Enescu Irina Stefania
Grupa 233

TEMA LABORATOR 8
(Laborator PLSQL 3)

1. Pentru fiecare job (titlu – care va fi afișat o singură dată) obțineți lista angajaților (nume și salariu) care lucrează în prezent pe jobul respectiv. Tratați cazul în care nu există angajați care să lucreze în prezent pe un anumit job. Rezolvați problema folosind:

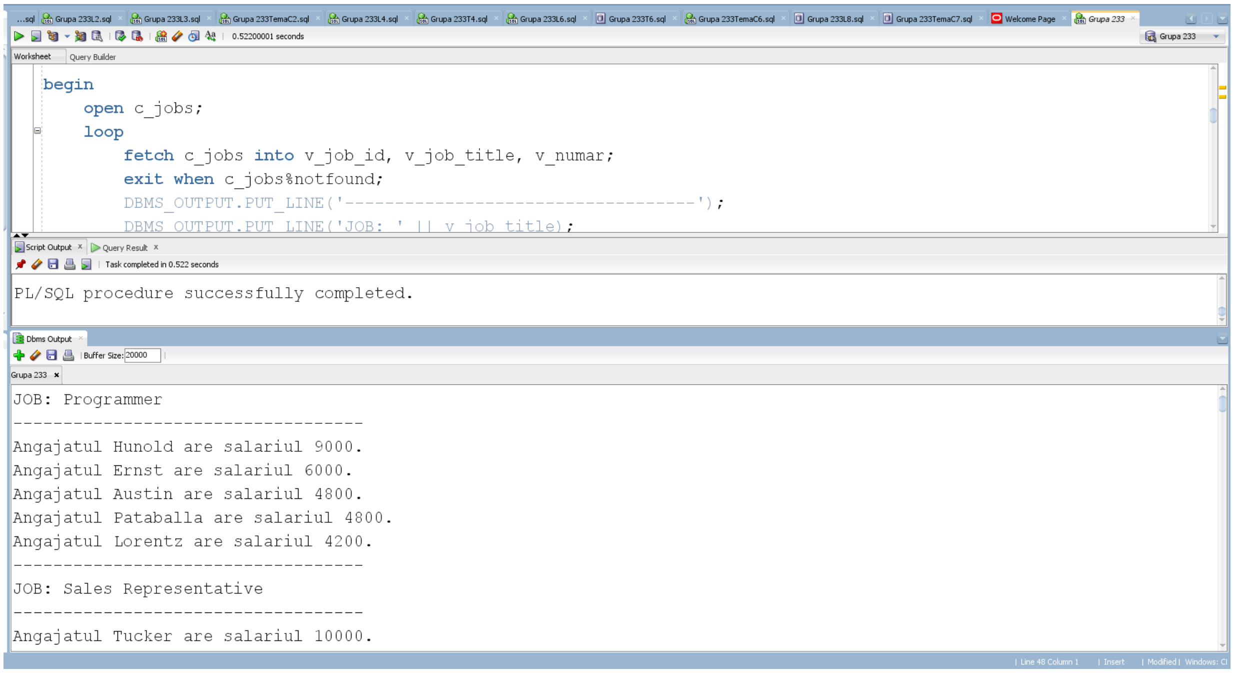
--> Am luat in considerare salariul din tabel, nu salariu + comision

a. cursoare clasice

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_job_id_emp employees.job_id%type;
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, job_id
      from employees;

begin
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      else
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when c_employees%notfound;
            if v_job_id = v_job_id_emp then
               DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_nume || ' are salariul ' || v_salariu || '.');
            end if;
         end loop;
```

```
          close c_employees;
        end if;
    end loop;
    close c_jobs;
end;
/
```



```
begin
    open c_jobs;
    loop
        fetch c_jobs into v_job_id, v_job_title, v_numar;
        exit when c_jobs%notfound;
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
```

```
PL/SQL procedure successfully completed.
```

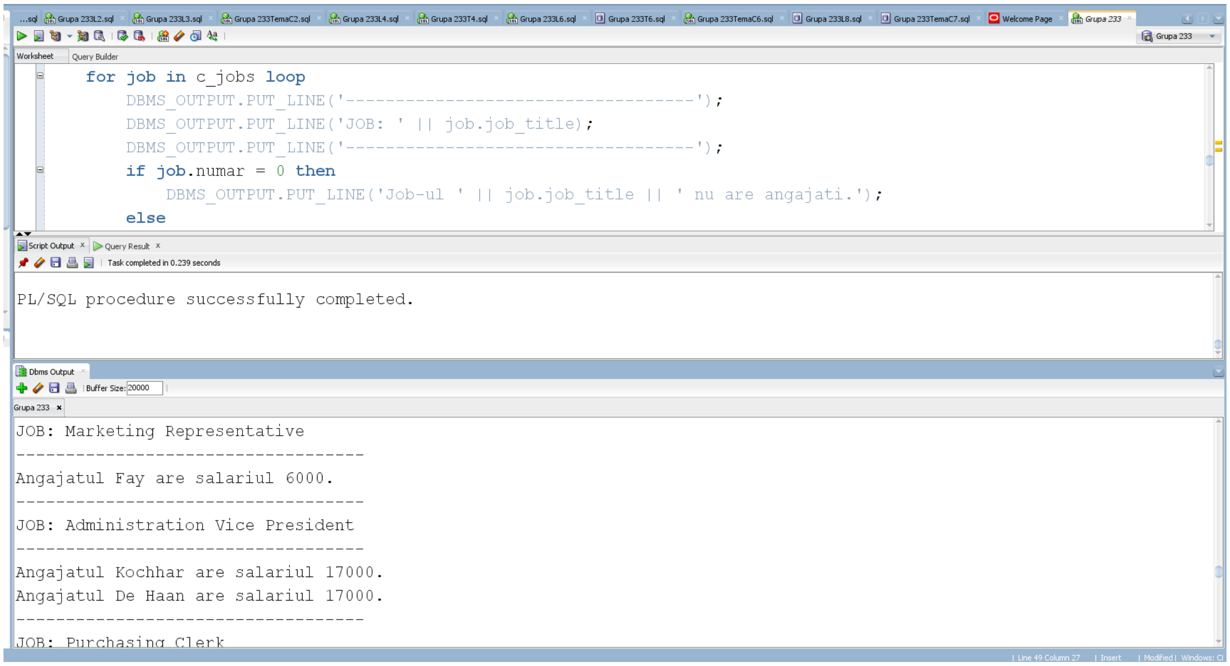
```
JOB: Programmer
-----------------------------------
Angajatul Hunold are salariul 9000.
Angajatul Ernst are salariul 6000.
Angajatul Austin are salariul 4800.
Angajatul Pataballa are salariul 4800.
Angajatul Lorentz are salariul 4200.
-----------------------------------
JOB: Sales Representative
-----------------------------------
Angajatul Tucker are salariul 10000.
```

b. ciclu cursoare

