

# GESTIONAREA UNUI CINAMATOGRAF

~PROIECT~

SISTEME DE GESTIONARE A BAZELOR DE  
DATE

FLORESCU MIRUNA-ȘTEFANIA  
GRUPA 233

# 1. Prezentarea pe scurt a bazei de date

Baza de date are ca scop gestionarea unui cinematograf. Principalul obiectiv al acesteia este ca cei care se ocupă de gestionarea cinematografului să aibă acces cât mai rapid la date și la informații despre difuzările filmelor, bilete, distribuitori, angajați și datele acestora. Organizarea datelor într-un sistem de gestionare a bazelor de date duce la eficientizarea organizării afacerii, la transmiterea de date mai eficientă și lipsită de erori și conduce la îmbunătățirea experienței clienților.

Baza de date este compusă din 10 tabele, dintre care 9 sunt entități independente, iar una este tabelă asociativă. Acestea sunt:

**CLIENT (#ID\_CLIENT)**

**BILET (#ID\_BILET)**

**CINEMA (#ID\_CLIENT)**

**DIFUZARE (#ID\_CINEMA, #ID\_FILM)**

**FILM (#ID\_FILM)**

**DISTRIBUTOR (ID\_DISTRIBUTOR)**

**ANGAJAT (#ID\_ANGAJAT)**

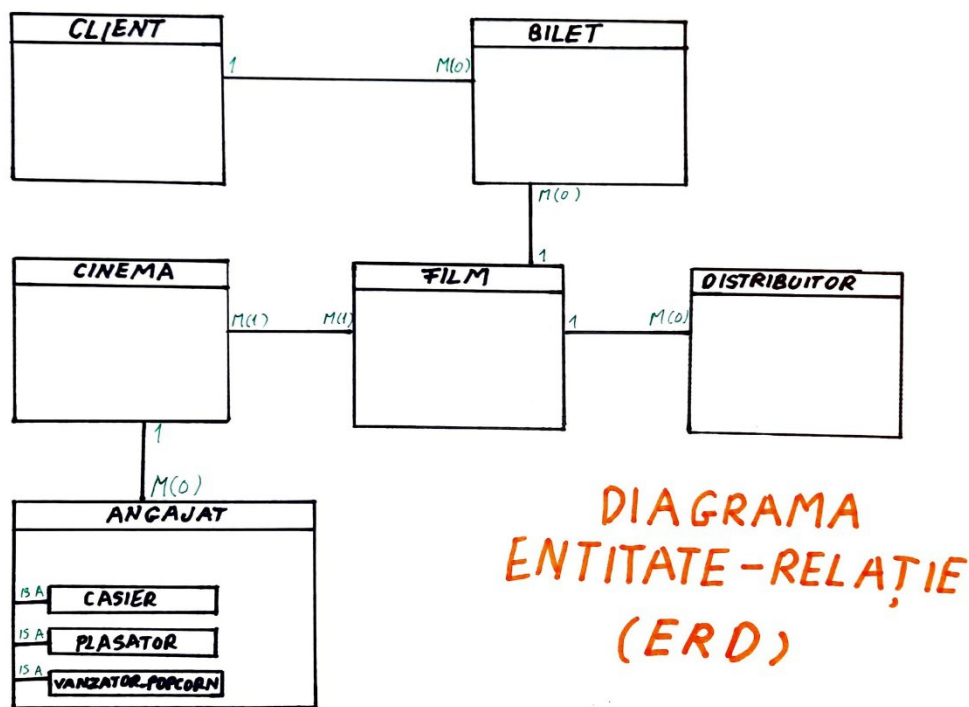
**CASIER (#ID\_ANGAJAT)**

**VANZATOR\_POPCORN (#ID\_ANGAJAT)**

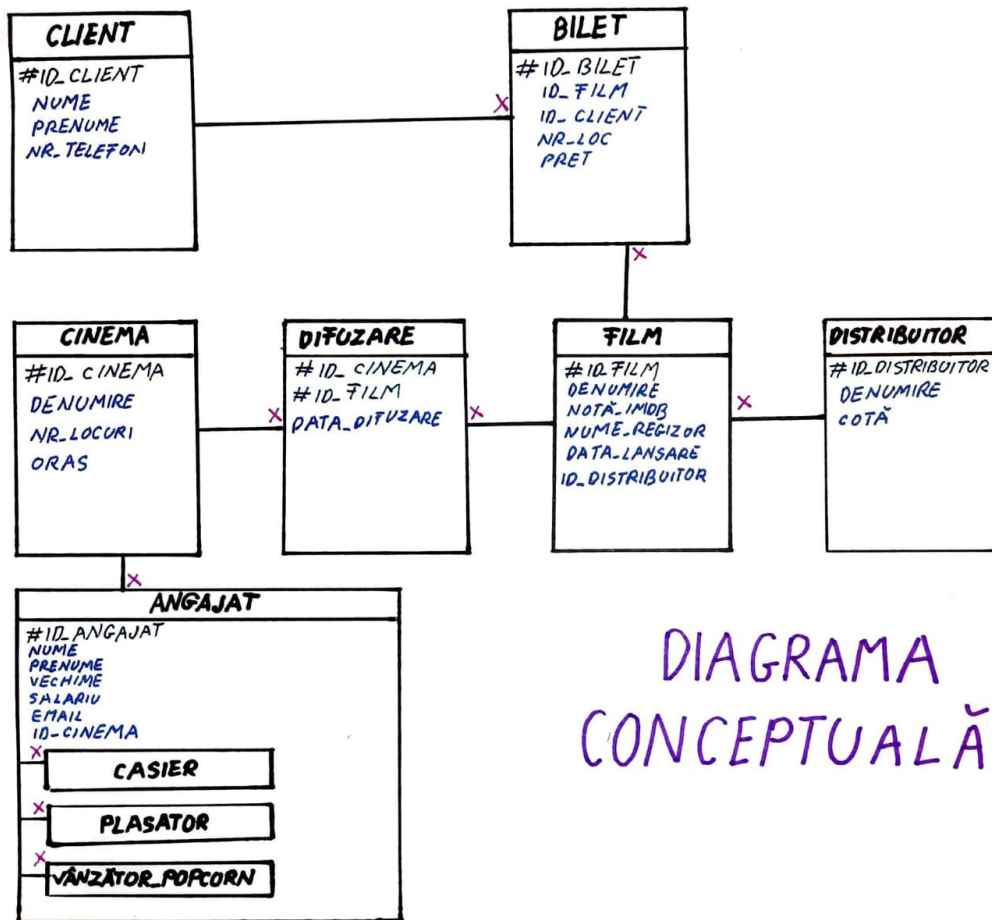
**PLASATOR (#ID\_ANGAJAT)**

Tabelele CASIER, VANZATOR, PLASATOR reprezintă tipuri de angajați și referă la tabela ANGAJAT. DISTRIBUTOR este tabela asociativă, care are cheia primară compusă din id-ul tablei CINEMA și id-ul tablei FILM.

## 2.Diagrama entitate-relație (ERD)



### 3. Diagrama conceptuală



## 4. Definirea tabelelor

```
CREATE TABLE CINEMA(  
    ID_CINEMA NUMBER(10) NOT NULL PRIMARY KEY,  
    DENUMIRE VARCHAR2(50),  
    NR_LOCURI NUMBER(10),  
    ORAS VARCHAR2(50)  
);
```

```
CREATE TABLE DISTRIBUTOR(  
    ID_DISTRIBUTOR NUMBER(10) NOT NULL PRIMARY KEY,  
    DENUMIRE VARCHAR2(50),  
    COTA FLOAT(10) --Cota de piata  
);
```

```
CREATE TABLE FILM(  
  
    ID_FILM NUMBER(10) NOT NULL PRIMARY KEY,  
    DENUMIRE VARCHAR2(50),  
    NOTA_IMDB FLOAT(10),  
    NUME_REGIZOR VARCHAR(100),  
    DATA_LANSARE DATE,  
    ID_DISTRIBUTOR NUMBER(10),  
  
    CONSTRAINT FK_FILM_DISTRIBUTOR FOREIGN KEY (ID_DISTRIBUTOR) REFERENCES  
