

CERINTA

Definiți un **vector** 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

Tratați excepțiile care pot apărea.

REZOLVARI, EXPLICATII, POZE

- Explicatie: am definit vectorul si am facut tabelul.

```
create or replace type tip_orase_aux is varray(10) of varchar(20);
```

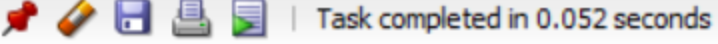
```
/
```

```
create table excursie_aux (  
    cod_excursie number(4) primary key,  
    denumire varchar2(20),  
    status varchar2(10)
```

```
);
```

```
/
```

```
alter table excursie_aux  
add (orase tip_orase_aux);
```



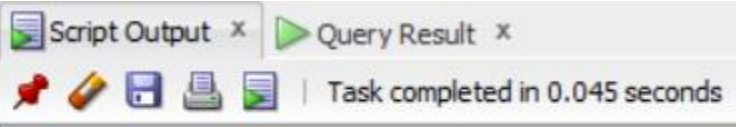
```
Type TIP_ORASE_AUX compiled

Table EXCURSIE_AUX created.

Table EXCURSIE_AUX altered.
```

- Punctul a: am inserat niste randuri si am dat COMMIT. Pentru excursia cu codul 5 nu am pus orase in vectorul de orase, ca sa putem trata in cerintele urmatoare si comportamentul blocurilor cand nu exista orase intr-o excursie.

```
begin
  -- a): Inserati 5 inregistrari in tabel.
  insert into excursie_aux
  values (1, 'Excursie 1', 'Disponibil', tip_orase_aux('Viena', 'Barcelona', 'Berlin'));
  insert into excursie_aux
  values (2, 'Excursie 2', 'Disponibil', tip_orase_aux('Tokyo', 'Brasov'));
  insert into excursie_aux
  values (3, 'Excursie 3', 'Disponibil', tip_orase_aux('Roma', 'New York'));
  insert into excursie_aux
  values (4, 'Excursie 4', 'Disponibil', tip_orase_aux('Londra', 'Amsterdam', 'Paris',
'Lisabona'));
  insert into excursie_aux
  values (5, 'Excursie 5', 'Anulat', tip_orase_aux());
  commit;
end;
```



```
PL/SQL procedure successfully completed.
```

- Punctul b

- Part 1 (adaugarea unui oras in lista): am citit input-ul (codul excursiei pentru care sa adaugam un oras, numele orasului de adaugat). Folosind o exceptie, am tratat cazul in care excursia cu codul respectiv nu exista.

Am preluat datele din vectorul de orase si le-am stocat intr-o variabila.

Folosind “extend”, am marit dimensiunea vectorului cu 1 ca sa mai adaug un oras; l-am pus la finalul vectorului (folosind “count” am aflat noua dimensiune a vectorului, iar indexul ultimei pozitii este dat de dimensiune).

Ca schimbarile sa aiba loc in tabel, am dat un “update” in care am dat replace la vectorul vechi cu noul vector modificat si am dat “commit”.

Am inclus o poza care arata ca a functionat codul pentru un cod care nu exista, si pentru un cod care exista.

declare

 t_cod_excursie excursie_aux.cod_excursie%type;

 t_oras varchar(20);

 t_orase excursie_aux.orase%type;

begin

 -- punctul b) - part 1: adaugati un oras nou de vizitat la finalul listei

 t_cod_excursie := &cod;

 t_oras := '&oras';

 select orase

 into t_orase

 from excursie_aux

 where cod_excursie = t_cod_excursie;

 t_orase.extend;

 t_orase(t_orase.count) := initcap(t_oras);

 update excursie_aux

 set orase = t_orase

 where cod_excursie = t_cod_excursie;

 commit;

 exception

 when no_data_found then


 dbms_output.put_line('Codul excursiei nu a fost gasit');

end;

Input: 0 "orice"

PL/SQL procedure successfully completed.