```
declare
    cursor c_jobs is
        select j.job_id, j.job_title, count(e.employee_id) as numar
        from employees e right join jobs j on (j.job_id = e.job_id)
        group by j.job_id, job_title;
    cursor c_employees is
        select last_name, salary, job_id
        from employees;

begin
    for job in c_jobs loop
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        DBMS_OUTPUT.PUT_LINE('JOB: ' || job.job_title);
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        if job.numar = 0 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || job.job_title || ' nu are angajati.');
        else
            for emp in c_employees loop
                if job.job_id = emp.job_id then
```

```
         DBMS_OUTPUT.PUT_LINE('Angajatul ' || emp.last_name || ' are salariul ' || emp.salary || '.');
        end if;
      end loop;
    end if;
  end loop;
end;
/
```



```
    for job in c_jobs loop
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        DBMS_OUTPUT.PUT_LINE('JOB: ' || job.job_title);
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        if job.numar = 0 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || job.job_title || ' nu are angajati.');
        else
```

Task completed in 0.239 seconds

```
PL/SQL procedure successfully completed.
```

Buffer Size: 20000

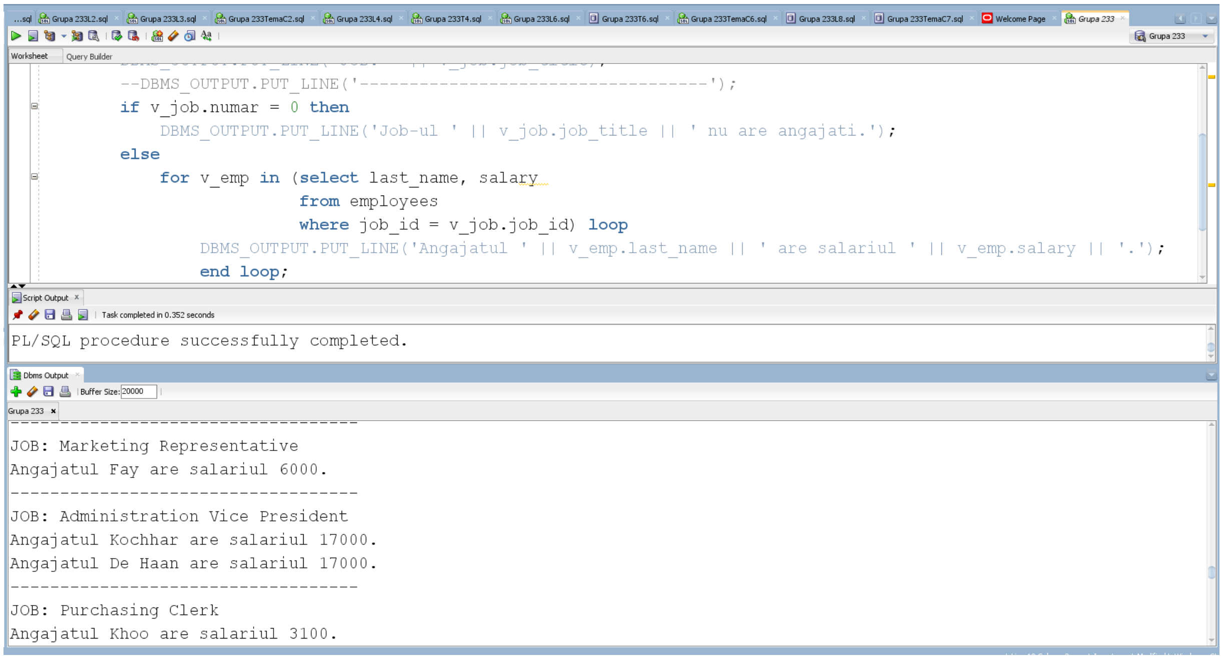
Grupa 233

```
JOB: Marketing Representative
-----------------------------------
Angajatul Fay are salariul 6000.
-----------------------------------
JOB: Administration Vice President
-----------------------------------
Angajatul Kochhar are salariul 17000.
Angajatul De Haan are salariul 17000.
-----------------------------------
JOB: Purchasing Clerk
```

c. ciclu cursoare cu subcereri

```
begin
  for v_job in (select j.job_id, j.job_title, count(e.employee_id) as numar
         from employees e right join jobs j on (j.job_id = e.job_id)
         group by j.job_id, job_title) loop
    DBMS_OUTPUT.PUT_LINE('-----------------------------------');
    DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job.job_title);
    --DBMS_OUTPUT.PUT_LINE('-----------------------------------');
    if v_job.numar = 0 then
      DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job.job_title || ' nu are angajati.');
    else
      for v_emp in (select last_name, salary
             from employees
             where job_id = v_job.job_id) loop
     DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_emp.last_name || ' are salariul ' || v_emp.salary || '.');
        end loop;
    end if;
  end loop;
end;
```

/



