**Tema 6**

1.

create table info\_pam (

utilizator varchar2(30),

data date,

comanda varchar2(100),

nr\_linii number,

eroare varchar2(200)

);

2.

-- functia de la ex 2

CREATE OR REPLACE FUNCTION f2\_pam

(nume\_ang employees.last\_name%TYPE)

RETURN NUMBER IS

salariu employees.salary%TYPE;

utilizator varchar2(30);

data date;

nr\_linii number;

eroare varchar2(200) := 'Operatie efectuata cu succes';

BEGIN

SELECT salary INTO salariu

FROM employees

WHERE last\_name = nume\_ang;

SELECT user, sysdate INTO utilizator, data

FROM dual;

SELECT count(\*)INTO nr\_linii

FROM employees

WHERE last\_name = nume\_ang;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

RETURN salariu;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

0, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista angajati cu numele dat');

RETURN -1;

WHEN TOO\_MANY\_ROWS THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi angajati cu numele dat');

RETURN -1;

WHEN OTHERS THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare');

RETURN -1;

END f2\_pam;

/

-- procedura de la ex 4

CREATE OR REPLACE PROCEDURE p4\_pam(nume\_ang employees.last\_name%TYPE)

IS

salariu employees.salary%TYPE;

utilizator varchar2(30);

data date;

nr\_linii number;

eroare varchar2(200) := 'Operatie efectuata cu succes';

BEGIN

SELECT salary INTO salariu

FROM employees

WHERE last\_name = nume\_ang;

SELECT user, sysdate INTO utilizator, data

FROM dual;

SELECT count(\*)INTO nr\_linii

FROM employees

WHERE last\_name = nume\_ang;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

dbms\_output.put\_line('Salariul este ' || salariu);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

0, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista angajati cu numele dat');

WHEN TOO\_MANY\_ROWS THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi angajati cu numele dat');

WHEN OTHERS THEN

eroare := SQLERRM;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT salary INTO salariu FROM employees WHERE last\_name = nume\_ang;',

nr\_linii, eroare);

COMMIT;

RAISE\_APPLICATION\_ERROR(-20002, 'Alta eroare');

END p4\_pam;

/

3.

CREATE OR REPLACE FUNCTION fct\_pam(oras locations.city%TYPE DEFAULT 'Munich')

RETURN NUMBER IS

utilizator varchar2(30);

data date;

eroare varchar2(200) := 'Operatie efectuata cu succes';

v\_nr\_orase number;

v\_nr\_ang number;

v\_nr\_ang\_cu\_2\_joburi number;

BEGIN

SELECT count(\*) INTO v\_nr\_orase

FROM locations l

WHERE lower(l.city) = lower(oras);

SELECT user, sysdate INTO utilizator, data

FROM dual;

IF v\_nr\_orase = 0 THEN

eroare := 'Nu exista niciun oras cu acest nume!';

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT count(\*) INTO v\_nr\_orase FROM locations l WHERE lower(l.city)=lower(oras);',

0, eroare);

ELSE

SELECT count(e.employee\_id) INTO v\_nr\_ang

FROM employees e, departments d, locations l

WHERE e.department\_id=d.department\_id and d.location\_id=l.location\_id and lower(l.city)=lower(oras);

IF v\_nr\_ang = 0 THEN

eroare := 'Nu exista niciun angajat in orasul cu acest nume!';

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT count(e.employee\_id) INTO v\_nr\_ang...WHERE lower(l.city)=lower(oras);',

0, eroare);

ELSE

SELECT count(count(\*)) INTO v\_nr\_ang\_cu\_2\_joburi

FROM employees e, departments d, locations l, job\_history j

WHERE e.department\_id=d.department\_id and d.location\_id=l.location\_id

and e.employee\_id=j.employee\_id and lower(l.city)=lower(oras)

GROUP BY j.employee\_id

HAVING count(\*) >= 2;

INSERT INTO info\_pam

VALUES (utilizator, data,

'SELECT count(count(\*)) INTO v\_nr\_ang\_cu\_2\_joburi...WHERE lower(l.city)=lower(oras);',

v\_nr\_ang\_cu\_2\_joburi, eroare);

END IF;

END IF;

RETURN v\_nr\_ang\_cu\_2\_joburi;

END fct\_pam;

/