

# Proiect - Sisteme de gestiune ale bazelor de date

Laura - Maria Tender

January 2021

## Contents

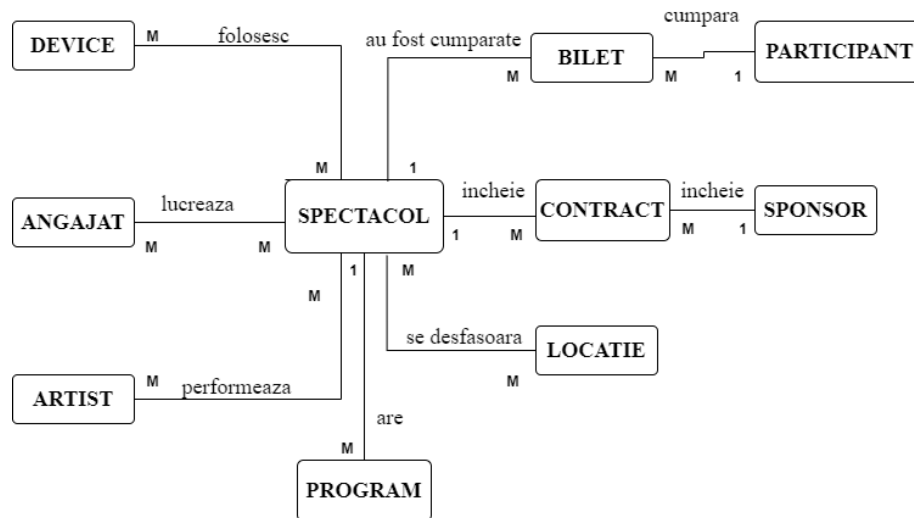
1	Prezentarea bazei de date	2
2	Diagrama ERD	2
3	Diagrama conceptuală	3
4	Implementarea modelului	4
5	Adăugarea datelor	7
6	Subprogram stocat care utilizează un tip de colecție studiat	16
7	Subprogram stocat care utilizează un tip de cursor studiat	17
8	Funție care utilizează 3 tabele	19
9	Procedură care utilizează 5 tabele	21
10	Trigger LMD la nivel de comanda	25
11	Trigger LMD la nivel de linie	25
12	Trigger LDD	26
13	Pachetul p_cerinte	28
14	Pachetul p_get_data	28
15	Concluzii	32

## 1 Prezentarea bazei de date

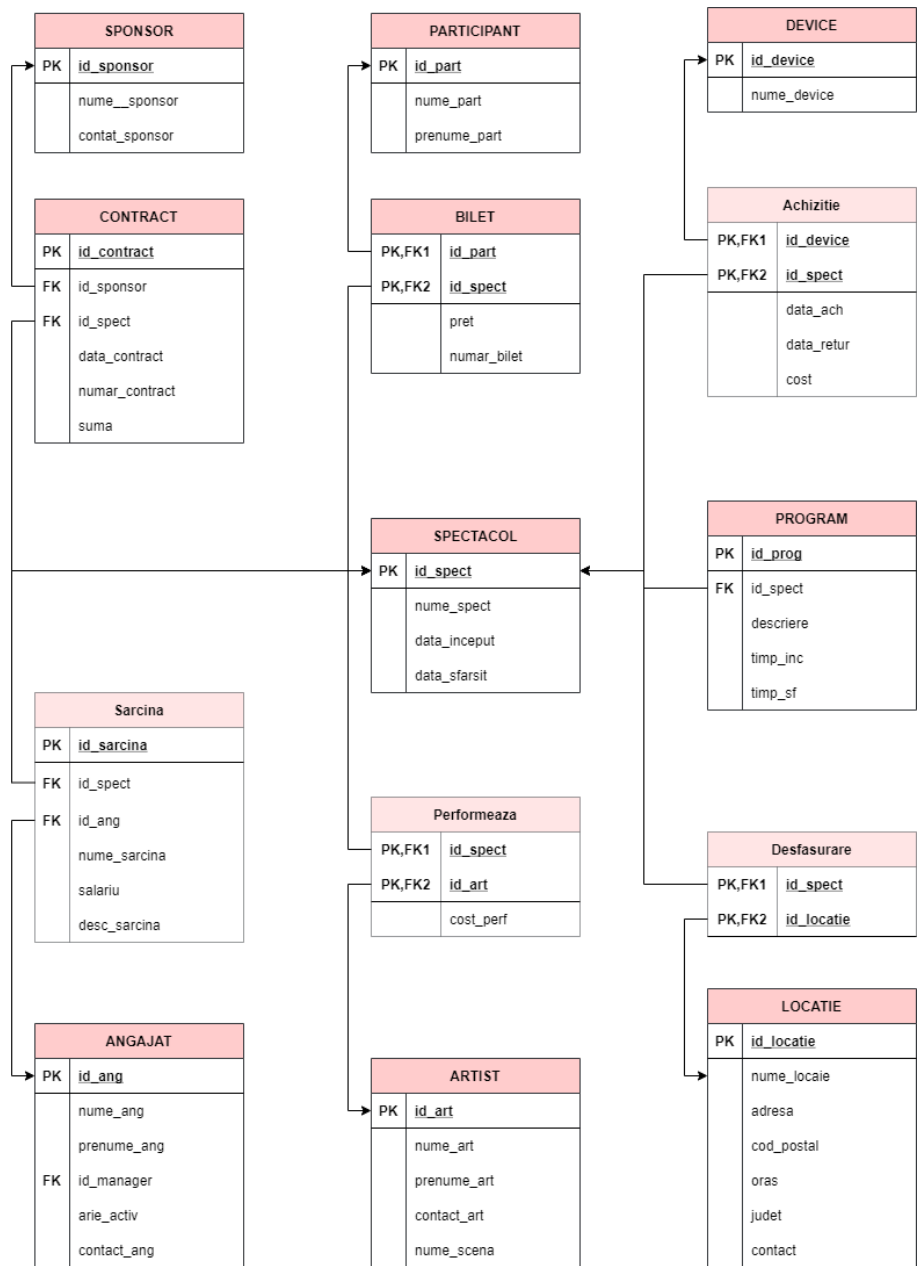
În acest proiect, am ales să descriu modelul unei firme organizatoare de spectacole. La spectacole performează mai mulți artiști. Acestea se pot desfășura pe o perioadă mai lungă, în mai multe locații și au un program. În cadrul spectacolelor angajații au sarcini pentru care sunt plătiți individual. Există o ierarhie a angajaților. Participarea la spectacole se face pe baza unui bilet. Pentru organizarea spectacolelor pot fi închiriate sau achiziționate device-uri. De asemenea, spectacolele pot fi sponsorizate de firme cu care sunt încheiate contracte.

Am gândit modelul cu următoarele entități: spectacol, artist, locație, program, angajat, bilet, participant, device, sponsor, contract. Acesta va include și următoarele tabele asociative: performează (între artist și spectacol), desfășurare (între locație și spectacol), sarcină (între angajat și spectacol), achiziție (între device și spectacol).

## 2 Diagrama ERD



### 3 Diagrama conceptuală



## 4 Implementarea modelului

La crearea tabelelor am implementat constrângerile de cheie primară, de cheie externă, *NOT NULL*, *CHECK* și *UNIQUE*.

Pentru a implementa coloanele contact am folosit următorul *TYPE*.

```
1 CREATE OR REPLACE TYPE contact IS OBJECT (telefon VARCHAR(10),
2     email VARCHAR(30));
3 /
4 CREATE OR REPLACE TYPE contacte IS TABLE OF contact;
5 /
```

Codul următor este crearea tabelului locație cu constrângerile necesare printre care și *CHECK* și *UNIQUE*. Tabelul conține tabelul Contact ca nested table.

```
1 CREATE TABLE Locatie(
2     id_locatie INT PRIMARY KEY,
3     nume_locatie VARCHAR(20) NOT NULL,
4     adresa VARCHAR(50) NOT NULL,
5     cod_postal VARCHAR(10) CHECK (length(cod_postal) = 6),
6     oras VARCHAR(15) NOT NULL,
7     judet VARCHAR(15) NOT NULL,
8     contact contacte)
9     NESTED TABLE contact STORE AS date_contact;
10 /
11
12 ALTER TABLE Locatie
13 ADD UNIQUE(adresa, oras, judet);
```

Tabelul spectacol

```
1 CREATE TABLE Spectacol(
2     id_spect INT PRIMARY KEY,
3     nume_spect VARCHAR(20) NOT NULL,
4     data_inceput DATE NOT NULL,
5     data_sfarsit DATE NOT NULL
6 );
```

Tabelul sponsor

```
1 CREATE TABLE Sponsor(
2     id_sponsor INT PRIMARY KEY,
3     nume_sponsor VARCHAR(15) NOT NULL,
4     contact_sponsor contacte)
5     NESTED TABLE contact_sponsor STORE AS date_contact_s;
6 /
```

Tabelul contract

```
1 CREATE TABLE Contract(
2     id_contract INT PRIMARY KEY,
3     id_sponsor INT,
4     id_spect INT,
5     data_contract DATE NOT NULL,
6     numar_contract VARCHAR(10) NOT NULL UNIQUE,
7     suma NUMBER(6),
8     CONSTRAINT FK_Contract_Sponsor FOREIGN KEY (id_sponsor)
9     REFERENCES Sponsor(id_sponsor),
```

```

9      CONSTRAINT FK_Contract_Spectacol FOREIGN KEY (id_spect)
      REFERENCES Spectacol(id_spect)
10 );

