Bază de date relațională pentru administrarea unui cinematograf

Proiect la materia SGBD

Copilău Andrei

Seria 2

Grupa 232

An universitar 2023-2024

Cuprins

1. [Prezentarea bazei de date si a utilitatii acesteia: 3](#_Toc187427997)
2. [Diagrama entitate-relație(ERD) 4](#_Toc187427998)
3. [Diagrama conceptuală 5](#_Toc187427999)
4. [Definirea tabelelor și a constrângerilor 6](#_Toc187428000)
5. [Inserare de date in tabele 11](#_Toc187428001)
6. [Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul. 19](#_Toc187428002)
7. [Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul. 22](#_Toc187428003)
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 create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate. 25](#_Toc187428004)
9. [Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate. 29](#_Toc187428005)
10. [Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul. 35](#_Toc187428006)
11. [Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul. 37](#_Toc187428007)
12. [Definiți un trigger de tip LDD. Declanșați trigger-ul. 39](#_Toc187428008)

# Prezentarea bazei de date si a utilitatii acesteia:

Prin acest paragraf voi prezenta modelul real al bazei de date pe care o voi crea, regulile de funcționare, cât și infrastructura utilizată pentru implementarea acestuia. Proiectul meu are ca temă implementarea si exploatarea unei baze de a unui cinematograf.

Cu ajutorul acestui model de date, voi putea gestiona, atât clienții care doresc sa își rezerve bilete pentru a viziona un film într-o sala de cinema si la un program anume, cat si angajații care se ocupa de preluarea acestor rezervări si departamentele unde lucrează aceștia. In total voi avea 8 tabele dintre care unul va fi asociativ si va face legătura intre tabelele clienți, bilete si filme. Fiecare tabel va avea o implementare importanta deoarece va ajuta la stocarea eficienta si organizata a datelor.

Un client își poate cumpăra mai multe bilete care pot conține mai multe locuri, pentru un anumit film care va fi difuzat într-o anumita sala, la o data anume. După ce clientul își alege biletul, este realizata o rezervare care ulterior va fi preluata de un angajat care va lucra pe un departament anume.

SGBD utilizat: Oracle Database 19c

Configurația hardware: 2 GB RAM

Configurația software:

* Windows 11 Pro
* Nu am utilizat o mașină virtuală

# Diagrama entitate-relație(ERD)

O imagine care conține text, captură de ecran, Font, Grafică

Descriere generată automat

# Diagrama conceptuală

O imagine care conține captură de ecran, text, proiectare

Descriere generată automat

# Definirea tabelelor și a constrângerilor

* CLIENTI

CREATE TABLE clienti(

client\_id NUMBER(3) PRIMARY KEY,

nume VARCHAR(50) NOT NULL,

varsta NUMBER(3),

nr\_telefon VARCHAR(10) UNIQUE

);

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

* BILETE

CREATE TABLE bilete(

bilet\_id NUMBER(4) PRIMARY KEY,

client\_id NUMBER(3),

film\_id NUMBER(3),

nr\_locuri NUMBER(2) CHECK (nr\_locuri<10),

pret NUMBER(3) CHECK (pret<1000),

angajat\_id NUMBER(3),

FOREIGN KEY (client\_id) REFERENCES clienti(client\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id),

FOREIGN KEY (angajat\_id) REFERENCES angajati(angajat\_id)

);

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

* FILME

CREATE TABLE filme(

film\_id NUMBER(3) PRIMARY KEY,

titlul VARCHAR(50) NOT NULL,

gen VARCHAR(20),

an\_lansare DATE,

regizor VARCHAR(50),

CHECK (an\_lansare>=to\_date('01-01-2024', 'DD-MM-YYYY') AND an\_lansare<=to\_date('31-12-2024', 'DD-MM-YYYY')));

O imagine care conține text, captură de ecran, software, linie

Descriere generată automat

* REZERVARI

CREATE TABLE rezervari(

bilet\_id NUMBER(4),

client\_id NUMBER(3),

film\_id NUMBER(3),

sala\_id NUMBER(2),

data\_achizitionare DATE NOT NULL,

FOREIGN KEY (bilet\_id) REFERENCES bilete(bilet\_id),

FOREIGN KEY (client\_id) REFERENCES clienti(client\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id),