    DISTRIBUTOR(ID_DISTRIBUTOR) ON DELETE CASCADE  
);
```

```
CREATE TABLE DIFUZARE(  
    ID_CINEMA NUMBER(10) NOT NULL,  
    ID_FILM NUMBER(10) NOT NULL,  
    DATA_DIFUZARE DATE DEFAULT SYSDATE,  
    CONSTRAINT FK_DIF_FILM FOREIGN KEY (ID_FILM) REFERENCES FILM(ID_FILM) ON DELETE CASCADE,  
    CONSTRAINT FK_DIF_CINEMA FOREIGN KEY (ID_CINEMA) REFERENCES CINEMA(ID_CINEMA) ON DELETE  
    CASCADE,
```

```
PRIMARY KEY(ID_FILM,ID_CINEMA)
);
```

```
CREATE TABLE CLIENT (
  ID_CLIENT NUMBER(10) NOT NULL PRIMARY KEY,
  NUME VARCHAR2(20),
  PRENUME VARCHAR2(20),
  NR_TELEFON NUMBER(10)
);
```

```
CREATE TABLE BILET(

  ID_BILET NUMBER(10) NOT NULL PRIMARY KEY,
  ID_FILM NUMBER(10),
  ID_CLIENT NUMBER(10),
  NR_LOC INT,
  PRET NUMBER(10),
  CONSTRAINT FK_BILET_FILM FOREIGN KEY (ID_FILM) REFERENCES FILM (ID_FILM) ON DELETE CASCADE,
  CONSTRAINT FK_BILET_CLIENT FOREIGN KEY (ID_CLIENT) REFERENCES CLIENT (ID_CLIENT) ON DELETE
  CASCADE
);
```

```
CREATE TABLE ANGAJAT(
  ID_ANGAJAT NUMBER(10) NOT NULL PRIMARY KEY,
  NUME VARCHAR2(20),
  PRENUME VARCHAR2(20),
  EMAIL VARCHAR2(50),
  SALARIU NUMBER(10),
  VECHIME NUMBER(10), -- Numarul de luni de cand este angajat
  ID_CINEMA,
  CONSTRAINT FK_ANGAJAT_CINEMA FOREIGN KEY (ID_CINEMA) REFERENCES CINEMA(ID_CINEMA) ON DELETE
  CASCADE
);
```

```
CREATE TABLE CASIER(
  ID_ANGAJAT NUMBER(10) NOT NULL PRIMARY KEY,
  PREGATIRE INT , --Nivelul de pregatire in domeniu
  CONSTRAINT FK_ANGAJAT_CASIER FOREIGN KEY(ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT) ON
  DELETE CASCADE
```

);

```
CREATE TABLE PLASATOR(  
    ID_ANGAJAT NUMBER(10) NOT NULL PRIMARY KEY,  
    CONSTRAINT FK_ANGAJAT_PLASATOR FOREIGN KEY(ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT) ON  
DELETE CASCADE,  
    TURA VARCHAR2(20) --Retinem daca lucreaza in tura de zi sau de noapte  
);
```

```
CREATE TABLE VANZATOR_POPCORN(  
    ID_ANGAJAT NUMBER(10) NOT NULL PRIMARY KEY,  
    CONSTRAINT FK_ANGAJAT_VANZATOR FOREIGN KEY(ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT) ON  
DELETE CASCADE,  
    NR_ORE VARCHAR2(20) --Numarul de ore lucrate pe saptamana  
);
```