```
        --DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        if v_job.numar = 0 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job.job_title || ' nu are angajati.');
        else
            for v_emp in (select last_name, salary
                          from employees
                          where job_id = v_job.job_id) loop
                DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_emp.last_name || ' are salariul ' || v_emp.salary || '.');
                end loop;
```

Task completed in 0.352 seconds

```
PL/SQL procedure successfully completed.
```

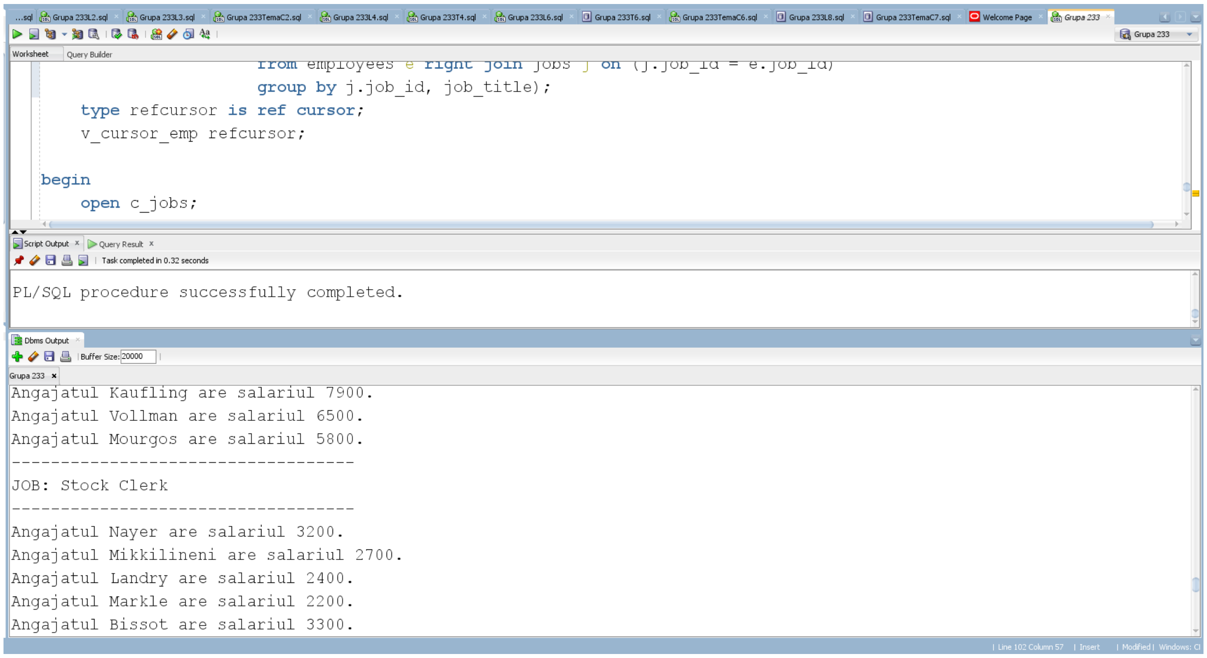
```
-----------------------------------
JOB: Marketing Representative
Angajatul Fay are salariul 6000.
-----------------------------------
JOB: Administration Vice President
Angajatul Kochhar are salariul 17000.
Angajatul De Haan are salariul 17000.
-----------------------------------
JOB: Purchasing Clerk
Angajatul Khoo are salariul 3100.
```

d. expresii cursor

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   cursor c_jobs is (select j.job_id, j.job_title, count(e.employee_id) as numar, cursor (select last_name, salary
                                                      from employees ee
                                                      where ee.job_id = j.job_id)
                from employees e right join jobs j on (j.job_id = e.job_id)
                group by j.job_id, job_title);
   type refcursor is ref cursor;
   v_cursor_emp refcursor;

begin
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar, v_cursor_emp;
      exit when c_jobs%notfound;
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      else
         loop
            fetch v_cursor_emp into v_nume, v_salariu;
```

```
                exit when v_cursor_emp%notfound;
                DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_nume || ' are salariul ' || v_salariu || '.');
            end loop;
        end if;
    end loop;
end;
/
```



```
                        from employees e right join jobs j on (j.job_id = e.job_id)
                        group by j.job_id, job_title);
    type refcursor is ref cursor;
    v_cursor_emp refcursor;

begin
    open c_jobs;
```

```
PL/SQL procedure successfully completed.
```

```
Angajatul Kaufling are salariul 7900.
Angajatul Vollman are salariul 6500.
Angajatul Mourgos are salariul 5800.
-----------------------------------
JOB: Stock Clerk
-----------------------------------
Angajatul Nayer are salariul 3200.
Angajatul Mikkilineni are salariul 2700.
Angajatul Landry are salariul 2400.
Angajatul Markle are salariul 2200.
Angajatul Bissot are salariul 3300.
```

