Proiect
Sisteme de Gestiune a Bazelor de Date

- Gestionarea unor Cinematografe -

Bălăiță Cosmin-Neculai

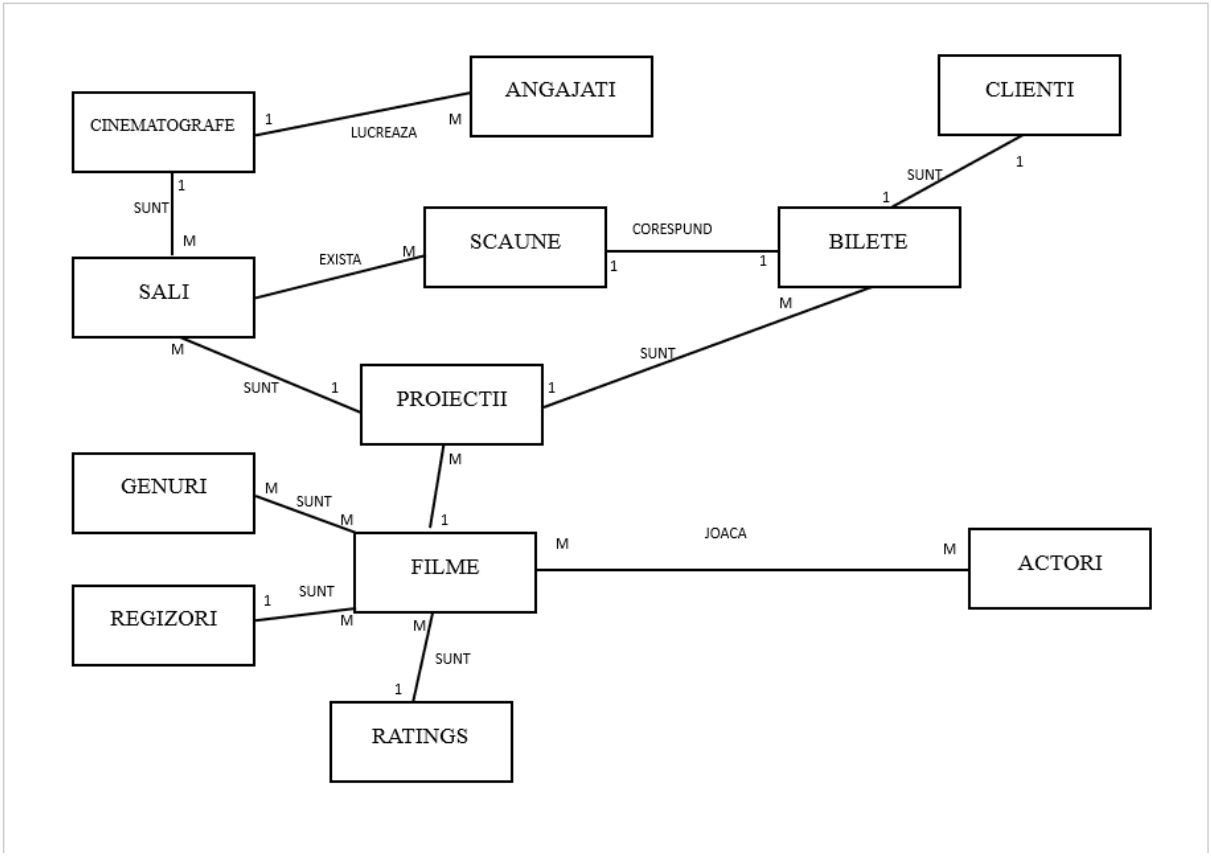
Grupa 241

Ex 1: Prezentarea pe scurt baza de date (utilitatea ei)

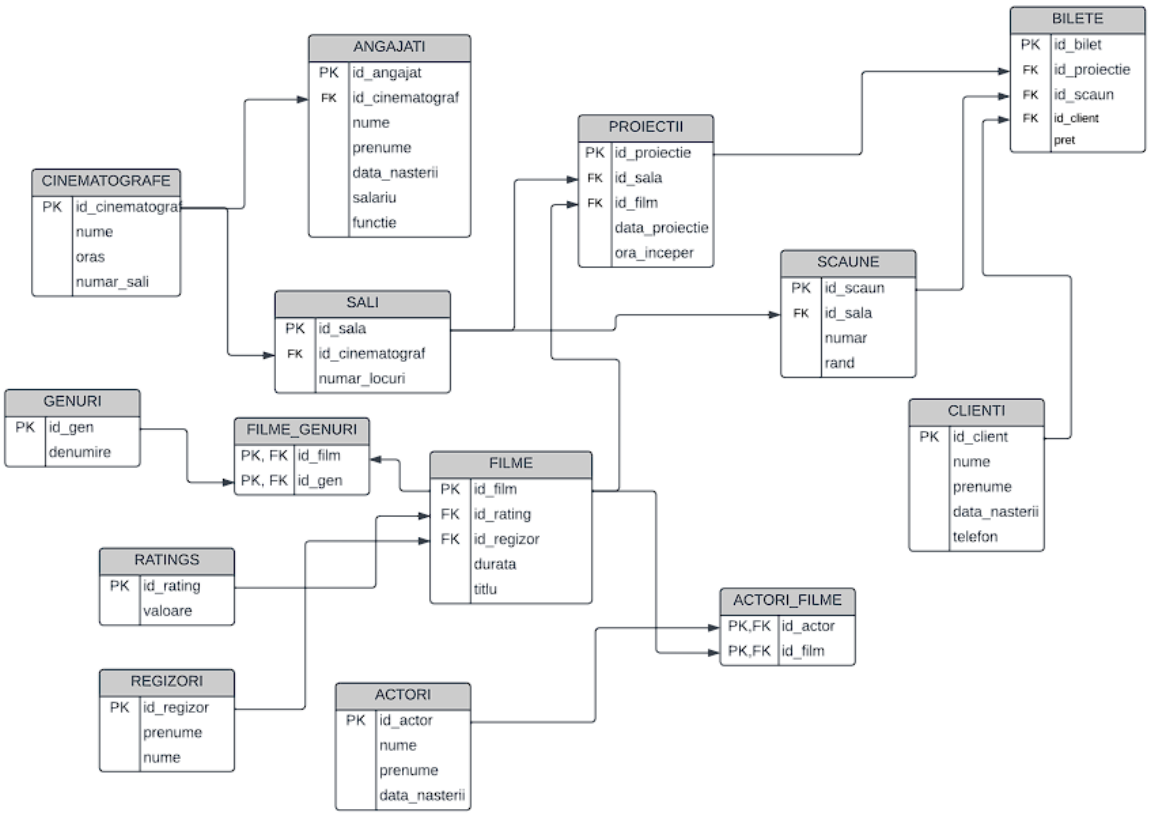
Această bază de date este concepută pentru a gestiona informațiile legate de industria cinematografică, permițând urmărirea și organizarea eficientă a datelor asociate cu filmele, regizorii, actorii, cinematografele, angajații, sălile de cinema, proiecțiile și vânzarea de bilete.
Câteva dintre utilizările majore ale acestei baze de date sunt:

- **Informații despre Filme și Regizori:** Păstrarea detaliilor despre filme, inclusiv titlu, durată, rating, genuri și regizori.
- **Gestionarea Actorilor și Rolurilor în Filme:** Asocierea actorilor cu filmele în care au jucat, inclusiv informații despre nume, prenume și data nașterii actorilor.
- **Gestionarea Cinematografelor și a Sălilor de Cinema:** Urmărirea informațiilor despre cinematografe, inclusiv nume, oraș, număr de săli și detaliile legate de fiecare sală.
- **Programarea proiecțiilor de filme:** Organizarea proiecțiilor, inclusiv detaliile despre sala de cinema, film, data și ora proiecției.
- **Vânzarea de Bilete:** Gestionarea procesului de vânzare a biletelor, inclusiv informații despre clienți, prețurile biletelor, locurile și proiecțiile la care s-au vândut biletele.
- **Angajați și Managementul Cinematografelor:** Urmărirea informațiilor despre angajații cinematografelor.

Ex 2 :Diagrama entitate-relație



Ex 3: Diagrama Conceptuală



Ex 4: Implementarea diagramei conceptuale (definire tabele)

-- CREARE TABELA REGIZORI
```
CREATE TABLE REGIZORI(
   ID_REGIZOR INT PRIMARY KEY,
   PRENUME VARCHAR2(20) NOT NULL,
   NUME VARCHAR2(20) NOT NULL
   );
```

-- CREARE TABELA RATINGS
```
CREATE TABLE RATINGS(
   ID_RATING INT PRIMARY KEY,
   VALOARE VARCHAR2(6) NOT NULL
   );
```

-- CREARE TABELA GENURI
```
CREATE TABLE GENURI(
   ID_GEN INT PRIMARY KEY,
   DENUMIRE VARCHAR2(20) NOT NULL
   );
```

-- CREARE TABELA FILME
```
CREATE TABLE FILME(
```

```
   ID_FILM INT PRIMARY KEY,
   TITLU VARCHAR2(50) NOT NULL,
   DURATA INT NOT NULL,
   ID_RATING INT,
   ID_REGIZOR INT,
   CONSTRAINT FK_FILME_RATINGS FOREIGN KEY (ID_RATING) REFERENCES RATINGS
(ID_RATING),
   CONSTRAINT FK_FILME_REGIZORI FOREIGN KEY (ID_REGIZOR) REFERENCES
REGIZORI (ID_REGIZOR)
);

-- CREARE TABELA ASOCIATIVA FILME_GENURI
CREATE TABLE FILME_GENURI(
   ID_FILM INT,
   ID_GEN INT,
   CONSTRAINT PK_FILME_GENURI PRIMARY KEY(ID_FILM, ID_GEN),
   CONSTRAINT FK_FILME_GENURI_FILME FOREIGN KEY (ID_FILM) REFERENCES
FILME (ID_FILM),
   CONSTRAINT FK_FILME_GENURI_GENURI FOREIGN KEY (ID_GEN) REFERENCES
GENURI (ID_GEN)
);

-- CREARE TABELA ACTORI
CREATE TABLE ACTORI (
   ID_ACTOR INT PRIMARY KEY,
   NUME VARCHAR2(20) NOT NULL,
   PRENUME VARCHAR2(20) NOT NULL,
   DATA_NASTERII DATE
   );

--CREARE TABELA ASOCIATIVA FILME_ACTORI
CREATE TABLE ACTORI_FILME
(
   ID_ACTOR INT,
   ID_FILM INT,
   CONSTRAINT PK_ACTORI_FILME PRIMARY KEY (ID_ACTOR, ID_FILM),
   CONSTRAINT FK_ACTORI_FILME_ACTORI FOREIGN KEY (ID_ACTOR) REFERENCES
ACTORI (ID_ACTOR),
   CONSTRAINT FK_ACTORI_FILME_FILME FOREIGN KEY (ID_FILM) REFERENCES
FILME (ID_FILM)
);

--CREARE TABELA CINEMATOGRAFE
CREATE TABLE CINEMATOGRAFE
(
   ID_CINEMATOGRAF INT PRIMARY KEY,
   NUME VARCHAR2(50),
   ORAS VARCHAR2(50) NOT NULL,
   NUMAR_SALI INT
);



--CREARE TABELA ANGAJATI
CREATE TABLE ANGAJATI
```

```
(
   ID_ANGAJAT        INT PRIMARY KEY,
   NUME              VARCHAR2 (100),
   PRENUME           VARCHAR2 (100),
   DATA_NASTERII     DATE,
   FUNCTIE           VARCHAR2 (50),
   SALARIU           INT,
   ID_CINEMATOGRAF    INT,
   CONSTRAINT FK_ANGAJAT_CINAMA FOREIGN KEY (ID_CINEMATOGRAF)
REFERENCES CINEMATOGRAFE (ID_CINEMATOGRAF)
);

--CREARE TABELA SALI
CREATE TABLE SALI
(
   ID_SALA INT PRIMARY KEY,
   NUMAR_LOCURI INT,
   ID_CINEMATOGRAF INT,
   CONSTRAINT FK_SALA_CINEMA FOREIGN KEY (ID_CINEMATOGRAF) REFERENCES
CINEMATOGRAFE (ID_CINEMATOGRAF)
);

--CREARE TABELA PRIECTII
CREATE TABLE PROIECTII
(
   ID_PROIECTIE INT PRIMARY KEY,
   ID_SALA INT,
   ID_FILM INT,
   DATA_PROIECTIE DATE,
   ORA_INCPERE TIMESTAMP,
   CONSTRAINT FK_PROIECTIE_SALA FOREIGN KEY (ID_SALA) REFERENCES SALI
(ID_SALA),
   CONSTRAINT FK_PREIECTIE_FILM FOREIGN KEY (ID_FILM) REFERENCES FILME
(ID_FILM)
);

-- CREARE TABELA SCACUNE
CREATE TABLE SCAUNE
(
   ID_SCAUN INT PRIMARY KEY,
   ID_SALA INT,
   NUMAR INT,
   RAND INT,
   CONSTRAINT FK_SCAUN_SALA FOREIGN KEY (ID_SALA) REFERENCES SALI
(ID_SALA)
   );

--CREARE TABELA CLIENTI
CREATE TABLE CLIENTI
(
   ID_CLIENT INT PRIMARY KEY,
   NUME VARCHAR2(50),
   PRENUME VARCHAR2(50),
   DATA_NASTERII DATE,
   TELEFON VARCHAR2(15) UNIQUE);
```

--CREARE TABELA BILETE

```
CREATE TABLE BILETE(
   ID_BILET INT PRIMARY KEY,
   PRET INT,
   ID_PROIECTIE INT,
   ID_SCAUN INT,
   ID_CLIENT INT,
   CONSTRAINT FK_BILET_CLIENT FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI
(ID_CLIENT),
   CONSTRAINT FK_BILET_SCAUN FOREIGN KEY (ID_SCAUN) REFERENCES SCAUNE
(ID_SCAUN),
   CONSTRAINT FK_BILET_PROIECTIE FOREIGN SKEY (ID_PROIECTIE) REFERENCES
PROIECTII (ID_PROIECTIE)

);
```