```

Pentru tabelul Program, am ales să rețin timpul de început și de sfârșit ca *TIMESTAMP*.

```

1 CREATE TABLE Program(
2     id_prog INT,
3     id_spect INT,
4     descriere VARCHAR(50) NOT NULL,
5     timp_inc TIMESTAMP NOT NULL,
6     timp_sf TIMESTAMP NOT NULL,
7     CONSTRAINT PK_Program PRIMARY KEY(id_prog, id_spect),
8     CONSTRAINT FK_Program_Spectacol FOREIGN KEY (id_spect)
      REFERENCES Spectacol(id_spect)
9 );

```

Tabelul Angajat

```

1 CREATE TABLE Angajat(
2     id_ang INT PRIMARY KEY,
3     nume_ang VARCHAR(15) NOT NULL,
4     prenume_ang VARCHAR(25) NOT NULL,
5     id_manager INT,
6     arie_activ VARCHAR(15),
7     contact_ang contacte,
8     CONSTRAINT FK_Angajat FOREIGN KEY (id_manager) REFERENCES
      Angajat(id_ang))
9     NESTED TABLE contact_ang STORE AS date_contact_a;
10 /

```

Tabelul Sarcina

```

1 CREATE TABLE Sarcina(
2     id_sarcina INT,
3     id_spect INT,
4     id_ang INT,
5     nume_sarcina VARCHAR(20) NOT NULL,
6     salariu NUMBER(5),
7     desc_sarcina VARCHAR(50),
8     CONSTRAINT PK_Sarcina PRIMARY KEY (id_sarcina, id_spect, id_ang
9     ),
10     CONSTRAINT FK_sarcina_Spectacol FOREIGN KEY (id_spect)
      REFERENCES Spectacol(id_spect),
11     CONSTRAINT FK_Sarcina_Angajat FOREIGN KEY (id_ang) REFERENCES
      Angajat(id_ang)
12 );

```

Tabelul Artist

```

1 CREATE TABLE Artist(
2     id_art INT PRIMARY KEY,
3     nume_art VARCHAR(15) NOT NULL,
4     prenume_art VARCHAR(25) NOT NULL,
5     contact_art contacte,
6     nume_scena VARCHAR(15))
7     NESTED TABLE contact_art STORE AS date_contact_ar;
8 /

```

#### Tabelul Performeaza

```
1 CREATE TABLE Performeaza(  
2     id_art INT,  
3     id_spect INT,  
4     cost_perf NUMBER(4),  
5     CONSTRAINT PK_Performeaza PRIMARY KEY (id_art, id_spect),  
6     CONSTRAINT FK_Performeaza_Artist FOREIGN KEY (id_art)  
7     REFERENCES Artist(id_art),  
8     CONSTRAINT FK_Performeaza_Spectacol FOREIGN KEY (id_spect)  
9     REFERENCES Spectacol(id_spect)  
10 );  
11 /
```

#### Tabelul Participant

```
1 CREATE TABLE Participant(  
2     id_part INT PRIMARY KEY,  
3     nume_part VARCHAR(20) NOT NULL,  
4     prenume_part VARCHAR(25) NOT NULL  
5 );
```

#### Tabelul Bilet

```
1 CREATE TABLE Bilet(  
2     id_part INT,  
3     id_spect INT,  
4     pret NUMBER(6, 2),  
5     numar_bilet NUMBER(10),  
6     CONSTRAINT PK_Bilet PRIMARY KEY (id_part, id_spect),  
7     CONSTRAINT FK_Bilet_Participant FOREIGN KEY (id_part)  
8     REFERENCES Participant(id_part),  
9     CONSTRAINT FK_Bilet_Spectacol FOREIGN KEY (id_spect) REFERENCES  
10    Spectacol(id_spect)  
11 );
```

#### Tabelul Device

```
1 CREATE TABLE Device(  
2     id_device INT PRIMARY KEY,  
3     nume_device VARCHAR(15) NOT NULL  
4 );
```

În tabelul Achizitie, dacă data de retur este nulă atunci device-ul a fost cumpărat, nu închiriat.

```
1 CREATE TABLE Achizitie(  
2     id_device INT,  
3     id_spect INT,  
4     data_ach DATE NOT NULL,  
5     data_retur DATE,  
6     cost NUMBER(8, 2),  
7     CONSTRAINT PK_Achizitie PRIMARY KEY (id_device, id_spect),  
8     CONSTRAINT FK_Achizitie_Device FOREIGN KEY (id_device)  
9     REFERENCES Device(id_device),  
10    CONSTRAINT FK_Achizitie_Spectacol FOREIGN KEY (id_spect)  
11    REFERENCES Spectacol(id_spect)  
12 );
```

#### Tabelul Desfasurare

```

1 CREATE TABLE Desfasurare(
2     id_spect INT,
3     id_locatie INT,
4     CONSTRAINT PK_Desfasurare PRIMARY KEY(id_spect, id_locatie),
5     CONSTRAINT FK_Desfasurare_Spectacol FOREIGN KEY (id_spect)
6     REFERENCES Spectacol (id_spect),
7     CONSTRAINT FK_Desfasurare_Locatie FOREIGN KEY (id_locatie)
8     REFERENCES Locatie (id_locatie)
9 );

```

## 5 Adăugarea datelor

Pentru a obține id-urile am folosit următoarea secvență:

```

1 CREATE SEQUENCE seq_id
2 MINVALUE 1
3 START WITH 1
4 INCREMENT BY 1
5 CACHE 10;

```

În cadrul operațiilor de *INSERT* folosesc funcții din pachetul *p-get\_data* care poate fi găsit la secțiunea 14.

Inserări în tabelul Spectacol

```

1 INSERT INTO Spectacol
2 VALUES (seq_id.nextval, 'Comedy Show', '16-DEC-20', '16-DEC-20');
3 INSERT INTO Spectacol
4 VALUES (seq_id.nextval, 'Poveste Craciun', '29-DEC-20', '29-DEC-20'
5 );
6 INSERT INTO Spectacol
7 VALUES (seq_id.nextval, 'Comedy Show', '23-JAN-21', '23-JAN-21');
8 INSERT INTO Spectacol
9 VALUES (seq_id.nextval, 'Lacul lebedelor', '10-MAR-21', '10-MAR-21'
10 );
11 INSERT INTO Spectacol
12 VALUES (seq_id.nextval, 'Festival vara', '13-AUG-21', '15-AUG-21');
13 INSERT INTO Spectacol
14 VALUES (seq_id.nextval, 'Tosca', '13-MAY-21', '13-MAY-21');

```

Inserări în tabelul Locație

```

1 INSERT INTO Locatie
2 VALUES (seq_id.nextval, 'Club 99', 'Bd. Dacia, nr.22', NULL, '
3 Bucuresti', 'Sector 2', contacte(contact('0726688658', '
4 manager@99club.ro'), contact('0733500301', 'office@club99.ro'))
5 );
6 INSERT INTO Locatie
7 VALUES (seq_id.nextval, 'Teatrul Coquette', 'Calea Calarasi, nr.94'
8 , NULL, 'Bucuresti', 'Sector 3', contacte(contact('0754990017',
9 'acasa@teatrulcoquette.ro'), contact('0721191675', '
10 rezervari@teatrulcoquette.ro')));
11 INSERT INTO Locatie
12 VALUES (seq_id.nextval, 'Comics Club', 'str. Nerva Traian, nr. 4',
13 '031045', 'Bucuresti', 'Sector 3', contacte(contact('021139437'
14 , 'secretariat.tnb@gmail.com')));
15 INSERT INTO Locatie

```



```

8 VALUES (seq_id.nextval, 'Teatrul National', 'bd. Nicolae Balcescu
   Nr.2', '010051', 'Bucuresti', 'Sector 1', contacte(contact('
   0741426426', 'events@comicsclub.ro')));
9 INSERT INTO Locatie
10 VALUES (seq_id.nextval, 'Stirbei sala mare', 'Strada Stirbei Voda
   36', '070000', 'Buftea', 'Ilfov', contacte(contact('0732332906',
   'office@domeniu.ro')));
11 INSERT INTO Locatie
12 VALUES (seq_id.nextval, 'Stirbei sala mica', 'Strada Stirbei Voda
   37', '070000', 'Buftea', 'Ilfov', contacte(contact('0732332906',
   'office@domeniu.ro')));
13 INSERT INTO Locatie
14 VALUES (seq_id.nextval, 'Stirbei pavilion', 'Strada Stirbei Voda 39
   ', '070000', 'Buftea', 'Ilfov', contacte(contact('0732332906',
   'office@domeniu.ro')));
15 INSERT INTO Locatie
16 VALUES (seq_id.nextval, 'Stirbei gradina', 'Strada Stirbei Voda 38'
   , '070000', 'Buftea', 'Ilfov', contacte(contact('0732332906', '
   office@domeniu.ro')));
17 INSERT INTO Locatie
18 VALUES (seq_id.nextval, 'Stirbei domeniu', 'Strada Stirbei Voda 40'
   , '070000', 'Buftea', 'Ilfov', contacte(contact('0732332906', '
   office@domeniu.ro')));
19 INSERT INTO Locatie
20 VALUES (seq_id.nextval, 'Opera nationala', 'Bd. Mihail Kogalniceanu
   70-72', NULL, 'Bucuresti', 'Sector 5', contacte(contact('
   0757030555', 'bilete@operanb.ro')));

