

## **Gestiunea activității instituțiilor teatrale**

1. Proiectul își propune să proiecteze o bază de date, care să gestioneze o parte din problemele unor instituții teatrale dintr-o țară.

### **Regulile de funcționare ale modelului sunt următoarele:**

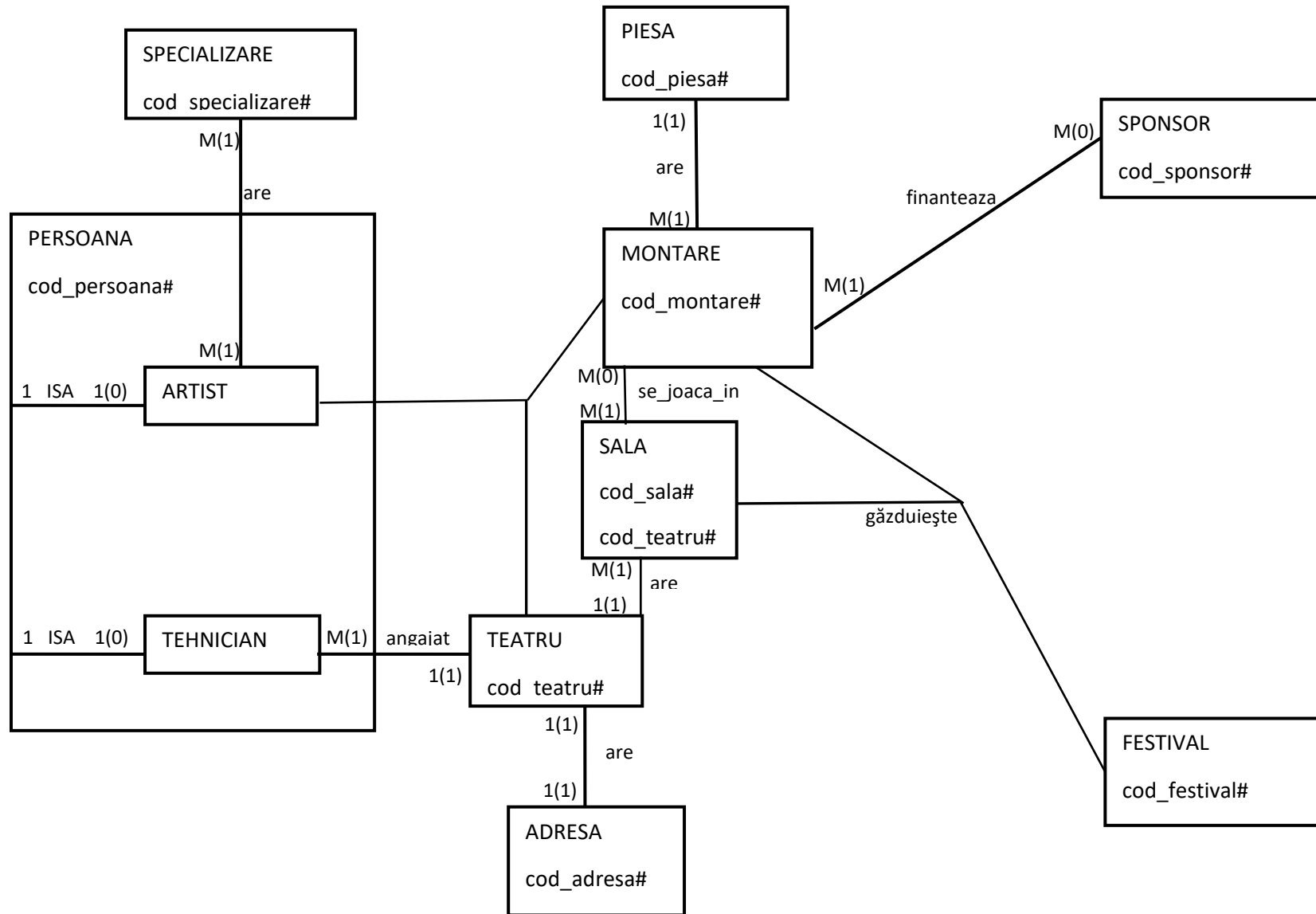
- 1) Teatrele pot monta piese.
- 2) Prin proiect vom înțelege punerea în scenă a unei piese de teatru (montarea unei piese).
- 3) Persoanele care lucrează la un astfel de proiect pot fi artiști: actori, regizori, scenariști, scenografi, compozitori, muzicieni, coregrafi sau personal tehnic: maestru sunet, maestru lumini, butafor și alții.
- 4) La realizarea unei montări, participă întreg personalul tehnic al teatrului unde se montează piesa.
- 5) Montarea unei piese poate beneficia de sume de bani din partea unor sponsori.
- 6) Montările pot fi prezentate atât în teatrul în care au fost montate, cât și în sălile altor teatre, în cadrul festivalurilor de teatru.

### **Restricțiile impuse asupra modelului sunt următoarele:**

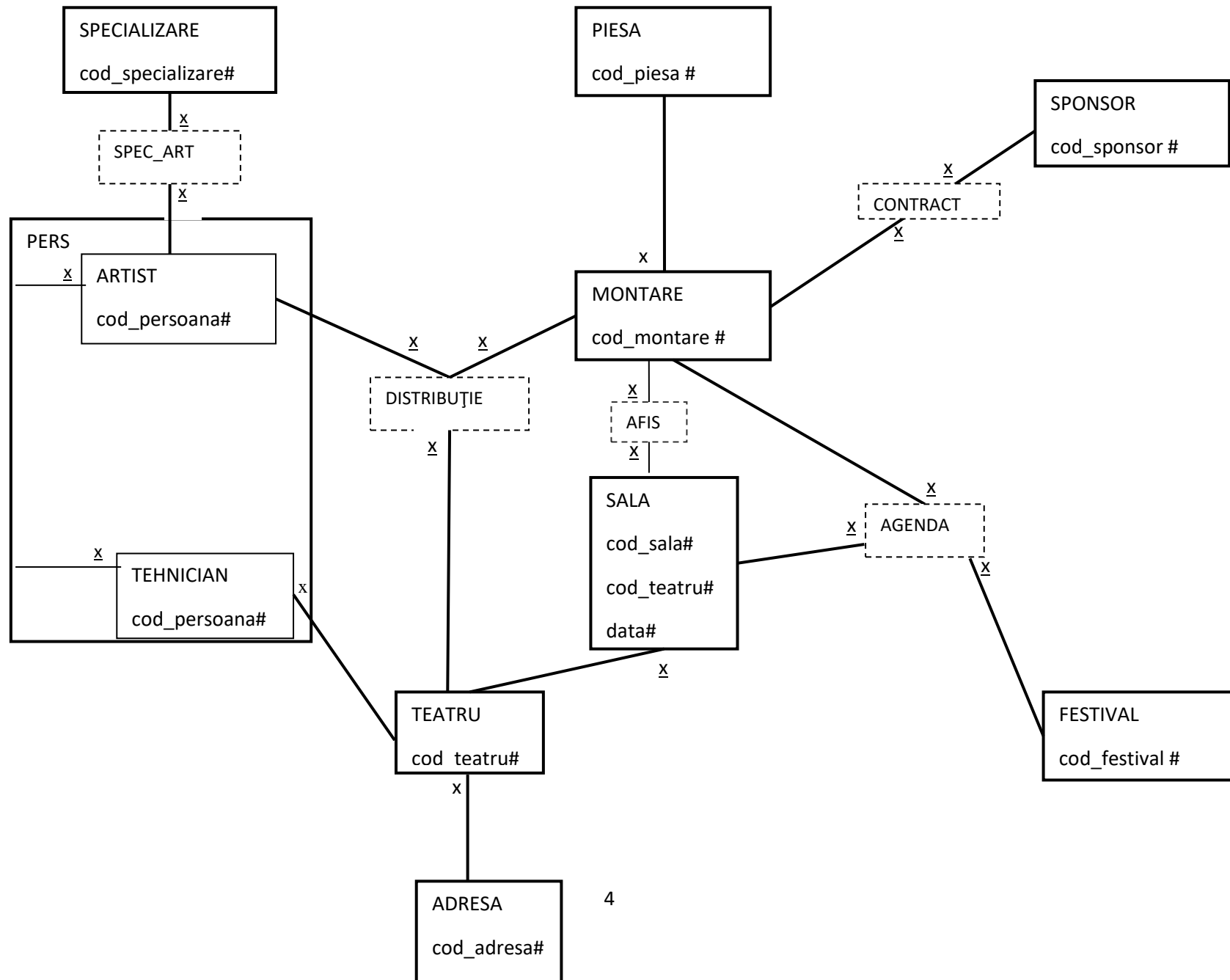
- 1) O instituție teatrală își are sediul la o singură adresă.
- 2) La o adresă, își are sediul o singură instituție teatrală.
- 3) Teatrele pot avea una sau mai multe săli în care se pot susține spectacole.
- 4) O sală aparține unui singur teatru.
- 5) O montare poate fi jucată într-una sau mai multe săli.
- 6) O piesă de teatru are cel puțin o montare.
- 7) O piesă poate fi montată la mai multe teatre și chiar la același teatru de mai multe ori, în regii diferite.
- 8) Un artist poate avea mai multe specializări.
- 9) Pentru fiecare specializare, există cel puțin un artist care o deține.
- 10) Un artist poate lucra în același timp la mai multe proiecte.
- 11) Un tehnician poate face parte din personalul unui singur teatru.
- 12) Din echipa tehnică a unui teatru face parte cel puțin un angajat.
- 13) O montare poate să nu fie prezentată în cadrul niciunui festival de specialitate, sau poate participa la mai multe festivaluri.

- 14) Reprezentațiile din cadrul unui festival sunt găzduite de teatre din localitatea în care se desfășoară festivalul.
- 15) Un teatru poate găzdui reprezentații într-un festival, doar în sălile în care s-a montat cel puțin o producție proprie.
- 16) Un sponsor finanțează unul sau mai multe proiecte.
- 17) Un proiect poate să nu fie sponsorizat, sau poate fi finanțat de mai mulți sponsori.
- 18) Într-o distribuție, există un unic actor care joacă fiecare rol.

## 2. Diagrama Entitate/Relație



### 3. Diagrama conceptuală



## Entitățile prezente în model sunt următoarele:

1) TEATRU = instituție a cărei activitate constă, în principal, în montarea și prezentarea de piese de teatru. Modelul de date gestionează doar o parte dintre instituțiile teatrale din România. **Cheia primară** asociată acestei entități este **cod\_teatru**.

2) PERSOANA = persoană direct implicată în activitatea de punere în scenă a pieselor într-un teatru. Pentru entitatea PERSOANĂ, **cheia primară** este **cod\_persoana**.

3) ARTIST = subentitate a entității PERSOANA, ce conține informații specifice artiștilor implicați în procesul de montare a unui spectacol de teatru (actori, regizori, scenariști, scenografi, compozitori, muzicieni, coregrafi). **Cheia primară** a entității este **cod\_persoana**, moștenită de la superentitate.

4) TEHNICIAN = subentitate a entității PERSOANA, ce conține informații specifice angajaților care se ocupă de asigurarea calității elementelor tehnice necesare pentru desfășurarea unui spectacol de teatru (sonorizare, lumini, realizarea unor elemente de decor). **Cheia primară** a entității este **cod\_persoana**, moștenită de la superentitate.

5) SPECIALIZARE = aria de activitate a unui artist. Printre instanțele acestei entități, regăsim: actor, regizor, scenarist, scenograf, compozitor, muzician, coregraf. Entității SPECIALIZARE îi asociem **cheia primară** **cod\_specializare**.

6) PIESA = operă literară destinată reprezentării scenice, care este montată cel puțin o dată la unul din teatrele din evidența modelului. **Cheia primară** asociată entității PIESA este **cod\_piesa**.

5) MONTARE = punere în scenă a unei piese, în cadrul unei instituții teatrale incluse în model. **Cheia primară** a entității MONTARE este **cod\_montare**.

7) SALA = entitate dependentă de TEATRU, care conține informații referitoare la un spațiu din incinta unui teatru, a cărui amenajare permite desfășurarea de spectacole în interiorul său. Aici se pot număra, pe lângă sălile a căror destinație principală sunt reprezentațiile, săli precum foaierele, sau chiar holuri ale teatrului. **Cheia primară** a acestei entități este compusă din **cod\_sala** și **cod\_teatru**.

8) ADRESA = informații privind localizarea fiecărei instituții teatrale ce face obiectul modelului proiectat. **Cheia primară** este **cod\_adresa**.

9) SPONSOR = persoană fizică sau juridică, care finanțează cel puțin un proiect aparținând uneia dintre instituțiile teatrale din cadrul modelului proiectat. **Cheia primară** a acestei entități este **cod\_sponsor**.

10) FESTIVAL = eveniment cultural, la care participă cel puțin una dintre montările pe care le are în evidență modelul. **Cheia primară** a entității este **cod\_festival**.

## Relațiile dintre entitățile modelului sunt următoarele:

1) Relația ARTIST\_are\_SPECIALIZARE desemnează asocierea dintre un artist și specializările pe care acesta le deține. Ținând cont de regulile 9 și 10 impuse asupra modelului, putem deduce tipul de cardinalitate al acestei relații: **cardinalitate minima 1:1 și cardinalitate maximă n:m**.

2) Relațiile PERSOANA\_ISA\_ARTIST și PERSOANA\_ISA\_TEHNICIAN reprezintă asocierile dintre o instanță a entității artist și calitatea acesteia de face parte din subentitățile ARTIST, respectiv TEHNICIAN. Cardinalitățile sunt cele specifice relațiilor dintre superentități și subentități: **cardinalitate minima 1:0 și cardinalitate maxima 1:1**.

3) Relația ARTIST\_realizează\_MONTARE\_la\_TEATRU (relație de tip 3) desemnează asocierea dintre un artist, un proiect la care acesta lucrează și instituția teatrală, în cadrul căreia se realizează proiectul respectiv.

4) Relația TEHNICIAN\_angajat\_la\_TEATRU reprezintă asocierea dintre o instanță a subentității TEHNICIAN și instituția la care acesta lucrează. Având în vedere restricțiile impuse asupra modelului la punctele 11 și 12, rezultă următoarele cardinalități pentru această relație: **cardinalitate minima 1:1 și cardinalitate maximă m:1**.

5) Relația TEATRU\_are\_ADRESA desemnează asocierea dintre o instituție teatrală și adresa la care aceasta își are sediul. Având în vedere restricțiile impuse asupra modelului la punctele 1 și 2, rezultă următoarele cardinalități pentru această relație: **cardinalitate minima 1:1 și cardinalitate maximă 1:1**.

6) Relația TEATRU\_are\_SALA reprezintă asocierea dintre un teatru și o sală, aparținând acestuia. Ținând cont de regulile 3 și 4, deducem cardinalitățile corespunzătoare acestei relații: **cardinalitate minima 1:1 și cardinalitate maximă 1:m**.

7) Relația PIESA\_are\_MONTARE desemnează asocierea dintre piesele de teatru și punerile lor în scenă. Conform regulilor 6 și 7, putem stabili următoarele cardinalități: **cardinalitate minima 1:1 și cardinalitate maximă 1:m**.

8) Relația MONTARE\_se\_joaca\_in\_SALA reprezintă asocierea dintre o punere în scenă a unei piese și spațiile în care reprezentațiile acesteia se desfășoară. Potrivit regulii 5, deducem cardinalitățile acestei relații: **cardinalitate minima 0:1 și cardinalitate maximă n:m**.

9) Relația MONTARE\_se\_joaca\_la\_FESTIVAL\_in\_SALA (relație de tip 3) desemnează asocierea dintre un festival, o montare care participă la acesta și sala în care este găzduit spectacolul, în cadrul festivalului. Relația reflectă ce montări se desfășoară în ce săli, la ce festivaluri.

10) Relația SPONSOR\_finanțează\_MONTARE desemnează asocierea dintre sponsori și producțiile pe care aceștia le susțin. Având în vedere regulile 17 și 18, decidem cardinalitățile: și **cardinalitate minima 0:1 și cardinalitate maximă n:m**.

## **Descrierea atributelor asociate entităților:**

### **1) Atributele entității TEATRU:**

- cod\_teatru: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul teatrului
- denumire: variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea teatrului
- cod\_adresa: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul adresei; atributul este o cheie externă, care referențiază cheia primară cod\_adresă, a entității ADRESĂ
- data\_infiintare: variabilă de tip dată calendaristică, care reprezintă data înființării teatrului respectiv

### **2) Atributele entității PERSOANA:**

- cod\_persoana: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul persoanei
- nume: variabilă de tip caracter, de lungime maximă 25, care reprezintă numele de familie al persoanei; valoarea atributului trebuie să fie diferită de NULL
- prenume: variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele persoanei
- data\_nasterii: variabilă de tip dată calendaristică, care reprezintă data nașterii persoanei respective
- nr\_telefon: variabilă de tip caracter, de lungime maximă 25, care reprezintă numărul de telefon al persoanei
- email: variabilă de tip caracter, de lungime maximă 25, care reprezintă adresa de e-mail a persoanei; nu pot exista două persoane, în model, cu aceeași adresa de e-mail
- profesie: variabilă de tip caracter, de lungime maximă 10, care reprezintă categoria profesională din care face parte persoana; valori posibile: “artist”, “tehnician”

#### **2.1) Atributele subentității ARTIST:**

- cod\_persoana: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul artistului
- sex: variabilă de tip caracter, de lungime 1, luând valorile ‘m’ sau ‘f’, care reprezintă sexul artistului

#### **2.2) Atributele subentității TEHNICIAN:**

- cod\_persoana: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul artistului
- data\_angajarii: variabilă de tip dată calendaristică, care reprezintă data angajării tehnicianului respective; valoarea implicită a acestui atribut este data curentă
- cod\_teatru: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul teatrului la care lucrează tehnicianul; atributul este o cheie externă, care referențiază cheia primară cod\_adresă, a entității ADRESĂ

### 3) Atributele entității SPECIALIZARE:

- cod\_specializare: variabilă de tip întreg, de lungime maximă 2, care reprezintă codul specializării
- denumire: variabilă de tip caracter, de lungime maximă 40, care reprezintă denumirea specializării artistului (valori posibile: “actor”, “regizor”, “scenarist”, “scenograf”, “compozitor”, “muzician”, “coregraf”)
- observații: variabilă de tip caracter, de lungime maximă 2000, care include mențiuni suplimentare despre specializarea artistului (valori posibile: “în curs de pregătire”, “în curs de perfecționare” etc.)

### 4) Atributele entității PIESA:

- cod\_piesa: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul piesei
- titlu: variabilă de tip caracter, de lungime maximă 70, care reprezintă titlul piesei de teatru
- autor: variabilă de tip caracter, de lungime maximă 50, care reprezintă numele autorului
- an\_publicare: variabilă de tip dată calendaristică, care reprezintă data publicării piesei respective
- gen: variabilă de tip caracter, de lungime maximă 20, care reprezintă genul piesei (valori posibile: “comedie”, “dramă”, “spectacol pentru copii” etc)

### 5) Atributele entității MONTARE:

- cod\_montare: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul montării
- cod\_piesa: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul piesei care este montată; atributul este o cheie externă, care referențiază cheia primară cod\_piesă, a entității PIESA
- cod\_teatru: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul teatrului la care se realizează montarea; atributul este o cheie externă, care referențiază cheia primară cod\_piesă, a entității TEATRU
- stagiune: variabilă de tip caracter, de lungime maximă 10, care reprezintă stagiunea în cadrul căreia se realizează montarea (valori posibile: “2020-2021”, “2021-2022” etc.)
- buget: variabilă de tip numeric (real), de lungime maximă 6 (dintre care 2 cifre zecimale), care reprezintă bugetul alocat montării; valoarea acestui atribut trebuie să fie diferită de NULL, iar valoarea implicită a acestuia este 1000
- data\_premierei: variabilă de tip dată calendaristică, care reprezintă data premierei montării respective; valoarea implicită a atributului este data curentă



#### 6) Atributele entității SALA:

- cod\_sala: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul sălii
- cod\_teatru: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul teatrului de care aparține respectiva sală
- denumire: variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea sălii
- capacitate: variabilă de tip întreg, de lungime maximă 4, care reprezintă numărul de locuri disponibile în acea sală; valoarea acestui atribut trebuie să fie diferită de NULL

#### 7) Atributele entității ADRESĂ:

- cod\_adresa: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul adresei
- oras: variabilă de tip caracter, de lungime maximă 40, care reprezintă numele orașului în care se găsește adresa respectivă
- strada: variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea străzii pe care se află adresa respectivă
- numar: variabilă de tip întreg, de lungime maximă 4, care reprezintă numărul la care se găsește adresa respective

#### 8) Atributele entității SPONSOR:

- cod\_sponsor: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul sponsorului
- nume: variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea sponsorului
- nr\_telefon: variabilă de tip caracter, de lungime maximă 25, care reprezintă numărul de telefon al sponsorului
- email: variabilă de tip caracter, de lungime maximă 25, care reprezintă adresa de e-mail a sponsorului; nu pot exista doi sponsori, în model, cu aceeași adresa de e-mail

#### 9) Atributele entității FESTIVAL:

- cod\_festival: variabilă de tip întreg, de lungime maximă 4, care reprezintă codul festivalului
- denumire: variabilă de tip caracter, de lungime maximă 50, care reprezintă denumirea festivalului
- locație: variabilă de tip caracter, de lungime maximă 50, care reprezintă locația în care se desfășoară festivalul
- data\_inceperii: variabilă de tip dată calendaristică, care reprezintă data la care începe festivalul respectiv
- data\_incheierii: variabilă de tip dată calendaristică, care reprezintă data la care se încheie festivalul respectiv

**Schemele relaționale corespunzătoare diagramei conceptuale realizate sunt următoarele:**

- TEATRU(cod\_teatru#, denumire, cod\_adresa, data\_infiintare);
- PERSOANA(cod\_persoana#, nume, prenume, data\_n, nr\_tel, e\_mail, profesie);
- ARTIST(cod\_persoana#, sex);
- TEHNICIAN(cod\_persoana#, data\_ang, cod\_teatru);
- SPECIALIZARE(cod\_specializare#, denumire, observatii);
- PIESA(cod\_piesa#, titlu, autor, an\_publicare, gen);
- MONTARE(cod\_montare#, cod\_piesa, cod\_teatru, stagiune, buget, data\_premierei);
- SALA(cod\_sala#, cod\_teatru#, nume, capacitate);
- ADRESA(cod\_adresa#, oras, strada, nr);
- SPONSOR(cod\_sponsor#, nume, nr\_tel, e\_mail);
- FESTIVAL(cod\_festival#, denumire, locatie, data\_inceperii, data\_incheierii);
- DISTRIBUTIE(cod\_montare#, cod\_artist#, cod\_teatru#, functie, rol);
- SPEC\_ART(cod\_artist#, cod\_specializare#);
- AFIS (cod\_montare#, cod\_sala#, cod\_teatru#, data#, ora);
- CONTRACT (cod\_sponsor#, cod\_montare#, suma);
- AGENDA (cod\_festival#, cod\_montare#, cod\_sala#, data, ora);

4.

create sequence secv

start with 1

increment by 1

maxvalue 100

nocycle

nocache;

```
create sequence secv
start with 1
increment by 1
maxvalue 100
nocycle
nocache;
```

create table teatru(

cod\_teatru number(4) constraint pk\_teatru primary key,

denumire varchar2(50),

cod\_adresa number(4),

data\_infiintare date

);

alter table teatru

add constraint fk\_teatru\_adr foreign key(cod\_adresa) references adresa(cod\_adresa) on delete cascade;

create table persoana(

cod\_persoana number(4) constraint pk\_pers primary key,

nume varchar2(25) constraint not\_null\_nume not null,

prenume varchar2(25),

data\_nasterii date,

nr\_telefon number(10),

email varchar2(25) constraint uniq\_email unique

profesie varchar2(10)

);

```

create table artist(
    cod_persoana number(4) constraint pk_art primary key,
    sex char(1),
    constraint fk_art_pers foreign key(cod_persoana) references persoana(cod_persoana) on
delete cascade
);

create table tehnician(
    cod_persoana number(4) constraint pk_teh primary key,
    data_angajarii date default sysdate,
    cod_teatru number(4),
    constraint fk_tehn_pers foreign key(cod_persoana) references persoana(cod_persoana) on
delete cascade,
    constraint fk_tehn_teatru foreign key(cod_teatru) references teatru(cod_teatru) on delete set
null
);

create table specializare(
    cod_specializare number(2) constraint pk_spec primary key,
    denumire varchar2(40),
    observatii varchar2(2000)
);

create table piesa(
    cod_piesa number(4) constraint pk_piesa primary key,
    titlu varchar2(70),
    autor varchar2(50),
    an_publicare number(4),
    gen varchar2(25));

```

```

create table montare(
    cod_montare number(4) constraint pk_montare primary key,
    cod_piesa number(4),
    cod_teatru number(4),
    stagiune varchar2(10),
    data_premierei date,
    buget number(6, 2) constraint not_null_buget not null,
    constraint fk_mont_piesa foreign key (cod_piesa) references piesa(cod_piesa) on delete set
null,
    constraint fk_mont_teatru foreign key (cod_teatru) references teatru(cod_teatru) on delete set
null
);

```

```

alter table montare
modify buget default 1000;

```

```

create table sala(
    cod_sala number(4),
    cod_teatru number(4),
    denumire varchar2(50),
    capacitate number(4),
    constraint pk_sala_c primary key(cod_sala, cod_teatru),
    constraint fk_sala_teatru foreign key (cod_teatru) references teatru(cod_teatru) on delete
cascade
);

```

```

create table adresa(
    cod_adresa number(4) constraint pk_adr primary key,
    oras varchar2(40),
    strada varchar2(50),
    numar number(4));

create table sponsor(
    cod_sponsor number(4) constraint pk_sponsor primary key,
    nume varchar2(50),
    nr_telefon varchar2(25),
    email varchar2(25) constraint unq_email_sp unique
);

```

```

create table festival(
    cod_festival number(4) constraint pk_fest primary key,
    denumire varchar2(50),
    locatie varchar2(50),
    data_inceperii date,
    data_incheierii date,
    constraint ck_di_df check(data_incheierii > data_inceperii)
);

```

```

create table distributie(
    cod_montare number(4),
    cod_artist number(4),
    cod_teatru number(4),
    rol varchar2(50),
    constraint pk_dist primary key(cod_montare, cod_artist, cod_teatru),

```

```
constraint fk_dist_art foreign key(cod_artist) references artist(cod_persoana) on delete set null,
```

```
constraint fk_dist_mont foreign key(cod_montare) references montare(cod_montare) on delete cascade,
```

```
constraint fk_dist_teatru foreign key(cod_teatru) references teatru(cod_teatru) on delete cascade
```

```
);
```

```
alter table distributie
```

```
add functie varchar2(30);
```

```
create table spec_art(
```

```
cod_artist number(4),
```

```
cod_specializare number(2),
```

```
constraint pk_spec_art primary key(cod_artist, cod_specializare),
```

```
constraint fk_specart_art foreign key(cod_artist) references artist(cod_persoana) on delete cascade,
```

```
constraint fk_specart_spec foreign key(cod_specializare) references  
specializare(cod_specializare) on delete cascade
```

```
);
```

```
create table afis(
```

```
cod_montare number(4),
```

```
cod_sala number(4),
```

```
cod_teatru number(4),
```

```
data date,
```

```
ora date,
```

```
constraint pk_afis primary key(cod_montare, cod_sala, cod_teatru, data),
```

```
constraint fk_afis_mont foreign key(cod_montare) references montare(cod_montare) on delete cascade,
```

```
constraint fk_afis_sala foreign key(cod_sala, cod_teatru) references sala(cod_sala,  
cod_teatru) on delete cascade);
```

```
create table contract(  
    cod_sponsor number(4),  
    cod_montare number(4),  
    suma number(6, 2),  
    constraint pk_cont primary key(cod_sponsor, cod_montare),  
    constraint fk_cont_spon foreign key(cod_sponsor) references sponsor(cod_sponsor) on delete  
set null,  
    constraint fk_cont_mont foreign key(cod_montare) references montare(cod_montare) on  
delete cascade  
);
```

```
create table agenda(  
    cod_festival number(4),  
    cod_montare number(4),  
    cod_teatru number(4),  
    cod_sala number(4),  
    data date,  
    ora date,  
    constraint pk_agenda primary key(cod_festival, cod_montare, cod_sala, cod_teatru),  
    constraint fk_ag_fest foreign key(cod_festival) references festival(cod_festival) on delete  
cascade,  
    constraint fk_ag_montare foreign key(cod_montare) references montare(cod_montare) on  
delete cascade,  
    constraint fk_ag_sala foreign key(cod_sala, cod_teatru) references sala(cod_sala, cod_teatru)  
on cascade  
);
```



5.

insert into adresa values (secv.nextval, 'Bucuresti', 'Nicolae Balcescu', 2);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Schitu Magureanu', 1);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Magheru', 20);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Constantin Mile', 16);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Carol', 21);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Sfantul Dumitru', 2);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Doctor Iuliu Barasch', 15);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Calea Victoriei', 40);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Mihai Eminescu', 89);  
insert into adresa values (secv.nextval, 'Bucuresti', 'Uverturii', 70);  
insert into adresa values (secv.nextval, 'Braila', 'Mihai Eminescu', 2);  
insert into adresa values (secv.nextval, 'Buzau', 'Nicolae Balcescu', 50);  
insert into adresa values (secv.nextval, 'Cluj', 'Stefan cel Mare', 24);  
insert into adresa values (secv.nextval, 'Constanta', 'Mircea cel Batran', 97);  
insert into adresa values (secv.nextval, 'Craiova', 'Alexandru Ioan Cuza', 11);  
insert into adresa values (secv.nextval, 'Iasi', 'Agatha Barescu', 18);  
insert into adresa values (secv.nextval, 'Ploiesti', 'Toma Caragiu', 13);  
insert into adresa values (secv.nextval, 'Sibiu', 'Corneliu Coposu', 2);  
insert into adresa values (secv.nextval, 'Timisoara', 'Marasesti', 2);

select \* from adresa;

Query Result x

All Rows Fetched: 19 in 0.045 seconds

	COD_ADRESA	ORAS	STRADA	NUMAR
1	1	Bucuresti	Nicolae Balcescu	2
2	2	Bucuresti	Schitu Maquireanu	1
3	3	Bucuresti	Maqheru	20
4	4	Bucuresti	Constantin Mile	16
5	5	Bucuresti	Carol	21
6	6	Bucuresti	Sfantul Dumitru	2
7	7	Bucuresti	Doctor Iuliu Barasch	15
8	8	Bucuresti	Calea Victoriei	40
9	9	Bucuresti	Mihai Eminescu	89
10	10	Bucuresti	Uverturii	70
11	11	Braila	Mihai Eminescu	2
12	12	Buzau	Nicolae Balcescu	50
13	13	Cluj	Stefan cel Mare	24
14	14	Constanta	Mircea cel Batran	97
15	15	Craiova	Alexandru Ioan Cuza	11
16	16	Iasi	Aqatha Barescu	18
17	17	Ploiesti	Toma Caragiu	13
18	18	Sibiu	Corneliu Coposu	2
19	19	Timisoara	Marasesti	2

insert into teatru values (secv.nextval, 'Teatrul National Bucuresti', secv.currval, '31-DEC-1852');

insert into teatru values (secv.nextval, 'Teatrul Bulandra', secv.currval, '17-SEP-1947');

insert into teatru values (secv.nextval, 'Teatrul Notarra', secv.currval, '1-JAN-1947');

insert into teatru values (secv.nextval, 'Teatrul Mic', secv.currval, '22-DEC-1964');

insert into teatru values (secv.nextval, 'Teatrul Foarte Mic', secv.currval, '1-JAN-1979');

insert into teatru values (secv.nextval, 'Teatrul de Comedie', secv.currval, '5-JAN-1961');

insert into teatru values (secv.nextval, 'Teatrul Evreiesc de Stat', secv.currval, '19-AUG-1876');

insert into teatru values (secv.nextval, 'Teatrul Odeon', secv.currval, '25-DEC-1911');

insert into teatru values (secv.nextval, 'Teatrul Metropolis', secv.currval, '1-JAN-2007');

insert into teatru values (secv.nextval, 'Teatrul Masca', secv.currval, '24-MAY-1990');

insert into teatru values (secv.nextval, 'Teatrul Maria Filotti', secv.currval, '1-JAN-1864');

insert into teatru values (secv.nextval, 'Teatrul George Ciprian', secv.currval, '5-APR-1996');

insert into teatru values (secv.nextval, 'Teatrul National Lucian Blaga', secv.currval, '14-MAY-1919');

insert into teatru values (secv.nextval, 'Teatrul National Constanta', secv.currval, '1-JAN-1951');

insert into teatru values (secv.nextval, 'Teatrul National Craiova', secv.currval, '29-JUN-1850');

insert into teatru values (secv.nextval, 'Teatrul National Vasile Alecsandri', secv.currval, '1-JAN-1840');

insert into teatru values (secv.nextval, 'Teatrul Toma Caragiu', secv.currval, '15-APR-1957');

insert into teatru values (secv.nextval, 'Teatrul Radu Stanca', secv.currval, '1-JUN-1788');

insert into teatru values (secv.nextval, 'Teatrul National Mihai Eminescu', secv.currval, '22-SEP-1875');

select \* from teatru;

