# Interfața pachetului hw\_lab10

|  |
| --- |
| /\*  3. (adaptat)  Definiți un pachet cu un subprogram care găsește salariul maxim al angajaților care lucrează într-un oraș,  un cursor care returnează angajații care au salariul cel puțin egal cu un salariu dat, și un cursor ajutător  care folosește subprogramul pentru a returna angajații cu salariul cel puțin egal cu salariul maxim dintr-un oraș.  \*/  CREATE OR REPLACE PACKAGE hw\_lab10 AS      CURSOR c\_angajati\_cu\_cel\_putin\_salariul(min\_sal NUMBER) RETURN angajat%ROWTYPE;      CURSOR c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(nume\_oras oras.nume%TYPE) RETURN angajat%ROWTYPE;      FUNCTION salariu\_maxim\_din\_oras(nume\_oras oras.nume%TYPE) RETURN NUMBER;      oras\_nu\_exista EXCEPTION;      PRAGMA EXCEPTION\_INIT (oras\_nu\_exista, -20003);  END hw\_lab10; |

# Implementarea pachetului hw\_lab10

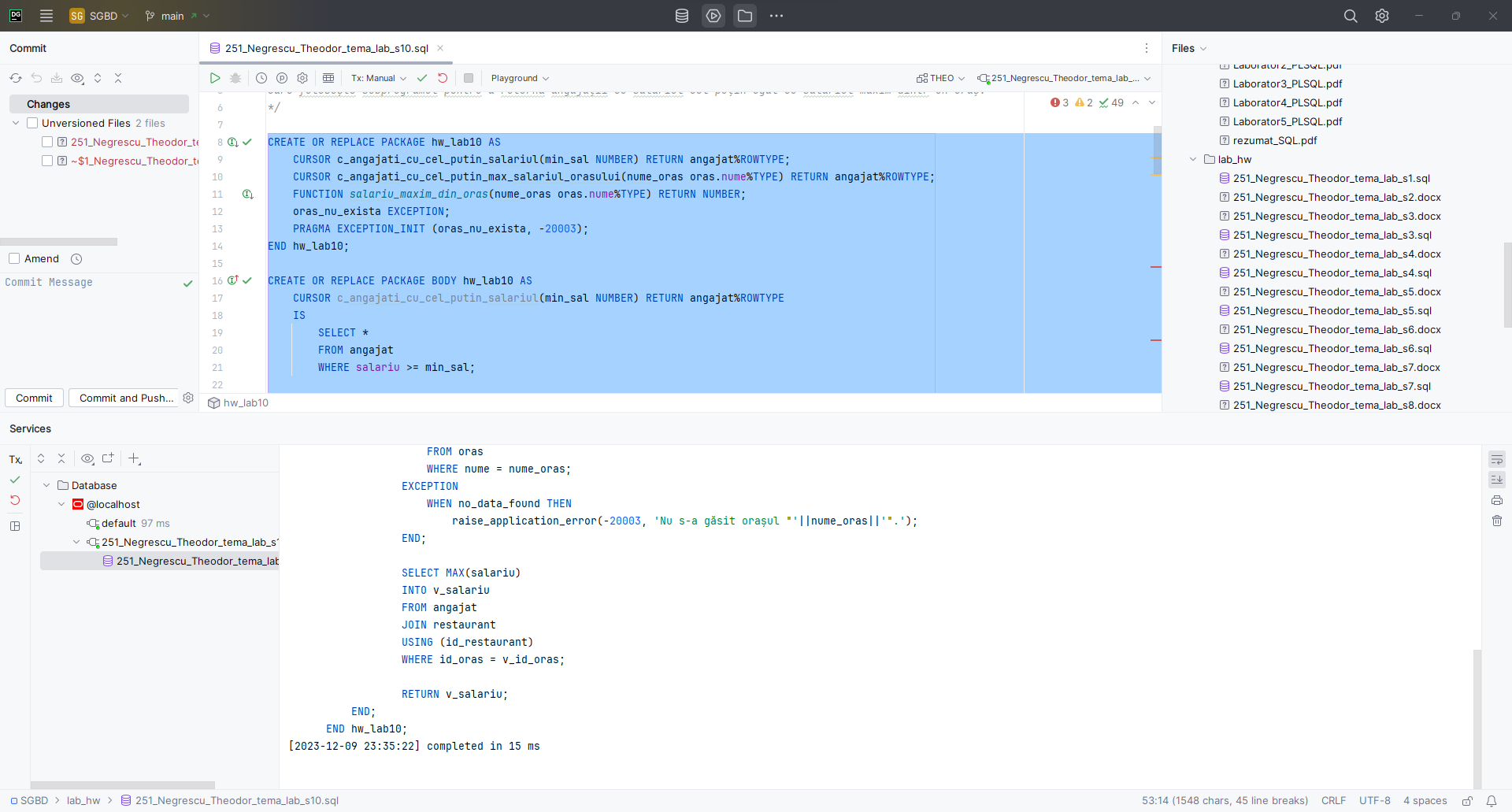
|  |
| --- |
| CREATE OR REPLACE PACKAGE BODY hw\_lab10 AS      CURSOR c\_angajati\_cu\_cel\_putin\_salariul(min\_sal NUMBER) RETURN angajat%ROWTYPE      IS          SELECT \*          FROM angajat          WHERE salariu >= min\_sal;      CURSOR c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(nume\_oras oras.nume%TYPE) RETURN angajat%ROWTYPE      IS          SELECT \*          FROM angajat          WHERE salariu >= salariu\_maxim\_din\_oras(nume\_oras);      FUNCTION salariu\_maxim\_din\_oras(nume\_oras oras.nume%TYPE) RETURN NUMBER      IS          v\_id\_oras oras.id\_oras%TYPE;          v\_salariu BINARY\_INTEGER;          BEGIN              BEGIN                  SELECT id\_oras                  INTO v\_id\_oras                  FROM oras                  WHERE nume = nume\_oras;              EXCEPTION                  WHEN no\_data\_found THEN                      raise\_application\_error(-20003, 'Nu s-a găsit orașul "'||nume\_oras||'".');              END;              SELECT MAX(salariu)              INTO v\_salariu              FROM angajat              JOIN restaurant              USING (id\_restaurant)              WHERE id\_oras = v\_id\_oras;              RETURN v\_salariu;      END;  END hw\_lab10; |