Ex 5: Adăugare informații in tabele

```
--Regizori

Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(1, 'Rober', 'Altman');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(2, 'Andrei', 'Tarkovsky');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(3, 'David', 'Lean');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(4, 'Shonda', 'Rimes');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(5, 'Buster', 'Keaton');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(6, 'Ben', 'Affleck');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(7, 'Andrei', 'Mihailescu');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(8, 'Dave', 'Franco');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(9, 'Kenneth', 'Branagh');
Insert Into Regizori(Id_Regizor, Prenume, Nume)
Values(10, 'Greta', 'Gerwig');



---Ratings
INSERT INTO RATINGS (ID_RATING, VALOARE)
values (1,'*');
INSERT INTO RATINGS (ID_RATING, VALOARE)
values (2,'**');
INSERT INTO RATINGS (ID_RATING, VALOARE)
values (3,'***');
INSERT INTO RATINGS (ID_RATING, VALOARE)
values (4,'****');
INSERT INTO RATINGS (ID_RATING, VALOARE)
values (5,'*****');



--Genuri
Insert into Genuri(id_gen,denumire)
values(1,'DRAMA');
Insert into Genuri(id_gen,denumire)
values(2,'ACTIUNE');
Insert into Genuri(id_gen,denumire)
values(3,'SCI-FI');
Insert into Genuri(id_gen,denumire)
values(4,'HORROR');
Insert into Genuri(id_gen,denumire)
```

```
values(5,'ANIMATIE');
Insert into Genuri(id_gen,denumire)
values(6,'COMEDIE');
Insert into Genuri(id_gen,denumire)
values(7,'DRAGOSTE');
Insert into Genuri(id_gen,denumire)
values(8,'THRILLER');
Insert into Genuri(id_gen,denumire)
values(9,'DOCUMENTAR');
--Actori
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (1, 'Johnny', 'Depp', TO_DATE('1963-06-09', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (2, 'Brad', 'Pitt', TO_DATE('1963-12-18', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (3, 'Tom', 'Hanks', TO_DATE('1956-07-09', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (4, 'Leonardo', 'DiCaprio', TO_DATE('1974-11-11', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (5, 'Robert', 'De Niro', TO_DATE('1943-08-17', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (6, 'Al', 'Pacino', TO_DATE('1940-04-25', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (7, 'Morgan', 'Freeman', TO_DATE('1937-06-01', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (8, 'Denzel', 'Washington', TO_DATE('1954-12-28', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (9, 'Russell', 'Crowe', TO_DATE('1964-04-07', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (10, 'Kevin', 'Spacey', TO_DATE('1959-07-26', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (11, 'Robert', 'Downey Jr.', TO_DATE('1965-04-04', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (12, 'Will', 'Smith', TO_DATE('1968-09-25', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (13, 'Ellen', 'Pompeo', TO_DATE('1969-11-10', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (14, 'Patrick', 'Dempsey', TO_DATE('1966-01-13', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (15, 'Sandra', 'Oh', TO_DATE('1971-07-20', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (16, 'Katherine', 'Heigl', TO_DATE('1978-11-24', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (17, 'Justin', 'Chambers', TO_DATE('1970-07-11', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (18, 'Helena', 'Bonham Carter', TO_DATE('1966-05-26', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (19, 'Orlando', 'Bloom', TO_DATE('1977-01-13', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (20, 'Keira', 'Knightley', TO_DATE('1985-03-26', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (21, 'Ian', 'Somerhalder', TO_DATE('1978-12-08', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (22, 'Paul', 'Wesley', TO_DATE('1982-07-23', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
VALUES (23, 'Nina', 'Dobrev', TO_DATE('1989-01-09', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
Values (24, 'Candice', 'King', TO_DATE('1987-05-13', 'YYYY-MM-DD'));
INSERT INTO ACTORI (ID_ACTOR, NUME, PRENUME, DATA_NASTERII)
Values (25, 'Kat', 'Graham', TO_DATE('1989-09-05', 'YYYY-MM-DD'));

--Filme

Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (1, 'Pirates of the Caribbean: The Curse of the Black Pearl', 143, 4, 1);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (2, 'Harry Potter and the Deathly Hallows: Part 2', 130, 4, 1);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (3, 'The Lord of the Rings: The Return of the King', 201, 4, 3);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (4, 'The Lord of the Rings: The Fellowship of the Ring', 178, 5, 3);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (5, 'Shadowhunters', 42, 4, 4);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (6, 'The Vampire Diaries', 43, 4, 5);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
```

```
VALUES (7, 'Avengers: Endgame', 181, 4, 6);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (8, 'Avengers: Infinity War', 149, 2, 6);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (9, 'The Godfather', 175, 5, 7);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (10, 'Avatar', 162, 4, 8);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (11, 'The Dark Knight', 152, 5, 9);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (12, 'The Dark Knight Rises', 164, 4, 9);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (13, 'The Wolf of Wall Street', 180, 2, 1);
Insert into Filme (ID_FILM, TITLU, DURATA, ID_RATING, ID_REGIZOR)
VALUES (14, 'The Departed', 151, 4, 10);

--Filme-genuri
Insert into Filme_genuri(id_film,id_gen)
values (1,1);
Insert into Filme_genuri(id_film,id_gen)
values (2,2);
Insert into Filme_genuri(id_film,id_gen)
values (3,3);
Insert into Filme_genuri(id_film,id_gen)
values (4,4);
Insert into Filme_genuri(id_film,id_gen)
values (5,5);
Insert into Filme_genuri(id_film,id_gen)
values (6,6);
Insert into Filme_genuri(id_film,id_gen)
values (7,7);
Insert into Filme_genuri(id_film,id_gen)
values (8,8);
Insert into Filme_genuri(id_film,id_gen)
values (9,9);
Insert into Filme_genuri(id_film,id_gen)
values (10,1);
Insert into Filme_genuri(id_film,id_gen)
values (11,2);
Insert into Filme_genuri(id_film,id_gen)
values (12,3);
Insert into Filme_genuri(id_film,id_gen)
values (13,4);
Insert into Filme_genuri(id_film,id_gen)
values (14,5);
Insert into Filme_genuri(id_film,id_gen)
values (1,6);
Insert into Filme_genuri(id_film,id_gen)
values (1,7);
Insert into Filme_genuri(id_film,id_gen)
values (2,8);
Insert into Filme_genuri(id_film,id_gen)
values (3,9);

--Filme-actori


Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (1, 1);
Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (2, 1);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (3, 1);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (4, 1);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (5, 1);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (6, 1);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (7, 2);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (8, 2);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (9, 2);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
```

```
VALUES (10, 2);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (11, 3);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (12, 3);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (13, 3);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (14, 3);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (15, 4);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (16, 4);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (17, 4);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (18, 4);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (19, 5);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES  (20, 5);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (21, 5);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (22, 5);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (23, 6);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (24, 6);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (1, 6);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUEs (2, 6);
Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (3, 7);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (4, 7);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (5, 7);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (6, 8);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (7, 8);

   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (8, 8);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (9, 9);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (10, 9);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (11, 9);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (12, 9);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (13, 10);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (14, 10);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (15, 11);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (16, 11);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (17, 11);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (18, 12);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUEs (19, 12);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (20, 12);
   Insert into  Actori_Filme (ID_ACTOR, ID_FILM)
VALUES  (21, 13);
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (22, 13);
   Insert into  Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (23, 14);
```

```
   Insert into Actori_Filme (ID_ACTOR, ID_FILM)
VALUES (24, 14);

--cinematografe
Insert into Cinematografe (id_cinematograf, nume, oras)
Values (1, 'Cinema City', 'Cluj-Napoca');
Insert into Cinematografe (id_cinematograf, nume, oras)
Values (2, 'Cinema City', 'Bucuresti');
   Insert into Cinematografe (id_cinematograf, oras)
Values (3,  'Timisoara');
   Insert into Cinematografe (id_cinematograf, nume, oras)
Values (4, 'Cinema Magic', 'Iasi');
   Insert into Cinematografe (id_cinematograf, nume, oras)
Values (5, 'Cinema City', 'Constanta',);
   Insert into Cinematografe (id_cinematograf, nume, oras)
Values (6, 'Cinema City', 'Brasov');
   Insert into Cinematografe (id_cinematograf, nume, oras)
Values (7, 'Cinema City', 'Sibiu');
   Insert into Cinematografe (id_cinematograf, nume, oras)
Values  (8, 'Cinema City', 'Oradea');
select * from cinematografe;
--sali

Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (1, 1, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (2, 1, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (3, 1, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (4, 1, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (5, 1, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (6, 1, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (7, 1, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (8, 1, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (9, 1, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (10, 1, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (11, 2, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (12, 2, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (13, 2, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (14, 2, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (15, 2, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (16, 2, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (17, 2, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (18, 2, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (19, 2, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (20, 2, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (21, 2, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (22, 2, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (23, 2, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (24, 2, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (25, 2, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (26, 3, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
```

```
values (27, 3, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (28, 3, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (29, 3, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (30, 3, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (31, 3, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (32, 3, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (33, 3, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (34, 3, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (35, 3, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (36, 3, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (37, 3, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (38, 3, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (39, 3, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (40, 3, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (41, 4, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (42, 4, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (43, 4, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (44, 4, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (45, 5, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (46, 5, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (47, 5, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (48, 5, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (49, 6, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (50, 6, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (51, 6, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (52, 6, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (53, 6, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (54, 6, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (55, 6, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (56, 6, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (57, 6, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (58, 6, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (59, 6, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (60, 6, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (61, 7, 100);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (62, 7, 110);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (63, 7, 120);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (64, 7, 130);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
```

```
values (65,8, 140);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (66, 8, 150);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (67, 8, 160);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (68, 8, 170);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (69, 8, 180);
   Insert into Sali (id_sala, id_cinematograf, numar_locuri)
values (70, 8, 110);

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (1, 1, 'Popescu', 'Ion', TO_DATE('1990-01-01', 'YYYY-MM-DD'), 2000, 'Casier');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (2, 1, 'Ionescu', 'Maria', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 2500, 'Vanator de bilete');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (3, 2, 'Dumitrescu', 'Elena', TO_DATE('1988-07-20', 'YYYY-MM-DD'), 2200, 'Agent de securitate');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (4, 3, 'Vasilescu', 'Andrei', TO_DATE('1992-09-12', 'YYYY-MM-DD'), 2300, 'Tehnician proiectie');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (5, 2, 'Mihai', 'Cristina', TO_DATE('1994-03-25', 'YYYY-MM-DD'), 2100, 'Asistent manager');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (6, 1, 'Constantinescu', 'Ana', TO_DATE('1991-12-08', 'YYYY-MM-DD'), 2400, 'Operator sonor');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (7, 3, 'Gheorghe', 'Mihai', TO_DATE('1987-06-14', 'YYYY-MM-DD'), 2600, 'Manager evenimente');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (8, 2, 'Stanescu', 'Adriana', TO_DATE('1993-04-30', 'YYYY-MM-DD'), 1900, 'Curier');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (9, 1, 'Diaconu', 'Dan', TO_DATE('1989-11-18', 'YYYY-MM-DD'), 2800, 'Director cinematograf');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie
VALUES (10, 3, 'Stoica', 'Larisa', TO_DATE('1995-08-22', 'YYYY-MM-DD'), 2000, 'Casier');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (11, 1, 'Nistor', 'Alexandru', TO_DATE('1986-02-05', 'YYYY-MM-DD'), 2100, 'Agent de paza');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (12, 2, 'Popa', 'Andreea', TO_DATE('1994-07-19', 'YYYY-MM-DD'), 2300, 'Operator lumini');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (13, 3, 'Radu', 'Catalina', TO_DATE('1990-03-10', 'YYYY-MM-DD'), 2500, 'Vanator de bilete');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (14, 1, 'Dinu', 'Victor', TO_DATE('1988-09-28', 'YYYY-MM-DD'), 2200, 'Coordonator proiecte');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (15, 2, 'Avram', 'Mara', TO_DATE('1993-05-16', 'YYYY-MM-DD'), 2700, 'Asistent director');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (16, 3, 'Moldovan', 'Razvan', TO_DATE('1987-12-03', 'YYYY-MM-DD'), 2400, 'Operator proiectie');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (17, 1, 'Dobre', 'Simona', TO_DATE('1991-10-14', 'YYYY-MM-DD'), 1900, 'Asistent manager');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (18, 2, 'Nedelcu', 'Andrei', TO_DATE('1996-04-27', 'YYYY-MM-DD'), 2000, 'Curier');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (19, 3, 'Barbu', 'Elena', TO_DATE('1989-06-08', 'YYYY-MM-DD'), 2100, 'Casier');
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (20, 1, 'Dumitrache', 'Raluca', TO_DATE('1992-08-17', 'YYYY-MM-DD'), 2200, 'Agent de securitate');


INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (21, 4, 'Cristea', 'Andrei', TO_DATE('1993-11-05', 'YYYY-MM-DD'), 2300, 'Operator proiectie');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (22, 4, 'Gheorghiu', 'Maria', TO_DATE('1987-09-25', 'YYYY-MM-DD'), 2600, 'Manager cinematograf');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (23, 4, 'Dumitru', 'Elena', TO_DATE('1992-04-15', 'YYYY-MM-DD'), 2100, 'Casier');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (24, 4, 'Munteanu', 'Gabriel', TO_DATE('1996-08-12', 'YYYY-MM-DD'), 1900, 'Asistent manager');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (25, 4, 'Antonescu', 'Cristina', TO_DATE('1991-06-30', 'YYYY-MM-DD'), 2400, 'Tehnician sunet');


INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (26, 5, 'Iancu', 'Adrian', TO_DATE('1990-02-15', 'YYYY-MM-DD'), 2200, 'Agent de securitate');


INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (27, 5, 'Pop', 'Elena', TO_DATE('1984-07-10', 'YYYY-MM-DD'), 2000, 'Casier');
```

```
INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (28, 5, 'Mocanu', 'Alexandra', TO_DATE('1989-12-20', 'YYYY-MM-DD'), 2500, 'Vanator de bilete');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (29, 5, 'Georgescu', 'Ionut', TO_DATE('1994-03-01', 'YYYY-MM-DD'), 2300, 'Tehnician proiectie');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (30, 5, 'Radu', 'Andreea', TO_DATE('1985-11-18', 'YYYY-MM-DD'), 2800, 'Director cinematograf');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (31, 6, 'Dinu', 'Raluca', TO_DATE('1992-10-05', 'YYYY-MM-DD'), 2100, 'Agent de paza');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (32, 6, 'Vasile', 'Mihai', TO_DATE('1983-05-20', 'YYYY-MM-DD'), 2400, 'Operator lumini');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (33, 6, 'Ilie', 'Catalina', TO_DATE('1991-01-10', 'YYYY-MM-DD'), 2500, 'Vanator de bilete');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (34, 6, 'Balasoiu', 'Victor', TO_DATE('1988-08-28', 'YYYY-MM-DD'), 2200, 'Coordonator proiecte');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (35, 6, 'Popescu', 'Mara', TO_DATE('1993-07-16', 'YYYY-MM-DD'), 2700, 'Asistent director');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (36, 7, 'Pavel', 'Alexandru', TO_DATE('1987-12-03', 'YYYY-MM-DD'), 2400, 'Operator proiectie');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (37, 7, 'Nicolae', 'Simona', TO_DATE('1991-10-14', 'YYYY-MM-DD'), 1900, 'Asistent manager');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (38, 7, 'Cristina', 'Andrei', TO_DATE('1996-04-27', 'YYYY-MM-DD'), 2000, 'Curier');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (39, 7, 'Georgiana', 'Elena', TO_DATE('1989-06-08', 'YYYY-MM-DD'), 2100, 'Casier');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (40, 7, 'Dinca', 'Raluca', TO_DATE('1992-08-17', 'YYYY-MM-DD'), 2200, 'Agent de securitate');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (41, 8, 'Simion', 'Andrei', TO_DATE('1991-06-10', 'YYYY-MM-DD'), 2600, 'Manager evenimente');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (42, 8, 'Irina', 'Adriana', TO_DATE('1994-12-30', 'YYYY-MM-DD'), 2000, 'Curier');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (43, 8, 'Marian', 'Dan', TO_DATE('1986-08-18', 'YYYY-MM-DD'), 2400, 'Operator sonor');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (44, 8, 'Valentina', 'Mihai', TO_DATE('1993-01-25', 'YYYY-MM-DD'), 2100, 'Agent de paza');

INSERT INTO Angajati (id_angajat, id_cinematograf, nume, prenume, data_nasterii, salariu, functie)
VALUES (45, 8, 'Ionescu', 'Andreea', TO_DATE('1995-07-12', 'YYYY-MM-DD'), 2300, 'Operator lumini');

--Proiectii
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (1, 1, 1, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (2, 2, 2, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('14:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (3, 3, 3, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('16:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (4, 6, 4, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('18:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (5, 7, 5, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('20:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (6, 8, 6, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('22:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (7, 9, 7, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (8, 10, 8, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('14:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (9, 11, 9, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('16:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (10, 12, 10, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('18:00', 'HH24:MI'));
```

```
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (11, 13, 11, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('20:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (12, 14, 12, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('22:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (13, 15, 13, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (14, 16, 14, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('14:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (15, 17, 1, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('16:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (16, 18, 2, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('18:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (17, 19, 3, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('20:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (18, 20, 4, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('22:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (19, 21, 5, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (20, 22, 6, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('14:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (21, 23, 7, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('16:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (22, 24, 8, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('18:00', 'HH24:MI'));

   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (23, 25, 9, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('20:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (24, 26, 10, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('22:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (25, 27, 11, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (26, 28, 12, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('14:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (27, 29, 13, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('16:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (28, 30, 14, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('18:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (29, 31, 1, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('20:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (30, 32, 2, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('22:00', 'HH24:MI'));
   Insert into proiectii (id_proiectie, id_sala, id_film, data_proiectie, ora_incepere)
values (31, 33, 3, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('12:00', 'HH24:MI'));

--SCAUNE

Insert into scaune (id_scaun, id_sala, numar, rand)
values (1, 1, 1, 1);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (2, 1, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (3, 1, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (4, 1, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (5, 1, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (6, 1, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (7, 1, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (8, 1, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (9, 1, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (10, 1, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (11, 1, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (12, 1, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (13, 1, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (14, 1, 14, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (15, 1, 15, 3);
```

```
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (16, 1, 16, 4);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (17, 1, 17, 4);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (18, 1, 18, 4);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (19, 2, 1, 1);
   Insert into scaune  (id_scaun, id_sala, numar, rand)
values (20, 2, 2, 1);
   InSert into scaune (id_scaun, id_sala, numar, rand)
values (21, 2, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (22, 2, 4, 1);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (23 , 2, 5, 1);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (24, 2, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (25, 2, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (26, 2, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (27, 2, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (28, 2, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (29, 2, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (30, 3, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values  (31, 3, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (32, 3, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (33, 3, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (34, 3, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (35, 3, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (36, 3, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (37, 3, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (38, 3, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (39, 3, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (40, 3, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (41, 3, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (42, 3, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (43, 3, 14, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (44, 3, 15, 3);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (45, 4, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (46, 4, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (47, 4, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (48, 4, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (49, 4, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (50, 4, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (51, 4, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (52, 4, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (53, 4, 9, 2);
```

```
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (54, 4, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (55, 4, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (56, 4, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (57, 4, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (58, 5, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (59, 5, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (60, 5, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (61, 5, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (62, 5, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (63, 5, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (64, 5, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (65, 5, 8, 2);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (66, 5, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (67, 5, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (68, 5, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (69, 5, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (70, 5, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (71, 5, 14, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (72, 5, 15, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (73, 6, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (74, 6, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (75, 6, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (76, 6, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (77, 6, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (78, 6, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (79, 6, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (80, 6, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (81, 6, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (82, 6, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (83, 6, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (84, 6, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (85, 6, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (86, 6, 14, 3);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (87, 6, 15, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (88, 7, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (89, 7, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (90, 7, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (91, 7, 4, 1);
```

```
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (92, 7, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (93, 7, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (94, 7, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (95, 7, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (96, 7, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (97, 7, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (98, 7, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (99, 7, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (100, 7, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (101, 7, 14, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (102, 7, 15, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (103, 8, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (104, 8, 2, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (105, 8, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (106, 8, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (107, 8, 5, 1);
Insert into scaune (id_scaun, id_sala, numar, rand)
values (108, 8, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (109, 8, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (110, 8, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (111, 8, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (112, 8, 10, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (113, 8, 11, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (114, 8, 12, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (115, 8, 13, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (116, 8, 14, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (117, 8, 15, 3);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (118, 9, 1, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (119, 9, 2, 1);
   Insert into scaune  (id_scaun, id_sala, numar, rand)
values (120, 9, 3, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (121, 9, 4, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (122, 9, 5, 1);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (123, 9, 6, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (124, 9, 7, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (125, 9, 8, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (126, 9, 9, 2);
   Insert into scaune (id_scaun, id_sala, numar, rand)
values (127, 9, 10, 2);

--CLIENTI
INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
```

