

Proiect Sisteme de Gestiune a BD

An II

An Universitar 2024-2025

Stîngă Alexandru-Ionuț

Seria 24

Grupa 243

Cuprins

Introducere	3
Cerinta 1: Prezentați pe scurt baza de date (utilitatea ei).	4
Cerinta 2 : Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD, model de diagrama entitate-relație; nu se va accepta alt format).	5
Cerinta 3 : Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.....	6
Cerinta 4 : Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, adăugând toate constrângerile de integritate necesare (chei primare, cheile externe etc).....	7
Cerinta 5: Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).	17
Cerinta 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.....	30
Cerinta 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.....	36
Cerinta 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.	41
Cerinta 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	45
Cerinta 10 : Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.....	50
Cerinta 11: Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.	52
Cerinta 12: Definiți un trigger de tip LDD. Declanșați trigger-ul.....	55
Cerinta 13 : Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).....	58

Introducere

Tema poriectului este baza de date a unui muzeu ce detine si o platforma de vindere a biletelor unde clientii trebuie sa aiba cont de utilizator.

Infrastructura utilizata pentru implementarea acestuia: versiunea SGBD este Oracle Versiunea 19C, aplicatie SQL Developer, Sistemul are 8 GB RAM si utilizeaza Windows 10, versiunea 22H2 si nu ruleaza nicio Masina Virtuala.

Cerinta 1: Prezentati pe scurt baza de date (utilitatea ei).

Site-ul unui muzeu permite crearea unui cont de Utilizator.

Utilizatorul poate achizitiona Bilete pentru Evenimentele organizate in respectiva perioada sau pentru diferitele Galerii ale muzeului.

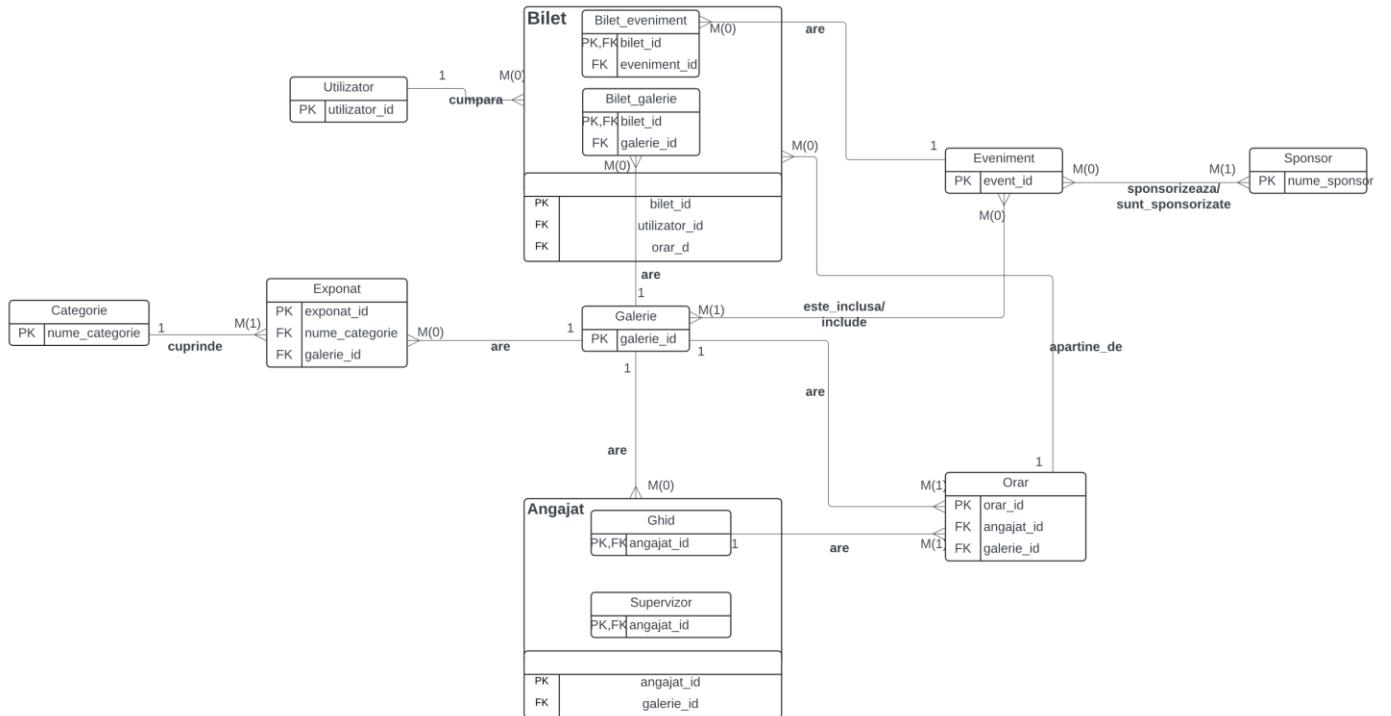
Evenimentele reprezinta accesul neconditionat la anumite galerii din muzeu conform tematicii alese de organizatori. Totodata, orice eveniment are cel putin un Sponsor. Orice sponsor poate sa sponsorizeze mai multe evenimente.

Galeriile contin, bineinteles, un numar variabil de Exponate. Aceste exponate sunt impartite, la randul lor, in mai multe Categorii. Un exponat poate sa apartina doar unei singure categorii.

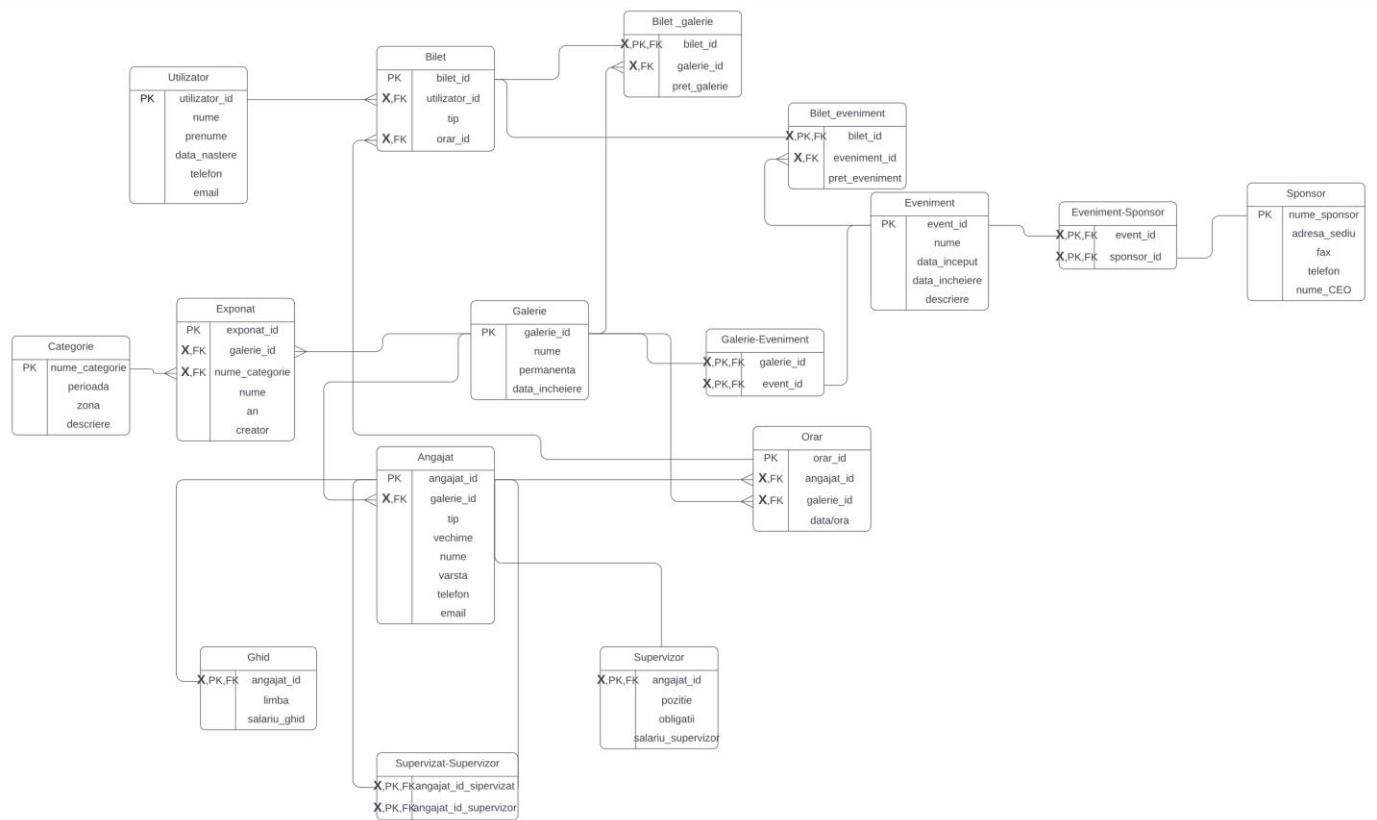
Normal ca la fiecare galerie lucreaza mai multi Angajati. Acestia sunt de doua tipuri: Ghizi si Supervizori. Un utilizator poate la cumpararea biletului sa decida daca va face turul impreuna cu un ghid. Supervizorii ii au in subordine pe ghizi, insa si supervisorii au supervisorii lor. Un angajat are ca supervisor orice persoana ce se afla deasupra lui in ierarhia conducerii Muzeului. Bineinteles ca ghizii sunt in punctul cel mai de jos al ierarhie, motiv pentru care nu pot fi in supervisorii altor angajati.

Ghizii, galeriile si biletele sunt legate prin intermediul Orarului, deoarece toate vizitele sunt realizate impreuna cu un ghid.

Cerinta 2 : Realizați diagrama entitate-relație (ERD): entitățile, relațiile și attributele trebuie definite în limba română (vezi curs SGBD, model de diagrama entitate-relație; nu se va accepta alt format).



Cerinta 3 : Pornind de la diagrama entitate-relație realizată diagrama conceptuală a modelului propus, integrând toate attributele necesare: entitățile, relațiile și attributele trebuie definite în limba română.



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

```
CREATE SEQUENCE utilizator_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE galerie_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE event_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE exponat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE bilet_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE angajat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE orar_seq START WITH 1 INCREMENT BY 1;
CREATE TABLE Utilizator (
    utilizator_id INT DEFAULT utilizator_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(50),
    prenume VARCHAR(50),
    data_nastere DATE,
    telefon VARCHAR(15),
    email VARCHAR(100) );
CREATE TABLE Galerie (
    galerie_id INT DEFAULT galerie_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    permanenta VARCHAR(1),
    data_incheiere DATE );
CREATE TABLE Eveniment (
    event_id INT DEFAULT event_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    data_inceput DATE,
    data_incheiere DATE,
```

```

descriere VARCHAR(300) );

CREATE TABLE Sponsor (
    nume_sponsor VARCHAR(100) PRIMARY KEY,
    adresa_sediu VARCHAR(255),
    fax VARCHAR(20),
    telefon VARCHAR(15),
    nume_CEO VARCHAR(100) );

CREATE TABLE Categorie (
    nume_categorie VARCHAR(100) PRIMARY KEY,
    perioada VARCHAR(50),
    zona VARCHAR(50),
    descriere VARCHAR(300) );

CREATE TABLE Exponat(
    exponat_id INT DEFAULT exponat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    nume_categorie VARCHAR(100) REFERENCES Categorie(nume_categorie),
    nume VARCHAR(100),
    an INT,
    creator VARCHAR(100) );

CREATE TABLE Angajat (
    angajat_id INT DEFAULT angajat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    tip VARCHAR(50),
    vechime INT,
    nume VARCHAR(100),
    varsta INT,
    telefon VARCHAR(15),
    email VARCHAR(100));

```

```

CREATE TABLE Ghid (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    limba VARCHAR(50),
    salariu_ghid NUMBER(10, 2));

CREATE TABLE Supervizor (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    pozitie VARCHAR(100),
    obligatii VARCHAR(4000),
    salariu_supervizor NUMBER(10, 2));

CREATE TABLE Orar (
    orar_id INT DEFAULT orar_seq.NEXTVAL PRIMARY KEY,
    angajat_id INT REFERENCES Angajat(angajat_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    data_orar DATE );

CREATE TABLE Bilet (
    bilet_id INT DEFAULT bilet_seq.NEXTVAL PRIMARY KEY,
    utilizator_id INT REFERENCES Utilizator(utilizator_id),
    tip VARCHAR(50),
    orar_id INT REFERENCES Orar(orar_id));

CREATE TABLE Bilet_Galerie (
    bilet_id INT PRIMARY KEY ,
    FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    pret_galerie NUMBER(10, 2 ));

CREATE TABLE Bilet_Eveniment (
    bilet_id INT PRIMARY KEY ,

```

```

FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
event_id INT REFERENCES Eveniment(event_id),
pret_eveniment NUMBER(10, 2 );

CREATE TABLE Supervizat_Supervizor (
angajat_id_supervizat INT,
angajat_id_supervizor INT,
PRIMARY KEY(angajat_id_supervizat,angajat_id_supervizor),
FOREIGN KEY(angajat_id_supervizat) REFERENCES Angajat (angajat_id),
FOREIGN KEY(angajat_id_supervizor) REFERENCES Angajat(angajat_id));

CREATE TABLE Galerie_Eveniment (
galerie_id INT,
event_id INT,
PRIMARY KEY(galerie_id,event_id),
FOREIGN KEY(galerie_id) REFERENCES Galerie (galerie_id),
FOREIGN KEY(event_id) REFERENCES Eveniment(event_id));

CREATE TABLE Eveniment_Sponsor (
event_id INT,
nume_sponsor VARCHAR(100),
PRIMARY KEY(event_id,nume_sponsor),
FOREIGN KEY(event_id) REFERENCES Eveniment (event_id),
FOREIGN KEY(nume_sponsor) REFERENCES Sponsor(nume_sponsor));

CREATE SEQUENCE utilizator_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE galerie_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE event_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE exponat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE bilet_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE angajat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE orar_seq START WITH 1 INCREMENT BY 1;

```

Sequence UTILIZATOR_SEQ created.

Sequence GALERIE_SEQ created.

Sequence EVENT_SEQ created.

Sequence EXPONAT_SEQ created.

Sequence BILET_SEQ created.

Sequence ANGAJAT_SEQ created.