Dbms Output

 Buffer Size: 20000

SGBD-231 x

Codul excursiei nu a fost gasit


Input: 4 "Galati"

 Task completed in 3.567 seconds

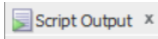
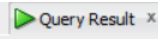

```
when no_data_found then
    dbms_output.put_line('Codul excursiei nu a fost gasit');
end;

PL/SQL procedure successfully completed.
```

Dbms Output

 Buffer Size: 20000

SGBD-231 x

 Script Output x  Query Result x
 All Rows Fetched: 5 in 0.006 seconds

	COD_EXCURSIE	DENUMIRE	STATUS	ORASE
1	1	Excursie 1	Disponibil	GRUPA231.TIP_ORASE_AUX('Viena', 'Barcelona', 'Berlin')
2	2	Excursie 2	Disponibil	GRUPA231.TIP_ORASE_AUX('Tokyo', 'Brasov')
3	3	Excursie 3	Disponibil	GRUPA231.TIP_ORASE_AUX('Roma', 'New York')
4	4	Excursie 4	Disponibil	GRUPA231.TIP_ORASE_AUX('Londra', 'Amsterdam', 'Paris', 'Lisabona', 'Galati')
5	5	Excursie 5	Anulat	GRUPA231.TIP_ORASE_AUX()

- Part 2 (adaugarea unui oras nou care sa fie al 2-lea de vizitat): am primit input-ul (cod, nume) si am tratat exceptia necesara (nu exista o excursie cu acel cod). Am preluat vectorul de orase din excursia respectiva si am stocat datele intr-o variabila vector.
Am folosit “extend” pentru a mari dimensiunea variabilei vector cu 1 (pentru ca adaugam un oras in plus); incepand de la a 2-a pozitie, am mutat toate orasele cu o pozitie in dreapta si am pus orasul primit ca input pe a 2-a pozitie. Am folosit un “update” pentru a aplica schimbarile in tabel si am dat “commit”.

```

declare
    t_cod_excursie excursie_aux.cod_excursie%type;
    t_oras          varchar(20);
    t_orase         excursie_aux.orase%type;
begin
    -- punctul b) - part 2: adaugati un oras nou, care sa fie al 2-lea de vizitat
    t_cod_excursie := &cod;
    t_oras          := '&oras';

    select orase
    into t_orase
    from excursie_aux
    where cod_excursie = t_cod_excursie;

    t_orase.extend;
    if t_orase.count = 1 then
        t_orase(1) := t_oras;
    else
        for i in reverse 3..t_orase.last loop
            t_orase(i) := t_orase(i - 1);
        end loop;
        t_orase(2) := initcap(t_oras);
    end if;

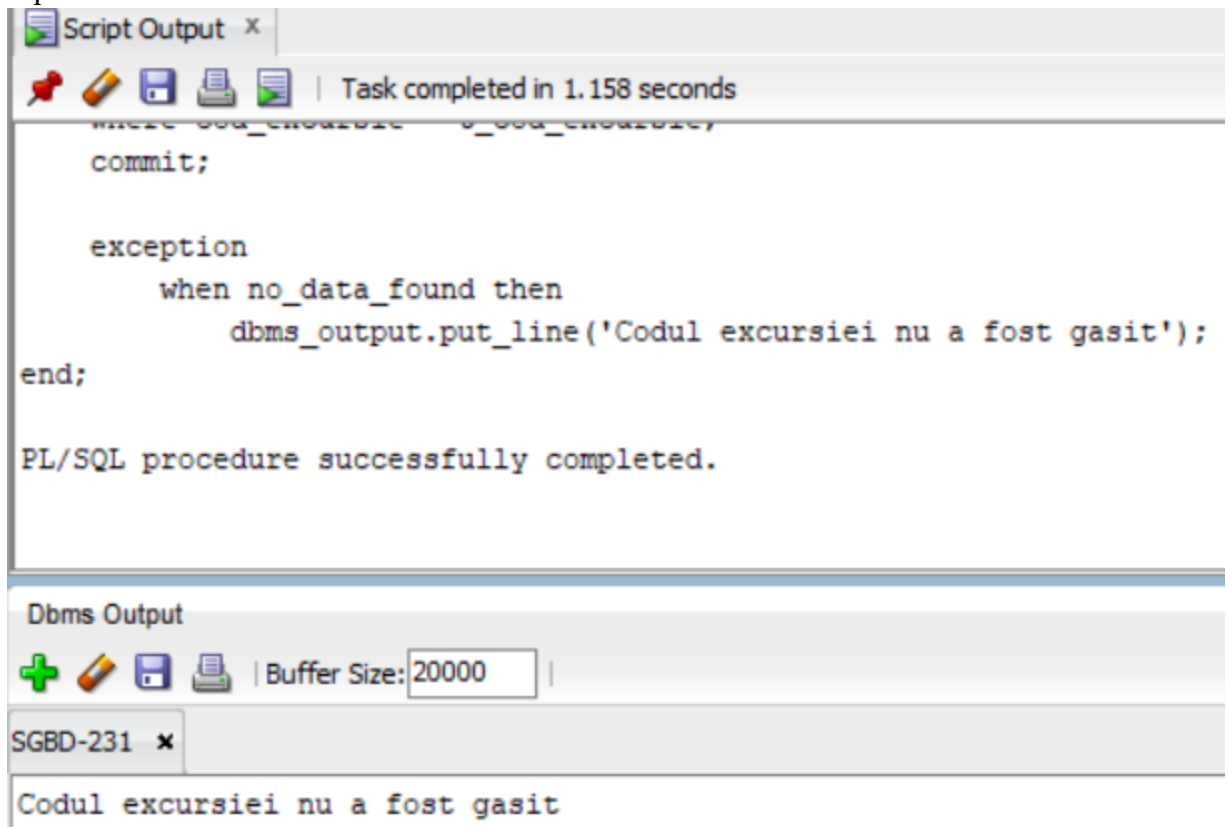
    update excursie_aux
    set orase = t_orase
    where cod_excursie = t_cod_excursie;
    commit;

    exception
        when no_data_found then
            dbms_output.put_line('Codul excursiei nu a fost gasit');

