PROIECT BAZE DE DATE

Fănică Narcis-Alexandru - 244

# **Gestionare Netflix**

**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

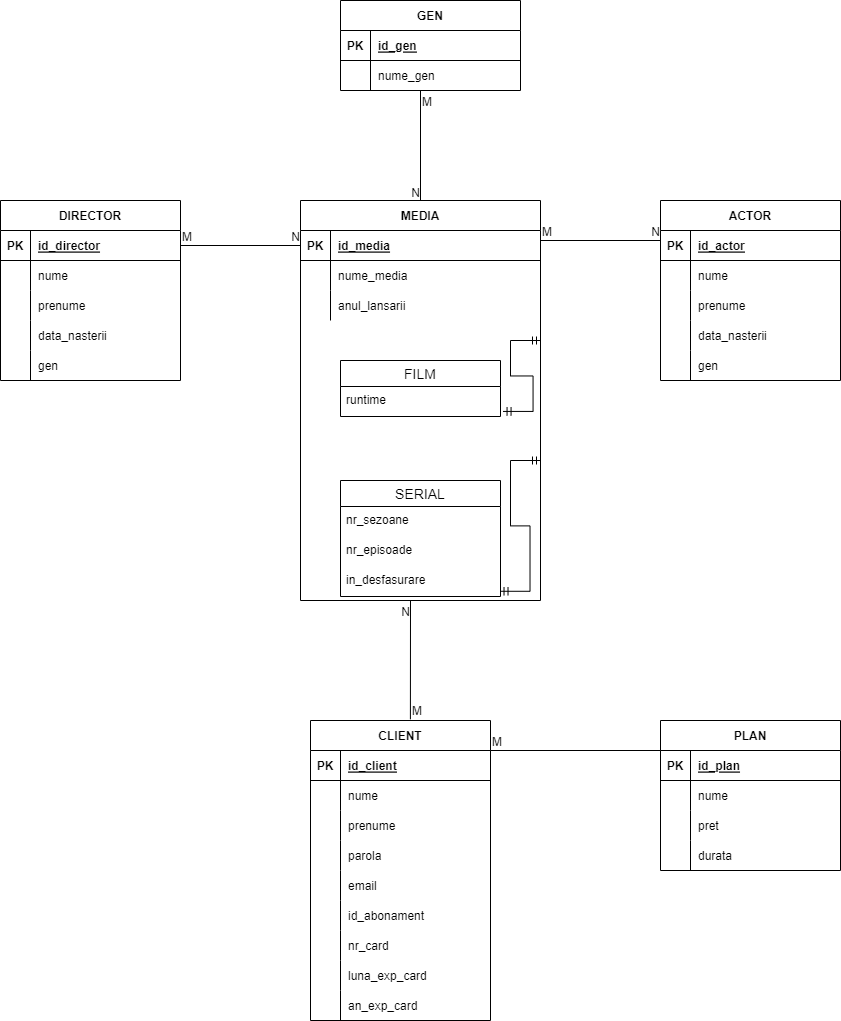
Netflix reprezintă un serviciu de transmitere al filmelor și serialelor contra costul unui abonament plătit lunar. În baza acestui abonament, pot fi redate oricâte filme și seriale se află la un moment dat pe platformă. Filmele au o durată determinată exprimată în minute, în timp ce serialele au un număr variabil de sezoane și episoade la un moment dat. Acestea pot fi incheiate, însemnând că nu vor mai fi lansate noi episoade sau pot fi în desfășurare, ceea ce înseamnă că noi episoade pot apărea periodic.

Toate filmele și serialele au unul sau mai mulți actori precum și unul sau mai mulți directori ce au lucrat la regizarea acestora.

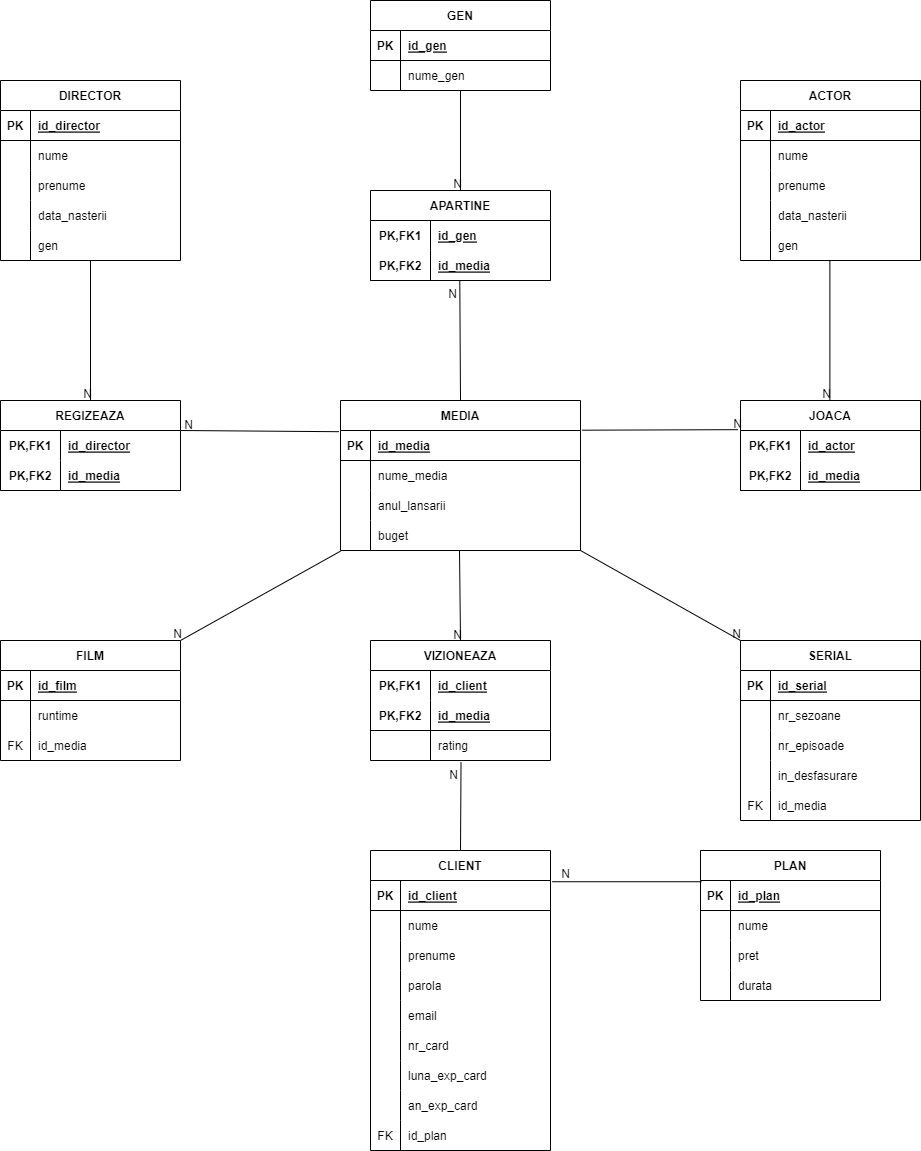
În plus, fiecare film și serial apartine unuia sau mai multor genuri.

Pentru a avea acces la aceste filme și seriale, clienții sunt nevoiți să își creeze un abonament lunar în baza unui plan pentru care vor plăti o anumită sumă predeterminată în funcție de abonamentul ales și de facilitățile acestuia.