FOREIGN KEY (sala\_id) REFERENCES sali(sala\_id),

PRIMARY KEY(bilet\_id, client\_id, film\_id) --cheie primara compusa

);

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

* DEPARTAMENTE

CREATE TABLE departamente(

departament\_id NUMBER(2) PRIMARY KEY,

nume\_departament VARCHAR(20) UNIQUE

);

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

* ANGAJATI

CREATE TABLE angajati(

angajat\_id NUMBER(3) PRIMARY KEY,

departament\_id NUMBER(2),

nume\_angajat VARCHAR(50),

specializare VARCHAR(20),

salariu NUMBER(5),

data\_angajare DATE,

FOREIGN KEY (departament\_id) REFERENCES departamente(departament\_id)

);

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

* SALI

CREATE TABLE sali(

sala\_id NUMBER(2) PRIMARY KEY,

capacitate NUMBER(2) CHECK (capacitate>0)

);

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

* PROGRAM

CREATE TABLE program(

program\_id VARCHAR(20) PRIMARY KEY,

sala\_id NUMBER(2),

film\_id NUMBER(3),

data\_incepere DATE,

FOREIGN KEY (sala\_id) REFERENCES sali(sala\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id)

);

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

# Inserare de date in tabele

* CLIENTI

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (1, 'Andrei', 20, '0771469060');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (2, 'Amalia', 22, '0758932147');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (3, 'Catalin', 45, '0726415983');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (4, 'Mariana', 47, '0743058692');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (5, 'Bogdan', 16, '0779263815');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (6, 'Alex', 7, '0715472098');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (7, 'Karina', 17, '0738106429');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (8, 'Nicoleta', 41, '0762148359');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (9, 'Ema', 35, '0719835443');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (10, 'Antonio', 26, '0723459809');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (11, 'Rares', 20, '0773572100');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (12, 'Marius', 55, '0727301854');

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

* BILETE

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (1, 1, 1, 1, 20, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (2, 2, 3, 2, 40, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (3, 3, 4, 1, 30, 5);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (4, 4, 5, 4, 120, 6);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (5, 5, 7, 2, 40, 6);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (6, 6, 10, 1, 20, 7);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (7, 7, 9, 3, 60, 5);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (8, 8, 2, 2, 60, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (9, 9, 6, 1, 30, 7);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (10, 10, 8, 5, 150, 6);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (11, 11, 5, 3, 60, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (12, 12, 10, 1, 30, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (13, 1, 9, 2, 40, 4);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id) VALUES (14, 3, 6, 1, 30, 7);

O imagine care conține text, captură de ecran, afișaj, număr

Descriere generată automat

* FILME

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (1, 'Forrest Gump', 'Drama', to\_date('15-05-2024', 'DD-MM-YYYY'), 'Robert Zemekis');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (2, 'Inception', 'SF', to\_date('20-07-2024', 'DD-MM-YYYY'), 'Christopher Nolan');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (3, 'Interstellar', 'SF', to\_date('28-03-2024', 'DD-MM-YYYY'), 'Christopher Nolan');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (4, 'Shutter Island', 'Mister', to\_date('30-10-2024', 'DD-MM-YYYY'), 'Martin Scorsese');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (5, 'Fast Furious', 'Actiune', to\_date('26-05-2024', 'DD-MM-YYYY'), 'Louis Leterrier');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (6, 'Se7en', 'Thriller', to\_date('12-08-2024', 'DD-MM-YYYY'), 'David Fincher');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (7, 'The Godfather', 'Actiune', to\_date('01-09-2024', 'DD-MM-YYYY'), 'Francis Ford Coppola');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (8, 'Matrix', 'SF', to\_date('22-11-2024', 'DD-MM-YYYY'), 'Lilly Wackowski');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (9, 'Dune', 'SF', to\_date('07-07-2024', 'DD-MM-YYYY'), 'Denis Villeneuve');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (10, 'Bad Boys', 'Comedie', to\_date('16-12-2024', 'DD-MM-YYYY'), 'Michael Bay');