```

end;

Input: 0 "orice"



Script Output x

Task completed in 1.158 seconds

```
where cod_excursiei = v_cod_excursiei;  
commit;  
  
exception  
    when no_data_found then  
        dbms_output.put_line('Codul excursiei nu a fost gasit');  
end;  
  
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

SGBD-231 x

Codul excursiei nu a fost gasit

Input: 4 “Galati”

Script Output x

Task completed in 2.253 seconds

```
where cod_excursie = o_cod_excursie;  
commit;  
  
exception  
    when no_data_found then  
        dbms_output.put_line('Codul excursiei nu a fost gasit');  
end;  
  
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

SGBD-231 x

Script Output x Query Result x

All Rows Fetched: 5 in 0.015 seconds

	COD_EXCURSIE	DENUMIRE	STATUS	ORASE
1	1	Excursie 1	Disponibil	GRUPA231.TIP_ORASE_AUX('Viena', 'Barcelona', 'Berlin')
2	2	Excursie 2	Disponibil	GRUPA231.TIP_ORASE_AUX('Tokyo', 'Brasov')
3	3	Excursie 3	Disponibil	GRUPA231.TIP_ORASE_AUX('Roma', 'New York')
4	4	Excursie 4	Disponibil	GRUPA231.TIP_ORASE_AUX('Londra', 'Galati', 'Amsterdam', 'Paris', 'Lisabona')
5	5	Excursie 5	Anulat	GRUPA231.TIP_ORASE_AUX()

- Part 3 (inversarea a 2 orase): am primit input-ul (codul excursiei si cele 2 orase). Am tratat exceptia necesara (codul nu exista).
Am preluat orasele din tabel si le-am stocat intr-un vector. Daca exista cel putin 2 orase in vector, putem face o inversare. Am iterat prin tot vectorul si am salvat indecsii pentru ambele orase (care, la inceput, sunt 0). Daca cel putin unul din indecsi ramane 0, inseamna ca cel putin unul din orase nu a fost gasit (deci nu continuam cu inversarea).
Daca ambele orase au fost gasite, folosim o variabila auxiliara si realizam un swap. Apoi dam “update” si “commit”.

```

declare
    t_cod_excursie excursie_aux.cod_excursie%type;
    t_oras_1      varchar(20);
    t_oras_2      varchar(20);
    t_aux         varchar(20);
    t_orase       excursie_aux.orase%type;
    t_index_1     number(2);
    t_index_2     number(2);
begin
    -- punctul b) - part 3: inversati 2 orase al caror nume e specificat
    t_cod_excursie := &cod;
    t_oras_1      := '&oras';
    t_oras_2      := '&oras';

    select orase
    into t_orase
    from excursie_aux
    where cod_excursie = t_cod_excursie;

    if t_orase.count > 1 then
        for i in 1..t_orase.last loop
            if lower(t_orase(i)) = lower(t_oras_1) then
                t_index_1 := i;
            elsif lower(t_orase(i)) = lower(t_oras_2) then
                t_index_2 := i;
            end if;

            if t_index_1 > 0 and t_index_2 > 0 then
                t_aux := t_orase(t_index_1);
                t_orase(t_index_1) := t_orase(t_index_2);
                t_orase(t_index_2) := t_aux;
            end if;
        end loop;
    end if;

