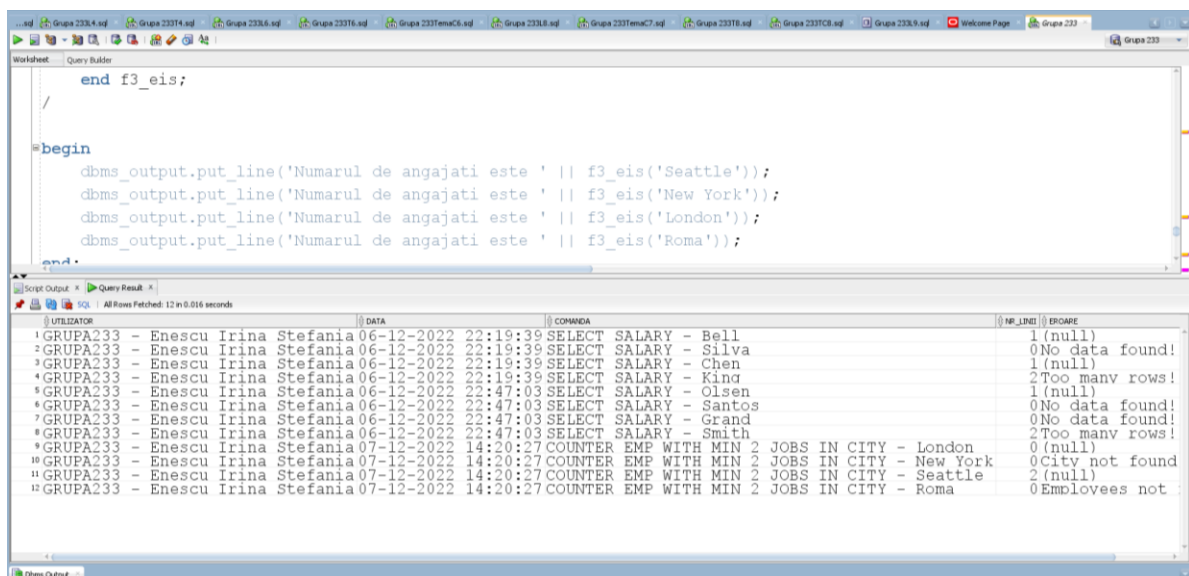
```

/

```



Tabelul arata astfel:



4. Definiți o procedură stocată care mărește cu 10% salariile tuturor angajaților conduși direct sau indirect de către un manager al cărui cod este dat ca parametru. Tratați cazul în care nu există niciun manager cu codul dat. Inserați în tabelul info\_eis informațiile corespunzătoare fiecărui caz determinat de valoarea dată pentru parametru.

```

drop table emp_eis;
create table emp_eis as select * from employees;

```

Am rezolvat problema doar pentru angajatii conduși direct:

```

create or replace procedure p4_marire_eis (v_manager_id employees.manager_id%type) is
    exista_manager number := 0;

