Universitatea de Matematică și Informatică Antonia-Adelina Popovici
Informatică, anul 2, semestrul I

Proiect – SISTEME DE GESTIUNE A BAZELOR DE DATE

Sistem de gestiune a unui lanț de cinematografe

CUPRINS

1. Descrierea proiectului ales

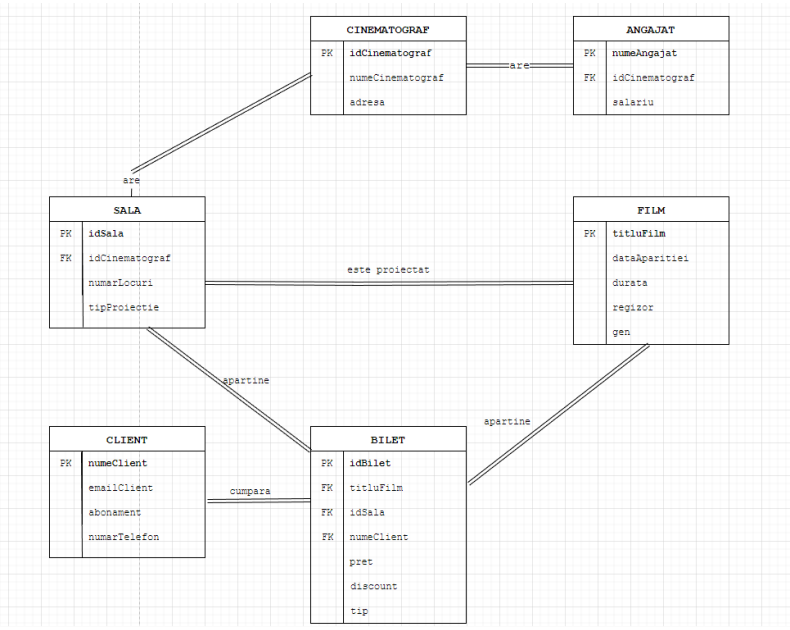
a. Prezentarea bazei de date

Proiectul presupune un sistem de gestionare a unui lanţ de cinematografe. Modelul cuprinde date despre filme difuzate, cinematografe, săli accesibile, bilete, clienţi, angajaţi (FILM, CINEMATOGRAF, SALA, BILET, CLIENT, ANGAJAT) . Pe lângă aceste entităţi independente, are şi un tabel asociativ – PROIECŢIE – între tabelul SALA şi tabelul FILM Aceste informaţii pot fi uşor găsite prin utilizarea unui motor de căutare în funcţie de diferite condiţii care face posibilă accesarea rapidă şi uşoară a datelor dorite.

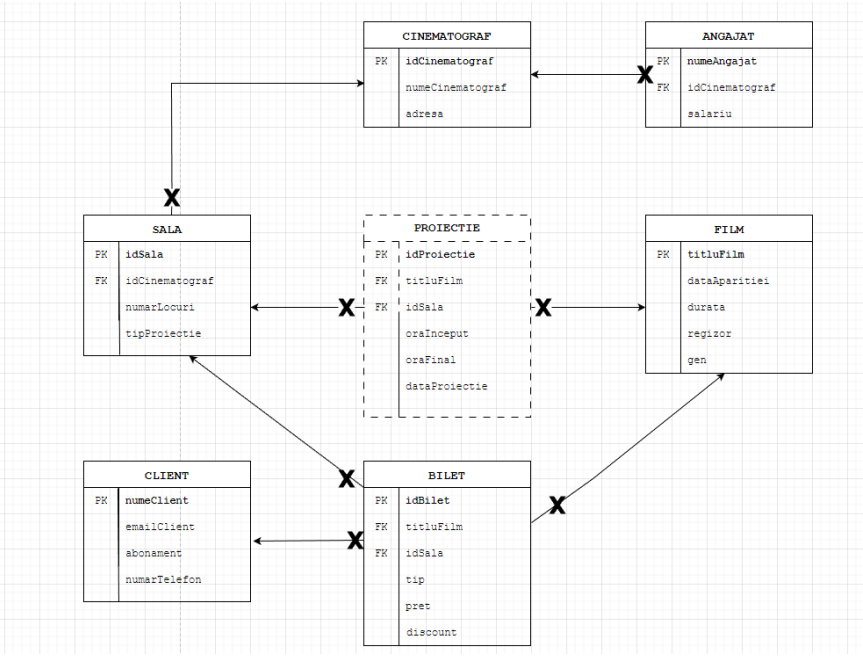
Sistemul poate fi astfel utilizat cu scopul de a gestiona atât filmele proiectate, precum şi unde şi când au loc.

Modelul este implementat în sistemul de gestiune a bazelor de date Oracle, iar interacţiunea cu baza de date a fost posibilă prin utilizarea programului SQL Developer.

b. Diagrama entitate-relaţie



c. Diagrama conceptuală

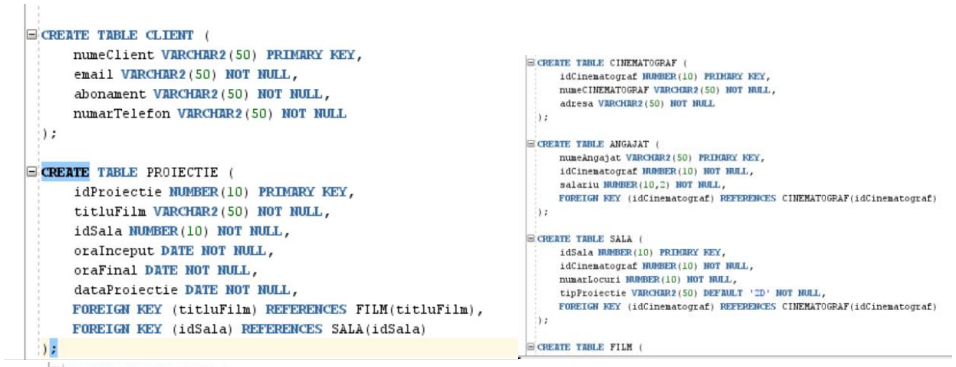


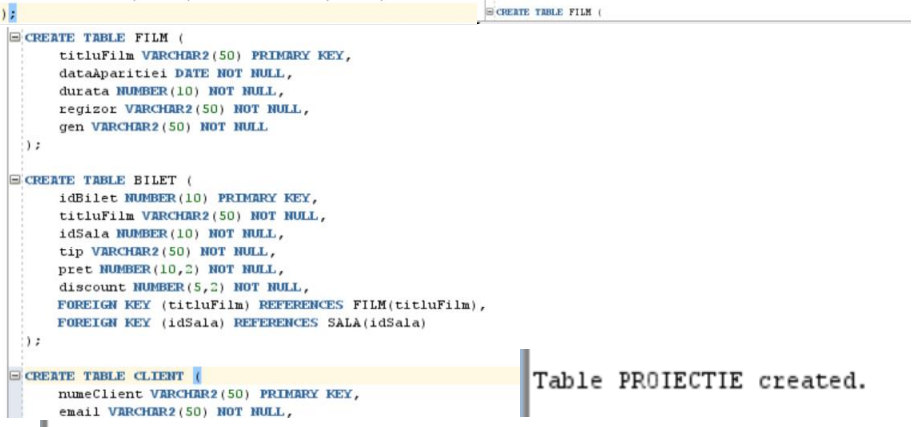
2. Implementarea bazei de date în Oracle

a. Crearea tabelelor

```
CREATE TABLE CINEMATOGRAF (
    idCinematograf NUMBER(10) PRIMARY KEY,
    numeCinematograf VARCHAR2(50) NOT NULL,
    adresa VARCHAR2(50) NOT NULL
);

CREATE TABLE ANGAJAT (
    numeAngajat VARCHAR2(50) PRIMARY KEY,
    idCinematograf NUMBER(10) NOT NULL,
    salariu NUMBER(10,2) NOT NULL,
    FOREIGN KEY (idCinematograf) REFERENCES CINEMATOGRAF(idCinematograf)
);

CREATE TABLE SALA (
    idSala NUMBER(10) PRIMARY KEY,
    idCinematograf NUMBER(10) NOT NULL,
    numarLocuri NUMBER(10) NOT NULL,
    tipProiectie VARCHAR2(50) NOT NULL,
    FOREIGN KEY (idCinematograf) REFERENCES CINEMATOGRAF(idCinematograf)
);

CREATE TABLE PROIECTIE (
    idProiectie NUMBER(10) PRIMARY KEY,
    titluFilm VARCHAR2(50) NOT NULL,
    idSala NUMBER(10) NOT NULL,
    oraInceput TIMESTAMP NOT NULL,
    oraFinal TIMESTAMP NOT NULL,
    dataProiectie DATE NOT NULL,
    FOREIGN KEY (titluFilm) REFERENCES FILM(titluFilm),
    FOREIGN KEY (idSala) REFERENCES SALA(idSala)
);

CREATE TABLE FILM (
    titluFilm VARCHAR2(50) PRIMARY KEY,
    dataAparitiei DATE NOT NULL,
    durata NUMBER(10) NOT NULL,
    regizor VARCHAR2(50) NOT NULL,
    gen VARCHAR2(50) NOT NULL
);

CREATE TABLE BILET (
    idBilet NUMBER(10) PRIMARY KEY,
```

```
    titluFilm VARCHAR2(50) NOT NULL,
    numeClient VARCHAR2(50),
    idSala NUMBER(10) NOT NULL,
    tip VARCHAR2(50) NOT NULL,
    pret NUMBER(10,2) NOT NULL,
    discount NUMBER(5,2) NOT NULL,
    FOREIGN KEY (numeClient) REFERENCES CLIENT(numeClient),
    FOREIGN KEY (titluFilm) REFERENCES FILM(titluFilm),
    FOREIGN KEY (idSala) REFERENCES SALA(idSala)
);

CREATE TABLE CLIENT (
    numeClient VARCHAR2(50) PRIMARY KEY,
    email VARCHAR2(50) NOT NULL,
    abonament VARCHAR2(50) NOT NULL,
    numarTelefon VARCHAR2(50) NOT NULL
);
```






```
Table PROIECTIE created.
Table CINEMATOGRAF created.
Table FILM created.
Table ANGAJAT created.
Table BILET created.
Table SALA created.
Table CLIENT created.
```