```



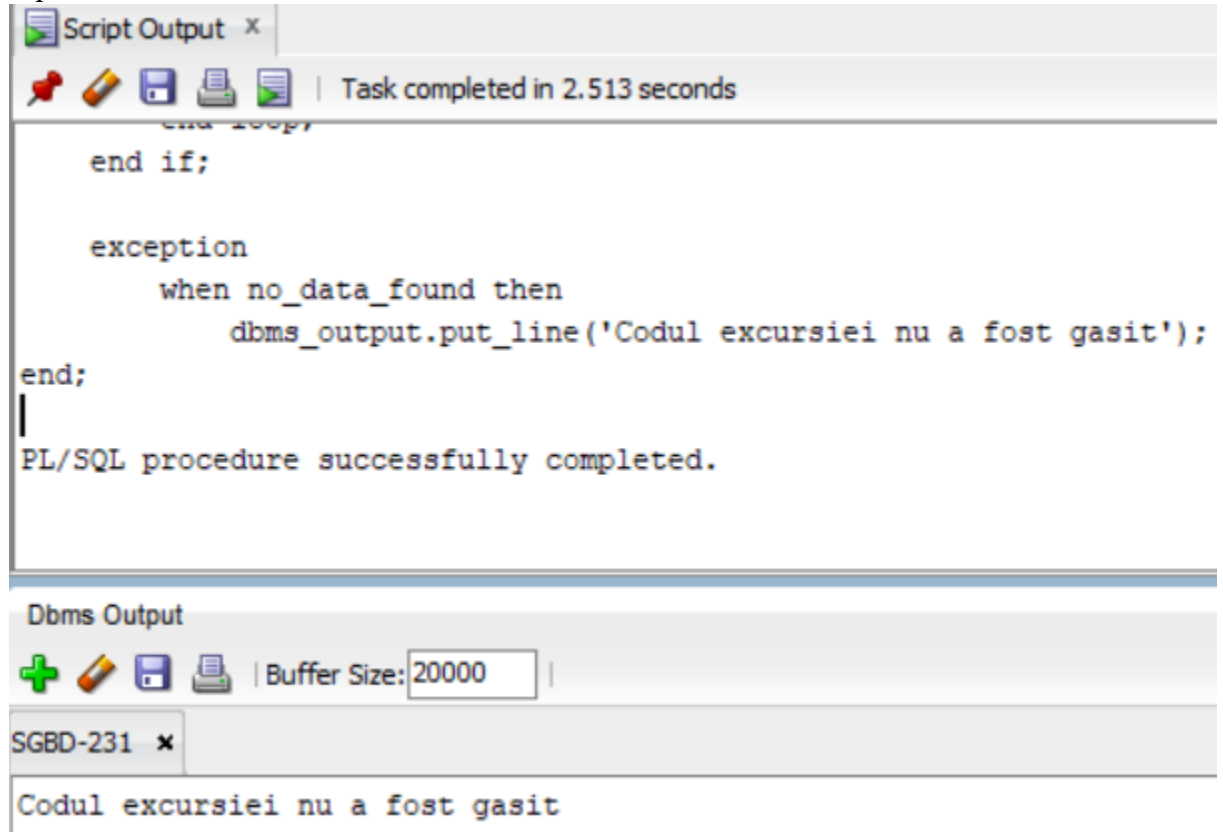
```

        update excursie_aux
        set orase = t_orase
        where cod_excursie = t_cod_excursie;
        exit;
    end if;
end loop;
end if;

exception
    when no_data_found then
        dbms_output.put_line('Codul excursiei nu a fost gasit');
end;