**2.Realizați diagrama entitate-relație (ERD).**

****

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

****

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

**CREATE TABLE PLAN**

**(**

**id\_plan NUMBER(2) PRIMARY KEY,**

**nume VARCHAR(100) NOT NULL,**

**pret NUMBER(3) NOT NULL,**

**durata NUMBER(3) NOT NULL**

**);**

**CREATE TABLE CLIENTI**

**(**

**id\_client NUMBER(5) PRIMARY KEY,**

**nume VARCHAR(50) NOT NULL,**

**prenume VARCHAR(50) NOT NULL,**

**parola VARCHAR(100) NOT NULL,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**nr\_card NUMBER(16) NOT NULL,**

**an\_exp\_card NUMBER(4) NOT NULL,**

**luna\_exp\_card NUMBER(2) NOT NULL,**

**id\_plan NUMBER(2),**

**CONSTRAINT fk\_plan FOREIGN KEY (id\_plan) REFERENCES PLAN(id\_plan)**

**);**

**CREATE TABLE MEDIA**

**(**

**id\_media NUMBER(5) PRIMARY KEY,**

**nume\_media VARCHAR(100) NOT NULL,**

**anul\_lansarii NUMBER(4),**

**buget NUMBER(10)**

**);**

**CREATE TABLE FILME**

**(**

**id\_film NUMBER(5) PRIMARY KEY,**

**runtime NUMBER(3),**

**CONSTRAINT fk\_film FOREIGN KEY(id\_film) REFERENCES MEDIA(id\_media)**

**);**

**CREATE TABLE SERIALE**

**(**

**id\_serial NUMBER(5) PRIMARY KEY,**

**nr\_sezoane NUMBER(3),**

**nr\_episoade NUMBER(4),**

**in\_desfasurare VARCHAR(1),**

**CONSTRAINT fk\_serial FOREIGN KEY(id\_serial) REFERENCES MEDIA(id\_media)**

**);**

**CREATE TABLE GEN**

**(**

**id\_gen NUMBER(3) PRIMARY KEY,**

**nume\_gen VARCHAR(100) NOT NULL**

**);**

**CREATE TABLE ACTORI**

**(**

**id\_actor NUMBER(5) PRIMARY KEY,**

**nume VARCHAR(100) NOT NULL,**

**prenume VARCHAR(100) NOT NULL,**

**data\_nasterii DATE,**

**gen VARCHAR(1)**

**);**

**CREATE TABLE DIRECTOR**

**(**

**id\_director NUMBER(5) PRIMARY KEY,**

**nume VARCHAR(100) NOT NULL,**

**prenume VARCHAR(100) NOT NULL,**

**data\_nasterii DATE,**

**gen VARCHAR(1)**

**);**

**CREATE TABLE VIZIONEAZA**

**(**

**id\_client NUMBER(5),**

**id\_media NUMBER(5),**

**rating NUMBER(3, 1),**

**CONSTRAINT pk\_vizioneaza PRIMARY KEY (id\_client, id\_media),**

**CONSTRAINT fk\_vizioneaza\_client FOREIGN KEY (id\_client) REFERENCES CLIENTI(id\_client),**

**CONSTRAINT fk\_vizioneaza\_media FOREIGN KEY (id\_media) REFERENCES MEDIA(id\_media)**

**);**

**CREATE TABLE JOACA**

**(**

**id\_actor NUMBER(5),**

**id\_media NUMBER(5),**

**CONSTRAINT pk\_joaca PRIMARY KEY (id\_actor, id\_media),**

**CONSTRAINT fk\_joaca\_actor FOREIGN KEY (id\_actor) REFERENCES ACTORI(id\_actor),**

**CONSTRAINT fk\_joaca\_media FOREIGN KEY (id\_media) REFERENCES MEDIA(id\_media)**

**);**

**CREATE TABLE REGIZEAZA**

**(**

**id\_director NUMBER(5),**

**id\_media NUMBER(5),**

**CONSTRAINT pk\_regizeaza PRIMARY KEY (id\_director, id\_media),**

**CONSTRAINT fk\_regizeaza\_director FOREIGN KEY (id\_director) REFERENCES DIRECTOR(id\_director),**

**CONSTRAINT fk\_regizeaza\_media FOREIGN KEY (id\_media) REFERENCES MEDIA(id\_media)**

**);**

**CREATE TABLE APARTINE**

**(**

**id\_media NUMBER(5),**

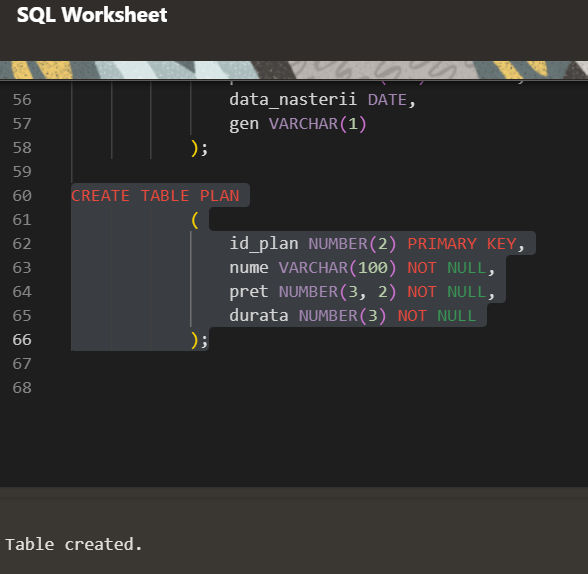
**id\_gen NUMBER(3),**

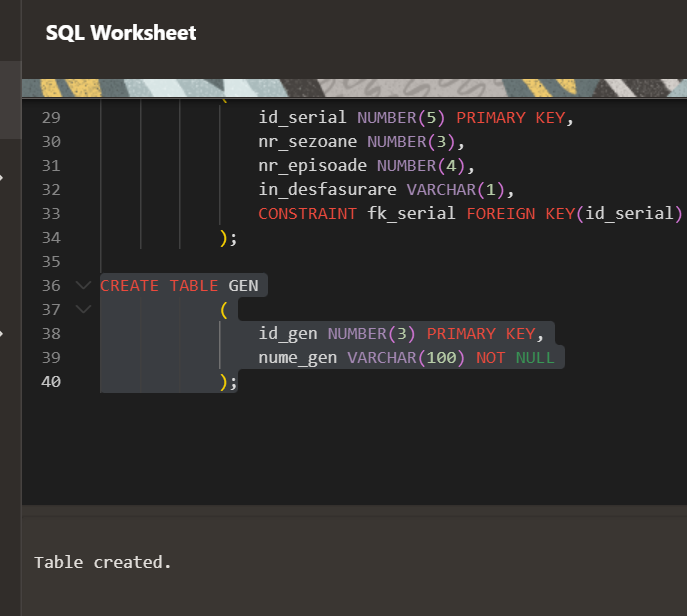
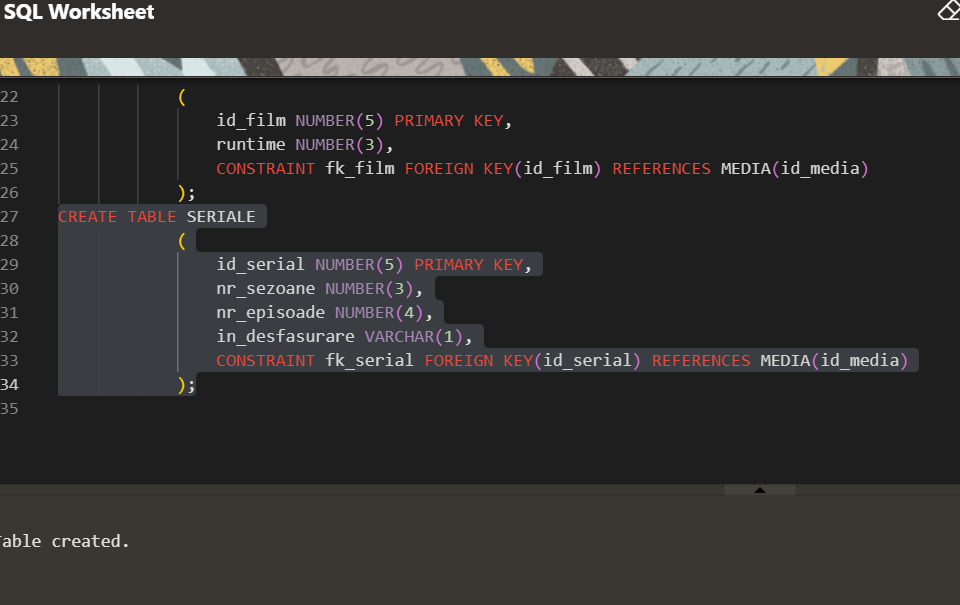
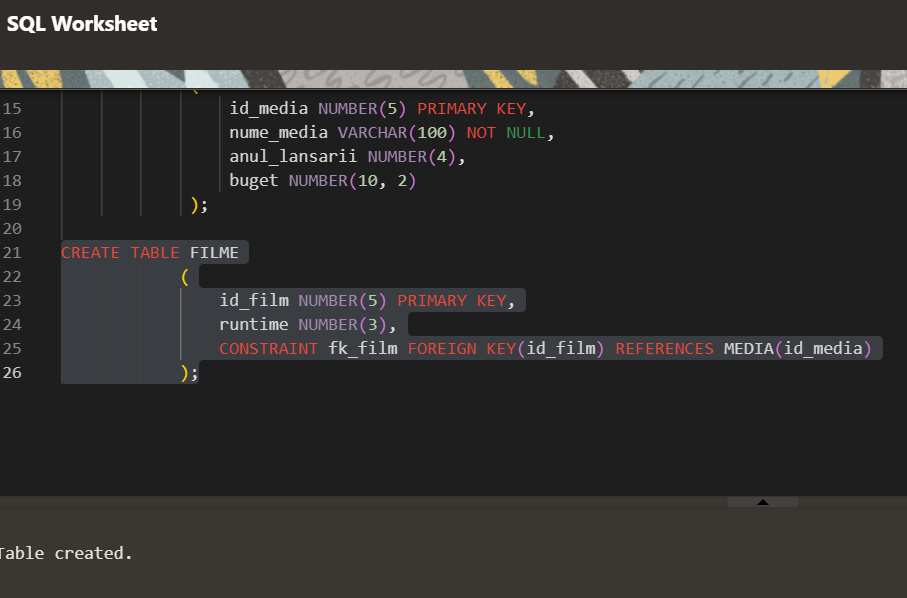
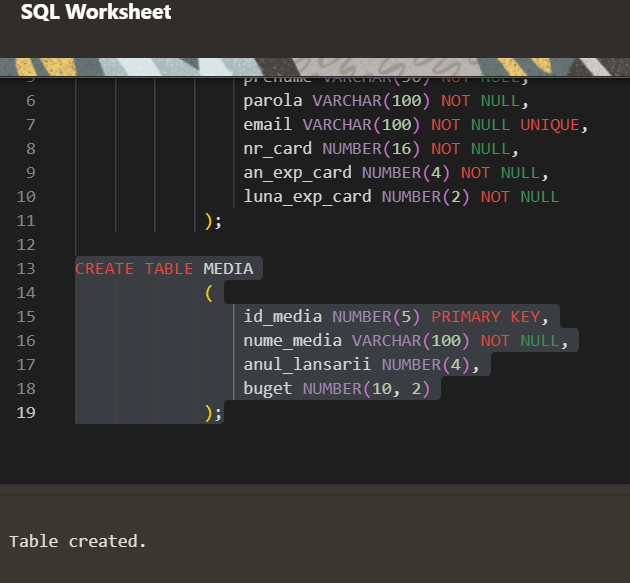
**CONSTRAINT pk\_apartine PRIMARY KEY (id\_gen, id\_media),**

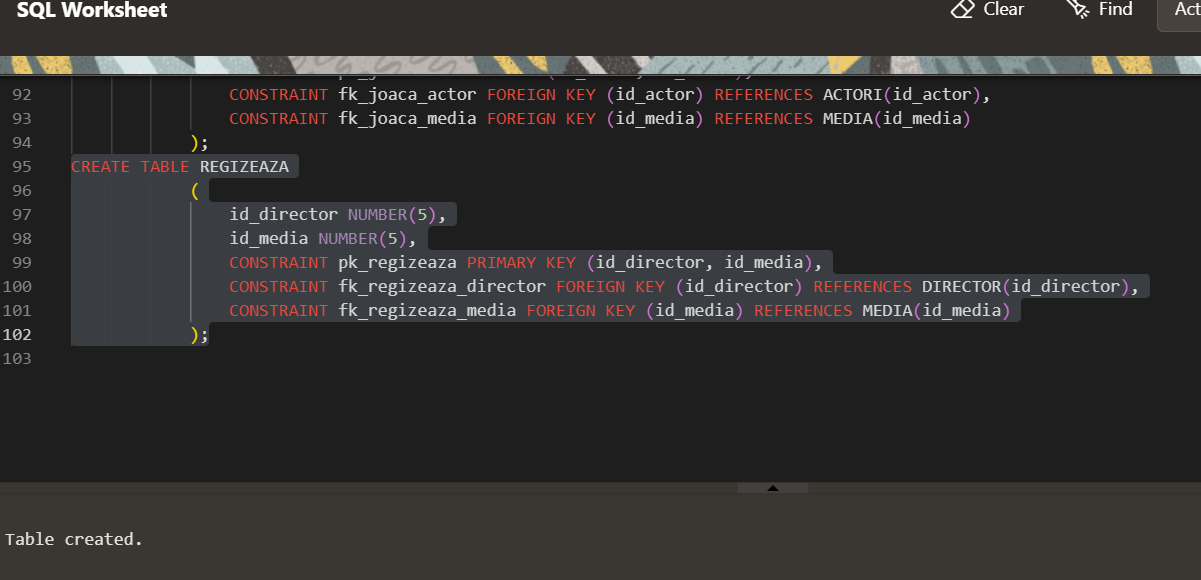
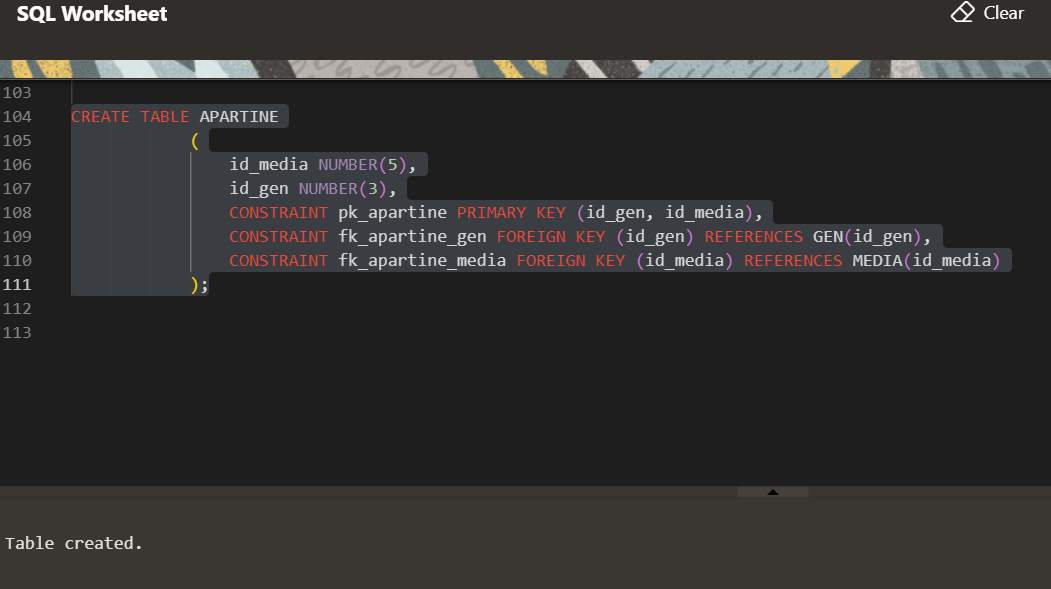
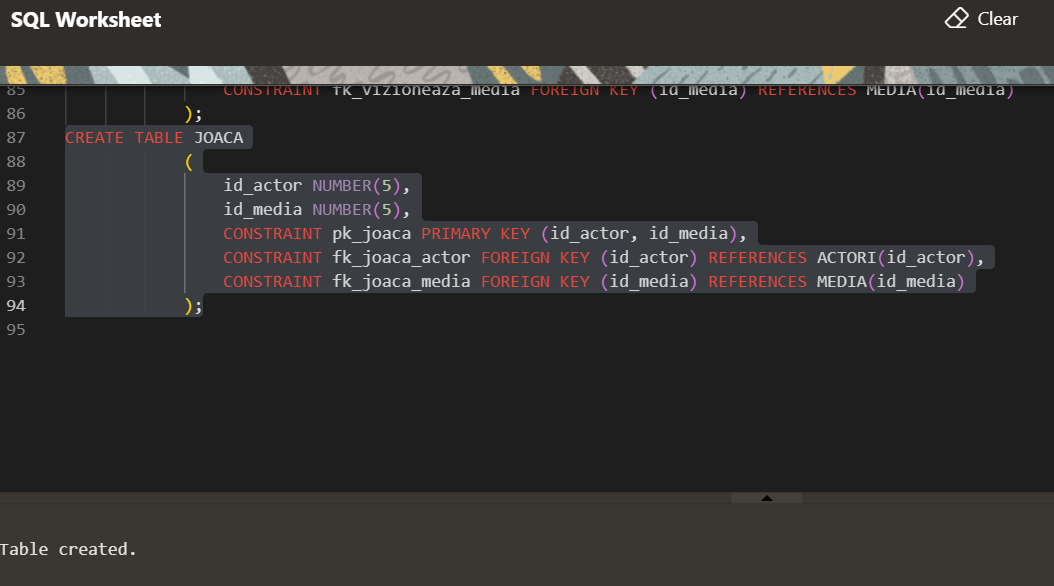
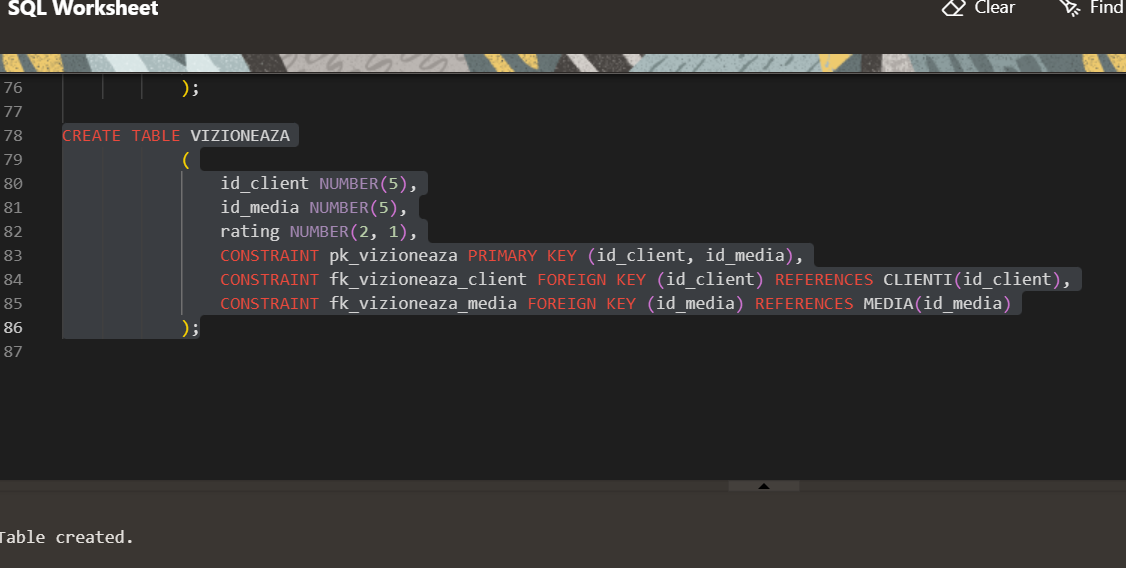
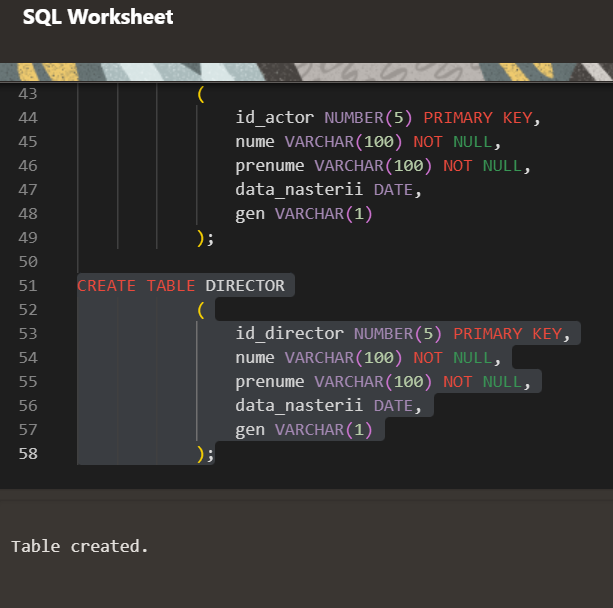
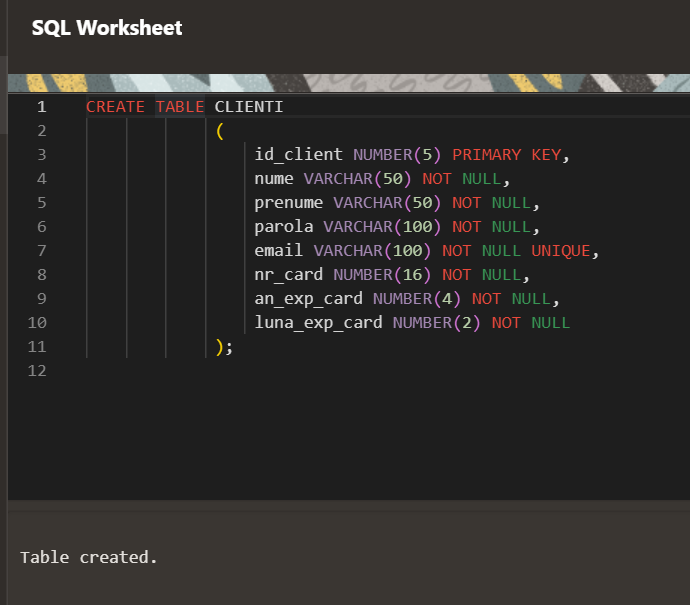
**CONSTRAINT fk\_apartine\_gen FOREIGN KEY (id\_gen) REFERENCES GEN(id\_gen),**