b. Insearea datelor în tabele

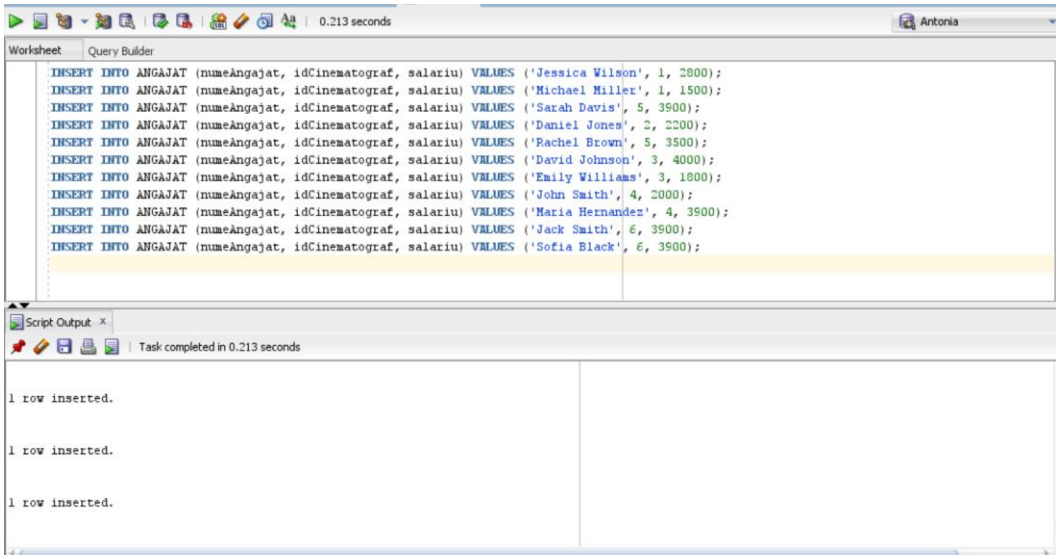
```
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (1,
'CinemaPlex', '3 Downing Street, London, UK');
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (2,
'Balzac Theatre', '5 rue de la Paix, Paris, France');
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (3,
'Hollywood Multiplex 3', '9 Unter den Linden, Berlin, Germany');
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (4,
'Star Cinema', '17 Istiklal Caddesi, Istanbul, Turkey');
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (5,
'FMI Cinema', 'Bulevardul Mihail Kogălniceanu 36-46, București, Romania');
INSERT INTO CINEMATOGRAF (idCinematograf, numeCinematograf, adresa) VALUES (6,
'Broadway Cinema', '13 Abbey Road, London, UK');

INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Christopher
Taylor', 1, 2400);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Jessica
Wilson', 1, 2800);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Michael
Miller', 1, 1500);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Sarah Davis',
5, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Daniel
Jones', 2, 2200);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Rachel
Brown', 5, 3500);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('David
Johnson', 3, 4000);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Emily
Williams', 3, 1800);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('John Smith',
4, 2000);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Maria
Hernandez', 4, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Jack Smith',
6, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Sofia Black',
6, 3900);

INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (1,
1, 50, '2D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (2,
1, 60, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (3,
2, 70, '4D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (4,
6, 80, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (5,
6, 90, '2D');
```

```
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (6,
3, 100, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (7,
4, 110, '4D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (8,
4, 120, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (9,
5, 130, '2D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (10,
5, 140, '3D');


INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The Godfather', TO_DATE('1972-03-24','YYYY-MM-DD'), 175, 'Francis Ford
Coppola', 'Crime');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The Shawshank Redemption', TO_DATE('1994-09-23','YYYY-MM-DD'), 142,
'Frank Darabont', 'Drama');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The Dark Knight', TO_DATE('2008-07-18','YYYY-MM-DD'), 152, 'Christopher
Nolan', 'Action');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Pulp Fiction', TO_DATE('1994-09-10','YYYY-MM-DD'), 154, 'Quentin
Tarantino', 'Crime');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Forrest Gump', TO_DATE('1994-07-06','YYYY-MM-DD'), 142, 'Robert
Zemeckis', 'Drama');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The Godfather: Part II', TO_DATE('1974-12-20','YYYY-MM-DD'), 202,
'Francis Ford Coppola', 'Crime');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The Matrix', TO_DATE('1999-03-31','YYYY-MM-DD'), 136, 'Lana Wachowski',
'Sci-Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Interstellar', TO_DATE('2014-11-05','YYYY-MM-DD'), 169, 'Christopher
Nolan', 'Sci-Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Inception', TO_DATE('2010-07-16','YYYY-MM-DD'), 148, 'Christopher
Nolan', 'Sci-Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Avatar', TO_DATE('2009-12-10','YYYY-MM-DD'), 162, 'James Cameron', 'Sci-
Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Terminator 2: Judgment Day', TO_DATE('1991-07-03','YYYY-MM-DD'), 137,
'James Cameron', 'Sci-Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Alien', TO_DATE('1979-05-25','YYYY-MM-DD'), 117, 'Ridley Scott', 'Sci-
Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('Blade Runner', TO_DATE('1982-06-25','YYYY-MM-DD'), 117, 'Ridley Scott',
'Sci-Fi');
```

```
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('E.T. the Extra-Terrestrial', TO_DATE('1982-06-11','YYYY-MM-DD'), 115,
'Steven Spielberg', 'Sci-Fi');
INSERT INTO FILM (titluFilm, dataAparitiei, durata, regizor, gen)
VALUES ('The War of the Worlds', TO_DATE('1953-10-30','YYYY-MM-DD'), 85, 'Byron
Haskin', 'Sci-Fi');


-- ++ the godfather
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (3, 'The Dark Knight', 3, timestamp '2022-12-24 14:00:00', timestamp
'2022-12-24 17:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (4, 'Pulp Fiction', 4, timestamp '2022-12-24 15:00:00', timestamp '2022-
12-24 18:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (5, 'Forrest Gump', 5, timestamp '2022-12-24 11:00:00', timestamp '2022-
12-24 14:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (6, 'The Godfather: Part II', 6, timestamp '2022-12-24 17:00:00',
timestamp '2022-12-24 20:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (7, 'Alien', 10, timestamp '2022-12-24 12:00:00', timestamp '2022-12-24
15:15:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (8, 'Terminator 2: Judgment Day', 2, timestamp '2022-12-24 17:00:00',
timestamp '2022-12-24 19:30:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (9, 'Avatar', 3, timestamp '2022-11-25 14:00:00', timestamp '2022-11-25
17:00:00', TO_DATE('2022-11-25','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (10, 'Inception', 8, timestamp '2022-12-24 15:00:00', timestamp '2022-12-
24 18:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (11, 'Interstellar', 5, timestamp '2022-12-24 16:00:00', timestamp '2022-
12-24 19:00:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (12, 'Avatar', 6, timestamp '2022-12-12 17:00:00', timestamp '2022-12-12
20:00:00', TO_DATE('2022-12-12','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
```

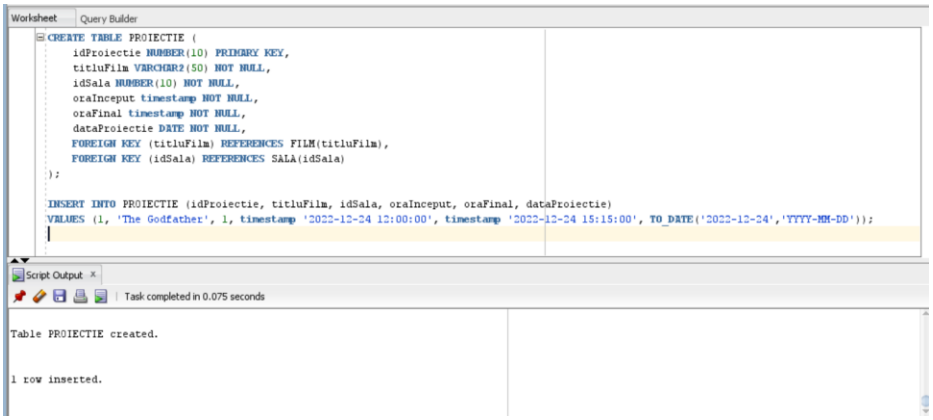
```
VALUES (13, 'Alien', 7, timestamp '2023-01-04 12:00:00', timestamp '2023-01-04
14:30:00', TO_DATE('2023-01-04','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (14, 'Terminator 2: Judgment Day', 6, timestamp '2023-01-01 13:00:00',
timestamp '2023-01-01 15:30:00', TO_DATE('2023-01-01','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (15, 'E.T. the Extra-Terrestrial', 9, timestamp '2023-03-05 14:00:00',
timestamp '2023-03-05 16:30:00', TO_DATE('2023-03-05','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (16, 'E.T. the Extra-Terrestrial', 5, timestamp '2023-01-08 15:00:00',
timestamp '2023-01-08 17:30:00', TO_DATE('2023-01-08','YYYY-MM-DD'));
INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
VALUES (17, 'Blade Runner', 10, timestamp '2023-02-01 16:00:00', timestamp '2023-
02-01 18:30:00', TO_DATE('2023-02-01','YYYY-MM-DD'));


INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('John Smith', 'johnsmith@gmail.com', 'yes', '123-456-7890');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Jane Doe', 'janedoe@gmail.com', 'no', '123-456-7891');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Bob Johnson', 'bobjohnson@gmail.com', 'no', '123-456-7892');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Alice Williams', 'alicewilliams@gmail.com', 'yes', '123-456-7893');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Mike Brown', 'mikebrown@gmail.com', 'no', '123-456-7894');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Emily Davis', 'emilydavis@gmail.com', 'yes', '123-456-7895');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('William Thompson', 'williamthompson@gmail.com', 'no', '123-456-7896');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Sarah Johnson', 'sarahj@gmail.com', 'No', '234-567-8901');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Robert Williams', 'rwilliams@gmail.com', 'Yes', '345-678-9012');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Jessica Davis', 'jdavis@gmail.com', 'No', '456-789-0123');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Amelie Rousseau', 'amelierousseau@gmail.com', 'no', '+33 987 654 321');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Giovanni Bianchi', 'giovannibianchi@gmail.com', 'yes', '+39 123 456
789');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Chiara Marino', 'chiaramarino@gmail.com', 'no', '+39 987 654 321');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Marc Dupont', 'marcdupont@gmail.com', 'yes', '+33 123 456 789');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Sophie Martin', 'sophiemartin@gmail.com', 'no', '+33 987 654 321');
```

```
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Mario Rossi', 'mariorossi@gmail.com', 'yes', '+39 123 456 789');
INSERT INTO CLIENT (numeClient, email, abonament, numarTelefon)
VALUES ('Isabella Verdi', 'isabellaverdi@gmail.com', 'no', '+39 987 654 321');


INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (1, 'The Godfather', 1, 'regular', 15.50, 'John Smith', 0.00);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (2, 'The Shawshank Redemption', 2, 'student', 12.75, 'Jane Doe', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (3, 'The Dark Knight', 3, 'child', 10.00, 'Bob Johnson', 0.25);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (4, 'Pulp Fiction', 4, 'regular', 15.50, 'Alice Williams', 0.00);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (5, 'Forrest Gump', 5, 'student', 12.75, 'Mike Brown', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (6, 'The Godfather: Part II', 6, 'regular', 12.99, 'Emily Davis', 0);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (7, 'Blade Runner', 10, 'regular', 12.99, 'William Thompson', 0);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (8, 'Interstellar', 5, 'regular', 12.99, 'Sarah Johnson', 0);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (9, 'Inception', 8, 'regular', 12.99, 'Robert Williams', 0);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (10, 'Avatar', 6, 'regular', 12.99, 'Jessica Davis', 0);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (11, 'Blade Runner', 10, 'student', 15.00, 'Giovanni Bianchi', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (12, 'Interstellar', 5, 'student', 15.00, 'Chiara Marino', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (13, 'Inception', 8, 'child', 15.00, 'Marc Dupont', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (14, 'Avatar', 6, 'student', 15.00, 'Sophie Martin', 0.00);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (15, 'Terminator 2: Judgment Day', 6, 'student', 15.00, 'Mario Rossi',
0.33);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (16, 'Alien', 7, 'student', 15.00, 'Isabella Verdi', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (17, 'The Godfather: Part II', 6, 'regular', 15.00, 'Amelie Rousseau',
0.00);
```