(1, 'Popescu', 'Ion', TO_DATE('1990-01-01', 'YYYY-MM-DD'), '0721123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(2, 'Ionescu', 'Maria', TO_DATE('1985-05-15', 'YYYY-MM-DD'), '0732123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(3, 'Dumitrescu', 'Elena', TO_DATE('1988-07-20', 'YYYY-MM-DD'), '0743123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(4, 'Vasilescu', 'Andrei', TO_DATE('1992-09-12', 'YYYY-MM-DD'), '0754123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(5, 'Mihai', 'Cristina', TO_DATE('1994-03-25', 'YYYY-MM-DD'), '0765123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(6, 'Constantinescu', 'Ana', TO_DATE('1991-12-08', 'YYYY-MM-DD'), '0776123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(7, 'Gheorghe', 'Mihai', TO_DATE('1987-06-14', 'YYYY-MM-DD'), '0787123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(8, 'Stanescu', 'Adriana', TO_DATE('1993-04-30', 'YYYY-MM-DD'), '0798123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(9, 'Diaconu', 'Dan', TO_DATE('1989-11-18', 'YYYY-MM-DD'), '0809123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(10, 'Stoica', 'Larisa', TO_DATE('1995-08-22', 'YYYY-MM-DD'), '0810123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(11, 'Nistor', 'Alexandru', TO_DATE('1986-02-05', 'YYYY-MM-DD'), '0821123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(12, 'Popa', 'Andreea', TO_DATE('1994-07-19', 'YYYY-MM-DD'), '0832123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(13, 'Radu', 'Catalina', TO_DATE('1990-03-10', 'YYYY-MM-DD'), '0843123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(14, 'Dinu', 'Victor', TO_DATE('1988-09-28', 'YYYY-MM-DD'), '0854123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(15, 'Avram', 'Mara', TO_DATE('1993-05-16', 'YYYY-MM-DD'), '0865123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(16, 'Moldovan', 'Razvan', TO_DATE('1987-12-03', 'YYYY-MM-DD'), '0876123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(17, 'Dobre', 'Simona', TO_DATE('1991-10-14', 'YYYY-MM-DD'), '0887123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(18, 'Nedelcu', 'Andrei', TO_DATE('1996-04-27', 'YYYY-MM-DD'), '0898123456');

INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(19, 'Barbu', 'Elena', TO_DATE('1989-06-08', 'YYYY-MM-DD'), '0909123456');

```
INSERT INTO CLIENTI (ID_CLIENT, NUME, PRENUME, DATA_NASTERII, TELEFON)
VALUES
(20, 'Dumitrache', 'Raluca', TO_DATE('1992-08-17', 'YYYY-MM-DD'), '0910123456');

--bILETE
INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (1, 1, 1, 1, 20);
INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (2, 2, 2, 2, 20);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (3, 3, 3, 3, 20);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (4, 4, 4, 4, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (5, 5, 5, 5, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (6, 6, 6, 6, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (7, 7, 7, 7, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (8, 8, 8, 8, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (9, 9, 9, 9, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (10, 10, 10, 10, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (11, 11, 11, 11, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (12, 12, 12, 12, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (13, 13, 13, 13, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (14, 14, 14, 14, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (15, 15, 15, 15, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (16, 16, 16, 16, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (17, 1, 17, 17, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (18, 2, 18, 18, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (19, 3, 19, 19, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (20, 4, 20, 20, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (21, 5, 21, 1, 21);
INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (22, 6, 22, 2, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (23, 7, 23, 3, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (24, 8, 24, 4, 21);
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (25, 9, 25, 5, 21) ;
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (26, 10, 26, 6, 21) ;
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (27, 11, 27, 7, 21) ;
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (28, 12, 28, 8, 21) ;
   INSERT INTO BILETE (ID_BILET, ID_PROIECTIE, ID_SCAUN, ID_CLIENT, PRET)
VALUES (29, 13, 29, 9, 21) ;
```

Ex 6: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

Cerința: Pentru un film dat (prin id) să se afișeze:

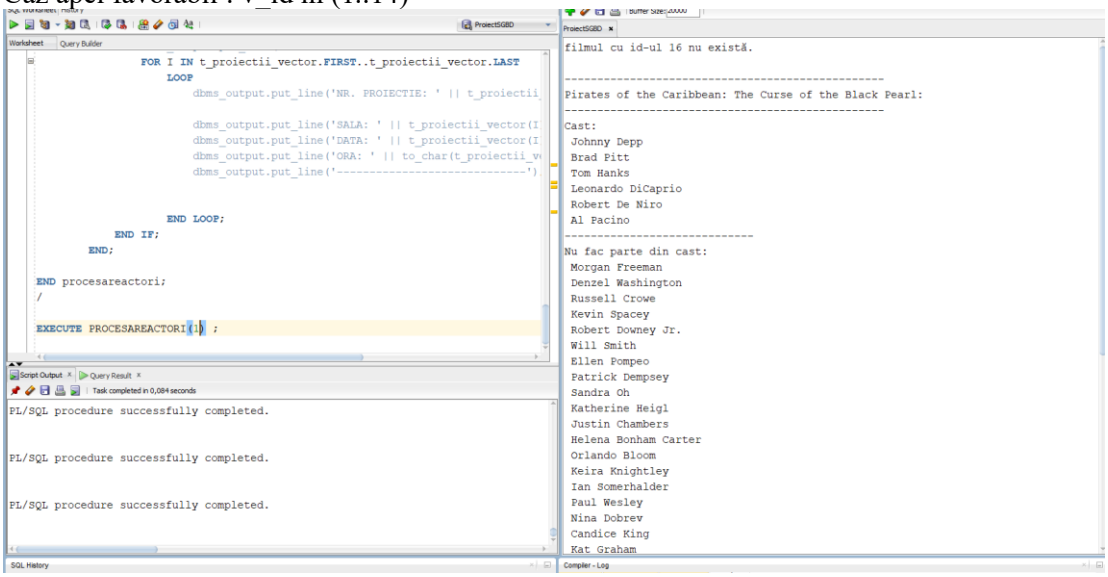
a) Lista de actori care joaca in acel film. – Tabel indexat
b) Lista de actori care nu joaca in film – Tabel Imbricat
c) Programările filmului – Vector

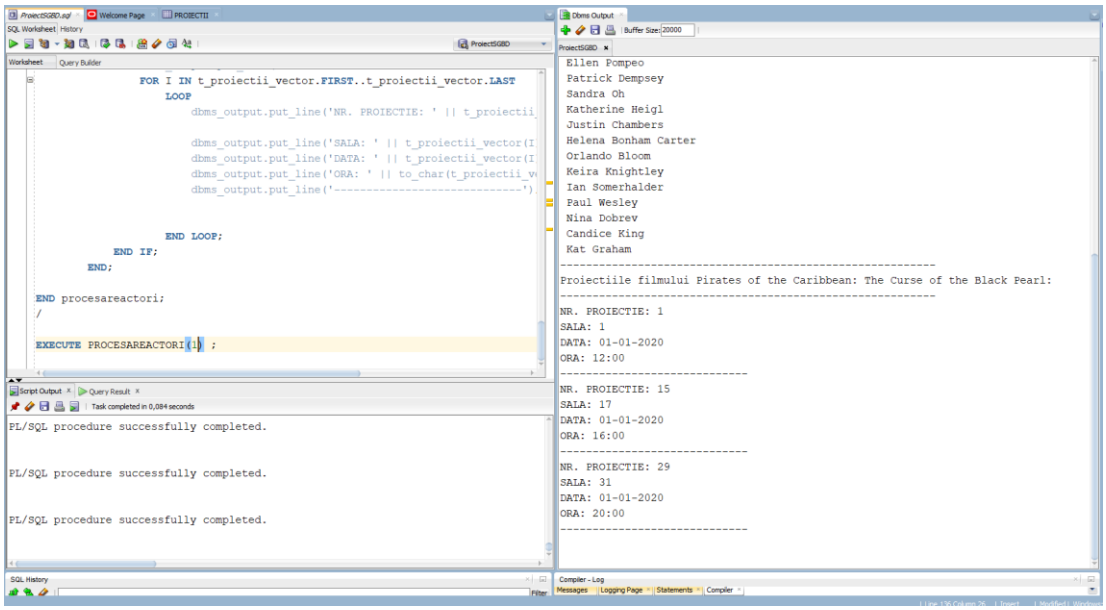
```
CREATE OR REPLACE PROCEDURE procesareactori(v_id filme.id_film%TYPE)
AS
   --record pentru a memora informatiile despre actori
   --folosit pt tabelul indexat si cel imbricat
   TYPE infoactori IS RECORD
                  (
                     actor_id actori.id_actor%TYPE,
                     nume     actori.nume%TYPE,
                     prenume  actori.prenume%TYPE
                  );
   -- record pentru a memora info despre proiectiile filmului
   TYPE infoproiectie IS RECORD
                    (
                       id_proiectie   proiectii.id_proiectie%TYPE,
                       ora            proiectii.ora_incepere%TYPE,
                       data_proiectie proiectii.data_proiectie%TYPE,
                       sala           proiectii.id_sala%TYPE
                    );
   nume_film          VARCHAR2(100); --numele filmului cu id_ul v_id

   --tabela indexat pentru memorarea actorilor care joaca in film
   TYPE actori_index_table IS TABLE OF infoactori INDEX BY PLS_INTEGER;
   t_actori_index     actori_index_table;

   --tabel imbricat pt a memora actorii care NU joaca in film
   TYPE actori_imbricat IS TABLE OF infoactori;
   t_actori_imbricat  actori_imbricat  := actori_imbricat();

   --varray pentru a retine proiectiile filmului
   TYPE proiectii_vector IS VARRAY(100) OF infoproiectie;
   t_proiectii_vector proiectii_vector := proiectii_vector();
BEGIN
   BEGIN
      -- Verificare daca exista filmul cu ID-ul dat

      SELECT titlu
      INTO nume_film
      FROM filme
      WHERE id_film = v_id;
       EXCEPTION
      WHEN no_data_found THEN
          dbms_output.put_line('filmul cu id-ul ' || V_ID || ' nu exista.');
      RETURN;
       END;


      ---memorare informatii despre actorii care joaca in filmul dat in tabelul indexat
      BEGIN
         SELECT A.id_actor,
              A.nume,
              A.prenume BULK COLLECT
```

```
INTO t_actori_index
FROM actori A
        JOIN actori_filme af ON af.id_actor = A.id_actor
        JOIN filme F ON F.id_film = af.id_film
WHERE F.id_film = v_id;

---memorare in tabelul imbricat a actorilor care nu joaca in film
FOR I IN (SELECT A.id_actor, A.nume, A.prenume
         FROM actori A
         WHERE A.id_actor NOT IN (SELECT a1.id_actor
                                  FROM actori a1
                                           JOIN actori_filme af ON af.id_actor = a1.id_actor
                                  WHERE af.id_film = v_id))
   LOOP
      t_actori_imbricat.extend;
      t_actori_imbricat(t_actori_imbricat.LAST).actor_id := I.id_actor;
      t_actori_imbricat(t_actori_imbricat.LAST).nume := I.nume;
      t_actori_imbricat(t_actori_imbricat.LAST).prenume := I.prenume;
   END LOOP;

FOR I IN (SELECT id_proiectie, ora_incepere, data_proiectie, id_sala
         FROM proiectii
         WHERE id_film = v_id)
   LOOP
      t_proiectii_vector.extend;
      t_proiectii_vector(t_proiectii_vector.LAST).id_proiectie := I.id_proiectie;
      t_proiectii_vector(t_proiectii_vector.LAST).ora := I.ora_incepere;
      t_proiectii_vector(t_proiectii_vector.LAST).data_proiectie := I.data_proiectie;
      t_proiectii_vector(t_proiectii_vector.LAST).sala := I.id_sala;
   END LOOP;
dbms_output.put_line('-------------------------------------------------');
dbms_output.put_line(nume_film || ':');
dbms_output.put_line('-------------------------------------------------');

IF t_actori_index.COUNT = 0 THEN
   dbms_output.put_line('-----------------------------');
   dbms_output.put_line(nume_film || ':');
   dbms_output.put_line('Nu exista actori pentru filmul cu ID-ul ' || v_id);
   dbms_output.put_line('-----------------------------');
ELSE
   dbms_output.put_line('Cast:');
   FOR I IN t_actori_index.FIRST..t_actori_index.LAST
      LOOP
         dbms_output.put_line(' ' || t_actori_index(I).nume || ' ' || t_actori_index(I).prenume);
      END LOOP;
END IF;

IF t_actori_imbricat.COUNT = 0 THEN
   dbms_output.put_line('-----------------------------');
   dbms_output.put_line('TOTI ACTORII SUNT IN CAST');
   dbms_output.put_line('-----------------------------');
ELSE
   dbms_output.put_line('-----------------------------');
   dbms_output.put_line('Nu fac parte din cast:');
   FOR I IN t_actori_imbricat.FIRST..t_actori_imbricat.LAST
```

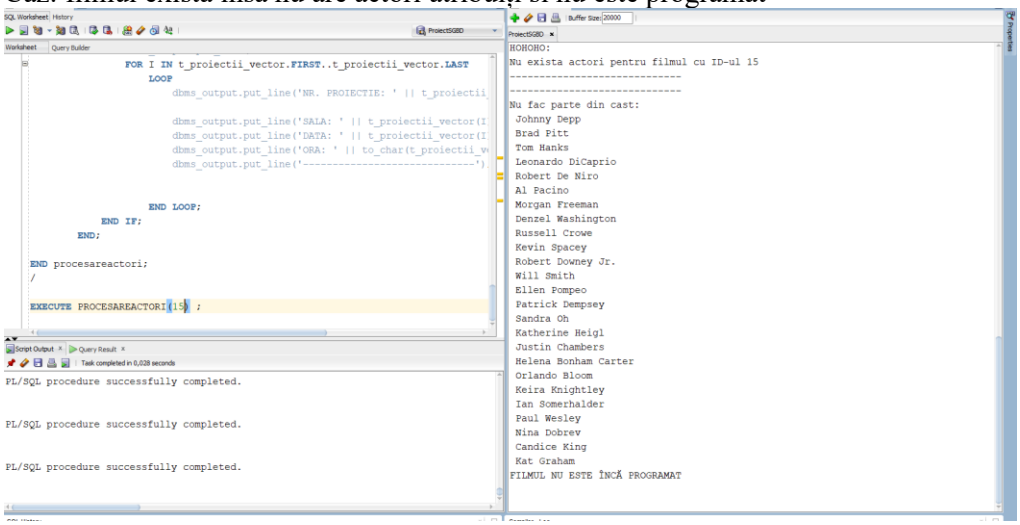
```
            LOOP
                dbms_output.put_line(' ' || t_actori_imbricat(I).nume || ' ' ||
t_actori_imbricat(I).prenume);
            END LOOP;
        END IF;


        IF t_proiectii_vector.COUNT = 0 THEN
           dbms_output.put_line('FILMUL NU ESTE INCA PROGRAMAT');
        ELSE
           dbms_output.put_line('-----------------------------------------------------------');
           dbms_output.put_line('Proiectiile filmului ' || nume_film || ': ');
           dbms_output.put_line('-----------------------------------------------------------');
           FOR I IN t_proiectii_vector.FIRST..t_proiectii_vector.LAST
              LOOP
                 dbms_output.put_line('NR. PROIECTIE: ' || t_proiectii_vector(I).id_proiectie || ' ');

                 dbms_output.put_line('SALA: ' || t_proiectii_vector(I).sala);
                 dbms_output.put_line('DATA: ' || t_proiectii_vector(I).data_proiectie);
                 dbms_output.put_line('ORA: ' || to_char(t_proiectii_vector(I).ora, 'HH24:MI'));
                 dbms_output.put_line('------------------------------');


              END LOOP;
        END IF;
      END;

END procesareactori;
/
```

