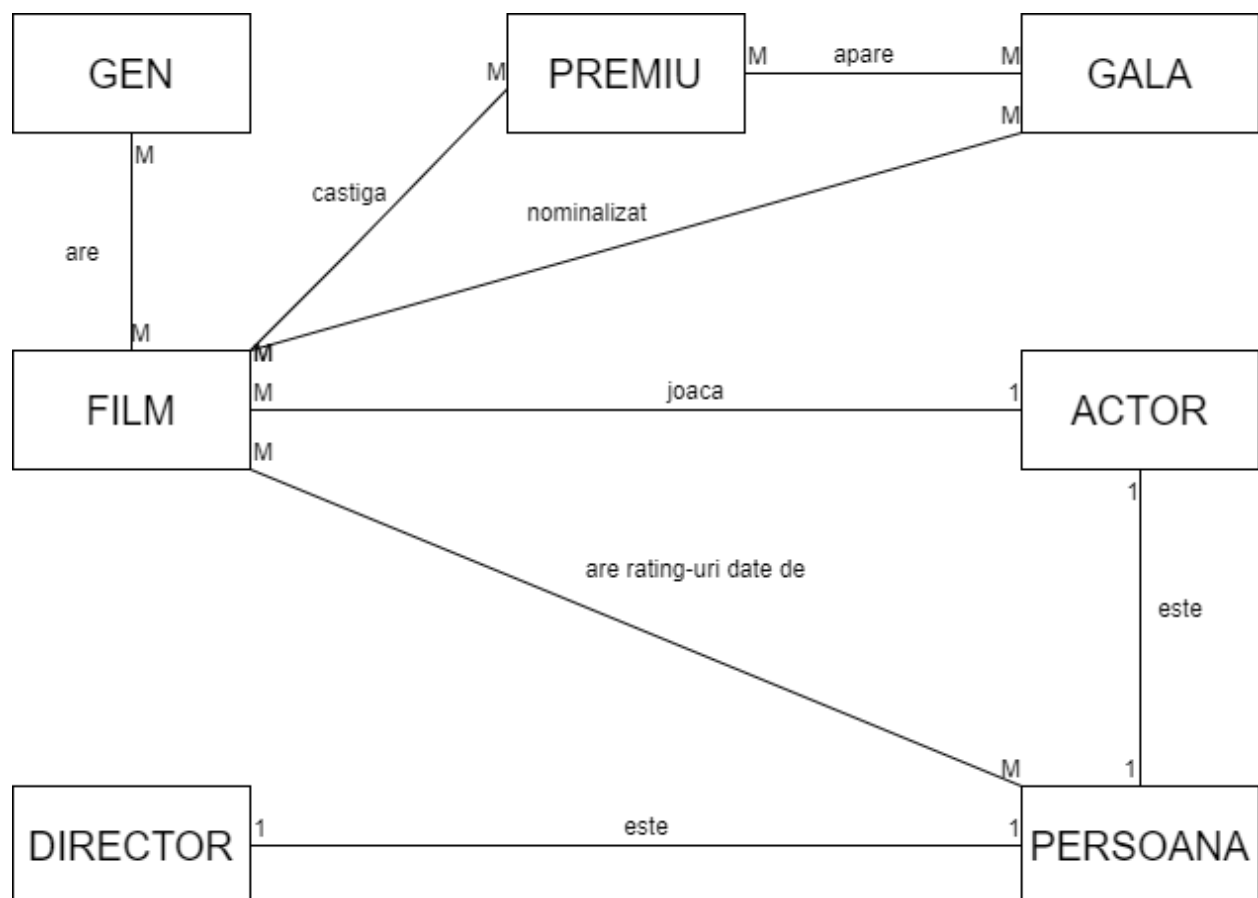


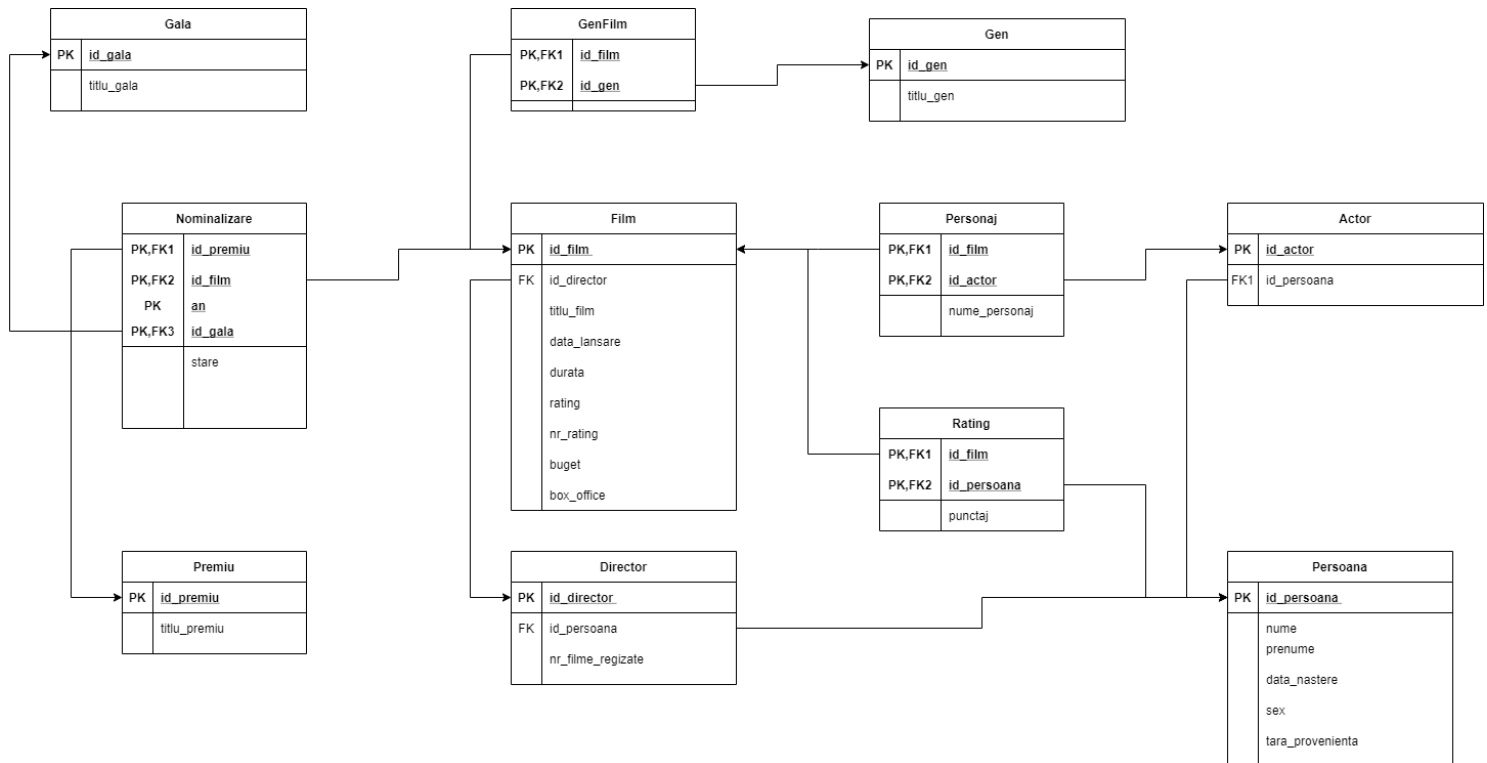
1. Prezentați pe scurt baza de date (utilitatea ei).

Baza mea de date tine evidenta filmelor. Un film poate fi incadrat in mai multe genuri si unui gen ii apartin o varietate de filme. In mod asemanator, un film poate fi nominalizat pentru mai multe premii iar un premiu este dat mai multor filme (in functie de an),toate caestea petrecandu-se intr-o anumita gala de premii. Intr-un film apar mai multi actori, fiecare interpretand un personaj, iar un actor poate aparea in mai multe filme. Un film are un singur director. De asemenea, filmul are o multime de ratinguri date de persoane, care la randul lor pot da rating mai multor filme.