O imagine care conține text, Font, linie, număr

Descriere generată automat

* REZERVARI

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (1, 1, 1, 1, to\_date('19-05-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (2, 2, 3, 2, to\_date('28-03-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (3, 3, 4, 2, to\_date('01-11-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (4, 4, 5, 3, to\_date('27-05-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (5, 5, 7, 4, to\_date('16-09-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (6, 6, 10, 5, to\_date('18-12-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (7, 7, 9, 5, to\_date('07-07-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (8, 8, 2, 1, to\_date('25-07-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (9, 9, 6, 3, to\_date('14-08-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (10, 10, 8, 4, to\_date('23-11-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (11, 11, 5, 3, to\_date('26-05-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (12, 12, 10, 5, to\_date('18-12-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (13, 1, 9, 5, to\_date('27-07-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (14, 3, 6, 3, to\_date('28-08-2024', 'DD-MM-YYYY'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (15, 4, 5, 3, to\_date('29-05-2024', 'DD-MM-YYYY'));

O imagine care conține text, captură de ecran, afișaj, număr

Descriere generată automat

* DEPARTAMENTE

INSERT INTO departamente(departament\_id, nume\_departament) VALUES (1, 'vanzare');

INSERT INTO departamente(departament\_id, nume\_departament) VALUES (2, 'management');

INSERT INTO departamente(departament\_id, nume\_departament) VALUES (3, 'resurse umane');

INSERT INTO departamente(departament\_id, nume\_departament) VALUES (4, 'marketing');

INSERT INTO departamente(departament\_id, nume\_departament) VALUES (5, 'ingrijire');

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat

* ANGAJATI

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (1, 2, 'Copilau Andrei', 'manager', 10000, to\_date('14-01-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (2, 2, 'Fran Daniela', 'asistent management', 9000, to\_date('17-02-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (3, 2, 'Alex Constantin', 'asistent management', 8500, to\_date('20-02-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (4, 1, 'Kobel Dan', 'casier', 3000, to\_date('07-07-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (5, 1, 'Florentin Rus', 'casier', 3000, to\_date('07-07-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (6, 1, 'Andu Marian', 'casier', 3500, to\_date('02-03-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (7, 1, 'Stamate Mircea', 'casier', 4000, to\_date('21-04-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (8, 3, 'Dumitru Beatrice', 'specialist HR', 5500, to\_date('22-02-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (9, 3, 'Done Albert', 'specialist HR', 5500, to\_date('22-02-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (10, 4, 'Gheorghe Mariana', 'brand manager', 6000, to\_date('13-03-2023', 'DD-MM-YYYY'));

