Temă SGBD 4 15.11.2023

1. Cerință: Din laboratorul PLSQL 2, rezolvați exercițiul E2 (cu tablou imbricat) cu multiset.

Definiți un tip colecție denumit tip\_orase\_\*\*\*. Creați tabelul excursie\_\*\*\* cu următoarea structură: cod\_excursie NUMBER(4), denumire VARCHAR2(20), orase tip\_orase\_\*\*\* (ce va conține lista orașelor care se vizitează într-o excursie, într-o ordine stabilită; de exemplu, primul oraș din listă va fi primul oraș vizitat), status (disponibilă sau anulată).

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

/

create table excursie (

cod\_excursie number(4),

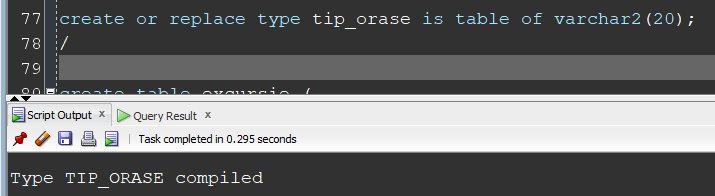
denumire varchar2(20),

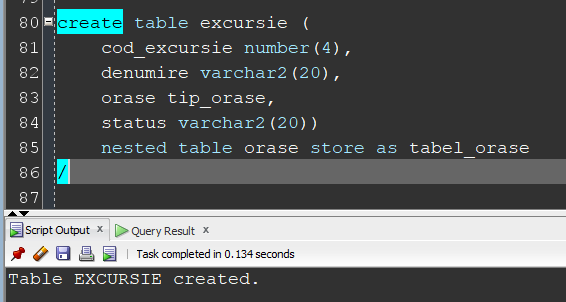
orase tip\_orase,

status varchar2(20))

nested table orase store as tabel\_orase

/





a. Inserați 5 înregistrări în tabel.

begin

insert into excursie

values (1, 'Norvegia', tip\_orase('Oslo', 'Bergen', 'Trondheim', 'Tromso'), 'disponibila');

insert into excursie

values (2, 'Munte', tip\_orase('Sinaia', 'Brasov', 'Predeal'), 'disponibila');

insert into excursie

values (3, 'Grecia', tip\_orase('Salonic', 'Atena'), 'anulata');

insert into excursie

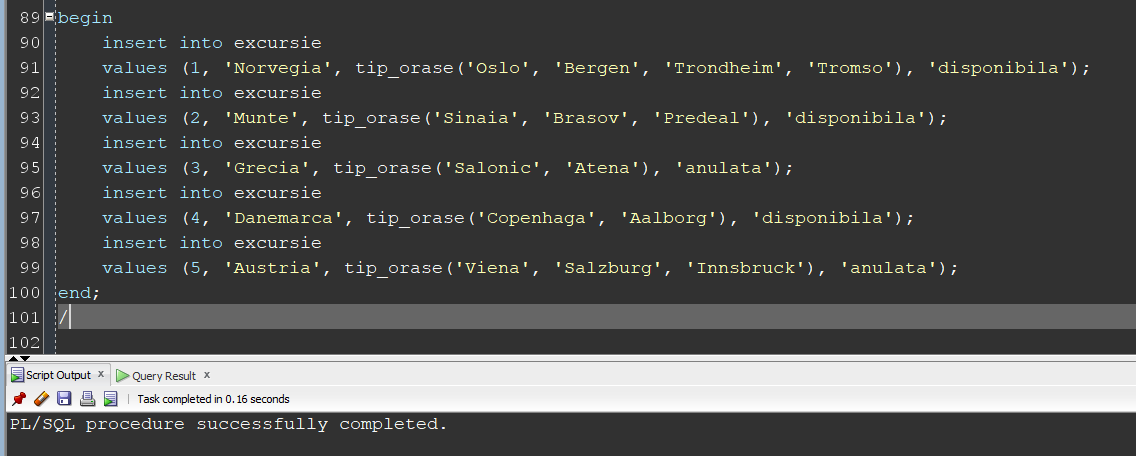
values (4, 'Danemarca', tip\_orase('Copenhaga', 'Aalborg'), 'disponibila');