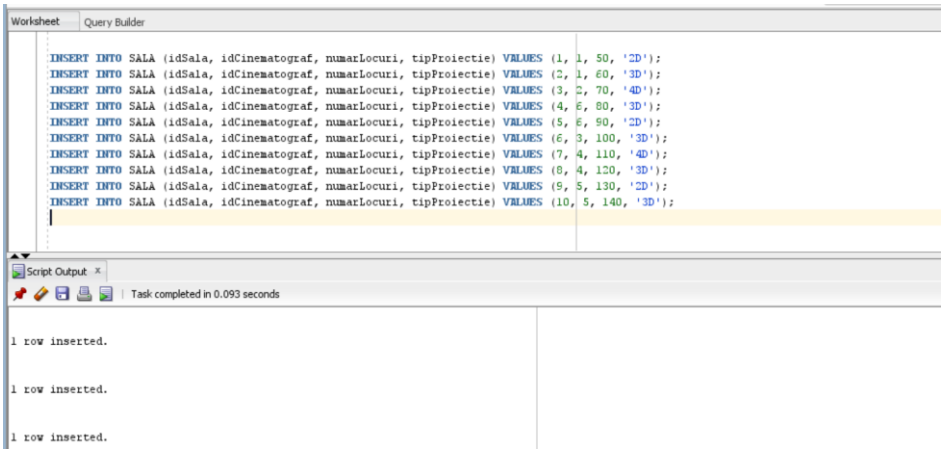
```
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Jessica Wilson', 1, 2800);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Michael Miller', 1, 1500);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Sarah Davis', 5, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Daniel Jones', 2, 2200);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Rachel Brown', 5, 3500);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('David Johnson', 3, 4000);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Emily Williams', 3, 1800);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('John Smith', 4, 2000);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Maria Hernandez', 4, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Jack Smith', 6, 3900);
INSERT INTO ANGAJAT (numeAngajat, idCinematograf, salariu) VALUES ('Sofia Black', 6, 3900);
```

```
1 row inserted.

1 row inserted.

1 row inserted.
```



```
CREATE TABLE PROIECTIE (
    idProiectie NUMBER(10) PRIMARY KEY,
    titluFilm VARCHAR2(50) NOT NULL,
    idSala NUMBER(10) NOT NULL,
    oraInceput timestamp NOT NULL,
    oraFinal timestamp NOT NULL,
    dataProiectie DATE NOT NULL,
    FOREIGN KEY (titluFilm) REFERENCES FILM(titluFilm),
    FOREIGN KEY (idSala) REFERENCES SALA(idSala)
);

INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal, dataProiectie)
VALUES (1, 'The Godfather', 1, timestamp '2022-12-24 12:00:00', timestamp '2022-12-24 15:15:00', TO_DATE('2022-12-24','YYYY-MM-DD'));
```

```
Table PROIECTIE created.

1 row inserted.
```



```
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (1, 1, 50, '2D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (2, 1, 60, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (3, 2, 70, '4D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (4, 6, 80, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (5, 6, 90, '2D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (6, 3, 100, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (7, 4, 110, '4D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (8, 4, 120, '3D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (9, 5, 130, '2D');
INSERT INTO SALA (idSala, idCinematograf, numarLocuri, tipProiectie) VALUES (10, 5, 140, '3D');
```

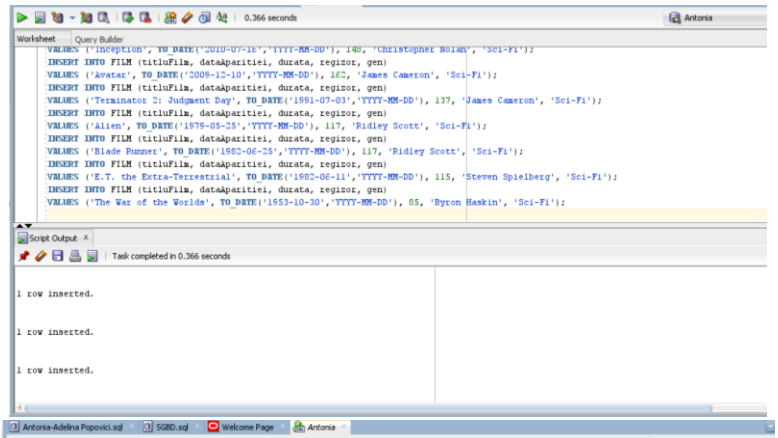
```
1 row inserted.

1 row inserted.

1 row inserted.
```

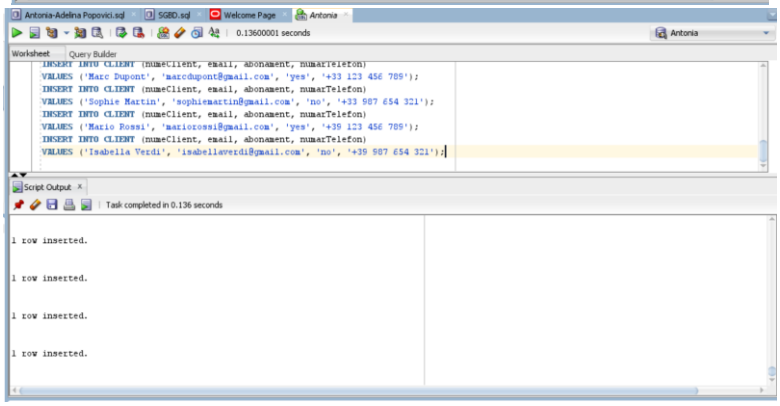
```
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (1, 'The Godfather', 1, 'regular', 15.50, 'John Smith', 0.00);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (2, 'The Shawshank Redemption', 2, 'student', 12.75, 'Jane Doe', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (3, 'The Dark Knight', 3, 'child', 10.00, 'Bob Johnson', 0.25);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (4, 'Pulp Fiction', 4, 'regular', 15.50, 'Alice Williams', 0.00);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (5, 'Forrest Gump', 5, 'student', 12.75, 'Mike Brown', 0.50);
INSERT INTO BILET (idBilet, titluFilm, idSala, tip, pret, numeClient, discount)
VALUES (6, 'The Godfather: Part II', 6, 'regular', 12.99, 'Emily Davis', 0);
```

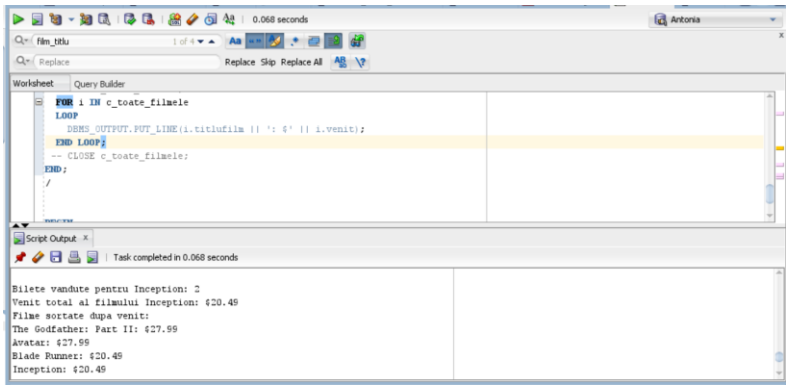
Script Output

Task completed in 0.165 seconds

```
1 row inserted.


1 row inserted.
```







	NUMEANGAJAT	IDCINEMATOGRAF	SALARIU
1	Christopher Taylor	1	2400
2	Jessica Wilson	1	2800
3	Michael Miller	1	1500
4	Sarah Davis	5	3900
5	Daniel Jones	2	2200
6	Rachel Brown	5	3500
7	David Johnson	3	4000
8	Emily Williams	3	1800
9	John Smith	4	2000
10	Maria Hernandez	4	3900
11	Jack Smith	6	3900
12	Sofia Black	6	3900