```

Inserări în tabelul Desfășurare Observație: dacă există mai multe spectacole cu același nume, funcția p\_get\_data.f\_id\_spectacol va da eroare. Apelând funcția p\_cerinte.f\_afis\_spectacole putem vedea toate spectacolele cu acel nume și să alegem id-ul spectacolului dorit. În cazul inserărilor mele acelea erau id-urile dar la o nouă rulare a scriptului vor fi obținute alte valori.

```

1 INSERT INTO Desfasurare
2 VALUES (1, p_get_data.f_id_locatie('Club 99'));
3 INSERT INTO Desfasurare
4 VALUES (p_get_data.f_id_spectacol('Poveste Craciun'), p_get_data.
   f_id_locatie('Teatrul Coquette'));
5 INSERT INTO Desfasurare
6 VALUES (3, p_get_data.f_id_locatie('Comics Club'));
7 INSERT INTO Desfasurare
8 VALUES (p_get_data.f_id_spectacol('Tosca'), p_get_data.f_id_locatie
   ('Opera nationala'));
9 INSERT INTO Desfasurare
10 VALUES (p_get_data.f_id_spectacol('Lacul lebedelor'), p_get_data.
   f_id_locatie('Teatrul National'));
11 INSERT INTO Desfasurare
12 VALUES (p_get_data.f_id_spectacol('Festival vara'), p_get_data.
   f_id_locatie('Stirbei sala mare'));
13 INSERT INTO Desfasurare
14 VALUES (p_get_data.f_id_spectacol('Festival vara'), p_get_data.
   f_id_locatie('Stirbei sala mica'));
15 INSERT INTO Desfasurare
16 VALUES (p_get_data.f_id_spectacol('Festival vara'), p_get_data.
   f_id_locatie('Stirbei pavilion'));
17 INSERT INTO Desfasurare

```

```

18 VALUES (p_get_data.f_id_spectacol('Festival vara'), p_get_data.
    f_id_locatie('Stirbei gradina'));
19 INSERT INTO Desfasurare
20 VALUES (p_get_data.f_id_spectacol('Festival vara'), p_get_data.
    f_id_locatie('Stirbei domeniu'));

```

Inserări în tabelul Artist

```

1 INSERT INTO Artist
2 VALUES (seq_id.nextval, 'Teohari', 'Claudiu', contacte(contact('
    0727799769', 'teo_stand-up@gmail.com')), 'Teo');
3 INSERT INTO Artist
4 VALUES (seq_id.nextval, 'Dragu', 'Viorel', contacte(contact('
    0737700789', 'vio_stand-up@gmail.com')), 'Vio');
5 INSERT INTO Artist
6 VALUES (seq_id.nextval, 'Bojog', 'Costel', contacte(contact('
    0717765769', 'costel_bojog@gmail.com')), 'Costel');
7 INSERT INTO Artist
8 VALUES (seq_id.nextval, 'Nita', 'Gabriela', NULL, NULL);
9 INSERT INTO Artist
10 VALUES (seq_id.nextval, 'Nitu', 'Celina', NULL, NULL);
11 INSERT INTO Artist
12 VALUES (seq_id.nextval, 'Covache', 'Marius', contacte(contact('
    0759220221', 'marius_covache@gmail.com')), NULL);
13 INSERT INTO Artist
14 VALUES (seq_id.nextval, 'Serban', 'Alex', contacte(contact('
    0751094647', 'alex_serban@gmail.com')), NULL);
15 INSERT INTO Artist
16 VALUES (seq_id.nextval, 'Ignat', 'Stefan', NULL, NULL);
17 INSERT INTO Artist
18 VALUES (seq_id.nextval, 'Pascariu', 'Daniel', NULL, NULL);
19 INSERT INTO Artist
20 VALUES (seq_id.nextval, 'Nita', 'Greta', NULL, NULL);
21 INSERT INTO Artist
22 VALUES (seq_id.nextval, 'Enache', 'Robert', NULL, NULL);
23 INSERT INTO Artist
24 VALUES (seq_id.nextval, 'Bogaerde', 'Jasmine', NULL, 'Birdy');
25 INSERT INTO Artist
26 VALUES (seq_id.nextval, 'Trupa', 'Trupa', NULL, 'Ntg but tvs');
27 INSERT INTO Artist
28 VALUES (seq_id.nextval, 'Trupa', 'Trupa', NULL, 'Roosevelt');
29 INSERT INTO Artist
30 VALUES (seq_id.nextval, 'Trupa', 'Trupa', NULL, 'The 1975');
31 INSERT INTO Artist
32 VALUES (seq_id.nextval, 'August', 'David', NULL, NULL);

```

Inserări în tabelul Performeaza

```

1 INSERT INTO Performeaza
2 VALUES (p_get_data.f_id_artist('Teo'), 1, 500);
3 INSERT INTO Performeaza
4 VALUES (p_get_data.f_id_artist('Vio'), 1, 500);
5 INSERT INTO Performeaza
6 VALUES (p_get_data.f_id_artist('Costel'), 1, 500);
7 INSERT INTO Performeaza
8 VALUES (p_get_data.f_id_artist('Nita', 'Gabriela'), p_get_data.
    f_id_spectacol('Poveste Craciun'), 300);
9 INSERT INTO Performeaza

```

```

10 VALUES (p_get_data.f_id_artist('Nitu', 'Celina'), p_get_data.
    f_id_spectacol('Poveste Craciun'), 300);
11 INSERT INTO Performeaza
12 VALUES (p_get_data.f_id_artist('Covache', 'Marius'), 3, 250);
13 INSERT INTO Performeaza
14 VALUES (p_get_data.f_id_artist('Serban', 'Alex'), 3, 250);
15 INSERT INTO Performeaza
16 VALUES (p_get_data.f_id_artist('Ignat', 'Stefan'), p_get_data.
    f_id_spectacol('Tosca'), 250);
17 INSERT INTO Performeaza
18 VALUES (p_get_data.f_id_artist('Pascariu', 'Daniel'), p_get_data.
    f_id_spectacol('Tosca'), 250);
19 INSERT INTO Performeaza
20 VALUES (p_get_data.f_id_artist('Nita', 'Greta'), p_get_data.
    f_id_spectacol('Lacul lebedelor'), 250);
21 INSERT INTO Performeaza
22 VALUES (p_get_data.f_id_artist('Enache', 'Robert'), p_get_data.
    f_id_spectacol('Lacul lebedelor'), 250);
23 INSERT INTO Performeaza
24 VALUES (p_get_data.f_id_artist('August', 'David'), p_get_data.
    f_id_spectacol('Festival vara'), 1000);
25 INSERT INTO Performeaza
26 VALUES (p_get_data.f_id_artist('August', 'David'), p_get_data.
    f_id_spectacol('Festival vara'), 1000);
27 INSERT INTO Performeaza
28 VALUES (p_get_data.f_id_artist('Birdy'), p_get_data.f_id_spectacol(
    'Festival vara'), 2000);
29 INSERT INTO Performeaza
30 VALUES (p_get_data.f_id_artist('Ntg but tvs'), p_get_data.
    f_id_spectacol('Festival vara'), 2000);
31 INSERT INTO Performeaza
32 VALUES (p_get_data.f_id_artist('The 1975'), p_get_data.
    f_id_spectacol('Festival vara'), 2000);
33 INSERT INTO Performeaza
34 VALUES (p_get_data.f_id_artist('Roosevelt'), p_get_data.
    f_id_spectacol('Festival vara'), 1500);

```

#### Inserări în tabelul Sponsor

```

1 INSERT INTO Contract
2 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Softbinator'), 1,
    '14-DEC-20', '000534', 3000);
3 INSERT INTO Contract
4 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Kiss FM'), 3, '15-
    JAN-21', '000538', 1000);
5 INSERT INTO Contract
6 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Elle'), p_get_data.
    f_id_spectacol('Lacul lebedelor'), '28-FEB-21', '000596',
    2000);
7 INSERT INTO Contract
8 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Aqua Carpatica'),
    p_get_data.f_id_spectacol('Festival vara'), '01-AUG-21', '
    000535', 5000);
9 INSERT INTO Contract
10 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Pepsi'),
    p_get_data.f_id_spectacol('Festival vara'), '01-AUG-21', '
    000536', 5000);
11 INSERT INTO Contract

```

```

12 VALUES (seq_id.nextval, p_get_data.f_id_sponsor('Kiss FM'),
           p_get_data.f_id_spectacol('Festival vara'), '01-AUG-21', '
           000537', 5000);

```

Inserări în tabelul Contract

```

1 INSERT INTO Sponsor
2 VALUES (seq_id.nextval, 'Softbinator', contacte(contact('0757404781
           ', 'contact@softbinator.com')));
3 INSERT INTO Sponsor
4 VALUES (seq_id.nextval, 'Aqua Carpatica', contacte(contact('
           8477305616', 'contact@aquacarpatica.com')));
5 INSERT INTO Sponsor
6 VALUES (seq_id.nextval, 'Pepsi', contacte(contact('021467704', '
           contact@pepsi.com')));
7 INSERT INTO Sponsor
8 VALUES (seq_id.nextval, 'Kiss FM', contacte(contact('0213188000', '
           contact@kiss.com')));
9 INSERT INTO Sponsor
10 VALUES (seq_id.nextval, 'Elle', contacte(contact('0212030800', '
           contact@elle.com')));