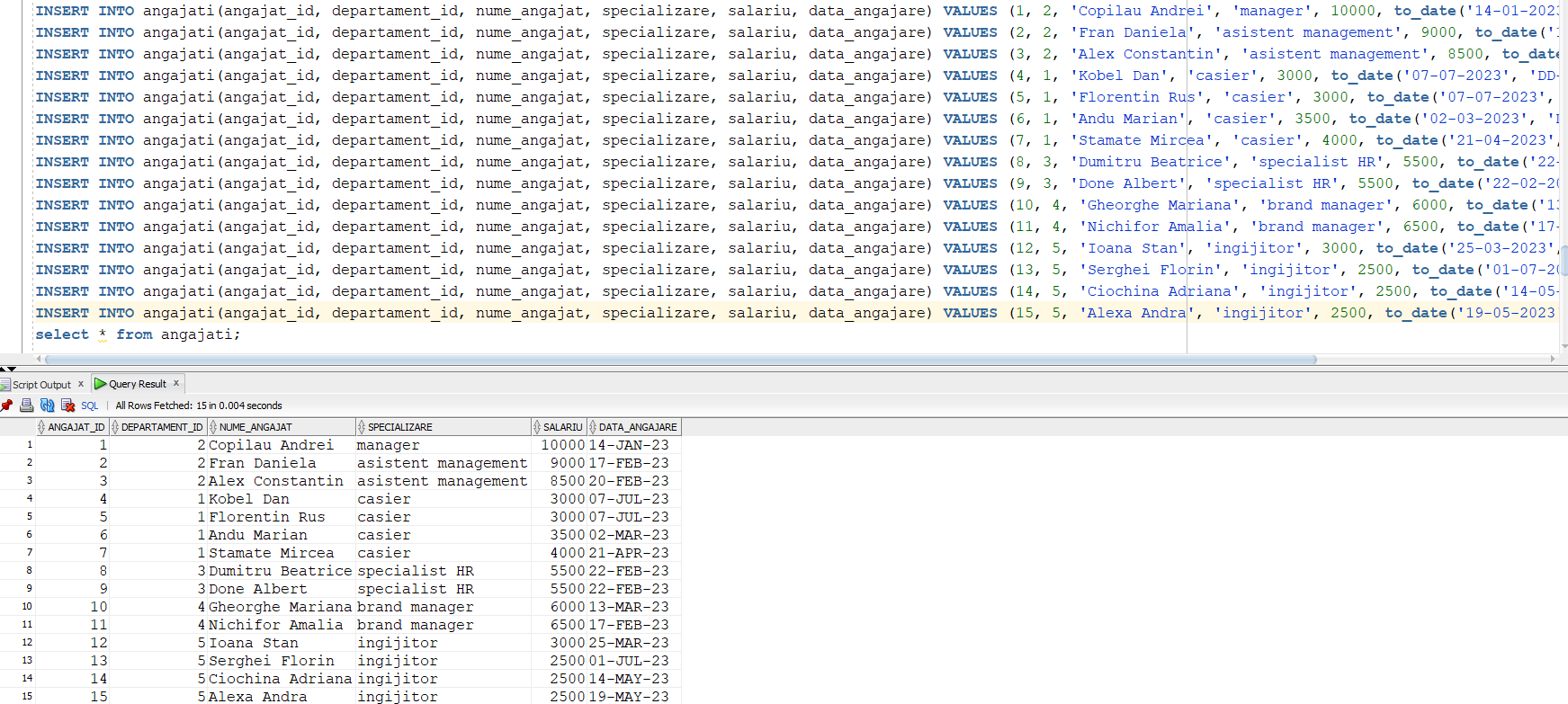
INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (11, 4, 'Nichifor Amalia', 'brand manager', 6500, to\_date('17-02-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (12, 5, 'Ioana Stan', 'ingijitor', 3000, to\_date('25-03-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (13, 5, 'Serghei Florin', 'ingijitor', 2500, to\_date('01-07-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (14, 5, 'Ciochina Adriana', 'ingijitor', 2500, to\_date('14-05-2023', 'DD-MM-YYYY'));

INSERT INTO angajati(angajat\_id, departament\_id, nume\_angajat, specializare, salariu, data\_angajare) VALUES (15, 5, 'Alexa Andra', 'ingijitor', 2500, to\_date('19-05-2023', 'DD-MM-YYYY'));



* SALI

INSERT INTO sali(sala\_id, capacitate) VALUES (1, 50);

INSERT INTO sali(sala\_id, capacitate) VALUES (2, 50);

INSERT INTO sali(sala\_id, capacitate) VALUES (3, 60);

INSERT INTO sali(sala\_id, capacitate) VALUES (4, 65);

INSERT INTO sali(sala\_id, capacitate) VALUES (5, 70);

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

* PROGRAM

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#1', 1, 1, to\_date('15-05-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#2', 1, 2, to\_date('20-07-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#3', 2, 3, to\_date('28-03-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#4', 2, 4, to\_date('30-10-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#5', 3, 5, to\_date('26-05-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#6', 3, 6, to\_date('12-08-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#7', 4, 7, to\_date('01-09-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#8', 4, 8, to\_date('22-11-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#9', 5, 9, to\_date('07-07-2024', 'DD-MM-YYYY'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#10', 5, 10, to\_date('16-12-2024', 'DD-MM-YYYY'));

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

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

**Enunț:** Să se afișeze pentru o sală dată ca parametru, filmele, numărul de bilete vândute pentru fiecare film selectat și clienții care au rezervat aceste bilete.

