Tema Lab 2 PL/SQL – Ungureanu Matei

E2. 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 respectiv ă;

- 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.

create or replace type tip\_orase\_ums as varray(15) of varchar2(50);

/

create table excursie\_ums (

cod\_excursie NUMBER(4),

denumire VARCHAR2(20),

orase tip\_orase\_ums,

status varchar2(15));

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (1, 'Tur Romania', tip\_orase\_ums('Constanta', 'Tulcea', 'Galati', 'Iasi', 'Cluj', 'Timisoara',

'Hunedoara', 'Sinaia', 'Brasov', 'Predeal', 'Bucuresti'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (2, 'Tur Europa', tip\_orase\_ums('Bucuresti', 'Budapesta', 'Viena', 'Berna', 'Paris',

'Amsterdam', 'Berlin', 'Praga', 'Varsovia', 'Kiev'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (3, 'Tur USA Est', tip\_orase\_ums('Boston', 'New York City', 'Washington', 'Miami'),

'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (4, 'Tur USA Vest', tip\_orase\_ums('San Francisco', 'Las Vegas', 'Los Angeles',

'San Diego'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (5, 'Tur Asia Est', tip\_orase\_ums('Tokyo', 'Seoul', 'Beijing', 'Shanghai', 'Taipei', 'Hong Kong'),

'disponibilă');

declare

excursie excursie\_ums%rowtype;

index1 pls\_integer;

index2 pls\_integer;

temp varchar2(50);

oras1 varchar2(50) := 'Paris';

oras2 varchar2(50) := 'Berlin';

index\_de\_sters pls\_integer;

oras\_de\_sters varchar2(50) := 'Berna';

v\_cod number(4) := 5;

numar\_excursii pls\_integer;

excursie\_de\_anulat number(4);

minim pls\_integer := 100;

begin

select \* into excursie

from excursie\_ums

where cod\_excursie = 2;

excursie.orase.extend;

excursie.orase(excursie.orase.LAST) := 'Chisinau';

excursie.orase.extend;

excursie.orase(2) := 'Belgrad';

for i in 1..excursie.orase.count loop

if excursie.orase(i) = oras1 then

index1 := i;

elsif excursie.orase(i) = oras2 then

index2 := i;

end if;

if excursie.orase(i) = oras\_de\_sters then

index\_de\_sters := i;

end if;

end loop;

temp := excursie.orase(index1);

excursie.orase(index1) := excursie.orase(index2);

excursie.orase(index2) := temp;

for i in index\_de\_sters..excursie.orase.count - 1 loop

excursie.orase(i) := excursie.orase(i + 1);

end loop;

excursie.orase.trim;

update excursie\_ums

set orase = excursie.orase

where cod\_excursie = 2;

select \* into excursie

from excursie\_ums

where cod\_excursie = v\_cod;

dbms\_output.put('In excursia ' || v\_cod || ' sunt vizitate ' || excursie.orase.count || ' orase:');

for i in 1..excursie.orase.count loop

dbms\_output.put(excursie.orase(i));

end loop;

dbms\_output.put\_line('');

select count(\*)

into numar\_excursii

from excursie\_ums;

for i in 1..numar\_excursii loop

select \* into excursie

from excursie\_ums

where cod\_excursie = i;

if excursie.orase.count < minim then

minim := excursie.orase.count;

end if;

dbms\_output.put('Excursia ' || i || ':');

for j in 1..excursie.orase.count loop

dbms\_output.put(excursie.orase(j));

end loop;

dbms\_output.put\_line('');

end loop;

for i in 1..numar\_excursii loop

select \* into excursie

from excursie\_ums

where cod\_excursie = i;

if excursie.orase.count = minim then

update excursie\_ums

set status = 'anulată'

where cod\_excursie = i;

end if;

end loop;

end;