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

## 5) Inserarea informațiilor în tabele

### CINEMA

```
INSERT INTO CINEMA VALUES(1,'Cinema City',260,'Targu Jiu');
INSERT INTO CINEMA VALUES(2,'Cityplex',100,'Cluj');
INSERT INTO CINEMA VALUES(3,'Movieplex City',200,'Timisoara');
INSERT INTO CINEMA VALUES(4,'Hollywood Multiplex Operations',300,'Bucuresti');
INSERT INTO CINEMA VALUES(5,'Grand Cinema',400,'Iasi');
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

ID_CINEMA	DENUMIRE	NR_LOCURI	ORAS
1	Cinema City	260	Targu Jiu
2	Cityplex	100	Cluj
3	Movieplex City	200	Timisoara
4	Hollywood Multiplex Operations	300	Bucuresti
5	Grand Cinema	400	Iasi

### DISTRIBUTOR

```
INSERT INTO DISTRIBUTOR VALUES(1,'Intercom Film',11.1);
INSERT INTO DISTRIBUTOR VALUES(2,'Forum Film',17.9);
INSERT INTO DISTRIBUTOR VALUES(3,'Ro-Image 200',17.9);
INSERT INTO DISTRIBUTOR VALUES(4,'MediaPro Distribution',24.5);
```



INSERT INTO DISTRIBUTOR VALUES(5,'Odeon Cineplex',16.1);

ID_DISTRIBUTOR	DENUMIRE	COTA
1	Intercom Film	11.1
2	Forum Film	17.9
3	Ro-Image 200	17.9
4	MediaPro Distribution	24.5
5	Odeon Cineplex	16.1

## FILM

INSERT INTO FILM VALUES(1,'Interstellar',8.7,'Christopher Nolan',TO\_DATE('7-11-2014','DD-MM-YYYY'),1);  
 INSERT INTO FILM VALUES(2,'Inception',8.8,'Christopher Nolan',TO\_DATE('30-7-2010','DD-MM-YYYY'),2);  
 INSERT INTO FILM VALUES(3,'Edward Scissorhands',7.9,'Tim Burton',TO\_DATE('14-12-1990','DD-MM-YYYY'),1);  
 INSERT INTO FILM VALUES(4,'Final Destination',6.7,'James Wong',TO\_DATE('14-3-2000','DD-MM-YYYY'),3);  
 INSERT INTO FILM VALUES(5,'La vita e bella',8.6,'Roberto Benigni',TO\_DATE('10-10-1997','DD-MM-YYYY'),3);

ID_FILM	DENUMIRE	NOTA_IMDB	NUME_REGIZOR	DATA_LANSARE	ID_DISTRIBUTOR
1	Interstellar	8.7	Christopher Nolan	07-NOV-14	1
2	Inception	8.8	Christopher Nolan	30-JUL-10	2
3	Edward Scissorhands	7.9	Tim Burton	14-DEC-90	1
4	Final Destination	6.7	James Wong	14-MAR-00	3
5	La vita e bella	8.6	Roberto Benigni	10-OCT-97	3

## DIFUZARE

INSERT INTO DIFUZARE VALUES(1,2,TO\_DATE('17-4-2019','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(1,1,TO\_DATE('17-4-2019','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(3,2,TO\_DATE('16-1-2019','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(2,2,TO\_DATE('17-12-2018','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(5,1,TO\_DATE('7-7-2009','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(1,5,TO\_DATE('16-8-2018','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(5,3,TO\_DATE('4-9-2019','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(3,3,TO\_DATE('5-11-2022','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(4,3,TO\_DATE('9-3-2017','DD-MM-YYYY'));  
 INSERT INTO DIFUZARE VALUES(1,3,TO\_DATE('23-12-2022','DD-MM-YYYY'));

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

ID_CINEMA	ID_FILM	DATA_DIFUZARE
1	2	17-APR-19
1	1	17-APR-19
3	2	16-JAN-19
2	2	17-DEC-18
5	1	07-JUL-09
1	5	16-AUG-18
5	3	04-SEP-19
3	3	05-NOV-22
4	3	09-MAR-17
1	3	23-DEC-22

## BILET

INSERT INTO BILET VALUES(1,2,1,104,20);

INSERT INTO BILET VALUES(2,1,1,200,25);

INSERT INTO BILET VALUES(3,1,6,10,25);

INSERT INTO BILET VALUES(4,1,5,34,25);

INSERT INTO BILET VALUES(5,5,4,275,30);

ID_BILET	ID_FILM	ID_CLIENT	NR_LOC	PRET
1	2	1	104	20
2	1	1	200	25
3	1	6	10	25
4	1	5	34	25
5	5	4	275	30

## CLIENT

```
INSERT INTO CLIENT VALUES(1,'Ciobanu','Marius',0737485509);
INSERT INTO CLIENT VALUES(2,'Popescu','Flavius',0733685509);
INSERT INTO CLIENT VALUES(3,'Dragu','Daniel',0737000899);
INSERT INTO CLIENT VALUES(4,'Boltasu','Andreea',0787485009);
INSERT INTO CLIENT VALUES(5,'Lungu','Laurentiu',0735679509);
INSERT INTO CLIENT VALUES(6,'Lungu','Marian',0745469509);
```

ID_CLIENT	NUME	PRENUME	NR_TELEFON
1	Ciobanu	Marius	737485509
2	Popescu	Flavius	733685509
3	Dragu	Daniel	737000899
4	Boltasu	Andreea	787485009
5	Lungu	Laurentiu	735679509
6	Lungu	Marian	745469509

## ANGAJAT