Sequence ORAR_SEQ created.

```
[i]CREATE TABLE Utilizator (
    utilizator_id INT DEFAULT utilizator_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(50),
    prenume VARCHAR(50),
    data_nasterere DATE,
    telefon VARCHAR(15),
    email VARCHAR(100) );
```

Script Output X | Task completed in 0.07 seconds

Table UTILIZATOR created.

```
[i]CREATE TABLE Galerie (
    galerie_id INT DEFAULT galerie_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    permanente VARCHAR(1),
    data_inchidere DATE );
```

Script Output X | Task completed in 0.025 seconds

Table GALERIE created.

```
CREATE TABLE Eveniment (
    event_id INT DEFAULT event_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    data_inceput DATE,
    data_incheiere DATE,
    descriere VARCHAR(300) );
```

Script Output X | Task completed in 0.026 seconds

Table EVENIMENT created.

```
CREATE TABLE Sponsor (
    nume_sponsor VARCHAR(100) PRIMARY KEY,
    adresa_sediu VARCHAR(255),
    fax VARCHAR(20),
    telefon VARCHAR(15),
    nume_CEO VARCHAR(100) );
```

Script Output X | Task completed in 0.026 seconds

Table SPONSOR created.

```
CREATE TABLE Categorie (
    nume_categorie VARCHAR(100) PRIMARY KEY,
    perioada VARCHAR(50),
    zona VARCHAR(50),
    descriere VARCHAR(300) );
```

Script Output X | Task completed in 0.025 seconds

Table CATEGORIE created.

```
CREATE TABLE Exponat(
    exponat_id INT DEFAULT exponat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    nume_categoria VARCHAR(100) REFERENCES Categorie(nume_categoria),
    nume VARCHAR(100),
    an INT,
    creator VARCHAR(100));
```

Script Output x | Task completed in 0.026 seconds

Table EXPONAT created.

```
CREATE TABLE Angajat (
    angajat_id INT DEFAULT angajat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    tip VARCHAR(50),
    vechime INT,
    nume VARCHAR(100),
    varsta INT,
    telefon VARCHAR(15),
    email VARCHAR(100));
```

Script Output x | Task completed in 0.028 seconds

Table ANGAJAT created.

```
CREATE TABLE Ghid (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    limba VARCHAR(50),
    salariu_ghid NUMBER(10, 2));
```

Script Output x | Task completed in 0.022 seconds

Table GHID created.

```
CREATE TABLE Supervisor (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    pozitie VARCHAR(100),
    obligatii VARCHAR(4000),
    salariu_supervizor NUMBER(10, 2));
```

Script Output X | Task completed in 0.025 seconds

Table SUPERVIZOR created.

```
CREATE TABLE Orar (
    orar_id INT DEFAULT orar_seq.NEXTVAL PRIMARY KEY,
    angajat_id INT REFERENCES Angajat(angajat_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    data_orar DATE );
```

Script Output X | Task completed in 0.024 seconds

Table ORAR created.

```
CREATE TABLE Bilet (
    bilet_id INT DEFAULT bilet_seq.NEXTVAL PRIMARY KEY,
    utilizator_id INT REFERENCES Utilizator(utilizator_id),
    tip VARCHAR(50),
    orar_id INT REFERENCES Orar(orar_id));
```

Script Output X | Task completed in 0.021 seconds

Table BILET created.

```
CREATE TABLE Bilet_Galerie (
    bilet_id INT PRIMARY KEY ,
    FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    pret_galerie NUMBER(10, 2 ) );
```

Script Output X | Task completed in 0.025 seconds

Table BILET_GALERIE created.

```
CREATE TABLE Bilet_Eveniment (
    bilet_id INT PRIMARY KEY ,
    FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
    event_id INT REFERENCES Eveniment(event_id),
    pret_eveniment NUMBER(10, 2 ) );
```

Script Output X | Task completed in 0.027 seconds

Table BILET_EVENTIMENT created.

```
CREATE TABLE Supervizat_Supervizor (
    angajat_id_supervizat INT,
    angajat_id_supervizor INT,
    PRIMARY KEY(angajat_id_supervizat,angajat_id_supervizor),
    FOREIGN KEY(angajat_id_supervizat) REFERENCES Angajat (angajat_id),
    FOREIGN KEY(angajat_id_supervizor) REFERENCES Angajat(angajat_id));
```

Script Output X | Task completed in 0.019 seconds

Table SUPERVIZAT_SUPERVIZOR created.

```
CREATE TABLE Galerie_Eveniment (
    galerie_id INT,
    event_id INT,
    PRIMARY KEY(galerie_id,event_id),
    FOREIGN KEY(galerie_id) REFERENCES Galerie (galerie_id),
    FOREIGN KEY(event_id) REFERENCES Eveniment(event_id));
```

Script Output X
| Task completed in 0.026 seconds

Table GALERIE EVENIMENT created.

```
CREATE TABLE Eveniment_Sponsor (
    event_id INT,
    nume_sponsor VARCHAR(100),
    PRIMARY KEY(event_id,nume_sponsor),
    FOREIGN KEY(event_id) REFERENCES Eveniment (event_id),
    FOREIGN KEY(nume_sponsor) REFERENCES Sponsor(nume_sponsor));
```

Script Output X
| Task completed in 0.022 seconds

Table EVENIMENT_SPONSOR created.

Cerinta 5: Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).

```
INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
```

```
('Popescu', 'Ion', TO_DATE('1985-05-15', 'YYYY-MM-DD'), '0741234567',  
'popion@yahoo.com');
```

```
INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
```

```
('Ionescu', 'Maria', TO_DATE('1990-07-20', 'YYYY-MM-DD'), '0742345678',  
'maria9ionescu@gmail.com');
```

```
INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
```

```
('Georgescu', 'Alexandru', TO_DATE('1978-03-10', 'YYYY-MM-DD'), '0743456789',  
'alexandru_georgescu@yahoo.com');
```

```
INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
```

```
('Vasilescu', 'Elena', TO_DATE('1982-11-25', 'YYYY-MM-DD'), '0744567890',  
'elena.vasilescu@gmail.com');
```

```
INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
```

```
('Marin', 'Daniel', TO_DATE('2000-01-30', 'YYYY-MM-DD'), '0745678901',  
'daniel00marin@yahoo.com');
```

```
INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
```

```
('Dacia', 'Calea Floreasca 166, Bucuresti', '0211234567', '0721123456', 'Christophe Dridi');
```

```
INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
```

```
('BCR', 'Bd. Regina Elisabeta 5, Bucuresti', '0212345678', '0722234567', 'Sergiu Manea');
```

```
INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
```

```
('Petrom', 'Str. Coralilor 22, Bucuresti', '0213456789', '0723345678', 'Christina Verchere');
```

```
INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
```

```
('Bitdefender', 'Str. Barbu Vacarescu 54, Bucuresti', '0214567890', '0724456789', 'Florin Talpes');
```

```
INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
```

```
('eMAG', 'Str. Bdul. Pierre de Coubertin 3-5, Bucuresti', '0215678901', '0725567890', 'Iulian Stanciu');
```

```
INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
```

(1, 'Pictura sec XIX', 'Ciobanas cu turma de oi', 1895, 'Nicolae Grigorescu');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES

(2, 'Sculptura sec XIX', 'Domnitorul Alexandru Ioan Cuza', 1860, 'Ion Georgescu');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES

(3, 'Arta Populara sec XX', 'Covor Oltenesc', 1920, 'Artisan Popular');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES

(4, 'Fotografie sec XX', 'Revolutia din 1989', 1989, 'Fotoreporter Necunoscut');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES

(5, 'Arta Contemporana', 'Ingerul', 2005, 'Ion Barladeanu');

INSERT INTO EXPOONAT (GALERIE_ID, NUME_CATEGORIE, NUME, AN, CREATOR)
VALUES ('21', 'Pictura Moderna', 'Clar de luna', '1990', 'ION TUCULESCU');

INSERT INTO EXPOONAT (GALERIE_ID, NUME_CATEGORIE, NUME, AN, CREATOR)
VALUES ('2', 'Sculptura sec XIX', 'Bust Stefan cel Mare', '1850', 'ION GEORGESCU');

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES

('Galeria de Arta Moderna', 'Y', NULL);

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES

('Galeria de Istorie Antica', 'N', TO_DATE('2023-12-31', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES

('Galeria de Stiinta', 'Y', NULL);

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES

('Galeria de Arta Contemporana', 'N', TO_DATE('2024-06-30', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES

('Galeria de Fotografie', 'Y', NULL);

INSERT INTO GALERIE (NUME, PERMANENTA) VALUES ('Galeria Art Safari', 'Y');

INSERT INTO Categorie (nume_categorie, perioada, zona, descriere) VALUES

('Pictura sec XIX', '1800-1900', 'Europa de Est', 'Picturi din secolul XIX din Europa de Est');

INSERT INTO Categorie (nume_categorie, perioada, zona, descriere) VALUES

('Sculptura sec XIX', '1800-1900', 'Europa de Est', 'Sculpturi din secolul XIX din Europa de Est');

```
INSERT INTO Categorie (nume_categorie, perioada, zona, descriere) VALUES
('Arta Populara sec XX', '1900-2000', 'Romania', 'Obiecte de arta populara romaneasca din
secolul XX');

INSERT INTO Categorie (nume_categorie, perioada, zona, descriere) VALUES
('Fotografie sec XX', '1900-2000', 'Romania', 'Fotografii istorice din Romania secolului XX');

INSERT INTO Categorie (nume_categorie, perioada, zona, descriere) VALUES
('Arta Contemporana', '2000-prezent', 'Romania', 'Lucrari de arta contemporana din Romania');

INSERT INTO CATEGORIE (NUME_CATEGORIE, PERIOADA, ZONA, DESCRIERE)
VALUES ('Pictura Moderna', '1900-2000', 'Romania', 'O expozitie de lucrari din perioada
Moderna a Romaniei');

INSERT INTO CATEGORIE (NUME_CATEGORIE, PERIOADA, ZONA, DESCRIERE)
VALUES ('Murale din Oceania', 'Necunoscuta', 'Oceania', 'O colectie de lucrari murale aduse din
Oceania ');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Noaptea Muzeelor', TO_DATE('2024-05-18', 'YYYY-MM-DD'), TO_DATE('2024-05-18',
'YYYY-MM-DD'), 'Eveniment special cu acces gratuit.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Expozitia de Vara', TO_DATE('2024-06-01', 'YYYY-MM-DD'), TO_DATE('2024-08-31',
'YYYY-MM-DD'), 'Expozitie temporara de vara.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Festivalul de Arta', TO_DATE('2024-09-15', 'YYYY-MM-DD'), TO_DATE('2024-09-30',
'YYYY-MM-DD'), 'Festival de arta contemporana.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Saptamana Iстoriei', TO_DATE('2024-10-01', 'YYYY-MM-DD'), TO_DATE('2024-10-07',
'YYYY-MM-DD'), 'Saptamana dedicata istoriei.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Zilele Culorii', TO_DATE('2024-11-10', 'YYYY-MM-DD'), TO_DATE('2024-11-20', 'YYYY-
MM-DD'), 'Eveniment dedicat oricarui tip de pată de culoare.');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1,
'ghid', 3, 'Ivan Ion', 32, '0741122334', 'ion_ivan13@yahoo.com');
```

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 5, 'Vladimirescu Cornel', 28, '0755566778', 'corneliusvlad@gmail.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 2, 'Grigorescu Ionel', 40, '0766677889', 'ionel-grigorita@yahoo.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 4, 'Negoita Florin', 35, '0777788990', 'florinita@gmail.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 1, 'Dragulici Marin', 26, '0788899001', 'dragurin@yahoo.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 6, 'Popa Vlad', 45, '0741000000', 'vladivici.popa@yahoo.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 8, 'Veira Radu', 50, '0755000000', 'veiradu@gmail.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 3, 'Istratie Ionut', 35, '0766000000', 'ionut.istratie@yahoo.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 7, 'Stanciu Mario', 48, '0777000000', 'ggmariom8stanciu@gmail.com');

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 2, 'Andreea Dumitrescu', 30, '0788000000', 'andreea.dumitrescu@yahoo.com');

INSERT INTO ANGAJAT (GALERIE_ID, TIP, VECHIME, NUME, VARSTA, TELEFON, EMAIL) VALUES ('2', 'supervizor', '10', 'Ivan Ion', '50', '0753254325', 'ivanovick09@gmail.com');

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (5, 'Engleză', 3500.50);

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (6, 'Germană', 3700.75);

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (7, 'Franceză', 3400.00);

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (8, 'Spaniolă', 3600.20);

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (9, 'Italiană', 3550.80);

INSERT INTO SUPERVIZOR (ANGAJAT_ID, POZITIE, OBLIGATII, SALARIU_SUPERVIZOR) VALUES ('21', 'Curator', 'Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor', '5800');

INSERT INTO Supervizor (angajat_id, pozitie, obligatii, salariu_supervizor)

VALUES (10, 'Supervizor de galerie', 'Coordonarea personalului, planificarea și organizarea evenimentelor și expozițiilor, gestionarea operațiunilor zilnice ale galeriei', 4500.00);

INSERT INTO Supervizor (angajat_id, pozitie, obligatii, salariu_supervizor)

VALUES (11, 'Curator', 'Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor', 4700.00);

INSERT INTO Supervizor (angajat_id, pozitie, obligatii, salariu_supervizor)

VALUES (12, 'Manager de muzeu', 'Supervizarea întregii activități a muzeului, elaborarea strategiilor de dezvoltare, gestionarea resurselor umane și financiare', 5000.00);

INSERT INTO Supervizor (angajat_id, pozitie, obligatii, salariu_supervizor)

VALUES (13, 'Conservator', 'Restaurarea și conservarea operelor de artă, elaborarea planurilor de conservare și restaurare, documentarea și cercetarea obiectelor de artă', 4800.00);

INSERT INTO Supervizor (angajat_id, pozitie, obligatii, salariu_supervizor)

VALUES (14, 'Manager Human Resources', 'Organizarea personalului și rezolvarea oricăror conflicte interne ce pot apărea de-a lungul timpului.', 4600.00);

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (5, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (6, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (7, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (8, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (9, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 1);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 2);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 3);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'galerie', 4);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 5);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 1);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 2);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'eveniment', 3);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 4);

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 5);

```
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (1, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (2, 3, 60.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (3, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (4, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (5, 3, 60.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (6, 1, 0.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (7, 1, 0.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (8, 5, 75.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (9, 3, 80.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (10, 4, 70.00);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (6, 12);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 1);
```

```

INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 5);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 3);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 4);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (4, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 2);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 4);
INSERT INTO GALERIE_EVENTIMENT (GALERIE_ID, EVENT_ID) VALUES ('21', '2');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'Petrom');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'Bitdefender');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Bitdefender');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Popescu', 'Ion', TO_DATE('1985-05-15', 'YYYY-MM-DD'), '0741234567', 'popion@yahoo.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Ionescu', 'Maria', TO_DATE('1990-07-20', 'YYYY-MM-DD'), '0742345678', 'maria9ionescu@gmail.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Georgescu', 'Alexandru', TO_DATE('1978-03-10', 'YYYY-MM-DD'), '0743456789', 'alexandru_georgescu@yahoo.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Vasilescu', 'Elena', TO_DATE('1982-11-25', 'YYYY-MM-DD'), '0744567890', 'elena.vasilescu@gmail.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Marin', 'Daniel', TO_DATE('2000-01-30', 'YYYY-MM-DD'), '0745678901', 'daniel100marin@yahoo.com');

```

	UTILIZATOR_ID	NUME	PRENUME	DATA_NASTERE	TELEFON	EMAIL
1	1	Popescu	Ion	15-MAY-85	0741234567	popion@yahoo.com
2	2	Ionescu	Maria	20-JUL-90	0742345678	maria9ionescu@gmail.com
3	3	Georgescu	Alexandru	10-MAR-78	0743456789	alexandru_georgescu@yahoo.com
4	4	Vasilescu	Elena	25-NOV-82	0744567890	elena.vasilescu@gmail.com
5	5	Marin	Daniel	30-JAN-00	0745678901	daniel00marin@yahoo.com

```

INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
('Dacia', 'Calea Floreasca 166, Bucuresti', '0211234567', '0721123456', 'Christophe Dridi');

INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
('BCR', 'Bd. Regina Elisabeta 5, Bucuresti', '0212345678', '0722234567', 'Sergiu Manea');

INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
('Petrom', 'Str. Coralilor 22, Bucuresti', '0213456789', '0723345678', 'Christina Verchere');

INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
('Bitdefender', 'Str. Barbu Vacarescu 54, Bucuresti', '0214567890', '0724456789', 'Florin Talpes');

INSERT INTO Sponsor (nume_sponsor, adresa_sediu, fax, telefon, nume_CEO) VALUES
('eMAG', 'Str. Bdul. Pierre de Coubertin 3-5, Bucuresti', '0215678901', '0725567890', 'Iulian Stanciu');

```

NUME_SPONSOR	ADRESA_SEDIU	FAX	TELEFON	NUME_CEO
1 Dacia	Calea Floreasca 166, Bucuresti	0211234567	0721123456	Christophe Dridi
2 BCR	Bd. Regina Elisabeta 5, Bucuresti	0212345678	0722234567	Sergiu Manea
3 Petrom	Str. Coralilor 22, Bucuresti	0213456789	0723345678	Christina Verchere
4 Bitdefender	Str. Barbu Vacarescu 54, Bucuresti	0214567890	0724456789	Florin Talpes
5 eMAG	Str. Bdul. Pierre de Coubertin 3-5, Bucuresti	0215678901	0725567890	Iulian Stanciu

```

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(1, 'Pictura sec XIX', 'Ciobanasi cu turma de oi', 1895, 'Nicolae Grigorescu');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(2, 'Sculptura sec XIX', 'Domnitorul Alexandru Ioan Cuza', 1860, 'Ion Georgescu');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(3, 'Arta Populara sec XX', 'Covor Oltenesc', 1920, 'Artisan Popular');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(4, 'Fotografie sec XX', 'Revolutia din 1989', 1989, 'Fotoreporter Necunoscut');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(5, 'Arta Contemporana', 'Ingerul', 2005, 'Ion Barladeanu');

```

EXONAT_ID	GALERIE_ID	NUME_CATEGORIE	NUME	AN	CREATOR
1	6	1 Pictura sec XIX	Ciobanasi cu turma de oi	1895	Nicolae Grigorescu
2	7	2 Sculptura sec XIX	Domnitorul Alexandru Ioan Cuza	1860	Ion Georgescu
3	8	3 Arta Populara sec XX	Covor Oltenesc	1920	Artisan Popular
4	9	4 Fotografie sec XX	Revolutia din 1989	1989	Fotoreporter Necunoscut
5	10	5 Arta Contemporana	Ingerul	2005	Ion Barladeanu

```

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES
('Galeria de Arta Moderna', 'Y', NULL);

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES
('Galeria de Istorie Antica', 'N', TO_DATE('2023-12-31', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES
('Galeria de Stiinta', 'Y', NULL);

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES
('Galeria de Arta Contemporana', 'N', TO_DATE('2024-06-30', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, permanenta, data_incheiere) VALUES
('Galeria de Fotografie', 'Y', NULL);

```

GALERIE_ID	NUME	PERMANENTA	DATA_INCHEIERE
1	1 Galeria de Arta Moderna	Y	(null)
2	2 Galeria de Istorie Antica	N	31-DEC-23
3	3 Galeria de Stiinta	Y	(null)
4	4 Galeria de Arta Contemporana	N	30-JUN-24
5	5 Galeria de Fotografie	Y	(null)

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES ('Noaptea Muzeelor', TO_DATE('2024-05-18', 'YYYY-MM-DD'), TO_DATE('2024-05-18', 'YYYY-MM-DD'), 'Eveniment special cu acces gratuit.');				
INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES ('Expozitia de Vara', TO_DATE('2024-06-01', 'YYYY-MM-DD'), TO_DATE('2024-08-31', 'YYYY-MM-DD'), 'Expozitie temporara de vara.');				
INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES ('Festivalul de Arta', TO_DATE('2024-09-15', 'YYYY-MM-DD'), TO_DATE('2024-09-30', 'YYYY-MM-DD'), 'Festival de arta contemporana.');				
INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES ('Saptamana Iстoriei', TO_DATE('2024-10-01', 'YYYY-MM-DD'), TO_DATE('2024-10-07', 'YYYY-MM-DD'), 'Saptamana dedicata istoriei.');				
INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES ('Zilele Culorii', TO_DATE('2024-11-10', 'YYYY-MM-DD'), TO_DATE('2024-11-20', 'YYYY-MM-DD'), 'Eveniment dedicat tuturor tipurilor de pată de culoare.');				
EVENT_ID	NUME	DATA_INCEPUT	DATA_INCHEIERE	DESCRIERE
1	1 Noaptea Muzeelor	18-MAY-24	18-MAY-24	Eveniment special cu acces gratuit.
2	2 Expozitia de Vara	01-JUN-24	31-AUG-24	Expozitie temporara de vara.
3	3 Festivalul de Arta	15-SEP-24	30-SEP-24	Festival de arta contemporana.
4	4 Saptamana Iстoriei	01-OCT-24	07-OCT-24	Saptamana dedicata istoriei.
5	5 Zilele Culorii	10-NOV-24	20-NOV-24	Eveniment dedicat tuturor tipurilor de pată de culoare.

```

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Pictura sec XIX', '1800-1900', 'Europa de Est', 'Picturi din secolul XIX din Europa de Est');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Sculptura sec XIX', '1800-1900', 'Europa de Est', 'Sculpturi din secolul XIX din Europa de Est');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Arta Populara sec XX', '1900-2000', 'Romania', 'Obiecte de arta populara romaneasca din secolul XX');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Fotografie sec XX', '1900-2000', 'Romania', 'Fotografii istorice din Romania secolului XX');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Arta Contemporana', '2000-prezent', 'Romania', 'Lucrari de arta contemporana din Romania');

```

NUME_CATEGORIE	PERIOADA	ZONA	DESCRIERE
1 Pictura sec XIX	1800-1900	Europa de Est	Picturi din secolul XIX din Europa de Est
2 Sculptura sec XIX	1800-1900	Europa de Est	Sculpturi din secolul XIX din Europa de Est
3 Arta Populara sec XX	1900-2000	Romania	Obiecte de arta populara romaneasca din secolul XX
4 Fotografie sec XX	1900-2000	Romania	Fotografii istorice din Romania secolului XX
5 Arta Contemporana	2000-prezent	Romania	Lucrari de arta contemporana din Romania

```

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 3, 'Ivan Ion', 32, '0741122334', 'ion_ivan13@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 5, 'Vladimirescu Cornel', 28, '0755566778', 'corneliusvlad@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 2, 'Grigorescu Ionel', 40, '0766677889', 'ionel-grigorita@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 4, 'Negoita Florin', 35, '0777788990', 'florinita@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 1, 'Dragulici Marin', 26, '0788899001', 'dragurin@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 6, 'Popa Vlad', 45, '0741000000', 'vladivici.popa@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 8, 'Veira Radu', 50, '0755000000', 'veiradu@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 3, 'Istratie Ionut', 35, '0766000000', 'ionut.istratie@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 7, 'Stanciu Mario', 48, '0777000000', 'ggmariom@stanciu@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 2, 'Andreea Dumitrescu', 30, '0788000000', 'andreea.dumitrescu@yahoo.com')

```

ANGAJAT_ID	GALERIE_ID	TIPI	VECHIME	NUME	VARSTA	TELEFON	EMAIL
1	5	1 ghid	3	Ivan Ion	32	0741122334	ion_ivan13@yahoo.com
2	6	3 ghid	5	Vladimirescu Cornel	28	0755566778	corneliusvlad@gmail.com
3	7	1 ghid	2	Grigorescu Ionel	40	0766677889	ionel-grigorita@yahoo.com
4	8	1 ghid	4	Negoita Florin	35	0777788990	florinita@gmail.com
5	9	3 ghid	1	Dragulici Marin	26	0788899001	dragurin@yahoo.com
6	10	1 supervizor	6	Popa Vlad	45	0741000000	vladivici.popa@yahoo.com
7	11	1 supervizor	8	Veira Radu	50	0755000000	veiradu@gmail.com
8	12	3 supervizor	3	Istratie Ionut	35	0766000000	ionut.istratie@yahoo.com
9	13	3 supervizor	7	Stanciu Mario	48	0777000000	ggmariom@stanciu@gmail.com
10	14	1 supervizor	2	Andreea Dumitrescu	30	0788000000	andreea.dumitrescu@yahoo.com

```

INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (5, 'Engleză', 3500.50);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (6, 'Germană', 3700.75);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (7, 'Franceză', 3400.00);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (8, 'Spaniolă', 3600.20);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (9, 'Italiană', 3550.80);

```

ANGAJAT_ID	LIMBA	SALARIU_GHID
1	5 Engleză	3500.5
2	6 Germană	3700.75
3	7 Franceză	3400
4	8 Spaniolă	3600.2
5	9 Italiană	3550.8

```

INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (10, 'Supervisor de galerie', 'Coordonarea personalului, planificarea și organizarea evenimentelor și expozițiilor, gestionarea operațiunilor zilnice ale galeriei');
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (11, 'Curator', 'Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor', 4700.00);
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (12, 'Manager de muzeu', 'Supervizarea întregii activități a muzeului, elaborarea strategiilor de dezvoltare, gestionarea resurselor umane și financiare', 5000.00);
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (13, 'Conservator', 'Restaurarea și conservarea operelor de artă, elaborarea planurilor de conservare și restaurare, documentarea și cercetarea obiectelor de artă');
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (14, 'Manager Human Resources', 'Organizarea personalului și rezolvarea oricărora conflicte interne ce pot apărea de-a lungul timpului.', 4600.00);

```

ANGAJAT_ID	POZITIE	OBLIGATII
1	10 Supervisor de galerie	Coordonarea personalului, planificarea și organizarea evenimentelor și expozițiilor, gestionarea operațiunilor zilnice ale galeriei
2	11 Curator	Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor
3	12 Manager de muzeu	Supervizarea întregii activități a muzeului, elaborarea strategiilor de dezvoltare, gestionarea resurselor umane și financiare
4	13 Conservator	Restaurarea și conservarea operelor de artă, elaborarea planurilor de conservare și restaurare, documentarea și cercetarea obiectelor de artă
5	14 Manager Human Resources	Organizarea personalului și rezolvarea oricărora conflicte interne ce pot apărea de-a lungul timpului.

```

INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (5, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (6, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (7, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (8, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (9, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));

```

	ORAR_ID	ANGAJAT_ID	GALERIE_ID	DATA_ORAR
1	1	5	1	18-MAY-24
2	2	6	3	18-MAY-24
3	3	7	1	18-MAY-24
4	4	8	1	18-MAY-24
5	5	9	3	18-MAY-24

```

INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 1);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 2);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 3);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'galerie', 4);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 5);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 1);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 2);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'eveniment', 3);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 4);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 5);

```

	BILET_ID	UTILIZATOR_ID	TIP	ORAR_ID
1	1	1	galerie	1
2	2	2	galerie	2
3	3	3	galerie	3
4	4	4	galerie	4
5	5	5	galerie	5
6	6	6	eveniment	1
7	7	7	eveniment	2
8	8	8	eveniment	3
9	9	9	eveniment	4
10	10	10	eveniment	5

```

INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (1, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (2, 3, 60.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (3, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (4, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (5, 3, 60.00);

```

	BILET_ID	GALERIE_ID	PRET_GALERIE
1	1	1	50
2	2	3	60
3	3	1	50
4	4	1	50
5	5	3	60

```

INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (6, 1, 0.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (7, 1, 0.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (8, 5, 75.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (9, 3, 80.00);
INSERT INTO Bilet_Eveniment (bilet_id, event_id, pret_eveniment) VALUES (10, 4, 70.00);

```

	BILET_ID	EVENT_ID	PRET_EVENIMENT
1	6	1	0
2	7	1	0
3	8	5	75
4	9	3	80
5	10	4	70

```

INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (6, 12);

```

	ANGAJAT_ID_SUPERVIZAT	ANGAJAT_ID_SUPERVIZOR
1	5	10
2	5	11
3	5	14
4	6	12
5	7	10
6	7	11
7	7	14
8	8	10
9	8	11
10	8	14

```

INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 5);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 3);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 4);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (4, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 2);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 4);

```

	GALERIE_ID	EVENT_ID
1	1	1
2	1	3
3	1	5
4	2	1
5	2	2
6	3	1
7	3	4
8	4	1
9	5	1
10	5	4

```

INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'Petrom');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'Bitdefender');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Bitdefender');

```

	EVENT_ID	NUME_SPONSOR
1	1	BCR
2	1	eMAG
3	2	BCR
4	2	Petrom
5	3	Bitdefender
6	3	eMAG
7	4	Dacia
8	4	eMAG
9	5	Bitdefender
10	5	Dacia

Cerinta 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.

