Sisteme de Gestiune a Bazelor de Date

# GESTIUNEA UNUI SITE DE FILME

Micu Diana-Roberta

Grupa 244

Facultatea de Matematică și Informatică

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

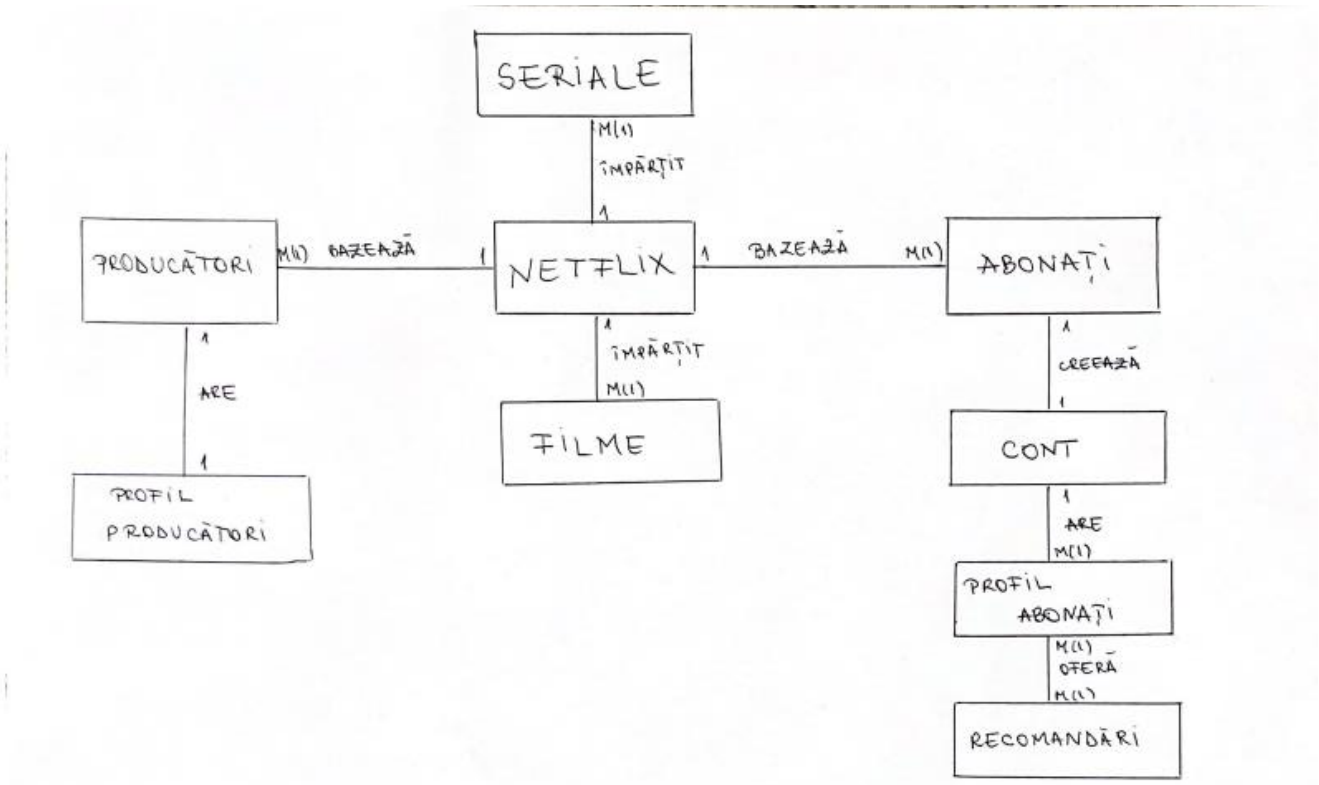
Compania americană de filme Netflix se bazează pe două tipuri de utilizatori: abonați și producători. Producătorii sunt cei care încarcă filmele pe site, în timp ce abonații sunt cei care vizionează. Netflix este împărțit în 2 categorii: filme și seriale.

Abonații își creează un cont cu datele personale: nume, prenume, email, telefon, data nașterii, numărul cardului. De asemenea, se cunoaște data la care a fost activat contul. Fiecare cont are dreptul la maxim 5 profiluri individuale. Profilurile pot fi stabilite în funcție de categoria de vârstă (adulți sau copii). Fiecare profil va oferi separat recomandări în funcție de vârstă și preferințe.

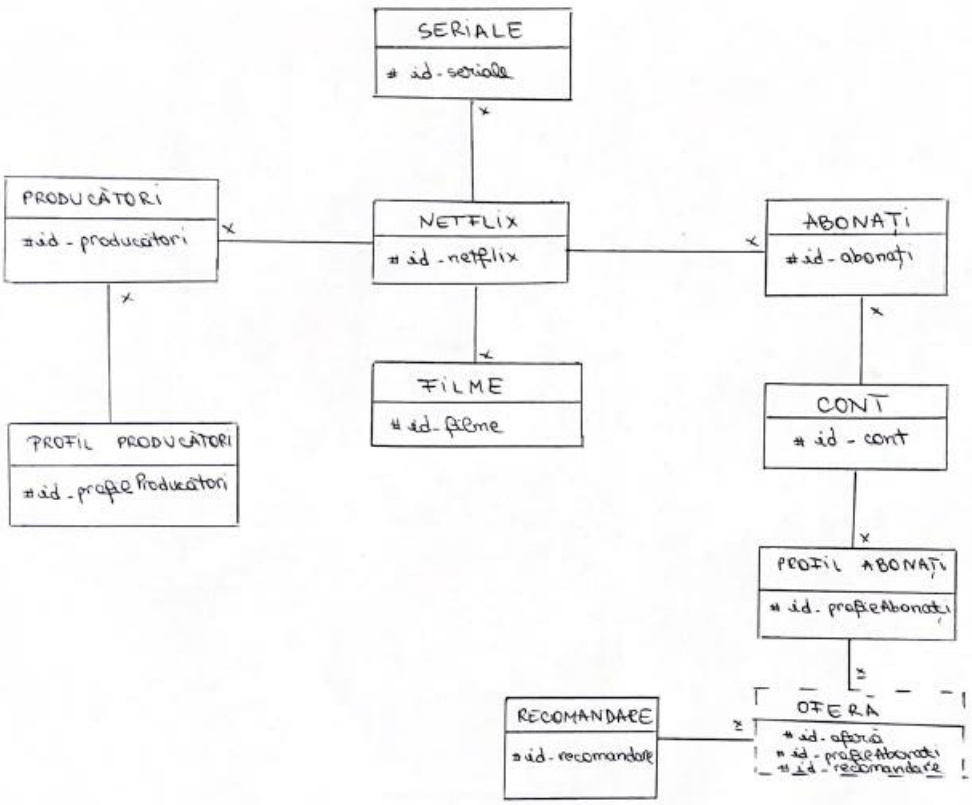
Profilul unui producător va include informații despre filmul pe care dorește să îl încarce: anul apariției, genul, limita de vârstă, recenzii.

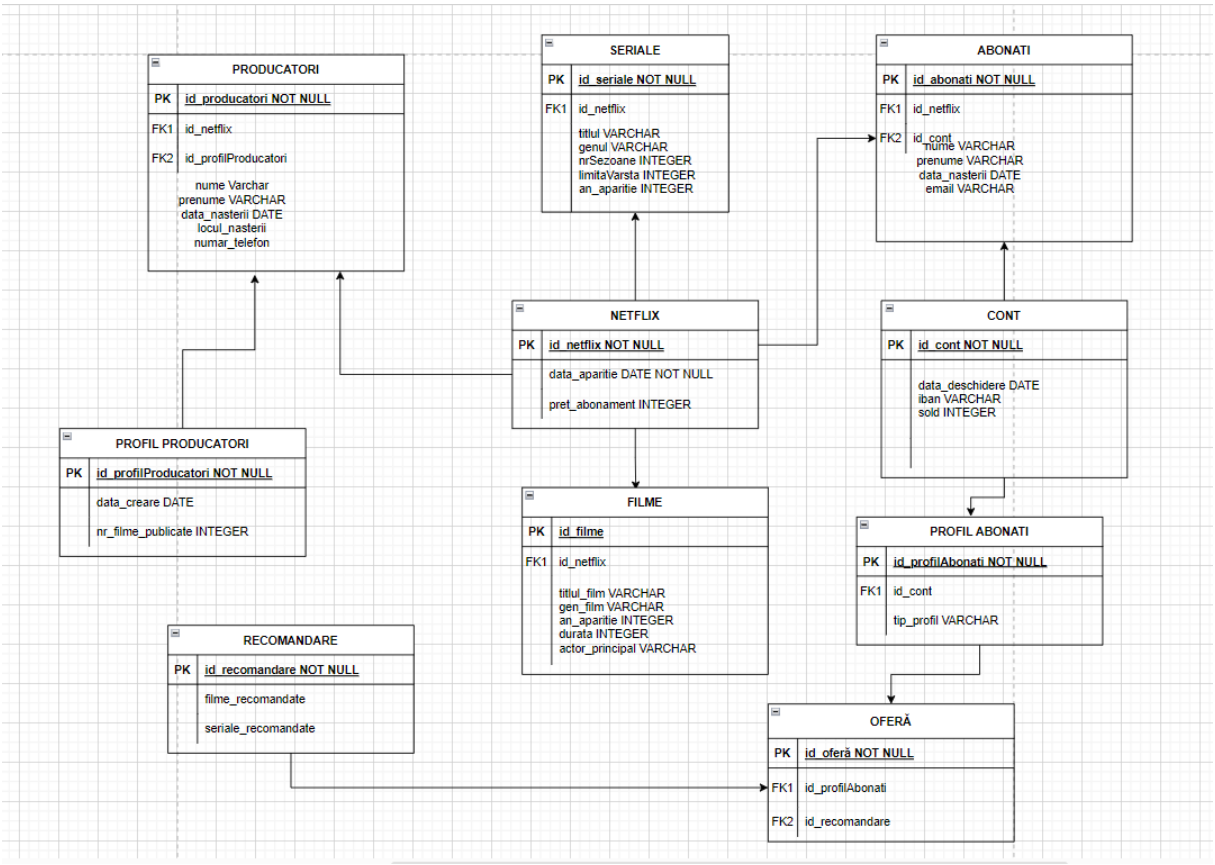
În cazul în care abonații și producătorii au avut parte de situații nefericite, ei pot închide contul oricând.