```
INSERT INTO ANGAJAT VALUES(1,'Munteanu','Teodor','munteanu_teodor@gmail.com',4000,24,1);
INSERT INTO ANGAJAT VALUES(2,'Neamtu','Iulia','neamtu.iulia20@yahoo.com',2000,12,3);
INSERT INTO ANGAJAT VALUES(3,'Florea','Mariana','florea_mariana@yahoo.com',3000,13,5);
INSERT INTO ANGAJAT VALUES(4,'Badea','Liliana','badea.liliana3000@gmail.com',2000,6,4);
INSERT INTO ANGAJAT VALUES(5,'Stefanescu','Ion','stefanescu_ion@yahoo.com',2000,3,1);
```

ID_ANGAJAT	NUME	PRENUME	EMAIL	SALARIU	VECHIME	ID_CINEMA
1	Munteanu	Teodor	munteanu_teodor@gmail.com	4000	24	1
2	Neamtu	Iulia	neamtu.iulia20@yahoo.com	2000	12	3
3	Florea	Mariana	florea_mariana@yahoo.com	3000	13	5
4	Badea	Liliana	badea.liliana3000@gmail.com	2000	6	4
5	Stefanescu	Ion	stefanescu_ion@yahoo.com	2000	3	1

## CASIER

```
INSERT INTO CASIER VALUES(1,1);
```

```
INSERT INTO CASIER VALUES(2,2);
```

```
INSERT INTO CASIER VALUES(3,3);
```

```
INSERT INTO CASIER VALUES(4,1);
```

```
INSERT INTO CASIER VALUES(5,2);
```

ID_ANGAJAT	PREGATIRE
1	1
2	2
3	3
4	1
5	2

## PLASATOR

```
INSERT INTO PLASATOR VALUES(1,'Zi');
```

```
INSERT INTO PLASATOR VALUES(2,'Noapte');
```

```
INSERT INTO PLASATOR VALUES(3,'Zi');
```

```
INSERT INTO PLASATOR VALUES(4,'Noapte');
```

```
INSERT INTO PLASATOR VALUES(5,'Noapte');
```

ID_ANGAJAT	TURA
1	Zi
2	Noapte
3	Zi
4	Noapte
5	Noapte

## VANZATOR\_POPCORN

```
INSERT INTO VANZATOR_POPCORN VALUES(1,40);
INSERT INTO VANZATOR_POPCORN VALUES(2,35);
INSERT INTO VANZATOR_POPCORN VALUES(3,38);
INSERT INTO VANZATOR_POPCORN VALUES(4,40);
INSERT INTO VANZATOR_POPCORN VALUES(5,30);
```

ID_ANGAJAT	NR_ORE
1	40
2	35
3	38
4	40
5	30

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

**Definiți un subprogram stocat care primește ca parametru numele unui regizor și afișează toate filmele din baza de date regizate de acesta. Apelați subprogramul.**

```
CREATE OR REPLACE PROCEDURE PROCEDURA_EX_6 (V_REGIZOR FILM.NUME_REGIZOR%TYPE)
IS TYPE LISTA_FILM IS TABLE OF VARCHAR2(50) INDEX BY PLS_INTEGER;
LISTA_FILME LISTA_FILM;
BEGIN
    SELECT DENUMIRE
    BULK COLLECT INTO LISTA_FILME
    FROM FILM
    WHERE NUME_REGIZOR=V_REGIZOR;

    FOR i IN LISTA_FILME.FIRST..LISTA_FILME.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Filmul ' || LISTA_FILME(i) || ' a fost regizat de ' || V_REGIZOR);
```

```

END LOOP;

END PROCEDURA_EX_6;
/
BEGIN
    PROCEDURA_EX_6('Christopher Nolan');
END;
/

BEGIN
    PROCEDURA_EX_6('Tim Burton');
END;
/

```

Procedure created.

Statement processed.

Filmul Interstellar a fost regizat de Christopher Nolan

Filmul Inception a fost regizat de Christopher Nolan

Statement processed.

Filmul Edward Scissorhands a fost regizat de Tim Burton

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

**Definiți un subprogram stocat care primește ca parametru numele unui cinema și afișează numele, preumele și salariul tuturor angajaților săi.**

```

CREATE OR REPLACE PROCEDURE PROCEDURA_EX_7 (V_DENUMIRE CINEMA.DENUMIRE%TYPE)
IS
    CURSOR c IS
    SELECT A.NUME,A.PRENUME, A.SALARIU
    FROM ANGAJAT A, CINEMA C
    WHERE A.ID_CINEMA = C.ID_CINEMA

```

```

AND C.DENUMIRE = V_DENUMIRE;
BEGIN
  FOR i IN c LOOP
    DBMS_OUTPUT.PUT_LINE('In Cinema-ul ' || V_DENUMIRE || ' este angajat ' || INITCAP(i.NUME) || ' ' ||
INITCAP(i.PRENUME) || ' , care are salariul de ' || i.SALARIU || ' de lei' );
  END LOOP;
END PROCEDURA_EX_7;
/
BEGIN
  PROCEDURA_EX_7('Cinema City');
END;
/

Procedure created.

