Tema PLSQL S12 - Oprea Tudor - 241

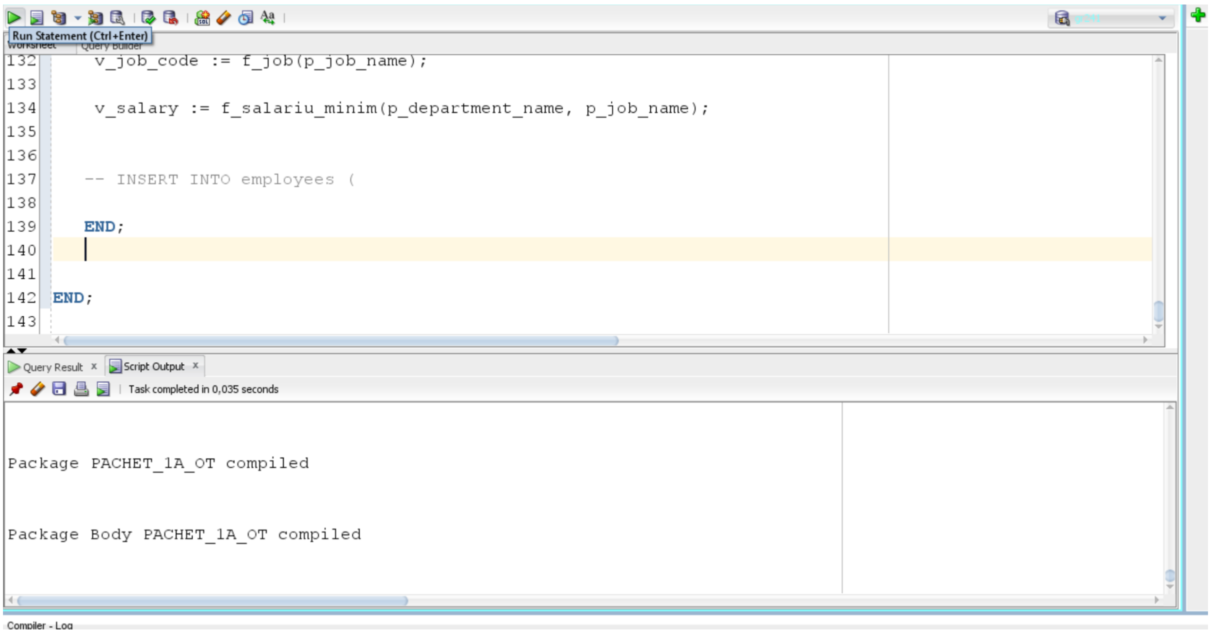
-- ex. 1 a)

```
CREATE OR REPLACE PACKAGE pachet_1a_OT AS

  -- generarea codului angajatului
  FUNCTION f_cod_angajat RETURN INTEGER;

  -- cel mai mic salariu pentru un departament si un job dat

  FUNCTION f_salariu_minim(p_department_name VARCHAR2, p_job_name VARCHAR2)
RETURN NUMBER;
  -- codul managerului dupa nume si prenume

  FUNCTION  f_manager(v_nume employees.first_name%TYPE, v_prenume
employees.last_name%TYPE)
  RETURN NUMBER;
  -- codul departamentului dupa nume
  FUNCTION  f_departament(v_nume_dep departments.department_name%TYPE)
       RETURN NUMBER;
  -- codul jobului dupa nume
   FUNCTION  f_job(v_nume_job VARCHAR)
       RETURN NUMBER;
  -- procedura de adaugare
  PROCEDURE adaugare_angajat(p_name VARCHAR2, p_surname VARCHAR2, p_phone
VARCHAR2, p_email VARCHAR2, p_manager_name VARCHAR2, p_manager_surname
VARCHAR2, p_department_name VARCHAR2, p_job_name VARCHAR2);
END ;

/

-- > corpul pachetului:
CREATE OR REPLACE PACKAGE BODY pachet_1a_OT AS

  FUNCTION f_cod_angajat RETURN INTEGER
  IS
    v_employee_code INTEGER;
  BEGIN
    SELECT MAX(employee_id) INTO v_employee_code FROM employees;
    RETURN v_employee_code+1;
    --luam cel mai mare id si returnam acel id+1
  END f_cod_angajat;

  FUNCTION f_salariu_minim(p_department_name VARCHAR2, p_job_name VARCHAR2)
RETURN NUMBER
  IS
    v_min NUMBER;
  BEGIN

    SELECT MIN(e.salary) INTO v_min
    FROM employees e
    INNER JOIN departments d ON e.department_id = d.department_id
    INNER JOIN jobs j ON e.job_id = j.job_id
    WHERE d.department_name = p_department_name AND j.job_title = p_job_name;

    RETURN v_min;
   END;



   FUNCTION  f_manager(v_nume employees.first_name%TYPE, v_prenume
employees.last_name%TYPE)
        RETURN NUMBER is
            v_id number;
        BEGIN
            select manager_id into v_id
            from employees
            where first_name = v_nume
            and last_name = v_prenume;

            return v_id;

        EXCEPTION
            when NO_DATA_FOUND then
                DBMS_OUTPUT.PUT_LINE('Nu exista manager/angajat');
            when TOO_MANY_ROWS then