2. Realizați diagrama entitate-relație (ERD).




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


SERIALE
# id - seriale
PRODUCĂTORI
# id - producători
NETFLIX
# id - netflix
ABONAȚI
# id - abonați
FILME
# id - filme
CONT
# id - cont
PROFIL PRODUCĂTORI
# id - profilProducători
PROFIL ABONAȚI
# id - profilAbonați
RECOMANDARE
# id - recomandare
OFERĂ
# id - oferă
# id - profilAbonați
# id - recomandare


PRODUCATORI
PK id_producatori NOT NULL
FK1 id_netflix
FK2 id_profilProducatori
nume Varchar
prenume VARCHAR
data_nasterii DATE
locul_nasterii
numar_telefon
SERIALE
PK id_seriale NOT NULL
FK1 id_netflix
titlul VARCHAR
genul VARCHAR
nrSezoane INTEGER
limitaVarsta INTEGER
an_aparitie INTEGER
ABONATI
PK id_abonati NOT NULL
FK1 id_netflix
FK2 id_cont
nume VARCHAR
prenume VARCHAR
data_nasterii DATE
email VARCHAR
NETFLIX
PK id_netflix NOT NULL
data_aparitie DATE NOT NULL
pret_abonament INTEGER
CONT
PK id_cont NOT NULL
data_deschidere DATE
iban VARCHAR
sold INTEGER
PROFIL PRODUCATORI
PK id_profilProducatori NOT NULL
data_creare DATE
nr_filme_publicate INTEGER
FILME
PK id_filme
FK1 id_netflix
titlul_film VARCHAR
gen_film VARCHAR
an_aparitie INTEGER
durata INTEGER
actor_principal VARCHAR
PROFIL ABONATI
PK id_profilAbonati NOT NULL
FK1 id_cont
tip_profil VARCHAR
RECOMANDARE
PK id_recomandare NOT NULL
filme_recomandate
seriale_recomandate
OFERĂ
PK id_oferă NOT NULL
FK1 id_profilAbonati
FK2 id_recomandare

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

+

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

--Primul tabel

```
CREATE TABLE NETFLIX (

id_netflix NUMBER(5) CONSTRAINT PK_NETFLIX PRIMARY KEY,

data_aparitie DATE NOT NULL,

pret_abonament INTEGER

);


SELECT * FROM NETFLIX;


INSERT INTO NETFLIX VALUES(1, to_date('29-08-1997','dd-mm-yyyy'), 50);

INSERT INTO NETFLIX VALUES(2, to_date('29-08-1997','dd-mm-yyyy'), 50);

INSERT INTO NETFLIX VALUES(3, to_date('29-08-1997','dd-mm-yyyy'), 50);

INSERT INTO NETFLIX VALUES(4, to_date('29-08-1997','dd-mm-yyyy'), 50);

INSERT INTO NETFLIX VALUES(5, to_date('29-08-1997','dd-mm-yyyy'), 50);
```

Query Result

All Rows Fetched: 5 in 0,168 seconds

| | ID_NETFLIX | DATA_APARITIE | PRET_ABONAMENT |
|---|---|---|---|
| 1 | 1 | 29-08-1997 | 50 |
| 2 | 2 | 29-08-1997 | 50 |
| 3 | 3 | 29-08-1997 | 50 |
| 4 | 4 | 29-08-1997 | 50 |
| 5 | 5 | 29-08-1997 | 50 |

--Al doilea tabel

```
CREATE TABLE SERIALE (

id_seriale NUMBER(5) CONSTRAINT PK_SERIALE PRIMARY KEY,

titlul VARCHAR(100),

genul VARCHAR(100),
```

```
nrSezoane INTEGER,
limitaVarsta INTEGER,
an_aparitie INTEGER,
id_netflix NUMBER(5),
CONSTRAINT FK_SERIALE_NETFLIX FOREIGN KEY(id_netflix) REFERENCES NETFLIX(id_netflix)
);

SELECT * FROM SERIALE;

INSERT INTO SERIALE VALUES (1, 'Lost', 'Drama', 3, 18, 2015, 1);
INSERT INTO SERIALE VALUES (2, 'Emily in Paris', 'Romantic', 2, 12, 2018, 1);
INSERT INTO SERIALE VALUES (3, 'Hunger Games', 'SF', 4, 21, 2020, 1);
INSERT INTO SERIALE VALUES (4, 'Papusa', 'Horror', 2, 21, 2008, 1);
INSERT INTO SERIALE VALUES (5, 'Pandemia', 'Documentar', 1, 16, 2020, 1);
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0,004 seconds

| | ID_SERIALE | TITLUL | GENUL | NRSEZOANE | LIMITAVARSTA | AN_APARITIE | ID_NETFLIX |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Lost | Drama | 3 | 18 | 2015 | 1 |
| 2 | 2 | Emily in Paris | Romantic | 2 | 12 | 2018 | 2 |
| 3 | 3 | Hunger Games | SF | 4 | 21 | 2020 | 3 |
| 4 | 4 | Papusa | Horror | 2 | 21 | 2008 | 4 |
| 5 | 5 | Pandemia | Documentar | 1 | 16 | 2020 | 5 |