```

Input: 0 "orice" "orice"



The screenshot displays the Oracle SQL Developer interface. At the top, the 'Script Output' window shows the execution status: 'Task completed in 2.513 seconds'. Below this, the PL/SQL code is visible, including the exception handling block. The 'Dbms Output' window at the bottom shows the output of the script, which is 'Codul excursiei nu a fost gasit'. The 'SGBD-231' window is also visible at the bottom.

```

end loop;

end if;

exception
    when no_data_found then
        dbms_output.put_line('Codul excursiei nu a fost gasit');
end;

```

PL/SQL procedure successfully completed.

Dbms Output

Buffer Size: 20000

SGBD-231

Codul excursiei nu a fost gasit

Input: 4 "londra" "lisabona"

Script Output x

Task completed in 4.358 seconds

```
end loop;  
end if;  
  
exception  
    when no_data_found then  
        dbms_output.put_line('Codul excursiei nu a fost gasit');  
end;  
  
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

SGBD-231 x

Script Output x Query Result x

All Rows Fetched: 5 in 0.011 seconds

	COD_EXCURSIE	DENUMIRE	STATUS	ORASE
1	1	Excursie 1	Disponibil	GRUPA231.TIP_ORASE_AUX('Viena', 'Barcelona', 'Berlin')
2	2	Excursie 2	Disponibil	GRUPA231.TIP_ORASE_AUX('Tokyo', 'Brasov')
3	3	Excursie 3	Disponibil	GRUPA231.TIP_ORASE_AUX('Roma', 'New York')
4	4	Excursie 4	Disponibil	GRUPA231.TIP_ORASE_AUX('Lisabona', 'Amsterdam', 'Paris', 'Londra')
5	5	Excursie 5	Anulat	GRUPA231.TIP_ORASE_AUX()

- Part 4 (stergerea unui oras): verificarea inputului, preluarea datelor, exceptia pentru codul excursiei, etc.
Daca exista cel putin un oras, putem continua (daca nu sunt orase, nu e nimic de sters). Iteram prin tot vectorul pana gasim orasul cu denumirea primita ca input.
Odata ce am gasit orasul, incepand cu orasul din dreapta, mutam toate orasele spre stanga cu o pozitie, apoi eliminam ultimul element din vector folosind "trim".
Daca orasul nu a fost gasit (variabila de index ramane 0), nu facem nimic.
Finalizam cu un "update" si un "commit".

```

declare
    t_cod_excursie excursie_aux.cod_excursie%type;
    t_oras         varchar(20);
    t_orase        excursie_aux.orase%type;
    t_index        number(2);
begin
    -- punctul b) - part 4: eliminati din lista un oras
    t_cod_excursie := &cod;
    t_oras         := '&oras';

    select orase
    into t_orase
    from excursie_aux
    where cod_excursie = t_cod_excursie;

    if t_orase.count > 0 then
        for i in 1..t_orase.last loop
            if lower(t_orase(i)) = lower(t_oras) then
                t_index := i;
                exit;
            end if;
        end loop;

        if t_index > 0 then
            for i in t_index..(t_orase.count - 1) loop
                t_orase(i) := t_orase(i + 1);
            end loop;
            t_orase.trim(1);
        end if;