CREATE OR REPLACE PROCEDURE f6(id\_sala IN NUMBER) IS

TYPE tablou\_filme IS VARRAY(10) OF VARCHAR2(50);

lista\_filme tablou\_filme;

TYPE tablou\_clienti IS TABLE OF VARCHAR2(50);

lista\_clienti tablou\_clienti;

TYPE tablou\_bilete IS TABLE OF NUMBER INDEX BY VARCHAR2(50);

bilete\_per\_film tablou\_bilete;

total\_locuri NUMBER;

BEGIN

SELECT DISTINCT(f.titlul) BULK COLLECT INTO lista\_filme

FROM filme f

JOIN program p ON f.film\_id = p.film\_id

JOIN sali s ON p.sala\_id = s.sala\_id

WHERE s.sala\_id = id\_sala;

DBMS\_OUTPUT.PUT\_LINE('Filmele selectate din sala '|| id\_sala ||' sunt: ');

FOR i IN 1..lista\_filme.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(lista\_filme(i));

END LOOP;

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

SELECT c.nume BULK COLLECT INTO lista\_clienti

FROM clienti c

JOIN rezervari r ON c.client\_id = r.client\_id

JOIN program p ON r.film\_id = p.film\_id

JOIN sali s ON p.sala\_id = s.sala\_id

WHERE s.sala\_id = id\_sala;

DBMS\_OUTPUT.PUT\_LINE('Clientii din sala '|| id\_sala ||' sunt: ');

FOR i IN 1..lista\_clienti.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(lista\_clienti(i));

END LOOP;

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

FOR i IN 1..lista\_filme.COUNT LOOP

SELECT SUM(b.nr\_locuri) INTO total\_locuri

FROM bilete b

JOIN filme f ON b.film\_id = f.film\_id

JOIN program p ON f.film\_id = p.film\_id

JOIN sali s ON p.sala\_id = s.sala\_id

WHERE s.sala\_id = id\_sala AND f.titlul = lista\_filme(i);

bilete\_per\_film(lista\_filme(i)) := total\_locuri;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Numarul de bilete pentru fiecare film: ');

FOR i IN 1..lista\_filme.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Filmul '|| lista\_filme(i) ||' are nr de bilete vandute: '||bilete\_per\_film(lista\_filme(i)));

END LOOP;

END;

/

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

--Apelul subprogramului

BEGIN

f6(1);

END;

/

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

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

**Enunț:** Să se afișeze pentru fiecare departament, numărul de angajați și angajații care lucrează în acesta.

CREATE OR REPLACE PROCEDURE f7 IS

--cursor explicit

CURSOR c1 IS

SELECT d.departament\_id, d.nume\_departament, COUNT(a.angajat\_id)

FROM departamente d JOIN angajati a ON d.departament\_id = a.departament\_id

GROUP BY d.departament\_id, d.nume\_departament;

--cursor parametrizat

CURSOR c2(dep\_id departamente.departament\_id%TYPE) IS

SELECT nume\_angajat

FROM angajati

WHERE departament\_id = dep\_id;

var\_id departamente.departament\_id%TYPE;

var\_nume departamente.nume\_departament%TYPE;

nr\_ang NUMBER;

BEGIN

OPEN c1;

LOOP

FETCH c1 INTO var\_id, var\_nume, nr\_ang;

EXIT WHEN c1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Departamentul '|| var\_nume ||' are ' ||nr\_ang||' angajati:');

FOR i IN c2(var\_id) LOOP

DBMS\_OUTPUT.PUT\_LINE('-'|| i.nume\_angajat);

END LOOP;

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

END LOOP;

CLOSE c1;

END;

/

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat

--Apelul subprogramului

BEGIN

f7;

END;

/

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

# 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 create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

**Enunț:** Pentru un client dat, afișez numele, titlul filmului și data achiziționării biletului. Dacă clientul a realizat mai multe rezervări se va genera o eroare, la fel si daca nu se vor găsi date pentru clientul dat sau daca întâmpinam alte probleme.