/

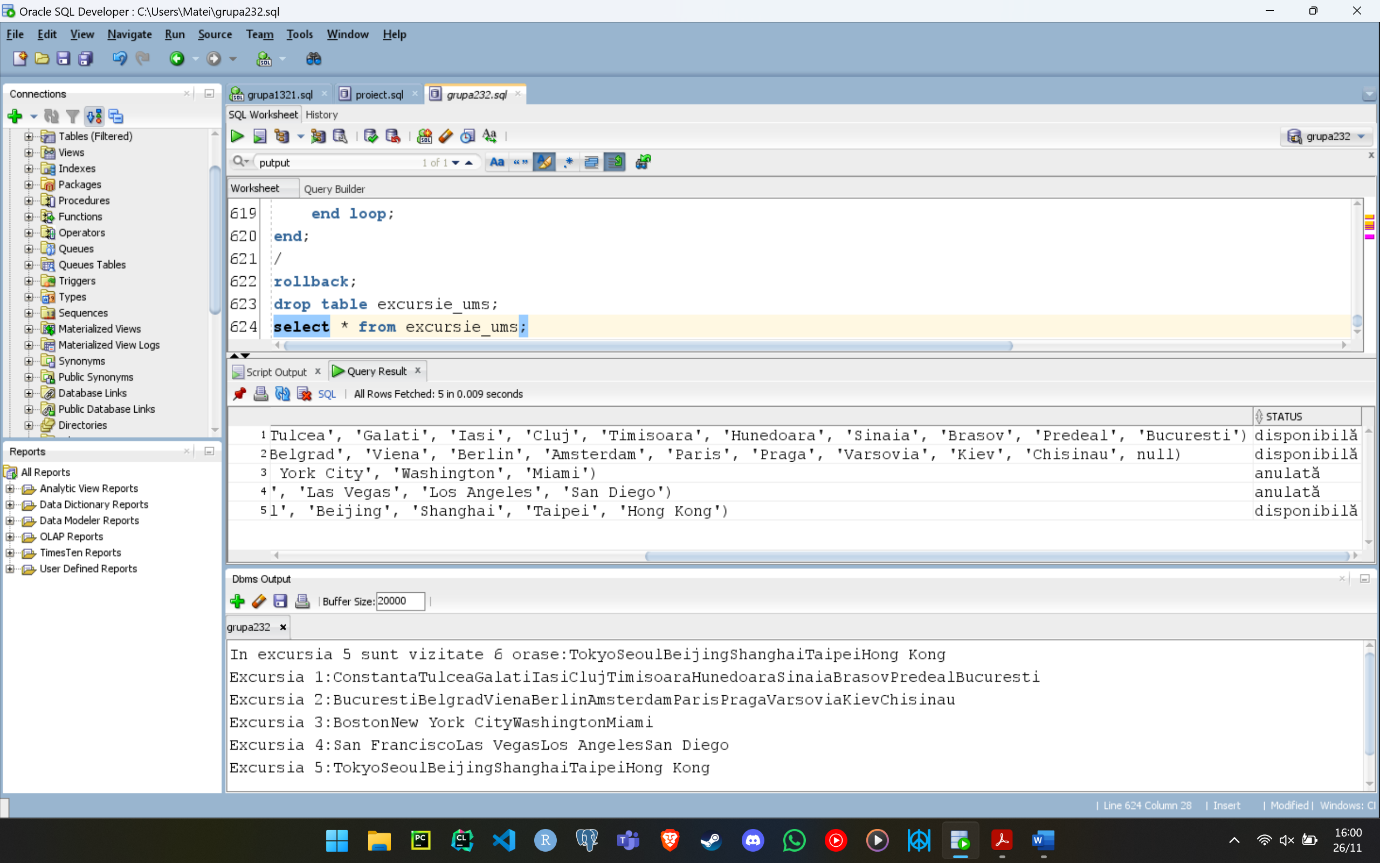
Rezultat:  
In excursia 5 sunt vizitate 6 orase:TokyoSeoulBeijingShanghaiTaipeiHong Kong

Excursia 1:ConstantaTulceaGalatiIasiClujTimisoaraHunedoaraSinaiaBrasovPredealBucuresti

Excursia 2:BucurestiBelgradVienaBerlinAmsterdamParisPragaVarsoviaKievChisinau

Excursia 3:BostonNew York CityWashingtonMiami

Excursia 4:San FranciscoLas VegasLos AngelesSan Diego

Excursia 5:TokyoSeoulBeijingShanghaiTaipeiHong Kong

Aici am o problema: apare un null la sfarsit in excursia 2 si nu stiu de ce. Bag 2 orase si fac extend de 2 ori, apoi elimin unul, adica mut tot de la ala mai la stanga, si fac trim sa sterg ultimul oras, dar vad ca tot e null.