**CONSTRAINT fk\_apartine\_media FOREIGN KEY (id\_media) REFERENCES MEDIA(id\_media)**

**);**

****

****

****

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

**INSERT INTO PLAN**

**VALUES (1, 'Bronze', 40, 30);**

**INSERT INTO PLAN**

**VALUES (2, 'Silver', 45, 30);**

**INSERT INTO PLAN**

**VALUES (3, 'Gold', 50, 30);**

**INSERT INTO PLAN**

**VALUES (4, 'Platinum', 80, 60);**

**INSERT INTO PLAN**

**VALUES (5, 'Diamond', 110, 60);**

**INSERT INTO CLIENTI**

**VALUES (1, 'Popescu', 'Ion', 'parolapop', 'pop@mail.com', 4689190009801234, 2025, 9, 1);**

**INSERT INTO CLIENTI**

**VALUES (2, 'Andrei', 'Stefan', 'pstf', 'stefan@mail.com', 4689696019901234, 2026, 1, 1);**

**INSERT INTO CLIENTI**

**VALUES (3, 'Ion', 'Andrei', 'paarolamea', 'and@mail.com', 4689190009804567, 2025, 7, 3);**

**INSERT INTO CLIENTI**

**VALUES (4, 'Radu', 'Razvan', 'razvanpar123', 'razvan@mmail.com', 4689190010924567, 2023, 2, 1);**

**INSERT INTO CLIENTI**

**VALUES (5, 'Rica', 'Vlad', 'vald45r3', 'vald@mail.com', 4689190451001234, 2024, 7, 1);**

**INSERT INTO CLIENTI**

**VALUES (6, 'Popescu', 'Vladimir', 'popvld23', 'idk@mail.com', 2934190009801234, 2023, 10, 2);**

**INSERT INTO CLIENTI**

**VALUES (7, 'Radu', 'Calin', 'radurec00', 'calin.radu@mail.com', 4689988709801234, 2027, 1, 4);**

**INSERT INTO CLIENTI**

**VALUES (8, 'Dumitrescu', 'Denisa', 'denisadu45', 'denis@mail.com', 7665190009801234, 2026, 11, 4);**

**INSERT INTO CLIENTI**

**VALUES (9, 'Dumitriu', 'Fabian', 'cabala11', 'fab@mail.com', 4987190009801234, 2024, 4, 5);**

**INSERT INTO CLIENTI**

**VALUES (10, 'Andrei', 'Calin', 'calinpidk', 'wallhacker@mail.com', 9633490009801234, 2027, 10, 3);**

**INSERT INTO GEN**

**VALUES(1, 'Actiune');**

**INSERT INTO GEN**

**VALUES(2, 'Aventure');**

**INSERT INTO GEN**

**VALUES(3, 'Comedie');**

**INSERT INTO GEN**

**VALUES(4, 'Drama');**

**INSERT INTO GEN**

**VALUES(5, 'Thriller');**

**INSERT INTO GEN**

**VALUES(6, 'Mister');**

**INSERT INTO ACTORI**

**VALUES(1, 'Pitt', 'Brad', TO\_DATE('18-12-1963', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(2, 'Freeman', 'Morgan', TO\_DATE('01-06-1937', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(3, 'DiCaprio', 'Leonardo', TO\_DATE('08-11-1970', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(4, 'De Niro', 'Robert', TO\_DATE('17-09-1943', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(5, 'Damon', 'Matt', TO\_DATE('08-10-1970', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(6, 'Caine', 'Michael', TO\_DATE('10-10-1945', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(7, 'Portman', 'Natalie', TO\_DATE('09-06-1981', 'DD-MM-YYYY'), 'F');**