Query Result x

SQL | All Rows Fetched: 19 in 0.021 seconds

	COD_TEATRU	DENUMIRE	COD_ADRESA	DATA_INFINTARE
1	1	Teatrul National Bucuresti	1	31-DEC-52
2	2	Teatrul Bulandra	2	17-SEP-47
3	3	Teatrul Notarra	3	01-JAN-47
4	4	Teatrul Mic	4	22-DEC-64
5	5	Teatrul Foarte Mic	5	01-JAN-79
6	6	Teatrul de Comedie	6	05-JAN-61
7	7	Teatrul Evreiesc de Stat	7	19-AUG-76
8	8	Teatrul Odeon	8	25-DEC-11
9	9	Teatrul Metropolis	9	01-JAN-07
10	10	Teatrul Masca	10	24-MAY-90
11	11	Teatrul Maria Filotti	11	01-JAN-64
12	12	Teatrul George Ciprian	12	05-APR-96
13	13	Teatrul National Lucian Blaga	13	14-MAY-19
14	14	Teatrul National Constanta	14	01-JAN-51
15	15	Teatrul National Craiova	15	29-JUN-50
16	16	Teatrul National Vasile Alecsandri	16	01-JAN-40
17	17	Teatrul Toma Caragiu	17	15-APR-57
18	18	Teatrul Radu Stanca	18	01-JUN-88
19	19	Teatrul National Mihai Eminescu	19	22-SEP-75

insert into specializare (cod\_specializare, denumire) values (secv.nextval, 'actor');

insert into specializare (cod\_specializare, denumire) values (secv.nextval, 'regizor');

insert into specializare (cod\_specializare, denumire) values (secv.nextval, 'scenograf');

insert into specializare (cod\_specializare, denumire) values (secv.nextval, 'muzica');

insert into specializare (cod\_specializare, denumire) values (secv.nextval, 'costume');

select \* from specializare;

Query Result x

SQL | All Rows Fetched: 5 in 0.016 seconds

	COD_SPECIALIZARE	DENUMIRE	OBSERVATII
1	1	actor	(null)
2	2	regizor	(null)
3	3	scenograf	(null)
4	4	muzica	(null)
5	5	costume	(null)

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'O noapte furtunoasa', 'Ion Luca Caragiale', 'comedie', 1879);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Trei surori', 'Anton Pavlovici Cehov', 'drama', 1900);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Revizorul', 'Nicolai Vasilivici Gogol', null, 1836);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'O scrisoare pierduta', 'Ion Luca Caragiale', 'comedie', 1884);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'D-ale carnavalului', 'Ion Luca Caragiale', 'comedie', 1885);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Livada de visini', 'Anton Pavlovici Cehov', null, 1904);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Omul cu martoaga', 'George Ciprian', null, 1927);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Furtuna', 'William Shakespeare', 'comedie', 1610);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Cum va place', 'William Shakespeare', 'comedie', 1600);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Totul e bine cand se termina cu bine', 'William Shakespeare', 'comedie', 1623);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Richard al III-lea', 'William Shakespeare', 'istorica', 1591);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Romeo si Julieta', 'William Shakespeare', 'tragedie', 1597);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Hamlet', 'William Shakespeare', 'tragedie', 1600);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Regele Lear', 'William Shakespeare', 'tragedie', 1606);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Visul unei nopti de vara', 'William Shakespeare', 'comedie', 1595);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Faust', 'Johann Wolfgang von Goethe', 'drama', 1790);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Micul Print', 'Antoine de Saint Exupery', null, 1943);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Gaitele', 'Alexandru Kiritescu', 'comedie', 1829);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Lectia', 'Eugen Ionesco', 'drama', 1950);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Angajare de clovn', 'Matei Visniec', 'drama', 1986);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Scandal in culise', 'Michael Frayn', 'comedie', 1982);

insert into piesa (cod\_piesa, titlu, autor, gen, an\_publicare) values (secv.nextval, 'Pescarusul', 'Anton Pavlovici Cehov', 'drama', 1895);

insert into piesa (cod\_piesa, titlu, autor, an\_publicare) values (secv.nextval, 'Pescarusul', 'Boris Akunin', 2012);

select * from piesa;				
Query Result x				
All Rows Fetched: 23 in 0.011 seconds				
	COD_PIESA	TITLU	AUTOR	AN_PUBLICARE
1	1	0 noapte furtunoasa	Ion Luca Caragiale	1879
2	2	Trei surori	Anton Pavlovici Cehov	1900
3	3	Revizorul	Nicolai Vasilievici Gogol	(null)
4	4	0 scrisoare pierduta	Ion Luca Caragiale	1884
5	5	D-ale carnavalului	Ion Luca Caragiale	1885
6	6	Livada de visini	Anton Pavlovici Cehov	(null)
7	7	Omul cu martoaga	George Ciprian	(null)
8	8	Furtuna	William Shakespeare	1610
9	9	Cum va place	William Shakespeare	1600
10	10	Totul e bine cand se termina cu bine	William Shakespeare	1623
11	11	Richard al III-lea	William Shakespeare	1591
12	12	Romeo si Julieta	William Shakespeare	1597
13	13	Hamlet	William Shakespeare	1600
14	14	Regele Lear	William Shakespeare	1606
15	15	Visul unei nopti de vara	William Shakespeare	1595
16	16	Faust	Johann Wolfgang von Goethe	1790
17	17	Micul Print	Antoine de Saint Exupery	(null)
18	18	Gaitetele	Alexandru Kiritescu	1829
19	19	Lectia	Eugen Ionesco	1950
20	20	Angajare de clovn	Matei Visniec	1986
21	21	Scandal in culise	Michael Frayn	1982
22	22	Pescarusul	Anton Pavlovici Cehov	1895
23	23	Pescarusul	Boris Akunin	(null)

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Alexa', 'Felix', 'af@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Rusu', 'Mircea', 'rn@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Olaru', 'Orodel', 'oo@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Puric', 'Dan', 'pd@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Calin', 'Mihai', 'cm@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Movila', 'Irina', 'mi@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Marinescu', 'Medeea', 'mm@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Ton', 'Diana Ruxandra', 'idr@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Calota', 'Armand', to\_date('22-07-1962', 'dd-mm-yyyy'), 'ca@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Bârbora', 'Cecilia', to\_date('10-05-1963', 'dd-mm-yyyy'), 'bc@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Maci', 'Eugenia', to\_date('03-02-1951', 'dd-mm-yyyy'), 'me@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Goia', 'Ilinca', to\_date('06-03-1969', 'dd-mm-yyyy'), 'gi@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ivanov', 'Vlad', to\_date('04-08-1969', 'dd-mm-yyyy'), 'iv@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ionescu', 'Rodica', to\_date('01-05-1977', 'dd-mm-yyyy'), 'ir@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Krasovski', 'Yuri', 'ki@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Popescu Farcasanu', 'Florinela', 'pff@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Mihalev', 'Kiril', 'mk@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Smirnov', 'Serghei', 'ss@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Bogomaz', 'Dimitrii', 'bd@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Boguta', 'Dorian', 'bd1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Crasmaru', 'Damian', to\_date('10-05-1931', 'dd-mm-yyyy'), 'cd@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ionescu', 'Dragos', to\_date('28-09-1967', 'dd-mm-yyyy'), 'id@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Biris', 'Silviu', to\_date('06-10-1973', 'dd-mm-yyyy'), 'bs@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Alexandru', 'Matei', to\_date('25-12-1927', 'dd-mm-yyyy'), 'am@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Zamfirescu', 'Raluca', to\_date('01-11-1924', 'dd-mm-yyyy'), 'zr@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Ciochiu', 'Simona', 'cs@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Cerkasski', 'Serghei', 'cs1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Mokriv', 'Mihail', 'mm1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Rogalev', 'Igor', 'ri@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Cristea', 'Eugen', to\_date('18-04-1952', 'dd-mm-yyyy'), 'ce@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Cercel', 'Florina', to\_date('28-01-1943', 'dd-mm-yyyy'), 'cf@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Georgescu', 'Alexandru', to\_date('24-01-1948', 'dd-mm-yyyy'), 'ga@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Gheorghe', 'Radu', 'gr@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Hasnas', 'Alexandru', to\_date('13-03-1933', 'dd-mm-yyyy'), 'ha@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Morgenstern', 'Maia', to\_date('01-05-1962', 'dd-mm-yyyy'), 'mm2@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Rizea', 'Marius', to\_date('24-11-1975', 'dd-mm-yyyy'), 'rm@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ionescu', 'Serban', to\_date('23-09-1950', 'dd-mm-yyyy'), 'is@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Musatescu', 'Bogdan', to\_date('01-06-1941', 'dd-mm-yyyy'), 'mb@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Cobzariu', 'Marcelo', to\_date('28-03-1970', 'dd-mm-yyyy'), 'cm1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ionescu', 'Ioan Andrei', to\_date('12-04-1974', 'dd-mm-yyyy'), 'iaa@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Andreescu', 'Miruna', to\_date('14-02-1977', 'dd-mm-yyyy'), 'am@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Andreescu', 'Miruna', to\_date('14-02-1977', 'dd-mm-yyyy'), 'am1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Mirea', 'Eugen', to\_date('24-03-1977', 'dd-mm-yyyy'), 'me1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ciochina', 'Anda', to\_date('09-10-1977', 'dd-mm-yyyy'), 'ca2@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Raileanu', 'Costin', to\_date('30-09-1977', 'dd-mm-yyyy'), 'rc1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Vermesan', 'Linda', to\_date('14-02-1977', 'dd-mm-yyyy'), 'vl1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ghetu', 'Theodor', to\_date('11-05-1980', 'dd-mm-yyyy'), 'gt1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ene', 'Iasmina', to\_date('19-06-1980', 'dd-mm-yyyy'), 'ei1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, email, profesie) values (secv.nextval, 'Ichim', 'Vladimir', to\_date('23-08-1980', 'dd-mm-yyyy'), 'iv1@', 'artist');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Bisa', 'Emanuel', 'be@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Spalatu', 'Ionel', 'si@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Ene', 'Valentin', 'ev@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Cudalbeanu', 'Stefan', 'cs2@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Staicu', 'Gheorghe', 'sg@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Chirita', 'Adina', 'ca1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Solomonescu', 'George', 'sg1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Deaconu', 'Vasile', 'dv@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Custura', 'Radu', 'cr@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Frincu', 'Angelina', 'fa@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Stamate', 'Elena', 'se@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Trifan', 'Victor', 'tv@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Ternat', 'Laurentiu', 'il@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Bajenaru', 'Stefan', 'bs1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Verzea', 'Dumitra', 'vd@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Chesca', 'Diana', 'cd1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Surdu', 'Ion', 'si1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Dragu', 'Petre', 'dp@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Hangulescu', 'Serban', 'hs@', 'tehnician');



insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Ghimpatu', 'Sorin', 'gs@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Bercaru', 'Tudor', 'bt@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Dimov', 'Simona', 'ds@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Petrescu', 'Paul', 'pp@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Crivoi', 'Stefan', 'cs3@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Jianu', 'Grigore', 'jg@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Boieru', 'Traian', 'bt1@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Enescu', 'Mihail', 'em@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Radulescu', 'Pavel', 'rp@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Chereches', 'Istvan', 'ci@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, email, profesie) values (secv.nextval, 'Prundea', 'Alexandru', 'pa@', 'tehnician');

insert into persoana (cod\_persoana, nume, prenume, data\_nasterii, profesie) values (78, 'Rusu', 'Mircea', to\_date('19-04-1984', 'dd-mm-yyyy'), 'artist');

select \* from persoana;

Query Result x

SQL | Fetched 50 rows in 0.026 seconds

	COD_PERSOANA	NUME	PRENUME	DATA_NASTERII	NR_TELEFON	EMAIL	PROFESIE
1	1	Alexa	Felix	(null)	(null)	af@	artist
2	2	Rusu	Mircea	(null)	(null)	rn@	artist
3	3	Olaru	Orodel	(null)	(null)	oo@	artist
4	4	Puric	Dan	(null)	(null)	pd@	artist
5	5	Calin	Mihai	(null)	(null)	cm@	artist
6	6	Movila	Irina	(null)	(null)	mi@	artist
7	7	Marinescu	Medeea	(null)	(null)	mm@	artist
8	8	Ion	Diana Ruxandra	(null)	(null)	idr@	artist
9	9	Calota	Armand	22-JUL-62	(null)	ca@	artist
10	10	Bărbora	Cecilia	10-MAY-63	(null)	bc@	artist
11	11	Maci	Euqenia	03-FEB-51	(null)	me@	artist
12	12	Goia	Ilinca	06-MAR-69	(null)	qi@	artist
13	13	Ivanov	Vlad	04-AUG-69	(null)	iv@	artist
14	14	Ionescu	Rodica	01-MAY-77	(null)	ir@	artist
15	15	Krasovski	Yuri	(null)	(null)	ki@	artist
16	16	Popescu Farcasanu	Florinela	(null)	(null)	pff@	artist
17	17	Mihalev	Kiril	(null)	(null)	mk@	artist
18	18	Smirnov	Serghei	(null)	(null)	ss@	artist
19	19	Boqomaz	Dimitrii	(null)	(null)	bd@	artist
20	20	Boquta	Dorian	(null)	(null)	bd1@	artist
21	21	Crasmaru	Damian	10-MAY-31	(null)	cd@	artist
22	22	Ionescu	Dragos	28-SEP-67	(null)	id@	artist
23	23	Biris	Silviu	06-OCT-73	(null)	bs@	artist
24	24	Alexandru	Matei	25-DEC-27	(null)	am@	artist
25	25	Zamfirescu	Raluca	01-NOV-24	(null)	zr@	artist
26	26	Ciochiu	Simona	(null)	(null)	cs@	artist
27	27	Cerkasski	Serghei	(null)	(null)	cs1@	artist
28	28	Mokriv	Mihail	(null)	(null)	mm1@	artist
29	29	Rogalev	Igor	(null)	(null)	ri@	artist
30	30	Cristea	Euqen	18-APR-52	(null)	ce@	artist
31	31	Cercel	Florina	28-JAN-43	(null)	cf@	artist

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'F');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

insert into artist (cod\_persoana, sex) values (secv.nextval, 'M');

[illegible]

insert into tehnician (cod\_persoana, data\_angajarii, cod\_teatru)  
values (secv.nextval, '03-AUG-1976', 1);

insert into tehnician (cod\_persoana, cod\_teatru) values  
(secv.nextval, 1);

insert into tehnician (cod\_persoana, cod\_teatru) values  
(secv.nextval, 1);

insert into tehnician (cod\_persoana, data\_angajarii, cod\_teatru)  
values (secv.nextval, '25-FEB-1980', 1);

insert into tehnician (cod\_persoana, cod\_teatru) values  
(secv.nextval, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values  
(secv.nextval, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (47, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (48, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (49, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (50, 3);

insert into tehnician (cod\_persoana, cod\_teatru) values (51, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (52, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (53, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (54, 5);

insert into tehnician (cod\_persoana, cod\_teatru) values (55, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (56, 2);

insert into tehnician (cod\_persoana, cod\_teatru) values (57, 2);

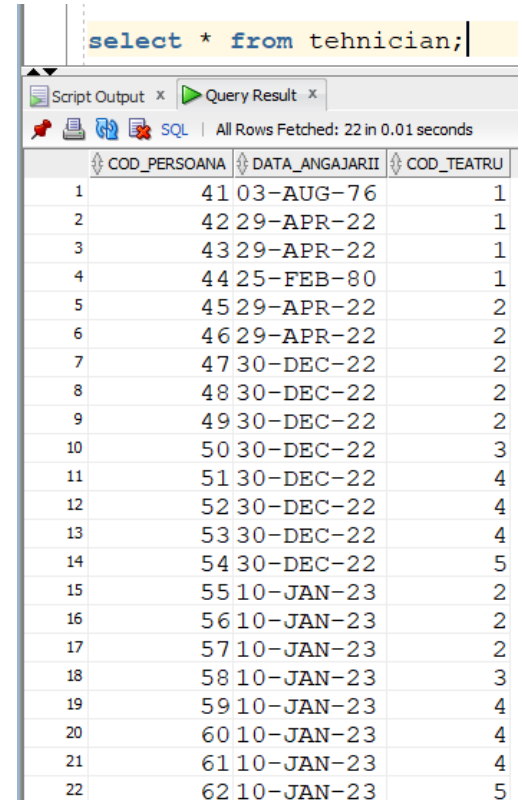
insert into tehnician (cod\_persoana, cod\_teatru) values (58, 3);

insert into tehnician (cod\_persoana, cod\_teatru) values (59, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (60, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (61, 4);

insert into tehnician (cod\_persoana, cod\_teatru) values (62, 5);



	COD_PERSOANA	DATA_ANGAJARII	COD_TEATRU
1	41	03-AUG-76	1
2	42	29-APR-22	1
3	43	29-APR-22	1
4	44	25-FEB-80	1
5	45	29-APR-22	2
6	46	29-APR-22	2
7	47	30-DEC-22	2
8	48	30-DEC-22	2
9	49	30-DEC-22	2
10	50	30-DEC-22	3
11	51	30-DEC-22	4
12	52	30-DEC-22	4
13	53	30-DEC-22	4
14	54	30-DEC-22	5
15	55	10-JAN-23	2
16	56	10-JAN-23	2
17	57	10-JAN-23	2
18	58	10-JAN-23	3
19	59	10-JAN-23	4
20	60	10-JAN-23	4
21	61	10-JAN-23	4
22	62	10-JAN-23	5

[illegible]

```

insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 3);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 2);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 4);
insert into spec_art (cod_artist, cod_specializare) values (secv.nextval, 1);
insert into spec_art (cod_artist, cod_specializare) values (8, 5);

```

select \* from spec\_art;

Query Result: x All Rows Fetched: 47 in 0.014 seconds

	COD_ARTIST	COD_SPECIALIZARE
1	1	2
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	3
9	9	1
10	10	1
11	11	1
12	12	1
13	13	1
14	14	1
15	15	2
16	16	3
17	17	4
18	18	5
19	19	1
20	20	1
21	21	1
22	22	1
23	23	1
24	24	1
25	25	1
26	26	1
27	27	2
28	28	3
29	29	4
30	30	1
31	31	1
32	32	1
33	33	1
34	34	1
..	..	..

```

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei, buget)
values (secv.nextval,1,1,'2012-2013',to_date('09-10-2012', 'dd-mm-yyyy'), 3000);

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei) values
(secv.nextval,2,1,'2019-2020',to_date('05-12-2019', 'dd-mm-yyyy'));

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei) values
(secv.nextval,3,1,'2013-2014',to_date('06-10-2013', 'dd-mm-yyyy'));

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei, buget)
values (secv.nextval,5,1,'2021-2022',to_date('26-10-2021', 'dd-mm-yyyy'), 3000);

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei) values
(secv.nextval,4,1,'2021-2022',to_date('19-01-2022', 'dd-mm-yyyy'));

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei, buget)
values (secv.nextval,6,1,'1991-1992', to_date('23-04-1992', 'dd-mm-yyyy'), 2000);

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei, buget)
values (secv.nextval,6, 1, '2009-2010', to_date('13-05-2010', 'dd-mm-yyyy'), 2500);

insert into montare (cod_montare, cod_piesa, cod_teatru, stagiune, data_premierei) values
(secv.nextval,7, 1, '2011-2012', to_date('19-11-2011', 'dd-mm-yyyy'));

```

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval,15, 1, '2003-2004', to\_date('09-10-2003', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval,18, 1, '1977-1978', to\_date('29-11-1977', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval,19, 1, '2008-2009', to\_date('16-05-2009', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval,14, 2, '2018-2019', to\_date('08-11-2018', 'dd-mm-yyyy'), 3000);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 4, 2, '2007-2008', to\_date('30-01-2008', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, buget) values (secv.nextval, '1', '3', 1500);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval,13, 4, '2006-2007', to\_date('06-10-2006', 'dd-mm-yyyy'), 1750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval,15, 4, '2010-2011', to\_date('11-06-2011', 'dd-mm-yyyy'), 1750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 8, 4, '2018-2019', to\_date('15-01-2019', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval, 13, 5, '2011-2012', to\_date('16-11-2011', 'dd-mm-yyyy'), 2750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 4, 6, '2010-2011', to\_date('15-05-2011', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 13, 13, '2001-2002', to\_date('02-10-2001', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval, 18, 13, '2001-2002', to\_date('28-02-2002', 'dd-mm-yyyy'), 1750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 1, 13, '2005-2006', to\_date('15-10-2005', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval, 11, 13, '2005-2006', to\_date('10-12-2005', 'dd-mm-yyyy'), 1500);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 14, 13, '2006-2007', to\_date('24-10-2006', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval, 18, 13, '2007-2008', to\_date('22-12-2007', 'dd-mm-yyyy'), 2250);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 6, 13, '2009-2010', to\_date('29-10-2009', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 15, 13, '2019-2020', to\_date('06-03-2020', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 3, 13, '2009-2010', to\_date('04-05-2010', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 6, 14);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 9, 14);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei, buget) values (secv.nextval, 1, 15, '2011-2012', to\_date('15-11-2011', 'dd-mm-yyyy'), 1750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, data\_premierei) values (secv.nextval, 13, 15, '2019-2020', to\_date('02-05-2020', 'dd-mm-yyyy'));

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 1, 16, '2010-2011', 1750);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 5, 17);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 8, 17);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 16, 18);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 5, 18);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 15, 18);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 4, 19, '2010-2011', 2000);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune) values (secv.nextval, 3, 19, '2009-2010');

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune) values (secv.nextval, 12, 19, '2009-2010');

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 2, 19, '2007-2008', 1700);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune) values (secv.nextval, 20, 19, '2005-2006');

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 20, 19, '2003-2004', 2300);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune) values (secv.nextval, 15, 19, '2000-2001');

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune) values (secv.nextval, 17, 18, '2021-2022');

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 18, 2);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 20, 11);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru) values (secv.nextval, 9, 14);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, buget) values (secv.nextval, 18, 11, 1750);



insert into montare (cod\_montare, cod\_piesa, cod\_teatru, buget) values (secv.nextval, 7, 11, 1500);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 22, 11, '2022-2023', 2250);

insert into montare (cod\_montare, cod\_piesa, cod\_teatru, stagiune, buget) values (secv.nextval, 23, 13, '2022-2023', 2250);

select \* from montare;

Script Output x Query Result x

SQL All Rows Fetched: 53 in 0.037 seconds

	COD_MONTARE	COD_PIESA	COD_TEATRU	STAGIUNE	DATA_PREMIEREI	BUGET
1	1	1	1	2012-2013	09-OCT-12	3000
2	2	2	1	2019-2020	05-DEC-19	1000
3	3	3	1	2013-2014	06-OCT-13	1000
4	4	5	1	2021-2022	26-OCT-21	3000
5	5	4	1	2021-2022	19-JAN-22	1000
6	6	6	1	1991-1992	23-APR-92	2000
7	7	6	1	2009-2010	13-MAY-10	2500
8	8	7	1	2011-2012	19-NOV-11	1000
9	9	15	1	2003-2004	09-OCT-03	1000
10	10	18	1	1977-1978	29-NOV-77	1000
11	11	19	1	2008-2009	16-MAY-09	1000
12	12	14	2	2018-2019	08-NOV-18	3000
13	13	4	2	2007-2008	30-JAN-08	1000
14	14	1	3	(null)	(null)	1500
15	15	13	4	2006-2007	06-OCT-06	1750
16	16	15	4	2010-2011	11-JUN-11	1750
17	17	8	4	2019-2020	06-MAR-20	1000
18	18	13	5	2011-2012	16-NOV-11	2750
19	19	4	6	2010-2011	15-MAY-11	1000
20	20	13	13	2001-2002	02-OCT-01	1000
21	21	18	13	2001-2002	28-FEB-02	1750
22	22	1	13	2005-2006	15-OCT-05	1000
23	23	11	13	2005-2006	10-DEC-05	1500
24	24	14	13	2006-2007	24-OCT-06	1000
25	25	18	13	2007-2008	22-DEC-07	2250
26	26	6	13	2009-2010	29-OCT-09	1000
27	27	15	13	2009-2010	06-MAR-10	1000
28	28	3	13	2009-2010	04-MAY-10	1000
29	29	6	14	(null)	(null)	1000
30	30	9	14	(null)	(null)	1000
31	31	1	15	2011-2012	15-NOV-11	1750

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (1, 1, 1, 'regizor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (1, 1, 8, 'scenograf');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 2, 'actor', 'jupan Dumitrache');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 3, 'actor', 'Nae Ipingescu');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 4, 'actor', 'Rica Venturiano');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 5, 'actor', 'Chiriac');

insert into distributie values (1, 9, 1, 'Spiridon', 'actor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 6, 'actor', 'Veta');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (1, 1, 7, 'actor', 'Zita');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (2, 1, 15, 'regizor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (2, 1, 16, 'scenograf');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (2, 1, 18, 'costume');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (2, 1, 17, 'muzica');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 9, 'actor', 'Andrei Sergheevici Prozorov');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 10, 'actor', 'Natalia Ivanovna');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 11, 'actor', 'Olga');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 12, 'actor', 'Masa');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 13, 'actor', 'Kulighin Feodor Ilici');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 19, 'actor', 'Tuzenbach Nicolai Lvovici');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 20, 'actor', 'Solionii Vasilii Vasilievici');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie, rol) values (2, 1, 20, 'actor', 'Solionii Vasilii Vasilievici');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (9, 1, 14, 'actor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (15, 4, 75, 'actor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (3, 1, 27, 'regizor');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (3, 1, 28, 'scenograf');

insert into distributie (cod\_montare, cod\_teatru, cod\_artist, functie) values (3, 1, 18, 'costume');

insert into distributie values (5, 70, 1, 'Zoe', 'actor');

insert into distributie values (5, 9, 1, 'Stefan Tipatescu', 'actor');

insert into distributie (cod\_montare, cod\_artist, cod\_teatru, functie) values (5, 27, 1, 'regizor');

insert into distributie (cod\_montare, cod\_artist, cod\_teatru, functie) values (1, 8, 1, 'costume');

insert into distributie (cod\_montare, cod\_artist, cod\_teatru, functie) values (4, 73, 1, 'regizor');

```

insert into distributie (cod_montare, cod_artist, cod_teatru, functie) values (12, 15, 2, 'regizor');
insert into distributie (cod_montare, cod_artist, cod_teatru, functie) values (14, 27, 3, 'regizor');
insert into distributie (cod_montare, cod_artist, cod_teatru, functie) VALUES (37, 1, 18,
'regizor');
insert into distributie (cod_montare, cod_artist, cod_teatru, functie) VALUES (46, 15, 18,
'regizor');
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 1, 'Sala "Ion
Caramitru"', 940);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 1, 'Sala
Studio', 541);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 1, 'Sala
"Pictura"', 230);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 1, 'Sala Mica',
150);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 2, 'Sala "Liviu
Ciulei"', 250);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 2, 'Sala "Toma
Caragiu"', 200);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 3, 'Sala "Horia
Lovinescu"', 339);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 3, 'Sala Studio
"George Constantin"', 100);
insert into sala (cod_sala, cod_teatru, denumire) values (secv.nextval, 6, 'Sala "Radu Beligan"');
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 8, 'Sala
"Majestic"', 300);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 11, 'Sala
Mare', 300);
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 11, 'Sala
Studio "Bujor Macrin"', 100);
insert into sala (cod_sala, cod_teatru, denumire) values (secv.nextval, 13, 'Sala Mare');
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 13, 'Studio
"Euphorion"', 200);
insert into sala (cod_sala, cod_teatru, denumire) values (secv.nextval, 14, 'Sala Mare');
insert into sala (cod_sala, cod_teatru, denumire) values (secv.nextval, 15, 'Sala "Amza Pellea"');
insert into sala (cod_sala, cod_teatru, denumire, capacitate) values (secv.nextval, 16, 'Sala
Mare', 750);

```

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 16, 'Sala "Teatru la Cub"', 150);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 16, 'Sala Studio "Teofil Valcu"', 100);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 16, 'Sala "Uzina cu Teatru"', 100);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 18, 'Sala Mare', 318);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 18, 'Sala Studio', 100);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 18, 'Sala "Faust"', 450);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 19, 'Sala Mare', 600);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 19, 'Sala 2', 200);

insert into sala (cod\_sala, cod\_teatru, denumire, capacitate) values (secv.nextval, 6, 'Sala Studio', 130);

```
select * from sala;
```

COD_SALA	COD_TEAUTRU	DENUMIRE	CAPACITATE
1	1	1 Sala "Ion Caramitru"	940
2	2	1 Sala Studio	541
3	3	1 Sala "Pictura"	230
4	4	1 Sala Mica	150
5	5	2 Sala "Liviu Ciulei"	250
6	6	2 Sala "Toma Caraciu"	200
7	7	3 Sala "Horia Lovinescu"	339
8	8	3 Sala Studio "George Constantin"	100
9	9	6 Sala "Radu Beligan"	(null)
10	10	8 Sala "Majestic"	300
11	11	11 Sala Mare	300
12	12	11 Sala Studio "Bujor Macrin"	85
13	13	13 Sala Mare	968
14	14	13 Studio "Euphorion"	104
15	15	14 Sala Mare	(null)
16	16	15 Sala "Amza Pellea"	(null)
17	17	16 Sala Mare	750
18	18	16 Sala "Teatru la Cub"	150
19	19	16 Sala Studio "Teofil Valcu"	100
20	20	16 Sala "Uzina cu Teatru"	100
21	21	18 Sala Mare	318
22	22	18 Sala Studio	125
23	23	18 Sala "Faust"	450
24	24	19 Sala Mare	600
25	25	19 Sala 2	200
26	26	6 Sala Studio	130

```

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (1, 1, 1, to_date('27-02-2022', 'dd-mm-yyyy'), to_date('18:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (2, 1, 1, to_date('01-03-2022', 'dd-mm-yyyy'), to_date('18:30', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (3, 2, 1, to_date('14-03-2022', 'dd-mm-yyyy'), to_date('19:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (4, 3, 1, to_date('23-04-2022', 'dd-mm-yyyy'), to_date('19:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (5, 4, 1, to_date('11-05-2022', 'dd-mm-yyyy'), to_date('19:30', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (36, 22, 18, to_date('12-05-2022', 'dd-mm-yyyy'), to_date('18:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (37, 23, 18, to_date('01-05-2022', 'dd-mm-yyyy'), to_date('18:30', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (38, 22, 18, to_date('15-05-2022', 'dd-mm-yyyy'), to_date('19:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (22, 13, 13, to_date('20-04-2022', 'dd-mm-yyyy'), to_date('19:00', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data, ora) values (28, 14, 13, to_date('19-05-2022', 'dd-mm-yyyy'), to_date('19:30', 'hh24:mi'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (6, 4, 1, to_date('31-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (7, 3, 1, to_date('31-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (8, 4, 1, to_date('1-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (9, 3, 1, to_date('2-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (10, 2, 1, to_date('3-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (11, 1, 1, to_date('4-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (12, 5, 2, to_date('5-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (13, 5, 2, to_date('6-12-2022', 'dd-mm-yyyy'));

insert into afis (cod_montare, cod_sala, cod_teatru, data) values (14, 7, 3, to_date('7-12-2022', 'dd-mm-yyyy'));

