Tema 7

E3

CREATE OR REPLACE FUNCTION f\_e3(

v\_oras locations.city%TYPE

) RETURN NUMBER IS

v\_numar\_angajati NUMBER;

v\_exista\_oras NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_exista\_oras

FROM locations

WHERE city = v\_oras;

IF v\_exista\_oras = 0 THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Orasul specificat nu exista.');

END IF;

SELECT COUNT(\*)

INTO v\_numar\_angajati

FROM (

SELECT e.employee\_id

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

JOIN locations l ON d.location\_id = l.location\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

WHERE l.city = v\_oras

GROUP BY e.employee\_id

HAVING COUNT(DISTINCT jh.job\_id) >= 2

);

IF v\_numar\_angajati = 0 THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista angajati in orasul dat.');

END IF;

RETURN v\_numar\_angajati;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista angajati in orasul dat.');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20005, 'A aparut o eroare');

END f\_e3;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Rezultat: '|| f\_e3('Seattle'));

END;

/

E4

CREATE OR REPLACE PROCEDURE p\_e4(v\_manager\_id employees.manager\_id%TYPE) IS

v\_exista\_manager NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_exista\_manager

FROM employees

WHERE employee\_id = v\_manager\_id;

IF v\_exista\_manager = 0 THEN

RAISE\_APPLICATION\_ERROR(-20005, 'Nu exista manager cu codul specificat.');

END IF;

UPDATE employees

SET salary = salary \* 1.1

WHERE employee\_id IN (

SELECT employee\_id

FROM employees

START WITH manager\_id = v\_manager\_id --Pentru subord directi si indirecti

CONNECT BY PRIOR employee\_id = manager\_id

);

DBMS\_OUTPUT.PUT\_LINE('Salariile au fost marite cu 10% pentru subordonatii managerului cu id: ' || v\_manager\_id);

EXCEPTION

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20006, 'A aparut o eroare');

END p\_e4;

/

BEGIN

p\_e4(101);

END;

/