**INSERT INTO ACTORI**

**VALUES(8, 'Blanchett', 'Cate', TO\_DATE('14-05-1969', 'DD-MM-YYYY'), 'F');**

**INSERT INTO ACTORI**

**VALUES(9, 'Hathaway', 'Anne', TO\_DATE('18-10-1985', 'DD-MM-YYYY'), 'F');**

**INSERT INTO ACTORI**

**VALUES(10, 'Shire', 'Talia', TO\_DATE('10-04-1961', 'DD-MM-YYYY'), 'F');**

**INSERT INTO ACTORI**

**VALUES(11, 'PAcino', 'Al', TO\_DATE('25-04-1940', 'DD-MM-YYYY'), 'M');**

**INSERT INTO ACTORI**

**VALUES(12, 'Shannon', 'Michael', TO\_DATE('07-09-1974', 'DD-MM-YYYY'), 'M');**

**INSERT INTO DIRECTOR**

**VALUES(1, 'Nolan', 'Christopher', TO\_DATE('04-09-1964', 'DD-MM-YYYY'), 'M');**

**INSERT INTO DIRECTOR**

**VALUES(2, 'Tarantino', 'Quentin', TO\_DATE('10-12-1945', 'DD-MM-YYYY'), 'M');**

**INSERT INTO DIRECTOR**

**VALUES(3, 'Scorsese', 'Martin', TO\_DATE('09-03-1949', 'DD-MM-YYYY'), 'M');**

**INSERT INTO DIRECTOR**

**VALUES(4, 'Spielberg', 'Stever', TO\_DATE('06-07-1950', 'DD-MM-YYYY'), 'M');**

**INSERT INTO DIRECTOR**

**VALUES(5, 'Zhao', 'Chloe', TO\_DATE('20-03-1985', 'DD-MM-YYYY'), 'F');**

**INSERT INTO DIRECTOR**

**VALUES(6, 'Ducournau', 'Julia', TO\_DATE('15-11-1986', 'DD-MM-YYYY'), 'F');**

**INSERT INTO MEDIA**

**VALUES (1, 'Wednesday', 2022, 75000000);**

**INSERT INTO MEDIA**

**VALUES (2, 'Batman Begins', 2005, 120000000);**

**INSERT INTO MEDIA**

**VALUES (3, 'The Dark Knight', 2008, 180000000);**

**INSERT INTO MEDIA**

**VALUES (4, 'The Dark Knight Rises', 2012, 195000000);**

**INSERT INTO MEDIA**

**VALUES (5, 'Peaky Blinders', 2013, 60000000);**

**INSERT INTO MEDIA**

**VALUES (6, 'Arrow', 2012, 50000000);**

**INSERT INTO MEDIA**

**VALUES (7, 'The Avengers', 2012, 220000000);**

**INSERT INTO MEDIA**

**VALUES (8, 'The Flash', 2014, 70000000);**

**INSERT INTO MEDIA**

**VALUES (9, 'A New Hope', 1977, 50000000);**

**INSERT INTO MEDIA**

**VALUES (10, 'The Empire Strikes Back', 1980, 72000000);**

**INSERT INTO MEDIA**

**VALUES (11, 'Loki', 2021, 100000000);**

**INSERT INTO MEDIA**

**VALUES (12, 'Wandavision', 2020, 110000000);**

**INSERT INTO FILME**

**VALUES (2, 134);**

**INSERT INTO FILME**

**VALUES (3, 145);**

**INSERT INTO FILME**

**VALUES (4, 151);**

**INSERT INTO FILME**

**VALUES (7, 139);**

**INSERT INTO FILME**

**VALUES (9, 143);**

**INSERT INTO FILME**

**VALUES (10, 124);**

**INSERT INTO SERIALE**

**VALUES (1, 1, 10, 'Y');**

**INSERT INTO SERIALE**

**VALUES (5, 6, 36, 'N');**

**INSERT INTO SERIALE**

**VALUES (6, 8, 123, 'N');**

**INSERT INTO SERIALE**

**VALUES (8, 9, 145, 'Y');**

**INSERT INTO SERIALE**

**VALUES (11, 1, 8, 'N');**

**INSERT INTO SERIALE**

**VALUES (12, 1, 8, 'N');**

**INSERT INTO APARTINE**

**VALUES (1, 1);**

**INSERT INTO APARTINE**

**VALUES (2, 3);**

**INSERT INTO APARTINE**

**VALUES (2, 4);**

**INSERT INTO APARTINE**

**VALUES (3, 5);**

**INSERT INTO APARTINE**

**VALUES (3, 1);**

**INSERT INTO APARTINE**

**VALUES (4, 2);**

**INSERT INTO APARTINE**

**VALUES (5, 6);**

**INSERT INTO APARTINE**

**VALUES (6, 5);**

**INSERT INTO APARTINE**

**VALUES (7, 1);**

**INSERT INTO APARTINE**

**VALUES (8, 3);**

**INSERT INTO APARTINE**

**VALUES (9, 1);**

**INSERT INTO APARTINE**

**VALUES (9, 2);**

**INSERT INTO APARTINE**

**VALUES (10, 3);**

**INSERT INTO APARTINE**

**VALUES (10, 4);**

**INSERT INTO APARTINE**

**VALUES (10, 5);**

**INSERT INTO APARTINE**

**VALUES (11, 1);**

**INSERT INTO APARTINE**

**VALUES (11, 4);**

**INSERT INTO JOACA**

**VALUES (1, 2);**

**INSERT INTO JOACA**

**VALUES (1, 4);**

**INSERT INTO JOACA**

**VALUES (2, 4);**

**INSERT INTO JOACA**

**VALUES (1, 6);**

**INSERT INTO JOACA**

**VALUES (3, 7);**

**INSERT INTO JOACA**

**VALUES (4, 1);**

**INSERT INTO JOACA**

**VALUES (4, 6);**

**INSERT INTO JOACA**

**VALUES (5, 6);**

**INSERT INTO JOACA**

**VALUES (5, 2);**

**INSERT INTO JOACA**

**VALUES (5, 3);**

**INSERT INTO JOACA**

**VALUES (5, 4);**

**INSERT INTO JOACA**

**VALUES (6, 1);**

**INSERT INTO JOACA**

**VALUES (6, 2);**

**INSERT INTO JOACA**

**VALUES (7, 4);**

**INSERT INTO JOACA**

**VALUES (7, 8);**

**INSERT INTO JOACA**

**VALUES (8, 8);**

**INSERT INTO JOACA**

**VALUES (8, 1);**

**INSERT INTO JOACA**

**VALUES (8, 2);**

**INSERT INTO JOACA**

**VALUES (9, 4);**

**INSERT INTO JOACA**

**VALUES (9, 8);**

**INSERT INTO JOACA**

**VALUES (9, 9);**

**INSERT INTO JOACA**

**VALUES (10, 2);**

**INSERT INTO JOACA**

**VALUES (10, 6);**

**INSERT INTO JOACA**

**VALUES (10, 9);**

**INSERT INTO JOACA**

**VALUES (10, 10);**

**INSERT INTO REGIZEAZA**

**VALUES (1, 2);**

**INSERT INTO REGIZEAZA**

**VALUES (1, 4);**

**INSERT INTO REGIZEAZA**

**VALUES (2, 4);**

**INSERT INTO REGIZEAZA**

**VALUES (1, 6);**

**INSERT INTO REGIZEAZA**

**VALUES (3, 7);**

**INSERT INTO REGIZEAZA**

**VALUES (4, 1);**

**INSERT INTO REGIZEAZA**

**VALUES (4, 6);**

**INSERT INTO REGIZEAZA**

**VALUES (5, 6);**

**INSERT INTO REGIZEAZA**

**VALUES (5, 2);**

**INSERT INTO REGIZEAZA**

**VALUES (5, 3);**

**INSERT INTO REGIZEAZA**

**VALUES (5, 4);**

**INSERT INTO REGIZEAZA**

**VALUES (6, 1);**

**INSERT INTO REGIZEAZA**

**VALUES (6, 2);**

**INSERT INTO VIZIONEAZA**

**VALUES (1, 2, 9.5);**

**INSERT INTO VIZIONEAZA**

**VALUES (1, 4, 4);**

**INSERT INTO VIZIONEAZA**

**VALUES (2, 6, 10);**

**INSERT INTO VIZIONEAZA**

**VALUES (3, 9, null);**

**INSERT INTO VIZIONEAZA**

**VALUES (3, 12, 6);**

**INSERT INTO VIZIONEAZA**

**VALUES (4, 11, null);**

**INSERT INTO VIZIONEAZA**

**VALUES (4, 10, 9);**

**INSERT INTO VIZIONEAZA**

**VALUES (4, 5, 5.5);**

**INSERT INTO VIZIONEAZA**

**VALUES (7, 8, null);**

**INSERT INTO VIZIONEAZA**

**VALUES (7, 4, 5.5);**

**INSERT INTO VIZIONEAZA**

**VALUES (8, 1, 9.5);**

**INSERT INTO VIZIONEAZA**

**VALUES (10, 2, 10);**

**INSERT INTO VIZIONEAZA**

**VALUES (9, 6, null);**

**INSERT INTO VIZIONEAZA**

**VALUES (9, 7, 7);**

**INSERT INTO VIZIONEAZA**

**VALUES (9, 11, 9.5);**

**INSERT INTO VIZIONEAZA**

**VALUES (10, 12, 9);**

**INSERT INTO VIZIONEAZA**

**VALUES (10, 1, null);**

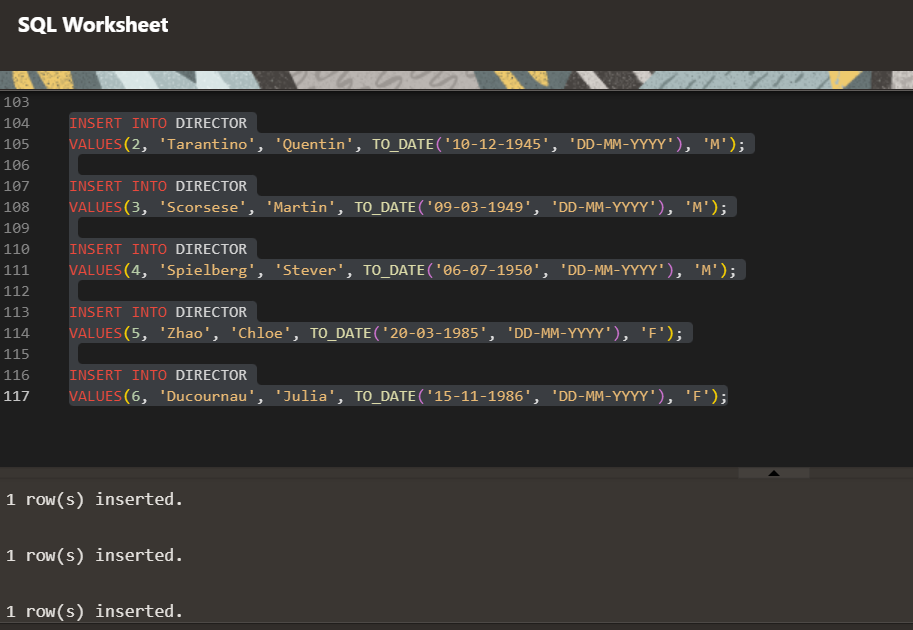
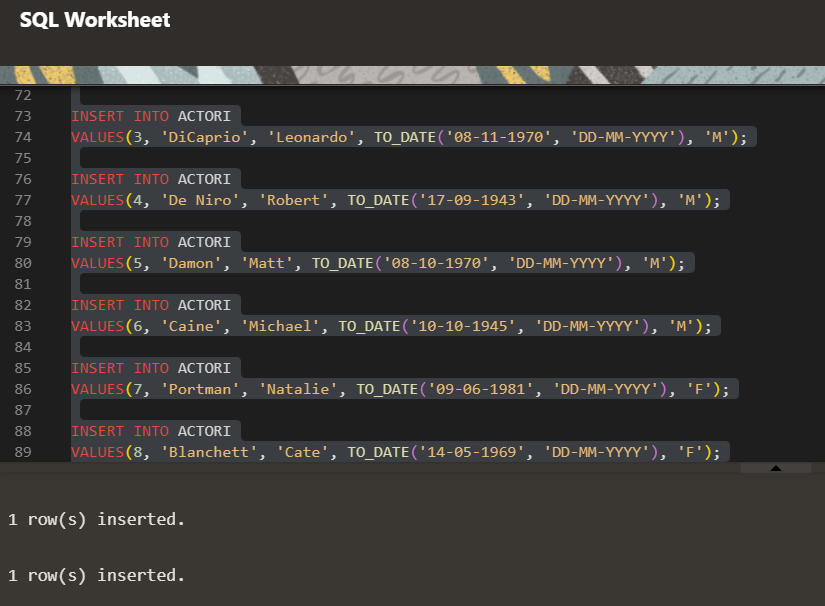
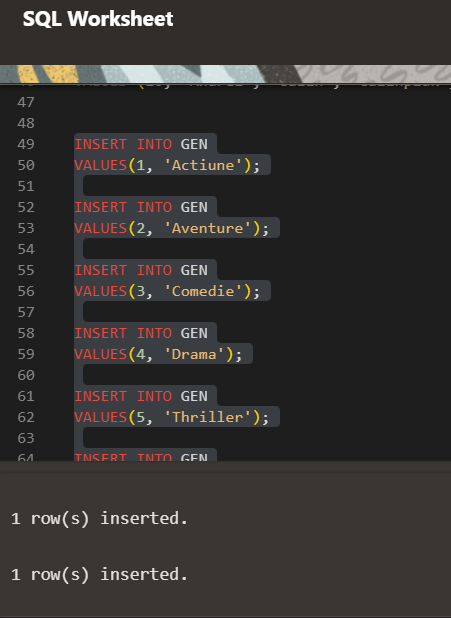
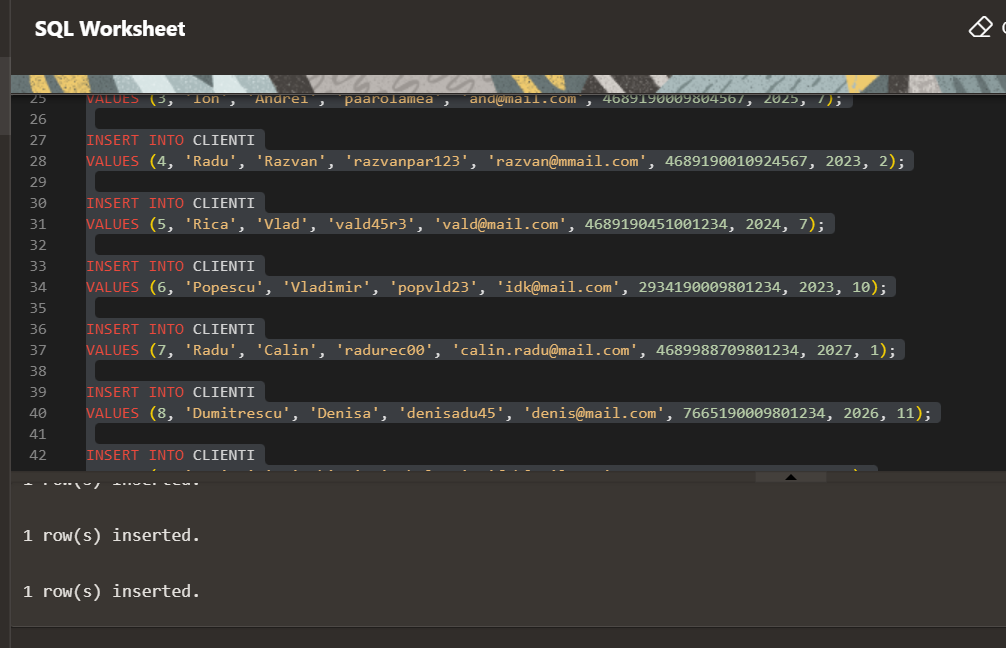
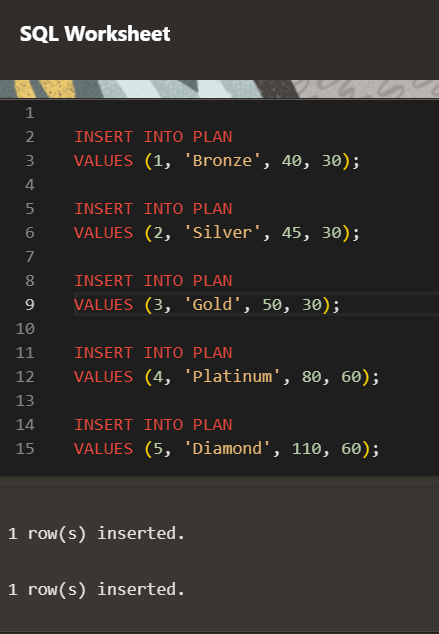
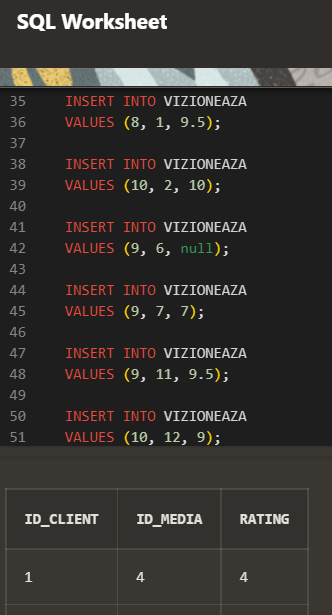
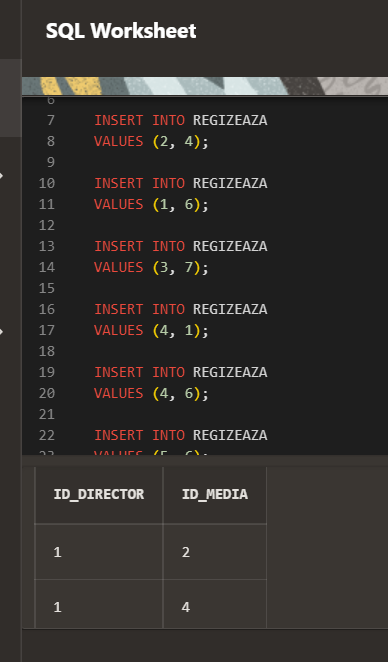
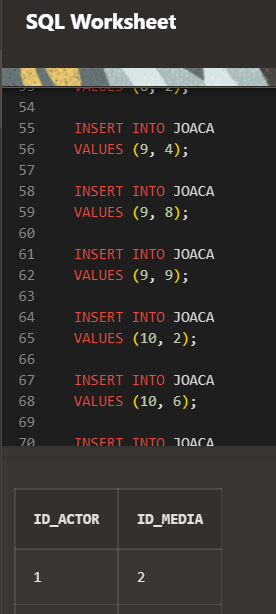
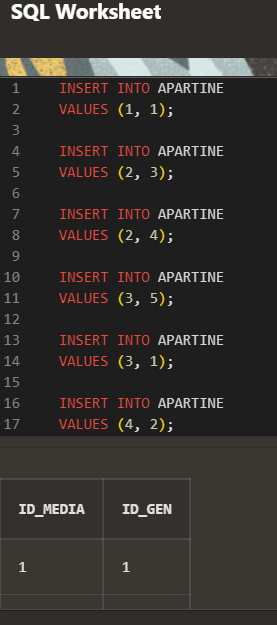
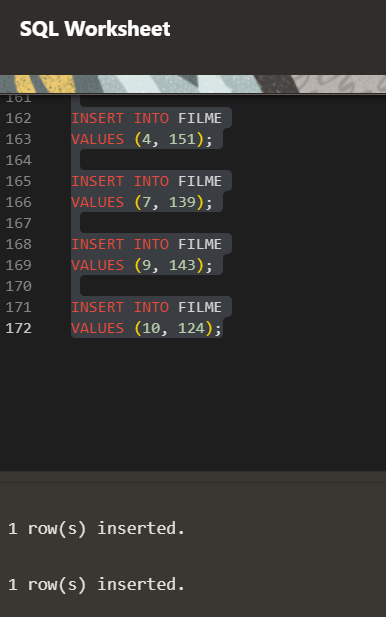
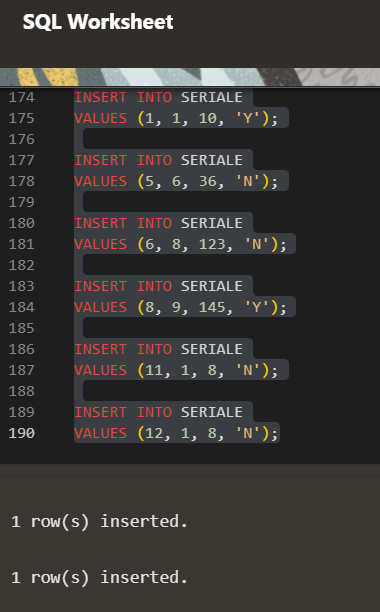
**INSERT INTO VIZIONEAZA**