```
--Al treilea tabel

CREATE TABLE FILME (
id_filme NUMBER(5) CONSTRAINT PK_FILME PRIMARY KEY,
titlul_film VARCHAR(100),
gen_film VARCHAR(100),
an_aparitie INTEGER,
durata INTEGER,
actor_principal VARCHAR(100),
```

id_netflix NUMBER(5),

CONSTRAINT FK_FILME_NETFLIX FOREIGN KEY(id_netflix) REFERENCES NETFLIX(id_netflix)

);

SELECT * FROM FILME;

INSERT INTO FILME VALUES (1, 'Friends', 'Comedie', 2000, 130, 'Brad Pitt', 1);

INSERT INTO FILME VALUES (2, 'The notebook', 'Romantic', 2006, 90, 'Angelina Jolie', 1);

INSERT INTO FILME VALUES (3, 'The story of my life', 'Documentar', 2019, 100, 'Cristiano Ronaldo', 1);

INSERT INTO FILME VALUES (4, 'Ana', 'Horror', 1999, 120, 'Vin Diesel', 1);

INSERT INTO FILME VALUES (5, 'The letter', 'Romantic', 2018, 97, 'Nina Dobrev', 1);

Script Output × Query Result ×

SQL | All Rows Fetched: 6 in 0,002 seconds

| | ID_FILME | TITLUL_FILM | GEN_FILM | AN_APARITIE | DURATA | ACTOR_PRINCIPAL | ID_NETFLIX |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Friends | Comedie | 2000 | 130 | Brad Pitt | 1 |
| 2 | 2 | The notebook | Romantic | 2006 | 90 | Angelina Jolie | 2 |
| 3 | 3 | The story of my life | Documentar | 2019 | 100 | Cristiano Ronaldo | 3 |
| 4 | 4 | Ana | Horror | 1999 | 120 | Vin Diesel | 4 |
| 5 | 5 | The letter | Romantic | 2018 | 97 | Nina Dobrev | 5 |
| 6 | 6 | The script | SF | 2008 | 98 | Ian Somerhalder | 5 |

--Al patrulea tabel

CREATE TABLE PRODUCATORI(

id_producatori NUMBER(5) CONSTRAINT PK_PRODUCATORI PRIMARY KEY,

nume VARCHAR(100),

prenume VARCHAR(100),

data_nasterii DATE NOT NULL,

locul_nasterii VARCHAR(100),

numar_telefon VARCHAR(50),

id_netflix NUMBER(5),

CONSTRAINT FK_PRODUCATORI_NETFLIX FOREIGN KEY(id_netflix) REFERENCES NETFLIX(id_netflix),

id_profilproducatori NUMBER(5),

CONSTRAINT FK_PRODUCATORI_PROFIL_PROD

FOREIGN KEY(id_profilproducatori) REFERENCES PROFIL_PRODUCATORI(id_profilproducatori)

);

INSERT INTO PRODUCATORI VALUES (1, 'Popa', 'Vasile', to_date('19-06-1987', 'dd-mm-yyyy'),'Bucuresti','0745000999', 1, 1);

INSERT INTO PRODUCATORI VALUES (2, 'Condrache', 'Maria', to_date('18-05-1967', 'dd-mm-yyyy'), 'Cluj-Napoca','0733222111', 1, 2);

INSERT INTO PRODUCATORI VALUES (3, 'Popa', 'Ana', to_date('09-02-1999', 'dd-mm-yyyy'),'Sibiu' ,'0987666555', 1, 3);

INSERT INTO PRODUCATORI VALUES (4, 'Enescu', 'Mihai', to_date('10-09-2000', 'dd-mm-yyyy'),'Timisoara', '0744321123', 1, 4);

INSERT INTO PRODUCATORI VALUES (5, 'Tudor', 'Andrei', to_date('14-07-1996', 'dd-mm-yyyy'),'Oradea' ,'0766897456', 1, 5);

SELECT * FROM PRODUCATORI;

Query Result

SQL | All Rows Fetched: 5 in 0,005 seconds

| | ID_PRODUCATORI | NUME | PRENUME | DATA_NASTERII | LOCUL_NASTERII | NUMAR_TELEFON | ID_NETFLIX | ID_PROFILPRODUCATORI |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Popa | Vasile | 19-06-1987 | Bucuresti | 0745000999 | 1 | 1 |
| 2 | 2 | Condrache | Maria | 18-05-1967 | Cluj-Napoca | 0733222111 | 1 | 2 |
| 3 | 3 | Popa | Ana | 09-02-1999 | Sibiu | 0987666555 | 1 | 3 |
| 4 | 4 | Enescu | Mihai | 10-09-2000 | Timisoara | 0744321123 | 1 | 4 |
| 5 | 5 | Tudor | Andrei | 14-07-1996 | Oradea | 0766897456 | 1 | 5 |

--Al cincilea tabel

CREATE TABLE PROFIL_PRODUCATORI (

id_profilproducatori NUMBER(5) CONSTRAINT PK_PROFIL_PRODUCATORI PRIMARY KEY,

data_creare DATE NOT NULL,

nr_filme_publicate INTEGER

);

INSERT INTO PROFIL_PRODUCATORI VALUES(1, to_date('19-06-2002', 'dd-mm-yyyy'), 4);

INSERT INTO PROFIL_PRODUCATORI VALUES(2, to_date('18-05-1987', 'dd-mm-yyyy'), 20);

INSERT INTO PROFIL_PRODUCATORI VALUES(3, to_date('24-04-2020', 'dd-mm-yyyy'), 6);

INSERT INTO PROFIL_PRODUCATORI VALUES(4, to_date('11-11-2021', 'dd-mm-yyyy'), 2);

INSERT INTO PROFIL_PRODUCATORI VALUES(5, to_date('12-12-2019', 'dd-mm-yyyy'), 10);

SELECT * FROM PROFIL_PRODUCATORI;

Query Result

All Rows Fetched: 5 in 0,004 seconds

| | ID_PROFILPRODUCATORI | DATA_CREARE | NR_FILME_PUBLICATE |
|---|---|---|---|
| 1 | 1 | 19-06-2002 | 4 |
| 2 | 2 | 18-05-1987 | 20 |
| 3 | 3 | 24-04-2020 | 6 |
| 4 | 4 | 11-11-2021 | 2 |
| 5 | 5 | 12-12-2019 | 10 |

--Al saselea tabel

CREATE TABLE ABONATI(

id_abonati NUMBER(5) CONSTRAINT PK_ABONATI PRIMARY KEY,

nume VARCHAR(100),

prenume VARCHAR(100),

data_nasterii DATE NOT NULL,

email VARCHAR(100),

id_netflix NUMBER(5),

CONSTRAINT FK_ABONATI_NETFLIX FOREIGN KEY(id_netflix) REFERENCES NETFLIX(id_netflix),

id_cont NUMBER(5),

CONSTRAINT FK_ABONATI_CONT FOREIGN KEY(id_cont) REFERENCES CONT (id_cont)

);

INSERT INTO ABONATI VALUES(1, 'Radu', 'Maria',to_date('12-09-2002','dd-mm-yyyy'),'radu.maria@yahoo.com',1,1);

INSERT INTO ABONATI VALUES(2, 'Bica', 'Ioana',to_date('09-10-2002','dd-mm-yyyy'),'bica.ioana@yahoo.com',1,2);

INSERT INTO ABONATI VALUES(3, 'Andrei', 'Monalisa',to_date('23-06-2000','dd-mm-yyyy'),'andrei.monalisa@yahoo.com',1,3);

INSERT INTO ABONATI VALUES(4, 'Pohrib', 'Andreea',to_date('10-08-2001','dd-mm-yyyy'),'pohrib.andreea@yahoo.com',1,4);

INSERT INTO ABONATI VALUES(5, 'Filimon', 'Raluca',to_date('07-07-1999','dd-mm-yyyy'),'filimon.raluca@yahoo.com',1,5);

SELECT * FROM ABONATI;

Script Output × Query Result ×

SQL | All Rows Fetched: 6 in 0,002 seconds

| | ID_ABONATI | NUME | PRENUME | DATA_NASTERII | EMAIL | ID_NETFLIX | ID_CONT |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Radu | Maria | 12-09-2002 | radu.maria@yahoo.com | 1 | 1 |
| 2 | 2 | Bica | Ioana | 09-10-2002 | bica.ioana@yahoo.com | 2 | 2 |
| 3 | 3 | Andrei | Monalisa | 23-06-2000 | andrei.monalisa@yahoo.com | 3 | 3 |
| 4 | 4 | Pohrib | Andreea | 10-08-2001 | pohrib.andreea@yahoo.com | 4 | 4 |
| 5 | 5 | Filimon | Raluca | 07-07-1999 | filimon.raluca@yahoo.com | 5 | 5 |
| 6 | 6 | Bica | Maria | 12-09-2002 | radu.maria@yahoo.com | 1 | 6 |

--Al saptelea tabel

CREATE TABLE CONT(

id_cont NUMBER(5) CONSTRAINT PK_CONT PRIMARY KEY,

data_deschidere DATE NOT NULL,

iban VARCHAR(100),

sold INTEGER

);

INSERT INTO CONT VALUES(1, to_date('12-09-2020','dd-mm-yyyy'), '5234.0000.1111.2222', 250);

INSERT INTO CONT VALUES(2, to_date('11-11-2021','dd-mm-yyyy'), '5234.0000.8888.9999', 150);

INSERT INTO CONT VALUES(3, to_date('12-12-2019','dd-mm-yyyy'), '1234.0000.5678.9123', 100);

INSERT INTO CONT VALUES(4, to_date('06-06-2018','dd-mm-yyyy'), '4444.0000.6666.7777', 50);

INSERT INTO CONT VALUES(5, to_date('07-07-2022','dd-mm-yyyy'), '1111.0000.2222.3333', 340);

SELECT * FROM CONT;

Script Output × Query Result ×

SQL | All Rows Fetched: 6 in 0,003 seconds

| | ID_CONT | DATA_DESCHIDERE | IBAN | SOLD |
|---|---|---|---|---|
| 1 | 1 | 12-09-2020 | 5234.0000.1111.2222 | 250 |
| 2 | 2 | 11-11-2021 | 5234.0000.8888.9999 | 150 |
| 3 | 3 | 12-12-2019 | 1234.0000.5678.9123 | 100 |
| 4 | 4 | 06-06-2018 | 4444.0000.6666.7777 | 50 |
| 5 | 5 | 07-07-2022 | 1111.0000.2222.3333 | 340 |
| 6 | 6 | 10-07-2022 | 1111.2222.3333.4444 | 270 |

--Al optelea tabel

CREATE TABLE PROFIL_ABONATI(

id_profilabonati NUMBER(5) CONSTRAINT PK_PROFIL_ABONATI PRIMARY KEY,

tip_profil VARCHAR(100),

id_cont NUMBER(5),

CONSTRAINT FK_PROFIL_ABONATI_CONT FOREIGN KEY (id_cont) REFERENCES CONT (id_cont)

);

INSERT INTO PROFIL_ABONATI VALUES (1, 'Adult', 1);

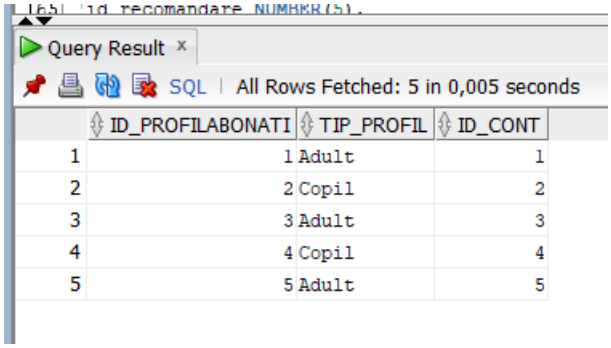
INSERT INTO PROFIL_ABONATI VALUES (2, 'Copil', 2);

INSERT INTO PROFIL_ABONATI VALUES (3, 'Adult', 3);

INSERT INTO PROFIL_ABONATI VALUES (4, 'Copil', 4);

INSERT INTO PROFIL_ABONATI VALUES (5, 'Adult', 5);

SELECT * FROM PROFIL_ABONATI;



Query Result

SQL | All Rows Fetched: 5 in 0,005 seconds

| | ID_PROFILABONATI | TIP_PROFIL | ID_CONT |
|---|---|---|---|
| 1 | 1 | Adult | 1 |
| 2 | 2 | Copil | 2 |
| 3 | 3 | Adult | 3 |
| 4 | 4 | Copil | 4 |
| 5 | 5 | Adult | 5 |

--Al noulea tabel

CREATE TABLE OFERA (

id_ofera NUMBER(5) CONSTRAINT PK_OFERA PRIMARY KEY,

id_profilabonati NUMBER(5),

CONSTRAINT FK_OFERA_PROFIL_ABO FOREIGN KEY (id_profilabonati)
REFERENCES PROFIL_ABONATI (id_profilabonati),

id_recomandare NUMBER(5),

CONSTRAINT FK_OFERA_RECOMANDARE FOREIGN KEY (id_recomandare)
REFERENCES RECOMANDARE (id_recomandare)

);

INSERT INTO OFERA VALUES (1, 1, 1);

INSERT INTO OFERA VALUES (2, 1, 3);

INSERT INTO OFERA VALUES (3, 1, 5);

INSERT INTO OFERA VALUES (4, 2, 2);

INSERT INTO OFERA VALUES (5, 2, 4);

INSERT INTO OFERA VALUES (6, 3, 1);

INSERT INTO OFERA VALUES (7, 3, 3);

INSERT INTO OFERA VALUES (8, 3, 5);

INSERT INTO OFERA VALUES (9, 4, 2);

INSERT INTO OFERA VALUES (10, 4, 4);

SELECT * FROM OFERA;

Query Result

SQL | All Rows Fetched: 10 in 0,004 seconds

| | ID_OFERA | ID_PROFILABONATI | ID_RECOMANDARE |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 3 |
| 3 | 3 | 1 | 5 |
| 4 | 4 | 2 | 2 |
| 5 | 5 | 2 | 4 |
| 6 | 6 | 3 | 1 |
| 7 | 7 | 3 | 3 |
| 8 | 8 | 3 | 5 |
| 9 | 9 | 4 | 2 |
| 10 | 10 | 4 | 4 |

-- Al zecelea tabel