CREATE OR REPLACE FUNCTION f8(id\_c IN NUMBER)

RETURN VARCHAR2 IS

var\_nume clienti.nume%TYPE;

var\_titlu filme.titlul%TYPE;

var\_data rezervari.data\_achizitionare%TYPE;

var\_mesaj VARCHAR2(100);

BEGIN

SELECT c.nume, f.titlul, r.data\_achizitionare

INTO var\_nume, var\_titlu, var\_data

FROM clienti c JOIN rezervari r ON c.client\_id = r.client\_id

JOIN filme f ON r.film\_id = f.film\_id

WHERE c.client\_id = id\_c;

var\_mesaj := 'Client: ' || var\_nume || ', Film: ' || var\_titlu || ', Data achiztionarii: ' || var\_data;

RETURN var\_mesaj;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Nu au fost gasite date pentru clientul respectiv.');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Prea multe rezervari pentru clientul respectiv.');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Alta eroare!');

END f8;

/

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

--Apelul subprogramului pe fiecare caz.

--Pentru primul caz alegem clientul cu id 1, care a realizat mai multe rezervari

BEGIN

-- Pentru clientul 1:

DBMS\_OUTPUT.PUT\_LINE(f8(1));

END;

/

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

--Pentru acest caz alegem clientul 2 pentru care se vor afisa informatiile respective

BEGIN

-- Pentru clientul 2:

DBMS\_OUTPUT.PUT\_LINE(f8(2));

END;

/

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

--Pentru ultimul caz alegem un client inexistent

BEGIN

-- Pentru un client inexistent:

DBMS\_OUTPUT.PUT\_LINE(f8(15));

END;

/

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

# Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

**Enunț:** Pentru un client și un film dat, se va verifica prima dată dacă clientul este unul valid și filmul este unul existent. În caz contrar va fi aruncată prima exceptie proprie numită ex\_film\_inexistent. Dacă ambii parametri sunt valizi, vom verifica daca acest client a făcut o rezervare. Dacă nu a făcut, vom returna a doua excepție proprie numită ex\_rezervare\_inexistenta, dacă va avea mai mult de 2 rezervări va intra pe cazul TOO\_MANY\_ROW, iar dacă a realizat o rezervare validă vom afișa informații despre aceasta.

CREATE OR REPLACE PROCEDURE f9(id\_client IN NUMBER, id\_film IN NUMBER)

IS

ex\_film\_inexistent EXCEPTION;

ex\_rezervare\_inexistenta EXCEPTION;

var\_client NUMBER;

var\_film NUMBER;

nr\_rezervari NUMBER;

var\_varsta clienti.varsta%TYPE;

var\_nume clienti.nume%TYPE;

var\_titlu filme.titlul%TYPE;

var\_data rezervari.data\_achizitionare%TYPE;

var\_sala sali.sala\_id%TYPE;

var\_pret bilete.pret%TYPE;

var\_angajat angajati.nume\_angajat%TYPE;

BEGIN

SELECT COUNT(\*) INTO var\_client

FROM clienti

WHERE client\_id = id\_client;

IF var\_client = 0 THEN

RAISE NO\_DATA\_FOUND;

END IF;

SELECT COUNT(\*) INTO var\_film

FROM filme

WHERE film\_id = id\_film;

IF var\_film = 0 THEN

RAISE ex\_film\_inexistent;

END IF;

SELECT COUNT(\*) INTO nr\_rezervari

FROM rezervari

WHERE client\_id = id\_client AND film\_id = id\_film;

IF nr\_rezervari = 0 THEN

RAISE ex\_rezervare\_inexistenta;

ELSIF nr\_rezervari > 1 THEN

RAISE TOO\_MANY\_ROWS;

ELSE

SELECT c.nume, f.titlul, r.data\_achizitionare, r.sala\_id, b.pret, a.nume\_angajat

INTO var\_nume, var\_titlu, var\_data, var\_sala, var\_pret, var\_angajat

FROM clienti c JOIN rezervari r ON c.client\_id = r.client\_id

JOIN filme f ON r.film\_id = f.film\_id

JOIN bilete b ON f.film\_id = b.film\_id

JOIN angajati a ON b.angajat\_id = a.angajat\_id

WHERE r.client\_id = id\_client AND r.film\_id = id\_film;

DBMS\_OUTPUT.PUT\_LINE('Detalii rezervare:');

DBMS\_OUTPUT.PUT\_LINE('~Client: '|| var\_nume);

