REZOLVARI

E2: In primul rand, am rezolvat exercitiul E1 folosind expresii cursor (un cursor care retine titlul job-ului, si inca un cursor imbricat care retine lista de angajati pentru fiecare job, cu detalii despre fiecare angajat – nume, salariu, comision).

Am parcurs datele stocate in cursor si le-am afisat (titlul fiecarui job cu numele fiecarui angajat + salariu).

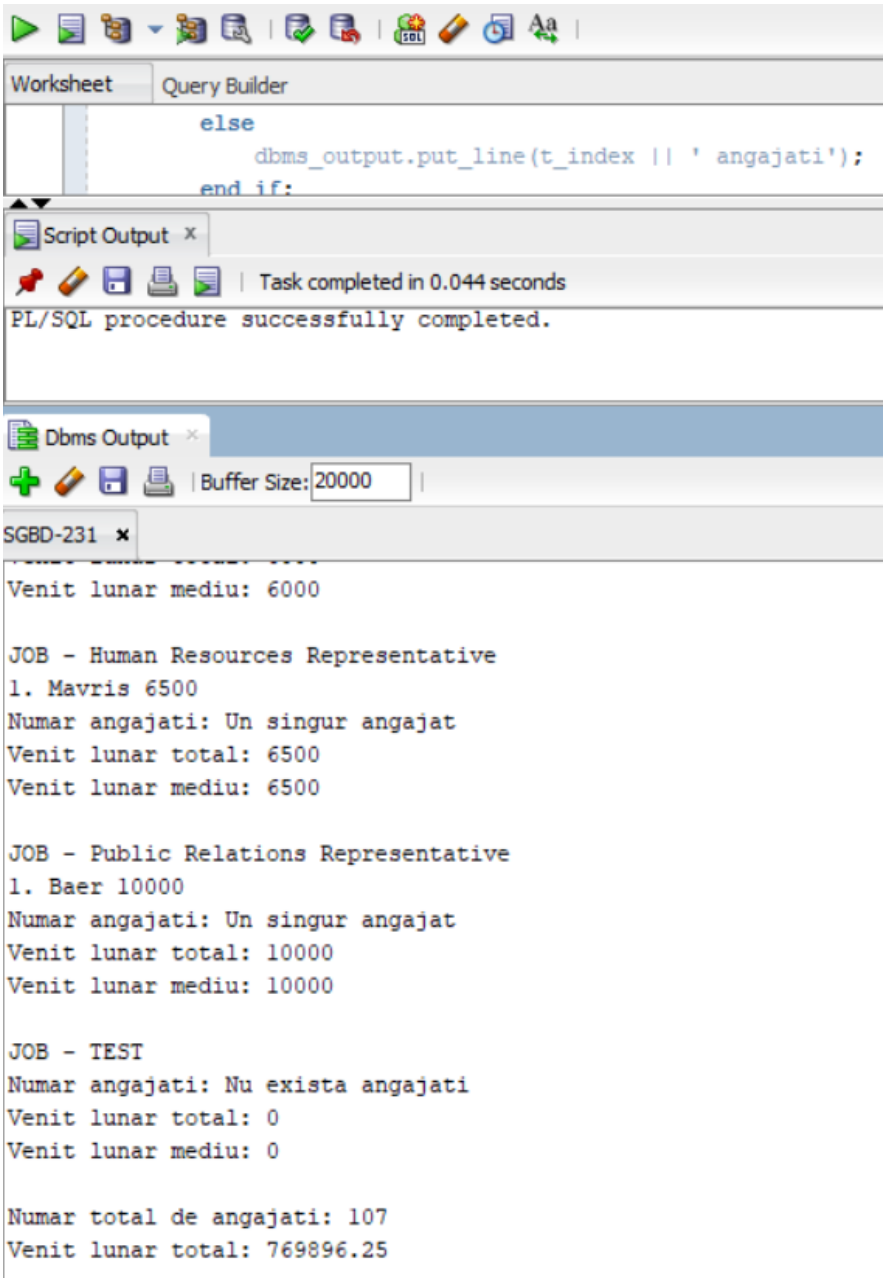
Pentru a asocia fiecarui angajat un numar de ordine (resetat pentru fiecare job), am folosit o variabila auxiliara care se incrementeaza pentru fiecare angajat pentru job-ul respectiv; aceasta isi reseteaza valoarea inapoi la “0” de fiecare data cand trecem la alt job. La final va trebui sa afisam numarul total de angajati, asadar am adunat valoarea din acea variabila auxiliara la o alta variabila de fiecare data cand trebuia sa fie resetata.