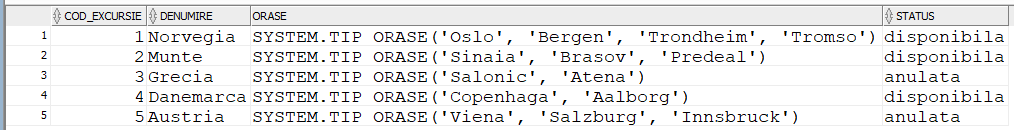
insert into excursie

values (5, 'Austria', tip\_orase('Viena', 'Salzburg', 'Innsbruck'), 'anulata');

end;

/





b. Actualizați coloana orase pentru o excursie specificată:

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

declare

v\_cod excursie.cod\_excursie%type := 5;

v\_oras varchar2(20) := 'Linz';

begin

update excursie

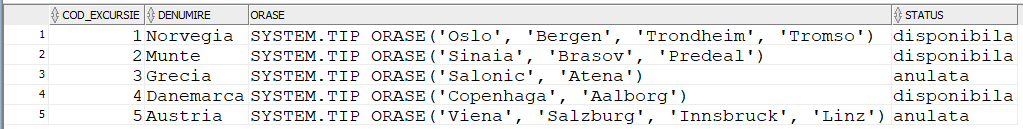
set orase = orase multiset union all tip\_orase(v\_oras)

where cod\_excursie = v\_cod;

end;

/

După actualizare:



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

declare

v\_cod excursie.cod\_excursie%type := 5;

v\_oras varchar2(20) := 'Graz';

v\_orase tip\_orase;

begin

select orase

into v\_orase

from excursie

where cod\_excursie = v\_cod;

update excursie

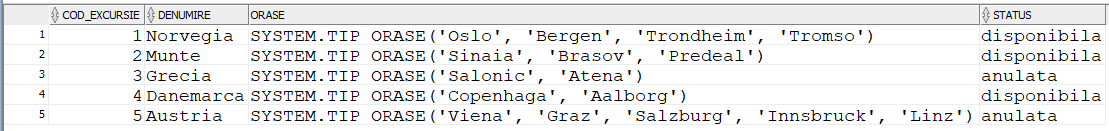
set orase = (tip\_orase(v\_orase(1)) multiset union all tip\_orase(v\_oras)) multiset union all (v\_orase multiset except tip\_orase(v\_orase(1)))

where cod\_excursie = v\_cod;

end;

/

După actualizare:



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

declare

v\_cod excursie.cod\_excursie%type := 5;

v\_orase tip\_orase;

oras1 varchar2(20) := 'Salzburg';

oras2 varchar2(20) := 'Innsbruck';

begin

select orase

into v\_orase

from excursie

where cod\_excursie = v\_cod;

for i in 1..cardinality(v\_orase) loop

if v\_orase(i) = oras1 then

v\_orase(i) := oras2;

elsif v\_orase(i) = oras2 then

v\_orase(i) := oras1;

end if;

end loop;

update excursie

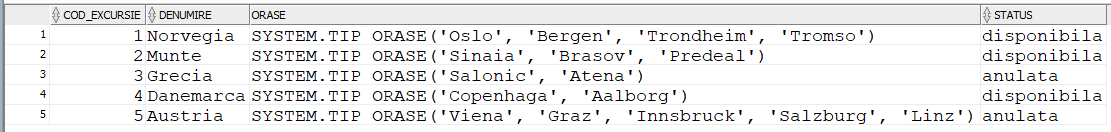
set orase = v\_orase

where cod\_excursie = v\_cod;

end;

/

După actualizare:



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

declare

v\_cod excursie.cod\_excursie%type := 5;

v\_oras varchar2(20) := 'Graz';

begin

update excursie

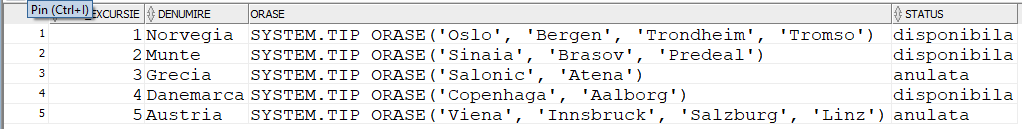
set orase = orase multiset except tip\_orase(v\_oras)

where cod\_excursie = v\_cod;

end;

/

După actualizare:



c. Pentru o excursie al cărui cod este dat, afișați numărul de orașe vizitate, respectiv numele orașelor.

declare

v\_cod excursie.cod\_excursie%type := 5;

v\_orase tip\_orase;

begin

select orase

into v\_orase

from excursie

where cod\_excursie = v\_cod;

dbms\_output.put\_line('Excursia viziteaza ' || cardinality(v\_orase) || ' orase: ');

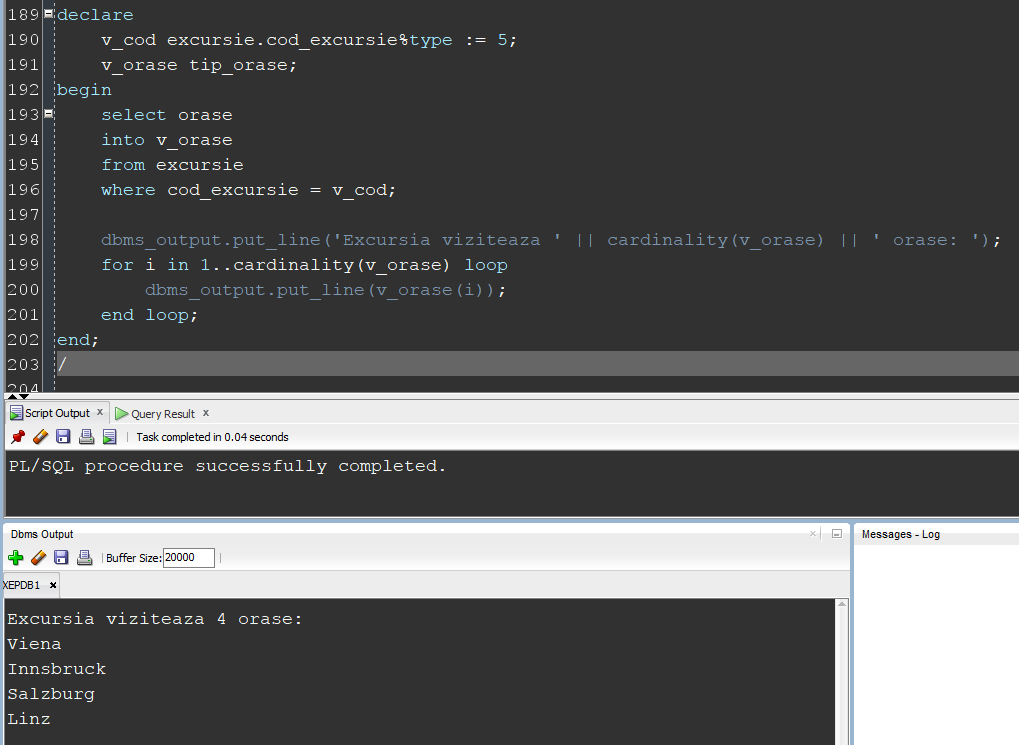
for i in 1..cardinality(v\_orase) loop

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

end loop;

end;

/



d. Pentru fiecare excursie afișați lista orașelor vizitate.