Sa se creeze o functie ce intoarce profitul obtinut din biletele cumpărate pentru un eveniment sau galerie in care exista cel putin un exponent ce face parte dintr-o categorie data ca parametru.

```
create or replace function Profit (categ in categorie.nume_categoria%type) return number as
```

```
    type IndexedCollection is table of galerie.galerie_id%type index by pls_integer;
    type NestedCollection is table of eveniment.event_id%type;
    type Vector is varray(100) of bilet_galerie.pret_galerie%type;
    galerii IndexedCollection;
    evenimente NestedCollection := NestedCollection();
    preturi Vector := Vector();
    profitTotal number := 0;
    exista number;
    ceva categorie.zona%type;

begin
    -- Verificarea existentei respectivei categorii
    select zona into ceva from categorie where upper(nume_categoria)=upper(categ);
    --Obtinerea listei galeriilor ce contin exponate cu categoria data ca parametru
    select distinct galerie_id bulk collect into galerii from exponat where upper(nume_categoria) =
    upper(categ);
    if galerii.count = 0 then
        raise_application_error(-20002,'Nu exista galerie ce sa aiba o categorie cu acel nume');
    end if;
    --Pentru lista de galerii se obtine in colectii:
    for i in 1..galerii.count loop
        --Lista de evenimente ce contin galeriile in care exista exponate din respectiva categorie
```

```

for r in (select distinct event_id from galerie_eveniment where galerie_id = galerii(i)) loop
    evenimente.extend;
    evenimente(evenimente.count) := r.event_id;
end loop;

--Lista cu preturile biletelor pentru galeriile din acea lista
for r in (select pret_galerie from bilet_galerie where galerie_id = galerii(i)) loop
    if preturi.count < preturi.limit then
        preturi.extend;
        preturi(preturi.count) := r.pret_galerie;
    else
        raise_application_error(-20001, 'Nu se respecta limita sueroioara a vectorului preturi');
    end if;
end loop;
end loop;

--Se adauga si biletele pentru evenimentele obtinute mai devreme
for i in 1..evenimente.count loop
    for r in (select pret_eveniment from bilet_eveniment where event_id = evenimente(i)) loop
        if preturi.count < preturi.limit then
            preturi.extend;
            preturi(preturi.count) := r.pret_eveniment;
        else
            raise_application_error(-20001, 'Nu se respecta limita sueroioara a vectorului preturi');
        end if;
    end loop;
end loop;

--Se calculeaza profitul total
for i in 1..preturi.count loop
    profitTotal := profitTotal + preturi(i);

```

```

end loop;

return profitTotal;

exception
when no_data_found then
    raise_application_error(-20003,'Nu exista o astfel de categorie');

end;
/

```

```

create or replace function Profit (categ in categorie.nume_categoria#type) return number as
type IndexedCollection is table of galerie.galerie_id#type index by pls_integer;
type NestedCollection is table of eveniment.event_id#type;
type Vector is varray(100) of bilet_galerie.pret_galerie#type;
galerii IndexedCollection;
evenimente NestedCollection := NestedCollection();
preturi Vector := Vector();
profitTotal number := 0;
exista number;
ceva categorie.zona#type;
begin
--Verificarea existentei respectivei categorii
select zona into ceva from categorie where upper(nume_categoria)=upper(categ);
--Obtinerea listei galeriilor ce contin exponate cu categoria data ca parametru
select distinct galerie_id bulk collect into galerii from exponat where upper(nume_categoria) = upper(categ);
if galerii.count = 0 then
    raise_application_error(-20002,'Nu exista galerie ce sa aiba o categorie cu acel nume');
end if;
--Pentru lista de galerii se obtine in colectii:
for i in 1..galerii.count loop
--Lista de evenimente ce contin galeriile in care exista exponate din respectiva categorie
    for r in (select distinct event_id from galerie_eveniment where galerie_id = galerii(i)) loop
        evenimente.extend;
        evenimente(evenimente.count) := r.event_id;
    end loop;
--Lista cu preturile biletelor pentru galeriile din acea lista

```

```

    for r in (select distinct event_id from galerie_eveniment where galerie_id = galerii(i)) loop
        evenimente.extend;
        evenimente(evenimente.count) := r.event_id;
    end loop;
--Lista cu preturile biletelor pentru galeriile din acea lista
    for r in (select pret_galerie from bilet_galerie where galerie_id = galerii(i)) loop
        if preturi.count < preturi.limit then
            preturi.extend;
            preturi(preturi.count) := r.pret_galerie;
        else
            raise_application_error(-20001, 'Nu se respecta limita suoerioara a vectorului preturi');
            end if;
        end loop;
    end loop;
--Se adauga si biletele pentru evenimentele obtinute mai devreme
    for i in 1..evenimente.count loop
        for r in (select pret_eveniment from bilet_eveniment where event_id = evenimente(i)) loop
            if preturi.count < preturi.limit then
                preturi.extend;
                preturi(preturi.count) := r.pret_eveniment;
            else
                raise_application_error(-20001, 'Nu se respecta limita suoerioara a vectorului preturi');
                end if;
            end loop;
        end loop;
--Se calculeaza profitul total

```

```

    for i in 1..preturi.count loop
        profitTotal := profitTotal + preturi(i);
    end loop;

    return profitTotal;
exception
    when no_data_found then
        raise_application_error(-20003,'Nu exista o astfel de categorie');
end;

```

Script Output x | Task completed in 0.068 seconds

Function PROFIT compiled

Afisarea si apelarea se efectueaza utilizand urmatorul bloc :

```

declare
    categ categorie.nume_categorie%type:='&p_num';
    galerie_inexistenta exception;
    limita_depasita exception;
    no_categorie exception;
    pragma exception_init(no_categorie,-20003);
    pragma exception_init(galerie_inexistenta,-20002);
    pragma exception_init(limita_depasita,-20001);

begin
    dbms_output.put_line(profit('categ'));

exception
    when galerie_inexistenta then
        dbms_output.put_line('Nu exista vreun exponat din categoria primita ca parametru.');
    when limita_depasita then
        dbms_output.put_line('Exista prea multe bilete pentru galerii sau evenimente in care exista exponate din categoria primita ca parametru');
    when no_categorie then
        dbms_output.put_line('Nu exista categoria data ca parametru');
end;

```

```

/
declare
    categ categorie.nume_categorie%type:='&p_num';
    galerie_inexistenta exception;
    limita_depasita exception;
    no_categorie exception;
    pragma exception_init(no_categorie,-20003);
    pragma exception_init(galerie_inexistenta,-20002);
    pragma exception_init(limita_depasita,-20001);
begin
    dbms_output.put_line(profit('categ'));
exception
    when galerie_inexistenta then
        dbms_output.put_line('Nu exista vreun exponat din categoria primita ca parametru.');
    when limita_depasita then
        dbms_output.put_line('Exista prea multe bilete pentru galerii sau evenimente in care exista exponate din categoria primita ca parametru');
    when no_categorie then
        dbms_output.put_line('Nu exista categoria data ca parametru');
end;
/

```

Daca categ este Sculptura sec XIX apare ca output 190 :

```

        dbms_output.put_line('Nu exista categoria data ca parametru');
end;

PL/SQL procedure successfully completed.

```



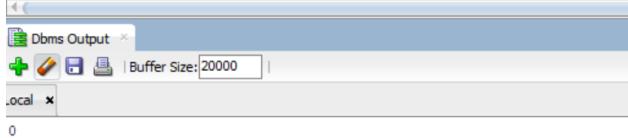
Daca categ este Pictura Moderna apare ca output 0, deoarece nu exista vreun bilet pentru galeria sau evenimentul in care exista un exponat facand parte din respectiva categorie :

```

        dbms_output.put_line('Nu exista categoria data ca parametru');
end;

PL/SQL procedure successfully completed.

```



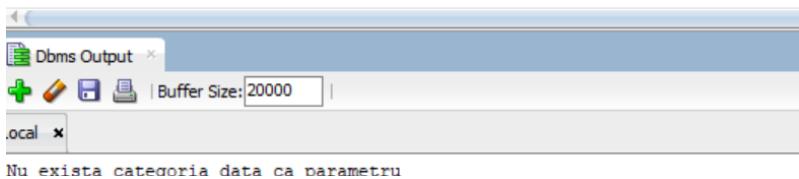
Daca introducem o categorie ce nu exista apare :

```

        dbms_output.put_line('Nu exista categoria data ca parametru');
end;

PL/SQL procedure successfully completed.

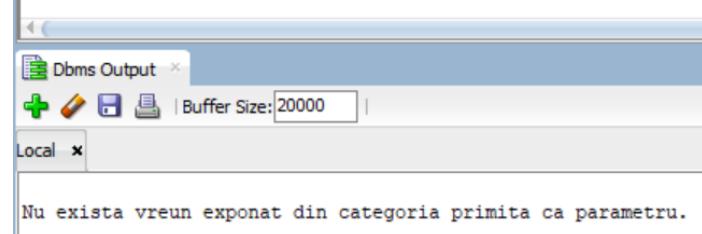
```



Daca se introduce Murale din Oceania se va descoperi ca nu exista galerii ce sa aibe exponate din acea categorie :

```
end;

PL/SQL procedure successfully completed.



Nu exista vreun exponat din categoria primita ca parametru.


```

Cerinta 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.

Pentru numele unui ghid primit ca parametru să se afiseze profitul adus de acesta (suma totală a biletelor ce au ca programul în cadrul orarului ghidului).

```
create or replace function ghidProfit(numele in angajat.nume%type) return number as
```

```
-- Cursor parametrizat pentru a extrage prețurile biletelor asociate unui orar specific.
```

```
cursor obtPretGal(idul orar.orar_id%type) is
```

```
select bg.pret_galerie from bilet_galerie bg
```

```
join bilet b on b.bilet_id=bg.bilet_id where b.orar_id=idul;
```

```
idul orar.orar_id%type;
```

```
profit number(4):=0;
```

```
idAngajat angajat.angajat_id%type;
```

```
tipul angajat.tip%type;
```

```
begin
```

```
-- Preia ID-ul și tipul angajatului pe baza numelui dat ca parametru.
```

```
select angajat_id, nvl(tip,"") into idAngajat,tipul from angajat
```

```
where trim(upper(nume))=trim((upper(numele)));
```

```
-- Verifica dacă angajatul este de tip "GHID"; altfel, arunca o excepție.
```

```
if trim(upper(tipul))<>'GHID' then
```

```
    raise_application_error(-20001,'Numele primit ca parametru nu aparține unui Ghid');
```

```
end if;
```

```
-- Iterează prin toate orarele asociate angajatului dat.
```

```
for orar_rec in (select o.orar_id from orar o
```

```
        join angajat a on a.angajat_id=o.angajat_id
```

```
        where a.nume=numele) loop
```

```

idul:=orar_rec.orar_id;
-- Aduna profitul din biletele de galerie asociate acestui orar.
for pret_rec in obtPretGal(idul) loop
    profit:=profit+pret_rec.pret_galerie;
end loop;
-- Aduna profitul din biletele de eveniment asociate acestui orar.
for pret_rec in (select be.pret_eveniment from bilet_eveniment be
                 join bilet b on b.bilet_id=be.bilet_id where b.orar_id=idul) loop
    profit:=profit+pret_rec.pret_eveniment;
end loop;
end loop;
return profit;
exception
when no_data_found then
    raise_application_error(-20000,'Nu exista angajat cu acest nume');
when too_many_rows then
    raise_application_error(-20002,'Exista mai multi angajati cu acest nume');
end;

create or replace function ghidProfit(numele in angajat.nume%type) return number as
-- Cursor parametrizat pentru a extrage preturile biletelelor asociate unui orar specific.
cursor obtPretGal(idul orar.orar_id%type) is
    select bg.pret_galerie from bilet_galerie bg
        join bilet b on b.bilet_id=bg.bilet_id where b.orar_id=idul;
idul orar.orar_id%type;
profit number(4):=0;
idAngajat angajat.angajat_id%type;
tipul angajat.tip%type;
begin
-- Preia ID-ul si tipul angajatului pe baza numelui dat ca parametru.
select angajat_id, nvl(tip,'') into idAngajat,tipul from angajat
    where trim(upper(nume))=trim((upper(numele)));
-- Verifica daca angajatul este de tip "GHID"; altfel, arunca o exceptie.
if trim(upper(tipul))<>'GHID' then
    raise_application_error(-20001,'Numele primit ca parametru nu apartine unui Ghid');
end if;
-- Itereaza prin toate orarele asociate angajatului dat.
for orar_rec in (select o.orar_id from orar o
                 join angajat a on a.angajat_id=o.angajat_id
                 where a.nume=numele) loop
    idul:=orar_rec.orar_id;
    -- Aduna profitul din biletele de galerie asociate acestui orar.
    for pret_rec in obtPretGal(idul) loop
        profit:=profit+pret_rec.pret_galerie;
    end loop;

```

```

-- Aduna profitul din biletele de eveniment asociate acestui orar.
for pret_rec in (select be.pret_eveniment from bilet_eveniment be
                  join bilet b on b.bilet_id=be.bilet_id where b.orar_id=idul) loop
    profit:=profit+pret_rec.pret_eveniment;
end loop;
end loop;
return profit;
exception
    when no_data_found then
        raise_application_error(-20000,'Nu exista angajat cu acest nume');
    when too_many_rows then
        raise_application_error(-20002,'Exista mai multi angajati cu acest nume');
end;
/

```

Script Output x | Task completed in 0.482 seconds

Function GHIDPROFIT compiled

Blocul de verificare a functiei :

declare

```

nume angajat.nume%type:='&p_nume';

no_Angajat exception;

not_Ghid exception;

many_Angajati exception;

pragma exception_init(no_Angajat,-20000);

pragma exception_init(not_Ghid,-20001);

pragma exception_init(many_Angajati,-20002);

begin

dbms_output.put_line('Profitul adus de ghidul'||nume||' este de '|ghidProfit(nume));

exception

when no_Angajat then

    dbms_output.put_line('Nu exista vreun angajat cu numele'||nume);

when not_Ghid then

    dbms_output.put_line('Angajatul'||nume||' nu este un ghid.');

when many_Angajati then

    dbms_output.put_line('Exista mai multi angajati cu numele'||nume);

when others then

```

```

dbms_output.put_line('Eroare necunoscuta.');

end;

/

declare
    nume angajat.nume%type:='sp_nume';
    no_Angajat exception;
    not_Ghid exception;
    many_Angajati exception;
    pragma exception_init(no_Angajat,-20000);
    pragma exception_init(not_Ghid,-20001);
    pragma exception_init(many_Angajati,-20002);
begin
    dbms_output.put_line('Profitul adus de ghidul'||nume||' este de '|ghidProfit(nume));
exception
    when no_Angajat then
        dbms_output.put_line('Nu exista vreun angajat cu numele'||nume);
    when not_Ghid then
        dbms_output.put_line('Angajatul'||nume||' nu este un ghid.');
    when many_Angajati then
        dbms_output.put_line('Exista mai multi angajati cu numele'||nume);
    when others then
        dbms_output.put_line('Eroare necunoscuta.');
end;
/

```

Daca numele introdus este Vladimirescu Cornel, atunci outputul va fi :

```

PL/SQL procedure successfully completed.

Dbms Output x
+ - Buffer Size: 20000
Local x

Profitul adus de ghidul Vladimirescu Cornel este de 150

```

Daca numele introdus este Ivan Ion atunci apare outputul, deoarece exista doua instante de Ivan Ion:

```

PL/SQL procedure successfully completed.

Dbms Output x
+ - Buffer Size: 20000
Local x

Exista mai multi angajati cu numele Ivan Ion

```

Daca numele introdus este Popa Vlad ce este un supervisor se va afisa :

PL/SQL procedure successfully completed.

The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. Below the title bar are standard icons for file operations and a 'Buffer Size: 20000' input field. The main area of the window contains the text: 'Angajatul Popa Vlad nu este un ghid.'

Daca numele introdus este orice string care sa nu reprezinte un angajat din tabela, de exemplu Loredana se va afisa :

PL/SQL procedure successfully completed.

The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. Below the title bar are standard icons for file operations and a 'Buffer Size: 20000' input field. The main area of the window contains the text: 'Nu există vreun angajat cu numele Loredana'.

Cerinta 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 exceptiile care pot apărea, incluzând exceptiile predefinite **NO_DATA_FOUND** și **TOO_MANY_ROWS**. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Creati o functie ce va afisa detaliile galeriei ce are mai putine orare decat cati angajati are. Daca exista mai multe galerii de acest tip, se va alege cea care are cea mai mare diferenta dintre nr-ul de orare si nr-ul de angajati. Informatiile afisate vor fi : numele galeriei, nr-ul de orare si nr-ul angajatilor.

```

create or replace function OrareSiAngajati(numele in galerie.nume%type) return varchar2 as
  type galerieRecord is record (nume galerie.nume%type, nr_orare number, nr_angajati number);
  returnare galerieRecord;
  idul galerie.galerie_id%type;
begin
  select g.galerie_id into idul from galerie g where upper(trim(numele))=upper(trim(g.nume));
  select g.nume, count(o.orar_id) as nr_orare,
    (select count(*) from angajat a where a.galerie_id=g.galerie_id) as nr_angajati
  into returnare from galerie g
  left join orar o on o.galerie_id=g.galerie_id
  group by g.nume, g.galerie_id
  having (select count(*) from angajat a where a.galerie_id=g.galerie_id)>count(o.orar_id)
  and (select count(*) from angajat a where a.galerie_id = g.galerie_id) - count(o.orar_id) =
    (select max((select count(*) from angajat a where a.galerie_id = gi.galerie_id) -
  count(oi.orar_id))
  from galerie gi
  left join orar oi on oi.galerie_id = gi.galerie_id

```

```

        group by gi.galerie_id);

if upper(trim(numele))=upper(trim(returnare.nume)) then
    return 'DA';
else
    return 'NU';
end if;

exception
when no_data_found then
    raise_application_error(-20000,'Nu exista galerie cu un astfel de nume');

when too_many_rows then
    raise_application_error(-20001,'Exista mai multe galerii ce au diferenta dintre nr de angajati
si orare maxima');

when others then
    raise_application_error(-20002,'Eroare neasteptata');

end;

```

```

create or replace function OrareSiAngajati(numele in galerie.nume%type) return varchar2 as
type galerieRecord is record (nume galerie.nume%type, nr_orare number, nr_angajati number);
returnare galerieRecord;
idul galerie.galerie_id%type;
begin
    select g.galerie_id into idul from galerie g where upper(trim(numele))=upper(trim(g.nume));
    select g.nume, count(o.orar_id) as nr_orare,
           (select count(*) from angajat a where a.galerie_id=g.galerie_id) as nr_angajati
    into returnare from galerie g
    left join orar o on o.galerie_id=g.galerie_id
    group by g.nume, g.galerie_id
    having (select count(*) from angajat a where a.galerie_id=g.galerie_id)>count(o.orar_id)
    and (select count(*) from angajat a where a.galerie_id = g.galerie_id) - count(o.orar_id) =
           (select max((select count(*) from angajat a where a.galerie_id = gi.galerie_id) - count(o.orar_id))
            from galerie gi
            left join orar oi on oi.galerie_id = gi.galerie_id
            group by gi.galerie_id);
    if upper(trim(numele))=upper(trim(returnare.nume)) then
        return 'DA';
    else
        return 'NU';
    end if;
exception
when no_data_found then
    raise_application_error(-20000,'Nu exista galerie cu un astfel de nume');
when too_many_rows then
    raise_application_error(-20001,'Exista mai multe galerii ce au diferenta dintre nr de angajati si orare maxima');

when others then
    raise_application_error(-20002,'Eroare neasteptata');
end;

```

Script Output X | Task completed in 0.027 seconds

Function ORARESIANGAJATI compiled

Blocul ce apeleaza:

```
declare
    nume galerie.nume%type:='&p_nume';
    no_Galerie exception;
    diverse exception;
    many_Galerii exception;
    pragma exception_init(no_Galerie,-20000);
    pragma exception_init(diverse,-20002);
    pragma exception_init(many_Galerii,-20001);
begin
    dbms_output.put_line(OrareSiAngajati(nume));
exception
    when no_Galerie then
        dbms_output.put_line('Nu exista vreo galerie denumita'||nume);
    when diverse then
        dbms_output.put_line('Eroare necunoscuta');
    when many_Galerii then
        dbms_output.put_line('Exista mai multe galerii cu diferenta maxima intre angajati si orare');
end;
/
```

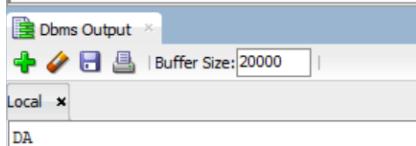
```

declare
    nume galerie.nume%type:='&p_num';
    no_Galerie exception;
    diverse exception;
    many_Galerii exception;
    pragma exception_init(no_Galerie,-20000);
    pragma exception_init(diverse,-20002);
    pragma exception_init(many_Galerii,-20001);
begin
    dbms_output.put_line(OrareSiAngajati(nume));
exception
    when no_Galerie then
        dbms_output.put_line('Nu exista vreo galerie denumita'||nume);
    when diverse then
        dbms_output.put_line('Eroare necunoscuta');
    when many_Galerii then
        dbms_output.put_line('Exista mai multe galerii cu diferenta maxima intre angajati si orare');
end;
/

```

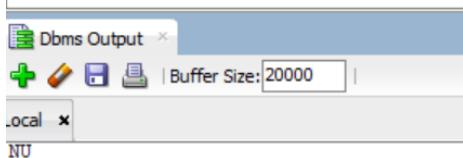
Daca input-ul este Galeria de Arta Moderna :

```
PL/SQL procedure successfully completed.
```



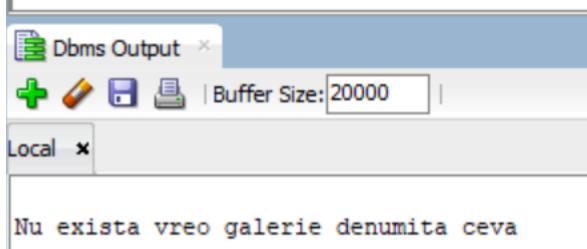
Daca input-ul este orice alta galerie din tabel :

```
PL/SQL procedure successfully completed.
```



Daca este introdus orice alt string ce nu reprezinta o galerie din baza de date:

```
PL/SQL procedure successfully completed.
```



In cazul de too_many_rows trebuie efectuate numeroase modificare in baza de date ce ar duce ca functia sa intoarca mereu too_many_rows apoi, daca stringul introdus este o galerie din tabel.

Cerinta 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 exceptii proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate

Sa se creeze o procedura in care se primesc ca parametri numele unei galerii si doua data calendaristice. Procedura va gasi evenimentele din care aceasta galerie face parte, ce se desfasoara in intervalul definit de cele doua date calendaristice, afisandu-le in parte detaliile, nr-ul de bilet vandute, si numele sponsorilor impreuna cu CEO-ul acestora.

```
create or replace procedure Galerie_Detalii ( numele in galerie.nume%type, data_s in eveniment.data_inceput%type, data_f in eveniment.data_incheiere%type) as
```

```
-- Declararea variabilelor pentru stocarea numarului de evenimente si ID-ul galeriei  
nr_evenimente number;  
id_galerie galerie.galerie_id%type;  
-- Declararea exceptiilor personalizate  
no_Events exception;  
invalid_interval exception;  
-- Asocierea codurilor de eroare pentru exceptii  
pragma exception_init(no_Events,-20002);  
pragma exception_init(invalid_interval,-20003);  
begin  
-- Cautarea ID-ului galeriei dupa nume; verificare insensibila la spatii si litere mari/mici  
select g.galerie_id into id_galerie from galerie g where  
upper(trim(g.nume))=upper(trim(numele));  
-- Verificarea daca intervalul de date este valid  
if data_s>data_f then  
    raise invalid_interval;
```

```

end if;

-- Determinarea numarului de evenimente in intervalul specificat
select count(*) into nr_evenimente from galerie_eveniment ge
join eveniment e on e.event_id=ge.event_id
where ge.galerie_id=id_galerie and ((e.data_inceput between data_s and data_f) or
(e.data_incheiere between data_s and data_f));

-- Verificarea daca exista evenimente in acest interval; daca nu, se ridica o exceptie
if nr_evenimente=0 then
    raise_application_error(-20002,'Nu exista evenimente in acest interval.');
end if;

-- Iterarea prin evenimentele gasite si afisarea detaliilor
for r in (select e.nume, e.data_inceput, e.data_incheiere,
s.nume_sponsor,s.nume_ceo,count(be.bilet_id) as nr_bilete from galerie_eveniment ge
join eveniment e on e.event_id=ge.event_id
left join eveniment_sponsor es on es.event_id=e.event_id
left join sponsor s on s.nume_sponsor=es.nume_sponsor
left join bilet_eveniment be on be.event_id=e.event_id
where ge.galerie_id=id_galerie and ((e.data_inceput between data_s and data_f) or
(e.data_incheiere between data_s and data_f))
group by e.nume, e.data_inceput, e.data_incheiere, s.nume_sponsor, s.nume_ceo) loop
    -- Afisarea informatiilor despre fiecare eveniment
    dbms_output.put_line('Eveniment: ' || r.nume ||
        ', Perioada: ' || r.data_inceput || ' - ' || r.data_incheiere ||
        ', Sponsor: ' || nvl(r.nume_sponsor, 'Nespecificat') ||
        ', CEO Sponsor: ' || nvl(r.nume_ceo, 'Nespecificat') ||
        ', Bilete vandute: ' || r.nr_bilete);
end loop;

exception
when no_data_found then

```

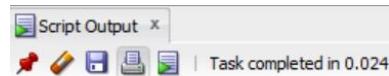
```

dbms_output.put_line('Nu exista galerie cu numele'||numele);
when too_many_rows then
    dbms_output.put_line('Exista mai multe galerii cu numele'||numele);
when no_events then
    dbms_output.put_line('Galeria'||numele||' nu este inclusa in niciun eveniment.');
when invalid_interval then
    dbms_output.put_line('Perioada oferita ca parametru nu este valida. Al doilea parametru este data de inceput, iar al treilea este data de sfarsit.');
when others then
    dbms_output.put_line('Eroare neasteptata!');
end;

create or replace procedure Galerie_Detalii ( numele in galerie.nume%type, data_s in eveniment.data_incep%type, data_f in eveniment.data_incheiere%type) as
-- Declararea variabilelor pentru stocarea numarului de evenimente si ID-ul galeriei
nr_evenimente number;
id_galerie galerie_id%type;
-- Declararea exceptiilor personalizate
no_Events exception;
invalid_interval exception;
-- Asocierea codurilor de eroare pentru exceptiile personalizate
pragma exception_init(no_Events,-20002);
pragma exception_init(invalid_interval,-20003);
begin
-- Cautarea ID-ului galeriei dupa nume; verificare insensibila la spatii si litere mari/mici
select g.galerie_id into id_galerie from galerie g where upper(trim(g.nume))=upper(trim(numele));
-- Verificarea daca intervalul de date este valid
if data_s>data_f then
    raise invalid_interval;
end if;
-- Determinarea numarului de evenimente in intervalul specificat
select count(*) into nr_evenimente from galerie_eveniment ge
join eveniment e on e.event_id=ge.event_id
left join eveniment_sponsor es on es.event_id=e.event_id
left join sponsor s on s.sponsor_id=es.sponsor_id
left join bilet_eveniment be on be.event_id=e.event_id
where ge.galerie_id=id_galerie and ((e.data_incep between data_s and data_f) or (e.data_incheiere between data_s and data_f));
-- Verificarea daca exista evenimente in acest interval; daca nu, se ridica o exceptie
if nr_evenimente=0 then
    raise_application_error(-20002,'Nu exista evenimente in acest interval.');
end if;

-- Iterarea prin evenimentele gasite si afisarea detaliilor
for r in (select e.nume, e.data_incep, e.data_incheiere, s.nume_sponsor,s.nume_ceo,count(be.bilet_id) as nr_bilete from galerie_eveniment ge
join eveniment e on e.event_id=ge.event_id
left join eveniment_sponsor es on es.event_id=e.event_id
left join sponsor s on s.sponsor_id=es.sponsor_id
left join bilet_eveniment be on be.event_id=e.event_id
where ge.galerie_id=id_galerie and ((e.data_incep between data_s and data_f) or (e.data_incheiere between data_s and data_f))
group by e.nume, e.data_incep, e.data_incheiere, s.nume_sponsor, s.nume_ceo) loop
    -- Afisarea informatiilor despre fiecare eveniment
    dbms_output.put_line('Eveniment: '||r.nume||
        ', Perioada: '||r.data_incep||' - '||r.data_incheiere ||
        ', Sponsor: '||nvl(r.nume_sponsor, 'Nespecificat') ||
        ', CEO Sponsor: '|| nvl(r.nume_ceo, 'Nespecificat') ||
        ', Bilete vandute: '||r.nr_bilete);
end loop;
exception
    when no_data_found then
        dbms_output.put_line('Nu exista galerie cu numele'||numele);
    when too_many_rows then
        dbms_output.put_line('Exista mai multe galerii cu numele'||numele);
    when no_events then
        dbms_output.put_line('Galeria'||numele||' nu este inclusa in niciun eveniment.');
    when invalid_interval then
        dbms_output.put_line('Perioada oferita ca parametru nu este valida. Al doilea parametru este data de inceput, iar al treilea este data de sfarsit.');
    when others then
        dbms_output.put_line('Eroare neasteptata!');
end;

```



Procedure GALERIE_DETALII compiled

Pentru urmatoarea secenta de cod se afiseaza:

```
begin
```

```
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-MAY-24','DD-MON-YY'),to_Date  
    ('12-NOV-24','DD-MON-YY'));
```

```
end;
```

```
begin
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-MAY-24','DD-MON-YY'),to_Date ('12-NOV-24','DD-MON-YY'));
end;
```

Script Output | Task completed in 0.062 seconds
PL/SQL procedure successfully completed.