2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```

CREATE TABLE Persoana (
    id_persoana number(4),
    nume varchar(20) NOT NULL,
    prenume varchar(20) NOT NULL,
    data_nastere date NOT NULL,
    sex char(1) NOT NULL,

```

```
tara_provenienta varchar(30),  
  
CONSTRAINT PK_Persoana PRIMARY KEY (id_persoana)  
);
```

```
CREATE TABLE Actor (  
  
    id_actor number(4),  
  
    id_persoana number(4) NOT NULL UNIQUE,  
  
    CONSTRAINT PK_Actor PRIMARY KEY (id_actor),  
  
    CONSTRAINT FK_Actor FOREIGN KEY (id_persoana) REFERENCES  
    Persoana(id_persoana) ON DELETE CASCADE  
);
```

```
CREATE TABLE Director (  
  
    id_director number(4),  
  
    id_persoana number(4) NOT NULL UNIQUE,  
  
    nr_filme_regizate number(2),  
  
    CONSTRAINT PK_Director PRIMARY KEY (id_director),  
  
    CONSTRAINT FK_Director FOREIGN KEY (id_persoana) REFERENCES  
    Persoana(id_persoana) ON DELETE CASCADE  
);
```

```
CREATE TABLE Film (  
  
    id_film number(4),  
  
    id_director number(4) NOT NULL,  
  
    titlu_film varchar(50) NOT NULL,  
  
    data_lansare date NOT NULL,
```

```
durata number(4),
rating number(4,2),
nr_rating number(2),
buget number(4,1),
box_office number(4,1),
CONSTRAINT PK_Film PRIMARY KEY (id_film),
CONSTRAINT FK_FilmDirector FOREIGN KEY (id_director) REFERENCES
Director(id_director) ON DELETE SET NULL
);
```

```
CREATE TABLE Personaj (
    id_film number(4),
    id_actor number(4),
    nume_personaj varchar(30),
    CONSTRAINT PK_Personaj PRIMARY KEY (id_film,id_actor),
    CONSTRAINT FK_ActorFilm FOREIGN KEY (id_film) REFERENCES Film(id_film) ON
DELETE CASCADE,
    CONSTRAINT FK_ActorFilm2 FOREIGN KEY (id_actor) REFERENCES Actor(id_actor)
ON DELETE SET NULL
);
```

```
CREATE TABLE Rating (
    id_film number(4),
    id_persoana number(4),
    punctaj number(4,2) NOT NULL,
    CONSTRAINT PK_Rating PRIMARY KEY (id_film,id_persoana),
```

CONSTRAINT FK_RatingFilm FOREIGN KEY (id_film) REFERENCES Film(id_film) ON DELETE CASCADE,

CONSTRAINT FK_RatingFilm2 FOREIGN KEY (id_persoana) REFERENCES Persoana(id_persoana) ON DELETE SET NULL

);

CREATE TABLE Gen (

id_gen number(4),

titlu_gen varchar(30) NOT NULL,

CONSTRAINT PK_Gen PRIMARY KEY (id_gen)

);

CREATE TABLE GenFilm (

id_film number(4),

id_gen number(4),

CONSTRAINT PK_GenFilm PRIMARY KEY (id_film,id_gen),

CONSTRAINT FK_GenFilm FOREIGN KEY (id_film) REFERENCES Film(id_film) ON DELETE CASCADE,

CONSTRAINT FK_GenFilm2 FOREIGN KEY (id_gen) REFERENCES Gen(id_gen) ON DELETE CASCADE

);

CREATE TABLE Gala (

id_gala number(4),

titlu_gala varchar(30) NOT NULL,

CONSTRAINT PK_Gala PRIMARY KEY (id_gala)

);

```
CREATE TABLE Premiu (  
    id_premiu number(4),  
    titlu_premiu varchar(30) NOT NULL,  
    CONSTRAINT PK_Premiu PRIMARY KEY (id_premiu)  
);
```

```
CREATE TABLE Nominalizare (  
    id_premiu number(4),  
    id_film number(4),  
    id_gala number(4),  
    an number(4),  
    stare varchar(20),  
    CONSTRAINT PK_Nominalizare PRIMARY KEY (id_film,id_premiu,id_gala,an),  
    CONSTRAINT FK_Nominalizare FOREIGN KEY (id_film) REFERENCES Film(id_film)  
ON DELETE CASCADE,  
    CONSTRAINT FK_Nominalizare2 FOREIGN KEY (id_gala) REFERENCES Gala(id_gala)  
ON DELETE CASCADE,  
    CONSTRAINT FK_Nominalizare3 FOREIGN KEY (id_premiu) REFERENCES  
Premiu(id_premiu) ON DELETE CASCADE  
);
```



Live SQL

SQL Worksheet

```
1 CREATE TABLE Persoana (  
2     id_persoana number(4),  
3     nume varchar(20) NOT NULL,  
4     prenume varchar(20) NOT NULL,  
5     data_nastere date NOT NULL,  
6     sex char(1) NOT NULL,  
7     tara_provenienta varchar(30),  
8     CONSTRAINT PK_Persoana PRIMARY KEY (id_persoana)  
9 );  
10  
11 CREATE TABLE Actor (  
12     id_actor number(4),  
13     id_persoana number(4) NOT NULL UNIQUE,  
14     CONSTRAINT PK_Actor PRIMARY KEY (id_actor),  
15     CONSTRAINT FK_Actor FOREIGN KEY (id_persoana) REFERENCES Persoana(id_persoana) ON DELETE CASCADE  
16 );  
17
```

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

INSERT INTO gen

VALUES (50,'Drama');

INSERT INTO gen

VALUES (51,'Actiune');