**VALUES (6, 10, null);**

**INSERT INTO VIZIONEAZA**

**VALUES (6, 7, null);**

**SELECT \* FROM VIZIONEAZA;**

** **

**6.** **Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.**

**O procedura ce nu primeste parametri si va afisa pentru fiecare film fiecare recenzie primita si suma totala a acestora din punctajul maxim posibil.**

**CREATE OR REPLACE PROCEDURE procedura6 IS**

**TYPE tip\_pentru\_tablou\_indexat IS RECORD**

**(**

**nume VARCHAR(200),**

**rating NUMBER(3,1)**

**);**

**TYPE tip\_pentru\_tablou\_imbricat IS RECORD**

**(**

**id\_film NUMBER(5),**

**nume VARCHAR(200)**

**);**

**TYPE tablou\_indexat IS TABLE OF tip\_pentru\_tablou\_indexat INDEX BY PLS\_INTEGER;**

**TYPE tablou\_imbricat IS TABLE OF tip\_pentru\_tablou\_imbricat;**

**t\_filme tablou\_imbricat;**

**t\_recenzii tablou\_indexat;**

**numar NUMBER(4);**

**suma NUMBER(4);**

**BEGIN**

**SELECT id\_film, nume\_media**

**BULK COLLECT INTO t\_filme**

**FROM FILME JOIN MEDIA ON (FILME.id\_film = MEDIA.id\_media);**

**FOR i in t\_filme.FIRST..t\_filme.LAST LOOP**

**DBMS\_OUTPUT.PUT('FILM: ' || t\_filme(i).nume || ' ');**

**numar := 0;**

**suma := 0;**

**t\_recenzii.DELETE(1);**

**SELECT C.nume, V.rating**

**BULK COLLECT INTO t\_recenzii**

**FROM VIZIONEAZA V JOIN CLIENTI C ON (C.id\_client = V.id\_client)**

**WHERE V.id\_media = t\_filme(i).id\_film AND V.rating IS NOT NULL;**

**SELECT COUNT(rating) \* 10, SUM(rating)**

**INTO numar, suma**

**FROM VIZIONEAZA**

**WHERE id\_media = t\_filme(i).id\_film;**

**IF numar > 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE(suma || '/' || numar);**

**FOR j in t\_recenzii.FIRST..t\_recenzii.LAST LOOP**

**DBMS\_OUTPUT.PUT\_LINE(t\_recenzii(j).nume || ' ' || t\_recenzii(j).rating);**

**END LOOP;**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE('');**

**DBMS\_OUTPUT.PUT\_LINE('Acest film nu are nicio recenzie inca');**

**END IF;**

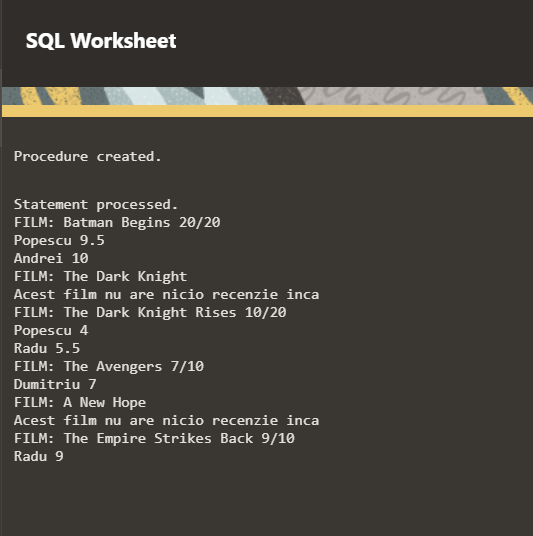
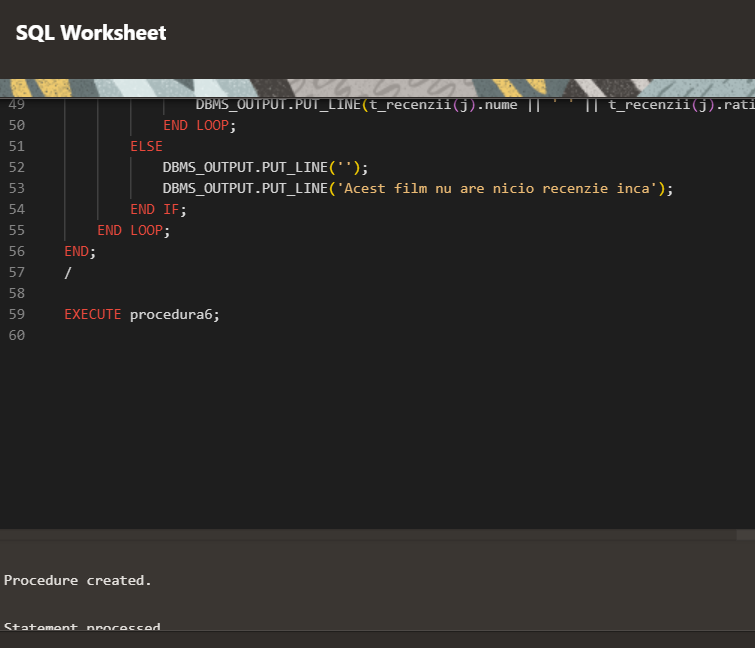
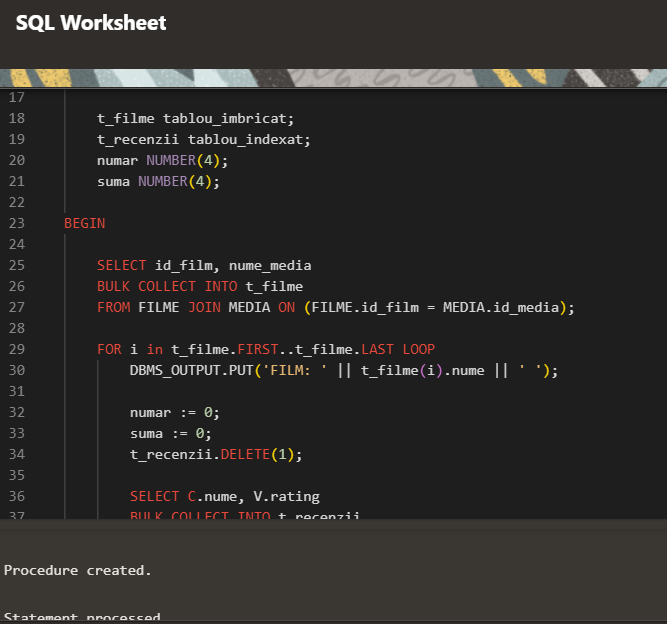
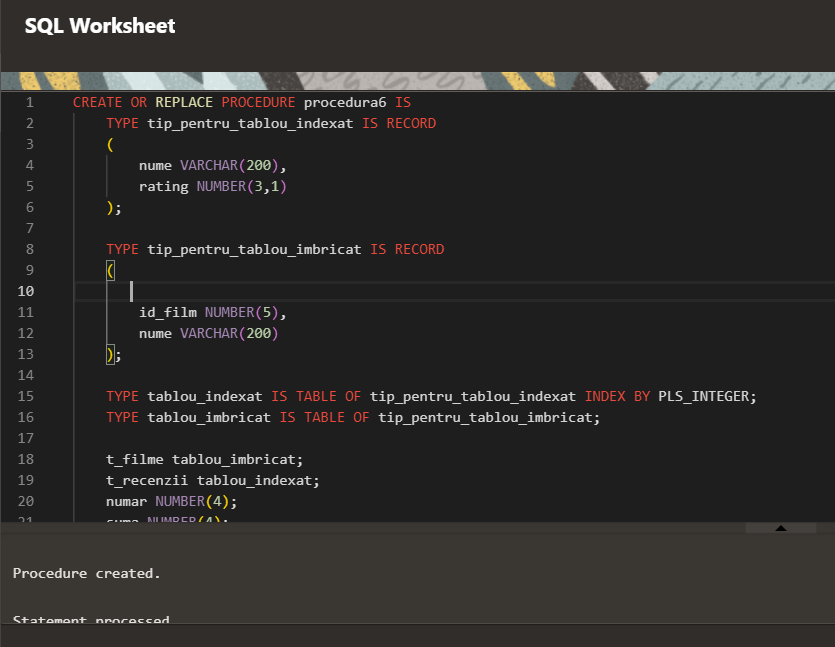
**END LOOP;**

**END;**

**/**

**EXECUTE procedura6;**

**/**



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

**O procedura ce primeste drept parametru numele unui gen de media si determina pentru toate filmele si serialele ce apartin genului respectiv numele fiecarui client ce l-a vizionat impreuna cu denumirea abonamentului sau.**

**CREATE OR REPLACE PROCEDURE procedura7 (p\_proc GEN.nume\_gen%TYPE) IS**

**CURSOR c IS**

**SELECT M.id\_media id\_media, M.nume\_media nume**

**FROM MEDIA M JOIN APARTINE A ON (A.id\_media = M.id\_media)**

**JOIN GEN G ON (G.id\_gen = A.id\_gen)**

**WHERE G.nume\_gen = p\_proc;**

**CURSOR d(p\_curs VIZIONEAZA.id\_media%TYPE) IS**

**SELECT C.nume nume, C.prenume prenume, P.nume nume\_abonament**

**FROM VIZIONEAZA V JOIN CLIENTI C ON (C.id\_client = V.id\_client)**

**JOIN PLAN P ON (P.id\_plan = C.id\_plan)**

**WHERE V.id\_media = p\_curs;**

**id\_pentru\_cursor MEDIA.id\_media%TYPE;**

**nume\_media MEDIA.nume\_media%TYPE;**

**numar\_media NUMBER :=0;**

**numar\_clienti NUMBER :=0;**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Pentru genul ' || p\_proc || ' : ');**