2. Modificați exercițiul anterior astfel încât să obțineți și următoarele informații:

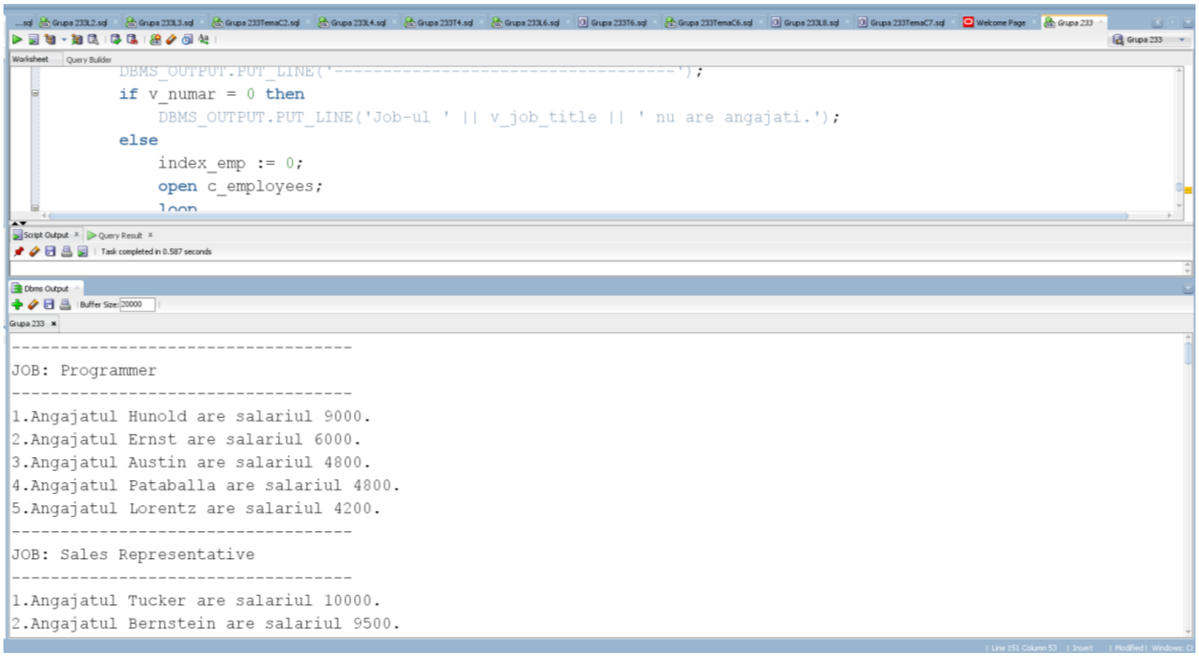
- un număr de ordine pentru fiecare angajat care va fi resetat pentru fiecare job

```
declare
    v_job_id jobs.job_id%type;
    v_job_title jobs.job_title%type;
    v_numar number(4);
    v_nume employees.last_name%type;
    v_salariu employees.salary%type;
    v_job_id_emp employees.job_id%type;
    index_emp number(4);
    cursor c_jobs is
        select j.job_id, j.job_title, count(e.employee_id) as numar
        from employees e right join jobs j on (j.job_id = e.job_id)
        group by j.job_id, job_title;
    cursor c_employees is
        select last_name, salary, job_id
        from employees;

begin
```

```
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      else
         index_emp := 0;
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when c_employees%notfound;
            if v_job_id = v_job_id_emp then
               index_emp := index_emp + 1;
               DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| '.');
            end if;
         end loop;
         close c_employees;
      end if;
   end loop;
   close c_jobs;
end;
/
```

--> Am luat in considerare salariul din tabel, nu salariu + comision



```
        DBMS_OUTPUT.PUT_LINE('-----------------------------------');
        if v_numar = 0 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
        else
            index_emp := 0;
            open c_employees;
            loop
```

Task completed in 0.587 seconds

```
-----------------------------------
JOB: Programmer
-----------------------------------
1.Angajatul Hunold are salariul 9000.
2.Angajatul Ernst are salariul 6000.
3.Angajatul Austin are salariul 4800.
4.Angajatul Pataballa are salariul 4800.
5.Angajatul Lorentz are salariul 4200.
-----------------------------------
JOB: Sales Representative
-----------------------------------
1.Angajatul Tucker are salariul 10000.
2.Angajatul Bernstein are salariul 9500.
```

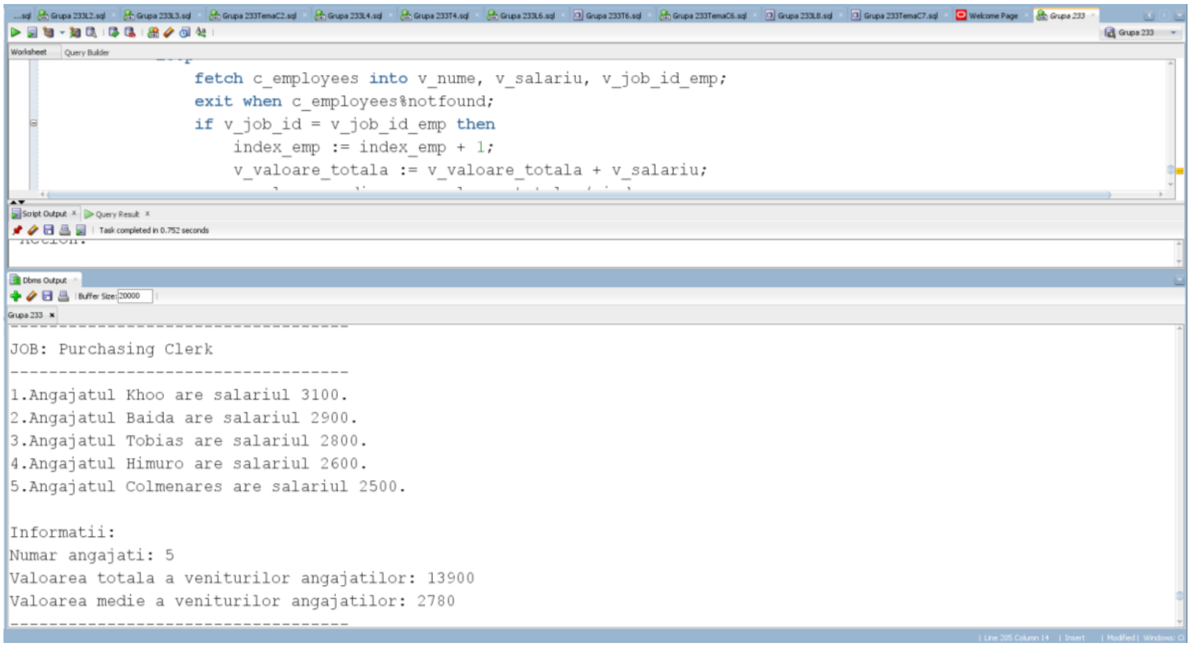
- pentru fiecare job
    - o numărul de angajați
    - o valoarea lunară a veniturilor angajaților
    - o valoarea medie a veniturilor angajaților

