Enunt:

2. 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).

a. Inserați 5 înregistrări în tabel.

b. Actualizați coloana orase pentru o excursie specificată:

- 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.

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

d. Pentru fiecare excursie afișați lista orașelor vizitate.

e. Anulați excursiile cu cele mai puține orașe vizitate.

3. Rezolvați problema anterioară folosind un alt tip de colecție studiat.

DROP TABLE EXCURSIE\_GTA;

CREATE OR REPLACE TYPE vector\_orase\_gta IS VARRAY(30) OF VARCHAR2(100);

/

CREATE TABLE excursie\_GTA(

cod\_excursie NUMBER(4) CONSTRAINT PK\_COD\_EXC999 PRIMARY KEY,

denumire VARCHAR2(20),

orase vector\_orase\_gta,

status VARCHAR2(15) CONSTRAINT CHK\_STATUS999 CHECK (STATUS IN ('DISPONIBIL', 'ANULAT')));

INSERT into excursie\_GTA

VALUES (100, 'Montenegro',vector\_orase\_GTA('Podgorica'), 'DISPONIBIL');

INSERT into excursie\_GTA

VALUES (200, 'Albania',vector\_orase\_GTA('Tirana','Shkodra'), 'ANULAT');

INSERT into excursie\_GTA

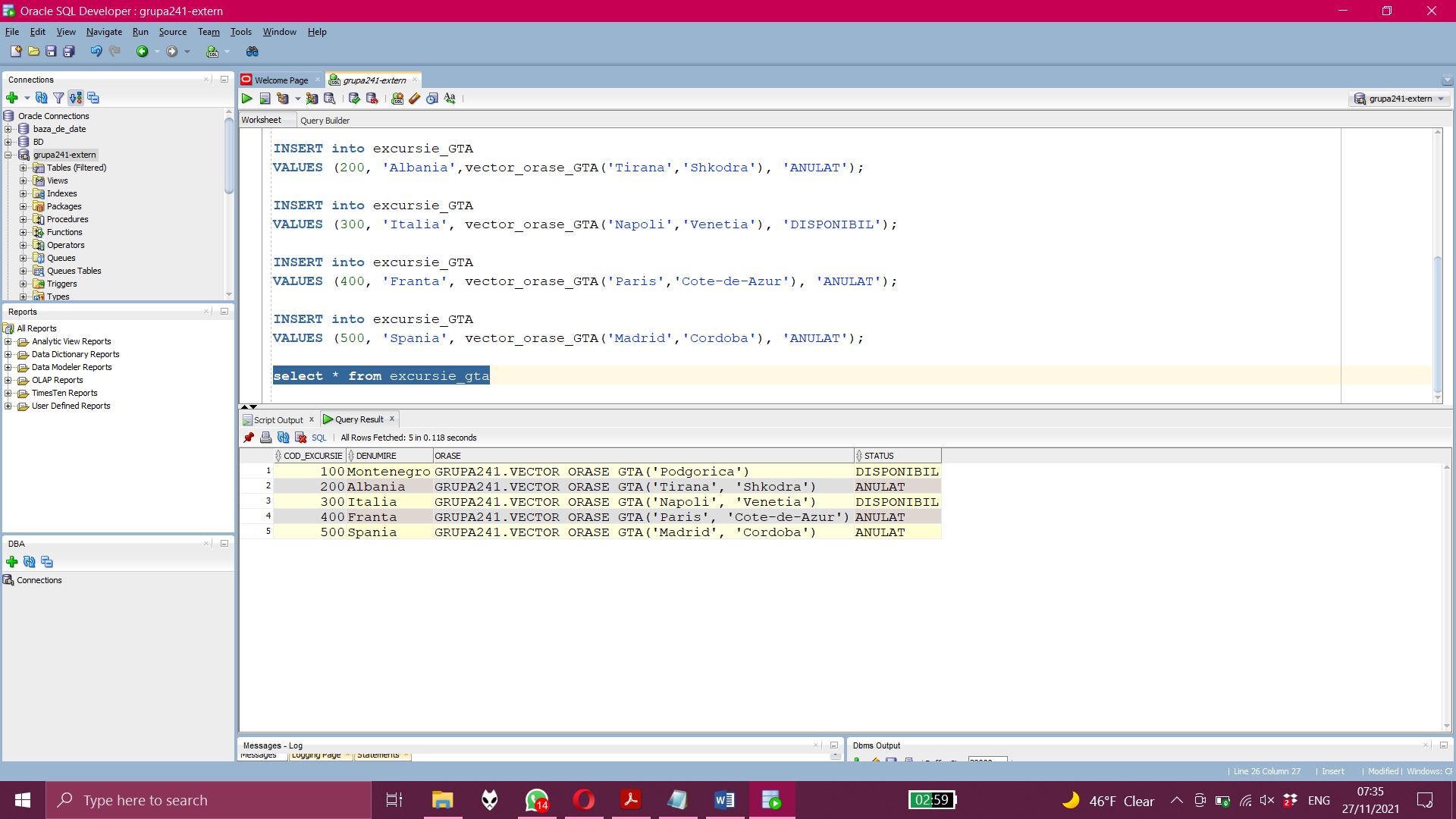
VALUES (300, 'Italia', vector\_orase\_GTA('Napoli','Venetia'), 'DISPONIBIL');

INSERT into excursie\_GTA

VALUES (400, 'Franta', vector\_orase\_GTA('Paris','Cote-de-Azur'), 'ANULAT');

INSERT into excursie\_GTA

VALUES (500, 'Spania', vector\_orase\_GTA('Madrid','Cordoba'), 'ANULAT');



--b

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

/

DECLARE

v\_id EXCURSIE\_GTA.cod\_excursie%type := &p\_id;

v\_lista\_orase tip\_orase\_GTA;

v\_nou\_oras VARCHAR2(100) :='&p\_oras';

begin

select orase

into v\_lista\_orase

from excursie\_GTA

where cod\_excursie= v\_id;

--facem loc pe pozitia 2 si copiem toate elementele cu o pozitie la dreapta

v\_lista\_orase.extend();

for i in reverse 3..v\_lista\_orase.last LOOP

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

end loop;

v\_lista\_orase(2):=v\_nou\_oras;

update excursie\_GTA

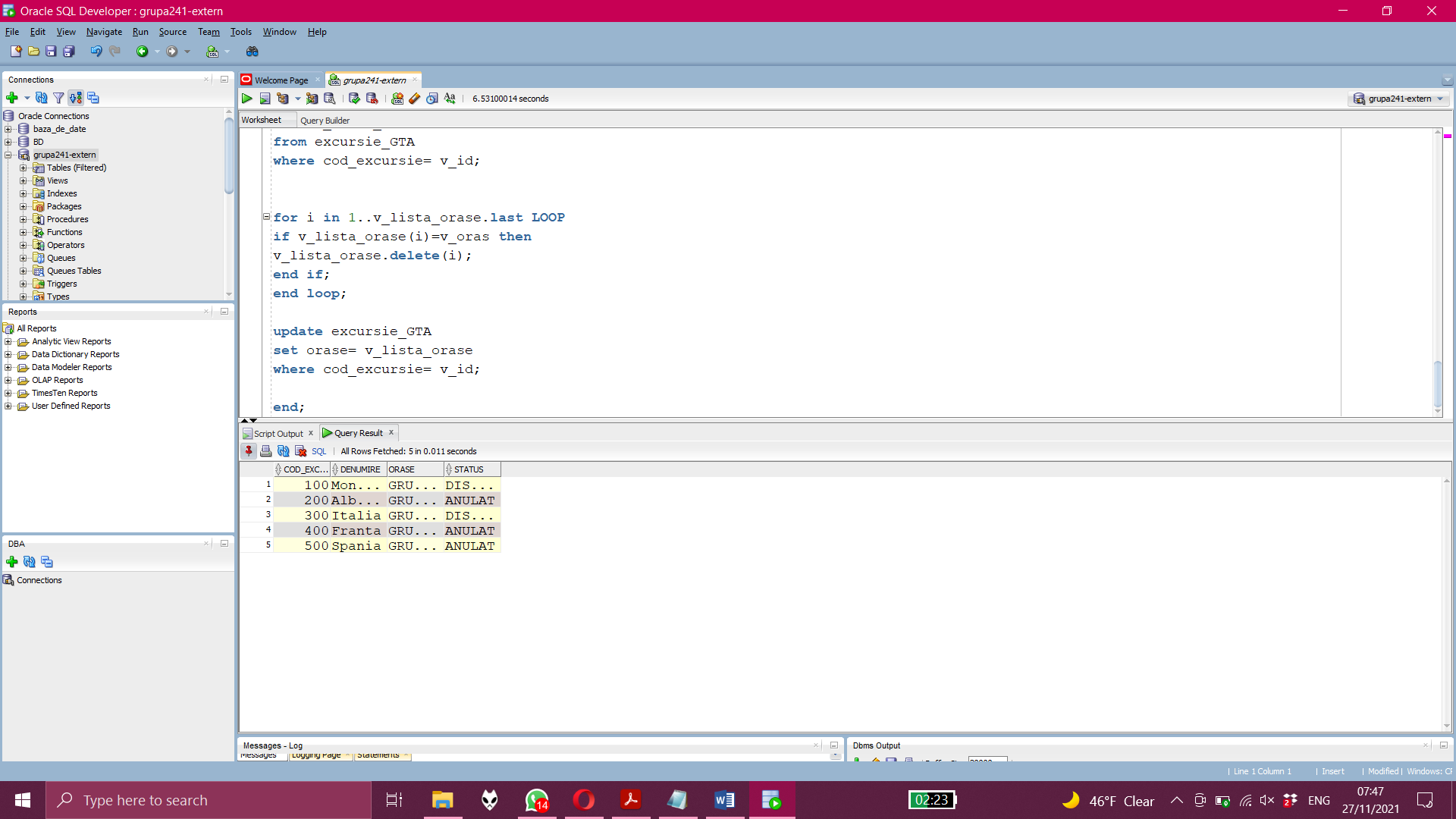
set orase= v\_lista\_orase

where cod\_excursie= v\_id;

end;

/

select \* from excursie\_GTA;



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

/

DECLARE

v\_id EXCURSIE\_GTA.cod\_excursie%type := &p\_id;

v\_lista\_orase vector\_orase\_GTA;

v\_nou\_oras VARCHAR2(100) :='&p\_oras';

begin

select orase

into v\_lista\_orase

from excursie\_GTA

where cod\_excursie= v\_id;

--facem loc pe pozitia 2 si copiem toate elementele cu o pozitie la dreapta

v\_lista\_orase.extend();

for i in reverse 3..v\_lista\_orase.last LOOP

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

end loop;

v\_lista\_orase(2):=v\_nou\_oras;

update excursie\_GTA

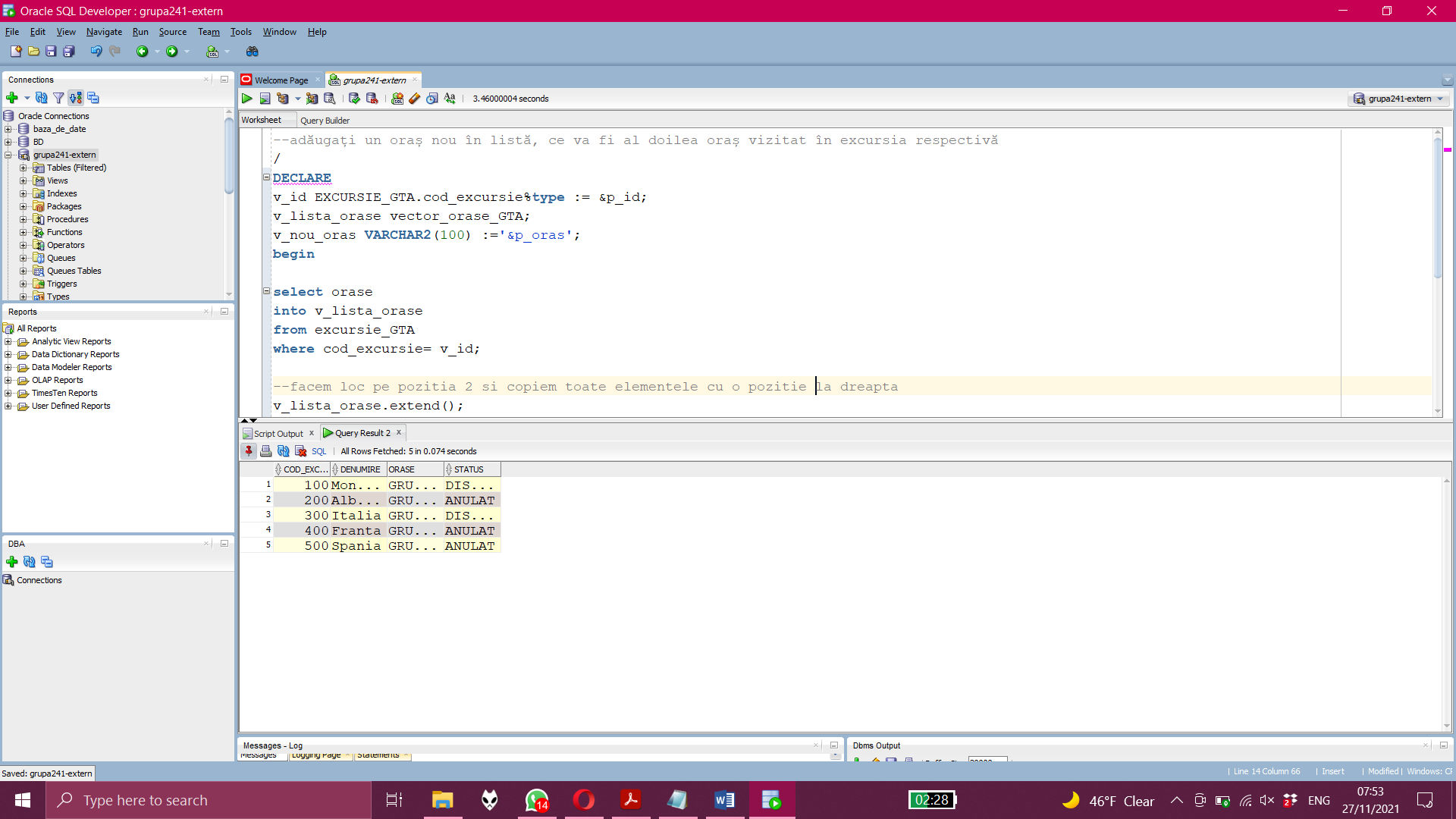
set orase= v\_lista\_orase

where cod\_excursie= v\_id;

end;

/

select \* from excursie\_GTA;



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

/

DECLARE

v\_id EXCURSIE\_GTA.cod\_excursie%type := &p\_id;

v\_primul\_oras VARCHAR2(100) :='&p\_oras1';

v\_aldoilea\_oras VARCHAR2(100) :='&p\_oras2';

v\_lista\_orase vector\_orase\_GTA;

begin

select orase

into v\_lista\_orase

from excursie\_GTA

where cod\_excursie= v\_id;

--punem in locul primului oras pe al doilea si invers

for i in 1..v\_lista\_orase.last LOOP

if v\_lista\_orase(i)=v\_primul\_oras then

v\_lista\_orase(i):=v\_aldoilea\_oras;

else

if v\_lista\_orase(i)=v\_aldoilea\_oras then

v\_lista\_orase(i):=v\_primul\_oras;

end if;

end if;

end loop;