Dbms Output | Buffer Size:20000

Local |

```
Eveniment: Noaptea Muzeelor, Perioada: 18-MAY-24 - 18-MAY-24, Sponsor: eMAG, CEO Sponsor: Iulian Stanciu, Bilete vândute: 2
Eveniment: Noaptea Muzeelor, Perioada: 18-MAY-24 - 18-MAY-24, Sponsor: BCR, CEO Sponsor: Sergiu Manea, Bilete vândute: 2
Eveniment: Festival de Arta, Perioada: 15-SEP-24 - 30-SEP-24, Sponsor: eMAG, CEO Sponsor: Iulian Stanciu, Bilete vândute: 1
Eveniment: Festival de Arta, Perioada: 15-SEP-24 - 30-SEP-24, Sponsor: Bitdefender, CEO Sponsor: Florin Talpes, Bilete vândute: 1
Eveniment: Zilele Culorii, Perioada: 10-NOV-24 - 20-NOV-24, Sponsor: Dacia, CEO Sponsor: Christophe Dridi, Bilete vândute: 1
Eveniment: Zilele Culorii, Perioada: 10-NOV-24 - 20-NOV-24, Sponsor: Bitdefender, CEO Sponsor: Florin Talpes, Bilete vândute: 1
```

```
begin
```

```
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-MAY-24','DD-MON-YY'),to_Date  
    ('12-NOV-24','DD-MON-YY'));
```

```
end;
```

```
begin
    Galerie_Detalii('Ceva',to_date ('01-MAY-24','DD-MON-YY'),to_Date ('12-NOV-24','DD-MON-YY'));
end;
```

Script Output | Task completed in 0.033 seconds
PL/SQL procedure successfully completed.

Dbms Output | Buffer Size:20000

Local |

```
Nu exista galerie cu numele Ceva
```

```
begin
```

```
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-MAY-24','DD-MON-YY'),to_Date  
    ('12-JAN-24','DD-MON-YY'));
```

```
end;
```

```
begin
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-MAY-24','DD-MON-YY'),to_Date ('12-JAN-24','DD-MON-YY'));
end;
```

Script Output | Task completed in 0.034 seconds
PL/SQL procedure successfully completed.

Dbms Output | Buffer Size:20000

Local |

```
Perioada oferita ca parametru nu este valida. Al doilea parametru este data de inceput, iar al treileia este data de sfarsit.
```

```
begin
```

```
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-DEC-24','DD-MON-YY'),to_Date ('12-  
    DEC-24','DD-MON-YY'));
```

```
end;
```



The screenshot shows the Oracle SQL Developer interface during the execution of a PL/SQL procedure. The code in the editor is:

```
begin
    Galerie_Detalii('Galeria de Arta Moderna',to_date ('01-DEC-24','DD-MON-YY'),to_Date ('12-DEC-24','DD-MON-YY'));
end;
```

The 'Script Output' tab shows the message: "Task completed in 0.032 seconds". The 'PL/SQL procedure successfully completed.' message is displayed in the main pane. The 'Dbms Output' tab shows the result: "Galeria Galeria de Arta Moderna nu este inclusa in niciun eveniment." (The gallery Galeria de Arta Moderna is not included in any event.)

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

Creati un trigger statement-level pe tabela exponat astfel incat daca se trece peste limita totala de 10 exponate, sa nu se mai poate insera exponate.

```
create or replace trigger Restrictie_Exponate before insert on exponat
```

```
declare
```

```
nr_exponete number;
```

```
begin
```

```
    select count(*) into nr_exponete from exponat;
```

```
    if nr_exponete>10 then
```

```
        raise_application_error(-20000,'S-a trecut peste limita totala de 10 exponate pe care galeriile le pot detine.');
```

```
    end if;
```

```
end;
```

The screenshot shows the SQL script editor with the trigger definition. The code is identical to the one above, with the exception of the trigger name 'Restrictie_Exponate'. Below the code, the 'Script Output' tab shows the message 'Task completed in 0.088 seconds'. At the bottom, it says 'Trigger RESTRICTIE_EXPONATE compiled'.

```
create or replace trigger Restrictie_Exponate before insert on exponat
declare
    nr_exponete number;
begin
    select count(*) into nr_exponete from exponat;
    if nr_exponete>10 then
        raise_application_error(-20000,'S-a trecut peste limita totala de 10 exponate pe care galeriile le pot detine.');
    end if;
end;
```

Aceasta este starea initiala a tablei Exponat:

v	v	v	v	v	v
1	6	1 Pictura sec XIX	Ciobanas cu turma de oi	1895	Nicolae Grigorescu
2	7	2 Sculptura sec XIX	Domnitorul Alexandru Ioan Cuza	1860	Ion Georgescu
3	8	3 Arta Populara sec XX	Covor Oltenesc	1920	Artisan Popular
4	9	4 Fotografie sec XX	Revolutia din 1989	1989	Fotoreporter Necunoscut
5	10	5 Arta Contemporana	Ingerul	2005	Ion Barladeanu
6	23	21 Pictura Moderna	Clar de luna	1990	Ion Tuculescu
7	24	2 Sculptura sec XIX	Bust Stefan cel Mare	1850	Ion Georgescu

Sunt scrise urmatoarele linii de inserare ce se executa in acelasi timp.

```
INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
```

```
(1, 'Pictura sec XIX', 'Cioban cu multe oi', 1905, 'Nicolae Grigorescu');
```

```
INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
```

(2, 'Sculptura sec XIX', 'Domnitorul Vlad Tepes', 1870, 'Ion Georgescu');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES

(3, 'Arta Populara sec XX', 'Covor Banatean', 1925, 'Artisan Popular');

INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES

(4, 'Fotografie sec XX', 'Doborarea Zidului Berlinului', 1983, 'Fotoreporter Necunoscut');

Daca ar fi de tip row-level, la ultima linie de inserare ar da eroare, pentru ca s-ar trece de limita impusa de 10 exponate. Cu toate acestea, deoarece trigger-ul este de tip statement-level, toate liniile sunt inserate.

```
INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(1, 'Pictura sec XIX', 'Cioban cu multe oi', 1895, 'Nicolae Grigorescu');
INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(2, 'Sculptura sec XIX', 'Domnitorul Vlad Tepes', 1870, 'Ion Georgescu');
INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(3, 'Arta Populara sec XX', 'Covor Oltenesc', 1920, 'Artisan Popular');
INSERT INTO Exponat (galerie_id, nume_categoria, nume, an, creator) VALUES
(4, 'Fotografie sec XX', 'Revolutia din 1989', 1989, 'Fotoreporter Necunoscut');

Script Output X
Task completed in 0.048 seconds
```

1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.

Tabela ajungand in urmatorul stagiu:

1	6	1 Pictura sec XIX	Ciobanas cu turma de oi	1895 Nicolae Grigorescu
2	7	2 Sculptura sec XIX	Domnitorul Alexandru Ioan Cuza	1860 Ion Georgescu
3	8	3 Arta Populara sec XX	Covor Oltenesc	1920 Artisan Popular
4	9	4 Fotografie sec XX	Revolutia din 1989	1989 Fotoreporter Necunoscut
5	10	5 Arta Contemporana	Ingerul	2005 Ion Barladeanu
6	23	21 Pictura Moderna	Clara de luna	1990 Ion Tuculescu
7	24	2 Sculptura sec XIX	Bust Stefan cel Mare	1850 Ion Georgescu
8	41	1 Pictura sec XIX	Cioban cu multe oi	1905 Nicolae Grigorescu
9	42	2 Sculptura sec XIX	Domnitorul Vlad Tepes	1870 Ion Georgescu
10	43	3 Arta Populara sec XX	Covor Banatean	1925 Artisan Popular
11	44	4 Fotografie sec XX	Doborarea Zidului Berlinului	1983 Fotoreporter Necunoscut

Daca se mai incercă inserarea altor exponate, se da raise la eroare :

```
Error report -
SQL Error: ORA-20000: S-a trecut peste limita totala de 10 exponate pe care galeriile le pot detine.
ORA-06512: at "PROIECT.RESTRICTIE_EXPONATE", line 6
ORA-04088: error during execution of trigger 'PROIECT.RESTRICTIE_EXPONATE'
20000. 00000 - "%s"
*Cause:   The stored procedure 'raise_application_error'
          was called which causes this error to be generated.
*Action:  Correct the problem as described in the error message or contact
          the application administrator or DBA for more information.
```

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

Creati un trigger de tip row-level pe tabela Galerie ce sa verifice inainte de inserare daca numele dat galeriei este unic si conditia ca atributul permanenta sa poata lua valori doar intre ‘Y’ si ‘N’.

```
create or replace trigger Verificare_Galerie before insert on Galerie for each row
```

```
declare
```

```
    nr_galerii number;
```

```
begin
```

```
    select count(*) into nr_galerii from galerie g
```

```
        where upper(trim(g.nume))=upper(trim(:new.nume));
```

```
    if nr_galerii>0 then
```

```
        raise_application_error(-20000,'Numele galeriilor trebuie sa fie unic.');
```

```
    end if;
```

```
    if upper(trim(:new.permanenta))='Y' then
```

```
        if :new.data_incheiere is not null then
```

```
            raise_application_error(-20001,'Galeria este trecuta ca permanenta, insa a fost adaugata o  
data de incheiere.');
```

```
        end if;
```

```
    elsif upper(trim(:new.permanenta))='N' then
```

```
        if :new.data_incheiere is null then
```

```
            raise_application_error(-20002,'Galeria este trecuta ca temporală, insa nu exista vreo data  
de incheiere.');
```

```
        end if;
```

```
    else
```

```
        raise_application_error(-20003,'Atributul permanenta poate sa aiba doar doua valori, Y daca  
este permanenta si N daca este temporală.');
```

```
    end if;
```

```
end;
```

```

create or replace trigger Verificare_Galerie before insert on Galerie for each row
declare
    nr_galerii number;
begin
    select count(*) into nr_galerii from galerie g
        where upper(trim(g.nume))=upper(trim(:new.nume));
    if nr_galerii>0 then
        raise_application_error(-20000,'Numele galeriilor trebuie sa fie unic.');
    end if;
    if upper(trim(:new.permanenta))='Y' then
        if :new.data_incheiere is not null then
            raise_application_error(-20001,'Galeria este trecuta ca permanenta, insa a fost adaugata o data de incheiere.');
        end if;
    elsif upper(trim(:new.permanenta))='N' then
        if :new.data_incheiere is null then
            raise_application_error(-20002,'Galeria este trecuta ca temporală, insa nu exista vreodata de incheiere.');
        end if;
    else
        raise_application_error(-20003,'Atributul permanenta poate sa aiba doar doua valori, Y daca este permanenta si N daca este temporală.');
    end if;
end;

```

Script Output X | Task completed in 0.032 seconds

Trigger VERIFICARE_GALERIE compiled

Daca avem linia de insert :

INSERT INTO Galerie (nume, permanenta)

VALUES ('Galeria de Arta Moderna', 'Y');

```

INSERT INTO Galerie (nume, permanenta)
VALUES ('Galeria de Arta Moderna', 'Y');

```

Script Output X | Task completed in 0.037 seconds

Error starting at line : 13 in command -
 INSERT INTO Galerie (nume, permanenta)
 VALUES ('Galeria de Arta Moderna', 'Y')
 Error at Command Line : 13 Column : 11
 Error report -
 SQL Error: ORA-20000: Numele galeriilor trebuie sa fie unic.
 ORA-06512: at "PROIECT.VERIFYCAR_UNICITATE_NUME_GALERIE", line 7
 ORA-04088: error during execution of trigger 'PROIECT.VERIFYCAR_UNICITATE_NUME_GALERIE'
 20000. 00000 - "%s"
 *Cause: The stored procedure 'raise_application_error'
 was called which causes this error to be generated.
 *Action: Correct the problem as described in the error message or contact
 the application administrator or DBA for more information.