declare

type tip\_coduri is table of excursie.cod\_excursie%type;

v\_coduri tip\_coduri;

v\_orase tip\_orase;

v\_nume excursie.denumire%type;

begin

select cod\_excursie

bulk collect into v\_coduri

from excursie;

for i in 1..cardinality(v\_coduri) loop

select denumire

into v\_nume

from excursie

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

dbms\_output.put\_line('Excursia ' || v\_nume || ': ');

select orase

into v\_orase

from excursie

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

for i in 1..cardinality(v\_orase) loop

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

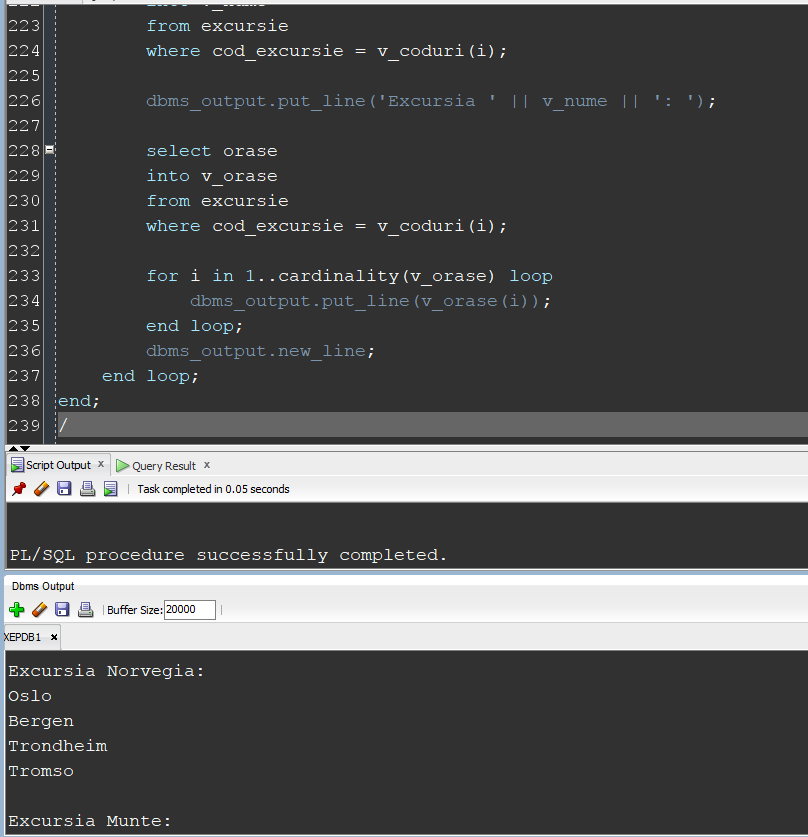
end loop;

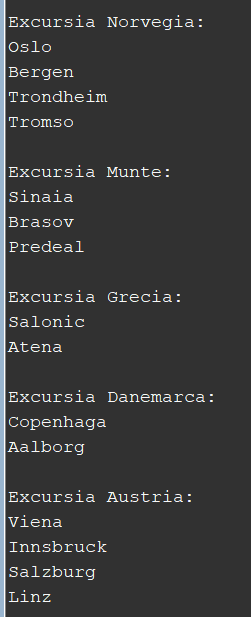
dbms\_output.new\_line;

end loop;

end;

/





e. Anulați excursiile cu cele mai puține orașe vizitate.

declare

type tip\_coduri is table of excursie.cod\_excursie%type;

v\_coduri tip\_coduri;

v\_orase tip\_orase;

v\_nume excursie.denumire%type;

v\_nr\_min number(4) := 1000;

begin

select cod\_excursie

bulk collect into v\_coduri

from excursie;

for i in 1..cardinality(v\_coduri) loop

select orase

into v\_orase

from excursie

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

if cardinality(v\_orase) < v\_nr\_min then

v\_nr\_min := cardinality(v\_orase);

end if;

end loop;

for i in 1..cardinality(v\_coduri) loop

select orase

into v\_orase

from excursie

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

if cardinality(v\_orase) = v\_nr\_min then

update excursie

set status = 'anulata'

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

end if;

end loop;

end;

/

După actualizare:

