

TEMA LABORATOR 6
(Laborator PLSQL 2)

1. Mențineți într-o colecție codurile celor mai prost plătiți 5 angajați care nu câștigă comision. Folosind această colecție măriți cu 5% salariul acestor angajați. Afișați valoarea veche a salariului, respectiv valoarea nouă a salariului.

Am folosit tabelul auxiliar emp_eis pentru a actualiza salariul.

```
drop table emp_eis;
create table emp_eis as select * from employees;
select * from emp_eis;

declare
    type vector is varray(5) of employees.employee_id%type;
    v vector;
    salariu_vechi employees.salary%type;

begin
    with aux as (select employee_id
                   from employees
                  where commission_pct is null
                  order by salary)
    select employee_id
    bulk collect into v
    from aux
    where rownum < 6;

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

        select salary into salariu_vechi
        from employees
        where employee_id = v(i);

        update emp_eis
        set salary = (salary + salary * 0.05)
        where employee_id = v(i);

        dbms_output.put_line('Angajatul cu codul ' || v(i) || ' castiga un salariu de ' || salariu_vechi ||
                              '. Salariul marit cu 5% este ' || (salariu_vechi + 0.05 * salariu_vechi) || '.');
    end loop;

end;
/
```

Oracle Developer 2.0.1 - Worksheet

```

declare
    type vector is varray(5) of employees.employee_id%type;
    v vector;
    salariu_vechi employees.salary%type;
begin
    with aux as (select employee_id
                  from employees
                  where commission_pct is null
    )

```

Script Output: Query Result

Fetches 50 rows in 0.049 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
130	Mozhe	Atkinson	MATKINSO	650.124.6234	30-OCT-97	ST CLERK	2800	(null)	121	50
131	James	Marlow	JAMRLOW	650.124.7234	16-FEB-97	ST CLERK	2500	(null)	121	50
132	John	Watson	HWATSON	650.124.3224	10-SEP-92	ST CLERK	2200	(null)	121	50
133	Jason	Mallin	JMALLIN	650.127.1934	14-JUN-96	ST CLERK	3300	(null)	122	50
134	Michael	Rodgers	MROGERS	650.127.1834	26-AUG-98	ST CLERK	2900	(null)	122	50
135	Ki	Gee	KGEE	650.127.1734	12-DEC-99	ST CLERK	2520	(null)	122	50

Group Output: Buffer Size: 20000

Grupa 233

```

Angajatul cu codul 132 castiga un salariu de 2100. Salariul marit cu 5% este 2205.
Angajatul cu codul 128 castiga un salariu de 2200. Salariul marit cu 5% este 2310.
Angajatul cu codul 136 castiga un salariu de 2200. Salariul marit cu 5% este 2310.
Angajatul cu codul 127 castiga un salariu de 2400. Salariul marit cu 5% este 2520.
Angajatul cu codul 135 castiga un salariu de 2400. Salariul marit cu 5% este 2520.

```

ctrl key down to perform "Go to Declaration"

Line 2 Column 23 | Insert | Modified | Windows: O

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. Intrați 5 înregistrări în tabel.

```
create or replace type tip_orase_eis as varray(10) of varchar2(60);
```

```
create table excursie_eis (cod_excursie number(4),
                          denumire varchar2(20),
                          orase tip_orase_eis,
                          status varchar2(20));
```

```
insert into excursie_eis
values (1, 'Prima mea excursie', tip_orase_eis('Brasov', 'Bucuresti'), 'disponibila');
```

```
insert into excursie_eis
values (2, 'Hai la mare', tip_orase_eis('Constanta', 'Mangalia', 'Fetesti'), 'disponibila');
```

```
insert into excursie_eis
values (3, 'Hai la munte', tip_orase_eis('Brasov', 'Sibiu', 'Busteni', 'Sinaia'), 'disponibila');
```

```
insert into excursie_eis
values (4, 'La aer curat', tip_orase_eis('Brasov', 'Sinaia'), 'anulata');
```

```
insert into excursie_eis
```

```
values (5, 'Strainatate', tip_orase_eis('Budapesta', 'Paris', 'Madrid'), 'disponibila');
```

```
select * from excursie_eis;
```

The screenshot shows the Oracle SQL Developer interface. The script editor contains the following SQL code:

```
orase tip_orase_eis,
status varchar2(20));

insert into excursie_eis
values (1, 'Prima mea excursie', tip_orase_eis('Brasov', 'Bucuresti'), 'disponibila');

insert into excursie_eis
values (2, 'Hai la mare', tip_orase_eis('Constanta', 'Mangalia', 'Fetesti'), 'disponibila');

insert into excursie_eis
values (3, 'Hai la munte', tip_orase_eis('Brasov', 'Sibiu', 'Busteni', 'Sinaia'), 'disponibila');

insert into excursie_eis
values (4, 'La aer curat', tip_orase_eis('Brasov', 'Sinaia'), 'anulata');
```

The query result table shows the following data:

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	disponibila
2	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti')	disponibila
3	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Sibiu', 'Busteni', 'Sinaia')	disponibila
4	La aer curat	GRUPA233.TIP ORASE EIS('Brasov', 'Sinaia')	anulata
5	Strainatate	GRUPA233.TIP ORASE EIS('Budapesta', 'Paris', 'Madrid')	disponibila

b. Actualizați coloana orase pentru o excursie specificată:

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

The screenshot shows the Oracle SQL Developer interface. The script editor contains the following SQL code:

```
declare
    v_cod_excursie_a excursie_eis.cod_excursie%type := 2;
    v_orase excursie_eis.orase%type;

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie_a;

    v_orase.extend();
    v_orase(v_orase.last) := 'Tulcea';

    update excursie_eis
```

The query result table shows the following data:

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	disponibila
2	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti', 'Tulcea')	disponibila
3	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Sibiu', 'Busteni', 'Sinaia')	disponibila
4	La aer curat	GRUPA233.TIP ORASE EIS('Brasov', 'Sinaia')	anulata
5	Strainatate	GRUPA233.TIP ORASE EIS('Budapesta', 'Paris', 'Madrid')	disponibila

```
declare
    v_cod_excursie_a excursie_eis.cod_excursie%type := 2;
    v_orase excursie_eis.orase%type;
```

```
begin
```

```

select orase into v_orase
from excursie_eis
where cod_excursie = v_cod_excursie_a;

```

```

v_orase.extend();
v_orase(v_orase.last) := 'Tulcea';

```

```

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_a;

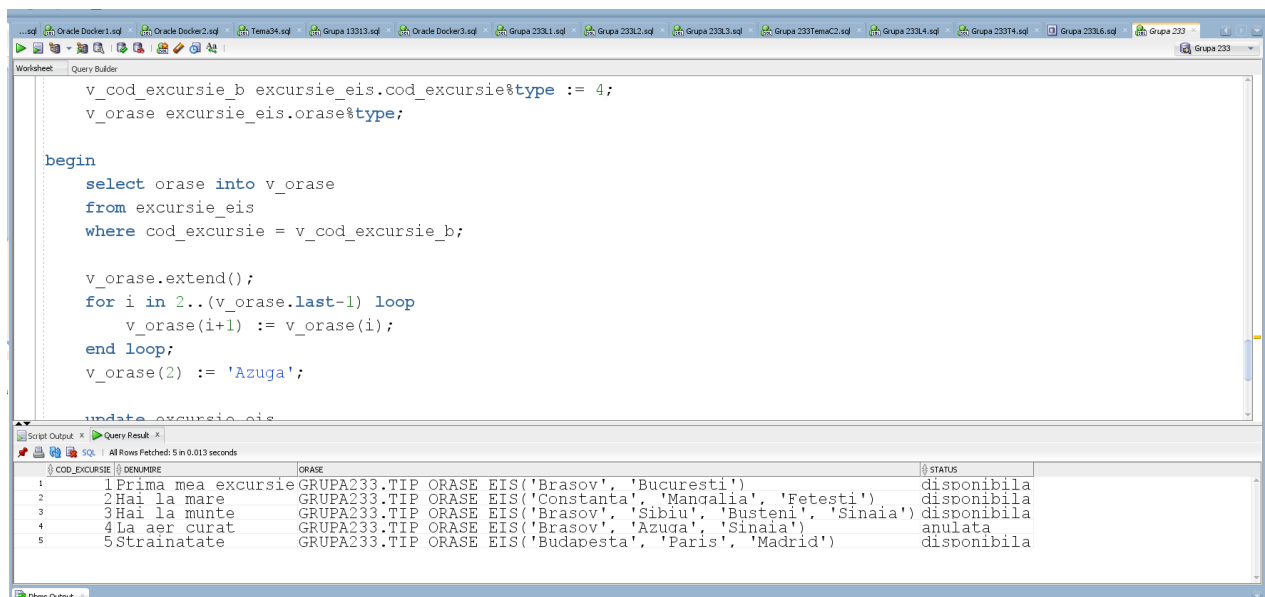
```

```

end;
/

```

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



```

declare
v_cod_excursie_b excursie_eis.cod_excursie%type := 4;
v_orase excursie_eis.orase%type;

```

```

begin
  select orase into v_orase
  from excursie_eis
  where cod_excursie = v_cod_excursie_b;

```

```

v_orase.extend();
for i in 2..(v_orase.last-1) loop
  v_orase(i+1) := v_orase(i);
end loop;
v_orase(2) := 'Azuga';

```

```

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_b;

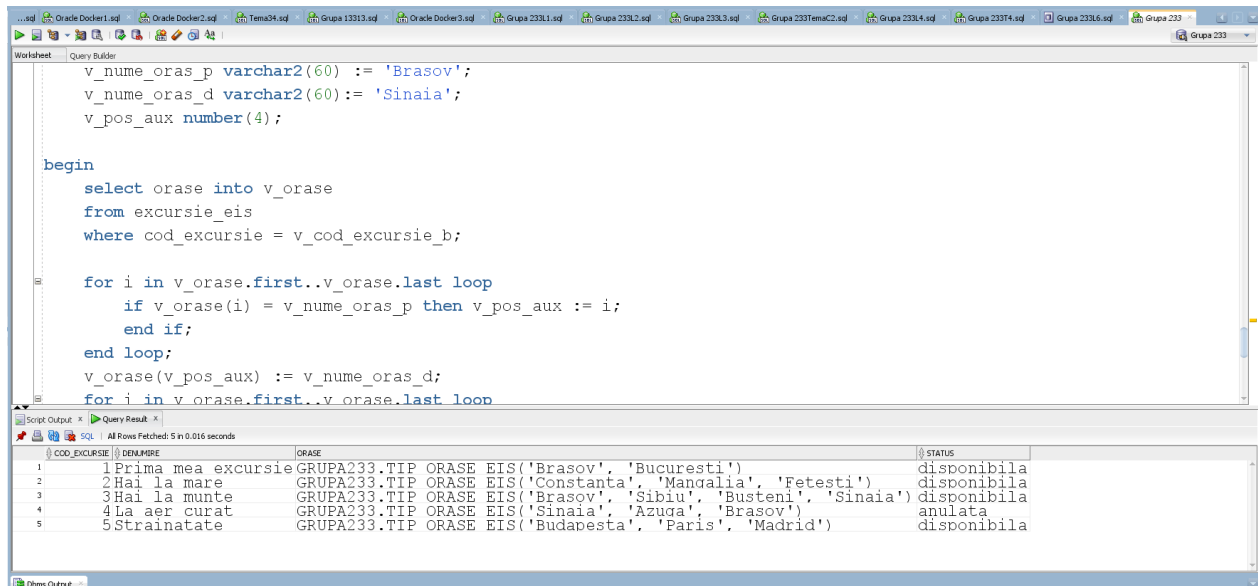
```

```

end;
/

```

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



```

declare
v_cod_excursie_b excursie_eis.cod_excursie%type := 4;
v_orase excursie_eis.orase%type;
v_num_oras_p varchar2(60) := 'Brasov';
v_num_oras_d varchar2(60) := 'Sinaia';
v_pos_aux number(4);

```

```

begin
select orase into v_orase
from excursie_eis
where cod_excursie = v_cod_excursie_b;

for i in v_orase.first..v_orase.last loop
    if v_orase(i) = v_num_oras_p then v_pos_aux := i;
    end if;
end loop;
v_orase(v_pos_aux) := v_num_oras_d;
for i in v_orase.first..v_orase.last loop
    if v_orase(i) = v_num_oras_d then v_pos_aux := i;
    end if;

```

```

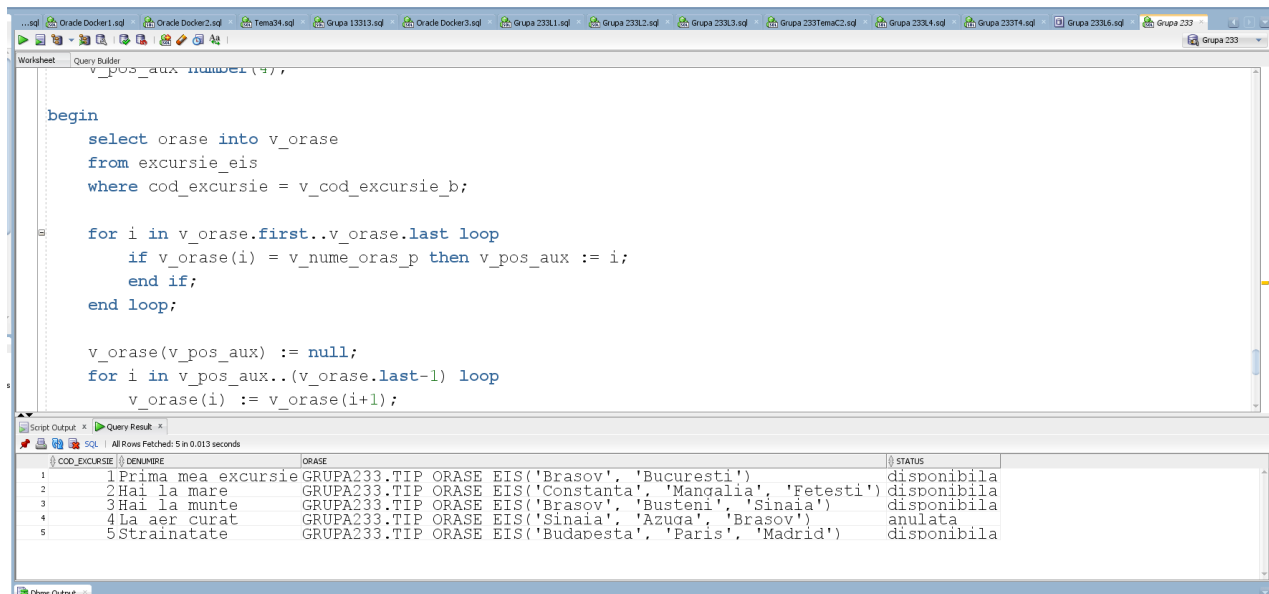
end loop;
v_orase(v_pos_aux) := v_nume_oras_p;

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_b;

end;
/

```

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



```

declare
v_cod_excursie_b excursie_eis.cod_excursie%type := 3;
v_orase excursie_eis.orase%type;
v_nume_oras_p varchar2(60) := 'Sibiu';
v_pos_aux number(4);

```

```

begin
select orase into v_orase
from excursie_eis
where cod_excursie = v_cod_excursie_b;

for i in v_orase.first..v_orase.last loop
if v_orase(i) = v_nume_oras_p then v_pos_aux := i;
end if;
end loop;

v_orase(v_pos_aux) := null;
for i in v_pos_aux..(v_orase.last-1) loop

```

```

        v_orase(i) := v_orase(i+1);
    end loop;

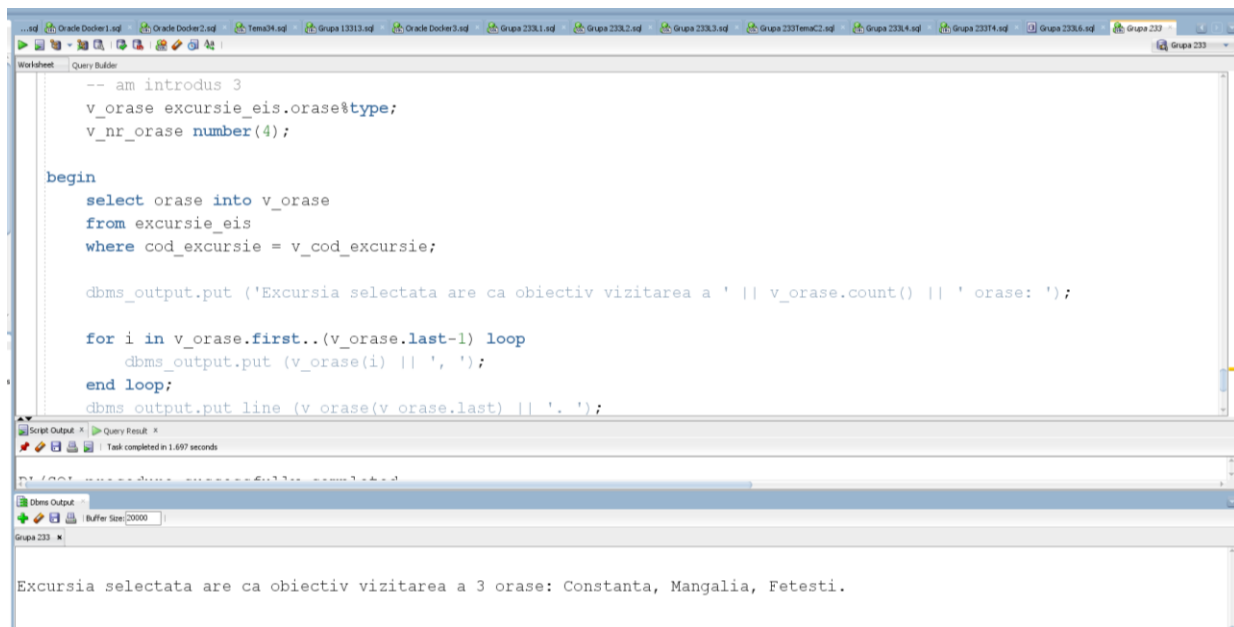
    v_orase.trim();

    update excursie_eis
    set orase = v_orase
    where cod_excursie = v_cod_excursie_b;

end;
/

```

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



The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Script Editor' tab. The script is as follows:

```

-- am introdus 3
v_orase excursie_eis.orase%type;
v_nr_orase number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie;

    dbms_output.put ('Excursia selectata are ca obiectiv vizitarea a ' || v_orase.count() || ' orase: ');

    for i in v_orase.first..(v_orase.last-1) loop
        dbms_output.put (v_orase(i) || ', ');
    end loop;
    dbms_output.put_line (v_orase(v_orase.last) || '. ');
end;

```

Below the script editor, the 'Script Output' window shows the execution results. The output is:

```

Excursia selectata are ca obiectiv vizitarea a 3 orase: Constanta, Mangalia, Fetesti.

```

The 'Task completed in 1.697 seconds' message is also visible in the 'Script Output' window.

```

declare
    v_cod_excursie excursie_eis.cod_excursie%type := &p_cod_excursie;
    -- am introdus 3
    v_orase excursie_eis.orase%type;
    v_nr_orase number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie;

    dbms_output.put ('Excursia selectata are ca obiectiv vizitarea a ' || v_orase.count() || ' orase: ');

    for i in v_orase.first..(v_orase.last-1) loop
        dbms_output.put (v_orase(i) || ', ');
    end loop;
    dbms_output.put_line (v_orase(v_orase.last) || '. ');
end;

```

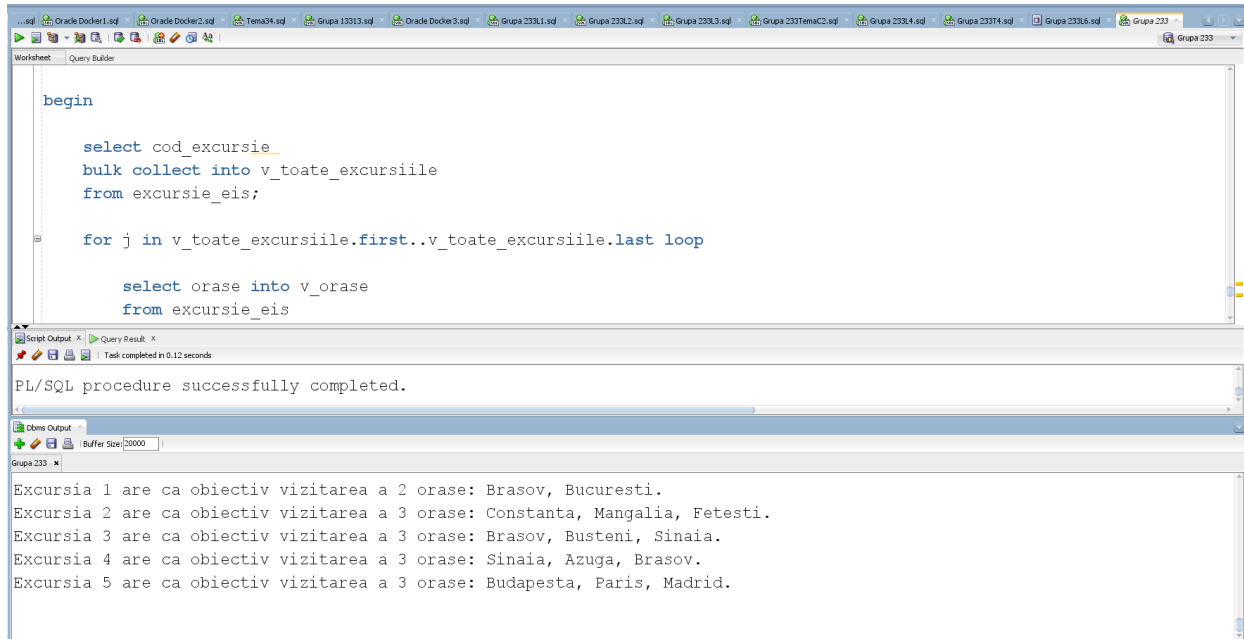
```

end loop;
dbms_output.put_line (v_orase(v_orase.last) || ' ');

end;
/

```

d. Pentru fiecare excursie afișați lista orașelor vizitate.



```

declare
    v_orase excursie_eis.orase%type;
    v_nr_orase number(4);
    type vector is varray(5) of excursie_eis.cod_excursie%type;
    v_toate_excursiile vector;

```

```

begin

```

```

    select cod_excursie
    bulk collect into v_toate_excursiile
    from excursie_eis;

```

```

    for j in v_toate_excursiile.first..v_toate_excursiile.last loop

```

```

        select orase into v_orase
        from excursie_eis
        where cod_excursie = v_toate_excursiile(j);

```

```

        dbms_output.put ('Excursia ' || v_toate_excursiile(j) || ' are ca obiectiv vizitarea a ' ||
v_orase.count() || ' orase: ');

```



```

for i in v_orase.first..(v_orase.last-1) loop
    dbms_output.put (v_orase(i) || ', ');
end loop;
dbms_output.put_line (v_orase(v_orase.last) || '. ');

```

```

end loop;

```

```

end;
/

```

e. Anulați excursiile cu cele mai puține orașe vizitate.



The screenshot shows the Oracle SQL Developer interface. The top pane displays a PL/SQL script for deleting tours with the fewest cities. The bottom pane shows the query results in a table format.

```

from excursie_eis
where cod_excursie = v_toate_excursiile(j);

if v_orase.count() <= v_numar_min then v_numar_min := v_orase.count();
end if;

end loop;

for j in v_toate_excursiile.first..v_toate_excursiile.last loop

select orase into v_orase
from excursie_eis

```

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	anulata
2	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti')	disponibila
3	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Busteni', 'Sinaia')	disponibila
4	La aer curat	GRUPA233.TIP ORASE EIS('Sinaia', 'Azuga', 'Brasov')	anulata
5	Strainatate	GRUPA233.TIP ORASE EIS('Budabesta', 'Paris', 'Madrid')	disponibila

```

declare
v_orase excursie_eis.orase%type;
v_nr_orase number(4);
type vector is varray(5) of excursie_eis.cod_excursie%type;
v_toate_excursiile vector;
v_excursii_anulate vector;
v_numar_min number(4);

```

```

begin
select cod_excursie
bulk collect into v_toate_excursiile
from excursie_eis;

select orase into v_orase
from excursie_eis
where cod_excursie = v_toate_excursiile(1);

v_numar_min := v_orase.count();

```

```

for j in v_toate_excursiile.first..v_toate_excursiile.last loop

    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_toate_excursiile(j);

    if v_orase.count() <= v_numar_min then v_numar_min := v_orase.count();
    end if;

end loop;

for j in v_toate_excursiile.first..v_toate_excursiile.last loop

    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_toate_excursiile(j);

    if v_orase.count() = v_numar_min then
        update excursie_eis
        set status = 'anulata'
        where cod_excursie = v_toate_excursiile(j);
    end if;

end loop;

end;
/

```

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

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.

```

create or replace type tip_orase_eis is table of varchar2(60);

create table excursie_eis (cod_excursie number(4),
    denumire varchar2(20),
    orase tip_orase_eis,
    status varchar2(20)) nested table orase store as nested_table;

insert into excursie_eis
values (10, 'Prima mea excursie', tip_orase_eis('Brasov', 'Bucuresti'), 'disponibila');

```

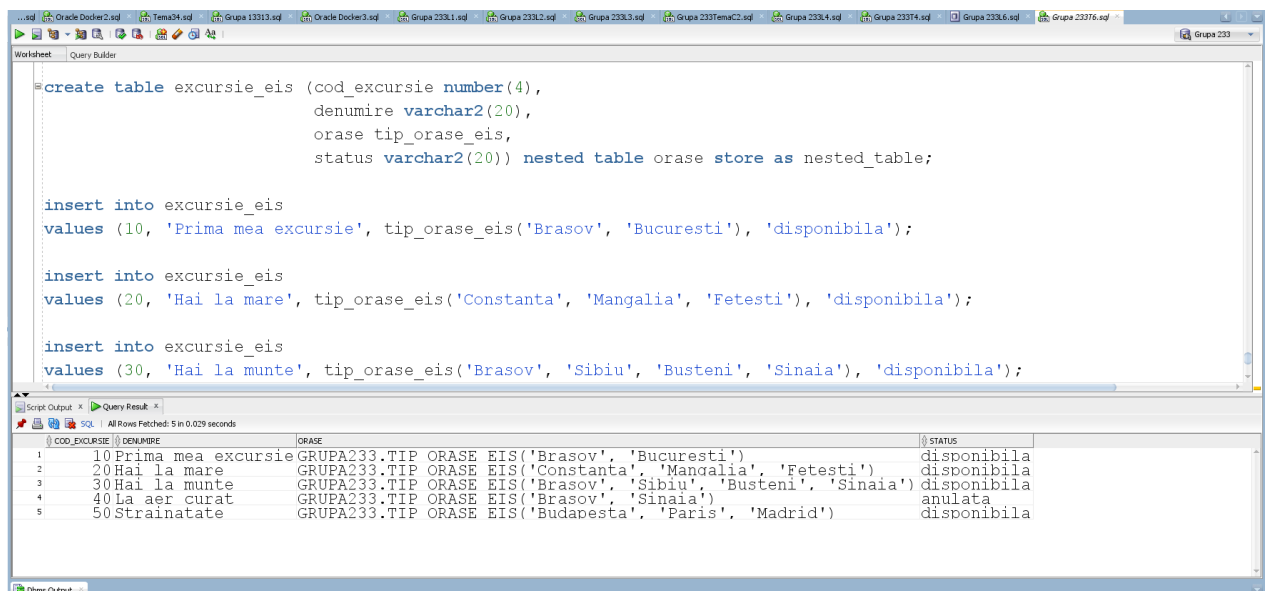
```
insert into excursie_eis
values (20, 'Hai la mare', tip_orase_eis('Constanta', 'Mangalia', 'Fetesti'), 'disponibila');
```

```
insert into excursie_eis
values (30, 'Hai la munte', tip_orase_eis('Brasov', 'Sibiu', 'Busteni', 'Sinaia'), 'disponibila');
```

```
insert into excursie_eis
values (40, 'La aer curat', tip_orase_eis('Brasov', 'Sinaia'), 'anulata');
```

```
insert into excursie_eis
values (50, 'Strainatate', tip_orase_eis('Budapesta', 'Paris', 'Madrid'), 'disponibila');
```

```
select * from excursie_eis;
```



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_a excursie_eis.cod_excursie%type := 20;
v_orase excursie_eis.orase%type;
```

```
begin
select orase into v_orase
from excursie_eis
where cod_excursie = v_cod_excursie_a;
```

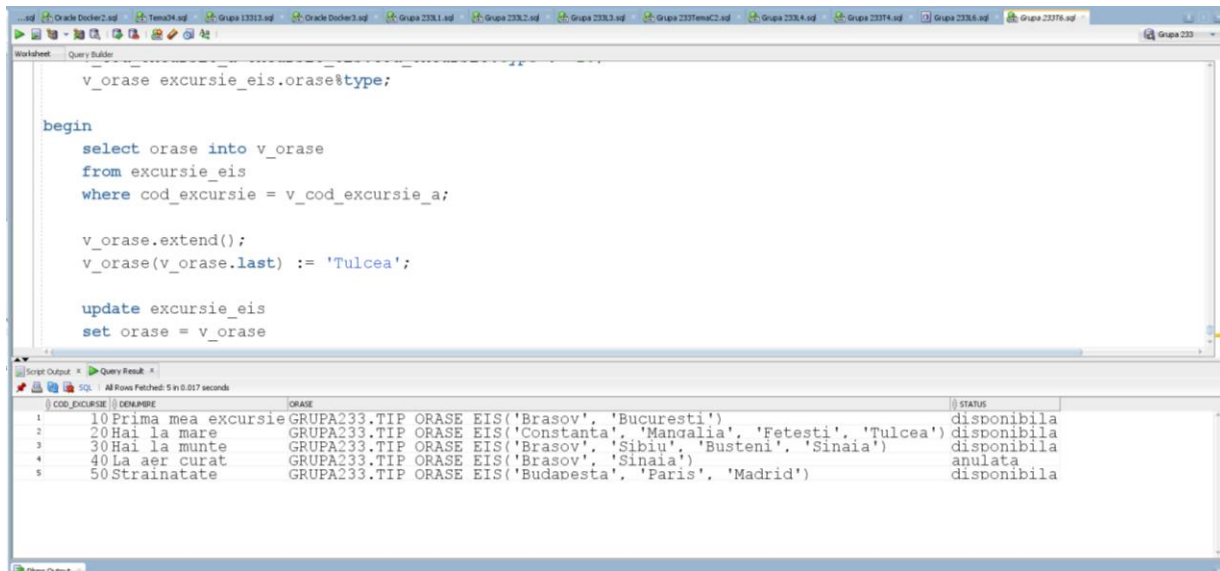
```
v_orase.extend();
v_orase(v_orase.last) := 'Tulcea';
```

```

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_a;

end;
/

```



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

```

declare
v_cod_excursie_b excursie_eis.cod_excursie%type := 40;
v_orase excursie_eis.orase%type;

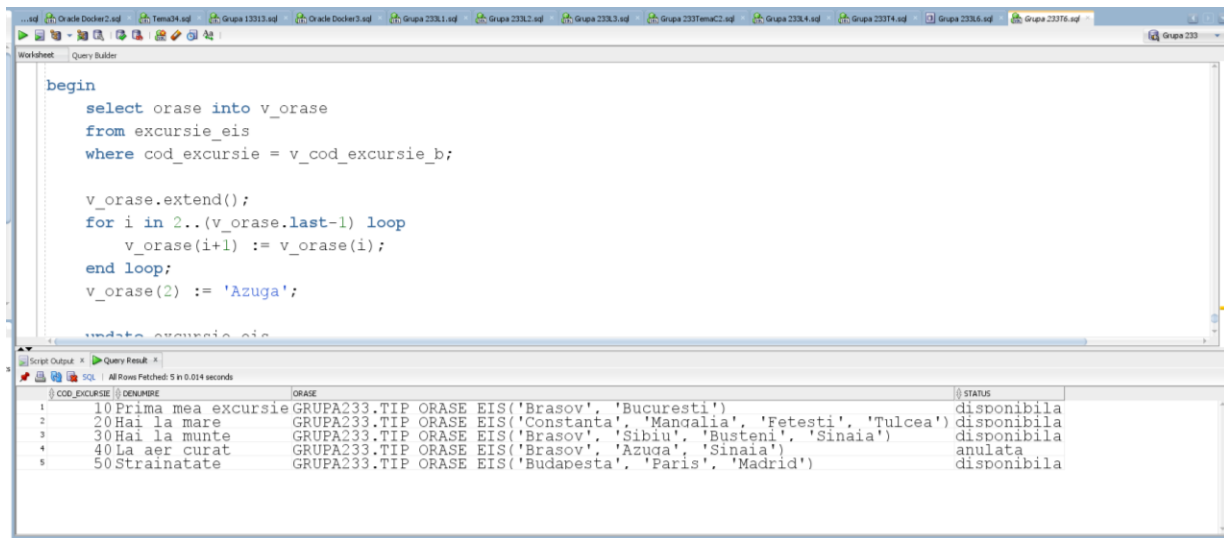
begin
select orase into v_orase
from excursie_eis
where cod_excursie = v_cod_excursie_b;

v_orase.extend();
for i in 2..(v_orase.last-1) loop
    v_orase(i+1) := v_orase(i);
end loop;
v_orase(2) := 'Azuga';

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_b;

end;
/

```



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

declare

```

v_cod_excursie_b excursie_eis.cod_excursie%type := 40;
v_orase excursie_eis.orase%type;
v_num_oras_p varchar2(60) := 'Brasov';
v_num_oras_d varchar2(60) := 'Sinaia';
v_pos_aux number(4);

```

begin

```

  select orase into v_orase
  from excursie_eis
  where cod_excursie = v_cod_excursie_b;

  for i in v_orase.first..v_orase.last loop
    if v_orase(i) = v_num_oras_p then v_pos_aux := i;
    end if;
  end loop;
  v_orase(v_pos_aux) := v_num_oras_d;
  for i in v_orase.first..v_orase.last loop
    if v_orase(i) = v_num_oras_d then v_pos_aux := i;
    end if;
  end loop;
  v_orase(v_pos_aux) := v_num_oras_p;

  update excursie_eis
  set orase = v_orase
  where cod_excursie = v_cod_excursie_b;

```

end;

/

```

declare
    v_cod_excursie_b excursie_eis.cod_excursie%type := 40;
    v_orase excursie_eis.orase%type;
    v_num_oras_p varchar2(60) := 'Brasov';
    v_num_oras_d varchar2(60) := 'Sinaia';
    v_pos_aux number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie_b;

    for i in v_orase.first..v_orase.last loop
        if v_orase(i) = v_num_oras_p then v_pos_aux := i;
        end if;
    end loop;
end;

```

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
10	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	disponibila
20	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti', 'Tulcea')	disponibila
30	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Sibiu', 'Busteni', 'Sinaia')	disponibila
40	La aer curat	GRUPA233.TIP ORASE EIS('Sinaia', 'Azuga', 'Brasov')	anulata
50	Strainatate	GRUPA233.TIP ORASE EIS('Budapesta', 'Paris', 'Madrid')	disponibila

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

```

declare
    v_cod_excursie_b excursie_eis.cod_excursie%type := 30;
    v_orase excursie_eis.orase%type;
    v_num_oras_p varchar2(60) := 'Sibiu';
    v_pos_aux number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie_b;

    for i in v_orase.first..v_orase.last loop

```

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
10	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	disponibila
20	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti', 'Tulcea')	disponibila
30	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Sibiu', 'Busteni', 'Sinaia')	disponibila
40	La aer curat	GRUPA233.TIP ORASE EIS('Sinaia', 'Azuga', 'Brasov')	anulata
50	Strainatate	GRUPA233.TIP ORASE EIS('Budapesta', 'Paris', 'Madrid')	disponibila

```

declare
    v_cod_excursie_b excursie_eis.cod_excursie%type := 30;
    v_orase excursie_eis.orase%type;
    v_num_oras_p varchar2(60) := 'Sibiu';
    v_pos_aux number(4);

```

```

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie_b;

```

```

for i in v_orase.first..v_orase.last loop
    if v_orase(i) = v_nume_oras_p then v_pos_aux := i;
    end if;
end loop;

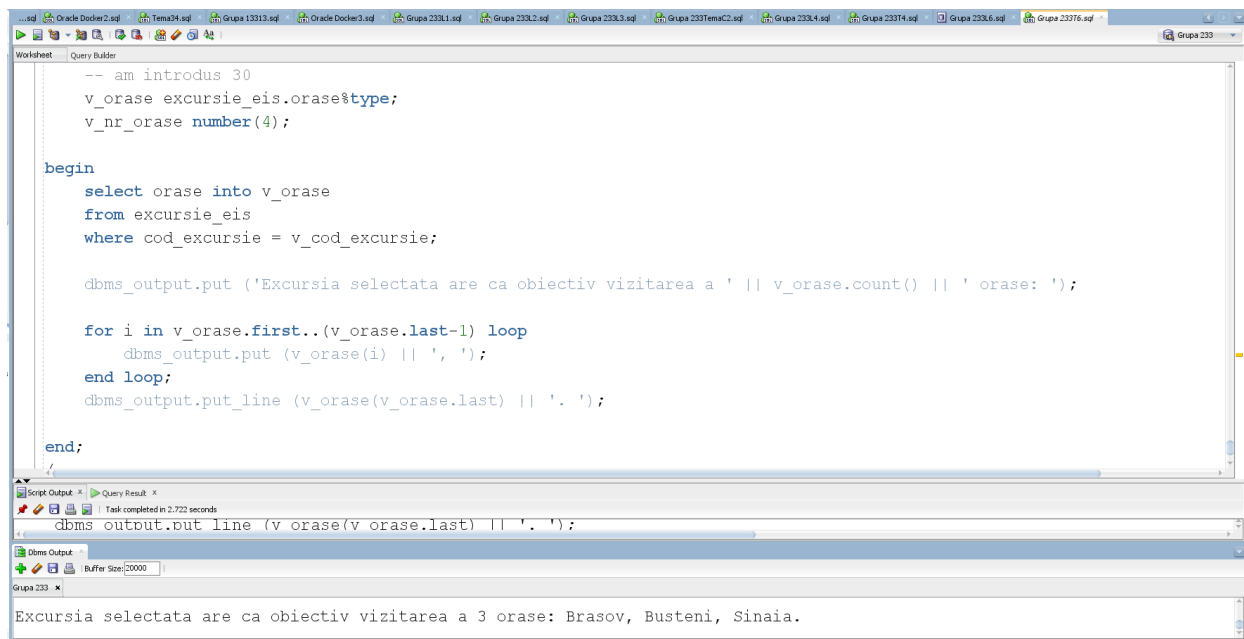
v_orase.delete(v_pos_aux);

update excursie_eis
set orase = v_orase
where cod_excursie = v_cod_excursie_b;

end;
/

```

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



The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Worksheet' tab. The script defines a procedure to select cities for a given excursion code and output the count and names. The script is as follows:

```

-- am introdus 30
v_orase excursie_eis.orase%type;
v_nr_orase number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie;

    dbms_output.put ('Excursia selectata are ca obiectiv vizitarea a ' || v_orase.count() || ' orase: ');

    for i in v_orase.first..(v_orase.last-1) loop
        dbms_output.put (v_orase(i) || ', ');
    end loop;
    dbms_output.put_line (v_orase(v_orase.last) || '. ');

end;

```

Below the script, the 'Script Output' tab shows the execution results. It indicates that the task was completed in 2.722 seconds. The output of the script is displayed in the 'DBMS Output' tab:

```

Excursia selectata are ca obiectiv vizitarea a 3 orase: Brasov, Busteni, Sinaia.

```

```

declare
    v_cod_excursie excursie_eis.cod_excursie%type := &p_cod_excursie;
    -- am introdus 30
    v_orase excursie_eis.orase%type;
    v_nr_orase number(4);

begin
    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_cod_excursie;

    dbms_output.put ('Excursia selectata are ca obiectiv vizitarea a ' || v_orase.count() || ' orase: ');

```

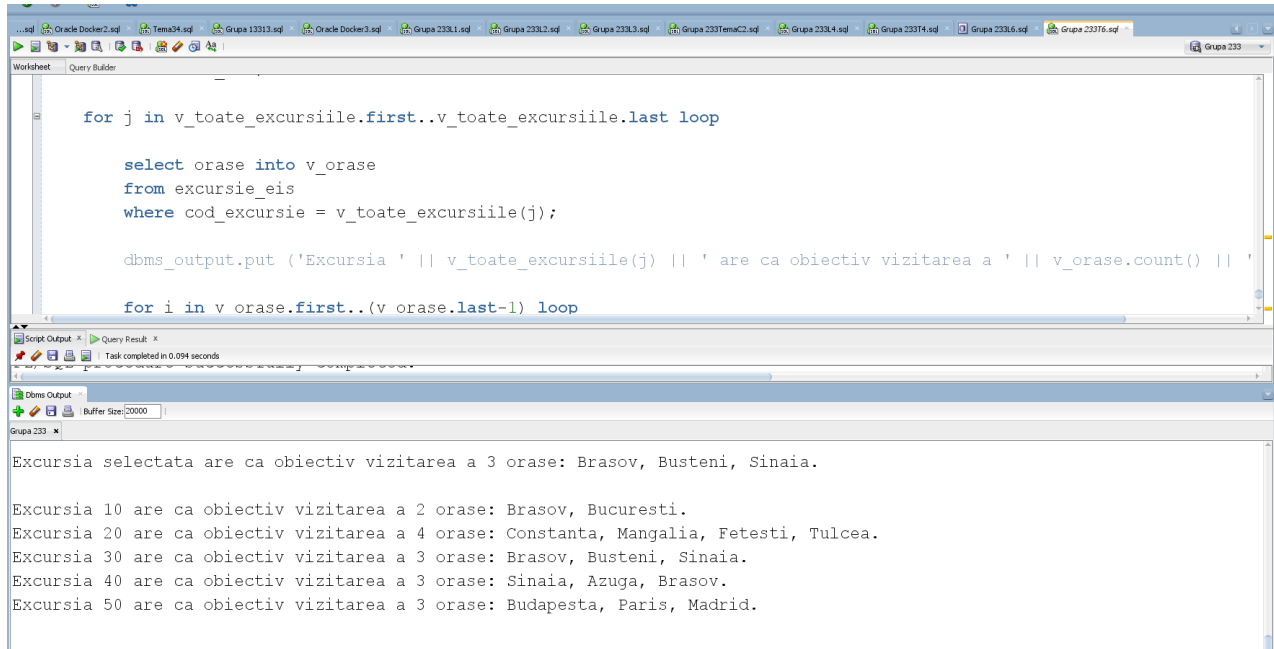
```

for i in v_orase.first..(v_orase.last-1) loop
    dbms_output.put (v_orase(i) || ', ');
end loop;
dbms_output.put_line (v_orase(v_orase.last) || '. ');

end;
/

```

d. Pentru fiecare excursie afișați lista orașelor vizitate.