```

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (19, 9, 6, to\_date('8-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (8, 4, 1, to\_date('12-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 3, 1, to\_date('13-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (10, 2, 1, to\_date('14-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (11, 1, 1, to\_date('15-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (12, 5, 5, to\_date('16-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (13, 3, 6, to\_date('17-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (14, 7, 3, to\_date('18-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (19, 9, 6, to\_date('19-12-2022', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('7-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('8-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('14-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (37, 21, 18, to\_date('15-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (37, 21, 18, to\_date('16-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 2, 1, to\_date('7-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 2, 1, to\_date('14-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('17-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('18-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (1, 1, 1, to\_date('24-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (37, 21, 18, to\_date('25-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (37, 21, 18, to\_date('26-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 2, 1, to\_date('27-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 2, 1, to\_date('19-1-2023', 'dd-mm-yyyy'));

insert into afis (cod\_montare, cod\_sala, cod\_teatru, data) values (9, 2, 1, to\_date('20-1-2023', 'dd-mm-yyyy'));

```
select cod_montare, cod_sala, cod_teatru, data, to_char(ora, 'hh24:mi') from afis;
```

	COD_MONTARE	COD_SALA	COD_TEATRU	DATA	TO_CHAR(ORA, 'HH24:MI')
1	1	1	1	127-FEB-22	18:00
2	2	1	1	101-MAR-22	18:30
3	3	2	2	114-MAR-22	19:00
4	4	3	3	123-APR-22	19:00
5	5	4	4	111-MAY-22	19:30
6	36	22	22	1812-MAY-22	18:00
7	37	23	23	1801-MAY-22	18:30
8	38	22	22	1815-MAY-22	19:00
9	22	13	13	1320-APR-22	19:00
10	28	14	14	1319-MAY-22	19:30
11	9	3	3	113-DEC-22	(null)
12	10	2	2	114-DEC-22	(null)
13	11	1	1	115-DEC-22	(null)
14	14	7	7	318-DEC-22	(null)
15	19	9	9	619-DEC-22	(null)
16	6	4	4	131-DEC-22	(null)
17	7	3	3	131-DEC-22	(null)
18	9	3	3	102-DEC-22	(null)
19	10	2	2	103-DEC-22	(null)
20	11	1	1	104-DEC-22	(null)
21	14	7	7	307-DEC-22	(null)
22	12	5	5	205-DEC-22	(null)
23	13	5	5	206-DEC-22	(null)
24	19	9	9	608-DEC-22	(null)
25	8	4	4	112-DEC-22	(null)
26	1	1	1	114-JAN-23	(null)
27	1	1	1	107-JAN-23	(null)
28	1	1	1	108-JAN-23	(null)
29	37	21	21	1815-JAN-23	(null)
30	37	21	21	1816-JAN-23	(null)
31	9	2	2	107-JAN-23	(null)
32	9	2	2	114-JAN-23	(null)

insert into sponsor(cod\_sponsor, nume) values (secd.nextval, 'Creat');

insert into sponsor(cod\_sponsor, nume) values (secd.nextval, 'Replika');

insert into sponsor(cod\_sponsor, nume) values (secd.nextval, 'Takt');

insert into sponsor(cod\_sponsor, nume) values (secd.nextval, 'Atelier Creat');

insert into sponsor(cod\_sponsor, nume) values (secd.nextval, 'Experiment');

```
select * from sponsor;
```

	COD_SPONSOR	NUME	NR_TELEFON	EMAIL
1	1	Creat	(null)	(null)
2	2	Replika	(null)	(null)
3	3	Takt	(null)	(null)
4	4	Atelier Creat	(null)	(null)
5	5	Experiment	(null)	(null)

```
insert into festival(cod_festival, denumire, locatie, data_inceperii, data_incheierii)
values(secv.nextval, 'Festvalul National de Teatru', 'Bucuresti', to_date('12-10-2022', 'dd-mm-yyyy'), to_date('19-10-2022', 'dd-mm-yyyy'));
```

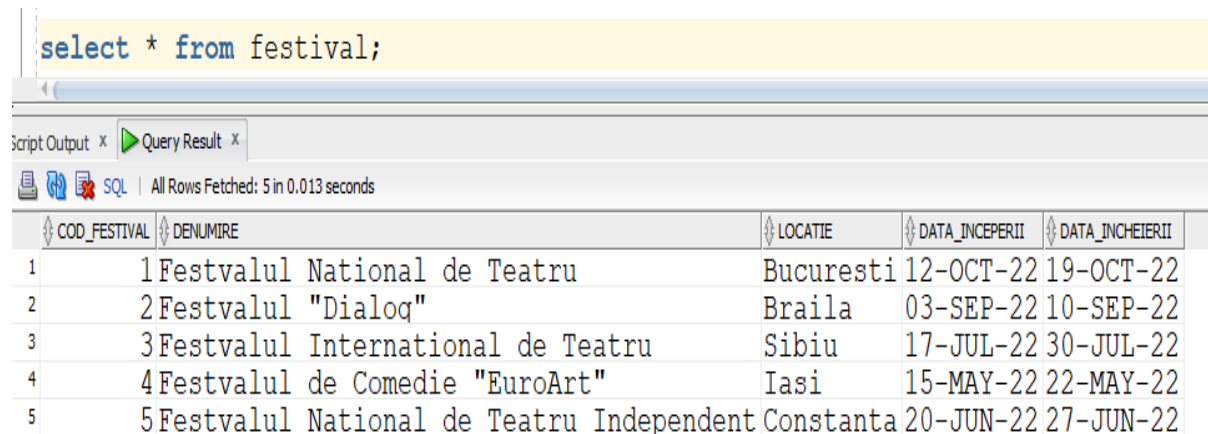
```
insert into festival(cod_festival, denumire, locatie, data_inceperii, data_incheierii)
values(secv.nextval, 'Festvalul "Dialog"', 'Braila', to_date('03-09-2022', 'dd-mm-yyyy'), to_date('10-09-2022', 'dd-mm-yyyy'));
```

```
insert into festival(cod_festival, denumire, locatie, data_inceperii, data_incheierii)
values(secv.nextval, 'Festvalul International de Teatru', 'Sibiu', to_date('17-07-2022', 'dd-mm-yyyy'), to_date('30-07-2022', 'dd-mm-yyyy'));
```

```
insert into festival(cod_festival, denumire, locatie, data_inceperii, data_incheierii)
values(secv.nextval, 'Festvalul de Comedie "EuroArt"', 'Iasi', to_date('15-05-2022', 'dd-mm-yyyy'), to_date('22-05-2022', 'dd-mm-yyyy'));
```

```
insert into festival(cod_festival, denumire, locatie, data_inceperii, data_incheierii)
values(secv.nextval, 'Festvalul National de Teatru Independent', 'Constanta', to_date('20-06-2022', 'dd-mm-yyyy'), to_date('27-06-2022', 'dd-mm-yyyy'));
```

`select * from festival;`



The screenshot shows a database query result in a web interface. The query is `select * from festival;`. The result is displayed in a table with 5 rows and 5 columns: `COD_FESTIVAL`, `DENUMIRE`, `LOCATIE`, `DATA_INCEPERII`, and `DATA_INCHEIERII`. The rows are numbered 1 to 5. The interface also shows 'Script Output' and 'Query Result' tabs, and a status bar indicating 'All Rows Fetched: 5 in 0.013 seconds'.

	COD_FESTIVAL	DENUMIRE	LOCATIE	DATA_INCEPERII	DATA_INCHEIERII
1		1Festvalul National de Teatru	Bucuresti	12-OCT-22	19-OCT-22
2		2Festvalul "Dialog"	Braila	03-SEP-22	10-SEP-22
3		3Festvalul International de Teatru	Sibiu	17-JUL-22	30-JUL-22
4		4Festvalul de Comedie "EuroArt"	Iasi	15-MAY-22	22-MAY-22
5		5Festvalul National de Teatru Independent	Constanta	20-JUN-22	27-JUN-22

```
insert into contract(cod_sponsor, cod_montare, suma) values(1, 1, 2000);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(1, 5, 1000);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(1, 7, 2500);
```

```
insert into contract(cod_sponsor, cod_montare) values(2, 8);
```

```
insert into contract(cod_sponsor, cod_montare) values(2, 10, 1250);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(3, 12, 2250);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(3, 16, 1000);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(4, 21, 3000);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(4, 30, 1750);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(5, 37, 2000);
```

```
insert into contract(cod_sponsor, cod_montare, suma) values(1, 37, 1550);
```



```

insert into contract(cod_sponsor, cod_montare, suma) values(3, 36, 850);
insert into contract(cod_sponsor, cod_montare, suma) values(2, 21, 2300);
insert into contract(cod_sponsor, cod_montare, suma) values(4, 22, 1900);
insert into contract(cod_sponsor, cod_montare, suma) values(5, 37, 2000);
insert into contract(cod_sponsor, cod_montare, suma) values(1, 14, 1850);
insert into contract(cod_sponsor, cod_montare, suma) values(2, 19, 1450);
insert into contract(cod_sponsor, cod_montare, suma) values(3, 20, 1350);
insert into contract(cod_sponsor, cod_montare, suma) values(3, 39, 1750);
insert into contract(cod_sponsor, cod_montare, suma) values(3, 4, 2400);

insert into contract(cod_sponsor, cod_montare, suma)
values(3, 1, 2500);

insert into contract(cod_sponsor, cod_montare, suma)
values(2, 4, 2400);

insert into contract(cod_sponsor, cod_montare, suma)
values(2, 3, 2300);

insert into contract(cod_sponsor, cod_montare, suma)
values(2, 28, 2500);

insert into contract(cod_sponsor, cod_montare, suma)
values(2, 40, 2800);

insert into contract(cod_sponsor, cod_montare, suma)
values(1, 40, 2200);

insert into contract(cod_sponsor, cod_montare, suma)
values(1, 28, 2600);

insert into contract(cod_sponsor, cod_montare, suma)
values(1, 3, 3200);

```

select \* from contract;

Script Output x Query Result x

All Rows Fetched: 22 in 0.003 seconds

	COD_SPONSOR	COD_MONTARE	SUMA
1	3	39	1750
2	1	1	2000
3	1	5	1000
4	1	7	2500
5	2	8	(null)
6	3	12	2250
7	3	16	1000
8	4	21	3000
9	4	30	1750
10	5	37	2000
11	3	4	2400
12	3	1	2500
13	2	4	2400
14	2	3	2300
15	2	28	2500
16	2	40	2800
17	1	40	2200
18	1	28	2600
19	1	3	3200
20	1	14	1850
21	2	19	1450
22	3	20	1350

```

insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(1, 12, 1, 2,
'12-OCT-2022', to_date('18:00', 'hh24:mi'));

insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(1, 3, 1, 3,
'14-OCT-2022', to_date('19:00', 'hh24:mi'));

insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(1, 14, 2, 5,
'18-OCT-2022', to_date('18:00', 'hh24:mi'));

insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(2, 32, 11,
12, '04-SEP-2022', to_date('18:00', 'hh24:mi'));

insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(2, 25, 11,
11, '07-SEP-2022', to_date('18:30', 'hh24:mi'));

```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data, ora) values(3, 37, 18, 21, '17-JUL-2022', to_date('18:00', 'hh24:mi'));
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 17, 18, 22, '19-JUL-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 29, 18, 22, '23-JUL-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(4, 33, 14, 15, '16-MAY-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(5, 19, 14, 15, '25-JUN-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 19, 14, 15, '25-JUN-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(1, 4, 1, 2, '13-OCT-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(1, 5, 1, 4, '15-OCT-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(1, 46, 2, 6, '17-OCT-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 4, 18, 21, '18-JUL-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 5, 18, 22, '21-JUL-2022');
```

```
insert into agenda(cod_festival, cod_montare, cod_teatru, cod_sala, data) values(3, 46, 18, 23, '26-JUL-2022');
```

```
insert into agenda (cod_festival, cod_montare, cod_teatru, cod_sala) values (2, 50, 11, 12);
```

```
insert into agenda (cod_festival, cod_montare, cod_teatru, cod_sala) values (2, 51, 11, 11);
```

```
insert into agenda (cod_festival, cod_montare, cod_teatru, cod_sala) values (2, 17, 11, 12);
```

select \* from agenda;

	COD_FESTIVAL	COD_MONTARE	COD_TEATRU	COD_SALA	DATA	ORA
1	2	50	11	12	(null)	(null)
2	2	51	11	11	(null)	(null)
3	2	17	11	12	(null)	(null)
4	1	4	1	2	13-OCT-22	(null)
5	1	5	1	4	15-OCT-22	(null)
6	1	46	2	6	17-OCT-22	(null)
7	3	4	18	21	18-JUL-22	(null)
8	3	5	18	22	21-JUL-22	(null)
9	1	12	1	2	12-OCT-22	01-APR-22
10	3	46	18	23	26-JUL-22	(null)
11	1	3	1	3	14-OCT-22	01-APR-22
12	1	14	2	5	18-OCT-22	01-APR-22
13	2	25	11	11	07-SEP-22	01-APR-22
14	3	17	18	22	19-JUL-22	(null)
15	3	29	18	22	23-JUL-22	(null)
16	5	19	14	15	25-JUN-22	(null)
17	2	32	11	12	04-SEP-22	01-APR-22
18	3	37	18	21	17-JUL-22	01-APR-22
19	4	33	14	15	16-MAY-22	(null)

## 6. (tip de subprogram: procedură, tipuri de colecții: tablou imbricat, tablou indexat)

Sa se obtina, pentru o denumire de sala data ca parametru, o lista a montarilor care se desfasoara in sala respectiva. Pentru fiecare montare, lista va cuprinde data premierei, genul piesei, precum si lista artistilor care au participat la realizarea montarii, impreuna cu specializarea fiecaruia. Artistii vor fi ordonati alfabetic, iar in cazul artistilor care indeplinesc mai multe functii in cadrul aceleiasi montari, acestea vor fi enumerate in dreptul numelui artistului, separate prin virgula. In cazul in care in sala data ca parametru nu este planificata nicio montare, se va afisa un mesaj corespunzator. Se vor trata cazurile in care nu exista nicio sala cu denumirea data ca parametru, sau exista mai multe sali cu denumirea data ca parametru.

```
CREATE OR REPLACE PROCEDURE info_sala(denumire_sala sala.denumire%TYPE)
```

```
IS
```

```
    TYPE info_pers IS RECORD (  
        prenume persoana.prenume%TYPE,  
        nume persoana.nume%TYPE,  
        spec specializare.denumire%TYPE,  
        cod_spec specializare.cod_specializare%TYPE  
    );
```

```
    TYPE lista_artisti IS TABLE OF info_pers;
```

```
    artisti lista_artisti;
```

```
    TYPE info_montare IS RECORD (  
        data_pr montare.data_premierei%TYPE,  
        gen piesa.gen%TYPE,  
        distrib lista_artisti  
    );
```

```
    TYPE lista_info_montari IS TABLE OF info_montare INDEX BY piesa.titlu%TYPE;
```

```
    piese lista_info_montari;
```

```
    cod sala.cod_sala%TYPE;
```

```
    p piesa.titlu%TYPE;
```

```
    nume_ant persoana.nume%TYPE;
```

```
    prenume_ant persoana.prenume%TYPE;
```

```
    specializari VARCHAR2(100);
```

```

BEGIN
    SELECT cod_sala INTO cod
    FROM sala
    WHERE denumire = denumire_sala;
    FOR i IN
        (SELECT m.cod_montare cod, titlu, data_premierei, gen
         FROM piesa p, montare m, afis a
         WHERE p.cod_piesa = m.cod_piesa
         AND m.cod_montare = a.cod_montare
         AND a.cod_sala = cod) LOOP
        SELECT p.prenume, p.num, s.denumire, s.cod_specializare BULK COLLECT INTO
        artisti
        FROM persoana p, artist a, distributie d, spec_art sa, specializare s
        WHERE p.cod_persoana = a.cod_persoana
        AND a.cod_persoana = d.cod_artist
        AND d.cod_montare = i.cod
        AND a.cod_persoana = sa.cod_artist
        AND sa.cod_specializare = s.cod_specializare
        AND functie = s.denumire
        ORDER BY 2, 1;
        SELECT i.data_premierei, i.gen INTO piese(i.titlu).data_pr, piese(i.titlu).gen
        FROM DUAL;
        piese(i.titlu).distrib := artisti;
    END LOOP;
    IF piese.COUNT > 0 THEN
        dbms_output.put_line('Detaliile montarilor desfasurate in ' || denumire_sala || ' sunt
        urmatoarele:');
    ELSE
        dbms_output.put_line('Nu exista inca nicio montare planificata in ' || denumire_sala);
    END IF;
    p := piese.FIRST;
    LOOP

```

```

EXIT WHEN p IS NULL;

    dbms_output.put_line('Montarea piesei "' || p || '" (' || NVL(piese(p).gen, 'gen nespecificat')
|| '), a carei data a premierei este ' || NVL(TO_CHAR(piese(p).data_pr, 'DD-MON-YYYY'),
'nespecificata') || '.');

    IF piese(p).distrib.COUNT > 0 THEN

        dbms_output.put_line('La realizarea acestei montari au participat urmatorii artisti: ');

    ELSE

        dbms_output.put_line('Nu se cunosc detalii despre artistii care au participat la aceast
montare.');
```

dbms\_output.new\_line();

p := piese.NEXT(p);

CONTINUE;

END IF;

prenume\_ant := piese(p).distrib(piese(p).distrib.FIRST).prenume;

nume\_ant := piese(p).distrib(piese(p).distrib.FIRST).nume;

specializari := ";

FOR i IN 1..piese(p).distrib.COUNT LOOP

    IF (piese(p).distrib(i).prenume <> prenume\_ant OR piese(p).distrib(i).nume <> nume\_ant) THEN

        dbms\_output.put\_line(' ' || prenume\_ant || ' ' || nume\_ant || ' - ' || specializari);

        specializari := piese(p).distrib(i).spec || ', ';

    ELSE

        specializari := piese(p).distrib(i).spec || ', ' || specializari;

    END IF;

    prenume\_ant := piese(p).distrib(i).prenume;

    nume\_ant := piese(p).distrib(i).nume;

END LOOP;

dbms\_output.put\_line(' ' || piese(p).distrib(piese(p).distrib.COUNT).prenume || ' ' || piese(p).distrib(piese(p).distrib.COUNT).nume

    || ' - ' || SUBSTR(specializari, 1, LENGTH(specializari) - 2) || '.');

dbms\_output.new\_line();

p := piese.NEXT(p);

END LOOP;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multe sali cu denumirea "' || denumire\_sala || "'.');)

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista nicio sala cu denumirea "' || denumire\_sala || "'.');)

END info\_sala;

/

```
CREATE OR REPLACE PROCEDURE info_sala(denumire_sala sala.denumire%TYPE)
IS
    TYPE info_pers IS RECORD (
        prenume persoana.prenume%TYPE,
        nume persoana.nume%TYPE,
        spec specializare.denumire%TYPE,
        cod_spec specializare.cod_specializare%TYPE
    );
    TYPE lista_artisti IS TABLE OF info_pers;
    artisti lista_artisti;
    TYPE info_montare IS RECORD (
        data_pr montare.data_premierei%TYPE,
        gen piesa.gen%TYPE,
        distrib lista_artisti
    );
    TYPE lista_info_montari IS TABLE OF info_montare INDEX BY piesa.titlu%TYPE;
    piese lista_info_montari;
    cod sala.cod_sala%TYPE;
    p piesa.titlu%TYPE;
    nume_ant persoana.nume%TYPE;
    prenume_ant persoana.prenume%TYPE;
    specializari VARCHAR2(100);
BEGIN
    SELECT cod_sala INTO cod
    FROM sala

    SELECT cod_sala INTO cod
    FROM sala
    WHERE denumire = denumire_sala;
    FOR i IN
        (SELECT m.cod_montare cod, titlu, data_premierei, gen
        FROM piesa p, montare m, afis a
        WHERE p.cod_piesa = m.cod_piesa
        AND m.cod_montare = a.cod_montare
        AND a.cod_sala = cod) LOOP
        SELECT p.prenume, p.nume, s.denumire, s.cod_specializare BULK COLLECT INTO artisti
        FROM persoana p, artist a, distributie d, spec_art sa, specializare s
        WHERE p.cod_persoana = a.cod_persoana
        AND a.cod_persoana = d.cod_artist
        AND d.cod_montare = i.cod
        AND a.cod_persoana = sa.cod_artist
        AND sa.cod_specializare = s.cod_specializare
        AND functie = s.denumire
        ORDER BY 2, 1;
        SELECT i.data_premierei, i.gen INTO piese(i.titlu).data_pr, piese(i.titlu).gen
        FROM DUAL;
        piese(i.titlu).distrib := artisti;
    END LOOP;
    IF piese.COUNT > 0 THEN
        dbms_output.put_line('Detaliile montarilor desfasurate in ' || denumire_sala || ' sunt urmatoarele:');
    ELSE
        dbms_output.put_line('Nu exista inca nicio montare planificata in ' || denumire_sala);
    END IF;
```

Script Output x Query Result x  
Task completed in 0.105 seconds  
Procedure INFO\_SALA compiled

```
SELECT cod_sala INTO cod
FROM sala
WHERE denumire = denumire_sala;
FOR i IN
    (SELECT m.cod_montare cod, titlu, data_premierei, gen
    FROM piesa p, montare m, afis a
    WHERE p.cod_piesa = m.cod_piesa
    AND m.cod_montare = a.cod_montare
    AND a.cod_sala = cod) LOOP
    SELECT p.prenume, p.nume, s.denumire, s.cod_specializare BULK COLLECT INTO artisti
    FROM persoana p, artist a, distributie d, spec_art sa, specializare s
    WHERE p.cod_persoana = a.cod_persoana
    AND a.cod_persoana = d.cod_artist
    AND d.cod_montare = i.cod
    AND a.cod_persoana = sa.cod_artist
    AND sa.cod_specializare = s.cod_specializare
    AND functie = s.denumire
    ORDER BY 2, 1;
    SELECT i.data_premierei, i.gen INTO piese(i.titlu).data_pr, piese(i.titlu).gen
    FROM DUAL;
    piese(i.titlu).distrib := artisti;
END LOOP;
IF piese.COUNT > 0 THEN
    dbms_output.put_line('Detaliile montarilor desfasurate in ' || denumire_sala || ' sunt urmatoarele:');
ELSE
    dbms_output.put_line('Nu exista inca nicio montare planificata in ' || denumire_sala);
END IF;
```

Script Output x  
Task completed in 0.073 seconds  
Procedure INFO\_SALA compiled

```

p := piese.FIRST;
LOOP
  EXIT WHEN p IS NULL;
  dbms_output.put_line('Montarea piesei "' || p || '" (' || NVL(piese(p).gen, 'gen nespecificat') || '), a carei data a premierei este '
    || NVL(TO_CHAR(piese(p).data_pr, 'DD-MON-YYYY'), 'nespecificata') || '.');
  IF piese(p).distrib.COUNT > 0 THEN
    dbms_output.put_line('La realizarea acestei montari au participat urmatoorii artisti: ');
  ELSE
    dbms_output.put_line('Nu se cunosc detalii despre artistii care au participat la aceast montare.');
```

```

    dbms_output.new_line();
    p := piese.NEXT(p);
    CONTINUE;
  END IF;
  prenume_ant := piese(p).distrib(piese(p).distrib.FIRST).prenume;
  nume_ant := piese(p).distrib(piese(p).distrib.FIRST).nume;
  specializari := '';
  FOR i IN 1..piese(p).distrib.COUNT LOOP
    IF (piese(p).distrib(i).prenume <> prenume_ant OR piese(p).distrib(i).nume <> nume_ant) THEN
      dbms_output.put_line(' ' || prenume_ant || ' ' || nume_ant || ' - ' || specializari);
      specializari := piese(p).distrib(i).spec || ', ';
    ELSE
      specializari := piese(p).distrib(i).spec || ', ' || specializari;
    END IF;
    prenume_ant := piese(p).distrib(i).prenume;
    nume_ant := piese(p).distrib(i).nume;
  END LOOP;
  dbms_output.put_line(' ' || piese(p).distrib(piese(p).distrib.COUNT).prenume || ' ' || piese(p).distrib(piese(p).distrib.COUNT).nume
```

Script Output x  
Task completed in 0.073 seconds

Procedure INFO SALA compiled

```

    p := piese.NEXT(p);
  END LOOP;
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    dbms_output.put_line('Exista mai multe sali cu denumirea "' || denumire_sala || '".');
  WHEN NO_DATA_FOUND THEN
    dbms_output.put_line('Nu exista nicio sala cu denumirea "' || denumire_sala || '".');
END info_sala;
/
```

Script Output x  
Task completed in 0.073 seconds

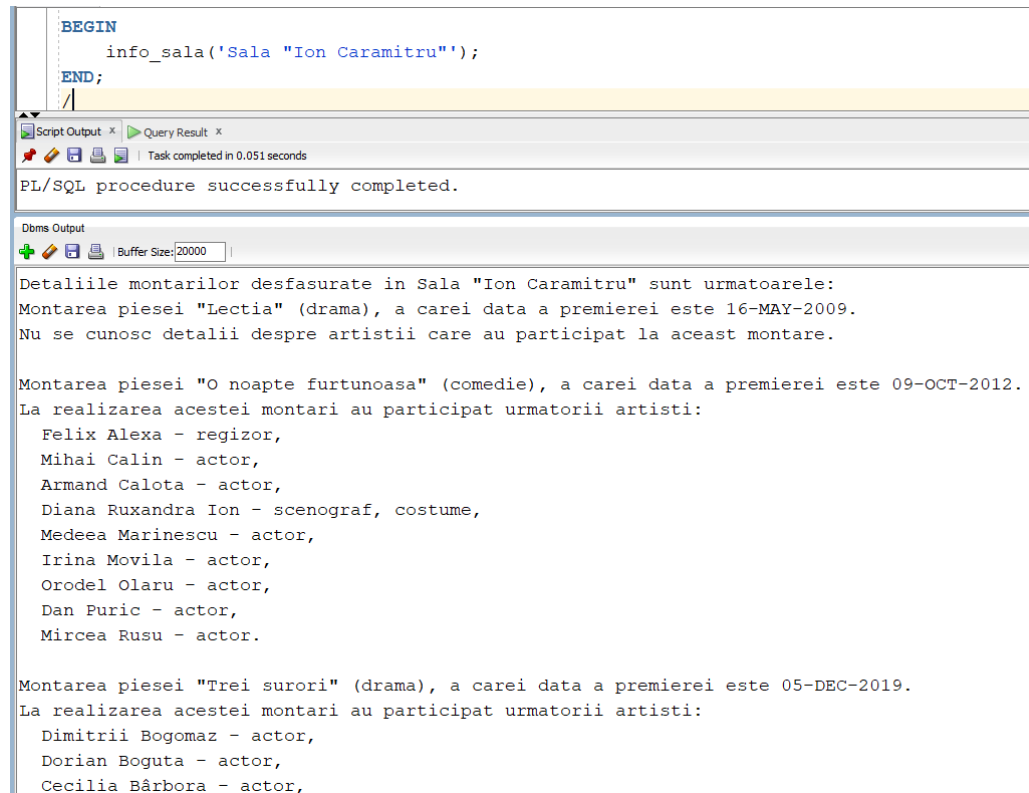
Procedure INFO SALA compiled

BEGIN

info\_sala('Sala "Ion Caramitru"');

END;

/



The screenshot shows a database IDE with a script editor at the top containing the following SQL code:

```
BEGIN
    info_sala('Sala "Ion Caramitru"');
END;
```

Below the script editor, a status bar indicates "Task completed in 0.051 seconds". A message pane displays "PL/SQL procedure successfully completed.".

The "Dbms Output" pane at the bottom shows the following text:

Detaliile montarilor desfasurate in Sala "Ion Caramitru" sunt urmatoarele:  
Montarea piesei "Lectia" (drama), a carei data a premierei este 16-MAY-2009.  
Nu se cunosc detalii despre artistii care au participat la aceast montare.

Montarea piesei "O noapte furtunoasa" (comedie), a carei data a premierei este 09-OCT-2012.  
La realizarea acestei montari au participat urmatorii artisti:

- Felix Alexa - regizor,
- Mihai Calin - actor,
- Armand Calota - actor,
- Diana Ruxandra Ion - scenograf, costume,
- Medeea Marinescu - actor,
- Irina Movila - actor,
- Orodel Olaru - actor,
- Dan Puric - actor,
- Mircea Rusu - actor.

Montarea piesei "Trei surori" (drama), a carei data a premierei este 05-DEC-2019.  
La realizarea acestei montari au participat urmatorii artisti:

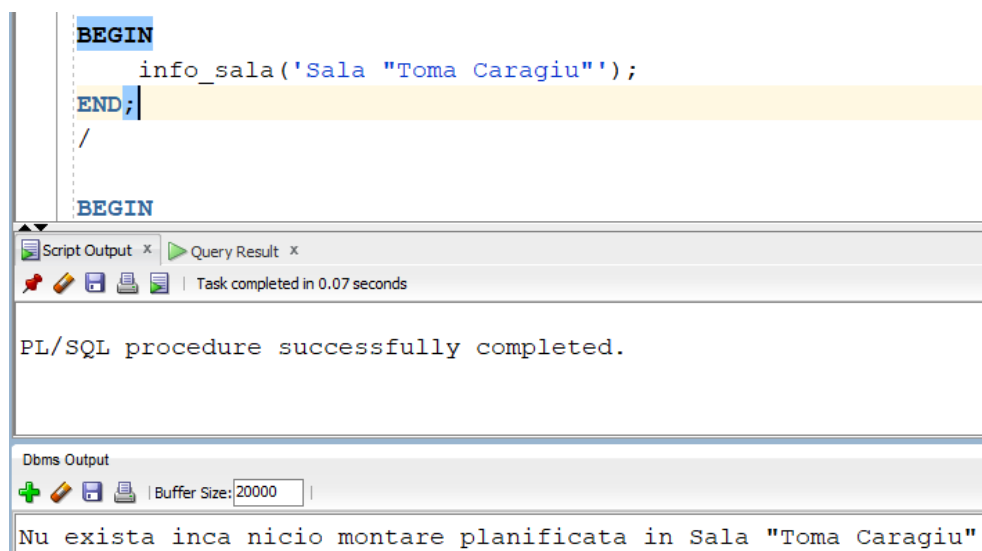
- Dimitrii Bogomaz - actor,
- Dorian Boguta - actor,
- Cecilia Bârbora - actor,

BEGIN

info\_sala('Sala "Toma Caragiu"');

END;

/



The screenshot shows a database IDE with a script editor at the top containing the following SQL code:

```
BEGIN
    info_sala('Sala "Toma Caragiu"');
END;
```

Below the script editor, a status bar indicates "Task completed in 0.07 seconds". A message pane displays "PL/SQL procedure successfully completed.".

The "Dbms Output" pane at the bottom shows the following text:

Nu exista inca nicio montare planificata in Sala "Toma Caragiu"

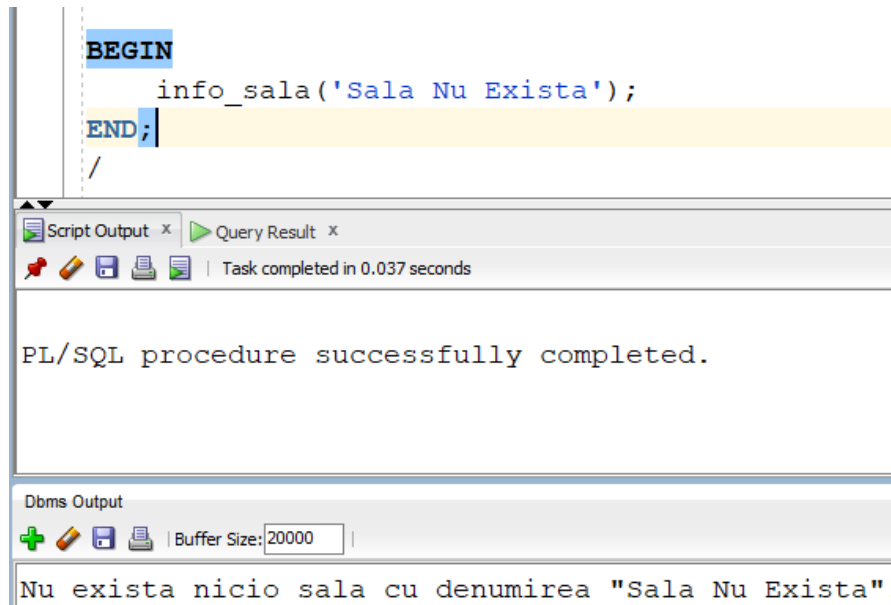


BEGIN

info\_sala('Sala Nu Exista');

END;

/



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

```
BEGIN
    info_sala('Sala Nu Exista');
END;
```

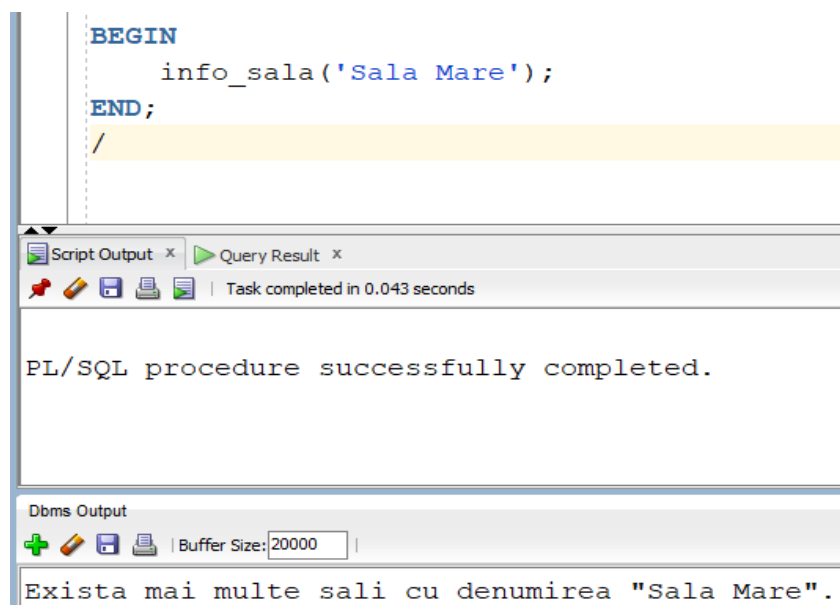
The script is executed, and the 'Script Output' window shows the message: 'PL/SQL procedure successfully completed.' The 'Dbms Output' window shows the message: 'Nu exista nicio sala cu denumirea "Sala Nu Exista"'.

BEGIN

info\_sala('Sala Mare');

END;

/



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

```
BEGIN
    info_sala('Sala Mare');
END;
```

The script is executed, and the 'Script Output' window shows the message: 'PL/SQL procedure successfully completed.' The 'Dbms Output' window shows the message: 'Exista mai multe sali cu denumirea "Sala Mare"'.

## 7. (tip de subprogram: procedură, tipuri de cursoare: cursor parametrizat, cursor dinamic)

Pentru un cod de festival dat de la tastatura, sa se obtina montarile care se afla pe agenda festivalului respectiv si care apartin teatrelor din localitatea in care se desfasoara festivalul. Lista va contine, pentru fiecare montare, codul acesteia, denumirea piesei, codul teatrului de care apartine si orasul in care acesta se afla. Apoi, in cazul in care numarul de montari obtinute anterior este nenul si mai mare sau egal decat 50% din totalul montarilor programate in festival, sa se obtina numele regizorilor care au realizat montarile respective, iar, in caz contrar, daca, numarul de montari este nenul, fondurile totale alocate pentru rezalizarea acestora (fondurile totale alocate unei montari sunt provenite din bugetul de stat si din contractele cu sponsorii). Fondurile totale vor fi afisate sub forma unei sume, in care fiecare termen reprezinta suma alocata pentru o montare. Daca numarul de montari obtinute initial este egal cu 0, se va afisa un mesaj corespunzator. Se va trata cazul in care nu exista niciun festival cu codul introdus.