```
INSERT INTO gen  
VALUES (52,'Comedie');
```

```
INSERT INTO gen  
VALUES (53,'SF');
```

```
INSERT INTO gen  
VALUES (54,'Horror');
```

```
INSERT INTO gen  
VALUES (55,'Romantic');
```

```
INSERT INTO gen  
VALUES (56,'Aventura');
```

```
INSERT INTO gen  
VALUES (57,'Thriller');
```

```
INSERT INTO gen  
VALUES (58,'Fictiune');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```



```
VALUES (100,'Brad','Pitt',TO_DATE('1975/05/03', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (101,'Morgan','Freeman',TO_DATE('1958/02/18', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (102,'Will','Smith',TO_DATE('1978/02/13', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (103,'Leonardo','DiCaprio',TO_DATE('1965/05/20', 'yyyy/mm/dd'),'M','Italia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (104,'Ryan','Gosling',TO_DATE('1968/09/18', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (105,'Ryan','Reynolds',TO_DATE('1978/10/18', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (106,'Jim','Carrey',TO_DATE('1958/12/25', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (107,'Dorian','Popa',TO_DATE('1988/02/28', 'yyyy/mm/dd'),'M','Romania');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)  
VALUES (108,'Robert','Downey',TO_DATE('1968/01/07', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (109,'Tom','Cruise',TO_DATE('1958/02/18', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (110,'Quentin','Tarantino',TO_DATE('1948/04/27', 'yyyy/mm/dd'),'M','Italia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (111,'Martin','Scorsese',TO_DATE('1948/01/18', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (112,'John','Karev',TO_DATE('1988/09/07', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (113,'Finn','Hudson',TO_DATE('1978/03/20', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (114,'Alex','Karev',TO_DATE('1968/09/18', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (115,'Scarlet','Johansson',TO_DATE('1990/01/05', 'yyyy/mm/dd'),'F','Rusia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (116,'Jeniffer','Lawrence',TO_DATE('1984/11/18', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (117,'Meryl','Streep',TO_DATE('1948/09/29', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (118,'Natalie','Portman',TO_DATE('1981/10/12', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (119,'Anne','Hathaway',TO_DATE('1980/01/08', 'yyyy/mm/dd'),'F','Anglia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (120,'Emma','Watson',TO_DATE('1981/09/17', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (121,'Emma','Roberts',TO_DATE('1905/12/30', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (122,'Emma','Stone',TO_DATE('1986/09/18', 'yyyy/mm/dd'),'F','Italia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (123,'Angelina','Jolie',TO_DATE('1968/01/01', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (124,'Kate','Winslet',TO_DATE('1971/09/28', 'yyyy/mm/dd'),'F','Anglia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (150,'Stoica','Vlad',TO_DATE('1995/05/03', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (151,'Chiva','Robert',TO_DATE('1958/10/23', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (152,'Baciu','Paul',TO_DATE('1978/02/13', 'yyyy/mm/dd'),'M','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (153,'Ionita','Andrei',TO_DATE('1965/05/20', 'yyyy/mm/dd'),'M','Italia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (154,'Colesiu','Bogdan',TO_DATE('1968/09/18', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (155,'Diaconu','Gabriel',TO_DATE('1978/10/18', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (156,'Toader','Andrei',TO_DATE('1958/12/25', 'yyyy/mm/dd'),'M','Canada');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (157,'Ilie','Andreea',TO_DATE('1948/09/29', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
```

```
VALUES (158,'Bulin','Diana',TO_DATE('1981/10/12', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (159,'Ciobanu','Maria',TO_DATE('1980/01/08', 'yyyy/mm/dd'),'F','Anglia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (160,'Ciobanu','Ioana',TO_DATE('1981/09/17', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (161,'Negroiu','Gabriela',TO_DATE('1905/12/30', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (162,'Raicu','Robertina',TO_DATE('1986/09/18', 'yyyy/mm/dd'),'F','Italia');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (163,'Ene','Mihaela',TO_DATE('1968/01/01', 'yyyy/mm/dd'),'F','America');
```

```
INSERT INTO persoana(id_persoana,prenume,nume,data_nastere,sex,tara_provenienta)
VALUES (164,'Radu','Alexandra',TO_DATE('1971/09/28', 'yyyy/mm/dd'),'F','Anglia');
```

```
INSERT INTO director(id_director,id_persoana)
VALUES (200,110);
```

```
INSERT INTO director(id_director,id_persoana)
VALUES (201,111);
```

```
INSERT INTO director(id_director,id_persoana)
VALUES (202,112);
```

```
INSERT INTO director(id_director,id_persoana)
VALUES (203,113);
```

```
INSERT INTO director(id_director,id_persoana)
VALUES (204,114);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (1,200,'Once Upon A Time',TO_DATE('2000/01/31', 'yyyy/mm/dd'),126,10,25);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (2,200,'Fight Club',TO_DATE('2017/04/19', 'yyyy/mm/dd'),142,1,15);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (3,202,'Mr&Mrs Smith',TO_DATE('2009/06/27', 'yyyy/mm/dd'),100,15,45);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (4,203,'Seven',TO_DATE('2011/09/06', 'yyyy/mm/dd'),129,21,17);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (5,204,'Bruce Almighty',TO_DATE('2020/01/30', 'yyyy/mm/dd'),171,17,2);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (6,200,'The Dark Knight',TO_DATE('2016/09/07', 'yyyy/mm/dd'),156,42,175);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (7,202,'Men In Black',TO_DATE('2014/10/29', 'yyyy/mm/dd'),110,7,15);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (8,202,'Titanic',TO_DATE('1999/05/10', 'yyyy/mm/dd'),120,60,225);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (9,203,'The Wolf',TO_DATE('2006/02/14', 'yyyy/mm/dd'),173,14,62);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (10,204,'Romeo + Juliet',TO_DATE('2010/09/19', 'yyyy/mm/dd'),186,10,15);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (11,200,'The Notebook',TO_DATE('2002/07/11', 'yyyy/mm/dd'),116,33,100);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (12,203,'La La Land',TO_DATE('2009/10/03', 'yyyy/mm/dd'),189,21,89);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (13,202,'Deadpool',TO_DATE('2003/11/30', 'yyyy/mm/dd'),193,46,75);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
```

```
VALUES (14,203,'Avengers',TO_DATE('2008/08/23', 'yyyy/mm/dd'),109,77,429);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (15,204,'The Hunger Games',TO_DATE('2000/05/10', 'yyyy/mm/dd'),88,53,220);
```

```
INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
VALUES (16,201,'Money Money',TO_DATE('2002/05/10', 'yyyy/mm/dd'),80,23,41);
```

```
INSERT INTO genfilm
VALUES (1,50);
```

```
INSERT INTO genfilm
VALUES (1,51);
```

```
INSERT INTO genfilm
VALUES (2,51);
```

```
INSERT INTO genfilm
VALUES (3,51);
```

```
INSERT INTO genfilm
VALUES (3,55);
```

```
INSERT INTO genfilm
VALUES (4,57);
```



```
INSERT INTO genfilm  
VALUES (5,52);
```

```
INSERT INTO genfilm  
VALUES (5,56);
```

```
INSERT INTO genfilm  
VALUES (6,52);
```

```
INSERT INTO genfilm  
VALUES (6,56);
```

```
INSERT INTO genfilm  
VALUES (7,52);
```

```
INSERT INTO genfilm  
VALUES (7,50);
```

```
INSERT INTO genfilm  
VALUES (8,55);
```

```
INSERT INTO genfilm  
VALUES (8,50);
```

```
INSERT INTO genfilm
```

```
VALUES (9,52);
```

```
INSERT INTO genfilm
```

```
VALUES (10,50);
```

```
INSERT INTO genfilm
```

```
VALUES (10,55);
```

```
INSERT INTO genfilm
```

```
VALUES (11,55);
```

```
INSERT INTO genfilm
```

```
VALUES (12,55);
```

```
INSERT INTO genfilm
```

```
VALUES (12,56);
```

```
INSERT INTO genfilm
```

```
VALUES (13,58);
```

```
INSERT INTO genfilm
```

```
VALUES (13,52);
```

```
INSERT INTO genfilm
```

```
VALUES (14,58);
```

```
INSERT INTO genfilm
```

```
VALUES (14,51);
```

```
INSERT INTO genfilm
```

```
VALUES (15,56);
```

```
INSERT INTO genfilm
```

```
VALUES (15,58);
```

```
INSERT INTO genfilm
```

```
VALUES (15,51);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (300,100);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (301,101);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (302,102);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (303,103);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (304,104);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (305,105);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (306,106);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (307,107);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (308,108);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (309,109);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (310,115);
```

```
INSERT INTO actor(id_actor,id_persoana)
VALUES (311,116);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (312,117);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (313,118);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (314,119);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (315,120);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (316,121);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (317,122);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (318,123);
```

```
INSERT INTO actor(id_actor,id_persoana)
```

```
VALUES (319,124);
```

```
INSERT INTO personaj
```

```
VALUES (1,300,'Ryan');
```

```
INSERT INTO personaj  
VALUES (1,302,'Bob');
```

```
INSERT INTO personaj  
VALUES (2,300,'Jack');
```

```
INSERT INTO personaj  
VALUES (2,301,'Martin');
```

```
INSERT INTO personaj  
VALUES (2,305,'Clara');
```

```
INSERT INTO personaj  
VALUES (3,300,'John');
```

```
INSERT INTO personaj  
VALUES (3,318,'Jade');
```

```
INSERT INTO personaj  
VALUES (4,301,'Alex');
```

```
INSERT INTO personaj  
VALUES (4,314,'Karina');
```

```
INSERT INTO personaj  
VALUES (4,309,'Robert');
```

```
INSERT INTO personaj  
VALUES (5,306,'Bruce');
```

```
INSERT INTO personaj  
VALUES (5,302,'Stephan');
```

```
INSERT INTO personaj  
VALUES (6,314,'Maya');
```

```
INSERT INTO personaj  
VALUES (7,302,'Ryan');
```

```
INSERT INTO personaj  
VALUES (7,319,'Tina');
```

```
INSERT INTO personaj  
VALUES (8,319,'Rose');
```

```
INSERT INTO personaj  
VALUES (8,303,'Jack');
```

```
INSERT INTO personaj
```

```
VALUES (9,303,'Jordan');
```

```
INSERT INTO personaj
```

```
VALUES (9,317,'Nicole');
```

```
INSERT INTO personaj
```

```
VALUES (10,302,'Romeo');
```

```
INSERT INTO personaj
```

```
VALUES (10,313,'Juliet');
```

```
INSERT INTO personaj
```

```
VALUES (11,304,'Noah');
```

```
INSERT INTO personaj
```

```
VALUES (11,312,'Babuna');
```

```
INSERT INTO personaj
```

```
VALUES (12,317,'Mia');
```

```
INSERT INTO personaj
```

```
VALUES (12,304,'Sebastian');
```

```
INSERT INTO personaj
```

```
VALUES (13,305,'Deadpool');
```



```
INSERT INTO personaj  
VALUES (13,311,'Rachel');
```

```
INSERT INTO personaj  
VALUES (14,308,'Iron Man');
```

```
INSERT INTO personaj  
VALUES (14,310,'Black Widow');
```

```
INSERT INTO personaj  
VALUES (15,311,'Katniss');
```

```
INSERT INTO rating  
VALUES (1,150,8.5);
```

```
INSERT INTO rating  
VALUES (2,151,9);
```

```
INSERT INTO rating  
VALUES (3,152,8);
```

```
INSERT INTO rating  
VALUES (4,153,3.5);
```

```
INSERT INTO rating  
VALUES (4,154,2);
```

```
INSERT INTO rating  
VALUES (5,154,6);
```

```
INSERT INTO rating  
VALUES (5,155,3);
```

```
INSERT INTO rating  
VALUES (6,155,9.5);
```

```
INSERT INTO rating  
VALUES (7,156,7.5);
```

```
INSERT INTO rating  
VALUES (8,157,8.5);
```

```
INSERT INTO rating  
VALUES (9,158,8.5);
```

```
INSERT INTO rating  
VALUES (10,159,8);
```

```
INSERT INTO rating
```

```
VALUES (10,160,9);
```

```
INSERT INTO rating
```

```
VALUES (11,160,10);
```

```
INSERT INTO rating
```

```
VALUES (11,155,9);
```

```
INSERT INTO rating
```

```
VALUES (12,161,10);
```

```
INSERT INTO rating
```

```
VALUES (13,162,9.5);
```

```
INSERT INTO rating
```

```
VALUES (14,163,9);
```

```
INSERT INTO rating
```

```
VALUES (15,164,7.5);
```

```
INSERT INTO rating
```

```
VALUES (16,164,5.5);
```

```
INSERT INTO gala
```

```
VALUES (600,'Academy Awards');
```

```
INSERT INTO gala
```

```
VALUES (601,'Golden Globe Awards');
```

```
INSERT INTO gala
```

```
VALUES (602,'British Academy Film Awards');
```

```
INSERT INTO gala
```

```
VALUES (603,'Emmy Award');
```

```
INSERT INTO premiu
```

```
VALUES (500,'Best Picture');
```

```
INSERT INTO premiu
```

```
VALUES (501,'Best Film Editing');
```

```
INSERT INTO premiu
```

```
VALUES (502,'Best Original Song');
```

```
INSERT INTO premiu
```

```
VALUES (503,'Best Visual Effects');
```

```
INSERT INTO premiu
```

```
VALUES (504,'Best Director');
```

```
INSERT INTO nominalizare
```

```
VALUES (500,1,600,2010,'premiat');
```

```
INSERT INTO nominalizare
```

```
VALUES (500,2,600,2010,'nominalizat');
```

```
INSERT INTO nominalizare
```

```
VALUES (500,3,600,2010,'nominalizat');
```

```
INSERT INTO nominalizare
```

```
VALUES (501,4,600,2011,'premiat');
```

```
INSERT INTO nominalizare
```

```
VALUES (501,1,601,2011,'nominalizat');
```

```
INSERT INTO nominalizare
```

```
VALUES (503,9,602,2010,'premiat');
```

```
INSERT INTO nominalizare
```

```
VALUES (504,6,600,2015,'nominalizat');
```

```
INSERT INTO nominalizare
```

```
VALUES (504,7,603,2015,'premiat');
```

```
INSERT INTO nominalizare
```