```
CREATE TABLE RECOMANDARE (
id_recomandare NUMBER(5) CONSTRAINT PK_RECOMANDARE PRIMARY KEY,
filme_recomandate VARCHAR(100),
seriale_recomandate VARCHAR(100)
);

INSERT INTO RECOMANDARE VALUES (1, 'Love', 'Spartacus');
INSERT INTO RECOMANDARE VALUES (2, 'Intalnire cu un star', 'Spider-man');
INSERT INTO RECOMANDARE VALUES (3, 'Amicii', 'Vikingii');
INSERT INTO RECOMANDARE VALUES (4, 'Battman', 'Violetta');
INSERT INTO RECOMANDARE VALUES (5, 'Coronavirus', 'Lost');

SELECT * FROM RECOMANDARE;
```

Query Result

SQL | All Rows Fetched: 5 in 0,005 seconds

| | ID_RECOMANDARE | FILME_RECOMANDATE | SERIALE_RECOMANDATE |
|---|---|---|---|
| 1 | 1 | Love | Spartacus |
| 2 | 2 | Intalnire cu un star | Spider-man |
| 3 | 3 | Amicii | Vikingii |
| 4 | 4 | Battman | Violetta |
| 5 | 5 | Coronavirus | Lost |

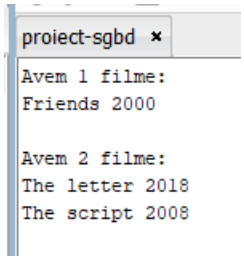
6. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze doua tipuri diferite de colectii studiate. Apelati subprogramul.

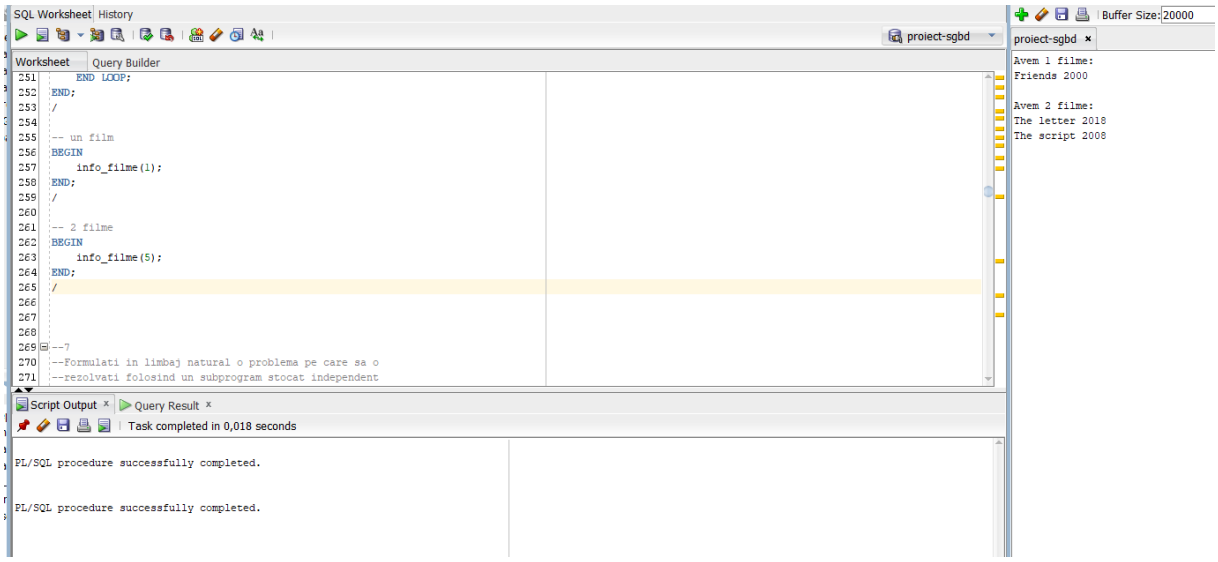
--Pentru un id de netflix introdus sa se afiseze toate

--titlurile de filme si anul de aparitie ale acestora.

```
CREATE OR REPLACE PROCEDURE info_filme (v_netflix_id netflix.id_netflix%TYPE)
IS

-- vector in care am retinut numele de filme

TYPE vec IS VARRAY(15) OF VARCHAR2(25);

v_filme vec :=vec();

--un tablou indexat in care am retinut anul de aparitie

TYPE tablou_indexat is TABLE OF NUMBER INDEX BY BINARY_INTEGER;

v_aparitie tablou_indexat;

BEGIN

   SELECT titlul_film BULK COLLECT INTO v_filme

   FROM filme

   WHERE id_netflix = v_netflix_id;


   SELECT an_aparitie BULK COLLECT INTO v_aparitie

   FROM filme

   WHERE id_netflix = v_netflix_id;


   --le numaram si le afisam

   DBMS_OUTPUT.PUT_LINE ('Avem ' || v_filme.count || ' filme:');

   --ne plimbam prin ele si vedem filmele si anul de aparitie
```

```
    FOR i IN v_filme.FIRST..v_aparitie.LAST loop
        DBMS_OUTPUT.PUT_LINE (v_filme(i) || ' ' || v_aparitie(i));
    END LOOP;
END;
/