	NUMECLIENT	EMAIL	ABONAMENT	NUMARTELEFON
1	John Smith	johnsmith@gmail.com	yes	123-456-7890
2	Jane Doe	janedoe@gmail.com	no	123-456-7891
3	Bob Johnson	bobjohnson@gmail.com	no	123-456-7892
4	Alice Williams	alicewilliams@gmail.com	yes	123-456-7893
5	Mike Brown	mikebrown@gmail.com	no	123-456-7894
6	Emily Davis	emilydavis@gmail.com	yes	123-456-7895
7	William Thompson	williamthompson@gmail.com	no	123-456-7896
8	Sarah Johnson	sarahj@gmail.com	No	234-567-8901
9	Robert Williams	rwilliams@gmail.com	Yes	345-678-9012
10	Jessica Davis	jdavis@gmail.com	No	456-789-0123
11	Amelie Rousseau	amelierousseau@gmail.com	no	+33 987 654 321
12	Giovanni Bianchi	giovannibianchi@gmail.com	yes	+39 123 456 789
13	Chiara Marino	chiaramarino@gmail.com	no	+39 987 654 321
14	Marc Dupont	marcdupont@gmail.com	yes	+33 123 456 789
15	Sophie Martin	sophiemartin@gmail.com	no	+33 987 654 321
16	Mario Rossi	mariorossi@gmail.com	yes	+39 123 456 789
17	Isabella Verdi	isabellaverdi@gmail.com	no	+39 987 654 321

	TITLUFILM	DATAAPARITIEI	DURATA	REGIZOR	GEN
1	The Godfather	24-MAR-72	175	Francis Ford Coppola	Crime
2	The Shawshank Redemption	23-SEP-94	142	Frank Darabont	Drama
3	The Dark Knight	18-JUL-08	152	Christopher Nolan	Action
4	Pulp Fiction	10-SEP-94	154	Quentin Tarantino	Crime
5	Forrest Gump	06-JUL-94	142	Robert Zemeckis	Drama
6	The Godfather: Part II	20-DEC-74	202	Francis Ford Coppola	Crime
7	The Matrix	31-MAR-99	136	Lana Wachowski	Sci-Fi
8	Interstellar	05-NOV-14	169	Christopher Nolan	Sci-Fi
9	Inception	16-JUL-10	148	Christopher Nolan	Sci-Fi
10	Avatar	10-DEC-09	162	James Cameron	Sci-Fi
11	Terminator 2: Judgment Day	03-JUL-91	137	James Cameron	Sci-Fi
12	Alien	25-MAY-79	117	Ridley Scott	Sci-Fi
13	Blade Runner	25-JUN-82	117	Ridley Scott	Sci-Fi
14	E.T. the Extra-Terrestrial	11-JUN-82	115	Steven Spielberg	Sci-Fi
15	The War of the Worlds	30-OCT-53	85	Byron Haskin	Sci-Fi

	IDSALA	IDCINEMATOGRAF	NUMARLOCURI	TIPPROIECTIE
1	1	1	50	2D
2	2	1	60	3D
3	3	2	70	4D
4	4	6	80	3D
5	5	6	90	2D
6	6	3	100	3D
7	7	4	110	4D
8	8	4	120	3D
9	9	5	130	2D
10	10	5	140	3D

	IDCINEMATOGRAF	NUMECINEMATOGRAF	ADRESA
1	1	CinemaPlex	3 Downing Street, London, UK
2	2	Balzac Theatre	5 rue de la Paix, Paris, France
3	3	Hollywood Multiplex 3	9 Unter den Linden, Berlin, Germany
4	4	Star Cinema	17 Istiklal Caddesi, Istanbul, Turkey
5	6	Broadway Cinema	13 Abbey Road, London, UK
6	5	FMI Cinema	Bd-ul Mihail Kogălniceanu, București, Romania

	TITLUFILM	DATAAPARITIEI	DURATA	REGIZOR	GEN
1	The Godfather	24-MAR-72	175	Francis Ford Coppola	Crime
2	The Shawshank Redemption	23-SEP-94	142	Frank Darabont	Drama
3	The Dark Knight	18-JUL-08	152	Christopher Nolan	Action
4	Pulp Fiction	10-SEP-94	154	Quentin Tarantino	Crime
5	Forrest Gump	06-JUL-94	142	Robert Zemeckis	Drama
6	The Godfather: Part II	20-DEC-74	202	Francis Ford Coppola	Crime
7	The Matrix	31-MAR-99	136	Lana Wachowski	Sci-Fi
8	Interstellar	05-NOV-14	169	Christopher Nolan	Sci-Fi
9	Inception	16-JUL-10	148	Christopher Nolan	Sci-Fi
10	Avatar	10-DEC-09	162	James Cameron	Sci-Fi
11	Terminator 2: Judgment Day	03-JUL-91	137	James Cameron	Sci-Fi
12	Alien	25-MAY-79	117	Ridley Scott	Sci-Fi
13	Blade Runner	25-JUN-82	117	Ridley Scott	Sci-Fi
14	E.T. the Extra-Terrestrial	11-JUN-82	115	Steven Spielberg	Sci-Fi
15	The War of the Worlds	30-OCT-53	85	Byron Haskin	Sci-Fi

	IDBILET	TITLUFILM	IDSALA	TIP	PRET	DISCOUNT	NUMECLIENT
1	1	The Godfather	1	regular	15.5	0	John Smith
2	2	The Shawshank Redemption	2	student	12.75	0.5	Jane Doe
3	3	The Dark Knight	3	child	10	0.25	Bob Johnson
4	4	Pulp Fiction	4	regular	15.5	0	Alice Williams
5	5	Forrest Gump	5	student	12.75	0.5	Mike Brown
6	6	The Godfather: Part II	6	regular	12.99	0	Emily Davis
7	7	Blade Runner	10	regular	12.99	0	William Thompson
8	8	Interstellar	5	regular	12.99	0	Sarah Johnson
9	9	Inception	8	regular	12.99	0	Robert Williams
10	10	Avatar	6	regular	12.99	0	Jessica Davis
11	11	Blade Runner	10	student	15	0.5	Giovanni Bianchi
12	12	Interstellar	5	student	15	0.5	Chiara Marino
13	13	Inception	8	child	15	0.5	Marc Dupont
14	14	Avatar	6	student	15	0	Sophie Martin
15	15	Terminator 2: Judgment Day	6	student	15	0.33	Mario Rossi
16	16	Alien	7	student	15	0.5	Isabella Verdi
17	17	The Godfather: Part II	6	regular	15	0	Amelie Rousseau

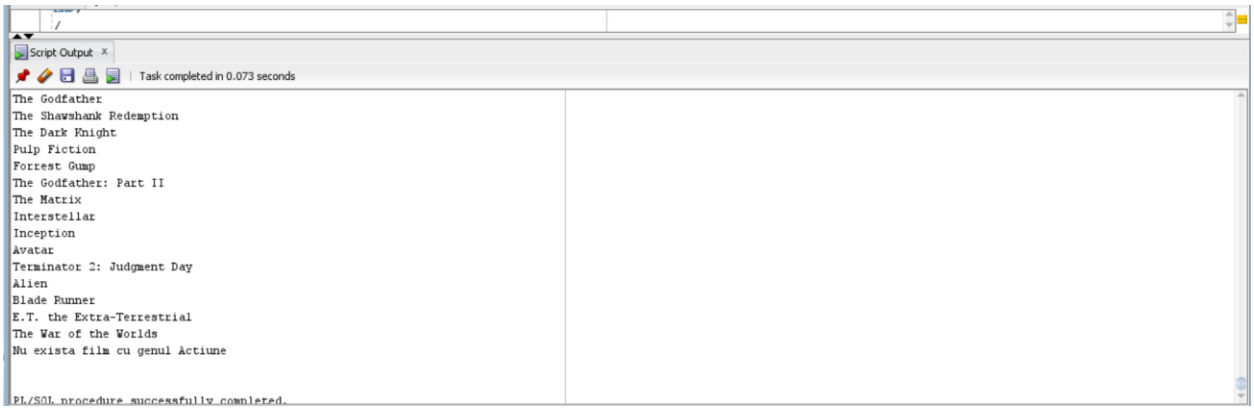
3. Probleme formulate în limbaj natural ce pot fi rezolvate utilizând PL/SQL

a. Problemă cu două tipuri de colecţii

Afişaţi toate filmele care pot fi proiectate şi filmele care au genul egal cu genul dat ca parametru.

```
exec dbms_output.enable;
SET SERVEROUTPUT ON SIZE 1000000;

CREATE OR REPLACE PROCEDURE colectie_ex6(gen_param in varchar2)
IS

    TYPE toate_filmele is table of film%rowtype;
    v_toate_filmele toate_filmele;
    film_negasit EXCEPTION;

    TYPE filme is varray(100) of film%rowtype;
    v_filme filme;
    v_film film%rowtype;

BEGIN
    SELECT * BULK COLLECT INTO v_toate_filmele from FILM;

    DBMS_OUTPUT.PUT_LINE('Toate filmele:');

    FOR i in 1..v_toate_filmele.COUNT loop
        v_film := v_toate_filmele(i);
        DBMS_OUTPUT.PUT_LINE(v_film.TitluFilm);
    END LOOP;

    SELECT * bulk collect into v_filme FROM film WHERE gen = gen_param;
    IF v_filme.COUNT = 0 THEN
        RAISE film_negasit;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Filme care sunt de genul: ' || gen_param || ':');

    FOR i IN 1 .. v_filme.COUNT LOOP
        v_film := v_filme(i);
        DBMS_OUTPUT.PUT_LINE(v_film.titlufilm);
    END LOOP;

    EXCEPTION
        WHEN film_negasit THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista film cu genul ' || gen_param);
```

```
END;
/

/*
BEGIN
 colectie_ex6('Sci-Fi');
colectie_ex6('Actiune');
END;
*/
```



