**Tema 3 (saptamana 6)  
Andrei Opran 242**

**E1**

DECLARE

    TYPE tablou\_indexat IS TABLE OF EMP\_AO.SALARY%TYPE INDEX BY PLS\_INTEGER;

    id\_angajati tablou\_indexat;

    salarii\_vechi tablou\_indexat;

    salarii\_noi tablou\_indexat;

BEGIN

    -- extragere id-uri angajati si salarii vechi

    SELECT

        EMPLOYEE\_ID, SALARY

    BULK COLLECT INTO id\_angajati, salarii\_vechi

    FROM (

        SELECT

            EMPLOYEE\_ID, SALARY

        FROM EMP\_AO

        WHERE COMMISSION\_PCT IS NULL

        ORDER BY SALARY

    )

    WHERE ROWNUM <= 5;

    -- actualizare salarii

    FOR i IN 1..id\_angajati.COUNT LOOP

        UPDATE EMP\_AO

        SET SALARY = salarii\_vechi(i) \* 1.05

        WHERE EMPLOYEE\_ID = id\_angajati(i);

    END LOOP;

    -- extragere salarii noi

    FOR i IN 1..id\_angajati.COUNT LOOP

        SELECT

            SALARY

        INTO salarii\_noi(i)

        FROM EMP\_AO

        WHERE EMPLOYEE\_ID = id\_angajati(i);

    END LOOP;

    -- afisare

    FOR i IN 1..id\_angajati.COUNT LOOP

        DBMS\_OUTPUT.PUT\_LINE('ID angajat: ' || id\_angajati(i));

        DBMS\_OUTPUT.PUT\_LINE('Salariu vechi: ' || salarii\_vechi(i));

        DBMS\_OUTPUT.PUT\_LINE('Salariu nou: ' || salarii\_noi(i));

        DBMS\_OUTPUT.PUT\_LINE(' ');

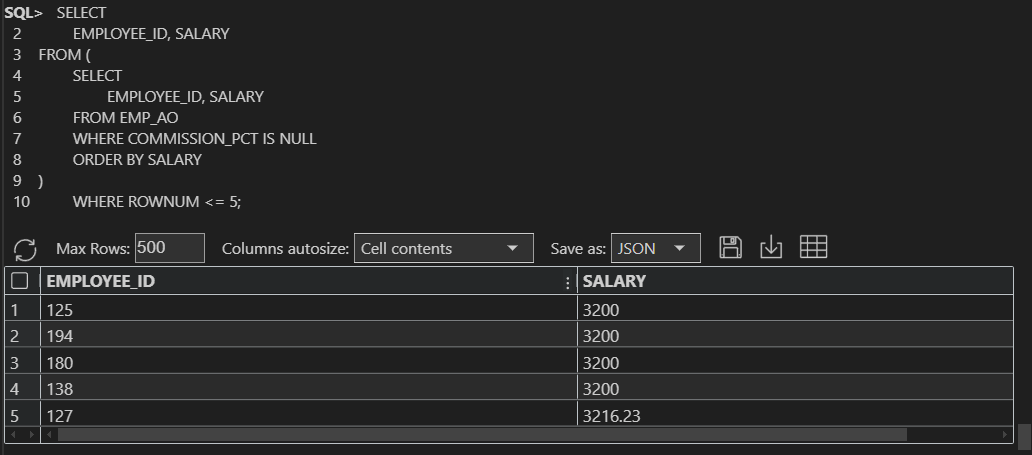
    END LOOP;

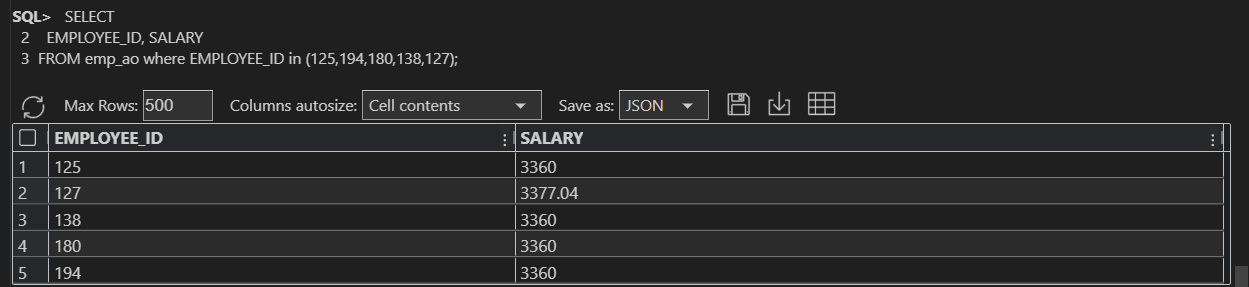
    -- ROLLBACK;