VALUES (504,8,602,2015,'nominalizat');

INSERT INTO nominalizare

VALUES (503,9,600,2014,'premiat');

≡  Live SQL

SQL Worksheet

```
200
201 INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
202 VALUES (13,202,'Deadpool',TO_DATE('2003/11/30','yyyy/mm/dd'),193,46,75);
203
204 INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
205 VALUES (14,203,'Avengers',TO_DATE('2008/08/23','yyyy/mm/dd'),109,77,429);
206
207 INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
208 VALUES (15,204,'The Hunger Games',TO_DATE('2000/05/10','yyyy/mm/dd'),88,53,220);
209
210 INSERT INTO film(id_film,id_director,titlu_film,data_lansare,durata,buget,box_office)
211 VALUES (16,201,'Money Money',TO_DATE('2002/05/10','yyyy/mm/dd'),80,23,41);
212
213 INSERT INTO genfilm
214 VALUES (1,50);
215
216 INSERT INTO genfilm
217 VALUES (1,51);
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

“Calculati intr-o colectie media rating-urilor si numarul total de rating-uri pentru fiecare film in parte si folositi-va de aceasta colectie pentru a actualiza valoarea

campului “rating” si “nr_rating” corespunzator fiecarui film astfel: Daca este mai mare decat 5, actualizam “rating” cu valoarea respectiva. Daca este mai mic decat 5 dar are cel puțin un review pozitiv, actualizam la 5. Altfel, daca toate sunt negative, actualizam la 2 . Afisati de asemenea si numele filmului impreuna cu rating-ul initial.”

```
CREATE OR REPLACE PROCEDURE exercitiul_6
```

```
AS
```

```
TYPE rec_rating IS RECORD ( id film.id_film%TYPE,
```

```
    rat rating.punctaj%TYPE,
```

```
    nr film.nr_rating%TYPE);
```

```
TYPE tab_rating IS TABLE OF rec_rating;
```

```
t tab_rating;
```

```
t_verif tab_rating := tab_rating();
```

```
medie NUMBER(4,2);
```

```
BEGIN
```

```
SELECT id_film,ROUND(AVG(punctaj), 2),COUNT(*)
```

```
BULK COLLECT INTO t
```

```
FROM rating
```

```
GROUP BY id_film
```

```
ORDER BY id_film;
```

```
DBMS_OUTPUT.PUT_LINE('Tabloul are ' || t.COUNT || ' elemente:');
```

```
FOR i IN t.FIRST..t.LAST LOOP
```

```
    IF t(i).rat>=5
```

```

        THEN DBMS_OUTPUT.PUT_LINE(t(i).id || ' FILM DE SUCCES CU ' || t(i).rat);

        medie := t(i).rat;

ELSE

    SELECT id_film,punctaj,1
    BULK COLLECT INTO t_verif
    FROM rating
    WHERE (id_film = t(i).id) AND (punctaj>=5);

    IF t_verif.COUNT > 0

        THEN DBMS_OUTPUT.PUT_LINE(t(i).id || ' FILM CU UN REVIEW POZITIV ' ||
t(i).rat);

        medie := 5;

    ELSE

        DBMS_OUTPUT.PUT_LINE(t(i).id || ' FILM PROST CU ' || t(i).rat);

        medie := 2;

    END IF;

END IF;

UPDATE film

SET rating = medie

WHERE id_film=t(i).id;

UPDATE film

SET nr_rating = t(i).nr

WHERE id_film=t(i).id;

END LOOP;

```


END;

/

BEGIN

exercitiul_6;

END;

/



Live SQL

SQL Worksheet

```
38      END IF;
39  END IF;
40
41  UPDATE film
42  SET rating = medie
43  WHERE id_film=t(i).id;
44
45  UPDATE film
46  SET nr_rating = t(i).nr
47  WHERE id_film=t(i).id;
48 END LOOP;|
49 END;
50 /
51 BEGIN
52     exercitiul_6;
53 END;
54 /
55
```

Procedure created.

Statement processed.

Tabloul are 16 elemente:

```
1 FILM DE SUCCES CU 8.5
2 FILM DE SUCCES CU 9
3 FILM DE SUCCES CU 8
4 FILM PROST CU 2.75
5 FILM CU UN REVIEW POZITIV 4.5
6 FILM DE SUCCES CU 9.5
7 FILM DE SUCCES CU 7.5
8 FILM DE SUCCES CU 8.5
9 FILM DE SUCCES CU 8.5
10 FILM DE SUCCES CU 8.5
11 FILM DE SUCCES CU 9.5
12 FILM DE SUCCES CU 10
```

7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

“Folosiți-va de un cursor pentru a afișa următoarele date pentru fiecare actor:

Dacă nu a jucat în niciun film, să se precizeze acest lucru.

Dacă a jucat într-un film în care s-au investit mai mulți bani decât cei câștigați, să se menționeze în ce film și cât a fost paguba.

Altfel, să se afișeze în câte filme a jucat și care este numele celui mai profitabil dintre ele.”

```
CREATE OR REPLACE PROCEDURE exercitiul_7
```

```
AS
```

```
    CURSOR c IS
```

```
    SELECT p.id_persoana,p.preume,p.nume,f.titlu_film,(f.box_office-f.buget) AS profit
```

```
    FROM persoana p RIGHT JOIN actor a ON (p.id_persoana=a.id_persoana)
```

```
    LEFT JOIN personaj per ON (a.id_actor=per.id_actor)
```

```
    LEFT JOIN film f ON (f.id_film=per.id_film)
```

```
    ORDER BY p.id_persoana;
```

```
    idp persoana.id_persoana%TYPE;
```

```
    preume persoana.nume%TYPE;
```

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

```
    titlu film.titlu_film%TYPE;
```

```
    profit film.buget%TYPE;
```

```
    preume_max persoana.nume%TYPE;
```

```

nume_max persoana.nume%TYPE;

titlu_max film.titlu_film%TYPE;

profit_max film.buget%TYPE;

numar NUMBER(3) :=0;

BEGIN

    OPEN c;

    FETCH c INTO idp,prenume_max,nume_max,titlu_max,profit_max;

    CLOSE c;

    FOR i IN c LOOP

        profit := i.profit; titlu := i.titlu_film; nume := i.nume; prenume := i.prenume; numar := numar
+ 1;

        IF i.id_persoana = idp

            THEN

                IF i.profit > profit_max

                    THEN profit_max := i.profit; titlu_max := i.titlu_film; nume_max := i.nume; prenume_max
:= i.prenume;

                END IF;

            ELSE

                IF profit_max IS NOT NULL AND profit_max >= 0

                    THEN dbms_output.put_line('Dintr-un total de ' || numar || ' filme, cel mai profitabil film
in care a jucat ' || prenume_max || ' ' || nume_max || ' a fost ' || titlu_max || ' cu un profit de '
|| profit_max || ' milioane.');
```

```

        numar := 0;

    END IF;

    IF profit IS NULL

    THEN dbms_output.put_line(prenume || ' ' || nume || ' nu a jucat in niciun film.');
```

o pierdere de ' || profit || ' milioane.');

```

    ELSIF profit < 0

    THEN dbms_output.put_line(prenume || ' ' || nume || ' a jucat in ' || titlu || ', film ce a avut
o pierdere de ' || profit || ' milioane.');
```

o pierdere de ' || profit || ' milioane.');

```

    END IF;

    END LOOP;

    numar := numar + 1;

    IF profit IS NULL

    THEN dbms_output.put_line(prenume || ' ' || nume || ' nu a jucat in niciun film.');
```

o pierdere de ' || profit || ' milioane.');

```

    ELSIF profit < 0

    THEN dbms_output.put_line(prenume || ' ' || nume || ' a jucat in ' || titlu || ', film ce a avut o
o pierdere de ' || profit || ' milioane.');
```

o pierdere de ' || profit || ' milioane.');

```

    ELSE dbms_output.put_line('Dintr-un total de ' || numar || ' filme, cel mai profitabil film in
care a jucat ' || prenume_max || ' ' || nume_max || ' a fost ' || titlu_max || ' cu un profit de ' ||
profit_max || ' milioane.');
```

o pierdere de ' || profit || ' milioane.');

```

    END IF;

END;

/

BEGIN

exercitiul_7;

END;

/
```

Live SQL

SQL Worksheet

```
42      ELSIF profit < 0
43      THEN dbms_output.put_line(prenume|| ' ' || nume|| ' a jucat in ' || titlu|| ', film ce a avut o pierdere de ' || profit|| ' milioane.');
```

```
44      END IF;
45  END LOOP;
46  numar := numar + 1;
47  IF profit IS NULL
48  THEN dbms_output.put_line(prenume|| ' ' || nume|| ' nu a jucat in niciun film.');
```

```
49  ELSIF profit < 0
50  THEN dbms_output.put_line(prenume|| ' ' || nume|| ' a jucat in ' || titlu|| ', film ce a avut o pierdere de ' || profit|| ' milioane.');
```