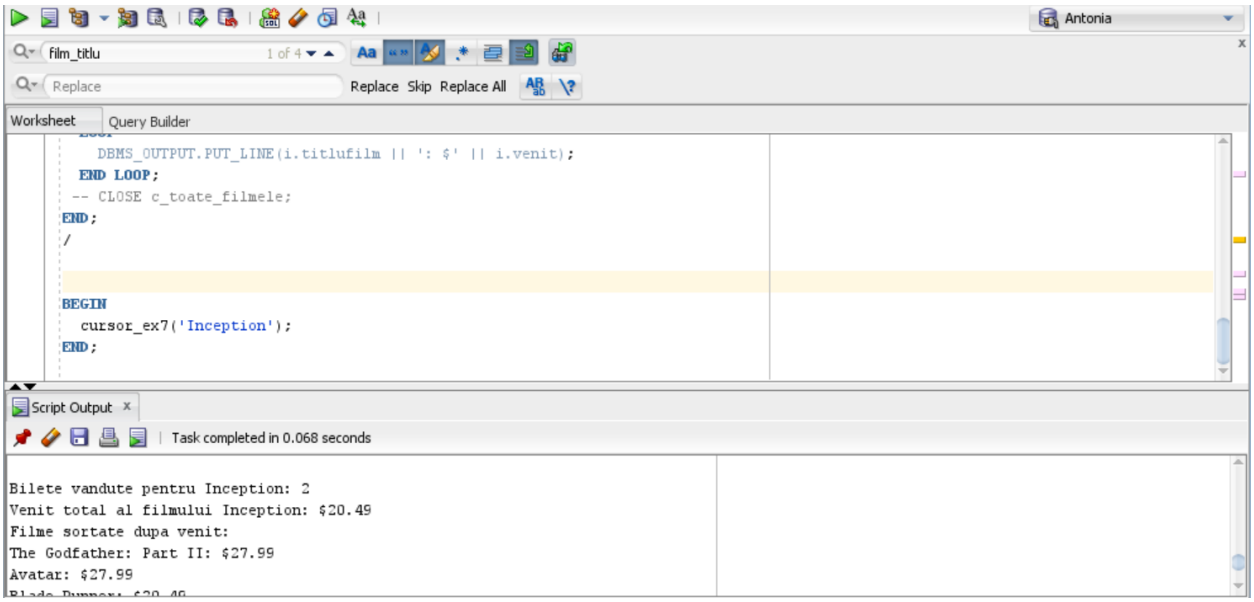
b. Problemă cu două tipuri de cursoare

Creaţi un subprogram stocat independent care va avea ca parametru titlul unui film şi va întoarce:

- numărul total de bilete vândute pentru filmul menţionat
- venitul total generat de vânzarea de bilete pentru filmul respectiv
- o listă cu toate filmele, sortate conform venitului

```
EXEC DBMS_OUTPUT.ENABLE;
SET SERVEROUTPUT ON SIZE 1000000;

CREATE OR REPLACE PROCEDURE cursor_ex7 (film_titlu IN VARCHAR2)
AS
  -- variable
  bilete_in_total INTEGER;
  total_venit NUMBER(10, 2);

  -- cursor parametrizat
  CURSOR c_vanzari_bilete (p_film_titlu VARCHAR2) IS
    SELECT COUNT(*), SUM(pret - (pret * discount))
    FROM bilet
    WHERE titlufilm = p_film_titlu;

  -- cursor neparametrizat
  CURSOR c_toate_filmele IS
    SELECT titlufilm, SUM(pret - (pret * discount)) as venit
```

```
    FROM bilet
    GROUP BY titlufilm
    ORDER BY venit DESC;

BEGIN
  -- deschidere cursor si trimitere date in variabile
  OPEN c_vanzari_bilete(film_titlu);
  FETCH c_vanzari_bilete INTO bilete_in_total, total_venit;
 CLOSE c_vanzari_bilete;

  DBMS_OUTPUT.PUT_LINE('Bilete vandute pentru ' || film_titlu || ': ' ||
bilete_in_total);
  DBMS_OUTPUT.PUT_LINE('Venit total al filmului ' || film_titlu || ': $' ||
total_venit);

  -- sortate dupa suma
  DBMS_OUTPUT.PUT_LINE('Filme sortate dupa venit:');
 -- OPEN c_toate_filmele;
  FOR i IN c_toate_filmele
  LOOP
    DBMS_OUTPUT.PUT_LINE(i.titlufilm || ': $' || i.venit);
  END LOOP;
 -- CLOSE c_toate_filmele;
END;
/
BEGIN
  cursor_ex7('Inception');
END;
```



c. Problemă de tip funcţie ce se foloseşte de trei dintre tabelele definite

Calculaţi şi returnaţi venitul total generat de un cinematograf al cărui ID este dat ca parametru.

```
EXECUTE DBMS_OUTPUT.enable();

CREATE OR REPLACE FUNCTION functie_8 (p_cinematograf_id IN NUMBER, p_dataproiectie IN
DATE)
RETURN NUMBER
AS
  v_venit NUMBER(10,2);
  v_cinematograf NUMBER(10);
BEGIN
    --  tratare caz: exista cinematograful dat ca parametru?
    SELECT idcinematograf INTO v_cinematograf FROM cinematograf WHERE idcinematograf
= p_cinematograf_id;

    IF v_cinematograf IS NULL THEN
      RAISE_APPLICATION_ERROR(-20000, 'Nu exista cinematograful ' || v_cinematograf);
    END IF;

    SELECT SUM(pret * (1 - discount)) INTO v_venit
    FROM BILET t
    JOIN PROIECTIE p ON t.titlufilm = p.titlufilm AND t.idsala = p.idsala
    JOIN SALA h ON p.idsala = h.idsala
    WHERE h.idcinematograf = p_cinematograf_id AND p.dataproiectie = p_dataproiectie;

    -- tratare caz: nu exista proiectii pentru data mentionata
    IF v_venit IS NULL THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nu exista proiectii pentru data ' ||
p_dataproiectie);
    END IF;

    RETURN v_venit;
END;
/

DECLARE
   n number(4);
BEGIN
--   n :=  functie_8(89, '24-DEC-22');
--   n :=  functie_8(6, '24-DEC-22');
--   n :=  functie_8(6, '20-DEC-22');
   dbms_output.put_line('Venitul este: ' || n);
END;
/
```

```
DECLARE
   n number(4);
BEGIN
   n :=  functie_8(89, '24-DEC-22');
--   n :=  functie_8(6, '24-DEC-22');
--   n :=  functie_8(6, '20-DEC-22');
   dbms_output.put_line('Venitul este: ' || n);
END;
/
```

Script Output | Explain Plan | Task completed in 0.089 seconds

```
ORA-06512: at "C##USERNOU.FUNCTIE_8", line 8
ORA-06512: at line 4
01403. 00000 -  "no data found"
*Cause:    No data was found from the objects.
*Action:   There was no data from the objects which may be due to end of fetch.
```

Worksheet | Query Builder

```
DECLARE
   n number(4);
BEGIN
-- n :=  functie_8(89, '24-DEC-22');
 n :=  functie_8(6, '24-DEC-22');
-- n :=  functie_8(6, '20-DEC-22');
   dbms_output.put_line('Venitul este: ' || n);
END;
/
```

Script Output | Explain Plan | Task completed in 0.044 seconds

```
Function FUNCTIE_8 compiled

Venitul este: 42


PL/SQL procedure successfully completed.
```

Worksheet | Query Builder

```
DECLARE
   n number(4);
BEGIN
-- n :=  functie_8(89, '24-DEC-22');
-- n :=  functie_8(6, '24-DEC-22');
 n :=  functie_8(6, '20-DEC-22');
   dbms_output.put_line('Venitul este: ' || n);
END;
/
```

Script Output | Explain Plan | Task completed in 0.071 seconds

```
-- n :=  functie_8(6, '24-DEC-22');
 n :=  functie_8(6, '20-DEC-22');
   dbms_output.put_line('Venitul este: ' || n);
END;
Error report -
ORA-20001: Nu exista proiectii pentru data 20-DEC-22
ORA-06512: at "C##USERNOU.FUNCTIE_8", line 21
ORA-06512: at line 6
```

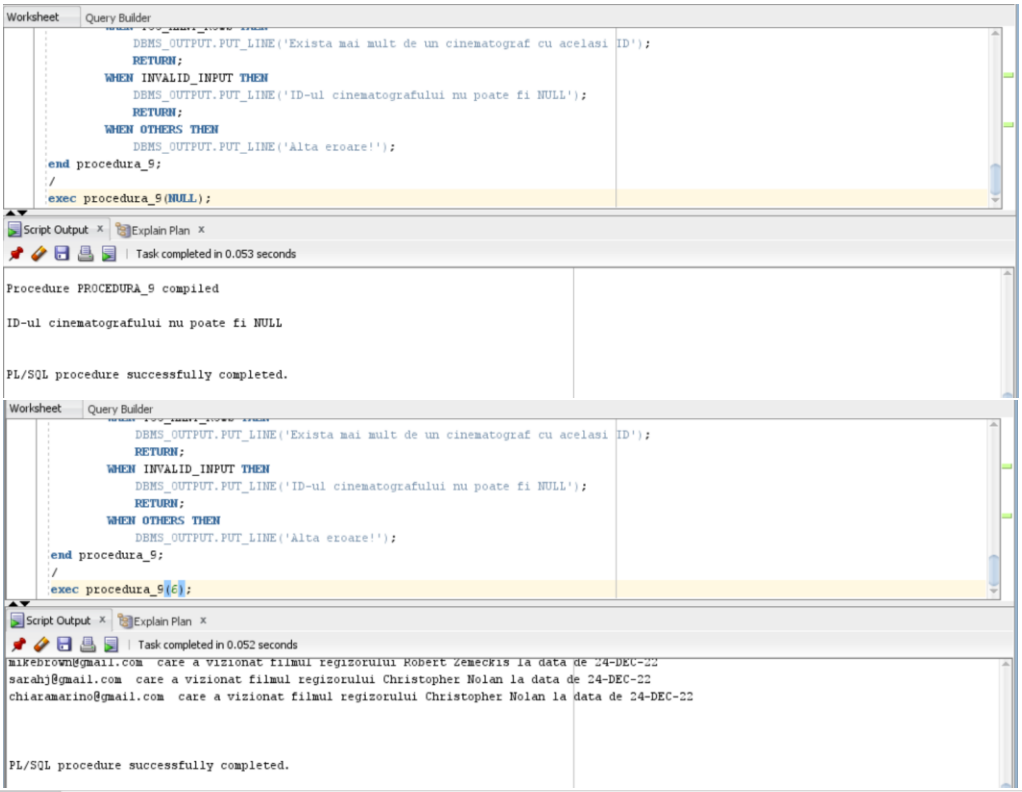
d. Problema de tip procedură ce se foloseşte de cinci dintre tabelele definite