De asemenea, pentru valoarea lunara a veniturilor angajatilor si valoarea totala lunara am procedat la fel; datele sunt diferite, dar mecanismul este identic.

```
declare
   type refcursor is ref cursor;
   cursor t_info is
      select j.job_title,
      cursor (
         select last_name as nume, salary as salariu, nvl(commission_pct, 0) as comision
         from employees e
         where e.job_id = j.job_id)
      from jobs j;

   t_cursor refcursor;
   t_job jobs.job_title%type;
   t_nume employees.last_name%type;
   t_salariu employees.salary%type;
   t_comision employees.commission_pct%type;

   t_index       number;
```

```
    t_nr_angajati number;
    t_venit_lunar number;

    t_nr_angajati_total number := 0;
    t_venit_lunar_total number := 0;
begin
    open t_info;
    loop
        fetch t_info into t_job, t_cursor;
        exit when t_info%notfound;
        dbms_output.put_line('JOB - ' || t_job);

        t_index       := 0;
        t_venit_lunar := 0;
        loop
            fetch t_cursor into t_nume, t_salariu, t_comision;
            exit when t_cursor%notfound;

            t_index       := t_index + 1;
            t_venit_lunar := t_venit_lunar + t_salariu + t_salariu * t_comision;

            dbms_output.put_line(t_index || '. ' || t_nume || ' ' || t_salariu);
        end loop;

        t_nr_angajati_total := t_nr_angajati_total + t_index;
        t_venit_lunar_total := t_venit_lunar_total + t_venit_lunar;

        dbms_output.put('Numar angajati: ');
```

```
        if t_index = 0 then
            dbms_output.put_line('Nu exista angajati');
        elsif t_index = 1 then
            dbms_output.put_line('Un singur angajat');
        else
            dbms_output.put_line(t_index || ' angajati');
        end if;

        dbms_output.put_line('Venit lunar total: ' || t_venit_lunar);
        dbms_output.put('Venit lunar mediu: ');
        if t_index != 0 then
            dbms_output.put_line(round(t_venit_lunar / t_index, 3));
        else
            dbms_output.put_line('0');
        end if;
        dbms_output.new_line;
    end loop;
    close t_info;

    dbms_output.put_line('Numar total de angajati: ' || t_nr_angajati_total);
    dbms_output.put_line('Venit lunar total: ' || t_venit_lunar_total);
end;
```