```

declare
    v_orase excursie_eis.orase%type;
    v_nr_orase number(4);
    type tablou_imbricat is table of excursie_eis.cod_excursie%type;
    v_toate_excursiile tablou_imbricat;

```

begin

```

select cod_excursie
bulk collect into v_toate_excursiile
from excursie_eis;

```

```

for j in v_toate_excursiile.first..v_toate_excursiile.last loop

```

```

    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_toate_excursiile(j);

```



```

        dbms_output.put ('Excursia ' || v_toate_excursiile(j) || ' are ca obiectiv vizitarea a ' ||
v_orase.count() || ' orase: ');

        for i in v_orase.first..(v_orase.last-1) loop
            dbms_output.put (v_orase(i) || ', ');
        end loop;
        dbms_output.put_line (v_orase(v_orase.last) || '. ');

    end loop;

end;
/

```

e. Anulați excursiile cu cele mai puține orașe vizitate.

```

declare
    v_orase excursie_eis.orase%type;
    v_nr_orase number(4);
    type tablou_imbricat is table of excursie_eis.cod_excursie%type;
    v_toate_excursiile tablou_imbricat;
    v_numar_min number(4);

begin
    select cod_excursie
    bulk collect into v_toate_excursiile
    from excursie_eis;

    select orase into v_orase
    from excursie_eis
    where cod_excursie = v_toate_excursiile(1);

    v_numar_min := v_orase.count();

    for j in v_toate_excursiile.first..v_toate_excursiile.last loop

        select orase into v_orase
        from excursie_eis
        where cod_excursie = v_toate_excursiile(j);

        if v_orase.count() <= v_numar_min then v_numar_min := v_orase.count();
        end if;

    end loop;

    for j in v_toate_excursiile.first..v_toate_excursiile.last loop

        select orase into v_orase
        from excursie_eis

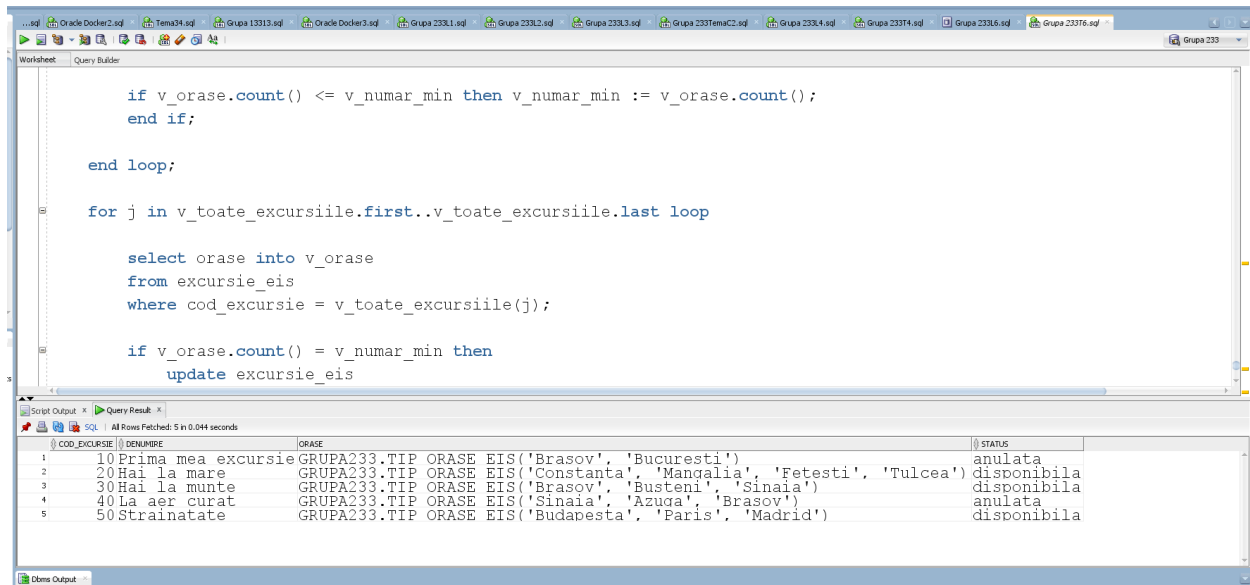
```

```
where cod_excursie = v_toate_excursiile(j);
```

```
if v_orase.count() = v_numar_min then  
  update excursie_eis  
  set status = 'anulata'  
  where cod_excursie = v_toate_excursiile(j);  
end if;
```

```
end loop;
```

```
end;  
/
```



The screenshot displays the Oracle SQL Developer environment. The main window shows a PL/SQL script with the following logic: it updates the status of excursions to 'anulata' (cancelled) if the number of available excursions (v_orase.count()) is equal to a minimum value (v_numar_min). The script iterates through a collection of excursion codes (v_toate_excursiile) and updates the status for each.

```
if v_orase.count() <= v_numar_min then v_numar_min := v_orase.count();  
end if;  
  
end loop;  
  
for j in v_toate_excursiile.first..v_toate_excursiile.last loop  
  
  select orase into v_orase  
  from excursie_eis  
  where cod_excursie = v_toate_excursiile(j);  
  
  if v_orase.count() = v_numar_min then  
    update excursie_eis
```

Below the script, the "Query Results" window shows the output of the script. It contains a table with columns: COD_EXCURSIE, DENUMIRE, ORASE, and STATUS. The results show five rows of excursion data, with the status updated to 'anulata' for the first row and 'disponibila' (available) for the others.

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
10	Prima mea excursie	GRUPA233.TIP ORASE EIS('Brasov', 'Bucuresti')	anulata
20	Hai la mare	GRUPA233.TIP ORASE EIS('Constanta', 'Mangalia', 'Fetesti', 'Tulcea')	disponibila
30	Hai la munte	GRUPA233.TIP ORASE EIS('Brasov', 'Busteni', 'Sinaia')	disponibila
40	La aer curat	GRUPA233.TIP ORASE EIS('Sinaia', 'Azuga', 'Brasov')	anulata
50	Strainatate	GRUPA233.TIP ORASE EIS('Budabesta', 'Paris', 'Madrid')	disponibila