update excursie\_GTA

set orase= v\_lista\_orase

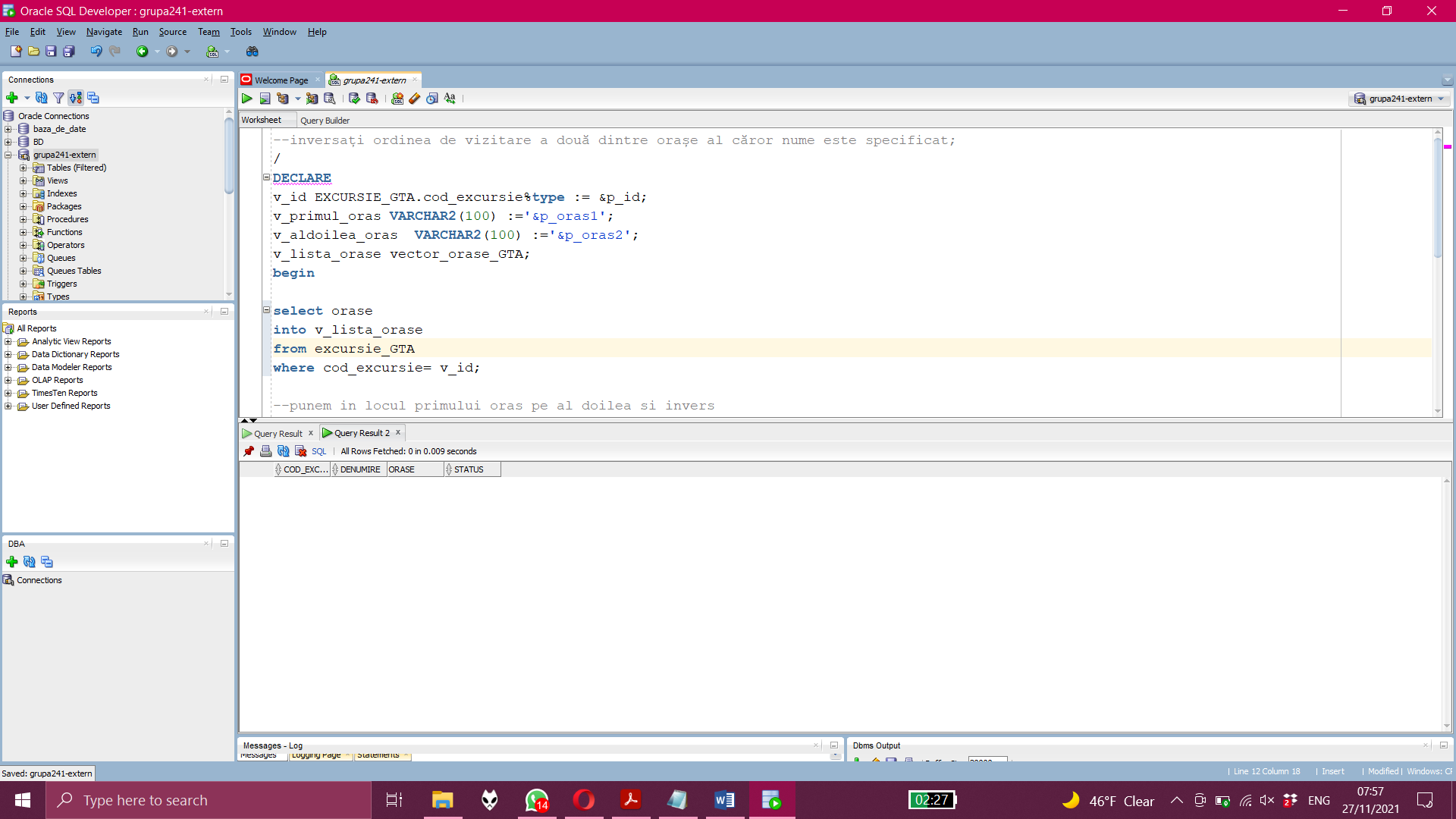
where cod\_excursie= v\_id;

end;

/

select \* from excursie\_GTA;

rollback;



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

/

DECLARE

v\_id EXCURSIE\_GTA.cod\_excursie%type := &p\_id;

v\_oras VARCHAR2(100) :='&p\_oras';

v\_lista\_orase vector\_orase\_GTA;

begin

select orase

into v\_lista\_orase

from excursie\_GTA

where cod\_excursie= v\_id;

for i in 1..v\_lista\_orase.last

LOOP

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

v\_lista\_orase.delete(i);

end if;

end loop;

update excursie\_GTA

set orase= v\_lista\_orase

where cod\_excursie= v\_id;

end;

/

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

/

DECLARE

v\_id EXCURSIE\_GTA.cod\_excursie%type := &p\_id;

v\_numar number(5):= 0;

v\_lista\_orase vector\_orase\_gta;

begin

select orase

into v\_lista\_orase

from excursie\_gta

where cod\_excursie= v\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista date ');

--numaram cate orase se gasesc si le afisez

for i in 1..v\_lista\_orase.last LOOP

v\_numar := v\_numar +1;

DBMS\_OUTPUT.PUT(v\_lista\_orase(i)|| ' ');