```

Inserări în tabelul Program

```

1 INSERT INTO Program
2 VALUES (seq_id.nextval, 1, 'Show-ul lui Teo', timestamp '2020-12-16
           20:00:00', timestamp '2020-12-16 20:20:00');
3 INSERT INTO Program
4 VALUES (seq_id.nextval, 1, 'Show-ul lui Vio', timestamp '2020-12-16
           20:20:00', timestamp '2020-12-16 20:40:00');
5 INSERT INTO Program
6 VALUES (seq_id.nextval, 1, 'Show-ul lui Costel', timestamp '
           2020-12-16 20:40:00', timestamp '2020-12-16 21:00:00');
7 INSERT INTO Program
8 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Poveste Craciun'
           ), 'Teatru', timestamp '2020-12-29 19:00:00', timestamp '
           2020-12-29 21:00:00');
9 INSERT INTO Program
10 VALUES (seq_id.nextval, 3, 'Stand-up comedy', timestamp '2021-01-23
           20:00:00', timestamp '2021-01-23 21:00:00');
11 INSERT INTO Program
12 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Lacul lebedelor'
           ), 'Balet', timestamp '2021-03-10 19:00:00', timestamp '
           2021-03-10 21:00:00');
13 INSERT INTO Program
14 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Tosca'), 'Opera'
           , timestamp '2021-05-13 20:00:00', timestamp '2021-05-13
           22:00:00');
15 INSERT INTO Program
16 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
           'Roosevelt', timestamp '2021-08-13 20:00:00', timestamp '
           2021-08-13 21:30:00');
17 INSERT INTO Program
18 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
           'Nothing but Thieves', timestamp '2021-08-13 21:30:00',
           timestamp '2021-08-13 23:00:00');
19 INSERT INTO Program
20 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
           'Birdy', timestamp '2021-08-14 20:00:00', timestamp '

```

```

21 INSERT INTO Program
22 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
          'The 1975', timestamp '2021-08-15 20:00:00', timestamp '
          2021-08-15 21:30:00');
23 INSERT INTO Program
24 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
          'August David', timestamp '2021-08-15 21:30:00', timestamp '
          2021-08-15 23:00:00');

```

Inserări în tabelul Device

```

1 INSERT INTO Device
2 VALUES (seq_id.nextval, 'Camera video');
3 INSERT INTO Device
4 VALUES (seq_id.nextval, 'Lumini');
5 INSERT INTO Device
6 VALUES (seq_id.nextval, 'Boxa');
7 INSERT INTO Device
8 VALUES (seq_id.nextval, 'Microfon');
9 INSERT INTO Device
10 VALUES (seq_id.nextval, 'Computer');

```

Inserări în tabelul Achizitie

```

1 INSERT INTO Achizitie
2 VALUES (p_get_data.f_id_device('Camera video'), 3, '23-JAN-21', '
          24-JAN-21', 50);
3 INSERT INTO Achizitie
4 VALUES (p_get_data.f_id_device('Lumini'), 3, '23-JAN-21', '24-JAN
          -21', 30);
5 INSERT INTO Achizitie
6 VALUES (p_get_data.f_id_device('Boxa'), 3, '23-JAN-21', '24-JAN-21'
          , 50);
7 INSERT INTO Achizitie
8 VALUES (p_get_data.f_id_device('Microfon'), 3, '23-JAN-21', '24-JAN
          -21', 20);
9 INSERT INTO Achizitie
10 VALUES (p_get_data.f_id_device('Computer'), 3, '23-JAN-21', '24-JAN
          -21', 150);
11 INSERT INTO Achizitie
12 VALUES (p_get_data.f_id_device('Camera video'), p_get_data.
          f_id_spectacol('Tosca'), '13-MAY-21', '14-MAY-21', 50);
13 INSERT INTO Achizitie
14 VALUES (p_get_data.f_id_device('Lumini'), p_get_data.f_id_spectacol
          ('Tosca'), '13-MAY-21', '14-MAY-21', 30);
15 INSERT INTO Achizitie
16 VALUES (p_get_data.f_id_device('Boxa'), p_get_data.f_id_spectacol('
          Tosca'), '13-MAY-21', '14-MAY-21', 50);
17 INSERT INTO Achizitie
18 VALUES (p_get_data.f_id_device('Microfon'), p_get_data.
          f_id_spectacol('Tosca'), '13-MAY-21', '14-MAY-21', 20);
19 INSERT INTO Achizitie
20 VALUES (p_get_data.f_id_device('Computer'), p_get_data.
          f_id_spectacol('Tosca'), '13-MAY-21', '14-MAY-21', 150);

```

Inserări în tabelul Angajat

```

1 INSERT INTO Angajat

```

```

2 VALUES (seq_id.nextval, 'Badea', 'Nicolae', NULL, 'Resurse Umane',
   contacte(contact('0721111111', 'nicolae_badea@gmail.com')));
3 INSERT INTO Angajat
4 VALUES (seq_id.nextval, 'Costin', 'Ana', p_get_data.f_id_angajat('
   Badea', 'Nicolae'), 'Logistica', contacte(contact('0722222222',
   'ana_costin@gmail.com')));
5 INSERT INTO Angajat
6 VALUES (seq_id.nextval, 'Enache', 'Sorin', p_get_data.f_id_angajat(
   'Badea', 'Nicolae'), 'IT', contacte(contact('0744444444', '
   sorin_enache@gmail.com')));
7 INSERT INTO Angajat
8 VALUES (seq_id.nextval, 'David', 'Radu', p_get_data.f_id_angajat('
   Enache', 'Sorin'), 'Promovare', contacte(contact('0733333333',
   'radu_david@gmail.com')));
9 INSERT INTO Angajat
10 VALUES (seq_id.nextval, 'Jurca', 'Traian', p_get_data.f_id_angajat(
   'Badea', 'Nicolae'), 'HoReCa', contacte(contact('0755555555', '
   traian_jurca@gmail.com')));
11 INSERT INTO Angajat
12 VALUES (seq_id.nextval, 'Lunga', 'Eduard', p_get_data.f_id_angajat(
   'Jurca', 'Traian'), 'Fundraising', contacte(contact('0766666666',
   'eduard_lunga@gmail.com')));

```

Inserări în tabelul Sarcină

```

1 INSERT INTO Sarcina
2 VALUES (seq_id.nextval, 1, p_get_data.f_id_angajat('Costin', 'Ana')
   , 'Organizare', 200, NULL);
3 INSERT INTO Sarcina
4 VALUES (seq_id.nextval, 3, p_get_data.f_id_angajat('Costin', 'Ana')
   , 'Organizare', 200, NULL);
5 INSERT INTO Sarcina
6 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Poveste Craciun'
   ), p_get_data.f_id_angajat('Badea', 'Nicolae'), 'Organizare',
   250, NULL);
7 INSERT INTO Sarcina
8 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Lacul lebedelor'
   ), p_get_data.f_id_angajat('Enache', 'Sorin'), 'Imagine', 200,
   NULL);
9 INSERT INTO Sarcina
10 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Lacul lebedelor'
   ), p_get_data.f_id_angajat('David', 'Radu'), 'Coordonare', 200,
   NULL);
11 INSERT INTO Sarcina
12 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Tosca'),
   p_get_data.f_id_angajat('Enache', 'Sorin'), 'Imagine', 200,
   NULL);
13 INSERT INTO Sarcina
14 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Tosca'),
   p_get_data.f_id_angajat('David', 'Radu'), 'Coordonare', 200,
   NULL);
15 INSERT INTO Sarcina
16 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
   p_get_data.f_id_angajat('Costin', 'Ana'), 'Coordonare', 400,
   NULL);
17 INSERT INTO Sarcina
18 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
   p_get_data.f_id_angajat('Enache', 'Sorin'), 'IT', 400, NULL);

```

```

19 INSERT INTO Sarcina
20 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
    p_get_data.f_id_angajat('David', 'Radu'), 'Promovare', 400,
    NULL);
21 INSERT INTO Sarcina
22 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
    p_get_data.f_id_angajat('Lunga', 'Eduard'), 'FR', 400, '
    Comunicare sponsori');
23 INSERT INTO Sarcina
24 VALUES (seq_id.nextval, p_get_data.f_id_spectacol('Festival vara'),
    p_get_data.f_id_angajat('Jurca', 'Traian'), 'HoReCa', 400, '
    Responsabil produse comercializate');

```

Inserări în tabelul Participant

```

1 INSERT INTO Participant
2 VALUES (seq_id.nextval, 'Macovei', 'Gheorghe');
3 INSERT INTO Participant
4 VALUES (seq_id.nextval, 'Lascu', 'Dan');
5 INSERT INTO Participant
6 VALUES (seq_id.nextval, 'Leonida', 'Tudor');
7 INSERT INTO Participant
8 VALUES (seq_id.nextval, 'Olteanu', 'Dan');
9 INSERT INTO Participant
10 VALUES (seq_id.nextval, 'Popa', 'Valentin');
11 INSERT INTO Participant
12 VALUES (seq_id.nextval, 'Popescu', 'Mihaela');
13 INSERT INTO Participant
14 VALUES (seq_id.nextval, 'Sandu', 'Florin');
15 INSERT INTO Participant
16 VALUES (seq_id.nextval, 'Stanescu', 'Constantin');
17 INSERT INTO Participant
18 VALUES (seq_id.nextval, 'Ursea', 'Lucian');

```

Inserări în tabelul Bilet

```

1 INSERT INTO Bilet
2 VALUES(p_get_data.f_id_participant('Macovei', 'Gheorghe'),
    p_get_data.f_id_spectacol('Festival vara'), 250, p_get_data.
    f_numar_bilet(p_get_data.f_id_spectacol('Festival vara')));
3 INSERT INTO Bilet
4 VALUES(p_get_data.f_id_participant('Ursea', 'Lucian'), p_get_data.
    f_id_spectacol('Festival vara'), 250, p_get_data.f_numar_bilet(
    p_get_data.f_id_spectacol('Festival vara')));
5 INSERT INTO Bilet
6 VALUES(p_get_data.f_id_participant('Stanescu', 'Constantin'),
    p_get_data.f_id_spectacol('Festival vara'), 250, p_get_data.
    f_numar_bilet(p_get_data.f_id_spectacol('Festival vara')));
7 INSERT INTO Bilet
8 VALUES(p_get_data.f_id_participant('Sandu', 'Florin'), p_get_data.
    f_id_spectacol('Festival vara'), 250, p_get_data.f_numar_bilet(
    p_get_data.f_id_spectacol('Festival vara')));
9 INSERT INTO Bilet
10 VALUES(p_get_data.f_id_participant('Popescu', 'Mihaela'),
    p_get_data.f_id_spectacol('Festival vara'), 250, p_get_data.
    f_numar_bilet(p_get_data.f_id_spectacol('Festival vara')));
11 INSERT INTO Bilet
12 VALUES(p_get_data.f_id_participant('Popa', 'Valentin'), p_get_data.
    f_id_spectacol('Festival vara'), 250, p_get_data.f_numar_bilet(