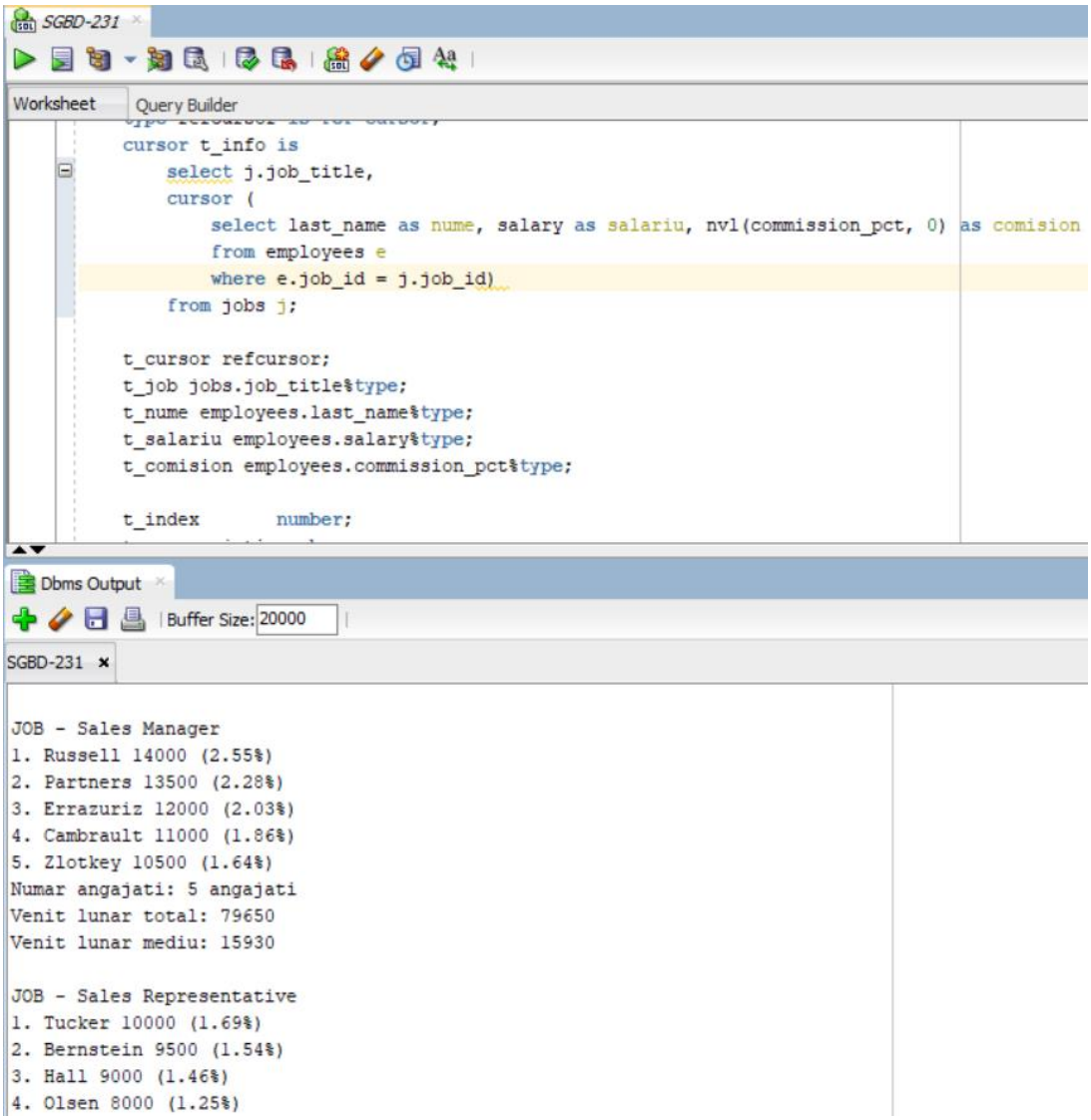

Worksheet
Query Builder
else
dbms_output.put_line(t_index || ' angajati');
end if;
Script Output
Task completed in 0.044 seconds
PL/SQL procedure successfully completed.
Dbms Output
Buffer Size: 20000
SGBD-231
Venit lunar mediu: 6000
JOB - Human Resources Representative
1. Mavris 6500
Numar angajati: Un singur angajat
Venit lunar total: 6500
Venit lunar mediu: 6500
JOB - Public Relations Representative
1. Baer 10000
Numar angajati: Un singur angajat
Venit lunar total: 10000
Venit lunar mediu: 10000
JOB - TEST
Numar angajati: Nu exista angajati
Venit lunar total: 0
Venit lunar mediu: 0
Numar total de angajati: 107
Venit lunar total: 769896.25