**OPEN c;**

**LOOP**

**FETCH c into id\_pentru\_cursor, nume\_media;**

**EXIT WHEN c%NOTFOUND;**

**numar\_media := numar\_media + 1;**

**DBMS\_OUTPUT.PUT\_LINE(nume\_media || ': ');**

**numar\_clienti :=0;**

**FOR i in d(id\_pentru\_cursor) LOOP**

**numar\_clienti := numar\_clienti + 1;**

**DBMS\_OUTPUT.PUT\_LINE(numar\_clienti || '. ' || i.prenume || ' ' || i.nume || ' cu abonament de tipul ' || i.nume\_abonament );**

**END LOOP;**

**IF numar\_clienti = 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE('Nimeni nu a vizionat acest film/serial.');**

**END IF;**

**END LOOP;**

**CLOSE c;**

**IF numar\_media = 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE('Nu exista filme/seriale ce apartin acestui gen.');**

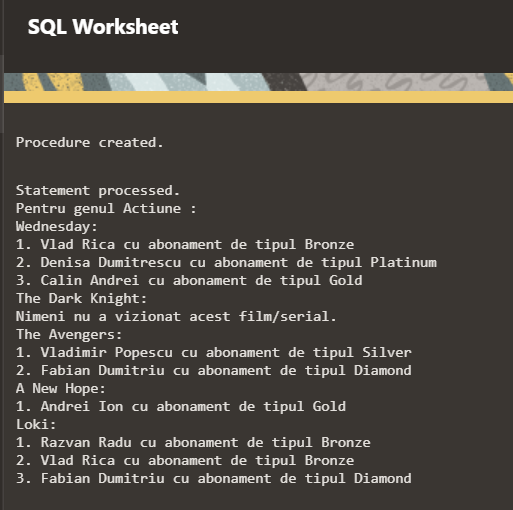
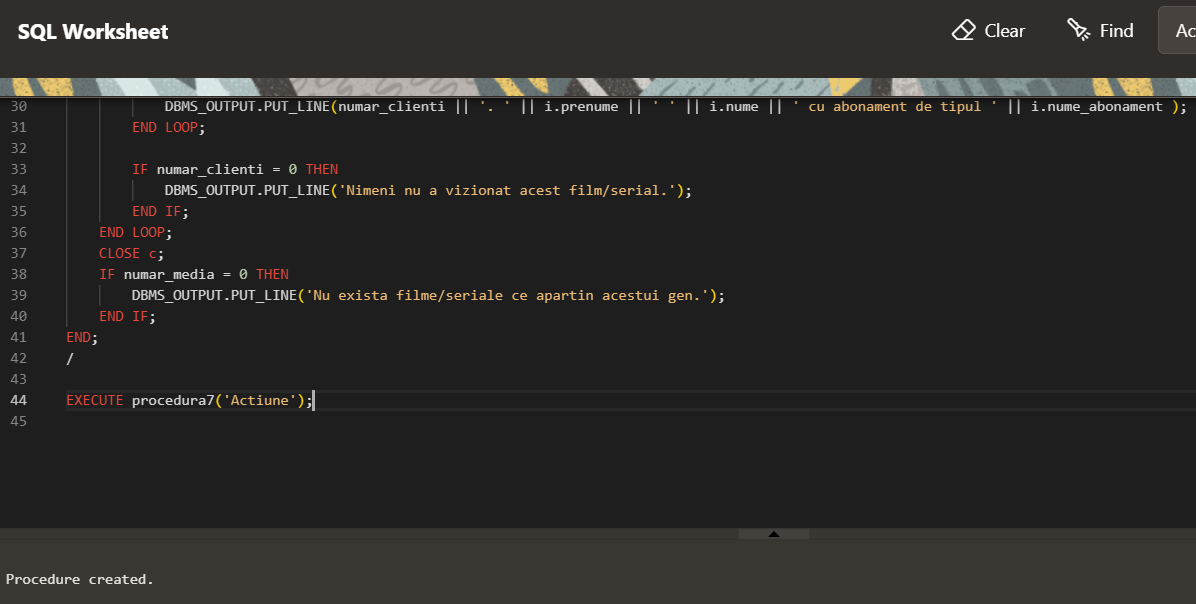
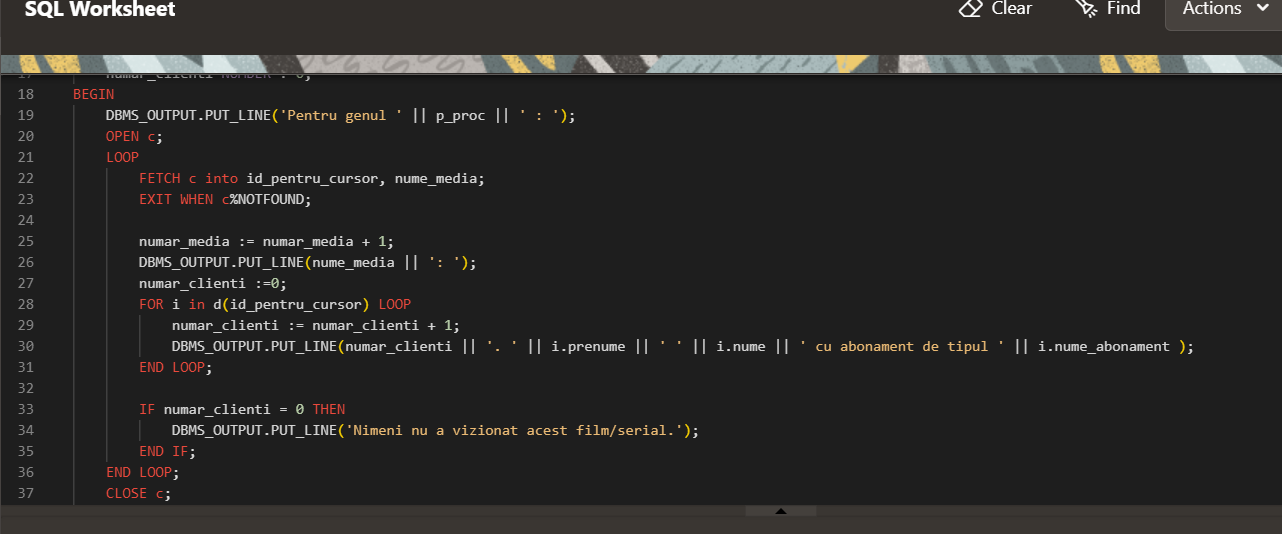
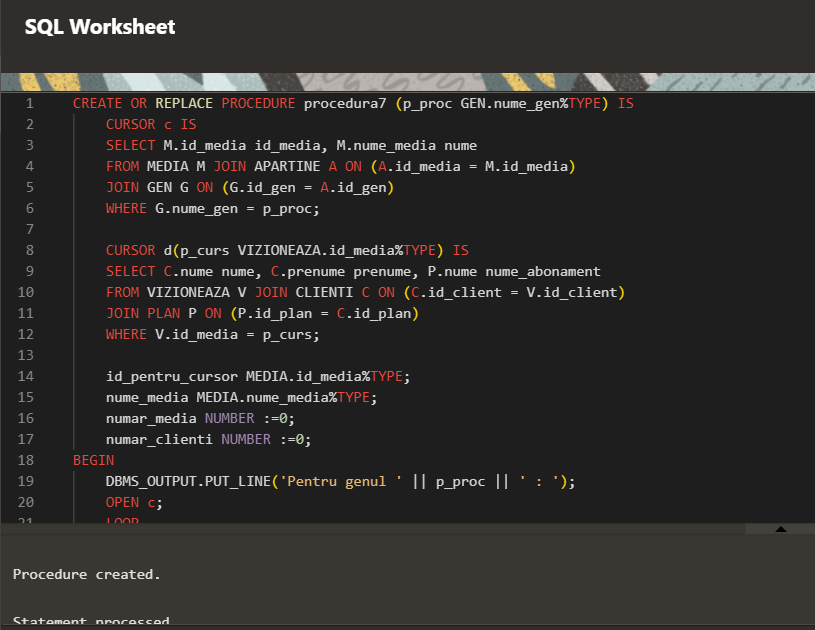
**END IF;**

**END;**

**/**

**EXECUTE procedura7('Actiune');**

**/**

****

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

**Functie ce va primi numele de familie al unui client si va intoarce bugetul total al tuturor filmelor si serialelor vizionate de acesta.**