```

```

        p_get_data.f_id_spectacol('Festival vara')));
13 INSERT INTO Bilet
14 VALUES(p_get_data.f_id_participant('Popa', 'Valentin'), 1, 30,
        p_get_data.f_numar_bilet(1));
15 INSERT INTO Bilet
16 VALUES(p_get_data.f_id_participant('Lascu', 'Dan'), 3, 30,
        p_get_data.f_numar_bilet(3));
17 INSERT INTO Bilet
18 VALUES(p_get_data.f_id_participant('Leonida', 'Tudor'), p_get_data.
        f_id_spectacol('Poveste Craciun'), 40, p_get_data.f_numar_bilet
        (p_get_data.f_id_spectacol('Poveste Craciun')));
19 INSERT INTO Bilet
20 VALUES(p_get_data.f_id_participant('Olteanu', 'Dan'), p_get_data.
        f_id_spectacol('Poveste Craciun'), 40, p_get_data.f_numar_bilet
        (p_get_data.f_id_spectacol('Poveste Craciun')));
21 INSERT INTO Bilet
22 VALUES(p_get_data.f_id_participant('Popescu', 'Mihaela'),
        p_get_data.f_id_spectacol('Lacul lebedelor'), 45, p_get_data.
        f_numar_bilet(p_get_data.f_id_spectacol('Lacul lebedelor')));
23 INSERT INTO Bilet
24 VALUES(p_get_data.f_id_participant('Sandu', 'Florin'), p_get_data.
        f_id_spectacol('Lacul lebedelor'), 45, p_get_data.f_numar_bilet
        (p_get_data.f_id_spectacol('Lacul lebedelor')));
25 INSERT INTO Bilet
26 VALUES(p_get_data.f_id_participant('Ursea', 'Lucian'), p_get_data.
        f_id_spectacol('Tosca'), 45, p_get_data.f_numar_bilet(
        p_get_data.f_id_spectacol('Tosca')));
27 INSERT INTO Bilet
28 VALUES(p_get_data.f_id_participant('Macovei', 'Gheorghe'),
        p_get_data.f_id_spectacol('Tosca'), 45, p_get_data.
        f_numar_bilet(p_get_data.f_id_spectacol('Tosca')));

1 COMMIT;

```



## 6 Subprogram stocat care utilizează un tip de colecție studiat

Atunci când funcția `p_get_data.f.id_sponsor` aruncă eroarea (-20013, 'Exista mai multi sponsori cu numele dat') putem afla id-ul sponsorului dorit doar văzând toți sponsorii cu acel id. Astfel, în această cerință, afișăm id-ul și datele de contact ale tuturor sponsorii cu un nume dat ca parametru. Procedura `f_afis_sponsor` este inclusă în pachetul `p_cerinte`, despre care am scris și la secțiunea 13.

```
1 PROCEDURE f_afis_sponsor(v_nume sponsor.nume_sponsor%TYPE);

1 PROCEDURE f_afis_sponsor(v_nume sponsor.nume_sponsor%TYPE) IS
2     TYPE tablou_indexat IS TABLE OF Sponsor.id_sponsor%TYPE
3     INDEX BY BINARY_INTEGER;
4     t tablou_indexat;
5     TYPE contact_record IS RECORD (tel VARCHAR(10), mail
6     VARCHAR(30));
7     TYPE tablou_date IS TABLE OF contact_record INDEX BY
8     BINARY_INTEGER;
9     t_date tablou_date;
10    BEGIN
11        SELECT id_sponsor
12        BULK COLLECT INTO t
13        FROM Sponsor
14        WHERE nume_sponsor = v_nume;
15
16        IF t.COUNT = 0 THEN
17            DBMS_OUTPUT.PUT_LINE('Nu a fost gasit niciun sponsor cu
18            acest nume. ');
19        ELSE
20            DBMS_OUTPUT.PUT_LINE('Exista ' || t.count || ' sponsori cu
21            numele ' || v_nume);
22            FOR i IN t.FIRST..t.LAST LOOP
23                SELECT tt.*
24                BULK COLLECT INTO t_date
25                FROM sponsor s, TABLE(s.contact_sponsor) tt
26                WHERE s.id_sponsor = t(i);
27                DBMS_OUTPUT.PUT_LINE('id ' || t(i) || ' - contact:
28                ');
29                FOR j IN t_date.FIRST..t_date.LAST LOOP
30                    DBMS_OUTPUT.PUT_LINE(' telefon: ' || t_date(j)
31                    .tel || ', email: ' || t_date(j).mail);
32                END LOOP;
33                t_date.delete;
34            END LOOP;
35        END IF;
36    END f_afis_sponsor;
```

În această procedură am folosit tablouri indexate cu id-urile sponsorilor, cât și cu record-uri contact. De asemenea, am arătat cum putem lucra cu datele stocate cu ajutorul nested table și object.

```
1 BEGIN
2     p_cerinte.f_afis_sponsor('Softbinator');
```

```

3 END;
4 /

```

```

BEGIN
    p_cerinte.f_afis_sponsor('Softbinator');
END;
/

```

Script Output x

Task completed in 0.036 seconds

```

Exista 1 sponsori cu numele Softbinator
id 9 - contact:
    telefon: 0757404781, email: contact@softbinator.com

PL/SQL procedure successfully completed.

```

## 7 Subprogram stocat care utilizează un tip de cursor studiat

Atunci când funcția `p_get_data.f_id_spectacol` aruncă eroarea(-20004, 'Exista mai multe spectacole cu numele dat. Apeleaza `f_afis_spectacole` pentru a afla care este cel cautat.') putem afla id-ul spectacolului dorit doar văzând toate spectacolele cu acel id. Astfel, în această cerință, afișăm id-ul, perioada și artiștii pentru toate spectacolele cu un nume dat ca parametru. Procedura `f_afis_spectacole` este inclusă în pachetul `p_cerinte`, despre care am scris și la secțiunea 13.

```

1  PROCEDURE f_afis_spectacole(v_nume spectacol.nume_spect%TYPE);

1  PROCEDURE f_afis_spectacole(v_nume spectacol.nume_spect%TYPE)
   IS
2      TYPE refcursor IS REF CURSOR;
3      CURSOR c IS
4          SELECT id_spect, data_inceput, data_sfarsit,
5                 CURSOR(SELECT nume_art, prenume_art, nume_scena
6                     FROM performeaza p join artist a using (
7                     id_art)
8                     WHERE p.id_spect = s.id_spect)
9          FROM spectacol s
10         WHERE nume_spect = v_nume;
11     v_cursor refcursor;
12     v_id spectacol.id_spect%TYPE;
    v_inc spectacol.data_inceput%TYPE;

```

```

13      v_sf spectacol.data_sfarsit%TYPE;
14      v_numa_a artist.numa_art%TYPE;
15      v_pnuma_a artist.pnuma_art%TYPE;
16      v_nume artist.numa_scena%TYPE;
17      exista boolean := false;
18  BEGIN
19      OPEN c;
20      LOOP
21          FETCH c INTO v_id, v_inc, v_sf, v_cursor;
22          EXIT WHEN c%NOTFOUND;
23          exista := true;
24          IF v_inc = v_sf THEN
25              DBMS_OUTPUT.PUT_LINE('La spectacolul cu id-ul ' ||
v_id || ' din data de ' || to_char(v_inc, 'MM/DD/YYYY') || '
participa artistii: ');
26              ELSE
27                  DBMS_OUTPUT.PUT_LINE('La spectacolul cu id-ul ' ||
v_id || ' din perioada ' || v_inc || ' - ' || v_sf || '
participa artistii: ');
28              END IF;
29      LOOP
30          FETCH v_cursor
31          INTO v_numa_a, v_pnuma_a, v_nume;
32          EXIT WHEN v_cursor%NOTFOUND;
33
34          IF v_nume is NULL THEN
35              DBMS_OUTPUT.PUT_LINE(' ' || v_pnuma_a || ' ' ||
v_numa_a);
36          ELSE
37              DBMS_OUTPUT.PUT_LINE(' ' || v_nume);
38          END IF;
39      END LOOP;
40      END LOOP;
41      CLOSE c;
42      IF exista = false THEN
43          DBMS_OUTPUT.PUT_LINE('Nu a fost gasit niciun spectacol cu
acest nume. ');
44      END IF;
45      END f_afis_spectacole;

```

În această procedură am folosit o expresie cursor cu datele despre spectacol și artiști.

```

1 BEGIN
2     p_cerinte.f_afis_spectacole('Comedy Show');
3 END;
4 /

```

```

-- exercitiul 7
BEGIN
    p_cerinte.f_afis_spectacole('Comedy Show');
END;
/

```

Script Output x Query Result x

Task completed in 0.042 seconds

```

La spectacolul cu id-ul 1 din data de 12/16/2020 participa artistii:
    Teo
    Vio
    Costel
La spectacolul cu id-ul 3 din data de 01/23/2021 participa artistii:
    Marius Covache
    Alex Serban

PL/SQL procedure successfully completed.