DBMS\_OUTPUT.PUT\_LINE('~Film: '|| var\_titlu);

DBMS\_OUTPUT.PUT\_LINE('~Data achizitiei: '|| var\_data);

DBMS\_OUTPUT.PUT\_LINE('~Sala: '|| var\_sala);

DBMS\_OUTPUT.PUT\_LINE('~Pret: '|| var\_pret);

DBMS\_OUTPUT.PUT\_LINE('~Nume angajat: '|| var\_angajat);

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Nu au fost gasite date pentru clientul respectiv.');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Clientul are mai multe rezervari la acest film!');

WHEN ex\_film\_inexistent THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Nu exista un film cu acest id.');

WHEN ex\_rezervare\_inexistenta THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista nicio rezervare pentru acest client');

END;

/

O imagine care conține text, captură de ecran, software, Pagină web

Descriere generată automat

--Apelul procedurii pe fiecare caz.

BEGIN

-- Pentru clientul inexistent

f9(20,7);

END;

/

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

BEGIN

-- Pentru filmul inexistent

f9(5,11);

END;

/

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

BEGIN

-- Rezervare inexistenta

f9(5,8);

END;

/

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

BEGIN

-- Client cu mai multe rezervari pentru un singur film

f9(4,5);

END;

/

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

BEGIN

-- Client cu o singura rezervare valida.

f9(2,3);

END;

/

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

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

**Enunț:** Am creat un trigger care îmi afișează o eroare în momentul în care încerc să inserez mai mult de 10 filme în tabela filme.

CREATE OR REPLACE TRIGGER t10

BEFORE INSERT ON filme

DECLARE

nr\_filme NUMBER;

BEGIN

SELECT COUNT(film\_id) INTO nr\_filme

FROM filme;

IF nr\_filme >= 10 THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Eroare: Nu se pot difuza mai mult de 10 filme momentan.');

END IF;

END;

/

--Incerc sa inserez al 11-lea film.

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor)

VALUES (11, 'Harry Potter', 'SF', to\_date('19-09-2024', 'DD-MM-YYYY'), 'Chris Columbus');

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat

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

**Enunț:** Am creat un trigger care îmi afișează o eroare atunci când încerc să inserez sau să modific data de achiziționare a biletului la film, iar această dată este înainte de data de lansare a filmului.

CREATE OR REPLACE TRIGGER t11

BEFORE INSERT OR UPDATE ON rezervari

FOR EACH ROW

DECLARE

var\_data DATE;

BEGIN

SELECT an\_lansare INTO var\_data

FROM filme

WHERE film\_id = :NEW.film\_id;

IF :NEW.data\_achizitionare < var\_data THEN

RAISE\_APPLICATION\_ERROR(-20011, 'Eroare: Data achizitionarii biletului nu este corecta.');

END IF;

END;