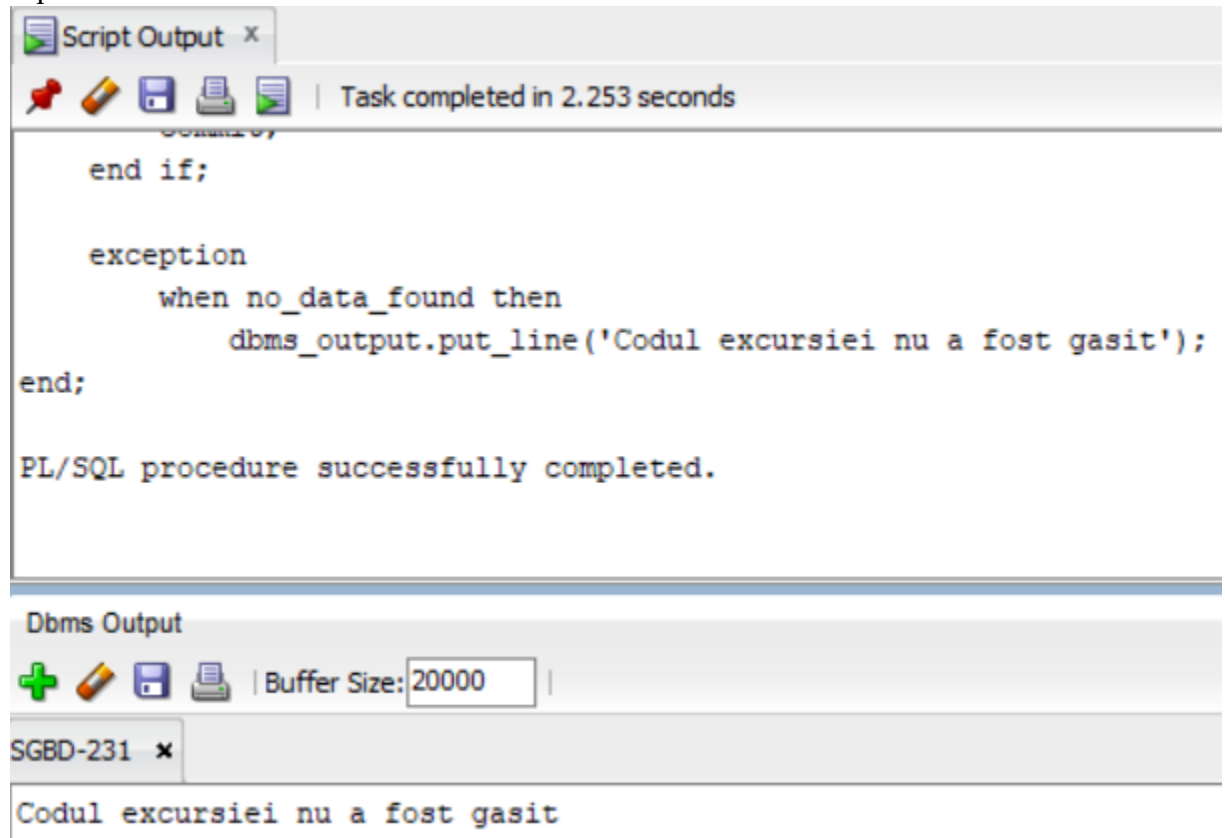
        update excursie_aux

```

```
set orase = t_orase
where cod_excursie = t_cod_excursie;
commit;
end if;

exception
when no_data_found then
dbms_output.put_line('Codul excursiei nu a fost gasit');
end;
```

Input: 0 "orice"



The screenshot displays the Oracle SQL Developer interface. At the top, a 'Script Output' window shows the execution of a PL/SQL script. The script code is visible, including an 'end if;' statement, an 'exception' block with a 'when no_data_found' condition that outputs 'Codul excursiei nu a fost gasit', and an 'end;' statement. Below the script code, the message 'PL/SQL procedure successfully completed.' is displayed. The 'Task completed in 2.253 seconds' is also shown. Below the 'Script Output' window, the 'Dbms Output' window is visible. It shows the output of the script, which is 'Codul excursiei nu a fost gasit'. The 'Buffer Size' is set to 20000. The window title is 'SGBD-231'.

Script Output x

Task completed in 2.253 seconds

```
end if;

exception
when no_data_found then
dbms_output.put_line('Codul excursiei nu a fost gasit');
end;

PL/SQL procedure successfully completed.
```

Dbms Output






Buffer Size: 20000

SGBD-231 x

Codul excursiei nu a fost gasit

Input: 4 "amsterDAM"

Script Output x





 | Task completed in 2.496 seconds

