**Tema 4**

1)

-- Panait Ana

DECLARE

TYPE v IS varray(5) OF emp\_pam.employee\_id%TYPE;

coduri v;

sal\_vechi INTEGER;

sal\_nou INTEGER;

BEGIN

select employee\_id bulk collect into coduri

from (select employee\_id

from emp\_pam

where commission\_pct is null

order by salary asc)

where rownum <= 5;

for i in 1..5 loop

select salary into sal\_vechi

from emp\_pam

where employee\_id = coduri(i);

update emp\_pam

set salary = salary + salary \* 0.05

where employee\_id = coduri(i);

select salary into sal\_nou

from emp\_pam

where employee\_id = coduri(i);

dbms\_output.put\_line('Salariatul cu id-ul ' || coduri(i) ||' avea salariul ' || sal\_vechi || ' si acum are salariul ' || sal\_nou);

end loop;

END;

/

2)

create or replace type tip\_orase\_pam as varray(20) of varchar2(20);

/

create table excursie\_pam (

cod\_excursie NUMBER(4) PRIMARY KEY,

denumire VARCHAR2(20),

orase tip\_orase\_pam,

status VARCHAR2(11)

);

a)

insert into excursie\_pam values(1, 'Excursie1', tip\_orase\_pam('Oras1', 'Oras2', 'Oras3'), 'disponibila');

insert into excursie\_pam values(2, 'Excursie2', tip\_orase\_pam('Oras4', 'Oras5'), 'anulata');

insert into excursie\_pam values(3, 'Excursie3', tip\_orase\_pam('Oras6', 'Oras7', 'Oras8', 'Oras9'), 'disponibila');

insert into excursie\_pam values(4, 'Excursie4', tip\_orase\_pam('Oras10'), 'anulata');

insert into excursie\_pam values(5, 'Excursie5', tip\_orase\_pam('Oras11', 'Oras12', 'Oras13'), 'disponibila');

b)

* adaugati un oras nou în lista, ce va fi ultimul vizitat în excursia respectiva

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras\_nou';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

v\_orase.extend;

v\_orase(v\_orase.last) := nume\_oras;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* adaugati un oras nou în lista, ce va fi al doilea oras vizitat în excursia respective

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras\_nou';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

v\_orase.extend;

for i in reverse 3..(v\_orase.last) loop

v\_orase(i) := v\_orase(i-1);

end loop;

v\_orase(2) := nume\_oras;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* inversati ordinea de vizitare a doua dintre orase al caror nume este specificat

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras\_1 varchar2(20) := '&nume\_oras\_1';

nume\_oras\_2 varchar2(20) := '&nume\_oras\_2';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

for i in v\_orase.first..v\_orase.last loop

if v\_orase(i) = nume\_oras\_1 then

v\_orase(i) := nume\_oras\_2;

elsif v\_orase(i) = nume\_oras\_2 then

v\_orase(i) := nume\_oras\_1;

end if;

end loop;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* eliminati din lista un oras al carui nume este specificat

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras';

indx\_oras number;

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

for i in v\_orase.first..v\_orase.last loop

if v\_orase(i) = nume\_oras then

indx\_oras := i;

end if;

end loop;

for i in indx\_oras..(v\_orase.last-1) loop

v\_orase(i) := v\_orase(i+1);

end loop;

v\_orase.trim();

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

c)

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

dbms\_output.put\_line('Numarul de orase vizitate este ' || v\_orase.count);

dbms\_output.new\_line;

for i in v\_orase.first..v\_orase.last loop

dbms\_output.put\_line(v\_orase(i));

end loop;

END;

/

d)

DECLARE

type v is varray(20) of number;

v\_orase excursie\_pam.orase%type;

v\_coduri v := v();

BEGIN

select cod\_excursie

bulk collect into v\_coduri

from excursie\_pam;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

dbms\_output.put('Excursia cu codul ' || v\_coduri(i) || ' are orasele: ');

for j in v\_orase.first..v\_orase.last loop

dbms\_output.put(v\_orase(j) || ' ');

end loop;

dbms\_output.new\_line;

end loop;

END;

/

e)

DECLARE

type v is varray(20) of number;

v\_orase excursie\_pam.orase%type;

v\_coduri v := v();

ids\_nou v := v();

minim number := -1;