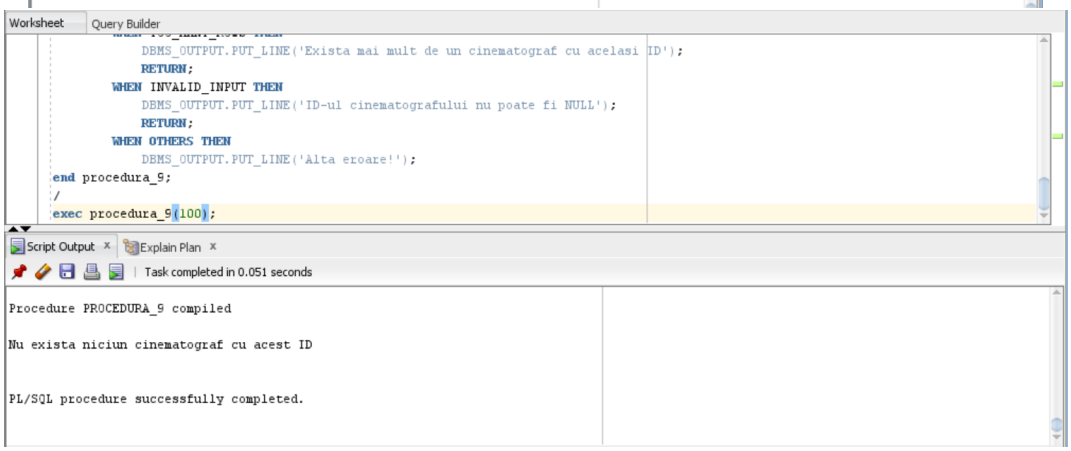
Pentru codul unui cinematograf trimis ca parametru, afişaţi:

- numele cinematografului
- mail-urile clienţilor care au fost la cinematograful menţionat
- regizorii filmelor vizionate

```
exec dbms_output.enable;
SET SERVEROUTPUT ON SIZE 1000000;

CREATE OR REPLACE PROCEDURE
       procedura_9(cod_cinema cinematograf.idcinematograf%TYPE) IS
    cnume cinematograf.numecinematograf%TYPE;
     CURSOR c1 IS
        SELECT email n, regizor t, idcinematograf cid2, dataproiectie dd
        FROM client sp JOIN bilet b ON (sp.numeclient=b.numeclient)
                       JOIN film f ON (b.titlufilm=f.titlufilm)
                       JOIN sala s ON (b.idsala=s.idsala)
                       join proiectie p on (p.titlufilm=f.titlufilm)
        WHERE idcinematograf=cod_cinema;
            TYPE_MISMATCH EXCEPTION;
         INVALID_INPUT EXCEPTION;

  BEGIN

    IF cod_cinema IS NULL THEN
        RAISE INVALID_INPUT;
    END IF;


        SELECT c.numecinematograf
        INTO cnume
        FROM cinematograf c
        WHERE idcinematograf=cod_cinema;
        DBMS_OUTPUT.PUT_LINE('Cinematograful ' || cnume ||' a avut urmatorii clienti:
');
        DBMS_OUTPUT.NEW_LINE();

        FOR i in c1 LOOP
            DBMS_OUTPUT.PUT_LINE(i.n || ' '  || ' care a vizionat filmul regizorului
' || i.t || ' la data de ' || i.dd);
        END LOOP;
        DBMS_OUTPUT.NEW_LINE();

   EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista niciun cinematograf cu acest ID' );
```

```
        RETURN;
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Exista mai mult de un cinematograf cu acelasi ID');
        RETURN;
    WHEN INVALID_INPUT THEN
        DBMS_OUTPUT.PUT_LINE('ID-ul cinematografului nu poate fi NULL');
        RETURN;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!');
end procedura_9;
/
exec procedura_9(6);
```



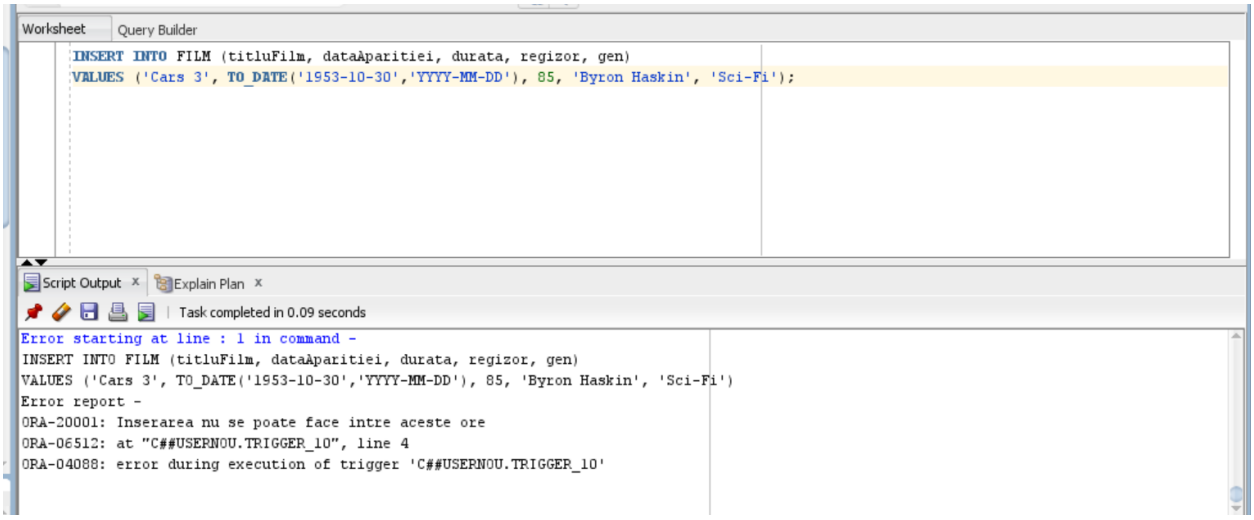


4. Triggere

a. Trigger LMD la nivel de comandă

Trigger care nu permite inserarea, ştergerea sau actulizarea datelor în tabelul FILM în anumite intervale definite.

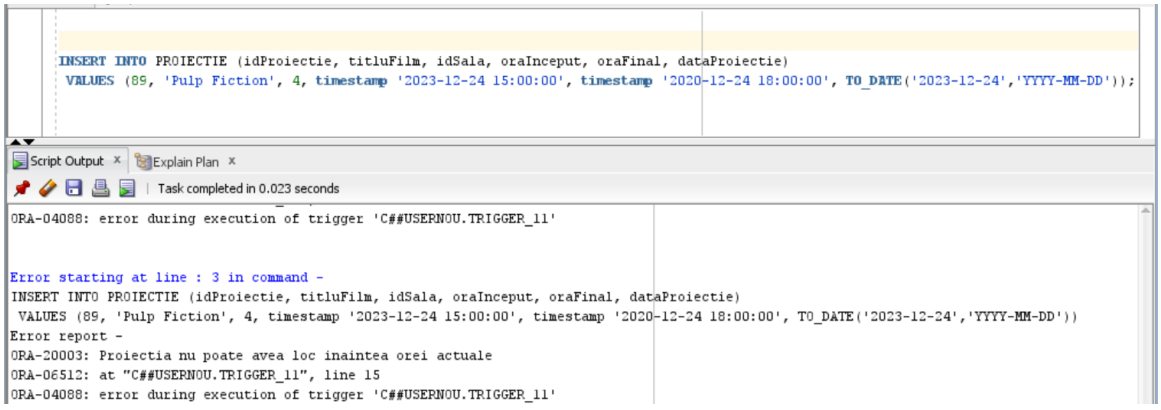
```
CREATE OR REPLACE TRIGGER trigger_10
BEFORE INSERT OR DELETE OR UPDATE on FILM
BEGIN
    IF (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 and 14) THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20001,'Inserarea nu se poate face intre aceste
ore');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20002,'Stergerea nu se poate face intre aceste
ore');
        ELSE
            RAISE_APPLICATION_ERROR(-20003,'Actualizarea nu se poate face intre
aceste ore');
        END IF;
--    ELSE  RAISE_APPLICATION_ERROR(-20003,'ok');
    END IF;
END;
```