-- un film
BEGIN
    info_filme(1);
END;
/

-- 2 filme
BEGIN
    info_filme(5);
END;
/
```






7. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelati subprogramul.

--Pentru fiecare abonat sa se afiseze filmele recomandate pentru

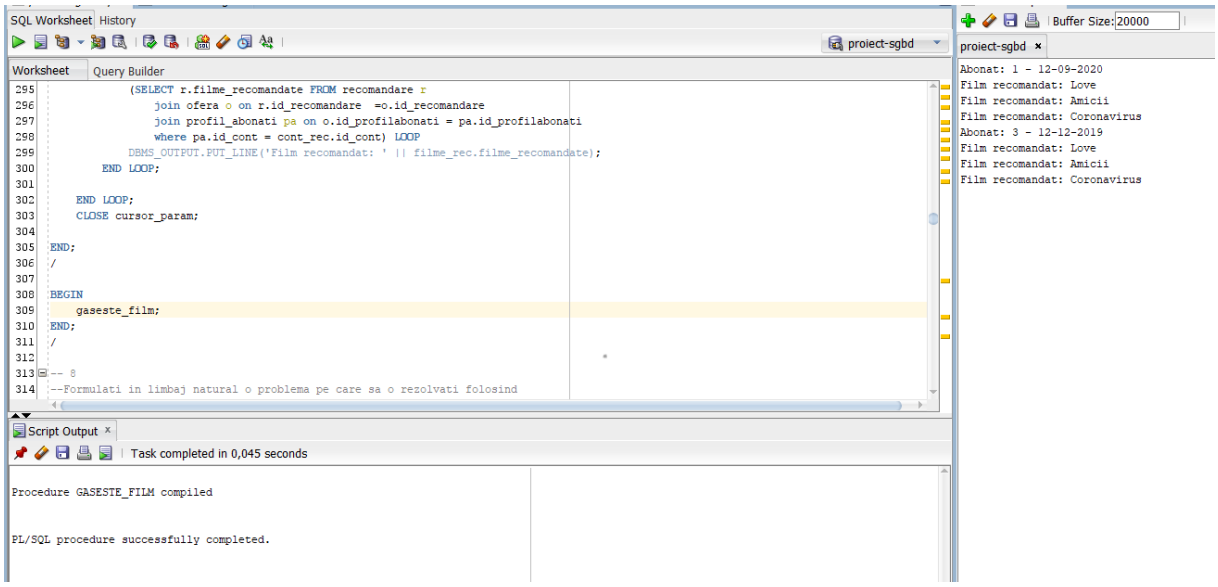
--cei care si-au deschis contul intre 2019 si 2021.

```
select a.nume ||' '|| a.prenume, r.filme_recomandate, data_deschidere from recomandare r
join ofera o on o.id_recomandare = r.id_recomandare
join profil_abonati pa on o.id_profilabonati = pa.id_profilabonati
join cont c on c.id_cont = pa.id_profilabonati
join abonati a on a.id_cont = c.id_cont
where EXTRACT(YEAR FROM c.data_deschidere) BETWEEN 2019 AND 2020;
```

```
CREATE OR REPLACE PROCEDURE gaseste_film IS cont_rec CONT%ROWTYPE;
filme_rec recomandare.filme_recomandate%TYPE;
CURSOR cursor_param (min_year DATE, max_year DATE) IS
    SELECT c.id_cont, c.data_deschidere FROM CONT c
    WHERE c.data_deschidere BETWEEN min_year AND max_year;
```

```
BEGIN
   OPEN cursor_param (to_date('01-01-2019', 'dd-mm-yyyy'),
   to_date('31-12-2020', 'dd-mm-yyyy'));
   LOOP
   FETCH cursor_param into cont_rec.id_cont, cont_rec.data_deschidere;
   EXIT WHEN cursor_param%notfound;

   DBMS_OUTPUT.PUT_LINE('Abonat: ' || cont_rec.id_cont || ' - ' ||
cont_rec.data_deschidere);

      FOR filme_rec IN
         (SELECT r.filme_recomandate FROM recomandare r
            join ofera o on r.id_recomandare  =o.id_recomandare
            join profil_abonati pa on o.id_profilabonati = pa.id_profilabonati
            where pa.id_cont = cont_rec.id_cont) LOOP
         DBMS_OUTPUT.PUT_LINE('Film recomandat: ' || filme_rec.filme_recomandate);
      END LOOP;

   END LOOP;
   CLOSE cursor_param;

END;
/

BEGIN
   gaseste_film;
END;
/
```

```
Abonat: 1 - 12-09-2020
Film recomandat: Love
Film recomandat: Amicii
Film recomandat: Coronavirus
Abonat: 3 - 12-12-2019
Film recomandat: Love
Film recomandat: Amicii
Film recomandat: Coronavirus
```




8. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 dintre tabelele definite. Definiti minim 2 exceptii. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

-- Folosindu-va de TABELA NETFLIX, ABONATI SI FILME,

-- sa se afiseze daca durata filmului ales este potrivita

-- pentru a se uita la un film abonatul, deoarece are un interval de

-- timp in care se poate uita. Daca depaseste acest interval

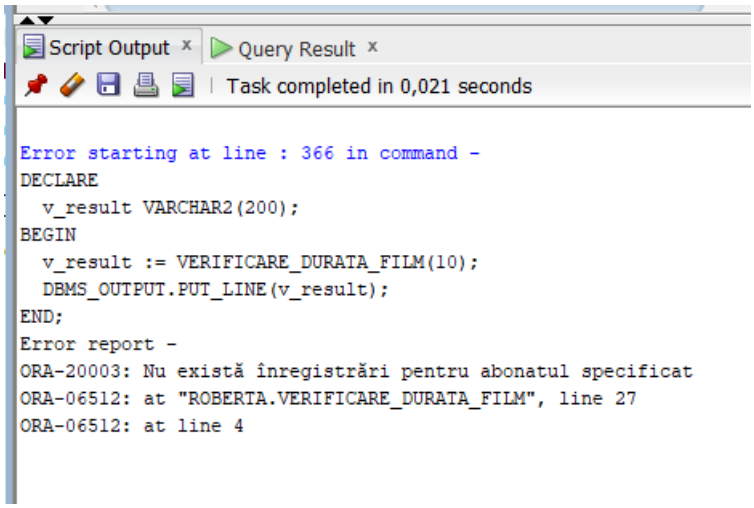
-- primeste o eroare. De asemenea el vrea sa profite de timpul liber

-- asa ca daca durata este prea mica la fel -> primeste o eroare.

```
CREATE OR REPLACE FUNCTION VERIFICARE_DURATA_FILM(
  abonat_id IN NUMBER
) RETURN VARCHAR2 AS
  v_durata INTEGER;
  v_titlu_film VARCHAR2(100);
```

```
  v_nume_abonat VARCHAR2(100);
  v_result VARCHAR2(200);
BEGIN
  SELECT f.durata, f.titlul_film, a.nume
  INTO v_durata, v_titlu_film, v_nume_abonat
  FROM ABONATI a
  JOIN FILME f ON a.id_netflix = f.id_netflix
  JOIN NETFLIX n ON a.id_netflix = n.id_netflix
  WHERE a.id_abonati = abonat_id;

  IF v_durata > 121 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Durata filmului ''' || v_titlu_film || ''' este prea
mare pentru abonatul ''' || v_nume_abonat || ''');
  ELSIF v_durata < 91 THEN
    RAISE_APPLICATION_ERROR(-20002, 'Durata filmului ''' || v_titlu_film || ''' este prea
mică pentru abonatul ''' || v_nume_abonat || ''');
  ELSE
    v_result := 'Durata filmului ''' || v_titlu_film || ''' este în intervalul dorit pentru abonatul ''' ||
v_nume_abonat || '''';
  END IF;

  RETURN v_result;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20003, 'Nu există înregistrări pentru abonatul
specificat');
END;
/


DECLARE
  v_result VARCHAR2(200);
```

```
BEGIN
  v_result := VERIFICARE_DURATA_FILM(10);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```




```
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := VERIFICARE_DURATA_FILM(1);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
Error starting at line : 374 in command -
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := VERIFICARE_DURATA_FILM(1);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
Error report -
ORA-20001: Durata filmului "Friends" este prea mare pentru abonatul "Radu"
ORA-06512: at "ROBERTA.VERIFICARE_DURATA_FILM", line 17
ORA-06512: at line 4
```

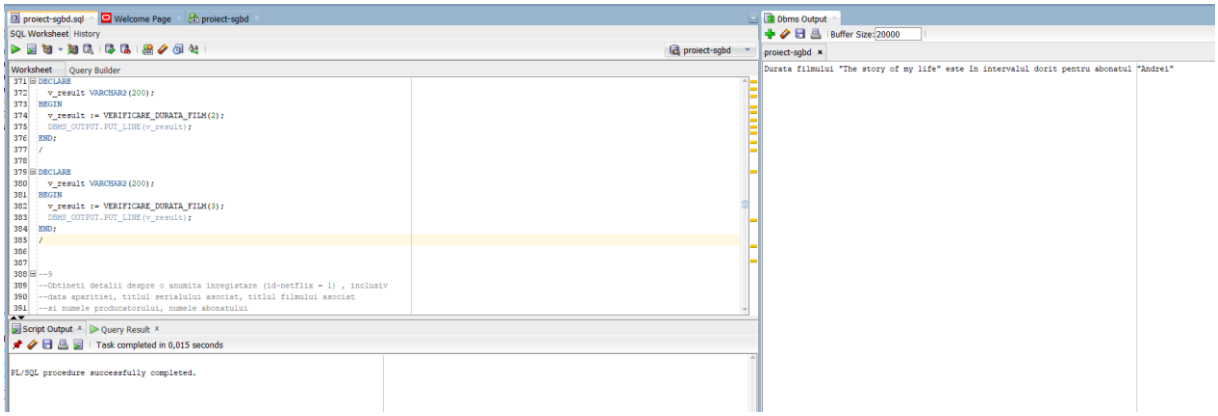
```
DECLARE
  v_result VARCHAR2(200);
BEGIN
```

```
  v_result := VERIFICARE_DURATA_FILM(2);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
Error starting at line : 382 in command -
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := VERIFICARE_DURATA_FILM(2);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
Error report -
ORA-20002: Durata filmului "The notebook" este prea mică pentru abonatul "Bica"
ORA-06512: at "ROBERTA.VERIFICARE_DURATA_FILM", line 19
ORA-06512: at line 4
```

```
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := VERIFICARE_DURATA_FILM(3);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
proiect-sgbd
Durata filmului "The story of my life" este în intervalul dorit pentru abonatul "Andrei"
```

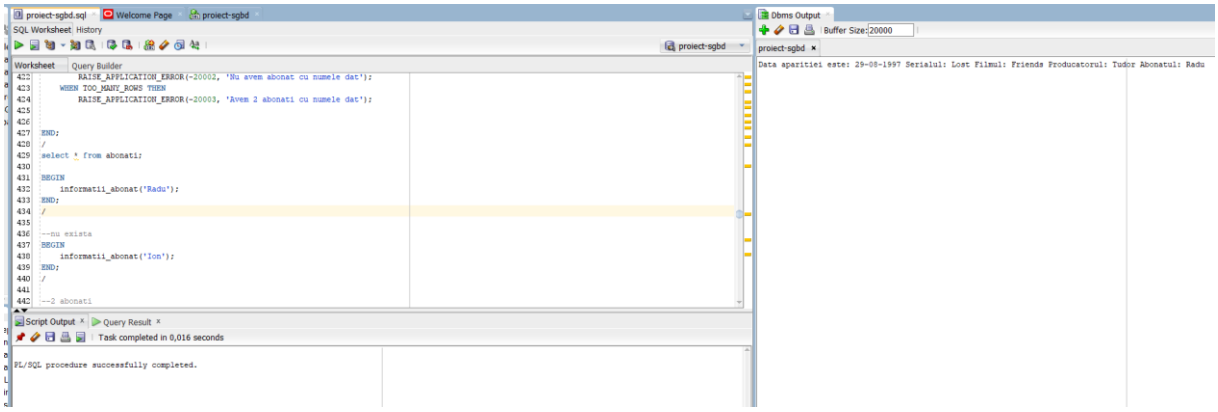



9. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa utilizeze intr-o singura comanda SQL 5 dintre tabelele definite. Tratati toate exceptiile care pot aparea, incluzand exceptiile NO_DATA_FOUND si TOO_MANY_ROWS. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

```
--Obtineti detalii despre o anumita inregistare (id-netflix = 1) , inclusiv
--data aparitiei, titlul serialului asociat, titlul filmului asociat
--si numele producatorului, numele abonatului
--Tabele folosite NETFLIX, SERIALE, FILME, PRODUCATORI, ABONATI

CREATE OR REPLACE PROCEDURE informatii_abonat ( nume_abonat_param
abonati.nume%TYPE)
IS
v_data_aparitie netflix.data_aparitie%TYPE;
v_titlul_serial seriale.titlul%TYPE;
v_titlul_film filme.titlul_film%type;
nume_producator producatori.nume%TYPE;
v_id_abonat abonati.id_abonati%TYPE;

BEGIN
 select id_abonati into v_id_abonat from abonati where nume = nume_abonat_param ;

 SELECT netflix.data_aparitie, seriale.titlul, filme.titlul_film, producatori.nume
 INTO v_data_aparitie, v_titlul_serial, v_titlul_film, nume_producator
 FROM NETFLIX
 JOIN SERIALE ON seriale.id_netflix = netflix.id_netflix
 JOIN FILME ON filme.id_netflix = netflix.id_netflix
 JOIN PRODUCATORI ON producatori.id_netflix = netflix.id_netflix
 JOIN ABONATI ON abonati.id_netflix = netflix.id_netflix
 WHERE id_abonati = v_id_abonat
 and rownum = 1;

 DBMS_OUTPUT.PUT_LINE ('Data aparitiei este: ' || v_data_aparitie || ' Serialul: '
   || v_titlul_serial || ' Filmul: ' || v_titlul_film
   ||' Producatorul: ' || nume_producator || ' Abonatul: ' || nume_abonat_param);
```

```
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20002, 'Nu avem abonat cu numele dat');
   WHEN TOO_MANY_ROWS THEN
      RAISE_APPLICATION_ERROR(-20003, 'Avem 2 abonati cu numele dat');



END;
/



BEGIN
   informatii_abonat('Radu');
END;
/
```

```
Data aparitiei este: 29-08-1997 Serialul: Lost Filmul: Friends Producatorul: Tudor Abonatul: Radu
```




```
--nu exista
BEGIN
   informatii_abonat('Ion');
END;
/
```

```
Error starting at line : 410 in command -
BEGIN
    informatii_abonat('Ion');
END;
Error report -
ORA-20002: Nu avem abonat cu numele dat
ORA-06512: at "ROBERTA.INFORMATII_ABONAT", line 29
ORA-06512: at line 2
```

--2 abonati

BEGIN

informatii_abonat('Bica');

END;

/

```
Error starting at line : 454 in command -
BEGIN
    informatii_abonat('Bica');
END;
Error report -
ORA-20003: Avem 2 abonati cu numele dat
ORA-06512: at "ROBERTA.INFORMATII_ABONAT", line 31
ORA-06512: at line 2
```

10. Definiti un trigger de tip LMD la nivel de comanda. Declansati trigger-ul.

-- TRIGGER CARE INTERZICE STERGEREA DATELOR DIN TABELUL PRODUCATOR

CREATE OR REPLACE TRIGGER trigger10

BEFORE INSERT OR UPDATE OR DELETE ON PRODUCATORI

BEGIN

RAISE_APPLICATION_ERROR(-20001, 'Nu se pot sterge date din tabelul PRODUCATOR');

END;

/

DELETE FROM PRODUCATORI WHERE id_producatori=1;

```
Error starting at line : 430 in command -
DELETE FROM PRODUCATORI WHERE id_producatori=1
Error report -
ORA-20001: Nu se pot sterge date din tabelul PRODUCATOR
ORA-06512: at "ROBERTA.TRIGGER10", line 2
ORA-04088: error during execution of trigger 'ROBERTA.TRIGGER10'
```

DROP TRIGGER trigger10;

11. Definiti un trigger de tip LMD la nivel de linie. Declansati trigger-ul.

```
--TRIGGER CARE INTERZICE MICSORAREA PRETULUI LA ABONAMENT
CREATE OR REPLACE TRIGGER trigger11
  BEFORE UPDATE ON NETFLIX
  FOR EACH ROW
BEGIN
  IF (:NEW.pret_abonament < :OLD.pret_abonament )
  THEN RAISE_APPLICATION_ERROR(-20002, 'Pretul nu poate sa se modifice');
  END IF;
END;
/
UPDATE NETFLIX
SET pret_abonament= pret_abonament - 71
where id_netflix=1;
```

```
Error starting at line : 445 in command -
UPDATE NETFLIX
SET pret_abonament= pret_abonament - 71
where id_netflix=1
Error report -
ORA-20002: Pretul nu poate sa se modifice
ORA-06512: at "ROBERTA.TRIGGER11", line 3
ORA-04088: error during execution of trigger 'ROBERTA.TRIGGER11'
```

```
DROP TRIGGER trigger11;
```

12. Definiti un trigger de tip LDD. Declansati trigger-ul.

```
--12
-- TRIGGER CARE AFISEAZA UN MESAJ DE FIECARE DATA CAND ESTE RULATA O COMANDA LDD
CREATE TABLE exercitiul12
(utilizator VARCHAR2(30),
nume_bd VARCHAR2(50),
eveniment VARCHAR2(20),
nume_obiect VARCHAR2(30),
data DATE);
```

```
CREATE OR REPLACE TRIGGER trigger12
  AFTER CREATE OR ALTER OR DROP ON SCHEMA
DECLARE
BEGIN
  INSERT INTO exercitiul12 VALUES (
    SYS.LOGIN_USER,
    SYS.DATABASE_NAME,
    SYS.SYSEVENT,
    SYS.DICTIONARY_OBJ_NAME,
    SYSDATE
  );
END;
/

ALTER TABLE ABONATI ADD varsta NUMBER(2);
ALTER TABLE ABONATI DROP COLUMN varsta;
CREATE INDEX ind ON ABONATI('ceva');
DROP INDEX ind;

select * from exercitiul12;

DROP TRIGGER trigger12;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 10 in 0,002 seconds

| | UTILIZATOR | NUME_BD | EVENIMENT | NUME_OBIECT | DATA |
|---|---|---|---|---|---|
| 1 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 2 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 3 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 4 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 5 | ROBERTA | XE | CREATE | IND | 19-05-2023 |
| 6 | ROBERTA | XE | DROP | IND | 19-05-2023 |
| 7 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 8 | ROBERTA | XE | ALTER | ABONATI | 19-05-2023 |
| 9 | ROBERTA | XE | CREATE | IND | 19-05-2023 |
| 10 | ROBERTA | XE | DROP | IND | 19-05-2023 |

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