```

```

numar_linii number := 0;
begin
    select 1 into exista_manager
    from employees
    where v_manager_id = employee_id;

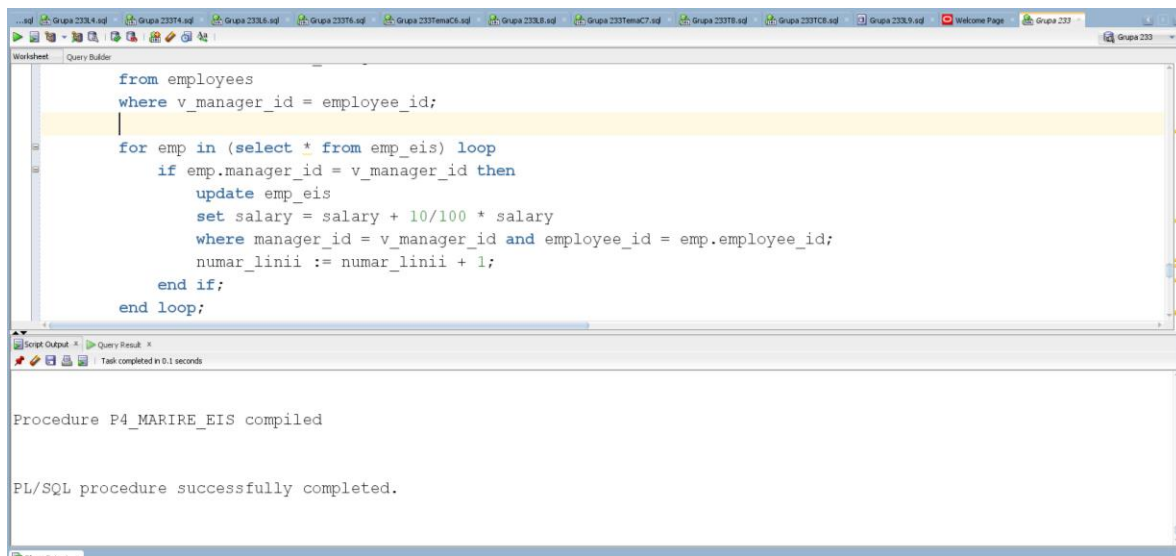
    for emp in (select * from emp_eis) loop
        if emp.manager_id = v_manager_id then
            update emp_eis
            set salary = salary + 10/100 * salary
            where manager_id = v_manager_id and employee_id = emp.employee_id;
            numar_linii := numar_linii + 1;
        end if;
    end loop;

    if numar_linii = 0 then
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'RAISE SALARY - ' || v_manager_id, numar_linii, 'Employees not found!');
    else
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'RAISE SALARY - ' || v_manager_id, numar_linii, null);
    end if;

exception
    when NO_DATA_FOUND then
        insert into info_eis values (user || ' - Enescu Irina Stefania', to_char(sysdate, 'DD-MM-YYYY
HH24:MI:SS'), 'RAISE SALARY - ' || v_manager_id, numar_linii, 'Manager not found!');

end p4_marire_eis;
/

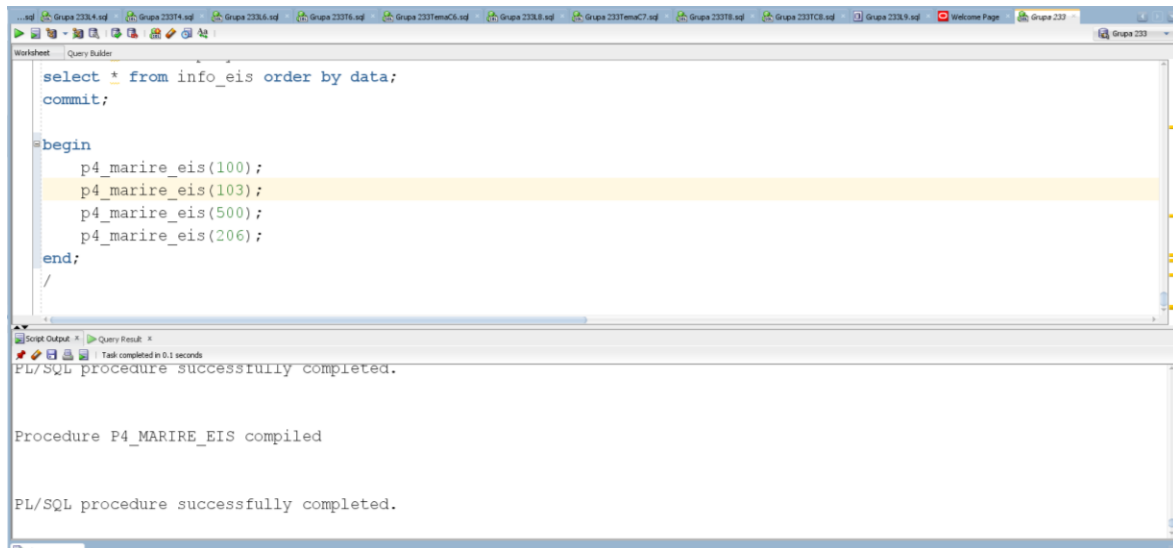
```