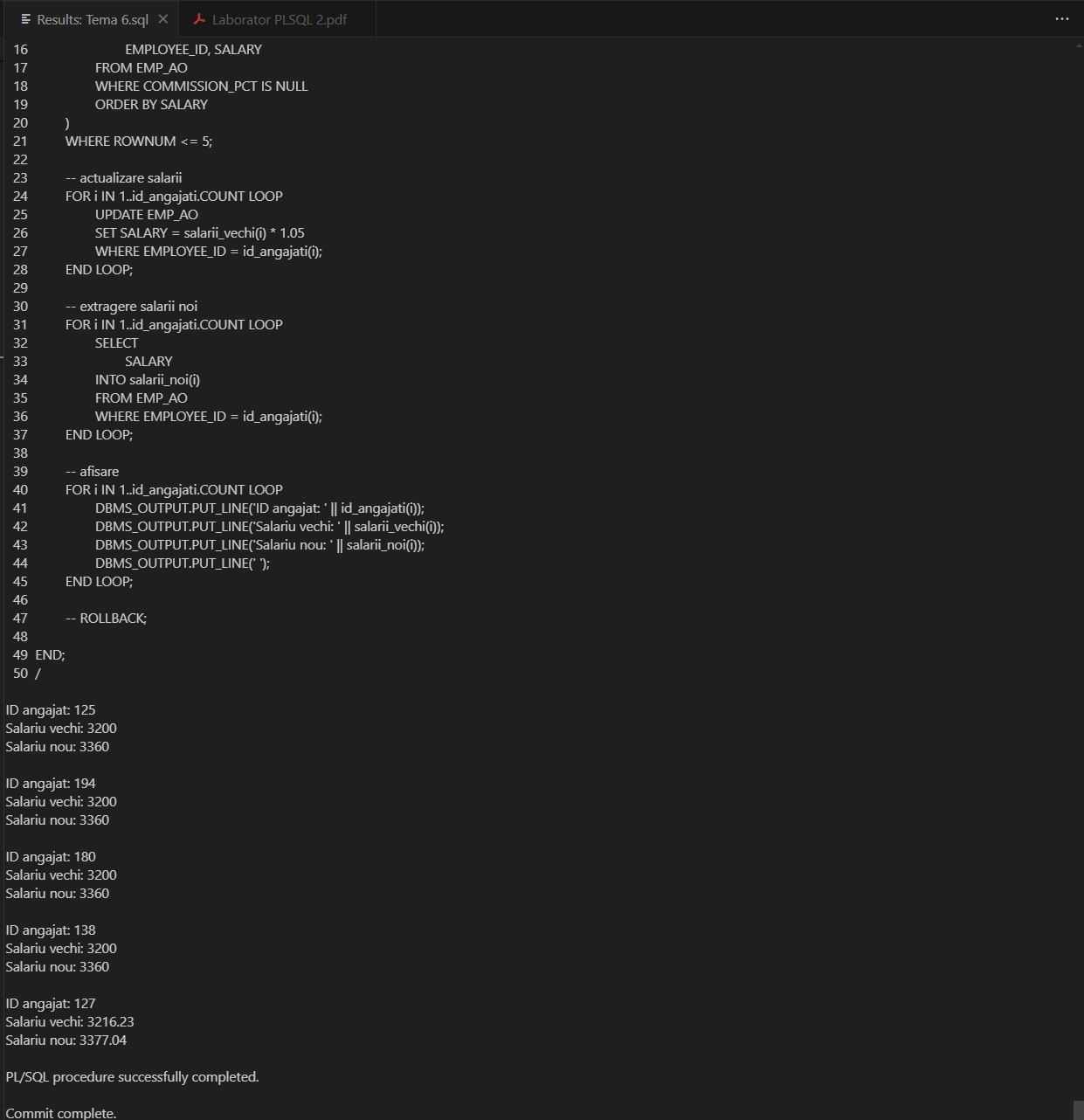
END;

/

**Top 5 angajati cu cele mai mici salarii si care nu castiga commission:**

**Inainte de rulare:  
**

**Dupa rulare:  
**

**Rezulat consola:  
**

**E2**

DECLARE

    TYPE tip\_orase\_ao IS TABLE OF VARCHAR(256);