```
CREATE OR REPLACE TYPE lista_mont_loc IS TABLE OF NUMBER(4);
```

```
/
```

```
CREATE OR REPLACE PROCEDURE info_festival(cod festival.cod_festival%TYPE)
```

```
IS
```

```
    CURSOR c_mont_loc(cod festival.cod_festival%TYPE) IS
```

```
        SELECT m.cod_montare, titlu, t1.denumire, ad1.oras
```

```
        FROM agenda a, montare m, piesa p, sala s, adresa ad1, adresa ad2, teatru t1, teatru t2
```

```
        WHERE a.cod_festival = cod
```

```
        AND a.cod_montare = m.cod_montare
```

```
        AND a.cod_sala = s.cod_sala
```

```
        AND s.cod_teatru = t1.cod_teatru
```

```
        AND t1.cod_adresa = ad1.cod_adresa
```

```
        AND m.cod_teatru = t2.cod_teatru
```

```
        AND t2.cod_adresa = ad2.cod_adresa
```

```
        AND ad1.oras = ad2.oras
```

```
        AND m.cod_piesa = p.cod_piesa;
```

```
    TYPE c_tip is REF CURSOR;
```

```
    v_cursor c_tip;
```

```
    v_montare montare.cod_montare%TYPE;
```

```
    v_piesa piesa.titlu%TYPE;
```

```

v_teatru teatru.denumire%TYPE;
v_oras adresa.oras%TYPE;
v_nume persoana.nume%TYPE;
v_prenume persoana.prenume%TYPE;
v_suma contract.suma%TYPE := 0;
i NUMBER := 1;
v_denumire_fest festival.denumire%TYPE;
v_nr NUMBER;
v_total NUMBER;
v_buget montare.buget%TYPE;
montari lista_mont_loc := lista_mont_loc();
expresie VARCHAR2(100) := "";
BEGIN
    SELECT denumire INTO v_denumire_fest
    FROM festival
    WHERE cod_festival = cod;
    OPEN c_mont_loc(cod);
    LOOP
        FETCH c_mont_loc INTO v_montare, v_piesa, v_teatru, v_oras;
        montari.extend;
        montari(i) := v_montare;
        i := i + 1;
        EXIT WHEN c_mont_loc%NOTFOUND;
        IF i = 2 THEN
            dbms_output.new_line();
            dbms_output.put_line('Piesele prezentate in festival, care au fost montate in teatre din
localitatea organizatoare sunt urmatoarele: ');
            END IF;
            dbms_output.put_line(' piesa "' || v_piesa || '" , montata la ' || v_teatru || ', din ' || v_oras || ');
        END LOOP;
        v_nr := c_mont_loc%ROWCOUNT;
        CLOSE c_mont_loc;

```

```

SELECT COUNT(cod_montare) INTO v_total
FROM agenda a
WHERE a.cod_festival = cod;
IF v_nr >= 0.5 * v_total AND v_nr > 0 THEN
    OPEN v_cursor FOR SELECT DISTINCT nume, prenume
        FROM persoana p, artist a, distributie d, spec_art sa, specializare s
        WHERE p.cod_persoana = a.cod_persoana
        AND a.cod_persoana = d.cod_artist
        AND a.cod_persoana = sa.cod_artist
        AND sa.cod_specializare = s.cod_specializare
        AND s.cod_specializare = cod
        AND d.cod_montare IN (SELECT * FROM TABLE(montari));
    dbms_output.put_line('Regizorii care au realizat montari in teatrele din localitatea
organizatoare sunt: ');
    LOOP
        FETCH v_cursor INTO v_nume, v_prenume;
        EXIT WHEN v_cursor%NOTFOUND;
        dbms_output.put_line(' ' || v_prenume || ' ' || v_nume);
    END LOOP;
    CLOSE v_cursor;
ELSIF v_nr > 0 THEN
    OPEN v_cursor FOR SELECT SUM(suma), buget, m.cod_montare
        FROM contract c, montare m
        WHERE c.cod_montare(+) = m.cod_montare
        AND m.cod_montare IN (SELECT * FROM TABLE(montari))
        GROUP BY m.cod_montare, buget;
    v_total := 0;
    dbms_output.new_line();
    dbms_output.put_line('Pentru realizarea montarilor din localitatea organizatoare, s-au alocat
fonduri in valoare de: ');
    LOOP
        FETCH v_cursor INTO v_suma, v_buget, v_montare;

```

```

EXIT WHEN v_cursor%NOTFOUND;

expresie := expresie || (v_buget + NVL(v_suma, 0)) || ' + ';

v_total := v_total + NVL(v_suma, 0) + v_buget;

END LOOP;

dbms_output.put_line(SUBSTR(expresie, 1, LENGTH(expresie) - 3) || ' = ' || v_total);

CLOSE v_cursor;

ELSE

    dbms_output.put_line('In festival nu a fost prezentata nicio piesa montata in localitatea organizatoare.');
```

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista niciun festival cu codul introdus.');

END info\_festival;

/

CREATE OR REPLACE TYPE lista\_mont\_loc IS TABLE OF NUMBER(4);

/

Script Output x

Task completed in 0.098 seconds

Type LISTA\_MONT\_LOC compiled

```

BEGIN
SELECT denumire INTO v_denumire_fest
FROM festival
WHERE cod_festival = cod;
OPEN c_mont_loc(cod);
LOOP
    FETCH c_mont_loc INTO v_montare, v_piesa, v_teatru, v_oras;
    montari.extend;
    montari(i) := v_montare;
    i := i + 1;
    EXIT WHEN c_mont_loc%NOTFOUND;
    IF i = 2 THEN
        dbms_output.new_line();
        dbms_output.put_line('Piese prezentate in festival, care au fost montate in teatre din localitatea organizatoare sunt urmatoarele: ');
    END IF;
    dbms_output.put_line(' piesa "' || v_piesa || '" montata la ' || v_teatru || ', din ' || v_oras || ');';
END LOOP;
v_nr := c_mont_loc%ROWCOUNT;
CLOSE c_mont_loc;
SELECT COUNT(cod_montare) INTO v_total
FROM agenda a
WHERE a.cod_festival = cod;
IF v_nr >= 0.5 * v_total AND v_nr > 0 THEN
    OPEN v_cursor FOR SELECT DISTINCT nume, prenume
                        FROM persoana p, artist a, distributie d, spec_art sa, specializare s
```

Script Output x

Task completed in 0.096 seconds

Procedure INFO\_FESTIVAL compiled

```

CREATE OR REPLACE PROCEDURE info_festival(cod festival.cod_festival%TYPE)
IS
    CURSOR c_mont_loc(cod festival.cod_festival%TYPE) IS
        SELECT m.cod_montare, titlu, t1.denumire, ad1.oras
        FROM agenda a, montare m, piesa p, sala s, adresa ad1, adresa ad2, teatru t1, teatru t2
        WHERE a.cod_festival = cod
        AND a.cod_montare = m.cod_montare
        AND a.cod_sala = s.cod_sala
        AND s.cod_teatru = t1.cod_teatru
        AND t1.cod_adresa = ad1.cod_adresa
        AND m.cod_teatru = t2.cod_teatru
        AND t2.cod_adresa = ad2.cod_adresa
        AND ad1.oras = ad2.oras
        AND m.cod_piesa = p.cod_piesa;
    TYPE c_tip IS REF CURSOR;
    v_cursor c_tip;
    v_montare montare.cod_montare%TYPE;
    v_piesa piesa.titlu%TYPE;
    v_teatru teatru.denumire%TYPE;
    v_oras adresa.oras%TYPE;
    v_nume persoana.nume%TYPE;
    v_prenume persoana.prenume%TYPE;
    v_suma contract.suma%TYPE := 0;
    i NUMBER := 1;
    v_denumire_fest festival.denumire%TYPE;

```

Script Output x | Task completed in 0.09 seconds

Procedure INFO\_FESTIVAL compiled

```

        FROM persoana p, artist a, distributie d, spec_art sa, specializare s
        WHERE p.cod_persoana = a.cod_persoana
        AND a.cod_persoana = d.cod_artist
        AND a.cod_persoana = sa.cod_artist
        AND sa.cod_specializare = s.cod_specializare
        AND s.cod_specializare = cod
        AND d.cod_montare IN (SELECT * FROM TABLE(montari));
    dbms_output.put_line('Regizorii care au realizat montari in teatrele din localitatea organizatoare sunt: ');
    LOOP
        FETCH v_cursor INTO v_nume, v_prenume;
        EXIT WHEN v_cursor%NOTFOUND;
        dbms_output.put_line(' ' || v_prenume || ' ' || v_nume);
    END LOOP;
    CLOSE v_cursor;
    ELSIF v_nr > 0 THEN
        OPEN v_cursor FOR SELECT SUM(suma), buget, m.cod_montare
        FROM contract c, montare m
        WHERE c.cod_montare(+) = m.cod_montare
        AND m.cod_montare IN (SELECT * FROM TABLE(montari))
        GROUP BY m.cod_montare, buget;

        v_total := 0;
        dbms_output.new_line();
        dbms_output.put_line('Pentru realizarea montarilor din localitatea organizatoare, s-au alocat fonduri in valoare de: ');
        LOOP
            FETCH v_cursor INTO v_suma, v_buget, v_montare;

```

Script Output x | Query Result x | Task completed in 0.096 seconds

Procedure INFO\_FESTIVAL compiled

```

        LOOP
            FETCH v_cursor INTO v_suma, v_buget, v_montare;
            EXIT WHEN v_cursor%NOTFOUND;
            expresie := expresie || (v_buget + NVL(v_suma, 0)) || ' + ';
            v_total := v_total + NVL(v_suma, 0) + v_buget;
        END LOOP;
        dbms_output.put_line(SUBSTR(expresie, 1, LENGTH(expresie) - 3) || ' = ' || v_total);
        CLOSE v_cursor;
    ELSE
        dbms_output.put_line('In festival nu a fost prezentata nicio piesa montata in localitatea organizatoare.');

```

Script Output x | Query Result x | Task completed in 0.096 seconds

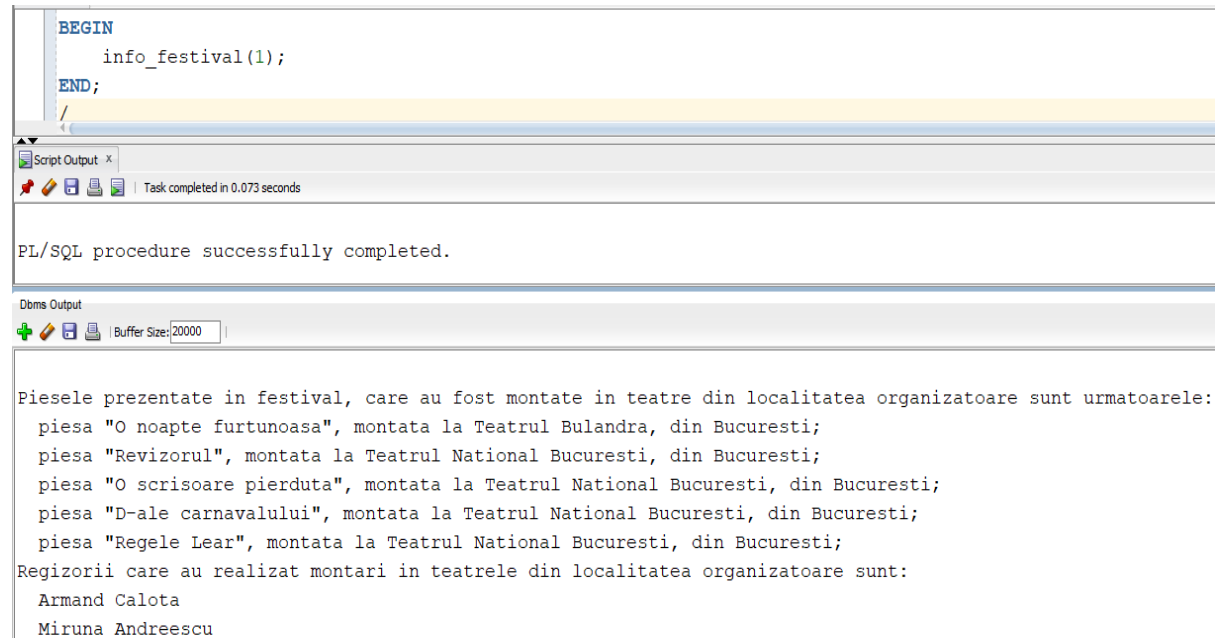
Procedure INFO\_FESTIVAL compiled

BEGIN

info\_festival(1);

END;

/



```
BEGIN
    info_festival(1);
END;
/
```

Script Output x | Task completed in 0.073 seconds

PL/SQL procedure successfully completed.

Dbms Output | Buffer Size: 20000

Pieseie prezentate in festival, care au fost montate in teatre din localitatea organizatoare sunt urmatoarele:

- piesa "O noapte furtunoasa", montata la Teatrul Bulandra, din Bucuresti;
- piesa "Revizorul", montata la Teatrul National Bucuresti, din Bucuresti;
- piesa "O scrisoare pierduta", montata la Teatrul National Bucuresti, din Bucuresti;
- piesa "D-ale carnavalului", montata la Teatrul National Bucuresti, din Bucuresti;
- piesa "Regele Lear", montata la Teatrul National Bucuresti, din Bucuresti;

Regizorii care au realizat montari in teatrele din localitatea organizatoare sunt:

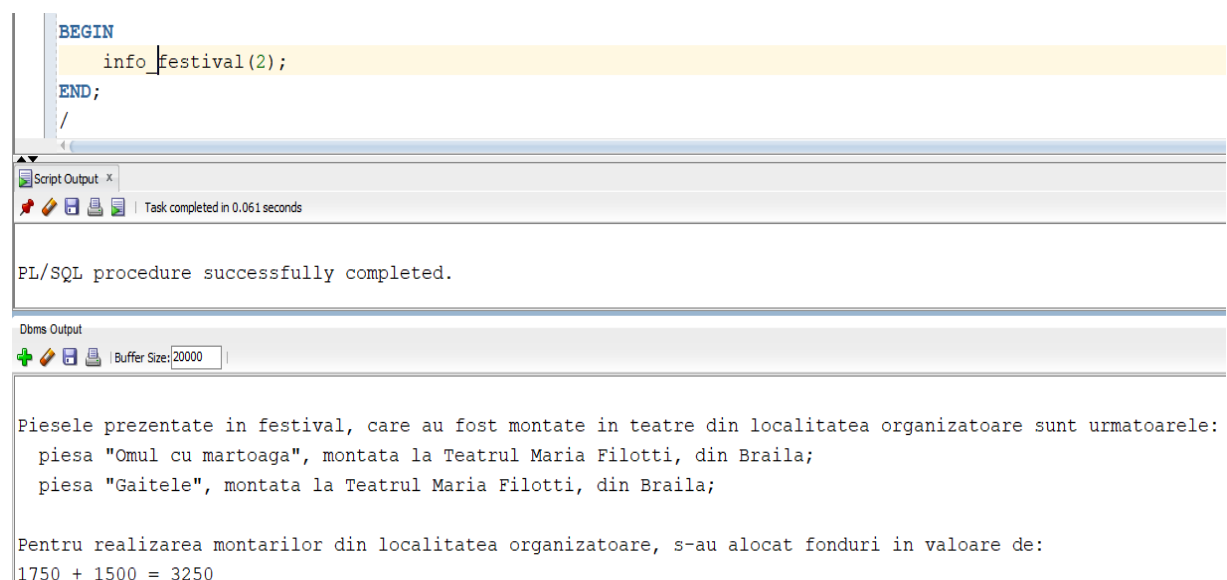
- Armand Calota
- Miruna Andreescu

BEGIN

info\_festival(2);

END;

/



```
BEGIN
    info_festival(2);
END;
/
```

Script Output x | Task completed in 0.061 seconds

PL/SQL procedure successfully completed.

Dbms Output | Buffer Size: 20000

Pieseie prezentate in festival, care au fost montate in teatre din localitatea organizatoare sunt urmatoarele:

- piesa "Omul cu martoaga", montata la Teatrul Maria Filotti, din Braila;
- piesa "Gaitele", montata la Teatrul Maria Filotti, din Braila;

Pentru realizarea montarilor din localitatea organizatoare, s-au alocat fonduri in valoare de:

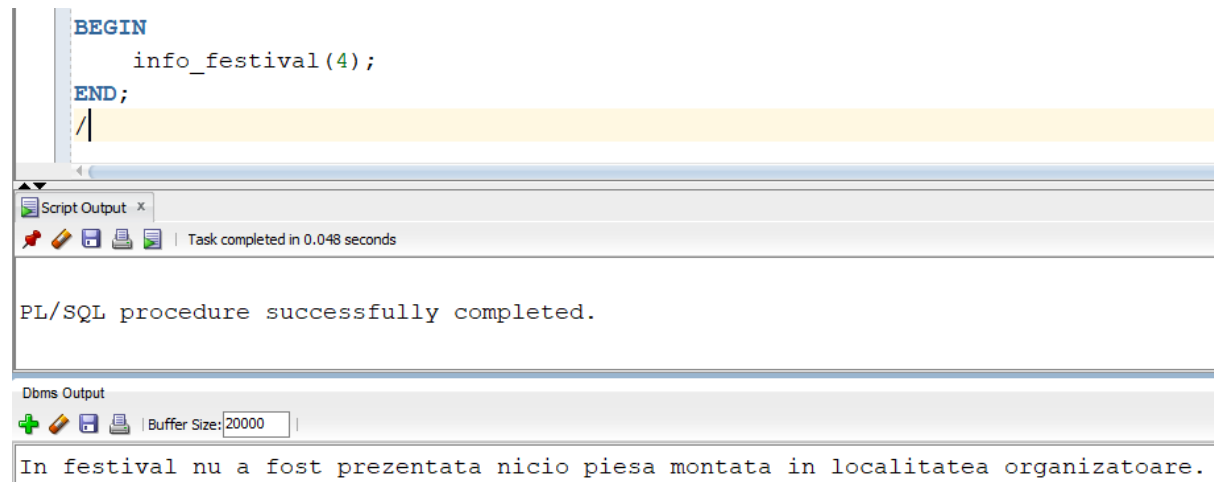
1750 + 1500 = 3250

BEGIN

info\_festival(4);

END;

/

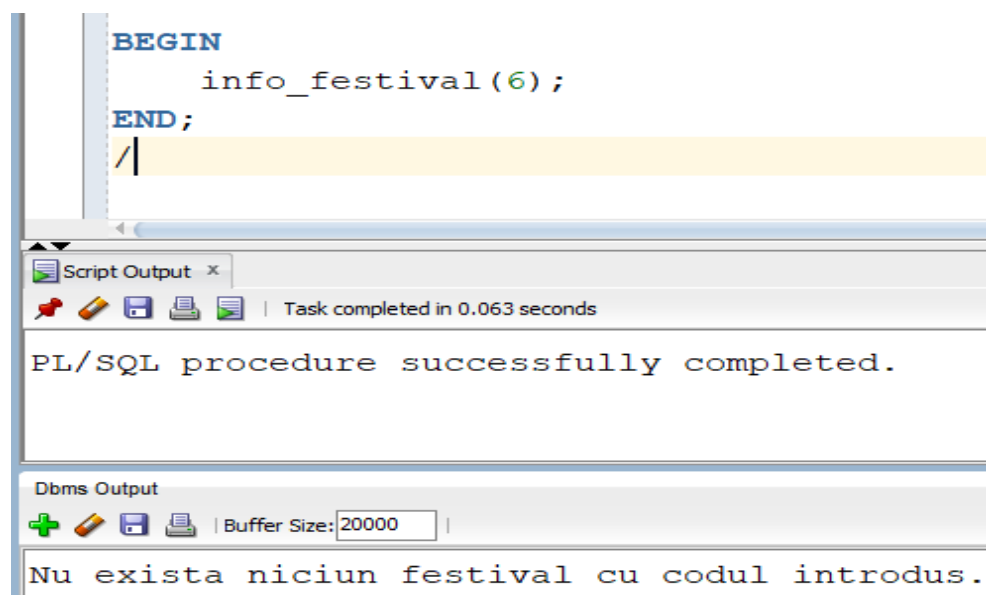


BEGIN

info\_festival(6);

END;

/





## 8.

Pentru o luna din an data de la tastatura (prin numărul corespunzător), si pentru un nume de artist dat, sa se determine numarul de spectacole in care artistul este implicat in luna respectiva, precum si lista acestora, care va contine titlurile pieselor, si salile in care acestea sunt programate.

Se vor trata urmatoarele cazuri:

- luna introdusa se afla in afara stagiunii teatrale (septembrie-iunie)
- luna introdusa are o valoare mai mica decat 1, sau mai mare decat 12
- nu exista niciun artist cu numele si prenumele introduse
- exista mai multi artisti cu numele si prenumele introduse.

```
CREATE OR REPLACE TYPE nume_prenume IS OBJECT (  
    prenume VARCHAR2(25),  
    nume VARCHAR2(25));  
/
```

```
CREATE OR REPLACE TYPE spect_sala IS OBJECT (  
    titlu VARCHAR2(70),  
    s VARCHAR2(50),  
    map MEMBER FUNCTION f RETURN VARCHAR2);  
/
```

```
CREATE OR REPLACE TYPE BODY spect_sala AS  
    MAP MEMBER FUNCTION f RETURN VARCHAR2 IS  
    BEGIN  
        RETURN titlu;  
    END f;  
END;  
/
```

```
CREATE OR REPLACE TYPE lista_spect IS TABLE OF spect_sala;  
/
```

```

CREATE OR REPLACE FUNCTION info_act_luna(np nume_prenume, luna NUMBER)
RETURN lista_spect IS

    np_aux nume_prenume;

    spectacole lista_spect;

    IN_AFARA_STAGIUNII EXCEPTION;

    LUNA_INVALIDA EXCEPTION;

BEGIN

    SELECT nume_prenume(nume, prenume) INTO np_aux
    FROM persoana p, artist a
    WHERE p.cod_persoana = a.cod_persoana
    AND nume = np.nume AND prenume = np.prenume;

    IF luna = 7 OR luna = 8 THEN

        RAISE IN_AFARA_STAGIUNII;

    ELSIF luna < 1 OR luna > 12 THEN

        RAISE LUNA_INVALIDA;

    END IF;

    SELECT DISTINCT spect_sala(titlu, s.denumire) BULK COLLECT INTO spectacole
    FROM montare m, piesa p, afis a, sala s, distributie d, artist art, persoana pers
    WHERE m.cod_piesa = p.cod_piesa
    AND m.cod_montare = a.cod_montare
    AND a.cod_sala = s.cod_sala
    AND d.cod_montare = m.cod_montare
    AND d.cod_artist = art.cod_persoana
    AND art.cod_persoana = pers.cod_persoana
    AND pers.nume = np.nume AND pers.prenume = np.prenume
    AND EXTRACT (MONTH FROM data) = luna;

    RETURN spectacole;

EXCEPTION

    WHEN IN_AFARA_STAGIUNII THEN

        RAISE_APPLICATION_ERROR(-20001, 'Luna introdusa se afla in afara stagiunii
teatrale.');
```

```

    WHEN LUNA_INVALIDA THEN

```

```

        RAISE_APPLICATION_ERROR(-20002, 'Luna introdusa nu exista.');
```

WHEN NO\_DATA\_FOUND THEN

```

        RAISE_APPLICATION_ERROR(-20003, 'Nu exista niciun artist cu numele introdus.');
```

WHEN TOO\_MANY\_ROWS THEN

```

        RAISE_APPLICATION_ERROR(-20004, 'Exista mai multi artisti cu numele ' ||
np.prenume || ' ' || np.numa || ' ');
```

END info\_act\_luna;

/

The screenshot shows a SQL script editor with the following code:

```

CREATE OR REPLACE TYPE nume_prenume IS OBJECT (
    prenume VARCHAR2(25),
    nume VARCHAR2(25));
/
```

Below the editor, the 'Script Output' window displays the message: 'Task completed in 0.077 seconds' and 'Type NUME\_PRENUME compiled'.

The screenshot shows a SQL script editor with the following code:

```

CREATE OR REPLACE TYPE spect_sala IS OBJECT (
    titlu VARCHAR2(70),
    s VARCHAR2(50),
    map MEMBER FUNCTION f RETURN VARCHAR2);
/
```

Below the editor, the 'Script Output' window displays the message: 'Task completed in 0.084 seconds' and 'Type SPECT\_SALA compiled'.

The next screenshot shows the body of the type being compiled:

```

CREATE OR REPLACE TYPE BODY spect_sala AS
    MAP MEMBER FUNCTION f RETURN VARCHAR2 IS
    BEGIN
        RETURN titlu;
    END f;
END;
/
```

Below the editor, the 'Script Output' window displays the message: 'Task completed in 0.084 seconds' and 'Type Body SPECT\_SALA compiled'.