```
-- 13

CREATE OR REPLACE PACKAGE ex13 AS
PROCEDURE info_filme (v_netflix_id netflix.id_netflix%TYPE);
PROCEDURE gaseste_film;
FUNCTION VERIFICARE_DURATA_FILM(abonat_id IN NUMBER)
   RETURN VARCHAR2;
PROCEDURE informatii_abonat ( nume_abonat_param abonati.nume%TYPE);
END ex13;
/

CREATE OR REPLACE PACKAGE BODY ex13 AS
   PROCEDURE info_filme (v_netflix_id netflix.id_netflix%TYPE) IS
-- vector in care am retinut numele de filme
TYPE vec IS VARRAY(15) OF VARCHAR2(25);
v_filme vec :=vec();
--un tablou indexat in care am retinut anul de aparitie
TYPE tablou_indexat is TABLE OF NUMBER INDEX BY BINARY_INTEGER;
v_aparitie tablou_indexat;
BEGIN
   SELECT titlul_film BULK COLLECT INTO v_filme
```

```
    FROM filme
    WHERE id_netflix = v_netflix_id;

    SELECT an_aparitie BULK COLLECT INTO v_aparitie
    FROM filme
    WHERE id_netflix = v_netflix_id;

    --le numaram si le afisam
    DBMS_OUTPUT.PUT_LINE ('Avem ' || v_filme.count || ' filme:');
    --ne plimbam prin ele si vedem filmele si anul de aparitie
    FOR i IN v_filme.FIRST..v_aparitie.LAST loop
        DBMS_OUTPUT.PUT_LINE (v_filme(i) || ' ' || v_aparitie(i));
    END LOOP;
END info_filme;

PROCEDURE gaseste_film IS v_filme_recom
recomandare%ROWTYPE;
CURSOR cursor_param (min_year DATE, max_year DATE) IS
    SELECT * FROM CONT
    WHERE data_deschidere BETWEEN min_year AND max_year;

BEGIN
    FOR cont_rec IN cursor_param (to_date('01-01-2019', 'dd-mm-yyyy'),
    to_date('31-12-2020', 'dd-mm-yyyy')) LOOP
    DBMS_OUTPUT.PUT_LINE('Abonat: ' || cont_rec.id_cont || ' - ' ||
cont_rec.data_deschidere);

    FOR filme_rec IN
        (SELECT r.filme_recomandate FROM recomandare r
            join ofera o on r.id_recomandare  =o.id_recomandare
             join profil_abonati pa on o.id_profilabonati = pa.id_profilabonati
```

```
            where pa.id_cont = cont_rec.id_cont) LOOP
    DBMS_OUTPUT.PUT_LINE('Film recomandat: ' || filme_rec.filme_recomandate);
    END LOOP;

    END LOOP;

    CLOSE CURSOR_PARAM;



END gaseste_film;

FUNCTION VERIFICARE_DURATA_FILM(
  abonat_id IN NUMBER
) RETURN VARCHAR2 AS
  v_durata INTEGER;
  v_titlu_film VARCHAR2(100);
  v_nume_abonat VARCHAR2(100);
  v_result VARCHAR2(200);
BEGIN
  SELECT f.durata, f.titlul_film, a.nume
  INTO v_durata, v_titlu_film, v_nume_abonat
  FROM ABONATI a
  JOIN FILME f ON a.id_netflix = f.id_netflix
  JOIN NETFLIX n ON a.id_netflix = n.id_netflix
  WHERE a.id_abonati = abonat_id;

  IF v_durata > 121 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Durata filmului ''' || v_titlu_film || ''' este prea
mare pentru abonatul ''' || v_nume_abonat || ''');
  ELSIF v_durata < 91 THEN
```

```
    RAISE_APPLICATION_ERROR(-20002, 'Durata filmului "' || v_titlu_film || '" este prea
mică pentru abonatul "' || v_nume_abonat || '"');

  ELSE

    v_result := 'Durata filmului "' || v_titlu_film || '" este în intervalul dorit pentru abonatul "' ||
v_nume_abonat || '";

  END IF;


  RETURN v_result;

EXCEPTION

  WHEN NO_DATA_FOUND THEN

    RAISE_APPLICATION_ERROR(-20003, 'Nu există înregistrări pentru abonatul
specificat');

END verificare_durata_film;


PROCEDURE informatii_abonat ( nume_abonat_param abonati.nume%TYPE)

IS

v_data_aparitie netflix.data_aparitie%TYPE;

v_titlul_serial seriale.titlul%TYPE;

v_titlul_film filme.titlul_film%type;

nume_producator producatori.nume%TYPE;

v_id_abonat abonati.id_abonati%TYPE;


BEGIN

 select id_abonati into v_id_abonat from abonati where nume = nume_abonat_param ;


 SELECT netflix.data_aparitie, seriale.titlul, filme.titlul_film, producatori.nume

 INTO v_data_aparitie, v_titlul_serial, v_titlul_film, nume_producator

 FROM NETFLIX

 JOIN SERIALE ON seriale.id_netflix = netflix.id_netflix

 JOIN FILME ON filme.id_netflix = netflix.id_netflix

 JOIN PRODUCATORI ON producatori.id_netflix = netflix.id_netflix

 JOIN ABONATI ON abonati.id_netflix = netflix.id_netflix
```

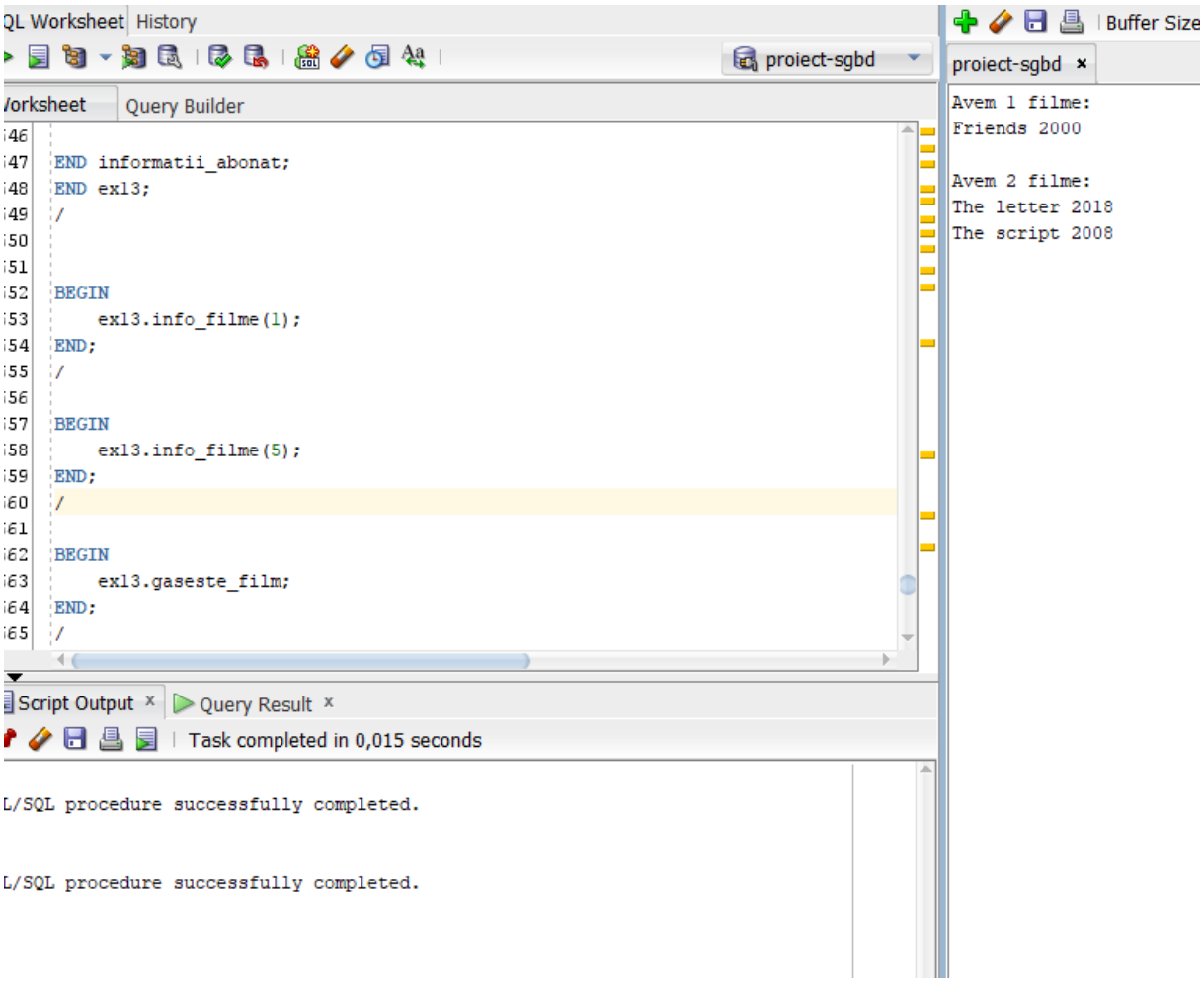
```
 WHERE id_abonati = v_id_abonat
 and rownum = 1;