```
51  ELSE dbms_output.put_line('Dintr-un total de ' || numar || ' filme, cel mai profitabil film in care a jucat ' || prenume_max|| ' ' || nume_ma
52  END IF;
53  END;
54  /
55  BEGIN
56  exercitiul_7;
57  END;
```

Procedure created.

Statement processed.

Dintr-un total de 4 filme, cel mai profitabil film in care a jucat Brad Pitt a fost Mr&Mrs Smith cu un profit de 30 milioane.
Morgan Freeman a jucat in Seven, film ce a avut o pierdere de -4 milioane.
Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Morgan Freeman a fost Fight Club cu un profit de 14 milioane.
Will Smith a jucat in Bruce Almighty, film ce a avut o pierdere de -15 milioane.
Dintr-un total de 4 filme, cel mai profitabil film in care a jucat Will Smith a fost Once Upon A Time cu un profit de 15 milioane.
Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Leonardo DiCaprio a fost Titanic cu un profit de 165 milioane.
Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Ryan Gosling a fost La La Land cu un profit de 68 milioane.
Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Ryan Reynolds a fost Deadpool cu un profit de 29 milioane.
Jim Carrey a jucat in Bruce Almighty, film ce a avut o pierdere de -15 milioane.
Dorian Popa nu a jucat in niciun film.
Dintr-un total de 1 filme, cel mai profitabil film in care a jucat Robert Downey a fost Avengers cu un profit de 352 milioane.
Tom Cruise a jucat in Seven, film ce a avut o pierdere de -4 milioane.

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

“Creați o funcție care să returneze numele celui mai vârstnic actor care a jucat într-un film regizat de directorul al cărui nume este trimis ca parametru.”

CREATE OR REPLACE FUNCTION exercitiul_8

(nume_dir IN persoana.nume%TYPE DEFAULT 'Tarantino')

RETURN persoana.nume%TYPE

IS

```
dir_zero EXCEPTION;
```

```
dir_many EXCEPTION;
```

```
TYPE var_dir IS VARRAY(10) OF director.id_director%TYPE;
```

```
id_dir var_dir := var_dir();
```

```
var_num persoana.num%TYPE;
```

```
BEGIN
```

```
SELECT id_director
```

```
BULK COLLECT INTO id_dir
```

```
FROM director d JOIN persoana p ON (p.id_persoana=d.id_persoana)
```

```
WHERE p.num = num_dir;
```

```
IF id_dir.COUNT>1 THEN
```

```
RAISE dir_many;
```

```
END IF;
```

```
IF SQL%NOTFOUND THEN
```

```
RAISE dir_zero;
```

```
END IF;
```

```
SELECT num
```

```
INTO var_num
```

```
FROM persoana p JOIN actor a ON (p.id_persoana=a.id_persoana)
```

```
JOIN personaj per ON (per.id_actor=a.id_actor)
```

```

JOIN film f ON (f.id_film=per.id_film)

WHERE f.id_director = id_dir(1) AND p.data_nastere = ( SELECT MIN(data_nastere)

                                FROM persoana p JOIN actor a ON

(p.id_persoana=a.id_persoana)

                                JOIN personaj per ON (per.id_actor=a.id_actor)

                                JOIN film f ON (f.id_film=per.id_film)

                                WHERE f.id_director = id_dir(1));

RETURN var_nume;

EXCEPTION

    WHEN dir_zero THEN RAISE_APPLICATION_ERROR (-20000,'Nu exista un director cu
acest nume');

    WHEN dir_many THEN RAISE_APPLICATION_ERROR (-20000,'Sunt mai multi directori cu
acest nume');

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista
actori care sa fi lucrat intr-un film regizat de acest director.');
```

WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001,'Sunt cel putin
2 actori care sunt cei mai varstnici.');

```

END;

/

DECLARE

    abc persoana.nume%TYPE;

BEGIN

    abc := exercitiul_8;

    --abc := exercitiul_8('Bora');

    --abc := exercitiul_8('Karev');
```

```
--abc := exercitiul_8('Scorsese');

--abc := exercitiul_8('Hudson');

DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);

END;
```

Live SQL

SQL Worksheet

```

40 WHEN dir_zero THEN RAISE_APPLICATION_ERROR (-20000,'Nu exista un director cu acest nume');
41 WHEN dir_many THEN RAISE_APPLICATION_ERROR (-20000,'Sunt mai multi directori cu acest nume');
42 WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista actori care sa fi lucrat intr-un film regizat de acest director.');
```

Function created.

Statement processed.

Actorul cerut are numele Streep

Live SQL

SQL Worksheet

```

47 DECLARE
48     abc persoana.nume%TYPE;
49 BEGIN
50     --abc := exercitiul_8;
51     --abc := exercitiul_8('Bora');
52     --abc := exercitiul_8('Karev');
53     abc := exercitiul_8('Scorsese');
```

Function created.

ORA-20001: Nu exista actori care sa fi lucrat intr-un film regizat de acest director.
ORA-06512: at line 7
ORA-06512: at "SYS.DBMS_SQL", line 1721


```

47 DECLARE
48     abc persoana.nume%TYPE;
49 BEGIN
50     --abc := exercitiul_8;
51     abc := exercitiul_8('Bora');
52     --abc := exercitiul_8('Karev');
53     --abc := exercitiul_8('Scorsese');
54     --abc := exercitiul_8('Hudson');
55     DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
56 END;

```

Function created.

ORA-20000: Nu exista un director cu acest nume ORA-06512: at "SQL_KPMGZWESL:
ORA-06512: at line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

47 DECLARE
48     abc persoana.nume%TYPE;
49 BEGIN
50     --abc := exercitiul_8;
51     --abc := exercitiul_8('Bora');
52     abc := exercitiul_8('Karev');
53     --abc := exercitiul_8('Scorsese');
54     --abc := exercitiul_8('Hudson');
55     DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
56 END;

```

Function created.

ORA-20000: Sunt mai multi directori cu acest nume ORA-06512: at "SQL_KPMGZW
ORA-06512: at line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

47 DECLARE
48     abc persoana.nume%TYPE;
49 BEGIN
50     --abc := exercitiul_8;
51     --abc := exercitiul_8('Bora');
52     --abc := exercitiul_8('Karev');
53     --abc := exercitiul_8('Scorsese');
54     abc := exercitiul_8('Hudson');
55     DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
56 END;

```

Function created.

ORA-20001: Sunt cel putin 2 actori care sunt cei mai varstnici. ORA-06512: at '
ORA-06512: at line 8
ORA-06512: at "SYS.DBMS_SQL", line 1721

9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

“ Definiti un subprogram stocat care primeste ca parametrii un gen de film si o tara introdusa de la tastatura. In cazul in care tara este mentionata, se va returna numarul de actori din acea tara care au jucat intr-un film de acel gen si se va afisa acest numar. In caz contrar, pentru fiecare gen se va afisa numarul de actori pe fiecare tara in parte si se va returna numarul total de actori care au jucat pe acel gen.”

```
CREATE OR REPLACE PROCEDURE exercitiul_9
(nr OUT NUMBER,
nume_gen IN gen.titlu_gen%TYPE,
nume_tara IN persoana.tara_provenienta%TYPE := NULL)
IS

gen_zero EXCEPTION;
actori_tara_zero EXCEPTION;
actori_zero EXCEPTION;

TYPE var IS VARRAY(5) OF gen.id_gen%TYPE;
var_gen var := var();

BEGIN

SELECT id_gen
BULK COLLECT INTO var_gen
FROM gen
WHERE titlu_gen = nume_gen;
```

```
IF SQL%NOTFOUND THEN
```

```
RAISE gen_zero;
```

```
END IF;
```

```
IF (nume_tara IS NOT NULL)
```

```
THEN
```

```
    SELECT COUNT(DISTINCT p.id_persoana)
```

```
    INTO nr
```

```
    FROM persoana p JOIN actor a ON (p.id_persoana=a.id_persoana)
```

```
    JOIN personaj per ON (per.id_actor=a.id_actor)
```

```
    JOIN film f ON (f.id_film=per.id_film)
```

```
    JOIN genfilm gf ON (f.id_film=gf.id_film)
```

```
    JOIN gen g ON (g.id_gen=gf.id_gen)
```

```
    WHERE titlu_gen = nume_gen AND tara_provenienta = nume_tara;
```

```
IF (nr=0) THEN
```

```
RAISE actori_tara_zero;
```

```
END IF;
```

```
DBMS_OUTPUT.PUT_LINE('Pe acest gen au jucat ' || nr || ' actori din ' || nume_tara);
```

```
ELSE
```

```
nr := 0;
```

```
FOR i IN (SELECT tara_provenienta,COUNT(DISTINCT p.id_persoana) AS numar
```

```
    FROM persoana p JOIN actor a ON (p.id_persoana=a.id_persoana)
```

```

        JOIN personaj per ON (per.id_actor=a.id_actor)

        JOIN film f ON (f.id_film=per.id_film)

        JOIN genfilm gf ON (f.id_film=gf.id_film)

        JOIN gen g ON (g.id_gen=gf.id_gen)

        WHERE titlu_gen = nume_gen

        GROUP BY tara_provenienta) LOOP

        DBMS_OUTPUT.PUT_LINE('Pe acest gen au jucat ' || i.numar || ' actori din ' ||
i.tara_provenienta );

        nr := nr + i.numar;

    END LOOP;

    IF (nr=0) THEN

        RAISE actori_zero;

    END IF;

END IF;

