

EXERCITII

1. 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. Inșerați în tabelul info_*** informațiile corespunzătoare fiecărui caz determinat de valoarea dată pentru parametru.

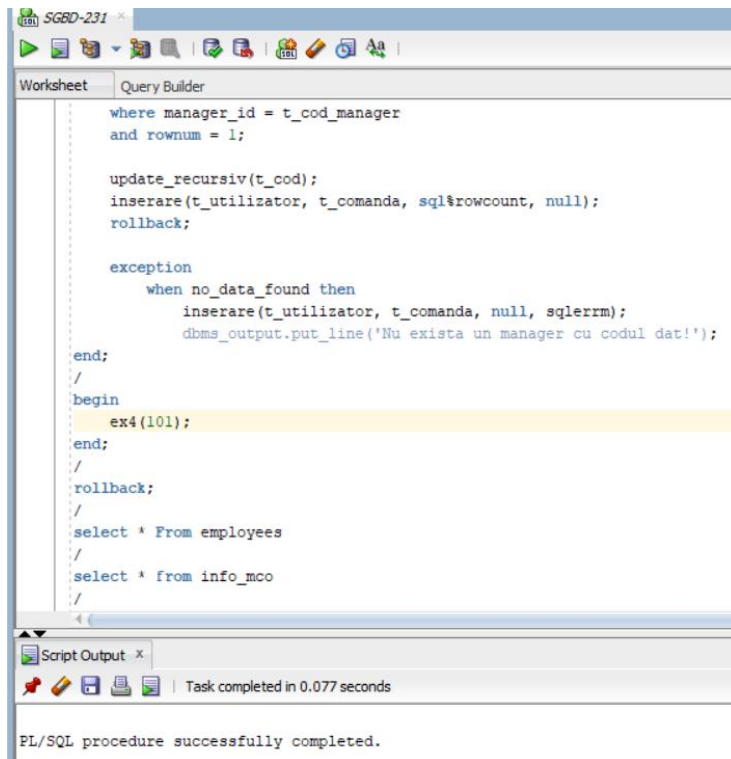
Explicatie – In primul rand, verific daca exista managerul cu acel id printr-un simplu “SELECT” la inceput. In cazul in care nu exista, am tratat exceptia.

Apoi, am apelat o procedura separata care se ocupa de partea recursiva si gaseste fiecare subordonat (direct sau indirect), modificandu-i salariul.

In final, inserez datele in tabelul “INFO_MCO” cu inca o procedura separata, declarata cu “PRAGMA AUTONOMOUS_TRANSACTION”.

Mai jos am inclus niste screenshot-uri care arata ce se intampla cand pun un ID valid (si am aratat cum se modifica salariul pentru un subordonat aleator) si unul invalid (nu exista un manager).

COD CARE EXISTA:



```
where manager_id = t_cod_manager
and rownum = 1;

update_recurziv(t_cod);
inserare(t_utilizator, t_comanda, sql%rowcount, null);
rollback;

exception
when no_data_found then
  inserare(t_utilizator, t_comanda, null, sqlerrm);
  dbms_output.put_line('Nu exista un manager cu codul dat!');

end;
/
begin
  ex4(101);
end;
/
rollback;
/
select * From employees
/
select * from info_mco
/
```

Script Output x

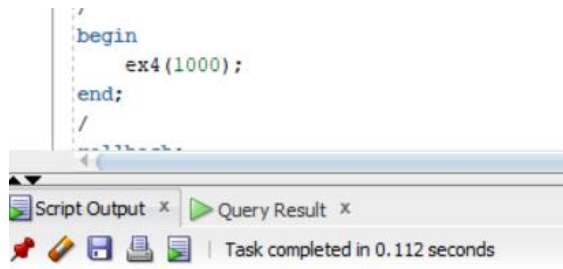
Task completed in 0.077 seconds

PL/SQL procedure successfully completed.

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	112	Jose	Manuel	Uрман	JMURMAN 515.124.4469	07-MAR-98	FI_ACCOUNT	47704.12

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-MAR-98	FI_ACCOUNT	52474.53

COD CARE NU EXISTA:



PL/SQL procedure successfully completed.