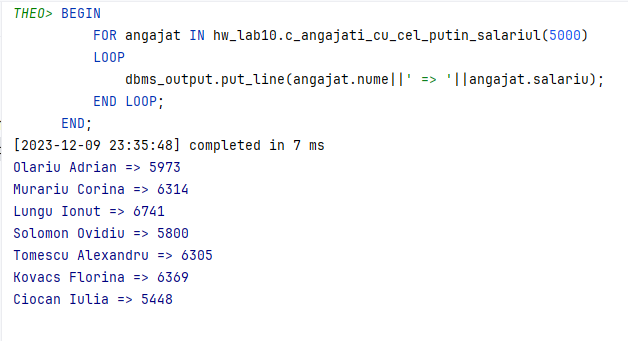
# Testarea pachetului hw\_lab10

|  |
| --- |
| BEGIN      FOR angajat IN hw\_lab10.c\_angajati\_cu\_cel\_putin\_salariul(5000)      LOOP          dbms\_output.put\_line(angajat.nume||' => '||angajat.salariu);      END LOOP;  END;  /  DECLARE      CURSOR c\_test\_data      IS          SELECT 1 as test\_nr, 'Cluj' AS nume FROM dual          UNION          SELECT rownum + 1 AS test\_nr, nume FROM oras;  BEGIN      FOR test IN c\_test\_data      LOOP          dbms\_output.PUT\_LINE('--- Test '||test.test\_nr||': '||test.nume||' ---');          dbms\_output.PUT('-- Salariul maxim în '||test.nume||': ');          BEGIN              dbms\_output.PUT\_LINE(hw\_lab10.salariu\_maxim\_din\_oras(test.nume));          EXCEPTION              WHEN hw\_lab10.oras\_nu\_exista THEN                  dbms\_output.PUT\_LINE(SQLERRM);          END;          dbms\_output.PUT\_LINE('-- Angajați cu salariul cel puțin egal cu maximul din acest oras:');          BEGIN              FOR angajat IN hw\_lab10.c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(test.nume)              LOOP                  dbms\_output.put\_line(angajat.nume||' '||angajat.salariu);              END LOOP;          EXCEPTION              WHEN hw\_lab10.oras\_nu\_exista THEN                  dbms\_output.PUT\_LINE(SQLERRM);          END;      END LOOP;  END;  / |