 DBMS_OUTPUT.PUT_LINE ('Data aparitiei este: ' || v_data_aparitie || ' Serialul: '
   || v_titlul_serial || ' Filmul: ' || v_titlul_film
   ||' Producatorul: ' || nume_producator || ' Abonatul: ' || nume_abonat_param);


 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20002, 'Nu avem abonat cu numele dat');
   WHEN TOO_MANY_ROWS THEN
      RAISE_APPLICATION_ERROR(-20003, 'Avem 2 abonati cu numele dat');


END informatii_abonat;
END ex13;
/
```

- Un film

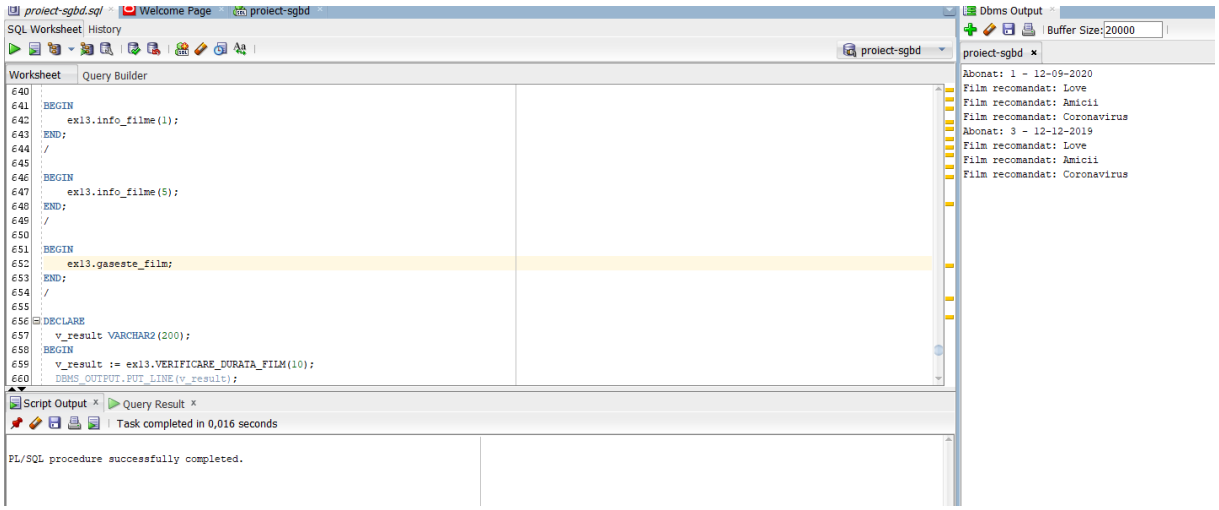
```
BEGIN
   ex13.info_filme(1);
END;
/
```

- 2 filme

```
BEGIN
   ex13.info_filme(5);
END;
/
```




```
BEGIN
    ex13.gaseste_film;
END;
/
```




- No data_found

```
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(10);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
Error starting at line : 667 in command -
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(10);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
Error report -
ORA-20003: Nu există înregistrări pentru abonatul specificat
ORA-06512: at "ROBERTA.EX13", line 78
ORA-06512: at line 4
```

- Prea mare -> eroare

```
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(1);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
Error starting at line : 675 in command -
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(1);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
Error report -
ORA-20001: Durata filmului "Friends" este prea mare pentru abonatul "Radu"
ORA-06512: at "ROBERTA.EX13", line 68
ORA-06512: at line 4
```

- Prea mica -> eroare

```
DECLARE
```

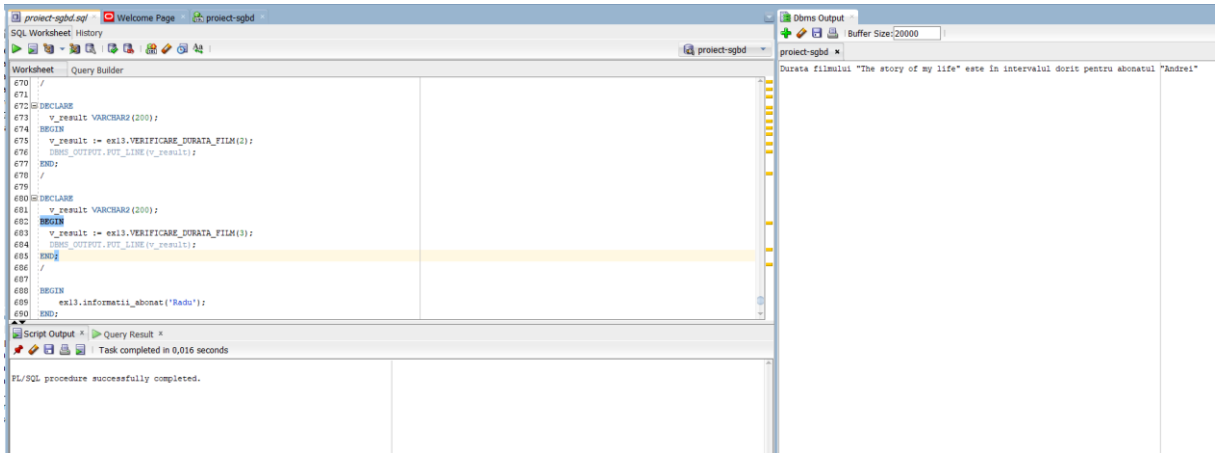
```
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(2);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

```
Error starting at line : 683 in command -
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(2);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
Error report -
ORA-20002: Durata filmului "The notebook" este prea mică pentru abonatul "Bica"
ORA-06512: at "ROBERTA.EX13", line 70
ORA-06512: at line 4
```

- merge

```
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := ex13.VERIFICARE_DURATA_FILM(3);
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
/
```

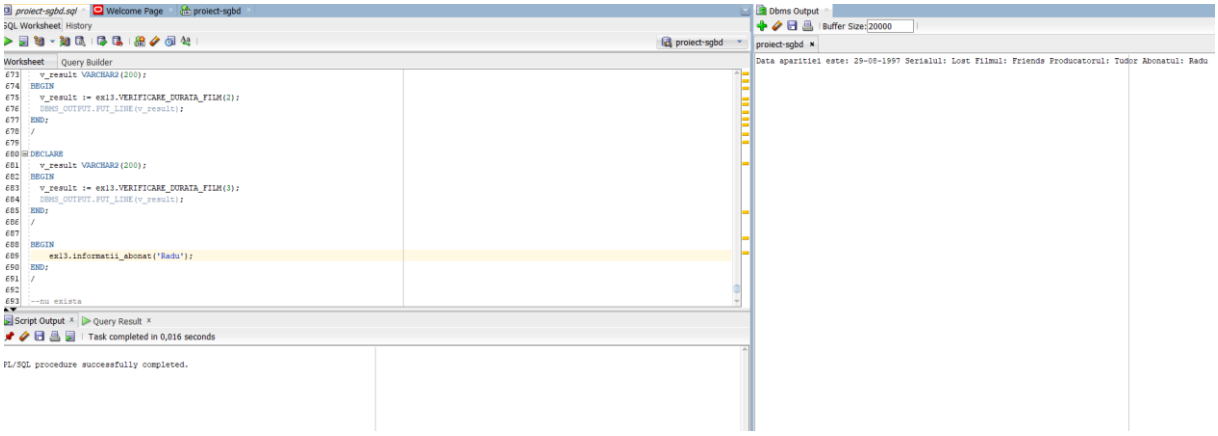
```
Durata filmului "The story of my life" este în intervalul dorit pentru abonatul "Andrei"
```



- merge

```
BEGIN
   ex13.informatii_abonat('Radu');
END;
/
```

```
Data aparitiei este: 29-08-1997 Serialul: Lost Filmul: Friends Producatorul: Tudor Abonatul: Radu
```




```
--nu exista
BEGIN
   ex13.informatii_abonat('Ion');
END;
/
```

```
--nu exista
BEGIN
    ex13.informatii_abonat('Ion');
END;
/
```

```
--2 abonati
BEGIN
   ex13.informatii_abonat('Bica');
END;
/
```

```
Error starting at line : 711 in command -
BEGIN
    ex13.informatii_abonat('Bica');
END;
Error report -
ORA-20003: Avem 2 abonati cu numele dat
ORA-06512: at "ROBERTA.EX13", line 111
ORA-06512: at line 2
```