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_job_id_emp employees.job_id%type;
   index_emp number(4);
   v_valoare_totala number(7);
   v_valoare_medie number(7);
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, job_id
      from employees;

begin
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
         DBMS_OUTPUT.PUT_LINE('Informatii: ');
         DBMS_OUTPUT.PUT_LINE('Numar angajati: 0');
         DBMS_OUTPUT.PUT_LINE('Valoarea totala a veniturilor angajatilor: 0');
         DBMS_OUTPUT.PUT_LINE('Valoarea medie a veniturilor angajatilor: 0');
      else
         index_emp := 0;
         v_valoare_totala := 0;
         v_valoare_medie := 0;
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when c_employees%notfound;
            if v_job_id = v_job_id_emp then
               index_emp := index_emp + 1;
               v_valoare_totala := v_valoare_totala + v_salariu;
```

```
                v_valoare_medie := v_valoare_totala / index_emp;
                DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| '.');
              end if;
           end loop;
           close c_employees;
           DBMS_OUTPUT.PUT_LINE('');
           DBMS_OUTPUT.PUT_LINE('Informatii: ');
           DBMS_OUTPUT.PUT_LINE('Numar angajati: ' || index_emp);
           DBMS_OUTPUT.PUT_LINE('Valoarea totala a veniturilor angajatilor: ' || v_valoare_totala);
           DBMS_OUTPUT.PUT_LINE('Valoarea medie a veniturilor angajatilor: ' || v_valoare_medie);
        end if;
   end loop;
   close c_jobs;
end;
/
```

--> Am luat in considerare salariul din tabel, nu salariu + comision



```
                fetch c_employees into v_nume, v_salariu, v_job_id_emp;
                exit when c_employees%notfound;
                if v_job_id = v_job_id_emp then
                    index_emp := index_emp + 1;
                    v_valoare_totala := v_valoare_totala + v_salariu;
```

Task completed in 0.752 seconds

```
-----------------------------------
JOB: Purchasing Clerk
-----------------------------------
1.Angajatul Khoo are salariul 3100.
2.Angajatul Baida are salariul 2900.
3.Angajatul Tobias are salariul 2800.
4.Angajatul Himuro are salariul 2600.
5.Angajatul Colmenares are salariul 2500.

Informatii:
Numar angajati: 5
Valoarea totala a veniturilor angajatilor: 13900
Valoarea medie a veniturilor angajatilor: 2780
-----------------------------------
```

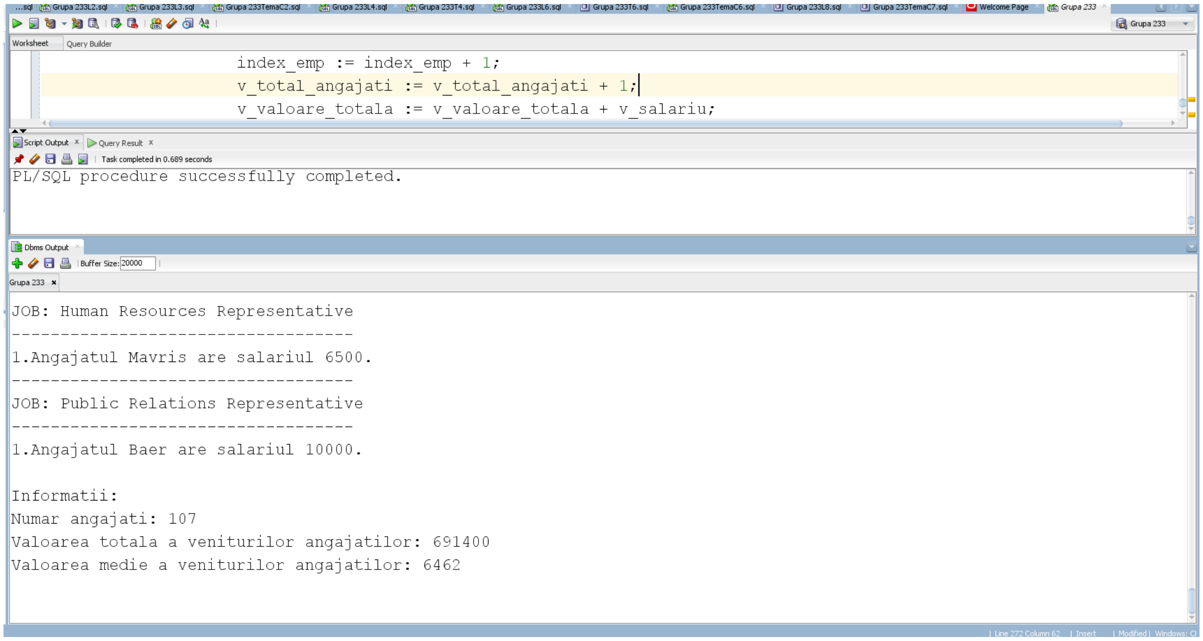
- indiferent job
    - o numărul total de angajați
    - o valoarea totală lunară a veniturilor angajaților
    - o valoarea medie a veniturilor angajaților

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_job_id_emp employees.job_id%type;
   index_emp number(4);
   v_valoare_totala number(7);
   v_valoare_medie number(7);
   v_total_angajati number(7);
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, job_id
      from employees;

begin
   v_total_angajati := 0;
   v_valoare_medie := 0;
   v_valoare_totala := 0;
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      else
         index_emp := 0;
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when c_employees%notfound;
            if v_job_id = v_job_id_emp then
               index_emp := index_emp + 1;
               v_total_angajati := v_total_angajati + 1;
               v_valoare_totala := v_valoare_totala + v_salariu;
               DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| '.');
            end if;
         end loop;
         close c_employees;
      end if;
   end loop;
   v_valoare_medie := v_valoare_totala / v_total_angajati;
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('Informatii: ');
   DBMS_OUTPUT.PUT_LINE('Numar angajati: ' || v_total_angajati);
```

```
    DBMS_OUTPUT.PUT_LINE('Valoarea totala a veniturilor angajatilor: ' || v_valoare_totala);
    DBMS_OUTPUT.PUT_LINE('Valoarea medie a veniturilor angajatilor: ' || v_valoare_medie);
    close c_jobs;
end;
/
```