```
CREATE OR REPLACE TYPE lista_spect IS TABLE OF spect_sala;
/
```

Script Output x

Task completed in 0.085 seconds

Type LISTA\_SPECT compiled

```
CREATE OR REPLACE FUNCTION info_act_luna(np nume_prenume, luna NUMBER) RETURN lista_spect IS
    np_aux nume_prenume;
    spectacole lista_spect;
    IN_AFARA_STAGIUNII EXCEPTION;
    LUNA_INVALIDA EXCEPTION;
BEGIN
    SELECT nume_prenume(nume, prenume) INTO np_aux
    FROM persoana p, artist a
    WHERE p.cod_persoana = a.cod_persoana
    AND nume = np.nume AND prenume = np.prenume;
    IF luna = 7 OR luna = 8 THEN
        RAISE IN_AFARA_STAGIUNII;
    ELSIF luna < 1 OR luna > 12 THEN
        RAISE LUNA_INVALIDA;
    END IF;
    SELECT DISTINCT spect_sala(titlu, s.denumire) BULK COLLECT INTO spectacole
    FROM montare m, piesa p, afis a, sala s, distributie d, artist art, persoana pers
    WHERE m.cod_piesa = p.cod_piesa
    AND m.cod_montare = a.cod_montare
    AND a.cod_sala = s.cod_sala
    AND d.cod_montare = m.cod_montare
    AND d.cod_artist = art.cod_persoana
    AND art.cod_persoana = pers.cod_persoana
    AND pers.nume = np.nume AND pers.prenume = np.prenume
    AND EXTRACT (MONTH FROM data) = luna;

```

Script Output x

Task completed in 0.164 seconds

Function INFO\_ACT\_LUNA compiled

```
EXCEPTION
    WHEN IN_AFARA_STAGIUNII THEN
        RAISE_APPLICATION_ERROR(-20001, 'Luna introdusa se afla in afara stagiunii teatrale.');
```

```
    WHEN LUNA_INVALIDA THEN
        RAISE_APPLICATION_ERROR(-20002, 'Luna introdusa nu exista.');
```

```
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista niciun artist cu numele introdus.');
```

```
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20004, 'Exista mai multi artisti cu numele ' || np.prenume || ' ' || np.nume || '.');
```

```
END info_act_luna;
/
```

Script Output x Query Result x

Task completed in 0.078 seconds

Function INFO\_ACT\_LUNA compiled

DECLARE

np.nume\_prenume := nume\_prenume(null, null);

spectacole.lista\_spect := lista\_spect();

BEGIN

np.nume := 'Smirnov';

np.prenume := 'Serghei';

spectacole := info\_act\_luna(np, 3);

IF spectacole.COUNT = 0 THEN

dbms\_output.put\_line('Artistul nu participa la niciun spectacol programat in luna data.');

ELSIF

spectacole.COUNT = 1 THEN

dbms\_output.put\_line('In cursul lunii date, artistul participa la piesa ' || spectacole(1).titlu || ', ce are loc in ' ||

spectacole(1).s || '.');

ELSE

dbms\_output.put\_line('In cursul lunii date, artistul participa la ' || spectacole.COUNT || ' piese. Acestea sunt urmatoarele:');

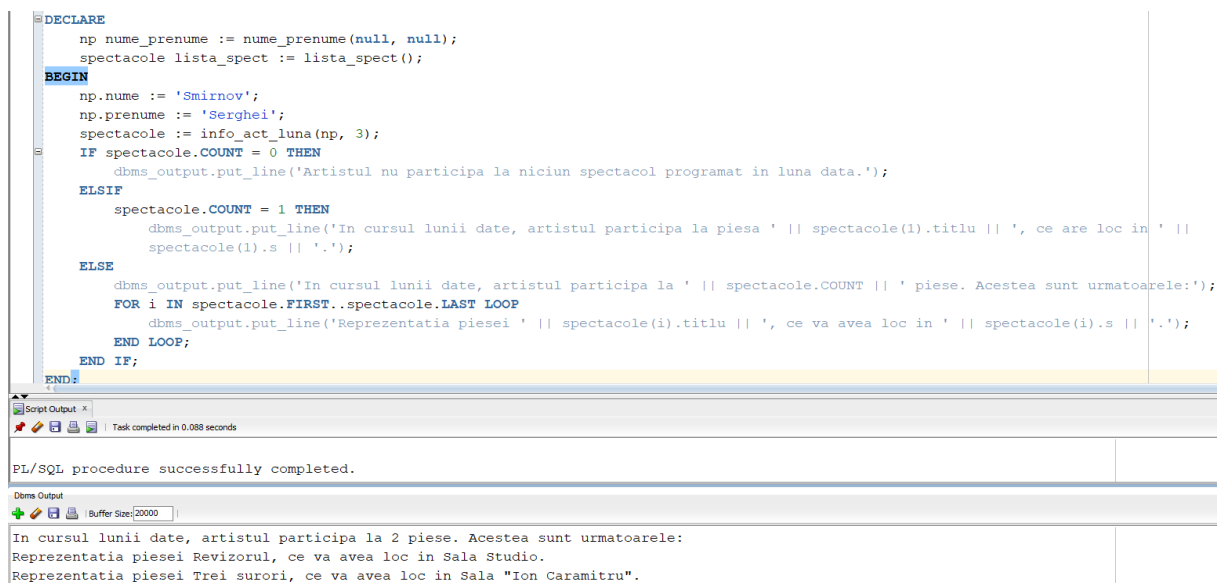
FOR i IN spectacole.FIRST..spectacole.LAST LOOP

dbms\_output.put\_line('Reprezentatia piesei ' || spectacole(i).titlu || ', ce va avea loc in ' || spectacole(i).s || '.');

END LOOP;

END IF;

END; /



```
DECLARE
np.nume_prenume := nume_prenume(null, null);
spectacole.lista_spect := lista_spect();
BEGIN
np.nume := 'Smirnov';
np.prenume := 'Serghei';
spectacole := info_act_luna(np, 3);
IF spectacole.COUNT = 0 THEN
dbms_output.put_line('Artistul nu participa la niciun spectacol programat in luna data.');

ELSIF



spectacole.COUNT = 1 THEN



dbms_output.put_line('In cursul lunii date, artistul participa la piesa ' || spectacole(1).titlu || ', ce are loc in ' || spectacole(1).s || '.');



ELSE



dbms_output.put_line('In cursul lunii date, artistul participa la ' || spectacole.COUNT || ' piese. Acestea sunt urmatoarele:');



FOR i IN spectacole.FIRST..spectacole.LAST LOOP



dbms_output.put_line('Reprezentatia piesei ' || spectacole(i).titlu || ', ce va avea loc in ' || spectacole(i).s || '.');



END LOOP;



END IF;



END;



PL/SQL procedure successfully completed.



Dbms Output



In cursul lunii date, artistul participa la 2 piese. Acestea sunt urmatoarele:  
Reprezentatia piesei Revizorul, ce va avea loc in Sala Studio.  
Reprezentatia piesei Trei surori, ce va avea loc in Sala "Ion Caramitru".


```

DECLARE

np nume\_prenume := nume\_prenume(null, null);

spectacole lista\_spect := lista\_spect();

BEGIN

np.nume := 'Smirnov';

np.prenume := 'Serghei';

spectacole := info\_act\_luna(np, 2);

IF spectacole.COUNT = 0 THEN

dbms\_output.put\_line('Artistul nu participa la niciun spectacol programat in luna data.');

ELSIF

spectacole.COUNT = 1 THEN

dbms\_output.put\_line('In cursul lunii date, artistul participa la piesa ' || spectacole(1).titlu  
|| ', ce are loc in ' ||

spectacole(1).s || '.');

ELSE

dbms\_output.put\_line('In cursul lunii date, artistul participa la ' || spectacole.COUNT || '  
piese. Acestea sunt urmatoarele:');

FOR i IN spectacole.FIRST..spectacole.LAST LOOP

dbms\_output.put\_line('Reprezentatia piesei ' || spectacole(i).titlu || ', ce va avea loc in ' ||  
spectacole(i).s || '.');

END LOOP;

END IF;

END;/



```
DECLARE
np nume_prenume := nume_prenume(null, null);
spectacole lista_spect := lista_spect();
BEGIN
np.nume := 'Smirnov';
np.prenume := 'Serghei';
spectacole := info_act_luna(np, 2);
IF spectacole.COUNT = 0 THEN
dbms_output.put_line('Artistul nu participa la niciun spectacol programat in luna data.');
```

DECLARE

np.nume\_prenume := nume\_prenume(null, null);

spectacole.lista\_spect := lista\_spect();

BEGIN

np.nume := 'Ion';

np.prenume := 'Diana Ruxandra';

spectacole := info\_act\_luna(np, 2);

IF spectacole.COUNT = 0 THEN

dbms\_output.put\_line('Artistul nu participa la niciun spectacol programat in luna data.');

ELSIF

spectacole.COUNT = 1 THEN

dbms\_output.put\_line('In cursul lunii date, artistul participa la piesa "' ||  
spectacole(1).titlu || '"', ce are loc in ' ||

spectacole(1).s || '.');

ELSE

dbms\_output.put\_line('In cursul lunii date, artistul participa la ' || spectacole.COUNT || '  
piese. Acestea sunt urmatoarele:');

FOR i IN spectacole.FIRST..spectacole.LAST LOOP

dbms\_output.put\_line('Reprezentatia piesei ' || spectacole(i).titlu || ', ce va avea loc in ' ||  
spectacole(i).s || '.');

END LOOP;

END IF;

END;/

```
DECLARE
np.nume_prenume := nume_prenume(null, null);
spectacole.lista_spect := lista_spect();
BEGIN
np.nume := 'Ion';
np.prenume := 'Diana Ruxandra';
spectacole := info_act_luna(np, 2);
IF spectacole.COUNT = 0 THEN
dbms_output.put_line('Artistul nu participa la niciun spectacol programat in luna data.');

ELSIF



spectacole.COUNT = 1 THEN



dbms_output.put_line('In cursul lunii date, artistul participa la piesa "' || spectacole(1).titlu || '"', ce are loc in ' || spectacole(1).s || '.');



ELSE



dbms_output.put_line('In cursul lunii date, artistul participa la ' || spectacole.COUNT || ' piese. Acestea sunt urmatoarele:');



FOR i IN spectacole.FIRST..spectacole.LAST LOOP



dbms_output.put_line('Reprezentatia piesei ' || spectacole(i).titlu || ', ce va avea loc in ' || spectacole(i).s || '.');



END LOOP;



END IF;



END;



/


```

Script Output x

Task completed in 0.074 seconds

PL/SQL procedure successfully completed.

Dbms Output

Buffer Size: 20000

In cursul lunii date, artistul participa la piesa "O noapte furtunoasa", ce are loc in Sala "Ion Caramitru".

DECLARE

np nume\_prenume := nume\_prenume(null, null);

spectacole lista\_spect := lista\_spect();

BEGIN

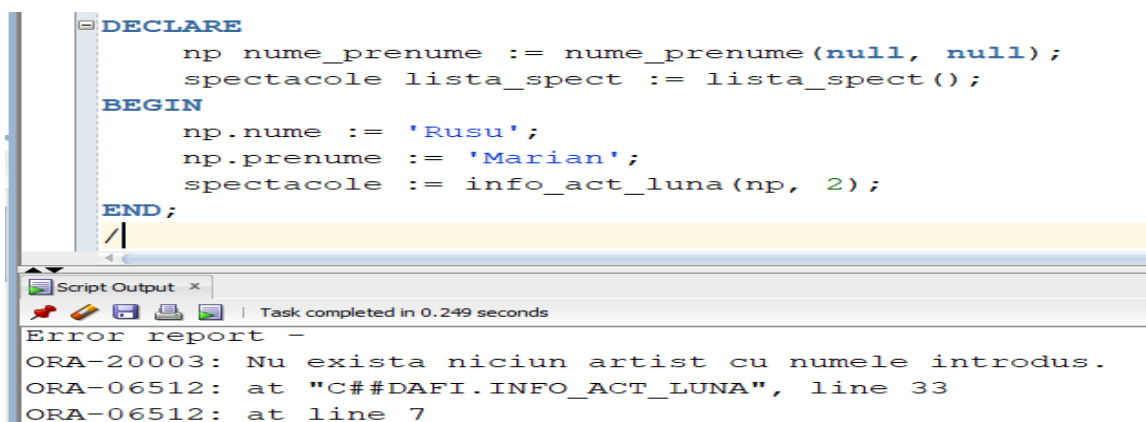
np.nume := 'Rusu';

np.prenume := 'Marian';

spectacole := info\_act\_luna(np, 2);

END;

/



The screenshot shows a SQL script execution window. The script contains the following code:

```
DECLARE
    np nume_prenume := nume_prenume(null, null);
    spectacole lista_spect := lista_spect();
BEGIN
    np.nume := 'Rusu';
    np.prenume := 'Marian';
    spectacole := info_act_luna(np, 2);
END;
```

The script output shows the task completed in 0.249 seconds. Below the output, an error report is displayed:

```
Error report -
ORA-20003: Nu exista niciun artist cu numele introdus.
ORA-06512: at "C##DAFI.INFO_ACT_LUNA", line 33
ORA-06512: at line 7
```

DECLARE

np nume\_prenume := nume\_prenume(null, null);

spectacole lista\_spect := lista\_spect();

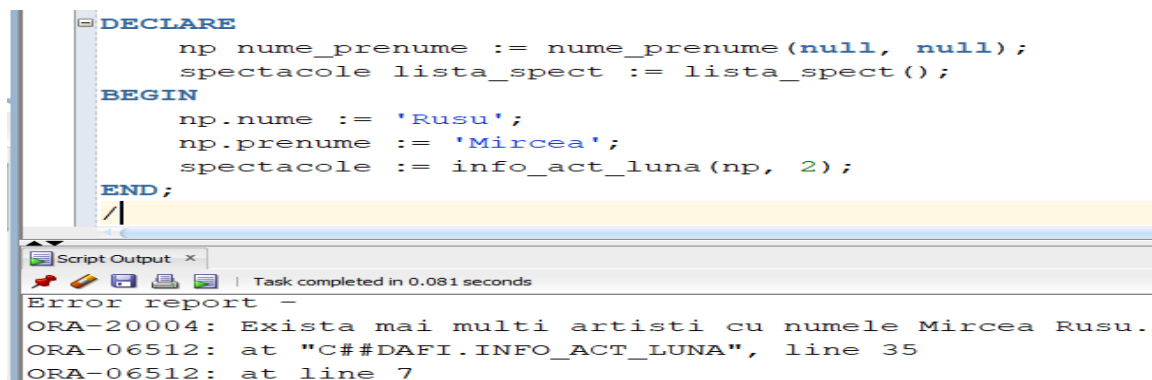
BEGIN

np.nume := 'Rusu';

np.prenume := 'Mircea';

spectacole := info\_act\_luna(np, 2);

END;/



The screenshot shows a SQL script execution window. The script contains the following code:

```
DECLARE
    np nume_prenume := nume_prenume(null, null);
    spectacole lista_spect := lista_spect();
BEGIN
    np.nume := 'Rusu';
    np.prenume := 'Mircea';
    spectacole := info_act_luna(np, 2);
END;
```

The script output shows the task completed in 0.081 seconds. Below the output, an error report is displayed:

```
Error report -
ORA-20004: Exista mai multi artisti cu numele Mircea Rusu.
ORA-06512: at "C##DAFI.INFO_ACT_LUNA", line 35
ORA-06512: at line 7
```



DECLARE

np nume\_prenume := nume\_prenume(null, null);

spectacole lista\_spect := lista\_spect();

BEGIN

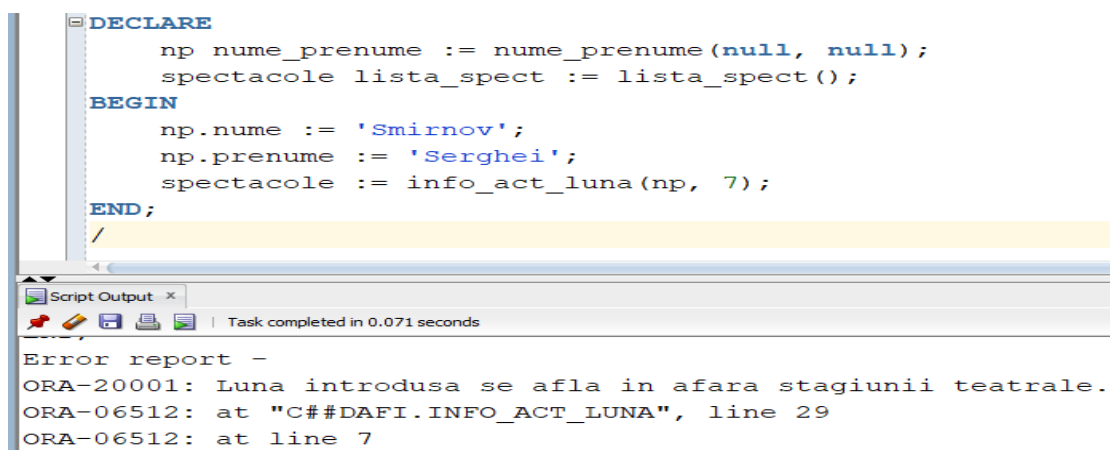
np.nume := 'Smirnov';

np.prenume := 'Serghei';

spectacole := info\_act\_luna(np, 7);

END;

/



The screenshot shows a SQL Developer window with a PL/SQL script. The script declares variables and calls a function. Below the script, the 'Script Output' window shows the execution result, which includes an error message.

```
DECLARE
  np nume_prenume := nume_prenume(null, null);
  spectacole lista_spect := lista_spect();
BEGIN
  np.nume := 'Smirnov';
  np.prenume := 'Serghei';
  spectacole := info_act_luna(np, 7);
END;
```

Script Output x

Task completed in 0.071 seconds

Error report -

ORA-20001: Luna introdusa se afla in afara stagiunii teatrale.

ORA-06512: at "C##DAFI.INFO\_ACT\_LUNA", line 29

ORA-06512: at line 7

DECLARE

np nume\_prenume := nume\_prenume(null, null);

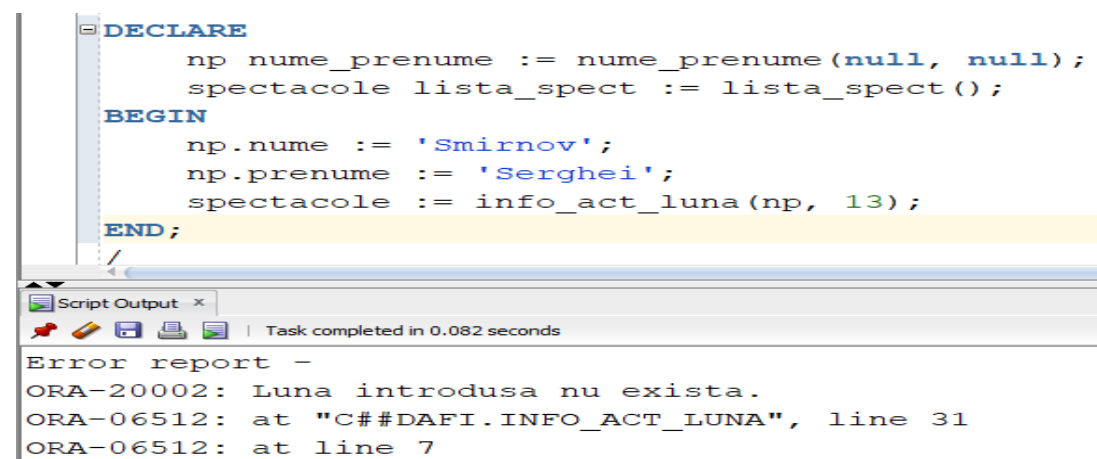
spectacole lista\_spect := lista\_spect();

BEGIN

np.nume := 'Smirnov'; np.prenume := 'Serghei';

spectacole := info\_act\_luna(np, 13);

END;/



The screenshot shows a SQL Developer window with a PL/SQL script. The script declares variables and calls a function. Below the script, the 'Script Output' window shows the execution result, which includes an error message.

```
DECLARE
  np nume_prenume := nume_prenume(null, null);
  spectacole lista_spect := lista_spect();
BEGIN
  np.nume := 'Smirnov';
  np.prenume := 'Serghei';
  spectacole := info_act_luna(np, 13);
END;
```

Script Output x

Task completed in 0.082 seconds

Error report -

ORA-20002: Luna introdusa nu exista.

ORA-06512: at "C##DAFI.INFO\_ACT\_LUNA", line 31

ORA-06512: at line 7

## 9.

Sa se modifice (temporar) structura tabelii piesa, adaugandu-se o coloana "informatii\_suplimentare", care va contine urmatoarele date despre o piesa: bugetul alocat, diferenta dintre bugetul alocat montarii piesei si bugetul mediu alocat unei montari din teatrul respectiv, numarul de montari, numarul de reprezentatii, respectiv numarul de prezente la Festivalul International de Teatru ale piesei, pentru fiecare dintre teatrele in care aceasta a fost pusa in scena. Apoi, sa se implementeze o procedura care populeaza coloana "informatii\_suplimentare", pentru piesele publicate intr-un interval dat ca parametru, si, pentru o piesa din intervalul dat, obtine primele trei pozitii din clasamentul montarilor acesteia (clasamentul va contine denumirea teatrului si diferenta dintre bugetul alocat montarii si bugetul mediu alocat unei montari din teatrul respectiv), montarile fiind ordonate descrescator, in functie de diferenta dintre bugetul alocat montarii piesei date ca parametru si bugetul mediu alocat unei montari, in teatrul respectiv. Se vor trata urmatoarele cazuri: nu exista nicio piesa cu titlul dat, exista mai multe piese cu titlul dat, cei doi ani dati ca parametri nu formeaza un interval valid, piesa cu titlul dat nu a fost publicata in intervalul format de cei doi ani dati ca parametri.

```
CREATE OR REPLACE TYPE piesa_teatru IS OBJECT (
```

```
    t VARCHAR2(50),
```

```
    buget NUMBER(6, 2),
```

```
    dif NUMBER(6, 2),
```

```
    nr_montari NUMBER,
```

```
    nr_reprezentatii NUMBER,
```

```
    nr_participari NUMBER
```

```
);
```

```
/
```

```
CREATE OR REPLACE TYPE tab_pt IS TABLE OF piesa_teatru;
```

```
/
```

```
ALTER TABLE piesa
```

```
ADD (informatii_suplimentare tab_pt) NESTED TABLE informatii_suplimentare STORE AS  
coloana_col;
```

```
CREATE OR REPLACE PROCEDURE situatie_montari (an1 piesa.an_publicare%TYPE, an2  
piesa.an_publicare%TYPE, p piesa.titlu%TYPE) IS
```

```
    cod_p piesa.cod_piesa%TYPE;
```

```
    an piesa.an_publicare%TYPE;
```

```
    t piesa.titlu%TYPE;
```

```
    TYPE v_coduri IS TABLE OF piesa.cod_piesa%TYPE;
```

```

lista_piese_interval v_coduri := v_coduri();
info tab_pt := tab_pt();
nr NUMBER := 1;
dif_ant montare.buget%TYPE := 0;
buget_mediu montare.buget%TYPE;
IN_AFARA_PERIOADEI_DE_INTERES EXCEPTION;
INTERVAL_INVALID EXCEPTION;
BEGIN
    SELECT cod_piesa INTO cod_p
    FROM piesa
    WHERE titlu = p;
    SELECT an_publicare INTO an
    FROM piesa
    WHERE cod_piesa = cod_p;
    IF an1 > an2 THEN
        RAISE INTERVAL_INVALID;
    END IF;
    IF an < an1 OR an > an2 THEN
        RAISE IN_AFARA_PERIOADEI_DE_INTERES;
    END IF;
    SELECT cod_piesa BULK COLLECT INTO lista_piese_interval
    FROM piesa
    WHERE an_publicare BETWEEN an1 AND an2;
    FOR i IN lista_piese_interval.FIRST..lista_piese_interval.LAST LOOP
        SELECT titlu INTO t
        FROM piesa
        WHERE cod_piesa = lista_piese_interval(i);
        dbms_output.put_line('Informatii despre montarile piesei ' || t || ':');
        info.DELETE;
        nr := 1;
        FOR j IN (SELECT t1.denumire ttr, buget,

```

```

        (SELECT SUM(buget) FROM montare m, teatru t WHERE cod_piesa =
lista_piese_interval(i)

        AND m.cod_teatru = t.cod_teatru

        AND t.denumire = t1.denumire)

- (SELECT AVG(buget) FROM montare m, teatru t WHERE m.cod_teatru =
t.cod_teatru

        AND t.denumire = t1.denumire) as dif,

        (SELECT COUNT(m.cod_montare) FROM montare m, teatru t WHERE cod_piesa =
lista_piese_interval(i)

        AND m.cod_teatru =

t.cod_teatru

        AND t.denumire =

t1.denumire) as nr_mont,

        (SELECT COUNT(m.cod_montare) FROM montare m, afis a, teatru t WHERE
cod_piesa = lista_piese_interval(i)

        AND m.cod_montare = a.cod_montare

        AND m.cod_teatru = t.cod_teatru

        AND t.denumire = t1.denumire) as nr_reprez,

        (SELECT COUNT(m.cod_montare) FROM montare m, agenda a, festival f, teatru t
WHERE cod_piesa = lista_piese_interval(i)

        AND m.cod_montare =

a.cod_montare

        AND m.cod_teatru = t.cod_teatru

        AND t.denumire = t1.denumire

        AND a.cod_festival = f.cod_festival

        AND f.denumire = 'Festivalul

International de Teatru') as nr_particip

        FROM montare m, teatru t1

        WHERE m.cod_piesa = lista_piese_interval(i)

        AND m.cod_teatru = t1.cod_teatru

        ORDER BY 3 DESC) LOOP

info.extend;

info(nr) := piesa_teatru(j.ttr, j.buget, j.dif, j.nr_mont, j.nr_reprez, j.nr_particip);

nr := nr + 1;

END LOOP;

```

```

FOR j IN info.FIRST..info.LAST LOOP
    dbms_output.put_line(' La ' || info(j).t || ', unde bugetul alocat a fost de ' || info(j).buget ||
');
    IF info(j).nr_montari = 1 THEN
        dbms_output.put_line(' s-a realizat o montare a piesei, ');
        IF info(j).nr_reprezentatii = 0 THEN
            dbms_output.put_line(' care nu are programate reprezentatii in teatrul unde a fost
realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' si nici participari in festivaluri. ');
            ELSIF info(j).nr_participari = 1 then
                dbms_output.put_line(' dar are programata o participare in festival. ');
            ELSE
                dbms_output.put_line(' dar are programate ' || info(j).nr_participari || ' participari
in festivaluri. ');
            END IF;
        ELSIF info(j).nr_reprezentatii = 1 THEN
            dbms_output.put_line(' care are programata o reprezentatie in teatrul unde a fost
realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' dar nicio participare in festivaluri. ');
            ELSIF info(j).nr_participari = 1 then
                dbms_output.put_line(' si o participare in festival. ');
            ELSE
                dbms_output.put_line(' si programate ' || info(j).nr_participari || ' participari in
festivaluri. ');
            END IF;
        ELSE
            dbms_output.put_line(' care are programate ' || info(j).nr_reprezentatii || '
reprezentatii in teatrul unde a fost realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' dar nicio participare in festivaluri. ');
            ELSIF info(j).nr_participari = 1 then
                dbms_output.put_line(' si o participare in festival. ');

```

```

ELSE
    dbms_output.put_line(' si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

```

END IF;

ELSE
    dbms_output.put_line(' s-au realizat ' || info(j).nr_montari || ' montari ale piesei, ');
    IF info(j).nr_reprezentatii = 0 THEN
        dbms_output.put_line(' care nu au programate reprezentatii in teatrul unde a fost realizata');
```

IF info(j).nr\_participari = 0 THEN

```

        dbms_output.put_line(' si nici participari in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

        dbms_output.put_line(' dar au programata o participare in festival.');
```

ELSE

```

        dbms_output.put_line(' , dar au programate ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

ELSIF info(j).nr\_reprezentatii = 1 THEN

```

        dbms_output.put_line(' au programata o reprezentatie in teatrul unde a fost realizata');
```

IF info(j).nr\_participari = 0 THEN

```

        dbms_output.put_line(' , dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

        dbms_output.put_line(' , si o participare in festival.');
```

ELSE

```

        dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

ELSE

```

        dbms_output.put_line(' au programate ' || info(j).nr_reprezentatii || ' reprezentatii in teatrul unde au fost realizate');
```

IF info(j).nr\_participari = 0 THEN

```

        dbms_output.put_line(' dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

        dbms_output.put_line(' , si o participare in festival.');
```

ELSE

```

        dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

END IF;

END IF;

```

        dbms_output.new_line();
```

END LOOP;

```

dbms_output.new_line();
```

UPDATE piesa

SET informatii\_suplimentare = info

WHERE cod\_piesa = lista\_piese\_interval(i);

END LOOP;

SELECT informatii\_suplimentare INTO info

FROM piesa

WHERE cod\_piesa = cod\_p;

```

        dbms_output.put_line('Primele trei pozitii in clasamentul montarilor piesei ' || p || ' sunt ocupate
de: ');
```

nr := 0;

FOR i IN info.FIRST..info.LAST LOOP

IF nr <= 3 THEN

IF info(i).dif <> dif\_ant THEN

IF nr < 3 THEN

nr := nr + 1;

```

        dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' ||
info(i).buget
```

|| ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu

alocat unei montari

in teatrul respectiv este de ' || info(i).dif || '.');

END IF;

ELSE

```

        dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' ||
info(i).buget
```

|| ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari

in teatrul respectiv este de ' || info(i).dif || '');

END IF;

END IF;

dif\_ant := info(i).dif;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista nicio piesa cu titlul dat.');

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multe piese cu titlul dat.');

WHEN INTERVAL\_INVALID THEN

dbms\_output.put\_line('Cei doi ani introdusi nu formeaza un interval valid.');

WHEN IN\_AFARA\_PERIOADEI\_DE\_INTERES THEN

dbms\_output.put\_line('Piesa introdusa a fost publicata in afara perioadei alese.');

WHEN OTHERS THEN

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

END situatie\_montari; /

```
CREATE OR REPLACE PROCEDURE situatie_montari (an1 piesa.an_publicare%TYPE, an2 piesa.an_publicare%TYPE, p piesa.titlu%TYPE) IS
    cod_p piesa.cod_piesa%TYPE;
    an piesa.an_publicare%TYPE;
    t piesa.titlu%TYPE;
    TYPE v_coduri IS TABLE OF piesa.cod_piesa%TYPE;
    lista_piese_interval v_coduri := v_coduri();
    info tab_pt := tab_pt();
    nr NUMBER := 1;
    dif_ant montare.buget%TYPE := 0;
    buget_mediu montare.buget%TYPE;
    IN_AFARA_PERIOADEI_DE_INTERES EXCEPTION;
    INTERVAL_INVALID EXCEPTION;
BEGIN
    SELECT cod_piesa INTO cod_p
    FROM piesa
    WHERE titlu = p;
    SELECT an_publicare INTO an
    FROM piesa
    WHERE cod_piesa = cod_p;
    IF an1 > an2 THEN
        RAISE INTERVAL_INVALID;
    END IF;
    IF an < an1 OR an > an2 THEN
        RAISE IN_AFARA_PERIOADEI_DE_INTERES;
    END IF;
    SELECT cod_piesa BULK COLLECT INTO lista_piese_interval
    FROM piesa
    WHERE an_publicare BETWEEN an1 AND an2;
    FOR i IN lista_piese_interval.FIRST..lista_piese_interval.LAST LOOP
        SELECT titlu INTO t
```

Script Output x  
Task completed in 0.115 seconds

Procedure SITUATIE\_MONTARI compiled



```

SELECT titlu INTO t
FROM piesa
WHERE cod_piesa = lista_piese_interval(i);
dbms_output.put_line('Informatii despre montarile piesei ' || t || ':');
info.DELETE;
nr := 1;
FOR j IN (SELECT t1.denumire ttr, buget,
              (SELECT SUM(buget) FROM montare m, teatru t WHERE cod_piesa = lista_piese_interval(i)
               AND m.cod_teatru = t.cod_teatru
               AND t.denumire = t1.denumire)
            - (SELECT AVG(buget) FROM montare m, teatru t WHERE m.cod_teatru = t.cod_teatru
               AND t.denumire = t1.denumire) AS dif,
              (SELECT COUNT(m.cod_montare) FROM montare m, teatru t WHERE cod_piesa = lista_piese_interval(i)
               AND m.cod_teatru = t.cod_teatru
               AND t.denumire = t1.denumire) AS nr_mont,
              (SELECT COUNT(m.cod_montare) FROM montare m, afis a, teatru t WHERE cod_piesa = lista_piese_interval(i)
               AND m.cod_montare = a.cod_montare
               AND m.cod_teatru = t.cod_teatru
               AND t.denumire = t1.denumire) AS nr_reprez,
              (SELECT COUNT(m.cod_montare) FROM montare m, agenda a, festival f, teatru t WHERE cod_piesa = lista_piese_interval(i)
               AND m.cod_montare = a.cod_montare
               AND m.cod_teatru = t.cod_teatru
               AND t.denumire = t1.denumire
               AND a.cod_festival = f.cod_festival
               AND f.denumire = 'Festivalul International de Teatru') AS nr_particip
            FROM montare m, teatru t1
            WHERE m.cod_piesa = lista_piese_interval(i)
            AND m.cod_teatru = t1.cod_teatru
            ORDER BY 3 DESC) LOOP
    info.extend;
    info(nr) := piesa_teatru(j.ttr, j.buget, j.dif, j.nr_mont, j.nr_reprez, j.nr_particip);

```

Script Output x  
Task completed in 0.115 seconds

Procedure SITUATIE\_MONTARI compiled

```

info(nr) := piesa_teatru(j.ttr, j.buget, j.dif, j.nr_mont, j.nr_reprez, j.nr_particip);
nr := nr + 1;
END LOOP;
FOR j IN info.FIRST..info.LAST LOOP
    dbms_output.put_line(' La ' || info(j).t || ', unde bugetul alocat a fost de ' || info(j).buget || ',');
    IF info(j).nr_montari = 1 THEN
        dbms_output.put_line(' s-a realizat o montare a piesei, ');
        IF info(j).nr_reprezentatii = 0 THEN
            dbms_output.put_line(' care nu are programate reprezentatii in teatrul unde a fost realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' si nici participari in festivaluri. ');
            ELSIF info(j).nr_participari = 1 THEN
                dbms_output.put_line(' dar are programata o participare in festival. ');
            ELSE
                dbms_output.put_line(' dar are programate ' || info(j).nr_participari || ' participari in festivaluri. ');
            END IF;
        ELSIF info(j).nr_reprezentatii = 1 THEN
            dbms_output.put_line(' care are programata o reprezentatie in teatrul unde a fost realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' dar nicio participare in festivaluri. ');
            ELSIF info(j).nr_participari = 1 THEN
                dbms_output.put_line(' si o participare in festival. ');
            ELSE
                dbms_output.put_line(' si programate ' || info(j).nr_participari || ' participari in festivaluri. ');
            END IF;
        ELSE
            dbms_output.put_line(' care are programate ' || info(j).nr_reprezentatii || ' reprezentatii in teatrul unde a fost realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' dar nicio participare in festivaluri. ');
            ELSIF info(j).nr_participari = 1 THEN
                dbms_output.put_line(' si o participare in festival. ');
            ELSE
                dbms_output.put_line(' si programate ' || info(j).nr_participari || ' participari in festivaluri. ');
            END IF;
        END IF;
    END IF;
END LOOP;

```

Script Output x  
Task completed in 0.115 seconds

Procedure SITUATIE\_MONTARI compiled