    TYPE excursie\_ao IS RECORD(

        cod\_excursie NUMBER(4),

        denumire VARCHAR2(20),

        orase tip\_orase\_ao,

        status BOOLEAN DEFAULT TRUE -- TRUE = DISPONIBILA, FALSE = ANULATA

    );

    TYPE tabel\_excursie\_ao IS TABLE OF excursie\_ao INDEX BY BINARY\_INTEGER;

    t tabel\_excursie\_ao;

    cod\_excursie NUMBER(4) := &cod;

    oras\_nou\_1 varchar2(20) := '&primul\_oras\_nou';

    oras\_nou\_2 varchar2(20) := '&al\_doilea\_oras\_nou;

    inversare\_1 varchar2(20) := '&oras\_1';

    inversare\_2 varchar2(20) := '&oras\_2';

    ok1 BOOLEAN := FALSE;

    ok2 BOOLEAN := FALSE;

    oras\_de\_sters varchar2(20) := '&oras\_de\_sters';

    ok3 BOOLEAN := FALSE;

    EROARE\_INVERSARE EXCEPTION;

    EROARE\_STERGERE EXCEPTION;

BEGIN

    t(1).cod\_excursie := 1;

    t(1).denumire := 'Excursie 1';

    t(1).orase := tip\_orase\_ao('Bucuresti', 'Cluj-Napoca', 'Iasi');

    t(1).status := TRUE;

    t(2).cod\_excursie := 2;

    t(2).denumire := 'Excursie 2';

    t(2).orase := tip\_orase\_ao('Constanta', 'Timisoara', 'Brasov', 'Craiova');

    t(2).status := TRUE;

    t(3).cod\_excursie := 3;

    t(3).denumire := 'Excursie 3';

    t(3).orase := tip\_orase\_ao('Galati', 'Oradea');

    t(3).status := FALSE;

    t(4).cod\_excursie := 4;

    t(4).denumire := 'Excursie 4';

    t(4).orase := tip\_orase\_ao('Ploiesti', 'Campina', 'Busteni');

    t(4).status := TRUE;

    t(5).cod\_excursie := 5;

    t(5).denumire := 'Excursie 5';

    t(5).orase := tip\_orase\_ao('Bacau', 'Focsani');

    t(5).status := FALSE;

    -- -- adaugare oras in excursia cu cod-ul "cod\_excursie"

    t(cod\_excursie).orase.EXTEND;

    t(cod\_excursie).orase(t(cod\_excursie).orase.COUNT) := oras\_nou\_1;

-- adaugarea a altui oras in excursia cu cod-ul "cod\_excursie"

    t(cod\_excursie).orase.EXTEND;

    t(cod\_excursie).orase(t(cod\_excursie).orase.COUNT) := oras\_nou\_2;

-- inversarea a doua orase specificate

    FOR i IN 1..t(cod\_excursie).orase.COUNT LOOP

        IF t(cod\_excursie).orase(i) = inversare\_1 THEN

            t(cod\_excursie).orase(i) := inversare\_2;

            ok1 := TRUE;

        ELSIF t(cod\_excursie).orase(i) = inversare\_2 THEN

            t(cod\_excursie).orase(i) := inversare\_1;

            ok2 := TRUE;

        END IF;

    END LOOP;

    IF NOT ok1 OR NOT ok2 THEN

        RAISE EROARE\_INVERSARE;

    END IF;

    -- eliminati un oras specificat

    FOR i IN 1..t(cod\_excursie).orase.COUNT LOOP

        IF t(cod\_excursie).orase(i) = oras\_de\_sters THEN

            FOR j in i..t(cod\_excursie).orase.COUNT-1 LOOP

                t(cod\_excursie).orase(j) := t(cod\_excursie).orase(j+1);

            END LOOP;

            t(cod\_excursie).orase.DELETE(t(cod\_excursie).orase.COUNT);

        ok3 := TRUE;

        EXIT; -- altfel nu functioneaza pentru stergerea primului oras din lista

        END IF;

    END LOOP;

    IF NOT ok3 THEN

        RAISE EROARE\_STERGERE;

    END IF;

    -- afisare

    FOR i IN 1..t.COUNT LOOP

        DBMS\_OUTPUT.PUT\_LINE('EXCURSIA CU CODUL ' || t(i).cod\_excursie || ':');