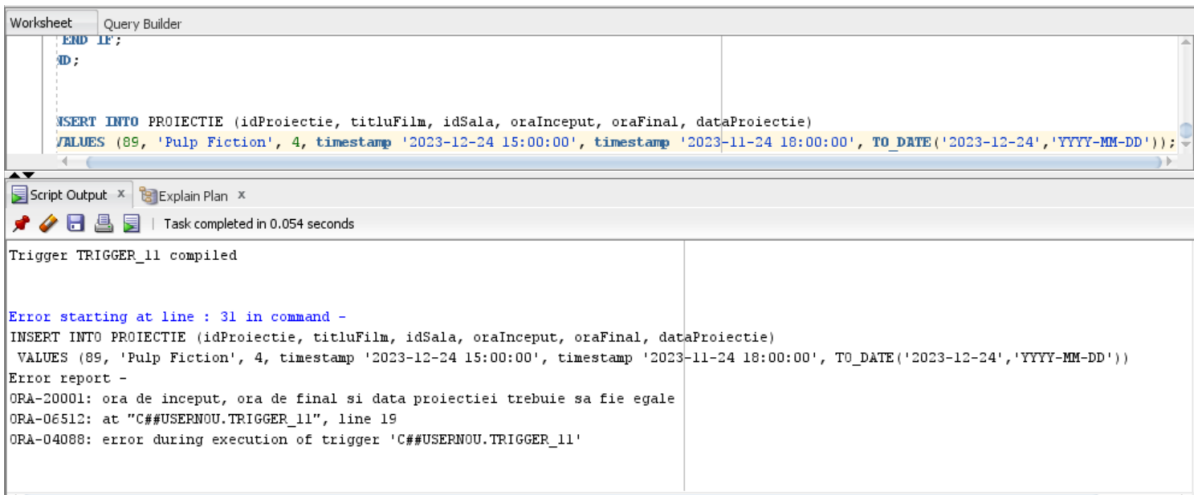


b. Trigger LMD la nivel de linie

Trigger care nu permite inserarea sau actulizarea datelor în tabelul PROIECŢIE dacă data proiecţiei este înainte de data actuală sau dacă filmele au loc la date diferite în atributele oraInceput, oraFinal, dataProiectie.

```
create or replace TRIGGER trigger_11
AFTER INSERT OR UPDATE ON proiectie
FOR EACH ROW
DECLARE
  v_sysdate DATE;
BEGIN
  SELECT SYSDATE INTO v_sysdate FROM DUAL;

  IF :NEW.dataproiectie < v_sysdate THEN
    RAISE_APPLICATION_ERROR(-20001, 'proiectia trebuie dupa sysdate');
  END IF;

  IF :NEW.orainceput < v_sysdate THEN
    RAISE_APPLICATION_ERROR(-20001, 'ora de inceput trebuie dupa sysdate');
  END IF;

  IF :NEW.orafinal < v_sysdate THEN
    RAISE_APPLICATION_ERROR(-20001, 'ora de final trebuie dupa sysdate');
    end if;

  IF :NEW.orafinal != :NEW.orainceput then
      RAISE_APPLICATION_ERROR(-20001, 'ora de inceput, ora de final si data
proiectiei trebuie sa fie egale');
  END IF;

  IF :NEW.orafinal != :NEW.dataproiectie then
      RAISE_APPLICATION_ERROR(-20001, 'ora de inceput, ora de final si data
proiectiei trebuie sa fie egale');
  END IF;
END;

/
-- INSERT INTO PROIECTIE (idProiectie, titluFilm, idSala, oraInceput, oraFinal,
dataProiectie)
 VALUES (89, 'Pulp Fiction', 4, timestamp '2023-12-24 15:00:00', timestamp '2023-11-
24 18:00:00', TO_DATE('2023-12-24','YYYY-MM-DD'));
```

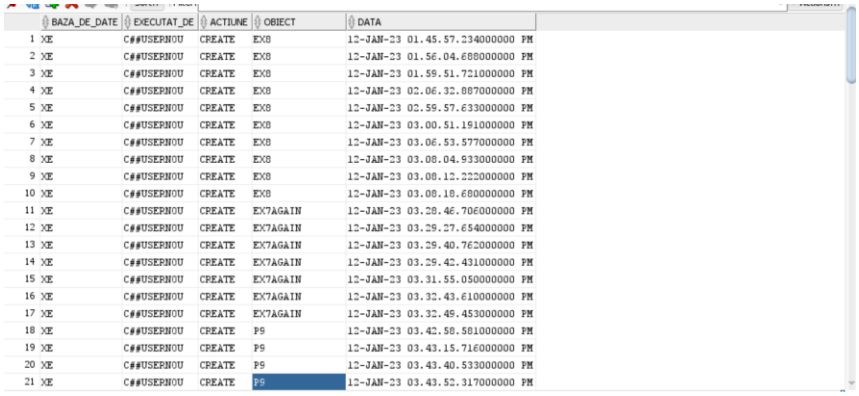




c. Trigger LDD

Trigger care salvează acțiunile efectuate de un User și le introduce într-un tabel.

```
CREATE TABLE trig12
   (baza_de_date VARCHAR2(50),
   executat_de VARCHAR2(30),
   actiune VARCHAR2(20),
   obiect VARCHAR2(30),
   data TIMESTAMP(3));

CREATE OR REPLACE TRIGGER trigger_12
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
   INSERT INTO trig12
   VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER,
           SYS.SYSEVENT,
           SYS.DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));
END;
/
```



	BAZA_DE_DATE	EXECUTAT_DE	ACTIUNE	OBIECT	DATA
1	XE	C##USERNOU	CREATE	EX8	12-JAN-23 01.45.57.234000000 PM
2	XE	C##USERNOU	CREATE	EX8	12-JAN-23 01.56.04.688000000 PM
3	XE	C##USERNOU	CREATE	EX8	12-JAN-23 01.59.51.721000000 PM
4	XE	C##USERNOU	CREATE	EX8	12-JAN-23 02.06.32.887000000 PM
5	XE	C##USERNOU	CREATE	EX8	12-JAN-23 02.59.57.633000000 PM
6	XE	C##USERNOU	CREATE	EX8	12-JAN-23 03.00.51.191000000 PM
7	XE	C##USERNOU	CREATE	EX8	12-JAN-23 03.06.53.577000000 PM
8	XE	C##USERNOU	CREATE	EX8	12-JAN-23 03.08.04.933000000 PM
9	XE	C##USERNOU	CREATE	EX8	12-JAN-23 03.08.12.222000000 PM
10	XE	C##USERNOU	CREATE	EX8	12-JAN-23 03.08.18.680000000 PM
11	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.28.46.706000000 PM
12	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.29.27.654000000 PM
13	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.29.40.762000000 PM
14	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.29.42.431000000 PM
15	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.31.55.050000000 PM
16	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.32.43.610000000 PM
17	XE	C##USERNOU	CREATE	EX7AGAIN	12-JAN-23 03.32.49.453000000 PM
18	XE	C##USERNOU	CREATE	P9	12-JAN-23 03.42.58.581000000 PM
19	XE	C##USERNOU	CREATE	P9	12-JAN-23 03.43.15.716000000 PM
20	XE	C##USERNOU	CREATE	P9	12-JAN-23 03.43.40.533000000 PM
21	XE	C##USERNOU	CREATE	P9	12-JAN-23 03.43.52.317000000 PM

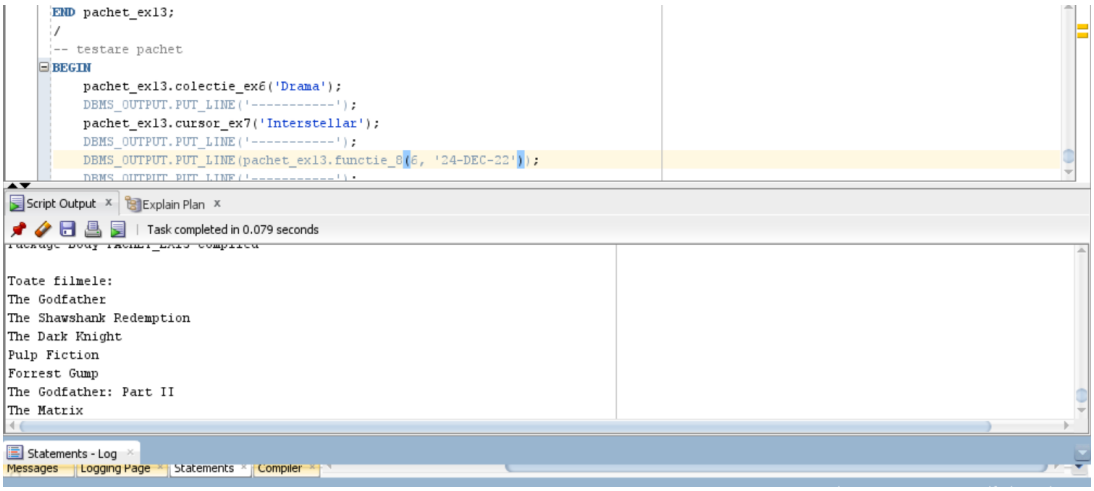
5. Pachet ce conține toate obiectele definite în cadrul proiectului