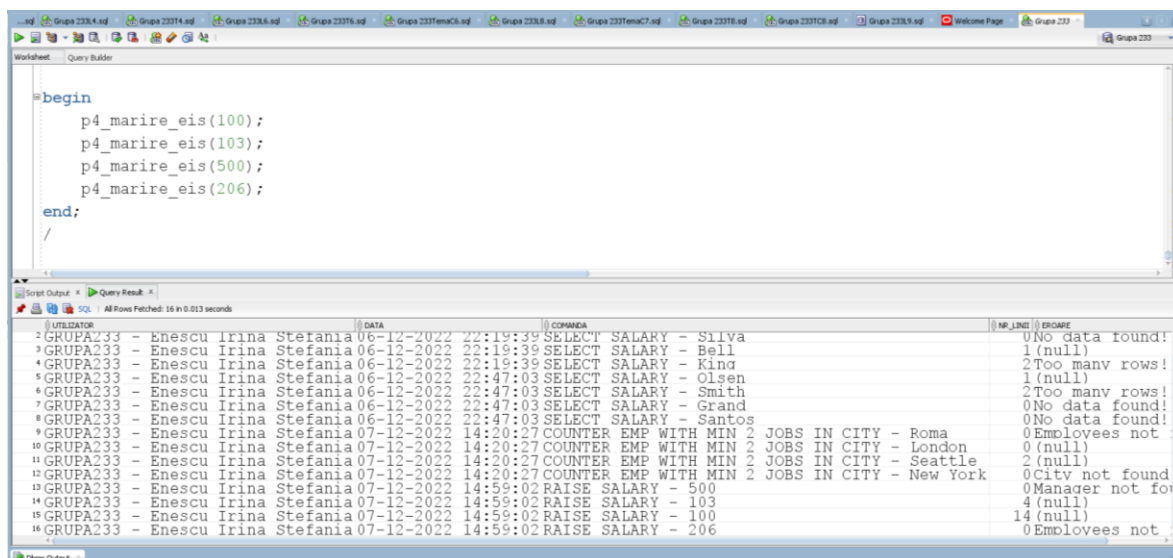


Am testat procedura pe fiecare caz:

```
begin
  p4_marire_eis(100);
  p4_marire_eis(103);
  p4_marire_eis(500);
  p4_marire_eis(206);
end;
/
```



Tabelul arata astfel:



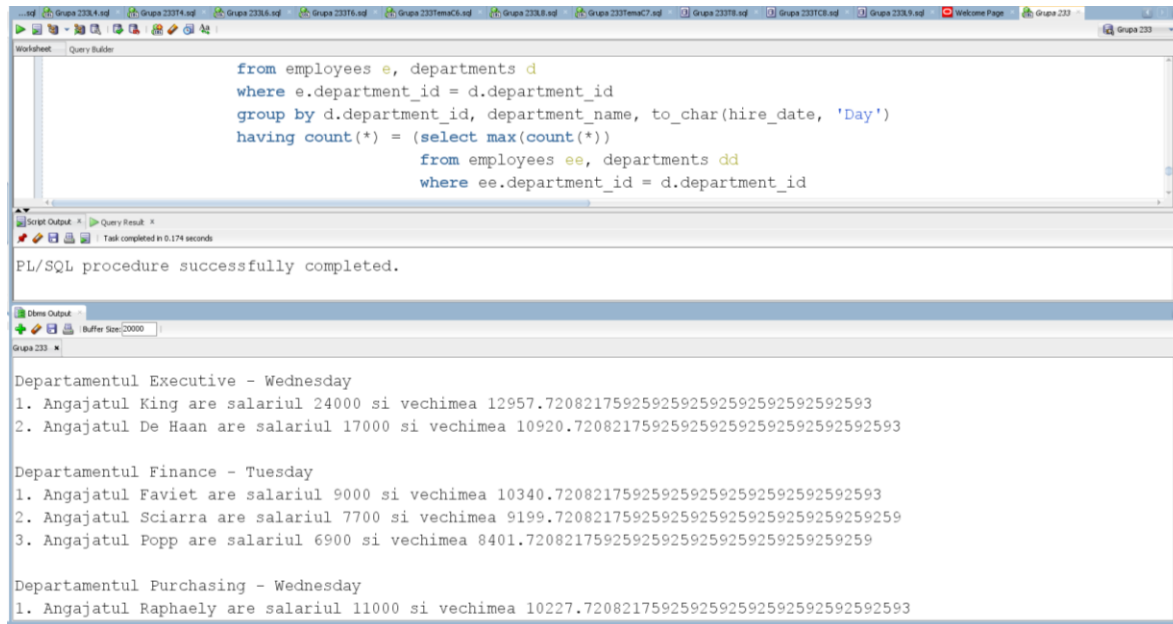
5. Definiți un subprogram care obține pentru fiecare nume de departament ziua din săptămână în care au fost angajate cele mai multe persoane, lista cu numele acestora, vechimea și venitul lor lunar. Afișați mesaje corespunzătoare următoarelor cazuri:

- într-un departament nu lucrează niciun angajat;
- într-o zi din săptămână nu a fost nimeni angajat.

Observații:

- Numele departamentului și ziua apar o singură dată în rezultat.
- Rezolvați problema în două variante, după cum se ține cont sau nu de istoricul joburilor angajaților.

Am rezolvat doar varianta în care nu se ține cont de istoricul joburilor angajaților.



create or replace procedure p5\_eis is

index\_emp number := 0;

cursor cdep is (select d.department\_id dep\_id, department\_name dep\_name, to\_char(hire\_date, 'Day') dep\_day, count(\*) dep\_count

from employees e, departments d

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

group by d.department\_id, department\_name, to\_char(hire\_date, 'Day')

having count(\*) = (select max(count(\*))

from employees ee, departments dd

where ee.department\_id = d.department\_id

group by ee.department\_id, dd.department\_name, to\_char(hire\_date, 'Day'));

cursor cemp is (select \* from employees);

begin

for dep in cdep loop

dbms\_output.put\_line('Departamentul ' || dep.dep\_name || ' - ' || dep.dep\_day);

if dep.dep\_count = 0 then

dbms\_output.put\_line('Nu lucreaza nici un angajat in acest departament.');

else

for emp in cemp loop

if emp.department\_id = dep.dep\_id and to\_char(emp.hire\_date, 'Day') = dep.dep\_day then

index\_emp := index\_emp + 1;

```

        dbms_output.put_line(index_emp || ' Angajatul ' || emp.last_name || ' are salariul ' ||
emp.salary || ' si vechimea ' || to_char(sysdate - emp.hire_date));
    end if;
end loop;
end if;
dbms_output.put_line("");
index_emp := 0;
end loop;
end p5_eis;
/

begin
    p5_eis;
end;
/

```

Raspunsurile sunt:

Departamentul Executive - Wednesday

1. Angajatul King are salariul 24000 si vechimea 12957.7208217592592592592592592593
2. Angajatul De Haan are salariul 17000 si vechimea 10920.7208217592592592592592592593

Departamentul Finance - Tuesday

1. Angajatul Faviat are salariul 9000 si vechimea 10340.7208217592592592592592592593
2. Angajatul Sciarra are salariul 7700 si vechimea 9199.7208217592592592592592592593
3. Angajatul Popp are salariul 6900 si vechimea 8401.7208217592592592592592592593

Departamentul Purchasing - Wednesday

1. Angajatul Raphaely are salariul 11000 si vechimea 10227.7208217592592592592592592593
2. Angajatul Baida are salariul 2900 si vechimea 9114.7208217592592592592592592593