Statement processed.
In Cinema-ul Cinema City este angajat Munteanu Teodor , care are salariul de 2000 de lei
In Cinema-ul Cinema City este angajat Stefanescu Ion , care are salariul de 2000 de lei

```

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

**Definiți un subprogram care primește ca parametru numele unui client si intoarce filmul cu nota mai mare de 8 pe IMDB la care acesta și-a luat bilet.**

```

CREATE OR REPLACE FUNCTION FUNCTIE_EX_8
( V_NUME CLIENT.NUME%TYPE )
RETURN VARCHAR IS
  V_NUME_FILM film.denumire%TYPE;
BEGIN

  SELECT F.DENUMIRE INTO V_NUME_FILM
  FROM CLIENT C, BILET B, FILM F

```

```
WHERE C.ID_CLIENT = B.ID_CLIENT
AND B.ID_FILM = F.ID_FILM
AND UPPER(C.NUME) = UPPER(V_NUME)
AND F.NOTA_IMDB > 8;
```

```
RETURN V_NUME_FILM;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
RAISE_APPLICATION_ERROR(-20000,'Nu exista niciun client cu acest nume care a vizionat filme cu nota peste
8 pe IMDB!');
```

```
WHEN TOO_MANY_ROWS THEN
```

```
RAISE_APPLICATION_ERROR(-20001,'Exista mai multi clienti cu acest nume sau un client care a vizionat mai
multe filme cu nota peste 8 pe IMDB!');
```

```
WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');
```

```
END FUNCTIE_EX_8;
```

```
/
```

```
Function created.
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||
FUNCTIE_EX_8('Ciobanu'));
```

```
END;
```

```
/
```

```
ORA-20001: Exista mai multi clienti cu acest nume sau un client care a vizionat mai multe filme cu nota peste 8 pe IMDB! ORA-06512: at "SQL_QAW#D3BH3ZDRHEWYVBVIRBQMFQ.FUNCTIE_EX_8", line 22
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||
FUNCTIE_EX_8('Boltasu'));
```

```
END;
```

```
/
```

```
Statement processed.
```

```
Filmul cu nota peste 8 pe IMDB vizionat de client este: La vita e bella
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||
FUNCTIE_EX_8('Ionescu'));
```



END;

/

```
ORA-20000: Nu exista niciun client cu acest nume care a vizionat filme cu nota peste 8 pe IMDB! ORA-06512: at "SQL_QAWMJBHJZDRHEWYVBVIRBQMFQ.FUNCTIE_EX_8", line 20
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' || FUNCTIE\_EX\_8('Lungu'));

END;

/

```
ORA-20001: Exista mai multi clienti cu acest nume sau un client care a vizionat mai multe filme cu nota peste 8 pe IMDB! ORA-06512: at "SQL_QAWMJBHJZDRHEWYVBVIRBQMFQ.FUNCTIE_EX_8", line 22
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

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

**Definiți un subprogram stocat de tip procedură care primește ca parametru denumirea unui cinematograf si o data și afișează orașul în care se afla Cinema-ul, denumirea filmului difuzat in acea dată, distribuitorul acestuia si prețul biletului.**

```
CREATE OR REPLACE PROCEDURE PROCEDURA_EX_9 (V_DENUMIRE_CINEMA CINEMA.DENUMIRE%TYPE,
V_DATA DIFUZARE.DATA_DIFUZARE%TYPE)
```

IS

V\_ORAS CINEMA.ORAS%TYPE;

V\_DENUMIRE\_FILM FILM.DENUMIRE%TYPE;

V\_NUME\_DISTRIBUTOR DISTRIBUTOR.DENUMIRE%TYPE;

V\_PRET BILET.PRET%TYPE;

BEGIN

```
SELECT DISTINCT C.ORAS, F.DENUMIRE, DS.DENUMIRE,B.PRET INTO
V_ORAS,V_DENUMIRE_FILM,V_NUME_DISTRIBUTOR,V_PRET-- Folosim DISTINCT pentru ca putem avea mai
multe bilete la acelasi film pe locuri diferite
```

```
FROM FILM F, DIFUZARE D, CINEMA C, DISTRIBUTOR DS, BILET B
```

```
WHERE F.ID_FILM = D.ID_FILM
```

```
AND D.ID_CINEMA = C.ID_CINEMA
AND DS.ID_DISTRIBUTOR = F.ID_DISTRIBUTOR
AND B.ID_FILM = F.ID_FILM
AND C.DENUMIRE = V_DENUMIRE_CINEMA
AND D.DATA_DIFUZARE = V_DATA;
```

```
DBMS_OUTPUT.PUT_LINE('In orasul ' || V_ORAS || ' se difuzeaza filmul ' || V_DENUMIRE_FILM || ', distribuit
de ' || V_NUME_DISTRIBUTOR || ' si are pretul biletului de ' || V_PRET || ' lei.');
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
RAISE_APPLICATION_ERROR(-20001,'La aceasta data nu se difuzeaza niciun film la acest cinema');
```

```
WHEN TOO_MANY_ROWS THEN
```

```
RAISE_APPLICATION_ERROR(-20002,'La aceasta data la cinema se difuzeaza mai multe filme in acelasi timp!');
```

```
WHEN OTHERS THEN
```

```
RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');
```

```
END PROCEDURA_EX_9;
```

```
/
```

```
Procedure created.
```

```
BEGIN
```

```
PROCEDURA_EX_9('Cinema City','16-AUG-18');
```

```
END;
```

```
/
```

```
Statement processed.
```

```
In orasul Targu Jiu se difuzeaza filmul La vita e bella, distribuit de Ro-Image 200 si are pretul biletului de 30 lei.
```

```
BEGIN
```

```
PROCEDURA_EX_9('Cinema City','17-APR-19');
```

```
END;
```

```
/
```

```
ORA-20005: La aceasta data la cinema se difuzeaza mai multe filme in acelasi timp! ORA-06512: at "SQL_WGQKLWKSABCIDLZFPBAUZWNCIM.PROCEDURA_EX_9", line 22
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
BEGIN
```