	NR_Linii	EROARE
1 loop update employees set salary = salary + salary / 10 where employee_id = 1.employee_id; update_recurziv(1.employee_id); end loop;	1	(null)
2 loop update employees set salary = salary + salary / 10 where employee_id = 1.employee_id; update_recurziv(1.employee_id); end loop;	(null)	ORA-01403: no data found

create or replace procedure inserare(utilizator info_mco.utilizator%type,

comanda info_mco.comanda%type, nr_linii info_mco.nr_linii%type,

eroare info_mco.eroare%type) as

pragma autonomous_transaction;

begin

insert into info_mco

values (utilizator, sysdate, comanda, nr_linii, eroare);

commit;

end;

/

create or replace procedure update_recurziv(t_cod_manager employees.manager_id%type) as

begin

for i in (select employee_id from employees where manager_id = t_cod_manager) loop

update employees

```

        set salary = salary + salary / 10
        where employee_id = i.employee_id;
        dbms_output.put_line('Employee: ' || i.employee_id || ' Manager: ' || t_cod_manager);
        update_recurziv(i.employee_id);
    end loop;
end;
/

create or replace procedure ex4(t_cod_manager employees.manager_id%type) as
    t_utilizator varchar2(20);
    error_message varchar2(50) := "";
    t_comanda info_mco.comanda%type := 'for i in (select employee_id from employees where
manager_id = t_cod_manager) loop update employees set salary = salary + salary / 10 where employee_id
= i.employee_id; update_recurziv(i.employee_id); end loop;';
    t_cod employees.manager_id%type;
begin
    select user
    into t_utilizator
    from dual;

    select manager_id
    into t_cod
    from employees
    where manager_id = t_cod_manager
    and rownum = 1;

    update_recurziv(t_cod);
    inserare(t_utilizator, t_comanda, sql%rowcount, null);
    rollback;

    exception
        when no_data_found then

```

```
        inserare(t_utilizator, t_comanda, null, sqlerrm);  
        dbms_output.put_line('Nu exista un manager cu codul dat!');  
end;  
  
/  
  
begin  
    ex4(101);  
end;
```

2. 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:

a. Numele departamentului și ziua apar o singură dată în rezultat.

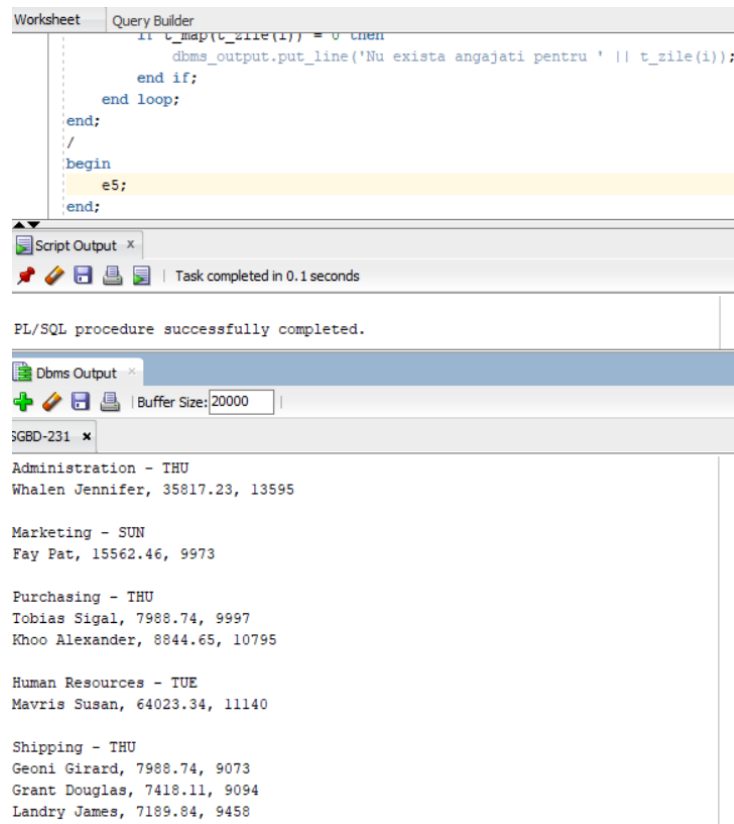
b. Rezolvați problema în două variante, după cum se ține cont sau nu de istoricul joburilor angajaților.

Explicatie – In primul rand, am folosit un ciclu cursor ca sa trec prin fiecare “department_id”.

Apoi, pentru fiecare departament am facut rost de nume, de ziua saptamanii in care au fost cele mai multe persoane angajate si de numarul de angajati in ziua respectiva, stocand aceste date in 3 variabile (2 “SELECT”-uri).

Am afisat numele departamentului; in cazul in care departamentul nu avea angajati, am pus un mesaj corespunzator. Altfel, am folosit un ciclu cursor ca sa trec prin fiecare persoana care a fost angajata in ziua respectiva pentru acel departament, afisand numele, venitul si vechimea.