        DBMS\_OUTPUT.PUT\_LINE('DENUMIRE: ' || t(i).denumire);

        DBMS\_OUTPUT.PUT('ORASE: ');

        FOR j IN 1..t(i).orase.COUNT LOOP

            DBMS\_OUTPUT.PUT(t(i).orase(j) || ' ');

        END LOOP;

        DBMS\_OUTPUT.NEW\_LINE;

        DBMS\_OUTPUT.PUT\_LINE('STATUS: ' || CASE WHEN t(i).status THEN 'Disponibila' ELSE 'Anulata' END);

        DBMS\_OUTPUT.PUT\_line(' ');

    END LOOP;

EXCEPTION

    WHEN EROARE\_INVERSARE THEN

        DBMS\_OUTPUT.PUT\_LINE('Nu s-a putut efectua inversarea deoarece nu exista orasele specificate in excursia cu codul ' || cod\_excursie);

    WHEN EROARE\_STERGERE THEN

        DBMS\_OUTPUT.PUT\_LINE('Nu s-a putut efectua stergerea deoarece nu exista orasul specificat in excursia cu codul ' || cod\_excursie);

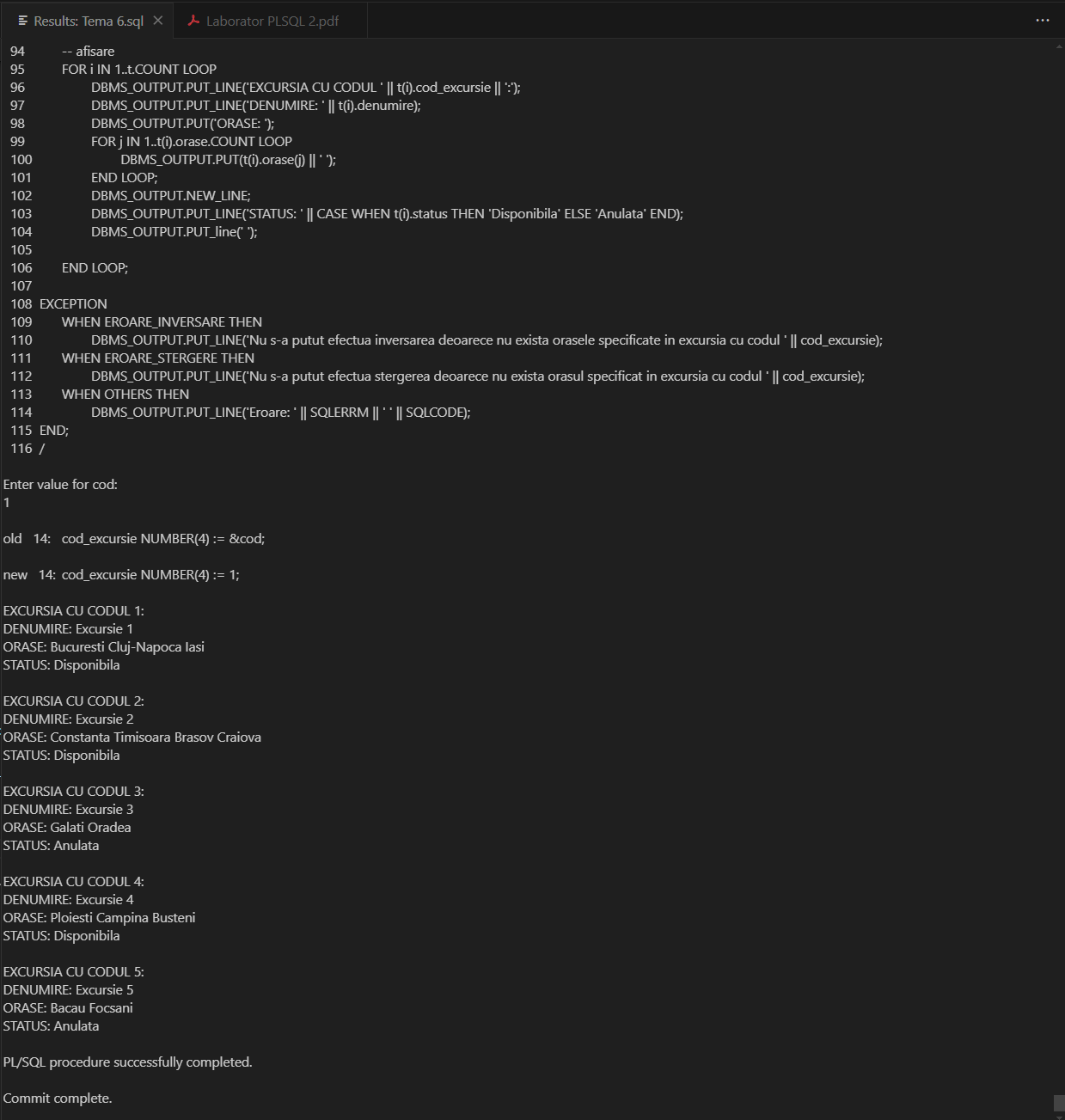
    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM || ' ' || SQLCODE);