EXCEPTION

    WHEN gen_zero THEN RAISE_APPLICATION_ERROR (-20002,'Nu exista un gen cu acest
nume');

    WHEN actori_tara_zero THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista actori din
tara mentionata care sa fi lucrat pe acest gen de film.');
```

```

    WHEN actori_zero THEN RAISE_APPLICATION_ERROR(-20004,'Nu exista actori care sa fi
lucrat pe acest gen de film.');
```

```

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista
actori ceruti care sa fi lucrat pe acest gen de film.');
```

```

END;

/
```

```
DECLARE
```

```
    abc NUMBER;
```

```
BEGIN
```

```
    exercitiul_9(abc,'Actiune');
```

```
    --exercitiul_9(abc,'Actiune','America');
```

```
    --exercitiul_9(abc,'Actiune','Italia');
```

```
    --exercitiul_9(abc,'SF');
```

```
    --exercitiul_9(abc,'Istoric');
```

```
    DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
```

```
END;
```



Live SQL

/

SQL Worksheet

```
60      WHEN gen_zero THEN RAISE_APPLICATION_ERROR(-20002,'Nu exista un gen');
61      WHEN actori_tara_zero THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista actori din tara');
62      WHEN actori_zero THEN RAISE_APPLICATION_ERROR(-20004,'Nu exista actori');
63      WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista actori');
64  END;
65  /
66
67  DECLARE
68      abc NUMBER;
69  BEGIN
70      exercitiul_9(abc,'Actiune');
71      --exercitiul_9(abc,'Actiune','America');
72      --exercitiul_9(abc,'Actiune','Italia');
73      --exercitiul_9(abc,'SF');
74      --exercitiul_9(abc,'Istoric');
75      DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
76  END;
77  /
```

Procedure created.

Statement processed.

Pe acest gen au jucat 1 actori din Rusia
Pe acest gen au jucat 6 actori din America
Pe acest gen au jucat 1 actori din Canada
Numarul total de actori este 8

```

67 DECLARE
68     abc NUMBER;
69 BEGIN
70     --exercitiul_9(abc,'Actiune');
71     --exercitiul_9(abc,'Actiune','America');
72     exercitiul_9(abc,'Actiune','Italia');
73     --exercitiul_9(abc,'SF');
74     --exercitiul_9(abc,'Istoric');
75     DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
76 END;
77 /

```

Procedure created.

ORA-20003: Nu exista actori din tara mentionata care sa fi lucrat pe acest gen de film.

ORA-06512: at line 6

ORA-06512: at "SYS.DBMS_SQL", line 1721

```

67 DECLARE
68     abc NUMBER;
69 BEGIN
70     --exercitiul_9(abc,'Actiune');
71     exercitiul_9(abc,'Actiune','America');
72     --exercitiul_9(abc,'Actiune','Italia');
73     --exercitiul_9(abc,'SF');
74     --exercitiul_9(abc,'Istoric');
75     DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
76 END;
77 /

```

Procedure created.

Statement processed.

Pe acest gen au jucat 6 actori din America

Numarul total de actori este 6

```

67 DECLARE
68     abc NUMBER;
69 BEGIN
70     --exercitiul_9(abc,'Actiune');
71     --exercitiul_9(abc,'Actiune','America');
72     --exercitiul_9(abc,'Actiune','Italia');
73     exercitiul_9(abc,'SF');
74     --exercitiul_9(abc,'Istoric');
75     DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
76 END;
77 /

```

Procedure created.

ORA-20004: Nu exista actori care sa fi lucrat pe acest gen de film. ORA-06512: at
ORA-06512: at line 7
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

67 DECLARE
68     abc NUMBER;
69 BEGIN
70     --exercitiul_9(abc,'Actiune');
71     --exercitiul_9(abc,'Actiune','America');
72     --exercitiul_9(abc,'Actiune','Italia');
73     --exercitiul_9(abc,'SF');
74     exercitiul_9(abc,'Istoric');
75     DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || abc);
76 END;
77 /

```

Procedure created.

ORA-20002: Nu exista un gen cu acest nume ORA-06512: at "SQL_KPMGZWESLJMKBNDRG
ORA-06512: at line 8
ORA-06512: at "SYS.DBMS_SQL", line 1721

10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

“Definiti un trigger astfel incat la inserarea unei valori in tabelul “GENFILM”, sa se verifice castigul acestuia. Daca este peste 100, atunci se pot insera pana la 5 genuri de film, altfel eroare. Iar daca este sub 100, sunt premise pana la 3 genuri de film, altfel eroare.”

```

CREATE OR REPLACE TRIGGER exercitiul_10

AFTER INSERT on genfilm

DECLARE

numar NUMBER(2);

punctaj_nou film.rating%TYPE;

BEGIN

FOR i IN (SELECT MAX(box_office) AS castig, COUNT(*) AS numar_gen

        FROM genfilm g JOIN film f ON (g.id_film=f.id_film)

        GROUP BY g.id_film

        ORDER BY COUNT(*) DESC)

LOOP

    IF (i.castig > 100) THEN

        IF ((i.numar_gen) > 5) THEN

            RAISE_APPLICATION_ERROR(-20002, 'Un film popular nu poate avea mai mult de 5

genuri');

        END IF;

    ELSIF ((i.numar_gen) > 3) THEN

        RAISE_APPLICATION_ERROR(-20002, 'Un film obisnuit nu poate avea mai mult de 3

genuri');

    END IF;

END LOOP;

END;

/

```



```
INSERT INTO genfilm  
VALUES (3,52);
```

```
INSERT INTO genfilm  
VALUES (3,53);
```

```
INSERT INTO genfilm  
VALUES (15,52);
```

```
INSERT INTO genfilm  
VALUES (15,53);
```

```
INSERT INTO genfilm  
VALUES (15,54);
```

```
SELECT COUNT(*)  
FROM genfilm  
WHERE id_film=15;
```

```
SELECT COUNT(*)  
FROM genfilm  
WHERE id_film=3;
```



SQL Worksheet

```
13 LOOP
14     IF (i.castig > 100) THEN
15         IF ((i.numar_gen) > 5) THEN
16             RAISE_APPLICATION_ERROR(-20002, 'Un film popular nu poate avea mai mult de 5 genuri');
17         END IF;
18     ELSIF ((i.numar_gen) > 3) THEN
19         RAISE_APPLICATION_ERROR(-20002, 'Un film obisnuit nu poate avea mai mult de 3 genuri');
20     END IF;
21 END LOOP;
22 END;
23 /
24
25 INSERT INTO genfilm
26 VALUES (3,52);
27
28 INSERT INTO genfilm
29 VALUES (3,53);
30
```

Trigger created.

1 row(s) inserted.

```
28 INSERT INTO genfilm
29 VALUES (3,53);
30
31 INSERT INTO genfilm
32 VALUES (15,52);
33
```

ORA-20002: Un film obisnuit nu poate avea mai mult de 3 genuri ORA-06512:
ORA-06512: at "SQL_OGIRGYQLWJMHAUOCNWPVPGMMN.EXERCITIUL_10", line 17
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

31 INSERT INTO genfilm
32 VALUES (15,52);
33
34 INSERT INTO genfilm
35 VALUES (15,53);
36
37 INSERT INTO genfilm
38 VALUES (15,54);
39
40 SELECT COUNT(*)
41 FROM genfilm
42 WHERE id_film=15;
43
44 SELECT COUNT(*)
45 FROM genfilm

```

1 row(s) inserted.

1 row(s) inserted.

ORA-20002: Un film popular nu poate avea mai mult de 5 genuri ORA-06512:
ORA-06512: at "SQL_OGIRGYQLWJMHAUOCNWPVPGMMN.EXERCITIUL_10", line 14
ORA-06512: at "SYS.DBMS_SQL", line 1721

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

“Definiti un trigger astfel incat, pentru orice LMD asupra coloanei “punctaj” din tabelul “RATING”, coloanele “rating” si “nr_rating” din tabelul “FILM” sa fie actualizate corespunzator.”