Departamentul Sales - Tuesday

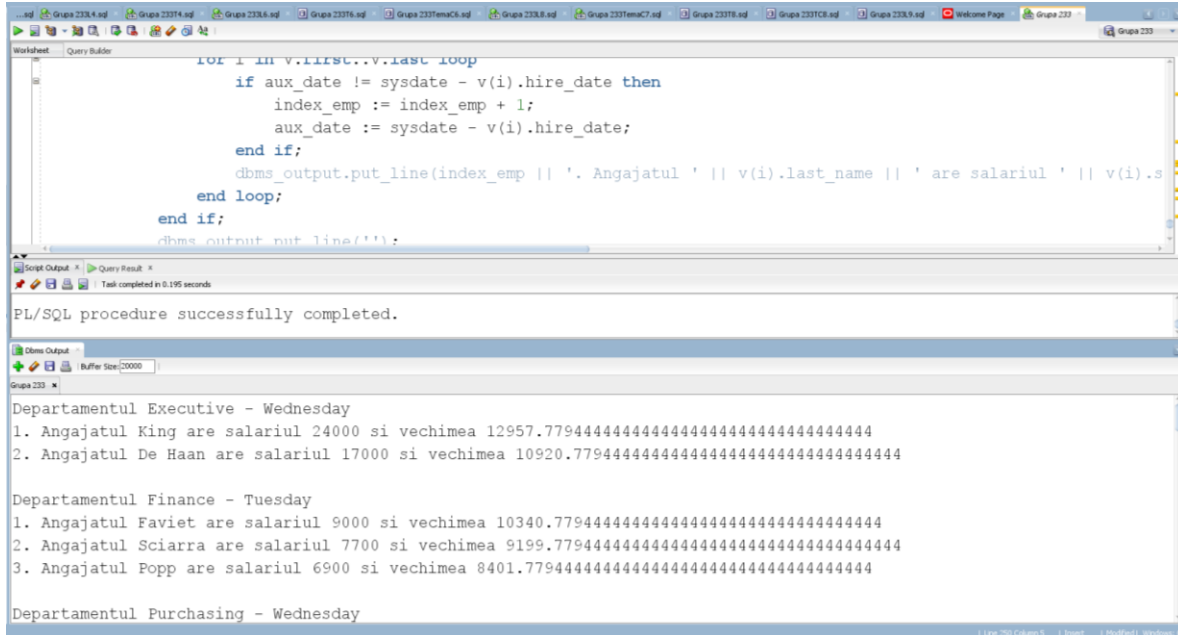
1. Angajatul Russell are salariul 14000 si vechimea 9563.7208217592592592592592592593
2. Angajatul Tuvault are salariul 7000 si vechimea 8415.7208217592592592592592592593
3. Angajatul King are salariul 10000 si vechimea 9808.7208217592592592592592592593
4. Angajatul Sewall are salariul 7000 si vechimea 8800.7208217592592592592592592593
5. Angajatul Vishney are salariul 10500 si vechimea 9157.7208217592592592592592592593
6. Angajatul Ozer are salariul 11500 si vechimea 9402.7208217592592592592592592593
7. Angajatul Smith are salariul 7400 si vechimea 8688.7208217592592592592592592593
8. Angajatul Taylor are salariul 8600 si vechimea 9024.7208217592592592592592592593
9. Angajatul Johnson are salariul 6200 si vechimea 8373.7208217592592592592592592593

Departamentul Shipping - Thursday

1. Angajatul Weiss are salariul 8000 si vechimea 9638.7208217592592592592592592593
2. Angajatul Fripp are salariul 8200 si vechimea 9372.7208217592592592592592592593
3. Angajatul Landry are salariul 2400 si vechimea 8728.7208217592592592592592592593
4. Angajatul Atkinson are salariul 2800 si vechimea 9169.7208217592592592592592592593
5. Angajatul Seo are salariul 2700 si vechimea 9064.7208217592592592592592592593