Caz apel favorabil : v_id in (1..14)

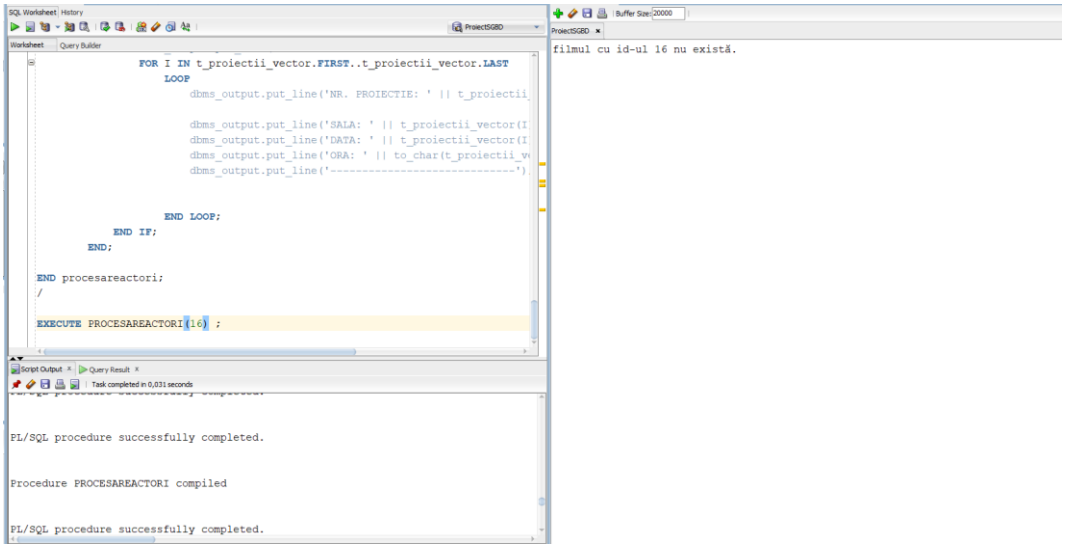




Caz: filmul există însă nu are actori atribuiți si nu este programat



Caz: Filmul nu exista

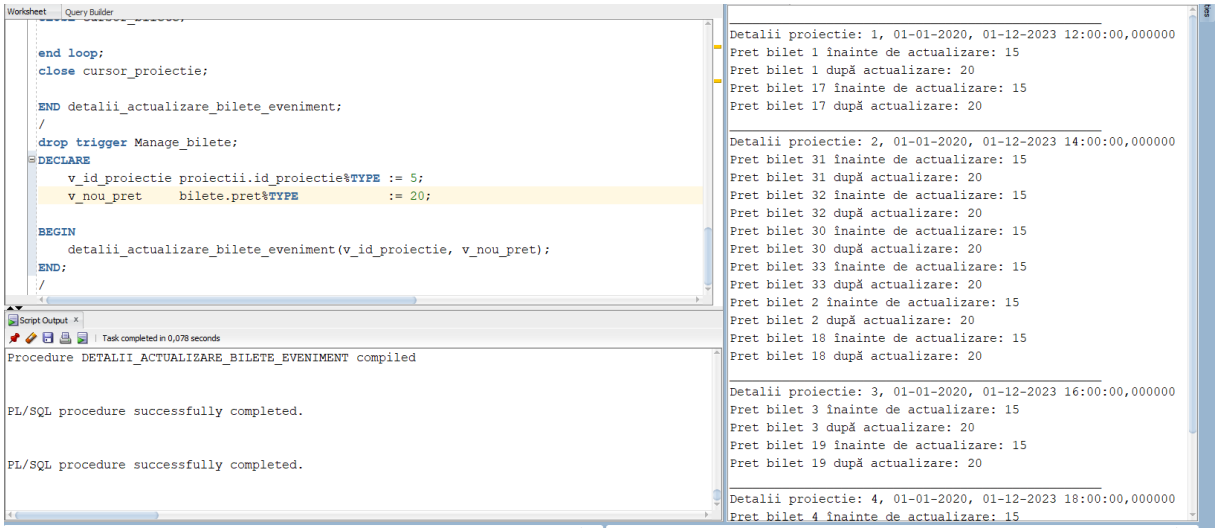


Ex 7: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul

Cerința: Pentru o proiectie data prin id_ul său, actualizați prețul fiecărui bilet cumpărat.

```
CREATE OR REPLACE PROCEDURE detalii_actualizare_bilete_eveniment(
   p_id_proiectie IN proiectii.id_proiectie%TYPE,
   p_nou_pret IN bilete.pret%TYPE
) AS
   -- Cursor pentru obtinerea informatiilor despre proiectie
   CURSOR cursor_proiectie IS
      SELECT *
      FROM proiectii
      WHERE id_proiectie < p_id_proiectie;

   proiectie_rec proiectii%ROWTYPE;

   -- Cursor parametrizat pentru obtinerea si blocarea detaliilor biletelor pentru proiectie
   CURSOR cursor_bilete(p_id_proiectie_param proiectii.id_proiectie%TYPE) IS
      SELECT *
      FROM bilete
      WHERE id_proiectie = p_id_proiectie_param
      FOR UPDATE OF pret; -- Actualizăm doar câmpul PRET

   bilet_rec bilete%ROWTYPE;

BEGIN
   -- Deschide cursorul pentru informatii despre proiectie
   OPEN cursor_proiectie;
   loop
   FETCH cursor_proiectie INTO proiectie_rec;
   EXIT WHEN cursor_proiectie%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('_____________________________________________________________');
   DBMS_OUTPUT.PUT_LINE('Detalii proiectie: ' || proiectie_rec.id_proiectie || ', ' ||
proiectie_rec.data_proiectie || ', ' || proiectie_rec.ora_incepere);

   -- Deschide cursorul parametrizat pentru detaliile biletelor si actualizează PRET-ul
OPEN cursor_bilete(proiectie_rec.id_proiectie);

-- Parcurge cursorul pentru fiecare înregistrare
LOOP
   FETCH cursor_bilete INTO bilet_rec;

   -- Iesire din bucla când nu mai există înregistrări
   EXIT WHEN cursor_bilete%NOTFOUND;

   DBMS_OUTPUT.PUT_LINE('Pret bilet ' || bilet_rec.id_bilet || ' înainte de actualizare: ' ||
bilet_rec.pret);

   -- Actualizare PRET
```

```
    UPDATE bilete
    SET pret = p_nou_pret
    WHERE CURRENT OF cursor_bilete;

    DBMS_OUTPUT.PUT_LINE('Pret bilet ' || bilet_rec.id_bilet || ' după actualizare: ' || p_nou_pret);
END LOOP;

-- Închide cursorul parametrizat pentru detaliile biletelor
CLOSE cursor_bilete;

end loop;
close cursor_proiectie;

END detalii_actualizare_bilete_eveniment;
/
drop trigger Manage_bilete;
DECLARE
    v_id_proiectie proiectii.id_proiectie%TYPE := 5;
    v_nou_pret     bilete.pret%TYPE           := 20;

BEGIN
    detalii_actualizare_bilete_eveniment(v_id_proiectie, v_nou_pret);
END;
/
```



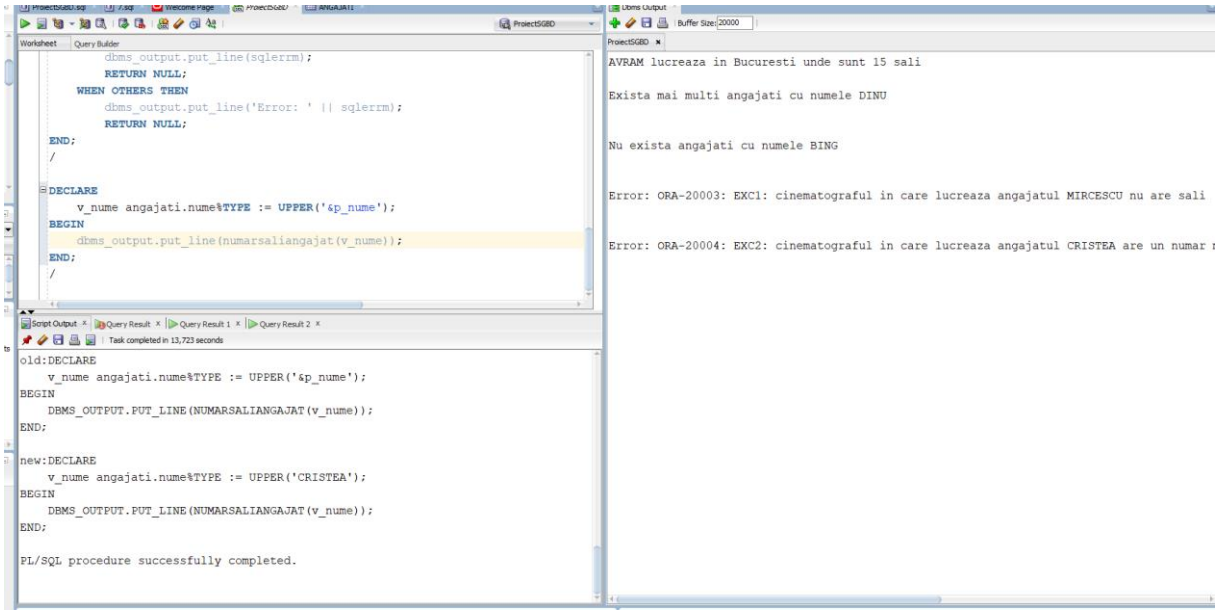
Ex 8: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

Cerință: Să se determine orașul și numărul de săli ale cinematografului în care lucrează un angajat al cărui nume este dat de la tastatură. În cazul în care cinematograful nu are săli funcționale atunci se va

propune închiderea acestuia, iar în cazul în care cinematograful are mai puțin de 5 săli se va recomanda redistribuirea angajatului la un alt cinematograf.

```
CREATE OR REPLACE FUNCTION numarsaliangajat(numele angajati.nume%TYPE)
   RETURN VARCHAR2
   IS
   rez   VARCHAR2(150) := '';
   orass cinematografe.oras%TYPE;
   nr    NUMBER;
   exc1 EXCEPTION; --exceptie pt cazul cu 0 sali
   exc2 EXCEPTION; --exceptie pt cazul cu <=4 sali
BEGIN
   SELECT nvl(COUNT(S.id_sala), 0) AS num_sala, C.oras
   INTO nr, orass
   FROM angajati A
         JOIN cinematografe C ON A.id_cinematograf = C.id_cinematograf
         LEFT JOIN sali S ON C.id_cinematograf = S.id_cinematograf --pt cazul in care nu exista sali
   --in cinematograful in care lucreaza
   WHERE UPPER(A.nume) = UPPER(numele)
   GROUP BY C.oras;
   IF nr = 0 THEN
      raise_application_error(-20003, 'EXC1: cinematograful in care lucreaza angajatul ' || numele ||
                              ' nu are sali. Se recomanda inchiderea');
   ELSIF nr < 5 THEN
      raise_application_error(-20004, 'EXC2: cinematograful in care lucreaza angajatul ' || numele ||
                              ' are un numar mic de sali. Se recomanda redistribuirea angajatului la alt
cinematograf');
   END IF;

   rez := numele || ' lucreaza in ' || orass || ' unde sunt ' || to_char(nr) || ' sali' || rez;

   RETURN rez;
EXCEPTION
   WHEN too_many_rows THEN
      dbms_output.put_line('Exista mai multi angajati cu numele ' || numele);
      RETURN NULL;
   WHEN no_data_found THEN
      dbms_output.put_line('Nu exista angajati cu numele ' || numele);
      RETURN NULL;
   WHEN exc1 THEN
      dbms_output.put_line(sqlerrm);
      RETURN NULL;
   WHEN exc2 THEN
      dbms_output.put_line(sqlerrm);
      RETURN NULL;
   WHEN OTHERS THEN
      dbms_output.put_line('Error: ' || sqlerrm);
      RETURN NULL;
END;
/

DECLARE
   v_nume angajati.nume%TYPE := UPPER('&p_nume');
BEGIN
   dbms_output.put_line(numarsaliangajat(v_nume));
```

END;
/



Ex 9: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

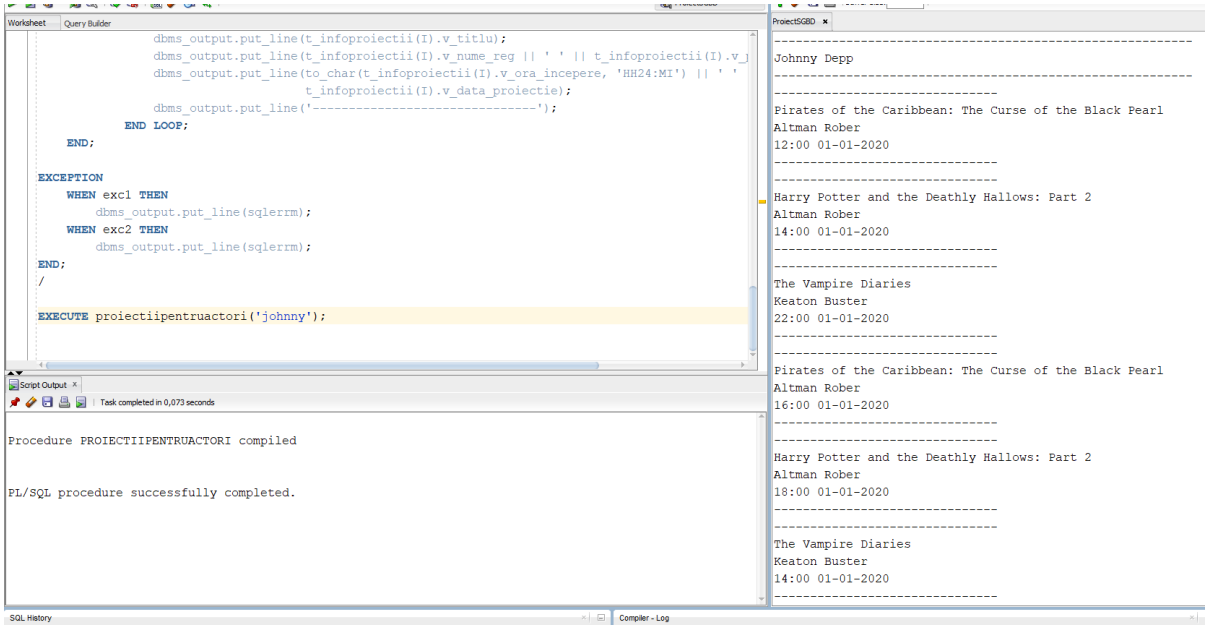
Cerinta: Pentru un actor al carui nume este dat la tastatura, sa se afiseze proiectiile in care acesta joaca (filmul, regizorul, data si ora). Sa se trateze cazul in care actorul nu joaca in niciun film si cazul in care filmul in care joaca nu este programat.

```
CREATE OR REPLACE PROCEDURE proiectiipentruactori(v_nume IN actori.nume%TYPE)
AS
   --pentru a retine info despre proiectii
   TYPE infoproiectii IS RECORD
                     (
                        v_nume_a         actori.nume%TYPE,
                        v_prenume_a      actori.prenume%TYPE,
                        v_titlu          filme.titlu%TYPE,
                        v_nume_reg       regizori.nume%TYPE,
                        v_prenume_reg    regizori.prenume%TYPE,
                        v_ora_incepere   proiectii.ora_incepere%TYPE,
                        v_data_proiectie proiectii.data_proiectie%TYPE,
                        v_id_proiectie   proiectii.id_proiectie%TYPE
                     );
   TYPE proiectii_table IS TABLE OF infoproiectii;
   t_infoproiectii proiectii_table;
   var_nume         actori.nume%TYPE; --var folosita pentru no_data_found si too_many_rows
```