E3. Rezolvați problema anterioară folosind un alt tip de colecție studiat.

create or replace type tip\_orase\_ums as table of varchar2(50);

/

create table excursie\_ums (

cod\_excursie NUMBER(4),

denumire VARCHAR2(20),

orase tip\_orase\_ums,

status varchar2(15))

nested table orase store as orase\_storage;

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (1, 'Tur Romania', tip\_orase\_ums('Constanta', 'Tulcea', 'Galati', 'Iasi', 'Cluj', 'Timisoara',

'Hunedoara', 'Sinaia', 'Brasov', 'Predeal', 'Bucuresti'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (2, 'Tur Europa', tip\_orase\_ums('Bucuresti', 'Budapesta', 'Viena', 'Berna', 'Paris',

'Amsterdam', 'Berlin', 'Praga', 'Varsovia', 'Kiev'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (3, 'Tur USA Est', tip\_orase\_ums('Boston', 'New York City', 'Washington', 'Miami'),

'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (4, 'Tur USA Vest', tip\_orase\_ums('San Francisco', 'Las Vegas', 'Los Angeles',

'San Diego'), 'disponibilă');

insert into excursie\_ums (cod\_excursie, denumire, orase, status)

values (5, 'Tur Asia Est', tip\_orase\_ums('Tokyo', 'Seoul', 'Beijing', 'Shanghai', 'Taipei', 'Hong Kong'),

'disponibilă');

declare

excursie excursie\_ums%rowtype;

index1 pls\_integer;

index2 pls\_integer;

temp varchar2(50);

oras1 varchar2(50) := 'Paris';

oras2 varchar2(50) := 'Berlin';

index\_de\_sters pls\_integer;

oras\_de\_sters varchar2(50) := 'Berna';

v\_cod number(4) := 5;

numar\_excursii pls\_integer;

excursie\_de\_anulat number(4);

minim pls\_integer := 100;

begin

select cod\_excursie, orase, status

into excursie.cod\_excursie, excursie.orase, excursie.status

from excursie\_ums

where cod\_excursie = 2;

excursie.orase.extend;

excursie.orase(excursie.orase.LAST) := 'Chisinau';

excursie.orase.extend;

excursie.orase(2) := 'Belgrad';

for i in 1..excursie.orase.count loop

if excursie.orase(i) = oras1 then

index1 := i;

elsif excursie.orase(i) = oras2 then

index2 := i;

end if;

if excursie.orase(i) = oras\_de\_sters then

index\_de\_sters := i;

end if;

end loop;

temp := excursie.orase(index1);

excursie.orase(index1) := excursie.orase(index2);

excursie.orase(index2) := temp;

excursie.orase.delete(index\_de\_sters);

update excursie\_ums

set orase = excursie.orase

where cod\_excursie = 2;

select cod\_excursie, orase, status

into excursie.cod\_excursie, excursie.orase, excursie.status

from excursie\_ums

where cod\_excursie = v\_cod;

dbms\_output.put('In excursia ' || v\_cod || ' sunt vizitate ' || excursie.orase.count || ' orase:');

for i in 1..excursie.orase.count loop

dbms\_output.put(excursie.orase(i));

end loop;

dbms\_output.put\_line('');

select count(\*)

into numar\_excursii

from excursie\_ums;

for i in 1..numar\_excursii loop

select cod\_excursie, orase, status

into excursie.cod\_excursie, excursie.orase, excursie.status

from excursie\_ums

where cod\_excursie = i;

if excursie.orase.count < minim then

minim := excursie.orase.count;

end if;

dbms\_output.put('Excursia ' || i || ':');

for j in 1..excursie.orase.count loop

dbms\_output.put(excursie.orase(j));

end loop;

dbms\_output.put\_line('');

end loop;

for i in 1..numar\_excursii loop

select cod\_excursie, orase, status

into excursie.cod\_excursie, excursie.orase, excursie.status

from excursie\_ums

where cod\_excursie = i;

if excursie.orase.count = minim then

update excursie\_ums

set status = 'anulată'

where cod\_excursie = i;

end if;

end loop;

end;