                DBMS_OUTPUT.PUT_LINE('Exista mai multi angajati cu acest nume');
            when OTHERS then
                DBMS_OUTPUT.PUT_LINE('Alta eroare');
        END;


    FUNCTION  f_departament(v_nume_dep departments.department_name%TYPE)
        RETURN NUMBER IS
            v_cod_dep NUMBER;
        BEGIN
            select department_id into v_cod_dep
            from departments
            where department_name = v_nume_dep;
```

```
      return v_cod_dep;

    EXCEPTION
      when NO_DATA_FOUND then
        DBMS_OUTPUT.PUT_LINE('Nu exista manager/angajat');
      when TOO_MANY_ROWS then
        DBMS_OUTPUT.PUT_LINE('Exista mai multi angajati cu acest nume');
      when OTHERS then
        DBMS_OUTPUT.PUT_LINE('Alta eroare');
    END;

 FUNCTION  f_job(v_nume_job VARCHAR)
    RETURN NUMBER is
      v_job NUMBER;
    BEGIN
      select job_id into v_job
      from jobs
      where job_title = v_nume_job;

      return v_job;
    EXCEPTION
      when NO_DATA_FOUND then
        DBMS_OUTPUT.PUT_LINE('Nu exista acest job');
      when TOO_MANY_ROWS then
        DBMS_OUTPUT.PUT_LINE('Exista mai multe joburi cu acest nume');
      when OTHERS then
        DBMS_OUTPUT.PUT_LINE('Alta eroare');
    END;



 PROCEDURE adaugare_angajat(p_name VARCHAR2, p_surname VARCHAR2, p_phone
VARCHAR2, p_email VARCHAR2, p_manager_name VARCHAR2, p_manager_surname
VARCHAR2, p_department_name VARCHAR2, p_job_name VARCHAR2)
 IS
  v_employee_code INTEGER;
  v_manager_code INTEGER;
  v_department_code INTEGER;
  v_job_code INTEGER;
  v_salary NUMBER;
 BEGIN

  v_employee_code := f_cod_angajat();

  v_manager_code := f_manager(p_manager_name, p_manager_surname);

  v_department_code := f_departament(p_department_name);
```

```
   v_job_code := f_job(p_job_name);

   v_salary := f_salariu_minim(p_department_name, p_job_name);


  -- INSERT INTO employees (


  END;


END;
```

Nu este complet exercitiul 1 a, atat am apucat sa scriu.



```
    v_job_code := f_job(p_job_name);

    v_salary := f_salariu_minim(p_department_name, p_job_name);


   -- INSERT INTO employees (

   END;


END;
```

Task completed in 0,035 seconds

```
Package PACHET_1A_OT compiled


Package Body PACHET_1A_OT compiled
```