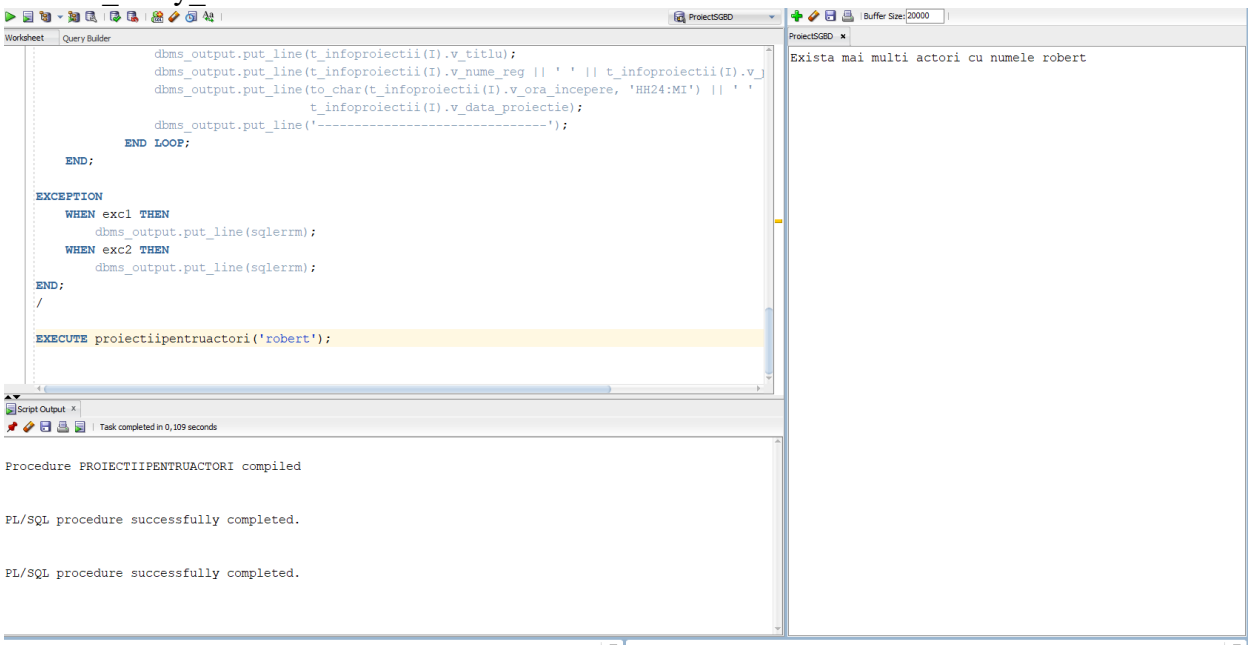
```
   exc1 EXCEPTION;
   exc2 EXCEPTION;

BEGIN
   BEGIN
      SELECT nume
      INTO var_nume
      FROM actori
      WHERE UPPER(nume) = UPPER(v_nume);

   EXCEPTION
      WHEN no_data_found THEN
         dbms_output.put_line('Nu exista actorul cu numele ' || v_nume);
         RETURN;
      WHEN too_many_rows THEN
         dbms_output.put_line('Exista mai multi actori cu numele ' || v_nume);
         RETURN;
   END;


   BEGIN
      SELECT DISTINCT A.nume,
                      A.prenume,
                      F.titlu,
                      R.nume,
                      R.prenume,
                      P.ora_incepere,
                      P.data_proiectie,
                      P.id_proiectie BULK COLLECT
      INTO t_infoproiectii
      FROM actori A
           LEFT JOIN actori_filme af ON af.id_actor = A.id_actor
           LEFT JOIN filme F ON F.id_film = af.id_film
           LEFT JOIN regizori R ON R.id_regizor = F.id_regizor
           LEFT JOIN proiectii P ON P.id_film = F.id_film
      WHERE UPPER(A.nume) = UPPER(v_nume);
      dbms_output.put_line('------------------------------------------------------------');
      dbms_output.put_line(t_infoproiectii(1).v_nume_a || ' ' || t_infoproiectii(1).v_prenume_a);
      dbms_output.put_line('------------------------------------------------------------');
      -- Exc1: Actorul nu joacă în niciun film
      IF t_infoproiectii(1).v_titlu IS NULL THEN
         dbms_output.put_line('!NU JOACA IN NICIUN FILM!');
         raise_application_error(-20001, 'Exc1: Actorul nu joacă în niciun film.');
      END IF;

      FOR I IN t_infoproiectii.FIRST .. t_infoproiectii.LAST
         LOOP
            -- Exc2: Actorul joacă în filme, dar unul dintre filme nu este programat încă
            IF t_infoproiectii(I).v_id_proiectie IS NULL THEN
               dbms_output.put_line('!!JOACA INTR-UN FILM NEPROGRAMAT!!');
               raise_application_error(-20002,
                                       'Exc2: Actorul joacă în filme, dar unul dintre filme nu este programat
încă.');
            END IF;
```

```
            dbms_output.put_line('------------------------------');
            dbms_output.put_line(t_infoproiectii(I).v_titlu);
            dbms_output.put_line(t_infoproiectii(I).v_nume_reg || ' ' ||
t_infoproiectii(I).v_prenume_reg);
            dbms_output.put_line(to_char(t_infoproiectii(I).v_ora_incepere, 'HH24:MI') || ' ' ||
                        t_infoproiectii(I).v_data_proiectie);
            dbms_output.put_line('------------------------------');
          END LOOP;
   END;

EXCEPTION
   WHEN exc1 THEN
      dbms_output.put_line(sqlerrm);
   WHEN exc2 THEN
      dbms_output.put_line(sqlerrm);
END;
/

EXECUTE proiectiipentruactori('&p_nume');
```

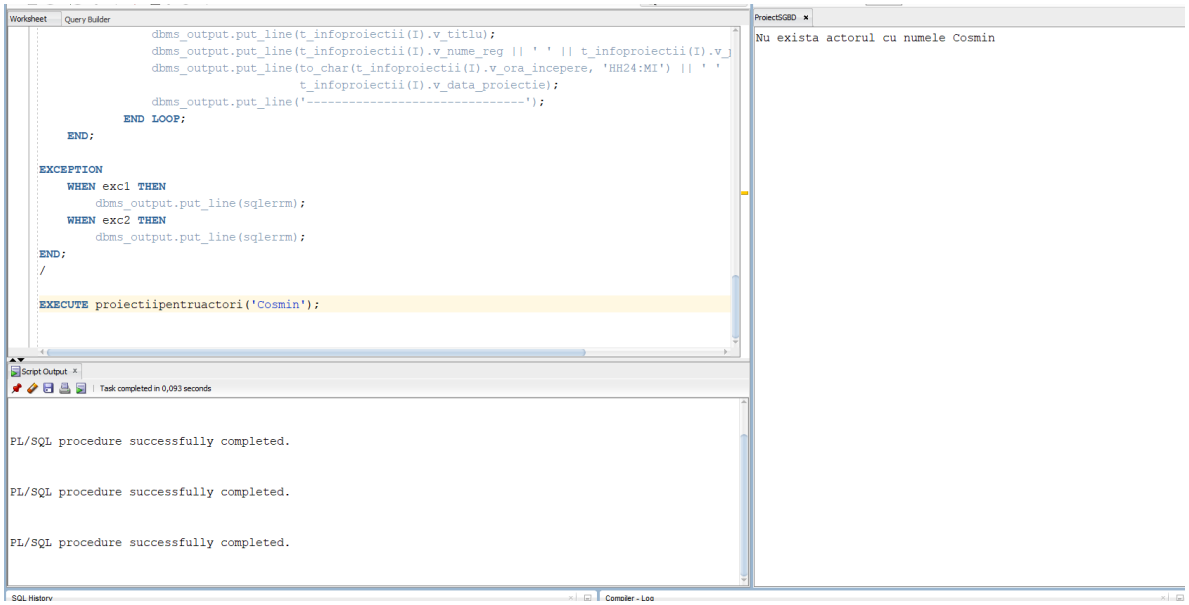
Caz favorabil:



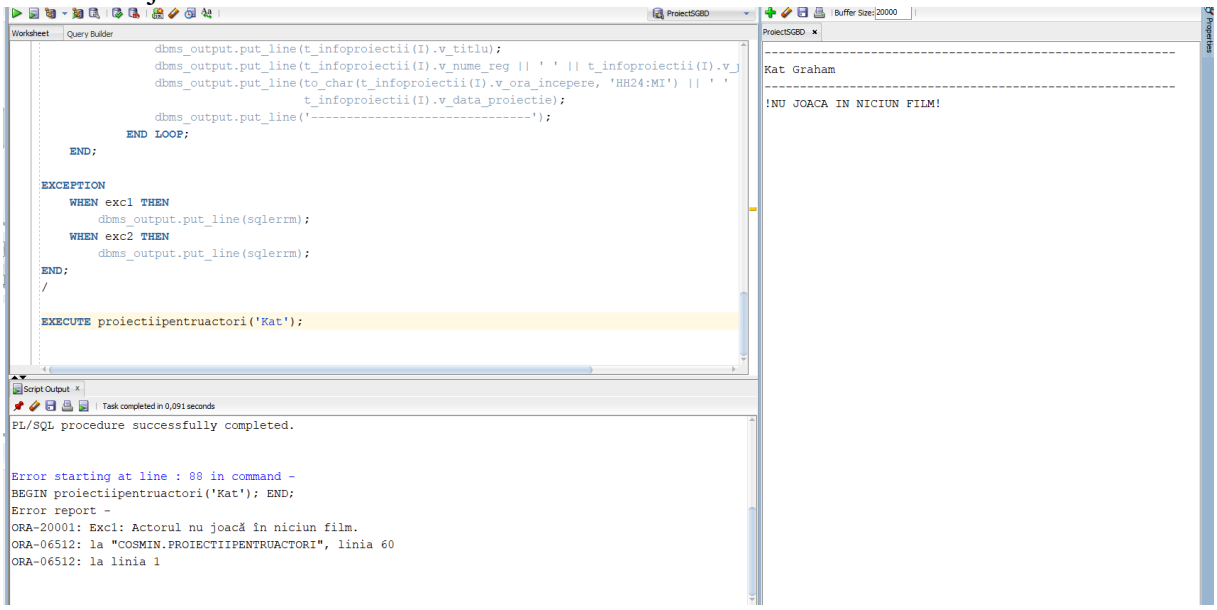
Caz too_many_rows



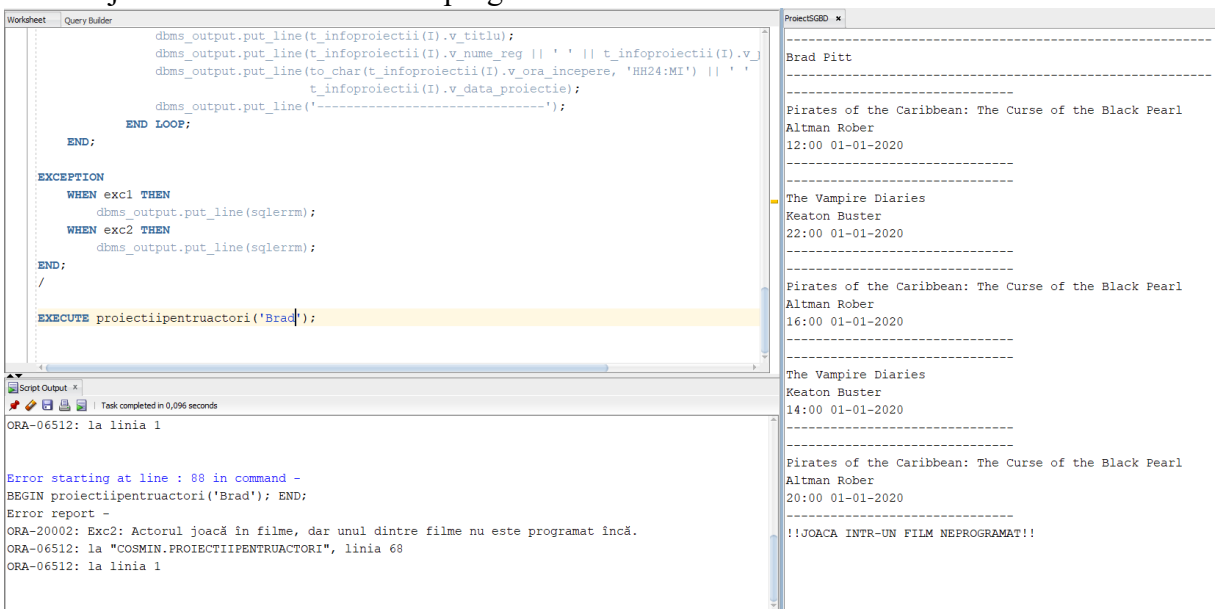
No_data_found



Actorul nu joaca in niciun film



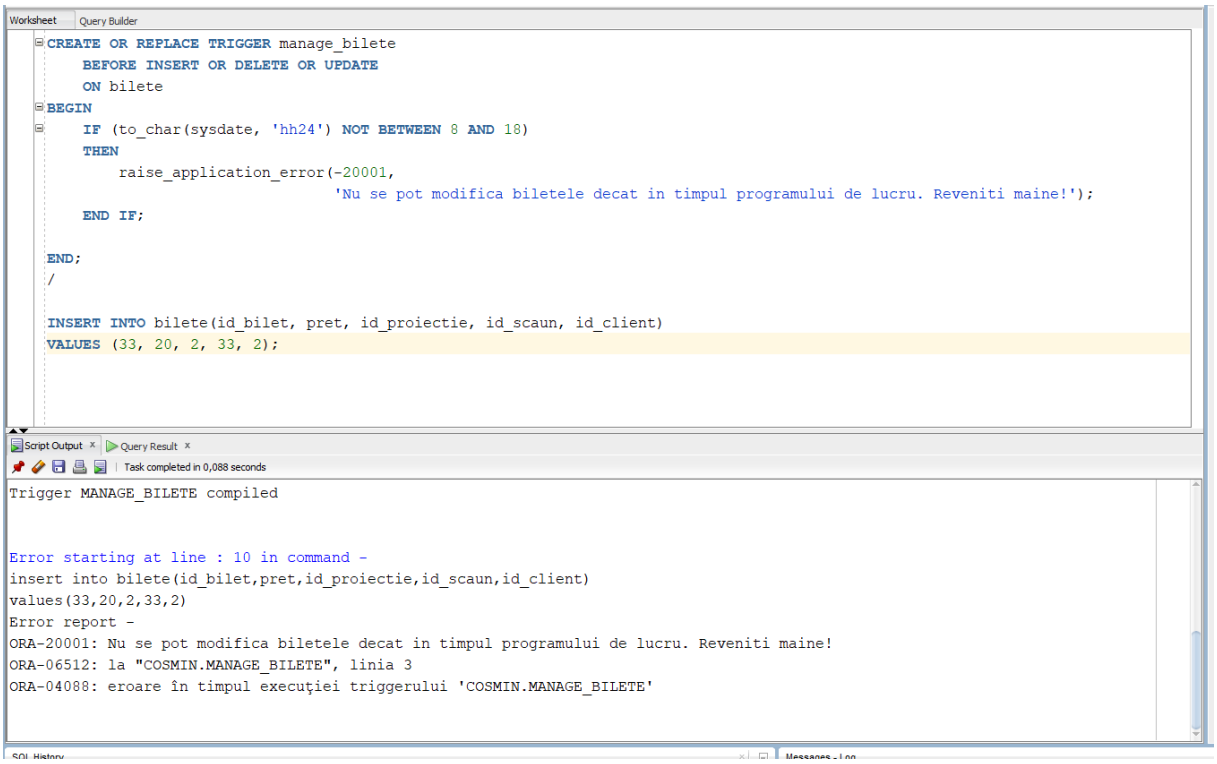
Actorul joaca in filme dar nu sunt programate.