**CREATE OR REPLACE FUNCTION functie8 (nume\_client CLIENTI.nume%TYPE)**

**RETURN NUMBER IS**

**numar\_clienti NUMBER;**

**numar\_media NUMBER;**

**buget\_total NUMBER;**

**buget\_film NUMBER;**

**NO\_CUSTOMERS EXCEPTION;**

**NO\_MEDIA EXCEPTION;**

**TOO\_MANY\_CUSTOMERS EXCEPTION;**

**id\_cl CLIENTI.id\_client%TYPE;**

**TYPE tablou\_imbricat IS TABLE OF VIZIONEAZA.id\_media%TYPE;**

**t tablou\_imbricat := tablou\_imbricat();**

**BEGIN**

**SELECT COUNT(\*)**

**INTO numar\_clienti**

**FROM CLIENTI**

**WHERE CLIENTI.nume = nume\_client;**

**IF numar\_clienti <= 0 THEN**

**RAISE NO\_CUSTOMERS;**

**END IF;**

**IF numar\_clienti > 1 THEN**

**RAISE TOO\_MANY\_CUSTOMERS;**

**END IF;**

**SELECT id\_client**

**INTO id\_cl**

**FROM CLIENTI**

**WHERE CLIENTI.nume = nume\_client;**

**SELECT id\_media**

**BULK COLLECT INTO t**

**FROM VIZIONEAZA**

**WHERE id\_client = id\_cl;**

**IF t.COUNT() = 0 THEN**

**RAISE NO\_MEDIA;**

**END IF;**

**buget\_total :=0;**

**FOR i in t.FIRST()..t.LAST() LOOP**

**buget\_film :=0;**

**SELECT buget**

**INTO buget\_film**

**FROM MEDIA**

**WHERE id\_media = t(i);**

**buget\_total := buget\_total + buget\_film;**

**END LOOP;**

**RETURN buget\_total;**

**EXCEPTION**

**WHEN NO\_CUSTOMERS THEN**

**DBMS\_OUTPUT.PUT\_LINE('Nu exista client cu acest prenume');**

**RETURN -1;**

**WHEN NO\_MEDIA THEN**

**DBMS\_OUTPUT.PUT\_LINE('Clientul acesta nu a vizionat inca niciun film/serial');**

**RETURN -1;**

**WHEN TOO\_MANY\_CUSTOMERS THEN**

**DBMS\_OUTPUT.PUT\_LINE('Exista mai multi clienti cu acest prenume!');**

**RETURN -1;**

**WHEN OTHERS THEN**

**DBMS\_OUTPUT.PUT\_LINE(SQLCODE);**

**DBMS\_OUTPUT.PUT\_LINE(SQLERRM);**

**RETURN -1;**

**END;**

**/**

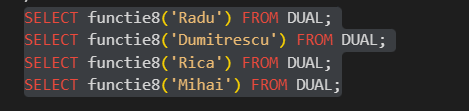
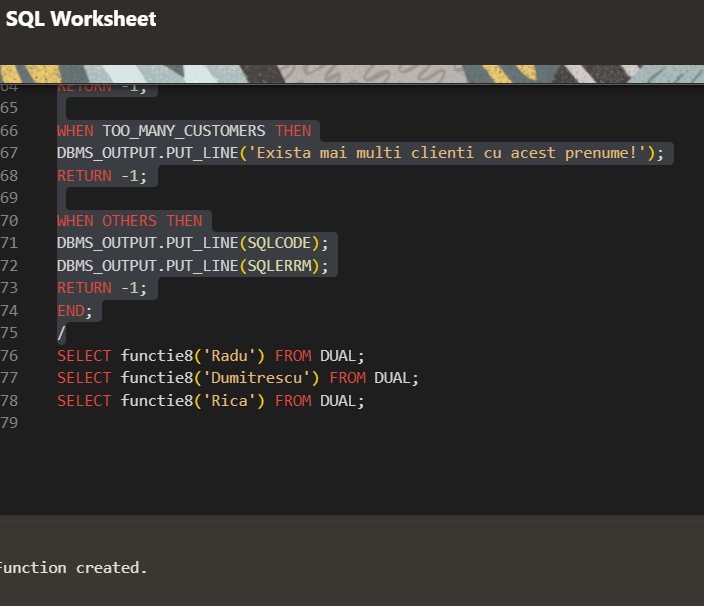
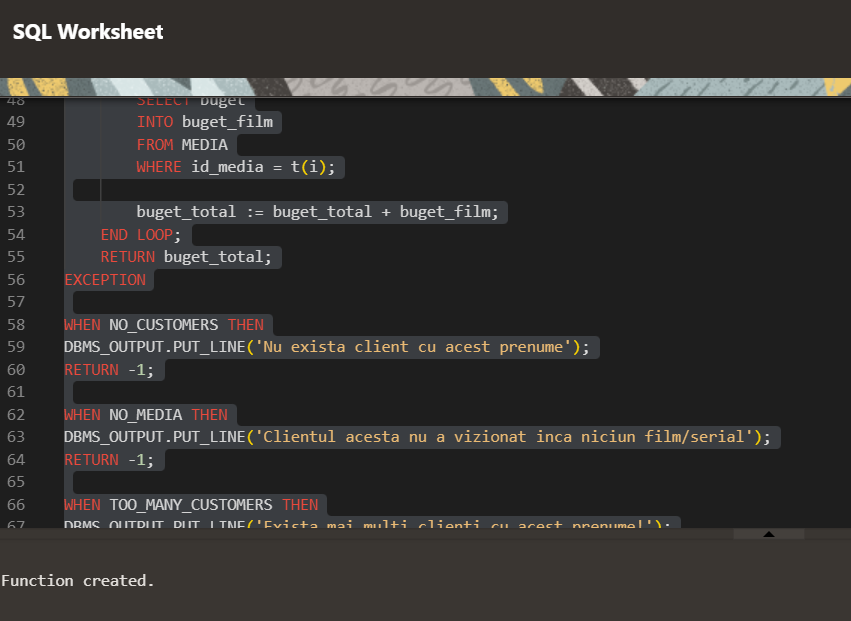
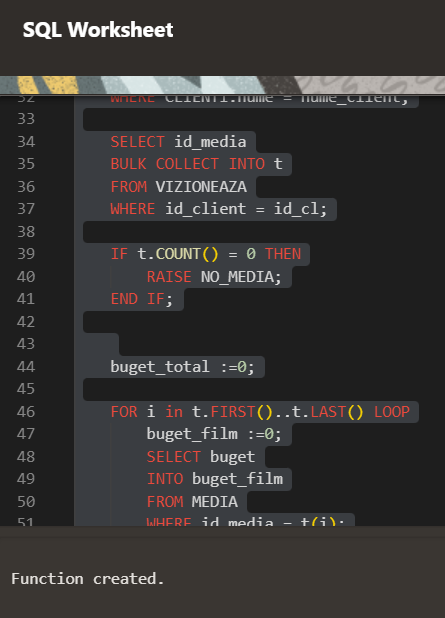
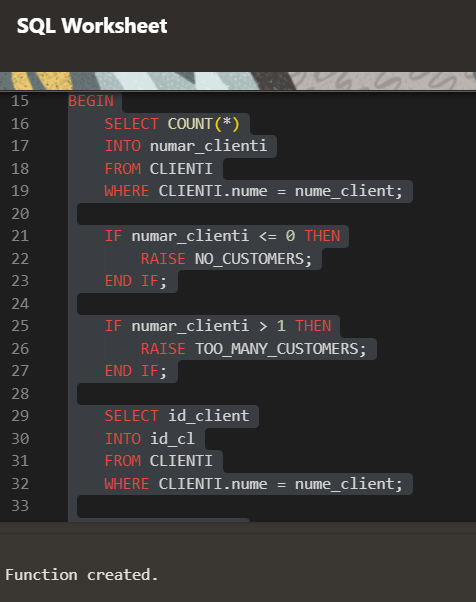
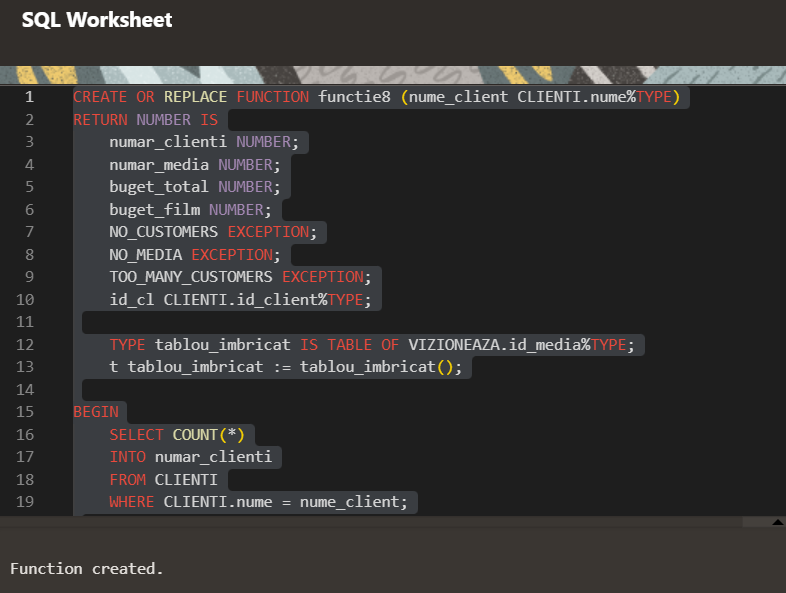
**SELECT functie8('Radu') FROM DUAL;**

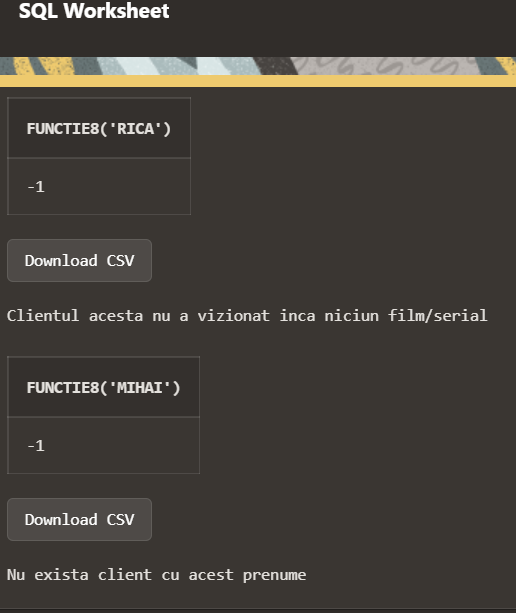
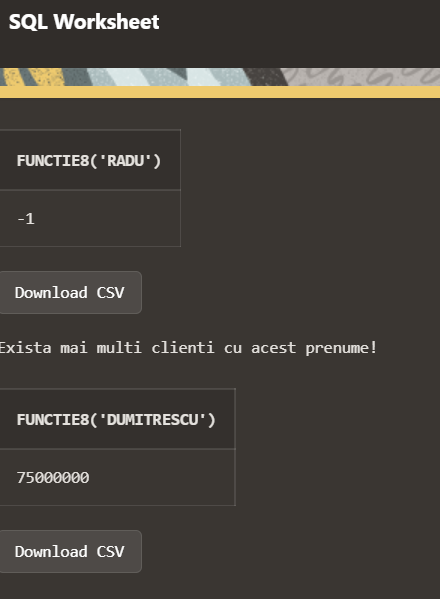
**SELECT functie8('Dumitrescu') FROM DUAL;**

**SELECT functie8('Rica') FROM DUAL;**

**SELECT functie8('Mihai') FROM DUAL;**

**/**

****

****

**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**

**O procedura ce primeste drept parametru prenumele unui actor si afiseaza numele sau complet numarul de filme si seriale in care a jucat rating-ul mediu si total al filmelor si serialelor in care a jucat precum si numarul de spectatori al filmelor si serialelor in care a jucat**

**CREATE OR REPLACE PROCEDURE procedura9 (prenume\_actor ACTORI.prenume%TYPE) IS**

**nume ACTORI.nume%TYPE;**

**nr\_media NUMBER;**

**buget\_media NUMBER;**

**numar\_rating NUMBER;**

**suma\_rating NUMBER;**

**avg\_rating NUMBER;**

**numar\_clienti NUMBER;**

**BEGIN**

**SELECT A.nume, COUNT(J.id\_media), SUM(M.buget), COUNT(V.rating)\*10, SUM(V.rating), ROUND(AVG(V.rating), 2), COUNT(C.id\_client)**