6. Angajatul Vargas are salariul 2500 si vechimea 8917.720821759259259259259259259259259  
7. Angajatul Geoni are salariul 2800 si vechimea 8343.720821759259259259259259259259259  
8. Angajatul Bull are salariul 4100 si vechimea 9421.720821759259259259259259259259259  
9. Angajatul Grant are salariul 2600 si vechimea 8364.720821759259259259259259259259259

Am rezolvat doar varianta in care nu se tine cont de istoricul joburilor angajatilor.  
Am ordonat angajatii descrescator in functie de vechime.



```

create or replace procedure p5_eis is
    index_emp number := 0;
    type vector is varray(1000) of employees%rowtype;
    cursor cdep is (select d.department_id dep_id, department_name dep_name, to_char(hire_date,
'Day') dep_day, count(*) dep_count
                    from employees e, departments d
                    where e.department_id = d.department_id
                    group by d.department_id, department_name, to_char(hire_date, 'Day')
                    having count(*) = (select max(count(*))
                                     from employees ee, departments dd
                                     where ee.department_id = d.department_id
                                     group by ee.department_id, dd.department_name, to_char(hire_date, 'Day')));
    cursor cemp is (select * from employees);
    v vector;
    aux employees%rowtype;
    aux_date varchar2(78);
    begin
        for dep in cdep loop
            dbms_output.put_line('Departamentul ' || dep.dep_name || ' - ' || dep.dep_day);
            if dep.dep_count = 0 then
                dbms_output.put_line('Nu lucreaza nici un angajat in acest departament.');
            else
                v := vector();
                for emp in cemp loop
                    if emp.department_id = dep.dep_id and to_char(emp.hire_date, 'Day') = dep.dep_day then
                        v.extend();
                    end if;
                end loop;
            end if;
        end loop;
    end;

```

```

        v(v.last) := emp;
    end if;
end loop;
for i in v.first..v.last loop
    for j in v.first..v.last loop
        if (sysdate - v(i).hire_date) >= (sysdate - v(j).hire_date) then
            aux := v(i);
            v(i) := v(j);
            v(j) := aux;
        end if;
    end loop;
end loop;
index_emp := 1;
aux_date := sysdate - v(v.first).hire_date;
for i in v.first..v.last loop
    if aux_date != sysdate - v(i).hire_date then
        index_emp := index_emp + 1;
        aux_date := sysdate - v(i).hire_date;
    end if;
    dbms_output.put_line(index_emp || ' . Angajatul ' || v(i).last_name || ' are salariul ' ||
v(i).salary || ' si vechimea ' || to_char(sysdate - v(i).hire_date));
end loop;
end if;
dbms_output.put_line("");
end loop;
end p5_eis;
/

begin
    p5_eis;
end;
/

```

Cazul in care doi angajati au aceeasi vechime se observa la:

Departamentul Accounting - Tuesday

1. Angajatul Higgins are salariul 12000 si vechimea 10410.783275462962962962962962963
1. Angajatul Gietz are salariul 8300 si vechimea 10410.783275462962962962962962963

Rezultatle sunt:

Departamentul Executive - Wednesday

1. Angajatul King are salariul 24000 si vechimea 12957.783275462962962962962962963
2. Angajatul De Haan are salariul 17000 si vechimea 10920.783275462962962962962962963

Departamentul Finance - Tuesday

1. Angajatul Faviat are salariul 9000 si vechimea 10340.783275462962962962962962963
2. Angajatul Sciarra are salariul 7700 si vechimea 9199.783275462962962962962962963