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

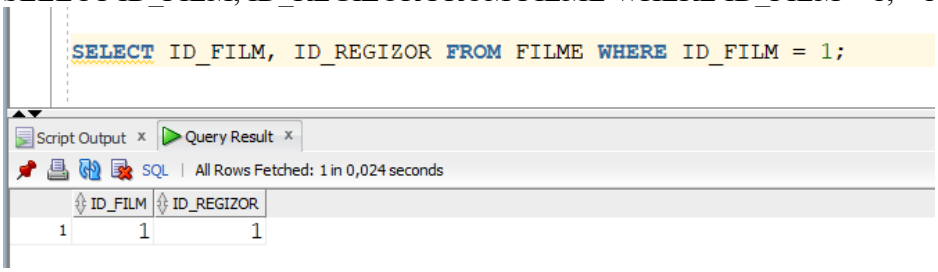
Trigger-ul MANAGE_BILETE este declanșat atunci când se încearcă Insert/Delete/Update pe tabela Bilete în afara programului de lucru al cinematografelor (8-18)

```
create or replace trigger manage_bilete
   before insert or delete or update
   on bilete
begin
   if (to_char(sysdate, 'hh24') not between 8 and 18)
   then
      RAISE_APPLICATION_ERROR(-20001,
                  'Nu se pot modifica biletele decât in timpul programului de lucru. Reveniți
mâine!');
   end if;

end;
/

insert into bilete(id_bilet, pret, id_proiectie, id_scaun, id_client)
values (33, 20, 2, 33, 2);
```



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

```
CREATE OR REPLACE TRIGGER modificare_regizor
   BEFORE UPDATE OF id_regizor
   ON filme
   FOR EACH ROW
BEGIN
   IF :NEW.id_regizor <> :OLD.id_regizor THEN
      raise_application_error(-20000, 'Nu puteti modifica regizorul unui film?');
   END IF;
END;
/
SELECT ID_FILM, ID_REGIZOR FROM FILME WHERE ID_FILM = 1; --1
```




```
UPDATE FILME
SET ID_REGIZOR = 2
WHERE ID_FILM=1;
```

```
CREATE OR REPLACE TRIGGER MODIFICARE_REGIZOR
BEFORE UPDATE OF ID_REGIZOR ON FILME
FOR EACH ROW
BEGIN
IF :NEW.ID_REGIZOR <> :OLD.ID_REGIZOR THEN
RAISE_APPLICATION_ERROR (-20000, 'Nu puteti modifica regizorul unui film?');
END IF;
END;
/

SELECT ID_FILM, ID_REGIZOR FROM FILME WHERE ID_FILM = 1; --1

UPDATE FILME
SET ID_REGIZOR = 2
WHERE ID_FILM=1;
```

Script Output | Query Result

Task completed in 0,164 seconds

```
Trigger MODIFICARE_REGIZOR compiled


Error starting at line : 13 in command -
UPDATE FILME
SET ID_REGIZOR = 2
WHERE ID_FILM=1
Error report -
ORA-20000: Nu puteti modifica regizorul unui film?
ORA-06512: la "COSMIN.MODIFICARE_REGIZOR", linia 3
ORA-04088: eroare în timpul execuţiei triggerului 'COSMIN.MODIFICARE_REGIZOR'
```

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

```
--tabel in care retinem toate evenimentele
CREATE TABLE activitate_baza_de_date
(
   nume_baza_de_date VARCHAR2(50),
   user_logat       VARCHAR2(50),
   eveniment        VARCHAR2(2000),
   data_eveniment    TIMESTAMP
);
--eveniment de login
CREATE OR REPLACE TRIGGER evenimente_after_logon
   AFTER LOGON
   ON SCHEMA
BEGIN
   INSERT INTO activitate_baza_de_date
   VALUES (SYS.database_name, SYS.login_user, 'Utilizator logat', systimestamp);
END;
/
--eveniment de logoff
CREATE OR REPLACE TRIGGER evenimente_before_logoff
   BEFORE LOGOFF
   ON SCHEMA
BEGIN
   INSERT INTO activitate_baza_de_date
   VALUES (SYS.database_name, SYS.login_user, 'utilizator delogat', systimestamp);
```

```
END;
/
--erori
CREATE OR REPLACE TRIGGER evenimente_after_servererror
   AFTER SERVERERROR
   ON SCHEMA
BEGIN
   INSERT INTO activitate_baza_de_date
   VALUES (SYS.database_name, SYS.login_user, dbms_utility.format_error_stack, systimestamp);
   dbms_output.put_line(dbms_utility.format_error_stack || ' ' || SYS.login_user);
END;
/
```

```
CREATE OR REPLACE TRIGGER EVENIMENTE_AFTER_LOGON
AFTER LOGON ON SCHEMA
BEGIN
  INSERT INTO ACTIVITATE_BAZA_DE_DATE
  VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER, 'Utilizator logat', SYSTIMESTAMP);
END;
/

CREATE OR REPLACE TRIGGER EVENIMENTE_BEFORE_LOGOFF
BEFORE LOGOFF ON SCHEMA
BEGIN
  INSERT INTO ACTIVITATE_BAZA_DE_DATE
  VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER, 'utilizator delogat', SYSTIMESTAMP);
END;
/
```

Script Output | Query Result

Task completed in 0,145 seconds

```
ORA-06512: la "COSMIN.MANAGE_BILETE", linia 4
ORA-04088: eroare în timpul execuţiei triggerului 'COSMIN.MANAGE_BILETE'


Trigger EVENIMENTE_AFTER_LOGON compiled


Trigger EVENIMENTE_BEFORE_LOGOFF compiled


Trigger EVENIMENTE_AFTER_SERVERERROR compiled
```

Declansari:

```
SELECT *
FROM activitate  baza  de  date;
```

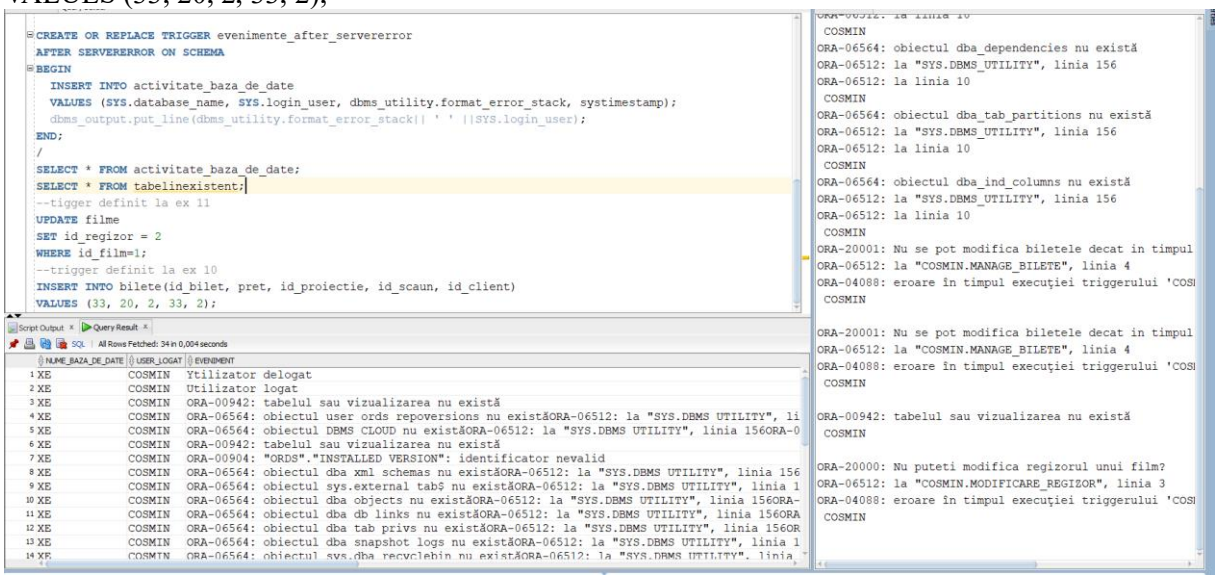
Script Output | Query Result

All Rows Fetched: 34 in 0,004 seconds

	NUME_BAZA_DE_DATE	USER_LOGAT	EVENIMENT
1	XE	COSMIN	Ytilizator delogat
2	XE	COSMIN	Utilizator logat
3	XE	COSMIN	ORA-00942: tabelul sau vizualizarea nu există
4	XE	COSMIN	ORA-06564: obiectul user ords repoversions nu existăORA-06512: la "SYS.DBMS UTILITY", li
5	XE	COSMIN	ORA-06564: obiectul DBMS CLOUD nu existăORA-06512: la "SYS.DBMS UTILITY", linia 156ORA-0
6	XE	COSMIN	ORA-00942: tabelul sau vizualizarea nu există
7	XE	COSMIN	ORA-00904: "ORDS"."INSTALLED VERSION": identificator nevalid
8	XE	COSMIN	ORA-06564: obiectul dba xml schemas nu existăORA-06512: la "SYS.DBMS UTILITY", linia 156
9	XE	COSMIN	ORA-06564: obiectul sys.external tab$ nu existăORA-06512: la "SYS.DBMS UTILITY", linia 1
10	XE	COSMIN	ORA-06564: obiectul dba objects nu existăORA-06512: la "SYS.DBMS UTILITY", linia 156ORA-
11	XE	COSMIN	ORA-06564: obiectul dba db links nu existăORA-06512: la "SYS.DBMS UTILITY", linia 156ORA
12	XE	COSMIN	ORA-06564: obiectul dba tab privs nu existăORA-06512: la "SYS.DBMS UTILITY", linia 156OR
13	XE	COSMIN	ORA-06564: obiectul dba snapshot logs nu existăORA-06512: la "SYS.DBMS UTILITY", linia 1
14	XE	COSMIN	ORA-06564: obiectul sys.dba recyclebin nu existăORA-06512: la "SYS.DBMS UTILITY", linia

```
SELECT *
FROM tabelinexistent;
--tigger definit la ex 11
UPDATE filme
SET id_regizor = 2
WHERE id_film = 1;
--trigger definit la ex 10
INSERT INTO bilete(id_bilet, pret, id_proiectie, id_scaun, id_client)
```

VALUES (33, 20, 2, 33, 2);



Ex 13: Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

```
CREATE OR REPLACE PACKAGE pachet_cinematograf AS
   -- Tipuri definite
   -- Cursor pentru obtinerea informatiilor despre proiectie
   CURSOR cursor_proiectie(p_id_proiectie proiectii.id_proiectie%TYPE) IS
      SELECT *
      FROM proiectii
      WHERE id_proiectie < p_id_proiectie;

   -- Cursor parametrizat SELECT ... FOR UPDATE pentru obtinerea si blocarea detaliilor biletelor pentru proiectie
   CURSOR cursor_bilete(p_id_proiectie bilete.id_proiectie%TYPE) IS
      SELECT b.*
      FROM bilete b
      WHERE b.id_proiectie = p_id_proiectie
         FOR UPDATE OF b.pret; -- Actualizăm doar câmpul PRET


   TYPE infoactori IS RECORD
                (
                   actor_id actori.id_actor%TYPE,
                   nume     actori.nume%TYPE,
                   prenume  actori.prenume%TYPE
                );

   TYPE infoproiectie IS RECORD
                 (
                    id_proiectie   proiectii.id_proiectie%TYPE,
                    ora            proiectii.ora_incepere%TYPE,
                    data_proiectie proiectii.data_proiectie%TYPE,
                    sala           proiectii.id_sala%TYPE
                 );
   TYPE infoproiectii2 IS RECORD
```

```
                (
                    v_nume_a         actori.nume%TYPE,
                    v_prenume_a      actori.prenume%TYPE,
                    v_titlu          filme.titlu%TYPE,
                    v_nume_reg       regizori.nume%TYPE,
                    v_prenume_reg    regizori.prenume%TYPE,
                    v_ora_incepere   proiectii.ora_incepere%TYPE,
                    v_data_proiectie proiectii.data_proiectie%TYPE,
                    v_id_proiectie   proiectii.id_proiectie%TYPE
                );
    TYPE actori_index_table IS TABLE OF infoactori INDEX BY PLS_INTEGER;
    TYPE actori_imbricat IS TABLE OF infoactori;
    TYPE proiectii_vector IS VARRAY(100) OF infoproiectie;
    TYPE proiectii_table IS TABLE OF infoproiectii2;

    -- Proceduri
    PROCEDURE procesareactori(v_id filme.id_film%TYPE);
    PROCEDURE detalii_actualizare_bilete_eveniment(p_id_proiectie proiectii.id_proiectie%TYPE,
                                        p_nou_pret bilete.pret%TYPE);
    PROCEDURE proiectiipentruactori(v_nume IN actori.nume%TYPE);

    -- Functii
    FUNCTION numarsaliangajat(numele angajati.nume%TYPE) RETURN VARCHAR2;

END pachet_cinematograf;
/

CREATE OR REPLACE PACKAGE BODY pachet_cinematograf AS

    PROCEDURE procesareactori(v_id filme.id_film%TYPE) AS
        t_actori_index     actori_index_table;
        t_actori_imbricat  actori_imbricat  := actori_imbricat();
        t_proiectii_vector proiectii_vector := proiectii_vector();
        nume_film          VARCHAR2(100); --numele filmului cu id_ul v_id
    BEGIN
        BEGIN
            -- Verificare daca exista filmul cu ID-ul dat

            SELECT titlu
            INTO nume_film
            FROM filme
            WHERE id_film = v_id;
        EXCEPTION
            WHEN no_data_found THEN
                dbms_output.put_line('filmul cu id-ul ' || v_id || ' nu există.');
                RETURN;
        END;


        ---memorare informatii despre actorii care joaca in filmul dat in tabelul indexat
        BEGIN
            SELECT A.id_actor,
                   A.nume,
                   A.prenume BULK COLLECT
            INTO t_actori_index
```

```
FROM actori A
        JOIN actori_filme af ON af.id_actor = A.id_actor
        JOIN filme F ON F.id_film = af.id_film
WHERE F.id_film = v_id;

---memorare in tabelul imbricat a actorilor care nu joaca in film
FOR I IN (SELECT A.id_actor, A.nume, A.prenume
         FROM actori A
         WHERE A.id_actor NOT IN (SELECT a1.id_actor
                                  FROM actori a1
                                           JOIN actori_filme af ON af.id_actor = a1.id_actor
                                  WHERE af.id_film = v_id))
    LOOP
        t_actori_imbricat.extend;
        t_actori_imbricat(t_actori_imbricat.LAST).actor_id := I.id_actor;
        t_actori_imbricat(t_actori_imbricat.LAST).nume := I.nume;
        t_actori_imbricat(t_actori_imbricat.LAST).prenume := I.prenume;
    END LOOP;

FOR I IN (SELECT id_proiectie, ora_incepere, data_proiectie, id_sala
          FROM proiectii
          WHERE id_film = v_id)
    LOOP
        t_proiectii_vector.extend;
        t_proiectii_vector(t_proiectii_vector.LAST).id_proiectie := I.id_proiectie;
        t_proiectii_vector(t_proiectii_vector.LAST).ora := I.ora_incepere;
        t_proiectii_vector(t_proiectii_vector.LAST).data_proiectie := I.data_proiectie;
        t_proiectii_vector(t_proiectii_vector.LAST).sala := I.id_sala;
    END LOOP;
dbms_output.put_line('-------------------------------------------------');
dbms_output.put_line(nume_film || ':');
dbms_output.put_line('-------------------------------------------------');

IF t_actori_index.COUNT = 0 THEN
    dbms_output.put_line('------------------------------');
    dbms_output.put_line(nume_film || ':');
    dbms_output.put_line('Nu exista actori pentru filmul cu ID-ul ' || v_id);
    dbms_output.put_line('------------------------------');
ELSE
    dbms_output.put_line('Cast:');
    FOR I IN t_actori_index.FIRST..t_actori_index.LAST
        LOOP
            dbms_output.put_line(' ' || t_actori_index(I).nume || ' ' || t_actori_index(I).prenume);
        END LOOP;
END IF;

IF t_actori_imbricat.COUNT = 0 THEN
    dbms_output.put_line('------------------------------');
    dbms_output.put_line('TOTI ACTORII SUNT IN CAST');
    dbms_output.put_line('------------------------------');
ELSE
    dbms_output.put_line('------------------------------');
    dbms_output.put_line('Nu fac parte din cast:');
    FOR I IN t_actori_imbricat.FIRST..t_actori_imbricat.LAST
        LOOP
```

```
                dbms_output.put_line(' ' || t_actori_imbricat(I).nume || ' ' ||
t_actori_imbricat(I).prenume);
            END LOOP;
        END IF;


        IF t_proiectii_vector.COUNT = 0 THEN
           dbms_output.put_line('FILMUL NU ESTE ÎNCĂ PROGRAMAT');
        ELSE
           dbms_output.put_line('-------------------------------------------------------');
           dbms_output.put_line('Proiectiile filmului ' || nume_film || ': ');
           dbms_output.put_line('-------------------------------------------------------');
           FOR I IN t_proiectii_vector.FIRST..t_proiectii_vector.LAST
              LOOP
                 dbms_output.put_line('NR. PROIECTIE: ' || t_proiectii_vector(I).id_proiectie || ' ');

                 dbms_output.put_line('SALA: ' || t_proiectii_vector(I).sala);
                 dbms_output.put_line('DATA: ' || t_proiectii_vector(I).data_proiectie);
                 dbms_output.put_line('ORA: ' || to_char(t_proiectii_vector(I).ora, 'HH24:MI'));
                 dbms_output.put_line('----------------------------');


              END LOOP;
           END IF;
      END;

   END procesareactori;

   PROCEDURE detalii_actualizare_bilete_eveniment(
   p_id_proiectie IN proiectii.id_proiectie%TYPE,
   p_nou_pret IN bilete.pret%TYPE
) AS
   -- Cursor pentru obtinerea informatiilor despre proiectie
   CURSOR cursor_proiectie IS
      SELECT *
      FROM proiectii
      WHERE id_proiectie < p_id_proiectie;

   proiectie_rec proiectii%ROWTYPE;

   -- Cursor parametrizat pentru obtinerea si blocarea detaliilor biletelor pentru proiectie
   CURSOR cursor_bilete(p_id_proiectie_param proiectii.id_proiectie%TYPE) IS
      SELECT *
      FROM bilete
      WHERE id_proiectie = p_id_proiectie_param
      FOR UPDATE OF pret; -- Actualizăm doar câmpul PRET

   bilet_rec bilete%ROWTYPE;

BEGIN
   -- Deschide cursorul pentru informatii despre proiectie
   OPEN cursor_proiectie;
   loop
   FETCH cursor_proiectie INTO proiectie_rec;
   EXIT WHEN cursor_proiectie%NOTFOUND;
```