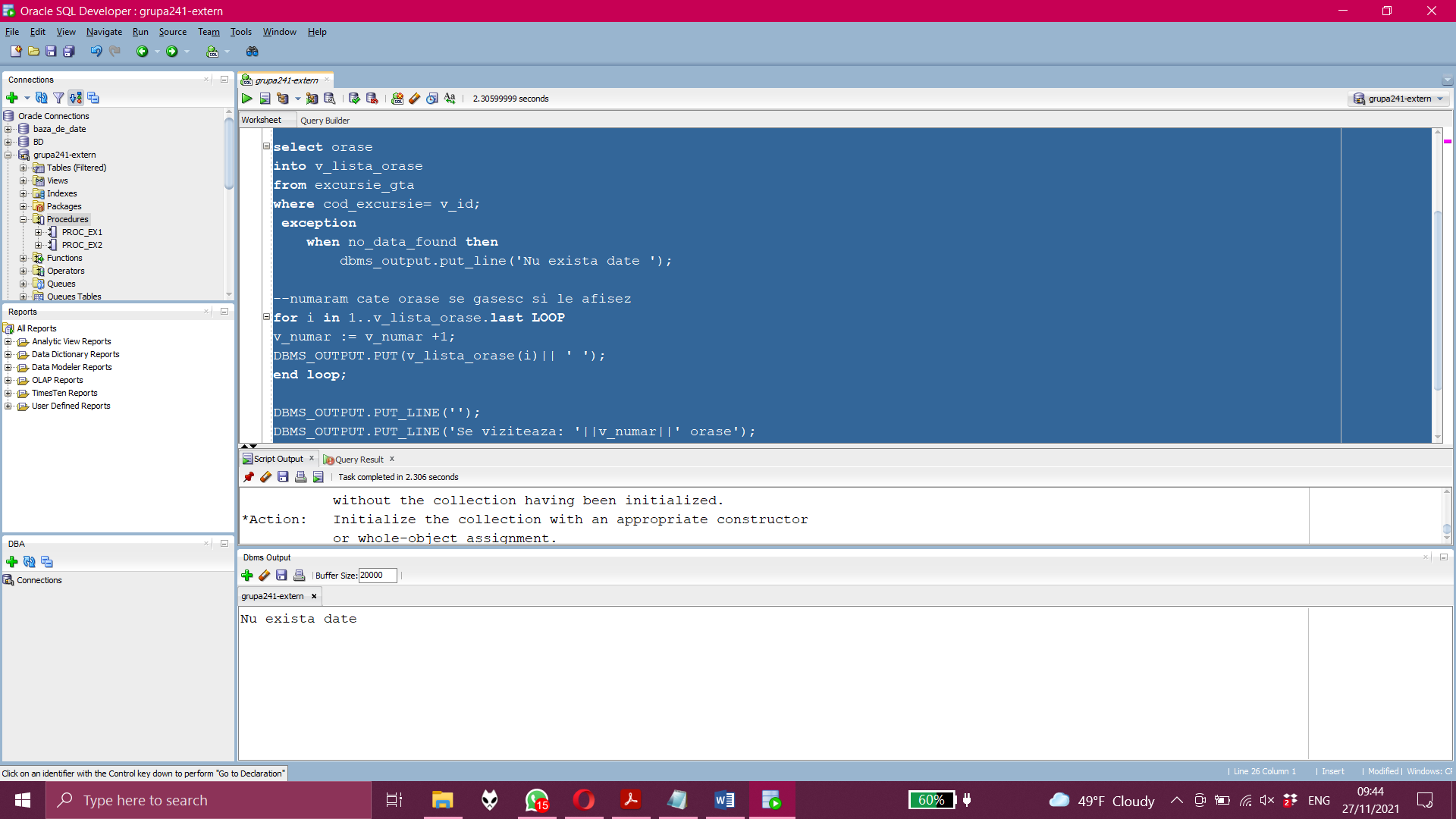
end loop;

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('Se viziteaza: '||v\_numar||' orase');

end;

/



--d.Pentru fiecare excursie afișați lista orașelor vizitate.

/

DECLARE

TYPE tablou\_indexat IS TABLE OF excursie\_GTA%ROWTYPE INDEX BY BINARY\_INTEGER;

t tablou\_indexat;

begin

--pun toata tabela excursie\_GTA intr-un tablou indexat

SELECT \*

BULK COLLECT INTO t

FROM excursie\_GTA;

--parcurg tabloul afisand denumirea si parcurgand fiecare lista de orase in parte ca sa afisez orasele

FOR i IN t.FIRST..t.LAST LOOP

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

for j in t(i).orase.first..t(i).orase.last loop

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

end loop;

DBMS\_OUTPUT.PUT\_line('');