```

        dbms_output.put_line(' si o participare in festival.');
```

```

    ELSE
        dbms_output.put_line(' si ' || info(j).nr_participari || ' participari in festivaluri.');
```

```

    END IF;
END IF;
ELSE
    dbms_output.put_line(' s-au realizat ' || info(j).nr_montari || ' montari ale piesei, ');
    IF info(j).nr_reprezentatii = 0 THEN
        dbms_output.put_line(' care nu au programate reprezentatii in teatrul unde a fost realizata');
```

```

        IF info(j).nr_participari = 0 THEN
            dbms_output.put_line(' si nici participari in festivaluri.');
```

```

        ELSIF info(j).nr_participari = 1 then
            dbms_output.put_line(' dar au programata o participare in festival.');
```

```

        ELSE
            dbms_output.put_line(' , dar au programate ' || info(j).nr_participari || ' participari in festivaluri.');
```

```

        END IF;
    ELSIF info(j).nr_reprezentatii = 1 THEN
        dbms_output.put_line(' au programata o reprezentatie in teatrul unde a fost realizata');
```

```

        IF info(j).nr_participari = 0 THEN
            dbms_output.put_line(' , dar nicio participare in festivaluri.');
```

```

        ELSIF info(j).nr_participari = 1 then
            dbms_output.put_line(' , si o participare in festival.');
```

```

        ELSE
            dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

```

        END IF;
    ELSE
        dbms_output.put_line(' au programate ' || info(j).nr_reprezentatii || ' reprezentatii in teatrul unde au fost realizate');
```

```

        IF info(j).nr_participari = 0 THEN
            dbms_output.put_line(' dar nicio participare in festivaluri.');
```

```

        ELSIF info(j).nr_participari = 1 then
            dbms_output.put_line(' , si o participare in festival.');
```

```

        ELSE
            dbms_output.put_line(' , si o participare in festival.');
```

Script Output x  
Task completed in 0.115 seconds

Procedure SITUATIE\_MONTARI compiled

```

        dbms_output.put_line(' , si o participare in festival.');
```

```

    ELSE
        dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

```

    END IF;
END IF;
END IF;
dbms_output.new_line();
END LOOP;
dbms_output.new_line();
UPDATE piesa
SET informatii_suplimentare = info
WHERE cod_piesa = lista_piese_interval(i);
END LOOP;
SELECT informatii_suplimentare INTO info
FROM piesa
WHERE cod_piesa = cod_p;
dbms_output.put_line('Primele trei pozitii in clasamentul montarilor piesei ' || p || ' sunt ocupate de ');
nr := 0;
FOR i IN info.FIRST..info.LAST LOOP
    IF nr <= 3 THEN
        IF info(i).dif <> dif_ant THEN
            IF nr < 3 THEN
                nr := nr + 1;
                dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' || info(i).buget
                    || ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari
                    in teatrul respectiv este de ' || info(i).dif || '.');
```

```

            END IF;
        ELSE
            dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' || info(i).buget
                || ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari
                in teatrul respectiv este de ' || info(i).dif || '.');
```

Script Output x  
Task completed in 0.115 seconds

Procedure SITUATIE\_MONTARI compiled

```

    dif_ant := info(i).dif;
END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        dbms_output.put_line('Nu exista nicio piesa cu titlul dat.');
```

```

    WHEN TOO_MANY_ROWS THEN
        dbms_output.put_line('Exista mai multe piese cu titlul dat.');
```

```

    WHEN INTERVAL_INVALID THEN
        dbms_output.put_line('Cei doi ani introdusi nu formeaza un interval valid.');
```

```

    WHEN IN_AFARA_PERIOADEI_DE_INTERES THEN
        dbms_output.put_line('Piesa introdusa a fost publicata in afara perioadei alese.');
```

```

    WHEN OTHERS THEN
        dbms_output.put_line('Alta eroare.');
```

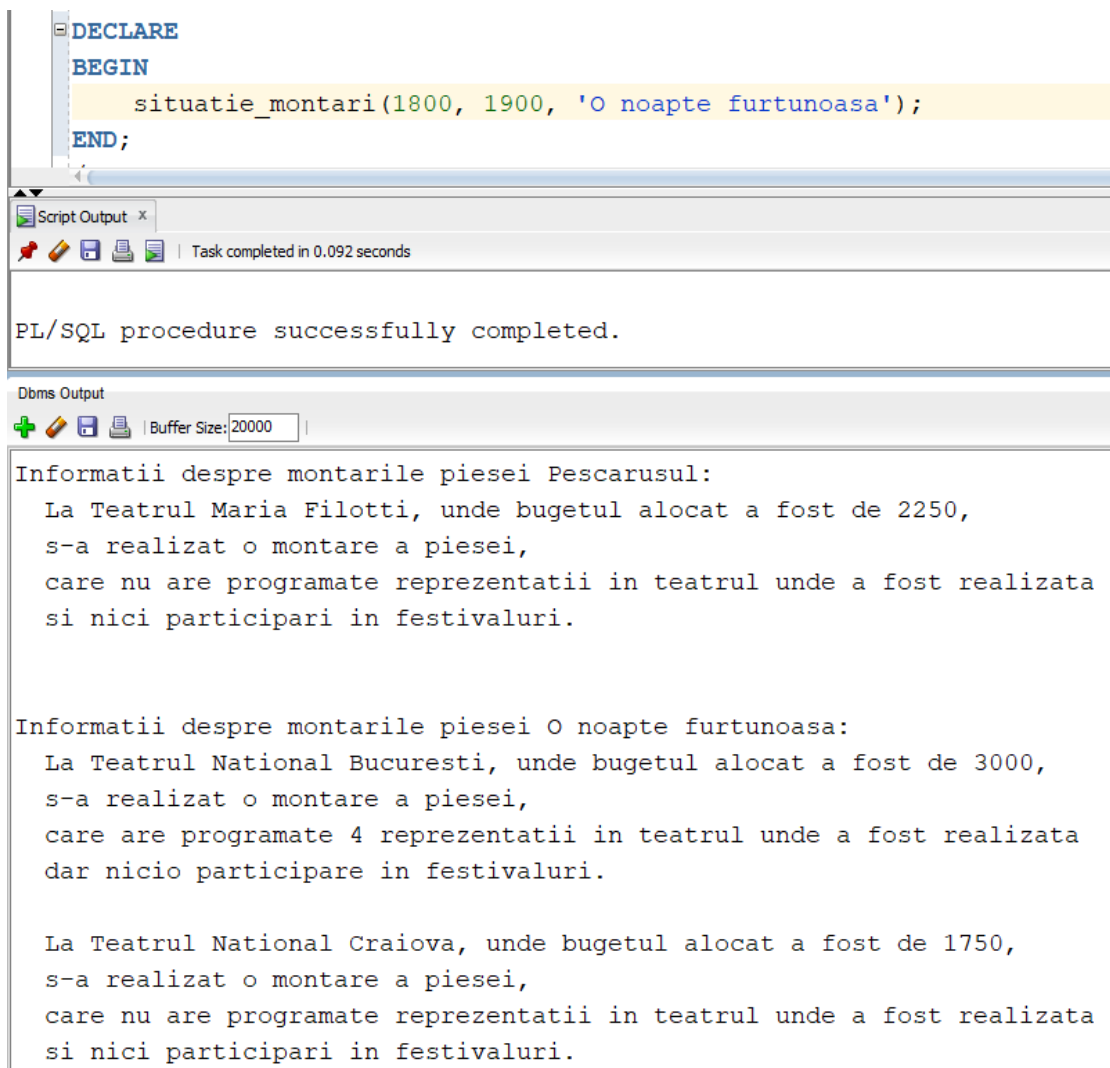
```

END situatie_montari;
/
```

Script Output x  
Task completed in 0.084 seconds

Procedure SITUATIE\_MONTARI compiled

```
DECLARE
BEGIN
    situatie_montari(1800, 1900, 'O noapte furtunoasa');
END;
/
```



The screenshot displays a database IDE interface. At the top, a script editor shows the following PL/SQL code:

```
DECLARE
BEGIN
    situatie_montari(1800, 1900, 'O noapte furtunoasa');
END;
```

Below the script editor, a 'Script Output' window indicates that the task was completed successfully in 0.092 seconds. The output text reads: 'PL/SQL procedure successfully completed.'

At the bottom, a 'Dbms Output' window shows the results of the procedure. It contains three paragraphs of text:

Informatii despre montarile piesei Pescarusul:  
La Teatrul Maria Filotti, unde bugetul alocat a fost de 2250,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

Informatii despre montarile piesei O noapte furtunoasa:  
La Teatrul National Bucuresti, unde bugetul alocat a fost de 3000,  
s-a realizat o montare a piesei,  
care are programate 4 reprezentatii in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

La Teatrul National Craiova, unde bugetul alocat a fost de 1750,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

La Teatrul Notarra, unde bugetul alocat a fost de 1500,  
s-a realizat o montare a piesei,  
care are programate 2 reprezentatii in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

La Teatrul National Vasile Alecsandri, unde bugetul alocat a fost de 1750,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

Informatii despre montarile piesei Trei surori:

La Teatrul National Mihai Eminescu, unde bugetul alocat a fost de 1700,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

La Teatrul National Bucuresti, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

Informatii despre montarile piesei Revizorul:

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata

Informatii despre montarile piesei Revizorul:

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

La Teatrul National Mihai Eminescu, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

La Teatrul National Bucuresti, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

Informatii despre montarile piesei O scrisoare pierduta:

La Teatrul National Mihai Eminescu, unde bugetul alocat a fost de 2000,  
s-a realizat o montare a piesei,  
care nu are programate reprezentatii in teatrul unde a fost realizata  
si nici participari in festivaluri.

La Teatrul de Comedie, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programate 2 reprezentatii in teatrul unde a fost realizata  
dar nicio participare in festivaluri.

La Teatrul National Bucuresti, unde bugetul alocat a fost de 1000,  
s-a realizat o montare a piesei,  
care are programata o reprezentatie in teatrul unde a fost realizata

La Teatrul Bulandra, unde bugetul alocat a fost de 1000, s-a realizat o montare a piesei, care are programata o reprezentatie in teatrul unde a fost realizata dar nicio participare in festivaluri.

Informatii despre montarile piesei D-ale carnavalului:

La Teatrul National Bucuresti, unde bugetul alocat a fost de 3000, s-a realizat o montare a piesei, care are programata o reprezentatie in teatrul unde a fost realizata si o participare in festival.

La Teatrul Radu Stanca, unde bugetul alocat a fost de 1000, s-a realizat o montare a piesei, care are programate 3 reprezentatii in teatrul unde a fost realizata si o participare in festival.

La Teatrul Toma Caragiu, unde bugetul alocat a fost de 1000, s-a realizat o montare a piesei, care nu are programate reprezentatii in teatrul unde a fost realizata si nici participari in festivaluri.

Informatii despre montarile piesei Gaitetele:

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 1750, s-au realizat 2 montari ale piesei, care nu au programate reprezentatii in teatrul unde a fost realizata si nici participari in festivaluri.

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 2250, s-au realizat 2 montari ale piesei,

La Teatrul National Lucian Blaga, unde bugetul alocat a fost de 2250, s-au realizat 2 montari ale piesei, care nu au programate reprezentatii in teatrul unde a fost realizata si nici participari in festivaluri.

La Teatrul Maria Filotti, unde bugetul alocat a fost de 1750, s-a realizat o montare a piesei, care nu are programate reprezentatii in teatrul unde a fost realizata si nici participari in festivaluri.

La Teatrul National Bucuresti, unde bugetul alocat a fost de 1000, s-a realizat o montare a piesei, care are programate 2 reprezentatii in teatrul unde a fost realizata dar nicio participare in festivaluri.

La Teatrul Bulandra, unde bugetul alocat a fost de 1000, s-a realizat o montare a piesei, care nu are programate reprezentatii in teatrul unde a fost realizata si nici participari in festivaluri.

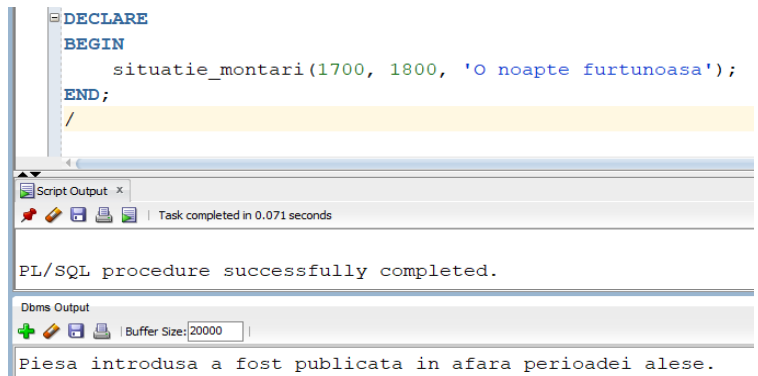
Primele trei pozitii in clasamentul montarilor piesei O noapte furtunoasa sunt ocupate de:

1. Teatrul National Bucuresti, unde bugetul alocat a fost de 3000, iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari in teatrul respectiv este de 1409.09.
2. Teatrul National Craiova, unde bugetul alocat a fost de 1750, iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari in teatrul respectiv este de 375.
3. Teatrul Notarra, unde bugetul alocat a fost de 1500, iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari in teatrul respectiv este de 0.
3. Teatrul National Vasile Alecsandri, unde bugetul alocat a fost de 1750, iar diferenta dintre bugetul alocat acestei montari si bugetul mediu alocat unei montari in teatrul respectiv este de 0.

```

DECLARE
BEGIN
    situatie_montari(1700, 1800, 'O noapte furtunoasa');
END;
/

```



```

DECLARE
BEGIN
    situatie_montari(1700, 1800, 'O noapte furtunoasa');
END;
/

```

Script Output x  
Task completed in 0.071 seconds

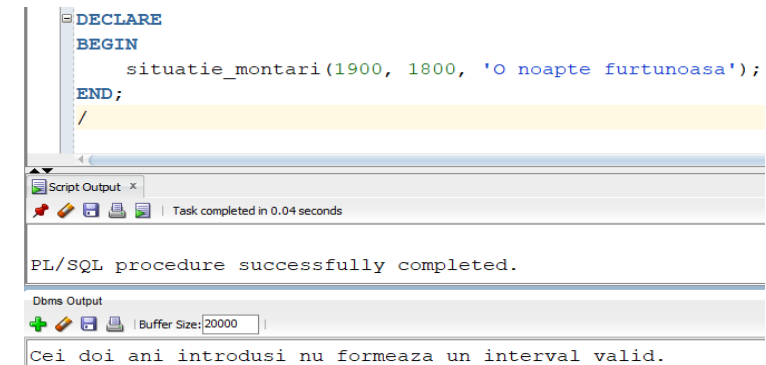
PL/SQL procedure successfully completed.

Dbms Output  
Buffer Size: 20000  
Piesa introdusa a fost publicata in afara perioadei alese.

```

DECLARE
BEGIN
    situatie_montari(1900, 1800, 'O noapte furtunoasa');
END;
/

```



```

DECLARE
BEGIN
    situatie_montari(1900, 1800, 'O noapte furtunoasa');
END;
/

```

Script Output x  
Task completed in 0.04 seconds

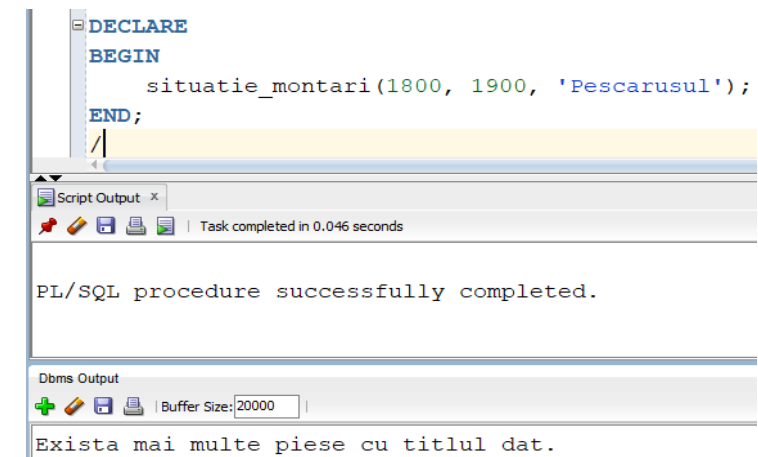
PL/SQL procedure successfully completed.

Dbms Output  
Buffer Size: 20000  
Cei doi ani introdusi nu formeaza un interval valid.

```

DECLARE
BEGIN
    situatie_montari(1800, 1900, 'Pescarusul');
END;
/

```



```

DECLARE
BEGIN
    situatie_montari(1800, 1900, 'Pescarusul');
END;
/

```

Script Output x  
Task completed in 0.046 seconds

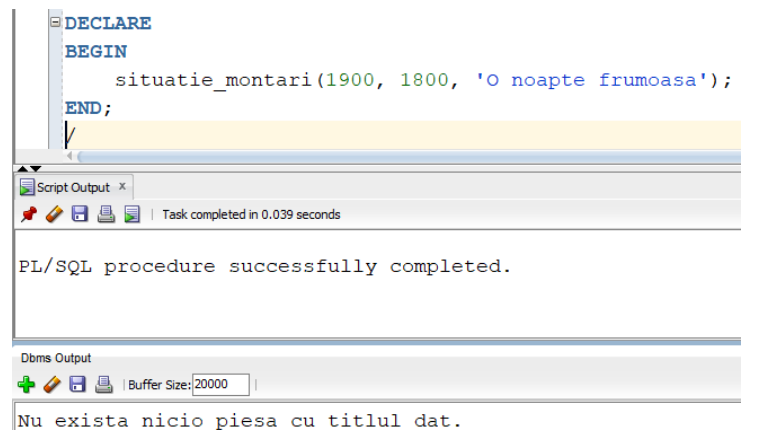
PL/SQL procedure successfully completed.

Dbms Output  
Buffer Size: 20000  
Exista mai multe piese cu titlul dat.

```

DECLARE
BEGIN
    situatie_montari(1900, 1800, 'O noapte frumoasa');
END;
/

```



```

DECLARE
BEGIN
    situatie_montari(1900, 1800, 'O noapte frumoasa');
END;
/

```

Script Output x  
Task completed in 0.039 seconds

PL/SQL procedure successfully completed.

Dbms Output  
Buffer Size: 20000  
Nu exista nicio piesa cu titlul dat.

```

ALTER TABLE piesa
DROP COLUMN informatii_suplimentare;

```

**10.** Sa se defineasca un trigger LMD la nivel de comanda, care sa nu permita efectuarea de inserari in tabela TEHNICIAN, in situatia in care in luna curenta au fost angajati deja un numar de tehnicieni mai mare sau egal decat jumatate din numarul reprezentatiilor programate in luna curenta, si nici efectuarea de stingeri din tabela, daca numarul total al tehnicienilor este mai mic decat jumatate din numarul reprezentatiilor programate in luna curenta.

```
CREATE OR REPLACE TRIGGER conditii_angajare
```

```
    BEFORE INSERT OR DELETE ON tehnician
```

```
DECLARE
```

```
    nr_ang_total NUMBER;
```

```
    nr_ang_luna NUMBER;
```

```
    nr_spect_luna NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(cod_persoana) INTO nr_ang_total
```

```
    FROM tehnician;
```

```
    SELECT COUNT(cod_persoana) INTO nr_ang_luna
```

```
    FROM tehnician
```

```
    WHERE EXTRACT (MONTH FROM data_angajarii) = EXTRACT (MONTH FROM sysdate)
```

```
    AND EXTRACT (YEAR FROM data_angajarii) = EXTRACT (YEAR FROM sysdate);
```

```
    SELECT COUNT(cod_montare) INTO nr_spect_luna
```

```
    FROM afis
```

```
    WHERE EXTRACT (MONTH FROM data) = EXTRACT (MONTH FROM sysdate)
```

```
    AND EXTRACT (YEAR FROM data) = EXTRACT (YEAR FROM sysdate);
```

```
    IF nr_ang_luna >= nr_spect_luna / 2 THEN
```

```
        IF INSERTING THEN
```

```
            RAISE_APPLICATION_ERROR(-20001, 'Au fost angajati deja suficienti tehnicieni, pentru spectacolele din luna curenta.');
```

```
        END IF;
```

```
    END IF;
```

```
    IF nr_ang_total < nr_spect_luna / 2 THEN
```

```
        IF DELETING THEN
```

```
            RAISE_APPLICATION_ERROR(-20002, 'Nu se pot efectua concedieri, deoarece deja exista un deficit de personal tehnic.');
```

```
        END IF;
```

END IF; END;

/

```
CREATE OR REPLACE TRIGGER conditii_angajare
BEFORE INSERT OR DELETE ON tehnician
DECLARE
    nr_ang_total NUMBER;
    nr_ang_luna NUMBER;
    nr_spect_luna NUMBER;
BEGIN
    SELECT COUNT(cod_persoana) INTO nr_ang_total
    FROM tehnician;
    SELECT COUNT(cod_persoana) INTO nr_ang_luna
    FROM tehnician
    WHERE EXTRACT (MONTH FROM data_angajarii) = EXTRACT (MONTH FROM sysdate)
    AND EXTRACT (YEAR FROM data_angajarii) = EXTRACT (YEAR FROM sysdate);
    SELECT COUNT(cod_montare) INTO nr_spect_luna
    FROM afis
    WHERE EXTRACT (MONTH FROM data) = EXTRACT (MONTH FROM sysdate)
    AND EXTRACT (YEAR FROM data) = EXTRACT (YEAR FROM sysdate);
    IF nr_ang_luna >= nr_spect_luna / 2 THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20001, 'Au fost angajati deja suficienti tehnicieni, pentru spectacolele din luna curenta.');
        END IF;
    END IF;
    IF nr_ang_total < nr_spect_luna / 2 THEN
        IF DELETING THEN
            RAISE_APPLICATION_ERROR(-20002, 'Nu se pot efectua concedieri, deoarece deja exista un deficit de personal tehnic.');
        END IF;
    END IF;
END;
```

Script Output x Query Result x  
Task completed in 0.069 seconds  
Trigger CONDITII\_ANGAJARE compiled

DELETE FROM tehnician WHERE cod\_persoana BETWEEN 47 AND 62;

```
DELETE FROM tehnician WHERE cod_persoana BETWEEN 47 AND 62;
```

Query Result x Script Output x  
Task completed in 0.089 seconds  
16 rows deleted.

DELETE FROM tehnician WHERE cod\_persoana = 6;

```
DELETE FROM tehnician WHERE cod_persoana = 6;
```

Query Result x Script Output x  
Task completed in 0.08 seconds  
Error starting at line : 567 in command -  
DELETE FROM tehnician WHERE cod\_persoana = 6  
Error report -  
ORA-20002: Nu se pot efectua concedieri, deoarece deja exista un deficit de personal tehnic.  
ORA-06512: at "C##DAFI.CONDITII\_ANGAJARE", line 23  
ORA-04088: error during execution of trigger 'C##DAFI.CONDITII\_ANGAJARE'

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (47, 2);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (48, 2);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (49, 2);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (50, 3);

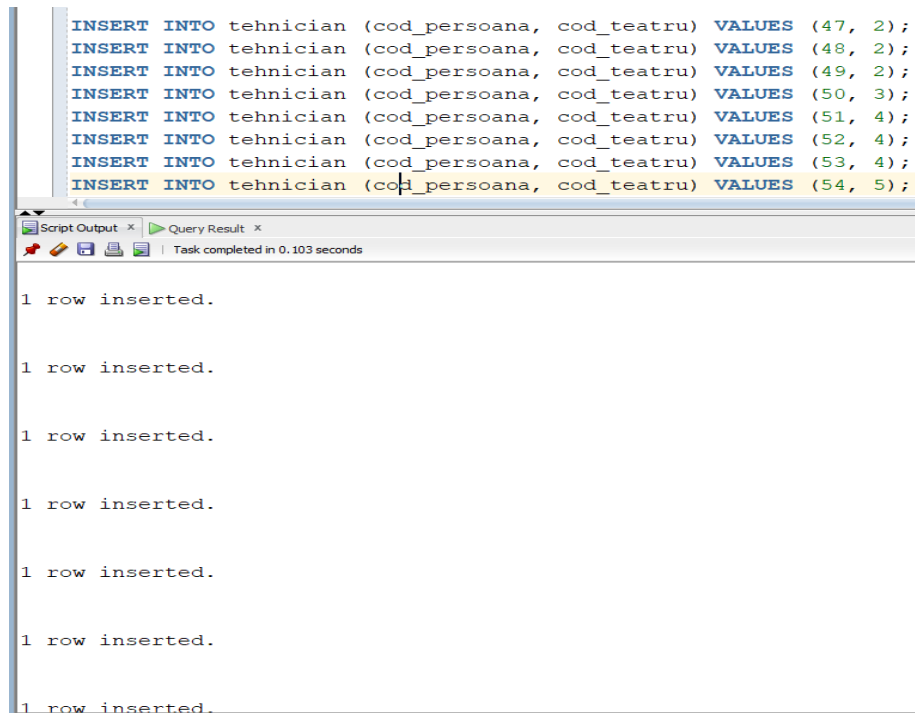


INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (51, 4);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (52, 4);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (53, 4);

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (54, 5);

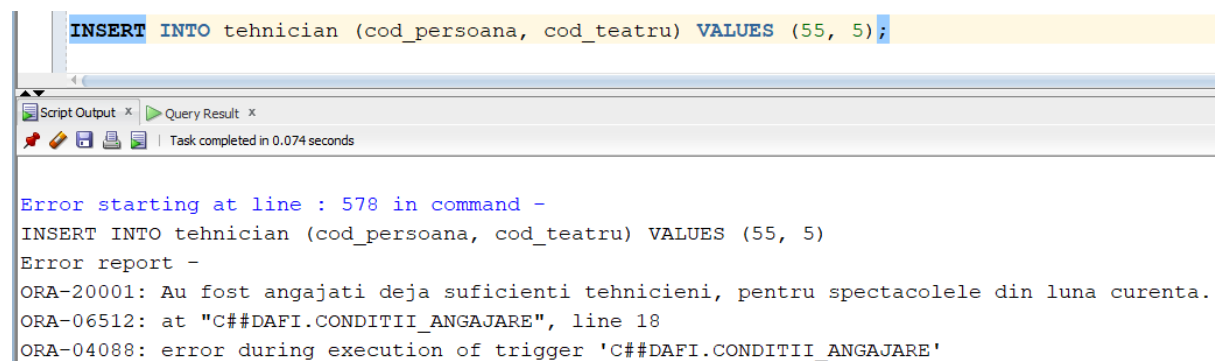


```
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (47, 2);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (48, 2);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (49, 2);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (50, 3);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (51, 4);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (52, 4);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (53, 4);
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (54, 5);
```

Script Output x Query Result x  
Task completed in 0.103 seconds

1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.  
1 row inserted.

INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (55, 5);

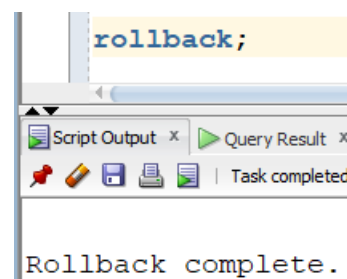


```
INSERT INTO tehnician (cod_persoana, cod_teatru) VALUES (55, 5);
```

Script Output x Query Result x  
Task completed in 0.074 seconds

Error starting at line : 578 in command -  
INSERT INTO tehnician (cod\_persoana, cod\_teatru) VALUES (55, 5)  
Error report -  
ORA-20001: Au fost angajati deja suficienti tehnicieni, pentru spectacolele din luna curenta.  
ORA-06512: at "C##DAFI.CONDITII\_ANGAJARE", line 18  
ORA-04088: error during execution of trigger 'C##DAFI.CONDITII\_ANGAJARE'

rollback;



```
rollback;
```

Script Output x Query Result x  
Task completed

Rollback complete.

11. Sa se defineasca un trigger LMD la nivel de linie, care sa nu permita modificarea datelor de desfasurare a reprezentatiilor din festivaluri, daca noua data nu se afla in intervalul de desfasurare al festivalului respectiv, si stergerea de inregistrari de pe agenda festivalurilor, daca numarul de montari care se afla deja pe agenda festivalului si apartin aceluiasi gen dramatic ca cel al montarii care se doreste a fi inserata, este mai mic decat 30% din totalul montarilor de pe agenda festivalului.