```
PROCEDURA_EX_9('Cinema City','20-MAY-19');
```

```
END;
```

```
/
```

```
ORA-20004: La aceasta data nu se difuzeaza niciun film la acest cinema ORA-06512: at "SQL_WGQKLWKSABCDLZFPBAUZWCM.PROCEDURA_EX_9", line 20
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

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

**Definiți un trigger care să nu permită inserarea unui film în tabela DIFUZARE în perioada 14.03.2020 - 20-06-2021, cinematografele fiind închise din cauza pandemiei.**

```
CREATE OR REPLACE TRIGGER TRIGGER_EX_10
BEFORE INSERT ON DIFUZARE
BEGIN
    IF SYSDATE BETWEEN TO_DATE('14-03-2020','DD-MM-YYYY') AND TO_DATE('20-06-2021','DD-MM-
YYYY')
    THEN RAISE_APPLICATION_ERROR(-20001,'Cinematografele sunt inchise si nu pot fi difuzate filme!');
    END IF;
END;
/
INSERT INTO DIFUZARE VALUES(2,4,TO_DATE(SYSDATE));
/
DROP TRIGGER TRIGGER_EX_10;
/
```

Trigger created.

```
ORA-20001: Cinematografele sunt inchise din cauza pandemiei si nu pot fi difuzate filme in aceasta perioada! ORA-06512: at "SQL_WGQKLWKSABCDLZFPBAUZWCM.TRIGGER_EX_10", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

Trigger dropped.

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

**Definiți un trigger de tip LMD la nivel de linie care să nu permită mărirea salariului cu mai mult de 30% pentru angajații cu vechime de cel puțin un an , respectiv 15% pentru angajații cu vechime mai mică de un an.**

```
CREATE OR REPLACE TRIGGER TRIGGER_EX_11
BEFORE UPDATE OF SALARIU ON ANGAJAT
FOR EACH ROW
BEGIN
  IF (:OLD.VECHIME < 12 AND :NEW.SALARIU > 1.15 * :OLD.SALARIU) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Nu se poate mari salariul angajatului cu mai mult de 15%!');
  END IF;
  IF (:OLD.VECHIME >= 12 AND :NEW.SALARIU > 1.3 * :OLD.SALARIU) THEN
    RAISE_APPLICATION_ERROR(-20002, 'Nu se poate mari salariul angajatului cu mai mult de 30%!');
  END IF;
END;
/
UPDATE ANGAJAT
SET SALARIU = SALARIU * 1.17
WHERE VECHIME < 12;
/
```

Trigger created.

ORA-20001: Nu se poate mari salariul angajatului cu mai mult de 15%!  
ORA-06512: at "SQL\_BHSTQHDAGUHHVCYHKOAFBMESH.TRIGGER\_EX\_11", line 3  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

```
UPDATE ANGAJAT
SET SALARIU = SALARIU * 3.1
WHERE VECHIME >= 12;
/
```

ORA-20002: Nu se poate mari salariul angajatului cu mai mult de 30%! ORA-06512: at "SQL\_BHSTQHDAGUH#VCYHKOAFBMESH.TRIGGER\_EX\_11", line 6  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

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

**Definiți un trigger de tip LDD care nu va permite ștergerea tabelii DIFUZARE dacă în tabelă sunt înregistrate 10 sau mai multe difuzări ale filmelor. Declanșați trigger-ul!**

```
CREATE OR REPLACE TRIGGER TRIGGER_EX_12
BEFORE DROP ON SCHEMA
DECLARE
NR_DIFUZARI NUMBER(10);
BEGIN
IF(SYS.DICTIONARY_OBJ_NAME='DIFUZARE') THEN

SELECT COUNT(*) INTO NR_DIFUZARI
FROM DIFUZARE;

IF (NR_DIFUZARI)>=10 THEN
RAISE_APPLICATION_ERROR(-20001,'Nu puteti efectua aceasta comanda pentru ca prea multe date ar fi
pierdute!');
END IF;

END IF;
END;
/

DROP TABLE DIFUZARE;
/
```

```

Error report -
ORA-00604: error occurred at recursive SQL level 1
ORA-20001: Nu puteti efectua aceasta comanda pentru ca prea multe date ar fi pierdute!
ORA-06512: at line 10
00604. 00000 - "error occurred at recursive SQL level %s"
*Cause:      An error occurred while processing a recursive SQL statement

```

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

```
CREATE OR REPLACE PACKAGE PACHET_EX_13 AS
```

```

    PROCEDURE PROCEDURA_EX_6 (V_REGIZOR FILM.NUME_REGIZOR%TYPE);
    PROCEDURE PROCEDURA_EX_7 (V_DENUMIRE CINEMA.DENUMIRE%TYPE);
    FUNCTION FUNCTIE_EX_8( V_NUME CLIENT.NUME%TYPE ) RETURN VARCHAR;
    PROCEDURE PROCEDURA_EX_9 (V_DENUMIRE_CINEMA CINEMA.DENUMIRE%TYPE, V_DATA
    DIFUZARE.DATA_DIFUZARE%TYPE);
END PACHET_EX_13;
/

```

```
CREATE OR REPLACE PACKAGE BODY PACHET_EX_13 AS
```

```

    PROCEDURE PROCEDURA_EX_6 (V_REGIZOR FILM.NUME_REGIZOR%TYPE)
    IS TYPE LISTA_FILM IS TABLE OF VARCHAR2(50) INDEX BY PLS_INTEGER;
    LISTA_FILME LISTA_FILM;
BEGIN
    SELECT DENUMIRE
    BULK COLLECT INTO LISTA_FILME
    FROM FILM
    WHERE NUME_REGIZOR=V_REGIZOR;

    FOR i IN LISTA_FILME.FIRST..LISTA_FILME.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Filmul ' || LISTA_FILME(i) || ' a fost regizat de ' || V_REGIZOR);
    END LOOP;

```