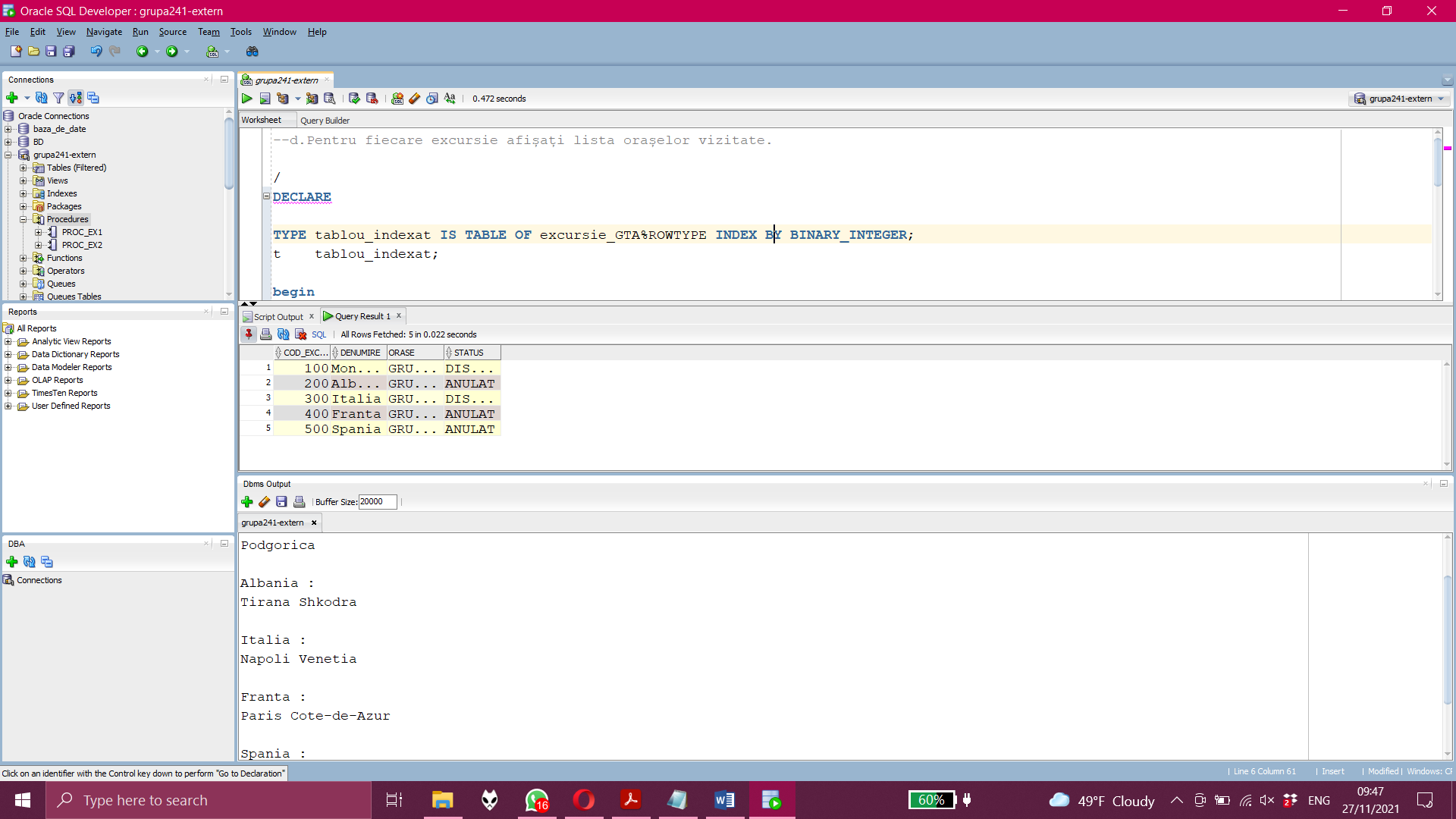
DBMS\_OUTPUT.PUT\_line('');

END LOOP;

end;

/

select \* from excursie\_GTA;



--e.Anulați excursiile cu cele mai puține orașe vizitate. (DIFICULTATE MEDIE)

/

declare

nr\_min number(3) :=0;

TYPE tablou\_indexat IS TABLE OF excursie\_GTA%ROWTYPE INDEX BY BINARY\_INTEGER;

t tablou\_indexat;

begin

SELECT \*

BULK COLLECT INTO t

FROM excursie\_GTA;

nr\_min:=t(1).orase.count;

--calculez minimul de elemente din listele de orase

for i in 2..t.last loop

if t(i).orase.count<nr\_min then

nr\_min:=t(i).orase.count;

end if;

end loop;

for i in 1..t.last loop

if t(i).orase.count=nr\_min then

update excursie\_GTA

set status='ANULAT'

where cod\_excursie=t(i).cod\_excursie;

end if;

end loop;

end;

/

select \* from excursie\_GTA;

rollback;