E3: prin select-ul de la inceput, am calculat suma totala alocata lunar pentru plata salariilor si comisioanelor tuturor angajatilor. Apoi, pentru fiecare angajat am calculat cat la suta din aceasta suma castiga lunar, impartind venitul sau lunar la suma totala si inmultind cu 100.

```
declare
   type refcursor is ref cursor;
   cursor t_info is
      select j.job_title,
      cursor (
         select last_name as nume, salary as salariu, nvl(commission_pct, 0) as comision
         from employees e
         where e.job_id = j.job_id)
      from jobs j;

   t_cursor refcursor;
   t_job jobs.job_title%type;
   t_nume employees.last_name%type;
   t_salariu employees.salary%type;
   t_comision employees.commission_pct%type;

   t_index       number;
   t_nr_angajati number;
   t_venit_lunar number;

   t_nr_angajati_total number := 0;
   t_venit_lunar_total number := 0;

   t_suma_totala number;
begin
```

```
select sum(salary + salary * nvl(commission_pct, 0))
into t_suma_totala
from employees;
dbms_output.put_line('Suma totala alocata lunar: ' || t_suma_totala);
dbms_output.new_line;

open t_info;
loop
   fetch t_info into t_job, t_cursor;
   exit when t_info%notfound;
   dbms_output.put_line('JOB - ' || t_job);

   t_index       := 0;
   t_venit_lunar := 0;
   loop
      fetch t_cursor into t_nume, t_salariu, t_comision;
      exit when t_cursor%notfound;

      t_index       := t_index + 1;
      t_venit_lunar := t_venit_lunar + t_salariu + t_salariu * t_comision;

      dbms_output.put_line(t_index || '. ' || t_nume || ' ' ||
      t_salariu || ' (' ||
      round(100 * (t_salariu + t_salariu * t_comision) / t_suma_totala, 2)
      || '%)');
   end loop;

   t_nr_angajati_total := t_nr_angajati_total + t_index;
```

```
        t_venit_lunar_total := t_venit_lunar_total + t_venit_lunar;

        dbms_output.put('Numar angajati: ');
        if t_index = 0 then
            dbms_output.put_line('Nu exista angajati');
        elsif t_index = 1 then
            dbms_output.put_line('Un singur angajat');
        else
            dbms_output.put_line(t_index || ' angajati');
        end if;

        dbms_output.put_line('Venit lunar total: ' || t_venit_lunar);
        dbms_output.put('Venit lunar mediu: ');
        if t_index != 0 then
            dbms_output.put_line(round(t_venit_lunar / t_index, 3));
        else
            dbms_output.put_line('0');
        end if;
        dbms_output.new_line;
    end loop;
    close t_info;

    dbms_output.put_line('Numar total de angajati: ' || t_nr_angajati_total);
    dbms_output.put_line('Venit lunar total: ' || t_venit_lunar_total);
end;
```


SGBD-231
Worksheet
Query Builder
cursor t_info is
select j.job_title,
cursor (
select last_name as nume, salary as salariu, nvl(commission_pct, 0) as comision
from employees e
where e.job_id = j.job_id)
from jobs j;
t_cursor refcursor;
t_job jobs.job_title%type;
t_nume employees.last_name%type;
t_salariu employees.salary%type;
t_comision employees.commission_pct%type;
t_index number;
Dbms Output
Buffer Size: 20000
SGBD-231
JOB - Sales Manager
1. Russell 14000 (2.55%)
2. Partners 13500 (2.28%)
3. Errazuriz 12000 (2.03%)
4. Cambrault 11000 (1.86%)
5. Zlotkey 10500 (1.64%)
Numar angajati: 5 angajati
Venit lunar total: 79650
Venit lunar mediu: 15930
JOB - Sales Representative
1. Tucker 10000 (1.69%)
2. Bernstein 9500 (1.54%)
3. Hall 9000 (1.46%)
4. Olsen 8000 (1.25%)