Daca avem linia de insert ce nu respecta constrangerea pentru permanenta :

INSERT INTO Galerie (nume, permanenta)

VALUES ('Galeria de Arta', 'C');

```

INSERT INTO Galerie (nume, permanenta)
VALUES ('Galeria de Arta', 'C');

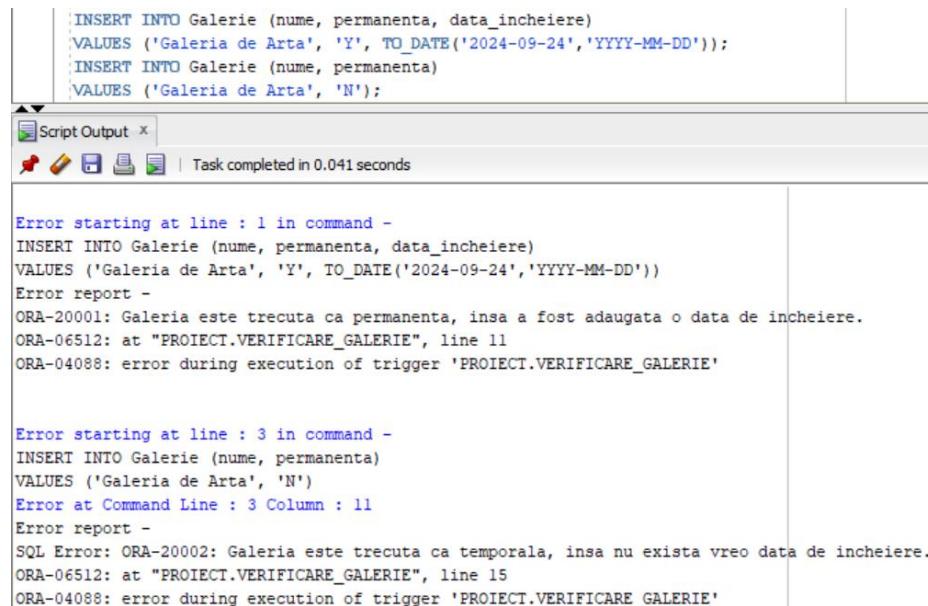
```

Script Output X | Task completed in 0.014 seconds

Error starting at line : 1 in command -
 INSERT INTO Galerie (nume, permanenta)
 VALUES ('Galeria de Arta', 'C')
 Error at Command Line : 1 Column : 11
 Error report -
 SQL Error: ORA-20003: Atributul permanenta poate sa aiba doar doua valori, Y daca este permanenta si N daca este temporală.
 ORA-06512: at "PROIECT.VERIFYCAR_GALERIE", line 18
 ORA-04088: error during execution of trigger 'PROIECT.VERIFYCAR_GALERIE'

Daca avem linia de insert prin care nu se respecta corelatia dintre data_incheiere si permanenta :

```
INSERT INTO Galerie (nume, permanenta, data_incheiere)
VALUES ('Galeria de Arta', 'Y', TO_DATE('2024-09-24','YYYY-MM-DD'));
INSERT INTO Galerie (nume, permanenta)
VALUES ('Galeria de Arta', 'N');
```



```
INSERT INTO Galerie (nume, permanenta, data_incheiere)
VALUES ('Galeria de Arta', 'Y', TO_DATE('2024-09-24','YYYY-MM-DD'));
INSERT INTO Galerie (nume, permanenta)
VALUES ('Galeria de Arta', 'N');

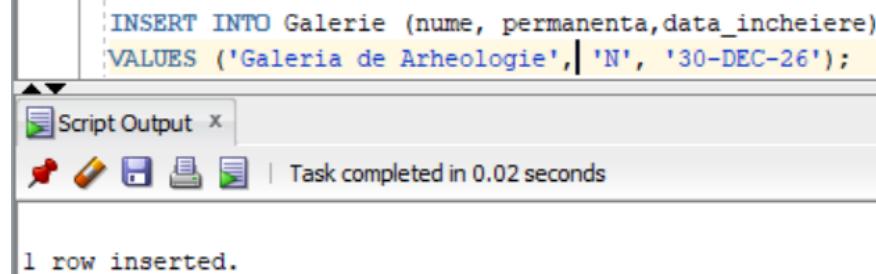
Script Output X
Task completed in 0.041 seconds

Error starting at line : 1 in command -
INSERT INTO Galerie (nume, permanenta, data_incheiere)
VALUES ('Galeria de Arta', 'Y', TO_DATE('2024-09-24','YYYY-MM-DD'))
Error report -
ORA-20001: Galeria este trecuta ca permanenta, insa a fost adaugata o data de incheiere.
ORA-06512: at "PROIECT.VERIFICARE_GALERIE", line 11
ORA-04088: error during execution of trigger 'PROIECT.VERIFICARE_GALERIE'

Error starting at line : 3 in command -
INSERT INTO Galerie (nume, permanenta)
VALUES ('Galeria de Arta', 'N')
Error at Command Line : 3 Column : 11
Error report -
SQL Error: ORA-20002: Galeria este trecuta ca temporala, insa nu exista vreo data de incheiere.
ORA-06512: at "PROIECT.VERIFICARE_GALERIE", line 15
ORA-04088: error during execution of trigger 'PROIECT.VERIFICARE_GALERIE'
```

Daca avem ceva care respecta conditiile se insereaza normal :

```
INSERT INTO Galerie (nume, permanenta,data_incheiere)
VALUES ('Galeria de Arheologie', 'N', '30-DEC-26');
```



```
INSERT INTO Galerie (nume, permanenta,data_incheiere)
VALUES ('Galeria de Arheologie', 'N', '30-DEC-26');

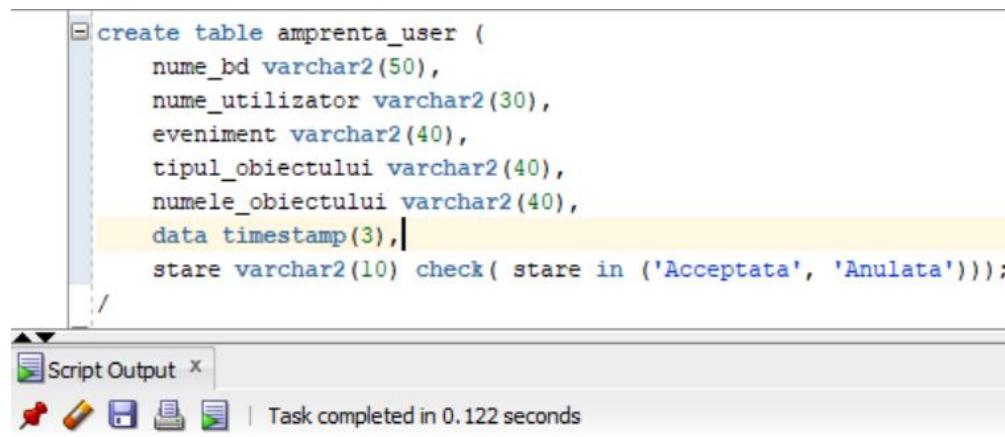
Script Output X
Task completed in 0.02 seconds

1 row inserted.
```

Cerinta 12: Definiți un trigger de tip LDD. Declanșați trigger-ul.

Creati un trigger de tip LDD ce sa insereze toate modificarile aduse schemei de catre utilizator, permitand ca acestea sa fie realizate doar daca ora efectuarii este in timpul programului de lucru (8 -18).

```
create table amprenta_user (
    nume_bd varchar2(50),
    nume_utilizator varchar2(30),
    eveniment varchar2(40),
    tipul_objectului varchar2(40),
    numele_objectului varchar2(40),
    data timestamp(3),
    stare varchar2(10) check( stare in ('Acceptata', 'Anulata'));
```



```
create table amprenta_user (
    nume_bd varchar2(50),
    nume_utilizator varchar2(30),
    eveniment varchar2(40),
    tipul_objectului varchar2(40),
    numele_objectului varchar2(40),
    data timestamp(3),|
    stare varchar2(10) check( stare in ('Acceptata', 'Anulata'));
```

Script Output X

Table AMPRENTA_USER created.

```
create or replace trigger AuditSiOre_de_Lucru before create or alter or drop on schema
declare
    ora number;
    pragma autonomous_transaction;
--Am utilizat aceasta instructiune deoarece fara ea raise_application_error intrerupea orice
--tranzactie, inclusive cele de tip audit pe care voiam sa le introduce in tabela Amprenta_User.
begin
```

```

ora:= to_number(to_char(sysdate,'HH24'));

insert into amprenta_user
values (
    sys.database_name,
    sys.login_user,
    sys.sysevent,
    sys.dictionary_obj_type,
    sys.dictionary_obj_name,
    current_timestamp,
    case
        when ora not between 8 and 17 then 'Anulata'
        else 'Acceptata'
    end
);

commit;

if ora not between 8 and 17 then
    raise_application_error(-20000, 'Nu sunt permise modificari asupra bazei de date in orele in care nu se lucreaza.');
end if;

```

end;

```

create or replace trigger AuditSiOre_de_Lucru before create or alter or drop on schema
declare
    ora number;
    pragma autonomous_transaction;
begin
    ora:= to_number(to_char(sysdate,'HH24'));
    insert into amprenta_user
    values (
        sys.database_name,
        sys.login_user,
        sys.sysevent,
        sys.dictionary_obj_type,
        sys.dictionary_obj_name,
        current_timestamp,
        case
            when ora not between 8 and 17 then 'Anulata'
            else 'Acceptata'
        end
    );
    commit;
    if ora not between 8 and 17 then
        raise_application_error(-20000, 'Nu sunt permise modificari asupra bazei de date in orele in care nu se lucreaza.');
    end if;
end;

```

```

drop table exemplu;
create table exemplu (col1 number,col2 varchar2(20));

```

The screenshot shows the SQL script editor with the following code:

```

| drop table exemplu;
| create table exemplu (col1 number,col2 varchar2(20));

```

Below the editor is the Script Output window with the message "Task completed in 0.218 seconds".

Table EXEMPLU dropped.

Table EXEMPLU created.

NSOR	NUME_BD	NUME_UTILIZATOR	EVENIMENT	TIPUL_OBIECTULUI	NUMELE_OBIECTULUI	DATA	STARE
1	ORCL2	PROIECT	CREATE	TABLE	EXEMPLU	02-JAN-25 05.53.41.395000000 PM	Acceptata
2	ORCL2	PROIECT	DROP	TABLE	EXEMPLU	02-JAN-25 05.58.08.584000000 PM	Acceptata
3	ORCL2	PROIECT	CREATE	TABLE	EXEMPLU	02-JAN-25 05.58.08.780000000 PM	Acceptata

Daca dupa ora 18 se incearca alte instructiuni LDD, precum un drop, da eroare :

The screenshot shows the SQL script editor with the following code:

```

| drop table exemplu;

```

Below the editor is the Script Output window with the message "Task completed in 0.023 seconds".

The Query Result window displays the following error report:

```

Error report -
ORA-04088: error during execution of trigger 'PROIECT.AUDITSIORE_DE_LUCRU'
ORA-00604: error occurred at recursive SQL level 1
ORA-20000: Nu sunt permise modificari asupra bazei de date in orele in care nu se lucreaza.
ORA-06512: at line 9
04088. 00000 - "error during execution of trigger '%s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.

```

Si asa arata tabela :

NSOR	NUME_BD	NUME_UTILIZATOR	EVENIMENT	TIPUL_OBIECTULUI	NUMELE_OBIECTULUI	DATA	STARE
1	ORCL2	PROIECT	CREATE	TABLE	EXEMPLU	02-JAN-25 06.50.03.781000000 PM	Anulata
2	ORCL2	PROIECT	DROP	TABLE	EXEMPLU	02-JAN-25 05.58.08.584000000 PM	Acceptata
3	ORCL2	PROIECT	CREATE	TABLE	EXEMPLU	02-JAN-25 05.58.08.780000000 PM	Acceptata
4	ORCL2	PROIECT	DROP	TABLE	EXEMPLU	02-JAN-25 06.49.11.157000000 PM	Anulata
5	ORCL2	PROIECT	DROP	TABLE	EXEMPLU	02-JAN-25 06.50.01.798000000 PM	Anulata
6	ORCL2	PROIECT	ALTER	TABLE	EXEMPLU	02-JAN-25 06.50.06.363000000 PM	Anulata

Cerinta 13 : Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

Sa se creeze un pachet de gestionare a inventarului galeriilor muzeului cand acestea sunt incluse in diferite evenimente, in care se introduce o noua posibilitate pentru exponate, anume acestea sa fie intr-o stare ce necesita **restaurare**. Restaurarea va incepe de la o anumita data si va dura un anumit nr de zile, in care exponatul nu va fi prezent in repectiva galerie. Modul prin care se decide daca un exponat necesita restaurarea este prin **numele categoriei si id-ul galeriei** din care face parte. Totodata, va aparea o noua relatie numita „**Expus**”, ce are loc intre exponat, galerie si eveniment, cu conditia ca respectivul exponat sa nu fie in proces de restaurare pe parcursul desfasurarii evenimentului. Scopul pachetului este aflarea numarului total de exponate ce vor fi prezentate in cadrul unui eveniment, dar si numarul exponatelor per fiecare galerie inclusa in eveniment.

Header

```
create or replace package Inventar_Eventamente as
```

```
    type Expus is record(  
        exponat_id exponat.exponat_id%type,  
        galerie_id galerie.galerie_id%type,  
        event_id eveniment.event_id%type  
    );
```

```
    type Restaurare is record(  
        exponat_id exponat.exponat_id%type,  
        data_inceput date,  
        durata number  
    );
```

```
--Folosit drept tipul listei returnate de functie nr_exp_galerie, reprezentand perechea formata dintr-un nume de galerie si nr-ul de exponate din acea galerie ce sunt eligibile pentru un anumit eveniment
```

```
    type returnare is record(
```

```

nr_exponate number,
nume_galerie galerie.nume%type);

type tablou_returnare is table of returnare;
type tablou_expus is table of Expus;
type tablou_restaurare is table of Restaurare;
lista_expus tablou_expus:=tablou_expus();
list_restaurare tablou_restaurare:=tablou_restaurare();
procedure resetare_tablouri;
procedure adauga_expus (id_exp in exponat.exponat_id%type,
                        id_gal in galerie.galerie_id%type,
                        id_event in eveniment.event_id%type);
procedure setare_restaurare (nume in categorie.nume_categorie%type,
                            inceput in date,
                            dur in number);
procedure raport(numele in eveniment.nume%type);
function nr_exp_event(id_event in eveniment.event_id%type) return number;
function nr_exp_galerie(id_event in eveniment.event_id%type) return tablou_returnare;
function eligibil (id_event eveniment.event_id%type,
                   data_s in date,
                   data_t in date) return number;
end Inventar_Evenimente;