END;

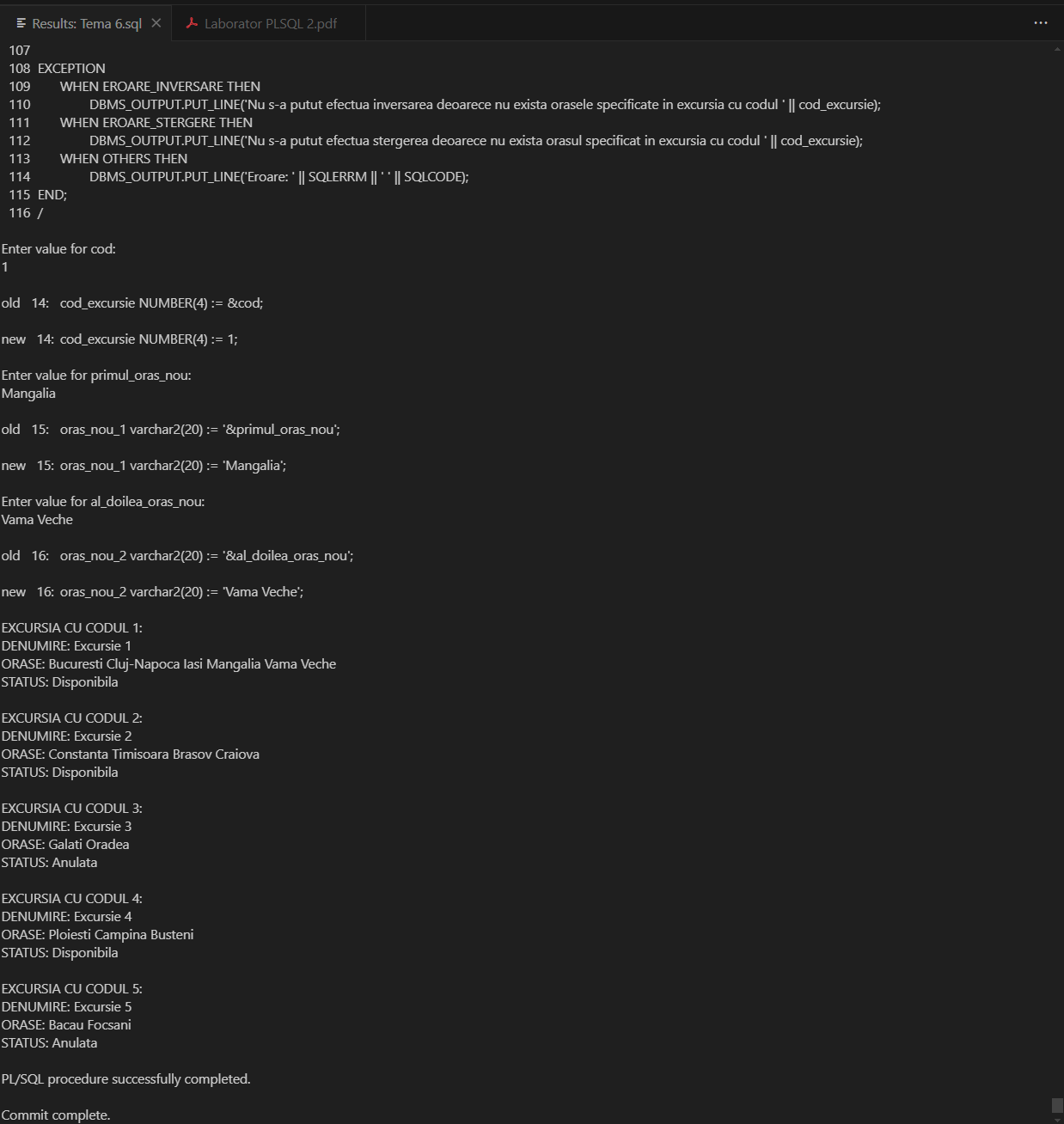
/

-- a. Inserați 5 înregistrări în tabel.