--> Am luat in considerare salariul din tabel, nu salariu + comision



```
                    index_emp := index_emp + 1;
                    v_total_angajati := v_total_angajati + 1;
                    v_valoare_totala := v_valoare_totala + v_salariu;
```

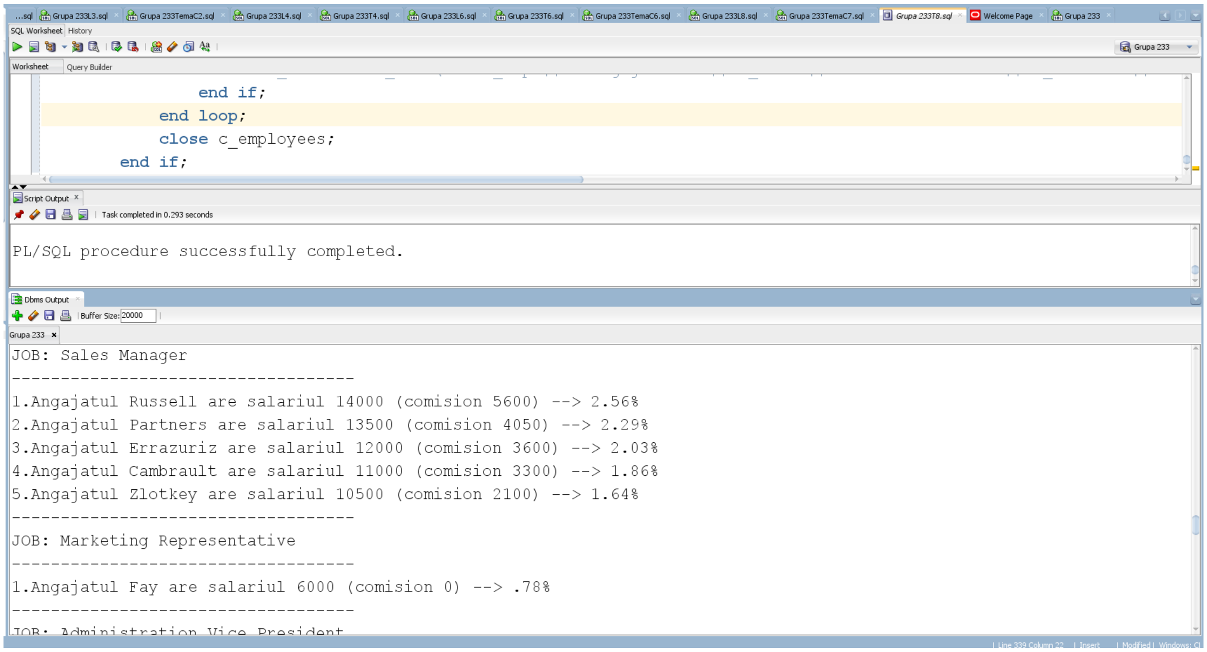
Task completed in 0.689 seconds

```
PL/SQL procedure successfully completed.
```

```
JOB: Human Resources Representative
-----------------------------------
1.Angajatul Mavris are salariul 6500.
-----------------------------------
JOB: Public Relations Representative
-----------------------------------
1.Angajatul Baer are salariul 10000.

Informatii:
Numar angajati: 107
Valoarea totala a veniturilor angajatilor: 691400
Valoarea medie a veniturilor angajatilor: 6462
```

3. Modificați exercițiul anterior astfel încât să obțineți suma totală alocată lunar pentru plata salariilor și a comisioanelor tuturor angajaților, iar pentru fiecare angajat cât la sută din această sumă câștigă lunar.



```
                end if;
            end loop;
            close c_employees;
        end if;
```

Task completed in 0.293 seconds

```
PL/SQL procedure successfully completed.
```

```
JOB: Sales Manager
-----------------------------------
1.Angajatul Russell are salariul 14000 (comision 5600) --> 2.56%
2.Angajatul Partners are salariul 13500 (comision 4050) --> 2.29%
3.Angajatul Errazuriz are salariul 12000 (comision 3600) --> 2.03%
4.Angajatul Cambrault are salariul 11000 (comision 3300) --> 1.86%
5.Angajatul Zlotkey are salariul 10500 (comision 2100) --> 1.64%
-----------------------------------
JOB: Marketing Representative
-----------------------------------
1.Angajatul Fay are salariul 6000 (comision 0) --> .78%
-----------------------------------
JOB: Administration Vice President
```