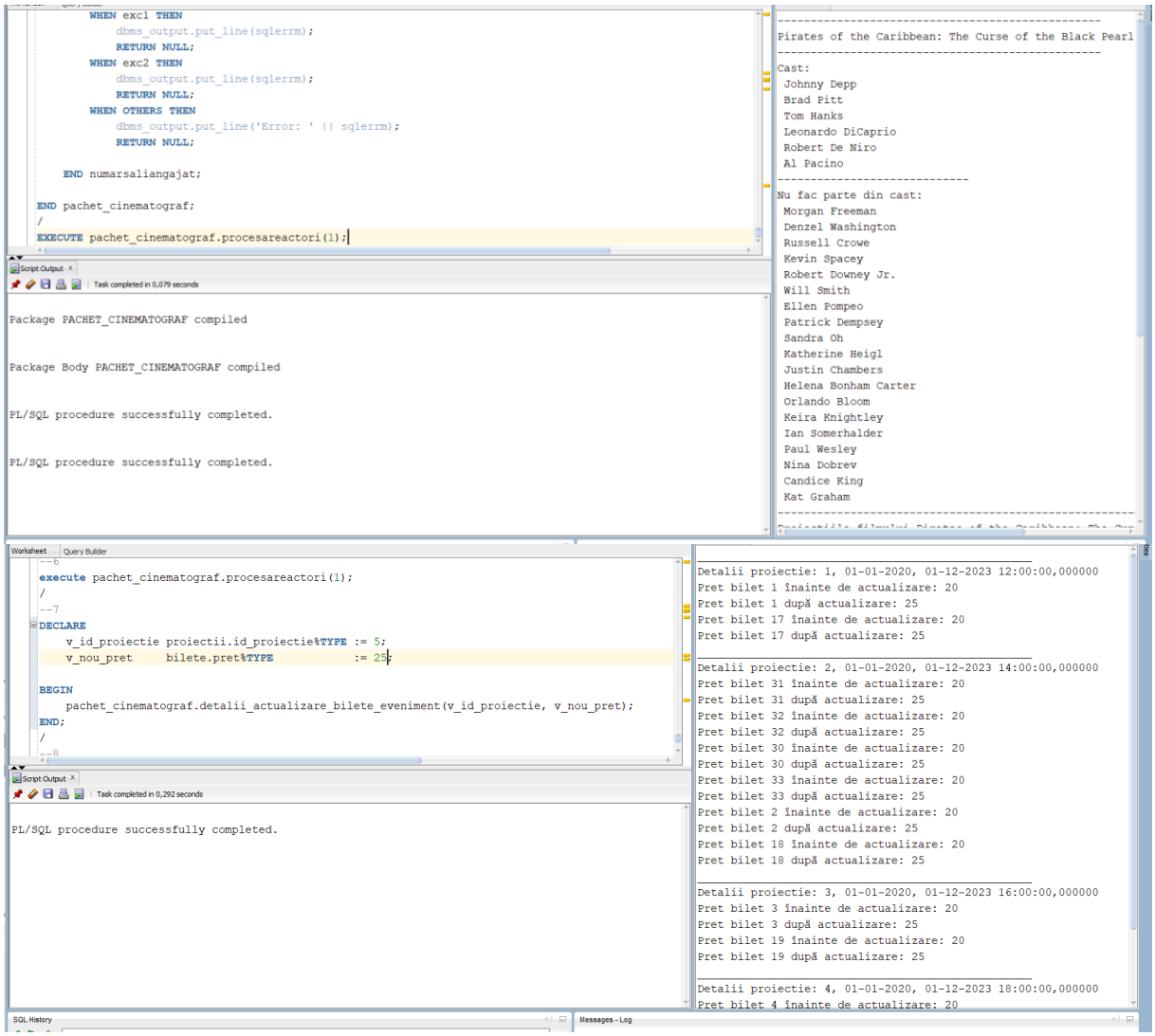
```
DBMS_OUTPUT.PUT_LINE('_____________________________________________________');
   DBMS_OUTPUT.PUT_LINE('Detalii proiectie: ' || proiectie_rec.id_proiectie || ', ' ||
proiectie_rec.data_proiectie || ', ' || proiectie_rec.ora_incepere);

   -- Deschide cursorul parametrizat pentru detaliile biletelor si actualizează PRET-ul
OPEN cursor_bilete(proiectie_rec.id_proiectie);

-- Parcurge cursorul pentru fiecare înregistrare
LOOP
   FETCH cursor_bilete INTO bilet_rec;

   -- Iesire din bucla când nu mai există înregistrări
   EXIT WHEN cursor_bilete%NOTFOUND;

   DBMS_OUTPUT.PUT_LINE('Pret bilet ' || bilet_rec.id_bilet || ' înainte de actualizare: ' ||
bilet_rec.pret);

   -- Actualizare PRET
   UPDATE bilete
   SET pret = p_nou_pret
   WHERE CURRENT OF cursor_bilete;

   DBMS_OUTPUT.PUT_LINE('Pret bilet ' || bilet_rec.id_bilet || ' după actualizare: ' || p_nou_pret);
END LOOP;

-- Închide cursorul parametrizat pentru detaliile biletelor
CLOSE cursor_bilete;

end loop;
close cursor_proiectie;

END detalii_actualizare_bilete_eveniment;
   PROCEDURE proiectiipentruactori(v_nume IN actori.nume%TYPE) IS
      t_infoproiectii proiectii_table;
      var_nume       actori.nume%TYPE; --var folosita pentru no_data_found si too_many_rows

      exc1 EXCEPTION;
      exc2 EXCEPTION;
   BEGIN
      BEGIN
      SELECT nume
      INTO var_nume
      FROM actori
      WHERE UPPER(nume) = UPPER(v_nume);

   EXCEPTION
      WHEN no_data_found THEN
         dbms_output.put_line('Nu exista actorul cu numele ' || v_nume);
         RETURN;
      WHEN too_many_rows THEN
         dbms_output.put_line('Exista mai multi actori cu numele ' || v_nume);
         RETURN;
   END;

   BEGIN
```

```
        SELECT DISTINCT A.nume,
                        A.prenume,
                        F.titlu,
                        R.nume,
                        R.prenume,
                        P.ora_incepere,
                        P.data_proiectie,
                        P.id_proiectie BULK COLLECT
        INTO t_infoproiectii
        FROM actori A
             LEFT JOIN actori_filme af ON af.id_actor = A.id_actor
             LEFT JOIN filme F ON F.id_film = af.id_film
             LEFT JOIN regizori R ON R.id_regizor = F.id_regizor
             LEFT JOIN proiectii P ON P.id_film = F.id_film
        WHERE UPPER(A.nume) = UPPER(v_nume);
        dbms_output.put_line('-----------------------------------------------------------');
        dbms_output.put_line(t_infoproiectii(1).v_nume_a || ' ' || t_infoproiectii(1).v_prenume_a);
        dbms_output.put_line('-----------------------------------------------------------');
        -- Exc1: Actorul nu joacă în niciun film
        IF t_infoproiectii(1).v_titlu IS NULL THEN
           dbms_output.put_line('!NU JOACA IN NICIUN FILM!');
           raise_application_error(-20001, 'Exc1: Actorul nu joacă în niciun film.');
        END IF;

        FOR I IN t_infoproiectii.FIRST .. t_infoproiectii.LAST
           LOOP
              -- Exc2: Actorul joacă în filme, dar unul dintre filme nu este programat încă
              IF t_infoproiectii(I).v_id_proiectie IS NULL THEN
                 dbms_output.put_line('!!JOACA INTR-UN FILM NEPROGRAMAT!!');
                 raise_application_error(-20002,
                                         'Exc2: Actorul joacă în filme, dar unul dintre filme nu este programat
încă.');
              END IF;
              dbms_output.put_line('-------------------------------');
              dbms_output.put_line(t_infoproiectii(I).v_titlu);
              dbms_output.put_line(t_infoproiectii(I).v_nume_reg || ' ' ||
t_infoproiectii(I).v_prenume_reg);
              dbms_output.put_line(to_char(t_infoproiectii(I).v_ora_incepere, 'HH24:MI') || ' ' ||
                                   t_infoproiectii(I).v_data_proiectie);
              dbms_output.put_line('-------------------------------');
           END LOOP;
     END;

EXCEPTION
     WHEN exc1 THEN
        dbms_output.put_line(sqlerrm);
     WHEN exc2 THEN
        dbms_output.put_line(sqlerrm);


     END proiectiipentruactori;

     FUNCTION numarsaliangajat(numele angajati.nume%TYPE) RETURN VARCHAR2 IS
        rez   VARCHAR2(150) := '';
        orass cinematografe.oras%TYPE;
```

```
        nr    NUMBER;
        exc1 EXCEPTION; --exceptie pt cazul cu 0 sali
        exc2 EXCEPTION; --exceptie pt cazul cu <=4 sali

    BEGIN
        SELECT nvl(COUNT(S.id_sala), 0) AS num_sala, C.oras
        INTO nr, orass
        FROM angajati A
                JOIN cinematografe C ON A.id_cinematograf = C.id_cinematograf
                LEFT JOIN sali S ON C.id_cinematograf = S.id_cinematograf --pt caul in care nu exista
sali
        --in cinematograful in care lucreaza
        WHERE UPPER(A.nume) = UPPER(numele)
        GROUP BY C.oras;
        IF nr = 0 THEN
            raise_application_error(-20003, 'EXC1: cinematograful in care lucreaza angajatul ' || numele ||
                                    ' nu are sali. Se recomanda inchiderea');
        ELSIF nr < 5 THEN
            raise_application_error(-20004, 'EXC2: cinematograful in care lucreaza angajatul ' || numele ||
                                    ' are un numar mic de sali. Se recomanda redistribuirea angajatului la alt
cinematograf');
        END IF;

        rez := numele || ' lucreaza in ' || orass || ' unde sunt ' || to_char(nr) || ' sali' || rez;

        RETURN rez;
    EXCEPTION
        WHEN too_many_rows THEN
            dbms_output.put_line('Exista mai multi angajati cu numele ' || numele);
            RETURN NULL;
        WHEN no_data_found THEN
            dbms_output.put_line('Nu exista angajati cu numele ' || numele);
            RETURN NULL;
        WHEN exc1 THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN exc2 THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN OTHERS THEN
            dbms_output.put_line('Error: ' || sqlerrm);
            RETURN NULL;

    END numarsaliangajat;

END pachet_cinematograf;
/


--6
execute pachet_cinematograf.procesareactori(1);
/
--7
DECLARE
    v_id_proiectie proiectii.id_proiectie%TYPE := 5;
```

```
    v_nou_pret     bilete.pret%TYPE          := 25;

BEGIN
    pachet_cinematograf.detalii_actualizare_bilete_eveniment(v_id_proiectie, v_nou_pret);
END;
/
--8
DECLARE
    v_nume angajati.nume%TYPE := UPPER('&p_nume');
BEGIN
    dbms_output.put_line(pachet_cinematograf.numarsaliangajat(v_nume));
END;
/
--9
EXECUTE proiectiipentruactori('&p_nume');
--10
```

Am apelat procedurile si functiile definite, pentru cazurile favorabile:



```
        WHEN excl THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN exc2 THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN OTHERS THEN
            dbms_output.put_line('Error: ' || sqlerrm);
            RETURN NULL;

    END numarsaliangajat;

END pachet_cinematograf;
/
EXECUTE pachet_cinematograf.procesareactori(1);
```

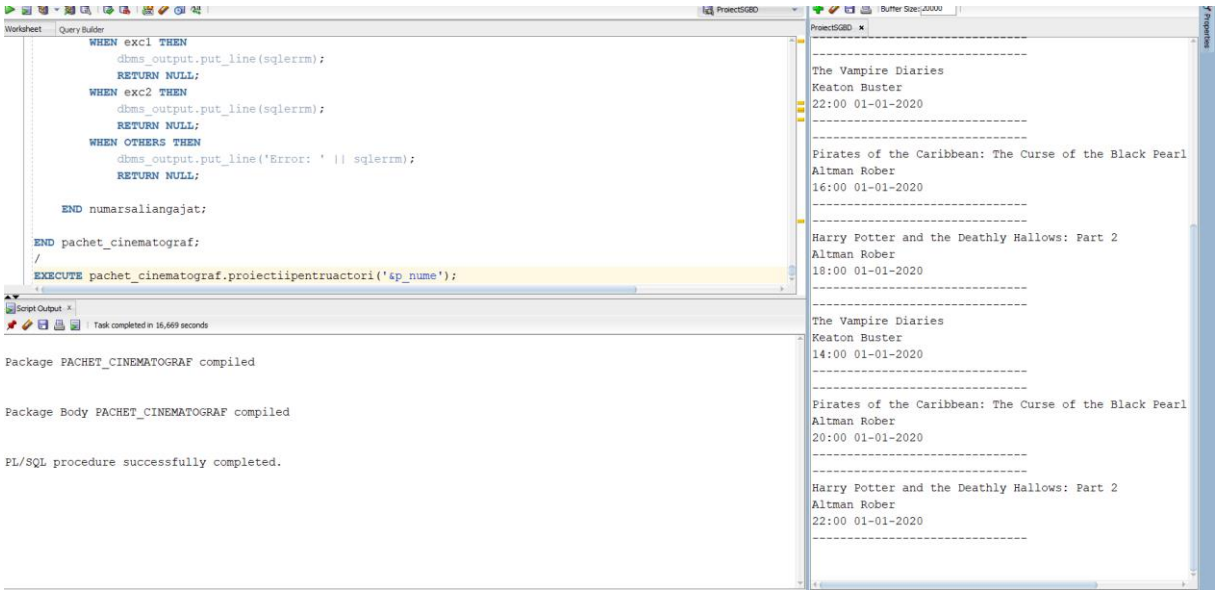
```
Package PACHET_CINEMATOGRAF compiled

Package Body PACHET_CINEMATOGRAF compiled

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

```
--------------------------------------------------
Pirates of the Caribbean: The Curse of the Black Pearl
--------------------------------------------------
Cast:
 Johnny Depp
 Brad Pitt
 Tom Hanks
 Leonardo DiCaprio
 Robert De Niro
 Al Pacino
------------------------------
Nu fac parte din cast:
 Morgan Freeman
 Denzel Washington
 Russell Crowe
 Kevin Spacey
 Robert Downey Jr.
 Will Smith
 Ellen Pompeo
 Patrick Dempsey
 Sandra Oh
 Katherine Heigl
 Justin Chambers
 Helena Bonham Carter
 Orlando Bloom
 Keira Knightley
 Ian Somerhalder
 Paul Wesley
 Nina Dobrev
 Candice King
 Kat Graham
```

```
--6
execute pachet_cinematograf.procesareactori(1);
/
--7
DECLARE
    v_id_proiectie proiectii.id_proiectie%TYPE := 5;
    v_nou_pret     bilete.pret%TYPE           := 25;