```
END PROCEDURA_EX_6;
```

```

PROCEDURE PROCEDURA_EX_7 (V_DENUMIRE CINEMA.DENUMIRE%TYPE)
IS

```

```

CURSOR c IS
SELECT A.NUME,A.PRENUME, A.SALARIU
FROM ANGAJAT A, CINEMA C
WHERE A.ID_CINEMA = C.ID_CINEMA
AND C.DENUMIRE = V_DENUMIRE;
BEGIN
FOR i IN c LOOP
    DBMS_OUTPUT.PUT_LINE('In Cinema-ul ' || V_DENUMIRE || ' este angajat ' || INITCAP(i.NUME) || ' ' ||
INITCAP(i.PRENUME) || ' , care are salariul de ' || i.SALARIU || ' de lei' );
    END LOOP;
END PROCEDURA_EX_7;

FUNCTION FUNCTIE_EX_8( V_NUME CLIENT.NUME%TYPE )
RETURN VARCHAR IS
    V_NUME_FILM film.denumire%TYPE;
BEGIN

    SELECT F.DENUMIRE INTO V_NUME_FILM
    FROM CLIENT C, BILET B, FILM F
    WHERE C.ID_CLIENT = B.ID_CLIENT
    AND B.ID_FILM = F.ID_FILM
    AND UPPER(C.NUME) = UPPER(V_NUME)
    AND F.NOTA_IMDB > 8;

    RETURN V_NUME_FILM;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000,'Nu exista niciun client cu acest nume care a vizionat filme cu nota
peste 8 pe IMDB!');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20001,'Exista mai multi clienti cu acest nume sau un client care a vizionat mai
multe filme cu nota peste 8 pe IMDB!');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002,'Alta eroare!');

END FUNCTIE_EX_8;

PROCEDURE PROCEDURA_EX_9 (V_DENUMIRE_CINEMA CINEMA.DENUMIRE%TYPE, V_DATA
DIFUZARE.DATA_DIFUZARE%TYPE)
IS
    V_ORAS CINEMA.ORAS%TYPE;

```

```

V_DENUMIRE_FILM FILM.DENUMIRE%TYPE;
V_NUME_DISTRIBUTOR DISTRIBUTOR.DENUMIRE%TYPE;
V_PRET BILET.PRET%TYPE;
BEGIN
    SELECT DISTINCT C.ORAS, F.DENUMIRE, DS.DENUMIRE,B.PRET INTO
V_ORAS,V_DENUMIRE_FILM,V_NUME_DISTRIBUTOR,V_PRET-- Folosim DISTINCT pentru ca putem avea mai
multe bilete la acelasi film pe locuri diferite
    FROM FILM F, DIFUZARE D, CINEMA C, DISTRIBUTOR DS, BILET B
    WHERE F.ID_FILM = D.ID_FILM
    AND D.ID_CINEMA = C.ID_CINEMA
    AND DS.ID_DISTRIBUTOR = F.ID_DISTRIBUTOR
    AND B.ID_FILM = F.ID_FILM
    AND C.DENUMIRE = V_DENUMIRE_CINEMA
    AND D.DATA_DIFUZARE = V_DATA;

    DBMS_OUTPUT.PUT_LINE('In orasul ' || V_ORAS || ' se difuzeaza filmul ' || V_DENUMIRE_FILM || ',
distribuit de ' || V_NUME_DISTRIBUTOR || ' si are pretul biletului de ' || V_PRET || ' lei. ');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001,'La aceasta data nu se difuzeaza niciun film la acest cinema');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20002,'La aceasta data la cinema se difuzeaza mai multe filme in acelasi timp!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20003,'Alta eroare!');
END PROCEDURA_EX_9;

END PACHET_EX_13;
/
BEGIN
    PACHET_EX_13.PROCEDURA_EX_6('Christopher Nolan');

    PACHET_EX_13.PROCEDURA_EX_7('Cinema City');
    PACHET_EX_13.PROCEDURA_EX_7('Grand Cinema');

    PACHET13.PROCEDURA_EX_9('Cinema City','16-AUG-18');
    --PACHET_EX_13.PROCEDURA_EX_9('Cinema City','17-APR-19');
    --PACHET_EX_13.PROCEDURA_EX_9('Cinema City','20-MAY-19');

    DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||
PACHET_EX_13.FUNCTIE_EX_8('Boltasu'));

```



```
-- DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||  
PACHET_EX_13.FUNCTIE_EX_8('Ciobanu'));
```

```
-- DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||  
PACHET_EX_13.FUNCTIE_EX_8('Ionescu'));
```

```
END;
```

```
/
```

```
Statement processed.  
Filmul Interstellar a fost regizat de Christopher Nolan  
Filmul Inception a fost regizat de Christopher Nolan  
In Cinema-ul Cinema City este angajat Munteanu Teodor , care are salariul de 4000 de lei  
In Cinema-ul Cinema City este angajat Stefanescu Ion , care are salariul de 2000 de lei  
In Cinema-ul Grand Cinema este angajat Florea Mariana , care are salariul de 3000 de lei  
In orasul Targu Jiu se difuzeaza filmul La vita e bella, distribuit de Ro-Image 200 si are pretul biletului de 30 lei.  
Filmul cu nota peste 8 pe IMDB vizionat de client este: La vita e bella
```

```
PACHET_EX_13.PROCEDURA_EX_9('Cinema City','17-APR-19');
```

```
ORA-20002: La aceasta data la cinema se difuzeaza mai multe filme in acelasi timp! ORA-06512: at "SQL_NORFVTAWMNSJYTDGZMKTW.PACHET_EX_13", line 75  
ORA-06512: at line 8  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
PACHET_EX_13.PROCEDURA_EX_9('Cinema City','20-MAY-19');
```