```
CREATE OR REPLACE TRIGGER conditii_agenda_festival
```

```
    BEFORE UPDATE OF data OR DELETE ON agenda
```

```
    FOR EACH ROW
```

```
DECLARE
```

```
    pragma autonomous_transaction;
```

```
    data_inc festival.data_inceperii%TYPE;
```

```
    data_fin festival.data_incheierii%TYPE;
```

```
    gen_p piesa.gen%TYPE;
```

```
    nr_montari_gen NUMBER;
```

```
    nr_montari_fest NUMBER;
```

```
BEGIN
```

```
    IF UPDATING THEN
```

```
        SELECT data_inceperii INTO data_inc
```

```
        FROM festival
```

```
        WHERE cod_festival = :OLD.cod_festival;
```

```
        SELECT data_incheierii INTO data_fin
```

```
        FROM festival
```

```
        WHERE cod_festival = :OLD.cod_festival;
```

```
        IF (:NEW.data < data_inc) OR (:NEW.data > data_fin) THEN
```

```
            RAISE_APPLICATION_ERROR(-20001, 'Nu se poate modifica data reprezentatiei la  
valoarea ' || :NEW.data || ', deoarece aceasta data se afla ' || ' in afara intervalului de desfasurare  
al unui festival la care montarea participa.');
```

```
        END IF;
```

```
    ELSE
```

```
        SELECT gen INTO gen_p
```

```
        FROM montare m, piesa p
```

```
        WHERE m.cod_montare = :OLD.cod_montare
```

```

AND m.cod_piesa = p.cod_piesa;

SELECT COUNT(m.cod_montare) INTO nr_montari_gen

FROM agenda a, montare m, piesa p

WHERE a.cod_festival = :OLD.cod_festival

AND a.cod_montare = m.cod_montare

AND m.cod_piesa = p.cod_piesa

AND p.gen = gen_p;

SELECT COUNT(cod_montare) INTO nr_montari_fest

FROM agenda

WHERE cod_festival = :OLD.cod_festival;

IF nr_montari_gen <= 0.3 * nr_montari_fest THEN

    RAISE_APPLICATION_ERROR(-20004, 'Montarea nu poate fi stearsa de pe agenda
festivalului, deoarece proportia genurilor dramatice prezente '

    || 'in festival ar deveni dezechilibrata.');
```

END IF;

END IF;

END;

/

```

CREATE OR REPLACE TRIGGER conditii_agenda_festival
BEFORE UPDATE OF data OR DELETE ON agenda
FOR EACH ROW
DECLARE
    pragma autonomous_transaction;
    data_inc festival.data_inceperii%TYPE;
    data_fin festival.data_incheierii%TYPE;
    gen_p piesa.gen%TYPE;
    nr_montari_gen NUMBER;
    nr_montari_fest NUMBER;
BEGIN
    IF UPDATING THEN
        SELECT data_inceperii INTO data_inc
        FROM festival
        WHERE cod_festival = :OLD.cod_festival;
        SELECT data_incheierii INTO data_fin
        FROM festival
        WHERE cod_festival = :OLD.cod_festival;
        IF (:NEW.data < data_inc) OR (:NEW.data > data_fin) THEN
            RAISE_APPLICATION_ERROR(-20001, 'Nu se poate modifica data reprezentatiei la valoarea ' || :NEW.data || ', deoarece aceasta data se afla '
            || 'in afara intervalului de desfasurare al unui festival la care montarea participa.');
```

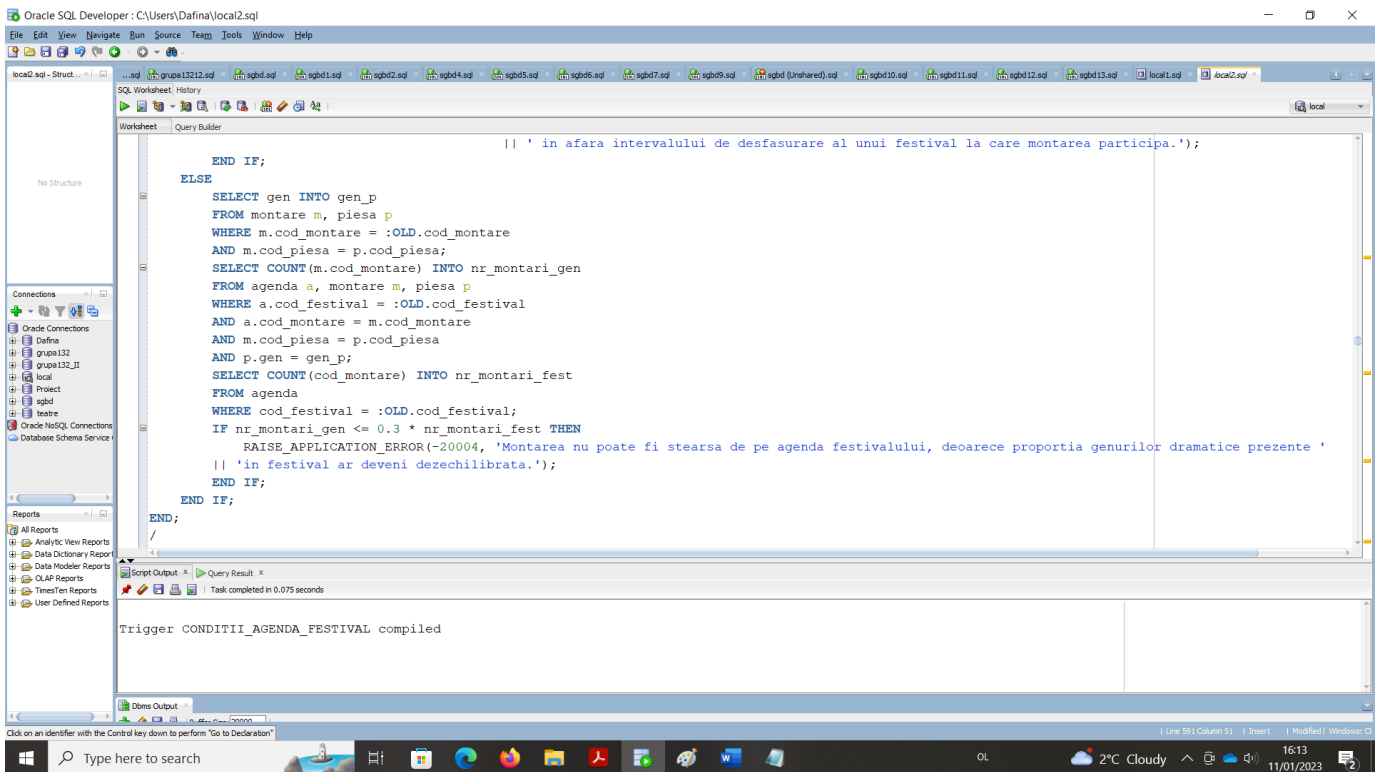
END IF;

ELSE

SELECT gen INTO gen\_p

FROM montare m, piesa p

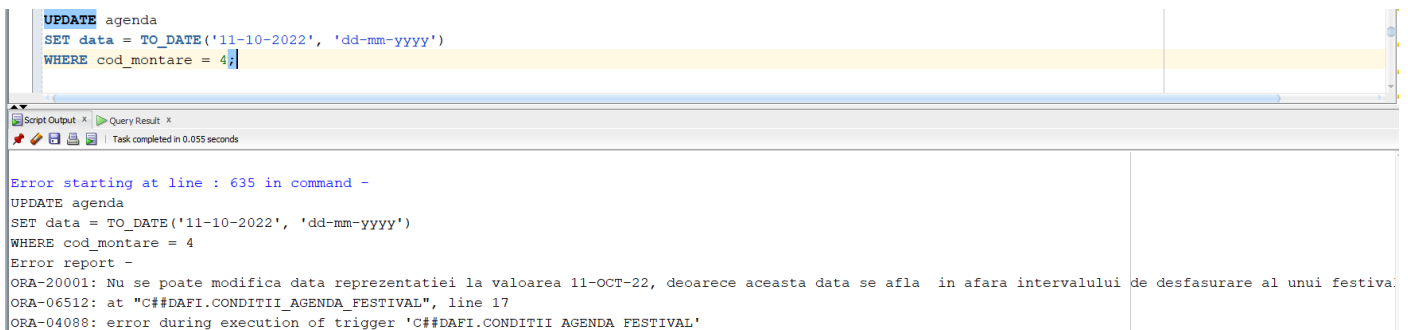
Trigger CONDITII\_AGENDA\_FESTIVAL compiled



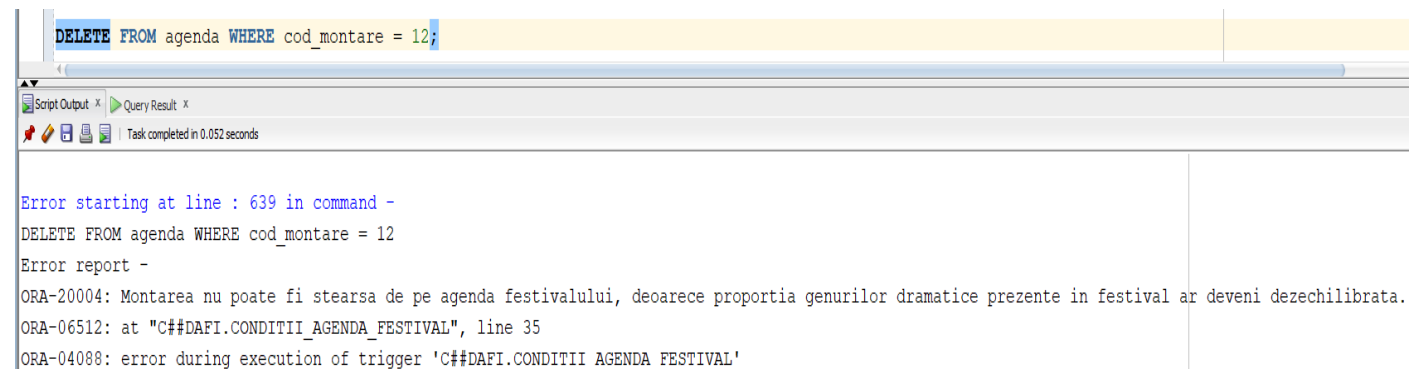
UPDATE agenda

SET data = TO\_DATE('11-10-2022', 'dd-mm-yyyy')

WHERE cod\_montare = 4;



DELETE FROM agenda WHERE cod\_montare = 12;



**12.** Sa se defineasca un trigger LDD, care, dupa executarea unei comenzi de CREATE, ALTER, sau DROP asupra unei tabele, sa declanseze inserarea in tabelul INFO\_LDD, a urmatoarelor informatii despre comanda: utilizatorul care a executat-o, momentul de timp la care aceasta a fost executata, tipul comenzii (CREATE/ALTER/DROP) si numele obiectului asupra careia a fost efectuata actiunea. De asemenea, se vor afisa mesaje prin care utilizatorul este informat ca au fost inserate datele despre comanda in tabela INFO\_LDD, si vor fi afisate mesaje de avertizare, dupa comanda ALTER efecutata asupra tabelelor care fac parte din schema de baza a proiectului (au data crearii diferita de data curenta), respectiv asupra acelor tabele pe care au mai fost aplicate comenzi LDD in ziua curenta.

```
CREATE TABLE info_ldd (  
    utilizator VARCHAR2(30),  
    moment TIMESTAMP(3),  
    comanda VARCHAR2(10),  
    tabela VARCHAR2(20)  
);  
  
CREATE OR REPLACE TRIGGER gestiune_comenzi_ldd  
    AFTER CREATE OR ALTER OR DROP ON SCHEMA  
DECLARE  
    data DATE;  
BEGIN  
    IF SYS.SYSEVENT = 'CREATE' THEN  
        dbms_output.put_line('S-au adaugat detaliile crearii efectuate, in tabela INFO_LDD. ');  
    ELSIF SYS.SYSEVENT = 'ALTER' THEN  
        dbms_output.put_line('S-au adaugat detaliile modificarii efectuate, in tabela INFO_LDD.'  
);  
  
        SELECT created INTO data  
        FROM dba_objects  
        WHERE object_name = SYS.DICTIONARY_OBJ_NAME;  
  
        IF ROUND(SYSDATE - data) = 0 THEN  
            dbms_output.put_line('A fost modificata structura unei tabele care NU face parte din  
schema de baza a proiectului.');
```

```
        ELSE  
            dbms_output.put_line('Atentie! A fost modificata structura unei tabele care face parte  
din schema de baza a proiectului.');
```

```

END IF;

SELECT last_ddl_time INTO data

FROM dba_objects

WHERE object_name = SYS.DICTIONARY_OBJ_NAME;

IF ROUND(SYSDATE - data) = 0 THEN

    dbms_output.put_line('Atentie! Ati efectuat modificari asupra unui obiect care a mai
    suferit modificari structurale in ultimele 24 de ore.');
```

END IF;

```

ELSE

    dbms_output.put_line('S-au adaugat detaliile stergerii efectuate, in tabela INFO_LDD.');
```

END IF;

```

INSERT INTO info_ldd VALUES (

    SYS.LOGIN_USER,

    SYSTIMESTAMP(3),

    SYS.SYSEVENT,

    SYS.DICTIONARY_OBJ_NAME

);

END;

```

```

CREATE OR REPLACE TRIGGER gestiune_comenzi_ldd
AFTER CREATE OR ALTER OR DROP ON SCHEMA
DECLARE
    data DATE;
BEGIN
    IF SYS.SYSEVENT = 'CREATE' THEN
        dbms_output.put_line('S-au adaugat detaliile crearii efectuate, in tabela INFO_LDD. ');
    ELSIF SYS.SYSEVENT = 'ALTER' THEN
        dbms_output.put_line('S-au adaugat detaliile modificarii efectuate, in tabela INFO_LDD. ');
        SELECT created INTO data
        FROM dba_objects
        WHERE object_name = SYS.DICTIONARY_OBJ_NAME;
        IF ROUND(SYSDATE - data) = 0 THEN
            dbms_output.put_line('A fost modificata structura unei tabele care NU face parte din schema de baza a proiectului.');
```

ELSE

```

            dbms_output.put_line('Atentie! A fost modificata structura unei tabele care face parte din schema de baza a proiectului.');
```

END IF;

```

        SELECT last_ddl_time INTO data
        FROM dba_objects
        WHERE object_name = SYS.DICTIONARY_OBJ_NAME;
        IF ROUND(SYSDATE - data) = 0 THEN
            dbms_output.put_line('Atentie! Ati efectuat modificari asupra unui obiect care a mai suferit modificari structurale in ultimele 24 de ore.');
```

END IF;

```

    ELSE
```

```

    ELSE
        dbms_output.put_line('S-au adaugat detaliile stergerii efectuate, in tabela INFO_LDD.');
```

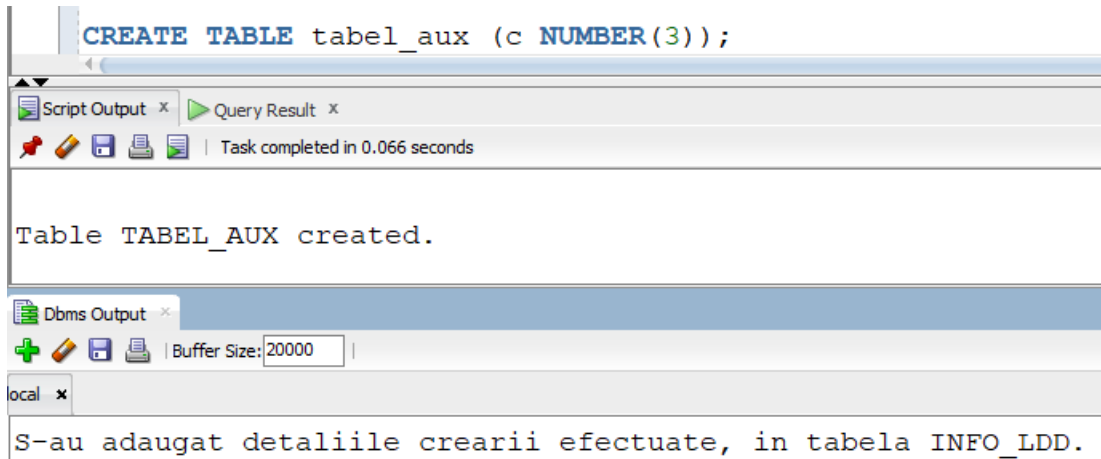
END IF;

```

    INSERT INTO info_ldd VALUES (
        SYS.LOGIN_USER,
        SYSTIMESTAMP(3),
        SYS.SYSEVENT,
        SYS.DICTIONARY_OBJ_NAME
    );
END;

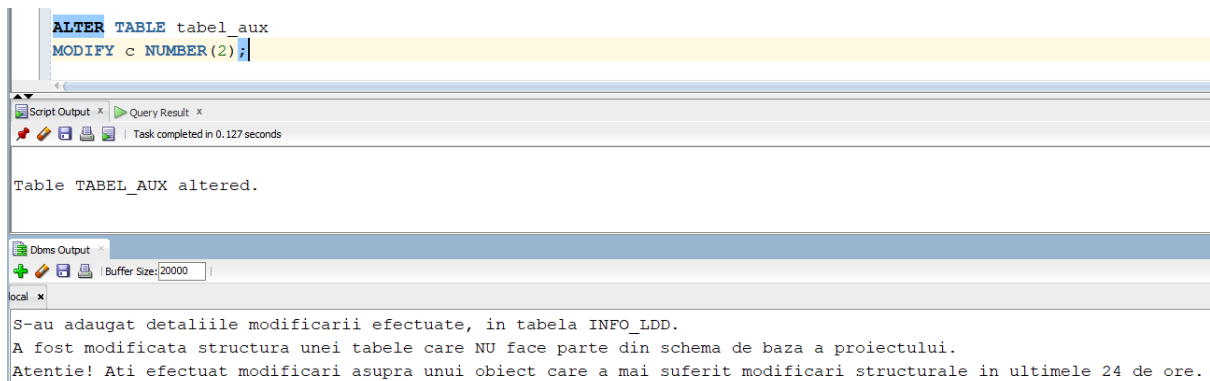
```

CREATE TABLE tabel\_aux (c NUMBER(3));

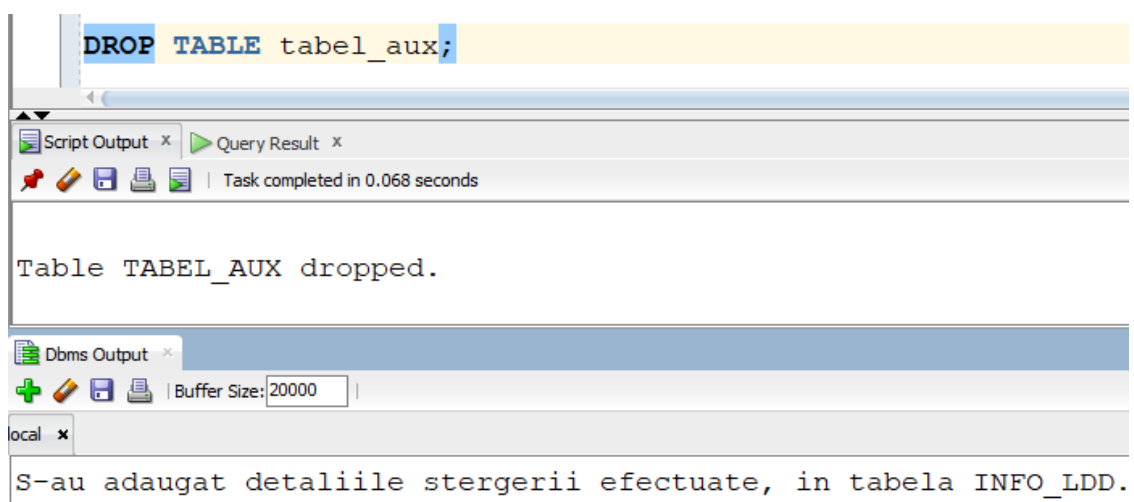


ALTER TABLE tabel\_aux

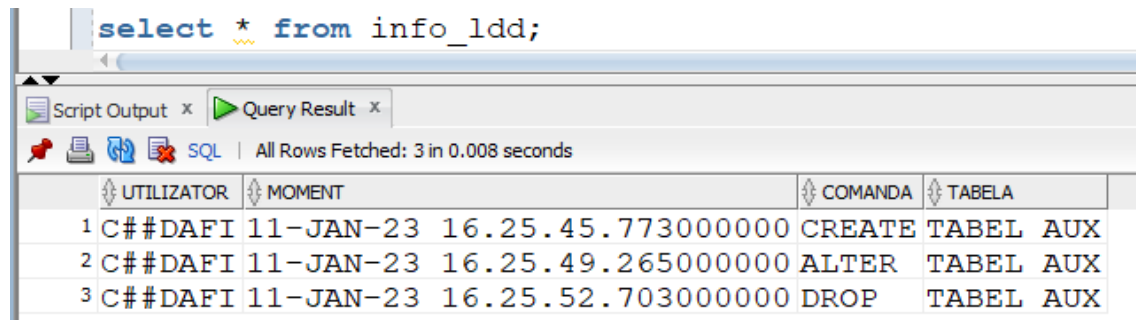
MODIFY c NUMBER(2);



DROP TABLE tabel\_aux;



SELECT \* FROM info\_ldd;

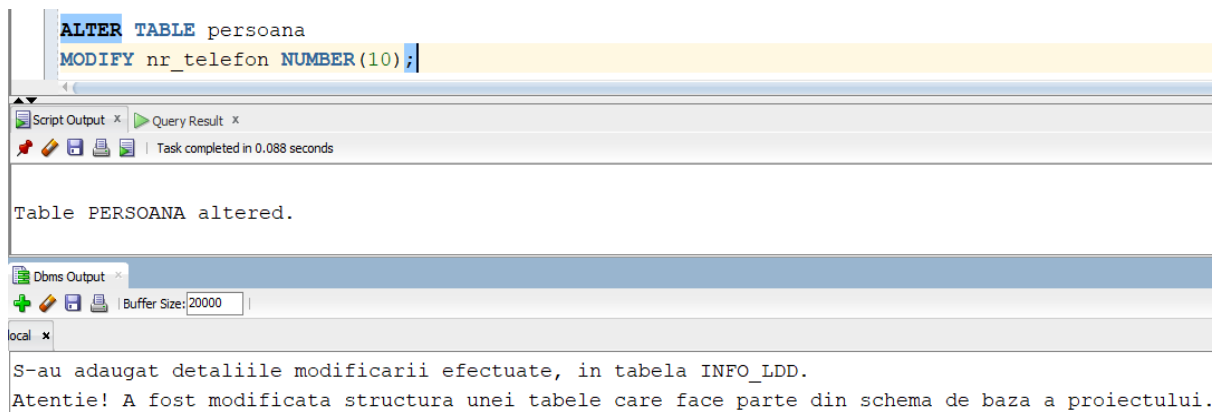


The screenshot shows a SQL query result in a table with 5 columns: UTILIZATOR, MOMENT, COMANDA, and TABELA. The first column is implicitly an index. The data is as follows:

	UTILIZATOR	MOMENT	COMANDA	TABELA
1	C##DAFI	11-JAN-23 16.25.45.773000000	CREATE	TABEL AUX
2	C##DAFI	11-JAN-23 16.25.49.265000000	ALTER	TABEL AUX
3	C##DAFI	11-JAN-23 16.25.52.703000000	DROP	TABEL AUX

ALTER TABLE persoana

MODIFY nr\_telefon NUMBER(10);



The screenshot shows a SQL script execution with the following commands and output:

```
ALTER TABLE persoana
MODIFY nr_telefon NUMBER(10);
```

Task completed in 0.088 seconds

Table PERSOANA altered.

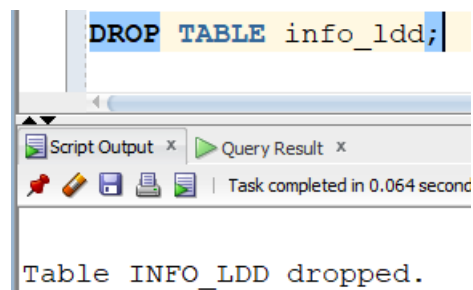
Dbms Output

Buffer Size: 20000

local

S-au adaugat detaliile modificarii efectuate, in tabela INFO\_LDD.  
Atentie! A fost modificata structura unei tabele care face parte din schema de baza a proiectului.

DROP TABLE info\_ldd;



The screenshot shows a SQL script execution with the following command and output:

```
DROP TABLE info_ldd;
```

Task completed in 0.064 seconds

Table INFO\_LDD dropped.



13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

CREATE OR REPLACE PACKAGE exercitii AS

PROCEDURE info\_sala(denumire\_sala sala.denumire%TYPE);

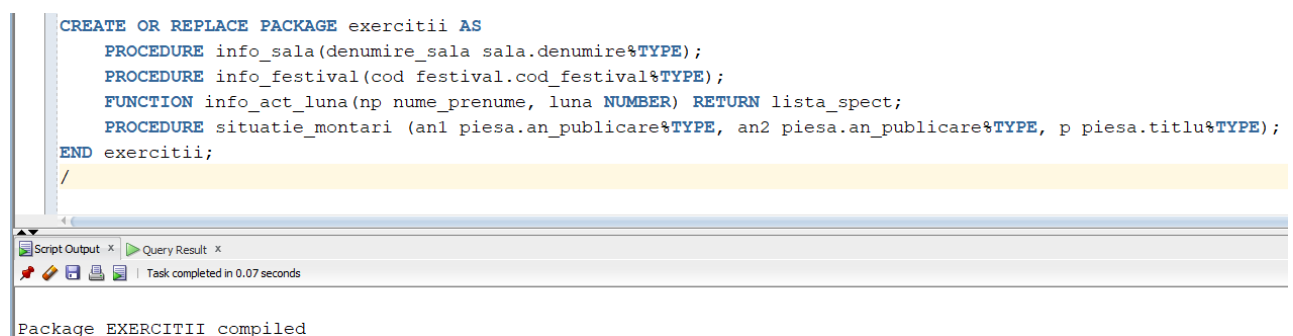
PROCEDURE info\_festival(cod festival.cod\_festival%TYPE);

FUNCTION info\_act\_luna(np nume\_prenume, luna NUMBER) RETURN lista\_spect;

PROCEDURE situatie\_montari (an1 piesa.an\_publicare%TYPE, an2  
piesa.an\_publicare%TYPE, p piesa.titlu%TYPE);

END exercitii;

/



```
CREATE OR REPLACE PACKAGE exercitii AS
  PROCEDURE info_sala(denumire_sala sala.denumire%TYPE);
  PROCEDURE info_festival(cod festival.cod_festival%TYPE);
  FUNCTION info_act_luna(np nume_prenume, luna NUMBER) RETURN lista_spect;
  PROCEDURE situatie_montari (an1 piesa.an_publicare%TYPE, an2 piesa.an_publicare%TYPE, p piesa.titlu%TYPE);
END exercitii;
/
```

Script Output x Query Result x  
Task completed in 0.07 seconds

Package EXERCITII compiled

CREATE OR REPLACE PACKAGE BODY exercitii AS

PROCEDURE info\_sala(denumire\_sala sala.denumire%TYPE) IS

TYPE info\_pers IS RECORD (  
    prenume persoana.prenume%TYPE,  
    nume persoana.nume%TYPE,  
    spec specializare.denumire%TYPE,  
    cod\_spec specializare.cod\_specializare%TYPE  
);

TYPE lista\_artisti IS TABLE OF info\_pers;  
artisti lista\_artisti;

TYPE info\_montare IS RECORD (  
    data\_pr montare.data\_premierei%TYPE,  
    gen piesa.gen%TYPE,  
    distrib lista\_artisti  
);

TYPE lista\_info\_montari IS TABLE OF info\_montare INDEX BY piesa.titlu%TYPE;

```

piese lista_info_montari;
cod_sala.cod_sala%TYPE;
p.piesa.titlu%TYPE;
nume_ant_persoana.nume%TYPE;
prenume_ant_persoana.prenume%TYPE;
specializari VARCHAR2(30);
BEGIN
SELECT cod_sala INTO cod
FROM sala
WHERE denumire = denumire_sala;
FOR i IN
(SELECT m.cod_montare cod, titlu, data_premierei, gen
FROM piesa p, montare m, afis a
WHERE p.cod_piesa = m.cod_piesa
AND m.cod_montare = a.cod_montare
AND a.cod_sala = cod) LOOP
SELECT p.prenume, p.nume, s.denumire, s.cod_specializare BULK COLLECT INTO
artisti
FROM persoana p, artist a, distributie d, afis af, spec_art sa, specializare s
WHERE p.cod_persoana = a.cod_persoana
AND a.cod_persoana = d.cod_artist
AND d.cod_montare = i.cod
AND i.cod = af.cod_montare
AND a.cod_persoana = sa.cod_artist
AND sa.cod_specializare = s.cod_specializare
AND functie = s.denumire
ORDER BY 2, 1;
SELECT i.data_premierei, i.gen INTO piese(i.titlu).data_pr, piese(i.titlu).gen
FROM DUAL;
piese(i.titlu).distrib := artisti;
END LOOP;

```

```

dbms_output.put_line('Detaliile montarilor desfasurate in ' || denumire_sala || ' sunt
urmatoarele:');

p := piese.first;

LOOP

    EXIT WHEN p IS NULL;

    dbms_output.put_line('Montarea piesei "' || p || '" (' || NVL(piese(p).gen, 'gen nespecificat')
|| '), a carei data a premierei este '

                        || NVL(TO_CHAR(piese(p).data_pr, 'DD-MON-YYYY'), 'nespecificata') ||
');

    IF piese(p).distrib.COUNT > 0 THEN

        dbms_output.put_line('La realizarea acestei montari au participat urmatorii artisti: ');

    ELSE

        dbms_output.put_line('Nu se cunosc detalii despre artistii care au participat la aceast
montare.');
```

```

        p := piese.next(p);

        CONTINUE;

    END IF;

    prenume_ant := piese(p).distrib(piese(p).distrib.FIRST).prenume;
    nume_ant := piese(p).distrib(piese(p).distrib.FIRST).nume;
    specializari := "";

    FOR i IN 1..piese(p).distrib.count LOOP

        IF (piese(p).distrib(i).prenume <> prenume_ant OR piese(p).distrib(i).nume <>
nume_ant) THEN

            dbms_output.put_line(' ' || prenume_ant || ' ' || nume_ant || ' - ' || specializari);

            specializari := piese(p).distrib(i).spec || ', ';

        ELSE

            specializari := piese(p).distrib(i).spec || ', ' || specializari;

        END IF;

        prenume_ant := piese(p).distrib(i).prenume;
        nume_ant := piese(p).distrib(i).nume;

    END LOOP;

    dbms_output.put_line(' ' || piese(p).distrib(piese(p).distrib.COUNT).prenume || ' ' ||
piese(p).distrib(piese(p).distrib.COUNT).nume

                        || ' - ' || SUBSTR(specializari, 1, LENGTH(specializari) - 2) || '.');
```

```

        dbms_output.new_line();

        p := piese.next(p);
    END LOOP;

EXCEPTION

    WHEN TOO_MANY_ROWS THEN

        dbms_output.put_line('Exista mai multe sali cu denumirea "' || denumire_sala || '".');

    WHEN NO_DATA_FOUND THEN

        dbms_output.put_line('Nu exista nicio sala cu denumirea "' || denumire_sala || '".');

END info_sala;


PROCEDURE info_festival(cod_festival.cod_festival%TYPE)
IS
    CURSOR c_mont_loc(cod_festival.cod_festival%TYPE) IS
        SELECT m.cod_montare, titlu, t1.denumire, ad1.oras
        FROM agenda a, montare m, piesa p, sala s, adresa ad1, adresa ad2, teatru t1, teatru t2
        WHERE a.cod_festival = cod
        AND a.cod_montare = m.cod_montare
        AND a.cod_sala = s.cod_sala
        AND s.cod_teatru = t1.cod_teatru
        AND t1.cod_adresa = ad1.cod_adresa
        AND m.cod_teatru = t2.cod_teatru
        AND t2.cod_adresa = ad2.cod_adresa
        AND ad1.oras = ad2.oras
        AND m.cod_piesa = p.cod_piesa;
    TYPE c_tip is REF CURSOR;
    v_cursor c_tip;
    v_montare montare.cod_montare%TYPE;
    v_piesa piesa.titlu%TYPE;
    v_teatru teatru.denumire%TYPE;
    v_oras adresa.oras%TYPE;
    v_nume persoana.nume%TYPE;

```

```

v_prenume persoana.prenume%TYPE;
v_suma contract.suma%TYPE := 0;
i NUMBER := 1;
v_denumire_fest festival.denumire%TYPE;
v_nr NUMBER;
v_total NUMBER;
v_buget montare.buget%TYPE;
montari lista_mont_loc := lista_mont_loc();
BEGIN
    SELECT denumire INTO v_denumire_fest
    FROM festival
    WHERE cod_festival = cod;
    OPEN c_mont_loc(cod);
    LOOP
        FETCH c_mont_loc INTO v_montare, v_piesa, v_teatru, v_oras;
        montari.extend;
        montari(i) := v_montare;
        i := i + 1;
        EXIT WHEN c_mont_loc%NOTFOUND;
        IF i = 2 THEN
            dbms_output.new_line();
            dbms_output.put_line('Piesele prezentate in festival, care au fost montate in teatre din
localitatea organizatoare sunt urmatoarele: ');
        END IF;
        dbms_output.put_line(' piesa "' || v_piesa || '" , montata la ' || v_teatru || ' , din ' || v_oras || ');');
    END LOOP;
    v_nr := c_mont_loc%ROWCOUNT;
    CLOSE c_mont_loc;
    SELECT COUNT(cod_montare) INTO v_total
    FROM agenda a
    WHERE a.cod_festival = cod;
    IF v_nr >= 0.5 * v_total AND v_nr > 0 THEN

```

```

OPEN v_cursor FOR SELECT DISTINCT nume, prenume
    FROM persoana p, artist a, distributie d, spec_art sa, specializare s
    WHERE p.cod_persoana = a.cod_persoana
    AND a.cod_persoana = d.cod_artist
    AND a.cod_persoana = sa.cod_artist
    AND sa.cod_specializare = s.cod_specializare
    AND s.cod_specializare = cod
    AND d.cod_montare IN (SELECT * FROM TABLE(montari));

    dbms_output.put_line('Regizorii care au realizat montari in teatrele din localitatea
organizatoare sunt: ');

    LOOP

        FETCH v_cursor INTO v_nume, v_prenume;

        EXIT WHEN v_cursor%NOTFOUND;

        dbms_output.put_line(' ' || v_prenume || ' ' || v_nume);

    END LOOP;

    CLOSE v_cursor;

ELSIF v_nr > 0 THEN

    OPEN v_cursor FOR SELECT SUM(suma), buget, m.cod_montare
        FROM contract c, montare m
        WHERE c.cod_montare(+) = m.cod_montare
        AND m.cod_montare IN (SELECT * FROM TABLE(montari))
        GROUP BY m.cod_montare, buget;

    v_total := 0;

    LOOP

        FETCH v_cursor INTO v_suma, v_buget, v_montare;