BEGIN
    pachet_cinematograf.detalii_actualizare_bilete_eveniment(v_id_proiectie, v_nou_pret);
END;
/
--8
```

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

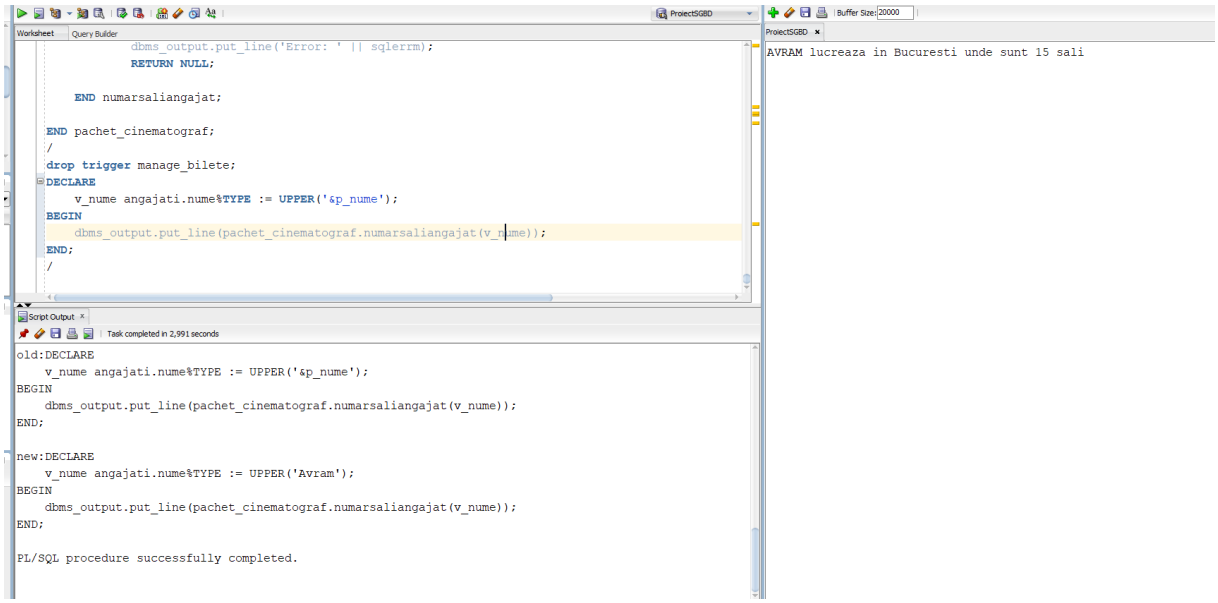
```
_____
Detalii proiectie: 1, 01-01-2020, 01-12-2023 12:00:00,000000
Pret bilet 1 înainte de actualizare: 20
Pret bilet 1 după actualizare: 25
Pret bilet 17 înainte de actualizare: 20
Pret bilet 17 după actualizare: 25
_____
Detalii proiectie: 2, 01-01-2020, 01-12-2023 14:00:00,000000
Pret bilet 31 înainte de actualizare: 20
Pret bilet 31 după actualizare: 25
Pret bilet 32 înainte de actualizare: 20
Pret bilet 32 după actualizare: 25
Pret bilet 30 înainte de actualizare: 20
Pret bilet 30 după actualizare: 25
Pret bilet 33 înainte de actualizare: 20
Pret bilet 33 după actualizare: 25
Pret bilet 2 înainte de actualizare: 20
Pret bilet 2 după actualizare: 25
Pret bilet 18 înainte de actualizare: 20
Pret bilet 18 după actualizare: 25
_____
Detalii proiectie: 3, 01-01-2020, 01-12-2023 16:00:00,000000
Pret bilet 3 înainte de actualizare: 20
Pret bilet 3 după actualizare: 25
Pret bilet 19 înainte de actualizare: 20
Pret bilet 19 după actualizare: 25
_____
Detalii proiectie: 4, 01-01-2020, 01-12-2023 18:00:00,000000
Pret bilet 4 înainte de actualizare: 20
```



```
        WHEN excl THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN exc2 THEN
            dbms_output.put_line(sqlerrm);
            RETURN NULL;
        WHEN OTHERS THEN
            dbms_output.put_line('Error: ' || sqlerrm);
            RETURN NULL;

    END numarsaliangajat;

END pachet_cinematograf;
/
EXECUTE pachet_cinematograf.proiectiipentruactori('&p_nume');
```

```
Package PACHET_CINEMATOGRAF compiled

Package Body PACHET_CINEMATOGRAF compiled

PL/SQL procedure successfully completed.
```

```
The Vampire Diaries
Keaton Buster
22:00 01-01-2020
--------------------------------
--------------------------------
Pirates of the Caribbean: The Curse of the Black Pearl
Altman Rober
16:00 01-01-2020
--------------------------------
--------------------------------
Harry Potter and the Deathly Hallows: Part 2
Altman Rober
18:00 01-01-2020
--------------------------------
--------------------------------
The Vampire Diaries
Keaton Buster
14:00 01-01-2020
--------------------------------
--------------------------------
Pirates of the Caribbean: The Curse of the Black Pearl
Altman Rober
20:00 01-01-2020
--------------------------------
--------------------------------
Harry Potter and the Deathly Hallows: Part 2
Altman Rober
22:00 01-01-2020
--------------------------------
```

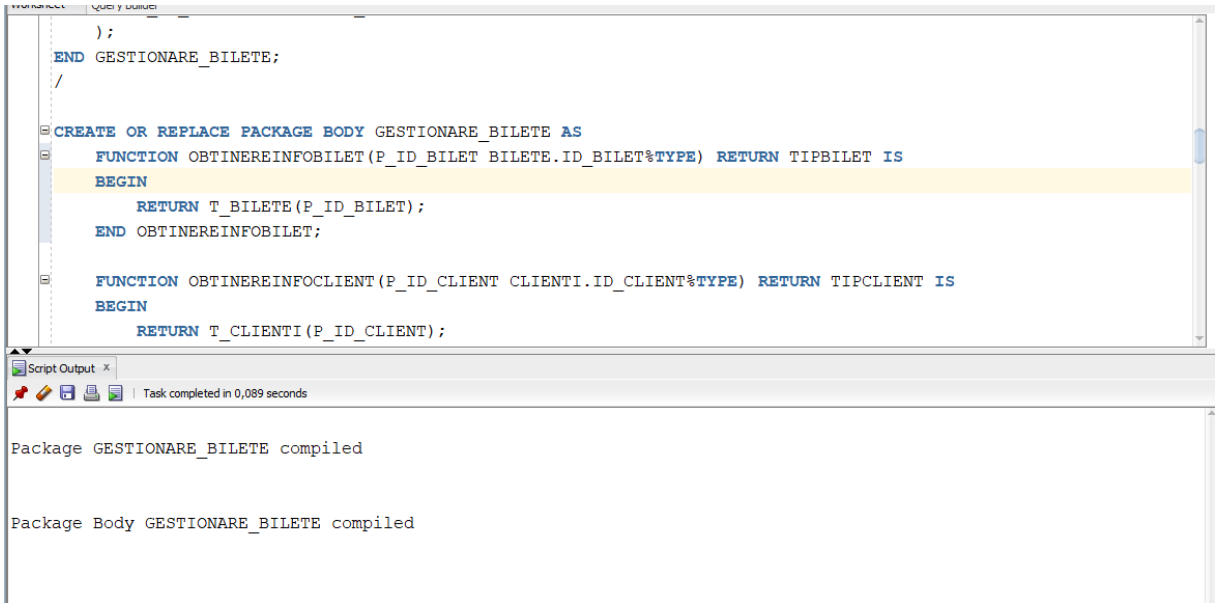


Ex 14: Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

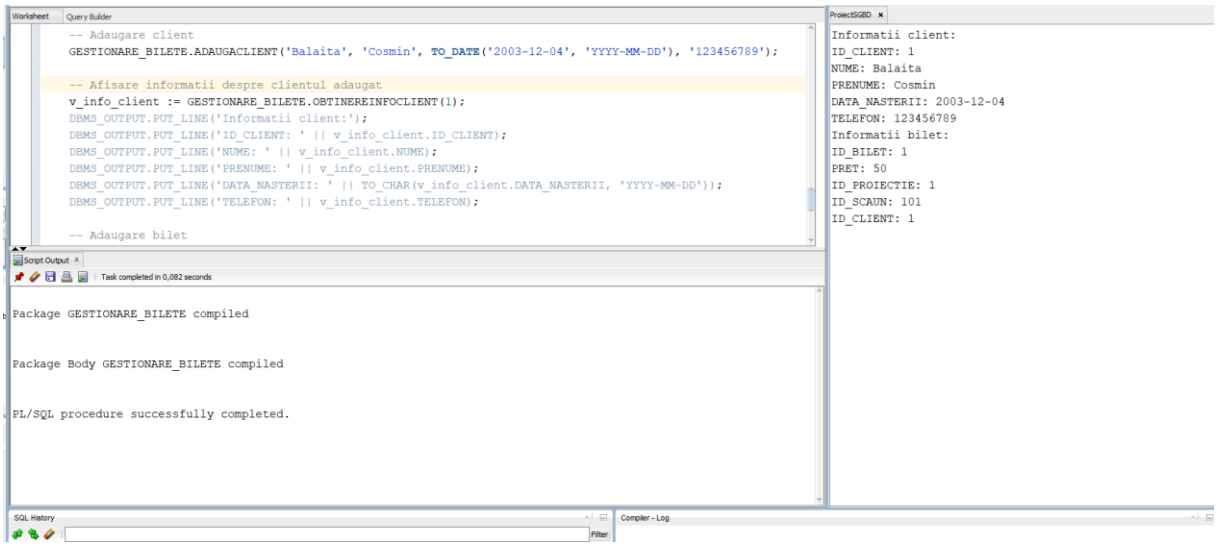
```
CREATE OR REPLACE PACKAGE GESTIONARE_BILETE IS

    TYPE TIPBILET IS RECORD (
        ID_BILET BILETE.ID_BILET%TYPE,
        PRET BILETE.PRET%TYPE,
        ID_PROIECTIE BILETE.ID_PROIECTIE%TYPE,
        ID_SCAUN BILETE.ID_SCAUN%TYPE,
        ID_CLIENT BILETE.ID_CLIENT%TYPE
    );

    TYPE TIPCLIENT IS RECORD (
        ID_CLIENT CLIENTI.ID_CLIENT%TYPE,
        NUME CLIENTI.NUME%TYPE,
        PRENUME CLIENTI.PRENUME%TYPE,
        DATA_NASTERII CLIENTI.DATA_NASTERII%TYPE,
        TELEFON CLIENTI.TELEFON%TYPE
    );

    TYPE TabelBilete IS TABLE OF TIPBILET INDEX BY PLS_INTEGER;
    T_bilete TabelBilete;

    TYPE TabelClienti IS TABLE OF TIPCLIENT INDEX BY PLS_INTEGER;
    T_clienti TabelClienti;

    FUNCTION OBTINEREINFOBILET(P_ID_BILET BILETE.ID_BILET%TYPE) RETURN
TIPBILET;
```

```
   FUNCTION OBTINEREINFOCLIENT(P_ID_CLIENT CLIENTI.ID_CLIENT%TYPE) RETURN
TIPCLIENT;
   PROCEDURE ADAUGACLIENT(
      P_NUME CLIENTI.NUME%TYPE,
      P_PRENUME CLIENTI.PRENUME%TYPE,
      P_DATA_NASTERII CLIENTI.DATA_NASTERII%TYPE,
      P_TELEFON CLIENTI.TELEFON%TYPE
   );
   PROCEDURE ADAUGABILET(
      P_PRET BILETE.PRET%TYPE,
      P_ID_PROIECTIE BILETE.ID_PROIECTIE%TYPE,
      P_ID_SCAUN BILETE.ID_SCAUN%TYPE,
      P_ID_CLIENT BILETE.ID_CLIENT%TYPE
   );
END GESTIONARE_BILETE;
/

CREATE OR REPLACE PACKAGE BODY GESTIONARE_BILETE AS
   FUNCTION OBTINEREINFOBILET(P_ID_BILET BILETE.ID_BILET%TYPE) RETURN
TIPBILET IS
   BEGIN
      RETURN T_BILETE(P_ID_BILET);
   END OBTINEREINFOBILET;

   FUNCTION OBTINEREINFOCLIENT(P_ID_CLIENT CLIENTI.ID_CLIENT%TYPE) RETURN
TIPCLIENT IS
   BEGIN
      RETURN T_CLIENTI(P_ID_CLIENT);
   END OBTINEREINFOCLIENT;

   PROCEDURE ADAUGACLIENT(
      P_NUME CLIENTI.NUME%TYPE,
      P_PRENUME CLIENTI.PRENUME%TYPE,
      P_DATA_NASTERII CLIENTI.DATA_NASTERII%TYPE,
      P_TELEFON CLIENTI.TELEFON%TYPE
   ) IS
   BEGIN
      T_clienti(T_clienti.COUNT + 1).id_client := T_clienti.COUNT + 1;
      T_clienti(T_clienti.COUNT).nume := P_NUME;
      T_clienti(T_clienti.COUNT).prenume := P_PRENUME;
      T_clienti(T_clienti.COUNT).data_nasterii := P_DATA_NASTERII;
      T_clienti(T_clienti.COUNT).telefon := P_TELEFON;
   END ADAUGACLIENT;

   PROCEDURE ADAUGABILET(
      P_PRET BILETE.PRET%TYPE,
      P_ID_PROIECTIE BILETE.ID_PROIECTIE%TYPE,
      P_ID_SCAUN BILETE.ID_SCAUN%TYPE,
      P_ID_CLIENT BILETE.ID_CLIENT%TYPE
   ) IS
   BEGIN
      T_bilete(T_bilete.COUNT + 1).id_bilet := T_bilete.COUNT + 1;
      T_bilete(T_bilete.COUNT).pret := P_PRET;
      T_bilete(T_bilete.COUNT).id_proiectie := P_ID_PROIECTIE;
      T_bilete(T_bilete.COUNT).id_scaun := P_ID_SCAUN;
```

```
        T_bilete(T_bilete.COUNT).id_client := P_ID_CLIENT;
    END ADAUGABILET;
END GESTIONARE_BILETE;
/
```




```
DECLARE
    v_info_bilet GESTIONARE_BILETE.TIPBILET;
    v_info_client GESTIONARE_BILETE.TIPCLIENT;
BEGIN
    -- Adaugare client
    GESTIONARE_BILETE.ADAUGACLIENT('Balaita', 'Cosmin', TO_DATE('2003-12-04', 'YYYY-
MM-DD'), '123456789');

    -- Afisare informatii despre clientul adaugat
    v_info_client := GESTIONARE_BILETE.OBTINEREINFOCLIENT(1);
    DBMS_OUTPUT.PUT_LINE('Informatii client:');
    DBMS_OUTPUT.PUT_LINE('ID_CLIENT: ' || v_info_client.ID_CLIENT);
    DBMS_OUTPUT.PUT_LINE('NUME: ' || v_info_client.NUME);
    DBMS_OUTPUT.PUT_LINE('PRENUME: ' || v_info_client.PRENUME);
    DBMS_OUTPUT.PUT_LINE('DATA_NASTERII: ' ||
TO_CHAR(v_info_client.DATA_NASTERII, 'YYYY-MM-DD'));
    DBMS_OUTPUT.PUT_LINE('TELEFON: ' || v_info_client.TELEFON);

    -- Adaugare bilet
    GESTIONARE_BILETE.ADAUGABILET(50.0, 1, 101, 1);

    -- Afisare informatii despre biletul adaugat
    v_info_bilet := GESTIONARE_BILETE.OBTINEREINFOBILET(1);
    DBMS_OUTPUT.PUT_LINE('Informatii bilet:');
    DBMS_OUTPUT.PUT_LINE('ID_BILET: ' || v_info_bilet.ID_BILET);
    DBMS_OUTPUT.PUT_LINE('PRET: ' || v_info_bilet.PRET);
    DBMS_OUTPUT.PUT_LINE('ID_PROIECTIE: ' || v_info_bilet.ID_PROIECTIE);
    DBMS_OUTPUT.PUT_LINE('ID_SCAUN: ' || v_info_bilet.ID_SCAUN);
    DBMS_OUTPUT.PUT_LINE('ID_CLIENT: ' || v_info_bilet.ID_CLIENT);
END;
/
```


Worksheet
Query Builder
-- Adaugare client
GESTIONARE_BILETE.ADAUGACLIENT('Balaita', 'Cosmin', TO_DATE('2003-12-04', 'YYYY-MM-DD'), '123456789');
-- Afisare informatii despre clientul adaugat
v_info_client := GESTIONARE_BILETE.OBTINEREINFOCLIENT(1);
DBMS_OUTPUT.PUT_LINE('Informatii client:');
DBMS_OUTPUT.PUT_LINE('ID_CLIENT: ' || v_info_client.ID_CLIENT);
DBMS_OUTPUT.PUT_LINE('NUME: ' || v_info_client.NUME);
DBMS_OUTPUT.PUT_LINE('PRENUME: ' || v_info_client.PRENUME);
DBMS_OUTPUT.PUT_LINE('DATA_NASTERII: ' || TO_CHAR(v_info_client.DATA_NASTERII, 'YYYY-MM-DD'));
DBMS_OUTPUT.PUT_LINE('TELEFON: ' || v_info_client.TELEFON);
-- Adaugare bilet
Script Output
Task completed in 0,082 seconds
Package GESTIONARE_BILETE compiled
Package Body GESTIONARE_BILETE compiled
PL/SQL procedure successfully completed.
ProiectSGBD
Informatii client:
ID_CLIENT: 1
NUME: Balaita
PRENUME: Cosmin
DATA_NASTERII: 2003-12-04
TELEFON: 123456789
Informatii bilet:
ID_BILET: 1
PRET: 50
ID_PROIECTIE: 1
ID_SCAUN: 101
ID_CLIENT: 1
SQL History
Filter
Compiler - Log