```

## 8 Funcție care utilizează 3 tabele

Fie o funcție care primește ca parametru id-ul unui spectacol, face un raport al cheltuielilor și câștigurilor și întoarce bilanțul total. Această funcție este `f.costuri_spectacol` și este inclusă în pachetul `p_cerinte`.

```

1  FUNCTION f_costuri_spectacol (v_id spectacol.id_spect%TYPE)
2  RETURN NUMBER;

1  FUNCTION f_costuri_spectacol(v_id spectacol.id_spect%TYPE)
2  RETURN NUMBER IS
3      v_into NUMBER(10, 2);
4      v_res NUMBER(10, 2) := 0;
5      exp exception;
6      exista number;
7  BEGIN
8
9      SELECT count(*)
10         INTO exista
11         FROM Spectacol
12         WHERE id_spect = v_id;
13
14         IF exista = 0 then
15             raise exp;
16         END IF;
17
18         SELECT NVL(SUM(suma), 0)
19            INTO v_into
20            FROM contract
21            WHERE id_spect = v_id;

```

```

22
23         DBMS_OUTPUT.PUT_LINE('Spectacolul a fost sponsorizat cu
suma de ' || v_into || ' RON.');
```

```

24
25         v_res := v_res + v_into;
26
27         SELECT NVL(SUM(salariu), 0)
28         INTO v_into
29         FROM sarcina
30         WHERE id_spect = v_id;
31
32         DBMS_OUTPUT.PUT_LINE('Salariile platite au fost in valoare
de ' || v_into || ' RON.');
```

```

33
34         v_res := v_res - v_into;
35
36         SELECT NVL(SUM(pret), 0)
37         INTO v_into
38         FROM bilet
39         WHERE id_spect = v_id;
40
41         DBMS_OUTPUT.PUT_LINE('Au fost cumparate bilete in valoare
de ' || v_into || ' RON.');
```

```

42
43         v_res := v_res + v_into;
44
45         SELECT NVL(SUM(cost), 0)
46         INTO v_into
47         FROM achizitie
48         WHERE id_spect = v_id;
49
50         DBMS_OUTPUT.PUT_LINE('Au fost achizitionate device-uri in
valoare de ' || v_into || ' RON.');
```

```

51
52         v_res := v_res - v_into;
53         RETURN v_res;
54
55     EXCEPTION
56     WHEN exp THEN RAISE_APPLICATION_ERROR (-20026, 'Nu a fost gasit
niciun spectacol cu acest id.');
```

```

57         WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20023, 'Eroare la
obtinerea costurilor spectacolului.');
```

```

58     END f_costuri_spectacol;
```

Funcția folosește date din 4 tabele: contract, sarcina, bilet, achizitie.

```

1 BEGIN
2     DBMS_OUTPUT.PUT_LINE('Total ' || p_cerinte.f_costuri_spectacol(
p_get_data.f_id_spectacol('Festival vara')));
3 END;
4 /
```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Total ' || p_cerinte.f_costuri_spectacol(p_get_data.f_id_spectacol('Festival vara')));
END;
/

```

Script Output x Query Result x

Task completed in 0.062 seconds

Spectacolul a fost sponsorizat cu suma de 15000 RON.  
 Salariile platite au fost in valoare de 2000 RON.  
 Au fost cumparate bilete in valoare de 1500 RON.  
 Au fost achizitionate device-uri in valoare de 0 RON.  
 Total 14500

PL/SQL procedure successfully completed.

În cazul in care nu există niciun spectacol cu id-ul dat funcția se comporta astfel:

```

1 BEGIN
2     DBMS_OUTPUT.PUT_LINE('Total ' || p_cerinte.f_costuri_spectacol
3     (1000));
4 END;
/

```

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Total ' || p_cerinte.f_costuri_spectacol(1000));
END;
/

```

Script Output x

Task completed in 0.066 seconds

Error starting at line : 962 in command -

```

BEGIN
    DBMS_OUTPUT.PUT_LINE('Total ' || p_cerinte.f_costuri_spectacol(1000));
END;

```

Error report -

ORA-20026: Nu a fost gasit niciun spectacol cu acest id.  
 ORA-06512: at "USER2.P\_CERINTE", line 137  
 ORA-06512: at line 2

## 9 Procedură care utilizează 5 tabele

Fie o procedură care afișează toate informațiile despre un spectacol de care ar avea nevoie un posibil participant - când și unde se desfășoară, artiștii care iau parte, programul detaliat și, bineînțeles, sponsorii vor fi și ei afișați. Această procedură este f\_more\_info\_spectacol și este inclusă în pachetul p\_cerinte.

```

1 PROCEDURE f_more_info_spectacol(v_id spectacol.id_spect%TYPE);

1 PROCEDURE f_more_info_spectacol(v_id spectacol.id_spect%TYPE)
  IS
2     TYPE artist_record IS RECORD (
3         nume artist.nume_art%TYPE,

```

```

4         prenume_artist.prenume_art%TYPE,
5         scena_artist.nume_scena%TYPE);
6     TYPE tablou_indexat_a IS TABLE OF artist_record INDEX BY
    BINARY_INTEGER;
7     t_art tablou_indexat_a;
8
9     v_num_s spectacol.nume_spect%TYPE;
10    v_data_i spectacol.data_inceput%TYPE;
11    v_data_sf spectacol.data_sfarsit%TYPE;
12
13    TYPE tablou_indexat_s IS TABLE OF sponsor.nume_sponsor%TYPE
    INDEX BY BINARY_INTEGER;
14    t_spon tablou_indexat_s;
15
16    TYPE program_record IS RECORD (
17        descriere program.descriere%TYPE,
18        t_i program.timp_inc%TYPE,
19        t_s program.timp_sf%TYPE);
20    TYPE tablou_indexat_p IS TABLE OF program_record INDEX BY
    BINARY_INTEGER;
21    t_prog tablou_indexat_p;
22
23    TYPE tablou_indexat_l IS TABLE OF locatie.nume_locatie%TYPE
    INDEX BY BINARY_INTEGER;
24    t_loc tablou_indexat_l;
25    BEGIN
26        SELECT nume_spect, data_inceput, data_sfarsit
27        INTO v_num_s, v_data_i, v_data_sf
28        FROM spectacol
29        WHERE id_spect = v_id;
30
31        IF v_data_i = v_data_sf THEN
32            DBMS_OUTPUT.PUT_LINE('Spectacolul ' || v_num_s || '
are loc pe ' || to_char(v_data_i, 'MM/DD/YYYY'));
33        ELSE
34            DBMS_OUTPUT.PUT_LINE('Spectacolul ' || v_num_s || '
are loc pe ' || v_data_i || ' - ' || v_data_sf);
35        END IF;
36
37        SELECT nume_locatie
38        BULK COLLECT INTO t_loc
39        FROM locatie join desfasurare using (id_locatie)
40        WHERE id_spect = v_id;
41
42        IF t_loc.count = 1 THEN
43            DBMS_OUTPUT.PUT_LINE('la ' || t_loc(1));
44        ELSE
45            DBMS_OUTPUT.PUT_LINE('in locatiile: ');
46
47            FOR i IN t_loc.FIRST..t_loc.LAST LOOP
48                DBMS_OUTPUT.PUT(t_loc(i) || ' ');
49            END LOOP;
50            DBMS_OUTPUT.PUT_LINE('');
51        END IF;
52
53        SELECT nume_art, prenume_art, nume_scena
54        BULK COLLECT INTO t_art

```

```

55     FROM artist join performeaza using (id_art)
56     WHERE id_spect = v_id;
57
58     DBMS_OUTPUT.PUT('La spectacol participa artistii ');
59     FOR i IN t_art.FIRST..t_art.LAST LOOP
60         IF t_art(i).scena is NULL THEN
61             DBMS_OUTPUT.PUT(t_art(i).prenume || ' ' || t_art
(i).nume || ' ');
62         ELSE
63             DBMS_OUTPUT.PUT(t_art(i).scena || ' ');
64         END IF;
65     END LOOP;
66     DBMS_OUTPUT.PUT_LINE('');
67
68     SELECT descriere, timp_inc, timp_sf
69     BULK COLLECT INTO t_prog
70     FROM program
71     WHERE id_spect = v_id;
72
73     DBMS_OUTPUT.PUT('cu programul ');
74     FOR i IN t_prog.FIRST..t_prog.LAST LOOP
75         DBMS_OUTPUT.PUT_LINE(t_prog(i).descriere || ' ' ||
to_char(t_prog(i).t_i, 'HH24:MI') || ' - ' || to_char(t_prog(i)
.t_s, 'HH24:MI') || ' ');
76     END LOOP;
77
78     SELECT nume_sponsor
79     BULK COLLECT INTO t_spon
80     FROM sponsor join contract using(id_sponsor)
81     WHERE id_spect = v_id
82     ORDER BY suma DESC;
83
84     DBMS_OUTPUT.PUT('Spectacolul este sponsorizat de ');
85     FOR i IN t_spon.FIRST..t_spon.LAST LOOP
86         DBMS_OUTPUT.PUT_LINE(t_spon(i) || ' ');
87     END LOOP;
88
89     EXCEPTION
90     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20024, 'Nu
exista spectacol cu acest id. ');
91     WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20025, 'Eroare la
obtinerea informatiilor. ');
92     END f_more_info_spectacol;

```

Procedura utilizează 8 tabele: spectacol, locație, desfășurare, artist, performează, program, sponsor, contract și stochează date în tabele imbricate. De asemenea am folosit structuri de tip record. Am extras ora și minutul din timestamp pentru o afișare mai frumoasă.

```

1 BEGIN
2     p_cerinte.f_more_info_spectacol(p_get_data.f_id_spectacol('
Festival vara'));
3 END;
4 /

```



```

942 BEGIN
943     p_cerinte.f_more_info_spectacol(p_get_data.f_id_spectacol('Festival vara'));
944 END;
945 /

```

Script Output x Query Result x

Task completed in 0.047 seconds

Spectacolul Festival vara are loc pe 13-AUG-21 - 15-AUG-21  
 in locatiile:  
 Stirbei sala mare Stirbei sala mica Stirbei pavilion Stirbei gradina Stirbei domeniu  
 La spectacol participa artistii Birdy Ntg but tvs Roosevelt The 1975 David August  
 cu programul Roosevelt 20:00 - 21:30  
 Nothing but Thieves 21:30 - 23:00  
 Birdy 20:00 - 22:00  
 The 1975 20:00 - 21:30  
 August David 21:30 - 23:00  
 Spectacolul este sponsorizat de Aqua Carpatica  
 Kiss FM  
 Pepsi

PL/SQL procedure successfully completed.