/

Rezultat:  
In excursia 5 sunt vizitate 6 orase:TokyoSeoulBeijingShanghaiTaipeiHong Kong

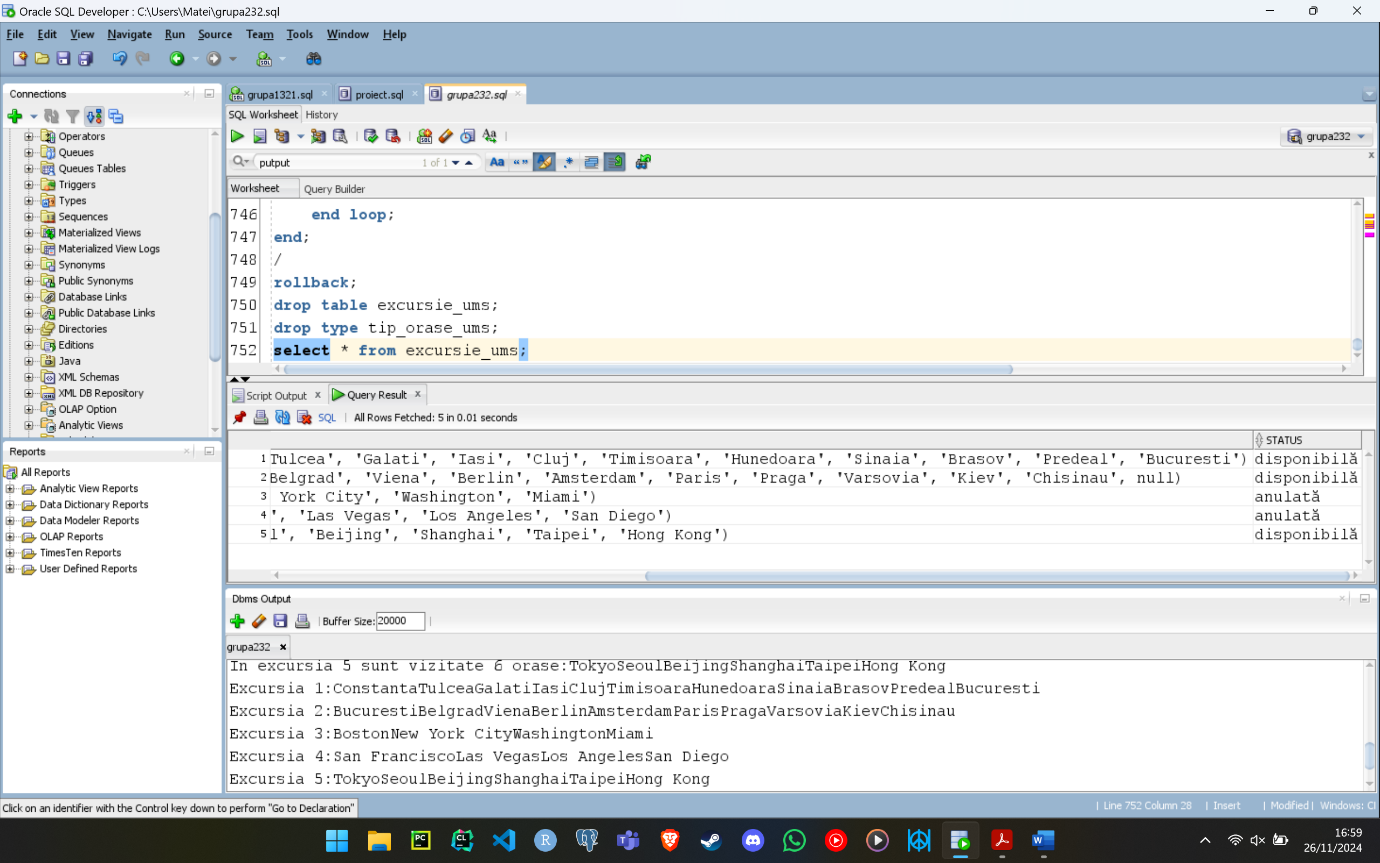
Excursia 1:ConstantaTulceaGalatiIasiClujTimisoaraHunedoaraSinaiaBrasovPredealBucuresti