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_comision employees.salary%type;
   v_job_id_emp employees.job_id%type;
   index_emp number(4);
   v_total_angajati number(7);
   v_valoare_totala_calcul number(7) := 0;
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, commission_pct*salary, job_id
      from employees;

begin
   v_total_angajati := 0;
   open c_employees;
   loop
      fetch c_employees into v_nume, v_salariu, v_comision, v_job_id_emp;
      exit when c_employees%notfound;
      v_valoare_totala_calcul := v_valoare_totala_calcul + v_salariu + nvl(v_comision, 0);
   end loop;
   close c_employees;
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      else
         index_emp := 0;
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_comision, v_job_id_emp;
            exit when c_employees%notfound;
            if v_job_id = v_job_id_emp then
               index_emp := index_emp + 1;
               v_total_angajati := v_total_angajati + 1;
               DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| ' (comision ' || nvl(v_comision, 0) || ') --> ' || trunc((v_salariu + nvl(v_comision,
0))*100/v_valoare_totala_calcul, 2) || '%');
            end if;
         end loop;
         close c_employees;
      end if;
   end loop;
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('Informatii: ');
   DBMS_OUTPUT.PUT_LINE('Numar angajati: ' || v_total_angajati);
   DBMS_OUTPUT.PUT_LINE('Suma totala alocată lunar pentru plata salariilor si a comisioanelor tuturor
angajaților: ' || v_valoare_totala_calcul);
   close c_jobs;
end;
/
```

--> Am facut trunc la procent ca sa fie mai usor de citit cu doua zecimale.

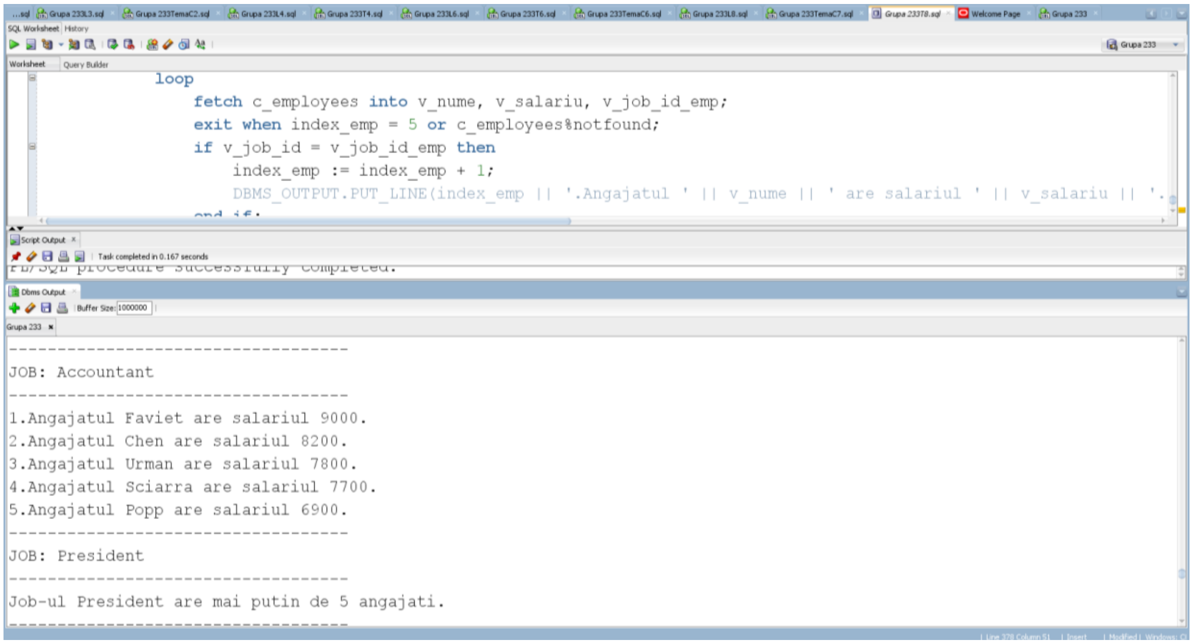
4. Modificați exercițiul anterior astfel încât să obțineți pentru fiecare job primii 5 angajați care câștigă cel mai mare salariu lunar. Specificați dacă pentru un job sunt mai puțin de 5 angajați.

```
declare
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_job_id_emp employees.job_id%type;
   index_emp number(4);
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, job_id
      from employees
      order by salary desc;

begin
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('------------------------------------');
      if v_numar = 0 then
```

```
          DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
        elsif v_numar < 5 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' are mai putin de 5 angajati.');
        else
          index_emp := 0;
          open c_employees;
          loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when index_emp = 5 or c_employees%notfound;
            if v_job_id = v_job_id_emp then
              index_emp := index_emp + 1;
              DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| '.');
            end if;
          end loop;
          close c_employees;
        end if;
    end loop;
    close c_jobs;
end;
/
```

--> Am luat in considerare salariul din tabel, nu salariu + comision.



```
                loop
                    fetch c_employees into v_nume, v_salariu, v_job_id_emp;
                    exit when index_emp = 5 or c_employees%notfound;
                    if v_job_id = v_job_id_emp then
                        index_emp := index_emp + 1;
                        DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu || '.
                    end if;
```