În cazul in care nu există niciun spectacol cu id-ul dat funcția se comporta astfel:

```

1 BEGIN
2     p_cerinte.f_more_info_spectacol(1000);
3 END;
4 /

```

```

BEGIN
    p_cerinte.f_more_info_spectacol(1000);
END;
/

```

Script Output x

Task completed in 0.055 seconds

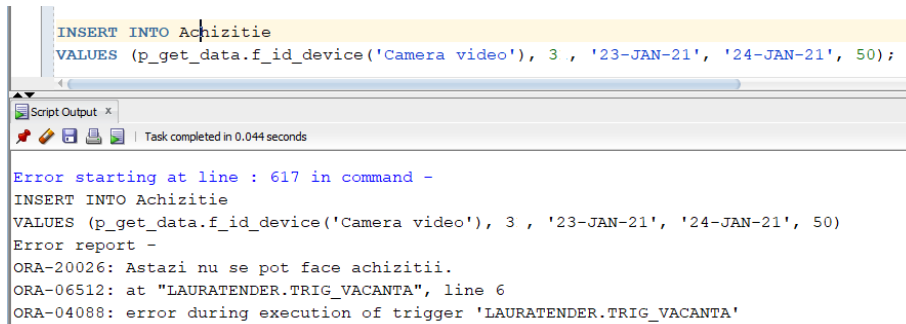
Error starting at line : 974 in command -  
 BEGIN  
 p\_cerinte.f\_more\_info\_spectacol(1000);  
 END;  
 Error report -  
 ORA-20024: Nu exista spectacol cu acest id.  
 ORA-06512: at "USER2.P\_CERINTE", line 231  
 ORA-06512: at line 2

## 10 Trigger LMD la nivel de comanda

Fie un trigger care nu permite achizițiile în prima zi de Crăciun și în prima zi din an. Astfel, înainte de inserarea în tabelul achiziție, este aruncată o eroare dacă ziua este una dintre cele menționate.

```
1 CREATE OR REPLACE TRIGGER trig_vacanta
2   BEFORE INSERT ON achizitie
3 DECLARE
4   zi NUMBER := EXTRACT(DAY FROM sysdate);
5   luna NUMBER := EXTRACT(MONTH FROM sysdate);
6 BEGIN
7   IF (zi = 25 and luna = 12) or (zi = 1 and luna = 1) THEN
8     RAISE_APPLICATION_ERROR(-20026, 'Astazi nu se pot face
9     achizitii. ');
10  END IF;
11 END;
```

```
1 INSERT INTO Achizitie
2 VALUES (p_get_data.f_id_device('Camera video'), 3 , '23-JAN-21', '
    24-JAN-21', 50);
```



## 11 Trigger LMD la nivel de linie

Fie un trigger care verifică că timpul din program este corect (timpul de început este mai mic decât cel de final) și că acesta corespunde cu perioada în care se desfășoară spectacolul.

```
1 CREATE OR REPLACE TRIGGER trig_insert_program
2   BEFORE INSERT ON Program
3   FOR EACH ROW
4 DECLARE
5   v_inc spectacol.data_inceput%TYPE;
6   v_sf spectacol.data_sfarsit%TYPE;
```

```

7      exceptie EXCEPTION;
8 BEGIN
9     SELECT s.data_inceput, s.data_sfarsit
10    INTO v_inc, v_sf
11   FROM spectacol s
12  WHERE s.id_spect = :NEW.id_spect;
13
14     IF (NOT((:NEW.timp_inc > TRUNC(v_inc)) and (:NEW.timp_inc < :
15     NEW.timp_sf) and (:NEW.timp_sf < (TRUNC(v_sf) + INTERVAL '1'
16     DAY - INTERVAL '1' SECOND))))
17     THEN RAISE exceptie;
18   END IF;
19 EXCEPTION
20   WHEN EXCEPTIE THEN
21     RAISE_APPLICATION_ERROR(-20025,'Programul nu este corect.')
```

În acest trigger am definit o excepție proprie, am comparat variabile cu tipurile de date DATE și TIMESTAMP cu ajutorul funcției TRUNC și am obținut finalul zilei (TRUNC(v\_sf) + INTERVAL '1' DAY - INTERVAL '1' SECOND).

```

1 INSERT INTO Program
2 VALUES (seq_id.nextval, 3, 'Stand-up Comedy', timestamp '2020-12-29
    19:00:00', timestamp '2020-12-29 21:00:00');
```

```

-- exercitiul 11
INSERT INTO Program
VALUES (seq_id.nextval, 3, 'Stand-up Comedy', timestamp '2020-12-29 19:00:00', timestamp '2020-12-29 21:00:00');

Error starting at line : 984 in command -
INSERT INTO Program
VALUES (seq_id.nextval, 3, 'Stand-up Comedy', timestamp '2020-12-29 19:00:00', timestamp '2020-12-29 21:00:00')
Error report -
ORA-20025: Programul nu este corect.
ORA-06512: at "USER2.TRIG_INSERT_PROGRAM", line 16
ORA-04088: error during execution of trigger 'USER2.TRIG_INSERT_PROGRAM'
```

Show-ul cu id-ul 3 are loc pe 31 Ianuarie, iar programul nu corespundea.

## 12 Trigger LDD

Fie un trigger care să nu permită efectuarea de comezi LDD decât pentru utilizatorul LAURATENDER.

```

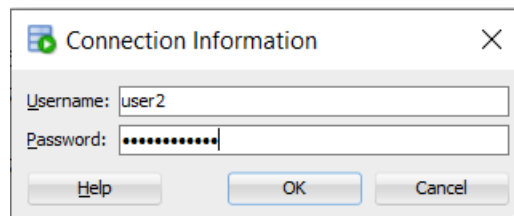
1 CREATE OR REPLACE TRIGGER trig_LDD_owner
2 BEFORE CREATE OR DROP OR ALTER ON SCHEMA
3 BEGIN
4     IF USER != UPPER('LAURATENDER') THEN
5     RAISE_APPLICATION_ERROR(-20025,'Nu ai drepturi pentru
6     comenzi LDD.');
```

```
7 END;  
8 /
```

Am creat un nou utilizator căruia pentru rapiditate i-am oferit toate drepturile.

```
SQL> connect sys as sysdba  
Enter password:  
Connected.  
  
SQL> create user user2 identified by testpassword;  
  
User created.  
  
SQL> grant all privileges to user2;  
  
Grant succeeded.  
  
SQL>
```

M-am conectat cu userul nou creat.



```
1 CREATE TABLE Test(  
2     id INT PRIMARY KEY,  
3     nume VARCHAR(15) NOT NULL  
4 );
```

```
Error starting at line : 635 in command -  
CREATE TABLE Test(  
    id INT PRIMARY KEY,  
    nume VARCHAR(15) NOT NULL  
)  
Error report -  
ORA-00604: error occurred at recursive SQL level 1  
ORA-20025: Nu ai drepturi pentru comenzi LDD.  
ORA-06512: at line 3  
00604. 00000 - "error occurred at recursive SQL level %s"  
*Cause:      An error occurred while processing a recursive SQL statement  
              (a statement applying to internal dictionary tables).  
*Action:     If the situation described in the next error on the stack  
              can be corrected, do so; otherwise contact Oracle Support.
```

## 13 Pachetul p\_cerinte

```
1 CREATE OR REPLACE PACKAGE p_cerinte AS
2     -- exercitiul 6
3     PROCEDURE f_afis_sponsor(v_numa sponsor.nume_sponsor%TYPE);
4     -- exercitiul 7
5     PROCEDURE f_afis_spectacole(v_numa spectacol.nume_spect%TYPE);
6     --exercitiul 8
7     FUNCTION f_costuri_spectacol (v_id spectacol.id_spect%TYPE)
8     RETURN NUMBER;
9     --exercitiul 9
10    PROCEDURE f_more_info_spectacol(v_id spectacol.id_spect%TYPE);
11 END;
12 /
```

Corpul pachetului curpinde subprogramele de la cerințele 6-9. Nu voi adăuga aici integral corpul pachetului pentru a menține documentul cât mai succint și mai ușor de urmărit. Acesta poate fi găsit și în fișierul sursă al proiectului.

```
1 CREATE OR REPLACE PACKAGE BODY p_cerinte AS
2 -- ...
3 END p_cerinte;
4 /
```