```
CREATE OR REPLACE TRIGGER exercitiul_11
```

```
BEFORE INSERT OR DELETE OR UPDATE OF punctaj on rating
```

```
FOR EACH ROW
```

```
DECLARE
```

```
numar NUMBER(2);
```

```
punctaj_nou film.rating%TYPE;
```

BEGIN

SELECT nr_rating,rating

INTO numar,punctaj_nou

FROM film

WHERE id_film = :NEW.id_film OR id_film = :OLD.id_film;

IF INSERTING THEN

punctaj_nou := ((punctaj_nou)*numar+(:NEW.punctaj))/(numar+1);

UPDATE film

SET nr_rating = (numar+1)

WHERE id_film = :NEW.id_film;

ELSIF DELETING THEN

IF (numar > 1)

THEN punctaj_nou := ((punctaj_nou)*numar-(:OLD.punctaj))/(numar-1);

ELSE punctaj_nou := 0;

END IF;

UPDATE film

SET nr_rating = (numar-1)

WHERE id_film = :OLD.id_film;

ELSE

```
punctaj_nou := ((punctaj_nou)*numar-(:OLD.punctaj)+(:NEW.punctaj))/numar;  
END IF;
```

```
UPDATE film  
SET rating = punctaj_nou  
WHERE id_film = :NEW.id_film OR id_film = :OLD.id_film;
```

```
END;
```

```
/
```

```
EXECUTE exercitiul_6;
```

```
/
```

```
SELECT rating,nr_rating  
FROM film  
WHERE id_film = 11;
```

```
INSERT INTO rating  
VALUES (11,156,8);
```

```
DELETE FROM rating  
WHERE id_film = 11 AND id_persoana = 156;
```

```
UPDATE rating
```

SET punctaj = 10

WHERE id_film = 11 AND id_persoana = 155;

/



SQL Worksheet

```
35 END IF;  
36  
37 UPDATE film  
38 SET rating = punctaj_nou  
39 WHERE id_film = :NEW.id_film OR id_film = :OLD.id_film;  
40  
41 END;  
42 /  
43  
44 SELECT rating,nr_rating  
45 FROM film  
46 WHERE id_film = 11;  
47  
48 INSERT INTO rating  
49 VALUES (11,156,8);  
50  
51 DELETE FROM rating
```

Trigger created.

RATING	NR_RATING
9.5	2

[Download CSV](#)

```

48 INSERT INTO rating
49 VALUES (11,156,8);
50
51 SELECT rating,nr_rating
52 FROM film
53 WHERE id_film = 11;
54

```

1 row(s) inserted.

RATING	NR_RATING
9	3

```

55 DELETE FROM rating
56 WHERE id_film = 11 AND id_persoana = 156;
57
58 SELECT rating,nr_rating
59 FROM film
60 WHERE id_film = 11;
61
62 UPDATE rating
63 SET punctaj = 10

```

1 row(s) deleted.

RATING	NR_RATING
9.5	2

```

62 UPDATE rating
63 SET punctaj = 10
64 WHERE id_film = 11 AND id_persoana = 155;
65
66 SELECT rating,nr_rating
67 FROM film
68 WHERE id_film = 11;

```

1 row(s) updated.

RATING	NR_RATING
10	2

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

“Definiti un trigger astfel incat, modificarea unui tabel asociativ sa nu fie permisa, sa dea eroare, iar pentru alt table sa se afiseze user-ul care a facut modificarea.

```
CREATE OR REPLACE TRIGGER exercitiul_12
```

```
BEFORE ALTER ON SCHEMA
```

```
DECLARE
```

```
y user_tables.table_name%TYPE;
```

```
v VARCHAR2(30);
```

```
BEGIN
```

```
v := USER;
```

```
SELECT ora_dict_obj_name
```

```
INTO y
```

```
FROM DUAL;
```

```
IF y IN ('PERSONAJ','GENFILM','RATING','NOMINALIZARE')
```

```
THEN
```

```
    RAISE_APPLICATION_ERROR(-20001,'Nu ai voie sa modifici un tabel asociativ!');
```

```
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Tabelul a fost modificat de catre ' || v);
```

```
END IF;
```

```
END;
```

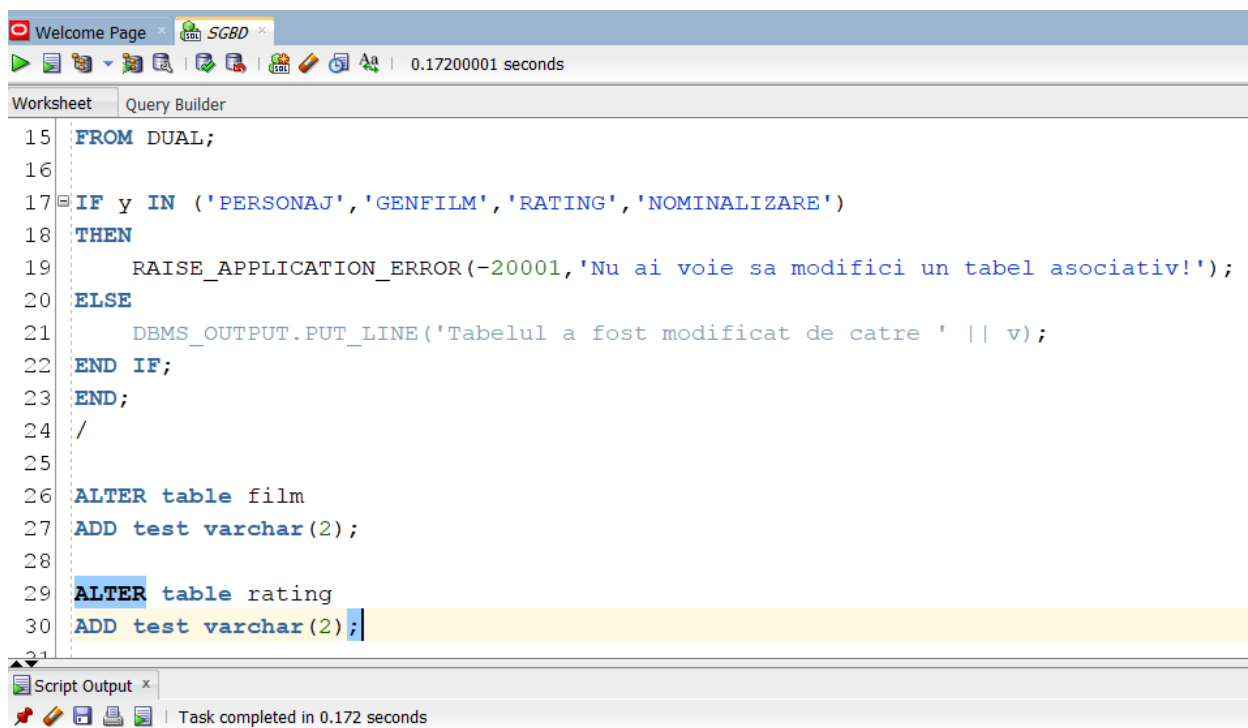
```
ALTER table film
```


ADD test varchar(2);

ALTER table rating

ADD test varchar(2);

Am adaugat poza din Oracle SQL developer deoarece in Sql Live nu imi permitea.



```
15 FROM DUAL;
16
17 IF y IN ('PERSONAJ', 'GENFILM', 'RATING', 'NOMINALIZARE')
18 THEN
19     RAISE_APPLICATION_ERROR(-20001, 'Nu ai voie sa modifici un tabel asociativ!');
20 ELSE
21     DBMS_OUTPUT.PUT_LINE('Tabelul a fost modificat de catre ' || v);
22 END IF;
23 END;
24 /
25
26 ALTER table film
27 ADD test varchar(2);
28
29 ALTER table rating
30 ADD test varchar(2);
```

Trigger EXERCITIUL_12 compiled

Table FILM altered.

Error starting at line : 29 in command -

ALTER table rating



ADD test varchar(2)

Error report -

ORA-00604: error occurred at recursive SQL level 1

ORA-20001: Nu ai voie sa modifici un tabel asociativ!

13. Am atasat codul in fisierul sql.

 **Live SQL**

SQL Worksheet

```
217
218 DECLARE
219     abc persoana.nume%TYPE;
220     a NUMBER;
221 BEGIN
222     exercitiul_13.exercitiul_6;
223     --exercitiul_13.exercitiul_7;
224     --abc := exercitiul_13.exercitiul_8;
225     --abc := exercitiul_13.exercitiul_8('Bora');
226     --abc := exercitiul_13.exercitiul_8('Karev');
227     --abc := exercitiul_13.exercitiul_8('Scorsese');
228     --abc := exercitiul_13.exercitiul_8('Hudson');
229     --DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
230     --exercitiul_13.exercitiul_9(a,'Actiune');
231     --exercitiul_13.exercitiul_9(a,'Actiune','America');
232     --exercitiul_13.exercitiul_9(a,'Actiune','Italia');
233     --exercitiul_13.exercitiul_9(a,'SF');
234     --exercitiul_13.exercitiul_9(a,'Istoric');
235     --DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || a);
236 END;
```

Statement processed.

Tabloul are 16 elemente:

- 1 FILM DE SUCCES CU 8.5
- 2 FILM DE SUCCES CU 9
- 3 FILM DE SUCCES CU 8
- 4 FILM PROST CU 2.75
- 5 FILM CU UN REVIEW POZITIV 4.5
- 6 FILM DE SUCCES CU 9.5
- 7 FILM DE SUCCES CU 7.5
- 8 FILM DE SUCCES CU 8.5
- 9 FILM DE SUCCES CU 8.5
- 10 FILM DE SUCCES CU 8.5

```

217
218 DECLARE
219     abc persoana.nume%TYPE;
220     a NUMBER;
221 BEGIN
222     --exercitiul_13.exercitiul_6;
223     --exercitiul_13.exercitiul_7;
224     abc := exercitiul_13.exercitiul_8;
225     --abc := exercitiul_13.exercitiul_8('Bora');
226     --abc := exercitiul_13.exercitiul_8('Karev');
227     --abc := exercitiul_13.exercitiul_8('Scorsese');
228     --abc := exercitiul_13.exercitiul_8('Hudson');
229     DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
230     --exercitiul_13.exercitiul_9(a,'Actiune');|
231     --exercitiul_13.exercitiul_9(a,'Actiune','America');
232     --exercitiul_13.exercitiul_9(a,'Actiune','Italia');
233     --exercitiul_13.exercitiul_9(a,'SF');
234     --exercitiul_13.exercitiul_9(a,'Istoric');
235     --DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || a);
236 END;

```

Statement processed.

Actorul cerut are numele Streep