```
end if;

exception
    when no_data_found then
        dbms_output.put_line('Codul excursiei nu a fost gasit');
end;
```





PL/SQL procedure successfully completed.

Dbms Output

 | Buffer Size: 20000

SGBD-231 x

Script Output x | Query Result x

 | All Rows Fetched: 5 in 0.007 seconds

	COD_EXCURSIE	DENUMIRE	STATUS	ORASE
1	1	Excursie 1	Disponibil	GRUPA231.TIP_ORASE_AUX('Viena', 'Barcelona', 'Berlin')
2	2	Excursie 2	Disponibil	GRUPA231.TIP_ORASE_AUX('Tokyo', 'Brasov')
3	3	Excursie 3	Disponibil	GRUPA231.TIP_ORASE_AUX('Roma', 'New York')
4	4	Excursie 4	Disponibil	GRUPA231.TIP_ORASE_AUX('Lisabona', 'Paris', 'Londra')
5	5	Excursie 5	Anulat	GRUPA231.TIP_ORASE_AUX()

- Punctul c (afisarea nr. de orase si numele oraselor pentru o anumita excursie):
primim codul ca si input (tratam exceptia de NO_DATA_FOUND), selectam
orasele si le punem intr-o variabila vector, afisam numarul de orase cu “count” si
apoi iteram prin vector si afisam fiecare oras.

```
declare
    t_cod_excursie excursie_aux.cod_excursie%type;
    t_orase         excursie_aux.orase%type;
begin
    -- punctul c): afisati nr. de orase si numele oraselor pt. o excursie
    t_cod_excursie := &cod;

    select orase
    into t_orase
    from excursie_aux
    where cod_excursie = t_cod_excursie;

    dbms_output.put_line(t_orase.count || ' orase');
    for i in 1..t_orase.count loop
        dbms_output.put_line(t_orase(i) || ' ');
    end loop;

    exception
        when no_data_found then
            dbms_output.put_line('Codul excursiei nu a fost gasit');
end;
```

Input: 0

The screenshot displays the Oracle SQL Developer interface. At the top, the 'Script Output' and 'Query Result' tabs are visible. Below them, a status bar indicates 'Task completed in 1.837 seconds'. The main editor shows a PL/SQL procedure with a loop and an exception handler. The procedure successfully completed, as indicated by the message 'PL/SQL procedure successfully completed.' in the 'Script Output' pane. The 'Dbms Output' pane shows the output of the procedure, which is 'Codul excursiei nu a fost gasit'.

```
dbms_output.put_line('Codul excursiei nu a fost gasit');  
end loop;  
  
exception  
    when no_data_found then  
        dbms_output.put_line('Codul excursiei nu a fost gasit');  
end;  
  
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

SGBD-231 x

Codul excursiei nu a fost gasit

Input: 1

The screenshot displays the Oracle SQL Developer interface. At the top, the 'Script Output' and 'Query Result' tabs are visible. Below them, a status bar indicates 'Task completed in 0.538 seconds'. The main editor shows the same PL/SQL procedure as in the previous screenshot. The procedure successfully completed, as indicated by the message 'PL/SQL procedure successfully completed.' in the 'Script Output' pane. The 'Dbms Output' pane shows the output of the procedure, which is '3 orase', 'Viena', 'Barcelona', and 'Berlin'.

```
dbms_output.put_line('Codul excursiei nu a fost gasit');  
end loop;  
  
exception  
    when no_data_found then  
        dbms_output.put_line('Codul excursiei nu a fost gasit');  
end;  
  
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

SGBD-231 x

3 orase
Viena
Barcelona
Berlin

- Punctul d (afisarea oraselor pentru fiecare excursie): am folosit o variabila de tip “record”, care sa retina un ID si un vector de orase.

Am folosit “bulk collect” pentru a aduna toate datele din tabel in variabila, apoi am iterat prin aceasta, afisand ID-ul si orasele respective.

In cazul in care pentru o anumita excursie nu exista orase, am afisat “NU_SUNT_ORASE”.

declare

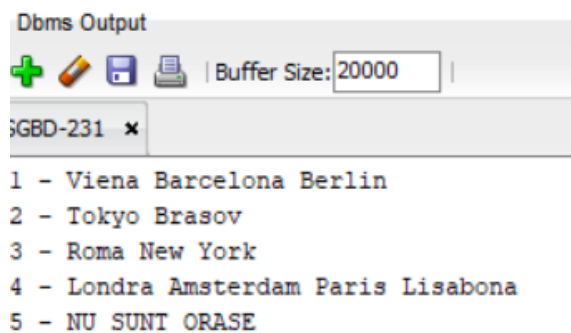
```
type t_info_cod_orase is record (  
    cod   excursie_aux.cod_excursie%type,  
    orase excursie_aux.orase%type  
);  
type t_tablou_info is table of t_info_cod_orase;  
t_info t_tablou_info;
```

begin