De asemenea, am folosit un VARRAY de 7 elemente ca sa retin cele 7 zile ale saptamanii, indexul fiind ziua, iar valoarea fiind un numar (0 sau 1). Valoarea este 0 daca nu au fost persoane angajate in acea zi a saptamanii; altfel, este 1. Stocand aceste informatii, am afisat la final zilele in care nu au fost persoane angajate deloc.



```
Worksheet Query Builder
if t_map(t_zile(i)) = 0 then
  dbms_output.put_line('Nu exista angajati pentru ' || t_zile(i));
end if;
end loop;
end;
/
begin
  e5;
end;
```

Script Output x
Task completed in 0.1 seconds

PL/SQL procedure successfully completed.

Dbms Output x
Buffer Size: 20000

SGBD-231 x

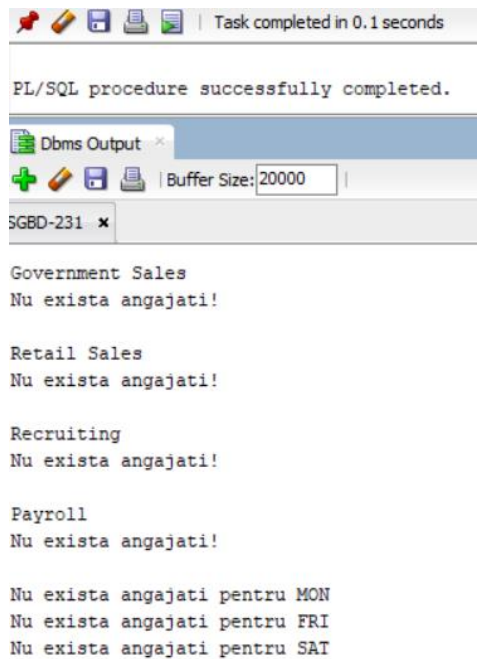
Administration - THU
Whalen Jennifer, 35817.23, 13595

Marketing - SUN
Pay Pat, 15562.46, 9973

Purchasing - THU
Tobias Sigal, 7988.74, 9997
Khoo Alexander, 8844.65, 10795

Human Resources - TUE
Mavris Susan, 64023.34, 11140

Shipping - THU
Geoni Girard, 7988.74, 9073
Grant Douglas, 7418.11, 9094
Landry James, 7189.84, 9458
..



Task completed in 0.1 seconds

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

SGBD-231 x

Government Sales
Nu exista angajati!

Retail Sales
Nu exista angajati!

Recruiting
Nu exista angajati!

Payroll
Nu exista angajati!

Nu exista angajati pentru MON
Nu exista angajati pentru FRI
Nu exista angajati pentru SAT

create or replace procedure e5 as

```
t_nume departments.department_name%type;
```

```
t_zi varchar2(3);
```

```
t_numar number(2);
```

```
type t_zi_val is table of number index by varchar2(3);
```

```
t_map t_zi_val;
```

```
type t_zile_vect is varray(7) of varchar2(3);
```

```
t_zile t_zile_vect := t_zile_vect('MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN');
```

```
begin
```

```
for i in 1..7 loop
```

```
    t_map(t_zile(i)) := 0;
```

```
end loop;
```

```
for i in (select department_id from departments) loop
```

```
    select Nume, Zi, Numar
```

```
    into t_nume, t_zi, t_numar
```

```

from (
    select d.department_name as Nume, to_char(e.hire_date, 'DY') as Zi,
    count(e.employee_id) as Numar
    from departments d
    left join employees e on e.department_id = d.department_id
    where d.department_id = i.department_id
    group by d.department_name, to_char(e.hire_date, 'DY')
    order by Numar desc)
where rownum = 1;

dbms_output.put(t_nume);
if t_numar = 0 then
    dbms_output.new_line;
    dbms_output.put_line('Nu exista angajati!');
else
    dbms_output.put_line(' - ' || t_zi);
    t_map(t_zi) := 1;
    for j in (select last_name || ' ' || first_name as Nume,
        salary + nvl(commission_pct, 0) * salary as Venit,
        floor(sysdate - hire_date) as Vechime
        from employees
        where department_id = i.department_id
        and to_char(hire_date, 'DY') = t_zi
        order by Vechime) loop
        dbms_output.put_line(j.Nume || ', ' || j.Venit || ', ' || j.Vechime);
    end loop;
end if;

dbms_output.new_line;
end loop;

```

```
for i in 1..7 loop
  if t_map(t_zile(i)) = 0 then
    dbms_output.put_line('Nu exista angajati pentru ' || t_zile(i));
  end if;
end loop;
end;
/
begin
  e5;
end;
```


3. Modificați exercițiul anterior astfel încât lista cu numele angajaților să apară într-un clasament creat în funcție de vechimea acestora în departament. Specificați numărul poziției din clasament și apoi lista angajaților care ocupă acel loc. Dacă doi angajați au aceeași vechime, atunci aceștia ocupă aceeași poziție în clasament.