/

--Am inserat inca un bilet pentru un film

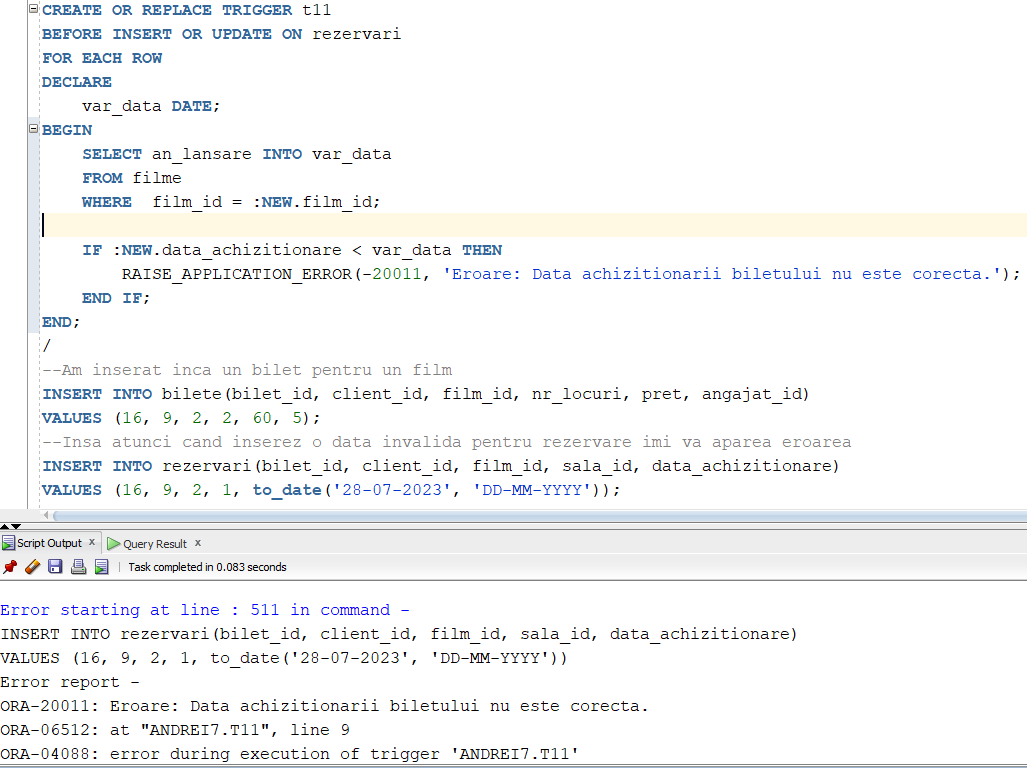
INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret, angajat\_id)

VALUES (16, 9, 2, 2, 60, 5);

--Insa atunci cand inserez o data invalida pentru rezervare imi va aparea eroarea

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare)

VALUES (16, 9, 2, 1, to\_date('28-07-2023', 'DD-MM-YYYY'));



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

**Enunț:** Am creat un trigger și două tabele în care se vor stoca erorile/modificările care vor apărea atunci când lucrez asupra bazei de date, iar triggerul inserează aceste date in tabelele respective.

CREATE TABLE erori\_proiect(

nume\_baza VARCHAR2(50),

user\_baza VARCHAR2(50),

eroare VARCHAR2(2000),

data\_eroare DATE

);

CREATE TABLE modificari\_proiect(

nume\_baza VARCHAR2(50),

user\_baza VARCHAR2(50),

modificare VARCHAR2(200),

tabel\_modificat VARCHAR2(100),

data\_modificare DATE

);

CREATE OR REPLACE TRIGGER t12

AFTER CREATE OR ALTER OR DROP OR SERVERERROR ON SCHEMA

DECLARE

var\_event VARCHAR2(100);

BEGIN

var\_event := SYS.SYSEVENT;

IF var\_event IN ('CREATE', 'ALTER', 'DROP') THEN

INSERT INTO modificari\_proiect(nume\_baza, user\_baza, modificare, obiect\_modificat, data\_modificare)

VALUES( SYS.DATABASE\_NAME,

SYS.LOGIN\_USER,

SYS.SYSEVENT,

SYS.DICTIONARY\_OBJ\_NAME,

SYSDATE);

ELSIF var\_event = 'SERVERERROR' THEN

INSERT INTO erori\_proiect(nume\_baza, user\_baza, eroare, data\_eroare)

VALUES( SYS.DATABASE\_NAME,

SYS.LOGIN\_USER,

SYS.SYSEVENT,

SYSDATE);

END IF;

END;

/

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

*1-Dacă încerc să dau drop la un tabel inexistent, această acțiune se va considera o eroare(sau daca întâmpin orice alta eroare de sistem).*

drop table exemplu;

O imagine care conține text, captură de ecran, afișaj, software

Descriere generată automat

*2-Dacă creez un nou tabel, voi stoca în modificari\_proiect aceasta informație.*

create table exemplu(id\_ex NUMBER);

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

*3-Dacă voi încerca sa modific noul tabel creat sau orice alt tabel, voi stoca aceasta informație in tabelul modificari\_proiect.*

ALTER TABLE exemplu

ADD nume\_exemplu VARCHAR2(50);

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat

*4-Dacă voi da drop la noul tabel creat sau la oricare tabel din baza de date, voi stoca această informație in tabelul modificari\_proiect.*

drop table exemplu;

O imagine care conține text, electronice, captură de ecran, software

Descriere generată automat