```
-- punctul d) pt. fiecare excursie afisati lista oraselor vizitate  
select cod_excursie, orase  
bulk collect into t_info  
from excursie_aux;
```

```
for i in t_info.first..t_info.last loop  
    dbms_output.put(t_info(i).cod || ' - ');  
    if (t_info(i).orase.count) > 0 then  
        for j in 1..t_info(i).orase.last loop  
            dbms_output.put(t_info(i).orase(j) || ' ');  
        end loop;  
    else  
        dbms_output.put_line('NU_SUNT_ORASE');  
    end if;  
    dbms_output.put_line("");  
end loop;
```

end;



```
Dbms Output  
+ | Buffer Size: 20000 |  
JGBD-231 x  
1 - Viena Barcelona Berlin  
2 - Tokyo Brasov  
3 - Roma New York  
4 - Londra Amsterdam Paris Lisabona  
5 - NU_SUNT_ORASE
```


- Punctul e (anularea excursiilor cu cele mai putine orase vizitate): pentru acest subpunct am exclus excursia cu codul 5 (delete from excursie_aux where cod_excursie = 5).

Ca la subpunctul trecut, am folosit o variabila record si “bulk collect” pentru a stoca toate datele din EXCURSIE_AUX intr-o variabila.

Am presupus ca numarul minim de orase pentru o excursie este numarul de orase din prima excursie, stocand acest numar intr-o variabila auxiliara. Apoi, am iterat prin fiecare excursie, numarand cate orase sunt si modificand valoarea variabilei auxiliare daca e necesar.

In final, am iterat iar prin vector, si am modificat excursiile care aveau numarul de orase egal cu numarul de orase retinut in variabila auxiliara, schimbând status-ul la “Anulat”.

declare

t_nr_minim_orase number(2);

type t_info_cod_orase is record (
cod excursie_aux.cod_excursie%type,
orase excursie_aux.orase%type
);

type t_tablou_info is table of t_info_cod_orase;

t_info t_tablou_info;

begin

-- punctul e) anulati excursiile cu cele mai putine orase vizitate

select cod_excursie, orase
bulk collect into t_info
from excursie_aux;

t_nr_minim_orase := t_info(1).orase.count;

for i in 2..t_info.last loop

if t_info(i).orase.count < t_nr_minim_orase then

t_nr_minim_orase := t_info(i).orase.count;

end if;

end loop;

for i in 1..t_info.last loop

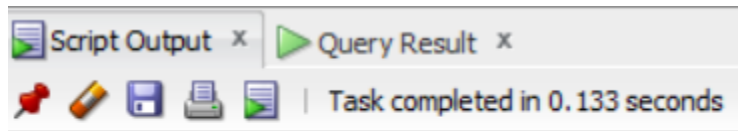
if t_info(i).orase.count = t_nr_minim_orase then

update excursie_aux

set status = 'Anulat'

where cod_excursie = t_info(i).cod;

```
end if;  
end loop;  
commit;  
end;
```



PL/SQL procedure successfully completed.

	COD_EXCURSIE	DENUMIRE	STATUS	ORASE
1	1	Excursie 1	Disponibil	GRUPA231.TIP_ORASE_AUX('Viena', 'Barcelona', 'Berlin')
2	2	Excursie 2	Anulat	GRUPA231.TIP_ORASE_AUX('Tokyo', 'Brasov')
3	3	Excursie 3	Anulat	GRUPA231.TIP_ORASE_AUX('Roma', 'New York')
4	4	Excursie 4	Disponibil	GRUPA231.TIP_ORASE_AUX('Londra', 'Amsterdam', 'Paris', 'Lisabona')