```

```

create or replace package Inventar_Evenimente as
    type Expus is record(
        exponat_id exponat.exponat_id%type,
        galerie_id galerie.galerie_id%type,
        event_id eveniment.event_id%type
    );
    type Restaurare is record(
        exponat_id exponat.exponat_id%type,
        data_inceput date,
        durata number
    );
    type returnare is record(
        nr_exponate number,
        nume_galerie galerie.nume%type);
    type tablou_returnare is table of returnare;
    type tablou_expus is table of Expus;
    type tablou_restaurare is table of Restaurare;
    lista_expus tablou_expus:=tablou_expus();
    list_restaurare tablou_restaurare:=tablou_restaurare();
    procedure resetare_tablouri;
    procedure adauga_expus (id_exp in exponat.exponat_id%type,
                           id_gal in galerie.galerie_id%type,
                           id_event in eveniment.event_id%type);
    procedure setare_restaurare (nume in categorie.nume_categorie%type,
                                 inceput in date,
                                 dur in number);
    procedure raport(numele in eveniment.nume%type);

        function nr_exp_event(id_event in eveniment.event_id%type) return number;
        function nr_exp_galerie(id_event in eveniment.event_id%type) return tablou_returnare;
        function eligibil (id_event eveniment.event_id%type,
                           data_s in date,
                           data_t in date) return number;
    end Inventar_Evenimente;
/

```

Script Output X | Task completed in 0.05 seconds

Package INVENTAR_EVENTIMENTE compiled

Body

create or replace package body Inventar_Evenimente as

--Utilizata pentru golirea celor doua colectii

procedure resetare_tablouri is

begin

list_restaurare:=tablou_restaurare();

lista_expus:=tablou_expus();

end resetare_tablouri;

--Toate exponatele ce sunt din categoria primita ca parametru sunt programate pentru restaurare

procedure setare_restaurare (nume in categorie.nume_categorie%type,

inceput in date,

dur in number) is

```

numele categorie.nume_categorie%type;
numar number;

begin
    numele:=nume;
    select count(*) into numar from categorie where
trim(upper(nume_categorie))=trim(upper(num)));
    if numar=0 then
        raise no_data_found;
    end if;
    for e in (select exponat_id from exponat where
trim(upper(nume_categorie))=trim(upper(nume))) loop
        list_restaurare.extend;
        list_restaurare(list_restaurare.count):=restaurare(
            exponat_id => e.exponat_id,
            data_inceput => incep,
            durata => dur);
    end loop;
exception
    when no_data_found then
        dbms_output.put_line('Nu exista categorie cu numele'||nume);
    end setare_restaurare;
--Se adauga in tabloul expus daca este eligibil respectivul exponat
procedure adauga_expus (id_exp in exponat.exponat_id%type,
                        id_gal in galerie.galerie_id%type,
                        id_event in eveniment.event_id%type) is
begin
    for i in 1..list_restaurare.count loop
        if list_restaurare(i).exponat_id=id_exp then

```

```

        if eligibil (id_event, list_restaurare(i).data_inceput,
list_restaurare(i).data_inceput+list_restaurare(i).durata) =0 then
            return;
        end if;
    end if;
end loop;

lista_expus.extend;
lista_expus(lista_expus.count):=expus(id_exp,id_gal,id_event);
end adauga_expus;

--Se returneaza nr-ul de exponete valabile pentru evenimentul primit ca parametru
function nr_exp_event(id_event in eveniment.event_id%type) return number as
contor number:=0;
begin
    for i in 1..lista_expus.count loop
        if lista_expus(i).event_id=id_event then
            contor:=contor+1;
        end if;
    end loop;
    return contor;
end nr_exp_event;

--Se verifica daca un exponent este eligibil pentru un anumit eveniment
function eligibil (id_event in eveniment.event_id%type,
                    data_s in date,
                    data_t in date) return number is
sfarsit date;
inceput date;
begin
    select data_inceput,data_incheiere into inceput,sfarsit from eveniment where
event_id=id_event;

```

```
if not ((inceput between data_s and data_t) or (sfarsit between data_s and data_t) or (data_s  
between inceput and sfarsit) or (data_t between inceput and sfarsit)) then
```

```
    return 1;
```

```
else
```

```
    return 0;
```

```
end if;
```

```
end eligibil;
```

```
--Se returneaza o lista compusa din perechi de numele galeriei si nr-ul exponatelor valide de la  
acea galerie
```

```
function nr_exp_galerie(id_event in eveniment.event_id%type) return tablou_returnare is
```

```
    lista_returnare tablou_returnare:=tablou_returnare();
```

```
    numar number;
```

```
begin
```

```
    for g in (select ge.galerie_id, gal.nume from galerie_eveniment ge join galerie gal on  
ge.galerie_id=gal.galerie_id where event_id=id_event) loop
```

```
        numar:=0;
```

```
        for i in 1..lista_expus.count loop
```

```
            if lista_expus(i).galerie_id=g.galerie_id and lista_expus(i).event_id=id_event then
```

```
                numar:=numar+1;
```

```
            end if;
```

```
        end loop;
```

```
        lista_returnare.extend;
```

```
        lista_returnare(lista_returnare.count):=returnare(numar,g.nume);
```

```
    end loop;
```

```
    return lista_returnare;
```

```
end;
```

```
--Procedura in care sunt folosite functiile nr_exp_event si nr_exp_galerie si se prezinta un raport  
al nr-ului de exponate si nr de exponate per fiecare galerie pentru un eveniment primit ca  
parametru
```

```
procedure raport(numele in eveniment.nume%type) as
```

```

id_event eveniment.event_id%type;
lista_returnare tablou_returnare:=tablou_returnare();
begin
    select event_id into id_event from eveniment e where e.numel=numele;
    dbms_output.put_line('Evenimentul "'||numele||" are "'||nr_exp_event(id_event)||" exponete ce
vor putea fi vazute.');
    dbms_output.put_line('Acestea sunt galeriile si cate exponete au fiecare: ');
    lista_returnare:=nr_exp_galerie(id_event);
    for i in 1..lista_returnare.count loop
        dbms_output.put_line('Galerie:'||lista_returnare(i).nume_galerie||' Nr:
'||lista_returnare(i).nr_exponate);
    end loop;
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu acest nume.');
    when too_many_rows then
        dbms_output.put_line('Prea multe evenimente cu acelasi nume.');
end report;
end Inventar_Evenimente;
/

```

```

create or replace package body Inventar_Evenimente as
procedure resetare_tablouri is
begin
    list_restaurare:=tablou_restaurare();
    lista_expus:=tablou_expus();
end resetare_tablouri;

procedure setare_restaurare (nume in categorie.nume_categorie%type,
                            inceput in date,
                            dur in number) is
    numele categorie.nume_categorie%type;
    numar number;
begin
    numele:=nume;
    select count(*) into numar from categorie where trim(upper(nume_categorie))=trim(upper(nume));
    if numar=0 then
        raise no_data_found;
    end if;
    for e in (select exponat_id from exponat where trim(upper(nume_categorie))=trim(upper(numele))) loop
        list_restaurare.extend;
        list_restaurare(list_restaurare.count):=restaurare(
            exponat_id => e.exponat_id,
            data_inceput => inceput,
            durata => dur);
    end loop;
exception

```

```

when no_data_found then
    dbms_output.put_line('Nu exista categorie cu numele'||nume);
end setare_restaureare;

procedure adauga_expus (id_exp in exponat.exponat_id%type,
                        id_gal in galerie.galerie_id%type,
                        id_event in eveniment.event_id%type) is
begin
    for i in 1..list_restaureare.count loop
        if list_restaureare(i).exponat_id=id_exp then
            if eligibil(id_event,list_restaureare(i).data_inceput,list_restaureare(i).data_inceput+list_restaureare(i).durata)=0 then
                return;
            end if;
        end if;
    end loop;
    lista_expus.extend;
    lista_expus(lista_expus.count):=expus(id_exp,id_gal,id_event);
end adauga_expus;

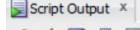
function nr_exp_event(id_event in eveniment.event_id%type) return number as
contor number:=0;
begin
    for i in 1..lista_expus.count loop
        if lista_expus(i).event_id=id_event then
            contor:=contor+1;
        end if;
    end loop;

    return contor;
end nr_exp_event;

function eligibil (id_event in eveniment.event_id%type,
                   data_s in date,
                   data_t in date) return number is
    sfarsit date;
    inceput date;
begin
    select data_inceput,data_incheiere into inceput,sfarsit from eveniment where event_id=id_event;
    if not ((inceput between data_s and data_t) or (sfarsit between data_s and data_t)
            or (data_s between inceput and sfarsit) or (data_t between inceput and sfarsit)) then
        return 1;
    else
        return 0;
    end if;
end eligibil;

function nr_exp_galerie(id_event in eveniment.event_id%type) return tablou_returnare is
    lista_returnare tablou_returnare:=tablou_returnare();
    numar number;
begin
    for g in (select ge.galerie_id, gal.nume from galerie_eveniment ge join galerie gal on ge.galerie_id=gal.galerie_id where event_id=id_event) loop
        numar:=0;
        for i in 1..lista_expus.count loop
            if lista_expus(i).galerie_id=g.galerie_id and lista_expus(i).event_id=id_event then
                numar:=numar+1;
            end if;
        end loop;
        lista_returnare.extend;
        lista_returnare(lista_returnare.count):=returnare(numar,g.nume);
    end loop;
    return lista_returnare;
end;
procedure raport(numele in eveniment.nume%type) as
    id_event eveniment.event_id%type;
    lista_returnare tablou_returnare:=tablou_returnare();
begin
    select event_id into id_event from eveniment e where e.nume=numele;
    dbms_output.put_line('Evenimentul'||numele||' are'||nr_exp_event(id_event)||' exponate ce vor putea fi vazute.');
    dbms_output.put_line('Acestea sunt galerile si cate exponate au fiecare: ');
    lista_returnare:=nr_exp_galerie(id_event);
    for i in 1..lista_returnare.count loop
        dbms_output.put_line('Galerie:'||lista_returnare(i).nume_galerie||' Nr: '||lista_returnare(i).nr_exponate);
    end loop;
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu acest nume.');
    when too_many_rows then
        dbms_output.put_line('Prea multe evenimente cu acelasi nume.');
    end raport;
end Inventar_Evenimente;

```



| Task completed in 0.058 seconds

Package Body INVENTAR EVENIMENTE compiled

Blocuri de Verificare:

Acesta va fi prototipul blocurilor de schimbar, diferenta fiind doar numele categoriei si numele evenimentului

declare

```
    nume_categorie categorie.nume_categorie%type:='Sculptura sec XIX';
    nume_eveniment eveniment.nume%type:='Eveniment';
    id_eveniment eveniment.event_id%type;
    d date:=to_date('10-MAY-24','DD-MON-YY');
    durata number:=20;

begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categorie,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele'||nume_eveniment);
end;
```

Nu exista evenimentul

```

declare
    nume_categoria categorie.nume_categoria&type:='Sculptura sec XIX';
    nume_eveniment eveniment.nume&type:='Eveniment';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('10-MAY-24','DD-MON-YY');
    durata number:=20;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment');
end;

```

Nu exista eveniment cu numele Eveniment

Nu exista categoria

```

declare
    nume_categoria categorie.nume_categoria&type:='Ceva';
    nume_eveniment eveniment.nume&type:='Expozitia de Vara';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('28-MAY-24','DD-MON-YY');
    durata number:=1;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment');
end;

```

Nu exista categoria cu numele Ceva
 Evenimentul Expozitia de Vara are 4 exponate ce vor putea fi vazute.
 Acestea sunt galeriile si cate exponate au fiecare:
 Galerie:Galeria de Istorie Antica Nr: 3
 Galerie:Galeria Art Safari Nr: 1

Cand apare influenta restaurarilor

```

declare
    nume_categoria categorie.nume_categoria&type:='Pictura Moderna';
    nume_eveniment eveniment.nume&type:='Expozitia de Vara';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('28-MAY-24','DD-MON-YY');
    durata number:=10;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment');
end;

```

Evenimentul Expozitia de Vara are 3 exponate ce vor putea fi vazute.
 Acestea sunt galeriile si cate exponate au fiecare:
 Galerie:Galeria de Istorie Antica Nr: 3
 Galerie:Galeria Art Safari Nr: 0

Mod normal

```

declare
    nume_categoria categorie.nume_categoria&type:='Pictura Moderna';
    nume_eveniment eveniment.nume&type:='Expozitia de Vara';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('28-MAY-24','DD-MON-YY');
    durata number:=1;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment');
end;

```

Evenimentul Expozitia de Vara are 4 exponate ce vor putea fi vazute.
 Acestea sunt galeriile si cate exponate au fiecare:
 Galerie:Galeria de Istorie Antica Nr: 3
 Galerie:Galeria Art Safari Nr: 1

Cand nu este influentat de restaurarile exponatelor

```

declare
    nume_categoria categorie.nume_categoria&type:='Pictura Moderna';
    nume_eveniment eveniment.nume&type:='Noaptea Muzeelor';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('28-MAY-24','DD-MON-YY');
    durata number:=1;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment);
end;

```

Evenimentul Noaptea Muzeelor are 10 exponate ce vor putea fi vazute.
 Acestea sunt galeriile si cate exponate au fiecare:
 Galerie:Galeria de Arta Moderna Nr: 2
 Galerie:Galeria de Istorie Antica Nr: 3
 Galerie:Galeria de Stiinta Nr: 2
 Galerie:Galeria de Arta Contemporana Nr: 2
 Galerie:Galeria de Fotografie Nr: 1

Cand apare influenta restaurarilor

```

declare
    nume_categoria categorie.nume_categoria&type:='Sculptura sec XIX';
    nume_eveniment eveniment.nume&type:='Noaptea Muzeelor';
    id_eveniment eveniment.event_id&type;
    d date:=to_date('10-MAY-24','DD-MON-YY');
    durata number:=20;
begin
    Inventar_Evenimente.resetare_tablouri;
    select event_id into id_eveniment from eveniment where nume=nume_eveniment;
    Inventar_Evenimente.setare_restaureaza(nume_categoria,d,durata);
    for g in (select galerie_id from galerie_eveniment where event_id=id_eveniment) loop
        for r in (select e.exponat_id as idul from exponat e where e.galerie_id=g.galerie_id) loop
            Inventar_Evenimente.adauga_expus(r.idul,g.galerie_id,id_eveniment);
        end loop;
    end loop;
    Inventar_Evenimente.raport(nume_eveniment);
exception
    when no_data_found then
        dbms_output.put_line('Nu exista eveniment cu numele ''||nume_eveniment);
end;

```

Evenimentul Noaptea Muzeelor are 7 exponate ce vor putea fi vazute.
 Acestea sunt galeriile si cate exponate au fiecare:
 Galerie:Galeria de Arta Moderna Nr: 2
 Galerie:Galeria de Istorie Antica Nr: 0
 Galerie:Galeria de Stiinta Nr: 2
 Galerie:Galeria de Arta Contemporana Nr: 2
 Galerie:Galeria de Fotografie Nr: 1