```
PL/SQL procedure successfully completed.
```

```
-----------------------------------
JOB: Accountant
-----------------------------------
1.Angajatul Faviet are salariul 9000.
2.Angajatul Chen are salariul 8200.
3.Angajatul Urman are salariul 7800.
4.Angajatul Sciarra are salariul 7700.
5.Angajatul Popp are salariul 6900.
-----------------------------------
JOB: President
-----------------------------------
Job-ul President are mai putin de 5 angajati.
-----------------------------------
```

5. Modificați exercițiul anterior astfel încât să obțineți pentru fiecare job top 5 angajați. Dacă există mai mulți angajați care respectă criteriul de selecție care au același salariu, atunci aceștia vor ocupa aceeași poziție în top 5.

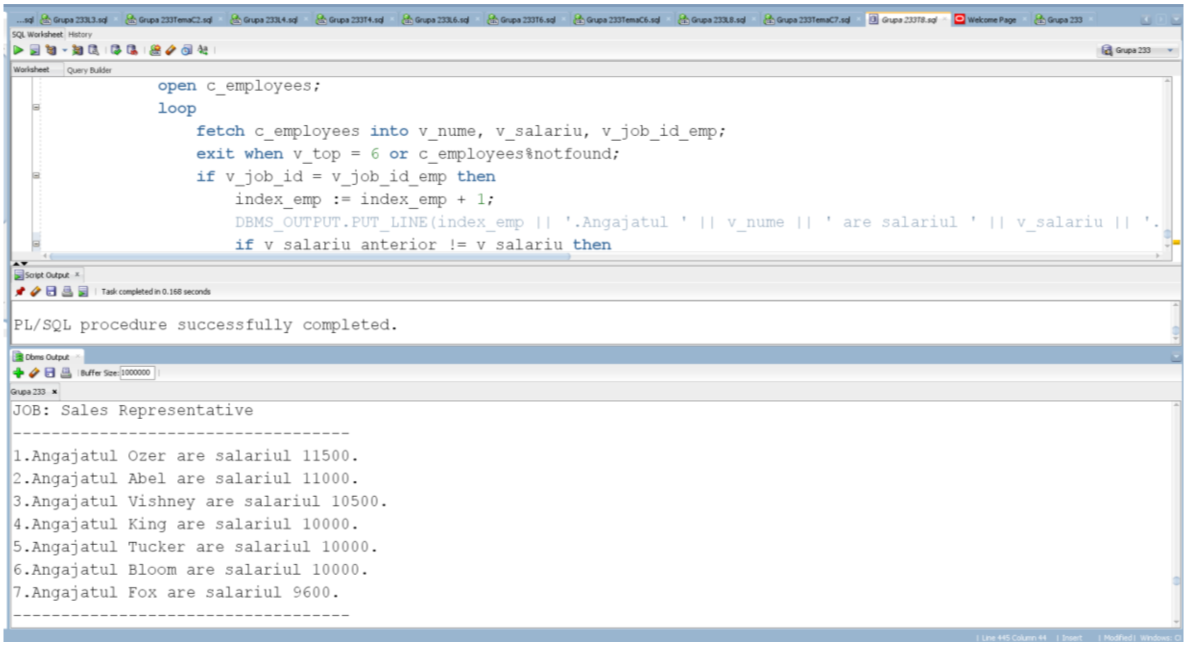
```
declare
```

```
   v_job_id jobs.job_id%type;
   v_job_title jobs.job_title%type;
   v_numar number(4);
   v_nume employees.last_name%type;
   v_salariu employees.salary%type;
   v_job_id_emp employees.job_id%type;
   index_emp number(4);
   v_top number(4);
   v_salariu_anterior employees.salary%type;
   cursor c_jobs is
      select j.job_id, j.job_title, count(e.employee_id) as numar
      from employees e right join jobs j on (j.job_id = e.job_id)
      group by j.job_id, job_title;
   cursor c_employees is
      select last_name, salary, job_id
      from employees
      order by salary desc;

begin
   open c_jobs;
   loop
      fetch c_jobs into v_job_id, v_job_title, v_numar;
      exit when c_jobs%notfound;
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      DBMS_OUTPUT.PUT_LINE('JOB: ' || v_job_title);
      DBMS_OUTPUT.PUT_LINE('-----------------------------------');
      if v_numar = 0 then
         DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' nu are angajati.');
      elsif v_numar < 5 then
            DBMS_OUTPUT.PUT_LINE('Job-ul ' || v_job_title || ' are mai putin de 5 angajati.');
      else
         index_emp := 0;
         v_top := 1;
         v_salariu_anterior := 0;
         open c_employees;
         loop
            fetch c_employees into v_nume, v_salariu, v_job_id_emp;
            exit when v_top = 6 or c_employees%notfound;
            if v_job_id = v_job_id_emp then
               index_emp := index_emp + 1;
               DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu
|| '.');
               if v_salariu_anterior != v_salariu then
                  v_salariu_anterior := v_salariu;
                  v_top := v_top + 1;
               end if;
            end if;
         end loop;
```

```
            close c_employees;
        end if;
    end loop;
    close c_jobs;
end;
/
```

--> Am luat in considerare salariul din tabel, nu salariu + comision



...sql | Grupa 233L3.sql | Grupa 233TemaC2.sql | Grupa 233L4.sql | Grupa 233T4.sql | Grupa 233L6.sql | Grupa 233T6.sql | Grupa 233TemaC6.sql | Grupa 233L8.sql | Grupa 233TemaC7.sql | Grupa 233T8.sql | Welcome Page | Grupa 233
SQL Worksheet
History
Grupa 233
Worksheet
Query Builder

```
            open c_employees;
            loop
                fetch c_employees into v_nume, v_salariu, v_job_id_emp;
                exit when v_top = 6 or c_employees%notfound;
                if v_job_id = v_job_id_emp then
                    index_emp := index_emp + 1;
                    DBMS_OUTPUT.PUT_LINE(index_emp || '.Angajatul ' || v_nume || ' are salariul ' || v_salariu || '.
                    if v_salariu_anterior != v_salariu then
```

Script Output
Task completed in 0.168 seconds

```
PL/SQL procedure successfully completed.
```

Dbms Output
Buffer Size: 1000000
Grupa 233

```
JOB: Sales Representative
-----------------------------------
1.Angajatul Ozer are salariul 11500.
2.Angajatul Abel are salariul 11000.
3.Angajatul Vishney are salariul 10500.
4.Angajatul King are salariul 10000.
5.Angajatul Tucker are salariul 10000.
6.Angajatul Bloom are salariul 10000.
7.Angajatul Fox are salariul 9600.
-----------------------------------
```