Excursia 2:BucurestiBelgradVienaBerlinAmsterdamParisPragaVarsoviaKievChisinau

Excursia 3:BostonNew York CityWashingtonMiami

Excursia 4:San FranciscoLas VegasLos AngelesSan Diego

Excursia 5:TokyoSeoulBeijingShanghaiTaipeiHong Kong

Aici e aceeasi problema cu null, desi am folosit delete la stergere in loc de trim.

E4. Adaptați cerințele exercițiilor 9 și 10 pentru diagrama proiectului prezentată la materia Baze de Date din anul I. Rezolvați aceste exerciții în PL/SQL, folosind baza de date proprie.

9. (ENUNT ORIGINAL) Definiți tipul subordonati\_\*\*\* (vector, dimensiune maximă 10, menține numere). Creați tabelul manageri\_\*\*\* cu următoarele câmpuri: cod\_mgr NUMBER(10), nume VARCHAR2(20), lista subordonati\_\*\*\*. Introduceți 3 linii în tabel. Afișați informațiile din tabel. Ștergeți tabelul creat, apoi tipul.

(ENUNT ADAPTAT) Definiti tipul genuri (vector, dimensiune maxima 10, mentine siruri de caractere). Creati tabelul seriale cu urmatoarele campuri: cod\_serial NUMBER(10), nume VARCHAR2(50), gen genuri. Introduceți 3 linii in tabel. Afisati informatiile din tabel. Stergeti tabelul creat, apoi tipul.

create or replace type genuri as varray(10) of varchar2(50);

/

create table seriale (

cod\_serial number(10),

nume varchar2(50),

gen genuri);

declare

v\_genuri seriale.gen%TYPE;

numar\_seriale pls\_integer;

begin

insert into seriale

values (1, 'Game of Thrones', genuri('action', 'adventure', 'fantasy', 'serial drama',

'tragedy'));

insert into seriale

values (2, 'Andor', genuri('action-adventure', 'drama', 'political spy thriller',

'science fiction', 'dystopia'));

insert into seriale

values (3, 'Peaky Blinders', genuri('historical drama', 'crime drama'));

select count(\*)

into numar\_seriale

from seriale;

for i in 1..numar\_seriale loop

select gen into v\_genuri

from seriale

where cod\_serial=i;

dbms\_output.put('Serialul ' || i || ': ');

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

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

end loop;

dbms\_output.put\_line('');

end loop;

end;

/

select \* from seriale;

drop table seriale;