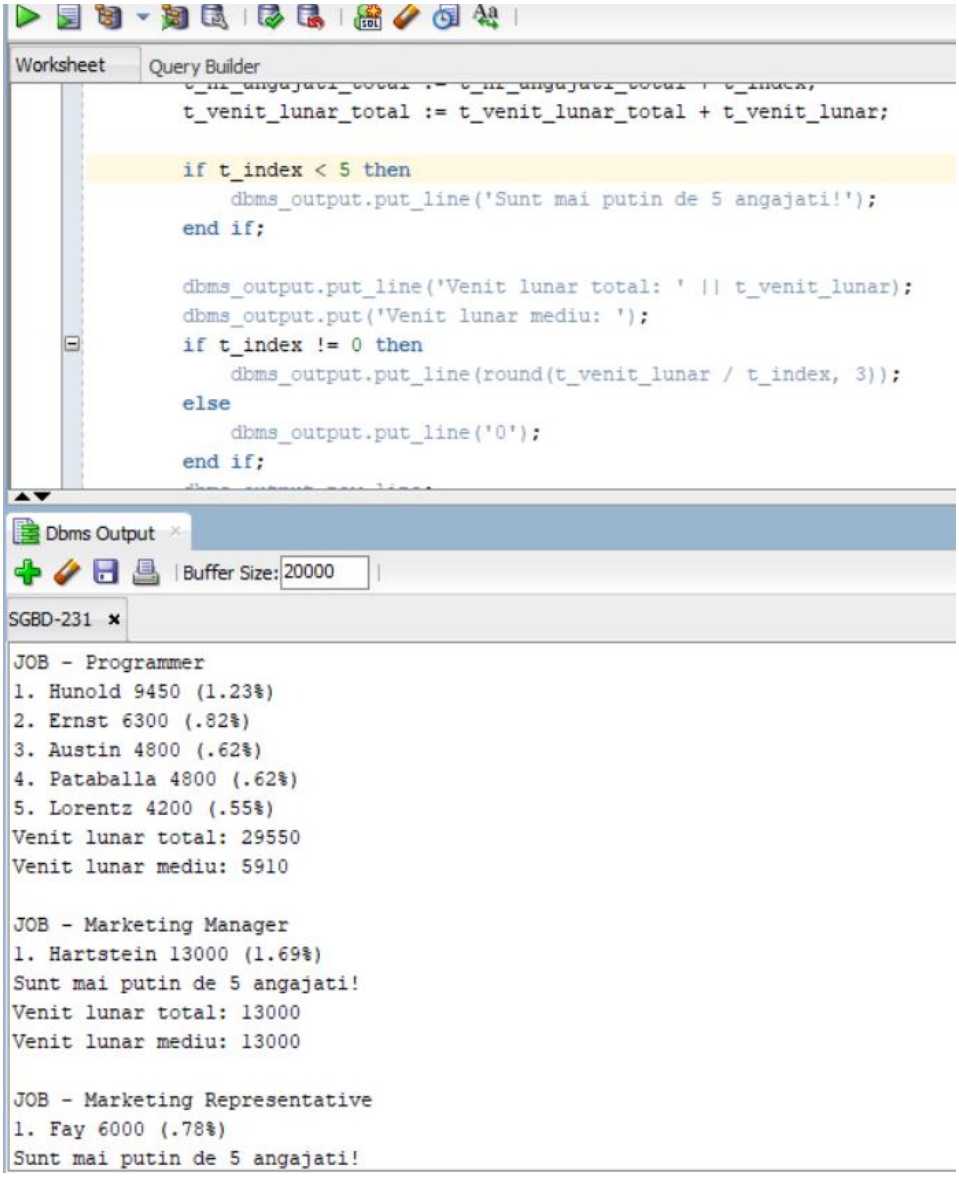
E4: in cursorul imbricat, am ordonat angajatii descrescator dupa salariu. Am modificat cerinta de "exit" pentru subcursor: se va iesi din LOOP mai devreme, daca se ajunge la al 5-lea angajat.