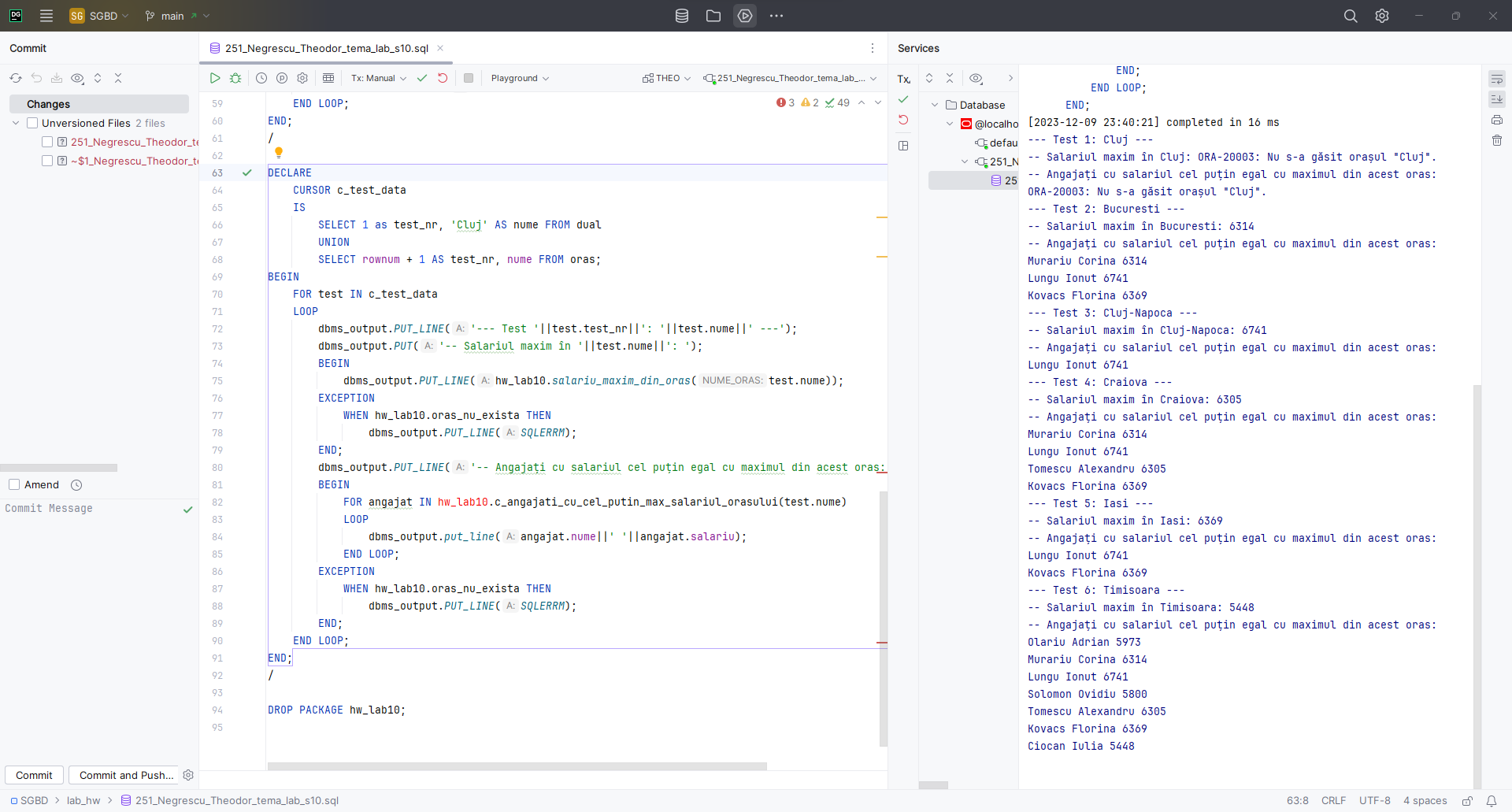
# Screenshot-uri

DataGrip afișează erori și warning despre cod neutilizat deoarece nu întelege folosirea unui cursor dintr-un pachet într-un FOR i.e. *FOR angajat IN hw\_lab10.c\_angajati\_cu\_cel\_putin\_salariul(5000)* afișează o eroare despre „hw\_lab10” negăsit deși codul funcționează.