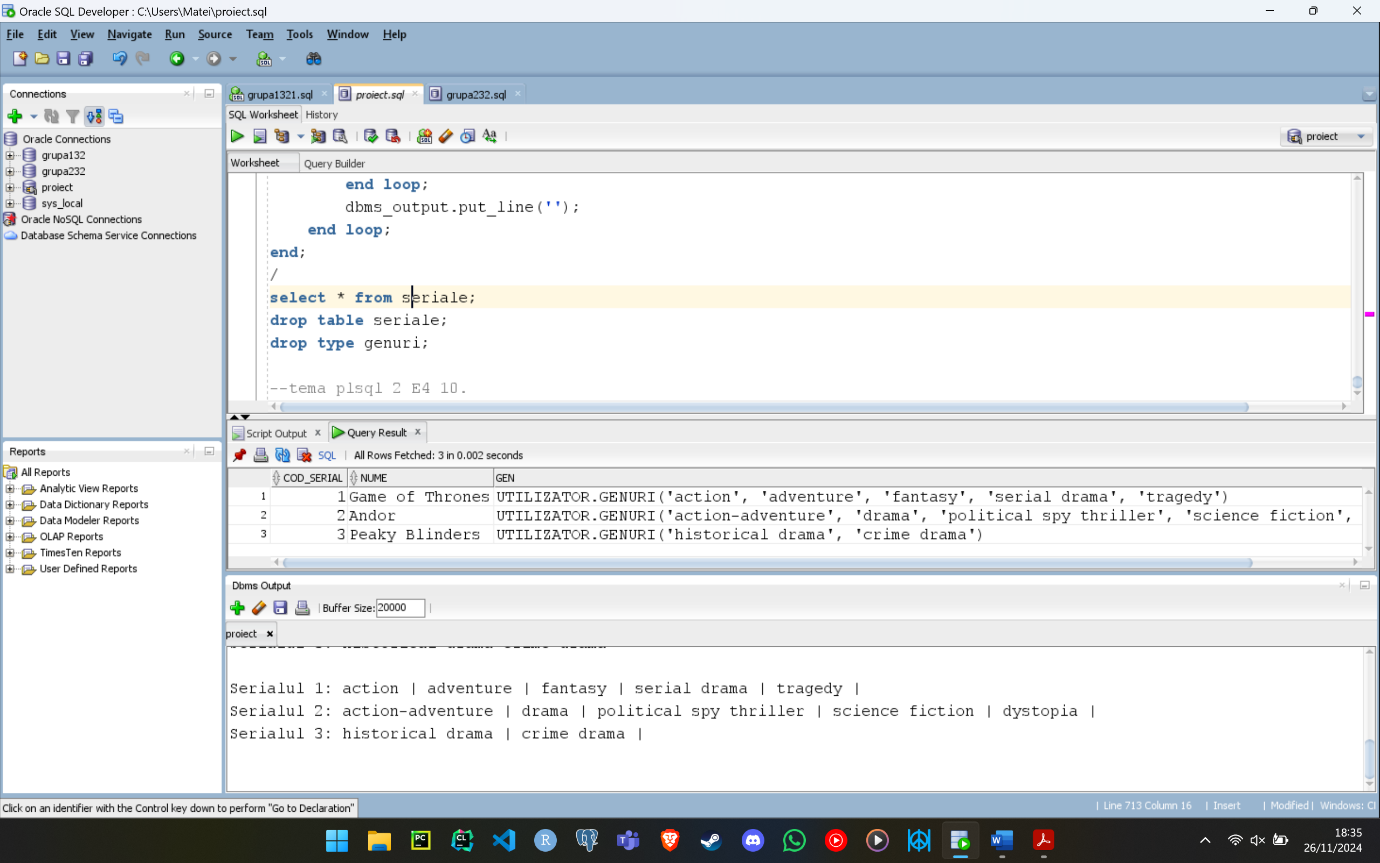
drop type genuri;

Rezultat:

Serialul 1: action | adventure | fantasy | serial drama | tragedy |

Serialul 2: action-adventure | drama | political spy thriller | science fiction | dystopia |

Serialul 3: historical drama | crime drama |



10. (ENUNT ORIGINAL) Creați tabelul emp\_test\_\*\*\* cu coloanele employee\_id și last\_name din tabelul employees. Adăugați în acest tabel un nou câmp numit telefon de tip tablou imbricat. Acest tablou va menține pentru fiecare salariat toate numerele de telefon la care poate fi contactat. Inserați o linie nouă în tabel. Actualizați o linie din tabel. Afișați informațiile din tabel. Ștergeți tabelul și tipul.

(ENUNT ADAPTAT) Creati tabelul filme\_test cu coloanele id\_film si nume\_film din tabelul filme. Adaugati in acest tabel un nou camp numit gen de tip tablou imbricat. Acest tablou va mentine pentru fiecare film toate genurile din care face parte. Inserati o linie noua in tabel. Actualizati o linie din tabel. Afisati informatiile din tabel. Stergeti tabelul si tipul.

create table filme\_test as

select id\_film, nume\_film

from filme;

create or replace type genuri is table of varchar(50);

/

alter table filme\_test

add (gen genuri)

nested table gen store as genuri\_storage;

insert into filme\_test

values (106, 'The Martian', genuri('science fiction', 'action', 'adventure', 'drama'));

update filme\_test

set gen = genuri('science fiction', 'action', 'adventure', 'fantasy')

where id\_film=105;

select a.nume\_film, b.\*

from filme\_test a, table (a.gen) b;

drop table filme\_test;

drop table genuri;