Pentru a verifica daca sunt mai putin de 5 angajati, am verificat variabila auxiliara folosita pentru numarul de angajati pentru fiecare job intr-un "IF". In cazul in care sunt mai putin de 5, se afiseaza un mesaj corespunzator.

```
declare
   type refcursor is ref cursor;
   cursor t_info is
      select j.job_title,
      cursor (
         select last_name as nume, salary as salariu, nvl(commission_pct, 0) as comision
         from employees e
         where e.job_id = j.job_id
         order by salariu desc)
      from jobs j;

   t_cursor refcursor;
   t_job jobs.job_title%type;
   t_nume employees.last_name%type;
   t_salariu employees.salary%type;
   t_comision employees.commission_pct%type;

   t_index       number;
   t_nr_angajati number;
   t_venit_lunar number;

   t_nr_angajati_total number := 0;
   t_venit_lunar_total number := 0;
```

```
    t_suma_totala number;
begin
    select sum(salary + salary * nvl(commission_pct, 0))
    into t_suma_totala
    from employees;
    dbms_output.put_line('Suma totala alocata lunar: ' || t_suma_totala);
    dbms_output.new_line;

    open t_info;
    loop
        fetch t_info into t_job, t_cursor;
        exit when t_info%notfound;
        dbms_output.put_line('JOB - ' || t_job);

        t_index       := 0;
        t_venit_lunar := 0;
        loop
            fetch t_cursor into t_nume, t_salariu, t_comision;
            exit when t_cursor%notfound or t_index = 5;

            t_index       := t_index + 1;
            t_venit_lunar := t_venit_lunar + t_salariu + t_salariu * t_comision;

            dbms_output.put_line(t_index || '. ' || t_nume || ' ' ||
            t_salariu || ' (' ||
            round(100 * (t_salariu + t_salariu * t_comision) / t_suma_totala, 2)
            || '%)');
```

```
        end loop;

        t_nr_angajati_total := t_nr_angajati_total + t_index;
        t_venit_lunar_total := t_venit_lunar_total + t_venit_lunar;

        if t_index < 5 then
            dbms_output.put_line('Sunt mai putin de 5 angajati!');
        end if;

        dbms_output.put_line('Venit lunar total: ' || t_venit_lunar);
        dbms_output.put('Venit lunar mediu: ');
        if t_index != 0 then
            dbms_output.put_line(round(t_venit_lunar / t_index, 3));
        else
            dbms_output.put_line('0');
        end if;
        dbms_output.new_line;
    end loop;
    close t_info;

    dbms_output.put_line('Numar total de angajati: ' || t_nr_angajati_total);
    dbms_output.put_line('Venit lunar total: ' || t_venit_lunar_total);
end;
```


Worksheet
Query Builder
t_venit_lunar_total := t_venit_lunar_total + t_venit_lunar;
if t_index < 5 then
dbms_output.put_line('Sunt mai putin de 5 angajati!');
end if;
dbms_output.put_line('Venit lunar total: ' || t_venit_lunar);
dbms_output.put('Venit lunar mediu: ');
if t_index != 0 then
dbms_output.put_line(round(t_venit_lunar / t_index, 3));
else
dbms_output.put_line('0');
end if;
Dbms Output
Buffer Size: 20000
SGBD-231
JOB - Programmer
1. Hunold 9450 (1.23%)
2. Ernst 6300 (.82%)
3. Austin 4800 (.62%)
4. Pataballa 4800 (.62%)
5. Lorentz 4200 (.55%)
Venit lunar total: 29550
Venit lunar mediu: 5910
JOB - Marketing Manager
1. Hartstein 13000 (1.69%)
Sunt mai putin de 5 angajati!
Venit lunar total: 13000
Venit lunar mediu: 13000
JOB - Marketing Representative
1. Fay 6000 (.78%)
Sunt mai putin de 5 angajati!