Explicație – am folosit cateva variabile extra: “t_index” pentru pozitia in clasament, “t_vechime” pentru vechimea angajatului curent si “t_aux” pentru vechimea ultimului angajat, ca sa putem verifica daca au aceeasi vechime (daca au acelasi clasament).

Am stocat datele legate de angajati intr-un cursor pe care l-am utilizat cu “OPEN”, “CLOSE” si “FETCH INTO”. Mereu am dat “FETCH INTO” la primul angajat inainte de loop (ca sa pot stoca vechimea in “t_aux”), apoi am intrat in loop si am afisat urmasorii angajati (comparand vechimea). Daca nu au aceeași vechime, cresc “t_index” cu 1 si trec pe alt rand; altfel, “t_index” ramane la fel.

In rest, totul functioneaza aproape la fel ca la ultimul exercitiu.

```
Dbms Output - x
+ | Buffer Size: 20000
SGBD-231 x
5. Taylor Jonathon 26767 9754
6. Sewall Sarath 22695 9530
7. Smith William 22073 9418
8. Tuvault Oliver 20880 9145
9. Johnson Charles 17689 9103

Executive - WED
1. King Steven 40585 13687
2. De Haan Lex 74564 11650

Finance - TUE
1. Faviet Daniel 66602 11070
2. Sciarra Ismael 56982 9929
3. Popp Luis 51062 9131

Accounting - TUE
1. Gietz William 61422 11140 Higgins Shelley 88803 11140

SGBD-231 x
Government Sales
Nu exista angajati!

Retail Sales
Nu exista angajati!

Recruiting
Nu exista angajati!

Payroll
Nu exista angajati!

Nu exista angajati pentru MON
Nu exista angajati pentru FRI
Nu exista angajati pentru SAT
```

create or replace procedure e6 as

t_nume_dep departments.department_name%type;

t_zi varchar2(9);

t_numar number(2);

type t_zi_val is table of number index by varchar2(3);

t_map t_zi_val;

type t_zile_vect is varray(7) of varchar2(9);

t_zile t_zile_vect := t_zile_vect('MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN');

t_index number(2);

t_nume_ang varchar2(50);

t_venit number(6);

t_vechime number(6);

t_aux number(6);

cursor t_cursor (p_cod departments.department_id%type, p_zi varchar2) is

select last_name || ' ' || first_name as Nume,

salary + nvl(commission_pct, 0) * salary as Venit,

floor(sysdate - hire_date) as Vechime

from employees

where department_id = p_cod

and to_char(hire_date, 'DY') = p_zi

order by Vechime desc;

begin

for i in 1..7 loop

t_map(t_zile(i)) := 0;

end loop;

for i in (select department_id from departments) loop

```

select Nume, Zi, Numar
into t_nume_dep, t_zi, t_numar
from (
    select d.department_name as Nume, to_char(e.hire_date, 'DY') as Zi,
    count(e.employee_id) as Numar
    from departments d
    left join employees e on e.department_id = d.department_id
    where d.department_id = i.department_id
    group by d.department_name, to_char(e.hire_date, 'DY')
order by Numar desc)
where rownum = 1;

```

```

dbms_output.put(t_nume_dep);
if t_numar = 0 then
    dbms_output.new_line;
    dbms_output.put_line('Nu exista angajati!');
else
    t_map(t_zi) := 1;
    dbms_output.put_line(' - ' || t_zi);
    t_index := 1;
    t_vechime := -1;
    open t_cursor(i.department_id, t_zi);
    fetch t_cursor into t_nume_ang, t_venit, t_vechime;
    dbms_output.put('1. ' || t_nume_ang || ' ' || t_venit || ' ' || t_vechime);
    t_index := 2;
    t_aux := t_vechime;
    loop
        fetch t_cursor into t_nume_ang, t_venit, t_vechime;
        exit when t_cursor%notfound;
        if t_vechime != t_aux then

```

```

        dbms_output.new_line;
        dbms_output.put(t_index || '.' || t_ume_ang || ' ' || t_venit || ' ' || t_vechime);
        t_index := t_index + 1;
    else
        dbms_output.put(' ' || t_ume_ang || ' ' || t_venit || ' ' || t_vechime);
    end if;

    t_aux := t_vechime;
end loop;
close t_cursor;
dbms_output.new_line;
end if;
dbms_output.new_line;
end loop;

for i in 1..7 loop
    if t_map(t_zile(i)) = 0 then
        dbms_output.put_line('Nu exista angajati pentru ' || t_zile(i));
    end if;
end loop;
end;
/

begin
    e6;
end;
```