```

217
218 DECLARE
219     abc persoana.nume%TYPE;
220     a NUMBER;
221 BEGIN
222     --exercitiul_13.exercitiul_6;
223     exercitiul_13.exercitiul_7;
224     --abc := exercitiul_13.exercitiul_8;
225     --abc := exercitiul_13.exercitiul_8('Bora');
226     --abc := exercitiul_13.exercitiul_8('Karev');
227     --abc := exercitiul_13.exercitiul_8('Scorsese');|
228     --abc := exercitiul_13.exercitiul_8('Hudson');
229     --DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
230     --exercitiul_13.exercitiul_9(a,'Actiune');
231     --exercitiul_13.exercitiul_9(a,'Actiune','America');
232     --exercitiul_13.exercitiul_9(a,'Actiune','Italia');
233     --exercitiul_13.exercitiul_9(a,'SF');
234     --exercitiul_13.exercitiul_9(a,'Istoric');
235     --DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || a);
236 END;

```

Statement processed.

Dintr-un total de 4 filme, cel mai profitabil film in care a jucat Brad Pitt a fost Mr&Mrs Smith cu un profit de 30 milioane.

Morgan Freeman a jucat in Seven, film ce a avut o pierdere de -4 milioane.

Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Morgan Freeman a fost Fight Club cu un profit de 14 milioane.

Will Smith a jucat in Bruce Almighty, film ce a avut o pierdere de -15 milioane.

Dintr-un total de 4 filme, cel mai profitabil film in care a jucat Will Smith a fost Once Upon A Time cu un profit de 15 milioane.

Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Leonardo DiCaprio a fost Titanic cu un profit de 165 milioane.

Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Ryan Gosling a fost La La Land cu un profit de 68 milioane.

Dintr-un total de 2 filme, cel mai profitabil film in care a jucat Ryan Reynolds a fost Deadpool cu un profit de 29 milioane.

Jim Carrey a jucat in Bruce Almighty, film ce a avut o pierdere de -15 milioane.

Dorian Popa nu a jucat in niciun film.

Dintr-un total de 1 filme, cel mai profitabil film in care a jucat Robert Downey a fost Avengers cu un profit de 352 milioane.

```

217
218 DECLARE
219     abc persoana.nume%TYPE;
220     a NUMBER;
221 BEGIN
222     --exercitiul_13.exercitiul_6;
223     --exercitiul_13.exercitiul_7;
224     --abc := exercitiul_13.exercitiul_8;
225     --abc := exercitiul_13.exercitiul_8('Bora');
226     --abc := exercitiul_13.exercitiul_8('Karev');|
227     --abc := exercitiul_13.exercitiul_8('Scorsese');
228     --abc := exercitiul_13.exercitiul_8('Hudson');
229     --DBMS_OUTPUT.PUT_LINE('Actorul cerut are numele ' || abc);
230     --exercitiul_13.exercitiul_9(a,'Actiune');
231     exercitiul_13.exercitiul_9(a,'Actiune','America');
232     --exercitiul_13.exercitiul_9(a,'Actiune','Italia');
233     --exercitiul_13.exercitiul_9(a,'SF');
234     --exercitiul_13.exercitiul_9(a,'Istoric');
235     DBMS_OUTPUT.PUT_LINE('Numarul total de actori este ' || a);
236 END;

```

Statement processed.

Pe acest gen au jucat 6 actori din America

Numarul total de actori este 6

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.

“Sa se defineasca un pachet cu urmatoarele obiecte: Un cursor ce primeste toate id-urile de film, o functie ce primeste ca parametru un id_film si intoarce vector cu cu id-urile actorilor ce joaca in acel film, o procedura ce se foloseste de cursor pentru a returna un vector de vectori de id_actor pentru toate filmele din baza de date si un vector cu un singur vector cu elementul 0 in caz ca nu exista actori pentru filmul respectiv si o procedura ce primeste ca parametru un vector de vectori de id_actor si afiseaza numele filmului si ce actori au jucat in el sau daca nu a jucat niciunul si intoarce numarul total de personaje din baza de date.”

```
CREATE TYPE vector_actor IS VARRAY(10) OF number(4);
```

```
CREATE TYPE mat_actor IS VARRAY(20) OF vector_actor;
```

```
CREATE OR REPLACE PACKAGE exercitiul_14
```

```
IS
```

```
CURSOR c
```

```
IS
```

```
SELECT id_film FROM film
```

```
ORDER BY id_film;
```

```
FUNCTION actori_film( id film.id_film%TYPE)
```

```
RETURN vector_actor;
```

```
PROCEDURE mat_actori(m_actori OUT mat_actor);
```

```
PROCEDURE afiseaza(af_actori IN mat_actor,
```

```
nr_personaje OUT NUMBER);
```

```
END;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY exercitiul_14
```

```
IS
```

```
FUNCTION actori_film( id film.id_film%TYPE)
```

```
RETURN vector_actor
```

```
IS
```

```
v_actori vector_actor := vector_actor();
```

```
BEGIN

SELECT id_actor

BULK COLLECT INTO v_actori

FROM personaj

WHERE id_film = id;

IF v_actori.COUNT = 0

THEN v_actori.EXTEND; v_actori(1) := 0;

END IF;

RETURN v_actori;

END;
```

```
PROCEDURE mat_actori(m_actori OUT mat_actor) IS

BEGIN

m_actori := mat_actor(vector_actor());

FOR i IN c LOOP

    m_actori.EXTEND;

    m_actori(i.id_film) := vector_actor();

    m_actori(i.id_film) := actori_film(i.id_film);

END LOOP;

END;
```

```
PROCEDURE afiseaza(af_actori IN mat_actor,

    nr_personaje OUT NUMBER) IS
```

```

nume_film film.titlu_film%TYPE;

actor_nume persoana.nume%TYPE;

actor_prenume persoana.prenume%TYPE;

BEGIN

    nr_personaje := 0;

    FOR i in 1..(af_actors.COUNT)-1 LOOP

        SELECT titlu_film

        INTO nume_film

        FROM film

        WHERE id_film=i;

        DBMS_OUTPUT.PUT('In filmul ' || nume_film);

        IF(af_actors(i)(1) = 0)

        THEN DBMS_OUTPUT.PUT_LINE(' nu e niciun actor inregistrat');

        ELSE DBMS_OUTPUT.PUT(' au jucat actorii: ');

        FOR j IN 1..af_actors(i).LAST LOOP

            nr_personaje := nr_personaje + 1;

            SELECT prenume,nume INTO actor_prenume,actor_nume

            FROM persoana

            WHERE id_persoana = (SELECT id_persoana

                                FROM actor

                                WHERE id_actor = af_actors(i)(j));

            IF j = af_actors(i).LAST

            THEN DBMS_OUTPUT.PUT_LINE(actor_prenume || ' ' || actor_nume || '.');

            ELSE DBMS_OUTPUT.PUT(actor_prenume || ' ' || actor_nume || ', ');

        END LOOP;

    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Nr. de personaje: ' || nr_personaje);

END;

```

```
END IF;  
END LOOP;  
END IF;  
END LOOP;  
END;  
END;  
/
```

```
DECLARE
```

```
nr NUMBER;
```

```
m_actori mat_actor := mat_actor(vector_actor());
```

```
BEGIN
```

```
exercitiul_14.mat_actori(m_actori);
```

```
exercitiul_14.afiseaza(m_actori,nr);
```

```
DBMS_OUTPUT.PUT_LINE('In total, avem in baza de date ' || nr || ' personaje.');
```

```
END;
```




SQL Worksheet

```
77      THEN DBMS_OUTPUT.PUT_LINE(actor_prenume || ' ' || actor_nume || '.');
78      ELSE DBMS_OUTPUT.PUT(actor_prenume || ' ' || actor_nume || ', ');
79      END IF;
80      END LOOP;
81      END IF;
82      END LOOP;
83  END;
84  END;
85  /
86
87  DECLARE
88
89  nr NUMBER;
90  m_actori mat_actor := mat_actor(vector_actor());
91
92  BEGIN
93  exercitiul_14.mat_actori(m_actori);
94  exercitiul_14.afiseaza(m_actori,nr);
95  DBMS_OUTPUT.PUT_LINE('In total, avem in baza de date ' || nr || ' personaje.');
```

```
96  END;
97
In filmul Bruce Almighty au jucat actorii: Will Smith, Jim Carrey.
In filmul The Dark Knight au jucat actorii: Anne Hathaway.
In filmul Men In Black au jucat actorii: Will Smith, Kate Winslet.
In filmul Titanic au jucat actorii: Leonardo DiCaprio, Kate Winslet.
In filmul The Wolf au jucat actorii: Leonardo DiCaprio, Emma Stone.
In filmul Romeo + Juliet au jucat actorii: Will Smith, Natalie Portman.
In filmul The Notebook au jucat actorii: Ryan Gosling, Meryl Streep.
In filmul La La Land au jucat actorii: Ryan Gosling, Emma Stone.
In filmul Deadpool au jucat actorii: Ryan Reynolds, Jeniffer Lawrence.
In filmul Avengers au jucat actorii: Robert Downey, Scarlet Johansson.
In filmul The Hunger Games au jucat actorii: Jeniffer Lawrence.
In filmul Money Money nu e niciun actor inregistrat
In total, avem in baza de date 30 personaje.
```