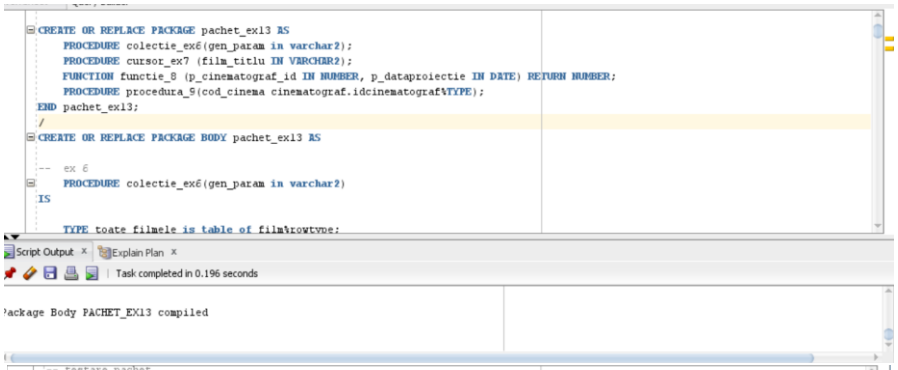
```
CREATE OR REPLACE PACKAGE pachet_ex13 AS
    PROCEDURE colectie_ex6(gen_param in varchar2);
    PROCEDURE cursor_ex7 (film_titlu IN VARCHAR2);
    FUNCTION functie_8 (p_cinematograf_id IN NUMBER, p_dataproiectie IN DATE) RETURN
NUMBER;
    PROCEDURE procedura_9(cod_cinema cinematograf.idcinematograf%TYPE);
END pachet_ex13;
/
CREATE OR REPLACE PACKAGE BODY pachet_ex13 AS

--  ex 6
    PROCEDURE colectie_ex6(gen_param in varchar2)
IS

    TYPE toate_filmele is table of film%rowtype;
    v_toate_filmele toate_filmele;
    film_negasit EXCEPTION;

    TYPE filme is varray(100) of film%rowtype;
    v_filme filme;
    v_film film%rowtype;

BEGIN
    SELECT * BULK COLLECT INTO v_toate_filmele from FILM;

    DBMS_OUTPUT.PUT_LINE('Toate filmele:');

    FOR i in 1..v_toate_filmele.COUNT loop
        v_film := v_toate_filmele(i);
        DBMS_OUTPUT.PUT_LINE(v_film.TitluFilm);
    END LOOP;

    SELECT * bulk collect into v_filme FROM film WHERE gen = gen_param;
    IF v_filme.COUNT = 0 THEN
        RAISE film_negasit;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Filme care sunt de genul: ' || gen_param || ':');

    FOR i IN 1 .. v_filme.COUNT LOOP
        v_film := v_filme(i);
        DBMS_OUTPUT.PUT_LINE(v_film.titlufilm);
    END LOOP;

    EXCEPTION
```

```
        WHEN film_negasit THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista film cu genul ' || gen_param);
END;


-- ex 7
    PROCEDURE cursor_ex7 (film_titlu IN VARCHAR2)
AS
  -- variabile
  bilete_in_total INTEGER;
  total_venit NUMBER(10, 2);

  -- cursor parametrizat
  CURSOR c_vanzari_bilete (p_film_titlu VARCHAR2) IS
    SELECT COUNT(*), SUM(pret - (pret * discount))
    FROM bilet
    WHERE titlufilm = p_film_titlu;

  -- cursor neparametrizat
  CURSOR c_toate_filmele IS
    SELECT titlufilm, SUM(pret - (pret * discount)) as venit
    FROM bilet
    GROUP BY titlufilm
    ORDER BY venit DESC;

BEGIN
  -- deschidere cursor si trimitere date in variabile
  OPEN c_vanzari_bilete(film_titlu);
  FETCH c_vanzari_bilete INTO bilete_in_total, total_venit;
 CLOSE c_vanzari_bilete;

  DBMS_OUTPUT.PUT_LINE('Bilete vandute pentru ' || film_titlu || ': ' ||
bilete_in_total);
  DBMS_OUTPUT.PUT_LINE('Venit total al filmului ' || film_titlu || ': $' ||
total_venit);

  -- sortate dupa suma
  DBMS_OUTPUT.PUT_LINE('Filme sortate dupa venit:');
 -- OPEN c_toate_filmele;
  FOR i IN c_toate_filmele
  LOOP
    DBMS_OUTPUT.PUT_LINE(i.titlufilm || ': $' || i.venit);
  END LOOP;
 -- CLOSE c_toate_filmele;
END;


-- exercitiul 8
    FUNCTION functie_8 (p_cinematograf_id IN NUMBER, p_dataproiectie IN DATE)
RETURN NUMBER
AS
  v_venit NUMBER(10,2);
  v_cinematograf NUMBER(10);
BEGIN
    --  tratare caz: exista cinematograful dat ca parametru?
```

```
    SELECT idcinematograf INTO v_cinematograf FROM cinematograf WHERE idcinematograf
= p_cinematograf_id;

    IF v_cinematograf IS NULL THEN
      RAISE_APPLICATION_ERROR(-20000, 'Nu exista cinematograful ' || v_cinematograf);
    END IF;

    SELECT SUM(pret * (1 - discount)) INTO v_venit
    FROM BILET t
    JOIN PROIECTIE p ON t.titlufilm = p.titlufilm AND t.idsala = p.idsala
    JOIN SALA h ON p.idsala = h.idsala
    WHERE h.idcinematograf = p_cinematograf_id AND p.dataproiectie = p_dataproiectie;

    -- tratare caz: nu exista proiectii pentru data mentionata
    IF v_venit IS NULL THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nu exista proiectii pentru data ' ||
p_dataproiectie);
    END IF;

    RETURN v_venit;
END;


-- exercitiul 9
PROCEDURE
       procedura_9(cod_cinema cinematograf.idcinematograf%TYPE) IS
    cnume cinematograf.numecinematograf%TYPE;
     CURSOR c1 IS
        SELECT email n, regizor t, idcinematograf cid2, dataproiectie dd
        FROM client sp JOIN bilet b ON (sp.numeclient=b.numeclient)
                       JOIN film f ON (b.titlufilm=f.titlufilm)
                       JOIN sala s ON (b.idsala=s.idsala)
                       join proiectie p on (p.titlufilm=f.titlufilm)
        WHERE idcinematograf=cod_cinema;
            TYPE_MISMATCH EXCEPTION;
         INVALID_INPUT EXCEPTION;

  BEGIN

    IF cod_cinema IS NULL THEN
        RAISE INVALID_INPUT;
    END IF;


        SELECT c.numecinematograf
        INTO cnume
        FROM cinematograf c
        WHERE idcinematograf=cod_cinema;
        DBMS_OUTPUT.PUT_LINE('Cinematograful ' || cnume ||' a avut urmatorii clienti:
');
        DBMS_OUTPUT.NEW_LINE();

        FOR i in c1 LOOP
            DBMS_OUTPUT.PUT_LINE(i.n || ' '  || ' care a vizionat filmul regizorului
' || i.t || ' la data de ' || i.dd);
```

```
        END LOOP;
        DBMS_OUTPUT.NEW_LINE();

    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista niciun cinematograf cu acest ID' );
            RETURN;
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai mult de un cinematograf cu acelasi ID');
            RETURN;
        WHEN INVALID_INPUT THEN
            DBMS_OUTPUT.PUT_LINE('ID-ul cinematografului nu poate fi NULL');
            RETURN;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Alta eroare!');
end procedura_9;

END pachet_ex13;
/
-- testare pachet
BEGIN
    pachet_ex13.colectie_ex6('Drama');
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.cursor_ex7('Interstellar');
    DBMS_OUTPUT.PUT_LINE('-----------');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.functie_8(6, '24-DEC-22'));
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.procedura_9(6);

end;
```





```
CREATE OR REPLACE PACKAGE pachet_ex13 AS
    PROCEDURE colectie_ex6(gen_param in varchar2);
    PROCEDURE cursor_ex7 (film_titlu IN VARCHAR2);
    FUNCTION functie_8 (p_cinematograf_id IN NUMBER, p_dataproiectie IN DATE) RETURN NUMBER;
    PROCEDURE procedura_9(cod_cinema cinematograf.idcinematograf%TYPE);
END pachet_ex13;
/
CREATE OR REPLACE PACKAGE BODY pachet_ex13 AS

--  ex 6
    PROCEDURE colectie_ex6(gen_param in varchar2)
IS

    TYPE toate filmele is table of film%rowtype;
```

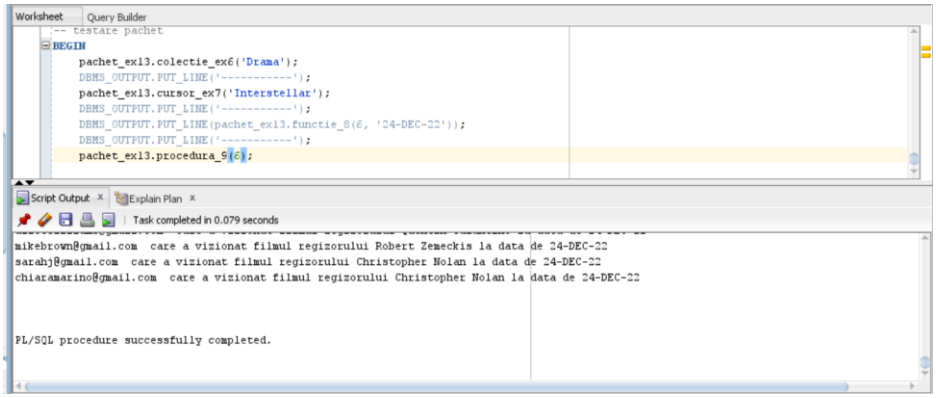
```
Script Output | Explain Plan
Task completed in 0.196 seconds

Package Body PACHET_EX13 compiled
```

```
-- testare pachet
BEGIN
    pachet_ex13.colectie_ex6('Drama');
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.cursor_ex7('Interstellar');
    DBMS_OUTPUT.PUT_LINE('-----------');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.functie_8(6, '24-DEC-22'));
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.procedura_9(6);
```

```
Script Output | Explain Plan
Task completed in 0.079 seconds

Forrest Gump
-----------
Bilete vandute pentru Interstellar: 2
Venit total al filmului Interstellar: $20.49
Filme sortate dupa venit:
The Godfather: Part II: $27.99
Avatar: $27.99
Blade Runner: $20.49
Inception: $20.49
```

```
-- testare pachet
BEGIN
    pachet_ex13.colectie_ex6('Drama');
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.cursor_ex7('Interstellar');
    DBMS_OUTPUT.PUT_LINE('-----------');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.functie_8(6, '24-DEC-22'));
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.procedura_9(6);
```

```
Script Output | Explain Plan
Task completed in 0.079 seconds

Alien: $7.5
The Dark Knight: $7.5
The Shawshank Redemption: $6.375
Forrest Gump: $6.375
-----------
42.37
-----------
Cinematograful Broadway Cinema a avut urmatorii clienti:
```



```
Worksheet | Query Builder
-- testare pachet
BEGIN
    pachet_ex13.colectie_ex6('Drama');
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.cursor_ex7('Interstellar');
    DBMS_OUTPUT.PUT_LINE('-----------');
    DBMS_OUTPUT.PUT_LINE(pachet_ex13.functie_8(6, '24-DEC-22'));
    DBMS_OUTPUT.PUT_LINE('-----------');
    pachet_ex13.procedura_9(6);
```

```
Script Output | Explain Plan
Task completed in 0.079 seconds

mikebrown@gmail.com  care a vizionat filmul regizorului Robert Zemeckis la data de 24-DEC-22
sarahj@gmail.com  care a vizionat filmul regizorului Christopher Nolan la data de 24-DEC-22
chiaramarino@gmail.com  care a vizionat filmul regizorului Christopher Nolan la data de 24-DEC-22


PL/SQL procedure successfully completed.
```