BEGIN

select cod\_excursie

bulk collect into v\_coduri

from excursie\_pam;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

if minim = -1 or minim > v\_orase.count then

minim := v\_orase.count;

end if;

end loop;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

if v\_orase.count = minim then

update excursie\_pam

set status = 'anulata'

where cod\_excursie = v\_coduri(i);

end if;

end loop;

END;

/

3)

create or replace type tip\_orase\_pam is table of varchar2(20);

/

create table excursie\_pam (

cod\_excursie NUMBER(4) PRIMARY KEY,

denumire VARCHAR2(20),

orase tip\_orase\_pam,

status VARCHAR2(11)

) nested table orase store as tbl\_orase\_pam;

/

a)

insert into excursie\_pam values(1, 'Excursie1', tip\_orase\_pam('Oras1', 'Oras2', 'Oras3'), 'disponibila');

insert into excursie\_pam values(2, 'Excursie2', tip\_orase\_pam('Oras4', 'Oras5'), 'anulata');

insert into excursie\_pam values(3, 'Excursie3', tip\_orase\_pam('Oras6', 'Oras7', 'Oras8', 'Oras9'), 'disponibila');

insert into excursie\_pam values(4, 'Excursie4', tip\_orase\_pam('Oras10'), 'anulata');

insert into excursie\_pam values(5, 'Excursie5', tip\_orase\_pam('Oras11', 'Oras12', 'Oras13'), 'disponibila');

b)

* adaugati un oras nou în lista, ce va fi ultimul vizitat în excursia respective

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras\_nou';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

v\_orase.extend;

v\_orase(v\_orase.last) := nume\_oras;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* adaugati un oras nou în lista, ce va fi al doilea oras vizitat în excursia respective

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras\_nou';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

v\_orase.extend;

for i in reverse 3..(v\_orase.last) loop

v\_orase(i) := v\_orase(i-1);

end loop;

v\_orase(2) := nume\_oras;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* inversati ordinea de vizitare a doua dintre orase al caror nume este specificat

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras\_1 varchar2(20) := '&nume\_oras\_1';

nume\_oras\_2 varchar2(20) := '&nume\_oras\_2';

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

for i in v\_orase.first..v\_orase.last loop

if v\_orase(i) = nume\_oras\_1 then

v\_orase(i) := nume\_oras\_2;

elsif v\_orase(i) = nume\_oras\_2 then

v\_orase(i) := nume\_oras\_1;

end if;

end loop;

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

* eliminati din lista un oras al carui nume este specificat

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

nume\_oras varchar2(20) := '&nume\_oras';

indx\_oras number;

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

for i in v\_orase.first..v\_orase.last loop

if v\_orase(i) = nume\_oras then

indx\_oras := i;

end if;

end loop;

v\_orase.delete(indx\_oras);

update excursie\_pam

set orase = v\_orase

where cod\_excursie = v\_cod;

END;

/

c)

DECLARE

v\_orase excursie\_pam.orase%TYPE;

v\_cod excursie\_pam.cod\_excursie%TYPE := &cod;

BEGIN

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_cod;

dbms\_output.put\_line('Numarul de orase vizitate este ' || v\_orase.count);

for i in v\_orase.first..v\_orase.last loop

dbms\_output.put\_line(v\_orase(i));

end loop;

END;

/

d)

DECLARE

type v is varray(20) of number;

v\_orase excursie\_pam.orase%TYPE;

v\_coduri v := v();

BEGIN

select cod\_excursie

bulk collect into v\_coduri

from excursie\_pam;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

dbms\_output.put('Excursia cu codul ' || v\_coduri(i) || ' are orasele: ');

for j in v\_orase.first..v\_orase.last loop

dbms\_output.put(v\_orase(j) || ' ');

end loop;

dbms\_output.new\_line;

end loop;

END;

/

e)

DECLARE

type v is varray(20) of number;

v\_orase excursie\_pam.orase%TYPE;

v\_coduri v := v();

ids\_nou v := v();

minim number := -1;

BEGIN

select cod\_excursie

bulk collect into v\_coduri

from excursie\_pam;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

if minim = -1 or minim > v\_orase.count then

minim := v\_orase.count;

end if;

end loop;

for i in v\_coduri.first..v\_coduri.last loop

select orase

into v\_orase

from excursie\_pam

where cod\_excursie = v\_coduri(i);

if v\_orase.count = minim then

update excursie\_pam

set status = 'anulata'

where cod\_excursie = v\_coduri(i);

end if;

end loop;

END;

/