Orașul Cluj nu există în baza de date și a fost introdus în c\_test\_data pentru a testa excepția oras\_nu\_exista



# Întregul fișier 251\_Negrescu\_Theodor\_tema\_lab\_s10.sql

|  |
| --- |
| /\*  3. (adaptat)  Definiți un pachet cu un subprogram care găsește salariul maxim al angajaților care lucrează într-un oraș,  un cursor care returnează angajații care au salariul cel puțin egal cu un salariu dat, și un cursor ajutător  care folosește subprogramul pentru a returna angajații cu salariul cel puțin egal cu salariul maxim dintr-un oraș.  \*/  CREATE OR REPLACE PACKAGE hw\_lab10 AS      CURSOR c\_angajati\_cu\_cel\_putin\_salariul(min\_sal NUMBER) RETURN angajat%ROWTYPE;      CURSOR c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(nume\_oras oras.nume%TYPE) RETURN angajat%ROWTYPE;      FUNCTION salariu\_maxim\_din\_oras(nume\_oras oras.nume%TYPE) RETURN NUMBER;      oras\_nu\_exista EXCEPTION;      PRAGMA EXCEPTION\_INIT (oras\_nu\_exista, -20003);  END hw\_lab10;  CREATE OR REPLACE PACKAGE BODY hw\_lab10 AS      CURSOR c\_angajati\_cu\_cel\_putin\_salariul(min\_sal NUMBER) RETURN angajat%ROWTYPE      IS          SELECT \*          FROM angajat          WHERE salariu >= min\_sal;      CURSOR c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(nume\_oras oras.nume%TYPE) RETURN angajat%ROWTYPE      IS          SELECT \*          FROM angajat          WHERE salariu >= salariu\_maxim\_din\_oras(nume\_oras);      FUNCTION salariu\_maxim\_din\_oras(nume\_oras oras.nume%TYPE) RETURN NUMBER      IS          v\_id\_oras oras.id\_oras%TYPE;          v\_salariu BINARY\_INTEGER;          BEGIN              BEGIN                  SELECT id\_oras                  INTO v\_id\_oras                  FROM oras                  WHERE nume = nume\_oras;              EXCEPTION                  WHEN no\_data\_found THEN                      raise\_application\_error(-20003, 'Nu s-a găsit orașul "'||nume\_oras||'".');              END;              SELECT MAX(salariu)              INTO v\_salariu              FROM angajat              JOIN restaurant              USING (id\_restaurant)              WHERE id\_oras = v\_id\_oras;              RETURN v\_salariu;      END;  END hw\_lab10;  BEGIN      FOR angajat IN hw\_lab10.c\_angajati\_cu\_cel\_putin\_salariul(5000)      LOOP          dbms\_output.put\_line(angajat.nume||' => '||angajat.salariu);      END LOOP;  END;  /  DECLARE      CURSOR c\_test\_data      IS          SELECT 1 as test\_nr, 'Cluj' AS nume FROM dual          UNION          SELECT rownum + 1 AS test\_nr, nume FROM oras;  BEGIN      FOR test IN c\_test\_data      LOOP          dbms\_output.PUT\_LINE('--- Test '||test.test\_nr||': '||test.nume||' ---');          dbms\_output.PUT('-- Salariul maxim în '||test.nume||': ');          BEGIN              dbms\_output.PUT\_LINE(hw\_lab10.salariu\_maxim\_din\_oras(test.nume));          EXCEPTION              WHEN hw\_lab10.oras\_nu\_exista THEN                  dbms\_output.PUT\_LINE(SQLERRM);          END;          dbms\_output.PUT\_LINE('-- Angajați cu salariul cel puțin egal cu maximul din acest oras:');          BEGIN              FOR angajat IN hw\_lab10.c\_angajati\_cu\_cel\_putin\_max\_salariul\_orasului(test.nume)              LOOP                  dbms\_output.put\_line(angajat.nume||' '||angajat.salariu);              END LOOP;          EXCEPTION              WHEN hw\_lab10.oras\_nu\_exista THEN                  dbms\_output.PUT\_LINE(SQLERRM);          END;      END LOOP;  END;  /  DROP PACKAGE hw\_lab10; |