```
ORA-20001: La aceasta data nu se difuzeaza niciun film la acest cinema ORA-06512: at "SQL_NORFVTAWMNSJYTDGZMKTW.PACHET_EX_13", line 73  
ORA-06512: at line 9  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||  
PACHET_EX_13.FUNCTIE_EX_8('Ciobanu'));
```

```
ORA-20001: Exista mai multi clienti cu acest nume sau un client care a vizionat mai multe filme cu nota peste 8 pe IMDB! ORA-06512: at "SQL_NORFVTAWMNSJYTDGZMKTW.PACHET_EX_13", line 49  
ORA-06512: at line 12  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
DBMS_OUTPUT.PUT_LINE('Filmul cu nota peste 8 pe IMDB vizionat de client este: ' ||  
PACHET_EX_13.FUNCTIE_EX_8('Ionescu'));
```

```
ORA-20000: Nu exista niciun client cu acest nume care a vizionat filme cu nota peste 8 pe IMDB! ORA-06512: at "SQL_NORFVTAWMNSJYTDGZMKTW.PACHET_EX_13", line 47  
ORA-06512: at line 13  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

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

**Definiți un pachet care să conțină un vector de filme, un vector de distribuitori și o procedură care primește ca parametru denumirea unui film și afișează data lansării, distribuitorul și cota de piață a acestuia cu ajutorul vectorilor!**

```
CREATE OR REPLACE PACKAGE PACHET_EX_14 AS
```

```
TYPE FILM_TYPE IS RECORD(  
  ID FILM.ID_FILM%TYPE,  
  NUME FILM.DENUMIRE%TYPE,  
  DATA FILM.DATA_LANSARE%TYPE  
);
```

```
FILM_OBJ FILM_TYPE;
```

```
TYPE DISTRIBUTOR_TYPE IS RECORD(  
  ID DISTRIBUTOR.ID_DISTRIBUTOR%TYPE,  
  NUME DISTRIBUTOR.DENUMIRE%TYPE,  
  COTA_PIATA DISTRIBUTOR.COTA%TYPE );
```

```
DISTRIBUTOR_OBJ DISTRIBUTOR_TYPE;
```

```
NR_FILME NUMBER(10);
```

```
NR_DISTRIBUTORI NUMBER(10);
```

```
TYPE VECTOR_FILME IS VARRAY(100) OF FILM_TYPE;  
VECTOR1 VECTOR_FILME:= VECTOR_FILME();
```

```
TYPE VECTOR_DISTRIBUTORI IS VARRAY(100) OF DISTRIBUTOR_TYPE;  
VECTOR2 VECTOR_DISTRIBUTORI:= VECTOR_DISTRIBUTORI();
```

```
PROCEDURE PROCEDURA(V_DENUMIRE FILM.DENUMIRE%TYPE);  
END PACHET_EX_14;  
/
```

```
CREATE OR REPLACE PACKAGE BODY PACHET_EX_14 AS
```

```
PROCEDURE PROCEDURA(V_DENUMIRE FILM.DENUMIRE%TYPE)  
IS  
ID_FILM_OBJ FILM.ID_FILM%TYPE;  
ID_DISTRIBUTOR_OBJ DISTRIBUTOR.ID_DISTRIBUTOR%TYPE;
```

```
BEGIN
```

```
SELECT COUNT(*) INTO NR_FILME FROM FILM;
```

```
FOR i IN 1..NR_FILME LOOP
```

```
SELECT ID_FILM,DENUMIRE,DATA_LANSARE  
INTO FILM_OBJ  
FROM FILM  
WHERE ID_FILM=i;  
VECTOR1.EXTEND;  
VECTOR1(i):=FILM_OBJ;  
END LOOP;
```

```
SELECT COUNT(*) INTO NR_DISTRIBUTORI FROM DISTRIBUTOR;
```

```
FOR i IN 1..NR_DISTRIBUTORI LOOP  
SELECT ID_DISTRIBUTOR,DENUMIRE,COTA  
INTO DISTRIBUTOR_OBJ  
FROM DISTRIBUTOR
```

```

        WHERE ID_DISTRIBUTOR=i;
        VECTOR2.EXTEND;
        VECTOR2(i):=DISTRIBUTOR_OBJ;
    END LOOP;

    SELECT ID_FILM,ID_DISTRIBUTOR
    INTO ID_FILM_OBJ,ID_DISTRIBUTOR_OBJ
    FROM FILM
    WHERE DENUMIRE=V_DENUMIRE;

    FOR i IN 1..NR_FILME LOOP
        IF(VECTOR1(i).ID=ID_FILM_OBJ) THEN
            DBMS_OUTPUT.PUT_LINE('Filmul ' || V_DENUMIRE || ' are data de lansare: ' || VECTOR1(i).DATA
|| '.');
        END IF;
    END LOOP;

    FOR i IN 1..NR_DISTRIBUTORI LOOP
        IF(VECTOR2(i).ID=ID_DISTRIBUTOR_OBJ) THEN
            DBMS_OUTPUT.PUT_LINE('Filmul ' || V_DENUMIRE || ' este distribuit de catre ' ||
INITCAP(VECTOR2(i).NUME) || ', care are cota de piata ' || VECTOR2(i).COTA_PIATA || '.');
        END IF;
    END LOOP;

    END PROCEDURA;

END PACHET_EX_14;
/

Package created.

Package Body created.

BEGIN
    PACHET_EX_14.PROCEDURA('Edward Scissorhands');
END;
```

Statement processed.

Filmul Edward Scissorhands are data de lansare: 14-DEC-90.

Filmul Edward Scissorhands este distribuit de catre Intercom Film , care are cota de piata 11.1.