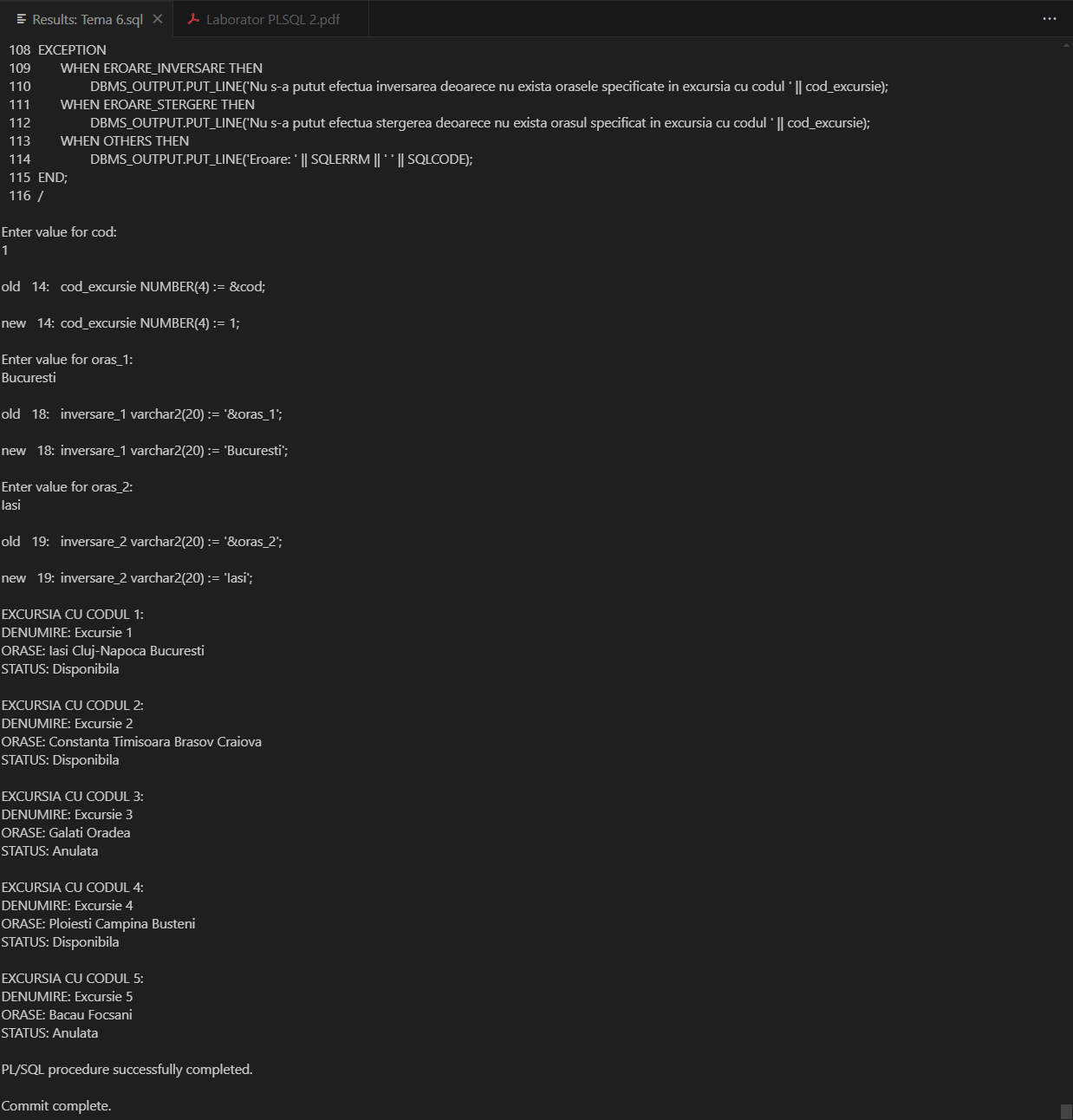
****

-- - adăugați un oraș nou în listă, ce va fi ultimul vizitat în excursia respectivă;

-- - adăugați un oraș nou în listă, ce va fi al doilea oraș vizitat în excursia respectivă;

****

-- - inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;

****

-- - eliminați din listă un oraș al cărui nume este specificat.