        EXIT WHEN v_cursor%NOTFOUND;

        v_total := v_total + NVL(v_suma, 0) + v_buget;

    END LOOP;

    CLOSE v_cursor;

    dbms_output.new_line();

    dbms_output.put_line('Pentru realizarea montarilor din localitatea organizatoare, s-au
alocat fonduri in valoare de: ' || v_total);

```

```

ELSE

    dbms_output.put_line('In festival nu a fost prezentata nicio piesa montata in localitatea
organizatoare.');
```

```

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        dbms_output.put_line('Nu exista niciun festival cu codul introdus.');
```

```

END info_festival;
```

```

FUNCTION info_act_luna(np nume_prenume, luna NUMBER) RETURN lista_spect IS
```

```

    np_aux nume_prenume;
    spectacole lista_spect;
    IN_AFARA_STAGIUNII EXCEPTION;
    LUNA_INVALIDA EXCEPTION;
BEGIN

    SELECT nume_prenume(nume, prenume) INTO np_aux
    FROM persoana p, artist a
    WHERE p.cod_persoana = a.cod_persoana
    AND nume = np.nume AND prenume = np.prenume;

    IF luna = 7 OR luna = 8 THEN

        RAISE IN_AFARA_STAGIUNII;

    ELSIF luna < 1 OR luna > 12 THEN

        RAISE LUNA_INVALIDA;

    END IF;

    SELECT DISTINCT spect_sala(titlu, s.denumire) BULK COLLECT INTO spectacole
    FROM montare m, piesa p, afis a, sala s, distributie d, artist art, persoana pers
    WHERE m.cod_piesa = p.cod_piesa
    AND m.cod_montare = a.cod_montare
    AND a.cod_sala = s.cod_sala
    AND d.cod_montare = m.cod_montare
    AND d.cod_artist = art.cod_persoana
    AND art.cod_persoana = pers.cod_persoana
```

```

AND pers.num = np.num AND pers.prenume = np.prenume
AND EXTRACT (MONTH FROM data) = luna;
RETURN spectacole;
EXCEPTION
    WHEN IN_AFARA_STAGIUNII THEN
        RAISE_APPLICATION_ERROR(-20001, 'Luna introdusa se afla in afara stagiunii
teatrale.');
```

```

    WHEN LUNA_INVALIDA THEN
        RAISE_APPLICATION_ERROR(-20002, 'Luna introdusa nu exista.');
```

```

    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista niciun artist cu numele introdus.');
```

```

    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20004, 'Exista mai multi artisti cu numele ' ||
np.prenume || ' ' || np.num || '.');
```

```

END info_act_luna;
```

```

PROCEDURE situatie_montari (an1 piesa.an_publicare%TYPE, an2
piesa.an_publicare%TYPE, p piesa.titlu%TYPE) IS
```

```

    cod_p piesa.cod_piesa%TYPE;
    an piesa.an_publicare%TYPE;
    t piesa.titlu%TYPE;
    TYPE v_coduri IS TABLE OF piesa.cod_piesa%TYPE;
    lista_piese_interval v_coduri := v_coduri();
    info tab_pt := tab_pt();
    nr NUMBER := 1;
    dif_ant montare.buget%TYPE := 0;
    buget_mediu montare.buget%TYPE;
    IN_AFARA_PERIOADEI_DE_INTERES EXCEPTION;
    INTERVAL_INVALID EXCEPTION;
```

```

BEGIN
    SELECT cod_piesa INTO cod_p
    FROM piesa
```



```

WHERE titlu = p;
SELECT an_publicare INTO an
FROM piesa
WHERE cod_piesa = cod_p;
IF an1 > an2 THEN
    RAISE INTERVAL_INVALID;
END IF;
IF an < an1 OR an > an2 THEN
    RAISE IN_AFARA_PERIOADEI_DE_INTERES;
END IF;
SELECT cod_piesa BULK COLLECT INTO lista_piese_interval
FROM piesa
WHERE an_publicare BETWEEN an1 AND an2;
FOR i IN lista_piese_interval.FIRST..lista_piese_interval.LAST LOOP
    SELECT titlu INTO t
    FROM piesa
    WHERE cod_piesa = lista_piese_interval(i);
    dbms_output.put_line('Informatii despre montarile piesei ' || t || ':');
    info.DELETE;
    nr := 1;
    FOR j IN (SELECT t1.denumire ttr, buget,
        (SELECT SUM(buget) FROM montare m, teatru t WHERE cod_piesa =
lista_piese_interval(i)
            AND m.cod_teatru = t.cod_teatru
            AND t.denumire = t1.denumire)
        - (SELECT AVG(buget) FROM montare m, teatru t WHERE m.cod_teatru =
t.cod_teatru
            AND t.denumire = t1.denumire) as dif,
        (SELECT COUNT(m.cod_montare) FROM montare m, teatru t WHERE cod_piesa
= lista_piese_interval(i)
            AND m.cod_teatru =
t.cod_teatru

```

```

                                AND t.denumire =
t1.denumire) as nr_mont,

        (SELECT COUNT(m.cod_montare) FROM montare m, afis a, teatru t WHERE
cod_piesa = lista_piese_interval(i)

                                AND m.cod_montare = a.cod_montare
                                AND m.cod_teatru = t.cod_teatru
                                AND t.denumire = t1.denumire) as nr_reprez,

        (SELECT COUNT(m.cod_montare) FROM montare m, agenda a, festival f, teatru t
WHERE cod_piesa = lista_piese_interval(i)

                                AND m.cod_montare =
a.cod_montare

                                AND m.cod_teatru = t.cod_teatru
                                AND t.denumire = t1.denumire
                                AND a.cod_festival = f.cod_festival
                                AND f.denumire = 'Festivalul
International de Teatru') as nr_particip

        FROM montare m, teatru t1
        WHERE m.cod_piesa = lista_piese_interval(i)
        AND m.cod_teatru = t1.cod_teatru
        ORDER BY 3 DESC) LOOP
info.extend;
info(nr) := piesa_teatru(j.ttr, j.buget, j.dif, j.nr_mont, j.nr_reprez, j.nr_particip);
nr := nr + 1;
END LOOP;
FOR j IN info.FIRST..info.LAST LOOP
    dbms_output.put_line(' La ' || info(j).t || ', unde bugetul alocat a fost de ' || info(j).buget ||
');
    IF info(j).nr_montari = 1 THEN
        dbms_output.put_line(' s-a realizat o montare a piesei, ');
        IF info(j).nr_reprezentatii = 0 THEN
            dbms_output.put_line(' care nu are programate reprezentatii in teatrul unde a fost
realizata');
            IF info(j).nr_participari = 0 THEN
                dbms_output.put_line(' si nici participari in festivaluri.');
```

```

ELSIF info(j).nr_participari = 1 then
    dbms_output.put_line(' dar are programata o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' dar are programate ' || info(j).nr_participari || ' participari
in festivaluri.');
```

END IF;

ELSIF info(j).nr\_reprezentatii = 1 THEN

```

    dbms_output.put_line(' care are programata o reprezentatie in teatrul unde a fost
realizata');
```

IF info(j).nr\_participari = 0 THEN

```

    dbms_output.put_line(' dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

    dbms_output.put_line(' si o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' si programate ' || info(j).nr_participari || ' participari in
festivaluri.');
```

END IF;

ELSE

```

    dbms_output.put_line(' care are programate ' || info(j).nr_reprezentatii || '
reprezentatii in teatrul unde a fost realizata');
```

IF info(j).nr\_participari = 0 THEN

```

    dbms_output.put_line(' dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

    dbms_output.put_line(' si o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

END IF;

ELSE

```

    dbms_output.put_line(' s-au realizat ' || info(j).nr_montari || ' montari ale piesei, ');
```

IF info(j).nr\_reprezentatii = 0 THEN

```

    dbms_output.put_line(' care nu au programate reprezentatii in teatrul unde a fost
realizata');
```

```

IF info(j).nr_participari = 0 THEN
    dbms_output.put_line(' si nici participari in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

    dbms_output.put_line(' dar au programata o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' , dar au programate ' || info(j).nr_participari || '
participari in festivaluri.');
```

END IF;

ELSIF info(j).nr\_reprezentatii = 1 THEN

```

    dbms_output.put_line(' au programata o reprezentatie in teatrul unde a fost
realizata');
```

IF info(j).nr\_participari = 0 THEN

```

    dbms_output.put_line(' , dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

    dbms_output.put_line(' , si o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

ELSE

```

    dbms_output.put_line(' au programate ' || info(j).nr_reprezentatii || ' reprezentatii in
teatrul unde au fost realizate');
```

IF info(j).nr\_participari = 0 THEN

```

    dbms_output.put_line(' dar nicio participare in festivaluri.');
```

ELSIF info(j).nr\_participari = 1 then

```

    dbms_output.put_line(' , si o participare in festival.');
```

ELSE

```

    dbms_output.put_line(' , si ' || info(j).nr_participari || ' participari in festivaluri.');
```

END IF;

END IF;

END IF;

```

    dbms_output.new_line();
```

END LOOP;

```

dbms_output.new_line();

UPDATE piesa
SET informatii_suplimentare = info
WHERE cod_piesa = lista_piese_interval(i);
END LOOP;

SELECT informatii_suplimentare INTO info
FROM piesa
WHERE cod_piesa = cod_p;

dbms_output.put_line('Primele trei pozitii in clasamentul montarilor piesei ' || p || ' sunt
ocupate de: ');

nr := 0;

FOR i IN info.FIRST..info.LAST LOOP
    IF nr <= 3 THEN
        IF info(i).dif <> dif_ant THEN
            IF nr < 3 THEN
                nr := nr + 1;

                dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' ||
info(i).buget
|| ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu
alocat unei montari
in teatrul respectiv este de ' || info(i).dif || '.');

            END IF;
        ELSE
            dbms_output.put_line(' ' || nr || '.' || info(i).t || ', unde bugetul alocat a fost de ' ||
info(i).buget
|| ', iar diferenta dintre bugetul alocat acestei montari si bugetul mediu
alocat unei montari
in teatrul respectiv este de ' || info(i).dif || '.');

        END IF;
    END IF;
    dif_ant := info(i).dif;
END LOOP;

EXCEPTION

```

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista nicio piesa cu titlul dat.');

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multe piese cu titlul dat.');

WHEN INTERVAL\_INVALID THEN

dbms\_output.put\_line('Cei doi ani introdusi nu formeaza un interval valid.');

WHEN IN\_AFARA\_PERIOADEI\_DE\_INTERES THEN

dbms\_output.put\_line('Piesa introdusa a fost publicata in afara perioadei alese.');

END situatie\_montari;

END exercitii;

/

```
CREATE OR REPLACE PACKAGE BODY exercitii AS
PROCEDURE info_sala(denumire_sala sala.denumire%TYPE) IS
    TYPE info_pers IS RECORD (
        prenume persoana.prenume%TYPE,
        nume persoana.nume%TYPE,
        spec specializare.denumire%TYPE,
        cod_spec specializare.cod_specializare%TYPE
    );
    TYPE lista_artisti IS TABLE OF info_pers;
    artisti lista_artisti;
    TYPE info_montare IS RECORD (
        data_pr montare.data_premierei%TYPE,
        gen piesa.gen%TYPE,
        distrib lista_artisti
    );
    TYPE lista_info_montari IS TABLE OF info_montare INDEX BY piesa.titlu%TYPE;
    piese lista_info_montari;
    cod_sala cod_sala%TYPE;
    p piesa.titlu%TYPE;
    nume_ant persoana.nume%TYPE;
    prenume_ant persoana.prenume%TYPE;
    specializari VARCHAR2(30);
BEGIN
    SELECT cod_sala INTO cod
    FROM sala
```

Script Output x Query Result x  
Task completed in 0.298 seconds

Package Body EXERCITII compiled

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

Obiectele incluse în pachetul creat constituie un flux de acțiuni necesare și interdependente, care sunt utile în momentul în care un regizor dorește să realizeze o nouă montare. În primul rând, acesta dorește să cunoască numărul de actori, respectiv actrite, care sunt disponibili pentru a se angaja într-un nou proiect, la respectivul moment de timp. În funcție de acest lucru, regizorul selectează din repertoriul teatral pe care îl preferă, acele piese în care există un număr de roluri masculine, respectiv feminine, mai mic sau egal decât numărul de artiști disponibili. Următorul pas, în procesul realizării montării, este găsirea unei săli de capacitate convenabilă regizorului, iar finalitatea întregului flux de acțiuni este reprezentată de participările în festivaluri de specialitate.

```
CREATE OR REPLACE PACKAGE montare_noua AS
-- tip inregistrare, care retine date despre o montare care se doreste a fi realizata
  TYPE montare IS RECORD(
    gen piesa.gen%TYPE,
    tip_sala NUMBER(4),
    t teatru.denumire%TYPE
  );
  FUNCTION nr_actori_disponibili(mf CHAR) RETURN NUMBER;
  PROCEDURE piese_candidat(nr_act_m NUMBER, nr_act_f NUMBER, gen
piesa.gen%TYPE);
  FUNCTION exista_sala(m montare) RETURN BOOLEAN;
  PROCEDURE festivaluri(m montare);
END montare_noua;
/
```

```
CREATE OR REPLACE PACKAGE montare_noua AS
-- tip inregistrare, care retine date despre o montare care se doreste a fi realizata
  TYPE montare IS RECORD(
    gen piesa.gen%TYPE,
    tip_sala NUMBER(4),
    t teatru.denumire%TYPE
  );
  FUNCTION nr_actori_disponibili(mf CHAR) RETURN NUMBER;
  PROCEDURE piese_candidat(nr_act_m NUMBER, nr_act_f NUMBER, gen piesa.gen%TYPE);
  FUNCTION exista_sala(m montare) RETURN BOOLEAN;
  PROCEDURE festivaluri(m montare);
END montare_noua;
/
```

Script Output x Query Result x  
Task completed in 0.093 seconds

Package MONTARE\_NOUA compiled

```
CREATE OR REPLACE PACKAGE BODY montare_noua AS
```

```
-- tip inregistrare, in care vom stoca numarul de roluri masculine, respectiv feminine, dintr-o piesa
```

```
TYPE info_roluri_piesa IS RECORD(
```

```
    p piesa.titlu%TYPE,
```

```
    nr_act_m NUMBER(2),
```

```
    nr_act_f NUMBER(2)
```

```
);
```

```
-- tip inregistrare, care retine, despre un festival, denumirea, numarul de reprezentatii si data de inceput a acestuia
```

```
TYPE fest IS RECORD(
```

```
    d festival.denumire%TYPE,
```

```
    nr_reprez NUMBER,
```

```
    data_inceput date
```

```
);
```

```
-- functie care obtine numarul de actori care ar putea participa la realizarea montarii (un actor este considerat disponibil, daca acesta nu este implicat,
```

```
-- in luna curenta, in mai mult de 4 reprezentatii)
```

```
FUNCTION nr_actori_disponibili(mf CHAR) RETURN NUMBER IS
```

```
    nr NUMBER(2);
```

```
BEGIN
```

```
    SELECT COUNT(cod_persoana) INTO nr
```

```
    FROM artist a, spec_art sa
```

```
    WHERE a.cod_persoana = sa.cod_artist
```

```
    AND sa.cod_specializare = 1
```

```
    AND a.sex = mf
```

```
    AND (SELECT COUNT(m.cod_montare)
```

```
        FROM distributie d, montare m, afis af
```

```
        WHERE d.cod_artist = a.cod_persoana
```

```
        AND d.cod_montare = m.cod_montare
```

```
        AND m.cod_montare = af.cod_montare
```

```
        AND EXTRACT (MONTH FROM af.data) = EXTRACT (MONTH FROM sysdate)) <= 4;
```



```

RETURN nr;

END nr_actori_disponibili;

-- procedura care primeste ca parametrii un numar de roluri masculine, un numar de roluri
feminine si un gen dramatic

-- si determina lista pieselor care ar putea fi montate, respectand conditiile date

PROCEDURE piese_candidat(nr_act_m NUMBER, nr_act_f NUMBER, gen
piesa.gen%TYPE) IS

    TYPE tip_lista_piese IS TABLE OF info_roluri_piesa;

    lista_piese tip_lista_piese := tip_lista_piese();

BEGIN

    WITH aux AS

    (SELECT p1.titlu,

    (SELECT COUNT(cod_persoana)

    FROM distributie d, montare m, piesa p, artist a, spec_art sa

    WHERE m.cod_piesa = p.cod_piesa

    AND p.titlu = p1.titlu

    AND d.cod_montare = m.cod_montare

    AND d.cod_artist = a.cod_persoana

    AND a.cod_persoana = sa.cod_artist

    AND sa.cod_specializare = 1

    AND a.sex = 'M') as nr_m,

    (SELECT COUNT(cod_persoana)

    FROM distributie d, montare m, piesa p, artist a, spec_art sa

    WHERE m.cod_piesa = p.cod_piesa

    AND p.titlu = p1.titlu

    AND d.cod_montare = m.cod_montare

    AND d.cod_artist = a.cod_persoana

    AND a.cod_persoana = sa.cod_artist

    AND sa.cod_specializare = 1

    AND a.sex = 'F') as nr_f

    FROM piesa p1)

```

```

SELECT titlu, (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) nrm,
(SELECT MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu) nrf

BULK COLLECT INTO lista_piese

FROM piesa p

WHERE (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) > 0 AND (SELECT
MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu) > 0

AND nr_act_m >= (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) AND
nr_act_f >= (SELECT MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu)

AND p.gen = gen;

dbms_output.new_line();

dbms_output.put_line('Piesele care ar putea fi montate sunt: ');

FOR i IN lista_piese.FIRST..lista_piese.LAST LOOP

    dbms_output.put_line('Piesa ' || lista_piese(i).p || ', in care exista ' || lista_piese(i).nr_act_m
|| ' roluri masculine si '

        || lista_piese(i).nr_act_f || ' roluri feminine.');
```

```

END LOOP;

END piese_candidat;

-- functie care verifica daca in teatrul in care se doreste a se realiza noua montare, exista cel
putin o sala cu capacitatea potrivita

FUNCTION exista_sala(m montare) RETURN BOOLEAN IS

    b BOOLEAN;

    nr NUMBER;

BEGIN

    SELECT COUNT(cod_sala) INTO nr

    FROM sala s, teatru t

    WHERE s.cod_teatru = t.cod_teatru

    AND t.denumire = m.t

    AND s.capacitate >= m.tip_sala - 50 AND s.capacitate <= m.tip_sala + 50;

    IF nr > 0 THEN b := true;

        ELSE b := false;

    END IF;

    RETURN b;

END exista_sala;
```

-- procedura care obtine, pentru o montare, lista festivalurilor in care aceasta ar putea fi jucata (se considera ca o montare poate fi jucata intr-un

-- festival, daca pe agenda acestuia nu exista deja mai mult de 3 montari de acelasi gen cu al noii montari)

PROCEDURE festivaluri(m montare) IS

TYPE tip\_lista\_festivaluri IS TABLE OF fest;

lista\_festivaluri tip\_lista\_festivaluri := tip\_lista\_festivaluri();

BEGIN

SELECT denumire, (SELECT COUNT(cod\_montare) FROM agenda WHERE cod\_festival = f.cod\_festival), data\_inceperii BULK COLLECT INTO lista\_festivaluri

FROM festival f

WHERE (SELECT COUNT(m.cod\_montare)

FROM agenda a, montare m, piesa p

WHERE a.cod\_festival = f.cod\_festival

AND a.cod\_montare = m.cod\_montare

AND m.cod\_piesa = p.cod\_piesa

AND p.gen = m.gen) <= 3;

dbms\_output.new\_line();

dbms\_output.put\_line('Exista ' || lista\_festivaluri.COUNT || ' festivaluri in care montarea ar putea fi prezentata:');;

FOR i IN lista\_festivaluri.FIRST..lista\_festivaluri.LAST LOOP

dbms\_output.put\_line('Festivalul ' || lista\_festivaluri(i).d || ', care incepe la data de ' || lista\_festivaluri(i).data\_inceput

|| ', in cadrul caruia sunt deja programate ' || lista\_festivaluri(i).nr\_reprez || ' spectacole.');

END LOOP;

END festivaluri;

END montare\_noua;

/

```

CREATE OR REPLACE PACKAGE BODY montare_noua AS
-- tip inregistrare, in care vom stoca numarul de roluri masculine, respectiv feminine, dintr-o piesa
TYPE info_roluri_piesa IS RECORD(
    p piesa.titlu%TYPE,
    nr_act_m NUMBER(2),
    nr_act_f NUMBER(2)
);
-- tip inregistrare, care retine, despre un festival, denumirea, numarul de reprezentatii si data de inceput a acestuia
TYPE fest IS RECORD(
    d festival.denumire%TYPE,
    nr_reprez NUMBER,
    data_inceput date
);
-- functie care obtine numarul de actori care ar putea participa la realizarea montarii (un actor este considerat disponibil, daca acesta nu este implica
-- in luna curenta, in mai mult de 4 reprezentatii)
FUNCTION nr_actors_disponibili(mf CHAR) RETURN NUMBER IS
    nr NUMBER(2);
BEGIN
    SELECT COUNT(cod_persoana) INTO nr
    FROM artist a, spec_art sa
    WHERE a.cod_persoana = sa.cod_artist
    AND sa.cod_specializare = 1
    AND a.sex = mf
    AND (SELECT COUNT(m.cod_montare)
    FROM distributie d, montare m, afis af

```

Script Output x Query Result x  
Task completed in 0.068 seconds

Package Body MONTARE\_NOUA compiled

```

(SELECT COUNT(cod_persoana)
FROM distributie d, montare m, piesa p, artist a, spec_art sa
WHERE m.cod_piesa = p.cod_piesa
AND p.titlu = pl.titlu
AND d.cod_montare = m.cod_montare
AND d.cod_artist = a.cod_persoana
AND a.cod_persoana = sa.cod_artist
AND sa.cod_specializare = 1
AND a.sex = 'F') as nr_f
FROM piesa pl)
SELECT titlu, (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) nrm, (SELECT MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu) nrf
BULK COLLECT INTO lista_piese
FROM piesa p
WHERE (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) > 0 AND (SELECT MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu) > 0
AND nr_act_m >= (SELECT MAX(nr_m) FROM aux WHERE p.titlu = aux.titlu) AND nr_act_f >= (SELECT MAX(nr_f) FROM aux WHERE p.titlu = aux.titlu)
AND p.gen = gen;
dbms_output.new_line();
dbms_output.put_line('Piese care ar putea fi montate sunt: ');
FOR i IN lista_piese.FIRST..lista_piese.LAST LOOP
    dbms_output.put_line('Piesa ' || lista_piese(i).p || ', in care exista ' || lista_piese(i).nr_act_m || ' roluri masculine si '
    || lista_piese(i).nr_act_f || ' roluri feminine. ');
END LOOP;
END piese_candidat;
-- functie care verifica daca in teatrul in care se doreste a se realiza noua montare, exista cel putin o sala cu capacitatea potrivita
FUNCTION exista_sala(m montare) RETURN BOOLEAN IS
    h BOOLEAN;

```

Script Output x Query Result x  
Task completed in 0.104 seconds

Package Body MONTARE\_NOUA compiled

```

AND (SELECT COUNT(m.cod_montare)
     FROM distributie d, montare m, afis af
     WHERE d.cod_artist = a.cod_persoana
     AND d.cod_montare = m.cod_montare
     AND m.cod_montare = af.cod_montare
     AND EXTRACT (MONTH FROM af.data) = EXTRACT (MONTH FROM sysdate)) <= 4;
RETURN nr;
END nr_actori_disponibili;
-- procedura care primeste ca parametrii un numar de roluri masculine, un numar de roluri feminine si un gen dramatic
-- si determina lista pieselor care ar putea fi montate, respectand conditiile date
PROCEDURE piese_candidat(nr_act_m NUMBER, nr_act_f NUMBER, gen piesa.gen%TYPE) IS
    TYPE tip_lista_piese IS TABLE OF info_roluri_piesa;
    lista_piese tip_lista_piese := tip_lista_piese();
BEGIN
    WITH aux AS
    (SELECT pl.titlu,
     (SELECT COUNT(cod_persoana)
      FROM distributie d, montare m, piesa p, artist a, spec_art sa
      WHERE m.cod_piesa = p.cod_piesa
      AND p.titlu = pl.titlu
      AND d.cod_montare = m.cod_montare
      AND d.cod_artist = a.cod_persoana
      AND a.cod_persoana = sa.cod_artist
      AND sa.cod_specializare = 1
      AND a.sex = 'M') as nr_m,
     (SELECT COUNT(cod_persoana)

```

Script Output x Query Result x  
Task completed in 0.068 seconds

Package Body MONTARE\_NOUA compiled

```

BEGIN
    SELECT COUNT(cod_sala) INTO nr
    FROM sala s, teatru t
    WHERE s.cod_teatru = t.cod_teatru
    AND t.denumire = m.t
    AND s.capacitate >= m.tip_sala - 50 AND s.capacitate <= m.tip_sala + 50;
    IF nr > 0 THEN b := true;
    ELSE b := false;
    END IF;
    RETURN b;
END exista_sala;
-- procedura care obtine, pentru o montare, lista festivalurilor in care aceasta ar putea fi jucata (se considera ca o montare poate fi jucata intr-un
-- festival, daca pe agenda acestuia nu exista deja mai mult de 3 montari de acelasi gen cu al noii montari)
PROCEDURE festivaluri(m montare) IS
    TYPE tip_lista_festivaluri IS TABLE OF fest;
    lista_festivaluri tip_lista_festivaluri := tip_lista_festivaluri();
BEGIN
    SELECT denumire, (SELECT COUNT(cod_montare) FROM agenda WHERE cod_festival = f.cod_festival), data_inceperii BULK COLLECT INTO lista_festivaluri
    FROM festival f
    WHERE (SELECT COUNT(m.cod_montare)
           FROM agenda a, montare m, piesa p
           WHERE a.cod_festival = f.cod_festival
           AND a.cod_montare = m.cod_montare
           AND m.cod_piesa = p.cod_piesa
           AND p.gen = m.gen) <= 3;
    dbms_output.new_line();

```

Script Output x Query Result x  
Task completed in 0.068 seconds

Package Body MONTARE\_NOUA compiled

```

    dbms_output.new_line();
    dbms_output.put_line('Exista ' || lista_festivaluri.COUNT || ' festivaluri in care montarea ar putea fi prezentata:');
    FOR i IN lista_festivaluri.FIRST..lista_festivaluri.LAST LOOP
        dbms_output.put_line('Festivalul ' || lista_festivaluri(i).d || ', care incepe la data de ' || lista_festivaluri(i).data_inceput
                             || ', in cadrul caruia sunt deja programate ' || lista_festivaluri(i).nr_reprez || ' spectacole.');
```

```

    END LOOP;
END festivaluri;

END montare_noua;
/

```

Script Output x Query Result x  
Task completed in 0.068 seconds

Package Body MONTARE\_NOUA compiled

DECLARE

nr1 NUMBER(2);

nr2 NUMBER(2);

m montare\_noua.montare;

BEGIN

nr1 := montare\_noua.nr\_actori\_disponibili('M');

dbms\_output.put\_line('Numarul de actori care sunt disponibili pentu a participa la montare: ' || nr1);

nr2 := montare\_noua.nr\_actori\_disponibili('F');

dbms\_output.put\_line('Numarul de actrite care sunt disponibile pentu a participa la montare: ' || nr2);

montare\_noua.piese\_candidat(nr1, nr2, 'comedie');

m.gen := 'comedie';

m.t := 'Teatrul National Bucuresti';

m.tip\_sala := 500;

IF montare\_noua.exista\_sala(m) THEN

dbms\_output.new\_line();

dbms\_output.put\_line('S-a gasit o sala potrivita.');

ELSE

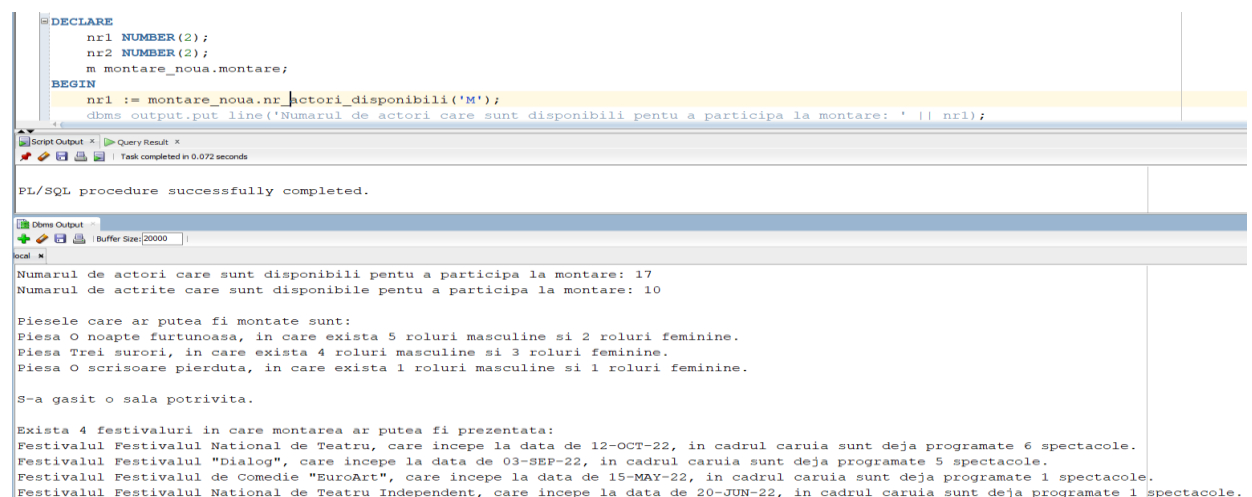
dbms\_output.new\_line();

dbms\_output.put\_line('Nu s-a gasit o sala potrivita.');

END IF;

montare\_noua.festivaluri(m);

END;/



The screenshot shows a PL/SQL execution environment. The top pane displays the PL/SQL code, which includes variable declarations, a BEGIN block with several assignments and output statements, and an IF-ELSE conditional block. The bottom pane shows the output of the execution, which includes the results of the output statements and the results of the festivaluri procedure.

```
DECLARE
  nr1 NUMBER(2);
  nr2 NUMBER(2);
  m montare_noua.montare;
BEGIN
  nr1 := montare_noua.nr_actori_disponibili('M');
  dbms_output.put_line('Numarul de actori care sunt disponibili pentu a participa la montare: ' || nr1);
  nr2 := montare_noua.nr_actori_disponibili('F');
  dbms_output.put_line('Numarul de actrite care sunt disponibile pentu a participa la montare: ' || nr2);
  montare_noua.piese_candidat(nr1, nr2, 'comedie');
  m.gen := 'comedie';
  m.t := 'Teatrul National Bucuresti';
  m.tip_sala := 500;
  IF montare_noua.exista_sala(m) THEN
    dbms_output.new_line();
    dbms_output.put_line('S-a gasit o sala potrivita.');

PL/SQL procedure successfully completed.



Numarul de actori care sunt disponibili pentu a participa la montare: 17  
Numarul de actrite care sunt disponibile pentu a participa la montare: 10



Piese care ar putea fi montate sunt:  
Piesa O noapte furtunoasa, in care exista 5 roluri masculine si 2 roluri feminine.  
Piesa Trei surori, in care exista 4 roluri masculine si 3 roluri feminine.  
Piesa O scrisoare pierduta, in care exista 1 roluri masculine si 1 roluri feminine.



S-a gasit o sala potrivita.



Exista 4 festivaluri in care montarea ar putea fi prezentata:  
Festivalul Festivalul National de Teatru, care incepe la data de 12-OCT-22, in cadrul caruia sunt deja programate 6 spectacole.  
Festivalul Festivalul "Dialog", care incepe la data de 03-SEP-22, in cadrul caruia sunt deja programate 5 spectacole.  
Festivalul Festivalul de Comedie "EuroArt", care incepe la data de 15-MAY-22, in cadrul caruia sunt deja programate 1 spectacole.  
Festivalul Festivalul National de Teatru Independent, care incepe la data de 20-JUN-22, in cadrul caruia sunt deja programate 1 spectacole.


```