Rezultat:

The Martian science fiction

The Martian action

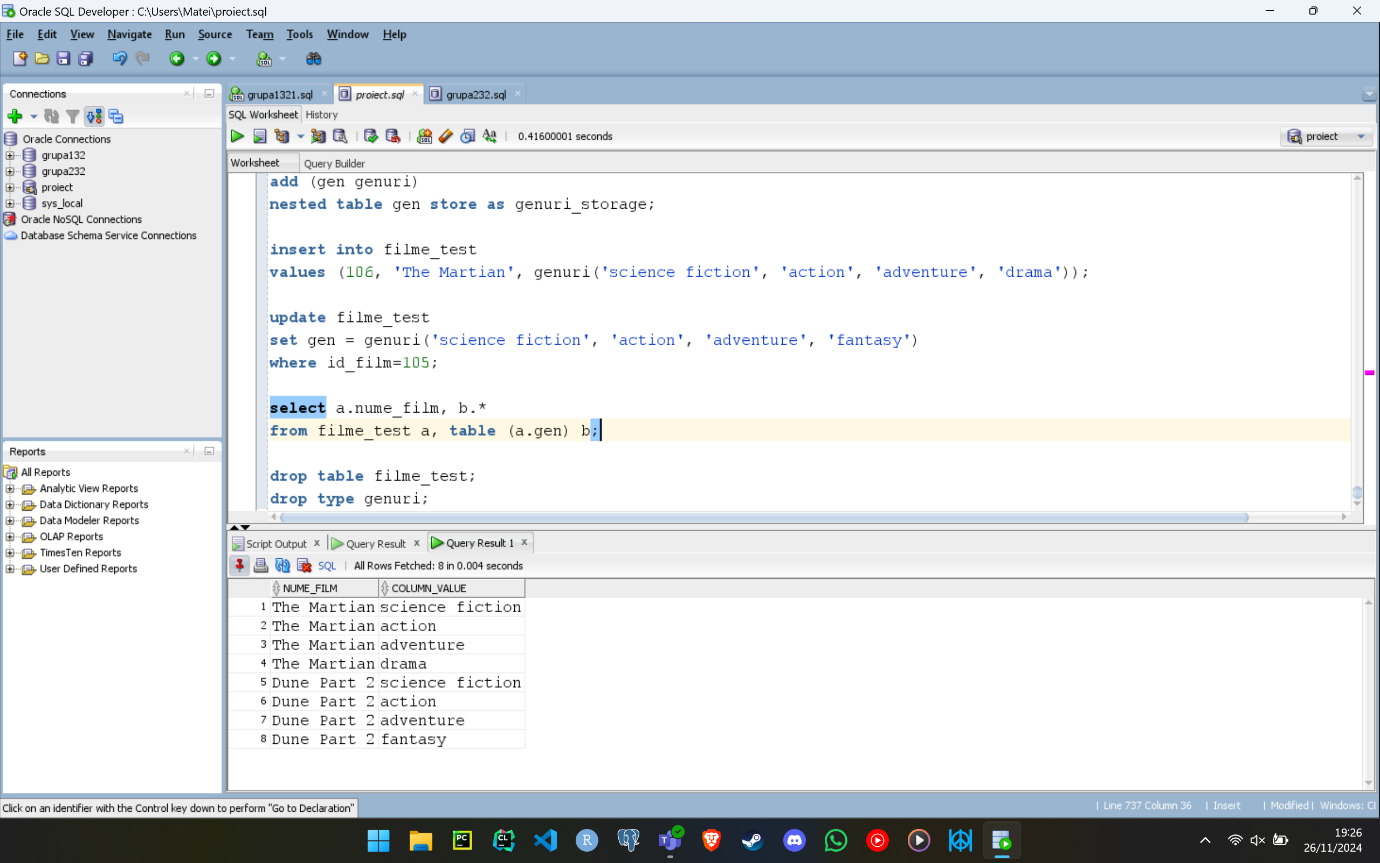
The Martian adventure

The Martian drama

Dune Part 2 science fiction

Dune Part 2 action

Dune Part 2 adventure

Dune Part 2 fantasy