## 14 Pachetul p\_get\_data

În acest pachet se află funcții și proceduri ajutătoare implementate în afară cerințelor pentru a îmbunătăți programul. În comentarii voi descrie ce face fiecare.

```
1 CREATE OR REPLACE PACKAGE p_get_data AS
2     -- obtine id-ul unui spectacol cu numele dat
3     FUNCTION f_id_spectacol(v_numa spectacol.nume_spect%TYPE)
4     RETURN spectacol.id_spect%type;
5
6     -- obtine id-ul unei locatii cu numele dat
7     FUNCTION f_id_locatie(v_numa locatie.nume_locatie%TYPE)
8     RETURN locatie.id_locatie%type;
9
10    -- obtine id-ul unui artist cu numele si prenumele date ca
    parametree
11    FUNCTION f_id_artist(v_numa artist.nume_art%TYPE, v_prenume
    artist.prenume_art%TYPE)
12    RETURN artist.id_art%type;
13
14    -- overload pe functia f_id_artist
15    -- obtine id-ul unui artist cu numele de scena dat ca parametru
16    FUNCTION f_id_artist (v_numa artist.nume_scena%TYPE)
17    RETURN artist.id_art%type;
18
19    -- obtine id-ul unui sponsor cu numele dat
20    FUNCTION f_id_sponsor(v_numa sponsor.nume_sponsor%TYPE)
21    RETURN sponsor.id_sponsor%type;
```

```

22
23 -- obtine id-ul unui device cu numele dat
24 FUNCTION f_id_device(v_nume device.nume_device%TYPE)
25     RETURN device.id_device%type;
26
27 -- obtine id-ul unui angajat cu numele si prenumele date ca
    parametrii
28 FUNCTION f_id_angajat(v_nume angajat.nume_ang%TYPE, v_prenume
    angajat.prenume_ang%TYPE)
29     RETURN angajat.id_ang%type;
30
31 -- obtine id-ul unui participant cu numele si prenumele date
    ca parametrii
32 FUNCTION f_id_participant(v_nume participant.nume_part%TYPE,
    v_prenume participant.prenume_part%TYPE)
33     RETURN participant.id_part%type;
34
35 -- afiseaza informatii despre toate locatiile cu numele dat ca
    parametru
36 PROCEDURE f_afis_locatii(v_nume locatie.nume_locatie%TYPE);
37
38 -- genereaza numarul biletului
39 FUNCTION f_numar_bilet(v_id spectacol.id_spect%TYPE)
40     RETURN bilet.numar_bilet%type;
41 END p_get_data;
42 /

1 CREATE OR REPLACE PACKAGE BODY p_get_data AS
2
3     FUNCTION f_id_spectacol(v_nume spectacol.nume_spect%TYPE)
4     RETURN spectacol.id_spect%type IS nume_s spectacol.id_spect%
    type;
5     BEGIN
6         SELECT id_spect
7         INTO nume_s
8         FROM spectacol
9         WHERE nume_spect = v_nume;
10        RETURN nume_s;
11    EXCEPTION
12        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20003, 'Nu
        exista spectacol cu numele dat. Verifica daca numele introdus
        este corect. ');
13        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20004, '
        Exista mai multe spectacole cu numele dat. Apeleaza
        f_afis_spectacole pentru a afla care este cel cautat. ');
14        WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20005, 'Eroare la
        obtinerea id-ului spectacolului. ');
15    END f_id_spectacol;
16
17
18    FUNCTION f_id_locatie(v_nume locatie.nume_locatie%TYPE)
19    RETURN locatie.id_locatie%type IS nume_l locatie.id_locatie%
    type;
20    BEGIN
21        SELECT id_locatie
22        INTO nume_l
23        FROM locatie
24        WHERE nume_locatie = v_nume;

```

```

25     RETURN nume_l;
26 EXCEPTION
27     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20006, 'Nu
    exista locatie cu numele dat. ');
28     WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20007, '
    Exista mai multe locatii cu numele dat ');
29     WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20008, 'Eroare la
    obtinerea id-ului locatiei. ');
30 END f_id_locatie;
31
32
33 FUNCTION f_id_artist(v_numa artist.nume_art%TYPE, v_prename
    artist.prenume_art%TYPE)
34 RETURN artist.id_art%type IS nume_a artist.id_art%type;
35 BEGIN
36     SELECT id_art
37     INTO nume_a
38     FROM artist
39     WHERE nume_art = v_numa AND prenume_art = v_prename;
40     RETURN nume_a;
41 EXCEPTION
42     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20009, 'Nu
    exista artist cu acest numele ');
43     WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20010, '
    Exista mai multi artisti cu acest nume ');
44     WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20011, 'Eroare la
    obtinerea id-ului artistului. ');
45 END f_id_artist;
46
47 FUNCTION f_id_artist (v_numa artist.nume_scena%TYPE)
48 RETURN artist.id_art%type IS nume_a artist.id_art%type;
49 BEGIN
50     SELECT id_art
51     INTO nume_a
52     FROM artist
53     WHERE nume_scena = v_numa;
54     RETURN nume_a;
55 EXCEPTION
56     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20009, 'Nu
    exista artist cu acest numele de scena. ');
57     WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20010, '
    Exista mai multi artisti cu acest nume de scena ');
58     WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20011, 'Eroare la
    obtinerea id-ului artistului. ');
59 END f_id_artist;
60
61 FUNCTION f_id_sponsor(v_numa sponsor.nume_sponsor%TYPE)
62 RETURN sponsor.id_sponsor%type IS nume_s sponsor.id_sponsor%
    type;
63 BEGIN
64     SELECT id_sponsor
65     INTO nume_s
66     FROM sponsor
67     WHERE nume_sponsor = v_numa;
68     RETURN nume_s;
69 EXCEPTION
70     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20012, 'Nu

```

```

71      exista sponsor cu numele dat. ');
      WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20013, '
72      Exista mai multi sponsori cu numele dat');
      WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20014, 'Eroare la
      obtinerea id-ului sponsorului. ');
73  END f_id_sponsor;
74
75  FUNCTION f_id_device(v_num device.nume_device%TYPE)
76      RETURN device.id_device%TYPE IS id_d device.id_device%TYPE;
77  BEGIN
78      SELECT id_device
79      INTO id_d
80      FROM device
81      WHERE nume_device = v_num;
82      RETURN id_d;
83  EXCEPTION
84      WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20016, 'Nu
      exista device cu numele dat. ');
85      WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20017, '
      Exista mai multe device-uri cu numele dat. ');
86      WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20018, 'Eroare la
      obtinerea id-ului device-ului. ');
87  END f_id_device;
88
89  FUNCTION f_id_angajat(v_num angajat.nume_ang%TYPE, v_prenume
      angajat.prenume_ang%TYPE)
90      RETURN angajat.id_ang%type IS nume_a angajat.id_ang%type;
91  BEGIN
92      SELECT id_ang
93      INTO nume_a
94      FROM angajat
95      WHERE nume_ang = v_num AND prenume_ang = v_prenume;
96      RETURN nume_a;
97  EXCEPTION
98      WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20019, 'Nu
      exista angajat cu acest nume. ');
99      WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20020, '
      Exista mai multi angajati cu acest nume. ');
100     WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20021, 'Eroare la
      obtinerea id-ului angajatului. ');
101  END f_id_angajat;
102
103  FUNCTION f_id_participant(v_num participant.nume_part%TYPE,
      v_prenume participant.prenume_part%TYPE)
104      RETURN participant.id_part%type IS nume_p participant.id_part%
      type;
105  BEGIN
106      SELECT id_part
107      INTO nume_p
108      FROM participant
109      WHERE nume_part = v_num AND prenume_part = v_prenume;
110      RETURN nume_p;
111  EXCEPTION
112      WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20021, 'Nu
      exista participant cu acest nume. ');
113      WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20022, '
      Exista mai multi participanti cu acest nume. ');

```



```

114         WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20023,'Eroare la
115         obtinerea id-ului participantului.');
```

```

116     END f_id_participant;
117
118     PROCEDURE f_afis_locatii(v_nume locatie.nume_locatie%TYPE) IS
119         TYPE tablou_indexat IS TABLE OF Locatie%ROWTYPE INDEX BY
120         BINARY_INTEGER;
121         t tablou_indexat;
122     BEGIN
123         SELECT *
124         BULK COLLECT INTO t
125         FROM Locatie
126         WHERE nume_locatie = v_nume;
127
128         IF t.COUNT = 0 THEN
129             DBMS_OUTPUT.PUT_LINE('Nu a fost gasita nicio locatie cu
130             acest nume.');
```

```

131         ELSE
132             DBMS_OUTPUT.PUT_LINE('Exista ' || t.count || ' locatii cu
133             numele: ' || v_nume);
134             FOR i IN t.FIRST..t.LAST LOOP
135                 DBMS_OUTPUT.PUT_LINE('id ' || t(i).id_locatie || '
136                 ' || t(i).oras || ' ' || t(i).judet || ' ' || t(i).adresa);
137             END LOOP;
138         END IF;
139     END f_afis_locatii;
140
141     FUNCTION f_numar_bilet(v_id spectacol.id_spect%TYPE)
142     RETURN bilet.numar_bilet%type IS v_nr bilet.numar_bilet%
143     type;
144     BEGIN
145         SELECT max(numar_bilet) + 1
146         INTO v_nr
147         FROM bilet
148         where id_spect = v_id;
149         IF v_nr is NULL THEN
150             RETURN 1;
151         ELSE
152             RETURN v_nr;
153         END IF;
154     END f_numar_bilet;
155
156 END p_get_data;
157 /
```

## 15 Concluzii

Mi-am dorit ca în acest proiect să modelez baza de date pentru organizarea de spectacole, să creez un schelet care poate fi dezvoltat pentru o aplicație mai complexă și să abordez niște cerințe utile pentru adaptabilitatea proiectului și utilizarea bazei de date cu sens pentru a obține informații importante integrând cunoștințele acumulate pe parcursul semestrului și alte lucruri pe care am fost îndrumată să le caut singură și să le folosesc pentru un proiect mai performant.