3. Angajatul Popp are salariul 6900 si vechimea 8401.783275462962962962962962962963

Departamentul Purchasing - Wednesday

1. Angajatul Raphaely are salariul 11000 si vechimea 10227.783275462962962962962962962963

2. Angajatul Baida are salariul 2900 si vechimea 9114.783275462962962962962962962963

Departamentul Sales - Tuesday

1. Angajatul King are salariul 10000 si vechimea 9808.783275462962962962962962962963

2. Angajatul Russell are salariul 14000 si vechimea 9563.783275462962962962962962962963

3. Angajatul Ozer are salariul 11500 si vechimea 9402.783275462962962962962962962963

4. Angajatul Vishney are salariul 10500 si vechimea 9157.783275462962962962962962962963

5. Angajatul Taylor are salariul 8600 si vechimea 9024.783275462962962962962962962963

6. Angajatul Sewall are salariul 7000 si vechimea 8800.783275462962962962962962962963

7. Angajatul Smith are salariul 7400 si vechimea 8688.783275462962962962962962962963

8. Angajatul Tuvault are salariul 7000 si vechimea 8415.783275462962962962962962962963

9. Angajatul Johnson are salariul 6200 si vechimea 8373.783275462962962962962962962963

Departamentul Shipping - Thursday

1. Angajatul Weiss are salariul 8000 si vechimea 9638.783275462962962962962962962963

2. Angajatul Bull are salariul 4100 si vechimea 9421.783275462962962962962962962963

3. Angajatul Fripp are salariul 8200 si vechimea 9372.783275462962962962962962962963

4. Angajatul Atkinson are salariul 2800 si vechimea 9169.783275462962962962962962962963

5. Angajatul Seo are salariul 2700 si vechimea 9064.783275462962962962962962962963

6. Angajatul Vargas are salariul 2500 si vechimea 8917.783275462962962962962962962963

7. Angajatul Landry are salariul 2400 si vechimea 8728.783275462962962962962962962963

8. Angajatul Grant are salariul 2600 si vechimea 8364.783275462962962962962962962963

9. Angajatul Geoni are salariul 2800 si vechimea 8343.783275462962962962962962962963

Departamentul Shipping - Wednesday

1. Angajatul Davies are salariul 3100 si vechimea 9443.783275462962962962962962962963

2. Angajatul Nayer are salariul 3200 si vechimea 9275.783275462962962962962962962963

3. Angajatul Dilly are salariul 3600 si vechimea 9247.783275462962962962962962962963

4. Angajatul Bissot are salariul 3300 si vechimea 9240.783275462962962962962962962963

5. Angajatul Dellinger are salariul 3400 si vechimea 8932.783275462962962962962962962963

6. Angajatul McCain are salariul 3200 si vechimea 8925.783275462962962962962962962963

7. Angajatul Rogers are salariul 2900 si vechimea 8869.783275462962962962962962962963

8. Angajatul Jones are salariul 2800 si vechimea 8666.783275462962962962962962962963

9. Angajatul Markle are salariul 2200 si vechimea 8309.783275462962962962962962962963

Departamentul IT - Wednesday

1. Angajatul Hunold are salariul 9000 si vechimea 12026.783275462962962962962962962963

2. Angajatul Austin are salariul 4800 si vechimea 9296.783275462962962962962962962963

Departamentul Marketing - Sunday

1. Angajatul Fay are salariul 6000 si vechimea 9243.783275462962962962962962962963

Departamentul Human Resources - Tuesday

1. Angajatul Mavris are salariul 6500 si vechimea 10410.783275462962962962962962962963

Departamentul Public Relations - Tuesday

1. Angajatul Baer are salariul 10000 si vechimea 10410.783275462962962962962962962963

Departamentul Accounting - Tuesday

1. Angajatul Higgins are salariul 12000 si vechimea 10410.783275462962962962962962962963

1. Angajatul Gietz are salariul 8300 si vechimea 10410.783275462962962962962962962963

Departamentul Purchasing - Thursday

1. Angajatul Khoo are salariul 3100 si vechimea 10065.783275462962962962962962962963

2. Angajatul Tobias are salariul 2800 si vechimea 9267.783275462962962962962962962963

Departamentul Administration - Thursday

1. Angajatul Whalen are salariul 4400 si vechimea 12865.783275462962962962962962962963

Departamentul Marketing - Saturday

1. Angajatul Hartstein are salariul 13000 si vechimea 9790.783275462962962962962962962963