**INTO nume, nr\_media, buget\_media, numar\_rating, suma\_rating, avg\_rating, numar\_clienti**

**FROM ACTORI A LEFT JOIN JOACA J ON (J.id\_actor = A.id\_actor)**

**LEFT JOIN MEDIA M ON (M.id\_media = J.id\_media)**

**LEFT JOIN VIZIONEAZA V ON (V.id\_media = M.id\_media)**

**LEFT JOIN CLIENTI C ON (C.id\_client = V.id\_client)**

**WHERE A.prenume = prenume\_actor**

**GROUP BY A.nume;**

**DBMS\_OUTPUT.PUT\_LINE('Actorul ' || prenume\_actor || ' ' || nume || ': ');**

**DBMS\_OUTPUT.PUT\_LINE('Numar filme/seriale in care a jucat:' || nr\_media);**

**DBMS\_OUTPUT.PUT\_LINE('Rating mediu: '|| avg\_rating || '/ 10');**

**DBMS\_OUTPUT.PUT\_LINE('Rating total: ' || suma\_rating || '/' || numar\_rating);**

**DBMS\_OUTPUT.PUT\_LINE('Numar spectatori:' || numar\_clienti);**

**EXCEPTION**

**WHEN TOO\_MANY\_ROWS**

**THEN RAISE\_APPLICATION\_ERROR(-20007, 'Exista mai multi actori cu acest prenume');**

**WHEN NO\_DATA\_FOUND**

**THEN RAISE\_APPLICATION\_ERROR(-20006, 'Nu exista actori cu acest prenume');**

**END;**

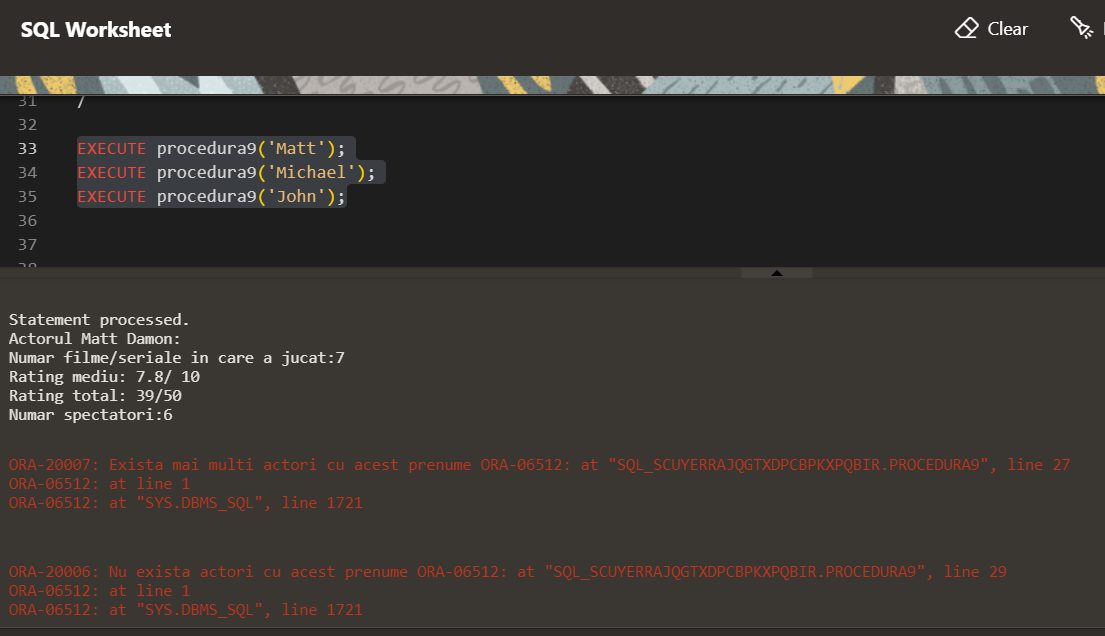
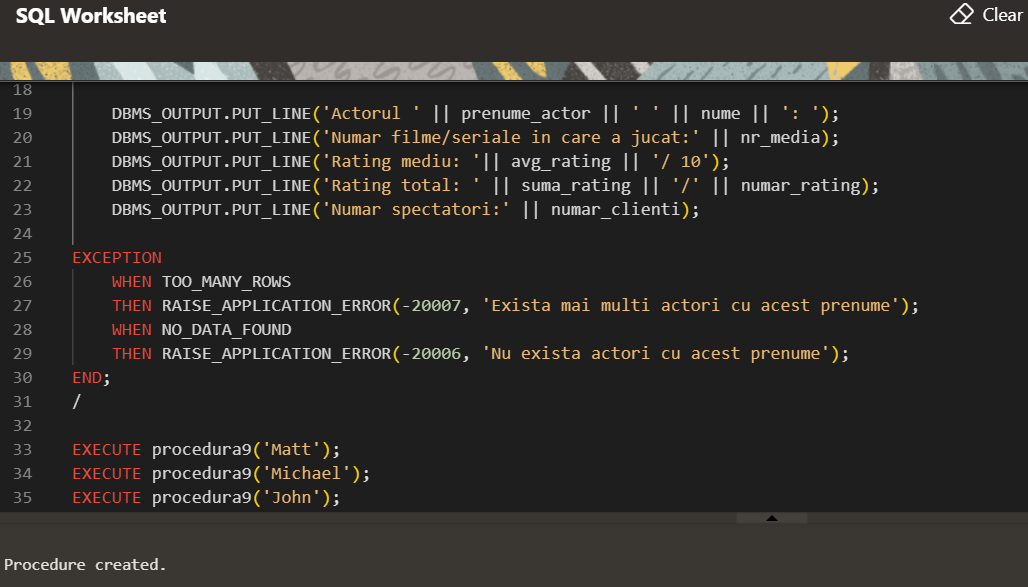
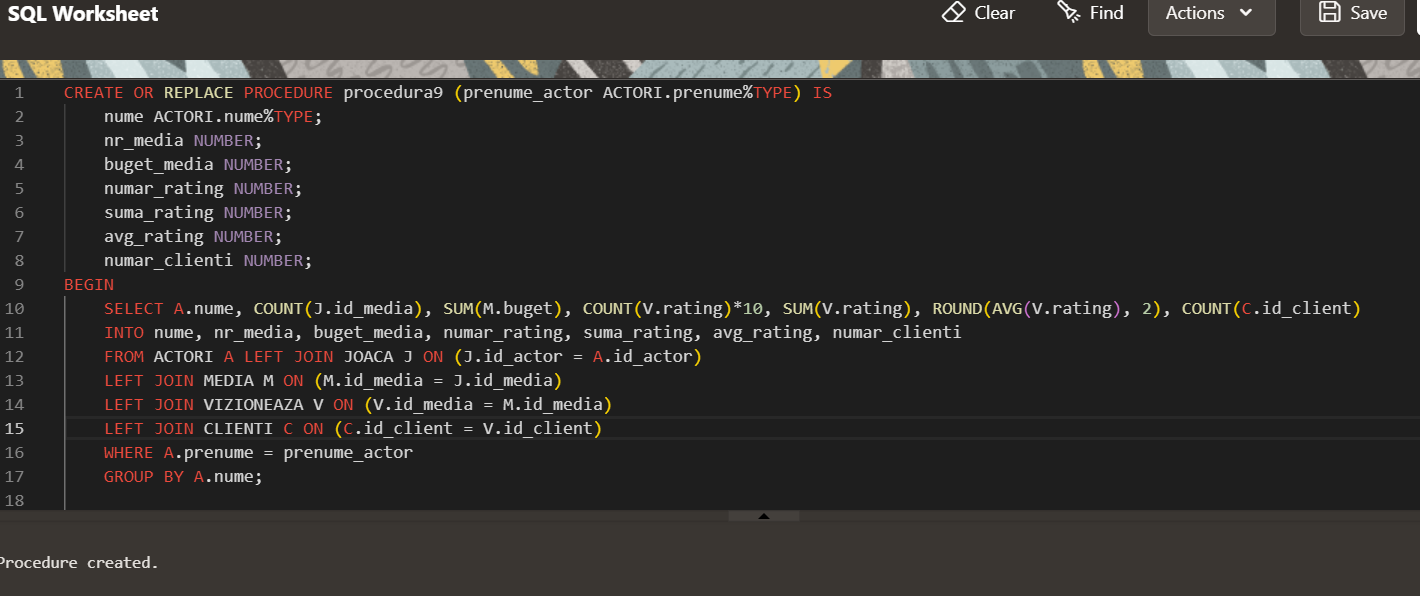
**/**

**EXECUTE procedura9('Matt');**

**EXECUTE procedura9('Michael');**

**EXECUTE procedura9('John');**

**/**

****

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

**Trigger te tip LMD la nivel de comanda ce impiedica stergerea filmelor si serialelor din lista de vizionate.**

**CREATE OR REPLACE TRIGGER trigger10**

**BEFORE DELETE ON VIZIONEAZA**

**BEGIN**

**RAISE\_APPLICATION\_ERROR(-20011, 'Nu puteti elimina un film/serial din lista de vizionate!');**

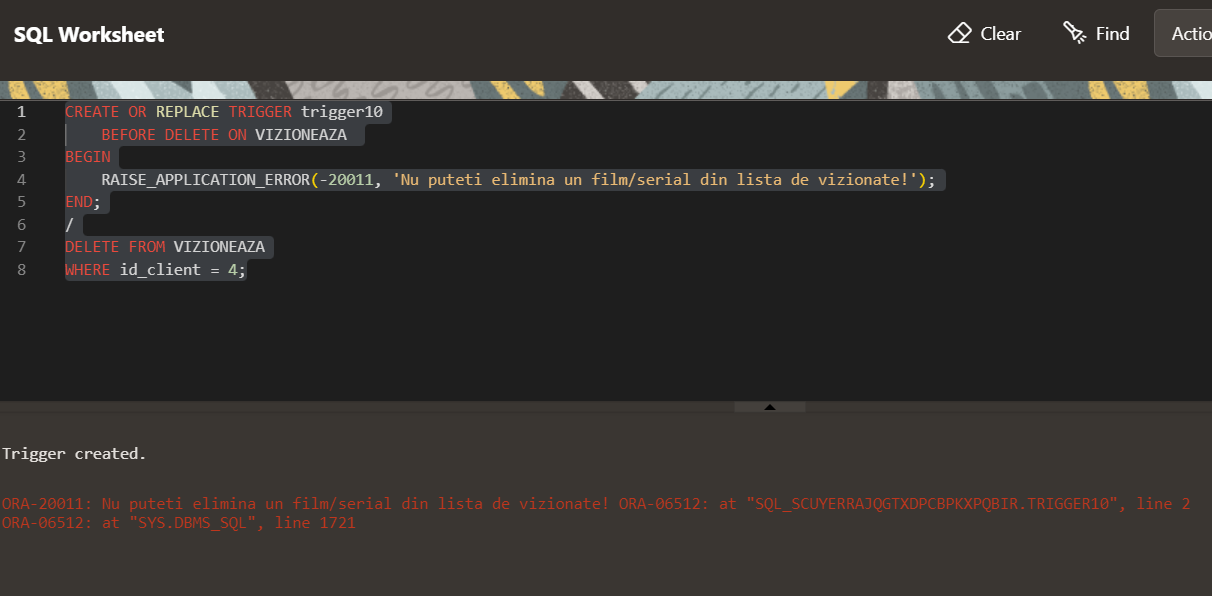
**END;**

**/**

**DELETE FROM VIZIONEAZA**

**WHERE id\_client = 4;**

**/**

****

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

**Trigger de tip LMD la nivel de linie ce impiedica crearea sau modificarea unui rating oferit de un utilizator astfel incat acesta sa fie mai mare de 10.**

**CREATE OR REPLACE TRIGGER trigger\_11**

**BEFORE INSERT OR UPDATE ON VIZIONEAZA**

**FOR EACH ROW**

**BEGIN**

**IF(:NEW.rating > 10) THEN**

**RAISE\_APPLICATION\_ERROR(-20010, 'Nota accordata unui film/serial nu poate fi mai mare de 10!');**

**END IF;**

**END;**

**/**

**INSERT INTO VIZIONEAZA**

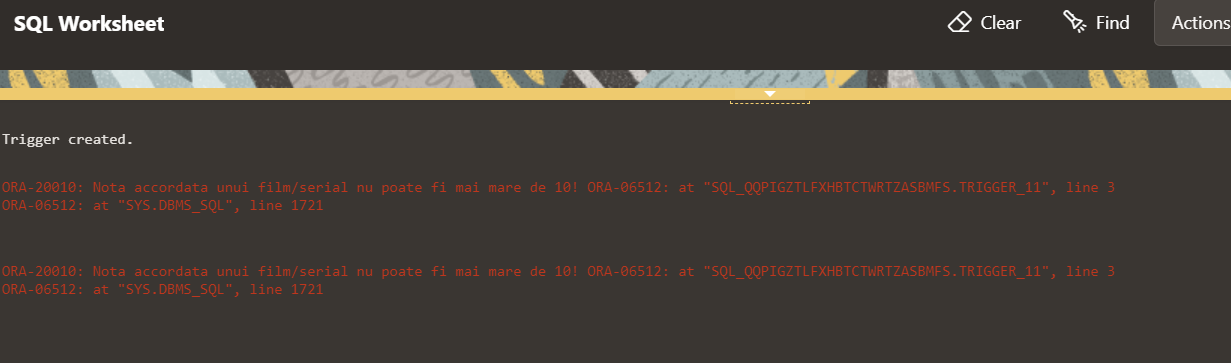
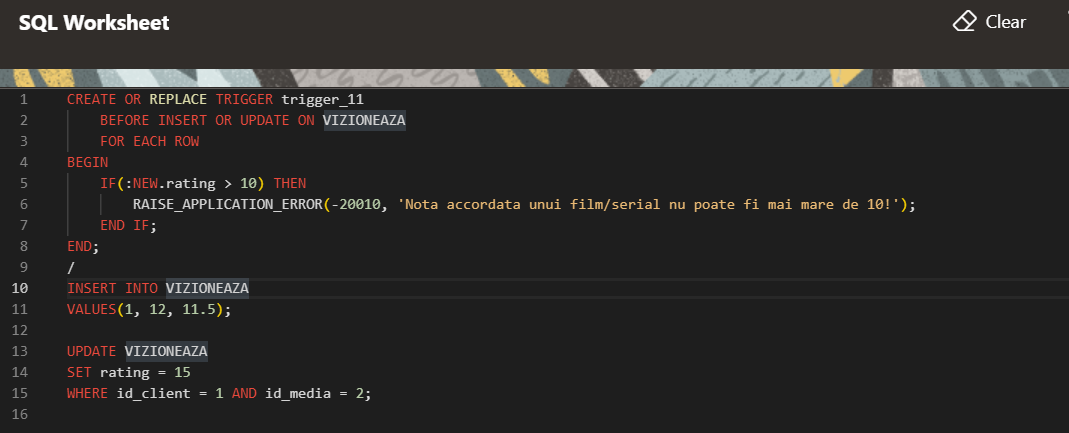
**VALUES(1, 12, 11.5);**

**UPDATE VIZIONEAZA**

**SET rating = 15**

**WHERE id\_client = 1 AND id\_media = 2;**

**/**

****

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

**Trigger de tip LDD ce afiseaza faptul ca a fost executata o comanda LDD de fiecare data cand este cazul**

**CREATE OR REPLACE TRIGGER trigger12**

**AFTER CREATE OR ALTER OR DROP ON SCHEMA**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('A fost efectuata o comanda LDD');**

**END;**

**/**

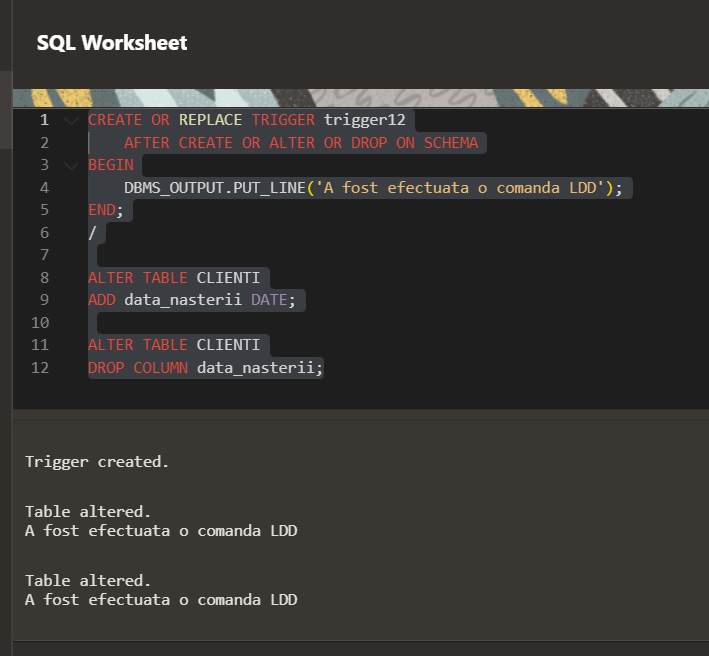
**ALTER TABLE CLIENTI**

**ADD data\_nasterii DATE;**

**ALTER TABLE CLIENTI**

**DROP COLUMN data\_nasterii;**

**/**

****